8-2009

Support Vector Methods for Higher-Level Event Extraction in Point Data

Jon Devine

SUPPORT VECTOR METHODS FOR HIGHER-LEVEL EVENT EXTRACTION IN POINT DATA

By

Jon Devine

B.A. Economics and Finance, University of North Carolina at Wilmington

B.A. Environmental Studies, University of North Carolina at Wilmington

M.S. Resource Economics and Policy, University of Maine

A THESIS

Submitted in Partial Fulfillment of the

Requirements for the

Degree of

Master of Science

(in Spatial Information Science and Engineering)

The Graduate School

The University of Maine

August, 2009

Advisory Committee:

Kate Beard, Professor of Spatial Information Science and Engineering

Tony Stefanidis, Associate Professor, Dept. of Earth Systems and Geoinformation Sciences, George Mason University

Silvia Nittel, Associate Professor of Spatial Information Science and Engineering

Neal Pettigrew, Professor School of Marine Sciences

LIBRARY RIGHTS STATEMENT

In presenting this thesis in partial fulfillment of the requirements for an advanced degree at The University of Maine, I agree that the Library shall make it freely available for inspection. I further agree that permission for "fair use" copying of this thesis for scholarly purposes may be granted by the Librarian. It is understood that any copying or publication of this thesis for financial gain shall not be allowed without my written permission.

Signature:

Date:

SUPPORT VECTOR METHODS FOR HIGHER-LEVEL EVENT EXTRACTION IN POINT DATA

By Jon Devine

Thesis Advisors: Dr. Kate Beard & Dr. Tony Stefanidis

An Abstract of the Thesis Presented
in Partial Fulfillment of the Requirements for the
Degree of Master of Science
(in Spatial Information Science and Engineering)
August, 2009

Phenomena occur both in space and time. Correspondingly, ability to model spatiotemporal behavior translates into ability to model phenomena as they occur in reality. Given the complexity inherent when integrating spatial and temporal dimensions, however, the establishment of computational methods for spatiotemporal analysis has proven relatively elusive.

Nonetheless, one method, the spatiotemporal helix, has emerged from the field of video processing. Designed to efficiently summarize and query the deformation and movement of spatiotemporal events, the spatiotemporal helix has been demonstrated as capable of describing and differentiating the evolution of hurricanes from sequences of images. Being derived from image data, the representations of events for which the spatiotemporal helix was originally created appear in areal form (e.g., a hurricane covering several square miles is represented by groups of pixels).

Many sources of spatiotemporal data, however, are not in areal form and instead appear as points. Examples of spatiotemporal point data include those from an epidemiologist recording the time and location of cases of disease and environmental observations collected by a geosensor at the point of its location. As points, these data cannot be directly incorporated into the spatiotemporal helix for analysis.

However, with the analytic potential for clouds of point data limited, phenomena represented by point data are often described in terms of events. Defined as change units localized in space and time, the concept of events allows for analysis at multiple levels. For instance lower-level events refer to occurrences of interest described by single data streams at point locations (e.g., an individual case of a certain disease or a significant change in chemical concentration in the environment) while higher-level events describe occurrences of interest derived from aggregations of lower-level events and are frequently described in areal form (e.g., a disease cluster or a pollution cloud).

Considering that these higher-level events appear in areal form, they could potentially be incorporated into the spatiotemporal helix. With deformation being an important element of spatiotemporal analysis, however, at the crux of a process for spatiotemporal analysis based on point data would be accurate translation of lower-level event points into representations of higher-level areal events. A limitation of current techniques for the derivation of higher-level events is that they imply bias a priori regarding the shape of higher-level events (e.g., elliptical, convex, linear) which could limit the description of the deformation of higher-level events over time.

The objective of this research is to propose two newly developed kernel methods, support vector clustering (SVC) and support vector machines (SVMs), as means for

translating lower-level event points into higher-level event areas that follow the distribution of lower-level points. SVC is suggested for the derivation of higher-level events arising in point process data while SVMs are explored for their potential with scalar field data (i.e., spatially continuous real-valued data). Developed in the field of machine learning to solve complex non-linear problems, both of these methods are capable of producing highly non-linear representations of higher-level events that may be more suitable than existing methods for spatiotemporal analysis of deformation.

To introduce these methods, this thesis is organized so that a context for these methods is first established through a description of existing techniques. This discussion leads to a technical explanation of the mechanics of SVC and SVMs and to the implementation of each of the kernel methods on simulated datasets. Results from these simulations inform discussion regarding the application potential of SVC and SVMs.

ACKNOWLEDGEMENTS

Due to the complexity resulting from the varied geography behind this thesis (residence in three states during the time period), there are many people who should be acknowledged and thanked for their assistance in getting the work completed. First among these would be Dr. Tony Stefanidis who agreed to take me on as a student after I complete my program in Resource Economics and Policy. His willingness, ideas, and support laid the groundwork for my research. Tony made great efforts to expose me to the larger research community by sending to a variety of academic conferences both international and domestic.

Having been out of Maine for the last couple years that I have been working on this project, it was key having support from the Department of Spatial Information Science and Engineering. Dr. Kate Beard was instrumental in being a link to the department and by always doing what she could to help me through. The classes that I was fortunate enough to have taken with her, particularly those in spatial analysis, were valuable to framing my research. I would also like to thank my other committee members from the University of Maine, Dr. Silvia Nittel and Dr. Neal Pettigrew, for their guidance and comments.

After Virgina, my thesis and I traveled to North Carolina. While here I benefitted tremendously from the encouragement of my parents, particularly my father. If I had a data or procedural question, he was always there and provided access to resources and counsel.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

Chapter 1

INTRODUCTION

In many applications, understanding how phenomena evolve through space and time is an important step in developing an understanding of phenomenological occurrence and behavior. For example, in epidemiology, the identification of causes of disease often involves description of how outbreaks evolve through space and time. The evolution of a disease outbreak can be an integral component of epidemiological research, providing opportunity to compare changes in spatiotemporal behavior (e.g., disease spread into certain areas and avoidance of others) with the distribution of potential risk factors. Similarly, in a range of other application domains (e.g., meteorology, security, and environmental science) description of the spatiotemporal behavior of phenomena, such as deformation (i.e., expansion and/or contraction in certain directions) and movement through space, are important elements for directing hypothesis generation and forming explanatory models. Development of computational methods to support such spatiotemporal analysis has long been a research objective.

Significant advances towards this goal have been accomplished by examining the spatial and temporal dimensions independently. Geographic information systems (GIS) are a notable example that ignore the temporal dimension and focus primarily on spatial relationships. Advances have also been made with respect to the temporal dimension. Among these developments include the refinement of statistical techniques for time series (Waller and Gotway, 2004) and the establishment of a theoretical framework for temporal analyses (Allen, 1983).

Meanwhile, conceptual frameworks based on events and involving both the spatial and temporal dimensions have emerged. A variety of definitions for events have been suggested (e.g., Quine, 1985; Peuquet, 1994; Claramunt and Jiang, 1985) .These definitions have been generalized in terms of change units localized in space and time (Beard et al., 2008). and with such a conceptualization it is possible to describe changes in the spatiotemporal behavior of phenomena at multiple levels. For instance, lower-level events describe changes that are more localized (i.e., occur over a smaller area and/or for a shorter duration) while higher-level events tend to be more distributed in space and time and are defined by aggregations of lower-level events.

Complementing the introduction of events as a conceptual framework for spatiotemporal analysis have been advances in a number of technologies. Positioning technologies such as GPS and wireless sensor networks particularly geosensor networks (Stefanidis and Nittel, 2004) are contributing to significant increases in the availability of spatiotemporal data. The increased availability of spatiotemporal data is concurrently driving the need for better computational techniques for spatiotemporal analysis. One such method, the spatiotemporal helix, recently emerged from the fields of remote sensing and video processing (Agouris and Stefanidis, 2003). The helix summarizes the deformation and movement of spatiotemporal events such as hurricanes that have been extracted from image sequences (Stefanidis, Eickhorst, et al., 2003). The events for which the spatiotemporal helix was originally created are strictly areal in form (e.g., the hurricanes which encompass several square miles are extracted from images as groups of connected pixels).

Many sources of spatiotemporal data, are not image based but collected at points. Such data include disease cases recorded by epidemiologists or readings on the concentration of pollutants made by geosensors at point locations. While collected in point form, aggregates over these points may be the analytical units of interest. For example, epidemiologists may be interested in disease clusters, the regions of unexpected spatiotemporal concentration of disease cases, and similarly, environmental scientists may want to investigate the areal extent of pollution clouds. These aggregations are referred to as higher level areal events in contrast to the point based events

If these higher-level areal events can be derived from lower-level point events with sufficient spatial detail through a process that is replicable overtime (i.e., resemble a series of events extracted from a sequence of images), they could serve as input to the spatiotemporal helix. Correspondingly, an existing method for spatiotemporal analysis could be rendered accessible to additional sources of spatiotemporal point data. At the crux of this process, however, is the implementation of techniques capable of translating groups of lower-level point events to higher-level areal events with a degree of spatial detail sufficient to enable meaningful description of spatiotemporal behavior.

1.1. Problem Statement

Methods for the translation of lower-level event points to higher-level areal events have long been a subject of research. An early and well-known example is John Snow's cholera map. His mapping of the location of cholera deaths in London's 1854 outbreak and subsequent identification of regions with concentrations of cases helped determine the cause of the disease. While John Snow relied on relatively subjective means (i.e.,

walking the streets and observing the concentrations of deaths relative to potential causes), the field of spatial analysis has since developed numerous quantitative methods to describe the relative distribution of point events in space and current techniques are specialized to describe higher-level events arising from different forms of point data.

1.1.1 Representing Higher-Level Events from Point Process Data

Perhaps the most traditional form of spatiotemporal point data is point process data. A spatial point process is a stochastic mechanism that generates a countable set of events. John Snow's cholera cases are an example of a spatial point process. They represent a set of discrete events (cholera deaths) distributed in continuous space (they could have occurred anywhere in the city). A common analysis objective for spatial point processes is the determination of areas of unexpected concentrations or clusters (Waller and Gotway, 2004).

Well-known modern techniques for describing and delineating clusters include the scan statistic, hierarchical clustering, and kernel density estimation (Waller and Gotway, 2004; Levine and Associates, 2007). Limitations of these techniques are that they either impose shape biases or they are incapable of extracting explicit spatial boundaries (i.e., are continuous and do not generate event/non-event boundaries). For example, the scan statistic represents clustering in terms of minimal bounding ellipses (Kulldorff, 1997) and hierarchical clustering imposes either elliptical or convex shapes (Levine and Associates, 2007). In spatiotemporal analysis, where deformation may be a research interest, use of these techniques may not adequately represent the actual spatial extent of higher-level events. To illustrate the limitations of the scan statistic and

hierarchical clustering to describe the shape of events, consider an example of a higher-level event whose true shape is parabolic. Both the scan statistic and hierarchical clustering approaches would either overestimate the true spatial extent (by including the empty portion of the "U") or become so diluted that the higher-level event is not detected at all. Meanwhile, kernel density estimation (KDE) does not impose any explicit bias in terms of shape, but describes the relative distribution of lower-level events continuously throughout a study area (Waller and Gotway, 2007). Results from this approach do not produce any explicit extractable representations of higher-level events that could potentially serve as input for the spatiotemporal helix.

1.1.2 Representing Higher-Level Events from Geosensor Data

As with spatiotemporal point process data, a common research objective with geosensor network data in point form is the representation of the areal extent of higher level events. However, important distinctions exist between these two types of spatiotemporal point data which have direct implications on how higher-level events can be extracted. In point process data, points arise from binary fields, are discrete, and each point represents a low-level event (e.g., an instance of disease). Meanwhile the rest of the study area, where there are no points, denotes the absence of events. In contrast, point observations from geosensor networks are observations onscalar fields, are real-valued, and can depict either the presence or absence of a low-level event at each sensor location.

Being real-valued, statistical methods are required for the extraction of lower-level events from geosensor readings. These methods range from the relatively simple, such as thresholds, to more complex techniques like multivariate regression (Beard et al.,

2008). The results of these techniques effectively reduce real-valued readings to discrete ones (i.e., lower-level event or non- event). The transformation of a time series to a discrete labeled event that can be represented by one or two bits has added advantages for the performance of sensor nodes which are highly power, compute and communication constrained. (Chintalapudi and Govindan, 2003; Nowak and Mitra, 2003; Nowak, Mitra, et al. 2004; Duckham, Nittel, et al. 2005; Worboys and Duckham, 2006). This transformation also has a conceptual benefit, where the separation of continuously varying fields into homogenous regions implies the existence of salient boundaries delimiting the extent of higher-level events (Duckham, Nittel et al., 2005).

Approaches based on the spatial distribution of the event and non-event readings for the representation of these boundaries of higher-level events have been developed in both the engineering and geographic literature. As is the case with techniques for higher-level areal event extraction in point process data, a limitation of methods for the representation of higher-level events in geosensor data is bias. The majority of these techniques have been either graph (e.g., Duckham, Nittel et al. 2005; Sadeq and Duckham, forthcoming) or gradient-based (e.g., Chintalapudi and Govindan 2003) and all result in linear representations of higher-level areal events. In reality, most events are likely to have complex non-linear forms (e.g., a pollution cloud) and the loss of spatial detail caused by the imposition of linearity could inhibit ability to describe changes in shape over time.

1.2 Thesis Objectives

Objectives for spatiotemporal analysis in "security informatics" were recently summarized in terms of three questions 1) How to identify regions within the study area with high or low concentrations of events? 2) How to determine if any areas of variant concentration are the result of random variation or are statistically significant, and, if the variation is not random are there explanatory variables that can explain this deviation? 3) How to identify significant changes in the distribution of events (Zeng, Chang et al. 2004)? An umbrella term, security informatics covers a diverse collection of research domains including homeland security, law enforcement, and public health among others and can be defined as the application of information technology for the maintenance of public safety and well-being (Zeng, Chang et al. 2004).

The first two of these questions are traditional research objectives in spatial analysis, notably in epidemiology and criminology, and address the need to establish the presence of higher-level events. In the case of point process data, this typically involves statistically testing potential clusters against an expected distribution (e.g., complete spatial randomness). For geosensor data, statistical testing occurs as part of the determination of lower-level events, and the presence of higher-level events is signaled when lower-level events are detected.

The third question describes the central objective of this thesis, the description of deformation of higher-level events over time. With point process data, resolution of the first two questions typically involves the generation of representations of the spatial extent of higher-level events (e.g., with the spatial scan statistic or hierarchical clustering). With geosensor data, this process involves interpolation based on the relative

distribution of sensor nodes with event and non-event readings. In both cases, spatiotemporal analysis of deformation involves examining how these representations of higher-level events change over time.

The bias of current methods for representing the areal extent of higher-level events from both point process and geosensor data limits the ability to precisely describe changes in shape over time. This shortcoming is the motivation for this thesis, and the objective of this research is to investigate the potential of two newly developed machine learning methods for extracting and representing the spatial extent of higher-level events from point data. Machine learning takes advantage of computational power to resolve problems too complex to be solved through explicit programming. Many of these problems involve non-linearity and the derivation of decision functions to interpolate results beyond observations collected at points.

A class of techniques in machine learning, kernel methods, has demonstrated ability to generate non-linear decision functions with attractive generalization properties (Scholkopf and Smola, 2002). These decision functions can be interpreted as delimiting the spatial extent of higher-level events (Ben-Hur et al., 2001; Zeng, Change et al, 2004; Chang and Zeng et al., 2005) which, combined with their non-linearity, suggests that these techniques could be used to describe deformation over time.

Kernel methods rely on transformations to higher, potentially infinite, dimensional non-linear feature spaces. The functions used to conduct these transformations are called kernels, which have the sole requirement of being symmetric and positive semi-definite (Christianini and Shawe-Taylor, 2000; Scholkopf and Smola, 2002). Evaluated at each data point, kernel functions produce what is called a feature

space. This space, can be interpreted in a manner similar to KDE. However, unlike KDE where the final objective is a continuous kernel-based representation of relative density throughout the study area (Waller and Gotway, 2004), kernel methods involve a second computational step that results in the derivation of a decision function in feature space. These decision functions provide a theoretical basis for the extraction of explicit boundaries and can be remapped in the initial domain of the data to delineate higher-level events.

The decision function applied in feature space defines each of the various kernel methods. In the case of support vector clustering (SVC), an unsupervised learning method with only one class of data point (i.e., all points are lower-level events), the decision function is a minimal bounding hypersphere. Results from this algorithm identify point clusters and SVC is thus proposed as a method for spatiotemporal analysis of point process data.

For the geosensor network data considered in this thesis, there are two different classes of data points (i.e., points describing lower-level events and non-event points). The assignment of sensor points to one class or the other is not dissimilar to the process of developing a training set for supervised learning algorithms. Correspondingly, a method for supervised learning, SVMs, is proposed as a means for deriving representations of the boundaries of higher-level events from geosensor data. SVMs implement a separating decision function that divides two classes in feature space with a maximum separating hyperplane. As with SVC, this decision function can be mapped back to the initial spatial domain and be interpreted as a representation of the areal extent

of higher-level events. The focus of this thesis is the deformation of higher-level events derived from either point process or geosensor data. More explicitly the objectives are to:

1. Demonstrate the applicability of SVC and SVMs for the derivation of representations of higher-level events and
2. Critique issues related to their potential for spatiotemporal analysis.

1.3 Thesis Organization

Chapter 2 provides an introduction and description of existing techniques for extracting representations of higher-level events from both point process and geosensor data. Discussion begins with working definitions for events in the context of the two types of point data that are of interest in this research. These definitions motivate selection of different techniques for the extraction of higher-level events. Limitations in these methods are the basis for considering SVMs and SVC.

Chapter 3 introduces the algorithms for SVC and SVMs along with a general background on statistical/machine learning techniques and other kernel methods. Discussion addresses key concepts including margins, optimization, and geometric interpretations. SVMs were developed prior to SVC, and while it may be more natural to discuss SVMs first, the order of presentation is reversed in this chapter.

Chapter 4 provides an interpretation of SVC and SVM results. Kernel density estimation (KDE), an established technique for spatial analysis, is used to interpret the results from SVC. While clustering is a perceptual concept, the types of higher-level events captured in geosensor spatial fields of time series data allow for comparison

against hypothetical "ground truths." To demonstrate the ability of SVMs to produce higher-level event boundaries that follow the distribution of lower-level event geosensor point locations, comparisons are made between SVM and existing techniques. In describing how complex boundaries for higher-level events can be derived using SVC and SVMs, several issues involving their implementation are introduced. Chapter 5 presents these issues along with recommendation for future research.

Chapter 2

RESEARCH CONTEXT

Given the potential for spatiotemporal data to contribute to research in a range of applications, discussion of theoretical issues related to spatiotemporal phenomena has been an active topic and, in the modern era, can be traced back to Clark's (1959, 1962) work involving "geographical change" in spatial patterns over time. More recent advances have examined spatiotemporal autocorrelation (e.g., Knox, 1964; Cliff and Ord, 1981), diffusion, and time geography (e.g, Hagerstrand, 1967; Pred, 1977; Parkes and Thrift, 1980). This early work highlighted the substantial complexities involved in modeling phenomena with both the spatial and temporal dimensions (e.g., Miller, 1991; Hazelton, 1991).

2.1 Events

Increasingly, these modeling difficulties have been addressed conceptually through the definition of events. The spatiotemporal analysis literature provides several working definitions for events. Worboys (2005) compares events to objects and emphasizes the relative transience in time of events compared to objects. Claramaunt and Jiang (2000) describe events as an application driven concept describing patterns of change. Guarlnik and Srivastava (1999) suggest that events involve a qualitative change in dynamic behavior. Peuquet (1994) simply defined events as change in a location or an object. In a recent review and generalization of the literature involving the conceptual nature of events, Beard et al. (2008) characterized the majority of existing definitions for events as involving change localized in space and time.

Furthering their discussion, Beard et al. (2008) refined the definition of events as change units with spatial, temporal, and thematic dimensions. Such a conceptualization facilitates discussion of primitive, or low-level, events as well as higher-level composite events that are aggregations of primitive events. This distinction between primitive and higher-level events is applied in this thesis, where the kernel methods that are introduced for applications in spatiotemporal analysis are suggested as a means to interpolate the spatial extent of higher-level areal events based on the distribution of lower-level point-based events.

In point process data each point represents the location of a lower-level event (e.g. an instance of a crime). The change in such a case is the change from the expected state of no crime. Likewise, in epidemiology, the expected state is the absence of disease and individual cases indicate change from that state with the observation of each case. A common research objective for point process data involves the aggregation of individual low-level events into groups. These aggregations have been called "hot-spots" or "clusters" depending on the field of application (e.g., Braga, 2001; Kulldorff, 1996) and represent higher-level events where the change involves rates of incidence over areas rather than observations at points (i.e.) the change is in the rate of incidence).

The same framework of primitive and higher level events can be used to describe spatiotemporal phenomena detected by geosensor networks. A distinction between point process and geosensor data is that raw geosensor data are observed snapshots on continuous distributions. These observations are typically real-valued and do not necessarily denote the presence of low-level events (e.g., the concentration of a chemical in the air, the moisture content of the soil). Determination of the presence of events from

these real-values typically involves statistical techniques that can range from relatively simple approaches, such as thresholds, to more sophisticated methods such as regression, trend analysis, and Fourier analysis (Beard et al., 2008).[1]

For both point process and geosensor forms of point data, analysis is commonly focused on higher-level areal events rather than lower-level point based events. Correspondingly, a number of techniques have been developed to translate collections of lower-level point events into higher-level areal event. A consistent problem with established techniques for both geosensor network and point process data developed to conduct these translations is that they typically imply some form of bias a priori regarding the shape of higher-level event boundaries (e.g., elliptical/circular shape, convexity, or linearity). This chapter presents, their limitations, and motivates consideration of kernel methods as a means for deriving representations of the boundaries of higher-level areal events based on collections of lower-level point-based events.

2.2 Existing Cluster Extraction Methods for Point Process Data

Literally dozens of clustering techniques have been created to assess clustering within point process spatial data (Waller and Gotway 2004). Much of the literature provides comparative analyses of clustering techniques (e.g. Hill, Ding et al., 2000; Zeng, Chang et al., 2004; Aamodt, Samuelson et al., 2006; Wheeler, 2007). From these comparative analyses and from the volume of applications, a few of dominant clustering techniques have emerged. The following subsections critique three such techniques in

[1] The objective of this research is to present kernel methods for the aggregation of lower-level events, and does not explore the appropriateness of these various techniques for the identification of lower-level events. Rather, the approaches described in this thesis are general enough that they can incorporate any method as long as it can produce labels identifying sensors detecting lower-level events.

order to motivate consideration of the support vector method for cluster analysis appearing in the next chapter.

2.2.1 Hierarchical Clustering

Hierarchical clustering methods are among the oldest clustering techniques. These methods have been widely applied in a range of applications that include pattern analysis and data mining, as well as spatial applications. In spatial applications, hierarchical clustering techniques iterate comparisons of spatial attributes against test criteria. Features that pass one set of tests are successively passed to the next level of the hierarchy until the process exhausts itself. Test criteria include nearest neighbor distance (Johnson, 1967; D'Andrade, 1978), a correlation-based centroid method (King,1967), median clusters (Gowers, 1967), and minimum error (Ward, 1963) among others. Due to their simplicity, hierarchical methods are often used as a basis of comparison against other methods (e.g., Zeng et al., 2004).[2]

Hierarchical techniques offer some advantages in that results can be computed very rapidly, there is no bias a priori regarding the number of component polygons of potential clusters, and statistical inferences can be drawn regarding the significance of clusters. The following discussion describes a version of hierarchical methods that employs nearest neighbor distance as a test criterion.

The nearest neighbor hierarchical clustering algorithm involves iterative comparison of a nearest neighbor test value to a standard error computed under the assumption of complete standard randomness (see Equations 2.1-2.3) and generates

[2] Hierarchical clustering is one of the algorithms included in the CrimeStat package created by Ned Levine and Associates with funding from the Department of Justice. It can be downloaded free of charge from http://www.icpsr.umich.edu/CRIMESTAT/.

hierarchies of clusters that can be represented in a dendogram (see Figure 2.1). The first iteration begins with the computation of a distance matrix for all of the points. For each point, the nearest neighbor is found and the distance between nearest neighbors is compared against the quantity in Equations 2.1-2.3 which is a function of the statistical significance level (α), the area of the study region (A), and the number of points (N).

$$CI_{MeanRandomDist} = \frac{1}{2}\sqrt{\frac{A}{N}} \pm t(SE_{d(ran)}) \tag{2.1}$$

$$CI_{MeanRandomDist} = \frac{1}{2}\sqrt{\frac{A}{N}} \pm t * \sqrt{\frac{(4-\pi)A}{4\pi N^2}} \tag{2.2}$$

$$CI_{MeanRandomDist} = \frac{1}{2}\sqrt{\frac{A}{N}} \pm t * \frac{0.26136}{\sqrt{N^2/A}} \tag{2.3}$$

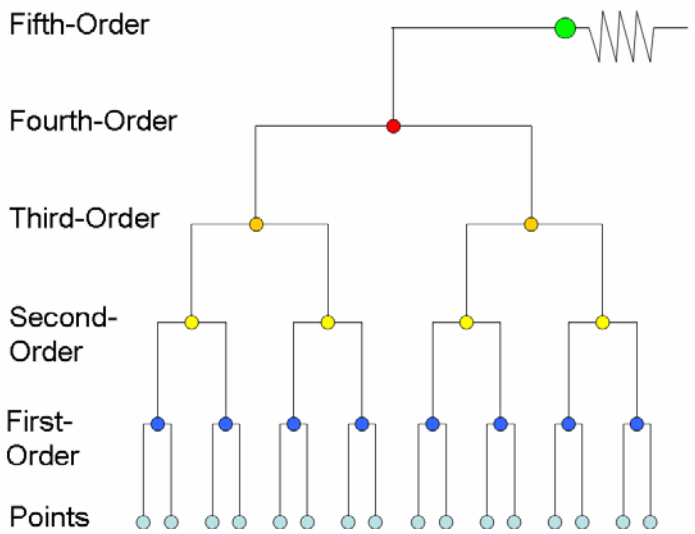

Figure 2.1 Dendogram representation of the hierarchical clustering algorithm for a data set that exhausts itself after 5 iterations. The base represents the input data set. Each step upwards in the diagram represents a test against a given criterion.

