2016

Identifying Data Centers from Satellite Imagery

Adam Buskirk
South Dakota State University

Follow this and additional works at: http://openprairie.sdstate.edu/etd

Part of the Mathematics Commons, Spatial Science Commons, and the Statistics and Probability Commons

Recommended Citation

This Thesis - Open Access is brought to you for free and open access by Open PRAIRIE: Open Public Research Access Institutional Repository and Information Exchange. It has been accepted for inclusion in Theses and Dissertations by an authorized administrator of Open PRAIRIE: Open Public Research Access Institutional Repository and Information Exchange. For more information, please contact michael.biondo@sdstate.edu.
IDENTIFYING DATA CENTERS FROM SATELLITE IMAGERY

BY

ADAM BUSKIRK

A thesis submitted in partial fulfillment of the requirements for the

Master of Science

Major in Mathematics

Specialization in Statistics

South Dakota State University

2016
IDENTIFYING DATA CENTERS FROM SATELLITE IMAGERY

This thesis is approved as a creditable and independent investigation by a candidate for the Master of Science degree and is acceptable for meeting the thesis requirements for this degree. Acceptance of this thesis does not imply that the conclusions reached by the candidate are necessarily the conclusions of the major department.

Yunpeng Pan, Ph.D.
Thesis Advisor

Kurt Cogswell, Ph.D.
Head, Mathematics & Statistics

Dean, Graduate School

Date

Date

Date
ACKNOWLEDGMENTS

This research was supported in part by the South Dakota Board of Regents Competitive Research Grant Program FY16, and in part by the South Dakota State University Research/Scholarship Support Fund 2015. The assistance of Dr. Brian Moore of University Networking Systems & Services was indispensable in making this work possible.
CONTENTS

LIST OF SYMBOLS ......................................................... vi

LIST OF FIGURES ........................................................ viii

LIST OF TABLES .......................................................... x

ABSTRACT ................................................................. xi

1 Introduction ............................................................. 1

1.1 Landsat 8 data properties ........................................ 1

1.2 Data centers and their satellite-recognizable features ....... 2

2 DMR-family histogram descriptors ................................. 4

2.1 Measure-theoretic foundations and the continuous DMR .... 4

2.2 Applying continuous DMR-family descriptors ................ 9

2.3 Discrete DMR-family descriptors ............................... 17

2.4 Computing the DMR descriptor ................................. 19

3 Theory of the radial Fourier descriptor ........................... 25

3.1 Theory of Fourier coefficients .................................... 26

3.2 The discrete Fourier transform and fast Fourier transform algorithm ... 29

3.3 The radial Fourier descriptor .................................... 35

4 Support vector machines ............................................. 40

4.1 The Separating Hyperplane Theorem ............................ 40
<table>
<thead>
<tr>
<th>Chapter</th>
<th>Section Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.2</td>
<td>Theory of the support vector and support vector machine classifiers</td>
<td>49</td>
</tr>
<tr>
<td>4.3</td>
<td>Software for SVM: LIBSVM</td>
<td>51</td>
</tr>
<tr>
<td>5</td>
<td>Parameter selection and use of the evolutionary algorithm</td>
<td>53</td>
</tr>
<tr>
<td>5.1</td>
<td>Parameters for the differential-magnitude and radius descriptor</td>
<td>53</td>
</tr>
<tr>
<td>5.2</td>
<td>Parameters for the radial Fourier descriptor</td>
<td>54</td>
</tr>
<tr>
<td>5.3</td>
<td>Cellular evolutionary algorithms</td>
<td>55</td>
</tr>
<tr>
<td>5.4</td>
<td>Exploring the cellular evolutionary algorithm</td>
<td>61</td>
</tr>
<tr>
<td>6</td>
<td>Methodology, results and conclusions</td>
<td>70</td>
</tr>
<tr>
<td>6.1</td>
<td>Landsat data obtained</td>
<td>70</td>
</tr>
<tr>
<td>6.2</td>
<td>Hardware utilized</td>
<td>72</td>
</tr>
<tr>
<td>6.3</td>
<td>Differential-magnitude and radius descriptor results</td>
<td>72</td>
</tr>
<tr>
<td>6.4</td>
<td>Radial Fourier descriptor results</td>
<td>73</td>
</tr>
<tr>
<td>6.5</td>
<td>Conclusions</td>
<td>76</td>
</tr>
</tbody>
</table>

REFERENCES                                                                 | 80   |

INDEX                                                                       | 82   |
LIST OF SYMBOLS

\( \mathbb{n} \) The set of the first \( n \) whole numbers, \( \{1, 2, \cdots, n\} \)

\( \mathbb{R}^+ \) The positive real numbers; similarly \( \mathbb{R}^{0+} \) for the nonnegative reals, etc.

\( \langle x, y \rangle \) The inner product of two vectors in a Hilbert space

\( a \perp b \) The vector \( a \) is orthogonal to the vector \( b \); i.e. \( \langle a, b \rangle = 0 \)

\( \mathcal{N}(f) \) The null space of \( f \), which is \( f^{-1}(\{0\}) \)

\( \mathbb{T}^n \) The \( n \)-dimensional torus, \( \mathbb{T}^n \overset{\text{def}}{=} S^1 \times \cdots \times S^1 \) (\( n \) times).

\( B_\epsilon(v) \) The open ball of radius \( \epsilon \) around \( v \), \( \{x : d(x, v) < \epsilon\} \)

\( \bar{B}_\epsilon(v) \) The closed disk of radius \( \epsilon \) around \( v \), \( \{x : d(x, v) \leq \epsilon\} \)

\( \mathcal{B}(X) \) The Borel sets associated with the topological space \( X \)

\( \mu_X \) The measure associated with the measure space \( X \)

\( \lambda \) Lebesgue measure

\( d_v \) The unary distance function, with respect to the metric \( d \), \( d_v(x) \overset{\text{def}}{=} d(v, x) \)

\( D_V f \) Directional derivative of \( f : \mathbb{R}^n \to \mathbb{R} \) with respect to the vector field \( V \)

\( \Delta_V I \) Image gradient of \( I : \mathbb{Z}^n \to \mathbb{R} \) with respect to the vector field \( V \)

\( \omega_N \) The primitive \( N \)-th root of unity. When such an \( N \) is understood, \( \omega \).
\hat{f} \text{ The Fourier coefficients of } f : \mathbb{T} \to \mathbb{C}, \text{ the Fourier transform of } f : \mathbb{R} \to \mathbb{C}, \text{ or}

\text{the discrete Fourier transform of } f : \mathbb{Z}/n\mathbb{Z} \to \mathbb{C}.

N(v) \text{ The neighborhood of the vertex } v \text{ in a graph: } \{v\} \cup \{v' \in G_v : (v, v') \in G_v\}
# LIST OF FIGURES

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.1</td>
<td>The plot of $f(x, y) = x^3 - y^2$.</td>
<td>9</td>
</tr>
<tr>
<td>2.2</td>
<td>The plot of $D_t f(x, y)$.</td>
<td>10</td>
</tr>
<tr>
<td>2.3</td>
<td>DMR-induced regions for $f$.</td>
<td>11</td>
</tr>
<tr>
<td>2.4</td>
<td>The plot of $f(x) = x^{-1} \cos x^{-1}$.</td>
<td>14</td>
</tr>
<tr>
<td>2.5</td>
<td>The plot of $f^4(x)$.</td>
<td>14</td>
</tr>
<tr>
<td>2.6</td>
<td>Attracting period-4 orbits of $f^4$.</td>
<td>15</td>
</tr>
<tr>
<td>2.7</td>
<td>Pre-MR counting matrix for the points 0, 1/16, 1/8, and 1/4, respectively.</td>
<td>18</td>
</tr>
<tr>
<td>2.8</td>
<td>A general sketch of the DMR algorithm.</td>
<td>21</td>
</tr>
<tr>
<td>3.1</td>
<td>Translation and rotation invariance of representative functions</td>
<td>35</td>
</tr>
<tr>
<td>3.2</td>
<td>Sinusoidal reconstruction of data center structure</td>
<td>37</td>
</tr>
<tr>
<td>4.1</td>
<td>Diagram of Separating Hyperplane Theorem</td>
<td>45</td>
</tr>
<tr>
<td>4.2</td>
<td>The three possibilities for the altitude from $x$ to $y_R$.</td>
<td>46</td>
</tr>
<tr>
<td>5.1</td>
<td>Evolutionary improvement probabilities and variance</td>
<td>59</td>
</tr>
<tr>
<td>5.2</td>
<td>Cellular evolutionary algorithm test, semiseparable</td>
<td>63</td>
</tr>
<tr>
<td>5.3</td>
<td>The effect of initial outliers on cEA behavior</td>
<td>65</td>
</tr>
<tr>
<td>5.4</td>
<td>Cellular evolutionary algorithm test, mixed</td>
<td>67</td>
</tr>
<tr>
<td>5.5</td>
<td>Maximum fitness by cell parameters</td>
<td>68</td>
</tr>
<tr>
<td>6.1</td>
<td>True positive and negative rates for DMR configurations</td>
<td>73</td>
</tr>
<tr>
<td>6.2</td>
<td>True positive and negative rates for individual RFD sampling configurations</td>
<td>75</td>
</tr>
</tbody>
</table>
LIST OF TABLES

1.1 Landsat 8 product band properties ........................................ 2

2.1 $f$’s attractor’s Lyapunov exponents ...................................... 16

2.2 Computational costs associated with DMR .............................. 24

3.1 Add. and mult. counts after FFT .......................................... 32

3.2 Add. and mult. counts after iterated FFT ............................... 34

5.1 Fitness convergence for $D_{26}$, $D_{108}$, $D_{26}^{*}$, and $D_{108}^{*}$ .......... 64

6.1 Landsat training & evolutionary algorithm scene IDs ................. 71
ABSTRACT

IDENTIFYING DATA CENTERS FROM SATELLITE IMAGERY

ADAM BUSKIRK

2016

We develop two different descriptors which can be utilized to describe satellite imagery. The first, the differential-magnitude and radius descriptor, describes a scene by computing the directional gradient of the scene with respect to a vector field whose solutions are circles around a pixel to be described, and then counts pixels in a descriptor matrix according to the magnitude of this gradient and the distance at which this magnitude occurs. The second, the radial Fourier descriptor, extracts from the scene a sequence of annuloid sectors, and uses this to approximate the behavior of the image on a circle around the point to be described. The fast Fourier transform is then used to obtain a description of this function in the frequency domain; the absolute values of these complex-valued frequencies form the descriptor. A set of data to test and perform parameter selection for these procedures using 79 Landsat 8 imagery scenes was constructed. A cellular evolutionary algorithm was then utilized for parameter selection by training and testing support vector machine classifiers using LIBSVM from this dataset utilizing classification accuracy as a fitness function. We then analyze the classification success associated with the two methods equipped with their optimized parameters.
Chapter 1

Introduction

The problem this work labors to solve is this: given a scene taken from satellite imagery, how do we automate the procedure of structure recognition? For this problem, we specifically focus on using the freely accessible Landsat 8 imagery \[17\], and the task of recognizing large data centers such as those which power the internet titans of today.

1.1 Landsat 8 data properties

Landsat 8 data is delivered in the GeoTIFF data format, consisting of eleven bands of data as described in Table 1.1. The first seven bands and band 9 effectively span wavelengths from $0.43 \, \mu m$ to $2.29 \, \mu m$, with band 9 falling between bands 5 and 6. Band 8 is a higher-resolution band which can be used with bands 2, 3, and 4 to form an approximation of a standard RGB image with 15 meter resolution; however, the “panchromatic” wavelengths are mostly red and green. Bands 10 and 11 are thermal bands with high wavelengths. Each band is named in the form “[Landsat scene ID]_B[Band number].TIF,” provided as a grayscale TIFF image file, which may be opened by ordinary image editing software like the GNU Image Manipulation Program \[16\], but also equipped with additional metadata regarding the specifics of the projection used and corrections done to the data.

Furthermore, included is a metadata file which contains information about the properties of the image, including the properties of the projection, the longitude and latitude of the four corners of the image, image attributes like cloud cover, and, of course,
<table>
<thead>
<tr>
<th>Band</th>
<th>Description</th>
<th>Wavelength (µm)</th>
<th>Resolution (m)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Coastal aerosol</td>
<td>0.43-0.45</td>
<td>30</td>
</tr>
<tr>
<td>2</td>
<td>Blue</td>
<td>0.45-0.51</td>
<td>30</td>
</tr>
<tr>
<td>3</td>
<td>Green</td>
<td>0.53-0.59</td>
<td>30</td>
</tr>
<tr>
<td>4</td>
<td>Red</td>
<td>0.64-0.67</td>
<td>30</td>
</tr>
<tr>
<td>5</td>
<td>Near Infrared (NIR)</td>
<td>0.85-0.88</td>
<td>30</td>
</tr>
<tr>
<td>6</td>
<td>SWIR 1</td>
<td>1.57-1.65</td>
<td>30</td>
</tr>
<tr>
<td>7</td>
<td>SWIR 2</td>
<td>2.11-2.29</td>
<td>30</td>
</tr>
<tr>
<td>8</td>
<td>Panchromatic</td>
<td>0.50-0.68</td>
<td>15</td>
</tr>
<tr>
<td>9</td>
<td>Cirrus</td>
<td>1.36-1.38</td>
<td>30</td>
</tr>
<tr>
<td>10</td>
<td>Thermal Infrared (TIRS) 1</td>
<td>10.60-11.19</td>
<td>100</td>
</tr>
<tr>
<td>11</td>
<td>Thermal Infrared (TIRS) 2</td>
<td>11.50-12.51</td>
<td>100</td>
</tr>
</tbody>
</table>

Table 1.1: Description of the Landsat 8 data bands, replicated from United States Geological Survey [17, http://landsat.usgs.gov/band_designations_landsat_satellites.php]

information about exactly when the image was taken.

1.2 Data centers and their satellite-recognizable features

Data center structures frequently exhibit a number of large-scale characteristics which aid in their identification. Data center buildings are often large buildings, relatively broad and with relatively few floors; furthermore, they are often longer structures, lined with cooling towers on one side and vehicular access on the other. Often data centers will also be associated with their own power stations; the long roofs of some are lined with solar panels, or smaller cooling towers, or other auxiliary infrastructure for the servers. While some of the finer details may not be visible in the 30 meter resolution of Landsat 8 imagery, the heat emitted by the cooling towers will often be an identifiable signature for the structure. This gives us a list of features we may utilize to tell whether a given scene in Landsat imagery represents a data center:
• It will almost certainly possess the uniquely blueish colors peculiar to human-made structures, in contrast to the earthen colors of forest and field.

• It will likely have, somewhere in the vicinity, the sharp changes across multiple bands formed by the edges of the building; this edge may also have a rough texture along it, formed by the regular placement of the building’s cooling towers, air intake fans, and other support structures for the servers.

• It is likely to be relatively warm, or to be near to hot spots in the image.

This list of properties, one is drawn to notice, is largely localized, rather than a property associated with a single pixel. Indeed, it would seem rather absurd to think a single spectrum at a point, a simple tuple of 11 rational numbers, would reveal all of a structure’s secrets; and this draws us towards descriptors at a pixel as a function of the surrounding neighborhood.
Chapter 2

DMR-family histogram descriptors

The first type of descriptor we developed based, in large part, upon the concepts utilized by Dalal and Triggs [7], we call the “differential-magnitude and radius” descriptor; more generally, there is a family of similar descriptors utilizing the concept which we refer to as “DMR-family” (Histogram) descriptors. The general concept is that we can describe a scene in an image using the gradient, taken as a radial derivative around the point to be described. The pixels may then be classified by such statistics as directional gradient and distance from the center, and the quantity of pixels lying in each class can prove to be valuable in determining what the scene depicts.

2.1 Measure-theoretic foundations and the continuous DMR

While the problem we approach is naturally discretized, for the purposes of understanding, we may define an abstraction of the (DMR) histogram descriptor which may be applied generally to describe both the discrete case and a continuous interpretation which may assist in understanding the DMR histogram idea.

Our development of the basic terminology of measure theory, which provides the theoretical basis for both continuous and discrete DMR, is based on Ash and Doleans-Dade [2], and that text is recommended for a more complete description of these ideas.

**Definition 2.1.1.** If \( \mathcal{F} \) is a collection of subsets of some set \( \Omega \) which satisfies the four properties below, we call \( \mathcal{F} \) a \( \sigma \)-field.

\( (A) \ \Omega \in \mathcal{F} \)
(B) If $A \in \mathcal{F}$, then $A^C \in \mathcal{F}$. (Closure under complements)

(C) If $A_1, A_2, \cdots, A_n \in \mathcal{F}$, then $\bigcap_{i=1}^{n} A_i \in \mathcal{F}$. (Closure under finite intersections)

(D) If $\{A_i : i \in \mathbb{N}\} \subseteq \mathcal{F}$, then $\bigcap_{i=1}^{\infty} A_i \in \mathcal{F}$. (Closure under countable intersections)

If $\mathcal{F}_0$ satisfies only properties (A), (B), and (C), then it is called a field. If $\mathcal{L}$ is a collection of subsets of $\Omega$, define $\sigma(\mathcal{L})$ to be

$$
\sigma(\mathcal{L}) \overset{\text{def}}{=} \bigcap_{s \in \Sigma(\mathcal{L})} s,
\Sigma(\mathcal{L}) = \{s : \mathcal{L} \subseteq s, s \text{ a } \sigma\text{-field}\},
$$

the minimal $\sigma$-field which encompasses all of the sets in $\mathcal{L}$.

Furthermore, a space equipped with a $\sigma$-field $(X, \mathcal{F})$ is called a measurable space.

**Definition 2.1.2.** The Borel sets of a topological space $(X, \tau_X)$, denoted $\mathcal{B}(X)$, is defined to be the smallest $\sigma$-field containing the topology of the space; that is,

$$
\mathcal{B}(X) \overset{\text{def}}{=} \sigma(\tau_X)
$$

**Definition 2.1.3.** We call $f : X \to Y$ a measurable function if $X$ and $Y$ are measurable spaces and whenever $V \in \mathcal{F}_Y$, then $f^{-1}(V) \in \mathcal{F}_X$. That is, the preimage under $f$ of every measurable set in $Y$ is measurable in $X$.

