Robust Nonparametric Distribution Forecast with Backtest-based Bootstrap and Adaptive Residual Selection

Presenter: Longshaokan (Marshall) Wang, longsha@amazon.com

Authors: Longshaokan Wang, Lingda Wang, Mina Georgieva, Paulo Machado, Abinaya Ulagappa, Safwan Ahmed, Yan Lu, Arjun Bakshi, Farhad Ghassemi

04-19-22

Amazon

IEEE ICASSP 2022
Farhad Ghassemej
Abinaya Ulagappa
Mina Georgieva
Arjun Bakshi
Longshaokan Wang
Safwan Ahmed
Yan Lu
Paulo Machado
Outline

• Introduction
• Method
• Experiments
• Summary
Motivation

• Planners and optimization systems often require distribution forecast
  • Product manufacturing
  • Inventory Allocation
• Quantifying uncertainty associated with point forecast
• **Goal**: Develop accurate and efficient method for generating distribution forecast at scale
Summary

• Proposed a flexible plug-and-play framework that can extend an arbitrary Point Forecast model to produce Distribution Forecast
• Extended bootstrapping predictive residuals with backtest and covariate sampling
• Proposed an adaptive residual selector
• Proposed a new formula for applying bootstrapped residuals
• Empirical evaluation on real-world data
Summary

• The proposed Distribution Forecast framework has the following advantages:
  • Incorporates different sources of forecast uncertainty by design
  • Integrates well with an arbitrary PF model to produce DF
  • Is robust to model misspecification
  • Has negligible inference time latency
  • Retains interpretability for model diagnostics
  • State-of-the-art (SOTA) performance on internal and public datasets
  • Can provide more accurate point forecast through Bagging
Outline

- Introduction
- **Method**
- Experiments
- Summary
Backtesting
Backtesting (cont.)

Training data: \( \mathcal{D} = \{(X_t^i, Y_t^i)\}_{i=1,2,...,n}^{t=s_i, s_i+1,...,d_i} \)

Split points: \( j = a, a + l, a + 2l, \ldots, \max_i(d_i) - 1 \)

Training split: \( \mathcal{A}_j = \{(X_t^i, Y_t^i) \in \mathcal{D} | t \leq j\} \)

Test split: \( \mathcal{B}_j = \{(X_t^i, Y_t^i) \in \mathcal{D} | t > j\} \)

Predictive residuals from one split: \( \{Y_t^i - \hat{f}_j(Y^{s_i:j}_{i}, X^{s_i:j}_{i}, X^{(j+1):t}_{i}) | (X_t^i, Y_t^i) \in \mathcal{B}_j\} \)
Residual Selection

Training Data → Backtester → Predictive Residual Collection

PF Model → Trained PF Model

Training Data → Residual Selector

Forecasting

Test/New Data → Trained PF Model → Point Forecast

Predictive Residual Collection → Trained Residual Selector

Selected Residuals → Bootstrap → Distribution Forecast
Residual Selection (cont.)

- Heuristics-based residual selection:
  - time series ID: \( g(\mathcal{E}, \mathcal{M}, \mathcal{M}_{fut}) = \{ \varepsilon_{l,j}^t \in \mathcal{E} \mid l = i \} \) for time series \( i \)
  - time gap: \( g(\mathcal{E}, \mathcal{M}, \mathcal{M}_{fut}) = \{ \varepsilon_{l,j}^t \in \mathcal{E} \mid t - j = k_i \} \)
  - PF magnitude: \( g(\mathcal{E}, \mathcal{M}, \mathcal{M}_{fut}) = \{ \varepsilon_{l,j}^t \in \mathcal{E} \mid \hat{Y}_{l,j}^t \in (\hat{Y}_{i}^{d_i+k_i} \cdot \frac{1}{\lambda}, \hat{Y}_{i}^{d_i+k_i} \cdot \lambda) \} \)
  - discount ratio, price...

- Algorithm-based residual selection:
  - dCor + threshold search + Kolmogorov-Smirnov test
  - Fit a model to predict residuals from meta information
Bootstrapping
Bootstrapping (cont.)

