Data Based Sparse Learning for Underwater Sensor Array Networks

Shougui Cai
College of Information Science and Electronic Eng.
Zhejiang University
Hangzhou, China
shouguicai@zju.edu.cn

Wen Xu
College of Information Science and Electronic Eng.
Zhejiang University
Hangzhou, China
wxu@zju.edu.cn

ABSTRACT

Developing methods to effectively clean and reduce the amount of data at sensor level is very meaningful for conveniently sharing and transferring information in underwater networks, due to the limitation of bandwidth and power supply. In this paper, we view this issue from the perspective of machine learning and get a sparse representation model by training the feedfoward neural networks. Results on SWellEx96 experiment show that the learned feature space can explain a given 1323-elements sample-covariance matrices with only 11 basis functions, averagely. Simulations show that the neural network model has more ability to resist sound speed profile mismatch, comparing with conventional matched-field processing method.

CCS CONCEPTS

• **Computer systems organization** → **Embedded systems**; *Redundancy*; Robotics; • **Networks** → Network reliability;

KEYWORDS

Array signal processing,machine learning,sparse representation,source localization

ACM Reference format:
Shougui Cai and Wen Xu. 2017. Data Based Sparse Learning for Underwater Sensor Array Networks . In *Proceedings of 12th ACM International Conference on Underwater Networks and Systems, Halifax,NS,Canada, November 2017 (WUWNet'17),* 5 pages.
https://doi.org/10.475/123_4

1 INTRODUCTION

Networked sensor arrays are being increasingly used in underwater source localization[1]. No matter the network is consisted of multiples arrays, or distributed with only single sensors on the arrays, or just networked with only one array[2], a standard approach is bearing estimations at individual arrays firstly, then communication to the fusion center and get the source location at the center[3–6]. However, the accuracy will be worse than optimal solutionwhile the optimal solution involves a centralized and coherent processing of all sensor outputs[7]. But this optimal solution is unattractive, specially in the case of large arrays. Due to the limitation of energy used, it is not realistically to process all the incoming data. In order to improve the accuracy, Wax and Thomas proposed a new scheme for decentralized processing based on communicating the sample-covariance matrices of the sub-arrays to the fusion center[8]. While, accuracy improved, but this method increase the communication load than the standard method. Communication is also a big challenge in underwater networks. Another kind of well studied method for source localization is matched-field processing (MFP)[9–11], but it mostly works for the situation of a single array. Multiple arrays based mfp is an on-going research[12, 13]. Meanwhile, MFP method also needs to process all data, coherently or incoherently.

As the increasing number of manned platforms deploying receiver arrays or fixed/mobiled nodes in distributed underwater sensor networks, we are facing the scene of underwater big data[14]. We want to have a method that can achieve the performace as good as optimal solution, but does not cost too much communication or energy. Now, machine learning technology is more and more mature. As models in machine learning area usually have strong learning and representation ability, we could try to introduce some smart into our system, to extract task-relevant information, discover new patterns and construct a useful representation. It will be very convenient for us to share and transfer information between different platforms once the data can be reduced to small useful finite sets.

2 MACHINE LEARNING BASED DATA SPARSE REPRESENTATION

In this section,we discuss an effective scheme using machine learning based methods to represent the information present in measured data with a sparse format.

2.1 Neural Networks Models and Function Approximation

As we known,neural networks models can be viewed as a mathematical function f. Taking feedforward neural network(FNN) as an example, they define a mapping $y = f(x;\theta)$ between input x and output y by parameter θ, which needed to be learned by a rule. Feedforward networks are called networks

WUWNet'17, November 2017, Halifax,NS,Canada

ACM ISBN 123-4567-24-567/08/06...$15.00
https://doi.org/10.475/123_4

because they are typically represented by composing together many different functions. We might have two function f^1, f^2 connected in a chain[15], to form $f(x) = f^2(f^1(x))$.

FNN extend linear models to represent nonlinear transformed format $\phi(x)$ of input x. The transform function ϕ can be thinked as providing a set of features describing x, or as providing a new representation for x. The key problem here is how to choose the mapping ϕ.

The strategy of machine learning is to learn ϕ. In a feedforward network, ϕ defines a hidden layer $h = \phi(x; w^{(1)})$, then the total model is $y = f(x; \theta) = hw^{(2)}$. Obviously, we need to learn ϕ from a broad class of functions, and parameters w mapping $\phi(x)$ to the desired output.

