2016

The Geometry of Data: Distance on Data Manifolds

Casey Chu
Harvey Mudd College

The Geometry of Data: Distance on Data Manifolds

Casey Chu

Weiqing Gu, Advisor

Dagan Karp, Reader

HARVEY
MUDD
COLLEGE

Department of Mathematics

May, 2016

Abstract

The increasing importance of data in the modern world has created a need for new mathematical techniques to analyze this data. We explore and develop the use of geometry—specifically differential geometry—as a means for such analysis, in two parts. First, we provide a general framework to discover patterns contained in time series data using a geometric framework of assigning distance, clustering, and then forecasting. Second, we attempt to define a Riemannian metric on the space containing the data in order to introduce a notion of distance intrinsic to the data, providing a novel way to probe the data for insight.

Acknowledgments

I would like to thank Professor Gu for her constant support and visionary insight throughout writing this thesis, as well as Harvey Mudd College's incredible faculty and academic program. The breadth and quality of courses here have given me many valuable perspectives from which to see the world, and every course I have taken here has inspired some part of this thesis.

Preface

In this preface, I outline the structure of this senior thesis, which is organized into two parts.

In Part I, I develop a methodology to detect patterns in time series data. Although likely employed before, this methodology has not been laid out as explicitly as or in the generality that this document does. It consists of three components. In Chapter 2, I break the time series into segments of a given duration and define a **distance** metric on this space of segments that quantifies how similar two segments are. A group of segments sufficiently close together under this metric constitutes a pattern; standard **clustering** algorithms such as k-means clustering, or BUBBLE clustering may be used to find these groups, as explored in Chapter 3. Finally, I use these clusters to **forecast** the time series, as in Chapter 4. I present the results of this approach applied to stock price data in Chapter 5.

Inspired by this problem, I turn to the geometry of data in Part II. The methodology above rests crucially on how distance is defined on the space of data, raising the following question: is there a general way of assigning distance to a space of data, without domain-specific knowledge? In Chapter 7, I attempt to define a **Riemannian metric** on the space of data, in such an intrinsic way. In Chapter 8, I review **information geometry**, and in Chapter 9, I attempt to use it to gain a new perspective on the problem, developing a **duality** between statistical manifolds and the space of data.

Many avenues for further research are left open in these later chapters; these open questions are outlined in Chapter 10. My hope is that the results in this thesis pave the way for further research in this topic. I invite the reader to take a look at and think about these questions, questions that are becoming increasingly relevant in today's world of big data.

Contents

Part I

Geometric discovery of patterns in time series

Chapter 1

Introduction: patterns in time series

This first part will attempt to solve the following easy-to-state problem:

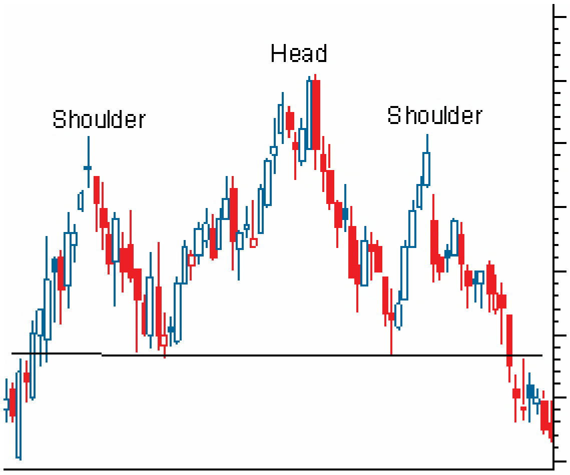
Key Question 1. *Given one or many time series, how do we extract patterns that occur frequently in the data? Furthermore, can we use these patterns to forecast the behavior of a time series into the immediate future?*

In the following chapters, we seek to develop a methodology to solve this problem.

1.1 Examples of applications

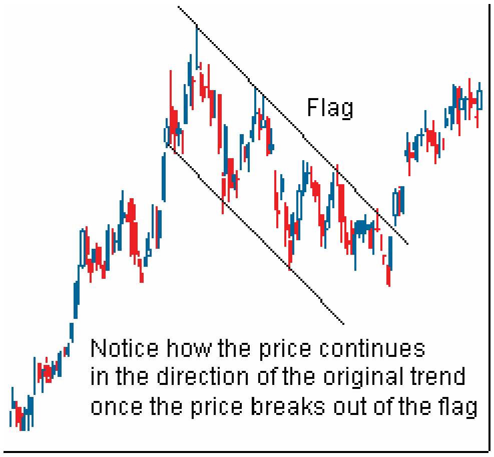
One immediate application of an algorithm that extracts patterns is to stock data. Suppose we have the price and volume data for the past five years of 100 stocks. There are patterns that occur frequently in stock data that supposedly indicate the trajectory of the stock. The so-called head-and-shoulders pattern is one example, depicted in Figure 1.1. Its formation is said to indicate that the price will fall in the near future if it breaks the so-called shoulder line, and similarly, an inverse head-and-shoulders pattern is said to indicate that the price will rise under the same condition. In theory, the head-and-shoulders patterns are driven by many factors, including the psychology and traders, news, or the economic climate.

There are other known patterns in stock data, such as the flag pattern, as shown in Figure 1.2. Are there other patterns that have not been recognized



Source: Chart by MetaStock

Figure 1.1* A head-and-shoulders pattern in stock price data.



Notice how the price continues in the direction of the original trend once the price breaks out of the flag

Source: Chart by MetaStock

Figure 1.2* A flag pattern in stock price data.

but are strong predictors of future stock price? An algorithm that extracts such patterns from time series would prove useful in such an analysis.

Another application has to do with shopping data. Imagine that we are a supermarket, and we have access to a list of transactions by customer. For example, we know that Customer A purchased milk and eggs on Tuesday, followed by bread and cheese on Thursday. Such a record can be interpreted as a time series of purchases, and an algorithm that extracts patterns may be able to provide crucial insights, such as whether, for example, a purchase of eggs is likely to be followed by a purchase of cheese.

1.2 The setup

Let us first define what we mean by a time series. Informally, a time series is a sequence of observations from some set Σ made at some times T. Thus, formally, we write the following definition.

Definition 1 (Time series)**.** *A **time series** is a function $q : T \to \Sigma$, where T is a finite index set and Σ is an arbitrary set.*

We will take T to be finite subsets of $\mathbb{R}$, and we will use the convention that $q(0)$ corresponds to the value of the time series at the present time, meaning that $q(t)$ for $t < 0$ denotes values in the past. Also, whenever we enumerate values of $T = \{t_1, \ldots, t_M\}$, we will assume that $t_1 < \cdots < t_M$. We will call each t_i a *sample time*.

Let us define two ways to take smaller time series from an existing time series q. The first is the *prefix time series*, which restricts the domain of q to the first i sample times.

Definition 2 (Prefix time series)**.** *Let $q : \{t_1, \ldots, t_N\} \to \mathbb{R}^n$ be a time series. The **prefix time series** $q_{1:i} : \{t_1, \ldots, t_i\} \to \mathbb{R}^n$ of q is another time series, defined for $1 \leq i \leq N$, is defined by $q_{1:i}(t) = q(t)$.*

Notice that the prefix time series $q_{1:N}$ is simply q itself.

The second is the *history* at time t of a time series q, which restricts the domain of q but also shifts it so that time t is the present.

Definition 3 (History of a time series)**.** *The **history** at time t of a time series $q : T \to \mathbb{R}^n$ is another time series $q_t : T' \to \mathbb{R}^n$, such that $q_t(\tau) = q(t + \tau)$ for all $\tau \in T'$, where $T' = (T - t) \cap \{t \in \mathbb{R} \mid t \leq 0\}$.*

Notice that the history of q at time $t = 0$ is simply q itself. It is also convenient to collect all histories into one object. Thus, we define an *orbit* as follows.

Definition 4 (Orbit of a time series)**.** *The **orbit** of a time series $q : T \to \mathbb{R}^n$ is the set of its histories*

$$Q = \{q_t \mid t \in T\}.$$

1.3 The problem statement

The problem that this part of the thesis aims to solve is then written more formally as follows. Suppose we are given a set of time series

$$\{q_1 : T_1 \to \Sigma, \dots, q_k : T_k \to \Sigma\}.$$

What are the most common patterns that occur in these time series? The concept of a "pattern" is intentionally vague—we will define it more formally in the following chapters. For now, let us see how our two examples fit into this framework.

In the stock price example, suppose we are given the data for k stocks, that is, information such as the price and volume of the stock for each day for the past, say, five years. In this case, each time series represents a stock, or more explicitly, the price, volume, etc. of the stock at a given time. The codomain Σ is $\mathbb{R}^n$, where n is the number of pieces of information given for each time.

In the shopping example, suppose we are given the transaction data for k customers. In this case, each time series represents a customer, or more explictly, what the customer purchased at a given time. The codomain Σ is the power set of the set of possible items; it contains sets of purchased items.

1.4 Related work

The problem of forecasting stock prices from historical data is obviously heavily studied. However, less attention has been placed on the unsupervised extraction of patterns from stock data and, more generally, time series data. Most similar to the approach taken in this thesis are the following. Choi and Chukkapalli (2009) apply the same broad framework that this thesis does to identify patterns in time series data, although their approach of using change points and an autoregressive model to define distance is not applicable to the generic problem. Guo et al. (2007) use a clustering approach as well, although they use a neural network known as a self-organizing map that is difficult to apply the probabilistic interpretation that is needed for prediction.

The shopping data problem is a canonical example problem in *sequential pattern mining*, which was introduced by Agrawal and Srikant (1995). Han et al. (2007) provide a review of common techniques in sequential pattern mining, including GSP by Srikant and Agrawal (1996), SPADE by Zaki (2001), and PrefixSpan by Pei et al. (2001). These methods are effective but take advantage of the specific power-set structure of the shopping data problem. On the other hand, we would like to handle this problem more generically, so that the same technique can be applied to any time series data, including stock price data.

1.5 Looking ahead

To perform pattern discovery in a time series, we must first quantify what it means for two time series to contain similar patterns. To do so, we will define a *distance* $d(p, q)$ between two time series p and q to give a non-negative real number that is low when the pattern contained in p is close to the pattern contained in q. Defining this function is the subject of Chapter 2.

Once this distance is defined, we can consider the orbit of one time series, the set of all of its histories. We want to group together the different histories in the orbit based on their distance, thus grouping based on patterns contained inside the original time series. This process is called *clustering* and is the subject of Chapter 3.

Once we have different time series grouped into different patterns, we would like to use these patterns to forecast the behavior of a time series in the immediate future. This is the subject of Chapter 4.

We shall see that each of these stages have interesting and varied math underlying them.

Chapter 2

Defining distance

In this chapter, let $p : T_1 \to \Sigma$ and $q : T_2 \to \Sigma$ be two time series. We want to define a distance function $d(p, q)$ that indicates whether p and q contain the same pattern (for some region of time close to the present). In analogy with physical distance, we would like $d(p, q)$ to be small when p and q do contain the same pattern, and $d(p, q)$ to be large when p and q do not.

Before we can quantify similarity between time series, we must quantify similarity between the values that the time series takes on, namely elements of Σ. Concretely, we would like a distance $\hat{d} : \Sigma \times \Sigma \to \mathbb{R}$ between elements of Σ. Once we do this, we will be able to aggregate this similarity between elements of Σ to create a measure of similarity on time series with codomain Σ.

This distance on Σ is domain-specific and will depend on the problem at hand. For the stock price problem, $\Sigma = \mathbb{R}^n$ comes with a natural choice of distance, namely Euclidean distance

$$\hat{d}(x, y) = ||x - y||,$$

although a better choice may be hand-picked. For the shopping problem, Σ is the power set of possible items, so we need a function that compares sets of items for similarity. One simple distance to define is

$$\hat{d}(X, Y) = \begin{cases} 1 & \text{if } X \cap Y = \varnothing \\ 0 & \text{otherwise} \end{cases}.$$

This measures whether two shopping baskets share at least one item in common.

From now on, we will assume the existence of a distance $\hat{d}$ on Σ, and proceed to define distance between two time series on Σ.

2.1 Euclidean distance

One simple approach to define distance between two time series on Σ is to again use Euclidean distance.

Since we only care about the recent history of p and q when looking for patterns, we will assume from now on that T_1 and T_2 are bounded below by $t = -w$ for some window width $w > 0$. Suppose for simplicity that $T_1 = T_2 = \{t_1, \dots, t_k\}$; if $\Sigma = \mathbb{R}^n$, then we can always use some kind of interpolation to ensure that $T_1 = T_2$. Then the *Euclidean distance* between p and q is defined as follows.

Definition 5 (Euclidean distance)**.** *The* ***Euclidean distance*** *between two time series $p : T \to \Sigma$ and $q : T \to \Sigma$ is*

$$d(p,q) = \sqrt{\sum_{t \in T} \hat{d}\big(p(t), q(t)\big)^2}.$$

This meets some of the requirements for a distance between time series. For example, if p and q are the same as each other, then $d(p,q) = 0$.

One problem with this approach in the case of $\Sigma = \mathbb{R}^n$ is if the two time series differ by vertical shift, namely that

$$p(t) \approx q(t) + c$$

for some constant $c \in \mathbb{R}^n$. Then the Euclidean distance is large, even though in our problem we consider vertically-shifted patterns to be equivalent. To solve this, instead of calculating the distance between p and q, we calculate the distance between the two series, shifted by their mean.

$$d'(p,q) = d(p - \bar{p}, q - \bar{q}),$$

where

$$\bar{p} = \frac{1}{k}\sum_{i=1}^{k} p(t_i) \qquad \bar{q} = \frac{1}{k}\sum_{i=1}^{k} q(t_i).$$

One more difficult problem to fix is a difference in pace. Consider Figure 2.1. Visually, the patterns are the same; however, the Euclidean distance is not small because the corresponding features of the two time series are not aligned at the same time. How can we define a distance that accounts for such a difference in pace?

2.2 Dynamic time warping

One way we can account for this difference in pace is using a technique called *dynamic time warping*. Dynamic time warping is a technique that assesses the similarity between two time series that potentially vary in speed. We follow the presentation of Müller (2007). Consider the time series $p : T_1 \to \Sigma$ and $q : T_2 \to \Sigma$, and assume that $|T_1| = M$ and $|T_2| = N$ with M not necessarily equal to N. Dynamic time warping attempts to align the two time series such that the first and last sample of p and q are aligned with each other, but the samples in between are allowed to be out of alignment, as in Figure 2.1.

We encapsulate this idea with the concept of an *alignment*.[1]

Definition 6 (Alignment)**.** *Let $p : T_1 \to \Sigma$ and $q : T_2 \to \Sigma$ be two time series, with $T_1 = \{t_1, \ldots, t_M\}$ and $T_2 = \{s_1, \ldots, s_N\}$. An* ***alignment*** *between p and q is a sequence of elements $(a_1, \ldots, a_\ell)$ with $a_i \in T_1 \times T_2$, satisfying the following two conditions*:

1. *Boundary conditions*: $a_1 = (t_1, s_1)$*, and* $a_\ell = (t_M, s_N)$.
2. *Step size condition*: *If* $a_i = (t_j, s_k)$*, then* $a_{i+1} \in \{(t_j, s_{k+1}), (t_{j+1}, s_k), (t_{j+1}, s_{j+1})\}$.

An alignment associates the times in T_1 with times in T_2. The boundary conditions ensure that every time in T_1 is paired with a time in T_2, and vice versa. The step size condition ensures that as we move forwards in time in T_1, we do not move backwards in time in T_2. We can visualize alignments by plotting T_1 on one axis and T_2 on the other axis, as in Figure 2.2. Then alignments are paths that run from the bottom-left to the top-right in steps of 1 sample time.

[1] This is a *warping path* in Müller (2007).

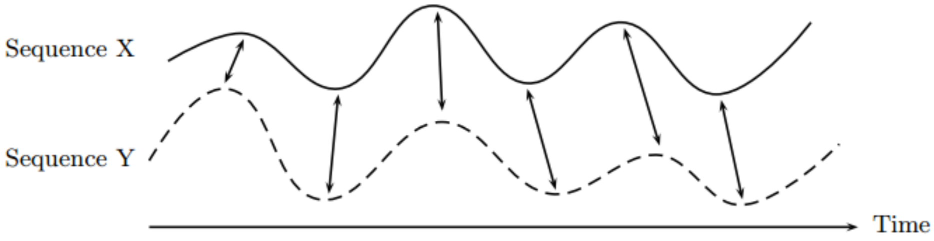


Figure 2.1* Two time series with similar features, although they vary in pace.

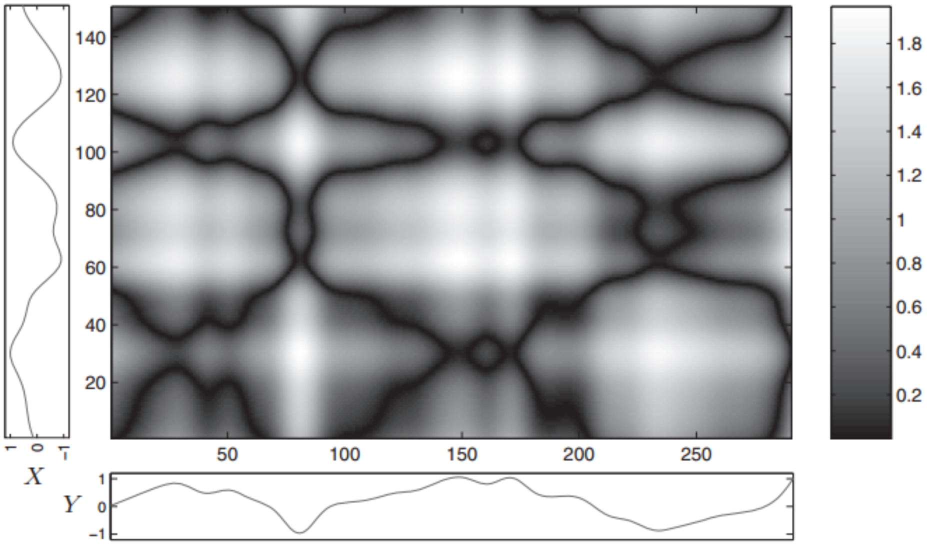


Figure 2.2* Two time series, one on each axis. $\hat{d}(p(t_i), q(s_j))$ is plotted as grayscale, with lower values corresponding to darker areas. Alignments are monotonic paths from the lower-left corner to the upper-right corner.

There are many possible such alignments. Intuitively, the "best" alignment is one where the features of p are aligned with the features of q. To find the best alignment, we minimize the following cost function over all possible alignments.

Definition 7 (Cost of an alignment)**.** *The **cost** of an alignment X is*

$$c(X) = \sum_{(t,s)\in X} \hat{d}\big(p(t), q(s)\big)\,.$$

This gives us the following definition for the DTW (dynamic time warping) distance:

Definition 8 (DTW distance)**.** *The **DTW distance** between two time series p and q is*

$$d_{DTW}(p,q) = \min_{X \text{ is an alignment of } p \text{ and } q} c(X).$$

This is a distance we can use to measure how similar two time series are. Note that it is not a metric, as it does not satisfy the triangle inequality, and

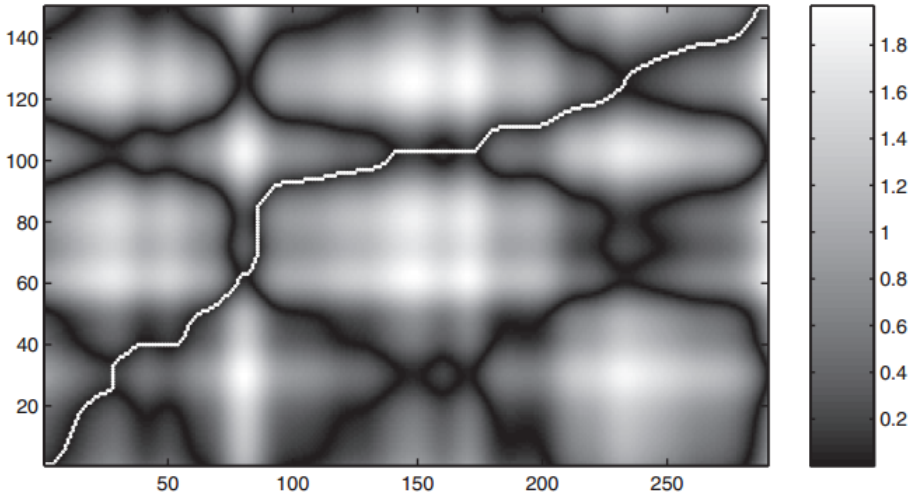


Figure 2.3* The optimal alignment (white) runs along a “valley” of dark.

it is not positive-definite, meaning that it is possible that $d(p,q) = 0$ with $p \neq q$.

One way to carry out the optimization problem of calculating the DTW is to enumerate all possible alignments and computing the minimum value of the cost function. However, even with a heuristic search algorithm like A* search, this is prohibitively expensive in the worst case with a computational complexity that is exponential in M and N. Luckily, there exists an efficient computation of the DTW distance in $O(MN)$ time using dynamic programming.

The idea is that the DTW distance between p and q is related to the DTW distance between their prefixes.

Definition 9 (Accumulated cost matrix)**.** *The **accumulated cost matrix** of two time series p and q is*

$$D_{i,j} = d_{DTW}(p_{1:i}, q_{1:j}),$$

for $1 \leq i \leq M$ *and* $1 \leq j \leq N$.

Note that under this definition, we can calculate $d_{DTW}(p,q) = D_{M,N}$. It turns out that D can be computed efficiently using the following recurrence relation.

Theorem 1. *The accumulated cost matrix D of two time series $p : \{t_1, \ldots, t_M\} \to \Sigma$ and $q : \{s_1, \ldots, s_N\} \to \Sigma$ satisfies the following recurrence relation:*

$$\begin{aligned} D_{i,1} &= \sum_{1 \le k \le i} \hat{d}\big(p(t_k), q(s_1)\big) \\ D_{1,j} &= \sum_{1 \le k \le j} \hat{d}\big(p(t_1), q(s_k)\big) \\ D_{i+1,j+1} &= \min(D_{i,j+1},\, D_{i+1,j},\, D_{i,j}) + \hat{d}\Big(p(t_{i+1}), q(s_{j+1})\Big)\,. \end{aligned}$$

Proof. First, we prove the expression for $D_{i,1} = d_{DTW}(p_{1:i}, q_{1:1})$. There is only one alignment between $p_{1:i} : \{t_1, \ldots, t_i\} \to \mathbb{R}^n$ and $q_{1:1} : \{s_1\} \to \mathbb{R}^n$, namely $X = \big((t_1, s_1), \ldots, (t_i, s_1)\big)$, so

$$D_{i,1} = d_{DTW}(p_{1:i}, q_{1:1}) = c(X) = \sum_{1 \le k \le i} \hat{d}\big(p(t_k), q(s_1)\big)\,.$$

The argument for the expression for $D_{1,j}$ is identical.

