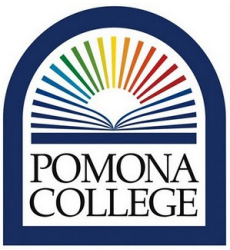
Pomona College



Senior Thesis in Mathematics

Detecting and Estimating Filamentary Structures in the Presence of Background Noise

Author:
Karl Kumbier

Advisor:
Dr. Jo Hardin

Submitted to Pomona College in Partial Fulfillment
of the Degree of Bachelor of Arts

April 4, 2013

Detecting and Estimating Filamentary Structures in the Presence of Background Noise

Karl Kumbier

April 4, 2013

Contents

1 Detecting Filaments **4**
- 1.1 Defining Anisotropic Strips . . . 4
- 1.2 Thresholds . . . 8
  - 1.2.1 Defining Counting Thresholds . . . 9
  - 1.2.2 Defining Length Thresholds . . . 9
  - 1.2.3 Defining Detection Thresholds . . . 9
- 1.3 Behavior of the Algorithm . . . 10

2 Estimation Methods **11**
- 2.1 Medial Axis of Support . . . 12
- 2.2 Path Density Gradient Field . . . 13
  - 2.2.1 Flows . . . 13
  - 2.2.2 Estimating Filaments with Path Densities . . . 13

3 Estimating Filaments **15**
- 3.1 Adapting Filament Detection . . . 15
  - 3.1.1 Covering the Filament . . . 16
  - 3.1.2 Mapping the Filament . . . 17

4 Results **18**

5 Conclusion **25**

Introduction

Imagine looking at a satellite image and trying to identify a filamentary structure such a road or river. In many cases, there will be a considerable amount of noise in the image from trees, buildings, or other sources obscuring parts of the filament. Furthermore, there is the possibility that while trying to sort through all of the noise, your eyes are tricked into recognizing a filament when none is actually present.

The preceding problem is analogous to the following. Consider the set $X_i \in \mathbb{R}^2$, $i \in 1, ..., n$, of data points uniformly distributed on the unit square. Suppose that a fraction of these points, which we denote ϵ_n, appear to lie on some function $f : [0, 1] \to [0, 1]$ *(Figure 1)*. We would like to have a systematic approach to distinguish between situations when there is a subset of points on f, and when there is not. If we determine that some fraction of points lie on f, we would also like to have the ability to estimate the function. This type of pattern recognition and estimation is a topic of great interest in the field of machine learning, with applications ranging from medical imaging of blood vessels to analyzing galaxy distribution throughout the universe [6].

A possible solution for filament detection, has been proposed by Arias-Castro *et al.* for functions that satisfy the Hölder condition,

$$|f(x) - f(y)| \leq \beta |x - y|^{\alpha}$$

where $\alpha \in (1, 2]$ and $\beta > 0$. Their method involves generating sets of anisotropic strips with constant area that cover the unit square, counting the data points in these strips, and finding runs of strips containing a significant number of points that are *good continuations*, which we define in Section 1.1. The authors use the length of these runs to test the hypotheses:

$$\begin{aligned} \mathbf{H}_0 : &\quad X_i \sim Uniform(0, 1)^2 \\ \mathbf{H}_1 : &\quad X_i \sim (1 - \epsilon_n) Uniform(0, 1)^2 + \epsilon_n Uniform(graph(f)) \end{aligned}$$

If a filament is present, their method generates a set of strips that covers f over its domain. This covering suggests the possibility of estimating the filament by taking the midline of each strip. However, the initial covering must use relatively large strips to ensure that the entire filament is contained in a run, which produces only a rough estimate of f, denoted $\hat{f}$. Additionally, $\hat{f}$ will be a piecewise linear estimate, and we assume that f is some smooth function. To correct for these problems, we propose applying Chandler *et al.'s* method for automatic locally adaptive smoothing of level sets [3]. The general idea is to use a kernel smoother on $\hat{f}$ to produce a smooth estimate, denoted $\tilde{f}$. We then map $\tilde{f}$ to a new space where it is easier to estimate, and iterate through the process until we have an accurate estimate of the filament.

While other estimation techniques have been suggested, the assumptions that they make either limit the types of filaments that they are able to estimate

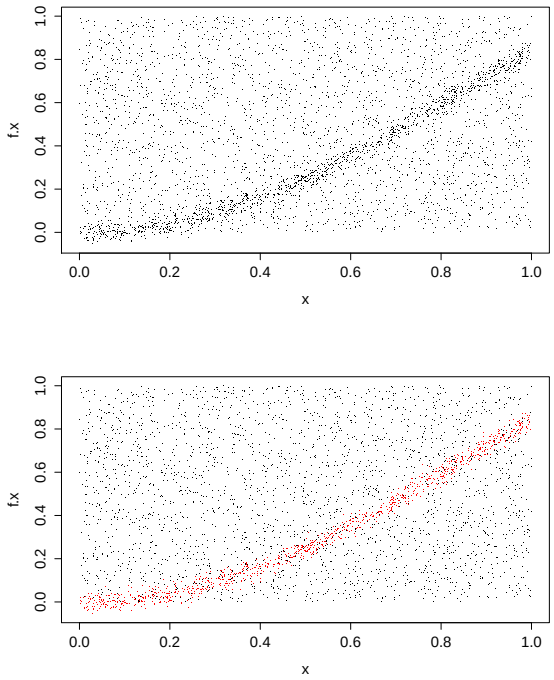


Figure 1: 2500 points uniformly distributed on the unit square with $\epsilon_n = 1000$ points uniformly distributed on the function $f(x) = x^2 + \epsilon \sim N(0, .025^2)$

[4] or produce estimates that are unsatisfactory[5]. We show that our proposed method has the potential to work for a wide variety of filaments, while producing an estimate that is smooth.

1 Detecting Filaments

Detecting filamentary structures in the midst of background noise is a challenging problem that depends on the number of observations, the fraction of points on the filament, and the length of the filament. The algorithm proposed by Arias-Castro *et al.* utilizes counting and length thresholds $N^*, L_n^* \in \mathbb{N}$ that we define in Section 1.2, and runs as follows:

1. Construct sets of regions, R, with constant area but varied width, thickness, and slopes, that cover the unit square.
2. Determine the number of data points contained within each region
$$N(R) = \#\{i : X_i \in R\}$$
3. Identify significant strips, which we define to be those that contain more points than our counting threshold N^*.
$$s(R) = \mathbb{1}_{\{N(R) > N^*\}}$$
4. Find the length of the longest path of significant strips that are *good continuations*, which we define as L_n^{max}
5. Compare the longest path to the decision threshold L_n^*.

$$\begin{aligned} L_n^{max} &\leq L_n^*, \text{ fail to reject } \mathbf{H}_0 \\ L_n^{max} &> L_n^*, \text{ reject } \mathbf{H}_0 \end{aligned}$$

1.1 Defining Anisotropic Strips

Central to the algorithm discussed above is the assumption that if $f \in$ Hölder(α, β) is present, the algorithm produces a run of strips that contains f. One can see that any run of strips that does not contain f over its entire domain would most likely contain strips that are not identified as significant. Because of this, we would probably fail to find a longest run exceeding L_n^* if the assumption did not hold. To ensure this covering property, we define our anisotropic strips in the following manner.

Let $J = \lceil log_2 n \rceil$ be the maximum value of the range $0 \leq j \leq J$ which the strips are indexed over, where $j \in \mathbb{N}$. The width and thickness of the strips are given by $w = 2^{-j}$ and $t = 2^{-(J-j)+1}$ respectively. The variables k, ℓ_1 and ℓ_2 control the slope and location of the strips, and S represents the maximum absolute slope that the algorithm can detect. If we let $\delta_1 = t/4$ and $\delta_2 = t/(4w)$,

the strip $R(j, k, \ell_1, \ell_2)$ will be centered at $c = ((k+1/2)w, \ell_1\delta_1)$ and have a slope of $s = \ell_2\delta_2$, where $0 \leq k < w^{-1}, 0 \leq \ell_1 \leq \delta_1^{-1}, -S\delta_2^{-1} \leq \ell_2 \leq S\delta_2^{-1}$ [1].

