Matched-field source localization using sparsely-coded machine learning and data-model mixed training

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

Source localization is a basic problem in ocean acoustics. The matched-field processing (MFP) is a popularly used approach to solve this problem and there have been many researches on it. However, MFP is sensitivity to the mismatch problem and performs well only when the ocean environment is accurately known. Machine learning learns inference directly from observation and can be designed to learn a generic model suitable for different scenarios. In this paper, source localization is view as a machine learning problem and a prediction model was learned by training a sparsely-coded feed-forward neural network with mixed environment model data. Sparsely-coded network is applied for preventing the model from over-learning. Results on SWellEx96 experiment show that the learned model can achieve good positioning performances in source range estimation for varying sound-speed profiles. Machine learning model is more tolerant and has potential advantages in underwater source localization, compared with Bartlett matched-field processing.

CCS CONCEPTS

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

KEYWORDS

Machine learning, source localization

ACM Reference format:
Shougui Cai and Wen Xu. 2017. Matched-field source localization using sparsely-coded machine learning and data-model mixed training . 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

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

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

1 INTRODUCTION

Matched-field processing (MFP) is a common technique for source localization in an acoustic wave-guide[1–3]. MFP localization matches measured acoustic pressure field data on an arrary of sensors with a replica field computed by a numerical propagation model for an assumed source range and depth. The processor output is maximum at the true source range and depth. However, MFP requires a pretty good knowledge of environment, which means significant errors in the environment model can be introduced into the depth and range localization predictions.[4, 5]

Machine learning methods do not require a good a prior information and can implement a required calculation through learning from examples. A well designed structure can learn a generic model that works in different kinds of scenarios. This is meaningful for improving the tolerance of conventional methods by introducing machine learning methods into underwater acoustics. Previous works had use artificial neural networks to classify whale sounds[6], locate targets[7] area and discriminate depth[8]. A notable recent example of using machine learning methods in underwater acoustic is the application of nonlinear classification to source localization[9]. It seems there is no discussion on how using machine learning to help solve the mismatch problem in ocean acoustics.

In this paper, the source localization problem is viewed within a machine learning framework. As the sound-speed profiles (ssp) in the water layer is the most important parameter needed to be known accurately[10], we primary focus on the ssp mismatch problem. Two different degrees of error (a large one and a slight one) in the knowledge of the sound-speed profile were chose to train and test the model. The large errors has significant chang in shape, while the slight one just has small shift (within 0.5m/s at the same depth) in sound speed. Effects of such errors on postioning performace for various methods, including Bartlett matched-field processing, matched-covariance estimation(MCE) and feed-forward neural networks based method (FNN), are compared. Treating different sound speed profiles as different application scenes, a generic model was learned by data-model mixed training, the trained model was tested on varying ssp. In Niu's work[9], he used a dense neural networks to train his model, the model performs well on Noise09 experimental data, which verified that FNN can achieve a good prediction performance when source localization is solved as a classification problem. However, as he said, the FNN classifier will be overfitting and predictes poorly when the SNR of training data is low.

In order to overcome this problem, a sparsely-coded neural networks was used in this paper. Besides, our models were trained and tested on SWell96 experimental or simulated data.

2 NEURAL NETWORKS BASED SOURCE LOCALIZATION

In this section, we discuss how to establish a source localization prediction model using neural networks and how to learn parameters in the model.

2.1 Neural Networks Models and Function Approximation

As we known, neural networks models can be viewed as a mathematical function f. Taking feed-forward neural network(FNN) as an example, it defines a mapping $y = f(x; \theta)$ between input x and output y by parameter θ, which needed to be learned by a rule. Feed-forward networks are called networks because they are typically represented by composing together many different functions. There might have two function f^1, f^2 connected in a chain[11], 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 regarded 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 feed-forward network, ϕ defines a hidden layer $h = \phi(x; w^{(1)})$, then the total model is $y = f(x; \theta) = hw^{(2)}$, where parameters w mapping $\phi(x)$ to the desired output. Obviously, a criterion is needed to choose basis functions ϕ from a broad class of functions.

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

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

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

Regularization can help solve over-fitting problem during the model training, but sometimes may cause under-fitting. There are two main kinds 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 Training the neural networks with sparse constraint

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.

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

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 the least possible amount of resource is used to store or transfer the data.

What needed to be done 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.

$$h = g(W^{(1)}x + b^{(1)}) \tag{4}$$

Neurons between hidden layer and output layer are connected by a linear combination:

$$z = W^{(2)T}h + b^{(2)} \tag{5}$$

The output of the model is normalized by $softmax$ function, which is a common choice for multi-class classification task[12]:

$$p(y_k|x) = softmax(z)_k = \frac{\exp(z_k)}{\sum_j \exp(z_j)} \tag{6}$$

where g is the nonlinear function, h is the the desired the sparse data, x is the measures, $p(y_k|x)$ is the probability that measured signal x transmitted from position k, W and b is the parameters needed to learned.

