Hierarchical forecasts: a case study from pricing in e-commerce

IIF Workshop on Forecast Reconciliation

Stefan Birr, Adele Gouttes
Zalando SE

Overview

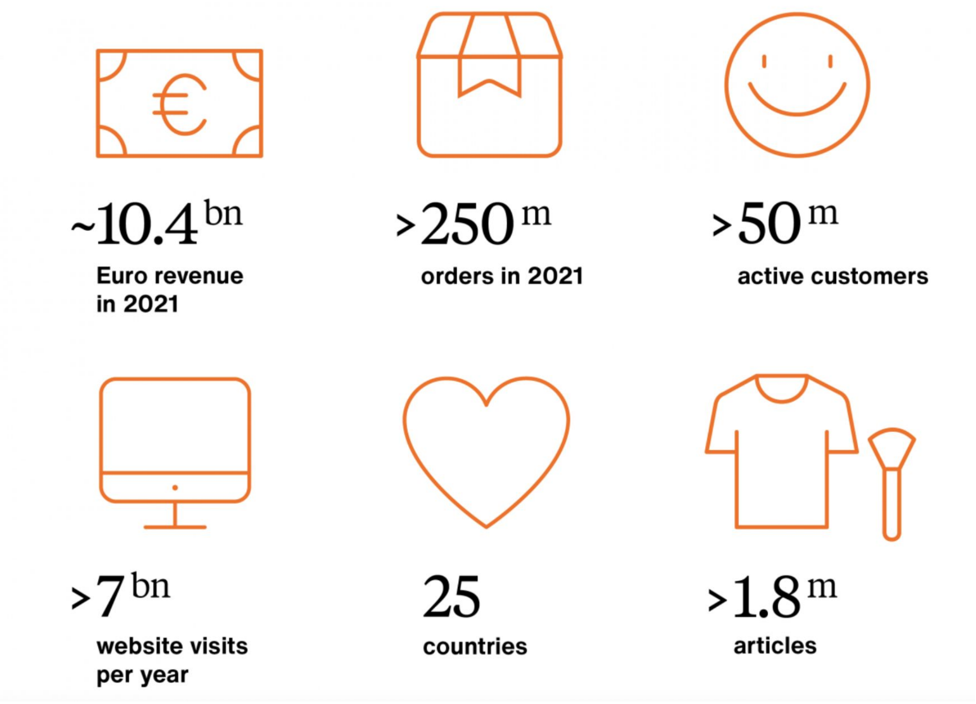
1. Introduction to our use case
2. Why hierarchical forecast?
3. How to reconcile our forecast?
4. Experiment: Middle out with forecast proportions
5. Experiment: Adapting MinT for our use
6. Challenge: Reconcile a forecast grid
7. Conclusion

