Comparison of classification algorithms in the task of object recognition on radar images of the MSTAR base

A.A. Borodinov[1], V.V. Myasnikov[1,2]

[1]Samara National Research University, 34 Moskovskoe Shosse, 443086, Samara, Russia
[2]Image Processing Systems Institute – Branch of the Federal Scientific Research Centre "Crystallography and Photonics" of Russian Academy of Sciences, 151 Molodogvardeyskaya st., 443001, Samara, Russia

Abstract

The present work is devoted to the analysis of local objects on radar images. In comparison, the following algorithms are used: decision tree; Bayesian classifier for normal distribution; Nearest neighbor method; Support Vector Method (SVM). As preliminary processing of images provided by a synthetic aperture radar. The research is carried out on the objects from the base of radar images MSTAR. The paper presents the results of the conducted studies.

Keywords: Classification of images; Synthetic aperture radar; Classification; Decision tree; C4.5; CART; SVM; MSTAR

1. Introduction

Radar satellite imagery obtained with synthetic aperture radars allows obtaining images of good quality in difficult weather conditions, as well as in cases of low illumination. A certain complexity in the processing of the obtained images is the speckle noise that is present on the radar images. The recognition of images on radar images is used in various fields, such as agriculture, forestry, relief analysis, oil spill monitoring and equipment recognition. Studies of various algorithms for the classification of radar images has been conducted previously, but often they had been compared the obtained data with the data from other articles. Such an approach may lead to inaccurate results of the analysis of the results obtained. Also, most articles use for comparison only the most popular classification algorithms, such as SVM, AdaBoost and neural networks. In this paper, the study adopted classifiers, which are used in works on this topic less often: decision trees, k nearest neighbors method, naive Bayesian classifier. The purpose of this paper is to fill this gap. All tests were conducted using the public database of radar images of MSTAR military equipment.

2. Statement of the classification problem

The task of object recognition on an image can be divided into two main subtasks:

- search for an object in the image and selection of areas of interest;
- recognition and classification of the found object or area of interest. [1]

The first subtask is aimed at finding objects for classification. Often information about the location, size, orientation, availability and number of goals is initially missing. In this case, it is necessary to determine the unknown parameters required for further selection of the object or local area of interest.

The second subtask is applied to the entire image and allows you to decide which of the several classes the image being processed belongs to. The goal of the classification is the construction of a decision function. The decision function for each feature vector relates the corresponding class. In this article, we consider only the classification problem.

In connection with the need to process a large number of images for training and testing, as well as low performance of some algorithms, there is a need to reduce the dimensionality of the feature space. There are various methods used to solve this problem. Such methods include the most popular ones: the method of principal components, factor analysis, the method of independent components, self-organizing maps of Kohonen and others. In this paper, the principal component method is applied.

3. Principal component analysis

The Principal Components Analysis (PCA) method is one of the most widely used methods for reducing the dimensionality of a feature space with the loss of the least amount of information. This method reduces to calculating the eigenvalues of the covariance matrix of the analyzed image. [2] Algorithms for calculating the covariance matrix operate in the line-by-line mode of reading the image, which allows achieving high performance and low requirements for RAM. [3]

4. Evaluation of classification results

To assess the results of the classification, the sliding control method is used. Sliding control (cross-validation, CV) is a statistical method for assessing the generalization of the quality of classification. It is a more reliable and thorough assessment method, compared to the usual sequential division of a data set into a training and test sample. With the sliding control, the data is repeatedly divided into training and test sets and fed to the classifier's input.

The paper uses a modified method of sliding control with multiple partitioning [4] in which the entire volume of data is divided into a specified number of parts of N (equal to 1). The number of iterations of learning in this algorithm corresponds to the number of blocks N. There is also a stratification of classes and samples, allowing reducing the dispersion of estimates of sliding control. This leads to a decrease in the confidence interval and a more accurate classification quality. Applying class stratification makes it possible to break each class in a given ratio. At each iteration of the algorithm, K parts are randomly selected as the training sample and L parts as the test sample. This partition can be described as follows:

$$\mathrm{N} \geq \mathrm{K} + \mathrm{L}, \Omega_{\mathrm{O}} \cup \Omega_{\mathrm{T}} = \Omega', \Omega' \subseteq \Omega, \Omega_{\mathrm{O}} \cap \Omega_{\mathrm{T}} = \emptyset,$$

$$\Omega_{\mathrm{O}} = \bigcup_{i=0}^{\mathrm{K}-1} \Omega_{\mathrm{O}i}\,, \Omega_{\mathrm{T}} = \bigcup_{j=0}^{\mathrm{L}-1} \Omega_{\mathrm{T}j}.$$

Where Ω_{O} is a training sample, Ω_{T} is a test sample, Ω is the original sample.

