Forecasting at Amazon

Problems, Methods and Systems

Tim Januschowski– tjnsch@amazon.com

Amazon Development Center Germany, Berlin

37th International Symposium on Forecasting, Cairns, Australia

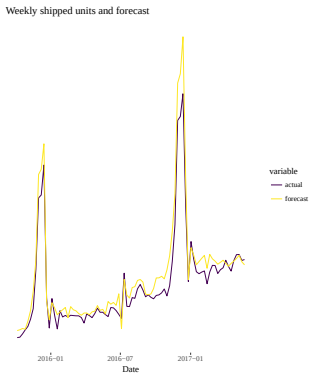
My Perspective

- Machine Learning Perspective on Forecasting: accuracy and automation
- Not: Given specific problem and set of existing methods, how do we solve the problem best?
- But: Given a number of different forecasting problems, which methods should we have to address them?

Outline

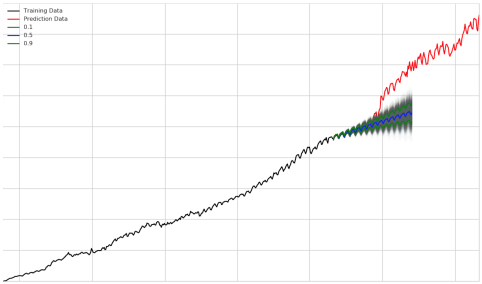
- Examples of Forecasting Problems at Amazon
- Forecasting Methods & Systems developed in Berlin

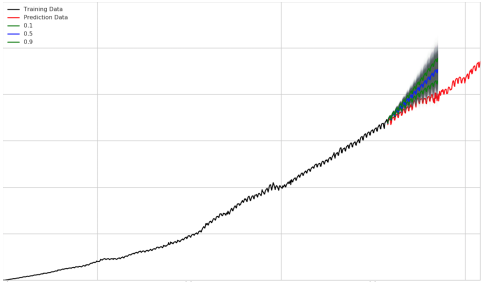
Forecasting Problems at Amazon I: Retail Demand



- Problem: predict overall Amazon retail demand years into the future.
- Decision Problems: topology planning, market entry/segment analyses

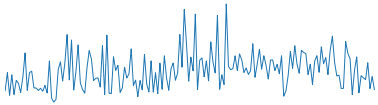
Forecasting Problems at Amazon II: AWS Compute Capacity

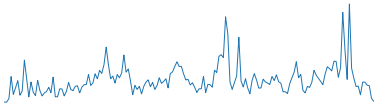




- Problem: predict AWS compute capacity demand
- Decision Problem: how many servers to order when and where

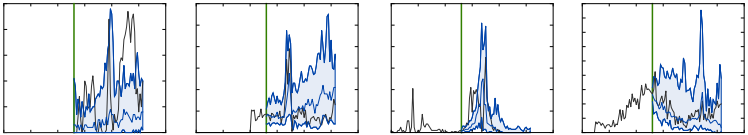
Forecasting Problems at Amazon III: Staff Planning





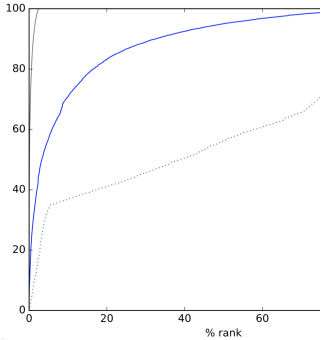
- Problem: predict attendance rate of Distribution Centre staff
- Decision Problems: how to schedule staff and when to hire how much staff

Forecasting Problems at Amazon IV: Retail Product Forecasting



- Problem: predict the demand for each product available on Amazon websites
- Decision Problems: how many units to order when, when to mark products down

Forecasting Problems at Amazon V: Product Translations

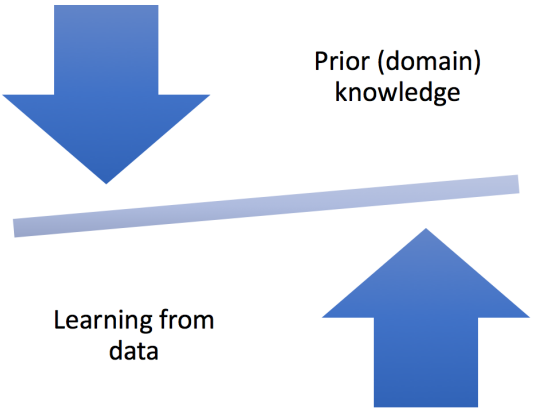


- Problem: predict which products are going to sell in another country (cold start)
- Decision Problem: which products should be offered in other countries

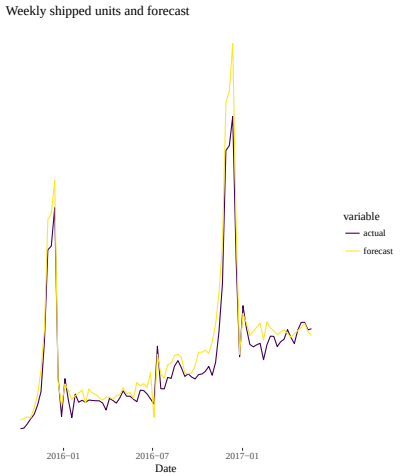
Taxonomy of Forecasting Problems: Dimensions

- number of time series/ratio of scientists per time series
- training of scientists: econometrics, statistics, machine learning, computer science
- forecast horizons: years to days
- time granularities: years, months, weeks, days
- aggregation granularity (for hierarchically organized time series)
- latency of forecast production/forecast computation frequency
- consumer of forecast/degree of automation/human interaction with forecast
- characteristics of time series
- *forecasting methods*: white vs black box (impose structure, parameter sharing, transparency)

