Summer 2017

A Boundary-Based Measure for Gerrymandering

Carson Sprock
San Jose State University

A BOUNDARY-BASED MEASURE FOR GERRYMANDERING

A Thesis

Presented to

The Faculty of the Department of Mathematics

San José State University

In Partial Fulfillment

of the Requirements for the Degree

Master of Science

by

Carson Sprock

August 2017

The Designated Thesis Committee Approves the Thesis Titled

A BOUNDARY-BASED MEASURE FOR GERRYMANDERING

by

Carson Sprock

APPROVED FOR THE DEPARTMENT OF MATHEMATICS

SAN JOSÉ STATE UNIVERSITY

August 2017

Dr. Elizabeth Gross	Department of Mathematics
Dr. Wasin So	Department of Mathematics
Dr. Edward Schmeichel	Department of Mathematics

ABSTRACT

A BOUNDARY-BASED MEASURE FOR GERRYMANDERING

by Carson Sprock

This paper presents a new measure for quantifying legislative gerrymandering based intuitive observation that in a non-gerrymandered district a randomly placed observer should be able to walk in a straight line and only cross the boundary of the district once, when the district is exited. We make this notion precise in terms of the expected value of such crossings. The result is the Boundary Intersection Number, or BIN. Properties of the BIN score are proven and its computational properties discussed.

ACKNOWLEDGEMENTS

I would like thank Dr. Gross for her patience and encouragement during the writing of this thesis.

I would also like to thank Dr. Schmeichel for our discussions about gerrymandering, the results of which became the main idea presented in this paper.

TABLE OF CONTENTS

LIST OF FIGURES

Figure

CHAPTER 1

INTRODUCTION

The term "gerrymander" originated in 1812 during a feud between the Republican governor of Massachusetts Elbridge Gerry and the rival Federalist Party. At the time the Federalists were the minority party and Gerry was trying to prosecute members of the Federalist press when he signed a controversial districting plan that favored the Republicans. The Federalists responded by ridiculing Gerry and the districting plan in the press with a cartoon comparing one of the districts to a salamander, dubbing it a "gerrymander" [5]. The term has stayed in the American political lexicon and refers broadly to the "manipulation of district lines for partisan purposes" [11]. The eponymous gerrymander is shown in Figure 1.

Figure 1.1: The original gerrymander

The problem of gerrymandering originates with the power of a legislature to create its own districts and districting plans. This can allow self-interested partisans to essentially choose their own constituencies and given insufficient restrictions, a party with a minority in the population can achieve a majority in the legislature through the creation of cleverly constructed districting plans.

To illustrate how this can be achieved, consider a two-party election contest. To win, a party needs only 51% of the votes. Any more beyond 51% can be considered "wasted." An effective partisan gerrymander attempts to "pack" opposition voters into a few districts with large majorities and spread their own voters over many districts with smaller majorities ("cracking"). The goal is to engineer artificial landslides for the opposition in a few districts while winning smaller yet secure majorities for their own party in many districts [11].

Figure 1.2 illustrates how gerrymandering works. In this example, 3/5 of voters belong to the Blue party and 2/5 the Red party. The left-most picture shows five districts drawn in such a way that each party is represented proportionately, giving Blue a majority. The right-most picture shows how it is possible to engineer a Red majority through gerrymandering. Since the partisan distribution of voters over the geographic area to be gerrymandered is usually conducive to gerrymandering, gerrymandered districts can look highly irregular as their boundaries are drawn deliberately to incorporate and avoid certain areas of the map.

In addition to gerrymandering, other legal and constitutional requirements often result in misshapen districts as legislatures attempt to meet them. Sometimes districts are required to be geographically contiguous (i.e. topologically connected) and to have equal populations [11]. This requirement can result in misshapen figures if the district border follows the borders of other administrative units such as counties, townships, or the state boundary. The Voting Rights Act of 1965 also

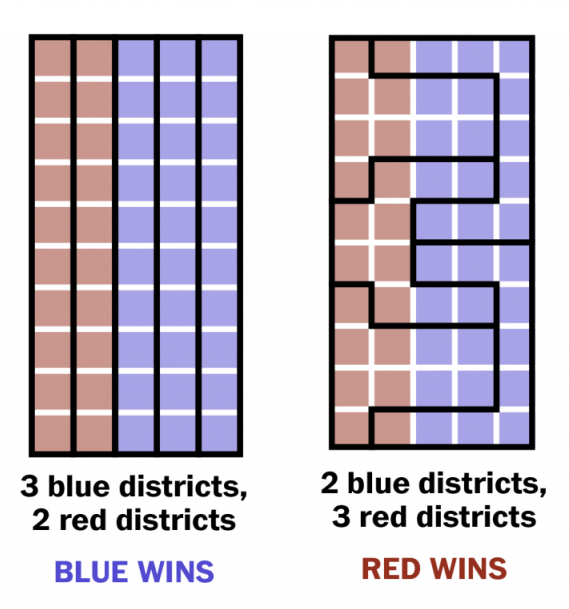

Figure 1.2: A example of gerrymandering (adapted from [7])

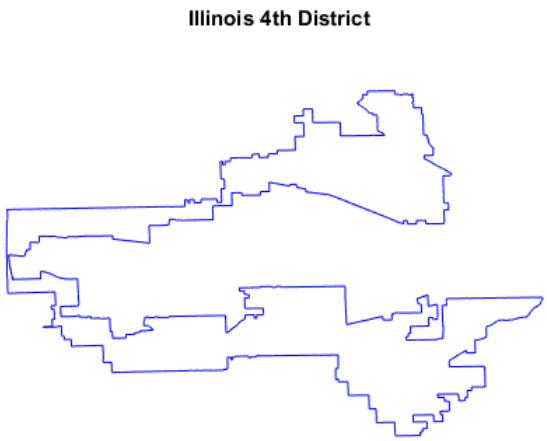

Figure 1.3: Illinois 4th Congressional District, an example of "packing".

forbids the dilution of minority districts, which can lead to the creation of oddly shaped districts such as Illinois' 4th Congressional District in Figure 3 [11, 6]. This district provides a good example of packing because it was created to connect to geographically separated groups of voters. The contiguity requirement is satisfied by long tails connected by a thin stretch of highway.

CHAPTER 2

REVIEW OF GERRYMANDERING MEASURES

Beyond "I know it when I see it," there is no agreed upon standard for determining if a district or districting plan is gerrymandered and multiple measures have been proposed for quantifying gerrymandering [15, 11, 9]. Most measures of gerrymandering are attempts to quantify shape. As "shape" is a rather ambiguous concept, it is unsurprising there are many competing metrics using different properties of the geometric figures used to represent districts. There is also disagreement among scholars over which aspects of shape are the most important for gerrymandering and some authors do not correctly specify which aspect of shape is being measured [9].

Taylor [14] identifies four aspects of shape: elongation, indentation, separation and puncturedness. While he fails to make these four concepts precise, elongation roughly refers to how "stretched out" the shape is and he uses the example of a long, thin rectangle such as the one in the right panel of Figure 2.1 . Separation refers to a shape having two or more disconnected components, and puncturedness refers to the existence of "holes" in a shape.

Taylor argues that indentations are the most important of these four. Indentations in districts could result from attempts to avoid including a group of voters in the district or to pack them into a neighboring one. He hypothesizes the number of reflex angles (angles whose internal measure is greater than 180°) is related to how "indented" a district is, which in turn could be evidence of gerrymandering.

