

Probabilistic Analysis

■ 4.1 Introduction.

THE recent years have seen a great number of exciting developments in the field of nonlinear diffusion filtering of images. As summarized in Chapter 2 and Section 3.6, many theories have been proposed that result in edge-preserving scale spaces possessing various interesting properties. One striking feature unifying many of these frameworks—including the one introduced in the previous chapter—is that they are deterministic. Usually, one starts with a set of “common-sense” principles which an image smoothing operation should satisfy. Examples of these are the axioms in [1] and the observation in [49] that, in order to achieve edge preservation, very little smoothing should be done at points with high gradient. From these principles, a nonlinear scale space is derived, and then it is analyzed—again, deterministically. Note, however, that since the objective of these techniques is usually restoration or segmentation of images in the presence of noise, a natural question to ask would be:

Do nonlinear diffusion techniques solve standard estimation or detection problems? (*)

An affirmative answer would help us understand which technique is suited best for a particular application, and aid in designing new algorithms. It would also put the tools of the classical detection and estimation theory at our disposal for the analysis of these techniques, making it easier to tackle an even more crucial question:

Given a probabilistic noise model, can one characterize the performance of the nonlinear diffusion techniques? (**)

Attempts to address these issues in the literature have remained scarce—most likely, because the complex nature of the nonlinear partial differential equations (PDEs) considered and of the images of interest make this analysis prohibitively complicated. Most notable exceptions are [63, 75] which establish qualitative relations between the Perona-Malik equation [49] and gradient descent procedures for estimating random fields modeled by Gibbs distributions. Bayesian ideas are combined in [76] with snakes and region growing for image segmentation. In [5], concepts from robust statistics are used to

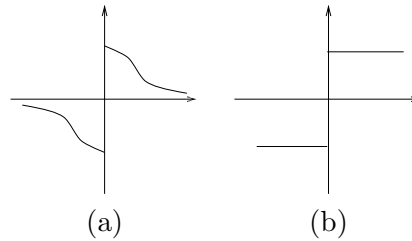


Figure 4.1. Functions F from the right-hand side of the SIDE: (a) generic form; (b) the signum function.

modify the Perona-Malik equation. In [38], a connection between random walks and diffusions is used to obtain a new evolution equation.

The goal of this chapter is to move forward the discussion of questions (*) and (**). We consider a very simple nonlinear diffusion (a variant of those introduced in the previous chapter) which provides a multiscale sequence of segmentations of its initial condition. In Sections 4.3 and 4.4, we apply our algorithm to 1-D signals, and describe edge detection problems which are solved optimally by this diffusion. These are binary classification problems: each sample has to be classified as coming from one of two classes, subject to the constraint on the number of “edges”—i.e., changes between the two classes. One of these problems turns out to be the minimization of a special case of the 1-D Mumford-Shah functional [44]. We describe an efficient implementation of the 1-D diffusion, requiring $O(N \log N)$ computations in the worst case, where N is the size of the input signal. In Section 4.5, we point out that the same 1-D problem can also be solved via dynamic programming and via linear programming, but that our method has certain advantages over both. To analyze the performance (Section 4.6), we simplify even further, by considering signals with only one change in mean. Our performance measure is the accuracy in locating the change. More precisely, the probability of events of the form “the detected change location is more than p samples away from the actual one” is analyzed. We show that the asymptotic probabilities of these events can be obtained directly from the classical change detection paper by Hinkley [28]. We also derive non-asymptotic lower bounds on these probabilities. The robustness of the algorithm—which is experimentally confirmed both in this chapter and in Chapter 3—is analyzed theoretically by showing the optimality with respect to a certain H_∞ -like criterion. In Section 4.7, we treat segmentation of 2-D images.

■ 4.2 Background and Notation.

In this chapter, we consider a special case of SIDEs (3.11) in 1-D, which results if one drops the “monotonically decreasing” requirement on F , and takes $F(v) = \text{sgn}(v)$ instead (see Figure 4.1, (b)). Specifically, we are interested in the evolution of the

following equation:

$$\begin{aligned} \dot{u}_1 &= \frac{\text{sgn}(u_2 - u_1)}{m_1}, \quad \dot{u}_N = \frac{\text{sgn}(u_{N-1} - u_N)}{m_N}, \\ \dot{u}_n &= \frac{1}{m_n}(\text{sgn}(u_{n+1} - u_n) - \text{sgn}(u_n - u_{n-1})), \\ &\text{for } n = 2, \dots, N-1, \end{aligned} \quad (4.1)$$

with the initial condition

$$\mathbf{u}(0) = \mathbf{u}^0, \quad (4.2)$$

where, as we explain in Section 4.4, \mathbf{u}^0 is either the signal to be processed or a sequence of logarithms of pointwise likelihood ratios. As in the previous chapter, N stands for the number of samples in the signals under consideration. Boldface letters denote these signals, whose entries are always denoted by the same letter with subscripts 1 through N : $\mathbf{u} = (u_1, \dots, u_N)^T$. Initially the finest segmentation is assumed: each pixel is a separate region, i.e. $m_n = 1$, for $n = 1, \dots, N$. As soon as u_i becomes equal to u_{i+1} , these values stay equal forever, and their equations are replaced with

$$\dot{u}_i = \dot{u}_{i+1} = \frac{(\text{sgn}(u_{i+2} - u_{i+1}) - \text{sgn}(u_i - u_{i-1}))}{m_i + m_{i+1}}, \quad (4.3)$$

which is Equation (3.10). The rest of the present chapter deals with the particular SIDE (4.1,4.2,4.3).

We apply the SIDE (4.1,4.2) to binary classification problems. Given an observation \mathbf{y} , the goal is to label each sample as coming from one of two classes, i.e. to produce a binary signal \mathbf{h} whose entries are zeros and ones. We call any such binary signal \mathbf{h} an *hypothesis*.

Definition 4.1. We denote the set of all N -dimensional binary signals by $\{0, 1\}^N$. A member \mathbf{h} of this set is called an **hypothesis** with the following interpretation. Specifically, if $h_i \neq h_{i+1}$, we say that an **edge** (or, equivalently, a **change**) is hypothesized at the location i , and we call $\text{sgn}(h_{i+1} - h_i)$ **the sign of the edge**. We say that an edge is directed **upward** (**downward**) if $h_{i+1} > h_i$ ($h_{i+1} < h_i$). ■

Definition 4.2. A **statistic** is a function $\phi : \mathbb{R}^N \times \{0, 1\}^N \rightarrow \mathbb{R}$. The **optimal hypothesis** $\mathbf{h}^*(\mathbf{u})$ for a signal $\mathbf{u} \in \mathbb{R}^N$ with respect to ϕ is

$$\mathbf{h}^*(\mathbf{u}) \stackrel{\text{def}}{=} \arg \max_{\mathbf{h} \in \{0, 1\}^N} \phi(\mathbf{u}, \mathbf{h}).$$

The best hypothesis among those whose number of edges does not exceed some constant ν is

$$\mathbf{h}_{\leq \nu}^*(\mathbf{u}) \stackrel{\text{def}}{=} \arg \max_{\mathbf{h} \in \{0, 1\}^N, \mathbf{h} \text{ has } \nu \text{ or fewer edges}} \phi(\mathbf{u}, \mathbf{h}). \quad \blacksquare$$

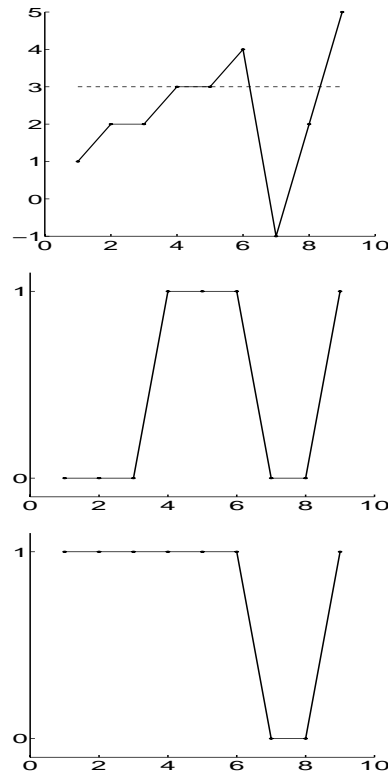


Figure 4.2. Illustrations of Definitions 4.1 and 4.3: a sequence with three α -crossings, where $\alpha=3$ (top); the hypothesis generated by the three α -crossings (middle); the hypothesis generated by the two rightmost α -crossings (bottom).

Note that an hypothesis is uniquely defined by the set of its edges and the sign of one of the edges. Therefore, binary classification problems can also be viewed as edge detection problems. For the problems considered in this chapter, the optimal edge locations will typically be level crossings of some signal.

Definition 4.3. A signal \mathbf{u} is said to have an α -crossing at the location i if $(u_i - \alpha)(u_j - \alpha) < 0$, where $j = \min\{n: n > i, u_n \neq \alpha\}$. (In other words, u_j is the first sample to the right of i which is not equal to α .) We call $\text{sgn}(\alpha - u_i)$ the **sign of the α -crossing**, and say that the α -crossing is directed **upward (downward)** if $u_i < \alpha$ ($u_i > \alpha$). We define the hypothesis **generated** by a set of α -crossings $\{g_1, \dots, g_\nu\}$ of \mathbf{u} as the hypothesis whose edges are at g_1, \dots, g_ν and for which the sign of each edge is the same as the sign of the corresponding α -crossing. ■

To illustrate Definitions 4.1, 4.2, and 4.3, let us consider an example.

Example 4.1. Illustration of the definitions of edges, α -crossings, and statistics.

Suppose

$$\mathbf{u} = (1, 2, 2, 3, 3, 4, -1, 2, 5)^T$$

(see top of Figure 4.2), and $\alpha = 3$. Then \mathbf{u} has three α -crossings, at locations $g_1 = 3$, $g_2 = 6$, and $g_3 = 8$. The second one is directed downward, and the other two are directed upward. The hypothesis \mathbf{h}_1 generated by these three α -crossings must therefore have upward edges at 3 and 8 and a downward edge at 6:

$$\mathbf{h}_1 = (0, 0, 0, 1, 1, 1, 0, 0, 1)^T,$$

as depicted in the middle plot of Figure 4.2. The hypothesis \mathbf{h}_2 generated by the α -crossings g_2 and g_3 will only have a downward edge at 6 and an upward edge at 8:

$$\mathbf{h}_2 = (1, 1, 1, 1, 1, 1, 0, 0, 1)^T,$$

as shown in the last plot of Figure 4.2. If we define a statistic ϕ by

$$\begin{aligned} \phi(\mathbf{u}, \mathbf{h}) &= \mathbf{h}^T(\mathbf{u} - \mathbf{a}), \\ \text{where } \mathbf{a} &= (3, \dots, 3)^T \in \mathbb{R}^9, \end{aligned}$$

then we have:

$$\begin{aligned} \phi(\mathbf{u}, \mathbf{h}_1) &= 3, \\ \phi(\mathbf{u}, \mathbf{h}_2) &= -1. \quad \blacksquare \end{aligned}$$

■ 4.3 SIDE as an Optimizer of a Statistic.

