Driving adaptive modeling with data science

Data Science Guest Lecture
Cardiff Metropolitan University
2022-11-21

Robert Caulk, PhD
Founder, Lead Software Developer

Elin Törnquist, PhD
Lead Research Scientist
What is Emergent Methods?

- Open-source computational research company
- Specializing in generalizing adaptive modeling for time-series data
- Developers of creative ML softwares, FreqAI and JaiRevAI
What is FreqAI?
Real-time adaptive modeling toolkit for making actionable market forecasts

User-friendly machine learning sandbox
• 100% open-sourced code base
• Interactive knowledge base with over 10k posts (Discord)
• More than 15 unique developer contributors
• Hundreds of active users finding and reporting bugs

Generalized framework
• Foundational - connects a wide variety of open-source machine learning libraries
• Adaptive - Reinforcement Learning, Decision Trees, Neural Networks, SVM, DBSCAN
• Scientifically sound - industry standard outlier detection methods and statistically safe data handling

Journal of Open Source Software (under review)
Adaptive modeling core engine
Designed for real-time modeling of time-series systems

Use-case specific plug-in
- Example input data:
  1. Building sensor data
  2. Images/fiber optic signals
  3. Network activity
- Example predictions:
  1. Control building HVAC
  2. Predict animal behavior
  3. Identify suspicious activity

Feature set engineering
- Feature development:
  - Signal analysis
  - Multiple time-scales
  - Add feature recency
  - Create Labels
- Cleaned features:
  - Auto NaN truncation
  - Outlier detection and removal
  - Data scaling
  - Dimensionality reduction with PCA

Adaptive learning
- Select from:
  - Decision Trees
  - Reinforcement Learning
  - Neural Networks

Model training
- Build/update historic data:
  - Build if first training
  - Update each retraining

Model inference
- Predictions

Post-processing
- Denormalization
- Gather statistical quantities
- Determine prediction confidence

Gather Raw data
Handle Predictions

XGBoost
TensorFlow
LightGBM
PyTorch
CatBoost
Adaptive modeling core engine

Feature engineering

Data source
- Price data
- On-chain metrics
- Sentiment metrics

Signal analysis:
- Moving averages
- Bollinger band
- Wavelet transform

Feature creation

Raw training features

Time stamp

Feature columns
Characterizing the parameter space
Dissimilarity Index

\[ d_{ab} = \sqrt{\sum_{j=1}^{p} (X_{a,j} - X_{b,j})^2} \]

\[ \bar{d} = \sum_{a=1}^{n} \left( \sum_{b=1}^{n} \frac{d_{ab}}{n} \right) / n \]

\[ DI_k = d_k / \bar{d} \]

- Training data \( X \)
  - Distances \( d \)
  - Minimum distance \( d_k \)

- New data point \( X_k \)


\[ d_k < \bar{d} \rightarrow DI_k < 1 \]

Inlier

\[ d_k > \bar{d} \rightarrow DI_k > 1 \]

Outlier
Characterizing the parameter space
Outlier detection

- Support Vector Machine - plane fitting
  - Linear SVM - fast, but likely too coarse

- DBSCAN - clustering
  - Challenging to define hyperparameters
Adding creative metrics

Inlier metric

1. Compute distances between each point and the lookback points

2. Fit weibull distributions to each lookback point

3. Compute point quantiles from weibull distributions

4. Include inlier-metric in training data
Adaptive modeling core engine

Data handling
Adaptive modeling core engine
Common pitfalls

- Improper NaN handling
  - Don’t blindly fill/replace NaNs
- Improper normalization
  - Normalize test/prediction features to training parameter space
- Tossing the kitchen sink in the dishwasher
  - Preference features from other data sources over redundant signal analysis on the same data source
  - Don’t underestimate the value of computational performance
- Naive stacking of outlier detection/dimensionality reduction methods
  - PCA is great but it may magnify/buffer other methods
- Training a single model to do it all
  - Passing 10 rows x 100 columns vs 100 columns x 10 rows
  - Don’t assume globality
Ongoing experiment (3 weeks)

Configuration

- Compare performance on three popular gradient boosted decision tree algorithms
- Aggregate/select best prediction confidence \textit{(Consumer)}
- Track resource usage of each algorithm
- Server details: four identical 12 core Xeon X5660 2.8 GHz 64gb DDR3

\textit{Consumer}

- **XGBoost**
- **LightGBM**
- **CatBoost**

\begin{figure}
\centering
\includegraphics[width=\textwidth]{decision_tree.png}
\caption{Decision Tree}
\end{figure}

\textit{image source: https://www.saedsayad.com/decision_tree.htm}
Ongoing experiment (3 weeks)
Distributed deployment

- Multiple instance message communication with websockets (*Consumer*)
- GitLab continuous integration + Docker swarm experimental prototyping and deployment
Ongoing experiment (3 weeks)

Preliminary results

<table>
<thead>
<tr>
<th>Model</th>
<th>Training time [s]</th>
<th>Inference time [s]</th>
<th>Profit [%]</th>
<th>RAM usage [Gb]</th>
<th>CPU load [%]</th>
</tr>
</thead>
<tbody>
<tr>
<td>Consumer</td>
<td>N/A</td>
<td>N/A</td>
<td>2.4</td>
<td>2.5</td>
<td>5</td>
</tr>
<tr>
<td>XGBoost</td>
<td>91.7</td>
<td>0.17</td>
<td>2.2</td>
<td>5.4</td>
<td>35</td>
</tr>
<tr>
<td>LightGBM</td>
<td>169.8</td>
<td>0.05</td>
<td>1.8</td>
<td>5.8</td>
<td>37</td>
</tr>
<tr>
<td>CatBoost</td>
<td>256.4</td>
<td>0.52</td>
<td>0.6</td>
<td>6.0</td>
<td>40</td>
</tr>
</tbody>
</table>

![Graph showing data](chart.png)
Real-time data handling
Challenges

- Crash resilience
- Data collection/storage
- Time to prediction
- Prediction handling
- Threading
Who are Emergent Methods?

Robert Caulk, PhD
Founder / Lead Developer
France

Elin Törnquist, PhD
Lead Research Scientist
Sweden

Tim Pogue
Lead Large-Scale Systems Engineer
USA

Wagner Costa Santos
Software Developer / Data Analyst
Brazil

Andy Lawless
Software Developer / Quality Assurance
UK

Steven Caulk, MBA
Head of Business Logistics
USA

contact@emergentmethods.ai