2017

Machine Learning on Statistical Manifold

Bo Zhang
Harvey Mudd College

Machine Learning on Statistical Manifold

Bo Zhang

Weiqing Gu, Advisor

Nicholas Pippenger, Reader

Department of Mathematics

May, 2017

Abstract

This senior thesis project explores and generalizes some fundamental machine learning algorithms from the Euclidean space to the statistical manifold, an abstract space in which each point is a probability distribution. In this thesis, we adapt the *optimal separating hyperplane*, the *k-means clustering method*, and the *hierarchical clustering method* for classifying and clustering probability distributions. In these modifications, we use the *statistical distances* as a measure of the dissimilarity between objects. We describe a situation where the clustering of probability distributions is needed and useful. We present many interesting and promising empirical clustering results, which demonstrate the statistical-distance-based clustering algorithms often outperform the same algorithms with the Euclidean distance in many complex scenarios. In particular, we apply our statistical-distance-based hierarchical and k-means clustering algorithms to the univariate normal distributions with $k = 2$ and $k = 3$ clusters, the bivariate normal distributions with diagonal covariance matrix and $k = 3$ clusters, and the discrete Poisson distributions with $k = 3$ clusters. Finally, we prove the k-means clustering algorithm applied on the discrete distributions with the Hellinger distance converges not only to the partial optimal solution but also to the local minimum.

Contents

List of Figures

List of Tables

Acknowledgments

Author thanks Professor Weiqing Gu for her insight and advice, Professor Pippenger for being the second reader, and Harvey Mudd College mathematics department for all the support during the process.

Chapter 1

Background

By the end of 1970s, many techniques for extracting information from data had been available. However, many of them are linear models. By the 1980s, computing technology had improved sufficiently so that non-linear methods became feasible computationally. In mid 1980s, classification and regression trees were introduced, and many model selection methods including cross-validation were explored. In 1986, Hastie and Tibshirani coined the term generalized additive models for a class of non-linear extensions to generalized linear models, and provided a practical software implementation. Since that time, inspired by the advent of machine learning and other disciplines, statistical learning has emerged as a new subfield in statistics.

In particular, focus has been given to supervised and unsupervised modeling and prediction, and three broad class of algorithms: regression, classification, and clustering have been widely explored and applied to our everyday life.

Many applications seek to extract information and produce learning algorithms for observations in the Euclidean space, represented by a vector, or an n-tuple. A common practice under such circumstances is to extract certain features from the objects we are interested in and consider these features as vectors in the Euclidean space.

In some cases, extracting features from objects is very difficult, and the distance in the feature space may not best represent the similarity between two objects we are interested in.

In this senior thesis project, the object we are interested in is known as the *probability distribution*. Just like vectors reside in the Euclidean space, the set of probability distributions resides on the *statistical manifold*.

To the knowledge of the author, classification on statistical manifold has not been discussed in literature.

On the other hand, the idea of clustering on statistical manifold has been mentioned in many different contexts. For example, in a conference proceedings by Lee et al. (2007), clustering a set of multinomial distributions is discussed and implemented by applying a simple k-means algorithm with an appropriate distance measure. However, not much effort has been put into systematically studying the clustering problem on the statistical manifold.

The primary aim of this senior thesis project is to generalize some classification algorithms and formalize some clustering algorithms from the Euclidean space to the statistical manifold. The goal of such algorithms is to classify and cluster probability distributions.

Chapter 2

Introduction

2.1 Manifolds

An n-dimensional manifold M is a set of points such that each point has n-dimensional extensions in its neighborhood. Such a neighborhood is topologically equivalent to an n-dimensional Euclidean space. Because a manifold M is locally equivalent to an n-dimensional Euclidean space, we introduce a local coordinate system $\xi = (\xi_1, \xi_2, ..., \xi_n)$. Each point on the manifold is uniquely specified by this induced coordinate system.

A two-dimensional Euclidean space is the most straightforward example of a manifold. It is convenient to use an orthonormal Cartesian coordinate system $\xi = (x, y)$, or a polar coordinate $\xi = (r, \theta)$.

2.2 Probability Distributions

In this section, we briefly review probability distributions, the major objects we are interested in in this project.

2.2.1 Discrete Distributions

The mathematical definition of a discrete probability function $p(x)$ is a function that satisfies the following properties.

- The probability that a random variable X takes satisfies

$$P(X = x) = p(x) \tag{2.1}$$

- $p(x)$ is non-negative for all real-valued x.

- The sum of $p(x)$ over all possible x is equal to 1:

$$\sum_{x} p(x) = 1 \tag{2.2}$$

A discrete probability function is often called a *probability mass function*. It takes a discrete number of values, but not necessarily finite. In fact, it can take countably infinite discrete values. Each of the discrete values has a certain probability of occurrence that is between zero and one. The condition that the probabilities sum to one means that at least one of the values has to occur. A typical example of a discrete distribution is a histogram. Figure 2.1 illustrates a discrete probability mass function.

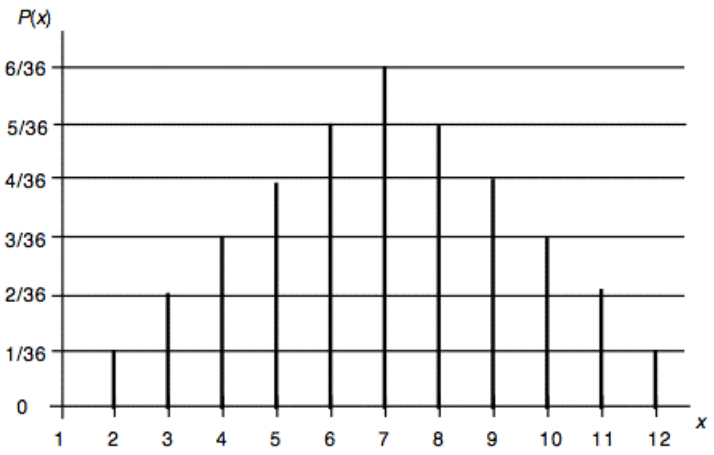


Figure 2.1 An Illustration of a Probability Mass Function

2.2.2 Continuous Distributions

Similarly, the mathematical definition of a continuous probability function $f(x)$ is a right-continuous function that satisfies the following properties.

- The probability that a random variable X takes value between a and b is given by

$$P(a \leq X \leq b) = \int_{a}^{b} f(x)dx \tag{2.3}$$

- $f(x)$ is non-negative for all real-valued x.

- The integral of $f(x)$ is equal to 1:

$$\int_{-\infty}^{\infty} f(x)dx = 1 \tag{2.4}$$

A continuous distribution function is referred to as a probability density function. Note for a probability density function, the probability at a single point is always zero. Probabilities are measured over intervals, not at single points. The area under the curve between two distinct points defines the probability for that interval. The property that the integral must equal one is equivalent to the property for discrete distributions that the sum of all the probabilities must equal one. A univariate normal distribution is a typical example of a continuous probability density function. Figure 2.2 illustrates a standard normal distribution.

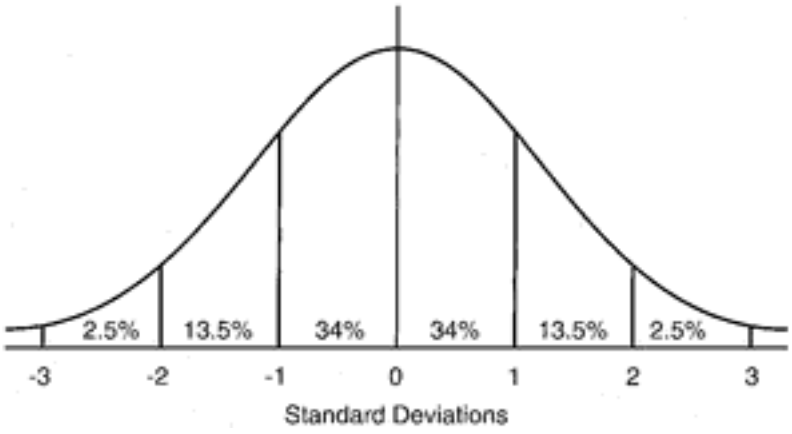


Figure 2.2 An Illustration of a Probability Density Function

2.3 Statistical Manifolds

A statistical manifold is a differentiable manifold where each point represents a probability distribution. The set of all probability measures consists of an infinite-dimensional statistical manifold. Typically, we work with some finite-dimensional sub-manifolds. Note the statistical manifold associates each location in parameter space to a probability density function. Figure 2.3 illustrates this pictorially.

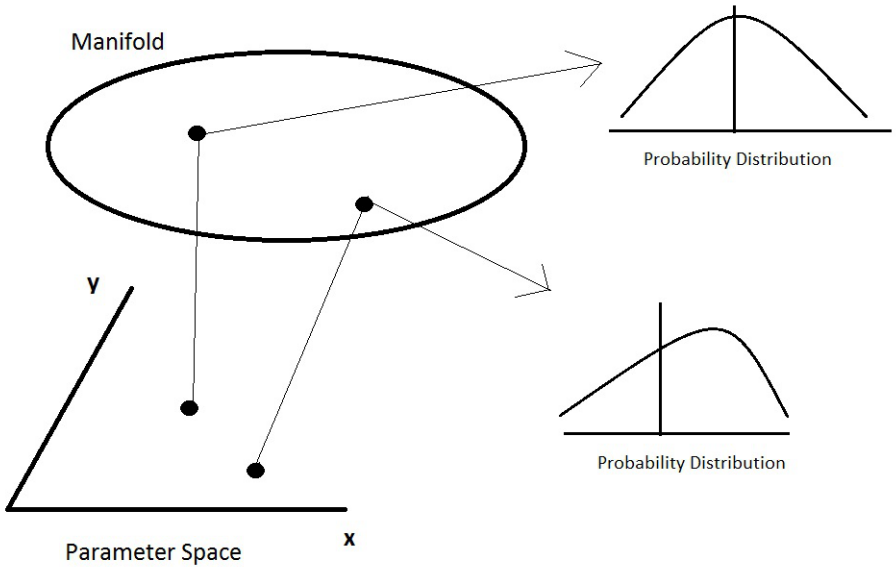


Figure 2.3 An Illustration of a Statistical Manifold

It is important to keep in mind that we always have two spaces: a parameter space where the parameters live and a manifold on which probability distributions reside. This duality allows us to generalize many concepts from the Euclidean space to the statistical manifold. In the rest of this chapter, we briefly review two most important statistical manifolds, namely the manifold of discrete distributions and the manifold of univariate normal distributions.

2.3.1 Manifold of Discrete Distributions

Let x take values from $X = (0, 1, 2, ..., n)$. A distribution for x is specified by $n + 1$ probabilities

$$p_i = \mathbf{P}(x = i), \quad i = 0, 1, 2, ..., n \tag{2.5}$$

Hence, a probability distribution is given by the vector

$$\mathbf{p} = (p_0, p_1, p_2, ..., p_n) \tag{2.6}$$

where $\sum_{i=0}^{n} p_i = 1$, and each $p_i > 0$. The set of all such probability vectors $\mathbf{p}$ form a statistical manifold, and the coordinate system is given by

$$\xi = (p_1, p_2, ..., p_n) \tag{2.7}$$

Note here p_0 is not a free variable because of the constraint $\sum_{i=0}^{n} p_i = 1$. Such a statistical manifold is called a probability simplex, and denoted by S_n Amari (2013). Figure 2.4 illustrates the case of S_2 and S_3.

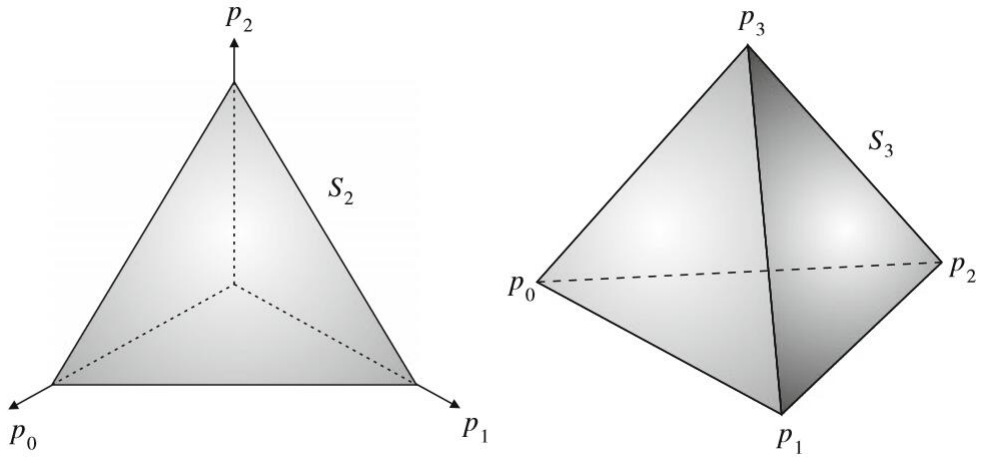


Figure 2.4 Probability Simplex

2.3.2 Manifold of Gaussian Distributions

The probability density function of a univariate Gaussian random variable is given by:

$$f(x; \boldsymbol{\theta}) = \frac{1}{\sqrt{2\pi}\sigma} \exp(\frac{-(x-\mu)^2}{2\sigma^2}) \tag{2.8}$$

Note the natural parametrization of a univariate normal distribution is $\boldsymbol{\theta} = (\mu, \sigma)$, where μ is the mean of the distribution, and σ^2 is the variance. Hence, the set of all the univariate Gaussian distributions is a two-dimensional manifold, where a point denotes a probability density function and

$$\xi = (\mu, \sigma) \quad \sigma > 0 \tag{2.9}$$

is the coordinate system. Note since σ is always positive, the parameter space corresponds to the upper half plane of $\boldsymbol{R^2}$.

Other coordinate systems also work. For example, Let us consider the first and second moments of x, m_1 and m_2, given by

$$m_1 = \mu; \qquad m_2 = \mu^2 + \sigma^2 \tag{2.10}$$

In this case, $\xi = (m_1, m_2)$ forms another coordinate system Amari (2013).

Chapter 3

Approach

3.1 Overview

In this section, we discuss some most widely-used machine learning algorithms in the Euclidean space $\mathbb{R}^n$ and discuss the approaches to generalize these algorithms to statistical manifold.

A rough classification of machine learning algorithms consists of *supervised learning* and *unsupervised learning*. Supervised learning is where we have input variables $\mathbb{x}$ and an output variable y and learn the best mapping function from the input to the output. It is called supervised learning because the process of learning from the training dataset can be thought of as being supervised by the correct answers. The algorithm iteratively makes predictions on the training data and is corrected using the correct answers available. Classification algorithms, in particular, are among the supervised learning algorithms.

Unsupervised learning is where we only have input data $\mathbb{x}$ but no corresponding output variables. The goal for unsupervised learning is to model the underlying structure or distribution in the data in order to learn more about the data. They are called unsupervised learning algorithms because there are no correct answers available to supervise the learning process. Clustering algorithms are among the unsupervised algorithms.

In the following sections, we are going to review some of the most popular classification and clustering algorithms. First, we present two important classification algorithms, known as the *optimal separating hyperplane* and the *support vector machine*.

3.2 Classification

The goal of classification is to learn a mapping from the input $\mathbf{x}$ to output y, where the input $\mathbf{x}$ is a feature vector in Euclidean space $\mathbb{R}^n$ and the output $y \in \{1, 2, ..., k\}$. Note here k denotes the number of classes. If $k = 2$, we say this is a binary classification problem, and we often assume $y \in \{0, 1\}$. If $k \geq 3$, we say this is a multi-class classification problem. Classification problem is everywhere in our daily life and has numerous applications. Notably, classification algorithms are widely used in spam detection and filtering, handwriting recognition, and face recognition.