Recall that a polygon is a subset of $\mathbb{R}^2$ that is bounded by a finite chain of straight line segments closing in a loop. For a polygonal district $P \subset \mathbb{R}^2$ let R and N be the number of reflex and non-reflex angles respectively. Taylor's *indentation index* for district P is

$$I(P) = \frac{N - R}{N + R} \ . \tag{2.1}$$

When P is convex all angles are non-reflexive, so $R = 0$ and $I(P) = 1$. Since $I(P) \leq 1$, a convex polygon is considered the least gerrymandered by this measure. Young [15] demonstrates several problems with this measure using the example of a "squeezed" square (see left panel of Figure 2.1). Since a square is convex, it has an indentation score of one and is considered to be not gerrymandered by the indentation criteria. But one can obtain a figure with a score of zero by slightly "pushing" each face towards the interior. The indentations in this polygon are so mild that the shape should not be considered any more gerrymandered than the square, but Taylor's measure gives it a lower score than the long thin rectangle, and pushing the faces in further has no effect on the score. Young's example illustrates the problem with Taylor's measure, namely that the measure cannot effectively discriminate between more or less gerrymandered figures since it is unaffected by the magnitude of the indentations.

The aspect of shape that most measures of gerrymandering attempt to quantify is that of compactness (as distinct from the topological concept). Compactness is the notion that the bulk of the shape should be "closely packed" and its use is advocated by Polsby and Popper [11]. Simple shapes like circles, squares and hexagons conform to the intuitive notion of compactness [10]. Unfortunately there is no agreed upon definition of compactness either and there are conflicting definitions in the gerrymandering literature used by different authors.

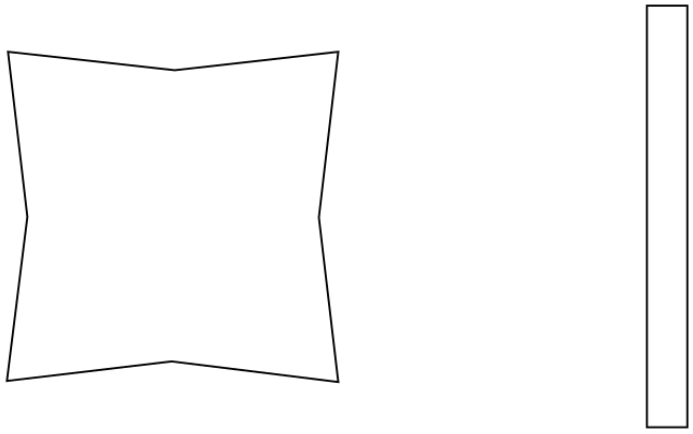

Figure 2.1: Left: a "squeezed" square. Right: a long, thin rectangle.

Taylor asserts the four aspects of shape he identifies are in fact aspects of compactness while others treat compactness as a separate property [9, 11, 15]. Young criticizes Taylor's measure since shapes that have their area dispersed such as a long and thin rectangle such as the one shown on the right panel of Figure 2.1, are compact according to his indentation criteria [15, 14].

Reock [12] argues that compact shapes are those that make efficient use of perimeter to enclose area and suggests the circle is the ideal shape since it has the lowest perimeter to area ratio. He proposes using the ratio of the shape's area to the area of the smallest circumscribing circle,

$$\frac{\text{Area of circumscribing circle}}{\text{Area}} \,. \tag{2.2}$$

However, Young has pointed out that an arbitrarily misshapen district can receive a high score so long as it is contained within a compact area [15]. The counterexample Young uses is that of a "coiled snake" that is coiled in a roughly circular area (see right panel of Figure 2.3). This shape fills in much of the area of its circumscribing circle yet does not conform to our intuitive notion of gerrymandering. In their

extensive survey of gerrymandering methods, Polsby and Popper [11] also criticize the measure for being too sensitive to extreme points and use the example of a square with a "spike" protruding from one of the square's corners (see Figure 2.2), the angular configuration of the spike can have a dramatic effect on the size of the circumscribing circle. If the spike is parallel along one of the square's diagonals, the circumscribing circle will enclose a significant amount of area not covered by the square. However, "folding" the spike towards the square shrinks the size of the circle resulting in a higher compactness score. There is no rationale for considering the "folded spike" to be less gerrymandered than the former configuration [11].

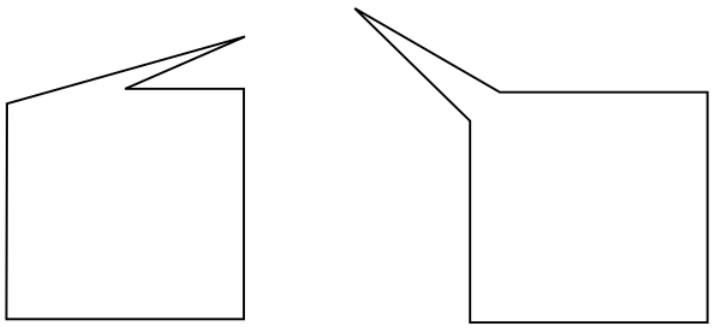

Figure 2.2: Left: square with folded spike. Right: square with protruding spike.

Polsby and Popper [11] argue that perimeter is an aspect of shape important for gerrymandering since it is directly related to the configuration of the shape and propose using the area of the circle with the same perimeter instead of the circumscribing circle,

$$\frac{\text{Area of the circle of equal perimeter}}{\text{Area}}. \tag{2.3}$$

The incorporation of perimeter into the compactness measure solves the problem

raised by Young, since an arbitrarily misshapen district like the "coiled snake" will have a longer perimeter as the district border wanders about a confined area.

The measure created by Boyce and Clark [2] attempts to take the configuration of the district borders into account directly rather than using a single related parameter like area or perimeter. Let r_i from $i = 1, \ldots, n$ be equally spaced radii originating at the center of the shape and terminating at the farthest intersection point and let ℓ_i be the length of each r_i. The index is computed by

$$C(P) = \sum_{i=1}^{n} \left| \frac{\ell_i}{\sum_i \ell_i} - \frac{1}{n} \right| \tag{2.4}$$

which has minimum value of 0 which occurs when all r_i are of equal length. For example, this will occur when P is a circle [9]. The measure is sensitive to the value of n and MacEachren [9] finds that the measure stabilizes for values of $n > 100$. Young points out this measure is susceptible to the same problem as Reock's measure (2.2) since it does not take into account the perimeter of the shape.

Another popular measure is the *moments of area method* which captures how the points of the shape are distributed about an axis [9]. Let dA be a small piece of area, then the *second moment of area* in polar form is

$$\int_P r^2 dA$$

The *moments of area ratio* is

$$\frac{A^2}{2\pi \int_P r^2 \, dA} \tag{2.5}$$

which takes a maximum value of 1 when P is a circle. Like Boyce and Clark's measure in equation 2.4, the moments of area method tries to take the whole of the shape into account. However, this method has the same problem as those described in [12] and [11] since shapes like the coiled snake would score well on this measure [15]. Polsby and Popper also points out for that shapes where the bulk of the area is

concentrated about the center but have distorted and irregular boundaries would still receive a high compactness score according to 2.5. They use the example of "sawblade" pictured in Figure 2.3 to illustrate this point.

