GLOBAL JOURNAL OF ENVIRONMENTAL SCIENCES VOL 10, NO.1&2, 2011: 35-46
 ISSN 1596-6194
www.globaljournalseries.com; Email: info@globaljournalseries.com

SUPERVISED REMOTE SENSING IMAGE CLASSIFICATION: AN EXAMPLE OF A ONE-AGAINST-ONE MULTI-CLASS POLYNOMIAL KERNEL BASED SUPPORT VECTOR MACHINE

ONUWA OKWUASHI, ETIM EYO AND ANIEKAN EYOH

(Received 27, June 2011; Revision Accepted 23, August 2011)

ABSTRACT

Software like ILWIS and GRASS GIS can be employed for remote sensing image processing and geographic information systems applications. The modules of the aforementioned image processing software are based on conventional multi-class classifiers/algorithms such as maximum likelihood classifier. These conventional multi-class classifiers/algorithms are usually written in programming languages such as C, C^{++}, and python. The objective of this research is to experiment the use of a binary classifier/algorithm for multi-class remote sensing task, implemented in MATLAB. MATLAB is a programming language just like C, C^{++}, and python. In this research, the support vector machine binary classifier/algorithm based on a one-against-one approach implemented in MATLAB is applied to remote sensing multi-class problem. Both simulated and empirical satellite remote sensing data are used to train and test a one-against-one support vector machine classifier. For the purpose of validating the experiment, the resulting classified satellite image is compared with the ground truth data. The polynomial kernel function is used for the modelling. In the simulated application, 25 pixels are used for the experiment, out of which 6 pixels are used for training while 19 pixels are used for testing. Out of the 19 tested pixels 18 pixels are correctly classified while only 1 pixel is left unclassified. In the empirical application, 256 and 7182 pixels are unclassified and misclassified respectively out of a total of 62500 pixels; and the computed overall accuracy of the experiment is 88.1%. The satisfactory result of the experiment indicates substantial agreement between the classification result and the reference data.

KEY WORDS: Image Classification; One-Against-One; Remote Sensing; Support Vector Machine

1. Preamble

Support Vector Machine (SVM) (Cortes & Vapnik, 1995) is intrinsically a binary classifier (Melgani & Bruzzone, 2004). However, applications of binary classification are very limited especially in remote sensing land cover classification where most of the classification problems involve more than two classes. Two prominent methods of implementing multi-class tasks using binary classifiers are: one-against-one (1A1) and one-against-all (1AA). One major disadvantage of 1A1 and 1AA is that both methods often yield unclassified regions. The objective of this work therefore is to illustrate how a binary 1A1 polynomial kernel based SVM classifier can be applied to multi-class satellite remote sensing task. Both simulated and empirical data are applied in this research to illustrate the implementation of a 1A1 SVM approach.

2. Support vector machine

The concept of the SVM was introduced by Cortes and Vapnik (1995). SVM employs the principle of Structural Risk Minimization (SRM), which makes them robust and independent of

underlying data distributions (Joachims, 1999).

Given a binary classification problem that belongs to classes -1 and +1 respectively; these two classes can be separated with a linear hyperplane (see Figure 1). To separate these two sets of objects, we need to choose a few training samples. Now, let us assume that our training set has *n*-training samples, that is, $(x_1, y_1),(x_2, y_2),...,(x_n, y_n)$, where $x_i \in \Re^N$ is an *N* dimensional vector that belongs to one of classes $y_i \in \{-1,+1\}$. The stated binary classification problem can only be separated using a linear decision function (Vapnik, 2000),

$$f(x) = w \cdot x + b \quad (1)$$

where $w \in \Re^N$ is a vector that determines the orientation of our desired hyperplane required for the separation, and $b \in \Re$ is called the "bias."

