Stochastic GIS cellular automata for land use change simulation: application of a kernel based model

O. Okwuashi[1], J. McConchie[2], P. Nwilo[3], E. Eyo[4]

[1]School of Geography, Environment, and Earth Sciences,
Victoria University of Wellington,
Wellington,
New Zealand
+64-211227793
onuwa.okwuashi@vuw.ac.nz

[2]Opus International Consultants,
P.O. Box 12003,
Wellington,
New Zealand
+64-272115973
Jack.McConchie@opus.co.nz

[3]Department of Surveying and Geoinformatics,
University of Lagos,
Lagos,
Nigeria
+234-8035725644
pcnwilo@yahoo.com

[4]Department of Surveying and Geoinformatics,
Federal University of Technology,
Minna,
Nigeria
+234-8067080950
geoetim@yahoo.com

1. Introduction

Cellular Automata (CA) are dynamic mathematical systems (which are discrete in time and space, operate in uniform regular lattice, and are characterized by local interaction) that can be used with GIS to simulate land use change. The use of Artificial Neural Network (ANN) for GIS cellular automata calibration is one of the most popular stochastic GIS CA calibration techniques.

ANNs shortcomings are that ANNs are "black-box" models, and have a static nature in which causal factors are undynamic (Kocabas and Dragicevic 2007); and might suffer difficulties with generalization and produce models that may overfit the data (Karystinos and Pados 2000). This study introduces the use of a kernel based model called "Support Vector Machine (SVM)" for calibrating the GIS cellular automata. SVMs are robust, dynamic, and unsusceptible to overfitting. SVM results are compared with the ANN.

2. CA calibration using support vector machine

A more detailed SVM introduction is given in Cortes and Vapnik (1995) and Watanachaturaporn et al. (2004). Given a training dataset which consists of *n* training

samples $(x_1, y_1),(x_2, y_2),...,(x_n, y_n)$, where $x_i \in \Re^N$, such that $y_i \in \{-1,+1\}$. The objective of SVMs is to find a linear decision function defined by $f(x) = w \cdot x + b$, where $w \in \Re^N$, and $b \in \Re$ is a bias. The hyperplanes for the two classes are represented by $y_i(w \cdot x + b) \geq 1$. Slack variables $\xi > 0$ account for misclassification. $y_i(w \cdot x + b) \geq 1 - \xi_i$ represents the hyperplanes for the two classes. The optimal hyperplane $f(x) = 0$ is located where the margin between the two classes is maximized and error minimized. The constrained optimization problem is,

$$Min: \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{n} \xi_i \tag{1}$$

$$Subject \quad to: y_i(w \cdot x + b) \geq 1 - \xi_i, \ for \quad i = 1,2,...,n.$$

The constant C, $0 < C < \infty$, is called the penalty value. Equation 1 is solved by constructing the Lagrangian,

$$L(w,b,\xi,\alpha,\beta) = \tfrac{1}{2} w \cdot w + C\sum_{i=1}^{n} \xi_i - \sum_{i=1}^{n} \alpha_i (y_i(w \cdot z_i) + b) - 1 + \xi_i) - \sum_{i=1}^{n} \beta_i \xi_i \tag{2}$$

and finding the saddle point of $L(w,b,\xi,\alpha,\beta)$. The dual form of the solution of (2) becomes:

$$Max: \sum_{i=1}^{n} \alpha_i - \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j (x_i \cdot x_j) \tag{3}$$

$$Subject \quad to: \quad \sum_{i=1}^{n} \alpha_i y_i = 0 \quad and \quad 0 \leq \alpha_i \leq C, \quad for \quad i = 1,2,...,n.$$

$\alpha_i \geq 0$ are called the Lagrange multipliers. According to the Karush-Kuhn-Tucker (KKT) optimality condition (Fletcher 1987); some of the multipliers will be zero. Multipliers with nonzero values are called the support vectors. The result from the optimizer, called an optimal solution, is the set $\alpha^0 = (\alpha_1^0,...,\alpha_n^0)$; w and b are calculated from $w^0 = \sum_{i=1}^{n} y_i \alpha_i^0 x_i$ and $b^0 = \frac{1}{2}\left[w^0 \cdot x_{+1}^0 + w^0 \cdot x_{-1}^0\right]$, where x_{+1}^0 and x_{-1}^0 are the support vectors of class labels +1 and -1. The decision rule is then applied to classify the dataset into two classes +1 and −1,

$$f(x) = sign\left[\sum_{i=1}^{n} y_i \alpha_i^0 (x_i \cdot x) + b^0\right]. \tag{4}$$

