

ASSESSMENT OF NEURAL NETWORK SCHEMES TO CLASSIFY CLOUD DATA

FAISAL HOSSAIN

TUFA DINKU

NEERAJ AGARWAL

EMMANOUIL N. ANAGNOSTOU

The University of Connecticut

ABSTRACT

Using remotely-sensed data from the Tropical Rainfall Measuring Mission (TRMM), a cloud classification study was undertaken employing neural network schemes. The objective of this study was to assess the accuracy of each scheme for classifying clouds. In the first phase, a data preprocessing and feature selection scheme was undertaken to identify a suitable set of features that could be useful in cloud classification. In the next phase, seven neural network classification schemes were implemented to understand the utility of each of these schemes. Parametric schemes performed poorly, while the perceptron, K-nearest neighbor approaches and the least means square algorithm yielded promising results. Further study is proposed so as to improve rainfall prediction.

INTRODUCTION

Clouds can be of various types depending on their morphological characteristics and propensity to precipitation. Clouds are classified by their appearance and height into two broad categories, stratiform and convective. Definite weather patterns are usually associated with certain clouds or combinations of clouds, so their study is important in weather analysis and forecasting. Cloud classification

also improves the parameterization of atmospheric heating due to release of latent heat in precipitating clouds systems [1]. This in turn helps in understanding the distribution of heating and the forcings of atmospheric circulation systems over a broad range of scales. One of the major sources of uncertainty in predicting climate impacts of global warming is in predicting how clouds will respond to and alter radiation fluxes [2]. An accurate identification of clouds helps in minimizing the uncertainty associated with predictive capabilities of climatological models. Airport control towers also require accurate cloud-type data for providing safe navigational guidance to incoming and outgoing aircraft [3]. Accurate cloud classification hence is important in various fields of application.

In this study, the main interest is cloud classification for the purpose of better rainfall estimation from radar data. The relation between radar-measured quantities and rainfall is different for different rain-bearing cloud types. Accordingly, accurate cloud classification is likely to improve the accuracy of rainfall estimation from radar observations. Recent cloud classification studies suggest that there are major discrepancies among various algorithms (classification methods) that rely on remotely sensed data. Some studies indicate that the use of texture-based pattern recognition features derived from remote-sensing can significantly improve the accuracy of classification [4]. However, the capabilities and accuracies which can be attained with the spatial information provided by remote sensing remain poorly understood overall, and the optimal feature set is open to question. Most cloud classification algorithms rely on linear and parametric schemes [4]. In this work, a comparison of cloud classification approaches is made using neural networks, starting with the selection of appropriate features and then turning to the classification schemes themselves.

In this study we have made use of already-classified cloud data of reasonable accuracy for our training and testing schemes. The data was acquired from the Tropical Rainfall Measuring Mission (TRMM)—a joint venture of the National Aeronautics and Space Administration (NASA) and Japan's National Space Development Agency (NASDA) that was launched in 1997. The TRMM satellite carries, among other instruments, precipitation radar (PR), a Microwave Imager (TMI), and Visible and Infra-red sensors [5]. Apart from the raw data of the different sensors, the TRMM project also provided products related to cloud classification. Reflectivity data from PR and the "rain" product have been used for this study.

The purposes of this study are to: 1) implement and test feature selection and data preprocessing schemes; and 2) implement and test cloud classification schemes using neural networks.

BACKGROUND

Limitations in extracting on the multispectral features from remotely sensed data have led to the development of expert system tools. Artificial Intelligence

(AI) techniques have come into increased use for analysis of remotely sensed data [6]. Due to limited knowledge of the physical processes in the environment and the inherent noise in remotely sensed data, environment systems often cannot be accurately represented by numerical values describing their physical properties and interactions yet still may be amenable to categorization. In remote sensing applications with special emphasis on cloud classification for better rainfall estimation, where pattern recognition and multivariate analysis are a common requirement, coupled numeric/symbolic systems may be useful. In this respect, given the computing power that is now available, the concept of neural networks, or a “connectionist” approach, is believed to have considerable potential for automatic cloud classification as suggested by applications to automated pattern recognition [7]. In the following section, the various approaches used for cloud classification are discussed with reference to remotely sensed data.

APPROACHES USED FOR CLOUD CLASSIFICATION

Cloud types may be distinguished using many criteria. Stratiform systems are characterized by widespread slow-ascent velocity fields, low rainfall (of less than 100 mm h^{-1}), and convective systems are driven by radiative heating and have strong up and downdraft, with smaller areal coverage. Anagnostou and Kummerow [8] reported that discrimination between convective and stratiform systems is an important part due to the differences in homogeneities of rainfall with the satellite’s field of view. The inhomogeneity affects rainfall retrieval algorithms and hence, if the inhomogeneity (and the non-linear nature of brightness-temperature-rainfall rate relations) is not taken into account, there would be severe underestimation of the highly variable convective rainfall.

Based on the inherent properties of the convective and stratiform systems, one of the most popular radar-separation clues in classifying clouds as stratiform or not is the presence of a radar brightband [9]. However, this method has its limitations because the brightband is not always exhibited in the early or late stages of development of stratiform precipitation [10]. Previous satellite-based classification schemes relied on: 1) a combination infra-red and visible channels to classify cloud types [11]; 2) cloud liquid-water paths detected at microwave wavelengths [12]; and 3) combined infra-red and microwave satellite data [13]. Although all of these studies validated their retrieval rainfall with independent ground-truth measurements, none of them reported validation of their cloud classification estimates.

A study by Welch et al. [14], used three approaches to cloud classification: 1) traditional stepwise discriminant analysis (SDA); 2) the feed-forward back-propagation (FFBP) neural network; and 3) the probabilistic neural network (PNN). The authors classified remotely sensed data from arctic regions into

five cloud types and achieved an overall classification accuracy of 85.6 percent, 87.6 percent, and 87.0 percent for SDA, FFBP, and PNN respectively. From a pool of 232 candidate spectral and textural features, the following nine features were found most valuable in the discrimination of cloud type:

1. Visible-channel cloud fraction;
2. Mean albedo of cloudy pixels;
3. Surface temperature;
4. Cloud-top temperature;
5. Infra-red-channel cloud fraction;
6. Low-cloud fraction;
7. Midlevel-cloud fraction;
8. High-cloud fraction; and
9. Multilayer-cloud index.