The usefulness of the SIDE (4.1,4.2) in solving edge detection problems comes from its ability to maximize certain statistics. One of the properties of such a statistic $\phi(\mathbf{u}, \mathbf{h})$ is that the optimal hypothesis $\mathbf{h}^*(\mathbf{u})$ is generated by the set of all α -crossings of \mathbf{u} , for some number α . It is shown below in Proposition 4.1 that every edge of $\mathbf{h}_{\leq \nu}^*(\mathbf{u})$ also occurs at an α -crossing of \mathbf{u} . Furthermore, Proposition 4.2 describes how to find these edges using the SIDE (4.1): it says that the α -crossings of the solution $\mathbf{u}(t)$ to the SIDE (4.1) then generate the constrained optimal hypothesis $\mathbf{h}_{\leq \nu}^*(\mathbf{u}^0)$, where \mathbf{u}^0 is the initial data, and ν is the number of α -crossings of $\mathbf{u}(t)$. We note that when we talk about α -crossings of \mathbf{u} in Propositions 4.1 and 4.2, we will allow α to be a function of $\frac{1}{N} \sum_{i=1}^N u_i$. This is necessary to cover some important examples considered later in Section 4.4. On the other hand, $\frac{1}{N} \sum_{i=1}^N u_i(t)$ is constant throughout the evolution of the SIDE, as easily verified by summing up the equations (4.1). Thus, $\alpha(\frac{1}{N} \sum_{i=1}^N u_i)$ will also stay constant during the evolution of the SIDE. We therefore will simply refer to α , dropping its argument to avoid notational clutter. It will, however, be our implicit assumption throughout the remainder of this chapter that whenever we mention α -crossings of a signal \mathbf{u} , α is allowed to be a function of $\frac{1}{N} \sum_{i=1}^N u_i$.

Proposition 4.1. *Let $\alpha : \mathbb{R} \rightarrow \mathbb{R}$ be a scalar function, and let $\mathbf{a} : \mathbb{R} \rightarrow \mathbb{R}^N$ be the vector function whose every entry is α : $\mathbf{a}(x) = (\alpha(x), \dots, \alpha(x))^T$ for any $x \in \mathbb{R}$. Define the following statistic:*

$$\phi(\mathbf{u}, \mathbf{h}) = \mathbf{h}^T \left(\mathbf{u} - \mathbf{a} \left(\frac{1}{N} \sum_{i=1}^N u_i \right) \right). \quad (4.4)$$

Then every edge of $\mathbf{h}_{\leq \nu}^*(\mathbf{u}^0)$ occurs at an α -crossing of \mathbf{u}^0 , for any $\mathbf{u}^0 \in \mathbb{R}^N$.

Proof is in Appendix B. ■

Proposition 4.2. *Suppose that ϕ is the statistic described in Proposition 4.1. Fix the initial condition \mathbf{u}^0 of the SIDE (4.1), and let $\mathbf{u}(t)$ be the corresponding solution. Then $\alpha(\frac{1}{N} \sum_{i=1}^N u_i(t))$ is constant during the evolution of the SIDE, as verified by summing up the equations (4.1). Let $\nu_\alpha(t)$ be the number of α -crossings of $\mathbf{u}(t)$. Then, for any time instant $t_f > 0$,*

$$\mathbf{h}_{\leq \nu_\alpha(t_f)}^*(\mathbf{u}^0) = \mathbf{h}^*(\mathbf{u}(t_f)). \quad \blacksquare$$

The proof is in Appendix B. We note that Proposition 1 of [51] is a different formulation of the same result: in [51], we explicitly listed the properties of ϕ which are used in the proof. The equivalence of the two formulations is also shown in Appendix B.

This proposition says that, if the SIDE is evolved until $\nu_\alpha(t)$ α -crossings remain, then these α -crossings are the optimal edges, where “optimality” means maximizing the statistic $\phi(\mathbf{u}^0, \mathbf{h})$ subject to the constraint that the hypothesis have $\nu_\alpha(t)$ or fewer edges. It is verified in the next subsection that $\nu_\alpha(t)$ is a non-increasing function of time, with $\nu_\alpha(\infty) = 0$. Unfortunately, $\nu_\alpha(t)$ is not guaranteed to assume every integer value between $\nu_\alpha(0)$ and 0: during the evolution of the SIDE, α -crossings can disappear in pairs. We will show in the next subsection, however, that no more than two α -crossings can disappear at the same time. We will also show that, even if for some integer $\nu < \nu_\alpha(0)$ there is no t such that $\nu_\alpha(t) = \nu$ (i.e. $\nu_\alpha(t)$ goes from $\nu + 1$ directly to $\nu - 1$), we can still easily find $\mathbf{h}_{\leq \nu}^*(\mathbf{u}^0)$ using the set of α -crossings of the solution $\mathbf{u}(t)$ to the SIDE at the time t when $\nu_\alpha(t) = \nu + 1$. If the desired number of edges is greater than or equal to the initial number of α -crossings, $\nu \geq \nu_\alpha(0)$, then, from the definitions of $\mathbf{h}_{\leq \nu}^*(\mathbf{u}^0)$ and $\mathbf{h}^*(\mathbf{u}^0)$, we immediately have:

Proposition 4.3. *Suppose that ϕ is the statistic described in Proposition 4.1. If $\nu \geq \nu_\alpha(0)$, then*

$$\mathbf{h}_{\leq \nu}^*(\mathbf{u}^0) = \mathbf{h}^*(\mathbf{u}^0). \quad \blacksquare$$

In the remainder of the chapter, we assume that $\nu < \nu_\alpha(0)$.

In Section 4.4, we will give examples of detection problems whose solution is equivalent to maximizing the statistic ϕ . We will therefore be able to utilize the SIDE for optimally solving these problems. Before we do that, however, we describe how to efficiently implement the SIDE.

■ **4.3.1 Implementation of the SIDE Via a Region Merging Algorithm.**

We now show how to solve efficiently the optimization problem treated in Proposition 4.2 of the previous subsection, using the SIDE (4.1).

Problem 4.1. *Define a statistic*

$$\phi(\mathbf{u}, \mathbf{h}) = \mathbf{h}^T \left(\mathbf{u} - \mathbf{a} \left(\frac{1}{N} \sum_{i=1}^N u_i \right) \right),$$

where $\mathbf{a}(x) = (\alpha(x), \dots, \alpha(x))^T \in \mathbb{R}^N$, and α is a real-valued function of a real argument. Given an integer ν and a signal \mathbf{u}^0 , we are interested in finding the best hypothesis $\mathbf{h}_{\leq \nu}^*(\mathbf{u}^0)$ among all the hypotheses with ν or fewer edges, where “the best” means the one maximizing $\phi(\mathbf{u}^0, \cdot)$. ■

Proposition 4.2 relates $\mathbf{h}_{\leq \nu}^*(\mathbf{u}^0)$ to the solution $\mathbf{u}(t)$ of the SIDE whose initial data is \mathbf{u}^0 . Namely, it says that if ν is the number of α -crossings of $\mathbf{u}(t)$ ¹, then these α -crossings generate the hypothesis $\mathbf{h}_{\leq \nu}^*(\mathbf{u}^0)$. It is, however, not guaranteed that for every integer ν there is a time instant t when the solution $\mathbf{u}(t)$ has exactly ν α -crossings. Therefore, in order to compute the solution to Problem 4.1, we need to deal with two issues:

- (A) How to calculate the locations of the α -crossings of $\mathbf{u}(t)$?
- (B) How to compute $\mathbf{h}_{\leq \nu}^*(\mathbf{u}^0)$ for every integer ν ?

We first consider issue A. In order to find the α -crossings of the solution to the SIDE, one can certainly use a finite difference scheme to numerically integrate the equation. There is, however, a much faster way, which exploits the special structure of the equation. It turns out that, during the evolution of the SIDE, α -crossings cannot be created or shifted: they can only be erased. We therefore only need to compute the order in which they disappear. We now make these statements precise.

Lemma 4.1. *Suppose that at time t_0 , the solution $\mathbf{u}(t_0)$ to the SIDE has no α -crossing at the location i . Then $\mathbf{u}(t)$ has no α -crossing at i , either, for all $t \geq t_0$.*

Example 4.2 (Example 4.1, continued). *Illustration of Lemma 4.1.*

We illustrate this Lemma by evolving the SIDE (4.1) for the initial condition

$$\mathbf{u}^0 = (1, 2, 2, 3, 3, 4, -1, 2, 5)^T$$

(top of Figure 4.2) and $\alpha = 3$. The values of the solution for several time instants are recorded in Table 1.

¹Just as in the previous subsection, we abuse notation by dropping the argument $\frac{1}{N} \sum_{i=1}^N u_i(t)$ of α .

t	$\mathbf{u}^T(t)$
$t = 0$	$(1, 2, 2, 3, 3, 4, -1, 2, 5)$
$t = \frac{1}{2}$	$(1\frac{1}{2}, 2, 2, 3, 3, 3, 0, 2, 4\frac{1}{2})$
$t = 1$	$(2, 2, 2, 2\frac{2}{3}, 2\frac{2}{3}, 2\frac{2}{3}, 1, 2, 4)$
$t = 1\frac{1}{2}$	$(2\frac{1}{6}, 2\frac{1}{6}, 2\frac{1}{6}, 2\frac{1}{3}, 2\frac{1}{3}, 2\frac{1}{3}, 2, 2, 3\frac{1}{2})$
$t = 1\frac{2}{3}$	$(2\frac{2}{9}, 2\frac{2}{9}, 2\frac{2}{9}, 2\frac{2}{9}, 2\frac{2}{9}, 2\frac{2}{9}, 2\frac{1}{6}, 2\frac{1}{6}, 3\frac{2}{3})$
$t = 2$	$(2\frac{1}{4}, 2\frac{1}{4}, 2\frac{1}{4}, 2\frac{1}{4}, 2\frac{1}{4}, 2\frac{1}{4}, 2\frac{1}{4}, 2\frac{1}{4}, 3\frac{1}{3})$
$t = 2\frac{2}{3}$	$(2\frac{1}{3}, 2\frac{1}{3}, 2\frac{1}{3}, 2\frac{1}{3}, 2\frac{1}{3}, 2\frac{1}{3}, 2\frac{1}{3}, 2\frac{1}{3}, 2\frac{2}{3})$

At the beginning, $\mathbf{u}(t)$ has three α -crossings, at locations 3, 6, and 8. At time $t = \frac{1}{2}$, the first two α -crossings disappear. Between $t = 2$ and $t = 2\frac{2}{3}$, the α -crossing located at 8 also disappears. ■

In order to prove Lemma 4.1, we need to re-define a *region* (i, j) of a signal \mathbf{u} as the set of all samples $\{u_i, \dots, u_j\}$ either between two consecutive α -crossings $i - 1$ and j of \mathbf{u} , or to the left of the leftmost α -crossing j (in which case $i = 1$), or to the right of the rightmost α -crossing $i - 1$ (in which case $j = N$)². For example, referring to Table 1, $\mathbf{u}(0)$ has four regions: $(1, 3)$, $(4, 6)$, $(7, 8)$, and $(9, 9)$. Similarly, $\mathbf{u}(2)$ has two regions: $(1, 8)$ and $(9, 9)$. Note that a sample $u_k(t)$ belonging to a region (i, j) can only cross the level α if all other samples in its region do so at the same time:

Lemma 4.2. *Let (i, j) be a region of $\mathbf{u}(t_0)$, where $t_0 \geq 0$, and let the intensity values inside this region be above α : $u_i(t_0) > \alpha, \dots, u_j(t_0) > \alpha$. Let t_1 be the first time instant after t_0 at which one of the values inside the region, say $u_k(t_1)$, becomes equal to α :*

$$t_1 = \inf\{t > t_0 : \exists k, i \leq k \leq j, u_k(t) = \alpha\}.$$

Then

$$u_i(t_1) = u_{i+1}(t_1) = \dots u_j(t_1).$$

Proof. Notice that, according to Equations (4.1,4.3), $u_k(t)$ can be decreasing only if it is a local maximum (i.e., if $u_k(t) \geq u_{k\pm 1}(t)$). Thus, at time t_1 , we have that

- $u_k(t_1) = \alpha$ is a local maximum;
- all other values inside the region (i, j) are $\geq \alpha$.

