

NEW METHODS OF SPECTRAL-DENSITY BASED GRAPH CONSTRUCTION
AND THEIR APPLICATION TO HYPERSPSCTRAL IMAGE ANALYSIS

by

Jeffrey Stevens
A Dissertation
Submitted to the
Graduate Faculty
of
George Mason University
in Partial Fulfillment of
The Requirements for the Degree
of
Doctor of Philosophy
Computational Sciences and Informatics

Committee:

Dr. John J. Qu, Dissertation Director

Dr. Ronald G. Resmini, Committee
Member

Dr. Juan R. Cebal, Committee Member

Dr. Robert S. Weigel, Committee Member

Dr. Jason Kinser, Acting Director,
Department of Computational and Data
Sciences

Dr. Donna M. Fox, Associate Dean, Office
of Student Affairs & Special Programs,
College of Science

Dr. Peggy Agouris, Dean, College of
Science

Date: _____

Fall Semester 2017
George Mason University
Fairfax, VA

New Methods of Spectral-Density Based Graph Construction and Their Application to
Hyperspectral Image Analysis

A Dissertation submitted in partial fulfillment of the requirements for the degree of
Doctor of Philosophy at George Mason University

by

Jeffrey Stevens
Master of Science
State University of New York at Buffalo, 1994
Bachelor of Science
State University of New York at Buffalo, 1992

Director: John J. Qu, Professor
Department of Computational Sciences and Informatics

Fall Semester 2017
George Mason University
Fairfax, VA

Copyright 2017 Jeffrey Stevens
All Rights Reserved

DEDICATION

To my loving wife Monica and my children: Jared, Chris, Marissa, Lauren, and Ryan.

ACKNOWLEDGEMENTS

First, thank you to my committee, Dr. Qu, Dr. Resmini, Dr. Weigel, and Dr. Cebral; your collective contributions and guidance helped make this possible. I want to express my deepest appreciation to Dr. Resmini for pushing me through those initial months and always being available as a sounding board throughout this process. Additionally, I'd like to offer my sincere appreciation to Dr. David Messinger (Rochester Institute of Technology) and Dr. Arthur Flexer (Austrian Research Institute for Artificial Intelligence) who, despite being affiliated with other institutions, made time to answer many questions along the way. It has truly been an honor.

I need to recognize some exceptional researchers of whom I've had the privilege of reviewing their articles, research, and instructional material: Dr. Nathan Cahill (Rochester Institute of Technology), Dr. Ulrike von Luxburg (University of Tübingen), and Dr. Jean Gallier (University of Pennsylvania). Though we never met or spoke, your aptitude for research, and more importantly your incredible drive to teach, not just through academic papers, but by posting example code to promote deeper understanding as well as taking the time to document that extra step in a proof helped me through some difficult times. I am truly humbled.

Last but not least, thank you to my family and friends for your support and encouragement throughout this endeavor. It meant the world to me.

TABLE OF CONTENTS

	Page
List of Tables	viii
List of Figures	ix
List of Equations	xii
List of Abbreviations	xiv
List of Symbols	xvi
Abstract	xviii
1 Introduction	1
1.1 Hyperspectral Imaging	3
1.2 Hyperspectral Data Analysis	5
1.2.1 Common Perspectives on Spectra	6
1.2.2 Hyperspectral Data Models	11
1.3 Graphs as a Means for HSI Data Analysis	14
1.4 Research Objectives and Contributions	25
1.4.1 Survey spectral-graph construction methods	27
1.4.2 Develop new spectral-graph construction techniques	27
1.4.3 Study Impacts of High-Dimensional Data	28
1.4.4 Develop Classification and Anomaly Detection Algorithms	28
1.5 Dissertation Organization	29
2 Common HSI Graph Construction Techniques	30
2.1 ϵ -Threshold Graphs (aka ϵ -NN Graphs)	30
2.2 k -Nearest Neighbor (k -NN) Graphs	31
2.3 Density Weighted k -NN (DW k -NN) Graphs	33
2.4 Adaptive Nearest-Neighbor (ANN) Graphs	36
2.5 Edge Reweighting	39
2.6 Ensuring Graph Connectivity	43
2.7 Spatial-Spectral Methods	45
2.7.1 Forcing relationships between proximal neighbors	46

2.7.2	Modifying edge weights	48
2.7.3	Fusing information while remapping the data	52
3	New Methods of HSI Graph Construction	54
3.1	Shared Nearest-Neighbor (SNN) Graphs	59
3.2	Mutual Proximity Graphs	67
4	Graph Construction Performance	73
4.1	Metrics	74
4.2	Experimental Datasets	78
4.3	Results	80
4.3.1	Adjacency symmetrification method	82
4.3.2	Adaptive vs. fixed nearest-neighbor lists	87
4.3.3	Construction performance with varying k	89
4.3.4	Edge reweighting effectiveness	108
4.3.5	Primary-metric prescaling versus edge reweighting	113
4.4	Summary and Contributions	116
5	Concentration of Measure and Hubness in Graphs of Hyperspectral Data ..	118
5.1	Concentration of Measure	119
5.2	Enter Hubness	126
5.3	Concentration of Measure in HSI Data	134
5.4	Hubness in Graphs of HSI Data	139
5.5	Summary and Contributions	149
6	Graph-based HSI Analysis Algorithms	151
6.1	Segmentation, Clustering, and Classification	151
6.1.1	Segmentation and Clustering	152
6.1.2	Classification	154
6.2	Anomaly Detection	159
6.3	Target Detection	162
6.4	Change Detection	166
6.5	Physical Unit Determination	168
7	Schroedinger Eigenmaps	169

7.1 Modified SNN Cluster-Potentials for HSI Classification	175
7.2 Hubness-Based Barrier Potentials for Anomaly Detection	178
7.3 Summary and Contributions	181
8 Conclusions and Future Research Opportunities	182
Appendix A: HSI Data Sets with Ground Truth	186
Appendix B: HSI Data Sets	189
Appendix C: Related Publications	196
References	197

LIST OF TABLES

Table	Page
Table 1-1. Adjacency matrices for the directed and simple graphs of Figure 1-9	19
Table 4-1. Classification Accuracy Datasets	79
Table 5-1. Hyperspectral Evaluation Data	135
Table 5-2. Hubness and Intrinsic Dimension Estimates	141
Table 5-3. Hub Characterization (Good vs. Bad)	149
Table 7-1. SNN-SSSE Classification Accuracy	178

LIST OF FIGURES

Figure	Page
Figure 1-1. Electromagnetic spectrum; image courtesy of Wikimedia.org.	2
Figure 1-2. A hypercube is a 2-D spatial arrangement of spectra (c) typically collected a line at a time (a,b) (Shaw and Burke, 2003).	4
Figure 1-3. Geometric interpretation of spectra leads to subspace algorithms or those exploiting convex set geometry (modified from Manolakis et al., 2003). ..	7
Figure 1-4. The statistical perspective models spectra as a random vectors.	8
Figure 1-5. Spectroscopic analysis of spectra identifies key electronic, vibrational, or rotational features in the spectrum (Edwards and Galuska).	10
Figure 1-6. Two-band scatter plot of hyperspectral data shown with isocontours of Mahalanobis distance.	12
Figure 1-7. Linear mixing between three endmembers.	13
Figure 1-8. Primitive graph elements and basic graph characteristics.	16
Figure 1-9. 2-D directed (a) and an undirected (bidirectional) simple graph (b).	17
Figure 1-10. Directed, superset symmetric, and mutual 3-NN graph construction.	21
Figure 1-11. Two-dimensional graph created by the mutual 4-NN algorithm: a) with border regions, b) without.	24
Figure 2-1. ϵ -threshold graph construction.	31
Figure 2-2. Codensity histogram with node-specific NN assigned by DW k -NN.	35
Figure 2-3. Example dataset displaying three clusters of different density and edges weighting methods.	42
Figure 2-4. Connecting a graph via minimum spanning tree. (a) data points, (b) 1-NN graph, (c) MST, (d) k -NN+MST graph. Green edges are MST additions.	44
Figure 2-5. Encoding spatial relationships into an adjacency matrix.	45
Figure 2-6. Locally-weighted NN connects a pixel with global and local neighbors (Mercovich et al., 2011).	47
Figure 2-7. Common nearest neighbors indicate community structure (modified from Fan and Messinger, 2014).	50
Figure 3-1. Pairwise Euclidean distance matrix for (a) <i>SalinasA</i> in native order and (b) class-sorted order. Classes are outlined in white in (b).	55
Figure 3-2. Distance reweighting. (a) fixed-scale heat kernel and (b) adaptive heat kernel. Classes are outlined in white.	56
Figure 3-3. DBI and SC internal cluster-validation methods.	57
Figure 3-4. Local scale ratios (σ_i/σ_j) for (a) real data and (b) synthetic data of relatively uniform class-density scales. Class blocks are outlined in white.	58

Figure 3-5. Rank-dependent SNN similarity. Dotted ellipses touching the edge of each node denote its 3-NN.	64
Figure 3-6. Lack of the mutual constraint can lead to an undesired similarity score. .	66
Figure 3-7. 2-D scatterplot of distances from nodes i (x axis) and j (y axis) to all other nodes and their MP.	68
Figure 3-8. Example dataset displaying (a) three clusters of different density, (b) Euclidean distance weighting, and three edge reweighting methods: (c) ZMP, (d), SNN, and (e) MP. Performance measures (f).	70
Figure 4-1. Rank and adjusted-rank scoring for evaluating the structure of k -NN lists.	77
Figure 4-2. <i>Indian Pines</i> dataset for assessment of NN health. (a) true color image and (b) ground truth map (Baumgardner, et al., 1992).	80
Figure 4-3. Summary classification accuracy for all scenes using three adjacency-matrix types and four edge-allocation methods.	81
Figure 4-4. Differential accuracy results for fixed and adaptive graphs at low (top) and high (bottom) k values.	84
Figure 4-5. Differential ϕ -edge ratio plots for the maximum k row in Figure 4-4.	86
Figure 4-6. Differential ϕ -edge ratios between fixed and adaptive graph variants by scene.	88
Figure 4-7. k -NN Dashboard #1: Accuracies, edges counts, and cross ground-truth edges for <i>Indian Pines</i>	94
Figure 4-8. ANN Dashboard #1: Accuracies, edges counts, and cross ground-truth edges for <i>Indian Pines</i>	97
Figure 4-9. k -NN Dashboard #2: Impacts from forcing bidirectional edges.	100
Figure 4-10. ANN Dashboard #2: Impacts from forcing bidirectional edges.	102
Figure 4-11. k -NN Dashboard #3: Miscellaneous measures that vary with k	105
Figure 4-12. ANN Dashboard #3: Miscellaneous measures that vary with k	107
Figure 4-13. Adaptive edge-reweighting effectiveness is detrimental to the NN health.	110
Figure 4-15. Differential ϕ -edge ratio plotted versus the differential confusion score for prescaling results.	114
Figure 5-1. Concentration of Euclidean (top row), cosine distances (middle row), and normalized Euclidean distance (bottom row).	125
Figure 5-2. Skewness in the O^5 histogram of random normal <i>i.i.d</i> data for $p = 3, 20, \& 100$ (top row). O^5 vs. the L_2 (middle row) and cosine (bottom row) distances from the mean. Modified from Radovanović et al. (2010a).	128
Figure 5-3. Decrease in probability of extreme pixels in multidimensional <i>i.i.d</i> . data.	129
Figure 5-4. Hub location and skewness of the 30-occurrences histogram for an example <i>Microscene</i> hypercube.	130
Figure 5-5. Hubness of Euclidean and cosine distances for <i>i.i.d</i> . data on $[0,1]^d$	132
Figure 5-6. Hub location with respect to the data (left) and cluster means (right) for the <i>hyperdesert</i> scene.	133
Figure 5-7. Distance concentration of Table 5-1 datasets shaded by intrinsic dimension.	137

Figure 5-8. Distance concentration of permuted HSI data. Notice these data are concentrated by much smaller multiples of the codensity measurement.	138
Figure 6-1. Diffusion geometry coordinates (bands 5-6-7) for the Moffet field scene as shown in He et al. (2009).	156
Figure 6-2. Isocontours of anomalousness for (a) TAD and (b) RX (Basener and Messinger, 2007).	160
Figure 6-3. Anomaly detection in (a) spectral space, (b) CTD space, and (c) ROC analysis (Albano and Messinger, 2012).	162
Figure 6-4. Detection statistic image for (a) the SMF in CTD space, (b) subspace ACE in spectral space, and (c) ROC analysis (Albano et al., 2013).	164
Figure 6-5. Graph-theoretic processing for tile-based HSI change detection (Albano et al., 2011).	168
Figure 7-1. SNN-SSSE with spectral through spatial weightings for the <i>Indian Pines</i> , <i>Salinas</i> , and <i>Pavia University</i> scenes.	177
Figure 7-2. SHARE image of Avon, NY (left), RX (center) and HAAD (right) anomaly detection images.	180

LIST OF EQUATIONS

Equation	Page
(1.1) Multivariate Normal Distribution	11
(1.2) Linear Mixture Model	13
(1.3) Minimum & Maximum Node Degrees for k-NN Graphs Built from Directed Edges or by Enforcing the Superset Symmetry Criterion	23
(1.4) Minimum & Maximum Node Degrees for k-NN Graphs Built from Enforcing the Mutuality Criterion	23
(2.1) Codensity	34
(2.2) Non-Iterative Contextual Dissimilarity Measure	39
(2.3) Localized Scaling (Zelnick-Manor & Perona)	40
(2.4) Heat Kernel (Shi and Malik)	41
(2.5) Cosine Distance Based Similarity Measure (He et al.)	41
(2.6) Spatial-Spectral Method with Local Encoding (Gillis and Bowles)	48
(2.7) Adaptive Spatial-Spectral Method with Common Neighbors (Fan & Messinger)	49
(2.8) Spatial-Spectral Distance (Hou et al.)	51
(2.9) Weighted Spatial-Spectral Distance (Benedetto et al.)	51
(3.1) Shared Nearest Neighbors	60
(3.2) Adaptive Shared Nearest Neighbors	60
(3.3) Shared Nearest Neighbor Cosine Similarity	60
(3.4) Rank-Based Shared Nearest Neighbors	63
(3.5) Dissimilarity Enhanced Rank-Based Shared Nearest Neighbors	66
(3.6) Distance Probability	67
(3.7) Mutual Probability	69
(3.8) Mutual Probability with Independent Probabilities	69
(5.1) Relative Contrast	120
(5.2) Relative Variance	121
(5.3) Hubness Measure	130
(5.4) Data Concentration Measure	136
(6.1) Weighted Vertex Volume	161
(6.2) Normalized Edge Volume	167
(7.1) Incidence Matrix	169
(7.2) Graph Laplacian	170
(7.3) Normalized Graph Laplacian	170
(7.4) Random Walk Laplacian	170
(7.5) Manifold Embedding	171

(7.6) Equivalent Manifold Embedding	171
(7.7) Schroedinger Eigenmaps	172
(7.8) Cluster Potential Matrix Sub Components	173
(7.9) Example Cluster Potential Matrix Primitive	173
(7.10) Example Cluster Potential Matrix	174
(7.11) Spatially Weighted Cluster Potential (Benedetto et al.)	174
(7.12) Spatial-Spectral Cluster Potential (Cahill et al.)	175
(7.13) SNN Spatial-Spectral Cluster Potential Modifier	175

LIST OF ABBREVIATIONS

ACE	Adaptive Cosine Estimator
ANN	Adaptive Nearest Neighbors
AVIRIS	Advanced Visible/Infrared Imaging Spectrometer
BRDF	Bidirectional Reflectance Distribution Function
C2I	Correct to Incorrect
CDF	Cumulative Distribution Function
CoM	Concentration of Measure
CTD	Commute Time Distance
DBI	Davies-Bouldin Index
DW k -NN	Density Weighted k -NN
DW k -NN with CDF	Density Weighted k -NN with CDF
<i>edim</i>	extrinsic dimension
EM	Endmember
GIC	Grupo de Inteligencia Computacional
GPGPU	General Purpose Graphical Processing Unit
GSD	Ground Sample Distance
HAAD	Hubness-Assisted Anomaly Detection
HSI	Hyperspectral Imaging
HYDICE	Hyperspectral Digital Imagery Collection Experiment
I2C	Incorrect to Correct
ICA	Independent Component Analysis
<i>idim</i>	intrinsic dimension
IFOV	Instantaneous Field of View
<i>i.i.d</i>	Independent and Identically Distributed
ISOMAP	Isometric Mapping
JSPCD	Joint Spatial-Pixel Characteristic Distance
k -NN	k -Nearest Neighbors
K-S	Kolmogorov-Smirnov Test
LLE	Locally Linear Embedding
LMM	Linear Mixture Model
MNF	Minimum Noise Fraction
MP	Mutual Proximity
MSI	Multispectral Imaging
MST	Minimum Spanning Tree
MVN	Multivariate Normal
NaN	Not a Number

NEV	Normalized Edge Volume
NICDM	Non Iterative Contextual Dissimilarity Measure
NIST	National Institute of Standards and Technology
NN	Nearest Neighbor
NNN	Natural Nearest Neighbors
PCA	Principal Component Analysis
RBF	Radial Basis Function
RIT	Rochester Institute of Technology
ROC	Receiver Operator Characteristic
ROSiS	Reflective Optics System Imaging Spectrometer
RX	Reed-Xiaoli detector
SAM	Spectral Angle Mapper
SC	Silhouette Coefficient
SDEL	Standard Deviation of Edge Weights
SE	Schroedinger Eigenmap
SEM	Stochastic Expectation Maximization
Sk -NN	Shortest Path k -NN
SMF	Spectral Matched Filter
SNN	Shared Nearest Neighbors
SNR	Signal to Noise Ratio
SSSE	Spatial-Spectral Schroedinger Eigenmaps
SOC	Surface Optics Corporation
SVD	Singular Value Decomposition
SVM	Support Vector Machine
TAD	Topological Anomaly Detector
USGS	United States Geological Survey
WEKA	Waikato Environment for Knowledge Analysis
WVV	Weighted Vertex Volume
ZMP	Zelnick-Manor and Perona

LIST OF SYMBOLS

$ \cdot $	Set size
$\ \cdot\ _p$	L_p vector norm
$\ v_i - v_j\ $	Spatial distance between nodes i and j
$\{\cdot\}$	Set notation
α	Fusion ratio parameter for Schroedinger Eigenmaps
$d^-(v)$	Node in-degree
$d^+(v)$	Node out-degree
$dist(i,j)$	Spectral space distance between v_i and v_j
$\delta(G)$	Minimum node-degree
$\Delta(G)$	Maximum node-degree
$\mathbf{1}$	Vector of ones
α	Fractional value of potential matrix in Schroedinger Eigenmap construction
a_{ij}	Adjacency matrix component
a_i	Abundance of i^{th} endmembers
\mathbf{a}	Abundance vector
\mathbf{A}	Adjacency matrix
\mathbf{B}	Incidence matrix
b_{ij}	Incidence matrix component
c	Scalar multiplier for concentration measure tests
c_{ij}	Number of common neighbors between nodes i and j
\mathbf{C}	Covariance matrix
δ_{ij}	Weighting parameter in Schroedinger Eigenmaps
d	Feature (spectral) space dimension
$deg(v)$	Degree of node v
$d(v_i, v_j), d(i, j), \delta_{ij}$	Feature-space distance between nodes i and j
D	Dataset (spectral image)
D_{min}	Minimum distance from reference pixel
D_{max}	Maximum distance from reference pixel
e	Any edge
e_{ij}	Edge between nodes i and j
E	Edge set
g	Number of global neighbors
G	Graph
h	Hubness measure
k	Number of neighbors
$k(i), k_i$	Neighborhood size for pixel i
k_{avg}	Average number of neighbors in a graph
k_{min}	Minimum k value in a NN list
k_{max}	Maximum k value in a NN list

λ	Eigenvalue or wavelength
l	Number of local neighbors
L	Graph Laplacian matrix
\mathbf{m}, μ	Data mean
\mathcal{M}	Manifold
n	Number of vertices/pixels
\mathbb{N}	Set of integers
\mathbb{N}^+	Set of positive integers
$N(v)$	Neighborhood of vertex v
$NN_k(v)$	Nearest-neighbor list of size k for vertex v
O^k	k -occurrence value (in degree)
p	Intrinsic (inherent) dimension of dataset, manifold dimension
P_{ij}	Path between nodes i & j
Q	Cluster node set
r	Size of window centered on test pixel
$rank(i)$	Rank of node i in a nearest neighbor list
\mathbb{R}^d	Set of real d -dimensional vectors
\mathbb{R}^+	Set of real positive numbers
σ	Radial basis function scalar related to distance
s_i, σ_i	Codensity of node i
\mathbf{S}	Set of basis vectors for the LMM
\mathbf{s}_i	Basis vector in LMM (an endmember) and standard deviation
θ_{ij}	Spectral angle between v_i and v_j
\mathbf{W}	Weighted adjacency matrix
w_{ij}	Weighted adjacency matrix component
$var(\mathbf{v})$	Vector variance
v	Any vertex
v_i	Vertex i
V	Vertex set
$V_{k,l}^{(i,j)}$	Schroedinger Eigenmap submatrix
\mathbf{V}	Potential matrix
x_i	Feature (band) number within \mathbf{x}
\mathbf{x}	Pixel spectrum
y_i	Manifold coordinate from i^{th} Eigenmap
z	z -score value

ABSTRACT

NEW METHODS OF SPECTRAL-DENSITY BASED GRAPH CONSTRUCTION AND THEIR APPLICATION TO HYPERSPECTRAL IMAGE ANALYSIS

Jeffrey Stevens, Ph.D.

George Mason University, 2017

Dissertation Director: Dr. John J. Qu

The past decade has seen the emergence of many hyperspectral image (HSI) analysis algorithms based on graph theory and derived manifold-coordinates. Yet, despite the growing number of algorithms, there has been limited study of the graphs constructed from spectral data themselves. Which graphs are appropriate for various HSI analyses—and why? This research aims to begin addressing these questions as the performance of graph-based techniques is inextricably tied to the graphical model constructed from the spectral data. We begin with a literature review providing a survey of spectral graph construction techniques currently used by the hyperspectral community, starting with simple constructs demonstrating basic concepts and then incrementally adding components to derive more complex approaches. Throughout this development, we discuss algorithm advantages and disadvantages for different types of hyperspectral analysis. A focus is provided on techniques influenced by spectral density through which the concept of community

structure arises. Through the use of simulated and real HSI data, we demonstrate density-based edge allocation produces more uniform nearest neighbor lists than non-density based techniques through increasing the number of intracluster edges, facilitating higher k -nearest neighbor (k -NN) classification performance. Imposing the common mutuality constraint to symmetrify adjacency matrices is demonstrated to be beneficial in most circumstances, especially in rural (less cluttered) scenes. Many complex adaptive edge-reweighting techniques are shown to slightly degrade nearest-neighbor list characteristics. Analysis suggests this condition is possibly attributable to the validity of characterizing spectral density by a single variable representing data scale for each pixel. Additionally, it is shown that imposing mutuality hurts the performance of adaptive edge-allocation techniques or any technique that aims to assign a low number of edges (< 10) to any pixel. A simple k bias addresses this problem.

Many of the adaptive edge-reweighting techniques are based on the concept of codensity, so we explore codensity properties as they relate to density-based edge reweighting. We find that codensity may not be the best estimator of local scale due to variations in cluster density, so we introduce and compare two inherently density-weighted graph construction techniques from the data mining literature: shared nearest neighbors (SNN) and mutual proximity (MP). MP and SNN are not reliant upon a codensity measure, hence are not susceptible to its shortcomings. Neither has been used for hyperspectral analyses, so this presents the first study of these techniques on HSI data. We demonstrate MP and SNN can offer better performance, but in general none of the reweighting techniques improve the quality of these spectral graphs in our neighborhood structure tests.

As such, these complex adaptive edge-reweighting techniques may need to be modified to increase their effectiveness.

During this investigation, we probe deeper into properties of high-dimensional data and introduce the concept of concentration of measure (CoM)—the degradation in the efficacy of many common distance measures with increasing dimensionality—as it relates to spectral graph construction. CoM exists in pairwise distances between HSI pixels, but not to the degree experienced in random data of the same extrinsic dimension; a characteristic we demonstrate is due to the rich correlation and cluster structure present in HSI data. CoM can lead to hubness—a condition wherein some nodes have short distances (high similarities) to an exceptionally large number of nodes. We study hub presence in 49 HSI datasets of varying resolutions, altitudes, and spectral bands to demonstrate hubness effects are negligible in a k -NN classification example (generalized counting scenarios), but we note its impact on methods that use edge weights to derive manifold coordinates or splitting clusters based on spectral graph theory requires more investigation.

Many of these new graph-related quantities can be exploited to demonstrate new techniques for HSI classification and anomaly detection. We present an initial exploration into this relatively new and exciting field based on an enhanced Schroedinger Eigenmap classification example and compare results to the current state-of-the-art approach. We produce equivalent results, but demonstrate different types of misclassifications, opening the door to combine the best of both approaches to achieve truly superior performance. A separate less mature hubness-assisted anomaly detector (HAAD) is also presented.

1

INTRODUCTION

Remote sensing is the field of study concerned with acquiring information about objects of interest without physical contact (Schott, 2007). Remote acquisition of information has many military and civilian applications as well as an ever-growing set of commercial uses as evidenced by the explosion of imagery consumers and providers such as Google Earth, precision agriculture markets, and financial institutions. The list seems endless as any smart phone user can see their home from space or the road at the touch of a button. Remote sensors have continued to evolve and applications can be found across the electromagnetic spectrum (Figure 1-1).

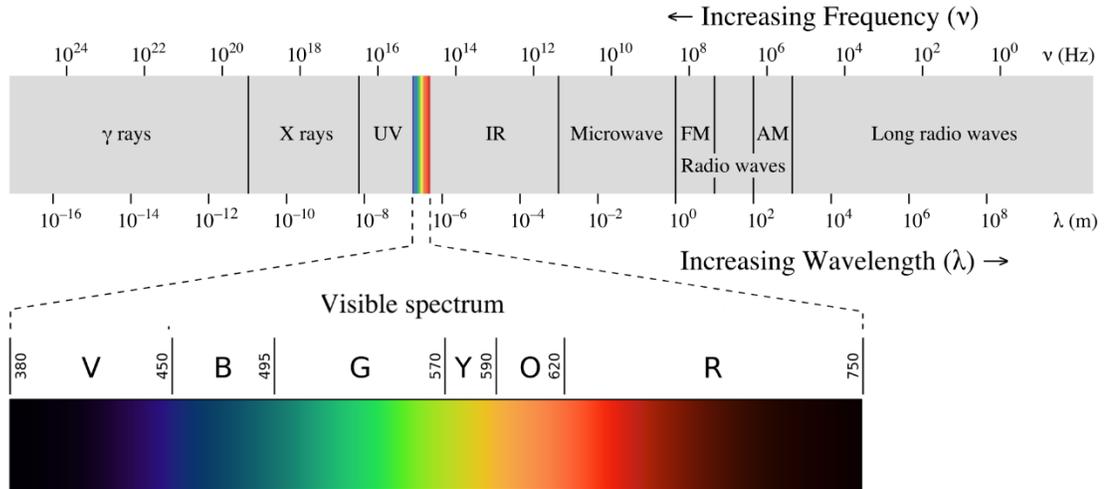


Figure 1-1. Electromagnetic spectrum; image courtesy of Wikimedia.org.

This research focuses on data from the visible-, near-, and shortwave-infrared parts of the spectrum dominated by solar reflected radiation (0.4-2.5 μm), but can easily be extended to the thermal regimes. Specifically, we focus our attention on hyperspectral imaging systems (imaging spectrometers) that acquire (hundreds) of bands across the spectral range. The rich spectral content of these data enable finer material discrimination than possible with the more common multispectral imaging data (MSI), comprised of fewer broader bands (tens of bands or less).

The remainder of this chapter will provide a brief introduction to HSI and common data analysis algorithms, citing the need to explore nonlinear methods over the more prevalent linear techniques. We provide a smooth transition from the common statistical and geometric techniques of the past several decades into analyzing graphs built from HSI

data and conclude with the specific spectral-graph related questions addressed by this research and detailed in subsequent chapters.

1.1 Hyperspectral Imaging

HSI sensors acquire hundreds of spectral radiance measurements for each two-dimensional (2-D) spatial location in the acquired scene. The spectral measurement for a particular scene location (pixel) is called a *spectrum* (plural; *spectra*) and contains information about the chemical and physical properties of the sensed area and is the basic unit of measure in HSI; it may be thought of as a vector in a d -dimensional space, \mathbb{R}^d , where d is the number of spectral bands (Schott, 2007). The spectrum, $\mathbf{x} = [x_1, x_2, \dots, x_d]^T$, is thus a vector composed of the individual brightness measurements from each band, and traces out a curve known as a spectral signature when plotted as a function of wavelength (Figure 1-2 bottom). The aggregate collection of all such measurements is called a hypercube, a three dimensional structure comprised of two spatial (x,y) dimensions and a wavelength dimension (λ) (Figure 1-2, top; Shaw and Burke, 2003).

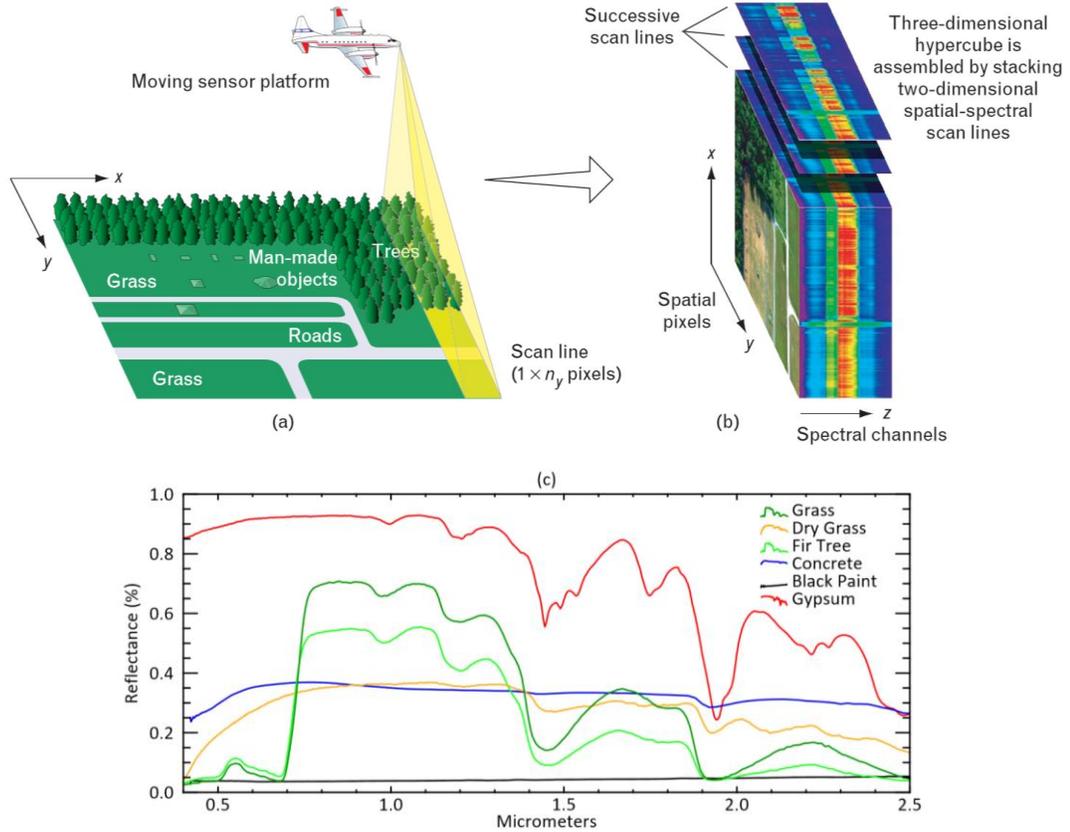


Figure 1-2. A hypercube is a 2-D spatial arrangement of spectra (c) typically collected a line at a time (a,b) (Shaw and Burke, 2003).

Spectra may represent physical units such as reflectivity, emissivity, or absorptivity or spectral radiance data which also contains information about the atmosphere and the sensor. We adopt a spectral reflectance convention from here on without loss of generality.

The term *spectral* has multiple definitions based on the discussed discipline. Spectral remote sensing involves the use of several wavelength regimes for information acquisition, so spectral infers spectroscopy in this case. Graphs built from spectral data in this sense can be called spectral graphs (Benedetto et al., 2012a). This is not to be confused with spectral graph-theory which is concerned with the analysis of eigendecompositions of

matrices associated with graph characteristics. The corresponding definition should be clear from the context of the discussion.

Pixels composed of predominantly one material in its instantaneous field of view (IFOV) will display spectra very similar to that of pure materials (Figure 1-2, bottom), while pixels composed of many materials will present as a composite spectrum where the mixing relationship is determined by the areal distribution and physical structure of those constituents. The goal of hyperspectral imaging is to analyze these spectra and separate them into their constituent parts so that quantitative statements can be made about pixel composition. Composition is a general term that may refer to solid materials when sensing the ground or gaseous species when sensing the atmosphere to name a few (Manolakis, Jairam, Zhang, and Rossacci, 2007; Griffin, Kerekes, Farrar, and Burke, 2001; Theiler, Foy, and Fraser, 2005).

1.2 Hyperspectral Data Analysis

Many different approaches for processing HSI data exist, that differ in their perspective on what the *spectrum* represents. As such, these methods utilize different means of characterizing spectral space and the subsequent techniques used in data analysis. We review the three most common spectrum perspectives and their spectral-space characterizations in Chapters 1.2.1 and 1.2.2 respectively to provide the requisite context of common data models, only to build out from there towards a graphical representations of HSI data.

1.2.1 Common Perspectives on Spectra

There are three prevailing perspectives (or interpretations) on spectra from which algorithms have been developed: geometric, statistical, and spectroscopic. Each perspective results in a suite of methods and data models for processing spectral data that address a broad range of applications to include target detection, anomaly detection, classification, quantification, and change detection. A brief qualitative description of these models is provided in the remainder of this section and the reader is referred to Manolakis et al. (2003); Manolakis, Lockwood, Cooley, and Jacobson (2009); Matteoli, Diani, and Corsini (2010); and Eismann (2012) for excellent reviews of common statistical and geometric data models.

The GEOMETRIC perspective models each spectrum as a single deterministic point in \mathbb{R}^d . Spectra from a hypercube may therefore be thought of as a scattered set of points in d -D hyperspace. However, significant band-to-band correlation exists and the data reside in a lower dimensional subspace, \mathbb{R}^p , where $p \ll d$ (Schlamm, Resmini, Messinger, and Basener, 2010). Basis vectors for this subspace can be obtained from eigenvectors of the spectral covariance matrix \mathbf{C} (Manolakis et al., 2003; Boardman, 1990; Schott, 2007), or by finding the pure materials (endmembers) in the scene (Keshava, 2003; Boardman, 1994). These methods are called vector subspace models and linear mixture models respectively. In the former case, orthogonal basis vectors are guaranteed by the eigen- or singular-value decomposition (SVD) of the spectral covariance matrix, whereas in the latter, the basis vectors are typically not orthogonal. Algorithms based on these geometric models treat either the background spectra, target spectra, or both, as existing within subspaces. Some common examples are the adaptive and generalized likelihood ratio test

subspace detectors (Manolakis et al., 2003) and orthogonal subspace projection (Harsanyi and Chang, 1994).

Eigenvectors of the covariance may be obtained from principal components analysis (PCA) (Eismann, 2012), independent component analysis (ICA), or the SVD (Boardman, 1990) through maximizing data variance along each principal direction. These basis vectors do not represent real materials, but rather linear combinations of the canonical orthogonal spectral axes that best span the space (Boardman, 1990). A two dimensional subspace existing in three dimensions is shown in the left side of Figure 1-3.

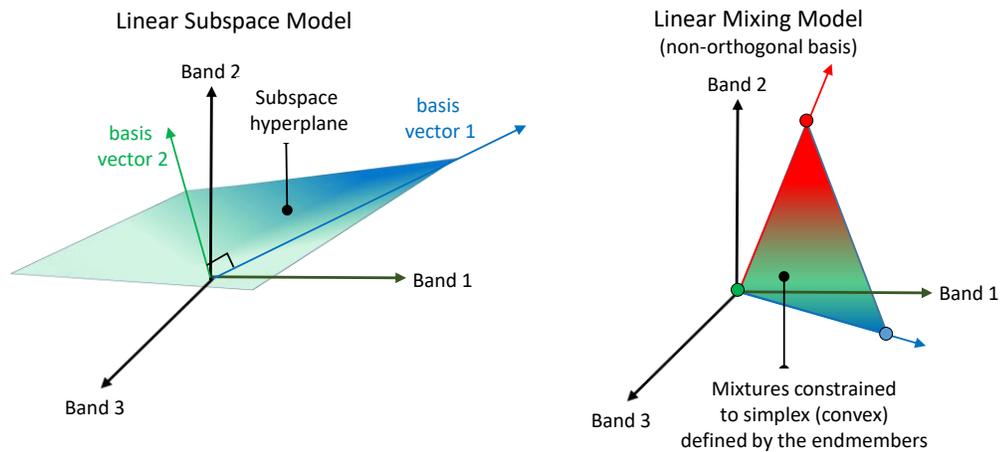


Figure 1-3. Geometric interpretation of spectra leads to subspace algorithms or those exploiting convex set geometry (modified from Manolakis et al., 2003).

A second common geometric model is the linear mixing model (LMM). The linear mixing model states spectra can be represented as linear combinations of scene endmembers (pure materials in the scene). These endmember spectra are not linearly independent and form the vertices of a $p-1$ dimensional simplex in spectral space, where p

is the number of endmembers. Convex set theory states if spectra are linear combinations of endmembers, then all spectra must reside inside within this simplex (Boardman, 1994) (Figure 1-3, right). This model is the most common model for representing the synthesis of mixed pixels from distinct endmembers (Keshava and Mustard, 2002).

Both geometric representations have been used in many anomaly and target detection algorithms over the past several decades (Manolakis et al., 2009; Stein et al., 2002; Boardman, 1998; Matteoli et al., 2010).

The STATISTICAL perspective considers each spectrum as a d -D random vector, i.e., a realization of a random spectral distribution and thus lives within a region of spectral space delineated by a probability density function (Figure 1-4).

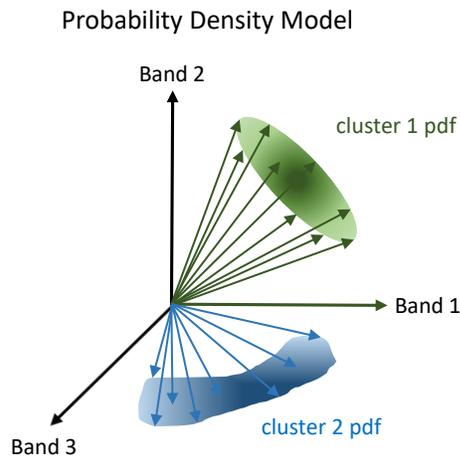


Figure 1-4. The statistical perspective models spectra as a random vectors.

Many parametric probability density functions possess elliptical symmetry, are quickly computed, and are simple to implement (Manolakis et al., 2003; Willis, 2009), with

the multivariate normal model (MVN) model being most prevalent in the HSI literature (Schlamm et al., 2011). Stochastic expectation maximization (SEM), the Reed-Xiaoli (RX) detector, and the spectral matched filter (SMF) are common statistical methods making use of the MVN for classification, anomaly detection, and target detection respectively (Manolakis et al., 2003; Eismann, 2012; Stein et al., 2002; Matteoli et al., 2010). However, as the spatial and spectral resolutions of hyperspectral data increase, parametric statistical models, especially the MVN, are less able to represent the actual data distributions (Schlamm and Messinger, 2011; Ziemann and Messinger, 2014a). As a result, techniques not reliant on parametric models need to be developed.

The SPECTROSCOPIC perspective is physics-based (Eismann, 2012), where each spectrum represents a material's chemical and/or physical properties related to interactions of electromagnetic radiation with matter (e.g., reflectivity, emissivity, transmissivity, or absorptivity). A material's spectral response can be used to identify composition by examining the locations and depth of features in the spectrum. Each feature represents an identifying (diagnostic) aspect of the material due to its chemistry or physical structure. Figure 1-5 shows a spectroscopic markup of an infrared transmittance spectrum indicating some important vibrational modes.

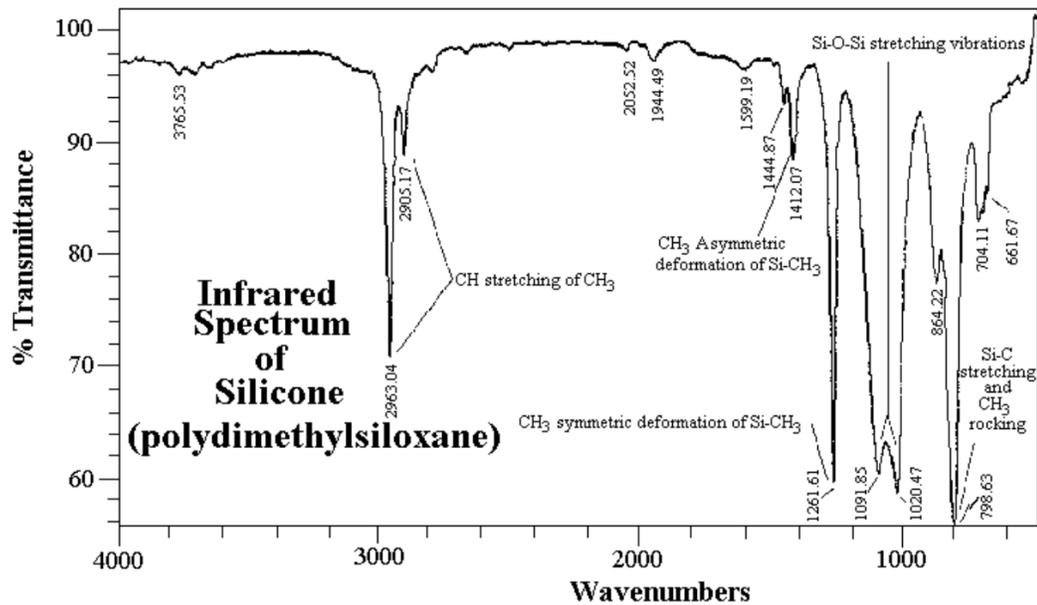


Figure 1-5. Spectroscopic analysis of spectra identifies key electronic, vibrational, or rotational features in the spectrum (Edwards and Galuska).

Some spectroscopic methods rely heavily on derivative analysis for the identification of these features so are most commonly found in lab and other terrestrial systems that have high signal-to-noise (SNR) ratios. The presence of these features at particular wavelengths are the data required to perform various HSI analyses.

Physical quantities may also be derived from these data. For example, the quantity of a gaseous species within an IFOV is directly attributable to a differential radiance signal caused by the presence of the gas. Analysis of the differential magnitude can be used to estimate the quantity in mass or number of molecules (Turcotte and Davenport, 2010). Other common analyses include water depth extraction (Gillis, Bowles, Lamela, Rhea, and Bachmann, 2005) and temperature extraction (Jellison and Miller, 2006).

1.2.2 Hyperspectral Data Models

Hyperspectral analysis algorithms need to autonomously characterize spectral space in order to adapt to scene content. There are three predominant means of modeling spectral space: parametric statistics, non-statistical, and nonparametric statistics; in the order of greatest to the least occurrence in the literature.

Parametric statistical models are analytic constructs with inherent symmetries and smoothness due to their functional forms. Both are elliptically symmetric hence share many of the same advantages and optimal performance under the assumption of symmetric probability distributions described by the spectral covariance matrix (Manolakis et al., 2003; Willis, 2009).

The most ubiquitous model found in the literature is the MVN () due to its simplicity and ease of use (Manolakis, Rossacci, Cipar, Lockwood, Cooley, and Jacobson, 2005). In (1), \mathbf{m} represents the data mean and all other terms are as previously described.

$$f(\mathbf{x}) = \frac{1}{(2\pi)^{\frac{d}{2}} |\mathbf{C}|^{\frac{1}{2}}} e^{-\frac{1}{2}(\mathbf{x}-\mathbf{m})^T \mathbf{C}^{-1}(\mathbf{x}-\mathbf{m})} \quad (1.1)$$

A limitation of the MVN model is that hyperspectral data typically violate the assumptions of multivariate normality (Manolakis et al., 2003). This non-normality is easily observed in the scatter plot of Figure 1-6 where the level sets (isocontours) of equal multivariate normal probability (Mahalanobis distance) are shown as concentric rings. The red triangles are considered more anomalous than the green square to the lower left, contrary to human interpretation. This situation arises because the data are not best described by a MVN, albeit these models have been very successful despite this limitation.

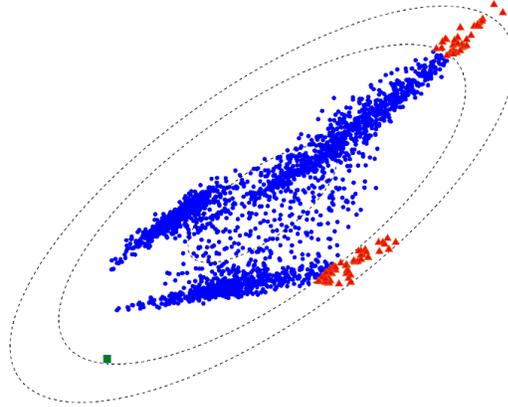


Figure 1-6. Two-band scatter plot of hyperspectral data shown with isocontours of Mahalanobis distance.

The degree by which data deviates from MVN has been an active area of research, and various screening methods have been developed indicating when this model is appropriate; see Schlamm and Messinger (2011) for details. Manolakis et al. (2003) has suggested these data are better modeled by the t distribution instead of the MVN.

The non-normal distribution of HSI data has been described in many publications (Manolakis et al., 2003, 2006). A few investigators have utilized non parametric statistical models to characterize spectral data under the premise that a more realistic descriptions of the non-normal HSI data will result in better analysis algorithms. Automated learning of these nonparametric models has proven to be a limiting factor (Matteoli, Veracini, Diani, and Corsini 2013).

The LMM is a physics-based model where each spectrum in a d band hyperspectral image can be represented as a weighted linear sum of no more than $d + 1$ endmembers.

Representing the abundance of the i^{th} endmember, \mathbf{s}_i , as a_i , the linear mixing model for p endmembers is given by

$$\begin{aligned} \mathbf{x} &= a_1\mathbf{s}_1 + a_2\mathbf{s}_2 + \dots + a_p\mathbf{s}_p \\ &= \sum_{i=1}^p a_i\mathbf{s}_i = \mathbf{S}\mathbf{a}, \end{aligned} \tag{1.2}$$

where \mathbf{S} represents a matrix of endmember basis vectors $[\mathbf{s}_1, \mathbf{s}_2, \dots, \mathbf{s}_p]$, and $\mathbf{a} = [a_1, a_2, \dots, a_p]^T$ is a matrix of weights (or coefficients) applied to the endmember basis to reconstruct the pixel spectrum, \mathbf{x} . Under the LMM assumption, any pixel must reside within the convex hull of the endmembers as shown in Figure 1-7.

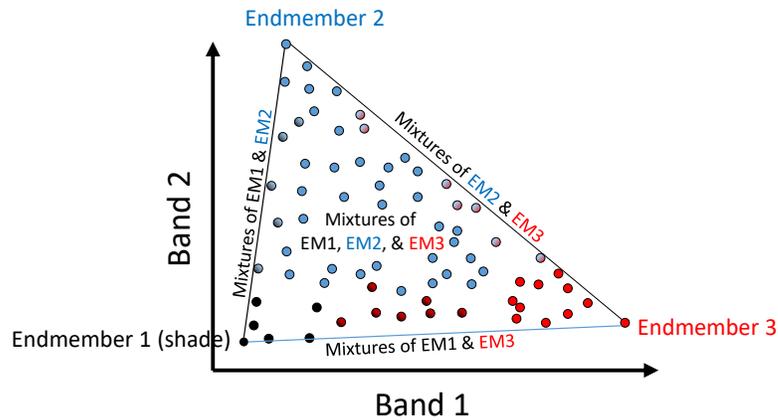


Figure 1-7. Linear mixing between three endmembers.