- First obtain point forecast \( \hat{Y}_{i,d_i+1} = f(Y_{i,s_i:d_i}, X_{i,s_i:d_i}, X_{i,d_i+1}) \) and selected residuals \( \mathcal{G} = \hat{g}(\mathcal{E}, \mathcal{M}, \mathcal{M}_{i,d_i+1}) \)
- For \( b = 1, 2, \ldots, B \), draw \( \varepsilon_b \in \mathcal{G} \)
- Generate 1-step bootstrap forecast:
  - Backtest-Additive:
    \[
    \hat{Y}_{i,b,\text{Add.},d_i+1} = \hat{Y}_{i,d_i+1} + \varepsilon_b
    \]
  - Backtest-Multiplicative:
    \[
    r_b = \varepsilon_b / \hat{Y}_b
    \]
    \[
    \hat{Y}_{i,b,\text{Multi.},d_i+1} = \hat{Y}_{i,d_i+1} \cdot (1 + r_b) = \hat{Y}_{i,d_i+1} + \hat{Y}_{i,d_i+1} / \hat{Y}_b \cdot \varepsilon_b
    \]
Practical Considerations

• Backtest and residual selection steps can be efficiently parallelized
• Negligible inference latency to obtain distribution forecast given point forecast
• Can generate quantile forecast for arbitrary quantiles w/o retraining
• Retains interpretability for model diagnostics
Outline

• Introduction
• Method
• **Experiments**
• Summary
Setup

• Data:
  • Sales data from Amazon.com
    • Between 01/01/2017 and 01/10/2021
    • 76 products
    • 147 covariates capturing information on pricing, supply constraints, trend, seasonality, special events, and product attributes
  • M4-hourly competition data (Makridakis 2018)
• 100-fold backtest for evaluation, separate from backtest for computing residuals
• Evaluation metric: Absolute Coverage Error (ACE):
  \[
  \text{CO}(\mathcal{D}_{\text{test}}; \tau) = \frac{1}{|\mathcal{D}_{\text{test}}|} \sum_{\mathcal{D}_{\text{test}}} I\{Y_i^t \leq \hat{Y}_i^t(\tau)\}
  \]
  \[
  \text{ACE}(\mathcal{D}_{\text{test}}; \tau) = |\text{CO}(\mathcal{D}_{\text{test}}; \tau) - \tau|
  \]
• Results averaged across backtest folds, 24-week/48-hour horizon for Sales/M4 data, 10 seeds for deep learning models, and target quantiles 0.1, 0.2, ..., 0.9.
Comparison Against Classic Bootstrap Approaches

- Compare the proposed Backtest-Additive (BA) and Backtest-Multiplicative (BM) with bootstrap with fitted residuals (FR) ([Hyndman 2018](#)) and bootstrap with fitted models (FM) ([Pan 2016](#)).

**Table 1:** ACE comparison of different bootstrap DF approaches integrated with different PF models.

<table>
<thead>
<tr>
<th>Bootstrap\PF</th>
<th>Ridge</th>
<th>SVR</th>
<th>RF</th>
<th>NN</th>
</tr>
</thead>
<tbody>
<tr>
<td>FR</td>
<td>0.102(−0%)</td>
<td>0.195(−0%)</td>
<td>0.207(−0%)</td>
<td>0.176(−0%)</td>
</tr>
<tr>
<td>FM</td>
<td>0.095(−7%)</td>
<td>0.218(+12%)</td>
<td>0.171(−17%)</td>
<td>0.125(−29%)</td>
</tr>
<tr>
<td><strong>BA</strong></td>
<td>0.069(−32%)</td>
<td>0.065(−67%)</td>
<td>0.055(−73%)</td>
<td>0.077(−56%)</td>
</tr>
<tr>
<td><strong>BM</strong></td>
<td><strong>0.038(−63%)</strong></td>
<td><strong>0.061(−69%)</strong></td>
<td><strong>0.027(−87%)</strong></td>
<td><strong>0.048(−73%)</strong></td>
</tr>
</tbody>
</table>
Comparison Against SOTA Approaches

- Compare the proposed bootstrap methods with SOTA approaches including Quantile Lasso, Quantile Gradient Boosting, DeepAR (Salinas 2020), Deep Factors (Wang 2019), MQ-CNN (Wen 2017), DSSM (Rangapuram 2018), and TFT (Lim 2021).

