Chapter 2
Preliminaries

This chapter contains technical background for the material addressed throughout the rest of the book. We begin in Section 2.1 with an overview of necessary topics from graph theory, which provides us with much of the language and infrastructure for manipulating and describing networks and network data. We then turn in Section 2.2 to a brief review of fundamental elements from probability and statistical inference, which will provide us with most of the language and principles used here for the modeling and analysis of network data. Finally, in Section 2.3, we discuss, through a series of examples, some of the unique challenges inherent in the statistical analysis of network data. Readers sufficiently familiar with both graph theory and statistical inference may wish to skip this chapter and move directly to Chapter 3, after perhaps a quick detour to glance through the examples of Section 2.3 and the issues raised therein.

2.1 Background on Graphs

We have seen that the term 'network,' as commonly used, refers simply to a collection of elements and their inter-relations. The sub-field of mathematics known as graph theory lends precision to this notion. In particular, it provides a body of definitions, tools, techniques, and results for discussing graphs and their properties. These will play an especially important role in the material of Chapters 3 and 4. We review here some of the basic terminology and concepts on graphs, discuss common families of graphs, introduce certain important connections between graphs and matrix algebra, and then briefly visit the topic of graph data structures and algorithms.

E.D. Kolaczyk, *Statistical Analysis of Network Data*, Springer Series in Statistics,
DOI 10.1007/978-0-387-88146-1_2, © Springer Science+Business Media, LLC 2009

2.1.1 Basic Definitions and Concepts

Formally, a *graph* $G = (V, E)$ is a mathematical structure consisting of a set V of *vertices* (also commonly called *nodes*) and a set E of *edges* (also commonly called *links*), where elements of E are unordered pairs $\{u, v\}$ of distinct vertices $u, v \in V$. The number of vertices $N_v = |V|$ and the number of edges $N_e = |E|$ are sometimes called the *order* and *size* of the graph G, respectively. Often, and without loss of generality,[1] we will label the vertices simply with the integers $1, \ldots, N_v$, and the edges, analogously. A graph $H = (V_H, E_H)$ is a *subgraph* of another graph $G = (V_G, E_G)$ if $V_H \subseteq V_G$ and $E_H \subseteq E_G$. An *induced subgraph* of G is a subgraph $G' = (V', E')$, where $V' \subseteq V$ is a prespecified subset of vertices and $E' \subseteq E$ is the collection of edges to be found in G among that subset of vertices.

As defined, a graph has no edges for which both ends connect to a single vertex (called *loops*) and no pairs of vertices with more than one edge between them (called *multi-edges*). A graph with either of these properties is called a *multi-graph*. For simplicity, and reflecting the bulk of common practice, the presentation in this book will concentrate primarily on graphs, and not multi-graphs, though reference to the latter will be made where appropriate. When it is necessary to indicate explicitly that a graph G is not a multi-graph, we will refer to it as a *simple* graph, and its edges, as *proper* edges.

A graph G for which each edge in E has an ordering to its vertices (i.e., so that $\{u, v\}$ is distinct from $\{v, u\}$, for $u, v \in V$) is called a *directed graph* or *digraph.* Such edges are called *directed edges* or *arcs*, with the direction of an arc $\{u, v\}$ read from left to right, from the *tail* u to the *head* v. Note that there is a natural extension of digraphs to *multi-digraphs,* where multiple arcs (i.e., *multi-arcs*) share the same head and tail. Note too, however, that digraphs may have two arcs between a pair of vertices without their being multi-arcs if the vertices play opposite roles of head and tail for the respective arcs. In this case, the two arcs are said to be *mutual*.

It is necessary to have a language for discussing the connectivity of a graph. One of the most basic notions of connectivity is that of adjacency. Two vertices $u, v \in V$ are said to be *adjacent* if joined by an edge in E. Similarly, two edges $e_1, e_2 \in E$ are adjacent if joined by a common endpoint in V. A vertex $v \in V$ is *incident* on an edge $e \in E$ if v is an endpoint of e. From this follows the notion of the *degree* of a vertex v, say d_v, defined as the number of edges incident on v. The *degree sequence* of a graph G is the sequence formed by arranging the vertex degrees d_v in non-decreasing order. The sum of the elements of the degree sequence is equal to twice the number of edges in the graph (i.e., twice the size of the graph). Note that for digraphs, vertex degree is replaced by *in-degree* (i.e., d_v^{in}) and *out-degree* (i.e., d_v^{out}), which count the number of edges pointing in towards and out from a vertex, respectively. Hence, digraphs have both an in-degree sequence and an out-degree sequence.

[1] Technically, a graph G is unique only up to relabellings of its vertices and edges that leave the structure unchanged. Two graphs that are equivalent in this sense are called *isomorphic*.

It is also useful to be able to discuss the concept of movement about a graph. For example, a *walk* on a graph G, from v_0 to v_l, is an alternating sequence $\{v_0, e_1, v_1, e_2, \ldots, v_{l-1}, e_l, v_l\}$, where the endpoints of e_i are $\{v_{i-1}, v_i\}$. The *length* of this walk is said to be l. Refinements of a walk include *trails*, which are walks without repeated edges, and *paths*, which are trails without repeated vertices. A trail for which the beginning and ending vertices are the same is called a *circuit*. Similarly, a walk of length at least three, for which the beginning and ending vertices are the same, but for which all other vertices are distinct from each other, is called a *cycle*. Graphs containing no cycles are called *acyclic*. In a digraph, these notions generalize naturally. For example, a *directed walk* from v_0 to v_l proceeds from tail to head along arcs between v_0 and v_l.

A vertex v in a graph G is said to be *reachable* from another vertex u if there exists a walk from u to v. The graph G is said to be *connected* if every vertex is reachable from every other. A *component* of a graph is a maximally connected subgraph. That is, it is a connected subgraph of G for which the addition of any other remaining vertex in V would ruin the property of connectivity. For a digraph, there are two variations of the concept of connectedness. A digraph G is *weakly connected* if its underlying graph (i.e., the result of stripping away the labels 'tail' and 'head' from G) is connected. It is called *strongly connected* if every vertex v is reachable from every u by a directed walk.

A common notion of *distance* between vertices on a graph is defined as the length of the shortest path(s) between the vertices (which we set equal to infinity if no such path exists). This distance is often referred to as *geodesic distance*, with 'geodesic' being another name for shortest paths. The value of the longest distance in a graph is called the *diameter* of the graph.

Finally, it is not uncommon to equip (or 'decorate') a graph G with auxiliary numerical values on its vertices, edges, or both. For example, edges $e \in E$ are often accompanied by *edge weights*. In fact, extending the notion of edge weights to all pairs of vertices, the edge set E itself can be represented through a set $\{w_e\}$ of such weights, i.e., $w_e = 1$ if $e \in E$ and 0 if $e \notin E$. When edges are weighted, the corresponding length of a walk (trail, path, etc.) is measured as the sum of the values of the weights along the edges traversed in the walk. The notion of distance generalizes accordingly. These concepts extend naturally to digraphs.

Similarly, graph labellings may be used in representing a multi-graph as a decorated graph. Specifically, given a multi-graph, we can define a graph G having the same vertex set V and having an edge set E such that distinct elements $u, v \in V$ have an edge between them if there is at least one multi-edge between them in the multi-graph. Then, equip each vertex $v \in V$ with a label denoting the number of loops possessed by v in the multi-graph, and similarly, equip each edge with the number of multi-edges it represents.

Of course, in this book a particularly common source of labels for graph vertices and edges will be in the form of measurements of functions or processes on a given graph. However, we postpone introducing the necessary notation for such quantities until Section 2 of this chapter.

2.1.2 Families of Graphs

Graphs come in all ‘shapes and sizes,’ as it were, but there are a number of families of graphs that are commonly encountered in practice. Such prevalence typically is due to some combination of relevance and tractability, with the latter being of an analytical or computational nature, or both. We describe here a handful of the most common examples of graph families.

A *complete* graph is a graph where every vertex is joined to every other vertex by an edge. Figure 2.1 shows, on the left, a representation of a complete graph. This concept is perhaps most useful in practice through its role in defining a *clique*, which is a complete subgraph. A subgraph H of a graph G is said to be a maximal clique if it is complete and no other such subgraph contains it. Cliques are an extreme form of a ‘highly inter-connected’ subgraph, the existence and detection of which is often of interest in the analysis of a network. We will encounter additional, more flexible forms of this idea in Chapter 4.

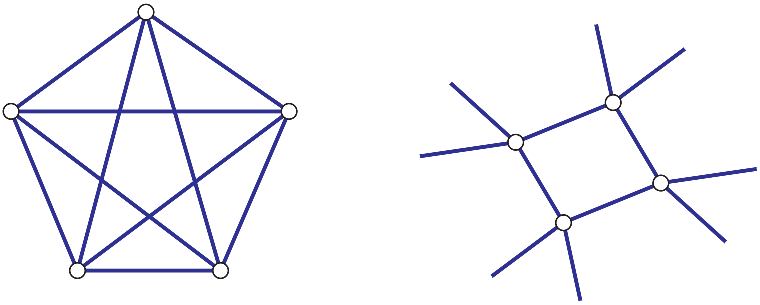

Fig. 2.1 Left: a complete graph. Right: a portion of a 4-regular graph.

A *regular* graph is a graph in which every vertex has the same degree. A regular graph with common degree d is called d*-regular.* Figure 2.1 shows, on the right, a portion of a 4-regular graph, such as is found in the common lattice defining, say, the squares on a chess board.[2] Regular graphs arise commonly in the study of various quantities in physics and chemistry (e.g., crystal structure), and arise frequently in geo-spatial settings (e.g., as a model of pixel adjacencies in image processing).

A connected graph with no cycles is called a *tree*. The disjoint union of such graphs is called a *forest*. Trees are of fundamental importance in the analysis of networks. They serve, for example, as a key data structure in the efficient design of many computational algorithms. A digraph whose underlying graph is a tree is called a *directed tree*. Often such trees have associated with them a special vertex

[2] Technically, to render this example truly 4-regular, the board must be made cyclic, in the manner of a circle, by joining the top and bottom edges (thus creating a tube) and then the open ends (thus creating a so-called toroidal lattice).

called a *root*, which is distinguished by being the only vertex from which there is a directed path to every other vertex in the graph. Such a graph is called a *rooted tree.* An example is shown in Figure 2.2. A vertex preceding another vertex on a path from the root is called an *ancestor*, while a vertex following another vertex is called a *descendant*. Immediate ancestors are called *parents*, and immediate descendants, *children*. A vertex without any children is called a *leaf*. Given a rooted tree of this sort, it is not uncommon to represent it diagrammatically without any indication of its directedness, as this is to be understood from the definition of the root.

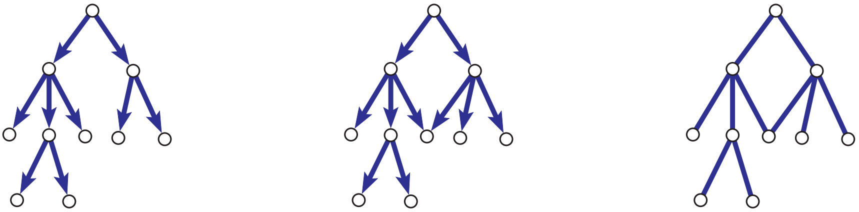

Fig. 2.2 Left: a rooted tree. Center: a DAG. Right: the undirected graph underlying the DAG.

An important generalization of the concept of a tree is the *directed acyclic graph* (i.e., the DAG). A DAG, as its name implies, is directed and has no cycles. However, unlike a directed tree, its underlying graph is not a tree, in that replacing the arcs with undirected edges leaves a (simple) graph that contains cycles. Nevertheless, it is often possible to still design efficient computational algorithms on DAGs that take advantage of this near-tree-like structure. An example of a DAG is shown in Figure 2.2, along with its underlying graph.

A *bipartite* graph is a graph $G = (V,E)$ such that the vertex set V may be partitioned into two disjoint sets, say V_1 and V_2, and each edge in E has one endpoint in V_1 and the other in V_2. Such graphs are typically used to represent ‘membership’ networks, for example, with ‘members’ (e.g., people) denoted by vertices in V_1, and the corresponding ‘organizations’ (e.g., clubs), by vertices in V_2. It is not uncommon to accompany a bipartite graph with at least one of two possible induced graphs. Specifically, a graph $G_1 = (V_1,E_1)$ may be defined on the vertex set V_1 by assigning an edge to any pair of vertices that both have edges in E to at least one common vertex in V_2. Similarly, a graph G_2 may be defined on V_2. Figure 2.3 shows an example of a bipartite graph G and its induced graph G_1.

