Underwater source localization using multi-frequency machine learning

Emma Ozanich
Haiqiang Niu and Peter Gerstoft
Scripps Institution of Oceanography, UCSD
Example problem: Ship Range for Noise 09 Experiment

- **Training data**
  - Jan. 31, 2009 01:43-2:05
  - 2 m/s

- **Test-Data-1**
  - Jan. 31, 2009 01:01-01:24
  - -2 m/s

- **Test-Data-2**
  - 4 m/s
Example problem: Ship Range for Noise 09 Experiment

- **Training data**
  - Jan. 31, 2009 01:43-2:05
  - 2 m/s

- **Test-Data-1**
  - Jan. 31, 2009 01:01-01:24
  - -2 m/s

- **Test-Data-2**
  - 4 m/s
Example problem: Ship Range for Noise 09 Experiment

Comparison method: Matched-Field Processing (model-based)

1. **Model the sound propagation.**
   Parameters: water depth, sound speed profile of water and sea floor, density, layers...

2. **Compare the modeled sound pressure** \( (p_c) \) **and observations** \( (p_s) \).

   \[
   \text{maximizing } p_c(r)^H D p_c(r)
   \]

   \[
   D(f) = \frac{1}{N_s} \sum_{s=1}^{N_s} \tilde{p}_s(f) \tilde{p}_s^H(f).
   \]

   16 sensors spanning from 128 – 143 m depth (15 m)

   \( R = 0.1 – 2.86 \text{ km} \)

   \( Z_s = 5 \text{ m} \)

   \( Z_r = 128 – 143 \text{ m} \)

   \( D = 152 \text{ m} \)

   \( \Delta z = 1 \text{ m} \)

   Layer
   \[
   \begin{align*}
   C_p &= 1572 – 1593 \text{ m/s} \\
   \rho &= 1.76 \text{ g/cm}^3 \\
   \alpha_p &= 2.0 \text{ dB/\lambda}
   \end{align*}
   \]

   Halfspace
   \[
   \begin{align*}
   C_p &= 5200 \text{ m/s} \\
   \rho &= 1.8 \text{ g/cm}^3 \\
   \alpha_p &= 2.0 \text{ dB/\lambda}
   \end{align*}
   \]
Example problem: Matched-Field Processing (MFP)

- "Ambiguity surfaces" show match at each modeled range
- MFP has challenges due to sidelobes

\[
E_{\text{MAPE}} = \frac{100}{N} \sum_{i=1}^{N} \left| \frac{R_{pi} - R_{gi}}{R_{gi}} \right|
\]

\(R_{pi} = \text{predicted range}, \ R_{gi} = \text{ground truth range.}\)

Model Replicas (\(p_c\))

Data Replicas (\(p_c\))

300-950 Hz, \(\Delta f = 10\) Hz
(66 frequencies)
Feed-forward neural network, also called Multilayer Perceptron

One hidden layer:
- For inputs $x_n$, $z_{nj} = \sigma ((w^{(1)}_j)^T \cdot x_n)$
- $\sigma (x) = \text{sigmoid}(x) = (\exp(-x)+1)^{-1}$
- Softmax output:

$$ f(z_{nj}) = \frac{e^{z_{nj}}}{\sum_{k=1}^{M} e^{z_{nk}}}, j = 1, \ldots, M $$

Output is a probability, where maximum bin is model prediction
Example problem: Feed-Forward Neural Network (FNN)

- Inputs for FNN (feature engineering)

  Sound pressure

  \[ p(f) = S(f)g(f, r) + n, \]

  Normalize pressure to reduce the effect of \(|S(f)|\)

  \[ \tilde{p}(f) = \frac{p(f)}{\sqrt{\sum_{l=1}^{L} |p_l(f)|^2}} = \frac{p(f)}{||p(f)||_2} \]

  Sample Covariance Matrix to reduce effect of source phase

  \[ C(f) = \frac{1}{N_s} \sum_{s=1}^{N_s} \tilde{p}_s(f)\tilde{p}_s^H(f) \]

- Vectorize \( \text{re}\{C\} \) and \( \text{imag}\{C\} \). Concatenate for multiple frequencies.