Next, we prove the expression for $D_{i+1,j+1}$. Let $X = (a_1, \ldots, a_\ell, a_{\ell+1})$ be an optimal alignment between $p_{1:i+1}$ and $q_{1:j+1}$. The boundary condition implies that $a_{\ell+1} = (p(t_{i+1}), q(s_{j+1}))$, so that the cost of X can be decomposed as

$$c(X) = c(X') + \hat{d}\Big(p(t_{i+1}), q(s_{j+1})\Big)\,,$$

for $X' = (a_1, \ldots, a_\ell)$. It is easy to check that X' is itself an alignment.

The step size condition implies that

$$a_\ell \in \{(t_i, s_{j+1}), (t_{i+1}, s_j), (t_i, s_j)\},$$

so that because $c(X)$ is optimal,

$$c(X') = \min(D_{i,j+1},\, D_{i+1,j},\, D_{i,j}).$$

It follows that

$$D_{i+1,j+1} = c(X) = \min(D_{i,j+1},\, D_{i+1,j},\, D_{i,j}) + \hat{d}\Big(p(t_{i+1}), q(s_{j+1})\Big)\,.$$

□

For ease of computation, we formally extend $D_{i,j}$ where i and j can now be 0.

Corollary 1. *The accumulated cost matrix D of two time series $p : \{t_1, \ldots, t_M\} \to \mathbb{R}^n$ and $q : \{s_1, \ldots, s_N\} \to \mathbb{R}^n$ satisfies the following recurrence relation:*

$$\begin{aligned} D_{0,0} &= 0 \\ D_{i,0} &= \infty \\ D_{0,j} &= \infty \\ D_{i+1,j+1} &= \min(D_{i,j+1}, D_{i+1,j}, D_{i,j}) + \hat{d}\Big(p(t_{i+1}), q(s_{j+1})\Big) \end{aligned}$$

for $1 \leq i < M$ and $1 \leq j < N$.

Proof. We can obtain the equations in Theorem 1 from these equations. □

This recurrence relation allows us to calculate the DTW distance $d_{DTW}(p, q)$ as $D_{M,N}$ efficiently in $O(MN)$ time by iteratively computing $D_{i,j}$ for each i in increasing order, for all j in increasing order.

This measure of distance is ideal for pattern discovery in time series, because unlike Euclidean distance, dynamic time warping allows for variance in the speed of the pattern, as well as the height of the pattern. Indeed, it is used for many time series applications already, such as speech recognition from an audio signal.

Chapter 3

Clustering on a metric space

Once we have a way of specifying the distance between two time series, we can *cluster* them into different groups, where ideally we'd like time series containing similar patterns to be in the same group. We can approach this problem in different ways. In this chapter, let $P = \{p_1, \ldots, p_N\}$ be N time series, with the distance between two time series given by a distance function $d(p, q)$. Recall that we've defined d such that $d(p, q)$ is close to 0 if p and q contain similar patterns, and $d(p, q) \gg 0$ if p and q do not. We would like to cluster P, meaning that we would like to partition P into k sets $O_1, \ldots, O_k$ that contain mutually close objects.

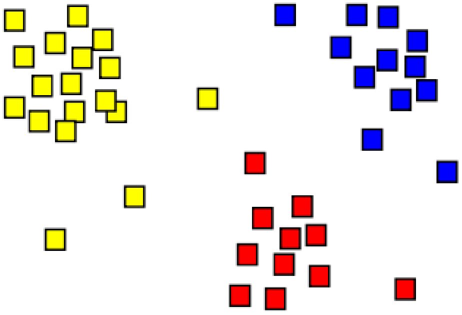


Figure 3.1* The clustering of points in $\mathbb{R}^2$ into 3 clusters.

3.1 Isometric embedding and clustering in $\mathbb{R}^n$

One approach to cluster is to note that there exist many clustering algorithms for points in $\mathbb{R}^n$, the most common being k-means clustering. Therefore, if we embed our time series into $\mathbb{R}^n$ as points, then we can take advantage of such algorithms. This process, *isometric embedding*, consists of taking a set of objects with known distances between them and finding corresponding points in $\mathbb{R}^n$ that have the same, or similar, distances.

3.1.1 Multidimensional scaling

Multidimensional scaling is one technique to isometrically embed a set of objects P into $\mathbb{R}^n$. Formally, we would like a mapping $f : P \to \mathbb{R}^n$ such that

$$||f(p) - f(q)|| \approx d(p, q).$$

for all $p, q \in P$. It is not always possible to find an embedding that makes this equality hold for a given n. Instead, we can formulate this problem as the following optimization problem

$$\underset{x_1,\ldots,x_N \in \mathbb{R}^n}{\text{minimize}} \quad \sum_{i<j} \Big(||x_i - x_j|| - d(p_i, p_j) \Big)^2 .$$

Multidimensional scaling uses the solution to this optimization problem to isometrically embed a set of objects. For completeness, an algorithm to compute the optimal coordinates is given in the following theorem.

Theorem 2. *Let $P = \{p_1, \ldots, p_N\}$ be a set with a distance $d(p, q)$ defined on it. Then a set of coordinates $\{x_1, \ldots, x_N\} \subseteq \mathbb{R}^n$ that solve the following optimization problem*

$$\underset{x_1,\ldots,x_N \in \mathbb{R}^n}{\text{minimize}} \quad \sum_{i<j} \Big(||x_i - x_j|| - d(p_i, p_j) \Big)^2$$

can be computed as follows.

Let A be a matrix with entries $A_{ij} = -\frac{1}{2} d(p_i, p_j)^2$, and let $B = HAH$, where $H = I - \frac{1}{N} ee^T$, where I is the identity matrix and e is a vector of 1s. *Let $\{v_1, \ldots, v_n\}$ be the eigenvectors of B corresponding to the n largest eigenvalues $\lambda_1 \geq \cdots \geq \lambda_n$ of B. Then $x_i = \sqrt{\lambda_i} v_i$.*

Cox and Cox (2000) provide the details of using this theorem. Figure 3.3 depicts an example of multidimensional scaling, used to place American cities into $\mathbb{R}^2$ from only a table of flight distances.

	Atlanta	Chicago	DC	Denver	Houston	LA	Miami	NYC	SF	Seattle
Atlanta		587	543	1212	701	1936	604	748	2139	2182
Chicago	587		597	920	940	1745	1188	713	1858	1737
DC	543	597		1494	1220	2300	923	205	2442	2329
Denver	1212	920	1494		879	831	1726	1631	949	1021
Houston	701	940	1220	879		1374	968	1420	1645	1891
LA	1936	1745	2300	831	1374		2339	2451	347	959
Miami	604	1188	923	1726	968	2339		1092	2594	2734
NYC	748	713	205	1631	1420	2451	1092		2571	2408
SF	2139	1858	2442	949	1645	346	2594	2571		678
Seattle	2182	1737	2329	1021	1891	959	2734	2408	678	

Table 3.2* A table of distances between American cities in miles.

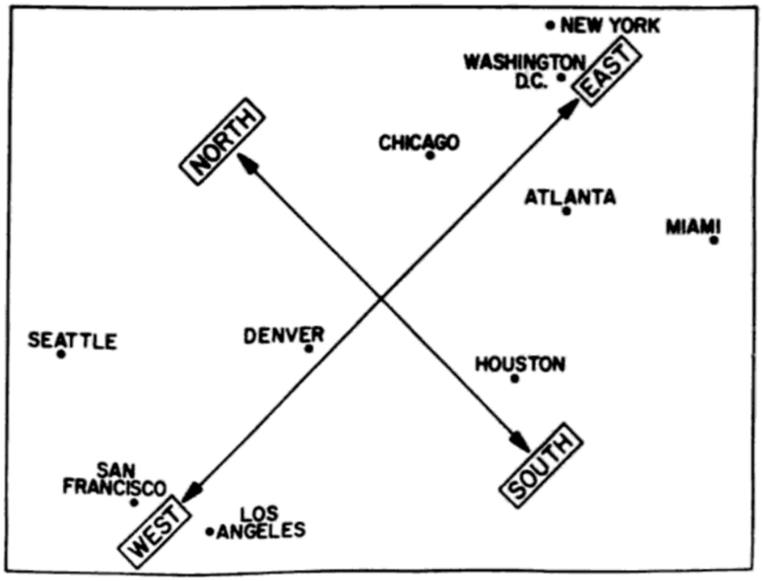


Figure 3.3* The result of using multidimensional scaling to to map the cities into $\mathbb{R}^2$, with cardinal directions manually added in.

Multidimensional scaling represents a simple solution to the embedding of time series into $\mathbb{R}^n$. However, its running time is $O(N^2)$, where N is the number of points, which makes it impractical for data that is potentially real-time. Also, it requires the calculation of the distance between each pair of time series, which is not desirable when using an expensive distance function like the DTW distance. An example of a more sophisticated method of isometric embedding is known as FastMap and is described by Faloutsos and Lin (1995). It is an approximate method, but it runs much faster than classical multidimensional scaling.

3.1.2 *k*-means clustering

Once we have the points $X \subset \mathbb{R}^n$ corresponding to our time series in P, we can run one of many clustering algorithms that operate in $\mathbb{R}^n$. The most common such algorithm is called k-means clustering. The algorithm works as follows. The goal is to partition X into k clusters $O_1, \ldots, O_k$.

Before describing the algorithm, we generalize the notion of the average of a set of numbers to the *centroid* of a set of points in $\mathbb{R}^n$.

Definition 10 (Centroid)**.** *The* ***centroid*** *of a set of points $X \subset \mathbb{R}^n$ is*

$$\hat{X} = \frac{1}{|X|} \sum_{x \in X} x.$$

The k-means algorithm works as follows. We initialize k *cluster centers* $o_1, \ldots, o_k$ to random points in $\mathbb{R}^n$ (for example, take a k-element subset of X). Then the set X is partitioned such that the points closest to the cluster center o_i is assigned to the cluster O_i. Then we set $o_i = \hat{O}_i$, the centroid of the set O_i, and repeat until convergence. The result is a partition of X into k clusters.

Since each point in X is a representative of a time series in P, we have achieved a partition of P into clusters, with the clusters representing different patterns. However, many practical problems appear in using k-means for this application. In particular, it requires our entire dataset to fit into memory, which is not necessarily the case. In fact, we might not even have the entire dataset yet, in the case of real-time data.

Before moving onto an improved algorithm, notice that in the algorithm, each cluster is represented by a *cluster center o_i*, which is updated until a desirable partition is achieved. The BIRCH algorithm described later will generalize the notion of a cluster center.

3.1.3 Gaussian mixture models

It is worth noting that a technique known as *Gaussian mixture models* is a generalization of k-means clustering that assigns a *probability* that each data point belongs to a particular cluster, instead of simply membership. Having a probability may become important when we attempt to forecast using the discovered patterns; using Gaussian mixture models to cluster should be explored in future research.

3.1.4 The BIRCH algorithm

Another clustering algorithm is the BIRCH algorithm, as described by Zhang et al. (1996). It is an *online* algorithm, meaning that it can efficiently process a stream of data points, instead of requiring that we initially have the entire dataset.

Intuitively, it works as follows. Suppose we are provided with a new point that we must either place into one of several existing clusters or place into a newly created cluster. The algorithm takes the point and finds its closest existing cluster and inserts the point into that cluster. If the resulting cluster is too “big” in some sense to be a cluster, the algorithm splits that cluster into two clusters.

We quantify this last part with the following definition.

Definition 11 (Radius)**.** *The **radius** of a cluster $O \subseteq X$ is the root-mean-square distance between its points and its centroid*:

$$R(O) = \sqrt{\frac{1}{|O|} \sum_{x \in O} ||x - \hat{O}||^2}.$$

We can use the radius to quantify when to split the cluster: when the radius of a cluster exceeds some threshold R_0, we create two clusters of smaller radius.

This algorithm can be made efficient by using a different representation of existing clusters. Instead of cluster centers as in k-means, we define *cluster features*.

Definition 12 (Cluster feature)**.** *The **cluster feature** (CF) of a cluster $O \subseteq X$ is a triple (N, s, S^2), where*

$$N = |O| \qquad s = \sum_{x \in O} x \qquad S^2 = \sum_{x \in O} ||x||^2.$$

Lemma 1. *If (N, s, S^2) is a cluster feature of a cluster O, then its centroid and radius are*

$$\hat{O} = \frac{s}{N} \qquad R(O) = \sqrt{\frac{S^2 - 2\langle s, \hat{O}\rangle + ||\hat{O}||^2}{N}}$$

respectively.

Proof.

$$\begin{aligned} \hat{O} &= \frac{1}{|O|}\sum_{x\in O} x \\ &= \frac{s}{N}. \\ R(O) &= \sqrt{\frac{1}{|O|}\sum_{x\in O} ||x - \hat{O}||^2} \\ &= \sqrt{\frac{1}{|O|}\sum_{x\in O} \left[||x||^2 - 2\langle x, \hat{O}\rangle + ||\hat{O}||^2\right]} \\ &= \sqrt{\frac{S^2 - 2\langle s, \hat{O}\rangle + ||\hat{O}||^2}{N}}. \end{aligned}$$

□

Lemma 2. *If the cluster feature of O is (N, s, S^2), then the cluster feature of $O \cup \{x\}$ is $(N + 1, s + o, S^2 + ||x||^2)$.*

Proof. This is straightforward to verify through computation. □

Therefore, instead of maintaining a list of clusters in memory, we can maintain a list of cluster features. When we insert a new point, we find its closest existing cluster by calculating the distance to the centroid of each cluster feature. For efficient access of the list of cluster features, we can store the list of cluster features in a *CF tree* (cluster feature tree).

Definition 13 (CF tree)**.** *A **CF tree** with maximum branching factor $B \geq 2$ and threshold radius R_0 is a tree with cluster features as nodes, where the number of children of each node does not exceed B, and the radius of the cluster corresponding to each leaf node is less than R_0.*

Theorem 3. *The following algorithm for the insertion of a new item x into a CF tree preserves the CF tree structure*:

1. *Descend the CF tree by recursively choosing the node whose centroid is closest to x, until we reach a leaf cluster feature o.*

2. *If the radius of o would exceed R_0 if x were added to its corresponding cluster, insert a new cluster feature corresponding to an empty cluster as a sibling of o. Let o now denote this new cluster feature.*

3. *Insert x into each of the clusters on the path from the leaf o to the root of the tree, and update each cluster feature appropriately to reflect the addition of x.*

4. *If, due to the insertion of a new cluster feature in step 2, the number of children of a node o' exceeds B, split the node as follows. Let o_1 and o_2 be the two child nodes of o' whose centroids are farthest from each other. Remove every child node of o' except o_1 and o_2, and reinsert these removed child nodes as child nodes of o_1 and o_2, depending on which one is closer. Update o_1 and o_2 to reflect the insertion of these new child nodes.*

Proof. Steps 2 and 4 preserve the constraint on leaf node radius and branching factor, respectively. □

The simplest version of the BIRCH algorithm is the following two-step process for clustering a set of points in $\mathbb{R}^n$. First, initialize an empty CF tree and insert every point into the tree. Second, perform a global clustering method such as k-means clustering on the centroids of each leaf node in the CF tree. The resulting clusters of centroids can be regarded as clusters of the original data, containing points corresponding to the centroids they contain. This second step is necessary because the structure of the CF tree is strongly determined by the order in which points are inserted. One true cluster can easily be split up across two cluster features. The global clustering algorithm serves to fix these differences. The BIRCH algorithm remains fast even though it takes advantage of a global clustering algorithm, as it only operates on a compact representation of the points.

The main advantage of BIRCH is speed and that it requires only a single pass over the points, as contrasted with a global method like k-means clustering.

3.2 Clustering beyond $\mathbb{R}^n$

The two-step process of mapping our time series into $\mathbb{R}^n$ and then performing clustering is easy to understand but is roundabout and expensive. Can we avoid having to embed our time series into $\mathbb{R}^n$? It turns out that the BIRCH algorithm has a natural generalization, called BUBBLE, that avoids having to do this.

3.2.1 The BUBBLE algorithm

Notice that the BIRCH algorithm relies on calculating the centroids of each cluster, which can only be performed in $\mathbb{R}^n$. In other words, the BIRCH algorithm can only cluster subsets of $\mathbb{R}^n$. Can we cluster arbitrary sets, provided that there is a distance defined on that set? This is achieved by an algorithm called BUBBLE, as described by Ganti et al. (1999).

To achieve this goal, the BUBBLE algorithm defines a generalization of the centroid.

Definition 14 (Clustroid)**.** *The* ***clustroid*** *of a set X is*

$$\hat{X} = \arg\min_{x \in X} \sum_{x' \in X} d(x, x')^2,$$

the element in X that minimizes the total squared distance to every other element in the set.

We also generalize the radius:

Definition 15 (Radius)**.** *The* ***radius*** *of a cluster O is the root-mean-square distance between its points and its clustroid*:

$$R(O) = \sqrt{\frac{1}{|O|} \sum_{x \in O} d(x, \hat{O})^2}.$$

We are now ready to generalize cluster feature.

Definition 16 (Cluster feature*)**.** *The* ***cluster feature**** *of a cluster O is a tuple containing the following information*:

1. *the number of elements* $|O|$
2. *the clustroid* $\hat{O}$
3. $2p$ *elements of* O*, where* p *is user-defined*
4. *the value of* $\sum_{x' \in X} d(x, x')^2$ *for each representative element*
5. *the radius* $R(O)$.

All of these statistics in the cluster feature* can be incrementally maintained as elements are inserted and play a role in the maintenance of the clusters, just as the cluster feature did in the BIRCH algorithm. For details, refer to Ganti et al. (1999).

With BUBBLE, we have an efficient clustering method for clustering time series into different patterns. These clusters correspond to patterns discovered in the time series. How can we use the discovered patterns for prediction?

Chapter 4

Forecasting with discovered clusters

Suppose now that our time series are now all clustered into different patterns. We now know that, for example, our time series exhibits Pattern 1 at time t_1, Pattern 2 at time t_2, etc. One naive probabilistic approach to forecasting uses what is called a *Markov chain*.

4.1 Markov chain

We will use daily stock price data as an example. Imagine that we've labeled each day in our AAPL stock data as exhibiting Patterns 1, 2, or 3. Suppose that in our training data, we notice that if on one day the time series exhibits Pattern 1, the next day it exhibits Pattern 1 a quarter of the time, Pattern 2 a half of the time, and Pattern 3 a quarter of the time. Now suppose that we observe that today, the AAPL stock exhibits Pattern 1. Then one naive approach would be to predict that tomorrow, the AAPL stock will exhibit Pattern 1 with probability $\frac{1}{4}$, Pattern 2 with probability $\frac{1}{2}$, and Pattern 3 with probability $\frac{1}{4}$.

To formalize this, we define the notion of a *Markov chain*.

Definition 17 (Markov chain)**.** *A **Markov chain** is a sequence of random variables $X_1, X_2, \ldots$ such that for all $t \in \mathbb{N}$,*

$$P(X_{t+1} \mid X_1, \ldots, X_t) = P(X_{t+1} \mid X_t).$$

The random variables $X_1, X_2, \ldots$ represent the state of the system; the condition states that the state X_{t+1} at time $t + 1$ depends only on the state X_t at time t, and nothing else before.

If we let X_t be the pattern exhibited by the stock at time t, this is exactly the assumption that we made at the beginning of this section. Formally, then, to forecast a time series that has k patterns, we will let $\{X_1, X_2, \ldots\}$ be a Markov chain of patterns; in other words, X_t takes on values in $\{1, \ldots, k\}$, where $X_t = \alpha$ indicates that the stock is exhibiting Pattern α at time t. We will empirically determine $P(X_{t+1} = \beta \mid X_t = \alpha)$ by counting the number of times in our training data that Pattern α transitioned to Pattern β and dividing by the total number of times in our training data. Then, if we would like to predict the pattern that will be exhibited tomorrow, knowing that today the stock is exhibiting Pattern α, we simply take $P(X_{t+1} = \beta \mid X_t = \alpha)$ to be the probability that tomorrow's pattern is Pattern β.

We can generalize this approach in the following way. With the above approach to forecasting, we must know that today the stock exhibits a specific Pattern α. However, forecasting may be more accurate if we could express that we believe that today the stock exhibits, for example, 50% Pattern 1 and 50% Pattern 2. We can formalize this example as saying that

$$P(X_t = \alpha) = \begin{cases} \frac{1}{2} & \text{if } \alpha = 1 \\ \frac{1}{2} & \text{if } \alpha = 2 \end{cases}.$$

Then, to predict tomorrow's state, we can use the product rule

$$P(X_{t+1}) = \sum_{\alpha} P(X_{t+1} \mid X_t = \alpha)\, P(X_t = \alpha). \tag{4.1}$$

We can simplify this calculation with the following notation.

Definition 18 (State vector)**.** *Let X be a random variable taking on values in a set $\{\alpha_1, \ldots, \alpha_k\}$ with probabilities $P(X = \alpha_i) = p_i$. Then we say that X has a* ***state vector***

$$\begin{bmatrix} p_1 \\ \vdots \\ p_k \end{bmatrix}.$$

Note that $\sum_i p_i = 1$. Note that we will often abuse notation and write

$$X = \begin{bmatrix} p_1 \\ \vdots \\ p_k \end{bmatrix}$$

for the statement that X is a random variable with $P(X = \alpha_i) = p_i$.

Next, we define the *transition matrix* of a Markov chain.

Definition 19 (Transition matrix)**.** *The* ***transition matrix*** *of a Markov chain* $\{X_1, X_2, \ldots\}$ *whose random variables take on values of a set* $\{\alpha_1, \ldots, \alpha_k\}$ *is a k-by-k matrix A whose entries are*

$$A_{ij} = P(X_{t+1} = \alpha_i \mid X_t = \alpha_j).$$

Let A be the transition matrix of a Markov chain, and let X_t and X_{t+1} be today's and tomorrow's state vector respectively. In our situation, the state vector describes how likely we believe each pattern is at the current time. Then with this additional notation, we can rewrite the prediction rule Equation 4.1 as simply

$$X_{t+1} = AX_t,$$

where the multiplication on the right is matrix-vector multiplication. This makes it very easy to predict the probabilities of a given pattern being exhibited at the next time.