Due to the fact that FFBP required about two orders of magnitude more data, Welch et al. [14] concluded that the PNN was the best of the three ways of classification studied. On the other hand, it was noted that the PNN was slowest in evaluating the test data, while the FFBP was extremely fast.

In a related study, Anagnostou and Kummerow employed a convective stratiform (c/s) separation technique based only on the brightness temperature observations from the microwave imager [8]. Their scheme probabilistically related a quantity called the variability index (VI) to the stratiform fraction of precipitation over the satellite's field of view, using the brightness channels of 19.35, 22.235, 37.0, and 85.5 GHz.

Bankert [15] used a probabilistic neural network (PNN) on 95 expertly classified satellite images (taken from 7 maritime regions) to classify them into 10 cloud types (cirrus, cirrocumulus, cirrostratus, cumulonimbus, nimbostratus, stratocumulus, stratus, cumulus, cumulonimbus, and clear). Using the PNN technique, a 79.8 percent correct classification was achieved. It was noted that if a more general method using five cloud types were used, 91.2 percent of the samples could be classified correctly. Bankert therefore suggested that a two-layer, four-network system employing the PNN be used, in which the general classification (5 cloud type) of a sample is followed by specific classification (10 cloud type).

While it was noted that the use of a neural network model employing the back propagation algorithm showed great potential for pattern recognition and was superior in classification accuracy to statistical methods, it was also computationally more expensive and required long learning times for big data sets. On the other hand, if prior knowledge of the form of distribution of data was available, statistical methods outperformed neural networks in classifying test data and were an order of magnitude lower in computation complexity.

METHODOLOGY

Data Pre-Processing and Feature Selection

The TRMM sensor package consisted of a TRMM Microwave Imager (TMI)—a nine-channel, passive MW radiometer based on the SSMI [5], and a Precipitation Radar (PR) that provided a 3-D image of land and ocean rainfall and profiles of reflectivity in vertical and horizontal directions. A TRMM product called 2A23 classified cloud types as convective, stratiform, or mixed, and product 2A25 showed the 3-D structure of the rainfall. In [1], effective reflectivity and spectral width (of reflectivity) were studied for their cloud classification value. It was suggested that the horizontal and vertical reflectivity and rainfall profiles could be of use in discriminating cloud types from remotely sensed data.

The entire feature set considered in this study therefore was:

1. Storm Height (Cloud Height) (HT);
2. Reflectivity at Cloud Top (CTZ);
3. Horizontal Variation of Cloud Top Reflectivity (HVCTZ);
4. Bright-Band Intensity (BBI);
5. Horizontal Variation of Reflectivity (HVZ);
6. Vertical Variation of Reflectivity Type 1 (VVZ1);
7. Vertical Variation of Reflectivity Type 2 (VVZ2);
8. Horizontal Variation of Rainfall (HVR);
9. Vertical Variation of Rainfall Type 1 (VVRI); and
10. Vertical Variation of Rainfall Type 2 (VVR2).

Six regions in the southern United States were chosen for which adequate expertly-classified classes data pertaining to cloud type from TRMM remote sensing was available. Figures 1, 2, and 3 show typical cases of the PR scan with convective, stratiform, and mixed-cloud images respectively. As discussed earlier, many of the salient features distinguishing the two types of clouds can be easily observed from the figures.

In order to understand the utility of each feature in terms of its discriminatory property, histogram analyses for each feature with respect to the cloud types were made. Data sets pertaining to a mixed case of rainfall were studied as it was believed that would offer the most critical test of the discriminatory power of the features. While many have tried more sophisticated measures to assess the discriminatory power of individual features (e.g., the Bhattacharya distance [14]), the simple histogram analyses was believed to be adequate in this study given the lower number of classes. Data corresponding to the features was normalized to allow a common platform for comparison and easier analysis. Figures 4 through 10 show histograms of features used in convective/stratiform discrimination (only the ones revealing noticeable discrimination are presented). One histogram is for

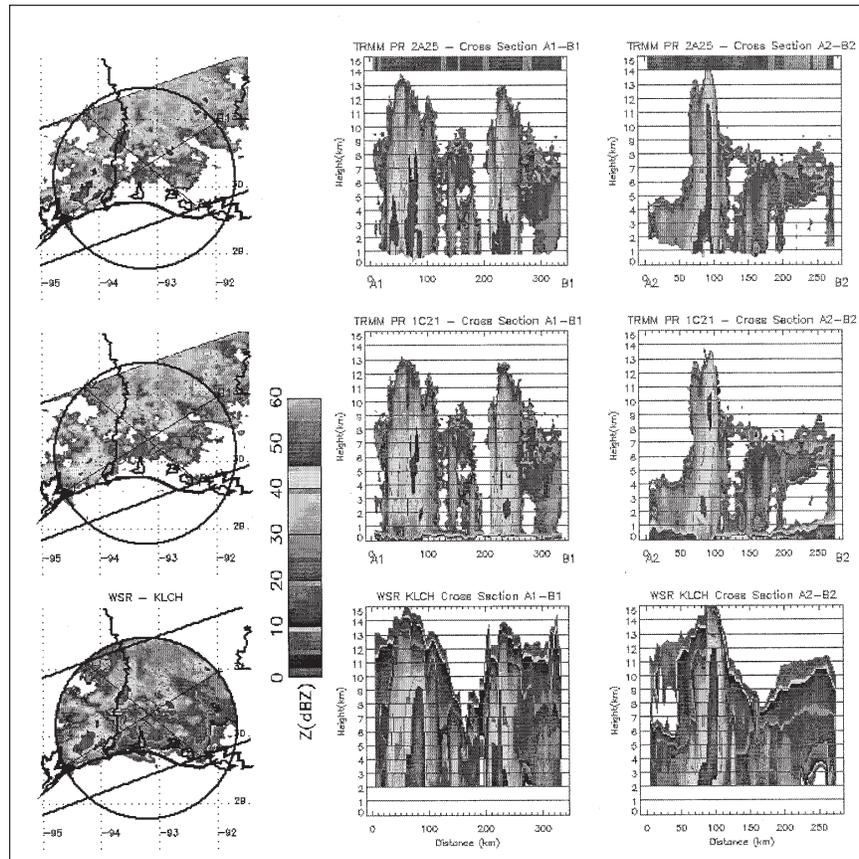


Figure 1. A typical case of convective rainfall (high rainfall, greater height, smaller areal coverage and no brightband).