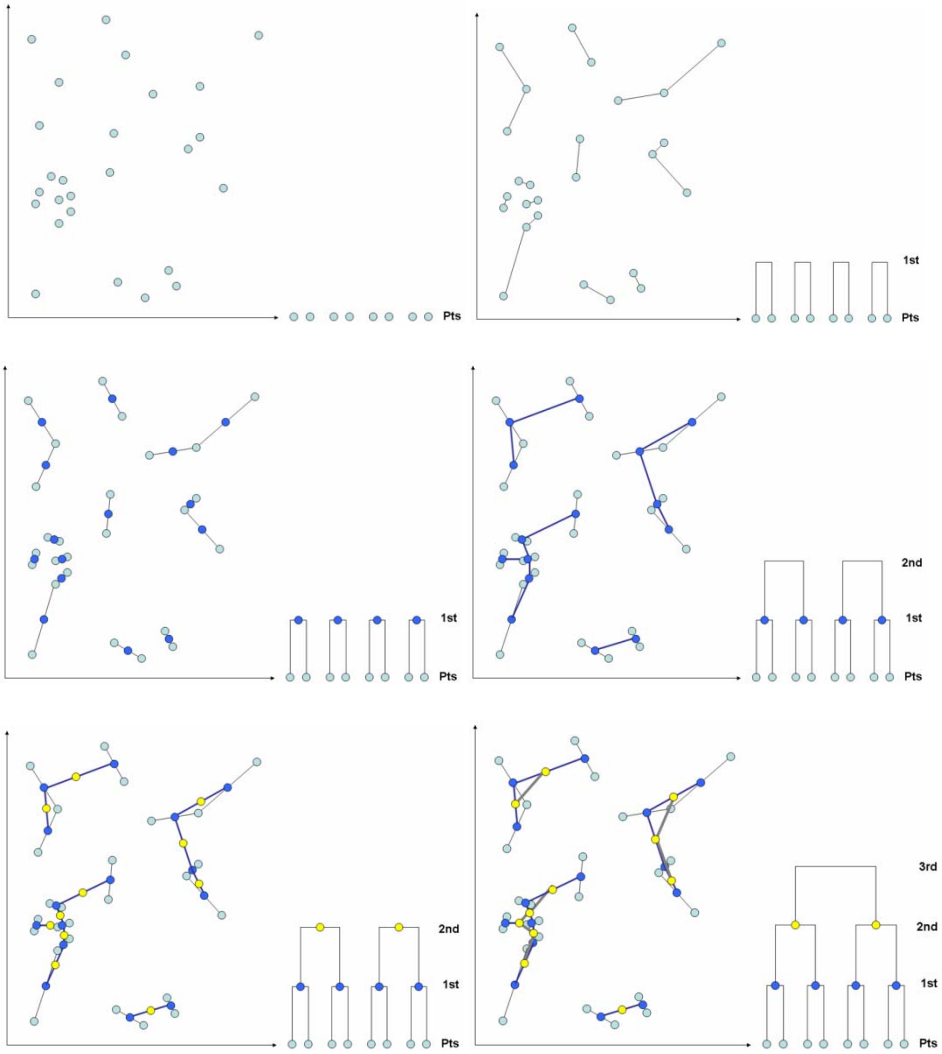

Figure 2.2 Series of frames depicting the hierarchical clustering algorithm with point plots on the left side of each frame and the corresponding view of the dendogram on the right. Starting with the raw point data (frame 1), nearest neighbor distances are compared against values computed using Equation 2.3 (frame 2). Pairs with distances less than the critical value are then used to define a new subset of points which are the midpoints of nearest neighbor distances (frame 3). This subset is then tested with a new critical value reflecting the corresponding number of points in the subset (frame 4). The process is repeated until the either the data are exhausted or the algorithm fails.

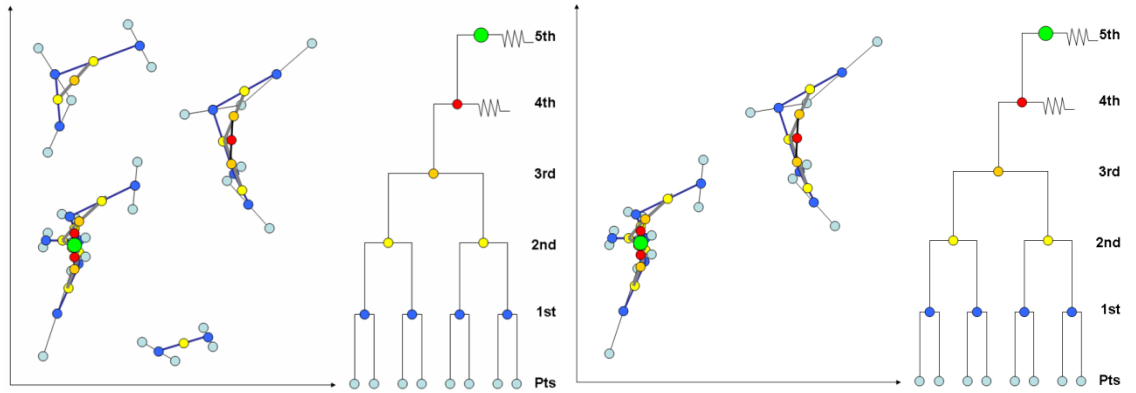

Figure 2.3 Application of user defined minimum cluster size. The concentrations in the upper left and lower right are eliminated because they do not contain enough points.

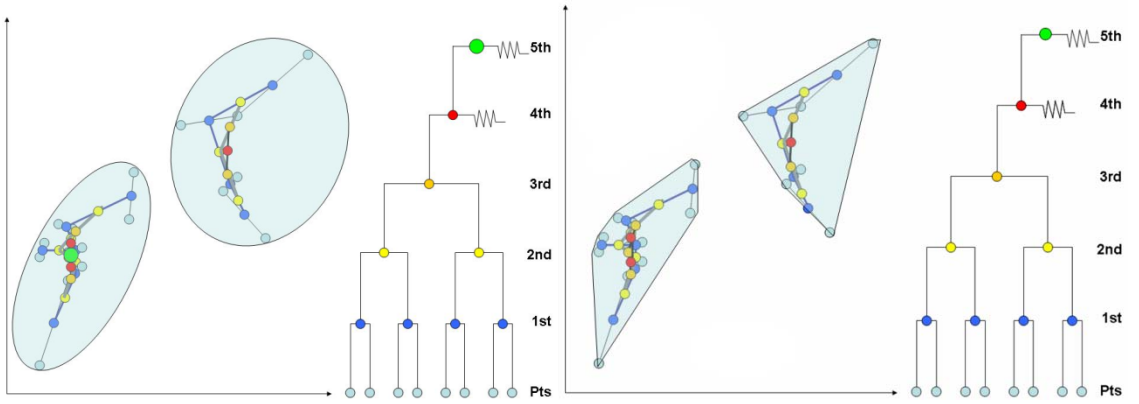

Figure 2.4 Derivation of estimates of the areal extent of clusters from hierarchical methods, either the minimum bounding ellipse or the convex hull can be used.

2.2.2 Scan Statistic

With strong roots in epidemiology, the spatial scan statistic has become a popular, if not the de facto, cluster detection method in a range of disciplines (Waller and Gotway, 2004). Scan statistics, as a class, operate through the use of a scanning window of fixed shape and varying size which exhaustively passes through the study area and generates statistics for potential clusters (Kulldorff, 1997). Like the hierarchical clustering methods, the scan statistic can be used for statistical inference regarding the location of

clusters, does not have any bias a priori regarding the number of clusters, and can be calculated using a freeware package.[3,4]

To scan the study area, a series of windows are generated at each point in the study area. The window size is incrementally, successively capturing the next nearest neighbor, until a pre-determined maximum percentage of the study area has been covered. At each location and for each window size the number of observed instances is counted. Counts are compared against the number of hypothetical instances that would appear in the window under an assumed distribution with the likelihood ratio

$$\lambda = \frac{L(\hat{Z})}{L_0} \tag{2.4}$$

where the numerator represents the number of instances observed and the denominator represents the number of instances expected. Assumptions regarding covariates in the underlying population can be built into the ratio for each window and compared against a test statistic derived from Monte Carlo simulations to estimate the significance of potential clusters (Kulldorff, 1997).[5]

The scan statistic is most commonly applied to analyses of point in polygon data where counts within polygons (e.g., census tracts, counties) are aggregated to some point of central measure (e.g., centroid, county seat). However, the scan statistic can also handle point process data by entering the geographical coordinates for each instance as an

[3] SaTScan is funded by the National Institutes of Health is available for free download from (www.satscan.org/).
[4] Scan statistic results as generated by SaTScan can detect multiple clusters, but the algorithm is designed primarily for the determination of the most likely cluster. As a result, the p-values used for statistical inference are conservatively biased (Kulldorff, 2006).
[5] Comparison against the test statistic is represented by a p-value. Output from the spatial scan statistic is presented in terms of the most likely cluster.

input point (as would be done for a point of central measure) and by assigning equal weight to each instance.

The spatial scan statistic has been a subject of intense study, and since its introduction by Kulldorff in 1997, where it was presented for use with circular scanning windows for assumed Bernoulli and Poisson distributions, its applicability has been greatly extended. Currently, the spatial scan statistic can also operate under assumptions of ordinal (Jung, Kulldorff et al., 2006) and exponential (Huang, Kulldorff et al., 2006) distributions and can use an elliptical shaped scanning window (Kulldorff, Huang et al. 2006). In studies comparing the spatial scan statistic against other cluster detection methods conducted by Aamodt et al. (2006), Zeng et al. (2004), and Song and Kulldorff (2003) the scan statistic proved capable of identifying potential clusters. Song and Kulldorff (2003) noted the scan statistic's high power and usefulness when the size and shape of potential clusters is unknown. Aamodt et al. (2006) also found that the spatial scan statistic performed well in situations where little is known about potential clusters and for clusters with lower relative risks. Zeng et al. (2004) found that the spatial scan statistic was able to determine the location of a cluster in the same area as risk-adjusted hierarchical clustering methods, but concluded that the spatial scan statistic offered less spatial precision due to its bias regarding the shape of clusters.

This often stated limitation of the scan statistic has been countered by Kulldorff's (2006) argument that clusters remain a perceptual construct and that the representation of the general area of the true underlying cluster is often sufficient. While valid in the context of the point-in-polygon type data where spatial aggregation of instances over polygons already diminishes spatial precision, this argument is much weaker in the

context of point process data where the precision made available by the exact locations of instances can be exploited for a much greater degree and used for spatiotemporal analysis of deformation.

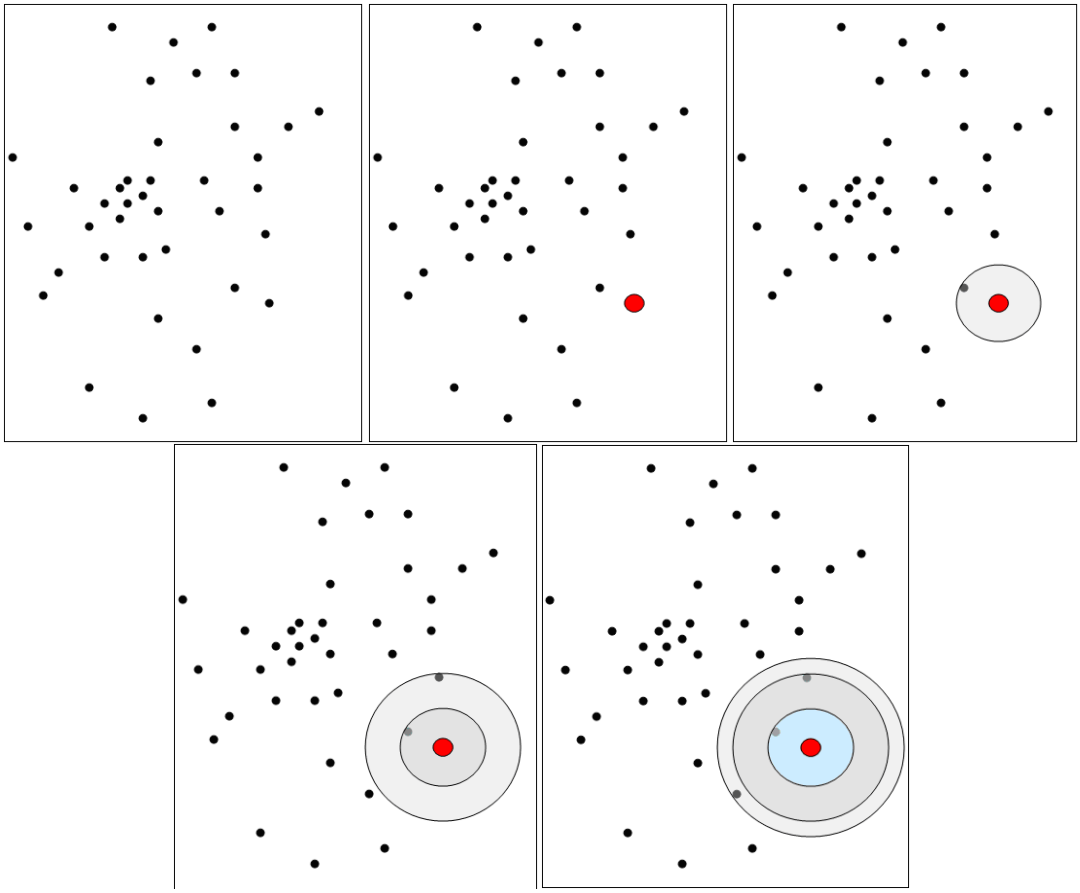

Figure 2.5 An illustration of the spatial scan statistic. Scanning windows successively increase in size and counts within each window are compared against the expected for each size at each location. The p-values corresponding to each window (from likelihood ratio in Eq. 2.4) can be used to determine the most likely clusters. The spatial extent of clusters is estimated as being those polygons which house the points corresponding to the most likely scans. In point process data, the spatial extent of the window is estimated as the window itself.

2.2.3 Kernel Density Estimation

Kernel density estimation (KDE) describes the probability density of events $f(s)$ as opposed to the intensity of events $\lambda(s)$. In the context of spatial point processes,

probability density refers to the relative probability of observing an event at a given point *s*. Intensity refers to the number of events expected per unit area at locations.

Correspondingly, KDE integrates to one while kernel intensity estimation would integrate to the mean number of events per unit area. The distinction between these two methods is a difference of proportionality that stretches or shrinks peaks, but does so proportionally and generates a consistent relative representation of the distribution of events regardless of whether kernel density or kernel intensity is modeled. The kernel density function can be derived from the kernel intensity function simply by dividing kernel values by the total area. Due to the similarity of the naming conventions and the results for these two methods, the use of terminology has been inconsistent in the literature (Waller and Gotway, 2004). This section describes kernel density estimation (KDE).

KDE has been widely used in spatial applications to generate smooth representations of the distribution of low-level point process events throughout a study area. Instead of being a means for the derivation of representations for the spatial extent of clusters (as was the case with the hierarchical clustering and scan statistic examples already presented), KDE provides continuous description of the relative density of event instances throughout an entire study area, and is more a descriptive than extractive method. KDE is discussed in this review of clustering techniques because, despite not providing polygonal representations of clusters, KDE does address some of the limitations highlighted in the discussion of hierarchical clustering and the scan statistic – especially those related to a lack of spatial precision. In addition, through the concept of

kernels, KDE creates a conceptual bridge to the introduction of the support vector techniques in Chapter 4.

Kernels are weighting functions often used in non-parametric (Bayesian) estimation. The sole requirements placed upon functions to be considered for kernels is that they be symmetric about the origin and integrate to 1 (Waller, 2004). The motivation behind their use for density estimation in spatial point processes is that kernels can be used to provide smoothed representation of point distributions' probability functions and therefore are useful in visualizing both hot and cold spots within study areas. The functional form for KDE (under the assumption of no correlation between the x and y directions) is

$$\hat{f}_h(x) = \frac{1}{Nh_x h_y} \sum_{i=1}^{N} K\left(\frac{x - x_i}{h_x}\right) K\left(\frac{y - y_i}{h_y}\right) \tag{2.5}$$

where N is the number of points, h_x and h_y are smoothing parameters (bandwidths), and K represents a symmetric function with the property that

$$\int_D K(s)ds = 1. \tag{2.6}$$

A variety of different kernels have been applied with radial basis functions with the Gaussian and the quartic being the most popular (see Table 2.1).

Table 2.1 One Dimensional Versions of Two Kernels Common in Spatial Applications

Kernel	Functional Form	Selected Applications
Gaussian	$\frac{1}{\sqrt{2\pi h}}\exp\left[-\left(\frac{x-x_i}{2h}\right)^2\right]$	(Baxter, Beardah et al. 1997; Kelsall and Diggle 2007)
Quartic (Biweight)	$\left(\frac{15}{16h}\right)\left[1-\left(\frac{x-x_i}{h}\right)^2\right]^2 I\left(\frac{\lvert x-x_i\rvert}{h}\le 1\right)$	(Seaman and Powell 1996; Berke 2004)

Intuitively, the concept of KDE is not entirely dissimilar to that of the scan statistic in that, like the scan statistic's scanning window, a density function defined by the appropriate kernel centers itself on each point in the distribution and measures the relative distance from that point to the rest of the distribution. However, unlike the scan statistic, the entire surface of the study area, rather than just the area covered by a scanning window, is assigned a value reflecting the density of instances for all points in the study area. In the case of radial basis functions, such as the Gaussian the relative weight of values diminishes in all directions with distance from the point upon which the kernel is centered.

In areas with tight concentrations of points, the values from kernels will overlap, and it is through this overlap that KDE represents areas of high or low concentrations (see Figures 2.6 and 2.7). Determining the extent of overlap, is the bandwidth parameter *h*, which can be varied to provide more smooth or rough (large or small bandwidth values) representations of density. While the determination of an appropriate value for this parameter is an open research topic (e.g., Wand and Jones 1995) and one that is always subject to the nature of the application, several methods have emerged to suggest

bandwidth values. Among these methods is asymptotic mean squared error (AMISE) which is defined as the limit of the expected value of

$$\int \left[\bar{\lambda}(s) - \lambda(s)\right]^2 ds \tag{2.7}$$

as the sample size goes to infinity. From an expansion of the components of AMISE, a relatively simple selection rule is

$$\hat{h}_u = \hat{\sigma}_u N^{-1/(\dim+4)} \tag{2.8}$$

Where $\hat{\sigma}_u$ is the sample standard deviation of the coordinates in the *u*-dimension (e.g., *x* or *y*), *N* is the number of instances in the study area, and *dim* is the number of dimensions in the data (Scott, 1992).

By summing all of the corresponding kernel values from every instance location in the study space, regions with tighter concentrations have higher density values and regions without points will have relatively low values. Normalizing these values by the number of points in the study area bounds the range of values throughout the study area between 0 and 1, and therefore provides a representation of probability density. Kernel density estimation is relatively easy to program (as was done by the author in Matlab to create Figures 2.6 and 2.7), An application is also available as part of the freeware package CrimeStat.

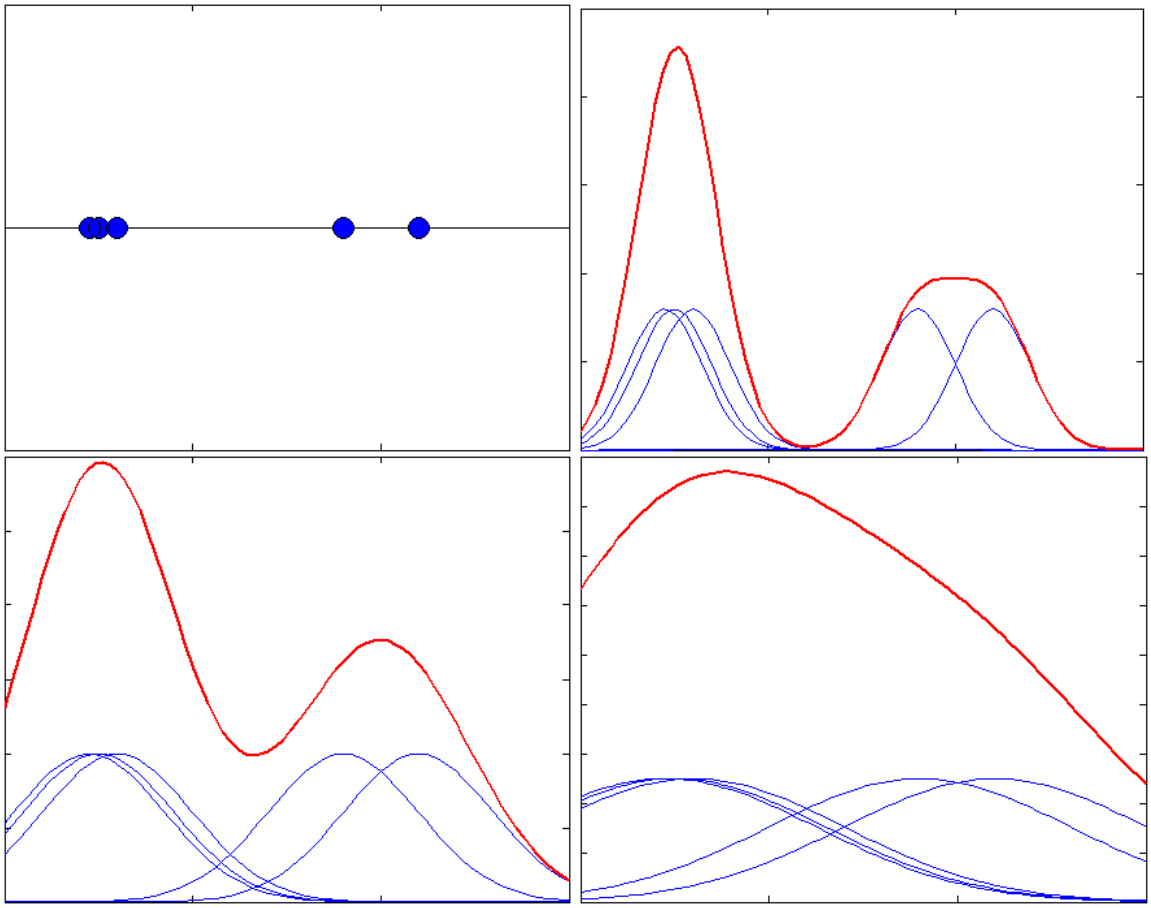

Figure 2.6 One dimensional kernel density example. The first frame shows the input data. The second, third, and fourth frames show kernel density estimates from Gaussian kernels with bandwidths 1, 2, and 4 respectively.

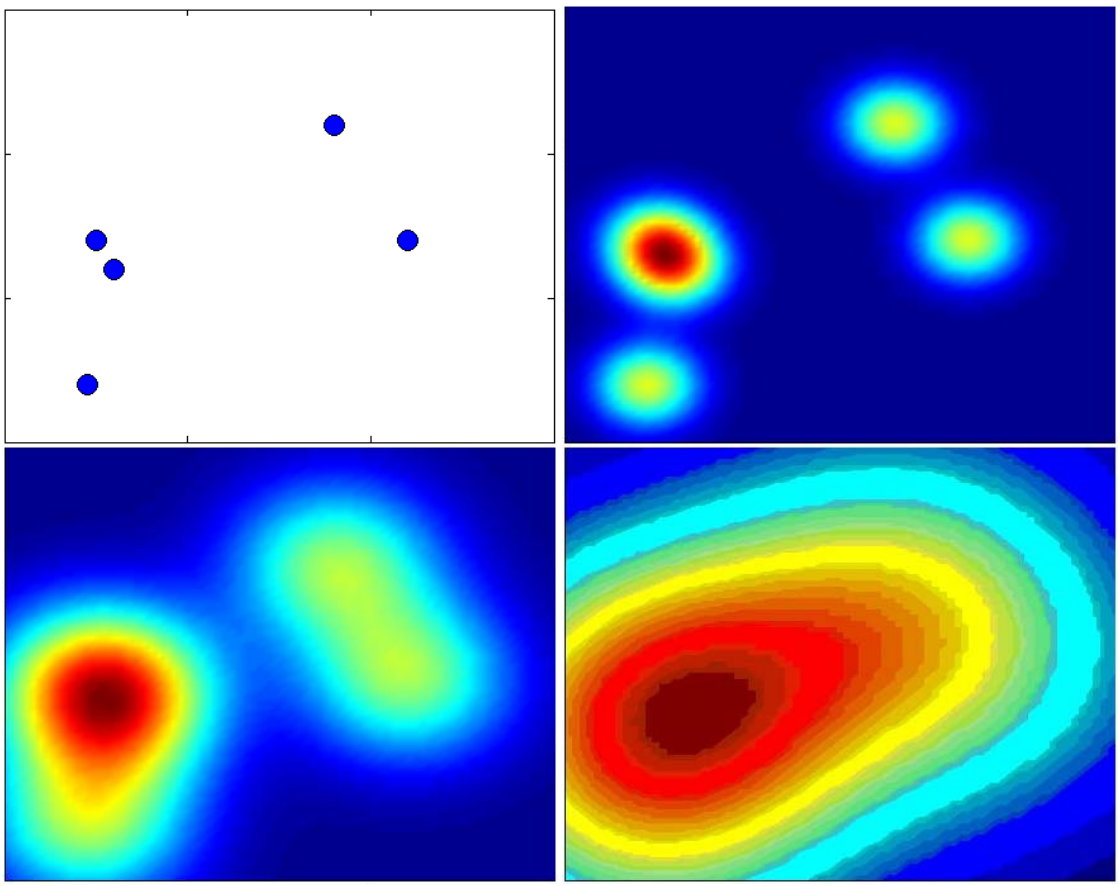

Figure 2.7 Two dimensional kernel density example. As in Figure 2.6, the first frame shows the input data, and the second, third, and fourth frames show kernel density estimates resulting from Gaussian kernels with bandwidths 1, 2, and 4.

2.2.4 Limitations of Current Methods for the Estimation of the Spatial Extent of Clustering Events

While far from a complete review of available clustering techniques, the previous discussion illustrates some of the limitations of some of the most popular methods for the estimation of the spatial extent for aggregates of point-based events. One major limitation is the lack of spatial precision. Both hierarchical clustering and the scan statistic rely on the spatially simplistic convex hull and minimum bounding ellipse, for estimating the spatial extents.

Even considering that the exact spatial extent of point-based aggregate events remains unknown and that their definition likely varies with the application focus, these

methods likely limit accuracy. For example, when the minimum bounding ellipse is used and the true shape of the cluster is parabolic, the ellipse could suggest a larger area than is warranted (due to the inclusion of the inside of the "U"). At the same time, the identification of such a cluster is problematic since the concentration described by the cluster would be diluted due to the inclusion of an area relatively devoid of instances (again the inside of the "U").

Representing parabolic clusters with convex hulls, creates similar representational problems. As with minimum bounding ellipses, the convex hull could distort the representation of clusters to include regions that are not actually part of the cluster (e.g., the portion inside the "U") or, with sufficient area, could dilute the true cluster to the point that it is no longer extractable.

Kernel density estimation can represent such spillovers in terms of relative concentration and can do so with complex shapes that follow the distribution of instances (see Figures 2.7 and 2.8). Kernel density estimation, however does not explicitly construct a boundary representation. The support vector clustering methods presented in the following section provide a means for delimiting discrete regions from kernel-based transformations.

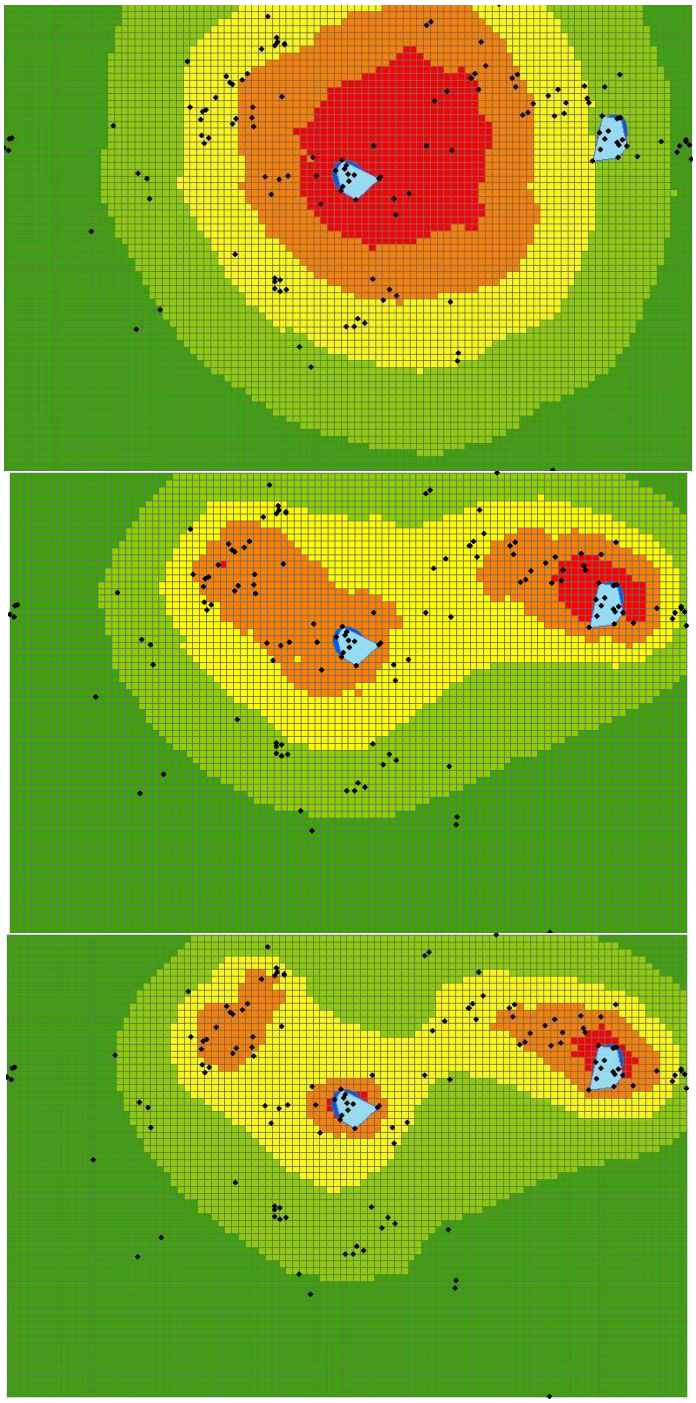

Figure 2.8 Kernel density (color ramp from green to red) and hierarchical cluster/hot spot representations (convex hulls in blue). Each of the frames show kernel density with different bandwidths. The top frame has the widest bandwidth and the bottom frame has the smallestThe first frame shows kernel density estimates. The second and third frames show the same kernel result overlaid by hierarchical clustering results corresponding to convex hull (frame 2) and minimum bounding ellipses (frame 3).

