Appendix A
Basic Facts from Optimization

Since the fabric of the universe is most perfect and the work of a most wise Creator, nothing at all takes place in the universe in which some rule of maximum or minimum does not appear.

—L. Euler

In engineering practice, there are often many possible or feasible solutions to a given problem. For instance, there might be multiple models that can explain the same observed data. In such situations, it is desirable to find a solution that is better than others in the sense that it optimizes certain objective function, e.g., it maximizes a likelihood function. To make this book more self-contained, we review in this appendix some of the key facts and tools from optimization. This appendix is by no means meant to be a complete tutorial in optimization. The reader is referred to (Bertsekas 1999; Boyd and Vandenberghe 2004) for details.

A.1 Unconstrained Optimization

The goal of *unconstrained optimization* is to find the minimum value of a function $f : \mathbb{R}^n \to \mathbb{R}$ as well as the point $\boldsymbol{x}^* \in \arg\min_{\boldsymbol{x}} f(\boldsymbol{x})$ at which the function achieves its minimum value, i.e., a point $\boldsymbol{x}^*$ such that $f(\boldsymbol{x}^*) \leq f(\boldsymbol{x}), \forall \boldsymbol{x} \in \mathbb{R}^n$. Notice that in general, the minimum value or optimal solution of a function may not exist, and even if it does exist, it may not be unique. For simplicity and convenience, unless otherwise stated, we will always assume that the function f is twice differentiable. We denote the gradient and Hessian of the function f by ∇f and $\nabla^2 f$, respectively. Notice that $\nabla f(\boldsymbol{x})$ is an n-dimensional vector and $\nabla^2 f(\boldsymbol{x})$ is an $n \times n$ matrix. More precisely, they are defined to be

R. Vidal et al., *Generalized Principal Component Analysis*, Interdisciplinary Applied Mathematics 40, DOI 10.1007/978-0-387-87811-9

$$\nabla f(\boldsymbol{x}) \doteq \begin{bmatrix} \frac{\partial f(\boldsymbol{x})}{\partial x_1} \\ \frac{\partial f(\boldsymbol{x})}{\partial x_2} \\ \vdots \\ \frac{\partial f(\boldsymbol{x})}{\partial x_n} \end{bmatrix}, \quad \nabla^2 f(x) \doteq \begin{bmatrix} \frac{\partial^2 f(x)}{\partial x_1 \partial x_1} & \frac{\partial^2 f(x)}{\partial x_1 \partial x_2} & \cdots & \frac{\partial^2 f(\boldsymbol{x})}{\partial x_1 \partial x_n} \\ \frac{\partial^2 f(\boldsymbol{x})}{\partial x_2 \partial x_1} & \frac{\partial^2 f(\boldsymbol{x})}{\partial x_2 \partial x_2} & \cdots & \frac{\partial^2 f(\boldsymbol{x})}{\partial x_2 \partial x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial^2 f(\boldsymbol{x})}{\partial x_n \partial x_1} & \frac{\partial^2 f(\boldsymbol{x})}{\partial x_n \partial x_2} & \cdots & \frac{\partial^2 f(\boldsymbol{x})}{\partial x_n \partial x_n} \end{bmatrix}, \tag{A.1}$$

where $\boldsymbol{x} = \left[x_1, \ldots, x_n\right]^\top$. Sometimes, we use $\nabla_{\boldsymbol{x}} f(\boldsymbol{x}, \boldsymbol{y})$ to indicate the gradient with respect to $\boldsymbol{x}$ only, and similarly for the Hessian.

A.1.1 Optimality Conditions

We use $N(\boldsymbol{x}, \varepsilon)$ to denote an ε-ball around the point $\boldsymbol{x}$. We say that a point $\boldsymbol{x}^*$ is a local minimum of f if there exists an $\varepsilon > 0$ such that $f(\boldsymbol{x}^*) \leq f(\boldsymbol{x})$ for all $\boldsymbol{x} \in N(\boldsymbol{x}^*, \varepsilon)$. We say that $\boldsymbol{x}^*$ is a strict minimum if equality holds only if $\boldsymbol{x} = \boldsymbol{x}^*$. If the size of the neighborhood can be arbitrarily large, we say that $\boldsymbol{x}^*$ is the global minimum of the function.

It is not difficult to prove by contradiction (see Exercise A.1) that a necessary condition for a point $\boldsymbol{x}^*$ to be a local minimum is that the gradient $\nabla f(\boldsymbol{x})$ vanish at $\boldsymbol{x}^*$, or more precisely,

$$\nabla f(\boldsymbol{x}^*) = \mathbf{0}. \tag{A.2}$$

The following proposition gives sufficient conditions for a point $\boldsymbol{x}^*$ to be a local minimum in terms of its gradient and Hessian.

Proposition A.1 (Second-Order Sufficient Optimality Conditions)**.** *If a point $\boldsymbol{x}^* \in \mathbb{R}^n$ satisfies the conditions*

$$\nabla f(\boldsymbol{x}^*) = 0, \quad \nabla^2 f(\boldsymbol{x}^*) \succ \mathbf{0}, \tag{A.3}$$

then $\boldsymbol{x}^*$ is a (strict) local minimum of $f(x)$.

In practice, the above conditions can be used to find all possible local minima of a given function. Of course, in general, local minima of a function are often not unique and do not have to be the global minimum. However, if f is convex defined on a convex domain, then every local minimum must be the global minimum.

A.1.2 Convex Set and Convex Function

Definition A.2 (Convex Set)**.** *A set $\mathcal{X} \subseteq \mathbb{R}^n$ is said to be* convex *if for every $\boldsymbol{x}, \boldsymbol{y} \in \mathcal{X}$ and $\lambda \in [0, 1]$, we have $\lambda \boldsymbol{x} + (1 - \lambda)\boldsymbol{y} \in \mathcal{X}$.*

For convenience, the empty set is considered a special convex set. In most optimization problems that we consider in this book, we are searching for the minimum of a function over a convex domain. It is easy to verify many useful properties of convex sets. For example, the intersection of any two convex sets is also a convex set (see Exercise A.2).

Given any set (convex or not), we can associate with it a convex set as follows:

Definition A.3 (Convex Hull)**.** *Given a set $\mathcal{X} = \{\boldsymbol{x}_i\} \subseteq \mathbb{R}^n$, we define its* convex hull*, denoted by conv$(\mathcal{X})$, to be*

$$conv(\mathcal{X}) \doteq \left\{ \boldsymbol{y} : \boldsymbol{y} = \sum_{i=1}^{k} \lambda_i \boldsymbol{x}_i, \text{ where } k \in \mathbb{N}, \lambda_i \geq 0 \text{ and } \sum_{i=1}^{k} \lambda_i = 1 \right\}. \tag{A.4}$$

It is easy to show that a convex hull must be a convex set and that the convex hull of a convex set is the convex set itself (see Exercise A.2).

Definition A.4 (Convex Function)**.** *A function $f : \mathcal{X} \to \mathbb{R}$ defined on a convex domain $\mathcal{X} \subseteq \mathbb{R}^n$ is said to be* convex *if for all $\boldsymbol{x}, \boldsymbol{y} \in \mathcal{X}$ and $\lambda \in [0, 1]$, we have*

$$f(\lambda \boldsymbol{x} + (1 - \lambda)\boldsymbol{y}) \leq \lambda f(\boldsymbol{x}) + (1 - \lambda) f(\boldsymbol{y}). \tag{A.5}$$

We say that f is strictly convex if the inequality is strict for $\boldsymbol{x} \neq \boldsymbol{y}$ and $\lambda \in (0, 1)$.

Convex functions are extremely important for optimization largely because their minima and maxima have some very useful properties.

Theorem A.5 (Minima of Convex Function)**.** *If a convex function f defined over a convex domain $\mathcal{X} \subseteq \mathbb{R}^n$ has a minimum, then it has the following properties:*

1. *Every local minimum of f is also a global minimum.*
2. *The set of all minima of f is a convex set.*
3. *If the function f is strictly convex, it has a unique minimum $\boldsymbol{x}^*$.*

Proof. Let f^* denote the global minimum value of f over $\mathcal{X}$, and choose a point $\boldsymbol{x}^*$ where f reaches the global minimum value, i.e., $f(\boldsymbol{x}^*) = f^*$.

1. To prove the first statement, let us assume for the sake of contradiction that f has a local minimum at $\boldsymbol{y}^*$. Then, due to the convexity of f, we have that for all $\lambda \in [0, 1]$,

$$f(\lambda \boldsymbol{x}^* + (1 - \lambda)\boldsymbol{y}^*) = f(\boldsymbol{y}^* + \lambda(\boldsymbol{x}^* - \boldsymbol{y}^*)) \leq \lambda f(\boldsymbol{x}^*) + (1 - \lambda) f(\boldsymbol{y}^*).$$

   If $f(\boldsymbol{x}^*) < f(\boldsymbol{y}^*)$, then $\lambda f(\boldsymbol{x}^*) + (1 - \lambda) f(\boldsymbol{y}^*) < f(\boldsymbol{y}^*)$ for every $\lambda \in (0, 1]$. Therefore, we have $f(\boldsymbol{y}^* + \lambda(\boldsymbol{x}^* - \boldsymbol{y}^*)) < f(\boldsymbol{y}^*)$ for all $\lambda \in (0, 1]$. This contradicts the assumption that $\boldsymbol{y}^*$ is a local minimum of f.
2. To prove the second statement, we need to show that for every $c \in \mathbb{R}$, the set $\{x : f(x) \leq c\}$ is convex. We leave this as an exercise to the reader (see Exercise A.3). The claim then follows by choosing $c = f^*$.

3. To prove the third statement, let us assume for the sake of contradiction that f has two different local minima $\boldsymbol{x}^*$ and $\boldsymbol{y}^* \neq \boldsymbol{x}^*$. Due to the first statement, we have $f(\boldsymbol{x}^*) = f(\boldsymbol{y}^*)$. Since f is strictly convex, we further have

$$f(\lambda \boldsymbol{x}^* + (1-\lambda)\boldsymbol{y}^*) < \lambda f(\boldsymbol{x}^*) + (1-\lambda) f(\boldsymbol{y}^*) = f(\boldsymbol{x}^*)$$

for all $\lambda \in (0,1)$. Since the domain $\mathcal{X}$ is convex, $\lambda \boldsymbol{x}^* + (1-\lambda)\boldsymbol{y}^* \in \mathcal{X}$. This contradicts that $\boldsymbol{x}^*$ is the global minimum of f over $\mathcal{X}$. Therefore, the minimum $\boldsymbol{x}^*$ must be unique.

□

Sometimes, we are also interested in the maximum value of a convex function f over a convex set $\mathcal{X}$. We have the following statement.

Theorem A.6 (Maxima of Convex Function over Compact Convex Domain)**.** *Let f be a convex function defined on a compact convex domain $\mathcal{X}$. Then f reaches its maximum value at the boundary of $\mathcal{X}$. More precisely, we have*

$$\max_{x \in \mathcal{X}} f(\boldsymbol{x}) = \max_{x \in \partial \mathcal{X}} f(\boldsymbol{x}),$$

where $\partial \mathcal{X}$ denotes the boundary of the set $\mathcal{X}$.

We leave the proof as an exercise for the reader to become familiar with the properties of convex functions (see Exercise A.3).

Besides the above notion of (strict) convexity, the following two relaxed notions of convexity are also often used.

Definition A.7 (Quasiconvex)**.** *A function $f : \mathcal{X} \to \mathbb{R}$ defined on a convex domain $\mathcal{X} \subseteq \mathbb{R}^n$ is said to be* quasiconvex *if for all $\boldsymbol{x}, \boldsymbol{y} \in \mathcal{X}$ and $\lambda \in [0,1]$, we have*

$$f(\lambda \boldsymbol{x} + (1-\lambda)\boldsymbol{y}) \leq \max\{f(\boldsymbol{x}), f(\boldsymbol{y})\}.$$

Definition A.8 (Pseudoconvex)**.** *A function $f : \mathcal{X} \to \mathbb{R}$ defined on a convex domain $\mathcal{X} \subseteq \mathbb{R}^n$ is said to be* pseudoconvex *if for all $\boldsymbol{y} \in \mathbb{R}^n$ such that $\nabla f(\boldsymbol{x})^\top (\boldsymbol{y} - \boldsymbol{x}) \geq 0$, we have*

$$f(\boldsymbol{x}) \leq f(\boldsymbol{y}).$$

A.1.3 Subgradient

Sometimes, the function we are trying to minimize is not necessarily smooth everywhere. In this case, the "gradient" of the function cannot be evaluated at every point. This leads to a generalized notion of the gradient called a *subgradient*.

Definition A.9 (Subgradient of a Convex Function)**.** *The subgradient of a convex function $f : \mathcal{X} \to \mathbb{R}$ at a point $\boldsymbol{x} \in \mathcal{X}$, where $\mathcal{X}$ is convex, is defined to be the set*

$$\partial f(\boldsymbol{x}) \doteq \{\boldsymbol{v} \in \mathbb{R}^n : f(\boldsymbol{y}) \geq f(\boldsymbol{x}) + \boldsymbol{v}^\top(\boldsymbol{y} - \boldsymbol{x}),\ \forall \boldsymbol{y} \in \mathcal{X}\}. \tag{A.6}$$

Most conditions and results for minimizing a smooth convex function generalize to a nonsmooth convex function if one replaces gradient with subgradient. For instance, a point $\boldsymbol{x}^*$ is a minimum of a convex function f if and only if $\mathbf{0} \in \partial f(\boldsymbol{x}^*)$.

A.1.4 Gradient Descent Algorithm

There is an extremely rich history and literature on how to optimize a function. For many of the problems in this book, we are mostly interested in a simple method that can be easily implemented to obtain the optimal solution. Hence, in this section, we introduce a few simple methods that are pertinent to these problems, even though they do not necessarily represent the most advanced optimization techniques.

Almost all methods for minimizing a function f are based on a very simple idea. We begin with an initial guess $\boldsymbol{x} = \boldsymbol{x}^0$, and successively update $\boldsymbol{x}$ to $\boldsymbol{x}^1, \boldsymbol{x}^2, \ldots$, such that the value $f(\boldsymbol{x})$ decreases at each iteration; that is, $f(\boldsymbol{x}^{i+1}) \leq f(\boldsymbol{x}^i)$. Of course, the safest way to ensure a decrease of the value of the objective function is to follow the "direction of descent," which in our case would be the opposite direction to the gradient vector $\nabla f(\boldsymbol{x}^i)$. This idea gives rise to the classic *steepest descent method* for searching for the minimum. At each iteration, the variables are updated as

$$\boldsymbol{x}^{i+1} = \boldsymbol{x}^i - \alpha^i \nabla f(\boldsymbol{x}^i), \tag{A.7}$$

for some scalar $\alpha^i > 0$, called the *step size*.

There exist many different choices for the step size α^i, and the simplest one is of course to set it to be a small constant, but that does not always result in a decrease in the value of $f(x)$ at each iteration. Instead, α^i is often chosen to be the value α^* that is given by solving a one-dimensional minimization problem:

$$\alpha^* = \arg\min_{\alpha \geq 0} f(\boldsymbol{x}^i - \alpha \nabla f(\boldsymbol{x}^i)). \tag{A.8}$$

This is called the *minimization rule*.

Although the vector $-\nabla f(\boldsymbol{x}^i)$ points to the steepest descent direction locally around $\boldsymbol{x}^i$, it is not necessarily the best choice for searching for the minimum at a larger scale. For instance, if $f(\boldsymbol{x})$ can be approximated by a quadratic function $f(\boldsymbol{x}) \approx \frac{1}{2}(\boldsymbol{x} - \boldsymbol{x}^*)^\top K(\boldsymbol{x} - \boldsymbol{x}^*) + c$ and the matrix K has very large condition number,[1]

[1] That is, the ratio $\text{cond}(K) = \frac{\lambda_{\max}}{\lambda_{\min}}$ between the largest and smallest eigenvalues of K is large.

then the simple gradient descent method typically has very poor convergence. In general, it is easy to establish that the error $f(\boldsymbol{x}^i) - f(\boldsymbol{x}^*)$ of gradient-based methods necessarily drops on the order of $o(i^{-1})$. Further, for a general class of objective functions, one can show that the optimal rate of convergence for gradient-based methods does not exceed $o(i^{-2})$ (Nemirovskii and Yudin 1979).

To improve the convergence of the gradient descent method, one can generalize the variable update equation to the form

$$\boldsymbol{x}^{i+1} = \boldsymbol{x}^i - \alpha^i D^i \nabla f(\boldsymbol{x}^i), \tag{A.9}$$

where $D^i \in \mathbb{R}^{n \times n}$ is a positive definite symmetric matrix to be determined in each particular algorithm. The steepest descent method in (A.7) becomes a particular case of (A.9), where $D^i \equiv I$. In general, D^i can be viewed as a weight matrix that adjusts the descent direction according to more sophisticated local information about the function f than the gradient alone. A simple choice for D^i would be a diagonal matrix that scales the descent speed differently in each axial direction. A more principled choice for D^i would be the inverse of the Hessian $D^i = [\nabla^2 f(x^i)]^{-1}$, which gives the classical Newton's method. This method typically has a much faster convergence rate than simple gradient-based methods. For example, it finds the minimum of a quadratic function in one step. In general, it can also be established that under fairly general conditions, optimization schemes based on Newton-type iterations often have a linear convergence rate, that is, the error $f(\boldsymbol{x}^i) - f(\boldsymbol{x}^*)$ reduces on the order of $o(\rho^i)$ for some $\rho \in (0, 1)$.

Despite the fast convergence of Newton's method, in many modern high-dimensional optimization problems that we encounter in this book, this choice is not very practical, because it is extremely costly to compute and store the Hessian matrix and its inverse. Hence, most modern optimization methods for large-scale optimization rely on smart modifications to the classical gradient descent method that are based on only first-order derivatives of the objective function. For interested readers, we point to the seminal work of (Nesterov 1983; Beck and Teboulle 2009) on *accelerated proximal gradient* algorithms that achieve a convergence rate of $o(i^{-2})$ for a large class of convex objective functions.

A.1.5 Alternating Direction Minimization

In many optimization problems that we encounter in this book, we are required to minimize an objective function f that has special structures. For example, if we partition the variables $\boldsymbol{x} \in \mathbb{R}^n$ into, say, N blocks $\boldsymbol{x} = (x_1, \ldots, x_N)$, it may be very convenient to minimize f with respect to one block of variables at a time.

Such methods are also known in the optimization literature as *block coordinate descent* (BCD) methods (Tseng 2001) or alternating direction minimization (ADM) methods, especially when $N = 2$. For example, in the matrix factorization problem

discussed in Section 2.1.2, our goal is to obtain a factorization (U, V) that best approximates a given matrix M by minimizing the objective function

$$\|M - UV^\top\|_F^2. \tag{A.10}$$

If we fix one factor, say U, then finding the best V that minimizes the error is a simple quadratic problem and has a closed-form solution. Hence, it is rather natural to minimize such an objective function by iteratively minimizing with respect one factor at a time. As another example, in some of the convex optimization problems that we utilize for recovering low-rank matrices or sparse vectors, the objective function is often of the special form

$$f(\boldsymbol{x}) = f_0(\boldsymbol{x}) + \sum_{i=1}^{N} f_i(x_i), \tag{A.11}$$

where $f_i(\cdot)$ is a component that depends only on the ith block variables x_i, and $f_0(x)$ typically is a simple function with nice properties. Such a function is said to have a *separable* form. Again, it is natural to minimize such a function in a (block) coordinate descent fashion especially if $f_0(\boldsymbol{x}) + f_i(x_i)$ is much easier to minimize with respect to each coordinate block x_i.

Below, we formally describe the block coordinate descent (BCD) method, as a special version of what was described in (Tseng 2001):

- Initialization. Choose any $\boldsymbol{x}^0 = (x_1^0, \ldots, x_N^0) \in \mathbb{R}^n$.
- For the $(i+1)$th iteration, $i \geq 0$, given $\boldsymbol{x}^i = (x_1^i, \ldots, x_N^i) \in \mathbb{R}^n$ from the previous iteration, choose $s = i \,(\mathrm{mod}\, N)$ and compute
  - $x_s^{i+1} = \arg\min_{x_s} f(x_1^i, \ldots, x_{s-1}^i, x_s, x_{s+1}^i, \ldots, x_N^i)$;
  - $x_j^{i+1} = x_j^i, \quad \forall j \neq s.$
- Repeat the process till convergence or the maximum number of iterations has been reached.

Although the above alternating minimization scheme is very widely used in engineering solutions for real-world optimization problems, theoretically it is important to know about when it is guaranteed to converge, at least to a local minimum of the objective function f. There has been a vast amount of classical literature that characterizes the convergence of the BCD method for various classes of objective functions. We here summarize some of the well-known convergence results, which are helpful in justifying the optimization techniques used for problems in this book. For detailed and rigorous proofs of these results, we refer the reader to the references given below.

Proposition A.10 (Convergence of Block Coordinate Descent)**.** *Given a function $f : \mathbb{R}^n \to \mathbb{R}$ bounded from below, the BCD method converges to a stationary point of f under each of the following conditions:*

- *The function f is strictly convex (Warga 1963).*
- *The function f is pseudoconvex (Zadeh 1970).*
- *The function f is quadratic (Luo and Tseng 1993).*
- *The function f is pseudoconvex in each pair of blocks (x_j, x_k) for every $j, k \in \{1, \ldots N\}$ (Tseng 2001).*
- *The function f has unique minimum in each coordinate block (Luenberger 1973).*

In fact, if the function f is not pseudoconvex, a counterexample (Powell 1973) exists in which the method may cycle without approaching any stationary point of f. The last result suggests that the alternating minimization scheme for the matrix factorization problem is guaranteed to converge to a stationary point.

A.2 Constrained Optimization

In this section, we consider the problem of minimizing a function $f : \mathbb{R}^n \to \mathbb{R}$ subject to equality constraints on the variable $\boldsymbol{x} \in \mathbb{R}^n$, i.e.,

$$\boldsymbol{x}^* = \arg\min f(\boldsymbol{x}) \quad \text{subject to} \quad h(\boldsymbol{x}) = 0, \tag{A.12}$$

where $h = [h_1, h_2, \ldots, h_m]^\top$ is a smooth (multidimensional) function (or map) from $\mathbb{R}^n$ to $\mathbb{R}^m$. For each constraint $h_i(\boldsymbol{x}) = 0$ to be independently effective at the minimum $\boldsymbol{x}^*$, we often assume that their gradients

$$\nabla h_1(\boldsymbol{x}^*),\ \nabla h_2(\boldsymbol{x}^*),\ \ldots,\ \nabla h_m(\boldsymbol{x}^*) \quad \in \mathbb{R}^n \tag{A.13}$$

are linearly independent. If so, the constraints are called *regular*.

A.2.1 Optimality Conditions and Lagrangian Multipliers

For simplicity, we always assume that the functions f and h are at least twice continuously differentiable. Then the main theorem of Lagrange is as follows.

Theorem A.11 (Lagrange multiplier theorem; necessary conditions)**.** *Let $\boldsymbol{x}^*$ be a local minimum of a function f subject to regular constraints $h(\boldsymbol{x}^*) = 0$. Then there exists a unique vector $\lambda^* = [\lambda_1^*, \lambda_2^*, \ldots, \lambda_m^*]^\top \in \mathbb{R}^m$, called* Lagrange multipliers*, such that*

$$\nabla f(\boldsymbol{x}^*) + \sum_{i=1}^{m} \lambda_i^* \nabla h_i(\boldsymbol{x}^*) = \boldsymbol{0}. \tag{A.14}$$

Furthermore, we have

$$\boldsymbol{v}^\top \Big(\nabla^2 f(\boldsymbol{x}^*) + \sum_{i=1}^{m} \lambda_i^* \nabla^2 h_i(\boldsymbol{x}^*) \Big) \boldsymbol{v} \geq \boldsymbol{0} \tag{A.15}$$

for all vectors $\boldsymbol{v} \in \mathbb{R}^n$ *that satisfy* $\nabla h_i(\boldsymbol{x}^*)^\top \boldsymbol{v} = 0$, *for* $i = 1, 2, \ldots, m$.

Theorem A.12 (Lagrange multiplier theorem; sufficient conditions)**.** *Assume that* $\boldsymbol{x}^* \in \mathbb{R}^n$ *and* $\lambda^* = [\lambda_1^*, \lambda_2^*, \ldots, \lambda_m^*]^\top \in \mathbb{R}^m$ *satisfy*

$$\nabla f(\boldsymbol{x}^*) + \sum_{i=1}^{m} \lambda_i^* \nabla h_i(\boldsymbol{x}^*) = 0, \quad h_i(\boldsymbol{x}^*) = 0, \ i = 1, 2, \ldots, m, \tag{A.16}$$

and furthermore, we have

$$\boldsymbol{v}^\top \Big(\nabla^2 f(\boldsymbol{x}^*) + \sum_{i=1}^{m} \lambda_i^* \nabla^2 h_i(\boldsymbol{x}^*) \Big) \boldsymbol{v} \geq \boldsymbol{0}, \tag{A.17}$$

for all vectors $\boldsymbol{v} \in \mathbb{R}^n$ *that satisfy* $\nabla h_i(\boldsymbol{x}^*)^\top \boldsymbol{v} = 0$, *for* $i = 1, 2, \ldots, m$. *Then* $\boldsymbol{x}^*$ *is a strict local minimum of* f *subject to* $h(\boldsymbol{x}) = 0$.

The Lagrangian function
If we define for convenience the *Lagrangian function* $\mathscr{L} : \mathbb{R}^{n+m} \to \mathbb{R}$ as

$$\mathscr{L}(\boldsymbol{x}, \lambda) \doteq f(\boldsymbol{x}) + \lambda^\top h(\boldsymbol{x}), \tag{A.18}$$

then the necessary conditions in Theorem A.11 can be written as

$$\nabla_{\boldsymbol{x}} \mathscr{L}(\boldsymbol{x}^*, \lambda^*) = \boldsymbol{0}, \qquad \nabla_{\lambda} \mathscr{L}(\boldsymbol{x}^*, \lambda^*) = \boldsymbol{0}, \tag{A.19}$$

$$\boldsymbol{v}^\top \nabla_{\boldsymbol{x}}^2 \mathscr{L}(\boldsymbol{x}^*, \lambda^*) \boldsymbol{v} \geq \boldsymbol{0}, \ \forall \boldsymbol{v} : \boldsymbol{v}^\top \nabla h(\boldsymbol{x}^*) = \boldsymbol{0}. \tag{A.20}$$

The conditions (A.19) give a system of $n + m$ equations with $n + m$ unknowns: the entries of $\boldsymbol{x}^*$ and λ^*. If the constraint $h(\boldsymbol{x}) = 0$ is regular, then in principle, this system of equations is independent, and we should be able to solve for $\boldsymbol{x}^*$ and λ^*. The solutions will contain all the (local) minima, but it is possible that some of them need not be minima at all. Nevertheless, whether we are able to solve these equations or not, they usually provide rich information about the minima of the constrained optimization. We illustrate how we can utilize the necessary conditions for Lagrange multipliers to find the optimal solution to a constrained optimization problem with the following example.

Example A.13 [Matrix Lagrange Multipliers] Consider the problem of projecting a given matrix $M \in \mathbb{R}^{n \times n}$ onto the space of orthogonal matrices $O(n) = \{U \in \mathbb{R}^{n \times n} : U^\top U = I\}$. That is, we want to find a matrix $U \in \mathbb{R}^{n \times n}$ that minimizes

$$\min_U \|M - U\|_F^2 \quad \text{subject to} \quad U^\top U = I. \tag{A.21}$$

Notice that there are n^2 constraints in $U^\top U = I$. This suggests using n^2 Lagrange multipliers, which can be conveniently represented as the entries of a matrix $\Lambda \in \mathbb{R}^{n\times n}$. However, since the matrix $U^\top U$ is symmetric, there are only $n(n+1)/2$ independent constraints. Therefore, the matrix Λ needs to be chosen to be symmetric. Now, since the inner product between the two matrices A and B can be conveniently written as $\langle A, B\rangle = \text{trace}(A^\top B)$, the Lagrangian function can be written as

$$\mathscr{L}(U, \Lambda) = \|M - U\|_F^2 + \text{trace}(\Lambda(U^\top U - I)). \tag{A.22}$$

The necessary condition $\frac{\partial \mathscr{L}}{\partial U} = \mathbf{0}$ in Theorem A.11 gives

$$(U - M) + U\Lambda = \mathbf{0}. \tag{A.23}$$

This gives $\Lambda = U^\top M - I$. Since Λ is symmetric, so is $U^\top M$. Let $M = W\Sigma V^\top$ be the singular value decomposition of M. Both W, V are orthogonal matrices. If the singular values of M are all different, then in order for $U^\top M = U^\top W\Sigma V^\top$ to be symmetric, we must have $U^\top W = V$; hence $U = WV^\top$.

As we see from the above example, for some constrained optimization problems, the necessary conditions of the Lagrangian alone allow us to solve for the optimal solution. In general, of course, this is not always possible, and we have to resort to numerical solutions to find the optimal solution.

A.2.2 Augmented Lagrange Multipler Methods

If we are not able to solve for the minima from the equations given by the necessary conditions, we must resort to a numerical optimization scheme. The basic idea is to try to convert the original constrained optimization to an unconstrained one by introducing extra *penalty* terms to the objective function. A typical choice is the *augmented Lagrangian function* $\mathscr{L}_c : \mathbb{R}^{n+m} \to \mathbb{R}$, defined as

$$\mathscr{L}_c(\boldsymbol{x}, \lambda) \doteq f(\boldsymbol{x}) + \lambda^\top h(\boldsymbol{x}) + \frac{c}{2}\|h(\boldsymbol{x})\|^2, \tag{A.24}$$

where $c > 0$ is a positive penalty parameter. It is reasonable to expect that for very large c, the location $\boldsymbol{x}^*$ of the global minimum of the unconstrained minimization

$$(\boldsymbol{x}^*, \lambda^*) = \arg\max_{\lambda} \min_{\boldsymbol{x}} \mathscr{L}_c(\boldsymbol{x}\lambda) \tag{A.25}$$

should be very close to the global minimum of the original constrained minimization.

Proposition A.14 (Convergence of ALM (Bertsekas 1999)). *For $i = 0, 1, \ldots$, let $\boldsymbol{x}^i$ be a global minimum of the unconstrained optimization problem*

$$\min_{\boldsymbol{x}} \mathscr{L}_{c^i}(\boldsymbol{x}, \lambda^i), \tag{A.26}$$

where $\{\lambda^i\}$ is bounded, $0 < c^i < c^{i+1}$ for all i, and $c^i \to \infty$. Then the limit of the sequence $\{\boldsymbol{x}^i\}$ is a global minimum of the original constrained optimization problem.

This result leads to the classical augmented Lagrangian algorithm for solving the constrained optimization problem (A.12) via the following iteration:

$$\begin{aligned} \boldsymbol{x}^{i+1} &= \arg\min_{\boldsymbol{x}} \mathscr{L}_{c_i}(\boldsymbol{x}, \lambda^i), \\ \lambda^{i+1} &= \lambda^i + c^{i+1} h(\boldsymbol{x}^{i+1}). \end{aligned} \tag{A.27}$$

It is easy to see that if $\{\lambda^i\}$ is a bounded sequence and $c^i \to \infty$, then we must have $h(\boldsymbol{x}^i) \to \mathbf{0}$; hence the constraint will be enforced at the point $\boldsymbol{x}^*$ to which the algorithm converges. Moreover, the limit point λ^* of the bounded sequence $\{\lambda^i\}$ will be the desired Lagrange multiplier in Theorem A.11.

A.2.3 Alternating Direction Method of Multipliers

A very common class of optimization problems that one encounters in practice is to optimize some convex objective function subject to a set of linear constraints. Very often, including some cases we have encountered in this book, the objective function f has a separable form that makes it amenable to simpler optimization schemes such as alternating minimization, discussed earlier. For example, consider the following optimization problem:

$$\min_{\boldsymbol{x}} f(\boldsymbol{x}) = \sum_{i=1}^{N} f_i(x_i), \quad \text{subject to} \quad \sum_{i=1}^{N} A_i x_i = \boldsymbol{b}, \tag{A.28}$$

where each f_i is a convex function. In some cases, the component functions f_i need not be smooth. For instance, in the robust PCA problem discussed in Section 3.2, we aim to solve

$$\min_{L,S} \|L\|_* + \lambda \|S\|_1, \quad \text{subject to} \quad D = L + S, \tag{A.29}$$

where the nuclear norm $\|\cdot\|_*$ and the ℓ^1-norm $\|\cdot\|_1$ are not smooth.

In this subsection, we show how to use the augmented Lagrangian method to solve this class of optimization problems in an effective and efficient way. Notice that the augmented Lagrangian function for this class of problems precisely

resembles the separable form (A.11) studied earlier. We are particularly interested in simple and scalable algorithms that utilize only first-order information of the objective function and do not involve any expensive computations.

Most of the cases in which we are interested in this book involve (or can be reduced to) only two terms, say

$$\min_{\boldsymbol{x},\boldsymbol{y}} f(\boldsymbol{x}) + g(\boldsymbol{y}) \quad \text{subject to} \quad A\boldsymbol{x} + B\boldsymbol{y} = \boldsymbol{b}, \tag{A.30}$$

where $f : \mathbb{R}^n \to \mathbb{R}$ and $g : \mathbb{R}^p \to \mathbb{R}$ are two convex functions, and $A \in \mathbb{R}^{m\times n}$, $B \in \mathbb{R}^{m\times p}$, and $\boldsymbol{b} \in \mathbb{R}^m$ together specify m linear constraints. For simplicity, we will first illustrate the basic algorithm and results using the two-term problem, and we will later discuss how to generalize to multiple terms.

Let us define the augmented Lagrangian function for problem (A.30):

$$\mathscr{L}_\mu(\boldsymbol{x},\boldsymbol{y};\lambda) \doteq f(\boldsymbol{x}) + g(\boldsymbol{y}) + \langle \lambda, A\boldsymbol{x} + B\boldsymbol{y} - \boldsymbol{b}\rangle + \frac{1}{2\mu}\|A\boldsymbol{x} + B\boldsymbol{y} - \boldsymbol{b}\|_2^2, \tag{A.31}$$

where $\mu > 0$ is a penalty parameter. According to the augmented Lagrangian method, μ should be a decreasing sequence converging to 0. Then, following the classical augmented Lagrangian method (Bertsekas 1999), we can solve problem (A.30) via the following iteration:

$$\begin{aligned} (\boldsymbol{x}^{i+1},\boldsymbol{y}^{i+1}) &= \arg\min_{\boldsymbol{x},\boldsymbol{y}} \mathscr{L}_\mu(\boldsymbol{x},\boldsymbol{y};\lambda^i), \\ \lambda^{i+1} &= \lambda^i + (A\boldsymbol{x}^{i+1} + B\boldsymbol{y}^{i+1} - \boldsymbol{b})/\mu. \end{aligned} \tag{A.32}$$

However, the joint minimization over both $\boldsymbol{x}$ and $\boldsymbol{y}$ can be very difficult. Fortunately, as in the case of the robust PCA problem, the minimization over $\boldsymbol{x}$ or $\boldsymbol{y}$ with the other variables fixed is often much simpler. This leads to the *alternating direction method of multipliers* (ADMM), which follows the following iteration scheme:

$$\begin{aligned} \boldsymbol{x}^{i+1} &= \arg\min_{\boldsymbol{x}} \mathscr{L}_\mu(\boldsymbol{x},\boldsymbol{y}^i;\lambda^i), \\ \boldsymbol{y}^{i+1} &= \arg\min_{\boldsymbol{y}} \mathscr{L}_\mu(\boldsymbol{x}^{i+1},\boldsymbol{y};\lambda^i), \\ \lambda^{i+1} &= \lambda^i + (A\boldsymbol{x}^{i+1} + B\boldsymbol{y}^{i+1} - \boldsymbol{b})/\mu. \end{aligned} \tag{A.33}$$

This alternating direction technique is known as the Douglas–Rachford operator splitting method and is known to converge to the global optimal solution (see (Ma 2012) and references therein).