Consequently, it must be that the values at all the samples inside the region (i, j) are equal to α . ■

Proof of Lemma 4.1. A proof similar to the one above applies to the variant of Lemma 4.2 when $u_i(t_0), \dots, u_j(t_0)$ are less than α . Thus, the value $u_k(t)$ at any location k can

²Note that this definition is somewhat different from the one in Chapter 3, where all pixels in a region had the same value.

cross the level α only when its whole region does so. This means that the evolution of the SIDE cannot create or shift α -crossings; it can only remove them. ■

We now show how to calculate the order in which the regions disappear. It turns out that this ordering depends on how the removal of a region influences the statistic $\phi(\mathbf{u}(t), \mathbf{h})$. Define the *energy* E_{ij} of the region (i, j) by

$$\begin{aligned} E_{ij}(t) &= \frac{1}{\rho_{ij}} \left| \sum_{n=i}^j (u_n(t) - \alpha) \right|, \\ \rho_{ij} &= \begin{cases} 1 & \text{if } i = 1 \text{ or } j = N \\ 2 & \text{otherwise.} \end{cases} \end{aligned} \quad (4.5)$$

Note that the energy measures the contribution of the region (i, j) to $\phi(\mathbf{u}(t), \mathbf{h})$. Summing up the equations (4.1) from $n = i$ to $n = j$, we see that, for every region (i, j) , $\dot{E}_{ij}(t) = -1$. A region (i, j) is erased when the values of all its samples reach α —i.e., when the energy $E_{ij}(t)$ becomes equal zero. Since all the energies are diminished with the same speed, it follows that the first region to disappear will be the one for which $E_{ij}(0)$ is the smallest. Applying this reasoning recursively, we then remove the region with the next smallest energy, etc., obtaining the following algorithm to compute the α -crossings of $\mathbf{u}(t)$.

1. **Initialize.** Let A be the set of all α -crossings of \mathbf{u}^0 , ordered from left to right, and let $\bar{\nu} = \nu_\alpha(0)$ be the total number of α -crossings.
2. **Compute the energies.** Denote the elements of the set A by $g_1, \dots, g_{\bar{\nu}}$, and form $\bar{\nu} + 1$ regions: $(1, g_1), (g_1 + 1, g_2), \dots, (g_{\bar{\nu}} + 1, N)$. For each region (i, j) , compute its energy E_{ij} , as defined by (4.5).
3. **Remove the region with minimal energy.** Let (i_m, j_m) be the region for which E_{ij} is the smallest (if there are several regions with the smallest energy, choose any one). Merge the region (i_m, j_m) with its neighbors by re-defining A and $\bar{\nu}$ via

$$A \leftarrow A \setminus \{i_m, j_m\}, \quad \bar{\nu} \leftarrow \text{the size of the new } A.$$

4. **Iterate.** If $\bar{\nu}$ is greater than the desired number of edges, go to step 2. ■

To illustrate the algorithm, we apply it to the signal of Example 4.1 (top of Figure 4.2).

Example 4.3 (Examples 4.1 and 4.2, continued). *Illustration of the region merging algorithm.*

We again consider

$$\mathbf{u}^0 = (1, 2, 2, 3, 3, 4, -1, 2, 5)^T.$$

Iteration 1. There are three α -crossings, and four regions: $(1, 3)$, $(4, 6)$, $(7, 8)$, and $(9, 9)$, with the energies $E_{13} = 4$, $E_{46} = 0.5$, $E_{78} = 2.5$, and $E_{99} = 2$, respectively. The

region (4,6) has the smallest energy, and therefore it is removed first, by merging it with its two neighbors to form the new region (1,8).

Iteration 2. There are now two regions, (1,8) and (9,9), with the energies $E_{1,8} = 8$ and $E_{9,9} = 2$, respectively. They are merged, to form one region (1,9). Note that the order in which the regions disappear is in agreement with Table 1. ■

We now show that the algorithm is fast. The initialization steps 1 and 2 take $O(N)$ time. Step 3 merges either two regions (if $i_m = 1$ or $j_m = N$) or three regions (otherwise). The energy of the new region is essentially the sum of the energies of its constituent regions, and therefore the recomputation of energies after a merging takes $O(1)$ time. If a binary heap [11] is used to store the energies, the size of the heap at every iteration of the algorithm will be equal to the current number of distinct regions, which is $\bar{\nu} + 1$, where $\bar{\nu}$ is the current number of α -crossings. Therefore, finding the minimal element of the heap (step 3) at each iteration will take $O(\log \bar{\nu} + 1)$ time, which means that the algorithm will run in $O(\sum_{\bar{\nu}=\nu+1}^{\nu_\alpha(0)} \log \bar{\nu} + N)$ time. The worst case is when $\nu = 1$ and $\nu_\alpha(0) = N - 1$. Then the computational complexity is $O(N \log N)$. However, if the desired number of edges ν is comparable with the initial number $\nu_\alpha(0)$ (which can happen in low-noise scenarios), the complexity is $O(N)$.

We still have to address Question (B) which we posed at the beginning of this subsection, namely, how to find $\mathbf{h}_{\leq \nu}^*(\mathbf{u}^0)$ for every integer ν . If there is a time instant t at which $\mathbf{u}(t)$ has exactly ν α -crossings, then, according to Proposition 4.2, these α -crossings generate the hypothesis $\mathbf{h}_{\leq \nu}^*(\mathbf{u}^0)$, which means that Problem 4.1 is solved. The scenario which we need to consider now is when there is no such time t at which $\mathbf{u}(t)$ has exactly ν α -crossings. As we showed above, computing the locations of α -crossings of $\mathbf{u}(t)$ involves removing regions one at a time (Step 3 of the algorithm). Thus, at most two α -crossings can disappear at the same time. Therefore, if $\mathbf{u}(t)$ never has ν α -crossings, it must go from $\nu + 1$ α -crossings directly to $\nu - 1$. It turns out that, in this case, one can still compute $\mathbf{h}_{\leq \nu}^*(\mathbf{u}^0)$ by running the algorithm above until $\nu - 1$ α -crossings remain, and then doing post-processing whose computational complexity is $O(N)$. Specifically, the following proposition holds.

Proposition 4.4. *Suppose that there is a time instant t during the evolution of the SIDE such that $\mathbf{u}(t^-)$ has $\nu + 1$ α -crossings, at locations $g_1, \dots, g_{\nu+1}$. The hypothesis generated by these α -crossings is $\mathbf{h}_{\leq \nu+1}^*(\mathbf{u}^0)$. Suppose further that the region $(g_k + 1, g_{k+1})$ disappears at time t , so that $\mathbf{u}(t^+)$ has $\nu - 1$ α -crossings, at locations $g_1, \dots, g_{k-1}, g_{k+2}, \dots, g_{\nu+1}$. The hypothesis generated by these α -crossings is $\mathbf{h}_{\leq \nu-1}^*(\mathbf{u}^0)$. Then one of the following four possibilities must happen.*

- (i) $\mathbf{h}_{\leq \nu}^*(\mathbf{u}^0) = \mathbf{h}_{\leq \nu-1}^*(\mathbf{u}^0)$.
- (ii) $\mathbf{h}_{\leq \nu}^*(\mathbf{u}^0)$ has edges at the locations $g_2, \dots, g_{\nu+1}$.
- (iii) $\mathbf{h}_{\leq \nu}^*(\mathbf{u}^0)$ has edges at the locations g_1, \dots, g_ν .
- (iv) $\mathbf{h}_{\leq \nu}^*(\mathbf{u}^0)$ has edges at the locations $g_1, \dots, g_{k-1}, g_{k+2}, \dots, g_{\nu+1}$, as well as one edge at the location which is an element of the set $\{1, 2, \dots, g_1 - 1, g_{\nu+1} + 1, \dots, N - 1\}$.

Thus, finding $\mathbf{h}_{\leq \nu}^*(\mathbf{u}^0)$ is achieved by running the SIDE and doing post-processing of complexity $O(N)$. ■

Proposition 4.4 is the recipe for obtaining the optimal hypothesis $\mathbf{h}_{\leq \nu}^*(\mathbf{u}^0)$ from the $\nu + 1$ α -crossings of $\mathbf{u}(t^-)$. It says that either ν or $\nu - 1$ of these α -crossings coincide with the edges of $\mathbf{h}_{\leq \nu}^*(\mathbf{u}^0)$. Cases (ii) and (iii) describe the only two subsets consisting of ν α -crossings which can generate $\mathbf{h}_{\leq \nu}^*(\mathbf{u}^0)$. If only $\nu - 1$ α -crossings of $\mathbf{u}(t^-)$ coincide with the edges of $\mathbf{h}_{\leq \nu}^*(\mathbf{u}^0)$, then either there are no other edges (Case (i)), or the remaining edge is easily found in linear time (Case (iv)).

We note that a slight correction to what was reported in [51] is in order: although the statement that the complexity of the post-processing is $O(N)$ is correct, the specific post-processing procedure given there is somewhat different from the one outlined in Proposition 4.4 above, and therefore, it may be *incorrect* for some data sequences.

As one can infer from the statement of this proposition, its proof is rather technical and amounts to analyzing various scenarios of the disappearance of the α -crossings. This proposition is a direct corollary of Proposition 4.1 and the following lemma.