This leaves the question of how exactly to assign a state vector to the current time. Indeed, the clustering approach outlined in Chapter 3 assigns a single pattern (cluster) to each time step. A more powerful approach such as *Gaussian mixture models* will assign not just a single pattern to a time but a probability of each pattern to a single time.

The Markov assumption may be overly restrictive; it seems unreasonable that the pattern of a stock on one day depends only on the pattern of the stock the day before. While true, as a first attempt at prediction, this simplistic model will help us assess whether our method of pattern discovery is remotely effective. Also, recall that when we determine which pattern of a stock is exhibiting, we are actually taking into account the previous recent history of the stock at that time; thus, looking one time step prior actually involves considering more of the stock's recent history than just one day.

4.2 Other time series forecasting methods

This Markov chain approach is only one possible approach. Notice that by discovering patterns in a time series, we have effectively converted our original time series $p : T \to \Sigma$ into a time series $p' : T \to \mathbb{P}$, where $\mathbb{P}$ is a space of patterns. Therefore, after applying this methodology, we can employ any other method for forecasting time series to this new time series

p'. In this way, we can think of this method as a dimensional reduction method, where we've reduced the information in a time series to what really matters, the pattern exhibited at each time.

Chapter 5

Results

To summarize, the methodology I have described to discover patterns in and to forecast time series is as follows. First, we define a distance between histories at different times, using Euclidean distance or dynamic time warping. Next, we use this distance to cluster, using isometric embedding followed by clustering in $\mathbb{R}^n$, or using a clustering method like BUBBLE to cluster the time series directly. Then, we can use a Markov chain approach or any number of existing forecasting methods to forecast the time series.

5.1 Implementing the framework

To put this general framework of this method for pattern discovery to the test, I implemented a basic version of this framework. First, I acquired historical daily price data for the S&P 500 index for the past ten years; this data is depicted in Figure 5.1. This resulted in a time series with 2727 data points, one for each day, excluding weekends and holidays. Next, I split the data into overlapping moving windows of width $w = 15$, resulting in 2712 segments of 15 days each. From each of these segments, I subtracted the mean of the price within the segment so that the price was now centered around 0 within each segment.

Next, I used Euclidean distance to calculate the similarity between each of these segments, as described in Chapter 2. Since these segments were already elements of $\mathbb{R}^{15}$, I did not need to use isometric embedding to embed them in $\mathbb{R}^n$ before clustering. I used k-means clustering with $k = 6$ clusters to determine the patterns in Table 5.2. Each of the 2712 segments was assigned to a pattern.

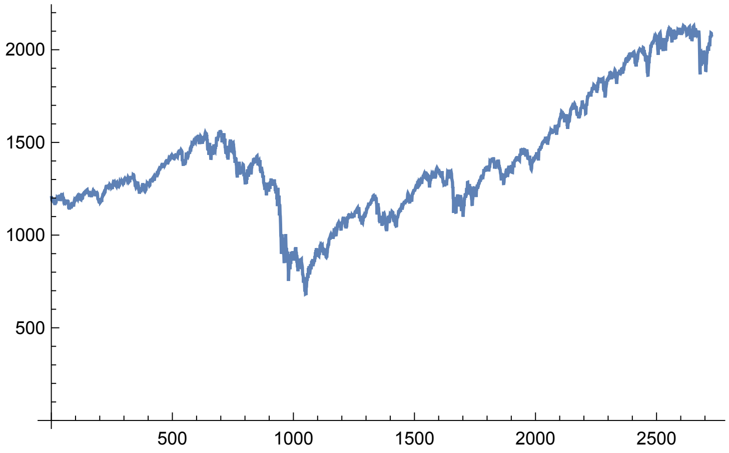


Figure 5.1 The S&P 500 index from January 1, 2005 to November 1, 2015, plotted with days on the horizontal axis and the index on the vertical axis.

Finally, I constructed a Markov chain, as described in Chapter 4, using the observed transitions in the stock data. The states of this Markov chain (the different patterns) and their transition probabilities are illustrated in Figure 5.3.

5.2 Discussion and future work

First, let us evaluate this approach to pattern discovery using these preliminary results. Of what quality are the discovered patterns? There are several observations, observations that mostly point favorably towards this method:

- If the algorithm is rerun, the same patterns are discovered, meaning that the patterns discovered are not particularly volatile, as is sometimes the case in k-means clustering. This means that this approach to pattern discovery is consistent.

- The patterns that we have discovered are comparatively uninteresting. They are mostly increasing or decreasing trends (Patterns 1, 4, and 6), although patterns like Patterns 2 and 5 are somewhat interesting. They are no head-and-shoulders patterns, but they may prove to be more useful for predicting behavior.

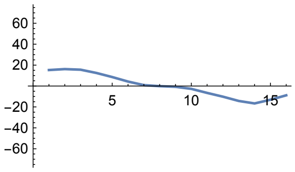
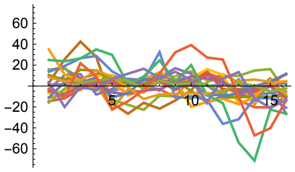
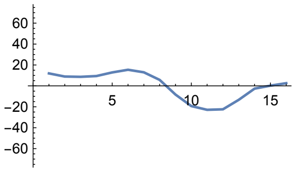
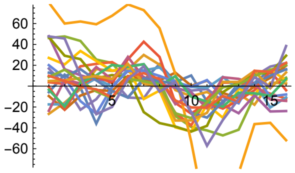
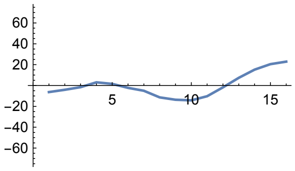
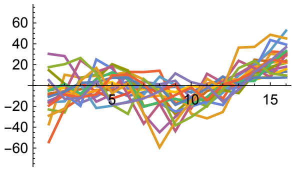
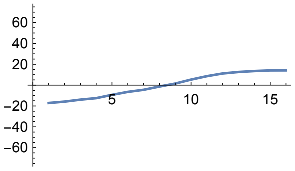
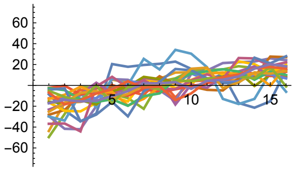
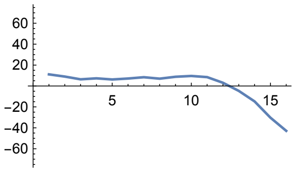
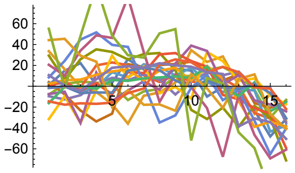
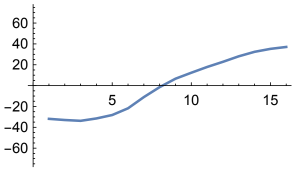
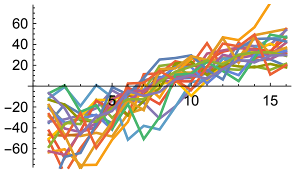
Label	Pattern	Examples	Count
1			785
2			203
3			249
4			983
5			243
6			249

Table 5.2 Patterns discovered in the S&P 500 data.

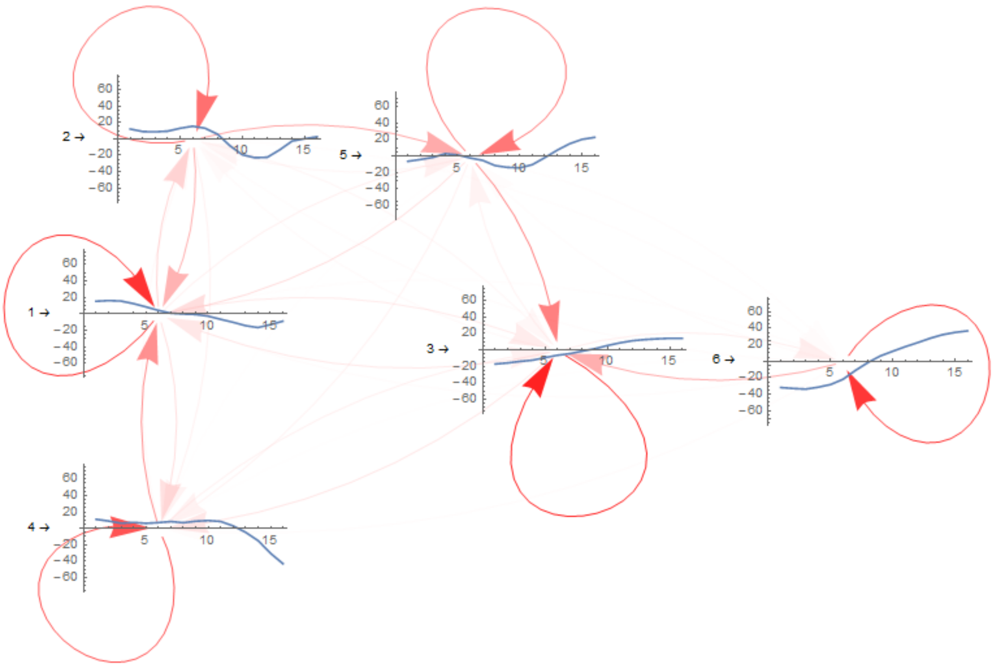


Figure 5.3 The Markov chain of patterns, learned from the S&P 500 data. Darker arrows indicate higher probabilities of transitions between the connected states.

- Looking at the patterns discovered, we find that each pattern is represented at many different times in the S&P 500. It might have been problematic if there were patterns with only one or two examples within the data.
- The variance is high within some of the patterns; in other words, there are some examples within a pattern that, by eye, do not seem to correspond very well to the pattern in question. For example, in Pattern 5, there are extreme outliers that on average cancel to produce a flatline pattern in the first ten days. Perhaps this is an indicator that a way to eliminate outliers in the data is needed.

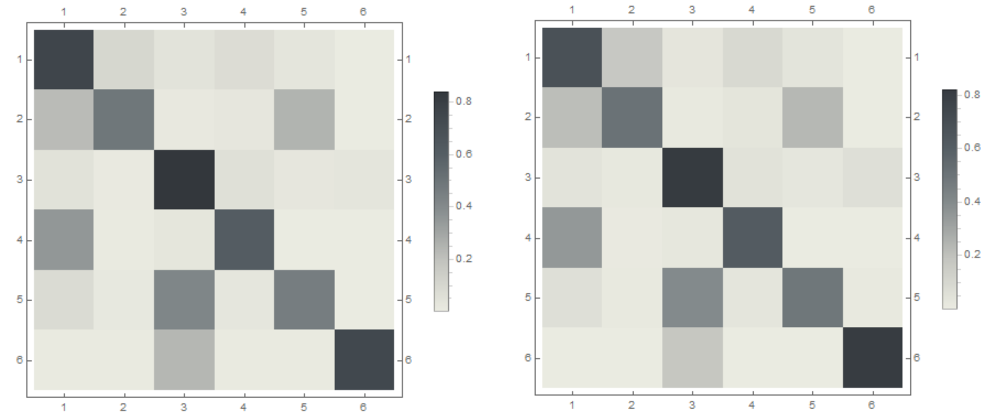
Overall, the clustering approach to pattern discovery seems promising, especially as we move to more sophisticated measures of distance. There is potentially a need for a way to eliminate extreme segments, ones that don't clearly belong to one cluster.

Next, let us evaluate the Markov chain model as an approach to prediction. The primary assumption is that historical data can be used to predict future data. One simple way to test this assumption is to calculate the Markov chain transition probabilities for data prior to a certain date and data after a certain date and compare the probabilities. If they are consistent, then it is plausible that the Markov chain model is an acceptable way to perform forecasting. These probabilities are depicted in Figure 5.4; we can see that they are mostly consistent for data before and after May 21, 2010. Figure 5.5 illustrates that the difference between the two matrices is small. Since they are consistent, this Markov chain approach seems very promising.

One observation is that the transition probabilities are highest for self-transitions, transitions from a state to itself. This makes a lot of sense; in fact, it forms the basis for a stock trading technique called momentum trading, in which one bets on rising stocks to continue rising and falling stocks to continue falling. This may pose a challenge, however, to this method: this momentum strategy may be hard to improve upon.

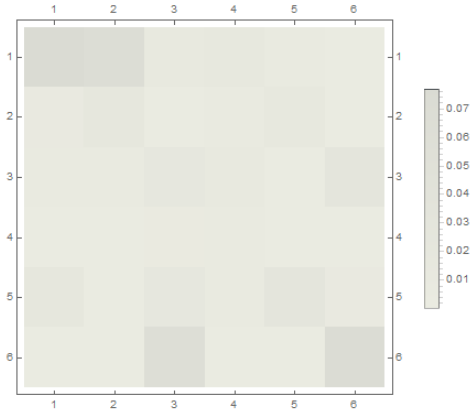
For true prediction, I hope to use the strategy outlined in Chapter 4, to consider a state vector of patterns rather than just a single pattern. To do this, I will need to move to using a Gaussian mixture model to cluster rather than simply k-means clustering.

There are two hyperparameters in this algorithm: k, the number of patterns to look for, and w, the width of each segment of the time series. These were both chosen arbitrarily; it would be nice to have an iterative scheme to choose these parameters. To tune k, we could incrementally increase k until the quality of patterns degrades, where the quality is assessed by some criteria. To tune the window width w, the same principle could in theory be applied. However, observe that Patterns 2, 3, and 5 all exhibit some sharp change around days 5 and 10, raising the possibility of an intrinsic time scale of around 5 days. In fact, this is very natural, as the trading week lasts 5 days (trading does not occur on the weekends). I believe that it is possible to extract this intrinsic time scale from the data, an exciting direction of research in itself.



$$\begin{pmatrix} 0.7580 & 0.0988 & 0.0403 & 0.0707 & 0.0290 & 0.0029 \\ 0.2220 & 0.4880 & 0.0133 & 0.0231 & 0.2500 & 0.0033 \\ 0.0458 & 0.0088 & 0.8400 & 0.0538 & 0.0230 & 0.0290 \\ 0.3530 & 0.0081 & 0.0267 & 0.6090 & 0.0020 & 0.0020 \\ 0.0783 & 0.0141 & 0.4200 & 0.0248 & 0.4590 & 0.0035 \\ 0.0052 & 0.0051 & 0.2380 & 0.0052 & 0.0051 & 0.7410 \end{pmatrix} \quad \begin{pmatrix} 0.6810 & 0.1670 & 0.0278 & 0.0868 & 0.0364 & 0.0010 \\ 0.2130 & 0.5090 & 0.0097 & 0.0314 & 0.2340 & 0.0024 \\ 0.0407 & 0.0154 & 0.8200 & 0.0426 & 0.0237 & 0.0573 \\ 0.3500 & 0.0089 & 0.0223 & 0.6150 & 0.0022 & 0.0022 \\ 0.0572 & 0.0120 & 0.4010 & 0.0300 & 0.4880 & 0.0120 \\ 0.0021 & 0.0021 & 0.1760 & 0.0021 & 0.0021 & 0.8160 \end{pmatrix}$$

Figure 5.4 The transition probabilities of the Markov chain learned from the S&P 500 data before (left) and after (right) May 21, 2010.



$$\begin{pmatrix} 0.0769 & 0.0679 & 0.0125 & 0.0161 & 0.0074 & 0.0019 \\ 0.0091 & 0.0217 & 0.0036 & 0.0083 & 0.0164 & 0.0009 \\ 0.0051 & 0.0066 & 0.0193 & 0.0111 & 0.0007 & 0.0283 \\ 0.0029 & 0.0007 & 0.0044 & 0.0062 & 0.0002 & 0.0002 \\ 0.0211 & 0.0021 & 0.0193 & 0.0052 & 0.0289 & 0.0085 \\ 0.0030 & 0.0030 & 0.0625 & 0.0030 & 0.0030 & 0.0745 \end{pmatrix}$$

Figure 5.5 The absolute difference between the transition probabilities.

Part II

Introducing a Riemannian metric on data manifolds

Chapter 6

Introduction: data as a manifold

In Chapter 2, we developed a distance d on the set Σ of elements that the time series we are studying take on. In Chapter 3, we thought of Σ as not simply a set but also a *space*, in which it is possible to perform clustering. The distances we defined, however, are not applicable to data in general. For example, the dynamic time warping distance that we used is definable between time series windows, but it is not a useful notion between, say, images of different handwritten numbers, or the characteristics of different voters' voting preferences. Is it possible to define a notion of distance on a set that contains *arbitrary* data, a unified definition that can be applied to *any* type of data?

In the following chapters, we will attempt to generalize from the discussion in Part I and move to a more abstract setting. This chapter will make concrete what is meant by "data" and motivate the use of Riemannian geometry to study it. Chapter 7 will attempt to define a Riemannian metric on the space of data. Then, Chapter 8 will introduce basic concepts in a relatively new field called information geometry that will allow us to view the space of data in a new light, in Chapter 9.

6.1 Manifolds, Riemannian metrics, and geodesics

This section will describe some basic concepts in Riemannian geometry at an intuitive level. For a rigorous description, see any standard textbook on Riemannian geometry, such as do Carmo (1992).

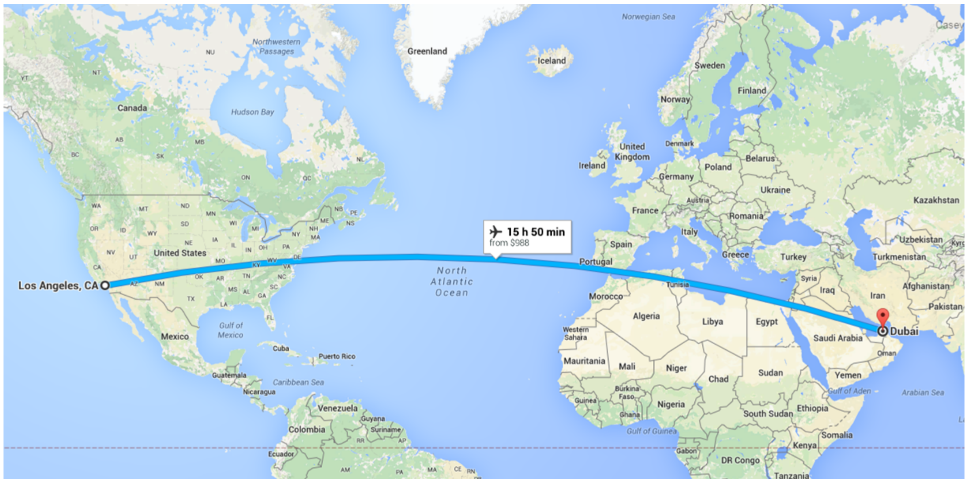


Figure 6.1* Google Maps uses the Mercator projection and erronenously depicts the shortest path between Los Angeles and Dubai as (approximately) the straight line connecting them.

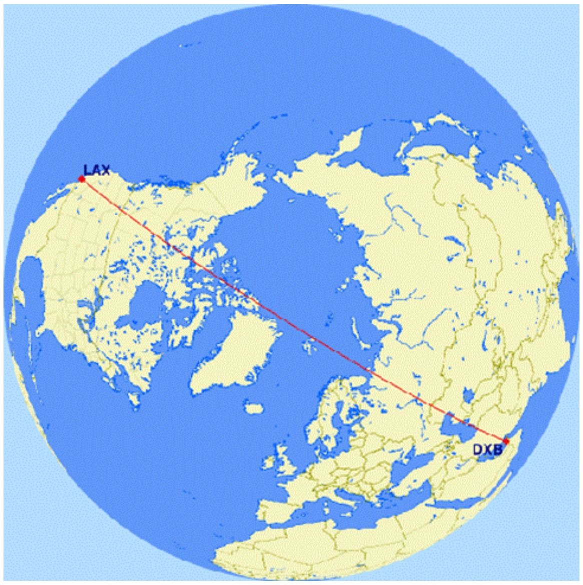


Figure 6.2* The true shortest path travels north of the Arctic Circle.

A **differentiable manifold**, or manifold for short, is a generalization of surfaces embedded in $\mathbb{R}^3$ that are in some sense smooth. Whereas surfaces must be two-dimensional, manifolds may be n-dimensional. One intuitive characteristic of n-dimensional manifolds is that if at any point, you zoom in far enough, it looks flat like a copy of $\mathbb{R}^n$. Since a curve looks like $\mathbb{R}$ when zoomed in far enough, and a surface looks like $\mathbb{R}^2$ when zoomed in far enough, curves and surfaces are one- and two-dimensional manifolds if they are smooth enough. The Earth, for example, can be thought of as a two-dimensional manifold, since at each point on the Earth, it locally looks like a flat, two-dimensional plane.

Moreover, whereas surfaces are usually thought of as embedded in $\mathbb{R}^3$ (or a higher-dimensional space), manifolds can be thought of as independent objects, free of any embedding space. Just as the Earth can be described using a bunch of flat two-dimensional maps that cover the entire surface, n-dimensional manifolds are described with by a bunch of subsets of $\mathbb{R}^n$ that act as maps that cover the entire manifold. The technical term for this collection of maps is an *atlas* for the manifold.

Riemannian manifolds add additional structure, namely the ability to redefine distance between points. This is best explained with an example. The map shown in Figure 6.1 uses the Mercator projection, and a line connecting two points is not necessarily the shortest path. A flight from Los Angeles to Dubai—which takes the shortest path—does not take the path in Figure 6.1, even though it is a straight line on this particular map. Instead, it takes the path in Figure 6.2, which would appear very different from a straight line on the map in Figure 6.1. Thus, in order to fully describe a manifold, we must somehow specify how shortest paths are distorted, in addition to an atlas. This true shortest path between to points is called a **geodesic** between those two points.