In most cases, our parametric model defines a distribution $p(y|x; \theta)$. Simply, we can use the principle of maximum likelihood to learn parameters in model.

$$J(\theta) = -E_{x,y\sim p_{data}} \log p_{model}(y|x) \tag{1}$$

where the specific form of p_{model} is defined by networks. As maximum likelihood is a consistent estimator, the model is capable of representing the training distribution.

2.2 Regularization for neural networks

There are two main kind of regularization strategies for neural networks, one is weight-level regularization, another is neuron-level regularization with activation penalty.

$$\tilde{J}(\theta; X, y) = J(\theta; X, y) + \alpha\Omega(\theta) + \beta\Omega(h) \tag{2}$$

where $\Omega(\theta)$ is parameter norm penalty, $\Omega(h)$ is penalty on the activations of the units, α,β are hyper parameters that weight the relative contribution of the norm penalty term. Weight decay term penalize the size of the model parameters, while, the activation penalty term encouraging their activations to be sparse.

2.3 Learn useful sparse representations from data

In practical applications, we not only want the representation to be sparse, but also want the model features to be sparse, the latter saves the storage and calculation on sensor nodes. Thus, we use $L1$ norm to promote sparse neurons activations, and constrain the norm of each column of the weight matrix to prevent any one hidden unit from having very large weights.

$$\begin{aligned} \tilde{J}(\theta) = & -E_{x,y\sim p_{data}} \log p_{model}(y|x) + \lambda\|h\|_1 \\ s.t. \quad & \|W_i\|_2 \le C \quad \forall i = 1, \cdots, M \end{aligned} \tag{3}$$

In the equation, M is the number of neurons in hidden layer. When a useful sparse representation of any given data is learned, each datum will then be encoded as a sparse code, thus we can use the least possible amount of resource to store or transfer the data.

What we need to do here is learning a set of basis functions $\phi(x)$ that make the given data can be represented sparsely. As the neuron networks do, we use a linear model with nonlinear function to fit the basis functions.

$$h = g(Wx + b) \tag{4}$$

where g is the nonlinear function, h is the the desired the sparse data, x is the measures, W and b is the parameters needed to learned.

3 SIMULATION AND EXPERIMENTAL RESULTS

A notable recent example of using machine learning method in underwater acoustic is the application of nonlinear classification to source localization[16]. In this section, we implement a simple FNN with just one hidden layer to learn source range directly from observed acoustic data, and compare the performance of the classifier with the conventional matched-field processing method (Bartlett) in terms of simulation data and experimental data, respectively. In addition, we briefly discuss the influence of sound speed profile(ssp) mismatch on the performance of FNN classifier and improve the tolerance of the classifier by training the model using data sampled under different ssp.

Simulation environment is the widely studied SWell96Ex test, conducting in a 216m deep shallow waveguide environment. The ship proceeded northward at a speed of 2.5 m/s.

During the experiment, the source ship has two sound source, a deep source (J-15) and a shallow source (J-13). In all the simulations, we used the shallow sound source, which was towed at a depth of about 9m and transmitted 9 frequencies between 109Hz and 385Hz.

3.1 Parameter Settings

In simulation part, acoustic data used to train and test the neural network is generated by kraken. Snapshot N_s is 10, number of vertical array elements L is 21, input layer neurons D of FNN is $L^2 \times N_{fre}$(number of frequency used), there is an input data preprocessing here, actual input data is the normalized sample-covariance matrices of measured pressure at each frequency. The number of neurons in the output layer (number of classes) $K = 300$. We just give the same amount of dimensions the original data are encoded in to provide enough dimensions to learn over-complete features at first,which means neurons in hidden layer $M = D$. Specifically,the cost function now becomes

$$\tilde{J}(\theta) = -\frac{1}{N}\sum_{n=1}^{N}\sum_{k=1}^{K} t_{nk} \ln y_{nk} + \lambda\|h\|_1 \tag{5}$$

where t_{nk} and y_{nk} are predictive and real probability of sample data x belongs to class k separately. Moreover we will choose $C = 1$ here. For the sake of learning speed, we use the $ReLU$ activation in hidden layer and use the $softmax$ function in the output layer because this case is a multiple class problem. The training set is 3000 samples of uniform sampling between 1.82-8.65km, test set is another 300 data samples sampling from the same range. The noise in the simulation is set to complex gaussian white noise.

Experimental data gets from SWell96Ex Event S5, we use the receive sound pressure of vertical line array(VLA) to train our neural network. The array recorded a total of 75 min of

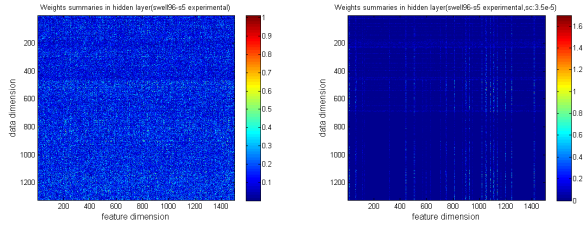

Figure 1: Weights summaries in hidden layer(left:no constraint,right:with constraint). Sparse constraint training makes the weight coefficient show the group structure,either all zero,or basic is not zero.