For each partition obtained, the classifier is set up on the training sample and the quality value of the classifier is calculated on the test sample.

The functional quality of the algorithm on the sample has the following form:

$$\mathrm{CV}(\mu, \Omega') = \frac{1}{\mathrm{N}} \sum_{p=1}^{\mathrm{N}} \frac{1}{\mathrm{N}} \sum_{q=1}^{\mathrm{N}} \mathrm{Q}\big(\mu\big(\Omega' \backslash \Omega_{\mathrm{O}pq}\big), \Omega_{\mathrm{O}pq}\big).$$

μ is the learning method.

5. Algorithms of classification

5.1. Bayesian Classifier Gaussian Case

The classification method based on the naive Bayesian classifier is a learning algorithm with the teacher, in which the Bayes theorem is applied with a strict (naive) assumption of independence between each pair of characteristics [5]. The assumption of independence makes it possible to get rid of a complex scheme for evaluating the parameters of the classifier. This allows us to apply the algorithm to large samples. Also, the classification is quite accurate: insufficient for high-precision classification systems, but satisfactory for rough estimation and comparison with other algorithms. Proceeding from the Bayes theorem:

$$\mathrm{P}(y|x_1, \dots, x_n) = \frac{P(y)P(y|x_1, \dots, x_n)}{P(x_1, \dots, x_n)}$$

and the independence assumption, we obtain:

$$P(x_i|y, x_1, \dots, x_{i-1}, x_{i+1}, \dots, x_n) = P(x_i|y),$$

that can be rewritten:

$$\mathrm{P}(y|x_1, \dots, x_n) = \frac{P(y) \prod_{i=1}^{n} P(x_i|y)}{P(x_1, \dots, x_n)},$$

and get the resulting classifier-function $\hat{y}$:

$$\hat{y} = arg \max_{y} P(y) \prod_{i=1}^{n} P(x_i|y).$$

and use the a posteriori maximum estimate to estimate $P(y)$ and $P(x_i|y)$.

Naive Bayesian classifiers differ, mainly, by the assumptions they make about $P(x_i|y)$.

In this paper, the densities used are the Gaussian case, which is based on the use of the probability density of the form:

$$P(x_i|y) = \frac{1}{\sqrt{2\pi\sigma_y^2}} e^{\left(-\frac{(x_i - \mu_y)^2}{2\sigma_y^2}\right)},$$

where μ_y and σ_y^2 are the mathematical expectation and the correlation matrix.

5.2. KNeighbors

The nearest neighbor's algorithm k refers to metric classification algorithms with training sample Ω_{O}. Such algorithms refer object u to that class $y \in \mathrm{Y}$, for which the total weight of the nearest objects from the training sample is maximal:

$$a(u, \Omega_{\mathrm{O}}) = arg \max\nolimits_{y \in Y} \Gamma_y(u, \Omega_{\mathrm{O}}), \text{и } \Gamma_y(u, \Omega_{\mathrm{O}}) = \sum_{i=1}^{\mathrm{K}} \left[y_u^{(i)} = y\right] \omega(i, u).$$

Where the weight function $\omega(i,u)$ estimates the degree of importance of the i-th neighbor for the classification of the object u. The function $\Gamma_y(u,\Omega_O)$is an estimate of the closeness of the object u to the class y. The importance function is chosen to be non-negative and not increasing in i. The selection criteria are due to the fact that the smaller the distance between the sampled objects u and $x_u^{(i)}$, the greater the probability of a correct classification. In the algorithm k of the nearest neighbors, the object u is referred to a class with more elements among the k nearest neighbors $x_u^{(i)}$, $i = \overline{1,k}$:

$$\omega(i,u) = [i \le k]\omega_i, a(u,\Omega_O,k) = arg\max_{y\in Y} \sum_{i=1}^{k}\left[y_u^{(i)} = y\right]\omega_i.$$

As a metric, the Euclidean metric is most often chosen because of its simplicity and comprehensibility. Three metrics are studied: Euclidean, Minkowski and Manhattan distance.