As a result of this construction, the strips have constant area $2/n$, and at each level of j, they have constant width and thickness. The center and slope of the strips are defined in such a way that given j, strips with identical slopes do not overlap each other horizontally, but do overlap vertically by $t/4$.

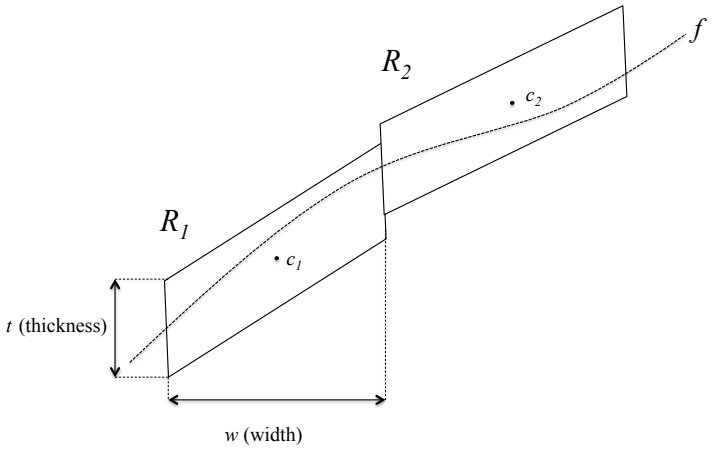


Figure 2: Anisotropic strips that are good continuations of each other. Regions $R_1(j, k_1, \ell_{1,1}, \ell_{1,2})$ and $R_2(j, k_2, \ell_{2,1}, \ell_{2,2})$ have centers $c_1 = ((k_1 + 1/2)w, \ell_{1,1}\delta_1)$ and $c_2 = ((k_2 + 1/2)w, \ell_{2,1}\delta_1)$ with slopes $s_1 = \ell_{1,2}\delta_2$ and $s_2 = \ell_{2,2}\delta_2$ respectively.

The strips are grouped into levels based on j, $\mathcal{R}(j) = \{R(j, k, \ell_1, \ell_2) : k, \ell_1, \ell_2\}$ of equal width and thickness, and organized into a directed graph $\mathcal{G}(j) = (\mathcal{V}(j), \mathcal{E}(j))$. For this graph, the vertices are given by the regions $\mathcal{R}(j) = \mathcal{V}(j)$ and the edges between these vertices represent strips that are *good continuations* of each other. We define strips as *good continuations* if the regions are horizontally adjacent, and have altitudes and slopes that are less than δ_1 and δ_2 apart respectively [1].

We claim that strips constructed in this manner efficiently cover Hölder(α, β) functions that satisfy the conditions $\alpha \in (1, 2]$ and $\beta > 0$. As evidence for this claim, we present the following three lemmas with proof.

Lemma 1.1 shows that there is some value j^* for which the sizes of the strips will be optimally associated with f. Lemma 1.2 shows that if we let $j = j^*$, given some interval of defined length $I_k = [kw, (k + 1)w)$ in the domain of f,

there is a $R(j^*, k, \ell_1, \ell_2)$ that contains f over the interval I_k. Lemma 1.3 shows that these regions can be combined into a run of strips that are *good continuations* of one another.

Lemma 1.1. *For any fixed (α, β) that satisfy $1 < \alpha \leq 2$, $\beta > 0$ we have that for sufficiently large n, there exists a $j^* = j^*(\alpha, \beta; n)$ such that:*

$$2\beta w^\alpha \leq t < 16\beta w^\alpha$$

Proof. Let us expand our notation for strip width and thickness, $w(j) = 2^{-j}$ and $t(j) = 2^{-(J-j)+1}$, such that each take real arguments. Define $j^+ = j^+(\alpha, \beta, n)$ so that

$$\begin{aligned} 2\beta w(j^+)^\alpha &= t(j^+) \\ 2\beta 2^{-\alpha j^+} &= 2^{-(J-j^+)+1} \end{aligned}$$

If we let $j^* = \lceil j^+ \rceil$ then $w(j^+)/2 \leq w(j^*) \leq w(j^+)$ and $t(j^+) \leq t(j^*) \leq 2t(j^+)$. By hypothesis, $1 < \alpha \leq 2$, $2^\alpha \leq 4$, giving us:

$$2\beta w(j^*)^\alpha \leq 2\beta w(j^+)^\alpha = t(j^+) \leq t(j^*) \leq 2t(j^+) = 4\beta w(j^+)^\alpha \leq 16\beta w(j^*)^\alpha$$

□

Here, we have shown that for a given Hölder(α, β) function f, there is some value of j, denoted j^*, for which the graph $\mathcal{G}(j^*)$ will be optimally associated with f. In other words, we have set a bound for $t(j^*)$ which we use in subsequent lemmas. We note that $j^* \leq J$ and as J is defined by n, we are only guaranteed this result for sufficiently large n.

Lemma 1.2. *Let $j = j^*(\alpha, \beta)$ and suppose f is a Hölder(α, β) function with a domain containing the interval $I_k = [kw, (k+1)w)$. Set $x_k = (k + 1/2)w$, let $\ell_{1,k}\delta_1$ be the closest multiple of δ_1 to $f(x_k)$ and $\ell_{2,k}\delta_2$ be the closest multiple of δ_2 to $f'(x_k)$. Then:*

$$graph(f|I_k) \subset R(j, k, \ell_{1,k}, \ell_{2,k})$$

We say that the region $R(j, k, \ell_{1,k}, \ell_{2,k})$ is associated to f on the interval I_k.

Proof. By definition, we have that $f \in$Hölder(α, β) satisfies $f : [0, 1] \to [0, 1]$ and

$$|f'(x) - f'(y)| \leq \alpha\beta|x - y|^{\alpha-1}, \qquad x, y \in [0, 1]$$

from whence it follows

$$|f(x) - f(y) - f'(y)(x - y)| \leq \beta|x - y|^\alpha \tag{1}$$

We define $f_k(x)$ to be a function with constant slope tangent to f at x_k. Using (1) with $I_k = [kw, (k + 1)w)$,

$$|f(x) - f_k(x)| \leq \beta(w/2)^\alpha \leq t/4, \qquad x \in I_k$$

If we consider the midline of region $R(j,k,\ell_1,\ell_2)$, $g_{j_k}(x) = l_1\delta_1 + l_2\delta_2(x - x_k)$, we get

$$\begin{aligned} |f_{j_k}(x) - g_{j_k}(x)| &\leq |f(x_k) - l_1\delta_1| + |f'(x_k) - l_2\delta_2||x - x_k| && (2)\\ &\leq \delta_1/2 + \delta_2 w/4 && (3)\\ &\leq t/8 + t/16 && (4) \end{aligned}$$

We see that (2) follows from the fact that the distance between $f_{j_k}(x)$ and $g_{j_k}(x)$ is at most the distance between them at x_k plus any deviation that occurs due to differences in their slopes *(Figure 3)*. Therefore by (4) we have,

$$|f(x) - g_{j_k}(x)| < t/2$$

□

In other words, the distance between f and the midline of the region $R(j,k,\ell_1,\ell_2)$ will be less than half the thickness of that region on the interval I_k. We can therefore conclude that at our optimal level j^*, there exists a strip $R(j,k,\ell_1,\ell_2)$ that covers f over the interval I_k *(Figure 3)*.