convective and the other is for stratiform conditions. The abscissa represents the normalized discrimination parameter (a generic name). Based on the discriminatory power revealed graphically by the histograms, the following features were finally selected for cloud classification schemes:

1. HVZ;
2. BBI;
3. HVCTZ;
4. CTZ;
5. HT;
6. CTZ*HT

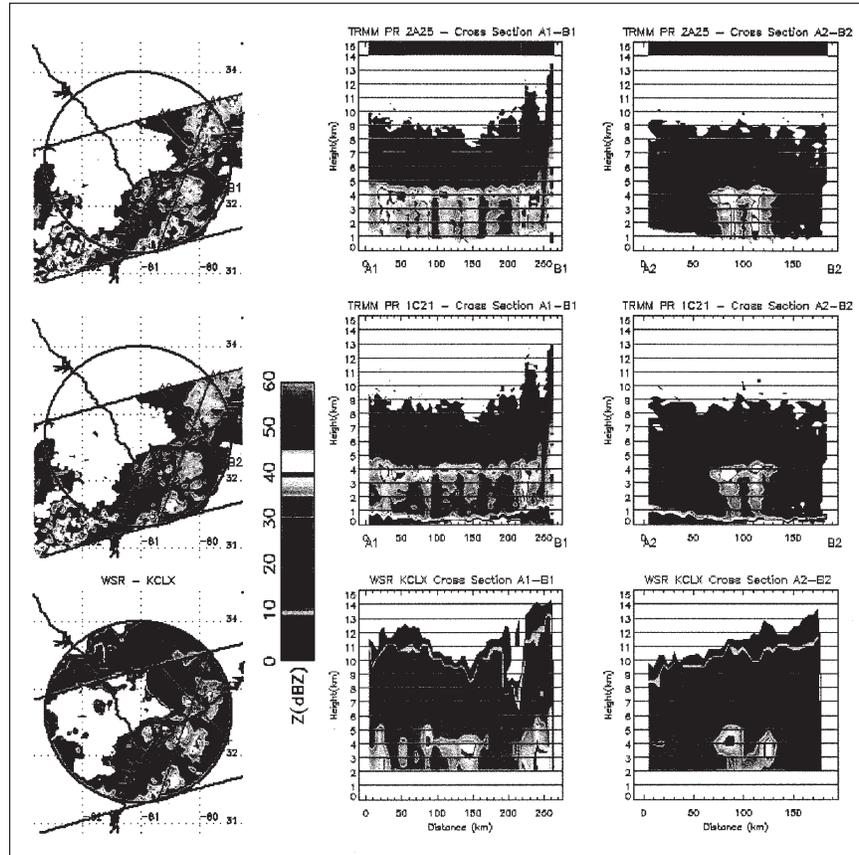


Figure 2. A typical case of stratiform rainfall (low rainfall, higher areal coverage, brightband, low cloud top height).

Classification Schemes

The following schemes were employed for the cloud classification:

1. Linear decision rule (Appendix A);
2. Quadratic decision rule (Appendix A);
3. Probabilistic neural network (PNN) (Appendix B);
4. K-nearest neighbor approach (KNN) (Appendix B);
5. Single-layer perceptron (SLP) and Multi-layered perceptron (MLP) (Appendix C); and
6. Least means square algorithm (LMS) (Appendix D).

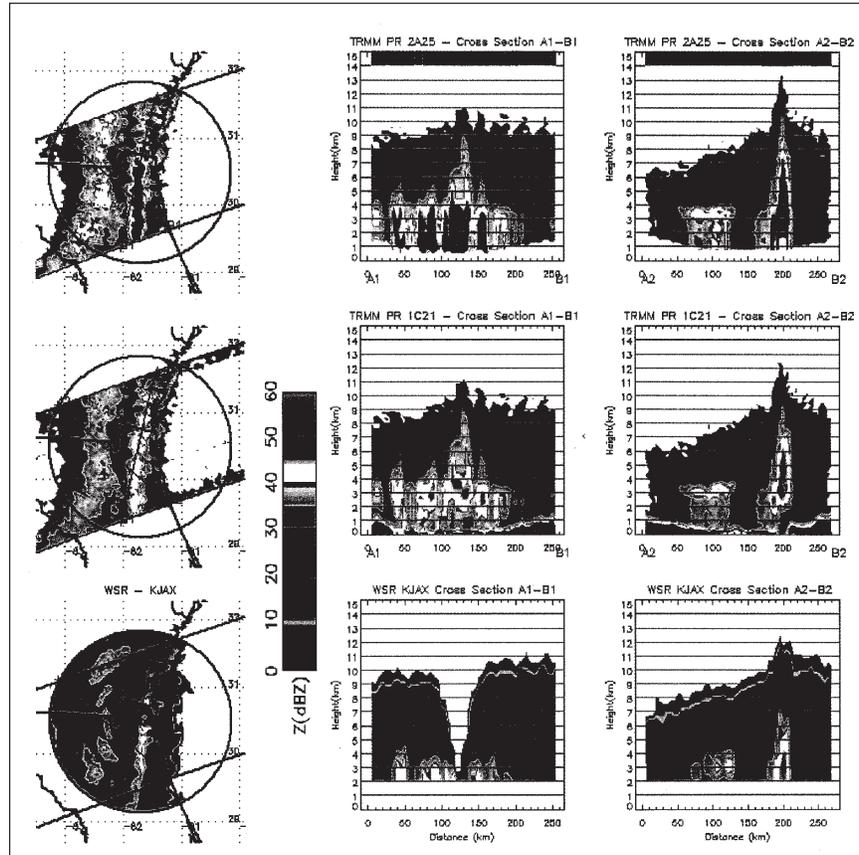


Figure 3. A typical mixed case (convective and stratiform) of rainfall.

