Bayesian Machine Learning. Use Cases

Evgeny Burnaev, Alexey Zaytsev

Skoltech, Moscow, Russia

Skoltech

Skolkovo Institute of Science and Technology

Outline

Uncertainty

- Repeatable events $\Rightarrow$ classical (frequentist) interpretation of probability
- Bayesian view: probabilities provide a quantification of uncertainty
- Consider an uncertain (non-repeatable) event:
  - "whether the Arctic ice cap will have disappeared by the end of the century?"
  - we can generally have some idea how quickly we think the polar ice is melting
  - we obtain fresh data: e.g. from an Earth observation satellite we may revise our opinion on the rate of ice loss
  - we need to quantify our expression of uncertainty and make precise revisions of uncertainty in the light of new data

Probability vs. complexity (Kolmogorov)

- Data model: $y = f(\mathbf{x}, \mathbf{w}) + \varepsilon$, ε is a noise
- Quantify uncertainty about model parameters $\mathbf{w}$?
- Prior $p(\mathbf{w})$ captures our assumptions about $\mathbf{w}$ before observing the data!

Probability vs. complexity (Kolmogorov):

— It is almost impossible to predict random rare events $\Rightarrow$ their description is very long $\Rightarrow$ complex

— $\mathbf{w}$ defines "complexity" of the model

— $p(\mathbf{w})$ quantifies this complexity, as "small probability" $\equiv$ "complex"

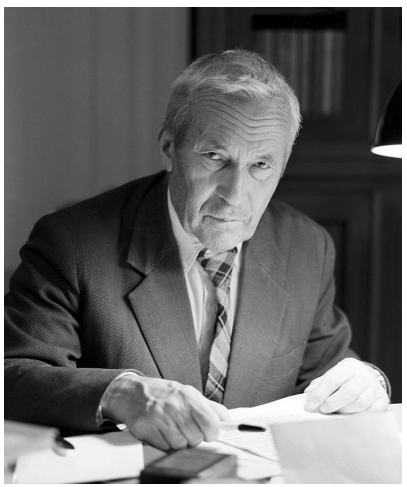


Figure – Kolmogorov A.N. (1903-1987)

Example: curve fitting problem

- Observed data $\mathcal{D}_m = \{\mathbf{X}, \mathbf{y}\} = \{(\mathbf{x}_i, y_i)\}_{i=1}^m$ influences the conditional probability $p(\mathbf{w}|\mathcal{D}_m)$:

$$p(\mathbf{w}|\mathcal{D}_m) = \frac{p(\mathcal{D}_m|\mathbf{w})p(\mathbf{w})}{p(\mathcal{D}_m)}$$

- $p(\mathcal{D}_m|\mathbf{w})$ is a likelihood function (how probable the observed data set is for different settings of the parameter vector $\mathbf{w}$)
- Normalization constant (evidence)

$$p(\mathcal{D}_m) = \int p(\mathcal{D}_m|\mathbf{w})p(\mathbf{w})d\mathbf{w}$$

- General form:

$$\text{posterior} \sim \text{likelihood} \times \text{prior}$$

$$\log \text{posterior} \sim \log \text{likelihood} + \log \text{prior}$$

Bayesian vs. Frequentist

- **Frequentist setting**:
  - $\mathbf{w}$ is a fixed parameter,
- **Bayesian setting**:
  - the uncertainty in the parameters is expressed through a probability distribution over $\mathbf{w}$,
  - we reduce uncertainty about $\mathbf{w}$ by observing more and more data
- The inclusion of prior knowledge arises naturally

Why go Bayesian?

- Regualization
- Ensembling
- Uncertainty estimation
- On-line / continual learning
- Quantization
- Compression
- ...

Ensembling

- A Bayesian neural network is an infinite ensemble of neural networks
- One sample from the posterior

$$\mathbf{w} \sim p(\mathbf{w}|\mathcal{D}_m)$$

- Predictive distribution

$$p(y^*|\mathbf{x}^*, \mathcal{D}_m) = \int p(y^*|\mathbf{x}^*, \mathbf{w}) p(\mathbf{w}|\mathcal{D}_m) d\mathbf{w}$$

- Unbiased estimate

$$\mathbb{E}_{\mathbf{w} \sim p(\mathbf{w}|\mathcal{D}_m)} p(y^*|\mathbf{x}^*, \mathbf{w}) \approx \frac{1}{M} \sum_{i=1}^{M} p(y^*|\mathbf{x}^*, \mathbf{w}_i), \;\; \mathbf{w}_i \sim p(\mathbf{w}|\mathcal{D}_m)$$

- Higher accuracy, more robust

On-line / incremental learning

- The dataset arrives in independent parts

$$\mathcal{D} = \mathcal{D}_1 \cup \mathcal{D}_2 \cup \ldots \cup \mathcal{D}_k$$

- We can train on the first dataset as usual ...

$$p(\mathbf{w}|\mathcal{D}_1) = \frac{p(\mathcal{D}_1|\mathbf{w})p(\mathbf{w})}{\int p(\mathcal{D}_1|\mathbf{w})p(\mathbf{w})d\mathbf{w}}$$

