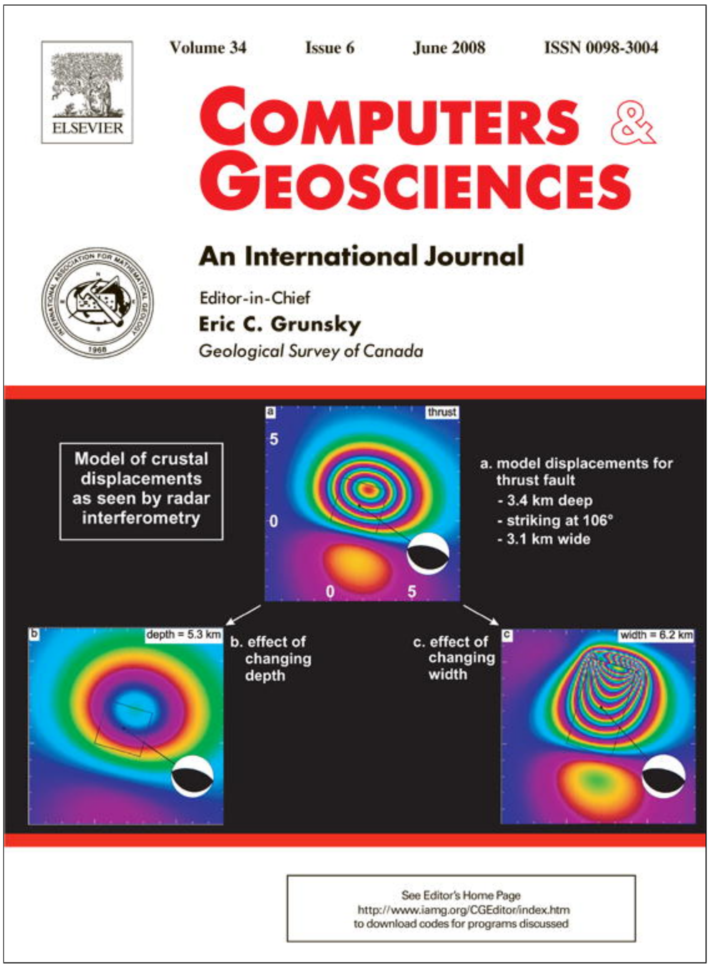
Volume 34
Issue 6
June 2008
ISSN 0098-3004
ELSEVIER
COMPUTERS & GEOSCIENCES
An International Journal
Editor-in-Chief
Eric C. Grunsky
Geological Survey of Canada
Model of crustal displacements as seen by radar interferometry
a. model displacements for thrust fault
- 3.4 km deep
- striking at 106°
- 3.1 km wide
b. effect of changing depth
c. effect of changing width
depth = 5.3 km
width = 6.2 km
See Editor's Home Page
http://www.iamg.org/CGEditor/index.htm
to download codes for programs discussed

ELSEVIER

Computers & Geosciences 34 (2008) 592–602

COMPUTERS & GEOSCIENCES

www.elsevier.com/locate/cageo

Cellular automata for simulating land use changes based on support vector machines

Qingsheng Yang[a,b], Xia Li[a,c,*], Xun Shi[d]

[a]School of Geography and Planning, Sun Yat-sen University, 135 West Xingang Road, Guangzhou 510275, PR China
[b]Department of Tourism and Environment, Guangdong University of Business Studies, Guangzhou 510230, PR China
[c]Guangzhou Institute of Geography, Guangzhou 510275, PR China
[d]Department of Geography, Dartmouth College, Hanover, NH 03755, USA

Received 9 March 2006; received in revised form 31 March 2007; accepted 22 August 2007

Abstract

Cellular automata (CA) have been increasingly used to simulate urban sprawl and land use dynamics. A major issue in CA is defining appropriate transition rules based on training data. Linear boundaries have been widely used to define the rules. However, urban land use dynamics and many other geographical phenomena are highly complex and require nonlinear boundaries for the rules. In this study, we tested the support vector machines (SVM) as a method for constructing nonlinear transition rules for CA. SVM is good at dealing with nonlinear complex relationships. Its basic idea is to project input vectors to a higher dimensional Hilbert feature space, in which an optimal classifying hyperplane can be constructed through structural risk minimization and margin maximization. The optimal hyperplane is unique and its optimality is global. The proposed SVM-CA model was implemented using Visual Basic, ArcObjects®, and OSU-SVM. A case study simulating the urban development in the Shenzhen City, China demonstrates that the proposed model can achieve high accuracy and overcome some limitations of existing CA models in simulating complex urban systems.

Keywords: Cellular automata; Support vector machines; Nonlinear transition rules; Urban simulation

1. Introduction

Since first introduced by Ulam (1976) in 1948, cellular automata (CA) have been widely used to simulate nonlinear complex systems (Wolfram, 1984; Itami, 1994). Recently, there is an increasing application of CA in the simulation of urban systems. Some important example studies include Batty and Xie's (1994, 1997) pioneering work that simulates land use dynamics in the city of Buffalo, NY with CA and GIS; Clarke's et al. (1997) simulation of urbanization of the San Francisco Bay Area; White and Engelen's (1997) use of constrained CA in simulating land use changes in Cincinnati, OH; and Li and Yeh's (2001, 2002) development of a neural network-based CA model for simulating rapid urban development in Southern China. These studies have demonstrated CA as a powerful tool for simulating complex urban dynamics.

*Corresponding author at: School of Geography and Planning, Sun Yat-sen University, 135 West Xingang Road, Guangzhou 510275, PR China. Tel.: +80 20 33047500; fax: +80 20 84115833.
E-mail addresses: qsyang2002@163.com (Q. Yang), lixia@mail.sysu.edu.cn (X. Li), xun.shi@dartmouth.edu (X. Shi).