The theory behind the classification is briefly discussed in the appendices [16, 17]. All classification schemes were cross-validated.

RESULTS AND DISCUSSION

Linear and Quadratic Decision Rules

Average classification errors of 41.52 percent and 50 percent were obtained with the linear decision rule and the quadratic decision rule respectively. Tables 1 and 2 present the confusion matrices for the linear and quadratic rule classifications. In the discrimination histograms, it may be seen that the assumption of Gaussian distribution of the features as required by these two parametric schemes

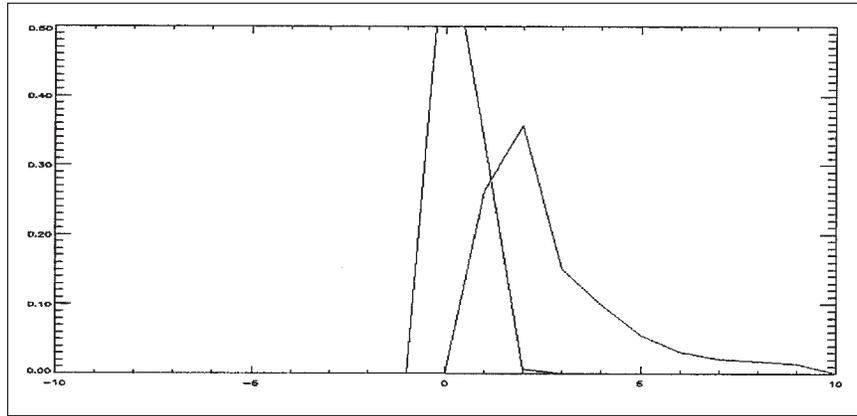


Figure 4. Histogram for Cloud Top Height (HT).

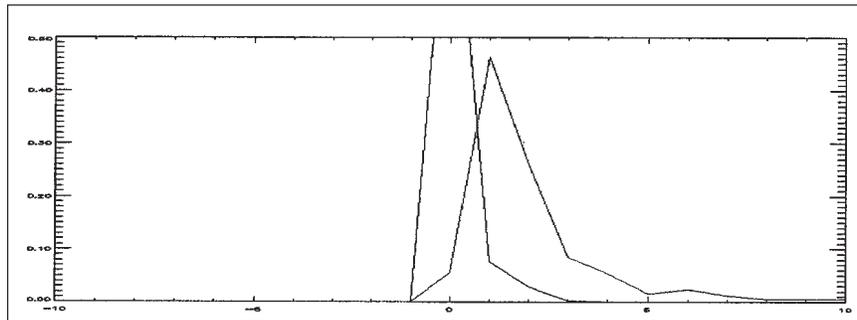


Figure 5. Histogram for Cloud top height * height.

is not fully satisfied for all cases. The results obtained with the parametric schemes do not compare well with that achieved by Welch et al. [14], who, using a traditional step-wise discriminant analysis, achieved an error in classification of about 15 percent. Perhaps the more sophisticated and larger choice of features employing the Bhattacharya distance [14] could be the reason for the better classification obtained with the parametric schemes.

Probabilistic Neural Network (PNN)

Unsatisfactory results were obtained with the PNN. As above, the assumption that the kernel function (*Parzen window*) by which the probability distribution is

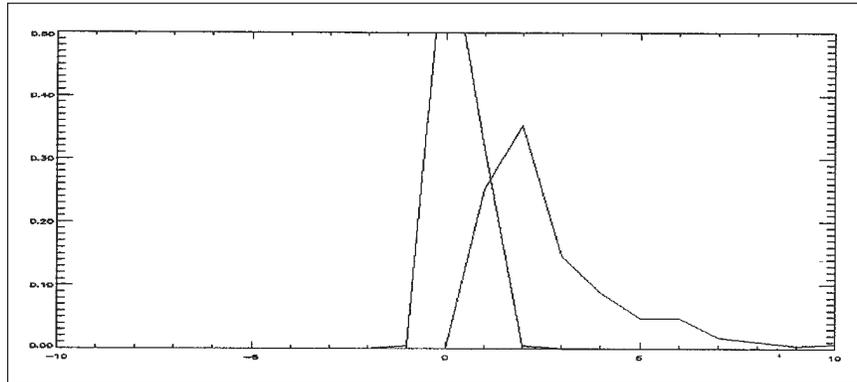


Figure 6. Histogram for horizontal variation of Cloud Top Reflectivity (HVCTZ).

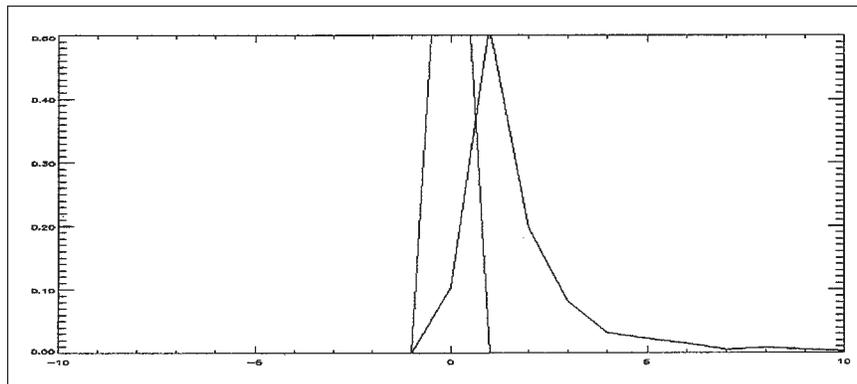


Figure 7. Histogram for Brightband Intensity.

modeled at each point of the data set (testing) is of multivariate Gaussian form is most likely not satisfied for all features. Hence, the apparent disagreement with [14] and [15] in terms of the suitability of the PNN might be explained.

