Chapter 2

Vector Quantization and Principal Component Analysis

Most of the unsupervised learning algorithms originate from one of the two basic unsupervised learning models, vector quantization and principal component analysis. In particular, principal curves are related to both areas: conceptually, they are originated from principal component analysis whereas practical methods to estimate principal curves often resemble to basic vector quantization algorithms. This chapter describes these two models.

2.1 Vector Quantization

Vector quantization is an important topic of information theory. Vector quantizers are used in lossy data compression, speech and image coding [GG92], and clustering [Har75]. Vector quantization can also be considered as the simplest form of unsupervised learning where the manifold to fit to the data is a set of vectors. Kohonen's self-organizing map [Koh97] (introduced in Section 3.2.2) can also be interpreted as a generalization of vector quantization. Furthermore, our new definition of principal curves (to be presented in in Section 4.1) has been inspired by the notion of an *optimal vector quantizer*. One of the most widely used algorithms for constructing locally optimal vector quantizers for distributions or data sets is the Generalized Lloyd (GL) algorithm [LBG80] (also known as the *k*-means algorithm [Mac67]). Both the HS algorithm (Section 3.1.1) and the polygonal line algorithm (Section 5.1) are similar in spirit to the GL algorithm. This section introduces the concept of optimal vector quantization and describes the GL algorithm.

2.1.1 Optimal Vector Quantizer

A k-point vector quantizer is a mapping $q : \mathbb{R}^d \to \mathbb{R}^d$ that assigns to each input vector $\mathbf{x} \in \mathbb{R}^d$ a *codepoint* $\hat{\mathbf{x}} = q(\mathbf{x})$ drawn from a finite *codebook* $\mathcal{C} = \{\mathbf{v}_1, \ldots, \mathbf{v}_k\} \subset \mathbb{R}^d$. The quantizer is completely described by the codebook $\mathcal{C}$ together with the partition $\mathcal{V} = \{V_1, \ldots, V_k\}$ of the input space where $V_\ell = q^{-1}(\mathbf{v}_\ell) = \{\mathbf{x} : q(\mathbf{x}) = \mathbf{v}_\ell\}$ is the set of input vectors that are mapped to the ℓth codepoint by q.

The distortion caused by representing an input vector $\mathbf{x}$ by a codepoint $\hat{\mathbf{x}}$ is measured by a non-negative *distortion measure* $\Delta(\mathbf{x}, \hat{\mathbf{x}})$. Many such distortion measures have been proposed in different areas of application. For the sake of simplicity, in what follows, we assume that $\Delta(\mathbf{x}, \hat{\mathbf{x}})$ is the most widely used squared error distortion, that is,

$$\Delta(\mathbf{x}, \hat{\mathbf{x}}) = \|\mathbf{x} - \hat{\mathbf{x}}\|^2. \tag{5}$$

The performance of a quantizer q applied to a random vector $\mathbf{X} = (X_1, \ldots, X_d)$ is measured by the expected distortion,

$$\Delta(q) = E[\Delta(\mathbf{X}, q(\mathbf{X}))] \tag{6}$$

where the expectation is taken with respect to the underlying distribution of $\mathbf{X}$. The quantizer q^* is *globally optimal* if $\Delta(q^*) \leq \Delta(q)$ for any k-point quantizer q. It can be shown that q^* exists if $\mathbf{X}$ has finite second moments, so the answer to Question 1 in Section 1.1.3 is yes. Interestingly, however, the answers to Questions 2 and 3 are no in general. Finding a globally optimal vector quantizer for a given source distribution or density is a very hard problem. Presently, for $k > 2$ codepoints there seem to be no concrete examples of optimal vector quantizers for even the most common model distributions such as Gaussian, Laplacian, or uniform (in a hypercube) in any dimensions $d > 1$.

Since global optimality is not a feasible requirement, algorithms, even in theory, are usually designed to find *locally optimal* vector quantizers. A quantizer q is said to be locally optimal if $\Delta(q)$ is only a local minimum, that is, slight disturbance of any of the codepoints will cause an increase in the distortion. Necessary conditions for local optimality will be given in Section 2.1.3. We also describe here a theoretical algorithm, the Generalized Lloyd (GL) algorithm [LBG80], to find a locally optimal vector quantizer of a random variable.

In practice, the distribution of $\mathbf{X}$ is usually unknown. Therefore, the objective of *empirical quantizer design* is to find a vector quantizer based on $\mathcal{X}_n = \{\mathbf{X}_1, \ldots, \mathbf{X}_n\}$, a set of independent and identical copies of $\mathbf{X}$. To design a quantizer with low distortion, most existing practical algorithms attempt to implement the empirical loss minimization principle introduced for the general unsupervised learning model in Section 1.1.3. The performance of a vector quantizer q on $\mathcal{X}_n$ is measured by the *empirical distortion* of q given by

$$\Delta_n(q) = \frac{1}{n} \sum_{i=1}^{n} \Delta(\mathbf{X}_i, q(\mathbf{X}_i)). \tag{7}$$

The quantizer q_n^* is *globally optimal* on the data set $\mathcal{X}_n$ if $\Delta_n(q_n^*) \leq \Delta_n(q)$ for any k-point quantizer q. Finding an empirically optimal vector quantizer is, in theory, possible since the number of different partitions of $\mathcal{X}_n$ is finite. However, the systematic inspection of all different partitions is computationally infeasible. Instead, most practical methods use an iterative approach similar in spirit to the GL algorithm.

It is of both theoretical and practical interest to analyze how the expected loss of the empirically best vector quantizer $\Delta(q_n^*)$ relates to the best achievable loss $\Delta_n(q^*)$, even though q^* is not known and q_n^* is practically infeasible to obtain. Consistency (Question 4 in Section 1.1.3) of the estimation scheme means that the expected loss of the q_n^* converges in probability to the best achievable loss as the number of the data points grows, therefore, if we have a perfect algorithm and unlimited access to data, we can get arbitrarily close to the best achievable loss. A good convergence rate (Question 5) is important to establish upper bounds for the probability of error for a given data size. We start the analysis of the empirical loss minimization principle used for vector quantization design by presenting results on consistency and rate of convergence in Section 2.1.2.