3 SIMULATION AND EXPERIMENTAL RESULTS

In this section, same as Niu's work[9], we implemented a simple FNN with just one hidden layer to learn source range directly from SWellEx96 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, the influence of sound speed profile(ssp) mismatch on the performance of FNN classifier is investigated by simulations. The tolerance of the classifier was improved by training the model using data sampled under different ssp.

Simulation environment is the widely studied SWell96Ex test, conducted in a 216m deep shallow waveguide environment. 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, the shallow sound source was used, which was towed at a depth of about 9m and transmitted 9 frequencies between 109Hz and 385Hz.

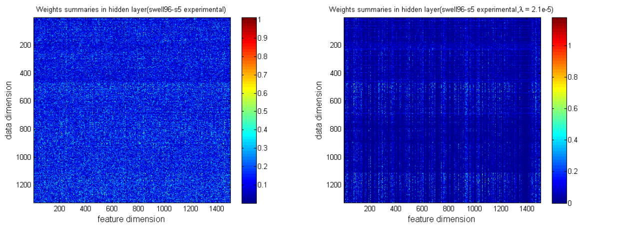

Figure 1: Weights summaries in hidden layer (left: no constraint, right: with constraint). Sparse constraint training makes the weight coefficient show the group structure, either all zero, or basic is not zero.

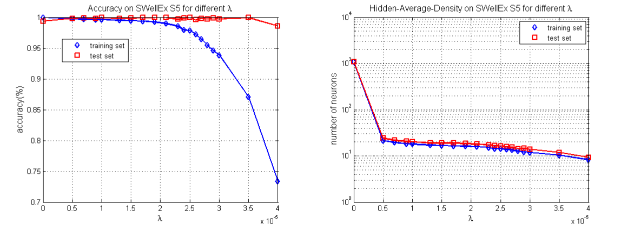

Figure 2: Model accuracy and Average-activation-density for different λ. Regularization on neuron-level significantly reduces the average activation density without much loss in model accuracy.

3.1 Parameter settings

In simulation part, acoustic data used to train and test the neural network was simulated using kraken with environment model. We also use the normalized sample-covariance matrices of measured pressure at each frequency as model input data. In input layer, number of neurons D is $L^2 \times N_{fre}$(number of frequency used). The number of neurons in the output layer (number of classes) $K = 300$. Simply, the number of neurons in hidden layer was set to be equal to the input layer, i.e. $M = D$. Fast Fourier transform duration is 1-seconds, snapshot N_s for constructing SCMs is 10. The number of vertical array elements L is 21. Specifically, the cost function now becomes

$$\tilde{J}(\theta) = -\frac{1}{N}\sum_{n=1}^{N}\sum_{k=1}^{K} t_{nk} \ln y_{nk} + \lambda \|h\|_1 \tag{7}$$

where t_{nk} and y_{nk} are real and predictive probability of sample data x belongs to class k separately. We chose $C = 1$ here. For the sake of learning speed and sparsity of hidden neurons, $ReLU$ activation was applied in hidden layer. The training set is 3000 samples sampled uniformly between range 1.82-8.65km, test set is another 300 data sampled from the same range. The noise in the simulations is all set to be complex gaussian white noise.

Experimental data was got from SWell96Ex Event S5 vertical line array (VLA) . The array recorded a total of 75 min of data. In order to facilitate processing, 0-50min data was took as a training set.

Consistent with the simulation part, the trajectory was divided into 300 grids, 25m each. The snapshot was set as 1-second 3000 sample-covariance matrix (SCM) would be got, the sample-covariance matrix was averaged at every two snapshots. At the time of training, 9/10(2700)of samples were took 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 accuracy and average activation density of FNN's hidden layer. As the coefficient grows, the model accuracy on training set and test set also drops, but are slower on test set. When the coefficient

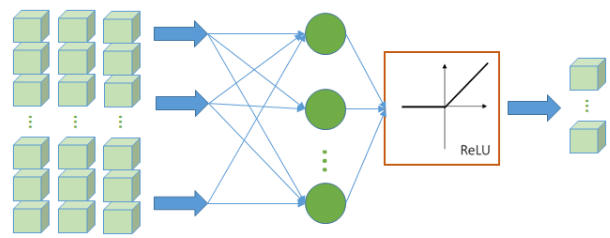

Figure 3: The learned sparse representation model. The learned feature space spans data(scm) space likelihood that few basis functions ϕ explain a given data.

is too big, the model accuracy on training set becomes much lower than test set, which indicates under-fitting.

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 stable. This phenomenon is a good news for us to train a sparsely-coded neural networks.

Compared to the case of training without sparse constraints, sparsely-coded neural network makes the weight coefficient in the hidden layer show the group structure, the element of weight vector is either all zero, or basic is not zero. Take the case $\lambda = 2.1 \times 10^{-5}$ for example, the number of feature vectors reduced from 1500 to 740, 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, bright and dark strips can be seen distributed along the data dimension in Fig.1. The data dimension is arranged according to the frequency relationship when plotting 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 2.1e-5, the average activation neuron number is just 16, which is greatly reduced, the activation rate is only 1.1%. These mean, in this example, the input 1323-elements SCM data space can be represented

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

by 740 feature vectors. Even more, averagely, each data sample can be represented by only 16 feature, as Fig.1 shows. The metadata is being compressed well. To sum up, by using regularization strategy on neural networks, a sparse and low rank model can be builded, where sparse means the transferred representation is sparse, low rank means the rank of learned weight matrix is low. The learned sparse coding model is Illustrated in Fig.3. A sparse vector can be formed by filtering the data measured in sensor nodes through the pre-trained sparsely-coded neuron networks. Decisions, such as target location can be made by further processing.