As a last example, we mention the class of planar graphs. A graph G is said to be *planar* if, informally speaking, it may be drawn in the plane, with vertices as dots and edges as lines, in such a way that no pair of edges intersect anywhere other than at vertices to which they are jointly incident. Planar graphs are often an appropriate representation of networks with a spatial component, such as many technological networks, and have a number of special properties induced by the structural requirement that they ‘lie in’ the plane.

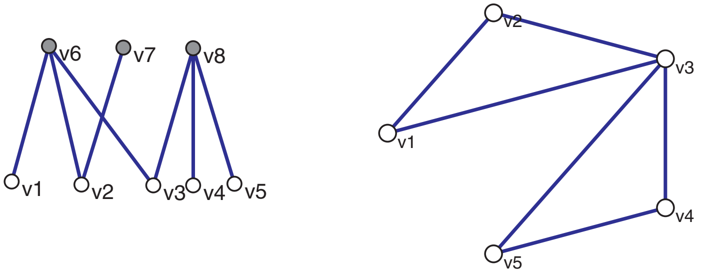

Fig. 2.3 Left: a bipartite graph. Right: a graph induced by the bipartite graph on the ‘white’ vertex set.

2.1.3 Graphs and Matrix Algebra

We shall see that it is frequently useful in the modeling and analysis of network data to be able to characterize a graph G and certain aspects of its structure using matrices and matrix algebra. Our ability to do so in a rigorous manner derives from a formal blending of graph theory with matrix algebra, in a field called algebraic graph theory, the roots of which go back to Kirchoff and his study of electrical networks. We briefly describe a handful of the elements from this area that will be of particular use to us in various parts of this book.

The fundamental connectivity of a graph G may be captured in an $N_v \times N_v$ binary, symmetric matrix $\mathbf{A}$ with entries

$$A_{ij} = \begin{cases} 1, & \text{if } \{i,j\} \in E \ , \\ 0, & \text{otherwise} \ , \end{cases} \tag{2.1}$$

where we use the integers $1,\ldots,N_v$ generically to denote the elements of V and we represent an edge $e \in E$ explicitly as an unordered pair of vertices $i, j \in V$. In words, $\mathbf{A}$ is non-zero for entries whose row-column indices correspond to vertices in G joined by an edge, and zero, for those that are not.

The matrix $\mathbf{A}$, called the *adjacency matrix*, is useful not only for storing connectivity information, but also in that certain operations on $\mathbf{A}$ yield additional information concerning G. For example, the row sum $A_{i+} = \sum_j A_{ij}$ is simply equal to the degree d_i of vertex i. Note that, by symmetry, $A_{i+} = A_{+i}$. Furthermore, if we let $\mathbf{A}^r$ denote the r-th power of $\mathbf{A}$, then the entry A^r_{ij} yields the number of walks of length r between i and j on G. Finally, there are many interesting and useful relations involving the eigenvalues of G. For example, it can be shown that G is a regular graph if and only if the maximum degree d_{max} of G is an eigenvalue of $\mathbf{A}$.

An adjacency matrix may also be defined for digraphs, by adjusting the definition in (2.1) so that $A_{ij} = 1$ if $\{i,j\} \in E$ represents a directed edge from i to j. Of course, $\mathbf{A}$ is now no longer symmetric. However, it still contains similarly useful additional information. For example, $A_{i+} = d^{out}_i$ and $A_{+j} = d^{in}_j$.

Another useful matrix capturing the fundamental structure in G is the *incidence matrix* $\mathbf{B}$, an $N_v \times N_e$ binary matrix with entries

$$B_{ij} = \begin{cases} 1, & \text{if vertex } i \text{ is incident to edge } j \ , \\ 0, & \text{otherwise} \ . \end{cases} \tag{2.2}$$

Given their definitions, it is natural to expect that there be a relation between $\mathbf{A}$ and $\mathbf{B}$. This is indeed true, if we extend $\mathbf{B}$ to a signed incidence matrix, say $\tilde{\mathbf{B}}$, where the entries '1' in (2.2) are given plus or minus signs indicating an arbitrarily assigned 'orientation' of the corresponding edge.[3] It then can be shown that $\tilde{\mathbf{B}}\tilde{\mathbf{B}}^T = \mathbf{D} - \mathbf{A}$, where $\mathbf{D} = \mathsf{diag}[(d_i)_{i \in V}]$ is a diagonal matrix containing the degree sequence.

The $N_v \times N_v$ matrix $\mathbf{L} = \mathbf{D} - \mathbf{A}$ is in fact quite important in its own right and is called the *Laplacian* of the graph G. Motivation for the term Laplacian, in analogy to the Laplacian from multivariable calculus (i.e., the sum of second partial derivatives of a function), may be found in the fact that, for a real-valued vector $\mathbf{x} \in \mathbb{R}^{N_v}$, we have

$$\mathbf{x}^T \mathbf{L} \mathbf{x} = \sum_{\{i,j\} \in E} (x_i - x_j)^2 \ . \tag{2.3}$$

The closer this value is to zero, the more similar are the elements of $\mathbf{x}$ at adjacent vertices in V. Hence, the Laplacian is useful in providing, through (2.3), some sense of the 'smoothness' of functions on a graph G, with respect to the connectivity of G. We will see this fact exploited in Chapter 8, for the purpose of regression and classification on network graphs.

The properties of $\mathbf{L}$, particularly the properties of its eigenvalues and eigenvectors, have much to say about the structure of G. Since $\mathbf{L}$ can be shown to be a positive semi-definite matrix, the eigenvalues are all non-negative. And because $\mathbf{L}\mathbf{1} = \mathbf{0}$, where $\mathbf{1}$ and $\mathbf{0}$ are $n_v \times 1$ vectors of ones and zeroes, respectively, the smallest eigenvalue λ_1 of $\mathbf{L}$ is equal to zero. The second smallest eigenvalue, λ_2, is typically nontrivial, and arguably the most important of all of the eigenvalues. For example, roughly speaking, the larger λ_2 is, the more 'connected' G is, and the more difficult it is to separate G into disconnected subgraphs by selectively eliminating some small number of edges. We will see in Chapter 4 that results of this kind are useful, for instance, in designing tools for clustering the vertices of the graph into meaningful subgraphs.

2.1.4 Graph Data Structures and Algorithms

Facilitating the transition from graphs as purely mathematical objects to graphs as practical tools for use in the analysis of network data are data structures and algorithms for graphs. The study of graph data structures and algorithms is basic to the field of computer science, where much effort has gone into the development of

[3] More formally, we arbitrarily assign each vertex in an edge j the role of either head or tail, as in a digraph, and let $\tilde{B}_{ij} = 1$ if vertex i is incident to edge j as a tail, and -1, if as a head.

efficient methods for the storage and manipulation of a graph, as well as methods for computing various characteristics of and answering different questions about graphs. We briefly highlight some of the most relevant elements of this literature now. Their role in what follows will be most evident in Chapters 3 and 4.

2.1.4.1 Data Structures

There are two common data structures for representing a graph G. The first is the $N_v \times N_v$ adjacency matrix $\mathbf{A}$ defined previously. This choice is often a natural one, given that matrices are fundamental data objects in most programming and software environments. However, if the graph is particularly large, and especially if the graph is sparse (e.g., if $N_e \sim N_v$), then it can be preferable to use a collection of *adjacency lists*. This is because the adjacency matrix in such cases will be both large and filled primarily with zeros, due to the fact that it explicitly represents both present and absent edges, while adjacency lists, on the other hand, store only the information on edges that are present. Specifically, an adjacency-list representation of a graph G is simply an array of size N_v, ordered with respect to the ordering of the vertices in V, each element of which is a list, where the i-th list contains the set of all vertices j for which there is an edge from i to j. A variation on this idea is an *edge list*, a simple two-column list of all vertex pairs that are joined by an edge.

The sum of the lengths of the adjacency lists will be N_e for a directed graph and $2N_e$ for an undirected graph. Therefore, the total amount of computer memory space required for storing a graph in an adjacency list representation is[4] only $O(N_v + N_e)$. When G is sparse, this memory requirement will be much less than the value $O(N_v^2)$ associated with storage of G through an adjacency matrix. On the other hand, when G is dense (i.e., $N_e \sim N_v^2$), the memory requirements for the two methods will be comparable. In addition, the simplicity of the adjacency matrix representation may sometimes be felt to outweigh any memory disadvantages, especially for smaller graphs.[5]

Note that for either adjacency matrix or adjacency list representations, it is easy to store annotated graphs in a similar fashion. For example, an edge weight w_e, for $e = \{i, j\} \in E$, may be stored in the (i, j)-th location of the adjacency matrix, or in addition to the value j (or i) in the list for i (or j), in an adjacency-list representation.

[4] For two sequences $(h_n)_{n\geq 0}$ and $(g_n)_{n\geq 0}$, we say $h_n = O(g_n)$, read as 'h_n is big-oh of g_n', if there exists a constant c such that eventually $|h_n/g_n| \leq c$, for large n.

[5] Ultimately, a comparison of the two methods of data representation for graphs can be more subtle than we have described. For example, sparse matrix storage methods allow one to effectively store the adjacency matrix in significantly less than $O(N_v^2)$ space. In addition, there can be important interactions between data representation and algorithm design that impact computational efficiency.

2.1.4.2 Algorithms

There are numerous questions we might wish to ask in regards to a given graph. Some are quite simple and their answers may be obtained in a straightforward manner directly from the data structures just discussed. For example, are vertices i and j linked by an edge? What is the degree of vertex i? For many other questions, however, more work may be required, but it is usually still feasible to obtain an answer in a reasonably efficient manner. For example, what is the shortest path(s) between vertices i and j? How many connected components does the graph have? And, for a directed graph, does it have cycles or is it acyclic? Finally, for certain questions, there is likely no efficient algorithm. An example of such a question is the one that asks for a maximal clique in a given graph.

Thus algorithmic complexity is frequently an important issue in the analysis of network data. Computational complexity theory distinguishes between ‘tractable’ and ‘intractable’ problems by breaking them into two groups – those that are solvable with an algorithm that runs in polynomial time, and those that are not. A polynomial-time algorithm with n inputs will run in time $O(n^p)$ for some $p > 0$. If an algorithm is not polynomial for any choice of p, it is super-polynomial. For example, combinatorially exhaustive algorithms are often exponential, in the sense that they run in $O(a^n)$ time for some $a > 1$. Except for settings with very small n, it is unlikely to be feasible in practice to wait for super-polynomial algorithms to complete; the same is effectively true even for polynomial algorithms having sufficiently large p, with respect to n.

In the case of graph algorithms, the complexity of an algorithm usually is a function of both N_v and N_e. Ideally, one would like the running time of such algorithms to scale linearly in these values (i.e., with $p = 1$), while also keeping memory requirements under control. In practice, for graphs of any reasonably large dimension, a cubic algorithm (i.e., with $p = 3$) is generally pushing the upper limits for what may be computed in an acceptable amount of time. In the case of massive network graphs, however, often nothing less efficient than linear running time is feasible. The design of efficient algorithms is frequently nontrivial, with a more naive, inefficient algorithm being improved only through clever use of storage, indexing, or redundancies in the quantities to be computed.

Consider, for example, the notion of a ‘search’ on a graph, which we can picture as a process of moving outward from a given vertex towards the other vertices in the graph, seeking to fulfill a given criterion. The two most basic search algorithms, *breadth-first search* (BFS) and *depth-first search* (DFS), each seek to ‘discover’ the set of all vertices $j \in V$ that are reachable from a given source vertex i_s. Both algorithms run in $O(N_v + N_e)$ time, and differ from each other primarily in the manner in which they choose to discover other vertices. The BFS algorithm works outward from i_s, first discovering vertices adjacent to i_s (i.e., one ‘hop’ away), then continuing to vertices two hops away, then three hops away, and so on, until all reachable vertices are discovered. The output of the algorithm is a tree, rooted at i_s and organized so that the path from i_s to a reachable vertex j in the tree corresponds to the shortest path from i_s to j in G. Conversely, the DFS algorithm, as its name suggests, instead proceeds from i_s by delving as deeply into G as possible from the first adjacent vertex to i_s, after which it iteratively backtracks to the most recently discovered vertex j for which there are undiscovered edges to be explored.

