Applied Artificial Intelligence: An International Journal

Publication details, including instructions for authors and subscription information:
http://www.tandfonline.com/loi/uaai20

EFFECTIVENESS OF SUPPORT VECTOR MACHINE FOR CRIME HOT-SPOTS PREDICTION

Keivan Kianmehr [a] & Reda Alhajj [a b]

[a] Department of Computer Science , University of Calgary , Calgary, Alberta, Canada

[b] Department of Computer Science , Global University , Beirut, Lebabnon

Published online: 19 May 2008.

To cite this article: Keivan Kianmehr & Reda Alhajj (2008) EFFECTIVENESS OF SUPPORT VECTOR MACHINE FOR CRIME HOT-SPOTS PREDICTION, Applied Artificial Intelligence: An International Journal, 22:5, 433-458, DOI: 10.1080/08839510802028405

To link to this article: http://dx.doi.org/10.1080/08839510802028405

Applied Artificial Intelligence, 22:433–458

ISSN: 0883-9514 print/1087-6545 online
DOI: 10.1080/08839510802028405

Taylor & Francis
Taylor & Francis Group

EFFECTIVENESS OF SUPPORT VECTOR MACHINE FOR CRIME HOT-SPOTS PREDICTION

Keivan Kianmehr[1] and Reda Alhajj[1,2]
[1]*Department of Computer Science, University of Calgary, Calgary, Alberta, Canada*
[2]*Department of Computer Science, Global University, Beirut, Lebanon*

□ *Crime hot-spot location prediction is important for public safety. The output from the prediction can provide useful information to improve the activities aimed at detecting and preventing safety and security problems. Location prediction is a special case of spatial data mining classification. For instance, in the public safety domain, it may be interesting to predict location(s) of crime hot spots. In this study, we present a support vector machine (SVM)-based approach to predict the location as an alternative to existing modeling approaches. Support vector machine forms the new generation of machine-learning techniques used to find optimal separability between classes within datasets. We compare the performance of two types of SVMs techniques: two-class SVMs and one-class SVMs. We also compared SVM with a neural network-based approach and spatial auto-regression-based approach. Experiments on two different spatial datasets demonstrate that the former approach performs slightly better and the latter one gives reasonable results. Furthermore, in this study, we provide a general framework to customize the spatial data classification task for other spatial domains that have datasets similar to the analyzed crime datasets.*

Discovering hidden information from huge databases generates the need for new techniques and tools that can intelligently and automatically transform data into useful information and knowledge. Data mining is such a kind of technique (Shapiro and Frawley 1991; Chen, Han, and Yu 1996). Data mining employs algorithms and techniques from statistics, machine-learning, artificial intelligence, databases, and data warehousing, etc. (Dunham 2002). Different studies have shown that these techniques give reasonable and sometimes outperforming results in data mining tasks. Some of the most popular tasks of data mining are association and sequence analysis, classification, clustering, and regression (Dunham 2002). Finally, data mining techniques have been successfully applied to analyze spatial data.

Address correspondence to Keivan Kianmehr, University of Calgary, Department of Computer Science, 2500 University Drive NW, Calgary, Alberta, T2N 1N4, Canada. E-mail: alhajj@cpsc.ucalgary.ca

Spatial data are the data related to objects that occupy space. A spatial database stores spatial objects represented by spatial data types and spatial relationships among such objects. Spatial data carries topological and/or distance information; it is often organized by spatial index structures and accessed by spatial access methods. These distinct features of a spatial database bring challenges and opportunities for mining knowledge from spatial data (Lu, Han, and Ooi 1993).

Spatial data mining is a subfield of data mining; it is a process that uses a variety of data analysis tools to discover spatial patterns and relationships in spatial data that may be used to make valid predictions (Koperski, Adhikary, and Han 1996; Koperski and Han 1995). This has wide applications in geographic information systems (GIS), remote sensing, image databases exploration, medical imaging, robot navigation, and other areas where spatial data are used. The main methods for spatial data analysis include spatial association rules extraction, clustering, and classification.

Data objects stored in a database are identified by their attributes. Classification finds a set of rules that determine the class of each classified object according to its attributes. Objects with similar attribute values are classified into the same class. For instance, if the unemployment in a city is high and the population of the city is high then the crime rate in that city is high. Spatial classification deals with datasets that contain spatial objects. In spatial classification, attribute values of neighboring objects may also be relevant for the membership of objects in a certain group. This neighborhood factor, which is called spatial autocorrelation, plays a significant role in spatial data analysis and poses challenges in spatial data classification as well (Shekhar, Zhang, Huang, and Vatsavai 2003). Traditional methods for data mining cannot be effectively applied to spatial data classification since they do not consider spatial autocorrelations among objects. Therefore, there is the need to integrate the existing techniques into new approaches to the spatial data classification task.

Location prediction is a special case of spatial data classification in which we are interested to predict the location. For instance, as an example from public safety, we are interested in predicting the location of crime hot spots. Such an analysis is an important part of public safety. The output of the analysis can provide useful information to improve the activities aimed at preventing and detecting safety and security problems. Therefore, the availability of a location prediction inquiry system can be a helpful tool for public safety experts.

A wide variety of research has considered the use of data mining techniques like neural network (Ozesmi and Ozesmi 1999) logistic regression (Ozesmi and Mitsch 1997), and a decision tree to extract patterns from spatial datasets to predict location. However, these previous studies have shown that traditional techniques do a poor job in predicting location task

because they do not consider the spatial relations between spatial objects. To guarantee the spatial dependencies of objects, several techniques have been proposed by scientists. Predicting Locations Using Map Similarity (PLUMS) by Chawla and Shekhar (2000) is a method for supervised spatial data classification based on using map similarity measures. A spatial autoregressive regression (SAR) technique has also been proposed by spatial statisticians (Lesage 1997).

The special goal of this research project is to analyze and explore crime datasets. However, the framework we are providing in this article can be applied to different domains. In our selected crime datasets, the location is described by Euclidian coordinates or latitude and longitude. Each location is also identified by its crime rate and several related attributes. In this study, we develop a model to automatically classify the locations as either hot-spot crime member or hot-spot crime nonmember. Our system will accept a predefined level of crime rate as input. Based on this value, the system will label a certain portion of the dataset as hot-spot members and nonmembers. If a location's crime rate is above the predefined level of crime, it will be labeled as a member of hot-spot crime class or positive sample—otherwise a nonmember of hot-spot crime or negative sample. The user specifies from the dataset a certain portion to be used as input to the system.

We are using two different approaches for choosing the certain portion of the dataset: random selection and clustering. After labeling the specified certain percentage of the dataset, the system uses the labeled portion as a training set for building a classifier by applying the support vector machines (SVM) algorithm. Since we are classifying the locations into positive and negative samples, we can approach this task as a binary classification problem. It means the system automatically classifies a location with the crime rate above the predefined level as a member of crime hot-spot class or positive, and a location with crime rate below the predefined level as a nonmember of hot-spot class or negative.

The SVM algorithm (Boser, Guyon, and Vapnik 1992; Vapnik 1998) is the technique that we chose for a binary classification task in this study. It has received great consideration because of its distinguished performance in a wide variety of application domains such as object recognition, speaker identification, face detection, handwriting recognition, and text categorization, among others (Cristianini and Taylor 2000). Generally, SVM is useful for pattern recognition in complex datasets. It usually solves classification problems by learning from examples. As it is obvious from its name, the binary classification method requires negative and positive examples to establish a statistical relationship and to build a classifier model. However, in reality, many types of datasets suffer from lack of reliable negative or nonmember of positive class examples. This is our main motivation

to extend this problem to crime classification. We compare the performance of one-class classification with two-class classification in crime location prediction. The one-class SVM approach has been recently applied in several studies, such as gene expression classification (Kowalczyk and Raskutti 2002), text categorization (Manevitz and Yousef 2001), and text summarization (Kruengkrai and Jaruskulchai 2003).

