1 The Network Machine Learning Landscape

We begin by exploring the network machine learning landscape. We start with high-level concepts about networks, introducing basic terminology required to understand network data. This will make Chapter 3 more digestible when we formally introduce network data structures and how to manipulate them.

We also describe the different types of problems in network machine learning, and how network data fit into the types of problems practitioners need to address.

This chapter covers the following:

1. How do networks fit into the world of machine learning, and why are they important?
2. What does it mean to learn from a network, and what kinds of things should we try to learn?
3. How can we use network-valued data to understand the world better?
4. What challenges might we encounter when we analyze network-valued data?

This high-level understanding provides the context necessary for understanding the practical significance of the methods that we will develop throughout the book.

1.1 What Is Network Machine Learning?

Machine learning is a field of inquiry devoted to understanding and building methods that learn; that is, methods that leverage data to improve predictive performance on some set of tasks [1].

Machine learning has grown enormously over the past few decades, and its use-cases are rapidly pervading modern life. For instance, a photo editor working in pattern recognition might want to automatically segment an object so that they can blur the background (see Figure 1.1.1). An audio engineer might want to identify and separate unique instruments in a song. A student might have software draft an essay. All of these examples use data to learn a pattern that can be leveraged to accomplish some task.

1.1.1 Traditional Machine Learning Leverages Tabular Data Structures

In traditional machine learning, data follow a *tabular format*. The data are arranged in a table or array, where each row represents a single observation or data point, and each column represents a feature or dimension. This tabular structure is convenient

https://doi.org/10.1017/9781009405379.003 Published online by Cambridge University Press

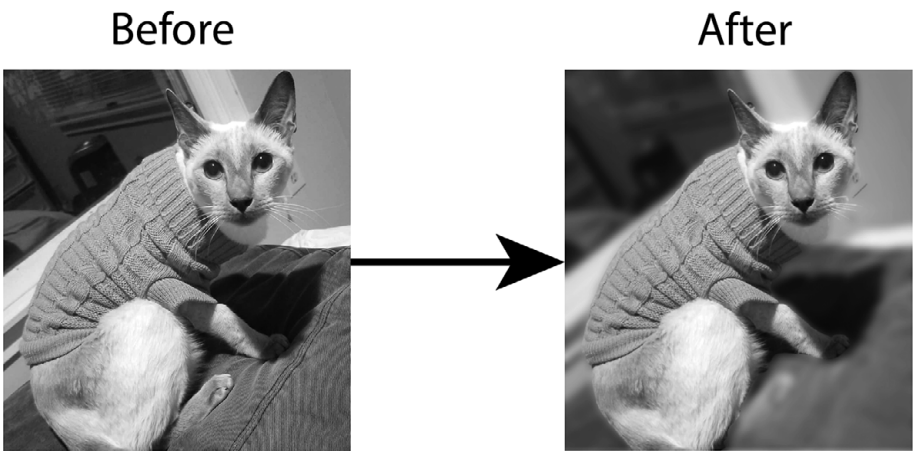


Figure 1.1.1 Learning how to segment an image to blur the background. A machine learning system is trained using numerous images with the foreground segmented out. This trained system is then used to segment out the foreground on new images, and then the background is blurred.

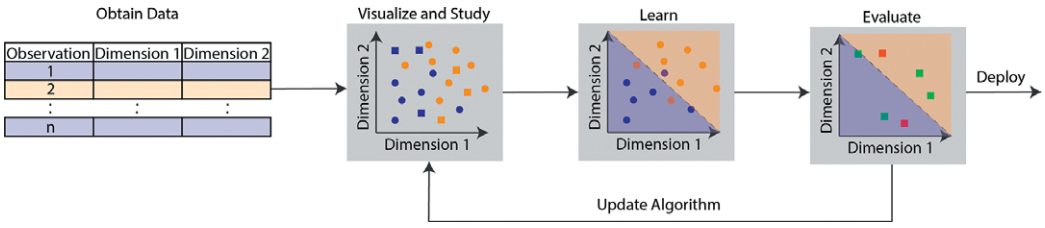


Figure 1.1.2 Machine learning systems start by obtaining inputs as tabular data, where the rows are observations and the columns are features, i.e., *dimensions* of each observation.

because it allows techniques developed in one domain of machine learning to be modified or applied to problems in another domain without reinventing the wheel. For example, given a tabular dataset where each row represents a particular lobster with columns representing its length and sex, we might try to predict the lobster's claw size. We could accomplish this by fitting a linear regression model for the claw size of other lobsters depending on their length and biological sex. For a more detailed discussion of tabular data structures, we recommend the `pandas` tutorial on tabular data [2].