0098-3004/$ - see front matter © 2007 Elsevier Ltd. All rights reserved.
doi:10.1016/j.cageo.2007.08.003

CA models can generate complex global patterns by using simple local rules. These rules determine how a *cell* will evolve under certain conditions. While the rules are only applied at the neighborhood level—the notion of neighborhood is central to the CA paradigm (Couclelis, 1997)—they should represent the impact of factors at different (i.e., local, regional, and global) spatial scales (Wu and Webster, 1998; Li and Yeh, 2000). The transition rules are the key inputs in a CA model. Traditionally, the rules have been defined in linear forms, using methods such as multi-criteria evaluation (MCE) and logistical regression (Wu and Webster, 1998; Wu, 2002). Apparently, however, linear transition rules cannot adequately accommodate the nonlinear characteristics of complex urban systems, and there is a need for nonlinear transition rules. To address this problem, Li and Yeh (2002) define transition rules using neural networks, which improve the capability of CA in dealing with nonlinear complexity. However, they found it difficult to interpret the parameter values and the inference process (Li and Yeh, 2004). Moreover, the training of neural networks may result in local rather than global optimization (Vapnik, 1998). As neural networks are not well-controlled learning machines (Vapnik, 1998), there is research interest in exploring other methods for defining nonlinear transition rules for CA.

This paper presents a study that tests the support vector machines (SVM) as a method for defining nonlinear transition rules for CA. SVM is a data mining technique whose performance has been proven in many applications, such as credit scoring (Baesens et al., 2003), financial time series prediction (Gestel et al., 2001), spam categorization (Drucker et al., 1999) and brain tumor classification (Lu et al., 1999). The strength of this technique lies with its ability to model nonlinearity (Martens et al., 2007). That SVM training always finds a global solution is in contrast to the case of neural networks, where many local minima usually exist (Vapnik, 1998). SVM operates by projecting input vectors to a Hilbert space in which they can be linearly classified by a hyperplane. The hyperplane is derived by applying a kernel function to certain *support vectors* (Vapnik, 1998; Joachims, 2002; Ambriola et al., 2003). In the SVM-CA model developed in this study, the transition rule is constructed by combining the output from SVM and other contextual and constraint information. The final output from the transition rule is a *development probability*. This proposed model was tested using urban development data from Shenzhen City, China, for the period 1988–2004. The model was also used to forecast the development status of the city in 2010.

2. Support vector machines and SVM-CA urban model

2.1. Support vector machines (SVM)

SVM projects input vectors to a higher dimensional Hilbert feature space, where an optimal separating hyperplane can be constructed (Chang and Lin, 2003; Cherkassky and Ma, 2004; Webb, 2002). A linear separation in the Hilbert space can be constructed using kernel functions. Based on the principles of structural risk minimization (SRM) and maximum classification margin, SVM minimizes the upper bound of the expected generalization error, which leads to a global optimization (Vapnik, 1998). This section outlines the procedure and readers are referred to Vapnik's (1998) for details.

Consider the scenario of separating a set of training vectors belonging to two separate classes, $T=\{(x_1,y_1),(x_2,y_2),\ldots,(x_l,y_l)\}$, where $x_i \in x = R^n$, $y_i \in y\{1,-1\}$, $i=1,2,\ldots,l$, where y_i is the class of sample i, and x_i represents a group of attributes of sample i. In our case, y_i indicates whether cell i will be converted to an urban area or not, and x_i represents the variable relevant to that conversion.

A separating hyperplane H can be created:

$$H:\ w\cdot x+b=0, \tag{1}$$

where w is a weight vector and b is a scalar. The separating hyperplane is the optimal hyperplane if all training data are separated without error and the distance between the closest vector and the hyperplane is maximal. The hyperplane can be described as follows:

$$\begin{cases} w\cdot x+b\geqslant 1 & \text{if}\quad y_i=1,\\ w\cdot x+b\leqslant -1 & \text{if}\quad y_i=-1, \end{cases} \tag{2}$$

which is equivalent to

$$y_i(w\cdot x+b)\geqslant 1,\quad i=1,\ldots,l. \tag{3}$$

The separating hyperplane can then be formalized as a decision function:

$$f(x)=\mathrm{sgn}(w\cdot x+b). \tag{4}$$

The two parameters of the separating hyperplane decision function, w and b, can be obtained by solving the following optimization problem:

$$\min L(w,b,\xi) = \frac{1}{2}(w^{\mathrm{T}} \cdot w) + C\left(\sum_{i=1}^{l} \xi_i\right) \tag{5}$$

subject to

$$\begin{cases} y_i(w \cdot x + b) \geqslant 1 - \xi_i, & i = 1, \ldots, l, \\ \xi_i \geqslant 0, & i = 1, \ldots, l. \end{cases} \tag{6}$$

The variables ξ_i are *slack variables,* representing the error in the classification. The first part of the objective function tries to maximize the margin between the two classes in the feature space, whereas the second part minimizes the misclassification error. The positive real constant C is a tuning parameter in the algorithm.

The solution to this optimization problem is given by the saddle point of the Lagrange function:

$$L(w,b,\xi;\alpha) = L(w,b,\xi) - \sum_{i=1}^{l} \alpha_i \{y_i[wx+b] - 1 + \xi_i\} - \sum_{i=1}^{l} \xi_i, \tag{7}$$

where α_i are Lagrange multipliers. The Lagrangian has to be minimized with respect to w and b and maximized with respect to $\alpha_i > 0$.

The Lagrange multipliers α_i are then determined by the following optimization (dual problem):

$$\min_a \frac{1}{2} \sum_{i=1}^{l} \sum_{j=1}^{l} y_i y_j a_i a_j (x_i \cdot x_j) - \sum_{j=1}^{l} a_j \tag{8}$$

subject to

$$\sum_{i=1}^{l} y_i a_i = 0, \tag{9}$$

$$0 < a_i \leqslant c, \quad i = 1, \ldots, l. \tag{10}$$

Let $\alpha_0 = (\alpha_1^0, \ldots, \alpha_l^0)$ be a solution to this optimization problem. Then the normal of the vector w_0 corresponding to the optimal hyperplane equals

$$|w_0|^2 = 2W(\alpha_0) = \sum_{\text{support vector}} \alpha_i^0 \alpha_j^0 (x_i \cdot x_j) y_i y_j. \tag{11}$$

The separating rule, based on the optimal hyperplane, is the following decision function:

$$f(x) = \operatorname{sgn}\left[\sum_{i=1}^{l} y_i a_i^0 (x_i \cdot x) + b_0\right], \tag{12}$$

where x_i are the support vectors, a_i^0 are the corresponding Lagrange coefficients, and the constant b_0 is defined as

$$b_0 = \tfrac{1}{2}[(w_0 \cdot x^*(1)) + (w_0 \cdot x^*(-1))], \tag{13}$$

where $x^*(1)$ are some support vectors belonging to the first class and $x^*(-1)$ are some support vectors belonging to the second class.

