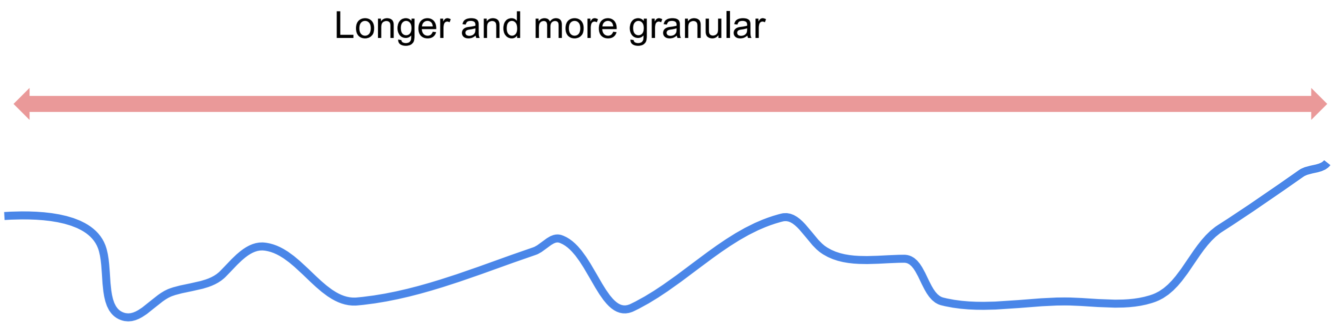
Boosted learning on level imbalance data through hierarchical data augmentation