- ... And then use the obtained posterior as the prior for the next step!

$$p(\mathbf{w}|\mathcal{D}_2, \mathcal{D}_1) = \frac{p(\mathcal{D}_2|\mathbf{w})p(\mathcal{D}_1|\mathbf{w})p(\mathbf{w})}{\int p(\mathcal{D}_2|\mathbf{w})p(\mathcal{D}_1|\mathbf{w})p(\mathbf{w})d\mathbf{w}} =$$

$$= \frac{p(\mathcal{D}_2|\mathbf{w})p(\mathbf{w}|\mathcal{D}_1)}{\int p(\mathcal{D}_2|\mathbf{w})p(\mathbf{w}|\mathcal{D}_1)d\mathbf{w}}$$

- Using these sequential updates, we can find $p(\mathbf{w}|\mathcal{D})$!

1. Bayesian Probability
2. Multi-fidelity modeling and Bayesian optimization
3. Digital pre-distortion and sparse deep networks
4. Time-series modeling

Example: variable fidelity regression for an airfoil

- A lift coefficient C_l and a drag coefficient C_d describe efficiency of an airfoil
- We can calculate C_l and C_d using low fidelity solver $y_l(\mathbf{x})$ and high fidelity solver $y_h(\mathbf{x})$
- We construct a regression model:
  - geometry of an airfoil, Mach number and angle of attack are inputs $\mathbf{x}$,
  - C_l, C_d are two outputs $y(\mathbf{x})$

Model	Fidelity	CPU time (s)	Sample size
Full potentials	Low	10	10000
Euler	High	600	100

- We need to process samples from both solvers to build an accurate regression

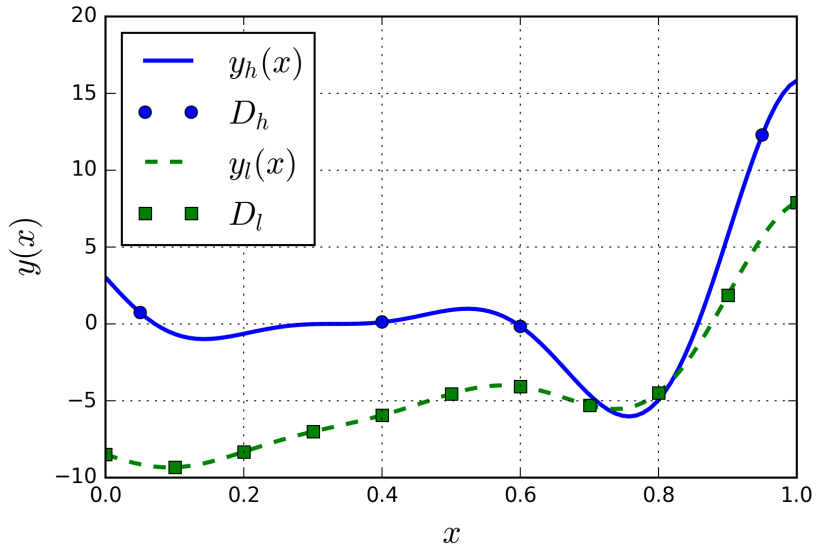
Examples of variable fidelity data

- $y_h(\mathbf{x})$ is a high fidelity function, $y_l(\mathbf{x})$ is a low fidelity function, $\mathbf{x} \in \mathbb{X} \subset \mathbb{R}^p$, $y_l \in \mathbb{R}$, $y_h \in \mathbb{R}$.

Low fidelity	High fidelity
CFD with coarse mesh	CFD with dense mesh
Full potential equations for CFD	Euler equations for CFD
Numerical experiments	Nature experiments
Noised data	Noise-free data

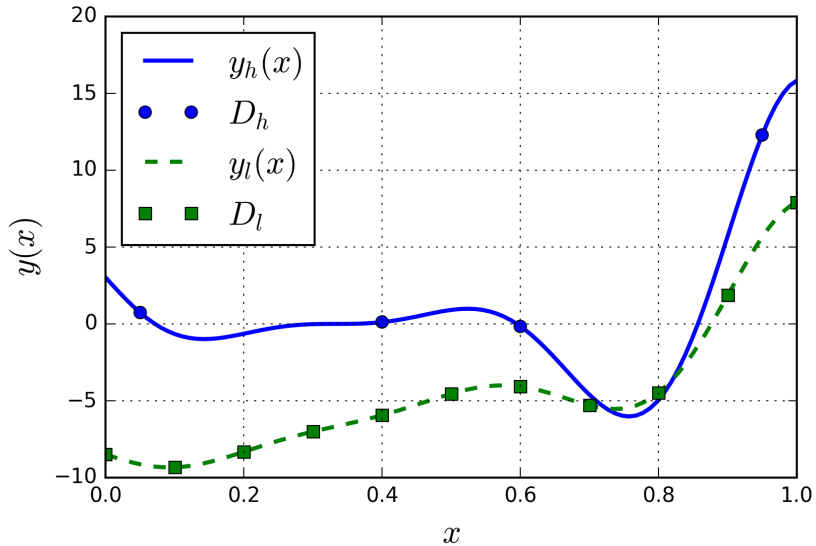
Regression problem for variable fidelity data