2.1.2 Consistency and Rate Of Convergence

Consistency of the empirical quantizer design under general conditions was proven by Pollard [Pol81, Pol82]. The first rate of convergence results were obtained by Linder et al. [LLZ94]. In particular, [LLZ94] showed that if the distribution of $\mathbf{X}$ is concentrated on a bounded region, there exists a constant c such that

$$\Delta(q_n^*) - \Delta(q^*) \leq c d^{3/2} \sqrt{\frac{k \log n}{n}}. \tag{8}$$

An extension of this result to distributions with unbounded support is given in [MZ97]. Bartlett et al. [BLL98] pointed out that the $\sqrt{\log n}$ factor can be eliminated from the upper bound in (8) by using an analysis based on sophisticated uniform large deviation inequalities of Alexander [Ale84] or Talagrand [Tal94]. More precisely, it can be proven that there exists a constant c' such that

$$\Delta(q_n^*) - \Delta(q^*) \leq c' d^{3/2} \sqrt{\frac{k \log(kd)}{n}}. \tag{9}$$

There are indications that the upper bound can be tightened to $O(1/n)$. First, in (4) we showed that if $k = 1$, the expected loss of the sample average converges to the smallest possible loss at a rate of $O(1/n)$. Another indication that an $O(1/n)$ rate might be achieved comes from a result of Pollard [Pol82]. He showed if $\mathbf{X}$ has a specially smooth and regular density, the difference between the codepoints of the empirically designed quantizers and the codepoints of the optimal quantizer obeys a multidimensional central limit theorem. As Chou [Cho94] pointed out, this implies that that within the class of distributions considered by [Pol82], the distortion redundancy decreases at a rate $O(1/n)$. Despite these suggestive facts, it was showed by [BLL98] that in general, the conjectured

$O(1/n)$ distortion redundancy rate does not hold. In particular, [BLL98] proved that for any k-point quantizer q_n which is designed by *any* method from n independent training samples, there exists a distribution on a bounded subset of $\mathbb{R}^d$ such that the expected loss of q_n is bounded away from the optimal distortion by a constant times $1/\sqrt{n}$. Together with (9), this result shows that the minimax (worst-case) distortion redundancy for empirical quantizer design is asymptotically on the order of $1/\sqrt{n}$. As a final note, [BLL98] conjectures that the minimax expected distortion redundancy is some constant times

$$d^a \sqrt{\frac{k^{1-b/d}}{n}}$$

for some values of $a \in [1, 3/2]$ and $b \in [2, 4]$.

2.1.3 Locally Optimal Vector Quantizer

Suppose that we are given a particular codebook $\mathcal{C}$ but the partition is not specified. An optimal partition $\mathcal{V}$ can be constructed by mapping each input vector $\mathbf{x}$ to the codepoint $\mathbf{v}_\ell \in \mathcal{C}$ that minimizes the distortion $\Delta(\mathbf{x}, \mathbf{v}_\ell)$ among all codepoints, that is, by choosing the nearest codepoint to $\mathbf{x}$. Formally, $\mathcal{V} = \{V_1, \ldots, V_k\}$ is the optimal partition of the codebook $\mathcal{C}$ if

$$V_\ell = \{\mathbf{x} : \Delta(\mathbf{x}, \mathbf{v}_\ell) \leq \Delta(\mathbf{x}, \mathbf{v}_m),\ m = 1, \ldots, k\}. \tag{10}$$

(A tie-breaking rule such as choosing the codepoint with the lowest index is required if more than one codepoint minimizes the distortion.) V_ℓ is called the *Voronoi region* or *Voronoi set* associated with the codepoint $\mathbf{v}_\ell$.

Conversely, assume that we are given a partition $\mathcal{V} = \{V_1, \ldots, V_k\}$ and an optimal codebook $\mathcal{C} = \{\mathbf{v}_1, \ldots, \mathbf{v}_k\}$ is needed to be constructed. To minimize the expected distortion, we have to set

$$\mathbf{v}_\ell = \arg\min_{\mathbf{v}} E[\Delta(\mathbf{X}, \mathbf{v}) | \mathbf{X} \in V_\ell]. \tag{11}$$

$\mathbf{v}_\ell$ is called the *centroid* or the *center of gravity* of the set V_ℓ, motivated by the fact that for the squared error distortion (5) we have $\mathbf{v}_\ell = E[\mathbf{X} | \mathbf{X} \in V_\ell]$.

It can be shown that the *nearest neighbor condition* (10) and the *centroid condition* (11) must hold for any locally optimal vector quantizer. Another necessary condition of local optimality is that boundary points occur with zero probability, that is,

$$P\{\mathbf{X} : \mathbf{X} \in V_\ell, \Delta(\mathbf{X}, \mathbf{v}_\ell) = \Delta(\mathbf{X}, \mathbf{v}_m), \ell \neq m\} = 0. \tag{12}$$

If we have a codebook that satisfies all three necessary conditions of optimality, it is widely believed that it is indeed locally optimal. No general theoretical derivation of this result has ever been obtained. For the particular case of discrete distribution, however, it can be shown that under mild restrictions, a vector quantizer satisfying the three necessary conditions is indeed locally optimal [GKL80].

2.1.4 Generalized Lloyd Algorithm

The nearest neighbor condition and the centroid condition suggest a natural algorithm for designing a vector quantizer. The GL algorithm alternates between an expectation and a partition step until the relative improvement of the expected distortion is less than a preset threshold. In the expectation step the codepoints are computed according to (11), and in the partition step the Voronoi regions are set by using (10). It is assumed that an initial codebook $\mathcal{C}^{(0)}$ is given. When the probability density of **X** is known, the GL algorithm for constructing a vector quantizer is the following.

Algorithm 1 (The GL algorithm for distributions)

Step 0 *Set* $j = 0$, *and set* $\mathcal{C}^{(0)} = \left\{\mathbf{v}_1^{(0)}, \ldots, \mathbf{v}_k^{(0)}\right\}$ *to an initial codebook.*

Step 1 (Partition) *Construct* $\mathcal{V}^{(j)} = \left\{V_1^{(j)}, \ldots, V_k^{(j)}\right\}$ *by setting*
$V_\ell^{(j)} = \left\{\mathbf{x} : \Delta\left(\mathbf{x}, \mathbf{v}_\ell^{(j)}\right) \le \Delta\left(\mathbf{x}, \mathbf{v}_m^{(j)}\right),\ m = 1, \ldots, k\right\}$ *for* $\ell = 1, \ldots, k$.

Step 2 (Expectation) *Construct* $\mathcal{C}^{(j+1)} = \left\{\mathbf{v}_1^{(j+1)}, \ldots, \mathbf{v}_k^{(j+1)}\right\}$ *by setting*
$\mathbf{v}_\ell^{(j+1)} = \arg\min_{\mathbf{v}} E\left[\Delta(\mathbf{X}, \mathbf{v}) \middle| \mathbf{X} \in V_\ell^{(j)}\right] = E\left[\mathbf{X} \middle| \mathbf{X} \in V_\ell^{(j)}\right]$ *for* $\ell = 1, \ldots, k$.

Step 3 *Stop if* $\left(1 - \frac{\Delta\left(q^{(j+1)}\right)}{\Delta\left(q^{(j)}\right)}\right)$ *is less than or equal to a certain threshold. Otherwise, let* $j = j + 1$ *and go to Step 1.*

Step 1 is complemented with a suitable rule to break ties. When a cell becomes empty in Step 1, one can split the cell with the highest probability, or the cell with the highest partial distortion into two, and delete the empty cell.