If the training data are linearly separable, then a set of $\{w, b\}$ pairs can be found such that the constraints in Eq. (4) are satisfied (Fig. 1(A)).

If the training data cannot be classified linearly then a projection function $\Phi(x)$ is used to map the training data from the original data space x to a Hilbert space X (Vapnik, 1998). In this higher dimensional space X, the linear SVM formulation of Eq. (4) can be applied (Figs. 1(B and C)) so that the data can be linearly separable. In the SVM optimization function, the feature information in the training data appear in the form of inner products $(x_i \cdot x_j)$ (Eq. (8)). This is also the case in the decision function (Eq. (12)). In the Hilbert space X, $(x_i \cdot x_j)$ is replaced by inner products $\Phi(x_i) \cdot \Phi(x_j)$.

It is usually difficult to acquire projection function $\Phi(x)$ due to its complexity. However, $\Phi(x_i) \cdot \Phi(x_j)$ can be replaced by kernel functions according to Hilbert–Schmidt theory (Vapnik, 2000). Hence the optimization problem of Eq. (8) with the same subjections becomes

$$\min_a \frac{1}{2} \sum_{i=1}^{l} \sum_{j=1}^{l} y_i y_j a_i a_j k(x_i \cdot x_j) - \sum_{j=1}^{l} a_j, \tag{14}$$

where $k(\cdot)$ is the kernel function. The decision function (Eq. (12)) then becomes

$$f(x) = \operatorname{sgn}\left[\sum_{i=1}^{l} y_i a_i^0 k(x_i, x) + b_0\right] \tag{15}$$

and the constant b_0 in Eq. (13) becomes

$$b_0 = y_i - \sum_{i=1}^{l} y_i a_i^0 k(x_i, x_j). \tag{16}$$

Many kernel functions can be used for this purpose, such as the polynomial function $k(x,y) = (x \cdot y + 1)^p$, the radial base function $k(x,y) = \exp[-(x-y)^2/2\sigma^2]$, and the sigmoid function $k(x,y) = \tanh(kx \cdot y - \delta)$.

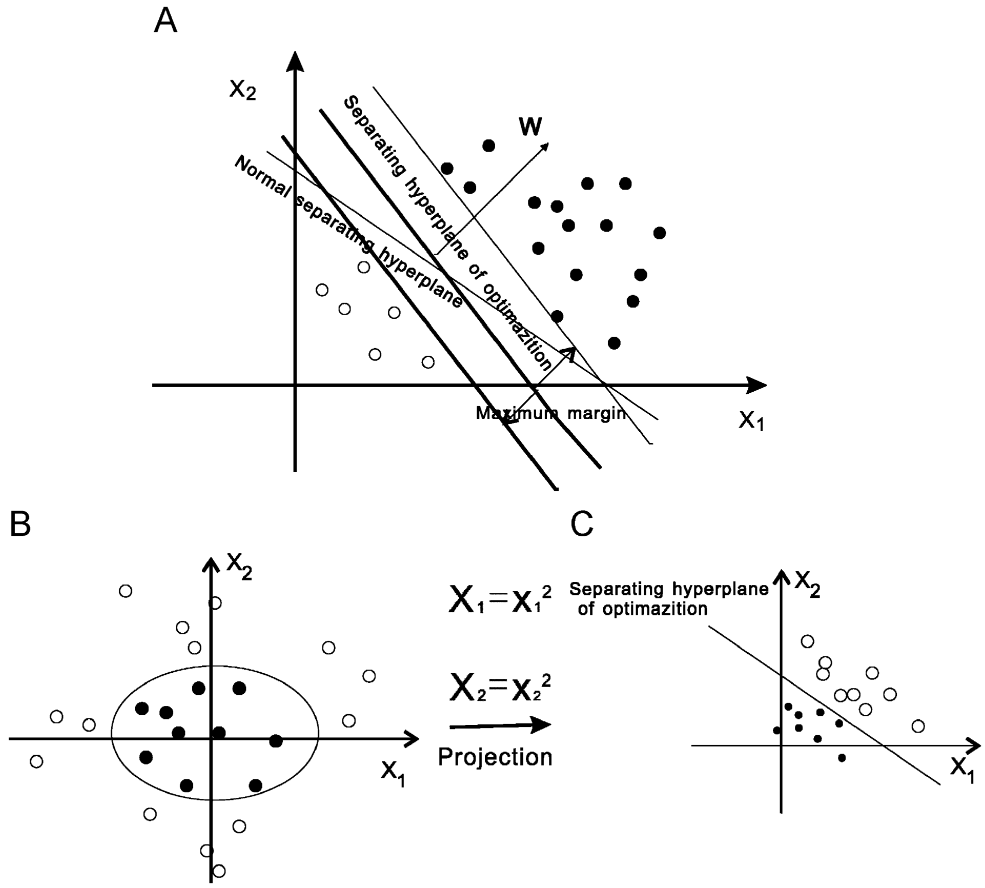

Fig. 1. Linear classification and nonlinear classification spaces, and mapped into linear classification space. (A) Linear classification, (B) nonlinear classification, and (C) linear classification after projection.

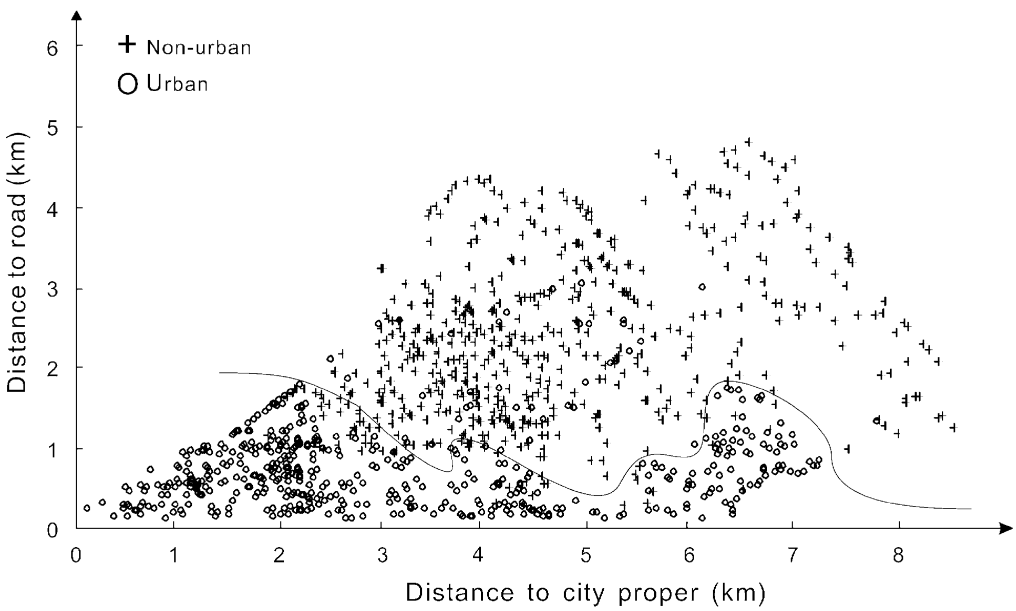

Fig. 2. Nonlinear classification boundaries between non-urban and urban cells.