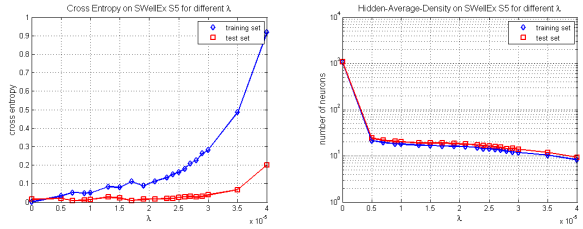

Figure 2: Model error and Average-activation-density for different λ. Regularization on neuron-level significantly reduces the average activation density without much cost in model error.

data. In order to facilitate processing, we took 0-50min data as a training set.

Consistent with the simulation part, we divide the trajectory into 300 grids, 25m each. We set 1 second as a snapshot and get 3000 sample-covariance matrix (SCM), the sample-covariance matrix is averaged at every two snapshots. At the time of training, we took 9/10(2700)of samples as training set and another 1/10(300) as test set.

3.2 The Effect of Sparse Constraint Training

The regularization degree on model affects the model error and average activation density of FNN's hidden layer. As the coefficient grows, the model error on training set and test set also grows, but slower on test set, which means the regularization helps improving the performance on test data in some sense. The model error is defined as the dissimilarity between true probability distribution and estimated probability distribution, thus the Kullback-Leibler(KL) divergence. In Fig.2, we use the cross entropy equivalently.

On the other hand, the regularization on neuron-level significantly reduces the average activation density, The order of magnitude drops from 10^3 to 10, and keep it stably. This phenomenon is a good news for us to train a sparse representation form data.

Compared to the case of training without sparse constraints, sparse constraint makes the weight coefficient in the hidden layer show the group structure, either all zero, or basic is not zero. Take the case $\lambda = 3.5 \times 10^{-5}$ for example, the number of feature vectors reduced from 1500 to 140, as shown in Fig.1. In addition, it can be seen that the relative size of learned weights is related to the frequency, and even at the same frequency, the the weight corresponding to real and imaginary parts is also different, we can see bright and dark strip distribute along the data dimension in Fig.1. The data dimension is arranged according to the frequency relationship when we plot the weights summaries in Fig.1.

Apart from being beneficial to feature selection, sparse constraint can also reduce the activation rate of neurons in the hidden layer, without significant reduction in accuracy. In our example, when the coefficient is 3.5e-5, the average activation neuron number is just 11, which is greatly reduced,

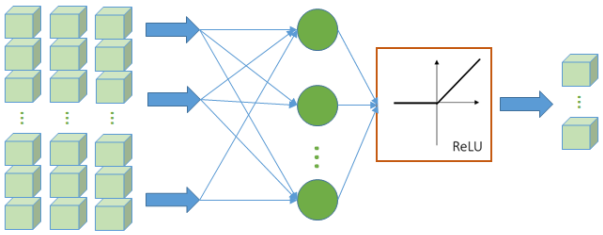

Figure 3: The learned sparse representation model.The learned feature space spans data(scm) space likelihood that few basis functions ϕ explain a given data.

the activation rate is only 0.7%. These mean, in this example, the input 1323-elements SCM data can be represented by only 140 feature vectors. Evean more, averagely, each data sample can be represented by only 11 feature, as Fig.1 showns. The metadata is being compressed well. To sum up, by using regularization strategy on neural networks, we get a sparse and low rank model, where sparse means the transferred representation is sparse, low rank means the rank of learned weight matrix is low. The learned sparse representation model is Illustrated in Fig.3. A sparse vector can be formed by filtering the data measured in sensor nodes through the pre-trained neuron networks. Decisions, such as target location can be made by further processing in fusion data.

3.3 Comparison with conventional matched-field processing method

As a comparison, here we use Bartlett Processor to position the source position. There are two kinds of replica-field used in Bartlett Processor, one is calculated from model by kraken(noted as bartlett 2), another is from measurement data(noted as bartlett 1), same as the training data used in FNN.