It is easy to see that $\Delta\left(q^{(j)}\right)$ is non-increasing and non-negative, so it must have a limit $\Delta\left(q^{(\infty)}\right)$. [LBG80] showed that if a limiting quantizer $\mathcal{C}^{(\infty)}$ exists in the sense that $\mathcal{C}^{(j)} \to \mathcal{C}^{(\infty)}$ as $j \to \infty$ (in the usual Euclidean sense), then the codepoints of $\mathcal{C}^{(\infty)}$ are the centroids of the Voronoi regions induced by $\mathcal{C}^{(\infty)}$, so $\mathcal{C}^{(\infty)}$ is a fixed point of the algorithm with zero threshold.

The GL algorithm can easily be adjusted to the case when the distribution of **X** is unknown but a set of independent observations $\mathcal{X}_n = \{\mathbf{x}_1, \ldots, \mathbf{x}_n\} \subset \mathbb{R}^d$ of the underlying distribution is known instead. The modifications are straightforward replacements of the expectations by sample averages. In Step 3, the empirical distortion

$$\Delta_n(q) = \frac{1}{n} \sum_{i=1}^{n} \Delta(\mathbf{x}_i, q(\mathbf{x}_i)) = \frac{1}{n} \sum_{\ell=1}^{k} \sum_{\mathbf{x} \in V_\ell} \|\mathbf{v}_\ell - \mathbf{x}\|^2$$

is evaluated in place of the unknown expected distortion $\Delta_n(q)$. The GL algorithm for constructing a vector quantizer based on the data set $\mathcal{X}_n$ is the following.

Algorithm 2 (The GL algorithm for data sets)

Step 0 *Set* $j = 0$*, and set* $\mathcal{C}^{(0)} = \left\{\mathbf{v}_1^{(0)}, \ldots, \mathbf{v}_k^{(0)}\right\}$ *to an initial codebook.*

Step 1 (Partition) *Construct* $\mathcal{V}^{(j)} = \left\{V_1^{(j)}, \ldots, V_k^{(j)}\right\}$ *by setting*

$$V_\ell^{(j)} = \left\{\mathbf{x} : \Delta\left(\mathbf{x}, \mathbf{v}_\ell^{(j)}\right) \leq \Delta\left(\mathbf{x}, \mathbf{v}_m^{(j)}\right),\ m = 1, \ldots, k\right\} \text{ for } \ell = 1, \ldots, k.$$

Step 2 (Expectation) *Construct* $\mathcal{C}^{(j+1)} = \left\{\mathbf{v}_1^{(j+1)}, \ldots, \mathbf{v}_k^{(j+1)}\right\}$ *by setting*

$$\mathbf{v}_\ell^{(j+1)} = \arg\min_{\mathbf{v}} \sum_{\mathbf{x} \in V_\ell^{(j)} \cap \mathcal{X}_n} \Delta(\mathbf{x}, \mathbf{v}) = \frac{1}{\left|V_\ell^{(j)}\right|} \sum_{\mathbf{x} \in V_\ell^{(j)} \cap \mathcal{X}_n} \mathbf{x} \text{ for } \ell = 1, \ldots, k.$$

Step 3 *Stop if* $\left(1 - \frac{\Delta_n\left(q^{(j+1)}\right)}{\Delta_n\left(q^{(j)}\right)}\right)$ *is less than a certain threshold. Otherwise, let* $j = j + 1$ *and go to Step 1.*

For a finite training set, the GL algorithm always converges in a finite number of iterations since the average distortion is non-increasing in both Step 1 and Step 2 and there is only a finite number of ways to partition the training set into k subsets.

2.2 Principal Component Analysis

Principal component analysis (PCA), which is also known as the *Karhunen-Loève transformation*, is perhaps the oldest and best-known technique in multivariate analysis. It was first introduced by Pearson [Pea01], who used it in a biological context. It was then developed by Hotelling [Hot33] in work done on psychometry. It appeared once again quite independently in the setting of probability theory, as considered by Karhunen [Kar47], and was subsequently generalized by Loève. For a full treatment of principal component analysis, see, e.g., [JW92].

Principal component analysis can be considered one of the simplest forms of unsupervised learning when the manifold to fit is a linear subspace. Principal components are also used for initialization in more sophisticated unsupervised learning methods.

The analysis is motivated by the following two problems.

1. Given a random vector $\mathbf{X} \in \mathbb{R}^d$, find the d'-dimensional linear subspace that captures most of the variance of $\mathbf{X}$. This is the problem of *feature extraction* where the objective is to reduce the dimension of the data while retaining most of its information content.

2. Given a random vector $\mathbf{X} \in \mathbb{R}^d$, find the d'-dimensional linear subspace that minimizes the expected distance of $\mathbf{X}$ from the subspace. This problem arises in the area of *data compression* where the task is to represent the data with only a few parameters while keeping low the distortion generated by the projection.

It turns out that the two problems have the same solution, and the solution lies in the eigenstructure of the covariance matrix of $\mathbf{X}$. Before we derive this result in Section 2.2.2, we introduce the definition and show some properties of curves in the d-dimensional Euclidean space in Section 2.2.1. Concepts defined here will be used throughout the thesis. After the analysis, in Section 2.2.3, we summarize some of the properties of the first principal component line. In subsequent definitions of principal curves, these properties will serve as bases for generalization. Finally, in Section 2.2.4 we describe a fast algorithm to find principal components of data sets. The significance of this algorithm is that it is similar in spirit to both the GL algorithm of vector quantization and the HS algorithm (Section 3.1.1) for computing principal curves of data sets.

2.2.1 One-Dimensional Curves

In this section we define curves, lines, and line segments in the d-dimensional Euclidean space. We also introduce the notion of the distance function, the expected Euclidean squared distance of a random vector and a curve. The distance function will be used throughout this thesis as a measure of the distortion when a random vector is represented by its projection to a curve. This section also contains some basic facts on curves that are needed later for the definition and analysis of principal curves (see, e.g., [O'N66] for further reference).

Definition 1 *A* curve *in d-dimensional Euclidean space is a continuous function $\mathbf{f} : I \to \mathbb{R}^d$, where $I = [a,b]$ is a closed interval of the real line.*

The curve $\mathbf{f}$ can be considered as a vector of d functions of a single variable t, $\mathbf{f}(t) = (f_1(t), \ldots, f_d(t))$, where $f_1(t), \ldots, f_d(t)$ are called the *coordinate functions*.

The Length of a Curve

The *length* of a curve $\mathbf{f}$ over an interval $[\alpha,\beta] \subset [a,b]$, denoted by $l(\mathbf{f},\alpha,\beta)$, is defined by

$$l(\mathbf{f},\alpha,\beta) = \sup \sum_{i=1}^{N} \|\mathbf{f}(t_i) - \mathbf{f}(t_{i-1})\|, \tag{13}$$

where the supremum is taken over all finite partitions of $[\alpha,\beta]$ with arbitrary subdivision points $\alpha = t_0 \le t_1 < \cdots \le t_N = \beta$, $N \ge 1$. The length of $\mathbf{f}$ over its entire domain $[a,b]$ is denoted by $l(\mathbf{f})$.