In the robust PCA problem, both A and B are identity operators, and the associated optimization problems for the two alternating minimizations are both very simple to solve. For instance, the minimization with respect to the sparse term is $\min_S \|S\|_1 + \alpha\|S - M\|$ for some fixed matrix M and constant α. The solution is

given by a simple entrywise soft thresholding. The minimization with respect to the low-rank term is a simple singular-value soft thresholding.

However, in many other problems, the operators A and B are not necessarily the identities, and the problem of minimizing each component may no longer be so simple, even for a case such as the ℓ^1-norm. Note that by completing the squares, we can write the iteration scheme (A.33) explicitly as

$$\begin{aligned} \boldsymbol{x}^{i+1} &= \arg\min_{\boldsymbol{x}} f(\boldsymbol{x}) + \frac{1}{2\mu}\|A\boldsymbol{x} + B\boldsymbol{y}^i - \boldsymbol{b} + \mu\lambda^i\|_2^2, \\ \boldsymbol{y}^{i+1} &= \arg\min_{\boldsymbol{y}} g(\boldsymbol{y}) + \frac{1}{2\mu}\|A\boldsymbol{x}^{i+1} + B\boldsymbol{y} - \boldsymbol{b} + \mu\lambda^i\|_2^2, \\ \lambda^{i+1} &= \lambda^i + (A\boldsymbol{x}^{i+1} + B\boldsymbol{y}^{i+1} - \boldsymbol{b})/\mu. \end{aligned} \tag{A.34}$$

Although one can always resort to some iterative scheme to find the minimum solutions to the above two subproblems, the computational cost can be very high. One technique proposed to simplify the above minimization approximates the quadratic penalty term $\frac{1}{2}\|A\boldsymbol{x} + B\boldsymbol{y}^i - \boldsymbol{b} + \mu\lambda^k\|_2^2$ with another proximal quadratic term

$$\begin{aligned} &\frac{1}{2\tau_1}\|\boldsymbol{x} - (\boldsymbol{x}^i - \tau_1 A^\top(A\boldsymbol{x}^i + B\boldsymbol{y}^i - \boldsymbol{b} + \mu\lambda^i))\|_2^2 \\ &\quad = \langle \boldsymbol{x} - \boldsymbol{x}^i, A^\top(A\boldsymbol{x}^i + B\boldsymbol{y}^i - \boldsymbol{b} + \mu\lambda^i)\rangle + \frac{1}{2\tau_1}\|\boldsymbol{x} - \boldsymbol{x}^i\|_2^2 + c, \end{aligned} \tag{A.35}$$

where c is a constant. Notice that this term can be interpreted to approximate the original quadratic term with its Taylor expansion at the previous iteration point x^i up to the second-order term, where $A^\top(A\boldsymbol{x}^i + B\boldsymbol{y}^i - \boldsymbol{b} + \mu\lambda^i)$ is the gradient of the quadratic term at $\boldsymbol{x}^i$, but the Hessian $A^\top A$ is approximated with a constant $1/\tau_1$. To ensure that the approximation is an upper bound of the original function, we want $\tau_1 < 1/\lambda_{\max}(A^\top A)$. If we do the same for the subproblem for updating $\boldsymbol{y}$, then the ADMM iteration scheme can be replaced by the so-called *alternating proximal gradient minimization* (APGM) scheme:

$$\begin{aligned} \boldsymbol{x}^{i+1} &= \arg\min_{\boldsymbol{x}} f(\boldsymbol{x}) + \frac{1}{2\mu\tau_1}\|\boldsymbol{x} - (\boldsymbol{x}^i - \tau_1 A^\top(A\boldsymbol{x}^i + B\boldsymbol{y}^i - \boldsymbol{b} + \mu\lambda^i))\|_2^2, \\ \boldsymbol{y}^{i+1} &= \arg\min_{\boldsymbol{y}} g(\boldsymbol{y}) + \frac{1}{2\mu\tau_2}\|\boldsymbol{y} - (\boldsymbol{y}^i - \tau_2 B^\top(A\boldsymbol{x}^{i+1} + B\boldsymbol{y}^i - \boldsymbol{b} + \mu\lambda^i))\|_2^2, \\ \lambda^{i+1} &= \lambda^i + (A\boldsymbol{x}^{i+1} + B\boldsymbol{y}^{i+1} - \boldsymbol{b})/\mu. \end{aligned} \tag{A.36}$$

Could the approximation affect the convergence of the ADMM method? The following result ensures that the global convergence remains intact if ADMM is replaced with APGM.

Proposition A.15 (Convergence of ADMM with Proximal Gradient (Ma 2012))**.** *For* $\tau_1 < 1/\lambda_{\max}(A^\top A)$ *and* $\tau_2 < 1/\lambda_{\max}(B^\top B)$*, the sequence* $\{(\boldsymbol{x}^i, \boldsymbol{y}^i, \lambda^i)\}$ *produced by the above APGM scheme* (A.36) *converges to the global optimal solution of problem* (A.30)*.*

This result is very useful. Although it is established only for the two-term problem, it essentially offers an effective solution for the multiterm problem (A.28): we can always partition the N terms into two blocks and apply the APGM scheme. The convergence is ensured. Of course, in practice, the speed of convergence could be different for different partitions.

A.3 Exercises

Exercise A.1 Show that a necessary condition for a point $\boldsymbol{x}^*$ to be a local minimum of a differentiable function f is that the gradient $\nabla f(\boldsymbol{x})$ vanish at $\boldsymbol{x}^*$, i.e., $\nabla f(\boldsymbol{x}^*) = 0$.

Exercise A.2 Show that:

1. The intersection of two convex sets is convex.
2. The convex hull of a set is convex.
3. The convex hull of a convex set is the set itself.

Exercise A.3 Let $f : \mathcal{X} \to \mathbb{R}$ be a convex function defined over a convex domain $\mathcal{X} \subseteq \mathbb{R}^n$. Show that:

1. For every $c \in \mathbb{R}$, the set $\{\boldsymbol{x} \in \mathcal{X} : f(\boldsymbol{x}) \leq c\}$ is convex.
2. If $\mathcal{X}$ is compact, then f reaches its maximum value at the boundary of $\mathcal{X}$, i.e., $\max_{\boldsymbol{x}\in\mathcal{X}} f(\boldsymbol{x}) = \max_{\boldsymbol{x}\in\partial\mathcal{X}} f(\boldsymbol{x})$.
3. f is pseudoconvex.
4. f is quasiconvex.

Appendix B
Basic Facts from Mathematical Statistics

A knowledge of statistics is like a knowledge of foreign languages or of algebra; it may prove of use at any time under any circumstances.

—A.L. Bowley

In the practice of science and engineering, data are often modeled as samples of a random variable (or vector) drawn from a certain probability distribution. Mathematical statistics deals with the problem of inferring the underlying distribution from the given samples. To render the problem tractable, we typically assume that the unknown distribution belongs to some parametric family (e.g., Gaussian), and formulate the problem as one of estimating the parameters of the distribution from the samples.

In this appendix, we provide a brief review of some of the most relevant concepts and results from mathematical statistics used in this book. The review is not meant to be exhaustive, but rather to make the book self-contained for readers who already have some basic knowledge in probability theory and statistics. For a more formal and thorough introduction to mathematical statistics, we refer the reader to the classic books (Wilks 1962) and (Bickel and Doksum 2000).

B.1 Estimation of Parametric Models

Assume that you are given independent and identically distributed (i.i.d.) samples from an unknown parametric distribution from which you wish to estimate some properties of the distribution. In this section, we show how to estimate the parameters of the distribution, such as the mean and variance, from the i.i.d. samples.

R. Vidal et al., *Generalized Principal Component Analysis*, Interdisciplinary Applied Mathematics 40, DOI 10.1007/978-0-387-87811-9

We study different types of estimators, such as minimum variance and maximum likelihood estimators, and their properties, such as unbiasedness, efficiency, and consistency.

B.1.1 Sufficient Statistics

Let $\boldsymbol{x}$ be a random variable or vector. For simplicity, we assume that the distribution of $\boldsymbol{x}$ has a density $p_\theta(\boldsymbol{x})$, where the parameter vector $\theta = [\theta_1, \theta_2, \dots, \theta_d]^\top \in \Theta \subset \mathbb{R}^d$, once known, uniquely determines the density function $p_\theta(\cdot)$. Now suppose that $\mathcal{X} = \{\boldsymbol{x}_j\}_{j=1}^N$ is a set of i.i.d. samples of $\boldsymbol{x}$ drawn according to the density $p_\theta(\boldsymbol{x})$. Then $\mathcal{X}$ has the density

$$p_\theta(\mathcal{X}) = \prod_{j=1}^{N} p_\theta(\boldsymbol{x}_j). \tag{B.1}$$

We call any real or vector-valued function of $\mathcal{X}$ a *statistic* and denote it by $T(\mathcal{X})$. The goal is to choose a function $T(\cdot)$ that gives a "good" estimate of the true parameter θ. To that end, we introduce the concept of *sufficient statistics*.

Definition B.1 (Sufficient Statistic)**.** *A statistic $T(\mathcal{X})$ is said to be* sufficient *for θ if the conditional distribution of $\mathcal{X}$ given $T(\mathcal{X})$, $p_\theta(\mathcal{X} \mid T(\mathcal{X}))$ is not a function of θ.*

Intuitively, a sufficient statistic $T(\mathcal{X})$ with respect to θ is a statistic that contains all the information that is useful to estimate θ. In other words, we can throw away the given samples and estimate θ from $T(\mathcal{X})$ without any loss of information. Unfortunately, the above definition is not very useful for finding sufficient statistics. Instead, one typically resorts to the following factorization theorem.

Theorem B.2 (Fisher–Neyman)**.** *A statistic $T(\mathcal{X})$ is sufficient for θ if and only if there exist a function $g(t, \theta)$ and a function $h(\mathcal{X})$ such that*

$$p_\theta(\mathcal{X}) = g(T(\mathcal{X}), \theta) h(\mathcal{X}). \tag{B.2}$$

Example B.3 (Sufficient Statistic of a Gaussian Random Variable). For Gaussian data $\boldsymbol{x}_j \sim \mathcal{N}(\boldsymbol{\mu}, \Sigma)$, where $\boldsymbol{x}_j \in \mathbb{R}^D$, $\boldsymbol{\mu} \in \mathbb{R}^D$ and $\Sigma \in \mathbb{R}^{D \times D}$, the statistic $T(\mathcal{X}) = (\sum_{j=1}^N \boldsymbol{x}_j, \sum_{j=1}^N \boldsymbol{x}_j \boldsymbol{x}_j^\top)$ is a sufficient statistic for $\theta = (\boldsymbol{\mu}, \Sigma)$, because

$$\begin{aligned} p_\theta(\mathcal{X}) &= \prod_{j=1}^{N} \frac{1}{(2\pi)^{D/2} \det(\Sigma)^{1/2}} \exp\Big(-\frac{(\boldsymbol{x}_j - \boldsymbol{\mu})^\top \Sigma^{-1} (\boldsymbol{x}_j - \boldsymbol{\mu})}{2}\Big) \\ &= \frac{\exp\big(-\frac{1}{2}(\operatorname{trace}\big(\Sigma^{-1} \sum_{j=1}^N \boldsymbol{x}_j \boldsymbol{x}_j^\top\big) - 2\boldsymbol{\mu}^\top \Sigma^{-1} \sum_{j=1}^N \boldsymbol{x}_j + \boldsymbol{\mu}^\top \boldsymbol{\mu})\big)}{(2\pi)^{ND/2} \det(\Sigma)^{N/2}} \\ &= g(T(\mathcal{X}), \theta) \cdot 1. \end{aligned} \tag{B.3}$$

B.1.2 Mean Square Error, Efficiency, and Fisher Information

Notice that sufficient statistics are not unique. For instance, $T(\mathcal{X}) = \mathcal{X}$ is a sufficient statistic, and every one-to-one function of a sufficient statistic is also a sufficient statistic. Therefore, it is important to devise some criteria for choosing a "good" sufficient statistic.

A popular measure of "goodness" of a statistic $T(\mathcal{X}) \in \mathbb{R}^d$ as an estimate of $\theta \in \mathbb{R}^d$ is the *mean squared error* (MSE) between $T(\mathcal{X})$ and θ:

$$R(\theta, T) = \mathbb{E}_\theta[\|T(\mathcal{X}) - \theta\|^2]. \tag{B.4}$$

In some literature, such a function is also referred to as the "risk function," whence the capital letter R. Notice that the expression $R(\theta, T)$ can be rewritten as

$$\begin{aligned} R(\theta, T) &= \mathbb{E}_\theta[\|T(\mathcal{X}) - \mathbb{E}_\theta[T(\mathcal{X})] + \mathbb{E}_\theta[T(\mathcal{X})] - \theta\|^2] \\ &= \mathbb{E}_\theta[\|T(\mathcal{X}) - \mathbb{E}_\theta[T(\mathcal{X})]\|^2] + \|\mathbb{E}_\theta[T(\mathcal{X})] - \theta\|^2 \\ &\doteq \mathrm{Var}_\theta(T(\mathcal{X})) + \|\boldsymbol{b}_\theta(T(\mathcal{X}))\|^2, \end{aligned} \tag{B.5}$$

where $\boldsymbol{b}_\theta(T(\mathcal{X})) = \mathbb{E}_\theta[T(\mathcal{X})] - \theta$ is called the *bias* of the estimate $T(\mathcal{X})$, and $\mathrm{Var}_\theta(T(\mathcal{X})) \in \mathbb{R}$ is the trace of the covariance matrix

$$\mathrm{Cov}_\theta(T(\mathcal{X})) \doteq \mathbb{E}_\theta[(T(\mathcal{X}) - \mathbb{E}_\theta[T(\mathcal{X})])(T(\mathcal{X}) - \mathbb{E}_\theta[T(\mathcal{X})])^\top] \in \mathbb{R}^{d\times d}. \tag{B.6}$$

We refer to $\mathrm{Var}_\theta(T(\mathcal{X}))$ as the "variance" of $T(\mathcal{X})$. Thus, a good estimate is one that has both small bias and small variance.

Example B.4 For Gaussian data $\boldsymbol{x}_j \sim \mathcal{N}(\boldsymbol{\mu}, \Sigma)$, where $\boldsymbol{x}_j \in \mathbb{R}^D$, $\boldsymbol{\mu} \in \mathbb{R}^D$ and $\Sigma \in \mathbb{R}^{D\times D}$, the statistic $T(\mathcal{X}) = \frac{1}{N}\sum_{j=1}^N \boldsymbol{x}_j$ is an unbiased estimator of $\boldsymbol{\mu}$, because

$$\mathbb{E}_\theta[T(\mathcal{X})] = \frac{1}{N}\sum_{j=1}^N \mathbb{E}_\theta[\boldsymbol{x}_j] = \frac{1}{N}N\boldsymbol{\mu} = \boldsymbol{\mu}. \tag{B.7}$$

We can use the MSE to compare two estimators. We define the *relative efficiency* of two estimators T_1 and T_2 as the ratio

$$\nu_{1,2}(\theta) \doteq \frac{R(\theta, T_2)}{R(\theta, T_1)}. \tag{B.8}$$

The larger the relative efficiency ν_{12}, the smaller the MSE of T_1 relative to that of T_2. Thus, T_1 gives a more accurate, or "sharper," estimate for θ.

Notice that in general, the relative efficiency is a function of θ. Therefore, one estimator could have lower MSE for some values of θ, and another estimator could have lower MSE for other values of θ. In fact, there is no such thing as a universally

optimal estimator that gives an error smaller than that of any other estimator for all θ. For instance, if the true parameter is θ_0, the estimator $S(\mathcal{X}) = \theta_0$ achieves the smallest possible error $R(\theta, S) = 0$. Thus, the universally optimal estimate, say T, would need to have $R(\theta_0, T) = 0$, too. Since θ_0 can be arbitrary, T would need to estimate every potential parameter θ perfectly, which is impossible except for trivial cases. One can view this as a manifestation of the so-called *no free lunch theorem* known in learning theory: without any prior knowledge about θ, we can expect a statistical estimate to be better than others most of the time, but we can never expect it to be the best *all the time*. Thus, in the future, whenever we claim that some estimate is "optimal," the claim will be in the restricted sense that it is optimal within a special class of estimates considered (e.g., unbiased estimates).

In the case of unbiased estimators, the MSE reduces to the variance. Therefore, we can compare two estimators by comparing their variances. Theorem B.5 gives a lower bound on the variance of an estimator, which allows us to evaluate the efficiency of an estimator by comparing its variance to this lower bound (see Definition B.6). Before stating the theorem, we need to introduce some notation.

Assume that $p_\theta(\boldsymbol{x})$ is differentiable with respect to θ and define the *Fisher information matrix* as

$$I(\theta) \doteq \mathbb{E}_\theta\Big[\big(\frac{\partial}{\partial\theta}\log p_\theta(\mathcal{X})\big)\big(\frac{\partial}{\partial\theta}\log p_\theta(\mathcal{X})\big)^\top\Big] \in \mathbb{R}^{d\times d}. \tag{B.9}$$

Also, assume that the function $\psi(\theta) \doteq \mathbb{E}_\theta[T(\mathcal{X})] = [\psi_1(\theta), \psi_2(\theta), \ldots, \psi_d(\theta)]^\top$ is differentiable with respect to θ and define

$$\frac{\partial\psi(\theta)}{\partial\theta} \doteq \begin{bmatrix} \frac{\partial\psi_1(\theta)}{\partial\theta_1} & \frac{\partial\psi_1(\theta)}{\partial\theta_2} & \cdots & \frac{\partial\psi_1(\theta)}{\partial\theta_d} \\ \frac{\partial\psi_2(\theta)}{\partial\theta_1} & \frac{\partial\psi_2(\theta)}{\partial\theta_2} & \cdots & \frac{\partial\psi_2(\theta)}{\partial\theta_d} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial\psi_d(\theta)}{\partial\theta_1} & \frac{\partial\psi_d(\theta)}{\partial\theta_2} & \cdots & \frac{\partial\psi_d(\theta)}{\partial\theta_d} \end{bmatrix} \in \mathbb{R}^{d\times d}. \tag{B.10}$$

We have the following result.

Theorem B.5 (Cramér–Rao Lower Bound)**.** *Let $T(\mathcal{X})$ be an estimator for θ and assume that the following regularity conditions on the density p_θ and the estimator $T(\mathcal{X})$ hold:*

1. *The information matrix is well defined. That is, for all $\mathcal{X}$ such that $p_\theta(\mathcal{X}) > 0$, $\frac{\partial}{\partial\theta}\ln p_\theta(\mathcal{X})$ exists and is finite.*
2. *The operations of integration with respect to $\mathcal{X}$ and differentiation with respect to θ commute, i.e.,*

$$\frac{\partial}{\partial\theta}\int T(\mathcal{X})p_\theta(\mathcal{X})d\mathcal{X} = \int T(\mathcal{X})\frac{\partial}{\partial\theta}p_\theta(\mathcal{X})d\mathcal{X}. \tag{B.11}$$

3. *For all θ, $\psi(\theta)$ is differentiable.*

We have that for all θ,

$$Cov_\theta(T(\mathcal{X})) \geq \frac{\partial \psi(\theta)}{\partial \theta} I(\theta)^{-1} \Big(\frac{\partial \psi(\theta)}{\partial \theta} \Big)^\top, \tag{B.12}$$

where the inequality is between positive semidefinite symmetric matrices.

In the case of an unbiased estimator we have $\psi(\theta) = \theta$ and $\psi'(\theta) = I$. Therefore, the information inequality gives the following lower bound for the variance of an unbiased estimate: $\mathrm{Cov}(T(\mathcal{X})) \geq I(\theta)^{-1}$. This bound is often referred to as the *Cramér–Rao lower bound*. Since $\mathcal{X} = \{\boldsymbol{x}_j\}_{j=1}^N$ are i.i.d. samples from $p_\theta(\boldsymbol{x})$, if we define $I_1(\theta) \doteq \mathbb{E}_\theta\big[\frac{\partial}{\partial \theta} \log p_\theta(\boldsymbol{x}_1)(\frac{\partial}{\partial \theta} \log p_\theta(\boldsymbol{x}_1))^\top\big] \in \mathbb{R}^{d\times d}$, we obtain

$$I(\theta) = N I_1(\theta). \tag{B.13}$$

Thus, the Cramér–Rao lower bound can be rewritten as $\mathrm{Cov}_\theta(T(\mathcal{X})) \geq \frac{1}{N} I_1(\theta)^{-1}$.

Definition B.6 (Efficiency)**.** *We define the efficiency of an unbiased estimator $T(\mathcal{X})$ as*

$$\nu(\theta) = \frac{\mathrm{trace}\left(I^{-1}(\theta)\right)}{Var_\theta(T(\mathcal{X}))} = \frac{1}{N} \frac{\mathrm{trace}\left(I_1^{-1}(\theta)\right)}{Var_\theta(T(\mathcal{X}))}. \tag{B.14}$$

An unbiased estimator $T(\mathcal{X})$ is called efficient *if it achieves the Cramér–Rao lower bound, i.e., if $\nu(\theta) = 1$ for all θ.*

Next, we describe a procedure for finding an efficient estimator whenever possible.

B.1.3 The Rao–Blackwell Theorem and Uniformly Minimum-Variance Unbiased Estimator

To find a good estimate for θ in the MSE sense, we can resort to the Rao–Blackwell theorem. This theorem allows us to take an arbitrary estimate $S(\mathcal{X})$ of θ and produce a new estimate $S^*(\mathcal{X})$ whose MSE is at least as good as that of $S(\mathcal{X})$.

Theorem B.7 (Rao–Blackwell)**.** *If $T(\mathcal{X})$ is a sufficient statistic for θ and $S(\mathcal{X})$ is any estimate of θ, then $\tilde{S}(\mathcal{X}) = \mathbb{E}_\theta[S(\mathcal{X}) \mid T(\mathcal{X})]$ is such that*

$$\forall \theta \; R(\theta, \tilde{S}) \leq R(\theta, S). \tag{B.15}$$

The above procedure for transforming an estimator using the Rao–Blackwell theorem is often called Rao–Blackwellization. This procedure can significantly improve the estimate of θ. However, it is not guaranteed to produce an optimal estimate of θ in the MSE sense.

As we mentioned earlier, to make the estimation problem well conditioned, we must restrict the class of estimates. For instance, we may require the estimate $S(\mathcal{X})$ to be unbiased, i.e., $b_\theta(S(\mathcal{X})) = 0$. Then the problem of finding the best unbiased estimate becomes

$$\min_{S(\cdot)} R(\theta, S) = \mathrm{Var}_\theta(S(\mathcal{X})) \quad \text{s.t.} \quad \mathbb{E}_\theta[S(\mathcal{X})] = \theta. \tag{B.16}$$

The optimal $S^*(\mathcal{X})$, if it exists, is called the *uniformly minimum variance unbiased* (UMVU) estimate. In general, an unbiased estimator of θ need not exist, and so $S^*(\mathcal{X})$ is not always well defined. However, if an unbiased estimator of θ does exist, then so does $S^*(\mathcal{X})$. Moreover, if the sufficient statistic $T(\mathcal{X})$ is complete, as defined next, then $S^*(\mathcal{X})$ is unique and can be found by Rao–Blackwellization.

Definition B.8 (Complete Statistic)**.** *A statistic T is said to be complete if for every real-valued function $g(\cdot)$ such that $\mathbb{E}_\theta[g(T(\mathcal{X}))] = 0$ for all θ, we have that $p_\theta(g(T(\mathcal{X})) = 0) = 1$ for all θ.*

Starting with an unbiased estimate $S(\mathcal{X})$ and a sufficient and complete statistic $T(\mathcal{X})$, the following theorem simplifies the computation of the UMVU estimate.

Theorem B.9 (Lehmann–Scheffé)**.** *If $T(\mathcal{X})$ is a complete sufficient statistic and $S(\mathcal{X})$ is any unbiased estimate of θ, then $S^*(\mathcal{X}) = \mathbb{E}_\theta[S(\mathcal{X}) \mid T(\mathcal{X})]$ is an UMVU estimate of θ. If further, $Var_\theta(S^*(\mathcal{X})) < \infty$ for all θ, then $S^*(\mathcal{X})$ is the unique UMVU estimator.*

While the above procedure gives us an optimal unbiased estimate in the MSE sense, the UMVU estimate is often too difficult to compute in practice. Furthermore, the property of unbiasedness is not invariant under functional transformation: if $T(\mathcal{X})$ is an unbiased estimate for θ, then $g(T(\mathcal{X}))$ is in general not an unbiased estimate for $g(\theta)$. To have the functional invariant property, we often resort to the so-called maximum likelihood estimator, as described next.

B.1.4 Maximum Likelihood (ML) Estimator

Recall that the joint distribution of the N i.i.d. samples $\{\boldsymbol{x}_j\}_{j=1}^N$ has the density $p_\theta(\mathcal{X}) = \prod_{j=1}^N p_\theta(\boldsymbol{x}_j)$, and consider this density a function of θ with $\mathcal{X}$ fixed. We call this function the *likelihood function* and denote it by $L(\theta, \mathcal{X}) = p_\theta(\mathcal{X})$. The *maximum likelihood (ML) estimate* of θ, it if exists, is given by the solution to the following optimization problem:

$$\hat{\theta}_N = \arg\max_{\theta \in \Theta} \Big(L(\theta, \mathcal{X}) = p_\theta(\mathcal{X}) = \prod_{j=1}^N p_\theta(\boldsymbol{x}_j) \Big), \tag{B.17}$$

where Θ is the space of parameters. Since the logarithmic function is monotonic, we may choose to maximize the log-likelihood function instead:

$$\hat{\theta}_N = \arg\max_{\theta\in\Theta} \Big(\ell(\theta, \mathcal{X}) = \log(L(\theta, \mathcal{X})) = \sum_{j=1}^{N} \log p_\theta(\boldsymbol{x}_j)\Big), \tag{B.18}$$

which often turns out to be more convenient to use in practice. Thus, a necessary condition for the optimality of $\hat{\theta}_N$ is that

$$\frac{\partial \ell(\theta, \mathcal{X})}{\partial \theta}\Big|_{\hat{\theta}_N} = 0. \tag{B.19}$$

The ML estimate is a more popular choice than the UMVU estimate, because its existence is easier to establish, and it is usually easier to compute than the UMVU estimate. Moreover, the ML estimate is invariant under functional transformations. That is, if $\hat{\theta}_N$ is an ML estimate of θ, then $g(\hat{\theta}_N)$ is an ML estimate of $g(\theta)$. Furthermore, when the sample size is large, the ML estimate is asymptotically optimal for a wide variety of parametric models. Thus, both UMVU and ML estimates give essentially the same answer, as explained next in more detail.

B.1.5 Consistency and Asymptotic Efficiency of the ML Estimator

In general, we would like an estimate $\hat{\theta}_N$ obtained from N samples $\{\boldsymbol{x}_j\}_{j=1}^N$ to perform better and better as the number of samples increases. In this section, we characterize the asymptotic properties of an estimator. To do so, we need to make a number of technical assumptions.

Assumption B.10 *Assume that the space of parameters Θ is compact and that the density $p_\theta(\boldsymbol{x})$ is continuous and twice differentiable in θ for all $\boldsymbol{x}$ and identifiable, i.e., $p_\theta \equiv p_{\theta_0} \iff \theta = \theta_0$. Assume also that there exists a function $K(\boldsymbol{x})$ such that $\mathbb{E}_{\theta_0}[K(\boldsymbol{x})] < \infty$ and $\log p_\theta(\boldsymbol{x}) - \log p_{\theta_0}(\boldsymbol{x}) \le K(\boldsymbol{x})$ for all $\boldsymbol{x}$ and θ.*

Given these assumptions, a first approach to characterizing the asymptotic behavior of an estimator is through the notion of *consistency*.

Definition B.11 (Consistency)**.** *An estimate $\hat{\theta}_N$ of θ is said to be consistent if it converges in probability to θ ($\hat{\theta}_N \to \theta$), i.e.,*

$$\lim_{N\to\infty} P\big[\|\hat{\theta}_N - \theta\| \ge \varepsilon\big] = 0, \quad \forall \varepsilon > 0. \tag{B.20}$$

The following classical result from statistics characterizes the consistency of the ML estimator.

Proposition B.12. *Let $\{\boldsymbol{x}_j\}_{j=1}^N$ be i.i.d. samples from $p_{\theta_0}(\boldsymbol{x})$. Under the regularity assumptions in B.10, every sequence of ML estimates $\hat{\theta}_N$ converges to θ_0 in probability. In other words, every maximum likelihood estimate is consistent.*

A second approach to characterizing the asymptotic behavior of an estimator is through the notion of *asymptotic unbiasedness.*

Definition B.13 (Asymptotic Unbiasedness)**.** *Let $\boldsymbol{\mu}_N = \mathbb{E}_\theta[\hat{\theta}_N] \in \mathbb{R}^d$ and $\Sigma_N = Cov_\theta(\hat{\theta}_N) \in \mathbb{R}^{d\times d}$. We say that an estimate $\hat{\theta}_N$ of θ is asymptotically unbiased if*

$$\lim_{N\to\infty} \sqrt{N}(\boldsymbol{\mu}_N - \theta) = 0, \quad \textit{and} \quad \lim_{N\to\infty} N\Sigma_N = \Sigma > 0 \tag{B.21}$$

for some positive definite symmetric matrix $\Sigma \in \mathbb{R}^{d\times d}$.

It is easy to see that asymptotic unbiasedness is a stronger property than consistency. That is, an estimate can be consistent but asymptotically biased. In addition, most "reasonable" estimates $\hat{\theta}_N$ (e.g., the ML estimate) are often asymptotically normally distributed with mean $\boldsymbol{\mu}_N$ and covariance matrix Σ_N due to the central limit theorem. Therefore, the asymptotic distribution of an asymptotically unbiased estimate is uniquely characterized by the parameters θ and Σ.

A third way to characterize the asymptotic behavior of an estimator is through the notion of *asymptotic efficiency*. Given two asymptotically unbiased estimates, say $\hat{\theta}_N^{(1)}$ and $\hat{\theta}_N^{(2)}$, their relative *asymptotic efficiency* is defined as the ratio

$$v_{1,2}(\theta) \doteq \frac{\det(\Sigma^{(2)})}{\det(\Sigma^{(1)})}, \tag{B.22}$$

where $\Sigma^{(i)} = \lim_{N\to\infty} N\mathrm{Cov}_\theta\big(\hat{\theta}_N^{(i)}\big)$, for $i = 1, 2$. The larger the efficiency ratio $v_{1,2}$, the smaller the asymptotic variance of $\hat{\theta}^{(1)}$, relative to that of $\hat{\theta}^{(2)}$. Thus, $\hat{\theta}^{(1)}$ gives a more accurate or "sharper" estimate for θ, although both $\hat{\theta}^{(1)}$ and $\hat{\theta}^{(2)}$ are asymptotically unbiased. Nevertheless, according to Theorem B.5, an estimate cannot be arbitrarily more efficient than others. That is, for every asymptotically unbiased estimate $\hat{\theta}_N$, using (B.13) and (B.21), its covariance matrix is bounded asymptotically from below by the Cramér–Rao bound:

$$\lim_{N\to\infty} N\Sigma_N = \Sigma \geq I_1(\theta)^{-1}. \tag{B.23}$$

Definition B.14 (Asymptotic Efficiency)**.** *An estimate $\hat{\theta}_N$ is said to be asymptotically efficient if it is asymptotically normal and achieves equality in the Cramér–Rao bound* (B.23).

Asymptotic efficiency is a desirable property for an estimate, and it is sometimes referred to as asymptotic optimality. It often can be shown that UMVU estimates are asymptotically efficient. We also have the following result.

Proposition B.15. *Let $\{\boldsymbol{x}_j\}_{j=1}^N$ be i.i.d. samples from $p_{\theta_0}(\boldsymbol{x})$. Assume that the regularity conditions in B.10 hold and that the Fisher information matrix $I_1(\theta_0)$ is positive definite. Then there is a consistent sequence of ML estimators $\hat{\theta}_N$ such that $\sqrt{N}(\hat{\theta}_N - \theta_0)$ converges in distribution to $\mathcal{N}(0, I_1(\theta_0)^{-1})$. In other words, the sequence $\hat{\theta}_N$ is asymptotically unbiased and asymptotically efficient.*

Proof. We here outline the basic ideas for a "proof," which can also be used to establish for other estimates their asymptotic unbiasedness or efficiency with respect to the ML estimate. Define the function

$$\psi(\boldsymbol{x}, \theta) \doteq \frac{\partial}{\partial \theta} \log p_\theta(\boldsymbol{x}) \in \mathbb{R}^d. \tag{B.24}$$

If the maximum likelihood estimate $\hat{\theta}_N$ exists, it must satisfy the equation

$$\frac{\partial \ell(\theta, \mathcal{X})}{\partial \theta}\Big|_{\hat{\theta}_N} = \sum_{j=1}^N \psi(\boldsymbol{x}_j, \hat{\theta}_N) = 0. \tag{B.25}$$

By the mean value theorem, we have

$$\sum_{j=1}^N \psi(\boldsymbol{x}_j, \hat{\theta}_N) - \sum_{j=1}^N \psi(\boldsymbol{x}_j, \theta) = \Big[\sum_{j=1}^N \frac{\partial \psi(\boldsymbol{x}_j, \theta_N^*)}{\partial \theta}\Big]\Big(\hat{\theta}_N - \theta\Big), \tag{B.26}$$

where θ_N^* is a point between θ and $\hat{\theta}_N$. Using (B.25), we obtain

$$\sqrt{N}\big(\hat{\theta}_N - \theta\big) = \Big[\frac{1}{N}\sum_{j=1}^N \frac{\partial \psi(\boldsymbol{x}_j, \theta_N^*)}{\partial \theta}\Big]^{-1}\Big(-N^{-\frac{1}{2}}\sum_{j=1}^N \psi(\boldsymbol{x}_j, \theta)\Big). \tag{B.27}$$

Now, it follows from Proposition B.12 that $\hat{\theta}_N$ is consistent. This implies that $\lim_{N\to\infty} \frac{1}{N}\sum_{j=1}^N \frac{\partial \psi(\boldsymbol{x}_j, \theta_N^*)}{\partial \theta} = \lim_{N\to\infty} \frac{1}{N}\sum_{j=1}^N \frac{\partial \psi(\boldsymbol{x}_j, \theta)}{\partial \theta}$. By the law of large numbers, the last limit is equal to

$$\begin{aligned}
\mathbb{E}_\theta\Big[\frac{\partial \psi(\boldsymbol{x}_1, \theta)}{\partial \theta}\Big] &= \mathbb{E}_\theta\Big[\frac{\partial^2}{\partial \theta^2}\log p_\theta(\boldsymbol{x}_1)\Big] = \int \frac{\partial}{\partial \theta}\Big(\frac{\frac{\partial}{\partial \theta} p_\theta(\boldsymbol{x}_1)}{p_\theta(\boldsymbol{x}_1)}\Big) p_\theta(\boldsymbol{x}_1) \\
&= \int \frac{p_\theta(\boldsymbol{x}_1)\frac{\partial^2}{\partial \theta^2} p_\theta(\boldsymbol{x}_1) - \frac{\partial}{\partial \theta} p_\theta(\boldsymbol{x}_1)(\frac{\partial}{\partial \theta} p_\theta(\boldsymbol{x}_1))^\top}{p_\theta(\boldsymbol{x}_1)^2} p_\theta(\boldsymbol{x}_1) \\
&= \frac{\partial^2}{\partial \theta^2}\int p_\theta(\boldsymbol{x}_1) - \int \frac{\partial}{\partial \theta}\log p_\theta(\boldsymbol{x}_1)(\frac{\partial}{\partial \theta}\log p_\theta(\boldsymbol{x}_1))^\top p_\theta(\boldsymbol{x}_1) \\
&= -\mathbb{E}_\theta\Big[\frac{\partial}{\partial \theta}\log p_\theta(\boldsymbol{x}_1)\Big(\frac{\partial}{\partial \theta}\log p_\theta(\boldsymbol{x}_1)\Big)^\top\Big] = -I_1(\theta).
\end{aligned}$$

The remaining term in (B.27) involves the sum of the random vectors $\frac{\partial}{\partial\theta}\log p_\theta(\boldsymbol{x}_j)$. These vectors are i.i.d. with mean $\mathbb{E}_\theta[\frac{\partial}{\partial\theta}\log p_\theta(\boldsymbol{x}_j)] = \int \frac{\partial}{\partial\theta} p_\theta(\boldsymbol{x}_j) = 0$ and covariance $\mathbb{E}_\theta[\frac{\partial}{\partial\theta}\log p_\theta(\boldsymbol{x}_j)(\frac{\partial}{\partial\theta}\log p_\theta(\boldsymbol{x}_j))^\top] = I_1(\theta)$. Thus, by the central limit theorem, the right-hand side of (B.27) converges in distribution to $\mathcal{N}(0, I_1(\theta)^{-1})$. That is, the ML estimate is asymptotically unbiased, and its asymptotic variance reaches the Cramér–Rao lower bound. □

When the sample size is large, one can appeal to the law of large numbers to derive an information-theoretic justification for the ML estimate, which can be somewhat more revealing. Notice that maximizing the log-likelihood function is equivalent to minimizing the following objective function:

$$\min_{\theta\in\Theta}\Big(H(\theta, N) \doteq \frac{1}{N}\sum_{j=1}^{N}\big(-\log p_\theta(\boldsymbol{x}_j)\big)\Big). \tag{B.28}$$

In information theory, the quantity $-\log p_\theta(\boldsymbol{x})$ is associated with the number of bits required to represent a random event $\boldsymbol{x}$ that has the probability $p_\theta(\boldsymbol{x})$ (Cover and Thomas 1991). When the sample size N is large, due to the law of large numbers, the quantity $H(\theta, N)$ converges to

$$\lim_{N\to\infty} H(\theta, N) = H(\theta) = \mathbb{E}_{\theta_0}[-\log p_\theta(\boldsymbol{x})] = \int \big(-\log p_\theta(\boldsymbol{x})\big)p_{\theta_0}(\boldsymbol{x})\, d\boldsymbol{x}, \tag{B.29}$$

where $p_{\theta_0}(\boldsymbol{x})$ is the true distribution. Notice that the above quantity is a measure similar to the notion of "entropy": $H(\theta)$ is asymptotically the average code length of the sample set $\{\boldsymbol{x}_j\}$ when we assume that it is of the distribution $p_\theta(\boldsymbol{x})$, while $\boldsymbol{x}$ is actually drawn according to $p_{\theta_0}(\boldsymbol{x})$. Thus, the goal of ML estimation is to find the $\hat{\theta}$ that minimizes the empirical entropy of the given sample set. This is obviously a smart thing to do, since such an estimate $\hat{\theta}$ gives the most compact representation of the given sample data if an optimal coding scheme is adopted (Cover and Thomas 1991). We refer to this as the "minimum entropy principle."