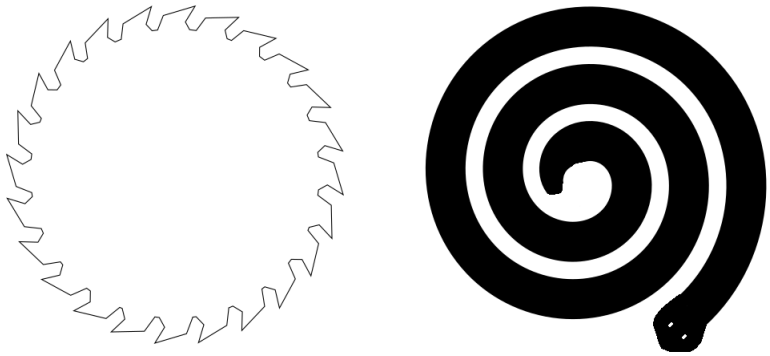

Figure 2.3: Left: sawblade. Right: coiled snake.

Chambers and Miller [3] introduce a new concept and measure they call "bizarreness" and demonstrate its use for identifying gerrymanders. They argue that a key feature of gerrymandered districts is that they are highly non-convex. Their measure is based on the probability the shortest path connecting two randomly selected points in the district is itself contained in the district. If the district boundary follows the state boundary then, irregular shape may not be the result of gerrymandering, so they adjust their measure to use the probability the path lies in the state.

Hodge et al. [6] devise a simpler version of the measure in [3] they call the *convexity coefficient*. Let $P \subset \mathbb{R}^2$ be a polygon and A its area. For a point $p \in P$, let $V_P(p) \subseteq D$ be the set of all points $q \in P$ such that the line segment pq is contained in P. $V_P(p)$ is called the *visible set* and its area is denoted by $A_P(p)$. For a given point p, the probability that a point q is visible from p is the ratio of the area of the

visible set to the total area A, $\frac{A_P(p)}{A}$. The convexity coefficient $\chi(P)$ is defined as

$$\chi(P) = \int_P \frac{A_P(p)}{A} dp \ . \tag{2.6}$$

Like Taylor [14]'s measure, the convexity coefficient assumes that convex shapes are ideal. Chambers et. al. and Hodge et. al. demonstrate its use on several districts known for giving misleading results on other measures such as Illinois' 4th and Maryland's 6th districts and conclude the convexity-based approach is the superior to Reock's measure among others discussed here [6, 3].

Ansolabehere and Palmer [1] argue for using several measures in conjunction with each other. To make results comparable across measures, they use the original gerrymander as the standard to which all other districts are compared. This relative approach has its appeal in that it allows the ranking of districts into more and less gerrymandered than some standard. While the choice of the original gerrymander as the standard may please historians and gerrymandering enthusiasts, it is still arbitrary. However, the use of multiple measures on a relative scale is likely the best approach for quantifying and identifying gerrymanders since no single measure adequately captures all the information relevant to gerrymandering.

Since each measure uses an implicit or explicit comparison to a known shape (or family of shapes), a simple way to classify measures could be based upon their *a priori* assumption of what constitutes an ideal shape or family of shapes. All measures can be can be categorized according to whether comparisons are made to a single shape or to a family of shapes. These categories can be further refined by the specific shape used and by what characteristics define a specific family of shapes. The measures created by Polsby and Popper [11], Reock [12] and Boyce and Clark [2] and the moment of area method each assume the circle represents the ideal shape. However it is impossible to create a districting scheme using circles. The

measures created by Taylor [14] and Chambers and Miller [3] compare the district to the family of convex shapes, i.e. all convex shapes are considered ideal and highly non-convex ones are indicative of gerrymandering. Relaxation of the restriction to a single impracticable shape such as a circle and enlarging the family of possible shapes is desirable. Of these family-based measures, Chambers and Miller's measure is superior than Taylor's because it takes the configuration of the boundary into account.

CHAPTER 3

THE BOUNDARY INTERSECTION NUMBER

3.1 Definitions and Theorems

Since gerrymandering has no well-defined definition, intuition can serve as a guide to selecting a good measure or combination of measures. An intuitive property that a non-gerrymandered district should possess would be what one might call the "as the crow flies" property, namely, *that a person inside the district should be able to travel in a straight line to any other part of the district without exiting the district.* The convexity coefficient of [3] and [6] exactly quantifies the probability that a randomly positioned observer can do this for a randomly selected destination inside the district. Here we introduce a new measure that similarly uses convexity, but focuses on capturing distortions in the polygon's boundaries. To be precise, we define a polygon $P \subset \mathbb{R}^2$ to be a planar set whose boundary is a collection of n points $v_1, \ldots, v_n$ and n edges $v_1v_2, \ldots, v_{n-1}v_n, v_nv_1$ such that no pair of non-consecutive edges share a point and no edges cross each other.

Another intuitive property of a non-gerrymandered district is that *a person traveling in a straight line should cross the district boundary exactly once on their way out of the district.* To be precise, for a point p in the district a person walking in the direction θ intersects the boundary k times. Since the person exits the district, this implies the person crosses the boundary at least once, so we have $k \geq 1$. In a non-gerrymandered district we would expect that k will be small for many points and many directions. We can formalize this concept and create a measure of gerrymandering called the *boundary intersection number* or BIN.

Definition 3.1.1. Boundary Intersection Number Let P be a polygon and $(p, \theta) \in P \times [0, 2\pi)$ a pair of random variables with joint probability distribution $Pr(p, \theta)$. Then the number of boundary intersections is a random variable $k : P \times [0, 2\pi) \longrightarrow \mathbb{Z}_+$. We define the BIN to be the expectation of $k(p, \theta)$.

$$\gamma(P) = \int_p \int_\theta k(p, \theta) Pr(p, \theta) d\theta dp \tag{3.1}$$

Since $k \geq 1$, we must also have $\gamma(P) \geq 1$ for all $P \subset \mathbb{R}^2$.

The BIN can be interpreted to be the expected number of times that a randomly placed observer crosses the boundary of the district while walking in a straight line in a randomly chosen direction. We assume that a perfectly convex polygon represents the ideal shape class for non-gerrymandered districts and deviations from convexity are indications of gerrymandering. In the following two proofs we show that convex polygons have $\gamma(P) = 1$, which establishes that the lower bound of the BIN corresponds to the least possible amount of gerrymandering.

Lemma 3.1.2. Let P be a polygon. If P is not convex, then $\gamma(P) > 1$.

Proof: Since P is not convex, there exists vertices A, B and C on P such that the angle formed by the joining of the segments AB and BC has interior angle greater than 180°. Let m_{BA} and m_{BC} be the midpoints of these segments. For a point p define $N_\epsilon(p) = \{q \in \mathbb{R}^2 \mid |q - p| < \epsilon\}$ be the ϵ-neighborhood about p and $Q_\epsilon(p) = P \cap N_\epsilon(p)$.