  \( S(f) \): Source term
  \( L \): Number of sensors
  \( N_s \): Number of snapshots
Example problem: Feed-Forward Neural Network (FNN)

- Classification:
  - Map range to bins
  - Use Kullbach-Liebler (KL, aka relative entropy) to compare softmax output with ‘one-hot’ vector

Example vector:

\[
\begin{align*}
  r_1 &: 1 \ 0 \ 0 \ \ldots \ 0 \ 0 \ 0 \ 0 \\
  r_2 &: 0 \ 1 \ 0 \ \ldots \ 0 \ 0 \ 0 \ 0 \\
  r_3 &: 0 \ 0 \ 1 \ \ldots \ 0 \ 0 \ 0 \ 0 \\
  \vdots \\
  r_K &: 0 \ 0 \ 0 \ \ldots \ 0 \ 0 \ 0 \ 1 \\
\end{align*}
\]

- Input: \(X_n\)
- Label: \(t_n\)
- \(t_{nk} = \begin{cases} 
  1 & \text{if } ||r - r_k|| \leq \Delta r / 2, \\
  0 & \text{otherwise.}
\end{cases}\)
Example problem: Feed-Forward Neural Network (FNN)

- (a)-(c) Test-Data-1
- (d)-(f) Test-Data-2

\[ E_{MAPE} = \frac{100}{N} \sum_{i=1}^{N} \left| \frac{R_{pi} - R_{gi}}{R_{gi}} \right| \]

Where \( R_{pi} \) = predicted range, \( R_{gi} \) = ground truth range.

550 Hz

\[ E_{MAPE} = 18\% \]

950 Hz

\[ E_{MAPE} = 12\% \]

300-950 Hz, \( \Delta f = 10 \text{ Hz} \)

(66 frequencies)
Example problem: Ship Range for Noise 09 Experiment

- (a)-(c) Test-Data-1
- (d)-(f) Test-Data-2
- Best FNN: $E_{MAPE} = 3\%$ (not shown)
Example problem: Support Vector and Random Forest

Random Forest (RF)

Support Vector Machine (SVM)

- $E_{\text{MAPE}} = 3\%$
- $E_{\text{MAPE}} = 3\%$
- $E_{\text{MAPE}} = 2\%$
- $E_{\text{MAPE}} = 2\%$
Example problem: Ship Range for SBCEx16 Experiment

- Cargo shipping lanes, 3 passing ships

![Map showing cargo shipping lanes and passing ships](image)

<table>
<thead>
<tr>
<th>Track No.</th>
<th>Data set</th>
<th>Time period</th>
<th>Ship name</th>
<th>Speed (m/s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Track1</td>
<td>Training-Data</td>
<td>13:00–13:33 (9/15)</td>
<td>KUMANO MARU</td>
<td>6.7</td>
</tr>
<tr>
<td>Track2</td>
<td>Test-Data-1</td>
<td>19:11–19:33 (9/16)</td>
<td>APL PHILIPPINES</td>
<td>10</td>
</tr>
<tr>
<td>Track3</td>
<td>Test-Data-2</td>
<td>19:29–19:54 (9/17)</td>
<td>NORDSPRING</td>
<td>8.0</td>
</tr>
</tbody>
</table>
Example problem: Ship Range for SBCEx16 Experiment

- Cargo ship spectral signatures
- Use 53-200 Hz, spacing of 3 Hz
Example problem: Ship Range for SBCEx16 Experiment

- (a)-(c) Test-Data-1. (a),(d) MFP; (b),(d) SVM; (c),(f) FNN.
- (d)-(f) Test-Data-2

\[ E_{MAPE} = 34.6\% \]
\[ E_{MAPE} = 1.5\% \]
\[ E_{MAPE} = 2.2\% \]
\[ E_{MAPE} = 36.1\% \]
\[ E_{MAPE} = 2.2\% \]
\[ E_{MAPE} = 3.9\% \]
Feed-Forward Neural Network: Regression

- **FNN with regression:**
  - Single output is estimate of range
  - Classification maximally separates classes while regression minimizes target error
  - Mean squared error
Noise 09 Results: FNN with Regression

- (a)-(c) Test-Data-1
- (d)-(f) Test-Data-2

One hidden layer

Two hidden layers

Three hidden layers

<table>
<thead>
<tr>
<th>(a)</th>
<th>(b)</th>
<th>(c)</th>
<th>(d)</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image1.png" alt="Graph" /></td>
<td><img src="image2.png" alt="Graph" /></td>
<td><img src="image3.png" alt="Graph" /></td>
<td><img src="image4.png" alt="Graph" /></td>
</tr>
</tbody>
</table>