Distance Between a Point and a Curve

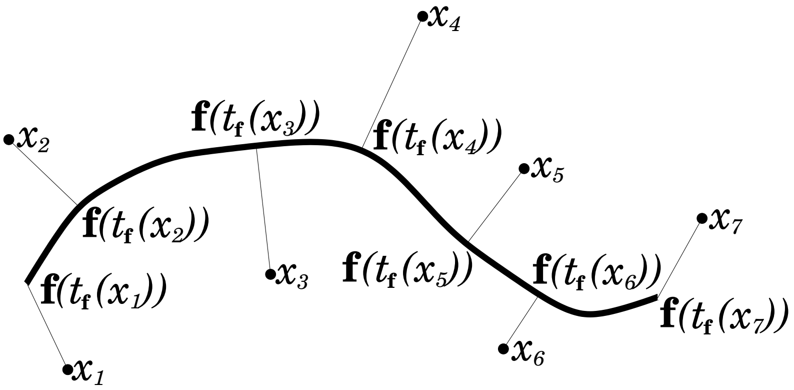
Let $\mathbf{f}(t) = (f_1(t), \ldots, f_d(t))$ be a curve in $\mathbb{R}^d$ parameterized by $t \in \mathbb{R}$, and for any $\mathbf{x} \in \mathbb{R}^d$ let $t_{\mathbf{f}}(\mathbf{x})$ denote the parameter value t for which the distance between $\mathbf{x}$ and $\mathbf{f}(t)$ is minimized (see Figure 2).

More formally, the *projection index* $t_{\mathbf{f}}(\mathbf{x})$ is defined by

$$t_{\mathbf{f}}(\mathbf{x}) = \sup\big\{t : \|\mathbf{x}-\mathbf{f}(t)\| = \inf_{\tau}\|\mathbf{x}-\mathbf{f}(\tau)\|\big\}, \tag{14}$$

where $\|\cdot\|$ denotes the Euclidean norm in $\mathbb{R}^d$. Accordingly, the *projection point* of $\mathbf{x}$ to $\mathbf{f}$ is $\mathbf{f}(t_{\mathbf{f}}(\mathbf{x}))$. The squared Euclidean distance of $\mathbf{f}$ and $\mathbf{x}$ is the squared distance of $\mathbf{x}$ from its projection point to $\mathbf{f}$, that is,

$$\Delta(\mathbf{x},\mathbf{f}) = \inf_{a\le t\le b}\|\mathbf{x}-\mathbf{f}(t)\|^2 = \|\mathbf{x}-\mathbf{f}(t_{\mathbf{f}}(\mathbf{x}))\|^2. \tag{15}$$



x_4
$\mathbf{f}(t_{\mathbf{f}}(x_3))$
$\mathbf{f}(t_{\mathbf{f}}(x_4))$
x_2
x_5
$\mathbf{f}(t_{\mathbf{f}}(x_2))$
x_7
x_3
$\mathbf{f}(t_{\mathbf{f}}(x_5))$
$\mathbf{f}(t_{\mathbf{f}}(x_6))$
$\mathbf{f}(t_{\mathbf{f}}(x_1))$
$\mathbf{f}(t_{\mathbf{f}}(x_7))$
x_6
x_1

Figure 2: Projecting points to a curve.

Arc Length Parameterization and the Lipschitz Condition

Two curves $\mathbf{f} : [a,b] \to \mathbb{R}^d$ and $\mathbf{g} : [a',b'] \to \mathbb{R}^d$ are said to be *equivalent* if there exist two nondecreasing continuous functions $\phi : [0,1] \to [a,b]$ and $\eta : [0,1] \to [a',b']$ such that

$$\mathbf{f}(\phi(t)) = \mathbf{g}(\eta(t)), \quad 0 \le t \le 1.$$

In this case we write $\mathbf{f} \sim \mathbf{g}$, and it is easy to see that $\sim$ is an equivalence relation. If $\mathbf{f} \sim \mathbf{g}$, then $l(\mathbf{f}) = l(\mathbf{g})$. A curve $\mathbf{f}$ over $[a,b]$ is said to be *parameterized by its arc length* if $l(\mathbf{f},a,t) = t-a$ for any $a \le t \le b$. Let $\mathbf{f}$ be a curve over $[a,b]$ with length L. It is not hard to see that there exists a unique arc length parameterized curve $\mathbf{g}$ over $[0,L]$ such that $\mathbf{f} \sim \mathbf{g}$.

Let $\mathbf{f}'$ be any curve with length $L' \le L$, and consider the arc length parameterized curve $\mathbf{g} \sim \mathbf{f}'$ with parameter interval $[0,L']$. By definition (13), for all $s_1,s_2 \in [0,L']$ we have $\|\mathbf{g}(s_1)-\mathbf{g}(s_2)\| \le |s_1-s_2|$. Define $\hat{\mathbf{g}}(t) = \mathbf{g}(L't)$ for $0 \le t \le 1$. Then $\mathbf{f}' \sim \hat{\mathbf{g}}$, and $\hat{\mathbf{g}}$ satisfies the Lipschitz condition, i.e., For all $t_1,t_2 \in [0,1]$,

$$\|\hat{\mathbf{g}}(t_1)-\hat{\mathbf{g}}(t_2)\| = \|\mathbf{g}(L't_1)-\mathbf{g}(L't_2)\| \le L'|t_1-t_2| \le L|t_1-t_2|. \tag{16}$$

On the other hand, note that if $\hat{\mathbf{g}}$ is a curve over $[0,1]$ which satisfies the Lipschitz condition (16), then its length is at most L.

Note that if $l(\mathbf{f}) < \infty$, then by the continuity of $\mathbf{f}$, its graph

$$G_{\mathbf{f}} = \mathbf{f}([a,b]) = \{\mathbf{f}(t) : a \le t \le b\} \tag{17}$$

is a compact subset of $\mathbb{R}^d$, and the infimum in (15) is achieved for some t. Also, since $G_{\mathbf{f}} = G_{\mathbf{g}}$ if $\mathbf{f} \sim \mathbf{g}$, we also have that $\Delta(\mathbf{x},\mathbf{f}) = \Delta(\mathbf{x},\mathbf{g})$ for all $\mathbf{g} \sim \mathbf{f}$.

