CLASSIFICATION OF LIDAR DATA BASED ON MULTI-CLASS SVM

F. Samadzadegan [a], B. Bigdeli [a], P. Ramzi [a, *]

[a] Department of Geomatics Engineering, Faculty of Engineering, University of Tehran, Kargar-shomali Avenue, Tehran, Iran - (samadz,bigdeli,pramzi)@ut.ac.ir

Commission VI, WG VI/4

KEY WORDS: LIDAR data, Classification, Classifier Fusion, Multi-class SVM, Genetic Algorithm, Parameter Optimization

ABSTRACT:

LIght Detection And Ranging (LIDAR) is a powerful remote sensing technology in the acquisition of the terrain surface information for object classification and extraction. Major benefits of this technique are its high level of automation during data capturing and its spatial resolution. Because of high complexities and difficulties in urban areas, many researchers focus on the using of LIDAR data in such area. Consequently, one of the challenging issues about LIDAR data is classification of these data in urban area for identification of different objects such as building, road and tree. Several urban classification methods have been proposed for classification of LIDAR data. Support Vector Machines (SVM), one of the new techniques for pattern classification; have been widely used in many application areas such as remote sensing. SVM is a binary classification method but in some researches like remote sensing or pattern recognition, we need more than two classes. One solution for this difficulty is to split the problem into a set of binary classification before combining them. Multi-class SVM is one solution for solving mentioned problem. The one-against-one and the one-against-all are the two most popular strategies for Multi-class SVM. One problem that faces the user of an SVM is how to choose a kernel and the specific parameters for that kernel. Applications of an SVM therefore require a search for the optimum settings for a particular problem.
This paper proposes a classification technique, which we call the Genetic Algorithm Multi-Class SVM (GASVM), that uses genetic algorithm as a method for kernel's parameter optimization for one of the Multi-class SVM classifiers. We have used genetic algorithm for optimizing γ and C parameters of RBF kernel in Multi-class SVM. The classification's results of LIDAR data by use of this presented technique clearly demonstrate higher classification accuracy.

1. INTRODUCTION

Remotely sensed data has been widely used to land cover classification and object extraction (Wehr, Lohr, 1999; Haitao, 2008). Light Detection And Ranging (LIDAR) is one of the recent remote sensing technologies that is widely used for Digital Terrain Model (DTM) data collection and also for other studies including 3D extraction, urban management, atmospheric studies, and so on (Clode, 2004; Alharthy, Bethel, 2003). Comparing to other remote sensing data sources, LIDAR has its advantages such as acquisition of very dense data in a short period of time. LIDAR data contains plenty of scene information, from which most ground features such as roads, buildings and trees are discernible. More recently, advancements in LIDAR enabled the acquisition of dense point clouds. Major benefits of this technique are its high level of automation during data capturing and its spatial resolution. With point densities of up to several points per square meter, LIDAR data has become a valuable additional source for the reconstruction of different urban objects (Wehr, Lohr, 1999).
Classification of LIDAR data into objects such as building, tree and road in complex area is a challenging research topic in pattern recognition and remote sensing studies (Bartels, Wei, 2006; Brzank, Heipke, 2007). Several urban classification methods have been proposed for classification of LIDAR data (Kraus, Pfeifer, 1998; Zhang, 2003).
The Support Vector Machine (SVM) has emerged in recent years as a popular approach to the classification of data. (SVM) were first suggested by Vapnik (1995) and have recently been used in a range of problems including pattern recognition (Pontil and Verri, 1998), bioinformatics (Yu, Ostrouchov, Geist, & Samatova, 1999), and text categorization (Joachims, 2000). SVM by itself is a binary classification but in some researches like remote sensing or pattern recognition, we usually have more than two classes. Multi-class SVM is the solution for this problem which is has been utilized in some researches (Wetson, Watkins, 1998; Naotosi, 2007).
When using SVM, one problem is confronted: how to set the best kernel parameters. Proper parameters setting can improve the SVM classification accuracy. A GA-based regularization parameter can also be optimized using GAs in (Frohlich and Chapelle,2003).
The parameters that should be optimized include penalty parameter C and the kernel function parameters such as the γ for the radial basis function (RBF) kernel. Huang and Wang used Genetic algorithm as a method for parameter optimization of Support Vector Machine (Huang, Wang, 2006). The objective of this research is to simultaneously optimize the parameters and feature subset without degrading the SVM