2.3 Geosensor Networks and Dynamic Spatial Scalar Fields

With the rapid advance of wireless sensor technology, geosensor networks have become an important source of spatiotemporal data. Representing real-valued observations collected at sensor location points, the data produced by these systems have been described as arising from *dynamic spatial scalar fields*. Formally, a dynamic spatial scalar field is defined in terms of scalar values V that vary according to an unknown assignment function f that is dynamic in terms of both the spatial S and temporal T domains. As such, a dynamic spatial scalar field can be written $f : T \rightarrow S \rightarrow V$ where S represents the spatial domain and T depicts the temporal domain (Duckham, Nittel et al., 2005).

A common research objective with this form of data is the extraction of estimates for the areal extent of higher-level events (e.g., wildfires, propagation of a toxic chemical) based on the observations collected at sensor points. Approaches to this problem have been developed from both engineering (Chintalapudi and Govindan, 2003, Nowak and Mitra, 2003, Nowak et al., 2004) and geographic perspectives (e.g., Worboys and Duckham, 2006, Duckham et al., 2005). In both cases the methods involve the discretization of the real-valued observations collected at sensor points into qualitative values.

Discretization at the sensor node level implies benefits both theoretically and in terms of network design. By assigning qualitative values at sensor nodes according to observations relative to application appropriate thresholds, salient event *boundaries* can be inferred between neighboring nodes with differing qualitative values.[6] Throughout the

[6] In the engineering literature these discontinuities have been called "edges" and have been differentiated from methods describing isolines, or contours, by discretization (Chintalapudi and Gonvidan, 2003).

study area, these boundaries can be used to define the spatial extent of events. Another advantage of discretization in geosensor networks relates to power consumption in wireless geosensor networks. With strict power constraints and communication being a primary power drain, reduction of the quantity of information to just a few qualitative bits reduces expensive communication time and therefore can result in significant power savings (Chintalapudi and Govindan 2003; Duckham, Nittel et al. 2005). An illustrative example of discretization is that made to a toxic air quality variable monitored by a geosensor network. The real-valued concentration observed at each sensor can be classified as either hazardous or benign according to established health standards. Consequently, instead of reporting the concentration observed at each geosensor node, all that needs to be transmitted is whether or not the reading at each location is in-event (toxic) or non-event (benign).

Relative to the volume of research related to engineering and deployment schemes for geosensor networks, little research attention has been devoted to the extraction of event boundaries in geosensor networks. In an early investigation of algorithms for the estimation of event boundaries, Chintalapudi and Govindan (2003) examined estimators derived from bases in statistics, image analysis, and classification. Based on their investigation, the most attractive of these methods used a classification scheme that compared qualitative values within neighborhoods and derived straight line estimates representing boundary portions that successfully separated sensor reading classes. Methods proposed in the wake of Chintalapudi and Govindan (2003) include those based on network trees (Nowak and Mitra, 2003) and triangulation (Worboys and Duckham, 2006). Major drawbacks of the algorithm proposed by Nowak and Mitra

(2003) include potentially poor spatial characteristics of the tree structure (i.e., neighboring nodes not appearing on the same branch) and resource demands required to maintain tree structure in the presence of mobility. Worboys and Duckham (2006) proposed a more flexible framework based on planar triangulation of sensor nodes. Under such a configuration, the spatial extent of events is modeled in terms of the qualitative values of the vertices of the resulting triangles where triangles containing at least one in-event node are considered as being in-event. While being less intensive from a computational standpoint, this method, as with other existing methods results in linear approximations for the spatial extent of events.

Given that environmental phenomena (e.g., chemical plumes) likely do not have linear boundaries, reliance on coarse, linear, approximations for the spatial extent of events can limit the spatial precision for any subsequent spatiotemporal analysis. The only known attempt to address non-linearity has been the work of Nowak, Mitra et al (2004) and their platelet-based algorithm relies on the highly restrictive and unrealistic assumption of a uniformly distributed geosensor network.

To address non-linearity in the boundaries of environmental phenomena, this research proposes a SVM-based approach for representing the spatial extent of events identified in geosensor network data. With their reliance on the maximum margin separator, a relatively simple method for classification, SVMs are efficient from a computational standpoint while simultaneously offering attractive and well-understood statistical properties (Christianini and Shawe-Taylor 2000; Scholkopf and Smola 2002; Shawe-Taylor and Christianini 2004). Non-linearity is achieved by derivation of these classifiers in a high-dimensional non-linear feature space which can be adapted to

incorporate technical constraints related to the sensors (i.e., sensing radius) and the geosensor network (i.e., sparseness of nodes). The following chapter presents the SVC and SVM algorithms for the estimation of the spatial extent of higher level events. Chapter 4 presents results from simulations as well as visual interpretations of the algorithms.

Chapter 3

SUPPORT VECTOR ALGORITHMS

The previous chapter outlined some of the limitations of current techniques for the estimation of the spatial extent of point-based events. This chapter describes support vector machines (SVMs) and support vector clustering (SVC) and begins with a brief history of the development of these methods before describing the methods themselves in detail. Section 3.1 presents the domain background from which support vector methods evolved. Section 3.2 and 3.3 introduce kernel methods and several definitions that are useful for the discussion of SVMs in Sections 3.4 and 3.5 as well as the description of SVC in Section 3.6 and 3.7.

3.1 Machine Learning

Advances in technology have enabled attention to increasingly sophisticated problems. The field of machine learning, in particular, addresses numerous complex problems and allows computers to solve problems without explicit programming. Machine learning defines learning as the process of acquiring knowledge through experience (Valiant, 1984), an apporach thought to be similar to the way that humans learn.[7] For machines the process of learning has been described in terms of a task (e.g., playing chess), a performance measure (e.g., winning), and a training experience (e.g., playing a game). As the machine repetitively completes the task under varying training experiences, it can be made to analyze its performance so that results improve with

[7] An illustrative analogy could be that of dogs trained to sniff. Given the strength of their olfactory senses, it has been suggested that how dogs perceive their environment is incomprehensible to humans. Nonetheless, through training examples humans can teach dogs to effectively use their sense of smell to find drugs, bombs, and trapped individuals.

practice (Mitchell, 1994). In this analysis, as in many other applications of machine learning, the learning task involves the prediction of an unknown feature of the data (i.e., where is the boundary of a higher-level event) given a set of data values (i.e., the lower level point-based events).

To describe the learning process and establish the symbology used throughout the remainder of this thesis, $x \in X$ refers to the *input* or *training data* and y a vector of *labels* associated with the input data points. As methods for machine learning have developed, several categories of learning have emerged, notably those of *supervised* and *unsupervised learning*. The existence of labels y in the input data distinguishes these two classes. Supervised learning is characterized by the presence of labels, y, associated with each input point and the learning process in supervised scenarios can be written in the form

$$f(x, y) = L(y, g(x)) \tag{3.1}$$

where g is referred to as the *decision function* and L as the *loss function*. When the true process is well approximated, the values generated by the prediction function will closely resemble the input labels and the loss function will approximate zero. As such, when the prediction function is well-formulated, it will have solid generalization properties and novel points (points outside the input data set) with unknown labels are more likely to be appropriately classified. SVMs are a supervised learning method with proven generalization properties that use a boundary as the prediction function (Shawe-Taylor, 2004). This boundary, when applied to a scenario such as that outlined for geosensors in the previous chapter, can be interpreted as representing the spatial extent of events as

captured by geosensor networks. This approach is, in part, the motivation for this research and is outlined in the next chapter.

In contrast to supervised learning techniques, unsupervised learning methods do not involve labels for the input data. Rather, unsupervised learning scenarios involve the distribution of the input data *x* alone. An obvious example of such an application would be that of clustering where the objective is the identification of concentrations of data points. Another example would be the identification of outliers. SVC is an unsupervised learning method designed for exactly such situations, and like SVMs, uses a boundary as the decision function. Also like SVMs, with geographic input data these boundaries can be interpreted as delimiting events and therefore can be used for spatiotemporal analysis. A description of SVC follows that of SVMs and an illustration of how SVC-generated results can be used for spatiotemporal analysis is also provided in the next chapter.

3.2 Development of Kernel Methods

Given that both SVMs and SVC are kernel-based methods, prior to discussion of either SVMs or SVC, kernel methods as an algorithmic class are introduced. These methods evolved out of the "non-linear revolution" that occurred in the pattern recognition literature which was enabled by the rapid advance of computing and processing power in the 1980s. These technological advances allowed far greater complexity to be modeled and for non-linear patterns to be identified with the advent of methods such as backpropagation in multi-layer neural-networks. While these new methods enabled entirely new fields including data-mining and bioinformatics, being largely based on gradient descent or greedy heuristics, they also suffered from local

minima. Due to their complexity they are still not completely understood from a statistical perspective (Shawe-Taylor, 2004).

To address these issues, kernel-based methods emerged in the early 1990s. These models have been described as being modular in that they are composed of two independent components. The first of these is a transformation from the space of the input data to a more complex, higher-dimensional, feature space. The second involves the application of established, relatively simple, means for describing the data in the feature space. Transformation through kernels allows attractive computational shortcuts and enables highly non-linear learning. Meanwhile, reliance on well-established and understood, linear methods or other relatively simple analytical methods in feature space facilitate efficient analysis while simultaneously addressing obstacles of local minima and overfitting. The following section provides a brief discussion of the basic concepts behind current kernel methods and a context for the introduction of SVMs and SVC.

3.3 Kernel Methods

A means of converting a linear classification algorithm to a non-linear scenario involves the addition of non-linear attributes to the algorithm that are non-linear functions of the original data. A common application of similar logic is in regression where the hypothesized relationship between the dependent and explanatory variables is non-linear (e.g., parabolic). Instead of relying upon a more complicated non-linear model, this relationship can be simply modeled using ordinary least squares by first transforming the explanatory variable (e.g., squaring the explanatory variable) and then performing a simple ordinary least squares regression.

This idea lies at the heart of all kernel methods where the input data set, *X*, is mapped to a higher-dimensional *feature space*, $F = \Phi(x_i) : x_i \in X$, where relatively simple methods can be used to find boundaries that can be re-projected back into input space and interpreted as decision boundaries. A simple illustrative example adapted from Scholkopf and Smola (2001) is presented in Figure 3.1. When combined with the ability of separating hyperplanes to be written in a dual form, in terms of Lagrangian multipliers rather than in the primal form, and with the interpretation of kernels as representations of inner products, thereby eliminating the need to explicitly determine feature spaces, this allows for a significant computational savings and is known as the "kernel trick" (Scholkopf and Smola, 2001). Capitalizing on this kernel trick are various methods that employ different feature space decision functions for different application objectives. Examples of such techniques include the SVM and SVC methods that are the basis for this research.[8]

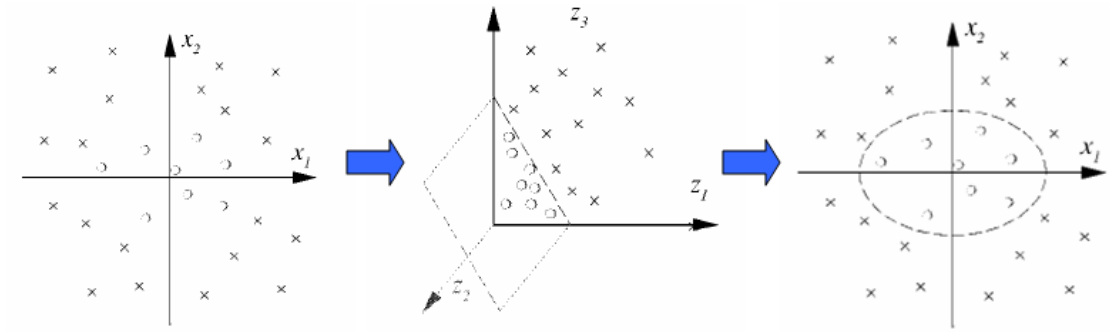

Figure 3.1. A representation of kernel based learning. The first frame shows an example of input data that would require a non-linear decision function. Given that derivation of such non-linear boundaries directly from this input space could be complex, the data are transformed into a higher-dimensional feature space $\left(x_1^2, \sqrt{2}x_1x_2, x_2^2\right)$ where more simple means (i.e., the linear hyperplane shown in the second frame) can be used to derive a

[8] Another kernel-based technique, kernel principal components analysis, has seen application in remote sensing where it has been used to classify features in image-based data (e.g., Yang, Tan et al. 2006).

boundary. This boundary can then be re-projected back into input space (frame 3) and used for classification of novel points.

However, for kernels to offer these attractive properties, they must satisfy certain conditions. Kernels are defined as a function K so that for all $x, z \in X$

$$K(x,z) = \langle \phi(x) \cdot \phi(z) \rangle \tag{3.2}$$

where ϕ is a non-linear mapping from an input space X to the feature space F. To be considered a kernel function for the kernel trick, a function $K(x,z)$ must be symmetric and any finite subset X of the Gram matrix[9] K must follow Mercer's theorem, implying that it must be positive semi-definite (Christianini, 2000). This ensures a unique solution to the problem of determining a minimum for the convex cost problem. Functions fulfilling these requirements are called Mercer's kernels but for brevity in this research, as is commonly done in the literature, they are simply referred to as kernels.

3.4 Support Vector Machines

Support vector machines (SVMs) are the most well-known and widely used class of kernel methods. SVMs use maximum margin linear separating hyperplanes in feature space provide a sparse solution decision boundary with the absence of local minima. Use of the "kernel trick" avoids potentially very complex calculations that may have been necessary to calculate such decision boundaries directly from input space and make these methods very attractive for a range of applications (Christianini, 2000).

Although most of the component mathematical concepts behind SVMs have been in existence for decades, Boser et al. combined them in 1992 to establish SVMs as a

[9] The Gram matrix is the symmetric positive semi-definite matrix of inner products *X'X*.

classification tool. Among these existing concepts includes, perhaps most notably, those of large margin hyperplanes (Duda, 1973) and the geometric interpretation of kernels as inner products in a feature space (Aizermann, 1964). Other component concepts include optimization techniques (Mangasarian, 1965) and slack variables (Smith, 1968; Bennett, 1992). Since their inception, SVMs have become widely applied in a range of application domains (Christianini, 2000; Bennett, 2000) and have been shown to offer very attractive generalization properties both mathematically (Vapnik, 1995; Devroye, 1996; Pontil, 1998) and in practice. SVMs have outperformed a variety of other techniques including application-tailored artificial neural networks (e.g., Scholkopf et al., 1997, Yang and Liu, 1999, Pontil and Verri, 1998).

As a kernel method, the first step in developing a SVM is to determine an appropriate kernel function and the corresponding transformation of the input data into a feature space. In practice, the selection of an appropriate kernel should be guided by domain knowledge relevant to the application and automated in supervised learning situations through comparison of predicted results against labels in the training set (Christianini, 2000). A list of commonly used kernels appears in Table 3.1.

In feature space, a maximum margin linear discriminating hyperplane (or other simple discriminants) can be found which is often less complex to calculate than its image would be in input space (see Figure 3.1).[10] Defining the margin is a subset of the transformed data points. These points are the *support vectors* and can be interpreted as the data points with the highest relevant information content. Indeed, consideration of

[10] For linearly inseparable cases, the margin can be made tolerant to outliers, or "soft," through the use of slack variables. This scenario, which is considered to be much more realistic assumption with real-world data, is described in a following section. The simple linearly separable SVM is described here first to help introduce the reader to base SVM concepts before introducing complications.

the support vectors alone can be used to derive the dividing hyperplane and this sparseness of the solution allows SVMs to scale efficiently to very large datasets. Once determined, this hyperplane can then be re-projected back into input space where it can be used to classify novel points through its interpretation as a boundary.

Table 3.1. Commonly Used Kernels[11]

Kernel	**Functional Form**
Polynomial	$(x_i \cdot x_j + 1)^d$
Gaussian	$\exp\left(-\frac{\|x_i - x_j\|^2}{2\sigma}\right)$
Three Layer Neural Network	$\frac{1}{1+\exp(vx_i x_j' - a)}$

SVMs are often used in supervised learning situations to suggest labels for novel points according the novel point's position relative to the decision function in input space. In this research, where an objective is to estimate the spatial extent of spatiotemporal events, interest is focused on the decision function itself. Given that the decision function gives an indication of the boundary between the in-event and non-event regions in input space, the decision function can be considered as an estimate of the spatial extent of an event at a given time (i.e., the entire study area can be considered as a set of novel points). By considering different point readings, and correspondingly different boundaries, at different points in time, SVMs provide a means of extrapolating the areal extent of spatiotemporal events at different time instances.

A potential application of SVMs with this interpretation would be geosensor network monitoring of air pollution (as described in Chapter 2). Some of the sensors may

[11] The three layer neural network kernel is only a Mercer kernel under certain conditions (Abe, 2005).

detect readings hazardous to human health and could be assigned a label of +1 while those with benign readings could be assigned the label -1 (i.e., a Boolean event predicate is applied). These labels, along with geographic information regarding the sensor locations, could then be fed into a SVM which would produce an estimate for the spatial extent of the event. These areal estimates could then be input into the spatiotemporal helix for comparison and analysis through time.

To illustrate how these methods could be put into practice, a simulated example is presented in the next chapter. The following subsection presents the algorithm and basic theory behind SVMs with a level of detail sufficient for the research purposes of this thesis. For more detailed and technical discussions of SVMs the reader is referred to Christianini and Shawe-Taylor (2000) and Vapnik (1998).

3.5 Support Vector Machine Concepts and Algorithm

The first SVM to be introduced was the binary maximal margin classifier (Boser, 1992). This SVM is characterized by its maximization of the geometric margin of linearly separable data depicting two classes in feature space. While the use of a maximum margin discriminant offers advantageous generalization properties (Bartlett, 1998; Shawe-Taylor, 1998), the assumption of linear separability will always produce a perfectly consistent hypothesis (no training error) for the decision function. As a result, the corresponding hypothesis for a decision boundary can easily be seen as having potential for overfitting data and therefore can defeat some of the attractive generalization principles which serve as a motivation for the use of SVMs in the first place. Therefore, a "soft margin" SVM algorithm employing slack variables for soft margin optimization is more commonly used (Christianini, 2000). Description of this technique appears in

Section 3.5.5 after the basic concepts related to SVMs have been established through discussion of the "hard-margin" linearly separable case.

To further clarify discussion, this introduction to the hard-margin maximum margin binary SVM algorithm is broken down into four subsections: margins, generalization, geometric interpretation, and optimization. Throughout this discussion $x_i \in \Re^n$ refers to a data point, y_i the label assigned each particular point (e.g., toxic or benign), w (weight vector) the scalar orientation of the hyperplane, and b (bias) the offset of the hyperplane from the origin. As such, the classification problem can be written $f(x_i) = sign(w \cdot x_i - b)$ where the optimal hyperplane correctly assigns the labels y_i.

3.5.1 Margins

Use of hyperplanes as decision functions has a history in both statistics, where hyperplanes are referred to as linear discriminants, and neural network research, where they have been called perceptrons. Regardless of the discipline, the underlying theory can be traced back to Fisher (1952). In the fifties and sixties neural network researchers, Rosenblatt (1956) in particular, began to propose iterative approaches for separating points with different labels using linear hyperplanes.

This early iterative approach is of interest not only historically, but also because many of the current methods for deriving hyperplanes in modern SVMs have strong roots in these early techniques. As a reference, the perceptron algorithm, as presented in Christianini and Shawe-Taylor (2000), is shown in Table 3.1. Starting with initial values for w and b, these parameters are updated according to the classification error in each iteration, until no errors are present. This system is guaranteed to converge in data that are linearly separable (Novikoff, 1962).

To clarify later discussion, several definitions are introduced here that address differences between Rosenblatt's (1956) early work and support vector machines in their current state. Firstly, the *functional margin of an example* with respect to a hyperplane is defined as the quantity

$$\gamma_i = y_i\left(\langle w \cdot x \rangle + b\right) \tag{3.3}$$

which will be positive for each point that is correctly classified and negative for each example that is misclassified. The *functional margin distribution of a hyperplane* is the distribution of the functional margins for the examples composing an entire training set with the *functional margin of the hyperplane with respect to a training set S* being the

distribution's minimum value. Note that in the original perceptron algorithm (Table 3.2), the functional margin is used to derive a separating hyperplane. By converging to a hyperplane solution that simply separates the data according to their labels, an infinite number of potential solutions exist and the selection of starting values for the parameters can influence the result (see Figure 3.2). This is because the problem as stated in the perceptron is ill-posed (no unique solution).

Given a linearly separable training set S and learning rate $\eta \in \Re^+$
$w_0 \leftarrow 0; b_0 \leftarrow 0; k \leftarrow 0$
$R \leftarrow \max_{1 \le i \le \ell} \|x_i\|$
repeat
    for $i = 1$ *to* ℓ
        if $y_i(\langle w_k \cdot x_i \rangle + b_k) \le 0$
            $w_{k+1} \leftarrow w_k + \eta y_i x_i$
            $b_{k+1} \leftarrow b_k + \eta y_i R^2$
            $k \leftarrow k + 1$
        end if
    end for
until no mistakes are made within the for loop
return (w_k, b_k) where k is the number of mistakes

Table 3.2 Perceptron Algorithm. The perceptron algorithm solves for a hyperplane in terms of the functional margin and yields non-unique solutions.

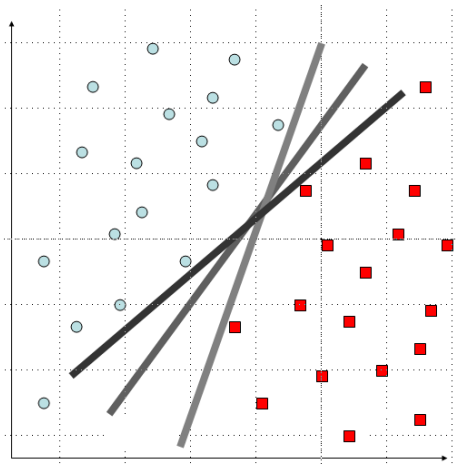

Figure 3.2. Graphical Representation of Potential Results from a Perceptron Algorithm. Since the problem is ill-posed, there is not a unique optimal solution.

Solving for a different quantity renders the problem well-posed so that a unique solution exists. By normalizing with the weight vector w,

$$\left(\frac{1}{\|w\|}w, \frac{1}{\|w\|}b\right), \tag{3.4}$$

a geometric interpretation of the margin, $\gamma = \frac{1}{\|w\|}$, can be obtained (see Figure 3.3). The Euclidean distance of a point to a hyperplane and is called the *geometric margin of an example*. The *geometric margin of a set* is the maximum geometric margin over all hyperplanes. Maximization of the geometric margin of a set yields a unique hyperplane solution. This is the boundary that SVMs.solve for. The following subsections describe this unique solution in greater detail.

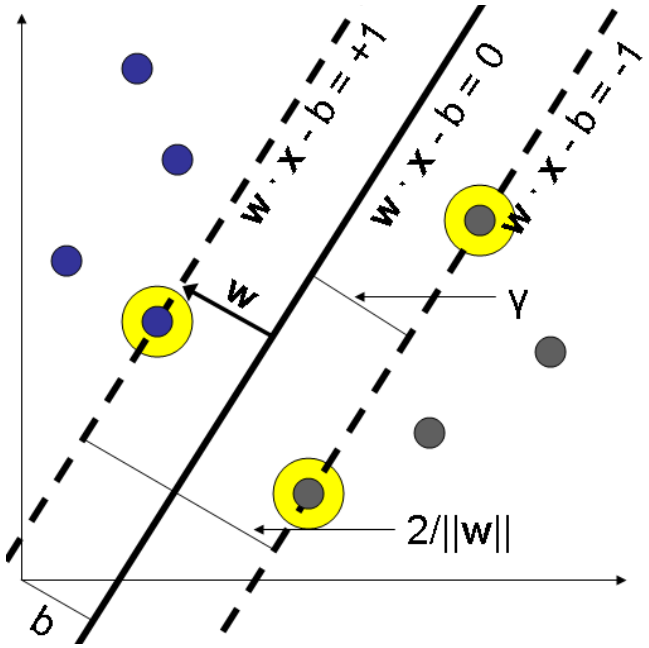

Figure 3.3 Representation of a Maximum Margin Linear Separating Hyperplane. This shows the linear functions for the boundary ($\mathbf{w}\cdot\mathbf{x}$-b=0) and the geometric margin $\gamma=1/\|w\|$. The support vectors are the points located closest to the decision boundary and are highlighted.

3.5.2 Generalization

In a high-dimension feature space, the use of a relatively simple decision boundary (such as a linear hyperplane) ensures that learning remains tractable. However, with the expressive power made available by the high dimensional space and a potentially very large number of degrees of freedom, a sophisticated theory of generalization is required to guide the tradeoff between overfitting and ability to generalize to novel data. By examining the potential separating hyperplanes in Figure 3.2 and 3.3 above, one may intuitively feel that the best hyperplane may be the one that splits the difference between the two classes. This intuition is correct, and can be proven through the use of a theory quantifying the tradeoff between the need to fit the training data well while also having to be general in terms of statistical bounds.

Key to the development of these bounds is Vapnik Chervonenkis (VC) theory which proves that bounds on the generalization error of novel points not included in the training set can be obtained. VC theory proves the existence of a distribution free bound on the generalization of a consistent hypothesis which is a function of the misclassification rate and measures of function capacity[12] for the training set. For linear functions, maximization of the geometric margin reduces functional capacity and therefore minimizes the bounds on generalization. Importantly, these bounds are not directly dependent on the dimensionality of the data and can suggest good performance even in very high-dimensional data. A volume of literature has been developed on theoretical proofs of these generalization properties. Examples of such work includes that by Vapnik (1995), Christianini and Shawe-Taylor (2000), and Devroye, Gyorfi et al. (1996).

While VC theory provides a conceptual basis for the strength of the generalization properties of SVMs, a growing body of literature that has demonstrated the effectiveness of SVMs relative to other techniques. Comparisons have been conducted in a range of applications, and SVMs have perform as well if not better than other commonly applied techniques such as artificial neural networks and even those designed for specific datasets (LeCun, 1995; Yang, 1999; Scholkopf, 1997; Pontil, 1998).

3.5.3 Geometric Interpretations

Having defined the 'best' hyperplane and described some of its attractive generalization properties, this section focuses on the techniques and the geometric interpretation of the solution. There are two geometric interpretations for the hyperplane solution. The first involves the maximization of the distance between two parallel

[12] Capacity refers to the ability of a machine to learn a training set without error.

supporting hyperplanes and the second involves the bisection distance between the points closest to each other on the respective convex hulls of each class in feature space. The interpretations are the dual of one another and the resulting hyperplane is identical regardless of the approach taken.

The first case can be seen as an extension of the perceptron. In Figure 3.2 we can see that there are an infinite number of hyperplanes that can separate the two classes when the functional margin is solved for. To solve for the unique maximum geometric margin hyperplane, pairs of parallel supporting hyperplanes are considered.[13] For a linearly separable set there exist values for *w* and *b* so that $w \cdot x_i - b \geq 1$ for the class with the label +1 and $w \cdot x_i - b \leq -1$ for the class with the label -1, by maximizing the distance between these hyperplanes the supporting planes are 'pushed' apart until they 'bump' into the small number of points on the boundaries of each of the classes (see Figure 3.4). These points are the support vectors and the distance between the supporting planes $\gamma = \frac{2}{\|w\|}$ is the maximum geometric margin. Maximizing the margin in such a manner is equivalent to minimizing $\|w\|/2$ in the following

$$\begin{aligned} &\min \ \frac{1}{2}\|w\|^2 \\ &\text{s.t.} \quad w \cdot x_i \geq b+1 \qquad y_i \in Class1 \\ &\qquad w \cdot x \leq b\text{-}1 \qquad y_i \in Class2 \end{aligned} \tag{3.5}$$

where the constraints can be simplified to $y_i(w \cdot x_i - b) \geq 1$.