Let L' be the line bisecting the angle ABC and let L be the line perpendicular to L' passing through B. Define $\epsilon_{BA} = \min_{\ell \in L} |\ell - m_{AB}|$ and ϵ_{BC} similarly. Choose $\epsilon < \min\{\epsilon_{BA}, \epsilon_{BC}\}$. Thus the line segment pq must pass through AB and BC for all $p \in Q_\epsilon(m_{BA})$ and $q \in Q_\epsilon(m_{BC})$ (see Figure 3.1). Therefore $k(p, q) > 1$ for all $p \in Q_\epsilon(m_{BA})$ and $q \in Q_\epsilon(m_{BC})$, so we have

$$\gamma(P) = \int_{p \in P} \int_{\theta} k(p,\theta) Pr(p,\theta) d\theta dp \geq \int_{p \in Q_\epsilon(m_{BA})} \int_{q \in Q_\epsilon(m_{BC})} k(p,q) Pr(p,q) dq dp > 1$$

□

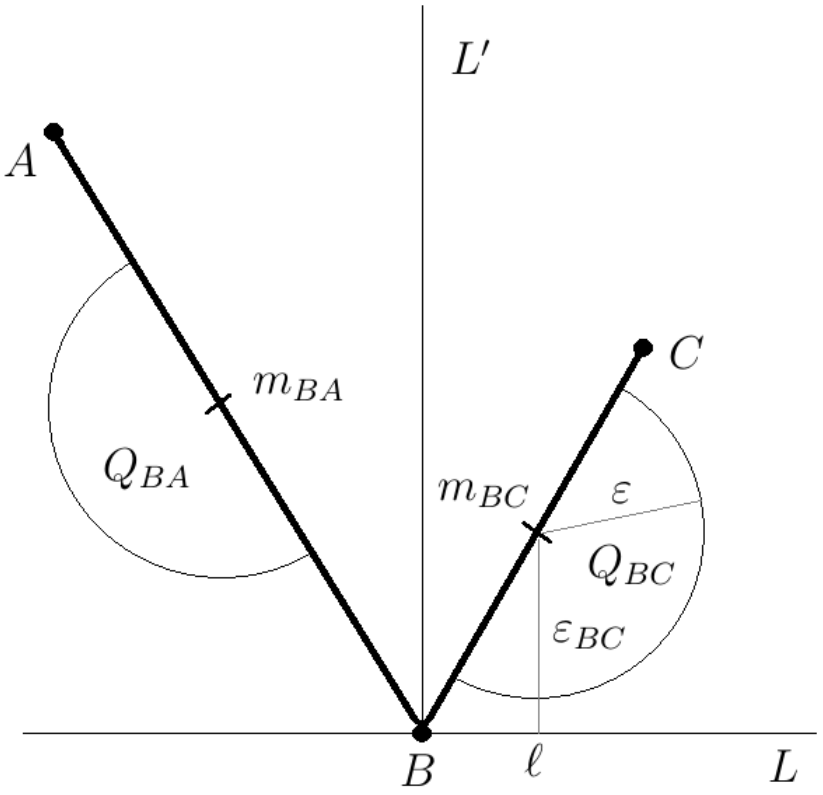

Figure 3.1: Diagram for Lemma 3.1.2.

Theorem 3.1.3. Let P be a polygon. Then $\gamma(P) = 1$ if and only if P is convex.

Proof: If P is convex, then any ray starting in P will cross the boundary exactly once while exiting P. Otherwise, if a ray intersected the boundary more than once, the second intersection would occur while the ray was reentering P.

Thus there would exist some point $q \in P$ such that part of the line segment pq that lies along the ray would lie outside the polygon, contradicting convexity.

For the other direction, by the contrapositive of the Lemma 3.2, if $\gamma(P) = 1$ then P is convex. $\square$

Theorem 3.1.4. If a polygon has P has n faces, then $\gamma(P) \leq n - 1$.

Proof: In the case where P is convex, $\gamma(P) = 1$ by Theorem 3.3. Since $n \geq 3$ (by definition of a polygon) therefore we have $\gamma(P) \leq n - 1$.

In the case where P is not convex, we use the fact that the expectation of a random variable X lies between the minimum and maximum values of said random variable, i.e. that $X_{\min} \leq E[X] \leq X_{\max}$. Denote $\min_{p,\theta} k(p, \theta) = k_{\min}$ and $\max_{p,\theta} k(p, \theta) = k_{\max}$.

It is clear that for any polygon P, there must exist a point $p \in P$ and a direction θ such that $k(p, \theta) = 1$. Thus $k_{\min} = 1$.

Let p be a point in P and θ a direction. Since P is a polygon, p is enclosed by the boundary of P. Therefore there must exist at least one edge which the ray in the direction of θ does not intersect. Therefore $k(p, \theta) \leq n - 1$ for all p and θ. Thus $k_{\max} \leq n - 1$.

Combining these two results, we have $1 = k_{\min} \leq \gamma(P) \leq k_{\max} \leq n - 1$. Therefore $\gamma(P) \leq n - 1$. $\square$

3.2 Estimation and Computation

We use the Law of Large Numbers to estimate $\gamma(P)$ using a Monte Carlo method. The Law of Large Number states that average of n independent identically distributed random variables converges to the expectation of the distribution as $n \to \infty$ [4]. Our random variable k does not have a known analytic expression for

its distribution so it must be estimated from the distributions of the constituent random variables p and θ, whose distributions are assumed to be uniform over their respective sample spaces. Furthermore, we assume that p and θ are independent, so each pair (p, θ) has the same probability since $Pr(p, \theta) = Pr(p)Pr(\theta) = \frac{1}{A}\frac{1}{2\pi} = \frac{1}{2\pi A}$.

Let $(p_i, \theta_i) \in P \times [0, 2\pi)$ for $i = 1, \ldots, n$ be n independent random variables with identical distributions $Pr(p, \theta)$. Then the $k_i = k(p_i, \theta_i)$ are independent and identically distributed. By the Law of Large Numbers $\frac{1}{n}\sum_{i=1}^{n} k_i \to \gamma(P)$ as $n \to \infty$, so we can estimate 3.1 by the sample mean of the k_i, which can be expressed as the sum

$$k_i(p_i, \theta_i) = \sum_{e \in E(P)} I_i(e) \tag{3.2}$$

where $I_i(e)$ is a binary indicator variable indicating whether or not the ray defined by θ_i intersects the edge e. In order to compute 3.2 we must solve the problem of determining $I(e)$ given a point and a direction. Fortunately this can be solved using some basic linear algebra.

We first make precise our definitions for rays and edges. Let $\mathbf{r} \in \mathbb{R}^2$ be a direction vector. A ray $\mathbf{R}$ in the direction of $\mathbf{r}$ can be parameterized as $\mathbf{R}(t) = t\mathbf{r}$ for $t \in \mathbb{R}^+$. Next, let u and v be adjacent vertices in P and let $\mathbf{u}$ and $\mathbf{v}$ be their respective coordinate vectors. The edge uv in $\mathbb{R}^2$ is the convex set

$$E_{uv} = \{(1 - \alpha)\mathbf{u} + \alpha\mathbf{v} \mid \alpha \in (0, 1)\} .$$

We next define the *visibility cone defined by* E_{uv} to be the set

$$VC_{uv} = \{\beta\mathbf{e} \mid \mathbf{e} \in E_{uv} \ , \beta > 0\} .$$

It is clear that any ray with a direction vector in a visibility cone will intersect the edge that defines the cone.

Theorem 3.2.1. Let $\mathbf{R}$ be a ray starting at the origin with direction vector $\mathbf{r}$. Then $\mathbf{R}$ intersects uv if and only if $\mathbf{r}$ is a positive linear combination of $\mathbf{u}$ and $\mathbf{v}$.