* Corresponding author. This is useful to know for communication with the appropriate person in cases with more than one author.

classification accuracy. Proposed GA-based approach significantly improves the classification accuracy.
In this paper we proposed a GA-based method for optimization of RBF's kernel for one-against-one multi-class SVM. The key idea in this paper relies on optimization of C and γ parameter using genetic algorithm.

2. SUPPORT VECTOR MACHINE (SVM)

One of the state-of-the-art classification methods which has been widely used in different applications is Support Vector Machine (SVM). In this section we will briefly describe the basic SVM concepts for classification problems. These concepts can also be found in (Kecman, 2001; Scho¨lkopf and Smola, 2000; Cristianini and Shawe-Taylor, 2000).
Consider a set of training examples $D = \{(x_1, y_1), (x_2, y_2), ..., (x_N, y_N)\}$ where the ith sample n $x_i \in R^n$ (n is the dimension of the input space) belongs to two separate classes labeled by $y_i \in \{-1,1\}$ .The classification problem is to find a hyper-plane in a high dimensional feature space Z , which divides the set of examples in the feature space such that all the points with the same label are on the same side of the hyper-plane. SVM is to construct a map z=φ(x) from the input space R^n to a high-dimensional feature space Z and to find an "optimal" hyper-plane $w^T z + b = 0$ in Z such that the separation margin between the positive and negative examples is maximized. A decision function of the classifier is then given by

$$f_{w,b} = \operatorname{sgn}[w^T z + b] \tag{1}$$

where w is a weight vector and b is a threshold. Without loss of generality, we consider the case when the training set is not linearly separable. The SVM classification amounts to finding w and b satisfying

$$\begin{aligned} &\min \quad \frac{1}{2} w^T w + c\sum_{i=1}^{N} \varepsilon_i \\ &S.t \quad \begin{cases} y_i[w^T \varphi(x_i) + b] \geq 1 - \varepsilon_i, i = 1,.., N \\ \varepsilon_i \geq 0 \qquad \mathrm{i} = 1,..., \mathrm{N} \end{cases} \end{aligned} \tag{2}$$

Where c > 0 is a regularization parameter for the trade off between model complexity and training error, and ε_i measures the (absolute) difference between $w^T z + b$ and y_i . Solving (1) directly is more complex because of a number of variables and unknown $\varphi(x)$ (Cortes, Vapnik, 1995). Thus, solving (1) is converted into solving a dual problem

$$\begin{aligned} &\max -\frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N} y_i y_j (\varphi(x_i)^T \varphi(x_j))\alpha_i\alpha_j + \sum_{i=1}^{N}\alpha_i \\ &S.t\begin{cases} \sum_{i=1}^{N}\alpha_i y_i = 0 \\ 0 \leq \alpha_i \leq c, i = 1,..., N \end{cases} \end{aligned} \tag{3}$$

Let a kernel function $K(x, y)$ satisfying $k(x_i, x_j) = \varphi(x_i)^T \varphi(x_j)$. The above dual problem becomes

$$\begin{aligned} &\min \ \frac{1}{2}\sum_{i=1}^{N}\sum_{i=1}^{N} y_i y_j k(x_i, x_j)\alpha_i\alpha_j - \sum_{i=1}^{N}\alpha_i \\ &S.t \begin{cases} \sum_{\mathrm{i}=1}^{\mathrm{N}}\alpha_{\mathrm{i}} y_i = 0 \\ 0 \leq \alpha_{\mathrm{i}} \leq c, i = 1,..., N \end{cases} \end{aligned} \tag{4}$$