Notice also that the $\hat{\theta}$ that minimizes $\int \big(-\log p_\theta(\boldsymbol{x})\big)p_{\theta_0}(\boldsymbol{x})\, d\boldsymbol{x}$ is the same as that which minimizes the so-called *Kullback–Leibler (KL) divergence* between the two distributions $p_{\theta_0}(\boldsymbol{x})$ and $p_\theta(\boldsymbol{x})$, i.e.,

$$KL\big(p_{\theta_0}(\boldsymbol{x}) \,||\, p_\theta(\boldsymbol{x})\big) \doteq \int \log\Big(\frac{p_{\theta_0}(\boldsymbol{x})}{p_\theta(\boldsymbol{x})}\Big)p_{\theta_0}(\boldsymbol{x})\, d\boldsymbol{x}. \tag{B.30}$$

One may show that under general conditions, the KL divergence is always nonnegative and becomes zero if and only if $\theta = \theta_0$. In essence, when the sample size is large, the ML objective is equivalent to minimizing the KL divergence.

However, the ML estimate is known to have very bad performance in some models even with a large number of samples. This is particularly the case when the models have many redundant parameters or the distributions are degenerate. Furthermore, both UMVU and ML estimates are not the optimal estimates in a

Bayesian[1] or minimax[2] sense. For instance, the ML estimate can be viewed as a special Bayesian estimate only when the parameter θ is uniformly distributed.

B.2 ML Estimation for Models with Latent Variables

In many practical situations, we need to estimate a statistical model in which only part of the random variables or vectors are observed, and the rest are "missing," or "hidden," or "latent," or "unobserved." For instance, suppose that two random vectors $(\boldsymbol{x}, \boldsymbol{z})$ have a joint distribution with density $p_\theta(\boldsymbol{x}, \boldsymbol{z})$, but only samples of $\boldsymbol{x}$, $\mathcal{X} = \{\boldsymbol{x}_j\}_{j=1}^N$, are observed, while the corresponding samples of $\boldsymbol{z}$, $\mathcal{Z} = \{\boldsymbol{z}_j\}_{j=1}^N$, are not available. As before, we wish to find an optimal estimate $\hat{\theta}$ for θ from the observations.

Since samples of $\boldsymbol{z}$ are not available, there is no way one can find the maximum likelihood estimate of θ from the *complete log-likelihood function*:

$$\ell_c(\theta, \mathcal{X}, \mathcal{Z}) = \sum_{j=1}^{N} \log p_\theta(\boldsymbol{x}_j, \boldsymbol{z}_j). \tag{B.31}$$

Instead, it makes sense to use the marginal distribution of $\boldsymbol{x}$, $p_\theta(\boldsymbol{x})$, and find the maximum likelihood estimate from the *incomplete log-likelihood function*[3]

$$\ell(\theta, \mathcal{X}) = \sum_{j=1}^{N} \log(p_\theta(\boldsymbol{x}_j)) = \sum_{j=1}^{N} \log\Big(\int p_\theta(\boldsymbol{x}_j, \boldsymbol{z}) d\boldsymbol{z}\Big). \tag{B.32}$$

The problem is now reduced to a standard ML estimation problem, and one can adopt any appropriate optimization method (say conjugate gradient) to find the maximum. Thus, it seems that there is no need to involve $\boldsymbol{z}$ at all.

In practice, however, there are several reasons why marginalizing over $\boldsymbol{z}$ may not be the best approach. First, for some models $p_\theta(\boldsymbol{x}, \boldsymbol{z})$, computing the marginal $p_\theta(\boldsymbol{x})$ can be intractable (e.g., summing over a combinatorial number of values for $\boldsymbol{z}$), or it can destroy good structures in the models. Second, directly maximizing $\ell(\theta, \mathcal{X})$ may turn out to be a very difficult optimization problem (e.g., high-dimensional, having many local minima). Third, in some applications, it is desirable to obtain an estimate of the unobservables $\boldsymbol{z}$ from the observables $\boldsymbol{x}$.

---

[1] A Bayesian estimate T^* is the solution to the problem $\min_T \int R(\theta, T)\pi(\theta)\, d\theta$ for a given prior distribution $\pi(\theta)$ of θ. That is, T^* is the best estimate in terms of its average risk.

[2] A minimax estimate T^* is the solution to the problem $\min_T \max_\theta R(\theta, T)$. That is, T^* is the best estimate according to its worst performance. Of course, such a T^* does not have to always exist or be easier to compute than the ML estimate.

[3] In this section, we assume that $\boldsymbol{z}$ is a continuous variable. Whenever $\boldsymbol{z}$ is discrete, we can simply replace the integrals by sums, as we will do in the next section when we cover mixture models.

B.2.1 Expectation Maximization (EM)

An alternative approach to marginalizing over the hidden variables is to take the expectation over the hidden variables. More specifically, instead of maximizing the incomplete log-likelihood $\ell(\theta, \mathcal{X})$, we can estimate the conditional density of the hidden variables given the observations $\mathcal{X}$ and an estimate θ^k for the parameters, i.e., $p_{\theta^k}(\mathcal{Z} \mid \mathcal{X})$, and maximize the expected value of the complete log-likelihood $\ell_c(\theta, \mathcal{X}, \mathcal{Z})$ with respect to the distribution $p_{\theta^k}(\mathcal{Z} \mid \mathcal{X})$.

This alternative approach has several potential advantages. First, it provides an estimate for the density of $\boldsymbol{z} \mid \boldsymbol{x}$, if needed. Second, the computation of the expected complete log-likelihood is often much simpler than the computation of the incomplete log-likelihood, as we will see. Third, the maximization of the expected log-likelihood is often much simpler than the maximization of the incomplete log-likelihood, as we will see.

In order to derive this alternative approach, let us recall the following identities:

$$\forall \boldsymbol{z} \;\; p_\theta(\boldsymbol{x}) = \frac{p_\theta(\boldsymbol{x}, \boldsymbol{z})}{p_\theta(\boldsymbol{z} \mid \boldsymbol{x})} \quad \text{and} \quad \forall \boldsymbol{x} \;\; \int p_\theta(\boldsymbol{z} \mid \boldsymbol{x})\, d\boldsymbol{z} = 1. \tag{B.33}$$

Using these identities, we can rewrite the incomplete log-likelihood as

$$\ell(\theta, \mathcal{X}) = \sum_{j=1}^{N} \log p_\theta(\boldsymbol{x}_j) = \sum_{j=1}^{N} \int p_\theta(\boldsymbol{z} \mid \boldsymbol{x}_j) \log \left(\frac{p_\theta(\boldsymbol{x}_j, \boldsymbol{z})}{p_\theta(\boldsymbol{z} \mid \boldsymbol{x}_j)} \right) d\boldsymbol{z} \tag{B.34}$$

$$= \max_{w_j} \sum_{j=1}^{N} \int w_j(\boldsymbol{z}) \log \left(\frac{p_\theta(\boldsymbol{x}_j, \boldsymbol{z})}{w_j(\boldsymbol{z})} \right) d\boldsymbol{z}, \tag{B.35}$$

where $w_j(\boldsymbol{z})$ is a density, i.e., $w_j(\boldsymbol{z}) \geq 0 \;\forall \boldsymbol{z}$ and $\int w_j(\boldsymbol{z}) d\boldsymbol{z} = 1 \;\forall j = 1, \ldots, N$. To see the last step, we use the method of Lagrange multipliers. The Lagrangian function is

$$\mathscr{L}(w_j, \lambda) = \int w_j(\boldsymbol{z}) \log \left(\frac{p_\theta(\boldsymbol{x}_j, \boldsymbol{z})}{w_j(\boldsymbol{z})} \right) d\boldsymbol{z} + \lambda \left(1 - \int w_j(\boldsymbol{z}) d\boldsymbol{z}\right). \tag{B.36}$$

Setting the variation of $\mathscr{L}$ with respect to w_j to zero, we obtain[4]

$$\frac{\partial \mathscr{L}}{\partial w_j} = \log \left(\frac{p_\theta(\boldsymbol{x}_j, \boldsymbol{z})}{w_j(\boldsymbol{z})} \right) - 1 - \lambda = 0 \implies w_j^*(\boldsymbol{z}) = p_\theta(\boldsymbol{x}_j, \boldsymbol{z}) e^{-\lambda - 1}. \tag{B.37}$$

[4]Here w_j is a function of $\boldsymbol{z}$, which is in general a continuous random variable. Therefore, we use the variation with respect to w_j in lieu of the derivative with respect to w_j. We can use the derivative, instead, whenever $\boldsymbol{z}$ is a discrete random variable.

Enforcing $\int w_j^*(z)dz = 1$, we obtain

$$w_j^*(z) = \frac{p_\theta(\boldsymbol{x}_j, z)}{\int p_\theta(\boldsymbol{x}_j, z)dz} = \frac{p_\theta(\boldsymbol{x}_j, z)}{p_\theta(\boldsymbol{x}_j)} = p_\theta(z \mid \boldsymbol{x}_j). \tag{B.38}$$

Thus, it follows from (B.34)–(B.35) that the maximization of $\ell(\theta, \mathcal{X})$ is equivalent to the following optimization problem:

$$\max_{\theta\in\Theta} \ell(\theta, \mathcal{X}) = \max_{\theta\in\Theta} \max_{\{w_j\}} \sum_{j=1}^{N} \int w_j(z) \log\left(\frac{p_\theta(\boldsymbol{x}_j, z)}{w_j(z)}\right) dz. \tag{B.39}$$

We solve the optimization problem on the right-hand side using an alternating maximization strategy (see Appendix A.1.5). Given θ, the optimal density $w_j(z)$ is given by $w_j^*(z) \doteq p_\theta(z \mid \boldsymbol{x}_j)$, which is the *a posteriori* density of z given $\boldsymbol{x}_j$ and θ. Given $w_j(z)$, the optimal parameter θ is given by

$$\begin{aligned}\theta^* &= \arg\max_{\theta\in\Theta} \sum_{j=1}^{N} \int w_j(z_j) \log p_\theta(\boldsymbol{x}_j, z_j)dz_j \qquad\qquad (B.40)\\ &= \arg\max_{\theta\in\Theta} \sum_{j=1}^{N} \mathbb{E}_{w_j}[\log(p_\theta(\boldsymbol{x}_j, z_j) \mid \boldsymbol{x}_j)] = \arg\max_{\theta\in\Theta} \mathbb{E}_w[\ell_c(\theta, \mathcal{X}, \mathcal{Z}) \mid \mathcal{X}],\end{aligned}$$

where the last expectation is taken with respect to the density $w(\mathcal{Z}) = \prod_{j=1}^{N} w_j(z_j) = p_\theta(\mathcal{Z} \mid \mathcal{X})$. Therefore, θ^* maximizes the expected complete log-likelihood taken with respect to the a posteriori density of the hidden variables given the observed ones. By alternating between these two steps, we obtain the well-known expectation maximization (EM) algorithm (Dempster et al. 1977) for maximizing the incomplete log-likelihood $\ell(\theta, \mathcal{X})$, which we summarize in Algorithm B.1.

Each iteration of this coordinate ascent algorithm does not decrease the value of the objective function in (B.39). Moreover, each iteration does not decrease the value of the incomplete log-likelihood because

$$\ell(\theta^{k+1}, \mathcal{X}) = \sum_{j=1}^{N} \int p_{\theta^{k+1}}(z \mid \boldsymbol{x}_j) \log \frac{p_{\theta^{k+1}}(\boldsymbol{x}_j, z)}{p_{\theta^{k+1}}(z \mid \boldsymbol{x}_j)} \, dz \tag{B.43}$$

$$\geq \sum_{j=1}^{N} \int p_{\theta^{k}}(z \mid \boldsymbol{x}_j) \log \frac{p_{\theta^{k+1}}(\boldsymbol{x}_j, z)}{p_{\theta^{k}}(z \mid \boldsymbol{x}_j)} \, dz \tag{B.44}$$

$$\geq \sum_{j=1}^{N} \int p_{\theta^{k}}(z \mid \boldsymbol{x}_j) \log \frac{p_{\theta^{k}}(\boldsymbol{x}_j, z)}{p_{\theta^{k}}(z \mid \boldsymbol{x}_j)} \, dz = \ell(\theta^k, \mathcal{X}). \tag{B.45}$$

Algorithm B.1 (Expectation Maximization)

Input: Data points $\{\boldsymbol{x}_j\}_{j=1}^N$ and initial parameter vector θ^0.

1: $k \leftarrow 0$.
2: **while** not converged **do**
3: **E-step:** For fixed $\theta = \theta^k$, solve for each $w_j(\boldsymbol{z})$, $j = 1, \ldots, N$, as

$$w_j^k(\boldsymbol{z}) = p_{\theta^k}(\boldsymbol{z} \mid \boldsymbol{x}_j). \tag{B.41}$$

4: **M-step:** For fixed w_j^k, solve for θ as

$$\theta^{k+1} = \arg\max_{\theta \in \Theta} \sum_{j=1}^{N} \int w_j^k(\boldsymbol{z}) \log(p_\theta(\boldsymbol{x}_j, \boldsymbol{z}))\, d\boldsymbol{z}. \tag{B.42}$$

5: **end while**
6: $k \leftarrow k + 1$.

Output: Converged parameter $\hat{\theta}$.

The first equality follows from (B.34), while the first inequality follows from (B.35) after replacing the optimal $w_j^*(\boldsymbol{z}) = p_{\theta^{k+1}}(\boldsymbol{z} \mid \boldsymbol{x}_j)$ by $p_{\theta^k}(\boldsymbol{z} \mid \boldsymbol{x}_j)$. The second inequality follows from (B.42) by replacing the optimal θ^{k+1} by θ^k, while the second equality follows from (B.34). When the cost function no longer increases, the process reaches a (local) extremum θ^* of the function $\ell(\theta, \mathcal{X})$.

The following result establishes the convergence of the EM algorithm.

Proposition B.16. *The expectation maximization process converges to one of the stationary points (extrema) of the log-likelihood function* $\ell(\theta, \mathcal{X})$.

For a more thorough exposition and complete proof of the convergence of the EM algorithm, one may refer to the paper (Wu 1983) and the book (McLanchlan and Krishnan 1997). See also Appendix A.1.5 for a discussion on the convergence of the alternating maximization approach. However, for the EM algorithm to converge to the maximum likelihood estimate (usually the global maximum) of $L(\theta, \mathcal{X})$, a good initialization is crucial.

Notice also that each step of the EM algorithm is in general a much simpler optimization problem than directly maximizing the incomplete log-likelihood $\ell(\theta, \mathcal{X})$. For many popular models (e.g., mixtures of Gaussians), one might even be able to find closed-form formulas for both steps, as shown next.

B.2.2 Maximum a Posteriori Expectation Maximization (MAP-EM)

Another alternative approach to marginalizing over the hidden variables is to take the maximum over the hidden variables. More specifically, instead of maximizing

the incomplete log-likelihood $\ell(\theta, \mathcal{X})$ with respect to θ, we maximize the complete log-likelihood $\ell_c(\theta, \mathcal{X}, \mathcal{Z})$ with respect to both θ and $\mathcal{Z}$, i.e.,

$$\max_{\theta\in\Theta} \max_{\{z_j\}} \prod_{j=1}^{N} p_\theta(\boldsymbol{x}_j, \boldsymbol{z}_j) \equiv \max_{\theta\in\Theta} \max_{\{z_j\}} \sum_{j=1}^{N} \log p_\theta(\boldsymbol{x}_j, \boldsymbol{z}_j). \tag{B.46}$$

Observe that this problem is equivalent to

$$\max_{\theta\in\Theta} \sum_{j=1}^{N} \max_{z_j} \log p_\theta(\boldsymbol{x}_j, \boldsymbol{z}_j) \equiv \max_{\theta\in\Theta} \sum_{j=1}^{N} \log p_\theta(\boldsymbol{x}_j, \hat{\boldsymbol{z}}_j), \tag{B.47}$$

where

$$\hat{\boldsymbol{z}}_j = \arg\max_{z} p_\theta(\boldsymbol{x}_j, \boldsymbol{z}) = \arg\max_{z} p_\theta(\boldsymbol{z} \mid \boldsymbol{x}_j), \tag{B.48}$$

is the maximum a posteriori (MAP) estimate of the latent variable $\boldsymbol{z}_j$ given $\boldsymbol{x}_j$. Therefore, when θ is fixed, we can solve for each $\boldsymbol{z}_j$ independently. This observation motivates us to consider an alternating maximization strategy (see Appendix A.1.5) for estimating θ. Specifically, given $\theta = \theta^k$, we solve for each hidden variable as $\hat{\boldsymbol{z}}_j^k = \arg\max_z p_{\theta^k}(\boldsymbol{z} \mid \boldsymbol{x}_j)$. Then, given $\mathcal{Z}$, we find the parameter θ that maximizes the complete log-likelihood with the hidden variables replaced by their MAP values, i.e., we estimate θ as $\hat{\theta}^{k+1} = \arg\max_\theta \sum_{j=1}^{N} \log p_\theta(\boldsymbol{x}_j, \hat{\boldsymbol{z}}_j^k)$.

For the sake of completeness, Algorithm B.2 summarizes this MAP-EM strategy. Notice that there is a clear connection with the EM algorithm: if in the EM algorithm wereplace $w_j^k(\boldsymbol{z})$ by the Dirac delta $\delta(\boldsymbol{z} - \hat{\boldsymbol{z}}_j^k)$, then the M-step of EM reduces to the

Algorithm B.2 (Maximum a Posteriori Expectation Maximization)

Input: Data points $\{\boldsymbol{x}_j\}_{j=1}^N$ and initial parameter vector θ^0.

1: $k \leftarrow 0$.

2: **while** not converged **do**

3: **MAP-step:** For fixed $\theta = \theta^k$, solve for each $\boldsymbol{z}_j$, $j = 1, \ldots, N$, as

$$\boldsymbol{z}_j^k = \arg\max_{z} p_{\theta^k}(\boldsymbol{z} \mid \boldsymbol{x}_j). \tag{B.49}$$

4: **M-step:** For fixed $\boldsymbol{z}_j^k$, solve for θ as

$$\theta^{k+1} = \arg\max_{\theta\in\Theta} \sum_{j=1}^{N} \log p_\theta(\boldsymbol{x}_j, \boldsymbol{z}_j^k). \tag{B.50}$$

5: **end while**

6: $k \leftarrow k + 1$.

Output: Converged parameter $\hat{\theta}$.

M-step of MAP-EM. Thus, we can view the MAP-EM algorithm pretty much as an EM algorithm in which the E-step is replaced by a MAP-step. This, of course, results in an approximation, and the resulting MAP-EM algorithm no longer provides an ML estimator for θ. In spite of this drawback, the MAP-EM algorithm is used as an approximate EM method, especially for mixture models, as discussed in the next section.

B.3 Estimation of Mixture Models

Mixture models are an important class of probabilistic models in which the data $\{\boldsymbol{x}_j\}_{j=1}^N$ are sampled from a distribution $p_\theta(\boldsymbol{x})$ that is a superposition of multiple distributions $\{p_{\theta_i}(\boldsymbol{x})\}_{i=1}^n$. Specifically, the mixture distribution is given by

$$p_\theta(\boldsymbol{x}) = \pi_1 p_{\theta_1}(\boldsymbol{x}) + \pi_2 p_{\theta_2}(\boldsymbol{x}) + \cdots + \pi_n p_{\theta_n}(\boldsymbol{x}), \tag{B.51}$$

where θ_i denotes the parameters of the ith distribution, $\pi_i > 0$ denotes the prior probability of drawing a point from the ith model and is such that $\sum_{i=1}^n \pi_i = 1$, and $\theta = (\theta_1, \ldots, \theta_n, \pi_1, \ldots, \pi_n)$ denotes the parameters of the mixture model. Such a distribution can be interpreted as the marginal distribution of a model with a latent random variable $z \in \{1, 2, \ldots, n\}$ that indicates the model from which $\boldsymbol{x}$ was sampled. To see this, notice that the marginal distribution can be written as

$$\begin{aligned} p_\theta(\boldsymbol{x}) &= \sum_z p_\theta(\boldsymbol{x}, z) = \sum_z p_\theta(\boldsymbol{x} \mid z) p_\theta(z) \\ &= \sum_{i=1}^n p_\theta(\boldsymbol{x} \mid z = i) p_\theta(z = i) = \sum_{i=1}^n p_{\theta_i}(\boldsymbol{x}) \pi_i, \end{aligned} \tag{B.52}$$

where $p_{\theta_i}(\boldsymbol{x}) \doteq p_\theta(\boldsymbol{x} \mid z = i)$ and $\pi_i \doteq p_\theta(z = i) > 0, i = 1, 2, \ldots, n$. The variables $\{\pi_i\}_{i=1}^n$ are often called the *mixing proportions*.

B.3.1 EM for Mixture Models

The EM algorithm is often used to estimate the parameters of a mixture model. Unlike the general EM algorithm, where the latent variable z is real-valued, in the case of a mixture model the latent variable z is discrete. Specifically, let $z_j \in \{1, \ldots, n\}$ be the latent variable associated with data point $\boldsymbol{x}_j$. In the E-step, we assume that we know the parameters $\theta^k = (\theta_1^k, \ldots, \theta_n^k, \pi_1^k, \ldots, \pi_n^k)$ of the mixture model and use them to compute the a posteriori distribution of $z_j \mid \boldsymbol{x}_j$, i.e.,

$$w_{ij}^k = p_{\theta^k}(z_j = i \mid \boldsymbol{x}_j) = \frac{p_{\theta^k}(\boldsymbol{x}_j \mid z_j = i) p_{\theta^k}(z_j = i)}{p_{\theta^k}(\boldsymbol{x}_j)} = \frac{p_{\theta_i^k}(\boldsymbol{x}_j)\pi_i^k}{\sum_{i=1}^n p_{\theta_i^k}(\boldsymbol{x}_j)\pi_i^k}. \tag{B.53}$$

In the M-step, we maximize the expected log-likelihood in (B.42),

$$\sum_{j=1}^N \sum_{i=1}^n w_{ij}^k \log(p_\theta(\boldsymbol{x}_j, z_j = i)) = \sum_{j=1}^N \sum_{i=1}^n w_{ij}^k \log(p_{\theta_i}(\boldsymbol{x}_j)\pi_i), \tag{B.54}$$

with respect to θ, and we obtain (see Exercise B.3)

$$\pi_i^{k+1} = \arg\max_{\pi_i} \sum_{j=1}^N w_{ij}^k \log(\pi_i) = \frac{\sum_{j=1}^N w_{ij}^k}{\sum_{j=1}^N \sum_{i=1}^n w_{ij}^k}, \tag{B.55}$$

$$\theta_i^{k+1} = \arg\max_{\theta_i} \sum_{j=1}^N w_{ij}^k \log(p_{\theta_i}(\boldsymbol{x}_j)). \tag{B.56}$$

Therefore, the parameters $\{\pi_i\}$ can be obtained in closed form. Whether the parameters $\{\theta_i\}$ can also be obtained in closed form will depend on the specific form of $p_{\theta_i}(\boldsymbol{x})$. Example B.17 shows that this is so for a mixture of Gaussians.

Once the model parameters are estimated from the EM algorithm, the "membership" $c_j \in \{1, 2, \ldots, n\}$ for a given sample point $\boldsymbol{x}_j$, i.e., the component distribution from which $\boldsymbol{x}_j$ is most likely drawn, can be determined by the Bayesian rule from its a posteriori probability:

$$c_j = \arg\max_{i=1,\ldots,n} p_\theta(z_j = i \mid \boldsymbol{x}_j) = \arg\max_{i=1,\ldots,n} \hat{w}_{ij}. \tag{B.57}$$

Example B.17 (EM for a Mixture of Gaussians). In the case that each mixture component is a Gaussian model with parameter $\theta_i = (\boldsymbol{\mu}_i, \Sigma_i)$, we have

$$p_{\theta_i}(\boldsymbol{x}) = \frac{1}{(2\pi)^{D/2} \det(\Sigma_i)^{1/2}} \exp\Big(- \frac{(\boldsymbol{x} - \boldsymbol{\mu}_i)^\top \Sigma_i^{-1} (\boldsymbol{x} - \boldsymbol{\mu}_i)}{2} \Big). \tag{B.58}$$

In the E-step, w_{ij}^k can be computed in closed form from (B.53) as

$$w_{ij}^k = \frac{p_{\theta_i^k}(\boldsymbol{x}_j)\pi_i^k}{\sum_{i=1}^n p_{\theta_i^k}(\boldsymbol{x}_j)\pi_i^k}. \tag{B.59}$$

Then the M-step is given by

$$\pi_i^{k+1} = \arg\max_{\pi_i} \sum_{j=1}^N w_{ij}^k \log \pi_i = \frac{\sum_{j=1}^N w_{ij}^k}{\sum_{j=1}^N \sum_{i=1}^n w_{ij}^k}, \tag{B.60}$$

$$\theta_i^{k+1} = \arg\max_{\theta_i} \sum_{j=1}^{N} w_{ij}^k \Big(-\frac{1}{2}(\boldsymbol{x}_j - \boldsymbol{\mu}_i)^\top \Sigma_i^{-1}(\boldsymbol{x}_j - \boldsymbol{\mu}_i) - \frac{1}{2}\det(\Sigma_i) \Big). \tag{B.61}$$

The above solution for the mixing proportions π_i^{k+1} follows from Exercise B.3, while the solution for $\theta_i^{k+1} = (\boldsymbol{\mu}_i^{k+1}, \Sigma_i^{k+1})$ follows from Exercise B.4 and is given by

$$\boldsymbol{\mu}_i^{k+1} = \frac{\sum_{j=1}^{N} w_{ij}^k \boldsymbol{x}_j}{\sum_{j=1}^{N} w_{ij}^k} \quad \text{and} \quad \Sigma_i^{k+1} = \frac{\sum_{j=1}^{N} w_{ij}^k (\boldsymbol{x}_j - \boldsymbol{\mu}_i^{k+1})(\boldsymbol{x}_j - \boldsymbol{\mu}_i^{k+1})^\top}{\sum_{j=1}^{N} w_{ij}^k}. \tag{B.62}$$

B.3.2 MAP-EM for Mixture Models

The EM algorithm for mixture models is based on alternating between computing the expected log-likelihood (E-step), which involves taking the expectation with respect to the latent variables, and maximizing the expected log-likelihood (M-step). As discussed in Appendix B.2.2, the MAP-EM algorithm is an alternative approach in which instead of taking the expectation, we directly maximize over the latent variables. As we will see in this section, this results in an approximate EM algorithm in which, in the E-step, each data point is assigned to the model that maximizes the posterior of the latent variables, whence the name MAP-EM.

To see this, let $z_j \in \{1, 2, \ldots, n\}$ be the latent variable denoting the model that generated $\boldsymbol{x}_j$. The MAP-EM algorithm finds the model parameters and latent variables that maximize the complete log likelihood, i.e.,

$$\max_{\theta \in \Theta} \sum_{j=1}^{N} \max_{z_j} \log p_\theta(\boldsymbol{x}_j, z_j) \equiv \max_{\theta \in \Theta} \sum_{j=1}^{N} \max_{i=1,\ldots,n} \log(p_\theta(\boldsymbol{x}_j | z_j = i)\pi_i). \tag{B.63}$$

Observe that this problem can be rewritten as[5]

$$\max_{\theta \in \Theta} \max_{\{w_{ij}\}} \sum_{j=1}^{N} \sum_{i=1}^{n} w_{ij} \log(p_{\theta_i}(\boldsymbol{x}_j)\pi_i), \tag{B.64}$$

[5]One may interpret this objective as follows. For each sample, we find the component distribution that maximizes the posterior. Once we have decided to "assign" $\boldsymbol{x}_j$ to the distribution $p_{\theta_i}(\boldsymbol{x})$, it takes $-\log p_{\theta_i}(\boldsymbol{x}_j)$ bits to encode $\boldsymbol{x}_j$. Thus, the above objective function is equivalent to minimizing the sum of the coding lengths given the membership of all the samples.

where $w_{ij} \in \{0, 1\}$ is an auxiliary variable encoding the assignment of points to models, which is defined as

$$w_{ij} = \begin{cases} 1 & \text{if } i = \arg\max_{\ell=1,\ldots,n} p_{\theta_\ell}(\boldsymbol{x}_j \mid z = \ell)\pi_\ell \\ 0 & \text{otherwise,} \end{cases} \tag{B.65}$$

and is such that for all $j = 1, \ldots, N$, $\sum_{i=1}^{n} w_{ij} = 1$.

Notice the striking connection between the *hard assignment* of points to models in (B.65) and the *soft assignment* done in the E-step of the EM algorithm for mixture models in (B.53). Notice also that when w_{ij} is fixed, the objective function in (B.64) is the same as that in the M-step of the EM algorithm for mixture models in (B.54). Thus, if we apply an alternating maximization strategy (see Appendix A.1.5) to the problem in (B.64), we obtain an algorithm that alternates between the following two steps:

MAP-step: Given θ, solve for w_{ij} such that $\sum_{i=1}^{n} w_{ij} = 1$. The optimal solution is given by (B.65) and involves assigning each data point to the model that maximizes the posterior probability, whence the name MAP-EM.

M-step: Given w_{ij}, solve for $\theta \in \Theta$. This problem is identical to the M-step in (B.54), whose solution is given by (B.55) and (B.56).

Notice that this MAP-EM algorithm for mixture models is a particular case of the MAP-EM algorithm described in Appendix B.2.2. Notice also that this MAP-EM algorithm for mixture models is very similar to the EM algorithm for mixture models, except that the *soft assignments* in the E-step in (B.53) are replaced by the *hard assignments* in (B.65). Thus, the MAP-EM algorithm is effectively an approximate EM algorithm.

Example B.18 (MAP-EM for a Mixture of Gaussians and the K-means Algorithm). In the case that each mixture component is a Gaussian model with parameter $\theta_i = (\boldsymbol{\mu}_i, \Sigma_i)$, we have

$$p_{\theta_i}(\boldsymbol{x}) = \frac{1}{(2\pi)^{D/2} \det(\Sigma_i)^{1/2}} \exp\Big(- \frac{(\boldsymbol{x} - \boldsymbol{\mu}_i)^\top \Sigma_i^{-1} (\boldsymbol{x} - \boldsymbol{\mu}_i)}{2} \Big). \tag{B.66}$$

In the E-step, given $\theta^k = (\theta_1^k, \ldots \theta_n^k, \pi_1^k, \ldots, \pi_n^k)$, w_{ij}^k can be computed in closed form as

$$w_{ij}^k = \begin{cases} 1 & \text{if } i = \arg\max_{\ell=1,\ldots,n} p_{\theta_\ell^k}(\boldsymbol{x}_j)\pi_\ell^k \\ 0 & \text{otherwise.} \end{cases} \tag{B.67}$$

Then, in the M-step, given w_{ij}^k, the mixing proportions π_i and the Gaussian parameters θ_i are given by

$$\pi_i^{k+1} = \frac{\sum_{j=1}^{N} w_{ij}^k}{\sum_{j=1}^{N}\sum_{i=1}^{n} w_{ij}^k}, \tag{B.68}$$

$$\boldsymbol{\mu}_i^{k+1} = \frac{\sum\limits_{j=1}^{N} w_{ij}^k \boldsymbol{x}_j}{\sum\limits_{j=1}^{N} w_{ij}^k} \quad \text{and} \quad \Sigma_i^{k+1} = \frac{\sum\limits_{j=1}^{N} w_{ij}^k (\boldsymbol{x}_j - \boldsymbol{\mu}_i^{k+1})(\boldsymbol{x}_j - \boldsymbol{\mu}_i^{k+1})^\top}{\sum\limits_{j=1}^{N} w_{ij}^k}. \tag{B.69}$$

Therefore, the MAP-EM algorithm alternates between assigning points to models using the MAP rule and recomputing the model parameters for each cluster.

Assume further that the mixture of Gaussians model is such that all mixing proportions are equal, i.e., $\pi_i = 1/n$ for all $i = 1, \ldots, n$, and all covariance matrices are equal to the identity matrix, i.e., $\Sigma_i = I$ for all $i = 1, \ldots, n$. In this case, the quantity $(\boldsymbol{x} - \boldsymbol{\mu}_i)^\top \Sigma_i^{-1} (\boldsymbol{x} - \boldsymbol{\mu}_i)$ reduces to the Euclidean distance $\|\boldsymbol{x} - \boldsymbol{\mu}_i\|^2$ from point $\boldsymbol{x}$ to the mean for the ith cluster $\boldsymbol{\mu}_i$. Therefore, the MAP-EM algorithm for a mixture of isotropic Gaussians with equal mixing proportions alternates between the following two steps:

MAP-step Given $\theta^k = (\boldsymbol{\mu}_1^k, \ldots \boldsymbol{\mu}_n^k)$, assign each point to its closest cluster center, i.e.,

$$w_{ij}^k = \begin{cases} 1 & \text{if } i = \arg\min\limits_{\ell=1,\ldots,n} \|\boldsymbol{x}_j - \boldsymbol{\mu}_\ell\|_2^2, \\ 0 & \text{otherwise.} \end{cases} \tag{B.70}$$

M-step Given w_{ij}^k, update each cluster center as the average of the points assigned to that cluster, i.e.,

$$\boldsymbol{\mu}_i^{k+1} = \frac{\sum\limits_{j=1}^{N} w_{ij}^k \boldsymbol{x}_j}{\sum\limits_{j=1}^{N} w_{ij}^k}. \tag{B.71}$$

This particular case of the MAP-EM algorithm gives rise to a very popular clustering algorithm called *K-means* (see (Lloyd 1957; Forgy 1965; Jancey 1966; MacQueen 1967)), where $-\log p_{\theta_i}(\boldsymbol{x})$ reduces to the simple Euclidean distance to a cluster center. This algorithm is discussed in more detail in Section 4.3.1.