[13] A hyperplane with all the elements of one class on a side is said to support that class.

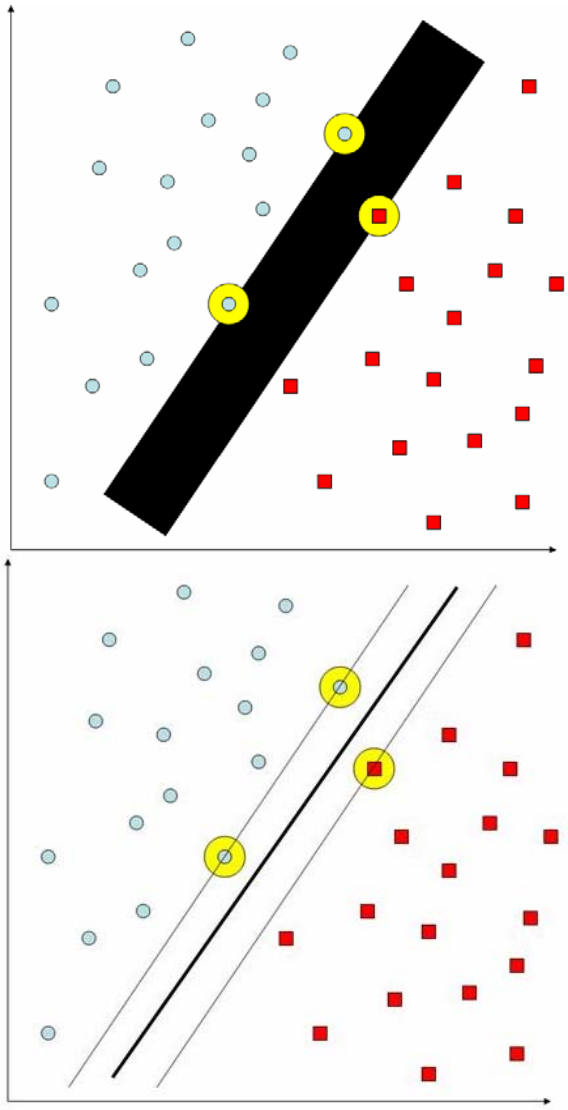

Figure 3.4. Representation of 'Pushing' Supporting Hyperplane Approach. By pushing potential candidate hyperplanes successively outward, they eventually 'bump' into data points. These data points that are 'bumped into' the support vectors. They define the 'fattest' separating plane which can be bisected to defined the maximum margin hyperplane.

The second geometric interpretation of maximum geometric margin problem involves the convex hulls of the points representing the two classes. Once the convex hulls are found for each of the classes in the training set, the points that are closest to one another from each can be determined. With this approach, these points are the support

vectors and by finding the plane that bisects the support vectors we can solve for the geometric margin maximizing hyperplane (see Figure 3.5). This approach implies the following quadratic problem:

$$\begin{aligned} &\min_{\alpha} \ \frac{1}{2}\|c-d\|^2 \\ &\quad c=\sum_{y_p \in Class1}^{p} \alpha_p x_p \qquad d=\sum_{y_q \in Class2}^{q} \alpha_q x_q \\ &\text{s.t.} \quad \sum_{y_p \in Class1}^{p} \alpha_p = 1 \qquad \sum_{y_q \in Class2}^{q} \alpha_q = 1 \\ &\quad \alpha_i \geq 0 \qquad i=1,\ldots,n \end{aligned} \tag{3.6}$$

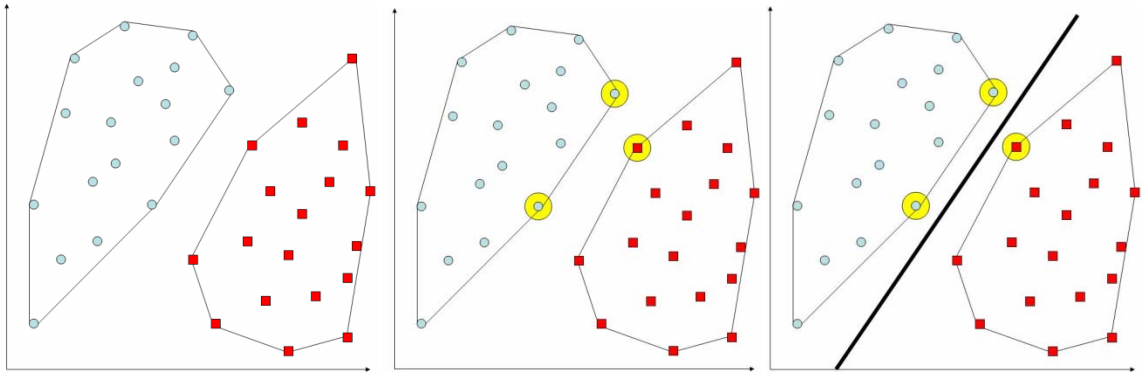

Figure 3.5. Graphical Representation of Convex Hull Approach. The convex hull approach where the nearest points on the convex hulls for the two classes can be used to derive the maximum geometric margin hyperplane.

3.5.4 Optimization

Having covered the basic definitions and generalization concepts related to SVMs, this section describes how one can derive an optimal hyperplane. From the two formulations described in Section 3.5.3, one can see that the problem of finding the hyperplane reduces to an optimization problem. More specifically, the problem involves the optimization of a convex function involving a quadratic objective function with linear inequality constraints (Christianini, 2000; Scholkopf, 2001). The discussion below is

presented in terms of the separating planes but could equivalently be derived from the convex hull formulation (Bennett, 2000).

Recall that the definition for the functional margin is the minimum of the functional margin of the distribution and can be written

$$y_i\left(\langle w \cdot x_i \rangle + b\right) = 1, \text{ or less compactly as,} \tag{3.7}$$

$$\begin{aligned} &\langle w \cdot x^+ \rangle + b = +1 \quad \text{and} \\ &\langle w \cdot x^- \rangle + b = -1. \end{aligned} \tag{3.8}$$

The geometric margin, γ, with w normalized can then be written as half of the difference between the two classes, or the distance from each class to a maximal margin hyperplane, as

$$\gamma = \frac{1}{2}\left(\left\langle \frac{w}{\|w\|} \cdot x^+ \right\rangle - \left\langle \frac{w}{\|w\|} \cdot x^- \right\rangle \right) \tag{3.9}$$

$$\gamma = \frac{1}{2\|w\|}\left(\langle w \cdot x^+ \rangle - \langle w \cdot x^- \rangle \right) \tag{3.10}$$

$$\gamma = \frac{1}{2\|w\|}\left((+1) - (-1) \right) \tag{3.11}$$

$$\gamma = \frac{2}{2\|w\|} \tag{3.12}$$

$$\gamma = \frac{1}{\|w\|}. \tag{3.13}$$

With the goal of maximizing the geometric margin, and with the geometric margin $\gamma = \frac{1}{\|w\|}$, the following optimization problem can be derived:

$$\begin{aligned} &\min \ \frac{1}{2}\|w\|^2 \\ &\text{s.t.} \ \ (w \cdot x_i) + b \geq 1 \qquad i = 1,...,\ell \end{aligned} \tag{3.14}$$

which implies the Lagrangian

$$L(w,b,\alpha) = \frac{1}{2}\langle w \cdot w \rangle - \sum_{i=1}^{\ell} \alpha_i \left[y_i (\langle w \cdot x_i \rangle + b) - 1 \right] \tag{3.15}$$

where $\alpha_i \geq 0$ are the Largrange multipliers.[14] Given that the problem involves the solution to a convex cost problem, the first order derivatives for the parameters can be written

$$\frac{\partial L(w,b,\alpha)}{\partial w} = w - \sum_{i=1}^{\ell} y_i \alpha_i x_i = 0 \tag{3.16}$$

$$\frac{\partial L(w,b,\alpha)}{\partial b} = \sum_{i=1}^{\ell} y_i \alpha_i = 0 \,. \tag{3.17}$$

Substituting these formulations back into the original Lagrangian, the following Wolfe dual expressed solely in terms of the multipliers can be derived (see Vapnik, 1998 for complete derivation):

$$\begin{aligned} &\min_{\alpha} \ \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} y_i y_j \alpha_i \alpha_j x_i x_j - \sum_{i=1}^{n} \alpha_i \\ &\text{s.t.} \ \ \sum_{i=1}^{n} y_i \alpha_i = 0 \\ &\qquad \alpha_i \geq 0 \quad i = 1,...,m \end{aligned} \tag{3.18}$$

[14] The Lagrangian multipliers indicate the activity of the constraints for a given data point. If the constraints are active at a given point, the value of the multiplier will be non-zero. For other points, where the solution is not influenced by the constraints, the Lagrange multipliers will equal zero.

which can be rewritten in terms of inner products as

$$\begin{aligned} \min_{\alpha} \quad & \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} y_i y_j \alpha_i \alpha_j \left\langle x_i \cdot x_j \right\rangle - \sum_{i=1}^{n} \alpha_i \\ \text{s.t.} \quad & \sum_{i=1}^{n} y_i \alpha_i = 0 \\ & \alpha_i \geq 0 \quad i = 1,...,m \end{aligned} \tag{3.19}$$

This formulation, containing a dot product, allows for the kernel substitution

$$\begin{aligned} \min_{\alpha} \quad & \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} y_i y_j \alpha_i \alpha_j K(x_i, x_j) - \sum_{i=1}^{n} \alpha_i \\ \text{s.t.} \quad & \sum_{i=1}^{n} y_i \alpha_i = 0 \\ & \alpha_i \geq 0 \quad i = 1,...,m \end{aligned} \quad . \tag{3.20}$$

The inequality constraints imply the following Kuhn-Tucker (KT) complementarity conditions:

$$\alpha_i^* \left[y_i \left(\left\langle w^* \cdot x_i \right\rangle + b^* \right) - 1 \right] = 0 \qquad i = 1...,\ell \,. \tag{3.21}$$

The KT conditions imply that for only those points where the functional margin is 1, and therefore the closest points to the hyperplane, are the α_i^* non-zero. Those points with non-zero values are the support vectors, and by looking at the expressions for both the objective and the constraints one can see that the solution is dependent only on these points.

The final formulation (Eq. 3.20) with the kernel substitution brings the discussion of the properties and theory behind SVMs that began with the concept of the "kernel trick" full circle. Key to these computation savings are the use of the dual formulation,

written in terms of the Lagrangian multipliers and an inner product feature space. As such, the feature space does not have to be explicitly defined. Rather, it can be implicitly determined through the use of kernels which allows for very high, even infinite, dimensional feature spaces to become environments for tractable learning (Burges, 1998; Christianini, 2000). In addition, it is noteworthy that the solution is solely dependent on the support vectors, which offers further computational savings.

3.5.5 Soft Margin SVMs

A limitation of the SVM is its assumption of perfect linear separation in feature space. Indeed, if the data are noisy, as they often are in real-world situations, this assumption can easily be violated and result in a primal objective function with an empty feasible set and an unbounded objective in the dual (Christianini, 2000). This section describes how SVM methods can be adapted to relax the assumption of complete separation through the use of slack variables and what is known as the "soft margin" SVM. By implementing this more tolerant decision function, the SVM becomes more robust to noise and hence is more commonly applied in practice.

Central to the development of this more robust estimator are the concepts of the margin distribution (Section 3.5.1) and that of the slack variable, ξ_i, which defines the extent to which a point fails to have a margin γ from the hyperplane (see Figure 3.6).

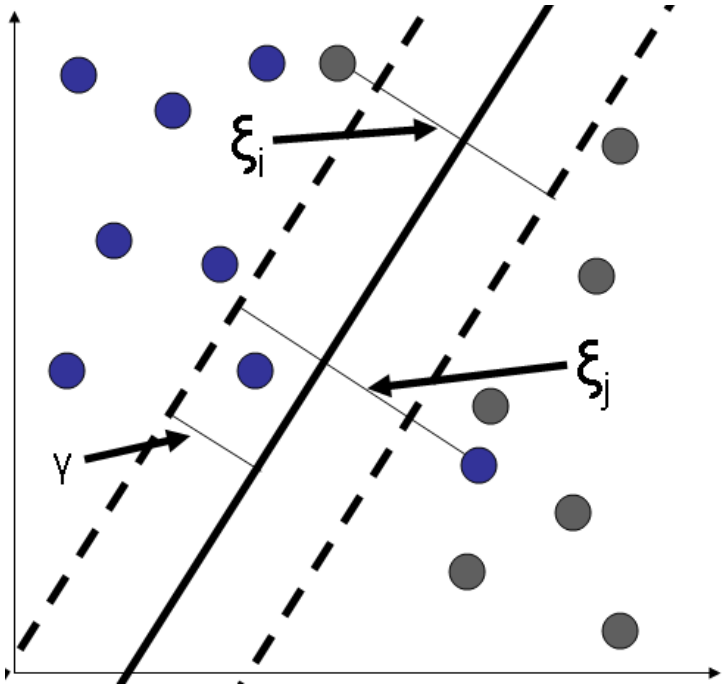

Figure 3.6. Representation of the soft-margin concept. The slack variables ξ for the points x_i and x_j are the amount by which each of these points fail to have the geometric margin γ. If ξ_i is greater than γ it will be misclassified.

The effect of introducing the slack variables is to relax the constraints on the functional margin (shown in Equations 3.7 and 3.8 in the beginning of the optimization subsection) so that they can be written

$$y_i\left(\langle w \cdot x_i \rangle + b\right) \geq 1 - \xi_i \text{, or less compactly as,} \tag{3.22}$$

$$\begin{aligned} &\left\langle w \cdot x^{+} \right\rangle + b \geq +1 - \xi_i \text{ and} \\ &\left\langle w \cdot x^{-} \right\rangle + b = -1 + \xi_i \text{ with} \\ &\xi_i \geq 0 \ \ \forall i \end{aligned} \tag{3.23}$$

As such, for an error to occur ξ_i must be greater than one and $\sum_i \xi_i$ becomes an upper bound on the number of training errors (Burges, 1998). As such, the primal objective function (from the optimization section) $\frac{\|w\|^2}{2}$ is instead written $\frac{\|w\|^2}{2} + C\left(\sum_i \xi_i\right)^k$ where

C is a user-defined parameter that controls the penalty for slack variables and where higher values of C indicate higher penalties and k is a user-defined integer value. When k = 1 or k =2, the problem continues to have a convex cost function and can be solved using quadratic programming. With k = 1, the Wolfe dual (the primal objective written in terms of the Lagrangian multipliers) continues to have the advantage that neither the ξ_i nor their Lagrangian multipliers appear in the formulation. The result is the following

$$\begin{aligned} &\min_{\alpha} \ \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} y_i y_j \alpha_i \alpha_j K(x_i, x_j) - \sum_{i=1}^{n}\alpha_i \\ &\text{s.t.} \ \sum_{i=1}^{n} y_i \alpha_i = 0 \\ &\quad 0 \le \alpha_i \le C \quad i = 1,\ldots,n \end{aligned} \tag{3.24}$$

where the significant difference from the hard margin (linearly separable) case is the presence of an upper bound on the Lagrangian multipliers α_i. The KKT conditions become

$$\begin{aligned} &\alpha_i^* \left[y_i \left(\left\langle w^* \cdot x_i \right\rangle + b^* \right) - 1 + \xi_i \right] = 0 \qquad i = 1\ldots,\ell \\ &\mu_i \xi_i = 0 \end{aligned} \tag{3.25}$$

This formulation is known as the 1-norm soft margin SVM.

With k = 2, the formulation is known as the 2-norm soft margin SVM and the Wolfe dual is written

$$\min_{\alpha} \quad \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} y_i y_j \alpha_i \alpha_j \left(K(x_i, x_j) + \frac{1}{C}\delta_{ij} \right) - \sum_{i=1}^{n} \alpha_i$$

$$\text{s.t.} \quad \sum_{i=1}^{n} y_i \alpha_i = 0 \tag{3.26}$$

$$0 \le \alpha_i \le C \quad i = 1,...,n$$

with the KKT conditions

$$\alpha_i^* \left[y_i \left(\langle w^* \cdot x_i \rangle + b^* \right) - 1 + \xi_i \right] = 0 \qquad i = 1...,\ell \tag{3.27}$$

where δ is the Kronecker delta and is equal to one when $i = j$ and 0 otherwise. This has the effect of weighting the diagonal by $\frac{1}{C}$, thereby adding $\frac{1}{C}$ to eigenvalues of the kernel matrix and rendering the problem better conditioned.

Historically, prior to the establishment of generalization bounds, the 1-norm SVM has been preferred in practice due to its similarity to the percentile error bound. However, since the generalization bounds for the 1 and 2-norm SVMs have been developed (Shawe-Taylor, 1999), both versions have become commonplace and the data and type of noise influence which performs better in practice (Christianini, 2000).

One problem with both soft margin SVM formulations is the presence of the parameter *C* and the necessity of deriving a value for it. A common approach to this issue is varying values of *C* through a range. However, the value of *C* is also sensitive to the scale of the feature space and this presents a further complication. Approaches for the determination of an appropriate parameter value can be computed in terms of the VC dimension, *v*. It has been shown that solutions for varying *C* are equivalent as those that result from varying *v*. This approach is advantageous in that manipulation of *v* not only

is independent of the scale of the feature space, but also allows for greater interpretation in that v represents a lower bound on the sum of the Lagrange multipliers α_i, therefore providing a lower bound on the number of support vectors, and providing a more transparent parameterization of the problem in terms of the noise level of the data (Christianini, 2000).

3.5.6 Mapping SVM Results back to Input Space

Once derived in feature space, the maximum margin separating hyperplane can be mapped from feature space back to input space. This process involves deriving the optimal parameter values for w^* and b^* (Eqs. 3.7-3.8) from the optimized Lagrange multipliers α^* obtained from Eq.3.20 or Eq. 3.26. The expression for w^* can be simply derived directly from the constraint (Eq. 3.5) where

$$w^* = \sum_{i=1}^{\ell} \alpha_i^* y_i x_i \tag{3.28}$$

Meanwhile, b^* may be obtained through the KT conditions (Eq. 3.9) and the optimized value w^* so that

$$b^* = y_i - \sum_{i=1}^{\ell} y_i \alpha_i^* K(x_i, x_j) \tag{3.29}$$

over all points with $\alpha_i > 0$ (from the ℓ support vectors). With these optimized values, the decision function

$$f(x) = \operatorname{sgn}\left(\sum_{i=1}^{m} y_i \alpha_i^* K(z, x_i) + b^* \right). \tag{3.30}$$

is obtained and is used to suggest labels for test points z (Scholkopf, 2002). With the initial input data representing geosensor locations and Boolean event predicate results and by inputting a mesh of test points z over the area considered as being monitored by the geosensor network, the results of the decision function can be used as a means of interpolating the spatial extent of events from these point readings.[15] The effect of the test function (Eq. 3.28) is the generation of a raster which gives positive results for locations hypothesized to be in-event and negative results for locations estimated to be non-event.

In addition, and more importantly for the purposes of this research, the areal event boundary can be estimated by finding those points for which the decision function is equal to zero. With the interpretation of SVM-derived decision function as a boundary for higher-level events and the ability of SVM decision functions to follow the distribution of lower-level events, SVMs may have the ability to provide accurate representations of the spatial extent of higher-level events. Chapter 4 describes implementation of SVMs for spatial boundary estimation, discusses some properties for spatial analysis, and provides comparison against existing techniques.

3.6 Support Vector Clustering

The previous sections of this chapter introduced the base concepts of SVMs, and set the stage for discussion of SVC. Of particular relevance are the concepts of duality (i.e., primal in terms of geometric margin vs. Wolfe dual in terms of Lagrangian multipliers), the "kernel trick," and soft margins. For SVMs, these concepts allowed for the derivation of maximum margin hyperplanes which can be solved efficiently as

[15] The spatial extent of the meshgrid should be dictated by application specific assumptions concerning the effective coverage of a geosensor network.

quadratic programming problems in terms of inner products kernels and Lagrangian multipliers. It will be shown that the process is much the same for SVC where, instead of maximum margin hyperplanes, the decision function is that of a minimum bounding hypersphere.

This difference in decision function is driven by the inherent difference in data between clustering and classification problems. For classification problems, where the objective is to delineate distinct classes, using decision functions to suggest class assignment for novel points based on a training set. In contrast, unsupervised learning involves input data from only one class. In such a situation, labels do not exist and the objective of learning is geared towards the description of patterns in the distribution of the data and the identification of clusters or, inversely, outliers.

Clusters, hot-spots, or areas of point concentration are of interest in a variety of geographic application scenarios. Some of these, along with methods for the estimation of their spatial extent, were discussed in Section 2.2. A common limitation of existing techniques is their inability to produce estimates of potentially complex spatial extents of clustering events. Given that the purpose of this thesis is to describe the spatial evolution of events, including how events deform with time, it seeks more sophisticated means for the estimation of the spatial extent of events. SVC is introduced as a method for describing spatiotemporal evolution of clustering events. In Chapter 5, a simulation demonstrates how SVC results could be incorporated in spatiotemporal data models such as the spatiotemporal helix for spatiotemporal analysis. The following subsection describes the SVC algorithm.

3.7 Support Vector Clustering Algorithm

Support vector methods can be used to solve supervised learning problems, as discussed in previous sections, and can also be adapted to address unsupervised learning problems where, in the absence of labeled training sets, the objective is generally the description of clustering in the data or the identification of outliers. As with SVMs, the algorithm for the unsupervised case involves the resolution of a convex cost function in a kernel-produced feature space.

Given the use of kernels, these methods have a close relationship to established kernel density estimation (KDE) methods (see Section 2.3) for cluster description. A key difference between the two lies in the form of the results produced. In KDE, the purpose is to generate a description of the relative spatial density of individual event instances. Therefore, KDE results in a map of the study area which highlights areas of high and low concentrations according to a given kernel. Meanwhile, the support vector methods presented in this chapter generate estimates for the spatial extent of clusters in terms of a boundary.

According to the stated objective of the analysis (outlier detection vs. cluster determination), the name given to this class of support vector methods has varied in the literature. When the objective is the determination of outliers, these methods have been called support vector novelty detection (Scholkopf, 2000). When the objective involves the inverse task of describing concentrations of points, these methods have been referred to as support vector clustering (Ben-Hur, et al. 2001). Finally, when the objective is simply a general description of the distribution of data these techniques have also been called support vector data description (Tax et al., 1999). Given the tradition of clustering

research in spatial analysis, this research refers to these methods as support vector clustering (SVC).

As with other kernel methods, the defining characteristic of SVC is the type of decision function applied in feature space. Where SVMs employed maximum margin hyperplanes separating data with different class labels, SVC, designed for data of only one class, implements minimum bounding hyperspheres. With $\{x_i\} \subseteq \chi$ a set of N data points in the data space $\chi \subseteq \Re^d$, the smallest sphere enclosing the data can be represented as

$$\left\|\Phi(x_i) - a\right\|^2 \leq R^2 \qquad \forall i \tag{3.31}$$

where $\|\cdot\|$ is the Euclidean norm, Φ is some non-linear transformation from χ to a high dimensional feature space, and a is the center of the minimum bounding sphere in that feature space. The differentiation of clustered points from outliers implies the use of slack variables ξ_i, with the identical definition presented in Section 3.5.5, and resulting in a soft margin bounding hypersphere so that Eq. 3.31 is written

$$\left\|\Phi(x_i) - a\right\|^2 \leq R^2 + \xi_i \qquad \forall i \tag{3.32}$$

with $\xi_i \geq 0$. To solve this problem and derive the hypersphere, implies the use of the following Lagrangian:

$$L = R^2 - \sum_{i=1}^{N}\left(R^2 + \xi_i - \left\|\Phi(x_i) - a\right\|^2\right)\alpha_i - \sum_{i=1}^{N}\xi_i\mu_i + C\sum_{i=1}^{N}\xi_i \tag{3.33}$$

with the Lagrangian multipliers $\alpha_i \geq 0$ and $\mu_i \geq 0$ and the constant C weighting the penalty term $C\sum \xi_i$ for points outside the hypersphere with the KKT conditions

$$\xi_i \mu_i = 0 \text{ and} \tag{3.34}$$
$$\alpha_i (R^2 + \xi_i - \left\|\Phi(x_i) - a\right\|^2) = 0 . \tag{3.35}$$

Minimization of this convex cost function imposes stationarity (first derivatives *L* with respect to *R*, *a*, ξ_i equal zero) and results in the following expressions:

$$\sum_i \alpha_i = 1 \tag{3.36}$$
$$a = \sum_i \alpha_i \Phi(x_i) \tag{3.37}$$
$$\alpha_i = C - \mu_i . \tag{3.38}$$

Several definitions relative to clustering can be drawn from the above equations. First of all, from the KKT (Eqs. 3.31 and 3.32) it follows that the feature space representation of an input data point x_i with $\xi_i > 0$ and $\alpha_i > 0$ lies outside the hypersphere in feature space and can be determined to be an outlier. In the terminology of SVC, such points are called *bounded support vectors*. Therefore, points x_i with $\xi_i = 0$ are mapped either inside or onto the surface of the sphere in feature space. Such points with $0 < \alpha_i < C$ are located on the surface of the sphere and are the (*unbounded*) *support vectors* while those with $\alpha_i = 0$ are points located inside the cluster and are simply called *interior points* (Ben-Hur et al., 2001).

Using the equations derived from the first order conditions (Eqs. 3.34-3.35), one can eliminate the variables *R*, *a*, and μ_i from the Lagrangian giving the Wolfe dual written entirely in terms of the Lagrange multipliers α_i

$$W = \sum_j \Phi(x_j \cdot x_j)\alpha_j - \sum_{i,j} \alpha_i \alpha_j \Phi(x_i \cdot x_j) . \tag{3.39}$$

with the constraint

$$0 \le \alpha_i \le C , \; i = 1,..., N . \tag{3.40}$$

At this point, an appropriate Mercer kernel *K*, can be introduced and with the determination that Mercer kernels are a representation of the inner product it can be substituted so that

$$W = \sum_j K(x_j, x_j)\alpha_j - \sum_{i,j} \alpha_i \alpha_j K(x_i, x_j) \tag{3.41}$$

Although any Mercer kernel can be used, the Gaussian kernel has been employed in SVC applications almost universally.[16] In this research the Gaussian kernel

$$K(x_i, x_j) = e^{-q\left\|x_i - x_j\right\|^2} \tag{3.42}$$

is considered with the kernel width parameter $q = 1/2\sigma$. With this kernel the Wolfe dual can be written

[16] Tax and Duin (1999) found that use of the polynomial kernel can lead to excessive influence of extreme outliers.

$$W = 1 - \sum_{i,j} \alpha_i \alpha_j K(x_i, x_j)^{17} \tag{3.43}$$

where the optimized values for α_i depict a solution in terms of the support vectors alone.

Both from the above formulations and intuitively, it is obvious that the hypersphere can be determined by the (unbounded) support vectors alone. As a result, these points, with $0 < \alpha_i < C$, alone are used to map the hypersphere back to input space and produce a representation of the cluster boundary in that space. The process involves feeding a mesh of novel points $z \in Z$ representing the spatial extent of the study area into the following expression for the distance of the feature space image of the point *z* from the center of the hypersphere:

$$R^2(z) = \left\| \Phi(z) - a \right\|^2 \tag{3.44}$$

With $a = \sum_i \alpha_i \Phi(x_i)$ (Equation 3.34) and kernel substitution for Φ, this formulation can be rewritten

$$R^2(z) = K(z,z) - 2\sum_i \alpha_i K(z, x_i) + \sum_{i,j} \alpha_i \alpha_j K(x_i, x_j) \tag{3.45}$$

$$R^2(z) = 1 - 2\sum_i \alpha_i K(z, x_i) + \sum_{i,j} \alpha_i \alpha_j K(x_i, x_j) \tag{3.46}$$

$$R^2(z) = 1 - 2\sum_i \alpha_i K(z, x_i) + S_X \tag{3.47}$$

where the x_i, x_j are support vectors and S_X being the constant sum of the product of the kernel-based representation of the inner product of support vectors with their Lagrangian multipliers $0 < \alpha_i < C$. With the Gaussian kernel in the first term equal to one and S_X, the only term that varies in Eqs. 3.42-3.44 with each novel point *z* is the second one.