To demonstrate the effectiveness and applicability of the proposed approach, it has been compared with two other existing approaches, namely, neural network and SAR. Neural network is a machine-learning technique that performs well in classical data mining tasks. Spatial autoregression, is a well-known statistical approach in spatial data analysis. The benefit of SAR is that it implements the spatial dependencies among spatial objects in the building model process by using a contiguity matrix, which represents the neighborhood relationship by using, for instance, Euclidean distance. We use the Lesage implementation (LeSage 1999a), which applies a Bayesian approach with Gibbs sampling to maximize the likelihood.

RELATED WORK

As described in the literature, hot-spot crime analysis includes two major techniques in the research literature: spatial statistics-based and spatial data mining. Alternative approaches have also been proposed to combine different techniques to refine the crime data analysis process. In the rest of this section, we will briefly describe several well-known approaches in the crime data analysis area.

Statistical-based techniques have been widely studied by researchers and many of the well defined models built, which are based on those techniques, are used in public safety departments and other related areas. For instance, many regression methods like logistic and Poisson are broadly used. When analyzing crime hot-spots or other types of spatial data, classical statistical modeling approaches apply models in the same manner as they would do for a nonspatial context. However, spatial data exhibit several distinct features not found typically in nonspatial data that can affect the use of classical models. Brown (1982) explored the spatial data for the existence of the autocorrelation; he discovered that it might affect the accuracy of regression results. Spatial autocorrelation is one of the most important properties of spatial data that differentiates it from other kinds of data. Anselin (1998) shows that considering the special aspect of spatial data (such as autocorrelation) is very critical for the crime environment analysis and other types of spatial data. Since then, classical models have been extended to spatial statistics by scientists in a way that they consider the special properties of the data in building the model. For example, spatial

autoregressive models, which are very well-known spatial statistical models, are used to model cross-sectional spatial data samples. Anselin provides a relatively complete treatment of these models from a maximum likelihood perspective (Anselin 1988; LeSage 1999b). However, spatial statistical techniques are computationally very complex and need more runtime compared to other techniques. Furthermore, applying these methods to different domains has shown that they are not able to discover unknown patterns, which are usually expected to be discovered by the models, in huge spatial data sets (Miller and Han 2001).

Scan statistics is another popular technique; it is used in a wide range of fields including crime hot-spot analysis (Kaminski, Jefferis, and Chanhatasilpa 2000). The main goal of scan statistics is to check for event clusters and specify their locations (Kulldorff 1997). In scan statistic, we assume that the number of crime events follows the Poisson distribution over a constant time. Then, the method imposes a window, which is the shortest length among all possible windows on the map and scans the window over the map area continuously, such that it contains a different set of events. Counting the number of neighboring cases at each location inside the window, the scan statistics is then defined as the maximum number of crime cases as the center of the window moves all over the map. The spatial scan statistics can be used for the special case of data where the location of crime events are identified by using the coordinates (latitude and longitude) and each area contains one person at risk. It can also be used for aggregate crime analysis (Zeng, Chang, and Chen 2004). The main drawback of this approach is that it is difficult to customize the clustering result. Further, its efficiency depends on fixed shape of regions.

The drawbacks of statistical approaches motivated researchers to explore and discover approaches for crime data analysis based on data mining techniques. For instance, clustering is widely used to determine the hot-spots in crime analysis. However, classical data mining techniques are not suitable for analyzing spatial data since they don't consider the spatial autocorrelations and assume that each input is independent of other inputs. Furthermore, the data analyzed by classical data mining are, in general, numerical and categorical mostly generated by implicit resources, while spatial data are often explicit. These issues have turned on the research interests into spatial data mining techniques (Shekhar et al. 2003).

Several clustering approaches that might be applied to crime data analysis have been described and compared in Murray and Estivill-Castro (1998). A k-means basic approach has been proposed in Murray (2000). Risk-adjusted nearest neighbor hierarchical (RNNH) clustering is a well-known spatial data mining technique based on nearest neighbor clustering which has been developed for the crime data analysis. It combines the

hierarchical clustering capabilities with kernel density techniques. The standard NNH approach clusters data points that are close together mostly based on some baseline factors. Risk-adjusted nearest neighbor hierarchical clustering basically specifies clusters of data points that are relative to the baseline factor. It adjusts the threshold distance inversely proportional to the density measure of the baseline factor. Such density measures are computed using kernel density based on the distances between the location and some or all other data points (Zeng et al. 2004; Levine 2002). Risk-adjusted nearest neighbor hierarchical clustering has been considered as a well-performed crime data analysis approach since it supports clusters with different shapes, and also because of its computational efficiency.

Kernel density estimation is another well-known method to locate crime hot-spots on the map. This method aggregates points inside a specified search radius to create a smooth surface that represents the density of crime events across the area. The kernel density method is becoming more popular since it helps in identifying more precisely the location, spatial extent, and intensity of crime hot-spots. Along with its popularity, this method has several issues as well. It is critical in making sure that the underlying data point is accurate; otherwise it may result in merely creating a nice-looking map of wrong hot-spots. Another problem that relates to the setting of range methods is when little regard is given to the legend thresholds that help to decide when a cluster of crimes can be defined as a hot-spot (toolkits[1]).

Another crime data analysis approach has been proposed in Xue (2003), which is based on "spatial choice study" from McFadden's discrete choice theorem (McFadden 1973). In this study, they show that the analysis of spatial choice provides a methodology suited to problems related to site selection in geographical space, and is concerned with human decision making-processes. Criminal incidents, like many other human initiated events, can be described as spatial choice procedure. They are linked to decision-making processes indicating preferences that individuals (for example, criminals) have for specific sites (crime locations) in terms of certain spatial attributes. In other words, spatial choice is a process that individuals use to choose a specific site in space to meet their objectives. Criminals choose possible sites in space to commit crimes. Their choices show certain patterns over a geographic region. According to their experiments using real crime incidents, spatial choice models can reveal the preferences of unknown criminals in space and give accurate predictions of spatial patterns of future crimes. There are two main reasons for the popularity of this approach. First, the discrete choice models help the analyst to get a better way of understanding, for instance, how a household trades-off play an important role among the wide range of choice factors. Second, the approach allows the analyst to examine the choice behavior based on both

accepted and rejected incidents, and to relate spatial behavior to geographical characteristics as well as the preferences and attitudes of individuals (Guo 2004). The issue with this approach is when they apply the discrete choice models to analyze a spatial choice context; they do the same technique as they do in case of a nonspatial context with little modification (Pellegrini and Fotheringham 2002). However, there are several significant features in spatial choice contexts which do not exist in nonspatial contexts. Ignoring these features can result in problems in the use of standard choice models.

As our contribution to the crime data analysis research, we propose an integrated approach in which we combine clustering and classification techniques in order to build a prediction model to predict the new crime hot-spots based on previously known hot-spot incidents.