2.2. The SVM-CA urban model

When using CA to simulate urban land use dynamics, the transition rule (the classification boundary that determines whether a non-urban cell will convert into an urban cell) can be highly complicated, and linear equations become insufficient (Fig. 2). In this study, we used SVM to construct nonlinear transition rules for the CA simulations. Specifically, in each iteration of simulation, the urban development probability is estimated based on the output from the decision function (Platt, 1999; Ana et al., 2004):

$$p_x = \frac{1}{1+\exp\{-[\sum_{i=1}^{l} y_i a_i^0 k(x_i, x) + b_0]\}}, \tag{17}$$

where p_x is the urban development probability of vector x in the current simulation iteration; x contains the values of the considered variables of the cell that needs to be classified; x_i the ith support vector; $k(\cdot)$ the kernel function; and l the number of support vectors. In this study we used a radial base function for k:

$$k(x, x_i) = \mathrm{e}^{-(||x-x_i||^2/2\sigma^2)}, \tag{18}$$

where σ is the width of the radial base function.

Considering the influence of neighborhood, the urban development probability of cell k in iteration t, $p_{k,t}$, can be refined as follows:

$$p_{k,t} = \frac{1}{1+\exp[-(\sum_{i=1}^{l} y_i a_i^0 \mathrm{e}^{-(||x_k-x_i||^2/2\sigma^2)} + b_0)]} \times \Omega_{3\times 3,k}^{t}, \tag{19}$$

where x_k is the vector of spatial variables of cell k; $\Omega_{3\times 3,k}^{t}$ the total number of urban cells within the 3×3 neighborhood of cell k in iteration t; and $||x_k - x_i||^2$ is calculated as follows:

$$||x_k - x_i||^2 = (x_k - x_i)^{\mathrm{T}} \cdot (x_k - x_i)$$

where

$$x_k = (x_{k1}, x_{k2}, \ldots, x_{ko})^{\mathrm{T}}$$

and

$$x_i = (x_{i1}, x_{i2}, \ldots, x_{io})^{\mathrm{T}}. \tag{20}$$

In the above equation x_{ko} is the oth spatial variable of cell k and x_{io} is the oth spatial variable of support vector i.

Some variables can be used to represent constraints to land development. For example, development is not allowed in lands protected for agricultural or ecological purposes, and is impossible in rivers and mountains. With the constraint factors considered, the urban development probability can be revised as (Wu, 1998)

$$p_{k,t} = \frac{1}{1+\exp[-(\sum_{i=1}^{l} y_i a_i^0 \mathrm{e}^{-(||x_k-x_i||^2/2\sigma^2)} + b_0)]} \times \Omega_{3\times 3,k}^{t} \times \prod_{j=1}^{m} \mathrm{cons}_{k,j}, \tag{21}$$

where $\mathrm{cons}_{k,j}$ is the jth constraint factor that is applicable to cell k and m the number of constraint factors. The value of a constraint factor varies between 0 and 1.

A stochastic disturbance term can be added to represent unknown errors, which may help to generate a more realistic pattern. The error term (RA) is defined as (White and Engelen, 1993)

$$\mathrm{RA} = 1 + (-\ln \gamma)^a, \tag{22}$$

where γ is a random number within the range of 0 and 1; and a is a factor controlling the magnitude of the perturbation. The urban development probability can then be revised as

$$p_{k,t} = [1 + (-\ln \gamma)^a] \times \frac{1}{1+\exp[-(\sum_{i=1}^{l} y_i a_i^0 \mathrm{e}^{-(||x_k-x_i||^2/2\sigma^2)} + b_0)]} \times \Omega_{3\times 3,k}^{t} \times \prod_{j=1}^{m} \mathrm{cons}_{j,k}. \tag{23}$$

In iteration t, $P_{k,t}$ is determined as follows:

$$\begin{cases} p_{k,t} \geqslant \alpha & \text{development}, \\ \text{Otherwise} & \text{undevelopment}, \end{cases} \tag{24}$$

where α is a predefined threshold.

The simulation of urban development is conducted by running the model iteratively until certain conditions are satisfied, e.g., the total amount of simulated urban land equals the actual amount of urban land. Fig. 3 shows the procedure of simulating urban development using the SVM-CA urban model.

3. A case study in Shenzhen City

3.1. Study area and data

The proposed model was used to simulate the urban development in Shenzhen City, China, a city that has experienced drastic urban expansion in recent years. The expansion has been thoroughly monitored using remote sensing (Yeh and Li, 2001; Li and Yeh, 1998). The spatial variables used in this simulation were derived from existing remote sensing and GIS data (Table 1). Specifically, the information about urban development was derived from the land use data that were generated through classification of Landsat TM images. The distance variables were calculated using the *Eucdistance* function in ArcGIS®. The number of developed