Overview

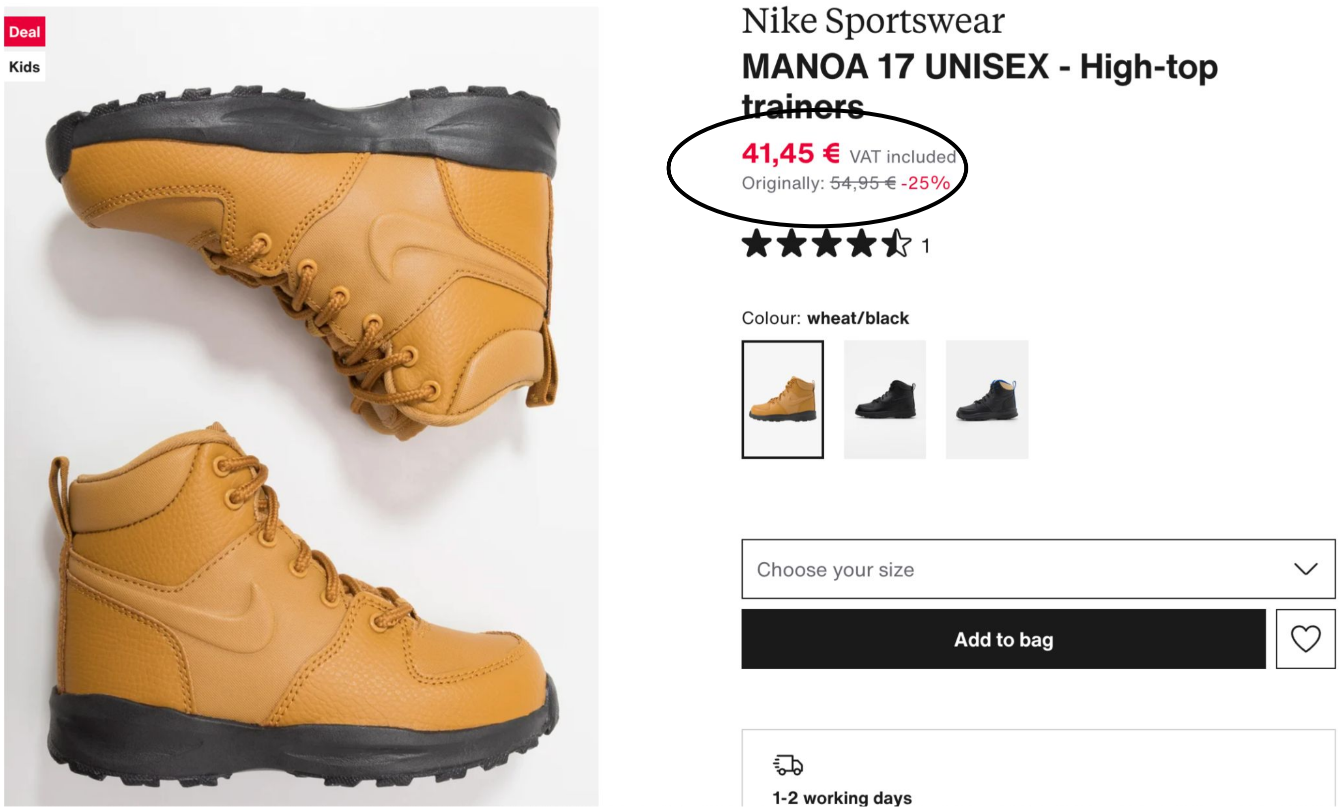
1. **Introduction to our use case**
2. Why hierarchical forecast?
3. How to reconcile our forecast?
4. Experiment: Middle out with forecast proportions
5. Experiment: Adapting MinT for our use
6. Challenge: Reconcile a forecast grid
7. Conclusion

Intro: Zalando

Fashion e-commerce in Europe



Algorithmic Pricing



Algorithmic Pricing @ Zalando

Our Sales Forecast:

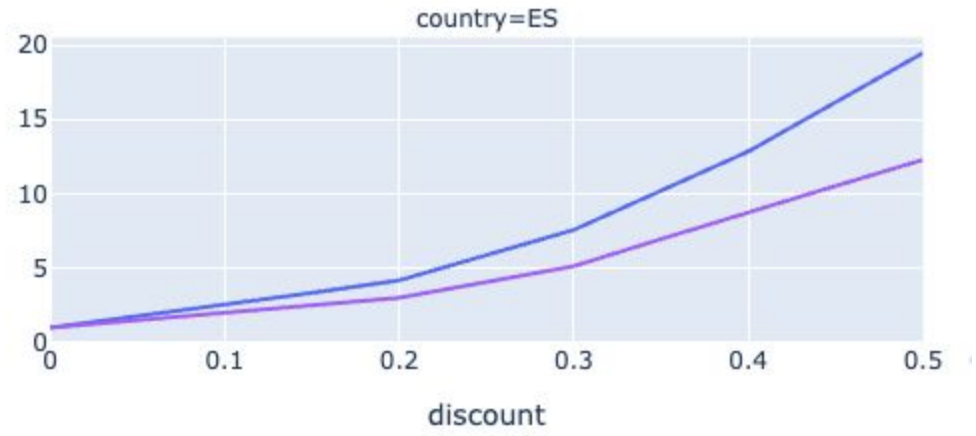
“How many items of article X would be bought in country Y in the next Z days
if we would give D% discount?”

