6-22-2003

Least squares support vector machines for direction of arrival estimation

Chaouki T. Abdallah

Judd A. Rohwer

Christos G. Christodoulou

Least Squares Support Vector Machines for Direction of Arrival Estimation

Judd A. Rohwer[1], Chaouki T. Abdallah[2], and *Christos G. Christodoulou[2]
[1]Sandia National Laboratories, Albuquerque, NM, jarohwe@sandia.gov
[2]Department of Electrical and Computer Engineering, University of New Mexico, Albuquerque, NM, {chaouki,christos}@eece.unm.edu

1 Introduction

Machine learning research has largely been devoted to binary and multiclass problems relating to data mining, text categorization, and pattern/facial recognition. Recently, popular machine learning algorithms, including support vector machines (SVM), have successfully been applied to wireless communication problems, notably spread spectrum receiver design [1], channel equalization [2]. This paper presents a multiclass implementation of SVMs for direction of arrival (DOA) estimation.

2 Least Squares Support Vector Machines

In a binary classification system the input sequence and a set of training labels are represented as $\{\mathbf{x}_k, y_k\}_{k=1}^{K}$, where $y_k = \{-1, 1\}$ represents the classification "label" of the input vector $\mathbf{x}_k$. If the two classes are linearly separable in the input space then the hyperplane separating the classes is defined as $\mathbf{wx}+b = 0$, $\mathbf{w}$ is a vector of weights and b is a bias term. If the input space is projected to a higher dimensional feature space then the hyperplane becomes $\mathbf{w\Gamma}(\mathbf{x}) + b = 0$, the nonlinear function $\mathbf{\Gamma}(\cdot)$ maps the input space to the feature space.

Suykens, et.al., [3] introduced a least squares SVM (LS-SVM) which is based on the Vapnik SVM classifier,

$$y(\mathbf{x}) = sign\left[\sum_{k=1}^{K} \alpha_k y_k \mathbf{\Gamma}(\mathbf{x}, \mathbf{x}_k) + b\right]. \quad (1)$$

The LS-SVM classifier is generated from the optimization problem:

$$\min_{\mathbf{w},b,\phi} \mathcal{L}_{LS}(\mathbf{w},b,\phi) = \frac{1}{2}\|\mathbf{w}\|^2 + \frac{1}{2}\psi\sum_{k=1}^{K}\phi_k^2, \text{ with constraints} \quad (2)$$

$$y_k\left[\mathbf{w}^T\varphi(\mathbf{x}_k) + b\right] \geq 1 - \phi_k, \; k = 1, \ldots, K, \quad (3)$$

Misclassifications, due to overlapping distributions, are accounted for with the slack variables ϕ_k, ψ is a regularization parameter that governs the complexity of the SVM, and the margin between the hyperplane and the data points in the feature space is maximize when $\mathbf{w}$ is minimized and the relationship $\varphi(\mathbf{x}_k)^T \varphi(\mathbf{x}_k) = \mathbf{\Gamma}(\mathbf{x}, \mathbf{x}_k)$ is due to Mercer's Theorem [4]. The Lagrangian of equation (2) is defined as

$$\mathcal{Z}_{LS}(\mathbf{w},b,\phi,\alpha) = \mathcal{L}_{LS}(\mathbf{w},b,\phi) - \sum_{k=1}^{K}\alpha_k\left\{y_k\left[\mathbf{w}^T\mathbf{\Gamma}(x_k) + b\right] - 1 + \phi_k\right\} \quad (4)$$

where α_k are Lagrangian multipliers, also known as "support vectors", that can either be positive or negative.

One-vs-one multiclass classification is based on binary LS-SVMs. For P distinct classes there are $\frac{P(P-1)}{2}$ hyperplanes that separate the classes with maximum margin. The hyperplanes with maximum margin are constructed in the LS-SVM training phase. The Decision Directed Acyclic Graph

Sandia is a multiprogram laboratory operated by Sandia Corporation, a Lockheed Martin Company, for the United States Department of Energy under Contract DE-AC04-94AL85000.

(DDAG) is a specific technique for one-vs-one multiclass classification [5]. The technique uses a tree structure to compare the test data to each of the hyperplanes. Through a series of elimination steps the best label is assigned to the input data. The LS-SVM algorithm for DOA estimation is based on the DDAG architecture with each node containing a binary LS-SVM classifier of the i^{th} and j^{th} classes, see Figure 1.