3.3 Comparison with conventional matched-field processing method

As a comparison, Bartlett Processor was used here to position the source position, as Niu did in his work. There are two main kinds of replica-field used in Bartlett Processor, one is simulated by kraken(noted as bartlett 2), another is 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 single frequency or a mutli-frequency, the accuracy of FNN is always better than Bartlett, and 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 coding 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[4, 5, 10]. 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.

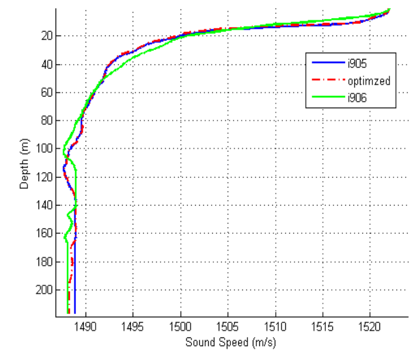

Figure 4: Plots of sound speed profiles.

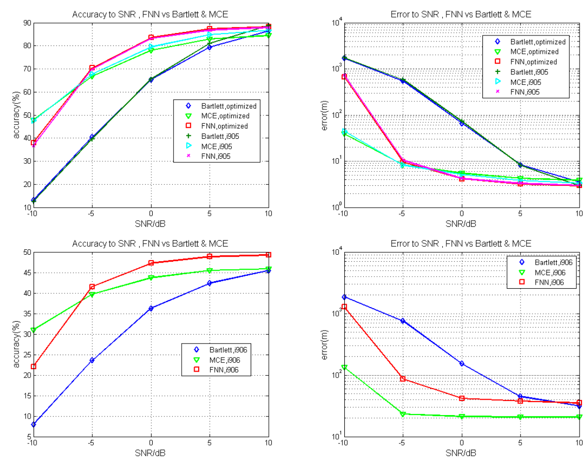

Figure 5: FNN positioning performance curve on simulation data(frequency:109,232,385Hz). FNN is also sensitive to ssp mismatch, but still performs better than MFP.

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 are plotted by 1000 times Monte Carlo simulation. 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 unchanged, but the absolute error of FNN becomes bigger than MCE. FNN is also but less sensitive to ssp mismatch than Bartlett.

3.5 Increase model tolerance by data-model mixed training

The simulation results show that training the model using data collected from different ssp can significantly improve

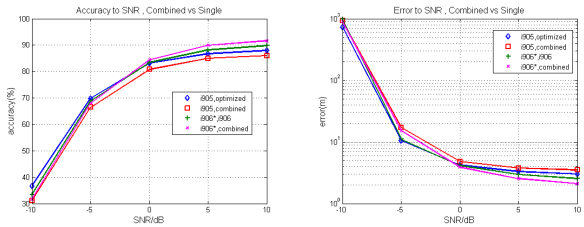

Figure 6: FNN positioning performance curve on simulation data. FNN model tolerance can be by significantly improved by data-model mixed training

the tolerance of the classifier, which means FNN can learn weights over a set of changing ssp.

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 from 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 case, 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 are similar.

4 CONCLUSION

It is attractive to see that the sparsely-code neural networks model predicts accurately on source localization, as well as the dense neural networks do, but needs fewer basis functions to span the data space. Combined with data-model mixed training, the model tolerance can be obviously increased, and performs better in sound speed profile mismatch case than bartlett matched-field processing method. This paper primarily focus on the fine-tune for feed-forward neural networks. It deserves more efforts to apply complicated machine learning methods on ocean acoustic source localization.

ACKNOWLEDGMENTS

The work is supported by ...

REFERENCES

[1] Homer P Bucker. Use of calculated sound fields and matched-field detection to locate sound sources in shallow water. *The Journal of the Acoustical Society of America*, 59(2):368–373, 1976.

[2] 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.

[3] 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.

[4] 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.

[5] 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.

[6] Aaron M Thode, Katherine H Kim, Susanna B Blackwell, Charles R Greene Jr, Christopher S Nations, Trent L McDonald, and A Michael Macrander. Automated detection and localization of bowhead whale sounds in the presence of seismic airgun surveys. *The Journal of the Acoustical Society of America*, 131(5):3726–3747, 2012.

[7] Ben Zion Steinberg, Mark J Beran, Steven H Chin, and James H Howard Jr. A neural network approach to source localization. *The Journal of the Acoustical Society of America*, 90(4):2081–2090, 1991.

[8] John M Ozard, Pierre Zakarauskas, and Peter Ko. An artificial neural network for range and depth discrimination in matched field processing. *The Journal of the Acoustical Society of America*, 90(5):2658–2663, 1991.

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

[10] 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.

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

[12] Christopher M Bishop. *Pattern recognition and machine learning*. springer, 2006.