3.2.1 Optimal Separating Hyperplane in Euclidean Space

We first consider an optimal separating hyperplane between two perfectly separate classes. This is usually used as a motivating example for introducing the more general technique called the *support vector machine*. Suppose we have a training set $(\mathbf{x_1}, y_1), (\mathbf{x_2}, y_2), ..., (\mathbf{x_N}, y_N)$, where $\mathbf{x_i}$ is a vector of features and y_i is either -1 or 1, representing two classes. Let us denote a hyperplane by $\beta_0 + \beta^T \mathbf{x} = 0$. Using elementary linear algebra knowledge, we observe the signed distance from each training data point $(\mathbf{x_i}, y_i)$ to this hyperplane is given by

$$D_i = \frac{1}{\|\beta\|} * (\beta_0 + \beta^T \mathbf{x_i}) \tag{3.1}$$

To get rid of the sign, we multiply the signed distance D_i by y_i. Assuming two classes can be perfectly separated, we solve the following optimization problem:

$$\max_{\beta_0, \beta, \|\beta\|=1} M \tag{3.2}$$

subject to

$$y_i(\beta_0 + \beta^T \mathbf{x_i}) \geq M \quad \textit{for } i = 1, 2, ..., N \tag{3.3}$$

Note here the constraint forces all the points are at least a distance M from the decision boundary defined by β and β_0, and we seek the largest such M and the associated parameters. Figure 3.1 gives an illustration of this algorithm.

However, the above optimization problem is not convex because of the constraint $\|\beta\| = 1$. Some algebraic manipulations get rid of this constraint and yield the following equivalent convex optimization problem:

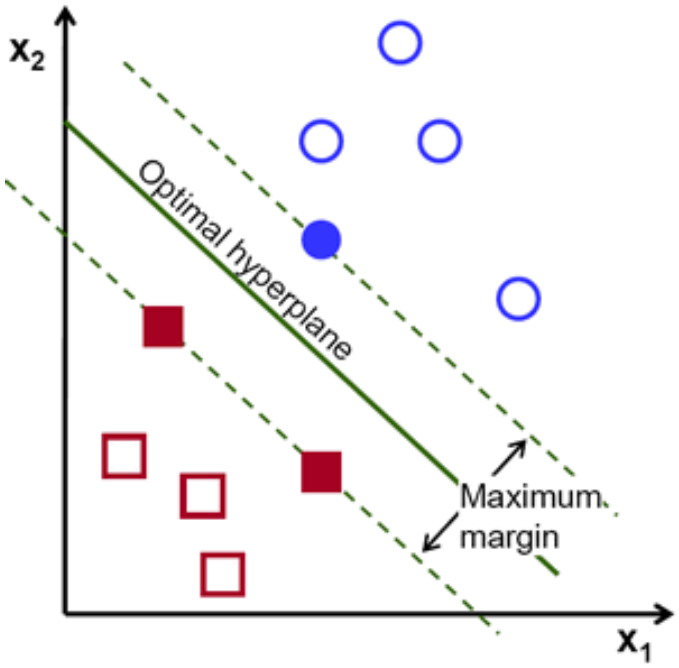


Figure 3.1 An Illustration of the Optimal Hyperplane

$$\min_{\beta_0,\beta} \frac{1}{2}\|\beta\|^2 \tag{3.4}$$

subject to

$$y_i(\beta_0 + \beta^T \mathbf{x_i}) \geq 1 \quad for\ i = 1, 2, ..., N \tag{3.5}$$

Note this optimal separating hyperplane algorithm requires the training data to be perfectly separable, while in practice, this is often not the case. In the next section, we generalize this algorithm to the more general situation where the training data is not necessarily separable.

3.2.2 Support Vector Machine

In many cases, the training data is not perfectly separable and the optimal separating hyperplane algorithm does not apply. Moreover, in the presence of outliers, it is not clear if finding a separating hyperplane is exactly what we want to do. Therefore, we need to make modifications to the optimal

separating hyperplane algorithm to accommodate these more general cases. We reformulate the algorithm as follows:

$$\max_{\beta_0,\beta,\|\beta\|=1,\epsilon_i} M \tag{3.6}$$

subject to

$$\begin{aligned} & y_i(\beta_0 + \beta^T \mathbf{x_i}) \geq M(1-\epsilon_i) \quad \textit{for } i = 1, 2, ..., N \\ & \sum_{i=1}^{n} \epsilon_i \leq C \\ & \epsilon_i \geq 0 \end{aligned} \tag{3.7}$$

where C is the tuning parameter and ϵ_i is the slack variable that allows the i^{th} observation to violate the margin or the boundary.

If $\epsilon_i = 0$, then the corresponding data $\mathbf{x_i}$ is on the correct side of the margin; if $\epsilon_i > 0$, then the corresponding $_i\mathbb{x}$ is on the wrong side of the margin, and we say that the i^{th} observation $\mathbf{x_i}$ has violated the margin. When $\epsilon_i < 1$, the corresponding $\mathbf{x_i}$ lies on the correct side of the hyperplane; when $\epsilon_i > 1$, $\mathbf{x_i}$ might lie on the wrong side of the hyperplane.

The tuning parameter C measures how much the algorithm is tolerant of violations to the margin. If C increases, the algorithm becomes more tolerant of violations to the margin, and the margin M will widen. Conversely, as C decreases, we become less tolerant of violations to the margin and the margin narrows. If $C = 0$, no violation of the margin is allowed and this is the case of optimal separating hyperplane.

Again, we need to make some algebraic manipulations to convert the above support vector machine algorithm into a convex optimization problem. After the modification, the algorithm becomes:

$$\min_{\beta_0,\beta} \frac{1}{2}\|\beta\|^2 + \lambda \sum_{i=1}^{n} \epsilon_i \tag{3.8}$$

subject to

$$\begin{aligned} & y_i(\beta_0 + \beta^T \mathbf{x_i}) \geq 1 - \epsilon_i \quad \textit{for } i = 1, 2, ..., N \\ & \epsilon_i \geq 0 \quad \textit{for } i = 1, 2, ..., N \end{aligned} \tag{3.9}$$

3.3 Clustering

Unlike classification, clustering is an unsupervised learning algorithm. Its aim is to group similar objects together, so that we can extract more information from the data. Below, we are going to review two clustering algorithms, known as the *k-means clustering* and the *hierarchical clustering*.

3.3.1 K-Means Clustering

The main idea of the k-means Clustering is to first define k centroids, one for each cluster. Then the algorithm associates each point in a given dataset to the nearest centroid. When no point is pending, the first step is completed and one groupage is done. At this point we need to re-calculate k new centroids as barycenters of the clusters resulting from the previous step. After we have these k new centroids, we associate each point to the nearest centroid again. This process is repeated until convergence. This iterative process is illustrated in Figure 3.2.

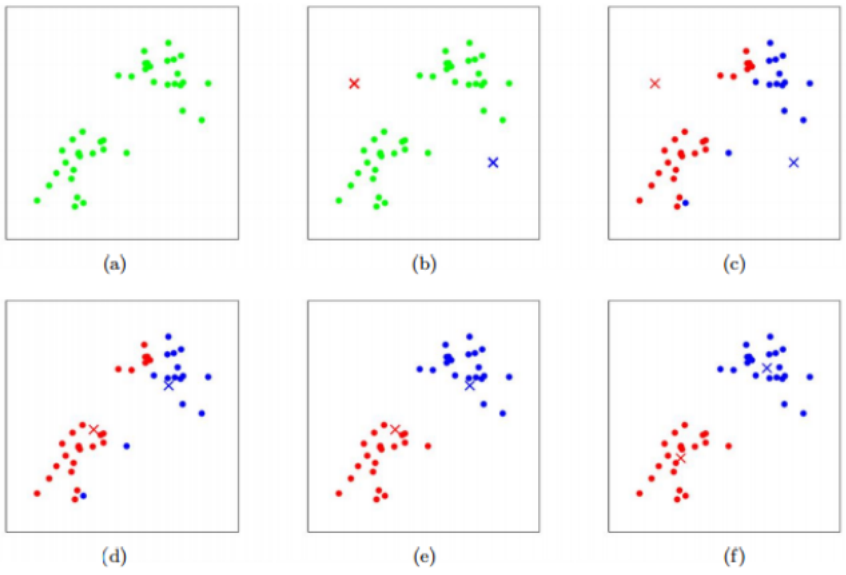


Figure 3.2 An Illustration of K-Means Clustering

3.3.2 Hierarchical Methods

One potential disadvantage of the k-means clustering is that it requires us to pre-specify the number of clusters K. Hierarchical clustering is an alternative approach which does not require a pre-specified k. Moreover, hierarchical clustering produces an attractive tree-based representation of the observations, called dendrogram, which largely facilitates our understanding of the structure of data.

The key to the hierarchical methods is twofold. First, a distance function, or a similarity measure, is required. Second, an internal quality criteria is needed to evaluate the compactness of the clusters. It usually measures the intra-cluster homogeneity, the inter-cluster separability or a combination of these two. Typically, a hierarchical method constructs the clusters from bottom to top by successively merging the clusters, until the desired cluster structure is achieved according to some quality criterion.

The algorithm proceeds by first creating a matrix whose $(i, j)^{th}$ entry is the similarity measure between i^{th} and j^{th} object, and each object initially represents a cluster of its own. Next the algorithm merges the two objects with the highest similarity and replaces the two original objects with this new pair. Accordingly, we update the matrix by recomputing the similarity between the merged pair and the rest. Note when computing the similarity between two clusters, we can either use a shortest distance (largest similarity), or largest distance (smallest similarity), from any member of one cluster to any member of the other cluster. We may stop the algorithm when some pre-specified quality criterion is met. Figure 3.3 illustrates this process. The dashed line indicates the quality criterion is met and the algorithm terminates at that point.

3.4 Discussion and Generalization to Statistical Manifold

Note all the four algorithms discussed in this chapter have a geometric flavor associated with them. The key idea in the optimal separating hyperplane and support vector machine algorithms is the notion of a decision boundary and a distance measure from each observation to the decision boundary. The key idea in the k-means clustering and hierarchical clustering is the notion of a centroid, a similarity measure, and an internal quality criterion.

To generalize these algorithms to the statistical manifold, we will need to

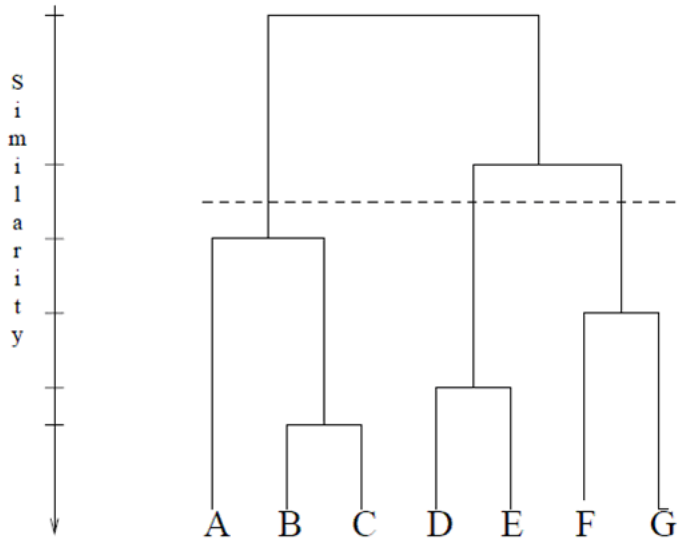


Figure 3.3 An Illustration of Hierarchical Clustering

define the analogies of these geometrical ideas on the statistical manifold. Recall an element in the Euclidean space is a vector, while an element on the statistical manifold is a probability distribution. To define geometrical objects on the statistical manifold, we first need to define a notion of distance to quantify the difference between two probability distributions. In the next chapter, we are going to introduce some widely used statistical distance measures.

Chapter 4

Statistical Distance

In this chapter, we record some most acclaimed and widely used statistical distances that quantify the difference between two probability measures.

4.1 Total Variation Distance

Total variation distance is sometimes referred to "the" statistical distance. For two probability measures P and Q on a sigma-algebra $\mathcal{F}$,

$$TV(P,Q) = sup_{A\in\mathcal{F}}|P(A) - Q(A)| \tag{4.1}$$

This definition is analogous to the *sup norm*, and is simply the largest difference that two probability measures assign to the same event.

4.2 Hellinger Distance

Hellinger distance is another widely used statistical distance. For two probability measures P and Q, with corresponding density $f(x)$,$g(x)$, the *squared Hellinger Distance* is defined to be

$$H^2(P,Q) = \frac{1}{2}\int(\sqrt{f(x)} - \sqrt{g(x)})^2 dx \tag{4.2}$$

Hellinger distance is related to the so-called *Bhattacharyya Coefficient* by the following equation Cieslak et al. (2012):

$$H^2(P,Q) = 1 - BC(P,Q) \tag{4.3}$$

where,

$$BC(P,Q) = \int \sqrt{f(x)g(x)}dx \tag{4.4}$$

By taking the negative logarithm of the Bhattacharya Coefficient, we obtain the *Bhattacharyya Distance*, yet another statistical distance measure.

4.3 Kullback-Leibler Divergence

Kullback-Leibler divergence is yet another well-known measure of the difference between two probability measures. For probability measure P and Q with density f(x) and g(x), the *KL divergence*, or the *relative entropy* is defined as follows:

$$D_{KL}(P||Q) = \int f(x) * log\frac{f(x)}{g(x)}dx \tag{4.5}$$

Note the *KL divergence* is not a distance since it is not symmetric. However, we could define a distance based upon it as follows Lin (2006):

$$JS(P,Q) = 0.5KL(P||T) + 0.5KL(Q||T) \tag{4.6}$$

where $T = 0.5P + 0.5Q$, and Equation 4.6 is known as the *Jensen-Shannon divergence* and it is in fact a proper statistical distance.

4.4 Fisher-Rao Metric

Another proper distance measure arises from the Fisher information matrix, which is a measure of the amount of information of the location parameter. For univariate distributions parametrized by n-dimensional parameter space, the $(i,j)^{th}$ entry of Fisher information matrix is calculated as the expectation of a product involving partial derivatives of the logarithm of the PDF's:

$$g_{ij}(\theta) = \int f(x;\theta)\frac{\partial \ln f(x;\theta)}{\partial \beta_i}\frac{\partial \ln f(x;\theta)}{\partial \beta_j}dx \tag{4.7}$$

The Fisher information matrix $G = (g_{ij})$ defines an inner product as follows:

$$\langle u,v\rangle_G = u^T Gv, and \quad ||u||_G = \sqrt{\langle u,v\rangle_G} \tag{4.8}$$

The distance between two points on the statistical manifold, i.e. the distance between two distributions, is given by the minimum of the lengths

of all the piecewise smooth paths that connect these two points. Recall the length of a path γ is calculated by

$$Length\ of\ \gamma = \int_\gamma ds = \int_\gamma ||(\gamma'(t))||_G dt \tag{4.9}$$

and

$$d_G(P, Q) = \min\{Length\ of\ \gamma\} \tag{4.10}$$

It has been shown in Costa et al. (2015) that in the case of univariate normal distribution, the Fisher-Rao distance between $P = p(x; \mu_1, \sigma_1)$ and $Q = p(x; \mu_2, \sigma_2)$ is given by

$$d_F(\boldsymbol{\theta_1}, \boldsymbol{\theta_2}) = \sqrt{2}\ln(\frac{\mathcal{F}(\boldsymbol{\theta_1}, \boldsymbol{\theta_2}) + (\mu_1 - \mu_2)^2 + 2(\sigma_1^2 + \sigma_2^2)}{4\sigma_1\sigma_2}) \tag{4.11}$$

where $\mathcal{F}(\boldsymbol{\theta_1}, \boldsymbol{\theta_2}) = \sqrt{((\mu_1 - \mu_2)^2 + 2(\sigma_1 - \sigma_2)^2)((\mu_1 - \mu_2)^2 + 2(\sigma_1 + \sigma_2)^2)}$.