For simplicity assume:

- One country only
- Forecast 2 time steps, daily resolution
- Discounts can be [0, 10, 20, 30, 40]

Output of our forecaster: Demand Curves

We predict the demand for each day and each article in dependence of the discount



Predicted demand for an article on two different days

Overview

1. Introduction to our use case
2. **Why hierarchical forecast?**
3. How to reconcile our forecast?
4. Experiment: Middle out with forecast proportions
5. Experiment: Adapting MinT for our use
6. Challenge: Reconcile a forecast grid
7. Conclusion

How we measure forecast performance

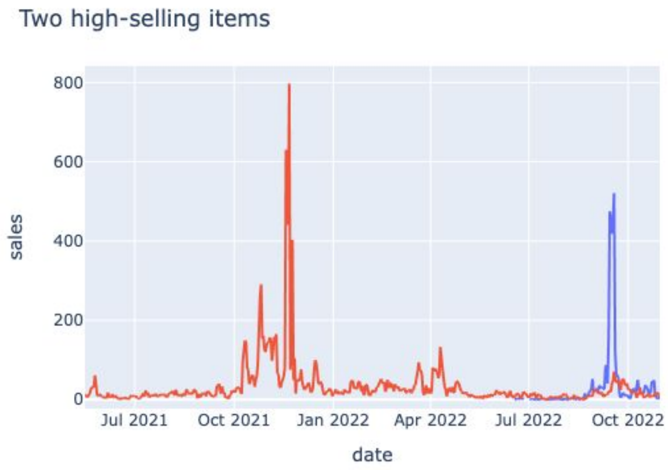
Metric: weighted and scaled RMSE on Sales

Model is used for pricing decisions at **two levels**:

- **Item:** Forecast for different prices, choose the best price
- **Country:** Need to match the expected sales targets

How our data looks like

- Items appear and disappear (~2% new items every week)
- ~350K time series per day
- Very different nature of time series:
  - High-sellers: Nice seasonality (for some of them)
  - Low-sellers: Very noisy



Our two forecast models

- Weekly frequency, 26 weeks forecast horizon: Transformer-based[1]
- ***Daily frequency, 7 days forecast horizon: LightGBM-based***

About the daily forecast (LightGBM):

Good

- Accurate at item-level
- Quick to train and iterate
- Easy to add/remove features

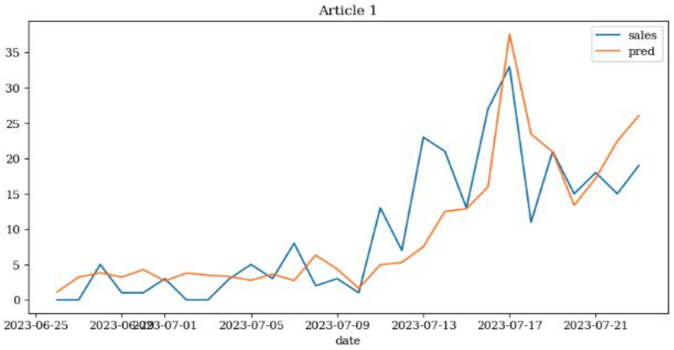
Bad

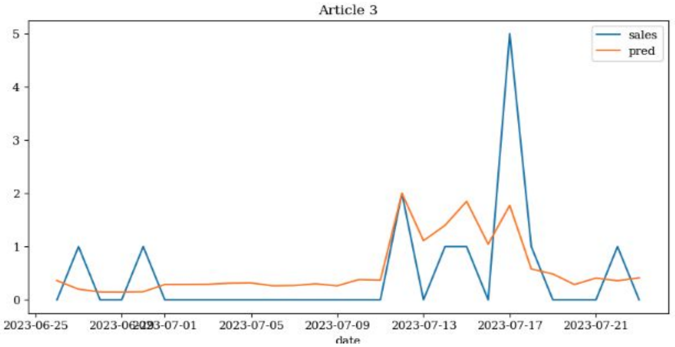
- Very inaccurate when aggregating bottom-up
- Biased forecast: can't predict less than 0 on low-sellers
- Statistical models are better at catching aggregated seasonality
- Trained on short history ~ 2 years

[1] Kunz et al. (2022) Deep Learning based Forecasting: a case study from the online fashion industry