In Figure 1.1.2, we explore how tabular data are used in a machine learning system using a basic classification task. Each observation is either blue or orange. Some of the data (the training set, circles) are used to train a machine learning algorithm (learning, the transparent blue and orange cells above and below the diagonal), and the remainder of the data are used to test the trained algorithm on new data (evaluating, the new colored red and green squares). The trained model can then be further refined (updated), or deployed for an intended use-case.

https://doi.org/10.1017/9781009405379.003 Published online by Cambridge University Press

1.1.2 What Is a Network?

Many people have some notion of a "network" as describing a system like the internet, or cell phone towers transmitting data. Networks have a specific definition in machine learning and data science. In a network:

1. We have a group of items (e.g., people in a social network), and
2. These items are interconnected through clearly defined relationships (e.g., which people are friends with one another).

Unlike tabular data where each row represents an independent observation, networks capture relationships between nodes. These connections cannot be easily represented in the tabular format without losing crucial structural information. This necessitates specialized approaches in network science for analysis and learning.

Networks can also be referred to as graphs. In this book, we primarily use the term "network" rather than "graph" to avoid confusion with the other use of the word graph (plots on x/y coordinate axes). However, we may encounter the word "graph" in some algorithm or tool names.

Each object in a network is called a *node*, or a vertex (for consistency we'll stick to "node" in this book). A connection between two nodes is called an *edge*. Figure 1.1.3(A) illustrates a simple network of business with eight nodes, with edges indicating transactions. We observe that some nodes are well-connected with other nodes – for example, the node in the center has edges with four other nodes – and we also have nodes which are not connected to anything, like the disconnected business.

Networks might also contain extra information. For example, each node might be associated with a feature vector. For instance, in the business transaction network shown in Figure 1.1.3, we might have information about the company size. We could reasonably assume that larger companies tend to have more business transactions with other companies. Figure 1.1.3(B) illustrates this scenario, where the network includes company size information represented by the size of each node. In this case, each node is associated with a scalar value (the company size), which can be thought of as a length-one vector.

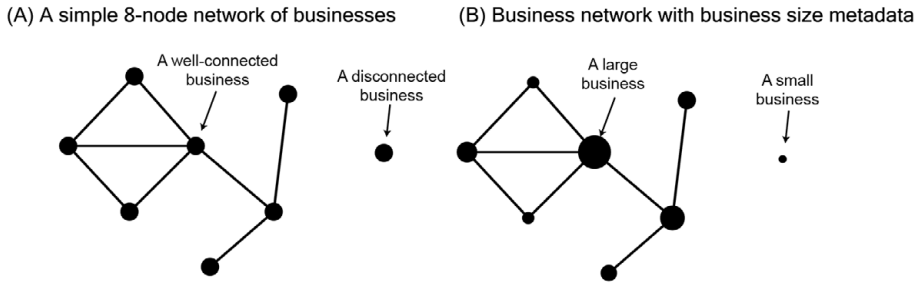


Figure 1.1.3 Panel **(A)** indicates a simple network, where nodes are businesses, and edges exist if a given pair of businesses transact with one another. Panel **(B)** indicates the same simple network, but each node also has a piece of information attached to it, the number of employees (indicated by the node size).

https://doi.org/10.1017/9781009405379.003 Published online by Cambridge University Press

1.1.3 Why Do We Study Networks?

Networks are ubiquitous and can be used to model almost any scenario involving objects interacting with each other. Some examples include:

- Social networks, where nodes represent people and edges represent friendships.
- Computer networks, where nodes represent computers and edges represent data transmission.
- Air traffic networks, where nodes represent cities and edges represent flights between them.
- Infection networks, where nodes represent people and edges represent disease transmissions.
- Brain networks in neuroscience, where nodes represent neurons and edges represent their white matter connections.

More examples include ecological networks, electrical networks, and gene networks. We frequently use networks to represent mathematical relationships:

- In Bayesian networks, nodes represent variables and edges represent conditional dependencies.
- Correlation networks can be created from data, where nodes represent variables and edges represent their correlations.
- We can create semantic networks in natural language processing, where nodes represent concepts and edges represent their semantic distance from one another.
- Neural networks in computer science, where nodes are artificial neurons and edges are weights between them.
- Attention networks in large language models like ChatGPT, where nodes are tokens (parts of words), and edges define the information that should be transferred between them.
- More generally, we can create Euclidean networks from any tabular dataset by representing data points as nodes and defining a distance or similarity metric between pairs of data points to create edges.

Our own daily cognitive processes can be viewed as movement through a vast semantic network, with nodes as concepts or ideas and edges as mental associations between them. To see this directly, visualize a concept or object. Perhaps the food you had for breakfast this morning, or the city you live in. Now, think about the connections between those concepts and others. Maybe you had eggs and toast for breakfast. Eggs and toast are connected with a multitude of other concepts in your mind: forks and silverware, kitchens, hunger, protein, carbohydrates, your morning routine, chickens, wheat, other breakfast foods, and so forth. Exploring these mental relationships is equivalent to walking along a small subset of your personal cognitive network.