Geometrical Properties of Curves

Let $\mathbf{f} : [a,b] \to \mathbb{R}^d$ be a differentiable curve with $\mathbf{f} = (f_1,\ldots,f_d)$. The *velocity* of the curve is defined as the vector function

$$\mathbf{f}'(t) = \left(\frac{df_1}{dt}(t),\ldots,\frac{df_d}{dt}(t)\right).$$

It is easy to see that $\mathbf{f}'(t)$ is tangent to the curve at t and that for an arc length parameterized curve $\|\mathbf{f}'(t)\| \equiv 1$. Note that for a differentiable curve $\mathbf{f} : [a,b] \to \mathbb{R}^d$, the length of the curve (13) over an interval $[\alpha,\beta] \subset [a,b]$ can be defined as

$$l(\mathbf{f},\alpha,\beta) = \int_\alpha^\beta \|\mathbf{f}'(t)\|dt.$$

The vector function

$$\mathbf{f}''(t) = \left(\frac{d^2 f_1}{dt^2}(t),\ldots,\frac{d^2 f_d}{dt^2}(t)\right)$$

is called the *acceleration* of the curve at t. For an arc length parameterized curve $\mathbf{f}''(t)$ is orthogonal to the tangent vector. In this case $\mathbf{f}''(t)/\|\mathbf{f}''(t)\|$ is called the *principal normal* to the curve at t. The vectors $\mathbf{f}'(t)$ and $\mathbf{f}''(t)$ span a plane. There is a unique arc length parameterized circle in this plane that goes through $\mathbf{f}(t)$ and has the same velocity and acceleration at t as the curve itself. The radius $r_{\mathbf{f}}(t) = 1/\|\mathbf{f}''(t)\|$ is called the *radius of curvature* of the curve $\mathbf{f}$ at t. The center $\mathbf{c}_{\mathbf{f}}(t)$ of the circle is called the *center of curvature* of $\mathbf{f}$ at t (Figure 3).

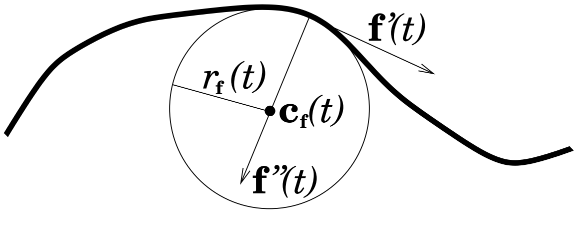


Figure 3: The velocity $\mathbf{f}'(t)$, the acceleration $\mathbf{f}''(t)$, the radius of curvature $r_{\mathbf{f}}(t)$, and the center of curvature $\mathbf{c}_{\mathbf{f}}(t)$ of an arc length parameterized curve.

The Distance Function and the Empirical Distance Function

Consider a d-dimensional random vector $\mathbf{X} = (X_1, \ldots, X_d)$ with finite second moments. The *distance function* of a curve $\mathbf{f}$ is defined as the expected squared distance between $\mathbf{X}$ and $\mathbf{f}$, that is,

$$\Delta(\mathbf{f}) = E\left[\Delta(\mathbf{X},\mathbf{f})\right] = E\left[\inf_t \|\mathbf{X} - \mathbf{f}(t)\|^2\right] = E\left[\|\mathbf{X} - \mathbf{f}(t_{\mathbf{f}}(\mathbf{X}))\|^2\right]. \tag{18}$$

In practical situations the distribution of $\mathbf{X}$ is usually unknown, but a data set $\mathcal{X}_n = \{\mathbf{x}_1, \ldots, \mathbf{x}_n\} \subset \mathbb{R}^d$ drawn independently from the distribution is known instead. In this case, we can estimate the distance function of a curve $\mathbf{f}$ by the *empirical distance function* defined as

$$\Delta_n(\mathbf{f}) = \frac{1}{n}\sum_{i=1}^{n} \Delta(\mathbf{x}_i, \mathbf{f}). \tag{19}$$

Straight Lines and Line Segments

Consider curves of the form

$$\mathbf{s}(t) = t\mathbf{u} + \mathbf{c}$$

where $\mathbf{u}, \mathbf{c} \in \mathbb{R}^d$, and $\mathbf{u}$ is a unit-vector. If the domain of t is the real line, $\mathbf{s}$ is called a *straight line*, or *line*. If $\mathbf{s}$ is defined on a finite interval $[a,b] \subset \mathbb{R}$, $\mathbf{s}$ is called a *straight line segment*, or *line segment*. Note that since $\|\mathbf{u}\| = 1$, $\mathbf{s}$ is arc length parameterized.

By (15), the squared distance of a point $\mathbf{x}$ and a line $\mathbf{s}$ is

$$\begin{aligned}
\Delta(\mathbf{x},\mathbf{s}) &= \inf_{t\in\mathbb{R}} \|\mathbf{x} - \mathbf{s}(t)\|^2 \\
&= \inf_{t\in\mathbb{R}} \|\mathbf{x} - (t\mathbf{u} + \mathbf{c})\|^2 \\
&= \|\mathbf{x} - \mathbf{c}\|^2 + \inf_{t\in\mathbb{R}} \left(t^2 - 2t(\mathbf{x} - \mathbf{c})^T\mathbf{u}\right) \\
&= \|\mathbf{x} - \mathbf{c}\|^2 - ((\mathbf{x} - \mathbf{c})^T\mathbf{u})^2
\end{aligned} \tag{20}$$

where $\mathbf{x}^T$ denotes the transpose of $\mathbf{x}$. The projection point of $\mathbf{x}$ to $\mathbf{s}$ is $\mathbf{c} + ((\mathbf{x} - \mathbf{c})^T\mathbf{u})\mathbf{u}$.

If $\mathbf{s}(t) = t\mathbf{u} + \mathbf{c}$ is a line segment defined over $[a,b] \subset \mathbb{R}$, the way the distance of a point $\mathbf{x}$ and the line segment is measured depends on the value of the projection index $t_{\mathbf{s}}(\mathbf{x})$. If $t_{\mathbf{s}}(\mathbf{x}) = a$ or $t_{\mathbf{s}}(\mathbf{x}) = b$, the distance is measured as the distance of $\mathbf{x}$ and one of the endpoints $\mathbf{v}_1 = a\mathbf{u} + \mathbf{c}$ or $\mathbf{v}_2 = b\mathbf{u} + \mathbf{c}$, respectively. If $\mathbf{x}$ projects to $\mathbf{s}$ between the endpoints, the distance is measured as if $\mathbf{s}$ were a line (Figure 4). Formally,

$$\Delta(\mathbf{x},\mathbf{s}) = \begin{cases} \|\mathbf{x} - \mathbf{v}_1\|^2 & \text{if } \mathbf{s}(t_{\mathbf{s}}(\mathbf{x})) = \mathbf{v}_1, \\ \|\mathbf{x} - \mathbf{v}_2\|^2 & \text{if } \mathbf{s}(t_{\mathbf{s}}(\mathbf{x})) = \mathbf{v}_2, \\ \|\mathbf{x} - \mathbf{c}\|^2 - ((\mathbf{x} - \mathbf{c})^T\mathbf{u})^2 & \text{otherwise.} \end{cases} \tag{21}$$

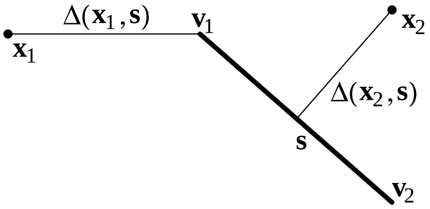


Figure 4: Distance of a point and a line segment. If a point $\mathbf{x}_1$ projects to one of the endpoints $\mathbf{v}_1$ of the line segment $\mathbf{s}$, the distance of $\mathbf{x}_1$ and $\mathbf{s}$ is identical to the distance of $\mathbf{x}_1$ and $\mathbf{v}_1$. If a point $\mathbf{x}_2$ projects to $\mathbf{s}$ between the endpoints, the distance is measured as if $\mathbf{s}$ were a line.