Lemma 4.3. *As in Proposition 4.4, let t be the time instant when the solution of the SIDE goes from $\nu + 1$ α -crossings to $\nu - 1$. Let $\mathbf{h} = \mathbf{h}_{\leq \nu}^*(\mathbf{u}^0)$, and suppose that it is generated by ν α -crossings of \mathbf{u}^0 : g_1, \dots, g_ν . (Note that this notation is different from the notation of Proposition 4.4.) Then at least $\nu - 1$ elements of the set $\{g_1, \dots, g_\nu\}$ are also α -crossings of $\mathbf{u}(t^-)$, with possible exception of either g_1 or g_ν . Furthermore, if exactly $\nu - 1$ elements of the set $\{g_1, \dots, g_\nu\}$ are α -crossings of $\mathbf{u}(t^-)$, they are also α -crossings of $\mathbf{u}(t^+)$,*

Proof is in Appendix B. ■

■ 4.4 Detection Problems Optimally Solved by the SIDE.

We now give examples of detection problems whose solution is equivalent to maximizing the statistic ϕ of Proposition 4.2. These problems can therefore be optimally solved by the SIDE.

■ 4.4.1 Two Distributions with Known Parameters.

Let \mathbf{y} be an observation of a sequence of N independent random variables. Suppose that each random variable has probability density function (pdf) either $f(y, \theta_0)$ or $f(y, \theta_1)$, where θ_0 and θ_1 are known. It is also known that the number of changes between the two pdf's does not exceed ν ; however, it is not known where these changes occur.

To obtain the maximum likelihood hypothesis [66], one has to maximize the log likelihood function

$$\sum_{i:h_i=1} \log f(y_i, \theta_1) + \sum_{i:h_i=0} \log f(y_i, \theta_0),$$

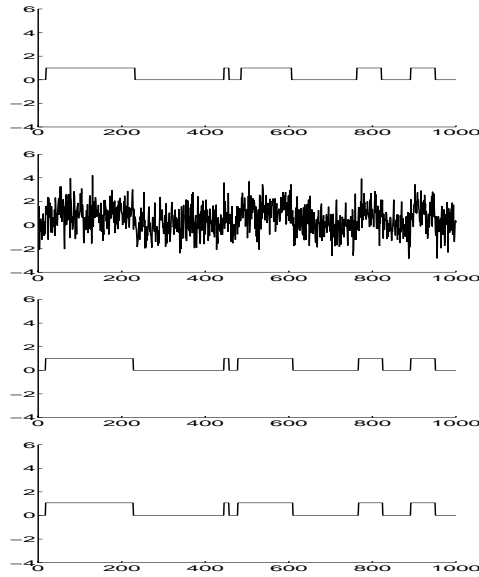


Figure 4.3. Edge detection for a binary signal in Gaussian noise.

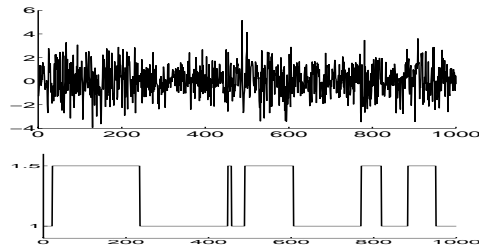


Figure 4.4. Detection of changes in variance of Gaussian noise.

where the hypothesis \mathbf{h} is such that the sample y_i is hypothesized to be from the pdf $f(y, \theta_{h_i})$. Note that by defining a signal consisting of pointwise log-likelihoods,

$$u_i^0 = \log f(y_i, \theta_1) - \log f(y_i, \theta_0), \quad (4.6)$$

we see that the log-likelihood is equal to

$$\mathbf{h}^T \mathbf{u}^0 + \sum_{i=1}^N \log f(y_i, \theta_0).$$

The second term is independent of \mathbf{h} , and therefore maximizing this function is equivalent to maximizing

$$\phi(\mathbf{u}^0, \mathbf{h}) \stackrel{\text{def}}{=} \mathbf{h}^T \mathbf{u}^0, \quad (4.7)$$

which is the statistic of Proposition 4.2 with $\alpha = 0$. Thus, the SIDE can be employed for finding the maximum likelihood hypothesis $\mathbf{h}_{\leq \nu}^*(\mathbf{u}^0)$, where \mathbf{u}^0 is related to the observation \mathbf{y} through (4.6).

Example 4.4. *Changes in mean in a Gaussian random vector.*

In this example, $f(y, \theta_j)$ is the Gaussian density with mean θ_j and variance 1. We took $\theta_0 = 0$ and $\theta_1 = 1$. We assumed that the right number of edges, 10, is known, and so the stopping rule for the SIDE was $\nu_\alpha(t) \leq 10$. (In Subsection 4.4.3, we will treat the situation when the number of edges is a random variable, rather than a known parameter.)

The pointwise log-likelihoods (4.6) in this case are

$$u_i^0 = y_i - \frac{1}{2}(\theta_1 + \theta_0). \tag{4.8}$$

(Note that, if $\mathbf{u}(t)$ is the solution to the SIDE with the initial condition \mathbf{u}^0 of (4.8), and $\mathbf{u}'(t)$ is the solution to the SIDE with the initial condition $\mathbf{u}'(0) = \mathbf{y}$, then $\mathbf{u}'(t) = \mathbf{u}(t) + \alpha'$, where

$$\alpha' = \frac{1}{2}(\theta_1 + \theta_0), \tag{4.9}$$

and therefore the zero-crossings of $\mathbf{u}(t)$ coincide with the α' -crossings of $\mathbf{u}'(t)$. Consequently, we can simply evolve the SIDE with the data \mathbf{y} as the initial condition, and look at its α' -crossings.)

Figure 4.3, from top down, depicts the true segmentation with ten edges, a corresponding observation \mathbf{y} , and the edges detected by the SIDE (the bottom plot will be explained in the next subsection). Note that the result is extremely accurate, despite the fact that the data is very noisy. The computations took 0.25 seconds on a Sparc Ultra 1, thanks to the fast implementation described in Subsection 4.3.1. ■

Example 4.5. *Changes in variance in a Gaussian random vector.*

Now $f(y, \theta_j)$ is a zero-mean Gaussian density with standard deviation θ_j ; $\theta_0 = 1$ and $\theta_1 = 1.5$. The changes between the two are at the same locations as the jumps in the previous example (see the top plot of Figure 4.3). The top plot of Figure 4.4 shows an observation \mathbf{y} . Again, we assume that the number of changes is known. The bottom plot of Figure 4.4 shows the changes detected by the SIDE, depicted as a binary sequence of θ_0 's and θ_1 's. In addition to being very accurate, the computations took just 0.25 seconds. ■

■ **4.4.2 Two Gaussian Distributions with Unknown Means.**

Suppose that $f(y, \theta_j)$ is the Gaussian density with mean θ_j and variance σ^2 . Let \mathbf{h} be an hypothesis, and let \mathbf{Y} be a sequence of N random variables which are conditionally

independent given \mathbf{h} , with the i -th random variable Y_i having conditional pdf $f(y, \theta_{h_i})$. Let ν be an upper bound on the number of edges in \mathbf{h} . Let K be the number of zeros in \mathbf{h} , and define $\sigma_1 = \frac{\sigma}{\theta_1 - \theta_0} \sqrt{N}$. Let the prior knowledge be as follows:

θ_0 and \mathbf{h} are unknown;

σ , σ_1 , and ν are known;

K is a random variable with the following discrete Gaussian probability mass function:

$$\text{pr}(K = k) = C \exp \left\{ -\frac{1}{2} \left(\frac{k - \frac{N}{2}}{\sigma_1} \right)^2 \right\}, \quad k = 1, \dots, N-1, \quad (4.10)$$

where C is a normalization constant.

We stress here that, even though our model assumes the knowledge of σ and σ_1 , we will shortly see that the SIDE never uses them in computing the optimal hypothesis; hence the title of this subsection is justified. The only parameter required by the SIDE is ν . We also point out that the distribution (4.10) is a reasonable one: it assigns larger probabilities to the hypotheses with roughly as many zeros as ones. The reason for choosing this particular form of the distribution is to make the generalized likelihood simplify to the statistic of Proposition 4.2. This implies that the SIDE can be used to find the optimal solution. More precisely, given an observation \mathbf{y} of \mathbf{Y} , we seek the best hypothesis in the generalized likelihood ratio sense [66]: the maximum likelihood estimates of the hypothesis and θ_0 are calculated for each value of K , and these estimates are then used in a multiple hypothesis testing procedure to estimate K . In other words, we seek

$$(\hat{\mathbf{h}}, \hat{\theta}_0, \hat{K}) = \arg \max_{\mathbf{h}, \theta_0, k} (\log f_1(\mathbf{y}|\mathbf{h}, \theta_0, k) + \log \text{pr}(K = k)),$$

where f_1 is the conditional pdf of \mathbf{Y} . After simplifying this formula, we obtain that $\hat{\mathbf{h}}$ must maximize

$$\begin{aligned} \phi(\mathbf{y}, \mathbf{h}) &\stackrel{\text{def}}{=} \mathbf{h}^T \mathbf{y} - \frac{N-k}{N} \sum_{i=1}^N y_i \\ &= \mathbf{h}^T \left(\mathbf{y} - \mathbf{a} \left(\frac{1}{N} \sum_{i=1}^N y_i \right) \right), \end{aligned}$$

where $\alpha(x) = x$, and, as in Proposition 4.2, $\mathbf{a}(x) = (\alpha(x), \dots, \alpha(x))^T \in \mathbb{R}^N$. Thus, according to Proposition 4.2, in order to find $\hat{\mathbf{h}}$, one has to evolve the SIDE whose initial condition is the observed signal: $\mathbf{u}^0 = \mathbf{y}$. The α -crossings of the solution $\mathbf{u}(t)$ will then coincide with the optimal edges, where

$$\alpha = \frac{1}{N} \sum_{i=1}^N u_i(t). \quad (4.11)$$

Thus, the only difference from Example 4.4 is that the threshold α' (4.9) is unknown, since $\frac{1}{2}(\theta_1 + \theta_0)$ is unknown. The threshold α (4.11) can be considered as an estimate of α' . If the underlying signal has roughly as many samples with mean θ_0 as ones with mean θ_1 , then α is a good estimate of α' , and we expect the estimates of the edge locations to be comparable to those in Example 4.4—i.e., despite less knowledge, the optimal estimates of the edge locations in this example would be similar to the optimal estimates of Example 4.4. This is confirmed by the experimental result for the data of Example 4.4, shown in the bottom plot of Figure 4.3, which is still very good and differs from the result of Example 4.4 in only two pixels out of the thousand. If the number of samples with mean θ_0 greatly differs from $\frac{N}{2}$, we would expect α to be a poor estimate of α' , which will lead to larger errors in the optimal estimates of edge locations. This situation, however, is of low probability, according to our model (4.10).

■ 4.4.3 Random Number of Edges and the Mumford-Shah Functional.

