6-20-2004

The use of machine learning in smart antennas

Chaouki T. Abdallah

Recommended Citation

Abdallah, Chaouki T.. "The use of machine learning in smart antennas." *IEEE Antennas and Propagation Society International Symposium* (2004): 321-324. doi:10.1109/APS.2004.1329637.

THE USE OF MACHINE LEARNING IN SMART ANTENNAS

***Christos G. Christodoulou[1], Judd A. Rohwer[2], and Chaouki T. Abdallah[1]**
[1] The University of New Mexico
[2]Sandia National Laboratories, Albuquerque, NM

Abstract—The goal here is to make arrays "smart" so that when one of the antenna elements in the array fails, the beamforming and beamsteering performance of the array degrades gracefully. Such an objective can be achieved in reconfiguring the array when an element is found to be defective, by either changing the material properties of the substrate or by applying appropriate loading in order to make the array functional again. Our approach is based on optimization using Machine learning and Support Vector Machines (SVM). The basic idea is to change the excitation coefficient for each array element (magnitude and phase) to optimize for changes due to the environment surrounding an array antenna. Using Support Vector Machines, one can train the antenna array to change its elements phase or excitation distribution in order to maintain a certain radiation pattern or to enhance its beam steering and nulling properties and solve the DOA problem as well.

1. Introduction to Machine Learning and Support Vector Machines (SVM)

Pattern classification is a machine learning process for observing input data and applying classification rules to generate binary or multiclass labels. In the binary case, a classification function is estimated using input/output training pairs with unknown probability distribution, $P(x,y)$, where x is sample vector of observations and y is a machine learning label. Let:

$$(x_1,y_1),\ldots,(x_N,y_N)\in\Re^n\times Y,\quad y_i=\{-1,+1\},i=1,\ldots N \tag{1}$$

The estimated classification function maps the input to a binary output, $f:\Re^n\rightarrow\{-1,+1\}$. The system is first trained with the given input/output data pairs, the sample size is length N, the input vectors are of dimension n. The test data, taken from the same probability distribution $P(x,y)$ is then applied to the classification function. The binary output label, +1, is generated if $f(x)\geq 0$, likewise -1 is the output label if $f(x)<0$. For the multiclass case $Y\in\Re^C$ where Y is a finite set and C is the size of the multiclass label set. The objective is to estimate the function which maps the input data to a finite set of output labels. Since the probability distribution of the input data is unknown, the classification function must be estimated by minimizing the empirical expected risk. The risk is defined as :

$$R(f)=\int L(f(x),y)dP(x,y) \tag{2}$$

and L is the loss function. By setting conditions of the minimization routine and the number of available data, the empirical risk converges towards the expected risk.

0-7803-8302-8/04/$20.00 ©2004 IEEE

1.1 Kernel Functions

Kernel-based machine learning algorithms utilize a projection of the input space to a higher dimensional feature space, F , via a nonlinear mapping,

$$\Gamma : \Re^N \to F \quad \text{and} \quad x \to \Gamma(x) \tag{3}$$

Kernel functions compute the scalar dot products of the input/output pairs in the feature space F. Without kernel functions scalar operations in the higher dimensional feature space would be prohibitively difficult. Essentially, an algorithm in the input space can be applied to the data in the feature space.

$$\Gamma(x) \cdot \Gamma(x_i) = k(x \cdot x_i) \tag{4}$$

A nonlinear algorithm in the input space, such as the classification functions, corresponds to a linear algorithm in the feature space. Figure 1 shows a nonlinear binary classification system that is not separable in the input space, but in the feature space the two classes are linearly separable with the optimal separating hyperplane.

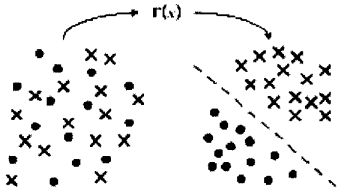

Fig. 1. The data in the input space is not linearly separable. The data in the higher dimensional feature space is linearly separable with the optimal separating hyperplane

1.2 Support Vector Machines, Binary and Multiclass

SVMs are a nonparametric machine learning algorithm with the capability of controlling the capacity through the support vectors. SVMs find a classification function that separates data classes, with the largest margin, using a hyperplane. The difference between all machine learning algorithms for pattern classification is the mathematical operations involved in calculating the optimal separating hyperplane. The data points near the optimal hyperplane are the "support vectors".

In binary classification systems the machine learning algorithm generates the output labels with a hyperplane separation The input sequence and a set of training labels are represented as $\{x_i, y_i\}_{i=1}^{N}, y_i = \{-1,+1\}$, $y_i \in [-1,1]$ represents the classification "label" of the input vector x which has dimension n and sample length N. If the two classes are linearly separable in the input space then the hyperplane is defined as $w^T x + b$, with w being a weight vector perpendicular to the separating hyperplane, and b is a bias that shifts the hyperplane parallel to itself. If the input space is projected into a higher dimensional feature space then the hyperplane becomes $w^T \Gamma(x) + b = 0$

The SVM algorithm is based on the hyperplane definition,

$$y_i \left[w^T \Gamma(x_i) + b \right] \geq 1, i = 1, \ldots, N \tag{5}$$

Given the training sets in equation (1) the binary SVM classifier is defined as

$$y(x) = sign\left[\sum_{i=1}^{N} \alpha_i y_i k(x, x_i) + b\right] \tag{6}$$

The non-zero $\alpha_i^{'}$s are "support values" and the corresponding data points, x_i are the "support vectors". Quadratic programming is one method of solving for the $\alpha_i^{'}$s and b in the standard SVM algorithm.