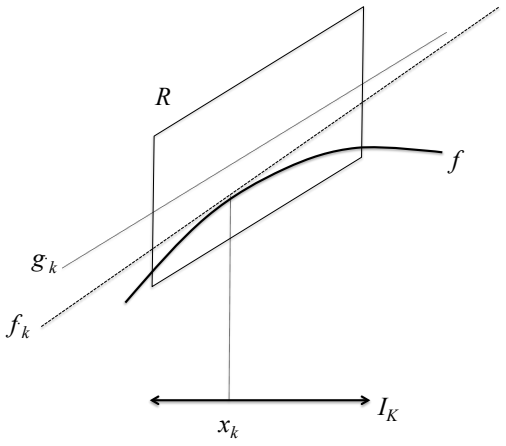


Figure 3: The region associated with f over the interval I_k. As shown in the proof of Lemma 1.2, this region will cover f entirely over the interval I_k

Lemma 1.3. *Let $j = j^*(\alpha,\beta)$ and suppose f is a Hölder(α,β) function on [0,1]. For each $k = 0,...,w^{-1} - 1$ consider the region $R_k \equiv R(j,k,\ell_{1,k},\ell_{2,k})$ associated to f by the manner described in Lemma 1.2. We claim that the R_k are neighbors in $\mathcal{G}(j)$. Thus R_k and R_{k+1} form good continuations and are*

connected by edges in $\mathcal{E}(j)$. We define $\mathcal{T}_j(f) \equiv \{R_k : 0 \leq k < w^{-1}\}$ to be the path in $\mathcal{G}(j)$ of spatially adjacent regions that form good continuations with one another.

Proof. Using the notation from Lemma 1.2, we claim it is enough to show that:

$$|g_{\dot{k}+1}(x_{k+1}) - g_{\dot{k}}(x_{k+1})| \leq t \tag{5}$$

and

$$|g'_{\dot{k}+1}(x_{k+1}) - g'_{\dot{k}}(x_{k+1})| \leq t/w \tag{6}$$

If these conditions hold, our strips satisfy the conditions for *good continuation*. This implies that there is an edge in $\mathcal{E}(j)$ connecting $R(j, k, \ell_1, \ell_2)$ to $R(j, k+1, \ell'_1, \ell'_2)$, where ℓ'_1 and ℓ'_2 are associated to $g_{\dot{k}+1}$. We know the following hold from the Hölder condition or the properties of our regions:

$$\begin{aligned} |g_{\dot{k}+1}(x_{k+1}) - f(x_{k+1})| &\leq \delta_1/2 = t/8, \\ |f(x_{k+1}) - f_{\dot{k}}(x_{k+1})| &\leq \beta w^{\alpha} \leq t/2, \\ |f_{\dot{k}}(x_{k+1}) - g_{\dot{k}}(x_{k+1})| &\leq \delta_1/2 + \delta_2 w/2 = t/4. \end{aligned}$$

If we combine these equations with the triangle equality, (5) follows. Additionally,

$$\begin{aligned} |g'_{\dot{k}+1}(x_{k+1}) - f(x_{k+1})| &\leq \delta_2/2 = t/(8w), \\ |f'(x_{k+1}) - f'(x_{k+1})| &\leq \alpha\beta w^{\alpha-1} \leq t/(2w), \\ |f'_{\dot{k}}(x_{k+1}) - g'_{\dot{k}}(x_{k+1})| &\leq \delta_2/2 = t/(8w). \end{aligned}$$

Combining these yields (6). □

With Lemma 1.3, we can conclude that the optimal covering regions described in Lemma 1.2 can be combined into a sequence of strips, each of which is horizontally adjacent to and shares a similar slope and altitude with the previous region. This guarantees us a set of strips that covers f over the interval [0,1], which in turn enables us to detect the filament if a large enough fraction of the data points lie on f.

1.2 Thresholds

While the covering property given to us by Lemma 1.3 is essential to Arias-Castro *et al.'s* algorithm, it alone does not guarantee us the ability to detect f. Given a set of strips that covers our function, we need to identify each of these regions as significant. In doing so, we end up with a run of significant strips that we can compare to our decision threshold L_n^*. Intuitively, one can see that if the fraction of points on the filament is too small, it is unlikely that the number of points in a given region would exceed N^*. Thus we define the threshold T^* to be the minimum fraction of points on f for which $\mathbf{H}_1$ will be detectable.

1.2.1 Defining Counting Thresholds

To reject our hypothesis of uniformly distributed data, we must first be able to identify individual strips as significant. This relies on a counting threshold N^*. Before defining this threshold, we introduce the notion of a *significant neighbor*. If some region R_i forms a good continuation with a region R_j and $s(R_j) = 1$, where $s(R) = \mathbb{1}_{\{N(R)>N^*\}}$, then we say that R_j is a *significant neighbor* of R_i. Let $p_0 < 1$ be the probability that a given strip has a *significant neighbor* under the null hypothesis, and define $N^+(\epsilon, \lambda)$ such that:

$$P[Poission(\lambda) > N^+(\epsilon, \lambda)] \leq \epsilon$$

By Poisson approximation of the binomial, we have that $\exists n_0$ such that $n \geq n_0$ implies,

$$P[Bin(n, \lambda/n) > N^+(\epsilon, \lambda)] \leq 2\epsilon$$

If we let $N^* = N^+(p_0/162, 2)$, then for $n \geq n_0$,

$$P[s(R) = 1|\mathbf{H}_0] = P[Bin(n, 2/n) > N^*] = p_0/81$$

1.2.2 Defining Length Thresholds

Maintaining the same notation used to set N^*, the length threshold is defined as $L_n^* = 3 \cdot log_{1/p_0}(n)$. This makes L_n^* substantially longer than the longest expected run of significant strips under the null hypothesis, a direct result of the Erdös Rényi Law [2]. In 1200 experiments with data generated under the null hypothesis and $n = 1024$, the longest observed run, L_n exceeded 3 in one case [1].

1.2.3 Defining Detection Thresholds

Set p_1 such that for all $\alpha \in (1, 2]$ and $n_1 > 0$,

$$log_{1/p_1}(n^{1/(1+\alpha)}) \geq 2L_n^*$$

This will ensure long runs under $\mathbf{H}_1$, again a result of the Erdös Rényi Law. Define $\Lambda^+(\epsilon)$ so that

$$P[Poisson(\Lambda^+(\epsilon)) < N^*] \leq \epsilon$$

If we let $\lambda^* = \Lambda^+((1 - p_1)/2)$ and,

$$T^*(\alpha, \beta, S) = 2\lambda^* \beta^{1/(1+\alpha)} \sqrt{1 + S^2}$$

Then we claim that $\mathbf{H}_1$ will be detectable if the fraction of points on the filament, ϵ_n, satisfies

$$\epsilon_n > T^* n^{-\alpha/(1+\alpha)}$$

This depends on the property

$$n \cdot \epsilon_n \cdot w(j^*(\alpha, \beta, n))/\sqrt{1+S^2} \geq \lambda^* \tag{7}$$

Looking at the relationship between $w(j^*)$ and $w(j^+)$ in Lemma 1.1, we see that by the definition of T^*,

$$n \cdot \epsilon_n \cdot w(j^*(\alpha, \beta, n))/\sqrt{1+S^2} \geq n^{1/1(1+\alpha)} \cdot T^* w(j^+(\alpha, \beta, n))/(2\sqrt{1+S^2}) = \lambda^*.$$

1.3 Behavior of the Algorithm

Consider a filament $f \in$Hölder(α, β) that contains a fraction of the data ϵ_n. Let $j = j^*$ and consider the sequence of adjacent strips $\mathcal{T}_j \equiv \{R_k : 0 \leq k < w^{-1}\}$. For each region in $\mathcal{T}_j$

$$N(R) \sim Bin(n, (1-\epsilon_n) area(R) + \epsilon_n \gamma(f, R))$$

where $area(R) = 2/n$ and γ denotes the arc length of f in the region R.