Taxonomy of Forecasting Problems: Forecasting Methods



Taxonomy of Forecasting Problems: Strategic Forecasting



- Example: Overall demand for retail products on Amazon
- lots of econometricians for few time series
- forecast horizon: years, time granularity: weekly at most
- runs irregularly or a few times per months
- high degree of interaction with forecast
- models which estimate uncertainty correctly, allow to enforce structure, allow for careful modelling of effects
- high counts, relatively smooth, trend breaks possible, long history (in most cases)

Taxonomy of Forecasting Problems: Operational Forecasting

- Example: Demand forecast for retail products
- millions of time series per scientists (machine learning & software development engineers)
- forecast horizon: days, weeks, at most months
- runs at least daily/on-demand
- hands-off approach
- models can be more black box as long as they are robust
- low counts, bursty, short history and life cycles, intermittent

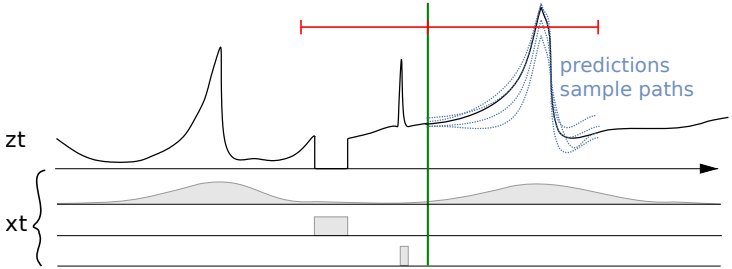
Taxonomy of Forecasting Problems: Tactical Forecasting

- Example: Ordering of compute racks for AWS
- 100s-1000s of time series per scientist
- varying roles from business analysts to machine learning to econometrics
- forecast horizon: months, granularity: weekly at most
- runs irregularly or at most every week
- limited degree of interaction with forecast, but some constraints on stability of forecast over time and automated output checking
- models estimate uncertainty correctly, some transfer of information across time series necessary
- high counts, relatively smooth, trend breaks possible, short history & life cycles possible, burstiness

Focus of this talk: Automatable Forecasts

- what forecasting methods do we need?
- what software/systems do we need?

Forecasting Methods: General Setup



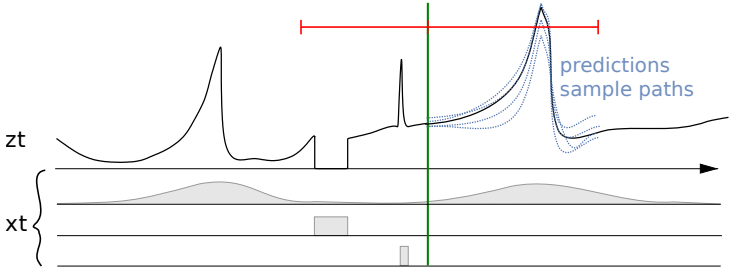
- Predict the future behavior of a time series given its past

$$\ldots, z_{t_0-3}, z_{t_0-2}, z_{t_0-1} \Longrightarrow P(z_{t_0}, z_{t_0+1}, \ldots z_T)$$

- Make optimal decisions

$$\text{best action} = \underset{a}{\operatorname{argmin}}\, \mathrm{E}_{\mathrm{P}}[\text{cost}(a, z_{t_0}, z_{t_0+1}, \ldots z_T)]$$

Forecasting Methods: General Setup



- Predict the future behavior of a time series given its past

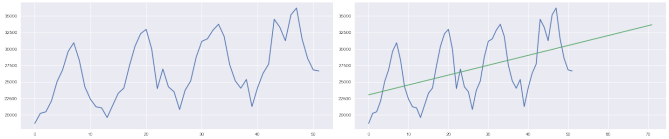
$$\ldots, z_{t_0-3}, z_{t_0-2}, z_{t_0-1} \Longrightarrow P(z_{t_0}, z_{t_0+1}, \ldots z_T)$$

- Make optimal decisions

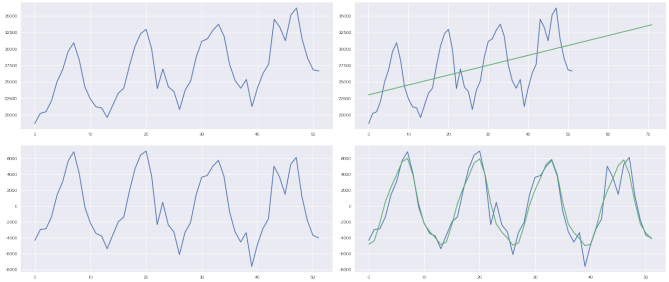
$$\text{best action} = \underset{a}{\operatorname{argmin}}\, \mathrm{E}_{\mathrm{P}}[\text{cost}(a, z_{t_0}, z_{t_0+1}, \ldots z_T)]$$

Forecasting Methods: The Classical Approach

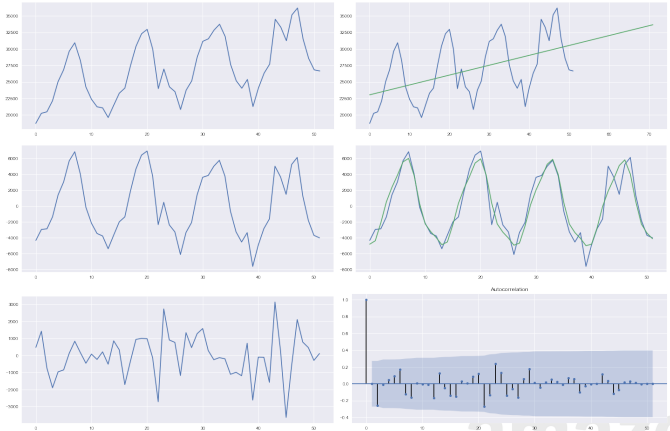
Forecasting Methods: The Classical Approach



Forecasting Methods: The Classical Approach



Forecasting Methods: The Classical Approach



The Classical Approach(es): Box-Jenkins, State-space Models, ...