Authors: Weijie Shen, Steve Thomas

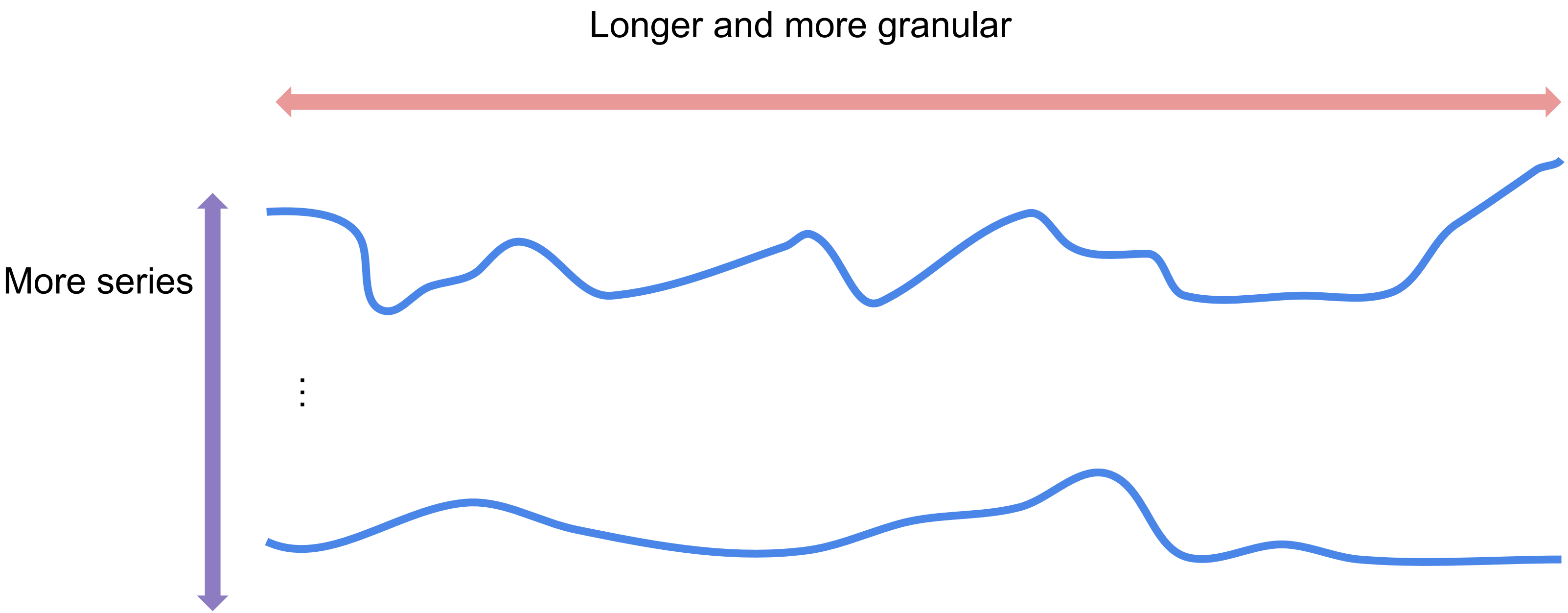
Presenter: Casey Lichtendahl

2023 IIF Workshop on Forecast Reconciliation, 2023-09-08

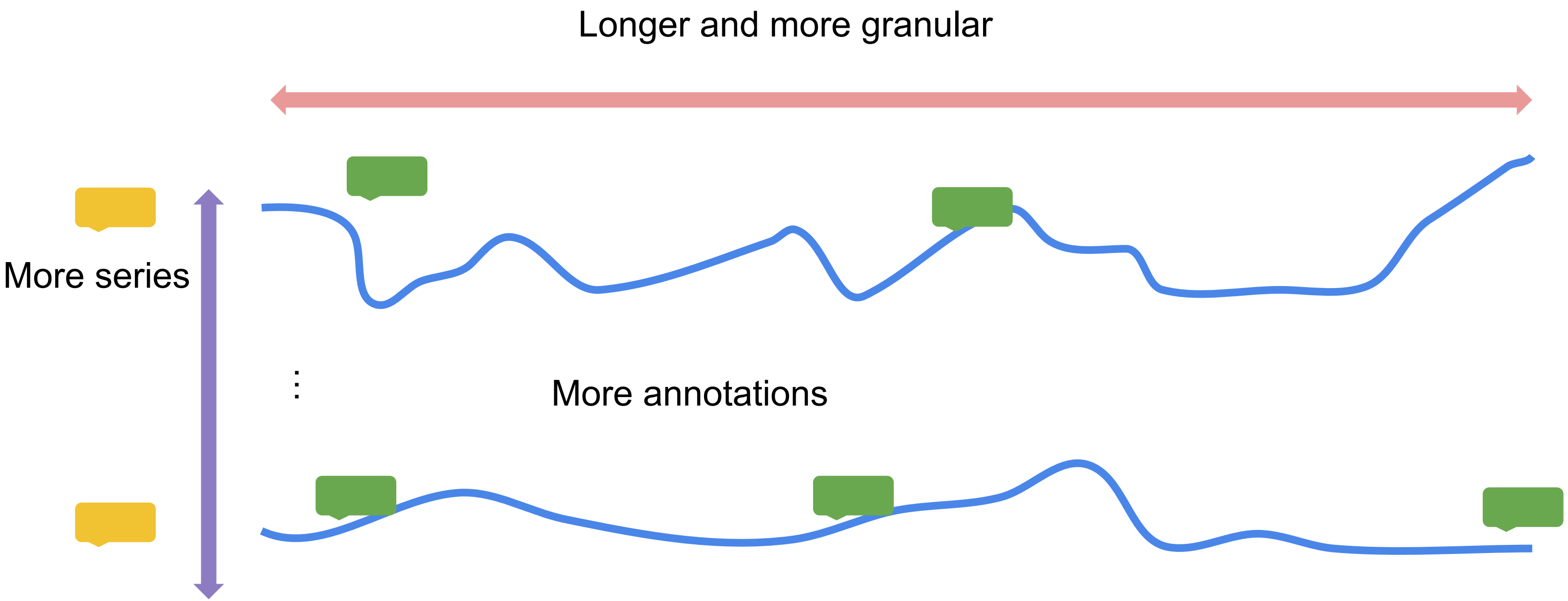
Big data in time series



Big data in time series



Big data in time series



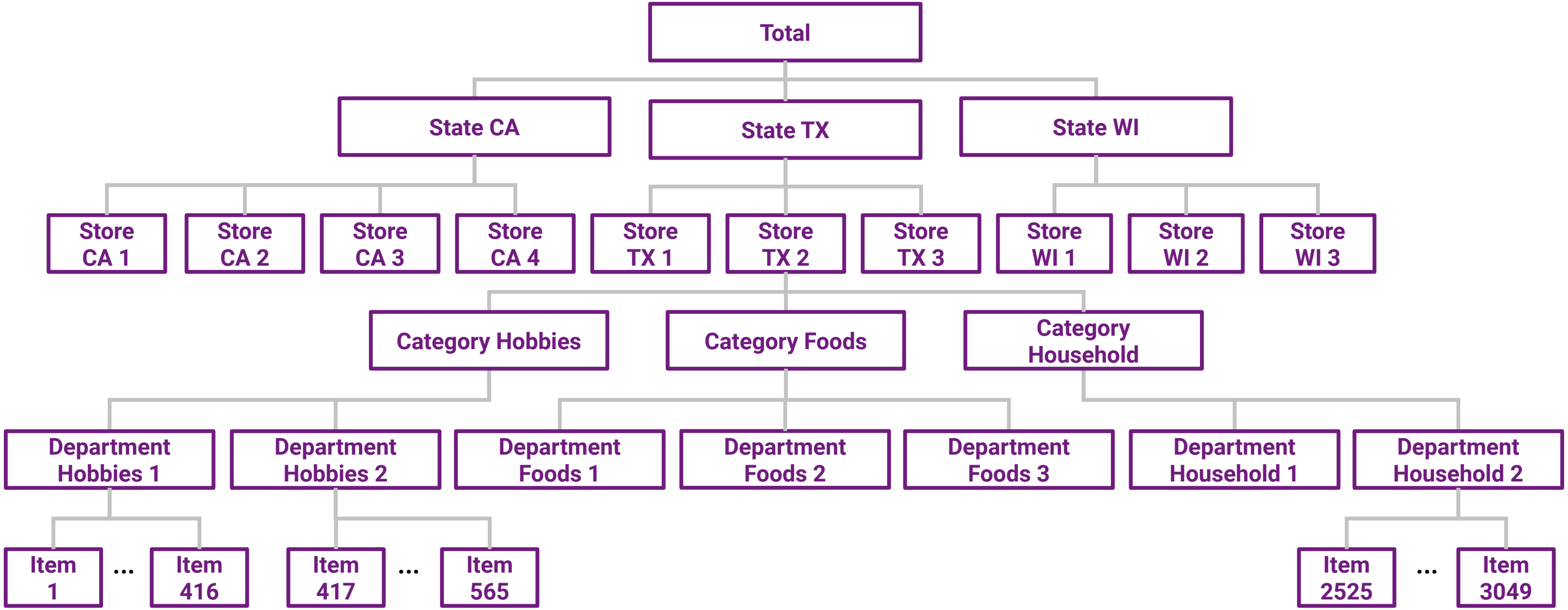
Static annotations lead to deep hierarchies