2.2.2 Principal Component Analysis

Consider a d-dimensional random vector $\mathbf{X} = (X_1, \ldots, X_d)$ with finite second moments and zero mean[1]. Let $\mathbf{u} \in \mathbb{R}^d$ be an arbitrary unit vector, and $\mathbf{s}(t) = t\mathbf{u}$ the corresponding straight line. Let $Y = t_{\mathbf{s}}(\mathbf{X}) = \mathbf{X}^T\mathbf{u}$ be the projection index of $\mathbf{X}$ to $\mathbf{s}$. From $E[\mathbf{X}] = 0$ it follows that $E[Y] = 0$, and so the variance of Y can be written as

$$\begin{aligned}\sigma_Y^2 &= E[(\mathbf{X}^T\mathbf{u})^2] = E[(\mathbf{u}^T\mathbf{X})(\mathbf{X}^T\mathbf{u})] \\ &= \mathbf{u}^T E[\mathbf{X}\mathbf{X}^T]\mathbf{u} = \mathbf{u}^T\mathbf{R}\mathbf{u} \\ &= \psi(\mathbf{u}) \qquad (22)\end{aligned}$$

where the $d \times d$ matrix $\mathbf{R} = E\left[(\mathbf{X} - E[\mathbf{X}])(\mathbf{X} - E[\mathbf{X}])^T\right] = E\left[\mathbf{X}\mathbf{X}^T\right]$ is the *covariance matrix* of $\mathbf{X}$. Since $\mathbf{R}$ is symmetric, $\mathbf{R} = \mathbf{R}^T$, and so for any $\mathbf{v}, \mathbf{w} \in \mathbb{R}^d$

$$\mathbf{v}^T\mathbf{R}\mathbf{w} = \mathbf{w}^T\mathbf{R}\mathbf{v}. \qquad (23)$$

To find stationary values of the projection variance $\psi(\mathbf{u})$, consider a small perturbation $\delta\mathbf{u}$, such that $\|\mathbf{u} + \delta\mathbf{u}\| = 1$. From (22) and (23) it follows that

$$\begin{aligned}\psi(\mathbf{u} + \delta\mathbf{u}) &= (\mathbf{u} + \delta\mathbf{u})^T\mathbf{R}(\mathbf{u} + \delta\mathbf{u}) \\ &= \mathbf{u}^T\mathbf{R}\mathbf{u} + 2(\delta\mathbf{u})^T\mathbf{R}\mathbf{u} + (\delta\mathbf{u})^T\mathbf{R}\,\delta\mathbf{u}.\end{aligned}$$

Ignoring the second order term $(\delta\mathbf{u})^T\mathbf{R}\delta\mathbf{u}$ and using the definition of $\psi(\mathbf{u})$ again, we have

$$\begin{aligned}\psi(\mathbf{u} + \delta\mathbf{u}) &= \mathbf{u}^T\mathbf{R}\mathbf{u} + 2(\delta\mathbf{u})^T\mathbf{R}\mathbf{u} \\ &= \psi(\mathbf{u}) + 2(\delta\mathbf{u})^T\mathbf{R}\mathbf{u}. \qquad (24)\end{aligned}$$

If $\mathbf{u}$ is such that $\psi(\mathbf{u})$ has a stationary value, to a first order in $\delta\mathbf{u}$, we have

$$\psi(\mathbf{u} + \delta\mathbf{u}) = \psi(\mathbf{u}). \qquad (25)$$

[1] If $E[\mathbf{X}] \neq 0$, then we subtract the mean from $\mathbf{X}$ before proceeding with the analysis.

Hence, (25) and (24) imply that

$$(\delta\mathbf{u})^T\mathbf{R}\mathbf{u} = 0. \tag{26}$$

Since $\|\mathbf{u}+\delta\mathbf{u}\|^2 = \|\mathbf{u}\|^2 + 2(\delta\mathbf{u})^T\mathbf{u} + \|\delta\mathbf{u}\|^2 = 1$, we require that, to a first order in $\delta\mathbf{u}$,

$$(\delta\mathbf{u})^T\mathbf{u} = 0. \tag{27}$$

This means that the perturbation $\delta\mathbf{u}$ must be orthogonal to $\mathbf{u}$. To find a solution of (26) with the constraint (27), we have to solve

$$(\delta\mathbf{u})^T\mathbf{R}\mathbf{u} - l(\delta\mathbf{u})^T\mathbf{u} = 0,$$

or, equivalently,

$$(\delta\mathbf{u})^T(\mathbf{R}\mathbf{u} - l\mathbf{u}) = 0. \tag{28}$$

For the condition (28) to hold, it is necessary and sufficient that we have

$$\mathbf{R}\mathbf{u} = l\mathbf{u}. \tag{29}$$

The solutions of (29), $l_1,\ldots,l_d$, are the *eigenvalues* of R, and the corresponding unit vectors, $\mathbf{u}_1,\ldots,\mathbf{u}_d$, are the *eigenvectors* of R. For the sake of simplicity, we assume that the eigenvalues are distinct, and they are indexed in decreasing order, i.e.,

$$l_1 > \ldots > l_d.$$

Define the $d \times d$ matrix $\mathbf{U}$ as

$$\mathbf{U} = [\mathbf{u}_1,\ldots,\mathbf{u}_d],$$

and let $\blacksquare$ be the diagonal matrix

$$\blacksquare = \mathrm{diag}[l_1,\ldots,l_d].$$

Then the d equations of form (29) can be summarized in

$$\mathbf{R}\mathbf{U} = \mathbf{U}\blacksquare. \tag{30}$$

The matrix $\mathbf{U}$ is orthonormal so $\mathbf{U}^{-1} = \mathbf{U}^T$, and therefore (30) can be written as