As in the two previous subsections, let \mathbf{y} be an observation of a sequence of N random variables, with conditional densities $p(y_i|\mathbf{h}) = f(y_i, \theta_{h_i})$, where $h_i \in \{0, 1\}^N$. Given \mathbf{h} , the y_i 's are conditionally independent. Let the number ν of edges in \mathbf{h} be a random variable with a probability mass function $p_\nu(\bar{\nu})$, $\bar{\nu} = 0, 1, \dots, N - 1$. To find the maximum *a posteriori* estimate of ν , we need to maximize the conditional probability of ν , which depends on the unknown \mathbf{h} . We therefore, as in the previous subsection, use the generalized likelihood strategy [66]: the maximum likelihood estimates of \mathbf{h} are calculated for each value of ν , and then these estimates are used in a multiple hypothesis test to estimate ν . In other words, we pick \mathbf{h} to maximize

$$\sum_{i=1}^N \log p(y_i|\mathbf{h}) + \log p_\nu(\bar{\nu}). \quad (4.12)$$

We saw in Subsection 4.4.1 that the \mathbf{h} -dependent part of the likelihood term is $\mathbf{h}^T \mathbf{u}^0$, where \mathbf{u}^0 is the sequence of log-likelihood ratios. Therefore, maximizing (4.12) is equivalent to minimizing the following statistic $\psi(\mathbf{u}^0, \mathbf{h})$:

$$\psi(\mathbf{u}^0, \mathbf{h}) = -\mathbf{h}^T \mathbf{u}^0 - \log p_\nu(\bar{\nu}). \quad (4.13)$$

We now relate (4.13) to the SIDE.

Proposition 4.5. *Suppose that p_ν is monotonically non-increasing: $p_\nu(0) \geq p_\nu(1) \geq \dots \geq p_\nu(N - 1)$. Then the minimizer of $\psi(\mathbf{u}^0, \cdot)$ (4.13) can be computed using the SIDE, with post-processing whose computational complexity is $O(N)$.*

Proof. Let \mathbf{h}_ψ be the hypothesis which minimizes $\psi(\mathbf{u}^0, \cdot)$, and, as previously, let

$$\mathbf{h}_{\leq \bar{\nu}}^*(\mathbf{u}^0), \text{ for } \bar{\nu} = 0, 1, \dots, N - 1 \quad (4.14)$$

be the hypothesis which achieves the maximal $\mathbf{h}^T \mathbf{u}^0$ among all the hypotheses with $\bar{\nu}$ or fewer edges. Suppose we could show that \mathbf{h}_ψ is actually one of the hypotheses (4.14).

Then we could compute \mathbf{h}_ψ as follows: run the SIDE to compute the N hypotheses (4.14), compute $\psi(\mathbf{u}^0, \cdot)$ for each of them, and pick the hypothesis which results in the smallest $\psi(\mathbf{u}^0, \cdot)$. To complete the proof, we now show that \mathbf{h}_ψ is indeed one of the hypotheses (4.14).

Let us fix an arbitrary hypothesis $\bar{\mathbf{h}}$ with $\bar{\nu}$ edges, and let $\nu^* \leq \bar{\nu}$ be the number of edges in the hypothesis $\mathbf{h}_{\leq \bar{\nu}}^*(\mathbf{u}^0)$. Then, by the definition of $\mathbf{h}_{\leq \bar{\nu}}^*(\mathbf{u}^0)$, we have:

$$\{\mathbf{h}_{\leq \bar{\nu}}^*(\mathbf{u}^0)\}^T \mathbf{u}^0 \geq \bar{\mathbf{h}}^T \mathbf{u}^0. \quad (4.15)$$

The monotonicity of p_ν implies

$$\log p_\nu(\nu^*) \geq \log p_\nu(\bar{\nu}). \quad (4.16)$$

Summing the two inequalities (4.15) and (4.16), we get:

$$\psi(\mathbf{u}^0, \mathbf{h}_{\leq \bar{\nu}}^*(\mathbf{u}^0)) \leq \psi(\mathbf{u}^0, \bar{\mathbf{h}}).$$

In other words, for an arbitrary hypothesis $\bar{\mathbf{h}}$, we found an hypothesis from among (4.14) which results in a smaller (or equal) $\psi(\mathbf{u}^0, \cdot)$. Therefore, the optimal hypothesis \mathbf{h}_ψ is among (4.14). \blacksquare

The second main result of this subsection is that having the exponential distribution p_ν is equivalent to specifying a stopping rule for the SIDE.

Proposition 4.6. *Let*

$$p_\nu(\bar{\nu}) = \frac{e^{-\lambda} - 1}{e^{-\lambda N} - 1} e^{-\lambda \bar{\nu}}, \text{ for } \bar{\nu} = 0, 1, \dots, N - 1, \quad (4.17)$$

and let \mathbf{h}_ψ be the hypothesis which minimizes $\psi(\mathbf{u}^0, \cdot)$ (4.13). Then the algorithm of Subsection 4.3.1, with a modified stopping rule, will produce \mathbf{h}_ψ . The new stopping criterion is:

$$\min_{(i,j) \text{ is a region of } \mathbf{u}(t)} E_{ij}(t) > \lambda,$$

where $E_{ij}(t)$ is defined as in (4.5).

Proof. Substituting p_ν (4.17) into (4.13), we see that minimizing $\psi(\mathbf{u}^0, \mathbf{h})$ with respect to \mathbf{h} is equivalent to minimizing the following function $\eta(\mathbf{h})$:

$$\eta(\mathbf{h}) = -\mathbf{h}^T \mathbf{u}^0 + \lambda \bar{\nu}. \quad (4.18)$$

Suppose that the solution to the SIDE has $\bar{\nu} + 1$ zero crossings at some time instant t , and call the hypothesis generated by these zero crossings \mathbf{h}_1 . Let the next region to be removed be (i^*, j^*) , and call the hypothesis resulting from its removal \mathbf{h}_2 . Let us denote by $E^*(t)$ the energy of the region (i^*, j^*) :

$$E^*(t) = \min_{(i,j) \text{ is a region of } \mathbf{u}(t)} E_{ij}(t).$$

In order to determine which hypothesis is better with respect to $\eta(\mathbf{h})$, we will look at $\eta(\mathbf{h}_2) - \eta(\mathbf{h}_1)$. First note that

$$(\mathbf{h}_2 - \mathbf{h}_1)^T \mathbf{u}^0 = - \left| \sum_{n=i^*}^{j^*} u_n^0 \right| = -\rho_{i^*,j^*} E^*(t),$$

and therefore

$$\eta(\mathbf{h}_2) - \eta(\mathbf{h}_1) = -(\mathbf{h}_2 - \mathbf{h}_1)^T \mathbf{u}^0 + \lambda(\bar{\nu} + 1 - \rho_{i^*,j^*}) - \lambda(\bar{\nu} + 1) = \rho_{i^*,j^*}(E^*(t) - \lambda).$$

So, if

$$E^*(t) > \lambda, \tag{4.19}$$

we prefer \mathbf{h}_1 to \mathbf{h}_2 —i.e., it is better not to remove the region (i^*, j^*) ; otherwise, \mathbf{h}_2 is better than \mathbf{h}_1 —i.e., we prefer to merge. Note that, by definition, $E^*(t)$ is a non-decreasing function of time, and so if (4.19) holds for some t , it will also hold for all future times. Also, if (4.19) is violated at some t , it is also violated for all past times. So, (4.19) is indeed a stopping rule.

The proof is not over, however, due to the fact that, as we remarked in Section 4.3, not every optimal hypothesis is generated by the SIDE. Therefore, we still need to rule out the possibility that the SIDE skips over the optimal hypothesis \mathbf{h}_ψ . In other words, we need to make sure that the following situation does not occur:

- $\mathbf{h}_\psi = \mathbf{h}_{\leq \bar{\nu}}^*(\mathbf{u}^0) \neq \mathbf{h}_{\leq \bar{\nu}-1}^*(\mathbf{u}^0)$, and
- $\rho_{i^*,j^*} = 2$, i.e., the SIDE's solution goes from $\bar{\nu} + 1$ zero-crossings directly to $\bar{\nu} - 1$.

The proof that this never happens is conceptually similar to the argument we made above when comparing \mathbf{h}_1 and \mathbf{h}_2 . Since it is rather technical, we give it in Appendix B. The final result is that, if the solution to the SIDE goes from $\bar{\nu} + 1$ zero-crossings directly to $\bar{\nu} - 1$, skipping $\mathbf{h}_{\leq \bar{\nu}}^*(\mathbf{u}^0)$, then $\mathbf{h}_{\leq \bar{\nu}}^*(\mathbf{u}^0)$ cannot be \mathbf{h}_ψ . Therefore, the algorithm of Subsection 4.3.1 with the stopping rule (4.19) will find \mathbf{h}_ψ . ■

The importance of Proposition 4.6 stems from the fact that it provides the stopping rule for the SIDE (4.1) when the probabilistic model for the number of changes is (4.17). A more difficult question, which is not addressed in this thesis, is the following. Suppose it is known that the probability mass function of ν is of the form (4.17), but the parameter λ is free to be chosen. How can one adaptively select λ ? One could approach this question by looking at the relative sizes of the energies $E^*(t)$ to decide at what time t the evolution starts to eliminate significant regions, rather than just those due to noise. This problem is conceptually similar to choosing the order of a model, and therefore various known techniques can be employed, like the minimum description length principle [53].

We conclude this section by showing that a specific conditional density $p(y_i|\mathbf{h})$ turns (4.18) into a special 1-D version of the Mumford-Shah functional [44]. It was proposed

in [44] to estimate a piecewise-smooth function u from its noisy observations y by minimizing over u and Γ the following cost:

$$\frac{1}{2} \int_{\mathbb{R}^2} (u - y)^2 + \gamma \int_{\mathbb{R}^2 \setminus \Gamma} |\nabla u|^2 + \lambda \bar{\nu}, \quad (4.20)$$

where Γ are the edges, i.e. the set on which u is discontinuous; $\bar{\nu}$ is the total length of the edges; and γ and λ are constants which control the smoothness of u within regions and the total length of the edges, respectively. If an approximation u is sought which is constant within each region [36, 43], the second term disappears. In 1-D, the integration is over \mathbb{R}^1 , and $\bar{\nu}$ is simply the number of the discontinuities in u . Assuming that we seek a piecewise-constant approximation, we discretize the 1-D version of (4.20):

$$\frac{1}{2} (\mathbf{u} - \mathbf{y})^T (\mathbf{u} - \mathbf{y}) + \lambda \bar{\nu}. \quad (4.21)$$

If one is looking for a binary approximation $\mathbf{u} = \mathbf{h} \in \{0, 1\}^N$, then $\mathbf{h}^T \mathbf{h} = \sum_{i=1}^N h_i$, and so if we define

$$u_i^0 = y_i - \frac{1}{2}, \quad (4.22)$$

then minimizing (4.21) is equivalent to minimizing $\eta(\mathbf{h})$ (4.18). We note that (4.22) defines the log-likelihood ratios for the situation when $p(y_i|\mathbf{h})$ is the Gaussian density with unit variance and mean h_i . Indeed, in this case

$$\begin{aligned} \log p(y_i|h_i = 1) - \log p(y_i|h_i = 0) &= -\frac{1}{2}(y_i - 1)^2 + \frac{1}{2}y_i^2 \\ &= y_i - \frac{1}{2}. \end{aligned}$$

We have just shown the following.