Store	State	Item	department	category	2011-01-29	…	2016-06-19
CA1	CA	1	Hobby 1	Hobby	xx		xx
…							
TX3	TX	100	Food 2	Food	xx		xx

Source: The M5 Competition Competitors' Guide

Static annotations lead to deep hierarchies

Store	State	Item	department	category	2011-01-29	…	2016-06-19



Google

Data source: Makridakis, S., Spiliotis, E., & Assimakopoulos, V. (2020a)

Static annotations lead to deep hierarchies

Store	State	Item	department	category	2011-01-29	…	2016-06-19
xx	CA	1	Hobby 1	Hobby	xx		xx
…							
TX3	TX	xx	xx	Food	xx		xx

Higher level series are more important because they are more

- Stable
- Inspiring
- Actionable

Google

Static annotations lead to deep hierarchies

Store	State	Item	department	category	2011-01-29	…	2016-06-19
xx	CA	1	Hobby 1	Hobby	xx		xx
…							
TX3	TX	xx	xx	Food	xx		xx

Higher level series are more important because they are more

- Stable
- Inspiring
- Actionable

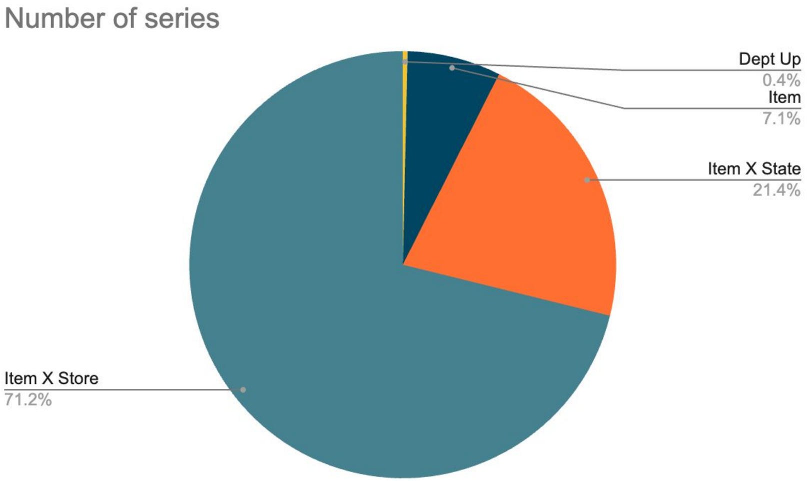
but

Higher level series are fewer in counts by orders of magnitude.

- Good for statistical models
- Bad for ML models

ML on level imbalance data

Level id	Aggregation Level	Number of Series	Proportion
1	All	1	0.36%
2	State	3	
3	Store	10	
4	Category	3	
5	Department	7	
6	State × Category	9	
7	State × Department	21	
8	Store × Category	30	
9	Store × Department	70	
10	Item	3049	7.1%
11	Item × State	9147	21.4%
12	Item × Store	30490	71.2%



Most series in a batch will be item-specific.

Use metrics like raw RMSE doesn't solve sampling inefficiencies.

Data source: Makridakis, S., Spiliotis, E., & Assimakopoulos, V. (2020a)

Solution: get more top series

Typical time series augmentation:

- Bootstrap noise ([1] C. Bergmeir, R. J. Hyndman, J. M. Benitez 2016)
- Transformation ([2] Q. Wen et. al 2022)
- Frequency domain
- …

E.g.