Another special case is the multivariate normal distribution with a diagonal covariance matrix $\Sigma = diag(\sigma_1^2, \sigma_2^2, ..., \sigma_p^2)$. Consider two multivariate normal distribution with parameters $\boldsymbol{\theta_1} = (\mu_{11}, \sigma_{11}, ..., \mu_{1p}, \sigma_{1p})$ and $\boldsymbol{\theta_2} = (\mu_{21}, \sigma_{21}, ..., \mu_{2p}, \sigma_{2p})$. The Fisher-Rao Metric between them is given by

$$\begin{aligned} &d_F(\boldsymbol{\theta_1}, \boldsymbol{\theta_2}) \\ =&\sqrt{2\sum_{i=1}^{p}(\ln \frac{\|(\frac{\mu_{1i}}{\sqrt{2}}, \sigma_{1i}) - (\frac{\mu_{2i}}{\sqrt{2}}, -\sigma_{2i})\| + \|(\frac{\mu_{1i}}{\sqrt{2}}, \sigma_{1i}) - (\frac{\mu_{2i}}{\sqrt{2}}, \sigma_{2i})\|}{\|(\frac{\mu_{1i}}{\sqrt{2}}, \sigma_{1i}) - (\frac{\mu_{2i}}{\sqrt{2}}, -\sigma_{2i})\| - \|(\frac{\mu_{1i}}{\sqrt{2}}, \sigma_{1i}) - (\frac{\mu_{2i}}{\sqrt{2}}, \sigma_{2i})\|})^2} \end{aligned} \tag{4.12}$$

Chapter 5

Results: Classification on Statistical Manifold

5.1 Overview

In Chapter 3, we review two classification algorithms, namely the optimal separating hyperplane and the support vector machine. In this chapter, we formulate an analogous optimal separating hyperplane algorithm on a statistical manifold.

First, we draw some analogies between the Euclidean space and the statistical manifold. In the traditional n-dimensional Euclidean space, each point is a vector, an n-tuple. On a statistical manifold, each point represents a probability distribution. In the Euclidean space, we have a measure of the distance between two points, namely the Euclidean distance; on a statistical manifold, we are also able to quantify the distance between two probability distributions, as discussed in Chapter 4.

Now we apply the idea of the optimal separating hyperplane to the statistical manifold. We do so for two different settings. First, our statistical manifold consists of just discrete distributions, and we do not assume any parametric form of them. Rather, we consider these discrete probability mass functions as mere histograms, and regard them as discrete distributions by partitioning them into a same number of intervals. Under this circumstance, the *total variation distance*, or the *Hellinger distance* can be used to define the distance.

Second, our statistical manifold consists of a family of distributions parametrized by a set of parameters θ. For instance, we may consider the

family of all univariate normal distributions parametrized by μ and σ. In this case, we may use some standard Riemannian metrics on this statistical manifold, such as the *Fisher-Rao Metric*.

5.2 Classification of Discrete Distributions

We start by considering the most straightforward case: a statistical manifold of discrete distributions endowed with the Hellinger distance. Recall the discrete version of the Hellinger distance between two distributions $P = (p_0, p_1, ..., p_k)$ and $Q = (q_0, q_1, ..., q_k)$ is given by

$$H^2(P, Q) = \frac{1}{2} \sum_{i=0}^{k} (\sqrt{p_i} - \sqrt{q_i})^2 \tag{5.1}$$

Note in order to define the Hellinger distance between two discrete probability distributions, the probability vectors need to have the same length. To do this, we may partition the support of P and Q into k bins of equal length, with endpoints $(x_0, x_1, ..., x_k)$ and mesh size $x_{i+1} - x_i = \mu$. Now we define our analogy of a hyperplane on the statistical manifold.

Let us associate each interval $[x_i, x_{i+1}]$ with a probability p_i. Each probability measure of the following form gives a point on the manifold of our interest:

$$\mathcal{P} = \{(p_0, p_1, ..., p_k) \quad | \quad \sum_{i=0}^{k} p_i = 1 \quad \text{and} \quad \mathbf{P}(x \in [x_i, x_{i+1}]) = p_i\} \tag{5.2}$$

Next, we want to define a hyperplane on the statistical manifold. To begin with, we want to define a line on a statistical manifold. Recall a line connects two points in Euclidean space is given by

$$\mathbf{x}(t) = (1 - t)\mathbf{x}_1 + t\mathbf{x}_2 \tag{5.3}$$

We cannot apply this technique directly on a statistical manifold, but we can certainly do this on the parameter space. In the case of discrete distributions, each distribution is parametrized by $\boldsymbol{\theta} = (p_1, p_2, ..., p_k)$. Note here $p_0 = 1 - \sum_{i=1}^{k} p_i$, and hence irrelevant. Define

$$\boldsymbol{\theta}(t) = (1 - t)\boldsymbol{\theta_1} + t\boldsymbol{\theta_2} \tag{5.4}$$

This is the geodesic connecting two distributions $p(x; \boldsymbol{\theta_1})$ and $p(x; \boldsymbol{\theta_2})$ Amari (2013). Note the corresponding family of distributions on the statistical

manifold is $p(x; \boldsymbol{\theta}(t))$. This can be our definition of an analog of a line on the statistical manifold.

It is worth noting in the case of discrete distribution, we have much simplicity:

$$p(x; \boldsymbol{\theta}(t)) = (1 - t)p(x, \boldsymbol{\theta_1}) + tp(x, \boldsymbol{\theta_2}) \tag{5.5}$$

However, this is in general not true, as we will discuss more complicated cases later.

Let us fix two points $p(x, \boldsymbol{\theta_1})$ and $p(x, \boldsymbol{\theta_2})$ on the manifold and consider the line $p(x; \boldsymbol{\theta}(t))$ connecting them. The Hellinger distance from the i^{th} discrete distribution $Q_i = (q_{i0}, ..., q_{ik})$ to this hypothetical line is given by

$$D_i = \min \sqrt{\frac{1}{2} \sum_{j=0}^{k} (\sqrt{p_j} - \sqrt{q_{ij}})^2} \tag{5.6}$$

where the minimization is taken over all $(p_0, p_1, ..., p_k) \in p(x; \boldsymbol{\theta}(t))$, or equivalently over t.

Eventually, we define a collection of observed distributions, parametrized by $\boldsymbol{\theta}$ and labeled by either $+1$ or -1, as *separable* if there exists a hyperplane separating one class from the other in its parameter space.

From all of our above discussion, we conclude the following:

- For a parametrized statistical manifold, we should think of both the abstract space of $p(x; \boldsymbol{\theta})$ and the associated parameter space
- An optimization algorithm involves both spaces. The parameter space tells us the separability of distributions and helps define the decision boundary, while the abstract space of distributions introduces the measure of distance between two distributions.

Now we are ready to formalize our analog of the *separable hyperplane algorithm* on the manifold of discrete distributions with the Hellinger distance. Consider a sequence of distributions $\{Q_i\}$, each of which is labeled $y_i = +1/-1$. Suppose $\{Q_i = (q_{i0}, q_{i1}, ..., q_{ik})\}$ are perfectly separable. We find the best separating hyperplane using the following algorithm:

$$
\begin{aligned}
&\max_{\boldsymbol{\theta_1}, \boldsymbol{\theta_2}, \beta_0, \boldsymbol{\beta^T}} D_i \\
&D_i = \min_t \sqrt{\frac{1}{2} \sum_j (\sqrt{p_{tj}} - \sqrt{q_{ij}})^2} \quad \textit{for each } i \\
&(p_{t0}, p_{t1}, ..., p_{tk}) = (1-t)p(x, \boldsymbol{\theta_1}) + tp(x, \boldsymbol{\theta_2}) \\
&y_i(\beta_0 + \sum_j \beta_j q_{ij}) \geq 0 \quad \textit{for each } i \\
&\beta_0 + \boldsymbol{\beta^T} \boldsymbol{\theta_1} = 0 \\
&\beta_0 + \boldsymbol{\beta^T} \boldsymbol{\theta_2} = 0
\end{aligned}
\tag{5.7}
$$

5.3 Classification of Univariate Normal Distributions

Consider the univariate Gaussian distribution with probability density function:

$$f(x; \boldsymbol{\theta}) = \frac{1}{\sqrt{2\pi}\sigma} \exp(\frac{-(x-\mu)^2}{2\sigma^2}) \tag{5.8}$$

Note the natural parametrization of univariate normal distribution is $\boldsymbol{\theta} = (\mu, \sigma)$. The parameter space is 2-dimensional. Note since σ is always positive, the parameter space corresponds to the upper half plane of $\boldsymbol{R^2}$.

A most common Riemannian metric for normal distribution is *Fisher-Rao Metric*. Given two univariate normal distributions with parameter $\boldsymbol{\theta_1} = (\mu_1, \sigma_1)$ and $\boldsymbol{\theta_2} = (\mu_2, \sigma_2)$, the *Fisher-Rao Metric*, or *Fisher information distance* is given by:

$$d_F(\boldsymbol{\theta_1}, \boldsymbol{\theta_2}) = \sqrt{2} \ln(\frac{\mathcal{F}(\boldsymbol{\theta_1}, \boldsymbol{\theta_2}) + (\mu_1 - \mu_2)^2 + 2(\sigma_1^2 + \sigma_2^2)}{4\sigma_1\sigma_2}) \tag{5.9}$$

where $\mathcal{F}(\boldsymbol{\theta_1}, \boldsymbol{\theta_2}) = \sqrt{((\mu_1 - \mu_2)^2 + 2(\sigma_1 - \sigma_2)^2)((\mu_1 - \mu_2)^2 + 2(\sigma_1 + \sigma_2)^2)}$.

The manifold of normal distributions is more interesting in that the linear interpolation of two points in the parameter space does not translates directly into a linear interpolation of distributions. Now we derive the correct form of $p(x; \boldsymbol{\theta}(t))$, where $\boldsymbol{\theta}(t) = (1-t)\boldsymbol{\theta_1} + t\boldsymbol{\theta_2}$.

First, we show the univariate normal distribution is in *exponential family*.

In fact, one can see:

$$\begin{aligned}
&\frac{1}{\sqrt{2\pi}\sigma}\exp(\frac{-(x-\mu)^2}{2\sigma^2}) \\
=&\frac{1}{\sqrt{2\pi}}\exp(-\log\sigma-\frac{x^2}{2\sigma^2}+\frac{\mu x}{\sigma^2}-\frac{\mu^2}{2\sigma^2}) \\
=&\frac{1}{\sqrt{2\pi}}\exp(\boldsymbol{\alpha}^{\boldsymbol{T}}T(x)-\log\sigma-\mu^2/2\sigma^2) \\
=&\frac{1}{\sqrt{2\pi}}\exp(\boldsymbol{\alpha}^{\boldsymbol{T}}T(x)-\phi(\theta))
\end{aligned} \tag{5.10}$$

where $T(x)=\begin{pmatrix}x\\x^2\end{pmatrix}$, $\boldsymbol{\alpha}=\begin{pmatrix}\mu/\sigma^2\\-1/(2\sigma^2)\end{pmatrix}$, and $\phi(\theta)=\log\sigma+\mu^2/2\sigma^2$. As a result, we have:

$$\begin{aligned}
&p(x;\boldsymbol{\theta}(t)) \\
&=p(x;(1-t)\boldsymbol{\theta_1}+t\boldsymbol{\theta_2}) \\
&=\frac{1}{\sqrt{2\pi}}\exp(t(\boldsymbol{\alpha_2}-\boldsymbol{\alpha_1})^T T(x)-\phi(t))
\end{aligned} \tag{5.11}$$

where $\boldsymbol{\alpha_1}=\begin{pmatrix}\mu_1/\sigma_1^2\\-1/(2\sigma_1^2)\end{pmatrix}$, $\boldsymbol{\alpha_2}=\begin{pmatrix}\mu_2/\sigma_2^2\\-1/(2\sigma_2^2)\end{pmatrix}$, $\phi(t)=\log\sigma_t+\mu_t^2/2\sigma_t^2$, and $\begin{pmatrix}\mu_t\\\sigma_t\end{pmatrix}=(1-t)\begin{pmatrix}\mu_1\\\sigma_1\end{pmatrix}+t\begin{pmatrix}\mu_2\\\sigma_2\end{pmatrix}$.

Equation 5.11 is a line on the statistical manifold of normal distributions. To conclude, we have found a metric on the statistical manifold, as well as an analog of a line on the manifold. Note in the case of univariate normal distributions, the separability condition is even more intuitive. Since the parameter space is two-dimensional, we only need a line separating the two classes. Let the training data be a sequence of normal distributions parametrized by $\mathbf{x_1}, \mathbf{x_2}, ..., \mathbf{x_n}$, where $\mathbf{x_i}=\begin{pmatrix}\mu_i\\\sigma_i\end{pmatrix}$, and each comes with a label $y_i \in \{+1, -1\}$. Let $\mathbf{DF}(p_1, p_2)$ be a metric defining the distance between two probability densities p_1 and p_2. Now we are ready to state the corresponding optimization algorithm:

$$
\begin{aligned}
&\max_{\boldsymbol{\theta_1},\boldsymbol{\theta_2},\beta_0,\boldsymbol{\beta^T}} D \\
&D_i \geq D \\
&D_i = \min_t \mathbf{DF}(p(x;\mathbf{x_i}), p(x;\boldsymbol{\theta}(t)) \quad \textit{for each } i \\
&p(x;\boldsymbol{\theta}(t)) = p(x;(1-t)\boldsymbol{\theta_1} + t\boldsymbol{\theta_2}) \\
&y_i(\beta_0 + \boldsymbol{\beta^T}\mathbf{x_i}) \geq 0 \quad \textit{for each } i \\
&\beta_0 + \boldsymbol{\beta^T}\boldsymbol{\theta_1} = 0 \\
&\beta_0 + \boldsymbol{\beta^T}\boldsymbol{\theta_2} = 0
\end{aligned}
\tag{5.12}
$$

Note here $p(x;\boldsymbol{\theta}(t))$ was given analytically in Equation 5.11, and when we take the metric **DF** to be the Fisher-Rao metric defined in Equation 5.9, the problem can be further simplified:

$$
\begin{aligned}
&\max_{\boldsymbol{\theta_1},\boldsymbol{\theta_2},\beta_0,\boldsymbol{\beta^T}} D \\
&\min_t d_F(\mathbf{x_i}, \boldsymbol{\theta}(t)) \geq D \quad \textit{for each } i \\
&\boldsymbol{\theta}(t) = (1-t)\boldsymbol{\theta_1} + t\boldsymbol{\theta_2} \\
&y_i(\beta_0 + \boldsymbol{\beta^T}\mathbf{x_i}) \geq 0 \quad \textit{for each } i \\
&\beta_0 + \boldsymbol{\beta^T}\boldsymbol{\theta_1} = 0 \\
&\beta_0 + \boldsymbol{\beta^T}\boldsymbol{\theta_2} = 0
\end{aligned}
\tag{5.13}
$$

Note in the case of univariate normal distributions, the Fisher-Rao metric is given analytically in the terms of the parameters. In general, this is not true and the more general system, i.e., Equation 5.12 is desired.