Recall that in Euclidean space, the length of a curve $\gamma : I \to \mathbb{R}^n$ is

$$\ell(\gamma) \equiv \int_I \sqrt{\tfrac{\mathrm{d}\gamma}{\mathrm{d}t} \cdot \tfrac{\mathrm{d}\gamma}{\mathrm{d}t}}\,\mathrm{d}t = \int_I \left|\frac{\mathrm{d}\gamma}{\mathrm{d}t}\right| \mathrm{d}t,$$

where $\cdot$ is the Euclidean dot product. Thus, to alter what is considered a geodesic on a manifold, we replace the dot product with an inner product of our choosing:

$$\ell(\gamma) \equiv \int_I \sqrt{\langle \tfrac{\mathrm{d}\gamma}{\mathrm{d}t}, \tfrac{\mathrm{d}\gamma}{\mathrm{d}t} \rangle_{\gamma(t)}}\,\mathrm{d}t,$$

where the curve γ is now defined on maps in the atlas. The inner product $\langle \cdot, \cdot \rangle_\cdot$ is called the **Riemannian metric** on the manifold. To complete the

Price/$1000	Baths	Bedrooms	Elevation/ft	Area/ft^2	Age/yr
999	2	1	10	1000	56
1350	2	2	9	2150	116
629	1	1	9	500	113

Table 6.3* Three rows of a dataset of housing prices in San Francisco.

example, the correct Riemannian metric for Mercator projection maps of the Earth, as computed by Rowe (2015), is

$$\langle u, v\rangle_{(x,y)} = \frac{4\pi^2R^2}{\cosh^2(\frac{2\pi y}{H})}\left[\frac{u_x v_x}{W^2} + \frac{u_y v_y}{H^2}\right],$$

where (x, y) denotes a point on the map, R is the radius of the Earth, H and W are the height and the width of the map, and u_x, u_y, v_x, and v_y denote the x and y components of u and v. By specifying a Riemannian metric along with an atlas, we capture the true geometry of the manifold.

The **distance** between two points of a manifold is then defined as the length of the geodesic connecting those two points.

6.2 The manifold hypothesis

Consider a dataset of housing prices in San Francisco, a sample of which is shown in Table 6.3. Each row corresponds to a house in San Francisco and contains 6 columns of numerical information, such as price, number of bedrooms, and age. In this way, we can view the entire dataset as a set of houses, where each house corresponds to a point in $\mathbb{R}^6$. If we had more information about each house, each new piece of information would adjoin an new dimension, assuming it is real-valued. We will restrict ourselves to numerical data in our discussion.

Therefore, one naive characterization of data might simply be to declare all (finite) subsets of $\mathbb{R}^n$ to be datasets. However, this ignores the condition that data must in some sense contains information. The *manifold hypothesis* is an assumption about the nature of data that captures this notion: it says that observed data tends to lies near a low-dimensional manifold embedded within the higher-dimensional space $\mathbb{R}^n$. It suffices to visualize the manifold as a curve, a surface, or a generalization thereof in higher dimensions.

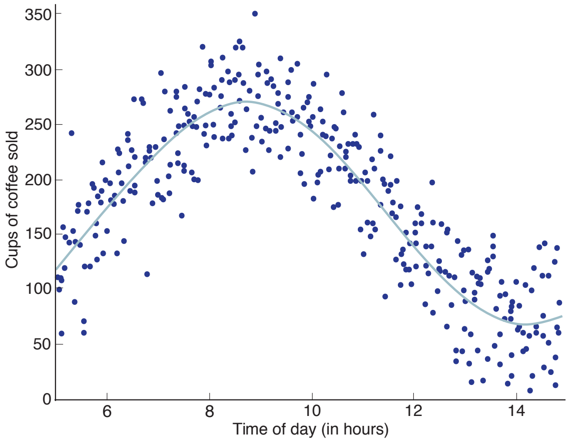


Figure 6.4* Coffee sales by hour lie on a one-dimensional curve in $\mathbb{R}^2$.

Two examples of the manifold hypothesis in action are found in Figure 6.4 and Figure 6.5.

Figure 6.4 depicts coffee sales vs. hour of the day over many days; we can see that the data is approximately constrained to a one-dimensional curve. Thus the data approximately lies on a one-dimensional manifold embedded in $\mathbb{R}^2$.

Figure 6.5 depicts a set of 64-by-64-pixel black-and-white images. Each image can be viewed as a point in $\mathbb{R}^{4096}$, with each dimension indicating the brightness of each of the $64^2 = 4096$ pixels. However, the images are taken with only three degrees of freedom: left-right pose, up-down pose, and lighting direction. This means that the images likely lie near a three-dimensional manifold embedded in $\mathbb{R}^{4096}$.

It is worth noting that there exists a recent algorithm due to Fefferman et al. (2013) that tests whether a particular dataset satisfies the manifold hypothesis, i.e. whether there exists a manifold such that the points in the dataset lie close to it. There are also many algorithms that attempt to find coordinates for the points on the manifold; generally, this is known as dimensional reduction. One such algorithm that directly uses the intuition of the manifold hypothesis is called *local tangent space alignment*, explored in Appendix A.

This brings us to the impetus behind this research. Since Riemannian geometry is a natural candidate of a tool to endow manifolds with a notion of distance, it is natural to ask if a Riemannian metric can be assigned to

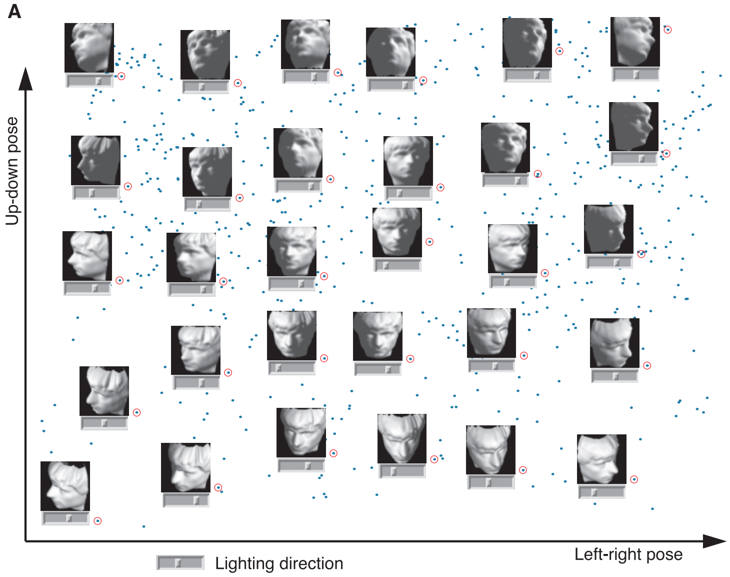


Figure 6.5* A set of 64-by-64-pixel images lie on a three-dimensional manifold in $\mathbb{R}^{4096}$.

the data manifold. Because the data manifold is difficult to extract from the embedding space, we will tackle a similar problem: can we assign a Riemannian metric to the *embedding space* $\mathbb{R}^n$, in such a way that embedded data manifold inherits useful properties? This, then, is the question to be tackled in this part:

Key Question 2. *Given data represented as points in $\mathbb{R}^n$, is it possible to assign a Riemannian metric to the embedding space $\mathbb{R}^n$ in a natural way*?

6.3 Data as a probability distribution

Data typically comes in the form of a finite sample of points. Because the machinery of Riemannian geometry deals with smooth manifolds, we must convert the observed data points into a smooth object to facilitate its translation into the world of manifolds. Probability theory provides one way to do so.

Suppose we are given a finite sample of m data points $\{x_1, \dots, x_m\} \subseteq \Sigma = \mathbb{R}^n$. As is standard in statistical learning theory, we assume that these points are samples drawn from some unknown probability distribution with a probability density function $p : \Sigma \to \mathbb{R}$, which will act as our smooth object. There are several ways to estimate p from $\{x_1, \ldots, x_m\}$, three examples of which are discussed below.

6.3.1 Maximum likelihood estimation

In *maximum likelihood estimation*, we assume that the observed data comes only from a family of possible probability distributions. For example, suppose we know that the data is one-dimensional and comes from a normal distribution, but with unknown parameters μ and σ. In this case, estimating μ and σ from the data points $\{x_1, \ldots, x_m\}$ will suffice to estimate the probability density function p.

In general, let Θ be the set of possible parameters, and let $p_\theta(x)$ denote the probability density function parameterized by $\theta \in \Theta$. Then the *maximum likelihood parameter* $\hat{\theta}$ is the parameter $\theta \in \Theta$ such that the *likelihood*, the probability of observing $\{x_1, \ldots, x_m\}$ under the probability density function p_θ, is maximized. The *maximum likelihood distribution* is the optimal distribution $p_{\hat{\theta}}$.

In the normal distribution example above, we let

$$\Theta = \{(\mu, \sigma) \mid \mu \in \mathbb{R}, \sigma \in \mathbb{R}^+\}$$

and maximize the likelihood

$$L(\theta) = \prod_{i=1}^{m} p_\theta(x_i)$$

while varying $\theta \in \Theta$. If $\hat{\theta}$ is the optimal $\theta \in \Theta$, then $p_{\hat{\theta}}$ is the estimated distribution. We will see an interesting way to view this method in Chapter 8.

The downside to this method is that the family of distributions must be specified a priori. While this may produce excellent results when there is reason to believe that data comes from a certain distribution, it is in general difficult to know from which family of distributions given data is from. Indeed, the construction of a probability density function from arbitrary data is better suited for so-called *non-parametric* methods, which do not require such prespecification.

6.3.2 Kernel density estimation

The most common non-parametric method for density estimation is known as *kernel density estimation*. In this method, the set of data points in Σ is

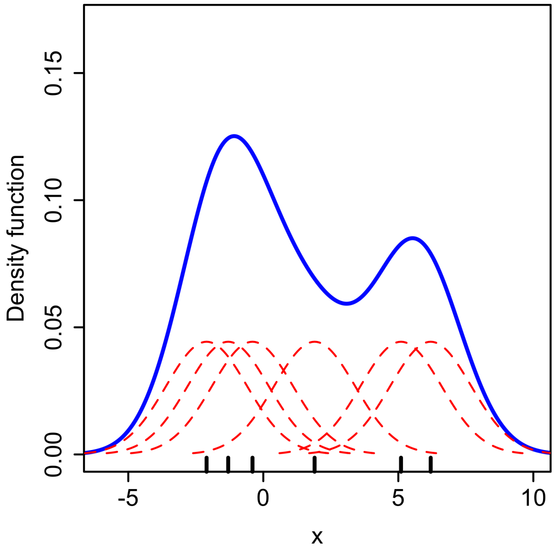


Figure 6.6* Kernel density estimation in one dimension. Kernel density estimation produces an estimate of the probability density function from a set of points (shown along the x-axis here) by summing normal distributions at those points.

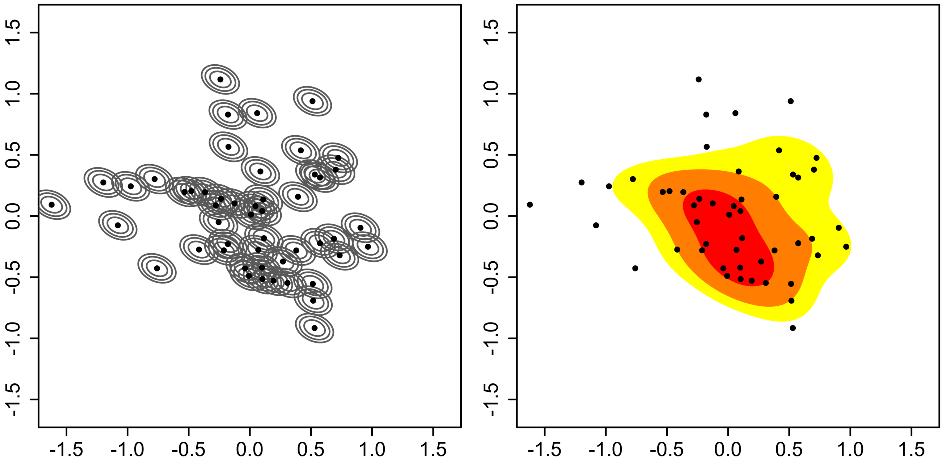


Figure 6.7* Kernel density estimation in two dimensions. The set of points and their associated normal distributions are on the left, and the estimated probability density function is on the right.

represented as a sum of Dirac delta functions centered on those points, and these spikes of mass 1 are then convolved with a kernel K, usually a normal distribution. We can think of this process as taking the finite set of observations and "smearing them" out into a smooth probability distribution. For example, let

$$K(x) = \frac{1}{\sqrt{2\pi}} \exp\Big(-\frac{1}{2}x^2\Big)$$

be the probability density function of the standard normal distribution. Then the estimated probability density function is

$$p(x) = \frac{1}{m} \sum_{i=1}^{m} K(x - x_i).$$

It can be proven that such an estimate converges to the true probability density function as the number of observed points m approaches ∞.

We can see illustrations of kernel density estimation in one and two dimensions in Figure 6.6 and Figure 6.7.

This process tends to work very well in low-dimensional spaces (with approximately $n < 6$), but convergence in higher dimensions is too slow for the probability density function to be accurate with a reasonable number of data points.

6.3.3 Deep density models

The recent increase in popularity of so-called deep neural networks has inspired new approaches to density estimation. Among them is the concept of *deep density models* as introduced by Rippel and Adams (2013). In it, techniques from training neural networks are used to construct an invertible function f from the space $\Sigma = \mathbb{R}^n$ of datapoints to the unit hypercube $[0,1]^n$, in such a way that the distribution of the image of the datapoints in the unit hypercube is prescribed. For example, the unit hypercube could be prescribed to have a uniform distribution, or a product of beta distributions. Once the function f is learned and the probability density function on the unit hypercube is specified to be $\bar{p}$, the estimated probability density function on Σ is given by the usual change of variables formula:

$$p(x) = |\det \mathrm{D}f|\, \bar{p}(f(x)).$$

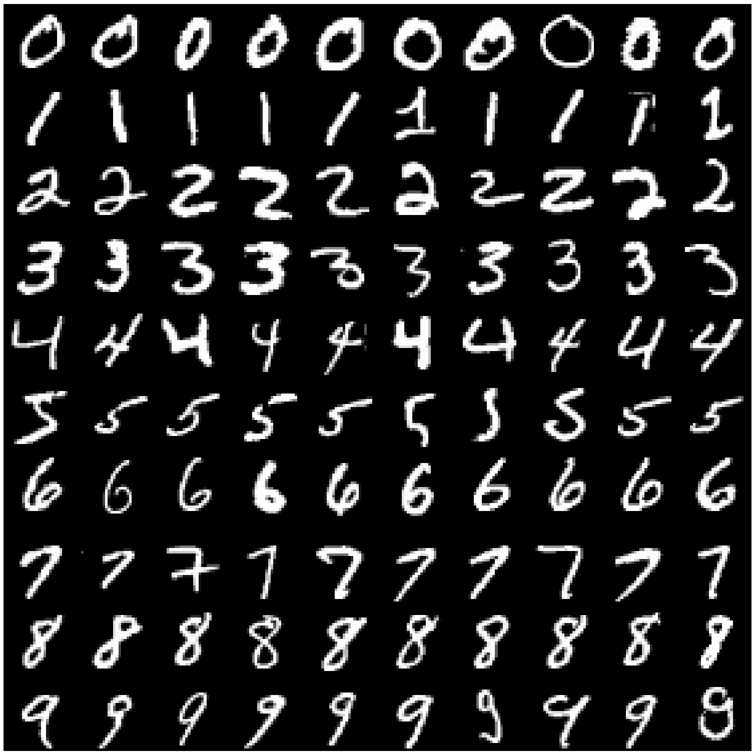


Figure 6.8* A subset of the MNIST dataset of handwritten numbers.

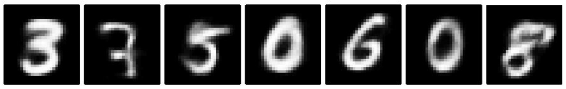


Figure 6.9* Generated samples from a deep density model trained on the MNIST dataset.

This method yields excellent results when tested on real, high-dimensional datasets. Rippel and Adams (2013) trained a three-layer deep density model to learn the probability distribution of the MNIST dataset, a dataset of 60,000 28-by-28-pixel images of handwritten digits, curated by LeCun et al. (2010). A subset of the MNIST dataset is shown in Figure 6.8. Once the probability distribution was learned, they then sampled directly from this distribution, resulting in the samples shown in Figure 6.9. These generated images are clearly recognizable as handwritten digits, indicating that the deep density model has constructed an accurate probability density function, even though the dimension of the embedding space is $28^2 = 784$.

From here, we will assume that the probability density function p corresponding to the observed data points has been found using one of these methods. It is clear that the state-of-the-art in density estimation is advancing in such a way that we will be able to rely on it as a means to convert the discrete set of data points into a smooth object.

Chapter 7

Proposals for Riemannian metrics on the data space

We would like to define a Riemannian metric on the data space $\Sigma = \mathbb{R}^n$, a continuous inner product on the tangent spaces of Σ. In this chapter, we use Σ to denote $\mathbb{R}^n$, viewed only as a differentiable manifold. For example, we will never use the vector space structure that Σ has as $\mathbb{R}^n$.

When trying to define the metric, it is useful to consider what properties we would like our metric to have. The most important property of the Riemannian metric is for the distance metric induced by it to act in a natural way. Recall that the Riemannian distance d between two points is the infimum of the lengths of all piecewise differentiable curves between those two points, where the length of a curve γ is defined as

$$\ell(\gamma) = \int \sqrt{\langle \gamma'(t), \gamma'(t) \rangle_{\gamma(t)}}\, \mathrm{d}t.$$

Since length-minimizing curves therefore are geodesics, it is therefore fruitful to think of the properties we would like for geodesics on the space. Let $x, y \in \Sigma$. Intuitively, where should a geodesic between x and y travel?

7.1 Behavior of geodesics

Recall that data tends to lie on a low-dimensional data manifold; therefore, it is natural that distance-minimizing curves should travel "along" the data manifold. It should avoid regions with low probability. This means that the metric tensor should be high in regions of low probability density, so that curves that pass through the region will have high lengths and therefore will not be length-minimizing. Similarly, we would like the metric tensor to be low in regions of high probability density so that length-minimizing curves will pass through those regions.

7.1.1 Scaled Euclidean metric

The simplest metric that satisfies this property simply scales the Euclidean dot product: we define

$$\langle u, v \rangle_x = \frac{u \cdot v}{p(x)^\alpha}, \tag{7.1}$$

where $\alpha > 0$, $x \in \Sigma$, and $u, v \in T_x\Sigma$. Written in matrix form, the metric induced by p is

$$g[p]_x = \frac{1}{p(x)^\alpha} I_n, \tag{7.2}$$

where I_n is the n-by-n identity matrix. We will attempt to determine α later.

We can numerically perform a sanity check on this metric to see if it matches our intuition. We will arbitrarily set $\alpha = 1$, so that the Riemannian metric is

$$\langle u, v \rangle_x = \frac{u \cdot v}{p(x)}.$$

First, let us try a simple choice of probability distribution, the two-dimensional standard multivariate normal distribution, with probability density function

$$p_0(x, y) = \frac{1}{2\pi} \exp\Big(-\frac{1}{2}(x^2 + y^2)\Big).$$

In Figure 7.1, we visualize this multivariate normal both as a contour plot and as a graph.

To understand how this metric behaves, we would like to visualize what all the points a distance at most r away from a given point p look like under this metric. This is known as a **geodesic ball** of radius r around a point p. To approximate what a geodesic ball of radius r looks like, we can pick a point p and integrate a geodesic for a time r in an arbitrary direction; the resulting point will be a distance r away from the original point. If we repeat this procedure for many different directions, eventually we will form a ring of points, all a distance r from the original point. This ring will outline the geodesic ball of radius r. If we repeat this procedure for increasing radii r, we will gain an understanding of what the space is like.

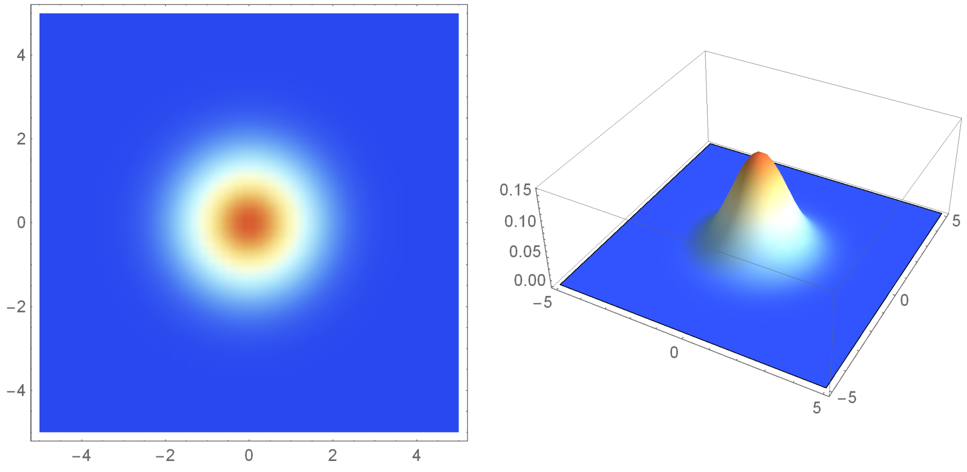


Figure 7.1 A standard normal distribution, visualized as a contour plot and as a graph.

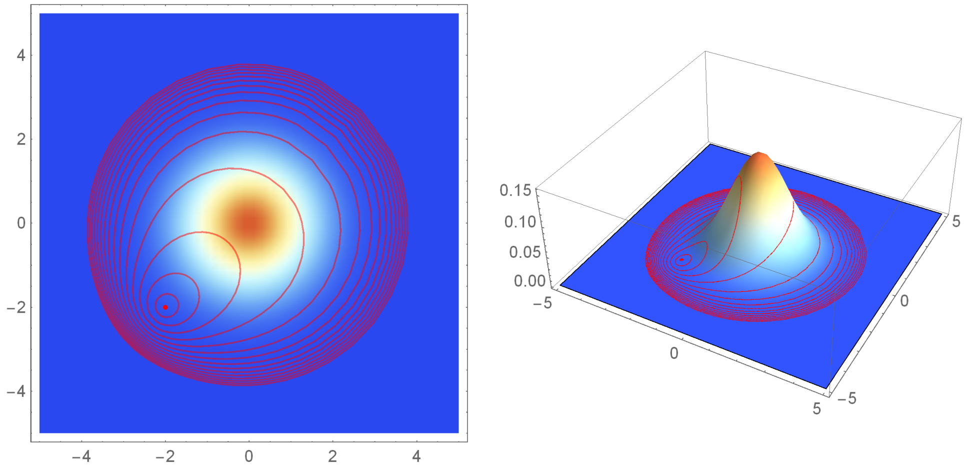


Figure 7.2 Geodesic balls (rings of constant radius) around the point $(-2, -2)$, under the scaled Euclidean metric ($\alpha = 1$) with the standard normal distribution. The high probability mass at the origin deforms them towards the origin.

Figure 7.2 depicts geodesic balls of various radii around the point $(-2,-2)$ under this metric, where we numerically integrate the geodesic equations using a Mathematica notebook adapted from Hartle (2002). Recall that in the Euclidean plane, rings of constant radius look like circles. However, because the multivariate normal distribution's probability mass is concentrated at the origin $(0,0)$, we would like the geodesic balls to favor that region; indeed, the rings of constant radius are skewed towards the higher-probability regions. Thus, the metric we have defined is acting as intended.