The Euclidean distance between two points x, y is defined in Euclidean n-dimensional space as:

$$r(x,y) = \sqrt{(x_1-y_1)^2+(x_2-y_2)^2+\dots+(x_n-y_n)^2} = \sqrt{\sum_{i=1}^{n}(x_i-y_i)^2}.$$

The Manhattan distance is defined as the sum of the moduli of the coordinate differences:

$$r(x,y) = \|\vec{x}-\vec{y}\| = \sum_{i=1}^{n}|x_i-y_i|.$$

Another metric on the Euclidean space, which is investigated in the paper is the Minkowski metric. It can be regarded as a generalization of the Euclidean and Manhattan distances. For the parameter $p = 2$, the Minkowski distance is generalized to the Euclidean distance, and for $\mathrm{p} = \infty$- to the Chebyshev distance. This metric is defined by the following formula:

$$r(x,y) = \left(\sum_{i=1}^{n}|x_i-y_i|^p\right)^{1/p}.$$

The drawbacks of metric algorithms include storage of the entire training sample.

5.3. Decision Tree (C4.5, CART)

A decision tree is a structure of a hierarchical type, in which branches a partition of the feature space is defined, and the sheets are elementary classification functions. There are various methods for constructing trees. In this paper, the algorithms C4.5 [6] and CART [7] will be considered.

C4.5, receiving the input sample Ω_O, builds the source tree, based on the following rules. If all objects in the sample belong to the same class or the sample is small, then the tree is a sheet marked with the most common class in the sample. Otherwise, a split criterion is selected that divides the sample into two or more samples. Then the criterion is chosen for the obtained partitions. This procedure is recursively applied for each sample received. One of the criteria is used to minimize the entropy value of the obtained sample partitions. The resulting source tree is then trimmed to avoid retraining. Based on the received tree, a decision function is constructed for classifying objects.

In the CART algorithm, a binary decision tree is recursively constructed. The tree is created to the maximum size without using the stopping rule, and then it is clipped. The algorithm builds not one but a sequence of nested truncated trees. The best division is selected based on the sliding control. The partition criterion is based on the Gini index.

5.4. SVM

The support vector machine is one of the most reliable methods among all known algorithms and is most often used for comparison with new algorithms. The function separating the classes is a separating hyperplane. The algorithm maximizes the shortest distance between the points closest to the points on the hyperplane [8]. In this paper, the linear separating function and the radial basis function are used as the separating function.

6. Experimental research

All the experimental studies were conducted on a PC Intel Core i5-4460, 16 GB RAM. All classification algorithms were written in the programming language Python 3.6. Also used were frameworks and libraries scikit-learn, openCV, numpy. As objects of classification, samples of military equipment from the public database of radar images MSTAR, presented in Figure 1, served.

For recognition, the magnitudes of the images of BMP-2, BTR-60, BTR-70 and T-72 were used. As preprocessing of images, the orientation of objects on centered images was normalized and cropped from 128 × 128 to 60 × 60 pixels. The resulting images are shown in Figure 2.

The target shooting angle is 15 and 17 degrees. The initial sample consists of 3438 images of different classes of objects. The number of images of each class is shown in Table 1.

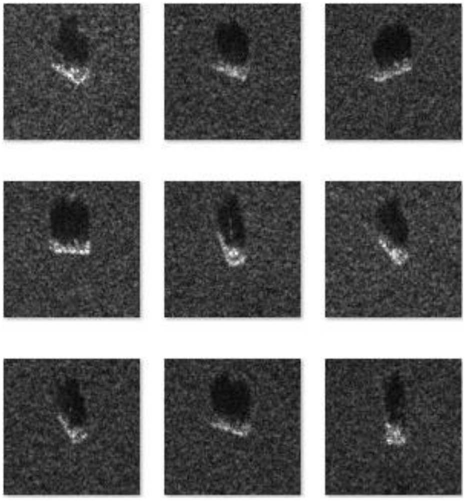

Fig. 1. Images of MSTAR objects.

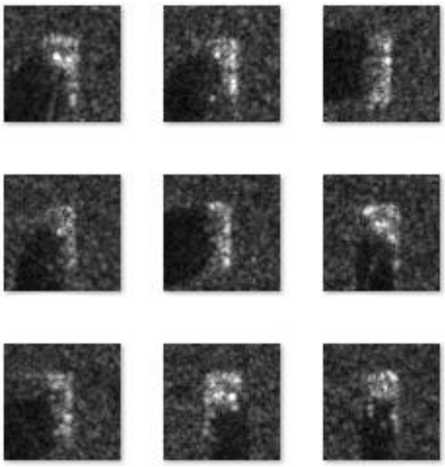

Fig. 2. Processed MSTAR images.

Table 1. Number of classification objects.

