Urban Land Cover Data Classification Prediction Using Multiclass RBF-SVM Model

Jiang Zhiyuan[1], *Huang Qingyi*[1], *Zhu Huiying*[1]

[1]Institute of System and Science, National University of Singapore, Singapore 119615

ABSTRACT

Urban land cover items prediction, based on the remote sensors' spatial and geometrical information, is becoming a prevailing area especially in urban monitoring, administration and programming. As a multi-classification task, a wide range of methods for urban land cover prediction continues to be proposed and assessed. In this paper, we review the SVMs (Support Vector Machines), a promising machine learning methodology with the introduction of multiclass mechanism and baseline approach. Furthermore, the complete procedure of approach is described aimed at modification of the baseline approach. The proposed approach is presented to optimize the RBF-SVM model, which mainly utilizes the tuning key parameters *C* and *γ* by grid search with cross validation to make a soft-margin hyperplane. Others including data scaling, data cleaning and unbalanced input adaption are also concerned to enhance the performance. The final model outperforms the baseline with the highest accuracy reaching nearly 87%. The RBF-SVM and related methodology may facilitate the further research regarding to remote spatial land cover prediction.

1. INTRODUCTION

Urban land cover classification is one of the widest used applications in the field of remote sensing. The detailed knowledge of land cover is an important input variable for several urban environmental monitoring, especially in some applications like the study of urban sprawl, urban development plan, population or architecture density and monitoring of urban growth[1][2]. Such target is also corresponding to the 3 strategic areas of AISingapore (healthcare, urban and fintech) in 2018.

The overall objective of the architecture recognition is to automatically categorize all pixels in an image into land cover classes or themes. Accordingly, the classification algorithms are important for the success of urban land cover classification process. Furthermore, how to improve the classification accuracy is the major challenge which effects the result. A large range of classification algorithms have been developed and applied for classifying data, e.g. SVMs[3], Random Forests[4], KNNs[5], DTs[6] and ANN[7]. However, these articles seldom apply in the area of urban cover land classification or lack the complete algorithm parameters optimization aimed at enhancing the accuracy.

In this page, SVM (Support Vector Machine) method[8] is mainly used to work as a classifier in urban land cover research. On this basis, procedures of data preparation and preprocessing are described and conducted completely. Thus, cross validation with grid search is used to tuning the significant parameters of RBF-SVMs (select the RBF as the kernel function of SVM), along with other aspects optimization. Finally, results and limits are displayed and discussed. Moreover, the dataset we used was obtained from the University of California, Irvine (UCI) Machine Learning Depository[9].

Fig. 1. Classified map of the study area, produced by classifying scale 20 segments with super-object variables from scale 40, 60, and 80 segments.

2. BASELINE APPROACH

2.1. SVMs (Support Vector Machines)

In paper[10], the author gave the brief definition of SVMs. Given a training set of instance-label pairs (x_i, y_i) and $y \in \{1, -1\}^l$, the support vector machines (SVM) require the solution of the following optimization problem:

$$\min_{w,b,\xi} \frac{1}{2} w^T w + C \sum_{i=1}^{l} \xi_i \quad (1)$$

$$subject\ to \quad y_i(w^T \phi(x_i) + b) \geq 1 - \xi_i,\ \xi_i \geq 0 \quad (2)$$

Here training vectors xi are mapped into a higher (maybe infinite) dimensional space by the function ϕ. SVM finds a linear separating hyperplane with the maximal margin in this higher dimensional space. $C > 0$ is the penalty parameter of the error term.

Furthermore, $K(x_i, x_j) = \phi(x_i)^T \phi(x_j)$ is called the kernel function. Three most common used basic kernel functions show as below, in this research, RBF was chosen:

- $linear$: $K(x_i, x_j) = X_i^T x_j$
- $polynomial$: $K(x_i, x_j) = (\gamma x_i^T x_j + \gamma)^d, \gamma > 0$
- RBF: $K(x_i, x_j) = exp(-\gamma \parallel x_i - x_j \parallel^2), \gamma > 0$

2.2 Multiclass SVMs

Traditional SVMs are used to handle the binary classification problem, in order to tackle the multiclass issues, two primary methodologies are introduced as OvO (One versus One) and OvA (One versus All)[11].

OvA (One versus All)

OvA strategy involves training a single classifier per class, with the samples of that class as positive samples and all other samples as negatives. This strategy requires the base classifiers to produce a real-valued confidence score for its decision, rather than just a class label; discrete class labels alone can lead to ambiguities, where multiple classes are predicted for a single sample.