We can see from Figure 1 that our optimal hyperplane needed to separate the two objects is,

$$y_i(w \cdot x + b) \geq 1 \ . \quad (2)$$

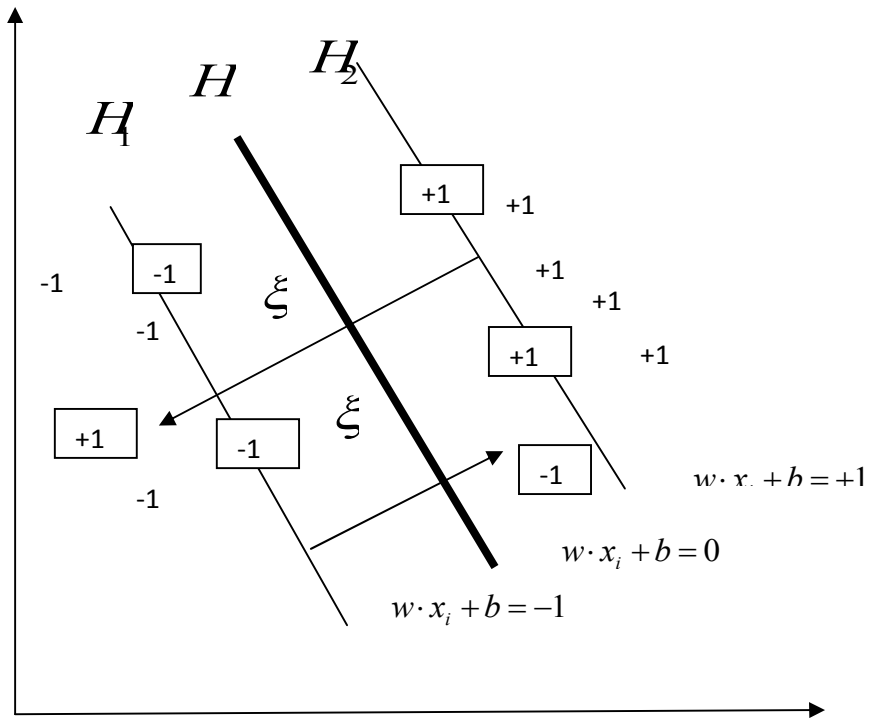


Figure 1: Separating non-separable data with linear separable hyperplanes. Adapted from *Ivanciuc* (2007, p. 318)

represent objects that were misclassified. Let us now incorporate the slack variable in equation 2; which can be revised as,

$$y_i(w \cdot x + b) \geq 1 - \xi_i \ . \quad (3)$$

We can see from Figure 1 that our optimal hyperplane is $f(x) = 0$, which lies between classes +1 and -1; it is actually located at the point of maximum separation between classes +1 and -1, as well as the point of minimum error in course of the separation. At this point, the solution to this problem can be found by solving the following constrained optimization problem (or primal problem) (Vapnik, 2000),

Minimise $\frac{1}{2} w \cdot w + C \sum_{i=1}^{n} \xi_i \quad (4)$

subject to: $y_i(w \cdot x + b) \geq 1 - \xi_i$, $\xi_i > 0$, and for $\forall i = 1,...,n$; where C, $0 < C < \infty$, is called the penalty value or regularization parameter.

According to Ivanciuc (2007), C is a trade-off between misclassified points and achieving the maximum margin during the training; C is usually chosen by trial-and-error. According to Vapnik (2000), we can solve the primal problem given in equation 4 using the Lagrangian function,

$$L(w,b,\xi,\alpha,\beta) = \tfrac{1}{2} w \cdot w + C \sum_{i=1}^{n} \xi_i - \sum_{i=1}^{n} \alpha_i (y_i (w \cdot x_i + b) - 1 + \xi_i) - \sum_{i=1}^{n} \beta_i \xi_i \quad (5)$$

and parameters $L(w, b, \xi, \alpha, \beta)$ must satisfy,

$$\frac{\partial L(w,\xi,\alpha,\beta)}{\partial w} = w - \sum_{i=1}^{n} \alpha_i y_i x_i = 0 \ , \quad (6)$$

$$\frac{\partial L(w,\xi,\alpha,\beta)}{\partial b} = -\sum_{i=1}^{n} \alpha_i y_i = 0 \ , \quad (7)$$

$$\frac{\partial L(w,\xi,\alpha,\beta)}{\partial \xi_i} = C - \alpha_i - \beta_i = 0 \ . \quad (8)$$

The optimization problem or dual form resulting from the application of equations 6 – 8 to the primal problem given in equation 4 can be expressed as,

Maximise: $\sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j (x_i \cdot x_j) \ , \quad (9)$

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

Therefore, the decision function for the linear case can be given as,

$$f(x) = sign\left[\sum_{i=1}^{n} y_i \alpha_i^0 (x_i \cdot x) + b^0 \right] \quad (10)$$

where x_i are the training samples; y_i are the target labels of the training samples (such that, $y_i \in \{-1,+1\}$); α_i^0 are the Lagrangian multipliers; b^0 is known as the "bias;" while x denotes the test set.