An alternative way to conceptualize this metric is one where the negative probability $-p(x)$ acts as a "gravitational potential" that attracts geodesics. This is visualized in Figure 7.3; again contours are points equidistant from $(-2,-2)$, in red.

This result is very promising for the scaled Euclidean metric. Let us now test this metric on a more complicated probability distribution. Consider a mixture of two normal distributions, one centered at $(-\frac{3}{2},0)$ and one centered at $(\frac{3}{2},0)$. Its probability density function is

$$p(x,y) = \tfrac{1}{2}p_0(x-\tfrac{3}{2},y) + \tfrac{1}{2}p_0(x+\tfrac{3}{2},y),$$

where p_0 is defined above, as the probability density function of the standard multivariate normal distribution.

Geodesic balls under the metric with this more complicated probability distribution are shown in Figure 7.4. Again, the rings of constant radii are biased towards regions of higher probability. Figure 7.5 provides another way to visualize the space; it depicts a selection of geodesics of the same length, starting from the point $(-2,-2)$. Recall that geodesics in the usual Euclidean plane are simply straight lines; in contrast, these geodesics tend to be deflected towards the regions of high probability. This metric therefore clearly satisfies the property that we desire.

Although promising, this metric is not particularly natural, especially because α is not defined. It is clear we need to develop more criteria for a natural metric.

7.2 Invariance under transformations

One important property we would like our metric to have is that geodesics should be invariant under transformations. In other words, we would like distances to in some sense remain the same even if we transform the

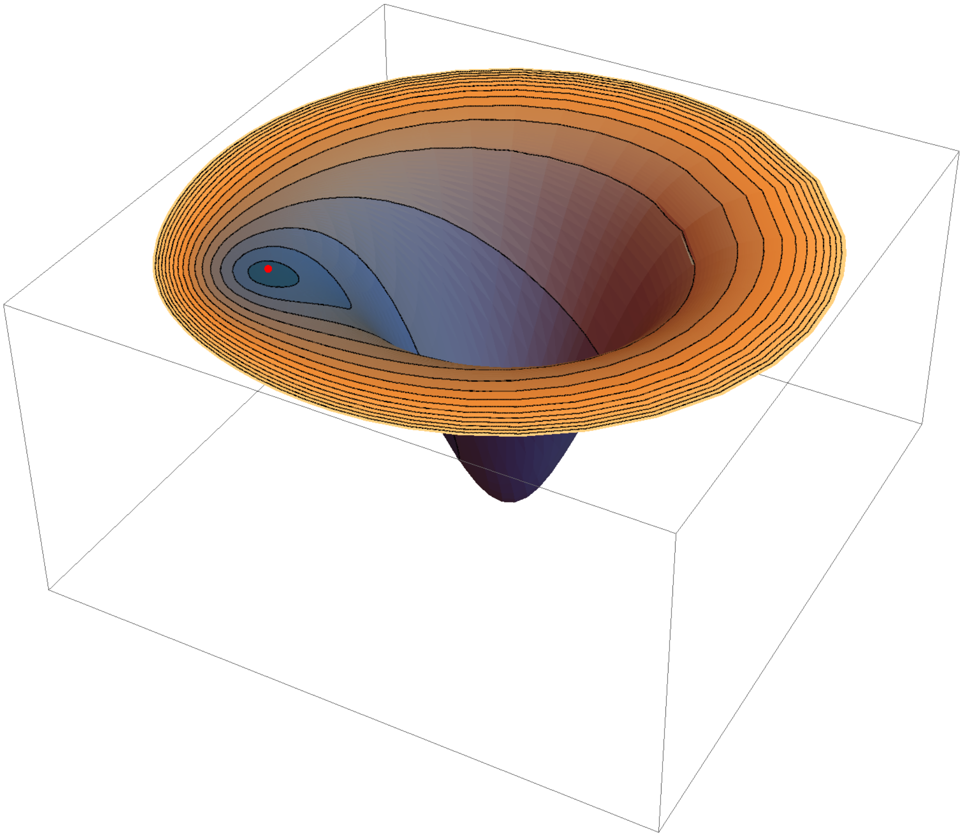


Figure 7.3 An alternative visualization of Figure 7.2. Geodesic balls of various radii around the red point are "attracted" towards the low points on the surface of negative probability.

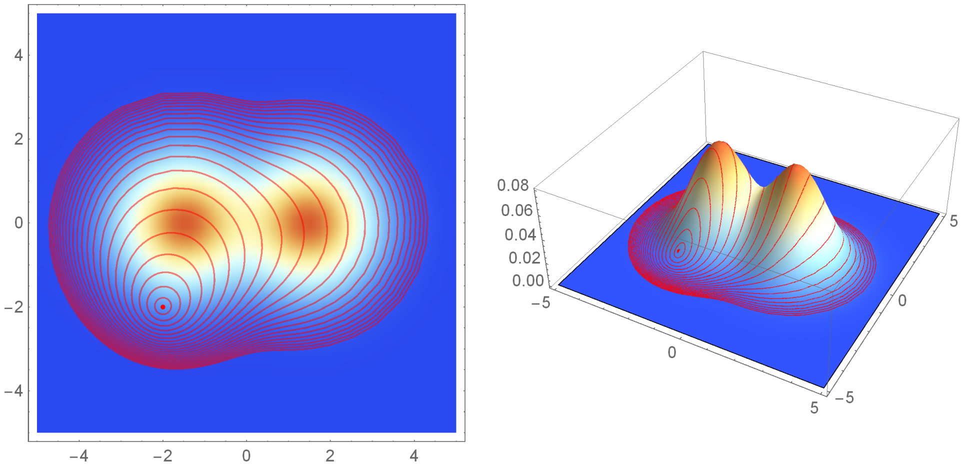


Figure 7.4 The contours represent points equidistant from $(-2,-2)$, under the scaled Euclidean metric $(\alpha = 1)$ with a mixture of two normal distributions centered at $(\frac{3}{2},0)$ and $(-\frac{3}{2},0)$.

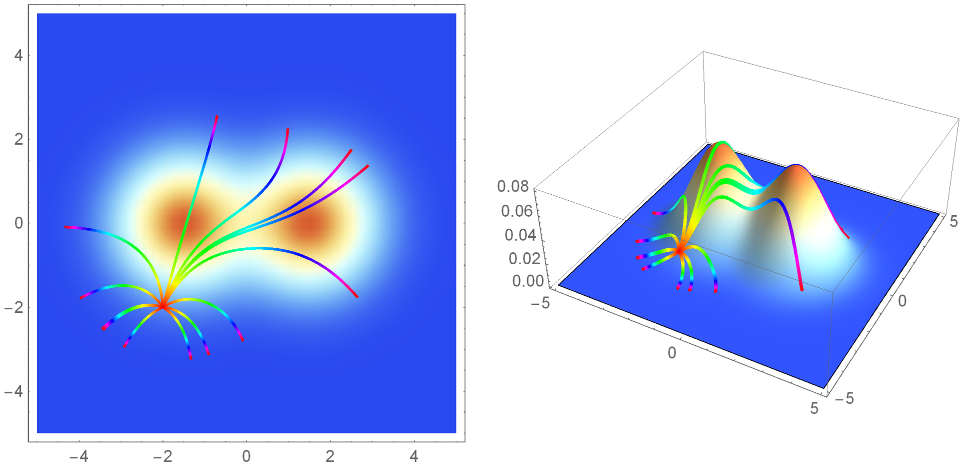


Figure 7.5 A selection of geodesics of the same length from $(-2,-2)$, when measured using the scaled Euclidean metric $(\alpha = 1)$.

probability distribution. As a basic example, suppose we are talking about a dataset of houses, with each house associated with information about how large the property is. Then we would expect the fact that point a to be closer to b than c not to change if the area is measured in square feet instead of square meters.

Concretely, suppose f is a diffeomorphism $\Sigma \to \Sigma'$, so that the probability density function $\bar{p} : \Sigma' \to \mathbb{R}$ induced by $p : \Sigma \to \mathbb{R}$ is

$$\bar{p} = \frac{p \circ f^{-1}}{|\det \mathrm{D} f|}. \tag{7.3}$$

Then if γ is a geodesic on Σ with respect to the metric induced by p, then $f \circ \gamma$ should be a geodesic on Σ' with respect to the metric induced by $\bar{p}$.

A justification for this requirement comes from an algebraic viewpoint. Suppose we consider two probability spaces to be isomorphic if there exists a diffeomorphism f between them such that their probability density functions are related as in Equation 7.3. Two Riemannian manifolds are isomorphic if there exists an isometry between them. Therefore, an object that combines the properties of a probability space and a Riemannian manifold—as we are trying to define—should have isomorphisms that are isomorphisms in *both* senses.

One easy way for this requirement to be satisfied is if we force the transformation f to be an **isometry**: that is,

$$\langle u, v \rangle_x = \langle \mathrm{D} f_x\, u, \mathrm{D} f_x\, v \rangle_{f(x)}$$

for all $x \in \Sigma$ and $u, v \in T_x\Sigma$.

Let $g[p]$ denote the matrix of the Riemannian metric induced by the probability distribution p, so that we can express this condition on the Riemannian metric in matrix form as

$$u^T g[p]_x v = u^T\, \mathrm{D} f_x^T\, g[\bar{p}]_{f(x)} \mathrm{D} f_x\, v.$$

Since this is true for all u and v, it suffices to write the condition as

$$g[p]_x = \mathrm{D} f_x^T\, g[\bar{p}]_{f(x)} \mathrm{D} f_x. \tag{7.4}$$

7.2.1 Testing the scaled Euclidean metric

Let us test the scaled Euclidean metric that we defined. If we plug in Equation 7.2 for g, then we find that

$$\frac{1}{p(x)^\alpha} I_n = \frac{1}{\bar{p}(f(x))^\alpha}\, \mathrm{D} f_x^T \mathrm{D} f_x,$$

or

$$I_n = |\det \mathrm{D}f_x|^{\alpha}\, \mathrm{D}f_x^T \mathrm{D}f_x.$$

If we take the determinant of both sides, we find that this implies that

$$1 = |\det \mathrm{D}f_x|^{\alpha n} (\det \mathrm{D}f_x)^2,$$

which is the case if $|\det \mathrm{D}f_x| = 1$ for all $x \in \Sigma$, or if $\alpha = -\frac{2}{n}$. Thus, isometries under the scaled Euclidean metric for arbitrary α are volume-preserving maps with orthogonal Jacobian; orthogonal linear maps $f \in O(n)$ are one possibility, although not particularly interesting.

Alternatively, if $\alpha = -\frac{2}{n}$, then all diffeomorphisms are isometries. We had previously constrained $\alpha > 0$ because this ensures the behavior of geodesics we desire, but it may be worth investigating this case as well. In this case, we have the Riemannian metric

$$\langle u, v \rangle_x = p(x)^{2/n} (u \cdot v),$$

or

$$g[p]_x = p(x)^{2/n} I_n.$$

It turns out that this metric has a nice property: the volume measure induced by the Riemannian metric is the probability density. We see this with a simple calculation:

$$\sqrt{\det g[p]}\, \mathrm{d}V = \sqrt{\det p^{2/n} I_n}\, \mathrm{d}V = \sqrt{p^2}\, \mathrm{d}V = p\, \mathrm{d}V.$$

This property, along with the invariance under diffeomorphisms, makes this metric an attractive one. Future research is necessary to determine when this metric is useful.

7.2.2 An invariant metric capturing the derivative of p

Returning to Equation 7.4, we have

$$g[p]_x = \mathrm{D}f_x^T g[\bar{p}]_{f(x)} \mathrm{D}f_x.$$

Suppose we enforce that f have an orthogonal Jacobian, so that $|\det \mathrm{D}f| = 1$, and $\bar{p} = p \circ f^{-1}$. One ansatz we might make is that

$$g[p] = \mathrm{D}p^T \,\mathrm{D}p,$$

where in this case D is the gradient operator. Note that this g is not necessarily positive-definite and thus not a Riemannian metric, but we proceed anyway. Equation 7.4 becomes

$$\mathrm{D}p_x^T\,\mathrm{D}p_x = \mathrm{D}f_x^T\,\mathrm{D}(p\circ f^{-1})_{f(x)}^T\,\mathrm{D}(p\circ f^{-1})_{f(x)}\mathrm{D}f_x,$$

which, by the chain rule, becomes

$$\mathrm{D}p_x^T\,\mathrm{D}p_x = \mathrm{D}f_x^T(\mathrm{D}f^{-1})_{f(x)}^T\,\mathrm{D}p_x^T\,\mathrm{D}p_x(\mathrm{D}f^{-1})_{f(x)}\mathrm{D}f_x,$$

which is true, meaning that the $\mathrm{D}p^T\,\mathrm{D}p$ is invariant, if only it were a metric.

In fact, similar logic says that

$$g[p] = C(\mathrm{D}p^T\,\mathrm{D}p)^k$$

is invariant as well, for any integer $k \geq 0$ and C not a function of p. Thus,

$$g[p] = \exp(\mathrm{D}p^T\,\mathrm{D}p)$$

is a true Riemannian metric, as the matrix exponential of a symmetric matrix is symmetric and positive-definite. The properties of this metric should be the subject of further study.

7.3 Pullback from unit hypercube

Another easy way to satisfy the property of invariance under transformations is as follows. Suppose we specify some standard Riemannian manifold Σ_0. Let $p : \Sigma \to \mathbb{R}$ be a probability density function, and let $f_\Sigma : \Sigma \to \Sigma_0$ be a diffeomorphism constructed in a standard way from p. Then we can use f_Σ to pullback the metric from Σ_0 to Σ, i.e.

$$\langle u, v\rangle_\Sigma = \langle \mathrm{D}f\,u, \mathrm{D}f\,v\rangle_{\Sigma_0}.$$

This way, f_Σ is an isometry automatically, and moreover, for any space Σ', $f_{\Sigma'}$ is an isometry. Thus, we can have $\Sigma \cong \Sigma'$ via $\Sigma \cong \Sigma_0 \cong \Sigma'$.

For simplicity, we can specify that the standard space Σ_0 be the unit hypercube $[0,1]^n$ with the Euclidean metric, and let us enforce that the diffeomorphism f_Σ is defined in such a way that the induced probability distribution

$$\bar{p} = \frac{p\circ f^{-1}}{|\det \mathrm{D}f|}$$

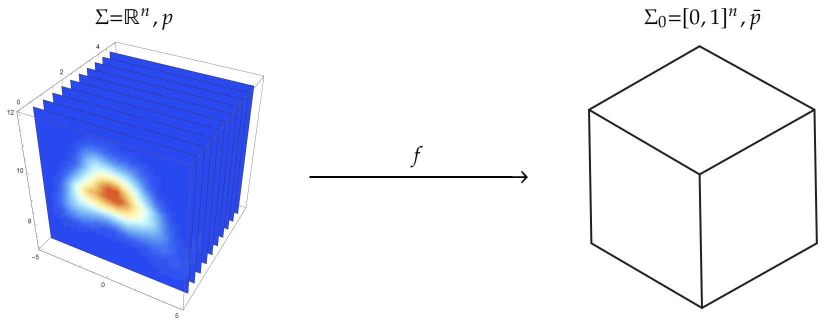


Figure 7.6 Suppose that f induces a uniform distribution on Σ_0. The Euclidean metric on Σ_0 induces a Riemannian metric on Σ via f.

on the hypercube is uniform. This makes sense because in the absence of information (as is the case if the hypercube is uniformly distributed), then the straight-line Euclidean distance is as good as any. This method of defining a metric is illustrated in Figure 7.6.

7.3.1 Deep density models

Rippel and Adams (2013), as described in Subsection 6.3.3, describe *deep density models*, a way to infer a diffeomorphism $f : \Sigma \to [0,1]^n$ from Σ to a unit hypercube such that the unit hypercube has a prescribed probability distribution. If we prescribe a uniform distribution for the unit hypercube, then the standard Euclidean metric is a natural choice for a metric on $[0,1]^n$. The metric is then defined as a pullback of the Euclidean metric under f. This metric may very well be the best metric defined so far. However, due to difficulty of implementing deep density models, it is perhaps worthwhile to attempt to define a simpler way to construct the diffeomorphisms.

7.3.2 Cumulative distribution functions

One simpler way to define a transformation from Σ to $[0,1]^n$, the unit hypercube of n dimensions, is using a high-dimensional analogue of cumulative distribution functions.

To see how, consider the one-dimensional case, where we would like a function from $\mathbb{R}$ to $[0,1]$ such that the probability distribution induced on

$[0,1]$ from $\mathbb{R}$ is uniform. The cumulative distribution function $F : \mathbb{R} \to [0,1]$

$$F(x) = \int_{-\infty}^{x} p(\xi)\,\mathrm{d}\xi$$

does the job. In this case, $F(x)$ represents the percentage of the total probability mass encountered in this axis up to x.

Now we generalize to n dimensions. Let $p : \mathbb{R}^n \to \mathbb{R}$ be a probability density function. Then the desired function $F : \mathbb{R}^n \to [0,1]^n$ is

$$F(x_1, \ldots, x_n) = \begin{bmatrix} \dfrac{\int_{-\infty}^{x_1} p(\xi, x_2, \ldots, x_n)\,\mathrm{d}\xi}{\int_{-\infty}^{\infty} p(\xi, x_2, \ldots, x_n)\,\mathrm{d}\xi} \\ \vdots \\ \dfrac{\int_{-\infty}^{x_n} p(x_1, x_2, \ldots, \xi)\,\mathrm{d}\xi}{\int_{-\infty}^{\infty} p(x_1, x_2, \ldots, \xi)\,\mathrm{d}\xi} \end{bmatrix},$$

where we have the same interpretation as in the one-dimensional case: the ith component of $F(x)$ represents the percentage of the total probability mass encountered in the ith axis in $\mathbb{R}^n$ up to x. With this, we can pullback the Euclidean metric from the unit hypercube as before.

This metric satisfies our requirement that geodesics travel in regions of high probability in simple cases, but does not necessarily to so in more complicated probability distributions. For example, if the low-density regions of a multimodal distribution are not exactly zero, the geodesics may cross through the low-density region.

7.4 Other approaches

7.4.1 Maximum likelihood curves

Another approach is to attempt to have geodesics directly be the curve of "maximum likelihood." To do this, we must define the likelihood of a curve. Note that the likelihood of a set of points is simply the product of their probabilities; in other words, the log-likelihood of a set of points is simply the sum of their log probabilities. Therefore, an initial conjecture for the log-likelihood of a curve $\gamma : I \to \Sigma$ might be a continuous generalization of the sum of log-probabilities:

$$\ell(\gamma) = \frac{1}{|I|} \int_I \log p(\gamma(t))\,\mathrm{d}t.$$

It is however unclear how to extract a Riemannian metric from this definition. More research in this direction is necessary.[1]

7.4.2 Prescribed scalar curvature

An entirely different approach is as follows. Recall that the **scalar curvature** $K : M \to \mathbb{R}$ defined on a manifold M is

$$K(x) = \frac{1}{n(n-1)} \sum_{i,j} \langle R(\partial_i, \partial_j)\partial_i, \partial_j \rangle,$$

where $\partial_1, \ldots, \partial_n$ is an orthonormal basis for the tangent space $T_x M$, and R is the Riemann curvature tensor. The scalar curvature describes how much the volume of an infinitesimal ball differs from the usual Euclidean volume at each point $x \in M$.

This concept is useful in our problem: we would like a Riemannian metric in which the infinitesimal volume is somehow dependent on the probability density function $p(x)$ at that point. The problem of determining a Riemannian metric from a prescribed scalar curvature is well-studied: Kazdan and Warner (1975) provide classical results in the subject, while Rosenberg (2007) provides a review of some modern results. The details of such a prescription for our problem has yet to be studied.

7.5 Summary of desired properties

Let us summarize the properties that the metric should satisfy:

1. Geodesics should lie "along the data manifold" and avoid regions where there is no data. The metric tensor should thus be lower in regions of high density, and higher in regions of low density.

2. Geodesics should be invariant under transformations of the data. Suppose f is a transformation $\Sigma \to \Sigma'$, so that the induced probability distribution on Σ' is
$$\bar{p} = \frac{p \circ f^{-1}}{|\det \mathrm{D} f|}.$$
Then if γ is a geodesic on Σ with respect to the metric induced by p, then $f \circ \gamma$ should be a geodesic on Σ' with respect to the metric induced by $\bar{p}$.

[1] Perhaps some inspiration could be drawn from the path integral formulation of quantum mechanics, to which this approach bears some superficial resemblance.

3. Distance on a space containing uniformly-distributed data should correspond to Euclidean distance.

7.6 Applications of a Riemannian metric

There are several potential applications to a Riemannian metric on the data space Σ; two are listed here.

First, geodesics on the space would allow for smooth interpolation between two high-dimensional points. This may have applications in computer vision, where smooth animations between images can be constructed by traveling along a geodesic between the two images.

Second, the metric we define has the potential to revolutionize machine learning techniques such as dimensional reduction and clustering by providing a more accurate measure of distance in data spaces than the Euclidean distance prevalent today.

Of course, the generality of such a notion of distance makes it likely that it will find a myriad of applications in unexpected domains.

Chapter 8

Background: statistical manifolds

We have been tackling the problem of turning the space that the data lives in into a Riemannian manifold. However, much work has gone into a related but different problem: transforming spaces of *probability distributions* into Riemannian manifolds. These manifolds are useful when trying to draw conclusions *from* data and thus may prove relevant to our research problem. In this chapter, we describe the basic results of this field, known as *information geometry*. This chapter follows the presentation of Amari and Nagaoka (2007), the best-known monograph on the subject.

Information geometry is the study of *statistical manifolds*, which allow us to consider all possible probability distributions on a sample space Σ as one object. Each point of such a manifold is a probability distribution:

Definition 20 (Statistical manifold)**.** *A **statistical manifold** M is a differentiable manifold of probability distribution functions $p_\theta : \Sigma \to \mathbb{R}$ with parameter $\theta \in \Theta$,*

$$M = \{p_\theta \equiv p(x;\theta) \mid \theta \in \Theta \subseteq \mathbb{R}^k\}.$$

This is related to the maximum likelihood estimation problem as described in Subsection 6.3.1: given a set of observations $\{x_1, \ldots, x_m\} \subseteq \Sigma$ and a family of distributions $\{p_\theta \mid \theta \in \Theta\}$, we seek the parameter $\hat{\theta} \in \Theta$ that most likely generated the observed data. Rephrased in the language of information geometry, this process is optimization over the statistical manifold M, seeking an optimal point $p^* \in M$.

