RADAR SIGNAL CLASSIFICATION USING PCA-BASED FEATURES

Amit Kumar Mishra and Bernard Mulgrew

IDCOM, University of Edinburgh, Edinburgh, UK

ABSTRACT

Principal component analysis (PCA) has been used in many applications ranging from social science to space science, for the purpose of data compression and feature extraction. Usage of PCA for synthetic aperture radar (SAR) image classification, though widely reported by remote-sensing researchers, has not been exploited much by automatic target recognition (ATR) community. In the present paper, PCA has been used in SAR-ATR using the MSTAR data base, and comparison has been made with the conventional *conditional Gaussian model based Bayesian classifier* [1]. The results have been compared based on percentage of correct classification, receiver operating characteristics (ROC), and performance with limited amount of training data. By all standards of comparison, the PCA based classifier was observed to outperform the conditional Gaussian model based Bayesian classifier (CGBC) or at the worst it performs at par. And given the computational and algorithmic simplicity of PCA based classifier, the new algorithm was concluded to be a highly prospective candidate for real time ATR systems.

1. INTRODUCTION

Principal component analysis (PCA) has been used in data analysis and data compression for a long time [2, 3]. Usage of PCA for feature extraction has shown many advantages in many fields. In the current work, we discuss the usage of PCA on radar data for the recognition of ground targets from their radar images. Though PCA has been used in works reported in open literature, for remote sensing data classification [4, 5, 6], use of the same for target recognition task has not been well exploited. The novelty of the present work lies in use of PCA for ATR exercise, and the development and analysis of a simple nearest neighbor based classification algorithm based on PCA-extracted features, which is conceptually simple and computationally extremely fast.

The rest of the paper has been arranged as follows. The next section gives an overview of the database used for testing the classification algorithms, which is followed by an overview of the classification algorithms used in the present work. This is followed by a report of the results, and then the conclusion. In the appendix the confusion matrices of some of the experiments have been given as a more complete form of result display.

2. DATABASE USED

Database used for the validation of the classifiers proposed, is the SAR images of five military ground targets. The dataset has been collected by the MSTAR program [7]. Moving and stationary target acquisition and recognition (MSTAR) program is a DARPA supported project for collecting a standardized mono-static SAR image database, collected using the Sandia National Laboratories Twin Otter SAR sensor payload operating at X-band. The targets used for the present experiments are the 2S1 tank (t000), D-7 land clearing vehicle (t005), T62 tank (t016), ZIL131 APC (t025), and ZSU-23 (t026). The target clips collected at an elevation of 17 degrees were taken to train the classifiers and those collected at elevation of 15 degrees were taken as test images. The image clips were resized to 96x96 pixels.

3. EXPERIMENTAL SETUP

In the present work, the conditional Gaussian model based Bayesian classifier [1] (CGBC) was taken as the benchmark for comparing the results. This is mainly because as per results reported in open literature; this classifier is one of the most successful algorithms for SAR-ATR. Secondly, due to the use of Bayesian classification algorithm, this algorithm is the theoretically best algorithm (given there is enough training data and correct probability density function is found for the database). In this, each pixel of the image clips is assumed to be from a Gaussian distribution, conditioned or depending on the target type and target pose.

$$r = s(\Theta, a) + w \tag{1}$$

where , r is the observed intensities of the pixels arranged in a one dimensional vector, w is additive Gaussian noise, s is the signal conditioned on Θ the target pose angle, and a the target type. The log-likelihood of an observed r, given $\{\Theta, a\}$ can be shown to be proportional to [1]:

$$-\sum_{i=1}^{N}\left[\log(\Sigma_i)+\left(\frac{r_i-M_i}{\Sigma_i}\right)^2\right] \quad (2)$$

Where r_i is the i^{th} pixel of the test-image-clip, Σ_i, M_i are the standard deviation and mean of the pixel respectively (as estimated from the training data), and N is the total number of pixels in the test-image-clip. In this method, the recognition is done as per the Bayesian rule of maximizing the probability

$$P(a \mid r) = P(r \mid a)P(a) \quad (3)$$

P(a), the probability of each type of vehicle was taken to be equal.