Proof: If $\mathbf{R}$ intersects the edge uv, then there exists $t > 0$ and $\alpha \in (0, 1)$ such that $t\mathbf{r} = (1 - \alpha)\mathbf{u} + \alpha\mathbf{v}$. Thus $\mathbf{r} = \frac{1-\alpha}{t}\mathbf{u} + \frac{\alpha}{t}\mathbf{v}$ is a positive linear combination. Conversely, let $\mathbf{r} = a\mathbf{u} + b\mathbf{v}$ be a positive linear combination of $\mathbf{u}$ and $\mathbf{v}$. We show that $\mathbf{r} \in VC_{uv}$. We choose $\beta = a + b$ and $\alpha = \frac{b}{a+b}$. Thus,

$$a = a + b - b = (a + b)\left(1 - \frac{b}{a + b}\right) = \beta(1 - \alpha)$$

and

$$b = (a + b)\left(\frac{b}{a + b}\right) = \beta\alpha$$

therefore $\mathbf{r} = \beta((1 - \alpha)\mathbf{u} + \alpha\mathbf{v})$ which implies that $\mathbf{r} \in VC_{uv}$. Therefore $\mathbf{R}$ intersects the edge uv. □

We can determine whether $\mathbf{R}$ intersects uv by solving the linear system $\mathbf{A}_{uv}\mathbf{x} = \mathbf{r}$ where $\mathbf{A}_{uv}$ is a 2×2 matrix whose columns are $\mathbf{u}$ and $\mathbf{v}$. The direction vector $\mathbf{r}$ intersects uv if $\mathbf{x} > \mathbf{0}$ (that is, each coordinate is positive).

The pseudo-code on the following page gives the procedure for the computation of the BIN estimate $\hat{\gamma}(P)$ using Lemma 3.1.

3.3 Implementation

We implement Algorithm 1 in Matlab for use with shapefiles compiled by [8]. Shapefiles are geographic information system (GIS) files that store non-topological geometry and metadata. Shapefiles support point, line and polygonal features [13]. Below we describe the implementation along with the functions created for it, the

Algorithm 1 Monte Carlo Procedure for Estimating BIN

Require: Sample of n points p_i uniformly from P
  for $i = 1, \ldots, n$ **do**
    Sample a direction θ_i uniformly from $[0, 2\pi)$
    Form vector $\mathbf{r}_i$ starting at p_i with direction θ_i
    Recenter P about p_i
    for all $e \in E(P)$ **do**
      solve $\mathbf{A}_e\mathbf{x} = \mathbf{r}_i$
      if $\mathbf{x} > \mathbf{0}$ **then**
        $I_i(e) \leftarrow 1$
      else
        $I_i(e) \leftarrow 0$
      end if
    end for
  end for
  return $\hat{\gamma} = \frac{1}{n} \sum_i \sum_e I_i(e)$

code for which can be found in the appendix. Unless otherwise noted, all functions were written for this paper.

The main function created for the implementation is `estimate_y()`, which computes an estimate of BIN for a user-defined number of sample points and also contains options for plotting the shape, sample points and sample vectors. This function calls upon two secondary functions `shapesample()` and `sample_k()`. The `shapesample()` function samples a user-defined number of points from inside the inputted shape. It does this by sampling randomly from inside the polygon by computing randomly selected points from inside the rectangle that encloses the polygon and discarding points that lie outside the polygon boundary. The coordinates of the enclosing rectangle are contained in a field in the shapefile, so they do not need to be computed. The function `sample_k()` generates a random intersection number k by computing the number of times a randomly generated vector centered at a given point $p = (x, y)$ crosses the boundary of the polygon.

The function `estimate_y()` calls on `shapesample()` to generate a set of points and then applies the `sample_k()` to these points. The average of the computed k's is returned as the BIN estimate.

Both `shapesample()` and `sample_k()` make use of a helper function called `shapeparts()`. Shapefiles store polygons as a list of coordinates. When Matlab parses the shapefile, disconnected components of a polygon are separated by an NaN value in the list, which makes it necessary to parse the coordinate list and separate each component for processing. The `shapeparts()` function takes a shapefile and returns a Matlab cell array where each entry contains the list of coordinates for a single, connected component of the shape. All operations are then conducted on each component and the results aggregated. See the comments contained in the code appendix for details of the inner workings.

3.4 Results

In order to use Algorithm 1 to compute an estimate of the BIN score, an appropriate value of n must be selected. The central limit theorem tells us to expect the estimated values $\hat{\gamma}_n$ produced by Algorithm 1 to converge with increasing n, so we must choose n to be sufficiently large. We do this empirically by computing the BIN scores over a range of values for n. For each value of n, m BIN scores are computed using Algorithm 1 and averaged. As n grows large the variance of the m scores decreases and the distribution of the BIN scores about the mean grows tighter. Figure 3.2 illustrates this using Illinois' 4th Congressional district. The distribution of BIN scores grows tighter as the scores begin to converge as n grows larger.

The rate of convergence will vary with the complexity of the boundary. Therefore n must be comparatively large for estimating the BIN of a gerrymandered district. We have found that $n > 1000$ tends to be a suitably large for the most complex districts. In order to reduce the possibility of variation effecting the BIN estimate, we recommend using the average of m BIN scores for a suitable m. The average computation time for the set is a little over three minutes when $n = 1000$ and computing the BIN scores for the entire House of Representatives takes several hours. This can be accomplished faster by running computations in parallel.

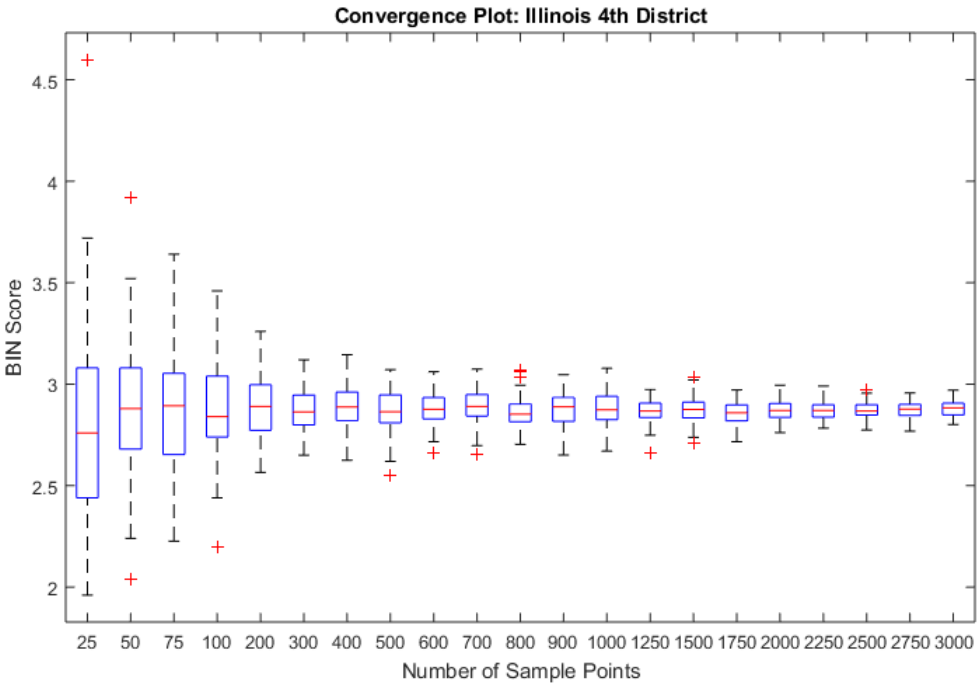

Figure 3.2: Convergence of BIN score for Illinois 4th District

The BIN scores were computed for all districts in the 114th Congress. The distribution of scores is shown in Figure 3.3 below which is asymmetrical and skewed rightward. Some of the outliers are potentially caused by extremely irregular coastlines or other natural boundaries.