[17] The distance between a point and itself is zero. This results in an expression for the exponential of zero which is equal to one.

Using these formulations, the radius of the minimum bounding hypersphere can be obtained through the consideration of the points $x_i \in Z$ where the x_i are the support vectors, or equivalently

$$R = \{R(x_i) \mid x_i \text{ is a support vector}\}. \tag{3.48}$$

This value, generated by the points lying on the hypersphere, can then be used to compare against the values generated through the expression in Eq. 3.46 and by identifying those points z with values that are equal, therefore determining contours representing the extent of the clusters in input space. This can be written

$$\{z \mid R(z) = R\}. \tag{3.49}$$

With geographic data, these contours can be used to estimate the spatial extent of events. The spatial precision afforded through the use of these techniques makes them attractive as a means of generating an areal representation which can be analyzed in terms of both movement and deformation. The process by which such an analysis could be conducted is outlined in the next chapter.

Chapter 4

IMPLEMENTATION

Current techniques for the extraction of areal event boundaries from point-based events, occurring at points in both point process and dynamic spatial scalar fields suffer from a priori biases. Methods typically applied to point process data such as hierarchical clustering or the scan statistic, impose elliptical or strictly convex forms on higher-level events. Methods applied to geosensor data impose linear representations of boundaries when, in actuality, the true forms are likely to be non-linear (e.g., a toxic cloud).

The support vector methods presented in the previous chapter describe two means of extracting non-linear representations of areal event boundaries. Like kernel density estimation, these support vector methods depict the distribution of observations of lower-level events through the implementation of non-linear kernel transformations. Unlike KDE, however, these methods provide a means for the extraction of explicit representations of the boundaries of higher-level events which can be incorporated into existing methods for spatiotemporal analysis. This chapter interprets these support vector algorithms visually, in terms of their intermediate results, and critiques their potential for application for obtaining areal event boundaries.

4.1 Support Vector Clustering

Due to its potential for describing point cluster event shapes with complex non-linear shapes, SVC has already been examined for its application potential in security informatics (Zeng, Chang, et al., 2004; Chang, Zeng, et al., 2005). The objectives in security informatics, outlined in the introduction, can be summarized as involving 1) the

identification of concentrations of lower-level event points, 2) statistical testing of concentrations to determine whether they are clusters, and 3) description of how clusters change shape and move over time. The third of these objective parallels the central motivation for this thesis, which is description of the evolution of higher-level event shapes.

In their purely spatial application of SVC, Zeng, Chang et al. (2004) investigated the application potential for input space representation of cluster boundaries and compared their results against those produced by the scan statistic and hierarchical clustering (Figure 4.1). Conclusions suggested further consideration of SVC for spatiotemporal applications due to SVC's capability to produce more complex representations of cluster boundaries. However, this work did not generate any suggestions for eventual spatiotemporal analysis with SVC.

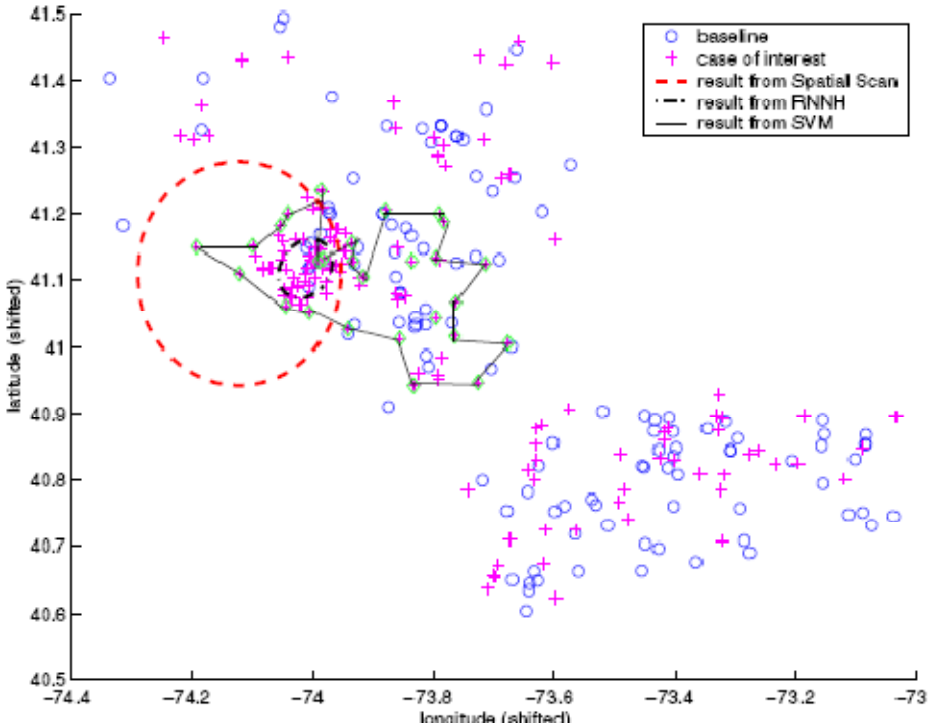

Figure 4.1. Comparison of scan statistic, hierarchical clustering, and SVC produced representations of cluster boundaries in space (Zeng, Chang, et al., 2004). The exact

approach taken by the authors to produce the SVC results is unclear given that their SVC results appear similar to a convex hull while kernel methods are strictly non-linear in that they are produced by non-linear kernels.

A later application of SVC was designed for spatiotemporal analysis (Zeng, Chang, et al., 2005). In this research the authors incorporated time directly into the SVC algorithm, exploiting the ability of SVC to handle high dimensional data. Rather than using the decision function to produce explicit representations of higher-level event boundaries, this approach used the decision function implicitly and produced results in terms of labels reflecting the location of points relative to the decision function (i.e., whether each point was clustered or not) As in the purely spatial case, these results were compared against those from a scan statistic (Figure 4.2). While integrating both space and time, with output consisting solely of clouds of clustered points this approach offers limited appicability for the description of spatiotemporal behavior.

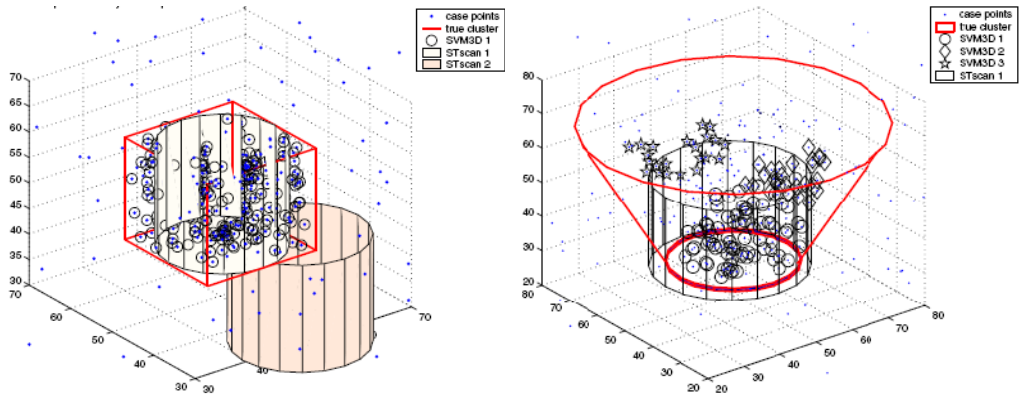

Figure 4.2. Comparison of scan statistic and SVC produced labels for points in space and time (Chang, Zeng, et al., 2005). As opposed to Figure 4.1, which was purely spatial, these results include temporal coordinates directly into computations. The second image highlights the limitations of the scan statistic to describe spatiotemporal behavior such as spread and contraction.

An objective of this research is to take a different approach to spatiotemporal analysis, using SVC-generated decision functions to produce purely spatial "snapshots"

of cluster boundaries through time and then compare differences in the shape and location of the higher-level events through time. Like Zeng, Chang et al. (2004), this approach uses SVC-produced representations of cluster boundaries to define clusters in terms of regions with high concentrations of event instances over a discrete period of time rather than as clouds of spatiotemporal points. To incorporate the temporal dimension into this framework, derivations of boundaries are repeated through time which results in a sequence of boundary representations not unlike those extracted from video sequences of image data. A method for describing event evolution such the spatiotemporal helix (Stefanidis et al., 2003) can then be used to compare sequences of these events over time.

4.1.1 Visual Interpretation of SVC Results

Considering that KDE is an established method for spatial analysis, and that both SVC and KDE are based on kernel transformations, this section provides a visual interpretation of the SVC algorithm that sequentially compares intermediate SVC results alongside those from KDE. Discussion then leads to some of the limitations of SVC, and concludes with some remarks about spatiotemporal modeling with SVC-produced representations of cluster boundaries. These comparisons begin with relatively simple versions of the SVC algorithm and migrate towards increasing complexity with the addition of parameters.

As described in Chapter 2, KDE is a method of representing the probability distribution of points throughout a study area. By evaluating kernel functions at each observation, summing the kernel values from all observations at each point in the study area, and then normalizing by the number of observations in the study area, a representation of the cumulative distribution of instances is produced (Figure 4.3). The

effect of these transformations is the production of a raster, or continuous representation throughout the study area, of the relative distribution of observations (Waller and Gotway, 2004). These representations have been used for spatial analysis of point patterns in applications including epidemiology (Bithell, 1990), crime analysis (Levine, 2007), and even archaeology (Baxter, Beardah et al., 1997). While effective in providing a visual representation describing areas that have relatively high or low concentrations of point events, KDE does not produce an explicit boundary delimiting areas of concentration. An explicit boundary representation of these areas of concentration is required for "snapshot" based approaches like the spatiotemporal helix.

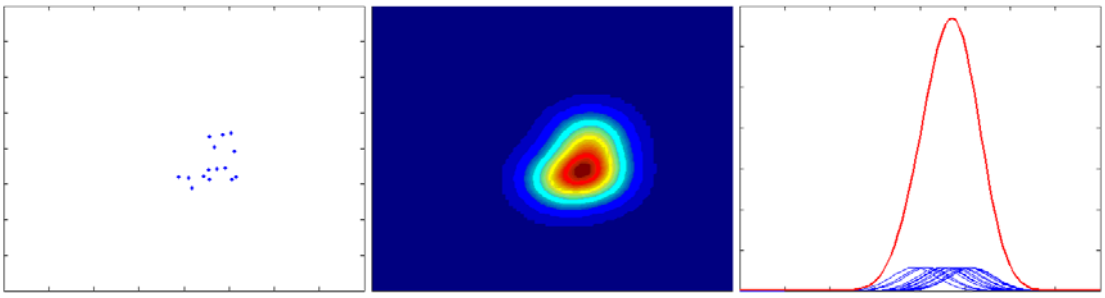

Figure 4.3 Kernel density estimation. The first image shows the distribution of points from which the kernel density estimates are derived. The second image is a cut-away view of the two-dimensional third image with a collapsed y dimension. For comparison purposes with SVC output, it is noteworthy that the weight/height of each of the points is the same (third image).

Like KDE, SVC transforms the data to a kernel based higher-dimensional feature space where a minimum bounding hypersphere is derived. Results from this procedure identify the points containing the most relevant information for describing the cluster boundary (the support vectors) and use these observations to construct a representation of clustering for the data (Figure 4.4). Compensating for the reliance on only the support vector subset of the initial population of observations are weights (α_i's) that are calculated as part of the derivation of the minimum bounding hypersphere (Figure 4.5).

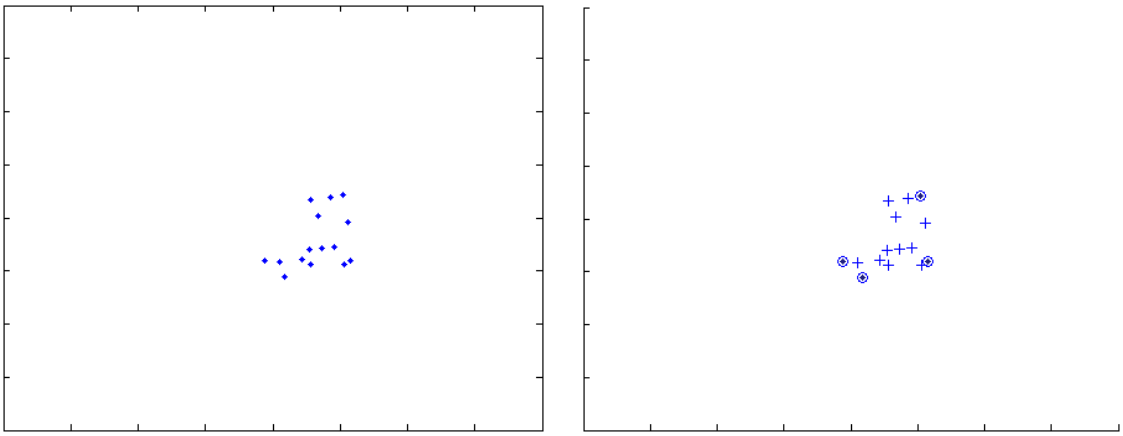

Figure 4.4. Point distribution and support vectors (circles in second image) and interior points (+'s in second image).

The minimum bounding hypersphere can be mapped back to the original input space to obtain a representation of a cluster boundary that can be used for spatiotemporal analysis. This process was outlined in Chapter 3 (Eqs. 3.45-3.47) and involves the derivation of a radius value from the weights α_i and the kernel transformation (Eq. 3.45). Given that the support vectors (x_i with $0 < \alpha_i < C$) are the only points that influence the shape of the cluster boundary, only these points are used to derive this radius value (Eq. 3.46). Once this value is obtained it can be used to test against a mesh of input points $z \in Z$ to find the locations in input space that are located on the cluster boundary (Eq. 3.47). Effectively, this process creates a raster of weighted kernel evaluations for points throughout the study area and is represented in the second image in Figure 4.5. Those points found to have distances equal to the radius of the hypersphere in feature space form the cluster boundary, those with distances less than the radius are considered to be part of the clustering, and those points with distances greater than the boundary are considered to be outliers. This process of assignment is shown in Figure 4.6 which demonstrates how representations for the areal extent of higher-level events can be derived.

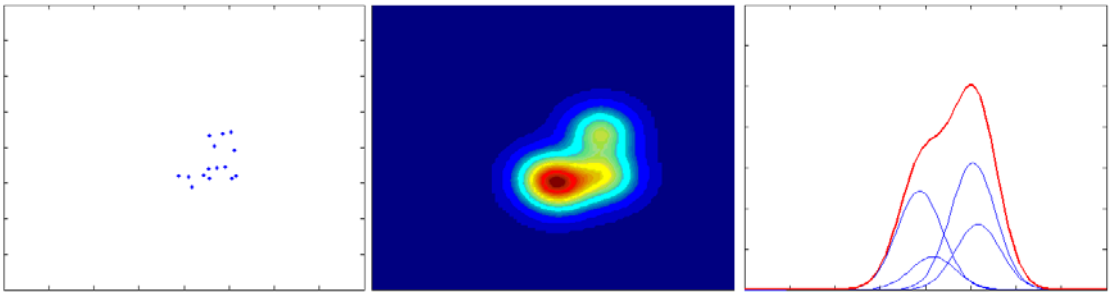

Figure 4.5 SVC output. This set of images is designed for comparison against those resulting from KDE that appear in Figure 4.3. The SVC algorithm "thins" the number of points so that only those relevant to the boundary contribute to the derivation of cluster boundaries. To compensate for this thinning, points are assigned weights (second image) which contrasts with KDE where each observation is equally weighted (1/Nh).

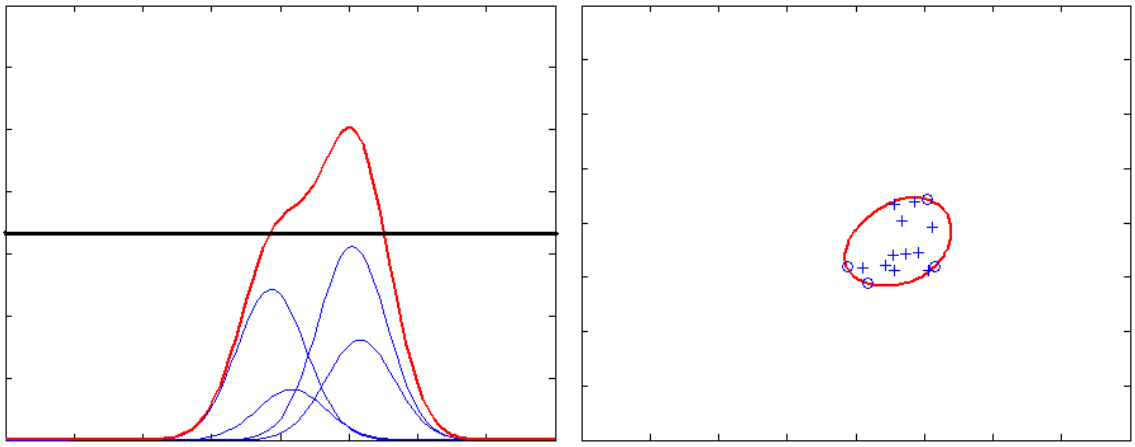

Figure 4.6. Derivation of cluster boundary with SVC. Points in input space are fed through the weighted kernel function for the radius of the hypersphere (represented by the horizontal line in the first image). Those points with values equal to those for the radius of the hypersphere delineate the cluster boundary (second image).[18]

4.1.2 SVC and Outliers

The previous example presented the basic mechanics of the SVC algorithm for a simple dataset without outliers. SVC is designed to handle more complex scenarios that rely on SVC to distinguish between clustered and outlying points. To illustrate how SVC identifies outliers, this section presents another example and discusses the corresponding results.

[18] The image with the weighted curves was created using 2-dimesional (x and y) distances while the points along the x-axis are shown in terms of 1-dimensional distances (just x direction). This causes some distortion, but not enough to distract from the illustrative purpose of the image.

In a kernel-induced feature space, points are organized into a minimum bounding hypersphere whose radius is used to provide an input-space representation of cluster boundaries. The weights play an important role. The only points holding influence on the derivation of the hypersphere are the support vectors (x_i with $0 < \alpha_i < C$). This is shown in Figure 4.6 and was described in Eqs. 4.1-4.3 where the support vectors were distinguished from the interior points in terms of the weights where interior points have $\alpha_i=0$ and the support vectors have weights $\alpha_i>0$.

The values assigned to the support vectors reflect their relative position from the center of the hypersphere. For the interior points, which are relatively close to the center and therefore do not influence the length of the radius, the values are zero. For the support vectors, which are on the radius, values are assigned so that those farther from the center have larger weights. This is because if they are to be located within the radius they require higher values due to their increased distance from neighboring points (the other points contribute relatively little to the value radius). This is illustrated in Figures 4.5 and 4.6 where the upper right support vector has the highest weight due to the fact that it is the farthest from the other three support vectors.

Controlling the size of the weights is the parameter C (Eq. 3.40). In all of the examples in the previous section, the parameter C was held equal to one which, with constraints Eqs 3.36-3.39 and

$$n_{outliters} < 1/C \tag{4.4}$$

where $n_{outliers}$ is the number of outliers and

$$p = 1/NC \tag{4.5}$$

being the upper bound on the percentage of outliers (Scholkopf, Williamson et al., 2000), effectively forced all points to be included in the cluster (no outliers). The remainder of this section is designed to illustrate the role that *C* plays in determining cluster boundaries in the presence of outliers. To do so, results are generated from the dataset in Figure 4.7.

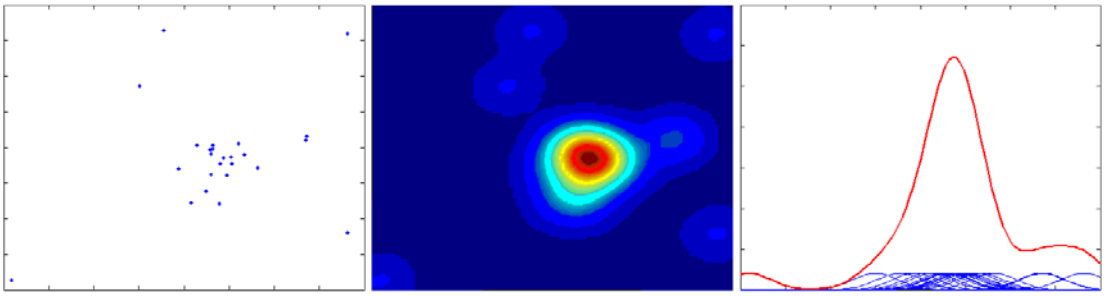

Figure 4.7 Dataset with outliers and corresponding KDE representations. Outliers are located in the lower left and the upper and lower right.

In Figure 4.8 the results are shown for SVC with C=1 (i.e., constraint Eq. 3.40 non-binding) in the presence of outliers. Due to the relatively large distance between the outlying points and the core of the concentration, the outliers have relatively large weights (image 2 of Figure 4.8). This is because with C=1 no compensation is given for points lying far from the rest of the distribution. As a result each of these points is considered as part of the cluster. The fact that the outlier in the bottom right is closer to the bulk of the distribution is reflected by the small portion of boundary located around this point (bottom right image of Figure 4.8). Meanwhile, the outliers in the upper right and lower left are located at the extreme end of radius and do not have any visible portion of the boundary represented near their location (just the points are included).

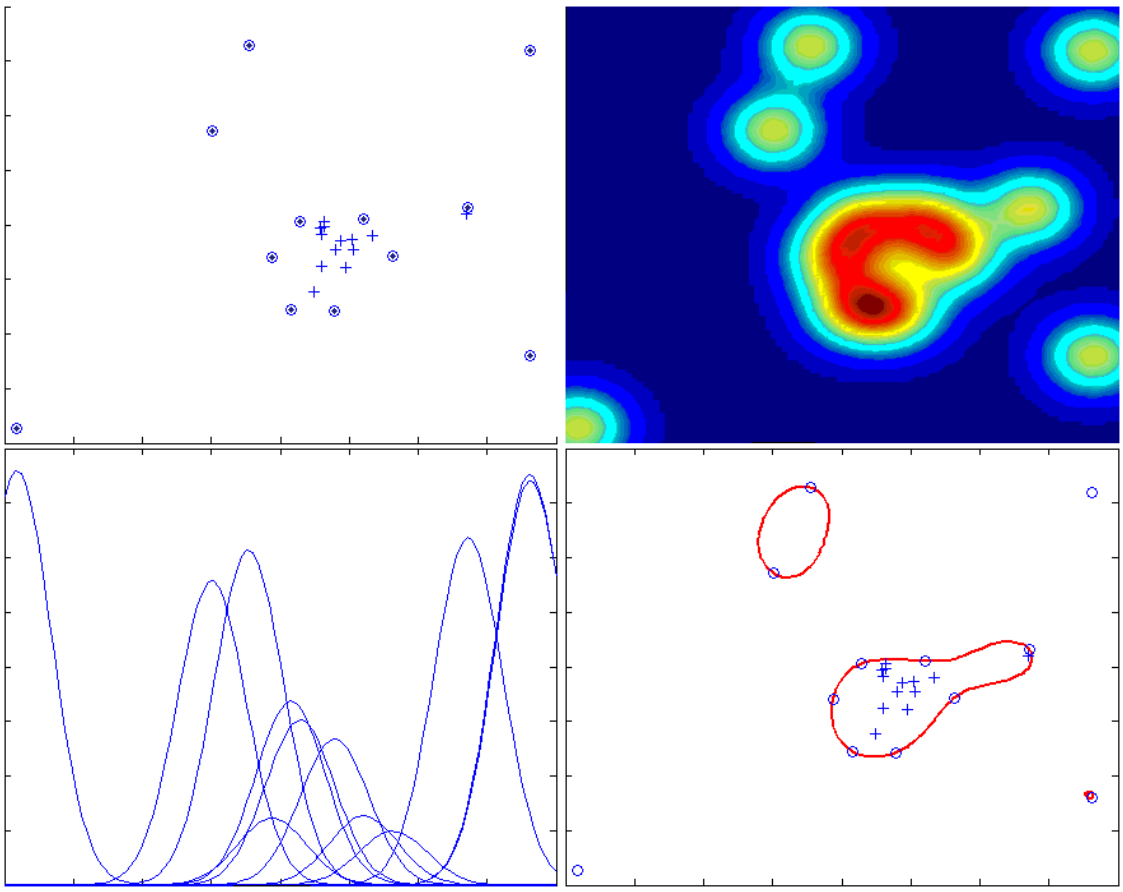

Figure 4.8. Results from running SVC with C=1 on data with outliers. The first image shows the support vectors and interior points. The second image shows the raster produced by testing a mesh of points. The third image shows the weights for the support vectors. The fourth image shows the SVC-produced cluster boundaries.

By allowing C to take on a value less than one, a percentage of the points are allowed to become *bounded support vectors* (i.e., be treated as outliers), limiting the influence that these outlying points have on the determination of the hypersphere/cluster boundary. C defines the upper limit that weights α_i can take (Eq. 3.40) and those points with $\alpha_i=C$, where the constraint is binding, are the bounded support vectors (i.e., outliers). The effect that this has on the generation of cluster boundaries is demonstrated in Figures 4.9 and 4.10.

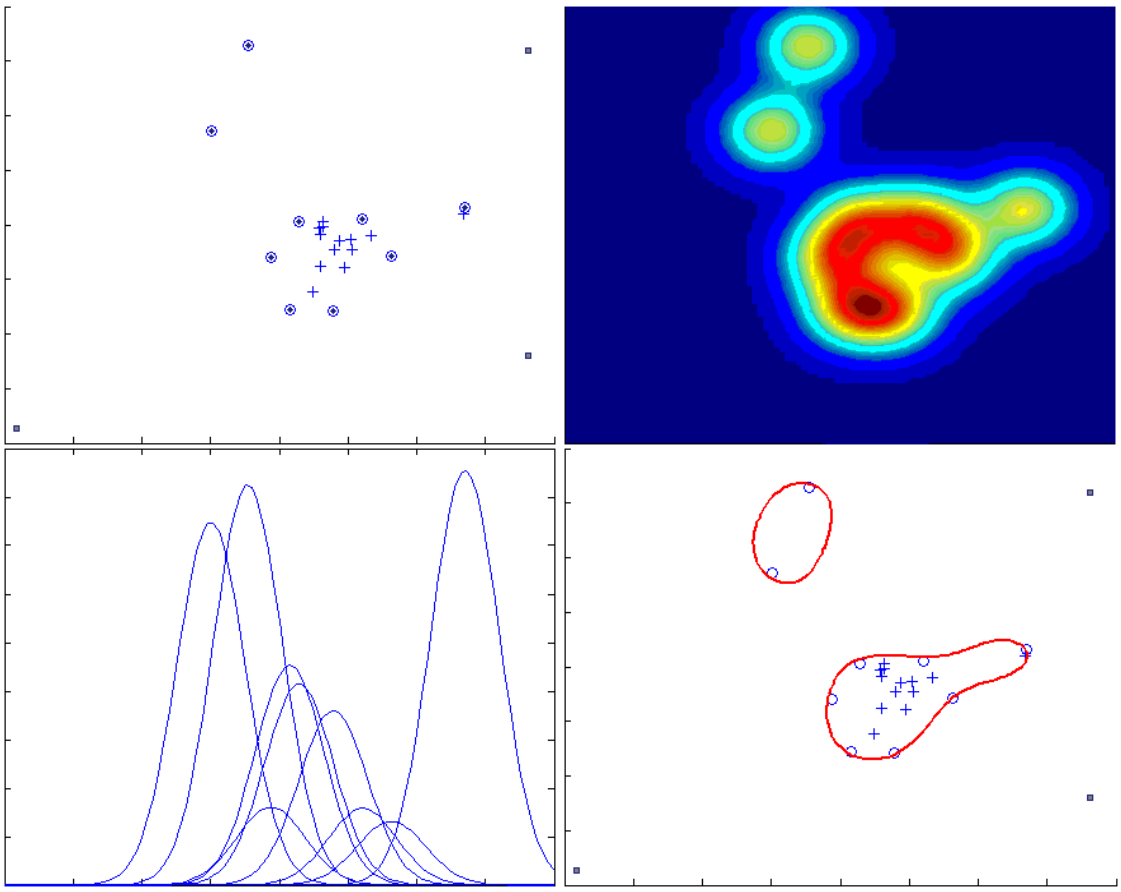

Figure 4.9. SVC results with C<1. The effect of reducing C allowed for outliers/ bounded support vectors. These are shown as the small filled squares in the first image. The second image shows the raster resulting from the weights which are shown in the third image. Note that the bounded support vectors no longer influence the boundary, as opposed to Figure 4.8, where the regions surrounding these points were green.