Image (c) User:Colin / Wikimedia Commons / CC BY-SA 3.0

PROS

- De-facto standard; widely used
- Decomposition $\rightarrow$ decoupling
- White box: explicitly model-based
- Embarassingly parallel

CONS

- Requires lots manual work by experts
- joint-learning of parameters across decomposition layers is involved
- Cannot learn patterns across time series
- Cannot handle cold-starts
- Model-based: all effects need to be explicitly modelled

$\Rightarrow$ Suitable for strategical forecasts & in situations when there is a lot of history available per time series

The Classical Approach(es): Box-Jenkins, State-space Models, ...

Image (c) User:Colin / Wikimedia Commons / CC BY-SA 3.0

PROS

- De-facto standard; widely used
- Decomposition $\rightarrow$ decoupling
- White box: explicitly model-based
- Embarassingly parallel

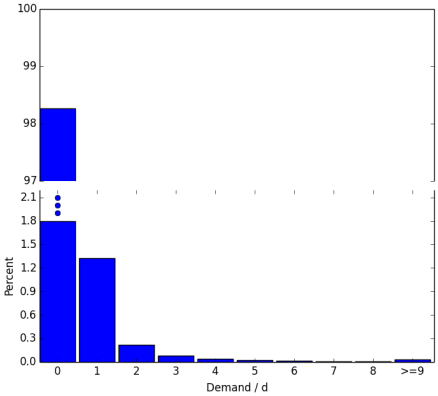
CONS

- Requires lots manual work by experts
- joint-learning of parameters across decomposition layers is involved
- Cannot learn patterns across time series
- Cannot handle cold-starts
- Model-based: all effects need to be explicitly modelled

$\Rightarrow$ Suitable for strategical forecasts & in situations when there is a lot of history available per time series

Local Methods

Amazon data typical violates Gaussianity assumptions:



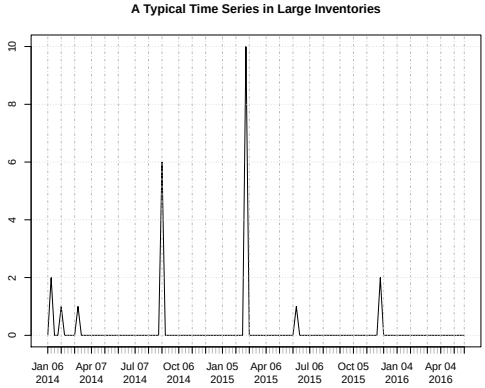


Figure: Left: Marginal histogram of demand values $\{z_{it}\}$ for a typical dataset. Right: A typical time series in large inventories.

Extensions to ETS

Incorporate non-Gaussian likelihoods, e.g. for $k \geq 2$

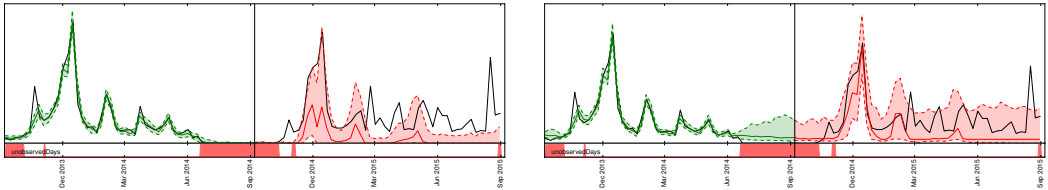
$$P(z|\{y^{(k)}\}) = P_{\text{poi}}(z-2|y^{(2)})^{\mathrm{I}_{\{z\geq 2\}}} \prod_{k=0}^{1} \sigma(\tilde{z}_k y^{(k)})^{\mathrm{I}_{\{z\geq k\}}}, \quad \tilde{z}_k := \mathrm{I}_{\{z=k\}} - \mathrm{I}_{\{z>k\}}.$$

with latent process:

$$y_t = \boldsymbol{a}_t^\top \boldsymbol{l}_{t-1} + b_t, \quad b_t = \boldsymbol{w}^\top \boldsymbol{x}_t, \quad \boldsymbol{l}_t = \boldsymbol{F}\boldsymbol{l}_{t-1} + \boldsymbol{g}_t \varepsilon_t, \quad \varepsilon_t \sim N(0,1).$$

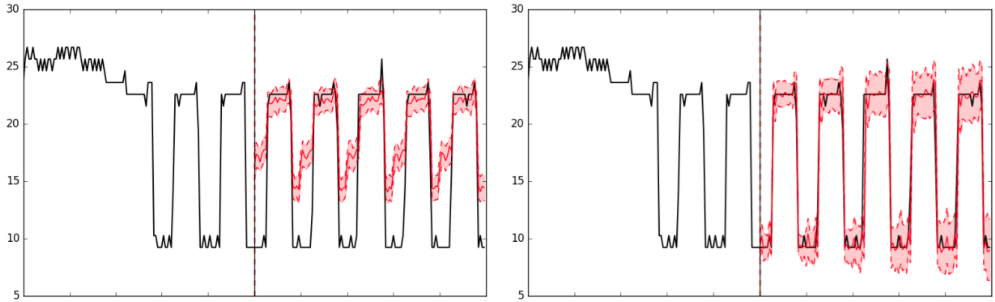
- b_t is the GLM deterministic linear function, $\boldsymbol{l}_t$ is a *latent state*.
- a *single* Gaussian innovation variable ε_t per time step.
- This *innovation state space model* (ISSM) is defined by $\boldsymbol{a}_t$, $\boldsymbol{g}_t$ and $\boldsymbol{F}$, as well as the prior $\boldsymbol{l}_0 \sim P(\boldsymbol{l}_0)$.
- Innovation vector $\boldsymbol{g}_t$ comes in terms of parameters to be learned (the innovation strengths), while $\boldsymbol{F}$ and $\boldsymbol{a}_t$ are fixed
- initial state $\boldsymbol{l}_0$ has to be specified (via a Gaussian prior distribution $P(\boldsymbol{l}_0)$, whose parameters (means, standard deviation) are learned from data)

Out of Stock Handling



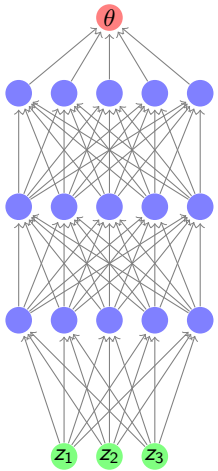
Demand forecast for an item which is partially out of stock. Each panel: Training range left (green), prediction range right (red), true targets black. In color: Median, P10 to P90. Bottom: Out of stock ($\geq$ 80% of day) marked in red. **Left**: Out of stock signal ignored. Demand forecast drops to zero, strong underbias in prediction range. **Right**: Out of stock regions treated as missing observations. Demand becomes uncertain in out of stock region. No underbias in prediction range.