In the method of using principal component analysis (PCA) [2, 3], the image pixels are assumed to be the ***observed variables***, depending upon the target type.

$$r = s(a) + w \quad (4)$$

where, *r* is the observed intensities of the pixels arranged in a one dimensional vector, and *w* is additive Gaussian noise. The database is arranged so that all image clips collected at a 15 degree elevation are taken as training data. Each image clip in the training data-set is from the same elevation but a different azimuth angle. Pixels of image clips are assumed as ***variables,*** taking different ***observations*** with changing azimuth angle. PCA is applied to the dataset to reduce the number of ***observed variables***. This is done in the following steps:

- For each image clip, the pixels are arranged into the observation vector, and consecutive image clips are taken as different observational values.
- All consecutive rearranged image-pixel vectors are stacked together to form the observation matrix.
- The observation matrix is normalized (to have unity variance), and all the observation vectors are zero centered. Let the final matrix be denoted by ***X.***
- From this observation matrix, the covariance matrix is found for the observation vector.

$$Q = X^H X \quad (5)$$

- Then the eigen-value operation is applied on ***Q,*** to get the eigen vectors.
- Eigen vectors corresponding to ***k*** largest eigen values are stacked together to form matrix ***V***.
- Using this matrix ***V***, the training dataset is reduced in dimension to ***k***. The final outputs from the training phase are the database in reduced dimension and the converting matrix ***V***.
- In test phase, the test image clip is reduced in dimension using the converting matrix ***V***.
- Next the Euclidean distance is found from each point in the training database, and the class giving the least distance is decided as the class of the test clip.

Hence this is a nearest neighbor (NN) classifier, taking the PCA-extracted data as features. We coined this simple classifier as PCA-NN classifier. In all the experiments reported in this paper, value of ***k*** has been kept to 20. Because, from previous experiments this number of principal components, have been shown to give the optimum classification performance [8].

4. RESULTS AND DISCUSSIONS

The classifiers will be compared based on .their performance over three criteria. Though, the more striking features will be presented in this section in the form of graphs and bar-charts, the confusion matrices from the major experiments have been given in the appendix, for more complete information on the performance of the algorithms.

The first criteria of comparison is the over all percentage of correct classification. This can be observed from table 1 and 4 in the appendix. For all the targets, the PCA-NN classifier performs better than the CGBC classifier.

As the second comparison criteria, the receiver operation characteristics (ROC) of the two classifiers were compared. The comparisons for the two algorithms for four of the targets are shown in figure 1. They show the percentage of correct classification (P_{cc}), versus the percentage of false alarm (P_{fa}) in a binary hypothesis test between the target of interest and all of the remaining targets. The probabilities of false alarm and correct classification have been calculated as per the standard works on SAR-ATR [1]. As can be observed, the performance of PCA-NN classifier is similar (target t000) or better (targets t005, t016 and t025) than the CGBC classifier.

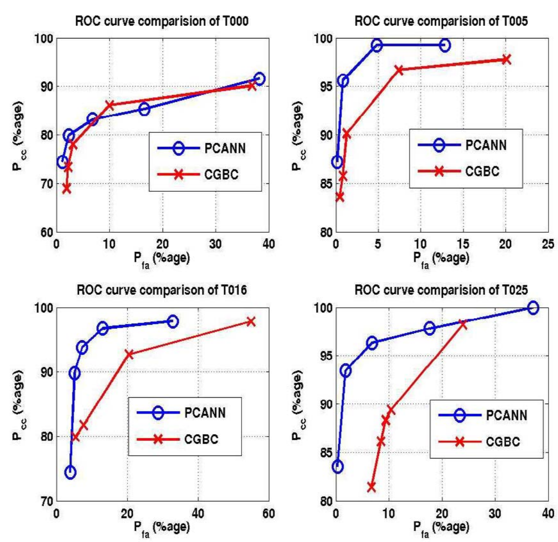

Fig.1 ROC comparisons for four of the targets

As the last criterion of comparison, the performances of the classifiers were studied with reduced amount of