Object / Angle	*15°*	*16°*
BMP-2	*587*	*698*
BTR-60	*195*	*256*
BTR-70	*196*	*233*
T-72	*582*	*691*

For all images from the general sample, the dimension was reduced. The list of investigated classifiers is given in Table 2.

Table 2. Investigated classifiers.

GaussianNB	Naive Gaussian Bayesian classifier
KNeighbor_1	Nearest neighbor method, Euclidean metric
KNeighbor_2	The nearest-neighbor method, the Manhattan distance
KNeighbor_3	The nearest-neighbor method, the Minkowski distance
CART	The decision tree based on the CART algorithm
C4.5	The decision tree based on the C4.5 algorithm
SVM_1	The method of support vectors for a linear separating function
SVM_2	The method of support vectors for a radial basis function

The value of the classification quality will be calculated as the average relative number of correctly classified objects from the test sample Ω_T. For the sliding control method, we specify the number of partitions and the number of iterations $N = 10$, $K = 6$ and L = 4, dividing the total sample in the ratio 6: 4. The method of the main components will reduce the dimension to 20 eigenvectors, retaining a significant part of the radar image information necessary for the classification of objects. Detailed classification results are presented in Table 3.

Table 3. Results of classification.

	BMP-2	BTR-60	BTR-70	T72	Avg.
GaussianNB	*0.10895*	*0.62011*	*0.78899*	*0.74476*	0.56570
KNeighbor_1	*0.98156*	*0.95438*	*0.98680*	*0.97407*	0.97420
KNeighbor_2	*0.98586*	*0.95476*	*0.98721*	*0.97448*	0.97558
KNeighbor_3	*0.98152*	*0.96915*	*0.98481*	*0.97208*	0.97689
CART	*0.85964*	*0.78115*	*0.77543*	*0.86927*	0.82137
C4.5	*0.87424*	*0.75564*	*0.85257*	*0.87975*	0.84055
SVM_1	*0.92018*	*0.66914*	*0.89281*	*0.87513*	0.83932
SVM_2	*0.96351*	*0.96602*	*0.95939*	*0.99392*	0.97071

The support vector machine with a radial basis function and the k nearest-neighbor method (with Minkowski distance) showed the best result of the classification of radar images.

6. Conclusion

It can be seen from the results of the conducted research that the best indicators of the classification of radar images of the MSTAR base are given by the method of the nearest neighbors, and by the support vector machine. In subsequent studies, it is planned to apply boosting algorithms, such as AdaBoost [9], and neural networks. Over the past few years, there have been many publications using neural networks [10] for the classification of radar images, so their study and comparison of classification results with the results obtained in this paper is of great interest.

References

[1] Gashnikov M, Glumov NI, Ilyasova NYu, Myasnikov VV, Popov SB, Sergeev VV, Soifer VA, Khramov AG, Chernov AV, Chernov VM, Chicheva MA, Fursov VA. Methods of computer image processing. Ed. Soifer VA. 2 nd ed. rev. Moscow: Fizmatlit, 2003; 784 p.

[2] Method of the main components URL: http://www.machinelearning.ru/wiki/index.php?title=Method_head_component (01/26/2017).

[3] Kuznetsov AV, Myasnikov VV. Comparison of algorithms for controlled element-by-element classification of hyperspectral images. Computer Optics 2014; 38(3): 495–502.

[4] Vorontsov KV. Combinatorial approach to the evaluation of the quality of learning algorithms. Mathematical problems in cybernetics. Ed. Lupanov OB. Moscow: Fizmatlit, 2004; 13: 5–36.

[5] Naive Bayes. URL: http://scikit-learn.org/stable/modules/naive_bayes.html (23.01.2017).

[6] Quinlan JR. C4.5: Programs for Machine Learning. San Mateo: Morgan Kaufmann Publishers Inc., 1993; 302 p.

[7]Wu X, Kumar V, Ross Quinlan J.Top 10 algorithms in data mining. Knowledge and Information Systems 2008; 14(1). DOI:10.1007/s10115-007-0114-2.

[8] Cortes C, Vapnik V. Support-Vector Networks. Machine Learning 1995; 20(3): 273–297.

[9] Sun Y, Liu ZP, Todorovic S, Li J. Adaptive Boosting for SAR Automatic Target Recognition. IEEE Trans. Aerosp. Electron. Syst. 2007; 43(1): 112–125.

[10] Profeta A, Rodriguez A, Clouse HS. Convolutional neural networks for synthetic aperture radar classification. Algorithms for Synthetic Aperture Radar Imagery XXIII 2016; 9843. DOI:10.1117/12.2225934.