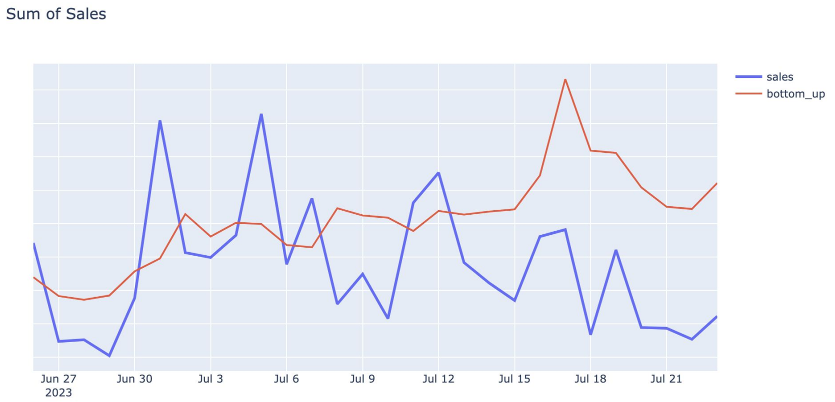
While we are doing good on article level, the aggregated forecast is lacking signal.

Article-level





Country-level



Overview

How we approached hierarchical forecasting

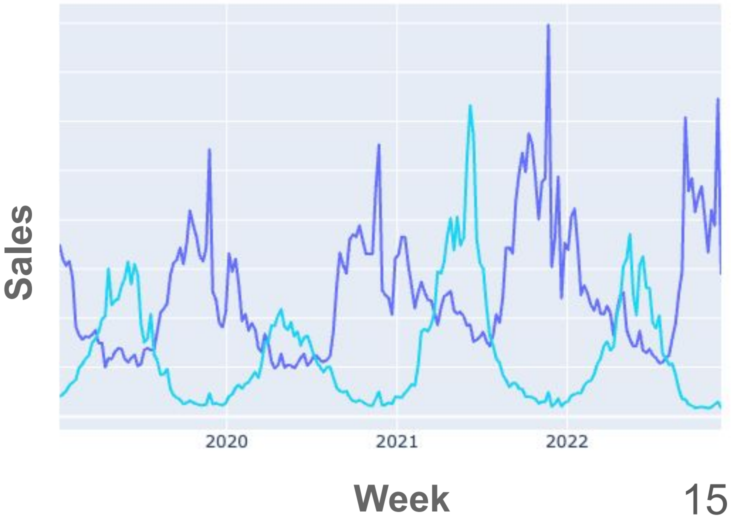
- Methods we are testing:
  - Baseline: Bottom-up
  - Basic: Top down approaches, middle out
  - MinT[1]
  - RNN based end2end method[2]
- Use Python: ***nixtla/hierarchicalforecast*** or GluonTS
- Success: we improve one of our three metrics and keep the rest stable
  - Item-level accuracy
  - Country-level bias
  - Country-level accuracy

[1]Wickramasuriya et al. "Optimal forecast reconciliation for hierarchical and grouped time series through trace minimization"

[2]Rangapuram et al. End-to-End Learning of Coherent Probabilistic Forecasts for Hierarchical Time Series

How to reconcile: The hierarchy

- Focus is on top- and bottom-level: country and item
- But we still want to use some middle hierarchy because:
  - Items are very heterogeneous
  - From country to item, there is a 1/350K ratio…
- Choosing the structure:
  - quality of the seasonality patterns
  - not too sparse (avoid series with many 0s)
  - reasonable ratio between parent/child levels



How to reconcile: The hierarchy

Mix of hierarchical/group structure:

- 0 - clothing, shoes, beauty items…
- 1 - season: autumn, summer, winter…
- 2 - details: coats, trousers…
- 3 - more details: winter coat, legging…

			
Category	Shoes	Clothes	Clothing
Season	Winter	Summer	Summer
Detail 1	Boots	T-shirt	Swimwear
Detail 2	Winter boots	Polo shirt	Swimsuits

Overview

1. Introduction to our use case
2. Why hierarchical forecast?
3. How to reconcile our forecast?
4. **Experiment: Middle out with forecast proportions**
5. Experiment: Adapting MinT for our use
6. Challenge: Reconcile a forecast grid
7. Conclusion