OvO (One versus One)

In OvO reduction, one training $K(K-1)/2K(K-1)/2$ binary classifiers for a K-way multiclass problem; each receives the samples of a pair of classes from the original training set and must learn to distinguish these two classes. At prediction time, a voting scheme is applied the equation follows: all $K(K-1)/2K(K-1)/2$ classifiers are applied to an unseen sample and the class that got the highest number of "+1" predictions gets predicted by the combined classifier.

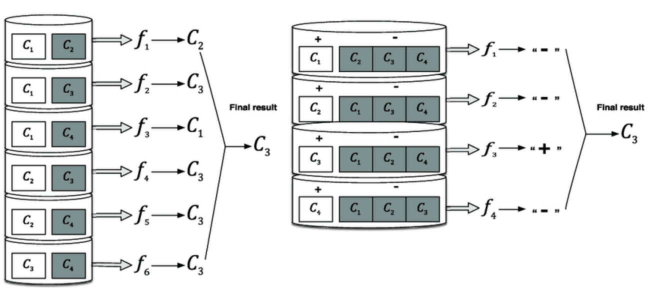


Fig. 2. The depiction of OvO and OvA.

2.3 Data exploration

In our research, the urban land cover dataset was obtained from the University of California, Irvine (UCI) Machine Learning Depository. This high spatial urban landcover dataset of Deerfield Beach, Florida, was created using geographic object-based image analysis methods, with 9 classifications labeled as: **trees, grass, soil, concrete, asphalt, buildings, cars, pools, and shadows.**

Data separation

The whole dataset contains 675 samples, which are already divided into 507 (75%) training samples and 147 (25%) test samples. In addition, each row has 147 variables of 7 coarser scales with 21 independent variables, which represent spectral, size, shape, and texture information. The details of the variables record in appendix. Accordingly, our objective is to build a model based on the training dataset, and apply to predict on the test dataset.

Data features selection

The dataset contains 1 class that means class target and 147 variables, obviously the variables dimensions are too huge. Since the 147 variables belong to 7 sets with the equivalent 21 variables, the only difference is the sensor image coarser. Thus, the first 21 variables are picked out as the variables, remaining parts work as the comparative datasets to backup. The same processes are applied to both training dataset and test dataset, so as to get the training samples with the shape of 507×21×1 and test samples with the shape of 147×21×1.

Target dummy code

In order to specify the targets better, the target Y is defined as the dummy codes of 9 different urban items from 1 to 9. Hence, the training target distributions, correlation metrics and variables distributions could be got as blow. In addition, there is no NAs or Null data inspected in the dataset, thus it's no need to handle the missing values.

2.4 Baseline multiclass experiment

Baseline RBF-SVMs steps:

1. Transform data to the format of an SVM package.
2. Try the RBF kernel to build model based on training samples.
3. Test.

After the basic features selection and dataset split, we used the *scikit-learn library* "*SVC*" to work as the SVM classifier, set the kernel to be "*RBF*" while other parameters are all default. The result shows the accuracy of training dataset reached 1.00, however, the test dataset accuracy is only 0.15 using the trained model. As such, the model is identified over-fitting. And the multiclass methods show no difference by using *'OvO'* or *'OvA'*. The performance of baseline model is shown in section 4.

3. PROPOSED APPROACH

The baseline model based on the training dataset, shows weak performance on test dataset, the root cause is that the specified hyperplanes confirmed by support vectors worked as hard margins. In addition, training dataset shows quite tremendous range gap between each column, and target labels quantities are imbalanced. Further process and optimization are necessary to be implemented in our proposed approaches.

As such, the proper procedures are proposed step by step as follows:

1. Scaling on training samples, apply the same scaler on test samples.
2. Implement C and γ to build soft margins for RBF kernel.
3. Split the dataset into training, validation and test dataset.
4. Apply grid search with cross validation to confirm C and γ.
5. Take unbalanced class numbers and excluding outliers into consideration.
6. Use the best parameters C and γ to training the whole training dataset.
7. Test on test samples.
8. Analyze the results.

3.1. Scaling the dataset

Scaling before applying SVM is very important. The main advantage of scaling is to avoid attributes in greater numeric ranges dominating those in smaller numeric ranges. Another advantage is to avoid numerical difficulties during the calculation. Because kernel values usually depend on the inner products of feature vectors, which are recommend linearly.