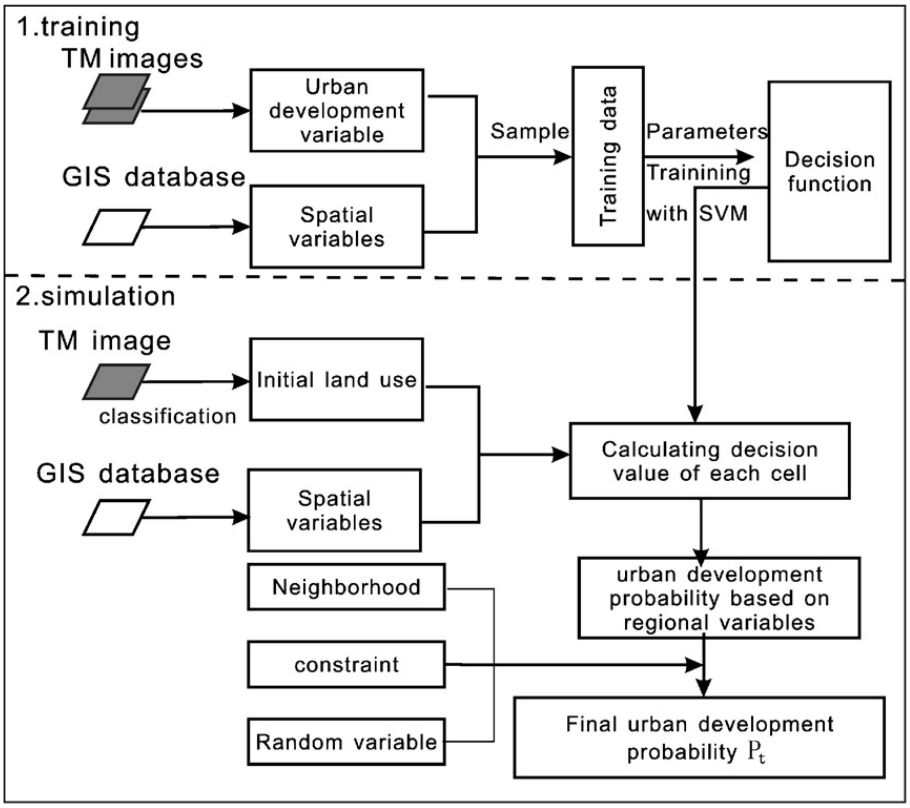

Fig. 3. SVM-CA model for simulation of urban dynamics.

Table 1
Variable in SVM-CA model

Target variable	y: urban development (1: converted to urban land; −1: not converted
Proximity variable	x_1: distance to the major (city proper) urban areas (unit: m) x_2: distance to the closest town centers (unit: m) x_3: distance to the closest roads (unit: m) x_4: distance to the closest railways (unit: m) x_5: distance to the closest expressways (unit: m)
Local variable	$\Omega^t_{3\times3}$: number of urban cells in the 3×3 neighborhood (0–9 pixels)
Constraint variables	c_1: water (0, water;1, others) c_2: forest (0, forest;1, others) c_3: urban green land (0, urban green land;1, others) c_4: crop protection land (0, crop protection land;1, others)

cells in the 3×3 neighborhood was counted using the *Focal* function of ArcGIS®. The data for the constraint variables were acquired from governmental agencies or through classification of remotely sensed data. The initial state for the simulation was created based on the classification of the 1988 TM image.

3.2. Simulation program

The proposed SVM-CA model was implemented using Visual Basic 6.0®, ArcObjects®, and an SVM package, OSU-SVM. ArcObjects provides access to spatial data, as well as tools for distance calculation and focal operations. OSU-SVM is an SVM toolbox developed at The Department of Electrical and Computer Engineering, Ohio State University, USA. It contains fundamental SVM, n-SVM, and one-class SVM classifiers for regression and classification, and is capable of dealing with large training sets (Ma, Zhao, and Ahalt, 2002, OSU SVM Classifier Matlab Toolbox, http://www.kernel-machines.org/, last accessed on May 23, 2006). OSU-SVM can be downloaded from http://sourceforge.net/projects/svm (last accessed on March 29, 2007). The CA modeling and the communication between different components of the model were programmed using Visual Basic.

3.3. Calibration (training)

Land development data from 1988 to 1993 were used to train the model. Since the images are large, a stratified sampling method (Congalton, 1991) was used to ensure that the training data had a reasonable size. The random stratified sample points were generated using ERDAS IMAGINE®. Their values for the considered spatial variables were then retrieved using the sample function of ArcGIS®. The sample set was used to obtain values for c in Eq. (10) and for σ, a_i^0, b_0 in Eq. (23). Particularly, c and σ were acquired using the grid-search method (Martens et al., 2007) and the search results are $c=1$ and $\sigma=1$. Table 2 lists some acquired support vectors and their corresponding Lagrange coefficients a_i^0, y_i, and $y_i a_i^0$. When c, σ, and a_i^0 are known, b_0 can be calculated with Eq. (16). Based on Eq. (23), the urban development probability of cell

Table 2
Part support vectors and a_i*, y_i, $y_i a_i$*

x_1	x_2	x_3	x_4	x_5	a_i*	y_i	$y_i a_i$*
25,982.00	2015.50	531.50	1501.00	27,934.00	0.94	1	0.94
18,474.50	1790.50	141.50	1050.00	14,385.50	0.94	1	0.94
24,514.50	5845.50	3200.50	4160.00	20,850.00	1.00	1	1.00
21,662.00	3448.00	1503.50	3252.50	8019.00	1.00	1	1.00
19,769.00	4452.50	3451.50	3708.50	11,130.50	0.98	1	0.98
26,341.00	5543.00	3200.00	850.00	26,972.50	0.77	1	0.77
25,460.00	1557.50	4830.50	6053.00	16,204.00	0.94	1	0.94
26,580.50	5402.00	3510.50	2433.00	24,446.50	0.96	1	0.96
19,909.00	3900.00	985.00	1520.50	22,106.00	0.25	1	0.25
16,523.00	934.00	50.00	1803.00	19,234.50	0.98	1	0.98
19,769.00	4452.50	3451.50	3708.50	11,130.50	0.21	1	0.21
24,416.50	4013.00	3688.00	2823.00	16,901.00	0.25	1	0.25
30,633.50	2728.00	971.00	2886.00	17,824.00	0.81	−1	−0.81
25,230.00	2862.50	269.50	1553.00	12,410.50	0.50	−1	−0.50
10,879.50	5301.00	728.00	2751.50	4699.00	0.76	−1	−0.76
2169.00	3699.00	250.00	1700.00	4826.00	0.53	−1	−0.53
9917.50	4114.00	750.00	3371.00	2732.00	0.81	−1	−0.81
16,178.00	3721.50	900.00	364.00	7115.00	0.75	−1	−0.75
8841.00	4733.50	2522.50	550.00	8492.50	0.79	−1	−0.79
13,371.00	4855.00	652.00	50.00	996.50	0.81	−1	−0.81