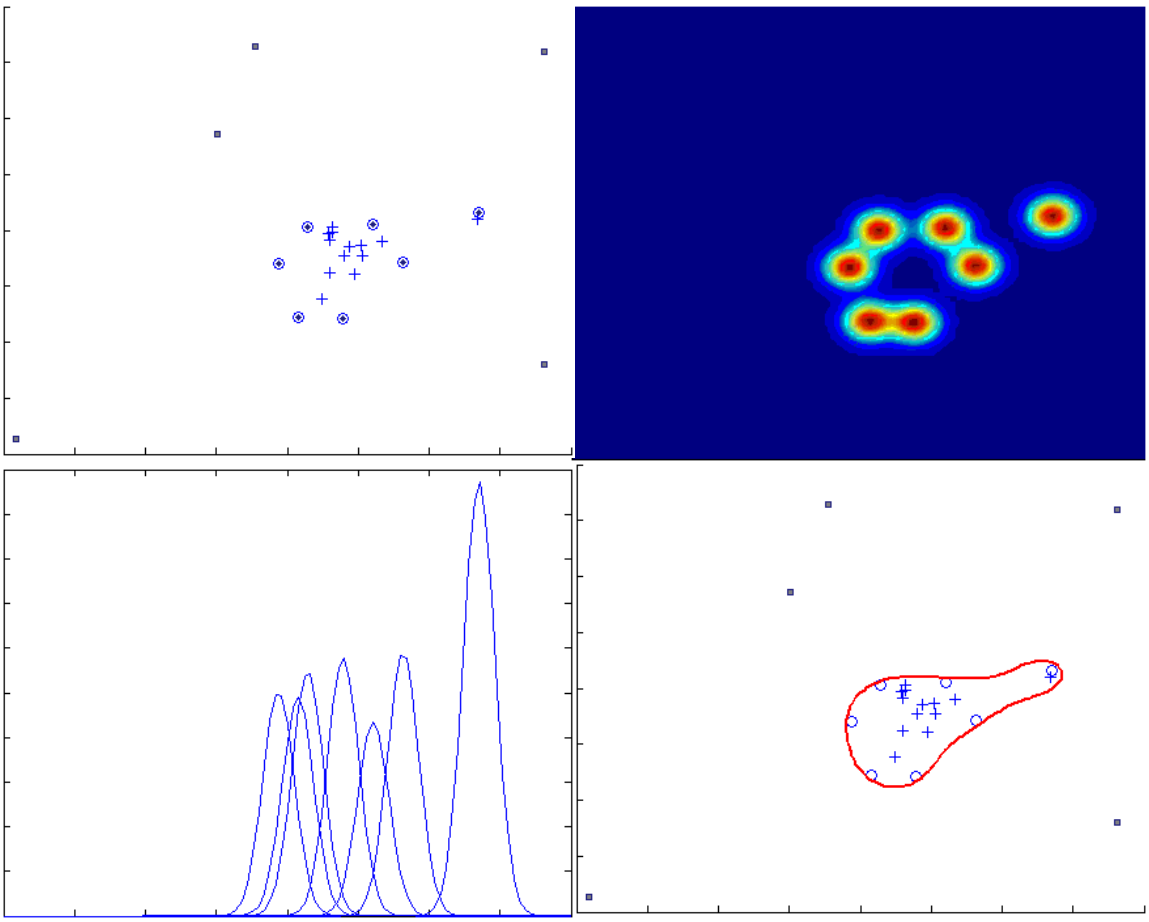

Figure 4.10. Second set of SVC results with $C<1$. The images in this figure result from a larger q (smaller σ) and a smaller value for C (larger percentage of points allowed to be outliers).

To demonstrate the potential of SVC to generate boundaries of higher-level events with actual application data, the algorithm was applied to a collection of crime data. These data represent point locations of homicides in the Pittsburgh area and are downloadable in point shapefile format from the University of Illinois Spatial Analysis Laboratory (http://www.sal.uiuc.edu/stuff/stuff-sum/data). The SVM algorithm was applied to these data to produce the cluster representations depicted in Figures 4.11-4.14.

These figures are effective in demonstrating an important drawback in the application of SVC for spatiotemporal analysis. While SVC is capable of producing highly non-linear representations of clustering and does not impose any bias in terms of

cluster shape a priori, SVC does involve bias implicit in the selection of parameters. For the remainder of the thesis, a priori bias in terms of shape imposed on higher-level events is referred to as *explicit bias* (i.e., bias resulting from the selection of method) while bias resulting from parameter selection is referred to as *implicit bias*. The effects of implicit bias on cluster shape are apparent in the examples in Figures 4.9-4.14 where both the number and shapes of the polygons composing the clusters vary according to the values selected for σ and C.

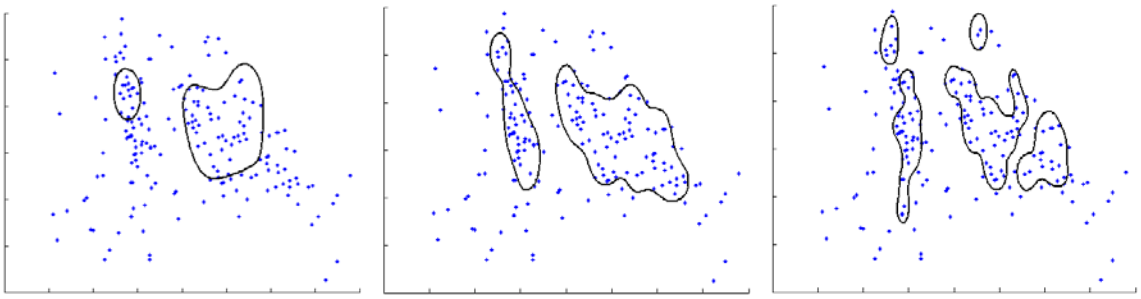

Figure 4.11 Different representations of cluster boundaries generated by SVC with different bandwidths. Bandwidth was manipulated indirectly using $q=1/2\sigma$. Values for q in the above images are 6, 12, and 24 (σ = 3, 6, 12). *C* was held constant (*C*=0.65).

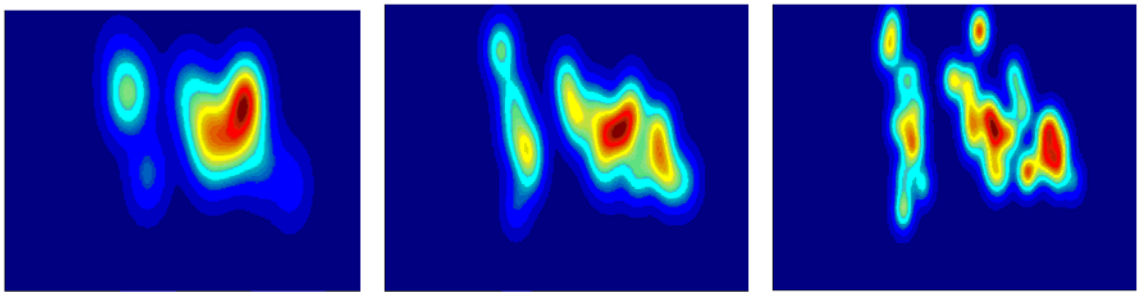

Figure 4.12. Different raster representations of clustering generated by SVC with different bandwidths. These images depict the weighted kernel Bandwidth manipulated indirectly using $q=1/2\sigma$. Values for q in the above images are 6, 12, and 24 (σ = 3, 6, 12). *C* was held constant (*C*=0.65).

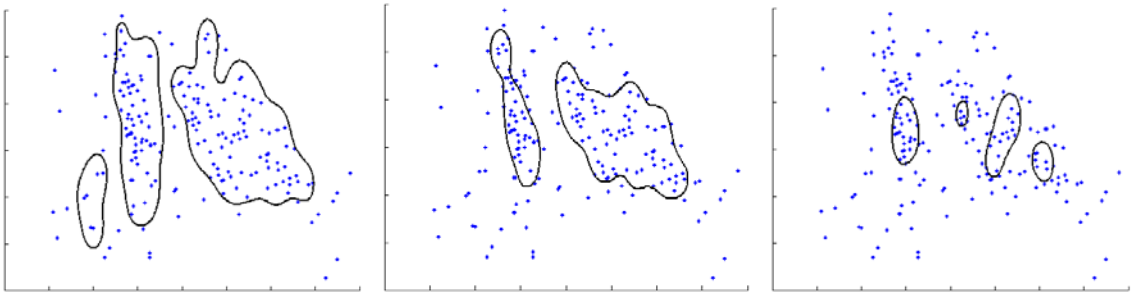

Figure 4.13 Different representations of cluster boundaries generated by SVC with different values for the outlier parameter C. Values for C in the above images are 0.50, 0.65, and 0.85. Bandwidth was held constant (q=12).

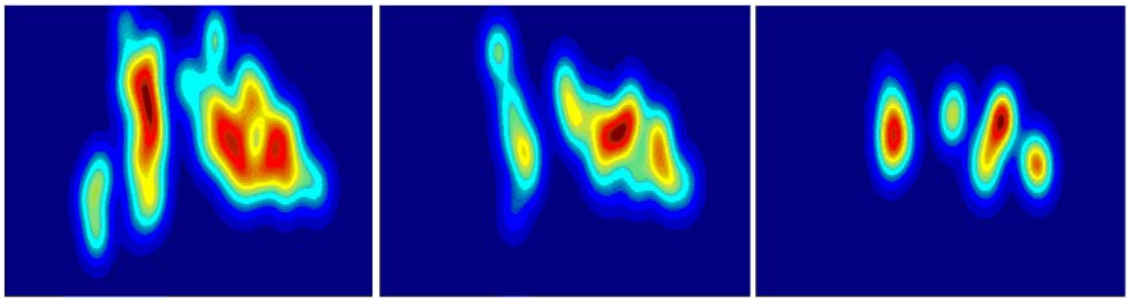

Figure 4.14 Different representations of cluster boundaries generated by SVC with different values for the outlier parameter C. Values for C in the above images are 0.50, 0.65, and 0.85. Bandwidth was held constant (q=12).

4.1.3 SVC and Parameter Value Selection

A major criticism of SVC is related to the difficulty of selecting appropriate parameter values. With SVC as with Gaussian kernels, parameter values must be selected for the bandwidth σ as well as for the parameter C that controls the influence of outliers. To generate the results for the simulated data used for this analysis of SVC, a variety of parameter settings were examined and adjusted according to the degree of clustering and noise in each frame. As shown in Figures 4.8-4.10 and Figures 4.11-4.14, the choice of parameter values can have a major affect on the representation of clustering, changing the number of individual clusters.

Given that clustering is largely a perceptual phenomenon (Kulldorff, 1997), determination of a "best" representation presents real challenges. Nonetheless, with KDE

an established tool for spatial analysis and with KDE also involving a bandwidth parameter σ that is also a feature of SVC, some guidance may be obtained from research already conducted on bandwidth selection from KDE. One example is that of asymptotic mean integrated squared error (AMISE) (Waller and Gotway 2004). Other techniques developed from KDE, such as varying bandwidths or weighting kernels according to underlying factors such as population, also warrant experimentation.

Perhaps more problematic than the determination of a value for bandwidth is the issue of selecting an appropriate value for *C*. By indirectly defining the percentage of outlying points (Ben-Hur et al, 2001), *C* also plays an indirect role in the determination of the representation of the clusters (Figures 4.13-4.14). However, in this regard SVC suffers from many of the same problems as other algorithms for unsupervised learning in that the learning process requires assumptions a priori that have an impact on results. K-means clustering is another such example where the selection of a value for *k* plays a large role in the assignment of points to clusters.

One potential means for determining a value for *C* can also answer the second of the questions posed in the introduction regarding statistical significance. Throughout this thesis we have focused on the third of these questions, addressing the ability to represent non-linear clusters for spatiotemporal analysis of deformation and movement. Statistical significance is central to several spatiotemporal research domains and whether or not a cluster is significant carries real weight in an application. For example, Kulldorff and Athas' (1998) investigation of potential brain cancer clusters in New Mexico with the scan statistic found no significant clusters which effectively barred brain cancer patients in the area from millions in federal funds. The scan statistic tests for significance using a

likelihood ratio (Eq. 2.4). The likelihood ratio is a function of observed (low-level) events over the expected number of events according to the application. A similar technique could be applied for SVC. Once an appropriate bandwidth was selected, C could be determined by scanning through values and testing for significance against likelihood ratios. If statistically significant clusters were found, these could be used to represent the clustering behavior in a point process. For example, clusters could be extracted that are significant at the $\alpha = 0.01, 0.05, 0.10$ levels.

Nonetheless, the determination of guidelines for the selection of appropriate parameter values is an on-going topic of research (e.g., Lee and Daniels 2004). Parameter selection is complex in that both the bandwidth parameter σ and the limit on percentage outliers C must be updated and evaluated in tandem (i.e., changing q requires a corresponding appropriate change in C). Given that the selection of these parameters impacts representations of higher-level cluster events, methods will have to be established to handle these sources of implicit bias before SVC can be suggested for widespread application.

4.2 Interpolating Higher-Level Events in Geosensor Data with SVMs

This section outlines how higher level areal events can be approximated from point data produced from geosensor networks using SVMs. For SVMs to be applied, it is only assumed that the sensor locations are known and that the Boolean values indicating whether or not a low-level event is occurring[19] can be communicated to a location where the SVM algorithm can process the data. These assumptions are not seen as unreasonable since many applications require tracking of sensor locations and since communication of Boolean value implies only minimal functionality. During deployment, the position of each sensor could be recorded and stored or the sensors could locate themselves using GPS and communicate their location. Boolean values could be communicated to a central location in centralized networks or to a "sink node" with processing capabilities in decentralized networks for the derivation of SVM representations of the higher level events. Both the location and Boolean label serve as the input into the SVM algorithm.

As with SVC, the SVM algorithm involves kernel transformation and can be interpreted using KDE. Correspondingly, as in the last section, visual representations of KDE are compared against those produced by SVMs. The key difference between the geosensor point data for which the SVM methods are described in this section and the SVC methods for point processes outlined in the previous section is the accompaniment of a Boolean label with the coordinate data in the SVMs algorithm. This additional

[19] For the purposes of this research low-level are assumed to have some endurance through time. For example, low-level events in this research considered to represent time periods where sensor readings surpassed a given threshold or followed a given trend direction. These low-level event definitions contrast those that are immediate and non-enduring such as inflection points in time series.

attribute renders the geosensor data to a form analogous to a training set and establishes where boundaries exist.

While able to produce non-linear representations of higher-level events without explicit bias, this SVM approach also offers conceptual benefits. Intuitively, when tasked with interpolating across an area from geosensor network data where the nodes are distributed through space one would expect the validity of these readings to diminish with distance (into sensor-less space) from each sensor. By conducting a kernel transformation at each sensor point (with a Gaussian or radial basis kernel function), the SVM algorithm incorporates this concept. As with KDE and SVC, SVMs sum the kernel evaluations at each point in order to make inferences about the area beyond observation. However, with the points being labeled, effectively assigning the sensors into classes (+1 in-event, -1 non-event), the process involves the summation of positive and negative (rather than just positive as was done with SVC) values.

The approach treats the data from each sensor node as an element of a training set and the rest of the study area as points with unknown labels. As with KDE and SVC, the kernel transformation in SVMs results in a feature space that can be represented as a raster. In this feature space, the maximum margin separating hyperplane is derived as a function of the points located along the boundary (the support vectors). With the assignment of the weights α_i, kernel evaluations of each class of like-labeled points are summed and boundary estimates are defined as the locations where the sum equals zero. This process is outlined in Figures 4.15-4.25. In order to communicate the process of this approach, data from a small hypothetical geosensor network are used (see Figure 4.15).

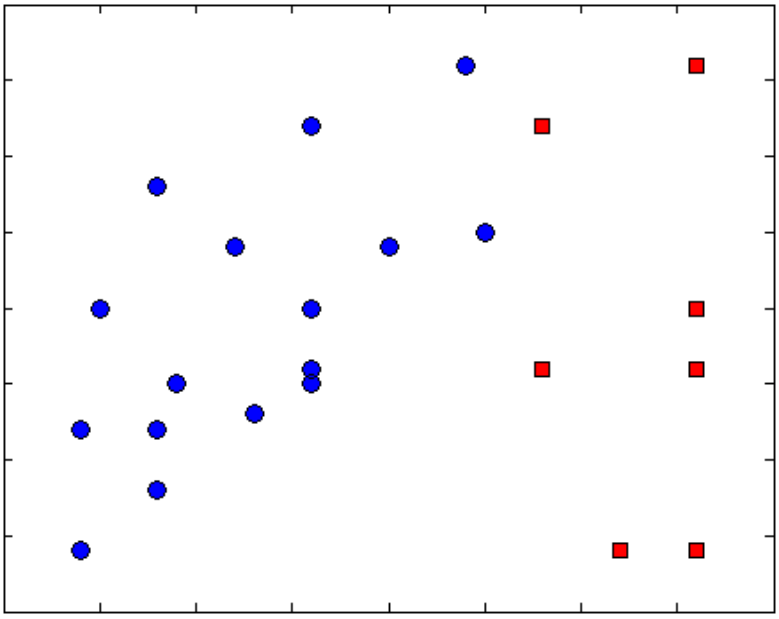

Figure 4.15 A hypothetical geosensor network displaying Boolean values for lower-level events. The blue dots represent the sensors not detecting low-level events while the red squares represent nodes that are detecting low-level events.

4.2.1 Visual Interpretation of SVM Results

To help interpret the SVM algorithm Figures 4.16-4.21 represent KDE results for both the lower-level event and non-event sensors independently. Figure 4.16 depicts what KDE would look like for the lower-level event subset of sensor locations and Figure 4.17 depicts what KDE would like for the non-event locations. Figure 4.20 depicts what KDE would look like for the non-event sensors when the labels (-1 for non-event points) are incorporated. Complementing all of these figures are cutaways (Figures 4.17, 4.19, and 4.21) which show what the underlying kernel evaluations look like. These are true cutaways (from the center of the study areas along the x-axis) and therefore give the appearance of unequal weighting (as results later from the SVM results). However, they merely reflect location in terms of both the x and y dimensions.

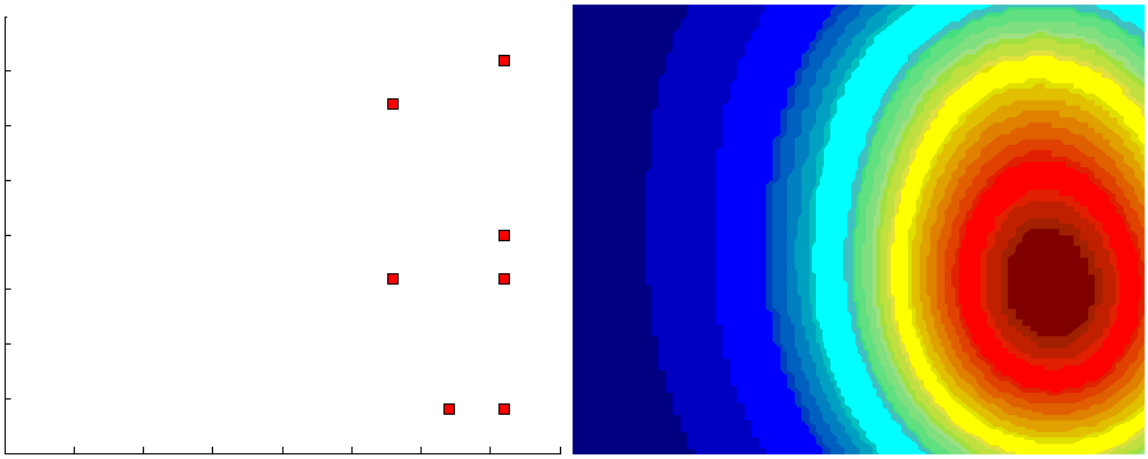

Figure 4.16 In-event sensor locations and KDE representing the relative concentration at sensor locations. The red regions denote areas of higher concentration and the blue regions denote areas of scarcity.

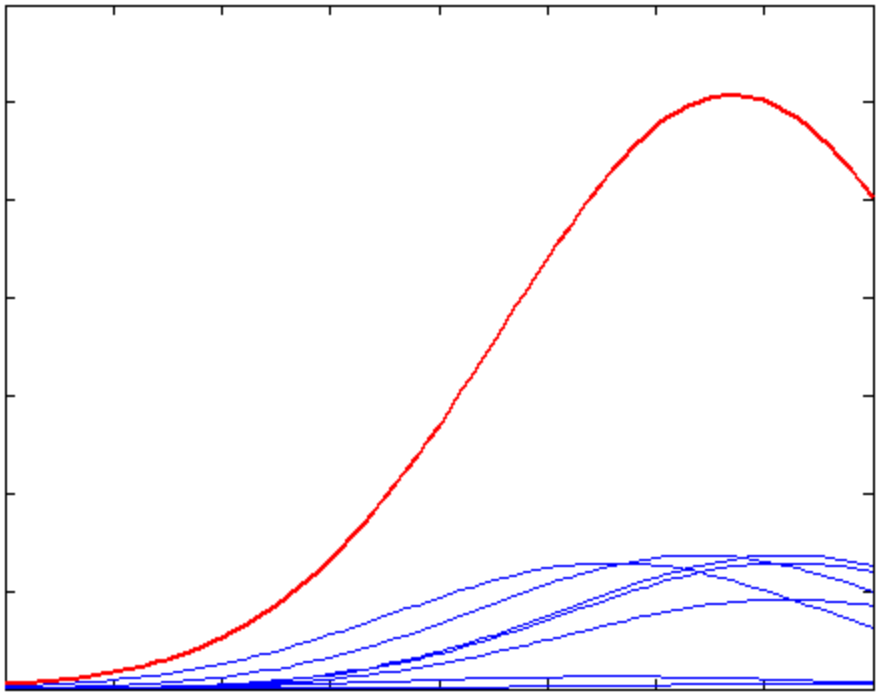

Figure 4.17 Cutaway of the KDE shown in Figure 4.16. Rather than collapsing a dimension as in Figure 4.3 and Figure 4.7 in the previous section, this image represents a cutaway from two-dimensional KDE (taken at the midpoint of the x-axis in Figure 4.16). All points are given equal weight, but distance in the hidden dimension gives the illusion of unequal weighting.

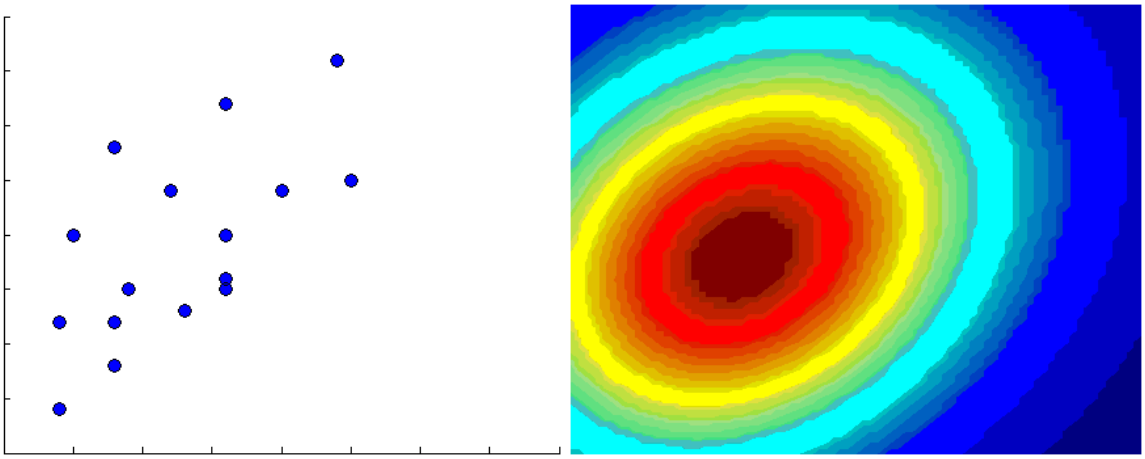

Figure 4.18 Non-event sensor locations and KDE representing the relative concentration of sensor locations. The red regions denote areas of higher concentration and the blue regions denote areas of scarcity.

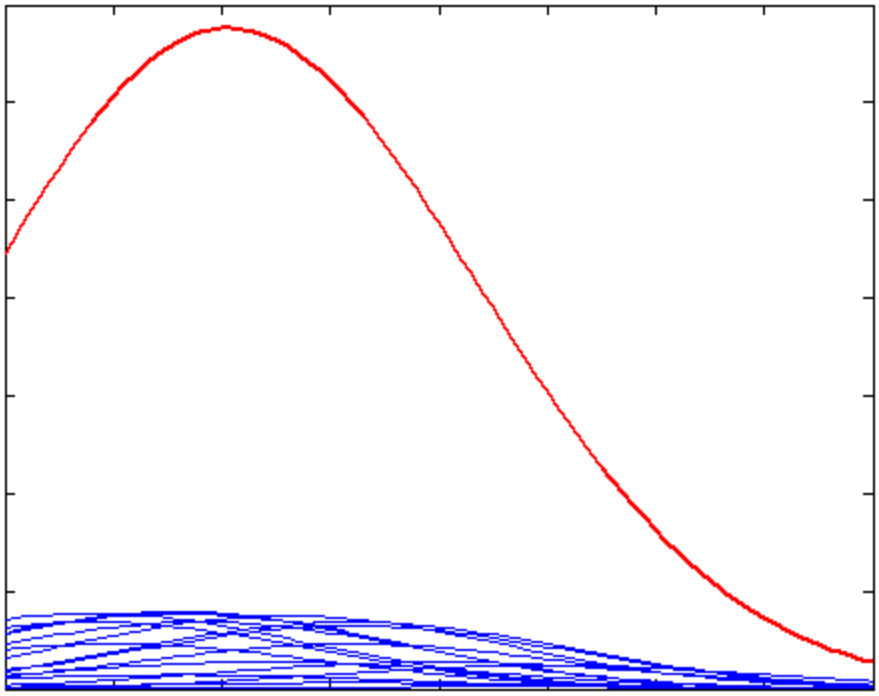

Figure 4.19 Cutaway of the KDE shown in Figure 4.18. As in Figure 4.17 all points are given equal weights.

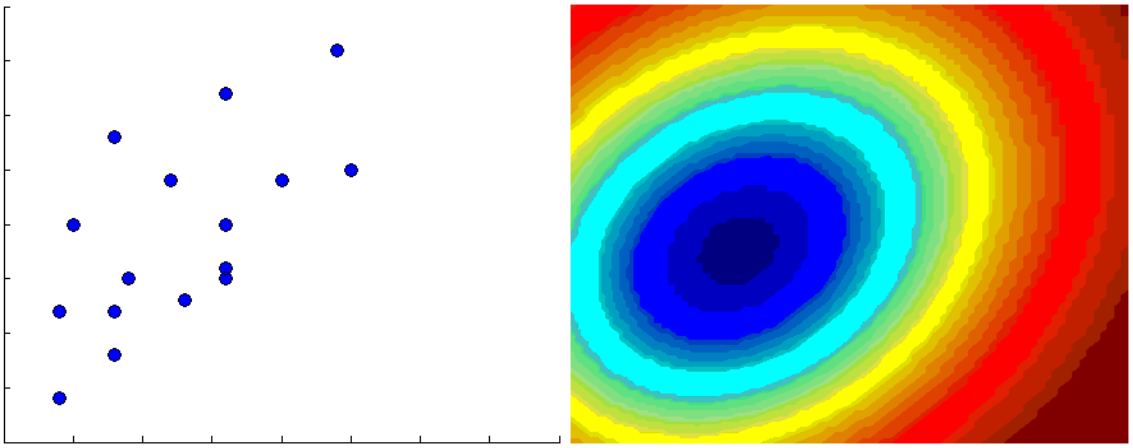

Figure 4.20 Non-event sensor locations and KDE representing the relative concentration of sensor locations and sensor labels. Note the change in the colors relative to Figure 4.19 due to the multiplication of the kernel evaluations by the negative non-event label.