For business-minded readers, this is an opportune time to learn about networks. Language models have become ubiquitous in the startup environment, but they hallucinate and need to be well-grounded. Using networks as their backbone is an excellent way to accomplish this goal. The application section of the book provides many tools useful for tackling real-world network problems. Some examples include

https://doi.org/10.1017/9781009405379.003 Published online by Cambridge University Press

analyzing social networks, optimizing supply chains, detecting fraud, developing recommendation systems, and using graph neural networks to develop new medical treatments.

The development of network machine learning follows a typical pattern in scientific advancement:

1. **Academic phase:** Researchers spend 10–20 years publishing proof-of-concept papers, exploring problem-solving approaches, and developing fundamental tools. This work is often done informally, with code existing primarily in Jupyter notebooks.
2. **Industry adoption:** As the field matures, companies begin to recognize the potential of these academic tools. They adapt them to enhance their products or services and develop user-friendly packages (like `networkx` or `graspologic`) to make the tools accessible to a broader audience.

Network machine learning is currently at a critical juncture. Its academic foundations have been established over the past two decades, and we are now seeing the transition of its tools from academia to industry. This presents a unique opportunity for early adoption of application-focused network machine learning tools in business contexts.

1.1.3.1 Why Do We Need Special Machine Learning Approaches for Networks?

In the wild, networks primarily exist as nodes and edges rather than in a tabular format. This presents a challenge, since techniques that were developed over decades for tabular data cannot be directly applied for network-valued data. However, we can adapt networks to traditional tabular formats using network representation tools. Once we transform these networks into more conventional structures, we can apply techniques from other domains of machine learning to analyze our networks.

Figure 1.1.4 illustrates the high-level process of network machine learning. By *network machine learning*, we refer to machine learning techniques applied to network-valued data (data which are a network, not a tabular structure). This process typically involves the following steps:

1. Obtain a network dataset,
2. Select a suitable representation for the network, based on the questions we wish to address,

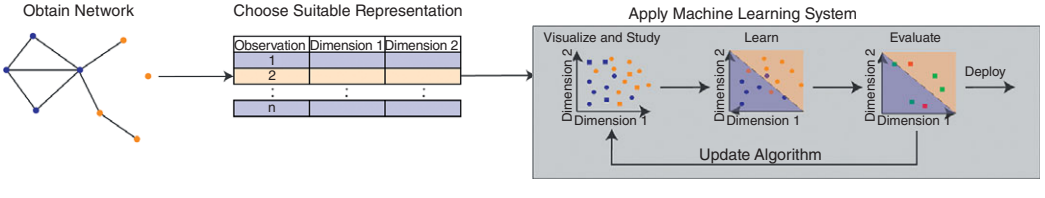


Figure 1.1.4 The network machine learning pipeline: We start with a network, choose a suitable representation, apply appropriate machine learning techniques, and interpret the results in the context of the original network.

https://doi.org/10.1017/9781009405379.003 Published online by Cambridge University Press

3. Apply appropriate machine learning techniques to our representation, and
4. Interpret the results in the context of the original network structure.

We will now explore the types of problems we might encounter in network machine learning. Then, in Chapter 2, we will walk through a simple end-to-end network analysis project.

1.2 Types of Network Machine Learning Problems

There are many different types of network machine learning systems. We broadly group them into the following categories:

- whether they involve one or multiple networks (single or multiple network learning systems),
- whether they require additional information in the form of network attributes (attributed or nonattributed network learning systems),
- whether they ask questions about an edge, a node, a group of edges/nodes, or about the network itself, and
- whether the approach can be used in isolation from a statistical model (nonmodel-based or model-based network learning systems).

To illustrate these criteria, we will use two running examples:

Example 1.2.1 Brain networks for musicians and nonmusicians
Brain networks have been acquired from a large group of people. The nodes represent areas of the brain, and the edges represent whether a pair of areas can communicate using neurons. Each network is either from a musician or a nonmusician.

Example 1.2.2 A pair of social networks for students at a school
We will also use social networks for students at two schools. Nodes represent students, and edges represent whether the students are connected on social media. Networks exist from two social media sites: Facebook and Instagram.

These categories are not mutually exclusive, and a network machine learning system will pull elements from several categories simultaneously. For instance, a system might be single network, node-attributed, and nonmodel-based like a community detection algorithm.

1.2.1 Single versus Multiple Network Learning Systems

1.2.1.1 Single Network Learning Systems

In many network learning scenarios, there is only a single sample: the network itself. A *single network learning system* derives insight from a single network, which is a

https://doi.org/10.1017/9781009405379.003 Published online by Cambridge University Press

single collection of nodes and edges. This differs from traditional machine learning frameworks, in which having a single sample would be disastrous: In that case, insight would be impossible, because we need multiple observations to identify trends or patterns.