B.3.3 A Case in Which EM Fails

One difficulty with the EM algorithm is that a stationary value θ^* to which the algorithm converges is not necessarily the global maximum. Furthermore,

for distributions as simple as a mixture of Gaussians, the global maximum of a likelihood function may not even exist, especially when some component distributions may become nearly singular. We illustrate this caution via the following example.

Example B.19 (ML Estimate of Two Mixed Gaussians (Vapnik 1995)). Consider a distribution $p(x)$, $x \in \mathbb{R}$, that is a mixture of two Gaussian (normal) distributions:

$$p(x, \mu, \sigma) = \frac{1}{2\sigma\sqrt{2\pi}} \exp\left\{-\frac{(x-\mu)^2}{2\sigma^2}\right\} + \frac{1}{2\sqrt{2\pi}} \exp\left\{-\frac{x^2}{2}\right\}, \tag{B.72}$$

where $\theta = (\mu, \sigma)$ are unknown. Then for given data $\mathcal{X} = \{x_1, x_2, \ldots, x_N\}$ and constant $A > 0$, there exists a small σ_0 such that for $\mu = x_1$, the log-likelihood will exceed A (regardless of the true μ, σ):

$$l(\mathcal{X}, \theta)\big|_{\mu=x_1, \sigma=\sigma_0} = \sum_{j=1}^{N} \ln p(x_i \mid \mu = x_1, \sigma = \sigma_0) \tag{B.73}$$

$$> \ln\left(\frac{1}{2\sigma_0\sqrt{2\pi}}\right) + \sum_{j=2}^{N} \ln\left(\frac{1}{2\sqrt{2\pi}} \exp\left\{-\frac{x_j^2}{2}\right\}\right) \tag{B.74}$$

$$= -\ln\sigma_0 - \sum_{j=2}^{N} \frac{x_j^2}{2} - N \ln 2\sqrt{2\pi} > A. \tag{B.75}$$

Therefore, the maximum of the log-likelihood does not even exist, and the ML objective would not provide a valid solution to estimating the unknown parameters. In fact, in this case, the true parameter corresponds to the largest (finite) local maximum of the log-likelihood.

From this simple example, we can see that the ML method does not apply to all probability densities.[6] If we insist on using it for mixtures of Gaussians, we should try to avoid the situation in which the variance can be arbitrarily small, i.e., $\sigma \to 0$. Unfortunately, this is often the case with random variables in high-dimensional spaces, where their distributions typically concentrate on low-dimensional subspaces or manifolds.

[6] It generally applies well to a class of density functions that are bounded by a common finite value from above. Hence EM would work well for generic Gaussians.

B.4 Model-Selection Criteria

So far, we have studied the following problem: given N independent samples $\{\boldsymbol{x}_j\}_{j=1}^N$ drawn from a distribution $p_\theta(\boldsymbol{x})$, where $p_\theta(\boldsymbol{x})$ belongs to a family of distributions indexed by the model parameter θ, obtain an estimate θ^* of θ. In doing so, we have assumed that the parameter $\theta \in \mathbb{R}^d$ is of fixed dimension d.

In practice, however, we may not know exactly the family of distributions to which the model belongs. Instead, we might know only that the model belongs to one of several possible families of distributions $p_{\theta(m)}(\boldsymbol{x})$, where m is a (discrete) index for the model families, $\theta(m) \in \mathbb{R}^{d(m)}$ is the vector of parameters for model family m, and $d(m)$ is the number of independent model parameters for that family. For instance, in the mixture model (B.51), the number of mixture components n could be unknown and would need to be estimated together with the mixture model parameters. In this case, for each value of n we can define a parameter vector $\theta(n) = (\theta_1, \ldots, \theta_n, \pi_1, \ldots, \pi_n)$ of dimension[7] $d(n) = nd + n - 1$. Therefore, the challenge is to choose among different models of different dimensions.

The problem of determining both the model type m and its parameter $\theta(m)$ is conventionally referred to as a *model selection* problem (as opposed to parameter estimation). Many important model-selection criteria have been developed in the statistics community and the algorithmic complexity community for general classes of models. These criteria include:

- The Akaike information criterion (AIC) (Akaike 1977) (also known as the C_p statistics (Mallows 1973)) and geometric AIC (G-AIC) (Kanatani 2003);
- The Bayesian information criterion (BIC) (also known as the Schwartz criterion); and
- Minimum description length (MDL) (Rissanen 1978) and minimum message length (MML) (Wallace and Boulton 1968).

Although these criteria were originally motivated and derived from different viewpoints (or in different contexts), they all share a common characteristic: the optimal model should be one that strikes a good *balance* between the *model complexity*, which typically depends on the dimension of the parameter space, and the *data fidelity* to the chosen model, which is typically measured as the sum of squared errors from the data points to the model. In fact, some of the criteria are essentially equivalent to each other despite their different origins. For instance, to a large extent, the AIC is equivalent to the C_p statistics, and the BIC is equivalent to the MDL.

In what follows, we give a brief review of the AIC and the BIC to illustrate the key ideas behind model selection. However, we emphasize here that in general, no model-selection criterion is always better than others under all circumstances, and the best criterion depends on the purpose of the model. For a more detailed exposition of these and many other model-selection criteria, we refer the reader to (Burnham and Anderson 2002).

[7] We subtract one parameter because $\sum_{i=1}^n \pi_i = 1$.

B.4.1 Akaike Information Criterion

Given N independent sample points $\mathcal{X} = \{\boldsymbol{x}_j\}_{j=1}^N$ drawn from a distribution $p_{\theta_0}(\boldsymbol{x})$, recall that the maximum-likelihood estimate $\hat{\theta}_N$ of the parameter θ is the one that maximizes the log-likelihood function $\ell(\theta, \mathcal{X}) = \sum_{j=1}^N \log p_\theta(\boldsymbol{x}_j)$.

The *Akaike information criterion* (AIC) for model selection is motivated from an information-theoretic viewpoint. In this approach, the quality of the obtained model is measured by the average code length used by the optimal coding scheme of $p_{\hat{\theta}_N}(\boldsymbol{x})$ for a random variable with actual distribution $p_{\theta_0}(\boldsymbol{x})$, i.e.,

$$\mathbb{E}_{\theta_0}[-\log p_{\hat{\theta}_N}(\boldsymbol{x})] = \int -\log\big(p_{\hat{\theta}_N}(\boldsymbol{x})\big) p_{\theta_0}(\boldsymbol{x})\, d\boldsymbol{x}. \tag{B.76}$$

The AIC relies on an approximation to the above expected log-likelihood loss that holds asymptotically as $N \to \infty$:

$$2\mathbb{E}_{\theta_0}[-\log p_{\hat{\theta}_N}(\boldsymbol{x})] \approx -2\ell(\hat{\theta}_N, \mathcal{X}) + 2d \doteq \mathrm{AIC}, \tag{B.77}$$

where d is the number of free parameters for the class of models of interest.

For Gaussian noise models with variance σ^2, we have

$$\ell(\hat{\theta}_N, \mathcal{X}) = -\frac{1}{2\sigma^2} \sum_{j=1}^N \|\boldsymbol{x}_j - \hat{\boldsymbol{x}}_j\|^2, \tag{B.78}$$

where $\hat{\boldsymbol{x}}_j = \mathbb{E}_{\hat{\theta}_N}[\boldsymbol{x}_j]$ is the best estimate of $\boldsymbol{x}_j$ given the model $p_{\hat{\theta}_N}(\boldsymbol{x})$. Thus, if σ^2 is known (or approximated by the empirical sample variance), minimizing the AIC is equivalent to minimizing the so-called C_p statistic:

$$C_p \doteq \frac{1}{\sigma^2} \sum_{j=1}^N \|\boldsymbol{x}_j - \hat{\boldsymbol{x}}_j\|^2 + 2d - N, \tag{B.79}$$

where the first term is obviously the mean squared error (a measure of data fidelity), and the second term is an affine function of the dimension of the parameter space (a measure of the complexity of the model).

Now consider multiple classes of models whose parameter spaces are of different dimensions and denote the dimension of model class m by $d(m)$. Then the AIC selects the model class m^* that minimizes the following objective function:

$$\mathrm{AIC}(m) = \frac{1}{\sigma^2} \sum_{j=1}^N \|\boldsymbol{x}_j - \hat{\boldsymbol{x}}_j(m)\|^2 + 2d(m), \tag{B.80}$$

where $\hat{\boldsymbol{x}}_j(m) = \mathbb{E}_{\hat{\theta}_N(m)}[\boldsymbol{x}_j]$ is the best estimate of $\boldsymbol{x}_j$ given the model $p_{\hat{\theta}_N(m)}(\boldsymbol{x})$, and $\hat{\theta}_N(m)$ is the maximum-likelihood estimate of θ for model family m.

B.4.2 Bayesian Information Criterion

The *Bayesian information criterion* (BIC) for model selection is motivated from a Bayesian inference viewpoint. In this approach, we assume a *prior* distribution of the model $p(\theta \mid m)$ and wish to choose the model class m^* that maximizes the *posterior* probability $p(m \mid \mathcal{X})$. Using the Bayesian rule, this is equivalent to maximizing

$$p(m \mid \mathcal{X}) \propto p(m)p(\mathcal{X} \mid m) = p(m) \int p(\mathcal{X} \mid \theta, m)p(\theta \mid m)\, d\theta. \tag{B.81}$$

If we assume that each model class is equally probable, this further reduces to maximizing the likelihood $p(\mathcal{X} \mid m)$ among all the model classes. This is equivalent to minimizing the negative log-likelihood $-2\log p(\mathcal{X} \mid m)$. With certain approximations, one can show that for general distributions, the following relationship holds asymptotically as $N \to \infty$:

$$\mathrm{BIC}(m) \doteq -2\log p(\mathcal{X} \mid m) = -2\ell(\hat{\theta}_N(m), \mathcal{X}) + \log(N)d(m) \tag{B.82}$$

$$= \frac{1}{\sigma^2} \sum_{j=1}^{N} \|\boldsymbol{x}_j - \hat{\boldsymbol{x}}_j(m)\|^2 + \log(N)d(m). \tag{B.83}$$

As before, $\hat{\theta}_N(m)$ is the maximum-likelihood estimate of θ given m, $d(m)$ is the number of parameters for class m, and σ^2 is the variance of a Gaussian noise model. Notice that when N and σ are known, the BIC is very similar to the AIC, except that the factor 2 in front of the second term in the AIC is replaced by $\log(N)$ in the BIC. Because we normally have $N \gg e^2$, the BIC penalizes complex models much more than the AIC does. Thus, the BIC tends to choose simpler models.

B.5 Robust Statistical Methods

For all the model-estimation and selection techniques discussed above, we have always assumed that the given data samples $\{\boldsymbol{x}_j\}_{j=1}^N$ are independent samples drawn from the same distribution $p_{\theta_0}(\boldsymbol{x})$. By an appeal to the law of large numbers, the asymptotic optimality of the estimate normally does not depend the particular set of samples given.[8] However, in many practical situations, the validity of the given

[8]The fact that almost all sets of i.i.d. samples are "typical" or "representative" of the given distribution has been at the heart of the development of Shannon's information theory.

data as independent samples of the model becomes questionable. Sometimes, the given data can be corrupted by or mixed with samples of a different (probabilistic) nature; or it can simply be the case that the given data are not a typical set of i.i.d. samples from the distribution in question. For the purpose of model estimation, these seemingly different interpretations are actually equivalent: we need to somehow infer the correct model while *accommodating* an atypical set of samples of the distribution (or the model). Obviously, this is an impossible task unless we impose some restrictions on how atypical the samples are. It is customary to assume that only a portion of the samples are different from or inconsistent with the rest of the data. Those samples are often referred to as *outliers*, and they may have a significant effect on the model inferred from data.

Unfortunately, despite centuries of interest and study,[9] there is no universally agreed definition of what an outlier is, especially for multivariate data. Roughly speaking, most definitions (or tests) for an outlier are based on one of the following guidelines:

1. The outliers are a set of samples that have relatively *large influence* on the estimated model parameters. A measure of influence is normally the difference between the model estimated with or without the sample in question.
2. The outliers are a set of *small-probability* samples with respect to the distribution in question. The given data set is therefore an atypical set if such small-probability samples constitute a significant portion of the data.
3. The outliers are a set of samples that are *not consistent* with (the model inferred from) the remainder of the data. A measure of inconsistency is normally the error residual of the sample in question with respect to the model.

Nevertheless, as we will soon see, for popular distributions such as the Gaussian, they all lead to more or less equivalent ways of detecting or accommodating outliers. However, under different conditions, different approaches that follow each of the above guidelines may give rise to solutions that can be more convenient and efficient than others.

B.5.1 Influence-Based Outlier Detection

When we try to estimate the parameter of the distribution $p_\theta(\boldsymbol{x})$ from a set of samples $\{\boldsymbol{x}_j\}_{j=1}^N$, every sample $\boldsymbol{x}_j$ might have an uneven effect on the estimated parameter $\hat{\theta}_N$. The samples that have a relatively large effect are called *influential samples*, and they can be regarded as outliers.

[9] The earliest documented discussions among astronomers about outliers or "erroneous observations" date back to the mid-eighteenth century. See (Barnett and Lewis 1983; Huber 1981; Bickel 1976) for a more thorough exposition of the studies of outliers in statistics.

To measure the influence of a particular sample $\boldsymbol{x}_j$, we may compare the difference between the parameter $\hat{\theta}_N$ estimated from all the N samples and the parameter $\hat{\theta}_N^{(j)}$ estimated from all but the jth sample. Without loss of generality, we here consider the maximum-likelihood estimate of the model:

$$\hat{\theta}_N = \arg\max_{\theta\in\Theta} \sum_{i=1}^{N} \log p_\theta(\boldsymbol{x}_i), \tag{B.84}$$

$$\hat{\theta}_N^{(j)} = \arg\max_{\theta\in\Theta} \sum_{i\neq j} \log p_\theta(\boldsymbol{x}_i), \tag{B.85}$$

and measure the influence of $\boldsymbol{x}_j$ on the estimation of θ by the difference

$$\hat{\theta}_N - \hat{\theta}_N^{(j)}. \tag{B.86}$$

Assume that $p_\theta(\boldsymbol{x})$ is analytic in θ and define the gradients of the above objective functions as

$$f(\theta) \doteq \sum_{i=1}^{N} \frac{1}{p_\theta(\boldsymbol{x}_i)} \frac{\partial p_\theta(\boldsymbol{x}_i)}{\partial \theta} \tag{B.87}$$

$$f^{(j)}(\theta) \doteq \sum_{i\neq j} \frac{1}{p_\theta(\boldsymbol{x}_i)} \frac{\partial p_\theta(\boldsymbol{x}_i)}{\partial \theta}. \tag{B.88}$$

If we now evaluate the function $f(\theta)$ at $\theta = \hat{\theta}_N^{(j)}$ using the Taylor series of $f(\theta)$ at $\theta = \hat{\theta}_N$, we obtain

$$f(\hat{\theta}_N^{(j)}) = f(\hat{\theta}_N) + f'(\hat{\theta}_N)(\hat{\theta}_N^{(j)} - \hat{\theta}_N) + o(\|\hat{\theta}_N - \hat{\theta}_N^{(j)}\|). \tag{B.89}$$

Since we have $f(\hat{\theta}_N) = 0$ and $f^{(j)}(\hat{\theta}_N^{(j)}) = 0$, the difference in the estimate caused by the jth sample is

$$\hat{\theta}_N^{(j)} - \hat{\theta}_N \approx \big(f'(\hat{\theta}_N)\big)^\dagger \Big[\frac{1}{p_{\hat{\theta}_N^{(j)}}(\boldsymbol{x}_j)} \frac{\partial p_{\hat{\theta}_N^{(j)}}(\boldsymbol{x}_j)}{\partial \theta}\Big]. \tag{B.90}$$

Notice that in the expression on the right-hand side, the factor $\big(f'(\hat{\theta}_N)\big)^\dagger$ is common for all samples.

Proposition B.20 (Approximate Sample Influence)**.** *The difference between the ML estimate $\hat{\theta}_N$ from N samples and the ML estimate $\hat{\theta}_N^{(j)}$ without the jth sample $\boldsymbol{x}_j$ depends approximately linearly on the quantity*

$$\frac{1}{p_{\hat{\theta}_N^{(j)}}(\boldsymbol{x}_j)} \frac{\partial p_{\hat{\theta}_N^{(j)}}(\boldsymbol{x}_j)}{\partial \theta}. \tag{B.91}$$

In the special case that $p_\theta(\boldsymbol{x})$ is the Gaussian distribution $\mathcal{N}(\boldsymbol{\mu}, \sigma^2 I)$ with σ^2 known, the above equation gives the influence of the jth sample on the estimate of $\boldsymbol{\mu}$:

$$\hat{\boldsymbol{\mu}}_N^{(j)} - \hat{\boldsymbol{\mu}}_N \approx \alpha(\boldsymbol{x}_j - \hat{\boldsymbol{\mu}}_N^{(j)}), \tag{B.92}$$

where α is some constant depending on σ. That is, the sample influence can be measured by the distance between the sample and the mean estimated without the sample; or equivalently, the smaller the probability of a sample with respect to the estimated (Gaussian) distribution, the larger its influence on the estimated mean. Therefore, the three guidelines for defining outliers become very much equivalent for a Gaussian distribution.

In general, to evaluate the influence of all the samples, one needs to estimate the model $N + 1$ times, which is reasonable only if each estimate is not too costly to compute. In light of this drawback, some first-order approximations of the influence values were developed in roughly the same period during which the sample influence function was proposed (Campbell 1978; Critchley 1985), when computational resources were scarcer than they are today. In robust statistics, formulas that approximate an influence function are referred to as *theoretical influence functions*.

B.5.2 Probability-Based Outlier Detection

Assume that the data are drawn from a zero-mean[10] multivariate Gaussian distribution $\mathcal{N}(\mathbf{0}, \Sigma_{\boldsymbol{x}})$. If there were no outliers, the maximum likelihood estimate of $\Sigma_{\boldsymbol{x}}$ would be given by $\hat{\Sigma}_N = \frac{1}{N}\sum_{j=1}^N \boldsymbol{x}_j \boldsymbol{x}_j^\top \in \mathbb{R}^{D\times D}$. Therefore, we could approximate the probability that a sample $\boldsymbol{x}_j$ comes from this Gaussian model by

$$p(\boldsymbol{x}_j; \hat{\Sigma}_N) = \frac{1}{(2\pi)^{D/2} \det(\hat{\Sigma}_N)^{1/2}} \exp\big(-\frac{1}{2}\boldsymbol{x}_j^\top \hat{\Sigma}_N^{-1} \boldsymbol{x}_j\big). \tag{B.93}$$

If we adopt the guideline that outliers are samples that have a small probability with respect to the estimated model, then the outliers are exactly those samples that have a relatively large residual:

[10] We here are interested only in how to robustly estimate the covariance, or "scale," of the distribution. In case the mean, or "location," of the distribution is not known, a separate robust procedure can be employed to determine the mean before the covariance; see (Barnett and Lewis 1983).

$$\varepsilon_j = \boldsymbol{x}_j^\top \hat{\Sigma}_N^{-1} \boldsymbol{x}_j, \quad j = 1, 2, \ldots, N, \tag{B.94}$$

also known as the *Mahalanobis distance*.[11]

In principle, we could use $p(\boldsymbol{x}_j, \hat{\Sigma}_N)$ or ε_j to determine whether $\boldsymbol{x}_j$ is an outlier. However, the above estimate of the covariance matrix $\Sigma_{\boldsymbol{x}}$ is obtained using all the samples, including the outliers themselves. Therefore, if $\hat{\Sigma}_N$ is very different from $\Sigma_{\boldsymbol{x}}$, the outliers could be incorrectly detected. In order to improve the estimate of $\Sigma_{\boldsymbol{x}}$, one can recompute $\hat{\Sigma}_N$ by discarding or downweighting samples that have low probability or large Mahalanobis distance. Let $w_j \in [0, 1]$ be a weight assigned to the jth point such that $w_j \approx 1$ if $\boldsymbol{x}_j$ is an inlier and $w_j \approx 0$ if $\boldsymbol{x}_j$ is an outlier. Then a new estimate of $\Sigma_{\boldsymbol{x}}$ can be obtained as

$$\hat{\Sigma}_N = \frac{\sum_{j=1}^N w_j \boldsymbol{x}_j \boldsymbol{x}_j^\top}{\sum_{j=1}^N w_j}. \tag{B.95}$$

Maximum-Likelihood-Type Estimators (M-Estimators)
If we choose $w(\varepsilon) \equiv \varepsilon$, the above expression gives the original estimate of the covariance matrix $\hat{\Sigma}_N = \frac{1}{N} \sum_{j=1}^N \boldsymbol{x}_j \boldsymbol{x}_j^\top$. Alternatively, if we simply want to discard all samples with a Mahalanobis distance larger than a certain threshold $\varepsilon_0 > 0$, we can choose the following weight function:

$$w(\varepsilon) = \begin{cases} \varepsilon, & \text{for } \varepsilon \le \varepsilon_0, \\ 0, & \text{for } \varepsilon > \varepsilon_0. \end{cases} \tag{B.96}$$

Nevertheless, under the assumption that the distribution is elliptically symmetric and is contaminated by an associated normal distribution, the following weight function gives a more robust estimate of the covariance matrix (Hampel 1974; Campbell 1980):

$$w(\varepsilon) = \begin{cases} \varepsilon, & \text{for } \varepsilon \le \varepsilon_0, \\ \varepsilon_0 \exp[-\frac{1}{2a}(\varepsilon - \varepsilon_0)^2] & \text{for } \varepsilon > \varepsilon_0, \end{cases} \tag{B.97}$$

with $\varepsilon_0 = \sqrt{D + b}$ for some suitable choice of positive values for a and b, and D denotes the dimension of the space. Many other weight functions have also been proposed in the statistics literature. They serve as the basis for a class of robust estimators, known as *M-estimators* (maximum-likelihood-type estimators) (Huber 1981; Barnett and Lewis 1983). Nevertheless, most M-estimators differ only in how the samples are downweighted, but no one of them seems to dominate the others in terms of performance in all circumstances.

[11] In fact, it can be shown (Ferguson 1961) that if the outliers have a Gaussian distribution of a different covariance matrix $a\Sigma$, then ε_j is a sufficient statistic for the test that maximizes the probability of correct decision about the outlier (in the class of tests that are invariant under linear transformations). The interested reader may want to find out how this distance is equivalent (or related) to the sample influence $\hat{\Sigma}_N^{(j)} - \hat{\Sigma}_N$ or the approximate sample influence given in (B.91).

Notice that calculating the robust estimate $\hat{\Sigma}_N$ as in (B.95) is not easy, because the weights w_j also depend on the resulting $\hat{\Sigma}_N$. There is no surprise that many known algorithms are based on Monte Carlo (Maronna 1976; Campbell 1980).

Multivariate Trimming (MVT)
One drawback of the M-estimators is that their "breakdown point" is inversely proportional to the dimension of the data space. The breakdown point is an important measure of robustness of any estimator. Roughly speaking, it is the largest proportion of contamination that the estimator can tolerate. Thus, the M-estimators become much less robust when the dimension of the data is high.

One way to resolve this problem is to modify the M-estimators by simply trimming out a percentage of the samples with relatively large Mahalanobis distance and then using the remaining samples to reestimate the covariance matrix. Then each time we have a new estimate of the covariance matrix, we can recalculate the Mahalanobis distance of every sample and reselect samples that need to be trimmed. We can repeat the above process until a stable estimate of the covariance matrix is obtained. This iterative scheme is known as *multivariate trimming* (MVT), another popular robust estimator. By construction, the breakdown point of MVT does not depend on the dimension of the problem and depends only on the chosen trimming percentage.

When the percentage of outliers is somehow known, it is relatively easy to determine how many samples need to be trimmed, and it usually takes only a few iterations for MTV to converge. However, if the percentage is wrongfully specified, MVT is known to have trouble converging, or it may converge to a wrong estimate of the covariance matrix.

B.5.3 Random-Sampling-Based Outlier Detection

When the outliers constitute a large portion (up to 50% or even more) of the data set, the (ML) estimate $\hat{\theta}_N$ obtained from all the samples can be so severely corrupted that the sample influence and the Mahalanobis distance computed based on it become useless in discriminating between outliers and valid samples.[12] This motivates estimating the model parameter θ using only a (randomly sampled) small subset of the samples to begin with. In this section, we describe two such methods: least median of squares (LMS) and random sample consensus (RANSAC).

[12]Thus, the iterative process is likely to converge to a local minimum other than the true model parameter. Sometimes, it can even be the case that the roles of inliers and outliers are exchanged with respect to the converged estimate.

Least Median Estimation
If we knew that fewer than half of the samples are potential outliers, we could use only half of the samples to estimate the model parameter. But which half of the samples should we use? We know that the maximum-likelihood estimate minimizes the sum of negative log-likelihoods:

$$\hat{\theta}_N = \arg\min_{\theta \in \Theta} \sum_{j=1}^{N} \Big(- \log(p_\theta(\boldsymbol{x}_j)) \Big). \tag{B.98}$$

Since outliers should have small probability, hence large negative log-likelihood, we can order the values of the negative log-likelihood and eliminate from the above objective half of the samples that have relatively larger values:

$$\hat{\theta}_{N/2} = \arg\min_{\theta \in \Theta} \sum_{j} \big(- \log(p_\theta(\boldsymbol{x}_j)) \big), \tag{B.99}$$

where the sum is over the points $\boldsymbol{x}_j$ such that

$$- \log(p_\theta(\boldsymbol{x}_j)) \leq \operatorname*{median}_{\boldsymbol{x}_\ell \in \mathcal{X}} \big(- \log(p_\theta(\boldsymbol{x}_\ell)) \big). \tag{B.100}$$

A popular approximation to the above objective is simply to minimize the median value of the negative log-likelihood:

$$\hat{\theta}_M \doteq \arg\min_{\theta \in \Theta} \operatorname*{median}_{\boldsymbol{x}_j \in \mathcal{X}} \big(- \log_\theta(p(\boldsymbol{x}_j)) \big). \tag{B.101}$$

We call $\hat{\theta}_M$ the *least median estimate*. In the case of a Gaussian noise model, $-\log p(\boldsymbol{x}_j, \theta)$ is proportional to the squared error:

$$- \log(p_\theta(\boldsymbol{x}_j)) \propto \|\boldsymbol{x}_j - \hat{\boldsymbol{x}}_j\|^2. \tag{B.102}$$

For this reason, the estimate $\hat{\theta}_M$ is often known as the *least median of squares* (LMS) estimate.[13]

However, without knowing θ, it is impossible to order the log-likelihoods or the squared errors, let alone compute the median. A typical method to resolve this difficulty is to *randomly sample* a number of small subsets of the data:

$$\mathcal{X}_1, \mathcal{X}_2, \ldots, \mathcal{X}_m \subset \mathcal{X}, \tag{B.103}$$

[13] The importance of the median for robust estimation was pointed out first in the article (Hampel 1974).

where each subset $\mathcal{X}_i$ is independently drawn and contains $k \ll N$ samples. If p is the fraction of valid samples (the "inliers"), one can show (see Exercise B.8) that with probability $q = 1 - (1 - p^k)^m$, one of the above subsets will contain only valid samples. In other words, if q is the probability that one of the selected subsets contains only valid samples, we need to randomly sample at least

$$m \geq \frac{\log(1-q)}{\log(1-p^k)} \tag{B.104}$$

subsets of k samples.

Using each subset $\mathcal{X}_i$, we can compute an estimate $\hat{\theta}_i$ of the model and use the estimate to compute the median for the remaining $N - k$ samples in $\mathcal{X} \setminus \mathcal{X}_i$:

$$\hat{M}_i \doteq \underset{\boldsymbol{x}_j \in \mathcal{X} \setminus \mathcal{X}_i}{\text{median}} \big(-\log(p_{\hat{\theta}_i}(\boldsymbol{x}_j)) \big). \tag{B.105}$$

Then the least median estimate $\hat{\theta}_M$ is approximated by the $\hat{\theta}_{i^*}$ that gives the smallest median $\hat{M}_{i^*} = \min_i \hat{M}_i$.

In the case of a Gaussian noise model, based on the order statistics of squared errors, we can use the median statistic to obtain an (asymptotically unbiased) estimate of the variance, or scale, of the error as follows:

$$\hat{\sigma} = \frac{N+5}{N\Phi^{-1}(0.5 + p/2)} \sqrt{\underset{\boldsymbol{x}_j \in \mathcal{X}}{\text{median}} \|\boldsymbol{x}_j - \hat{\boldsymbol{x}}_j\|^2}, \tag{B.106}$$

where $p = 0.5$ for the median statistic. Then one can use $\hat{\sigma}$ to find "good" samples in $\mathcal{X}$ whose squared errors are less than $\lambda\sigma^2$ for some chosen constant λ (normally less than 5). Using such good samples, we can recompute a more efficient (ML) estimate $\hat{\theta}$ of the model.

Random Sample Consensus (RANSAC)

In theory, the breakdown point of the least median estimate is up to 50% outliers. In many practical situations, however, outlying samples may constitute more than half of the data. Random sample consensus (RANSAC) (Fischler and Bolles 1981) is a method that is designed to work for such highly contaminated data.

In many respects, RANSAC is actually very similar to LMS. The main difference is that instead of looking at the median statistic,[14] RANSAC tries to find, among all the estimates $\{\hat{\theta}_i\}$ obtained from the subsets $\{\mathcal{X}_i\}$, the one that maximizes the number of samples that have an error residual (measured either by the negative log-likelihood or the squared error) smaller than a prespecified error tolerance:

[14] Which becomes meaningless when the fraction of outliers is over 50%.

$$\hat{\theta}_{i*} \doteq \arg\max_{\hat{\theta}_i} \#\{\boldsymbol{x}_j \in \mathcal{X} : -\log(\boldsymbol{x}_j, \hat{\theta}_i) \leq \tau\}. \tag{B.107}$$

In other words, $\hat{\theta}_{i*}$ achieves the highest "consensus" among all the random sample estimates $\{\hat{\theta}_i\}$, whence the name "random sample consensus" (RANSAC). To improve the efficiency of the estimate, we can recompute an ML estimate $\hat{\theta}$ of the model from all the samples that are consistent with $\hat{\theta}_{i*}$.

Notice that for RANSAC, one needs to specify the error tolerance τ a priori. In other words, RANSAC requires knowing the variance σ^2 of the error a priori, while LMS normally does not. There have been a few variations of RANSAC in the literature that relax this requirement. We here do not elaborate on them, and interested readers may refer to (Steward 1999) and references therein.

However, when the dimension of the model is large or the model has a large number of mixture components, random sampling techniques have not been very effective. The reason is largely that in this case, the number of subsets needed in (B.104) grows prohibitively large. The reader may refer to (Yang et al. 2006) for an empirical study that extends RANSAC-type ideas to the case of a mixture of subspaces.

B.6 Exercises

Exercise B.1 (ML Estimates of the Parameters of a Gaussian) Let $\boldsymbol{x} \in \mathbb{R}^D$ be a random vector with distribution $\mathcal{N}(\boldsymbol{\mu}_x, \Sigma_x)$, where $\boldsymbol{\mu}_x = \mathbb{E}(\boldsymbol{x}) \in \mathbb{R}^D$ and $\Sigma_x = \mathbb{E}(\boldsymbol{x} - \boldsymbol{\mu})(\boldsymbol{x} - \boldsymbol{\mu})^\top \in \mathbb{R}^{D \times D}$ are, respectively, the mean and the covariance of $\boldsymbol{x}$. Show that the maximum likelihood estimates of $\boldsymbol{\mu}_x$ and Σ_x are, respectively, given by

$$\hat{\boldsymbol{\mu}}_N \doteq \frac{1}{N}\sum_{j=1}^{N} \boldsymbol{x}_j \quad \text{and} \quad \hat{\Sigma}_N \doteq \frac{1}{N}\sum_{j=1}^{N} (\boldsymbol{x}_j - \hat{\boldsymbol{\mu}}_N)(\boldsymbol{x}_j - \hat{\boldsymbol{\mu}}_N)^\top. \tag{B.108}$$

Exercise B.2 (Invariance of ML Estimator) Let $\hat{\theta}_N$ be the maximum likelihood (ML) estimate of θ obtained from N i.i.d. samples $\{\boldsymbol{x}_j\}_{j=1}^N$ from the distribution $p_\theta(\boldsymbol{x})$. Show that $g(\hat{\theta}_N)$ is an ML estimate of $g(\theta)$. What are the conditions that need to be imposed on $g(\theta)$ in order for $g(\hat{\theta}_N)$ to be an ML estimate of $g(\theta)$?

Exercise B.3 (ML Estimates of the Mixing Proportions) Let $W = [w_{ij}] \in \mathbb{R}^{n \times N}$ be a left stochastic matrix, i.e., $w_{ij} \geq 0$ and $\sum_{i=1}^n w_{ij} = 1$ for all $j = 1, \ldots, N$. Let π be a stochastic vector, i.e., $\pi \in \Pi = \{(\pi_1, \ldots, \pi_n) : \pi_i \geq 0$, and $\sum_{i=1}^n \pi_i = 1\}$. Show that

$$\arg\max_{\pi\in\Pi} \sum_{j=1}^{N} w_{ij} \log(\pi_i) = \frac{\sum_{j=1}^{N} w_{ij}}{\sum_{i=1}^{n}\sum_{j=1}^{N} w_{ij}}. \tag{B.109}$$

Exercise B.4 (ML Estimates of the Parameters of a Mixture of Gaussians) Let $W = [w_{ij}] \in \mathbb{R}^{n\times N}$ be a left stochastic matrix, i.e., $w_{ij} \geq 0$ and $\sum_{i=1}^{n} w_{ij} = 1$ for all $j = 1, \ldots, N$. Show that the solution to the optimization problem

$$\max_{\boldsymbol{\mu}_i, \Sigma_i} \sum_{j=1}^{N} w_{ij} \Big(-\frac{1}{2}(\boldsymbol{x}_j - \boldsymbol{\mu}_i)^\top \Sigma_i^{-1} (\boldsymbol{x}_j - \boldsymbol{\mu}_i) - \frac{1}{2}\det(\Sigma_i) \Big) \tag{B.110}$$

is

$$\boldsymbol{\mu}_i = \frac{\sum_{j=1}^{N} w_{ij}\boldsymbol{x}_j}{\sum_{j=1}^{N} w_{ij}} \quad \text{and} \quad \Sigma_i = \frac{\sum_{j=1}^{N} w_{ij}(\boldsymbol{x}_j - \boldsymbol{\mu}_i)(\boldsymbol{x}_j - \boldsymbol{\mu}_i)^\top}{\sum_{j=1}^{N} w_{ij}}. \tag{B.111}$$

Exercise B.5 Study MATLAB's `gmdistribution` class, which is described at http://www.mathworks.com/help/stats/gmdistribution-class.html, and reproduce the example on clustering using Gaussian mixture models described at http://www.mathworks.com/help/stats/gaussian-mixture-models.html. That is, use the function `mvnrnd` to generate data sampled from a mixture of two Gaussians and the function `fitgmdist` to estimate the parameters of the mixture model from the data. Then plot the isocontours of the estimated distribution, the clustering of the data, and the soft assignment weights.

Exercise B.6 Reproduce three of the examples described at http://www.mathworks.com/help/stats/fitgmdist.html. Specifically, reproduce the examples entitled *Cluster Data Using a Gaussian Mixture Model*, *Regularize Gaussian Mixture Model Estimation*, and *Determine the Best Gaussian Mixture Fit Using AIC*. In addition, add a new example called *Determine the Best Gaussian Mixture Fit Using BIC* and compare it to AIC.

Exercise B.7 Implement the EM and MAP-EM algorithms for a mixture of Gaussians. The format of your function should be as follows.