For a nonlinear problem the transformation function ϕ maps the data into a higher dimensional space. Suppose there exists a function K, called a kernel function, such that,

$$K(x_i, x_j) \equiv \phi(x_i) \cdot \phi(x_j). \tag{5}$$

The formulation of the kernel function from the dot product is a special case of Mercer's theorem (Mercer 1909). The optimization problem becomes,

$$Max: \sum_{i=1}^{n} \alpha_i - \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j K(x_i, x_j) \tag{6}$$

$$Subject \quad to: \quad \sum_{i=1}^{n} \alpha_i y_i = 0 \qquad and \qquad 0 \le \alpha_i \le C, \qquad for \qquad i = 1,2,...,n.$$

while the decision function becomes,

$$f(x) = sign\left[\sum_{i=1}^{n} y_i \alpha_i^0 K(x_i, x) + b^0\right] \tag{7}$$

Examples of some well-known kernel functions are: linear kernel $(x \cdot x_i)$, polynomial function $(x \cdot x_i + 1)^d$, radial basis function $\exp\left(-\frac{\|x - x_i\|^2}{2\sigma^2}\right)$, and sigmoid function $\tanh(\kappa(x \cdot x_i) + \Theta)$.

Now we can calibrate our CA using SVM outputs. We can map the SVM outputs $f(x)$, into probabilities using a sigmoid function with parameters A and B (Platt 1999),

$$P(1/f) = \frac{1}{1 + \exp(Af + B)} \tag{8}$$

Let $P(1/f)$ be replaced with $P_{i,j}^t$, where $P_{i,j}^t$ is the development probability at time t. For stochastic CA, a stochastic perturbation term can be incorporated to represent unknown errors during the simulation; in order to ensure the predicted patterns approximate reality as closely as possible. The error term (RA) is given by White and Engelen (1993) as:

$$RA = 1 + (-\ln \gamma)^{\alpha} \ . \tag{9}$$

In order to increase control over the perturbation, (9) can be modified as (Okwuashi et al. 2009):

$$RA = \lambda\left\langle \beta + (-\ln \gamma)^{\alpha} \right\rangle . \tag{10}$$

where γ is a uniform random variable within the range 0 and 1; and λ, α, and β are parameters that control the magnitude of the perturbation.

The development probability at $t+1$ can be revised as:

$$P_{i,j}^{t+1} = \left[\lambda\{\beta + (-\ln \gamma)^{\alpha}\}\right] * P_{i,j}^t * \prod_{i,j=1}^{m} cons_{i,j} \tag{11}$$

where $cons_{i,j}$ denotes constraints' contributions.

$P_{i,j}^{t+1}$ is determined as:

$$\begin{cases} P_{i,j}^{t+1} \ge \psi & developed \\ Otherwise & undeveloped \end{cases} \tag{12}$$

where ψ implies predefined threshold.

3. Methodology

This study employed mainly remote sensing Landsat Thematic Mapper images acquired December 18, 1984 and February 6, 2000 covering Lagos, Nigeria; and several other data sources. Fourteen land use variables were extracted for learning (table 1). These variables were extracted with the ArcGIS.

The distance variables were calculated using the Euclidean Distance function. The number of developed cells in the 3 x 3 Moore's neighbourhood was first computed using the Focal Statistics function, while the updated neighbourhood and modelling was done in MATLAB. The stratified random sampling was used to extract the training data. The following values were used for the calibration: $\lambda = 0.058$, $\beta = 5$, $\alpha = 1$, $A = -1$, $B = 0$, $\psi = 0.2$, polynomial function of degree $d = 2$, and $C = 10e-08$. Basically, α_i^0, b^0, and $f(x)$ were computed. Only the neighbourhood variable x_{14}, was updated in every iteration to determine a new $f(x)$, $P_{i,j}^t$, and $P_{i,j}^{t+1}$. Undeveloped cells that have development probability greater than or equal to the threshold probability ψ were converted to developed cells. The visualization of results was done in ArcGIS. The ANN was trained with the method of *back-propagation,* using a 'two-layer feed-forward network' with 55 neurons in the hidden layer.