By Poisson approximation of the binomial,

$$N(R) \sim Poisson(\mu)$$

where by (7),

$$\mu \geq 1 + n \cdot \epsilon_n \cdot w/\sqrt{1+S^2} \geq \lambda^*$$

Thus for each $R \in \mathcal{T}_j$ and sufficiently large n,

$$P[N(R) > N^*] \geq p_1$$

Denote the regions in $\mathcal{T}_j$ to be $R_0, ..., R_{w^{-1}-1}$. We would like to find the expected length of the longest observed run L_n that satisfies,

$$N(R_0) > N^*, ..., N(R_{w^{-1}-1}) > N^*$$

and claim that

$$P[L_n > L_n^*] \to 1, \qquad n \to \infty. \tag{8}$$

Proof. Let $Z_i = \mathbb{1}_{\{N(R_k) > N^*\}}$, and note that $Z_i \sim Bernoulli(p_i)$ where $p_i \geq p_1$. If $m = w^{-1} \geq an^{1/(1+\alpha)}$ where a is some constant and $p = p_1$,

$$L_n > log_{1/p_1}(n^{1/(1+\alpha)})(1 + o_p(1))$$

follows from the Erdös Rényi Law.
As we have chosen p_1 such that

$$log_{1/p_1}(n^{1/(1+\alpha)}) \geq 2L_n^*,$$

(8) follows. □

If there is not a filament present in the data, we would like to show that the probability of observing a run of significant strips greater than L_n^* tends to 0 for increasing n.

Proof. Consider the probability that a run of length L begins at some region R. Based on our choice of N^*, we have that,

$$P[s(R) = 1|\mathbf{H}_0] \leq p_0/81$$

We note that each region in $\mathcal{G}(j)$ has 81 neighbors so that,

$$P[s(R') = 1 \text{ for at least one neighbor of } R|\mathbf{H}_0] \leq p_0$$

follows from Boole's inequality.

In order to observe a path of length L, we must have some starting point R that satisfies $s(R) = 1$. This region must be neighbors with some R' such that $s(R') = 1$ and so on. Thus, the probability of observing a run of length L will be bounded by p_0^L.

As there are at most $M_j = w^{-1}\delta_1^{-1}\delta_2^{-1}2S$ starting points for a run in $\mathcal{G}(j)$,

$$P[\text{there is at least one significant path of length L in } \mathcal{G}(j)|\mathbf{H}_0] \leq M_j p_0^L$$

follows from Boole's inequality. Taking log_2,

$$\begin{aligned} log_2(M_j) + log_2(p_0)L &= log_2(w^{-1}\delta_1^{-1}\delta_2^{-1}2S) + log_2(p_0)L \\ &= 2J - 2j + 3 + log_2(S) + log_2(p_0)L \\ &\leq 2J + C + log_2(p_0)L \end{aligned}$$

Letting $L = L_n^*$, the final expression tends to $-\infty$ for increasing n. From whence it follows

$$P[\text{there is a significant run of length } L_n^*] \to 0, \qquad n \to \infty.$$

□

2 Estimation Methods

Detecting a filament in some d-dimensional space gives rise to the question of how to estimate that filament. While techniques for doing so exist in literature, the assumptions that they make either limit the types of filaments that can be estimated [4], or produce less than satisfactory estimates [5].

2.1 Medial Axis of Support

The first of the methods we consider relies on estimating the support of the filament, denoted ξ, and using it as a basis for the estimate of f [4]. Genovese *et al.* show that under certain conditions, the best estimate for f is the medial axis of ξ. Before summarizing their filament estimation technique, it is useful to provide definitions for two key elements used in the process, Hausdorff distance and the smoothness of a filament.

Definition. *For any sets A,B we define the Hausdorff distance between these sets to be:*

$$d_H(A,B) = min\{\delta | A \subset B \oplus \delta and B \subset A \oplus \delta\}$$

where $A' \oplus \delta = \bigcup_{x \in A'} B(x,\delta)$*, and* $B(x,\delta)$ *is the closed ball centered at* x *with radius* δ*.*

Definition. *for any* x, y, z *on the filament* f*, let* $r(x,y,z)$ *be the radius of the circle tangent to* x, y, z*. We define the smoothness of* f*, to be:*

$$\Delta f = min_{x,y,z} r(x,y,z)$$

This can be interpreted as the largest possible radius for a ball that can roll freely around the filament.

With these definitions, we can know look at the method proposed by Genovese *et al.* for estimating f. A filament is represented as

$$Y_i = f(U_i) + \epsilon_i$$

where $f : [0,1] \to \mathbb{R}^d$ for $d > 1$, U_i come from some distribution H on $[0,1]$, and ϵ_i come from some mean 0 distribution F with noise level σ. The goal is to estimate the support ofof the marginal density of Y_i, $\xi = \bigcup_{0 \le u \le 1} B(f(u), \sigma)$.

Genovese *et al.* show that any estimate for the boundary of ξ can be converted into a set that is close in Hausdorff distance to the true filament. If the rate of convergence for the boundary estimator to the true boundary is r_n, then the rate of convergence for the filament estimator to the true filament will also be r_n [4]. In order to show this, Genovese *et al.* rely on several assumptions about the distributions H and F as well as the smoothness of the filament. Of these assumptions, the following about the error structure provide restrictions as to the types of filaments that we would like to estimate.

- F has support $B(0,\sigma)$ and a bounded, continuous density ϕ with respect to Lebesgue measure on $\mathbb{R}^2$ such that $\phi(y) > 0 \forall y \in B(0,\sigma)$.
- ϕ is non increasing
- ϕ is symmetric

- f is sufficiently smooth, in other words $\sigma < \Delta f$. If $f(0) \neq f(1)$, then $||f(1) - f(0)||/2 > \Delta f$

The second and third assumptions provide only minor limitations to the types of filaments that we might be interested in estimating. However, the first requires some bound on the distribution for σ. This is a problem since many common distributions, such as the normal distribution, do not satisfy this property. Thus we would be unable to apply this technique where $\epsilon_i \sim N(0, \sigma)$, a property that we might expect to be common for many filaments.

2.2 Path Density Gradient Field

Another method for estimating filaments assumes that the data $X_1, ..., X_n$ come from some distribution μ_X with density g_X. The idea behind this technique is to create a vector field from the gradient of the density. Genovese *et al.* show that under certain conditions, the flow of this field starting at a random point, in the direction of steepest ascent, leads to the filament, where the path density is large.

2.2.1 Flows

A key component to this method is the idea of flow. For some vector field V, let $\psi(t, x)$ denote the point obtained by starting at the point x and following the flow of V for time t. Consider any neighborhood U of $x \in \mathbb{R}^2$. For such a neighborhood, there exists a $U_1 \subset U$, an interval $I \subset \mathbb{R}$ that contains 0, and a smooth mapping $\psi : I \times U_1 \to U$ such that the following hold.

- $\psi(0, x) = x$
- $\frac{\partial}{\partial t}\psi(t, x) = V(\psi(t, x))$
- if $s, t, s + t \in I$, $\psi(s + t, x) = \psi(s, \psi(t, x))$

We note that the paths $t \mapsto \psi(t, x)$, known as integral curves or local flows, are unique in that they will be equal when their domains overlap. These local flows can be extended to global flows when the mapping $\psi : \mathbb{R} \times \mathbb{R}^2 \to \mathbb{R}^2$ satisfies the above conditions and $I = \mathbb{R}$. These global flows exist whenever V has a compact support [5].

2.2.2 Estimating Filaments with Path Densities

Let U_0 denote the compact support of g_X, and note that the vector field $V = \nabla g_X$ will have a unique global flow $\psi(t, x)$. For any $x, y \in U_0$, we have the equivalence relationship $x \sim y$ if $\psi(t, y) = x$ for some $t \in \mathbb{R}$.