Table 2: ACE comparison of backtest-based bootstrap integrated with the median forecast vs the default DF.

<table>
<thead>
<tr>
<th>DF</th>
<th>QLasso</th>
<th>QGB</th>
<th>DeepAR</th>
<th>DFact</th>
<th>MQCNN</th>
<th>DSSM</th>
<th>TFT</th>
</tr>
</thead>
<tbody>
<tr>
<td>Default</td>
<td>0.188</td>
<td>0.119</td>
<td>0.102</td>
<td>0.098</td>
<td>0.092</td>
<td>0.136</td>
<td>0.067</td>
</tr>
<tr>
<td>Median + BA</td>
<td>0.114</td>
<td>0.078</td>
<td><strong>0.100</strong></td>
<td><strong>0.067</strong></td>
<td>0.078</td>
<td>0.124</td>
<td><strong>0.058</strong></td>
</tr>
<tr>
<td>Median + BM</td>
<td><strong>0.039</strong></td>
<td><strong>0.036</strong></td>
<td>0.104</td>
<td>0.070</td>
<td><strong>0.071</strong></td>
<td><strong>0.112</strong></td>
<td>0.060</td>
</tr>
</tbody>
</table>
Robustness Against Model Assumptions

Table 3: ACE comparison of backtest-based bootstrap integrated with the median forecast vs the default DF from DeepAR under different pre-specified output distributions.

<table>
<thead>
<tr>
<th>DF\Output Dist.</th>
<th>Neg. Bin.</th>
<th>Student’s t</th>
<th>Normal</th>
<th>Gamma</th>
<th>Laplace</th>
<th>Poisson</th>
</tr>
</thead>
<tbody>
<tr>
<td>Default</td>
<td>0.102</td>
<td>0.192</td>
<td>0.162</td>
<td>0.138</td>
<td>0.114</td>
<td>0.134</td>
</tr>
<tr>
<td>Median + BA</td>
<td>0.100</td>
<td>0.169</td>
<td>0.116</td>
<td>0.157</td>
<td>0.094</td>
<td>0.128</td>
</tr>
<tr>
<td>Median + BM</td>
<td>0.104</td>
<td>0.165</td>
<td>0.111</td>
<td>0.156</td>
<td>0.088</td>
<td>0.125</td>
</tr>
</tbody>
</table>
Improving Accuracy of Point Forecast via Bagging

Table 4: Relative change in MAPE for Bagging PF compared to the original PF.

<table>
<thead>
<tr>
<th>Bootstrap \ PF Model</th>
<th>Ridge</th>
<th>SVR</th>
<th>RF</th>
<th>NN</th>
</tr>
</thead>
<tbody>
<tr>
<td>FR</td>
<td>+0.8%</td>
<td>+6.5%</td>
<td>+0.2%</td>
<td>+0.7%</td>
</tr>
<tr>
<td>FM</td>
<td>+0.4%</td>
<td>+6.6%</td>
<td>-3.8%</td>
<td>+2.6%</td>
</tr>
<tr>
<td><strong>BA</strong></td>
<td>-12.3%</td>
<td>-21.0%</td>
<td><strong>-10.0%</strong></td>
<td>+1.5%</td>
</tr>
<tr>
<td><strong>BM</strong></td>
<td>-22.1%</td>
<td>-31.8%</td>
<td>-5.3%</td>
<td>-13.4%</td>
</tr>
</tbody>
</table>
Outline

• Introduction
• Method
• Experiments
• Summary
Summary

• Proposed a Distribution Forecast framework with the following advantages:
  • Incorporates different sources of forecast uncertainty by design
  • Integrates well with an arbitrary PF model to produce DF
  • Is robust to model misspecification
  • Has negligible inference time latency
  • Retains interpretability for model diagnostics
  • State-of-the-art (SOTA) performance on internal and public datasets
  • Can provide more accurate point forecast through Bagging
Thank you!
longsha@amazon.com