**Definition 2.1.4.** A measure on a measurable space $(X, \mathcal{F})$ is a set function $\mu : \mathcal{F} \to \mathbb{R}^{0+}$ such that if $\{A_i, i \in \mathbb{N}\}$ is a collection of mutually disjoint elements of $\mathcal{F}$, then

$$
\mu \left( \bigcup_{i=1}^{\infty} A_i \right) = \sum_{i=1}^{\infty} \mu(A_i).
$$
**Theorem 2.1.5** (Carathéodory Extension Theorem). If \( \mu \) is a measure on a field \( \mathcal{F}_0 \) of subsets of \( \Omega \), and there exists some countable measurable cover of \( \Omega \), \( \{ A_i : i \in \mathbb{N} \} \), such that \( \mu(A_i) < \infty \) for all \( i \), then \( \mu \) has a unique extension to a measure on \( \sigma(\mathcal{F}_0) \), the minimal \( \sigma \)-field over \( \mathcal{F}_0 \).

For a proof of this theorem, see Ash and Doleans-Dade [2, p. 19].

Using this, we find that we can extend the pre-measure \( \mu' \) on the field generated by right semi-closed intervals (RSCI, intervals of the form \((-\infty, b], (a, b], (a, \infty) \) or \( \mathbb{R} \) itself), which is defined by decomposing each set in the field into a finite collection of disjoint RSCI intervals and summing

\[
\mu' : \begin{cases} 
(a, b] & \mapsto b - a \\
(-\infty, b] & \mapsto \infty \\
(a, \infty) & \mapsto \infty \\
\mathbb{R} & \mapsto \infty 
\end{cases}
\]

over this collection. Using the Carathéodory extension theorem gives us a measure \( \lambda \), called the **Borel measure** on the real line, which is defined on the \( \sigma \)-field generated by the RSCI intervals. As its name would suggest, this turns out to be the same as the Borel sets for \( \mathbb{R} \), since we may generate any RSCI interval from open sets (let \( b_n \downarrow b \); then \( (a, b] = \bigcap_n (a, b_n) \); the other types of RSCI intervals yield similarly) and an open interval may be generated from RSCI intervals (let \( a < b_n \uparrow b \); then \( (a, b) = \bigcup_n (a, b_n) \)); and any open subset of \( \mathbb{R} \) is a countable union of these intervals; so any \( \sigma \)-field containing one must also contain the other and all sets generated by the other, and thus the minimal \( \sigma \)-field for the RSCI sets is equivalent to the minimal \( \sigma \)-field generated by the open sets, \( \mathcal{B}(\mathbb{R}) \).

---

1Note that the definition of a \( \sigma \)-field does imply closure under countable unions; \( \forall i \in \mathbb{N}, A_i^c \in \mathcal{F} \Rightarrow_{(\mathcal{D})} \bigcap_n A_i^c \in \mathcal{F} \Rightarrow_{(\mathcal{B})} (\bigcap_n A_i^c)^c = \bigcup_n A \in \mathcal{F} \).
**Definition 2.1.6.** Given a measurable space \((X, \mathcal{F}_X)\) and a collection of measurable functions \(\gamma_i : X \to (Y_i, \mathcal{F}_{Y_i})\), \(i \in \bar{n}\), we may specify measurable sets in \(\mathcal{F}_X\) by their images in each \(Y_i\), using the function \(\Gamma : \mathcal{F}_{Y_1} \times \cdots \times \mathcal{F}_{Y_n} \to \mathcal{F}_X\) defined by

\[
\Gamma(b_1, b_2, \cdots, b_n) \overset{\text{def}}{=} \bigcap_{i \in \bar{n}} \gamma_i^{-1}(b_i), \quad b_i \in \mathcal{F}_{Y_i}. \tag{2.1}
\]

We call \(\Gamma\) the \((\gamma_1, \cdots, \gamma_n)\)-specification for \(X\) or, in general, a Gamma specification for \(X\). Furthermore, if we are given \(G_i \subseteq \mathcal{F}_i\), nonempty finite collections of measurable sets, we call \((G_1, G_2, \cdots, G_n)\) a bin configuration for \(\Gamma\).

A gamma specification for a space is essentially an alternative way of constructing some subcollection of measurable sets in the space, sets which may otherwise be rather difficult to denote.

**Example 2.1.7.** Let \(X = \mathbb{R}^2\) with the Borel sets, and consider \(\gamma_1\) and \(\gamma_2\) to be the projections of \(X\) onto the first and second components respectively, \(\pi_1, \pi_2 : \mathbb{R}^2 \to \mathbb{R}\). Then if \(b_1, b_2 \in \mathcal{B}(\mathbb{R})\), and with \(\Gamma\) defined by (2.1) above, \(\Gamma(b_1, b_2) = b_1 \times b_2\) specifies the measurable rectangle [2, p. 113] formed by \(b_1\) and \(b_2\) in \(\mathbb{R}^2\).

**Example 2.1.8.** Suppose \(X = \mathbb{R}^2\) and let \(\gamma_1 : \mathbb{R}^2 \to \mathbb{R}^{0+}\) and \(\gamma_2 : \mathbb{R}^2 \to S^1\) map each point to the \(r\) and \(\theta\) polar coordinates of that point respectively. Then \(\Gamma(b_1, b_2)\) defined with respect to these specifies bands in \(\mathbb{R}^2\) around \(0\) with distances in \(b_1\) and angles in \(b_2\); for example, \(\Gamma([1, 2], S^1)\) is the annulus around \(0\) with inner radius \(1\) and outer radius \(2\).

**Definition 2.1.9.** Consider some measurable function \(f : X \to \mathbb{R}\), with \(X\) a measure space with Borel sets derived from a metric \(d\), and a point \(v\). Let \(\gamma_1 = f\) and \(\gamma_2 = d_v : \)
$X \to \mathbb{R}^{0+}$, where $d_v(x) \overset{\text{def}}{=} d(v, x)$, and let $\Gamma$ be the $(\gamma_1, \gamma_2)$-specification for $X$. Then the \textbf{general Magnitude and Radius (MR) descriptor} with respect to $m \in \mathcal{B}(X)$ and $r \in \mathcal{B}(\mathbb{R}^{0+})$

$$\text{MR}(v; m, r) \overset{\text{def}}{=} \mu_X(\Gamma(m, r))$$

and with respect to a bin configuration $(M, R)$ for $\Gamma$, we define also the magnitude and radius descriptor with respect to $(M, R) = (\{m_i : i \in \bar{a}\}, \{r_j : j \in \bar{b}\})$ to be the matrix defined by

$$[\text{MR}(v; M, R)]_{ij} \overset{\text{def}}{=} \mu_X(\Gamma(m_i, r_j))$$

\textbf{Definition 2.1.10.} Consider some smooth function $f$ and a point $v$. Let

$$t_v(x) \overset{\text{def}}{=} \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix} \frac{x - v}{\|x - v\|}$$

be the vector field populated with unit vectors orthogonal to $x - v$, and let $D_{t_v}$ denote the directional derivative with respect to $t_v$. Let then $\gamma_1 = D_{t_v}f : \mathbb{R}^2 \setminus \{v\} \to \mathbb{R}$ and $\gamma_2 = d_v : \mathbb{R}^2 \setminus \{v\} \to [0, \infty)$ be the unary distance function defined above, and define $\Gamma$ as above by these. We define the \textbf{continuous Differential-Magnitude and Radius (DMR) descriptor} of $f$ at $v$ with respect to $(m, r) \in \mathcal{B}(\mathbb{R}) \times \mathcal{B}([0, \infty))$ by

$$\text{DMR}(v; m, r) \overset{\text{def}}{=} \lambda(\Gamma(m, r))$$

where $\lambda$ denotes the Lebesgue measure on $\mathbb{R}^2 \setminus \{v\}$. If $(M, R)$ is a bin configuration for $\Gamma$, where $M = \{m_i : i \in \bar{a}\} \subseteq \mathcal{B}(\mathbb{R})$ and $R = \{r_j : j \in \bar{b}\} \subseteq \mathcal{B}(\mathbb{R}^{0+})$, then we define the DMR descriptor with respect to the bin configuration $(M, R)$ as the matrix $\text{DMR}(v; M, R)$ defined by

$$[\text{DMR}(v; M, R)]_{ij} \overset{\text{def}}{=} \text{DMR}(v; m_i, r_j) = \lambda(\Gamma(m_i, r_j)).$$
Figure 2.1: The plot of $f(x, y) = x^3 - y^2$.

Intuitively, the DMR descriptor at a point is a histogram of the different types of behavior a path encircling that point would encounter, for bands of various radii.

2.2 Applying continuous DMR-family descriptors

Example 2.2.1. As an example of DMR, consider the function $f(x, y) = x^3 - y^2$ (Figure 2.1). To obtain the directional derivative of $f$, first we compute the gradient of $f$, which is $\nabla f(x, y) = (3x^2, -2y)^\top$, and the vector field $t_0$ is defined by

$$
  t_0(x, y) = \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix} \frac{(x, y)^\top - (0, 0)^\top}{\|(x, y)^\top - (0, 0)^\top\|} = \frac{(-y, x)^\top}{\sqrt{x^2 + y^2}}
$$

We compute the directional derivative of $f$ with respect to $t_0$ utilizing the dot product of the gradient of $f$ and $t_0$:

$$
  D_{t_0}f(x, y) = \nabla f \cdot t_0 = \frac{-3x^2y - 2xy}{\sqrt{x^2 + y^2}}
$$
Figure 2.2: The plot of $D_{t_0}f(x,y)$.

which is displayed in Figure 2.2.

Then, suppose we wish to determine the DMR for $f$ associated with the bin configuration $(M,R) = ((-\infty,0],(0,\infty)),([0,0.5],[0.5,1])$. A rough designation of the regions described is shown in Figure 2.3. It is evident from the formulation above that this is a continuous function (except, of course, at the point $(0,0)$, where it is undefined, but this is technically omitted from the domain of $D_{t_0}f$). Hence the boundaries of these regions describe their respective contents.

To find these boundaries, we consider the set of points where $D_{t_0}f(x,y) = 0$.

$$D_{t_0}f(x,y) = 0$$

$$\frac{-3x^2y - 2xy}{\sqrt{x^2 + y^2}} = 0$$
Figure 2.3: The regions of $D_{t_0}f$ designated by the bin configuration ($\{(-\infty, 0], (0, \infty)\}, \{[0, 0.5], (0.5, 1]\}$). The regions denoted by the (1,1), (2,1), (1,2), and (2,2) entries in the DMR matrix are blue, black, pink, and red, respectively.
\[-3x^2y - 2xy = 0\]

\[-3x^2y = 2xy\]

This equation implies one of three scenarios: \(x = 0, y = 0\), or, dividing through by \(-3xy\), \(x = -\frac{2}{3}\).

Note that the magnitudinal boundaries for the disk defined by the radial bin \([0, 0.5]\) are only those associated with \(x = 0\) and \(y = 0\). Hence the boundaries of the \((1, 1)\) and \((2, 1)\) entries in the DMR matrix are just the \(x\) and \(y\) axes and the circle of radius 0.5. That is,

\[\Gamma((\infty, 0], [0, 0.5]) = \{(x, y) \in \mathbb{R}^2 \setminus \{0\} : xy \geq 0, \ 0 < ||(x, y)|| \leq 0.5\}\]

\[\Gamma((0, \infty), [0, 0.5]) = \{(x, y) \in \mathbb{R}^2 \setminus \{0\} : xy < 0, \ 0 < ||(x, y)|| \leq 0.5\}\]

Each of which are two opposite quarter-circles of radius 0.5, and consequently

\[[\text{DMR}(0; M, R)]_{1,1} = [\text{DMR}(0; M, R)]_{2,1} = \frac{0.5^2\pi}{2} = \frac{\pi}{8}\]

For the radial bin \((0.5, 1]\), we compute only the DMR for \((-\infty, 0]\) and \((0.5, 1]\), which will have the same measure as that for \((0, \infty)\) and \((0.5, 1]\), since latter is a full-measure subset (excluding the boundary) of the former’s image under the linear with determinant 1 and thus measure-preserving map \(g(x, y) = (-x, y)\). They are complementary in the annulus with inner radius 0.5 and outer radius 1, so they must each be half of \(\pi - 0.5^2\pi = 0.75\pi\), and thus

\[[\text{DMR}(0; M, R)]_{1,2} = [\text{DMR}(0; M, R)]_{2,2} = \frac{0.5^2\pi}{2} = \frac{3\pi}{8}\]
Consolidating these results, we find that the DMR matrix for 0 with \((M, R)\), \(M = \{(-\infty, 0], (0, \infty)\}, R = \{[0, 0.5], (0.5, 1]\}, f is

\[
\text{DMR}(0; M, R) = \left\{ \frac{3\pi}{8}, \frac{3\pi}{8}, \frac{3\pi}{8}, \frac{3\pi}{8} \right\}
\]

We now continue to our second example, which applies the concepts of the MR descriptor to the field of dynamical systems.

**Example 2.2.2.** Consider the dynamical system defined by \(f(x) = x^{-1} \cos x^{-1}\) (Figure 2.4) almost everywhere on \(\mathbb{R}\) (excluding the countable collection of points which eventually map to 0). This system exhibits complicated dynamics; indeed, a plot of \(f^4\) (Figure 2.5) demonstrates that \(f\) actually has attracting period-4 orbits; and furthermore, it actually appears to have countably many of these, as shown in Figure 2.6. The concept of the Lyapunov exponent can be used to identify and distinguish between orbits which are attracted to distinct periodic orbits.

**Definition 2.2.3.** If \((x_n)\) is an orbit in a dynamical system \((\mathbb{R}, f)\) where \(f\) is smooth, then we define the **Lyapunov exponent** \(h(x_1)\) to be

\[
h(x_1) = \lim_{n \to \infty} \frac{1}{n} \sum_{i=1}^{n} \ln |f'(x_i)|
\]

if this limit exists. [1, p. 107]

It does, indeed, seem that the Lyapunov exponent is defined across \(f\), and we can (with an implicit assumption that the Lyapunov exponent \(h\) is a measurable function) treat the Lyapunov exponent as a function of study for use in the MR descriptor. A scan across \(x\) in the interval \([-1, 1]\) with 1000 points discovers a collection of 16 distinct
Figure 2.4: The plot of $f(x) = x^{-1} \cos x^{-1}$.

Figure 2.5: The plot of $f^4(x)$. 
periodic orbits which the sampled points collect at. These Lyapunov exponents are listed in table 2.1. From these exponents, we can define a partition of $\mathbb{R}$ into measurable sets $M$, which we can use as magnitudinal bins in which a point’s Lyapunov exponent may lie. Similarly we define five radial bins $R = \{(2^{-7}, 2^{-6}], (2^{-8}, 2^{-7}], (2^{-9}, 2^{-8}], (2^{-10}, 2^{-9}], [0, 2^{-10}]\}$, which will serve to partition in a $\Gamma$ specification, around a point of study $v$, the closed disk around it of radius $2^{-6}$, $\bar{B}_{2^{-6}}(v) = [v - 2^{-6}, v + 2^{-6}]$. We can then utilize Monte Carlo integration techniques to approximate the area of the different regions within a neighborhood of a point.

MR by Monte Carlo integration pseudocode

1. Let $v$ denote the point to be examined, and let $n$ be the number of samples desired.
Lyapunov exponent | magnitudinal bin | measure of bin
--- | --- | ---
-4.36315588781278 | $(-\infty, -3.9545\ldots]$ | $\infty$
-3.54588205355212 | $(-3.9545\ldots, -3.486\ldots]$ | 0.4678\ldots
-3.4274908129934 | $(-3.486\ldots, -3.3083\ldots]$ | 0.1783\ldots
-3.18923049705521 | $(-3.3083\ldots, -3.1122\ldots]$ | 0.1960\ldots
-3.03529219823209 | $(-3.1122\ldots, -3.0296\ldots]$ | 0.0826\ldots
-3.02391336257659 | $(-3.0296\ldots, -2.9386\ldots]$ | 0.0909\ldots
-2.85346818410071 | $(-2.9386\ldots, -2.8481\ldots]$ | 0.0905\ldots
-2.84280830011142 | $(-2.8481\ldots, -2.7371\ldots]$ | 0.1110\ldots
-2.63146109520362 | $(-2.7371\ldots, -2.6265\ldots]$ | 0.1105\ldots
-2.621676976804 | $(-2.6265\ldots, -2.4841\ldots]$ | 0.1424\ldots
-2.3466947588129 | $(-2.4841\ldots, -2.3422\ldots]$ | 0.1418\ldots
-2.3379374657944 | $(-2.3422\ldots, -2.1439\ldots]$ | 0.1983\ldots
-1.94995391253300 | $(-2.1439\ldots, -1.9463\ldots]$ | 0.1975\ldots
-1.94280170066492 | $(-1.9463\ldots, -1.6239\ldots]$ | 0.3224\ldots
-1.3058697283135 | $(-1.6239\ldots, -1.3026\ldots]$ | 0.3213\ldots
-1.30019669187372 | $(-1.3026\ldots, \infty)$ | $\infty$

Table 2.1: Approximate Lyapunov exponents of attractors discovered in a scan of $f$ over $[-1,1]$ and magnitudinal bins based on them.

Let $M$ and $R$ be the magnitudinal and radial bin collections specified in the example, respectively.

HistogramMatrix ← zeros($|M|, |R|$)

For $i ← 1\ldots n$,

Let $x$ be a random value sampled from the uniform distribution on the interval $[v - 2^{-6}, v + 2^{-6}] = \bigcup_{m \in M} m$.

For $a ← 1\ldots |M|$,

for $b ← 1\ldots |R|$,

if $h(x) \in m_a$ and $d(x,v) \in r_b$, then

HistogramMatrix(a,b) ← HistogramMatrix(a,b) + 1

return HistogramMatrix * $\lambda(\bigcup_{m \in M} m) / n$

The matrix returned by this code will, ignoring machine error and for large values of $n$, be a reasonable approximation of the MR matrix for $f$ at $v$ with respect to $M$ and
Figure 2.7 demonstrates a HistogramMatrix values generated by this code run (in Octave) at 0, 1/16, 1/8, and 1/4, with \( n = 10,000 \) points. (This data is more human-interpretable than the MR itself.) This data demonstrates a few properties of the points analyzed. It is evident that the last Lyapunov exponents is relatively common, as it is consistently the largest entry in each column. However, at 1/16 the proportion of orbits with a Lyapunov exponent of approximately \(-2.84280830011142\) is substantially larger within the disk around \(2^{-9}\). Additionally, points near 1/4 seem to be entirely attracted to the orbit(s) represented by \(-1.30019669187372\). It seems clear from this data that the neighborhoods around these four points have drastically different properties.