Moreover, the decision function of the classifier can be represented as

$$f(x) = \operatorname{sgn}[\sum_{i=1}^{N}\alpha_i y_i k(x_i, x) + b] \tag{5}$$

For convenient computation here, let $a_i = \alpha_i y_i$. Then (3) can be equivalently written as

$$\begin{aligned} &\min \ \frac{1}{2}\sum_{i=1}^{N}\sum_{i=1}^{N} a_i a_j k(x_i, x_j) - \sum_{i=1}^{N} a_i y_i \\ &S.t = \begin{cases} \sum_{i=1}^{N} a_i \qquad , i = 1,..., N \\ -c_i^1 \leq a_i \leq c_i^2 \end{cases} \end{aligned} \tag{6}$$

Where for i= 1,...,l , $c_i^1 = -c(\operatorname{sgn}(1 - y_i)), c_i^2 = c(\operatorname{sgn}(1 + y_i))$. Therefore, the learning problem in SVM is equivalent to the quadratic programming problem in (5) with N bounded variables.
One key aspect of the SVM model is that the data enters the above expressions (3 and 4) only in the form of the dot product of pairs. This leads to the resolution of the second problem mentioned above, namely that of non-linearly separable data. The basic idea with SVMs is to map the training data into a higher dimensional feature space via some mapping $\varphi(x)$ and construct a separating hyperplane with maximum margin there. This yields a non-linear decision boundary in the original input space.
Typical choices for kernels are:

- Polynomial $k(x, z) = (1 + \langle x.\mathrm{z}\rangle)^{\mathrm{d}}$ (7)
- RBF $k(x, z) = \exp(\frac{-\|x - z\|^2}{2\sigma^2})$ (8)
- Sigmoid $k(x, z) = \tanh(\langle x, \mathrm{z}\rangle - \theta)$ (9)

2.1 MULTI-CLASS SVM

SVMs are an example of a linear two-class classifier and it can only take two values: ±1 . For a remote sensing application, several classes are usually of interest. One solution for this difficulty is to split the problem into a set of binary classification before combining them (Hsu, Lin, 2001). The one against- one and the one-against-all are the two most popular strategies in this category.
One-against-one is the method that calculates each possible pair of classes of a binary classifier. Each classifier is trained on a subset of training examples of the two involved classes. In this

method, all N (N-1)/2 binary classifications are combined to estimate the final output. Final output is then created by a majority voting scheme. This approach is suitable for problem with large amount of data (Hsu, Lin, 2001).

The most important problem caused by this method is the existence of unclassifiable regions which is able to be solved using one-against-all technique. For an N-class problem, the one-against-all method constructs N SVM models (one SVM per class), which is trained to distinguish the samples of one class from samples of all remaining classes. In this method, the ith SVM is trained using all the learning examples in the ith class with positive labels and the others with negative labels and finally, N hyperplanes are obtained.

3. PARAMETER OPTIMIZATION

One of the most important design choices for SVMs is the kernel-parameter, which implicitly defines the structure of the high dimensional feature space where a maximal margin hyperplane will be found. Too rich a feature space would cause the system to overfit the data, and conversely the system might not be capable of separating the data if the kernels are too poor. Support vector classification with Gaussian RBF kernel is sensitive to the γ parameter. How to select γ of RBF kernel in SVM literature has been discussed in (Cristianini, 1998; Chapelle, Vapnik, 2002; Huang, suang, 2006).

Usually, practitioners select these parameters empirically by trying a finite number of values and keeping those that provide the least test error. However, for a large number of parameters, this approach is not feasible.

3.1 GENETIC ALGORITHM

A genetic algorithm (GA) is a search technique used in computing to find exact or approximate solutions to optimization and search problems. GA work with a set of candidate solutions called a population. Based on the Darwinian principle of 'survival of the fittest', the GA obtains the optimal solution after a series of iterative computations. GA generates successive populations of alternate solutions that are represented by a chromosome, i.e. a solution to the problem, until acceptable results are obtained. Associated with the characteristics of exploitation and exploration search, GA can deal with large search spaces efficiently, and hence has less chance to get local optimal solution than other algorithms.

A fitness function assesses the quality of a solution in the evaluation step. The crossover and mutation functions are the main operators that randomly impact the fitness value. Chromosomes are selected for reproduction by evaluating the fitness value. The fitter chromosomes have higher probability to be selected into the recombination pool using the roulette wheel or the tournament selection methods (Tang, et al, 1996).