Experiment: Middle-out with forecast proportions

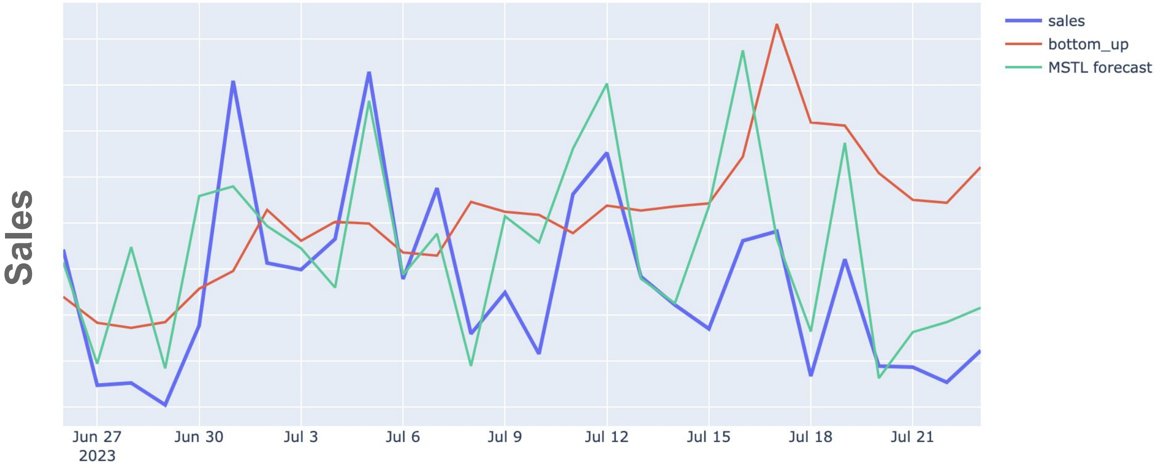
- First approach **Middle-out** from categories (Shoes, Accessoires, …)
  - Easy to scale to a large dataset
  - No problem with unbalanced data
- Some features we have are defined on an aggregated level:
  - Sales event that is happening on specific days in one or more countries
  - Promotion/vouchers that target a subgroup of our assortment

Idea: Create a forecast on aggregated level that utilises these features and use hierarchical forecasting to bring it down to the article level

Experiment with Middle-Out

On aggregate level the MSTL forecast clearly models the dynamics of the time series better, but the bottom up forecast is ‘randomly’ better on some dates

- Daily Forecast
- Model at article level uses lightgbm
- Model at aggregated level uses MSTL (yearly + weekly seasonality)



Middle-out: Results at country level

- Middle-out (forecast proportion + MSTL) clearly outperforms bottom-up
- The choice of the upper model matters

Scaled RMSE on **total sales** level (Top Level)

grid_date	2023-06-22	2023-06-29	2023-07-06	2023-07-13
middle-out arima	0.44	0.17	0.31	0.33
middle-out mstl	0.27	0.12	0.30	0.21
tree	0.29	0.24	0.28	0.73

Middle-out: Results at Article-level

- Performance of reconciled forecast similar to baseline
- Clear improvement on the last week

Scaled RMSE on **article** level (Bottom Level)

grid_date	2023-06-22	2023-06-29	2023-07-06	2023-07-13
middle-out arima	0.56	0.57	0.6	0.56
middle-out mstl	0.57	0.61	0.6	0.56
tree	0.53	0.57	0.6	0.65

Overview

1. Introduction to our use case
2. Why hierarchical forecast?
3. How to reconcile our forecast?
4. Experiment: Middle out with forecast proportions
5. **Experiment: Adapting MinT for our use**
6. Challenge: Reconcile a forecast grid
7. Conclusion

Experiment: Using MinT for reconciliation

Why?

- Use forecasts at all levels
- Learns from the forecast errors -> should get more improvement

Problems

- Unbalanced: what to do with series of different lengths?
- Our forecast is biased on low-sellers
- 350K series:
    - MUCH MORE than what the original implementation could handle
    - Will the covariance matrix of the error be informative?

Experiment: MinT - Unbalanced set

- Focus on upper levels (0-1-2): unbiased, balanced, small number of series.
- Backtesting over 1 year, 5 week rolling window.

