Machine Learning, Deep Learning and Possibilities for Radar Engineering

Amit Kumar Mishra
Department of Electrical Engineering
University of Cape Town
South Africa
akmishra@ieee.org

Abstract **There are two paradigms of data processing, viz. Discriminative and Generative. Of late there has been a lot of work in the domain of Discriminative Models especially with the advent of deep learning networks. In this work we briefly introduce the two models and discuss the use of the discriminative modelling based algorithms for radar data processing.**

Keywords—ANN, DNN, SVM, Discriminative model, Generative model, NCTR, Bistatic Sea Clueeter

I. Introduction

Data processing is one of the major components of radar system design. A major effort from any radar engineering firm is spent in processing the data so as to extract *information of value* from that. Mostly, till the date, radar data or signal processing has followed a *Generative Model*.

A generative model is one in which we follow the classic Bayes way of looking at data and extracting information from the same. This is done as follows. We want to know a certain information which is embedded in the parameter θ of the data distribution. So the data distribution can be modelled as $P(X|\theta)$, where X is the data observed. This is something we can measure from experiments. However this is not something we need in a real-life situation. In a practical application we need to find $P(\theta|X)$. This is where we use Bayes rule to links these two, viz. the function we want to find and the pdf that we can measure.

$$P(\theta|X) = P(X|\theta)*P(\theta)/P(X)$$

This has been the major way to extract information and to make decisions in radar systems. For example a wealth of sea clutter data has been collected to estimate $P(X|\theta)$ so that $P(\theta|X)$, where θ mostly represents the case of the presence of a target.

As opposed to the generative model, there is the competing *Discriminative Model* which endeavours to estimate $P(\theta|X)$ directly. This uses the wealth of algorithms known as Machine Learning to do the job. In the last decade machine learning, especially the one driven by artificial neural networks (ANN) have been proved to be very successful. This has come mainly after the introduction of the new family of ANN called deep neural networks (DNN). This became more of a main-stream after the introduction of powerful tools to design DNNs. E.g. Tensorflow [1], Torch [2], Caffe [3] etc. There has been some works on the application of DNN and machine learning tools in radar signal processing. The major focus of this work is to reinforce the concept of machine learning tools as "discriminative models". Once this is grasped properly the user can think in a completely different light.

We shall show the performance of machine learning tools in two applications, viz. in modelling sea-clutter data and in recognising targets from SAR images (SAR based NCTR).

II. Bistatic Seaclutter suppression using machine learning

Detecting small targets in the presence of sea clutter is a typical maritime radar problem. The nature of sea clutter returns can be very close to that of returns from a desired target. Without further processing, this would result in either a low probability of target detection or unacceptable probability of false alarm. This problem is commonly solved by applying CFAR (Constant False Alarm Rate) processing which is an adaptive process that determines a local optimum threshold level which improves the probability of target detection and reduces the probability of false alarm.

A great deal of investigation and modelling of sea clutter statistics have been done by Ward et al. [4], amongst others. Ward et al. have proposed a compound statistical model for sea clutter that characterises the statistics of different scattering mechanisms. In [4], two distinct groups of scattering mechanisms are identified. The first group is the largescale sea wave structure involving long wavelengths relative to the resolution of the radar system. This resolved sea structure results in spatial variations in mean sea clutter return. The second group consists of small scale structures, that include white caps (breaking waves), Bragg scattering from small scale capillary waves and spikey specular returns from cresting waves shortly before they break. The second group of mechanisms has a Gaussian distribution while the first group is modelled by a Gamma distribution. The Gamma distributed first group clutter modulates the mean power level of the Gaussian distributed second group clutter return. The resultant overall sea clutter distribution model is called K distribution.

In our work we explore a novel approach to sea clutter suppression by treating it as a machine learning problem. In this we compare the use of two machine learning techniques to model the classification boundary between sea clutter and small maritime targets. A comparison of the success of different machine learning algorithms has been conducted by [5] for atmospheric radar. The clutter types requiring suppression from [5] were ground clutter, sea clutter and anomalous propagation. In this study it was found that SVM and other machine learning algorithms outperformed kNN by a significant margin. It was also noted however that kNN was not suited to the classification problem due to a low number of features. In this paper however, the machine learning algorithms made the class distinction between clutter and target based on a high number of features. Other research papers have explored the alternative use of machine learning algorithms for clutter suppression [6], [7], [8], but not in the context of small maritime target detection.