However, in network learning, a single sample is not necessarily limiting. While there may be only one network, it is defined by a collection of many nodes and edges. Thus, it is still possible to learn about relationships that exist among the nodes, edges, or both. The caveat is that the conclusions drawn are limited by the specific network under study, which may not be representative of the global population of networks. This limitation is often not significant, depending on what the network represents. Most of the tools in this book are for single network learning systems.

1.2.1.2 Multiple Network Learning Systems

A *multiple network learning system* derives insight from multiple networks, each consisting of collections of nodes and edges. Unlike a single network learning system where conclusions can only be drawn based on characteristics of that particular network, a multiple network learning system can generate insights both within and across the networks.

The following are examples of multiple network learning systems:

- multiple network representation learning in Section 5.5,
- anomaly detection in Section 8.1,
- signal subnetworks in Section 8.2, and
- graph neural networks in Section 9.1.

1.2.2 Nonattributed versus Richly Attributed Network Learning Systems

While machine learning typically distinguishes between unsupervised and supervised learning, network learning employs more specific terminology for these concepts.

1.2.2.1 Nonattributed Network Learning Systems

The concept of a nonattributed network learning system is analogous to fully unsupervised machine learning. *Unsupervised learning* can be defined as a learning problem where the data provided to the algorithm do not include the desired solutions, known as *labels*. A network learning system is considered *nonattributed* if the data given to the system include only the nodes and edges at the time of the analysis.

For instance, consider the Facebook network from Example 1.2.2. Suppose that the information about which students attend which school has been lost. We hypothesize that there might be two groups of students in the network, called communities, and that if a student is in a particular community, they tend to be better friends with other students in the same community. We want to see if we can identify these communities programmatically, and perhaps recover the school information for each student. Figure 1.2.1 illustrates a nonattributed network learning problem. The groups of nodes that are heavily connected (indicated by the gray circles) correspond to the schools that each student attends.

Examples of nonattributed network learning systems include:

https://doi.org/10.1017/9781009405379.003 Published online by Cambridge University Press

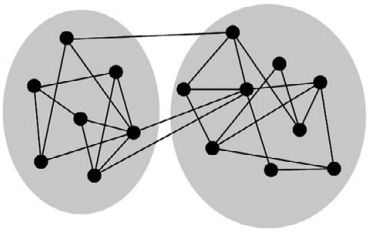


Figure 1.2.1 A school network. Nodes represent students, and edges indicate friendships. The gray circles indicate guesses as to which school each student belongs to.

- network embeddings in Section 5,
- community detection in Section 6.1,
- latent position comparisons in Section 7.1, and
- anomaly detection in Section 8.1.

1.2.2.2 Attributed Network Learning Systems

Similarly, the concept of an attributed network learning system is analogous to supervised or semisupervised machine learning. *Supervised learning* can be loosely defined as a learning problem where data provided to the algorithm include labels, and *semisupervised learning* can be loosely defined as a learning problem in which the data provided to the algorithm include some of the labels. A network is an *attributed network learning system* if, at the time of analysis, the network(s) include attributes in addition to nodes and edges. These attributes do not necessarily have to be scalar-valued labels: they can be vectors as well. There are four main types of attributed network learning systems. They are:

- Networks with node attributes,
- Networks with edge attributes,
- Networks with network attributes, and
- Networks with multiple-network attributes.

Networks with Node Attributes

When networks have *node attributes*, each node has an additional piece of information describing it. Returning to the school example, consider a scenario where, for each student, there is an additional piece of information: their school. The goal is to investigate whether the probability of two students being friends is higher in school one or in school two. A problem for networks with node attributes is shown in Figure 1.2.2(A).

Examples of problems that deal with node attributes include:

- joint representation learning in Section 5.6,
- model selection in Section 6.4,
- testing for differences in block matrices in Section 7.2, and
- testing for differences between groups of edges in Section 6.3.

https://doi.org/10.1017/9781009405379.003 Published online by Cambridge University Press

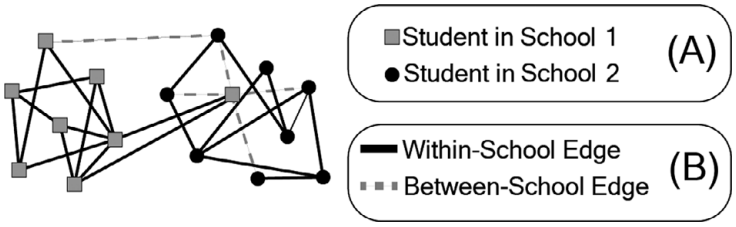


Figure 1.2.2 **(A)** The school network for Facebook, analyzed using node attributes. **(B)** The school network for Facebook, analyzed using edge attributes.