According to the Karush-Kuhn-Tucker (KKT) optimality condition (Fletcher, 1987), some of the the Lagrangian multipliers will be zero. The points of x_i whose Lagrangian multipliers are nonzero values are called "support vectors."

Now let us consider a nonlinearly separable problem; that is a case where a linear hyperplane

function ϕ to map the data onto a higher dimensional feature space (see Figure 2). In that case a kernel function K is introduced, such that (Vapnik, 2000),

$$K(x_i, x_j) = \phi(x_i)^T \phi(x_j) \ . \quad (11)$$

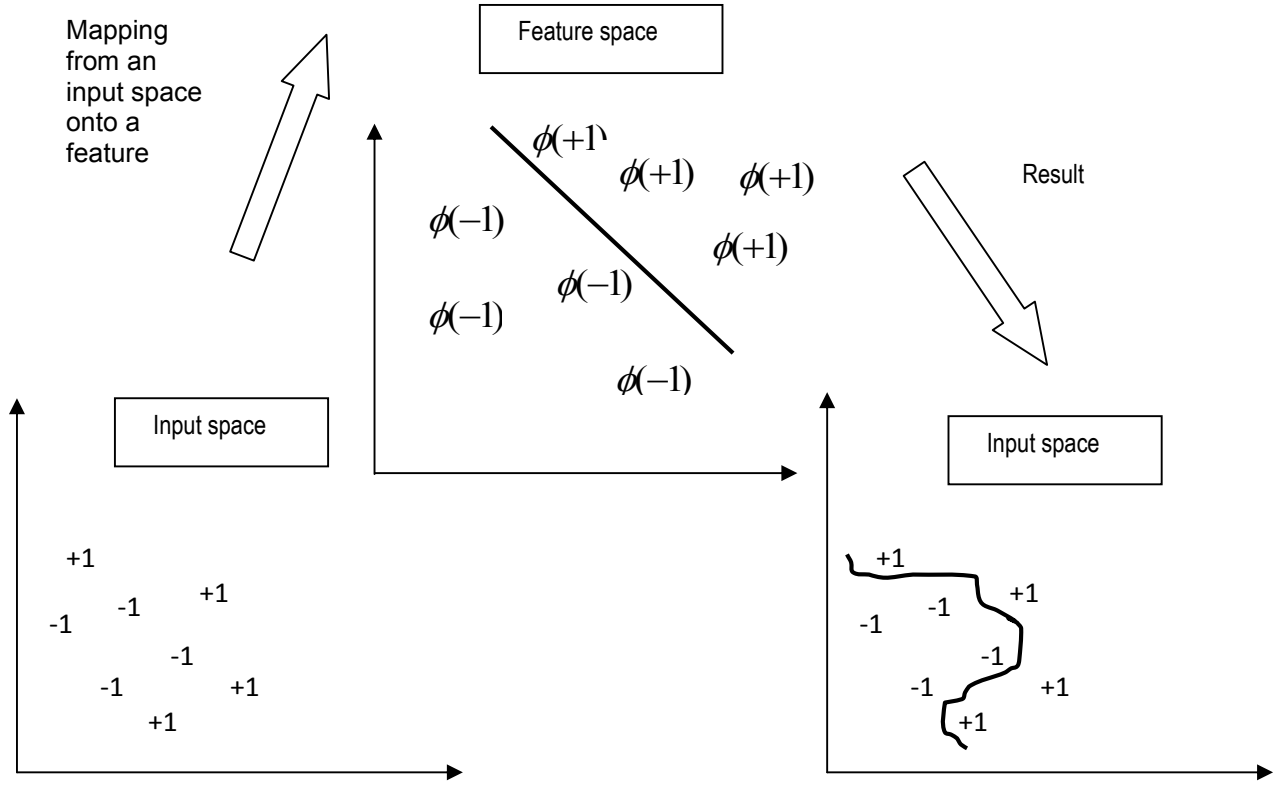


Figure 2: The process of classifying a nonlinearly separable data. Adapted from *Ivanciuc (2007, p. 323)*