Note that Eq. (1.2) also represents the subspace model, with simple substitution of the orthogonal basis vectors derived from PCA with those of the non-orthogonal basis

determined from endmember estimation algorithms and swapping the abundances with the projection of each spectrum on the corresponding eigenvector.

Many researchers have examined segmenting imagery into clusters, and then modeling each cluster separately via the methods described above. This approach is intuitively appealing upon inspection of Figure 1-6 where one can argue the scene is better modeled by a mixture of many clusters with differing characteristics (Funk and Theiler, 2001).

In summary, hyperspectral data

- are not linear
- are not best described by a multivariate normal model
- are clustered into regions of similar materials

These points have motivated investigators to continue exploration of new data models and analysis other than those previously described.

1.3 Graphs as a Means for HSI Data Analysis

Most of the statistical and geometric algorithms discussed thus far are linear methods, hence may be ineffective modeling nonlinearities present in HSI data most commonly caused by three phenomena: (1) variations in material reflectivity with illumination and viewing angle, i.e., the bidirectional reflectance distribution function (BRDF); (2) transmissive media such as water; and (3) intimate mixing found in many soils

(Keshava, 2003; Bachmann, Ainsworth, and Fusina, 2005). New methods are therefore required to analyze data exhibiting nonlinearities.

In 2005, Bachmann et al. from the Naval Research Laboratory explored a new geometric perspective in HSI data analysis. They viewed spectra as vertices (nodes) in a d -D graph, with edges connecting similar pixels. Their method models hyperspace geometry (or “structure”) without the limitations of strict statistical forms or linear subspaces. Bachmann et al. first applied methods derived from these spectral graphs towards the classification of wetland HSI imagery exhibiting nonlinear behavior. Several researchers have continued the development of graph-based techniques, building on the ideas presented in Bachmann’s seminal papers (Ziemann and Messinger, 2014a; Albano, Messinger, and Rotman, 2012b).

Hyperspectral data do not inherently exist as a graph, so must be converted into graphical form by selection of a function or heuristic to enable graph-based analysis. The next couple chapters describe algorithms for this task. However, review of some basic graph terminology is in order before discussion can begin. This material provides the basis for understanding the many common graph construction techniques found in the hyperspectral remote sensing literature (Chapter 2) as well as new methods presented in this work (Chapter 3).

A graph $G = (V, E)$ in its simplest form is defined as a pair of two finite sets: a vertex (or node) set and an edge set, denoted V and E respectively. Vertices are points in d -D space determined by the coordinates of each spectrum. The size of the vertex set, $|V|$, is the number of pixels in the hyperspectral image. We denote a general vertex by v ,

whereas specific vertices (pixels) are indicated using subscripts, e.g., v_i . Note that the terms node, vertex, and pixel are used interchangeably throughout the text based on the context of the discussion. Directed edges (and hence graphs) utilize ordered pairs of points indicating the source and sink of the edge, i.e., $e_{ij} = (v_i, v_j)$ represents an edge from v_i to v_j (Figure 1-8).

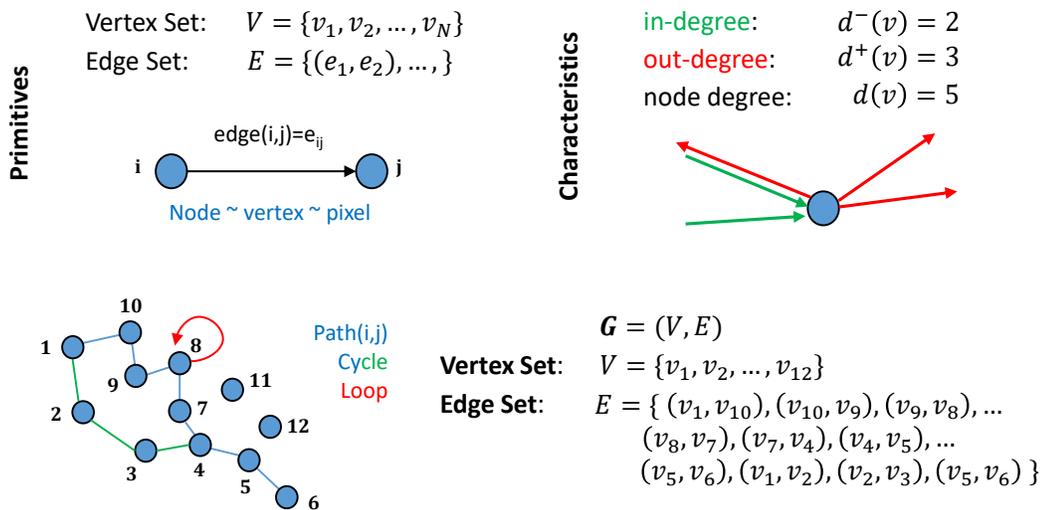


Figure 1-8. Primitive graph elements and basic graph characteristics.

A path P_{ij} , is an ordered set of edges tracing a path from v_i to v_j along edges existing within E . P_{16} is shown in blue to the bottom left of Figure 1-8. If the path comes back to the original vertex, it is called a cycle. In this case the cycle (C_{11}) sequence for the lower left graphs traverses from nodes 1,10,9,8,7,4,3,2,1, i.e., $\{P_{14}, P_{41}\}$. An edge from a vertex to itself is called a loop, i.e., $e_{11} = (v_1, v_1)$. Some graphs can also have multiple edges between the same two vertices.

There are several quantities related to the number of edges to a given node. The node in-degree $d^-(v)$ is the number of edges incident upon a vertex, the node out-degree $d^+(v)$ is the number of edges leaving a vertex, and the node degree $d(v)$ is the sum of the two. An example is provided to the upper right of Figure 1-8.

The edge set may be either directed or undirected (Figure 1-9). Directed edges (and hence graphs) utilize ordered pairs of points indicating the source and sink of the edge, i.e., (v_i, v_j) represents an edge from v_i to v_j . The ordering is irrelevant for undirected (bidirectional) edges, i.e., these edges simply indicate a relationship between pixels as shown in Figure 1-9b, with bold edges indicating bidirectionality. Undirected graphs without loops ($i \neq j$) and multiple edges between the same endpoints are called simple graphs (West, 2001) and are most common in the HSI literature. Many methods to define these relationships are discussed in the following subsections.

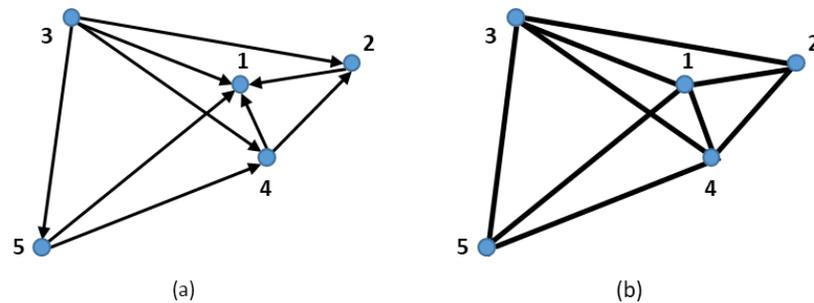


Figure 1-9. 2-D directed (a) and an undirected (bidirectional) simple graph (b).

Central to all spectral graph construction techniques is the concept of a vertex neighborhood. A neighborhood, $N(v)$, is a set of vertices that are related (deemed

“similar”) to the vertex being examined by some heuristic or analytical method. Edges are constructed between a vertex and all vertices in its neighborhood. Vertices with edges between them are said to be adjacent. A multitude of methods exist to define this neighborhood, and the size of the edge set, $|E|$, can vary greatly between methods for the same set of vertices. One of the simplest methods of neighborhood construction is to define the neighborhood as the k most similar pixels (fixed k). This is called a directed k -NN graph as each pixel is the source node of its nearest neighbor (NN) list and all neighboring pixels are the sink nodes¹, i.e., the edges are directed.

Vertex adjacency may be represented by multiple data structures. Of particular interest in spectral imaging is the adjacency matrix, $[A]_{ij} = a_{ij}$, where $a_{ij} = 1$ if an edge exists between v_i and v_j , and zero otherwise. A is therefore an $n \times n$ binary asymmetric or symmetric matrix for directed and undirected graphs respectively, where n is the number of pixels in the image (Table 1-1).

¹ We will sometimes denote the k nearest neighbors as NN_k .

Table 1-1. Adjacency matrices for the directed and simple graphs of Figure 1-9.

		Sink Node Number				
		1	2	3	4	5
Source Node Number	1	0	0	0	0	0
	2	1	0	0	0	0
	3	1	1	0	1	1
	4	1	1	0	0	0
	5	1	0	0	1	0

(a)

		Sink Node Number				
		1	2	3	4	5
Source Node Number	1	0	1	1	1	1
	2	1	0	1	1	0
	3	1	1	0	1	1
	4	1	1	1	0	1
	5	1	0	1	1	0

(b)

Each row in Table 1-1 represents the neighborhood for the pixel represented by the row number. Notice that the first pixel does not have a neighborhood, i.e. it is an isolated vertex from an out-degree perspective. Nodes can have no edges (in or out) and be truly isolated. This concept will be important when we discuss anomaly detection, and is a common occurrence in many of the methods to be discussed.

An HSI image with $O(10^5)$ pixels has an adjacency matrix of size $O(10^{10})$. The need for efficient encoding and/or approximation is evident. Adjacency matrices are sparse (mostly zeros) because $|NN| \ll n$, and it is common to find sparse matrix methods used to reduce storage and improve computational efficiency.

Edges may also be encoded with the magnitude of the relationship between vertices by replacing the unitary edge contributions in \mathbf{A} , with the value of a distance (similarity) measure. This weighted adjacency (affinity) matrix, $[\mathbf{W}]_{ij} = w_{ij}$, is composed of non-negative scalar values where $(v_i, v_j) \in E$, and zero otherwise (zero distances do not occur in practice due to natural spectral variability and noise). A graph may therefore be defined as $G = (V, E, w)$, where w is a mapping associating each edge of unitary contribution in \mathbf{A}

with a positive number representing vertex distance (similarity), i.e., $w: E \rightarrow (0, \infty)$, or simply $G = (V, \mathbf{W})$. We will see that this weighting matrix can be constructed from many different measures, especially those influenced by local spectral structure of the data.

Many common HSI graph construction methods result in simple graphs², so the asymmetric adjacency matrix resulting from the directed k -NN relationship (Figure 1-9a) must be modified. There are two basic criterion for creating simple graphs from directed k -NN graphs, resulting in symmetric adjacency matrices: superset symmetry and mutuality. The resultant graphs from these adjacency matrix symmetrification methods have very different characteristics. For example, Figure 1-10 displays the adjacency matrices, \mathbf{A} , for directed, superset symmetric, and mutual 3-NN graphs (top row). The directed 3-NN adjacency matrix shows a node's neighborhood list (a row) with three entries, i.e., $d^+(v) = 3$. The column to the right and the row to the bottom of each adjacency matrix display the out- and in-degrees, $d^+(v)$ and $d^-(v)$ respectively. The corresponding directed graph is shown below the adjacency matrix, where directional edge arrows are shaded similarly to their source node. Application of the superset symmetric and mutual criteria result in the adjacency matrices and graphs shown in the center panel and right panels respectively. Lightly shaded '1s' in the center column represent new adjacency matrix entries that symmetrify the directed 3-NN adjacency matrix by adding edges so that all directed edges become bidirectional (bold, no arrows), i.e., $d^+(v) = d^-(v)$. Notice edges span large regions of space and some nodes have more than three incident edges (e.g., node four has

² Simple and undirected graphs are not exactly the same, because undirected graphs can have self-loops and multiple edges between vertices. However, we use them interchangeably from this point forward.

six nearest neighbors). Dark shaded ‘0s’ in right column represent directed edges that are removed by enforcing the mutuality criterion. Notice the lack of edges spanning large regions of space (no edges to nodes six or seven) and no node has more than three incident edges, i.e., $d(v) \leq 3$. Notice the drastically different node in and out degrees for each adjacency matrix variant. This will be important in future chapters.

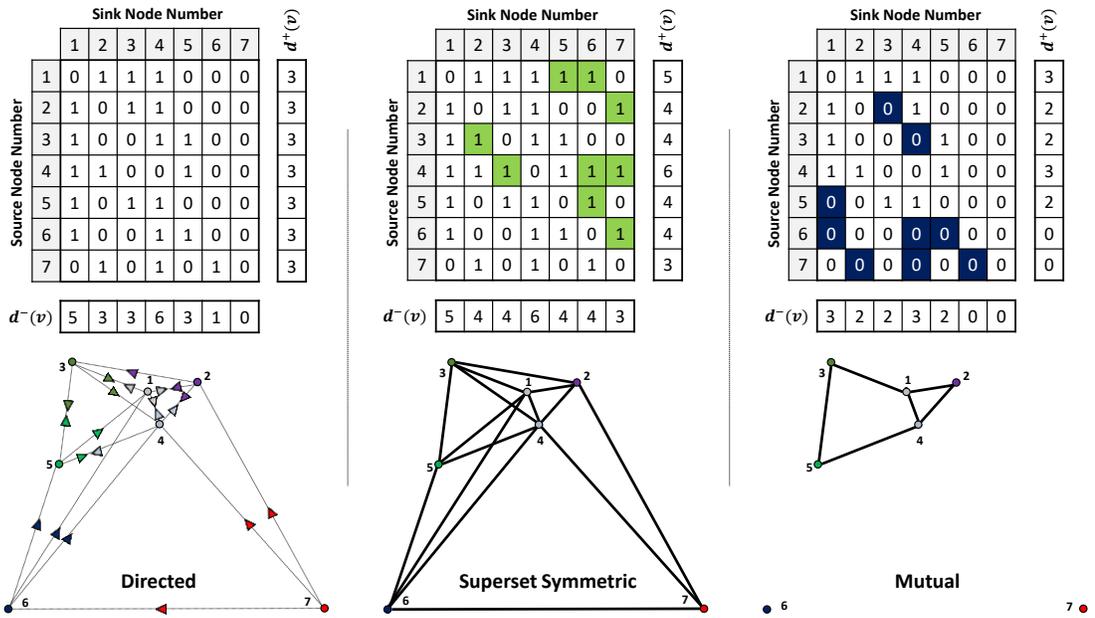


Figure 1-10. Directed, superset symmetric, and mutual 3-NN graph construction.

Generation of the superset symmetric k -NN graph is a trivial extension of the directed k -NN graph, where all vertex pairs, (v_i, v_j) , are connected if $v_i \in NN_k(v_j)$ **or** $v_j \in NN_k(v_i)$, where $NN_k(v)$ represents the k nearest neighborhood of vertex v , i.e., $W = \max\{W, W^T\}$. As a result, each node will have at least k neighbors with the node’s in-degree being proportional to the node’s local density (note that the node’s out-

degree is fixed at k) (Figure 1-10, center). Forcing adjacency-matrix symmetry in this manner—adding edges to create a bidirectional graph—may connect clusters of varying density, and permits edges spanning large regions of spectral space to overly connect outlier nodes. Stringers, long chains of single nodes, can also extend from virtually anywhere if the conditions are right. This can produce a larger subgraph diameter should they extend from cluster edges.

Generation of the mutual k -NN graph is also a trivial extension of the directed k -NN graph, where all vertex pairs (v_i, v_j) , are connected if $v_i \in NN_k(v_j)$ **and** $v_j \in NN_k(v_i)$, i.e., only existing bidirectional edges are retained ($W = \min\{W, W^T\}$). The resultant adjacency matrix is symmetric and a subset (subgraph) of the superset symmetric k -NN adjacency matrix (graph) (Figure 1-10, right). Forcing adjacency symmetry in this manner reduces the possibility of connecting clusters of varying density; hence edges typically do not span large regions of feature space, leaving outlier nodes unconnected from denser regions (Kontschieder, Donoser, and Bischof, 2009).

Adaptive variants of the fixed k -NN methods exist (Ziemann, Messinger, and Wenger, 2014b; Mercovich, Albano, and Messinger, 2011), where each node is assigned its own number of nearest neighbors, $k(i)$ ³, based on a user-defined criterion typically related to data density or scale. The aforementioned adjacency matrix symmetrification methods produce node degrees for v_i of at least ki and at most ki , for the symmetric and mutual criteria respectively. As such, the simple graph’s minimum and maximum node

³ We will abbreviate $k(i)$ as ki at times for ease of reading.

degrees, $\delta(G)$ and $\Delta(G)$ respectively, created by invoking the superset symmetry criterion (applies to directed graphs as well) are given by

$$\begin{aligned}\delta(G) &= \min(\mathbf{k}\mathbf{i}) \\ \Delta(G) &\in [\max(\mathbf{k}\mathbf{i}), n - 1],\end{aligned}\tag{1.3}$$

where $\mathbf{k}\mathbf{i}$ represents the vector of $k(i)$ values from the initial adaptive directed construction. $\Delta(G)$ has a maximum of $n-1$ because simple graphs do not have self-loops. The minimum and maximum node degrees for simple graphs created enforcing the mutuality criterion are given by

$$\begin{aligned}\delta(G) &\in [0, \min(\mathbf{k}\mathbf{i})] \\ \Delta(G) &= \max(\mathbf{k}\mathbf{i}).\end{aligned}\tag{1.4}$$

As seen in (1.3) the maximum node degree can potentially take on very large values in directed graphs or graphs created by invoking the superset symmetry criterion, while the maximum degree is restricted in graphs enforcing the mutuality criterion (1.4). We also observe that the only way to obtain isolated vertices in directed graphs or graphs created by the superset criterion is for $k(i) = 0$, a condition only possible with adaptive techniques, i.e., it is impossible for graphs constructed with a fixed positive k . Conversely, it is quite possible to have isolated vertices in graphs created by the mutuality criterion (1.4). We will see these node degree characteristics enable or inhibit the formation of vertices with special characteristics (hub vertices) in Chapter 6. Equations (1) and (2) apply to fixed k-NN graphs as well, where $k(i) = k, \forall i$.

Figure 1-11 displays an example spectral graph created from 2,643 vertices of a HSI dataset using the mutual 4-NN algorithm. The selected area consists of two homogenous regions and the border between them. Notice the significant edge density near the bulk of each distribution and the less dense sparsely connected region between the two clusters (mixing region).

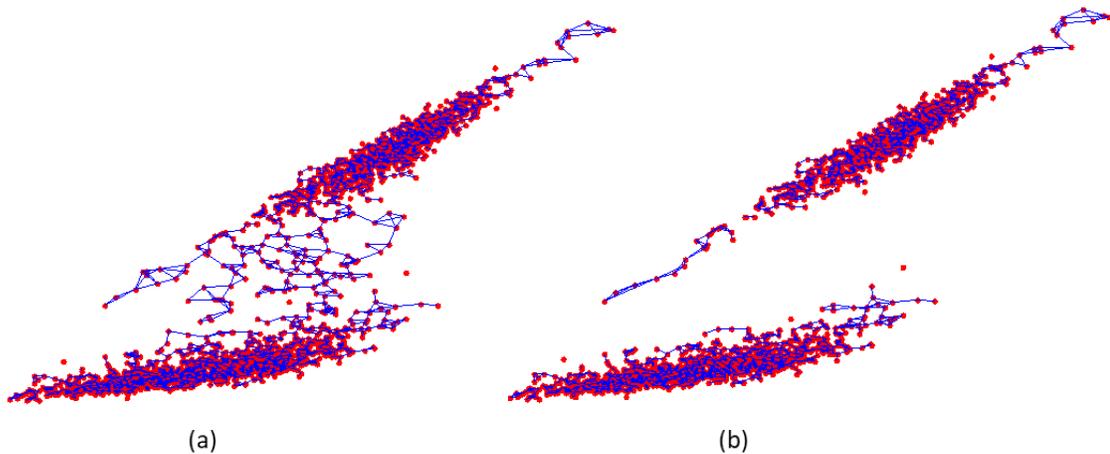


Figure 1-11. Two-dimensional graph created by the mutual 4-NN algorithm: a) with border regions, b) without.

Notice the isolated points not connected to either the dense or sparse regions in Figure 1-11; a condition resulting from the mutuality constraint. This is a common occurrence for many graph construction algorithms and must be mitigated if the graph analysis algorithm requires connected graphs, i.e., graphs in which a path exists between any two vertices in V by following edges in E .

The success of any graph-based analysis is intimately tied to the quality of the graph constructed from the spectral data (Mercovich et al., 2011). Edges must be judiciously chosen to accurately model community structure inherent in the data without over (under)

connecting the nodes. Many of these graph construction techniques are founded on the k -nearest neighbor (k -NN) relationship because of its adaptability to both scale and density as well as its ability to follow clusters of arbitrary shape. We will compare and contrast these methods in this research with the intent of providing a consolidated reference for these techniques as well as introduce new techniques addressing some of the weaknesses discovered in current methods.

1.4 Research Objectives and Contributions

Graph based analysis is a relatively new area of research in the HSI community. It has been just over a decade since its initial introduction, in contrast to the 30+ years of research into statistical and other geometric approaches. While much progress has been made, much research remains in order to fully understand the benefits and limitations of these techniques. The main benefit of graphical methods is the identification of clusters and trends of arbitrary shape, i.e., the data are not forced to predisposed distributional, functional, or geometric forms. For example, the MVN level sets indicated by the concentric rings in Figure 1-6 clearly do not model the data well, and the lower cluster in Figure 1-11 appears to have a slightly scalloped shape instead of a multivariate normal distribution. These characteristics can be better modeled through nonlinear graphical techniques instead of linear approximations (Prasad and Bruce, 2008).

A primary objective of this research is to advance the utility of graph-based approaches by improving the construction of the spectral graph itself. Of critical importance is the concept of spectral density (point density in spectral space) as it gives rise to the concept of a cluster—community structure within the data. Ideally, clusters

should be dense (small intracluster spacing) and be well separated from other clusters (large intercluster spacing). This is the basis of many common internal cluster-validation techniques that ratio cluster separation to cluster dispersion. So compact clusters with big separation are good, and wide close clusters are bad. However, clusters are generally not “well” separated in HSI data. Mixing trends connect dense clusters due to boundary pixels partially filled with multiple materials. For example, the sparse area between the two clusters shown in Figure 1-11a are mixed pixels along the boundary between materials. This is evident in Figure 1-11b which reproduces the figure, but with boundary pixels removed.

Mixed pixels are very much a part of HSI data structure and need to be represented. However, their importance in the data mode should not hold the same weight as those pixels in the denser communities of points where it should be easier to move between vertices. This deweighting can be encoded in the numbers of and/or strengths of the edges in the spectral graph. Spectral density metrics can be used to modify graph connectivity on a per pixel basis such that pixels in dense clusters are highly connected while those in less dense mixing trends are loosely connected. Note, it is possible mixed pixel regions can be very dense depending on the shape and size of the clusters with respect to the IFOV. For example, in the case where each pixel is comprised of multiple materials, every pixel will exist in these intermediate locations. Density metrics in this case would strongly connect the mixing trend. However, spatial correlations typically generate natural groupings in imagery, so this situation rarely arises.

The objectives detailed below describe focused areas of research targeted to improve graphical models of spectral data. Evaluating the improvement of the construction itself is difficult, so structure testing should be supplemented with applications that utilize graphical models, proving that any measured improvements in graph characteristics actually lead to improvements in graph-based analyses.

1.4.1 Survey spectral-graph construction methods

Many graph construction techniques have been described and employed in the HSI literature. However, no consolidated reference of techniques exists to our knowledge. As such, the first step in this research is a literature review of all utilized techniques and translate them into a common construct facilitating easier comparisons. This consolidated review was released as a technical survey paper to the remote sensing community through IEEE Transactions on Geoscience and Remote Sensing (TGRS) in August 2017 (Stevens, Resmini, and Messinger, 2017), filling a documentation gap in the remote sensing literature. This paper provides guidelines for parameter selection based on rigorous performance comparisons.

1.4.2 Develop new spectral-graph construction techniques

Armed with an understanding of the existing techniques, we focused on benefits and limitations of each technique, with the aim of generating and/or modifying new density-based graph-construction techniques to overcome discovered limitations. In doing so, we described the benefits and limitations towards various HSI analysis tasks such as classification, change detection, anomaly detection, and target detection.

Hyperspectral data are similar to extremely large databases with instances comprised of hundreds of attributes. In this case, the instances are pixels, and the attributes are the spectral bands. As such, we explored how the data mining community, and more broadly, the machine learning community creates graphs from large datasets with high numbers of attributes, and subsequently modified those techniques for spectral data. This development addressed a current research gap in that limited graph construction techniques have been investigated to date, so any new methods increase the number of tools available to the remote sensing community. These methods were also detailed in the 2017 IEEE TGRS paper.

1.4.3 Study Impacts of High-Dimensional Data

Given the high extrinsic dimension of hyperspectral data, it became evident a deeper probe into characteristics of high-dimensional data as they relate to graph construction was required because many of the studied methods utilize distance and similarity measures that are subject to the *curse of dimensionality* (Bellman, 1961). This led to studies on high-dimensional phenomenologies—specifically the concentration of measure and hubness—as they relate to graph construction.

1.4.4 Develop Classification and Anomaly Detection Algorithms

New measures of spectral similarity and graph creation provide the opportunity to modify existing state-of-the-art algorithms to leverage different types of information. As such, modification to two variants of the Schroedinger Eigenmaps with potential matrices (Czaja and Ehler, 2012; Cahill, Czaja, and Messinger, 2014) for classification and anomaly detection are presented as initial explorations into use of these new measures.

1.5 Dissertation Organization

The remainder of this dissertation is organized as follows. In Chapter 2, we introduce h simple constructs demonstrating basic concepts and then incrementally add components to derive more complex approaches. Two new methods from the data mining and machine learning community, SNN and MP, are introduced in Chapter 3. All discussed methods are compared in Chapter 4, where we develop guidelines for the application of each method based on parametric studies into model parameters. We examine the impacts of high dimensional data on graph construction in Chapter 5, where we tie these effects to codensity. Chapter 6 provides a review of several graph-based HSI analysis algorithms, leading to the incorporation of SNN and MP into Schroedinger Eigenmaps in Chapter 7. We conclude with contributions and possible future research in Chapter 8.

2

COMMON HSI GRAPH CONSTRUCTION TECHNIQUES

There are a plethora of construction techniques found in the HSI literature, each introduced with its own, and sometimes different nomenclature. As such, we offer a survey of these techniques within a common framework, starting with basic graph construction techniques and then followed by those influenced by data density from which the concept of community structure arises. Some of the algorithms presented in the literature are admittedly not very useful for HSI analysis; however, they do provide good conceptual content that facilitates building more advanced concepts. That same approach is followed here, and these instructional techniques are noted as such.

2.1 ϵ -Threshold Graphs (aka ϵ -NN Graphs)

The threshold graph is very simple to construct and the fastest of all described methods. Given all pairwise distances (or similarities), an undirected edge is placed between two nodes if the distance, $d(v_i, v_j)$ between them is less than a user-defined constant $\epsilon > 0$. Specifically, $e_{ij} = (v_i, v_j)$ is added to the edge set, E , if v_j lies within the hypersphere of radius ϵ constructed about v_i in spectral space, i.e., $e_{ij} \in E$ iff $d(v_i, v_j) \leq \epsilon$ and $i \neq j$. The similarity based equation is obtained by replacing $d(v_i, v_j)$ with $s(v_i, v_j)$ and switching the direction of the inequality. This technique is not adaptive to scale or density because ϵ is a fixed global threshold (Kontschieder et al., 2009), generally producing unconnected graphs.

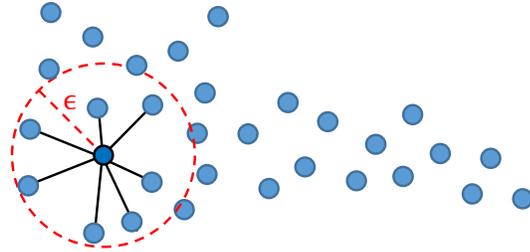


Figure 2-1. ϵ -threshold graph construction.

Hyperspectral data typically displays a multitude of densities, so this technique is not widely used, but provides the foundation for understanding other graph construction techniques.

2.2 k -Nearest Neighbor (k -NN) Graphs

Nearest neighbor graphs are very common graph construction techniques in many disciplines. An edge is placed between v_i and v_j if v_j is among the k -NN of v_i . The user defined parameter, k , is a global parameter indicating the number of edges exiting v_i , i.e., its out-degree. Given each node has its own set of k -NN, this relationship is not symmetric and therefore produces directed edges leading to an asymmetric adjacency matrix (von Luxburg, 2007). These graphs are also called directed k -NN graphs for this reason.

Construction of the k -NN graph is conceptually very simple, but computationally expensive due to the evaluation of all pairwise distances and subsequent sorting. Many traditional indexing methods (e.g., R-tree, k-d tree) fail in high dimensional spaces such that exhaustive searching for nearest neighbors can outperform even the most complex indexing scheme (Hinneburg, Aggarwal, and Keim, 2000). As such, several fast nearest

neighbor methods have been developed to generate approximate or exact k -NN lists (Merkwirth, Parlitz, and Lauterborn, 2000).

k -NN graphs are locally adaptive to both density and scale which makes them particularly well suited to model clusters of varying density (Albano et al., 2012b), and shape (Ertöz, Steinbach, and Kumar, 2003), or follow mixing trends between clusters. Unfortunately, a global k -NN construction tends to over connect vertices in low density regions since the nearest neighbor may span a significant distance (or similarity). Another difficulty is selection of the user defined parameter k ; values from five to 60 are common (Mercovich et al., 2011; Albano et al., 2012a) in the HSI literature. Selecting k too high tends to over connect the graph, whereas selecting k too low leaves the graph disjoint. This sensitivity to k is true of many k -NN variants and has prompted the development of adaptive algorithms that provide node-specific connectivity, $k(i)$.

Many graph construction and analysis techniques promote, or require undirected (bidirectional) edges, i.e., simple graphs, so we need to modify the asymmetric adjacency matrix resulting from the directed k -NN relationship. There are two means of creating simple graphs from directed k -NN graphs, producing symmetric adjacency matrices: superset symmetry and mutuality. Mutual and superset symmetric k -NN graphs were described in the introduction and are not replicated here. Symmetric adjacency matrices provide for added benefits as well. For example, memory requirements are cut in half because only the upper or lower triangular portions of the adjacency matrix need be stored. This can be quite substantial for even average sized HSI cubes. Additionally, sparse matrix operations operating on only the triangular portions can reduce computation time.

All k -NN variants can produce disconnected graphs, so post processing is often required to ensure graph connectivity if required by the analysis algorithm (Chapter 2.6). Relatively speaking, graphs created by invoking the mutuality criterion are more disconnected than their directed or superset symmetry counterparts. An example of a mutuality-induced disconnected graph is shown in the upper right corner of Figure 2-4.

2.3 Density Weighted k -NN (DW k -NN) Graphs

Dense groupings of points in feature (spectral) space share similar attributes of similar magnitude and are therefore related. It makes intuitive sense these similar intracluster nodes should be more heavily connected to each other than to extracenter nodes. Indeed, Kameshwaran and Malarvizhi (2014) state that density based measures are the key to finding nonlinear structure, and we will find they are used extensively in HSI graph generation, both implicitly or explicitly. Given HSI data clusters exist at varying scales and density, we desire adaptive algorithms that adjust to hyperspace characteristics proximal to a pixel's location. Additionally, the traditional notion of Euclidean density can be meaningless in high dimensional spaces due to the exponential growth in d -D volume (Ertöz et al., 2003), and even the discriminating power of Euclidean distance has been questioned in high dimensional spaces (Beyer, Goldstein, and Ramakrishnan, 1999). We note this here because many of the algorithms presented utilize traditional density-based constructs, and we should be mindful of the possible characteristics of high-dimensional spaces with their use.

Mercovich et al. (2011) introduced the concept of density weighted k -NN to encode stronger relationships between similar nodes and promote more effective clustering by

minimizing the impact of extracuster pixels. The first step in the process is to assign each pixel a codensity (distance) score given by

$$\delta(v_i) = \left(\frac{1}{k_{\max} - k_{\min} + 1} \right) \sum_{k=k_{\min}}^{k_{\max}} w_{ik}, \quad (2.1)$$

where k represents the indices of node i 's nearest neighbors provided in non-decreasing order. Distances to a range of NN is specified through k_{\min} and k_{\max} define the range of k values to average. Mercovich uses $k_{\min} = 1$ such that $\delta(v_i)$ represents the average distance of the pixel's k_{\max} neighbors. A node-specific number of neighbors, $k(i)$, is then assigned to each node based on its position in the histogram of codensity measures (indicated by the node's z -score). Mercovich found assignments based on integral z -score values in the range $[-3,3]$ worked well for HSI clustering. The maximum $k(i)$ assigned to a pixel does not need to be the same as the number of neighbors used in codensity estimation, i.e., any maximum (ki_{\max}) and minimum (ki_{\min}) number of edges can be assigned to a pixel based on its z -score. As such, we generalize the mapping equation such that $k(i) \in [ki_{\min}, ki_{\max}]$. Mercovich set $ki_{\max} = k_{\max}$.

Figure 2-2 shows a representative codensity distribution and the node specific connectivity count, $k(i)$, assigned by this method using $ki_{\max} = 30$, $ki_{\min} = 1$, six z -score regions, $k_{\min} = 1$, and $k_{\max} = 5$. The percentage of pixels assigned to each bin is shown above each step. Assigning $k(i)$ in this manner results in a large number of nodes with $\sim ki_{\max}/2$ edges and far fewer nodes with 1 or ki_{\max} edges as shown by the k

mapping overlaid on the codensity histogram in Figure 2-2. The k mapping is a quantized inverse function of codensity where pixels residing in lower density regions (higher codensity) receive few edges (towards ki_{min}) and pixels in high density regions (low codensity) receive more edges (towards ki_{max}).

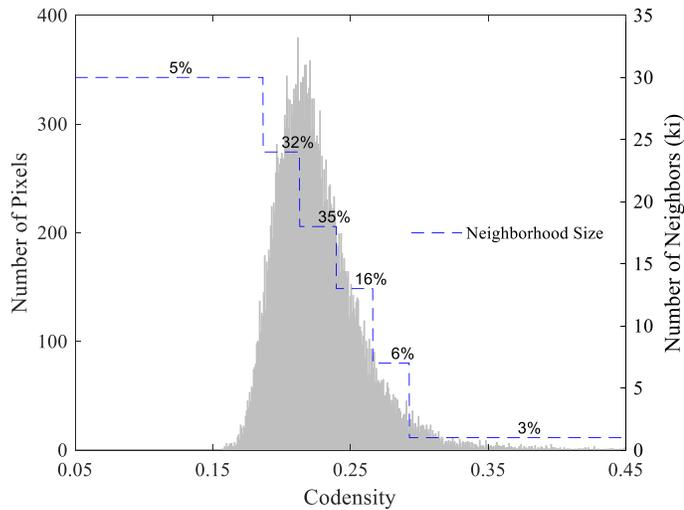


Figure 2-2. Codensity histogram with node-specific NN assigned by DW k -NN.

Mercovich et al. (2011) note that codensity distributions can take on non-normal forms based on scene content, but the normal assumption (through use of z -score) worked well for clustering. While some codensity distributions may appear relatively normal, they are never rigorously normal. We verified this by testing 56 chips of varying scene content, from different sensors, and spatial resolutions via the Kolmogorov-Smirnov test ($\alpha=0.005$). In no instance did the test report normality under these loose conditions. Most codensity distributions were unimodal with a positive skew (like Figure 2-2). Albeit, z -score use under unimodal conditions is a reasonable approximation for many codensity distributions.

HSI codensity distributions are more normal with increasing number of bands as seen in Figure 2-2 (128 bands) if the data have higher intrinsic dimensionality (Francois, Wertz, and Verleysen, 2007). However, codensity distributions from a small number of bands are generally not normal. This non-normal behavior changes the intended mapping of z -score based methods, so care must be taken with their application. In these situations the cumulative density function (CDF) may be substituted for z -score, providing a method to drive connections from ki_{max} to ki_{min} independent of the codensity functional form while maintaining the intended mapping.

For anomaly detection, the above ki_{min} can be appropriate. However, for classification, ki_{min} should be set on the interval [5,10] to avoid overly disconnecting pixels upon invocation of the mutuality criterion. This range was very consistent while trending many graph types over large ranges in k , improving absolute classification accuracies by upwards of ~4% percent (~2% on average).

2.4 Adaptive Nearest-Neighbor (ANN) Graphs

The natural nearest neighbor (NNN) graph described by Zou and Zhu (2011) provides a data-driven parameter-free method for the generation of node specific connectivity. This method was first applied to hyperspectral imagery by Ziemann, Messinger, and Albano (2013) while exploring target detection methods based on manifold approximations recovered from spectral data. The novel aspect of NNN is that there are no user-defined parameters and the algorithm autonomously produces a node-specific connectivity proportional to local density.

The NNN method maintains a list of the number of times node v_i has been identified as a neighbor by another node v_j , i.e., v_i 's in-degree or reverse nearest neighbor count. Node in-degrees are counted by sequentially examining the r^{th} nearest neighbor for every pixel where $r = 1, \dots, k_{max}$. For example, the first iteration adds edges between each node and its first nearest neighbor. $k(i)$ is incremented anytime pixel i appears as pixel j 's r^{th} neighbor. This process continues until every pixel has been declared a neighbor by another pixel. Naturally, nodes located in regions of high spectral density appear as neighbors of other nodes more often than those in low density regions so their in-degrees will be higher. The algorithm connects each node to its $k(i)$ nearest neighbors given by $NN_r(i)$ ⁴ once the stopping criterion has been met. Isolated vertices will not exist because each pixel is guaranteed to be a member of at least a two-pixel connected component because $k(i) \geq 1, \forall i$ is a termination requirement. High and low $k(i)$ values infer high and low density regions respectively. Like k -NN, this method generally produces unconnected graphs, so post processing methods to ensure connectivity are required for some analytical techniques.

As intuitively appealing as the NNN construction may be, it does suffer from a pathological condition that can produce prohibitively long execution times and lower utility. Assume a node exists that is distant from every other node. Given NNN iterates until each node has been declared a neighbor by another node, the distant node will cause

⁴ Note that $NN_r(i)$ is not v_i 's reverse NN list, but rather v_i 's nearest neighbor list.

$n - 1$ iterations until termination, producing a complete graph (each pixel is connected to every other pixel). This is obviously not the intent.

This sensitivity to isolated nodes led Ziemann et al. (2013) to add another stopping criterion (Ziemann *et al.*, 2014b). Iteration terminates if $k(i) \forall i$ are unchanged after examining the r^{th} nearest neighbors. The algorithm as described by Ziemann, is provided below where $nn_r(i)$ is the r^{th} nearest neighbor of pixel i , $NN_r(i)$ is the set of r neighbors of the i^{th} pixel, and $NB_0(r)$ is the number of nodes that have yet to be declared a neighbor by another node, i.e., those with $k(i) = 0$.

1) Initialize

$$r = 1, k(i) = 0, nn_0(i) = \emptyset, NN_0(i) = \emptyset, n = |V|, NB_0(0) = 0$$

2) Find nearest neighbors

$\forall i$, calculate the r^{th} nearest neighbor, $nn_r(i)$, and set $NN_r(i) = NN_{r-1}(i) \cup \{nn_r(i)\}$.

3) Identify reverse nearest neighbor counts

$\forall i$, count the number of times i occurs in $NN_r(j)$, $j = 1, \dots, n$ and set $k(i) = \text{count}$.

a. if $NB_0(r) \neq NB_0(r - 1)$ continue to 3b else end

b. If $\exists i$ such that $k(i) = 0$, increment r and return to step 2 else proceed to step 4.

4) Build graph

Connect v_i to its $k(i)$ nearest neighbors from $NN_r(i)$.

The additional stopping criterion 3a helps deter prohibitively long run times. Changing a single node at each iteration can still produce long execution times, but this has not been observed in practice. Ziemann's variant is called the adaptive nearest-neighbor (ANN) graph and is the starting point for many adaptive methods to follow. ANN may still produce isolated pixels and post processing may be required to ensure graph connectivity.

2.5 Edge Reweighting

The graph construction methods previously described create a graphs wherein edges are weighted by the metric used to determine distance (or similarity) to each node. These metrics are called primary metrics as they are used to determine initial edge weights. One may also change (reweight) initial edge weights based on some function or heuristic to reinforce particular properties from the primary metric space. In this research edge reweighting is based on measures of local spectral density, producing spectral-density weighted secondary measures. Reweighting does not change the structure of the graph, only the weights assigned to the edges, so in essence it is not a graph construction technique per se, but is addressed as such for continuity of the discussion. Note that some reweighting schemes we will encounter can sever edges by setting $w_{ij} = 0$, so they could be considered construction methods in their own right.

The non-iterative contextual dissimilarity measure (NICDM) introduced by Jegou, Schmid, Harzallah, and Verbeek (2010) rescales distances based on measures of each node's local scale as shown in (2.2). This method has not previously been used on HSI data, but provides a nice introduction for multiple methods in remaining subsections.

$$w_{ij} = \text{NICDM}(v_i, v_j) = \frac{d(v_i, v_j)}{\sqrt{\sigma_i \sigma_j}} \quad (2.2)$$

The spectral Euclidean distance, $d(v_i, v_j)$, is rescaled by the geometric mean of the local scaling parameters for each pixel, σ_i and σ_j . Jegou defines the local scaling as the distance to the k^{th} nearest neighbor, but any of the codensity measurements previously

described are viable local scale measures. Pixels i and j are therefore only similar if the spectral Euclidean distance between them is small relative to each pixel's local spectral scale⁵.

Zelnik-Manor and Perona (ZMP) (2004) introduce a locally adaptive scaling that transforms primary distance metrics into scaled affinities by taking the exponential of the negative squared NICDM as shown in (2.3), i.e., it is a radial basis function (RBF), or heat diffusion kernel, in spectral distance with parameters σ_i and σ_j .

$$w_{ij} = ZMP(v_i, v_j) = \exp\left(-\frac{d(v_i, v_j)^2}{c\sigma_i\sigma_j}\right) \quad (2.3)$$

Parameters are defined as per (2.2), but the negative exponential results in a rescaled space wherein the new similarity, $s_{ij} \in (0,1]$. Zelnik-Manor and Perona also used the distance to the k^{th} nearest neighbor, but again any scale determination method previously discussed is applicable. The additional scaling parameter c , is included to provide user control over the support region of the RBF, and is not in Zelnik-Manor and Perona (2004) formulation, but we add it here to facilitate future discussion.

Of course we're free to simply choose a single scale ($\sigma_i = \sigma_j, c = 1$) and take advantage of the heat diffusion mapping in its simplest form (2.4), which is the initial method of Shi and Malik (2000). Using the heat kernel with a fixed scale is a very common method of creating similarity matrices, and has been used by many authors (Hou, Zhang,

⁵ NICDM is not a true distance metric as it only satisfies positivity and symmetry axioms, and not the triangle inequality; see Jegou et al. (2010).

Ye, and Zheng, 2013; Cahill, Chew, Wenger, 2015; Benedetto et al., 2012a; Gillis and Bowles, 2012) in one regard or another. It provides a connection between the Laplace-Beltrami operator on a manifold and the graph Laplacian for the HSI data (Belkin and Niyogi, 2007). More on this aspect in Chapter 7.

$$w_{ij} = \exp\left(-\frac{d(v_i, v_j)^2}{\sigma^2}\right) \quad (2.4)$$

Equation (2.4) method is known as non-local means in the image processing literature if the sum of the weights for each pixel is normalized to one (Buades, Coll, and Morel, 2005).

He, Zhang, Wang, and Li (2009) use a simple global scaling parameter to control the influence of the heat diffusion kernel applied to a distance matrix. Weights are defined by

$$w_{ij} = \exp\left(-\frac{d(v_i, v_j)}{2\epsilon}\right), \quad (2.5)$$

where $d(v_i, v_j)$ denotes the cosine distance between pixels and w_{ij} is the pixel similarity measure derived from the cosine distance. ϵ is a user-defined constant controlling scaling properties. Squaring the exponential argument is the same weighting scheme used by Shi and Malik (2000) with $\sigma = 2 \epsilon$.

We model an example after that of Schnitzer, Flexer, Schedl, and Widmer (2012) to demonstrate the effectiveness of locally adaptive scaling. Figure 2-3a displays a simple 2-D grid of points representing three clusters of differing density. A Delaunay triangulation

is used to define edges between vertices to emphasize weight rescaling vice potential differences due to graph construction techniques (missing edges). Edge weights are initially set to the Euclidean distance between vertices and shaded such that bolder lines indicate stronger similarities due to shorter distances (b).

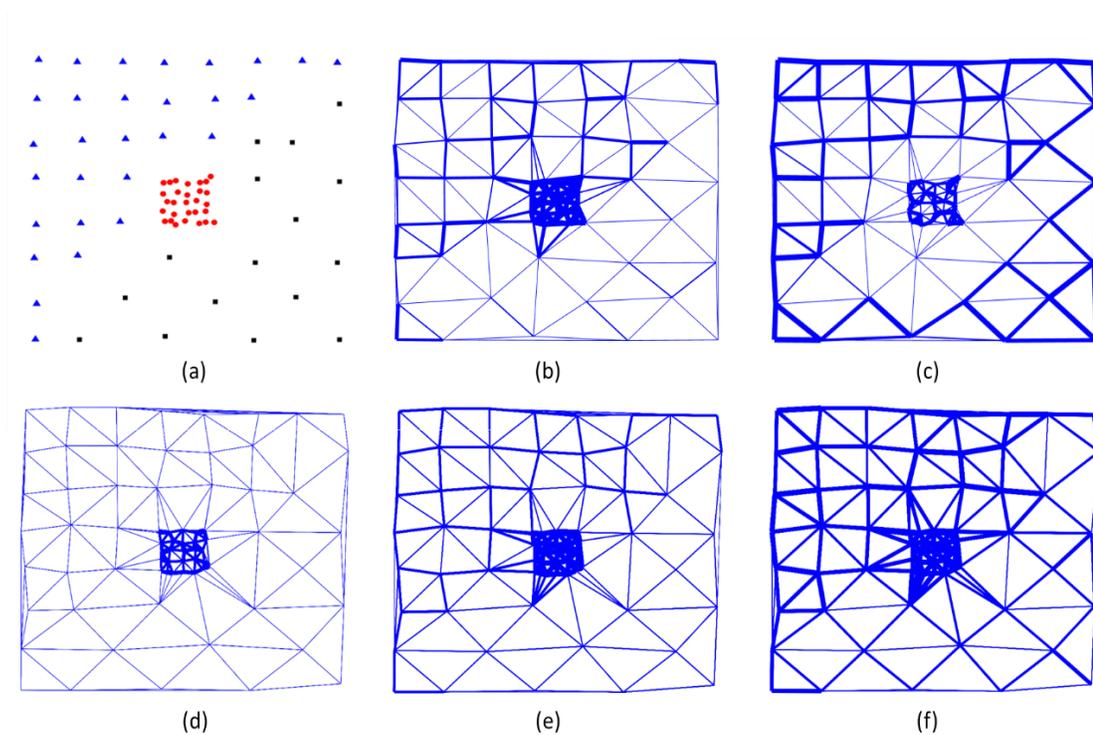


Figure 2-3. Example dataset displaying three clusters of different density and edges weighting methods.

The shorter the distance, the more similar the nodes and the thicker the edge. Notice nodes in the less dense outer clusters appear as dissimilar despite being quite similar (same average distances and density) to their surroundings. The thickest edges (shortest distances) are in the dense center cluster because Euclidean distance is a global measure. The resultant edge weights after the application of ZMP locally-adaptive scaling is shown in Figure 2-3c. Notice the stronger relationship between pixels in the lower density clusters and the weaker

edges in transition regions. Additionally, weaker edges are now apparent in the dense center cluster because weights are now relative to local density instead of a global distance scale. Figure 2-3d-f demonstrate a global heat kernel (2.4) at three scales commensurate with each cluster. Notice that the only the densest cluster has strong weights at a small local scale in (d). As the scale increases, the densest cluster becomes even more strongly connected, but not the intermediate cluster weights begin to get stronger. Lastly, at the largest scale, weights for the least dense cluster begin to strengthen while the two denser clusters are even stronger. Note that the data points shown in (a) need not be three clusters, but simply three regions of different density within the same cluster.

Other variants are also possible, albeit we focus on mainly the adaptive measures for this research given its advocacy in the literature.

2.6 Ensuring Graph Connectivity

There are two primary methods to transform disjoint graphs into connected k -NN graphs: merger with a spanning tree or spatially-connected neighborhoods. Both of these methods can be applied to any graph variant enabling analysis using algorithms requiring graph connectedness.