Function `[group,mu,Sigma,pi]=GMM(x,n,method,group0, restarts)`

Parameters

`x`	$D \times N$ matrix whose columns are the data points
`n`	number of groups
`method`	'EM', 'MAPEM'
`group0`	$1 \times N$ vector containing an initial soft or hard assignment of points to groups

Returned values

`group`	$1 \times N$ vector containing the soft or hard assignments of points to groups
`mu`	$D \times n$ matrix whose ith column is the mean for the ith group
`Sigma`	$D \times D \times n$ tensor whose ith slice is the covariance matrix of the ith group
`pi`	$n \times 1$ vector whose entries are the mixing proportions

Estimates the parameters of a Gaussian mixture model

Generate data from a mixture of two Gaussians in $\mathbb{R}^2$ with means $(-1, -1)$ and $(1, 1)$, equal covariance matrices $\sigma^2 I$, and equal mixing proportions $\pi_1 = \pi_2 = 1/2$. Increase σ from 0.1 to 1 and plot the clustering error as a function of σ. Plot also the error in the estimated parameters as a function of σ. Compare your results with those produced by the MATLAB function `fitgmdist`.

Exercise B.8 (RANSAC) Suppose you are given N data points such that $p\%$ are inliers and $(1 - p)\%$ are outliers. Suppose you wish to fit a model to the inliers and that $k \ll N$ is the minimum number of points needed to estimate the model.

1. Suppose that you sample k out of N data points with replacement. What is the probability that all k points are inliers?
2. Suppose that not all k points are inliers, and so you keep sampling k points m times. Show that the probability that after m trials all k points are inliers for the first time is $1 - (1 - p^k)^m$.
3. Show that the number of trials needed so that the probability that all k points are inliers is at least q is given by

$$m \geq \frac{\log(1 - q)}{\log(1 - p^k)}. \tag{B.112}$$

Appendix C
Basic Facts from Algebraic Geometry

Algebra is but written geometry; geometry is but drawn algebra.

—Sophie Germain

A centuries-old practice in science and engineering it to fit polynomials to a given set of data points. In this book, we often use the set of zeros of (multivariate) polynomials to model a given data set. In mathematics, polynomials and their zero sets are studied in algebraic geometry, with Hilbert's Nullstellensatz establishing the basic link between algebra (polynomials) and geometry (the zero set of polynomials, a geometric object). In order to make this book self-contained, we review in this appendix some of the basic algebraic notions and facts that are used in this book, especially in Chapter 5. In particular, we will introduce the special algebraic properties of multiple subspaces as algebraic sets. For a more systematic introduction to abstract algebra and algebraic geometry, the reader may refer to the classic texts of Lang (Lang 1993) and Eisenbud (Eisenbud 1996).

C.1 Abstract Algebra Basics

C.1.1 Polynomial Rings

Consider a D-dimensional vector space over a field R (of characteristic 0), denoted by R^D, where R is usually the field of real numbers $\mathbb{R}$ or the field of complex numbers $\mathbb{C}$.

Let $R[\boldsymbol{x}] = [x_1, x_2, \ldots, x_D]$ be the set of all polynomials in D variables $x_1, x_2, \ldots, x_D$. Then $R[\boldsymbol{x}]$ is a *commutative ring* with two basic operations: "summation" and "multiplication" of polynomials. The elements of R are called *scalars* or *constants*. A *monomial* is a product of the variables; its degree is the number

R. Vidal et al., *Generalized Principal Component Analysis*, Interdisciplinary Applied Mathematics 40, DOI 10.1007/978-0-387-87811-9

of the variables (counting repeats). A monomial of degree n is of the form $\boldsymbol{x}^{\boldsymbol{n}} = x_1^{n_1} x_2^{n_2} \cdots x_D^{n_D}$ with $0 \leq n_j \leq n$ and $n_1 + n_2 + \cdots + n_D = n$. Altogether, there are

$$M_n(D) \doteq \binom{D+n-1}{n} = \binom{D+n-1}{D-1}$$

different degree-n monomials.

Definition C.1 (Veronese Map)**.** *For given n and D, the* Veronese map *of degree n, denoted by* $\nu_n : R^D \to R^{M_n(D)}$*, is defined as*

$$\nu_n : \ [x_1, \ldots, x_D]^\top \mapsto [\ldots, \boldsymbol{x}^{\boldsymbol{n}}, \ldots]^\top, \tag{C.1}$$

where $\boldsymbol{x}^{\boldsymbol{n}}$ *are degree-n monomials of the form* $x_1^{n_1} x_2^{n_2} \cdots x_D^{n_D}$ *with* $\boldsymbol{n} = (n_1, n_2, \ldots, n_D)$ *chosen in the degree-lexicographic order.*

Example C.2 (The Veronese Map of Degree 2 in Three Variables). If $\boldsymbol{x} = [x_1, x_2, x_3]^\top \in R^3$, the Veronese map of degree 2 is given by

$$\nu_2(\boldsymbol{x}) = [x_1^2, x_1 x_2, x_1 x_3, x_2^2, x_2 x_3, x_3^2]^\top \in R^6.$$

In the context of kernel methods (Chapter 4), the Veronese map is usually referred to as the polynomial embedding, and the ambient space $R^{M_n(D)}$ is called the *feature* space.

A *term* is a scalar multiplying a monomial. A polynomial $p(\boldsymbol{x})$ is said to be *homogeneous* if all its terms have the same degree. Sometimes, the word *form* is used to mean a homogeneous polynomial. Every homogeneous polynomial $p(\boldsymbol{x})$ of degree n can be written as

$$p(\boldsymbol{x}) = \boldsymbol{c}_n^\top \nu_n(\boldsymbol{x}) = \sum c_{n_1,\ldots,n_D} x_1^{n_1} \cdots x_D^{n_D}, \tag{C.2}$$

where $c_{n_1,\ldots,n_D} \in R$ are the coefficients associated with the monomials $\boldsymbol{x}^{\boldsymbol{n}} = x_1^{n_1} \cdots x_D^{n_D}$.

In this book, we are primarily interested in the *algebra* of homogeneous polynomials in D variables.[1] Because of that, we view R^D as a projective space—the set of one-dimensional subspaces (meaning lines through the origin). Every one-dimensional subspace, say a line L, can be represented by a point $[a_1, a_2, \ldots, a_D]^\top \neq [0, 0, \ldots, 0]^\top$ on the line. The result is a projective $(D-1)$-space over R that can be regarded as the D-tuples $[a_1, a_2, \ldots, a_D]^\top$ of elements of R, modulo the equivalence relation $[a_1, a_2, \ldots, a_D]^\top \sim [ba_1, ba_2, \ldots, ba_D]^\top$ for all $b \neq 0$ in R.

If $p(x_1, x_2, \ldots, x_D)$ is a homogeneous polynomial of degree n, then for $b \in R$, we have

$$p(ba_1, ba_2, \ldots, ba_D) = b^n p(a_1, a_2, \ldots, a_D). \tag{C.3}$$

[1] For algebra of polynomials defined on R^D as an affine space, the reader may refer to (Lang 1993).

Therefore, whether $p(a_1, a_2, \ldots, a_D) = 0$ on a line L does not depend on the representative point chosen on the line L.

We may view $R[\boldsymbol{x}]$ as a *graded ring*, which can be decomposed as

$$R[\boldsymbol{x}] = \bigoplus_{i=0}^{\infty} R_i = R_0 \oplus R_1 \oplus \cdots \oplus R_n \oplus \cdots, \tag{C.4}$$

where R_i consists of all polynomials of degree i. In particular, $R_0 = R$ is the set of nonzero scalars (or constants). It is convention (and convenient) to define the degree of the zero element 0 in R to be infinite or -1. The set R_1 consists of all homogeneous polynomials of degree one, i.e., the set of 1-forms,

$$R_1 \doteq \left\{ b_1 x_1 + b_2 x_2 + \cdots + b_D x_D : \ [b_1, b_2, \ldots, b_D]^\top \in R^D \right\}. \tag{C.5}$$

Obviously, the dimension of R_1 as a vector space is also D; R_1 can also be viewed as the dual space $(R^D)^*$ of R^D. For convenience, we also define the following two sets:

$$R_{\leq m} \doteq \bigoplus_{i=0}^{m} R_i = R_0 \oplus R_1 \oplus \cdots \oplus R_m,$$

$$R_{\geq m} \doteq \bigoplus_{i=m}^{\infty} R_i = R_m \oplus R_{m+1} \oplus \cdots,$$

which are the set of polynomials of degree less than or equal to m and those of degree greater than or equal to m, respectively.

C.1.2 Ideals and Algebraic Sets

Definition C.3 (Ideal)**.** *An* ideal *in the (commutative) polynomial ring $R[\boldsymbol{x}]$ is an additive subgroup I (with respect to the summation of polynomials) such that if $p(\boldsymbol{x}) \in I$ and $q(\boldsymbol{x}) \in R[\boldsymbol{x}]$, then $p(\boldsymbol{x})q(\boldsymbol{x}) \in I$.*

From the definition, one can verify that if I, J are two ideals of $R[\boldsymbol{x}]$, their intersection $K = I \cap J$ is also an ideal. The previously defined set $R_{\geq m}$ is an ideal for every m. In particular, $R_{\geq 1}$ is the so-called *irrelevant ideal*, sometimes denoted by R_+.

An ideal is said to be *generated* by a subset $\mathcal{G} \subset I$ if every element $p(\boldsymbol{x}) \in I$ can be written in the form

$$p(\boldsymbol{x}) = \sum_{i=1}^{k} q_i(\boldsymbol{x}) g_i(\boldsymbol{x}), \quad \text{with } \ q_i(\boldsymbol{x}) \in R[\boldsymbol{x}] \ \text{ and } \ g_i(\boldsymbol{x}) \in \mathcal{G}. \tag{C.6}$$

We write $(\mathcal{G})$ for the ideal generated by a subset $\mathcal{G} \subset R[\boldsymbol{x}]$; if $\mathcal{G}$ contains only a finite number of elements $\{g_1, \ldots, g_k\}$, we usually write $(g_1, \ldots, g_k)$ in place of $(\mathcal{G})$. An ideal I is *principal* if it can be generated by one element (i.e., $I = p(\boldsymbol{x})R[\boldsymbol{x}]$ for some polynomial $p(\boldsymbol{x})$). Given two ideals I and J, the ideal that is generated by the product of elements in I and J,

$$\{f(\boldsymbol{x})g(\boldsymbol{x}),\ f(\boldsymbol{x}) \in I, g(\boldsymbol{x}) \in J\}$$

is called the *product ideal*, denoted by IJ.

An ideal I of the polynomial ring $R[\boldsymbol{x}]$ is *prime* if $I \neq R[\boldsymbol{x}]$ and if $p(\boldsymbol{x}), q(\boldsymbol{x}) \in R[\boldsymbol{x}]$ and $p(\boldsymbol{x})q(\boldsymbol{x}) \in I$ implies that $p(\boldsymbol{x}) \in I$ or $q(\boldsymbol{x}) \in I$. If I is prime, then for any ideals J, K with $JK \subseteq I$, we have $J \subseteq I$ or $K \subseteq I$.

A polynomial $p(\boldsymbol{x})$ is said to be *prime* or irreducible if $p(\boldsymbol{x})$ generates a prime ideal. Equivalently, $p(\boldsymbol{x})$ is irreducible if $p(\boldsymbol{x})$ is not a nonzero scalar and whenever $p(\boldsymbol{x}) = f(\boldsymbol{x})g(\boldsymbol{x})$, then one of $f(\boldsymbol{x})$ and $g(\boldsymbol{x})$ is a nonzero scalar.

Definition C.4 (Homogeneous Ideal)**.** *A homogeneous ideal of $R[\boldsymbol{x}]$ is an ideal that is generated by homogeneous polynomials.*

Note that the sum of two homogeneous polynomials of different degrees is no longer a homogeneous polynomial. Thus, a homogeneous ideal contains inhomogeneous polynomials too.

Definition C.5 (Algebraic Set)**.** *Given a set of homogeneous polynomials $J \subset R[\boldsymbol{x}]$, we may define a corresponding (projective) algebraic set $Z(J)$ as a subset of R^D to be*

$$Z(J) \doteq \{[a_1, a_2, \ldots, a_D]^\top \in R^D | f(a_1, a_2, \ldots, a_D) = 0, \forall f \in J\}. \tag{C.7}$$

If we view algebraic sets as the closed sets of R^D, this assigns a topology to the space R^D, which is called the *Zariski topology*.[2]

If $X = Z(J)$ is an algebraic set, an algebraic subset $Y \subset X$ is a set of the form $Y = Z(K)$ (where K is a set of homogeneous polynomials) that happens to be contained in X. A nonempty algebraic set is said to be *irreducible* if it is not the union of two nonempty smaller algebraic subsets. We call irreducible algebraic sets *algebraic varieties*. For instance, every subspace of R^D is an irreducible algebraic variety.

There is an inverse construction of algebraic sets. Given any subset $X \subseteq R^D$, we define the *vanishing ideal of X* to be the set of all polynomials that vanish on X:

$$I(X) \doteq \{f(\boldsymbol{x}) \in R[\boldsymbol{x}] | f(a_1, a_2, \ldots, a_n) = 0, \forall [a_1, a_2, \ldots, a_n]^\top \in X\}. \tag{C.8}$$

[2]This is because the intersection of any algebraic sets is an algebraic set; and the union of finitely many algebraic sets is also an algebraic set.

One can verify that $I(X)$ is an ideal. Treating two polynomials as equivalent if they agree at all the points of X, we get the *coordinate ring* $A(X)$ of X as the quotient $R[\boldsymbol{x}]/I(X)$ (see (Eisenbud 1996) for details).

Now let us consider a set of homogeneous polynomials $J \subset R[\boldsymbol{x}]$ (which is not necessarily an ideal) and a subset $X \subset R^D$ (which is not necessarily an algebraic set).

Proposition C.6. *The following assertions are true:*

1. $I(Z(J))$ *is an ideal that contains* J;
2. $Z(I(X))$ *is an algebraic set that contains* X.

Proposition C.7. *If* X *is an algebraic set and* $I(X)$ *is the vanishing ideal of* X*, then* X *is irreducible if and only if* I *is a prime ideal.*

Proof. If X is irreducible and $f(\boldsymbol{x})g(\boldsymbol{x}) \in I$, since $Z(\{I, f(\boldsymbol{x})\}) \cup Z(\{I, g(\boldsymbol{x})\}) = X$, then either $X = Z(\{I, f(\boldsymbol{x})\})$ or $X = Z(\{I, g(\boldsymbol{x})\})$. That is, either $f(\boldsymbol{x})$ or $g(\boldsymbol{x})$ vanishes on X and is in I. Conversely, suppose $X = X_1 \cup X_2$. If both X_1 and X_2 are algebraic sets strictly smaller than X, then there exist polynomials $f_1(\boldsymbol{x})$ and $f_2(\boldsymbol{x})$ that vanish on X_1 and X_2 respectively, but not on X. Since the product $f_1(\boldsymbol{x})f_2(\boldsymbol{x})$ vanishes on X, we have $f_1(\boldsymbol{x})f_2(\boldsymbol{x}) \in I$, but neither $f_1(\boldsymbol{x})$ nor $f_2(\boldsymbol{x})$ is in I. So I is not prime. □

C.1.3 Algebra and Geometry: Hilbert's Nullstellensatz

In practice, we often use an algebraic set to model a given set of data points, and the (ideal of) polynomials that vanish on the set provides a natural parametric model for the data. One question that is of particular importance in this context is whether there is a one-to-one correspondence between ideals and algebraic sets. This is in general not true, since the ideals $I = (f^2(\boldsymbol{x}))$ and $J = (f(\boldsymbol{x}))$ both vanish on the same algebraic set as the zero set of the polynomial $f(\boldsymbol{x})$. Fortunately, this turns out to be essentially the only case that prevents the one-to-one correspondence between ideals and algebraic sets.

Definition C.8 (Radical Ideal)**.** *Given a (homogeneous) ideal* I *of* $R[\boldsymbol{x}]$*, the (homogeneous) radical ideal of* I *is defined to be*

$$rad(I) \doteq \{f(\boldsymbol{x}) \in R[\boldsymbol{x}] | f(\boldsymbol{x})^m \in I \ \textit{for some integer} \ m\}. \tag{C.9}$$

We leave it to the reader to verify that $\text{rad}(I)$ is indeed an ideal and furthermore, that if I is homogeneous, then so is $\text{rad}(I)$.

Hilbert proved in 1893 the following important theorem that establishes one of the fundamental results in algebraic geometry:

Theorem C.9 (Nullstellensatz)**.** *Let* R *be an algebraically closed field (e.g.,* $R = \mathbb{C}$*). If* $I \subset R[\boldsymbol{x}]$ *is a (homogeneous) ideal, then*

$$I(Z(I)) = rad(I). \tag{C.10}$$

Thus, the maps $I \mapsto Z(I)$ and $X \mapsto I(X)$ induce a one-to-one correspondence between the collection of (projective) algebraic sets of R^D and (homogeneous) radical ideals of $R[\boldsymbol{x}]$.

One may find up to five different proofs for this theorem in (Eisenbud 1996).[3] The importance of the Nullstellensatz cannot be exaggerated. It is a natural extension of Gauss's fundamental theorem of algebra[4] to multivariate polynomials. One of the remarkable consequences of the Nullstellensatz is that it identifies a geometric object (algebraic sets) with an algebraic object (radical ideals).

In our context, we often assume that our data points are drawn from an algebraic set and use the set of vanishing polynomials as a parametric model for the data. Hilbert's Nullstellensatz guarantees that such a model for the data is well defined and unique. To some extent, when we fit vanishing polynomials to the data, we are essentially inferring the underlying algebraic set. In the next section, we will discuss how to extend Hilbert's Nullstellensatz to the practical situation in which we have only finitely many sample points from an algebraic set.

C.1.4 Algebraic Sampling Theory

We often face a common mathematical problem: how to identify a (projective) algebraic set $Z \subseteq R^D$ from a finite, though perhaps very large, number of sample points in Z. In general, the algebraic set Z is not necessarily irreducible,[5] and the ideal $I(Z)$ is not necessarily prime.

From an algebraic viewpoint, it is impossible to recover a continuous algebraic set Z from a finite number of discrete sample points. To see this, note that the set of all polynomials that vanish on one (projective) point $\boldsymbol{z}$ is a submaximal ideal[6] $\mathfrak{m}$ in the (homogeneous) polynomial ring $R[\boldsymbol{z}]$. The set of polynomials that vanish on a set of sample points $\{\boldsymbol{z}_1, \boldsymbol{z}_2, \ldots, \boldsymbol{z}_i\} \subseteq Z$ is the intersection

$$\mathfrak{a}_i \doteq \mathfrak{m}_1 \cap \mathfrak{m}_2 \cap \cdots \cap \mathfrak{m}_i, \tag{C.11}$$

which is a radical ideal that is typically much larger than $I(Z)$.

---

[3] Strictly speaking, for homogeneous ideals, for the one-to-one correspondence to be exact, one should consider only proper radical ideals.

[4] Every degree-n polynomial in one variable has exactly n roots in an algebraically closed field such as $\mathbb{C}$ (counting repeats).

[5] For instance, it is often the case that Z is the union of many subspaces or algebraic surfaces.

[6] The ideal of a point in the affine space is a maximal ideal; and the ideal of a point in the projective space is called a submaximal ideal. They both are "maximal" in the sense that they cannot be a subideal of any other homogeneous ideal of the polynomial ring.

Thus, some additional assumptions must be imposed on the algebraic set in order to make the problem of inferring $I(Z)$ from the samples well defined. Typically, we assume that the ideal $I(Z)$ of the algebraic set Z in question is generated by a set of (homogeneous) polynomials whose degrees are bounded by a relatively small n. That is,

$$\begin{aligned} I(Z) &\doteq \big(f_1, f_2, \ldots, f_s\big) \quad \text{s.t.} \quad \deg(f_j) \leq n, \\ Z(I) &\doteq \big\{z \in R^D \mid f_i(z) = 0, i = 1, 2, \ldots, s\big\}. \end{aligned}$$

We are interested in retrieving $I(Z)$ uniquely from a set of sample points $\{z_1, z_2, \ldots, z_i\} \subseteq Z$. In general, $I(Z)$ is always a proper subideal of $\mathfrak{a}_i$, regardless of how large i is. However, the information about $I(Z)$ can still be retrieved from $\mathfrak{a}_i$ in the following sense.

Theorem C.10 (Sampling of an Algebraic Set)**.** *Consider a nonempty set $Z \subseteq R^D$ whose vanishing ideal $I(Z)$ is generated by polynomials in $R_{\leq n}$. Then there is a finite sequence $F_N = \{z_1, \ldots, z_N\}$ such that the subspace $I(F_N) \cap R_{\leq n}$ generates $I(Z)$.*

Proof. Let $I_{\leq n} = I(Z) \cap R_{\leq n}$. This vector space generates $I(Z)$. Let $\mathfrak{a}_0 = R[\boldsymbol{x}] = I(\emptyset)$. Let $\mathfrak{b}_0 = \mathfrak{a}_0 \cap R_{\leq n}$ and let $A_0 = (\mathfrak{b}_0)$, the ideal generated by polynomials in $\mathfrak{a}_0$ of degree less than or equal to n. Since $1 \in R[\boldsymbol{x}] \cap R_{\leq n}$ is the generator of this ideal, we have $A_0 = R[\boldsymbol{x}]$. Since $Z \neq \emptyset$, then $A_0 \neq I(Z)$. Set $N = 1$ and pick a point $z_1 \in Z$. Then $1(z_1) \neq 0$ (1 is the function that assigns 1 to every point of Z). Let $\mathfrak{a}_1$ be the ideal that vanishes on $\{z_1\}$ and define $\mathfrak{b}_1 = \mathfrak{a}_1 \cap R_{\leq n}$. Further, let $A_1 = (\mathfrak{b}_1)$.[7] Since $I(Z) \subseteq \mathfrak{a}_1$, it follows that $I_{\leq n} \subseteq \mathfrak{b}_1$. If $A_1 = I(Z)$, then we are done. Suppose then that $I(Z) \subset A_1$.

Let us do the induction at this point. Suppose we have found a finite sequence $F_N = \{z_1, z_2, \ldots, z_N\} \subset Z$ with

$$I(F_N) = \mathfrak{a}_N \tag{C.12}$$

$$\mathfrak{b}_N = \mathfrak{a}_N \cap R_{\leq n} \tag{C.13}$$

$$A_N = (\mathfrak{b}_N) \tag{C.14}$$

$$\mathfrak{b}_0 \supset \mathfrak{b}_1 \supset \cdots \supset \mathfrak{b}_N \supseteq I_{\leq n}. \tag{C.15}$$

It follows that $I_{\leq n} \subseteq \mathfrak{b}_N$ and that $I(Z) \subseteq A_N$. If equality holds here, then we are done. If not, then there exist a function $g \in \mathfrak{b}_N$ not in $I(Z)$ and an element $z_{N+1} \in Z$ for which $g(z_{N+1}) \neq 0$. Set $F_{N+1} = \{z_1, \ldots, z_N, z_{N+1}\}$. Then one gets $\mathfrak{a}_{N+1}, \mathfrak{b}_{N+1}, A_{N+1}$ as before with

$$\mathfrak{b}_0 \supset \mathfrak{b}_1 \supset \cdots \supset \mathfrak{b}_N \supset \mathfrak{b}_{N+1} \supseteq I_{\leq n}. \tag{C.16}$$

[7] Here we are using the convention that (S) is the ideal generated by the set S. Recall also that the ring $R[\boldsymbol{x}]$ is *Noetherian* by the Hilbert basis theorem, and so all ideals in the ring are finitely generated (Lang 1993).

We obtain a descending chain of subspaces of the vector space $R_{\leq n}$. This chain must stabilize, since the vector space is finite-dimensional. Hence there is an N for which $\mathfrak{b}_N = I_{\leq n}$, and we are done. □

We point out that in the above proof, no clear bound on the total number N of points needed is given.[8] Nevertheless, from the proof of the theorem, the set of finite sequences of samples that satisfy the theorem is an open set. This is of great practical importance: with probability one, the vanishing ideal of an algebraic set can be correctly determined from a randomly chosen sequence of samples.

Example C.11 (A Hyperplane in $\mathbb{R}^3$). Consider a plane $P = \{z \in R^3 : f(z) = az_1 + bz_2 + cz_3 = 0\}$. Given any two points in general position in the plane P, $f(\boldsymbol{x}) = ax_1 + bx_2 + cx_3$ will be the only (homogeneous) polynomial of degree 1 that fits the two points. In terms of the notation introduced earlier, we have $I(P) = \big(\mathfrak{a}_2 \cap R_{\leq 1}\big)$.

Example C.12 (Zero Polynomial). When $Z = R^D$, the only polynomial that vanishes on Z is the zero polynomial, i.e., $I(Z) = (0)$. Since the zero polynomial is considered to be of degree -1, we have $(\mathfrak{a}_N \cap R_{\leq n}) = \emptyset$ for any given n (and large enough N).

The above theorem can be viewed as a first step toward an algebraic analogy to the well-known Nyquist–Shannon sampling theorem in signal processing, which stipulates that a continuous signal with a limited frequency bandwidth Ω can be uniquely determined from a sequence of discrete samples with a sampling rate higher than 2Ω. Here a signal is replaced by an algebraic set, and the frequency bandwidth is replaced by the bound on the degree of polynomials. It has been widely practiced in engineering that a curve or surface described by polynomial equations can be recovered from a sufficient number of sample points in general configuration, a procedure often loosely referred to as "polynomial fitting." However, the algebraic basis for this is often not clarified, and the conditions for the uniqueness of the solution are usually not well characterized or specified. This problem certainly merits further investigation.

C.1.5 Decomposition of Ideals and Algebraic Sets

Modeling a data set as an algebraic set does not stop at obtaining its vanishing ideal (and polynomials). The ultimate goal is to extract all the internal geometric or algebraic structures of the algebraic set. For instance, if an algebraic set consists of

[8] However, loose bounds can be obtained from the dimension of $R_{\leq n}$ as a vector space. In fact, in the algorithm, we implicitly used the dimension of $R_{\leq n}$ as a bound for N.

multiple subspaces, called a subspace arrangement, we need to know how to derive from its vanishing ideal the number of subspaces, their dimensions, and a basis of each subspace.

Thus, given an algebraic set X or equivalently its vanishing ideal $I(X)$, we want to decompose or segment it into a union of subsets each of which can no longer be further decomposed. As we mentioned earlier, an algebraic set that cannot be decomposed into smaller algebraic sets is called irreducible. As one of the fundamental finiteness theorems of algebraic geometry, we have the following.

Theorem C.13. *An algebraic set can have only finitely many irreducible components. That is, for some n,*

$$X = X_1 \cup X_2 \cup \cdots \cup X_n, \tag{C.17}$$

where $X_1, X_2, \ldots, X_n$ are irreducible algebraic varieties.

Proof. The proof is essentially based on the fact that the polynomial ring $R[\boldsymbol{x}]$ is Noetherian (i.e., finitely generated), and there are only finitely many prime ideals containing $I(X)$ that are minimal with respect to inclusion (See (Eisenbud 1996)). □

The vanishing ideal $I(X_i)$ of each irreducible algebraic variety X_i must be a prime ideal that is minimal over the radical ideal $I(X)$ – there is no prime subideal of $I(X_i)$ that includes $I(X)$. The ideal $I(X)$ is precisely the intersection of all the minimal prime ideals:

$$I(X) = I(X_1) \cap I(X_2) \cap \cdots \cap I(X_n). \tag{C.18}$$

This intersection is called a *minimal primary decomposition* of the radical ideal $I(X)$. Thus the primary decomposition of a radical ideal is closely related to the notion of "segmenting" or "decomposing" an algebraic set into multiple irreducible algebraic varieties: if we know how to decompose the ideal, we can find the irreducible algebraic variety corresponding to each primary component.

We are particularly interested in a special class of algebraic sets known as subspace arrangements. One of the goals of subspace clustering and modeling is to decompose a subspace arrangement into individual (irreducible) subspaces (see Chapter 5). In later sections, we will further study the algebraic properties of subspace arrangements.

C.1.6 Hilbert Function, Polynomial, and Series

Finally, we introduce an important invariant of algebraic sets, given by the Hilbert function. Knowing the values of the Hilbert function can be very useful in the identification of subspace arrangements, especially the number of subspaces and their dimensions.

Given a (projective) algebraic set Z and its vanishing ideal $I(Z)$, we can grade the ideal by degree as

$$I(Z) = I_0(Z) \oplus I_1(Z) \oplus \cdots \oplus I_i(Z) \oplus \cdots . \tag{C.19}$$

The *Hilbert function* of Z is defined to be

$$h_I(i) \doteq \dim(I_i(Z)). \tag{C.20}$$

Notice that $h_I(i)$ is exactly the number of linearly independent polynomials of degree i that vanish on Z. In this book, we also refer to h_I as the Hilbert function of the algebraic set Z.[9]

The *Hilbert series*, also known as the Poincaré series, of the ideal I is defined to be the power series[10]

$$\mathcal{H}(I,t) \doteq \sum_{i=0}^{\infty} h_I(i)t^i = h_I(0) + h_I(1)t + h_I(2)t^2 + \cdots . \tag{C.21}$$

Thus, given $\mathcal{H}(I,t)$, we know all the values of the Hilbert function h_I from its coefficients.

Example C.14 (Hilbert Series of a Polynomial Ring). The Hilbert series of the polynomial ring $R[\boldsymbol{x}] = \mathbb{R}[x_1, x_2, \ldots, x_D]$ is

$$\mathcal{H}(R[\boldsymbol{x}],t) = \sum_{i=0}^{\infty} \dim(R_i)t^i = \sum_{i=0}^{\infty} \binom{D+i-1}{i} t^i = \frac{1}{(1-t)^D}. \tag{C.22}$$

One can verify the correctness of the formula with the special case $D = 1$. Obviously, the coefficients of the Hilbert series of any ideal (as a subset of $R[\boldsymbol{x}]$) are bounded by those of $\mathcal{H}(R[\boldsymbol{x}],t)$, and hence the Hilbert series converges.

Example C.15 (Hilbert Series of a Subspace). The above formula can be generalized to the vanishing ideal of a subspace S of dimension d in $\mathbb{R}^D$. Let the codimension of the subspace be $c = D - d$. We have

$$\mathcal{H}(I(S),t) = \left(\frac{1}{(1-t)^c} - 1\right) \cdot \left(\frac{1}{(1-t)^{D-c}}\right) = \frac{1-(1-t)^c}{(1-t)^D}. \tag{C.23}$$

---

[9] In the literature, however, the Hilbert function of an algebraic set Z is sometimes defined to be the dimension of the homogeneous components of the coordinate ring $A(Z) \doteq R[\boldsymbol{x}]/I(Z)$ of Z, which is the codimension of $I_i(Z)$ as a subspace in R_i.

[10] In general, the Hilbert series can be defined for any finitely generated graded module $E = \bigoplus_{i=1}^{\infty} E_i$ using any Euler–Poincaré $\mathbb{Z}$-valued function $h_E(\cdot)$ as $\mathcal{H}(E,t) \doteq \sum_{i=0}^{\infty} h_E(i)t^i$ (Lang 1993). Here, for $E = I$, we choose $h_I(i) = \dim(I_i)$.

The following theorem, also due to Hilbert, reveals that the values of the Hilbert function of an ideal have some remarkable properties:

Theorem C.16 (Hilbert Polynomial)**.** *Let $I(Z)$ be the vanishing ideal of an algebraic set Z over $R[x_1, \ldots, x_D]$. Then the values of its Hilbert function $h_I(i)$ agree, for large i, with those of a polynomial of degree $\leq D$. This polynomial, denoted by $H_I(i)$, is called the* Hilbert polynomial *of $I(Z)$.*

Then in the above example, for the polynomial ring, the Hilbert function itself is a polynomial in i:

$$H_R(i) = h_R(i) = \binom{D+i-1}{i} = \frac{1}{(D-1)!}(D+i-1)(D+i-2)\cdots(i+1).$$

However, for a general ideal I (of an algebraic set), it is not necessarily true that all values of its Hilbert function h_I agree with those of its Hilbert polynomial H_I. They might agree only when i is large enough. Thus, for a given algebraic set (or ideal), it would be interesting to know how large i needs to be in order for the Hilbert function to coincide with a polynomial. As we will soon see, for subspace arrangements, there is a very elegant answer to this question. One can even derive closed-form formulas for the Hilbert polynomials. These results are very important and useful for the subspace clustering problem, both conceptually and computationally.

C.2 Ideals of Subspace Arrangements

In this book, the main problem that we study is how to cluster a collection of data points drawn from a subspace arrangement $\mathcal{A} = \{S_1, S_2, \ldots, S_n\}$, formally introduced in Chapter 5;[11] $Z_{\mathcal{A}} = S_1 \cup S_2 \cup \cdots \cup S_n$ is the union of all the subspaces, and $Z_{\mathcal{A}}$ can be naturally described as the zero set of a set of polynomials, which makes it an *algebraic set*. The solution to the above problem typically relies on inferring the subspace arrangement $Z_{\mathcal{A}}$ from the data points. Thus, knowing the algebraic properties of $Z_{\mathcal{A}}$ may significantly facilitate this task.

Although subspace arrangements seem to be a very simple class of algebraic sets, a full characterization of their algebraic properties is a surprisingly difficult, if not impossible, task. Subspace arrangements have been a centuries-old subject that still actively interweaves many mathematical fields: algebraic geometry and topology, combinatorics, and complexity theory, graph and lattice theory, etc. Although the results are extremely rich and deep, in fact only a few special classes of subspace arrangements have been well characterized.

[11] Unless stated otherwise, the subspace arrangement considered will always be a central arrangement, as in Definition 5.4.

In the remaining sections of this appendix, we examine some important concepts and properties of subspace arrangements that are closely related to the subspace-clustering problem. The purpose of these sections is twofold: 1. to provide a rigorous justification for the algebraic subspace clustering algorithms derived in Chapter 5; 2. to summarize some important in-depth properties of subspace arrangements, which may suggest potential improvements of the algorithms. For readers who are interested only in the basic subspace clustering algorithms and their applications, these sections can be skipped on a first reading.

Vanishing Ideal of a Subspace.
A d-dimensional subspace S can be defined by $k = D-d$ linearly independent linear forms $\{l_1, l_2, \ldots, l_k\}$:

$$S \doteq \{\boldsymbol{x} \in R^D : l_i(\boldsymbol{x}) = 0,\ i = 1, 2, \ldots, k = D - d\}, \tag{C.24}$$

where l_i is of the form $l_i(\boldsymbol{x}) = a_{i1}x_1 + a_{i2}x_2 + \cdots a_{iD}x_D$ with $a_{ij} \in R$. Let S^* denote the space of all linear forms that vanish on S. Then $\dim(S^*) \doteq k = D - d$. The subspace S is also called the zero set of S^*, i.e., points in the ambient space that vanish on all polynomials in S^*, which is denoted by $Z(S^*)$. We define

$$I(S) \doteq \{p \in R[\boldsymbol{x}] : p(\boldsymbol{x}) = 0, \forall \boldsymbol{x} \in S\}. \tag{C.25}$$

Clearly, $I(S)$ is an ideal generated by linear forms in S^*, and it contains polynomials of all degrees that vanish on the subspace S. Every polynomial $p(\boldsymbol{x})$ in $I(S)$ can be written as a superposition:

$$p = l_1 h_1 + l_2 h_2 + \cdots + l_k h_k \tag{C.26}$$

for some polynomials $h_1, h_2, \ldots, h_k \in R[\boldsymbol{x}]$. Furthermore, $I(S)$ is a prime ideal.[12]

Vanishing Ideal of a Subspace Arrangement
Given a subspace arrangement $Z_{\mathcal{A}} = S_1 \cup S_2 \cup \cdots \cup S_n$, its vanishing ideal is

$$I(Z_{\mathcal{A}}) = I(S_1) \cap I(S_2) \cap \cdots \cap I(S_n). \tag{C.27}$$

The ideal $I(Z_{\mathcal{A}})$ can be graded by the degree of its polynomials

$$I(Z_{\mathcal{A}}) = I_m(Z_{\mathcal{A}}) \oplus I_{m+1}(Z_{\mathcal{A}}) \oplus \cdots \oplus I_i(Z_{\mathcal{A}}) \oplus \cdots . \tag{C.28}$$

Each $I_i(Z_{\mathcal{A}})$ is a vector space that consists of forms of degree i in $I(Z_{\mathcal{A}})$, and $m \geq 1$ is the least degree of the polynomials in $I(Z_{\mathcal{A}})$. Notice that forms that vanish on $Z_{\mathcal{A}}$ may have degrees strictly less than n. One example is an arrangement of two lines and one plane in $\mathbb{R}^3$. Since any two lines lie on a plane, the arrangement can

[12] It is a prime ideal because for every product $p_1 p_2 \in I(S)$, either $p_1 \in I(S)$ or $p_2 \in I(S)$.

be embedded into a hyperplane arrangement of two planes, and there exist forms of second degree that vanish on the union of the three subspaces. The dimension of $I_i(Z_\mathcal{A})$ is known as the Hilbert function $h_I(i)$ of $Z_\mathcal{A}$.