Proposition 4.7. *If*

$$\begin{aligned} p_\nu(\bar{\nu}) &= \frac{e^{-\lambda} - 1}{e^{-\lambda N} - 1} e^{-\lambda \bar{\nu}}, \text{ for } \bar{\nu} = 0, 1, \dots, N - 1, \text{ and} \\ p(y_i|\mathbf{h}) &= \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}(y_i - h_i)^2}, \text{ for } i = 1, \dots, N, \end{aligned}$$

then the generalized likelihood function is (4.18), which is
a) a special case of the Mumford-Shah functional for 1-D signals, and
b) according to Proposition 4.6, is optimized by the SIDE. ■

■ 4.5 Alternative Implementations.

■ 4.5.1 Dynamic Programming.

The problem which is solved by the SIDE—namely, that of finding the optimal hypothesis $\mathbf{h}_{\leq \nu}^*(\mathbf{u}^0)$ with respect to the statistic ϕ of Proposition 4.2—can also be solved using dynamic programming. The basic idea is to scan the samples of \mathbf{u}^0 from left to right, and notice that the best hypothesis for the first $i + 1$ samples can be easily calculated from the best hypothesis for the first i samples. More precisely, let $\mathbf{u}^{0i} = (u_1^0, u_2^0, \dots, u_i^0)^T$ be the first i samples of \mathbf{u}^0 . Let $\phi(\mathbf{u}^{0i}, \mathbf{h}) = \mathbf{h}^T(\mathbf{u}^{0i} - \mathbf{a})$, where $\mathbf{h} \in \{0, 1\}^i$ and $\mathbf{a} = (\alpha, \dots, \alpha)^T \in \mathbb{R}^i$. Let $A_{i\bar{\nu}p}$, for $i = 1, \dots, N$, $\bar{\nu} = 0, \dots, i - 1$, and $p = 0, 1$, be the set of all hypotheses $\mathbf{h} \in \{0, 1\}^i$ which have exactly $\bar{\nu}$ edges, and which end with p (i.e., $h_i = p$). Let $\mathbf{h}_{i\bar{\nu}p}$ be the best hypothesis in the set $A_{i\bar{\nu}p}$. Since ϕ is additive, $\mathbf{h}_{i+1, \bar{\nu}, p}$ is either $(\mathbf{h}_{i\bar{\nu}p}^T, p)^T$ or $(\mathbf{h}_{i, \bar{\nu}-1, 1-p}^T, p)^T$. Therefore, once we have computed and stored $\mathbf{h}_{i\bar{\nu}p}$ and the corresponding values of ϕ for $\bar{\nu} = 0, \dots, \nu$, and $p = 0, 1$, it will take us one comparison to compute $\mathbf{h}_{i+1, \bar{\nu}, p}$ for each $\bar{\nu}$ and p . Thus, to go from $\mathbf{h}_{i, \bar{\nu}, p}$ to $\mathbf{h}_{i+1, \bar{\nu}, p}$, we will need $O(\nu)$ computations. The total computational complexity is therefore $O(\nu N)$. The required memory is also $O(\nu N)$. Thus, if ν is smaller than $\log N$, the dynamic programming approach is faster than the SIDE implemented as the region merging method of Subsection 4.3.1. If ν is comparable to $\log N$, the two methods are similar in speed, but the SIDE wins in terms of memory use, since it only requires $O(N)$ memory. Finally, if ν is larger than $\log N$, the SIDE wins both in complexity and memory. Another important advantage of the SIDE is that it generalizes to 2-D—as shown in Section 4.7, while the dynamic programming algorithm does not.

■ 4.5.2 An Equivalent Linear Program.

We now show that the problem of minimizing η (4.18), considered in Section 4.4, is a linear programming problem.

Let D be the $(N - 1)$ -by- N first-difference matrix. Then, for any hypothesis $\mathbf{h} \in \{0, 1\}^N$, we have that the ℓ^1 -norm of its first difference $\|D\mathbf{h}\|_1$ is exactly equal to the number of edges in \mathbf{h} . Therefore, we can re-write the problem of minimizing η (4.18) as follows:

$$\begin{aligned} & \text{minimize} && -\mathbf{h}^T \mathbf{u}^0 + \lambda \|D\mathbf{h}\|_1, && (4.23) \\ & \text{subject to} && \mathbf{h} \in \{0, 1\}^N. \end{aligned}$$

This problem is equivalent to the following linear program:

$$\text{minimize} \quad -\mathbf{h}^T \mathbf{u}^0 + \lambda \sum_{i=1}^{N-1} r_i \quad (4.24)$$

$$\text{subject to} \quad r_i \geq h_{i+1} - h_i \quad \text{for } i = 1, \dots, N - 1 \quad (4.25)$$

$$r_i \geq h_i - h_{i+1} \quad \text{for } i = 1, \dots, N - 1 \quad (4.26)$$

$$0 \leq h_i \leq 1 \quad \text{for } i = 1, \dots, N, \quad (4.27)$$

where we replaced the requirement that \mathbf{h} be binary with a seemingly less restrictive condition that \mathbf{h} belong to the unit hypercube $[0, 1]^N$. Constraints (4.25) and (4.26) mean that $r_i \geq |h_{i+1} - h_i|$. On the other hand, (4.24) means that r_i must be as small as possible, and therefore $r_i = |h_{i+1} - h_i|$. The fact that the constraints (4.27) are equivalent to $\mathbf{h} \in \{0, 1\}^N$ is a little less obvious; it is verified in Appendix B (Section B.5). We point out that any generic linear programming algorithm will lose out in speed to the SIDE, because the SIDE exploits the special structure of the problem (4.23).

■ 4.6 Performance Analysis.

This section is devoted to the performance analysis of the SIDE, for signals with one change in mean. We first consider the event that the detected change location is farther than a certain number of samples from the actual one. An expression for the lower bound of the probability of this event is derived in Subsection 4.6.1; it is applied to the analysis of signals with additive white Gaussian noise in Subsection 4.6.2. We address the robustness of our algorithm in Subsection 4.6.3, by showing its optimality with respect to a certain H_∞ -like performance criterion.

■ 4.6.1 Probability Bounds.

This section is devoted to the analysis of the example in Subsection 4.4.2, when the number of edges is equal to one. That is, in order to maximize

$$\phi(\mathbf{u}^0, \mathbf{h}) = \mathbf{h}^T \mathbf{u}^0 - \frac{N-k}{N} \sum_{i=1}^N u_i^0,$$

we evolve the SIDE (4.1) until exactly one α -crossing remains, where $\alpha = \frac{1}{N} \sum_{i=1}^N u_i^0$. We denote the correct hypothesis \mathbf{h}^c and the correct location of the edge g^c . Without loss of generality, we assume that the first g^c samples of \mathbf{u}^0 have mean θ_0 , the last $N - g^c$ samples have mean θ_1 , and that $d \stackrel{\text{def}}{=} \theta_1 - \theta_0 > 0$. We denote the detected edge location g^* , its sign z^* , the corresponding hypothesis \mathbf{h}^* , and the number of zeros in it k^* : if $z^* = 1$, then $k^* = g^*$; if $z^* = -1$, then $k^* = N - g^*$.

Pick two integers, p_0 and q_0 , satisfying $1 \leq p_0 \leq g^c \leq q_0 \leq N - 1$, so that the location g^c of the true edge is between p_0 and q_0 . The goal of this section is to compute a lower bound for the probability of the event

$$\{p_0 \leq g^* \leq q_0, z^* = 1\}, \quad (4.28)$$

which says that the detected edge location g^* is between p_0 and q_0 , and that the detected sign of the edge is correct. The strategy will be to find a lower bound for the probability of a simpler event which implies (4.28). Specifically, suppose that $g^* > g^c$ and $z^* = 1$. Then

$$\phi(\mathbf{u}^0, \mathbf{h}^*) - \phi(\mathbf{u}^0, \mathbf{h}^c) = \sum_{i=g^c+1}^{g^*} (-u_i^0 + \alpha).$$

Since \mathbf{h}^* is the optimal hypothesis, the above expression has to be positive. Thus, if

$$\sum_{i=g^c+1}^q (-u_i^0 + \alpha) < 0 \text{ for } q = q_0 + 1, \dots, N, \quad (4.29)$$

then g^* cannot be larger than q_0 . Similarly, if

$$\sum_{i=p+1}^{g^c} (u_i^0 - \alpha) < 0 \text{ for } p = 1, \dots, p_0 - 1, \quad (4.30)$$

then g^* cannot be smaller than p_0 . Similar arguments show that, if

$$\sum_{i=g^c+1}^N (-u_i^0 + \alpha) + \sum_{i=q}^N (-u_i^0 + \alpha) < 0 \text{ for } q = g^c + 1, \dots, N, \text{ and} \quad (4.31)$$

$$\sum_{i=1}^{g^c} (u_i^0 - \alpha) + \sum_{i=1}^p (u_i^0 - \alpha) < 0 \text{ for } p = 1, \dots, g^c, \quad (4.32)$$

then the detected sign is correct, i.e. $z^* = 1$. Thus, the simultaneous occurrence of the events (4.29)-(4.32) implies (4.28). If α were not random, the expressions in (4.29)-(4.32) would be sums of independent identically distributed random variables, and therefore we would be able to employ results from the theory of random walks. We will remove the randomness of α from (4.29)-(4.32) by introducing a non-random bound on how far α can be from its mean

$$m \stackrel{\text{def}}{=} \frac{1}{N}(g^c\theta_0 + (N - g^c)\theta_1).$$

In other words, suppose that there are two positive real numbers, δ_1 and δ_2 , such that

$$m - \delta_1 \leq \alpha \leq m + \delta_2. \quad (4.33)$$

Then each of the inequalities

$$\sum_{i=g^c+1}^q (-u_i^0 + m + \delta_2) < 0 \text{ for } q = q_0 + 1, \dots, N \quad (4.34)$$

implies the corresponding inequality in (4.29). Let us call A_q the event that the q -th inequality in (4.34) holds, for $q = q_0 + 1, \dots, N$. We shall similarly bound the events (4.30), by defining events A_p whose intersection implies (4.30):

$$A_p = \left\{ \sum_{i=p+1}^{g^c} (u_i^0 - (m - \delta_1)) < 0 \right\}, \text{ for } p = 1, \dots, p_0 - 1. \quad (4.35)$$

We shall call A'_q and A'_p the events which imply (4.31) and (4.32), respectively:

$$A'_q = \left\{ \left(\sum_{i=g^c+1}^N + \sum_{i=q}^N \right) (-u_i^0 + m + \delta_2) < 0, \right\} \text{ for } q = g^c + 1, \dots, N; \quad (4.36)$$

$$A'_p = \left\{ \left(\sum_{i=1}^{g^c} + \sum_{i=1}^p \right) (u_i^0 - (m - \delta_1)) < 0 \right\}, \text{ for } p = 1, \dots, g^c. \quad (4.37)$$