Merger with a minimum spanning tree (MST) is conceptually and practically simple. First, generate a MST that identifies the minimum weight tree using a subset of edges from the fully connected graph on the vertices, creating an undirected path to every vertex (Kruskal's or Prim's algorithm are examples). Supplementing the initial graph with MST edges to ensure connectivity is a simple matrix operation that can be performed in linear time (or less). Let W be the original weighted adjacency matrix, W_{MST} be the MST

weighted adjacency matrix, and W_C be the connected graph, then $W_C = \max\{W, W_{MST}\}$. Edges present in both adjacency matrices remain, while MST edges are added (Figure 2-4).

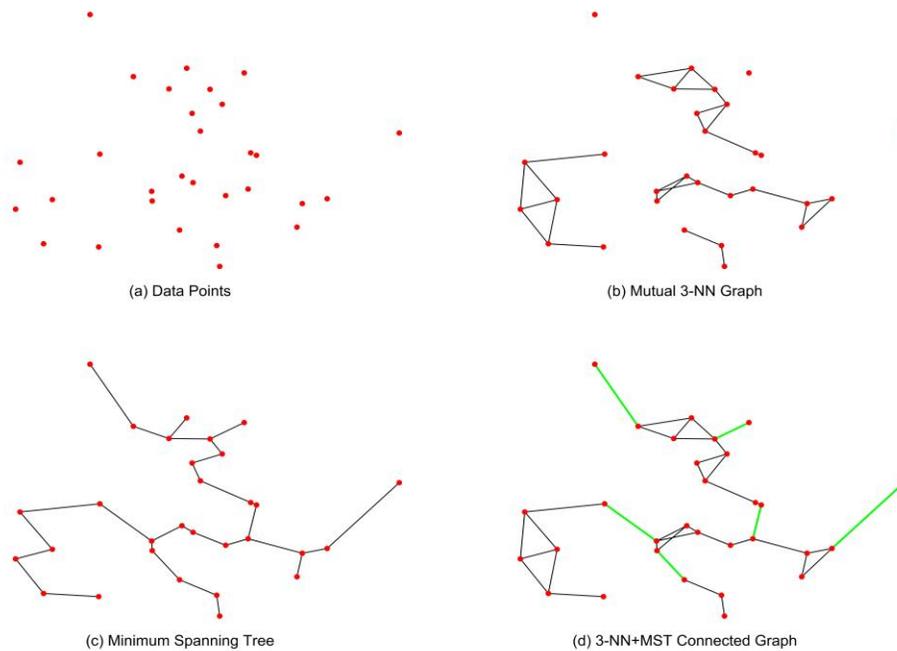


Figure 2-4. Connecting a graph via minimum spanning tree. (a) data points, (b) 1-NN graph, (c) MST, (d) k -NN+MST graph. Green edges are MST additions.

As can be seen, the disjoint graph in Figure 2-4(b) becomes connected in (d) by the addition of MST edges from (c), indicated in green.

The connected-neighborhood method takes advantage of the high spatial correlation present in imagery, where adjacent pixels have a high probability of possessing similar composition. This is easily visualized by thinking about a grassy field, parking lots, or sandy beaches. This assumption does break down at material boundaries, but largely holds across the scene (Shi and Malik, 2000). As such, connecting any pixel to its proximal neighbors using four- or eight-connected neighborhoods is a reasonable approach towards

connecting graphs (border pixels are unwanted edges, but these are fewer in number). Practically, this is accomplished through a logical OR of the proper subdiagonal patterns shown in Figure 2-6 with the graph's adjacency matrix.

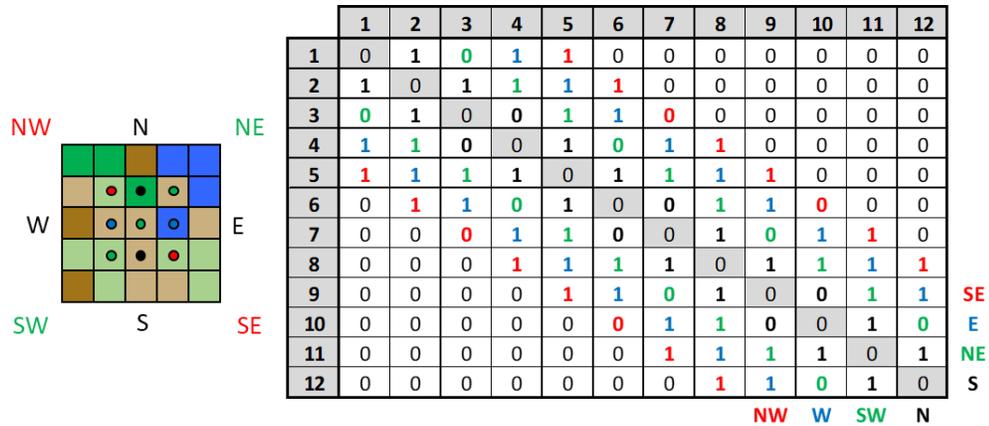


Figure 2-5. Encoding spatial relationships into an adjacency matrix.

Distances do not need to be calculated as proximal pixels are completely defined by their row and column indices. The two methods can also be used together. For example, assume all pairwise distance measures are not available from the initial construction method, e.g., fast k -NN algorithms do not return a matrix of all pairwise distances. An MST can always be extracted from a spatial four or eight connected graph, facilitating application of the MST method to any graph construction method.

2.7 Spatial-Spectral Methods

Imagery is spatially correlated. i.e., an adjacent pixel has a high probability of being composed of the same material as the pixel being studied. As such, it makes logical sense to include spatial information while constructing the graph or performing analysis,

especially segmentation, clustering, or classification. Three common techniques found in the literature are described in the sections to follow.

- Forcing relationships between proximal neighbors
- Modifying edge weights
- Fusing information while remapping the data

Each of them can be combined with any of the graphs built from the spectral data described in the previous sections.

2.7.1 Forcing relationships between proximal neighbors

A spatial-spectral approach called locally weighted NN is described by Mercovich et al. (2011). Pixels are connected to their $k = g + l$ nearest neighbors, where g represents a number of global neighbors determined by the k -NN algorithm (any of the fixed k -NN based methods previously described would suffice) and l represents the number pixels selected from a local spatial neighborhood centered on the pixel of interest. The l pixels in the local region with the lowest spectral angle (θ_{ij}) from the pixel of interest are used by Mercovich, but any other heuristic or metric could be used. The method may also be generalized to any technique producing a local neighborhood about each pixel, e.g., superpixel techniques that produce arbitrary shapes instead of rectangular regions centered on the test pixel. Figure 2-6 provides a toy example indicating the local and global pixels using triangles and dots at the pixel center respectively. A pixel's local neighborhood is indicated by a dotted black box centered on the pixel of interest.

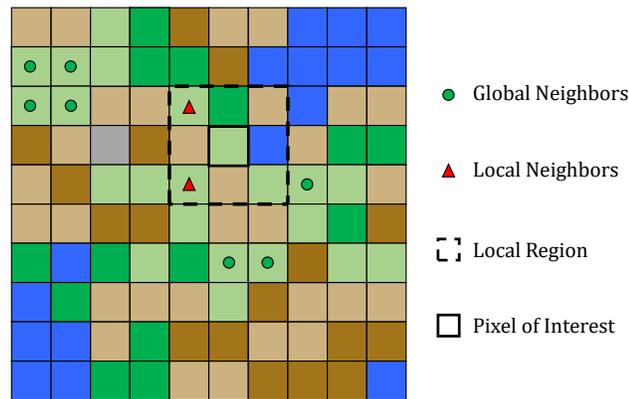


Figure 2-6. Locally-weighted NN connects a pixel with global and local neighbors (Mercovich et al., 2011).

The best global neighbors may also be the best local neighbors, so each pixel is not guaranteed to have an out degree of k , albeit the algorithm could be modified to ensure k neighbors from outside the $r \times r$ window. This graph is also directed and can be made simple (undirected) by one of the two methods previously discussed. Mercovich describes performance with five local and 25 global neighbors. Note that the method extends to adaptive k -NN methods with $k(i) = g(i) + l$ pixels, where $g(i)$ represents the number of adaptive neighbors.

Increasing local connectivity in this manner facilitates segmentation, clustering, and classification if regions demonstrate spatial coherence (Mohan, Sapiro, and Bosch, 2006). Additionally, using spectral angle within a local region may help connect pixels composed of similar materials but with varying illumination, e.g., in and out of shadow.

Spatially connecting pixels to their four- or eight-connected neighborhoods can be interpreted as a variant of the method where the selected local pixels are always based on relative position to the pixel of interest instead of a similarity metric.

Gillis and Bowles (2012) encode both spatial and spectral information into weights associated with only a pixel's spatial neighbors. Constructing the graph in this way reduces the number of edges and facilitates faster solutions due to sparser matrix operations. Weights are defined by a spectrally-modified spatial diffusion kernel given by (2.6) for all pixels within the $r \times r$ region centered on the pixel of interest.

$$w_{ij} = \exp(-\theta_{ij}) \cdot \exp\left(-\frac{\|v_i - v_j\|^2}{\sigma}\right) \quad (2.6)$$

θ_{ij} denotes the spectral angle between v_i and v_j , $\|v_i - v_j\|^2$ is the squared spatial distance, and σ is a user defined spatial-scaling parameter. This weighting has a similar general form (feature space metric times a spatial window) to that described in Fan and Messinger (2014) or Shi and Malik (2000), but the weights are only encoded into pixels within the $r \times r$ spatial neighborhood, where $r \in [2,7]$.

2.7.2 Modifying edge weights

Fan and Messinger (2014) describe a graph-based hyperspectral image classification technique based on a split/merge paradigm using normalized cuts with a locally adaptive spatial-spectral graph. The spatial-spectral similarity graph is based on a Shi and Malik (2000) construction composed of the product of feature (spectral) similarity and spatial proximity terms. Fan and Messinger's adaptation replaces the feature space term with a modified ZMP variant using locally adaptive scaling as shown in (2.7).

$$w_{ij} = \exp\left(-\frac{d(v_i, v_j)^2}{c_{ij}\sigma_i\sigma_j}\right) \cdot \exp\left(-\frac{\|v_i - v_j\|^2}{\sigma_d^2}\right) \quad (2.7)$$

The left hand side of (2.7) represents a Gaussian weighted locally-scaled spectral Euclidean distance, i.e., a spectral diffusion function similar to that in the ZMP method. The local scaling parameters, σ_i and σ_j , are defined in Zelnick-Manor and Perona (2004) as the distance to the k^{th} nearest neighbor; however in this treatment $k(i)$ is determined by the ANN method.

The additional scaling term in the denominator (c_{ij}) is an integer indicating the number of common neighbors between nodes i and j . This modification reduces (or eliminates) the impact of joining disparate regions by increasing the weight between nodes in the same cluster. Figure 2-7 shows two cases where the weight between AC is increased compared to that of AB despite being the same distance away because A and C have common neighbors. To the left of Figure 2-7, the common neighbor adjustment increases the similarity indicated by edge AC despite pixel A and B having the same density. In contrast, to the right of Figure 2-7, the AC similarity is increased because of the lower relative density (larger local scale) of pixel C to pixel A ($\sigma_B < \sigma_A$). In short, intuitively nodes A and C are more similar than nodes A and B due to their community structure. This aspect is reinforced via use of common neighbors in modifying the edge weights (2.7)—a concept we will exploit in Chapter 3.

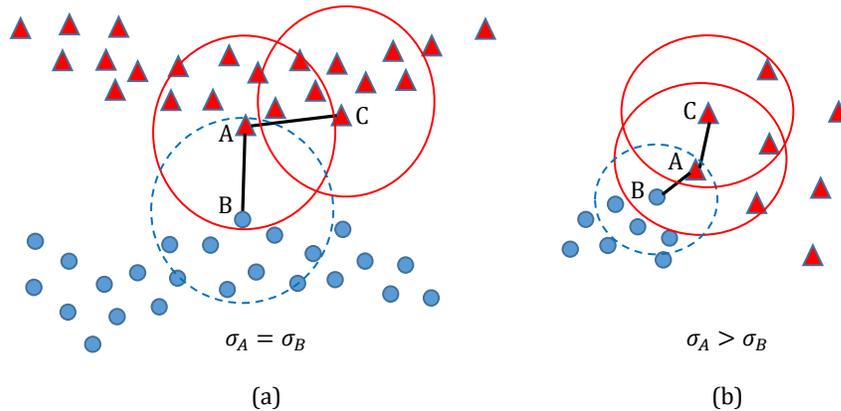


Figure 2-7. Common nearest neighbors indicate community structure (modified from Fan and Messinger, 2014).

The right hand side of (2.7) is a spatial diffusion (dampening) adjustment that performs two tasks: 1) it enforces connections between spatially proximal pixels (small $\|v_i - v_j\|$), exploiting spatial coherence present in imagery and 2) inhibits growth of larger disjoint scene-wide clusters despite any apparent spectral similarity. As such, the graph becomes overly segmented and must undergo a merging step for clustering or classification. The spatial dampening constant, σ_d , limits the size of spatially contiguous regions and is the only user-defined parameter.

The lack of common neighbors severs the edge, so technically this can be considered a graph construction technique instead of an edge reweighting method. Methods that spatially window data have the potential to sever spectrally-similar edges, so they can be considered graph construction techniques as well. Should spatial terms simply modify an existing weight, reweighting is more appropriate.

Hou et al. (2013) define a spatial-spectral distance measure called the joint spatial-pixel characteristic distance (JSPCD) given by

$$d(v_i, v_j) = \left(1 - \exp\left(\frac{d(v_i, v_j)^2}{2\sigma^2}\right) \right) \cdot \left(1 - \exp\left(\frac{\|v_i - v_j\|^2}{2\sigma_d^2}\right) \right), \quad (2.8)$$

where all terms are as previously described. An edge of unitary weight is defined between v_i and v_j if they are mutual k -NN based on the above distance measure. Worst case, this method is $O(n^2)$ due to evaluation of all pairwise distances, but the spatial diffusion kernel support can be leveraged to greatly reduce computation time.

Benedetto et al. (2012a) discuss three spatial-spectral methods that were generalized into a single equation by Cahill et al. (2014). A fusion parameter β controls the fractional weighting of measures on the spectral data ($0 \leq \beta \leq 1$); naturally $(1 - \beta)$ controls the contribution of the spatial coherency component.

$$d_\beta(v_i, v_j) = \sqrt{\beta \left(1 - \exp\left(\frac{d(v_i, v_j)^2}{2\sigma^2}\right) \right) \cdot (1 - \beta) \left(1 - \exp\left(\frac{\|v_i^p - v_j^p\|^2}{2\sigma_d^2}\right) \right)} \quad (2.9)$$

Just as in Hou et al. (2013), a mutual k -NN graph is constructed from the d_β measures, but the weighted adjacency matrix is allowed to take on real values according to the output of the heat kernel in (2.4). Substituting $d_\beta(v_i, v_j)$ into the standard heat diffusion equation and assigning $\sigma = 1$ provides the basis for three Benedetto variants (2012a) for performing dimensionality reduction prior to data classification.

2.7.3 Fusing information while remapping the data

A powerful approach detailed by Cahill et al. (2014) is to forgo fusing spatial and spectral information during graph construction and introduce them separately in the manifold embedding process itself. Cahill et al. propose creating Schroedinger Eigenmaps (Czaja and Ehler, 2012) defined based on a graph Laplacian constructed from spectral information alone, while encoding spatial information into cluster potentials for use during the manifold embedding process. This variant is called Spatial-Spectral Schroedinger Eigenmaps (SSSE). We defer discussion of Schroedinger Eigenmaps until Chapter 7, but touch on two inherent benefits of this technology here.

1. Spatial information is only encoded in the cluster potentials and is not used to modify the spectrally determined weights. As such, the size of the neighborhood can be made arbitrarily small while still accommodating edges between spectrally global neighbors. This avoids the over segmentation problem that can occur with windowing algorithms.
2. Separation of spatial and spectral information facilitates studying the impact of changing spatial-weighting characteristics. This is an inherent weakness of the previously mentioned methods that would require regeneration of the entire graph because the spatial measures are embedded in the spatial-spectral weights.

We will leverage this capability in Chapter 7 to study the impacts of changing the relative weighting of spatial and spectral contributions to manifold embedding and subsequent classification. Two new spectral similarity measures adapted from the data

mining and machine learning communities and introduced in the next chapter prior to comparing many of these methods in Chapter 4.

3

NEW METHODS OF HSI GRAPH CONSTRUCTION

Many of the methods in the previous chapter are built upon application of heat diffusion kernels on spectral measures where the region of support is controlled by scaling parameters in the denominator of the exponential. Fixed (σ) and adaptive ($c\sigma_i\sigma_j$) options are both possible. Visualizations of the reweighting process provide an intuitive basis for understanding their function. Figure 3-1a shows Euclidean distances in native column wise order as well as those grouped by class identifier to the right for the *SalinasA* scene. Notice the shortest distances generally exist within each class, where classes are outlined in white in (b). The upper left block is the *unclassified* class and can be ignored. One may notice the second block is exceptionally well separated from the others.

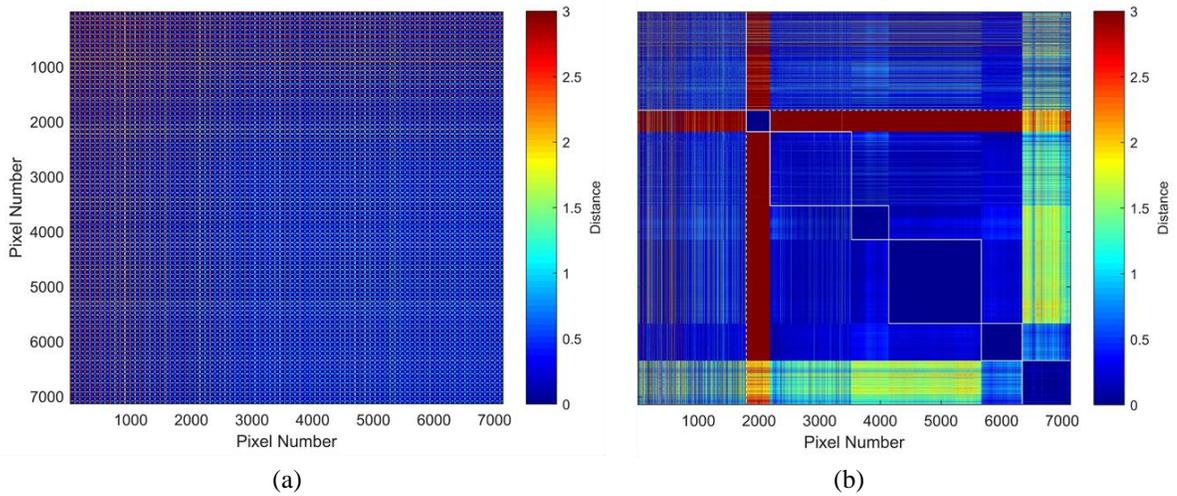


Figure 3-1. Pairwise Euclidean distance matrix for (a) *SalinasA* in native order and (b) class-sorted order. Classes are outlined in white in (b).

Application of fixed ($\sigma = 10$) and adaptive ($c = 1$) methods are shown in Figure 3-2 to the left and right respectively. Notice the strong difference between the techniques where the adaptive method provides a visually appealing block structure with fewer stronger edges in each class block. However, the fixed scaling method retains the same general blocky structure as Euclidean distance because of the monotonicity of the heat kernel, but the interclass separation has now increased.

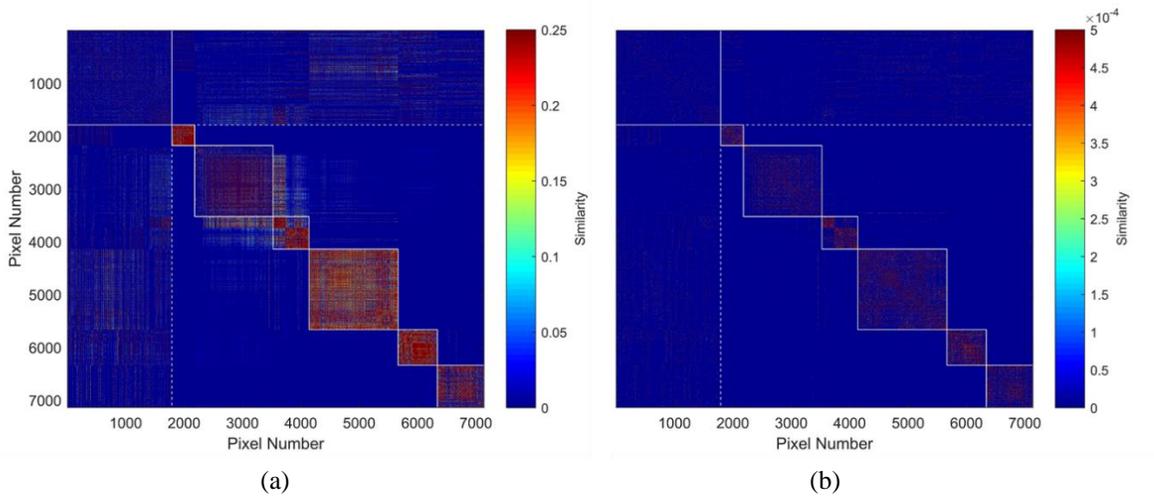


Figure 3-2. Distance reweighting. (a) fixed-scale heat kernel and (b) adaptive heat kernel. Classes are outlined in white.

Striations are apparent in the adaptively reweighted graph due to varying density in HSI clusters. Significant interclass scale variations exist indicating density is local to a pixel and not a class as a whole. Varying densities are expected, and suggests that the concept of scaling by codensity to “tighten” up clusters may not work well if clusters exhibit variation. However, it is possible, this interclass variability can be exploited by these techniques, so further testing is performed.

The Davies-Bouldin Index (DBI) and Silhouette Coefficients (SC) (Figure 3-3) were calculated for the original and reweighted matrices to determine if the clusters are better represented in the reweighted space. DBI and SC both show a reduction in separability after adaptive reweighting.

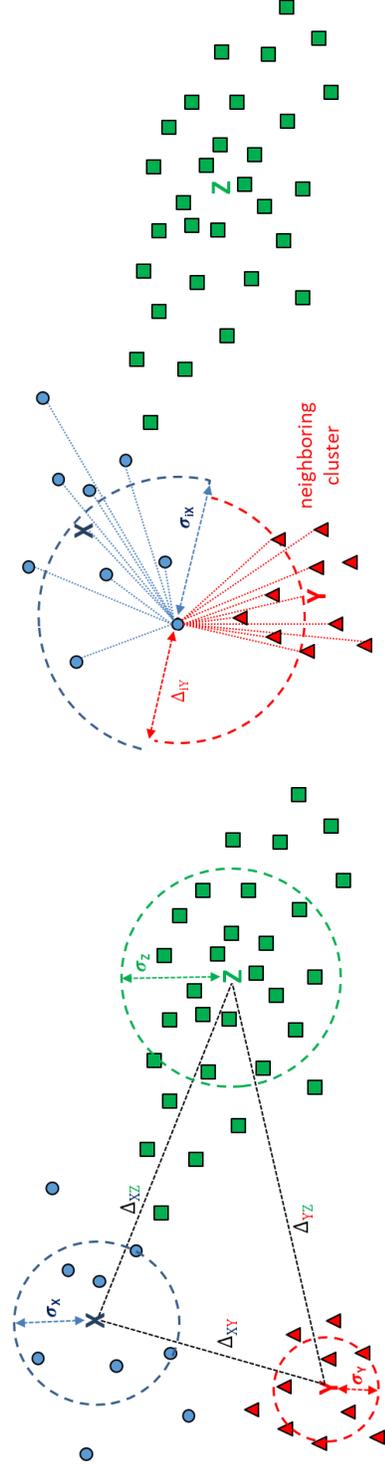


Figure 3-3. DBI and SC internal cluster-validation methods.

Davies-Bouldin Index		Silhouette Coefficient	
Cluster Dispersion	For each cluster C_i , calculate the average distance (σ_i) from the M intracluster pixels (v_j) to the cluster centroid (μ_i)	For each pixel v_i , calculate the average distance (σ_i) to the remaining $M-1$ intracluster pixels v_j	$\sigma_i = \frac{1}{M-1} \sum_{j=1}^{M-1} \ v_j - v_i\ $
Cluster Separation	For each cluster C_i , calculate the distance (Δ_{ij}) from its centroid (μ_i) to all other cluster centroids of all other clusters (μ_j)	For each pixel v_i in C_i , calculate the distance (Δ_{ij}) to all pixels v_j in the remaining $M-1$ clusters. The cluster $\{C\}$ with the lowest average distance ($\Delta_{i,C}$) is the <i>neighboring cluster</i> .	$\Delta_{ij} = \ v_j - v_i\ $ $\hat{\epsilon}(i) = \underset{j}{\operatorname{argmin}} \left(\frac{1}{ C } \sum_{j \in C} \Delta_{ij} \right)$
Base Ratio	For each cluster C_i , find the maximum ratio of the sum of pairwise cluster dispersions to their centroid separation	Ratio (scale) the difference between the <i>neighboring cluster</i> distance ($\Delta_{i,\hat{\epsilon}(i)}$) and each pixel's cluster dispersion to the maximum of the two values	$R_i = \max \left(\frac{\sigma_i + \sigma_j}{\Delta_{ij}} \right)$ $R_i = \frac{\Delta_{i,\hat{\epsilon}(i)} - \sigma_i}{\max(\Delta_{i,\hat{\epsilon}(i)}, \sigma_i)}$
Summary Dataset Metric	Average the maximum ratios from all N clusters.	Average the ratios from all N pixels in the image	$DBI = \frac{1}{N} \sum_{i=1}^N R_i$ $SC = \frac{1}{N} \sum_{i=1}^N R_i$

Many codensity-based reweighting methods such as (2.2), (2.3), (2.7) depend on the product of codensities ($\sigma_i\sigma_j$), so much can be learned about these algorithms through examination of this quantity. The ratio σ_i/σ_j should be ~ 1.0 for all intraclass pairwise pixels if they have similar densities. Any deviation from 1.0 indicates intraclass density variations exist, which is not unexpected in real data. The question to be answered is if these intracluster variation impede the utility of density-based scaling. These codensity ratios are visualized by reordering pixels by class number to create a new vector from which a $n \times n$ codensity similarity matrix S is created, where $[S]_{ij} = \sigma_i/\sigma_j$ (Figure 3-4, left). The resultant $n \times n$ visualization should appear as a blocked matrix if intraclass codensities are somewhat similar, but distinct for each class (Figure 3-4, right).

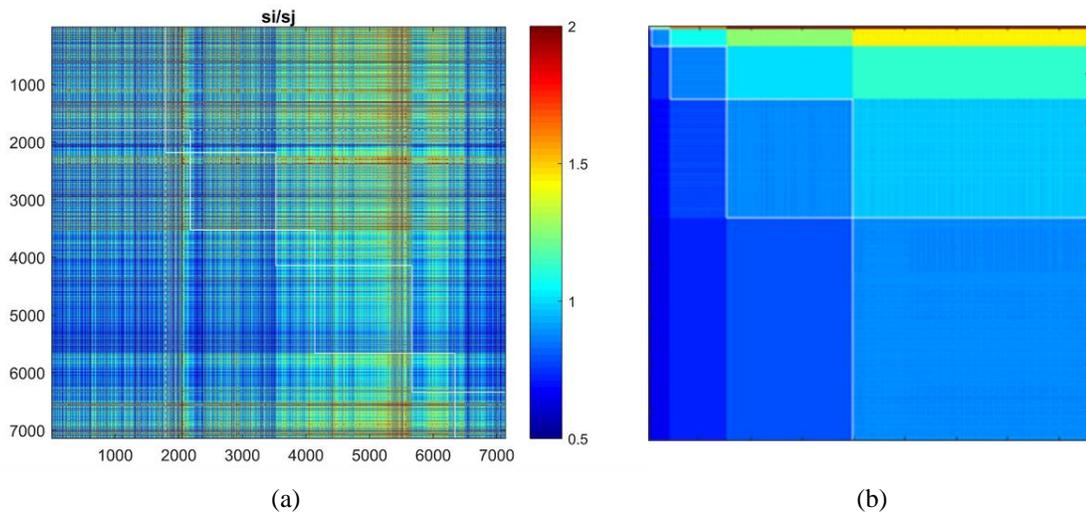


Figure 3-4. Local scale ratios (σ_i/σ_j) for (a) real data and (b) synthetic data of relatively uniform class-density scales. Class blocks are outlined in white.

Variations within the diagonal blocks to the left confirm intraclass codensity (or scale) variations are responsible for the pattern seen in Figure 3-2. Given this behavior, it is worth exploring reweighting measures less sensitive to the local scale estimation in such variable data density.

Several new spectral-density based graph construction techniques stemming from algorithms in the data-mining and machine learning literature are introduced next as potential alternatives to the codensity-based methods discussed in the previous chapter and discussed above. Of key interest are methods that do not rely on codensity because its viability as a measure of local scale is questionable, hence the impetus for examining variants of these particular techniques. Note that any of the spatial weighting techniques discussed in Chapter 2.7 may also be applied with these techniques to generate spatial-spectral variants.

3.1 Shared Nearest-Neighbor (SNN) Graphs

The SNN similarity (Jarvis and Patrick, 1973) is based on the premise that similar nodes should have overlapping neighborhoods. If this is true, the similarity of two nodes, as measured by some primary measure, can be reinforced or “confirmed” by the presence of shared neighbors. The strength of the similarity between two nodes may therefore be recast (reweighted) in terms of the number of shared neighbors, i.e., the higher the number of shared neighbors, the more similar the nodes. SNN can be very effective in the presence of clusters not well modeled by symmetric parametric distributions due to its ability to find regions of varying shape and density, and is widely used in the data mining community

(Patidar, Agrawal, and Mishra, 2012). Variants of this method are presented as features-space similarity measures based on this property as well as its codensity-independence.

Given a vertex set V consisting of $n = |V|$ vertices and a neighborhood size k , where $k \in \mathbb{N}^+$, let the k neighborhood set of vertex i be represented as $NN_k(i) \subseteq V$, and define the SNN similarity as the size of the neighborhood intersection between two nodes i and j (3.1).

$$SNN(i, j) = |NN_k(i) \cap NN_k(j)| \quad (3.1)$$

Whereas the initial implementation of SNN used fixed neighborhood sets (constant k), we recognize asymmetric neighborhoods as simply a generalization resulting from the use of adaptive density-based methods. We therefore define the more general density-based adaptive SNN measure as

$$SNN(i, j) = |NN_{k_i}(i) \cap NN_{k_j}(j)|, \quad (3.2)$$

where k_i and k_j are the size of the neighborhoods generated for vertices i and j respectively from any of the density-based techniques. The Jarvis and Patrick (1973) variant is a special case of this generalization where $k_i = k_j = k$.

An SNN cosine similarity (Houle, Kriegel, Kröger, Schubert, and Zimek, 2010) may also be defined and is given by

$$SNN_{cos}(i, j) = \frac{IV_i \cdot IV_j}{\sqrt{k_i} \sqrt{k_j}} = \frac{SNN(i, j)}{\sqrt{k_i \cdot k_j}}, \quad (3.3)$$

where IV_v is the edge indicator vector for node v and ki , kj , and $SNN(i, j)$ are as defined in (3.1) and (3.2).

From an imaging spectroscopy perspective, whereas symmetric globular clusters may exist in isolated regions of uniform material type, the presence of significant mixing can result in asymmetrically extended non-globular regions in hyperspace. Description by parametric forms is therefore difficult; hence the need to explore methods invariant to cluster shape such as SNN.

The concept of overlapping neighborhood counts was first used in HSI graph construction by Fan and Messinger (2014) to modify the strength of a Gaussian-diffusion spectral similarity measure derived from Euclidean distance. This treatment differs in that the spectral-similarity is based entirely on SNN counts and variations thereof, instead of simply being a modifier of another technique.

Creation of the SNN graph is conceptually simple, but more computationally expensive than the methods discussed thus far due to the intersection of nearest neighbor sets for counting shared neighbors. The SNN similarity is therefore a secondary measure generated from a primary distance (or similarity) metric. SNN graphs are generated from the following three steps:

1) Compute the pairwise (dis)similarity matrix

A primary distance (similarity) measure is used to define initial edges

2) Construct a mutual k-NN graph

Mutuality is imposed to avoid edges crossing regions of differing density and decrease computation time through a reduction in the number of required set intersections

3) Redefine edge weights in terms of shared neighbor counts

The number of neighbors shared by the two nodes becomes the new edge weight, replacing the (dis)similarity score from the primary metric. Note that initial edges can be severed if $SNN(i,j)=0$, changing the physical structure of the graph.

The SNN construct has many desirable qualities. First, SNN is built upon the k -NN relationship, which provides automatic density scaling because “nearest” is insensitive to local scale (Ertöz et al., 2003). Secondly, as a secondary metric built upon shared neighbors, it can overcome limitations in primary distance metrics due to contrast loss in high dimensions (Houle et al., 2010), an aspect of the curse of dimensionality (Chapter 5). Lastly, SNN has been proven to be resilient to the hub phenomenon affecting data mining in high dimensional spaces if used for distance prescaling (Flexer and Schnitzer, 2013). Note that whereas the HSI extrinsic dimension (number of bands) can be on the order of hundreds, the intrinsic (or inherent) dimensionality is often much less. Intrinsic dimensions on the order of tens of dimensions (or less) have been demonstrated (Schlamm et al., 2010; Heylen, Parente, and Scheunders, 2017). Regardless, it is prudent to study methods more resistant to challenges associated with the curse of dimensionality as intrinsic dimension will increase as higher spatial and spectral resolution system are developed.

Conversely, as a secondary metric, SNN similarity is more computationally expensive to the point of being prohibitive if performing neighborhood intersection over every node. Calculating the SNN graph from every pixel takes $O(n^2) + O(n^2 \log(n))$ time, where the quadratic is the result of the pairwise distance calculation, and the logarithmic term is from the intersection over the entire dataset. However, SNN construction can be performed in reasonable time if working from an initial mutual k -NN construct or other method to reduce the number of required intersections such as starting with a limited, but

higher value for k or spatially limiting the pixels that can be connected (Fan and Messinger, 2014).

Figure 2-3(d) demonstrates the effectiveness of counting shared neighbors on an example Delaunay triangulation similar to Figure 2-3b. Edge weights are reweighted such that nodes in regions of similar density are more strongly connected whereas those in transition (intercluster) regions are deweighted.

SNN edge weights are simply the number of nearest neighbors shared between two vertices associated by an edge. As such, this construct does not utilize any information about the node ordering within neighborhood lists. Clearly two nodes that have the same nearest neighbor are more likely to be similar than two nodes where the shared neighbor is the first in one list and the last in another. The SNN measure may be adjusted to include rank information (Jarvis and Patrick, 1973), wherein edge weights are redefined as

$$w_{ij} = \sum_{v \in \{NN(i) \cap NN(j)\}} (k - m + 1) * (k - n + 1), \quad (3.4)$$

where k denotes the size of the NN set, and m and n are the ranks (positions) of the common neighbors in $NN(i)$ and $NN(j)$ respectively. Jarvis and Patrick note that ranked-component contributions can also be added if multiplication is too severe. Each component of the summation in (3.4) has the same form, $(k - m) + 1$. The left part $(k - m)$ produces a scalar inversely proportional to ranking, i.e., lower ranks result in higher scalars, but spans $[0, k-1]$. The “+1” aligns component contributions to $[1, k]$, so that there are no zero components.

Users may threshold SNN weights as part of a clustering algorithm or leave the weights as is for analyses requiring connected graphs in subsequent processing (e.g., target detection). In the latter case, there are fewer artificial connections created during post processing to ensure connectivity, so the graph is more representative of data structure.

Figure 3-5 illustrates the utility of rank dependent weights in a 3-NN toy example and subsequently reweighting each edge with its SNN score. Notice the simple SNN count produces the same edge weight of two for w_{ij} , w_{jk} , and w_{ik} despite the fact v_j is not in the same cluster as v_i and v_k . The rank dependent SNN similarity measure shows that v_i and v_k are much more similar than v_i and v_j or v_j and v_k , in line with our expectations.

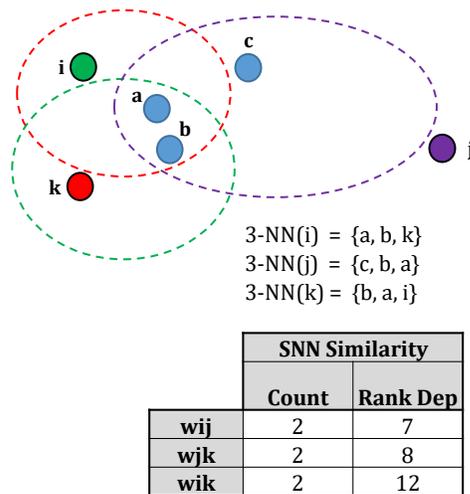


Figure 3-5. Rank-dependent SNN similarity. Dotted ellipses touching the edge of each node denote its 3-NN.

It is interesting to note that this particular situation is avoided by an initial construction enforcing mutual k -NN, i.e., edges to node v_j disappear because $v_j \notin NN(i)$ and $v_j \notin NN(k)$.

The mutuality constraint used in the construction of initial edges not only provides protection against edges spanning regions of different density, but also maintains a balance in nearest neighbor list size. Imbalanced nearest neighbor lists may cause dissimilar nodes to appear more similar than those that are in fact similar. For example, Figure 3-6 shows a case where the neighborhood size and geometric configuration is such that there is an imbalance in the intersections between pairs of nodes. In this case, v_i or v_k occupy a position in each other's nearest neighbor lists. This reduces the number of possible nearest neighbors in the intersection by one for each node. Node v_j doesn't have either v_i or v_k in its nearest neighbor list, hence has a larger number of nearest neighbors left from which to form a neighborhood intersection set with other nodes. This imbalance creates an extra term in the sum from (3.4) for v_j which may result in a higher similarity score and cause v_i and v_k to appear more similar to v_j than to each other (edge ik is weighted from two common neighbors edges, where edges ij and kj are weighted from three). Note that the magnitude of the imbalance effect decreases with increasing neighborhood size, so regions of low density are impacted the most e.g., lower density regions from adaptive nearest neighbor techniques.

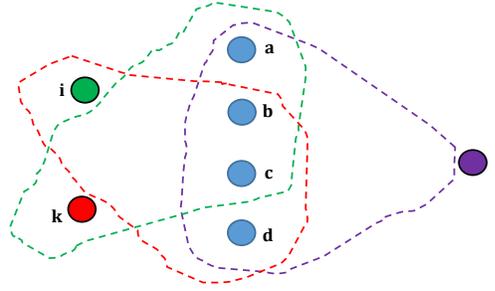


Figure 3-6. Lack of the mutual constraint can lead to an undesired similarity score.

The SNN similarity variants above do not account for the distance (or similarity) between nodes either, only their overlapping neighborhoods and ranks. A node may have the same rank with respect to two other nodes, but have vastly different distances (similarities) to those nodes. In this case it seems appropriate that this node contributes differently to the SNN score for each of the two nodes. A modified metric based Moëllic, Haugeard, and Pittel (2008) uses the shared nearest neighbor count, rank, and node similarity given by

$$w_{ij} = \sum_{v \in \{NN(i) \cap NN(j)\}} (k - m \cdot dsim_{iv} + 1) * (k - n \cdot dsim_{jv} + 1), \quad (3.5)$$

where $dsim$ is a dissimilarity measure between nodes. The more dissimilar the nodes, the larger the subtracted term, hence the smaller the component contribution. Similar nodes (low dissimilarity, small distance) result in smaller values being subtracted from k , increasing the component contribution. In this work, the cosine distance will be used as the dissimilarity metric to promote higher scores for those neighbors that may have varying illumination. The sensitivity of cosine distance to dark pixel selection is mitigated by the

fact only nearest neighbors are subject to the metric, i.e., dark pixels would be too far away to be initially selected.

The asymmetric neighborhood variants are given by simply replacing the k 's in (3.4) and (3.5) by ki and kj respectively.

3.2 Mutual Proximity Graphs

Introduced by Schnitzer et al. (2012), mutual proximity (MP) transforms distances into similarities such that pixels with similar nearest neighbors are brought closer together, whereas those with dissimilar nearest neighbors are pushed farther apart. In this sense, mutual proximity is akin to local scaling except mutual proximity is a global vice local transformation.

To calculate MP, distances between v_i and all other nodes are assumed to originate from a known probability density function (P). Using node i 's distance distribution, $P(\delta_i)$, the distance from v_i to v_j (δ_{ij}) can be interpreted as the probability v_j is a neighbor of v_i by

$$P(\delta > \delta_{ij}) = 1 - P(\delta \leq \delta_{ij}) = 1 - CDF(\delta_{ij}), \quad (3.6)$$

where CDF is the cumulative distribution function of $P(\delta_i)$. The probability a random node is a nearest neighbor of node i therefore increases with decreasing distance (or codensity). Obviously $\delta_{ij} = \delta_{ji}$ and $\delta_{ii} = 0$ produces a probability of 1.0. This concept is represented graphically in the bottom of Figure 3-7 for a normally distributed distance distribution. Marginal distance distributions are shown below and to the left of the scatterplot for nodes i and j respectively. Using the lower marginal distribution as a reference, the probability a

random node k with distance δ_{ik} is closest to node i is given by the dark shaded area to the right. Calculating the mutual proximity between nodes i and j is then reduced to a counting exercise over the region of joint support ($\delta_i > \delta_{ij}$ and $\delta_j > \delta_{ji}$), normalized by the total number of points. $P(\delta_i > \delta_{ij}) > P(\delta_i > \delta_{ik})$ therefore indicates node j is more likely a neighbor of node i than node k .

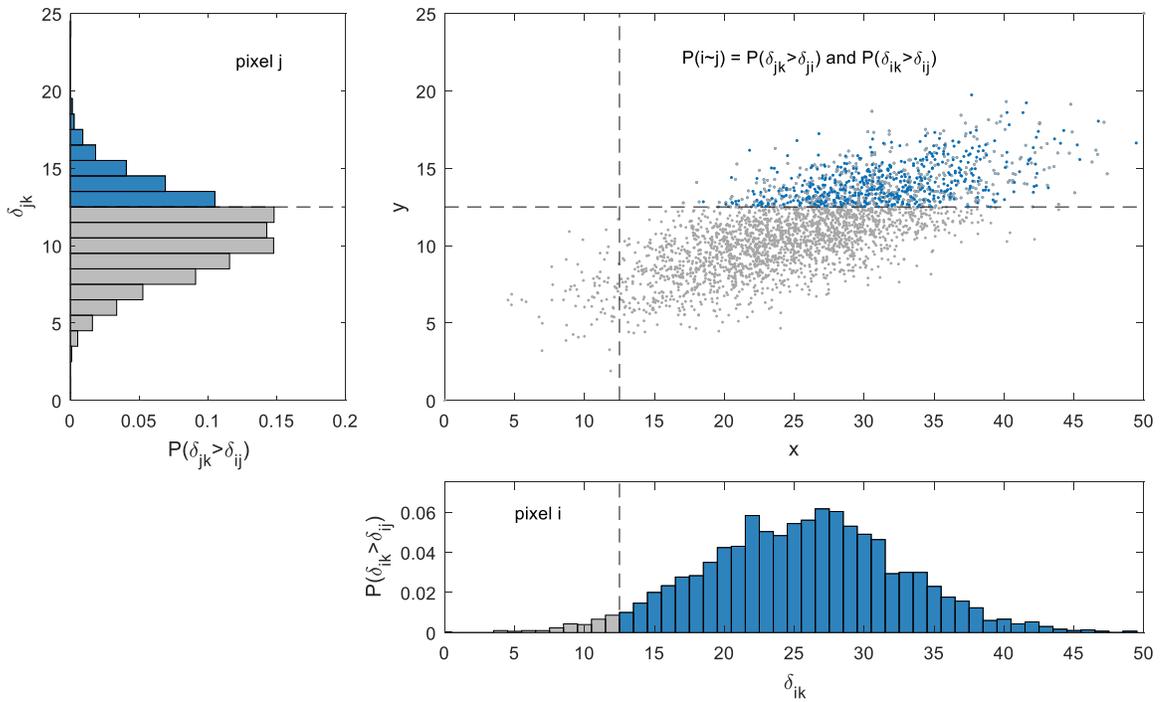


Figure 3-7. 2-D scatterplot of distances from nodes i (x axis) and j (y axis) to all other nodes and their MP.

The distance distributions for v_i and v_j are naturally different, so $P(\delta_i > \delta_{ij})$ and $P(\delta_j > \delta_{ji})$ are not the same. This is similar to the directed relationships that

emerge in k -NN graphs where v_i may be a nearest neighbor of v_j , but v_j may not be a nearest neighbor of v_i .

Calculating mutual proximity is conceptually simple; count the number of nodes having distance $> \delta_{ij}$ to both v_i and v_j and then divide by the number of nodes to normalize the probability (Schnitzer et al., 2012). This is shown graphically to the upper right in Figure 3-7 and represented by (3.7).

$$MP(\delta_{ij}) = \frac{|\{k : \delta_{ik} > \delta_{ij}\} \cap \{k : \delta_{jk} > \delta_{ji}\}|}{n} \quad (3.7)$$

The mutual proximity reweighting for the previous three-cluster example with variable densities is shown in Figure 3-8(e).

Counting exercises on large datasets can be computationally expensive, therefore we wish to find efficiencies wherever possible. For example, if we assume independent codensity distributions, then mutual proximity can be easily calculated via the product of the marginal distributions as

$$MP_I(\delta_{ij}) = P(\delta_i > \delta_{ij}) \cdot P(\delta_j > \delta_{ji}), \quad (3.8)$$

where the subscript ‘I’ indicates the independent marginal distribution assumption. Assuming independence did not adversely affect results on standard machine learning datasets in Schnitzer et al. (2012). Similar tests on HSI data are discussed in Chapter 4.

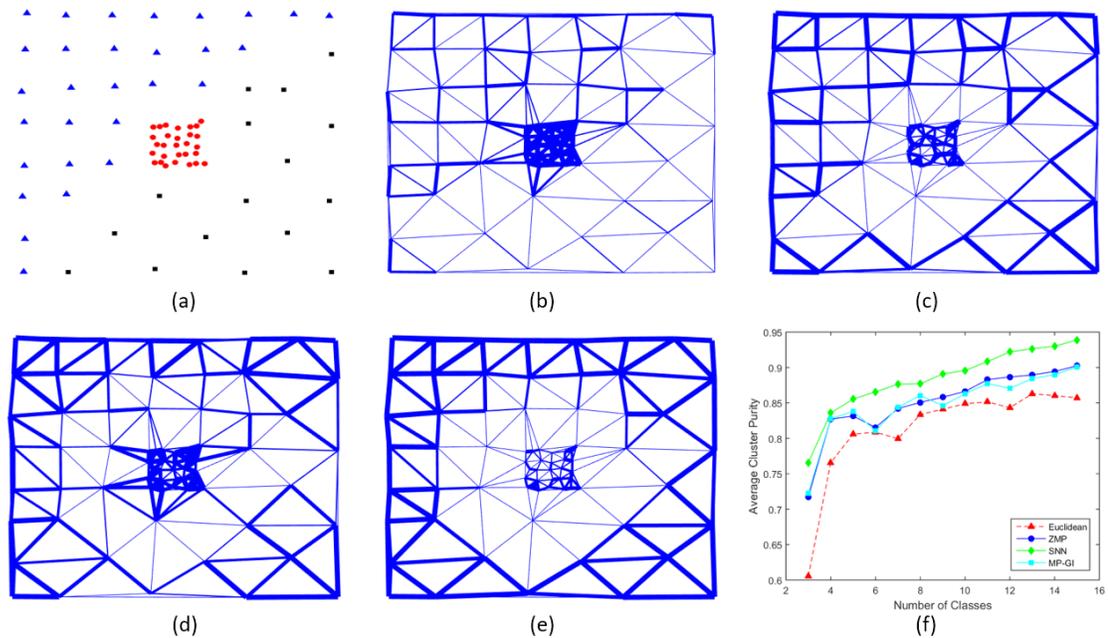


Figure 3-8. Example dataset displaying (a) three clusters of different density, (b) Euclidean distance weighting, and three edge reweighting methods: (c) ZMP, (d), SNN, and (e) MP. Performance measures (f).