- A low fidelity function $y_l(\mathbf{x})$ and a high fidelity function $y_h(\mathbf{x})$ model the same physical processes, but with different fidelity
- A low fidelity sample is $\mathcal{D}_l = (\mathbf{X}_l, \mathbf{y}_l) = \left\{\mathbf{x}_i^l, y_l(\mathbf{x}_i^l)\right\}_{i=1}^{m_l}$, and a high fidelity sample is $\mathcal{D}_h = (\mathbf{X}_h, \mathbf{y}_h) = \left\{\mathbf{x}_i^h, y_h(\mathbf{x}_i^h)\right\}_{i=1}^{m_h}$ with $\mathbf{x}_i^l, \mathbf{x}_i^h \in \mathbb{R}^d$, $y_l(\mathbf{x}), y_h(\mathbf{x}) \in \mathbb{R}$.

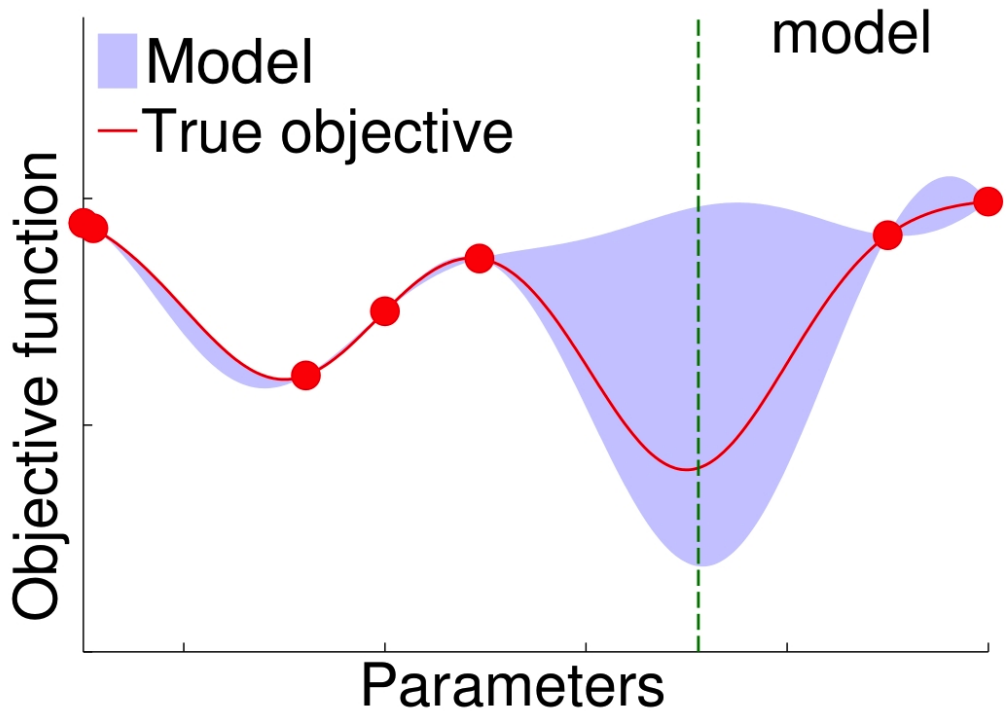


Regression problem for variable fidelity data

- Two samples $\mathcal{D}_l$ and $\mathcal{D}_h$ are available
- We want to construct a model $\widehat{y}_h(\mathbf{x}) \approx y_h(\mathbf{x})$ of the high fidelity function using both $\mathcal{D}_l$ and $\mathcal{D}_h$
- We also want to provide an uncertainty estimation for $\widehat{y}_h(\mathbf{x})$
- To mix variable fidelity data and to estimate uncertainty of prediction we can use a Gaussian process priors over high fidelity and low fidelity functions



- In high-dimensional case we need many functions evaluations to be certain in results
- Often each evaluation is costly, e.g. in case of experiments



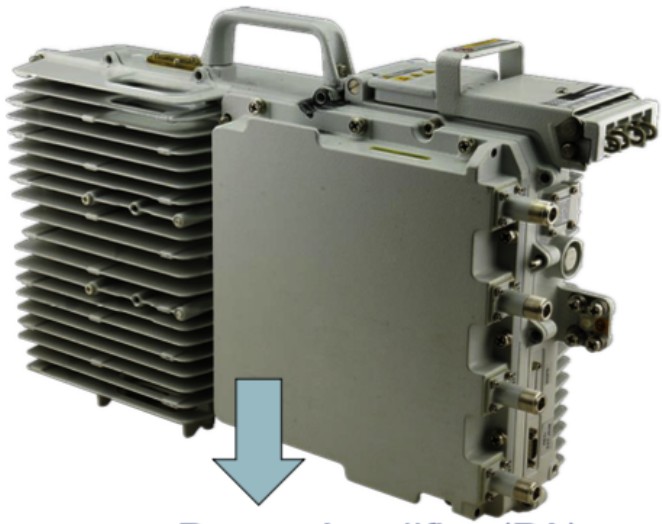
- Error bars are needed to see if a region is still promising