SUPPORT VECTOR MACHINE

Support vector machine perform classification, i.e., they identify data samples of members or nonmembers of a given class. Basically, SVMs are designed for two-class problems where there exists both positive and negative samples. In this case, SVMs attempt to construct a separating hyperplane between training data that have been labeled as either positive or negative data. Using this separating hyperplane, an unknown sample can be identified as a positive member or a negative member based on whether it is on the positive member side or negative member side of the hyperplane. However, in reality, it is difficult to find negative samples most of the time. There are cases where there is only one-class dataset. For instance, we may consider the handwritten samples of a single number in the area of handwritten number recognition. The classifier attempts to identify whether each of the samples is a member of the target number class or not. Because of the many existing one-class problems, which require the separation of a target class from the rest of nonmember features, the idea of SVMs was extended by scientists to one-class classification problems (Scholkopf, Platt, Taylor, Smola, and Williamson 1999; Tax and Duin 2001).

Another important issue that SVMs take into account is that for many real-world applications it is impossible to construct a separating hyperplane, as demonstrated by the input space in Figure 1. Support vector machine address this problem by mapping data from its original k-dimensional space, called input space, into a higher-dimensional feature space. It is in this feature space for which the separating hyperplane is constructed. The mechanism responsible for defining the mapping is called the kernel function Φ (Vapnik 1998). A simple example of this process where nonseparable data in two-dimensional input space is mapped into three-dimensional feature space is shown in Figure 1.

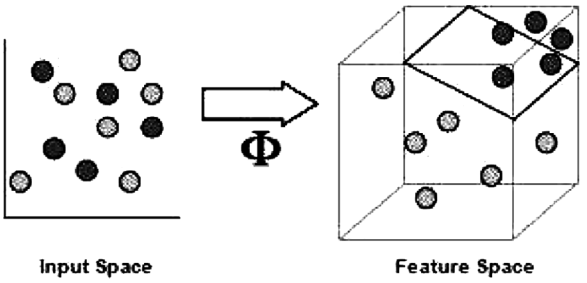

FIGURE 1 Support vector machine: mapping nonseparable data from input space to higher dimensional feature space where a separating hyperplane can be constructed.

Two-Class SVM

When used for classification, SVMs separate a given known set of $+1, -1$ labeled training data via a hyperplane that is maximally distant from the positive samples and negative samples (optimal separating hyperplane, Figure 2); plot the test data at the high-dimensional space and distinguish whether it belongs to positive or negative according to the optimal separating hyperplane.

As mentioned before, for most real-world problems that seem not to be linearly separable, SVMs can work in combination with the technique of "kernels," which automatically realizes a nonlinear mapping onto a feature space. The optimal separating hyperplane found by SVM is the feature space that corresponds to a nonlinear decision boundary in the input space (Vapnik 1998).

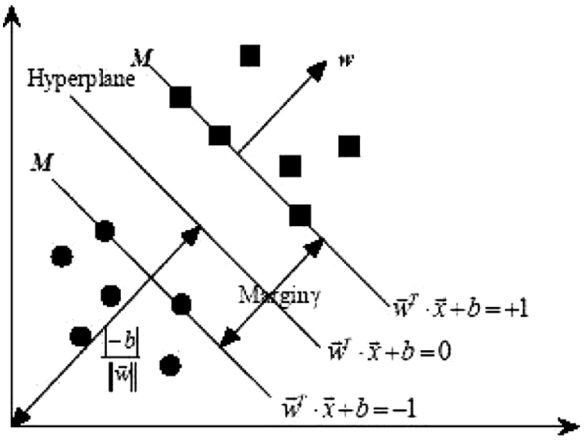

FIGURE 2 Definition of hyperplane and margin, circular dots and square dots represent samples of classes -1 and $+1$, respectively.

Linear Classification

Let the training data of two separable classes with n samples be represented by $(\vec{x}_1, \vec{y}_1)$, $(\vec{x}_2, \vec{y}_2), \ldots, (\vec{x}_n, \vec{y}_n)$, $i = 1, 2, \ldots, n$ where $\vec{x} \in R^N$ is an N dimensional space and $y_i \in \{-1, +1\}$ is the class index. Given a weight vector w and bias b (Figure 2), the two classes can be separated by two margins parallel to the hyperplane:

$$\vec{w}^T \cdot \vec{x}_i + b \geq 1, \quad y_i = +1 \tag{1}$$

$$\vec{w}^T \cdot \vec{x}_i + b \leq -1, \quad y_i = -1 \tag{2}$$

where $\vec{w} = (w_1, w_2, \ldots, w_n)^T$ is a vector of n elements. Inequalities (1) and (2) can be combined into the single inequality:

$$y_i(\vec{w}^T \cdot \vec{x}_i + b) \geq 1, \quad i = 1, 2, \ldots, n. \tag{3}$$

There exist a number of hyperplanes for each group of training data. The classification objective of SVM is to determine an optimal weight w_0 and an optimal bias b_0 such that the selected hyperplane separates the training data with maximum margin. The hyperplane determined by w_0 and b_0 is called optimal separating hyperplane. The equation of any given hyperplane can be written as:

$$\vec{w}^T \cdot \vec{x}_i + b = 0 \tag{4}$$

and the distance between the two corresponding margins is

$$\gamma(\vec{w}, b) = \min_{\{\vec{x}|y\pm 1\}} \frac{\vec{x}^T \cdot \vec{w}}{\|\vec{w}\|} - \max_{\{\vec{x}|y\pm 1\}} \frac{\vec{x}^T \cdot \vec{w}}{\|\vec{w}\|}. \tag{5}$$

The optimal separating hyperplane can be found by maximizing the above distance, or by minimizing the norm $\|w\|$ under inequality constraints (3), and

$$\gamma_{\max} = \gamma(\vec{w}_0, b_0) \frac{2}{\|\vec{w}_0\|}. \tag{6}$$

The saddle point of the following Lagrangian gives solutions to the above optimization problem:

$$L(\vec{w}, b, a) = \frac{1}{2} \vec{w}^T \cdot \vec{w} - \sum_{i=1}^{n} \alpha_i [y_i(\vec{w}^T \cdot \vec{x}_i + b) - 1] \tag{7}$$

where all $\alpha_i \geq 0$ are Lagrange multipliers. The solution of this optimization quadratic programming (QP) problem requires the gradient of $L(\vec{w}, b, a)$ with respect to w and b vanishes, which gives the following conditions

by the calculation of $\frac{\partial L}{\partial \vec{w}}\big|_{\vec{w}=\vec{w}_0} = 0$ and $\frac{\partial L}{\partial b}\big|_{b=b_0} = 0$:

$$\vec{w}_0 = \sum_{i=1} \alpha_i y_i \vec{x}_i \tag{8}$$

$$\sum_{i=1}^{n} \alpha_i y_i = 0. \tag{9}$$

By substituting Equation (8) and (9) into Equation (7), the QP problem becomes maximization of the following expression:

$$L(\alpha) = \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j (\vec{x}_i^T \cdot \vec{x}_j). \tag{10}$$

Under the constraints $\sum_{i=1}^{n} \alpha_i y_i = 0$ and $\alpha_i \geq 0, \quad i = 1, 2, \ldots, n$.