Separation in the feature space does not require that ϕ be determined explicitly; therefore it is more convenient to use the kernel function for our computation. The derivation of $K(x_i, x_j)$ from $\phi(x_i)^T \phi(x_j)$ is based on the Mercer's theorem (Mercer, 1909; Cristianini & Shawe-Taylor, 2000). The optimization problem for the nonlinear case can be derived by replacing $x \cdot x_i$ with $K(x_i, x_j)$ in equation 9, and we can revise equation 9 as,

Maximise: $$\sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j K(x_i, x_j) \quad (12)$$

$$\sum_{i=1}^{n}$$

$$f(x) = sign\left[\sum_{i=1}^{n} y_i \alpha_i^0 K(x_i, x) + b^0\right]. \quad (13)$$

Given two arbitrary support vectors $x_A \in$ class *A* and $x_B \in$ class *B*, the bias can be evaluated as,

$$b^0 = -\frac{1}{2}\sum_{i=1}^{n} y_i \alpha_i^0 \left[K(x_A, x_i) + K(x_B, x_i)\right] \text{ (Vapnik, 2000).} \quad (14)$$

The kernel $K(x_i, x_j)$ can be any of the following common kernel functions: the linear kernel $x \cdot x_i$, polynomial kernel $(x \cdot x_i + 1)^d$, and Radial Basis Function (RBF) kernel

$$K(x_i, x_j) = \exp\left(-\frac{\|x_i - x_j\|^2}{2\gamma^2}\right)$$ (Vapnik, 2000). The polynomial and RBF are nonlinear kernel functions. The parameters: gamma γ and polynomial order d control the shape of the separating hyperplane.

3. Simulated application: One-against-one SVM classification

Given a simulated ground truth data (Table 1) with a matrix size of 5 x 5, and equivalent simulated satellite remote sensing multi-spectral data that consist of three spectral bands (see Tables 2, 3, and 4), we intend to the classify the satellite data given in Tables 2, 3, & 4 into three classes: water, undeveloped, and developed. Our objective here is to use the satellite spectral bands given in Tables 2, 3, & 4 to derive the ground truth data given in Table 1. All the three spectral bands in Tables 2, 3, and 4 contain hypothetical DN values.

Table 1: Ground truth data (water=1, undeveloped cells=2, and developed cells=3)

1	1	1	1	2
1	1	1	2	2
3	3	1	2	2
3	3	2	2	2
3	3	3	2	2

Table 2: Band 1

1	0	4	2	26
8	10	9	27	20
42	40	7	26	24
47	43	22	29	30
46	45	50	23	25

Table 3: Band 2

78	73	72	74	103
75	70	80	104	101

Table 4: Band 3

30	36	34	37	66
33	38	31	67	63
90	93	39	68	62
97	96	60	65	61
92	98	99	66	64

To classify the satellite data given in Tables 2, 3, & 4, a training set has to be randomly selected. The training data (six pixels) consist of elements from the three classes (see Table 5).

Table 5: Training data

Water	Band 1 (1,2) = 0	Band 2 (1,2) = 73	Band 3 (1,2) = 36
Water	Band 1 (2,1) = 8	Band 2 (2,1) = 75	Band 3 (2,1) = 33
Undeveloped	Band 1 (1,5) = 26	Band 2 (1,5) = 103	Band 3 (1,5) = 66
Undeveloped	Band 1 (2,4) = 27	Band 2 (2,4) = 104	Band 3 (2,4) = 67
Developed	Band 1 (3,1) = 42	Band 2 (3,1) = 180	Band 3 (3,1) = 90
Developed	Band 1 (4,2) = 43	Band 2 (4,2) = 182	Band 3 (4,2) = 96

For modelling convenience let the remaining nineteen cells that were not used for training the classifier (see Table 6) represent the test set. Conventionally the size of the test set is usually smaller than that of the training set in machine learning. But for the purpose of illustration, let the remaining nineteen cells serve as the test set.