The accuracy and absolute mean error of different methods under different frequency are summed in table 1 and table 2. As we can see, whether it is in a narrowband or a wideband, the accuracy of FNN is always better than Bartlett, and

Table 1: Localization accuracy of FNN and MFP on SWell96Ex-S5 data

Methods	FNN	MCE	Bartlett 1	Bartlett 2
109Hz	89.3%	72.3%	37.7%	3.7%
232Hz	97%	91%	17.7%	4.3%
385Hz	99.7%	97.7%	14%	0.67%
109,232,385Hz	99%	99.7%	40.7%	7.7%

Table 2: Absolute mean error of FNN and MFP on SWell96Ex-S5 data(m)

Methods	FNN	MCE	Bartlett 1	Bartlett 2
109Hz	28.1	290.3	852.8	1219.5
232Hz	7.4	2.5	832.3	832.3
385Hz	0.08	0.58	1266.7	1756.3
109,232,385Hz	0.25	0.083	477.2	722.9

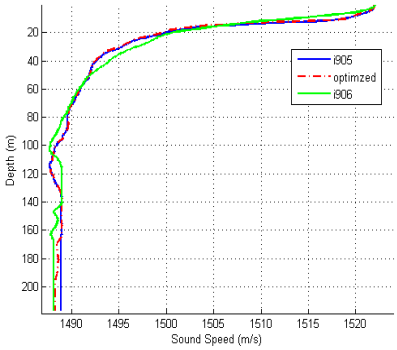

Figure 4: Plots of sound speed profiles.

not worse than direct data match(noted as MCE), which is more obvious when it come s to the comparison of absolute mean error. Thus, the study of neuron networks based sparse representation model is meaningful in positioning problem.

3.4 The influences of ssp mismatch on FNN classifier

In the MFP method, the model accuracy is heavily affected by the mismatch problem[17–19]. Fig.5 gives the FNN positioning results by simulations in different degrees change of sound speed profiles. Here, snapshot is 10 and SNR is 5dB. Comparing to optimized-ssp, the i905-ssp has only a very small change, within 0.5m/s at the same depth. The change in i906-ssp is much significant, which can be seen from the shape in Fig.4.

The performance curves for FNN, Bartlett, MCE is plotted by 1000 times Monte Carlo simulation in Fig.5. When the change in ssp is relatively small(the up two sub figures), FNN positioning best, MCE second and Bartlett worst. When the shape of ssp change(the down two figures), the accuracy order

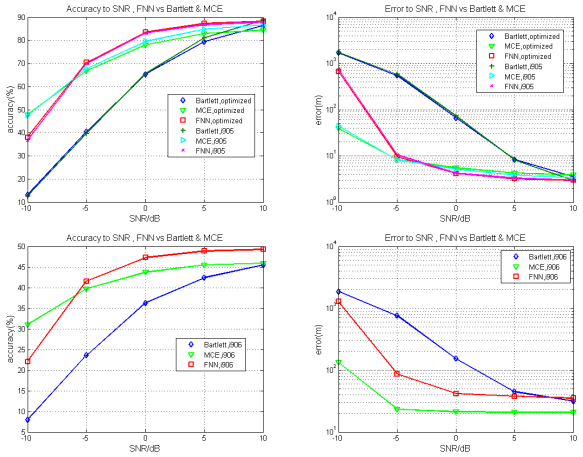

Figure 5: FNN positioning performance curve on simulation data(frequency:109,232,385Hz). FNN is also sensitive to ssp mismatch, but still performs better than MFP.

unchanged, but the absolute error of FNN becomes bigger than MCE.

3.5 Increase model tolerance by mixed data training

The simulation results show that training the model using data collected from different ssp can significantly improve the tolerance of the classifier, which means FNN can learn weights over a set of changing ssp. This is an attractive discovery.

As discussed in section 3.4, the FNN is also sensitive to ssp mismatch, but still performs better than Bartlett. When the environment ssp has a big change in the shape(such as from ssp-optimized to i906), the performance of the estimator drops about 40% in accuracy. In this section, by adding some data collected form i906-ssp, the positioning ability of FNN on i906*(which is little changed from i906,for the sake of testing) is as better as before. Although the accuracy for i905 has a little glissade compared with single data training, the performance for i906 improved. In general, the trained FNN classifier works well on both two different shape ssp. Note that, the legend 'i905,combined' means the model is trained by mixed data collected from ssp i906 and ssp optimized, then the model is tested on ssp i905, rests is similar.