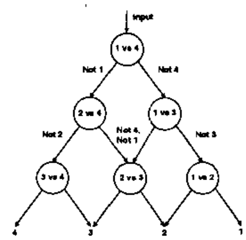

Figure 1: DAGSVM for Four Classes

3 SVMs and DOA Estimation

The process of DOA estimation is to monitor the outputs of $\mathbb{D}$ antenna elements and predict the angle of arrival of $\mathbb{L}$ signals, $\mathbb{L} < \mathbb{D}$. The output vector for each incident signal from the antenna elements is $\overline{\mathbf{a}}(\theta_l) = \begin{bmatrix} 1 & e^{-jk_l} & \dots & e^{-j(\mathbb{D}-1)k_l} \end{bmatrix}^T$, and the vector of incident DOAs is $\boldsymbol{\theta} = \begin{bmatrix} \theta_1, & \dots & ,\theta_{\mathbb{L}} \end{bmatrix}$. With a training process the learning algorithms generate DOA estimates based on the responses from the antenna elements, $\overline{\mathbf{a}}(\theta_l)$.

For the LS-SVM based approach to DOA estimation the output of the receiver is used to calculate the sample covariance matrix $\widehat{\mathbf{R}}_{rr}$ of the input data signal $\mathbf{x}_r(k)$, $\widehat{\mathbf{R}}_{rr} = \frac{1}{\mathbb{M}} \sum_{k=K-\mathbb{M}+1}^{K} \mathbf{x}_r(k)\, \mathbf{x}_r^T(k)$. The dimension of the observation matrix is $\mathbb{D} \times \mathbb{M}$, $\mathbb{M}$ is ideal sample size (window length), and the dimension of the sample covariance matrix is $\mathbb{D} \times \mathbb{D}$. The principal eigenvectors, $\mathbf{v}_1, \dots, \mathbf{v}_{\mathbb{D}}$, are calculated via eigen decomposition (ED) or subspace tracking techniques. Each eigenvector is used to calculate a covariance matrix, $\widehat{\mathbf{R}}_{vv_1}, \dots, \widehat{\mathbf{R}}_{vv_{\mathbb{D}}}$.

The LS-SVM DOA estimation algorithm includes preprocessing, training, and testing steps.

- Preprocessing for SVM Training

  1) Generate the $\mathbb{D} \times N$ training vectors for the P SVM classes, $\mathbb{D}$ is the number of antenna elements, N is the number of input data samples. 2) Generate the P sample covariance matrices, $\mathbb{C}$,with $\mathbb{M}$ samples from the $\mathbb{D} \times N$ data vector 3) Calculate the signal eigenvector, $\mathbb{S}$, from each of the P sample covariance matrices. 4) Calculate the $\mathbb{D} \times 1$ projection vectors, $\mathbb{C} \cdot \mathbb{S}$, for each of the P classes. 5) Store the projection vectors for the training phase and the eigenvectors for the testing phase.

- LS-SVM Training

  1) With the P projection vectors train the $\frac{P(P-1)}{2}$ nodes with the one-vs-one LS-SVM algorithm. 2) Store the LS-SVM variables, α_k and b from equation (1).

- Preprocessing for SVM Testing

  1) Acquire $\mathbb{D} \times N$ input signal from antenna array. 2) Generate the sample covariance matrix with $\mathbb{M}$ samples from the $\mathbb{D} \times N$ data vector. 3) Calculate the eigenvectors for the signal subspace and the noise subspace. 4) Generate the covariance matrices for each eigenvector

- LS-SVM Testing for the i^{th}/j^{th} DDAG Node.

  1) Calculate two $\mathbb{D} \times 1$ projection vectors with the desired eigenvector covariance matrix and the i^{th} and j^{th} eigenvectors from the training phase. 2) Test both projection vectors against the LS-SVM hyperplane for the i^{th}/j^{th} node. 3) Calculate the mean value of the two LS-SVM output vectors (labels). Select the mean value that is closest to a decision boundary, 0 or 1. Compare this value to the label definition at the DDAG node, then select the proper output label. 4) Repeat process for the next DDAG node in the evaluation path or declare the final DOA label.

- LS-SVM DDAG Error Control

  1) Review the MSE calculations for the DDAG evaluation path. 2) Apply error control and validation measures to classify the label as either an accurate DOA estimate or as NOISE.