training data. The reduction in training data was done in *two different ways*. First the training dataset was made sparse by discarding each alternate training clip (tables 2 and 5 in the appendix). In the second method, training data consisted of image clips with imaging platform azimuth from 0 to 180 degrees, while the test set had images from all azimuth angles. To analyze this more strong test, the test dataset was divided into two subsets, the first set (*set1*) consisting of images collected with azimuth 0 to 180 degrees and the second (*set 2*) from azimuth 180 to 360 degrees. The confusion matrices (table 3 and 6 in appendix) give the result for both test data sets. All the results have been presented in figure 2 and 3 in bar-chart form.
Looking at the over all performance of the classifiers with training data reduction, the loss of performance was more severe for CGBC classifier than for PCA-NN classifier

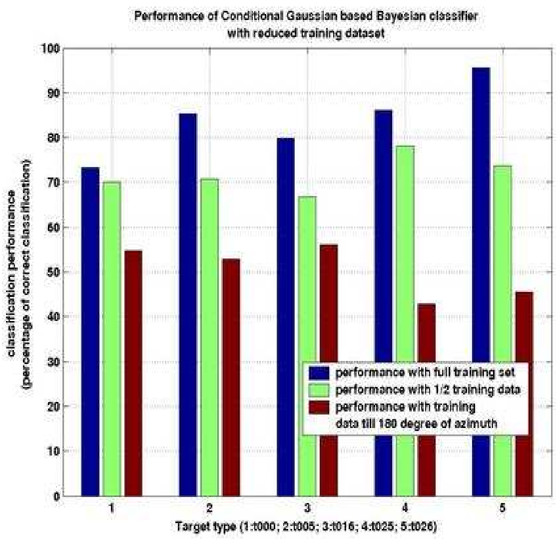

Fig.3 Performance of CGBC with reduced training dataset

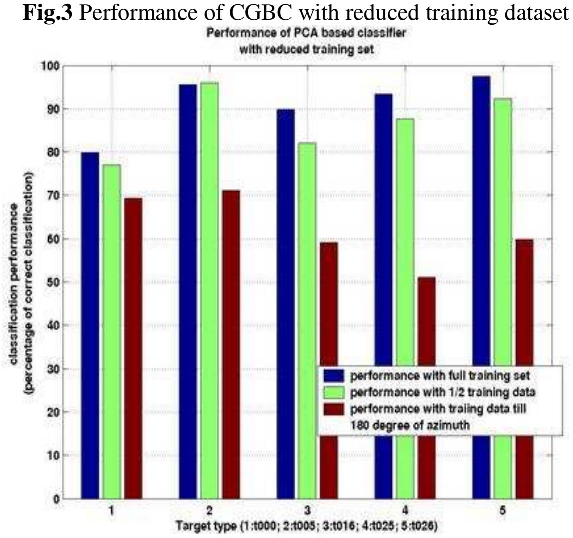

Fig.4 Performance of PCA-NN classifier with reduced training dataset

5. CONCLUSIONS

From the results found with the present work, it can be clearly concluded that the simple PCA based NN classifier out performs the CGB classifier in all the criteria of comparison. And due to the simplicity of the PCA-NN classifier, it takes several orders of less time for computation, than the CGB classifier. Given the reported success of CGB classifier, this makes a fairly strong algorithm for comparison. More over the extraction of the features of comparison (the data in PC-domain) is extremely fast for this new algorithm (just a matrix multiplication!). Hence the proposed PCA-NN classifier is a strong candidate for any real time ATR system, given its performance and computing speed.

6. ACKNOWLEDGEMENT AND DISCLAIMER

The authors acknowledge the electromagnetic remote sensing defense technology center (EMRSDTC) for funding this project and the royal academy of engineers.
Any views expressed are those of the authors and do not necessarily represent those of MOD or any other UK government department.

11. REFERENCES

[1] M.D. DeVore, and J.A. O'Sullivan, "Performance Complexity Study of Several Approaches to ATR from SAR Images", *IEEE Trans. on Aerospace and Electronic Systems*, vol.38, no.2, 632-648, April 2002.