- Decomposing series and bootstrap residuals.
- Injecting white noises, spikes, steps, slopes.
- Cropping, slicing, warping, flipping.
- Amplitude and phase perturbations.
- Shuffling, averaging, masking time series features.

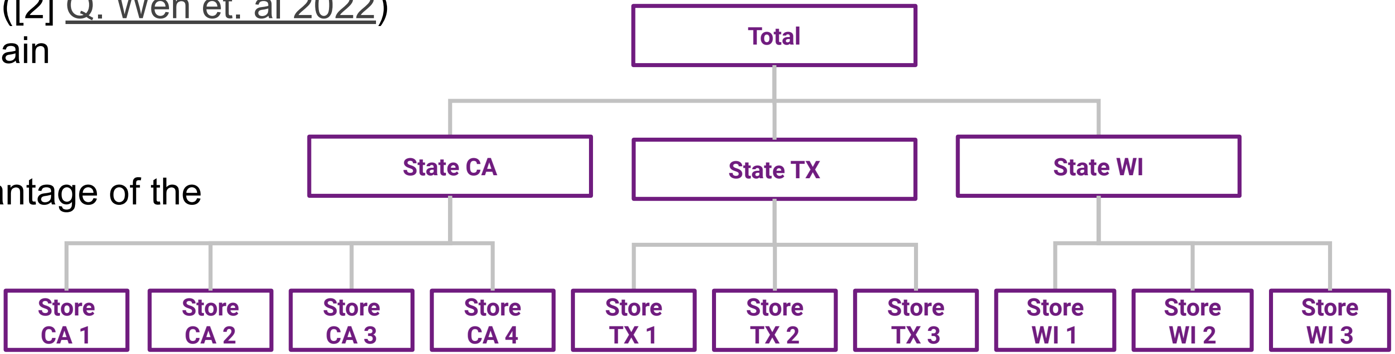
Source [1]: C. Bergmeir, R. J. Hyndman, J. M. Benitez 2016 Bagging exponential smoothing methods using STL decomposition and Box–Cox transformation

Source [2]: Q. Wen et. al 2022 Time Series Data Augmentation for Deep Learning: A Survey

Solution: get more top series

Typical time series augmentation:

- Bootstrap noise ([1] C. Bergmeir, R. J. Hyndman, J. M. Benitez 2016)
- Transformation ([2] Q. Wen et. al 2022)
- Frequency domain
- …



Q: How to take advantage of the hierarchy?
A: Adding series up

- CA1 + CA3
- CA + TX (no WI)
- CA1 + TX2 ?

Removing children
Adding sibling’s children, i.e. nibling?

Q: What state will that be?
A: It will be somewhere in between, say Arizona :)

Google

Source [1]: C. Bergmeir, R. J. Hyndman, J. M. Benitez 2016 Bagging exponential smoothing methods using STL decomposition and Box–Cox transformation

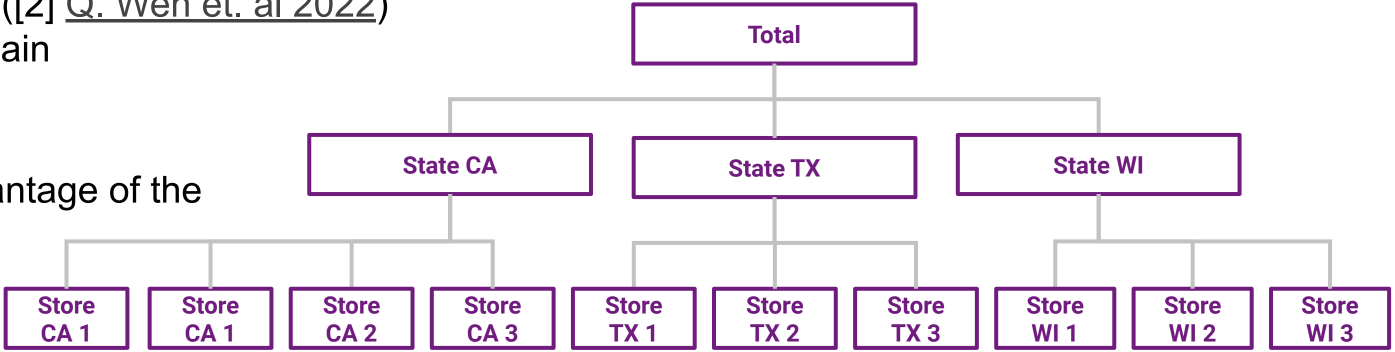
Source [2]: Q. Wen et. al 2022 Time Series Data Augmentation for Deep Learning: A Survey

Data source: Makridakis, S., Spiliotis, E., & Assimakopoulos, V. (2020a)

Solution: get more top series

Typical time series augmentation:

- Bootstrap noise ([1] C. Bergmeir, R. J. Hyndman, J. M. Benitez 2016)
- Transformation ([2] Q. Wen et. al 2022)
- Frequency domain
- …



Q: How to take advantage of the hierarchy?
A: Adding series up

- CA1 + CA3
- CA + TX (no WI)
- CA1 + TX2 ?

