Forecasting at Uber: A Brief Survey

Andrea Pasqua

Data Science Manager - Intelligent Decision Systems - Uber

Observe, Relate, Predict

Uber

Time Series at Uber

Use Cases for Forecasting at Uber

- Time series are ubiquitous at Uber

Use Cases for Forecasting at Uber

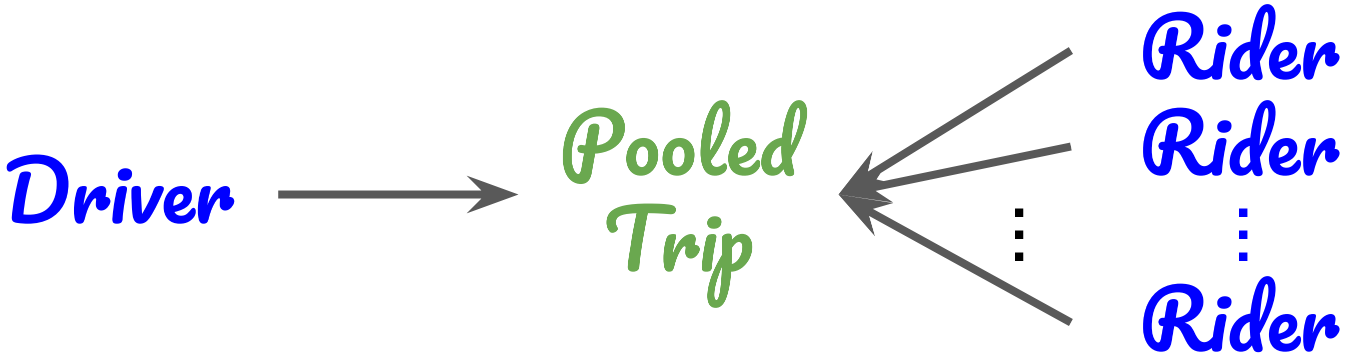
- Time series are ubiquitous at Uber
    - **Markets**: descriptors for supply, demand, transactions for multi-sided markets

Driver → Trip ← Rider

Supply *Transaction* *Demand*

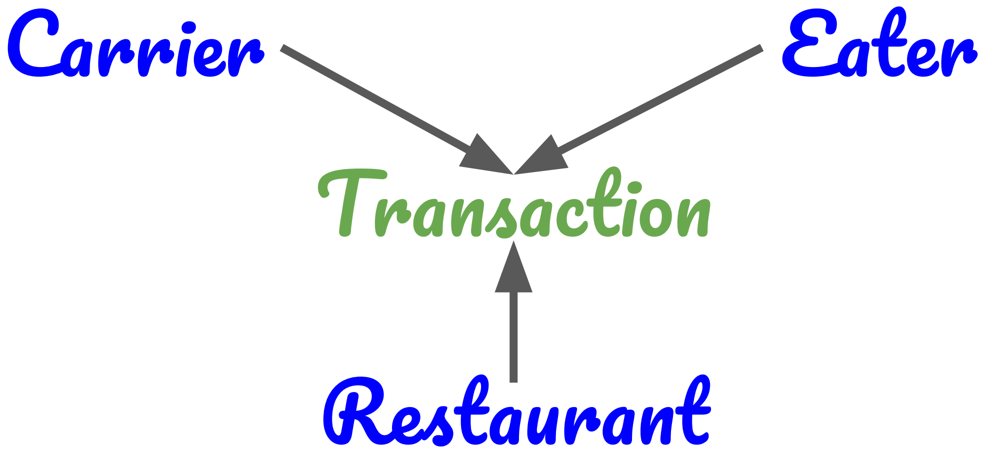
Use Cases for Forecasting at Uber

- Time series are ubiquitous at Uber
  - **Markets**: descriptors for supply, demand, transactions for multi-sided markets



Use Cases for Forecasting at Uber

- Time series are ubiquitous at Uber
  - **Markets**: descriptors for supply, demand, transactions for multi-sided markets

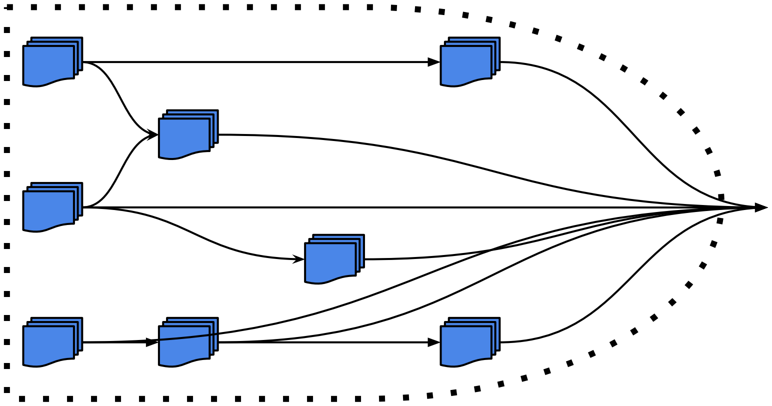


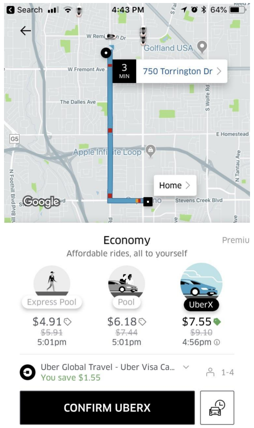
Use Cases for Forecasting at Uber

- Time series are ubiquitous at Uber
    - **Markets**
    - **Internal resources**: compute, storage and data resources