Land use variables	
Target variables	$\begin{cases} y = +1 & developed \\ y = -1 & undeveloped \end{cases}$
Proximity variables	x_1: distance to water x_2 : distance to residential area x_3: distance to industrial and commercial centres x_4 : distance to major roads x_5: distance to Lagos Island x_6 : distance to international airport x_7 : distance to local airport x_8: distance to Apapa Port x_9 : distance to Tin Can Island Port x_{10} : distance to all settlements x_{11}: distance to all vegetated cells
Weighted variables	x_{12} : population potential

	x_{13}: income potential
Local variable	x_{14}: Ω^{t}_{3x3} is the number of developed cells in the 3×3 Moore neighbourhood at time t ($0-9$ pixels)
Constraint variables	Major roads, water, and developed cells

Table 1. The fourteen land use variables

4. Discussion and results

Figure 1 presents the actual land use development derived from a remotely sensed image and the simulated result from the support vector machine GIS-CA model. The SVM result of the cell-by-cell comparison for periods, 1984-2000 is given by the confusion matrix in table 2 and the result from the ANN method is given by the confusion matrix in table 3.

Kappa statistic was calculated for the SVM and ANN respectively (see figure 2). The kappa coefficient can provide much better interpretation for measuring accuracy because it can address the difference between the actual agreement and chance agreement (Fung and LeDrew 1988). SVM performed better than ANN judging by their kappa coefficients.