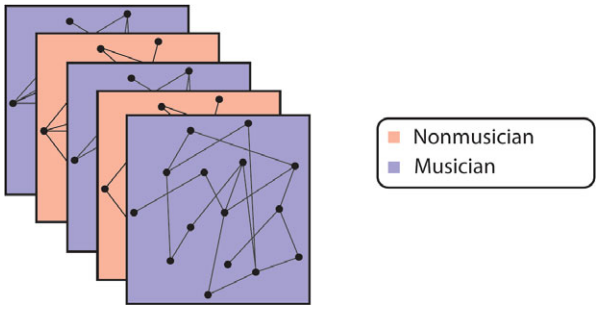


Figure 1.2.3 A collection of brain networks, where the nodes represent areas of the brain and the edges indicate which brain areas can communicate. The color of the box around each network indicates whether it comes from a musician or a nonmusician.

Networks with Edge Attributes

When networks have *edge attributes*, each edge has an additional piece of information describing it. For example, in the school network, we could entirely ignore the school assignments, and simply focus on whether the students have more friends within schools (solid edges) or between schools (dashed edges). This example is illustrated in Figure 1.2.2(B). Rather than focusing on the school assignments of the nodes, we focus on two groups of edges (the within-school edges, and the between-school edges).

An example of a problem with edge attributes is testing for differences between groups of edges in Section 6.3.

Networks with Network Attributes

When networks have *network attributes*, each network has an additional piece of information describing it. Referring back to Example 1.2.1, each brain network was from either a musician or a nonmusician. This piece of information characterizes each of the networks as either from a musician or a nonmusician individual, and applies to the entire collection of nodes and edges for a given network. Figure 1.2.3 illustrates a network with network attributes.

Examples of problems that leverage network attributes are signal subgraph estimation in Section 8.2, and graph neural networks in Section 9.1.

https://doi.org/10.1017/9781009405379.003 Published online by Cambridge University Press

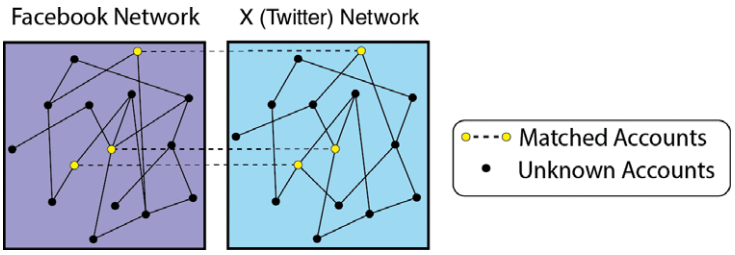


Figure 1.2.4 Two social networks from different social media sites. The nodes are people's accounts on the social media sites (they are the same for both sites) and the edges indicate which pairs of accounts follow one another for that particular site. For three people, we know a matching across the two networks (yellow nodes, dashed edges).

Networks with Multiple-Network Attributes

A network with *multiple-network attributes* consists of a collection of networks (each of which has nodes and edges) where we have additional information that describes how the nodes (or the edges) of the different networks relate to one another. Returning to Example 1.2.2, let's add another dimension. Imagine the accounts are, for all intents and purposes, anonymous, since identifying information is not shared by default on Facebook or Instagram accounts. However, we know that a particular Facebook account corresponds to the same person's Instagram account for three people. We want to see if we can use this cross-network attribute (the matched accounts) to discover suitable matchings for the remaining accounts in the network. A problem with multiple-network attributes is shown in Figure 1.2.4.

An example of a problem which leverages cross-network attributes is seeded graph matching, in Section 7.3.

1.2.3 Scope of Network Analysis

Network machine learning often tempts researchers to explore entire networks, even when questions are more focused. Consider a transportation network with stations as nodes and rush-hour ridership counts between stations as edges. A different network might exist for each week of the year. The following examples demonstrate questions about different network components.

1. Studying a single edge: Determining whether a particular route needs an additional train due to rush hour popularity might require examining this route across many networks to estimate passenger numbers.
2. Studying a single node: Deciding on station expansion could involve analyzing passenger throughput at a specific station.
3. Studying groups of nodes or edges: Evaluating the impact of canceling an entire line would require considering effects on other stations (groups of nodes) and passenger numbers on other lines (groups of edges).
4. Studying the entire network: Assessing the need for increased public transportation funding might entail analyzing the city's weekly cost across the entire transportation network.

https://doi.org/10.1017/9781009405379.003 Published online by Cambridge University Press

1.2.4 Model-Based versus Nonmodel-Based Network Learning Systems

Statistics forms a core component of network learning systems. This is because network learning problems inherently involve randomness, whether in the sampled networks, the set of acquired networks, the collected data, or other factors.

In statistics, we often sample observations from a population, and then use what we learn from the sample to extrapolate to the population. We often build statistical models in service of this.

Consider a statistical model as a conceptual framework that explicitly accounts for variation, randomness, or error in the networks we obtain compared to the entire population. The balance between model-based and nonmodel-based network learning is a core aim of this book, so we will explain the difference without reference to networks.

1.2.4.1 Model-Based Learning Systems

A *model-based learning system* requires a statistical model to derive meaning from an analysis.