The radar data was obtained from the experimental radar called NetRAD which was set up to collect maritime radar data over several days from a littoral zone near Simon's Town on the Cape Peninsula. NetRAD is a multistatic radar system consisting of a transceiver node and several receiver nodes. The radar operates at 2.4 GHz with a bandwidth of 50 MHz and is used to detect maritime targets. An illustration of the position of NetRAD for the datasets used is presented in Fig. 1. For this paper, only data received at the transceiver node was used and therefore the configuration is effectively monostatic.

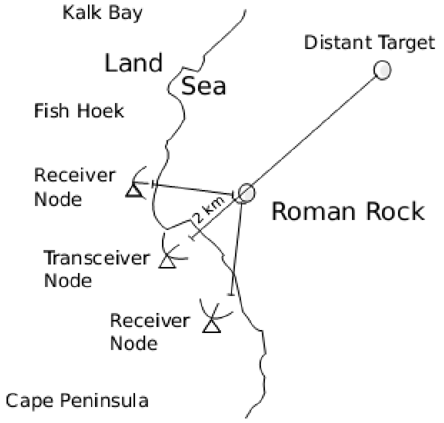

Figure 1. Illustrative position of NetRAD

The transmitted chirp is selected from a set of reference chirps ranging from 0.1 micros to 10 micros. For all the datasets used the sea state was 1 and the look angle was between 78 degrees and 94 degrees. The antenna was set up on the beach at shoulder height with 0 degree elevation resulting in a near 0 degree grazing angle. Each node records at 100 Mega-samples per second. An example of NetRAD data after range pulse compression is shown in Fig. 2 where there are clearly two targets, one stationary target and one moving target. There is also typical substantial close-range clutter on the far left of Fig. 2.

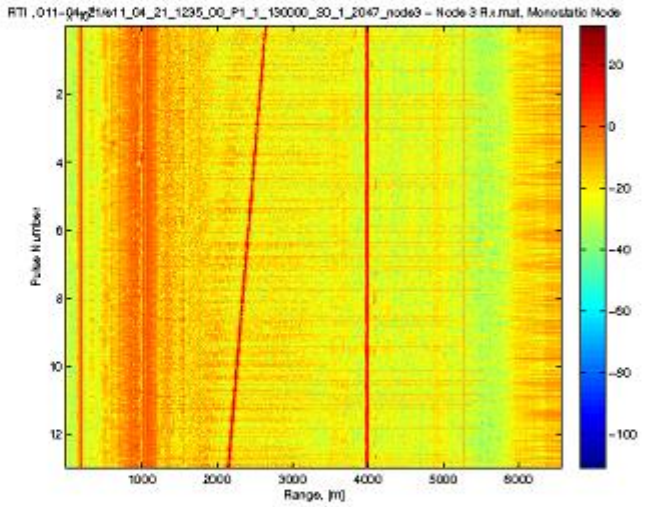

Fig. 2: Pulse compressed data from NetRAD operating in Simon's Town on the 21st of April 2011 at 12:35 SAST. The x axis is range (fast time) and the y axis is slow time. Note the solid red vertical line in the middle of the figure is Roman Rock and the weaker diagonal red line is a moving vessel. The red bands on the far left are close-range clutter returns

Feature Selection

There are two processes that are applied. The first process is a set of training exercises in which typical datasets are used where the two classes are pre-identified to the process. The feature values at the boundary are the decision thresholds for discriminating between the classes. A number of training exercises may be applied in order to reach a final boundary definition.

The second process is to apply the boundary to a dataset in which the classes have not been identified to the process. The class identification of the feature blocks may still however be known and the result is used as a validation test. Cross validation technique called leave-p-out was used to divide up the available dataset into training and validation tests.