K-Nearest Neighbor (KNN)

Excellent results were obtained with the KNN approach using k values of 7, 9, and 11. In all three cases the error in classification was found to be 0.0 percent. This could be useful; few previous workers have employed the KNN approach in classifying clouds—and KNN has the appealing property of being simple to

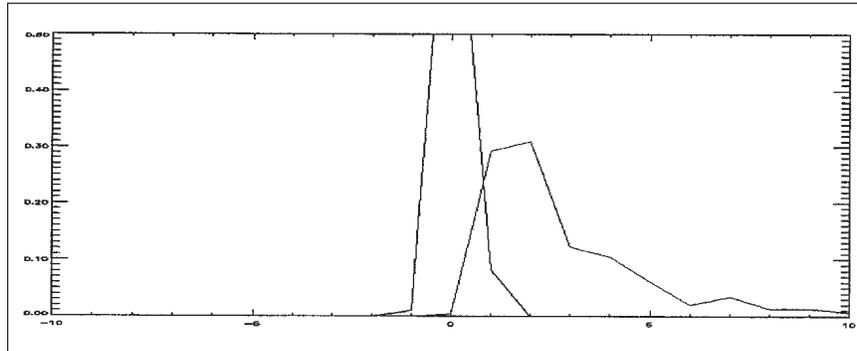


Figure 8. Histogram for Horizontal Variation of Reflectivity (HVZ).

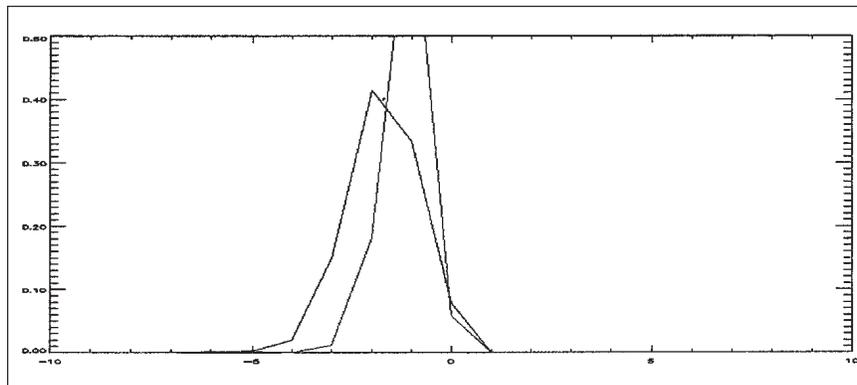


Figure 9. Histogram for Vertical Variation of Reflectivity (VVZ1).

implement with less intensive computational requirements during the training or testing phase. Table 3 shows the confusion matrix for the KNN approach.

Least Means Square Algorithm (LMS)

Excellent classification results were obtained using the LMS algorithm using two different learning-rate parameters ($\eta_1 = 0.001/|x|^2$, $\eta_2 = 0.0001/|x|^2$), with an error in classification of 1.4 and 0.0 for stratiform and convective rainfall respectively. Tables 4 and 5 show the confusion matrices for the LMS for two different learning rate parameters.

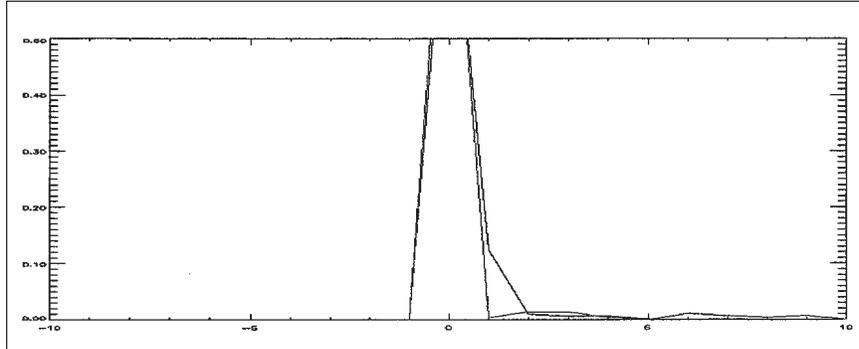


Figure 10. Histogram for Horizontal Variation of Rainfall (HVR).

Table 1. Confusion Matrix for Linear Decision Rule (41.52% Error)

Truth →	Stratiform	Convective
Stratiform	767	105
Convective	210	85

Table 2. Confusion Matrix for Quadratic Decision Rule (50% Error)

Truth →	Stratiform	Convective
Stratiform	862	10
Convective	293	2

Table 3. Confusion Matrix for KNN (0.0% Error)

Truth →	Stratiform	Convective
Stratiform	872	0
Convective	0	295

Table 4. Confusion Matrix for LMS (Learning Rate = η_1) (1.4% Error)

Truth →	Stratiform	Convective
Stratiform	859	13
Convective	0	295

Table 5. Confusion Matrix for LMS (Learning Rate = η_2) (0% Error)

Truth →	Stratiform	Convective
Stratiform	868	4
Convective	0	295

Single Layer Perceptron (SLP)

Excellent results also were obtained using the single layer perceptron: 0.0 percent error in classification. Training converged very fast, in just 100 iterations. Table 6 shows the confusion matrix for the SLP.

Multi Layer Perceptron (MLP)

Excellent results were obtained using the single layer perceptron: 0.6 percent error in classification. Training converged very fast, in just 50 iterations. Table 7 shows the confusion matrix for the MLP.

The summaries of the results showing the efficacy of each of the classification schemes are presented in Table 8.

PROPOSALS FOR FURTHER STUDY

1. Previous workers have employed information from brightness temperatures for classification of cloud types. It is proposed that extending this study so as to incorporate brightness temperature may improve classification.
2. Since classifying clouds into further categories would have more utility in rainfall estimation, additional cloud types should be included in future studies.
3. The most useful cloud classification for rainfall estimation is done with ground-based radars. Hence, cloud classification employing data from ground radar would enhance rainfall estimation and flood forecasting.