Consider two coins, each flipped twenty times. To determine if the probability of heads is the same or different for these coins, we use a hypothesis test. We have two hypotheses: that we have the same (first hypothesis) or different (second hypothesis) probability of landing on heads. A *hypothesis test* determines whether or not the data provide evidence against the first hypothesis.

This question requires a statistical model because "probability" has a specific statistical interpretation. To answer this question, several factors about the experiment require clarification:

- Are heads and tails the only possible outcomes for each coin? For example, could a thicker coin have a nonzero probability of landing on its edge?
- Is the probability of heads consistent across all twenty flips for each coin? Could different flipping techniques in the first and last ten flips affect the frequency of heads?
- Are the outcomes of coin flips independent? For instance, would saying a prayer after two tails influence the probability of getting heads on the next flip?
- Is the sample size sufficient to detect differences between the coins? Many statistical tests, including this one, can be interpreted in various ways. With small sample sizes, assumptions must be carefully considered, as certain testing approaches (like the chi-squared test) may only be meaningful with larger datasets.

In order to make a conclusion based on our hypothesis test, we need to be specific about the assumptions we make about these details of the our data sample, since the answer to our question depends on these assumptions. Therefore, we need to understand the assumptions that we made, so that we can make a decision based on the outcome of the hypothesis test that we performed.

1.2.4.2 Nonmodel-Based Learning Systems

Many questions can be addressed using either model-based or nonmodel-based techniques. While models can provide intuition, they are not always necessary to answer

https://doi.org/10.1017/9781009405379.003 Published online by Cambridge University Press

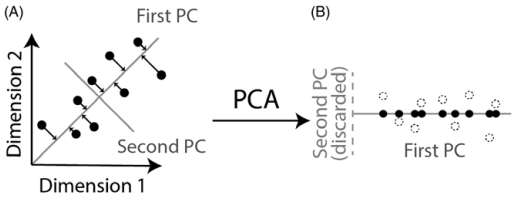


Figure 1.2.5 **(A)** Two-dimensional data shown in a scatter plot. The first principal component (PC) represents the axis of maximum variation, while the second PC is orthogonal to the first and captures the next highest variation. Arrows show data projection onto the first PC. **(B)** PCA projects data onto the first PC by reorienting the dataset along the first and second PCs (dashed circles and lines), then removing the second PC (solid circles). This reduces dimensionality from two to one, discarding the direction of less variation.

questions. A *nonmodel-based learning system* does not require a statistical model to derive meaning from an analysis.

To illustrate this concept, consider principal components analysis (`pca`), a nonmodel-based machine learning technique. `pca` reduces high-dimensional data to a more manageable number of dimensions, allowing for downstream analysis with other strategies. Figure 1.2.5 provides a visual illustration of `pca`, showing data reduction from two dimensions to one along the first principal component (PC).

`pca` can be executed without additional knowledge about the data, simply as an algorithm to make data more manageable for other machine learning techniques. However, understanding `pca` through the lens of a statistical model (assuming normally distributed observations) provides additional insight. In this context, principal components represent directions of maximum variation in the data.

In Figure 1.2.5(A), the black points spread more along the first PC than the second PC. The degree of variation preserved by each PC is indicated by its score. This interpretation is useful for machine learning tasks like K-means clustering, where distinguishable classes require variability between observations.

1.3 Challenges of Network Machine Learning

Like other branches of science, network machine learning is not without its challenges. The networks we analyze often contain random imperfections. Rather than collecting exhaustive data, we typically examine reasonably sized subsets. This may mean studying a social network with 1000 nodes instead of 100 million, or 200 brain networks rather than one from every person in the world.

Because our data are not perfect, we often use tools from *statistical learning*, a machine learning framework that uses statistics to refine problems, quantify the reasonableness of conclusions, and make rigorous inferences about networks. In this section, we will discuss several of the challenges that may arise as a result of random imperfections in network data. The remainder of the book will be dedicated

https://doi.org/10.1017/9781009405379.003 Published online by Cambridge University Press

to developing models and techniques that allow us to overcome some of these limitations.

1.3.1 We Might Imperfectly Observe the Network

In many branches of science, collecting network data can be an imperfect and noisy process. Only recently did we acquire a detailed map of the connectome of a complex organism, the fruit fly larva [3]. Even though fruit fly larvae are small, this was a substantial undertaking: The fruit fly brain is comprised of several thousand neurons (nodes), and hundreds of thousands of neuronal connections (edges), all of which fit into a volume smaller than a cubic millimeter. To study the brain at the neuron-scale, researchers sliced it into thousands of tiny pieces smaller than a human hair, and then examined each piece individually with a microscope. From there, algorithms and manual tracing were performed to reconstruct, slice-by-slice, which neurons were connected with which other neurons. To complicate matters further, all of this work was derived from a single organism. This means that each step allowed minimal room for error. The neuron-by-neuron map was used to construct a network (called the *connectome* of the organism) of connections between neurons.