1. Bayesian Probability
2. Multi-fidelity modeling and Bayesian optimization
3. Digital pre-distortion and sparse deep networks
4. Time-series modeling

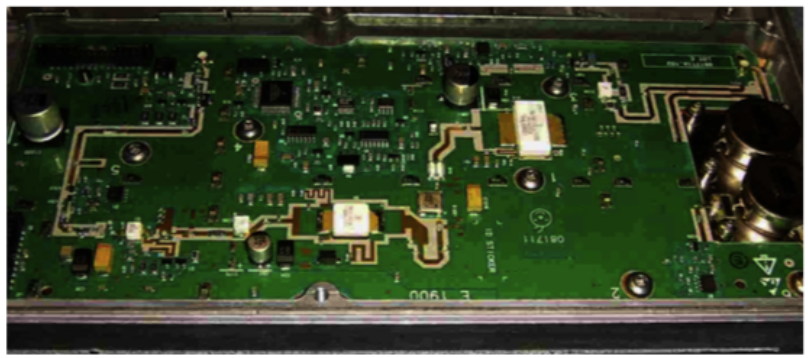
Power Amplifier (PA) as a part of base station

Macro Cell Tower

Remote Radio Head (RRH)



Microwave Power Amplifier (PA)



Digital Pre-Distortion (DPD)

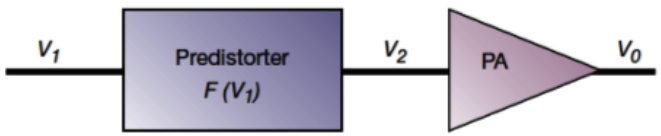
The problem - nonlinear PA behavior disturbs the transmission signal.
Current solution - apply Digital Pre-Distortion (DPD) to the input signal.

Simple example: if we knew that PA transforms the signal amplitude like so $|V\_0| = |V\_2|^{0.5}$ then DPD model could be $|V\_2| = |V\_1|^2$.

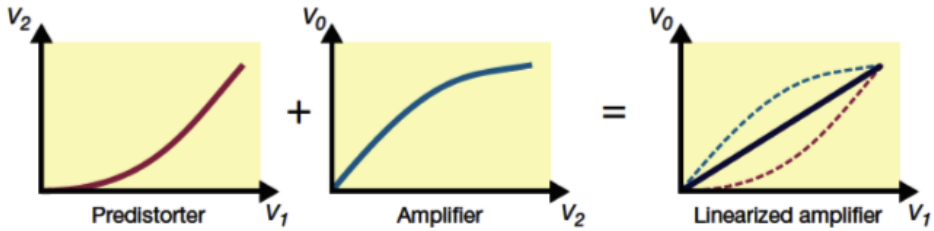
The ML problem is to estimate a function which is inverse to PA.

Main issues:

1. PA has a long memory so we have to search for an inverse function in the multidimensional space.
2. Current algorithms do not perform well.
3. The industry requires the creation of cheaper PA, compensating them with better DPD.



PA linearization (DPD) principle



Challenges

- We have to approximate function $F(\cdot)$
- The approximation should be efficient (small number of non-zero parameters) to be implemented in a hardware

- Deterministic neural network

$$p(y|\mathbf{x}, \mathbf{w}) = f(\mathbf{x}, \mathbf{w})$$

- Let us assume that $\mathbf{w} = g(\boldsymbol{\theta}, \varepsilon)$. E.g. $\mathbf{w} = \boldsymbol{\mu} + \boldsymbol{\sigma} \circ \varepsilon$ with $\varepsilon \sim \mathcal{N}(0, \mathrm{I})$ and $\boldsymbol{\theta} = (\boldsymbol{\mu}, \boldsymbol{\sigma})$. Then

$$p(y|\mathbf{x}, \mathbf{w}) = \int p(y|\mathbf{x}, \boldsymbol{\theta}, \varepsilon) p(\varepsilon) d\varepsilon = \mathbb{E}_{p(\varepsilon)}[f(\mathbf{x}, \boldsymbol{\theta}, \varepsilon)]$$

- Expected log-likelihood:

$$\mathbb{E}_{p(\varepsilon)} \log p(\mathbf{y}|\mathbf{X}, \boldsymbol{\theta}, \varepsilon) \to \max_{\boldsymbol{\theta}}$$

- Deterministic prediction:

$$p(y|\mathbf{x}, \mathbf{w}) \approx f(\mathbf{x}, \boldsymbol{\theta}, \mathbb{E}\varepsilon)$$

Variational inference for Bayesian NNs

- The posterior distribution

$$p(\mathbf{w})|\mathbf{X}, \mathbf{y}) = \frac{p(\mathbf{y}|\mathbf{X}, \mathbf{w})p(\mathbf{w})}{\int p(\mathbf{y}|\mathbf{X}, \mathbf{w})p(\mathbf{w})d\mathbf{w}}$$

- How to find it? Use (doubly stochastic) variational inference!