The evolutionary process operates many generations until termination condition satisfy.

4. THE PROPOSED METHOD

In our proposed methodology, we have used one-against-one multi-class SVM with RBF kernel. To implement our proposed approach, this research used the RBF kernel function for the SVM classifier because the RBF kernel function can analysis higher-dimensional data and requires that only two parameters, C and γ be defined. We used genetic algorithm as optimization method for selecting the best parameter of RBF kernel. Fig 2, illustrates the general structure of proposed methodology which contains three main steps.

a. Feature Extraction

The first step in every classification process is to extract proper features from data set. These features must contain useful information to discriminate between different regions of the surface. In our experiment, we have used different features extracted from two types of LIDAR data (Range data and Intensity data). All types of features used in this research are introduced in Table 1.

Table 1. Features vector for classification

Name	Formulation
First Pulse Intensity	FPI
Last Pulse Intensity	LPI
First Pulse Range	FPR
Last Pulse Range	LPR
NDDI	$\frac{FPR - LPR}{FPR + LPR}$
Opening	$A \circ B = (A \Theta B) \oplus B$
Mean	$Mean\_i = \sum_{i,j=0}^{N-1} i \times P(i,j)$
Entropy	$\sum_{i,j=0}^{N-1} P_{i,j} \times (-\ln P_{i,j})$
Standard Deviation	$\mathrm{var}\, iance\_i = \sum_{i,j=0}^{N-1} P(i,j) \times (i - Mean\_i)^2$
Homogeneity	$\sum_{i,j=0}^{N-1} \frac{P_{i,j}}{1 + (i+j)^2}$

b.Parameter Optimization Using GA.

The chromosome design, fitness function, and system architecture for the proposed GA-based parameter optimization are described as follows.

Chromosome design. When the RBF kernel is selected, the parameters (C and γ) used as input attributes must be optimized using our proposed GA-based system. Therefore, the chromosome comprises two parts, C, γ. However, these chromosomes have different parameters when other types of kernel functions are selected. The binary coding system was used to represent the chromosome. Fig. 1 shows the binary chromosome representation of our design.

g_c^1	$\cdots g_c^i \cdots$	$g_c^{n_c}$	$g_\sigma^1 \cdots g_\sigma^i \cdots$	$g_\sigma^{n_\sigma}$

Figure 1. The choromosom representation

In Fig. 1 $g_C^1 \sim g_C^{n_C}$ represents the value of parameter C, $g_\gamma^1 \sim g_\gamma^{n_\gamma}$ represents the parameter value γ. n_C is the number of bits representing parameter C, n_γ is the number of bits representing parameter σ. we can choose n_C and n_γ according to the calculation precision required, and the minimum and maximum value of the parameter is determined by the user (Huang, Wang, 2006).

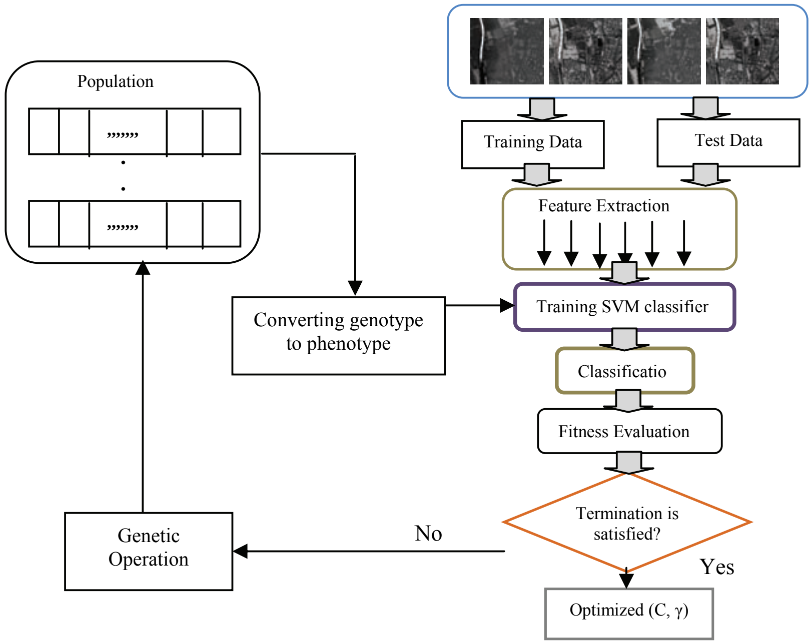


Figure 2. GA-based proposed method

In Fig.1 the bit strings representing the genotype of parameter C and γ should be transformed into phenotype by this equation:

$$P = \min_P + \frac{\max_P - \min_P}{2^l - 1} \times d \qquad (10)$$

P Phenotype of bit string
$\min_p$ Minimum value of the parameter
$\max_p$ Maximum value of the parameter
D Decimal value of bit string
L Length of bit string

Fitness Function. We used overall classification accuracy as fitness function. We have used error matrices of classification results as main evaluation method of interpretation the quality of each classification method. Each column in this matrix indicates the instances in a predicted class and each row represents the instances in an actual class. All the diagonal variants refer to the correct interpreted numbers of different classes found in reality. Overall accuracy yields one number of the whole error matrix. It's the sum of correctly classified samples divided by the total sample number from user set and reference set

$$OA = \frac{\sum_{i=1}^{k} N_{i,i}}{\left[\sum_{i=1}^{k} N_{.i} + \sum_{i=1}^{k} N_{i.}\right]} * 100\% \qquad (11)$$

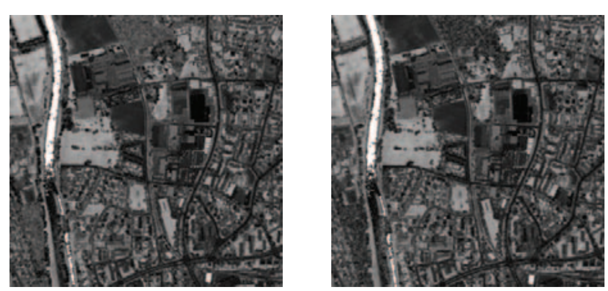
Where
$N_{i,j}$: (i, j)th entry in confusion matrix
$N_{i..}$: the sum of all columns for row i
$N_{.j}$: the sum of all rows for column i

For each chromosome representing C and σ training dataset is used to train the SVM classifier, while the testing dataset is used to calculate classification accuracy. When the classification accuracy is obtained, each chromosome is evaluated by fitness function.

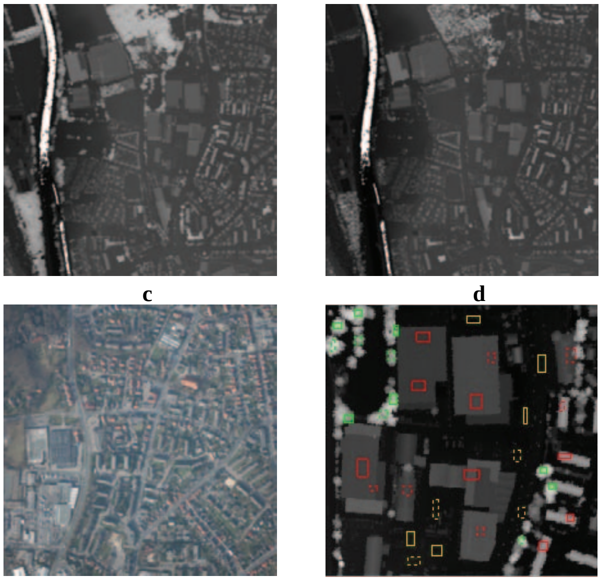
5. EXPERIMENT AND RESULT

5.1 Data Set

As mentioned above, a subset of LIDAR remote sensing data with four popular bands, first pulse intensity, last pulse intensity, first pulse range and last pulse range is classified by our proposed method based on SVMs. This sample of LIDAR data is an urban area recorded from city of Castrop-Rauxel which is located in the west of Germany. This dataset has enough complexity in urban area for evaluating our proposed method. The LIDAR data is classified into three main classes: building, tree and ground. Table.2 shows number of training and test samples selected for each class.



a **b**



c d

e f

Figure 3. Data set consist of a) First pulse intensity, b) Last pulse intensity, c) First pulse range, d) Last Pulse range, e) Aerial Image, f) Train and test data from selected area for Tree (green), Building (red) and Ground (yellow). Dashed lines show test data and continuous ones demonstrate training data.