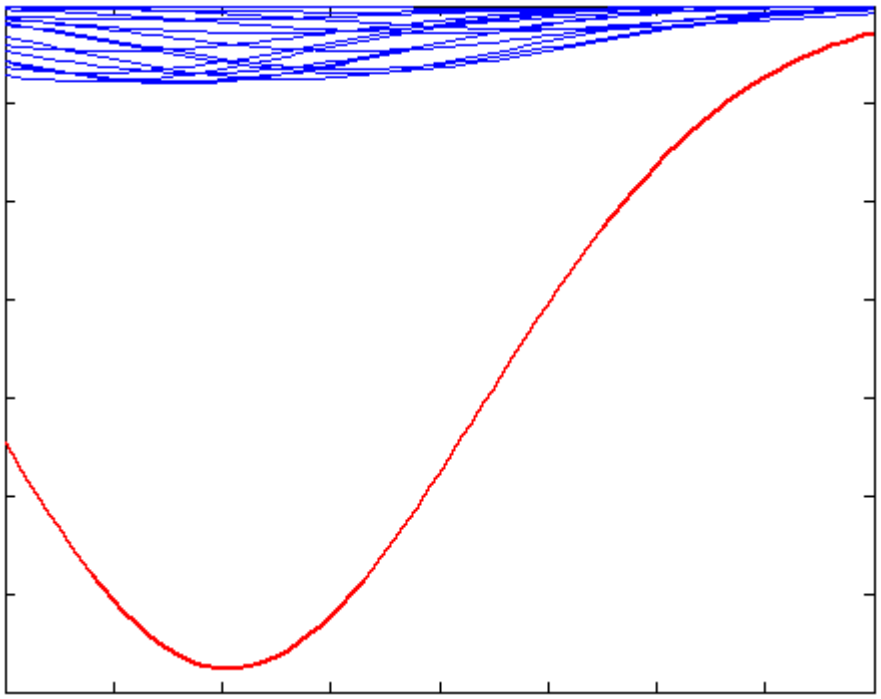

Figure 4.21 Cutaway of non-event sensor locations and KDE representing the relative concentration of sensor locations and sensor labels (negative/non-event according to the lower-level event predicate).

With the SVM algorithm (Eq. 3.20), the weights α_i are derived. By effectively reducing the number of data points to only those with the most relevant information - those located along the boundary – the weights reduce the number of calculations necessary to complete what would otherwise be a more computationally expensive process. In addition, in the context of geosensors, the identification of boundary nodes offers an additional benefit of highlighting local neighborhoods where nodes sleeping for power conservation could be activated to achieve greater spatial precision in later representation of higher-level events. The effect of introducing the weights is presented in Figures 4.22-4.23.

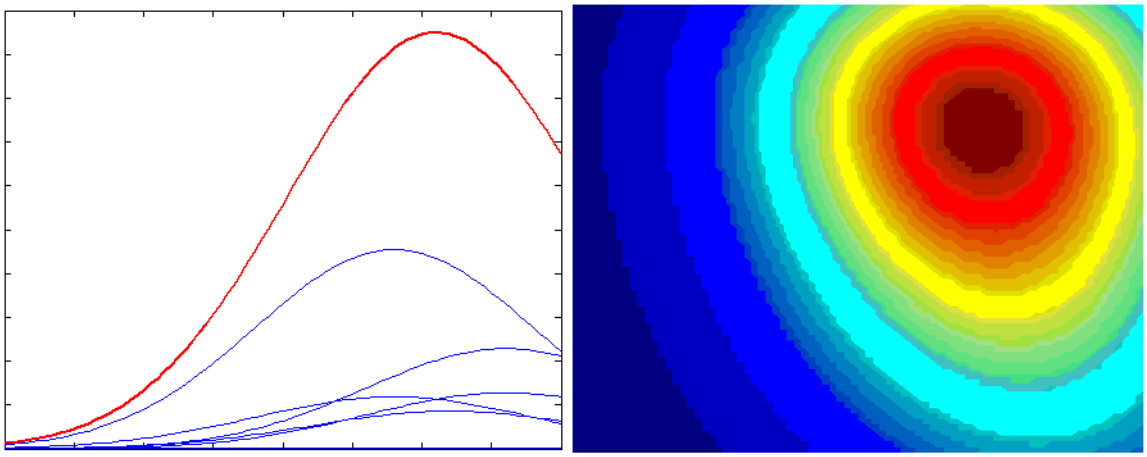

Figure 4.22 Cutaway and raster of in-event weighted kernel evaluations.

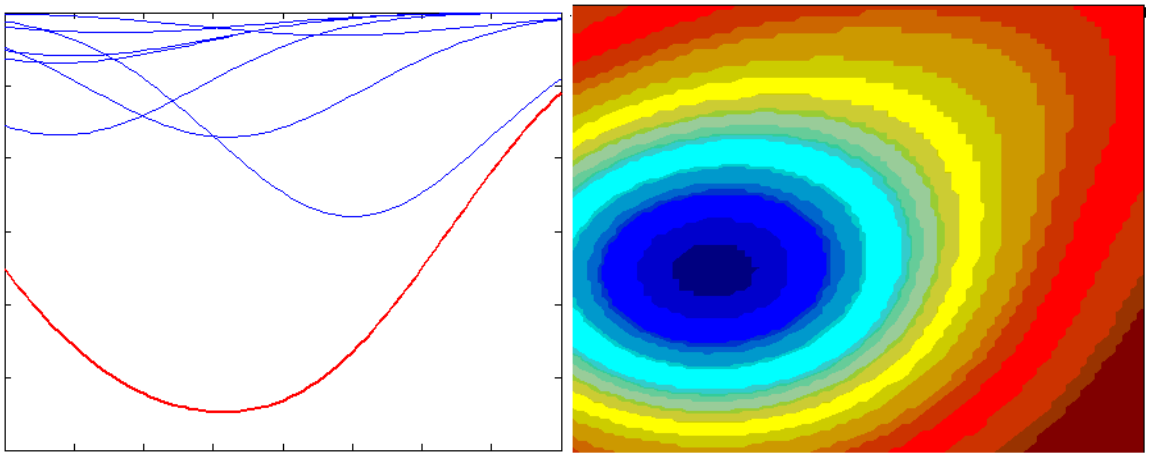
Figure 4.23 Cutaway and raster of non-event weighted kernel evaluations.

To derive the boundary, which is interpreted as a representation of the spatial extent of higher-level events, the in-event and non-event weighted kernel evaluations are summed. This is shown in the cutaway from the center of the study area in Figure 4.24 and in the rasters in Figures 4.25. The maximum feature space (kernel evaluated) distance between classes is obtained where the sum is equal to zero and this is where the SVM-generated boundary between the classes is estimated.

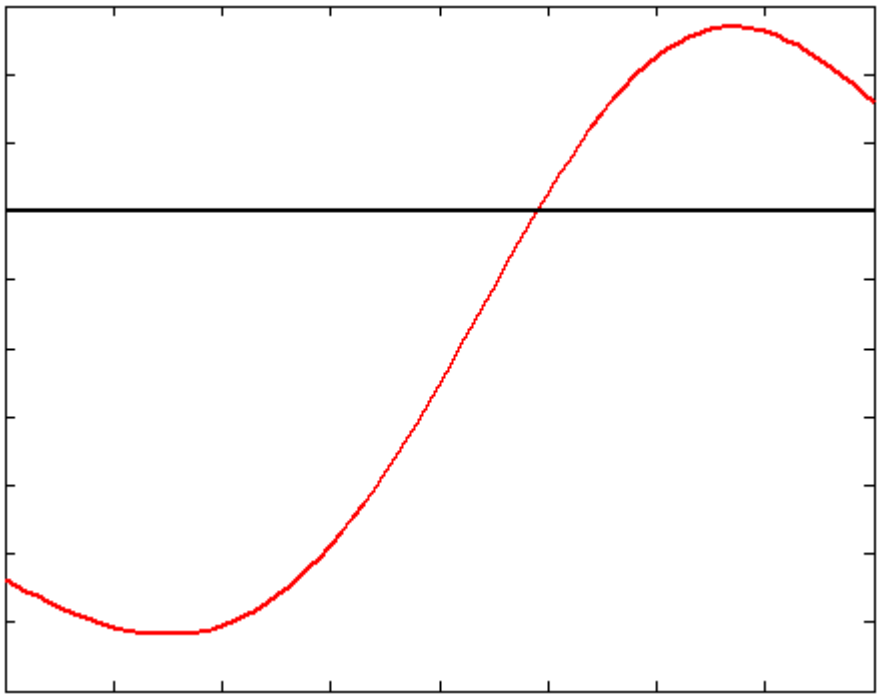

Figure 4.24 Summing of in-event and non-event weighted kernel evaluations. The boundary is defined as those locations where the sum is equal to zero (black line).

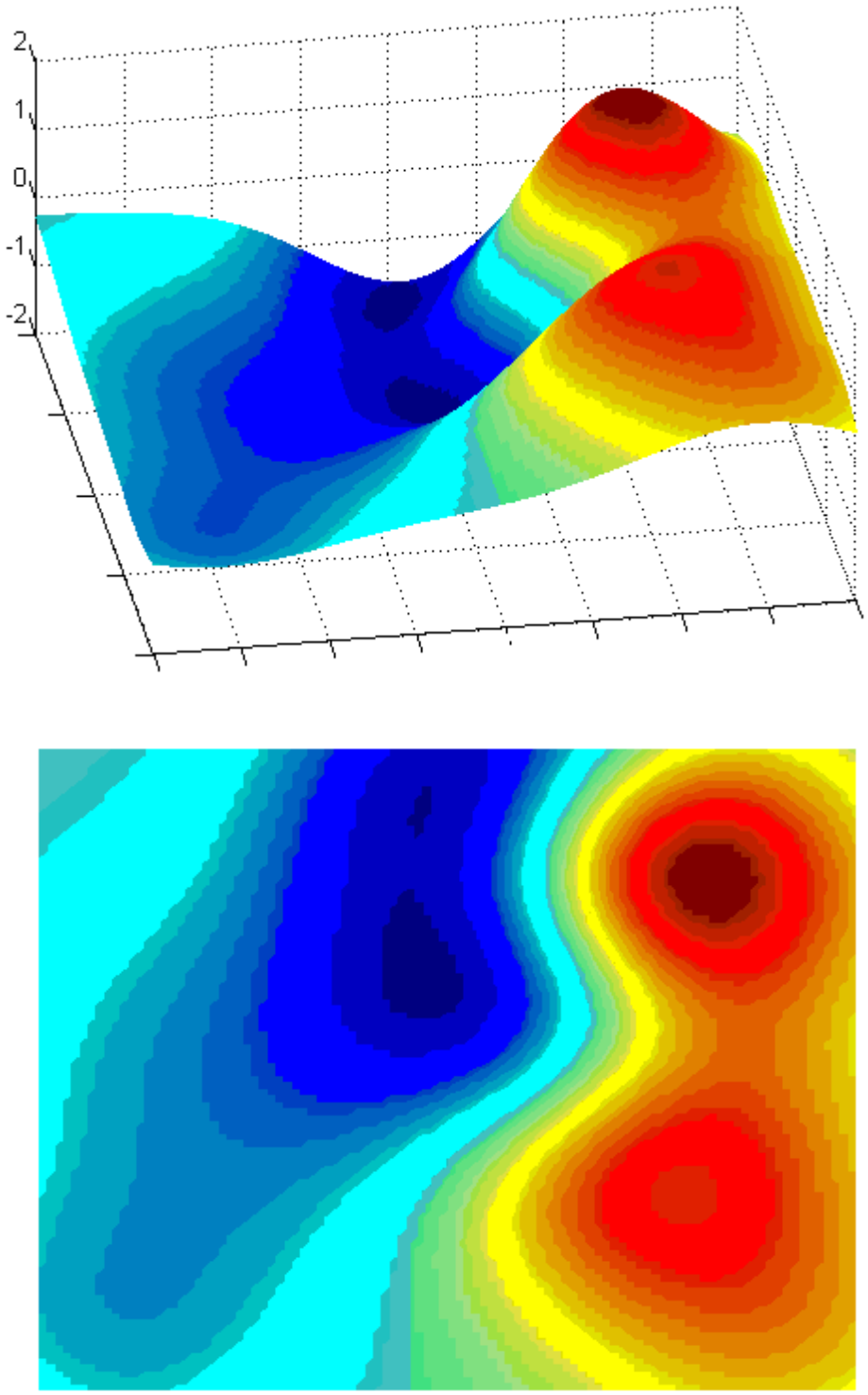

Figure 4.25 SVM raster results. These rasters depict the sums of the weighted kernel evaluation for the in-event and non-event sensors. The blue regions are negative valued (non-event) areas and the red regions are positive valued (in-event) areas.

Results from the SVM used in this demonstration delimit four regions which essentiallyresult from drawing contours on the raster. Two are delimited by the boundary and correspond to the regions of positive and negative values within the raster (separated by the thick black line in Figure 4.26). The other two are subsets of these regions and are bounded by the support vectors and represent the locations where the raster is greater than 1 or less than -1. The lines passing through the support vectors are equal to 1 or -1 and delimit these regions. Each of these four regions (i.e., the two separated by the black line decision function and the two bounding the support vectors) correspond to those depicted in the feature space representation in Figure 3.3. Results for the simulated network appear in Figure 4.26.

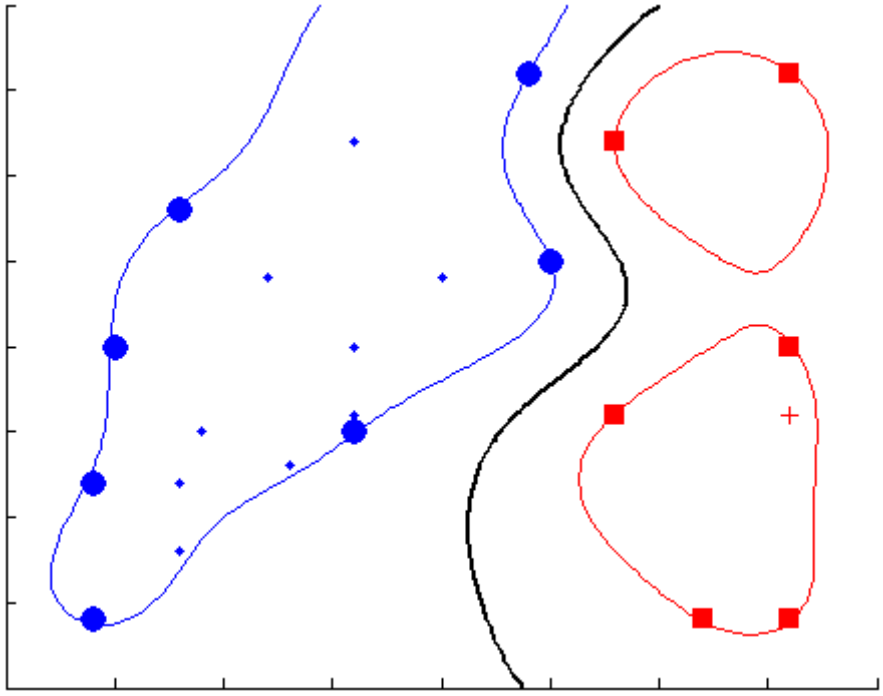

Figure 4.26 SVM generated representation of a higher-level event boundary. The decision boundary appears as the thick black line. The lighter red and blue lines depict the locations where the raster is equal to 1 and -1 respectively and correspond to the regions delimited in the hypothetical feature space shown in Figure 3.3.

4.2.2 SVMs and Selection of Parameter Values

As with SVC, a major criticism of SVMs is related to the difficulty involved in selecting appropriate values for parameters. While not subject to explicit bias in terms of shape, SVMs do involve implicit bias related to the selection of parameter values. With Gaussian kernels, the most commonly applied kernel function and the kernel used in this thesis, a principal parameter is that for bandwidth. Varying bandwidth can have a dramatic effect on the representation of boundaries (Figure 4.27). The problem of parameter selection is a persistent pattern analysis problem (e.g., Wang et al., 2003; Ban and Abe, 2004) and will be central to the implementation of SVM methods in spatial analysis in geosensor networks.

In geosensor networks, one possible source for assistance in determining bandwidth could be the concept of a sensing radius. Sensing radii could be used to give an approximation of how far it may be appropriate for the readings collected at one sensor location to contribute to the determination of values at locations without sensors. Other sources that could inform selection could be derived from domain knowledge regarding the nature of the application. An objective of future research will have to involve the establishment of guidelines for bandwidth selection.

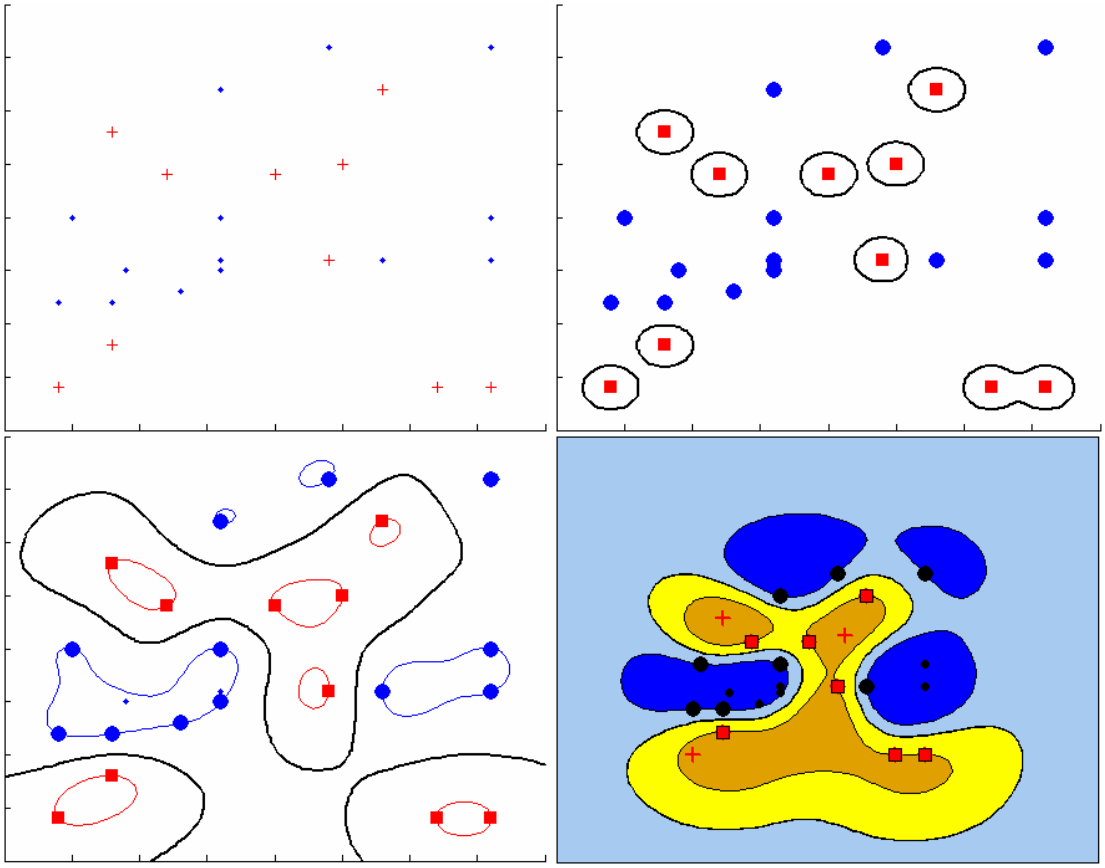

Figure 4.27 A second simulated geosensor network and the effect of varying bandwidth. In the first frame, the red crosses represent in-event sensor readings and the black dots represent non-event readings. The three frames show SVM estimates for the spatial extent of events with bandwidths of $\sigma = 10, 50, 70$.

As mentioned at the beginning of this section, an assumption is that each sensor can communicate a Boolean response indicating whether or not the sensor witnessed a low-level event or not. While in the above demonstrations it was assumed that the Boolean values recorded were accurate, a common problem in sensor networks, especially wireless sensor networks, is that the sensors are prone to error. With SVMs the soft margin parameter *C* can help address this issue (see Section 3.5.5). By allowing for values that are relatively extreme when compared to the rest of the distribution to appear on the opposite side of the decision function, errors in reporting can be accommodated and prevented from exerting inappropriate influence on the decision

boundary. The effects of varying *C*, however, also impact the shape of the boundary (see Figure 4.28). While attractive for these conceptual reasons, in practice the determination of *C* presents a problem. Values for C could be varied in accordance with expectations regarding the reliability of sensors relative to the level appropriate for the application, but, as with σ, a general framework needs to be developed for the selection of parameter values before SVMs can be widely applied in applications in geosensor networks.

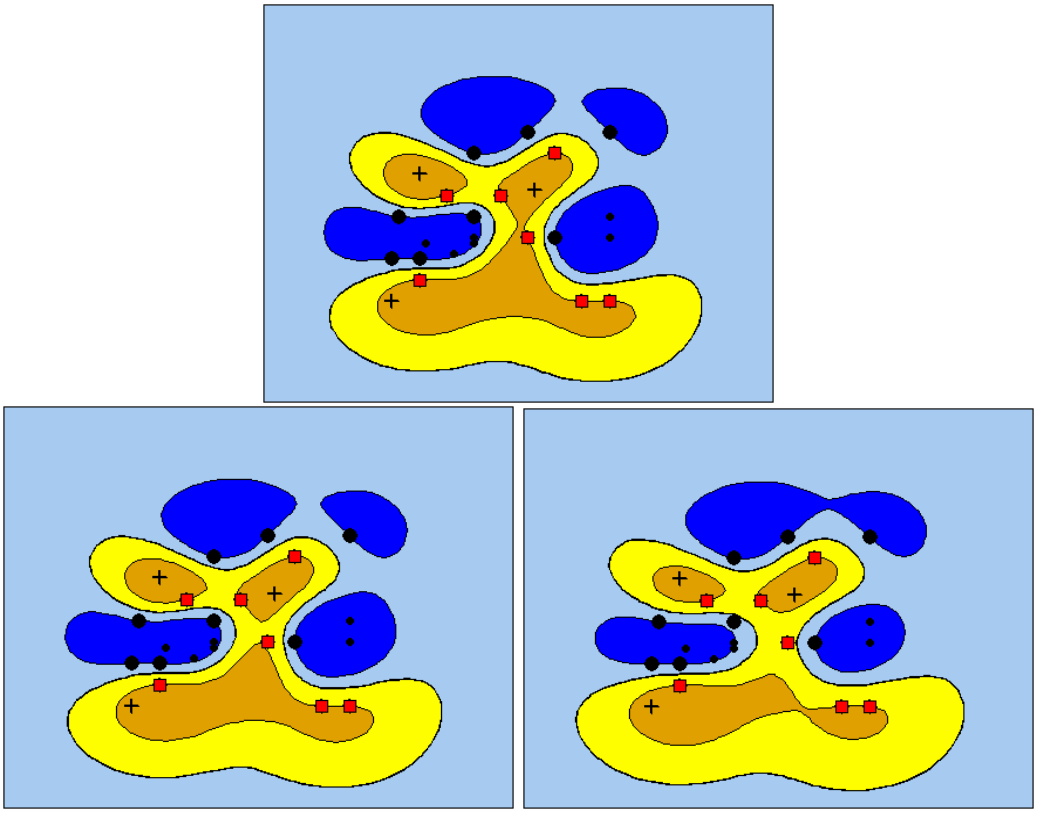

Figure 4.28 The effect of varying *C*. The three frames show SVM estimates for the spatial extent of events with constant bandwidths and decreasing values for *C*.

4.3 SVMs for Spatiotemporal Analysis: An Application-Based Example

To analyze the application potential of the SVM methods for spatiotemporal analysis, this section examines how well these methods are able to approximate irregular polygons derived to describe the shape of event over time in real-world settings. The event examined is a forest fire in Shenandoah National Park (files accessed from Shedd, 4/16/2009).

The analysis is based on three frames of the fire (Figure 4.29). The analysis considers the effect of sampling and parameter selection on the relative ability of the SVM methods to reproduce the original polygons. In other words, the time series of polygons are considered to be a "ground truth" and the SVM generated boundaries are compared against these ground truths in order to analyze the ability of these techniques to represent the actual spatial extent of events at different periods of time.

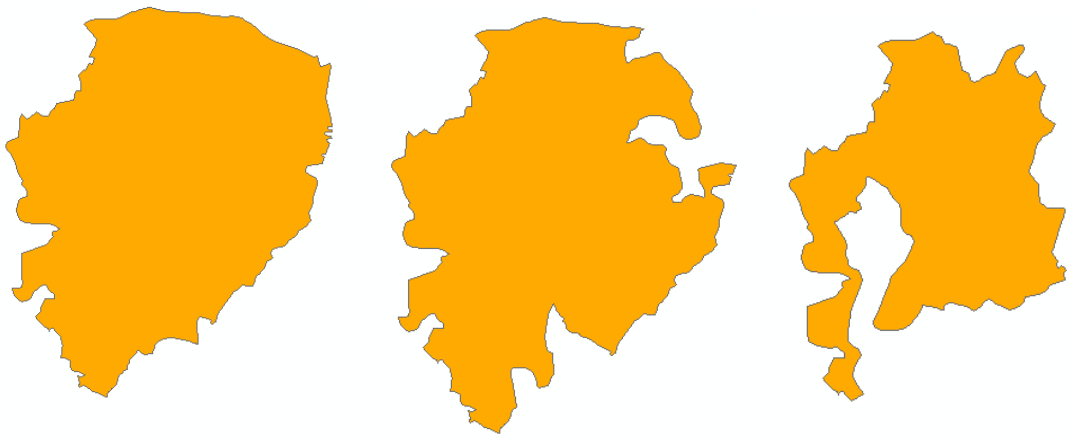

Figure 4.29 Three frames of fire boundaries analyzed with SVMs.

The analytic process began in ArcGIS where the polygons were imported. In ArcGIS random points were generated using Hawth's Tools. Hawth's Tools includes a random point generator that allows enforcement of constraints regarding the minimum distance between generated points. It is thought that this would better resemble potential

application scenarios where it would be unlikely that one sampling point would be located in immediate proximity to another. For each of the three frames of the fire, three different sampling schemes were created in order to investigate the effect of increasing the number of points on SVM's ability to describe the fire polygons. To gain insight into how effective SVMs might be in replicating the "ground truth" fire polygons, thirty different sets of points for each of the sampling schemes were analyzed. Frames from these results are presented in Figures 4.30-4.37. For each of the sets of results, the bandwidth was set so that it is equal to the average distance among all of the points and the number of points.

$$\sigma = \frac{average\ distance}{\sqrt{number\ of\ points}} \tag{4.6}$$

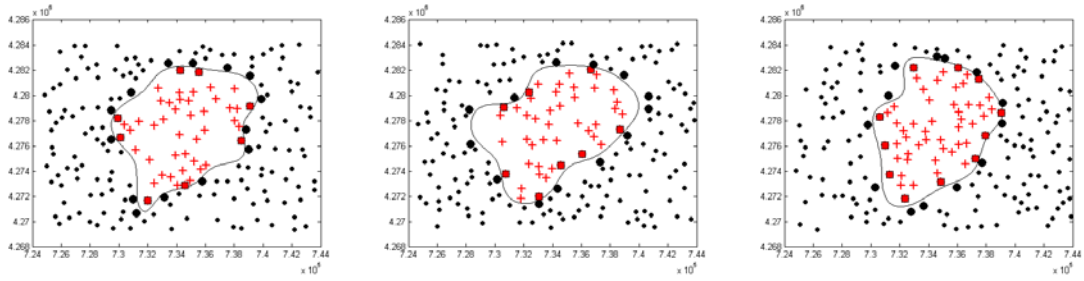

Figure 4.30 Results for the first of the three frames in 4.29 with 200 points. The support vectors appear as the red boxes or the larger black dots. The red crosses and the smaller black dots are the interior points.