The points situated at the two optimal margins have nonzero coefficients α_i among the solutions to Equation (10), and are called support vectors (SV). The bias b_0 can be calculated as follows:

$$b_0 = \frac{1}{2} \left(\min_{\{\vec{x}_i | y_i \pm 1\}} \vec{w}_0^T \cdot \vec{x}_i - \max_{\{\vec{x}_i | y_i \pm 1\}} \vec{w}_0^T \cdot \vec{x}_i \right). \tag{11}$$

After determining SV and the bias, the decision function that separates the two classes can be written as

$$f(\vec{x}) = \text{sign} \left[\sum_{i=1}^{n} \alpha_i y_i \vec{x}_i^T \cdot \vec{x} + b_0 \right] = \text{sign} \left[\sum_{\text{SV}} \alpha_i y_i \vec{x}_i^T \cdot \vec{x} + b_0 \right]. \tag{12}$$

Nonlinear Classification

As real-world problems are usually nonlinear, the following approach has been introduced into SVM to deal with these problems. The original training data x in the input space X is projected into a high-dimensional feature space F via a Mercer kernel operator K, and the optimal separating hyperplane is constructed in this feature space. In mathematical terms, the set of classifiers is transformed into the form

$$f(\vec{x}) = \text{sign} \left[\sum_{i \in \{\text{SV}\}} \alpha_i y_i K(\vec{x}_i, \vec{x}) + b_0 \right] \tag{13}$$

where K is a symmetric, positive-definite function, which satisfies Mercer's conditions:

$$K(\vec{x},\vec{y}) = \sum_{m1}^{\infty} \alpha_m \phi(\vec{x})^T \cdot \phi(\vec{y}), \quad \alpha_m \geq 0,$$
$$\iint K(\vec{x},\vec{y}) g(\vec{x}) g(\vec{y}) d\vec{x} d\vec{y} > 0, \quad \int g^2(\vec{x}) d\vec{x} < \infty. \tag{14}$$

The kernel represents a legitimate inner product in the input space:

$$K(\vec{x},\vec{y}) = \phi(\vec{x})^T \cdot \phi(\vec{y}). \tag{15}$$

In the F-space, the dual Lagrangian, given in Equation (10) is

$$L(\alpha) = \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j K(\vec{x}_i^T \cdot \vec{x}_j) - \lambda \sum_{i=1}^{n} \alpha_i y_i \tag{16}$$

subject to

$$\sum_{i=1}^{n} \alpha_i y_i = 0 \text{ and } \alpha_i \geq 0, \quad i = 1, 2, \ldots, n$$

and the decision function is

$$f(\vec{x}) = \text{sign}\left[\sum_{i \in \{\text{SV}\}} \alpha_i y_i K(\vec{x}_i, \vec{x}) + b_0 \right] \tag{17}$$

where

$$b_0 = -\frac{1}{2} \min_{\{\vec{x}_i | y_i = -1\}} \left(\sum_{j \in \{\text{SV}\}} \alpha_j y_j K(\vec{x}_i, \vec{x}_j) \right)$$
$$-\frac{1}{2} \max_{\{\vec{x}_i | y_i = -1\}} \left(\sum_{j \in \{\text{SV}\}} \alpha_j y_j K(\vec{x}_i, \vec{x}_j) \right). \tag{18}$$

One-Class SVM

There are two different approaches to one-class SVMs. The goal of the one-class SVM approach of Tax and Duin (2001), which is called the support vector data description (SVDD), is to find a hypersphere that covers

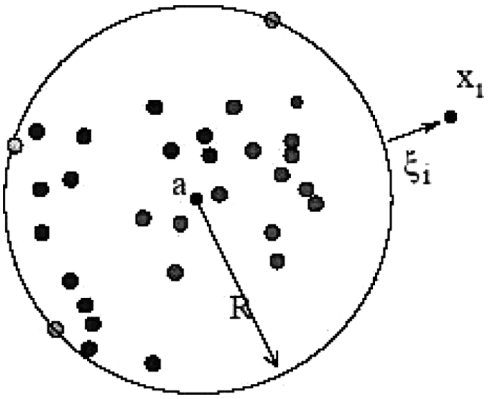

FIGURE 3 The hypersphere containing the target data, described by center a and radius R. Three objects are on the boundary—the support vectors. One object x_i, is outside and has $\xi_i > 0$.

as many training data points as possible, while keeping the radius of the hypersphere as small as possible. In other words, given l training data points, $x_i, (i = 1, 2, \ldots, l)$, find a hypersphere, which is as small as possible, to contain the training points in multi-dimensional space. Also, small portions of outliers are allowed to exist using a slack variable (ξ_i) as shown in Figure 3 (Gou, Kelley, and Graham 2005):

$$\text{Min}(R^2) + \frac{1}{vl}\sum_i \xi_i \tag{19}$$

subject to

$$(x_i - c)^{\mathrm{T}}(x_i - c) \leq R^2 + \xi_i, \ \ \xi_i \geq 0 \quad \text{for all } i \in [l] \tag{20}$$

where c and R are the center and radius of the sphere, respectively, T is the transpose, and $v \in (0, 1]$ is the trade-off between volume of the sphere and the number of training data points rejected. When v is large, the volume of the sphere is small; so more training points will be rejected than when v is small, where more training points will be contained within the sphere.

This optimization problem can be solved by the Lagrangian:

$$L(R, \xi, c, a_i, \beta_i) = R^2 + \frac{1}{vl}\sum_{i1}^{l} \zeta_i - \sum_{i=1}^{l} a_i\{R^2 + \zeta_i - (x_i^2 - 2cx_i + c^2)\} - \sum_{i=1}^{l} \beta_i \xi_i \tag{21}$$

where $a_i \geq 0$ and $\beta_i \geq 0$; setting the partial derivative of L with respect to R, a_i, and c equal to 0, we get

$$\sum_{i=1}^{l} a_i = 1 \tag{22}$$

$$0 \leq a_i \leq \frac{1}{vl} \tag{23}$$

$$c = \sum_{i=1}^{l} a_i x_i. \tag{24}$$

Substituting Equations (22)–(24) into Equation (21), we have the dual problem

$$\min_a \sum\nolimits_{i,j} a_i a_j (x_i \cdot x_j) - \sum\nolimits_i a_i (x_i . x_j) \tag{25}$$

subject to

$$0 \leq a_i \leq \frac{1}{vl}, \quad \sum_{i=1}^{l} a_i 1.$$

By calculating the distance between a test point (x) and the center C of the hypersphere, it is possible to determine whether (x) is inside the sphere or not. By using the following inequality, the position of the test point can be identified:

$$(x \cdot x) - 2 \sum_i a_i (x \cdot x_i) + \sum a_i a_j (x_i \cdot x_j) \leq R^2. \tag{26}$$

The assumption of the spherical distribution of the data, which has been considered thus far, is not always true. In other words, in reality the data are not always spherically distributed. Different types of kernel functions $K(x_i, x_j)$ can be used in order to build more flexible model that can deal with nonlinear data. Kernel functions and their usage in building powerful classifiers are described in more details in the next section.

The second approach in one-class SVMs was proposed by Scholkopf et al. (1999). Their approach is to construct a hyperplane maximally distant from the origin with all data lying on the opposite side from the origin as

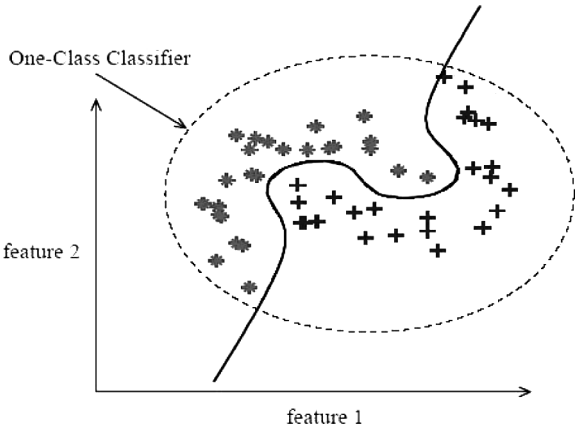

FIGURE 4 The solid line is the conventional classifier that distinguishes between positives and negatives, while the dashed line describes the dataset.