Example C.17 (Boolean Arrangement). The Boolean arrangement is the collection of coordinate hyperplanes $H_j \doteq \{\boldsymbol{x} : x_j = 0\}, 1 \leq j \leq D$. The vanishing ideal of the Boolean arrangement is generated by a single polynomial $p(\boldsymbol{x}) = x_1 x_2 \cdots x_D$ of degree D.

Example C.18 (Braid Arrangement). The braid arrangement is the collection of hyperplanes $H_{jk} \doteq \{\boldsymbol{x} : x_j - x_k = 0\}, 1 \leq j \neq k \leq D$. Similarly, the vanishing ideal of the Braid arrangement is generated by a single polynomial $p(\boldsymbol{x}) = \prod_{1 \leq j < k \leq D}(x_j - x_k)$.

Theorem C.19 (Regularity of Subspace Arrangements)**.** *The vanishing ideal $I(Z_\mathcal{A})$ of a subspace arrangement $Z_\mathcal{A} = S_1 \cup S_2 \cup \cdots \cup S_n$ is n-regular. This implies that $I(Z)$ has a set of generators with degree $\leq n$.*

Proof. For the concept of n-regularity and the proof of the above statement, please refer to (Derksen 2007) and references therein. □

Due to the above theorem, the subspace arrangement $Z_\mathcal{A}$ is uniquely determined as the zero set of all polynomials of degree up to n in its vanishing ideal, i.e., as the zero set of polynomials in

$$Z_\mathcal{A} = Z(I_{\leq n}),$$

where $I_{\leq n} \doteq I_0 \oplus I_1 \oplus \cdots \oplus I_n$.

Product Ideal of a Subspace Arrangement
Let $J(Z_\mathcal{A})$ be the ideal generated by the products of linear forms

$$\{l_1 \cdot l_2 \cdots l_n, \quad \forall l_j \in S_j^*, j = 1, \ldots, n\}.$$

Or equivalently, we can define $J(Z_\mathcal{A})$ to be the product of the n ideals $I(S_1), I(S_2), \ldots, I(S_n)$:

$$J(Z_\mathcal{A}) \doteq I(S_1) \cdot I(S_2) \cdots I(S_n).$$

Then the *product ideal* $J(Z_\mathcal{A})$ is a subideal of $I(Z_\mathcal{A})$. Nevertheless, the two ideals share the same zero set:

$$Z_\mathcal{A} = Z(J) = Z(I). \tag{C.29}$$

By definition, I is the largest ideal that vanishes on $Z_\mathcal{A}$. In fact, I is the *radical ideal* of the product ideal J, i.e., $I = \text{rad}(J)$. We may also grade the ideal $J(Z_\mathcal{A})$ by the degree

$$J(Z_{\mathcal{A}}) = J_n(Z_{\mathcal{A}}) \oplus J_{n+1}(Z_{\mathcal{A}}) \oplus \cdots \oplus J_i(Z_{\mathcal{A}}) \oplus \cdots . \tag{C.30}$$

Notice that unlike I, the lowest degree of polynomials in J always starts from n, the number of subspaces. The Hilbert function of J is denoted by $h_J(i) = \dim(J_i(Z_{\mathcal{A}}))$. As we will soon see, the Hilbert functions (or polynomials, or series) of the product ideal J and the vanishing ideal I have very interesting and important relationships.

C.3 Subspace Embedding and PL-Generated Ideals

Let $Z_{\mathcal{A}}$ be a central subspace arrangement $Z_{\mathcal{A}} = S_1 \cup S_2 \cup \cdots \cup S_n$. Let $Z_{\mathcal{A}'} = S'_1 \cup S'_2 \cup \cdots \cup S'_{n'}$ be another (central) subspace arrangement. If we have $Z_{\mathcal{A}} \subseteq Z_{\mathcal{A}'}$, then it is necessary that for all $S_j \subset Z_{\mathcal{A}}$, there exist $S'_{j'} \subset Z_{\mathcal{A}'}$ such that $S_j \subseteq S'_{j'}$. If so, we call

$$Z_{\mathcal{A}} \subseteq Z_{\mathcal{A}'}$$

a *subspace embedding*. Beware that it is possible that $n' < n$ for a subspace embedding, since more than one subspace S_j of $Z_{\mathcal{A}}$ may belong to the same subspace $S_{j'}$ of $Z_{\mathcal{A}'}$. The subspace arrangements in Theorem 5.14 are examples of subspace embeddings. If $Z_{\mathcal{A}'}$ happens to be a hyperplane arrangement, we call the embedding a *hyperplane embedding*.

Is the zero-set of each homogeneous component of $I(Z_{\mathcal{A}})$, in particular $I_m(Z_{\mathcal{A}})$, a subspace embedding of $Z_{\mathcal{A}}$? Unfortunately, this is not true, since counterexamples can be constructed.

Example C.20 (Five Lines in $\mathbb{R}^3$). Consider five points in $\mathbb{P}^2$ (or equivalently, five lines in $\mathbb{R}^3$). The Veronese embedding of order two of a point $\boldsymbol{x} = [x_1, x_2, x_3] \in \mathbb{R}^3$ is $[x_1^2, x_1x_2, x_1x_3, x_2^2, x_2x_3, x_3^2] \in \mathbb{R}^6$. For five points in general position, the matrix $\boldsymbol{V}_2 = [\nu_2(\boldsymbol{x}_1), \nu_2(\boldsymbol{x}_2), \ldots \nu_2(\boldsymbol{x}_5)]$ is of rank 5. Let $\boldsymbol{c}^\top$ be the only vector in the left null space of $\boldsymbol{V}_2$ such that $\boldsymbol{c}^\top \boldsymbol{V}_2 = 0$. Then $p(\boldsymbol{x}) = \boldsymbol{c}^\top \nu_2(\boldsymbol{x})$ is in general an irreducible quadratic polynomial. Thus, the zero set of $I_2(Z_{\mathcal{A}}) = p(\boldsymbol{x})$ is not a subspace arrangement but an (irreducible) cone in $\mathbb{R}^3$.

Nevertheless, the following statement allows us to retrieve a subspace embedding from any polynomials in the vanishing ideal $I(Z_{\mathcal{A}})$.

Theorem C.21 (Hyperplane Embedding via Differentiation)**.** *For every polynomial p in the vanishing ideal $I(Z_{\mathcal{A}})$ of a subspace arrangement $Z_{\mathcal{A}} = S_1 \cup S_2 \cup \cdots \cup S_n$ and n points $\{\boldsymbol{x}_i \in S_i\}_{i=1}^n$ in general position, the union of the hyperplanes $\cup_{i=1}^n H_i = \{\boldsymbol{x} : \nabla p(\boldsymbol{x}_i)^\top \boldsymbol{x} = 0\}$ is a hyperplane embedding of the subspace arrangement.*

Proof. The proof is based on the simple fact that the derivative (gradient) $\nabla f(\boldsymbol{x})$ of any smooth function $f(\boldsymbol{x})$ is orthogonal to (the tangent space of) its level set $f(\boldsymbol{x}) = c$. □

In the above statement, if we replace p with a collection of polynomials in the vanishing ideal, their derivatives give a subspace embedding in a similar fashion as the hyperplane embedding. When the collection contains all the generators of the vanishing ideal, the subspace embedding becomes tight: the resulting subspace arrangement coincides with the original one. This property has been used in the development of algebraic subspace clustering algorithms in Chapter 5.

Another concept that is closely related to subspace embedding is a *pl-generated ideal*.

Definition C.22 (pl-Generated Ideals)**.** *An ideal is said to be* pl-generated *if it is generated by products of linear forms.*

If the ideal of a subspace arrangement $Z_{\mathcal{A}}$ is pl-generated, then the zero set of every generator gives a hyperplane embedding of $Z_{\mathcal{A}}$.

Example C.23 (Hyperplane Arrangements). If $Z_{\mathcal{A}}$ is a hyperplane arrangement, then $I(Z_{\mathcal{A}})$ is always pl-generated, since it is generated by a single polynomial of the form[13]

$$p(\boldsymbol{x}) = (\boldsymbol{b}_1^\top \boldsymbol{x})(\boldsymbol{b}_2^\top \boldsymbol{x}) \cdots (\boldsymbol{b}_n^\top \boldsymbol{x}), \tag{C.31}$$

where $\boldsymbol{b}_i \in R^D$ are the normal vectors to the hyperplanes.

Obviously, the vanishing ideal $I(S)$ of a single subspace S is always pl-generated. The following example shows that this is also true for an arrangement of two subspaces.

Example C.24 (Two Subspaces). Let us show that for an arrangement $Z_{\mathcal{A}}$ of two subspaces, $I(Z_{\mathcal{A}})$ is always pl-generated. Let $Z_{\mathcal{A}} = S_1 \cup S_2$ and define $U^* \doteq S_1^* \cap S_2^*$ and $V^* \doteq S_1^* \setminus U^*, W^* \doteq S_2^* \setminus U^*$. Let $(u_1, u_2, \ldots, u_k)$ be a basis for U^*, $(v_1, v_2, \ldots, v_l)$ a basis for V^*, and $(w_1, w_2, \ldots, w_m)$ a basis for W^*. Then $I(Z_{\mathcal{A}}) = I(S_1) \cap I(S_2)$ is generated by $(u_1, \ldots, u_k, v_1 w_1, v_1 w_2, \ldots, v_l w_m)$.

Now consider an arrangement of n subspaces $Z_{\mathcal{A}} = S_1 \cup S_2 \cup \cdots \cup S_n$. By its definition, the product ideal $J(Z_{\mathcal{A}})$ is always pl-generated. Now, is the vanishing ideal $I(Z_{\mathcal{A}})$ always pl-generated? Unfortunately, this is not true. Below are some counterexamples.

Example C.25 (Lines in $\mathbb{R}^3$ (Björner et al. 2005)). For a central arrangement $Z_{\mathcal{A}}$ of r lines in general position in $\mathbb{R}^3$, $I(Z_{\mathcal{A}})$ is not pl-generated when $r = 5$ or $r > 6$. Example C.20 gives a proof for the case with $r = 5$.

Example C.26 (Planes in $\mathbb{R}^4$ (Björner et al. 2005)). For a central arrangement $Z_{\mathcal{A}}$ of r planes in general position in $\mathbb{R}^4$, $I(Z_{\mathcal{A}})$ is not pl-generated for all $r > 2$.

[13] In algebra, an ideal that is generated by a single generator is called a principal ideal.

However, can each homogeneous component $I_i(Z_{\mathcal{A}})$ be "pl-generated" when i is large enough? For instance, can it be that $I_n = J_n = S_1^* \cdot S_2^* \cdots S_n^*$? This is in general not true for an arbitrary arrangement. Below is a counterexample.

Example C.27 (Three Subspaces in $\mathbb{R}^5$; due to R. Fossum). Consider $R[\boldsymbol{x}] = \mathbb{R}[x_1, \ldots, x_5]$ and an arrangement $Z_{\mathcal{A}}$ of three three-dimensional subspaces in $\mathbb{R}^5$ whose vanishing ideals are given by, respectively,

$$I(S_1) = (x_1, x_2), \quad I(S_2) = (x_3, x_4), \quad I(S_3) = ((x_1 + x_3), (x_2 + x_4)).$$

Denote their intersection by $I = I(S_1) \cap I(S_2) \cap I(S_3)$. The intersection contains the element

$$x_1x_4 - x_2x_3 = (x_1 + x_3)x_4 - (x_2 + x_4)x_3 = x_1(x_2 + x_4) - x_2(x_1 + x_3).$$

Then every element $(x_1x_4 - x_2x_3)l(x_1, \ldots, x_5)$ with l a linear form is in $I_3(Z_{\mathcal{A}})$, the homogeneous component of elements of degree three. In particular, $(x_1x_4 - x_2x_3)x_5$ is in $I_3(Z_{\mathcal{A}})$. However, one can check that this element cannot be written in the form

$$\sum_i (a_ix_1 + b_ix_2)(c_ix_2 + d_ix_4)(e_i(x_1 + x_3) + f_i(x_2 + x_4))$$

for any $a_i, b_i, c_i, d_i, e_i, f_i \in \mathbb{R}$. Thus, $I_3(Z_{\mathcal{A}})$ is not spanned by $S_1^* \cdot S_2^* \cdot S_3^*$.

However, notice that the subspaces in the above example are not in "general position": their intersections are not of the minimum possible dimension. Could $I_n = J_n = S_1^* \cdot S_2^* \cdots S_n^*$ be instead true for n subspaces if they are in general position? The answer is yes. In fact, we can say more than that. As we will see in the next section, from the Hilbert functions of I and J, we actually have

$$I_i = J_i, \quad \forall i \geq n$$

if $S_1, S_2, \ldots, S_n$ are "transversal" (i.e., all intersections are of minimum possible dimension). In other words, J_i could differ from I_i only for $i < n$.

C.4 Hilbert Functions of Subspace Arrangements

In this section, we study the Hilbert functions of subspace arrangements defined in Section C.1.6. We first discuss a few reasons why in the context of generalized principal component analysis, it is very important to know the values of the Hilbert function for the vanishing ideal I or the product ideal J of a subspace arrangement. We then examine the values of the Hilbert function for a few special examples. Finally, we give a complete characterization of the Hilbert function, the Hilbert polynomial, and the Hilbert series of a general subspace arrangement. In particular,

we give a closed-form formula for the Hilbert polynomial of the vanishing ideal and the product ideal of the subspace arrangement.

C.4.1 Hilbert Function and Algebraic Subspace Clustering

In general, for a subspace arrangement $Z_{\mathcal{A}} = S_1 \cup S_2 \cup \cdots \cup S_n$ in general position, the values of the Hilbert function $h_I(i)$ of its vanishing ideal $I(Z_{\mathcal{A}})$ are invariant under a continuous change of the positions of the subspaces. They depend only on the dimensions of the subspaces $d_1, d_2, \ldots, d_n$ or their codimensions $c_i = D - d_i$, $i = 1, 2, \ldots, n$. Thus, the Hilbert function gives a rich set of invariants of subspace arrangements. In the context of subspace clustering, such invariants can help to determine the type of the subspace arrangement, such as the number of subspaces and their individual dimensions from a given set of (possibly noisy) sample points.

To see this, consider a sufficiently large number of sample points in general position $\mathcal{X} = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_N\} \subset Z_{\mathcal{A}}$ that are drawn from the subspaces, and let the embedded data matrix (via the Veronese map of degree i) be

$$\boldsymbol{V}_i \doteq [\nu_i(\boldsymbol{x}_1), \nu_i(\boldsymbol{x}_2), \ldots, \nu_i(\boldsymbol{x}_N)]^\top. \tag{C.32}$$

According to the algebraic sampling theorem of Appendix C.1.4, the dimension of $\text{Null}(\boldsymbol{V}_i)$ is exactly the number of linearly independent polynomials of degree i that vanish on $Z_{\mathcal{A}}$. That is, the following relation holds:

$$\dim(\text{Null}(\boldsymbol{V}_i)) = h_I(i), \tag{C.33}$$

or equivalently,

$$\text{rank}(\boldsymbol{V}_i) = \dim(R_i) - h_I(i). \tag{C.34}$$

Thus, if we know the Hilbert function for different subspace arrangements in advance, we can determine from the rank of the data matrix from which subspace arrangement the sample data points are drawn. The following example illustrates the basic idea.

Example C.28 (Three Subspaces in $\mathbb{R}^3$). Suppose that we know only that our data are drawn from an arrangement of three subspaces in $\mathbb{R}^3$. There are in total four different types of such arrangements, shown in Figure C.1. The values of their corresponding Hilbert functions are listed in Table C.1. Given a sufficiently large number N of sample points from one of the above subspace arrangements, the rank of the embedded data matrix $\boldsymbol{V}_3 \in \mathbb{R}^{N \times 10}$ can be, instead of any value between 1 and 10, only $10 - h_I(3) = 9, 8, 6, 3$, which correspond to the only four possible configurations of three subspaces in $\mathbb{R}^3$: three planes, two planes and one line, one plane and two lines, or three lines, respectively, as shown in Figure C.1.

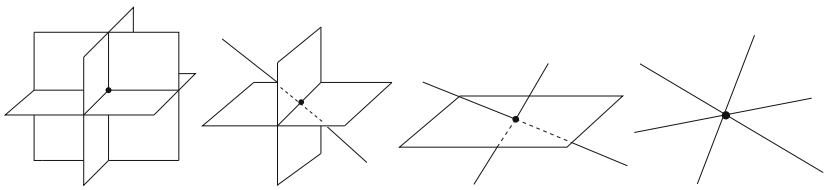

Fig. C.1 Four configurations of three subspaces in $\mathbb{R}^3$. The numbers are the codimensions (c_1, c_2, c_3) of the subspaces.

Table C.1 Values of the Hilbert functions of the four arrangements (assuming the subspaces are in general position).

c_1	c_2	c_3	$h_{I(Z,A)}(1)$	$h_{I(Z,A)}(2)$	$h_{I(Z,A)}(3)$
1	1	1	0	0	1
1	1	2	0	0	2
1	2	2	0	1	4
2	2	2	0	3	7

This suggests that given the dimensions of individual subspaces, we may know the rank of the embedded data matrix. Conversely, given the rank of the embedded data matrix, we can determine to a large extent the possible dimensions of the individual subspaces. Therefore, knowing the values of the Hilbert function will help us to at least rule out in advance impossible rank values for the embedded data matrix or the impossible subspace dimensions. This is particularly useful when the data are corrupted by noise, so that there is ambiguity in determining the rank of the embedded data matrix or the dimensions of the subspaces.

The next example illustrates how the values of the Hilbert function can help determine the correct number of subspaces.

Example C.29 (Overfit Hyperplane Arrangements in $\mathbb{R}^5$). Consider a data set sampled from a number of hyperplanes in general position in $\mathbb{R}^5$. Suppose we know only that the number of hyperplanes is at most 4, and we embed the data via the degree-4 Veronese map anyway. Table C.2 gives the possible values of the Hilbert function for an arrangement of 4, 3, 2, 1 hyperplanes in $\mathbb{R}^5$, respectively. Here we use the convention that the empty set has codimension 5 in $\mathbb{R}^5$.

The first row shows that if the number of hyperplanes is exactly equal to the degree of the Veronese map, then $h_I(4) = 1$, i.e., the data matrix $\boldsymbol{V}_4$ has a rank-1 null space. The following rows show the values of $h_I(4)$ when the number of hyperplanes is $n = 3, 2, 1$, respectively. If the rank of the matrix $\boldsymbol{V}_4$ matches any of these values, we know exactly the number of hyperplanes in the arrangement. Figure C.2 shows a superimposed plot of the singular values of $\boldsymbol{V}_4$ for sample points drawn from $n = 1, 2, 3, 4$ hyperplanes in $\mathbb{R}^5$, respectively.

Table C.2 Values of the Hilbert function of (codimension-1) hyperplane arrangements in $\mathbb{R}^5$.

c_1	c_2	c_3	c_4	$h_{I(Z_{\mathcal{A}})}(4)$	$\text{rank}(\boldsymbol{V}_4)$
1	1	1	1	1	69
1	1	1	5	5	65
1	1	5	5	15	55
1	5	5	5	35	35

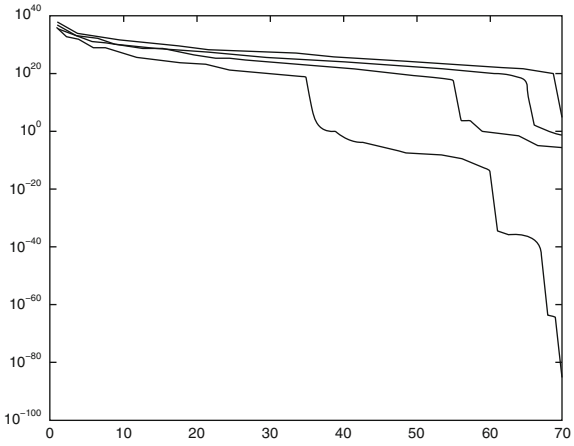

Fig. C.2 A superimposed semilog plot of the singular values of the embedded data matrix $\boldsymbol{V}_4$ for $n = 1, 2, 3, 4$ hyperplanes in $\mathbb{R}^5$, respectively. The rank drops at $35, 55, 65, 69$, which confirms the theoretical values of the Hilbert function.

Thus, in general, knowing the values of $h_I(i)$ even for $i > n$ may significantly help determine the correct number of subspaces in case the degree i of the Veronese map used for constructing the data matrix $\boldsymbol{V}_i$ is strictly higher than the number n of nontrivial subspaces in the arrangement.

The above examples show merely a few cases in which the values of the Hilbert function may facilitate solving the subspace clustering and modeling problem in Chapter 5, in particular the model-selection issue. It now remains as a question how to compute the values of the Hilbert function for arbitrary subspace arrangements.

Mathematically, we are interested in finding closed-form formulas, if they exist at all, for the Hilbert function (or the Hilbert polynomial, or the Hilbert series) of the subspace arrangements. As we will soon show, if the subspace arrangements are transversal (i.e., every intersection of subsets of the subspaces has the smallest possible dimension), we are able to show that the Hilbert function (of both I and J) agrees with the Hilbert polynomial (of both I and J) with $i \geq n$; and a closed-form formula for the Hilbert polynomial is known (and will be given later). However, no

general formula is known for the Hilbert function (or series) of I, especially for the values $h_I(i)$ with $i < n$. For those values, one can still compute them in advance numerically based on the identity

$$h_I(i) = \dim(\text{Null}(\boldsymbol{V}_i)) \tag{C.35}$$

from a sufficient set of samples on the subspace arrangements. The values for each type of arrangement needs to be computed only once, and the results can be stored in a table such as Table C.1 for each ambient space dimension D and number of subspaces n. We may later query these tables to retrieve information about the subspace arrangements and exploit relations among these values for different practical purposes.

However, computing the values of h_I numerically can be very expensive, especially when the dimension of the space (or the subspaces) is high. In order to densely sample the high-dimensional subspaces, the number of samples grows exponentially with the number of subspaces and their dimensions. Indeed, the MATLAB package that we are using runs out of the memory limit of 2 GB for computing the table for the case $D = 12$ and $n = 6$.

Fortunately, for most applications in image processing, computer vision, or systems identification, it is typically sufficient to know the values of $h_I(i)$ up to $n = 10$ and $D = 12$. For instance, for most images, the first $D = 12$ principal components already keep up to 99% of the total energy of the image, which is more than sufficient for any subsequent representation or compression purposes. Furthermore, if one chooses to use 2×2 blocks to represent a color image, then each block becomes one data point of dimension $2 \times 2 \times 3 = 12$. The number of segments sought for an image is typically less than ten. In system identification, the dimensions of the subspaces correspond to the orders of the systems, and they are typically less than 10.

C.4.2 Special Cases of the Hilbert Function

Before we study the Hilbert function for general subspace arrangements in the next section, we here give a few special cases for which we have computed certain values of the Hilbert function.

Example C.30 (Hyperplane Arrangements). Consider $Z_{\mathcal{A}} = S_1 \cup S_2 \cup \ldots \cup S_n \subset \mathbb{R}^D$ with each S_i a hyperplane. The subspaces S_i are of codimension 1, i.e., $c_1 = c_2 = \cdots = c_n = 1$. Then we have $h_I(n) = 1$, which is consistent with the fact that there is exactly one (factorable) polynomial of degree n that fits n hyperplanes. Furthermore, $h_I(i) = 0$ for all $i < n$, and

$$h_I(n+i) = \binom{D+i-1}{i}, \quad \forall i \geq 1.$$

We can generalize the case of hyperplanes to the following example.

Example C.31 (Subspaces Whose Duals Have No Intersection). Consider a subspace arrangement $Z_{\mathcal{A}} = S_1 \cup S_2 \cup \ldots \cup S_n \subset \mathbb{R}^D$ with $S_i^* \cap S_j^* = 0$ for all $i \neq j$. In other words, if the codimensions of $S_1, S_2, \ldots, S_n$ are $c_1, c_2, \ldots, c_n$, respectively, we have $c_1 + c_2 + \cdots + c_n \leq D$. Notice that hyperplane arrangements are a special case here. Generalizing the result in Example C.15, one can show that the Hilbert series of $I(Z_{\mathcal{A}})$ (and $J(Z_{\mathcal{A}})$) is

$$\mathcal{H}(I(Z_{\mathcal{A}}), t) = \mathcal{H}(J(Z_{\mathcal{A}}), t) = f(t) \doteq \frac{\prod_{i=1}^{n} \left(1 - (1-t)^{c_i}\right)}{(1-t)^D}. \tag{C.36}$$

The values of the Hilbert function $h_I(i)$ can be computed from the coefficients of the function $f(t)$ associated with t^i.

However, if the dual subspaces S_i^* have nontrivial intersections, the computation of Hilbert series and function becomes much more complicated. Below we give some special examples and leave the general study to the next section.

Example C.32 (Hilbert Function of Two Subspaces). We here derive a closed-form formula of $h_I(2)$ for an arrangement of $n = 2$ subspaces $Z_{\mathcal{A}} = S_1 \cup S_2$ in general position (see also Example C.24). Suppose their codimensions are c_1 and c_2, respectively. In $R_1 \sim \mathbb{R}^D$, the intersection of their dual subspaces S_1^* and S_2^* has the dimension

$$c \doteq \max\{c_1 + c_2 - D,\ 0\}. \tag{C.37}$$

Then we have

$$\begin{aligned} h_I(2) &= c \cdot (c+1)/2 + c \cdot (c_1 - c) + c \cdot (c_2 - c) + (c_1 - c) \cdot (c_2 - c) \\ &= c_1 \cdot c_2 - c \cdot (c-1)/2. \end{aligned} \tag{C.38}$$

Example C.33 (Three Subspaces in $\mathbb{R}^5$). Consider an arrangement of three subspaces $Z_{\mathcal{A}} = S_1 \cup S_2 \cup S_3 \subset \mathbb{R}^5$ in general position. After a change of coordinates, we may assume $S_1^* = \text{span}\{x_1, x_2, x_3\}$, $S_2^* = \text{span}\{x_1, x_4, x_5\}$, and $S_3^* = \text{span}\{x_2, x_3, x_4, x_5\}$. The value of $h_I(3)$ in this case is equal to $\dim(S_1^* \cdot S_2^* \cdot S_3^*)$. Firstly, we compute $S_1^* \cdot S_2^*$ and obtain a basis for it:

$$S_1^* \cdot S_2^* = \text{span}\{x_1^2, x_1x_4, x_1x_5, x_2x_1, x_2x_4, x_2x_5, x_3x_1, x_3x_4, x_3x_5\}.$$

From this, one can compute the basis for $S_1^* \cdot S_2^* \cdot S_3^*$:

$$\begin{aligned} S_1^* \cdot S_2^* \cdot S_3^* = \text{span}\{&x_1^2x_2, x_1x_2x_4, x_1x_2x_5, x_1x_2^2, x_2^2x_4, x_2^2x_5, x_1x_2x_3, x_2x_3x_4, \\ &x_2x_3x_5, x_1^2x_3, x_1x_3x_4, x_1x_3x_5, x_1x_3^2, x_3^2x_4, x_3^2x_5, x_1^2x_4, x_1x_4^2, \\ &x_1x_4x_5, x_2x_4^2, x_2x_4x_5, x_3x_4^2, x_3x_4x_5, x_1^2x_5, x_1x_5^2, x_2x_5^2, x_3x_5^2\}. \end{aligned}$$

Thus, we have $h_I(3) = 26$.

Table C.3 Values of the Hilbert function $h_I(5)$ for arrangements of five subspaces in $\mathbb{R}^3$.

c_1	c_2	c_3	c_4	c_5	$h_I(5)$
1	1	1	1	1	1
1	1	1	1	2	2
1	1	1	2	2	4
1	1	2	2	2	7
1	2	2	2	2	11
2	2	2	2	2	16

Example C.34 (Five Subspaces in $\mathbb{R}^3$). Consider an arrangement of five subspaces $S_1, S_2, \ldots, S_5$ in $\mathbb{R}^3$ of codimensions $c_1, c_2, \ldots, c_5$, respectively. We want to compute the value of $h_I(5)$, i.e., the dimension of homogeneous polynomials of degree five that vanish on the five subspaces $Z_{\mathcal{A}} = S_1 \cup S_2 \cup \cdots \cup S_5$. For all the possible values of $1 \leq c_1 \leq c_2 \leq \cdots \leq c_5 < 3$, we have computed the values of $\mathcal{D}_5^3$ and listed them in Table C.3. Notice that the values of $h_I(3)$ in the earlier Table C.1 form a subset of those of $h_I(5)$ in Table C.3. In fact, many relationships like this one exist among the values of the Hilbert function. If properly harnessed, they can significantly reduce the amount of work in computing the values of the Hilbert function.

Example C.35 (Five Subspaces in $\mathbb{R}^4$). Similar to the above example, we have computed the values of $h_I(5)$ for arrangements of five linear subspaces in $\mathbb{R}^4$. The results are given in Table C.4. In fact, using the numerical method described earlier, we have computed the values of $h_I(5)$ up to five subspaces in $\mathbb{R}^{12}$.

C.4.3 Formulas for the Hilbert Function

In this section, we give a general formula for the Hilbert polynomial of the subspace arrangement $Z_{\mathcal{A}} = S_1 \cup S_2 \cup \cdots \cup S_n$. However, due to limitations of space, we will not be able to give a detailed proof for all the results given here. Interested readers may refer to (Derksen 2007).

Let U be any subset of the set of indices $\underline{n} \doteq \{1, 2, \ldots, n\}$. We define the following ideals:

$$I_U \doteq \bigcap_{u \in U} I(S_u), \quad J_U \doteq \prod_{u \in U} I(S_u). \tag{C.39}$$

If U is empty, we use the convention $I_\emptyset = J_\emptyset = R$. We further define $V_U = \bigcap_{u \in U} S_u$, $d_U = \dim(V_U)$, and $c_U = D - d_U$.

Let us define polynomials $p_U(t)$ recursively as follows. First we define

Table C.4 Values of the Hilbert function $h_I(5)$ for arrangements of five subspaces in $\mathbb{R}^4$.

c_1	c_2	c_3	c_4	c_5	$h_I(5)$
1	1	1	1	1	1
1	1	1	1	2	2
1	1	1	1	3	3
1	1	1	2	2	4
1	1	1	2	3	6
1	1	1	3	3	8
1	1	2	2	2	8
1	1	2	2	3	11
1	1	2	3	3	14
1	1	3	3	3	17
1	2	2	2	2	15
1	2	2	2	3	19
1	2	2	3	3	23
1	2	3	3	3	27
1	3	3	3	3	31
2	2	2	2	2	26
2	2	2	2	3	31
2	2	2	3	3	36
2	2	3	3	3	41
2	3	3	3	3	46
3	3	3	3	3	51

$$p_\emptyset(t) = 1.$$

For $U \neq \emptyset$ and if $p_W(t)$ is already defined for all proper subsets W of U, then $p_U(t)$ is uniquely determined by the following equation:

$$\sum_{W \subseteq U} (-t)^{|W|} p_W(t) \equiv 0 \bmod (1-t)^{c_U}, \quad \deg(p_U(t)) < c_U. \tag{C.40}$$

Here $|W|$ is the number of indices in the set W.

With the above definitions, the Hilbert series of the product ideal J is given by

$$\mathcal{H}(J,t) = \frac{p_{\underline{n}}(t)t^n}{(1-t)^D}. \tag{C.41}$$

That is, the Hilbert series of the product ideal J depends only on the numbers $c_U, U \subseteq \underline{n}$. Thus, the values of the Hilbert function $h_J(i)$ are all combinatorial invariants—invariants that depend only on the values $\{c_U\}$ but not the particular position of the subspaces.

Definition C.36 (Transversal Subspaces)**.** *The subspaces $S_1, S_2, \ldots, S_n$ are called* transversal *if $c_U = \min\left(D, \sum_{u\in U} c_u\right)$ for all $U \subseteq \underline{n}$. In other words, the intersection of any subset of the subspaces has the smallest possible dimension.*

Notice that the notion of "transversality" defined here is less strong than the typical notion of "general position." For instance, according to the above definition, three coplanar lines (through the origin) in $\mathbb{R}^3$ are transversal. However, they are not "in general position."

Theorem C.37 (Hilbert Function of a Transversal Subspace Arrangement)**.** *Suppose that $S_1, S_2, \ldots, S_n$ are transversal. Then $\mathcal{H}(I,t) - f(t)$ and $\mathcal{H}(J,t) - f(t)$ are polynomials in t, where $f(t) = \frac{\prod_{i=1}^{n}\left(1-(1-t)^{c_i}\right)}{(1-t)^D}$.*

Thus, the difference between $\mathcal{H}(I,t)$ and $\mathcal{H}(J,t)$ is also a polynomial. We have the following corollary to the above theorem.

Corollary C.38. *If $S_1, S_2, \ldots, S_n$ are transversal, then $h_I(i) = H_I(i) = h_J(i) = H_J(i)$ for all $i \geq n$. That is, the Hilbert polynomials of both the vanishing ideal I and the product ideal J are the same, and the values of their Hilbert functions agree with the polynomial with $i \geq n$.*

One of the consequences of this corollary is that for transversal subspace arrangements, we must have $I_i = J_i$ for all $i \geq n$. This is a result that we have mentioned earlier, in Section C.3.

Example C.39 (Hilbert Series of Three Lines in $\mathbb{R}^3$). For example, suppose that $Z_{\mathcal{A}}$ is the union of three distinct lines (through the origin) in $\mathbb{R}^3$. Regardless of whether the three lines are coplanar, they are transversal. We have

$$\mathcal{H}(J(Z_{\mathcal{A}}),t) = \frac{7t^3 - 9t^4 + 3t^5}{(1-t)^3} = 7t^3 + 12t^4 + 18t^5 + \cdots.$$

However, one has

$$\mathcal{H}(I(Z_{\mathcal{A}}),t) = \frac{t + t^3 - t^4}{(1-t)^3} = t + 3t^2 + 7t^3 + 12t^4 + 18t^5 + \cdots$$

if the lines are coplanar, and

$$\mathcal{H}(I(Z_{\mathcal{A}}),t) = \frac{3t^2 - 2t^3}{(1-t)^3} = 3t^2 + 7t^3 + 12t^4 + 18t^5 + \cdots$$

if the three lines are not coplanar. Notice that the coefficients of these Hilbert series become the same starting from the term t^3.

Then, using the recursive formula (C.41) of the Hilbert series $\mathcal{H}(J,t)$, we can derive a closed-form formula for the values of the Hilbert function $h_I(i)$ with $i \geq n$:

Corollary C.40 (A Closed-Form Formula for Hilbert Function)**.** *If $S_1, S_2, \ldots, S_n$ are transversal, then*

$$h_I(i) = h_J(i) = \sum_U (-1)^{|U|} \binom{D+i-1-c_U}{D-1-c_U}, \quad i \geq n, \tag{C.42}$$

where $c_U = \sum_{m \in U} c_m$ and the sum is over all index subsets U of $\underline{n}$ for which $c_U < D$.