Chapter 6

Results: Clustering on Statistical Manifold

6.1 Overview

In this chapter, we discuss the application of the *hierarchical clustering methods* and the *k-means clustering* to the clustering problems on a statistical manifold. Recall in Chapter 3, we introduce these two algorithms and observe that the key idea in the k-means clustering and hierarchical clustering is the notion of a centroid, a similarity measure, and an internal quality criterion. Recall in Chapter 4, we introduce several statistical distances to measure the similarity between two distributions, and they can serve as our similarity measures on statistical manifold.

Now, we focus on defining an analogy for a centroid on the statistical manifold, which is essential in k-means clustering and and helps to define the quality criteria in hierarchical clustering model.

6.2 Centroid on Statistical Manifold

Suppose we are in $\mathbf{R}^n$. A centroid of a sequence of vectors $\mathbf{x_1}, \mathbf{x_2}, ..., \mathbf{x_n}$ in the standard Euclidean space is defined by

$$\mathbf{x}^* = \frac{1}{n} \sum_{1}^{n} \mathbf{x_i} \tag{6.1}$$

Note the situation is different on a statistical manifold. Consider a sequence of continuous probability distributions $p_1, p_2, ..., p_n$ from a general

statistical manifold. Note $\mathbf{p}^* = \frac{1}{n}\sum_1^n \mathbf{p_i}$ is still a probability distribution because

$$\int \mathbf{p}^* = \int \frac{1}{n}\sum_1^n \mathbf{p_i} = \frac{1}{n}\sum_1^n \int \mathbf{p_i} = 1 \tag{6.2}$$

Therefore, we see a straightforward analogy of the centroid on the statistical manifold yields another point on the statistical manifold.

However, the same is not true when we are restricted to a sub-manifold. Consider the manifold of univariate normal distributions. We can similarly apply Equation 6.2 and obtain $\mathbf{p}^* = \frac{1}{n}\sum_1^n \mathbf{p_i}$, where each p_i is a univariate normal distribution. However, this only yields a mixture of normal distributions, which is not itself a normal distribution and does not fall on this sub-manifold. Therefore, this formulation fails when we restrict ourselves to this sub-manifold, because the set of univeriate normal distributions is not a vector space.

Instead of directly taking average of normal distributions themselves, we can take the average of parameters in the two-dimensional Euclidean parameter space, and then map this centroid in parameter space back to our manifold. Note under this construction, the corresponding distribution is necessarily on the manifold. Mathematically, we have:

$$\begin{aligned} \theta^* &= \frac{1}{n}\sum_1^n \theta_\mathbf{i}; \\ p(x;\theta^*) &\in \mathcal{M}; \end{aligned} \tag{6.3}$$

where $\theta_\mathbf{i} = (\mu_i, \sigma_i)$, $\mathcal{M}$ denotes the manifold of univariate normal distributions, and $p(x;\theta^*)$ is our centroid on this manifold. We may define a centroid similarly for other parametrized families of distributions.

6.3 K-Means Clustering on Statistical Manifold

Now we can formulate the k-means algorithm to cluster distributions on statistical manifolds. Let $p(x;\theta)$ be a continuous probability distribution parametrized by θ. Suppose we have observed a sequence of distributions $p(x;\theta_1)$, $p(x;\theta_2)$, $p(x;\theta_3)$, ..., $p(x;\theta_n)$. We propose the following algorithm to cluster the distributions.

1. Fix k probability distributions on the statistical manifold, where k represents k clusters. These k distributions represent centroids for each of the k cluster.

2. Assign each $p(x;\theta_i)$ to the nearest centroid using a proper statistical distance, like the Fisher-Rao metric.

3. Recalculate each of the k centroids as described in subsection 6.2.

4. Repeat step 2 and 3 until the centroids no longer move.

6.4 Hierarchical Methods on Statistical Manifold

Recall in the hierarchical methods, we start by construct a matrix whose $(i,j)^{th}$ entry measures the similarity between i^{th} and j^{th} object. By using a proper statistical distance, we can apply this method to cluster distributions on a statistical manifold. It suffices to identify an internal quality criterion.

A simplest and most widely used internal quality criterion for clustering in the Euclidean space is *sum of squared error (SSE)*:

$$SSE = \sum_{1}^{k} \sum_{x_i \in C_k} (x_i - \mu_k)^2 \tag{6.4}$$

where C_k denotes each cluster our algorithm specifies and μ_k is the centroid of each cluster.

On a statistical manifold, we have $\mu_k = p(x;\theta^*)$, where θ^* is the average over all parameters of cluster k. Instead of the Euclidean distance, we adopt a proper statistical distance d_F and SSE becomes:

$$\begin{aligned} SSE &= \sum_{1}^{k} \sum_{p(x;\theta_i) \in C_k} d_F(p(x;\theta_i) - \mu_k) \\ where \quad \mu_k &= p(x;\theta^*) \\ \theta^* &= \frac{1}{n} \sum_{p(x;\theta_i) \in C_k} \theta_i \end{aligned} \tag{6.5}$$

We will make use of the concept of a centroid and the SSE later.

Chapter 7

Implementation and Generating Clusters

In this chapter, we discuss the effective implementation of our clustering algorithms and how to generate the clusters for clustering.

7.1 Implementation

The implementation of clustering algorithms is achieved in R, a widely-used, open-source statistical software. The clustering code consists of the following functions.

- Compute several statistical distances including the Hellinger distance for discrete distributions, and the Fisher-Rao metric for univariate and bivariate normal distributions.
- Perform hierarchical clustering based on the Euclidean distance.
- Perform hierarchical clustering based on the statistical distance.
- Compute the internal quality criterion for hierarchical clustering methods.
- Perform k-means clustering based on the Euclidean distance.
- Perform k-means clustering based on the statistical distance.
- Visualize the clustering.

- Compute and compare different clustering statistics.
- Generate simulated clusters.

Note all the clustering functions apply to the cases of univariate normal distributions, bivariate normal distributions, and Poisson distributions. The code is modularized and users may adapt the code to meet their specific demands.

7.2 Generating Clusters

First, we propose a situation where clustering methods applied to statistical manifold are useful. We consider the manifold of univariate normal distributions below and other statistical manifolds follow similarly.

Consider k distinct groups of objects. Each of these groups has an underlying, unknown univariate normal distribution which is parametrized by (μ_k, σ_k). In practice, each distribution may represent, for instance, the overall height distribution of citizens in k different countries, or the average SAT score of k different colleges. However, often in practice, each of the underlying distribution (μ_k, σ_k) cannot be directly obtained; rather it is estimated using samples.

Imagine a situation where we are able to obtain samples many times. For each underlying (μ_k, σ_k), We use t batches of samples to reconstruct and estimate the underlying distribution (μ_k, σ_k) t times. However, since these are merely estimates, we will not obtain precisely the underlying distributions. Rather, our estimated distributions will scatter around the true one on the statistical manifold. In the associated parameter space, one can imagine observing k cohorts of points, with each cohort centering around (μ_k, σ_k).

We summarize the above discussion below. To generate clusters on the statistical manifold of univariate normal distributions, we adopt the following steps:

1. Select k pairs of parameters (μ_k, σ_k).
2. For each k, do the following t times:
   (a) Generate n samples from the univariate normal distribution $f(x; \mu_k, \sigma_k)$.

(b) Reconstruct $f(x; \mu_k, \sigma_k)$ from these n samples and obtain unbiased estimates $(\tilde{\mu_k}, \tilde{\sigma_k})$.

Figure 7.1 Generating Clusters for k = 2, 3

7.1 and 7.2 are graphs illustrating this cluster-generating process for $k = 2, 3, 4, 5$, with parameters $t = 60$ and $n = 50$.

Note each graph represents the parameter space for the statistical manifold of univariate normal distributions. Each point represents one univariate normal distribution. The mean and standard deviation associated with each point are the empirical mean and empirical standard deviation, which are estimates of the true underlying mean and standard deviation. Different colors represent different true underlying distributions from which samples are drawn and used to reconstruct the distribution.

Note parameter t controls how many elements each cluster has and n controls how tight each cluster is. When we reconstruct the distribution and estimate parameters using a very small n, the resulting estimates can deviate from the true underlying parameters by a lot. For example, when we take $n = 10$, the resulting graph is much loosely clustered. See Figure 7.3 for an example.

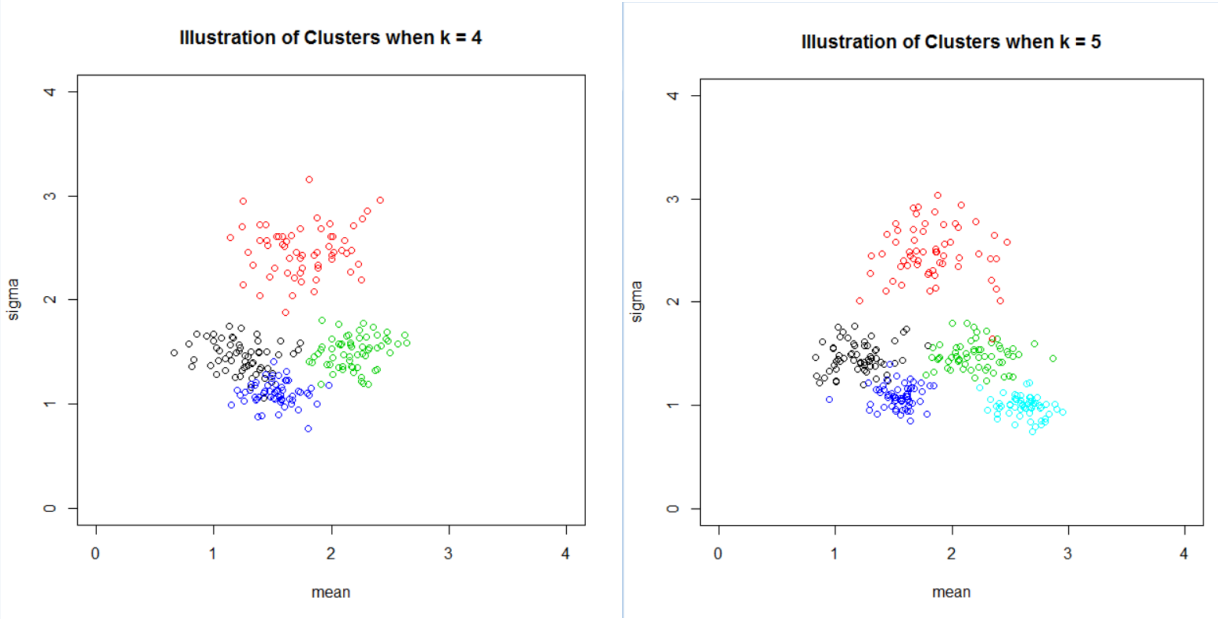


Figure 7.2 Generating Clusters for k = 4, 5

To implement the clustering algorithm, we are given an object consisting of an unknown number of clusters of points, as shown in Figure 7.4. Our goal is to cluster this object using algorithms discussed in the preceding chapters and compare our results with the true result. Note clustering is in general an unsupervised learning problem. But because of our way of generating these data, we can compare our clustering results with the ground truth.

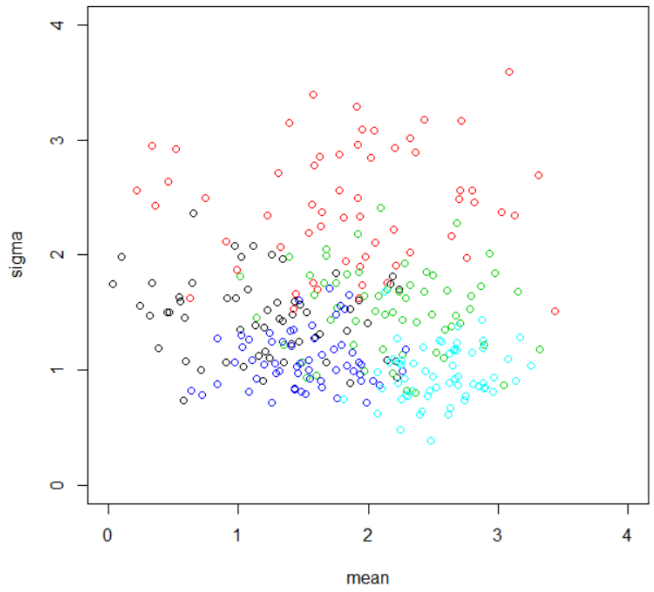


Figure 7.3 Generating Clusters for k = 5 and n = 10

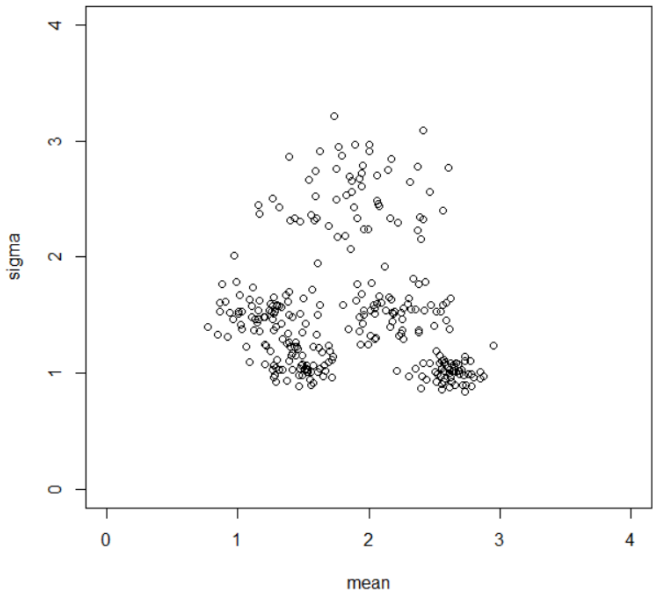


Figure 7.4 An Example of an object to be clustered

Chapter 8

Empirical Results: Clustering Univariate and Bivariate Normal Distributions

Now we discuss the empirical results obtained using clustering algorithms discussed in the previous chapters. This chapter focuses on clustering univariate normal distributions with $k = 2$, $k = 3$, and bivariate normal distributions with $k = 3$. Empirical results using the statistical distance are compared with the results using the Euclidean distance.

8.1 Univariate Normal Distribution with $k = 2$

First, we consider the case of univariate normal distributions with 2 clusters. As an illustrative example, consider $(\mu_1, \sigma_1) = (1, 1.5)$ against $(\mu_2, \sigma_2) = (2, 1.5)$. We let $n = 30$ and $t = 100$. Figure 8.1 provides an illustration of the parameter space under this set of parameters.