Comparisons of the different edge reweighting variants is provided in Figure 3-8. We repeat the data points and initial Delaunay triangulation edges in (a) and (b) for clarity. The resultant edge weights after the application of ZMP locally adaptive scaling, shared nearest neighbors, and mutual proximity are shown in Figure 3-8(c-e). Notice the stronger relationship between pixels in the lower density clusters and the weaker edges in transition regions. Additionally, weaker edges are now apparent in the dense center cluster because weights are now relative to local density instead of a global distance scale. Each graph is partitioned into varying numbers of clusters by the normalized cuts algorithm (Shi and Malik, 2000), with the resulting cluster uniformity shown in (f). The SNN method had the highest uniformity.

Euclidean distance is approximately normal due to the central limit theorem if features (bands) are generated from independent and identically distributed data (*i.i.d.*) (Jegou et al., 2010). Whereas most data are not *i.i.d.*, Schnitzer et al. (2012) point out this approximation increases in accuracy with increasing intrinsic dimensionality. However, the empirical distribution can be used if the data are not reasonably modeled by an analytical distribution, sacrificing speed for performance.

The computational complexity of mutual proximity is $O(n^2)$ at a minimum due to the evaluation of all pairwise distances and intersections. The intersection rescaling does take additional time, but if we can assume the marginal distributions follow a functional form, we can use a pixel subset, S , to estimate the distributional parameters ($|S| \ll n$). This reduces the number of rescaling calculations to $S * n$, resulting in linear rescaling complexity (Schnitzer *et al.*, 2012). Additionally, one may use a fast NN technique, pulling far more neighbors than typically needed, but still orders of magnitude smaller than the $|V|$. In this case, mutual proximity is calculated on the larger NN sets, with the larger part of the distribution simply being added as an offset because its sorted order doesn't matter.

We previously noted some of these techniques are computationally expensive without some means to reduce the number of pixels examined. Practically, running larger images on a single machine may become prohibitive. However, the recent availability of massively parallel commodity computing enables scalable processing on demand, reducing execution time to relevant levels (if you're willing to pay for it). A more pragmatic approach is the use of fast or approximate nearest-neighbor algorithms which can greatly decrease execution time with little reduction in performance (Benedetto et al. 2012a,

2012b). Quality degradation can be lessened by returning a larger k -NN list than necessary with fast or approximate k -NN, then whittling down those results. The use of partial k -NN lists does not preclude use of SNN because shared neighbors are likely within a fraction of nearest neighbors anyways. Additionally, errors in low mutual proximity values will not impact results as we're only practically interested in the higher MP values, so a large fraction of pixel distances don't need to be sorted and can just be lumped into a single term. For example, we return 20% of nearest neighbors and determine their sorted order for the empirical mutual proximity measure. The remaining 80% can simply be added as a bias without sorting all the distance because they don't matter.

We now have enough basic information to understand the rationale behind the five key performance issues studied in the next chapter.

4

GRAPH CONSTRUCTION PERFORMANCE

In the beginning of this thesis, we stated the performance of graph-based algorithms is dependent on the quality of the constructed graph. It is therefore good practice to study the resultant graph characteristics and relate them to common HSI processing tasks to ensure appropriate construction methods are selected for the desired analyses. For example, do mutual, superset symmetric, or directed graphs offer better performance? Does the answer change based on scene content? Should adaptive variants be utilized instead of fixed k methods? To begin to answer these questions, the performance of the previously described graph construction techniques is examined to determine which methods best preserve community structure of the data by answering five questions most relevant to data clustering, segmentation, and classification.

- Which method is best for symmetrifying adjacency (or affinity) matrices: mutuality or superset symmetry?
- Are spectral-density based adaptive construction techniques better than fixed neighborhood sizes for the same number of edges?
- How do symmetrified matrices perform with respect to their directed counterparts?
- Does edge reweighting after edge selection improve the community structure of neighborhood lists?

- Does prescaling primary metrics prior to k -NN graph construction improve performance?

Algorithms described for instructional purposes were not evaluated. Note that the intent of the following sections is to study the resultant graph structure (characteristics) from these construction techniques, not to offer new classification algorithms. In doing so, we will see that some graph construction methods are better suited for classification-like tasks. We defer the analysis of geodesic distances and manifolds, also commonly used for anomaly and target detection, to a future publication.

4.1 Metrics

A simple majority-rules k -NN classification was used to test the performance of each graph construction technique. Summary accuracy metrics representing the percentage of correct class assignments are produced for each combination of image and graph type; ties are broken by assignment to the class with the shortest distance to the test pixel. Note that we use the k -NN classifier as a metric for studying the health (or uniformity) of k -NN lists, not to achieve the best absolute classification accuracy. More robust graph-based classification techniques exist that are better suited for such a comparison.

Whereas the k -NN metric establishes summary classification performance for each graph construction technique, it shows limited difference between algorithms starting from the same set of initial edges, i.e., many of the adaptive techniques previously discussed start with an ANN graph. As such, their k -NN lists are identical so their majority-rules scores are the same (tie results may differ based on the metric). Essentially, the only difference between these graphs are the edge weights assigned after initial edge selection,

so many of the techniques previously discussed are referred to as edge reweighting schemes and produce “secondary” weights. To circumvent this problem, an additional test was applied to examine the structure of the k -NN list themselves. The simplest form of this test assigns a score to each nearest neighbor j based on its rank in pixel i 's k -NN list as $k(i) - rank(j) + 1, i \neq j$. Each pixel's rank-based neighborhood scores are summed and normalized to produce values in the interval $(0, k(i)]$. Scaling to $k(i)$ ensures that pixels with high similarity to many neighbors receive more weight than pixels with high similarity to a small number of neighbors.

Edge reweighting methods such as ZMP, MP, and SNN theoretically reshuffle the order of k -NN lists such that spectrally similar pixels more aligned with the density of the test pixel move closer whereas those in regions of differing density move farther away. As such, the above scoring method characterizes reshuffling by increasing the overall score when pixels of the same class as the test pixel move to lower ranks (closer to the test pixel) and dissimilar pixels move to higher ranks (farther away). While intuitively appealing, the scoring method does suffer from a shortcoming related to use of integer ranks. Two pixels that change ranks due to extremely small differences in their secondary edge weights will be assigned disproportionate changes in score due to integer ranking. Additionally, the method is insensitive to the case when pixels change weights dramatically, but retain their current ranking.

To combat both disproportionate changes in score and lack of changing ranks, an additional step is taken during the scoring process. The first and last scores (ranks) are locked at the number of neighbors for the test pixel $k(i)$, and one respectively. All pixels

that fall in between are interpolated to their floating-point positions in rank space spanned by the first and last pixels. This interpolated rank is then used as the pixel score. In this way, pixels that change rank due to small differences in secondary weights are assigned scores that are virtually identical. The adjusted rank metric also handles the case when pixels retain their original rankings but shift relative to the test pixel, changing their resultant scores even though the ranks are unchanged. The improved (adjusted rank) metric is illustrated in Figure 4-1 for a pixel with 19 neighbors. Notice that integral rank-based adjustments artificially increase node separation between the first two pixels, whereas fractional positions achieve weightings in line with node separation indicated by the distance metric, e.g., separation in rank space (left arrows) always equals one, while the adjusted rank score (right arrows) is proportional to pixel similarity.

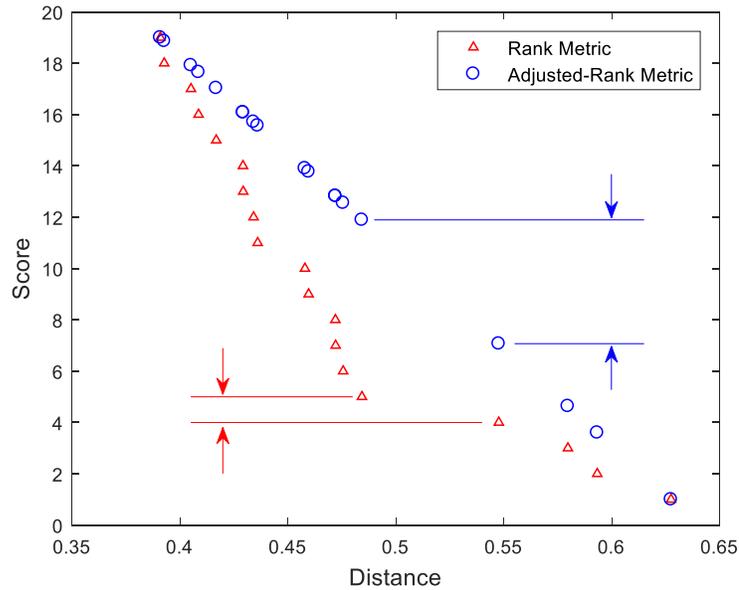


Figure 4-1. Rank and adjusted-rank scoring for evaluating the structure of k -NN lists.

As can be seen, the adjusted-rank scores of the two leftmost pixels (circles) are virtually the same because their distances from the pixel of interest are nearly identical. This is not true of the rank-only scoring that displays integer differentials (triangles). Additionally, there appears to be a separation between nearest neighbors into at least two clusters. Those in the cluster closest to the test pixel are assigned much larger relative scores than those pixels appearing to the right of the apparent cluster division. Bars to the left and right indicate the magnitude of the difference between two pixels for the rank-only and adjusted rank measures respectively. Note that this metric is appropriate for NN lists of the same size, so will only be used for the edge reweighting tests.

A simpler neighborhood health metric that overcomes the requirement for the same sized NN lists is the ϕ -edge ratio, defined as the number of edges between vertices with

differing class labels (ground truth class assignments) normalized by the number of edges in the graph (Ozaki, Shimbo, Komachi, and Matsumoto, 2011). Lower scores are thus indicative of more uniform NN lists. This metric is good for measuring changes in global k -NN health from prescaling tests as it quantifies the change in number of edges to similar pixels while avoiding the issues associated with weighted ranks when comparing NN lists of differing sizes for the same pixel.

Additionally, one may simply count the number of pixels with improved k -NN health scores from any metric. While not indicative of the degree to which any edge weight changed, this metric gives a good indication of the number of pixels impacted by reweighting or rescaling schemes and can be useful for visualizing trends due to its quantized nature (improved, no change, degraded).

4.2 Experimental Datasets

Several datasets with substantial ground truth coverage acquired from the Purdue Multispec website (Baumgardner, Biehl, and Landgrebe, 1992), the Telecommunications and Remote Sensing Laboratory, Pavia University via the Grupo de Inteligencia Computacional (GIC) University of País Vasco, Spain website, and the National Institute of Standards and Technology (NIST)-MITRE Corporation partnership were used for the performance evaluation (Table 4-1). Atmospheric absorption bands were removed from all datasets prior to evaluation, except for the *Microscene* which was acquired in a lab.

Table 4-1. Classification Accuracy Datasets

Dataset	Sensor	Data Type	GSD [m]	Bands	Scene Type	Source
<i>SalinasA</i>	AVIRIS	<i>rad.</i>	3.7	204	rural	GIC
<i>Salinas</i>	AVIRIS	<i>rad.</i>	3.7	204	rural	GIC
<i>Indian Pines</i>	AVIRIS	<i>rad.</i>	20	193	rural	Purdue
<i>Pavia</i>	ROSIS	<i>ref.</i>	1.3	102	urban	GIC
<i>Pavia Univ.</i>	ROSIS	<i>ref.</i>	1.3	102	urban	GIC
<i>Microscene</i>	SOC710	<i>ref.</i>	1.3E-4	80	~rural	MITRE

The reference section provides websites for downloading test data. MITRE-provided data is from a partnership with NIST.

Shorthand notation and acronyms: reflectance (*ref.*), radiance (*rad.*), ground sample distance (GSD), Reflective Optics System Imaging Spectrometer (ROSIS), Airborne Visible/Infrared Imaging Spectrometer (AVIRIS), Surface Optics Corporation (SOC). Sensor descriptions may be found in the references.

Some images were cropped to regions of dense ground truth coverage to avoid unneeded calculation, or reduced in scale by nearest neighbor resampling to increase spatial diversity. A true color composite of an example data set and associated ground truth map are shown in Figure 4-2a and b respectively. Appendix A contains image examples of all ground truth data.

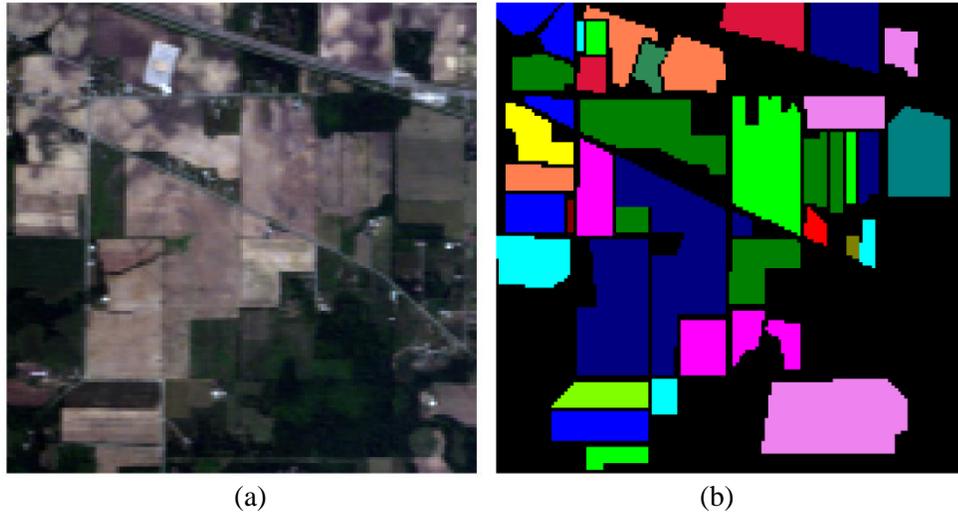


Figure 4-2. *Indian Pines* dataset for assessment of NN health. (a) true color image and (b) ground truth map (Baumgardner, et al., 1992).

Unclassified ground truth pixels were used in graph construction, but removed from the classification metrics as they could also belong to a ground truth class, but simply were not labeled.

4.3 Results

We present several studies addressing the questions posed at the start of Chapter 4 to demonstrate graph characteristics and determine construction practices required to promote the health of nearest neighbor lists for applications utilizing graph-based analysis. The intent here is to show how selection of certain graph construction methods/criteria compare to each other, not to maximize the value of any metric. We start with a comparison of overall classification accuracy to provide some high level perspective prior to specifically addressing the five questions posed at the start of this section.

For this perspective test, 12 graphs—representing combinations of three adjacency matrix types (superset symmetry, directed, and mutual) and four edge allocation methods

(k -NN, DW k -NN, DW k -NN with CDF, and ANN)—are constructed for each of the six Table 4-1 datasets using $k = k_{max} = \max(ki_{ANN})$. A k -NN classifier is used to generate overall accuracy measures, which are subsequently sorted in ascending order. Each method is then assigned a score equivalent to its rank in the sorted list, i.e., the highest overall accuracy receives 12 points, while the lowest receives one point. This process is repeated for all six images and the resultant scores are summed as a representation of general graph performance. Figure 4-3 displays the scoring results and peak-normalized marginal distributions associated with the adjacency matrix type and the edge allocation method.

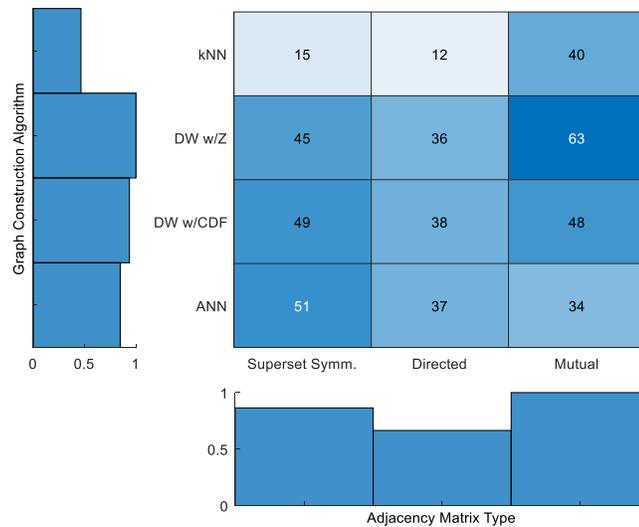


Figure 4-3. Summary classification accuracy for all scenes using three adjacency-matrix types and four edge-allocation methods.

The superset symmetric and directed adjacency columns clearly indicate the benefit of density-weighting as evidenced by the large step between k -NN and the density weighted variants. However, the same pattern is not apparent in the mutual columns, where the

mutual k -NN graph has a score on the order of the adaptive variants. This was traced back to a combination of ground truth sparsity for the urban scenes (especially the *Pavia* image) and pixel labeling. In the urban scenes, pixels of the same class exist that were not included in the ground truth pixel labeling, so they can reduce the number of edges to labeled pixels of the same class. Additionally, materials of interest are typically on smaller objects in clutter scenes, hence there is a higher probability of edges being labeled during ground truth generation. These edge pixels are composed of multiple materials, albeit largely the material of interest, and can switch class membership with varying k . This was explored by increasing k from 1 to 250 and tracking pixels that alternate from correct to incorrect, and vice versa, e.g., edges pixels in the *Pavia* scene comprise only ~25% of the ground truth data, but account for ~60% of pixels that continuously alternate class membership with k .

4.3.1 Adjacency symmetrification method

Two methods are described in Chapter 1.3 for converting asymmetric (directed) adjacency matrices into symmetric (bidirectional, undirected) variants: superset symmetry and mutuality. The superset symmetry method forces all directed edges to be bidirectional, while the mutuality constraint only retains existing bidirectional edges. Are either of these methods better than the other? Under what circumstances? To address these questions, the classification accuracy of both adjacency symmetrification methods are compared with their directed counterpart (the initial directed graph) as a relative measure of nearest neighbor uniformity. Full confusion matrix analysis is used to assess statistical significance of the results.

Figure 4-4 displays differential accuracies for combinations of graph type and image, where the differential is constructed such that positive values indicate the symmetrified graph (via mutuality or superset symmetry) outperforms the directed k -NN counterpart. This is different than the previous test in that it compares symmetrified adjacency variants directly to their directed parent instead of examining performance of all methods to each other. The first row of Figure 4-4 contains results for a small k and k_{max} values calculated as the average $k(i)$ ⁶ from the ANN algorithm. The second row is an analogous test, but with k set to the maximum ANN $k(i)$ value. Both are provided to demonstrate performance over a broad k range⁷. Note that ANN is the same in both rows as there is no notion of a k parameter. The k values for each scene are provided below the lower left of each graphic row. Scene ordering is shown in the lower left panel of the upper graph, and consistent for all tests. Fixed k -NN graphs nearly always benefit from invoking the mutuality criterion, while ANN shows a very small difference between all three adjacency matrices. Regions of overlap between symmetric and mutual variants are represented by an intermediate shading. Differential magnitudes are small, but noticeable.

⁶ This is also the number of iterations required for ANN to terminate.

⁷ Results were trended over tens of k values to establish these relationships. We only provide two here as an example demonstrating observed trends.

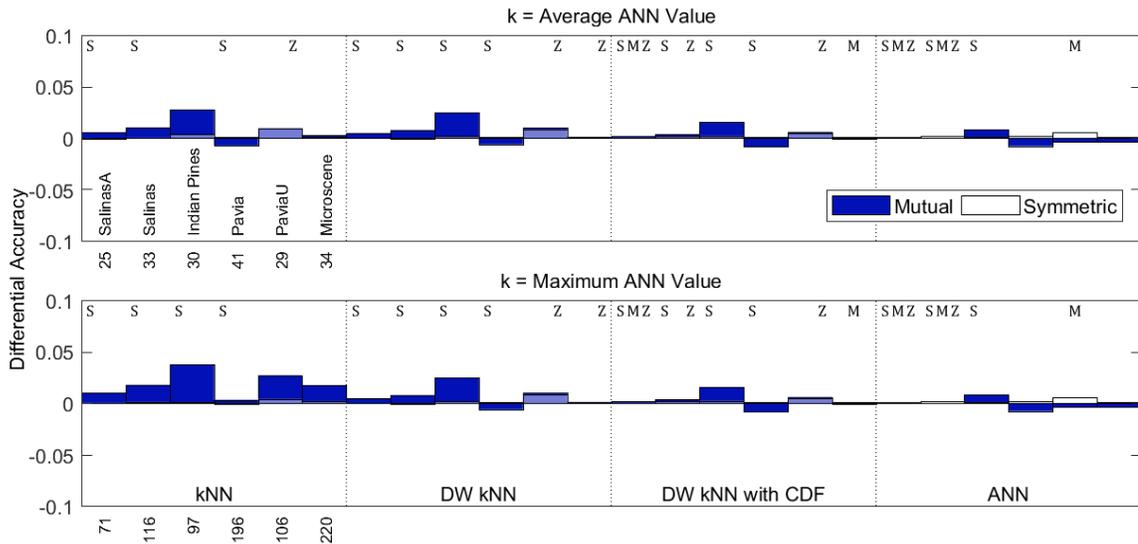


Figure 4-4. Differential accuracy results for fixed and adaptive graphs at low (top) and high (bottom) k values.

McNemar’s paired difference test was used to compare confusion matrices from each method and establish statistical significance with a critical value of $\alpha = 0.05$. Any test statistically the same as the directed k -NN result is indicated with an ‘S’ (superset symmetry) or ‘M’ (mutual) above the respective test. A ‘Z’ above the bar indicates the symmetric and mutual results are statistically the same. Apparent trends and their significance are discussed in the bullets to follow.

- Construction techniques enforcing adjacency symmetry by mutuality instead of superset symmetry are superior for capturing community structure in HSI data. This result is intuitive based on previous discussions, where the mutuality constraint provides protection from edges crossing regions of disparate densities, therefore promoting cluster uniformity.

- Superset symmetric results are statistically the same as the directed k -NN results for a large number of tests, suggesting adding additional edges in this manner does not produce much value.
- Mutuality is always the poorest performer for the *Pavia* scene due to the previously described edge pixel condition.
- Density-weighted construction methods generally have lower differential accuracy scores than the fixed k method, indicating that spectral-density weighting does a better job connecting intracluster pixels so there is less to be gained (or lost) from adjacency symmetrification, i.e., they natively produce higher edge allocation in denser regions more likely consisting of the same material.
- The general pattern of reduced differential accuracy scores moving left to right in Figure 4-4 is consistent across a broad range of k values where ANN typically has the smallest differential accuracy, followed by the DW k -NN techniques, and then fixed k -NN indicating fixed k nearest neighbor lists have more intercluster edges than their adaptive counterparts.
- The k -NN graph always displays the largest benefit from application of the mutuality constraint because there are an adequate number of neighbors for mutuality to reduce intercluster edges, i.e., all pixels are assigned k neighbors, and fewer pixels are subsequently disconnected from the graph. Isolated-pixel creation is problematic for mutual variants when k_i is low.

- Symmetric and mutual variants are typically statistically different, except in the case of ANN and density-weighted k -NN with CDF, wherein some rural (less cluttered) scenes begin to show no difference.

Examining the corresponding ϕ -edge ratios for the maximum k plots (Figure 4-5), we see that they look very similar, but the percentage of ϕ -edges is larger than the commensurate differential accuracy. This infers that differential accuracy is a function of ϕ -edge ratio, but that the relationship is not one-to-one. This is anticipated given adding some incorrect edges to a pixel with a large number of intracluster edges will likely not impact the overall classification accuracy. Similarly, adding correct edges to a correctly classified pixel does not improve the accuracy score.

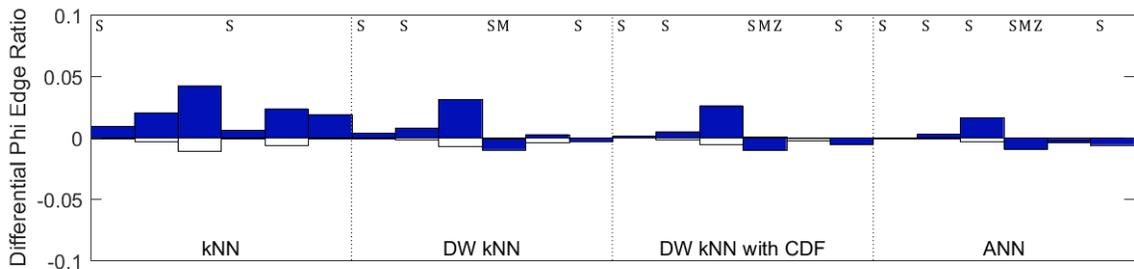


Figure 4-5. Differential ϕ -edge ratio plots for the maximum k row in Figure 4-4.

Grouping differential accuracies by scene type (rural vs. urban), we find the mutual method outperforms the directed and symmetric variants in scenes with large uniform regions by $\sim 1\%$ on average. Should an investigator choose to invoke mutuality in a cluttered urban scene, these data indicate doing so from a fixed, rather than adaptive, k -NN

graph can be beneficial, albeit more urban data with ground truth should be investigated before this can be taken as a generalization, as this may simply be an artifact of object size relative to the GSD. At a minimum, ensuring $k_{min} \in [5,10]$ will help mitigate this issue.

4.3.2 Adaptive vs. fixed nearest-neighbor lists

The previous section quantified the performance difference attained by invoking an adjacency matrix symmetrification method. However, those tests did not ensure the same number of edges in each graph construction. This section investigates the ability of adaptive and fixed k methods to encode community structure, while ensuring the number of edges are approximately the same, i.e., each adjacency matrix has essentially the same sparsity. We utilize the directed k -NN graph because the number of resultant edges is not dependent on the effects of a symmetrification method. Note that aligning the number of edges for the density weighted k -NN techniques involves distributing the total number of edges from ANN to each pixel based on the image's codensity distribution and is not as simple as assigning the average $k(i)$ value to k_{max} for those algorithms.

Investigation of the ϕ -edge ratio with the aforementioned $k(i)$ assignments is used to demonstrate differences in the number of intercluster edges resulting from each construction method. Figure 4-6 displays differential ϕ -edge ratios for each method across the same six datasets, where the differential is taken such that positive values indicate the adaptive technique creates fewer intercluster (more intracluster) edges than the fixed k technique enabling better clustering and higher classification accuracy.

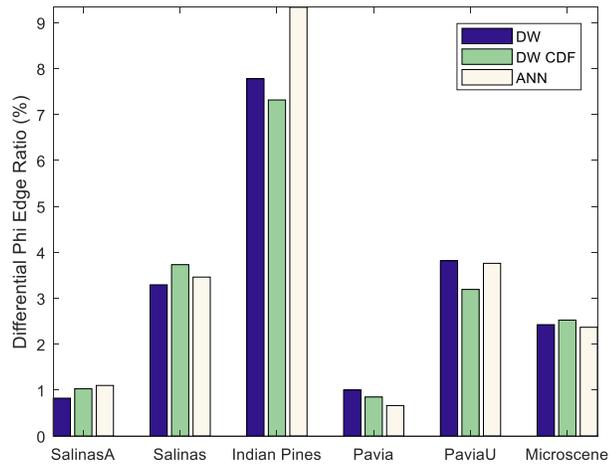


Figure 4-6. Differential ϕ -edge ratios between fixed and adaptive graph variants by scene.

Figure 4-6 demonstrates that adaptive techniques have lower ϕ -edge ratios than fixed k techniques. This was expected since these algorithms allocate more edges in dense (similar) regions, hence increasing the probability of connecting intracluster nodes. ϕ -edge ratio tests are pixel-specific, therefore produce distributions of values that need to be compared. We utilize Welch's Two Sample T-test to compare the means of each distribution. This test is fairly robust to deviations from normality for large sample sizes (true here), hence is applicable to our skewed distributions. Additionally, we compare the ϕ -edge ratio distributional forms with the Kolmogorov-Smirnov Test (K-S Test), because the test does not assume normality. The resultant p -values from both tests are combined with Fisher's Method to produce a single p -value used for significance testing with $\alpha = 0.05$. All tests rejected the null hypothesis that the fixed and adaptive k distributions were same (all results significant), hence no annotations are provided on the graphic.

An interesting point to note is that ϕ -edge ratios upwards of ~9% are observed in Figure 4-6, but no such confusion difference was observed in previous tests. This is due to the fact that adding intraclass neighbors to pixels that would have already been classified correctly has no effect on the classification result. However, it does decrease the ϕ -edge ratio because the neighborhood list becomes more uniform, i.e., the mapping from ϕ -edge ratio to classification accuracy is not one-to-one; more on this in the next section.

4.3.3 Construction performance with varying k

To support conclusions from analysis of the two k values, ki_{avg} and ki_{max} in Chapters 4.3.1 and 4.3.2 and provide a synoptic perspective across a broad k range, parametric studies on the four graph types in Figure 4-3 over $k \in [1,250]$ were performed. Several quantities were tracked over the entire range for the three adjacency matrix types (superset symmetric, directed, and mutual) via a series of three *dashboard* plots of six graphs each and displayed in Figure 4-7 through Figure 4-12. Each of these dashboards start with an introduction to the purpose of each graph, followed by the results for each respective experiment. Note that we only provide fixed k -NN and ANN graph construction results as the other density-weighted techniques are similar to ANN, but with varying magnitudes. We make additional notations about these other density-weighted graphs with each discussion if an interesting result was observed.

The x axis of many plots to come spans $k \in [1,250]$ for k -NN graph construction. Note that for ANN graphs, the neighborhood size of each pixel varies, so edges are only created up to the size of the adaptive neighborhood, e.g., if $ki = 20$ and $kj = 25$, edges from pixel i will not be scored when $k = 21$; however, edges will continue to be created

and scored for pixel j as long as $k \leq 25$. Pixel j ceases contributions after $k = 25$, and so on. This is very different than the fixed k -NN graph, where each pixel is assigned a neighborhood of size $k = 250$ on the last iteration. We note that when ki_{max} has been reached, so the ANN plots level off because edges are no longer being allocated. We do this to facilitate comparison over the full range of k values from the fixed k -NN tests.

On the first dashboard, we plot the *overall accuracy*, *average accuracy*, and *intracluster edges* for the k -NN graph in the top four plots (Figure 4-7). These three quantities use the y axis to the left. Additionally, the fractional number of pixels that are assigned to the *unlabeled* class by the k -NN algorithm are identified on the same plots (*pixels assigned to zero*) and read using the y axis on the right. *Pixels assigned to zero* are plotted to demonstrate method sensitivity to incomplete ground truth. The upper left and right plots show these quantities for the superset symmetric and directed k -NN graphs respectively, where the legend for all graphs is shown only once in the upper right plot. Mutual k -NN performance is shown to the center left, while a zoomed in cross graph summary graph is provided to the center right. All graphs show the average and maximum ki_{ANN} values as dotted vertical lines. Some observations are readily apparent.

- Overall and average accuracy generally decrease as k increases as seen in the superset symmetric and directed k -NN plots. These results are indicative of the fact that pixels are less apt to be from the same class as distance increases. Scene segmentation using an agglomerative strategy might be best served by seeding sub clusters with a small k value based on these results.

- The rate of performance degradation with k is slowest in the mutual k -NN graph indicating the benefit of the mutuality constraint.
- At low k , the mutual k -NN graph performs poorly, but has a sharp rise in performance until it rolls over and begins the general downward trend at a slower rate than the superset symmetric or directed variants. This is an interesting result given that the percentage of intraclass pixels is highest at those values of k , but is explainable given the previous discussions on the mutuality criterion. Imagine the case where two pixels in a dense cluster center are each assigned one neighbor. The chance these two pixels are mutual neighbors is small even though these pixels are in the same dense cluster center. As the nearest neighbor list grows, these pixels have an ever increasing chance of being neighbors, hence surviving invocation of the mutuality criterion. Given the dense nature of data near cluster centers, it does not take long before pixels have at least one neighbor in common. As such, the superset symmetric and directed graphs have best k -NN classification accuracy at very low k (< 7).
- Mutual k -NN graphs have the best performance over all methods as shown in the center right plot. The numbers on the plot at ki_{avg} and ki_{max} indicate mutual k -NN outperforms superset symmetric graphs by 4.8% and 6.1% respectively. Directed k -NN graphs falls in between this extreme. The mutual k -NN graph has the highest performance at high k because of its ability to throw away directional edges that are not confirmed by

bidirectional relationships. The superset criterion forces every edge to be bidirectional, even those that cross class boundaries due to natural spectral variability, overlapping clusters, or mixed pixels that are identified with a single class while possessing some smaller percentage of a second material.

Summary statistics on the number of edges from each pixel (as noted in the legend), with minimum and maximum ki values are shown as solid and dotted line types, respectively. Note that some of the curves between the adjacency matrix types overlap, making it difficult to comprehend the minimum to maximum relationship. A dotted vertical line is drawn from the minimum to maximum curves at $k = 175, 200, \& 225$ for the superset symmetric, directed, and mutual variants respectively to mitigate this difficulty without creating additional plots. Placement at different k values avoids possible obscuration conditions. Some noted observations:

- Adjacency matrices created by superset symmetry have the most edges, follow by directed adjacency matrices. This is to be expected from (1.3).
- Adjacency matrices created by invoking mutuality have a minimum node degree of zero at high k values, indicating there are some pixels (anomalies) far away from all other nodes.
- The maximum ki observed in the mutual variant has a maximum ki slightly less than k . This is to be expected from (1.4) and given the probability of any two nodes having the exact set of mutual neighbors is slim.

- The minimum and maximum ki curves are the same for the directed k -NN, so there is no apparent dotted vertical line because they are stacked on top of each other.

As for the number of edges that cross from labeled to unlabeled vertices, we see that this number just continues to grow slightly as k increases.

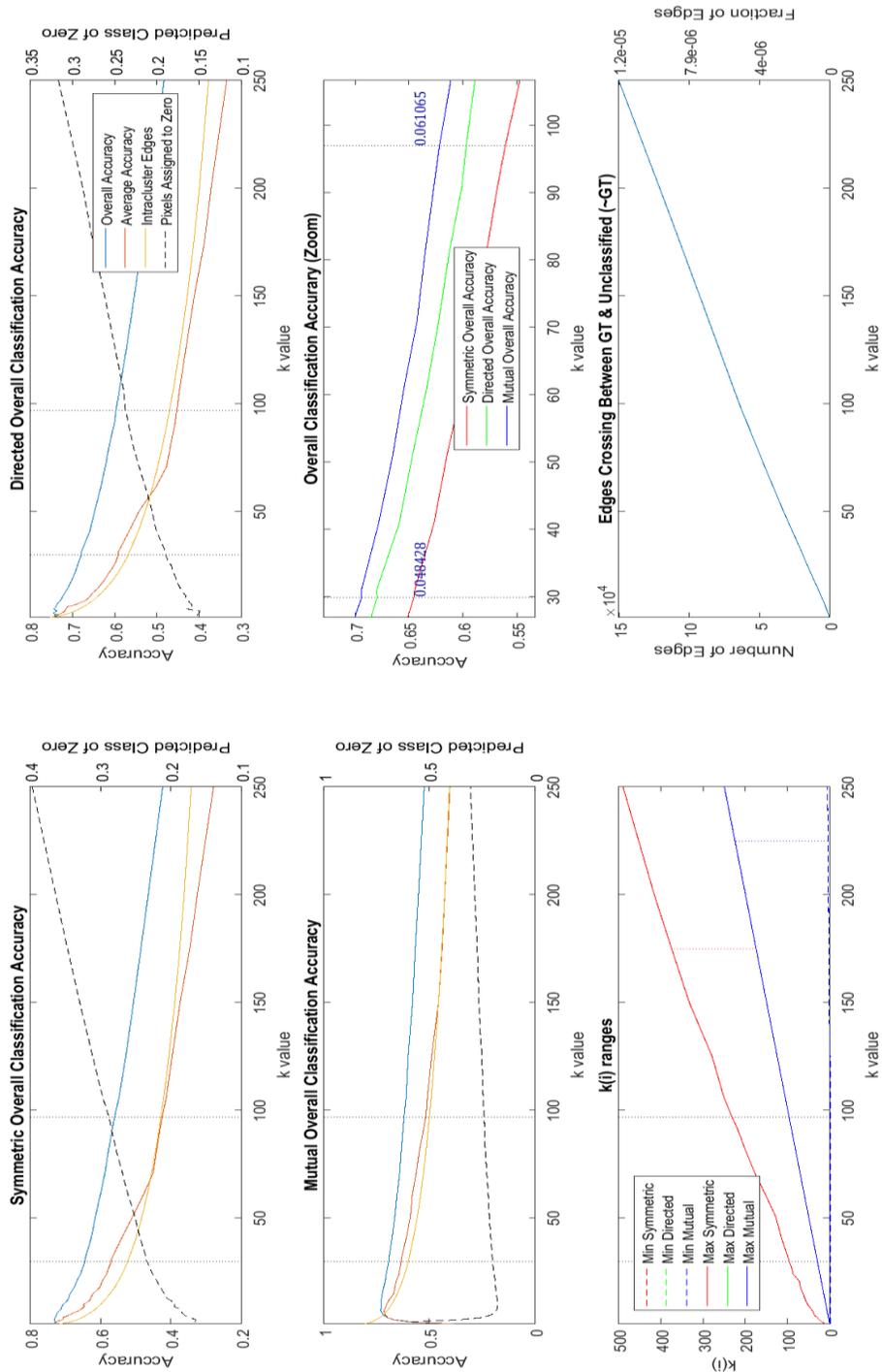


Figure 4-7. k -NN Dashboard #1: Accuracies, edges counts, and cross ground-truth edges for *Indian Pines*.

Similar observations for the ANN edge allocation technique are provided below.

- ANN absolute and average accuracies more rapidly declines than their fixed k counterparts, but quickly level off at higher accuracy values. The rapid decline is due to including NN as a function of k , instead of the order in which the edges were created, i.e., it may take some time for a node to be declared a NN by another node, but we check the $k = 1$ edge at the first iteration in this analysis. Checking the plots as a function of edge creation instead of simply k would change the rate at which both accuracies decline, but establish the same higher end result.
- All ANN graphs have a slope of zero after ki_{max} . Given there are no more new edges to be included in the analysis. The graphs are extended to the largest ki_{max} observed in all imagery to maintain plot consistency, so this is simply an artifact of the chosen range definition.
- The rate at which ANN curves approach the steady state after ki_{max} is slower than that of the fixed k graphs because not all pixels have ki_{max} neighbors. In fact, many have $k < ki_{max}$, so they stop contributing after $k > ki$. This has the impact of dropping many pixels from the analysis, only leaving those in dense regions (high ki values), where adding an additional edge has little impact on classification performance (high chance of already being correctly classified).
- Mutual graphs are still the best performing over the range $[ki_{min}, ki_{max}]$, but by a much smaller margin (a factor of 2 to 3 times less). The directed variant is closer to the mutual variant in this case. However, in other density-weighted graphs, the

directed variant has appeared in various locations between the superset symmetric and mutual ranges; albeit the relative ordering always remains the same.

- The ki plots display a very different story compared to the somewhat linear appearance in fixed k -NN graphs. First, all adjacency methods have vertices that are disconnected, i.e., $ki = 0$; this can be seen by all vertical lines extending from the maximum to zero. Some pixels always have a new edge added, hence the maximum ki is linear with k below the maximum ki value.
- The edges crossing labeled to unlabeled data display a maximum because initially, when many pixels are examined ($k < ki$), edges may still appear between some labeled and unlabeled pixels. Eventually, only pixels in denser regions (higher ki) remain. As such, only edges more likely to connect to pixels of the same class are added; this reduces rate of edge additions to unlabeled pixels and creates a reversal in the trend.

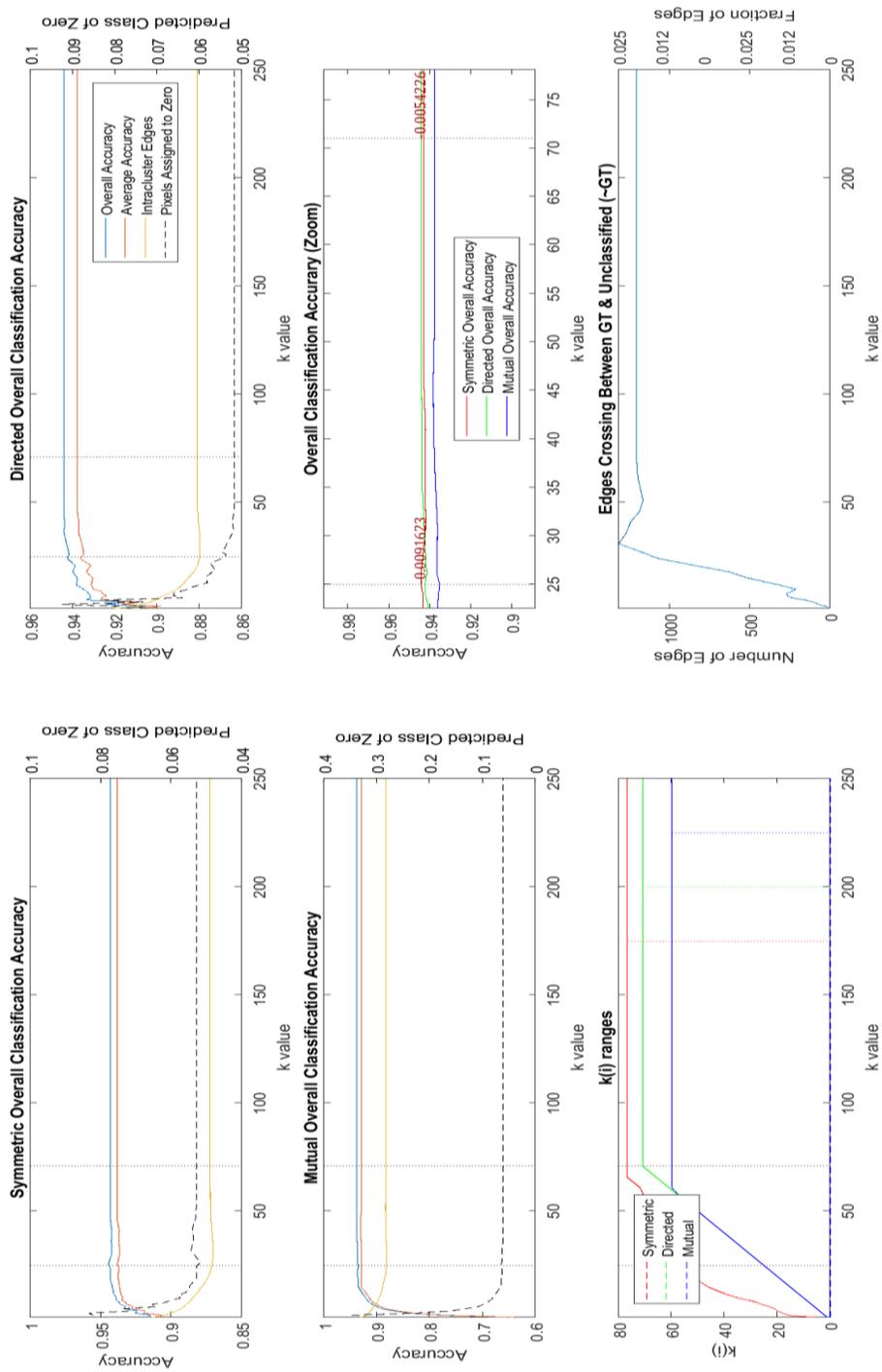


Figure 4-8. ANN Dashboard #1: Accuracies, edges counts, and cross ground-truth edges for *Indian Pines*.

The second dashboard tracks how nearest neighbor lists are impacted as edges are deleted (mutual criterion) or added (superset-symmetric criterion). We start by tracking the number of orphaned pixels for various values of k . Orphaned pixels are tracked by NaNs in this implementation, while pixels assigned to the *unclassified* class are labeled as class zero. Figure 4-9 observations are noted below.

- Starting to the upper left in Figure 4-9, we see that the superset symmetric and directed variants do not have any orphaned pixels because every pixel has some number of outward edges, i.e., $d^+(v) > 0$.
- The mutual k -NN graph does have a large number of orphaned pixels after invocation of the mutuality constraint. This was explained by the low probability of pixels being in each other's NN lists if the list sizes are small (low k). These numbers can be quite high as evidenced by the 50% level attained by the mutual graph in the upper left plot and shown on the right y axis.
- The number of pixels classified as the zero class (those that are better matches to unlabeled data or orphaned pixels) are ever increasing for the superset symmetric and directed graphs. This is due to increasing NN lists, which can contain more and more unlabeled data. We notice some protection from this phenomenon using the mutuality criterion as evidenced by the mutuality line having the lowest fraction of pixels classified as zero.
- The center left plot classifies the edges that were dropped or added from the mutuality or superset symmetric criteria, i.e., for each directed edge that is either made bidirectional or dropped, we count whether the addition or deletion would

have improved (solid lines) or degraded (dotted lines) the health of the neighborhood list. We can see that the net effect of pixels added by the superset criterion is to degrade the structure of the NN list, while pixels removed by the mutual criterion improves NN list health. This supports the previous results.

- The fraction of orphaned nodes created from the mutuality criterion shows a rapid decrease with k , and that (in general) cutting the edge between the pixels was the right thing to do, albeit, sometimes many good edges were severed (center right).
- The total number of ratable pixels shows a steady decline with k , indicating the rate at which pixels are classified as the unlabeled class. We notice an initially sharp rise in the mutual variant due to the reduction in orphaned pixels with k .
- As edges are added, pixels may switch classes due to the composition of its NN list. There are two conditions of changing class membership: pixels that were incorrect and changed to correct (I2C) and those that were correct and changed to incorrect (C2I). We plot the number of class-switching pixels for the superset symmetric and mutual variants from their directed counterpart (lower right). We observe that in general, the superset symmetric criterion causes more pixels to change to the wrong class while the mutual criterion results in more pixels becoming the correct class. Note that the initial decrease in the C2I-mutual plot is due to those pixels that begin initially correct in their directed graph, but are orphaned from their directed neighbor due to invoking mutuality.

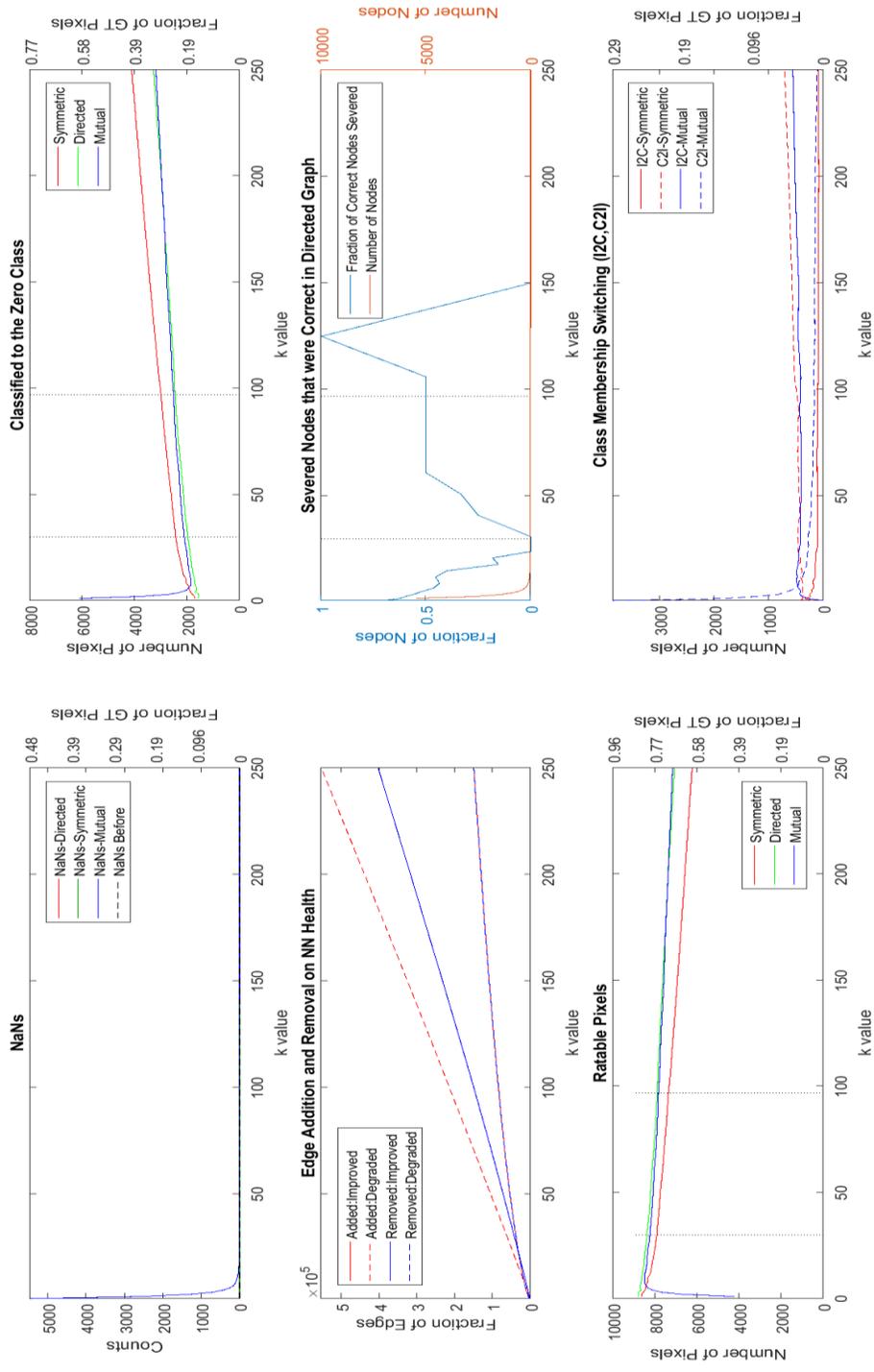


Figure 4-9. *k*-NN Dashboard #2: Impacts from forcing bidirectional edges.