Modeling Seasonality Patterns



Modeling hourly seasonality via features (left panel) versus latent state (right panel).

Local vs. Global Models



- **Deep Learning**: Compose complex (differentiable) black-box functions from simpler building blocks and learn them end-to-end.
- Recall our goal:

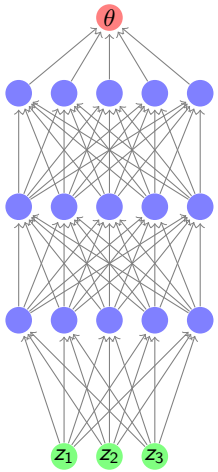
$$\ldots, z_{t_0-3}, z_{t_0-2}, z_{t_0-1} \Longrightarrow P_\theta(z_{t_0}, z_{t_0+1}, \ldots z_T)$$

- Directly parameterize P_θ with a black-box function $f(\cdot)$, learned using a deep network:

$$\theta = f(z_0, z_1, \ldots, z_{t_0-1})$$

- For example,

$$P_\theta(z_{t_0}, z_{t_0+1}, \ldots z_T) = \prod_{i=0}^{T-t_0-1} \mathcal{N}(z_{t_0+i} | \mu_i(\theta), \sigma_i(\theta))$$



- **Deep Learning**: Compose complex (differentiable) black-box functions from simpler building blocks and learn them end-to-end.
- Recall our goal:

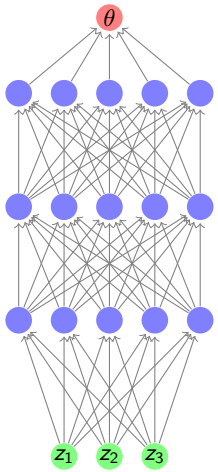
$$\ldots, z_{t_0-3}, z_{t_0-2}, z_{t_0-1} \Longrightarrow P_\theta(z_{t_0}, z_{t_0+1}, \ldots z_T)$$

- Directly parameterize P_θ with a black-box function $f(\cdot)$, learned using a deep network:

$$\theta = f(z_0, z_1, \ldots, z_{t_0-1})$$

- For example,

$$P_\theta(z_{t_0}, z_{t_0+1}, \ldots z_T) = \prod_{i=0}^{T-t_0-1} \mathcal{N}(z_{t_0+i} | \mu_i(\theta), \sigma_i(\theta))$$



- **Deep Learning**: Compose complex (differentiable) black-box functions from simpler building blocks and learn them end-to-end.
- Recall our goal:

$$\ldots, z_{t_0-3}, z_{t_0-2}, z_{t_0-1} \Longrightarrow P_\theta(z_{t_0}, z_{t_0+1}, \ldots z_T)$$

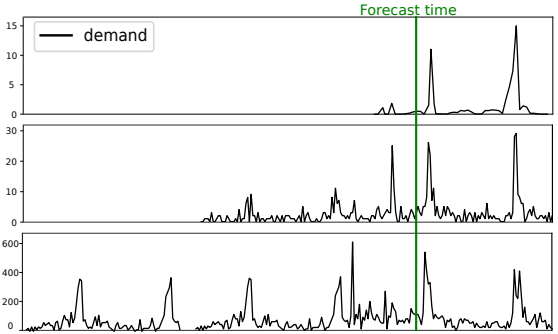
- Directly parameterize P_θ with a black-box function $f(\cdot)$, learned using a deep network:

$$\theta = f(z_0, z_1, \ldots, z_{t_0-1})$$

- For example,

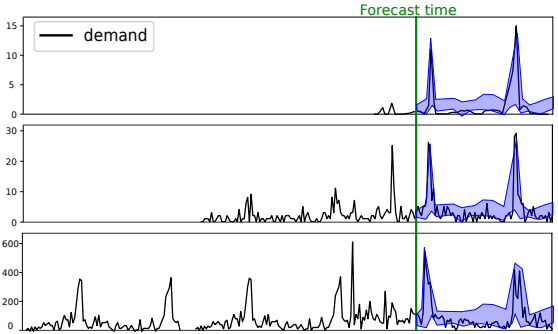
$$P_\theta(z_{t_0}, z_{t_0+1}, \ldots z_T) = \prod_{i=0}^{T-t_0-1} \mathcal{N}(z_{t_0+i} | \mu_i(\boldsymbol{\theta}), \sigma_i(\boldsymbol{\theta}))$$

Deep Learning for Forecasting



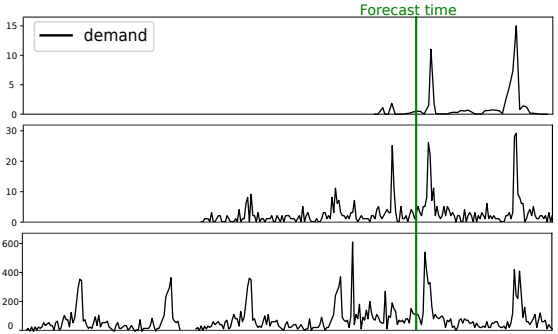
- Input:
  - time series past values $z_1, z_2, \ldots, z_{t_0-1}$
  - possibly some features $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_T$
- Output: (estimated) joint distribution $P_\theta(z_{t_0}, z_{t_0+1}, \ldots, z_T)$
- Trained by maximizing likelihood on past windows