2.3 Discrete DMR-family descriptors

While the continuous version of the DMR histogram descriptor is useful for understanding the concepts underlying the computations, the Landsat data we have is by nature discretized. While the MR descriptor itself can be utilized on functions over any measure space with minimal modification, we must define a substitute for the directional derivative expected by the DMR.

Suppose we are given a monochromatic image file, like those utilized in Landsat data. Then our image can be viewed as a map \( I : \bar{m} \times \bar{n} \rightarrow \mathbb{Z} \), which can in turn be extended to a map \( \mathcal{I} : \mathbb{Z} \times \mathbb{Z} \rightarrow \mathbb{R} \),

\[
\mathcal{I}(x_1, x_2) = \begin{cases} 
I(x_1, x_2) & \text{if } x_1 \in \bar{m}, x_2 \in \bar{n} \\
0 & \text{otherwise}
\end{cases}
\]

**Definition 2.3.1.** Let \( \mathcal{I} : \mathbb{Z} \times \mathbb{Z} \rightarrow \mathbb{R} \) and let \( \mathbb{Z} \times \mathbb{Z} \) be equipped with the counting
Figure 2.7: Pre-MR counting matrix for the points 0, 1/16, 1/8, and 1/4, respectively.
measure $C$. We define the **discrete Magnitude and Radius descriptor** to be

$$\text{MR}(v; m, r) \overset{\text{def}}{=} C(\Gamma(m, r))$$

where $\Gamma$ is the $\Gamma$-specification associated with $\gamma_1 = I$ and $\gamma_2 = d_v$, with $m$ from $B(\mathbb{R})$ and $r$ from $B(\mathbb{R}^{0+})$.

For any nonzero $v \in \mathbb{R}^2$, let $\Delta_v I : \mathbb{Z}^2 \to \mathbb{R}$ denote the image gradient of $I$ in the direction of $v$; and if $V : \mathbb{Z}^2 \to \mathbb{R}^2$ is a vector field, let $\Delta_V I : \mathbb{Z}^2 \to \mathbb{R}$ be defined by

$$[\Delta_V I](x) \overset{\text{def}}{=} [\Delta_{V(x)} I](x)$$

**Definition 2.3.2.** Let $I : \mathbb{Z} \times \mathbb{Z} \to \mathbb{R}$, and let $\mathbb{Z} \times \mathbb{Z}$ be equipped with the counting measure $C$. The **discrete Differential-Magnitude and Radius descriptor** of a point $v \in \mathbb{Z} \times \mathbb{Z}$ is determined by the $\Gamma$-specification for $\gamma_1 = \Delta_V I$ and $\gamma_2 = d_v$, with respect to $m \in B(\mathbb{R})$ and $r \in B(\mathbb{R}^{0+})$,

$$\text{DMR}(v; m, r) \overset{\text{def}}{=} C(\Gamma(m, r))$$

where $V : \mathbb{Z}^2 \to \mathbb{R}^2$ is the tangent unit vector field defined by

$$V(x_1, x_2) \overset{\text{def}}{=} \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix} \frac{x - v}{\|x - v\|}$$

Similarly, we define the discrete DMR descriptor with respect to $(M, R)$, where $M = \{m_i : i \in \bar{a}\}$ and $R = \{r_j : j \in \bar{b}\}$, as the $a \times b$ matrix defined by

$$[\text{DMR}(v; M, R)]_{ij} \overset{\text{def}}{=} \text{DMR}(v; m_i, r_j)$$

2.4 Computing the DMR descriptor

In practice, computing the discrete DMR descriptor can be optimized dramatically using efficient matrix operations. Our procedure exploits the fact that the DMR
computations can be readily translated, and thus may be computed en masse using translation operations. The general procedure we utilize follows the outline below.

We note a few facts about the efficiency of this procedure. There are three types of operations we consider “costly” in different proportions, which are comparisons (<, >, ≤, and ≥), additions (and subtractions), and multiplications. We will denote the cost of an operation requiring $c$ comparisons, $a$ additions, and $m$ multiplications with the 3-tuple $[c, a, m]$.

First note that the number of nonzero vectors $v$ in consideration will be approximately $\pi R^2 - 1$ (Figure 2.8 line 5; Table 2.2, vec), where $R$ is the supremum of the union of $R$ (if this does not exist then this computation is entirely untenable). For each of these vectors, we must compute a gradient of the image. Utilizing a convolution procedure to compute the gradient, with approximately 7 nonzero entries, this will cost approximately seven multiplications and six additions per pixel in the image (6-7, grd). Checking whether a given pixel in a gradient image is within a given bin will take 4 comparisons, if we assume $M$ and $R$ are both right semi-closed intervals—and for the sake of simplicity, we require this in practice (9-10, bnc). To compute this for each combination of bins will multiply this value by $|M| \cdot |R|$, which can be combined into the three-dimensional array $G^v$ in the algorithm above (8, $[G^v]$). To add up the $G^v$ matrices for each $v$ will require approximately $\pi R - 2$ operations at each pixel, one less than the approximate number of vectors under consideration (12, $|G|$).

The full procedure entails first computing a gradient for each vector (vec · grd), then computing $G^v$ for each of these gradients (vec · $[G^v]$) and then combining these all
General DMR Procedure

1. We are given an $X \times Y$ Landsat scene $I$ and bin configurations $(M_I, R_I)$, $I \in I$ associated with each layer.

2. For each Landsat layer $I \in I$,

3. Find the maximum radius under consideration, $\mathcal{R} = \sup \bigcup_{r \in R_I} r$.

4. For each integer vector $v = (x, y)^\top \in B_{\mathcal{R}}(0) \cap \mathbb{Z}^2$,

5. Compute the gradient of the image, $G^v = \Delta_v I$ for $v = \left(\begin{smallmatrix} 0 \\ -1 \\ 1 \\ 0 \end{smallmatrix}\right)$, in the direction orthogonal to $(x, y)^\top$.

6. For each $m_i \in M_I$ and $r_j \in R_I$,

7. Compute $G^v_{ij} : X \times Y \rightarrow \{0, 1\}$ such that $[G^v_{ij}] (w, z) = 1$ if $G^v_{wz} \in m_i$ and $|(w, z)| \in r_j$, and $[G^v_{ij}] (w, z) = 0$ otherwise.

8. Let $G^v : X \times Y \times |M_I| \times |R_I| \rightarrow \mathbb{Z}$ be defined by $G^v (w, z, i, j) = G^v_{ij} (w, z)$.

9. Let $G_I = \sum_v G^v$. We let the $|M_I| \times |R_I|$ matrix $\text{DMR}_I (w, z; M, R) = G(w, z, \cdot, \cdot)$ describe the point $(w, z)$ on layer $I$.

10. Let $c_I : |M_I| \times |R_I| \rightarrow |M_I| \times |R_I|$ be a bijective counting function

11. for each $I$.

12. At each layer, we reshape $G_I$ into a three-dimensional matrix

13. $\mathcal{G}_I : X \times Y \times |M_I| \times |R_I| \rightarrow \mathbb{Z}$ defined by $\mathcal{G}_I (w, z, k) = G_I (w, z, [c_I(k)]_1, [c_I(k)]_2)$.

14. We then append all of these $\mathcal{G}_I$ along the third index into a single three-dimensional matrix $\mathcal{G}$ which serves to describe our scene at every layer simultaneously.

Figure 2.8: A general sketch of the DMR algorithm.
into a single $G$ (which just costs $\lfloor G \rfloor$). If we let $P = \text{pix}$ be the total number of pixels in the image, $V = |B_R(0) \cap \mathbb{Z}^2|$ be the number of vectors to consider, and let $A = |M|$, $B = |R|$, then the total cost can be expressed as

$$V \cdot \text{grd} + V \cdot [G^v] + [G] = VP[0, 6, 7] + VABP[4, 0, 0] + P[0, V - 1, 0]$$

$$= [4VABP, 6PV + P(V - 1), 7PV]$$

$$= [4VABP, 7PV - P, 7PV]$$

By expression 2.2, we would thus expect a total of $4VABP$ comparisons, $7PV - P$ additions, and $7PV$ multiplications required to compute the DMR for a scene. To put these in terms of the actual data, with a $6000 \times 6000$ scene, approximately the largest contiguously useful space we may find within Landsat imagery, we could expect $P \approx 36,000,000$. With $A = |M| = B = |R| = 3$ on average, and considering $R$ neighborhoods of maximum radius $R = 10$ pixels, corresponding to a real-world radius of approximately 300 meters, we would estimate $V = \pi R^2 - 1 \approx 313.159$ vectors, but in practice 316 vectors are within this radius. Then (2.2) suggests that $409,536,000,000$ comparisons will be required, $79,596,000,000$ additions, and $79,632,000,000$ multiplications per layer of Landsat data. While modern hardware is very capable, this can still be quite an expensive procedure. If we double the resolution of the data—if the user is only interested in some feature found in the RGB data and computes a panchromatic image using band 8, or we use a different, 15 meter resolution data source—then $P$ would increase to $(2 \cdot 6000)^2 = 144,000,000$, $R$ would increase to 20, and consequently
$V$ would climb to 1256. We would then need to utilize 6,511,000,000 comparisons, 1,265,904,000,000 additions, and 1,266,048,000,000 multiplications. This leads us to recognize that this procedure does not scale well; in theory, the computational cost for increasing the resolution by a factor of $n$ scales with $O(n^4)$. Thus it seems evident that this procedure serves the user best in cases where the resolution of a scene is naturally limited.
<table>
<thead>
<tr>
<th>Abbr.</th>
<th>Long name</th>
<th>Formula or approximation</th>
</tr>
</thead>
<tbody>
<tr>
<td>pix</td>
<td>Pixels per image</td>
<td>Depends on size of scene to be analyzed</td>
</tr>
<tr>
<td>R</td>
<td>Max radius</td>
<td>$\sup \bigcup_{r \in R} r$</td>
</tr>
<tr>
<td>vec</td>
<td>Vectors under consideration</td>
<td>$\pi R^2 - 1$</td>
</tr>
<tr>
<td>grd</td>
<td>Cost for gradient</td>
<td>$\text{pix} \cdot [0, 6, 7]$</td>
</tr>
<tr>
<td>bnc</td>
<td>Cost to check a bin combination per gradient</td>
<td>$\text{pix} \cdot [4, 0, 0]$</td>
</tr>
<tr>
<td>$[G^v]$</td>
<td>Cost to compute all bin combinations for a gradient, yielding $G^v$</td>
<td>$</td>
</tr>
<tr>
<td>$[G]$</td>
<td>Cost to sum $G = \sum_v G^v$</td>
<td>$\text{pix} \cdot [0, \text{vec} - 1, 0]$</td>
</tr>
</tbody>
</table>

Table 2.2: A table of different sub-costs used to analyze the DMR computation procedure. For this discussion we use the notation $[c, a, m]$ to denote a computational cost of $c$ comparisons, $a$ additions, and $m$ multiplications. We assume reshaping operations are free.
Chapter 3

Theory of the radial Fourier descriptor

The general concept of the radial Fourier descriptor (henceforth RFD) is quite similar to that of the differential-magnitude and radius descriptor. Around any point in $\mathbb{R}^2$, we can circumscribe a circle with any positive radius. Each of these will be homeomorphic to the unit circle $\mathbb{T}$, and given a smooth function $f : \mathbb{R}^2 \to \mathbb{R}$ to be described (a stand-in for Landsat imagery, as in the DMR theory), the restriction of $f$ to a circle around a point can be viewed as a map from the circle $\mathbb{T}$ to $\mathbb{R}$, which permits one to describe it using concepts from Harmonic Analysis, namely, the function’s Fourier coefficients. These Fourier coefficients, collected at a variety of different radii, may prove to be useful descriptors of $f$ for the circumscribed point.

While this theory, again, does not apply directly to Landsat imagery, the discrete analogue of Fourier coefficients and the Fourier transform, the discrete Fourier transform (DFT), can be applied to discrete sequences of regional averages in an image, with regions circumscribing a point, and this itself may be a useful description of that point in the image. Furthermore, the sheer efficiency of the fast Fourier transform (FFT) algorithm may allow us to an alternative description of the traits discovered by the DMR methodology, but with a dramatically reduced cost associated with extracting these descriptors, as well as a lessened need for a precise calibration of the parameters involved, since the radial Fourier descriptor scales only in the dimension of the radii described, unlike the differential-magnitude and radius descriptor which is parameterized by both radial bins and also magnitudinal bins as well.
3.1 Theory of Fourier coefficients

We will give only a basic introduction into the theory of Fourier coefficients; it will generally be based on Katznelson [14], and that text is recommended for a more complete explanation of these concepts.

The general concept of Fourier coefficients, and more generally the Fourier transform, is that one may approximate a function $f$ to arbitrary precision using sinusoids; and, specifically, by a trigonometric polynomial.

**Definition 3.1.1.** A function $p$ on $\mathbb{R}$ of the form

$$p(t) = \sum_{n=-N}^{N} a_n e^{i\xi_n t}, \quad \xi_n \in \mathbb{R}$$

is a **trigonometric polynomial** on $\mathbb{T}$.

**Definition 3.1.2.** The **Fourier coefficients** $\hat{f}$ of $f \in L^1(\mathbb{T})$ are defined by $\hat{f} : \mathbb{N} \to \mathbb{C}$,

$$\hat{f}(n) \overset{\text{def}}{=} \frac{1}{2\pi} \int f(t) \cdot e^{-int} \, dt$$

We note here the domain of $f$; while readers familiar with the Fourier transform will recall that the Fourier transform of a function on the real line will also be a function on the real line, for a function $f$ on $\mathbb{T}$, the Fourier coefficient function $\hat{f}$ will be defined on the integers; the essential intuitive difference between these two cases may be seen in the compactness of $\mathbb{T}$ compared to that of $\mathbb{R}$. Similar to the case of the Stone-Weierstrass Theorem, which observes that polynomials in $C(X, \mathbb{R})$ with $X$ a compact subset of $\mathbb{R}$ can be approximated uniformly with **polynomials**, so also analogously can we approximate functions on $\mathbb{T}$ with **trigonometric polynomials**. In contrast, $\mathbb{R}$ is not a
compact space, and to approximate functions on a function with real domain we require stronger techniques. However, the functions we work with here naturally have $\mathbb{T}$ as a domain, and thus the additional muscle of the full Fourier transform is excessive.

The Fourier coefficient operator obeys a number of useful properties; these properties are absolutely essential for the radial Fourier descriptor.

**Theorem 3.1.3.** Suppose $f, g \in L^1(\mathbb{T})$. Then

(A) $[\hat{f} + \hat{g}](n) = \hat{f}(n) + \hat{g}(n)$

(B) If $a \in \mathbb{C}$, then $[\hat{af}](n) = a\hat{f}(n)$.

(C) Denote $f_\theta(t) = f(t - \theta)$ for $\theta \in \mathbb{T}$. Then $\hat{f_\theta}(n) = \hat{f}(n)e^{-in\theta}$.

**Proof.**

\[
[f + g](n) = \frac{1}{2\pi} \int [f + g](t) \cdot e^{-int} \, dt = \frac{1}{2\pi} \int f(t) \cdot e^{-int} \, dt + \frac{1}{2\pi} \int g(t) \cdot e^{-int} \, dt
\]

\[
= \hat{f}(n) + \hat{g}(n) \quad \text{(A)}
\]

\[
[af](n) = \frac{1}{2\pi} \int [af](t) \cdot e^{-int} \, dt = \frac{1}{2\pi} \int af(t) \cdot e^{-int} \, dt
\]

\[
= a \frac{1}{2\pi} \int f(t) \cdot e^{-int} \, dt = a\hat{f}(n) \quad \text{(B)}
\]

\[
\hat{f_\theta}(n) = \frac{1}{2\pi} \int f_\theta(t) \cdot e^{-int} \, dt
\]
\[
\begin{align*}
&= \frac{1}{2\pi} \int f(t - \theta) \cdot e^{-int} \, dt \\
&= \frac{1}{2\pi} \int f(t') \cdot e^{-in(t' + \theta)} \, dt' \\
&= \left[ \frac{1}{2\pi} \int f(t') \cdot e^{-int'} \, dt' \right] e^{-in\theta} \\
&= \hat{f}(n) \cdot e^{-in\theta}
\end{align*}
\] (C)

These properties equip the Fourier coefficients with properties invaluable to our objective. Properties (A) and (B) yield linearity. This is necessary to ensure that if, for example, a map \( f \) can be decomposed into \( f(t) = g(t) + h(t) \), where \( g(t) \) is considered an “ordinary” signal and \( h(t) \) is the extraordinary part, which we wish to detect, \( \hat{f} \) and \( \hat{g} \) will be distinguishable by an increase in \( ||\hat{f} - \hat{g}|| = ||\hat{h}|| \) (for an appropriate choice of norm \( || \cdot || \) on the set of functions from \( \mathbb{Z} \) to \( \mathbb{C} \)). Property (C) of Fourier coefficients yields an extremely valuable trait of the Fourier coefficients, describing how they operate under rotations of \( \mathbb{T} \). While we do not have simple invariance under rotations of \( \mathbb{T} \), we notice that

\[
|\hat{f}_\theta(n)| = |\hat{f}(n) \cdot e^{-in\theta}| = |\hat{f}(n)| \cdot |e^{-in\theta}| = |\hat{f}(n)| \cdot 1 = |\hat{f}(n)|
\]

and thus the absolute value of the Fourier coefficients of \( f \) are invariant under rotations of the space. This descriptor’s parallel in the discrete Fourier transform is precisely the radial Fourier descriptor, which we apply to the problem of identifying data centers.
3.2 The discrete Fourier transform and fast Fourier transform algorithm

**Definition 3.2.1.** Given a collection of \( N \) equally spaced complex-valued data points \( \{x_0, x_1, \cdots, x_{N-1}\} \), the **discrete Fourier transform** of these points is the collection \( \{X_k\} \) defined by
\[
X_k \overset{\text{def}}{=} \sum_{n=0}^{N-1} x_n \cdot e^{-2\pi ikn/N}, \quad k \in \mathbb{Z} \tag{3.2}
\]
Using function notation, if \( f : \mathbb{Z}/N\mathbb{Z} \rightarrow \mathbb{C} \) corresponds to the data points by \( f(k) = x_k \), then we define the discrete Fourier transform \( \hat{f} \) to be
\[
\hat{f}(k) \overset{\text{def}}{=} \sum_{n \in \mathbb{Z}/N\mathbb{Z}} f(n) \cdot e^{-2\pi ikn/N} \tag{3.3}
\]
For brevity of notation, and consistency with traditional notational conventions, we denote \( e^{2\pi i/N} \), the primitive \( N \)-th root of unity, with the symbol \( \omega \).