Figure 3.4 shows the BIN scores computed for six districts, increasing in boundary complexity and BIN score. The top-left tile shows a perfectly non-gerrymandered district, the state of Wyoming, which is its own congressional district and has a BIN score of 1. The top-right tile shows Iowa's 1st District with a BIN score of 1.46. This indicates a slight gerrymandering, as the state of Iowa is rather square and has no extremely complex natural boundaries.

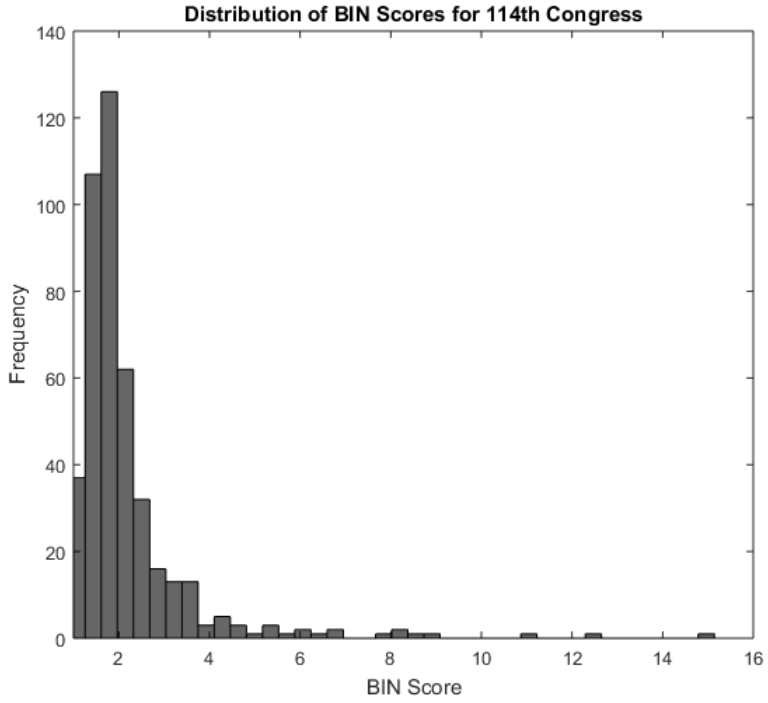

Figure 3.3: Distribution of BIN scores for the 114th Congress

The middle tiles of Figure 3.4 show two cases of gerrymandering. The middle-left tile shows Illinois' 4th District, which is a deliberate case of "packing" two parts of Chicago together to form a single district. The middle-right tile shows Maryland's 3rd District with a BIN score of 4.5. It is gerrymandered since only a very small portion of its boundary in the south-east corresponds to natural contours of the Chesapeake Bay and it contains disconnected pieces that are not islands.

The bottom two tiles of Figure 3.4 illustrate the disadvantages of the BIN score. Extremely complex natural boundaries can yield extremely high BIN scores resulting in a "false positive" for gerrymandering. An example of this is contained in the bottom-left tile, which shows an example of a district high BIN score but is not gerrymandered. Maryland's 4th District's high BIN score is mainly due to the

extremely rough coastline along the Chesapeake Bay. As noted by Chamber and Millers, complex natural boundaries can inflate their convexity coefficient and same is true of BIN for the same reasons. The bottom-right tile shows Louisiana's 1st District which has the highest BIN score of the 114th Congress of 15.98. This district is gerrymandered since it contains a disconnected component that is not an island and most of the irregular northern boundaries do not correspond to natural features. However, the BIN score is so high because of the extremely irregular Louisiana coastline and its many small islands. Intuitively, this district is no more gerrymandered than the districts in the middle panels but due to its complex geography it is given a BIN much higher than is likely appropriate.

One possible way to remedy this would be to ignore all subsequent boundary intersections after the first intersection for rays that intersect natural boundaries. Another is to count only intersections of boundaries that are not natural. This requires that metadata on the district boundary be available which was unavailable for this set.

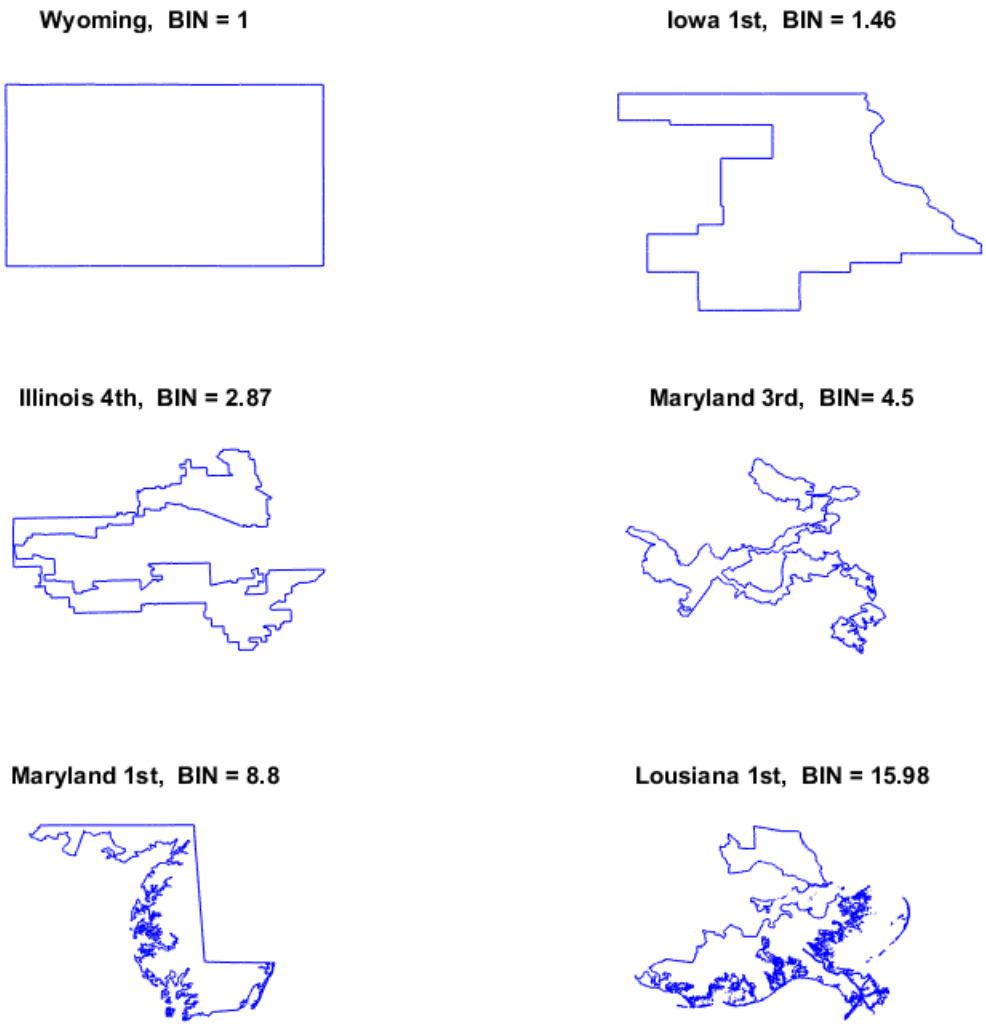

Figure 3.4: Districts of the 114th Congress

CHAPTER 4

CONCLUSION

The BIN score is a new gerrymander measure that uses boundary distortions to measure gerrymandering and belongs to the family of measures whose baseline ideal for a non-gerrymandered shape is convex. The measure represents the average number of times a randomly placed observer will cross the boundary of the district while traveling in a randomly chosen direction out of the district.

