Using Probabilistic Machine Learning to Estimate Ocean Mixed Layer Depth

Recovery from sparse in-situ observations informed from satellite data.

Dallas Foster
Oregon State University
Advisors: David John Gagne, Daniel Whitt

July 29th 2020
Ocean and Earth system processes are highly sensitive to ocean surface mixed layer depth (MLD)
- water mass formation and circulation
- air-sea exchange
- Biogeochemistry

Observational data is increasingly available, but still relatively sparse

Existing methods perform optimal interpolation, but do not inform with satellite sea surface data.

Want to quantify the sub-seasonal relationship between Sea Surface Salinity (SSS), Temperature (SST), Sea Level Height Anomaly (SSH) and MLD

https://www.gfdl.noaa.gov/oceanprocschem/
GOALS AND QUESTIONS

- Compare different ML approaches
- Quantify the uncertainty of MLD estimates
- Produce maps of the MLD as a function of SSS, SST, SSH
- On what spatio-temporal scales can we estimate the MLD reliably?
- What features are not being resolved in the analysis?
- How valuable are the various input data for estimating the MLD?
Argo float MLD data is:
- Sparse
- Non-normal
Preprocessing Steps:

1. **Divide Data**
   - Validate divisions

2. **Calculate Climatology**
   - Apply rolling average
   - Bin data into months
   - Average over bins

3. **Calculate Anomalies**
   - Bin data into months
   - Subtract binned climatology
   - Remove diurnal cycle

4. **Resample**
5. **Interpolate**
<table>
<thead>
<tr>
<th>Terminology</th>
<th>Interpretation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Machine Learning</td>
<td>A framework of building and <strong>fitting nonlinear models to data.</strong></td>
</tr>
<tr>
<td>Uncertainty Quantification</td>
<td>Techniques to determine how likely certain outcomes are if some aspects of the system are not exactly known.</td>
</tr>
<tr>
<td>$S, T, H, x, y$</td>
<td>Variables <strong>SST, SSH, SHA</strong> and the 2-dimensional spatial coordinates.</td>
</tr>
<tr>
<td>$d_{obs}, d, \sigma, \Sigma$</td>
<td>MLD <strong>observations</strong> (sparse grid), <strong>estimates</strong>, <strong>uncertainties</strong> (full grid).</td>
</tr>
<tr>
<td>$\theta, p(\theta)$</td>
<td>Model <strong>parameters</strong> and probability distribution (<strong>prior distribution</strong>).</td>
</tr>
<tr>
<td>$p(d_{obs}</td>
<td>d, \theta)$</td>
</tr>
</tbody>
</table>
Aleatoric Uncertainty

- Inherent noise in data
- Irreducible

Model must account for **aleatoric** and **epistemic** uncertainty

- **Monte Carlo** sampling of model
- **Bayesian** interpretation of model weights
- Specification of **noise model**

Epistemic Uncertainty

- Lack of knowledge, data
- Deficiency of model
A Gaussian Process (GP) $y$, observed at points $x$ is a sample from a multivariate normal distribution,

$$y(x) \sim N(0, K(x, x'))$$

$K$ is a covariance function that specifies the spatial relationships between points.

Allows us to predict the mean and variance of $y$ at new points $x_*$. 
Modeling Steps:
1. Generate dense MLD field
   \[ d = f(S, T, H) + \epsilon \]
2. GPR to sparse field
   \[ d_{\text{obs}} = L \cdot d + \sigma, \quad \sigma \sim N(0, \Sigma) \]
3. Compute loss and maximize weights according to
   \[ p(d|d_{\text{obs}}, \theta) \]
4. Repeat
TRADITIONAL MODELS

LINEAR MODEL

\[ d(x, y) = \begin{bmatrix} a(x, y) \\ \beta(x, y) \\ x(y, y) \end{bmatrix} \cdot \begin{bmatrix} S(x, y) \\ T(x, y) \\ H(x, y) \end{bmatrix} + c(x, y) + \varepsilon(x, y) \]