The usefulness of these two search algorithms, and the choice of which to use when, derives from the context of the actual task to be performed. For example, the BFS algorithm is at the core of standard algorithms requiring shortest path information, like Prim's algorithm for producing a minimum spanning tree[6] and Dijkstra's algorithm for finding all shortest paths from a single source i_s in a directed graph. The latter algorithms can be implemented to run in $O(N_e + N_v \log N_v)$ and $O(N_v^2 \log N_v + N_v N_e)$ time, respectively. The DFS algorithm is often a sub-routine in a larger algorithm, such as the 'topological sort algorithm,' which can be used to determine whether a directed graph G is acyclic or not, and algorithms for decomposing G into its strongly connected components, all of which can be implemented in linear time.

Interestingly, there are many problems for which it is unknown whether there even exists a polynomial-time algorithm to solve them. The study of such problems has been a deep and fundamental endeavor of theoretical computer science since the early 1970's. Within this area there is a formal framework for classifying problems by how difficult they *may* be. Decision problems (i.e., problems requiring a 'yes' or 'no' answer) that are shown to be *NP-hard* or *NP-complete* are considered quite likely to be intractable.[7] For optimization problems, wherein we seek to find a solution maximizing or minimizing a given objective function, the problem is referred to by these same terms if the associated decision problem of confirming or rejecting a candidate solution as optimal can be categorized as such. In the context of network graphs, the problem of finding a maximal clique in a graph is an example of an *NP*-complete optimization problem.

2.2 Background in Probability and Statistics

We here provide a brief overview of standard fundamental topics in probability and statistical inference. No attempt is made to be exhaustive in our presentation. Rather, our goal is simply to lay a foundation of terminology, notation, and concepts to be used later in the book, primarily in Chapters 5–9.

[6] A minimum spanning tree $T \in E$ is a tree that spans the full set of vertices V and for whom the sum of edge weights $W(T) = \sum_{e \in T} w_e$ is minimized.

[7] The designations *NP*-hard and *NP*-complete refer to two different but related notions of the difficulty of a problem. A precise understanding of the formal distinction between the two will not be necessary in this book. The interested reader is referred to the text by Cormen, Leiserson, Rivest, and Stein [100, Ch. 34].

2.2.1 Probability

2.2.1.1 Basic Definitions and Concepts

Given some sense of an ‘experiment’ (e.g., flipping a coin, determining if two classmates are friends, observing traffic volume on an Internet link, etc.) and an accompanying *sample space* Ω of potential outcomes of that experiment, we will denote the *probability* of an *event* $A \subseteq \Omega$ by $\mathbb{P}(A)$, where P is a function that assigns a number between zero and one to an event and that satisfies the Kolmogorov axioms. Specifically, we assume (i) $\mathbb{P}(A) \geq 0$ for any event A, (ii) $\mathbb{P}(\Omega) = 1$, and (iii) $\mathbb{P}(A \cup B) = \mathbb{P}(A) + \mathbb{P}(B)$ for any disjoint events A and B (i.e., such that $A \cap B = \emptyset$).

Two events A and B are said to be *independent* if $\mathbb{P}(A \cap B) = \mathbb{P}(A)\mathbb{P}(B)$. For events B such that $\mathbb{P}(B) > 0$, the *conditional probability* of A given B is $\mathbb{P}(A|B) = \mathbb{P}(A \cap B)/\mathbb{P}(B)$. It can be shown that A and B are independent if and only if $\mathbb{P}(A|B) = \mathbb{P}(A)$. From the facts that $\mathbb{P}(A \cap B) = \mathbb{P}(A|B)\mathbb{P}(B) = \mathbb{P}(B|A)\mathbb{P}(A)$ and $\mathbb{P}(B) = \mathbb{P}(B|A)\mathbb{P}(A) + \mathbb{P}(B|A^c)\mathbb{P}(A^c)$, where A^c indicates the complement of A within Ω, we obtain *Bayes rule*,

$$\mathbb{P}(A|B) = \frac{\mathbb{P}(B|A)\mathbb{P}(A)}{\mathbb{P}(B|A)\mathbb{P}(A) + \mathbb{P}(B|A^c)\mathbb{P}(A^c)} \ . \tag{2.4}$$

For three events A, B, and C, the events A and B are said to be *conditionally independent given C* if

$$\mathbb{P}(A \cap B \,|\, C) = \mathbb{P}(A \,|\, C)\mathbb{P}(B \,|\, C) \ . \tag{2.5}$$

The concept of a *random variable* as a mapping from a sample space Ω to the real-numbers $\mathbb{R}$, is used to formalize the notion of a measurement (i.e., data) obtained in association with an experiment. We will use upper case roman characters (e.g., X, Y, and Z) to denote random variables, and lower case (e.g., x, y, and z), to denote specific values taken on by the corresponding random variables. The *cumulative distribution function* (CDF) of a random variable X will be expressed as $F_X(x) = \mathbb{P}(X \leq x)$, and the *complementary CDF* (CCDF), as $\bar{F}_X(x) = 1 - F_X(x)$. The random variable X is said to ‘have distribution’ or ‘be distributed as’ F_X, which is often written as $X \sim F_X$. Similarly, we will use f_X to denote either the *probability mass function* (PMF), if X is a discrete random variable, or the *probability density function* (PDF), if X is a continuous random variable. For convenience, when the distinction is not important, we will typically use ‘PDF’ to refer to such functions. Furthermore, when the underlying random variable X is to be understood from context, we may write the CDF and PDF functions simply as F and f.

We represent the *expectation* of a random variable X as ${}_X \equiv \mathbb{E}(X) \equiv \mathbb{E}_X(X)$. For discrete random variables this is the sum of the quantities $x f_X(x)$ over the range of values x for which $f_X(x) > 0$, while for continuous random variables, it is the integral of the same.[8] In the case where the random variable $X = I_A$ is the *indicator*

[8] It is implicitly assumed in discussing an expectation that the corresponding sum or integral is well defined, in the sense of being finite; if it is infinite, the expectation is said not to exist.

of the event A (i.e., I_A is one if A occurs and zero if not), it may be shown that $\mathbb{E}(I_A) = \mathbb{P}(A)$.

For a function $g(x)$, the expectation of the random variable $Y = g(X)$ is simply $\mathbb{E}_Y(Y) = \mathbb{E}_X(g(X))$. Commonly occurring examples of g include $g(x) = x^k$, for k a positive integer, from which we obtain the *k-th moment* $\mathbb{E}(X^k)$ of X, and $g(x) = (X - \ _X)^2$, from which we obtain the *variance* $\sigma_X^2 \equiv \mathbb{V}(X) \equiv \mathbb{E}(X - \ _X)^2$ of X. The quantity $\sigma_X = \sqrt{\mathbb{V}(X)}$ is called the *standard deviation* of X. If a random variable X is transformed linearly as $g(X) = cX + b$, then the expectation and variance of the new random variable have the forms $\mathbb{E}(cX + b) = c\mathbb{E}(X) + b$ and $\mathbb{V}(cX + b) = c^2\mathbb{V}(X)$.

These concepts extend easily to several random variables. For example, in the case of two random variables X and Y, their *joint* CDF will be denoted $F_{X,Y}(x,y) = \mathbb{P}(X \leq x, Y \leq y)$, and their joint PDF, similarly as $f_{X,Y}(x,y)$. The *marginal* densities f_X and f_Y may be obtained by integrating (or summing, in the case of discrete random variables) the bivariate density $f_{X,Y}(x,y)$ over all values of y or x, respectively. The variables X and Y are said to be independent if $\mathbb{P}(X \in A, Y \in B) = \mathbb{P}(X \in A)\mathbb{P}(Y \in B)$, for all pairs of events A and B, in which case we have $f_{X,Y}(x,y) = f_X(x)f_Y(y)$. From this last fact, we also have the useful relation $\mathbb{E}(XY) = \mathbb{E}(X)\mathbb{E}(Y)$.

The *conditional* PDF of X, given $Y = y$, is defined as $f_{X|Y}(x|y) = f_{X,Y}(x,y)/f_Y(y)$, for all y such that $f_Y(y) > 0$. Note that, for fixed y, $f_{X|Y}(x|y)$ is itself a well-defined PDF in x, over an appropriate range of values. Hence, for example, there is an accompanying notion of *conditional expectation*, $\mathbb{E}(X|Y = y) = \int x f_{X|Y}(x|y)dx$. It should be noted that this quantity is a function of the value y.

The notion of conditional independence, as defined for events in (2.5), also extends to random variables. Specifically, the random variables X and Y are said to be *conditionally independent given Z* if

$$f_{X,Y|Z}(x,y|z) = f_{X|Z}(x|z)f_{Y|Z}(y|z) \ , \tag{2.6}$$

for all x, y, and z.

When we have a collection of n random variables $X_1, \ldots, X_n$ to which we wish to refer as a unit, in either vector or matrix form, we will typically do so through bold-faced type (e.g., $\mathbf{X} = (X_1, \ldots, X_n)^T$). We will denote the corresponding PDF by $f(\mathbf{x}) \equiv f(x_1, \ldots, x_n)$, the mean vector, by $\ \equiv (\mathbb{E}(X_1), \ldots, \mathbb{E}(X_n))^T$, and the *variance-covariance* matrix, by $\boldsymbol{\Sigma} \equiv \mathbb{V}(\mathbf{X}) \equiv [\mathsf{Cov}(X_i, X_j)]_{i,j}$, where $\mathsf{Cov}(X_i, X_j) = \mathbb{E}[(X_i - \ _{X_i})(X_j - \ _{X_j})]$ and $\mathsf{Cov}(X_i, X_i) = \mathbb{V}(X_i)$. Scaling the covariances by the standard deviations of the corresponding random variables, we obtain the *correlation* $\rho_{X_i,X_j} = \mathsf{corr}(X_i, X_j) = \mathsf{Cov}(X_i, X_j)/\sigma_{X_i}\sigma_{X_j}$.

If an n-length random vector $\mathbf{X}$ is transformed linearly as $\mathbf{CX} + \mathbf{b}$, where $\mathbf{C}$ is an $m \times n$ real-valued matrix and $\mathbf{b}$ is a m-length real-valued vector, then

$$\mathbb{E}(\mathbf{CX} + \mathbf{b}) = \mathbf{C}\mathbb{E}(\mathbf{X}) + \mathbf{b} \quad \text{and} \quad \mathsf{Cov}(\mathbf{CX} + \mathbf{b}) = \mathbf{C}\mathbb{V}(X)\mathbf{C}^T \ . \tag{2.7}$$

In the special case that the elements of $\mathbf{X}$ are uncorrelated (i.e., $\rho_{X_i,X_j} = 0$ for all $i \neq j$), the expression for the covariance reduces to $\sum_{i=1}^{n} C_{ii}^2 \mathbb{V}(X_i)$.

When the element X_i's in a random vector $\mathbf{X}$ are in fact all independent and 'copies' of a random variable X, with common distribution F_X, they will be said to be *independent and identically distributed* or 'iid'. This condition is often expressed in the notation $X_1, \ldots, X_n \sim F$. Note that in many cases of interest, the variables may be independent, but from different distributions.

For dependent random variables, it is common to use certain assumptions of conditional independence to simplify analytical and numerical calculations. The notion of a *Markov chain* is arguably the prototypical example. We say that a sequence of integer-valued random variables $X_0, X_1, X_2, \ldots$ is a (homogeneous) Markov chain if

$$\begin{aligned} \mathbb{P}\left(X_{m+1} = j \,|\, X_m = i, X_{m-1} = i_{m-1}, \ldots, X_0 = i_0\right) &= \mathbb{P}\left(X_{m+1} = j \,|\, X_m = i\right) \\ &= P_{ij} \ , \end{aligned} \tag{2.8}$$

for any choice of $i_0, \ldots, i_{m-1}, i_m, j$, called *states*, and any positive integer m, which is typically thought of as a time index. This Markov chain is an example of a discrete-time stochastic process. The first equality in (2.8) states that, given the past history of the process up through time m, the distribution of the process at time $m+1$ depends only on its state at time m. The second equality indicates that this conditional distribution does not vary with m. The values P_{ij} are called *transition probabilities*.

Define $P_{ij}^{(n)}$ to be the probability that, starting in state i, the Markov chain is found in state j at n units of time (or 'transitions') later. The chain is called *irreducible* if any state can be reached from any other in some finite number of transitions (i.e., $P_{ij}^{(n)} > 0$ for some finite n for all states i, j). Importantly, under appropriate conditions,[9] irreducible chains have a well-defined *stationary distribution*, call it $\{\pi_j\}$, in the sense that the limit

$$\lim_{n \to \infty} P_{ij}^{(n)} = \pi_j \tag{2.9}$$

exists for each state j, with $\pi_j \geq 0$ and $\sum_j \pi_j = 1$, and it is uninfluenced by the starting state i.