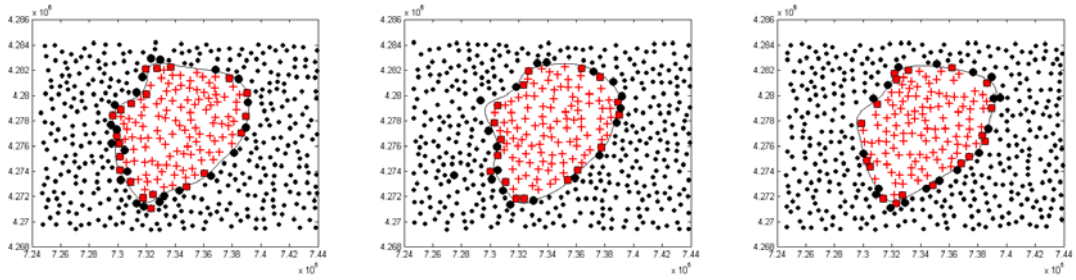

Figure 4.31 Results for the first of the three frames in 4.29 with 500 points. The support vectors appear as the red boxes or the larger black dots. The red crosses and the smaller black dots are the interior points.

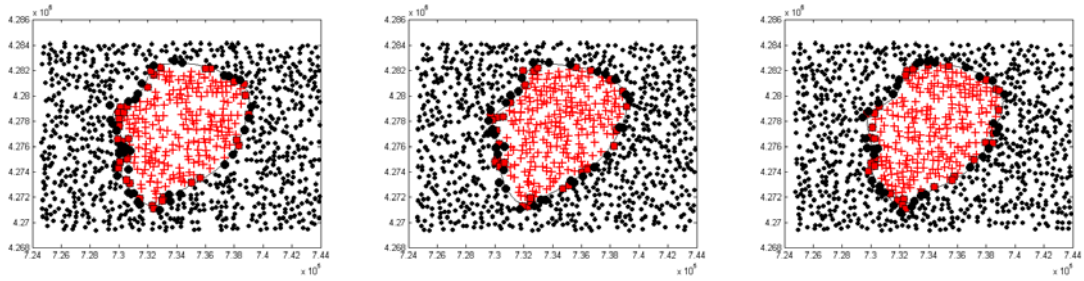

Figure 4.32 Results for the first of the three frames in 4.29 with 1000 points. The support vectors appear as the red boxes or the larger black dots. The red crosses and the smaller black dots are the interior points.

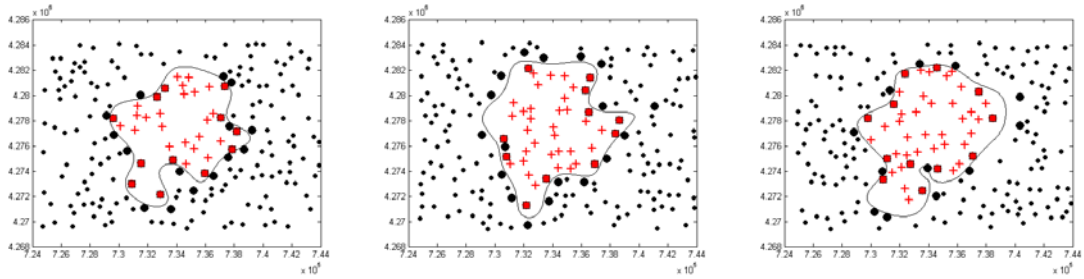

Figure 4.33 Results for the second of the three frames in 4.29 with 200 points. The support vectors appear as the red boxes or the larger black dots. The red crosses and the smaller black dots are the interior points.

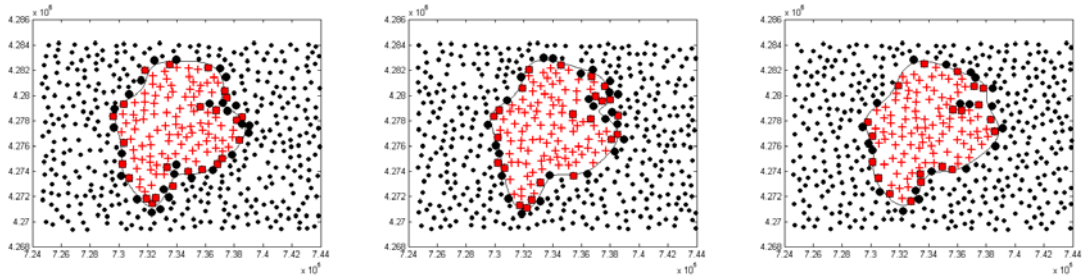

Figure 4.34 Results for the second of the three frames in 4.29 with 500 points. The support vectors appear as the red boxes or the larger black dots. The red crosses and the smaller black dots are the interior points.

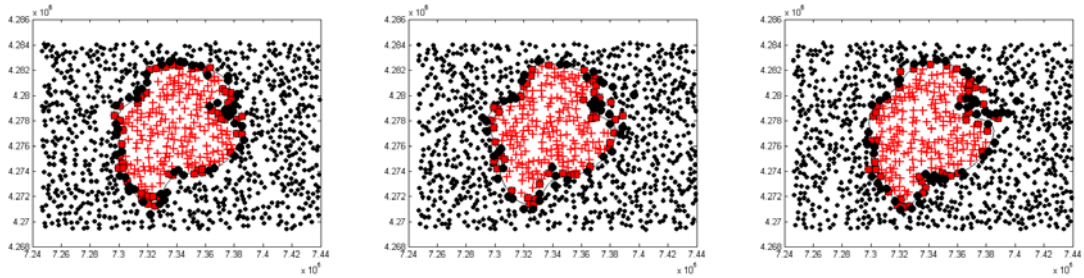

Figure 4.34 Results for the second of the three frames in 4.29 with 1000 points. The support vectors appear as the red boxes or the larger black dots. The red crosses and the smaller black dots are the interior points.

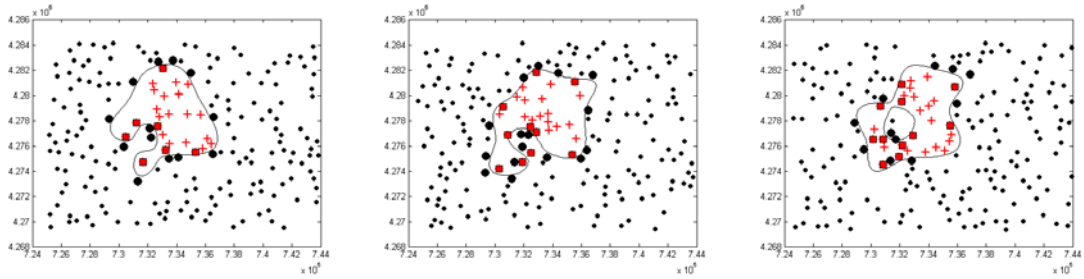

Figure 4.35 Results for the third of the three frames in 4.29 with 200 points. The support vectors appear as the red boxes or the larger black dots. The red crosses and the smaller black dots are the interior points.

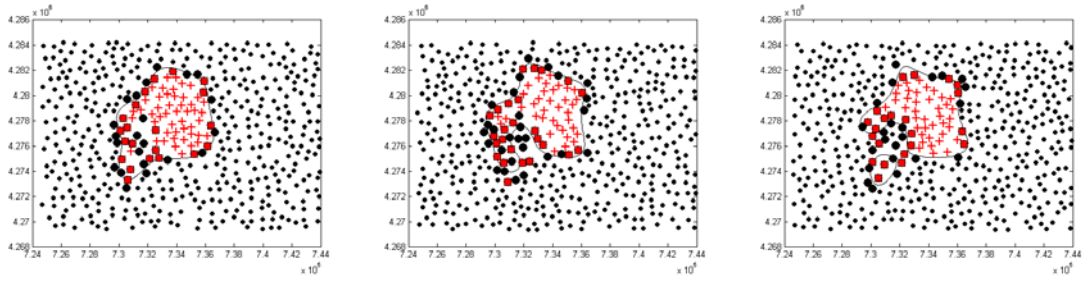

Figure 4.36 Results for the third of the three frames in 4.29 with 500 points. The support vectors appear as the red boxes or the larger black dots. The red crosses and the smaller black dots are the interior points.

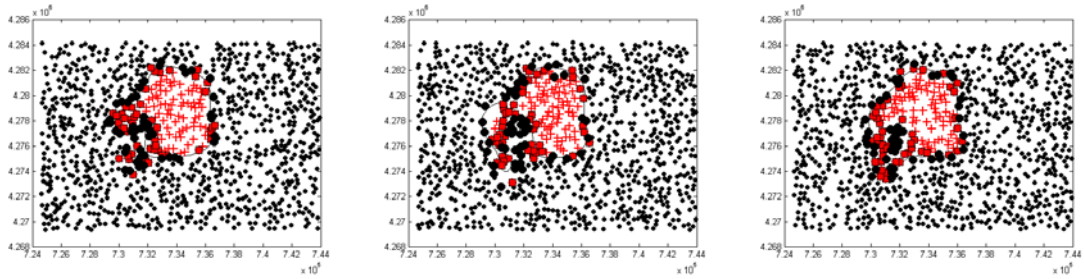

Figure 4.37 Results for the third of the three frames in 4.29 with 1000 points. The support vectors appear as the red boxes or the larger black dots. The red crosses and the smaller black dots are the interior points.

In each of the three frames it is evident that there is a significant amount of variability in each of the representations derived from the data with 200 points. In contrast, the representations derived from 500 and 1000 points were more consistent. In addition to offering analysis of a time series, the three frames examined allow for discussion regarding SVMs ability to represent complex polygons. The first of the polygons (first frame in Figure 4.29) is relatively simple with fewer irregularities. The third polygon is significantly more complex with narrow extensions protruding from the center of mass. The SVM methods were able to capture the complexity in both the second and third polygons. However, it does appear that the representation derived from the samples of 500 points provide a better approximation than those generated from samples with 1000 points. It is thought that this is due to the consistent application of the formula for the bandwidth. Adapting the method for the derivation of a bandwidth parameter may offer better approximations of the "ground truth" for more complex polygons. As emphasized in other sections of this thesis, this remains a topic of research for SVMs, a consistent method for the derivation of a bandwidth was used to illustrate this point.

Future research could examine other means for extracting an appropriate value for the bandwidth. An iterative approach, resembling that taken by Ben-Hur et al. 2001 where the number of polygons that result from the SVM are tracked in relation to the bandwidth. In controlled settings, where a set of polygons are taken as a "ground truth," these results could then be tested for their ability to represent the actual underlying event. Comparison between the SVM-produced and "ground truth" could be conducted using statistical methods such as the L2 error norm

$$L^2_{error\ norm} = \frac{area((O-P)\cup(P-O)}{area(O)} \tag{4.7}$$

where O is the area of the "ground truth" polygon and P is the area of the test polygon derived from the SVM (Sadeq and Duckham, forthcoming). Such testing could inform parameter selection.

Chapter 5

CONCLUSIONS

With spatiotemporal analysis promising improved description of the behavior of phenomena, the development of methods capable of representing spatiotemporal behaviors has long been a research objective. Recent advances in computational power have enabled the development of promising new approaches to spatiotemporal analysis. Among these methods is the spatiotemporal helix which can describe movement and deformation of phenomena. Initially designed for analysis of video sequences of images, the spatiotemporal helix was derived from series of areal extractions composed of groups of pixels.

Many spatiotemporal data, however, occur in point form. Given that analysis is not possible directly from point clouds, research efforts have framed spatiotemporal point data in terms of higher-level areal events that aggregate over lower-level point-based events. Considering that these higher level events are in areal form, just as extractions of events from image data, they could serve as input into existing techniques for spatiotemporal analysis such as the spatiotemporal helix. Given an emphasis on deformation behavior, at the crux of such an analytical effort would be the translation of groups of points into areal representations that accurately follow the distribution of points.

The purpose of this thesis was to introduce two newly developed machine learning approaches that estimate the spatial extent of higher-level events from point data. Support vector clustering (SVC), a technique for unsupervised learning, was

suggested as a means for deriving the areal extent for spatiotemporal point data. Support vector machines (SVMs), a supervised learning method, is proposed for analysis of geosensor network data. Both techniques rely upon kernel transformations to introduce non-linearity. Being non-linear, these techniques are capable of producing representations that do not impose a shape a priori upon the representations of higher-level events.

The algorithms for each of these techniques were implemented and compared with KDE in terms of their intermediate results. In the case of both SVC and SVMS, final results illustrated capability for producing representations that follow the distribution of lower-level event points. While these highly non-linear representations avoid explicit bias in terms of shape, they are affected by implicit bias in the selection of parameter values. In SVC, where only one class of data is considered (i.e., only lower-level events) this form of bias may be severe. The simulations showed representations that contained varying numbers of polygons as well as different shapes. This variation may be sufficient to bias interpretations of spatiotemporal behavior. Given that the geosensor network form of spatiotemporal data involves two classes of point-based events (i.e., events and non-events), the effects of parameter selection were less dramatic. The presence of points from the other class restricts the influence of parameters on the representation of higher-level events. However, the results from the comparison of SVM-generated polygons against hypothetical "ground truth" polygons revealed that the potential for bias resulting from parameter selection remains an issue with SVMs.

Future work using SVC and SVMs should be focused on the development of methods for the derivation of appropriate parameter values. Again, with KDE being

similar to SVC and SVMs, some guidance may be obtained from existing research. Techniques such as asymptotic mean squared integrated error (AMISE) have already been applied in KDE (Waller and Gotway, 2004) and may offer potential for application with both SVC methods. A challenge in developing such techniques for point process data, however, is that clustering is largely a perceptual phenomenon whose true shape cannot be known (Kulldorff, 1997). Depending on the application, different statistical tests of significance could be applied to SVC. An example of such tests could be the likelihood ratio implemented by the scan statistic which could be used to help derive valid representations of clusters.

With geosensor network data, experiments could be conducted that could guide the selection of appropriate parameter values. For example, when analyzing the spatiotemporal behavior of air pollution SVM geosensor results for a given property could be compared to extractions from other data sources, such as remotely sensed images. By comparing expected results from the image data with the results from the geosensor network, information regarding appropriate parameter values could be evaluated and improved. Such analysis would complement that conducted in the last chapter of this thesis. It is thought that iterative methods examining the effect of bandwidth selection on both the number of polygons and the number of support vectors could be particularly valuable. In addition, further tests could be conducted with varying distributions of sensor nodes in order to see how the number and dispersal of sensors may affect results.

This thesis work suggests great potential for the application of SVC and SVMs for spatiotemporal analysis. The description of spatiotemporal behavior from point data

is relevant to a wide range of fields and the refinement of these techniques could be significant in enabling inferences describing spatiotemporal behaviors.

REFERENCES

Aamodt, G., S. O. Samuelson, A. Skrondal (2006). "A Simulation Study of Three Methods for Detecting Disease Clusters." International Journal of Health Geographics 5(15).

Aizermann, M., E. Braverman, L.I. Rozoner (1964). "Theoretical Foundations of the Potential Function Method in Pattern Recognition." Automation and Remote Control 25: 821-837.

Allen, J.F. (1983). "Maintaining Knowledge about Temporal Intervals." Communications of the ACM 26(11): 823-843.

Bartlett, P. (1998). "The Sample Complexity of Pattern Classification with Neural Networks: The Size of the Weights is More Imporatant than the Size of the Network " IEEE Transactions on Information Theory 44(2): 525-536.

Baxter, M. J., C. C. Beardah, R.V.S Wright (1997). "Some Archaeological Applications of Kernel Density Estimates." Journal of Archaeological Science 24(4): 347-354.

Beard, K., H. Deese, N.R. Pettigrew (2008). "A Framework for the Visualization and Exploration of Events." Information Visualization 00: 1-18.

Bennett, K. P. and C. Campbell (2000). "Support Vector Machines: Hype or Halleujah?" ACM SIGKDD Explorations 2(2): 1-10.

Ben-Hur, A., D. Horn, H.T. Siegelmann, V. Vapnik (2001). "Support Vector Clustering." Journal of Machine Learning Research 2: 125-137.

Berke, O. (2004). "Exploratory Spatial Relative Risk Mapping." Preventative Veterinary Medicine 71(3-4): 173-182.

Bithell, J. F. (1990). "An Apllication of Density Estimation to Geographic Epidemiology." Statistics in Medicine 9(6): 691-701.

Boser, B. E., I. Guyon, V. Vapnik (1992). A Training Algorithm for Optimal Margin Classifiers. 5th Annual ACM Workshop on Computational Learning Theory, ACM Press.

Braga, A. A. (2001). "The Effects of Hot Spot Policing on Crime." Annals of the American Academy of Political and Social Science 578(1): 104-125.

Brundson, C. (1995). "Estimating Probability Surface for Geographical Point Data: An Adaptive Kernel Algorithm." Computers and Geosciece 21(7): 877-894.

Burges, C. J. C. (1998). "A Tutorial on Support Vector Machines for Pattern Recognition." Data Mining and Knowledge Discovery 2: 121-167.

Chang, W., D. Zeng, H. Chen (2005). Prospective Spatiotemporal Data Analysis for Security Informatics. IEEE Conference on Intelligent Transportation Systems, Vienna, Austria.

Chintalapudi, K. K. and R. Govindan (2003). "Localized Edge Detection in Sensor Fields." Ad Hoc Networks 1: 273-291.

Christianini, N. and J. Shawe-Taylor (2000). An Introduction to Support Vector Machines, Cambridge Univesity Press.

Claramunt, C. and B. Jiang (2001). Hierarchical reasoning in time and space. Proceedings of the 9th Internation Symposium on Spatial Data Handling, Beijing.

Cliff, A. D. and J. K. Ord (1981). Spatial Processes: Models & Applications, Pion Ltd.

Cressie, N. (1992). Statistics for Spatial Data. Terra Nova.

D'Andrade, R. (1978). "U-Statistic Hierarchical Clustering." Psychometrika 4: 58-67.

Devroye, L., L. Gyorfi, G. Lugosi (1996). A Probabilistic Theory of Pattern Recognition, Springer.

Duckham, M., S. Nittel, M. Worboys (2005). Monitoring Dynamic Spatial Fields Using Responsive Geosensor Networks. ACM International Workshop on Geographic Information Systems, Bremen, Germany, ACM.

Duda, R. O. and P. E. Hart (1973). Pattern Classification and Scene Analysis. New York, Wiley.

Fisher, R. (1952). Contributions to Mathematical Statistics, Wiley.

Gowers, J. C. (1967). "A Comparison of Some Methods of Cluster Analysis." Biometrics 23: 623-628.

Guarlnik, V. and J. Srivastava (1999). Event Detection in Time Series Data. Proceedings of the 5th ACM SIGKDD International Conference on Knowledge Disovery and Data Mining, San Diego.

Hagerstrand, T. (1967). Innovation diffusion as a spatial process, University of Chicago Press.

Hazelton, N. W. G. (1991). Integrating Time, Dynamic Modelling and Geographical Information Systems: Development of Four-Dimensional GIS. Unpublished dissertation. Melbourne, University of Melbourne.

Hill, E. G., L. Ding, L. Waller (2000). "A comparison of three tests to detect general clustering of a rare disease in Santa Clara County, California." Statistics in Medicine 19(10): 1363-1378.

Huang, L., M. Kulldorff, D. Gregorio (2006). "A Spatial Scan Statistic for Survival Data." Statistics in Medicine In press, electronic version available ahead of print (www.satscan.org).

Johnson, S. C. (1967). "Hierarchical Clustering Schemes." Psychometrika 2: 241-254.

Jung, I., M. Kulldorff, A.C. Classen (2006). "A Spatial Scan Statistic for Ordinal Data." Statistics in Medicine In press, electronic version available ahead of print (www.satscan.org).

Kelsall, J. E. and P. J. Diggle (2007). "Non-Parametric Estimation of Spatial Variation in Relative Risk." Statistics in Medicine 14(21-22): 2335-2342.

King, B. F. (1967). "Step-wise Clustering Procedures." Journal of the American Statistical Association 62: 86-101.

Knox, E. G. (1964). "The Detection of Space-Time Interactions." Applied Statistics 13: 25-29.

Kulldorff, M., E. J. Feuer, B.A. Miller, L.S. Freeman (1997). "Breast Cancer in the Northeastern United States: A Geographical Analysis." American Journal of Epidemiology 146: 161-170.

Kulldorff, M. (1997). "A Spatial Scan Statistic." Communications in Statistics: Theory and Methods 26: 1481-1497.

Kulldorff, M., W. F. Athas, E.J. Feuer, B.A. Miller, C.R. Key (1998). "Evaluating Cluster Alarms: A Space-Time Scan Statistic and Brain Cancer in Los Alamos, New Mexico." American Journal of Public Health 88: 1377-1380.

Kulldorff, M. (2005). "SaTScan." Retrieved April 16, 2007, from www.satscan.org.

Kulldorff, M. (2006). SaTScan User Guide Version 7.0.

Kulldorff, M., L. Huang, L. Pickle, L. Duczmal (2006). "An Elliptical Spatial Scan Statistic." Statistics in Medicine In press, electronic version available ahead of print (www.satscan.org).

LeCun, Y., L. D. Jackel, C. Cortes, J.S. Denker, H. Drucker, I. Guyon, U.A. Muller, E. Sackinger, V.Vapnik (1995). Comparison of Learning Algorithms for Handwritten Digit Recognition. International Conference on Artificial Neural Networks.

Levine, N. and Associates (2007). CrimeStat 3. Houston, TX, U.S. Dept. of Justice (available from http://www.icpsr.umich.edu/CRIMESTAT/).

Mangasarian, O. L. (1965). "Linear and Nonlinear Separation of Patterns by Linear Programming." Operations Research 13: 444-452.

Miller, H. J. (1991). "Modelling accessibility using space-time prism concepts within geographical information systems." International Journal of Geographic Information Science 5(3): 287-301.

Mitchell, T. M. (1997). Machine Learning. Singapore, McGraw Hill.

Nowak, R. and U. Mitra (2003). Boundary Estimation in Sensor Networks: Theory and Methods. Second International Workshop on Information Processing in Sensor Networks, Palo Alto, Springer.

Nowak, R., U. Mitra, R. Willet, C.J. Burges (2004). "Estimating Inhomogenous Fields Using Sensor Networks." IEEE Journal on Selected Areas in Communications 22(6): 999-1007.

Parkes, D. and N. Thrift (1980). Times, Spaces, and Places: A Chronogeographic Perspective, John Wiley & Sons.

Peuquet, D. (1994). "It's About Time - A Conceptual Framework for the Representation of Temporal Dynamics in Geographic Information Systems." Annals of the Association of American Geographers 84: 441-461.

Pontil, M. and A. Verri (1998). "Support Vector Machines for 3D Object Recognition." IEEE Transactions on Pattern Analysis and Machine Intelligence 20(6): 637-646.

Pred, A. R. (1967). Behavior and location, Royal University of Lund.

Rogerson, P. (1997). "Surveillance systems for monitoring the development of spatial patterns." Statistics in Medicine 16(18): 2081-2093.

Rosenblatt, F. (1956). "The Perceptron: A Probabilistic Model for Information Storage and Organization in the Brain." Psychological Review 65: 386-408.

Sadeq, M. J. and M. Duckham (forthcoming). "Effect of Neighborhood on In-Network Processing in Sensor Networks."

Scholkopf, B., K. Sung, C. Burges, F. Girosi, P. Niyogi, T. Poggio, V. Vapnik (1997). "Comparing Support Vector Machines with Gaussian Kernels to Radial Basis Function Classifiers." IEEE Transactions on Signal Processing 45(11): 2758-2765.

Scholkopf, B. and A. Smola (2002). Learning with Kernels. Cambridge, MA, MIT Press.

Scott, D. W. (1992). Multivariate Density Estimation: Theory, Practice, and Visualization. New York, John Wiley & Sons.

Seaman, D. E. and R. A. Powell (1996). "An Evaluation of the Accuracy of Kernel Density Estimators for Home Range Analysis." Ecology **77**(7): 2075-2085.

Shawe-Taylor, J., P. Bartlett, R. Williamson, M. Anthony (1998). "Structural Risk Minimization with Over Data-Dependent Hierarchies." IEEE Transactions on Information Theory 44(5): 1926-1940.

Shawe-Taylor, J. and N. Christianini (2004). Kernel Methods for Pattern Analysis. Cambridge, UK, Cambridge University.

Shedd, Justin. Employee of North Carolina State University and researcher involved in analysis of fire data provided the Shenandoah fire polygonal data (4/2009).

Smith, F. W. (1968). "Pattern Classifier Design by Linear Programming." IEEE Transactions on Computers C-17: 367-372.

Song, C. and M. Kulldorff (2003). "Power Evaluation of Disease Clustering Tests." International Journal of Health Geographics 2(9).

Spatial Analysis Laboratory, Department of Geography, University of Illinois at Urbana-Champaign http://www.sal.uiuc.edu/stuff/stuff-sum/data.

Stefanidis, A., K. Eickhorst, P. Agouris, P. Partsinevelos (2003). Modeling and Comparing Change Using Spatiotemporal Helixes. ACM-GIS'03, New Orleans, ACM Press.

Stefanidis, A. and S. Nittel (2004). GeoSensor Networks, CRC Press.

Valiant, L. G. (1984). "A Theory of the Learnable." Communications of the ACM **27**(11): 1134-1142.

Vapnik, V. (1995). The Nature of Statistical Learning Theory. New York, Wiley.

Vapnik, V. (1998). Statistical Learning Theory. New York, Wiley.

Waller, L. and C. A. Gotway (2004). Applied Spatial Statistics for Public Health Data. New York, Wiley.

Ward, J. H. (1963). "Hierarchical Grouping to Optimize an Objective Function." Journal of the American Statistical Association 58: 236-244.

Wheeler, D. (2007). "A Comparison of Spatial Clustering and Cluster Detection Techniques for Childhood Leukemia Incidence in Ohio, 1996-2003." International Journal of Health Geographics 6(13).

Woods Hole Oceanographic Institution (2008). "Underwater Vehicles - Puma." http://www.whoi.edu/page.do?pid=11399

Worboys, M. (2005). "Event Orietated Approaches to Geographic Phenomena." International Journal of Geographic Information Science 19(1): 1-28.

Worboys, M. and M. Duckham (2006). "Monitoring Qualitative Spatiotemporal Change for Geosensor Networks." International Journal of Geographic Information Science 20(10): 1087-1108.

Yang, Y. and J. Xin (1999). A Re-examination of Text Categorization Methods. Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieva, Berkeley, ACM

Zeng, D., W. Chang, H. Chen (2004). A Comparative Study of Spatio-Temporal Hotspot Analysis Techniques in Security Informatics. 2004 IEEE Intelligent Transportation Systems Conference.

BIOGRAPHY OF THE AUTHOR

Jon Devine grew up in Cary, North Carolina. After graduating from Cary High School he attended the University of North Carolina at Wilmington (UNCW) where he pursued a double major in Economics and Environmental Studies with minors in French and History. He graduated from UNCW in 2000 cum laude with honors in Economics. After spending a summer abroad in Marseille, France, Jon returned to Europe for a year of study in Nice, France and a year teaching conversational English to university and high school students in Corte, Corsica. Upon his return, Jon enrolled in the Department of Resource Economics and Policy (REP) at the University of Maine. Among the reasons that he chose the University of Maine was the unique combination of strength in both economics and spatial science. His advisor in the REP program, Dr. Kathleen Bell, specializes in spatial econometrics and through exposure to her research, as well as courses taken with the Department of Spatial Information Science and Engineering (SIE), Jon became interested in obtaining a second Master's degree in spatial science. After completing a Master's from REP, he became a full-time student in SIE. This program eventually brought Jon the Fairfax, Virginia for a semester in order to continue to work with Dr. Tony Stefanidis who had transferred from Maine to George Mason University. During this time, Jon began looking for work and landed a job as an economist with Cotton Incorporated. In that current role Jon continues to apply the computing and spatial skills he acquired from SIE. He is a candidate for the Master of Science degree in Spatial Information Science and Engineering from the University of Maine in May, 2009.