Example C.41 (Three Subspaces in $\mathbb{R}^4$). Suppose that $Z_{\mathcal{A}} = S_1 \cup S_2 \cup S_3$ is a transversal arrangement in $\mathbb{R}^4$. Let d_1, d_2, d_3 (respectively c_1, c_2, c_3) be the dimensions (respectively codimensions) of S_1, S_2, S_3. We make a table of $h_I(n)$ for $n = 3, 4, 5$:

c_1, c_2, c_3	d_1, d_2, d_3	$h_I(3)$	$h_I(4)$	$h_I(5)$
1, 1, 1	3, 3, 3	1	4	10
1, 1, 2	3, 3, 2	2	7	16
1, 1, 3	3, 3, 1	3	9	19
1, 2, 2	3, 2, 2	4	12	25
1, 2, 3	3, 2, 1	6	15	29
1, 3, 3	3, 1, 1	8	18	33
2, 2, 2	2, 2, 2	8	20	38
2, 2, 3	2, 2, 1	11	24	43
2, 3, 3	2, 1, 1	14	28	48
3, 3, 3	1, 1, 1	17	32	53

Note that the codimensions c_1, c_2, c_3 are almost determined by $h_I(3)$. They are uniquely determined by $h_I(3)$ and $h_I(4)$.

The corollary below is a general result that explains why the codimensions of the subspaces c_1, c_2, c_3 can be uniquely determined by $h_I(3), h_I(4), h_I(5)$ in the above example. The corollary also reveals a strong theoretical connection between the Hilbert function and the algebraic subspace clustering problem.

Corollary C.42 (Subspace Dimensions from the Hilbert Function)**.** *Consider a transversal arrangement of n subspaces. The codimensions $c_1, c_2, \ldots, c_n$ are uniquely determined by the values of the Hilbert function $h_I(i)$ for $i = n, n+1, \ldots, n+D-1$.*

As we have alluded to earlier, in the context of algebraic subspace clustering, these values of the Hilbert function are closely related to the ranks of the embedded data matrix $\boldsymbol{V}_i$ for $i = n, n+1, \ldots, n+D-1$. Thus, knowing these ranks, we should in principle be able to uniquely determine the (co)dimensions of all the individual subspaces. These results suggest that knowing the values of the Hilbert function, one can potentially develop better algorithms for determining the correct subspace arrangement from a given set of data.

C.5 Bibliographic Notes

Subspace arrangements constitute a very special but important class of algebraic sets that have been studied in mathematics for centuries (Björner et al. 2005; Björner 1994; Orlik 1989). The importance as well as the difficulty of studying subspace arrangements can hardly be exaggerated. Different aspects of their properties have been and are still being investigated and exploited in many mathematical fields, including algebraic geometry and topology, combinatorics and complexity theory, and graph and lattice theory. See (Björner 1994) for a general review. Although the results about subspace arrangements are extremely rich and deep, only a few special classes of subspace arrangements have been fully characterized. Nevertheless, thanks to the work of (Derksen 2007), the Hilbert function, Hilbert polynomial, and Hilbert series of the vanishing ideal (and the product ideal) of transversal subspace arrangements have recently become well understood. This appendix gives a brief summary of these theoretical developments. These results have provided a sound theoretical foundation for many of the methods developed in this book for clustering and modeling multiple subspaces.

References

Agarwal, P., & Mustafa, N. (2004). k-means projective clustering. In *ACM Symposium on Principles of Database Systems*.

Agarwal, S., Lim, J., Zelnik-Manor, L., Perona, P., Kriegman, D., & Belongie, S. (2005). Beyond pairwise clustering. In *IEEE Conference on Computer Vision and Pattern Recognition* (Vol. 2, pp. 838–845).

Aggarwal, G., Roy-Chowdhury, A., & Chellappa, R. (2004). A system identification approach for video-based face recognition. In *Proceedings of International Conference on Pattern Recognition* (pp. 23–26).

Akaike, H. (1977). A new look at the statistical model selection. *IEEE Transactions on Automatic Control, 16*(6), 716–723.

Aldroubi, A., Cabrelli, C., & Molter, U. (2008). Optimal non-linear models for sparsity and sampling. *Journal of Fourier Analysis and Applications, 14*(5–6), 793–812.

Aldroubi, A., & Zaringhalam, K. (2009). Nonlinear least squares in $\mathbb{R}^N$. *Acta Applicandae Mathematicae, 107*(1–3), 325–337.

Alessandri, A., & Coletta, P. (2001). Design of Luenberger observers for a class of hybrid linear systems. In *Proceedings of Hybrid Systems: Computation and Control* (pp. 7–18). New York: Springer.

Ali, S., Basharat, A., & Shah, M. (2007). Chaotic invariants for human action recognition. In *Proceedings of International Conference on Computer Vision*.

Amaldi, E., & Kann, V. (1998). On the approximability of minimizing nonzero variables or unsatisfied relations in linear systems. *Theoretical Computer Science, 209*, 237–260.

Anderson, B., & Johnson, R. (1982). Exponential convergence of adaptive identification and control algorithms. *Automatica, 18*(1), 1–13.

Arbelaez, P. (2006). Boundary extraction in natural images using ultrametric contour maps. In *Workshop on Perceptual Organization in Computer Vision*.

Arbelaez, P., Maire, M., Fowlkes, C., & Malik, J. (2009). From contours to regions: An empirical evaluation. In *IEEE Conference on Computer Vision and Pattern Recognition*.

Arora, S., Bhaskara, A., Ge, R., & Ma, T. (2014). Provable bounds for learning some deep representations. In *Proceedings of International Conference on Machine Learning*.

Avidan, S., & Shashua, A. (2000). Trajectory triangulation: 3D reconstruction of moving points from a monocular image sequence. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 22*(4), 348–357.

Ayazoglu, M., Li, B., Dicle, C., Sznaier, M., & Camps, O. (2011). Dynamic subspace-based coordinated multicamera tracking. In *IEEE International Conference on Computer Vision* (pp. 2462–2469)

R. Vidal et al., *Generalized Principal Component Analysis*, Interdisciplinary Applied Mathematics 40, DOI 10.1007/978-0-387-87811-9

Babaali, M., & Egerstedt, M. (2004). Observability of switched linear systems. In *Proceedings of Hybrid Systems: Computation and Control*. New York: Springer.

Bach, F. (2013). Convex relaxations of structured matrix factorizations. arXiv:1309.3117v1.

Bach, F., Mairal, J., & Ponce, J. (2008). *Convex sparse matrix factorizations*. http://arxiv.org/abs/0812.1869

Balluchi, A., Benvenuti, L., Benedetto, M. D., & Sangiovanni-Vincentelli, A. (2002). Design of observers for hybrid systems. In *Proceedings of Hybrid Systems: Computation and Control* (Vol. 2289, pp. 76–89). New York: Springer.

Baraniuk, R. (2007). Compressive sensing. *IEEE Signal Processing Magazine, 24*(4), 118–121.

Barbic, J., Safonova, A., Pan, J.-Y., Faloutsos, C., Hodgins, J. K., & Pollar, N. S. (2004). Segmenting motion capture data into distinct behaviors. In *Graphics Interface*.

Barnett, V., & Lewis, T. (1983). *Outliers in statistical data* (2nd ed.). New York: Wiley.

Basri, R., & Jacobs, D. (2003). Lambertian reflection and linear subspaces. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 25*(2), 218–233.

Beck, A., & Teboulle, M. (2009). A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM Journal on Imaging Sciences, 2*(1), 183–202.

Béjar, B., Zappella, L., & Vidal, R. (2012). Surgical gesture classification from video data. In *Medical Image Computing and Computer Assisted Intervention* (pp. 34–41).

Belhumeur, P., Hespanda, J., & Kriegeman, D. (1997). Eigenfaces vs. Fisherfaces: Recognition using class specific linear projection. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 19*(7), 711–720.

Belhumeur, P., & Kriegman, D. (1998). What is the set of images of an object under all possible illumination conditions? *International Journal of Computer Vision, 28*(3), 1–16.

Belkin, M., & Niyogi, P. (2002). Laplacian eigenmaps and spectral techniques for embedding and clustering. In *Proceedings of Neural Information Processing Systems (NIPS)* (pp. 585–591).

Beltrami, E. (1873). Sulle funzioni bilineari. *Giornale di Mathematiche di Battaglini, 11*, 98–106.

Bemporad, A., Ferrari, G., & Morari, M. (2000). Observability and controllability of piecewise affine and hybrid systems. *IEEE Transactions on Automatic Control, 45*(10), 1864–1876.

Bemporad, A., Garulli, A., Paoletti, S., & Vicino, A. (2003). A greedy approach to identification of piecewise affine models. In *Hybrid systems: Computation and control. Lecture notes in computer science* (pp. 97–112). New York: Springer.

Bemporad, A., Roll, J., & Ljung, L. (2001). Identification of hybrid systems via mixed-integer programming. In *Proceedings of IEEE Conference on Decision & Control* (pp. 786–792).

Benson, H. (1994). Concave minimization: Theory, applications and algorithms. In R. Horst & P. M. Pardalos (Eds.), *Handbook of global optimization* (vol. 2, pp. 43–148), Springer Verlag.

Bertsekas, D. P. (1999). *Nonlinear programming* (2nd ed.). *Optimization and computation* (Vol. 2) Belmont: Athena Scientific.

Bickel, P. J. (1976). Another look at robustness: A review of reviews and some new developments. *Scandinavian Journal of Statistics, 3*(28), 145–168.

Bickel, P. J., & Doksum, K. A. (2000). *Mathematical statistics: Basic ideas and selected topics* (2nd ed.). Upper Saddle River: Prentice Hall.

Billio, M., Monfort, A., & Robert, C. (1999). Bayesian estimation of switching ARMA models. *Journal of Econometrics, 93*(2), 229–255.

Bissacco, A., Chiuso, A., Ma, Y., & Soatto, S. (2001). Recognition of human gaits. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition* (Vol. 2, pp. 52–58).

Björner, A. (1994). Subspace arrangements. In *First European Congress of Mathematics, Vol. I (Paris, 1992). Progress in mathematics* (Vol. 119, pp. 321–370). Basel: Birkhäuser.

Björner, A., Peeva, I., & Sidman, J. (2005). Subspace arrangements defined by products of linear forms. *Journal of the London Mathematical Society, 71*(2), 273–288.

Blake, A., North, B., & Isard, M. (1999). Learning multi-class dynamics. *Advances in Neural Information Processing Systems, 11*, 389–395. Cambridge: MIT Press.

Bochnak, J., Coste, M., & Roy, M. F. (1998). *Real Algebraic Geometry*. New York: Springer.

Bottou, L., & Bengio, J. (1995). Convergence properties of the k-means algorithms. In *Neural Information Processing and Systems*.

Boult, T., & Brown, L. (1991). Factorization-based segmentation of motions. In *IEEE Workshop on Motion Understanding* (pp. 179–186).

Boyd, S., Parikh, N., Chu, E., Peleato, B., & Eckstein, J. (2010). Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends in Machine Learning, 3*(1), 1–122.

Boyd, S., & Vandenberghe, L. (2004). *Convex Optimization*. Cambridge: Cambridge University Press.

Bradley, P. S., & Mangasarian, O. L. (2000). k-plane clustering. *Journal of Global Optimization, 16*(1), 23–32.

Brandt, S. (2002). Closed-form solutions for affine reconstruction under missing data. In *In Proceedings Statistical Methods for Video Processing (ECCV'02 Workshop)*.

Broomhead, D. S., & Kirby, M. (2000). A new approach to dimensionality reduction theory and algorithms. *SIAM Journal of Applied Mathematics, 60*(6), 2114–2142.

Bruckstein, A., Donoho, D., & Elad, M. (2009). From sparse solutions of systems of equations to sparse modeling of signals and images. *SIAM Review, 51*(1), 34–81.

Buchanan, A., & Fitzgibbon, A. (2005). Damped Newton algorithms for matrix factorization with missing data. In *IEEE Conference on Computer Vision and Pattern Recognition* (pp. 316–322).

Burer, S., & Monteiro, R. D. C. (2005). Local minima and convergence in low-rank semidefinite programming. *Mathematical Programming, Series A, 103*(3), 427–444.

Burges, C. (2005). Geometric methods for feature extraction and dimensional reduction - a guided tour. In *The data mining and knowledge discovery handbook* (pp. 59–92). Boston: Kluwer Academic.

Burges, C. J. C. (2010). Dimension reduction: A guided tour. *Foundations and Trends in Machine Learning, 2*(4), 275–365.

Burnham, K. P., & Anderson, D. R. (2002). *Model selection and multimodel inference: A practical information-theoretic approach*. New York: Springer.

Burt, P. J., & Adelson, E. H. (1983). The Laplacian pyramid as a compact image code. *IEEE Transactions on Communications, 31*(4), 532–540.

Cai, J.-F., Candés, E. J., & Shen, Z. (2008). A singular value thresholding algorithm for matrix completion. *SIAM Journal of Optimization, 20*(4), 1956–1982.

Campbell, N. (1978). The influence function as an aid in outlier detection in discriminant analysis. *Applied Statistics, 27*(3), 251–258.

Campbell, R. J. (1980). Robust procedures in multivariate analysis I: Robust covariance analysis. *Applied Statistics, 29*, 231–237.

Candès, E. (2006). Compressive sampling. In *Proceedings of the International Congress of Mathematics*.

Candès, E. (2008). The restricted isometry property and its implications for compressed sensing. *Comptes Rendus Mathematique, 346*(9–10), 589–592.

Candès, E., & Donoho, D. (2002). *New tight frames of curvelets and optimal representations of objects with smooth singularities*. Technical Report. Stanford University.

Candès, E., Li, X., Ma, Y., & Wright, J. (2011). Robust principal component analysis? *Journal of the ACM, 58*(3).

Candès, E., & Plan, Y. (2010). Matrix completion with noise. *Proceedings of the IEEE, 98*(6), 925–936.

Candès, E., & Recht, B. (2009). Exact matrix completion via convex optimization. *Foundations of Computational Mathematics, 9*, 717–772.

Candès, E., & Recht, B. (2011). Simple bounds for low-complexity model reconstruction. *Mathematical Programming Series A, 141*(1–2), 577–589.

Candès, E., & Tao, T. (2005). Decoding by linear programming. *IEEE Transactions on Information Theory, 51*(12), 4203–4215.

Candès, E., & Tao, T. (2010). The power of convex relaxation: Near-optimal matrix completion. *IEEE Transactions on Information Theory, 56*(5), 2053–2080.

Candès, E., & Wakin, M. (2008). An introduction to compressive sampling. *IEEE Signal Processing Magazine, 25*(2), 21–30.

Cattell, R. B. (1966). The scree test for the number of factors. *Multivariate Behavioral Research, 1*, 245–276.

Cetingül, H. E., Wright, M., Thompson, P., & Vidal, R. (2014). Segmentation of high angular resolution diffusion MRI using sparse riemannian manifold clustering. *IEEE Transactions on Medical Imaging, 33*(2), 301–317.

Chan, A., & Vasconcelos, N. (2005a). Classification and retrieval of traffic video using auto-regressive stochastic processes. In *Proceedings of 2005 IEEE Intelligent Vehicles Symposium* (pp. 771–776).

Chan, A., & Vasconcelos, N. (2005b). Mixtures of dynamic textures. In *IEEE International Conference on Computer Vision* (Vol. 1, pp. 641–647).

Chandrasekaran, V., Sanghavi, S., Parrilo, P., & Willsky, A. (2009). Sparse and low-rank matrix decompositions. In *IFAC Symposium on System Identification*.

Chaudhry, R., Ravichandran, A., Hager, G., & Vidal, R. (2009). Histograms of oriented optical flow and binet-cauchy kernels on nonlinear dynamical systems for the recognition of human actions. In *IEEE Conference on Computer Vision and Pattern Recognition*.

Chen, G., Atev, S., & Lerman, G. (2009). Kernel spectral curvature clustering (KSCC). In *Workshop on Dynamical Vision*.

Chen, G., & Lerman, G. (2009a). Foundations of a multi-way spectral clustering framework for hybrid linear modeling. *Foundations of Computational Mathematics, 9*(5), 517–558.

Chen, G., & Lerman, G. (2009b). Spectral curvature clustering (SCC). *International Journal of Computer Vision, 81*(3), 317–330.

Chen, J.-Q., Pappas, T. N., Mojsilovic, A., & Rogowitz, B. E. (2003). Image segmentation by spatially adaptive color and texture features. In *IEEE International Conference on Image Processing*.

Chen, S., Donoho, D., & Saunders, M. (1998). Atomic decomposition by basis pursuit. *SIAM Journal of Scientific Computing, 20*(1), 33–61.

Chung, F. (1997). *Spectral graph theory*. Washington: Conference Board of the Mathematical Sciences.

Cilibrasi, R., & Vitányi, P. M. (2005). Clustering by compression. *IEEE Transactions on Information Theory, 51*(4), 1523–1545.

CMU (2003). MOCAP database. http://mocap.cs.cmu.edu.

Coifman, R., & Wickerhauser, M. (1992). Entropy-based algorithms for best bases selection. *IEEE Transactions on Information Theory, 38*(2), 713–718.

Collins, M., Dasgupta, S., & Schapire, R. (2001). A generalization of principal component analysis to the exponential family. In *Neural Information Processing Systems* (Vol. 14)

Collins, P., & Schuppen, J. V. (2004). Observability of piecewise-affine hybrid systems. In *Proceedings of Hybrid Systems: Computation and Control*. New York: Springer.

Comanicu, D., & Meer, P. (2002). Mean shift: A robust approach toward feature space analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI), 24*, 603–619.

Costeira, J., & Kanade, T. (1998). A multibody factorization method for independently moving objects. *International Journal of Computer Vision, 29*(3), 159–179.

Cour, T., Benezit, F., & Shi, J. (2005). Spectral segmentation with multiscale graph decomposition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Cover, T., & Thomas, J. (1991). *Elements of information theory*. Wiley.

Cox, T. F., & Cox, M. A. A. (1994). *Multidimensional scaling*. London: Chapman and Hall.

Critchley, F. (1985). Influence in principal components analysis. *Biometrika, 72*(3), 627–636.

Davis, C., & Cahan, W. (1970). The rotation of eigenvectors by a pertubation. *SIAM Journal on Numerical Analysis, 7*(1), 1–46.

Davison, M. (1983). *Multidimensional Scaling*. New York: Wiley.

De la Torre, F., & Black, M. J. (2004). A framework for robust subspace learning. *International Journal of Computer Vision, 54*(1), 117–142.

Delsarte, P., Macq, B., & Slock, D. (1992). Signal-adapted multiresolution transform for image coding. *IEEE Transactions on Information Theory, 38*, 897–903.

Dempster, A., Laird, N., & Rubin, D. (1977). Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society B, 39*(1), 1–38.

Deng, W., Lai, M.-J., Peng, Z., & Yin, W. (2013). Parallel multi-block admm with o(1/k) convergence. *UCLA CAM*.

Deng, Y., & Manjunath, B. (2001). Unsupervised segmentation of color-texture regions in images and video. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 23*(8), 800–810.

Derksen, H. (2007). Hilbert series of subspace arrangements. *Journal of Pure and Applied Algebra, 209*(1), 91–98.

DeVore, R. (1998). Nonlinear approximation. *Acta Numerica, 7*, 51–150.

DeVore, R., Jawerth, B., & Lucier, B. (1992). Image compression through wavelet transform coding. *IEEE Transactions on Information Theory, 38*(2), 719–746.

Ding, C., Zha, H., He, X., Husbands, P., & Simon, H. D. (2004). Link analysis: Hubs and authorities on the world wide web. *SIAM Review, 46*(2), 256–268.

Do, M. N., & Vetterli, M. (2002). Contourlets: A directional multiresolution image representation. In *IEEE International Conference on Image Processing*.

Donoho, D. (1995). Cart and best-ortho-basis: A connection. Manuscript.

Donoho, D. (1998). Sparse components analysis and optimal atomic decomposition. *Technical Report, Department of Statistics, Stanford University*.

Donoho, D., & Gavish, M. (2014). The optimal hard threshold for singular values is $4/\sqrt{3}$. *IEEE Transactions on Information Theory, 60*(8), 5040–5053.

Donoho, D., & Grimes, C. (2003). Hessian eigenmaps: Locally linear embedding techniques for high-dimensional data. *National Academy of Sciences, 100*(10), 5591–5596.

Donoho, D. L. (1999). Wedgelets: Nearly-minimax estimation of edges. *Annals of Statistics, 27*, 859–897.

Donoho, D. L. (2005). *Neighborly polytopes and sparse solution of underdetermined linear equations*. Technical Report. Stanford University.

Donoho, D. L. (2006). For most large underdetermined systems of linear equations the minimal ℓ^1-norm solution is also the sparsest solution. *Communications on Pure and Applied Mathematics, 59*(6), 797–829.

Donoho, D. L., & Elad, M. (2003). Optimally sparse representation in general (nonorthogonal) dictionaries via ℓ_1 minimization. *Proceedings of National Academy of Sciences, 100*(5), 2197–2202.

Donoho, D. L., Vetterli, M., DeVore, R., & Daubechies, I. (1998). Data compression and harmonic analysis. *IEEE Transactions on Information Theory, 44*(6), 2435–2476.

Donoser, M., Urschler, M., Hirzer, M., & Bischof, H. (2009). Saliency driven total variation segmentation. In *Proceedings of the International Conference on Computer Vision (ICCV)*.

Doretto, G., Chiuso, A., Wu, Y., & Soatto, S. (2003). Dynamic textures. *International Journal of Computer Vision, 51*(2), 91–109.

Doretto, G., & Soatto, S. (2003). Editable dynamic textures. In *IEEE Conference on Computer Vision and Pattern Recognition* (Vol. II, pp. 137–142).

Doretto, G., & Soatto, S. (2006). Dynamic shape and appearance models. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 28*(12), 2006–2019.

Doucet, A., Logothetis, A., & Krishnamurthy, V. (2000). Stochastic sampling algorithms for state estimation of jump Markov linear systems. *IEEE Transactions on Automatic Control, 45*(1), 188–202.

Duda, R., Hart, P., & Stork, D. (2000). *Pattern Classification* (2nd ed.). Wiley, New York.

Eckart, C., & Young, G. (1936). The approximation of one matrix by another of lower rank. *Psychometrika, 1*, 211–218.

Effros, M., & Chou, P. (1995). Weighted universal transform coding: Universal image compression with the Karhunen-Loéve transform. In *IEEE International Conference on Image Processing* (Vol. 2, pp. 61–64).

Efros, A. A., & Leung, T. K. (1999). Texture synthesis by non-parametric sampling. In *IEEE International Conference on Computer Vision* (pp. 1033–1038). Corfu, Greece.

Eisenbud, D. (1996). *Commutative algebra: With a view towards algebraic geometry*. *Graduate texts in mathematics*. New York: Springer.

Elad, M., & Bruckstein, A. (2001). On sparse signal representations. In *IEEE International Conference on Image Processing*.

Elad, M., & Bruckstein, A. (2002). A generalized uncertainty principle and sparse representation in pairs of bases. *IEEE Transactions on Information Theory, 48*(9), 2558–2567.

Elad, M., Figueiredo, M. A. T., & Ma, Y. (2010). On the role of sparse and redundant representations in image processing. *Proceedings of the IEEE, 98*(6), 972–982.

Elder, J., & Zucker, S. (1996). Computing contour closures. In *Proceedings of the European Conference on Computer Vision (ECCV)*.

Elhamifar, E., Sapiro, G., & Vidal, R. (2012a). Finding exemplars from pairwise dissimilarities via simultaneous sparse recovery. In *Neural Information Processing and Systems*.

Elhamifar, E., Sapiro, G., & Vidal, R. (2012b). See all by looking at a few: Sparse modeling for finding representative objects. In *IEEE Conference on Computer Vision and Pattern Recognition*.

Elhamifar, E., & Vidal, R. (2009). Sparse subspace clustering. In *IEEE Conference on Computer Vision and Pattern Recognition*.

Elhamifar, E., & Vidal, R. (2010). Clustering disjoint subspaces via sparse representation. In *IEEE International Conference on Acoustics, Speech, and Signal Processing*.

Elhamifar, E., & Vidal, R. (2011). Sparse manifold clustering and embedding. In *Neural Information Processing and Systems*.

Elhamifar, E., & Vidal, R. (2013). Sparse subspace clustering: Algorithm, theory, and applications. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 35*(11), 2765–2781.

Ezzine, J., & Haddad, A. H. (1989). Controllability and observability of hybrid systems. *International Journal of Control, 49*(6), 2045–2055.

Favaro, P., Vidal, R., & Ravichandran, A. (2011). A closed form solution to robust subspace estimation and clustering. In *IEEE Conference on Computer Vision and Pattern Recognition*.

Fazel, M., Hindi, H., & Boyd, S. (2003). Log-det heuristic for matrix rank minimization with applications to Hankel and Euclidean distance matrices. In *Proceedings of the American Control Conference* (pp. 2156–2162).

Fei-Fei, L., Fergus, R., & Perona, P. (2004). Learning generative visual models from few training examples: An incremental bayesian approach tested on 101 object categories. In *Workshop on Generative Model Based Vision*.

Felzenszwalb, P. F., & Huttenlocher, D. P. (2004). Efficient graph-based image segmentation. *International Journal of Computer Vision (IJCV), 59*(2), 167–181.

Feng, J., Xu, H., Mannor, S., & Yang, S. (2013). Online PCA for contaminated data. In *NIPS*.

Feng, X., & Perona, P. (1998). Scene segmentation from 3D motion. In *IEEE Conference on Computer Vision and Pattern Recognition* (pp. 225–231).

Ferguson, T. (1961). On the rejection of outliers. In *Proceedings of the Fourth Berkeley Symposium on Mathematical Statistics and Probability*.

Ferrari-Trecate, G., Mignone, D., & Morari, M. (2002). Moving horizon estimation for hybrid systems. *IEEE Transactions on Automatic Control, 47*(10), 1663–1676.

Ferrari-Trecate, G., Muselli, M., Liberati, D., & Morari, M. (2003). A clustering technique for the identification of piecewise affine systems. *Automatica, 39*(2), 205–217.

Feuer, A., Nemirovski, A. (2003). On sparse representation in pairs of bases. *IEEE Transactions on Information Theory, 49*(6), 1579–1581.

Figueiredo, M. A. T., & Jain, A. K. (2002). Unsupervised learning of finite mixture models. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 24*(3), 381–396.

Fischler, M. A., & Bolles, R. C. (1981). RANSAC random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM, 26*, 381–395.

Fisher, Y. (1995). *Fractal Image Compression: Theory and Application*. Springer-Verlag Telos.

Fitzgibbon, A., & Zisserman, A. (2000). Multibody structure and motion: 3D reconstruction of independently moving objects. In *European Conference on Computer Vision* (pp. 891–906).

Forgy, E. (1965). Cluster analysis of multivariate data: Efficiency vs. interpretability of classifications (abstract). *Biometrics, 21*, 768–769.

Freixenet, J., Munoz, X., Raba, D., Marti, J., & Cuff, X. (2002). Yet another survey on image segmentation. In *Proceedings of the European Conference on Computer Vision (ECCV)*.

Frey, B., Colmenarez, A., & Huang, T. (1998). Mixtures of local linear subspaces for face recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*.

Gabriel, K. R. (1978). Least squares approximation of matrices by additive and multiplicative models. *Journal of the Royal Statistical Society B, 40*, 186–196.

Ganesh, A., Wright, J., Li, X., Candès, E., & Ma, Y. (2010). Dense error correction for low-rank matrices via principal component pursuit. In *International Symposium on Information Theory*.

Geman, S., & McClure, D. (1987). Statistical methods for tomographic image reconstruction. In *Proceedings of the 46th Session of the ISI, Bulletin of the ISI* (Vol. 52, pp. 5–21).

Georghiades, A., Belhumeur, P., & Kriegman, D. (2001). From few to many: Illumination cone models for face recognition under variable lighting and pose. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 23*(6), 643–660.

Gersho, A., & Gray, R. M. (1992). *Vector Quantization and Signal Compression*. Boston: Kluwer Academic.

Gevers, T., & Smeulders, A. (1997). Combining region splitting and edge detection through guided Delaunay image subdivision. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Ghahramani, Z., & Beal, M. (2000). Variational inference for Bayesian mixtures of factor analysers. *Advances in Neural Information Processing Systems, 12*, 449–455.

Ghahramani, Z., & Hinton, G. (1996). The EM algorithm for mixtures of factor analyzers. *Technical Report CRG-TR-96-1, University of Toronto, Canada*.

Ghahramani, Z., & Hinton, G. E. (1998). Variational learning for switching state-space models. *Neural Computation, 12*(4), 963–996.

Ghoreyshi, A., & Vidal, R. (2007). Epicardial segmentation in dynamic cardiac MR sequences using priors on shape, intensity, and dynamics, in a level set framework. In *IEEE International Symposium on Biomedical Imaging* (pp. 860–863).

Gnanadesikan, R., & Kettenring, J. (1972). Robust estimates, residuals, and outlier detection with multiresponse data. *Biometrics, 28*(1), 81–124.

Goh, A., & Vidal, R. (2007). Segmenting motions of different types by unsupervised manifold clustering. In *IEEE Conference on Computer Vision and Pattern Recognition*.

Goh, A., & Vidal, R. (2008). Unsupervised Riemannian clustering of probability density functions. In *European Conference on Machine Learning*.

Goldfarb, D., & Ma, S. (2009). Convergence of fixed point continuation algorithms for matrix rank minimization. *Preprint*.

Golub, H., & Loan, C. V. (1996). *Matrix Computations* (2nd ed.). Baltimore: Johns Hopkins University Press.

Govindu, V. (2005). A tensor decomposition for geometric grouping and segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition* (pp. 1150–1157).

Gower, J. (1966). Some distance properties of latent root and vector methods used in multivariate analysis. *Biometrika, 53*, 325–338.

Gross, D. (2011). Recovering low-rank matrices from few coefficients in any basis. *IEEE Trans on Information Theory, 57*(3), 1548–1566.

Gruber, A., & Weiss, Y. (2004). Multibody factorization with uncertainty and missing data using the EM algorithm. In *IEEE Conference on Computer Vision and Pattern Recognition* (Vol. I, pp. 707–714).

H.Aanaes, Fisker, R., Astrom, K., & Carstensen, J. M. (2002). Robust factorization. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 24*(9), 1215–1225.

Haeffele, B., & Vidal, R. (2015). Global optimality in tensor factorization, deep learning, and beyond. *Preprint, http://arxiv.org/abs/1506.07540*.

Haeffele, B., Young, E., & Vidal, R. (2014). Structured low-rank matrix factorization: Optimality, algorithm, and applications to image processing. In *International Conference on Machine Learning*.

Hamkins, J., & Zeger, K. (2002). Gaussian source coding with spherical codes. *IEEE Transactions on Information Theory, 48*(11), 2980–2989.

Hampel, F., Ronchetti, E., Rousseeuw, P., & Stahel, W. (1986). *Robust statistics: The approach based on influence functions*. New York: Wiley.

Hampel, F. R. (1974). The influence curve and its role in robust estiamtion. *Journal of the American Statistical Association, 69*, 383–393.

Han, M., & Kanade, T. (2000). Reconstruction of a scene with multiple linearly moving objects. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition* (Vol. 2, pp. 542–549).

Han, M., & Kanade, T. (2001). Multiple motion scene reconstruction from uncalibrated views. In *Proceedings of IEEE International Conference on Computer Vision* (Vol. 1, pp. 163–170).

Hansen, M., & Yu, B. (2001). Model selection and the principle of minimum description length. *Journal of American Statistical Association, 96*, 746–774.

Haralick, R., & Shapiro, L. (1985). Image segmentation techniques. *Computer Vision, Graphics, and Image Processing, 29*(1), 100–132.

Hardt, M. (2014). Understanding alternating minimization for matrix completion. In *Symposium on Foundations of Computer Science*.

Haro, G., Randall, G., & Sapiro, G. (2006). Stratification learning: Detecting mixed density and dimensionality in high dimensional point clouds. In *Neural Information Processing and Systems*.

Haro, G., Randall, G., & Sapiro, G. (2008). Translated poisson mixture model for stratification learning. *International Journal of Computer Vision, 80*(3), 358–374.

Harris, J. (1992). *Algebraic Geometry: A First Course*. New York: Springer.

Hartley, R., & Schaffalitzky, F. (2003). Powerfactorization: An approach to affine reconstruction with missing and uncertain data. In *Proceedings of Australia-Japan Advanced Workshop on Computer Vision*.

Hartley, R., & Vidal, R. (2004). The multibody trifocal tensor: Motion segmentation from 3 perspective views. In *IEEE Conference on Computer Vision and Pattern Recognition* (Vol. I, pp. 769–775).

Hartley, R., & Zisserman, A. (2004). *Multiple view geometry in computer vision* (2nd ed.). Cambridge: Cambridge University Press.

Hastie, T. (1984). Principal curves and surfaces. *Technical Report, Stanford University*.

Hastie, T., & Stuetzle, W. (1989). Principal curves. *Journal of the American Statistical Association, 84*(406), 502–516.

Hastie, T., Tibshirani, R., & Friedman, J. (2001). *The Elements of Statistical Learning*. New York: Springer.

He, H., & Garcia, E. (2009). Learning from imbalanced data. *IEEE Transactions on Knowledge and Data Engineering, 21*(9), 1263–1284.

He, H., & Ma, Y. (2013). *Imbalanced Learning: Foundations, Algorithms, and Applications*. New York: Wiley.

Hinton, G., Osindero, S., & Teh, Y.-W. (2006). A fast learning algorithm for deep belief nets. *Neural Computation, 18*(7), 1527–1554.

Hirsch, M. (1976). *Differential Topology*. New York: Springer.

Ho, J., Yang, M., Lim, J., Lee, K., & Kriegman, D. (2003). Clustering appearances of objects under varying illumination conditions. In *Proceedings of International Conference on Computer Vision and Pattern Recognition*.

Hong, W., Wright, J., Huang, K., & Ma, Y. (2006). Multi-scale hybrid linear models for lossy image representation. *IEEE Transactions on Image Processing, 15*(12), 3655–3671.

Horn, R. A., & Johnson, C. R. (1985). *Matrix Analysis*. Cambridge: Cambridge University Press.

Hotelling, H. (1933). Analysis of a complex of statistical variables into principal components. *Journal of Educational Psychology, 24*, 417–441.

Householder, A. S., & Young, G. (1938). Matrix approximation and latent roots. *American Mathematical Monthly, 45*, 165–171.
Huang, K., Ma, Y., & Vidal, R. (2004). Minimum effective dimension for mixtures of subspaces: A robust GPCA algorithm and its applications. In *IEEE Conference on Computer Vision and Pattern Recognition* (Vol. II, pp. 631–638).
Huber, P. (1981). *Robust Statistics*. New York: Wiley.
Hubert, L., Meulman, J., & Heiser, W. (2000). Two purposes for matrix factorization: A historical appraisal. *SIAM Review, 42*(1), 68–82.
Hwang, I., Balakrishnan, H., & Tomlin, C. (2003). Observability criteria and estimator design for stochastic linear hybrid systems. In *Proceedings of European Control Conference*.
Hyndman, M., Jepson, A., & Fleet, D. J. (2007). Higher-order autoregressive models for dynamic textures. In *British Machine Vision Conference* (pp. 76.1–76.10). doi:10.5244/C.21.76.
Jacobs, D. (2001). Linear fitting with missing data: Applications to structure-from-motion. *Computer Vision and Image Understanding, 82*, 57–81.
Jain, A. (1989). *Fundamentals of Digital Image Processing*. Upper Saddle River: Prentice Hall.
Jain, P., Meka, R., & Dhillon, I. (2010). Guaranteed rank minimization via singular value projection. In *Neural Information Processing Systems* (pp. 937–945).
Jain, P., & Netrapalli, P. (2014). Fast exact matrix completion with finite samples. In http://arxiv.org/pdf/1411.1087v1.pdf.
Jain, P., Netrapalli, P., & Sanghavi, S. (2012). Low-rank matrix completion using alternating minimization. In http://arxiv.org/pdf/1411.1087v1.pdf.
Jancey, R. (1966). Multidimensional group analysis. *Australian Journal of Botany, 14*, 127–130.
Jarret, K., Kavukcuoglu, K., Ranzato, M., & LeCun, Y. (2009). What is the best multi-stage architecture for object recognition. In *International Conference on Computer Vision*.
Jhuo, I.-H., Liu, D., Lee, D., & Chang, S.-F. (2012). Robust visual domain adaptation with low-rank reconstruction. In *IEEE Conference on Computer Vision and Pattern Recognition* (pp. 2168–2175).
Johnson, C. (1990). Matrix completion problems: A survey. In *Proceedings of Symposia in Applied Mathematics*.
Jolliffe, I. (1986). *Principal Component Analysis*. New York: Springer.
Jolliffe, I. (2002). *Principal Component Analysis* (2nd ed.). New York: Springer.
Jordan, M. (1874). Mémoire sur les formes bilinéaires. *Journal de Mathématiques Pures et Appliqués, 19*, 35–54.
Juloski, A., Heemels, W., & Ferrari-Trecate, G. (2004). Data-based hybrid modelling of the component placement process in pick-and-place machines. In *Control Engineering Practice*. Amsterdam: Elsevier.
Kamvar, S., Klein, D., & Manning, C. (2002). Interpreting and extending classical agglomerative clustering methods using a model-based approach. *Technical Report 2002-11, Stanford University Department of Computer Science*.
Kanatani, K. (1998). Geometric information criterion for model selection. *International Journal of Computer Vision* (pp. 171–189).
Kanatani, K. (2001). Motion segmentation by subspace separation and model selection. In *IEEE International Conference on Computer Vision* (Vol. 2, pp. 586–591).
Kanatani, K. (2002). Evaluation and selection of models for motion segmentation. In *Asian Conference on Computer Vision* (pp. 7–12).
Kanatani, K. (2003). How are statistical methods for geometric inference justified? In *Workshop on Statistical and Computational Theories of Vision, IEEE International Conference on Computer Vision*.
Kanatani, K., & Matsunaga, C. (2002). Estimating the number of independent motions for multibody motion segmentation. In *European Conference on Computer Vision* (pp. 25–31).
Kanatani, K., & Sugaya, Y. (2003). Multi-stage optimization for multi-body motion segmentation. In *Australia-Japan Advanced Workshop on Computer Vision* (pp. 335–349).
Ke, Q., & Kanade, T. (2005). Robust ℓ^1-norm factorization in the presence of outliers and missing data. In *IEEE Conference on Computer Vision and Pattern Recognition*.