Use Cases for Forecasting at Uber

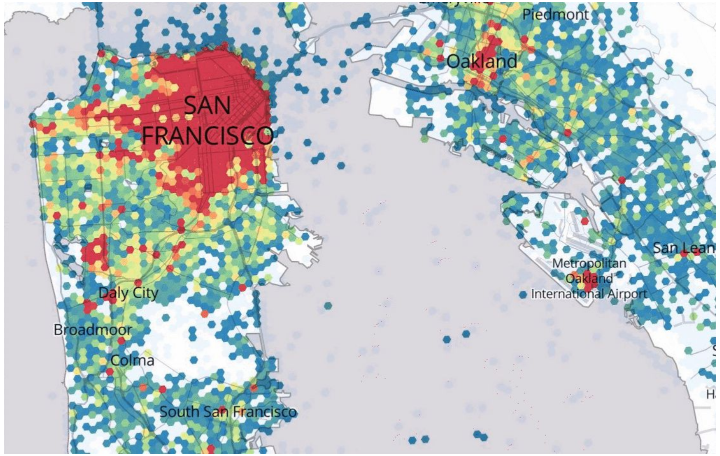
- Time series are ubiquitous at Uber
  - **Markets**
  - **Internal resources**
  - **Technical time series**: states of micro-services



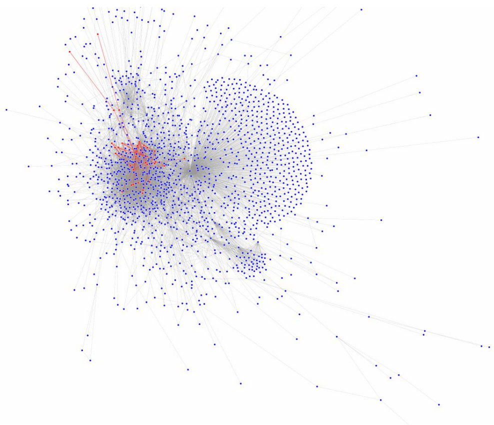


The Scale of our Domain

- Powerful combinatorics
  - By city, by neighbourhood, by market, by product, by app version
  - By micro-service, by trace, by product, by cohort



https://eng.uber.com/forecasting-introduction/



Visualization of a subset of micro-services: edges are API calls

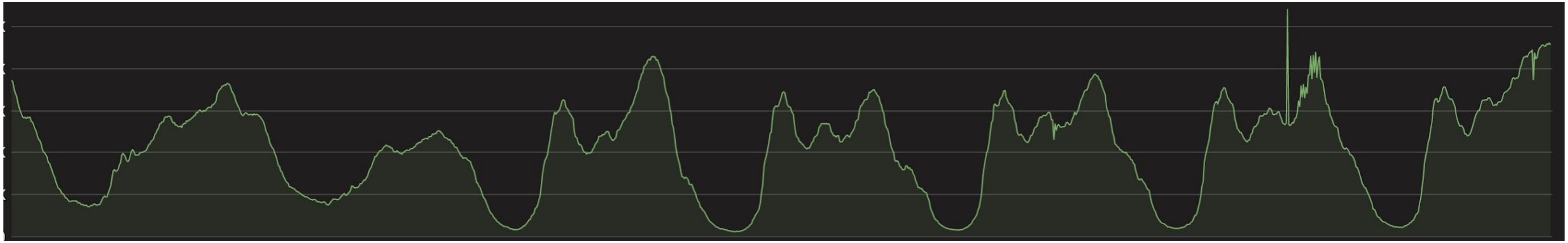
Use Cases for Forecasting at Uber

- Time series are ubiquitous at Uber

- Multiple horizons of interest
  - **Short term**: pricing and reliability
  - **Medium term**: insure markets are well-balanced
  - **Long term**: support Uber's growth without overspending

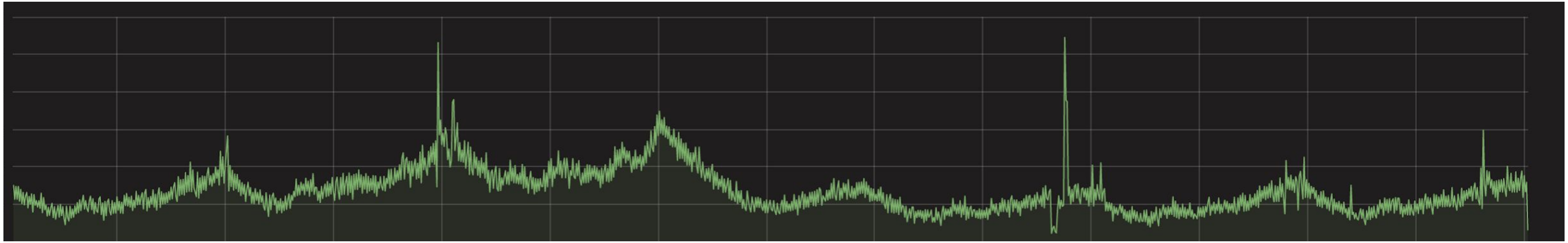
Compounded Challenges

- High cardinality
- Complex and variable patterns
- Shocks
- Sparse and short time series