$E_{MAPE} = 10\%$

$E_{MAPE} = 5\%$
Example problem: Multiple ships at SCE17

16 sensors spanning from 13m to 69.25m (56.25m)

\[ R = 10\,\text{m} - 20\,\text{km} \]
\[ z_s = 10\,\text{m} \]
\[ z_r = 13 - 69.25\,\text{m} \]
\[ \Delta z = 3.75\,\text{m} \]

<table>
<thead>
<tr>
<th>Material</th>
<th>( c_p )</th>
<th>( h )</th>
<th>( \rho )</th>
<th>( \alpha )</th>
</tr>
</thead>
<tbody>
<tr>
<td>MUD</td>
<td>1435 - 1485 m/s</td>
<td>10 m</td>
<td>1.6 g/cm³</td>
<td>0.005-0.01 dB/m @ 1kHz</td>
</tr>
<tr>
<td>SAND</td>
<td>1796 m/s, h=10m</td>
<td>10 m</td>
<td>2.2 g/cm³</td>
<td>0.15 dB/m @ 1kHz</td>
</tr>
<tr>
<td></td>
<td>1610-1640 m/s, h = 10 m</td>
<td>10 m</td>
<td>1.8 g/cm³</td>
<td>0.3 dB/m @ 1kHz</td>
</tr>
<tr>
<td></td>
<td>Halfspace (h -&gt; \infty)</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

credit: John Goff
Example problem: Multiple ships at SCE17

- 9 training tracks, 1 test track (bold, orange)
- Test track from different region
Example problem: Multiple ships at SCE17

- Tonals at ~300 Hz come from experiment ship R/V Endeavor.
Example problem: Multiple ships at SCE17

- 50-200 Hz used for experiment.
- Viking Bravery used as test track (not south of array).
Example problem: Multiple ships at SCE17

- Training MAPE = 2.5%
- Validation MAPE = 20.2%
PCA for high-dimensional input data

- Problem: not enough data to train variation in high dimensions
- $M$ is a real matrix
- $M'$ is a real matrix projected into a lower dimension

$2 \times 136 \times N_f$
PCA for high-dimensional input data

- Choose the top $k$ singular values & project into the reduced data space
- $k$ is chosen based on model performance
- $U$ is a unitary matrix of size $N_{\text{samples}} \times N_{\text{samples}}$
- $\Sigma$ is a diagonal matrix of size $N_{\text{samples}} \times N_{\text{features}}$
- $V$ is size $N_{\text{features}} \times N_{\text{features}}$

$$M = U \Sigma V^T$$

$$M' = MV[:,1:k]$$
Example problem: Multiple ships at SCE17

- 100 top components kept
- Explains ~20% of variance within data (sum of top $k$ normalized singular values)
Conclusions

• Machine learning (ML) models were used to predict cargo ship ranges
  • Trained on similar previous paths
  • 3-layer FNN, SVM, or RF

• ML models achieve lower error than MFP for real data from 1. controlled ship paths, 2. cargo ships in lane
  • Typically one training track, similar test tracks
  • Tracks are close in location, range, and time

• Larger variation between ship tracks results in worse ML performance
  • PCA may help improve results when data is limited relative to variation within features
Appendix: SVM

- SVM optimally divides the feature space by class (predicted label $t_n$, true label $y(x_n)$):

$$\min_{\tilde{w},b} \left[ C \sum_{n=1}^{N} \xi_n + \frac{||\tilde{w}||^2}{2} \right]$$

$$\xi_n = \begin{cases} 
0 & t_n y(\tilde{x}_n) \geq 1, \\
|t_n - y(\tilde{x}_n)| & t_n y(\tilde{x}_n) < 1
\end{cases}$$

$C > 0$.

- Multi-class problem solved by iterating 2-class problem

![Two-Class example of SVM in 2D space](image)
Appendix: RF

- Decision tree

For input $x_n = [x_{n1}, x_{n2}]$:

$$
\begin{align*}
\bar{x}_n \in \bar{x}_{\text{left}} & \quad x_{ni} \leq c \\
\bar{x}_n \in \bar{x}_{\text{right}} & \quad x_{ni} > c
\end{align*}
$$

Minimize the number of wrongly classified points per region by changing $c$.

- Random forest:
  - Generate hundreds of random trees
  - Keep most frequently occurring regions

Example of 2D decision tree for two classes

$c_1 = 1.9$ and $c_2 = 4.6$