Table 6: Test data

Band 1 (1,1) = 1	Band 2 (1,1) = 78	Band 3 (1,1) = 30
Band 1 (4,1) = 47	Band 2 (4,1) = 186	Band 3 (4,1) = 97
Band 1 (5,1) = 46	Band 2 (5,1) = 188	Band 3 (5,1) = 92
Band 1 (2,2) = 10	Band 2 (2,2) = 70	Band 3 (2,2) = 38
Band 1 (3,2) = 40	Band 2 (3,2) = 190	Band 3 (3,2) = 93
Band 1 (5,2) = 45	Band 2 (5,2) = 184	Band 3 (5,2) = 98
Band 1 (1,3) = 4	Band 2 (1,3) = 72	Band 3 (1,3) = 34
Band 1 (2,3) = 9	Band 2 (2,3) = 80	Band 3 (2,3) = 31
Band 1 (3,3) = 7	Band 2 (3,3) = 76	Band 3 (3,3) = 39
Band 1 (4,3) = 22	Band 2 (4,3) = 100	Band 3 (4,3) = 60
Band 1 (5,3) = 50	Band 2 (5,3) = 183	Band 3 (5,3) = 99
Band 1 (1,4) = 2	Band 2 (1,4) = 74	Band 3 (1,4) = 37
Band 1 (3,4) = 26	Band 2 (3,4) = 106	Band 3 (3,4) = 68
Band 1 (4,4) = 29	Band 2 (4,4) = 109	Band 3 (4,4) = 65
Band 1 (5,4) = 23	Band 2 (5,4) = 105	Band 3 (5,4) = 66
Band 1 (2,5) = 20	Band 2 (2,5) = 101	Band 3 (2,5) = 63
Band 1 (3,5) = 24	Band 2 (3,5) = 108	Band 3 (3,5) = 62
Band 1 (4,5) = 30	Band 2 (4,5) = 107	Band 3 (4,5) = 61
Band 1 (5,5) = 25	Band 2 (5,5) = 110	Band 3 (5,5) = 64

The formulation of the 1A1 technique is such that an $N(N-1)/2$ binary classifiers are required to train any two classes of interest; where N denotes the number of classes. One-against-one (1A1) classification is also called "pairwise classification." The rule of the 1A1 classification is that the class label that occurs most is assigned to that point otherwise that pixel is left unclassified.

The modelling was implemented in MATLAB using the polynomial kernel of degree, *d=2*. The penalty value was, *C=100*. The training and test results are given in Table 7.

Table 7: Training and test results (training result: b^0 & α^0; test result: $f(x)$)

Water (+1) versus Undeveloped (-1) ($b^0 = 2.3074$)		Developed (+1) versus Water (-1) ($b^0 = -7.1054e-15$)		Undeveloped (+1) versus Developed (-1) ($b^0 = 2.1741$)	
α^0	$f(x)$	α^0	$f(x)$	α^0	$f(x)$
0	1.0554	0.1429*1.0e-06	-1.9739	0	1.6661
0.2322*1.0e-07	-7.2163	0	2.4358	0.2843*1.0e-08	-1.2775
0.2322*1.0e-07	-7.0489	0	1.4966	0.2843*1.0e-08	1.6954
0	1.0018	0.9002*1.0e-06	-0.4052	0	-1.2283
	-7.0825		-0.2757		1.7061
	-7.0862		2.1642		1.6307
	1.0908		-1.2254		1.1327
	0.9062		-1.2160		-1.2432
	0.8577		-0.9749		1.6307
	-0.6363		0.6100		1.1327
	-7.2028		3.7322		-1.2432
	1.0005		-1.4550		1.6753
	-1.1863		1.3749		0.9623
	-1.2553		1.5435		0.9205
	-1.0338		0.8222		1.0050
	-0.7379		0.4043		1.1033
	-1.0377		0.5959		0.9787
	-1.0569		1.5756		0.9806
	-1.1998		0.7594		0.9257

From Table 7, scores with $f(x) > 0$ were coded 1, while scores with $f(x) < 0$ were coded 0. The results from the three binary classifiers are given in Tables 8, 9, and 10. These MATLAB codes were applied to the outcome of the three classifiers given in Tables 8, 9, and 10 to derive the result of the final classified satellite image given in Table 11:

```
WATER = (WATER_UNDEVELOPED==1) & DEVELOPED_WATER==0);
DEVELOPED = (DEVELOPED_WATER==1) & (UNDEVELOPED_DEVELOPED==0);
UNDEVELOPED = (UNDEVELOPED_DEVELOPED==1) & (WATER_UNDEVELOPED==0);

RESULT_1A1 = WATER + 3* DEVELOPED + 2* UNDEVELOPED
```

Only one cell remained unclassified; while eighteen cells were correctly classified. The resulting classified satellite image using the 1A1 SVM model (see Table 11) was compared with the actual ground truth data given in Table 1; the classification accuracy is therefore 18/19 = 94.74%. Using the cell-by-cell method of evaluation, the actual (ground truth data) and the predicted data (classified satellite image) were used to derive the confusion matrix given in Table 12. The classification overall accuracy (computed from Table 12) = Sum of diagonal elements ÷ Sum of all elements in the matrix. Therefore, overall accuracy = 24/25 = 96%.