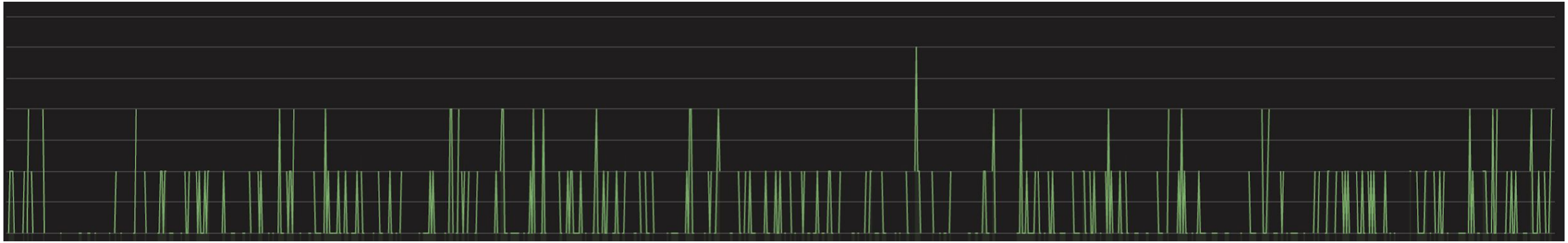
Compounded Challenges

- High cardinality
- Complex and variable patterns
- External shocks
- Sparse and short time series



Compounded Challenges

- High cardinality
- Complex and variable patterns
- External shocks
- Sparse and short time series



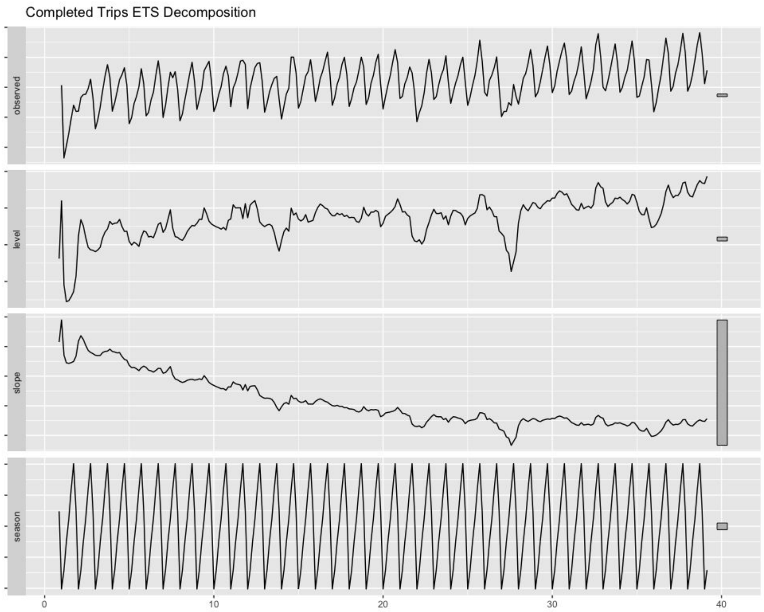
Classical Methods vs. Machine Learning

Classical Methods

- The state space paradigm
  - Decompose the time series into unobservable states
    - Structure equation encoding our beliefs
      - (typically with smoothing)
    - e.g. level, trend, seasonalities
  - Estimate each
  - Recombine and forecast one step at the time

Classical Methods

- The state space paradigm



Classical Methods

- The state space paradigm
- Models in this family are statistical
  - The states can be modeled as stochastic processes
  - Hence you can forecast prediction intervals (a.k.a. density forecasts)
  - A must for informed decision making

Classical Methods

- The state space paradigm
- Models in this family are statistical
- A very successful program!
  - Classical methods are hard to beat when you have
    - Sufficient history
    - Little metainformation
    - Few related time series

A Role for Machine Learning in Forecasting

- But at least one of the conditions above is often violated at Uber
  - Not enough history
    - Short time series in new markets
    - Sparse time series in some markets and for some technical metrics
  - Relevant meta-information
    - Holidays, weather, events
    - Incentives, technological changes
  - Many related time series
    - Uber time series often move with the pulse of the markets

A Survey of ML Techniques

A Survey Based on Personal Experience

- Generalized Linear Model (GLM)
  - The importance of featurization
- Quantile Random Forest (QRF)
  - Algorithms of greater power that learn nonlinearities

A Survey Based on Personal Experience

- Neural Networks (NN)
  - Unexpected challenges
- Hybrid Approaches
  - Best in class

Generalized Linear Model (GLM)

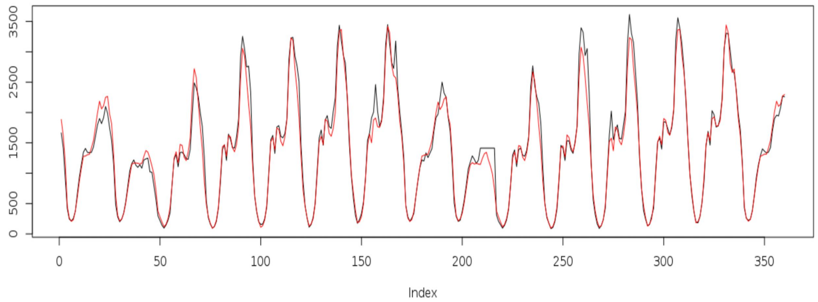
- A linking function to constrain the nonlinear structure