In our dataset, the variables numerical range difference showed huge. For instance, variable "*GLCM3*" occupied the range from 874 to 6341, meanwhile variable "*RECT*" just took 0.24 to 1. Such difference led to "*GLCM3*" domination across the training dataset. Accordingly, we built the linear scaler to convert all the variables into [-1, 1], the same scaler would be applied to the test dataset.

3.2 Cross validation on training dataset

Cross-validation is a model validation technique for assessing how the results of a statistical analysis will generalize to an independent dataset. It is mainly used in settings where the goal is prediction, and one wants to estimate how accurately a predictive model will perform in practice. For the previous baseline experiment, the lack of the validation couldn't guarantee the model's generalization to all samples. The goal of cross validation is to define a dataset to "test" the model in the training phase, in order to limit problems like overfitting.

In our proposed approach, the ***k-fold*** cross-validation were used to randomly partition the training dataset into k equal sized subsets. Throughout the k subsets, a single subsample is retained as the validation data for testing the model, and the k-1 subsets are used to training the model. The cross-validation process is then repeated k times and calculate the average score from k results. k was identified to be 3 in our experiment, thus the validation samples quantity is nearly equal to test samples.

In experiment, from "*sklearn.model_selection*" library, the "*cross_val_score*" package was imported to realize the cross validation.

3.3 Grid search to confirm *C* and *γ* of RBF kernel

Since the RBF kernel is a reasonable first choice, which could handle the case when the relation between class labels and attributes is nonlinear. There are two parameters for an RBF kernel C and γ to find out the soft margin between each classification. The goal is to identify good (C, γ) so that the classifier can enhance the prediction accuracy.

According to Lin's recommendation, the "grid-search" is usually used combined with the cross-validation in our case. Two major steps were followed: Firstly, tried exponentially growing sequences of C and γ to identify the coarse range for these 2 parameters. For C the approximate parameters were set to be [1, 10, 100, 1000], and γ was set to be [1e-1, 1e-2, 1e-3, 1e-4, 1e-5]. Secondly, a finer grid search on the neighborhood of first step ranges about C and γ. In doing this, confirm the C and γ to define the SVMs based on the soft margins.

In experiment, from "*sklearn.model_selection*" library, "*GridSearchCV*" package was used to try the grid search. In the first step, the best choice for (C, γ) to be (10, 0.01). After that, the further procedures to refine the best (C, γ) as narrow range C with range [10, 20] *step* = 1 and γ with range [0.001, 0.1] step = 0.001, the final best parameters for (C, γ) is (15, 0.15) with the highest validation accuracy 0.7968.

3.4 Optimization by balancing class weight

Since the training dataset class magnitudes were not balanced (the largest number of "*building*" label is 97, but the smallest number of "*pool*" label is 14). Thus, the weights of each class were taken into consideration to reduce the effect of the imbalance.

In experiment, from "*sklearn.model_selection*" library, in "*GridSearchCV*" the "*class_weight*" was set to be balance.

3.5 Optimization by eliminating outliers

Since the training dataset may contained several outlier samples, SVMs would be influenced by the outliers so as to change the margin between two classifications (SVMs outlier sensitivity shown in Fig. 3). Therefore, outlier elimination may be valuable to build a more appropriate model and enhance the prediction accuracy.

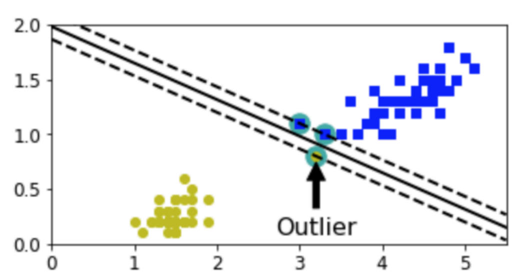


Fig. 3. SVM is sensitive to the outliers

In the experiment, exclude the outliers according to the boxplot of scaled training samples. The boxplot of training samples is shown in Fig. 4. The samples which contain the values beyond 6 were concerned as the outliers, and 7 records removed from the training dataset. Thus, the training dataset size reduced from 507 to 500. The modified training dataset would also process by the 8 steps described in section 3 and compared with the results of complete training dataset.