$$q_\varphi(\mathbf{w}) \approx p(\mathbf{w}|\mathbf{X}, \mathbf{y})$$
$$\mathrm{KL}(q_\varphi(\mathbf{w})\|p(\mathbf{w}|\mathbf{X}, \mathbf{y})) \to \min_\varphi$$

- Bayesian NN ELBO

$$\mathcal{L}(\varphi) = \mathbb{E}_{q_\varphi(\mathbf{w})} \log p(\mathbf{y}|\mathbf{X}, \mathbf{w}) - \mathrm{KL}(q_\varphi(\mathbf{w})\|p(\mathbf{w})) \to \max_\varphi$$

- We can consider a prior $\mathbf{w} = g(\boldsymbol{\theta}, \varepsilon) = \boldsymbol{\mu} + \boldsymbol{\sigma} \circ \varepsilon$ with $\varepsilon \sim \mathcal{N}(0, \mathrm{I})$ and $\boldsymbol{\theta} = (\boldsymbol{\mu}, \boldsymbol{\sigma})$
- We can impose specific hyper-prior on prior parameters and finally get sparse models automatically!
  - some parameters $\mu \approx 0$ and $\sigma \gg 1$, i.e. $\mathbf{w} \approx 0$
  - sparseness is adaptively tuned for a specific problem at hand

Macro-data

Banking performance depends on the macroeconomic situation, characterized by interbank foreign exchange rates, etc.