$$y = \lambda^{-1}(w \cdot x)$$

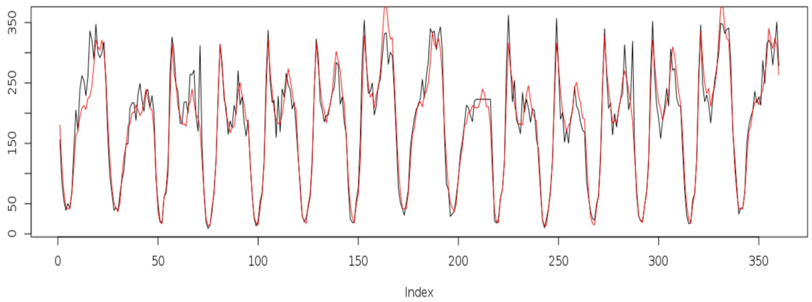
- Stochastic assumptions on the errors
    - Gaussian or Poisson error structure is common
    - Allows density forecasts and optimization via Maximum Likelihood (MLE)

GLM: Application to Incentive Allocations

- Forecasting in the medium term is key to allocate incentives to make our markets healthy and improve customer experience



Forecast (red) vs. Actual (black) for a large cluster



Forecast (red) vs. Actual (black) for a small cluster

GLM: The Importance of Featurization

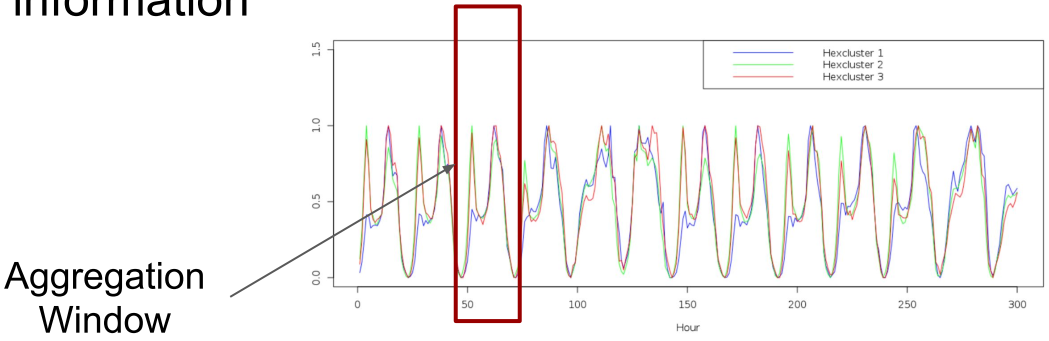
- This model learns across time series in a given city
- Some of the features convey past observed values of each series …
- … but geotemporal information is also a strong predictor
    - Neighborhood
    - Hour of day, day of week, etc.

GLM: The Importance of Featurization

- From one-hot encoding for geotemporal features

0	0	1	0	0	0	0	1	1	• • •	0	0	0	0	1	0	0	0	0

- To “fingerprinting”
  - i.e. robust aggregations of observed values by geotemporal units
  - Reduces the complexity of the model, while preserving all the relevant information



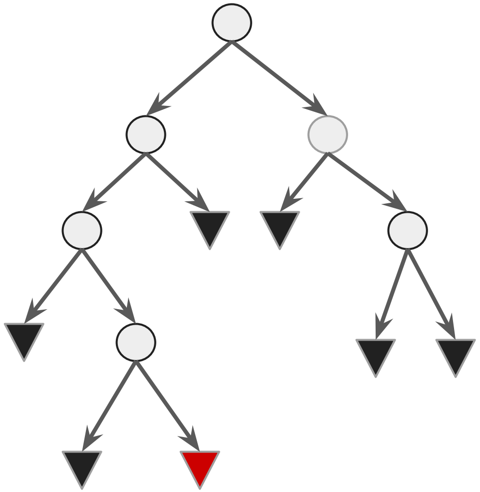
GLM: Aggregate Performance

- Incremental improvement over traditional methods
  - wMAPE: weighted Mean Absolute Percent Error is a common error metric

Error Metric	Baseline	GLM
wMAPE	-	↓ 12.5%

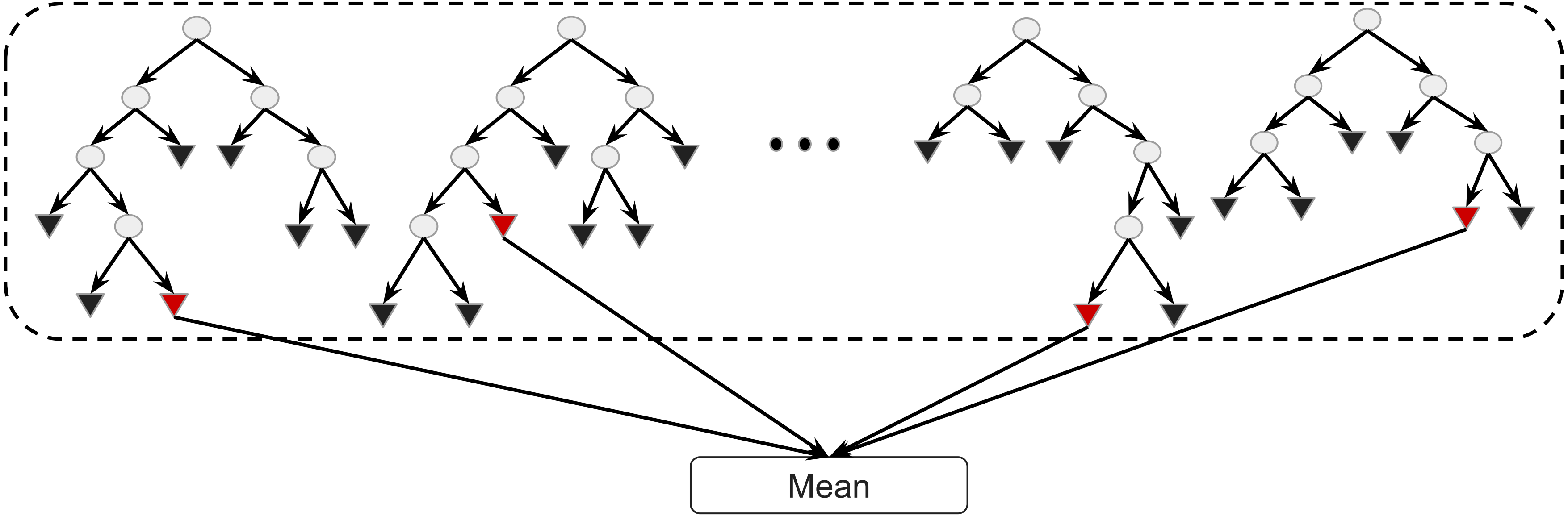
Quantile Random Forest (QRF)