We will eventually define a Riemannian metric as well as several affine connections on a statistical manifold M in pursuit of a solution to this problem.

8.1 Dual connections, divergence, and the projection theorem

We begin by defining some concepts in Riemannian geometry.

First, we generalize the notion of a metric connection on a Riemannian manifold M. Recall that an affine connection ∇ is *metric* if it satisfies

$$Z\langle X, Y\rangle = \langle \nabla_Z X, Y\rangle + \langle X, \nabla_Z Y\rangle$$

for all vector fields X, Y, and Z. The most commonly defined connections on Riemannian manifolds are metric; however, in information geometry, non-metric connections are actually quite useful. Towards this end, we define the notion of a dual connection.

Definition 21 (Dual connection)**.** *Let M be a Riemannian manifold with an affine connection ∇. Then its **dual connection** ∇^* is the affine connection that uniquely satisfies*

$$Z\langle X, Y\rangle = \langle \nabla_Z X, Y\rangle + \langle X, \nabla^*_Z Y\rangle$$

for all vector fields X, Y, and Z.

Notice that the condition that $\nabla = \nabla^*$ implies that ∇ is metric. To build intuition, consider the following behavior of parallel transport with respect to a connection and its dual.

Theorem 4. *Let Π and Π^* be the parallel transport along a curve γ with respect to ∇ and ∇^* respectively. Then for all vector fields X and Y,*

$$\langle \Pi X, \Pi^* Y\rangle = \langle X, Y\rangle.$$

Thus, if we parallel transport X under ∇, we must parallel transport Y under ∇^* to preserve the angle between them.

Now let us define flatness with respect to a connection. Consider:

Definition 22 (Affine coordinate system)**.** *Let M be a differentiable manifold with an affine connection ∇. Then $[\xi^i]$ is an **affine coordinate system** for ∇ if*

$$\nabla_{\partial_i}\partial_j = 0,$$

where $[\partial_i]$ is the natural basis of the tangent space for $[\xi^i]$.

Definition 23 (∇-flat)**.** *Let M be a differentiable manifold with an affine connection ∇. Then M is ∇-flat if there exists* ***affine coordinate system*** *for ∇.*

The intuition for flatness is that Euclidean space is flat with respect to its standard coordinates. There are several interesting properties of flat manifolds. For example, for any flat manifold, the curvature and torsion tensors are both $R = T = 0$. Additionally, the parallel transport of a vector between p and q does not depend on the curve used to connect them.

Now we define the notion of a divergence.

Definition 24 (Divergence)**.** *Let M be a differentiable manifold, and let $D(\cdot \,||\, \cdot) : M \times M \to \mathbb{R}$ be a smooth function satisfying $D(p \,||\, q) \geq 0$ and $D(p \,||\, q) = 0$ iff $p = q$ for all $p, q \in S$. Then D is a* ***divergence*** *if*

$$\langle X, Y \rangle^{(D)} = -D(X \,||\, Y)$$

is a Riemannian metric.

Some explanation of the notation is necessary. Suppose $\frac{\partial}{\partial \xi_i}$ and $\frac{\partial}{\partial \xi_j}$ are elements of T_pM. Then we define

$$D(\tfrac{\partial}{\partial \xi_i} \,||\, \tfrac{\partial}{\partial \xi_j}) = \frac{\partial}{\partial \xi^i} \frac{\partial}{\partial \xi'^j} D(p(\xi^1, \ldots, \xi^n) \,||\, p(\xi'^1, \ldots, \xi'^n)).$$

The most important point is that the divergence behaves almost like a distance metric between two points on a manifold, except it is not necessarily symmetric and does not satisfy the triangle inequality. A divergence on M not only induces a Riemannian metric on M; it also induces the divergence, as below.

Definition 25 (Affine connection induced by a divergence)**.** *The* ***affine connection*** *$\nabla^{(D)}$* ***induced by a divergence*** *D on M is*

$$\langle \nabla_X^{(D)} Y, Z \rangle^{(D)} = -D(XY \,||\, Z).$$

We define the dual divergence as follows:

Definition 26 (Dual divergence)**.** *Let D be a divergence. Its* ***dual divergence*** *D^* is defined as*

$$D^*(p \,||\, q) = D(q \,||\, p).$$

The canonical divergence is then induced by the Riemannian metric, the affine connection, and its dual, via the following definition.

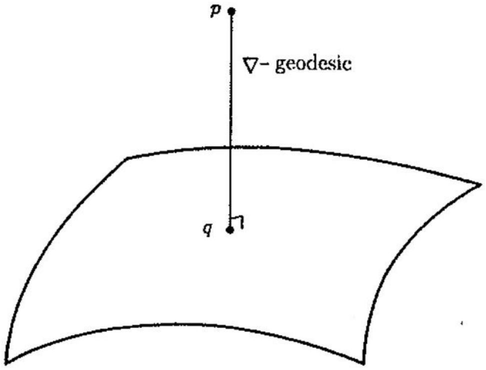


Figure 8.1* The projection theorem on Riemannian manifolds illustrated.

Definition 27 (Canonical divergence). *Let M be a Riemannian manifold with the symmetric affine connection ∇, and suppose that M is both ∇-flat and ∇^*-flat. The **canonical divergence** is the unique divergence that induces the Riemannian metric of M, $\nabla^{(D)} = \nabla$, and $\nabla^{(D^*)} = \nabla^*$.*

The canonical divergence allows a generalization of the projection theorem to hold:

Theorem 5 (Projection). *Let M be a Riemannian manifold with metric g, and let S be a submanifold of M. Let D be the canonical divergence with respect to g, ∇, and ∇^*. Then $q \in S$ is a stationary point of $D(p \,||\, \cdot)$ for $p \in M$ if and only if the ∇-geodesic connecting p and q is orthogonal to S at q.*

That is, if we want to minimize the canonical divergence between a point $p \in M$ and a submanifold $S \subseteq M$, then we simply project with the ∇-geodesic that is orthogonal to S. A visualization of the projection theorem is depicted in Figure 8.1. This theorem proves very useful in working with statistical manifolds.

8.2 The Fisher information metric

We are now ready to define a Riemannian metric called the *Fisher information metric* on a statistical manifold M. To motivate its definition, recall that maximum likelihood estimation maximizes the log-likelihood

$$\ell(\theta; x_1, \ldots, x_m) \equiv \log L(\theta; x_1, \ldots, x_m) = \sum_{i=1}^{m} \log p(x_i; \theta),$$

of a set of observations $\{x_1, \ldots, x_m\}$. We assume that Θ is one-dimensional. To find the maximum, we differentiate the log-likelihood and set it to 0:

$$0 = \frac{\partial \ell}{\partial \theta}.$$

Then, we solve for θ to find the parameter that most likely generated the observations we observed. Once we have a maximum likelihood estimator $\hat{\theta}$, we might be interested in *how* optimal this estimate is—how much can we trust this estimate? In the following, we will quantify how optimal a given maximum likelihood estimator is. The resulting quantity is called the *Fisher information* $I : \Theta \to \mathbb{R}$.

Since a measure of how optimal θ is is only relevant if θ is indeed a maximum, when interpreting the quantity $I(\theta)$ we will assume in the following that θ does indeed maximize the log-likelihood; in other words, we will assume that Σ does have the PDF $p(x; \theta)$ for a fixed θ.

In this case, θ is a local maximum of the log-likelihood function. If the log-likelihood function ℓ is sharply peaked around θ, then the values surrounding θ are extremely unlikely compared to θ, in which case θ is an excellent estimate. By contrast, if the log-likelihood function is relatively flat around θ, then surrounding parameters are less likely than θ, but still comparatively likely. In this case θ, is a poor estimator.

In calculus, the second derivative gives a measure of how sharply a function is curving; therefore, the second derivative of the log-likelihood function will be a good measure of how sharply peaked the log-likelihood is. Thus, we define the Fisher information of m points to be

$$I(\theta; x_1, \ldots, x_m) \equiv -\frac{\partial^2 \ell}{\partial \theta^2} = -\sum_{i=1}^{m} \frac{\partial^2}{\partial \theta^2} \log p(x_i; \theta),$$

where the minus sign is a convention to ensure that $I(\theta) \geq 0$ for an maximum θ. The better the estimator θ is, the greater $I(\theta)$ is.

Now we move to the limit where the number of observations $m \to \infty$. In this limit, by the law of large numbers,

$$\frac{1}{m} I(\theta; x_1, \ldots, x_m) \to -\mathbb{E}\left[\frac{\partial^2}{\partial \theta^2} \log p(X; \theta)\right],$$

where $\mathbb{E}$ denotes the expectation value with respect to X, a random variable with values in Σ distributed according to the probability density function $p(x;\theta)$. With this in mind, we define the Fisher information as follows:

Definition 28 (Fisher information)**.** *The* ***Fisher information*** $I : \Theta \to \mathbb{R}$ *is*

$$I(\theta) \equiv -\mathbb{E}\left[\frac{\partial^2}{\partial\theta^2}\log p(X;\theta)\right] = -\int p(x;\theta)\,\frac{\partial^2}{\partial\theta^2}\log p(x;\theta)\,\mathrm{d}x.$$

It measures how sharply the log-likelihood would peaked at the parameter θ, if θ is the true parameter, in the limit of an infinite number of observations.

It is useful to derive an alternate expression for the Fisher information. The **efficient score** $V(\theta;x)$ is defined as

$$V(\theta;x) = \frac{\partial}{\partial\theta}\log p(x;\theta).$$

Recall that the maximum likelihood condition was setting

$$0 = \frac{\partial\ell}{\partial\theta} = \sum_{i=1}^{n} V(\theta;x_i).$$

Therefore, it is not surprising that if θ is the maximum likelihood estimator, then

$$\begin{aligned}
\mathbb{E}[V(\theta;X)] &= \int V(\theta;x)\,p(x;\theta)\,\mathrm{d}x \\
&= \int\left[\frac{\partial}{\partial\theta}\log p(x;\theta)\right]p(x;\theta)\,\mathrm{d}x \\
&= \int \frac{\frac{\partial}{\partial\theta}p(x;\theta)}{p(x;\theta)}\cdot p(x;\theta)\,\mathrm{d}x \\
&= \int \frac{\partial}{\partial\theta}\,p(x;\theta)\,\mathrm{d}x \\
&= \frac{\partial}{\partial\theta}\int p(x;\theta)\,\mathrm{d}x \\
&= \frac{\partial}{\partial\theta}\,1 \\
&= 0.
\end{aligned}$$

Now differentiate both sides of

$$\mathbb{E}[V(\theta;X)] = \int V(\theta;x)\,p(x;\theta)\,\mathrm{d}x = 0$$

with respect to θ to get

$$\int \left[\frac{\partial}{\partial\theta}\,V(\theta;x)\right] p(x;\theta)\,\mathrm{d}x + \int V(\theta;x)\left[\frac{\partial}{\partial\theta}\,p(x;\theta)\right]\mathrm{d}x = 0.$$

The first term is

$$\begin{aligned}\int \left[\frac{\partial}{\partial\theta}\,V(\theta;x)\right] p(x;\theta)\,\mathrm{d}x &= \mathbb{E}\left[\frac{\partial^2}{\partial\theta^2}\log p(X;\theta)\right]\\ &= -I(\theta).\end{aligned}$$

The second term is

$$\begin{aligned}\int V(\theta;x)\left[\frac{\partial}{\partial\theta}\,p(x;\theta)\right]\mathrm{d}x &= \int V(\theta;x)\left[\frac{\partial}{\partial\theta}\log p(x;\theta)\right] p(x;\theta)\,\mathrm{d}x\\ &= \int \big(V(\theta;x)\big)^2\,p(x;\theta)\,\mathrm{d}x\\ &= \mathbb{E}\left[\big(V(\theta;X)\big)^2\right].\end{aligned}$$

Putting these two together, we have an alternate expression for the Fisher information:

$$I(\theta) = \mathbb{E}\left[\big(V(\theta;X)\big)^2\right] = \mathbb{E}\left[\left(\frac{\partial}{\partial\theta}\log p(X;\theta)\right)^2\right].$$

Additionally, because $\mathbb{E}[V(\theta;X)] = 0$, we also have

$$I(\theta) = \mathrm{Var}(V(\theta;X)),$$

where Var denotes the variance with respect to X.

This expression is best for generalization to where there are k parameters $(\theta_1,\ldots,\theta_k)$ instead of just one. Note that in general, we will abuse notation and let $\theta = (\theta_1,\ldots,\theta_k)$ represent the whole list of parameters. In this case, we can look at the mixed derivatives and define the matrix g as

$$g_{ij} \equiv -\mathbb{E}\left[\frac{\partial^2}{\partial\theta_i\partial\theta_j}\log p(X;\theta)\right] = \mathbb{E}\left[\left(\frac{\partial}{\partial\theta_i}\log p(X;\theta)\right)\left(\frac{\partial}{\partial\theta_j}\log p(X;\theta)\right)\right]$$

or, in terms of efficient scores,

$$g_{ij} \equiv -\mathbb{E}\left[\frac{\partial V_i}{\partial \theta_j}\right] = \mathbb{E}\left[V_i(\theta; X)\, V_j(\theta; X)\right],$$

with

$$V_i(\theta; x) \equiv \frac{\partial}{\partial \theta_i} \log p(x; \theta).$$

It is also the covariance of the efficient scores

$$g \equiv \operatorname{Cov}\big(V_1(\theta; X), \ldots, V_k(\theta; X)\big).$$

It turns out that this matrix can serve as a Riemannian metric on a statistical manifold M:

Definition 29 (Fisher information metric)**.** *The* ***Fisher information metric*** *is a metric on a statistical manifold M given by $G = [g_{ij}]$, where*

$$g_{ij} \equiv -\mathbb{E}\left[\frac{\partial^2}{\partial \theta_i \partial \theta_j} \log p(X; \theta)\right] = \mathbb{E}\left[\left(\frac{\partial}{\partial \theta_i} \log p(X; \theta)\right)\left(\frac{\partial}{\partial \theta_j} \log p(X; \theta)\right)\right].$$

Statistical manifolds are naturally equipped with the Fisher information metric, which is a natural metric as it is invariant to reparameterizations of θ.

8.3 The α-connection and Kullback-Leibler divergence

Now let us define a useful connection on statistical manifolds, the α-connection.

Definition 30 (α-connection)**.** *The* ***α-connection*** $\nabla^{(\alpha)}$ *on a statistical manifold M is given by the Christoffel symbols*

$$\Gamma^{(\alpha)}_{ij,k} = \mathbb{E}[(\partial_i \partial_j \ell_\theta + \tfrac{1-\alpha}{2} \partial_i \ell_\theta \partial_j \ell_\theta)(\partial_k \ell_\theta)],$$

where $\ell_\theta = \log p(x; \theta)$ and $\partial_i = \frac{\partial}{\partial \theta_i}$.

It is also invariant under reparameterization and enjoys special duality properties:

Theorem 6. *The connections $\nabla^{(\alpha)}$ and $\nabla^{(-\alpha)}$ are dual connections of each other.*

Theorem 7. *M is $\nabla^{(\alpha)}$-flat if and only if M is $\nabla^{(-\alpha)}$-flat.*

We also define the Kullback-Leibler divergence:

Definition 31 (Kullback-Leibler divergence)**.** *The* ***Kullback-Leibler divergence*** *$D^{(-1)} : M \times M \to \mathbb{R}$ on a statistical manifold M is*

$$D^{(-1)}(p \,||\, q) = \int p \log \frac{p}{q} \, \mathrm{d}x.$$

The Kullback-Leibler divergence is used in probability theory to measure how different two probability distributions p and q on the same space are and has an interpretation in terms of the entropy of the two distributions. What is important for our purposes, however, is the following remarkable theorem.

Theorem 8. *The Kullback-Leibler divergence is the canonical divergence with respect to the Fisher information metric and the ∓ 1-connection.*

This means that we can use the projection theorem as stated in the first section to minimize the Kullback-Leibler divergence.

8.4 The manifold of normal distributions

Now let us work through an example of a statistical manifold.

8.4.1 An application of the Levi-Civita connection

Recall that a normal distribution with mean μ and variance σ^2 is defined by the probability distribution function

$$p(x; \mu, \sigma^2) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right).$$

We can therefore view the space of normal distributions as a two-dimensional manifold $\mathcal{N}$, parameterized by μ and $\sigma > 0$. Moreover, the Fisher information metric defines a natural metric on this space with $\theta_1 = \mu$ and $\theta_2 = \sigma$.

To simplify calculations, we will actually take $\theta_1 = \mu$ and $\theta_2 = \sqrt{2}\sigma$. The probability distribution function becomes

$$p(x; \theta_1, \theta_2) = \frac{1}{\sqrt{\pi}\theta_2} \exp\left(-\frac{(x-\theta_1)^2}{\theta_2^2}\right).$$

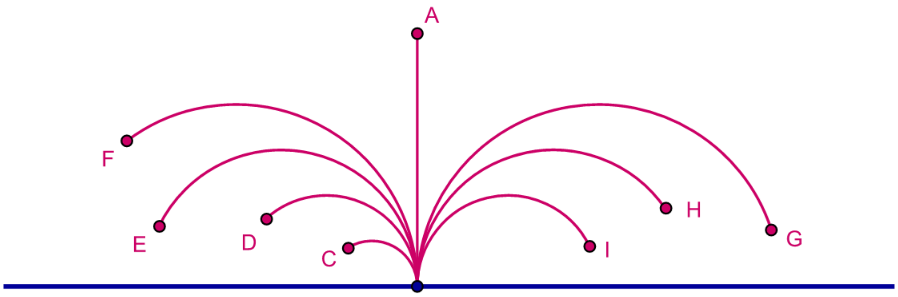


Figure 8.2* Geodesics in the Poincaré half-plane model.

Now we calculate the Fisher information metric from the expression

$$g_{ij} \equiv -\mathbb{E}\left[\frac{\partial^2}{\partial\theta_i\partial\theta_j}\log p(X;\theta)\right],$$

which we find to be

$$g_{11} = g_{22} = \frac{2}{\theta_2^2} \qquad g_{12} = g_{21} = 0.$$

Incidentally, this is a well-known situation in non-Euclidean geometry. The *Poincaré half-plane model* is the upper half-plane

$$\mathbb{H}^2 = \{(x_1, x_2) \in \mathbb{R}^2 \mid x_2 > 0\},$$

with the metric

$$g_{11} = g_{22} = \frac{1}{x_2^2} \qquad g_{12} = g_{21} = 0.$$

Compare this to the current situation: the manifold of normal distributions is

$$\mathcal{N} = \{(\theta_1, \theta_2) \in \mathbb{R}^2 \mid \theta_2 > 0\}$$

with the metric

$$g_{11} = g_{22} = \frac{2}{\theta_2^2} \qquad g_{12} = g_{21} = 0.$$

Since a scaling by 2 of metric does not affect geodesics, the geodesics in $\mathcal{N}$ with respect to the Levi-Civita (metric) connection, when parameterized by $\theta_1 = \mu$ and $\theta_2 = \sqrt{2}\sigma$, are the same as those of the half-plane model.

The geodesics of the half-plain model can be calculated by solving the geodesic equations

$$\ddot{x}_i + \sum_{j,k} \Gamma^i_{jk} \dot{x}_j \dot{x}_k = 0,$$

where the Γ^i_{jk} are the Christoffel symbols

$$\Gamma^2_{11} = \frac{1}{x_2} \qquad \Gamma^2_{22} = -\frac{1}{x_2} \qquad \Gamma^1_{12} = \Gamma^1_{21} = -\frac{1}{x_2}.$$

The solution is well known to be the semicircles

$$x_1(t) = c - r \tanh t \qquad x_2(t) = r \operatorname{sech} t.$$

and vertical lines

$$x_1(t) = c \qquad x_2(t) = e^t,$$

for $c, r \in \mathbb{R}$ with $r > 0$, as depicted in Figure 8.2. The geodesics in $\mathcal{N}$, therefore, are also semicircles centered on the μ-axis.

One neat application of these metric geodesics is the ability to interpolate normal distributions. That is, suppose we are given two normal distributions $(\mu_1, \sqrt{2}\sigma_1)$ and $(\mu_2, \sqrt{2}\sigma_2)$. To find the "average" normal distribution, we simply connect the two corresponding points in the θ_1-θ_2 plane with an arc of a semicircle, and find the point equidistant to both endpoints, according to the Fisher metric. This middle point is the "average" normal distribution. This is illustrated in Figure 8.3.

Note that this notion of average fits our expectation of what an average of probability distributions would be. With this average, the average of two normal distributions with the same variance σ but centered at μ_1 and μ_2 respectively is not simply a normal distribution with the same variance

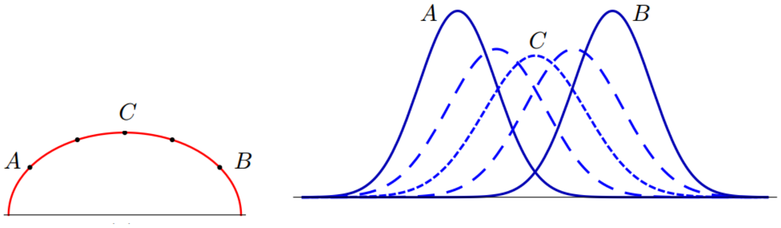


Figure 8.3* Normal distributions along a geodesic. The average of A and B is C.

centered at the arithmetic average between μ_1 and μ_2. Instead, the average of the two is a distribution that is much broader, covering both original distributions.

8.4.2 An application of the ±1-connection

A common problem in statistics is to devise good *estimators*. Suppose that we sample a point $x \in \Sigma$ from a probability distribution function $p(x;\theta)$ with an unknown parameter θ that we would like to determine. We define an estimator $\hat{\theta}(x)$ that is a function of the observation x that we would like to use to approximate the true value θ. The maximum likelihood estimator is one such estimator. Note that sampling m points fromΣ constitutes sampling one point from the m-fold Cartesian product Σ^m, so it suffices to consider sampling only a single point. Let us briefly consider some theory of estimators.