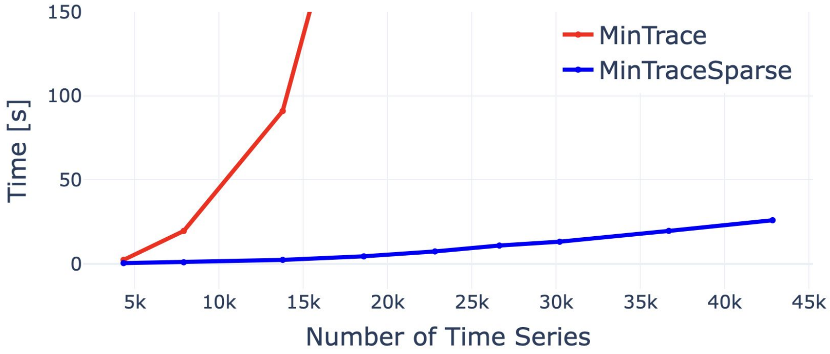
Level	Baseline	MinT - OLS	MinT - WLS	MinT - Shrink
0	12.31	12.16	10.10	**9.66**
1	13.74	13.92	11.80	**11.50**
2	15.73	16.66	15.32	**15.18**
3	**20.06**	20.86	20.23	20.26

Experiment: MinT - Handling 350K time series

- Optimise for memory usage:
  - Unless you have a very big machine, you SHOULDN'T load the full covariance matrix in memory
  - We use sparse matrix computation: works for methods without shrinkage

→ Big thanks to ***Mateusz Koren***:

Work in collaboration with ***Nixtla,*** changes are already available



Overview

1. Introduction to our use case
2. Why hierarchical forecast?
3. How to reconcile our forecast?
4. Experiment: Middle out with forecast proportions
5. Experiment: Adapting MinT for our use
6. **Challenge: Reconcile a forecast grid**
7. Conclusion

Functional reconciliation

Problem:

Article level: depends on the price

Aggregated forecasts: depend on the **overall discount level** (we use a sales weighted average to measure this)

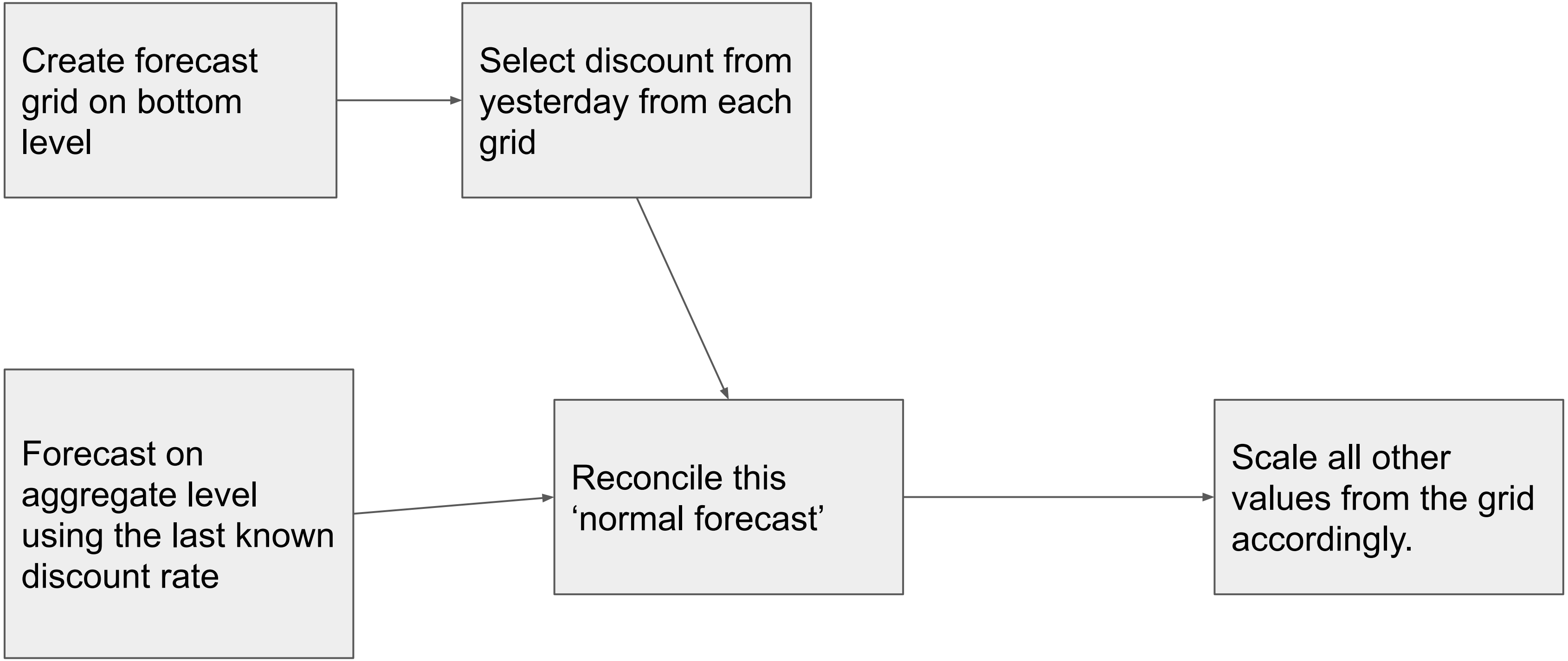
For future values this is unknown[1]