- Decision Trees
  - Greedy optimizer for Purity/Variance



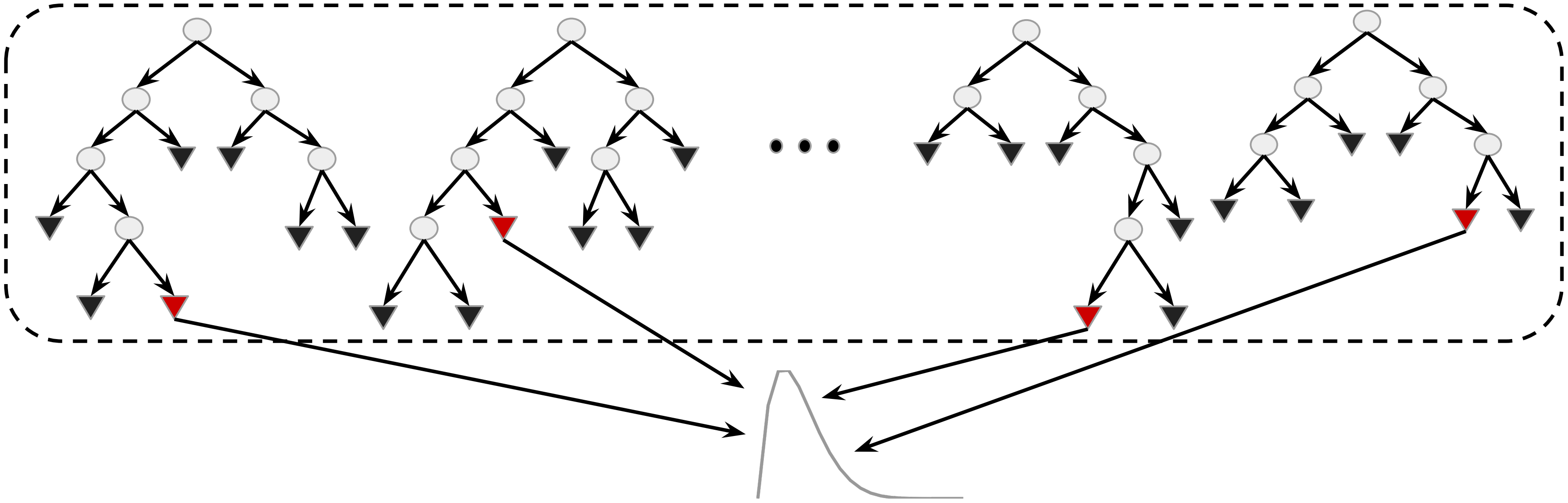
Quantile Random Forest (QRF)

- Add randomness and ensemble them in a forest
  - Less variance



Quantile Random Forest (QRF)

- From point to density forecast
  - Essential for informed decision-making



QRF: some remarks

- Nonlinearities can be learned (ML proper)

- Featurization is again paramount and it is hard work…
  - No sense of time in the algorithm itself
  - Yet easier than for Neural networks
  - Forgiving to categorical and feature with broadly different codomains

QRF: Aggregate performance

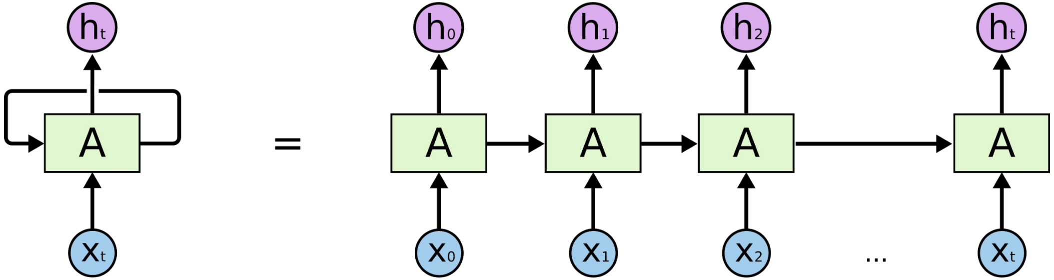
- Proved remarkably effective!