First we look at the results using the hierarchical clustering based on the statistical distance. In this case, we use the Fisher-Rao metric as the specific statistical distance and we use the internal quality criterion as defined in Section 6.4 to determine the number of clusters. Ideally, SSR reveals this information when we plot SSR against different number of clusters k. Figure 8.2a provides an illustration of such a plot. Note SSR strictly decreases as the number of clusters increases. However, it is the most abrupt drop that we are interested in. In this case, one can observe that when we move from 1 cluster to 2 clusters, SSR drops dramatically, indicating strong evidence for

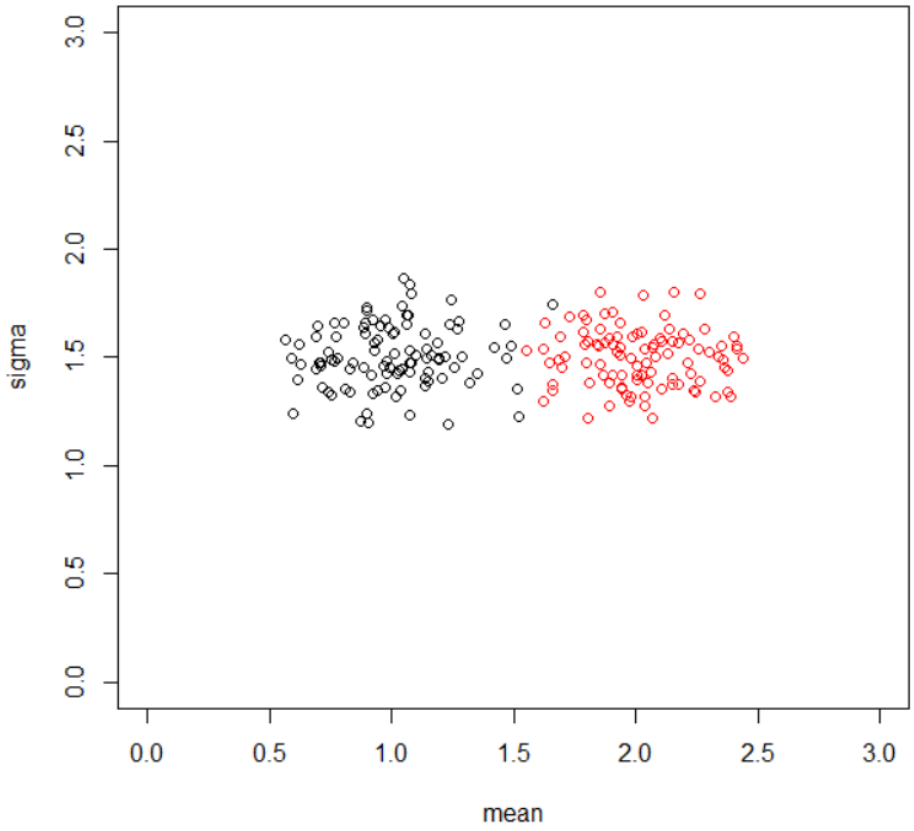


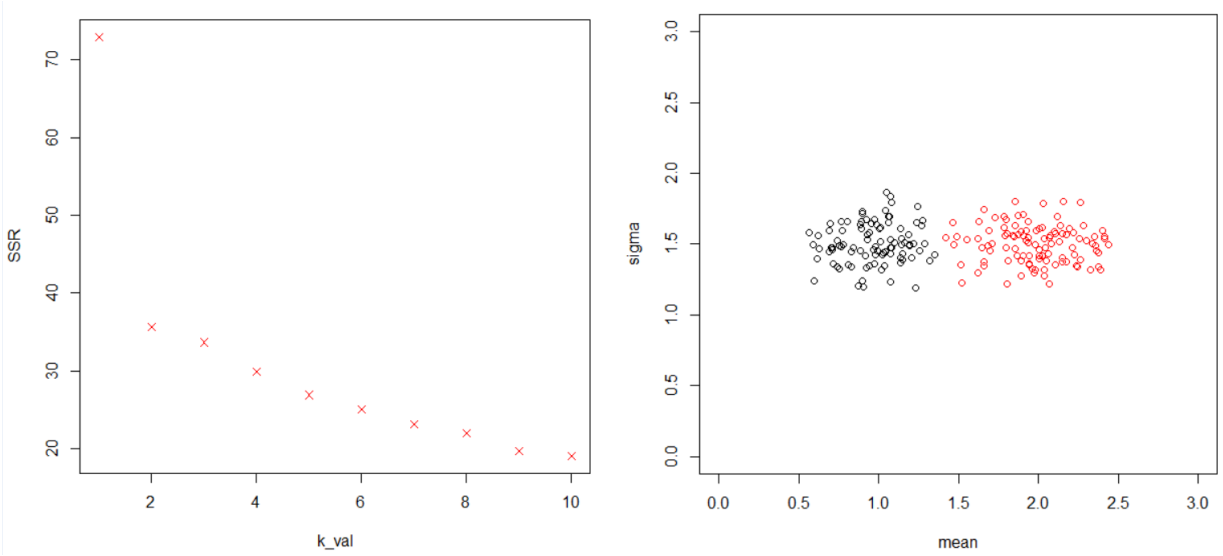
Figure 8.1 $(\mu_1, \sigma_1) = (1, 1.5)$ against $(\mu_2, \sigma_2) = (2, 1.5)$

$k = 2$ clusters.

After identifying the number of two clusters, we perform the hierarchical clustering upon this object. Figure 8.2b illustrates one such clustering. Compare this figure to Figure 8.1 and we can obtain the accuracy of this one-run of the algorithm. In this run, 193 out of 200 distributions are clustered correctly. 7 out of 200 distributions are clustered into the other family.

Then we test the k-means clustering based on the statistical distance. Using the same example, k-means achieves a lower mis-clustering rate (or a higher accuracy). Only 3 out of 200 elements are mis-clustered. Figure 8.3 illustrates the clustering result.

Finally, we perform the k-means clustering and hierarchical clustering



a. SSR against different number of clusters

b. Hierarchical Clustering based on Fisher-Rao Metric

Figure 8.2 Hierarchical Clustering

using the Euclidean distance. The graphs are omitted because they look similar to those using the statistical distance.

Using the idea from the Monte Carlo methods, we run the above experiments multiple times and record the *clustering accuracy*, defined as $1 - \frac{numberofmis-clustedelements}{numberoftotalelements}$. We report both the mean and the standard error. Table 8.2 summarizes this information for various methods. Note we use $k = 2$, $t = 100$, $n = 30$, $(\mu_1, \sigma_1) = (1, 1.5)$, $(\mu_2, \sigma_2) = (2, 1.5)$, and we repeat the experiments 100 times for each method. As one can observe from the table, in this low-dimensional case and with minimal k value, k-means algorithms based on different metrics perform better than the hierarchical methods. However, the same method using different metrics performs quite

Table 8.1 Results: Clustering when k = 2

Algorithm	Clustering Accuracy
Hierarchical Clustering with Fisher-Rao Metric	0.904 ± 0.006
Hierarchical Clustering with Euclidean Metric	0.922 ± 0.006
K-Means Clustering with Fisher-Rao Metric	0.965 ± 0.001
K-Means Clustering with Euclidean Metric	0.965 ± 0.01

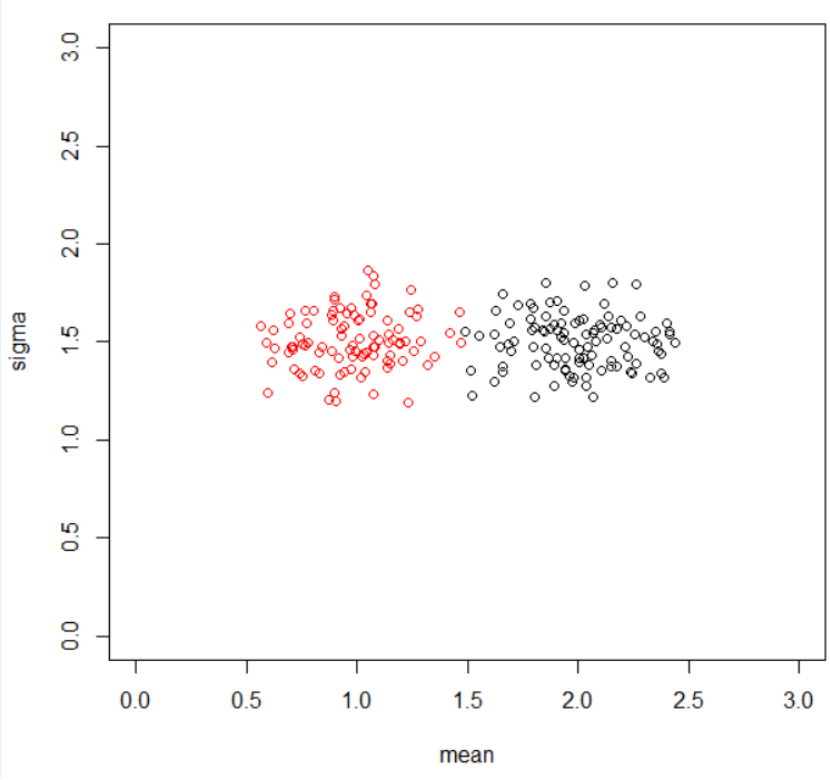


Figure 8.3 K-Means Clustering based on Fisher-Rao Metric

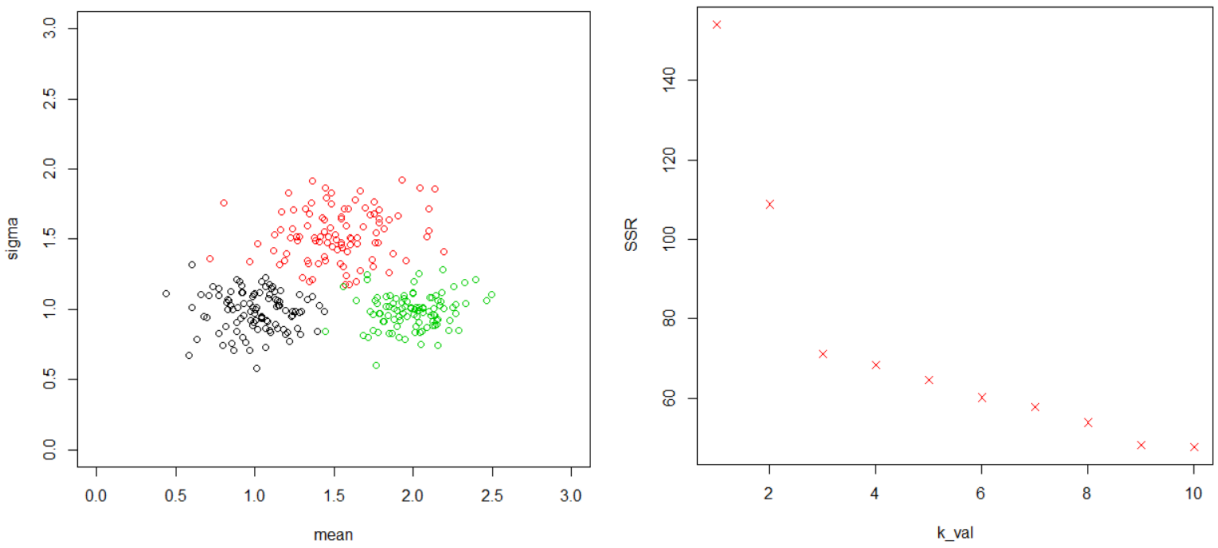
similarly.

8.2 Univariate Normal Distribution with k = 3

Next we analyze the case of univariate normal distribution when $k = 3$. This is the situation when clusters become more complicated and the Euclidean distance may not suffice. We illustrate again using a concrete example. Consider $(\mu_1, \sigma_1) = (1, 1)$, $(\mu_2, \sigma_2) = (1.5, 1.5)$, $(\mu_3, \sigma_3) = (2, 1)$. We run through a similar analysis as in the case of $k = 2$. Figure 8.4a illustrates the parameter space with the specified parameters.

First, we plot SSR against different k values to determine the number of clusters, as shown in Figure 8.4b. Again, the correct number of clusters is successfully identified, as indicated by the sharp drop at $k = 3$.

Next, we perform hierarchical clustering based on the Fisher-Rao metric



a. An object with parameters $(1, 1)$, $(1.5, 1.5)$, and $(2, 1)$

b. SSR against different number of clusters

Figure 8.4 Object and SSR

as well as the Euclidean metric. The result is shown in Figure 8.5. Note in this $k = 3$ case, hierarchical clustering based on the Fisher-Rao metric and Euclidean metric yields very different results. In fact, as one can tell, at least in this run, the Euclidean Metric yields a bad clustering result.

Finally, we perform k-means clustering based on the Fisher-Rao metric and the Euclidean metric. Figure 8.6 illustrates the result.

Similarly, we run the experiment 100 times and report the statistics. Note we use $k = 3$, $t = 100$, $n = 30$, $(\mu_1, \sigma_1) = (1, 1)$, $(\mu_2, \sigma_2) = (1.5, 1.5)$, and $(\mu_3, \sigma_3) = (2, 1)$.

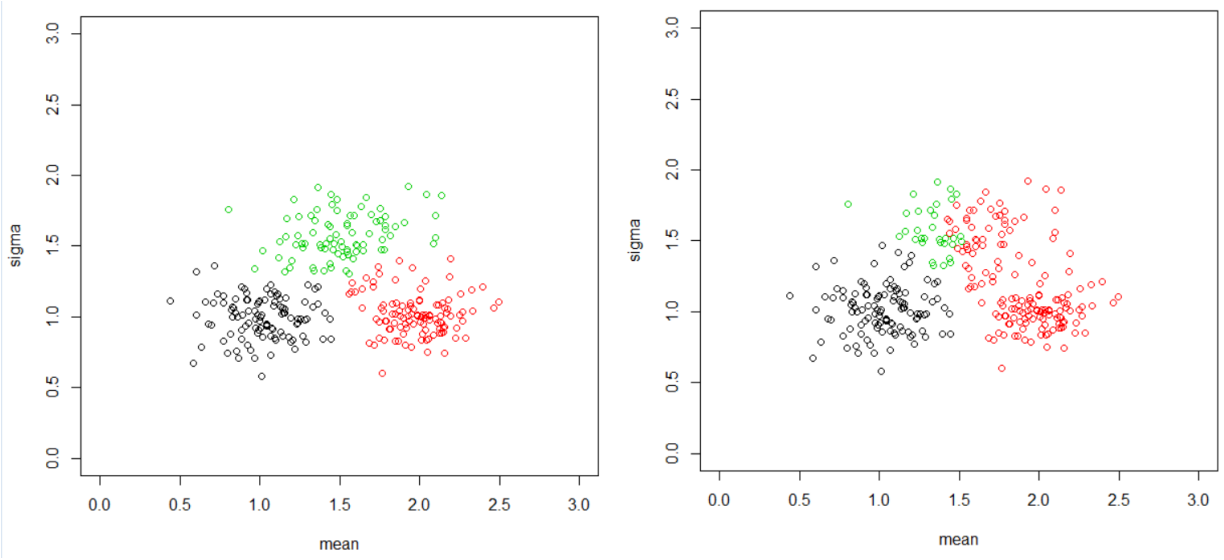
Table 8.2 Results: Clustering when k = 3

Algorithm	Clustering Accuracy
Hierarchical Clustering with Fisher-Rao Metric	0.905 ± 0.005
Hierarchical Clustering with Euclidean Metric	0.858 ± 0.007
K-Means Clustering with Fisher-Rao Metric	0.961 ± 0.001
K-Means Clustering with Euclidean Metric	0.940 ± 0.001

At first sight, using the same clustering methods, we see the Fisher-Rao metric in both cases (hierarchical clustering and k-means) yields better result.