$$\mathbf{U}^T\mathbf{R}\mathbf{U} = \blacksquare. \tag{31}$$

Thus, from (22) and (31) it follows that the principal directions along which the projection variance is stationary are the eigenvectors of the covariance matrix $\mathbf{R}$, and the stationary values themselves are the eigenvalues of $\mathbf{R}$. (31) also implies that the maximum value of the projection variance is the largest eigenvalue of **R**, and the principal direction along which the projection variance is maximal is the eigenvector associated with the largest eigenvalue. Formally,

$$\max_{\|\mathbf{u}\|=1} \psi(\mathbf{u}) = l_1, \tag{32}$$

and

$$\arg\max_{\|\mathbf{u}\|=1} \psi(\mathbf{u}) = \mathbf{u}_1. \tag{33}$$

The straight lines $\mathbf{s}_i(t) = t\mathbf{u}_i, i = 1, \ldots, d$ are called the *principal component lines* of **X**. Since the eigenvectors form an orthonormal basis of $\mathbb{R}^d$, any data vector $\mathbf{x} \in \mathbb{R}^d$ can be represented uniquely by its projection indices $t_i = \mathbf{u}_i^T \mathbf{x}, i = 1, \ldots, d$ to the principal component lines. The projection indices $t_i, \ldots, t_d$ are called the *principal components* of **x**. The construction of the vector $\mathbf{t} = [t_i, \ldots, t_d]^T$ of the principal components,

$$\mathbf{t} = \mathbf{U}^T \mathbf{x},$$

is the principal component analysis of **x**. To reconstruct the original data vector **x** from **t**, note again that $\mathbf{U}^{-1} = \mathbf{U}^T$ so

$$\mathbf{x} = (\mathbf{U}^T)^{-1}\mathbf{t} = \mathbf{U}\mathbf{t} = \sum_{i=1}^{d} t_i \mathbf{u}_i. \tag{34}$$

From the perspective of feature extraction and data compression, the practical value of principal component analysis is that it provides an effective technique for dimensionality reduction. In particular, we may reduce the number of parameters needed for effective data representation by discarding those linear combinations in (34) that have small variances and retain only those terms that have large variances. Formally, let $\S_{d'}$ be the d'-dimensional linear subspace spanned by the first d' eigenvectors of **R**. To approximate **X**, we define

$$\mathbf{X}' = \sum_{i=1}^{d'} t_i \mathbf{u}_i,$$

the projection of **X** to $\S_{d'}$. It can be shown by using (33) and induction that $\S_{d'}$ maximizes the variance of $\mathbf{X}'$,

$$E\left[\mathbf{X}'^2\right] = \sum_{i=1}^{d'} \psi(\mathbf{u}_i) = \sum_{i=1}^{d'} l_i,$$

and minimizes the variance of $\mathbf{X} - \mathbf{X}'$,

$$E\left[(\mathbf{X} - \mathbf{X}')^2\right] = \sum_{i=d'+1}^{d} \psi(\mathbf{u}_i) = \sum_{i=d'+1}^{d} l_i,$$

among all d'-dimensional linear subspaces. In other words, the solutions of both Problem 1 and Problem 2 are the subspace which is spanned by the first d' eigenvectors of **X**'s covariance matrix.

2.2.3 Properties of the First Principal Component Line

The *first principal component line* (Figure 5) of a random variable $\mathbf{X}$ with zero mean is defined as the straight line $\mathbf{s}_1 = t\mathbf{u}_1$ where $\mathbf{u}_1$ is the eigenvector which belongs to the largest eigenvalue l_1 of $\mathbf{X}$'s correlation matrix. The first principal component line has the following properties.

1. The first principal component line maximizes the variance of the projection of $\mathbf{X}$ to a line among all straight lines.
2. The first principal component line minimizes the distance function among all straight lines.
3. If the distribution of $\mathbf{X}$ is elliptical, the first principal component line is *self-consistent*, that is, any point of the line is the conditional expectation of $\mathbf{X}$ over those points of the space which project to this point. Formally,

$$\mathbf{s}_1(t) = E\left[\mathbf{X} | t_{\mathbf{f}}(\mathbf{X}) = t\right].$$

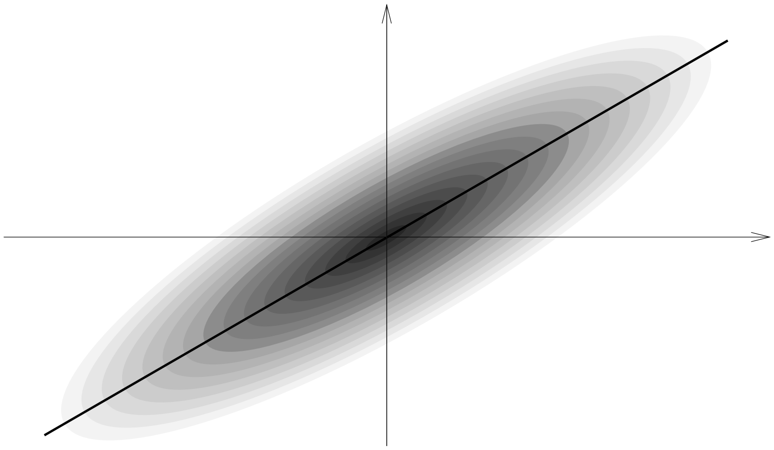


Figure 5: The first principal component line of an elliptical distribution in the plane.

Property 1 is a straightforward consequence of (33). To show Property 2, note that if $\mathbf{s}(t) = t\mathbf{u} + \mathbf{c}$ is an arbitrary straight line, then by (18) and (20),

$$\begin{aligned} \Delta(\mathbf{s}) &= E\left[\Delta(\mathbf{X},\mathbf{s})\right] \\ &= E\left[\|\mathbf{X}-\mathbf{c}\|^2 - ((\mathbf{X}-\mathbf{c})^T\mathbf{u})^2\right] \\ &= E\left[\|\mathbf{X}\|^2\right] + \|\mathbf{c}\|^2 - E\left[(\mathbf{X}^T\mathbf{u})^2\right] - (\mathbf{c}^T\mathbf{u})^2 \qquad (35) \\ &= \sigma_{\mathbf{X}}^2 - \psi(\mathbf{u}) + \|\mathbf{c}\|^2 - (\mathbf{c}^T\mathbf{u})^2 \\ &\leq \sigma_{\mathbf{X}}^2 - \psi(\mathbf{u}), \qquad (36) \end{aligned}$$

where (35) follows from $E[\mathbf{X}] = 0$. On the one hand, in (36) equality holds if and only if $\mathbf{c} = t\mathbf{u}$ for some $t \in \mathbb{R}$. Geometrically, it means that the minimizing line must go through the origin. On the other hand, $\sigma_{\mathbf{X}}^2 - \psi(\mathbf{u})$ is minimized when $\psi(\mathbf{u})$ is maximized, that is, when $\mathbf{u} = \mathbf{u}_1$. These two conditions together imply Property 2. Property 3 follows from the fact that the density of a random variable with an elliptical distribution is symmetrical about the principal component lines.