The discrete Fourier transform shares each of the properties of Theorem 3.1.3; however, property (C) must be modified slightly to match the new domain for our function.

**Theorem 3.2.2.** Suppose \( f, g : \mathbb{Z}/N\mathbb{Z} \rightarrow \mathbb{C} \). Then

(A) \( \widehat{f + g}(d) = \hat{f}(d) + \hat{g}(d) \)

(B) If \( a \in \mathbb{C} \), then \( \widehat{af}(d) = a\hat{f}(d) \).

(C) For \( r \in \mathbb{Z}/N\mathbb{Z} \), denote \( f_r(n) = f(n - r) \). Then \( \hat{f}_r(d) = \hat{f}(d) \cdot \omega^{-rd_m} \)

*Proof.* Parts (A) and (B) follow quite directly from the definition in a similar manner to the proofs of the corresponding parts of Theorem 3.1.3.
For part (C), suppose \( r \in \mathbb{Z}/N\mathbb{Z} \). Then

\[
\hat{f}_r(d) = \sum_{n \in \mathbb{Z}/N\mathbb{Z}} f_r(n) \cdot \omega^{-dn} \\
= \sum_{n \in \mathbb{Z}/N\mathbb{Z}} f(n - r) \cdot \omega^{-dn} \\
= \sum_{m \in \mathbb{Z}/N\mathbb{Z}} f(m) \cdot \omega^{-d(m+r)} \quad \text{(Change of variable } m = n - r) \\
= \left[ \sum_{m \in \mathbb{Z}/N\mathbb{Z}} f(m) \cdot \omega^{-dm} \right] \cdot \omega^{-dr} \\
= \hat{f}(d) \cdot \omega^{-dr} \quad \text{(C)}
\]

Note that the change of variable \( m = n - r \) is a translation, which preserves the finite group \( \mathbb{Z}/N\mathbb{Z} \) which indexes the summation.

We may simplify the computation of the discrete Fourier transform using a fast Fourier transform (FFT) algorithm. In particular, we describe the Cooley-Tukey FFT algorithm [6], which was introduced by James W. Cooley and John W. Tukey in 1964. Their particular implementation further improves upon the concepts discussed here by taking into account the storage space used in the computation as well, but we believe the details obfuscate the mathematical concepts underlying the FFT algorithm and these have been omitted to better convey the simplifications underlying the procedure.

We assume that the number of samples \( N \) is highly composite. While this may not always be the case, this condition is not generally difficult to meet; often we may force this to be the case by simply sampling our data at a particularly composite frequency.
Then we know $N = N_1 N_2$ and $N_1, N_2 \in \mathbb{N}$ are both greater than 1. Then for $n \in \{0, 1, \cdots, N - 1\}$, we may write $n = N_1 m + k$ for $m \in \{0, 1, \cdots, N_2 - 1\}$ and $k \in \{0, 1, \cdots, N_1 - 1\}$. Then

$$\hat{f}(d) = \sum_{n=0}^{N-1} f(n) \cdot \omega_N^{-dn}$$

$$= \sum_{k=0}^{N_1-1} \sum_{m=0}^{N_2-1} f(N_1 m + k) \cdot \omega_N^{-d(N_1 m + k)}$$

$$= \sum_{k=0}^{N_1-1} \sum_{m=0}^{N_2-1} f(N_1 m + k) \cdot (\omega_N^{N_1})^{-dm} \cdot \omega_N^{-dk}$$

$$= \sum_{k=0}^{N_1-1} \omega_N^{-dk} \sum_{m=0}^{N_2-1} f(N_1 m + k) \cdot (\omega_N^{N_1})^{-dm}$$

Note that $\omega_N^{N_1} = \omega_{N_1/N_1} = \omega_{N_2}$.

$$= \sum_{k=0}^{N_1-1} \omega_N^{-dk} \sum_{m=0}^{N_2-1} f(N_1 m + k) \cdot \omega_N^{-dm}$$

If we let $f_k$ be the restriction of $f$ to the coset $k + \mathbb{Z}/N_2\mathbb{Z}$ (mapping $N_1 m + k \to m$), then we observe that the interior summations are in fact also discrete Fourier transforms.

$$= \sum_{k=0}^{N_1-1} \omega_N^{-dk} \sum_{m=0}^{N_2-1} f_k(m) \cdot \omega_N^{-dm}$$

$$= \sum_{k=0}^{N_1-1} \omega_N^{-dk} \hat{f}_k(d)$$

Computing a discrete Fourier transform will, if we use the literal sequence of computations implied by equation (3.2), require a total of $N - 1$ complex additions and $N$
Table 3.1: The (theoretical) number of multiplications and additions required to compute a discrete Fourier transform of $2^{10}$ points with and without the use of one iteration of the fast Fourier transform for various factorizations $2^{10} = N_1N_2$. The new values are computed with (3.4), while the originals are taken from (3.2).

<table>
<thead>
<tr>
<th>$N_1$</th>
<th>$N_2$</th>
<th>Additions</th>
<th>Mult.</th>
<th>New add per old</th>
<th>New mult per old</th>
</tr>
</thead>
<tbody>
<tr>
<td>$2^4$</td>
<td>$2^6$</td>
<td>524800</td>
<td>4096</td>
<td>0.50097752</td>
<td>0.00390625</td>
</tr>
<tr>
<td>$2^2$</td>
<td>$2^8$</td>
<td>264960</td>
<td>8192</td>
<td>0.25293255</td>
<td>0.00781250</td>
</tr>
<tr>
<td>$2^3$</td>
<td>$2^7$</td>
<td>138112</td>
<td>16384</td>
<td>0.13184262</td>
<td>0.01562500</td>
</tr>
<tr>
<td>$2^4$</td>
<td>$2^6$</td>
<td>80832</td>
<td>32768</td>
<td>0.07716276</td>
<td>0.03125000</td>
</tr>
<tr>
<td>$2^5$</td>
<td>$2^5$</td>
<td>64480</td>
<td>65536</td>
<td>0.06155303</td>
<td>0.06250000</td>
</tr>
<tr>
<td>$2^6$</td>
<td>$2^4$</td>
<td>80880</td>
<td>131072</td>
<td>0.07720858</td>
<td>0.12500000</td>
</tr>
<tr>
<td>$2^7$</td>
<td>$2^3$</td>
<td>138232</td>
<td>262144</td>
<td>0.13195717</td>
<td>0.25000000</td>
</tr>
<tr>
<td>$2^8$</td>
<td>$2^2$</td>
<td>265212</td>
<td>524288</td>
<td>0.25317311</td>
<td>0.50000000</td>
</tr>
<tr>
<td>$2^9$</td>
<td>$2^1$</td>
<td>525310</td>
<td>1048576</td>
<td>0.50146437</td>
<td>1.00000000</td>
</tr>
<tr>
<td>$2^{10}$</td>
<td>-</td>
<td>1047552</td>
<td>1048576</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

Complex multiplications\(^1\)(henceforth written as a tuple containing addition count and multiplication count respectively, $[N-1, N]$) to calculate the DFT for a given value for $d$; for each of the $N$ values for $d$, this means our operation counts are $[N^2 - N, N^2]$. If we decompose the discrete Fourier transform as above, then there will be $N_1$ distinct $\hat{f}_k$ of length $N_2$ to compute, which will (if calculated by (3.2)) require $[N_2^2 - N_2, N_2^2]$ each, and thus $N_1 \cdot [N_2^2 - N_2, N_2^2] = [NN_2 - N, NN_2]$ to compute $\hat{f}_k(d)$ for each $k$ and $d$. From this point, computing $\hat{f}(d)$ takes $[N_1 - 1, N_1]$ operations; there are $N$ of these, so the total operation counts would be

$$[NN_2 - N_2, NN_2] + N \cdot [N_1 - 1, N_1] = [NN_2 - N_2 + NN_1 - N, NN_2 + NN_1]$$  \text{(3.4)}

For large values of $N$, this can yield quite a dramatic difference in the quantity of operations required, as can be seen in Table 3.1. When it is true that multiplication is

\(^1\)We assume that we do not have to compute the complex roots of unity, since these may be computed in advance and stored in a table to be used in any number of DFT computations of similar size, and thus this cost is effectively negligible.
substantially more expensive than addition, as is often the case\(^2\), the dramatic improvements in multiplication count may justify the technique, even disregarding the relatively mild improvement in addition count.

Furthermore, we can nest this procedure repeatedly. The expression in (3.4) can be generalized into the expression, for \( N = N_1N_2 \) and operation counts for an arbitrarily chosen DFT method \( * \) for a length \( N \) DFT denoted by \( \text{Ops}(N; *) \), we may write a (potentially) recursive equation

\[
\text{Ops}(N_1N_2; \text{fft}) = N_1 \cdot \text{Ops}(N_2; *) + [NN_1 - N, NN_1] \quad (3.5)
\]

Indeed, Table 3.2 shows the values for various factorizations \( 2^{10} = N_1N_2N_3 \). Each of these is a substantial improvement over the most basic DFT calculation, but we can reduce the number of each type of operation by over 95% (for \( N = 2^32^42^3 \)). When factoring \( 2^{10} \) fully, into \( 2 \cdot 2 \cdot 2 \cdot 2 \cdot 2 \cdot 2 \cdot 2 \cdot 2 \cdot 2 \cdot 2 \cdot 1 \) and iterating the FFT reduction ten times, we require only 10240 additions and 21504 multiplications, less than 1% of the additions we originally required and approximately 2.05% of the multiplications required originally, a dramatic reduction in computational complexity.

As can be seen in this example, for discrete Fourier transforms of highly composite sample sizes, the Cooley-Tukey FFT algorithm can dramatically decrease the number of operations required. This advantage will inform a key design decision in the radial Fourier descriptor. We always extract \( 2^n \) samples, so as to ensure that the full advantages of the fast Fourier transform may always be exploited.

\(^2\)This is usually the case, as complex multiplication involves four real multiplications and two real additions, whereas complex addition only involves two real additions, but when working with the fourth roots of unity, \( i, -1, -i, \) and obviously 1, multiplication can be extremely easy.
<table>
<thead>
<tr>
<th>$N_1$</th>
<th>$N_2$</th>
<th>$N_3$</th>
<th>Additions</th>
<th>Mult.</th>
<th>New add/old</th>
<th>New mult/old</th>
</tr>
</thead>
<tbody>
<tr>
<td>$2^1$</td>
<td>$2^1$</td>
<td>$2^8$</td>
<td>524288</td>
<td>528384</td>
<td>0.50048876</td>
<td>0.50390625</td>
</tr>
<tr>
<td>$2^1$</td>
<td>$2^2$</td>
<td>$2^7$</td>
<td>264192</td>
<td>268288</td>
<td>0.25219941</td>
<td>0.25585938</td>
</tr>
<tr>
<td>$2^1$</td>
<td>$2^3$</td>
<td>$2^6$</td>
<td>137216</td>
<td>141312</td>
<td>0.13098729</td>
<td>0.13476562</td>
</tr>
<tr>
<td>$2^1$</td>
<td>$2^4$</td>
<td>$2^5$</td>
<td>79872</td>
<td>83968</td>
<td>0.07624633</td>
<td>0.08007812</td>
</tr>
<tr>
<td>$2^1$</td>
<td>$2^5$</td>
<td>$2^4$</td>
<td>63488</td>
<td>67584</td>
<td>0.06060606</td>
<td>0.06445312</td>
</tr>
<tr>
<td>$2^1$</td>
<td>$2^6$</td>
<td>$2^3$</td>
<td>79872</td>
<td>83968</td>
<td>0.07624633</td>
<td>0.08007812</td>
</tr>
<tr>
<td>$2^1$</td>
<td>$2^7$</td>
<td>$2^2$</td>
<td>137216</td>
<td>141312</td>
<td>0.13098729</td>
<td>0.13476562</td>
</tr>
<tr>
<td>$2^1$</td>
<td>$2^8$</td>
<td>$2^1$</td>
<td>264192</td>
<td>268288</td>
<td>0.25219941</td>
<td>0.25585938</td>
</tr>
<tr>
<td>$2^2$</td>
<td>$2^1$</td>
<td>$2^7$</td>
<td>264192</td>
<td>268288</td>
<td>0.25219941</td>
<td>0.25585938</td>
</tr>
<tr>
<td>$2^2$</td>
<td>$2^2$</td>
<td>$2^6$</td>
<td>135168</td>
<td>139264</td>
<td>0.12903226</td>
<td>0.13281250</td>
</tr>
<tr>
<td>$2^2$</td>
<td>$2^3$</td>
<td>$2^5$</td>
<td>73728</td>
<td>77824</td>
<td>0.07038123</td>
<td>0.07421875</td>
</tr>
<tr>
<td>$2^2$</td>
<td>$2^4$</td>
<td>$2^4$</td>
<td>49152</td>
<td>53248</td>
<td>0.04692082</td>
<td>0.05078125</td>
</tr>
<tr>
<td>$2^2$</td>
<td>$2^5$</td>
<td>$2^3$</td>
<td>49152</td>
<td>53248</td>
<td>0.04692082</td>
<td>0.05078125</td>
</tr>
<tr>
<td>$2^2$</td>
<td>$2^6$</td>
<td>$2^2$</td>
<td>73728</td>
<td>77824</td>
<td>0.07038123</td>
<td>0.07421875</td>
</tr>
<tr>
<td>$2^2$</td>
<td>$2^7$</td>
<td>$2^1$</td>
<td>135168</td>
<td>139264</td>
<td>0.12903226</td>
<td>0.13281250</td>
</tr>
<tr>
<td>$2^3$</td>
<td>$2^1$</td>
<td>$2^6$</td>
<td>137216</td>
<td>141312</td>
<td>0.13098729</td>
<td>0.13476562</td>
</tr>
<tr>
<td>$2^3$</td>
<td>$2^2$</td>
<td>$2^5$</td>
<td>73728</td>
<td>77824</td>
<td>0.07038123</td>
<td>0.07421875</td>
</tr>
<tr>
<td>$2^3$</td>
<td>$2^3$</td>
<td>$2^4$</td>
<td>45056</td>
<td>49152</td>
<td>0.04301075</td>
<td>0.04687500</td>
</tr>
<tr>
<td>$2^3$</td>
<td>$2^4$</td>
<td>$2^3$</td>
<td>36864</td>
<td>40960</td>
<td>0.03519062</td>
<td>0.03906250</td>
</tr>
<tr>
<td>$2^3$</td>
<td>$2^5$</td>
<td>$2^2$</td>
<td>45056</td>
<td>49152</td>
<td>0.04301075</td>
<td>0.04687500</td>
</tr>
<tr>
<td>$2^3$</td>
<td>$2^6$</td>
<td>$2^1$</td>
<td>73728</td>
<td>77824</td>
<td>0.07038123</td>
<td>0.07421875</td>
</tr>
<tr>
<td>$2^4$</td>
<td>$2^1$</td>
<td>$2^5$</td>
<td>79872</td>
<td>83968</td>
<td>0.07624633</td>
<td>0.08007812</td>
</tr>
<tr>
<td>$2^4$</td>
<td>$2^2$</td>
<td>$2^4$</td>
<td>49152</td>
<td>53248</td>
<td>0.04692082</td>
<td>0.05078125</td>
</tr>
<tr>
<td>$2^4$</td>
<td>$2^3$</td>
<td>$2^3$</td>
<td>36864</td>
<td>40960</td>
<td>0.03519062</td>
<td>0.03906250</td>
</tr>
<tr>
<td>$2^4$</td>
<td>$2^4$</td>
<td>$2^2$</td>
<td>36864</td>
<td>40960</td>
<td>0.03519062</td>
<td>0.03906250</td>
</tr>
<tr>
<td>$2^4$</td>
<td>$2^5$</td>
<td>$2^1$</td>
<td>49152</td>
<td>53248</td>
<td>0.04692082</td>
<td>0.05078125</td>
</tr>
<tr>
<td>$2^5$</td>
<td>$2^1$</td>
<td>$2^4$</td>
<td>63488</td>
<td>67584</td>
<td>0.06060606</td>
<td>0.06445312</td>
</tr>
<tr>
<td>$2^5$</td>
<td>$2^2$</td>
<td>$2^3$</td>
<td>49152</td>
<td>53248</td>
<td>0.04692082</td>
<td>0.05078125</td>
</tr>
<tr>
<td>$2^5$</td>
<td>$2^3$</td>
<td>$2^2$</td>
<td>45056</td>
<td>49152</td>
<td>0.04301075</td>
<td>0.04687500</td>
</tr>
<tr>
<td>$2^5$</td>
<td>$2^4$</td>
<td>$2^1$</td>
<td>49152</td>
<td>53248</td>
<td>0.04692082</td>
<td>0.05078125</td>
</tr>
<tr>
<td>$2^6$</td>
<td>$2^1$</td>
<td>$2^3$</td>
<td>79872</td>
<td>83968</td>
<td>0.07624633</td>
<td>0.08007812</td>
</tr>
<tr>
<td>$2^6$</td>
<td>$2^2$</td>
<td>$2^2$</td>
<td>73728</td>
<td>77824</td>
<td>0.07038123</td>
<td>0.07421875</td>
</tr>
<tr>
<td>$2^6$</td>
<td>$2^3$</td>
<td>$2^1$</td>
<td>73728</td>
<td>77824</td>
<td>0.07038123</td>
<td>0.07421875</td>
</tr>
<tr>
<td>$2^7$</td>
<td>$2^1$</td>
<td>$2^2$</td>
<td>137216</td>
<td>141312</td>
<td>0.13098729</td>
<td>0.13476562</td>
</tr>
<tr>
<td>$2^7$</td>
<td>$2^2$</td>
<td>$2^1$</td>
<td>135168</td>
<td>139264</td>
<td>0.12903226</td>
<td>0.13281250</td>
</tr>
<tr>
<td>$2^8$</td>
<td>$2^1$</td>
<td>$2^1$</td>
<td>264192</td>
<td>268288</td>
<td>0.25219941</td>
<td>0.25585938</td>
</tr>
</tbody>
</table>

Table 3.2: The (theoretical) number of multiplications and additions required to compute a discrete Fourier transform of $2^{10}$ points with and without the use of two iterations of the fast Fourier transform for various factorizations $2^{10} = N_1 N_2 N_3$. 
3.3 The radial Fourier descriptor

The radial Fourier descriptor (RFD) is designed to take the concept of the Fourier transform around a circle around a point as a description of that point; the magnitudes of the sinusoids around a given point will act to describe the point we are interested in.