Similar to Dashboard #1, we describe how the observations change for the ANN edge allocation technique compared to the fixed k method (Figure 4-10).

- Despite the apparent zero count of orphaned pixels shown in the upper left, there are actually a small number of orphaned pixels created from termination of the ANN before each pixel has been identified as a neighbor of other pixels; it's just too small to see.
- The number of pixels classified to the zero class levels out instead of increasing with k (upper right). This protection is offered by the ANN algorithm because high numbers of edges are only assigned in dense spectral regions.
- While the edge addition and removal summary is essentially the same (center left), the ANN algorithm does reach a stable level because pixels in dense regions of spectral space receive high edge counts, so adding edges via the superset criterion has a high probability of being correct. Adding edges still has the highest chance of connecting interclass pixels, so the dotted line is the highest in the plot.
- Initially, the number of severed edges connecting a pixel to the correct class can be high (center right) for the mutuality criterion. However, this effect quickly goes away, and the mutual ANN graphs reach a low level intraclass edge severing.
- The number of ratable pixels remains higher as fewer pixels are assigned to the zero class (lower left) from smarter edge creation.
- Class switching pixels (lower right) tell the same story as with the k -NN graph, but the magnitude of the difference between switching modes is much lower due to higher edge allocation in dense spectrally-similar regions.

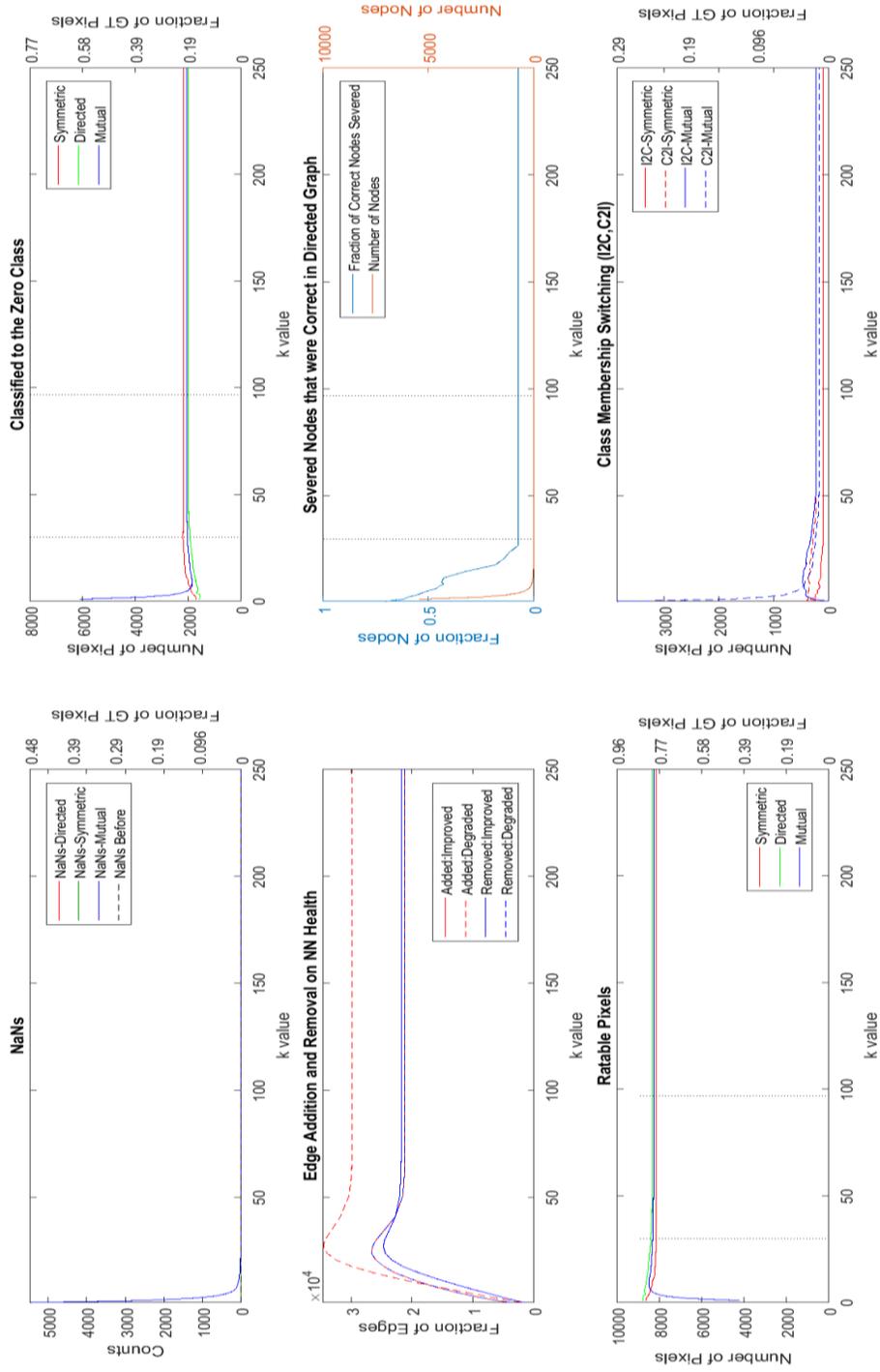


Figure 4-10. ANN Dashboard #2: Impacts from forcing bidirectional edges.

The third dashboard represents a series of miscellaneous measures that make it easier to see differences in absolute and average classification accuracies and track the effects of increasing k on fixed and adaptive techniques. Observations are noted below.

- Starting to the upper left in Figure 4-11, we see that the number of edges present in fixed k -NN graphs are monotonically increasing in k with the highest number of edges being allocated to superset symmetric graph and the least to the mutual variant. The directed graph falls in between. As k increases, all pixels receive larger NN lists, but these new edges don't necessarily result in edges in the final graph. The superset symmetric criterion will likely continue to add additional edges, however, many of them will simply be reconfirming an existing edge, so the increase in k will roll off. However, not all directed edges create bidirectional paths, so enforcing the mutuality criterion only removes some of the potential edges, resulting in a lower increase in edge count. The situation with directed edges naturally falls in between.
- The upper right shows the change in edge accuracy for each adjacency matrix variant, where we see the mutual criterion has the lowest ϕ -edge ratio, a direct consequence of the protection afforded by the mutuality criterion. Adding new edges by the superset symmetric method adds more intercluster edges hence has the highest ϕ -edge ratio. Again, the directed variant falls in between.
- We also see that the mutual criterion has the largest inaccuracy at low k values. This is due to the low probability of pixels having each other as nearest neighbors.

- The center two panels display the differential accuracy between the superset symmetric and mutual variants and their directed counterparts. The winning method over the interval $[k_{avg}, k_{max}]$ is indicated on each graph by a shaded region, where we see the directed k -NN graph performs better than the superset symmetric graph and the mutual graph outperforms the directed graph. These two results align with our discussions on superset symmetric graphs having the highest chance of creating intercluster edges, while the mutuality criterion provides protection via intercluster edges through “mutual” confirmation.
- Both situations reverse for $k < k_{min}$. At very low k , the superset symmetric criterion has the best chance of creating additional edges to intracluster pixels. It is only as k increases that this probability drops to the point relatively more intercluster edges as added. As for the mutuality criterion, this result is simply due to the low chance of having mutual neighbors with only a few edges. These two results do provide some guidance on setting a fixed k . 1) if using a low k , it is best to use the superset symmetric method. If enforcing mutuality, use at least $k = 10$ to get past the region of poor performance. Surprisingly, this $k = 10$ dividing point is consistent in the imagery tested. *We recommend all adaptive techniques that leverage mutuality add a bias to all k (~ 10) values to mitigate this situation.*
- The bottom left graphic is informational and shows that the $[k_{avg}, k_{max}]$ interval is not larger than any class – a condition that can begin to reverse classification accuracy as every added edge will be errant.

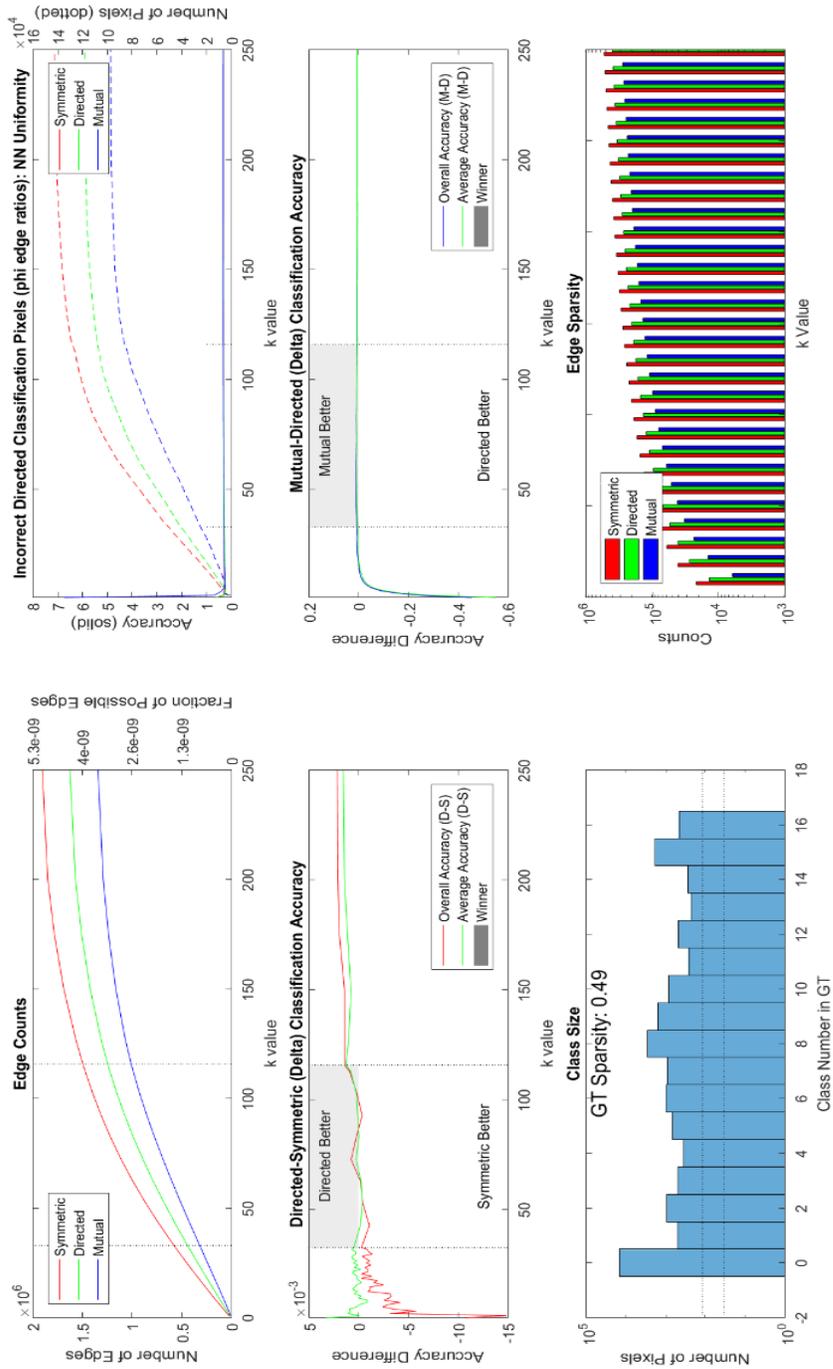


Figure 4-11. k -NN Dashboard #3: Miscellaneous additional measures that vary with k .

Similar to Dashboards #1 and #2, we describe how the observations change for the ANN edge allocation technique compared to the fixed k method. Figure 4-12 observations are noted below.

- The graphs for the number of edges and edge sparsity show a distinctly different pattern than the fixed k graphs. Here we clearly see what appears to be a limit on the number of edges, i.e., the graphs roll over to a relatively constant value somewhere in between k_{min} and k_{max} . This is explainable as adaptive techniques assign a pixel-specific number of edges to each pixel, with only the densest having out-degrees approaching k_{max} . As such, most pixels stop contributing edges to the summation and only a few pixels are left to contribute with each increasing k . Hence we see what appears to be a flattening somewhere in between, but in reality the graph isn't completely flat until k_{max} . At k_{max} , there are no more pixels with edges, so the graph is truly flat.
- ϕ -edge ratios demonstrate a faster roll over effects for the same reason.

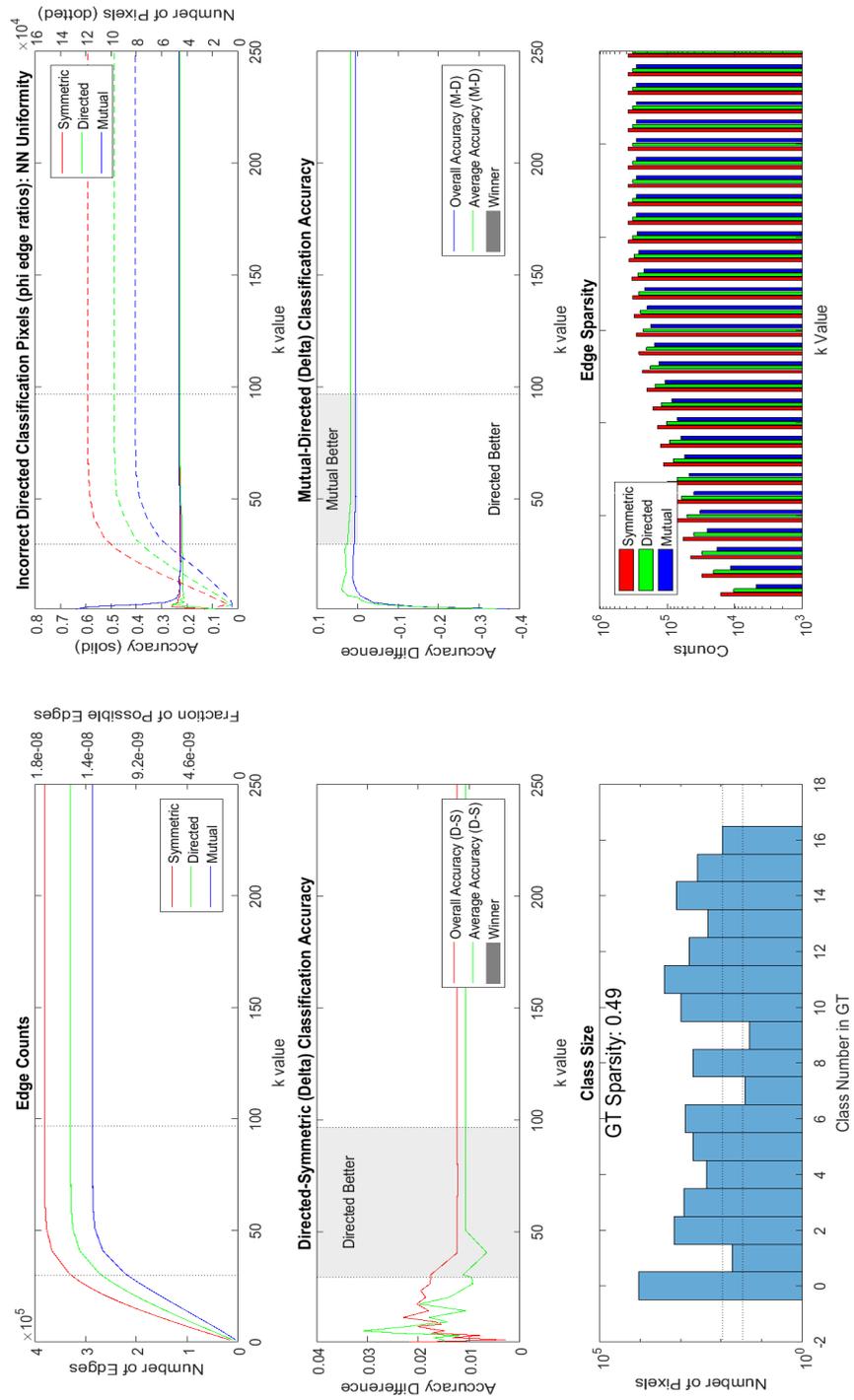


Figure 4-12. ANN Dashboard #3: Miscellaneous additional measures that vary with k.

To summarize, on average, density weighted methods perform as good as or better than the fixed k methods. The maximum overall accuracy differential from worst to best performer in these data is just shy of 7%, with 3% being typical.

4.3.4 Edge reweighting effectiveness

Simple confusion tests only demonstrate classification performance, and therefore cannot be used to evaluate the impact of edge reweighting because the original and reweighted nearest neighbor lists (and hence the classification results) are the same. Examination of the structure of the k -NN lists themselves provides evidence that pixel weights in each neighborhood have improved via edge reweighting techniques such that distances (similarities) of pixels like the test pixel are decreased (increased), whereas those unlike the test pixel are increased (decreased). The metric used to perform this analysis examines the structure of the neighborhoods themselves with respect to cluster uniformity and separation when class labels are available (Figure 4-1).

Seven adaptive graph construction techniques employing edge reweighting were applied to the six images from the previous studies; an eighth, ANN, was used as an unweighted control using primary distance measures. Both the weighted rank and improved pixel count metrics produced effectively the same conclusion—that edge reweighting (post edge selection) is slightly detrimental to the community structure of the neighborhood lists. In short, more pixels had the structure of their nearest neighbor lists degrade instead of improve. In one case, *Indian Pines*, upwards of ~25% of the ground truth pixels with labels had their community structure degrade (Figure 4-13). The leftmost side of the bars in Figure 4-13 is read down and the inset black bar is read up, where the

lower and upper x axes are the percentage of pixels with improved neighborhood health scores and the global average rank-adjusted scores respectively. Algorithms are labeled to the left and data sets to the right. Some patterns are evident.

- ANN occupies six of the top eight positions, indicating that NN lists are generally healthier without reweighting.
- SNN variants are closer to the top of the list, showing the power of counting shared neighbors, but on average are no better than not reweighting.
- The most complex scenes (*Pavia* and *PaviaU*) occupy more of the upper positions suggesting there is more relative benefit for complex (cluttered) scenes.
- Reweighting techniques utilizing codensity generally have poorer performance than those that don't.

A graph construction method that reinforces inherent community structure facilitates accurate analysis; however in this case it appears edge reweighting, by any method, does not improve community structure—from a class separability perspective. This is contrary to what may be inferred from other papers (Fan and Messinger, 2014; Gillis and Bowles, 2012). However, graph construction methods used in those analyses were always followed by an analysis algorithm, i.e., the viability of the new graph construction technique itself was not the focus of the effort. Results in this section suggest those analysis methods may perform better with a simpler graph construction method; avoiding the additional complexity and computation time.

Note this is not to say all reweighting is detrimental for all applications. Indeed, reweighting by the heat diffusion kernel with fixed scales does drive intracluster pixels closer to the test pixel given the cluster assumption is valid. The performance issue discussed above is particular to adaptive-density edge reweighting techniques.

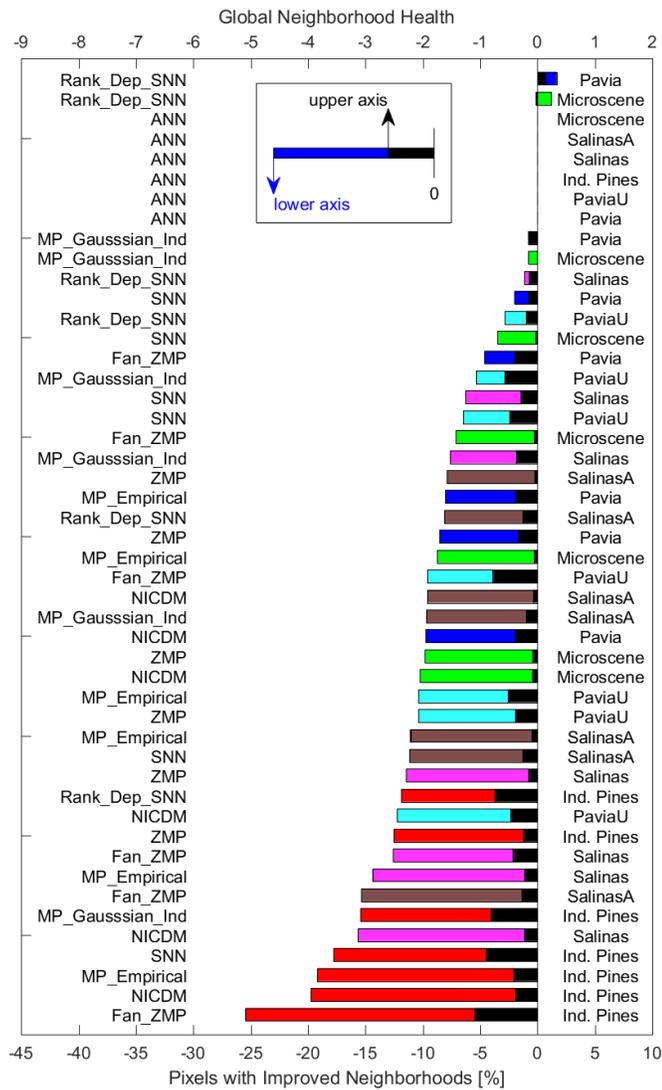


Figure 4-13. Adaptive edge-reweighting effectiveness is detrimental to the NN health.

Anomaly and target detection methods rely heavily on weighted inter-relationships between pixels, as do some classification and clustering algorithms (e.g., normalized cuts). In order to more fully assess graph-construction method impacts on these algorithms, intracluster manifold distance measures and connected component structure should be evaluated.

Overall, this result is interesting given the previous two sections and the utility these methods have shown in other fields, hence warrants more discussion. The pixels with improved neighborhoods is a quantized decision; improved, unchanged degraded. So despite the initially alarming large number of pixels with degraded neighborhood lists observed in Figure 4-13, we need to verify these are not small differences simply exacerbated by the sensitivity of the rank-based tests. The distributional forms of the adjusted rank measures were examined for significant deviations from their originally weighted counterpart using the K-S test ($\alpha = 0.05$). The original and reweighted results are statistically the same for the *Salinas*, *SalinasA*, *Pavia*, and *Microscene* datasets. All methods are statistically different for the *PaviaU* and *Indian Pines* datasets, except for the NICDM and ZMP algorithms. Examining the ZMP and NICDM inset black bars for the *PaviaU* and *Indian Pines* datasets, we observe they have the smallest global health differentials, so relatively speaking there was less change than in the other datasets.

The DBI and SC cluster validity measures were calculated to examine any overlapping nature of data clusters. The *PaviaU* and *Indian Pines* data had poor measures indicating a higher probability of class confusion for these data. To visualize this condition, we normalized the distribution of distances to a pixel by its codensity and examined the

interclass and intraclass distributions of codensity multiples. A possible strong overlap between some classes exists in these data (meadows, trees, asphalt, bitumen), hence neighbors are less likely from the same class than in a more isolated cluster situation. Note that a cluster's diameter may be larger than the separation to other clusters simply due to its shape (elongated instead of isotropic), so some overlap towards larger codensities is possible and most likely the cause of significant overlap at low codensity multipliers. Some classes display very clean separation (tiles, water, bare soil) and should not likely be confused.

Similar analysis to that of Figure 3-4 for all six datasets are examined, and significant interclass scale variation exists indicating that codensity scaling is local to a pixel and not a class as a whole (also not surprising). However, how well can density-based reweighting help in regions of varying density and overlapping classes? From these tests, we see that in most cases, pixels are simply shuffled around with little change to the accuracy of the neighborhood structure. In the case of overlapping classes (*PaviaU* and *Indian Pines*), the results can be detrimental by nearly any method.

One may question how density-based approaches degrade neighborhood uniformity when used for edge reweighting, but improve edge allocation per pixel as in the first two experiments of this section. Per the above discussions, the reweighted edges produce nearest neighbor characteristics similar to those of the original weights even though the magnitudes of those weights are different. However, the use of density for edge count allocation is more forgiving in the sense that small variations in density in dense regions still produce "dense" values. So, in a gross sense, density can drive the number of edges,

but in a fine sense, reassigning weights based on density measures does not offer any improvement as currently defined (and measured).

4.3.5 Primary-metric prescaling versus edge reweighting

Edge reweighting is applied to previously selected edges, whereas the primary metric scaling is applied to measures prior to edge selection; all examples shown thus far are based on reweighting preselected edges. Scaling metrics prior to edge selection facilitates inclusion of other information—density and/or shared neighbors—in the ranking process, so k -NN lists generated from prescaled primary metrics will differ from those presented thus far. Pixels in regions of similar density and spectral similarity should be brought closer together, whereas others are pushed away. As appealing as this may be, it often requires a primary metric for all pairwise distances (or similarities). This takes $O(n^2)$ time so is not practical even for modest size images on a single computer unless approximations can be made. However, to be complete, it is reasonable to explore how much performance is potentially sacrificed for speed by performing a primary metric scaling on a full primary metric matrix.

The same six images and four graph construction techniques from the first two studies are used to demonstrate differences in the number of between-class edges resulting from ZMP prescaling and its effect on classification accuracy (Figure 4-14).

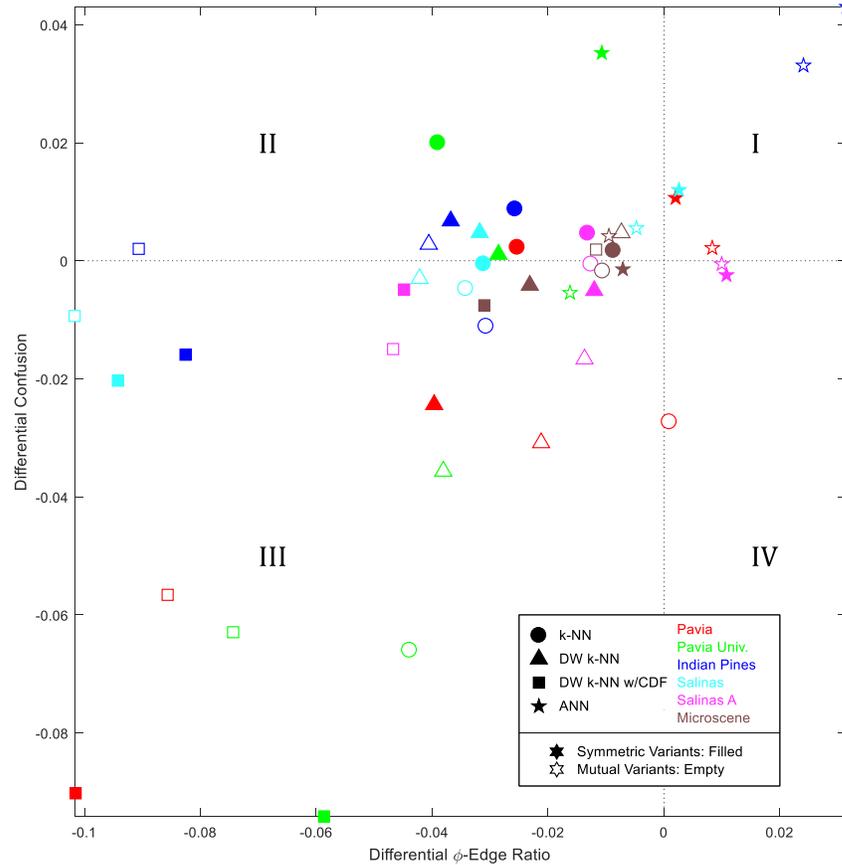


Figure 4-14. Differential ϕ -edge ratio plotted versus the differential confusion score for prescaling results.

As can be seen in Figure 4-14, scaling the primary distance metric via ZMP prior to graph construction can be detrimental to the health of neighborhood lists, i.e., most tests display a negative ϕ -edge ratio (quadrants II and III) indicating ZMP prescaling creates more intercluster (fewer intracluster) edges than simply using the primary metric. This is consistent with the results in the previous section, but surprising given the success of density weighting in other fields (Zelnick-Manor and Perona, 2004; Schnitzer et al., 2012). However, there is some benefit, especially in k -NN and ANN graphs, which all display positive differential confusion values despite the negative ϕ -edge ratios, which says the set

of edges is more semantically correct as a whole. This indicates prescaling prior to edge selection can increase performance and support the method employed by Hou et al. (2013) and Benedetto et al. (2012a) that use k -NN graphs

The mutual proximity method with the Gaussian independent assumption performed much worse than the ZMP method indicating the Gaussian assumption is not appropriate for these data, which is to be expected given pixels do not exhibit normal distance distributions (often skewed). To explore this concept further, prescaling by mutual proximity with the empirical distribution was tested on the *SalinasA* and *Indian Pines* data. Note that empirical mutual proximity uses the distribution of the data to determine the probability of pixels being neighbors and is not impacted by user selection of some arbitrary number of neighbors to average or any assumed distributional form. Empirical mutual proximity demonstrated the highest performance of all prescaling methods indicating that codensity-based measures may not be the best representation of local scale for hyperspectral data. This result warrants more study given that it indicates a potential change in expectation compared to previous efforts.

Flexer and Schnitzer (2012) studied a broad range of public machine learning datasets spanning low to very high intrinsic (hundreds) and extrinsic (thousands) dimensionality. For lower intrinsic-dimensional data, their results showed no gain, and often a reduction in performance (although the reduction was not statistically significant). HSI typically exhibits low intrinsic dimension (single digits to tens of dimensions) (Schlamm et al., 2010), so it is possible we are seeing the same effect here. Their results

did show significant improvement for higher intrinsic dimensionality data, so examination of more diverse, higher dimensionality HSI data should be added to future studies.

Putting performance aside, the choice of edge reweighting or metric prescaling (for appropriate imagery) may be decided by the number of pixels to be analyzed since quadratic run time is often impractical for even modest sized images. However, some of these routines execute quicker using fast nearest neighbor algorithms that first return a larger number of neighbors than required, followed by edge reweighting and subsequent edge selection. Additionally, the emergence of massively-parallel scalable (cloud) computing opens a new realm of possibilities for quadratic run time algorithms if approximations are not possible. However, a better algorithm always scales better than more cores as data sizes grow.

4.4 Summary and Contributions

We have provided a survey of many common graph construction techniques used by the HSI community as well as introduced two new methods from the computer vision and data mining literature: shared nearest neighbors and mutual proximity. Graph construction techniques were analyzed to reveal characteristics most beneficial to the health of neighborhood lists. Specifically, invoking mutuality was shown to be superior to superset symmetry for symmetrifying adjacency matrices in most cases (except at low k), and adaptive density weighted edge allocation methods were demonstrated to produce more uniform nearest neighbor lists. Advanced adaptive reweighting and primary-metric prescaling techniques were shown to slightly degrade the community structure of spectral graphs, and can be detrimental to analysis techniques requiring edge weights closely

representing community structure, not to mention the additional computational load. This research questions the reweighting approaches published thus far for creation of spectral graphs as they do not perform the same way on real data as they do in well-constructed examples with uniform cluster density to demonstrate their conceptual function.

The described neighborhood health metrics are good indicators for classification related tasks, but not complete for analyses that utilize edge weights, or require graph connectivity. As such, future research can quantitatively explore graph characteristics more important to hyperspectral anomaly and target detection, detailing which construction techniques are best suited for each task through analysis of the connected component structure and manifold distances on these graphs. Perhaps reweighted edges are beneficial to these techniques, albeit not beneficial to NN health.

While adaptive reweighting methods did perform slightly more poorly, there is still promise for these techniques if they can be modified, e.g., remove dependence on codensity and use larger fixed k values to generate higher probabilities for overlap in SNN and MP. Until such further research is conducted, there does not seem to be much benefit from adaptive edge reweighting, so best to stick with a single scalar for modifying the heat diffusion kernel until better local density measures are available. Additional metrics should be explored that measure the same phenomenon from different perspectives.

5

CONCENTRATION OF MEASURE AND HUBNESS IN GRAPHS OF HYPERSPETRAL DATA

In the previous chapter, we began to question the use of codensity for rescaling primary metrics or reweighting edges. We attributed this to local scale variations that impacted the utility of characterizing local scale by a single number, codensity. In this chapter, we continue of exploration of possible effects impacting the use of codensity to represent data scale in high-dimensional spaces.

The success of any graph-based analysis is naturally dependent upon the quality of the graph constructed from the spectral data (Mercovich et al., 2011). Edges must be judiciously chosen to accurately model community structure inherent in the data without over (or under) connecting the nodes. As discussed in Chapter 2, many common graph construction techniques are founded on the k -nearest neighbor (k -NN) relationship because of its adaptability to data scale and density as well as its ability to follow clusters of arbitrary shape. Characteristics of k -NN (graphs) in high-dimensional spaces have been explored by the data mining and machine-learning communities (Beyer et al., 1999; Radovanović, Nanopoulos, and Ivanović, 2010a; Durrant and Kabán, 2009). Their research indicates directed graphs (digraphs) and undirected graphs built without enforcing the mutuality criterion (either implicitly or explicitly) are subject to the impacts of hubs, a relatively newly discovered aspect of the curse of dimensionality (Radovanović et al.,

2010a). Most spectral graph generation methods found in the remote-sensing literature utilize variants enforcing mutuality (Albano et al., 2012b; Mercovich et al., 2011; Ziemann, Messinger, Albano, and Basener, 2012); the superset symmetric criterion may produce better performance at low k values (either fixed or adaptive in nature). Additionally, the use of digraphs to advance graph construction and/or analysis techniques should not simply be dismissed in favor of simpler undirected graphs without additional research.

The remainder of this chapter explores the hubness phenomenon and one of its requisite conditions, concentration of measure, in HSI data. We establish the relevance of these two aspects of the curse of dimensionality to hyperspectral data analysis through studies on 49 datasets, and offer perspective on their impacts to HSI analyses, especially as spectral resolutions continue to increase, e.g., AVIRIS next generation (AVIRIS-NG) has 425 bands.

5.1 Concentration of Measure

The curse of dimensionality is a catch all phrase referring to difficulties experienced while working in high dimensional spaces. Concentration of measure is one of many aspects of the curse affecting machine learning applications (Radovanović et al., 2010a; Schnitzer et al., 2012) through degradation in the efficacy of many common distance (and similarity) metrics with increasing dimensionality (Ertöz et al., 2003; Houle et al., 2010). That is to say, in high-dimensional vector spaces, all pairwise distances can appear similar, so much so that the concept of nearest neighbor itself is in question (Beyer et al., 1999). Other curse aspects include hyperspace sparseness (empty space phenomenon), volume

concentration to hyperspheres, computational inefficiencies, and diagonals being perpendicular to cardinal axes (Wang, 2011).

Euclidean distance measures will largely be discussed from this point forward; results apply equally well to other distance and similarity measures (Francois et al., 2007; Radovanović, Nanopoulos, and Ivanović, 2010b; Heylen et al., 2017).

Concentration of measure is typically expressed as a ratio of values representing data spread to data magnitude. Beyer et al. (1999) demonstrate that the relative contrast—the normalized differential distance between a point’s closest and farthest neighbors—decreases with increasing dimensionality in *i.i.d.* data, which can render the distance measure less efficient (even meaningless) in high dimensions. Using the origin as our reference point⁸, let $\mathbf{x}^{(i)}$ represent an instance (spectrum) of the dataset (spectral image) $D \in \mathbb{R}^d$ where $|D| = n$, $\|\cdot\|_p$ is the L_p -norm of the vector, $D_{max} = \max\|\mathbf{x}^{(i)}\|_p$ and $D_{min} = \min\|\mathbf{x}^{(i)}\|_p$ for $i = 1, \dots, n$, we find the relative contrast vanishes as dimensionality increases (5.1) (Beyer et al., 1999).

$$\lim_{d \rightarrow \infty} \frac{D_{max} - D_{min}}{D_{min}} \rightarrow 0 \quad (5.1)$$

The maximum value of the vector norm (pixel magnitude) is a function of p , such that $\|\mathbf{x}^{(i)}\|_p \in [0, \|\mathbf{1}_d\|_p]$ if the data have been normalized, e.g., $\|\mathbf{1}_d\|_2 = \sqrt{d}$ for the Euclidean (L_2) norm.

⁸ Distances to each pixel from the origin are simply the vector norms, so we center our data to simplify notation.

Other authors utilize the standard deviation and expected value to represent spread and magnitude (Francois et al., 2007; Durrant and Kabán, 2009). In this case, the relative variance also concentrates as

$$\lim_{d \rightarrow \infty} \frac{\sigma}{\mu} \rightarrow 0, \quad (5.2)$$

where $\sigma = \sqrt[p]{\text{var}(\|\mathbf{x}^{(i)}\|_p)}$ and $\mu = E[\|\mathbf{x}^{(i)}\|_p]$ (Francois, 2007). Kabán (2010) shows

that convergence of the relative contrast and relative variances are equivalent.

Both (5.1) and (5.2) are true regardless of the value of p , but to different degrees for *i.i.d.* data (Francois et al., 2007). Many authors have explored this phenomenon for $p \in \mathbb{R}^+$, where we find that concentration is generally less severe with smaller values of p and norms with $p \geq 3$ are all but meaningless in high-dimensional spaces (Hinneburg, et al. 2000; Aggarwal, Hinneburg, and Keim, 2001). We will focus solely on the L_2 norm (without loss of generality) because it is the most widely used norm in the spectral remote sensing community. The subscript ‘ p ’ or ‘2’ is dropped for notational convenience.

To illustrate this effect for two common metrics used in the spectral remote sensing literature—Euclidean and cosine distances—we recreate an analysis similar to that shown in Flexer and Schnitzer (2013) wherein we study the aforementioned contrasts as a function of increasing vector space (extrinsic) dimension⁹. The Figure 5-1a displays the behavior of Euclidean distance from the origin determined by the average statistics of 200 experiments,

⁹ The cosine similarity is more common in HSI analysis, but we use the cosine distance for continuity of the discussion. The same outcome applies.

each with $n = 10^4$ points randomly drawn from the unit hypercube, $[0,1]^d$, as well as the one standard deviation distances, the observed minimum and maximum distances from the origin, and the theoretical maximum (increases as \sqrt{d}) and minimum (0.0) distances in \mathbb{R}^d .

When dimensionality is low, the observed minimum and maximum norms are very close to 0.0 and \sqrt{d} respectively. However, as dimensionality increases, the average distance to the origin increases due to the addition of more terms in the L_2 norm summation. Contrary to initial intuition, the empirical maximum and minimum distances no longer span the possible theoretical range, but rather occupy a small region of hyperspace, i.e., a thin hypershell centered at the origin¹⁰. At the same time, the standard deviation of distances remains about the same such that the ratio of spread to magnitude decreases with increasing dimensionality, i.e., the distance measure concentrates (Figure 5-1b). Thus, the discriminating power of Euclidean distance as a primary measure diminishes as the number of dimensions increases. Francois et al. (2007) point out that concentration is more a function of the intrinsic rather than extrinsic dimension, hence this effect is practically less severe than that expected from theoretical examination of independent variables, so we will explore this effect in correlated HSI data. The implication for spectral remote sensing is that user-defined scales such as σ in the heat kernel reweighting (2.4) are more difficult to determine (Belkin and Niyogi, 2003).

Figure 5-1c demonstrates that the cosine distance concentrates as well, but for a different reason. The mean distance remains relatively constant and the standard deviation decreases with dimensionality, also concentrating measurements in high dimensions. The

¹⁰ This effect is largely independent of the number of samples (Beyer et al., 1999; Heylen et al., 2017).

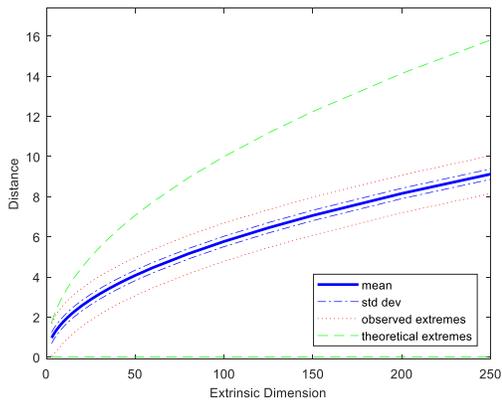
functional form of the concentration function is proven in (Pestov, 2000). The implication for spectral remote sensing is that user-defined thresholds for spectral-angle based metrics such as the spectral angle mapper (SAM) are more difficult to select if the cosine similarity does indeed concentrate.

It is interesting to note that Euclidean distance concentration (Figure 5-1a) can be made to look like the cosine distance concentration (Figure 5-1c) by simply normalizing the graph by \sqrt{d} (Figure 5-1e). In this case, the mean remains constant around $\frac{1}{\sqrt{3}} = 0.577$ and the standard deviation decreases. Zimek, Schubert, and Kreigel (2012) explain that these effects are simply the application of the central limit theorem with shrinking variance.

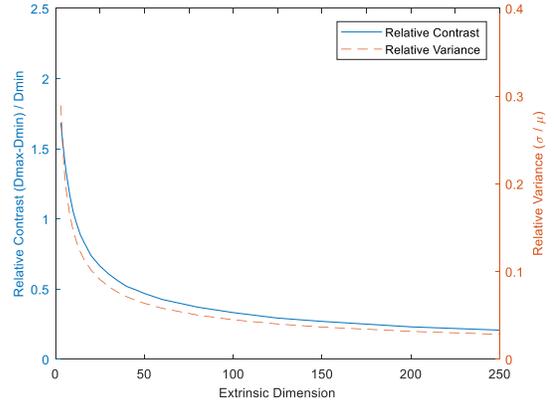
The traditional notion of density based on Euclidean distance can also be essentially meaningless in high dimensional spaces because data density decreases rapidly with increasing d -dimensional volume (Ertöz et al., 2003). As such, high dimensional spaces are sparse (Jimenez and Landgrebe, 1998) and methods using fixed distances to determine hyperspace density are destined to fail without modification (Ertöz, Steinbach, and Kumar, 2002).

From a spectral-graph construction perspective, this implies adjacency (affinity) relationships should be built upon measures that do not rely on fixed density or distance thresholds in high dimensional spaces. k -NN graphs satisfy this constraint, however many edge reweighting methods are based on some measure of distance or density in high dimensions, and this phenomena may skew the intended purpose. Despite the shortfalls of Euclidean and cosine distances as primary distance measures in high dimensional spaces,

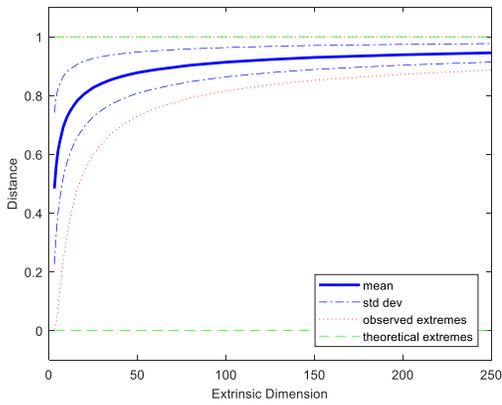
they can effectively serve as the basis for secondary measures such as shared nearest neighbors (Houle et al., 2010).



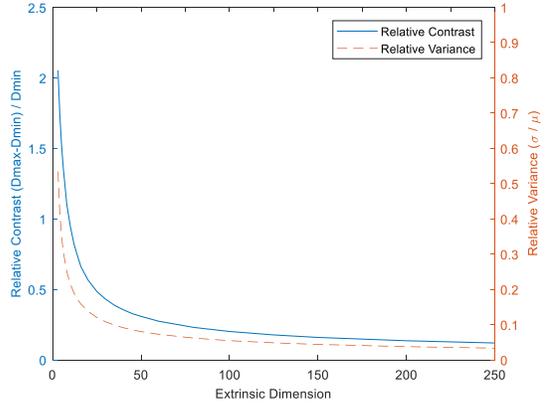
(a)



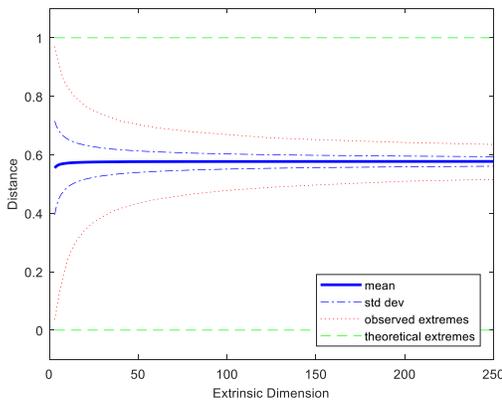
(b)



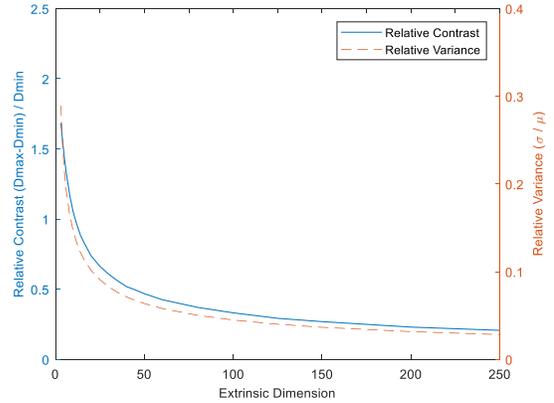
(c)



(d)



(e)



(f)

Figure 5-1. Concentration of Euclidean (top row), cosine distances (middle row), and normalized Euclidean distance (bottom row).

5.2 Enter Hubness

Distance concentration in high dimensional spaces may give rise to the hub phenomenon¹¹. The term hub originates in network theory and simply refers to a node with an in-degree greatly exceeding the average, i.e., nodes that have small distances (or equivalently high similarities) to a relatively large number of other nodes. We will see that hubs can unwantedly show up in nearest neighbor lists despite the lack of apparent similarity and have even been called obstinate nodes by Radovanović, Nanopoulos, and Ivanović (2010b). However, that does not mean all hubs are bad as hubs that promote strong intracluster relationships can be beneficial to many analyses. The existence of hubs in HSI graphs and their impacts on subsequent analysis (if any) has yet to be studied to our knowledge.

To understand the formation of hubs, recall the previous discussion, wherein we established data volume concentrates in a hypershell about the data mean. This hypershell has a small but non-negligible thickness (Francois et al., 2007), so there will be some pixels closer to the mean than others. Radovanović et al. (2010a) explain how these pixels have a higher probability of being included in neighborhood lists of many other points and therefore can become hubs.

Following the nomenclature of Flexer and Schnitzer (2013), let O^k represent the number of k -occurrences of pixel $i \in D$, where $D \in \mathbb{R}^d$, i.e., the number of times a node occurs in the k -NN lists of other nodes¹². As data dimensionality increases, the O^k

¹¹ Hubs are a general problem in high-dimensional machine learning (Radovanović *et al.*, 2010b).

¹² A k -occurrence is another name for a node's in-degree, $d^-(v)$. Note that the k -occurrence histogram may also be called the in-degree distribution. Nodes originating edges incident upon a vertex are also called reverse nearest neighbors.

distribution skews to the right (positive skew). This skew can be considerable if the right conditions exist (Zou and Zhu, 2011). We classify pixels with five times the number of average k -occurrences as hubs in this work.