With so many areas for potential error, it is almost impossible that the fruit fly connectome is absolutely perfect. We may have missed some neurons, overlooked connections, or made small mistakes anywhere in this complicated process. Figure 1.3.1 explores what it means for a network to be imperfectly observed. In Figure 1.3.1(A), we see the underlying network we are trying to obtain. In Figure 1.3.1(B), we see the actual network we obtained: we might be missing some of the nodes (gray dashed nodes), we might be missing some of the edges (gray dashed edges), or we might see edges which should not really be present (nonfaithful edges, gray solid edges). While a portion of the network might faithfully represent the underlying system (solid black nodes and edges), we do not actually know what part of our sample is faithful or unfaithful with respect to the true underlying network. The key is that when we build network machine learning systems, we desire insights that are robust to these imperfect observations. This is because in some cases, it might be impossible to ever obtain the data that we need otherwise.

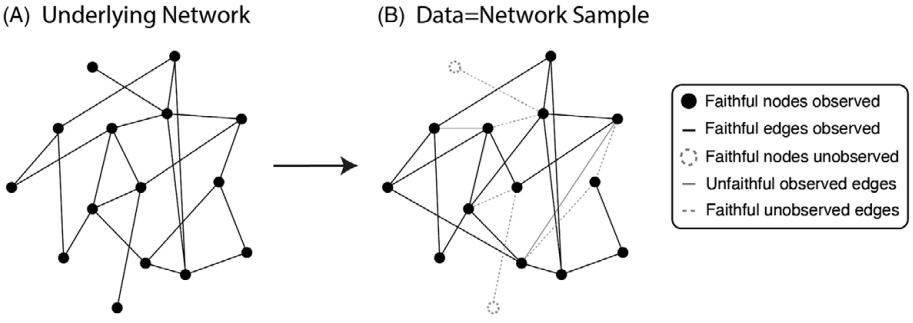


Figure 1.3.1 The true underlying network contains a lot of information that we did not measure properly in the course of observing or sampling the network.

https://doi.org/10.1017/9781009405379.003 Published online by Cambridge University Press

1.3.2 We Might Not See the Whole Network

When studying a network, we rarely observe the entire system perfectly. For instance, consider a social network, in which the nodes are people within the network, and edges represent friendships. If we wanted to study the network in its rawest form, we might need to collect data from millions, or billions, of accounts to construct a network that might take an infeasible amount of space just to store. Actually analyzing the network represents another huge hurdle; we would need to be able to devise techniques which could efficiently churn through terabytes worth of data. However, focusing our attention on a subset of the network involving a few thousand or hundred thousand people may allow us to ask richer questions. On this reduced subset of people, we are less limited by computational constraints and so we can apply a wider range of analytical techniques.

The fruit fly larva (*Drosophila*) connectome example illustrates how researchers have approaches studying large networks over time. While the complete connectome was mapped only recently, brain networks have been studied for decades. Previously, owing to economic, computational, and analytical challenges, investigations focused on subsets of the brain rather than the whole. Despite only collecting bits and pieces of the network, major insights were learned which directly informed the effort to collect the entire network.

In both of these cases, we can learn a lot of valuable information by reducing the size of a network, learning from it, and then applying what we learned to the entire network. Figure 1.3.2 illustrates sampling a subset of nodes from a network. Figure 1.3.2(A) depicts shows the true underlying network with both solid and dashed edges and nodes. Figure 1.3.2(B) depicts the sampled network, which only includes a subset of the nodes and edges.

1.3.3 We Might Only See a Subset of the Networks

Consider again Example 1.2.1 of musician and nonmusician brain networks. A team of psychologists might hypothesize that the brains of musicians tend to be better connected in areas responsible for fine motor coordination and hearing, which are

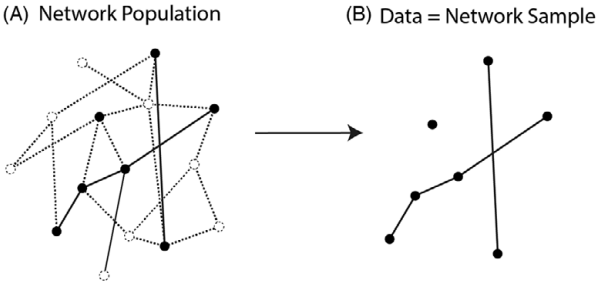


Figure 1.3.2 The true underlying network has both solid and dashed edges and nodes, shown in **(A)**. However, when we sample the network in **(B)**, our sample only includes the subset of nodes and edges that are solid.

https://doi.org/10.1017/9781009405379.003 Published online by Cambridge University Press