Deep Learning for Forecasting



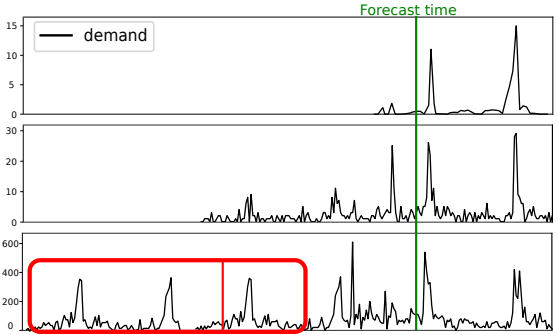
- Input:
  - time series past values $z_1, z_2, \ldots, z_{t_0-1}$
  - possibly some features $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_T$
- Output: (estimated) joint distribution $P_\theta(z_{t_0}, z_{t_0+1}, \ldots, z_T)$
- Trained by maximizing likelihood on past windows

Deep Learning for Forecasting



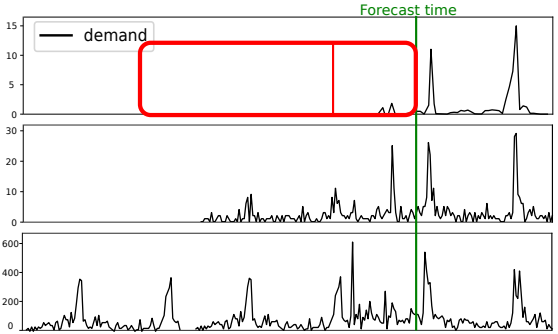
- Input:
  - time series past values $z_1, z_2, \ldots, z_{t_0-1}$
  - possibly some features $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_T$
- Output: (estimated) joint distribution $P_\theta(z_{t_0}, z_{t_0+1}, \ldots, z_T)$
- Trained by maximizing likelihood on past windows

Deep Learning for Forecasting



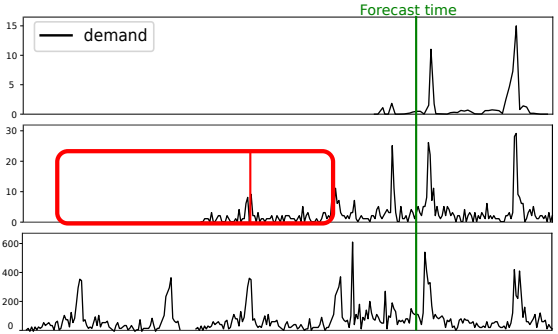
- Input:
  - time series past values $z_1, z_2, \ldots, z_{t_0-1}$
  - possibly some features $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_T$
- Output: (estimated) joint distribution $P_\theta(z_{t_0}, z_{t_0+1}, \ldots, z_T)$
- Trained by maximizing likelihood on past windows

Deep Learning for Forecasting



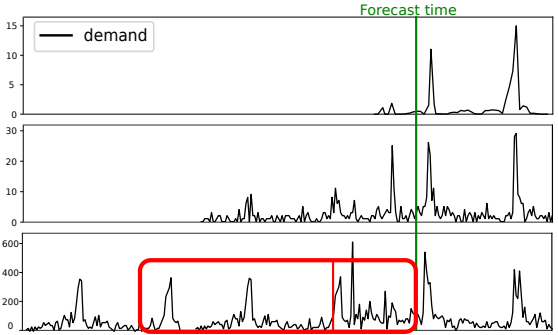
- Input:
  - time series past values $z_1, z_2, \ldots, z_{t_0-1}$
  - possibly some features $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_T$
- Output: (estimated) joint distribution $P_\theta(z_{t_0}, z_{t_0+1}, \ldots, z_T)$
- Trained by maximizing likelihood on past windows

Deep Learning for Forecasting



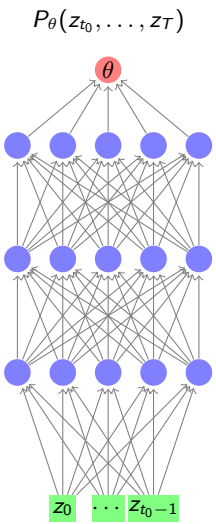
- Input:
  - time series past values $z_1, z_2, \ldots, z_{t_0-1}$
  - possibly some features $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_T$
- Output: (estimated) joint distribution $P_\theta(z_{t_0}, z_{t_0+1}, \ldots, z_T)$
- Trained by maximizing likelihood on past windows

Deep Learning for Forecasting



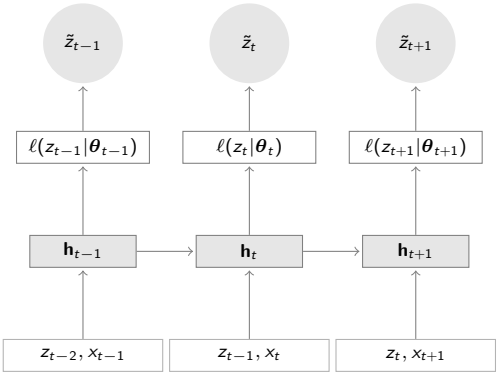
- Input:
  - time series past values $z_1, z_2, \ldots, z_{t_0-1}$
  - possibly some features $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_T$
- Output: (estimated) joint distribution $P_\theta(z_{t_0}, z_{t_0+1}, \ldots, z_T)$
- Trained by maximizing likelihood on past windows

From Feed-Forward to Recurrent Networks



Feed-forward models can already work well, but have a number of disadvantages:

- Outputs not correlated across time
- Models tend to be larger than comparable recurrent ones; need more data to train
- Fixed-length conditioning and prediction ranges