Table 2. Information of training and test sample of each class

Class	Number of training samples	Number of test samples
Tree	510	420
Building	1426	672
Ground	672	564

5.2 Experiment and results

To assess the capabilities of proposed method, we applied this method for classification of LIDAR data. Table 3 shows the results of optimization for C and γ.

Table 3. Results of GA based optimization

Optimized C	Optimized γ	Overall classification Accuracy (%)
94.432125	0.043678	OA=90.1283

Table 4. Confusion matrix of optimized SVM

		Reference Data		
		Building	Tree	Ground
Optimized SVM	Building	1124	48	13
	Tree	294	829	7
	Ground	75	9	2119
	OA=90.1283			

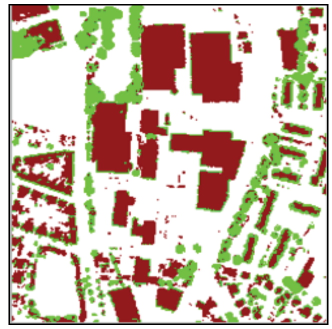
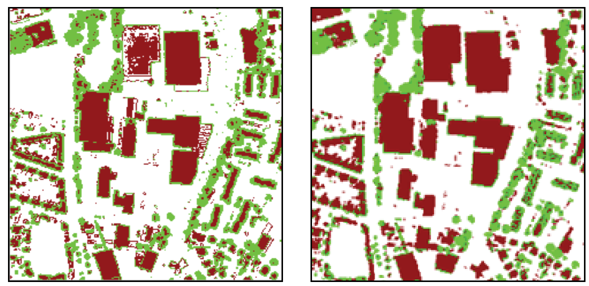


Figure 4. Result of optimized RBF SVM

Table 4 and figure 3 show results of classification using optimized C and γ. These results shows that proposed method produced high overall classification accuracy.
In this step, we compare results of proposed method by other classifiers such as minimum distance and maximum likelihood.



a b

Figure 5. Result of a)minimum distance and b)maximum likelihood

Table 5. Result of confusion matrix for minimum distance and maximum likelihood classifiers

		Reference Data		
		Building	Tree	Ground
Minimum Distance	Building	1075	81	16
	Tree	343	783	9
	Ground	76	22	2114
	OA= 85.9150			
likelihood Maximum	Building	1061	81	11
	Tree	326	771	2120
	Ground	106	34	2120
	OA= 87.4772			

Comparison of proposed method with another classifiers showed that this method improved overall accuracy in classification process. These results demonstrate that optimized SVM produced better classification results than Minimum Distance and Maximum Likelihood classifiers.

6. CONCLUSION

In this paper we have proposed optimization of RBF Multi-class SVM by use of genetic algorithm for classification of LIDAR data in an urban area. We have extracted some standard features from this dataset. In this paper we have used genetic algorithm based method for optimizations of two essential parameters of RBF SVM classifier contain C and γ.

Our GA based optimization method used overall classification accuracy as fitness function. We choose the best values for C and γ and used these parameters for classification of LIDAR data. Based on the mentioned results, optimized SVM produced better classification accuracy than other classifiers such as minimum distance and maximum likelihood.

7. REFRENCES

Wehr, A., Lohr, U., 1999. Airborne Laser Scanning – An Introduction and Overview, ISPRS Journal of Photogrammetry and Remote Sensing, V.54, pp 68-82.

Haitao, L., Haiyan, G.U., Yanshun, H., Jinghui, Y. 2008, Fusion of High-Resolution Aerial Emagery and LIDAR Data for Object-Oriented Urban Land-Cover Classification Based on SVM, ISPRS Workshop on Updating Geo-spatial Databases with Imagery & The 5th ISPRS Workshop on DMGISs.