Ideally, an estimator, however we choose to define it, will be *unbiased*:

Definition 32 (Unbiased estimator)**.** *An estimator $\hat{\theta} : \Sigma \to U$ is an **unbiased estimator** if $\mathbb{E}[\hat{\theta}] = \theta$ for all $\theta \in U$.*

It is also useful for an unbiased estimator to have minimal variance. The condition for an estimator to have minimal variance is defined by the Cramér-Rao inequality:

Theorem 9 (Cramér-Rao inequality)**.** *Let $\hat{\theta}$ be an unbiased estimator. Its covariance matrix S is bounded below by the inverse of the Fisher information matrix G^{-1}, in the sense that $S - G^{-1}$ is positive semidefinite.*

An estimator with such minimal variance is called an *efficient estimator*:

Definition 33 (Efficient estimator)**.** *An unbiased estimator $\hat{\theta} : \Sigma \to U$ is an **efficient estimator** if its covariance matrix S is equal to the inverse of the Fisher information matrix G^{-1}.*

Suppose for now that we know that an observation x is sampled from a normal distribution with unknown μ and σ. It is possible to define an efficient estimator for the parameters.

Consider the two coordinate systems, the *natural coordinates* $[\theta^i]$ and the *expectation coordinates* $[\eta^i]$ given by

$$\begin{aligned}\theta^1 &= \frac{\mu}{\sigma^2}\\ \theta^2 &= -\frac{1}{2\sigma^2}\\ \eta^1 &= \mathbb{E}[x] = \mu\\ \eta^2 &= \mathbb{E}[x^2] = \mu^2 + \sigma^2.\end{aligned}$$

These coordinates define two alternative parameterizations for $\mathcal{N}$; given θ^1 and θ^2, or η^1 and η^2, it is possible solve for μ and σ and thus fully determine the normal distribution. It turns out that $[\theta^i]$ is the affine coordinate system that makes M $\nabla^{(1)}$-flat, and $[\eta^i]$ is the affine coordinate system that makes M $\nabla^{(-1)}$-flat, which is a requirement for us to eventually apply the projection theorem.

Consider the following estimators for η:

$$\begin{aligned}\hat{\eta}^1 &= x\\ \hat{\eta}^2 &= x^2.\end{aligned}$$

It turns out that $\hat{\eta}$ as defined is an efficient estimator: it is unbiased by definition, and it is efficient as its covariance matrix becomes the Fisher information metric after some calculation. Thus, to estimate μ and σ, we simply estimate $\eta \approx \hat{\eta} = (x, x^2)$ and solve for μ and σ using the relations used to define η.

This process does not leverage the machinery of α-connections that we built. However, now suppose that the normal distribution that we are drawing from is known to have variance equal to its mean, that is $\sigma = \mu$. What is the best (most likely) estimate for μ and σ now?

The space of possible normal distributions is now a one-dimensional submanifold S of M. Thus, using the projection theorem, the probability distribution $\hat{\eta}_0$ that minimizes the Kullback-Leibler divergence $D(\hat{\eta} \,||\, \hat{\eta}_0)$ from $\hat{\eta}$ is the $\hat{\eta}_0 \in S$ that is connected to $\hat{\eta}$ with an orthogonal geodesic with respect to $\nabla^{(-1)}$ (because the Kullback-Leibler divergence is the canonical divergence with respect to $\nabla^{(-1)}$). It turns out that such a $\hat{\eta}_0$ also *maximizes* the likelihood of those parameters given x, and thus acts as the best estimator we can hope for that is constrained to S.

We have thus used the machinery of the α-connections and divergences, in order to arrive at a geometric understanding of maximum likelihood estimation.

It is remarkable that the maximum-likelihood problem can be solved with the geometric intuition of the projection theorem, as we did above. It turns out that the above process is easily generalizable to general exponential family distributions, thus covering most of the commonly encountered distributions. Thus, we see that information geometry is remarkably general; let us attempt to use it to gain insight into our problem.

Chapter 9

A new duality between data and statistical manifolds

Information geometry gives a way to add structure to statistical manifolds, a space of probability density functions on a data space Σ. Is there a way for the data space to borrow from the rich structure of a statistical manifold given in the previous chapter? The field of Bayesian statistics provides one possible link.

9.1 Bayesian inference

Bayesian statistics is a powerful branch of statistics that applies the theory of probability to hypotheses. This is best illustrated in contrast to classical *frequentist* statistics, which is perhaps more common than Bayesian statistics. Recall that frequentist statistics defines probability strictly as the frequency of repeated trials: the probability of a coin landing on heads is $\frac{1}{2}$ because in the limit of infinite trials, the number of heads will be $\frac{1}{2}$ the total number of trials. This strict adherence to the interpretation of probabilities, however, limits the power of frequentist statistics in inferring information from data.

Suppose for example that we are trying to determine the calcium carbonate content of a solid sample from a series of chemical experiments. After the data is collected, the frequentist statistician would shy away from asserting that "with a probability 0.8, the sample contains 100 ± 5 grams of calcium carbonate." He or she would instead prefer to express these conclusions in terms of confidence intervals or accepting and rejecting hypotheses. This is because in truth, the sample either does or does not contain 100 ± 5 grams—it does not change on repeated trials. Thus, the frequentist relies on circumlocutions, being careful not to use the word probability for beliefs.

In contrast, the Bayesian statistician would not have any qualms about that assertion, since in Bayesian statistics, probability is interpreted as a *degree of belief based on the given information*, rather than the frequency in the limit of repeated trials. This interpretation of probability is more in line with the layman's usage of probability and is thus arguably more intuitive. For example, even though a particular sports team either will or will not win their game tonight, a fan would have no problem estimating that there is, say, an 80% chance that they will win.

9.1.1 Bayes' theorem

Consider the following example of Bayesian inference, the same example that we used to illustrate maximum likelihood estimation in Subsection 6.3.1.[1] Suppose we know that a sequence of data points $\{x_1, \dots, x_m\}$ comes from a normal distribution with unknown parameters μ and σ, and we would like to estimate these unknown parameters.

The fundamental process of Bayesian inference relies on **Bayes' theorem**, which says that

$$p(\theta \mid X) = \frac{p(X \mid \theta)\, p(\theta)}{p(X)}.$$

This is typically written as

$$p(\theta \mid X) \propto p(X \mid \theta)\, p(\theta),$$

and is interpreted as the following: the probability of a particular parameter θ once we have observed X is proportional to the probability that we observe X given θ, times our existing belief of the probability of θ. The first term, $p(\theta \mid X)$, is known as the **posterior distribution**, and the last term, $p(\theta)$, is known as the **prior distribution**. We can disregard the constant in the bottom of the fraction, since it is easily recovered as

$$p(X) = \int p(X \mid \theta)\, p(\theta)\, \mathrm{d}\theta,$$

in order to make $p(\theta \mid X)$ a proper probability distribution that sums to 1.[2]

[1] Maximum likelihood estimation is a frequentist method.

[2] In practice, none of the distributions discussed here actually need to be normalized (or need to be normalizable at all), since it is still possible to interpret unnormalized distributions as the relative belief of one value versus another. One might see, for example, a uniform distribution on $\mathbb{R}$, which is unnormalizable.

Therefore, in our example, to infer parameters, we initialize our belief for p_θ to some initial prior distribution:

$$p_\theta \leftarrow p(\theta),$$

for all $\theta \in \Theta = \{(\mu, \sigma) \mid \mu \in \mathbb{R}, \sigma \in \mathbb{R}^+\}$. (The choice of prior distribution is discussed in the next section.) Then, for each piece of data x_i, we update our belief by multiplying according to Bayes' rule:

$$p_\theta \leftarrow p(x_i \mid \theta) \cdot p_\theta,$$

where, as usual,

$$p(x \mid \mu, \sigma) = \frac{1}{\sqrt{2\pi}\sigma} \exp\Big(-\frac{(x-\mu)^2}{2\sigma^2}\Big).$$

Finally, once all the data has been accounted for, we recover a proper probability distribution by normalizing it, as

$$p(\theta \mid X) = \frac{p_\theta}{\int p_\theta \, \mathrm{d}\theta}.$$

We then interpret $p(\theta \mid X)$ as our degree of belief that the true parameter is θ; with enough data, the distribution will be peaked at the true value of θ. There are several methods to obtain an explicit estimate for θ, if desired. For example, we may take the expected value of the distribution to obtain an estimate for θ, or we may take the maximum, which corresponds approximately to the maximum likelihood estimate in frequentist statistics. Note that the Bayesian posterior is more powerful than maximum likelihood methods, since it is able to capture situations where the distribution over θ is multimodal, for example.

9.1.2 Jeffreys prior

The prior distribution $p(\theta)$ represents the degree of belief that $\theta \in \Theta$ is the true parameter *before* any observations of data. This may be chosen by incorporating information from previous experiments or through the intuition of an expert in the domain. The prior tends not to matter much when lots of data is involved.

A so-called *uninformative prior* is typically used if we do not have any expectation about what the parameter might be. It may seem that setting the prior distribution to a uniform distribution would incorporate our ignorance about the value of the parameter, but this is naive, since, for example, specifying a uniform distribution on the standard deviation σ would not specify a uniform distribution on the variance σ^2, and vice versa. The uniform distribution is dependent on parameterization, in other words.

The **Jeffreys prior** is an example of a prior distribution that is invariant to such reparameterization and thus serves as a particularly natural uninformative prior. It relies on the invariance of the Fisher information metric $G(\theta)$ and is defined as

$$p(\theta)\,\mathrm{d}\theta \propto \sqrt{\det G(\theta)}\,\mathrm{d}\theta.$$

In Section 8.4, we have seen that on the manifold of normal distributions, the Fisher information metric is

$$G(\theta_1, \theta_2) = \begin{bmatrix} 2\theta_2^{-2} & 0 \\ 0 & 2\theta_2^{-2} \end{bmatrix},$$

where $\theta_1 = \mu$ and $\theta_2 = \sqrt{2}\sigma$. Thus, the Jeffreys prior on the manifold of normal distributions is

$$\begin{aligned} p(\mu, \sigma)\,\mathrm{d}\mu\,\mathrm{d}\sigma &\propto 2\theta_2^{-2}\,\mathrm{d}\theta_1\,\mathrm{d}\theta_2 \\ &\propto \sigma^{-2}\,\mathrm{d}\mu\,\mathrm{d}\sigma. \end{aligned}$$

9.2 Duality in Bayesian inference

Now let us use the idea of Bayesian inference to study the connection between data and statistical manifolds. Let Σ be a data space, a space of possible observations. Then let Θ be a statistical manifold on Σ, so that the elements of Θ are probability distributions on Σ.

Notice that Σ and Θ share a neat duality property. By selecting an element $\theta \in \Theta$, we obtain a probability distribution p_θ on Σ. Conversely, by selecting an element $x \in \Sigma$, we obtain a probability distribution $p(\theta \mid x)$ on Θ, the posterior distribution on Θ after having observed x. We can thus view both spaces from two different perspectives, the "data" perspective, and the "probability" perspective:

$$\begin{aligned} \Sigma &: x \leftrightarrow p(\theta \mid x) \\ \Theta &: \theta \leftrightarrow p(x \mid \theta). \end{aligned}$$

This new perspective leads to a number of interesting ideas.

Originally, we viewed Θ as containing probability distributions on Σ, but now we can think of Σ as containing probability distributions on Θ, since each element $x \in \Sigma$ corresponds to the posterior distribution on Θ after having observed x. In this way, we can view Σ as a statistical manifold as well, parameterized by Θ, under certain regularity conditions. Thus, we may also endow Σ with the Fisher information metric, as

$$g_{ij} = -\mathbb{E}_\theta\left[\frac{\partial^2}{\partial x_i \partial x_j}\log p(\theta \mid x)\right],$$

where the expected value is now taken over Θ instead of Σ like usual. This metric may have interesting properties, properties that are yet to be explored.

It appears limiting that this viewpoint seems not to support sampling multiple observations from Σ to update the posterior distribution on Θ, but this is not the case. If one wants to observe m samples, we simply take Σ to be the m-fold Cartesian product Σ^m; one sample of Σ^m corresponds to m samples of Σ.

Notice that endowing Σ with the Fisher information metric is only possible with the existence of another statistical manifold Θ for Σ to "reflect" on. What choice of Θ can we make—what choice is most natural? One possibility is that we choose the set of *all* probability distributions on Σ. Let us denote this space $H(\Sigma)$, for the *hypothesis space* on X. Note that this set is not necessarily a manifold: if Σ has infinite cardinality, then $H(\Sigma)$ would be an infinite-dimensional manifold. We will continue analysis nonetheless.

This notion allows us to consider the second hypothesis space, $H(H(\Sigma)) = H^2(\Sigma)$, the space of probability distributions on $H(\Sigma)$. Then because the Bayesian inference process above identifies $x \in \Sigma$ with the posterior distribution $p(\theta \mid x)$ on Θ, we can view Bayesian inference as providing a map $\Pi_\Theta : \Sigma \to H^2(\Sigma)$ that maps an observation x into the posterior distribution $p(\theta|x) \in H^2(\Sigma)$. Thus, under some regularity conditions, Σ can be viewed as a true statistical manifold, albeit potentially infinite-dimensional, through the pullback of the map Π_Θ. Moving one step further, the space of functions from $\Sigma \to H^2(\Sigma)$ can be viewed as the space of statistical manifolds on Σ.

9.3 Interpretation as vector space duality

This dual structure is reminiscent of the dual structure of vector spaces, so it is tempting to try to make an explicit connection. We will attempt to conceive of both Σ and Θ as subsets of vector spaces.

Let Σ be a space of observations, and let Θ be a statistical manifold. Elements of Θ are probability distributions on Σ, which are functions from $\Sigma \to \mathbb{R}$ satisfying certain conditions (non-negativity, integrate to 1, etc.). Recall that the set of functions $\Sigma \to \mathbb{R}$ forms a vector space, so Θ can be thought of as a subset of the vector space of functions $\Sigma \to \mathbb{R}$.

Recall also that this function space is canonically isomorphic to the free vector space on Σ, $\mathbb{R}[\Sigma]$. This space $\mathbb{R}[\Sigma]$ contains formal finite linear combinations of elements in Σ

$$a_1x_1 \oplus \cdots \oplus a_mx_m \in \mathbb{R}[\Sigma],$$

where $a_1, \ldots, a_m \in \mathbb{R}$, $x_1, \ldots, x_m \in \Sigma$, and $\oplus$ denotes the addition in $\mathbb{R}[X]$. Note that in this construction, we treat Σ as simply a set with no additive structure, ignoring any existing structure Σ may have. For example, suppose $\Sigma = \mathbb{R}^n$, and let $a, b \in \mathbb{R}$ and $x, y \in \mathbb{R}^n$; then $ax + by \neq ax \oplus by$, where $+$ denotes the standard addition in $\mathbb{R}^n$. Note that Σ can also be thought of as a subset of $\mathbb{R}[\Sigma]$.

To recap, $\Sigma \subseteq \mathbb{R}[\Sigma]$ with the identification $x \mapsto 1 \cdot x$, and $\Theta \subseteq (\Sigma \to \mathbb{R})$. Then $\mathbb{R}[\Sigma] \cong (\Sigma \to \mathbb{R})$ with the isomorphism $x \leftrightarrow \delta_x$, where δ_x is the Dirac delta or Kronecker delta centered at x. Therefore, we can think of Σ and Θ as actually the same type of object, subsets of $\mathbb{R}[\Sigma]$. This is in spite of their superficial differences, one being a space of observations and the other being a statistical manifold.

Let us consider what interpretation we may endow an element of $\mathbb{R}[\Sigma]$. We find that an observation $x \in \Sigma \subseteq \mathbb{R}[\Sigma] \cong (\Sigma \to \mathbb{R})$ is naturally identified as the probability distribution that assigns probability 1 to x and 0 to everything else. A natural interpretation for

$$a = a_1x_1 \oplus \cdots \oplus a_mx_m \in \mathbb{R}[\Sigma],$$

therefore, is as a sequence of observations. That is, if the a_i are integers, then we can think of a as representing the process of observing the data point x_1 a_1 times, etc. If the a_i are real (but non-negative), then we can think of a as representing an infinite sequence of observations where the element x_i occurs with frequency

$$\frac{x_i}{\sum_i x_i}.$$

The probability distribution corresponding to a would yield the observation a in the limit.

We would now like to endow this vector space $\mathbb{R}[\Sigma]$ with an inner product to fully exploit the duality properties of vector spaces. Let $a = \sum_i a_i x_i \in \mathbb{R}[\Sigma]$ and $p = \sum_i p_i \delta_i \in (\Sigma \to \mathbb{R})$. The standard inner product

$$\langle a, p \rangle = \sum_i a_i p_i$$

does not have an obvious interpretation. Instead, consider the similar product on $\mathbb{R}[\Sigma]$ defined by

$$(a \,||\, p) \equiv -\sum_i a_i \log p_i.$$

It turns out that this product has several nice properties.

First, when a is interpreted as an observation and p is interpreted as a probability distribution, this product is negative the log-likelihood of observing a under p:

$$(a \,||\, p) \equiv -\sum_i a_i \log p_i = -\log \prod_i p_i^{a_i}.$$

Second, when p is considered both an observation and a probability distribution, we recover the entropy of p:

$$H(p) = (p \,||\, p) = -\sum_i p_i \log p_i.$$

Indeed, this product $(p \,||\, q)$ is traditionally known as the **cross entropy**, and the Kullback-Leibler divergence between p and q (see Section 8.3) can be seen as

$$D(p \,||\, q) = (p \,||\, q) - (p \,||\, p).$$

It is a subject of further research as to whether this product leads to the duality properties similar to those a standard inner product endows on a vector space.

Chapter 10

Open questions

These last few chapters leave open many avenues for future research, some straightforward, some difficult. I collect them here, organized by topic.

10.1 On defining a Riemannian metric

- What is the role of α in the scaled Euclidean metric (Subsection 7.1.1)? What is the significance of $\alpha = -\frac{2}{n}$? Is there a natural way to choose α based on, say, the dimension of the data?

- The scaled Euclidean metric does not satisfy the invariance property in Section 7.2. Does it satisfy some other invariance property? Can conformal geometry be involved?

- How well do the scaled Euclidean metric (Subsection 7.1.1) and pullback metric with deep density models (Subsection 7.3.1) perform on high-dimensional, realistic data?

- What is the significance of the invariant exponential metric (Subsection 7.2.2)?

- Can the ideas of maximum likelihood curves (Subsection 7.4.1) and prescribed scalar curvature (Subsection 7.4.2) be developed further?

- The intrinsic distance given by our Riemannian metric may improve the accuracy of machine learning techniques. What tests can we perform to evaluate the effectiveness of the metric?

- What are some other applications of an intrinsic distance on the space of data?

10.2 On the duality between data and statistical manifolds

- What are the properties of the Fisher information metric when applied to the space of data, as in Section 9.2?
- What insight can be gained by thinking of the space of functions from $\Sigma \to H^2(\Sigma)$ as the space of statistical manifolds on Σ, as in Section 9.2?
- There is much literature on the space of probability distributions on a set. How do existing results about this space relate to the discussion in Section 9.3?
- How can we conceptualize the second hypothesis space $H^2(\Sigma)$ in terms of vector spaces, in a manner similar to Section 9.3?
- What role does Bayesian statistics play in the interpretation of the vector spaces in Section 9.3?

I invite the reader to think about these open questions, and, more generally, consider what insights can be drawn from thinking about data from a geometric viewpoint. There is clearly much to be done in this exciting area of mathematics!

Part III

Appendices

Appendix A

Local tangent space alignment

An important problem in the field of machine learning is that of *dimensional reduction*, or *manifold learning*. This is illustrated in Figure A.1. Suppose we are given many points in a high-dimensional space $\mathbb{R}^n$ (B) that we suspect lies on a d-dimensional manifold $M \subseteq \mathbb{R}^n$ for $d \ll n$ (A). This is the manifold hypothesis. Can we reconstruct coordinates in a low-dimensional space $\mathbb{R}^d$ for $d \ll n$ that preserves the structure of the original points (C)? Concretely, can we keep points that are nearby in $\mathbb{R}^n$ nearby in $\mathbb{R}^d$ and keep points that are apart in $\mathbb{R}^n$ apart in $\mathbb{R}^d$? The process of reconstructing coordinates in this way is called manifold learning.

We will explore one algorithm for manifold learning, *local tangent space alignment*, as explained by Zhang and Zha (2002, 2003).

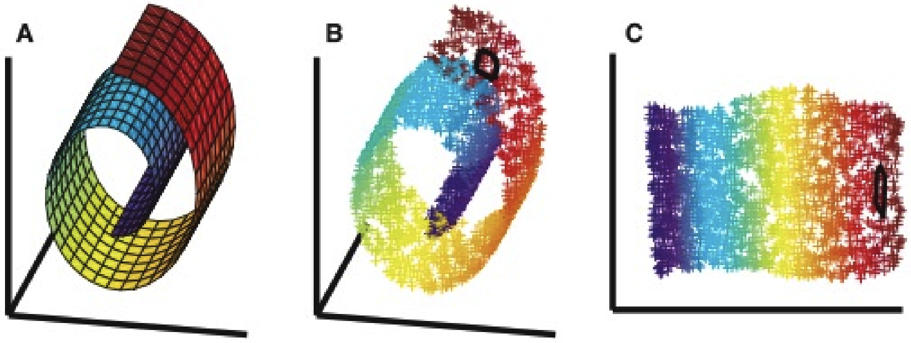


Figure A.1* Dimensional reduction in action.

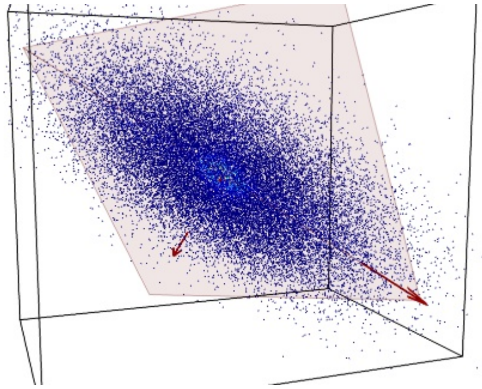


Figure A.2* Points that lie in a 2-dimensional subspace of $\mathbb{R}^3$.

A.1 Principal component analysis

First, we will study linear dimensional reduction using *principal component analysis* (PCA). This technique assumes that the data points $\{y_1, \ldots, y_N\} \subset \mathbb{R}^n$ lie approximately on a d-dimensional subspace of $\mathbb{R}^n$. For $d = 2$ and $n = 3$, this corresponds to the data lying on a plane through the origin. We will also assume that the data has zero mean.

If it is indeed the case that the data points lie approximately on a d-dimensional subspace, we should be able to decompose $\mathbb{R}^n$ into the d-dimensional subspace that approximately contains the data and the $(n-d)$-dimensional subspace in which the data points are approximately 0.

We would like to construct an orthonormal basis of $\mathbb{R}^n$ where the first d vectors span the first subspace and the other vectors span the second. When the data is expressed under such a basis, we will call the first d coordinates the data's *principal components*, and the other components *non-principal components*.