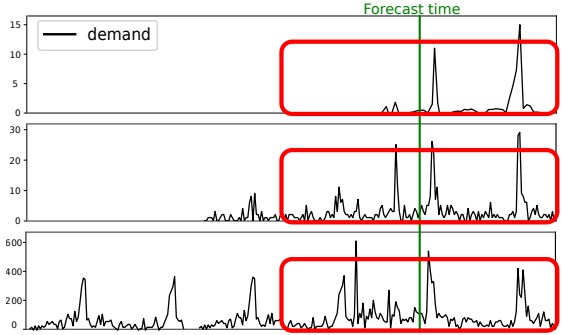


Autoregressive Recurrent Networks

$$z_t \sim \ell(z_t|\boldsymbol{\theta}_t)$$
$$\boldsymbol{\theta}_t = \theta(\mathbf{h}_t)$$
$$\mathbf{h}_t = \psi\left(\mathbf{h}_{t-1}, z_{t-1}, \mathbf{x}_t\right)$$

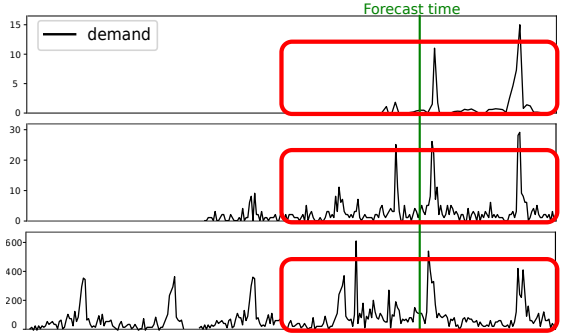
- The recurrent network $\psi(\cdot)$ can be implemented e.g. using stacks of LSTM cells.

DeepAR - Prediction



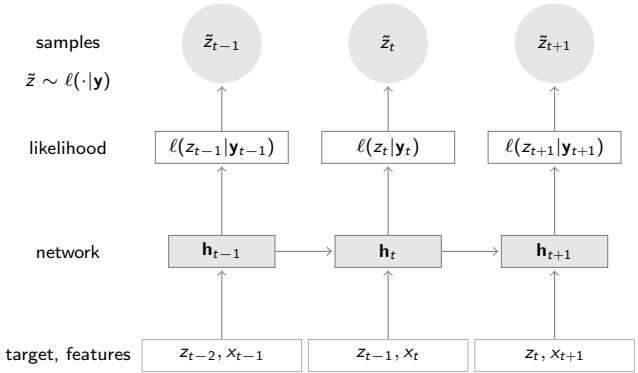
- target z_t is unobserved after the forecast time

DeepAR - Prediction



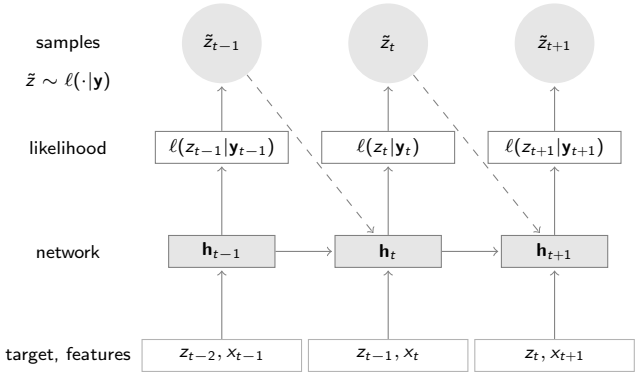
- target z_t is unobserved after the forecast time

DeepAR - prediction



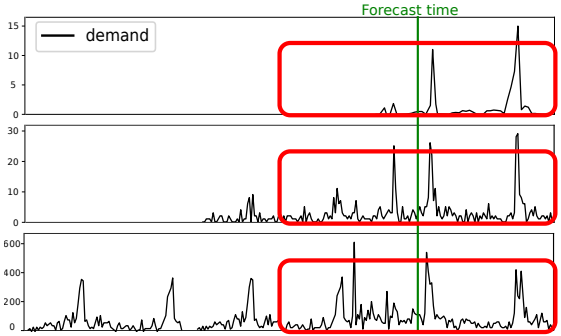
- z_t target, $\mathbf{x_t}$ features, h_t stack of several LSTM
- $\ell(z_t|\mathbf{y}_t)$ likelihood: Gaussian, negative binomial

DeepAR - prediction



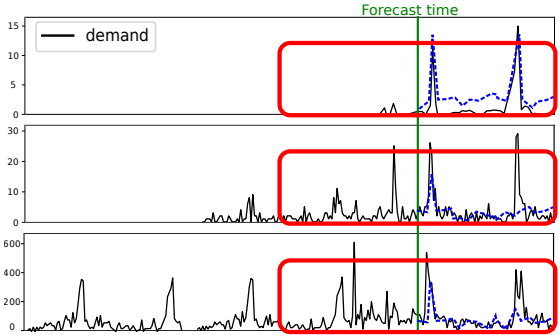
- z_t target, $\mathbf{x_t}$ features, h_t stack of several LSTM
- $\ell(z_t|\mathbf{y}_t)$ likelihood: Gaussian, negative binomial
- prediction: use sample $\tilde{z} \sim \ell(\cdot|\mathbf{y})$ instead of true target for unknown (future) values

DeepAR - Probabilistic Prediction



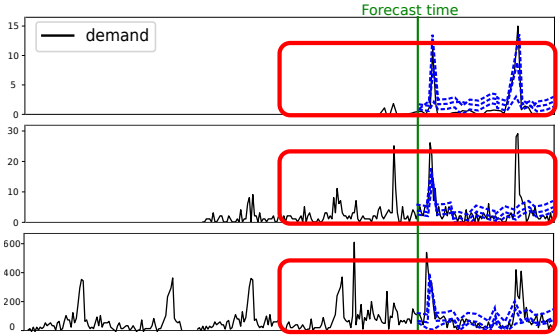
- now we have a generative model!
- the joint distribution is represented with *sample paths*
- one can calculate confidence intervals, marginal distributions, ...