Prior to exploring hubness in HSI data, we first explore hubness in random normal data to obtain a qualitative understanding of the phenomenon (Figure 5-2). We measure the Euclidean distance from the mean and cosine distance from the first cardinal axis for *i.i.d.* data of dimensions $p = 3, 20, \text{ and } 100$, in the left, middle, and right columns respectively, then construct 5-NN graphs. We form the in-degree distribution, O^5 , and observe trends in node connectivity as a function of distance from the mean (or first cardinal axis). As can be easily seen in Figure 5-2, in higher dimensions, the node in-degree increases as distance decreases. This effect produces a few nodes with much higher in-degrees than the rest of the dataset. These are hub nodes.

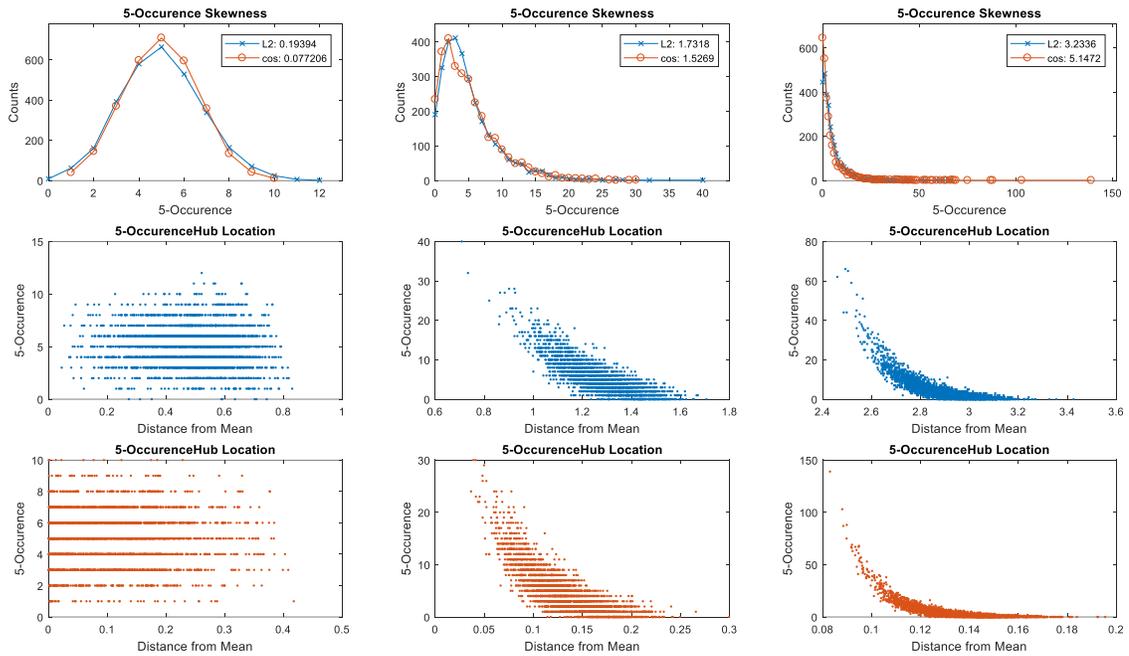


Figure 5-2. Skewness in the \mathcal{O}^5 histogram of random normal *i.i.d* data for $p = 3, 20, \& 100$ (top row). \mathcal{O}^5 vs. the L_2 (middle row) and cosine (bottom row) distances from the mean. Modified from Radovanović et al. (2010a).

Qualitatively, how can we explain these changes with dimension? Let us first begin by remembering the concentration phenomenon. In a single dimension, data will span the entire range. However, if we now add a second dimension (band), the probability of a point having simultaneous extreme values is now the product of the two independent probability distributions; hence it likely that the distribution will begin to show decreased density at the extremes (Figure 5-3).

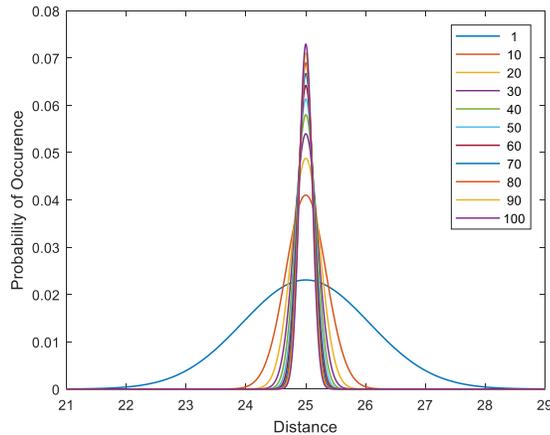


Figure 5-3. Decrease in probability of extreme pixels in multidimensional *i.i.d.* data.

This effect increases as we add additional dimensions, where it is increasingly difficult to find values that are simultaneously low (or high). As such, we should begin to see data concentrate towards the center of the distance distribution (denser regions in right column of Figure 5-2). Radovanović et al. (2010a) describe how there are some nodes closer to the mean as data concentrates. These nodes have a higher chance of being closer to the mean within the hypershell and at the same time closer to many other nodes. Hence we observe a few pixels, closest to the mean, but at the same time highly connected to other nodes (high 5-occurrence values).

We now create a similar plot to Figure 5-2 for real HSI data (Figure 5-4). The left graph shows the 30-occurrence histogram for a *Microscene* hypercube (Allen, Resmini, Deloye, and Stevens, 2013). Notice the long right tail (positive skew) and the few nodes with much higher in-degrees (filled circles) than most other nodes. Interestingly, hubs are anomalous pixels of sorts because they occupy the few farthest positions in the k -

occurrence histogram just as outliers occupy the few farthest positions in distance (or similarity) histograms (Radovanović et al., 2010a)¹³.

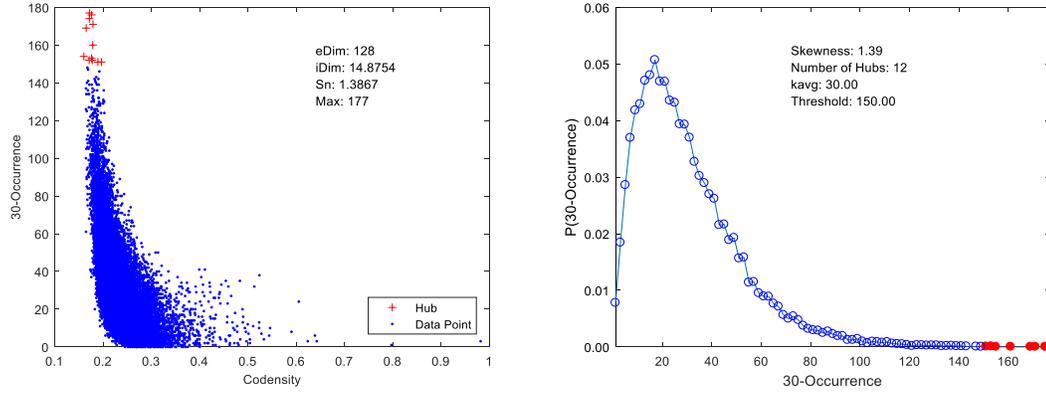


Figure 5-4. Hub location and skewness of the 30-occurrences histogram for an example *Microscene* hypercube.

From a geometric perspective, hubs often have shorter codensities than other nodes. This can easily be seen to the right of Figure 5-4 where all 12 hubs are located towards the far left of the codensity distribution, i.e., hubs exist in denser regions of spectral space.

Radovanović et al. (2010a) quantify hubness (h) as the skewness (normalized third moment) in the distribution of k -occurrences given by

$$h = \frac{E\{(O^k - \mu)^3\}}{\sigma^3}, \quad (5.3)$$

where μ and σ are the mean and standard deviation of the k -occurrence distribution respectively.

¹³ $k_{avg} = k = 30$ because the mutuality or superset symmetry criteria have not been enforced—yet.

As an example of how hubs can impact community structure, consider the case where the k -NN lists associated with many nodes all contain node b . Each node then has $(k-1)$ nearest neighbors from which to establish connectivity (edges) describing community structure amongst other nodes. This is especially true if k is small because hubs can occupy enough NN positions in a node's neighborhood to reduce the fidelity of inferred localized structure and algorithms like k -NN may repeatedly count hubs in NN lists.

Hubs also adversely impact the structure of the resultant graph because b can only have k -NN if invoking the mutuality criterion, i.e., many edges to node b are lost after imposing mutuality.

Edges incident upon hubs can be good or bad. Good edges enforce community structure (e.g., nodes in dense cluster centers), while bad edges may traverse regions of differing density to connect intercluster nodes, corrupting geodesic distances described by manifolds (Francois et al., 2007). We will study the existence of good and bad hub edges in graphs of HSI data towards the end of this chapter.

To understand how this skewness changes with dimensionality, we again turn to the Euclidean and cosine distances calculated from the uniform distribution on the unit hypercube. Figure 5-5 plots the skewness (hubness) of the 30-occurrence histogram as a function of data extrinsic dimensionality, where we see hubness is more a function of the intrinsic, not extrinsic, dimensionality (Radovanović et al., 2010a). Hubs exist in relatively denser regions of spectral space and to the right of the 30-occurrence histogram (filled circles to the left). Histogram bins with zero values have been removed in Figure 5-4 for ease of viewing. There are 12 hubs in these data with skewness of $h=1.39$.

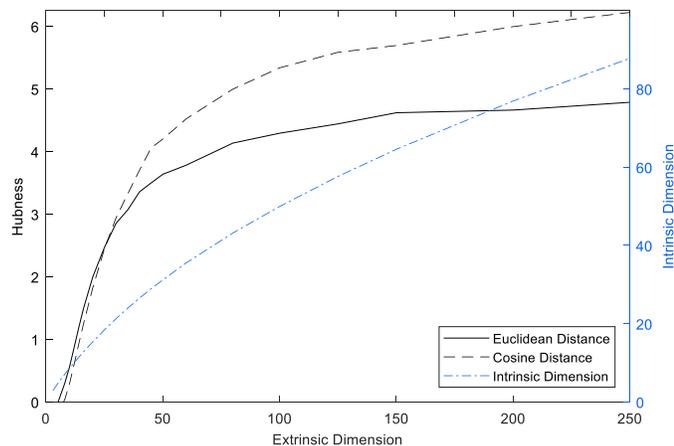


Figure 5-5. Hubness of Euclidean and cosine distances for *i.i.d.* data on $[0,1]^d$.

We will see that hubs do exist in graphs of HSI data, and the current trend towards higher spectral resolution systems that collect large areas will increase the probability of hubs, especially in urban scenes which can demonstrate higher intrinsic dimensionality due to the larger diversity of materials. However, higher dimensionality alone does not guarantee concentration of measure, so hubs are not always present in real high-dimensional data as shown in Durrant and Kabán (2009).

Real data are often comprised of multiple clusters and are therefore not well represented by the single cluster example of Figure 5-5. Radovanović et al. (2010a) prove that hubs are associated with cluster centers in multicluster data instead of the dataset mean. To verify this in HSI data, we plot 30-occurrences versus pixel distances from the data and cluster means in Figure 5-6; cluster means are determined with a simple k -NN algorithm with k set to twice the intrinsic dimension. Combining the concepts from Figure 5-4 (short

distances) and Figure 5-6 (dense spectral regions), we find that hubs are associated with the densest spectral clusters.

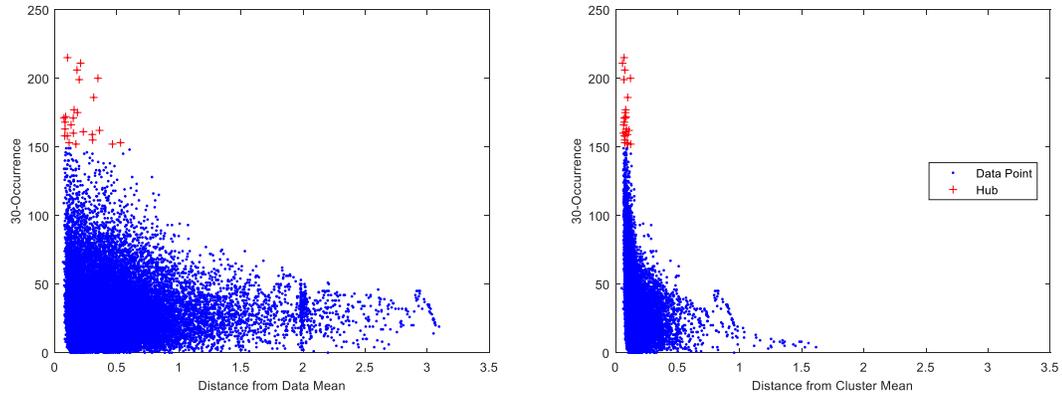


Figure 5-6. Hub location with respect to the data (left) and cluster means (right) for the *hyperdesert* scene.

Real data also exhibits some correlation, the impacts of which are explored in the next section. We will find correlation can strongly impact distance concentration in real data which can ultimately influence the presence of hubs.

Recall the three graph types discussed in Chapter 2: directed k -NN graphs, mutual k -NN graphs, and superset symmetric k -NN graphs. Of the three types, mutual k -NN graphs are the only types not susceptible to hubness because each node can have at most k incident edges as a result of the mutuality criterion; ΔG in (1.4). Most common methods of spectral graph generation make use of the mutuality criterion hence are not susceptible to hubs. However, recent studies have shown that using the symmetric criterion may be better for the small k conditions (Chapter 4), so the impact of hubs is a topic of concern. Additionally, while much of the remote sensing literature has focused on simple graphs, the possibility

of using directed graphs should not be ignored as this technology matures. Both undirected k -NN graphs created using the superset criterion and directed k -NN graphs are susceptible to hubness because they do not practically limit the number of edges incident upon a vertex; $\Delta(G)$ in (1.3).

In contrast to fixed k -NN methods, adaptive k -NN algorithms increase the size of k -NN lists in relatively higher density regions of spectral space per (1.3) and (1.4). These denser regions (near cluster centers) are exactly where hubs are located (Figure 5-6), so increasing the number of edges can facilitate the formation of hubs. Studying fixed and adaptive methods on the same datasets will provide insight into this situation.

5.3 Concentration of Measure in HSI Data

Let us begin by first exploring concentration of pairwise distances within HSI data because it is one of the necessary conditions for hubness to emerge (Radovanović et al., 2010a). Forty-nine datasets with varying numbers of bands, sensors, altitudes, spatial resolutions, data types (reflectance or radiance), and scene content (e.g., urban, rural, snow, ocean, and desert) are used to explore the magnitude of distance concentration in HSI data. Dataset details are provided in Table 5-1.

Table 5-1. Hyperspectral Evaluation Data

Dataset	Bands	Sensor	Alt. [km]	GSD [m]	Data Type	Source
HyperDesert	189	Hyperion	705	30	<i>rad.</i>	USGS
Deepwater-Rad	316	Prospectir	1.7	2.2	<i>rad.</i>	Spectir
Deepwater-Rad2	316	Prospectir	1.7	2.2	<i>rad.</i>	Spectir
Microscene13	80	Pika II	0.3	1E-4	<i>ref.</i>	NIST
MidWest	189	Hyperion	705	30	<i>rad.</i>	USGS
Rockville	189	Hyperion	705	30	<i>rad.</i>	USGS
BlueDotMS	128	SOC710	0.3	1E-4	<i>ref.</i>	NIST
RedSea-Rad	128	Prospectir	1.5	1.95	<i>rad.</i>	Spectir
Syracuse	189	Hyperion	705	30	<i>rad.</i>	USGS
JasperRidge	204	AVIRIS	19.4	19.37	<i>ref.</i>	USGS
Tuscon2	189	Hyperion	705	30	<i>rad.</i>	USGS
Tuscon	189	Hyperion	705	30	<i>rad.</i>	USGS
Botswana	145	Hyperion	705	30	<i>rad.</i>	GIC
Phoenix	189	Hyperion	705	30	<i>rad.</i>	USGS
Albuquerque	189	Hyperion	705	30	<i>rad.</i>	USGS
Cuprite	192	AVIRIS	20.3	20.3	<i>ref.</i>	USGS
NorthLA	189	Hyperion	705	30	<i>rad.</i>	USGS
MicroArray	80	Pika II	0.3	1E-4	<i>ref.</i>	NIST
Sacramento	189	Hyperion	705	30	<i>rad.</i>	USGS
Foster1	33	<i>Hama</i>	<i>gnd.</i>	1	<i>ref.</i>	<i>Manchstr</i>
MicrosceneLeaf	80	Pika II	0.3	1E-4	<i>ref.</i>	NIST
ColumbiaRiver	189	Hyperion	705	30	<i>rad.</i>	USGS
LunarLake	197	AVIRIS	21.4	21.4	<i>ref.</i>	USGS
Vegas	189	Hyperion	705	30	<i>rad.</i>	USGS
IPines	200	AVIRIS	20	20	<i>rad.</i>	GIC
Rochester2	189	Hyperion	705	30	<i>rad.</i>	USGS
Rochester	189	Hyperion	705	30	<i>rad.</i>	USGS
Cuprite-Rad	316	Prospectir	3.3	4	<i>rad.</i>	Spectir
Cuprite-Ref	316	Prospectir	3.3	4	<i>ref.</i>	Spectir
PaviaU	103	ROSIS	2.3	1.3	<i>ref.</i>	GIC
Foster2	33	<i>Hama</i>	<i>gnd.</i>	0.02	<i>ref.</i>	<i>Manchstr</i>
MoffetField	203	AVIRIS	21.5	21.5	<i>ref.</i>	USGS
Gulf-Rad	316	Prospectir	1.6	2	<i>rad.</i>	Spectir
PurdueIPines	193	AVIRIS	20	20	<i>rad.</i>	Purdue
Fairbanks	189	Hyperion	705	30	<i>rad.</i>	USGS
Gulf-Ref	316	Prospectir	1.6	2	<i>ref.</i>	Spectir
Reno-Ref	316	Prospectir	1.5	2	<i>ref.</i>	Spectir
Pavia	102	ROSIS	2.3	1.3	<i>ref.</i>	GIC
Beltsville-Rad	316	Prospectir	0.7	0.65	<i>rad.</i>	Spectir
Barilla	360	Prospectir	0.7	1	<i>ref.</i>	RIT
Reno-Rad	316	Prospectir	1.5	2	<i>rad.</i>	Spectir
CookeCity	126	HyMap	1.4	3	<i>ref.</i>	RIT
Beltsville-Ref	316	Prospectir	0.7	0.65	<i>ref.</i>	Spectir
Conesus	360	Prospectir	0.3	0.5	<i>ref.</i>	RIT
AvonField	360	Prospectir	0.7	1	<i>ref.</i>	RIT
DCmall	191	HYDICE	6	3	<i>rad.</i>	Purdue
KSC	176	AVIRIS	18	18	<i>rad.</i>	GIC
SalinasA	204	AVIRIS	3.7	3.7	<i>rad.</i>	GIC
Salinas	204	AVIRIS	3.7	3.7	<i>rad.</i>	GIC

Dataset characteristics. *Shorthand notation* and acronyms: reflectance (*ref.*), radiance (*rad.*), University of Manchester (*Manchstr*), Hamamatsu (*Hama*), ground based-side looking (*gnd.*), United States Geological Survey (USGS), Rochester Institute of Technology (RIT), Kennedy Space Center (KSC), National Institute of Standards & Technology (NIST), Grupo de Inteligencia Computacional (GIC), District of Columbia (DC), National (*Nat.*), Ground Sample Distance (GSD), Reflective Optics System Imaging Spectrometer (ROSIS), Airborne Visible/Infrared Imaging Spectrometer (AVIRIS), Surface Optics Corporation (SOC), Hyperspectral Digital Imagery Collection Experiment (HYDICE).

Most data may be obtained via sources in Appendix A. Bold datasets have associated class labels (ground truth).

To measure concentration, we determine the fraction of pixels from a dataset of size n that have at least 50% of all other pixels (j) within some scale factor (c) of pixel i 's codensity measurement, δ_i (5.4). Codensity is calculated as the average of the distances to each pixel's third through fifth nearest neighbors (2.1) in this experiment. The logical test inside the parenthesis results in a binary output $\{0,1\}$ which is summed to count the number of pixels passing the condition over $c \in [1,25]$.

$$\frac{1}{n-1} \sum_i \left(\left(\sum_j d(v_i, v_j) < c \cdot \delta_i \right) \geq \frac{n-1}{2} \right), \quad \forall j, j \neq i \quad (5.4)$$

Similar plots were used to explore the concentration phenomena (Durrant and Kabán, 2009) and (Beyer et al., 1999), and we adapt them here to be consistent with our use of codensity as a more robust measure of data scale than simply the 1-NN distance. Results are shown in Figure 5-7, where we see a very broad range of concentration amongst these data. Concentration curves are shaded by the data's intrinsic dimension (Heylen et al., 2017), where black represents data with the highest intrinsic dimension, and lite grey the lowest. A broad range of concentration characteristics are exhibited by these data, and we can see that data exhibiting higher intrinsic dimension is generally more concentrated than those with lower intrinsic dimension.

Table 5-1 datasets are sorted by concentration ranking, where the concentration rank is determined by sorting the codensity multiplier (x axis) values of a concentration curve's intersection with a line drawn from the upper left (0,1) to the lower right (25,0).

This assigns a concentration rank of one and 49 to intersections in the upper left and lower right respectively. The Spearman rank correlation of intrinsic dimension with concentration rank is 0.50, indicating some dependency, but there must be other factors at play.

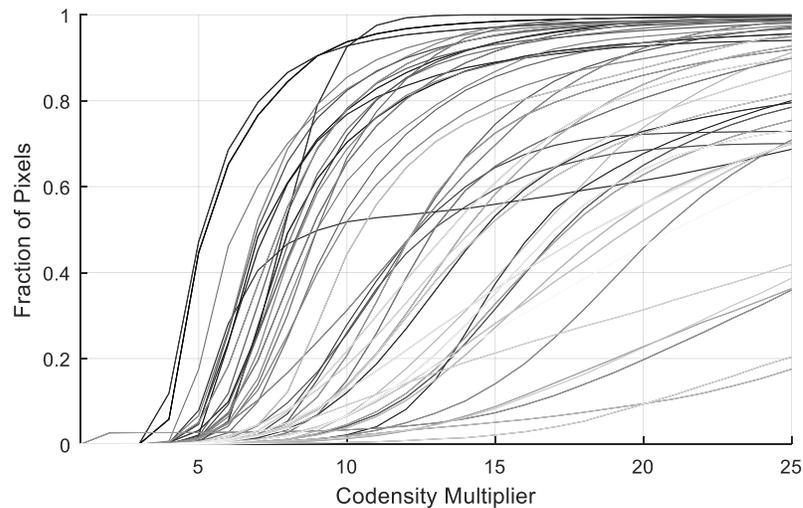


Figure 5-7. Distance concentration of Table 5-1 datasets shaded by intrinsic dimension.

To explore how the existence of cluster structure and correlation influences our results, we permuted the pixel locations in every band to reduce correlation and cluster structure, hence artificially increasing the data's intrinsic dimension without altering the distribution of intraband pixel intensities (Francois et al., 2007). The intrinsic dimension increased, but did not approach the extrinsic dimension due to the strong spatial coherence in HSI datasets, i.e., randomly selecting a band value from another pixel in a scene with a few large clusters has a good chance of selecting similar (intracluster) values, loosely preserving some band-to-band correlation. Similar concentration analysis to Figure 5-7

reveals all datasets but five (*KSC*, *MidWest*, *Foster1*, *Foster2*, *Deepwater-Rad*) are ~100% concentrated by a codensity multiplier of two ($2\delta_i$), demonstrating the importance of cluster structure and correlation in reducing the concentration effect.

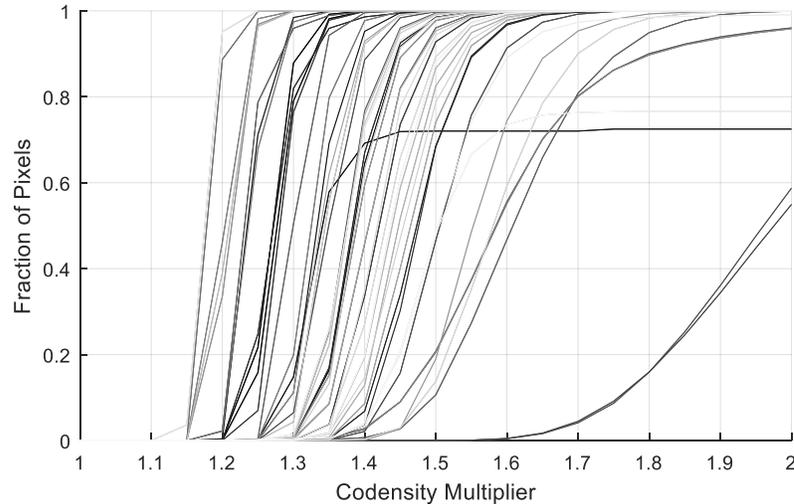


Figure 5-8. Distance concentration of permuted HSI data. Notice these data are concentrated by much smaller multiples of the codensity measurement.

For comparison purposes, *i.i.d.* data is ~100% concentrated by $1.2\delta_i$ and $1.05\delta_i$ of the codensity for data spanning the minimum (80) and maximum (360) dimensions of the data in Table 5-1, respectively. The *i.i.d.* data are much more concentrated than that observed in Figure 5-7, again indicating the concentration effect is far less severe in real data than synthetic examples due to the presence of band-to-band correlation and cluster structure (Durrant and Kabán, 2009; Kabán, 2011; Zimek et al., 2012).

The concentration phenomenon is not new to the spectral community, as it was discussed in the hyperspectral remote sensing literature by Jimenez and Landgrebe (1998) during development of a supervised classification method based on projection pursuit.

Concentration of measure has received little attention since their initial description. GSD's have decreased, and extrinsic dimensions of HSI data has increased over the past twenty years, hence the reason for a new study using a larger set of diverse imagery.

Notably, Beyer et al. (1999) also examined the concentration effect in hyperspectral imagery, although not noted as such. They found that hyperspectral data does concentrate such that the expected dynamic range is much smaller than anticipated, but does not approach that of *i.i.d.* data. Their result is supported by our experimentation. Beyer also noted that supplementing the feature (spectral) vector with a large number of additional features such as shape metrics actually decreased contrast between objects, contrary to the intent of the added features—an effect that should be studied in all high-dimensional datasets when spatial, spectral, and shape measures are possibly concatenated into large feature vectors.

5.4 Hubness in Graphs of HSI Data

We now explore hubness in graphs of HSI data and relate it to the distance concentration effect using the Table 5-1 datasets. There have been no studies on the hubness of HSI k -NN graphs using fixed or adaptive neighborhoods to our knowledge; however, hubness has been exploited to estimate the number of endmembers present in HSI data (Heylen et al., 2017). To quantify hubness in each image, we set $k = 30$ and calculate the skewness of the resulting 30-occurrence histogram using (5.3). We do this for both the original and intraband-permuted data to explore the impacts of data structure—interband correlation and natural clustering—on hubness.

Table 5-2 (left) displays the extrinsic dimension ($edim$), DBI, SC, intrinsic dimension ($idim$)¹⁴, hubness measure h , and the number of hubs ($nhubs$) of the original data. The DBI and SC cluster validity measures will be used to associate various quantities with cluster compactness and separability, and are described later in this section. Note the table is sorted by intrinsic dimension of the original, correlation- and structure-rich data to facilitate inspection of changes in hubness with intrinsic dimension.

Initial inspection of $idim$ reveals that HSI data exhibits low inherent dimensionality, often an order of magnitude, or more, below its $edim$ (Schlamm et al., 2010). The hubness measure is also low for these data, indicating a relatively lower potential of hubs for real data compared to the *i.i.d.* datasets of Figure 5-5 or the permuted HSI data. We also see only a few datasets exhibit hubs at $O_k > 5k_{avg}$; note that any row with $nhubs = 0$ have the “0” removed for ease of reading.

¹⁴ Intrinsic dimension is estimated via the method of Levina and Bickel (2005) to remain consistent with cited publications (Schnitzer et al., 2012; Heylen et al., 2017).

Table 5-2. Hubness and Intrinsic Dimension Estimates

Dataset Shortname	<i>edim</i>	<i>DBI</i>	<i>SC</i>	<i>idim</i>	<i>h</i>	<i>nhubs</i>	Δdim	Δh	$\Delta nhubs$	Δh	$\Delta nhubs$	As-Obtained ID	
Gulf-Ref	316	0.59	0.61	5.9	0.7	1	29.3	10.7	1212	0.1	4	Suwannee_0609-1331_ref.dat	
CookeCity	126	0.70	0.54	7.4	0		39.8	6.1	863	0.1		image_refl.dat	
Reno-Ref	316	0.66	0.52	7.6	0.1		72	17.7	1296	0.1		0913-1248_ref.dat	
Cuprite-Ref	316	0.78	0.43	7.7	0.3		32.2	13.3	1412	0.2		0614-1124_ref.dat	
SalinasA	204	0.58	0.63	7.9	0.4		23.8	3.7	116	-0.1		SalinasA.dat	
MoffetField	203	0.75	0.51	8.2	0.4		56	5.4	1206	0.1		M-f970620t01p02_r03_sc02.a.dat	
Foster2	33	0.74	0.48	8.3	0.6		12.5	1.9	289	0.1		ref_ribeira1bbb_reg1.dat	
Conesus	360	0.65	0.73	8.4	1.7	20	33.3	7.6	995	-0.3	12	0920-1714_pol_ref.dat	
MicrosceneLeaf	80	0.56	0.60	8.6	1.4	10	29.9	5.4	1176	0.1	23	20130215_microscene_18.dat	
Reno-Rad	316	0.62	0.55	8.7	0.3		27.8	11.1	1241	0.1		0913-1248_rad.dat	
Salinas	204	0.55	0.62	8.8	0.3		18.8	6	660	-0.1		Salinas.dat	
DCmall	191	0.58	0.57	8.8	0.1		18.6	7.4	827	0		WashingtonMall.dat	
PaviaU	103	0.66	0.51	8.9	0.6		37.2	8.9	1284	0.1		PaviaU.dat	
Barilla	360	0.68	0.51	9.1	0.2		54.9	14.9	1270	0		0920-1844_pol_ref.dat	
Pavia	102	0.69	0.53	9.5	0.7		35	5	1083	0.1	5	Pavia.dat	
PerduePines	193	0.71	0.47	9.6	0.2		16.4	4.1	522	0		19920612_AVIRIS_IndianPine_Site3.dat	
LunarLake	197	0.73	0.47	9.7	0.4		53.4	21.8	1394	0.1		L-f970623t01p02_r07_sc02.a.dat	
AvonField	360	0.62	0.54	9.7	0.3		48.3	10.3	1106	0		0920-1851_pol_ref.dat	
JasperRidge	204	0.89	0.35	9.8	0.4		52.3	7.8	1151	0		J-f970403t01p02_r03_sc05.a.dat	
KSC	176	1.61	0.91	10	2.1	38	24.3	2.4	624	-0.6	5	KSC.dat	
IPines	200	0.92	0.38	10.1	0.4		13.4	4.7	565	0		Indian_pines.dat	
Beltsville-Ref	316	0.75	0.45	10.1	0		28	11.9	1277	0		0810_2022_ref.dat	
Foster1	33	0.82	0.40	10.6	0.7		9.3	1.5	262	0		ref_braga1bb_reg1.dat	
Cuprite	192	0.93	0.33	11	0.5		51.5	16.9	1484	0.2		C-f970619t01p02_r02_sc04.b.dat	
Albuquerque	189	0.74	0.41	11.3	1	1	10.8	11.1	1164	0	3	EO1H0330362001292110PF.dat	
MidWest	189	0.76	0.37	11.3	1.1	1	17.2	8.1	814	0.1	6	EO1H0240342001268110KP.dat	
Cuprite-Rad	316	0.78	0.43	11.5	0.5		31	8.3	1263	0.1		0614-1124_rad.dat	
uArray	80	0.75	0.46	12.3	0.9	1	25.5	7.1	1251	0.1	6	20130215_microarray_3.dat	
Microscene13	80	0.81	0.37	12.3	1.3	4	27.3	5	995	0.1	13	20130215_microscene_13.dat	
Tuscon	189	0.73	0.40	12.5	0.3		8.5	7.4	630	0.1		EO1H0360382001121111PP.dat	
NorthLA	189	0.79	0.39	13.1	0.5		11.7	7.3	793	0		EO1H0410362001236112P2.dat	
Sacramento	189	0.86	0.37	13.1	0.4		13	6.2	804	0		EO1H0440332002077110KY.dat	
Beltsville-Rad	316	0.77	0.43	13.4	0.2		14.7	8	1144	0.1		0810_2022_rad.dat	
Tuscon2	189	0.70	0.42	13.6	0.6		11.1	7.2	1216	0		EO1H0360382001265111P0.dat	
Vegas	189	0.73	0.46	13.7	0.7		11.1	7.5	1072	0	1	EO1H0390352001222112P0.dat	
Rochester	189	0.81	0.51	13.7	1.9	38	12.4	1	135	0.3	43	EO1H0160302001237111PP.dat	
Rockville	189	0.75	0.37	13.8	0.7		12.5	8.1	1115	0		EO1H0150332002114111KZ.dat	
Rochester2	189	0.74	0.54	13.9	2.1	36	9.8	1.4	193	0.3	24	EO1H0170302001260112KP.dat	
Phoenix	189	0.76	0.40	14.6	0.8	1	11.5	11.8	1139	0	3	EO1H0370372001304111P2.dat	
HyperDesert	189	0.94	0.29	14.6	1.6	25	16.5	9.2	1047	0.2	46	EO1H0330372008364110KBa.dat	
BlueDotMS	128	1.02	0.31	14.9	1.4	12	42.6	4.3	888	0.2	41	I200_L0-511_11-8-2014_10.25.57.dat	
Gulf-Rad	316	0.82	0.46	15.3	0.6		16.8	9	1215	0.1		Suwannee_0609-1331_rad.dat	
Botswana	145	0.84	0.36	15.8	0.5		15.1	9.6	760	0		Botswana.dat	
ColumbiaRiver	189	0.82	0.41	16.4	1.7	23	12.9	0.1	24	0.1	39	EO1H0470282001221110KP.dat	
Syracuse	189	0.87	0.36	16.5	1	2	15.6	4.6	939	0.1	10	EO1H0150302001246110PP.dat	
RedSea-Rad	128	0.83	0.39	18.2	1.7	30	22.8	3.5	874	0	18	0509-0440_rad.dat	
Fairbanks	189	0.79	0.41	19.8	0.5		19.2	3.7	422	0		EO1H0690142002090110KZ.dat	
Deepwater-Rad2	316	0.86	0.27	30.1	1.4	18	12	9.3	884	0.1	31	0612-1615_rad_sub.dat	
Deepwater-Rad	316	0.86	0.27	30.1	1.4	18	11.8	8.4	896	0.1	31	0612-1615_rad_sub.dat	
	Original Data ($k=30$)							Permuted Data Δs			Adaptive Δs		

Extrinsic dimension (*edim*), Davies-Bouldin Index (*DBI*), Silhouette Coefficient (*SC*), intrinsic dimension (*idim*), hubness (*h*), and the number of hubs (*nhubs*) for the original HSI data using $k=30$ to left and sides of the table (sorted by intrinsic dimension). Corresponding differential permuted data measures (center), where the differential is taken as $\Delta metric = permuted - original$, such that positive values indicate permuted data exhibits more of each quantity. Note permuted differentials and hubness have been rounded for ease of reading, without loss of generality. The differential hubness (Δh) and hub count differences between the fixed k and ANN reveals that ANN produce more hubs than its fixed k analog. Hub count differences of zero have been removed for ease of reading. As-obtained dataset IDs are provided in the rightmost column.

Permuted data results are shown as differential metrics—hubness (Δh), inherent dimension ($\Delta idim$), and number of hubs ($\Delta nhubs$)—to enable easy inspection of the impacts of permuting intraband pixel locations (Table 5-2, center). All differential metrics are positive, where the differential is taken such that positive values indicate data with less structure displays stronger distance concentration and more hubs. Recall that permuting pixel locations within each band lowers the interband correlation, weakens the community structure of the data, and increases the intrinsic dimension. These conditions should all increase distance concentration (Durrant and Kabán, 2009, Francois et al., 2007; Kabán, 2010), facilitating the formation of hubs. One can easily notice that the hubness, intrinsic dimension, and the number of hubs are much higher for the permuted data than its more correlated and clustered counterpart. Some permuted datasets have upwards of 5% of their data as hubs.

Upon examination of the original and permuted data as a whole, the Spearman rank correlation¹⁵ between intrinsic dimension and hubness measure is 0.85, largely due to all permuted datasets having higher hubness and intrinsic dimensions than their original counterparts. These results indicate data structure may be a causative factor for lower intrinsic dimension and hubness, a similar result to that shown in other studies (Durrant and Kabán, 2009; Heylen et al., 2017). However, taking original and permuted datasets individually, we have much weaker Spearman rank correlations of 0.51 and 0.39 respectively, indicating maintenance of the marginal band-distributions (via permutation)

¹⁵ The Spearman rank correlation is used to account for the nonlinear relationship between hubness and intrinsic dimension (Figure 5-5).

does not guarantee the same characteristic measures. Note that the lower intrinsic dimension in structured high-dimensional data suggests the data lie along a manifold within \mathbb{R}^d (Francois et al., 2007), hence the success of data reduction techniques in HSI processing.

Two cluster validation measures, DBI and SC (Figure 3-3), are used to estimate the cluster separability in the Table 5-1 data. Both DBI and SC ratio a measure of cluster dispersion to cluster separation. Low and high values indicate better separability in DBI and SC respectively (Tan, Steinbach, and Kumar, 2006).

The previous discussion underscores importance of cluster structure and correlation in reducing distance concentration and subsequent hub formation.

So far, we have established that hubness in real data is not as severe as that observed in synthetic data generated with independent bands (Figure 5-5) or even in the same data with permuted pixel locations within each band. HSI data are often highly correlated; hence the intrinsic dimension is typically much lower than the extrinsic dimension (Schott, 2007; Schlamm et al., 2010). The small number of hubs in Table 5-2 suggests there is little risk regarding this phenomenon—at least for classification of these data by k -NN or similar counting techniques. However, some datasets have upwards of 38 hubs, and we will see this condition is exacerbated in adaptive k -NN graphs. As such, impacts of these hubs cannot be fully explored without data having a larger fraction of labels covering scenes displaying a higher number of hubs.

Several classification methods, e.g., normalized cuts, as well as anomaly and target detection algorithms rely on edge weights and subsequent geodesic distances along

manifolds. These algorithms must also be tested to appreciate the breadth of hub impacts to HSI analysis techniques, so more studies of this nature should be conducted as the use of graph- and manifold-based methods proliferate.

Some graph generation techniques produce disconnected graphs, hence can be connected by supplementation with a MST. These spanning trees may also be impacted by edges from cluster means to other classes if hub edges are included in the MST.

The Waikato Environment for Knowledge Analysis (WEKA) (Witten, Frank, and Hall, 2016) is used to explore relationships between the following 11 variables: hubness, intrinsic dimension, extrinsic dimension, GSD, altitude, data type, sensor, SC, DBI, concentration rank, and the number of hubs. Each of these 11 variables is used in its continuous form as well as its ordinal rank so that nonlinear correlations can be investigated. Two Weka functions are used to assess these relationships: the *CorrelationAttributeEval* and a *J48 decision tree* with 10-fold cross validation. These methods require nominal attributes, so continuous values are first binned using the *Discretize* function where the group (bin) number is the class identifier. Observations in these analyses are presented in the bulleted points to follow.

- HSI data containing compact, separated clusters is less concentrated.
- Lower altitude (smaller GSD) data is less concentrated than its larger GSD counterparts. This is attributable to less mixing with smaller pixels, so the data are more clustered (fewer mixing trends).
- Radiance imagery has a higher intrinsic dimension and higher number of hubs than reflectance imagery.

- Data with lower cluster structure measures (high DBI and low SC) display more hubs. However, hubs are present virtually over the entire DBI and SC range.
- The number of hubs has a strong dependence on the hubness measure (trivial) and intrinsic-dimension rank (using ranks accounts for potential nonlinearities not captured in the linear correlation coefficient). The dependence on intrinsic dimension can be observed in Table 5-2 by the fact there are more datasets with hubs towards higher intrinsic dimensions, albeit there are some datasets with higher hubness at lower intrinsic dimensions.
- Inherent dimension is not dependent on the number of bands (extrinsic dimension), and is more correlated with scene types and GSD.
- SC and DBI are inversely, nonlinearly proportional to each other (expected). This provides confidence both metrics are measuring similar phenomena, but does demonstrate they measure the phenomenon differently.

These relationships are aligned with the previous theoretical discussions, and supported by results found in other disciplines (Francois et al., 2007; Durrant and Kabán, 2009; Radovanović et al., 2010a).

Our discussion to this point has been on establishing the presence of hubness in real HSI data using directed fixed k -NN graphs (or variants enforcing the superset criterion). In Chapter 2, we discussed adaptive k -NN variants, wherein each pixel is assigned its own

neighborhood size. We hypothesized that hubness may be a larger problem in density-weighted adaptive graphs because larger neighborhoods are assigned to pixels in denser regions—exactly where hubs reside.

This hypothesis is tested using the Table 5-1 datasets by investigating the presence of hubness at a threshold of $5k_{avg}$, noting that $k_{avg} = \frac{1}{n} \sum_i k(i)$ for adaptive algorithms, where $n = |V|$ and k_{avg} varies by dataset. We construct directed fixed and adaptive graphs of the Table 5-1 data using approximately the same number of edges, i.e., $k = \lceil \mathbf{k}i_{avg} \rceil$, where the ceiling is taken to provide a conservative estimate. Differential hubness (Δh) and hub count differences ($\Delta nhubs$) are shown to the right of Table 5-2, where the differential is taken such that positive values indicate higher measures for the adaptive techniques. We can see that hubness is stronger, and correspondingly the number of hubs are higher, in graphs built from adaptive techniques as indicated by positive $\Delta nhub$ values. Hubs may therefore be a larger concern in adaptive k -NN graphs than in their fixed k counterparts for the same number of edges, and even more of a concern for situations when $\sum_i k(i) > nk$, where k is a typical number of fixed k -NN.

Also note that edge allocation by adaptive methods generally increased the number of hubs in datasets that already displayed hubness in their original directed graphs; i.e., hubs were only introduced into two of the remaining datasets (*Vegas* and *Pavia*), and in small quantities. Several authors state that the distribution of the data itself must also be considered when discussing concentration effects (Durrant and Kabán, 2009; Francois et al., 2007; Kabán, 2011), indirectly supporting this observation, and providing additional

proof to the claim data structure and intrinsic dimension contribute to concentration, which facilitates the creation of hubs.

Thus far, we have established that hubs exist in fixed, and to a greater degree, adaptive nearest-neighbor graphs of HSI data. However, the nature of these hubs is still unknown; i.e., are they good hubs that promote strong intracluster connectivity near cluster centers or bad hubs that have intercluster edges, corrupting geodesic distances calculated on manifolds (Francois et al., 2007). Radovanović et al. (2010b) mention that high intrinsic dimension and hubness alone are not sufficient to introduce bad hubs, but rather cluster assumption¹⁶ violations are also required for bad hubs to appear, i.e., natural cluster composition of two or more classes. Natural spectral variability and non-distinct clusters, resulting from spectral mixing, present an opportunity for this to occur. We wish to therefore understand the nature of HSI hubs given these very common characteristics of HSI data¹⁷.

Hub classification (good or bad) can be made by simply comparing the class label of each hub vertex to those in its reverse nearest neighbor list. We characterize the percentage of good hubs as $1.0 - \phi$ -edge ratio, where the ϕ -edge ratio is defined as the total number of edges between vertices with different class labels normalized by the number of edges in the reverse nearest neighbor list (Ozaki et al., 2011). Higher scores are thus indicative of hubs more connected to pixels of the same class (good hubs).

¹⁶ Pairwise edges in dense regions should be from vertices of the same class.

¹⁷ From a practical standpoint, incorrect class labels can also produce the appearance of bad hubs.

Seven of the datasets listed in Table 5-1 have associated class labels. Of them, only *Microscene13* and *BlueDotMS* produce hubs for both fixed and adaptive k -NN graphs by our current definition, $O^k \geq 5k_{avg}$. As such, we relax the criterion to $O^k \geq 4k_{avg}$, to increase the number of hubs for both methods, resulting in hub appearance in the *Pavia* dataset. Even with this relaxed hub definition, the *Salinas A*, *Salinas*, *PaviaU*, and *PurdueIPines* datasets do not exhibit any hubs.

The number of hubs for both the fixed k and ANN graphs of *Pavia*, *BlueDotMS*, and *Microscene13* are provided in Table 5-3, where we observe the ANN algorithm has generated more hubs due to the nature of its construction. Table 5-3 also displays the overall goodness measure (good), as well as the percent (%NN) of the reverse nearest neighbors used in the assessment (i.e., the percentage of nearest neighbors with ground truth) for each construction method. Although hubs are attributable to dense clusters, interclass (short circuiting) edges can still arise as indicated by goodness scores < 1.0 . Fixed and adaptive techniques do produce more good hubs than bad in this limited test, which is expected. While not exhaustive (only seven of 49 sets have ground truth, of which only three displayed hubness), this does provide some insight into hub character in HSI data. It appears most hubs are good, promoting string intracluster connectivity; however, short circuiting can occur. The impact of these short circuits will be explored in a future study exploring manifold distances through these graphs.

Table 5-3. Hub Characterization (Good vs. Bad)

Dataset	$4k_{avg}$ <i>nhubs</i>	<i>good</i>	%NN	ANN <i>nhubs</i>	<i>good</i>	%NN
<i>Pavia</i>	7	1.000	96.6	32	1.000	97.6
<i>BlueDotMS</i>	84	0.913	98.7	222	0.908	99.1
<i>Microscene13</i>	30	0.999	94.6	82	0.999	94.7

Fixed k -NN and ANN hub characterization to the left and right respectively. The number of hubs, overall hub goodness (*good*), and the percentage of the nearest neighbor lists that could be evaluated (%NN) are reported for each graph type. We see that some hubs do produce edges that cross cluster boundaries ($good < 1.0$).

Table 5-3 demonstrates that some hubs in graphs of HSI data produce upwards of 10% of their edges crossing class boundaries. To determine if these edges around hubs are responsible for more semantic errors than edges around normal vertices, we rerun the same analysis, but for non-hub vertices. There were mixed results for this small dataset, where normal vertices produced more and fewer semantic errors than hubs by 2-4.5%. Semantic errors from hubs near cluster centers may be more impactful to manifold techniques than semantic errors on class peripheries; hence this requires additional study on more datasets with class labels. Should hubs impact distances along manifolds, similarity methods less prone to hubness such as mutual proximity, localized scaling, and shared nearest neighbors should be investigated.

5.5 Summary and Contributions

Many HSI graph construction techniques are founded on the k -NN relationship because of its local adaptability to data scale, shape, and density. k -NN relationships can be impacted by hubness wherein some nodes influence a relatively large number of k -NN lists and potentially decrease the utility of the derived graphical construct. We presented

the first known study of hub presence and characterization in HSI graphs constructed from fixed and adaptive neighborhood-construction methods found in the hyperspectral remote sensing literature. The studies indicate that the hubness phenomenon does exist in graphs of HSI data (more so in adaptively built graphs), but not as strongly as that found in *i.i.d.* data of similar dimension. This fact is attributed to the relatively low inherent dimensionality of HSI data compared to its extrinsic dimension as well as cluster structure in HSI data. From a k -NN classification perspective, the presence of hubs did not negatively impact classification results (counting tasks), but expanded analyses are required to explore the hub impacts on analysis methods utilizing edge weights, e.g., spanning trees, target detection, spectral methods for clustering, segmentation, and classification. As this is a new area of study, much work remains to fully explore the implications on hyperspectral analyses using datasets with a fuller complement of class labels, scene diversity, and sensor characteristics.

6

GRAPH-BASED HSI ANALYSIS ALGORITHMS

Several mechanisms produce nonlinearities in hyperspectral data, e.g., transmissive media such as water (Bachmann et al., 2005; Gillis et al., 2005) and intimate mixing found in many fine soils and sands (Keshava, 2003). Additionally, higher spatial and spectral resolution systems produce complex clutter that may not be well modeled by traditional statistical and linear methods (Ziemann and Messinger, 2014a). As such, investigators are exploring non-linear methods to better represent the intrinsic structure of HSI data with the intent of producing more accurate analyses. This chapter provides an overview of the remote sensing literature from 2004-2017 to touch on the breadth of HSI graph- and manifold-based research and development.