The key observation is that when expressed in such a basis, the coordinates of the data points will have large variance in its principal components and nearly 0 variance in its non-principal components. Therefore, to find a principal component, we will attempt to maximize the variance when the data is projected onto that component.

Recall that the sample variance of N real numbers $q_1, \ldots, q_N$ with 0 mean is defined as

$$\sigma^2 = \frac{1}{N-1} \sum_i q_i^2.$$

Therefore, the sample variance E of the N data points when projected onto a unit vector u is

$$E(u) = \frac{1}{N-1}\sum_i ||u^T y_i||^2. \tag{A.1}$$

We wish to maximize the variance $E(u)$ while varying the direction u, subject to the constraint that $||u|| = 1$.

Define the sample covariance matrix Σ

$$\Sigma = \frac{1}{N-1}\sum_i y_i y_i^T = \frac{YY^T}{N-1},$$

so that we can rewrite Equation A.1 as

$$E(u) = u^T \Sigma u.$$

Optimizing this quantity for a symmetric Σ by varying u while requiring that that $u^T u = 1$ is a well-known optimization problem; this quantity is called the *Rayleigh quotient*. It turns out that if $\lambda_1 \geq \cdots \geq \lambda_n$ are eigenvalues of Σ, then the corresponding eigenvectors $u_1, \ldots, u_n$ are local optima of $E(u)$, where $E(u_1) \geq \cdots \geq E(u_n)$.[1]

Since Σ is a symmetric matrix, $\{u_1, \ldots, u_n\}$ forms an orthogonal basis for $\mathbb{R}^n$. Because we wanted to maximize the variance $E(u)$ for d components, we simply take the d eigenvectors of Σ corresponding to the d largest eigenvalues. Thus, $\{u_1, \ldots, u_d\}$ forms an orthonormal basis for the d-dimensional subspace in which our data lies.

In the following, let $\theta_i \in \mathbb{R}^d$ denote the coordinates of y_i when expressed in this basis, and let $Q : \mathbb{R}^d \to \mathbb{R}^n$ be the corresponding transformation, where $y_i = Q\theta_i$. The columns of Q^T are precisely $\{u_1, \ldots, u_d\}$.

This transformation Q solves the dimensional reduction problem for data lying in linear subspaces, as we have effectively summarized points in $\mathbb{R}^n$ with points in $\mathbb{R}^d$. We shall see next that this technique generalizes to non-linear dimensional reduction.

A.2 Approximating the tangent spaces using PCA

Suppose that we have observed k points $\{y_1, \ldots, y_k\} \subset \mathbb{R}^n$ that we suspect lies approximately on a d-dimensional manifold $M \subset \mathbb{R}^n$. At this point,

[1] This can be shown using calculus and Lagrange multipliers.

assume that the k points are nearby enough that they can be approximated well by an affine subspace (a translated linear subspace).

Let $\bar{y}$ be the mean of $\{y_1, \dots, y_k\}$:

$$\bar{y} = \frac{1}{k} \sum_i y_i.$$

If we run PCA on the set $\{y_1 - \bar{y}, \dots, y_k - \bar{y}\}$ (shifted so that the set has 0 mean), we obtain k PCA coordinates $\{\theta_1, \dots, \theta_k\} \subset \mathbb{R}^d$ and a linear transformation $Q : \mathbb{R}^d \to \mathbb{R}^n$ such that

$$y_i \approx \bar{y} + Q\theta_i. \tag{A.2}$$

At this point, we make the crucial observation that the tangent space $T_{\bar{y}}M$ at $\bar{y}$ can be approximated by the d-dimensional subspace $\operatorname{im} Q$ found using PCA.We see this in what follows.

Let $f : \mathbb{R}^d \to \mathbb{R}^n$ be an unknown parameterization of the manifold M, and let $\bar{x} \in \mathbb{R}^d$ be such that $f(\bar{x}) = \bar{y}$.[2] Then the tangent approximation of f at $\bar{y}$ is

$$f(x) \approx \bar{y} + \mathrm{d}f_{\bar{x}}(x - \bar{x}),$$

where $\mathrm{d}f_{\bar{x}}$ is the Jacobian matrix of f at $\bar{x}$. Evaluating this approximation at x_i, we have

$$y_i \approx \bar{y} + \mathrm{d}f_{\bar{x}}(x_i - \bar{x}). \tag{A.3}$$

Comparing Equation A.2 and Equation A.3, we see that

$$\mathrm{d}f_{\bar{x}}(x_i - \bar{x}) \approx Q\theta_i.$$

This is a statement of our prior observation that the tangent space can be approximated by the subspace spanned by the data's principal components.

Moving further, $\mathrm{d}f$ is regular because M is a manifold, so that we can invert $\mathrm{d}f_{\bar{x}}$ to get

$$x_i - \bar{x} \approx \mathrm{d}f_{\bar{x}}^{-1}\, Q\, \theta_i.$$

If we define $L = \mathrm{d}f_{\bar{x}}^{-1}Q$ and denote the error in this approximation by ϵ_i, we have that

$$x_i = \bar{x} + L\theta_i + \epsilon_i,$$

for some L.

[2] We assume that $\bar{y} \in M$.

Our goal is to minimize the total squared error ξ by selecting x_i and L appropriately:

$$\xi^2(x_1, \dots, x_k, L) = \sum_i ||\epsilon_i||^2 = \sum_i ||x_i - \bar{x} - L\theta_i||^2.$$

At this point, the minimization is trivial as we can simply select $x_i = \bar{x} + L\theta_i$ for any L, yielding $\xi = 0$. However, we will impose further constraints later; for now we will continue to simplify this expression.

To make progress, let us make a simplifying assumption. Recall that

$$\bar{y} = \frac{1}{k} \sum_i y_i.$$

We will assume also that

$$\bar{x} \approx \frac{1}{k} \sum_i x_i,$$

so that the squared error becomes

$$\xi^2(x_1, \dots, x_k, L) = \sum_i ||x_i - \frac{1}{k} \sum_j x_j - L\theta_i||^2$$

We will now switch to matrix notation. Let X be a matrix with columns $\{x_1, \dots, x_k\}$, Θ be a matrix with columns $\{\theta_1, \dots, \theta_k\}$, and e be a vector of 1's. Then we can rewrite ξ as

$$\xi^2(X, L) = ||X - \tfrac{1}{k} X e e^T - L\Theta||_F^2 = ||X(I - \tfrac{1}{k} e e^T) - L\Theta||_F^2,$$

where the norm is now the *Frobenius norm*, defined as

$$||A||_F^2 = \mathrm{tr}(A^T A) = \mathrm{tr}(A A^T).$$

Next, we will eliminate the dependence on L. Notice that if Θ was invertible, then for a fixed X, we could choose

$$L = X(I - \tfrac{1}{k} e e^T)\Theta^{-1}$$

and minimize ξ with $\xi = 0$. Unfortunately, Θ is rarely invertible. However, we can use a generalization of the inverse for this same purpose. For fixed X, it turns out that we can minimize ξ at a fixed X by choosing

$$L = X(I - \tfrac{1}{k} e e^T)\Theta^{+},$$

where Θ^+ is the *Moore-Penrose pseudoinverse* of Θ, which always exists. This further simplifies ξ to become

$$\xi^2(X) = ||X(I - \tfrac{1}{k}ee^T)(I - \Theta^+\Theta)||_F^2.$$

The solution X that minimizes ξ gives the coordinates of the k points of $\mathbb{R}^d$ as X's columns, such that $f(x_i) \approx y_i$. This simplified expression for ξ will be useful in the next step: the alignment of the tangent spaces.

A.3 Alignment of tangent spaces

Recall that in the previous section, we restricted our attention to a single neighborhood in M; we assumed that the k points we observe are close enough to be approximated linearly. Now we will relax this assumption.

Suppose now that we have observed N points $\{y_1, \ldots, y_N\} \subset \mathbb{R}^n$ that we suspect lies approximately on a d-dimensional manifold $M \subset \mathbb{R}^n$.

We will use the same idea as before, where we approximate the tangent space using the d-dimensional subspace found with PCA. The difference is that for each point, we will only use a point's k nearest neighbors to determine its local tangent space.

Let X be a matrix with $\{x_1, \ldots, x_N\}$ as its column. Then let S_i be an N-by-k matrix that extracts the columns corresponding to the k nearest neighbors to y_i. That is, define S_i such that XS_i preserves the columns of X corresponding to y_i's k nearest neighbors. Concretely, S_i is the N-by-N identity matrix with $N - k$ columns deleted.

Then instead of minimizing just

$$\xi^2(X) = ||X(I - \tfrac{1}{k}ee^T)(I - \Theta^+\Theta)||_F^2,$$

we now define

$$\xi_i^2(X) = ||XS_i(I - \tfrac{1}{k}ee^T)(I - \Theta_i^+\Theta_i)||_F^2$$

and minimize the total squared error for all neighborhoods:

$$\min_X \sum_i \xi_i^2(X).$$

We will now simplify

$$\begin{aligned}\sum_i \xi_i^2(X) &= \sum_i ||XS_i(I - \tfrac{1}{k}ee^T)(I - \Theta_i^+\Theta_i)||_F^2 \\ &= \sum_i \operatorname{tr}(XS_i(I - \tfrac{1}{k}ee^T)(I - \Theta_i^+\Theta_i)(I - \Theta_i^+\Theta_i)^T(I - \tfrac{1}{k}ee^T)^T S_i^T X^T) \\ &= \operatorname{tr}(XBX^T),\end{aligned}$$

where we've defined

$$B = \sum_i S_i(I - \tfrac{1}{k}ee^T)(I - \Theta_i^+\Theta_i)(I - \Theta_i^+\Theta_i)^T(I - \tfrac{1}{k}ee^T)^T S_i^T.$$

Therefore, the final minimization problem becomes

$$\min_X \operatorname{tr}(XBX^T).$$

We now impose the constraint that $XX^T = I$; as this matrix is proportional to the covariance matrix, this constraint ensures that each coordinate in $\mathbb{R}^d$ has the same variance.[3] This is an analogue of the Rayleigh quotient problem mentioned above; since we're looking for a minimum, the solution turns out to be such that X contains d eigenvectors of B corresponding to the d smallest eigenvalues. It turns out that the vector of 1s e is an eigenvector of B, so it is more productive to take the d eigenvectors of B corresponding to the 2nd through $(d+1)$th smallest eigenvalues.

The solution X gives the coordinates of the dimensionally-reduced coordinates in $\mathbb{R}^d$ as its columns.

A.4 Example

The following example demonstrates a neat application of dimensional reduction. Consider the images in Figure A.3. They are black-and-white 64-by-64 photos of a face from different angles, and can thus be regarded as vectors in $\mathbb{R}^{64^2}$, with one dimension for each pixel. In reality, however, since the images are of a face from different angles, we expect that the set of images exist in a low-dimensional manifold, perhaps isomorphic to $SO(3)$.

If we run the local tangent space alignment algorithm on these images and attempt to dimensionally reduce the points to $\mathbb{R}^2$, we produce the

[3] Note that this is one of many constraints we could have imposed. We chose this condition for convenience.

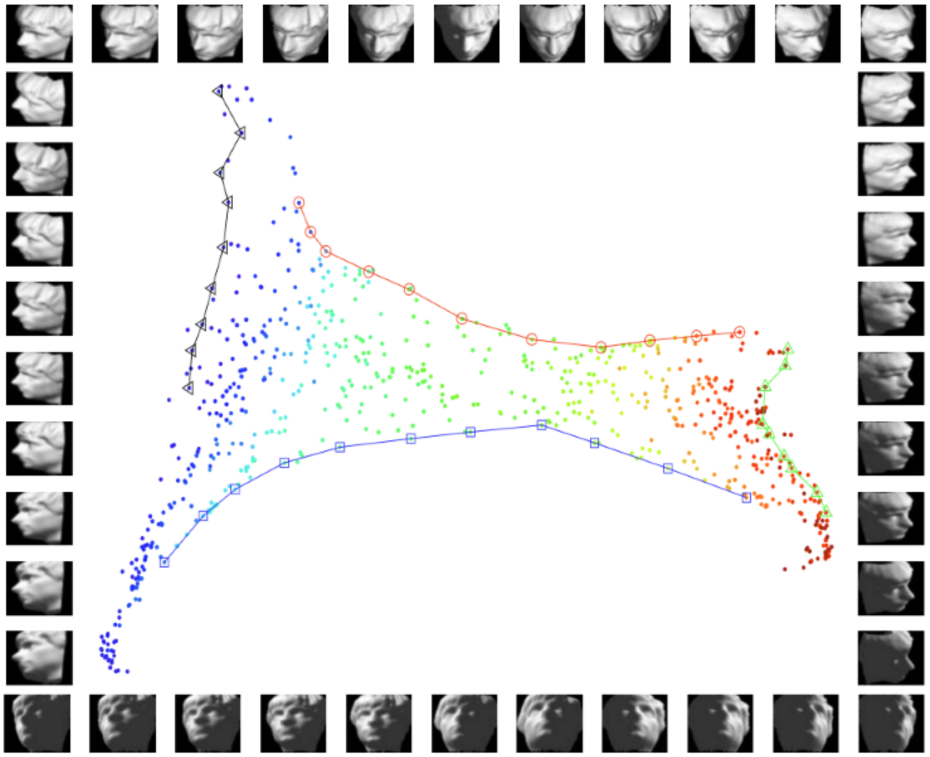


Figure A.3* The LTSA algorithm applied to a set of images.

map shown in Figure A.3. Each colored point is an image, and the images corresponding to the points on the edge of the map are shown along the edge. Notice how the images vary smoothly along the edge of the map, meaning that the algorithm has successfully extracted the low-dimensional structure from the high-dimensional space.

Image and table sources

The symbol * appearing in a figure indicates that the image or table is from an external source, listed below. Their inclusion in this work is intended to constitute "fair use," for non-profit, scholarly use only.

Figure 1.1, Figure 1.2: Investopedia (http://www.investopedia.com/university/charts/).

Figure 2.1, Figure 2.2, Figure 2.3: Müller (2007).

Figure 3.1: Hellisp, Wikimedia Commons (https://commons.wikimedia.org/w/index.php?curid=9442336). Public domain.

Table 3.2, Figure 3.3: Kruskal and Wish (1978).

Table 6.3: R2D3 (http://www.r2d3.us/visual-intro-to-machine-learning-part-1/).

Figure 6.1: Google Maps (https://www.google.com/maps).

Figure 6.2: flightsdubai.org (http://flightsdubai.org/Los-Angeles/Dubai-LosAngeles-flights.php5).

Figure 6.4: Microsoft Azure (https://azure.microsoft.com/en-us/documentation/articles/machine-learning-algorithm-choice/).

Figure 6.5: Tenenbaum et al. (2000).

Figure 6.6: Drleft, Wikipedia (https://en.wikipedia.org/w/index.php?curid=28979849). CC BY-SA 3.0.

Figure 6.7: Drleft, Wikimedia Commons (https://commons.wikimedia.org/w/index.php?curid=11500097). CC BY-SA 3.0.

Figure 6.8: LeCun et al. (2010).

Figure 6.9: Rippel and Adams (2013).

Figure 8.1: Amari and Nagaoka (2007).

Figure 8.2: Januszkaja, Wikimedia Commons (https://commons.wikimedia.org/w/index.php?curid=18176441). CC BY-SA 3.0.

Figure 8.3: Costa et al. (2015).

Figure A.1: Ryan Lei (https://ryanlei.wordpress.com/2011/04/05/ammai_07-nonlinear-dimensionality-reduction-by-locally-linear-embedding/).

Figure A.2: Gaël Varoquaux (http://gael-varoquaux.info/science/ica_vs_pca.html).

Figure A.3: Zhang and Zha (2002).

Bibliography

Agrawal, Rakesh, and Ramakrishnan Srikant. 1995. Mining sequential patterns. In *Proceedings of the Eleventh International Conference on Data Engineering*, 3–14. ICDE '95, Washington, DC, USA: IEEE Computer Society. doi:10.1109/ICDE.1995.380415. 7

Amari, Shun-ichi, and Hiroshi Nagaoka. 2007. *Methods of information geometry*, vol. 191. American Mathematical Soc. 65, 100

Choi, Ben, and Raj Chukkapalli. 2009. Applying machine learning methods for time series forecasting. In *Proceedings of IASTED International Conference on Artificial Intelligence and Applications*, 24–29. 6

Costa, Sueli IR, Sandra A Santos, and João E Strapasson. 2015. Fisher information distance: a geometrical reading. *Discrete Applied Mathematics* 197:59–69. doi:10.1016/j.dam.2014.10.004. 100

Cox, Trevor F, and Michael AA Cox. 2000. *Multidimensional scaling*. CRC Press. doi:10.1007/978-3-540-33037-0_14. 18

do Carmo, M.P. 1992. *Riemannian Geometry*. Mathematics (Boston, Mass.), Birkhäuser. 39

Eberly, David. 2005. Computing geodesics on a Riemannian manifold. URL http://www.geometrictools.com/Documentation/RiemannianGeodesics.pdf.

Faloutsos, Christos, and King-Ip Lin. 1995. FastMap: A fast algorithm for indexing, data-mining and visualization of traditional and multimedia datasets. In *Proceedings of the 1995 ACM SIGMOD International Conference on Management of Data*, 163–174. SIGMOD '95, New York, NY, USA: ACM. doi:10.1145/223784.223812. 20

Fefferman, Charles, Sanjoy Mitter, and Hariharan Narayanan. 2013. Testing the manifold hypothesis. *arXiv preprint* arXiv:1310.0425. 43

Ganti, Venkatesh, Raghu Ramakrishnan, Johannes Gehrke, Allison Powell, and James French. 1999. Clustering large datasets in arbitrary metric spaces. In *Data Engineering, 1999. Proceedings., 15th International Conference on*, 502–511. IEEE. doi:10.1109/ICDE.1999.754966. 24, 25

Gregory, Phil. 2005. *Bayesian Logical Data Analysis for the Physical Sciences: A Comparative Approach with Mathematica® Support*. Cambridge University Press.

Guo, Xinyu, Xun Liang, and Nan Li. 2007. Automatically recognizing stock patterns using RPCL neural networks. In *Proceedings of International Conference on Intelligent Systems and Knowledge Engineering 2007*. Atlantis Press. 7

Gupta, Manish, and Jiawei Han. 2011. Applications of pattern discovery using sequential data mining. *Pattern Discovery Using Sequence Data Mining: Applications and Studies* 1–23.

Han, Jiawei, Hong Cheng, Dong Xin, and Xifeng Yan. 2007. Frequent pattern mining: current status and future directions. *Data Mining and Knowledge Discovery* 15(1):55–86. doi:10.1007/s10618-006-0059-1. 7

Hancock, Edwin, and Marcello Pelillo. 2013. *Similarity-Based Pattern Recognition: Second International Workshop, SIMBAD 2013, York, UK, July 3-5, 2013, Proceedings*, vol. 7953. Springer.

Hartle, James B. 2002. *Gravity: an introduction to Einstein's general relativity*. Pearson Education India. 54

Hobson, Michael Paul, George P Efstathiou, and Anthony N Lasenby. 2006. *General relativity: an introduction for physicists*. Cambridge University Press.

Kazdan, Jerry L, and Frank W Warner. 1975. Scalar curvature and conformal deformation of Riemannian structure. *Journal of Differential Geometry* 10(1):113–134. 62

Korostelev, Aleksandr Petrovich, and Olga Korosteleva. 2011. *Mathematical statistics: asymptotic Minimax theory*, vol. 119. American Mathematical Soc.

Kruskal, Joseph B, and Myron Wish. 1978. *Multidimensional scaling*, vol. 11. Sage. 99

"Learner". 2012. Intuitive explanation of a definition of the Fisher information. Mathematics Stack Exchange. URL http://math.stackexchange.com/q/265933.

LeCun, Yann, Corinna Cortes, and Christopher J.C. Burges. 2010. MNIST handwritten digit database. URL http://yann.lecun.com/exdb/mnist/. 49, 100

Malkoun, Joseph. 2008. A few simple facts about the hyperbolic plane. URL http://malkoun.org/RG/hyperbolic.pdf.

Müller, Meinard. 2007. *Information retrieval for music and motion*, vol. 2. Springer. doi:10.1007/978-3-540-74048-3. 11, 99

Pei, Jian, Jiawei Han, Behzad Mortazavi-Asl, Helen Pinto, Qiming Chen, Umeshwar Dayal, and Mei-Chun Hsu. 2001. PrefixSpan: Mining sequential patterns efficiently by prefix-projected pattern growth. In *ICCCN*, 0215. IEEE. 7

Rippel, Oren, and Ryan Prescott Adams. 2013. High-dimensional probability estimation with deep density models. *arXiv preprint* arXiv:1302.5125. 47, 49, 60, 100

Rosenberg, Jonathan. 2007. Manifolds of positive scalar curvature: a progress report. *Surveys in differential geometry* 11:259–294. doi:10.4310/SDG.2006.v11.n1.a9. 62

Rowe, Glenn. 2015. Metric for the Mercator projection. Physics Pages. URL http://www.physicspages.com/2015/10/07/metric-for-the-mercator-projection/. 42

Srikant, Ramakrishnan, and Rakesh Agrawal. 1996. *Mining sequential patterns: Generalizations and performance improvements*. Springer. 7

Tenenbaum, Joshua B, Vin De Silva, and John C Langford. 2000. A global geometric framework for nonlinear dimensionality reduction. *Science* 290(5500):2319–2323. doi:10.1126/science.290.5500.2319. 99

Zaki, Mohammed J. 2001. Spade: An efficient algorithm for mining frequent sequences. *Machine learning* 42(1-2):31–60. doi:10.1023/A:1007652502315. 7

Zhang, Tian, Raghu Ramakrishnan, and Miron Livny. 1996. BIRCH: An efficient data clustering method for very large databases. In *Proceedings of the 1996 ACM SIGMOD International Conference on Management of Data*, 103–114. SIGMOD '96, New York, NY, USA: ACM. doi:10.1145/233269.233324. 21

Zhang, Z, and H Zha. 2002. Principal manifolds and nonlinear dimension reduction via local tangent space alignement. *Department of Computer Science and Engineering, Pennsylvania State University, Tech Rep CSE-02-019* . 91, 100

Zhang, Zhenyue, and Hongyuan Zha. 2003. Nonlinear dimension reduction via local tangent space alignment. In *Intelligent Data Engineering and Automated Learning*, 477–481. Springer. doi:10.1007/978-3-540-45080-1_66. 91