Let ε_1 be the union (upper) bound for the probability of $\cup_{p=1}^{g^c} \overline{A'_p}$, where the overbar denotes the complement:

$$\varepsilon_1 = \sum_{p=1}^{g^c} \Pr(\overline{A'_p}). \quad (4.38)$$

Suppose further that p_1 is a lower bound for the probability of the intersection of the events A_p :

$$p_1 \leq \Pr \left(\bigcap_{p=1}^{p_0-1} A_p \right). \quad (4.39)$$

Then

$$\Pr \left(\bigcap_{i=1}^{p_0-1} A_p \cap \left(\bigcap_{i=1}^{g^c} A'_p \right) \right) = \Pr \left(\bigcap_{i=1}^{p_0-1} A_p \cap \left(\overline{\bigcup_{i=1}^{g^c} \overline{A'_p}} \right) \right) \quad (4.40)$$

$$= \Pr \left(\bigcap_{i=1}^{p_0-1} A_p \right) - \Pr \left(\bigcap_{i=1}^{p_0-1} A_p \cap \left(\bigcup_{i=1}^{g^c} \overline{A'_p} \right) \right) \quad (4.41)$$

$$\geq p_1 - \Pr \left(\bigcup_{i=1}^{g^c} \overline{A'_p} \right) \quad (4.42)$$

$$\geq p_1 - \varepsilon_1, \quad (4.43)$$

where we used the identity $\overline{\bigcap A'_p} = \bigcup \overline{A'_p}$ in (4.40), the identity $\Pr(A \cap B) = \Pr(A) - \Pr(A \cap \overline{B})$ in (4.41), and the inequality $-\Pr(A \cap \overline{B}) \geq -\Pr(\overline{B})$ in (4.42). Similarly, the probability of the intersection of (4.34) and (4.36) is bounded from below by $p_2 - \varepsilon_2$, where

$$\varepsilon_2 = \sum_{q=g^c+1}^N \Pr(\overline{A'_q}),$$

$$p_2 \leq \Pr \left(\bigcap_{q=q_0+1}^N A_q \right).$$

Finally, if we denote by $1 - \varepsilon$ the probability of (4.33), we have that the intersection of the events (4.33)-(4.37) is lower-bounded by

$$(p_1 - \varepsilon_1)(p_2 - \varepsilon_2) - \varepsilon. \quad (4.44)$$

We showed earlier in this section that the intersection of these events implies the intersection of the events (4.29)-(4.32), which, in turn, implies the event (4.28). Thus, the above expression (4.44) is a lower bound for the probability of the event (4.28).

In [28], asymptotic probabilities of the events (4.28) are computed, for $N \rightarrow \infty$ and $g^c \rightarrow \infty$. When α is non-random (as in, e.g., our Examples 4.4 and 4.5 of Subsection 4.4.1), these asymptotic probabilities are also (non-asymptotic) lower bounds. In the process of computing these, lower bounds p_1 (4.39) and p_2 are also computed in [28]; these are asymptotically tight. In the next subsection, we describe a different method for computing p_1 and p_2 for the Gaussian case (i.e. when the model of Subsection 4.4.2 applies). Our method produces looser bounds than [28]; however, the derivation is conceptually much simpler and leads to easier computations.

■ 4.6.2 White Gaussian Noise.

Proposition 4.8. *Suppose that the following model applies:*

$$\mathbf{u}^0 = \mathbf{m} + \mathbf{w},$$

where $\mathbf{m} = (\theta_0, \dots, \theta_0, \theta_1, \dots, \theta_1)^T$ is the mean vector of \mathbf{u}^0 , and \mathbf{w} is zero-mean white Gaussian noise with variance σ^2 . Then

$$\begin{aligned} & Pr(p_0 \leq g^* \leq q_0 \text{ and } z^* = 1) \geq \\ & \left\{ \Phi \left(\frac{d_1 \sqrt{g^c - p_0 + 1}}{\sigma} \right) - \Phi \left(-\frac{d_1 \sqrt{g^c - p_0 + 1}}{\sigma} \right) - \sum_{p=1}^{g^c} \Phi \left(-\frac{d_1(p + g^c)}{\sigma \sqrt{3p + g^c}} \right) \right\} \times \\ & \times \left\{ \Phi \left(\frac{d_2 \sqrt{q_0 - g^c + 1}}{\sigma} \right) - \Phi \left(-\frac{d_2 \sqrt{q_0 - g^c + 1}}{\sigma} \right) - \sum_{q=g^c+1}^N \Phi \left(-\frac{d_2(2N - q - g^c)}{\sigma \sqrt{4N - 3q - g^c}} \right) \right\} - \\ & - \Phi \left(-\frac{\delta_2 \sqrt{N}}{\sigma} \right) - \Phi \left(-\frac{\delta_1 \sqrt{N}}{\sigma} \right), \end{aligned} \quad (4.45)$$

where

- Φ is the Gaussian cumulative distribution function;
- $d_1 = \frac{(N-g^c)d}{N} - \delta_1$;
- $d_2 = \frac{g^c d}{N} - \delta_2$;
- δ_1 and δ_2 are any positive real numbers such that $d_1 > 0$ and $d_2 > 0$.

Proof. The terms of this bound come from calculating the parameters p_1 , p_2 , ε_1 , ε_2 , and ε of the expression (4.44). We now present this calculation.

It follows from our noise model that $\alpha = \frac{1}{N} \sum_{i=1}^N u_i^0$ is Gaussian with mean $\frac{1}{N}(g^c \theta_0 + (N - g^c) \theta_1)$ and variance $\frac{\sigma^2}{N}$. Thus, the probability ε that (4.33) does not hold is

$$\varepsilon = 1 - \int_{-\delta_1}^{\delta_2} \mathcal{N}\left(0, \frac{\sigma^2}{N}\right) = \Phi\left(-\frac{\delta_1 \sqrt{N}}{\sigma}\right) + \Phi\left(-\frac{\delta_2 \sqrt{N}}{\sigma}\right), \quad (4.46)$$

where \mathcal{N} is the Gaussian density.

In order to find ε_1 , re-write the event $\overline{A'_p}$ as

$$2 \sum_{i=1}^p w_i + \sum_{i=p+1}^{g^c} w_i - (p + g^c) \left(\frac{(N - g^c)d}{N} - \delta_1 \right) > 0.$$

The sum of the noise samples is a zero-mean Gaussian random variable with variance $\sigma^2(4p + (g^c - p)) = \sigma^2(3p + g^c)$. Therefore, if we define

$$d_1 = \frac{(N - g^c)d}{N} - \delta_1,$$

then the probability of $\overline{A'_p}$ is

$$\Phi\left(-\frac{d_1(p + g^c)}{\sigma \sqrt{3p + g^c}}\right).$$

Substituting this into (4.38), we get:

$$\varepsilon_1 = \sum_{p=1}^{g^c} \Phi\left(-\frac{d_1(p + g^c)}{\sigma \sqrt{3p + g^c}}\right).$$

To find a lower bound p_1 (see Equation (4.39)), we define $S_0 = 0$ and

$$S_j = \frac{1}{\sigma} \sum_{i=g^c-j+1}^{g^c} w_i \text{ for } j = 1, \dots, g^c,$$

and note that the intersection of the events A_p of Equation (4.35) is equivalent to

$$S_j < \frac{d_1}{\sigma} j \text{ for } j = g^c - p_0 + 1, \dots, g^c - 1.$$

Also note that the S_j 's form the standard (discrete) Brownian motion [52], which can be viewed as a sampling of the standard continuous Brownian motion $S(t)$ at integer

times. We define $t_0 = g^c - p_0 + 1$, $s_0 = S(t_0)$, and $f(s_0)$ = the pdf of s_0 . Then

$$\begin{aligned} \Pr\left(\bigcap_{p=1}^{p_0-1} A_p\right) &= \Pr(S_j < \frac{d_1}{\sigma}j \text{ for } j = t_0, \dots, g^c - 1) \\ &\geq \Pr(S(t) < \frac{d_1}{\sigma}t \text{ for } t_0 \leq t \leq g^c - 1) \\ &= \int_{-\infty}^{\frac{d_1}{\sigma}t_0} \Pr\left(S(t) < \frac{d_1}{\sigma}t \text{ for } t > t_0 | s_0\right) f(s_0) ds_0. \end{aligned}$$

Conditioned on s_0 , the process $P(t) = S(t + t_0) - s_0$ is the standard Brownian motion for $t \geq 0$. Therefore, the integral above is equal to

$$\begin{aligned} \int_{-\infty}^{\frac{d_1}{\sigma}t_0} \Pr\left(P(t) < \frac{d_1}{\sigma}(t + t_0) - s_0 \text{ for } t > 0 | s_0\right) f(s_0) ds_0 &= \\ \int_{-\infty}^{\frac{d_1}{\sigma}t_0} \Pr\left(\sup_{t \geq 0} (P(t) - \frac{d_1}{\sigma}t) < \frac{d_1}{\sigma}t_0 - s_0 | s_0\right) f(s_0) ds_0. \end{aligned}$$

Given s_0 , $P(t) - \frac{d_1}{\sigma}t$ is a Brownian motion with drift $-\frac{d_1}{\sigma}$. If the drift is non-negative, then the probability inside the integral is zero [52]. We therefore assume that $d_1 > 0$, i.e., that

$$\delta_1 < \frac{(N - g^c)d}{N}.$$

Then the drift is negative, in which case the supremum is finite almost surely, and its probability distribution is [52]

$$1 - \exp\left(-2\frac{d_1}{\sigma}x\right) \text{ for } x \geq 0, \text{ and zero otherwise.}$$

Substituting this into the integral above, we get

$$\begin{aligned} \int_{-\infty}^{\frac{d_1}{\sigma}t_0} \left\{1 - \exp\left[-2\frac{d_1}{\sigma}\left(\frac{d_1}{\sigma}t_0 - s_0\right)\right]\right\} \frac{1}{\sqrt{2\pi t_0}} \exp\left(-\frac{s_0^2}{2t_0}\right) ds_0 &= \\ \Phi\left(\frac{d_1\sqrt{t_0}}{\sigma}\right) - \int_{-\infty}^{\frac{d_1}{\sigma}t_0} \frac{1}{\sqrt{2\pi t_0}} \exp\left(-\frac{(s_0 - \frac{2d_1 t_0}{\sigma})^2}{2t_0}\right) ds_0 &= \\ \Phi\left(\frac{d_1\sqrt{g^c - p_0 + 1}}{\sigma}\right) - \Phi\left(-\frac{d_1\sqrt{g^c - p_0 + 1}}{\sigma}\right). \end{aligned}$$

Combining the values for p_1 , ε_1 , and ε , obtained above, with similarly obtained values for p_2 and ε_2 (where $d_2 = \frac{g^c d}{N} - \delta_2$), we arrive at the expression (4.45). As we mentioned above, this bound is looser than those of [28]. For example, if N is very large, $\frac{g^c}{N} = 0.5$, $d = 3\sigma$, and $p_0 = q_0 = g^c$, then the asymptotic probability (from Table 3.3 of [28]) is 0.857, whereas our bound is 0.751.