On a final note, in regards to basic definitions and concepts, we point out that in many contexts we will find it is useful to consider a graph G itself as a random object. Formally, this is typically done by thinking of G as having been drawn at random from a collection of possible graphs, say $\mathscr{G}$, whose definition may be explicit or implicit. In such cases, the value $\mathbb{P}(G)$ will refer to the probability of drawing G from $\mathscr{G}$ under some stated sampling mechanism. Similarly, if $\eta(G)$ represents a real- or vector-valued quantity of characteristics of G (e.g., the number of vertices or

[9] Specifically, besides irreducibility, the chain must be *ergodic* in that (i) it is not constrained to cycle through states in a periodic manner (i.e., aperiodic), and (ii) from any given state, it is guaranteed to return to that state in a finite number of transitions, with the expected number of transitions also finite (i.e., positive recurrent).

the degree sequence), then $\mathbb{E}[\eta(G)]$ will denote the expected value of that quantity under the sampling on $\mathscr{G}$.

2.2.1.2 Asymptotic Statements

Often it is possible to produce statements regarding the behavior of a random quantity, as some other quantity, such as sample size, tends toward a limiting value. Such asymptotic statements can be quite useful in looking past the potentially horrendous (if not impossible) details of an exact calculation to concisely summarize the state of affairs when one is close enough to 'asymptopia.' The limiting relation in (2.9) is an example of such a statement. Expressions for the n-step transition probabilities $P_{ij}^{(n)}$ are generally not expressable in closed form other than in the simplest of cases. Nevertheless, under the assumed conditions, (2.9) will always hold true, and sometimes informative expressions for the limiting values π_j may be obtained. We quickly review here three canonical types of asymptotic statements that will be encountered in this book.

Let $A_1, A_2, \ldots$ be a sequence of events, in association with a suitably defined probability model. We say A_n occurs *with high probability* (WHP) if

$$\lim_{n \to \infty} \mathbb{P}(A_n) = 1 \ . \tag{2.10}$$

An example of such a statement is the *weak law of large numbers* (WLLN) for iid random variables $X_1, X_2, \ldots$, with common mean , which provides a characterization of the asymptotic behavior of the sample mean $\bar{X}_n = \sum_i X_i / n$. Specifically, it states that (2.10) holds, with A_n defined to be the event that $|\bar{X}_n - \quad| \leq \varepsilon$, for $\varepsilon > 0$ a fixed constant.

Probably more frequently cited in practice, is the analogous *strong law of large numbers* (SLLN), which states that, under the same conditions,

$$\mathbb{P}\left(\lim_{n \to \infty} \bar{X}_n = \quad \right) = 1 \ . \tag{2.11}$$

We note that the strong law implies the weak law, as the names suggest; the converse is not true.

Finally, there is the notion of a *central limit theorem*, the various versions of which state essentially that the probability distribution of suitably normalized sums of random variables tends in the limit of many such random variables to a Gaussian distribution[10] with mean zero and unit variance. For example, in the case of a sequence of iid random variables $X_1, X_2, \ldots$, with common mean and finite variance

[10] A random variable X is said to have a *Gaussian* or *normal* distribution, with mean parameter and variance parameter σ^2 if

$$f_X(x) = (2\pi\sigma^2)^{-1/2} \exp\{-(x - \quad)^2/(2\sigma^2)\} \ .$$

If $\quad = 0$ and $\sigma = 1$, Z is said to be a *standard normal* random variable.

$\sigma^2 > 0$, the central limit theorem states that

$$\lim_{n\to\infty} \mathbb{P}\left(\frac{\bar{X} -}{\sigma/\sqrt{n}} \leq x \right) = \Phi(x) \ , \tag{2.12}$$

where $\Phi(\cdot)$ is the CDF for a Gaussian random variable with mean zero and unit variance.

2.2.1.3 Simulation of Random Variables

Given a random variable X, with PDF f, it is frequently necessary in applications to compute quantities that can be expressed in integrals of the form

$$\mathbb{E}\left(g(X)\right) = \int g(x) f(x) dx \ , \tag{2.13}$$

for some function g of x (or sums of the analogous form, in the discrete case). The expression in (2.13) includes, for example, the expectation of X, its variance, and the probability that $X \in A$, for sets $A \subset \mathrm{I\!R}$. When computationally tractable, closed-form expressions are not available for this purpose, and numerical integration is infeasible, methods based on numerical simulation – that is, 'Monte Carlo' methods – generally may be used instead. Such methods work by generating copies $x_1, \ldots, x_n$ of X from f, and computing the stochastic approximation

$$\widehat{\mathbb{E}}\left(g(X)\right) = \frac{1}{n} \sum_{i=1}^{n} g(x_i) \ . \tag{2.14}$$

An appeal to an appropriate strong law of large numbers then guarantees that the sample average in (2.14) is close to (2.13), for suitably large n.

The challenge for the Monte Carlo approach therefore lies largely in developing suitable methods for simulating draws from the distribution f. In the most common case, we are interested in making draws of iid copies from f. The starting point for doing so is the now-standard ability to generate so-called pseudo-random numbers. That is, to simulate draws of a uniform random variable on $[0,1]$ (i.e., with PDF $f(x) = 1$ for $x \in [0,1]$, and zero elsewhere). Methods to simulate draws from more sophisticated distributions f then generally utilize pseudo-random numbers in clever ways. For example, if X is a random variable with continuous CDF F, we can use the fact that the quantity $U = F(X)$, as a random variable, is distributed uniformly on $[0,1]$ to generate draws from X, through the relation $X \sim F^{-1}(U)$. This is sometimes referred to as the *inverse transform* method.

Sometimes it is too difficult to generate samples from the distribution f that we desire, but we are instead able to sample from another distribution, say h. Rewriting the expression in (2.13) as

$$\mathbb{E}\left(g(X)\right) = \frac{\int g(x) w(x) h(x) dx}{\int w(x) h(x) dx} \ , \tag{2.15}$$

where $w(x) = f(x)/h(x)$, with the method of *importance sampling* we form the quantity

$$\widehat{\mathbb{E}}_h\left(g(X)\right) = \frac{(1/n)\sum_{i=1}^{n} w(x_i)g(x_i)}{(1/n)\sum_{i=1}^{n} w(x_i)} \quad , \tag{2.16}$$

using draws $x_1,,\ldots,x_n$ from h. The name 'importance sampling' derives from the fact that the method can be used to reduce the error in our approximation of (2.13), by concentrating our sampling in that part of the support of f where it is most needed.

Also commonly used in practice are methods based on Markov chains i.e., *Markov chain Monte Carlo* (MCMC) methods. These methods generate draws from a particular Markov chain, therefore yielding dependent random variables, where the chain is designed so that the limiting stationary distribution (e.g., as in (2.9), for the case of discrete states) is in fact the distribution f of interest to us. MCMC methods can therefore be substantially more complicated than methods like those just described. However, designed appropriately, they allow us to generate samples from quite complex distributions and are used frequently in network modeling.

Perhaps the most widely used type of MCMC algorithm is the Metropolis-Hastings algorithm, which is quite general and requires only that we can evaluate the PDF/PMF.[11] It is an example of a so-called *acceptance-rejection method*, wherein possible values are proposed for the next draw, but only some are accepted. For simplicity, consider the case where f is a PMF for a discrete random variable X. Suppose the current value generated from our Markov chain is $X_m = i$. We then propose a candidate state j for X_{m+1}, where j is drawn according to a proposal distribution $\{q_{ij}\}_i$. We accept the proposed state j with probability

$$\alpha_{ij} = \begin{cases} \min\left[\frac{f(j)q_{ji}}{f(i)q_{ij}}\, , 1\right], & \text{if } f(i)q_{ij} > 0 \ , \\ 1, & \text{otherwise} \ , \end{cases} \tag{2.17}$$

and reject with probability $1-\alpha_{ij}$.

The primary challenge here is to design an appropriate proposal distribution $\{q_{ij}\}$, one that not only produces a Markov chain with the appropriate stationary distribution,[12] but which also (effectively) converges to the stationary distribution in a reasonable number of iterations (called the 'burn-in' period) and moves around the space of possible values of X quickly enough (called 'good mixing'). Note that if we wish to simulate independent draws from f, it is necessary to thin the samples we obtain, leaving an appropriate interval between each sample we use, where the length of the interval will depend on the nature of the dependence in the underlying Markov chain.

[11] In fact, it is necessary only that we be able to evaluate the PDF/PMF up to a constant of proportionality.

[12] Hence, at a minimum the proposal distribution must produce an irreducible and ergodic Markov chain. In fact, the chain must also be *time-reversible*, in that the relation $f(i)q_{ij}\alpha_{ij} = f(j)q_{ji}\alpha_{ji}$ must hold for all i, j.

2.2.2 Principles of Statistical Inference

Statistical inference refers to the process whereby, given observations $x_1,\ldots,x_n$ of $X_1,\ldots,X_n \sim F$, we attempt to extract information – or 'learn' – about F from the data $\mathbf{x}$. We may be interested in something as simple as a scalar summary characteristic of F, such as the mean $_X$, or in something as potentially complex as the entire CDF F itself, or its PDF, f. In many cases, our observations are of the form (y_i,x_i) and it is the relation between the random variables Y and X that is of interest. Here the variable Y is called the *response* or the *outcome,* while the variable X is called the *predictor* or the *feature.* Common inferential goals in this setting include learning about the value of $\mathbb{E}(Y|X=x)$, as a function of x, or foretelling a yet-to-be observed value Y_* from the observation $X_* = x_*$.

Statistical inference typically is conducted within the context of a model(s). A *statistical model,* broadly speaking, refers to the specification, either explicitly or implicitly, of a collection $\mathscr{F}$ of CDFs to which F may belong. For example, it is common to specify models of the form $\mathscr{F} = \{f(x;\theta) : \theta \in \Theta\}$. If Θ is finite in dimension and does not grow in size with the sample size n, then $\mathscr{F}$ is said to be a *parametric* model. An example is the model $\mathscr{F} = \{$All Gaussian densities with mean and variance σ^2 : $\in \mathbb{R}, \sigma > 0\}$. On the other hand, if Θ is infinite in dimension or grows in size with n, then $\mathscr{F}$ is called a *nonparametric* model.[13] An example is the model $\mathscr{F} = \{$All CDFs F with continuous density $f\}$.

Most introductory (i.e., 'Stat 101') treatments of statistics deal almost exclusively with parametric models. However, the use of nonparametric models has become increasingly common across the sciences. This fact has important implications, as nonparametric modeling can bring with it issues not encountered in parametric modeling. Such issues can arise in the analysis of network data, as we discuss below in Section 2.3.

Standard treatments of statistical inference typically characterize problems as falling into either of two broad categories: *estimation* and *hypothesis testing.* In addition, there is the related problem of *prediction,* which may be viewed as a variant of the estimation problem.

Estimation consists of making an 'educated guess' about the value of a parameter θ, such as a mean or a vector of coefficients β in a regression model $\mathbb{E}(Y|\mathbf{X}=\mathbf{x}) = \mathbf{x}^T\beta$, based on the observed data and any other information built into the model $\mathscr{F}$. Estimates may be *point estimates* (i.e., estimates taking on a single value $\hat{\theta}$) or *set estimates* (i.e., estimates consisting of a range of possible values over a set), such as an interval $(\hat{\theta}^{low}, \hat{\theta}^{high})$. Sometimes it is not a parameter in a model that is of interest so much as an unobserved random variable, such as the next value X_{n+1} in a sequence of random variables $X_1,\ldots,X_n$ or a new response value Y_* corresponding to an observed predictor $\mathbf{X}_* = \mathbf{x}_*$. This form of estimation is typically called prediction. When Y is a discrete random variable (e.g., $Y = \pm 1$), the predic-

[13] The distinction between parametric and nonparametric models can be made more rigorous than this, but the definitions given here will suffice.

tion problem is called *classification.* Predictions too may take either point or set forms.

Hypothesis testing involves weighing the evidence contained in observations, say $x_1,\ldots,x_n$, in support of a hypothesized theory. This theory is said to constitute the *null hypothesis* and is typically written as H_0. Often one explicitly specifies an alternative to the null as well, the *alternative hypothesis*, which we will write as H_1. For example, a canonical case is that of testing whether a parameter θ is equal to a particular value θ_0 or not (i.e., $H_0 : \theta = \theta_0$ is posited against $H_1 : \theta \neq \theta_0$). Performing a hypothesis test generally amounts to computing a suitable *test statistic* $t_{obs} = T(\mathbf{x})$ from the data and comparing it to a *reference distribution* F_T that describes the (random) behavior of the variable T in the case that H_0 is true. If t_{obs} is found to lie in the 'extreme' regions of the support of F_T e.g., $F_T(t_{obs})$ or $\bar{F}_T(t_{obs})$ are sufficiently 'small,' then the null hypothesis is rejected. Otherwise, the null is retained. In practice, a reference distribution may derive from previous measurements, analytical results (such as asymptotic theory), or computationally intensive re-use of the original sample.