shown in Figure 4. In other words, given training $x_1, \ldots, \mathrm{x_l} \in \mathrm{R^N}$, where x_i is a feature vector, it is required to estimate a function that takes the value $+1$ in a small region capturing most of the data points, and -1 elsewhere (Manevitz and Yousef 2001). Formally, the function is written as

$$\begin{aligned} f(x) = +1 \quad &\text{if } x \in S, \\ -1 \quad &\text{if } x \in \overline{S}, \end{aligned} \tag{27}$$

where S and $\overline{S}$ are simple subsets of the input space and its complement, respectively. Let $\Phi\colon \Re^N \to F$ be a nonlinear mapping that maps the training data from $\Re^N$ to a feature space F. To separate the data set from the origin, solve the following primal optimization problem (Kruengkrai and Jaruskulchai 2003):

Minimize:

$$V(w, \xi, \rho) = \frac{1}{2}||w||^2 + \frac{1}{vl}\sum_{i=1}^{l} \xi_i - \rho \tag{28}$$

subject to

$$(w \cdot \Phi(\mathrm{x}_i)) \geq \rho - \xi_i, \xi_i \geq 0, \tag{29}$$

where $v \in (0, 1)$ is a parameter for controlling the trade-off between the number of outliers and model complexity, and ρ is the margin. Using the following decision function, a label can be assigned to a new given data

point (x) for classification task:

$$f(x) = sgn(w \cdot \Phi(x_i) - \rho). \tag{30}$$

Again, as in the former approach, introducing Lagrange multipliers α_i and using the Kuhn-Tucker condition, the derivatives with respect to the primal variables are set to zero to get

$$w = \sum_i \alpha_i \Phi(x_i) \tag{31}$$

where only a subset of points x_i that are closest to the hyperplane have non-zero values α_i. These points are called support vectors. Instead of solving the primal optimization problem directly, the following dual program can be considered:
Maximize:

$$w(\alpha) = \frac{1}{2} \sum\nolimits_{i,j} \alpha_i \alpha_j K(x_i, x_j) \tag{32}$$

subject to

$$0 \leq \alpha_i \leq \frac{1}{vl}, \quad \sum_i \alpha_i = 1. \tag{33}$$

From Equations (30) and (31), the decision function can be written as

$$f(x) = sgn\left(\sum_i \alpha_i K(x_i, x_j) - \rho \right). \tag{34}$$

$K(x_i, x_j) = (\Phi(x_i), \Phi(x_j))$ are the kernels (the dot products between mapped pairs of input points). As mentioned before, kernel functions allow more general decision functions when the data points are not linearly separable.

Kernel Functions

The idea of the kernel function is to enable operations to be performed in the feature space rather than the potentially high-dimensional input space. Hence, the inner product does not need to be evaluated in the feature space. This provides a way of addressing the curse of dimensionality. However, the computation is still critically dependent upon the number of training patterns, and to provide a good data distribution for a high-dimensional problem will generally require a large training set (Vapnik 1999).

The kernel theory is based upon reproducing kernel Hilbert spaces (RKHS) (Aronszajn 1950; Wahba 1990; Girosi 1997). An inner product in the feature space has an equivalent kernel in the input space, $K(x, x') = (\phi(x), \phi(x'))$, provided certain conditions hold. If K is a symmetric, positive-definite function, which satisfies Mercer's conditions:

$$K(x, x') = \sum_{m}^{\infty} a_m \phi_m(x) \phi_m, \quad a_m \geq 0 \tag{35}$$

$$\int\int K(x, x') g(x) g(x') dx dx' > 0, \quad g \in L_2, \tag{36}$$

then the kernel represents a legitimate inner product in the feature space. There are many kinds of valid functions that satisfy Mercer's conditions. Since two types of these functions are often used for classification problems—polynomial and Gaussian kernels—here we limit ourselves to these two kernel functions.

A polynomial mapping is a popular method for nonlinear modeling:

$$K(x, x') = \langle x, x' \rangle^d \tag{37}$$

$$K(x, x') = (\langle x, x' \rangle + 1)^d. \tag{38}$$

The second kernel is usually preferable as it avoids problems with the Hessian becoming zero.

Radial basis functions have received significant attention, most commonly with a Gaussian of the form

$$K(x, x') = \exp\left(-\frac{||x - x'||^2}{2\sigma^2} \right). \tag{39}$$

Classical techniques utilizing radial basis functions employ some method of determining a subset of centers. Typically, a method of clustering is first employed to select a subset of centers. An attractive feature of the SVM is that this selection is implicit, with each support vector contributing one local Gaussian function, centered at that data point.

EXPERIMENT METHODS

In order to select a certain representative portion of the crime datasets to be used as the training set by the system, we experiment with the

following approaches: For a given percentage of the data and a predefined level of crime rate, we select a subset of the crime dataset to label; and then based on the predefined level of crime rate, we specify a class label to each data point in the selected set. The data points that have the crime rate above the predefined rate are positive or members of hot-spot class, and data points with crime rates below the predefined rate are negative or nonmembers of hot-spot class. Then this labeled data set will be used as the training set in SVM classification. To select a given percentage of the data to be labeled, we use the k-means clustering algorithm. Then, we compare the result when the same percentage of the data is selected randomly. Finally, we compare the results of the SVM-based approach with two other existing approaches, namely, neural networks and SAR.

K-Means Clustering Algorithm

K-means clustering is a partitioning method that partitions the data points of the dataset into K clusters. K-means treats each data point in the dataset as an object having a location in space. It finds a partition in which objects within each cluster are as close to each other as possible, and as far from objects in other clusters as possible. Each cluster is defined by its data members and its centeroid. The centroid for each cluster is the point to which the sum of distances from all objects in that cluster is minimized.

In our experiments, the idea of using k-means cluster is to consider the data points within a certain distance from the k clusters centeroids as representative data points of the clusters. The value of k should be chosen in a way by the system such that a certain total percentage of the data set fall into a desired distance from a closest cluster center. By gathering a small set of data points from each cluster based on the above procedure, then we will have a portion of the whole data set ready to be labeled as the training set. Points having their crime rate above the predefined level of crime are labeled as members of hot-spot class and points having a crime rate less than the predefined level are labeled as nonmembers of hotspot class.

In order to compare the clustering approach and random approach for choosing a portion of the data set as the training set, and to compare the performance of one-class SVM against two-class SVM, we devise the following experimental plans:

1. Random Selection for Labeling + One-Class SVM: we randomly select a given percentage of the data points as a small representative portion of the crime dataset to be labeled. Then, the one-class SVM algorithm will use the output labeled set as the training set to build the classifier.

2. Random Selection for Labeling + Two-Class SVM: again we label a randomly selected portion of the dataset as the training set, and we then use the two-class SVM algorithm for classification.
3. Clustering Selection for Labeling + One-Class SVM: the k-means clustering algorithm is used to select a given percentage of the data points to be labeled. Then the one-class SVM algorithm will use the output labeled set as the training set to build the classifier.
4. Clustering Selection for Labeling + Two-Class SVM: again we label a given percentage of the dataset chosen by a k-means clustering algorithm, and then we use the two-class SVM algorithm for classification.
5. Complete Data Set + One-Class SVM: after choosing a certain percentage of the dataset for labeling by random selection or clustering technique, we label the rest of the dataset as negative samples and add them to the training set. Then, we pass the complete labeled dataset to one-class SVM as the training set.
6. Complete Data Set + Two-Class SVM: after choosing a certain percentage of the dataset for labeling by random selection or clustering technique, we label the rest of the dataset as negative samples and add them to the training set. Then we use the two-class SVM algorithm for classification.