Removing children
Adding sibling's children, i.e. nibling?

Q: What state will that be?
A: ~~It will be somewhere in between, say Arizona :)~~
Treats state features as continuous variables and creates weighted combinations.

Google

Source [1]: C. Bergmeir, R. J. Hyndman, J. M. Benitez 2016 Bagging exponential smoothing methods using STL decomposition and Box–Cox transformation

Source [2]: Q. Wen et. al 2022 Time Series Data Augmentation for Deep Learning: A Survey

Data source: Makridakis, S., Spiliotis, E., & Assimakopoulos, V. (2020a)

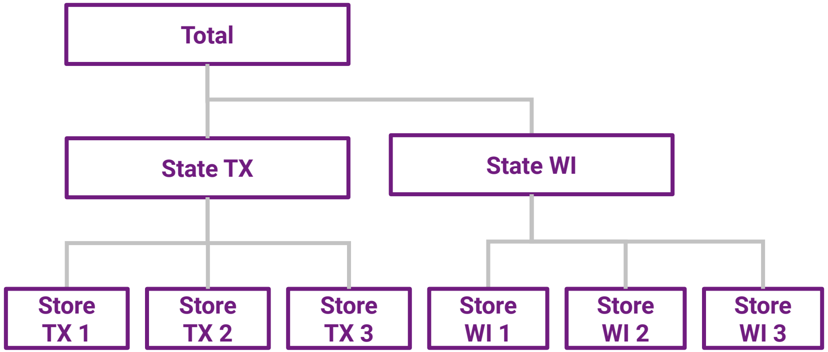
Solution: get more top series

We think of three main ways, by levels:

- Removing
- Swapping
- Random sum

Exploit
(Generate similar series)

Explore (Generate different series)



Data source: Makridakis, S., Spiliotis, E., & Assimakopoulos, V. (2020a)

Solution: get more top series

We think of three main ways, by levels:

- Removing
- Swapping
- Random sum

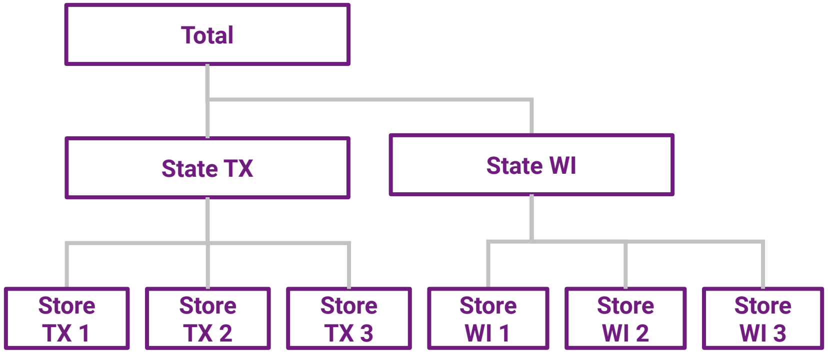
Exploit
(Generate similar series)

Explore (Generate different series)

$$Y'_{ij} = Y_i - \sum_{p \in S_{ij}} Y_p$$

Parameters to tune:

- Max / min % of children to remove



C_i is the set of children for node i

$$Y_i = \sum_{c \in C_i} Y_c , \; S_{ij} \subseteq C_i$$

Y_i is the i-th series to augment.
Y'_{ij} is the j-th augmented series for series i.
S_{ij} is the set of series to remove for series i.

Data source: Makridakis, S., Spiliotis, E., & Assimakopoulos, V. (2020a)

Solution: get more top series

We think of three main ways, by levels:

- Removing
- Swapping
- Random sum

Exploit
(Generate similar series)

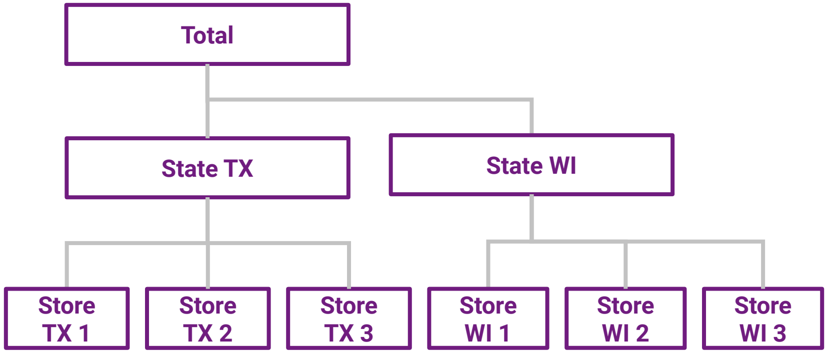
Explore (Generate different series)

$$Y'_{ij} = Y_i - \sum_{p \in S_{ij}} Y_p + \sum_{q \in A_{ij}} Y_q$$

$$A_{ij} \cap C_i = \emptyset, \; |S_{ij}| = |A_{ij}|$$

Parameters to tune:

- Max / min series to swap
- Min children size to swap



C_i is the set of children for node i

$$Y_i = \sum_{c \in C_i} Y_c, \; S_{ij} \subseteq C_i$$

Y_i is the i-th series to augment.
Y'_{ij} is the j-th augmented series for series i.
S_{ij} is the set of series to remove for series i.
A_{ij} is the set of series to add for series i.

Google

Data source: Makridakis, S., Spiliotis, E., & Assimakopoulos, V. (2020a)

Solution: get more top series

We think of three main ways, by levels:

- Removing
- Swapping
- Random sum

Exploit
(Generate similar series)

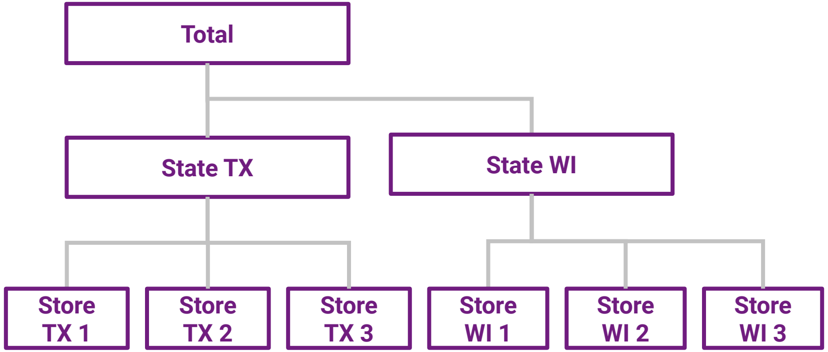
Explore (Generate different series)

$$Y'_{ij} = \sum_{c \in R_j} Y_c$$

$$R_j \subseteq \bigcup_i C_i$$

Parameters to tune:

- Max / min series to sum



C_i is the set of children for node i

$$Y_i = \sum_{c \in C_i} Y_c,\ S_{ij} \subseteq C_i$$

Y_i is the i-th series to augment.
Y'_{ij} is the j-th augmented series for series i.
R_j is the set of series to sum.

Data source: Makridakis, S., Spiliotis, E., & Assimakopoulos, V. (2020a)

M5 Experiment Setup

- Data from 2011-01-29 to 2016-05-22
- Daily backtests from 2015-05-24 to 2016-04-24
- Forecast length of 28 days
- Metrics are weighted root mean squared scaled error for P50 (WRMSSE) and weighted scaled pinball loss for P95 (WSPL)
- Backtests are aggregated the same way as horizons.
- Use a StarryNet[1] model that has comparable performance to top M5 models.
- Prediction intervals are trained and generated on top of fixed point forecasts

[1] StarryNet is a ML based forecasting algorithm developed by Google. See previous presentation at ISF.