For most inference problems that have been studied to any reasonable extent, there are likely a variety of methods that have been proposed. Differences among methods can be attributed to aspects ranging from slight variations in model specification (e.g., a variance is assumed known, versus unknown), to alternate technical 'machinery' (e.g., estimation through method of moments, versus maximum likelihood), to preferences in overall paradigm (e.g., frequentist versus Bayesian[14]). One's choice of methodology is generally influenced by some combination of empirical performance, computational tractability, theoretical properties (if such have been derived), and, of course, personal preference.

2.2.3 Methods of Statistical Inference: Tutorials

Rather than attempting a general overview of various methods of statistical inference, which is clearly beyond the scope of this chapter, we will instead review be-

[14] Traditionally, the two primary 'schools of thought' on inference in the statistics literature have been *frequentist* and *Bayesian.* In essence, they differ in (i) how they interpret probability, and (ii) how they view parameters. The frequentist interpretation of the probability $\mathbb{P}(A)$ is as the relative frequency at which the event A occurs in the limiting case of an infinite number of repetitions of an experiment and, as such, as an objective property of nature. A parameter θ is treated as a fixed but unknown constant. In particular, parameters are not treated as random. In contrast, formally, the Bayesian interpretation of the probability $\mathbb{P}(A)$ is as something that reflects a subjective level of belief in the occurrence of A. Accordingly, a parameter θ is viewed as a fixed quantity about which one nevertheless can make probability statements. Given an unknown parameter θ, with distribution F_θ (called the prior distribution), and data $\mathbf{x}$, with conditional distribution $F_{\mathbf{X}|\theta}$ (called the likelihood), inference on θ is conducted through statements involving the distribution $F_{\theta|\mathbf{X}}$ (called the posterior), which is defined through the use of Bayes rule. Informally, the Bayesian approach is often used as a convenient framework within which to combine models for both measurement (i.e., likelihood) and parameters (i.e., prior).

low just a handful of methods that will be most relevant to future chapters in this book, through the use of two tutorials. These tutorials serve to illustrate the chosen methods in the context of two classical settings: (i) inference of a mean based on a single sample, and (ii) regression and prediction with linear models. These two settings were chosen particularly for their likely familiarity, and will be useful as early as Section 2.3, when we contrast them with some of the settings to be seen later in the book in the analysis of network data.

2.2.3.1 Inference of a Mean

Suppose we have n iid observations $x_1, \ldots, x_n$ of a random variable $X \sim F$. We will consider the problem of performing statistical inference on the mean , arguably the most classical and canonical problem of its kind. While this problem is clearly important from a practical perspective, it also turns out that principles and techniques for this problem often lie at the core of methods of inference for more complicated parameters θ.

A natural estimator of is, of course, the sample mean $\bar{x} = \sum_i x_i / n$. While its usage can be motivated on intuitive grounds, we note that it is also a simple example of a *method of moments* estimator. That is, suppose our aim is to learn an unknown parameter θ in a model $\mathscr{F} = \{f_X(x;\theta) : \theta \in \Theta\}$. Denote the k-th moment of X as $\alpha_k(\theta) = \int x^k f(x;\theta)dx$. The method of moments dictates that we equate the first K moments, for a given choice of K, with their sample versions $\hat{\alpha}_k = \sum_i x_i^k / n$, and solve the resulting system of K equations for θ. Specifically, we form the system

$$\begin{aligned} \alpha_1(\theta) &= \hat{\alpha}_1 \\ \alpha_2(\theta) &= \hat{\alpha}_2 \\ \vdots &= \vdots \\ \alpha_K(\theta) &= \hat{\alpha}_K \end{aligned} \tag{2.18}$$

and set the estimator $\hat{\theta}$ to be that value of θ solving (2.18). For the problem of estimating a scalar mean , we have $\theta = \quad = \alpha_1$, and thus need only $K = 1$ equation (i.e., $\alpha_1(\theta) = \hat{\alpha}_1$). Solving this trivially yields the estimator $\hat{}^{MM} = \bar{x}$.

The sample mean also is the estimator that results from applying the *maximum likelihood* principle, in the case that our model class $\mathscr{F}$ assumes, say, a Gaussian distribution with mean and (known) variance $\sigma^2 = 1$. In maximum likelihood estimation, we estimate a parameter of interest θ by the value $\hat{\theta}^{ML}$ that maximizes the likelihood $\mathscr{L}(\theta)$, where

$$\mathscr{L}(\theta) = \prod_{i=1}^{n} f_X(x_i;\theta) \tag{2.19}$$

is the joint density of the X_i's interpreted as a function of θ. This approach is equivalent to maximizing the (natural) logarithm $\ell(\theta) = \log \mathscr{L}(\theta) = \sum_i \log f(x_i;\theta)$, called the *log-likelihood*, which is sometimes an easier object to deal with, both analytically and numerically. In the case that $\theta =$ and the density f is the Gaussian density described above, we find that

$$\ell(\quad) = c - \frac{1}{2}\sum_{i=1}^{n}(x_i - \quad)^2 \ , \tag{2.20}$$

where c is a constant with respect to the maximization in . A small calculation confirms that (2.20) is maximized by $\hat{\ }^{ML} = \bar{x}$.

In the same way that the data $x_1, \ldots, x_n$ are realizations of random variables $X_1, \ldots, X_n$, the sample mean $\bar{x}$ may itself be associated with a random variable, say $\bar{X}$. The distribution of this random variable is called the *sampling distribution*. The accuracy of $\bar{X}$ can be characterized by its *mean squared error*,

$$\mathsf{MSE}(\bar{X}) = \mathbb{E}(\bar{X} - \quad)^2 \ . \tag{2.21}$$

This quantity can be decomposed as $\mathsf{MSE}(\bar{X}) = \mathsf{bias}^2(\bar{X}) + \mathbb{V}(\bar{X})$, where $\mathsf{bias}(\bar{X}) = \mathbb{E}(\bar{X}) -$. Note that since $\mathbb{E}(\bar{X}) =$, the sample mean has zero bias. That is, it is an *unbiased* estimator, which may be interpreted as saying that on average $\bar{X}$ is 'on target.' The quantity $\mathbb{V}(\bar{X})$ captures the variability of $\bar{X}$ and typically accompanies the sample mean through its role in defining the *standard error*, $\mathsf{se}(\bar{X}) = \sqrt{\mathbb{V}(\bar{X})} = \sigma/\sqrt{n}$, where $\sigma^2 = \mathbb{V}(X)$. If σ is not known, an estimated standard error $\hat{\mathsf{se}}$ can be formed by replacing σ by the standard deviation of the data.

Often the standard error is used in a more formal fashion to create, with the sample mean, an interval estimate of the form

$$C(\mathbf{x}) = \left(\bar{x} - z_{\alpha/2}\,\hat{\mathsf{se}}\,,\, \bar{x} + z_{\alpha/2}\,\hat{\mathsf{se}}\right) \ . \tag{2.22}$$

Here $z_{\alpha/2}$ is that value for which a standard normal random variable Z has $\mathbb{P}(-z_{\alpha/2} \leq Z \leq z_{\alpha/2}) = 1 - \alpha$. Under fairly general conditions, it may be asserted that $\mathbb{P}(\ _X \in C(\mathbf{X})) \approx 1 - \alpha$, with the approximation improving with increasing sample size.[15] The interval $C(\mathbf{x})$ in (2.22) is called a $100(1-\alpha)\%$ *confidence interval.*

Now suppose that, rather than estimate , we wish to test the null hypothesis $H_0 : \ = \ _0$ versus the alternative, $H_1 : \ \neq \ _0$. A familiar test statistic is the Wald statistic

$$T(\mathbf{x}) = \frac{\bar{x} - \ _0}{\hat{\mathsf{se}}(\bar{x})} \ , \tag{2.23}$$

commonly called the *t-statistic* in this setting, which is just the observed $\bar{x}$ shifted by its mean (or 'centered') under the null hypothesis and scaled by its estimated standard error. For moderately large n, the value $T(\mathbf{X})$ has a distribution approximately equal to that of a standard normal distribution.

[15] Note that confidence intervals are a frequentist inference tool, and hence $\ _X$ is considered fixed. The randomness here is inherent in the interval $C(\mathbf{X})$, and hence the corresponding probability statement is in regards to how likely the event is that the random C contains the unknown, fixed $\ _X$.

Standard practice is to compare the observed value $T_{obs} = T(\mathbf{x})$, for $\mathbf{X} = \mathbf{x}$, to the distribution of a standard normal variable Z in either of two manners. One may specify a value α, and report whether or not $|T_{obs}| \geq z_{\alpha/2}$ holds, rejecting H_0 if it does, and retaining H_0, if not. Alternatively, one may simply report the probability $p = \mathbb{P}_{\ 0}(|T(\mathbf{X})| \geq |T_{obs}|)$, where $\mathbb{P}_{\ 0}$ indicates that the probability is calculated under the assumption that $\ = {}_0$. In the first case, the emphasis is on the test as a control procedure, and the probability of incorrectly rejecting H_0 is constrained to be (approximately) no more than α in value. The value α is called the *significance level* of the test. In the second case, the emphasis is on presenting some measure of the ‘weight of evidence’ against H_0, since a simple conclusion of ‘reject’ or ‘retain’ is not very informative. The probability p is called the *p-value* and is the smallest significance level α at which H_0 would be rejected. Smaller values of p (e.g., p less than $0.05, 0.01, 0.001$, etc.) indicates increasingly less support for the null hypothesis.

When there are multiple tests to be performed, it can become necessary to modify the manner in which we discuss error rates and control. This phenomenon is the so-called *multiple testing problem* (also referred to as the multiple comparisons problem). Suppose there are m independent samples $\mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(m)}$, and m corresponding test statistics $T_j = T(\mathbf{x}^{(j)})$, one for each sample $j = 1, \ldots m$. While we know how to control each individual test so as to have an individual significance level α, in some settings it may be desirable to instead control some analogous notion of error for the family of m tests as a whole. Note that if each test T_j is controlled at the α significance level, then the probability of making at least one false rejection over the set of m tests is $1 - (1-\alpha)^m$. For successively larger m, it is therefore necessary to use correspondingly smaller α for each individual test. The well-known *Bonferonni correction* procedure advocates using significance levels of α/m for each test, in an effort to crudely bound the overall error at a rate of α. Unfortunately, for large m this approach can lead to each individual test having very poor statistical power.[16]

A common approach to addressing this issue is to focus not on the family-wise error rate, but rather on the rate of false discoveries. The *false discovery rate* (FDR) is defined to be

$$\mathsf{FDR} = \mathbb{E}\left(\frac{R_{false}}{R} \,\middle|\, R > 0 \right) \mathbb{P}(R > 0) \quad , \tag{2.24}$$

where R is the number of rejections among our m tests and R_{false} is the number of false rejections. Benjamini and Hochberg [33] provide a method for controlling the FDR at a user-specified level γ by rejecting the null hypothesis for all tests associated with p-values $p_{(j)} \leq (j/m)\gamma$, where $p_{(j)}$ is the j-th smallest p-value among the m tests. A popular variation on this method is that of Storey [370], in which the rate of *positive* false discoveries (i.e., pFDR) is the focus, where the pFDR is just the quantity in (2.24) without the probability $\mathbb{P}(R > 0)$. In this context, Storey proposes an analogue of the p-value for multiple testing, called the q-value, as well as a method for estimating q-values in practice. With q-values one rejects all null

[16] The term *power* formally refers to the probability of rejecting the null hypothesis H_0 when the alternative H_1 is in fact true (i.e., the probability of detecting a deviation from the null).

hypotheses in a set of m tests for which the corresponding q-values are less than or equal to some user-specified q^*, with the expectation that approximately $q^* \times m$ of those rejections will be false discoveries.

Returning to the problem of estimating , let us examine a Bayesian approach.[17] From the Bayesian perspective the parameter itself is equipped with a prior distribution, F , and inference on is based on the posterior distribution $F_{\ |\mathbf{X}}$. Suppose that, conditional on , the observations $x_1, \ldots, x_n$ are modeled as iid from a Gaussian distribution with (unknown) mean and known variance σ^2, similar to the frequentist model above. Additionally, however, suppose that we model F as Gaussian too, with mean a and variance b^2. Our choice of a indicates our *a priori* belief about 'typical' values of , while the choice of b^2 reflects the strength of that belief.