In k-means clustering, we can see this improvement is significant in the sense that the difference is more than two standard errors. In the hierarchical methods, one can see the Euclidean-based method yields results with lower mean accuracy and higher standard error, while the same algorithm based on the Fisher-Rao metric gives higher accuracy and less variation.

When we look more closely into the results, the hierarchical method based on the Fisher-Rao metric outperforms the same method based on the Euclidean metric 70/100 times. More astoundingly, the k-means clustering based on the Fisher-Rao metric outperforms the Euclidean-based k-means clustering algorithm 98/100 times. These are good evidence that as the clustering becomes more complicated (number of clusters from 2 to 3), algorithms based on the statistical distance become more favorable.

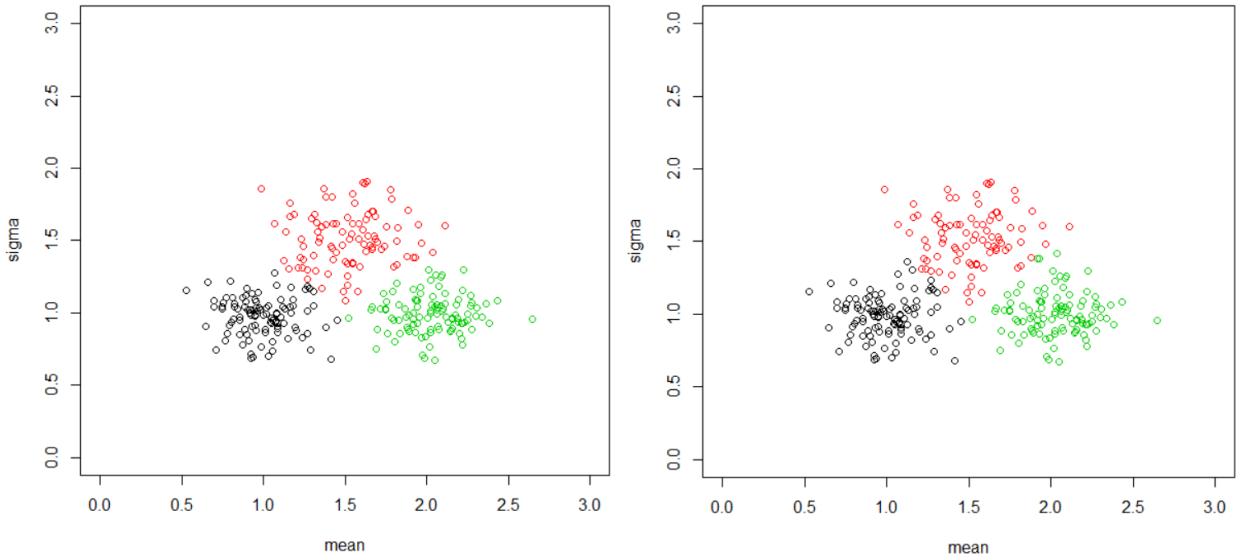


a. Fisher-Rao Metric **b.** Euclidean Metric

Figure 8.5 Hierarchical Clustering when k = 3

8.3 Bivariate Normal Distribution in Three-Dimensional Parameter Space

Next, we want to explore the algorithms in a higher-dimensional setting. We use the bivariate normal distribution as an illustrative example. In particular,



a. Fisher-Rao Metric

b. Euclidean Metric

Figure 8.6 K-Means Clustering when k = 3

we consider the case where $(\mu_1, \mu_2) = (\mu, \mu)$ and the covariance matrix is diagonal:

$$\Sigma = \begin{bmatrix} \sigma_1^2 & 0 \\ 0 & \sigma_2^2 \end{bmatrix} \tag{8.1}$$

Hence, we parameterize each bivariate normal distribution with a triple $(\mu, \sigma_1, \sigma_2)$. Note we adopt this seemingly restrictive setting because the parameter space $(\mu, \sigma_1, \sigma_2)$ is three-dimensional and therefore can be visualized. In general, we do not need to assume $\mu_1 = \mu_2$, or the correlation between x_1 and x_2 is 0. Here we adopt this setting just for the purpose of visualization.

First, we generate an object for clustering in the similar fashion as in the univariate normal distribution case. An example of such an object in the parameter space is illustrated in Figure 8.7. Note in Figure 8.7, $k = 3$, $t = 100$, $n = 30$, and $(\mu, \sigma_1, \sigma_2) = (1, 1, 2), (1.5, 1.5, 2.5), (2, 2, 3)$, respectively for the distribution underlying each cluster.

The Fisher-Rao metric between two bivariate normal distribution with diagonal covariance matrix is a special case of the formula given in 8.2, with $p = 2$, $\boldsymbol{\theta} = (\mu, \sigma_1, \sigma_2)$, and $\mu_{ki} = \mu_k$:

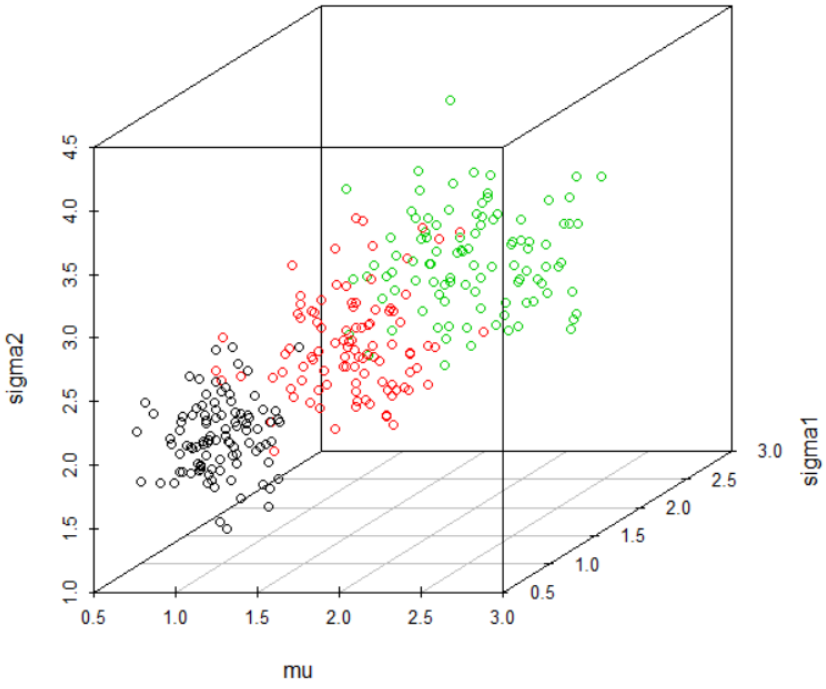
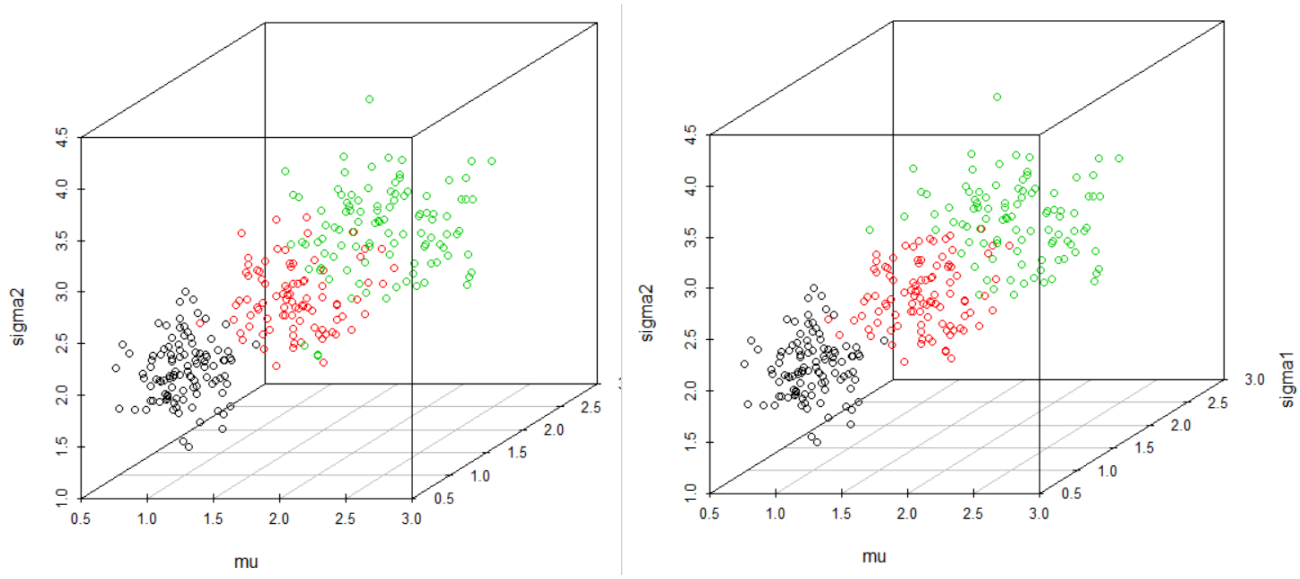


Figure 8.7 Clusters Generated for Bivariate Normal Distribution k = 3

$$d_F(\boldsymbol{\theta_1}, \boldsymbol{\theta_2}) = \sqrt{2\sum_{i=1}^{2}(\ln \frac{\|(\frac{\mu_1}{\sqrt{2}}, \sigma_{1i}) - (\frac{\mu_2}{\sqrt{2}}, -\sigma_{2i})\| + \|(\frac{\mu_1}{\sqrt{2}}, \sigma_{1i}) - (\frac{\mu_2}{\sqrt{2}}, \sigma_{2i})\|}{\|(\frac{\mu_1}{\sqrt{2}}, \sigma_{1i}) - (\frac{\mu_2}{\sqrt{2}}, -\sigma_{2i})\| - \|(\frac{\mu_1}{\sqrt{2}}, \sigma_{1i}) - (\frac{\mu_2}{\sqrt{2}}, \sigma_{2i})\|})^2} \tag{8.2}$$

Equipped with a closed form Fisher-Rao metric, we can now perform the hierarchical clustering based on both the Euclidean distance and the Fisher-Rao metric as before. We can likewise formulate the k-means clustering based on these two different metrics. Figure 8.8 and 8.9 give illustrations of a sample output from each algorithm.

To gain further insight into the algorithm performances, we run experiments on a larger scale. Here, we record the result obtained from replicating the experiment 100 times, with parameters $k = 3$, $t = 100$, $n = 30$, and $(\mu, \sigma_1, \sigma_2) = (1, 1, 2), (1.5, 1.5, 2.5), (2, 2, 3)$, respectively for each cluster.



a. Fisher-Rao Metric

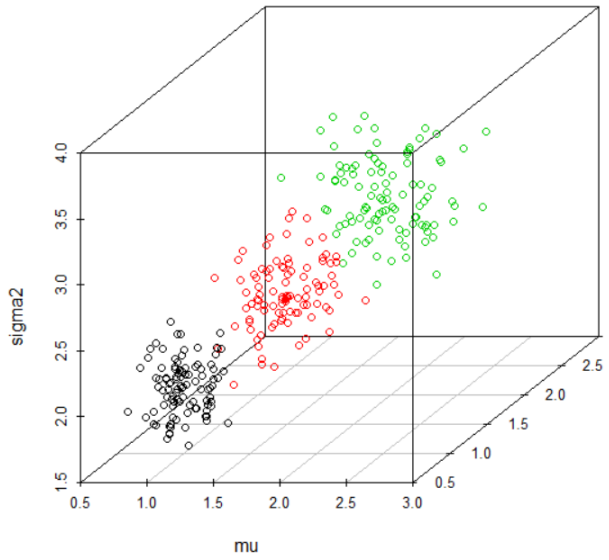
b. Euclidean Metric

Figure 8.8 Hierarchical Clustering when k = 3

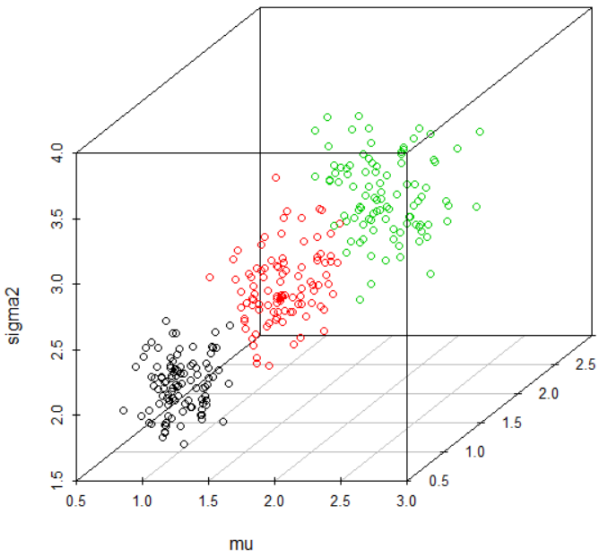
Table 8.3 Bivariate Normal Distribution Results: Clustering when k = 3

Algorithm	Clustering Accuracy
Hierarchical Clustering with Fisher-Rao Metric	0.860 ± 0.008
Hierarchical Clustering with Euclidean Metric	0.716 ± 0.012
K-Means Clustering with Fisher-Rao Metric	0.937 ± 0.001
K-Means Clustering with Euclidean Metric	0.877 ± 0.003

The results mimic those for the case of the univariate normal distribution with $k = 3$. However, we do observe that as the dimension gets higher, the same algorithms with the Fisher-Rao metric have further improved performance compared to those with the Euclidean distance. Also, the k-means clustering has overall better performance than the hierarchical methods. In fact, the k-means clustering equipped with the Fisher-Rao metric outperforms other alternative algorithms by a large margin.



a. Fisher-Rao Metric



b. Euclidean Metric

Figure 8.9 K-Means Clustering when k = 3

Chapter 9

Empirical Results: Clustering Discrete Poisson Distributions

In this chapter, we discuss the empirical results from clustering algorithms applied on discrete Poisson distributions.

9.1 Discrete Poisson Distribution with the Hellinger Distance

Next, we delve into the realm of discrete distributions. Recall in the cases of univariate and bivariate normal distributions, the statistical distance is given as a closed-form expression of the parameters. This is not the case in general. Here we look at the case where the statistical distance is given as an expression of two probability distributions, instead of a closed form expression of the parameters.

We use the Poisson distribution as our running example here. Recall the Poisson distribution has the following probability mass function:

$$P(x = k) = \frac{\lambda^k \exp(-\lambda)}{k!} \tag{9.1}$$

where λ is the mean and variance of the distribution. Figure 9.1 gives an illustration of Poisson distributions with different λ.