THE DATASETS

To test the different cases enumerated within our model, we downloaded two published crime datasets from the internet. Each datasets consist of the crime rate and related variables for each data point. The location of each data point is described by Euclidian coordinates or latitude and longitude in the dataset. The datasets were downloaded from http://www.terraseer.com/. We will provide a brief description for each dataset. For further information, please refer to http://www.terraseer.com/.

The first dataset is a small crime dataset (Anselin 1988) that records crime rate and 20 related variables in 49 neighborhoods in Columbus, Ohio (see Figures 5(a) and 5(b)). The problem is to distinguish between members and nonmembers of crime hot-spot class. Hot-spot crime locations are those places that have the crime rate above the predefined level of crime.

The second dataset (Messner et al. 1999) records 78 counties surrounding St. Louis, Missour, (see Figures 5(c) and 5(d)) homicides rate and related variables. Again, the problem here is to distinguish between members and nonmembers of homicide hot-spot class. Hot-spot homicide locations are those places that have the homicide rate above the predefined level of homicide.

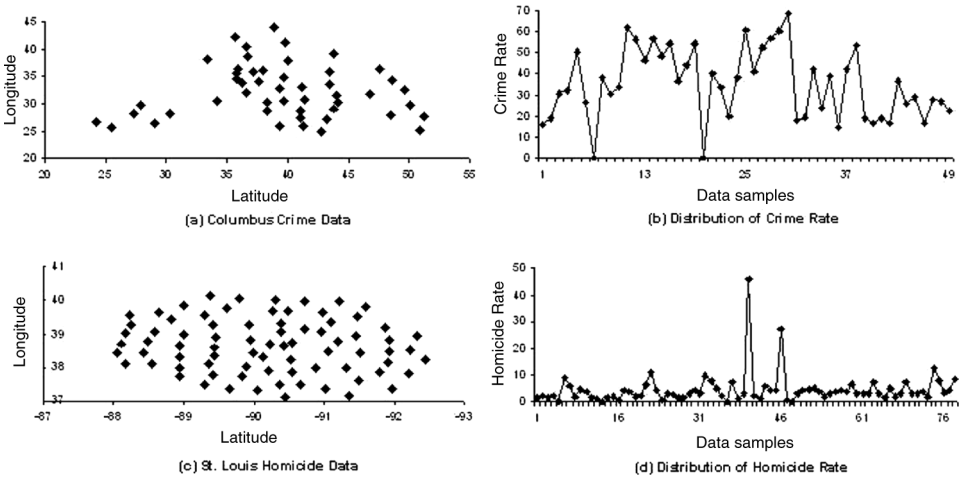

FIGURE 5 (a) Crime data and (b) its rate distribution; (c) Homicide data and (d) its rate distribution.

MODEL EVALUATION

Basically, n-fold cross validation is a method in which the data is randomly divided into n disjoint groups (Edelstein 1999). For example, suppose the data is divided into 10 groups. The first group is set aside for testing and the other nine are put together for model building. The model built on the 90% group is then used to predict the group that was set aside. This process is repeated a total of 10 times as each group in turn is set aside. Finally, a model is built using all the data. The mean of the 10 independent error rate predictions is used as the error rate for this final model. In our study, a five-fold cross validation method has been used to estimate the accuracy of the classification model.

EXPERIMENTS AND THE RESULTS

For our experiments, we used an Intel P4 2.4GHZ CPU personal computer with 1GB memory. The experiments were carried out by using a MATLAB interface of A Library for Support Vector Machines (LIBSVM) (Chang and Lin 2001) in Matlab 7. LIBSVM is a library for SVM classification and regression. The predefined level of crime depends on the knowledge of domain experts and is usually specified by crime experts. In this experiment, we assume that the data points of our datasets follow a normal distribution (Gaussian distribution) so that the optimal average value of the crime C is halfway between $C_{\min}$ and $C_{\max}$ as shown in Figure 6.

According to the definition of the Gaussian distribution, C_{average} and $C_{\max}$ are the mean (average) and the mean incremented by standard

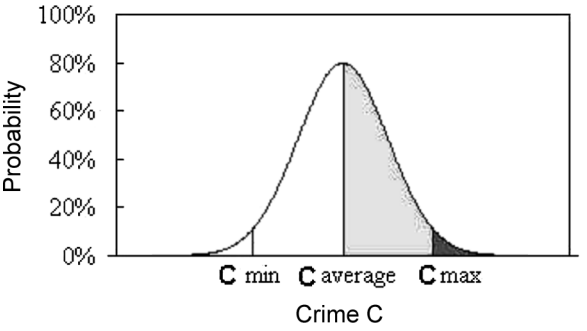

FIGURE 6 Gaussian distribution of crime rate C. The range for crime rate is between $C_{\min}$ and $C_{\max}$.

deviation (variability), respectively. In our first set of experiments, we considered C_{average} as the predefined level of crime rate, i.e., if the crime rate of a sample location is above C_{average}, then that location is a member of hot-spot class; otherwise, it would be a nonmember of the hot-spot class. In the second set of experiments, $C_{\max}$ was considered as the predefined value for hot-spots crime rate.

We selected 20% of the datasets to be labeled and used as training data in the SVM algorithm. In our experiments, we also take into account the case in which we assign the remaining 80% unlabeled data to negative class or hot-spot nonmember class, and give the complete labeled data set to the SVM to be trained. Then, we compare the result with the case where we ignore the remaining 80% unlabeled data. In our first approach to data selection, which is random data selection, we selected 20% of the dataset randomly. In order to get a more consistent result, we performed this experiment 20 times.

In the clustering data selection approach, we apply k-means clustering algorithm to the datasets first. This second approach of data selection chooses the data more wisely than the random selection. After labeling the data, we used the result set as input to the SVM algorithm. In both one-class and two-class SVMs, we applied different kernel functions to

TABLE 1 20% of the Dataset Is Selected Randomly and Labelled

		One-Class SVM			Two-Class SVM				
Data Set	C	Linear	Poly.	Gauss.	Linear	Poly.	Gauss.	NN	SAR
Columbus	35.13	63.00	60.50	68.50	62.00	63.50	60.50	64.25	72.00
	51.86	58.50	57.50	62.50	71.50	72.50	73.00	76.00	79.00
St. Louis	4.57	53.13	53.44	49.69	76.25	75.94	76.56	57.50	64.38
	10.58	50.63	50.94	48.44	95.63	94.69	95.94	87.50	88.13

TABLE 2 20% of the Dataset Is Selected Randomly and Labelled, and the Remaining 80% Is labelled as Negative Sample and Added to the Training Set

		One-Class SVM			Two-Class SVM				
Data Set	C	Linear	Poly.	Gauss.	Linear	Poly.	Gauss.	NN	SAR
Columbus	35.13	52.45	51.43	57.04	90.61	91.33	90.31	83.20	90.83
	51.86	51.22	51.53	55.10	94.184	94.39	94.49	92.60	95.42
St. Louis	4.57	49.94	49.62	38.65	94.94	95.26	95.13	82.05	82.56
	10.58	50.39	50.06	38.46	98.65	99.04	98.78	94.87	94.49

determine their influence on the classification accuracy. The results for Four different experiments are shown in Tables 1–4.

In each experiment, we first evaluate the effect of using different kernel function for one-class and two-class SVM technique. Here, we have chosen linear, polynomial, and Gaussian kernel functions. We applied the default values of LIBSVM for polynomial and Gaussian. We also changed the parameter in a range to see whether the result will be improved or not. We also performed each step 20 times and presented the average of 20 runs as the final result. In order to apply neural network technique, a three-layer model of neural network was used as the main classifier. For the input layer and the hidden layer, Tan-Sigmoid was chosen as the transfer function, which redistributes the input from the previous layer to the next layer at a range of $[-1, 1]$. Log-Sigmoid was also used as the transfer function on the final output layer to determine the result: positive/negative. Conjugate gradient back-propagation (CGB) with Powell-Beale restarts is used as the training algorithm for the network because of its decent performance and short running time. To apply an SAR approach to our problem, we needed to customize it for the comparison process. First, the dependent variable (label) had to be transformed to binary values based on our predefined crime rate. Second, we had to transform the prediction values to binary classes because the output of SAR is an estimated conditional probability.