Definition. *We say that y precedes x, $y \preceq x$, if $x \sim y$ and $t \geq 0$. The reverse evolution of $A \subset U_0$ under the flow ψ is defined as:*

$$V(A) = \{y_0 : y \preceq x, x \in A\}$$

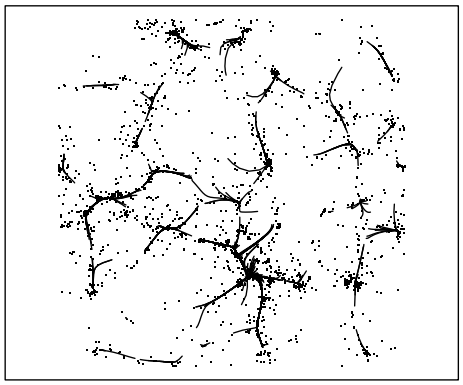
Definition. *define the path measure π to be the probability that the flow, starting from a random point, hits a set A as:*

$$\pi(A) = \mu_X(V(A))$$

Definition. *The path density of g_X, $p : \mathbb{R}^2 \to [0, \infty]$ gives the π probability associated with infinitesimal neighborhoods of x, and is defined as,*

$$p(x) = lim_{r \to 0} \frac{\pi(B(x, r))}{r}$$

The notion of *path density*, is used to capture filaments by finding sets for which p exceeds some threshold λ. As g_X is unknown, Genovese *et al.* use a kernel estimator to obtain $\hat{g}_n(x)$. Thus the path density used to find filaments is $\hat{p}_n(x)$, a weighted average of how many observed paths get close to x. Estimating a function using this process produces whisker-like structures close to the filament where $p > \lambda$ *(Figure 4)*. However, we expect f to be a smooth function, so this is a satisfactory result.



C

Figure 4: Estimate for a filament produced by Genovese *et al.* using the path density

3 Estimating Filaments

Using the method described in Section 1, we can detect a filament f by constructing a run of strips that contains f. This idea provides motivation for a new filament estimation technique. Using a modified greedy algorithm described in section 3.1.1, we generate a set of strips denoted W_1 that covers f. To ensure that the entire filament is contained in this set, each strip's thickness must be relatively large. By taking the midline of the covering W_1, we obtain a piecewise linear function that acts as a rough estimate for f, which we denote $\hat{f}_1$. While this provides a good starting point for an estimate, f is most likely a smooth function, so a piecewise linear estimate is not ideal. To address this issue, we apply Chandler *et al.'s* iterative process for automatic locally adaptive smoothing.

Using a kernel smoother on $\hat{f}_1$, we produce the smooth function $\tilde{f}_1$. We then map $\tilde{f}_1$ to a space where it is easier to estimate and iterate through the process until $||\tilde{f}_n - 1/2|| < \eta$. The problem of finding the proper value of η is left to further research, but we note that this value must be small enough so that we obtain an accurate estimate of f, but large enough that $||\tilde{f}_n - 1/2|| < \eta$ will be satisfied within a reasonable number of iterations. We also note that we consider the distance between $\tilde{f}_n$ and $1/2$ because our mapping takes points on $\tilde{f}_n$ to the line $y = 1/2$. Additionally, we could consider stopping the algorithm when $||\tilde{f}_i - \tilde{f}_{i+1}|| < \delta$, however finding the appropriate value of δ would present similar challenges.

Our proposed method addresses some of the problems seen in other filament estimation techniques. For example, we do not make assumptions about the noise level of the filament being bounded. Additionally, our estimates are smooth functions that do not have the whisker-like structures that are seen when using path densities to estimate filaments. Apart from addressing some of the shortcomings of other estimation techniques, our method does not rely on properly selecting tuning parameters, a process that can often be difficult [3].

3.1 Adapting Filament Detection

The ability to detect filaments using a greedy algorithm depends on correctly identifying the starting point and orientation of the filament. While this is not a trivial problem, methods such as principal component analysis could be used to do this. By constructing a neighborhood around each data point and computing principal components, we could determine if a filament started at that point, and if so, the orientation of the filament. In our simulations, we assume that the starting point and initial slope of the filament are known, and show that we can estimate a filament with this information. We leave the problem of determining the starting point to future work.

3.1.1 Covering the Filament

Based on a known starting point for a filament, we define the first strip, $R_1(l_1, \theta_1, t_1)$, in terms of its location, slope, and thickness, where location refers to the coordinates of the bottom left corner of the strip and slope is defined as $tan(\theta)$. If we let $r_{i-1} = (x_{i-1}, y_{i-1})$ denote the bottom right corner of $R_{i-1}(l_{i-1}, \theta_{i-1}, t_{i-1})$, we can define successive strips as $R_i(l_i, \theta_i, t_i)$, where $l_i = (x_{i-1}, y_{i-1} + 0.5(t_{i-1} - t_i))$ and θ_i and t_i are the values corresponding to the optimal slope and thickness for the i^{th} strip. We determine the optimal slope and thickness for R_i with a greedy algorithm, modified to ensure robustness to randomly occurring pockets of noise, that runs as follows.

- Let $T = [t_{min}, t_{max}]$ denote the range of strip thicknesses that we are willing to consider for the first covering, W_1. To ensure that W_1 contains the entire filament, we begin with large values of t_{min} and t_{max}. These values decrease in each successive iteration to allow us to narrow in on f. While we have not found the optimal rate for which to decrease these values, t_{min} should fall at a faster rate than t_{max}. This enables us to narrow in on the filament while still allowing us to detect filaments when the mapping has large changes in slope in successive iterations. Smaller increments for this range make the algorithm more computationally intensive, but also give us better results. We found that using an increment of 0.01 works for many filaments.

- Let $\Theta_i = [\theta_{i-1} + \theta_{min}, \theta_{i-1} + \theta_{max}]$ correspond to the range of slopes we are willing to consider for R_i, where slope is given by $tan(\theta)$ for $\theta \in \Theta_i$. With this definition, θ_{min} and θ_{max} represent the maximum allowable difference in slope between R_{i-1} and R_i. As the original filament is unknown, we start with larger values for θ_{min} and θ_{max}, allowing us to capture filaments that have large second derivatives. We decrease these values in successive iterations since the filament will be mapped to the line $y = 1/2$, suggesting that the derivative of the filament in the new space should converge to 0. Again, smaller increments in this range add computational intensity but result in improved performance. We found that using an increment of $\pi/16$ works for many filaments.

- With T and Θ_i, we consider $R_{i,t}(\theta)$, strips with a fixed thickness of $t \in T$ as functions of $\theta \in \Theta_i$.

- For each $\theta \in \Theta_i$, sum the number of points from our dataset contained in $R_{i,t}(\theta)$ and denote this value $n_{i,t}(\theta)$. We define $N_t(\theta) = \{n_{i,t}(\theta) | \theta \in \Theta_i\}$

- Let $\Theta^*_{i,t} = \{\theta \in \Theta_i | n_{i,t}(\theta) \geq 90^{th} quantile(N_t(\theta))\}$, then the optimal value of theta for $R_{i,t}(\theta)$ will be $\theta^*_{i,t} = median(\Theta^*_{i,t})$, with $n^*_{i,t}$ corresponding to the number of data points in $R_{i,t}(\theta^*_{i,t})$. The $\Theta^*_{i,t}$ subset is chosen to help ensure that our selection of $\theta^*_{i,t}$ is robust to any randomly occurring pockets of concentrated noise.

- Repeating this process at each level of thickness yields $\Theta_i^* = \{\theta_{i,t}^* | t \in T\}$ and $N_i^* = \{n_{i,t}^* | t \in T\}$.
- The optimal values for $R_i(l_i, \theta_i, t_i)$ are then defined as $\theta_i = \{\theta_{i,t}^* \in \Theta_i^* | n_{i,t}^* = max\{N_i^*\}\}$ and $t_i = \{t | \theta_i^* = \theta_{i,t}^*\}$.

3.1.2 Mapping the Filament

By running through the process described in Section 3.1.1 once, we generate W_1. We can obtain a piecewise linear estimate for a filament f by taking the midline of these strips, which we denote $\hat{f}_1$. However, many filaments that we would like to estimate for practical purposes are smooth. Additionally, using large values of t_{min} and t_{max} in the first iteration to ensure that W_1 does not lose the filament causes $\hat{f}_1$ to be a rough estimate and unsatisfactory as a final result. To solve this problem, we adapt Chandler *et al.'s* process for automatic locally adaptive smoothing of level sets to our filament problem. This involves smoothing $\hat{f}_1$ to obtain $\tilde{f}_1$ and mapping $\tilde{f}_1$ to a space where it can be better approximated. Iterating through this process results in a smooth and more accurate estimate of f in the original space.