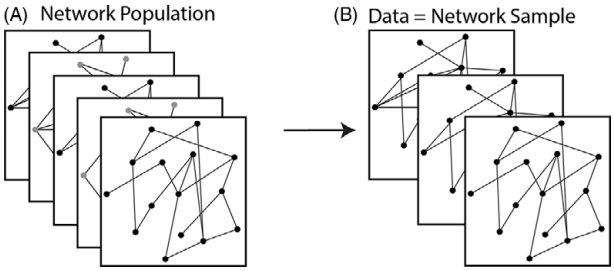


Figure 1.3.3 The true underlying population contains many more networks than we are able to actually sample. In **(A)**, we have black and gray networks, but in the sample in **(B)**, we only get to see the gray networks.

crucial skills for many instruments. To test this hypothesis, the psychologists could take one of two approaches. They could collect brain networks from every individual, or they could collect a subset of brain networks from groups of musicians and nonmusicians in their area. Figure 1.3.3 demonstrates sampling a subset of networks from a population. Despite the fact that the psychologists only studied a subset of musicians and nonmusicians, with some statistical assumptions they can derive conclusions that will apply more broadly than just to the group of people they analyzed.

1.3.4 Statistics Allows Us to Generalize to the Unseen from the Seen

If we collect a set of brain networks, we have collected a *sample*, a subset of objects from the larger population. Statistical analysis of the sample allows us to draw broader conclusions. Rather than limiting our findings to the specific group or network we sample, statistics enables us to rigorously extend conclusions to a more general population. Figure 1.3.4 summarizes this concept. Note that "network population" may have different interpretations depending on the specific question. It might refer to a population of many networks, from which we observe a sample with some level of uncertainty, or to a "true" single network, which we can only observe with some level of randomness because of how the nodes or edges were obtained.

We can learn many things about our network data without leveraging statistics at all. However, using statistical learning gives us a quantitative framework to generalize our findings. Conclusions can then be applied more broadly to the general population rather than being limited to our specific network data.

1.3.5 Other Challenges

Section 1.2 introduced several types of network machine learning problems. The field is rapidly evolving, with new problem formulations continually emerging. Researchers face numerous challenges, including:

1. Categorizing problems appropriately,
2. Reformulating research questions to align with existing techniques,

https://doi.org/10.1017/9781009405379.003 Published online by Cambridge University Press

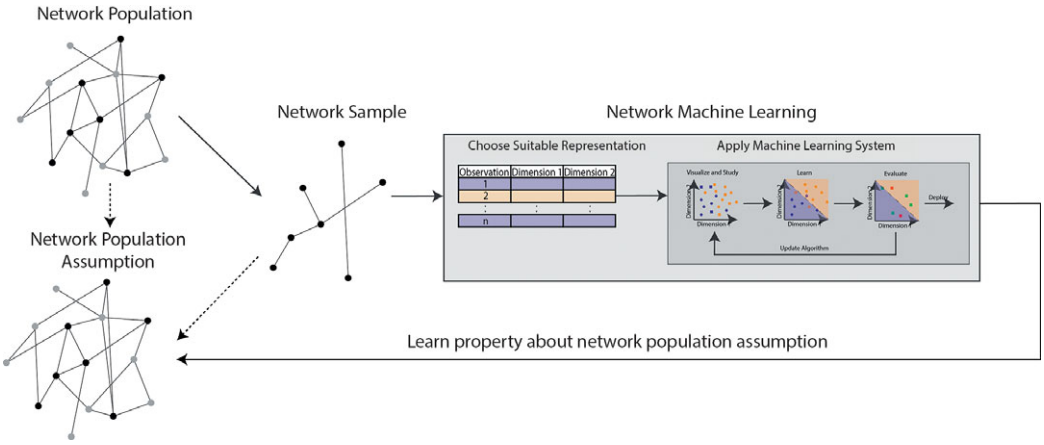


Figure 1.3.4 We have a network population, which in this case means a large network that we cannot properly observe. We obtain a sample of this network, and then use analyses on this sample and knowledge of how it was taken from the population to derive assumptions about our system. Ideally, this will reasonably represent the network population itself. Using network machine learning, we try to understand the properties of this network population.

3. Selecting suitable analytical strategies, and
4. Developing entirely new methods when necessary.

These challenges often require significant effort and resources to address. Chapter 2 presents an example data analysis in network machine learning, illustrating how researchers approach these problems in practice.

Bibliography

[1] Contributors to Wikimedia projects. Machine learning – Wikipedia; 2023. [Online; accessed Jan. 11, 2023.]

[2] What kind of data does pandas handle?; 2022. https://pandas.pydata.org/docs/getting_started/intro_tutorials/01_table_oriented.html [Online; accessed Jan. 11, 2023.]

[3] Winding M, Pedigo BD, Barnes CL, Patsolic HG, Park Y, Kazimiers T, et al. The connectome of an insect brain. Science. 2023 Mar.;379(6636):eadd9330.

https://doi.org/10.1017/9781009405379.003 Published online by Cambridge University Press