6.1 Segmentation, Clustering, and Classification

Segmentation, clustering, and classification are similar analyses wherein imagery is divided into components. The main difference between them is that classification assigns a label to a group while clustering and segmentation do not (Schott, 2007). Clusters may not be spatially connected, whereas segmentation results are spatially connected. The bulk of the surveyed graph- or manifold-based literature prior to 2012 is focused on these types of analyses.

6.1.1 Segmentation and Clustering

Mohan, Sapiro, and Bosch (2006) present a spatial-spectral technique for the segmentation of spectral imagery based on local linear embedding (LLE). Spatial coherence is introduced into the distance measure used in generating the LLE embedding by replacing the Euclidean distance component differences by those of a 3 x 3 spatial patch. Superpixel spectral similarities are high if the query pixel and its proximal neighbors (in the spatial window) are high. Mohan et al. report a 15% increase in classification accuracy over using spectral measures alone.

A divisive hierarchical clustering technique based on maximization of the graph modularity is described by Mercovich (2011). Graph modularity is a measure of the number of edges within a group of nodes to that from a random distribution. The larger the number, the more community structure present, i.e., maximizing this measure produces purer clusters. An adjacency graph of unitary weights is constructed using the method of Chapter 2.7.1. The maximum modularity of the graph is recursively examined and a subgraph split is performed if the modularity exceeds a threshold. This continues through each leaf in the tree with no restriction on the number of pixels in any node at each level. Following each split, a subgraphs are rebuilt and the process is repeated until a stopping criterion is met (either minimum cluster size or modularity threshold). Upon completion, the method produces a hierarchical cluster map where each level has varying levels of detail, which was proven to provide better separation of materials than k -means, especially when the number of classes is not known a priori. Mercovich notes the method is similar to normalized cuts, but uses the eigenvector corresponding to the maximum eigenvalue as

opposed to that corresponding to the minimum eigenvalue, enabling faster processing of large imagery.

Albano, Messinger, and Rotman (2012) describe the application of the commute time distance (CTD) transformation as a means to embed spectral data into a new feature space prior to traditional spectral analysis. The CTD feature space is based on a Markov model of a random walk on the graph, quantified by the average distance traveled from one vertex to another and back. Distance between two nodes in the CTD space is small if there are many paths between nodes or the weights between them is small, i.e., it does not take long to get from one node to the other (useful for clustering). Conversely, the distance between nodes in CTD space is large if there are few paths between nodes or the weights between them are large (useful for anomaly and rare target detection). Albano notes that unlike principal component imagery, CTD feature space images are correlated to physical materials. Additionally, the method separates large or small groupings of materials equally well in contrast to PCA where a material must be present in large enough quantities to impact the covariance matrix. Testing demonstrated some improvement in clustering results when compared to k -means clustering.

Gillis and Bowles (2012) describe a segmentation technique based on a modified normalized cuts algorithm of Shi and Malik (2000). A spatial-spectral graph is constructed (2.6) and normalized cuts recursively partitions the graph into segments. The main benefit of the technique was the novel encoding of spatial and spectral information into edges only connecting pixels within and $r \times r$ window centered on each pixel. The resultant adjacency matrix is very sparse, resulting in quicker solutions for larger image. Edge encoding into

only spatially proximal pixels results in potentially segmented clusters of the same material that may need to be merged afterwards.

Fan and Messinger (2014) describe a hyperspectral image segmentation technique based on a split-merge method using normalized cuts with a locally adaptive spatial-spectral graph (2.7). The overly segmented graph is passes through a segment merging step to produce the final segmentation. The method outperformed the agglomerative, *k*-means, and mixture model clustering on the two images tested.

6.1.2 Classification

Bachmann, Ainsworth, and Fusina (2005) describe a nonlinear method to classify wetland imagery using manifold coordinates derived by the isometric mapping (ISOMAP) algorithm that preserves geodesic distances in spectral space. Their move towards nonlinear methods was driven by the strong nonlinearities caused by varying fractions and depths of water in wetland imagery. Gilles et al. (2005) describes the mechanics for generating these nonlinear trends, where changing water depths trace out a curve in spectral space for fixed water quality (biomass and sedimentary load) and bottom types. Differing water quality and bottom conditions simply trace out different curves. Bachmann proves the data have a more compact representation in manifold coordinates than that of the traditional linear minimum noise fraction (MNF) as well as demonstrating enhanced separability of spectrally similar classes.

Bachmann et al. (2005) offers two solutions to address the high ISOMAP computational-complexity: 1) tiling the image and stitching the resultant sub manifolds together into an aligned global coordinate system (divide, conquer, and merge strategy)

and 2) registration to a global subsampled backbone manifold. Computational and memory scaling, better sub manifold alignment across tiles and artifact reduction are later provided (Bachmann, Ainsworth, and Fusina, 2006). High performance computing solutions using commercial general purpose graphical processing unit (GPGPU) hardware are also investigated to reduce execution times into the single minutes range for scene sizes $O(10^6)$ (Topping, 2009).

Chen, Crawford, and Ghosh (2005) implement a classification strategy called shortest path k -NN (Sk -NN) classifier which also use an ISOMAP-derived manifold embedding followed by the k -NN classification. They demonstrate Sk -NN is competitive with the more advanced best-basis hierarchical classifier and the hierarchical support vector machine (SVM) classifiers; all tested methods are better than k -NN in spectral space.

Motivated by Bachmann's success, He et al. (2009) derived manifold coordinate descriptions based on diffusion geometry. Spectrally decomposing the affinity matrix produces a set of diffusion maps (feature space bands) which can be rearranged to generate diffusion coefficients, embedding the nonlinear spectral data into a Euclidean diffusion space. Euclidean distance in this space corresponds to the diffusion distance which tracks with the data manifold. The authors employ a backbone-based alignment process similar to that described in Bachmann et al. (2005) to provide consistent scene-wide diffusion coordinates. Testing on several datasets showed better material separability than those based on a simple principal components analysis (PCA). Figure 6-1 shows a full scene diffusion map composite product (614x512) merged with backbone alignment processing

generated in 12 minutes (1.5 minutes backbone generation, 11 minutes for the full scene reconstruction).



Figure 6-1. Diffusion geometry coordinates (bands 5-6-7) for the Moffet field scene as shown in He et al. (2009).

He notes that diffusion geometry may be more scalable to full size imagery than ISOMAP.

Benedetto, et al. (2012b) describe three methods to include spatial information in Laplacian Eigenmaps for classification analysis. The graph Laplacian (Chapter 7) is a matrix on the nodes defined by $\mathbf{L} = \mathbf{D} - \mathbf{W}$, where \mathbf{D} is a diagonal matrix composed of node degrees, i.e., $[\mathbf{D}]_{ii} = \sum_j w_{ij}$, and \mathbf{W} is the weighted adjacency matrix. Solving for the eigenvectors of \mathbf{L} produces a set of manifold coordinates that are treated as a new feature space. Spatial information can be used in the construction of \mathbf{L} , and Benedetto et al. offers three variants.

The first is Eigen coordinate concatenation.

- Construct pure mutual k -NN spectral and pure mutual k -NN spatial graphs based on (2.9), i.e., $\beta = 1$ and $\beta = 0$ respectively, followed by heat kernel reweighting (2.4) with $\sigma = 1$. Note, all edges locations are identified by the spectral neighbors.
- Compute the Eigenmaps of each graph and simply concatenate Eigenmap coordinates into a supervector in manifold space, \mathbb{R}^{p+s} , where p and s are dimensions of the spectral and spatial manifold embeddings respectively.

The second method produces a fused single Laplacian created by elementwise multiplication or addition of the spatial and spectral Laplacians, as well as matrix multiplication trimmed to take on the same sparsity as the spectral weighted adjacency matrix. Subsequent determination of the Laplacian eigenvectors produces the manifold embedding.

The third method simply sets the β parameter in (2.9) for the distance measure, followed by subsequent heat kernel weighting and Laplacian generation.

All three methods provide superior results over classifications based on graphical constructs composed from spatial or spectral information alone. All top performing methods utilize spatial information in the graph construction as well as the spatial-spectral Laplacian.

Benedetto et al. (2012a) discuss the use of cluster potentials in Schroedinger Eigenmaps (Chapter 7) to improve classification accuracy. Cluster potentials provide additional information during the eigendecomposition of the Laplacian matrix in the form

of a small number of pixels that should remain close to each other in the embedding. This additional information was provided from a k means clustering on the data to indicate which pixels should be linked in the potential matrix. The authors demonstrate the use of cluster potentials improves results over standard Laplacian Eigenmap clustering. Note the method does have a sensitivity to the user-defined the number of clusters in that it controls the number of groupings employed in generating the cluster potential; this is a common problem for any algorithm requiring similar input.

Hou, Zhang, Ye, and Zheng (2013) propose a classification method based on Laplacian Eigenmaps using (2.8) called the Laplacian Eigenmap Pixel Distribution-Flow (LE PD-Flow). Laplacian Eigenmaps are created over a permuted set of eigensolutions to Laplacian matrices constructed over a set of spectral scales (σ) using a supervector comprised of spatial and spectral feature measures, and the rate of change (trend) is tracked across each spectral scale and used to influence the classification. They report better results than PCA and ISOMAP feature mapping followed by k -means classification, but the repeated eigendecomposition produces much longer run times than other methods. Hou et al. note this method is most appropriate for imagery with large spatially-homogeneous regions.

Spatial-spectral Schroedinger Eigenmaps (SSSE) are described by Cahill, Czaja, and Messinger (2014), wherein the Laplacian matrix is built upon spectral similarity alone, while the Schroedinger potential matrix (see Chapter 7) encodes spatial proximity as well as spectral similarity. This encoding not only forces spatially proximal pixels together, but also provides even stronger attraction between pixels that are spatially close and spectrally

similar. Unlike Benedetto (2012a), spatial relationships are simply encoded based on pixel location in the image and are not the result of a clustering algorithm, basically relying on the spatial correlation present in imagery. An advantage of this construct is the separation of spatial and spectral weights, where unlike the many methods in Chapter 2.7.2 that encode spatial relationships directly into the weight matrix, SSSE keeps them entirely separate, only using spatial information in the Eigenmap solution (Chapter 2.7.3). This facilitates much faster parametric studies on spatial effects and permits an arbitrarily small spatial scale without disconnecting spectrally similar neighbors at larger spatial distances. Classification results shown are the highest of any reviewed method.

Cahill, Chew, and Wenger (2015) demonstrate how to modify SSSE such that expert knowledge can be included in the Eigenmap solution for the benefit of spatial-spectral dimensionality reduction and classification. Given a small set of expert-labeled pixels, a matrix similar to the spatial cluster potential is created and added to the spatial-spectral cluster potential to further enforce that some pixels (the expert-labeled pixels) must be close together in the lower dimensional representation from the Eigenmap solution. This is akin to Benedetto using a clustering result to define cluster potential pixels, but only using a very small subset of the closest matches. Cahill et al. demonstrate the inclusion of a small number of expert labeled points can correct misclassification errors.

6.2 Anomaly Detection

Basener, Ientilucci, and Messinger (2007) describe the first HSI graph-based anomaly detection algorithm called the Topological Anomaly Detector (TAD). TAD first creates a graph from 10% of the shortest pairwise L_2 distances to connect the most similar

pixels. This construction connects the densest regions in spectral space which represents background (abundant) classes. The background is then defined as the set of connected components of size $n > 0.02|V|$. The degree of pixel “anomalouslyness” is the distance to any of these larger background graph elements. Therefore, isocontours in anomalousness are not bound to a smooth form as in statistical methods, but are rather data-driven and can take on arbitrary shapes as shown in Figure 6-2 for TAD and the RX algorithm (MVN assumption). Anomalies can be detected within pockets of low density surrounded by the background of the data as seen in the left side of Figure 6-2.

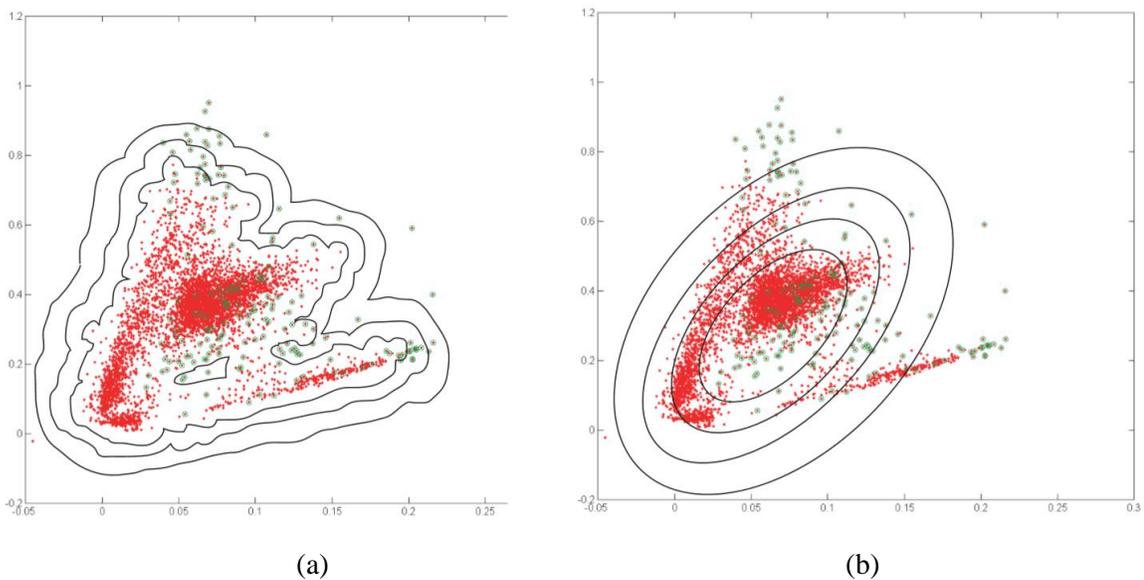


Figure 6-2. Isocontours of anomalousness for (a) TAD and (b) RX (Basener and Messinger, 2007).

TAD performed well when compared to RX, and was shown to be less sensitive to noise (Basener and Messinger, 2009). TAD can also be used to mask out anomalous pixels prior to covariance and subspace calculations for standard target detection algorithms akin

to RX filtering prior to covariance estimation. TAD prescreening resulted in a robust improvement in quadratic target detection algorithms (Basener, 2010).

Messinger and Albano (2011) describe an anomaly detection algorithm based on nodal connectivity characteristics measured by the weighted vertex volume (WVV), a ratio of the number of edges to the node (node degree) to the sum of its edge weights (6.1).

$$WVV(x_i) = \frac{\text{deg}(v_i)}{\sum_{v_j \in N(v_i)} w_{ij}}, \quad \forall v_j \in V \quad (6.1)$$

Anomalies are characterized by poor connectivity (exhibit low node degree) and are far from their nearest neighbors (large weights to their neighboring pixels), producing a low WVV. The inverse of the WVV is used as the anomalousness metric so anomalies appear bright in the detection image. Messinger and Albano note that the method is comparable if not better than RX, especially in high resolution scenes that display complex clutter not well modeled by statistical approaches.

Albano and Messinger (2012) explore the use of Euclidean CTD to enhance the separability of anomalies from background. The CTD transform (a preprocessing step) pushes outliers away from the bulk distribution (background) when fewer paths and/or larger weights between nodes exists, making anomalies more anomalous; anomalies are therefore easier to detect using standard algorithms (Figure 6-3).

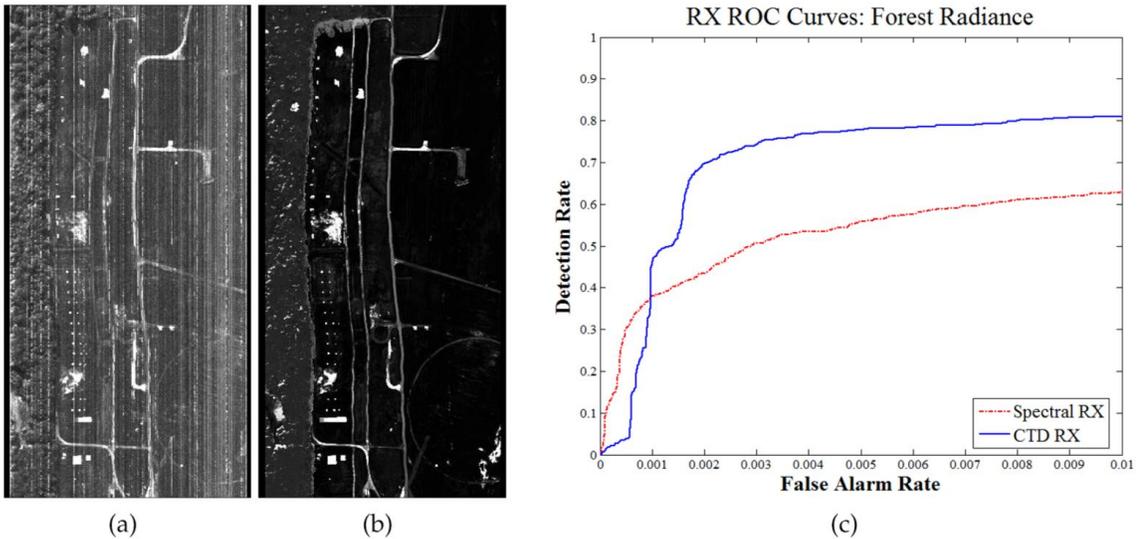


Figure 6-3. Anomaly detection in (a) spectral space, (b) CTD space, and (c) ROC analysis (Albano and Messinger, 2012).

Notice the lack of striping apparent in the RX algorithm. Statistical techniques are susceptible to structured noise such as striping (Mohan et al., 2006), a sensitivity not observed in the CTD method. We will see this same desirable characteristic present in Schroedinger Eigenmaps as well (Chapter 7).

6.3 Target Detection

The application of graph theory and manifolds to target detection is a relatively unexplored area (Ziemann et al., 2014a). This section reviews the few papers found investigating this topic.

Dorado-Munoz, Messinger, and Ziemann (2013) examine the performance of traditional target detection algorithms using background pixels identified by TAD. Specifically, TAD-generated background connected-component means and/or endmembers extracted from each component are used in traditional geometric methods,

while means and covariances estimated from the same components (or all component pixels combined) are used in traditional statistical methods. Investigators found that statistical approaches performed better using TAD-derived backgrounds, indicating the TAD background model has potential use in target detection. The ability to model nonlinear behavior using this nonparametric data-driven model is lost by resorting to methods that characterize their input data by parametric statistical models or subspaces, so the benefit is solely from identification of more representative background pixels from which to derive statistical and subspace models.

Albano, Messinger, and Ientilucci (2013) explore the use of CTD and physics-based models for target detection, wherein a large number of radiative transfer calculations are performed to generate representative spectra under varying acquisition conditions, e.g., time of day, water vapor levels, and acquisition geometries, a process similar to invariant methods (Healy and Slater, 1999). These calculated spectra are appended (injected) into the image prior to spectral graph creation via a mutual k -NN with MST union (to ensure graph connectivity). The location of the injected target spectra are known prior to the CTD transform, so it's easy to generate a mean target vector in CTD feature space. The background and target means and background covariance matrix (in CTD space) are employed in the standard spectral matched filter (SMF). The authors compare results with the subspace Adaptive Cosine Estimator (ACE) algorithm and show favorable results at all false alarm levels (Figure 6-4).

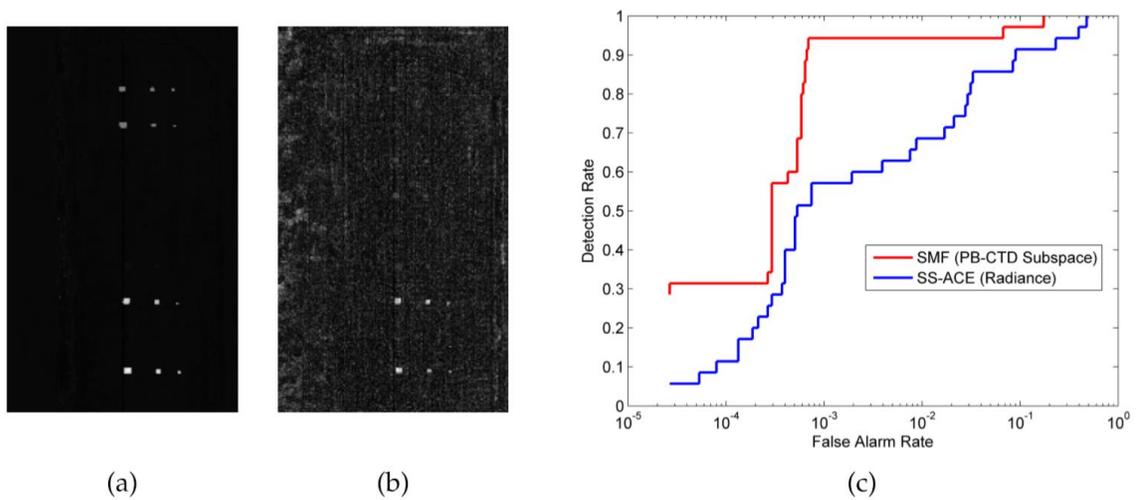


Figure 6-4. Detection statistic image for (a) the SMF in CTD space, (b) subspace ACE in spectral space, and (c) ROC analysis (Albano et al., 2013).

Manifold coordinates derived via LLE are applied to target detection in Ziemann, Messinger, and Albano (2013), where a user provides the intended dimensionality (p) of the embedding subspace to limit the number of recover manifold coordinates. The image is then sectioned into tiles of about 25-40 pixels on a side due to LLE's high computational burden and the target spectrum is appended (injected). A k -NN graph is created with NNN to produce a spectral-density weighted graph for each tile. The p dimensional manifold is recovered using LLE with a constrained least squares solution, and target detection is performed with ACE using the manifold coordinates of the injected target spectrum. Some additional target detections over those of ACE are noted, providing compelling results with few assumptions on the data. However, Ziemann mentions the algorithm does have a high false alarm rate if the target is not present in the scene and suggests a backbone approach such as that as used in Bachman et al. (2005) as a potential solution to globally align the target manifolds.

Ziemann and Messinger (2015) extend the LLE target detection methodology by adding a “cloud” of target vectors to each subimages similar to that of Albano et al. (2013). The uniformly-distributed cloud is 10% the size of the image with a diameter equal to the average distances to the first five scene-based NN to the target, and overconnected to their scene-based neighbors to ensure better separation of background and target pixels. All target-cloud points are removed prior to detection processing, but served to better formulate the LLE by mitigating the possibility of a single target spectrum collapsing into the background distribution. Ziemann notes that image tiling addresses computational complexity and does not impact target detection because performance should be independent from one region of the image to another, i.e., the presence of grass in one part of the scene should not impact detection over soil regions in another. A spectral angle based detection scheme in manifold space produced better performance than ACE for some targets.

Dorado-Munoz and Messinger (2016) extend the approaches of Albano et al. (2013) and Ziemann and Messinger (2015) to target detection using Schroedinger Eigenmaps. Target spectra are appended to each image and both fixed k -NN (Chapter 2.2) and DW k -NN (Chapter 2.3) graphs are created. Additional edges are constructed between these target material and its neighbors (standard graph construction), and also neighbors or their neighbors, i.e., neighbors that are within two edges of the target material. These additional connections improve the connectivity of the target with pixels that share some similarity, but not enough to comprise the target material’s NN list. These target NN relationships are encoded into a barrier potential matrix used in the Schroedinger Eigenmap

algorithm (see Chapter 7) to drive target (and similar) pixels towards the origin. A simple distance measure from the origin is used as the detections statistic. The SS-based method is compared to ACE on in-scene and field measured signatures. ACE outperformed the method using in-scene signatures; however, the SE-based algorithm was able to outperform ACE using field-measured signatures in some permutations, demonstrating the method’s potential as a new technique not requiring hypothesis testing.

Ziemann and Messinger (2015) and Dorado-Munoz and Messinger (2016) methods have proven the efficacy of spectral embedding for target detection for a single target material via injection into the data prior to graph construction. However, the target injection method does not scale well as a generalized target detection scheme looking for many materials at once. If each material requires a target cloud of size $0.1|V|$, the data size doubles at only ten materials. Conversely, pulling target materials towards the origin would entail a reduction in contrast between the target materials themselves. Even if enough contrast exists, a new anisotropic target detector is required compared to the isotropic variant used in the proof of concept. These challenges need to be further developed to extend these concepts into characterization tasks instead of simply targeted search.

6.4 Change Detection

Albano, Messinger, Schlamm, and Basener (2011) describe a tile-based change detection method based on normalized edge volume (NEV) and the standard deviation of edge lengths (SDEL) to cue analysts to regions requiring further change investigation. NEV is a ratio of graph “spread” to graph “connectedness” (6.2) as

$$NEV(G) = \frac{\sum_{(v_i, v_j) \in E} w_{ij}}{\sum_{v_i \in V} \deg(v_i)}, \quad (6.2)$$

while, the numerator is the sum of all the edge weights in the graph; the higher this number, the more spread in the graph. The denominator is the vertex volume of the graph, i.e., 2x the number of edges because of the undirected (bidirectional) nature of the graph built on the spectral data. High NEV indicates dispersed graphs with few connections, while low NEV indicates compact graphs with a high number of connections (denseness). Albano demonstrates NEV is directly proportional to the spectral diversity in the image, i.e., higher NEV indicates higher spectral diversity, so it can be used as a macro scale measure. Conversely, SDEL is a measure of fine scale change. Both can be used together to indicate change within a graph. For display (cueing) purposes, NEV and SDEL differences between the change detection image pairs is scaled and color coded to represent the degree of change (Figure 6-5).

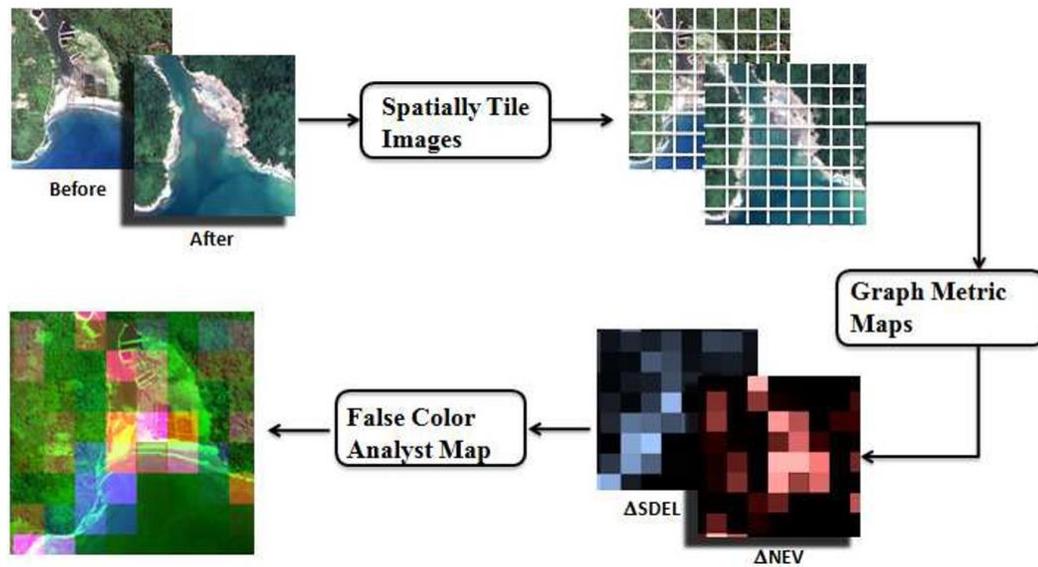


Figure 6-5. Graph-theoretic processing for tile-based HSI change detection (Albano et al., 2011).

Tile-based analysis provides resiliency to image misregistration always present in change detection pairs.

6.5 Physical Unit Determination

Manifold descriptions can also be used for physical unit retrieval. While different than the techniques previously discussed, a few studies are mentioned here for completeness.

Gilles et al. (2005) describe a methodology to retrieve water characteristics, bottom type, and bathymetry from physics-based approaches.

Bachmann et al. (2009) describe a method to perform bathymetric retrieval from manifold coordinates.

7

SCHROEDINGER EIGENMAPS

Additional variants of current state-of-the-art HSI graph processing algorithms can be constructed by leveraging the new similarity measures introduced in Chapter 3 to explore performance improvement. These state-of-the-art techniques are built upon the Laplacian and Schroedinger operators on a graph, which are reviewed below.

Let $V = \{v_1, v_2, \dots, v_n\}$ and $E = \{e_1, e_2, \dots, e_m\}$ represent numbered vertex and directed edge sets respectively. Relationships between V and E can be explicitly represented by an $n \times m$ incidence matrix \mathbf{B} , defined by (7.1).

$$b_{ij} = \begin{cases} +1, & \text{if } v_i = \text{source of } e_j \\ -1, & \text{if } v_i = \text{sink of } e_j \\ 0, & \text{otherwise} \end{cases} \quad (7.1)$$

\mathbf{B} is a simple matrix composed of three values $\{-1, 1, 0\}$ fully describing the source and sink vertices for each edge in the graph. Naturally undirected graphs are binary matrices $\{1, 0\}$.

The graph Laplacian (\mathbf{L}) is defined by (7.2), where we can explicitly see its dependence on graph structure due to its construction from \mathbf{B} . The matrix product $\mathbf{B}\mathbf{B}^T$ produces the sum of absolute node degrees on the diagonal (\mathbf{D}) and the signed weighted combinations on the off diagonals (\mathbf{W}).

$$\mathbf{L} = \mathbf{B}\mathbf{B}^T = \mathbf{D} - \mathbf{W} \quad (7.2)$$

The Laplacian matrix in (7.2) is called the unnormalized graph Laplacian, where each row sums to zero and $\mathbf{1}$ is in the nullspace of \mathbf{L} . In general, the incidence matrix has rank $n-c$, where c denotes the number of connected components, the dimension of the nullspace, and the multiplicity of the zero eigenvalue (Gallier, 2016). As such, the incidence matrix of a connected graph has rank $n-1$, and consequently so does \mathbf{L} .

Normalized variants of the Laplacian are needed for some applications and two options are common: the normalized graph Laplacian (7.3) and the random walk Laplacian (7.4).

$$\mathbf{L}_{sym} = \mathbf{D}^{-\frac{1}{2}}\mathbf{L}\mathbf{D}^{-\frac{1}{2}} = \mathbf{I} - \mathbf{D}^{-\frac{1}{2}}\mathbf{W}\mathbf{D}^{-\frac{1}{2}} \quad (7.3)$$

$$\mathbf{L}_{rw} = \mathbf{D}^{-1}\mathbf{L} = \mathbf{I} - \mathbf{D}^{-1}\mathbf{W} = \mathbf{D}^{-\frac{1}{2}}\mathbf{L}_{sym}\mathbf{D}^{\frac{1}{2}} \quad (7.4)$$

Note that the normalized variants require that \mathbf{D} has full rank, i.e., the graph G has no isolated vertices and every row of \mathbf{W} contains a positive entry. If this is true, then \mathbf{D} is invertible and its square root decomposition exists. Should \mathbf{D} not possess full rank, the normalized variants can be applied to each connected component separately.

We eventually focus on the Schroedinger Eigenmaps of Czaja and Ehler (2012), but first start with the Laplacian Eigenmaps of Belkin and Niyogi (2003) as a stepping stone. Given n data points in \mathbb{R}^d , specified by $\mathbf{v} = \{v_1, v_2, \dots, v_n\}$, we wish to find a lower

dimensional representation of the data $\mathbf{y} = \{y_1, \dots, y_p\}$, where $\mathbf{y} \in \mathbb{R}^p$ and $p \ll d$, i.e., the data are assumed to lie on a manifold $\mathcal{M} \in \mathbb{R}^p$. The manifold embedding \mathbf{y} is determined by the following three steps:

1. Construct a graph on the HSI data using any of the methods described in Chapters 2 or 3. We are free to include spatial information in this construction if so desired (Chapters 2.7.1 and 2.7.2).
2. Calculate the graph Laplacian, \mathbf{L} using (7.2), where $[\mathbf{D}]_{ii} = \sum_j w_{ij} = d(v_i)$ and $[\mathbf{W}]_{ij} = w_{ij}$.
3. Solve for the mapping $\mathbf{y} = \{y_1, \dots, y_p\}$ by minimizing

$$\arg \min_{\mathbf{y}^T \mathbf{D} \mathbf{y} = \mathbf{I}} \frac{1}{2} \sum_{ij} \|y_i - y_j\|^2 w_{ij}. \quad (7.5)$$

Minimizing (7.5) is the same as minimizing

$$\arg \min_{\mathbf{y}^T \mathbf{D} \mathbf{y} = \mathbf{I}} \text{tr}(\mathbf{y}^T \mathbf{L} \mathbf{y}) \quad (7.6)$$

(Belkin and Niyogi, 2003), and the solution is obtained by solving the generalized eigenvalue problem $\mathbf{L} \mathbf{y} = \lambda \mathbf{D} \mathbf{y}$. This particular solution incurs a penalty when proximal points in \mathbb{R}^d are not proximal in \mathbb{R}^p , ensuring the d -dimensional structure is maintained in the (lower) p -dimensional manifold representation.

Practically speaking, eigenvalues are ordered such that $0 = \lambda_0 \leq \lambda_1 \leq \dots \leq \lambda_{n-1}$ and the corresponding n eigenvectors create an orthonormal basis for \mathbf{L} . We use the first p nonzero eigenvectors of \mathbf{L} as a mapping from $\mathbb{R}^d \rightarrow \mathbb{R}^p$. Note that the multiplicity of the zero eigenvalue indicates the number of connected components in the graph (also the nullspace dimension), where the corresponding eigenvector of \mathbf{L} is $\mathbf{1}$. Standard statistical or geometrical processing algorithms may be applied to the data embedded into this lower dimensional space ($25 \leq p \leq 50$ is common), so this embedding can be interpreted as a preprocessing or data conditioning step.

Czaja and Ehler (2012) build upon the success of Laplacian Eigenmaps and create a generalized variant wherein additional information influencing the Eigenmap solution is incorporated. Their variant, called Schroedinger Eigenmaps, utilizes a potential matrix (\mathbf{V}) that is added to the graph Laplacian prior to manifold embedding (7.7).

$$(\mathbf{L} + \alpha\mathbf{V})f = \lambda\mathbf{D}f \quad (7.7)$$

The user-defined parameter, α , controls the relative significance, or more generally the fusion between the Laplacian and the potential matrix. Solving for the Eigenmap is performed by substituting the Schroedinger operator $\mathbf{S} = (\mathbf{L} + \alpha\mathbf{V})$ for \mathbf{L} in (7.6). Two types of potential matrices are described: barrier potentials and cluster potentials.

Barrier potentials result when \mathbf{V} is constructed as a nonnegative diagonal matrix. Positive entries in \mathbf{V} induce a penalty in the mapping, pulling points towards the manifold origin (Benedetto et al., 2012a), while zero entries drive points farther apart. This characteristic is exploited by Dorado-Munoz and Messinger (2016) in developing a target

detection algorithm wherein all target pixels are pulled towards the origin. The manifold-space vector magnitude is used as a detector, where smaller values are more indicative of the target material.

Cluster potentials are generated by the summation of nondiagonal submatrices defined for all points (i,j) deemed to belong to the same class. Let Q be a cluster node set, and $i,j \in Q$ (or generalized grouping). For all points (i,j) taken in order, define the submatrices $V_{k,l}^{(i,j)}$ associated with Q as

$$V_{k,l}^{(i,j)} = \begin{cases} 1, & (k,l) \in \{(i,i), (j,j)\} \\ -1, & (k,l) \in \{(i,j), (j,i)\} \\ 0, & \text{otherwise,} \end{cases} \quad (7.8)$$

where k and l are generic variables that take on the pairwise values of i and j respectively (Cahill et al., 2014). An example $V_{k,l}^{(i,j)}$ primitive for a cluster pair in a three vertex image is shown in (7.9).

$$V_{k,l}^{(i,j)} := \begin{bmatrix} 1 & -1 & 0 \\ -1 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad (7.9)$$

Summing each of these submatrices over a cluster results in the structure shown in (7.10). The ones in the corners are from boundary conditions and do not represent any real structure as the pixel order could be permuted to place the diagonal entry of 1 on any pixel.

$$V_{k,l}^{(i,j)} := \begin{bmatrix} 1 & -1 & & & & & \\ -1 & 2 & -1 & & & & \\ & -1 & 2 & \ddots & & & \\ & & \ddots & -1 & 2 & -1 & \\ & & & -1 & 2 & -1 & \\ & & & & -1 & 1 & 1 \end{bmatrix} \quad (7.10)$$

We note that pixels belonging to each cluster are in different places with respect to the full scene order, so \mathbf{V} is a scattered combination matrix of all subcluster matrices. The net effect of this construct is to provide a forcing function, promoting similar pixels in spectral space being proximal in manifold space (Czaja and Ehler, 2012; Cahill et al., 2014).

Several forms of cluster potentials are seen in the literature, and most focus on incorporating (fusing) spatial information to the data-dimensionality reduction step (manifold learning, spectral embedding) in clustering and segmentation (7.11) or injecting expert-user information in classification analyses (Cahill et al., 2015).

$$\mathbf{v} = \sum_{i=1}^k \sum_{x_j \in NN(i)} \mathbf{v}^{(ij)} \cdot \exp\left(-\frac{\|v_i - v_j\|^2}{2\sigma_d^2}\right) \quad (7.11)$$

The spatial heat-diffusion weighting to the right of (7.11) encodes stronger component attraction between proximal pixels, while $V_{k,l}^{(i,j)}$ encodes (identifies) the locations between spatial neighbors. Cahill et al. (2014) describe a cluster potential modification to promote even stronger attraction for pixels with higher similarity measures from possibly multiple metrics, given by

$$\mathbf{v} = \sum_{i=1}^k \sum_{x_j \in NN(i)} \mathbf{v}^{(ij)} \cdot \gamma_{ij} \cdot \exp\left(-\frac{\|v_i - v_j\|^2}{2\sigma_d^2}\right), \quad (7.12)$$

where γ_{ij} is chosen to provide more weight to those neighbors that exhibit additional similarities. For example, Cahill et al. (2014) offer use of the (2.4) or (2.6) spectral weighting equations, both providing greater attraction for spatial neighbors also displaying spectral similarities. The resultant family of methods is called Spatial-Spectral Schroedinger Eigenmaps (SSSE).

7.1 Modified SNN Cluster-Potentials for HSI Classification

Cahill et al.'s (2014) elegant generalization of SSSE provides a means to extend the family of methods with any similarity information in the Schroedinger Eigenmap construction. As such we offer extensions using SNN and MP measures (Chapter 3) in the generation of γ_{ij} (7.13).

$$\gamma_{ij} = \begin{cases} SNN_{ij}, & SSSE \text{ with SNN Potentials} \\ MP_{ij}, & SSSE \text{ with MP Potentials} \\ SNN_{ij} \cdot \exp\left(\frac{d(v_i, v_j)^2}{2\sigma^2}\right), & \text{spectrally similar with shared NN} \end{cases} \quad (7.13)$$

In (7.13), SNN_{ij} is the number of shared neighbors in spectral space and promotes the concept of community structure through common neighbors; SNN_{ij} and MP_{ij} are given by (3.1)-(3.4) and (3.7) respectively. The third option provides protection against situations like those shown to the left of Figure 2-7, where distances between neighbors may be similar, but cluster structure indicates otherwise, and is akin to the use of common

neighbors in Fan and Messinger (2014). Cahill et al. (2014) set the width of the spatial neighborhood to one pixel and $\alpha = \hat{\alpha} \cdot \text{tr}(\mathbf{L})/\text{tr}(\mathbf{V})$ so that the manifold construction can be more spatially or spectrally weighted through $\hat{\alpha}$. We adopt the same practice here.

Inclusion of these new measures may improve performance based on results reviewed in Chapter 4, so it is beneficial to explore how these new measures impact the derivation of manifold coordinates for HSI classification, comparing these new algorithms to Cahill's state-of-the-art performance. As such, we replicate the classification experiment described in Cahill et al. (2014), wherein 10% of the ground truth pixels for each class are used to determine class boundaries via linear discriminant analysis (MATLAB). We also ramp $\hat{\alpha}$ from 0 through 100 to skew the embedding from one based solely on spectral neighbors ($\hat{\alpha} = 0$) to one primarily based on spatial information ($\hat{\alpha} = 100$).

We call the bottom option in (7.13) the shared nearest neighbor SSSE (SNN-SSSE) algorithm for *Salinas*, *Indian Pines*, and *SalinasA* are shown in Figure 7-1. Notice the exceptional utility afforded by inclusion of spatially-based cluster potentials that exploit spatial coherency present in imagery. Notice that results on the left-hand side are based solely on edges built from spectral features (no spatial influence through \mathbf{V}). The influence of spatial neighbors (through cluster potentials) increases from left to right, enforcing spatial coherency and tightening up the results.

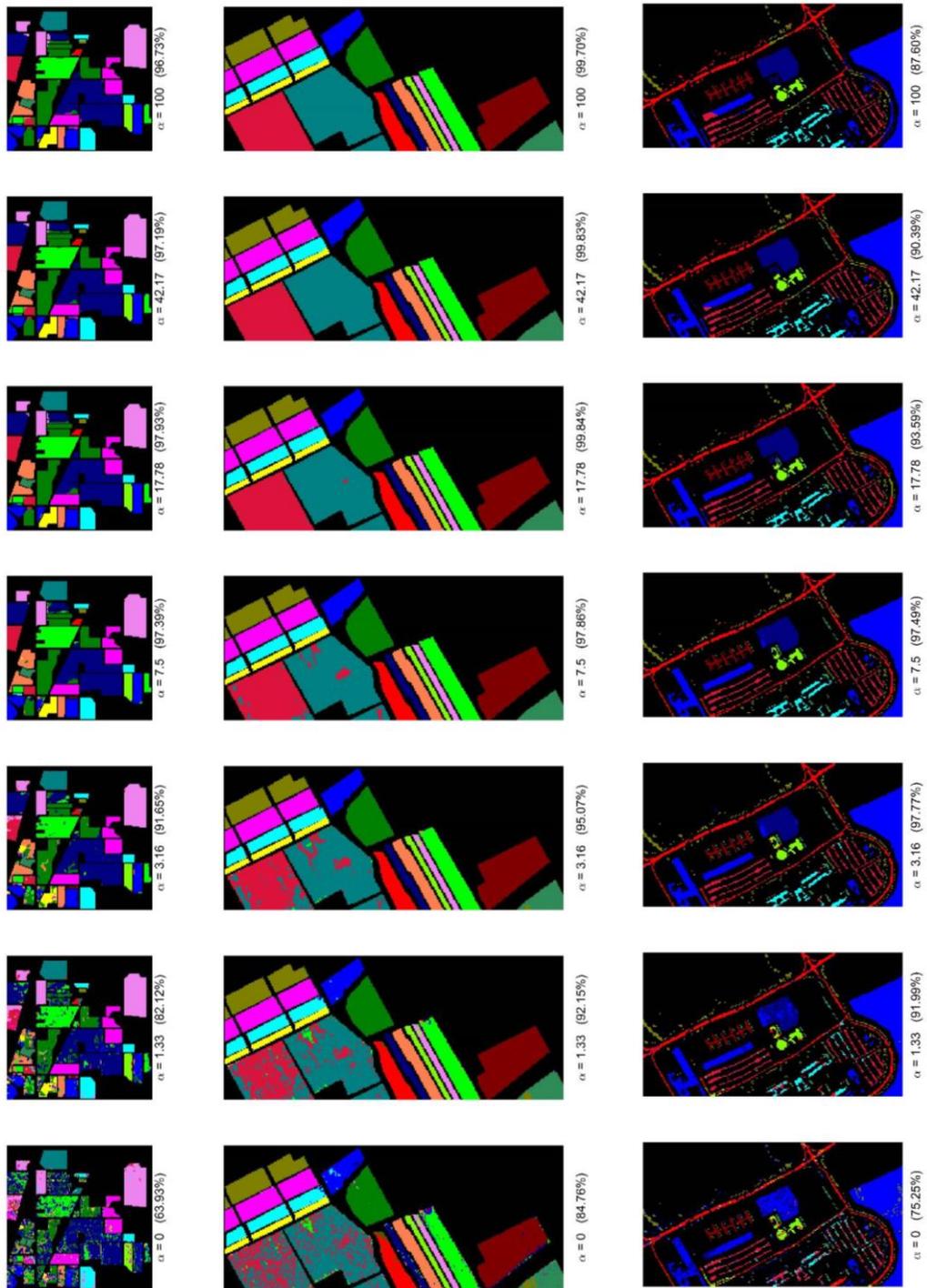


Figure 7-1. SNN-SSSE with spectral through spatial weightings for the *Indian Pines*, *Salinas*, and *Pavia University* scenes.

We compared SNN-SSSE confusion matrices with those of Cahill et al. (2014) using McNemar’s test and found statistically significant improvements for the *Microscene*, *Pavia*, and *Pavia University* scenes (Table 7-1). Cahill’s implementation demonstrated better performance on the *Salinas* scene. The only difference in this test is the inclusion of shared neighbors, which produce an increase in overall accuracy for more cluttered scenes.

Table 7-1. SNN-SSSE Classification Accuracy

Dataset	Sensor	Data Type	GSD [m]	Scene Type	Cahill et al. SSSE	SNN-SSSE
<i>SalinasA</i>	AVIRIS	<i>rad.</i>	3.7	rural	99.94	99.92
<i>Salinas</i>	AVIRIS	<i>rad.</i>	3.7	rural	99.91	99.84
<i>Indian Pines</i>	AVIRIS	<i>rad.</i>	20	rural	98.03	97.93
<i>Pavia</i>	ROSIS	<i>ref.</i>	1.3	urban	97.1	97.49
<i>Pavia Univ.</i>	ROSIS	<i>ref.</i>	1.3	urban	92.62	93.59
<i>Microscene</i>	SOC710	<i>ref.</i>	1.3E-4	~rural	98.44	98.56

Statistically significant differences indicated in **bold**.

Misclassified pixels were moderately different between the methods for all scenes, indicating there is something to be gained by fusing both approaches. The state-of-the-art algorithm described in Cahill et al. (2014) was consistently higher across all α values in the *SalinasA* and *Indian Pines* scenes even though the results were statistically equivalent.

7.2 Hubness-Based Barrier Potentials for Anomaly Detection

Barrier potentials are reviewed at the start of this chapter wherein we stated any identified barrier pixel would be drawn towards the manifold origin because of the potential well created at the node. Dorado-Munoz et al. (2016a) leveraged this fact to design a Schrodinger Eigenmap-based target detection scheme where all target-like pixels are

drawn towards the origin. We propose leveraging the same origin migration characteristic, but for the purposes of anomaly detection.

Assume a given set of pixels can be identified that describe the typical scene background. A logical thought is to define barrier potentials so that all those (background) pixels are drawn to the origin. Should this be possible, anomalousness could simply be defined in terms of Euclidean distance from the origin. This construct possesses two useful characteristics: 1) the anomaly detection algorithm is extremely simple, and 2) common (background) pixels are mapped to zero and display as a dark background.

Alas, Dorado-Munoz (2017b) describes how the inclusion of too many pixels in the barrier potential can reduce contrast between pixels as all points connected to barrier pixels are also drawn towards zero. So it is desirable to identify a few pixels that would draw a significant fraction of the background towards the origin, but not reduce the contrast to the point the embedded data is no longer useful. Recall from Chapter 5 that hubs are nodes with a much larger fraction of edges to other pixels than those on average, and exist in the center of the most spectrally dense spaces such as background clusters. We therefore posit that identifying a small number of barrier pixels that have higher connectivity may pull much of the background towards the origin, increasing the contrast of anomalous pixels. We call this approach hubness assisted anomaly detection (HAAD).

HAAD builds barrier potentials based on a fractional number of pixels possessing the highest hubness values (5.3) and then applies RX in the manifold space to detect anomalies on the background-suppressed manifold. HAAD is tested against the standard

RX method on imagery from the RIT SHARE 2012 collection (Giannandrea et al., 2013) using the Avon, New York scene collection from September of 2012 (Figure 7-2).