4 CONCLUSIONS

From the view of information theory, the objective of signal processing is to exact specific task-relevant information from data. How to exact the relevant information from measured data is an important issue.Machine learning may be able to provide ideas for this issue, because of its strong ability to learn and characterize. In this paper, we develop a method to learn sparse representation from data based on neural networks. At present, we only discuss the situation of a single

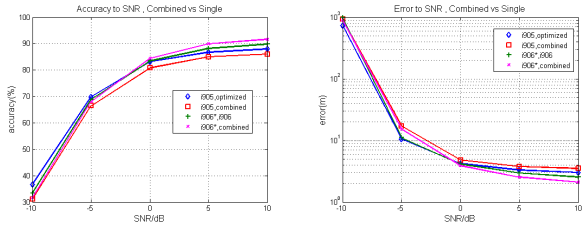

Figure 6: FNN positioning performance curve on simulation data. FNN model tolerance can be by significantly improved by mixed data training

linear array, the situation for multi-array will be discussed in future. Verification on SWell96Ex experimental data shows this method can help select feature vectors and extract task-related information contained in the signal. The pre-trained sparse representation model can help the underwater networks use the least possible amount of resource to store or transfer the data. Compared to the traditional MFP method, the neural network trained model have more ability to resist sound speed profile mismatch and can fine-tune to a feature extractor conveniently.

ACKNOWLEDGMENTS

The work is supported by ...

REFERENCES

[1] Michael J Roan and Elizabeth Hoppe. Distributed underwater source location estimation using a multiarray network. In *OCEANS 2009-EUROPE*, pages 1–4. IEEE, 2009.

[2] Josh G Erling, Michael J Roan, and Mark R Gramann. Performance bounds for multisource parameter estimation using a multiarray network. *IEEE Transactions on Signal Processing*, 55(10):4791–4799, 2007.

[3] Robert R Tenney and John R Delaney. A distributed aeroacoustic tracking algorithm. In *American Control Conference*, number 21, pages 1440–1450, 1984.

[4] Yaakov Bar-Shalom and Xiao-Rong Li. *Multitarget-multisensor tracking: principles and techniques*, volume 19. YBs London, UK:, 1995.

[5] Branko Ristic, Sanjeev Arulampalam, and Christian Musso. The influence of communication bandwidth on target tracking with angle only measurements from two platforms. *Signal Processing*, 81(9):1801–1811, 2001.

[6] Lance M Kaplan, Peter Molnar, and Qiang Le. Bearings-only target localization for an acoustical unattended ground sensor network. In *Proc. SPIE AeroSense*, 2001.

[7] Fe Schweppe. Sensor-array data processing for multiple-signal sources. *IEEE Transactions on Information Theory*, 14(2):294–305, 1968.

[8] Mati Wax and Thomas Kailath. Decentralized processing in sensor arrays. *IEEE transactions on acoustics, speech, and signal processing*, 33(5):1123–1129, 1985.

[9] Alexandra Tolstoy. *Matched field processing for underwater acoustics*. World Scientific, 1993.

[10] Arthur B Baggeroer, WA Kuperman, and Henrik Schmidt. Matched field processing: Source localization in correlated noise as an optimum parameter estimation problem. *The Journal of the Acoustical Society of America*, 83(2):571–587, 1988.

[11] Arthur B Baggeroer, William A Kuperman, and Peter N Mikhalevsky. An overview of matched field methods in ocean acoustics. *IEEE Journal of Oceanic Engineering*, 18(4):401–424, 1993.

[12] Brendan Nichols and Karim G Sabra. Cross-coherent vector sensor processing for spatially distributed glider networks. *The Journal of the Acoustical Society of America*, 138(3):EL329–EL335, 2015.

[13] Dag Tollefsen, Peter Gerstoft, and William S Hodgkiss. Multiple-array passive acoustic source localization in shallow water. *The Journal of the Acoustical Society of America*, 141(3):1501–1513, 2017.

[14] T. C Yang. Issues in underwater acoustic signal processing in the era of big data. 2015.

[15] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep learning*. MIT press, 2016.

[16] Haiqiang Niu, Peter Gerstoft, and Emma Reeves. Source localization in an ocean waveguide using supervised machine learning. *arXiv preprint arXiv:1701.08431*, 2017.

[17] A Tolstoy. Sensitivity of matched field processing to sound-speed profile mismatch for vertical arrays in a deep water pacific environment. *The Journal of the Acoustical Society of America*, 85(6):2394–2404, 1989.

[18] C Feuillade, DR Del Balzo, and Mary M Rowe. Environmental mismatch in shallow-water matched-field processing: Geoacoustic parameter variability. *The Journal of the Acoustical Society of America*, 85(6):2354–2364, 1989.

[19] Donald R Del Balzo, Christopher Feuillade, and Mary M Rowe. Effects of water-depth mismatch on matched-field localization in shallow water. *The Journal of the Acoustical Society of America*, 83(6):2180–2185, 1988.