**Definition 3.3.1.** If $\mathcal{I}: X \to \mathbb{R}$ is our image (or a function to be analyzed), we call $R_{\mathcal{I}}: X \times \mathbb{T} \to \mathbb{R}$ a **representative function** for $\mathcal{I}$ if $R_{\mathcal{I}}(x, \theta)$ in some way represents the behavior of $\mathcal{I}$ in the direction of the vector $(\cos(\theta), \sin(\theta))$ from $x$. We expect that $R_{\mathcal{I}}$ is invariant under translations of $\mathcal{I}$, so that for any $y \in \mathbb{R}^2$, $R_{\mathcal{I}}(x, \theta) \approx R_{\mathcal{I}+y}(x+y, \theta)$. Furthermore, we expect this representative function to behave similarly under rotation of the space; that is, if $[\phi] \in \text{SO}(2)$ rotates elements of $\mathbb{R}^2$ by $\phi$, then $R_{\mathcal{I}}(x, \theta) \approx R_{[\phi]\mathcal{I}}([\phi] x, \theta + \phi)$. These expectations are illustrated in Figure 3.1.

**Example 3.3.2.** In the continuous case, where $\mathcal{I}: \mathbb{R}^2 \to \mathbb{R}$, we can let $R_{\mathcal{I}}((x, y), \theta) = \mathcal{I}(x + r \cos(\theta), y + r \sin(\theta))$ for $r > 0$ be a representative function for $\mathcal{I}$. This simply
describes each point in $\mathcal{I}$ by restricting the function $\mathcal{I}$ to the circle of radius $r$ around that point.

**Example 3.3.3.** Given a measurable function $\mathcal{I} : \mathbb{R}^2 \to \mathbb{R}$, we can, for any $0 < \epsilon < 2\pi$, let

$$S_{x\theta} = \left\{ x + (a \cos \theta', a \sin \theta') : a \in [0, 1], \theta' \in \left[ \theta - \frac{\epsilon}{2}, \theta + \frac{\epsilon}{2} \right] \right\}$$

be the sector of width $\epsilon$ and radius 1 in the direction $\theta$ from $x$. Then we may define a representative function $R_S$ by

$$R(x, \theta) = \frac{1}{\lambda(S_{x\theta})} \int_{S_{x\theta}} \mathcal{I} \, d\lambda$$

which maps a pair $(x, \theta)$ to the average function value in the sector $S_{x\theta}$.

Given this definition, it is evident that at any given point $x$, a representative function can be viewed as a function $R_x$ on $\mathbb{T}$ defined by $R_x : \theta \mapsto R(x, \theta)$; if this function is an element of $L_1$, then we can then compute the Fourier coefficients at this point, $\hat{R}_x(n)$.

**Example 3.3.4.** Consider the function $\mathcal{I} = 1_A$, where $A = [-1.5, -0.5] \times [-0.5, 1.5]$, depicted in Figure 3.2. We can assign to $\mathcal{I}$ the representative function $R(x, \theta) = \mathcal{I}(x + (\cos \theta, \sin \theta))$. From this, we may obtain an expression for $R_0$ as a sum of sinusoids using the Fourier coefficients $\hat{R}_0$.

**Definition 3.3.5.** Suppose we are given an image $\mathcal{I} : \mathbb{Z}^2 \to \mathbb{R}$, inner and outer radii $r_0$ and $r_1$, $0 < r_0 < r_1$, and a “sampling rate” $n \in \mathbb{N}$, $n > 1$. Let

$$S_{x\theta} = \mathbb{Z}^2 \cap \left\{ x + (a \cos \theta', a \sin \theta') : a \in [r_0, r_1], \theta' \in [\theta - 2\pi/n, \theta + 2\pi/n] \right\}$$
Figure 3.2: Consider the function $1_A$, with $A = [-1.5, -0.5] \times [-0.5, 1.5]$ (first image). Restricting this to the unit circle around 0 yields the representative function $f(\theta) = 1_{[2\pi/3, 7\pi/6]}(\theta)$. We can then decompose this into a sequence of Fourier coefficients which represent sinusoids (second image), a sum of these reconstructing the behavior of the representative function. The above plot is constructed from 13 sinusoids, with each partial sum denoted in increasingly dark blue.
and define a representative function for $I$ by

$$R(x, \theta) = R_x(\theta) \overset{\text{def}}{=} \frac{1}{||S_{x\theta}||} \sum_{z \in S_{x\theta}} I(z)$$

which maps a vector $x$ and an angle $\theta$ to the average value of the annulus of inner radius $r_0$ and outer radius $r_1$ in the direction of $\theta$ from $x$. We define $R^*_x: \mathbb{Z}/n\mathbb{Z} \rightarrow \mathbb{R}$ as the restriction of $R_x$ to the cyclic subgroup $G_n \leq \mathbb{T}$ generated by $2\pi/n$, under the homomorphism $\phi : \mathbb{Z}/n\mathbb{Z} \rightarrow G_n$, $a \mapsto 2\pi a/n$; that is,

$$R^*_x : \mathbb{Z}/n\mathbb{Z} \xrightarrow{\phi} G_n \xrightarrow{R} \mathbb{R}$$

The radial Fourier descriptor of $I$ over the annulus $\text{ann}(x; r_0, r_1)$ is the vector-valued function

$$[\text{RFD}_I(x)]_k \overset{\text{def}}{=} \left| \hat{R}^*_x(k) \right|, \quad k = 0, \ldots, n - 1 \quad (3.6)$$

We observe a few valuable properties of the radial Fourier descriptor. The sector averages are approximately rotationally invariant; if we rotate by $\pi/2$—multiples of which make up all of the viable rotations of $\mathbb{Z}^2$—then we expect, if $4 \mid n$, that the sector $S_{[\pi/2]x, \theta + \pi/2}$ in the rotated space (where $[\pi/2]$ denotes the rotation-by-$\pi/2$ matrix) should be effectively identical (other than a change of general direction) to the sector $S_{x\theta}$ in the original $I$, including its average. Thus, a $\pi/2$ rotation of the space will induce a rotation of $n/4$ in $R^*$. Due to Theorem 3.2.2 (C), this rotation in $\mathbb{Z}/n\mathbb{Z}$ only introduces roots of unity into the modulus of the RFD, which drop out in the same manner as demonstrated in equation (3.1); so the RFD at a point in $I$ is identical to that of its image under the rotation. In general, we expect that rotations of $I$ will retain similar radial Fourier descriptors, despite the discretization of the procedure.
The intuition behind the radial Fourier descriptor is that it describes a given point in a scene by the magnitudes of the sinusoids needed to depict the image in a circle around that point. These magnitudes, and particularly the first few, give us valuable information about the general behavior of the neighborhood around a pixel; it is likely to pick up the corner of a building, for example, or the roads leading to and from a structure, and these clues in turn will allow the support vector machine to distinguish a data center from other structures.
Chapter 4
Support vector machines

The support vector machine (SVM) is a technology developed for solving classification problems; that is, problems which may be formulated in the form,

\[
\text{Given a (finite) set of data sampled from some space } X \text{ (some subspace of } \mathbb{R}^n \text{ or } \mathbb{C}^n \text{) classified in a continuous probabilistic manner into classes 1 or } -1, \{ (x_i, c_i) \in X \times \{1, -1\} : i \in I \}, \text{ how do we best extend this classification into a function } f : X \rightarrow \{1, -1\} \text{ such that } f \text{ reflects the underlying distributions of the two classes?}
\]

In particular, the SVM method attempts to find a hyperplane directly between and separating the two classes of points. Further extensions to the theory permit a generalization to a nonlinear classification and also to additional classes.

Our definition of a support vector machine roughly follows that on page 418 of Friedman et al. [11], and that text is offered for further information on support vector machines. In addition, the theoretical approach and notation we will use will be flavored by the field of functional analysis, for which we refer the reader to Kreyszig [15].

4.1 The Separating Hyperplane Theorem

Suppose \( \beta \in H \) a finite-dimensional Hilbert space over \( \mathbb{R} \), \( ||\beta|| = 1 \), and \( 0 \neq \beta_0 \in \mathbb{R} \). Then we may define a classification \( G \) of \( X \) into \( \{-1, 1\} \) by

\[
G(x) = \text{sgn} (\langle \beta, x \rangle + \beta_0) \tag{4.1}
\]
where the sign function \( \text{sgn} \) maps negatives to \(-1\) and nonnegatives to \(1\). Of course, such a classification could be obtained with any function from \(X\) to \(\mathbb{R}\) mapped through \(\text{sgn}\). In particular we utilize the fact that \(\langle \beta, \cdot \rangle\) can be viewed as a linear functional \(f(x) = \langle \beta, x \rangle\) on \(H\) by the Riesz Representation Theorem [15, p. 188], which naturally ranges over all of \(\mathbb{R}\) and thus has null space of one dimension less. This null space will partition \(H\) into two disjoint components, as will the affine subspace of \(H\) defined by the zeros of \(g(x) = \langle \beta, x \rangle + \beta_0\).

**Lemma 4.1.1.** The null space \(\mathcal{N}(g)\) of \(g(x) = \langle \beta, x \rangle + \beta_0, g : H \to \mathbb{R}\) with \(H\) a finite dimensional Hilbert space over \(\mathbb{R}\), partitions \(H \setminus \mathcal{N}(g)\) into two disjoint connected components; specifically, \([+1] = \{x \in H : g(x) > 0\}\) and \([-1] = \{x \in H : g(x) < 0\}\).

*Proof.* It is evident that every point of \(H \setminus \mathcal{N}(g)\) must fall into one of these two classes. Suppose that \(x, y \in [+1]\). We assume without loss of generality that \(g(y) \geq g(x)\). Then we may define a path \(p : [0, 1] \to H\) between them by \(p(t) = (1 - t)x + ty\). Then

\[
\begin{align*}
g(p(t)) &= g((1 - t)x + ty) \\
&= \langle \beta, (1 - t)x + ty \rangle + \beta_0 \\
&= (1 - t)\langle \beta, x \rangle + t\langle \beta, y \rangle + \beta_0 \\
&= (1 - t)(g(x) - \beta_0) + tg(y) - t\beta_0 + \beta_0 \\
&= g(x) - \beta_0 - tg(x) + t\beta_0 + tg(y) - t\beta_0 + \beta_0 \\
&= g(x) + tg(y) - tg(x) \\
&= g(x) + t[g(y) - g(x)]
\end{align*}
\]
Since we know $g(x) > 0$, $t > 0$, and $g(y) - g(x) > 0$, we thus know that for all $t$, $g(p(t)) \in [+1]$. Thus, there is a path from $x$ to $y$ contained completely in $[+1]$, so $x$ and $y$ share the same connected component of $[+1]$. But $x$ and $y$ were arbitrary; consequently, $[+1]$ is composed of a single connected component. The proof that $[-1]$ is a single connected component proceeds similarly, except we assume originally that $g(y) < g(x)$ and conclude that $g(x)$ and $[g(y) - g(x)]$ are negative and thus $g(p(t))$ is contained entirely in $[-1]$.

To see that $[-1]$ and $[+1]$ are, in fact, distinct components of $H$, note that $[-1]$ and $[+1]$ are disjoint open sets covering $H \setminus \mathcal{N}(g)$. For any point $x$ in the space, we find that the open ball $B_{[g(x)]}(x)$ is a neighborhood around the point contained entirely in the respective class set. Thus these two disjoint open sets form a disconnection of the space.

Thus, since $[-1]$ and $[+1]$ form a pair of disjoint open sets which are each connected, $H \setminus \mathcal{N}(g)$ consists of two disjoint connected components, $[-1]$ and $[+1]$. □

**Lemma 4.1.2.** The Minkowski sum $A + B = \{x + y : x \in A, y \in B\}$ of two compact subsets of $\mathbb{R}^n$ is compact.

**Proof.** If $A$, $B$ are compact and $z \in \overline{A + B}$, the closure of the Minkowski sum, then $z_n \to z$ for $z_n \in A + B$. Then $z_n = x_n + y_n$ for some $x_n \in A$, $y_n \in B$. Since $A$ is compact, $x_n$ has a convergent subsequence $x_{n_i} \to x$. Since differences of convergent sequences are convergent, $y_{n_i} = z_{n_i} - x_{n_i} \to y$, which is an element of $y$ since the $B$ is closed. Thus $z = \lim_i x_{n_i} + y_{n_i} = \lim_i x_{n_i} + \lim_i y_{n_i} = x + y$, and thus $z \in A + B$. Consequently, $A + B$ is closed.
Since $A$ and $B$ are compact, they are bounded, and thus $||x|| < M$ and $||y|| < N$ for all $x \in A, y \in B$. So $||x + y|| \leq ||x|| + ||y|| < M + N$, and thus $A + B$ is bounded also.

Since $A + B$ is closed and bounded, it is compact.

It now makes some sense to ask: given some finite dataset $D = \{(x_i, c_i) \in H \times \{-1,1\}, i \in I\}$, where $H \subseteq \mathbb{R}^n$, can we separate the two sets $c^- = \{x_i : c_i = -1\}$ and $c^+ = \{x_i : c_i = 1\}$ with a hyperplane? In the case that the convex hulls

$$h^- = \left\{ \sum_{x' \in c^-} a_{x'} x' : \forall x' \in c^- (0 \leq a_{x'} \leq 1), \sum_{x' \in c^-} a_{x'} = 1 \right\}$$

of $c^-$ and $h^+$ (defined similarly) of $c^+$ are disjoint, we are provided an answer by the Separating Hyperplane Theorem [4, p. 46].

**Theorem 4.1.3** (Separating Hyperplane Theorem). *If $A$ and $B$ are disjoint nonempty compact convex sets in $\mathbb{R}^n$, then there exist $\beta \in \mathbb{R}^n$, $||\beta|| = 1$, and $c \in \mathbb{R}$ such that*

$$\forall a \in A, \langle \beta, a \rangle < c \quad \text{and} \quad \forall b \in B, \langle \beta, b \rangle > c.$$

The affine hyperplane satisfying $\langle \beta, x \rangle = c$ is a separating hyperplane between the two sets $A$ and $B$. To prove Theorem 4.1.3, we will first need a lemma.

**Lemma 4.1.4.** *If $A$ is a closed convex subset of $\mathbb{R}^n$, there exists a unique $x \in A$ with minimal $L_2$ norm.*

**Proof.** The existence of points with minimal norm is guaranteed by the Nearest Point Theorem (see Bartle [3, p. 78]) applied to the point 0 and $A$. 

---

43
To see that the nearest point is unique, suppose \( N = ||x|| = ||y|| \). Then consider the point \( \frac{1}{2}(x + y) \); so
\[
\left|\left|\frac{x + y}{2}\right|\right| = \frac{1}{2}||x + y|| \leq \frac{1}{2}||x|| + \frac{1}{2}||y|| = \frac{N}{2} + \frac{N}{2} = N
\]
with equality only if \( x = cy \) for some \( c \geq 0 \). Without equality, \( \left|\left|\frac{x + y}{2}\right|\right| < N \), and thus \( x \) and \( y \) were not minimal, a contradiction. With equality, then
\[
N = ||x|| = ||cy|| = |c||y|| = cN
\]
which can only occur if \( c = 1 \), and thus \( x = 1y = y \). Thus a closed convex subset of \( \mathbb{R}^n \) contains a unique element of minimal norm. \( \square \)

And now we proceed to prove Theorem 4.1.3.

**Proof.** Consider the set \( C = \{ y - x : x \in A, y \in B \} \). This is the Minkowski sum of two closed sets \( -A \) and \( B \), and thus it is closed by Lemma 4.1.2. Thus, by Lemma 4.1.4, \( C \) contains a unique vector \( z = y - x \) of minimal norm.\(^1\) Furthermore, this vector must be nonzero; if it were not, then \( y - x = 0 \) which implies \( x = y \) for \( x \in A, y \in B \), contradicting \( A \) and \( B \)'s disjointness.

Now we have two points \( x \) and \( y \) which are as close as two points of \( A \) and \( B \) may be. Then let us define \( \beta = (y - x) / ||y - x|| \) and \( c = \langle \beta, \frac{1}{2}(x + y) \rangle \). Then note that since \( \beta \) is a unit vector, the linear functional \( p'(r) = \langle \beta, r \rangle \) acts as a signed distance function from the null space of \( p \),\(^2\) which is naturally orthogonal to \( y - x \) by properties of inner

---

\(^1\)Note that \( z \) is unique, but \( x \) and \( y \) may not be. In particular, this occurs when \( A \) and \( B \)'s closest points lie on two parallel surfaces.

\(^2\)This is evident for the subspace generated by \( \beta \), since then \( d(0, x) = \sqrt{\langle x, x \rangle} = \sqrt{\langle a\beta, a\beta \rangle} = |a|\sqrt{\langle \beta, \beta \rangle} = |a| = ||\beta, x|| \). If \( x \) is not in the subspace generated by \( \beta \), then it can be decomposed \( x = a\beta + \gamma \), where \( \gamma \) lies in the orthogonal complement of the subspace generated by \( \beta \), and then \( d(\gamma, x) = \sqrt{\langle x - \gamma, x - \gamma \rangle} = \sqrt{\langle a\beta, a\beta \rangle} = ||\beta, x|| \). For any other point \( \gamma + \gamma' \), \( d(x, \gamma + \gamma') = \sqrt{\langle x - \gamma - \gamma', x - \gamma - \gamma' \rangle} = \sqrt{\langle a\beta - \gamma', a\beta - \gamma' \rangle} = \sqrt{\langle a\beta, a\beta \rangle + \langle \gamma', \gamma' \rangle} \geq \sqrt{\langle a\beta, a\beta \rangle} = d(x, \gamma) \). Thus the minimum distance to \( \mathcal{N}(p) \) is indeed \( ||\beta, x|| \).
Figure 4.1: A diagram of the proof of the Separating Hyperplane Theorem. $C$ corresponds to the collection of vectors beginning in $A$ and ending in $B$. Lemma 4.1.4 guarantees that $C$ contains some unique minimal vector, which corresponds to a pair $(x, y) \in A \times B$ of nearest points. Then an affine hyperplane perpendicular to $y - x$ separates our points, and this yields $\beta$ and $c$ for the separating hyperplane theorem.