Error Metric	Baseline	GLM	QRF
wMAPE	-	↓ 12.5%	↓ 22%

Neural Networks (NN)

- Architecture of interconnected neurons with nonlinear activation functions and weights that can be learned

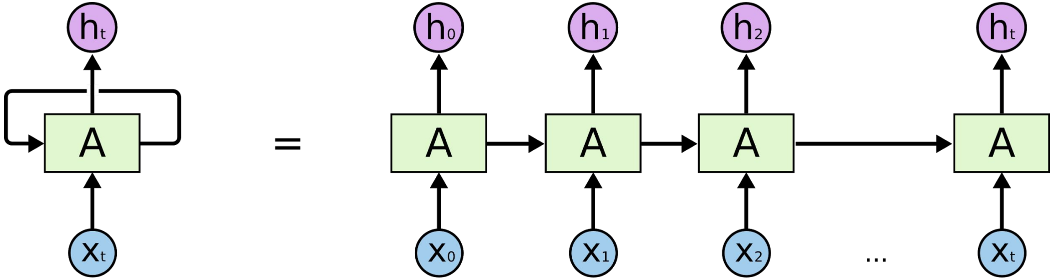
- A variant, Recurrent Neural Networks (RNN), has a sense of time!
  - An internal mutable state that allows for memory



http://colah.github.io/posts/2015-08-Understanding-LSTMs/

Long Short-Term Memory (LSTM)

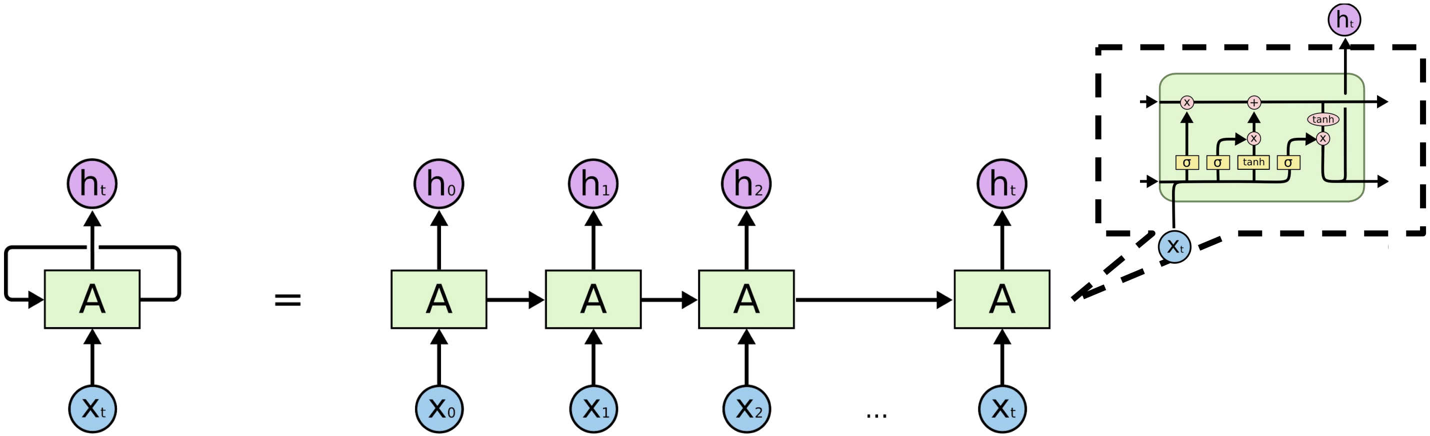
- Long-, Short-Term Memory cells are often used to build RNNs
  - Retain variable amount of past information
  - Fight off vanishing gradients



http://colah.github.io/posts/2015-08-Understanding-LSTMs/

Long Short-Term Memory (LSTM)

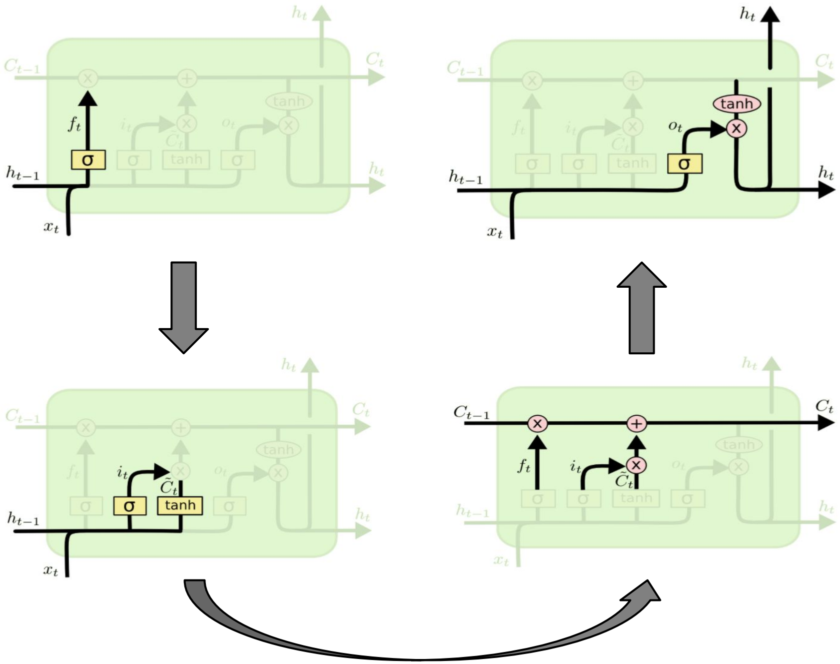
- LSTM Cells have an internal structure



http://colah.github.io/posts/2015-08-Understanding-LSTMs/

Long Short-Term Memory (LSTM)