Results

In presenting the results we follow a more pithy way of presenting them. Instead of just showing the probability of correct detection we show the probability of correct detection with the decrease in the amount of training data. Tables I and II show the results for SVM and kNN (for k = 3).

% data removed	P_d	P_{fa}
95%	0.850	0.507
75%	0.777	0.465
50%	0.740	0.505
0%	0.743	0.502

Table I. Summary of results from SVM classifier tests.

% data removed	P_d	P_{fa}
95%	0.920	$<1\times10^{-5}$
75%	0.910	2.16×10^{-5}
50%	0.901	1.07×10^{-5}
0%	0.900	1.07×10^{-5}

TABLE II: Summary of results from kNN classifier tests for differ amounts of data removed

III. AUTOMATIC TARGET RECOGNITION USING DEEP NN

ATR has been one of the active research areas for past few decad In this part we focus on the application of the current-age neu networks which are called deep neural networks (DNN) for the ta of synthetic aperture radar based ATR. We have used the clas MSTAR database here to validate the results.

Figure 3 shows the steps we take to process the data. We use a ra of machine learning algorithms. Nearest Neighbour, C4.5, Grafted C4.5, PART and Random Forest where implemented using the W framework. MLP and CNN were implemented using DI framework. All programs used Java 8.

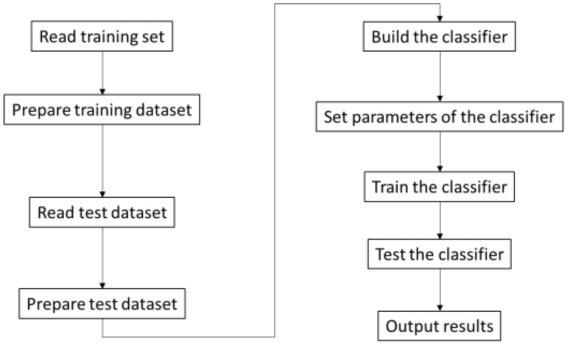

Figure 3 General flowchart of the methodology

We are using Nearest Neighbour as our benchmark and therefore we used the simplest implementation of this. We used the implementation of taking only the closest training data to test data. This means there are no parameters that need to be set and this method is very simple to understand and implement.

C4.5, Grafted C4.5 and PART are all mutations of each other and only require one parameter, condence factor, to be set. We used three values for the condence factor, 0.1, 0.3 and 0.5. We ran a program for each classifier with each condence factor on each sub-dataset.

The Random Forest classifier uses on parameter which is the number of trees. More trees leads to better performance but computation time is increased with diminishing returns. We ran the classifier on each subdataset a number of times increasing the number of trees from 10 to 300 increasing in step sizes of 10 trees.

The multi-layer perceptron had 3 varying parameters so resulted in many permutations. The hidden neuron count was varied from 10 to 45 increasing in steps of 5. The momentum was set to either 0, 0.1 or 0.2 and the learning rate varied from 0.01 to 0.31 with step sizes of 0.075. We used Stochastic Gradient Descent(SGD) as the optimization algorithm, a l2 with the value of the product of the learning rate and 0.005. The weight initialization was done using Xavier. We used on hidden layer and used 200 epochs for each parameter.

The main performance change we wanted to measure using the Convolutional Neural Network(CNN) was using a number of different updaters. The updaters used was Stochastic Gradient Descent(SGD), Nesterovs, RMS Propagation(RMSProp) and Ada Delta. The structure of the CNN was kept the same for all the updaters. The convolutional layers and their parameters for the CNN's that used each pixels on the targets as input can be seen in Table 3 and the layers and parameters for the CNN's that used features as inputs. The 'features as input' networks required batch processing and more layer because there was less(summarised) data for the network to train on. This results in over fitting and a much more sensitive dataset. In the table inputs refers to the third dimension of the data. Our input is one on the first layer since we are using 2 dimensional data. The output is the input of the next layer. We used batch learning to have a manageable amount of data per epoch and prevent over-fitting. We trained in batches of 95. The number of epochs used was 200.