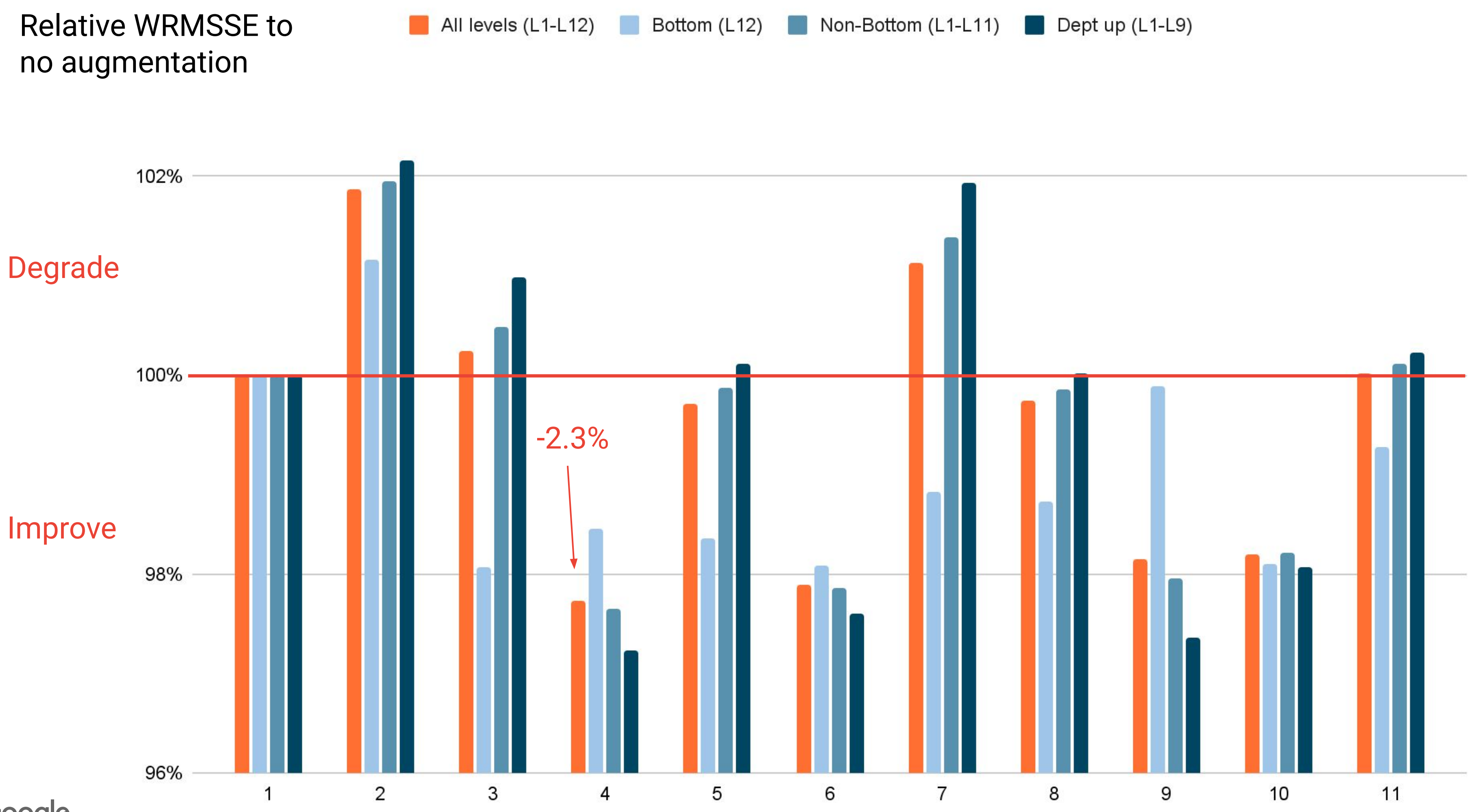
Google

M5 Experiment Results: point forecasts (WRMSSE)

Fraction of series that are augmented

Factorial

id	Sum prob	Swap prob	Remove prob	All levels (L1-L12)	Bottom (L12)	Non-Bottom (L1-L11)	Dept up (L1-L9)
1	0	0	0	0.71993	0.87562	0.70578	0.65973
2	0.3			0.73335	0.88582	0.71949	0.67394
3		0.3		0.72168	0.85878	0.70922	0.66623
4			0.3	0.70359	0.86205	0.68919	0.64151
5	0.15	0.15		0.71790	0.86118	0.70488	0.66052
6		0.15	0.15	0.70470	0.85880	0.69070	0.64393
7	0.15		0.15	0.72802	0.86535	0.71554	0.67250
8	0.1	0.1	0.1	0.71813	0.86455	0.70482	0.65980
9		0.1	0.1	0.70664	0.87464	0.69137	0.64233
10		0.2	0.2	0.70697	0.85895	0.69316	0.64701
11		0.1	0.2	0.72010	0.86925	0.70654	0.66127



Relative WRMSSE to no augmentation
All levels (L1-L12)
Bottom (L12)
Non-Bottom (L1-L11)
Dept up (L1-L9)
Degrade
Improve
102%
100%
98%
96%
-2.3%
1
2
3
4
5
6
7
8
9
10
11