Table 6. Confusion Matrix for Single Layer Perceptron (SLP) (0.0% Error)

Truth →	Stratiform	Convective
Stratiform	872	0
Convective	0	295

Table 7. Confusion Matrix for Multi Layer Perceptron (MLP) (0.0% Error)

Truth →	Stratiform	Convective
Stratiform	861	11
Convective	0	295

Table 8. Summary of Results

Scheme	Comment on Accuracy	% Error in Classification
SLP	Excellent	0.0
KNN	Excellent	0.0
MLP (2-layer)	Excellent	0.6
LMS	Very good	1.4
Linear Decision Rule	Not good	41.52
Quadratic Decision Rule	Poor	50.0
PNN	Poor	Slowest classification

CONCLUSION

1. By means of data pre-processing and feature-selection schemes, a set of six features for cloud classification were identified:
2. Parametric schemes (linear and quadratic decision rules and the probabilistic neural network (PNN)) for classification performed poorly given the nature of data and features that were selected in this study. This indicated that the assumption of Gaussian nature of the data was not satisfied for all the features.
3. The K-nearest neighbor (KNN), single layer perceptron (SLP), and multi-layer perceptron (MLP) classified clouds with very high accuracy.
4. Neural network classification schemes can be useful in cloud classification. Several lines of further study in this regard are suggested.

APPENDIX A: Quadratic Linear Decision Rules

Assumptions

- 1) Data corresponding to each class are distributed according to the Gaussian form so that the probability density function is well-defined and parameterized (and not estimated from the data itself)
- 2) The class characteristic of a data point is characterized by the posterior probabilities.
- 3) The decision boundaries for classification (in the training data) are ideally located at the points of intersection of the probability density curves for each class so that the probability of misclassification is minimized.

Quadratic Discriminant Rule:

Given the above assumptions, a data vector $[x]$ can be classified according to the class of argument "k" where k is given by,

$$k = \arg \min \left[\frac{1}{2} \ln |\Sigma_k| + \frac{1}{2} ([x] - \mu_k)^T \Sigma_k^{-1} ([x] - \mu_k) - \ln P(C_k) \right] \quad (\text{A.1})$$

Where the symbols have their usual significance (and, C refers to a class).

Linear Discriminant Rule

A further ASSUMPTION leads to a simplification of the Eq. A.1 (Quadratic Discriminant):

- 4) Covariance matrices for various classes are equal such that $\Sigma_k = \Sigma$ resulting in the Linear Discriminant Rule as

Given the above 4 assumptions, a data vector can be classified according to the class of argument k, where "k" is given by,

$$K = \arg \min \left[-\mu_k^T \Sigma^{-1} [x] - \ln P(C_k) + \frac{1}{2} \mu_k^T \Sigma^{-1} \mu_k \right] \quad (\text{A.2})$$

APPENDIX B: PNN and KNN

Probabilistic Neural Network (PNN)

Assumptions:

- The kernel function (*Parzen window*) by which the probability distribution is modeled at each point of the data set (testing) is assumed to be of multivariate Gaussian form.
- The size of the hypercube h plays the role of variance as in the Gaussian distribution.

Algorithm:

Step 1. Sort training data according to class. Compute $N_k, P(C_k)$.

Step 2. Initialize the size of the hypercube h . ($h = 0.01, 0.02, 0.1, \dots, 0.8$).

Step 3. Pick a data point x_{test} from testing data.

Step 4. a) Compute unconditional $p(x_{test})$ as

$$p(x_{test}) = \frac{1}{N} \sum_{n=1}^N \frac{1}{(2\pi h^2)^3} \exp\left(-\frac{\|x_{test} - x_{train}^n\|^2}{2h^2}\right) \quad (B.1)$$

where N refers to the entire training data set domain regardless of class.

Step 4. b) Compute class conditional $p(x_{test}|C_k)$ for each class as

$$p(x_{test}|C_k) = \frac{1}{N_k} \sum_{n_k=1}^{N_k} \frac{1}{(2\pi h^2)^3} \exp\left(-\frac{\|x_{test} - x_{train}^{n_k}\|^2}{2h^2}\right) \quad (B.2)$$

where N_k refers to the training domain of the class C_k

Step 5. Compute posterior probability of x_{test} for each class as,

$$P(x_{test}|C_k) = \frac{p(x_{test}|C_k)P(C_k)}{p(x_{test})} \quad (B.3)$$

Step 6. Classify x_{test} as belonging to class k such that,

$$k = \arg \max P(x_{test}|C_k)$$

Step 7. Go to Step 3 and repeat for all x_{test} . Compute % classification and PE(%).

Step 8. Go to Step 2 and repeat for $h = h + 0.01$.

K-Nearest Neighbor Approach

Algorithm:

Step 1. Initialize k (number of nearest neighbor) ($k = 1, 3, 5$).

Step 2. Pick a test data point x_{test}

- Step 3.** Identify the k nearest neighbors in training data set with respect to x_{test}
Step 4. Find K_k for each class.
Step 5. Classify x_{test} as belonging to C_k for which $k = \arg \max K_k$.
Step 6. Go to Step 2 and repeat for all other test data points.

APPENDIX C

Single and Multi Layer Perceptrons

Single Layer Perceptron (SLP)

Assumptions:

1. The data is linearly separable into the classes.
2. For this multi-class classification problem, the activation function can be used to approximate the posterior probabilities for each data vector. (*In other words, the sum of three exponential functions is approximated to a single exponential function.*)

Algorithm: (Generalized)

Step 1. Initialize w_j^1 —the weight vector for each class C_j .
Step 2. Start iteration, $ii = 1, 1000$ (say 1000 epochs).
 DO $ii = 1, 1000$
 $N_error=0$
 DO $n = 1, N_{train}$ (N_{train} is the total number of patterns)
 Pick a training data \mathbf{x}_{train}^n whose argument j is known.
 Compute $\mathbf{w}_j^n T \mathbf{x}_{train}^n$ for each class C_j . (The weight vector exist for each class)
 Compute k , where $k = \arg \max i (\mathbf{w}_i^n T \mathbf{x}_{train}^n)$
 If $j=k$ do nothing (go the next pattern, $n=n+1$) (right classification)
 Else
 $N_error=N_error+1$
 $w_j^{n+1} = w_j^n + \eta \mathbf{x}_{train}^n$ (C.1)
 $w_k^{n+1} = w_k^n - \eta \mathbf{x}_{train}^n$ (C.2)
 ENDIF
 ENDDO
 Print*, '%Misclassification =', $(N_error/N_{train}) * 100.0$
 ENDDO (End iteration once % Misclassification rate stabilizes to a nominal value (i.e., becomes constant))

Testing: (Cross-validation)

- Step 1.** Pick a test data vector \mathbf{x}_{test} .
Step 2. Compute $\mathbf{w}^T \mathbf{x}_{test}$ (where \mathbf{w} pertains to the trained weights).
Step 3. Assign \mathbf{x}_{test} to k , where $k = \arg \max i (\mathbf{w}_i^T \mathbf{x}_{test})$.