Table 3
SVM-CA simulation results for some cells

x_1	x_2	x_3	x_4	x_5	x_6	Actual value	Probability
1812.00	1834.50	471.50	1600.00	600.00	5	1	0.85
2836.50	2504.50	353.50	856.00	1350.00	4	1	0.79
7645.00	2988.50	756.50	2074.00	4258.00	1	1	0.38
9875.00	3516.50	1601.00	1710.50	3300.50	0	0	0.23
27,130.00	3529.00	2084.00	16,348.00	2598.50	0	0	0.02
30,051.00	5233.00	3760.50	14,865.50	6720.50	0	0	0.15
31,769.00	3610.00	1650.00	25,187.50	1209.50	0	1	0.64
30,171.50	814.00	515.00	27,178.50	672.50	0	1	0.59
26,501.00	3189.50	2850.00	18,166.00	7658.00	0	0	0.44
24,701.00	4313.50	7110.50	13,821.00	5784.50	0	0	0.31
20,751.50	2402.50	2511.00	8299.00	1543.50	0	0	0.03
18,230.50	4523.50	1718.50	18,173.00	2294.00	0	0	0.12
22,235.50	1947.50	1242.00	24,852.00	250.00	0	0	0.25
15,538.50	3699.00	1930.00	15,118.50	3413.00	0	0	0.38
9209.00	4991.00	550.00	6239.50	3970.00	2	1	0.71
11,787.50	4430.00	2722.50	2280.50	854.50	0	0	0.17
16,141.00	756.50	180.50	640.50	3335.50	2	1	0.49
15,366.50	5661.00	1188.50	17,800.00	1150.00	4	1	0.51
14,117.50	5377.00	2039.50	7114.00	559.00	0	0	0.07
13,022.00	2997.00	743.50	9637.50	710.50	0	0	0.34

k in iteration t was obtained as follows:

$$p_{k,t} = [1 + (-\ln r)^5] \times \frac{1}{1 + \exp[-(\sum_{i=1}^{l} y_i a_i^0 \mathrm{e}^{-0.5\times\|x_k - x_i\|^2})(-0.19154)]} \times \Omega_{3\times3,k}^t \times \prod_{j=1}^{m} cons_{k,j}. \quad (25)$$

We conducted heuristic experiments to determine the value for the predefined threshold α in Eq. (24), arriving at a value of 0.65 for this parameter. The simulation terminating condition was set such that the amount of simulated land conversion is equal to that of the actual urban development occurring during the simulated period.

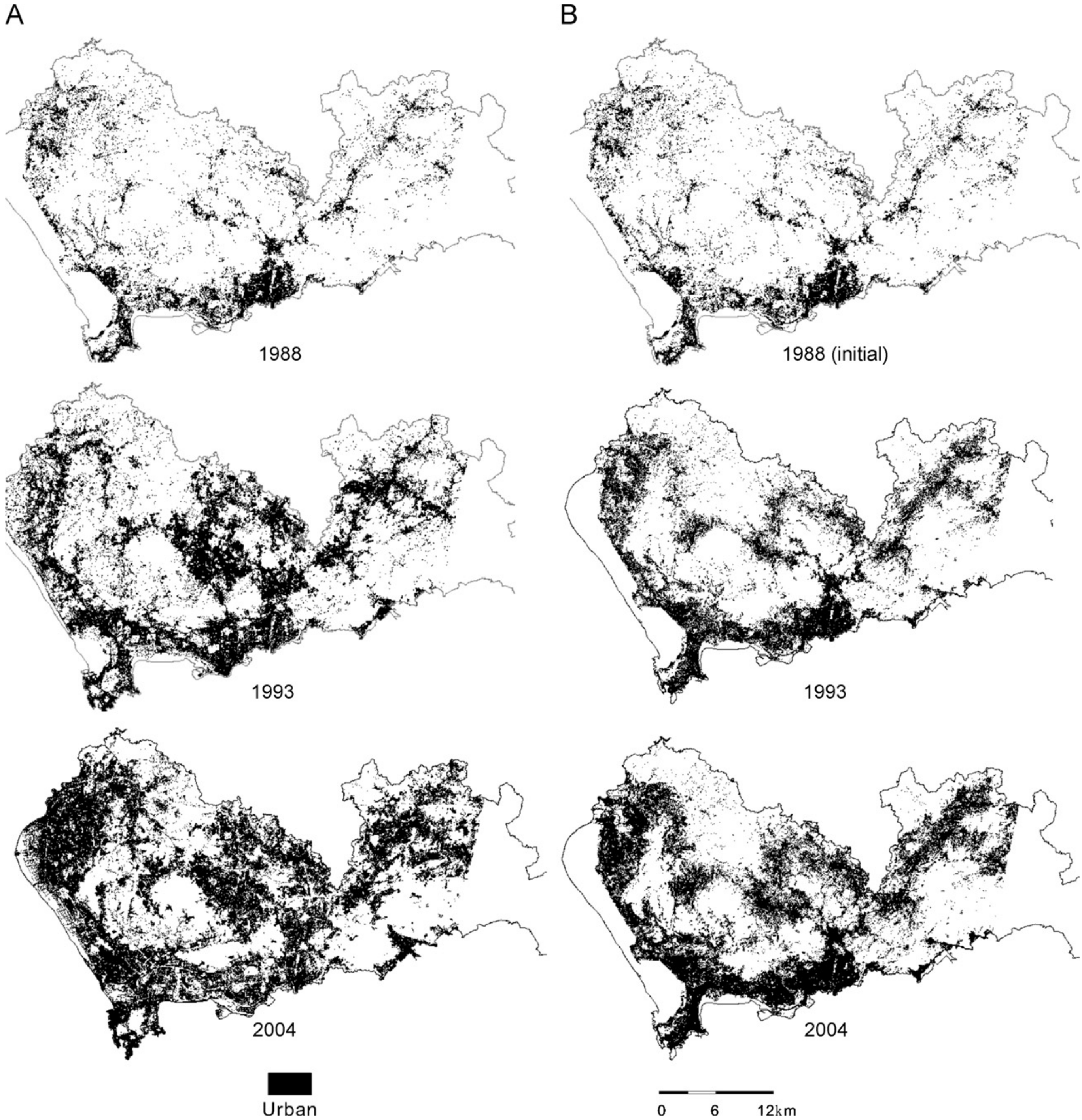

Fig. 4. Simulation of urban dynamics from 1988 to 2004 based on SVM-CA model. (A) Actual urban land of Shenzhen City and (B) simulated urban land of Shenzhen City.