DeepAR - Probabilistic Prediction



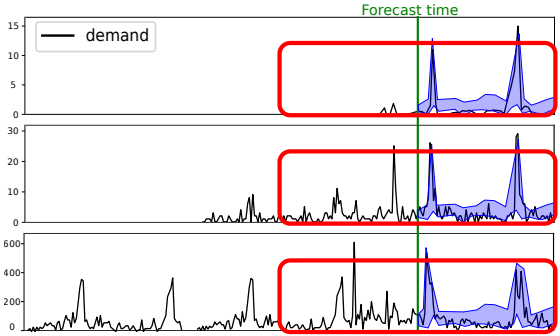
- now we have a generative model!
- the joint distribution is represented with *sample paths*
- one can calculate confidence intervals, marginal distributions, ...

DeepAR - Probabilistic Prediction



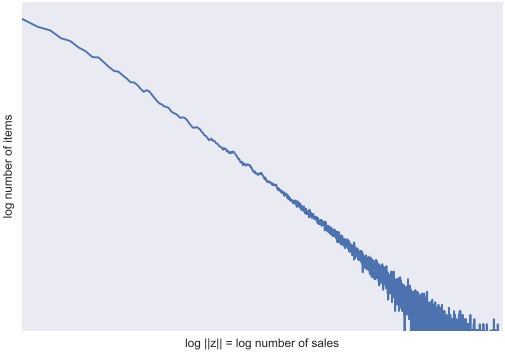
- now we have a generative model!
- the joint distribution is represented with *sample paths*
- one can calculate confidence intervals, marginal distributions, ...

DeepAR - Probabilistic Prediction



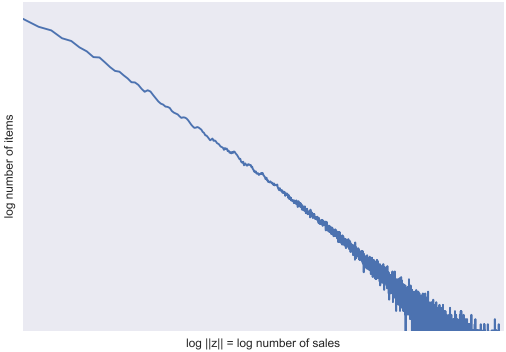
- now we have a generative model!
- the joint distribution is represented with *sample paths*
- one can calculate confidence intervals, marginal distributions, ...

Input scaling - what could go wrong



- Learning patterns across time series is difficult if their amplitudes differ
- In practice, mean(z_t) often follows a power-law
- Scale-free $\Rightarrow$ no good bucket separation!
- Occurs frequently in real data due to rich-get-richer phenomena

Handling different Amplitudes



- Scale the input and output of the neural network by item amplitude
- Weighted sampling to counter-balance power-law behaviour
- Crucial to get good accuracy across all amplitudes

Some Empirical Results

- Dataset: 500K weekly time series of sales of US Softlines products
- Baseline: Innovation State Space Model (Seeger et al., NIPS16)
- On average **15%** improvement for P50QL/P90QL (over 8 different time intervals)
- $<$ 10 features; little hyper-parameter tuning
- Training/predicting/evaluating 500K time-series takes less than 10 hours on a single AWS p2.xlarge instance (1 GPU & 4 CPU)

Some Empirical Results

- Dataset: 500K weekly time series of sales of US Softlines products
- Baseline: Innovation State Space Model (Seeger et al., NIPS16)
- On average **15%** improvement for P50QL/P90QL (over 8 different time intervals)
- $<$ 10 features; little hyper-parameter tuning
- Training/predicting/evaluating 500K time-series takes less than 10 hours on a single AWS p2.xlarge instance (1 GPU & 4 CPU)

Some Empirical Results

- Dataset: 500K weekly time series of sales of US Softlines products
- Baseline: Innovation State Space Model (Seeger et al., NIPS16)
- On average **15%** improvement for P50QL/P90QL (over 8 different time intervals)
- $<$ 10 features; little hyper-parameter tuning
- Training/predicting/evaluating 500K time-series takes less than 10 hours on a single AWS p2.xlarge instance (1 GPU & 4 CPU)

Some Empirical Results

- Dataset: 500K weekly time series of sales of US Softlines products
- Baseline: Innovation State Space Model (Seeger et al., NIPS16)
- On average **15%** improvement for P50QL/P90QL (over 8 different time intervals)
- $<$ 10 features; little hyper-parameter tuning
- Training/predicting/evaluating 500K time-series takes less than 10 hours on a single AWS p2.xlarge instance (1 GPU & 4 CPU)

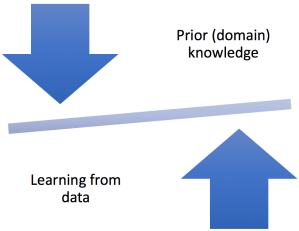
Some Empirical Results

- Dataset: 500K weekly time series of sales of US Softlines products
- Baseline: Innovation State Space Model (Seeger et al., NIPS16)
- On average **15%** improvement for P50QL/P90QL (over 8 different time intervals)
- $<$ 10 features; little hyper-parameter tuning
- Training/predicting/evaluating 500K time-series takes less than 10 hours on a single AWS p2.xlarge instance (1 GPU & 4 CPU)

Some Real-World Examples

Deep Learning in Forecasting: Conclusion and Outlook

- Deep Learning methodology applied to forecasting yields flexible, accurate, and scalable forecasting systems
- Models can learn complex temporal patterns across time series
- "Model-free" black-box approaches trained end-to-end can replace complex model-based forecasting systems
- Deep Learning tool boxes offer flexible and efficient ML programming environments (we use MXNet)
  - Example (i): use custom loss function
  - Example (ii): we can train on millions of time series

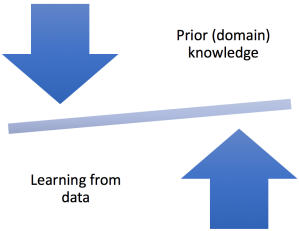


Future Work

- how to make our methods more data efficient?
- how to impose structure?
- how to explain results?