Step 4. Cycle through the testing data set (which is suitably divided for cross-validation).

Step 5. Compute the number of correct C_i .

Step 6. Compute % correct classification.

Step 7. Show outputs with the confusion matrices.

Multi Layer Perceptron (MLP)

Theory

Figure 11 conceptualizes the perceptron network designed for helicopter data. Here the following should be noted:

- x_0 is permanently set to -1.0 for the bias parameter w_0 .
- The weight vector \mathbf{w} exists for each class C_k and is a 7 dimensional vector.
- $\Sigma = \mathbf{w}^T \mathbf{x}$ (inner product that serves as the argument to the activation function).
- Activation Function $g(a)$: the logistic sigmoid function such that
 $g(a) = 1$ when $a \geq 0$
 $g(a) = 0$ when $a < 0$

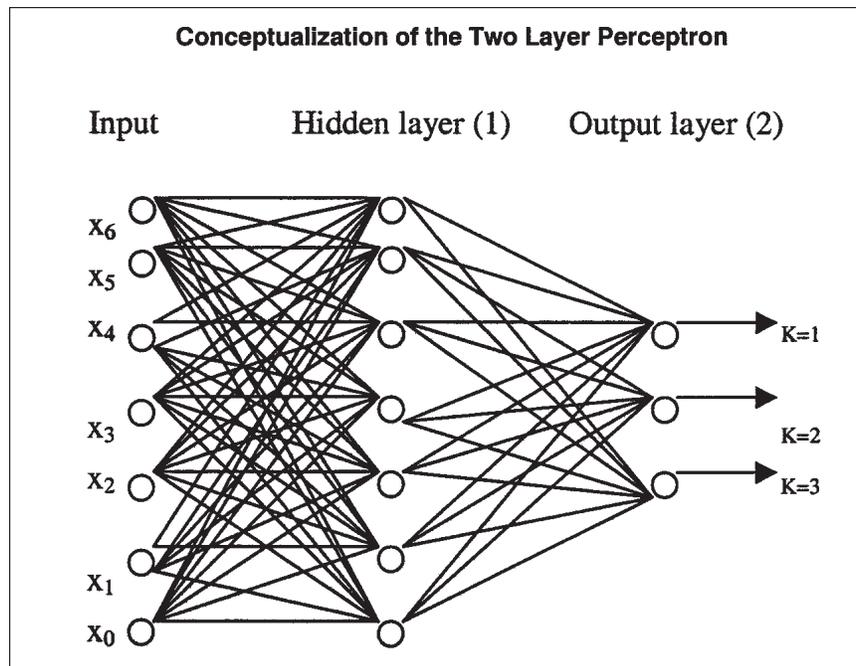


Figure 11. A two layered perceptron network.

- $w_{ji}^{(L)}$ refers to the weight in layer (L) connecting the i th node of the (L-1)th layer to the j th node of the Lth layer.
- The same number of hidden units (7) were chosen for the hidden layer.

Algorithm: (Back propagation)

(Training of Weights)

Step 1. Initialize w_{ji}^L —the weight vector for each class C_j of layer L using the method of Russo.

Step 2. Start iteration, $i = 1, 1000$ (say)

Step 3. Forward Propagation:

- Pick a training data $\mathbf{x}_{\text{train}}$ (having shuffled data)
- Compute a_j at hidden layer for each of the hidden nodes as $a_j = \sum w_{ji}^{(1)} x_j$
- Compute activation $g(a_j)$ for each node as,

$$g(a_j) = \frac{1}{1 + e^{-a_j}}$$

- Set $z_j = g(a_j)$ at the hidden layer
- Compute a_k for each output node as $a_k = \sum w_{ji}^{(2)} z_i$
- Compute the output activation y_k using the activation function as in step iii). ($y_k = g(a_k)$)
- Compute the error signal e_k for each output node as $e_k = y_k - t_k$ where t_k is the target value for class k (i.e., if $\mathbf{x}_{\text{train}}$ belongs to class k then t_k is 1.0. It is zero for the other classes). Summing up squares of e_k over k and then halving gives the value of the error function E at iteration i .

$$E = \frac{1}{2} \sum_{k=1}^3 (y_k - t_k)^2$$

Step 4. Backward propagation (Computing local gradients at each node)

- Compute δ_k at each output node. It is the same as the error signal e_k .
- Back propagate δ s to the hidden layer as

$$\delta_j = z_j (1 - z_j) \sum_{k=1}^3 w_{kj} (2) \delta_k$$

- Compute the derivative of the error function for the 1st and 2nd layer as

$$\frac{\partial E}{\partial w_{ji}} = \delta_j x_i$$

$$\frac{\partial E}{\partial w_{kj}} = \delta_k z_j$$

- iv) Update the weights by an increment (at each layer) Δw_{ji} as $\Delta w_{ji} = -\eta \delta_j x_i$ and $\Delta w_{kj} = -\eta \delta_k z_j$

OR apply a suitable gradient scheme (conjugate gradient, incremental gradient or memory less quasi-Newton method) to update the weights

Step 5. Repeat steps 3 to 4 till the error function minimizes.

APPENDIX D Least Means Square Algorithm

Least Mean Square Algorithm (Sequential or Online version)

Assumptions:

1. Successive input vectors $\mathbf{x}(1)$, $\mathbf{x}(2)$, $\mathbf{x}(N)$ are statistically independent of each other.
2. At time step n , the input vector $\mathbf{x}(n)$ is statistically independent of all previous samples of the desired response, i.e., the target vector $\mathbf{t}(1)$, $\mathbf{t}(2)$, $\mathbf{t}(n-1)$.
3. At time step n , the desired response (or target) $\mathbf{t}(n)$ is dependent on $\mathbf{x}(n)$, but statistically independent of all previous values of the targets.
4. The input vector $\mathbf{x}(n)$ and desired response $\mathbf{t}(n)$ are drawn from a Gaussian distributed population.