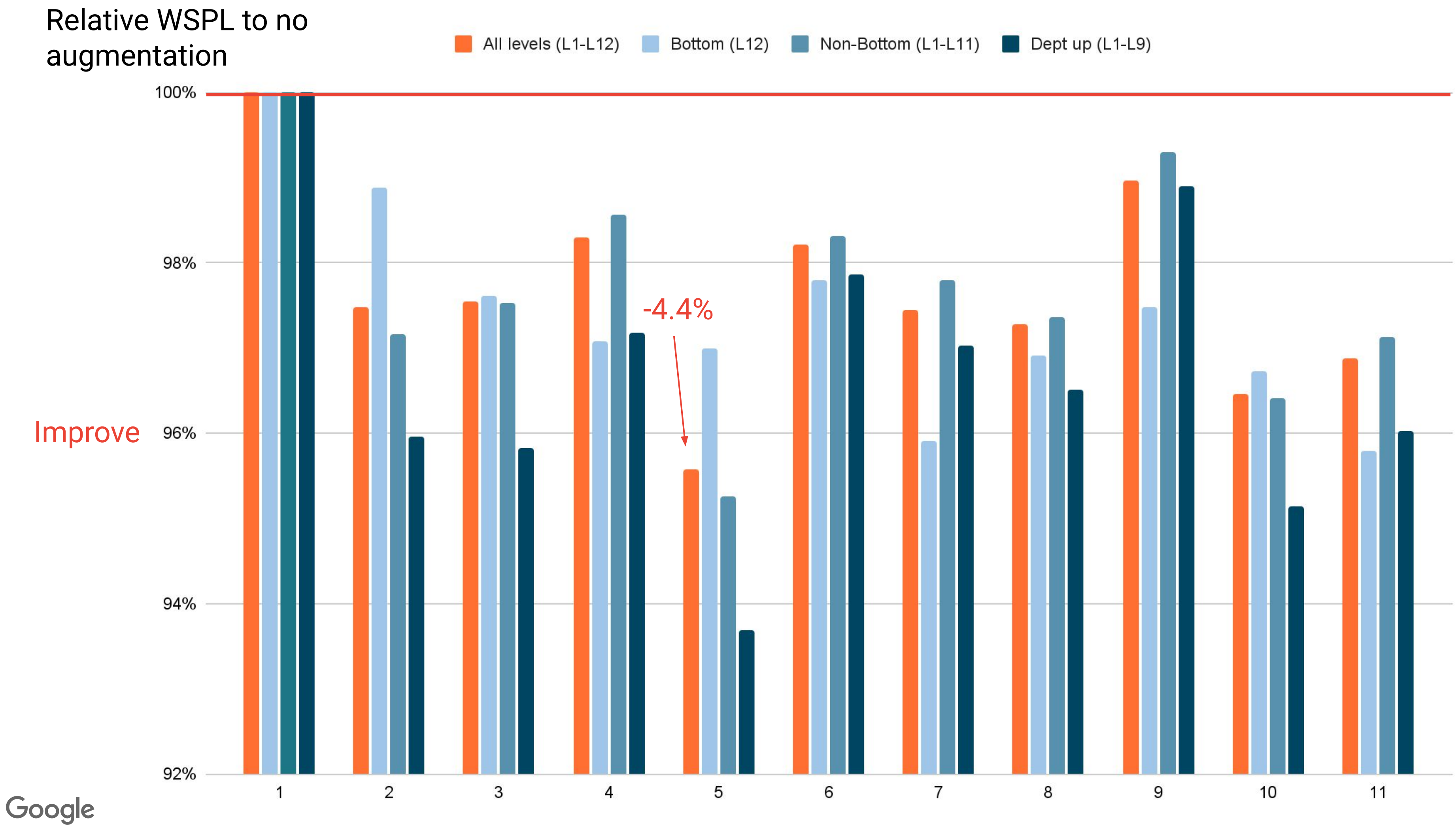
Google

M5 Experiment Results: prediction interval forecasts (WSPL)

Factorial

id	Sum prob	Swap prob	Remove prob	All levels (L1-L12)	Bottom (L12)	Non-Bottom (L1-L11)	Dept up (L1-L9)
1	0	0	0	0.14286	0.31305	0.12738	0.10706
2	0.3			0.13925	0.30953	0.12377	0.10273
3		0.3		0.13934	0.30556	0.12423	0.10260
4			0.3	0.14041	0.30393	0.12555	0.10405
5	0.15	0.15		0.13654	0.30365	0.12135	0.10030
6		0.15	0.15	0.14031	0.30616	0.12524	0.10477
7	0.15		0.15	0.13921	0.30025	0.12457	0.10387
8	0.1	0.1	0.1	0.13896	0.30338	0.12401	0.10333
9	0.1	0.1		0.14137	0.30518	0.12648	0.10589
10	0.2	0.2		0.13780	0.30280	0.12280	0.10187
11	0.1	0.2		0.13840	0.29986	0.12373	0.10281

Google


Relative WSPL to no augmentation
All levels (L1-L12)
Bottom (L12)
Non-Bottom (L1-L11)
Dept up (L1-L9)
100%
98%
96%
94%
92%
Improve
-4.4%
1
2
3
4
5
6
7
8
9
10
11
Google

Why does data augmentation improve intervals more?

Training loss for point forecasts: without augmentation

Training loss for interval forecasts: without augmentation

Google

Why does data augmentation improve intervals more?

Training loss for point forecasts: with augmentation

Training loss for interval forecasts: with augmentation

Key Takeaways

- We introduces a hierarchical data augmentation strategy with three variations for level imbalance data.
- On M5, we improve all level (L1-L12) point forecasts by 2.3% and interval forecasts by 4.4%.
- Improvements are not only for upper levels, but also for bottom levels.
- Hierarchical data augmentation helps, especially when training loss is unstable because of sampling issue.
- Interval forecasts likely benefit more from hierarchical data augmentation because of the reduced effective sample size and increased volatility of loss.
- The ideal configuration and intensity of augmentation depends on the data.