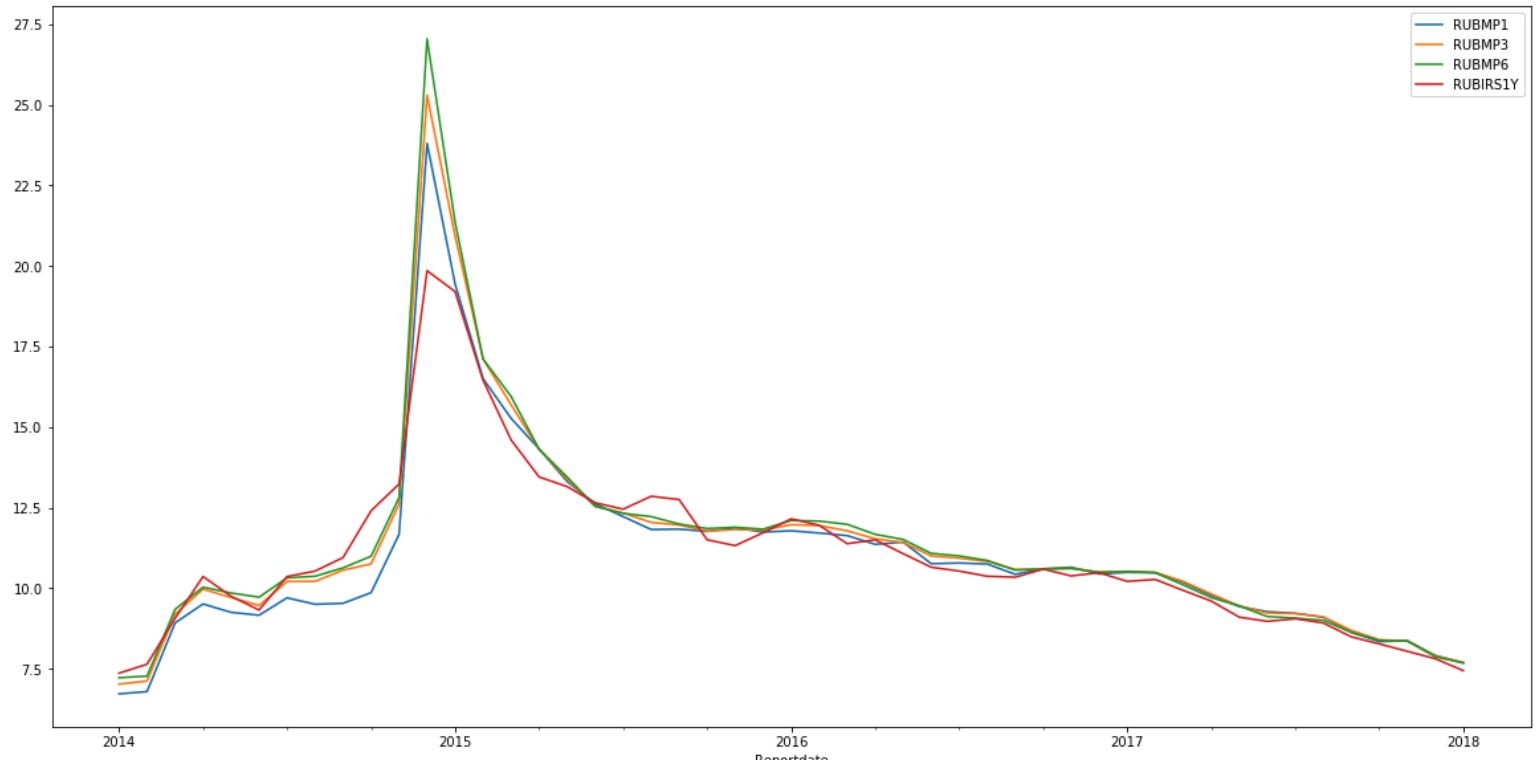


Figure – Ruble interbank rates

Vertical lines — moments of significant Deposits Churn

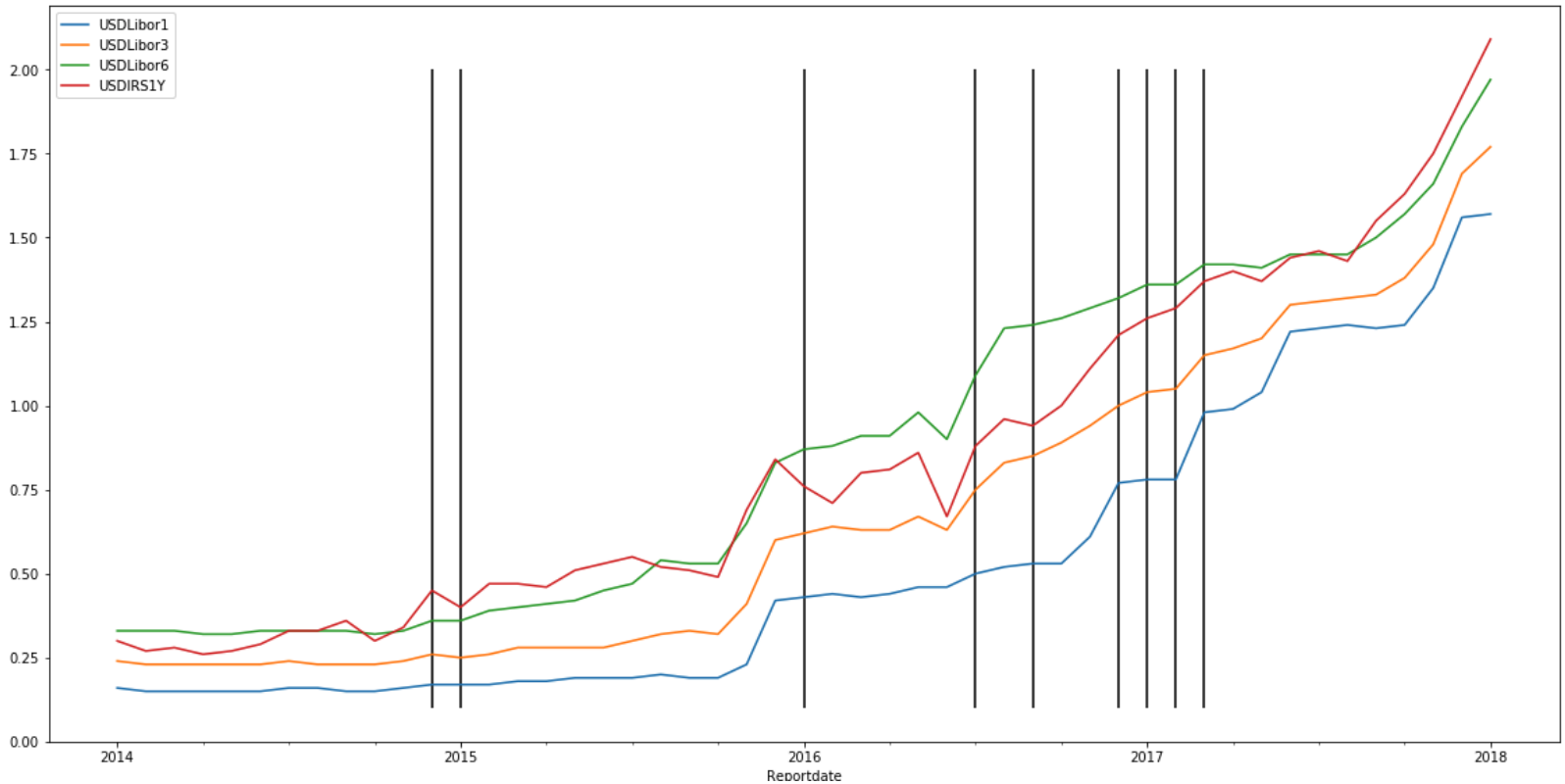


Figure – Currency interbank rates vs. time

Prediction of Income from Acquiring

Capital adequacy and liquidity risks $\Rightarrow$ long-term forecasting

- **Net Revenue from Acquiring**

Vintage — economic units grouped by some categorical characteristics and united by time interval

Forecast — the value of the **vintage**, or the **sum w.r.t. some vintages**

Acquiring

- 48 groups j (segment, territory, affiliation of a client to a bank)
- Vintage — w.r.t. to a starting month of a contract
- **Forecast**: for each group j total (w.r.t. vintages) Net Revenue 12 months ahead $(y_j^{t+1}, \ldots, y_j^{t+12})$

Code num	Segment	Terbank	Client	Num. of Vintages
0	Client CIB	Baykal bank	NON-SB	131
...	...	...	...	...
47	Client of "Corp. business"	South-West bank	SB	182