Note the Poisson distribution is parametrized by one parameter λ. One thing we can do is to estimate this parameter λ and use the estimates to cluster. Alternatively, we can measure the distance between two empirical probability mass functions using the Hellinger distance and run the clustering algorithms

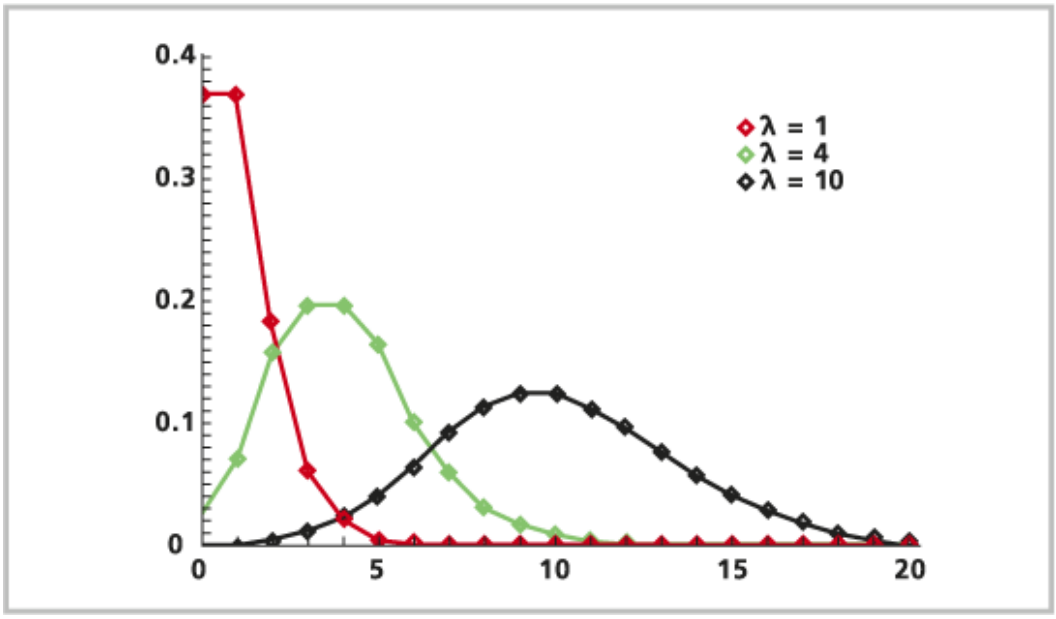


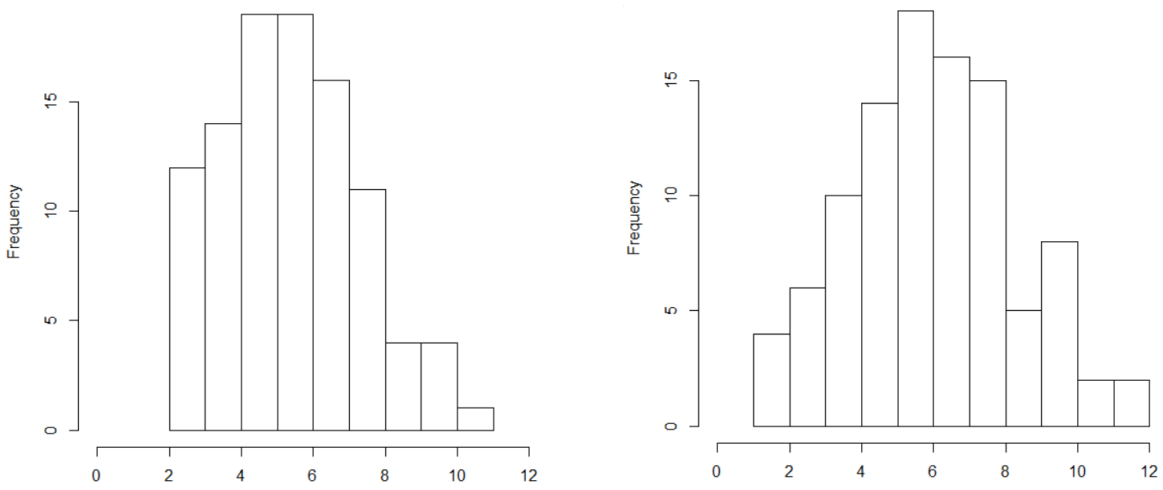
Figure 9.1 Poisson Distribution with Different Parameter

on them. The formula for the Hellinger distance is given in Equation 5.1. Again, we will compare the Hellinger-distance-based algorithms with the Euclidean-distance-based algorithms.

We use a concrete example to have a feeling of the difference between two metrics. Figure 9.2a and 9.2b illustrate two empirical Poisson distributions, one with $\lambda = 6$ and the other $\lambda = 6.5$. After normalizing them appropriately, we can calculate the distance between them. According to the Hellinger distance, these two histograms have a distance of 0.167, while the Euclidean distance between them is 0.990.

Now we introduce another empirical distribution, again with $\lambda = 6$, as shown in Figure 9.3. We compare Figure 9.3 with Figure 9.2a and 9.2b respectively using both the Hellinger and the Euclidean distance. In fact the Hellinger distance between 9.2a and 9.3 is 0.143, while the Hellinger distance between 9.2b and 9.3 is 0.152. As we can see, in this case, the Hellinger distance does capture the nuance and yields the expected result. On the other hand, the Euclidean distance between 9.2a and 9.3 is 0.114, while that between 9.2b and 9.3 is 0.0970. According to the Euclidean distance, the empirical distributions with different λ are closer, which is counterintuitive.

As in the cases of continuous distributions, we run hierarchical clustering



a. Empirical Poisson Distribution $\lambda = 6$ **b.** Empirical Poisson Distribution $\lambda = 6.5$

Figure 9.2 Two Empirical Poisson Distributions

and k-means clustering algorithms with both the Hellinger and Euclidean distances on simulated clusters. The cluster generating process is very similar to the process described in Section 7.2. Again, we replicate the experiment 100 times and record the results in Table 9.1. In this experiment, we have $k = 3$ underlying Poisson distributions, each with $\lambda_1 = 6$, $\lambda_2 = 8$, and $\lambda_3 = 10$. Each cluster contains 100 empirical distributions and each empirical distribution contains 30 data points. The similar pattern as in the continuous cases occurs. This time, the hierarchical method with the Hellinger distance outperforms that with the Euclidean distance by a very large margin. K-means clustering with the Hellinger distance also outperforms that with the Euclidean distance in a statistically significant way.

Table 9.1 Results: Clustering of Empirical Poisson Distributions

Algorithm	Clustering Accuracy
Hierarchical Clustering with Hellinger distance	0.792 ± 0.010
Hierarchical Clustering with Euclidean distance	0.674 ± 0.008
K-Means Clustering with Hellinger distance	0.922 ± 0.004
K-Means Clustering with Euclidean distance	0.901 ± 0.004

We can also look at the above statistics in greater details. 83 out of 100 runs, the hierarchical method with the Hellinger distance outperforms that

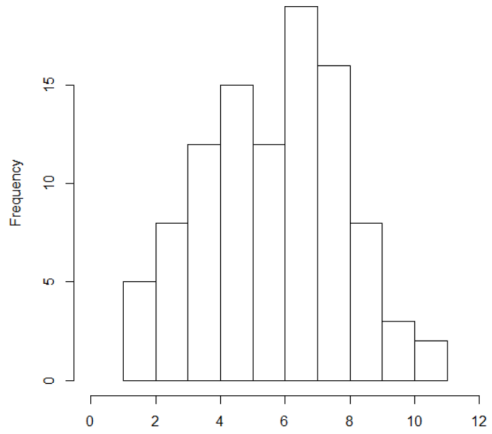
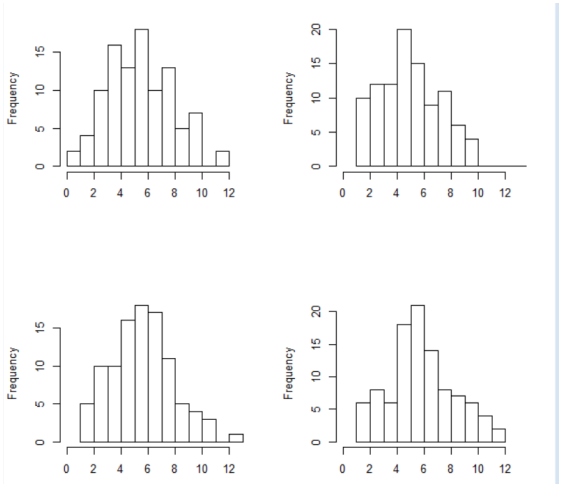


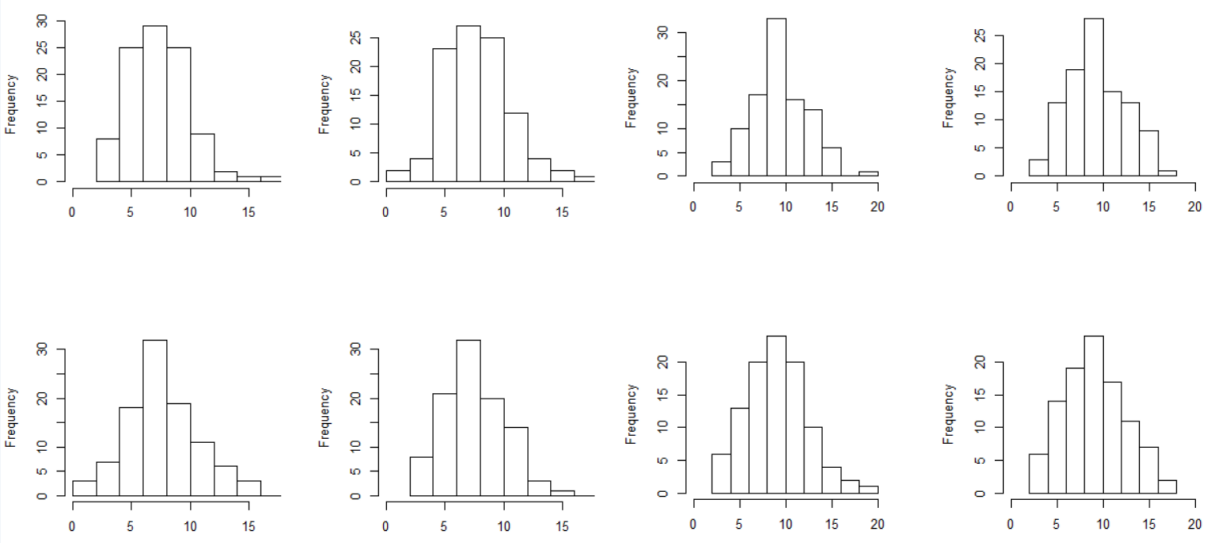
Figure 9.3 Another Empirical Poisson Distribution $\lambda = 6$

with the Euclidean distance. For k-means clustering, this statistic is 76 out of 100.

Unlike continuous distributions, we cannot visualize the clustering directly. To give readers a sense of the result, we pick some elements from each cluster determined by the k-means clustering with the Hellinger distance (See Figure 9.4, 9.5a, and **??**). Here, we use 100 data points in each empirical Poisson distribution. Other parameters are the same as before.

Figure 9.4 Representatives from the First Cluster





a. Representatives from the Second Cluster

b. Representatives from the Third Cluster

Figure 9.5 Representatives from the Clusters

9.2 Multidimensional Scaling

A technique closely related to our clustering algorithms is known as the multidimensional scaling, or MDS. Multidimensional scaling (MDS) is a technique that creates a map displaying the relative positions of a number of objects, given the dissimilarity matrix. The map may consist of one, two, three, or more dimensions. Note the goal of MDS is to preserve the between-object distance as well as possible.

In general, we consider a collection of objects (in our cases probability distributions) and form the dissimilarity matrix Δ where each entry δ_{ij} represents the dissimilarity between i^{th} and j^{th} object. The goal of MDS is to find vectors $\mathbf{x_1}, \mathbf{x_2}, ..., \mathbf{x_n}$ in $\mathbb{R}^d$ such that each vector $\mathbf{x_i}$ represents the i^{th} object and the following is true:

$$\|\mathbf{x_i} - \mathbf{x_j}\| \approx \delta_{ij}$$

In other words, MDS tries to find a lower-dimensional embedding from the statistical manifold to the Euclidean space $\mathbb{R}^d$. Note it is important to notice here it is often impossible to preserve all the between-object distances exactly. However, the advantage is obvious. By choosing the dimension to

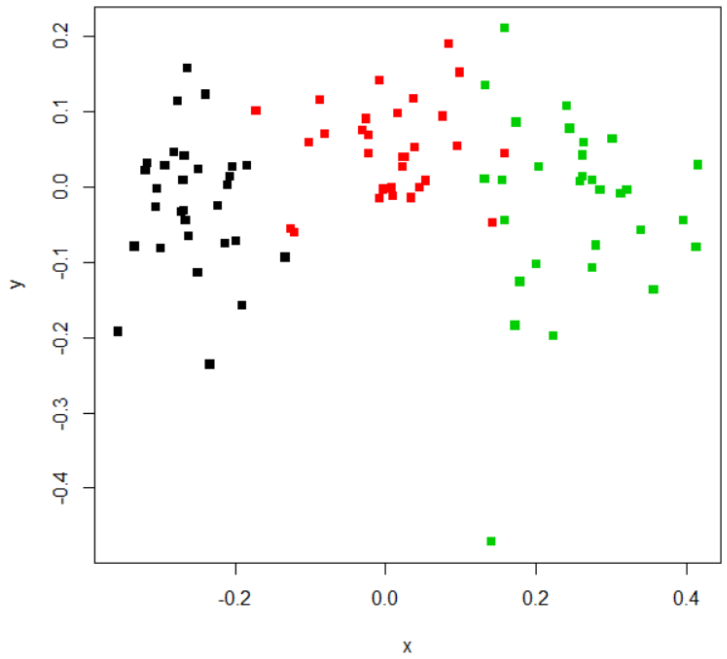
be 2 or 3, we can visualize the distance between objects.

The connection between MDS and the statistical-distance-based clustering algorithms we proposed is clear. Running a k-means clustering using the statistical distance is approximately applying MDS algorithm to the probability distributions first and then running a Euclidean-distance-based k-means clustering algorithm on the lower-dimensional embedding.

The advantage of running a statistical-distance-based k-means algorithm directly is also easy to see. MDS algorithm only creates a lower-dimensional approximation that does not capture all the nuance of the between-object dissimilarities. Moreover, as we will prove in the next chapter, statistical-distance-based k-means algorithm can converge to the local minimum directly and is convenient to implement.

However, in the case of clustering discrete Poisson distributions, the problem is of very high dimension and MDS can help us visualize. Figure 9.6 gives such an example. Note each point in the figure represents a discrete Poisson distribution. The Euclidean distance between each pair of points numerically is approximately equal to the Hellinger distance.

Figure 9.6 MDS with Discrete Poisson Distributions with $\lambda = 6, 8, 10$



Chapter 10

Results: Convergence and Optimality

In this section, we analyze the convergence property of the k-means algorithms proposed in Section 6.3. K-means-type algorithms are usually formulated as a mathematical program in the following way Selim and Ismail (1984):

$$\begin{aligned} P: \textit{minimize} \quad & P(W,Z) = \sum_{i=1}^{k}\sum_{j=1}^{m} w_{ij} D(x_j, z_i) \\ \textit{subject to} \quad & \sum_{i=1}^{k} w_{ij} = 1, \quad j = 1,2,...,m \\ & w_{ij} = 0 \quad or \quad 1, \quad j = 1,2,...,m; \qquad i = 1,2,...,k \end{aligned} \tag{10.1}$$

where $x_1, x_2, ..., x_m$ are objects to be clustered, $z_1, z_2, ..., z_k$ are centroids of each cluster, w_{ij} assigns each x_j to each cluster with centroid z_i, and $D(\cdot,\cdot)$ is the measure of dissimilarity. We will refer to the mathematical program 10.1 as Problem P from this point.

Using the above notation, k-means algorithm can be interpreted in the following way Selim and Ismail (1984):

1. Start with initial cluster centers $z_i^1, i = 1,2,...,k$. Set $l = 1$.

2. Assign each $x_j, j = 1,2,...,m$ to its nearest cluster center, which is equivalent to fixing w_{ij}.

3. Recompute the centers z_i^l by minimizing $F(\cdot, Z)$. If $z_i^{l-1} = z_i^l$, that is, the centers are fixed, then stop. Otherwise go to the last step.