The choice of Gaussian distribution for the prior is typically made for convenience, as it may then be shown that $F_{\ |\mathbf{X}}$ (i.e., the conditional distribution of , given $\mathbf{X} = \mathbf{x}$) is itself Gaussian as well, with expectation $\bar{}$ and variance τ^2, where

$$\bar{} = \mathbb{E}(\ |\mathbf{X} = \mathbf{x}) = w\bar{x} + (1-w)a\ , \tag{2.25}$$

for $w = (\mathsf{se}^{-2})/(\mathsf{se}^{-2} + b^{-2})$, and

$$\frac{1}{\tau^2} = \mathbb{V}^{-1}(\ |\mathbf{X} = \mathbf{x}) = \frac{1}{\mathsf{se}^2} + \frac{1}{b^2}\ . \tag{2.26}$$

Here se is short-hand for $\mathsf{se}(\bar{X}) = \sigma/\sqrt{n}$.

The posterior mean $\bar{}$ in (2.25) is a natural point estimate for , as it corresponds to the most likely value under the posterior[18] (i.e., the mode). It takes the form of a weighted average of the sample mean $\bar{x}$ and the prior mean a. The weights w and $1-w$ reflect how the variance b^2 in the prior (corresponding to how certain we are in asserting that $= a$) and the variance se^2 (summarizing the uncertainty in the data-based estimator $\bar{X}$) compare. Note that for n large enough, we will have $w \approx 0$, in which case $\bar{} \approx \bar{x}$. That is, with enough observations, the information in the data will overwhelm that in the prior; for lesser amounts of data, however, the prior information plays an increasingly greater role.

Producing a Bayesian interval estimate of under our model, in analogy to a frequentist confidence interval, amounts to finding a set C such that $\mathbb{P}(\ \in C|\mathbf{X} = \mathbf{x}) = 1 - \alpha$. The probability underlying the definition of C here is with respect to the posterior distribution,[19] and that, as we have already remarked, is Gaussian. Therefore, a natural choice is the interval $C = (\bar{} - z_{\alpha/2}\,\tau, \bar{} + z_{\alpha/2}\,\tau)$. For large

[17] There is also a Bayesian analogue of hypothesis testing, but we will not address that here, as it is somewhat more complicated to present and will play little to no role in the remainder of the book.

[18] Additionally, it is the estimator $\hat{}$ that minimizes the quantity $\mathbb{E}(\hat{}(\mathbf{X}) - \ )^2$, where the expectation is with respect to the joint distribution of and $\mathbf{X}$. As such, it is an example of a so-called *Bayes-optimal* estimator.

[19] Note that under the Bayesian paradigm, is modeled as a random variable and hence, unlike with a frequentist confidence interval, the probability statement is in regard to how likely the event is that the random be in the fixed (conditional on $\mathbf{X} = \mathbf{x}$) interval C.

n, this interval will be very close to the $100(1-\alpha)\%$ confidence interval given by $(\bar{x} - z_{\alpha/2}\,\mathsf{se}, \bar{x} + z_{\alpha/2}\,\mathsf{se})$.

On a final note, we point out that our illustration of doing statistical inference on a mean , and the use of the sample mean $\bar{x}$ to do so, can be deceptive in its simplicity. Much of what has been described above generalizes in a fairly explicit manner to the case of estimating an arbitrary parameter θ, based on iid observations $x_1, \ldots, x_n$ of a random variable $X \sim F$. For example, given an estimator $\hat{\theta}$, we may still define the mean squared error $\mathsf{MSE}(\hat{\theta}) = \mathbb{E}(\hat{\theta} - \theta)^2$, and the decomposition into bias and variance components is identical to that of the sample mean. Although not all estimates $\hat{\theta}$ will necessarily be unbiased, in many settings the bias is relatively small compared to the variance. However, $\mathsf{se}(\hat{\theta}) = \sqrt{\mathbb{V}(\hat{\theta})}$ may not always be amenable to estimation by a simple 'plug-in' method. In such settings, an estimator based on 'reuse' of the sample, such as a bootstrap estimator, may be preferable. See the text by Efron and Tibshirani [131]. Similarly, under fairly general conditions, the quantity $(\hat{\theta} - \theta)/\widehat{\mathsf{se}}(\hat{\theta})$ has a distribution that is approximately standard normal, for moderately large n, in the case of both method of moments and maximum likelihood estimators. This, therefore, means, for example, that confidence intervals of the form $(\hat{\theta} - z_{\alpha/2}\,\widehat{\mathsf{se}}(\hat{\theta})\,,\, \hat{\theta} + z_{\alpha/2}\,\widehat{\mathsf{se}}(\hat{\theta}))$ have approximate probability $1-\alpha$ of containing θ, and test statistics of the form $T(\mathbf{x}) = (\hat{\theta} - \theta_0)/\widehat{\mathsf{se}}(\hat{\theta})$ can be compared to $z_{\alpha/2}$ to test $H_0 : \theta = \theta_0$ versus $H_1 : \theta \neq \theta_0$ at the α significance level. Elements of the Bayesian methodology outlined above also generalize to a reasonable extent, but we will not go into detail here.

2.2.3.2 Linear Regression Inference

Consider now the situation where our data are of the form $(y_1, x_1), \ldots, (y_n, x_n)$, being n iid observations of random variables (Y, X) with joint distribution $F_{Y,X}$. Suppose we wish to study the relationship between Y and X. A common approach to doing so is *regression,* in which the focus is on the *regression function*

$$r(x) = \mathbb{E}(Y|X=x) = \int y f_{Y|X}(y|x)dy \quad . \tag{2.27}$$

We will review here various aspects of the *linear regression* framework, in which $r(x)$ is assumed to have a linear form. Not only is this framework classical and a still-standard part of the toolbox of most practitioners, but its underlying principles often remain at the core of more sophisticated methods of regression.

To begin, consider the *simple linear regression* model, which specifies that, given $X_i = x_i$, for each $i = 1, \ldots, n$,

$$y_i = \beta_0 + \beta_1 x_i + \varepsilon_i \quad , \tag{2.28}$$

where the ε_i are iid random variables for which $\mathbb{E}(\varepsilon_i|X_i = x_i) = 0$ and $\mathbb{V}(\varepsilon_i|X_i = x_i) = \sigma^2$. In other words, the y_i's are modeled as 'noisy' measurements of the line

$r(x) = \mathbb{E}(Y|X = x) = \beta_0 + \beta_1 x$, where the noise terms are, conditional on the X_i's, independent, with zero mean and constant variance.

Specification of a linear regression function relating Y_i and X_i reduces the problem of learning the function $r(x)$ to one of learning the coefficients (β_0, β_1). Intuitively, a good choice of coefficients is that pair (β_0, β_1) for whom the distances from the values y_i to the corresponding line are minimized. That is, we seek to minimize the size of the *residuals* $y_i - (\beta_0 + \beta_1 x_i)$. Traditionally this is done through minimization of the *residual sum of squares*

$$\mathsf{RSS} = \sum_{i=1}^{n} (y_i - (\beta_0 + \beta_1 x_i))^2 \quad . \tag{2.29}$$

The corresponding values, say $\hat{\beta}_0$ and $\hat{\beta}_1$, that achieve this minimum are called the *least squares estimates* of β_0 and β_1.

More generally, suppose that we observe iid data $(y_1, \mathbf{x}_1), \ldots, (y_n, \mathbf{x}_n)$, where $\mathbf{x}_i = (x_{i1}, \ldots, x_{ip})^T$ is now a $p \times 1$ vector. The *multiple linear regression* model specifies that, in analogy to (2.28),

$$y_i = \sum_{j=1}^{p} x_{ij}\beta_j + \varepsilon_i \quad , \tag{2.30}$$

where $\mathbb{E}(\varepsilon_i|\mathbf{X}_i = \mathbf{x}_i) = 0$ and $\mathbb{V}(\varepsilon_i|\mathbf{X}_i = \mathbf{x}_i) = \sigma^2$. Typically the first element of the vectors $\mathbf{x}$ is fixed at 1, which provides an intercept term for the model, the role played by β_0 in (2.28). Letting $\mathbf{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_n]^T$ be the $n \times p$ matrix obtained by stacking the n vectors $\mathbf{x}_i$ in rows, and writing $\beta = (\beta_1, \ldots, \beta_p)^T$, $\varepsilon = (\varepsilon_1, \ldots, \varepsilon_n)^T$, and $\mathbf{y} = (y_1, \ldots, y_n)^T$, the model in (2.30) can be re-expressed as $\mathbf{y} = \mathbf{X}\beta + \varepsilon$.

The model in (2.30) specifies the y_i to be 'noisy' measurements of the hyperplane defined by $r(\mathbf{x}) = \mathbf{x}^T\beta$, where the argument $\mathbf{x}$ is now a $p \times 1$ vector. Therefore, we again seek a value for β that minimizes the size of the residuals. In this case, employing our matrix shorthand, the residuals may be expressed as $y_i - \mathbf{x}_i^T\beta$, and the residual sum of squares takes the form $\mathsf{RSS} = (\mathbf{y} - \mathbf{X}\beta)^T(\mathbf{y} - \mathbf{X}\beta)$. A calculus argument shows that minimizing $\mathsf{RSS} = \mathsf{RSS}(\beta)$ in β yields the least squares estimate

$$\hat{\beta} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y} \quad . \tag{2.31}$$

Note that $\hat{\beta}$ is only well defined if the $p \times p$ matrix $\mathbf{X}^T\mathbf{X}$ is invertible, which is true if and only if the $n \times p$ matrix $\mathbf{X}$ is of full column rank p.

The estimate $\hat{\beta}$ is a linear combination of the elements of $\mathbf{y}$, which are realizations of a random vector $\mathbf{Y}$. Accordingly, it can be shown using the identities in (2.7), that $\mathbb{E}(\hat{\beta}|\mathbf{X}) = \beta$ and $\mathbb{V}(\hat{\beta}|\mathbf{X}) = \sigma^2(\mathbf{X}^T\mathbf{X})^{-1}$. So the least squares procedure yields, under the model in (2.30), unbiased estimates[20] of the elements β_j in β, with standard errors $\mathsf{se}(\hat{\beta}_j) = \sigma\sqrt{v_j}$, where v_j is the j-th element of the diagonal of $(\mathbf{X}^T\mathbf{X})^{-1}$. If

[20] In fact, the least squares estimator can be shown to have the smallest variance among all unbiased estimates of β.

σ is unknown, one may estimate the standard error $\mathsf{se}(\hat{\beta}_j)$ by $\hat{\mathsf{se}}(\hat{\beta}_j) = \hat{\sigma}\sqrt{v_j}$, where $\hat{\sigma}^2 = \mathsf{RSS}(\hat{\beta})/(n-p)$ is an unbiased estimate of σ^2.

Under fairly general conditions, for moderately large n, the sampling distribution of $\hat{\beta}$ is approximately that of a multivariate Gaussian[21] with mean vector β and covariance matrix $\sigma^2(\mathbf{X}^T\mathbf{X})^{-1}$. This approximation is exact in the case that the errors ε_i in (2.30) themselves have (univariate) Gaussian distributions, conditional on their $\mathbf{x}_i$'s. In either case, the availability of a sampling distribution allows us to accompany our point estimate $\hat{\beta}$ with confidence intervals and hypothesis tests, if desired, in a manner similar to that we saw for making inferences on a mean through the sample mean $\bar{x}$. For example, a $100(1-\alpha)\%$ confidence interval for β_j is given by

$$(\hat{\beta}_j - z_{\alpha/2}\,\hat{\mathsf{se}}(\hat{\beta}_j)\,,\,\hat{\beta}_j + z_{\alpha/2}\,\hat{\mathsf{se}}(\hat{\beta}_j)) \ . \tag{2.32}$$

Similarly, a test of $H_0 : \beta_j = \beta_j^{(0)}$ versus $H_1 : \beta_j \neq \beta_j^{(0)}$, at significance level α may be performed by comparing the test statistic

$$T_j = \frac{\hat{\beta}_j - \beta_j^{(0)}}{\hat{se}(\hat{\beta}_j)} \tag{2.33}$$

to the value $z_{\alpha/2}$.

When prediction is of interest, we may obtain a prediction of an unobserved value $Y_* = y_*$ from a given $\mathbf{x}_*$ in a straightforward manner using a plug-in approach. Specifically, we predict y_* with $\hat{y}_* = \mathbf{x}_*^T\hat{\beta}$. One may also define a notion of standard error for this prediction, and predictive intervals. The calculations are similar, but not identical to, those underlying the standard error and interval estimates for a parameter. They differ in that, to describe the variability in predicting the random variable $y_* = \mathbf{x}_*^T\beta + \varepsilon_*$, one must account for both the variability in estimating β and the variability due to not knowing ε_*. See Wasserman [392, Ch. 13.4] for details.