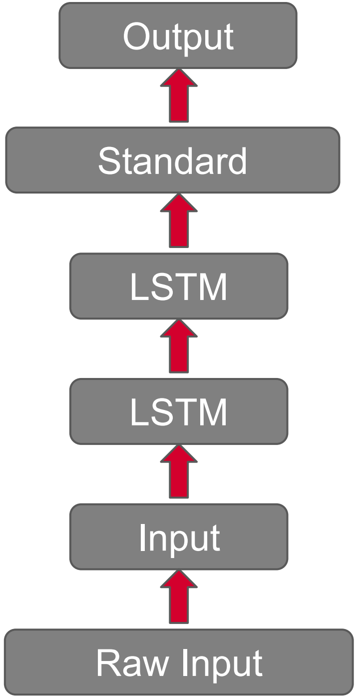
- An LSTM cell has internal structure
  - Three gates
    - Forget
    - Input
    - Output
  - Allows for selective and adaptive memory



http://colah.github.io/posts/2015-08-Understanding-LSTMs/

NN: Typical Architectures

- Not very deep architecture
  - If compared with computer vision
  - Many variations for different use cases
- Preprocessing is required and a challenge
  - Normalizing time series and features
- Used for
  - Incentive allocations
  - Anomaly detection
  - As part of hybrid models



NN: Aggregate Performance

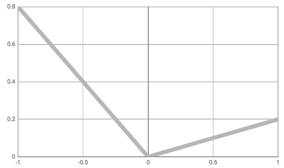
- Robust performance, but not an improvement over QRF

- Is this the final word?

Error Metric	Baseline	GLM	QRF	MLP	LSTM
wMAPE	-	↓ 12.5%	↓ 22%	↓ 12.5%	↓ 18.8%

NN: Recovering Density Forecasts

- Optimizing with a pin-ball loss estimates quantiles



- Monte Carlo drop-out method
  - [Engineering Uncertainty Estimation in Neural Networks for Time Series Prediction at Uber]() -- Uber Eng Blog

Hybrid Models

Hybrid Models

- They combine
  - Classical structural models (e.g. state space models)
  - Machine Learning models (e.g. Deep learning)
- They are hierarchical
  - Some of the parameters are local, i.e. fit per time series…
  - … others are global, i.e. learned across time series
  - The local parameters provide the specificity NNs often lack

An Event Lift Model

- Uber time series are significantly impacted by external events
  - Holidays
  - Extreme weather
- Classical models do well on ordinary days but have trouble learning from rare events

An Event Lift Model

- A hybrid solution
- A Generalized Additive Model (GAM) for the baseline forecast
  - Six months out
  - Detrending with exponential smoothing
  - Capturing seasonality with semi-parametric splines
- Quantile Random Forest to learn multiplicative event lifts across time series

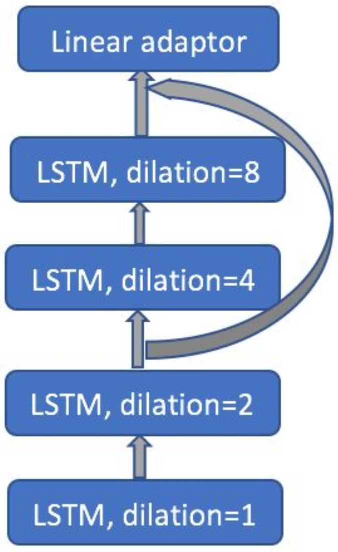
An Event Lift Model

- Improvement of six months ahead forecast for number of trips

Error Metric	Region	HW	GAM	GAM + QRF
wMAPE	US	-	↓ 8.1%	↓ **13.5%**
wMAPE	Latin America	-	↓ 32.2%	↓ **40.4%**