Clode, S., Kootsookos. P, Rottensteiner.F,. 2004. The Automatic Extraction of Roads from LIDAR Data. In IAPRSIS, Vol. XXXV-B3, pp. 231 – 236.

Alharthy, A. Bethel, J., 2003. Automated Road Extraction From LIDAR Data. In: Proceedings of ASPRS, Anchorage, Alaska, unpaginated, CD-ROM.

Bartels, M., Wei, H. 2006, Rule-based Improvement of Maximum Likelihood Classified LIDAR Data fused with co-registerd band, Computational Vision Group, School of Systems Engineering the University of Reading.

Brzank, A., Heipke, C. 2007, Classification of LIDAR into Water and Land in Coastal Areas. Institute of Photogrammetry and Geoinformation.

Kraus, K., Pfeifer, N. 1998, Determination of terrain models in wooded areas with airborne laser scanner data. ISPRS Jourmal of Photogrammetry and Remote Sensing, 53:193– 203.

Zhang, K., Cheng, S., Dean Whitman, Mei-Ling Shyu, Jianhua Yan, and Chengcui Zhang. 2003, A progressive morphological filter for removing non-ground measurements from airborne LIDAR data. IEEE Transactions on Geoscience and Remote Sensing, 41, No. 4, April 2003:872–882.

Pontil, M., & Verri, A. 1998 . Support vector machines for 3D object recognition. IEEE Transactions on Pattern Analysis and Machine Intelligence, 20(6), 637–646.

Yu, G. X., Ostrouchov, G., Geist, A., & Samatova, N. F. 2003. An SVMbased algorithm for identification of photosynthesis-specific genome features. Second IEEE computer society bioinformatics conference, CA, USA pp. 235–243.

Joachims, T., 2000. Estimating the Generalization Performance of an SVM Efficiently, Proc. of the 17th Intern. Conf. on Machine Learning, Stanford, CA, USA.

Weston, J and Watkins, C., 1998. Multi-class support vector machines, Technical report CSD-TR-98-04.

Naotosi, S,. 2007. A comparison of Multi-calss Support Vector Machine Methods for Face Recognition. Department of Electrical and Computer Engineering, The University of Maryland.

Fro¨hlich, H., and Chapelle, O., 2003 . Feature selection for support vector machines by means of genetic algorithms. Proceedings of the 15th IEEE international conference on tools with artificial intelligence, Sacramento, CA, USA pp. 142–148.

Huang, C., Wang, C,. 2006. A GA based Feature Selection and Parameter Optimization for Support Vector Machine, Expert Systems with Application 31, 231-240.

Kecman, V. 2001. Learning and soft computing. Cambridge, MA: The MIT Press.

Scho˝lkopf, B., & Smola, A. J. 2000 . Statistical learning and kernel methods. Cambridge, MA: MIT Press.

Cristianini, N., and Shawe-Taylor, J. 2000. An introduction to support vector machines. Cambridge: Cambridge University Press.

Chapelle, O., Vapnik, V., Bousquet, O., and Mukherjee, S., 2002. Choosing Multiple Parameters for Support Vector Machines, Machine Learning, vol. 46, pp. 131-159,

Cortes. C and Vapnik. V., 1995. Support vector networks, Machine Learning, vol. 20, no. 3, pp. 273-297.

Hsu, C., and Lin, C., 2001. A comparison of methods for multi-class support vector machines, Tech- nical report, Department of Computer Science and Information Engineering, National Taiwan University.

Cristianini, N.,. Shawe-Tayor, J.,Campbell, C., 1998. Dynamically adapting kernels in support vector machines," In M. Kearns, S. Solla, D. Cohn, eds, Adances in Neural Information Processing Systems, 11, Denver, CO, MIT Press.

Vapnik, V,. The Nature of Statistical Learning Theory. New York: Springer, 1995.

Tang, K. S., Man, K. F, Kwong, S and He, W,. 1996. Genetic algorithms and their applications, IEEE Signal Processing Magazine, vol. 13, pp. 22-37, Nov.