For the i^{th} iteration, obtain a covering W_i in the manner described by section 3.1.1. Take the piecewise linear estimate $\hat{f}_i$ obtained from the midline of W_i, and parameterize it as a path $\hat{f}(t) = (\hat{f}_{i,1}(t), \hat{f}_{i,2}(t))$ for $t \in [0,1]$, where $\hat{f}_i(0)$ represents the starting point of our estimate. We smooth this path using a box kernel with bandwidth b to obtain:

$$\tilde{f}_i(t) = \begin{cases} 1/(2b) \int_t^{t+b} \hat{f}_i(t)dt, & 0 \leq t < b \\ 1/(2b) \int_{t-b}^{t+b} \hat{f}_i(t)dt, & b \leq t \leq 1-b \\ 1/(2b) \int_{t-b}^{t} \hat{f}_i(t)dt, & 1-b < t \leq 1 \end{cases}$$

By choosing b to be relatively small, we benefit from smoother estimates as we iterate through this procedure, while avoiding having to select a global bandwidth [3].

Our transformation depends on a map $\ell_{\tilde{f}_i,w}$ which takes the estimate of our filament to a space where it is easier to estimate. We let F_i denote the estimate of our filament so that $\tilde{\mathbf{f}}_\mathbf{i} : [0,1] \to F_i$. In each iteration, we subset our data so as to only consider points within a distance w of F_i. This tuning parameter allows us to "zoom in" on the filament to obtain a more accurate estimate. While this is a desirable result, setting w to be too small amplifies F_i's error in the transformed space, making our mapped filament harder to estimate. To adjust for this, we define w as an increasing function t_{max} so that we subset the data less with each iteration.

Before defining $\ell_{\tilde{f}_i,w}$, we introduce the following notation. Let $P_{\tilde{f}_i}(x)$ be the orthogonal projection, in the functional sense, of x onto F_i. Then we have that for each $x \in \mathbb{R}^2$, $\exists t_x \in [0,1]$ such that $P_{\tilde{f}_i}(x) = \tilde{f}_i(t_x)$ and $\forall t \in$

$[0,1]$, $||\tilde{f}_i(t_x) - x|| \leq ||\tilde{f}_i(t) - x||$. The domain of our map is defined to be $C(\tilde{f}_i, w) = \{x \in \mathbb{R}^2 : ||P_{\tilde{f}_i}(x) - x|| \leq w/2\}$. With this notation, we can define $\ell_{\tilde{f}_i,w} : C(\tilde{f}_i, w) \to [0,1]$ as,

$$\ell_{\tilde{f}_i,w}(x) = (\tilde{\mathbf{f}_\mathbf{i}}^{-1}(P_{\tilde{f}_i}(x)), 1/2 + ||P_{\tilde{f}_i}(x) - x||/w(-1)^{(1-q)})$$

where $q = \mathbb{1}_{x_2 > p_2}$ for $x = (x_1, x_2)$ and $P_{\tilde{f}_i}(x) = (p_1, p_2)$. In other words if some point lies on F_i, it will be mapped to the line $y = 1/2$, while points above and below F_i will be mapped above and below the line $y = 1/2$ respectively.

4 Results

We examine the filament estimation method discussed in the previous section, applying it to data simulated using R software. In each of these simulations, we consider how our algorithm handles a filament represented as $f(x) + \epsilon$ where $f : [0,1] \to A \subset \mathbb{R}$ and ϵ is a noise term with unbounded support. While we only consider filaments in $\mathbb{R}^2$, the algorithm should extend to higher dimensions.

It is possible to define an inverse of the mapping procedure used in our algorithm and evaluate the accuracy based on the distance between the true filament and the estimated filament mapped back to the original space. For simplicity, our algorithm labels points on the estimated filament and we evaluate accuracy by comparing these data points with those that are actually on the filament.

It is important to note that in each iteration, we decrease the thickness of the regions in an attempt to estimate the filament more accurately. Because the support of our noise added to the filament is unbounded, we expect a certain percentage of our filament points to fall outside the strips at each level of thickness. While this is not a problem for relatively thick strips, the percentage increases as thickness decreases. Thus, if we iterate through the algorithm too many times, we expect a large percentage of the filament points to fall outside the strips. As a result, many points that should be labeled as on the filament are labeled as noise *(Table 1)*. This concern must be balanced with the fact that at each thickness level, we expect a certain number of background noise points to fall within the strips. This percentage corresponds to the area of the strips and therefore decreases as area decreases. Thus, if we do not iterate through the algorithm enough, many of the noise points will be labeled as part of the filament *(Table 1)*.

For many purposes though, we are concerned with finding f as opposed to labeling points as either on the filament or noise. With symmetric error centered at 0, the strips should converge to f as they converge towards the concentration in data points. This means that even though the percentage of filament points the strips capture decreases as their thickness decreases, they will still produce a reasonable estimate of f.

We first consider the simple filament $f(x) = x^2 + \epsilon$ where $\epsilon \sim N(0, 0.05^2)$. In this simulation, 1000 data points are distributed uniformly on the filament while

2500 data points are distributed uniformly on $[0,1] \times [-1,1]$. *Figure 5* shows the results of six iterations of the algorithm. Examining the fourth through sixth iterations, we can see one of the main issues that remains to be resolved with the algorithm, identifying the correct stopping point when noise is added to the filament. There will undoubtably be some error in any estimate for f in any iteration. At some point, the error that running another iteration corrects for is outweighed by the error that it picks up by trying to estimate again. This occurs between the fifth and sixth iterations of the simulation when the algorithm starts producing similar estimates but with different minor errors *(Figure 5)*. We chose to stop iterating when the estimated filament in the mapped space $\tilde{f}_i$ satisfied

$$||\tilde{f}_i - \tilde{f}_{i+1}|| < \delta.$$

Our cutoff was selected somewhat arbitrarily, and there is likely a better value that would provide an optimal estimate of f.