The BIN score, like Taylor [14], Reock [12] and Boyce and Clark [2] attempts to measure gerrymandering using the configuration of the district boundary. It has an advantage over Taylor's indentation index because BIN accounts for the magnitude of the boundary distortions because of integration over all points of the district. A polygon whose boundary distortions are confined to a small section of the boundary will have a low BIN score because observers starting at points distant from the distorted section will have a low probability of crossing it relative to observers starting at nearby points. The BIN score is superior to the measures of Reock and Boyce and Clark because it is not susceptible to the problem illustrated by the example of the "coiled snake", which would receive a very high BIN score.

The BIN score also has disadvantages compared to the convexity coefficient and the moments of area method in that shapes such as the "sawblade" that have low area dispersion and relatively low non-convexity but have highly irregular boundaries will appear more gerrymandered according to the BIN score than may be appropriate. This problem becomes acute when computing the BIN score of districts with irregular natural boundaries such as coastlines and rivers and in

extreme cases may falsely imply a district is gerrymandered when it is not.

We recommend using BIN in conjunction with the convexity coefficient. If a district has a high coefficient coefficient and a low BIN score, we conclude that the district is both highly convex with few boundary distortions and is thus to be considered not gerrymandered. However, if a district has a high BIN score and a high convexity score, we may be dealing with a shape like the "sawblade" or a district whose distortions are due to geographic features and we may wish give weight to the convexity coefficient. A low convexity coefficient and a high BIN score indicates a probable gerrymander.

There are two directions for future work on the BIN score. The first is to modify the BIN score and computation procedure to account for natural boundaries. This will require finding the appropriate data about geographic features and integrating it with the district boundary data. Another direction is to investigate the relationship the BIN score has with other measures of gerrymandering and compare their relative effectiveness at identifying gerrymanders.

BIBLIOGRAPHY

[1] Ansolabehere, S. and Palmer, M. (2016). A two-hundred year statistical history of the gerrymander. *Ohio St. LJ*, 77:741.

[2] Boyce, R. R. and Clark, W. A. (1964). The concept of shape in geography. *Geographical Review*, 54(4):561–572.

[3] Chambers, C. and Miller, A. D. (2010). A measure of bizarreness. *Quarterly Journal of Political Science*, 5(1):27–44.

[4] Feller, W. (1). *An Introduction to Probability Theory and its Applications, vol. 2*. John Wiley & Sons.

[5] Hatfield, M. O. (1997). *Vice Presidents of the United States, 1789–1993*. Government Printing Office, Washington DC.

[6] Hodge, J. K., Marshall, E., and Patterson, G. (2010). Gerrymandering and convexity. *The College Mathematics Journal*, 41(4):312–324.

[7] Ingram, C. (2015). This is the best explanation of gerrymandering you will ever see. https://www.washingtonpost.com/news/wonk/wp/2015/03/01/this-is-the-best-explanation-of-gerrymandering-you-will-ever-see/?utm˙term=.5cf029656cab/.

[8] Lewis, J. B., DeVine, B., Pitcher, L., and Martis, K. C. (2013). Digital boundary definitions of united states congressional districts, 1789-2012. http://cdmaps.polisci.ucla.edu/.

[9] MacEachren, A. M. (1985). Compactness of geographic shape: Comparison and evaluation of measures. *Geografiska Annaler. Series B. Human Geography*, pages 53–67.

[10] Niemi, R. G., Grofman, B., Carlucci, C., and Hofeller, T. (1990). Measuring compactness and the role of a compactness standard in a test for partisan and racial gerrymandering. *The Journal of Politics*, 52(04):1155–1181.

[11] Polsby, D. D. and Popper, R. D. (1991). The third criterion: Compactness as a procedural safeguard against partisan gerrymandering. *Yale Law & Policy Review*, 9(2):301–353.

[12] Reock, E. C. (1961). A note: Measuring compactness as a requirement of legislative apportionment. *Midwest Journal of Political Science*, 5(1):70–74.

[13] Staff, E. S. R. I. (1998). Esri shapefile technical description. https://www.esri.com/library/whitepapers/pdfs/shapefile.pdf.

[14] Taylor, P. J. (1973). A new shape measure for evaluating electoral district patterns. *American Political Science Review*, 67(03):947–950.