Deep Learning in Forecasting: Conclusion and Outlook

- Deep Learning methodology applied to forecasting yields flexible, accurate, and scalable forecasting systems
- Models can learn complex temporal patterns across time series
- "Model-free" black-box approaches trained end-to-end can replace complex model-based forecasting systems
- Deep Learning tool boxes offer flexible and efficient ML programming environments (we use MXNet)
  - Example (i): use custom loss function
  - Example (ii): we can train on millions of time series



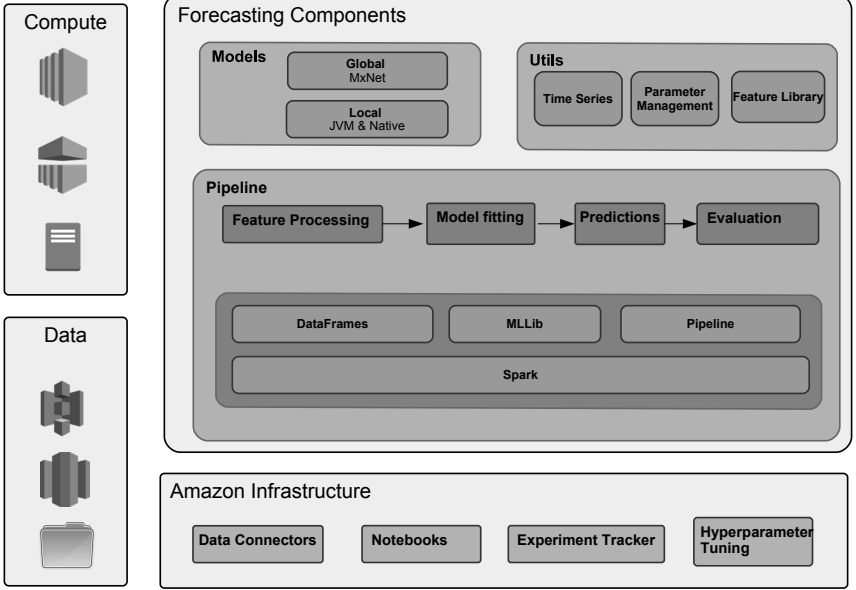
Future Work

- how to make our methods more data efficient?
- how to impose structure?
- how to explain results?

Forecasting Systems developed by Core ML Berlin: Constraints

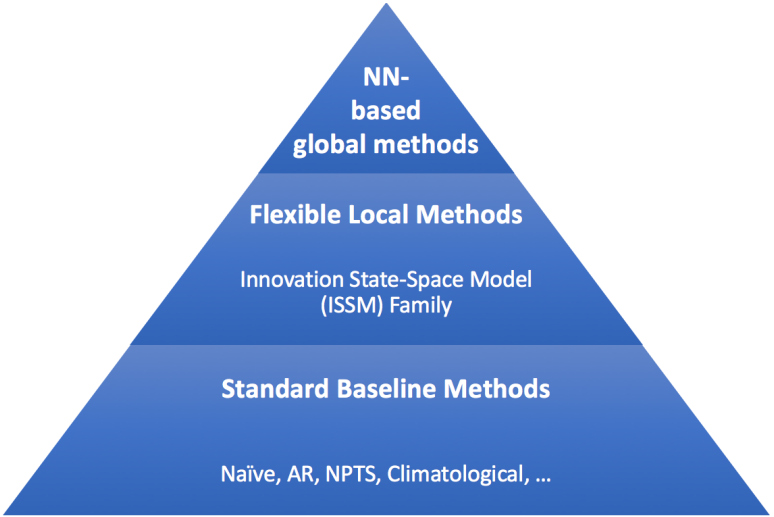
- Reproducible experiments
- Extensability
- Reusability
- Scalability
- Integration into Amazon infrastructure for data and Machine Learning infrastructure

Main components


Compute
Data
Forecasting Components
Models
Global
MxNet
Local
JVM & Native
Utils
Time Series
Parameter Management
Feature Library
Pipeline
Feature Processing
Model fitting
Predictions
Evaluation
DataFrames
MLLib
Pipeline
Spark
Amazon Infrastructure
Data Connectors
Notebooks
Experiment Tracker
Hyperparameter Tuning

```scala
val df: DataFrame = getData(spark)

val trainingRangeExtractor = new TimepointTrainingRangeExtractor()
  .setForecastStartTime(ZonedDateTime.parse("2014-01-01T00:00Z"))

val featureTransformer = new FlowFeatureTransformer()
  .setPredictLength(3)

val forecasters = new LocalForecasterTransformer()
  .setLocalForecaster(new NonSeasonalBaselineForecaster().setMethod("Drift"))

val evaluator = new ItemEvaluator()
  .setMetrics(Seq(ItemMetric.MAPE, ItemMetric.RMSE))

val pipeline = new Pipeline()
  .setStages(
    Array(
      trainingRangeExtractor,
      featureTransformer,
      forecasters,
      evaluator)
  )

val model = pipeline.fit(df)
val forecast = model.transform(df)
```

Forecasting Methods



References

Bayesian Intermittent Demand Forecasting for Large Inventories
M. Seeger, D. Salinas, V. Flunkert
NIPS 2016.

Approximate Bayesian Inference in Linear State Space Models for Intermittent Demand Forecasting at Scale, submitted.

DeepAR: Probabilistic Forecasting with Autoregressive Recurrent Networks
D. Salinas, V. Flunkert, J. Gasthaus
`https://arxiv.org/abs/1704.04110`

Probabilistic Demand Forecasting at Scale
J-H. Boese, V. Flunkert, J. Gasthaus, T. Januschowski, D. Lange, D. Salinas, S. Schelter, M. Seeger, B. Wang
VLDB 2017.

Getting into touch with us

- visiting positions across academic seniorities (starting from students)
- research grants of varying scales, e.g., `http://ara.amazon-ml.com/`
- AWS Grants.
- academic engagement events
- data sets
- and more

$\Rightarrow$ contact me for more information.

Thank you!