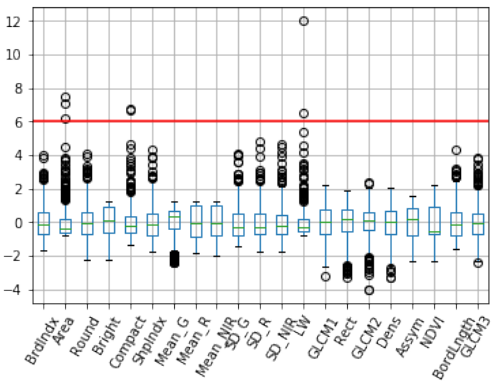


Fig. 4. Training samples boxplot of all variables, the red line indicates the upper boundary criteria to define the outliers

4. EXPERIMENTAL RESULTS

All the 507 scaled training samples we used to training the SVM models with 3 folds cross validation, the major tuning method of Grid-Search to confirm the parameters (C, γ) example is implemented in the Table 1.

Table 1. The grid search table for RBF-SVM

mean_score	score_std	param_C	param_gamma
0.744	0.00317	1	0.1
0.722	0.00778	1	0.01
0.416	0.03128	1	0.001
0.188	0.00187	1	0.0001
0.188	0.00187	1	1.00e-05
0.718	0.013	10	0.1
0.784	**0.00123**	**10**	**0.01**
0.722	0.01736	10	0.001
0.424	0.03426	10	0.0001
0.188	0.00187	10	1.00e-05
0.726	0.03297	100	0.1
0.754	0.02088	100	0.01
0.774	0.00809	100	0.001
0.722	0.01736	100	0.0001
0.426	0.03441	100	1.00e-05

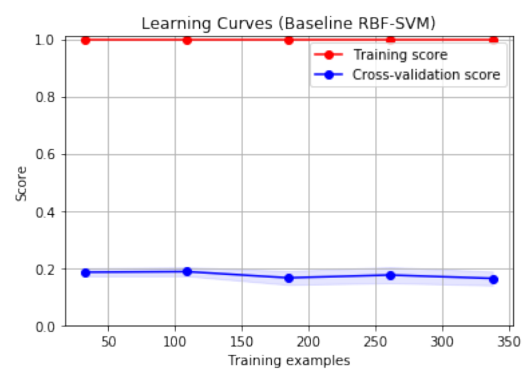
Adopt the tuning steps from coarse to refine described in section 3.3. On this basis, different (C, γ) parameters set are calculated of different RBF-SVM models. The different eventual best parameters set corresponding to each models we listed in Table 2.

Table 2. Each RBF-SVM proposed models (A-D) best parameters with the training accuracy, validation accuracy and test accuracy compared with the baseline model

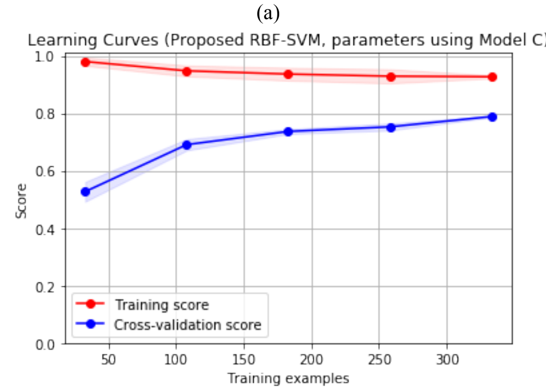
RBF-SVM Model	C	γ	Training accuracy	Valid accuracy	Test accuracy
Baseline	1	auto	1	-	0.155
Proposed A Imbalance With outlier	23	0.010	0.905	0.795	0.863
Proposed B Balanced With oulier	10	0.011	0.864	0.785	0.863
Proposed C Imbalance Without oulier	29	0.010	**0.916**	0.798	**0.869**
Proposed D Balanced Without oulier	22	0.010	0.910	0.784	0.839

From Table 2. Elimination the outliers of the training could increase both training accuracy and test accuracy, which indicates the outliers effect the soft margin hyperplane identification, and optimize the model performance from the comparison of Proposed model A and C. Another discovery is balancing the input gap deems decrease the model's performance from the comparison of Proposed model (A, B) and model (C, D).

Using the best performance Proposed model C, the learning curve is generated compared with the baseline model shown in Fig. 5. As such, prediction confusion metrics graph is generated in Fig. 6.