Table 8: Result for water versus undeveloped (water = 1; undeveloped = 0)

1	1	1	1	0
1	1	1	0	0
0	0	1	0	0
0	0	0	0	0
0	0	0	0	0

Table 9: Result for undeveloped versus water (undeveloped = 1; water = 0)

0	0	0	0	1
0	0	0	1	1
1	0	0	1	1
1	1	1	1	1
1	1	1	1	1

Table 10: Result for undeveloped versus developed (undeveloped = 1; developed = 0)

1	1	1	1	1
1	1	1	1	1
0	0	1	1	1
0	0	1	1	1
0	0	0	1	1

Table 11: Final classified satellite image for 1A1 SVM (unclassified =0, water =1, undeveloped =2, and developed =3)

1	1	1	1	2
1	1	1	2	2
3	0	1	2	2
3	3	2	2	2
3	3	3	2	2

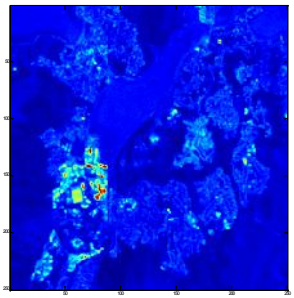
Table 12: Confusion matrix for 1A1 SVM result

	Reference data			
	Water	Undeveloped	Developed	Unclassified
Predicted data				
Water	8	0	0	0
Undeveloped	0	10	0	0
Developed	0	0	6	0
Unclassified	0	0	1	0

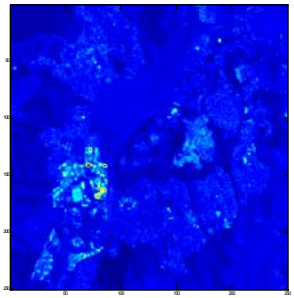
4. Empirical application: One-against-one SVM classification

satellite data were first reviewed in GIS (ArcGIS software); and all seven bands were extracted

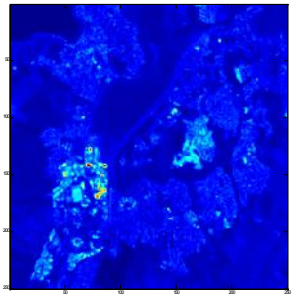
data must be in ASCII format for onward processing in MATLAB. In MATLAB the final study area was extracted from the original satellite image. Some regions of the satellite image were affected by cloud, which was why the final study area did not include those regions affected by cloud. All the seven bands were used for the classification experiment. The stratified random sampling was used to select the training data. The experiment was implemented with a polynomial kernel of degree *d = 20*, and penalty value *C = 100*. The resulting classified image was visualised in the GIS. The results of the one-against-one SVM experiment are given in Figure 4 and Table 13. The confusion matrix given in Table 13 was computed by comparing the result of the 1A1 SVM classification and the reference data given in Figure 4. Using the confusion matrix given in Table 13, the computed overall accuracy was 88.1%.