3.1 Simulation Results

Simulations of the LS-SVM DDAG DOA estimation algorithm are based on a complex system model that includes amplitude and phase distributions representative of the communication channel. The received signal at the receiver is modeled as

$$\mathbf{x}_r(t) = \sum_{d=1}^{\mathbb{D}} \sum_{l=1}^{\mathbb{L}} \overline{\mathbf{a}}(\theta_l)\, \alpha_{dl} s_d(t - \tau_{dl}) \cos(w_c(t - \tau_{dl})) + n_d(t). \tag{5}$$

This model includes $\mathbb{D}$ antenna array elements with steering array vector $\overline{\mathbf{a}}(\theta_l)$ and additive Gaussian noise $n_d(t)$. In addition, the model assumes $\mathbb{L}$ independent, resolvable signal paths. The multipath variable α_{dl} is defined as $\alpha_{dl} = \rho_{dl} e^{j(2\pi f_c(t-\tau_{dl})+\phi_c)}$. The amplitude of the received signal ρ_{dl}, includes the transmitted power $\sqrt{p_t(t)}$ and the attenuation due to the link gain and shadowing q^l. This variable is modeled as a fixed, Rayleigh, Ricean, or log-normal distributed random variable. The Doppler shift for each resolvable path is defined by $f_c = \frac{v_c w_c}{2\pi c}$; v_c is the velocity of the mobile in $\frac{m}{\sec}$, w_c is the carrier frequency, and c is the speed of propagation. A uniformly distributed carrier phase shift, ϕ_c, and a time delay for each signal path, τ_{dl}, are also included in the multipath variable, α_{dl}. The CDMA spreading code, $s_d(t - \tau_{dl})$, provides the processing gain at the correlator output.

The antenna array includes eight elements and the training and test signals are the complex outputs from the antenna array. The LS-SVM system includes four DOA classes and six DDAG nodes. Figure 2 shows results for a ten degree range per class. To completely test the LS-SVM DDAG's capabilities the simulations were automated to test a wide range of DOAs. As can been seen from Figure 2 the LS-SVM DDAG DOA estimation algorithm is extremely accurate. No misclassifications were logged. Additional simulations show that the LS-SVM DDAG system accurately classifies the DOAs for three to ten classes and DOA ranges from one degree to twenty degrees.

3.2 Multilabel Capability for Multiple DOAs

In DOA estimation for cellular systems there can be multiple DOAs for a given signal. This results from multipath effects induced by the environment. The machine learning system must be able to discriminate between a small number of independent DOAs that include signal components with similar time delays. With this constraint the machine learning algorithm must be a multiclass system and able to process multiple labels. The machine learning algorithm must generate multiclass labels, $y_i \in \chi$, where $\chi \in [-90, 90]$ is a set of real numbers that represent an appropriate range of expected DOA values, and multiple labels $y_i, i = 1 \ldots \mathbb{L}$ for $\mathbb{L}$ dominant signal paths. If antenna sectoring is used in the cellular system the multiclass labels are from the set $\chi \in [S_i]$, where S_i is field of view for the i^{th} sector. Multilabel classification is possible with the LS-SVM DDAG algorithm presented in Section 3. The LS-SVM algorithm for DOA estimation assigns DOA labels to each eigenvector in the signal subspace. By repeating the DDAG cycle for each eigenvector the multiclass algorithm has the capability of assigning multiple labels to the input signal.

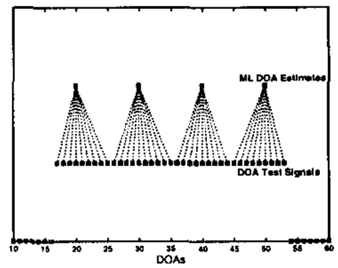

Figure 2: LS-SVM for DOA estimation, four classes with ten degree separation between each.

4 Conclusion

In this paper we presented a multiclass LS-SVM architecture for DOA estimation as applied to a CDMA cellular system. Simulation results show a high degree of accuracy, as related to the DOA classes and prove that the LS-SVM DDAG system has a wide range of performance capabilities. The broad range of our research in machine learning based DOA estimation includes multilabel and multiclass classification, classification accuracy, error control and validation, kernel selection, estimation of signal subspace dimension, and overall performance characterization of the LS-SVM DDAG DOA estimation algorithm.

References

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

[2] Daniel J. Sebald and James A. Bucklew, "Support Vector Machine Techniques for Nonlinear Equalization", IEEE Transactions On Signal Processing, vol. 48, no. 11, pp. 3217-3226, November 2000.

[3] J.A.K. Suykens, L. Lukas, P. Van Dooren, B. DeMoor, and J. Vandewalle, "Least Squares Support Vector Machine Classifiers: a Large Scale Algorithm", ECCTD'99 European Conf. on Circuit Theory and Design, pp. 839-842, August 1999.

[4] J.A.K. Suykens and J. Vandewalle, "Least Squares Support Vector Machine Classifiers", Neural Processing Letters, August 1998.

[5] John C. Platt, Nello Christianini, and John Shawe-Taylor, "Large Margin DAGs for Multiclass Classification", in *Advances in Neural Information Processing Systems*, vol. 12, pp. 547-553, Cambridge, MA, MIT Press, 2000.