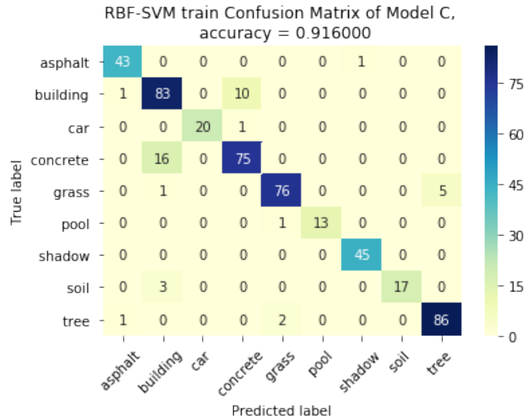


(a)

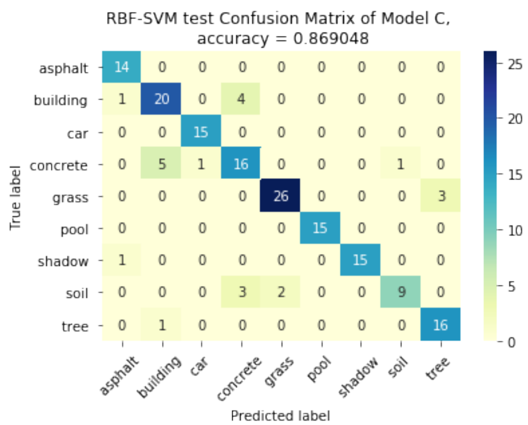


(b)

Fig. 5. (a) The baseline SVM model learning curve, (b) The proposed RBF-SVM model C's learning curve of training and validation



(a)



(b)

Fig. 6. (a) Proposed RBF-SVM model C training confusion metrics. (b) Confusion metrics of RBF-SVM model C applied in test result

From Fig. 6. the confusion metrics illustrates most samples prediction results are correct, accuracy reaches over 85%. The misclassification occurs in two class, respectively (*concrete, building*) and (*grass, tree*). These two sets of sample own the similar shape and RGB colors, which means model could be optimized to further extract more features of similar remote land cover items.

5. CONCLUSIONS

SVMs theories and methodologies were classical with high-performance applied into many domains. In this research, multiclass SVMs model with RBF kernel were traininged, validated and tested based on the urban land cover data, and applied to predict the urban landcover targets classifications. By modifying from the baseline approach, we proposed our RBF-SVMs models with several optimizations: scaling, tuning parameters by grid search, cross validation, balancing input class weight and exclusion of outliers. In the experiments, several models performances with different configuration or parameters were also compared with each other. The final prediction accuracy enhanced a lot from 15.48% (baseline SVMs model) to 86.90% (proposed RBF-SVMs model). Such results, methods and research procedures are valuable and useful in such similar areas of urban remote detection and telemetering.

REFERENCES

[1] L. Taha, "Assessment of urban land cover classification using Wishart and Support Vector Machine (SVM) based on different decomposition parameters of fully-polarimetric SAR," vol. 11, May 2017.

[2] L. Zhang, L. Sun, B. Zou, and W. M. Moon, "Fully polarimetric SAR image classification via sparse representation and polarimetric features," *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.*, vol. 8, no. 8, pp. 3923–3932, 2015.

[3] G. Mountrakis, J. Im, and C. Ogole, "Support vector machines in remote sensing: A review," *ISPRS J. Photogramm. Remote Sens.*, vol. 66, no. 3, pp. 247–259, 2011.

[4] M. Belgiu and L. Drăguţ, "Random forest in remote sensing: A review of applications and future directions," *ISPRS J. Photogramm. Remote Sens.*, vol. 114, pp. 24–31, 2016.

[5] Q. Meng, C. J. Cieszewski, M. Madden, and B. E. Borders, "K nearest neighbor method for forest inventory using remote sensing data," *GIScience Remote Sens.*, vol. 44, no. 2, pp. 149–165, 2007.

[6] M. A. Friedl and C. E. Brodley, "Decision tree classification of land cover from remotely sensed data," *Remote Sens. Environ.*, vol. 61, no. 3, pp. 399–409, 1997.

[7] A. E. Maxwell, T. A. Warner, and F. Fang, "Implementation of machine-learning classification in remote sensing: an applied review," *Int. J. Remote Sens.*, vol. 39, no. 9, pp. 2784–2817, 2018.

[8] C.-C. Chang and C.-J. Lin, "LIBSVM: A library for support vector machines," *ACM Trans. Intell. Syst. Technol.*, vol. 2, no. 3, pp. 1–27, Apr. 2011.

[9] B. Johnson and Z. Xie, "Classifying a high resolution image of an urban area using super-object information," *ISPRS J. Photogramm. Remote Sens.*, vol. 83, pp. 40–49, 2013.

[10] C.-W. Hsu, C.-C. Chang, and C.-J. Lin, "A practical guide to support vector classification," 2003.

[11] K.-M. Jung, "Support vector machines for unbalanced multicategory classification," *Math. Probl. Eng.*, vol. 2015, 2015.