The Algorithm:

Training

Step 1. Initialize weight vector for each class k $\mathbf{w}_k(0)$ (at time step 0).

Step 2. Start training iteration index, $i = 1, 1000$ (say)

Step 3. Cycle through the training data, by picking $\mathbf{x}(n)$, $n=1, 2, \dots, N$.

Step 4. For each class k , compute the error $e_k(n) = t_k(n) - \mathbf{w}_k^T(n)\mathbf{x}(n)$

Where $t_k(n)$ is the desired response class k , taken as 1 (for the logistic sigmoidal activation function) if $\mathbf{x}(n)$ belongs to class k and 0 if otherwise.

Step 5. Update weights as,

$$\mathbf{w}_k(n+1) = \mathbf{w}_k(n) + \eta \mathbf{x}(n) e_k(n)$$

Compute total error for each class $E_k(n) = \sum e_k(n)$

Step 6. Repeat Steps 3-5 for the next input pattern and continue till error becomes minimum.

Testing (with cross validation)

Step 7. Pick a testing data $\mathbf{x}_{\text{test}}(n)$ and evaluate for each class k , $\mathbf{w}_k^T(n)\mathbf{x}(n)$.

Step 8. Classify $\mathbf{x}_{\text{test}}(n)$ as belonging to class k where,

$$k = \arg \max (\mathbf{w}_k^T(n)\mathbf{x}(n))$$

Step 9. Compute % correct classification for each class k .

Note: To ensure convergence in training the weights, the learning parameter was chosen as,

$$0 < \eta < 2/(\text{sum of mean square values of input patterns}) \text{ or}$$

$$0 < \eta < 2/\text{Trace}(\mathbf{R}\mathbf{x}) \text{ where } \mathbf{R}(\mathbf{x}) \text{ is the correlation matrix (a } 6 \times 6 \text{ matrix)}$$

Weights were initialized by the thumb rule provided by Russo [18].

REFERENCES

1. C. R. Williams and W. L. Ecklund, Classification of Precipitation Clouds in the Tropics Using 915-MHz Wind Profilers, *Journal of Atmospheric Oceanic Technology*, 12, pp. 996-1012, 1995.
2. V. Ramanathan, The Greenhouse Theory of Climate Change: A Test by an Inadvertent Global Experiment, *Science*, 250, pp. 293-299, 1988.
3. F. Aviolat, T. Cornu, and D. Cattani, Automatic Cloud Observation Improved by Artificial Neural Network, *Journal of Atmospheric Oceanic Technology*, 15, pp. 114-126, 1998.
4. J. Lee, R. C. Weger, S. K. Sengupta, and R. M. Welch, A Neural Network Approach to Cloud Classification, *IEEE Transcripts Geoscience and Remote Sensing*, 28:5, pp. 846-855, 1990.
5. C. Kummerow, W. Barnes, T. Kozu, J. Shiue, and J. Simpson, The Tropical Rainfall Measuring Mission (TRMM) Sensor Package, *Journal of Atmospheric Oceanic Technology*, 15, pp. 809-817, 1998.
6. B. Nicolin and Gabler, A Knowledge-Based System for the Analysis of Aerial Images, *IEEE Transcripts Geoscience and Remote Sensing*, GE-25:3, pp. 317-328, 1987.
7. N. D. Ritter, T. L. Logan, and N. A. Bryant, Integration of Neural Network Technologies with Geographic Information Systems, *GIS Symposium Integrating Technology and Geoscience Applications*, Denver, Colorado, pp. 102-103, December 1988.
8. E. M. Anagnostou and C. Kummerow, Stratiform and Convective Classification of Rainfall Usage SSM/I 85-GHz Brightness Temperature Observations, *Journal of Atmospheric Oceanic Technology*, 14, pp. 570-575, 1997.
9. D. Rosenfield, E. Amitai, and D. B. Wolff, Classification of Rain Regimes by the Three-Dimensional Properties of Reflectivity Fields, *Journal of Applied Meteorology*, 32, pp. 17-39, 1995.
10. S. E. Yuter and R. A. Houze, Jr., Three-Dimensional Kinematic and Microphysical Evolution of Florida Cumulonimbus, Part II: Frequency Distribution of Vertical Velocity, Reflectivity and Differential Reflectivity, *Monthly Weather Review*, 123, pp. 1941-1963, 1995.

11. W. B. Rossow and R. A. Schiffer, ISCCP Cloud Data Products, *Bulletin of the American Meteorology Society*, 72, pp. 2-20, 1991.
12. G. W. Petty, *On the Response of the Special Sensor Microwave/Imager to Marine Environment—Implications for atmospheric Parameter Retrievals*, Ph.D. dissertation, University of Washington, 291 pp., 1990.
13. G. Liu, J. A. Curry, and R.-S. Sheu, Cloud Classification of Clouds Over the Western Equatorial Pacific Ocean Using Combined Infrared and Microwave Satellite, *Journal of Geophysical Research*, 100, pp. 13811-13826, 1995.
14. R. M. Welch, S. K. Sengupta, A. K. Goroch, P. Rabindra, N. Rangaraj, and M. S. Navar, Polar Cloud and Surface Classification Using AVHRR Imagery: An Interpolation of Methods, *Journal of Applied Meteorology*, 31, pp. 405-420, 1992.
15. R. L. Bankert, Cloud Classification of AVHRR Imagery in Maritime Regions Using a Probabilistic Neural Network, *Journal of Applied Meteorology*, 33, pp. 909-918, 1994.
16. S. Haykin, *Neural Networks*, Prentice Hall, New York, 1999.
17. C. M. Bishop, *Neural Networks for Pattern Recognition*, Oxford.
18. A. P. Russo, Neural Networks for Sonar Signal Processing, Tutorial No. 8, *IEEE Conference on Neural Networks for Ocean Engineering*, Washington, D.C., 1991.

Direct reprint requests to:

Faisal Hossain
261 Glenbrook Road, U-2037
University of Connecticut
Storrs, CT 06269-2037