To facilitate our later discussion, we make the following definition.

Definition 1 (Partial Optimal Solution)**.** A point $(W^\star, Z^\star)$ is a partial optimal solution of P if it satisfies the following:

$$\begin{aligned} P(W^\star, Z^\star) &\leq P(W, Z^\star) \quad \textit{for all } W \\ P(W^\star, Z^\star) &\leq P(W^\star, Z) \quad \textit{for all } Z \end{aligned} \tag{10.2}$$

Note a partial optimal solution satisfies the famous Kuhn-Tucker Conditions of Problem P, but it may not be a local minimum Selim and Ismail (1984).

It has been shown in Selim and Ismail (1984) that the usual k-means algorithm converges to a partial optimal solution of Problem P in a finite number of iterations. Therefore, we know k-means algorithm necessarily converges and it will always converge to a partial optimal solution. In the rest of this chapter, we will try to make the link between the partial optimal solution and the local minimum.

We record here a very important theorem proved in Selim and Ismail (1984). First, we give a useful definition.

Definition 2 ($A(W^\star)$)**.**

$$A(W^\star) = \{Z : Z \quad \textit{minimizes} \quad f(W^\star, Z)\} \tag{10.3}$$

Equipped with the above definition, we are ready to state the theorem that connects the partial optimality and local optimality.

Theorem 1 (Partial Optimality and Local Optimality)**.** Suppose $(W^\star, Z^\star)$ is a partial optimal solution for Problem P and $A(W^\star)$ is a singleton, then $W^\star$ is a local minimum for Problem P.

Now we may explore the relationship between the dissimilarity measure $D(\cdot, \cdot)$ and the local optimality. In the rest of this chapter, we show the k-means clustering with the Hellinger distance applied on the empirical Poisson distributions converges to a local minimum. The discrete version of the Hellinger distance was defined in Equation 5.1. We reproduce it below for the sake of completeness.

Definition 3 (Discrete Hellinger Distance)**.** The Hellinger distance between $P = (p_0, p_1, ..., p_k)$ and $Q = (q_0, q_1, ..., q_k)$ is given by

$$H^2(P, Q) = \frac{1}{2} \sum_{i=0}^{k} (\sqrt{p_i} - \sqrt{q_i})^2 \tag{10.4}$$

Recall we want to minimize

$$P(W, Z) = \sum_{i=1}^{k} \sum_{j=1}^{m} w_{ij} D(x_j, z_i) \tag{10.5}$$

Let

$$f_i(W_i, Z_i) = \sum_{j=1}^{m} w_{ij} D(x_j, z_i) \tag{10.6}$$

where W_i is the i^{th} row of W and z_i is the centroid of the i^{th} cluster.

We want to show $A(W^\star)$ is a singleton. First we define

$$A_i(W_i^\star) = \{z_i : z_i \quad minimizes \quad f_i(W_i^\star, z_i)\} \tag{10.7}$$

It is obvious to see that $A(W^\star)$ is a singleton if and only if each $A_i(W_i^\star)$ is a singleton Selim and Ismail (1984). For the simplicity of the following analysis, we simplify the subscript a little bit. We fix an i and let $w_{ij} = a_j$, $z_i = z$. Therefore, Equation 10.7 becomes:

$$A_i(W_i^\star) = \{z : z \quad minimizes \quad \sum_{j=1}^{m} a_j D(x_j, z)\} \tag{10.8}$$

We want to show each $A_i(W_i^\star)$ is a singleton, or equivalently, there is a unique centroid z that minimizes $\sum_{j=1}^{m} a_j D(x_j, z)$. We begin to search for the minimizer z. First we plug in the expression for the Hellinger distance and obtain the following equivalences:

$$
\begin{aligned}
& \textit{minimize} \quad \sum_{j=1}^{m} a_j \sum_{t} (\sqrt{x_{jt}} - \sqrt{z_t})^2 \\
= \; & \textit{minimize} \quad \sum_{j=1}^{m} \sum_{t} a_j (\sqrt{x_{jt}} - \sqrt{z_t})^2 \\
= \; & \textit{minimize} \quad \sum_{j=1}^{m} \sum_{t} a_j (x_{jt} + z_t - 2\sqrt{x_{jt}}\sqrt{z_t}) \\
= \; & \textit{minimize} \quad \sum_{j=1}^{m} \sum_{t} a_j x_{jt} + \sum_{j=1}^{m} \sum_{t} a_j z_t - \sum_{j=1}^{m} \sum_{t} 2\sqrt{x_{jt}}\sqrt{z_t} a_j
\end{aligned}
\tag{10.9}
$$

Note in Equation 10.9, t indexes the length of each vector x_j and the centroid z. Note a_j and x_{jt} are fixed and let $\sqrt{z_t} = y_t$. Equation 10.9 further simplifies to

$$
\begin{aligned}
& \textit{minimize} \quad \sum_{j=1}^{m} \sum_{t} a_j x_{jt} + \sum_{j=1}^{m} \sum_{t} a_j z_t - \sum_{j=1}^{m} \sum_{t} 2\sqrt{x_{jt}}\sqrt{z_t} a_j \\
= \; & \textit{minimize} \quad \sum_{j=1}^{m} \sum_{t} a_j z_t - \sum_{j=1}^{m} \sum_{t} 2\sqrt{x_{jt}}\sqrt{z_t} a_j \\
= \; & \textit{minimize} \quad \sum_{j=1}^{m} \sum_{t} a_j y_t^2 - \sum_{j=1}^{m} \sum_{t} 2 a_j \sqrt{x_{jt}} y_t \\
= \; & \textit{minimize} \quad \sum_{t} (\sum_{j=1}^{m} a_j) y_t^2 - 2 \sum_{t} (\sum_{j=1}^{m} a_j \sqrt{x_{jt}}) y_t
\end{aligned}
\tag{10.10}
$$

Let $Q = \sum_t (\sum_{j=1}^{m} a_j) y_t^2 - 2 \sum_t (\sum_{j=1}^{m} a_j \sqrt{x_{jt}}) y_t$. To minimize Q, it suffices to take partial derivative with respect to each y_t and set it to 0:

$$
0 = \frac{\partial Q}{\partial y_t} = (\sum_{j=1}^{m} a_j) y_t - 2(\sum_{j=1}^{m} a_j \sqrt{x_{jt}}) \tag{10.11}
$$

Hence, we obtain the optimal $y_t^\star$:

$$
y_t^\star = \frac{2 \sum_{j=1}^{m} a_j \sqrt{x_{jt}}}{\sum_{j=1}^{m} a_j} \tag{10.12}
$$

Accordingly, the optimal $z_t^\star = (y_t^\star)^2$:

$$z_t^\star = \left(\frac{2\sum_{j=1}^{m} a_j \sqrt{x_{jt}}}{\sum_{j=1}^{m} a_j}\right)^2 \tag{10.13}$$

Technically, $z_t = z_{it}$ as we fixed i to simplify the derivation. Knowing z_{it} for each i and t, we can recover $Z = (z_1, z_2, ..., z_k)$, where $z_i = (z_{i1}, z_{i2}, ..., z_{it})$. Indeed, the minimizer is unique and we have shown using the k-means clustering with the Hellinger distance not only converges to the partial optimal solution, but also to the local minimum.

Chapter 11

Discussion and Conclusion

In this thesis, we generalize some most widely-used machine learning algorithms to the statistical manifold. Instead of classifying and clustering vectors in the Euclidean space, we develop algorithms to classify and cluster the probability distributions.

We first study the problem of classifying probability distributions by formulating an analogous optimal separating hyperplane algorithm on the statistical manifold. The key of this generalization is to define a decision boundary on the statistical manifold. We do so by finding the line connecting two points in the parameter space and using the corresponding family of distributions as the decision boundary on the statistical manifold. We also utilize the parameter space to specify our separability conditions. However, when we calculate the margin, we do so using the statistical distance. We draw two important conclusions. First, we should always think about the statistical manifold and the associated parameter space when formulating the problem. Second, the optimization algorithm should always involve both spaces and the connection between two spaces is captured by our definition of the separating line on the statistical manifold.

Next, we focus on clustering the probability distributions. We first make a distinction between a centroid on a manifold and a centroid on a sub-manifold. When working with a sub-manifold of the statistical manifold, say the set of univariate normal distributions, we define a centroid on the sub-manifold by computing the centroid on the parameter space and then map it back to the sub-manifold. This notion of a centroid is directly applicable in the k-means clustering algorithm and is also used when we compute the internal quality criteria in hierarchical methods.

The generalization of the k-means clustering and the hierarchical methods

to the statistical manifold is straightforward to implement. We first propose a situation where clustering distributions is useful. Underlying each cluster of probability distributions lies a true, prescribed distribution. Each probability distribution in the cluster is one empirical distribution, created by drawing samples from the underlying true distribution. We can visualize the clusters by looking at the associated parameter space.

We generate artificial clusters to test our algorithms and compare them with some alternatives. We summarize the results from the empirical studies here. First we look at the clustering of univariate normal distributions. Algorithms we test and compare include the hierarchical method with Fisher-Rao and Euclidean metric and the k-means clustering with Fisher-Rao and Euclidean metric. When the problem is indeed simple, with only $k = 2$ clusters, there is no evidence that statistical-distance-based methods outperform the methods with Euclidean distance. In fact, all algorithms give very similar output.

Things become more interesting when we have $k = 3$ clusters. We test and compare the same algorithms and we have found there is evidence that both hierarchical and k-means methods with the Fisher-Rao metric outperform the same algorithms with the Euclidean metric. Moreover, this increase in the average performance is statistically significant. When we look more closely at the results, 70 out of 100 times the hierarchical method with the Fisher-Rao metric outperforms the same algorithm with the Euclidean metric and 98 out of 100 times we observe the same pattern in k-means algorithms. We conclude there is evidence suggesting the clustering algorithms based on the Fisher-Rao metric is superior to the same algorithm based on the Euclidean distance in the case of univariate normal distributions with $k = 3$ clusters. Moreover, we conjecture this is generally the case for more than 3 clusters.

Next, we explore the higher-dimensional parameter space. We study empirically the bivariate normal distributions with the one dimensional mean vector and the diagonal covariance matrix. The associated parameter space in this setting is three-dimensional. Using the same set of algorithms, we find the difference between the same algorithm with the different distance is much larger now. Hierarchical method with the Fisher-Rao metric achieves 0.860 ± 0.008 accuracy, compared to a 0.716 ± 0.012 using the Euclidean metric. Similarly, the k-means method with the Fisher-Rao metric achieves 0.937 ± 0.001, a large boost compared to the 0.877 ± 0.003 accuracy using the Euclidean metric. Again, we conjecture this is generally true for parameter space with higher dimensions.

In both the univariate and bivariate normal cases, the statistical distance is given as a closed form expression of the parameters of the probability distribution. However, in general, this does not need to be the case. In the last part of the empirical study, we apply our algorithms to the discrete Poisson distributions. Note the Poisson distribution is parametrized by one parameter λ. We use the Hellinger distance to quantify the distance between two empirical Poisson distributions. Note here the Hellinger distance is not a function of λ; instead, it makes use of the probability vector directly. Again, in this case, we observe the similar pattern: algorithms using the statistical distance (the Hellinger distance) outperforms the Euclidean-distance-based algorithms.

The last part of the paper establishes one convergence result. We have shown the k-means algorithm applied on the discrete distributions using the Hellinger distance converges not only to the partial optimal solution but also to the local minimum.

Chapter 12

Future Work

This thesis project can be taken in many possible directions. For one thing, although we propose an analog of an optimal separating hyperplane algorithm to classify the probability distributions, we have not implemented it. The main difficulty we encounter in the implementation is that our proposed optimization problem is complicated and not necessarily convex. To continue in this direction, one may try to investigate when the algorithm is convex, if at all. Also, the optimization problem depends on the statistical distance one uses and this choice of distance can affect the convexity of the problem.

Another direction of the future work is to conduct the empirical study more systematically. An accompanying code base has been developed for the purpose of clustering using the statistical distance. In this thesis, empirical results with univariate normal and bivariate normal distributions are presented. However, we compare the algorithms using one fixed set of parameters. In particular, the underlying true distributions are picked arbitrarily. It is meaningful to repeat the experiments on different parameter values and analyze whether or not the obtained results are sensitive to the changes in one or more parameters.

In particular, parameter n is the number of samples we use to construct the empirical distribution and it has a profound impact on the separability of the clusters. Intuitively, the more compact clusters are, the easier it is for algorithms to cluster. Hence, it is meaningful to study the algorithm performance with respect to the separability of the clusters generated. Moreover, one may conduct further empirical studies to verify or reject our two conjectures, namely:

- In the univariate normal case, when $k \geq 2$, the statistical-distance-based clustering algorithms on average outperform the Euclidean-distance-based ones.
- The statistical-distance-based clustering algorithms on average outperform the Euclidean-distance-based ones when the associated parameter space has higher dimension than 2.

Finally, one may try to prove more convergence results regarding our k-means-like algorithms. In this thesis, we have shown k-means-algorithm applied on the discrete distributions with the Hellinger distance converges to the local minimum. On can try to show the similar results for the same algorithm with different statistical distance, for instance, the Fisher-Rao metric. One may also be interested in proving our k-mean-like algorithms not only converge to the local minimum, but also the global minimum.

Bibliography

Amari, Shun-ichi. 2013. *Information Geometry and Its Applications: Survey*, 3–3. Berlin, Heidelberg: Springer Berlin Heidelberg. doi:10.1007/978-3-642-40020-9_1. URL http://dx.doi.org/10.1007/978-3-642-40020-9_1.

Cieslak, David A., T. Ryan Hoens, Nitesh V. Chawla, and W. Philip Kegelmeyer. 2012. Hellinger distance decision trees are robust and skew-insensitive. *Data Mining and Knowledge Discovery* 24(1):136–158. doi:10.1007/s10618-011-0222-1. URL http://dx.doi.org/10.1007/s10618-011-0222-1.

Costa, Sueli I.R., Sandra A. Santos, and Joao E. Strapasson. 2015. Fisher information distance: A geometrical reading. *Discrete Applied Mathematics* 197:59 – 69. doi:http://dx.doi.org/10.1016/j.dam.2014.10.004. URL http://www.sciencedirect.com/science/article/pii/S0166218X14004211. Distance Geometry and Applications.

Hastie, Trevor J., Robert John Tibshirani, and Jerome H. Friedman. 2009. *The elements of statistical learning : data mining, inference, and prediction*. Springer series in statistics, New York: Springer. URL http://opac.inria.fr/record=b1127878. Autres impressions : 2011 (corr.), 2013 (7e corr.).

Lee, S. M., A. L. Abbott, and P. A. Araman. 2007. Dimensionality reduction and clustering on statistical manifolds. In *2007 IEEE Conference on Computer Vision and Pattern Recognition*, 1–7. doi:10.1109/CVPR.2007.383408.

Lin, J. 2006. Divergence measures based on the shannon entropy. *IEEE Trans Inf Theor* 37(1):145–151. doi:10.1109/18.61115. URL http://dx.doi.org/10.1109/18.61115.

Murphy, Kevin P. 2012. *Machine Learning: A Probabilistic Perspective*. The MIT Press.

Selim, Shokri Z., and M. A. Ismail. 1984. K-means-type algorithms: A generalized convergence theorem and characterization of local optimality. *IEEE Transactions on Pattern Analysis and Machine Intelligence* PAMI-6(1):81–87. doi:10.1109/tpami.1984.4767478.