3.4. Simulation result

Table 3 shows specific simulation values for some cells and Fig. 4 presents a comparison of the actual urban development detected from remotely sensed data and the simulation results from the SVM-CA model. Visual inspection indicates that the simulation generated an urban morphology similar to the actual situation. For example, both maps show that the urban land development was mainly distributed around urban centers in the early 1990s, and spread along roads in the later years.

Cell-by-cell comparison was typically used to evaluate the simulation accuracy (Clarke's et al., 1997; Wu, 2002; Li and Yeh, 2002, 2004). It evaluates the similarity between the actual and simulated situations at the scale of a single cell. In this study, cell-by-cell comparison was used to evaluate the simulation accuracy too. The results of the cell-by-cell comparison for two periods, 1988–1993 and 1994–2004, were given by the confusion matrices in Tables 4 and 5, respectively. For 1988–1993, the accuracy for developed land is 67.88% and the overall accuracy is 87.25%. For 1994–2004, they become 71.09% and 84.90%, respectively. The kappa coefficients were calculated to quantify the actual degree of agreement (Cohen, 1960; Campbell, 1987; Fung and LeDrew, 1988; Congalton, 1991). The coefficient was 0.70 for 1988–1993 and 0.68 for 1994–2004. Considering the training data were only from the 1988 to 1993 images, the similar accuracies for the two periods might be an indication that the development mechanism is relatively stable in this region. This, in turn, provides justification for using the model trained by the 1988–1993 data to forecast future development.

To further assess the performance of this particular SVM-CA model, we applied a logistic regression-based CA (LR-CA) suggested by Wu (1998) to the same sample set extracted from the 1988 to 1993 TM imageries. Table 6 is the error matrix of the LR-CA. A comparison of Table 4 with Table 6 shows that the proposed nonlinear SVM-CA significantly outperforms the linear LR-CA. Moreover, the LR-CA may have multi-collinearity problems when the variables are highly correlated (Wu, 1998).

Assuming stability in the development mechanism in the Shenzhen area, we use the model calibrated with the 1988–1993 data to project future urban development there. The projection results for 2004–2010 are given in Fig. 5.

Table 4
Confusion matrix between actual and simulated urban in 1993 based on SVM-CA (number in each category refers to number of pixels in that category)

	Simulation		
	Non-urban	Urban	Accuracy (%)
Actual			
Non-urban	495,838	9388	98.14
Urban	91,229	192,804	67.88
Overall			87.25
Kappa		0.70	

Table 5
Confusion matrix between actual and simulated urban in 2004 based on SVM-CA (number in each category refers to number of pixels in that category)

	Simulation		
	Non-urban	Urban	Accuracy (%)
Actual			
Non-urban	432,064	22,391	95.07
Urban	96,781	238,023	71.09
Overall			84.90
Kappa		0.68	

Table 6
Confusion matrix between actual and simulated urban in 1993 based on LR-CA (number in each category refers to number of pixels in that category)

	Simulation		
	Non-urban	Urban	Accuracy (%)
Actual			
Non-urban	485,233	75,166	86.58
Urban	99,804	129,056	56.39
Overall			77.83
Kappa		0.44	

4. Conclusion

This paper presents an experiment using SVM to obtain nonlinear transition rules for CA simulation of urban land use dynamics. The basic idea of SVM is to project nonlinear input vectors to Hilbert space, where the projected vectors can be classified linearly using an optimal hyperplane. In Hilbert

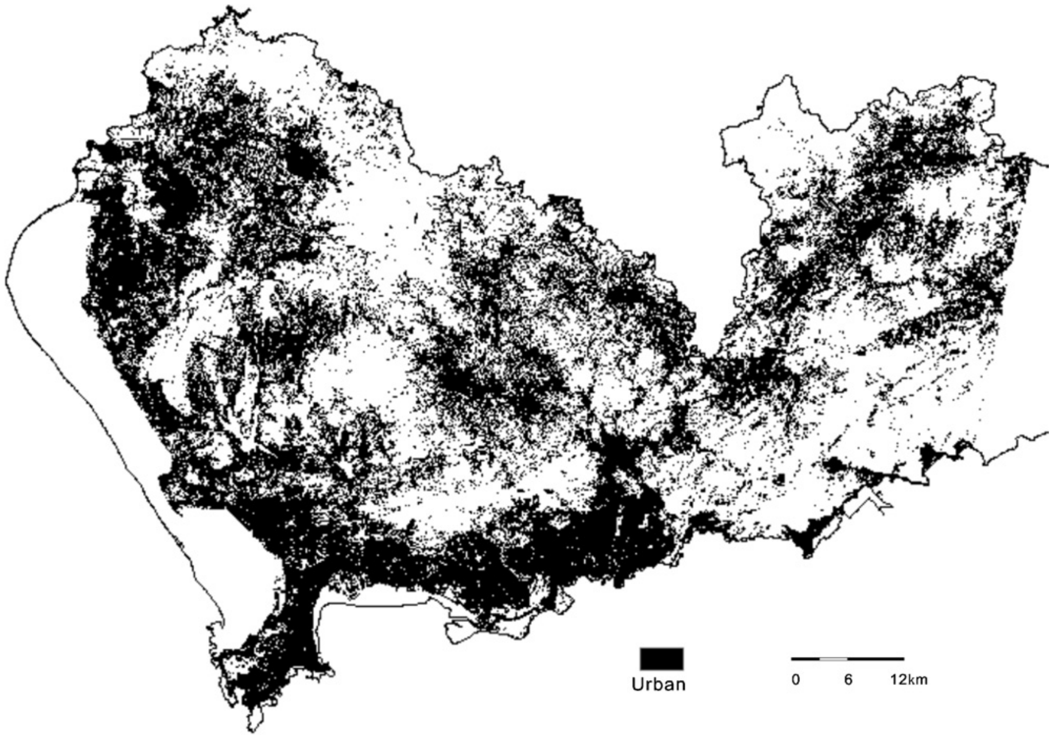

Fig. 5. Simulation of urban land use for 2010 based on SVM-CA model.

space, the decision function of the optimal separating hyperplane is constructed using the kernel function and the support vectors. The decision function of the optimal hyperplane is then used to form the transition rule for CA. All the parameter values in the transition rule are derived from training data.

The model is applied to the simulation of urban development in Shenzhen City, a fast-developing, new city in Southern China. It simulates the urban development of this city from 1988 to 2004 and also predicts development for the years 2004–2010. The validation results show good conformity between the actual and simulated urban development patterns, and the proposed method achieved a considerably higher overall accuracy when compared with the linear logistical regression model. In addition, the training process of SVM is based on SRM and maximal margin distance theory, which leads to a globally optimal solution, an advantage over artificial neural networks. Thus we conclude that the CA model based on SVM is a promising tool for simulating urban growth.