Then suppose any element $y_0 \in B$ were closer to $\mathcal{N}(p)$ than $y$; that is, $|p(y_0)| < |p(y)|$. If $p(y_0) < 0$, we may use the Intermediate Value Theorem applied to the function $t \rightarrow p((1 - t)y + ty_0)$ to obtain a point $y' \in B$ such that $p(y') > 0$; we substitute this in place of $y_0$. So $p(y_0) = \langle \beta, y_0 \rangle - c < \langle \beta, y \rangle - c = p(y)$ and thus $0 < \langle \beta, y - y_0 \rangle$.

Then consider $y_t = (1 - t)y_0 + ty = y_0 + t(y - y_0)$. Notice that if $a < b$, then

\[
p(y_a) = \langle \beta, y_0 + ay - y_0 \rangle - c = p(y_0) + a \langle \beta, y - y_0 \rangle
\]

\[
p(y_b) = \langle \beta, y_0 + by - y_0 \rangle - c = p(y_0) + b \langle \beta, y - y_0 \rangle
\]
Figure 4.2: The three possible “altitudes” which may be dropped from $x$ to the closed affine subspace $y_{R}$. In the first case, $y_{s}$ must be equidistant from the hyperplane, and thus at least as far from $x$. In the second case, $y_{s}$ being closest forces $y_{0}$ to be closer to $x$ than $y_{1}$. In the third case, $y_{s}$ is further from the hyperplane than either $y_{0}$ or $y$, which ensures it is further from $x$ than $y$.

Since $b - a > 0$,

\[ p(y_{b}) - p(y_{a}) = (b - a) \langle \beta, y - y_{0} \rangle > 0 \]

\[ p(y_{b}) > p(y_{a}) \]

and thus $t \rightarrow p(y_{t})$ is monotonically increasing. $y_{R} = \{ y_{t} : t \in \mathbb{R} \}$ is an affine subspace, and thus a closed set, and thus $x$ must be closest to some point within by the Nearest Point Theorem. Furthermore, we know that $y_{[0,1]} = \{ y_{t} : t \in [0,1] \} \subseteq B$ by the convexity of $B$, and this closed set, an image of $[0,1]$ in $\mathbb{R}^{n}$, must have its own nearest point, which must be $y_{1} = y$, the nearest point to $x$ in all of $B$ and thus also of any subset of $B$ containing it. There are three possibilities:

1. The closest point to $x$ in $y_{R}$ lies inside $y_{[0,1]}$. Then, it is in $B$ and thus must be $y$ itself, so $y - x$ is orthogonal to the affine subspace $y_{R}$ and thus $y - x \perp y - y_{0}$.

Thus,

\[ \langle y - x, y - y_{0} \rangle = 0 \]
\[
\frac{1}{\|y - x\|} \langle y - x, y - y_0 \rangle = 0
\]
\[
\left\langle \frac{y - x}{\|y - x\|}, y - y_0 \right\rangle = 0
\]
\[
\langle \beta, y \rangle - \langle \beta, y_0 \rangle = 0
\]
\[
\langle \beta, y \rangle = \langle \beta, y_0 \rangle
\]
\[
p(y) = \langle \beta, y \rangle - c = \langle \beta, y_0 \rangle - c = p(y_0)
\]

which contradicts our assumption that \(p(y_0) < p(y)\).

2. The closest point to \(x\) in \(y\) lies outside \(y_{[0,1]}\), in \(y_{(-\infty,0)} = \{y_t : t \in (-\infty,0)\}\).

Then \(y_0\) is the next closest point in \(y_{[0,1]}\), and since the distance from \(x\) to \(y_t\) increases monotonically as \(t > s\) increases, \(y_0\) is closer to \(y\); but this contradicts our assumption that \(y - x\) was the vector of least norm in the Minkowski sum of \(B\) and \(-A\), since then \(y_0 - x\) has lesser norm.

3. The closest point to \(x\) in \(y\) lies in \(y_{(1,\infty)}\). Then since \(s > 1\) and \(p(y_s)\), the distance of \(y_t\) from \(N(p)\), increases with \(t\), we know \(p(y_s) > p(y)\). Then \(p(y_s) - p(x)\) represents the minimum distance from \(y_s\) to \(N(p)\) plus the distance from \(N(p)\) to \(x\): it is intuitively the length of the path \(x \rightarrow \pi(x) \rightarrow \pi(y_s) \rightarrow y_s\), where \(\pi(z)\) denotes the nearest point of \(N(p)\) to \(z\), with the \(\pi(x) \rightarrow \pi(y_s)\) distance omitted. Then
\[
\|y_s - x\| \geq p(y_s) - p(x) > p(y) - p(x) = \|y - x\|
\]

But this is absurd, since by assumption \(y_s\) was the closest point in a set containing \(y\). This is a contradiction.
Thus, since assuming any other possibility always leads to a contradiction, for all $y' \in B$, $p(y') \geq p(y) > 0$. By symmetry in problem (and swapping $p$ with $-p$), we also have the fact that for all $x' \in A$, $p(x') \leq p(x) < 0$. Thus,

$$
\forall a \in A, \langle \beta, a \rangle = p(a) + c < c
$$

$$
\forall b \in B, \langle \beta, b \rangle = p(b) + c > c
$$

which concludes the proof. \qed

Indeed, this proof is actually slightly stronger than the Separating Hyperplane Theorem, as the condition that $p(x') \leq p(x)$ and $p(y') \geq p(y)$ can be transformed into

$$
p(x') \leq p(x) < 0 < p(y) \leq p(y')
$$

However, since $p(y) = -p(x)$, by

$$
p(y) = \langle \beta, y \rangle - \left\langle \beta, \frac{1}{2}(x + y) \right\rangle = \left\langle \beta, y - \frac{1}{2}(x + y) \right\rangle
$$

$$
= -\left\langle \beta, x - \frac{1}{2}(x + y) \right\rangle = -\left( \langle \beta, x \rangle - \left\langle \beta, \frac{1}{2}(x + y) \right\rangle \right) = -p(x)
$$

We in fact have the case that

$$
\forall z \in A \cup B, |p(y)| \leq |p(z)|
$$

So we can more completely say that the affine hyperplane $N(p)$ separates $A$ and $B$ with a margin of $2p(y) = ||y - x||$. Since $A$ and $B$ contain elements $x$ and $y$ with precisely this distance between them, we furthermore know that $N(p)$ provides a maximum margin between the two convex sets $A$ and $B$. 
Thus, if we have a dataset such that the convex hulls of the two classes are disjoint, then there exists a hyperplane which separates these two convex hulls by a positive margin, and thus also separates the two datasets by this margin.

4.2 Theory of the support vector and support vector machine classifiers

Unfortunately, with real-world data, we cannot always expect such convenient properties as those assumed for the Separating Hyperplane Theorem (4.1.3) in the previous chapter; in large real world datasets, overlap between the convex hulls of two datasets is quite nearly inevitable. While we can utilize “tricks” to separate such datasets in nonlinear manners—for example, we can add “statistics” to the data, like mapping 
$f : \mathbb{R}^n \to \mathbb{R}^{2n}$ by $f(x_1, \ldots, x_n) = (x_1, \ldots, x_n, x_1^2, \ldots, x_n^2)$, and then attempt to separate $\{(f(x_i), c_i)\}$ in $\mathbb{R}^{2n}$ rather than the original inseparable data [11, p. 102]—this only adds so much power to the strict separating hyperplane. The support vector classifier adds some slack to the separation criteria, and thus permits classification on ordinarily inseparable datasets.

**Definition 4.2.1.** A **support vector classifier** for the dataset $\{(x_i, c_i) \in \mathbb{R}^n \times \{-1, 1\} : i \in \mathbb{N}\}$ is the classifier 

$$G(x) = \text{sgn}(\langle \beta, x \rangle + \beta_0)$$

where $\beta \in \mathbb{R}^n$ and $\beta_0 \in \mathbb{R}$ minimize $||\beta||$ while satisfying the inequality 

$$c_i (\langle \beta, x_i \rangle + \beta_0) \geq 1 - \xi_i$$  \hspace{1cm} (4.2)

for all $i \in \mathbb{N}$ and for some collection $\{\xi_i : i \in \mathbb{N}\}$ all nonnegative such that $\sum_{i \in \mathbb{N}} \xi_i \leq C$, a constant determined in advance. [11, p. 419]
Technical information describing a method for computing support vector classifiers using Lagrange multipliers may be found in Friedman et al. [11, p. 420].

Note that if we select the constant $C$ above to be 0, we arrive at a condition which is identical to requiring a strictly separating hyperplane. Then each $\xi_i = 0$. Dividing by $||\beta||$ throughout reduces condition (4.2) to the inequality

$$c_i \left( \frac{\beta}{||\beta||}, x_i \right) + \frac{\beta_0}{||\beta||} \geq \frac{1}{||\beta||}$$

or, writing $\beta^* \leftarrow \beta/||\beta||$, $\beta^*_0 \leftarrow \beta_0/||\beta||$, and $M \leftarrow 1/||\beta||$,

$$c_i (\langle \beta^*, x_i \rangle + \beta^*_0) \geq M$$

which effectively transforms the problem into a maximization problem in the margin $M$ over $\beta^* \in S^{n-1}$, $\beta^*_0 \in \mathbb{R}$. Furthermore, since $\beta^* = \beta/||\beta||$ is a vector of norm 1, the left side, less $c_i$, is a signed distance from the null space of the function $x \rightarrow \langle \beta/||\beta||, x \rangle + \beta_0/||\beta||$, much like that constructed by Theorem 4.1.3. Thus, the support vector classifier is a proper generalization of the separating hyperplane concept.

The support vector machine classifier further generalizes the support vector classifier to permit nonlinear classification implicitly utilizing high-dimensional Hilbert spaces.

**Definition 4.2.2.** A **support vector machine classifier** for the dataset $\{(x_i, c_i) \in \mathbb{R}^n \times \{-1, 1\} : i \in \mathcal{N}\}$ is the classifier

$$G(x) = \text{sgn}(f(x))$$

where $f(x)$ is a solution function of the form

$$f(x) = \sum_{i \in \mathcal{N}} \alpha_i c_i K(x_i, x) + \beta_0$$
where $K$ is a symmetric positive definite function called the “kernel” function, and $\alpha_i, i \in \mathbb{N}$, maximize the value of the expression

$$\sum_{i \in \mathbb{N}} \alpha_i - \frac{1}{2} \sum_{i \in \mathbb{N}} \sum_{j \in \mathbb{N}} \alpha_i \alpha_j c_i c_j K(x_i, x_j)$$

subject to the constraints that $\sum_{i \in \mathbb{N}} \alpha_i c_i = 0$, $0 \leq \alpha_i \leq C$ for some $\beta_0$ chosen optimally and $C$ a tuning parameter [11, p. 424].

Part of the purpose of the “kernel” function is to implement the “trick” mentioned earlier, and generalize this further. Often $K$ is of the form $K(x, x') = \langle h(x), h(x') \rangle$ for some $h$ mapping $\mathbb{R}^n$ to a higher dimensional (possibly even infinite dimensional) Hilbert space. Additional information on this can be found in Friedman et al. [11, p. 428].

The most common types of kernel functions $K$ are, according to Friedman et al. [11, p. 424],

- $K(x, x') = (1 + \langle x, x' \rangle)^d$ (d-th degree polynomial)
- $K(x, x') = \exp(-\gamma ||x - x'||^2)$ (Radial basis)
- $K(x, x') = \tanh(\kappa_1 \langle x, x' \rangle + \kappa_2)$ (Neural network (or sigmoid))

4.3 Software for SVM: LIBSVM

The support vector machine implementation we use in this research is LIBSVM, a software package produced by Chang and Lin [5] at National Taiwan University, which provides bindings to a number of programming and scripting languages, including the GNU Octave [9] bindings we utilize.
LIBSVM internally uses a sequential minimal optimization (SMO) type decomposition method; for further details see Fan et al. [10].

The LIBSVM Octave binding \texttt{svmtrain} constructs SVM models for a dataset \( \{(x_i, c_i) \in \mathbb{R}^n \times \{-1, 1\} : i \in \mathbb{N}\} \) using two derived objects, a label vector \( L \) and a sample matrix \( M \) defined by

\[
L = \begin{pmatrix} c_1 \\ c_2 \\ \vdots \\ c_N \end{pmatrix}, \quad M = \begin{pmatrix} x_1^\top \\ x_2^\top \\ \vdots \\ x_N^\top \end{pmatrix} = \begin{pmatrix} x_{11} & x_{12} & \cdots & x_{1n} \\ x_{21} & x_{22} & \cdots & x_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ x_{N1} & x_{N2} & \cdots & x_{Nn} \end{pmatrix}
\]

In particular, we utilize the C-SVC methodology with a radial kernel, following the recommendations of the LIBSVM developers, cross-validating the choice of \( C \) as part of the evolutionary algorithm procedure.
Chapter 5

Parameter selection and use of the evolutionary algorithm

While the theory of both the differential-magnitude and radius and radial Fourier descriptors suggest they may be able to extract useful information from a Landsat 8 scene, they also require a substantial amount of tuning to ensure they capture features in the data which are useful to our classification procedure. For this reason, we utilize a cellular evolutionary algorithm to explore the parameter space and find effective choices of parameters for our primary task: the recognition of data centers depicted in the scene.

5.1 Parameters for the differential-magnitude and radius descriptor

The differential-magnitude and radius descriptor inherently requires a large quantity of parameters. For each layer $I$ to be analyzed in the image, we must construct a bin configuration $(M_I, R_I)$ to describe that layer.

The sizes of $M_I$ and $R_I$ can vary for an image. However, if we assume that each layer is assigned $A$ magnitudinal and $B$ radial bins, then we would expect that our evolutionary algorithm must explore an $18(A + B)$-dimensional parameter space (with a coordinate for the left and right boundaries of each RSCI element of $M$ and $R$ on each of the 9 layers of the Landsat data). With a common value of $A = B = 3$, this implies the parameter space the evolutionary algorithm is assigned to explore possesses 108 dimensions.
5.2 Parameters for the radial Fourier descriptor

The radial Fourier descriptor, at a minimum, requires a collection of RSCI intervals $R$ which describe the annulus to be divided into sectors from which we compute the fast Fourier transforms. Furthermore, we may include within the parameters a binary 9-tuple describing exactly which Landsat layers we wish to extract descriptors from, which is about 9 binary dimensions. We also add cost constants $C_{+1}$, $C_{-1}$ for each of our two classes, and sine lengths for each layer included, providing how much of each FFT’s output to actually include in the descriptor vector before omitting the rest of the (high frequency) components.

Again, the sizes of $R_I$ can vary dramatically for an image. If we assume that $A = |R_I|$, then the evolutionary algorithm explores an $18A$-dimensional space. Results from the evolutionary algorithm suggest, however, an average of only approximately 2.7 active layers per viable RFD annuli configuration are necessary for optimal classification results, and these only need on average around 12.8 annuli per configuration, or about 4.7 annuli per layer. Thus we would anticipate that the parameter space explored by the evolutionary algorithm effectively explores a parameter space containing around 25.6 real dimensions.

Overall, the evolutionary algorithm for the radial Fourier descriptor explores a parameter space with 9 binary dimensions, 2.7 integer-valued dimensions, and approximately 27.6 continuous real-valued dimensions.
5.3 Cellular evolutionary algorithms

The particular type of evolutionary algorithm we utilize to optimize our parameter selection for our descriptors is called a \textit{cellular evolutionary algorithm} (cEA). The general concept is not entirely unlike that of a cellular automata, modified with the concept that a cell represents some element of a set in which we wish to find maximums with respect to some measure of fitness.

The particular type of cellular evolutionary algorithm we utilize is largely original but not new; we developed this technique independently, simply for the purposes of basic parameter selection, but the actual design is remarkably similar to the simplest techniques described by the Networking and Emerging Optimization research group at the University of Malaga [8]. Furthermore, our formalization of these ideas is also our own.

\textbf{Definition 5.3.1.} Suppose we have some collection $X$ and a fitness function $f : X \rightarrow \mathbb{R}$ with respect to which we wish to find maximal members of $X$. A \textbf{cellular evolutionary operator} $\text{Ev}$ on $X$ and $f$ takes a graph $(G_v, G_e)$ with vertices associated to elements of $X$ by $C_n : G_v \rightarrow X$—we call $(G_v, G_e, C_n)$ a \textbf{cell ecosystem} over $X$—to another cell ecosystem $(G_v, G_e, C_{n+1})$ where

$$C_{n+1}(u) \overset{\text{def}}{=} \text{Ev}(N(u), C_n)$$

where $N(u)$ denotes the graph neighborhood of $u$. Iterating this procedure should yield probabilistic iterative improvement in the population $C_n(G_v)$. The process, considered as a map over all possible $u \in G_v$ taking $(G_v, G_e, C_n)$ to $(G_v, G_e, C_{n+1})$, is a \textbf{cellular}
evolutionary algorithm.