[2] I.T. Jolliffe, *Principal Component Analysis*, 2nd Ed., Springer press, 2002.

[3] Dunteman, and H. George, *Brief Description: Principal components analysis*, Newbury Park, Calif. ; London : Sage, 1989.

[4] M.R.A. Sadjadi, S. Ghaloum, and R. Zoughi, "Terrain classification in SAR image using principal component analysis and neural networks", *IEEE Trans. On Geoscience and Remote Sensing*, vol.31, no.2, 511-515, March 1993.

[5] M.Klenke, and V.Hochschild, "Reducing the radiometric terrain effect in SAR imagery by means of principal components analysis", IEEE Geoscience and Remote Sensing Symposium, vol.2, 1288-1290, 1999.

[6] T.Feingersh, B.G.H.Gorte, and H.J.C. VanLeeuwen, "Fusion of SAR and SPOT image data for crop mapping", IEEE Geoscience and Remote Sensing Symposium, vol.2, 873-875, 2001.

[7] https://www.sdms.afrl.af.mil/

[8] A.K.Mishra, and B.Mulgrew, "SAR-ATR using one and 2D PCA", International Radar Symposium (India), Bangalore, 19-22 December 2005.

9. APPENDIX

Table 1. Confusion matrices for CGBC based algorithm (full training dataset)

	t000	t005	t016	t025	t026	Performance
t000	201	2	18	49	4	73.35%
t005	2	234	7	17	14	85.40%
t016	20	1	219	23	10	79.92%
t025	1	2	30	236	5	86.13%
t026	2	4	4	2	262	95.62%

Table 2. Confusion matrices for CGBC based algorithm (sparse training dataset to ½ original size)

	t000	t005	t016	t025	t026	Performance
t000	192	2	13	67	0	70.07%
t005	3	194	14	46	17	70.80%
t016	48	1	183	32	9	66.79%
t025	17	13	21	214	9	78.10%
t026	9	14	32	17	202	73.72%

Table 3. Confusion matrices for CGBC based algorithm (training dataset for azimuth angles from 0 to 180 degrees only)

	t000	t005	t016	t025	t026	Performance (%)	Overall performance
t000 set1	112	0	22	3	0	81.7	54.74%
set2	38	1	48	49	1	27.7	
t005 set1	1	105	1	30	0	76.6	52.92%
set2	5	40	26	33	33	29.2	
t016 set1	17	7	97	11	5	70.8	56.20%
set2	52	5	57	21	1	41.1	
t025 set1	3	22	17	91	4	66.4	42.86%
set2	62	1	45	26	2	19.1	
t026 set1	1	6	9	20	101	73.7	45.62%
set2	28	24	52	8	24	17.5	

Table 4. Confusion matrices for PCA based algorithm (full training dataset)

	t000	t005	t016	t025	t026	Performance
t000	219	0	50	5	0	79.93%
t005	3	262	2	4	3	95.62%
t016	15	2	246	8	2	89.78%
t025	8	3	5	256	2	93.43%
t026	0	4	0	3	267	97.45%

Table 5. Confusion matrices for PCA based algorithm (sparse training dataset to ½ original size)

	t000	t005	t016	t025	t026	Performance
t000	211	0	52	8	3	77.01%
t005	1	263	3	7	0	95.99%
t016	31	3	225	7	7	82.12%
t025	13	10	8	240	3	87.59%
t026	0	12	0	9	254	92.34%

Table 6. Confusion matrices for PCA based algorithm (training dataset for azimuth angles from 0 to 180 degrees only)

	t000	t005	t016	t025	t026	Performance (%)	Overall performance
t000 set1	119	0	14	4	0	86.8	69.34%
set2	71	0	62	2	1	51.8	
t005 set1	0	127	0	3	7	92.7	71.12%
set2	9	68	6	37	17	49.6	
t016 set1	4	2	126	5	0	91.9	59.12%
set2	70	9	36	21	0	26.2	
t025 set1	2	2	6	125	2	91.2	51.09%
set2	62	29	30	15	1	10.9	
t026 set1	0	2	0	2	133	97.0	59.85%
set2	18	27	30	31	31	22.6	