Potential strategies:

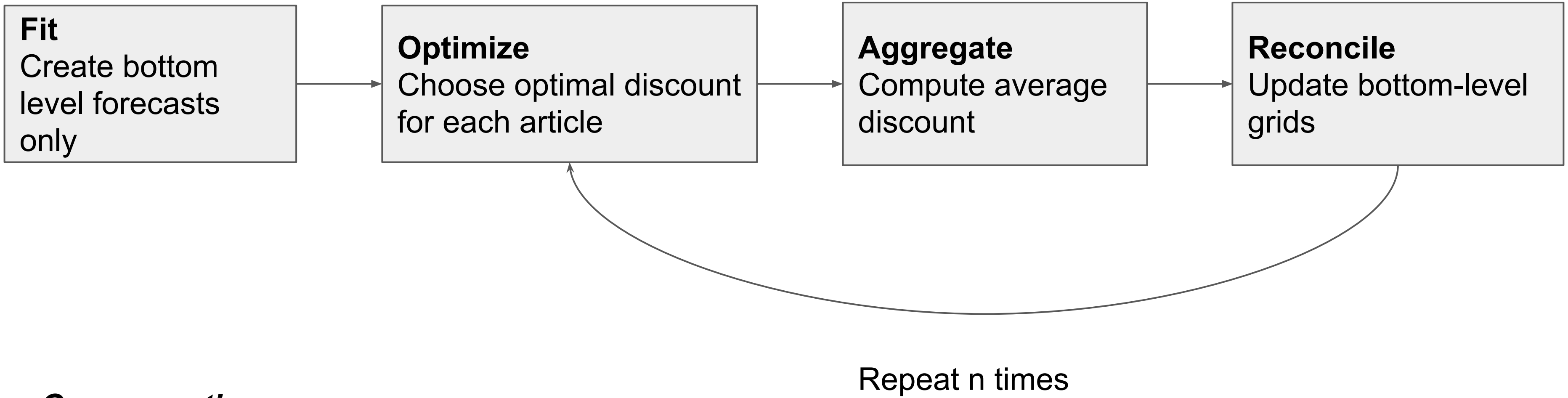
1) Create a separate forecast for the discount level for each hierarchy
2) Scaling the grid
3) Iterative Reconciliation

[1]In our experiments and backtesting we know the materialized discounts and can use them

Functional reconciliation: Scaling grids



Functional reconciliation: Iterative



Open question:

- Does it converge?
- How long does it need?

Overview

Conclusion

- Very low-level forecasts perform poorly on aggregated level: forecast reconciliation can help a lot here
- Basic methods like Middle-out with forecast proportion shows promising results

Practical challenges:

- High turnover in the product catalog: the number of products over which we aggregate varies over time.
- For multi-level methods (MinT, Rangapur et al), the scale of our data is a challenge

Challenges

Hierarchical pricing: The forecast is a function of price

- We might need to predict the future price, so a bi-variate forecasts (price & demand) may be ideal. What's the best way to aggregate price across a hierarchy? Or promotional data?
- Pricing effects might be different at country- and item-level: The sales of one item might be influenced by the sales of other items. How do we best reconcile them in a forecast setting?