Example 5.3.2. To reinforce the connection between cellular evolutionary algorithms and cellular automata, note that the famous cellular automata, Conway’s Game of Life [13], could be implemented using the definition above, disregarding the fitness function $f$. To do so, we let $X = \{0, 1\}$, $G_v = \mathbb{Z} \times \mathbb{Z}$, $G_e = \{(x, y) \in G_v \times G_v : x \neq y, |x_1 - y_1| \leq 1, |x_2 - y_2| \leq 1\}$, (that is, $\mathbb{Z}^2$ equipped with Moore neighborhoods), let $f$ be arbitrary, and let $\text{Ev}$ be defined

$$
\text{Ev}(N(v'), C) = \begin{cases} 
1 & \text{if } C(v') = 1, |\{x \in N(v') : C(v') = 1\}| \in \{2, 3\} \quad \text{(Survival)} \\
0 & \text{if } C(v') = 1, |\{x \in N(v') : C(v') = 1\}| \notin \{2, 3\} \quad \text{(Death)} \\
1 & \text{if } C(v') = 0, |\{x \in N(v') : C(v') = 1\}| = 3 \quad \text{(Birth)} \\
0 & \text{if } C(v') = 0, |\{x \in N(v') : C(v') = 1\}| \neq 3
\end{cases}
$$

For all intents and purposes, the graph associated with a cellular evolutionary algorithm will be finite, and furthermore we will exclusively use finite rectangular subsets of square lattices with edges only between von Neumann neighbors. That is, $N(v)$ is generally the set containing the four neighboring cells to $v$.

For both DMR and RFD, we utilize a fitness function based on the classification accuracies for each type of data. First of all, we compute DMRs or RFDs (respectively) for each Landsat scene sample. We then generate a collection of training/testing sample selections; these assign each sample to use as either training material, or for testing, or to simply leave that sample unused. For each of these sample selections, we compile a dataset containing all of the positive (data center labeled) training data and a random
subset of the negative (non-data center labeled) data of the same size. We construct positive and negative testing samples similarly. Then we train an SVM using the training data, and run predictions on the testing data, which yields true positive and true negative rates for that sample selection; we assign that sample selection the geometric mean of these values as its fitness. (We utilize the geometric mean of the true positive and negative rates here to punish all-positive or all-negative prediction strategies, which are assigned zero fitness by the geometric mean.) The arithmetic mean of these fitnesses for each sample selection is then considered the overall “accuracy” assigned to the bin or annuli configuration associated with that cell. This accuracy score is then multiplied by $p^n$ for $p$ slightly less than 1 and $n$ the number of samples utilized.

The parameter space $X$ our algorithm explores for the DMR is the set

$$\left[ \mathcal{P}_{\text{fin}}(\{ (a, b] : -\infty < a < b < \infty \}) \times \mathcal{P}_{\text{fin}}(\{ (a, b] : 0 \leq a < b < \infty \}) \right]^9$$

where $\mathcal{P}_{\text{fin}}(A)$ denotes the power set of $A$ restricted to its finite elements. For notation, we use $C(v)_{iM}$ to denote the collection of magnitude bins for the $i$-th layer, and $C(v)_{iR}$ to denote the collection of radial bins.

For our evolutionary operator on DMR, we utilize the following procedure.

DMR evolutionary operator pseudocode

1. Let $f$ be given.
2. Let $v, N(v)$ be the $v$ to be improved and the set of its neighbors.

---

1 Naturally, we always have more non-data center information than data centers, so we undersample the larger dataset so that it does not dominate the behavior of the SVM.

2 We indicate here band 10 by $i = 8$ and band 11 by $i = 9$ for simplicity of notation, since band 8 is redundant with 2 and 3 and is excessively large, and band 9 mostly detects clouds and has simply been omitted for that reason.
(Alternatively, for each \(v\) with neighbors \(N(v)\),)

Let \(v' = \arg \max_{u \in N(v)} f(C(u))\) be \(v\)'s fittest neighbor.

If \(f(C(v')) \leq f(C(v))\), let \(C^*(v) \leftarrow C(v)\). (End procedure.)

For \(i \in \{1, \ldots, 9\}\),

For \(\beta \in \{M, R\}\),

Select one of the cases below at random:

A) For random \((a_j, b_j), (a_k, b_k) \in C(v')_{i\beta}\) and \(t\) sampled uniformly from \([0, 1]\), add \(((1-t)a_j + ta_k, (1-t)b_j + tb_k)\) to \(C(v')_{i\beta}\) to create \(C^*(v)_{i\beta}\).

D) For random \((a_j, b_j) \in C(v')_{i\beta}\), create \(C^*(v)_{i\beta}\) by deleting this element.

T) For each \((a_j, b_j) \in C(v')_{i\beta}\), add \((a_j + a_\epsilon, b_j + b_\epsilon)\) to \(C^*(v)_{i\beta}\), where \(a_\epsilon\) and \(b_\epsilon\) are sampled from a normal distribution.

The evolutionary algorithm above has, itself, a few parameters which may be tweaked.

The relative probabilities of adding (A), deleting (D), or translating (T) the bins in \(X_{iM}\) and \(X_{iR}\) can be tuned, as also can be the parameters of the normal distribution in the translation; a few of our models have defined these parameters as functions on \(G_v\), assigning different evolutionary behavior to different regions of \(G\), but little improvement was noted in this case. Instead, our algorithm has defined the normal distribution parameters as a decreasing function of \(f(C(v'))\), so that fitter cells tend to exhibit smaller mutations, to minimize the probability of accidentally “overshooting” the region of improvement (Figure 5.1).

For the radial Fourier descriptor, the parameter space we explore is slightly extended, to

\[
\mathcal{P}_{\text{fin}}(\{(a, b) : 0 \leq a < b < \infty\}) \times \mathbb{N} \times \mathbb{Z}/2\mathbb{Z}^9 \times \mathbb{R} \times \mathbb{R}
\]
Figure 5.1: When near a local maximum, the evolutionary improvement region shrinks substantially. To increase the probability of improvement near maxima while maintaining rapid navigation of the space while far from maxima, the variance in our normally sampled mutations is a decreasing function of fitness. Note that we denote the probability density function of the evolution operator at the state $x_0$ with $p_{x_0}$.

That is, for each choice of parameter we associate a collection of annuli radii ($C(v)_{iA}$ for the annuli of the $i$-th bin at the cell $v$) for which to compute the radial Fourier descriptor, a quantity of these bins to take from each of these descriptors ($C(v)_{iN}$), and a toggle to turn on or off the layer entirely in the descriptor ($C(v)_{iT}$). In addition we add two SVM cost values ($C(v)_+$ and $C(v)_-$) for each of the two classes of data we work with; these are similar to the cost parameter $C$ described in Definition 4.2.1, but with this cost allowed to vary between the two classes of data, refining the boundary of the classification. So to compute the next value $C^*(v)$ of a cell $v$, we utilize the following procedure:

RFD evolutionary operator pseudocode

1  Let $f$ be given.
Let $v$, $N(v)$ be the $v$ to be improved and the set of its neighbors. (Alternatively, for each $v$ with neighbors $N(v)$,)

Let $v' = \arg \max_{u \in N(v)} f(C(u))$ be $v$'s fittest neighbor.

If $f(C(v')) \leq f(C(v))$, let $C^*(v) \leftarrow C(v)$. (End procedure.)

For $i \in \{1, \ldots, 9\}$,

Select one of the cases below at random:

A) For random $(a_j, b_j), (a_k, b_k) \in C(v')_i\alpha$ and $t$ sampled uniformly from $[0,1]$, add $((1-t)a_j + ta_k, (1-t)b_j + tb_k)$ to $C(v')_i\alpha$ to create $C^*(v)_i\alpha$.

D) For random $(a_j, b_j) \in C(v')_i\alpha$, create $C^*(v)_i\alpha$ by deleting this element.

T) For each $(a_j, b_j) \in C(v')_i\alpha$, add $(a_j + a_\epsilon, b_j + b_\epsilon)$ to $C^*(v)_i\alpha$, where $a_\epsilon$ and $b_\epsilon$ are sampled from a normal distribution.

$C^*(v)_iN \leftarrow \max(C(v')_iN + \epsilon, 2)$, where $\epsilon$ is distributed normal and then rounded towards 0.

Probabilistically toggle $C^*(v)_iT \leftarrow 1 - C^*(v')_iT$.

$C^*(v)_+ \leftarrow C(v')_+ + \epsilon$, $\epsilon$ distributed normal.

$C^*(v)_- \leftarrow C(v')_- + \epsilon$, $\epsilon$ distributed normal.

It is worth noting here that for both of the algorithms we have described, we have implemented a feature of problem-oriented evolutionary algorithms known as elitism. An evolutionary algorithm which exhibits elitism is allowed to retain perfectly, without mutation, the fittest species it has yet discovered. It is an intrinsically unrealistic assumption for attempts to model actual biological evolution, but in practice elitism is often necessary to prevent extinctions. This trait is depicted on both lines marked 5 in the DMR and RFD evolutionary operator descriptions above, which relate the fact that
in both algorithms we retain elitism for a cell within its neighborhood. Since the number of cells is intrinsically finite, some cell is maximal in general and in its neighborhood, and thus the fittest solution is always preserved.

5.4 Exploring the cellular evolutionary algorithm

To test whether we can expect the cellular evolutionary algorithm to perform well for our purposes, we utilize a simple model which solves a similar problem to that of the support vector machine from before, separating a dataset of two classes.

Two collections of 10000 points in \( \mathbb{R}^{108} \) were generated in two classes, \(+1\) and \(-1\), which were distributed by 108-dimensional multivariate normal\(^3\) distributions \( N(p, I_{108}) \) and \( N(n, I_{108}) \) respectively, where \( p = (1, 0, 0, \ldots, 0) \) and \( n = (-1, 0, 0, \ldots, 0) \). This composes the dataset \( D_{108} \subseteq \mathbb{R}^{108} \times \{ -1, +1 \} \). These were then projected down into \( \mathbb{R}^{26} \) to construct \( D_{26} \) omitting the last 82 coordinates; 26 and 108 dimensional data were chosen in order to correspond to the approximate dimension of the parameter space for the RFD and DMR procedures.

We then utilize a cellular evolutionary algorithm to obtain \( \beta \in S^{n-1}, \beta_0 \in \mathbb{R}, n = 26 \) or \( 108 \), which optimize the geometric means of the classification accuracies for \(-1\) and \(+1\) by the classifier

\[
C(x) = \text{sgn} \left( \langle \beta, x \rangle - \beta_0 \right).
\]

So our fitness function \( f \) is

\[
f(\beta, \beta_0) = \sqrt{\left( \frac{|\{(c, x) \in D_n : C(x) = c \} \cap [-1]|}{|[-1]|} \right) \left( \frac{|\{(c, x) \in D_n : C(x) = c \} \cap [+1]|}{|[+1]|} \right)}
\]

\(^3\)More specifically, each component of each data point was sampled (univariate) normal \( N(0, 1) \) and then 1 was added or subtracted from the first entry in the vector.
where \([c] \) for \(c \in \{-1, +1\} \) denotes the subset of \(D_n \) of class \(c \) as in chapter 2. Thus, in fact, this evolutionary algorithm explores \(S^{n-1} \times \mathbb{R} \); as the product space of a \(n-1\)-manifold and an \(n\)-manifold, this is still an \(n\)-manifold parameter space, and thus we still expect it to model the problem well while the simplicity of the problem makes it amenable to rapid computation and testing.

It is notable that the theoretical answer to this problem is relatively obvious. One would expect that \(\beta = (1, 0 , \cdots , 0) \) and \(\beta_0 = 0 \) would be the correct solution, since this would place a plane directly between the means of the two class distributions. We note that for our data, this yields a fitness of 0.83880.

We applied a cellular evolutionary algorithm to a randomized initial array of samples from \(S^{n-1} \times \mathbb{R} \) of varying sizes. For \(D_{26} \) we utilized 30 seconds of continuous computation, and for \(D_{108} \) we allowed 60 seconds to compensate for the more intensive computations necessary. Our evolutionary operator modified

\[
Ev : (\beta, \beta_0) \mapsto \left( \frac{\beta + \beta^*}{||\beta + \beta^*||}, \beta_0 + \beta_0^* \right)
\]

where \(\beta^* \) is distributed multivariate normal \(N(0, 0.05(1 - f(\beta, \beta_0))I_n)\) and \(\beta_0^* \) distributed \(N(0, (1 - f(\beta, \beta_0)))\). That is, the variance was reduced as a decreasing function of the fitness \(f \), as mentioned in the previous chapter, to refine evolution near the maxima.

Figure 5.2 demonstrates the results of this simple test case. In both cases, convergence to the predicted global maximum value of 0.83880 at \(((1, 0 , \cdots , 0), 0)\) was approached quite rapidly. Indeed, in the \(4 \times 4 \) \(D_{26} \) case, this theoretical global maximum was even surpassed, as seen in Table 5.1, by just short of 0.01. Despite the fact that the separation problem is intrinsically made more difficult by projecting the data
Figure 5.2: Charts for the evolutionary algorithm with the semiseparable 26 and 108 dimensional data $D_{26}$ and $D_{108}$, for cell arrays of sizes $4 \times 4$, $8 \times 8$, and $12 \times 12$. The procedure was given 30 seconds for $D_{26}$ and 60 seconds for $D_{108}$. 
<table>
<thead>
<tr>
<th>Data set</th>
<th>Array size</th>
<th>Time (s)</th>
<th>Gens.</th>
<th>Maximal fitness</th>
</tr>
</thead>
<tbody>
<tr>
<td>$D_{26}$</td>
<td>4 $\times$ 4</td>
<td>30</td>
<td>459</td>
<td>0.8418475218232813</td>
</tr>
<tr>
<td>$D_{26}$</td>
<td>8 $\times$ 8</td>
<td>30</td>
<td>114</td>
<td>0.8337412248413772</td>
</tr>
<tr>
<td>$D_{26}$</td>
<td>12 $\times$ 12</td>
<td>30</td>
<td>54</td>
<td>0.815771938464570</td>
</tr>
<tr>
<td>$D_{108}$</td>
<td>4 $\times$ 4</td>
<td>60</td>
<td>236</td>
<td>0.816695409501866</td>
</tr>
<tr>
<td>$D_{108}$</td>
<td>8 $\times$ 8</td>
<td>60</td>
<td>60</td>
<td>0.7581429614525218</td>
</tr>
<tr>
<td>$D_{108}$</td>
<td>12 $\times$ 12</td>
<td>60</td>
<td>27</td>
<td>0.709295936471969</td>
</tr>
<tr>
<td>$D^*_{26}$</td>
<td>4 $\times$ 4</td>
<td>30</td>
<td>313</td>
<td>0.689258332829572</td>
</tr>
<tr>
<td>$D^*_{26}$</td>
<td>6 $\times$ 6</td>
<td>30</td>
<td>146</td>
<td>0.6971169916161849</td>
</tr>
<tr>
<td>$D^*_{26}$</td>
<td>8 $\times$ 8</td>
<td>30</td>
<td>124</td>
<td>0.7047089612031339</td>
</tr>
<tr>
<td>$D^*_{26}$</td>
<td>10 $\times$ 10</td>
<td>30</td>
<td>80</td>
<td>0.691563301513177</td>
</tr>
<tr>
<td>$D^*_{26}$</td>
<td>12 $\times$ 12</td>
<td>30</td>
<td>58</td>
<td>0.6914053442084462</td>
</tr>
<tr>
<td>$D^*_{108}$</td>
<td>4 $\times$ 4</td>
<td>60</td>
<td>270</td>
<td>0.8358432867469834</td>
</tr>
<tr>
<td>$D^*_{108}$</td>
<td>6 $\times$ 6</td>
<td>60</td>
<td>128</td>
<td>0.789573811630104</td>
</tr>
<tr>
<td>$D^*_{108}$</td>
<td>8 $\times$ 8</td>
<td>60</td>
<td>69</td>
<td>0.7686593979130159</td>
</tr>
<tr>
<td>$D^*_{108}$</td>
<td>10 $\times$ 10</td>
<td>60</td>
<td>41</td>
<td>0.749539588467460</td>
</tr>
<tr>
<td>$D^*_{108}$</td>
<td>12 $\times$ 12</td>
<td>60</td>
<td>31</td>
<td>0.712908581298454</td>
</tr>
</tbody>
</table>

Table 5.1: Each of the dataset and array size combinations yielded different fitness convergence properties. Note that these numbers are the result of a randomized process and are only to be taken as general behavior trends. A particularly interesting example of this is depicted in figure 5.3.

into a lower dimensional space, since the preimage of any affine hyperplane under the projection will itself be an affine hyperplane acting to separate the higher dimensional space and thus the lower dimensional separation problem can be viewed as a subcase of the higher dimensional one, higher fitness is achieved in the problem of separating $D_{26}$ than $D_{108}$; this is an instance of the curse of dimensionality exhibited in the parameter space. Furthermore, only minor drops in fitness are observed with increases of array size, which themselves cause dramatic drops in generation count, hinting that the diversification of the ecosystem induced by a larger array can ameliorate the advantages of an 8-9 fold increase in generation count.

There is, however, some weakness in this test. While it is desirable to have the
Figure 5.3: If one of the randomized initial bins happens to be dramatically better than the rest, then it is possible for its descendents to take over the entire cell array rather rapidly. (Computation on $D_{108}^*$ with $10 \times 10$ array.)

to make for data as complex as our data center problem. Furthermore, the projection of $D_{108}$ onto $D_{26}$ is quite clean; it does not combine different clusters, as we might expect in a more complex problem.

To answer these concerns, we construct $D_{26}^*$ and $D_{108}^*$. To construct these, first we construct 100 points for each class $C^+$ and $C^-$ distributed $N(0, 3I_{108})$. Around each of these points $p$ we generate 100 points distributed $N(p, I_{108})$, and these form our $[+1]$ and $[-1]$ datasets. The result is an emulation of a sampling of a multivariate normal distribution with standard variance and mean selected from $C^+$ ($C^-$) uniformly, with the additional condition that each mean must be selected equitably. We then project this data into $\mathbb{R}^{26}$ as before to form $D_{26}^*$. 
Unlike in the previous case, no obvious hyperplane exists in this case for separating even the centers of the constituent normal distributions, much less the two collections of 10000 points. This yields a more interesting problem which is not necessarily “locally linear,” in the sense that an evolutionary neighborhood around $\left( \beta, \beta_0 \right)$ may contain multiple disjoint connected components fitter than $\left( \beta, \beta_0 \right)$ which may lead to entirely different local maxima.

The results of this test are found in Figure 5.4 and the latter half of Table 5.1. In this case it seems that a 8x8 cell array achieves the most efficient separation of $D^{*}_{26}$, rather than larger or smaller cell arrays; we conclude that for this particular problem this population size finds strength in the diversity it permits without overspending computational resources supporting too many weak strains of parameter. This diversity proves essential to navigating a complex parameter space which potentially has many different local maxima. A larger cell array can support a greater variety of strains which are maximums in their cellular neighborhood, and also permits the development of longer evolution sequences. For a $m \times n$ array we expect no more than $m + n - 2$ consecutive nonimprovements (the graph distance between the two corners of the array) before a strain is driven to extinction; if these mutations are not all deleterious, it is possible that this sequence of steps has navigated into another evolutionary basin of attraction, which may eventually lead to higher fitness than the original species. In this manner the size of the cell array contributes to the ability of the population to speciate.