Keshavan, R., Montanari, A., & Oh, S. (2010a). Matrix completion from a few entries. *IEEE Transactions on Information Theory*.

Keshavan, R., Montanari, A., & Oh, S. (2010b). Matrix completion from noisy entries. *Journal of Machine Learning Research, 11*, 2057–2078.

Keshavan, R. H. (2012). *Efficient algorithms for collaborative filtering*. Ph.D. Thesis. Stanford University.

Kim, J., Fisher, J., Yezzi, A., Cetin, M., & Willsky, A. (2005). A nonparametric statistical method for image segmentation using information theory and curve evolution. *PAMI, 14*(10), 1486–1502.

Kim, S. J., Doretto, G., Rittscher, J., Tu, P., Krahnstoever, N., & Pollefeys, M. (2009). A model change detection approach to dynamic scene modeling. In *Sixth IEEE International Conference on Advanced Video and Signal Based Surveillance, 2009 (AVSS '09)* (pp. 490–495).

Kim, S. J., Koh, K., Lustig, M., Boyd, S., & Gorinevsky, D. (2007). An interior-point method for large-scale l1-regularized least squares. *IEEE Journal on Selected Topics in Signal Processing, 1*(4), 606–617.

Kim, T., Lee, K., & Lee, S. (2010). Learning full pairwise affinities for spectral segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Kleinberg, J. M. (1999). Authorative sources in a hyberlinked environment. *Journal of the ACM, 48*, 604–632.

Kontogiorgis, S., & Meyer, R. (1989). A variable-penalty alternating direction method for convex optimization. *Mathematical Programming, 83*, 29–53.

Kruskal, J. (1964). Nonmetric multidimensional scaling: A numerical method. *Psychometrika*.

Kurita, T. (1995). An efficient clustering algorithm for region merging. *IEICE Transactions of Information and Systems, E78-D*(12), 1546–1551.

Lanczos, C. (1950). An iteration method for the solution of the eigenvalue problem of linear differential and integral operators. *Journal of Research of the National Bureau of Standards, 45*, 255–282.

Lang, S. (1993). *Algebra* (3rd ed.). Reading: Addison-Wesley.

Lee, J. A., & Verleysen, M. (2007). *Nonlinear Dimensionality Reduction* (1st ed.). New York: Springer.

Lee, K.-C., Ho, J., & Kriegman, D. (2005). Acquiring linear subspaces for face recognition under variable lighting. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 27*(5), 684–698.

Leonardis, A., Bischof, H., & Maver, J. (2002). Multiple eigenspaces. *Pattern Recognition, 35*(11), 2613–2627.

LePennec, E., & Mallat, S. (2005). Sparse geometric image representation with bandelets. *IEEE Transactions on Image Processing, 14*(4), 423–438.

Levina, E., & Bickel, P. J. (2006). Texture synthesis and non-parametric resampling of random fields. *Annals of Statistics, 34*(4), 1751–1773.

Li, B., Ayazoglu, M., Mao, T., Camps, O. I., & Sznaier, M. (2011). Activity recognition using dynamic subspace angles. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition* (pp. 3193–3200). New York: IEEE.

Lin, Z., Chen, M., Wu, L., & Ma, Y. (2011). The augmented Lagrange multiplier method for exact recovery of corrupted low-rank matrices. arXiv:1009.5055v2.

Lions, P., & Mercier, B. (1979). Splitting algorithms for the sum of two nonlinear operators. *SIAM Journal on Numerical Analysis, 16*(6), 964–979.

Liu, G., Lin, Z., Yan, S., Sun, J., & Ma, Y. (2013). Robust recovery of subspace structures by low-rank representation. *IEEE Trans. Pattern Analysis and Machine Intelligence, 35*(1), 171–184.

Liu, G., Lin, Z., & Yu, Y. (2010). Robust subspace segmentation by low-rank representation. In *International Conference on Machine Learning*.

Liu, Y. K., & Zalik, B. (2005). Efficient chain code with Huffman coding. *Pattern Recognition, 38*(4), 553–557.

Lloyd, S. (1957). *Least squares quantization in PCM*. Technical Report. Bell Laboratories. Published in 1982 in *IEEE Transactions on Information Theory, 28*, 128–137.

Luenberger, D. G. (1973). *Linear and Nonlinear Programming*. Reading: Addison-Wesley.
Luo, Z. Q., & Tseng, P. (1993). One the convergence rate of dual ascent methods for strictly convex minimization. *Mathematics of Operations Research, 18*, 846–867.
Ma, S. (2012). *Alternating proximal gradient method for convex minimization*. Technical Report.
Ma, Y., Derksen, H., Hong, W., & Wright, J. (2007). Segmentation of multivariate mixed data via lossy coding and compression. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 29*(9), 1546–1562.
Ma, Y., Soatto, S., Kosecka, J., & Sastry, S. (2003). *An Invitation to 3D Vision: From Images to Geometric Models*. New York: Springer.
Ma, Y., & Vidal, R. (2005). Identification of deterministic switched ARX systems via identification of algebraic varieties. In *Hybrid Systems: Computation and Control* (pp. 449–465). New York: Springer.
Ma, Y., Yang, A. Y., Derksen, H., & Fossum, R. (2008). Estimation of subspace arrangements with applications in modeling and segmenting mixed data. *SIAM Review, 50*(3), 413–458.
MacQueen, J. (1967). Some methods for classification and analysis of multivariate observations. In *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability* (pp. 281–297).
Madiman, M., Harrison, M., & Kontoyiannis, I. (2004). Minimum description length vs. maximum likelihood in lossy data compression. In *Proceedings of the 2004 IEEE International Symposium on Information Theory*.
Malik, J., Belongie, S., Leung, T., & Shi, J. (2001). Contour and texture analysis for image segmentation. *International Journal of Computer Vision, 43*(1), 7–27.
Mallat, S. (1999). *A Wavelet Tour of Signal Processing* (2nd ed.). London: Academic.
Mallows, C. (1973). Some comments on C_p. *Technometrics, 15*, 661–675.
Maronna, R. A. (1976). Robust M-estimators of multivariate location and scatter. *Annals of Statistics, 4*, 51–67.
Martin, D., Fowlkes, C., Tal, D., & Malik, J. (2001). A Database of Human Segmented Natural Images and its Application to Evaluating Segmentation Algorithms and Measuring Ecological Statistics. In *IEEE International Conference on Computer Vision*.
McLanchlan, G. J., & Krishnan, T. (1997). *The EM Algorithms and Extentions*. Wiley Series in Probability and Statistics. John Wiley & Sons, Inc.
Meila, M. (2005). Comparing clusterings: An axiomatic view. In *Proceedings of the International Conference on Machine Learning*.
Mercer, J. (1909). Functions of positive and negative types and their connection with the theory of integral equations. *Philosophical Transactions, Royal Society London, A, 209*(1909), 415–446.
Meyer, F. (2000). Fast adaptive wavelet packet image compression. *IEEE Transactions on Image Processing, 9*(5), 792–800.
Meyer, F. (2002). Image compression with adaptive local cosines. *IEEE Transactions on Image Processing, 11*(6), 616–629.
Minka, T. (2000). Automatic choice of dimensionality for PCA. In *Neural Information Processing Systems* (Vol. 13, pp. 598–604).
Mirsky, L. (1975). A trace inequality of John von Neumann. *Monatshefte für Mathematic, 79*, 303–306.
Mobahi, H., Rao, S., Yang, A., & Sastry, S. (2011). Segmentation of natural images by texture and boundary compression. *International Journal of Computer Vision, 95*(1), 86–98.
Mori, G., Ren, X., Efros, A., & Malik, J. (2004). Recovering human body configurations: Combining segmentation and recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*.
Muresan, D., & Parks, T. (2003). Adaptive principal components and image denoising. In *IEEE International Conference on Image Processing*.
Murphy, K. (1998). *Switching Kalman filters*. Technical Report. U.C. Berkeley.
Nascimento, J. C., Figueiredo, M. A. T., & Marques, J. S. (2005). Recognition of human activities using space dependent switched dynamical models. In *IEEE International Conference on Image Processing* (pp. 852–855).

Neal, R., & Hinton, G. (1998). A view of the EM algorithm that justifies incremental, sparse, and other variants. In M. Jordan (Ed.), *Learning in graphical models* (pp. 355–368). Boston: Kluwer Academic.

Negahban, S., Ravikumar, P., Wainwright, M., & Yu, B. (2010). A unified framework for analyzing m-estimators with decomposible regularizers. Available at http://arxiv.org/abs/1010.2731v1.

Nemirovskii, A. S., & Yudin, D. B. (1979). *Complexity of problems and efficiency of optimization methods* (in Russian). Moscow: Nauka.

Nesterov, Y. (1983). A method of solving a convex programming problem with convergence rate $O(1/k^2)$. *Soviet Mathematics Doklady, 27*(2), 372–376.

Ng, A., Weiss, Y., & Jordan, M. (2001). On spectral clustering: Analysis and an algorithm. In *Proceedings of Neural Information Processing Systems (NIPS)* (pp. 849–856).

Niessen, H., & A.Juloski (2004). Comparison of three procedures for identification of hybrid systems. In *Conference on Control Applications*.

Nunez, F., & Cipriano, A. (2009). Visual information model based predictor for froth speed control in flotation process. *Minerals Engineering, 22*(4), 366–371.

Ofli, F., Chaudhry, R., Kurillo, G., Vidal, R., & Bajcsy, R. (2013). Berkeley MHAD: A comprehensive multimodal human action database. In *IEEE Workshop on Applications of Computer Vision*.

Olshausen, B., & D.J.Field (1996). Emergence of simple-cell receptive field properties by learning a sparse code for natural images. *Nature, 381*(6583), 607–609.

Orlik, P. (1989). *Introduction to Arrangements*. *Conference Board of the Mathematical Sciences Regional Conference Series in Mathematics* (Vol. 72). Providence: American Mathematics Society.

Overschee, P. V., & Moor, B. D. (1993). Subspace algorithms for the stochastic identification problem. *Automatica, 29*(3), 649–660.

Patel, V. M., Gopalan, R., Li, R., & Chellappa, R. (2014). Visual domain adaptation: A survey of recent advances. *IEEE Signal Processing Magazine, 32*(3), 53–69.

Pavlovic, V., Moulin, P., & Ramchandran, K. (1998). An integrated framework for adaptive subband image coding. *IEEE Transactions on Signal Processing, 47*(4), 1024–1038.

Pavlovic, V., Rehg, J. M., Cham, T. J., & Murphy, K. P. (1999). A dynamic Bayesian network approach to figure tracking using learned dynamic models. In *Proceedings of the International Conference on Computer Vision* (pp. 94–101).

Pearson, K. (1901). On lines and planes of closest fit to systems of points in space. *The London, Edinburgh and Dublin Philosphical Magazine and Journal of Science, 2*, 559–572.

Peng, Z., Yan, M., & Yin, W. (2013). Parallel and distributed sparse optimization. In *Asilomar*.

Polito, M., & Perona, P. (2002). Grouping and dimensionality reduction by locally linear embedding. In *Proceedings of Neural Information Processing Systems (NIPS)*.

Powell, M. J. D. (1973). On search directions for minimization algorithms. *Mathematical Programming, 4*, 193–201.

Qiu, Q., Patel, V. M., Turaga, P., & Chellappa, R. (2012). Domain adaptive dictionary learning. In *European Conference on Computer Vision* (Vol. 7575, pp. 631–645).

Rabiee, H., Kashyap, R., & Safavian, S. (1996). Adaptive multiresolution image coding with matching and basis pursuits. In *IEEE International Conference on Image Processing*.

Rahimi, A., Darrell, T., & Recht, B. (2005). Learning appearance manifolds from video. In *IEEE Conference on Computer Vision and Pattern Recognition* (Vol. 1, pp. 868–875).

Ramchandran, K., & Vetterli, M. (1993). Best wavelet packets bases in a rate-distortion sense. *IEEE Transactions on Image Processing, 2*, 160–175.

Ramchandran, K., Vetterli, M., & Herley, C. (1996). Wavelets, subband coding, and best basis. *Proceedings of the IEEE, 84*(4), 541–560.

Rand, W. M. (1971). Objective criteria for the evaluation of clustering methods. *Journal of the American Statistical Association, 66*(336), 846–850.

Rao, S., Mobahi, H., Yang, A., & Sastry, S. (2009). Natural image segmentation with adaptive texture and boundary encoding. In *Asian Conference on Computer Vision, 1* (pp. 135–146).

Rao, S., Tron, R., Ma, Y., & Vidal, R. (2008). Motion segmentation via robust subspace separation in the presence of outlying, incomplete, or corrupted trajectories. In *IEEE Conference on Computer Vision and Pattern Recognition*.

Rao, S., Tron, R., Vidal, R., & Ma, Y. (2010). Motion segmentation in the presence of outlying, incomplete, or corrupted trajectories. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 32*(10), 1832–1845.

Rao, S., Yang, A. Y., Wagner, A., & Ma, Y. (2005). Segmentation of hybrid motions via hybrid quadratic surface analysis. In *IEEE International Conference on Computer Vision* (pp. 2–9).

Ravichandran, A., Chaudhry, R., & Vidal, R. (2009). View-invariant dynamic texture recognition using a bag of dynamical systems. In *IEEE Conference on Computer Vision and Pattern Recognition*.

Ravichandran, A., Chaudhry, R., & Vidal, R. (2013). Categorizing dynamic textures using a bag of dynamical systems. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 35*(2), 342–353.

Ravichandran, A., & Vidal, R. (2008). Video registration using dynamic textures. In *European Conference on Computer Vision*.

Ravichandran, A., & Vidal, R. (2011). Video registration using dynamic textures. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 33*(1), 158–171.

Ravichandran, A., Vidal, R., & Halperin, H. (2006). Segmenting a beating heart using polysegment and spatial GPCA. In *IEEE International Symposium on Biomedical Imaging* (pp. 634–637).

Recht, B., Fazel, M., & Parrilo, P. (2010). Guaranteed minimum-rank solutions of linear matrix equations via nuclear norm minimization. *SIAM Review, 52*(3), 471–501.

Ren, X., Fowlkes, C., & Malik, J. (2005). Scale-invariant contour completion using condition random fields. In *IEEE International Conference on Computer Vision*.

Ren, X., Fowlkes, C., & Malik, J. (2008). Learning probabilistic models for contour completion in natural images. *International Journal of Computer Vision, 77*, 47–63.

Rissanen, J. (1978). Modeling by shortest data description. *Automatica, 14*, 465–471.

Rose, K. (1998). Deterministic annealing for clustering, compression, classification, regression, and related optimization problems. *Proceedings of the IEEE, 86*(11), 2210–2239.

Rousseeuw, P. (1984). Least median of squares regression. *Journal of American Statistics Association, 79*, 871–880.

Roweis, S., & Saul, L. (2000). Nonlinear dimensionality reduction by locally linear embedding. *Science, 290*(5500), 2323–2326.

Roweis, S., & Saul, L. (2003). Think globally, fit locally: Unsupervised learning of low dimensional manifolds. *Journal of Machine Learning Research, 4*, 119–155.

Saisan, P., Bissacco, A., Chiuso, A., & Soatto, S. (2004). Modeling and synthesis of facial motion driven by speech. In *European Conference on Computer Vision* (Vol. 3, pp. 456–467).

Santis, E., Benedetto, M. D., & Giordano, P. (2003). On observability and detectability of continuous-time linear switching systems. In *Proceedings of IEEE Conference on Decision & Control* (pp. 5777–5782).

Schindler, K., & Suter, D. (2005). Two-view multibody structure-and-motion with outliers. In *IEEE Conference on Computer Vision and Pattern Recognition*.

Schölkopf, B., & Smola, A. (2002). *Learning with kernels*. Cambridge: MIT Press.

Schölkopf, B., Smola, A., & Muller, K. R. (1998). Nonlinear component analysis as a kernel eigenvalue problem. *Neural Computation, 10*, 1299–1319.

Selim, S., & Ismail, M. A. (1984). K-means-type algorithms: A generalized convergence theorem and characterization of local optimality. *IEEE Transaction on Pattern Analysis and Machine Intelligence, 6*(1), 81–87.

Sha, F., & Saul, L. (2005). Analysis and extension of spectral methods for nonlinear dimensionality reduction. In *Proceedings of International Conference on Machine Learning* (pp. 784–791).

Shabalin, A., & Nobel, A. (2010). *Reconstruction of a low-rank matrix in the presence of gaussian noise* (pp. 1–34). arXiv preprint 1007.4148

Shakernia, O., Vidal, R., & Sastry, S. (2003). Multi-body motion estimation and segmentation from multiple central panoramic views. In *IEEE International Conference on Robotics and Automation* (Vol. 1, pp. 571–576).

Shapiro, J. M. (1993). Embedded image coding using zerotrees of wavelet coefficients. *IEEE Transactions on Signal Processing, 41*(12), 3445–3463.

Shashua, A., & Levin, A. (2001). Multi-frame infinitesimal motion model for the reconstruction of (dynamic) scenes with multiple linearly moving objects. In *Proceedings of IEEE International Conference on Computer Vision* (Vol. 2, pp. 592–599).

Shekhar, S., Patel, V. M., Nguyen, H. V., & Chellappa, R. (2013). Generalized domain-adaptive dictionaries. In *IEEE Conference on Computer Vision and Pattern Recognition*.

Shi, J., & Malik, J. (1998). Motion segmentation and tracking using normalized cuts. In *IEEE International Conference on Computer Vision* (pp. 1154–1160).

Shi, J., & Malik, J. (2000). Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 22*(8), 888–905.

Shi, T., Belkin, M., & Yin, B. (2008). Data spectroscopy: Eigenspace of convolution operators and clustering. *arXiv:0807.3719v1*.

Shizawa, M., & Mase, K. (1991). A unified computational theory for motion transparency and motion boundaries based on eigenenergy analysis. In *IEEE Conference on Computer Vision and Pattern Recognition* (pp. 289–295).

Shum, H.-Y., Ikeuchi, K., & Reddy, R. (1995). Principal component analysis with missing data and its application to polyhedral object modeling. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 17*(9), 854–867.

Sikora, T., & Makai, B. (1995). Shape-adaptive DCT for generic coding of video. *IEEE Transactions on Circuits and Systems For Video Technology, 5*, 59–62.

Soltanolkotabi, M., & Candès, E. J. (2013). A geometric analysis of subspace clustering with outliers. *Annals of Statistics, 40*(4), 2195–2238.

Soltanolkotabi, M., Elhamifar, E., & Candès, E. J. (2014). Robust subspace clustering. *Annals of Statistics, 42*(2), 669–699.

Souvenir, R., & Pless, R. (2005). Manifold clustering. In *Proceedings of International Conference on Computer Vision* (Vol. I, pp. 648–653).

Spielman, D., Wang, H., & Wright, J. (2012). Exact recovery of sparsity-used dictionaries. *Conference on Learning Theory (COLT)*.

Starck, J.-L., Elad, M., & Donoho, D. (2003). Image decomposition: Separation of texture from piecewise smooth content. In *Proceedings of the SPIE* (Vol. 5207, pp. 571–582).

Steward, C. V. (1999). Robust parameter estimation in computer vision. *SIAM Review, 41*(3), 513–537.

Sturm, P. (2002). Structure and motion for dynamic scenes - the case of points moving in planes. In *Proceedings of European Conference on Computer Vision* (pp. 867–882).

Sun, A., Ge, S. S., & Lee, T. H. (2002). Controllability and reachability criteria for switched linear systems. *Automatica, 38*, 775–786.

Sun, J., Qu, Q., & Wright, J. (2015). Complete dictionary recovery over the sphere. Preprint. http://arxiv.org/abs/1504.06785

Szigeti, F. (1992). A differential algebraic condition for controllability and observability of time varying linear systems. In *Proceedings of IEEE Conference on Decision and Control* (pp. 3088–3090).

Szummer, M., & Picard, R. W. (1996). Temporal texture modeling. In *IEEE International Conference on Image Processing* (Vol. 3, pp. 823–826).

Taubin, G. (1991). Estimation of planar curves, surfaces, and nonplanar space curves defined by implicit equations with applications to edge and range image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 13*(11), 1115–1138.

Tenenbaum, J. B., de Silva, V., & Langford, J. C. (2000). A global geometric framework for nonlinear dimensionality reduction. *Science, 290*(5500), 2319–2323.

Tibshirani, R. (1996). Regression shrinkage and selection via the LASSO. *Journal of the Royal Statistical Society B, 58*(1), 267–288.

Tipping, M., & Bishop, C. (1999a). Mixtures of probabilistic principal component analyzers. *Neural Computation, 11*(2), 443–482.

Tipping, M., & Bishop, C. (1999b). Probabilistic principal component analysis. *Journal of the Royal Statistical Society, 61*(3), 611–622.

Torgerson, W. (1958). *Theory and Methods of Scaling*. New York: Wiley.

Torr, P., & Davidson, C. (2003). IMPSAC: Synthesis of importance sampling and random sample consensus. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 25*(3), 354–364.

Torr, P., Szeliski, R., & Anandan, P. (2001). An integrated Bayesian approach to layer extraction from image sequences. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 23*(3), 297–303.

Torr, P. H. S. (1998). Geometric motion segmentation and model selection. *Philosophical Transactions of the Royal Society of London, 356*(1740), 1321–1340.

Tremeau, A., & Borel, N. (1997). A region growing and merging algorithm to color segmentation. *Pattern Recognition, 30*(7), 1191–1204.

Tron, R., & Vidal, R. (2007). A benchmark for the comparison of 3-D motion segmentation algorithms. In *IEEE Conference on Computer Vision and Pattern Recognition*.

Tse, D., & Viswanath, P. (2005). *Fundamentals of Wireless Communications*. Cambridge: Cambridge University Press.

Tseng, P. (2000). Nearest q-flat to m points. *Journal of Optimization Theory and Applications, 105*(1), 249–252.

Tseng, P. (2001). Convergence of a block coordinate descent method for nondifferentiable minimization. *Journal of Optimization Theory and Applications, 109*(3), 475–494.

Tu, Z., & Zhu, S. (2002). Image segmentation by data-driven Markov Chain Monte Carlo. *PAMI, 24*(5), 657–673.

Tugnait, J. K. (1982). Detection and estimation for abruptly changing systems. *Automatica, 18*(5), 607–615.

Turaga, P., Veeraraghavan, A., Srivastava, A., & Chellappa, R. (2011). Statistical computations on special manifolds for image and video-based recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 33*(11), 2273–2286.

Turk, M., & Pentland, A. (1991). Face recognition using eigenfaces. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition* (pp. 586–591).

Udell, M., Horn, C., Zadeh, R., & Boyd, S. (2015). *Generalized low rank models*. Working manuscript.

Ueda, N., Nakan, R., & Ghahramani, Z. (2000). SMEM algorithm for mixture models. *Neural Computation, 12*, 2109–2128.

Vapnik, V. (1995). *The nature of statistical learning theory*. New York: Springer.

Varma, M., & Zisserman, A. (2003). Texture classification: Are filter banks necessary? In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.

Vasilescu, M., & Terzopoulos, D. (2002). Multilinear analysis of image ensembles: Tensorfaces. In *Proceedings of European Conference on Computer Vision* (pp. 447–460).

Vecchio, D. D., & Murray, R. (2004). Observers for a class of hybrid systems on a lattice. In *Proceedings of Hybrid Systems: Computation and Control*. New York: Springer.

Vetterli, M., & Kovacevic, J. (1995). *Wavelets and subband coding*. Upper Saddle River: Prentice-Hall.

Vidal, R. (2004). Identification of PWARX hybrid models with unknown and possibly different orders. In *American Control Conference* (pp. 547–552).

Vidal, R. (2005). Multi-subspace methods for motion segmentation from affine, perspective and central panoramic cameras. In *IEEE Conference on Robotics and Automation* (pp. 1753–1758).

Vidal, R. (2008). Recursive identification of switched ARX systems. *Automatica, 44*(9), 2274–2287.

Vidal, R., Chiuso, A., & Soatto, S. (2002a). Observability and identifiability of jump linear systems. In *IEEE Conference on Decision and Control* (pp. 3614–3619).

Vidal, R., Chiuso, A., Soatto, S., & Sastry, S. (2003a). Observability of linear hybrid systems. In *Hybrid Systems: Computation and Control* (pp. 526–539). New York: Springer.

Vidal, R., & Favaro, P. (2014). Low rank subspace clustering (LRSC). *Pattern Recognition Letters, 43*, 47–61.

Vidal, R., & Hartley, R. (2004). Motion segmentation with missing data by PowerFactorization and Generalized PCA. In *IEEE Conference on Computer Vision and Pattern Recognition* (Vol. II, pp. 310–316).

Vidal, R., & Ma, Y. (2004). A unified algebraic approach to 2-D and 3-D motion segmentation. In *European Conference on Computer Vision* (pp. 1–15).

Vidal, R., Ma, Y., & Piazzi, J. (2004). A new GPCA algorithm for clustering subspaces by fitting, differentiating and dividing polynomials. In *IEEE Conference on Computer Vision and Pattern Recognition* (Vol. I, pp. 510–517).

Vidal, R., Ma, Y., & Sastry, S. (2003b). Generalized Principal Component Analysis (GPCA). In *IEEE Conference on Computer Vision and Pattern Recognition* (Vol. I, pp. 621–628).

Vidal, R., Ma, Y., Soatto, S., & Sastry, S. (2006). Two-view multibody structure from motion. *International Journal of Computer Vision, 68*(1), 7–25.

Vidal, R., & Ravichandran, A. (2005). Optical flow estimation and segmentation of multiple moving dynamic textures. In *IEEE Conference on Computer Vision and Pattern Recognition* (Vol. II, pp. 516–521).

Vidal, R., & Sastry, S. (2003). Optimal segmentation of dynamic scenes from two perspective views. In *IEEE Conference on Computer Vision and Pattern Recognition* (Vol. 2, pp. 281–286).

Vidal, R., Soatto, S., Ma, Y., & Sastry, S. (2002b). Segmentation of dynamic scenes from the multibody fundamental matrix. In *ECCV Workshop on Visual Modeling of Dynamic Scenes*.

Vidal, R., Soatto, S., Ma, Y., & Sastry, S. (2003c). An algebraic geometric approach to the identification of a class of linear hybrid systems. In *IEEE Conference on Decision and Control* (pp. 167–172).

Vidal, R., Tron, R., & Hartley, R. (2008). Multiframe motion segmentation with missing data using PowerFactorization and GPCA. *International Journal of Computer Vision, 79*(1), 85–105.

von Luxburg, U. (2007). A tutorial on spectral clustering. *Statistics and Computing, 17*(4), 395–416.

Wallace, C., & Boulton, D. (1968). An information measure for classification. *The Computer Journal, 11*, 185–194.

Wallace, C., & Dowe, D. (1999). Minimum message length and Kolmogrov complexity. *The Computer Journal, 42*(4), 270–283.

Wallace, G. K. (1991). The JPEG still picture compression standard. *Communications of the ACM. Special issue on digital multimedia systems, 34*(4), 30–44.

Wang, J., Jia, Y., Hua, X., Zhang, C., & Quan, L. (2008a). Normalized tree partitioning for image segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition*.

Wang, J. M., Fleet, D. J., & Hertzmann, A. (2008b). Gaussian process dynamical models for human motion. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 30*(2), 283–298.

Wang, Y.-X., & Xu, H. (2013). Noisy sparse subspace clustering. In *International Conference on Machine learning*.

Ward, J. (1963). Hierarchical grouping to optimize and objective function. *Journal of the American Statistical Association, 58*, 236–244.

Warga, J. (1963). Minimizing certain convex functions. *SIAM Journal on Applied Mathematics, 11*, 588–593.

Wei, S., & Lin, Z. (2010). Analysis and improvement of low rank representation for subspace segmentation. Technical Report MSR-TR-2010-177, Microsoft Research Asia.

Weinberger, K. Q., & Saul, L. (2004). Unsupervised learning of image manifolds by semidefinite programming. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition* (pp. 988–955).

Wiberg, T. (1976). Computation of principal components when data are missing. In *Symposium on Computational Statistics* (pp. 229–326).

Wilks, S. S. (1962). *Mathematical Staistics*. New York: Wiley.

Williams, C. (2002). On a connection between kernel PCA and metric multidimensional scaling. *Machine Learning, 46*, 11–19.

Wolf, L., & Shashua, A. (2001a). Affine 3-D reconstruction from two projective images of independently translating planes. In *Proceedings of IEEE International Conference on Computer Vision* (pp. 238–244).

Wolf, L., & Shashua, A. (2001b). Two-body segmentation from two perspective views. In *IEEE Conference on Computer Vision and Pattern Recognition* (pp. 263–270).

Wolf, L., & Shashua, A. (2003). Learning over sets using kernel principal angles. *Journal of Machine Learning Research, 4*(10), 913–931.

Woolfe, F., & Fitzgibbon, A. (2006). Shift-invariant dynamic texture recognition. In *Proceedings of European Conference on Computer Vision*, pages II: 549–562.

Wright, J., Ganesh, A., Kerui, M., & Ma, Y. (2013). Compressive principal component analysis. *IMA Journal on Information and Inference, 2*(1), 32–68.

Wright, J., Ganesh, A., Rao, S., Peng, Y., & Ma, Y. (2009a). Robust principal component analysis: Exact recovery of corrupted low-rank matrices via convex optimization. In *NIPS*.

Wright, J., Ma, Y., Tao, Y., Lin, Z., & Shum, H.-Y. (2009b). Classification via minimum incremental coding length (MICL). *SIAM Journal on Imahing Sciences, 2*(2), 367–395.

Wu, J. (1983). On the convergence properties of the EM algorithm. *Annals of Statistics, 11*(1), 95–103.

Wu, Y., Zhang, Z., Huang, T., & Lin, J. (2001). Multibody grouping via orthogonal subspace decomposition. In *IEEE Conference on Computer Vision and Pattern Recognition* (Vol. 2, pp. 252–257).

Xiong, F., Camps, O., & Sznaier, M. (2011). Low order dynamics embedding for high dimensional time series. In *IEEE International Conference on Computer Vision* (pp. 2368–2374).

Xiong, F., Camps, O., & Sznaier, M. (2012). Dynamic context for tracking behind occlusions. In *European Conference on Computer Vision. Lecture notes in computer science* (Vol. 7576, pp. 580–593). Berlin/Heidelberg: Springer.

Xu, H., Caramanis, C., & Sanghavi, S. (2010). Robust pca via outlier pursuit. In *Neural Information Processing Systems (NIPS)*.

Yan, J., & Pollefeys, M. (2006). A general framework for motion segmentation: Independent, articulated, rigid, non-rigid, degenerate and non-degenerate. In *European Conference on Computer Vision* (pp. 94–106).

Yang, A., Wright, J., Ma, Y., & Sastry, S. (2008). Unsupervised segmentation of natural images via lossy data compression. *Computer Vision and Image Understanding, 110*(2), 212–225.

Yang, A. Y., Rao, S. R., & Ma, Y. (2006). Robust statistical estimation and segmentation of multiple subspaces. In *CVPR workshop on 25 years of RANSAC*.

Yang, J., Wright, J., Huang, T., & Ma, Y. (2010). Image super-resolution via sparse representation. *IEEE Transactions on Image Processing, 19*(11), 2861–2873.

Yang, M. H., Ahuja, N., & Kriegman, D. (2000). Face detection using mixtures of linear subspaces. In *IEEE International Conference on Automatic Face and Gesture Recognition*.

Yu, G., Sapiro, G., & Mallat, S. (2010). Image modeling and enhancement via structured sparse model selection. In *International Conference on Image Processing*.

Yu, G., Sapiro, G., & Mallat, S. (2012). Solving inverse problems with piecewise linear estimators: From gaussian mixture models to structured sparsity. *IEEE Transactions on Image Processing, 21*(5), 2481–2499.

Yu, S. (2005). Segmentation induced by scale invariance. In *IEEE Conference on Computer Vision and Pattern Recognition*.

Yuan, L., Wen, F., Liu, C., & Shum, H. (2004). Synthesizing dynamic texture with closed-loop linear dynamic system. In *European Conference on Computer Vision* (pp. 603–616).

Yuan, X., & Yang, J. (2009). Sparse and low-rank matrix decomposition via alternating direction methods. *Preprint*.

Zadeh, N. (1970). A note on the cyclic coordinate ascent method. *Management Science, 16*, 642–644.

Zelnik-Manor, L., & Irani, M. (2003). Degeneracies, dependencies and their implications in multi-body and multi-sequence factorization. In *IEEE Conference on Computer Vision and Pattern Recognition* (Vol. 2, pp. 287–293).

Zhang, K., Zhang, L., & Yang, M. (2014). Fast compressive tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 36*(10).

Zhang, T., Szlam, A., & Lerman, G. (2009). Median *k*-flats for hybrid linear modeling with many outliers. In *Workshop on Subspace Methods*.

Zhang, T., Szlam, A., Wang, Y., & Lerman, G. (2010). Randomized hybrid linear modeling via local best-fit flats. In *IEEE Conference on Computer Vision and Pattern Recognition* (pp. 1927–1934).

Zhang, Z., & Zha, H. (2005). Principal manifolds and nonlinear dimensionality reduction via tangent space alignment. *SIAM Journal on Scientific Computing, 26*(1), 313–338.

Zhou, F., la Torre, F. D., & Hodgins, J. K. (2008). Aligned cluster analysis for temporal segmentation of human motion. In *International Conference on Automatic Face and Gesture Recognition*.

Zhou, M., Wang, C., Chen, M., Paisley, J., Dunson, D., & Carin, L. (2010a). Nonparametric bayesian matrix completion. In *Sensor Array and Multichannel Signal Processing Workshop*.

Zhou, Z., Wright, J., Li, X., Candès, E., & Ma, Y. (2010b). Stable principal component pursuit. In *International Symposium on Information Theory*.

Zhu, Q., Song, G., & Shi, J. (2007). Untangling cycles for contour grouping. In *Proceedings of the International Conference on Computer Vision (ICCV)*.

Index

R. Vidal et al., *Generalized Principal Component Analysis*, Interdisciplinary Applied Mathematics 40, DOI 10.1007/978-0-387-87811-9