All the results from different techniques are expressed as the number of correctly classified samples in percentage on test data. The values of C

TABLE 3 20% of the Dataset Is Selected by Clustering and Labelled

		One-Class SVM			Two-Class SVM				
Data Set	C	Linear	Poly.	Gauss.	Linear	Poly.	Gauss.	NN	SAR
Columbus	35.13	63.89	61.11	69.44	63.33	64.06	63.33	52.43	75.00
	51.86	78.33	76.11	65.55	73.33	71.67	71.67	56.98	66.00
St. Louis	4.57	53.44	54.38	50.94	77.50	78.44	77.81	67.50	68.13
	10.58	51.25	51.56	24.38	95.94	95.31	96.87	87.50	88.75

TABLE 4 20% of the Dataset Is Selected by Clustering and Labelled and the Rest 80% Is Labeled as Negative Sample and Added to the Training Set

		One-Class SVM			Two-Class SVM				
Data Set	C	Linear	Poly.	Gauss.	Linear	Poly.	Gauss.	NN	SAR
Columbus	35.13	53.47	53.06	57.35	91.84	92.14	91.74	84.88	86.49
	51.86	53.37	53.37	57.76	95.10	94.69	95.00	90.40	92.50
St. Louis	4.57	51.47	51.09	39.23	95.26	95.30	95.32	85.13	79.87
	10.58	50.51	50.64	39.31	99.17	99.10	99.17	97.44	95.77

are set to the mean (μ) and the mean plus the standard deviation ($\mu + s$) calculated for each experimented dataset.

In Table 1, the result is displayed when 20% of the dataset is selected randomly and labelled. Then this portion of data is passed to SVM as the training set. We run the SVM classifier with different kernel functions on the data set. For one-class SVM, the Gaussian kernel performs better while in two-class SVM there is no specific regulation among different kernel functions. However, polynomial and Gaussian kernels perform better than the linear one. Between one-class SVM and two-class SVM, two-class SVM performs slightly better; however when we set the C value to the mean value of the crime rate, in the Columbus dataset, SVMs with linear and Gaussian kernels perform better than two-class SVMs. Based on our knowledge from the Columbus dataset, when we set the predefined crime rate to the mean value, the number of positive samples will be more than the case that we set the predefined value C to the mean plus standard deviation. Therefore, we can say that one-class SVM performs better when we have a small training set with more positive samples.

In Table 2, the result is displayed when 20% of the dataset is selected randomly and labelled, and the rest is labelled as negative samples and added to the training set. The difference between this experiment and the one reported in Table 1 is that we have increased the size of the training set by adding more negative samples. As can be seen from Table 2, the performance of the two-class SVM will be outstandingly increased; and as in the previous experiment, polynomial and Gaussian kernels perform better than the linear function. However, one-class SVM results in a noticeable decrease in classification performance as we have a larger training set with more negative samples.

In Table 3, the result is displayed when 20% of the dataset is selected by clustering and labelled. Then this portion of data is passed to SVM as the training set and we run the SVM classifier with different kernel functions on the data set. Compared to Table 1 (the first experiment), both the one-class SVM and the two-class SVM perform much better when we use clustering instead of random selection. As a matter of fact, the clustering

technique chooses the data more wisely than random selection. The size of k is chosen by the system to be able to select 20% of the original data set from the different clusters. It means selecting 20% of the dataset within a certain distance from the closest center of each cluster and then putting them all together and labelling them as the training set. As with the first experiment, the one-class SVM with linear and Gaussian kernels performs better than the two-class SVM on the Columbus dataset when C is set to the mean value of the crime rate, i.e., for a small training set with more positive samples. However, the two-class SVM performs much better when the number of negative samples is more than the positive ones, or at the same level.

Based on the results from the previous experiments, we can predict the expected result from the last experiment where 20% of the dataset is selected by clustering and labelled and the remaining 80% is labeled as a negative sample and added to the training set. The result is displayed in Table 4. In fact, the best result in this experiment is obtained by using two-class SVMs and the k-means clustering technique for data selection and by increasing the training set size by adding more negative samples. One-class SVM performs better compared to Table 2, since we are using clustering instead of random selection. However, it does a poor job compared to Table 3, where we have a smaller dataset with more positive sample than in Table 4.

According to our experiments, SAR (which is a statistical model) and neural network (which is a learning model) perform well. However, using two-class SVMs has two significant benefits. First, it is easy to use in a way such that it doesn't rely on extensive tuning of parameters by users, and it requires minimum tuning (Cristianini and Scholkopf 2002). This feature makes it easier to use than other statistical and learning models, which need more user involvement in the model tuning process. Second, having a very strong mathematical and theoretical base, SVMs are computationally efficient and outstandingly well-performing on classification problems (Cristianini and Scholkopf 2002).

Compared to SAR and neural network models, one-class SVMs have also two main advantages. First, one-class SVMs construct a hyperplane that is maximally distant from the origin with all data lying on the opposite side from the origin. This helps to find an optimal hypersphere, which contains all or most of the training points that belong to the positive samples. Furthermore, using different type of kernel functions gives an ability to one-class SVMs so that it represents various data distribution shapes in feature space (e.g., sphere shapes or very irregular shapes (Tax and Duin 2002)). Second, one-class SVMs make no assumption on the probability density of the data (Tax and Duin 2002). This is a useful benefit when the data do not follow any probability distribution (such as normal distribution), or insufficient data are available to test the distribution (Gou et al. 2005).

Although we generally obtain better performance when we apply two-class SVMs, one-class SVMs have several advantages over two-class SVMS. First, one-class SVMs perform well when the size of the training set is not large. Second, they are suitable for problems that don't have many negative samples, since they perform better for more positive samples.

To sum up, we can figure out from the results reported in the above tables that although one-class SVMs give reasonable results, two-class SVMs perform much better. Also, labeling the remainder (80%) of the dataset as negative samples and adding them to the training set improves the performance of two-class SVMs. However, one-class SVMs perform better without adding the rest (80%) of the data set to the training set. In both one-class and two-class SVMs, we get better results when we apply a k-means clustering algorithm for selecting 20% of the data compared to the random data selection approach. Although SAR sometimes gives better results compared to two-class SVMs, two-class SVMs still perform better in most of the experiments. Also, SAR is computationally very complex and needs more runtime compared to SVM, which is a very fast algorithm especially when applied to complex dataset.

SUMMARY AND CONCLUSIONS

In this study, we provided an inquiry system that can be used as a general framework in different domains and by the domain experts in order to customize the spatial data classification task. In our framework, we focused on some special types of spatial datasets like the ones we used for our experiments. As a case study in the area of public safety, we concentrated on the performance of one-class and two-class SVMs for predicting the hot-spot crime location when a predefined level of crime rate and a percentage for selecting a portion of that are given. We applied two different approaches for data selection. First we chose a certain portion of the data randomly and in the second approach, we applied a k-means clustering algorithm in order to make a more wise selection. Then, we labeled the selected portion of the data set as members and nonmembers of the crime hot-spot class based on the predefined level of crime rate. We also studied the case when the rest of the dataset are labeled as nonmember samples and added to the training set. Our experiments demonstrate that one-class SVMs give reasonable results when we choose appropriate parameters for the algorithm, especially when the size of the training set is small with more positive samples. However, two-class SVMs perform better. Based on our different results reported in Tables 1–4, we can conclude that two-class SVMs form an appropriate approach to hot-spot crime prediction, while a k-means clustering algorithm is used for data selection and by using the rest of the dataset as nonmember samples.