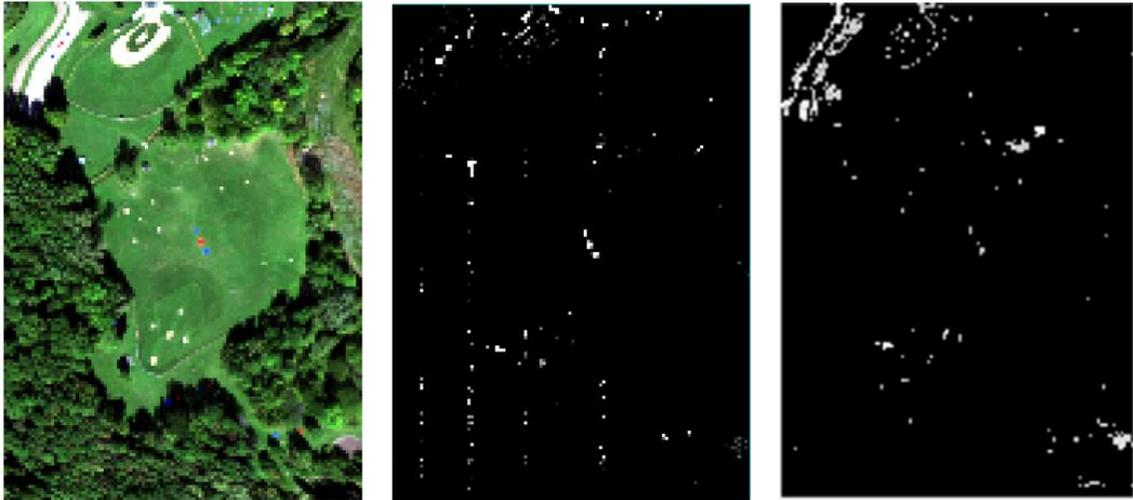


Figure 7-2. SHARE image of Avon, NY (left), RX (center) and HAAD (right) anomaly detection images.

Notice the HAAD implementation is not as susceptible to detector nonuniformities such as streaks—a sensitivity of linear methods like RX (Mohan et al., 2006). We studied the receiver operator characteristic (ROC) curves for detection of the red and blue painted panels to explore the utility of seeding barrier potentials with pixels displaying the highest hubness values (centralized tendency). The hub fraction in the barrier potential was varied from 1 to 25%. ROC analysis revealed HAAD is outperformed by the standard RX algorithm at all false alarm rates. Deeper investigation revealed that HAAD performance is limited by lack of shadow pixels in the barrier potential matrix. As such, the method should be further developed to include some number of shadow pixels to increase performance, creating a shadow-enhanced HAAD algorithm.

7.3 Summary and Contributions

New spectral similarity measures developed in this research were used to extend the current state-of-the-art in SSSE classification algorithms. While the new SNN-SSSE methods demonstrated equivalent performance to those provided by SSSE for most scenes, the SNN-SSSE algorithm was statistically better than SSSE in urban environments. We note that both SSSE and SNN-SSSE produce very different sets of misclassified pixels. This is very interesting and suggests the methods may be combined to achieve even greater classification accuracy. This should be explored in future research.

Hubness was used as a means to suppress the background contribution through barrier potentials in SSSE, exploring the further exposing anomalous pixels. Hubs are used based on their centralized tendency and higher connectivity, so fewer pixels can be used as barriers to suppress the background without overly squashing scene contrast. Use of barrier potentials in this manner has not been reported to our knowledge. Visually, this new method produces more visually appealing anomaly images not susceptible to structured noise. However, ROC analysis suggest this new method is not quite ready for prime time yet as the standard RX algorithm produced better results at all false alarm rates.

8

CONCLUSIONS AND DISCUSSION

This research explores the science (and art) of generating spectral graphs for the exploitation of hyperspectral imagery. Contributions resulting from this research can improve the performance of graph- and manifold-based analysis techniques through promoting the construction of better spectral graphs—a necessary first step that has not been rigorously studied in the past on HSI data. Recommendations for graph-construction parameter and algorithm selection are provided based on rigorous study of graph characteristics and classification performance. Specific recommendations (contributions) are summarized below.

1. Adaptive edge allocation via methods like ANN or density-weighted k -NN are better than fixed k techniques for representing community structure.
2. Mutuality is proven to be superior to superset symmetry for adjacency matrix symmetrification, especially for urban data. Superset symmetry performs on par or slightly better for rural data.
3. A density-weighted technique utilizing the cumulative density function instead of z -score is demonstrated to provide the intended edge-count mapping in diverse datasets, especially when the codensity distribution is not normal (e.g., MSI).

4. A neighborhood-size bias is introduced that enables mutual k -NN to be the highest performing method of adjacency symmetrification across a very broad range of k values and algorithms.
5. Two new similarity measures from the data mining and machine learning communities are introduced and modified to provide a data-dependent means of weighting graphs without reliance on codensity measures, a benefit that provides resiliency to the concentration of measure and hubness phenomena as data intrinsic dimensionality increases. These methods performed slightly higher than current techniques as a whole.

Despite the success of adaptive density-weighting for edge allocation, its use for adaptive density-based edge-reweighting did not provide any benefit. Investigation revealed this effect is due to the fact density is not a discriminating factor between clusters and intracluster density structure is variable, but somewhat similar for many clusters tested. Future research into HSI cluster-density structure should provide more insight into exactly why adaptive density-based reweighting methods do not provide the originally anticipated benefits. Density-based prescaling of primary measures did increase classification accuracy for some graphs despite increasing the number of ϕ -edges, so there is more to be learned about reweighting distance measures prior to edge selection instead of applying reweighting techniques to weights after edge selection.

In a deeper probe on codensity characteristics, we revisited aspects of distance measures in high-dimensional spaces and demonstrate concentration of measure and

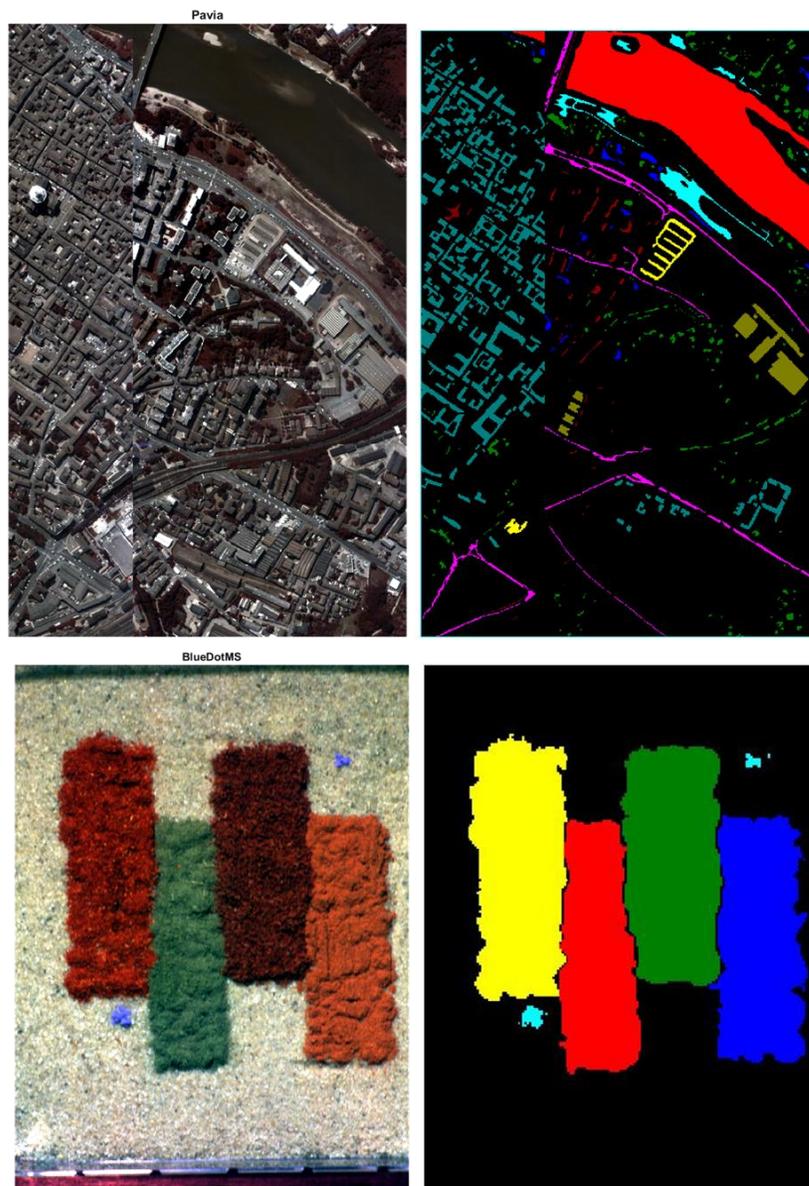
hubness—two aspects of the *curse of dimensionality*—appear in HSI data. We demonstrate these two phenomena are present to varying degrees, dependent on data (scene) characteristics, and are possible items of concern for analysis methods utilizing edge weights like Schroedinger Eigenmaps or normalized cuts. In contrast, observed low hub counts in fixed k -NN means they are not a concern for counting algorithms like k -NN. Graphs built from adaptive edge-allocation methods result in a larger number of hubs, but are still not an issue for k -NN.

As HSI sensor capabilities continue to increase, higher spatial and spectral resolution data can induce higher intrinsic dimensionality, resulting in a higher probability of impact from concentration of measure and hubness. Appearance of the hubness phenomenon is data dependent, and further study is required to identify data characteristics correlated with hub formation—specifically bad hubs—so they can be mitigated should they occur. Conversely, different construction methods can also be selected if characteristic measures indicate a propensity for the phenomena. While classification algorithms based on counting are not impacted from hubs crowding neighborhood lists, the question of how hubs (especially bad hubs) impact manifold learning or other algorithms dependent on edge weights requires further investigation and is the next logical step in this research.

SNN was incorporated into current spatial-spectral classification methods based on Schroedinger Eigenmaps, yielding better performance than the current state-of-the-art SSSE methods on urban data. The SNN algorithm produced equivalent performance on rural data, except in a single dataset where SSSE was slightly better. A very intriguing aspect of these classification results is that the misclassified pixels are moderately

different, suggesting a combination of both methods may be able to outperform either one individually. Additional research into this area is warranted to explore this exciting possibility.

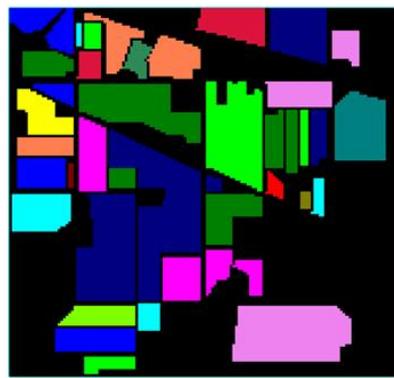
APPENDIX A: HSI DATA SETS WITH GROUND TRUTH

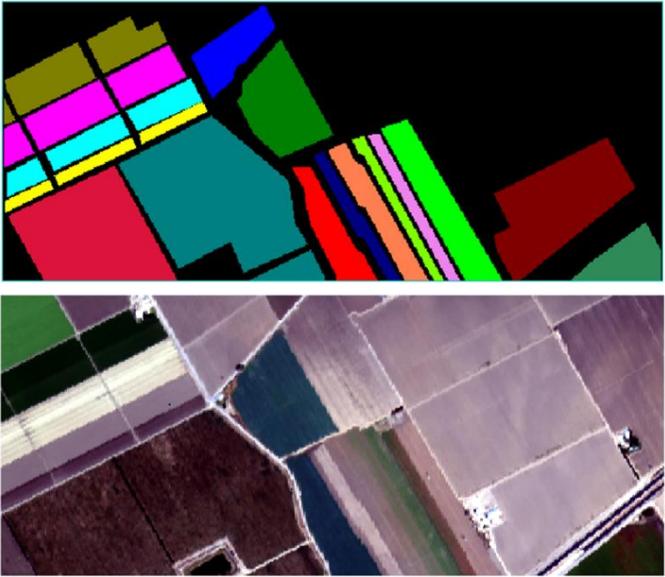
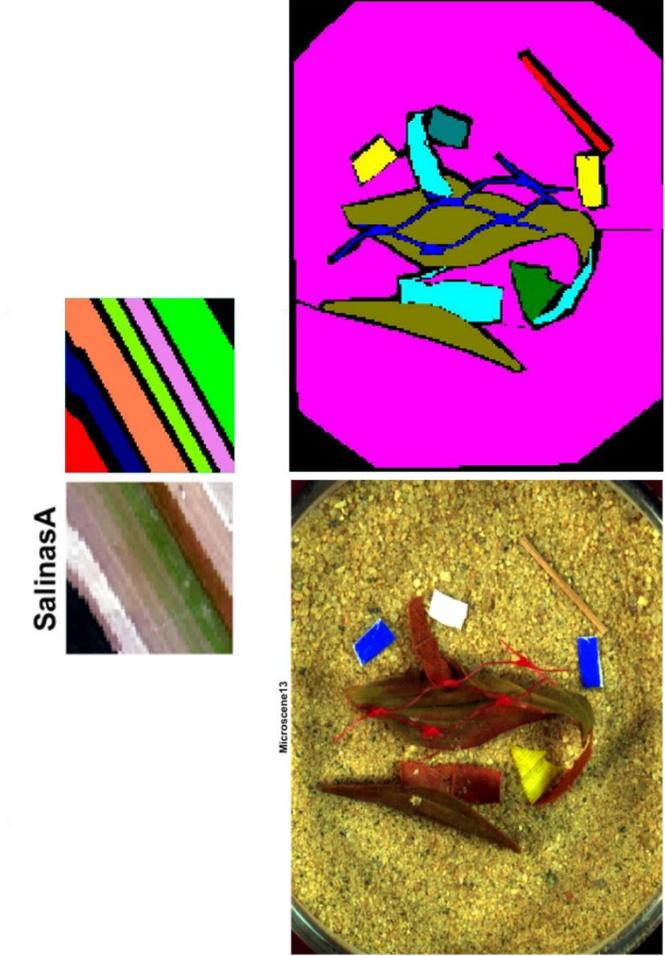


PaviaU



PerduePines



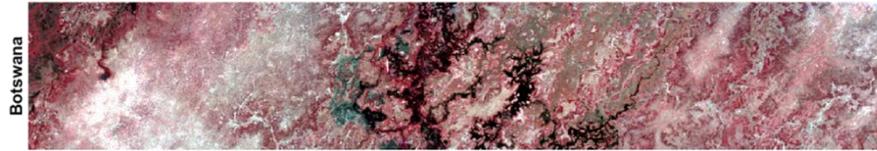


SalinasA

Microscop13

Salinas

APPENDIX B: HSI DATA SETS



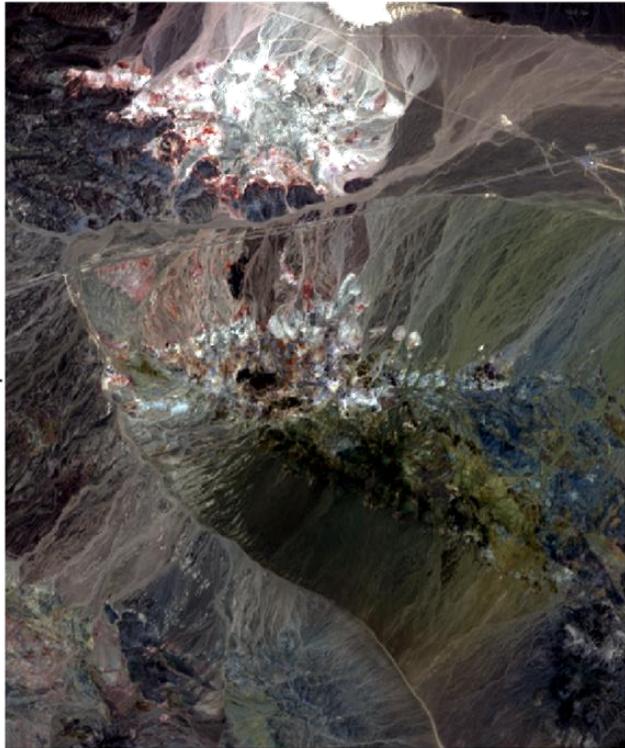
Cuprite-Rad

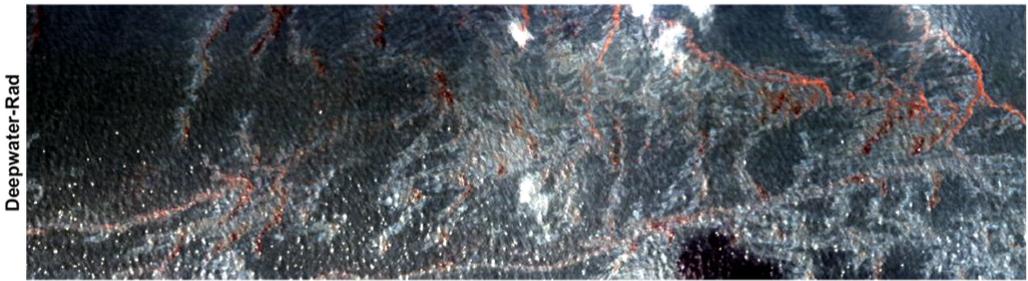


CookeCity



Cuprite







KSC



Microarray



JasperRidge



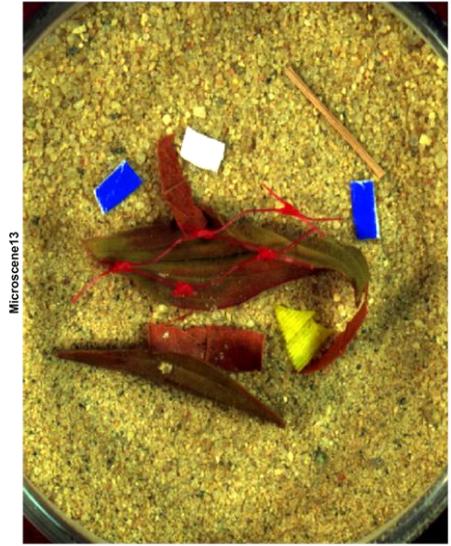
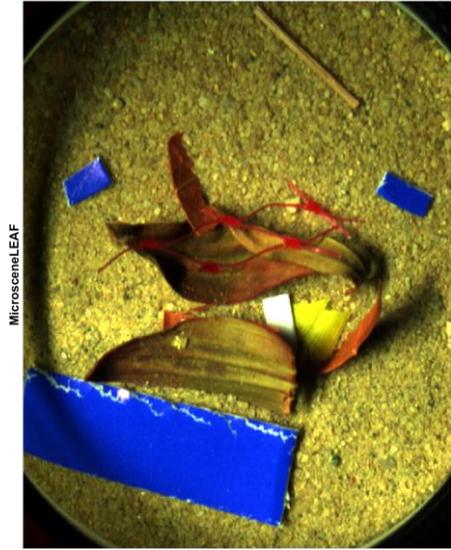
LunarLake



HyperDesert



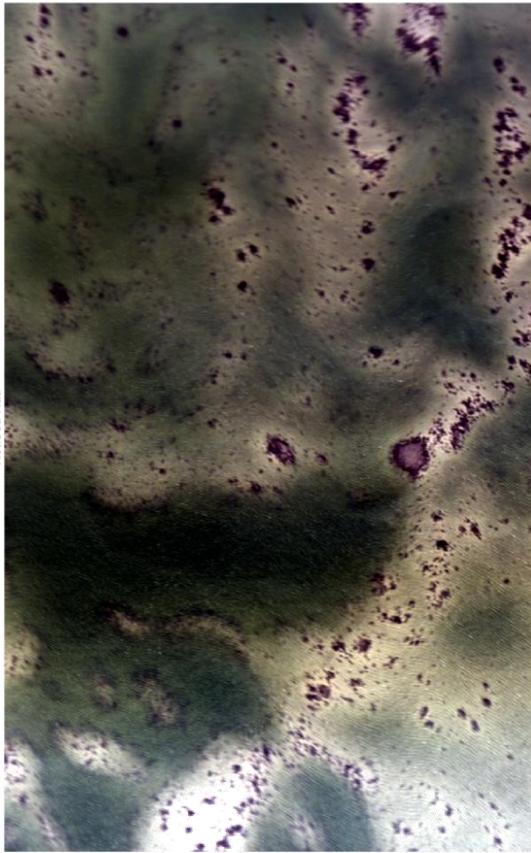
Gulf-Rad



Reno-Rad



RedSea-Rad



Phoenix





APPENDIX C: RELATED PUBLICATIONS

Conference Proceedings

Allen, D., R. Resmini, C. Deloye, and J. Stevens, 2013. “A microscene approach to the evaluation of hyperspectral system level performance.” In *Proc. of the SPIE, Algorithms and Technologies for Multispectral, Hyperspectral, and Ultraspectral Imagery XIX*, **8743**(1M), (April): 1-13.

Flexer, A. and J. Stevens, 2016. “Mutual proximity graphs for music recommendation.” *9th International Workshop on Machine Learning and Music (MML16)*.

Peer Reviewed Journal Publications

Stevens, Jeff, R. Resmini, and D. Messinger, *IEEE Transactions on Geoscience and Remote Sensing* **55**(10), (October): 5966 – 5983.

Flexer, A. and Jeff Stevens, 2017. “Mutual proximity graphs for improved reachability in music recommendation.” *Journal of New Music Research*, open access, published online 03 Aug 2017. doi: <https://doi.org/10.1080/09298215.2017.1354891>.

Submitted for Peer Review

Stevens, J. and A. Flexer, “Concentration of Measure and Hubness in Graphs of Hyperspectral Data.”

REFERENCES

- Albano, J., D. Messinger, A. Schlamm, and W. Basener. 2011. "Graph theoretic metrics for spectral imagery with application to change detection." *Proc. SPIE. Algorithms and Technologies for Multispectral, Hyperspectral, and Ultraspectral Imagery XVII* 804809 (May 2011): 1-12. doi: 10.1117/12.883574.
- Albano, J., and D. Messinger. 2012a. "Euclidean commute time distance embedding and its application to spectral anomaly detection." *Proc. SPIE Algorithms and Technologies for Multispectral, Hyperspectral, and Ultraspectral Imagery XVIII* 83902G (May 2012): 1-15. doi: 10.1117/12.918411.
- Albano, J., D. Messinger, and S. Rotman. 2012b. "Commute time distance transformation applied to spectral imagery and its utilization in material clustering." *Optical Engineering* 51(7): 1-19. doi: 10.1117/1.OE.51.7.076202.
- Albano, J., D. Messinger, and E. Ientilucci. 2013. "Spectral target detection using a physical model and a manifold learning technique." *Proc. SPIE. Algorithms and Technologies for Multispectral, Hyperspectral, and Ultraspectral Imagery XIX* 874318: 1-14. doi: 10.1117/12.2015587.
- Aggarwal, C., A. Hinneburg, and D. Keim. 2001. "On the surprising behavior of distance metrics in high dimensional space." *Database Theory, ICDT 2001, 8th International Conference*, (Springer, Berlin, Heidelberg): 420-434.
- Allen, D., R. Resmini, C. Deloye, and J. Stevens. 2013. "A microscene approach to the evaluation of hyperspectral system level performance." *Proc. of the SPIE, Algorithms and Technologies for Multispectral, Hyperspectral, and Ultraspectral Imagery XIX* 8743 (May 2013): 1-13. doi:10.1117/12.2015834.
- Bachmann, C., T. Ainsworth, and R. Fusina. 2004. "Improvements to land-cover and invasive species mapping from hyperspectral imagery in the Virginia Coast reserve." *Proc. IEEE International Geoscience and Remote Sensing Symposium* 6: 4180-4183. doi: 10.1109/IGARSS.2004.1370056.
- Bachmann, C., T. Ainsworth, and R. Fusina. 2005. "Exploiting manifold geometry in hyperspectral imagery." *IEEE Transactions on Geoscience and Remote Sensing* 43(3): 441-454. doi: 10.1109/TGRS.2004.842292.

- Bachmann, C., T. Ainsworth., and R. Fusina. 2006. "Improved Manifold Coordinate Representations of Large-Scale Hyperspectral Scenes." *IEEE Transactions on Geoscience and Remote Sensing* 44(10): 2786 – 2803. doi: 10.1109/TGRS.2006.881801.
- Bachmann, C., T. Ainsworth, R. Fusina, M. Montes, J. Bowles, D. Korwan, and D. Gillis. 2009. "Bathymetric retrieval from hyperspectral imagery using manifold coordinate representations." *IEEE Transactions on Geoscience and Remote Sensing* 47(3): 884-897.
- Basener, B., E. Ientilucci and D. Messinger. 2007. "Anomaly detection using topology." *Proc. SPIE, Algorithms and Technologies for Multispectral, Hyperspectral, and Ultraspectral Imagery XIII* 65650J (May 2007): 1-10. doi:10.1117/12.745429.
- Basener, W. and D. Messinger. 2009. "Enhanced detection and visualization of anomalies in spectral imagery." *Proc. SPIE Algorithms and Technologies for Multispectral, Hyperspectral, and Ultraspectral Imagery XV* 73341Q (April 2009): 1-10. doi: 10.1117/12.818672.
- Basener, W. 2010. "Clutter and anomaly removal for enhanced target detection." *Proc. SPIE, Algorithms and Technologies for Multispectral, Hyperspectral, and Ultraspectral Imagery XVI* 769525 (April 2010): 1-15. doi: 10.1117/12.850303.
- Baumgardner, M., L. Biehl, and D. Landgrebe. 1992. "220 Band AVIRIS Hyperspectral Image Data Set: June 12, 1992 Indian Pine Test Site 3." *Purdue University Research Repository*. doi: 10.4231/R7RX991C. <https://purr.purdue.edu/publications/1947/1>. 2015.
- Belkin, M. and P. Niyogi. 2003. "Laplacian Eigenmaps for dimensionality reduction and data representation." *Neural Computing* 15(6): 1373-1396. doi:10.1162/089976603321780317.
- Belkin, M. and P. Niyogi. 2008. "Convergence of Laplacian Eigenmaps." *Statistics* 19: 1–31.
- Bellman, R., 1961. *Adaptive Control Processes: A Guided Tour*. Princeton University Press.
- Benedetto, J., W. Czaja, J. Dobrosotskaya, T. Doster, K. Duke, and D. Gillis. 2012a. "Semi-supervised learning of heterogeneous data in remote sensing imagery." *Proc. SPIE, Independent Component Analyses, Compressive Sampling, Wavelets, Neural Net, Biosystems, and Nanoengineering X* 840104 (May 2012): 1-12. doi: 10.1117/12.919259.

- Benedetto, J. W., Czaja, J., Dobrosotskaya, T., Doster, K., Duke, and D. Gillis. 2012b. "Integration of heterogeneous data for classification in hyperspectral satellite imagery." *Proc. SPIE, Algorithms and Technologies for Multispectral, Hyperspectral, and Ultraspectral Imagery XVIII* 839027 (May 2012): 1-12. doi: 10.1117/12.919236.
- Beyer, K., J. Goldstein, R. Ramakrishnan, and U. Shaft. 1999. "When is "nearest neighbor" meaningful?" In *Database Theory—ICDT'99* (Jan 1999): 217-235.
- Boardman, J. 1990. "Inversion of high spectral resolution data." *Proc. of SPIE 1990*, (Orlando, FL): 2069-2072.
- Boardman, J. 1994. "Automating spectral unmixing of AVIRIS data using convex geometry concepts." *Summaries of the Fourth Annual JPL Airborne Geoscience Workshop*, (Jet Propulsion Laboratory, Pasadena, CA): 11–14.
- Boardman, J. 1998. "Leveraging the high dimensionality of AVIRIS data for improved sub-pixel target unmixing and rejection of false positives: Mixture Tuned Matched Filtering." *Proc. of the 7th Annual JPL Airborne Geoscience Workshop*, JPL Publication 97(1): 55.
- Buades, A., B. Coll, and J. Morel. 2005. "A non-local algorithm for image denoising." *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2005)* 2: 60-65.
- Cahill, N. W., Czaja, and D. Messinger. 2014. "Schroedinger Eigenmaps with nondiagonal potentials for spatial-spectral clustering of hyperspectral imagery." *Proc. SPIE, Algorithms and Technologies for Multispectral, Hyperspectral, and Ultraspectral Imagery XX* 908804 (June 2014): 1-13. doi: 10.1117/12.2050651.
- Cahill, N., S. Chew, and P. Wenger. 2015. "Spatial-spectral dimensionality reduction of hyperspectral imagery with partial knowledge of class labels." *Proc. SPIE, Algorithms and Technologies for Multispectral, Hyperspectral, and Ultraspectral Imagery XXI* 94720S (May 2015): 1-14. doi: 10.1117/12.2177139.
- Chen, Y. M., Crawford, and J. Ghosh. 2005. "Applying nonlinear manifold learning to hyperspectral data for land cover classification." *Proc. IGARSS 6* (July 2012): 4311–4314.
- Czaja, W. and M. Ehler. 2012. "Schroedinger Eigenmaps for the Analysis of Biomedical Data." *IEEE Transactions on Pattern Analysis and Machine Intelligence* 35(5): 1-8. doi: 10.1109/TPAMI.2012.270.

- Dorado-Munoz, L., D. Messinger, and A. Ziemann. 2013. "Target detection using the background model from the topological anomaly detection algorithm." *Proc. SPIE., Algorithms and Technologies for Multispectral, Hyperspectral, and Ultraspectral Imagery XIX* 87430M (May 2013): 1-12. doi: 10.1117/12.2015335.
- Dorado-Munoz, L. and D. Messinger. 2016a. "Initial study of Schroedinger eigenmaps for spectral target detection," *Optical Engineering* 55(8). <http://dx.doi.org/10.1117/1.OE.55.8.083101>.
- Dorado-Munoz, L. 2016b. "Spectral Target Detecting Using Schroedinger Eigenmaps." PhD. Thesis. Rochester Institute of Technology.
- Durrant, R., and A. Kabán. 2009. "When is 'nearest neighbor' meaningful: A converse theorem and implications." *Journal of Complexity* 25(4): 385-397.
- Edwards and Galuska. Available online: <http://www.wcaslab.com/tech/tbftir.htm>, <https://wikispaces.psu.edu/display/RemSens597K/F>
- Eismann, M. 2012. *Hyperspectral Remote Sensing*. Bellingham, WA: SPIE Press.
- Ertöz, L., M. Steinbach, and V. Kumar. 2002. "A new shared nearest neighbor clustering algorithm and its applications." *Workshop on Clustering High Dimensional Data and its Applications* at 2nd SIAM ICDM. 1-15.
- Ertöz, L., M. Steinbach, and V. Kumar. 2003. "Finding clusters of different sizes, shapes, and densities in noisy, high dimensional data." *Proc. of SIAM International Conference on Data Mining*. 47-58.
- Fan, L., and D. Messinger. 2014. "Graph based hyperspectral image segmentation with improved affinity matrix." *Proc. SPIE., Algorithms and Technologies for Multispectral, Hyperspectral, and Ultraspectral Imagery XX* 908802, (June): 1-12. doi: 10.1117/12.2050397.
- Flexer, A., and D. Schnitzer. 2013. "Can shared nearest neighbors reduce hubness in high-dimensional spaces?" *In Data Mining Workshops (ICDMW), 2013 IEEE 13th International Conference on*. 460-467.
- Foster, D., K. Amano, S. Nascimento, and M. Foster. 2006. "Frequency of metamerism in natural scenes." *Journal of the Opt. Soc. of America A* 23: 2359-2372.
- Francois, D., V. Wertz, and M. Verleysen. 2007. "The Concentration of Fractional Distances." *IEEE Transactions on Knowledge and Data Engineering* 19(7): 873-886. doi: 10.1109/TKDE.2007.1037.

- Funk, C., J. Theiler, D. Roberts, and C. Borel. 2001. "Clustering to improve matched filter detection of weak gas plumes in hyperspectral thermal imagery." In *IEEE Transactions on Geoscience and Remote Sensing* 39(7): 1410–1420.
- Gallier, J. 2016. "Spectral Theory of Unsigned and Signed Graphs Applications to Graph Clustering: a Survey." University of Pennsylvania, Available online: <https://arxiv.org/pdf/1601.04692.pdf>.
- Giannandrea, A., et al. 2013. "The SHARE 2012 data campaign." *Proc. SPIE, Algorithms and Technologies for Multispectral, Hyperspectral, and Ultraspectral Imagery XIX* 87430F (May 2013): 1-15. doi: 10.1117/12.2015935.
- Gillis, D., and J. Bowles. 2012. "Hyperspectral image segmentation using spatial-spectral graphs," *Proc. SPIE., Algorithms and Technologies for Multispectral, Hyperspectral, and Ultraspectral Imagery XVIII* 83901Q (May 2012): 1-11. doi: 10.1117/12.919743.
- Gillis D, J. Bowles, G. Lamela, W. Rhea, and C. Bachmann, M. Montes, and T. Ainsworth. 2005. "Manifold learning techniques for the analysis of hyperspectral ocean data." *Proc. SPIE, Algorithms and Technologies for Multispectral, Hyperspectral, and Ultraspectral Imagery XI* 5806 (June 2005): 342-351. doi:10.1117/12.603660.
- Griffin, M., J. Kerekes, K. Farrar, and H. Burke. 2001. "Characterization of gaseous effluents from modeling of LWIR hyperspectral measurements." *Proc. SPIE Algorithms and Technologies for Multispectral, Hyperspectral, and Ultraspectral Imagery VII* 4381: 360-369.
- Grupo de Inteligencia Computacional (GIC) *University of País Vasco, Spain*. HSI Ground Truth Website http://www.ehu.es/ccwintco/index.php?title=Hyperspectral_Remote_Sensing_Scenes
- Harsanyi, J. and C. Chang. 1994. "Hyperspectral Image Classification and Dimensionality Reduction: An Orthogonal Subspace Projection Approach." *IEEE Transactions on Geoscience and Remote Sensing* 32(4): 779-785. doi: 10.1109/36.298007.
- He, J., L. Zhang, Q. Wang, and Z. Li. 2009. "Using Diffusion Geometric Coordinates for Hyperspectral Imagery Representation." *Geoscience and Remote Sensing Letters, IEEE* 6(4): 767-771. doi: 10.1109/LGRS.2009.2025058.
- Healy, G. and D. Slater. 2009. "Models and methods for automated material identification in hyperspectral imagery acquired under unknown illumination and atmospheric conditions." *IEEE Transactions on Geoscience and Remote Sensing* 37(6): 2706-2717.

- Heylen, R., M. Parente, and P. Scheunders. 2017. "Estimation of the Number of Endmembers in a Hyperspectral Image via the Hubness Phenomenon," *IEEE Transactions on Geoscience and Remote Sensing* 55(4): 2191-2200. doi: 10.1109/TGRS.2016.2638541.
- Hinneburg, A., C. Aggarwal, and D. Keim. 2000. "What is the nearest neighbor in high dimensional spaces?" *Proc. Of the 26th International Conference on Very Large Databases* 99. Cairo, Egypt: 506-515.
- Hou, B., X. Zhang, Q. Ye, and Y. Zheng. 2013. "A Novel Method for Hyperspectral Image Classification Based on Laplacian Eigenmap Pixels Distribution-Flow." *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing* 6(3): 1602-1618. doi: 10.1109/JSTARS.2013.2259470.
- Houle, M., H. Kriegel, P. Kröger, E. Schubert, and A. Zimek. 2010. "Can shared-neighbor distances defeat the curse of dimensionality?" *Scientific and Statistical Database Management*. (Springer Berlin Heidelberg): 482-500.
- Jarvis, R., and E. Patrick. 1973. "Clustering using a similarity measure based on shared near neighbors." *IEEE Transactions on Computers* 100(11): 1025-1034.
- Jegou, H., C. Schmid, H. Harzallah, and J. Verbeek. 2010. "Accurate image search using the contextual dissimilarity measure." *IEEE Transactions on Pattern Analysis and Machine Intelligence* 32(1): 2-11.
- Jellison, G. and D. Miller. 2006. "Determination of gas plume temperature from molecular emission spectra." *Optical Engineering* 45(1). doi: 10.1117/1.2151814.
- Jimenez, L., and D. Landgrebe. 1998. "Supervised classification in high-dimensional space: geometrical, statistical, and asymptotical properties of multivariate data." *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)* 28(1): 39-54.
- Kabán, A. 2010. "On the distance concentration awareness of certain data reduction techniques." *Pattern Recognition* 44(2): 265-277.
- Kabán, A. 2011. "Non-parametric detection of meaningless distances in high dimensional data." *Statistics and Computing* 22(2): 375-385.
- Kameshwaran, K., and K. Malarvizhi. 2014. "Survey on Clustering Techniques in Data Mining." *International Journal of Computer Science and Information Technologies* 5(2): 2272-2276.

- Keshava, N., and J. Mustard. 2002. "Spectral Unmixing," *IEEE Signal Processing Magazine* 19(1): 44-57. doi:10.1109/79.974727.
- Keshava, N. 2003. "A Survey of Spectral Unmixing Algorithms," *Lincoln Laboratory Journal*, 14(1): 55-78. http://www.ll.mit.edu/publications/journal/pdf/vol14_no1/14_1survey.pdf
- Kontschieder, P., M. Donoser, and H. Bischof. 2009. "Improving affinity matrices by modified mutual kNN-Graphs." *33rd workshop of the Austrian association for pattern recognition (AAPR/OAGM)*. 1-11.
- Levina E., and J. Bickel. 2005. "Maximum likelihood estimation of intrinsic dimension." *Advances in Neural Information Processing Systems* 17: 777-784. Cambridge, MA: MIT Press.
- Manolakis, D., D. Marden, and G. Shaw. 2003. "Hyperspectral Image Processing for Automatic Target Detection Applications." *Lincoln Laboratory Journal*. 14(1): 79-116. http://www.ll.mit.edu/publications/journal/pdf/vol14_no1/14_1hyperspectralprocessing.pdf
- Manolakis, D., M. Rossacci, J. Cipar, R. Lockwood, T. Cooley, and J. Jacobson. 2005. "Statistical characterization of natural hyperspectral backgrounds using t - elliptically contoured distributions," *Proc. of the SPIE, Algorithms and Technologies for Multispectral, Hyperspectral, and Ultraspectral Imagery XI* 5806 (June 2005): 56-65. doi: 10.1117/12.602315.
- Manolakis, D., M. Rossacci, J. Cipar, R. Lockwood, T. Cooley, J. Jacobson. 2006. "Statistical Characterization of Natural Hyperspectral Backgrounds," *Proc. IEEE Geoscience and Remote Sensing Symposium*. 1624-1627.
- Manolakis, D., L. Jairam, D. Zhang, and M. Rossacci. 2007. "Statistical models for LWIR hyperspectral backgrounds and their applications in chemical agent detection." *Proc. SPIE Algorithms and Technologies for Multispectral, Hyperspectral, and Ultraspectral Imagery XIII* 6565 (May 2007): 1-12. doi: 10.1117/12.718378.
- Manolakis, D., R. Lockwood, T. Cooley, and J. Jacobson. 2009. "Is There a Best Hyperspectral Detection Algorithm?" *SPIE Newsroom*.
- Matteoli, S., M. Diani, and G. Corsini. 2010. "A tutorial overview of anomaly detection in hyperspectral images." *IEEE Aerospace and Electronic Systems Magazine* 25(7): 5-27.

- Matteoli, S., T. Veracini, M. Diani, and G. Corsini. 2013. "Models and Methods for Automated Background Density Estimation in Hyperspectral Anomaly Detection." *IEEE Transactions on Geoscience and Remote Sensing* 51(5): 2837-2852. doi: 10.1109/TGRS.2012.2214392.
- Mercovich, R., J. Albano, and D. Messinger. 2011. "Techniques for the graph representation of spectral imagery." presented at *IEEE 3rd Workshop on Hyperspectral Image and Signal Processing: Evolution in Remote Sensing (WHISPERS)*.
- Merkwirth, C., U. Parlitz, and W. Lauterborn. 2000. "Fast nearest-neighbor searching for nonlinear signal processing." *Physical Review E* 62(2): 2089-2097.
- Messinger, D., and J. Albano. 2011. "A graph theoretic approach to anomaly detection in hyperspectral imagery." *Presented at IEEE 3rd Workshop on Hyperspectral Image and Signal Processing: Evolution in Remote Sensing (WHISPERS)*. Lisbon, Portugal. (June 2011): 1-4.
- Moëllic, P., J. Haugeard, and G. Pittel. 2008. "Image Clustering Based on a Shared Nearest Neighbors Approach for Tagged Collections.", *Proc. of the 2008 international conference on Content-based image and video retrieval*. 269-278.
- Mohan, A., G. Sapiro, and E. Bosch. 2006. "Spatially coherent nonlinear dimensionality reduction and segmentation of hyperspectral images." *IEEE Geoscience and Remote Sensing Letters* 4(2): 1-5.
- NASA Jet Propulsion Laboratory Airborne Visible / Infrared Spectrometer (AVIRIS), website: <http://aviris.jpl.nasa.gov/>
- National Institute of Standards and Technology (NIST) Hyperspectral Microscopy Project, website: <https://www.nist.gov/programs-projects/hyperspectral-microscopy>
- Ozaki, K., M. Shimbo, M. Komachi, and Y. Matsumoto. 2011. "Using the mutual k-nearest neighbor graphs for semi-supervised classification of natural language data." *Proc. of the fifteenth conference on computational natural language learning*. Association for Computational Linguistics. 154-162.
- Patidar, A., J. Agrawal, and N. Mishra. 2012. "Analysis of different similarity measure functions and their impacts on shared nearest neighbor clustering approach." *International Journal of Computer Applications* 40(16): 1-5.
- Pestov, V. 2000. "On the geometry of similarity search: dimensionality curse and concentration of measure." *Information Processing Letters* 73(1): 47-51.

- Prasad, S. and L. Bruce. 2008. "Limitations of principal components analysis for hyperspectral target recognition," *IEEE Geoscience and Remote Sensing Letters* 5: 625–629.
- Radovanović, M., A. Nanopoulos, and M. Ivanović. 2010a. "Hubs in space: Popular nearest neighbors in high-dimensional data." *The Journal of Machine Learning Research* 11: 2487-2531.
- Radovanović, M., A. Nanopoulos, and M. Ivanović. 2010b. "On the existence of obstinate results in vector space models." *Proc. of the 33rd international ACM SIGIR conference on Research and development in information retrieval*. 186-193.
- Schlamm, A., R. Resmini, D. Messinger, and B. Basener. 2010. "A comparison study of dimension estimation algorithms." *Proc. of the SPIE, Algorithms and Technologies for Multispectral, Hyperspectral, and Ultraspectral Imagery XVI* 76952-D (April 2010): 1-8. doi: 10.1117/12.849125.
- Schlamm, A., and D. Messinger. 2011. "An empirical estimate of the multivariate normality of hyperspectral image data." *Proc. of the SPIE, Algorithms and Technologies for Multispectral, Hyperspectral, and Ultraspectral Imagery XVII* 80481J (May 2011): 1-11. doi:10.1117/12.881642.
- Schnitzer, D., A. Flexer, M. Schedl, and G. Widmer. 2012. "Local and global scaling reduce hubs in space." *The Journal of Machine Learning Research* 13(1): 2871-2902.
- Schott, J. 2007. *Remote Sensing – The Image Chain Approach, 2nd ed.* New York: Oxford University Press.
- Shaw, G., and H. Burke. 2003. "Spectral Imaging for Remote Sensing," *Lincoln Laboratory Journal* 14(1): 3-28.
- Shi, J., and J. Malik. 2000. "Normalized cuts and image segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence* 22(8): 888-905.
- SpecTIR Advanced Hyperspectral and Geospatial Solutions free data website. <http://www.spectir.com/free-data-samples/>.
- Stein, D., S. Beaven, L. Hoff, E. Winter, A. Schaum, and A. Stocker. 2002. "Anomaly Detection from Hyperspectral Imagery," *IEEE Signal Processing Magazine* 19(1): 58-69. doi: 10.1109/79.974730.

- Stevens, J., R. Resmini, and D. Messinger. 2017. "Spectral-Density-Based Graph Construction Techniques for Hyperspectral Image Analysis." *IEEE Transactions on Geoscience and Remote Sensing* 55(10): 5966-5983.
- Tan, P., M. Steinbach, and V. Kumar. 2006. *Introduction to Data Mining*. Boston, MA: Addison Wesley.
- Theiler, J. B. Foy and A. Fraser. 2005. "Characterizing non-Gaussian clutter and detecting weak gaseous plumes in hyperspectral imagery." *Proc. SPIE, Algorithms and Technologies for Multispectral, Hyperspectral, and Ultraspectral Imagery XI* 5806 (June 2005): 182-193. doi:10.1117/12.604075.
- Topping, R. 2009. "Remote Sensing Graphics processing units revolutionize hyperspectral data processing." *SPIE Newsroom*. doi: 10.1117/2.1200910.1834.
- Turcotte, C. and M. Davenport. 2010. "Gas plume quantification in downlooking hyperspectral longwave infrared images." *Proc. SPIE, Image and Signal Processing for Remote Sensing XVI* 78300D: 1-8. doi:10.1117/12.865990.
- United States Geological Survey (USGS) Earth Observing 1 (EO-1) Website. <https://eo1.usgs.gov/>.
- von Luxburg, U. 2007. "A tutorial on spectral clustering," *Statistics and Computing* 17(4): 395-416.
- Wang, J. 2011. *Geometric structure of high-dimensional data and dimensionality reduction*. Springer, Berlin, Heidelberg.
- West, D. 2001. *Introduction to Graph Theory, 2nd ed.* Englewood Cliffs, NJ: Prentice Hall.
- Wikimedia Commons: EM Spectrum Image, https://commons.wikimedia.org/wiki/File:EM_spectrumrevised.png
- Willis, C. 2009. "Hyperspectral Target Detection using Heavy-Tailed Distributions." *Proc. of the SPIE, Electro-Optical Remote Sensing, Photonic Technologies, and Applications III* 74820P (October 2009): 1-11. doi: 10.1117/12.830062.
- Witten, I., E. Frank, and M. Hall. 2005. *Data Mining: Practical machine learning tools and techniques, 3rd ed.* Burlington, MA: Morgan Kaufmann Publishing. Website for Weka Software: <http://www.cs.waikato.ac.nz/ml/weka/>; accessed on 11/08/2016.

- Zelnik-Manor, L. and P. Perona. 2004. "Self-tuning spectral clustering." *Advances in Neural Information Processing Systems (NIPS)*. 1601-1608. Cambridge, MA: MIT Press.
- Ziemann, A., D. Messinger, J. Albano and W. Basener. 2012. "Assessing the impact of background spectral graph construction techniques on the topological anomaly detection algorithm." *Proc. SPIE, Algorithms and Technologies for Multispectral, Hyperspectral, and Ultraspectral Imagery XVIII*, 83901Z (May 2012): 1-11. doi:10.1117/12.918889.
- Ziemann, A., D. Messinger, and J. Albano. 2013. "Target detection performed on manifold approximations recovered from hyperspectral data." *Proc. SPIE, Algorithms and Technologies for Multispectral, Hyperspectral, and Ultraspectral Imagery XIX* 874319 (May 2013): 1-16. doi: 10.1117/12.2015780.
- Ziemann, A., and D. Messinger. 2014a. "Hyperspectral target detection using graph theory models and manifold geometry via an adaptive implementation of locally linear embedding." *Proc. SPIE, Algorithms and Technologies for Multispectral, Hyperspectral, and Ultraspectral Imagery XX* 90880B (June 2014a): 1-14. doi: 10.1117/12.2050382.
- Ziemann, A., D. Messinger, and P. Wenger. 2014b. "An Adaptive K-Nearest Neighbor Graph Building Technique with Applications to Hyperspectral Imagery." *IEEE Image and Signal Processing Workshop (WNYISPW), IEEE Western New York*. 32-36.
- Ziemann, A. and D. Messinger. 2015. "An adaptive locally linear embedding manifold learning approach for hyperspectral target detection." *Proc. SPIE, Algorithms and Technologies for Multispectral, Hyperspectral, and Ultraspectral Imagery XXI* 94720O (May 2015): 1-16. doi: 10.1117/12.2177466.
- Zimek, A., E. Schubert, and H. Kriegel. 2012. "A survey on unsupervised outlier detection in high-dimensional numerical data." *Statistical Analysis and Data Mining* 5(5): 363-387.
- Zou, X., and Q. Zhu. 2011. "Adaptive neighborhood graph for ltsa learning algorithm without free-parameter." *International Journal of Computer Applications* 19(4): 28-33.

BIOGRAPHY

Jeffrey Stevens graduated from Grand Island High School, Grand Island, New York, in 1988. He received his Bachelors and Masters of Science in Electrical Engineering from the State University of New York at Buffalo in 1994 and 1996 respectively.