Layer Number	Layer Type	Size	Stride	Padding	Inputs	Outputs
1	Convolution	2x2	1x1	3x3	1	19
2	Sub-sampling	2x2	1x1	0	19	19
3	Convolution	4x4	2x2	0	19	40
4	Sub-sampling	2x2	1x1	0	40	40
5	Dense	n/a	n/a	n/a	40	50
6	Output	n/a	n/a	n/a	50	5

Table 3: Convolutional layers and their parameters with pixels as inputs

Results:

We recorded the training and testing times of each classifier. The results are shown in Table 4. We also display them as a ratio to the benchmark Nearest Neighbour classifier which allows us to understand the training times when comparing them to one another. It should also be noted that the classifiers Multilayer Perceptron and Convolutional Neural Network run a number epochs that is specified by the user.

Figure 4 shows the result of the different classifiers we have used when we have used the full amount of training dataset with us.

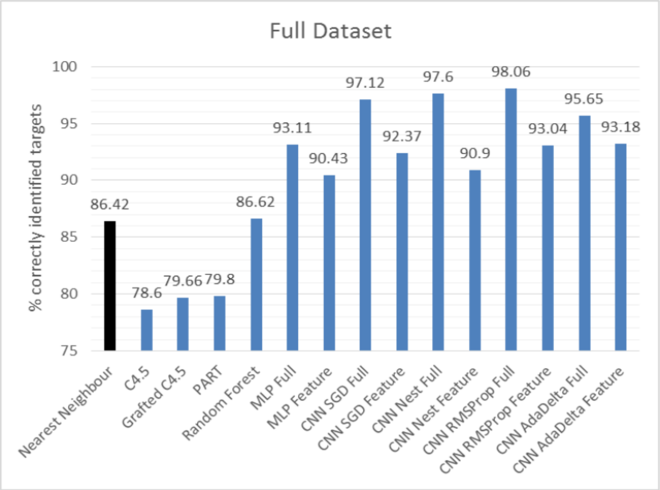

Figure 4 Percentage of correct classification with different machine learning algorithms (especially some of the powerful deep-learning algorithms)

IV. CONCLUSIONS

In this paper we have shown some of the uses of machine learning algorithms in the domain of radar. The main aim of the work has been to thrust on the fact that there is an alternate way to process radar signal. This alternate way known as discriminative modelling attempts to predict the posterior probability without following Bayes rule. This is data-driven and has been in use for a while. However with the advent of deep learning and powerful tools the use of discriminative models will become more possible and prevalent in the coming times. We have shown the use of these models for two of the challenging tasks in radar signal processing, viz. small boat detection in heavy (bistatic) sea clutter and automatic target recognition from SAR images.

REFERENCES

[1] https://www.tensorflow.org/

[2] torch.ch

[3] caffe.berkeleyvision.org

[4] K. D. Ward, S. Watts, and R. J. Tough, Sea clutter: scattering, the K distribution and radar performance. IET, 2006, vol. 20.

[5] T. Islam, M. A. Rico-Ramirez, D. Han, and P. K. Srivastava, “Artificial intelligence techniques for clutter identification with polarimetric radar signatures,” Atmospheric Research, vol. 109, pp. 95–113, 2012.

[6] R. Vicen-Bueno, R. Carrasco-Alvarez, M. Rosa-Zurera, J. C. Nieto-Borge, and M.-P. Jarabo-Amores, “Artificial neural network-based clutter reduction systems for ship size estimation in maritime radars,” EURASIP Journal on Advances in Signal Processing, vol. 2010, no. 1, p. 1, 2010.

[7] G. Mountrakis, J. Im, and C. Ogole, “Support vector machines in remote sensing: A review,” ISPRS Journal of Photogrammetry and Remote Sensing, vol. 66, no. 3, pp. 247–259, 2011.

[8] P.-L. Shui, D.-C. Li, and S.-W. Xu, “Tri-feature-based detection of floating small targets in sea clutter,” IEEE Transactions on Aerospace and Electronic Systems, vol. 50, no. 2, pp. 1416–1430, 2014