2.2.4 A Fast PCA Algorithm for Data Sets

In practice, principal component analysis is usually applied for sets of data points rather than distributions. Consider a data set $\mathcal{X}_n = \{\mathbf{x}_1, \ldots, \mathbf{x}_n\} \subset \mathbb{R}^d$, such that $\frac{1}{n}\sum_{i=1}^n \mathbf{x}_n = 0$. The first principal component line of $\mathcal{X}_n$ is a straight line $\mathbf{s}_1(t) = t\mathbf{u}_1$ that minimizes the empirical distance function (19),

$$\Delta_n(\mathbf{s}) = \frac{1}{n}\sum_{i=1}^n \Delta(\mathbf{x}_i, \mathbf{s}),$$

among all straight lines. The solution lies in the eigenstructure of the sample covariance matrix of the data set, which is defined as $\mathbf{R}_n = \frac{1}{n}\sum_{i=1}^n \mathbf{x}_n\mathbf{x}_n^T$. Following the derivation of PCA for distributions previously in this section, it can be shown easily that the unit vector $\mathbf{u}_1$ that defines the minimizing line $\mathbf{s}_1$ is the eigenvector which belongs to the largest eigenvalue of $\mathbf{R}_n$.

An obvious algorithm to minimize $\Delta_n(\mathbf{s})$ is therefore to find the eigenvectors and eigenvalues of $\mathbf{R}_n$. The crude method, direct diagonalization of $\mathbf{R}_n$, can be extremely costly for high-dimensional data since it takes $O(nd^3)$ operations. More sophisticated techniques, for example the power method (e.g., see [Wil65]), exist that perform matrix diagonalization in $O(nd^2)$ steps if only the first leading eigenvectors and eigenvalues are required. Since the $d \times d$ covariance matrix $\mathbf{R}_n$ must explicitly be computed, $O(nd^2)$ is also the theoretical lower limit of the computational complexity of this approach.

To break the $O(nd^2)$ barrier, several approximative methods were proposed (e.g., [Oja92], [RT89], [Föl89]). The common approach of these methods is to start from an arbitrary line, and to iteratively optimize the orientation of the line using the data so that it converges to the first principal component line. The characterizing features of these algorithms are the different learning rules they use for the optimization in each iteration.

The algorithm we introduce here is of the same genre. It was proposed recently, independently by Roweis [Row98] and Tipping and Bishop [TB99]. The reason we present it here is that there is a strong analogy between this algorithm designed for finding the first principal component line[2], and the HS algorithm (Section 3.1.1) for computing principal curves of data sets. Moreover, we also use a similar method in the inner iteration of the polygonal line algorithm (Section 5.1) to optimize the locations of vertices of the polygonal principal curve.

[2] The original algorithm in [Row98] and [TB99] can compute the first d principal components simultaneously. For the sake of simplicity, we present it here only for the first principal component line.

The basic idea of the algorithm is the following. Start with an arbitrary straight line, and project all the data points to the line. Then *fix the projection indices*, and find a new line that optimizes the distance function. Once the new line has been computed, restart the iteration, and continue until convergence.

Formally, let $\mathbf{s}^{(j)}(t) = t\mathbf{u}^{(j)}$ be the line produced by the jth iteration, and let $\mathbf{t}^{(j)} = \left[t_1^{(j)}, \ldots, t_n^{(j)}\right]^T = \left[\mathbf{x}_1^T\mathbf{u}^{(j)}, \ldots, \mathbf{x}_n^T\mathbf{u}^{(j)}\right]^T$ be the vector of projection indices of the data points to $\mathbf{s}^{(j)}$. The distance function of $\mathbf{s}(t) = t\mathbf{u}$ assuming the fixed projection vector $\mathbf{t}^{(j)}$ is defined as

$$\begin{aligned}\Delta_n\left(\mathbf{s}\middle|\mathbf{t}^{(j)}\right) = &= \sum_{i=1}^{n}\left\|\mathbf{x}_i - t_i^{(j)}\mathbf{u}\right\|^2 \\ &= \sum_{i=1}^{n}\|\mathbf{x}_i\|^2 + \|\mathbf{u}\|^2\sum_{i=1}^{n}\left(t_i^{(j)}\right)^2 - 2\mathbf{u}^T\sum_{i=1}^{n}t_i^{(j)}\mathbf{x}_i. \end{aligned} \tag{37}$$

Therefore, to find the optimal line $\mathbf{s}^{(j+1)}$, we have to minimize (37) with the constraint that $\|\mathbf{u}\| = 1$. It can be shown easily that the result of the constrained minimization is

$$\mathbf{u}^{(j+1)} = \underset{\|\mathbf{u}\|=1}{\arg\min}\,\Delta\left(\mathbf{s}\middle|\mathbf{t}^{(j)}\right) = \frac{\sum_{i=1}^{n}t_i^{(j)}\mathbf{x}_i}{\left\|\sum_{i=1}^{n}t_i^{(j)}\mathbf{x}_i\right\|},$$

and so $\mathbf{s}^{(j+1)}(t) = t\mathbf{u}^{(j+1)}$.

The formal algorithm is the following.

Algorithm 3 (The RTB algorithm)

Step 0 *Let* $\mathbf{s}^{(0)}(t) = t\mathbf{u}^{(0)}$ *be an arbitrary line. Set* $j = 0$.

Step 1 *Set* $\mathbf{t}^{(j)} = \left[t_1^{(j)}, \ldots, t_n^{(j)}\right]^T = \left[\mathbf{x}_1^T\mathbf{u}^{(j)}, \ldots, \mathbf{x}_n^T\mathbf{u}^{(j)}\right]^T$.

Step 2 *Define* $\mathbf{u}^{(j+1)} = \frac{\sum_{i=1}^{n}t_i^{(j)}\mathbf{x}_i}{\left\|\sum_{i=1}^{n}t_i^{(j)}\mathbf{x}_i\right\|}$, *and* $\mathbf{s}^{(j+1)}(t) = t\mathbf{u}^{(j+1)}$.

Step 3 *Stop if* $\left(1 - \frac{\Delta_n\left(\mathbf{s}^{(j+1)}\right)}{\Delta_n\left(\mathbf{s}^{(j)}\right)}\right)$ *is less than a certain threshold. Otherwise, let* $j = j+1$ *and go to Step 1.*

The standard convergence proof for the Expectation-Minimization (EM) algorithm [DLR77] applies to the RTB algorithm so it can be shown that $\mathbf{s}^{(j)}$ has a limit $\mathbf{s}^{(\infty)}$, and that the distance function $\Delta_n(\mathbf{s})$ has a local maximum in $\mathbf{s}^{(\infty)}$. Furthermore, [TB99] showed that the only stable local extremum is the global maximum so $\mathbf{s}^{(\infty)}$ is indeed the first principal component line.