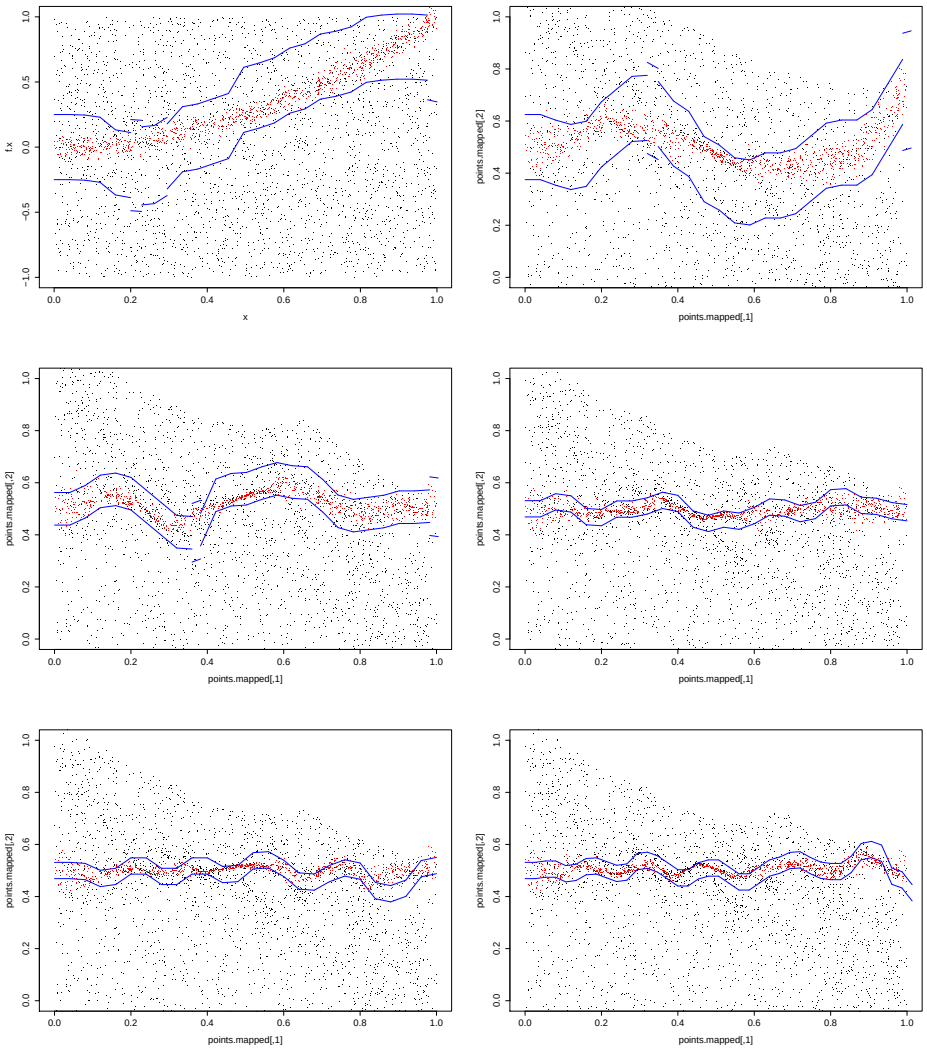


Figure 5: Six iterations of the algorithm applied to $f(x) = x^2 + \epsilon$, where $\epsilon \sim N(0, 0.05^2)$. The top left image shows the filament in the original space with points on the filament represented in red. Each subsequent picture shows the data mapped based on the procedure described in Section 3.1.1. Blue boxes represent the covering selected by the greedy algorithm described in Section 3.1.2.

In the previous simulation, we noted that it was desirable to have mean 0, symmetric noise added to our filament. The next simulation examines how our algorithm handles noise that does not satisfy these properties. Again, we use a filament of the form $f(x) = x^2 + \epsilon$, but here $\epsilon \sim exp(20)$. For this simulation,

Table 1: Classification Results for $f(x) = x^2 + \epsilon \sim N(0, 0.05^2)$

Iteration	False Positives (rate)	False Negatives (rate)
1	641(.26)	4 (.004)
2	554 (.22)	30 (.03)
3	405 (.16)	44 (.044)
4	295 (.12)	229 (.229)
5	283 (.11)	260 (.26)
6	354 (.14)	177 (.177)

we decrease the minimum thickness and area between the first four iterations at the same rates as the previous simulation. Although the algorithm loses the filament in the fourth iteration, we see that it is capable of recapturing it in subsequent iterations *(Figure 6)*.

In order to recapture the filament, we decreased the rate that the minimum thickness fell between iterations while keeping the rate at which the area fell constant. This suggests that dropping thickness at an decreasing rate between iterations while dropping area at a constant rate may help ensure that we do not lose the filament. We could also consider generating several coverings at each iteration, summing the points in each covering, and selecting the one with the greatest density of points. This would enable us to find the optimal thickness and area for the strips at each iteration. We leave these ideas to further study.

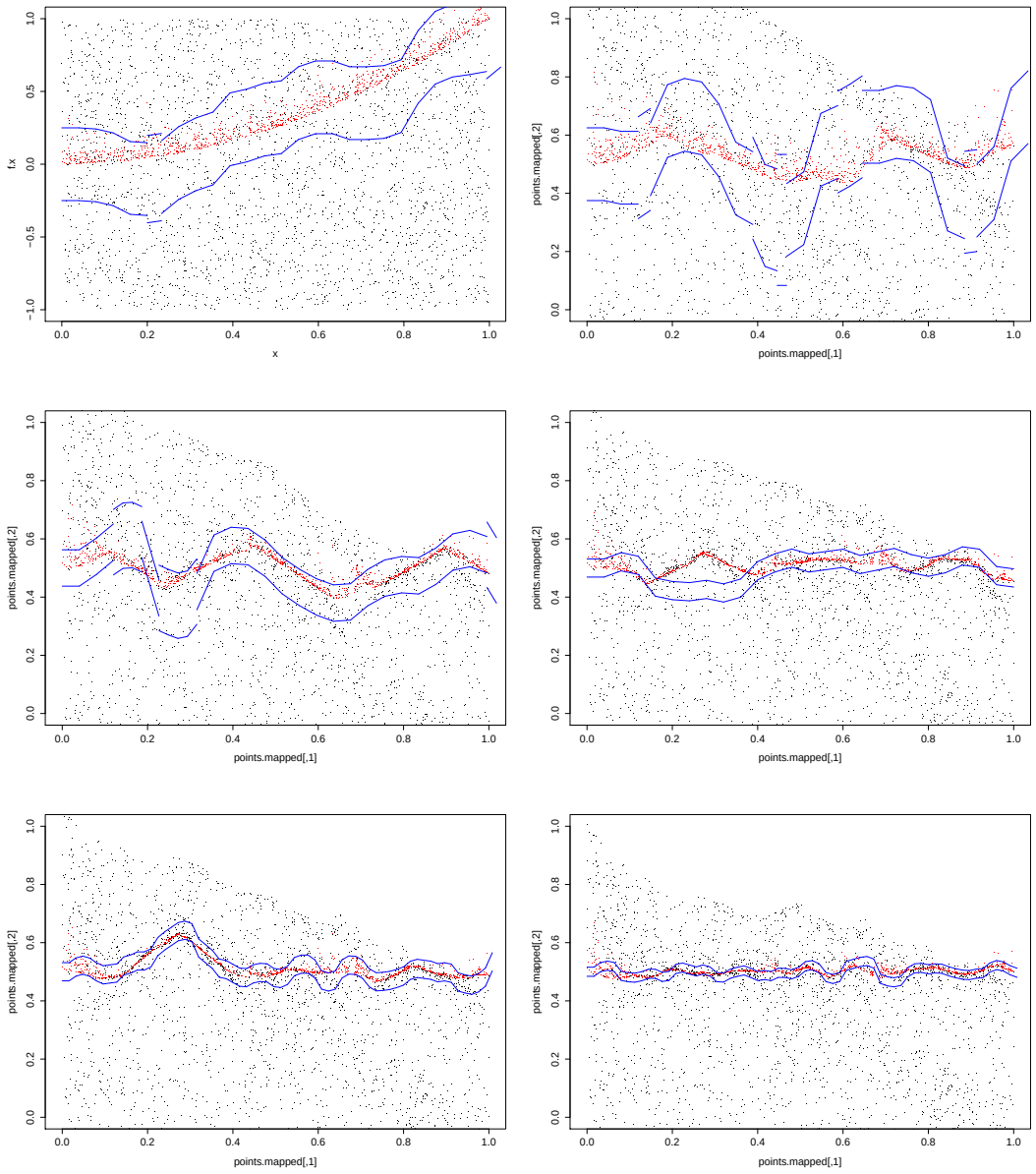


Figure 6: Six iterations of the algorithm applied to $f(x) = x^2 + \epsilon$ where $\epsilon \sim exp(20)$. The top left image shows the filament in the original space with points on the filament represented in red. Each subsequent picture shows the data mapped based on the procedure described in Section 3.1.1. Blue boxes represent the covering selected by the greedy algorithm described in Section 3.1.2.

Table 2: Classification Results for $f(x) = x^2 + \epsilon \sim exp(20)$

Iteration	False Positives (rate)	False Negatives (rate)
1	627(.25)	34 (.034)
2	613 (.25)	116 (.116)
3	594 (.24)	83(.083)
4	369 (.15)	295 (.295)
5	583 (.23)	80 (.08)
6	438 (.18)	241 (.241)

Next, we consider a more complex filament of the form $f(x) = sin(5x) + \epsilon$, where $\epsilon \sim N(0, 0.05^2)$. In this simulation, 1000 data points are distributed uniformly on the filament while 2500 data points are distributed uniformly on $[0, 1] \times [-1.5, 1.5]$. In *Figure 7*, we see that our algorithm has some trouble handling the curvature of the function. This is especially apparent in the original space where f changes from decreasing to increasing. Despite this, the algorithm does a reasonable job of mapping points on the filament to the line $y = 1/2$.