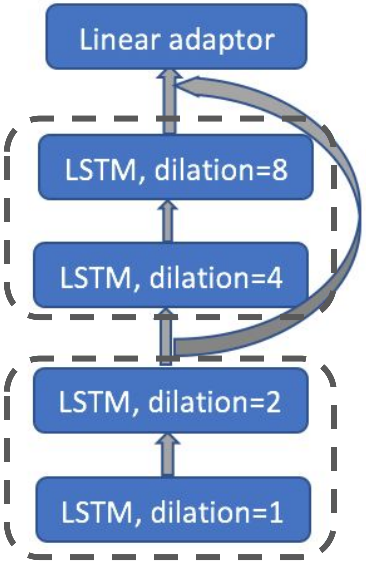
The M4 model

- The M4 model: an ES-RNN hybrid



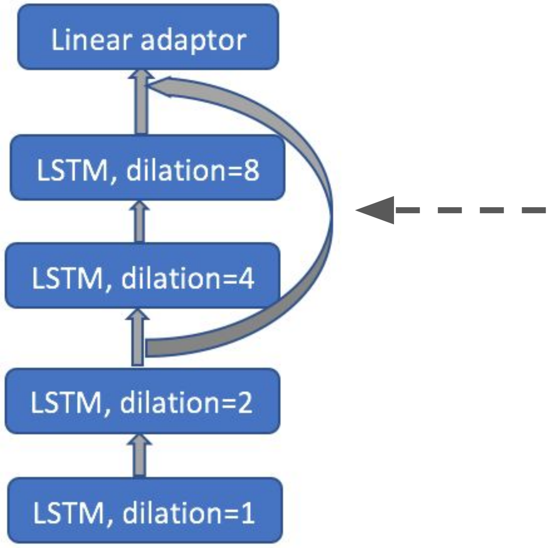
The M4 model

- The M4 model: an ES-RNN hybrid



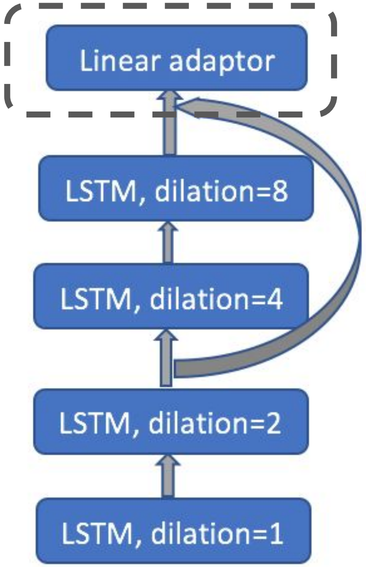
The M4 model

- The M4 model: an ES-RNN hybrid



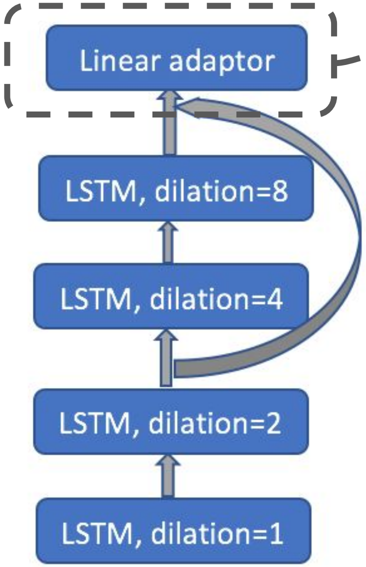
The M4 model

- The M4 model: an ES-RNN hybrid



The M4 model

- The M4 model: an ES-RNN hybrid



$$\boldsymbol{RNN_W \times Level_{\alpha} \times Seasonality_{\gamma}}$$

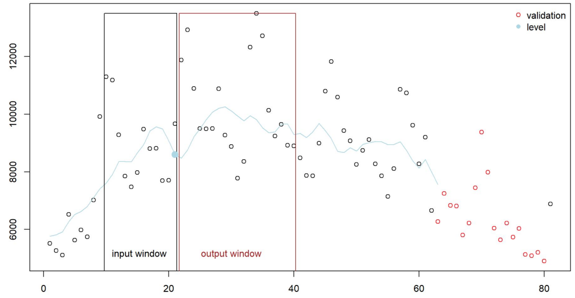
- Exponential smoothed structural components
- No need to capture local trend

Dynamic Computational Graphs

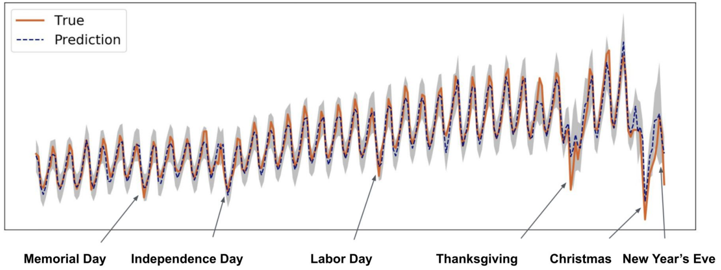
- Two types of parameters
  - The NNs weights ***W*** are learned across time series (robustness)
  - The smoothing coefficients ***α*** and ***γ*** time series by time series (specificity)

- The computational graph will vary from time series to time series
  - This mode is supported by several frameworks: Dynet, PyTorch and more recently TensorFlow

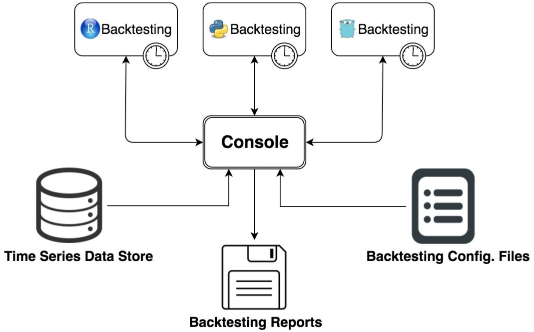
Learn more about Forecasting at Uber



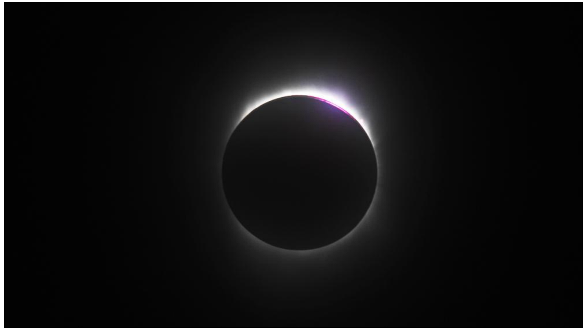
M4 Forecasting Competition: Introducing a New Hybrid ES-RNN Model



Engineering Uncertainty Estimation in Neural Networks for Time Series Prediction at Uber



Omphalos, Uber's Parallel and Language-Extensible Time Series Backtesting Tool



Forecasting at Uber: An Introduction

Thank you!