[15] Young, H. P. (1988). Measuring the compactness of legislative districts. *Legislative Studies Quarterly*, pages 105–115.

```
function BIN = estimate_y(N_points, shape, plot_shape,
   plot_arrows, plot_samples)


% This function estimates BIN using the Monte Carlo method
    with optional
% plotting. The shapesample() function is called to
   generate sample points
% from inside the shape. The sample_k() function is then
   applied over the
% list of sample points, returning the number of
   intersections of randomly
% choosen direction vectors starting at each sample point.
    The estimated BIN
% is the mean of these intersections.


% get sample points using shapesample()
shape_samples = shapesample(N_points,shape,plot_samples);
disp('Sampling Complete. Computing BIN...')


% compute boundary intersections k
[k, vec] = arrayfun(@(x,y) sample_k(x,y,1,shape),
```

```
    shape_samples(:,1), shape_samples(:,2),'un',0);

% convert k from cell to array
temp = zeros(length(k),1);
for i = 1:length(k)
    temp(i,1) = k{i};
end
BIN = mean(temp); % compute BIN estimate

msg = strcat('Estimated BIN = ',num2str(BIN));
disp(ms)

% plot shape with direction arrows
if (plot_arrows == 1 && plot_shape == 1)
    figure
    axis([shape.BoundingBox(1,1) shape.BoundingBox(1,2)
       shape.BoundingBox(2,1) shape.BoundingBox(2,2)])
    plot(shape.X,shape.Y, 'black')
    hold on

    P1 = [shape.BoundingBox(1,1); shape.BoundingBox(1,2)];
    P2 = [shape.BoundingBox(2,1); shape.BoundingBox(1,2)];

    d = sqrt(((P1(1) - P2(1))^2 + (P1(2) - P2(2))^2));
```

```
s = 0.1*d; %set direction arrow magnitude

% plot direction arrows
for i=1:N_points
      p0 = [shape_samples(i,1), shape_samples(i,2)]';
         p1 = [s*vec{i}(1)+shape_samples(i,1), s*vec{i
         }(2)+shape_samples(i,2)]';
      x0 = p0(1);
      y0 = p0(2);
      x1 = p1(1);
      y1 = p1(2);
      plot([x0;x1],[y0;y1]);   % Draw a line between
         p0 and p1

      p = p1-p0;

      alpha = 0.3;  % Size of arrow head relative to
         the length of the vector
      beta = 0.2;   % Width of the base of the arrow
         head relative to the length

      hu = [x1-alpha*(p(1)+beta*(p(2)+eps)); x1; x1-
         alpha*(p(1)-beta*(p(2)+eps))];
      hv = [y1-alpha*(p(2)-beta*(p(1)+eps)); y1; y1-
         alpha*(p(2)+beta*(p(1)+eps))];
```

```
            hold on
            plot(hu(:)',hv(:)')
            hold on
        end
        title(strcat(shape.STATENAME, ' District No',shape.
            DISTRICT, ', N = ',num2str(N_points), ', BIN = ',
            num2str(BIN)), 'FontSize',12)
        axis off

        % plot shape shape only
    elseif (plot_arrows == 0 && plot_shape == 1)
        figure
        axis([shape.BoundingBox(1,1) shape.BoundingBox(1,2)
            shape.BoundingBox(2,1) shape.BoundingBox(2,2)])
        plot(shape.X,shape.Y, 'black')
        axis off
        title(strcat(shape.STATENAME, ' District No',shape.
            DISTRICT, ', BIN = ',num2str(BIN)), 'FontSize',12)
    else
        %%%%%%%%
    end
    end

    function [k , rvec]= sample_k(x,y,N,shape)

    % Takes a point (x,y), the number of desired number of
```

```
   random vectors per
% point N (set to 1 by default when called by estimate_y),
    and a shape
% (in shapefile) and computes k (number  for boundary
% intersections). This is done by translating the shape to
    be centered on
% the point (x,y) which becomes the origin of a coordinate
    system whence
% random vector(s) generated and the number of boundary
   intersections by
% computed by solving a linear system for each edge of the
    polygon.

% translates shape to be centered at (x,y)
newShape = translateToOrigin(x,y, shape.X, shape.Y);
% stores X and Y coordinates of new shape as struct array
newShape = struct('X',newShape(:,1)', 'Y', newShape(:,2)')
   ;
% apply shapeparts() to newShape
[shape_parts,~] = shapeparts(newShape);
% generate N random unit vectors
rvec = randomUnitVector(N);
% apply comp_k() function to compute boundary
   intersections of each vector
k = arrayfun(@(x,y) comp_k(x,y, shape_parts),rvec(:,1),
```

```
    rvec(:,2));
% return total number of intersection
k = sum(k);
end

function k = comp_k(x,y, shape_parts)
% applies the edgeIntersect() function to each component
    of shape_parts

vec = [x,y];
temp = cellfun(@(shapes) edgeIntersect(vec, shapes),
    shape_parts);
k = sum(temp);
end

function vrand = randomUnitVector(n)

% returns n random unit vectors centered at the origin

theta_rand = 2*pi*rand([n,1]);
vrand = zeros(n,2);
vrand(:,1) = cos(theta_rand);
vrand(:,2) = sin(theta_rand);
end
```

```
%% translateToOrigin()

function[newCoords] = translateToOrigin(x0, y0, X,Y)
%translates the coordinates (X,Y) about (x0,y0)
newCoords = [X - x0; Y - y0]';
end

%% edgeIntersect()

function Y = edgeIntersect(vec, shape)

% Checks how many times the vector vec intersects the
   boundary of shape
% by applying checkEdge() to each pair of vertices that
   define an edge

n = size(shape,1);
temp = zeros(n,1);

for i=1:(n-1)
    temp(i) = checkEdge(vec, shape(i:(i+1),:));
end
Y = sum(temp);
end
```

```
%% checkEdge()

function k = checkEdge(vec, A)

% Determines if the vector vec is a positive linear
   combination of the
% rows of the 2x2 matrix A. To save computational time,
   vectors that lie
% in a quadrant different than either of the rows of A are
    ignored.

Qv = checkQuadrant(vec); %get quadrant of input vector

%get quadrants of rows of A
A1=checkQuadrant(A(1,:));
A2=checkQuadrant(A(2,:));

if (A1 == Qv || A2 == Qv)
    %either row of A lies in the same quadrant as the
       input vector, check
    %the sign of the coefficients of Av = b
    b = linsolve(A',vec');
    if (b(1)>=0 && b(2)>=0)
        k = 1;
    else
```

```
        k = 0;
    end
else
    k = 0;
end
end

%% checkQuadrant()

function quadrant = checkQuadrant(vec)

% Finds the quadrant of a 2D vector

x = vec(1); y = vec(2);

if (x >= 0 && y >= 0)
    quadrant = 1;
elseif (x < 0 && y >= 0)
    quadrant = 2;
elseif (x < 0 && y < 0)
    quadrant = 3;
else
    quadrant = 4;
end
end
```

```
function D_sample = shapesample(N, shape, plt)

% This function computes N uniform samples from inside the
    polygon in the shapefile
% record with optional plotting. Sampling is done by
   sampling points from
% inside the bounding box of the shape using sampleBox().
   The function inshape()
% determines if the point is contained in the polygon.
   Sample points falling outside the
% polygon and discarded. This process is repeated until
   enough samples
% inside the polygon have been found.


sample_points = sampleBox(N, shape);
                                                    %
   sampleBox()
parts = shapeparts(shape);

   % shapeparts()
q = arrayfun(@(x,y) inshape(x,y,parts), sample_points(:,1)
   , sample_points(:,2));   % apply inshape()
D_sample = [sample_points(q == 1,1), sample_points(q ==
   1,2)];
```

```
% continue generating samples until N samples is reached.

k = sum(q);   %while number of samples inside polygon < N,
    continue sampling
while k < N
    sample_points = sampleBox(N,shape);
    q = arrayfun(@(x,y) inshape(x,y,parts), sample_points
       (:,1), sample_points(:,2));
    k = k + sum(q);

    newSample = [sample_points(q == 1,1), sample_points(q
       == 1,2)];
    if k < N
        D_sample = [D_sample; newSample];
    else
        D_sample = [D_sample; newSample];
        i = randsample(k, N);
        D_sample = D_sample(i,:);
    end
end

% plot shape and sample points
if plt == 1
    figure
```

```
    axis([shape.BoundingBox(1,1) shape.BoundingBox(1,2)
       shape.BoundingBox(2,1) shape.BoundingBox(2,2)])
    scatter(D_sample(:,1),D_sample(:,2),'.', 'r')
    hold on
    plot(shape.X,shape.Y,'black')
    title(strcat(shape.STATENAME, ' ',shape.DISTRICT, ', N
       =',num2str(N)))
    axis off
end
end

%% inshape()

function IN = inshape(x,y, shape_parts)

% Binary indicator variable for whether a given point (x,y
   ) lies in the given
% shape_parts variable. Applies built-in function
   inpolygon() over polygons
% defined by the parts of shape_parts.

IN = cellfun(@(xv) inpolygon(x,y,xv(:,1),xv(:,2)),
   shape_parts);
IN = sum(IN);
end
```

```
%% sampleBox()

function sample_points = sampleBox(N, shape)

%This function returns N points sampled uniformly at
   random from inside the
%bounding box of shape.

Xrand = shape.BoundingBox(1,1) + (shape.BoundingBox(2,1) -
    shape.BoundingBox(1,1))*rand(N,1);
Yrand = shape.BoundingBox(1,2) + (shape.BoundingBox(2,2) -
    shape.BoundingBox(1,2))*rand(N,1);
sample_points = [Xrand, Yrand];
end

function [coords, indices]  = shapeparts(shape)

% This function creates a coordinate matrix for each
   disconnected polygon
% in the given shapefile record. The output D_parts is a #
   components-by-1
% cell where each entry contains the coordinate matrix for
    that component.

%create coordinate list
```

```
D = [shape.X', shape.Y'];

%components separated by NaN in shapefile coordinate list
NaNs = isnan(D(:,1));
separators = find(NaNs == 1);
%add 1 as the first coordinate to the separator index
   vector
separators = [1 separators']';

%initialize output cells
m = length(separators);
coords = cell(m-1,1);
indices = cell(m-1,1);

%gets coordinates contained between separator indices and
   stores each
%componant in cell of 'coords'
for t = 1:(m-1);

    %indices of consecutive component separators
    i_min = separators(t);
    i_max = separators(t+1);

    %interval in list corresponding to component
    %all values between i_min and i_max
```

```
        interval = (i_min+1):1:(i_max-1);
        indices{t,1} = interval;
        coords{t,1} = D(interval,:);
    end
end
```