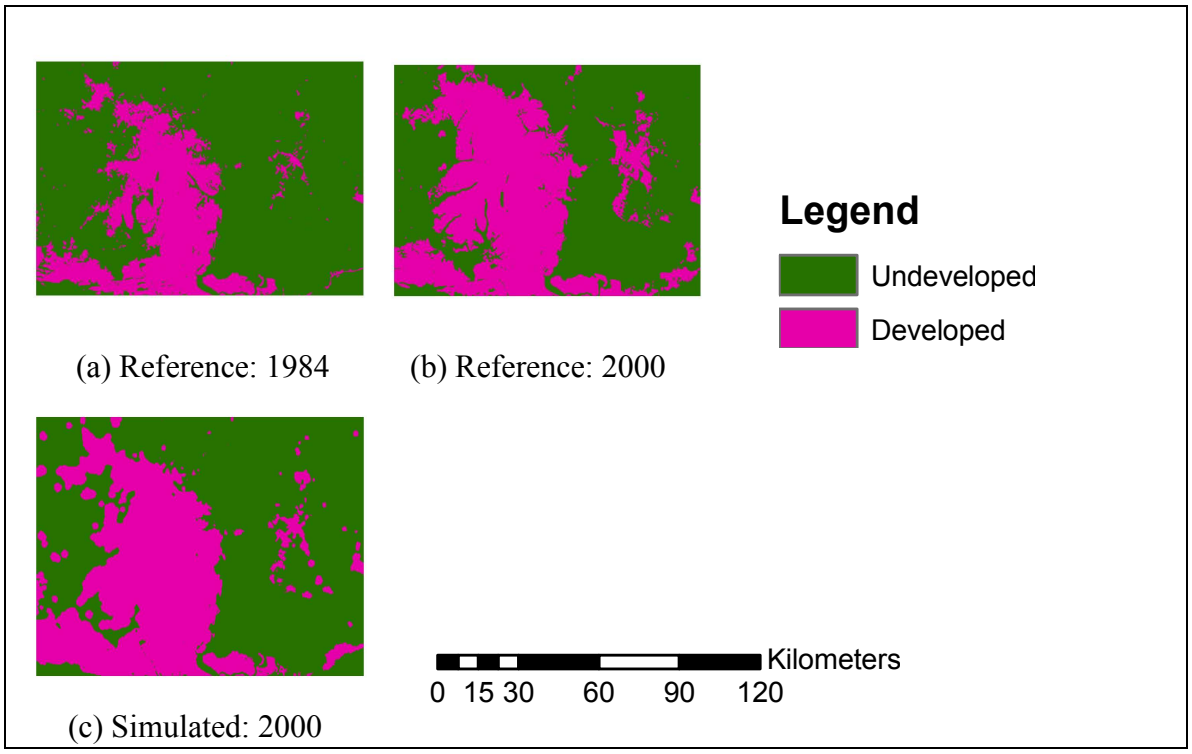

Figure 1. (a) Actual base year: 1984, (b) Actual target year: 2000, and (c) SVM simulated target year: 2000

	Reference data	
	Developed	Undeveloped
Predicted data		
Developed	57523	7706
Undeveloped	14440	95831

Table 2. SVM confusion matrix

	Reference data	
	Developed	Undeveloped
Predicted data		
Developed	56223	16644
Undeveloped	15740	86893

Table 3. ANN confusion matrix

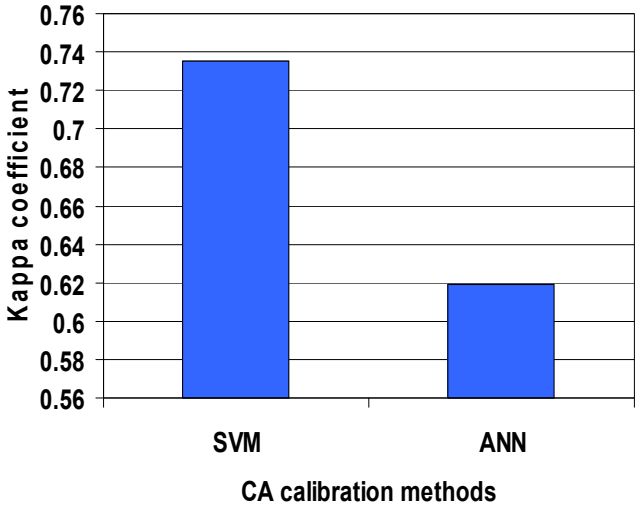

Figure 2. Validation of models: Kappa coefficients for SVM and ANN

5. Conclusion

The result of this modelling showed good conformity between the simulated and the actual land use development. The SVM posted a better result than the ANN model. SVMs are relatively new tools that have been applied to various fields of study, but have not been favourably adopted for modelling land use change. SVMs may be computationally intensive, but studies have shown that their results are highly accurate, and seem a promising tool for simulating land use change.

6. References

Cortes C and Vapnik V, 1995, Support vector networks. *Machine learning* 20(3): 273-297.
Fletcher R, 1987, *Practical Methods of Optimization*. 2nd Ed., John Wiley & Sons, New York.
Fung T and LeDrew E, 1988, The determination of optimal threshold levels for change detection using various accuracy indices. *Photogrammetric Engineering and Remote Sensing,* 54(10): 1449–1454.

Karystinos G and Pados D, 2000, On overfitting, generalization, and randomly expanded training sets. *IEEE Transactions on Neural Networks*, 11(5), 1050-1057.

Kocabas V and Dragicevic S, 2007, Enhancing a GIS Cellular Automata Model of Land Use Change: Bayesian Networks, Influence Diagrams and Causality. *Transactions in GIS*, 11(5): 681-702.

Mercer J, 1909, Functions of positive and negative type and their connection with the theory of integral equations. *Transactions of the London Philosophical Society,* A-209: 415-446.

Okwuashi O, McConchie J, Nwilo P, & Eyo E, 2009, Enhancing a GIS cellular automata model of land use change using support vector machine. *The 17th International Conference on Geoinformatics 2009*, Fairfax, Virginia, USA.

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

Watanachaturaporn P, Varshney P, and Arora M, 2004, Evaluation of factors affecting support vector machines for hyperspectral classification. *American Society for Photogrammetry & Remote Sensing (ASPRS) 2004 Annual Conference,* Denver, CO.

White R and 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(8):1175-1199.