Figure 5.5 compares the maximum fitness obtained during each of these computations. From this data we can make a few conclusions. We observe that the semiseparable
Figure 5.4: Charts for the evolutionary algorithm with the mixed 26 and 108 dimensional data $D^*_26$ and $D^*_108$, for cell arrays of sizes $4 \times 4$, $8 \times 8$, and $12 \times 12$. The procedure was given 30 seconds for $D^*_26$ and 60 seconds for $D^*_108$. 
classification is made easier by its projection onto lower dimensions, whereas the mixed data classification is rendered substantially more difficult as different classes of data are mixed together by the projection.

An additional benefit to larger cell arrays, though we have not seriously explored this particular approach, is the improved ability to parallelize the fitness computation. Theoretically, each of the cell fitnesses can be computed in parallel, which permits a much broader parameter space to be explored at approximately the same speed despite the increased size of the array.

Figure 5.5: The maximum fitness discovered, by generation, for each of our combinations of initial cell array size and dataset.
As an aside, we note some concerns regarding cell arrays of size less than $4 \times 4$. Such small cell arrays tend to permit degenerate population patterns, which prevent the development of a diverse collection of local maximum cells benefitting from the elitism intrinsic to the cellular evolutionary algorithm. In the $2 \times 2$ case only two opposite maximal cells can coexist, and whichever of these is inferior will receive no descendant cells. In the $3 \times 3$ case five maximal cells may coexist, occupying the corners and center of the array, but if the center is fittest (as we would expect to occur often), then no other cell may obtain descendants, so we would expect this to disproportionately favor the evolutionary strain of the center cell, since it will certainly destroy at least half of the strains represented by other maximal cells. Consequently, we should be careful using a cell array smaller than $4 \times 4$. 
Chapter 6

Methodology, results and conclusions

6.1 Landsat data obtained

To select parameters for the DMR, and RFD methods, we collected 79 Landsat 8 data scenes, shown in Table 6.1. These comprised 169 GiB of data, representing six different data centers and one scene representing several smaller data centers. These included Facebook’s data center in Altoona, IA (near Des Moines); a collection of smaller data centers in Charlotte, NC; Microsoft’s data center in San Antonio, TX; and Google’s data centers in Hamina, Finland, Mayes County, OK, Moncks Corner, SC, and The Dalles, OR. The climate of each data center’s region affected the availability of Landsat 8 data dramatically. Since the Landsat 8 satellite was launched on February 11, 2013 [17], Google’s data center in Hamina, Finland has only been visible in exactly one Landsat scene; it is almost always obfuscated by clouds. In contrast, Google’s data center in The Dalles, OR was visible in at least 22 of the (currently) 66 different Landsat 8 scenes available\(^1\).

Of these, a subset of six scenes were selected for use in with DMR, due to the substantially more complex computations necessary for that method; these are indicated by asterisks in Table 6.1. Furthermore, several scenes representing Altoona represent the site of Facebook’s data center before the actual data center was built or complete. These are indicated with daggers.

---

\(^1\)Due to an anomaly with the Landsat 8 satellite which occurred around October 2015, the thermal bands were zero-filled and thus unsuitable for use with this technique. As such these more recent scenes were omitted and in fact the 22 selected were an even greater proportion of the actually viable scenes.
<table>
<thead>
<tr>
<th>Site</th>
<th>Scene ID</th>
<th>Site</th>
<th>Scene ID</th>
</tr>
</thead>
<tbody>
<tr>
<td>Altoona†</td>
<td>LC80260312013252LGN00</td>
<td>Altoona†</td>
<td>LC80260312013268LGN00</td>
</tr>
<tr>
<td>Altoona†</td>
<td>LC80260312013300LGN00</td>
<td>Altoona†</td>
<td>LC80260312014063LGN00</td>
</tr>
<tr>
<td>Altoona†</td>
<td>LC80260312014079LGN00</td>
<td>Altoona†</td>
<td>LC80260312015066LGN00</td>
</tr>
<tr>
<td>Altoona†</td>
<td>LC80260312015258LGN00</td>
<td>Altoona†</td>
<td>LC80260312014080LGN00</td>
</tr>
<tr>
<td>Charlotte</td>
<td>LC80170362013317LGN00</td>
<td>Charlotte</td>
<td>LC80170362013333LGN00</td>
</tr>
<tr>
<td>Charlotte</td>
<td>LC80170362014016LGN00</td>
<td>Charlotte</td>
<td>LC80170362014080LGN00</td>
</tr>
<tr>
<td>Charlotte</td>
<td>LC80170362014070LGN00</td>
<td>Charlotte</td>
<td>LC80170362014240LGN00</td>
</tr>
<tr>
<td>Charlotte</td>
<td>LC80170362015067LGN00</td>
<td>Charlotte</td>
<td>LC80170362015067LGN00</td>
</tr>
<tr>
<td>Charlotte</td>
<td>LC80170362015291LGN00</td>
<td>Hamina†</td>
<td>LC8160182013237LGN00</td>
</tr>
<tr>
<td>MayesCounty</td>
<td>LC80270352013151LGN01</td>
<td>MayesCounty</td>
<td>LC80270352013179LGN00</td>
</tr>
<tr>
<td>MayesCounty</td>
<td>LC80270352013163LGN00</td>
<td>MayesCounty</td>
<td>LC80270352014006LGN00</td>
</tr>
<tr>
<td>MayesCounty†</td>
<td>LC80270352014022LGN00</td>
<td>MayesCounty</td>
<td>LC80270352014310LGN00</td>
</tr>
<tr>
<td>MayesCounty</td>
<td>LC80270352014118LGN00</td>
<td>MayesCounty</td>
<td>LC80270352015089LGN00</td>
</tr>
<tr>
<td>MayesCounty</td>
<td>LC80270352015041LGN01</td>
<td>MayesCounty</td>
<td>LC80270352015249LGN00</td>
</tr>
<tr>
<td>MayesCounty†</td>
<td>LC80270352015281LGN00</td>
<td>MoncksCorner</td>
<td>LC80160372013134LGN03</td>
</tr>
<tr>
<td>MoncksCorner</td>
<td>LC80160372013198LGN00</td>
<td>MoncksCorner</td>
<td>LC80160372013278LGN00</td>
</tr>
<tr>
<td>MoncksCorner</td>
<td>LC80160372013558LGN00</td>
<td>MoncksCorner</td>
<td>LC80160372014073LGN00</td>
</tr>
<tr>
<td>MoncksCorner</td>
<td>LC80160372014137LGN00</td>
<td>MoncksCorner</td>
<td>LC80160372014233LGN00</td>
</tr>
<tr>
<td>MoncksCorner†</td>
<td>LC80160372014281LGN00</td>
<td>MoncksCorner</td>
<td>LC80160372014345LGN00</td>
</tr>
<tr>
<td>MoncksCorner†</td>
<td>LC80160372014361LGN00</td>
<td>MoncksCorner</td>
<td>LC80160372015028LGN00</td>
</tr>
<tr>
<td>SanAntonio</td>
<td>LC80270402013243LGN00</td>
<td>SanAntonio</td>
<td>LC80270402013307LGN00</td>
</tr>
<tr>
<td>SanAntonio</td>
<td>LC80270402014011LGN00</td>
<td>SanAntonio</td>
<td>LC80270402014134LGN00</td>
</tr>
<tr>
<td>SanAntonio</td>
<td>LC80270402014182LGN00</td>
<td>SanAntonio</td>
<td>LC80270402014294LGN00</td>
</tr>
<tr>
<td>SanAntonio</td>
<td>LC80270402014358LGN00</td>
<td>SanAntonio</td>
<td>LC80270402015041LGN01</td>
</tr>
<tr>
<td>SanAntonio</td>
<td>LC80270402015073LGN00</td>
<td>SanAntonio</td>
<td>LC80270402015121LGN00</td>
</tr>
<tr>
<td>SanAntonio†</td>
<td>LC80270402015265LGN00</td>
<td>TheDalles</td>
<td>LC80450282013113LGN01</td>
</tr>
<tr>
<td>TheDalles</td>
<td>LC80450282013161LGN00</td>
<td>TheDalles</td>
<td>LC80450282013209LGN00</td>
</tr>
<tr>
<td>TheDalles</td>
<td>LC80450282013225LGN00</td>
<td>TheDalles</td>
<td>LC80450282013257LGN00</td>
</tr>
<tr>
<td>TheDalles</td>
<td>LC80450282013289LGN00</td>
<td>TheDalles</td>
<td>LC80450282013305LGN00</td>
</tr>
<tr>
<td>TheDalles</td>
<td>LC80450282014004LGN00</td>
<td>TheDalles</td>
<td>LC80450282014036LGN00</td>
</tr>
<tr>
<td>TheDalles</td>
<td>LC80450282014100LGN00</td>
<td>TheDalles</td>
<td>LC80450282014132LGN00</td>
</tr>
<tr>
<td>TheDalles</td>
<td>LC80450282014196LGN00</td>
<td>TheDalles</td>
<td>LC80450282014228LGN00</td>
</tr>
<tr>
<td>TheDalles</td>
<td>LC80450282014244LGN00</td>
<td>TheDalles</td>
<td>LC80450282014276LGN00</td>
</tr>
<tr>
<td>TheDalles</td>
<td>LC80450282015055LGN00</td>
<td>TheDalles</td>
<td>LC80450282015167LGN00</td>
</tr>
<tr>
<td>TheDalles</td>
<td>LC80450282015183LGN00</td>
<td>TheDalles</td>
<td>LC80450282015199LGN00</td>
</tr>
<tr>
<td>TheDalles†</td>
<td>LC80450282015231LGN00</td>
<td>TheDalles</td>
<td>LC80450282015263LGN00</td>
</tr>
<tr>
<td>TheDalles†</td>
<td>LC80450282015295LGN00</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 6.1: A table of the training data’s scene IDs. The scene ID is formatted in the form “L X S PPP RRR YYYY DDD GSI VV”, where L indicates that the data is Landsat, X denotes which sensors are used (C is OLI/TIRS), S denotes which Landsat satellite was used (8), PPP represents the Worldwide Reference System (WRS) path, RRR represents the WRS row, YYYY represents the year, DDD represents the day of year, GSI is the ground station identifier, and VV is the archive version number.
6.2 Hardware utilized

The evolutionary algorithm itself was implemented primarily on the Silvertip2 research server provided by South Dakota State University through University Networking Systems & Services. This computer was an IBM x3755 M3 equipped with 64 processor cores in four AMD Opteron 6282SE 16 core 2.6GHz processors, with 512 GB of RAM and running OpenSUSE Linux.

For the DMR procedure, we explicitly parallelized the DMR computation to utilize at most 16 of the machine’s cores at once. For RFD, the resource allocation associated with the problem was assigned primarily to the efficient fast Fourier transform implementation built into GNU Octave, which was provided by FFTW [12]; this software was in turn able to parallelize the computation efficiently.

6.3 Differential-magnitude and radius descriptor results

After seven iterations of the evolutionary algorithm optimizing DMR bin configurations, a collection of fairly fit bins were obtained.

The DMR descriptor calculations proved to be relatively costly. For this reason, the evolutionary algorithm was trained and tested with only six data samples. A 3 × 3 array was used despite the concerns mentioned previously in 5.4, for the sake of rapid computation.

The optimized DMR parameters are depicted in a scatter plot in Figure 6.1. Each point has coordinates which are the arithmetic means of the true positive (negative,
respectively) rates over a number of samplings; each sampling omitted from training the data from exactly one of the six Landsat 8 scenes, and tested against the omitted scene.

6.4 Radial Fourier descriptor results

After a few months of evolutionary algorithm work developing annuli configura-
tions (approximately January 2016 to late March 2016), a collection of optimized bins

Figure 6.1: True positive and negative rates for the optimized DMR configurations. Each mark indicates the arithmetic mean of a collection of samplings associated with a bin configuration. The dashed lines indicate contours of the fitness function for each configuration.
were constructed. We began with a $16 \times 16$ cell array, but eventually swapped to a $4 \times 4$ cell array, since it was the smallest square option which did not seem to induce any obvious suboptimal behavior by the evolutionary algorithm, as mentioned in section 5.4. Computing the fitness of a particular cell required approximately 3662 seconds, or approximately 1.017 hours. Thus we would expect that one iteration of the evolutionary algorithm would take approximately 16 hours.

The optimized parameters achieved a fitness of 0.83398, which was penalized down from an accuracy\(^2\) of 0.86284. Samplings associated with the optimal annuli configuration are indicated with blue marks in Figure 6.2. To construct that scatter plot, we utilized the results of the evolutionary algorithm described in Chapter 5. At each step of the evolutionary algorithm, we chose 10 different “samplings”, which randomly partitioned the Landsat data described above into collections of “training,” “testing,” and “unused” scenes. For each of these samples, we required that at least thirty training scenes were selected, and at least five testing scenes were selected. For all of the data, we computed RFD descriptors using the parameters under inspection, and then for each of the samplings chosen we trained an SVM using the training data, tested it against the testing data, and utilized these results to compute true positive and true negative accuracies for each sampling.

The distribution of the sample fitnesses in Figure 6.2 yields some confidence that the evolutionary algorithm has reached a local optimum of some sort; the samplings

\(^2\)Recall that by “accuracy” we refer to the arithmetic means for many samplings of the geometric means of the correctly identified class +1 vectors and correctly identified class −1 vectors for a given scene; the fitness itself is the accuracy penalized for utilizing an excess of annuli.
associated with the optimal configuration are not obviously distributed with substantially greater fitness than most of those associated with other annuli configurations; the obvious exception are those parameters at \((1, 0)\), which demonstrate an important fact about the RFD descriptor: it is not completely stable, and as in biological evolution mutations are often completely lethal to the cell.

Figure 6.2: The true positive and negative rates associated with the SVMs trained from the data for two classes. Each of the marks indicates a sampling from the data for some annuli configuration. The samplings associated with the optimal annuli configuration are in blue. The dashed lines are the contours of the per-sampling fitness function, to yield an idea of which samplings demonstrated comparable prediction accuracy.
6.5 Conclusions

To finally compare the efficacy and efficiency of the differential-magnitude and radius descriptor and the radial Fourier descriptor, we refer to Figures 6.3 and 6.4. We state a few caveats first, however.

Note that the problems faced by DMR and RFD are substantially different. The DMR testing procedure consistently trained less than six Landsat scenes against a randomly selected subset of the remainder, while RFD trained at least thirty scenes against at least five testing scenes. As such, RFD was able to accommodate substantially more information about the problem, but was generally tested against a far more diverse dataset as well. While we hope this advantage to be ameliorated by the averaging utilized over a collection of samples, it is likely that the RFD has a substantial advantage. This advantage is, however, well-earned by its higher efficiency.

From Figure 6.4, it seems evident that generally speaking, RFD operated better than DMR. RFD generally produced fairly even classification rates, with approximately 0.86 true positive and true negative classification rates. On the other hand, DMR tended to favor negative classifications, generally achieving approximately a 0.90 true negative classification rate but only approximately an average 0.75 true positive classification rate. In terms of accuracy, it seems that RFD is the better descriptor.

In terms of efficiency, RFD is clearly the better option, as was detailed in Sections 2.4 and the latter half of 3.2. Furthermore, the capability of the RFD evolutionary algorithm to optimize to compete with DMR’s several month long evolutionary lead
Figure 6.3: A plot of the true positive and true negative accuracies for the optimized DMR and RFD cell arrays by sampling.
Figure 6.4: A plot of the mean true positive and true negative accuracies for the optimized DMR and RFD cell arrays.
despite handling twelve to thirteen times more data testifies to the dramatic superiority of RFD over DMR in terms of efficiency.

Overall, the radial Fourier descriptor seems entirely better suited for SVM-based satellite imagery data center identification than the differential-magnitude and radius descriptor.
REFERENCES


INDEX

Landsat 8 imagery, 1, 2, 17, 22, 25, 53, 54, 56, 70, 73, 74, 79
LIBSVM, 51, 52
Lyapunov exponent, 13, 13, 15–17
magnitude and radius descriptor, 8, 15, 17
discrete, 19
general, 8, 13
measurable function, 5, 7, 13, 36
measurable space, 5, 7
measure space, 7, 17

Nearest Point Theorem, 43, 46
radial Fourier descriptor, 25, 27, 28, 33, 35, 38, 39, 54, 56, 58–61, 74–79
applied, 70
representative function, 35, 35, 36, 38
right semi-closed interval, 6, 6, 20, 53, 54

Separating Hyperplane Theorem, 43, 45, 48, 49
Stone-Weierstrass Theorem, 26
support vector classifier, 49, 50
support vector machine classifier, 39, 40, 50, 51, 57, 59, 61, 74, 79
C-SVC, 52

trigonometric polynomial, 26

\( \sigma \)-field, 4, 5, 6

bin configuration, 7, 8, 10, 11, 21, 53, 57, 72, 73
Borel measure, 6
Borel sets, vi, 5, 6, 7, 19
cell ecosystem, 56, 64
cellular evolutionary algorithm, 53–55, 56, 56, 58, 60–63, 65, 67, 69, 72–74, 76
cellular evolutionary operator, 56, 57, 59, 62
curse of dimensionality, 64
data center, 2, 28, 39, 53, 57, 70, 79
differential-magnitude and radius descriptor, 4, 9, 10, 12, 17, 19, 20, 22, 24, 25, 53, 56, 57, 60, 61, 72, 73, 76–79
applied, 70, 72
continuous, 8, 13
discrete, 19
evolutionary algorithm, 52, 55
Fourier coefficients, 25, 26, 26–28, 36
Fourier transform, 25–27
discrete, 25, 28, 29, 30
properties, 29
fast Fourier transform, 25, 30, 32, 33, 54
Gamma specification, 7, 7, 8, 15, 19
GNU Octave, 17, 51, 52

Intermediate Value Theorem, 45

kernel function (SVM), 51
neural network, 51
polynomial, 51
radial, 51, 52
sigmoid, 51