Although the algorithm works for this particular filament, the results suggest that it will not handle filaments with large second derivatives well due to the difficulty in generating an accurate covering. This shortcoming presents a major problem because large errors can create peaks in the mappings. These peaks are again difficult to cover, and increase the chance that our covering loses the filament. We can observe one of these peaks in the upper right image of *Figure 7*, but it is relatively small and is corrected for in future iterations.

The fact that our algorithm will only work for a limited class of functions is not surprising as Arias-Castro *et al.* showed their covering was only guaranteed for certain Hölder functions. Again, we may be able to correct for some of this problem by generating multiple coverings and selecting the one with the greatest density of points. However, there will be certain filaments that we cannot capture with our covering method.

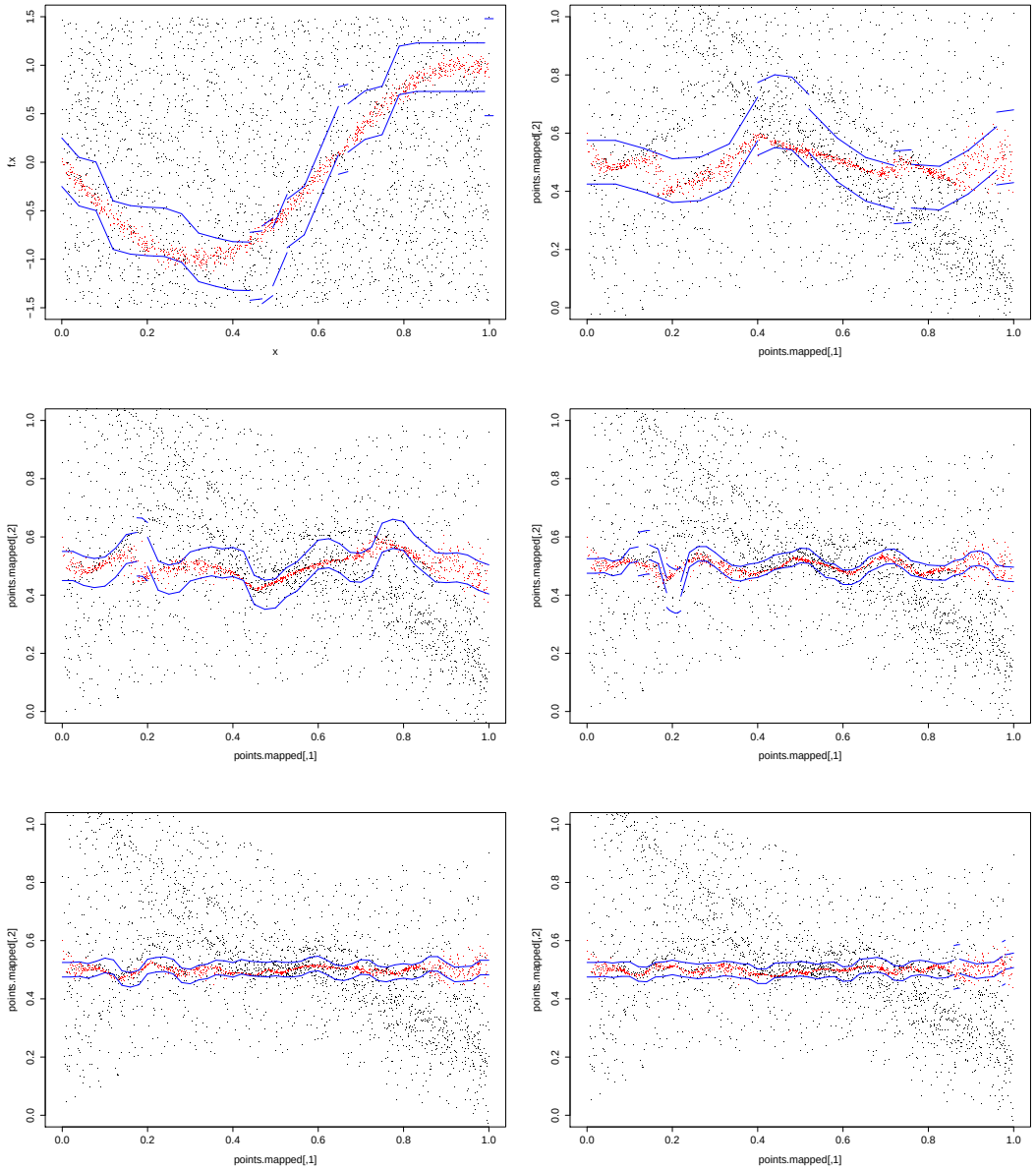


Figure 7: Six iterations of the algorithm applied to $f(x) = sin(5x) + \epsilon$ where $\epsilon \sim N(0, .05^2)$. The top left image shows the filament in the original space with points on the filament represented in red. Each subsequent picture shows the data mapped based on the procedure described in Section 3.1.1. Blue boxes represent the covering selected by the greedy algorithm described in Section 3.1.2.

Table 3: Classification Results for $f(x) = sin(5x) + \epsilon \sim N(0, .05^2)$

Iteration	False Positives (rate)	False Negatives (rate)
1	472 (.19)	46 (.046)
2	605 (.24)	34 (.034)
3	487 (.19)	92 (.092)
4	447 (.18)	142 (.142)
5	508 (.2)	101 (.101)
6	531 (.21)	57 (.057)

5 Conclusion

The results from these simulations suggest that our algorithm has the potential to estimate a wide variety of filaments. However, there are still minor concerns that must be addressed. One of the these is that if the covering loses the filament in any iteration, it may never be recaptured. We may be able to limit this by using a greedy algorithm to select the optimal thickness and area of the covering, based on the density of points in that covering. This would ensure the covering contains f if possible, given the constraints θ_{min} and θ_{max}. However, limiting the θ values we are willing to consider makes it impossible to capture filaments with relatively large second derivatives. That is not to say that we should consider any values for θ, as we still have assumptions about the smoothness of f. We merely note that there is a limit to the curvature of filaments we can estimate.

Selecting thickness and area with a greedy algorithm addresses our second concern as well. By choosing the optimal thickness and area for each covering, it eliminates the need to find a rate for decreasing these values between iterations. Finding this rate would prove to be a challenging problem as it appears to depend on the curvature of the filament. In addition, we could not adjust the rate based on curvature as filaments are unknown in practice. This would again limit the curvature of the filaments that our algorithm can estimate.

Finally, we have yet to find the optimal point at which to stop iterating. In the simulations, we saw the algorithm reaches a point where the differences between estimates in each iteration are minor. This is an appropriate time to stop, but we do not have a sense of how close subsequent iterations should be before stopping to produce the best estimate for f. We could easily examine this problem by comparing the distance between subsequent iterations in the mapped space with the distance between them in the original space.

References

[1] Ery Arias-Castro, David L. Donoho, and Xiaoming Huo. Adaptive multiscale detection of filamentary structures in a background of uniform random points. *Annals of Statistics*, 34:326–349, 2006.

[2] R. Arratia and M. S. Waterman. The erdos-renyi strong law for pattern matching with a given proportion of mismatches. *Annals of Probability*, 17:1152–1169, 1989.

[3] Gabriel Chandler and Leif T. Johnson. Automatic locally adaptive smoothing for tree-based set estimation. *Journal of Statistical Computation and Simulation*, 1:29652979, 2011.

[4] Christopher R. Genovese, Marco Perone-Pacifico, Isabella Verdinelli, and Larry Wasserman. The geometry of nonparametric filament estimation. *Journal of the American Statistical Association*, 107:1, 2012.

[5] Christopher R. Genovese, Marco Perone-Pacifico, Isabella Verdinelli, and Larry Wasserman. On the path density of a gradient field. *Annals of Statistics*, 40:1, 2012.

[6] Radu S. Stoica, Vicent J. Martinez, and Enn Saar. A three dimensional object point process for detection of cosmic filaments. *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, 56:459–477, 2007.