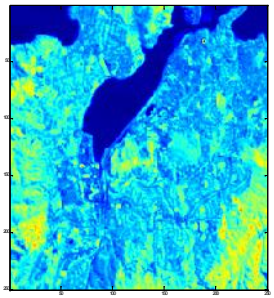
Band 1



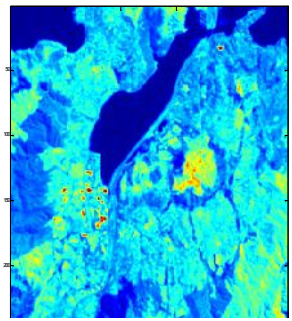
Band 2



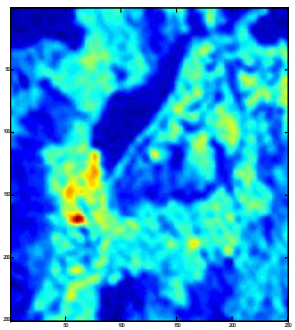
Band 3



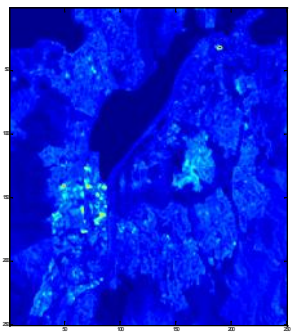
Band 4



Band 5



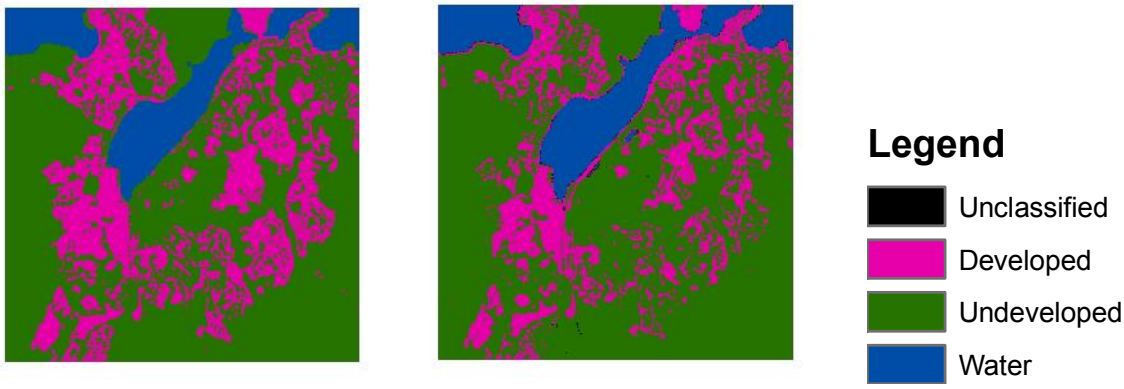
Band 6



Band 7

Satellite image

Figure 3 Extracted bands 1 - 7 of Landsat image of Porirua and original Landsat image of Porirua, New Zealand



Reference data 1A1 SVM

Figure 4 Classification result for 1A1 SVM

Table 13: Confusion matrix for 1A1 SVM classification (unclassified pixels = 256)

	Reference data			
	Developed	Undeveloped	Water	Unclassified
Predicted data				
Developed	11856	1472	1	0
Undeveloped	5307	36393	0	0
Water	95	308	6812	0
Unclassified	29	223	4	0

CONCLUSION

This study illustrated basically how the 1A1 SVM algorithm can be applied to the classification of satellite remote sensing data. The essence of first illustrating the experiment using simulated data was to help explain how the empirical experiment was implemented. In the simulated modelling, from Table 12, no water and undeveloped cell was wrongly predicted; while one developed cell was left unclassified. In the empirical modelling, from Table 13, 1472 and 1 undeveloped and water cells respectively were wrongly predicted as developed; 5307 developed cells were wrongly predicted as undeveloped; 95 and 308 developed and undeveloped cells pixels. This research has shown that the modification of binary classifiers like the support vector machine can help extend their use to solving multi-class problems; therefore binary classifiers could become veritable substitutes for conventional multi-class classifiers such as, K Nearest Neighbour (KNN) and Maximum Likelihood Classifier (MLC).

REFERENCES

Cortes, C. and Vapnik, V., 1995. Support vector networks. Machine learning, *20*(3): 273-297.

Fletcher, R. (Ed.), 1987. Practical methods of optimization. New York, NY: John Wiley & Sons.

Ivanciuc, O., 2007. Applications of support vector machines in Chemistry. In K. Lipkowitz, & T. Cundari (Eds.), Reviews in computational chemistry (pp. 291-400). Texas, TX: Wiley-VCH, John Wiley & sons, Inc.

Joachims, T., 1999. Making large-scale SVM learning practical. In B. SchÖlkopf, C. J. C. Burges, & A. J. Smola (Eds.), Advances in kernel methods - support vector learning (pp. 169-184). Cambridge, MA: MIT Press.

Melgani, F. and Bruzzone, L., 2004. Classification of Hyperspectral Remote Sensing Images with Support Vector Machines. IEEE Transactions On Geoscience And Remote Sensing, *42*(8):.

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.

Vapnik, V. N., 2000. The nature of statistical learning theory. New York, NY: Springer-Verlag.