■ 4.6.3 H_∞ -Like Optimality.

We continue the analysis of detecting one change in mean in a sequence of Gaussian random variables, but now we assume that both means are known, and therefore $\alpha = 0$, as in Example 4.4 of Subsection 4.4.1. We now show that the change location estimate produced by the SIDE—which in this case, according to Section 4.3, is the maximum likelihood estimate—is optimal according to an H_∞ -like criterion.

H_∞ estimation and control arose out of situations when there is no complete *a priori* knowledge of the system dynamics and of the statistical properties of the exogenous inputs. Model uncertainties and the lack of statistical information encountered in many applications have led to research in minimax estimation, producing so-called H_∞ algorithms which are robust to parameter variations [25, 26, 45, 61]. In these approaches, the quantity to be minimized is the H_∞ norm of the operator mapping the inputs to the desired function (either the output or a weighted error).

We will analyze the following problem.

Problem 4.2. *Let $d > 0$ be a known real number. Consider a step sequence $\mathbf{m}_{g^c} = (0, \dots, 0, d, \dots, d)^T$ of length N , whose first g^c entries are zeros. Let the observed signal be*

$$\mathbf{y} = \mathbf{m}_{g^c} + \mathbf{v},$$

where \mathbf{v} is an unknown disturbance. The objective is to estimate the location of change $g^c \in \{0, \dots, N\}$ ($g^c = 0$ and $g^c = N$ refer to the same hypothesis, corresponding to no change). ■

As stated in Section 4.3, if \mathbf{v} is a zero-mean white Gaussian noise, then the SIDE will find the ML estimate, i.e.

$$\hat{g}_{ML}^c = \arg \max_g \sum_{i=g+1}^N (y_i - \frac{d}{2}). \quad (4.47)$$

To analyze the robustness of this estimator, we define the following performance measure:

$$\mathcal{B}(f) = \sup_{g^c, \mathbf{v} \neq \mathbf{0}} \frac{|g^c - f(\mathbf{y})|}{\|\mathbf{v}\|_1},$$

where $f(\mathbf{y})$ is any estimator of g^c , and $\|\cdot\|_1$ stands for the ℓ^1 norm. Choosing the estimator which minimizes \mathcal{B} is similar in spirit to H_∞ estimation: we would like to minimize the worst possible error, over all possible disturbances. We presently show that the SIDE estimator (4.47) does minimize \mathcal{B} . This means that our estimator is robust: it has the best worst-case error performance among all estimators, and for all noise sequences.

We will prove the optimality of the SIDE estimator by showing two things:

- that $\mathcal{B}(f)$ is always larger than a certain constant (see Proposition 4.9 below), and
- that the SIDE estimator achieves this lower bound (see Proposition 4.10 below).

Proposition 4.9. *For any estimator f , $\mathcal{B}(f) \geq \frac{2}{d}$.*

Proof. Fix the noise level at $\|\mathbf{v}\|_1 = \frac{d}{2}$, and suppose that the observation is $\mathbf{y} = (0, \frac{d}{2}, d, \dots, d)^T$. The signal \mathbf{m}_{g^c} which resulted in this \mathbf{y} after adding noise of norm $\frac{d}{2}$ could be either $\mathbf{m}_1 = (0, d, d, \dots, d)^T$ or $\mathbf{m}_2 = (0, 0, d, \dots, d)^T$, in which cases $g^c = 1$ and $g^c = 2$, respectively. Thus,

$$\begin{aligned} \mathcal{B}(f) &\geq \sup_{g^c \in \{1,2\}} \frac{|g^c - f(\mathbf{y})|}{\frac{d}{2}} = \\ &= \frac{2}{d} \sup_{g^c \in \{1,2\}} |g^c - f(\mathbf{y})| = \\ &= \begin{cases} \frac{2}{d}|2 - f(\mathbf{y})| & \text{if } f(\mathbf{y}) = 1 \\ \frac{2}{d}|1 - f(\mathbf{y})| & \text{if } f(\mathbf{y}) \geq 2 \end{cases} \geq \\ &\geq \frac{2}{d}. \quad \blacksquare \end{aligned}$$

We will now show that the ML estimator achieves this bound.

Proposition 4.10. *For $\hat{g}_{ML}^c = f_{ML}(\mathbf{y})$, $\mathcal{B}(f_{ML}) = \frac{2}{d}$. Thus, the estimator \hat{g}_{ML}^c is optimal with respect to the criterion \mathcal{B} .*

Proof. Suppose that $\hat{g}_{ML}^c > g^c$. Then (4.47) implies

$$\begin{aligned} \sum_{i=\hat{g}_{ML}^c+1}^N (y_i - \frac{d}{2}) &> \sum_{i=g^c+1}^N (y_i - \frac{d}{2}) \Rightarrow \\ \sum_{i=g^c+1}^{\hat{g}_{ML}^c} (y_i - \frac{d}{2}) &< 0 \Rightarrow \\ \sum_{i=g^c+1}^{\hat{g}_{ML}^c} ((d + v_i) - \frac{d}{2}) &< 0 \Rightarrow \\ (\hat{g}_{ML}^c - g^c) \frac{d}{2} &< - \sum_{i=g^c+1}^{\hat{g}_{ML}^c} v_i \leq \|\mathbf{v}\|_1. \end{aligned}$$

Therefore, the smallest ℓ^1 norm of the disturbance required to create the error $\hat{g}_{ML}^c - g^c$ is $(\hat{g}_{ML}^c - g^c) \frac{d}{2}$. The calculations for $\hat{g}_{ML}^c < g^c$ are similar, and so

$$\mathcal{B}(f_{ML}) = \sup_{g^c, \mathbf{v} \neq 0} \frac{|g^c - f_{ML}(\mathbf{y})|}{\|\mathbf{v}\|_1} \leq \sup_{g^c} \frac{|g^c - f_{ML}(\mathbf{y})|}{|g^c - f_{ML}(\mathbf{y})| \frac{d}{2}} = \frac{2}{d}. \quad \blacksquare$$

■ 4.7 Analysis in 2-D.

In 1-D, there are always one fewer edges than regions. In 2-D, there is no such relationship between the total length of the boundaries and the total number of regions. From this difference stem two ways to generalize the SIDE (4.1) to 2-D. The first one is to evolve the intensity value inside each region according to

$$\dot{u}_i = \frac{1}{m_i} \operatorname{sgn}(\alpha - u_i) p_i, \quad (4.48)$$

where m_i is the number of pixels inside the i -th region, and p_i is the perimeter of the i -th region. (We still define the boundaries between regions as lines of α -crossings.) The second possible 2-D generalization of the SIDE has the same form, with p_i being the number of neighboring regions of the region i .

The 2-D SIDEs unfortunately do not find the global solution to the optimization problem of Proposition 4.2. However, we believe that a similar but weaker statement can be made—namely, that they find optimal coarsenings of the initial segmentation. We briefly repeat the definitions from Chapter 2 which are used in this statement.

A *segmentation* of a given image \mathbf{u}^0 is a partitioning of the domain Ω of its definition into several disjoint regions O_1, \dots, O_k ($\cup_{i=1}^k O_i = \Omega$). If a segmentation $E_1 = \cup O'_j$ can be obtained from a segmentation $E = \cup O_i$ by erasing edges, it is said that E_1 is *coarser* than E . More precisely, E_1 is *coarser* than E when for every region O_i of E there is a region O'_j of E_1 such that $O_i \subset O'_j$.

Conjecture 1. *As in Proposition 4.2, let $\phi(\mathbf{u}, \mathbf{h}) = \mathbf{h}^T(\mathbf{u} - \mathbf{a})$. Let E be the segmentation of \mathbf{u}^0 given by the α -crossings of \mathbf{u}^0 . Consider the SIDE (4.48) with \mathbf{u}^0 as the initial data and with p_i equal to the perimeter of the i -th region. Suppose the evolution is stopped when the total perimeter of the α -crossings is ν , and let E_1 be the segmentation given by these α -crossings. Then E_1 is the optimal coarsening of E , subject to the constraint that the segmentation have total edge length ν or less. ■*

Conjecture 2. *The situation is the same as in Conjecture 1, except that p_i is the number of neighbors of the i -th region. Suppose the evolution is stopped when the total number of regions is ν , and let E_1 be the resulting segmentation. Then E_1 is the optimal coarsening of E , subject to the constraint that the segmentation have ν or fewer regions. ■*

We note that, while (4.48) is the equation we have conjectures about, it is not the equation we use in practice for images. The reason is that thresholding the initial condition \mathbf{u}^0 with the threshold α typically leads to initial segmentations which are too coarse—that is, which have too few regions. Even if the evolution then provides the best coarsening of this initial segmentation, it may not be a good result. Better results are achieved when one first evolves the SIDE using (3.12), and then applies the threshold α to the image $\mathbf{u}(t)$. This is what was done in examples of Figure 4.5, which are experimental evidence of the fact that the algorithm works well and is very robust to degradations which do not conform well to the models of Section 4.3. The data on the left is a very blurry and noisy synthetic aperture radar image of two textures: forest

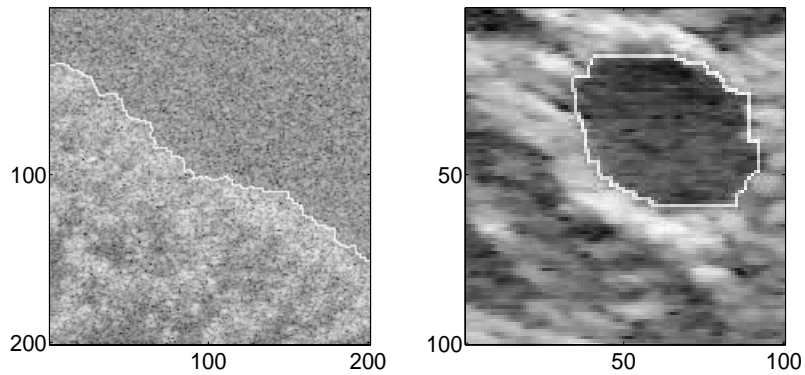


Figure 4.5. Edge detection in 2-D.

and grass. The pervasive speckle noise is inherent to this method of data collection. The algorithm was run on the raw data itself (which corresponds to assuming a Gaussian model with changes in mean—see Section 4.3), and stopped when two regions remained. The resulting boundary (shown superimposed onto the logarithm of the original image) is extremely accurate. The logarithm of a similarly blurry and noisy ultrasound image of a thyroid is shown on the right, with the boundary detected by the SIDE.