REFERENCES

Anselin, L. 1988. *Spatial Econometrics: Methods and Models*. Dordrecht: Kluwer Academic.

Aronszajn, N. 1950. *Trans. Am. Math. Soc.* 68:337.

Boser, B. E., I. Guyon, and V. Vapnik. 1992. A training algorithm for optimum margin classifiers. In: *Proc. of the Annual Workshop on Computational Learning Theory*, Pittsburgh, PA, 144–152.

Brown, M. 1982. *Economic Geography* 58:247.

Chang, C. and C. Lin. 2001. *LIBSVM: A Library for Support Vector Machines*. http://www.csie.ntu.edu.tw/~cjlin/libsvm.

Chawla, S. and W. W. Shekhar. 2000. In: *Predicting Locations Using Maps Similarity (PLUMS): A Framework for Spatial Data Mining. Proc. of ACM International Conference on Knowledge Discovery and Data Mining*, Boston, MA, 14–24.

Chen, M., J. Han, and P. Yu. 1996. *IEEE Transactions on Knowledge and Data Engineering* 8(6):866.

Cristianini, N. and B. Scholkopf. 2002. *AI Mag.* 23:31.

Cristianini, N. and J. Shawe-Taylor. 2000. *An Introduction to Support Vector Machines and Other Kernel-Based Learning Methods*. Cambridge: Cambridge University Press.

Dunham, M. 2002. *Data Mining: Introductory and Advanced Topics*. Upper Saddle River, NJ: Prentice Hall.

Edelstein, H. 1999. *Introduction to Data Mining and Knowledge Discovery*, 3rd ed. Potomac, MD: Two Crows Corp.

Girosi, F. 1997. *An Equivalence between Sparse Approximation and Support Vector Machines*. Technical Report. UMI Order Number: AIM-1606, Massachusetts Institute of Technology.

Gou, Q., M. Kelley, and C. Graham. 2005. *Ecological Modeling* 182:75.

Guo, J. 2004. *Addressing Spatial Complexities in Residential Location Choice Models*, PhD dissertation, University of Texas at Austin.

Home Office, 2004. Crime Reduction Toolkits: Public Transport. [Online]. http://www.crimereduction.co.uk/toolkits/pt00.htm. Accessed 20 April 2008.

Kaminski, R., E. Jefferis, and C. Chanhatasilpa. 2000. A spatial analysis of American police killed in the line of duty. In: *Atlas of Crime: Mapping the Criminal Landscape*, eds. L. Turnbull, H. E. Hendrix, and B. D. Dent, pp. 212–220, Phoenix, AZ: Oryx Press.

Koperski, K., J. Adhikary, and J. Han. 1996. Spatial data mining: Progress and challenges, In: *Proc. of ACM SIGMOD Workshop on Research Issues on Data Miining and Knowledge Discovery*, 55–70. Montreal, Canada.

Koperski, K. and J. Han. 1995. In: *Proc. of Int'l Symp. on Large Spatial Databases*, 47–66, Portland, Oregon;.

Kowalczyk, A. and B. Raskutti. 2002. *SIGKDD Explorations* 4:99.

Kruengkrai, C. and C. Jaruskulchai. 2003. Using One-Class SVMs for Relevant Sentence Extraction. *Proc. of the International Symposium on Communications and Information Technologies*.

Kulldorff, M. 1997. *Communications in Statistics: Theory and Methods* 26:1481.

Lesage, J. 1997. *Journal of Regional Analysis and Policy* 27:83.

LeSage, J. 1999a. *MATLAB Toolbox for Spatial Econometrics*, http://www.spatial-econometrics.com.

LeSage, J. 1999b. *The Theory and Practice of Spatial Econometrics*, unpublished manuscript, Department of Economics, University of Toledo, Ohio.

Levine, N. 2002. *CrimeStat: A Spatial Statistics Program for the Analysis of Crime Incident Locations* (Vol. 2.0). Houston, TX: Ned Levine and Associates; Washington, D.C.: National Institute of Justice.

Lu, W., J. Han, and B. Ooi. 1993. Discovery of general knowledge in large spatial databases. In: *Proc. of Far East Workshop on Geographic Information Systems*, 275–289.

Manevitz, L. and M. Yousef. 2001. *Machine Learning* 2:139.

McFadden, D. 1973. Conditional logit analysis of qualitative choice behavior. In: *Frontiers in Econometrics*, ed. P. Zarembka, pp. 105–142, New York: Academic Press.

Messner, S., L. Anselin, R. Baller, D. Hawkins, G. Deane, and S. Tolnay. 1999. *Journal of Quantitative Criminology* 15:423.

Miller, H. and J. Han. 2001. *Geographic Data Mining and Knowledge Discovery: An Overview*. Cambridge: Cambridge University Press.

Murray, A. 2000. *Geographical Analysis* 32:1.

Murray, A. and V. Estivill-Castro. 1998. *Journal of Geographical Information Science* 12:431.

Ozesmi, S. and U. Ozesmi. 1999. *Ecological Modeling* 116:15.

Ozesmi, U. and W. Mitsch. 1997. *Ecological Modeling* 101:139.
Pellegrini, P. A. and A. Fotheringham. 2002. *Progress in Human Geography* 26:487.
Scholkopf, B., J. C. Platt, J. S. J. Taylor, A. Smola, and R. C. Williamson. 1999. Estimating the support of a high dimensional distribution. *Neural Computation*, 1443–1471.
Shapiro, G. P. and W. Frawley. 1991. *Knowledge Discovery in Databases*. Cambridge, MA: AAAI/MIT Press.
Shekhar, S., P. Zhang, Y. Huang, and R. Vatsavai. 2004. Trend in spatial data mining. In: *Data Mining: Next Generation Challenges and Future Directions*, eds. H. Kargupta, A. Joshi, K. Sivakumar, and Y. Yesha, pp. 357–381, AAAI/MIT Press.
Tax, D. and R. Duin. 2001. Outliers and data descriptions. In R. L. Lagendijk, J. W. J. Heinjnsdijk, A. D. Pimentel, and M. H. F. Wilkinson (eds.), *Proc. ASCI 2001, 7th Annual Conference of the Advanced School for Computing and Imaging*, (Heijen, NL, May 30–June 1), ASCI, Delft, 234–241.
Tax, D. and R. Duin. 2002. *Machine Learning Research* 2:155.
Vapnik, V. 1998. *Statistical Learning Theory*. New York: John Wiley.
Vapnik, V. 1999. *The Nature of Statistical Learning Theory*, 2nd ed. New York: Springer.
Wahba, G. 1990. Spline models for observational data, CBMS-NFS Regional Conference Series, Society for Industrial and Applied Mathematics, Philadelphia, PA.
Xue, Y., D. B. 2003. *IEEE Transactions on Systems, Man, and Cybernetics: Part C* 33:78.
Zeng, D., W. Chang, and H. Chen. 2004. In: *Proc. of IEEE International Conference on Intelligent Transportation Systems*, 106–111, Washington, DC.

NOTE

1. Home Office, 2004. Crime Reduction Toolkits: Public Transport. [Online]. http://www.crimereduction.co.uk/toolkits/pt00.htm. Accessed 20 April 2008.