Least squares estimation, while generally motivated from a linear algebraic or geometric perspective, can also be arrived at through application of the maximum likelihood method. Specifically, assume that the conditional distributions $F_{\varepsilon_i|\mathbf{X}_i}$ of the ε_i, given $\mathbf{X}_i = \mathbf{x}_i$, are independent Gaussians, with mean zero and variance σ^2. Due to (2.30), this implies that the conditional distributions $F_{Y_i|\mathbf{X}_i}$ of the random variables Y_i, given their respective $\mathbf{X}_i = \mathbf{x}_i$, are also independent Gaussians, but with mean $\mathbf{x}_i^T\beta$ and variance σ^2. That is, of the form

$$f_{Y_i|\mathbf{X}_i}(y|\mathbf{x}) = (2\pi\sigma^2)^{-1/2}\exp\left\{-(y-\mathbf{x}^T\beta)^2/(2\sigma^2)\right\} \ . \tag{2.34}$$

[21] A $p\times 1$ random vector $(X_1,\ldots,X_p)^T$ is said to have a multivariate Gaussian, density, with $p\times 1$ mean vector and $p\times p$ covariance matrix Σ if

$$f(\mathbf{x}) = ((2\pi)^p|\Sigma|)^{-1/2}\exp\left\{-(1/2)(\mathbf{x}-\ )^T\Sigma^{-1}(\mathbf{x}-\ )\right\} \ .$$

In principle, application of the maximum likelihood method here requires us to maximize the joint density[22] $f_{\mathbf{Y},\mathbf{X}}$ as a function of the unknown parameters (β,σ). However, this density may be written as the product of the conditional density $f_{\mathbf{Y}|\mathbf{X}}$ and the marginal density $f_{\mathbf{X}}$, and only the conditional density is a function of these parameters. So it is sufficient to consider only the conditional density, as a *conditional likelihood*,

$$\mathscr{L}(\beta,\sigma) = \prod_{i=1}^{n} f_{Y_i|\mathbf{X}_i}(y_i|\mathbf{x}_i) \ . \tag{2.35}$$

Writing out (2.35), using (2.34), and taking logarithms, we find that the conditional log-likelihood of (β,σ) is given by

$$\ell(\beta,\sigma) = -\frac{n}{2}\log(2\pi) - n\log(\sigma) - \frac{1}{2\sigma^2}\sum_{i=1}^{n}(y_i - \mathbf{x}_i^T\beta)^2 \ . \tag{2.36}$$

Maximizing (2.36) in β involves only the last term, and can be seen to be equivalent to minimizing the quantity $\mathsf{RSS}(\beta)$, thus resulting in the least squares estimate $\hat{\beta}$ in (2.31). Additionally, maximizing (2.36) in σ^2 can be shown to yield the estimate $\mathsf{RSS}(\hat{\beta})/n$, which for large n is nearly the same as the unbiased estimate $\mathsf{RSS}(\hat{\beta})/(n-p)$.

This likelihood-based perspective on linear regression is also useful in facilitating a Bayesian approach to the problem. Suppose, for example, that in addition to our Gaussian assumption on the conditional distribution of the errors ε_i, we also model the elements β_j of β as independent Gaussian random variables with common mean zero and variance τ^2. Then it may be shown that the posterior distribution $F_{\beta|\mathbf{Y},\mathbf{X}}$ of β, given the data, is multivariate Gaussian with a log-density of the form

$$\log f(\beta|\mathbf{Y}=\mathbf{y},\mathbf{X}) \quad \sim \quad -\sum_{i=1}^{n}(y_i - \mathbf{x}_i^T\beta)^2 - \lambda\sum_{j=1}^{p}\beta_j^2 \ , \tag{2.37}$$

where $\lambda = \sigma^2/\tau^2$, and the symbol '$\sim$' indicates here that terms not involving β have been omitted.

The expression in (2.37) is maximized, as a function of β, by the value

$$\hat{\beta}^{ridge} = \left(\mathbf{X}^T\mathbf{X} + \lambda\mathbf{I}\right)^{-1}\mathbf{X}^T\mathbf{y} \ . \tag{2.38}$$

The value in (2.38) is the mode of the posterior and, since the posterior is Gaussian, the mean. It has been proposed from a non-Bayesian perspective under the name of *ridge regression,* where it is called a *ridge estimator.* Note that in the event that $\lambda = 0$, we simply recover the least squares (and maximum likelihood) estimator $\hat{\beta}$ in (2.31). This corresponds to the case where $\tau^2 \to \infty$, indicating a lack of any real information from the prior.

[22] Here as a sub-script, and similarly below, in a slight abuse of notation, $\mathbf{X}$ refers to the random vector $(X_1,\ldots,X_p)^T$.

For $\lambda > 0$, the estimate $\hat{\beta}^{ridge}$ will differ from $\hat{\beta}$. In particular, it will be biased, with the bias increasing as λ increases. On the other hand, note for example that the ridge estimator is well-defined even when $\mathbf{X}$ is not of full rank, unlike the estimator in (2.31). In exchange for this bias, there is the potential to decrease the variance below that of the least squares estimator. This is often possible, for example, in cases where $\mathbf{X}$ is nearly rank-deficient, since then the variances in $\mathbb{V}(\hat{\beta})$ can be quite large, due to instability in the inversion of the matrix $\mathbf{X}^T\mathbf{X}$. If the trade off between bias and variance is managed fruitfully, the result can be an estimator with an overall reduction in mean-square error (i.e., with $\mathsf{MSE}(\hat{\beta}^{ridge}) < \mathsf{MSE}(\hat{\beta})$). The nature of the trade off depends on the value of λ, which may be set either subjectively (in a purely Bayesian manner) or through data-adaptive methods.

More generally, the optimization of (2.37) to produce (2.38) can be viewed as just one instance of a larger class of procedures falling under the name of *complexity penalized least squares*, where estimators are defined through

$$\hat{\beta}^J = \arg\max_{\beta} \quad \mathsf{RSS}(\beta) + \lambda J(\beta) \ . \tag{2.39}$$

That is, we estimate β by that value which optimizes a penalized version of the least squares criterion $\mathsf{RSS}(\beta)$, where $J(\cdot)$ is an arbitrary penalty function and $\lambda > 0$ is a tuning parameter. Adjusting the magnitude of λ controls the balance between the terms $\mathsf{RSS}(\beta)$ and $J(\beta)$ in selecting $\hat{\beta}$.

There are a variety of choices for J. Like ridge regression, they effectively introduce bias into the estimation of β, in exchange for a reduction in variance. However, unlike ridge regression, some choices of J are designed to actively eliminate elements of the vectors $\mathbf{x}_i$ from the model, through assignment of the values $\hat{\beta}^J_j = 0$ to ‘unimportant’ variables X_j. That is, such methods perform *model selection* in addition to estimation. These choices are particularly useful in settings where p is large, compared to n. Note too that, while some choices of J allow for analytic solutions to $\hat{\beta}^J$, such as the choice $J(\beta) = \beta^T\beta$ underlying ridge regression, in most cases it is necessary to solve (2.39) using numerical optimization software.

The topic of linear regression modeling has been developed in extraordinary detail and is still a mainstay of statistical modeling. On the other hand, there are entire classes of problems for which linear regression models can be at least unsatisfactory, if not just plain inappropriate. Such problems typically involve nonlinearity in the variables X, and non-Gaussianity, or even non-additivity, of the noise variables ε. Nevertheless, various aspects of linear regression models are often present to a nontrivial extent in the regression models that have been developed to deal with issues like these, which can be useful for understanding and interpreting such models. For example, generalized linear models essentially extend the concept of maximum likelihood estimation in linear models to other likelihoods, which includes the class of logistic regression models used in classification of binary response variables Y. Similarly, many techniques in the area of kernel-based modeling, including those methods based on support vector machines, arguably consist of mapping original variables X to a new, larger space, applying established linear statistical methods, and then representing the result of those methods in the space of the original variables. On the other hand, the extent to which the inferential machinery for such other model classes is developed can often lag behind that of the class of linear models, sometimes decidedly so. Methods of estimation and prediction generally accompany the initial proposal of a model class, but practical methods for quantifying the uncertainty in such inferences, for example in the form of confidence and prediction intervals, may be lacking, due to analytical or computational hurdles, or both.

2.3 Statistical Analysis of Network Data: *Prelude*

The unique relational nature of network data means that one often can encounter certain challenges in statistical analysis that are not present in 'standard' statistics problems, like those covered in introductory statistics courses. Two major sources of these challenges are (i) the dependency of the quantities involved and (ii) their dimensionality. The effects of these two are often further compounded by the sheer volume of data obtained in many network analysis studies. We close this chapter with a set of three short examples illustrating our point, each of which is a vignette of a larger set of topics to be covered later in the book.

Example 2.1 (Inference of Gene Association Networks). Recall from Section 1.2.3 that the study of patterns of regulatory association among genes in organisms is a topic of fundamental interest to scientists in genetics. At an elementary level, such associations may be thought of as corresponding to the extent to which one gene plays a role in 'activating' or 'suppressing' the activity of another gene (indirectly through the actions of the protein product(s) produced in accordance with its genetic code), across all pairs of genes. Current technology allows us to obtain, fairly cheaply and efficiently, an indication of the relative level of activity of genes (i.e., their level of 'expression'), through the use of DNA micro-array chips. With micro-arrays, one can simultaneously measure the expression levels of all genes in an organism, under various experimental circumstances. If a pair of genes has expression levels that are sufficiently similar across experiments, that may be suggestive of a regulatory association between the two. However, just what criteria should be used to judge similarity?

Formally, we may suppose we have a set of N_v genes $V = \{1,\ldots,N_v\}$, and for each gene $i \in V$, we measure a vector of expression levels $\mathbf{x}_i = (x_{i1},\ldots,x_{in})^T$ under n separate experimental conditions. A simple but commonly used measure of similarity between a pair of genes $i, j \in V$ is the correlation, $\hat{\rho}_{ij}$, between the vectors $\mathbf{x}_i$ and $\mathbf{x}_j$. Genes for which $\hat{\rho}_{ij}$ is judged to be 'large' (either positively or negatively) are judged to be associated with each other, and this is indicated by assigning the pair $\{i, j\}$ to an edge set E. The resulting graph $G = (V,E)$ then represents an inferred gene regulatory network.

In order to judge when $|\hat{\rho}_{ij}|$ is 'large', it is compared to a threshold t. In so doing, one is implicitly conducting a test of the hypothesis $H_0 : \rho_{ij} = 0$ against the alternative $H_1 : \rho_{ij} \neq 0$, where ρ_{ij} can be interpreted as the correlation of expression levels between i and j across the universe of relevant experimental conditions. Such a test in fact does not differ much in principle from the test for the value of a mean based on the Wald statistic in (2.23). However, we are not conducting just one test here, but rather approximately N_v^2 tests simultaneously! In other words, we find ourselves faced with a multiple testing problem, with the number of tests scaling with the dimensionality of the network. In addition, note too that there is a built-in dependency among our test statistics, by virtue of correlating the vector $\mathbf{x}_i$ of expression levels for each gene with that of every other gene.

As we will see in Chapter 7, the importance of the multiple testing problem here is in the critical impact it has on the threshold t used and, more fundamentally, on the manner in which one seeks to control errors in inferring the presence or absence of edges in the underlying network graph. Methods based on FDR and similar principles are one approach to this problem. □

Example 2.2 (Modeling Social Ties). Sociologists have long been interested in understanding what factors influence the development of social ties among actors. Social theory posits many such potential factors. A central tenet of social network analysis is that it is possible to assess the relevance of a given factor(s) on the social dynamics of a population of interest through the statistical modeling and analysis of quantitative measurements of the underlying social network. A well-known example is *homophily* – the tendency for entities to associate with others like them, popularized in the common parlance through the phrase, "Birds of a feather flock together." The presence of homophily has been confirmed in numerous social network contexts, often along the lines of the types of characteristics one might expect, such as age, gender, and social class or role.