1.3 Least Squares Support Vector Machines, LS-SVM

Using the LS-SVM version of equation (6) allows its use for non-binary applications. The LS-SVM classifier is generated from the optimization problem:

$$\min_{w,b,\phi} L_{LS}(w,\phi) = \frac{1}{2}\|w\|^2 + \frac{1}{2}\gamma\sum_{i=1}^{n}\varphi_i^2 \tag{7}$$

γ and φ_i are the regularization and error variables, respectively. The minimization in equation (6) includes the constraints

$$y_i\left[w^T\Gamma(x_i) + b\right] = 1 - \phi_i, i = 1,\ldots,n \tag{8}$$

The LS-SVM includes one universal parameter, γ, that regulates the complexity of the machine learning model. This parameter is applied to the data in the feature space, the output of the kernel function. A small value of γ minimizes the model complexity, while a large value of γ promotes exact fitting to the training points. The error variable φ_i allows misclassifications for overlapping distributions. The Lagrangian of equation is defined as

$$Z_{LS}(w,b,\phi,\alpha) = L_{LS}(w,b,\phi) - \sum_{i=1}^{n}\alpha_i\left\{y_i\left[w^T\Gamma(x_i) + b\right] - 1 + \phi_i\right\} \tag{9}$$

where α_i are the Lagrangian multipliers.

2. How they are Applied to Array Antennas.

The aim is to cast the array antenna problem into the form of equation (9) and then train the LS-SVM algorithm. For example, for the direction of arrival (DOA) problem, the LS-SVM algorithm is trained with projection vectors generated from the signal subspace eigenvectors and the respective covariance matrices. This takes into consideration the number of antenna elements, the element separation, Doppler shift, number of incident signals, and it can also be trained to take into consideration any terrain or platform changes on which the antenna elements reside on. The training can be done for any kind of communication system such as CDMA, FDMA or TDMA. The output labels from the multiclass LS-SVM system are the DOA estimates. Figure 2, depicts an example of DOAs evaluated using machine learning and its comparison with the MUSIC algorithm [1-2].

Beam steering and power control [3] of the antenna elements can also be handled for various communication channels. In every situation the idea is to find the optimal hyperplane that will allow us to separate the desired input features to classify and train the SVS algorithm.

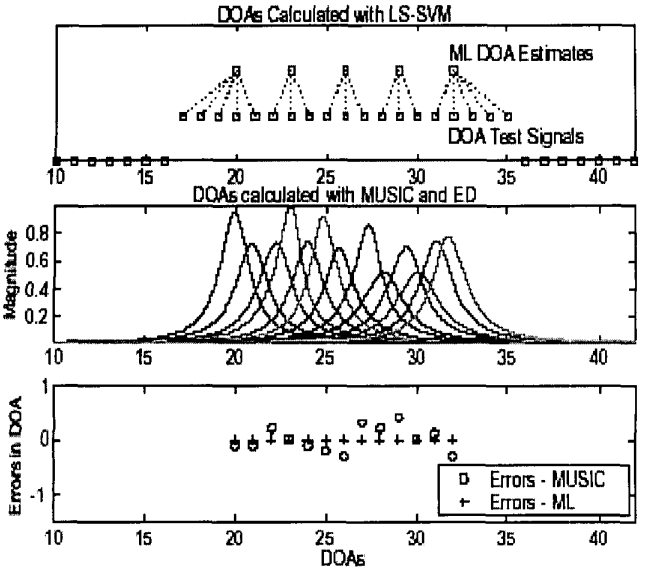

Fig. 2. DOA estimation using the LS-SVM and the MUSIC algorithms. The LS-SVM DOA estimation algorithm includes five classes and a three degree DOA range.

Conclusions. As array antennas are placed on complex surfaces and structures, more and more computational capabilities are developed to handle the demand of analyzing such antennas. However, once the antenna is built and placed on a certain platform, changes that affect both the shape of the structure and the effectiveness of the materials used to fabricate the antenna may occur. Machine learning is an approach that can handle some of these complexities in real-time fashion. An example was presented with a multiclass LS-SVM architecture for DOA estimation

References

1) Judd A. Rohwer and Chaouki T. Abdallah, "One-vs-One Multiclass Least Squares Support Vector Machines for Direction of Arrival Estimation", ACES Journal: Special Issue on Neural Network Applications in EM. vol. 18, no. 2, pp. 34-45, July 2003.

2) S. Chen, A.K. Samingan, and L. Hanzo, "Support Vector Machine Multiuser Receiver for DSCDMA Signals in Multipath Channels", IEEE Transactions On Neural Networks, vol. 12, no. 3, pp. 604-611, May 2001.

3) Johan A.K. Suykens, "Support Vector Machines: A Nonlinear Modelling and Control Perspective", European Journal of Control, vol 7, pp. 311-327, 2001