Acknowledgments

This study was supported by the National Outstanding Youth Foundation of China (Grant no. 40525002), the National Natural Science Foundation of China (Grant no. 40471105), and the Hi-tech Research and Development Program of China (863 Program) (Grant no. 2006AA12Z206).

References

Ambriola, M., Bellotti, R., Circella, M., Maglietta, R., Stramaglia, S., 2003. Supervised algorithms for particle classification by a transition radiation detector. Nuclear Instruments and Methods in Physics Research A 510 (3), 362–370.

Ana, M., Dragan, N., Leopold, C., 2004. Probabilistic SVM outputs for pattern recognition using analytical geometry. Neurocomputing 62, 293–303.

Baesens, B., Gestel, T.V., Viaene, S., Stepanova, M., Suykens, J., Vanthienen, J., 2003. Benchmarking state-of-the-art classification algorithms for credit scoring. Journal of the Operational Research Society 54 (6), 627–635.

Batty, M., Xie, Y., 1994. From cells to cities. Environment and Planning B: Planning and Design 21, 531–548.

Batty, M., Xie, Y., 1997. Possible urban automata. Environment and Planning B 24, 175–192.

Campbell, J.B., 1987. Introduction to Remote Sensing. The Guilford Press, New York, 551pp.

Chang, C., Lin, C., 2001. Training ν-support vector classifiers: theory and algorithms. Neural Computation 13, 2119–2147.

Cherkassky, V., Ma, Y., 2004. Practical selection of SVM parameters and noise estimation for SVM regression. Neural Networks 17, 113–126.

Clarke, K.C., Hoppen, S., Gaydos, L., 1997. A self-modifying cellular automaton of historical urbanization in the San

Francisco Bay area. Environment and Planning B 24, 247–261.

Cohen, J., 1960. A coefficient of agreement for nominal scale. Educational and Psychological Measurement 20, 37–46.

Congalton, R.G., 1991. A review of assessing the accuracy of classification of remotely sensed data. Remote Sensing of Environment 37, 35–46.

Couclelis, H., 1997. From cellular automata to urban models: new principles for model development and implementation. Environment and Planning B 24, 165–174.

Drucker, H., Wu, D., Vapnik, V., 1999. Support vector machines for spam categorization. IEEE Transactions on Neural Networks 10 (5), 1048–1054.

Fung, T., LeDrew, E., 1988. The determination of optimal threshold levels for change detection using various accuracy indices. Photogrammetric Engineering and Remote Sensing 54, 1449–1454.

Gestel, T.V., Suykens, J., Baesens, B., Lambrechts, A., Lanckriet, G., Vandaele, B., Moor, B., Vandewalle, J., 2001. Financial time series prediction using least squares support vector machines with the evidence framework. IEEE Transactions on Neural Networks 12 (4), 809–821.

Itami, R.M., 1994. Simulating spatial dynamics: cellular automata theory. Landscape and Urban Planning 30, 24–47.

Joachims, T., 2002. Learning to Classify Text Using Support Vector Machines: Methods, Theory and Algorithms. Kluwer Academic Publishers, Norwell, MA, 224pp.

Li, X., Yeh, A.G.O., 1998. Principal component analysis of stacked multi-temporal images for monitoring of rapid urban expansion in the Pearl River Delta. International Journal of Remote Sensing 19, 1501–1518.

Li, X., Yeh, A.G.O., 2000. Modelling sustainable urban development by the integration of constrained cellular automata and GIS. International Journal of Geographical Information Science 14, 131–152.

Li, X., Yeh, A.G.O., 2001. Calibration of cellular automata by using neural networks for the simulation of complex urban systems. Environment and Planning A 33, 1445–1462.

Li, X., Yeh, A.G.O., 2002. Neural-network-based cellular automata for simulating multiple land use changes using GIS. International Journal of Geographical Information Science 16, 323–343.

Li, X., Yeh, A.G.O., 2004. Data mining of cellular automata's transition rules. International Journal of Geographical Information Science 18, 723–744.

Lu, C., Gestel, T.V., Suykens, J., Huffel, S., Vergote, I., Timmerman, D., 1999. Preoperative prediction of malignancy of ovarium tumor using least squares support vector machines. Artificial Intelligence in Medicine 28 (3), 281–306.

Martens, D., Baesens, B., Van Gestel, T., Vanthienen, J., 2007. Comprehensible credit scoring models using rule extraction from support vector machines. European Journal of Operational Research 183 (3), 1466–1476.

Platt, J.C., 1999. Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. In: Smola, A., Bartlett, P., Schlkopf, B., Schuurmans, D. (Eds.), Advances in Large Margin Classifiers. MIT Press, Cambridge, MA, pp. 61–74.

Ulam, S.M., 1976. Adventures of a Mathematician. Charles Scribner's Sons, New York, 317pp.

Vapnik, V., 1998. Statistical Learning Theory. Wiley, New York, 768pp.

Vapnik, V., 2000. The Nature of Statistical Learning Theory, second ed. Springer, New York, 314pp.

Webb, A.R., 2002. Statistical Pattern Recognition, second ed. John Wiley & Sons, West Sussex, 496pp.

White, R., Engelen, G., 1993. Cellular automata and fractal urban form: a cellular modelling approach to the evolution of urban land-use patterns. Environment and Planning A 25, 1175–1199.

White, R., Engelen, G., 1997. Cellular automata as the basis of integrated dynamic regional analysis. Environment and Planning B: Planning and Design 24, 235–246.

Wolfram, S., 1984. Cellular automata as models of complexity. Nature 31 (4), 419–424.

Wu, F., 1998. SimLand: a prototype to simulate land conversion through the integrated GIS and CA with AHP-derived transition rules. International Journal of Geographical Information Science 12, 63–82.

Wu, F., 2002. Calibration of stochastic cellular automata: the application to rural–urban land conversions. International Journal of Geographical Information Science 16, 795–818.

Wu, F., Webster, C.J., 1998. Simulation of land development through the integration of cellular automata and multicriteria evaluation. Environment and Planning B 5, 103–126.

Yeh, A.G.O., Li, X., 2001. A constrained CA model for the simulation and planning of sustainable urban forms by using GIS. Environment and Planning B 28, 733–753.