A common technique for assessing the presence of social factors like homophily in a social network is to construct network regression models, with social ties as response variables Y and measurements relevant to the social factors as predictors X. More precisely, suppose that $G = (V, E)$ is a graph corresponding to an observed social network among individuals $i \in V$, with a social tie (e.g., friendship) between individuals $i, j \in V$ indicated by an edge $\{i, j\} \in E$. Let $Y_{ij} = Y_{ji} = 1$ if $\{i, j\} \in E$, and zero if not. That is, $\mathbf{Y} = [Y_{ij}]$ is just the adjacency matrix for G, treated as a random matrix. A regression model for the Y_{ij}'s might then specify a form for $\mathbb{P}(Y_{ij} = 1 | \mathbf{X}_i = \mathbf{x}_i, \mathbf{X}_j = \mathbf{x}_j)$, where $\mathbf{X}_i$ contains measurements on the i-th individual, such as age or gender.

We note, however, that often social theory hypothesizes factors that can be interpreted directly in terms of the presence or absence of a social tie as a function of other social ties. Homophily, for example, implies that 'friends of my friends are my friends'. Therefore, it can be expected that there is information on the event $\{Y_{ij} = 1\}$ in the values of the other random variables

$$\mathbf{Y}_{(-ij)} = \{Y_{kl} : \{k, l\} \neq \{i, j\}\} \ . \tag{2.40}$$

Thus, in building a social network regression model one may want to also include the variables $\mathbf{Y}_{(-ij)}$ as predictors for the Y_{ij}, and specify models for probabilities of the form $\mathbb{P}(Y_{ij} = 1|\mathbf{Y}_{(-ij)} = \mathbf{y}_{(-ij)}, \mathbf{X}_i = \mathbf{x}_i, \mathbf{X}_j = \mathbf{x}_j)$.

Regression models of this form fall under the class of *auto-regression* models, in that – unlike standard regression models – the response variable Y also plays a role as a predictor variable. This type of model, analogous to those encountered in time series analysis and the analysis of spatial data, explicitly seeks to capture the type of dependency often inherent in relational data. In addition, such models will potentially have a high number p of parameters, unless simplifications are made (e.g., assumptions of homogeneity of effect sizes across pairs $\{i, j\}$). We will examine models like these and others in more detail in Chapters 6 and 8. □

Example 2.3 (Estimation of an Internet 'Traffic Matrix'). It was mentioned in Section 1.2.1 that Internet service providers (ISPs) have a vested interest in monitoring the traffic on the Internet networks they maintain. One fundamental quantity that providers wish to monitor is the volume of data sent from each origin node in the network to each destination node. An example would be the number of bytes flowing between an origin and a destination in, say, a five-minute interval. If one organizes such traffic flows in a matrix form, say with origins for rows and destinations for columns, the result is what the Internet measurement community calls a *traffic matrix.*

If we represent a given network by a graph $G = (V, E)$, and assume for simplicity that all vertices can serve as both origins and destinations, then we see that it is necessary to monitor N_v^2 flows to obtain the traffic matrix of this network. It has traditionally been viewed as prohibitive to deploy throughout the network the technology infrastructure (i.e., measurement and communication devices) necessary to measure these flows directly. Easier to measure is the traffic flowing over the N_e links in the network. But such measurements are less useful for monitoring, as the flow over each link essentially consists of a super-position of traffic flows for all origin-destination pairs utilizing that link.

Interestingly, this observation can actually be exploited in a useful manner. Let $\mathbf{x}$ be a $N_v^2 \times 1$ vector of traffic flows observed in a given time period for all pairs of origins and destinations in G. Similarly, let $\mathbf{y}$ be a $N_e \times 1$ vector of the flows observed over all links during the same time period. Additionally, let $\mathbf{B}$ be a $N_e \times N_v^2$ binary matrix, with the element in the e-th row and ij-th column taking the value $B_{e;ij} = 1$ if edge e is traversed in going from the i-th origin to the j-th destination, and 0 if not. This is called the *routing matrix,* as it describes the routing of traffic over G. The super-position of origin-destination flows over links suggests the relation $\mathbb{E}(\mathbf{Y}|\mathbf{X} = \mathbf{x}) = \mathbf{Bx}$. This in turn seems to indicate that one might predict $\mathbf{x}$ from the measurements $\mathbf{Y} = \mathbf{y}$ by setting up a linear regression model $\mathbf{y} = \mathbf{Bx} + \varepsilon$, and solving for $\mathbf{X}$.

Unfortunately, it is easy to see that this problem will typically be ill-posed, since there are N_v^2 values to be learned from only N_e measurements, and generally N_e is substantially smaller than N_v^2. In other words, we have what looks like a standard regression problem, but with $p \gg n$, in which case there is no unique least squares solution. A good deal of success has been had in solving this problem, however, when the least squares problem is replaced by an appropriately specified complexity penalized least squares problem, a problem of the form

$$\hat{\mathbf{x}} = \arg\max_{\mathbf{x}} \left[(\mathbf{y} - \mathbf{B}\mathbf{x})^T (\mathbf{y} - \mathbf{B}\mathbf{x}) + \lambda\, J(\mathbf{x}) \right] \;, \tag{2.41}$$

for some penalty $J(\cdot)$. We will see additional details on this and related problems in Chapter 9. □

2.4 Additional Related Topics and Reading

A more thorough introduction to the topic of graph theory may be found in any of a number of introductory textbooks, such as those by Bollobás [42], Diestel [118], or Gross and Yellen [186]. Details on graph data structures and algorithms are in many computer science algorithms texts. See the text by Cormen, Leiserson, Rivest, and Stein [100], for example, from which most of the material of Section 2.1.4 was taken. For statistics and probability background, the text by Wasserman [392] is useful for its combination of coverage and conciseness, and we have drawn heavily on it for the presentation of material in Sections 2.2.1 and 2.2.2. In addition, Wasserman cites many other standard texts on specific topics. A more careful and complete exposition on the topic of probability alone, including Markov chains, may be found in standard introductory texts on the subject, such as that by Ross [333]. For more complete developments of the methods on simulation of random variables we described, and other related methods, see the texts by Liu [264] or Robert and Casella [325].

Exercises

2.1. For the graph in Figure 2.4, complete the following.

a. Determine a walk from vertex 2 to vertex 7 that is not a trail.

b. Determine a trail from vertex 2 to vertex 7 that is not a path.

c. Determine a path from vertex 2 to vertex 7.

d. How many paths are there from vertex 1 to vertex 7? What is the geodesic distance between these two vertices? Is there a unique shortest path between them?

e. How many connected subgraphs of order four can be found? How many of these subgraphs contain a cycle of length at least three? Of length four?

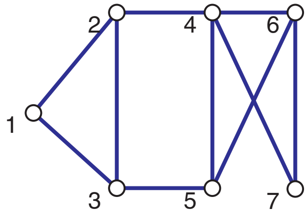

Fig. 2.4 Graph corresponding to Exercise 2.1.

2.2. For a network software package of your choice,[23] familiarize yourself with the basic aspects of its data structures and algorithms. For example, using the graph in Figure 2.4, create an appropriate input file(s), according to the format(s) available (e.g., adjacency, lists, etc.). Additionally, using the same graph as input, explore some of the basic algorithmic capabilities of your software by, for instance, calculating all shortest paths among all pairs of vertices, a minimal spanning tree from a given vertex, etc.

2.3. Recall the definition of the Laplacian $\mathbf{L} = \mathbf{D} - \mathbf{A}$ of a graph G, in Section 2.1.3, where $\mathbf{A}$ is the adjacency matrix and $\mathbf{D} = \mathrm{diag}[(d_i)_{i \in V}]$ is a diagonal matrix defined in terms of the degree sequence $\{d_i\}_{i \in V}$. An alternative form of the Laplacian – typically called the *normalized Laplacian* – is the matrix defined as

$$\tilde{\mathbf{L}} = \mathbf{D}^{-1/2}\mathbf{L}\mathbf{D}^{-1/2} = \mathbf{I} - \mathbf{D}^{-1/2}\mathbf{A}\mathbf{D}^{-1/2} \quad .$$

a. Both types of graph Laplacian can be shown to have non-negative eigenvalues, and minimal eigenvalue equal to zero, with corresponding eigenvector $\mathbf{1}$. In addition, the eigenvalues of $\tilde{\mathbf{L}}$ can be shown to all fall in the interval $[0,2]$. For the graph shown in Figure 2.4, compute $\mathbf{L}$ and $\tilde{\mathbf{L}}$ and verify numerically that these properties hold.

b. For a connected graph G, the second smallest eigenvalue of $\mathbf{L}$ and $\tilde{\mathbf{L}}$ will be non-zero, and its corresponding eigenvector is often found to be useful for partitioning G into two cohesive subgraphs.[24] A simple approach is to annotate each vertex in G with the sign of the corresponding entry in this eigenvector. Do this for the graph shown in Figure 2.4. Is the graph partitioned in a sensible manner?

[23] For an (incomplete) list of relevant network-related software tools and packages, consult the website for this book at *http://math.bu.edu/people/kolaczyk/SAND.html*.

[24] We will examine this usage of the Laplacian more formally later, in Chapter 4.3.3.

c. Try modifying the graph in Figure 2.4, by adding and deleting edges and vertices, and repeating parts (a) and (b). How do the eigenvalues and the eigenvector corresponding to the first non-zero eigenvalue behave under these changes? What happens if your graph becomes disconnected?

2.4. Suppose that $\mathbf{X} = (X_1, \ldots, X_n)^T$ is an n-length random vector, with expectation $\mathbb{E}(\mathbf{X}) =$ and covariance matrix $\mathbb{V}(\mathbf{X}) = \Sigma$. Let $\bar{X} = \sum_{i=1}^{n} X_i/n$ be the average of the elements of **X**. For this problem, you may wish to use the expressions in (2.7).

a. If the X_i all have a common mean, say $_i = \alpha$, show that $\mathbb{E}(\bar{X}) = \alpha$.

b. Show that the variance of $\bar{X}$ is given by the expression

$$\mathbb{V}(\bar{X}) = n^{-2} \left[\sum_{i=1}^{n} \Sigma_{ii} + 2 \sum_{i<j} \Sigma_{ij} \right] .$$

c. In the case where the elements of **X** are uncorrelated and have common variance, and hence $\Sigma = \mathsf{diag}[(\sigma^2, \ldots, \sigma^2)]$, for some $\sigma^2 > 0$, show that the expression in part (b) for the variance of $\bar{X}$ reduces to σ^2/n.

2.5. Recall the ridge regression estimator

$$\hat{\beta}^{ridge} = \left(\mathbf{X}^T\mathbf{X} + \lambda\mathbf{I}\right)^{-1} \mathbf{X}^T\mathbf{Y}$$

in equation (2.38).

a. Show that $\hat{\beta}^{ridge}$ can be obtained by maximizing (2.37) as a function of β, which is equivalent to solving

$$\frac{\partial}{\partial \beta} \log f(\beta | \mathbf{Y} = \mathbf{y}, \mathbf{X}) = 0 \ .$$

b. The data in Table 2.1 were generated according to the model

$$y_i = 1 + 1 \times x_{i1} + 1 \times x_{i2} + 0 \times x_{i3} + \varepsilon_i \ ,$$

for $i = 1, \ldots, 10$, where the x_{i1} are independent draws from a Gaussian distribution, with mean zero and unit variance, the x_{i2} are generated in the same manner, and independently of the x_{i1}, and the x_{i3} are formed by adding a small amount of noise to the averages $(x_{i1} + x_{i2})/2$. The errors ε_i are independent, among themselves and of the x variables, and follow a Gaussian distribution with mean zero and variance $(0.01)^2$.

Fit a linear regression to these data for Y as a function of X_1 and X_2 only. Then fit a linear regression for the full model (i.e., with X_3 included as well). The estimates you obtain for the coefficients of X_1 and X_2 and their corresponding standard errors should be quite different in the two cases, due to the fact that when X_3

Y	X_1	X_2	X_3
2.292	1.209	-0.472	0.3588
2.068	-1.577	2.291	0.3377
0.926	-0.605	1.231	0.3030
-1.419	-0.689	-1.047	-0.8682
4.451	0.814	0.585	0.7039
0.679	-1.681	0.617	-0.5338
3.226	0.304	1.441	0.8917
2.220	-0.202	0.360	0.0736
-0.649	-1.510	-1.262	-1.3820
-2.008	-1.081	-0.323	-0.7049

Table 2.1 Data for Exercise 2.5.

is in the model the three predictor variables are nearly linearly dependent, and therefore $\mathbf{X}^T\mathbf{X}$ is nearly singular.

The effect of the term $\lambda\mathbf{I}$ in $\hat{\beta}^{ridge}$ is to stabilize the operation of matrix inversion required by the least squares solution. Calculate the ridge regression estimates $\hat{\beta}^{ridge}$, in the model with X_1, X_2, and X_3, for various choices of λ, and compare your estimates to those obtained by least squares. How close can you get your estimates to the true value $\beta = (1,1,1,0)^T$ as you change λ ?