- Little to no spatial correlation between grid points
- Relatively few parameters required
- Trade off between performance and overfitting

FEED FORWARD ARTIFICIAL NEURAL NETWORK

- Universal function approximator
- Comprised of a series of simple nonlinear functions
  \[ h_l = f(\Delta h_{l-1} + b) \]
- Weights are updated through backpropagation
- Surplus of parameters
LINEAR MODEL

\[
d(x, y) = \left[ \begin{array}{c}
\alpha(x, y) \\
\beta(x, y) \\
\gamma(x, y)
\end{array} \right] \cdot \left[ \begin{array}{c}
S(x, y) \\
T(x, y) \\
H(x, y)
\end{array} \right] + c(x, y) + \epsilon(x, y)
\]

- Little to no spatial correlation between grid points
- Relatively few parameters required
- Trade off between performance and overfitting
FEED FORWARD ARTIFICIAL NEURAL NETWORK

- Universal function approximator

- Comprised of a series of simple nonlinear functions
  \[ h_i = f(Ah_{i-1} + b) \]

- Surplus of parameters
Parameterization Methods

- Have to make a decision about the output’s distribution
- Simple to implement, when possible
- Examples:
  - Least Squares Regression
  - Variational Neural Networks

Sampling Methods

- Initial distribution must be supplied
- Model must be run many times
- Examples:
  - Dropout
  - Variational AutoEncoders
  - Bayesian Neural Networks
PROBABILISTIC METHODS

**VARIATIONAL NEURAL NETWORK**
- Requires even more parameters
- Requires parameterization of noise model
- Better captures aleatoric uncertainty.

**BAYESIAN NEURAL NETWORK (FLIPOUT)**
- Parameterizes a prior noise model for each weight
- Requires double parameters and Monte Carlo sampling
- Can help capture epistemic uncertainty

**DROPOUT**
- Randomly set some weights to zero
- Creates an ensemble of models
- Computationally inexpensive
- Requires sampling to generate statistics

**VARIATIONAL AUTO ENCODERS**
- Learns a dimension reduction of input data
- Gaussian noise parameterization
Comparison of MLD Models by Likelihood of Data

Comparison of MLD Models by KS Statistic
ISSUES AND FUTURE WORK

Model Development

• Machine Learning models need further training

• Evaluate different parameterization of VNN, Flipout, VAE models

• Train and evaluate global models

Analysis

• Further estimate spatial resolution and accuracy of models

• Investigate temporal relationships, predictability

• Build framework for optimal assimilation of model and data
• Useful information can be extracted from surface data to estimate ocean mixed layer depths anomalies (MLD).

• Machine learning models are a promising approach to constructing models for estimating MLD.

• Simple noise parameterizations might be all that is necessary to get decent probabilistic estimates.
  – More analysis is needed!
BAYESIAN NEURAL NETWORK (FLIPOUT)

- Parameterizes a prior noise model for each weight
- Requires double parameters and Monte Carlo sampling
- Can help capture epistemic uncertainty

\[ S(x, y) \]
\[ T(x, y) \]
\[ H(x, y) \]
\[ w_{i,j} \sim N(\mu_{i,j}, \sigma_{i,j}) \]
VARIATIONAL NEURAL NETWORK

- Requires even more parameters
- Requires parameterization of noise model
- Better captures aleatoric uncertainty.

\[
d(x, y) \sim N(\mu(x, y), \sigma(x, y))
\]
• **Randomly** set some weights to zero

• Creates an **ensemble of models**

• Computationally **inexpensive**

• **Requires sampling** to generate statistics
VARIATIONAL AUTO ENCODERS

- Learns a **dimension reduction** of input data
- Gaussian **noise parameterization**

**Encoder Neural Network**

**Decoder Neural Network**

Latent Variables $\sim N(\mu, \sigma)$

**Introduction**

**Data**

**Modeling**

**Machine Learning**

**Uncertainty Quantification**

**Conclusion**