Revenue on a vintage level

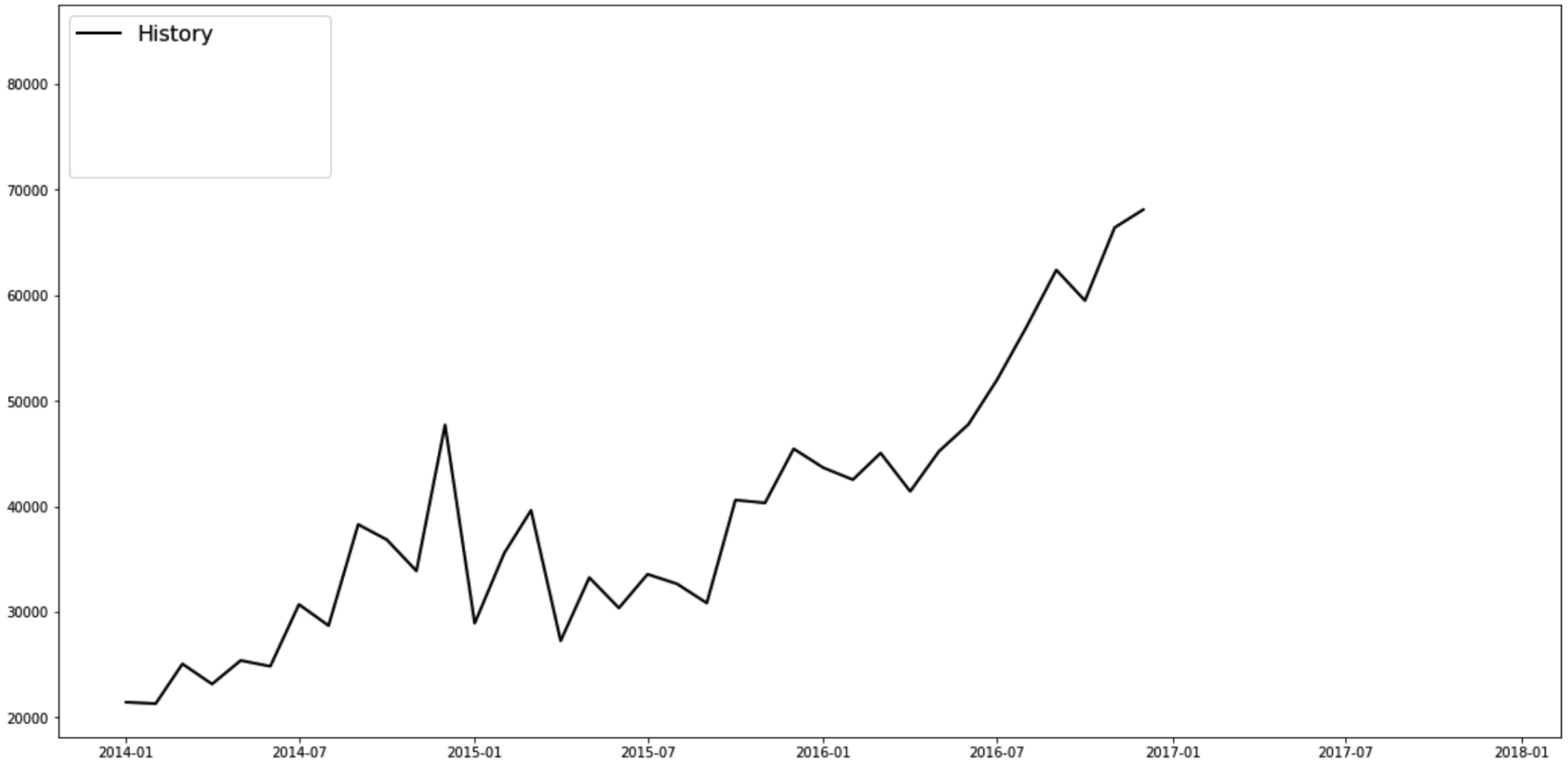


Figure – Revenue for a single vintage

Total revenue on a group level

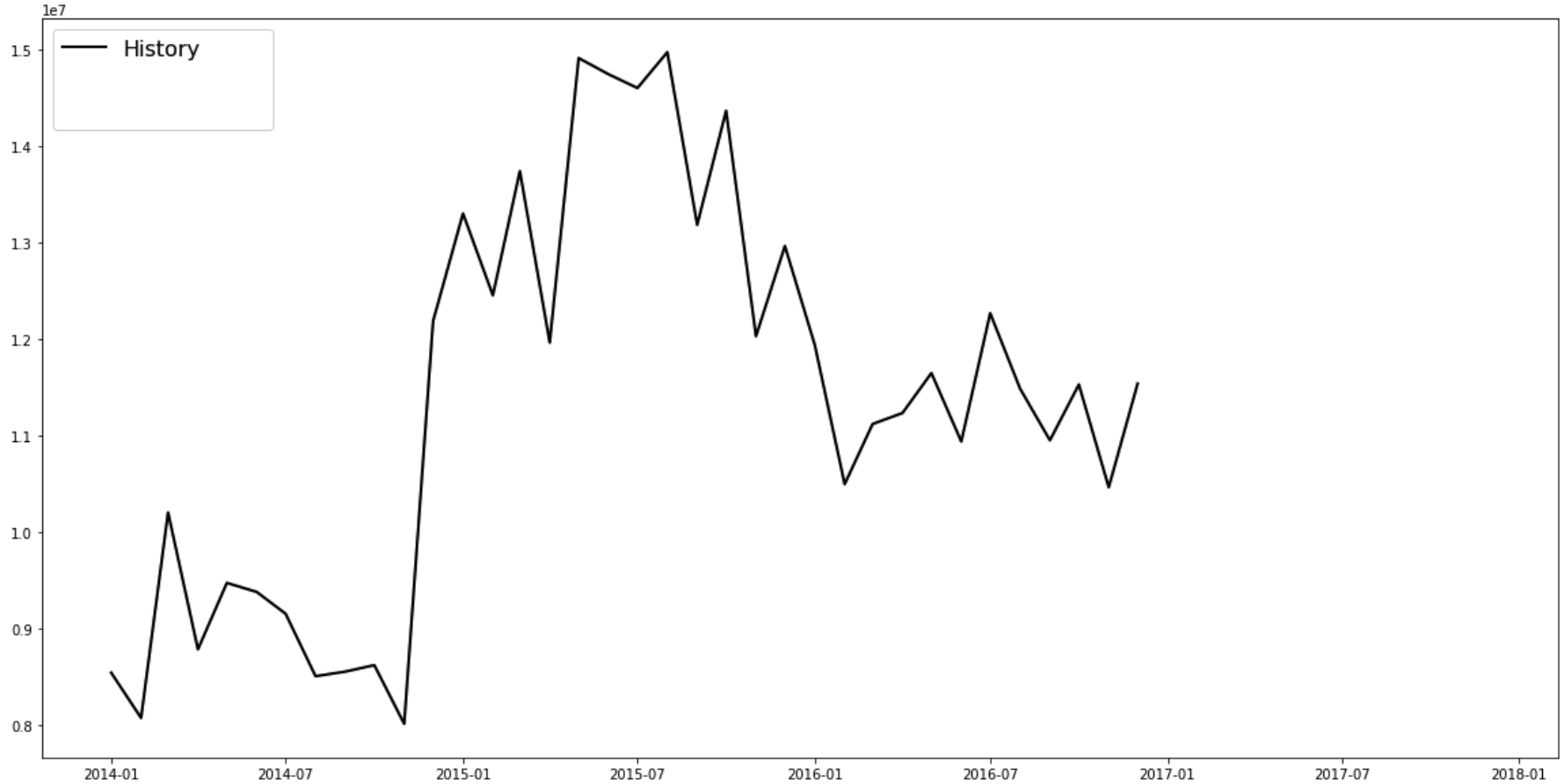


Figure – Total revenue on a group level

Acquiring: properties of data

- Forecast dynamics of time series $x_t \in \mathbb{R}^{n_x}$ ($n_x > 7000$ vintages)
- Time series are dependent due to territorial proximity and/or similar businesses

Idea

- Time-series close in a latent space should have similar predictions
- The prediction model must be different for distant latent points

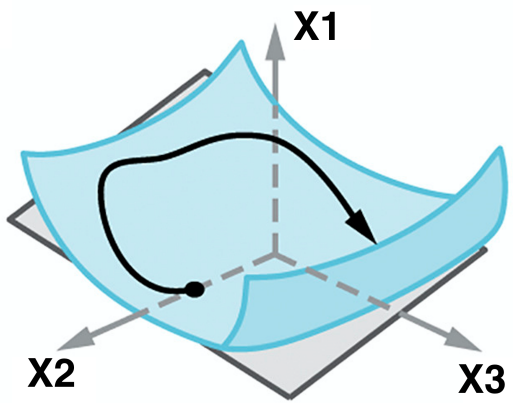


Figure – Dynamics in latent space

Dynamics in latent space

Dataset $\mathcal{D} = \{x_t, u_t, x_{t+1}\}_{t=1}^{T}$:

- $x_t \in \mathbb{R}^{n_x}$, $n_x \gg 1$ — time-series at moment t (revenue values in a vintage)
- u_t — control at time t (macro-data)

Assumptions:

- Dynamics of x_t is complex
- We can find a representation $z_t \in \mathbb{R}^{n_z}$, $n_z \ll n_x$, such that

$$z_{t+1} = A(z_t)z_t + B(z_t)u_t + o(z_t)$$
$$x_t = f(z_t)$$

$\Rightarrow$ Neural network generalization of Kalman filter

Qualitative description of the model

- **Control Embedding**

$$\text{Macro-data} \Rightarrow \text{features } u_t$$

- **Encoder**

$$x_t \Rightarrow z_t \sim \mathcal{N}(Encoder_\mu(x_t), Encoder_\sigma(x_t))$$

- **Transition in Latent Space**

$$z_t \Rightarrow z_{t+1} = A(z_t)z_t + B(z_t)u_t + o(z_t)$$

- **Decoder**

$$z_{t+1} \Rightarrow x_{t+1} \sim \mathcal{N}(Decoder_\mu(z_{t+1}), Decoder_\sigma(z_{t+1}))$$

Forecast on a vintage level

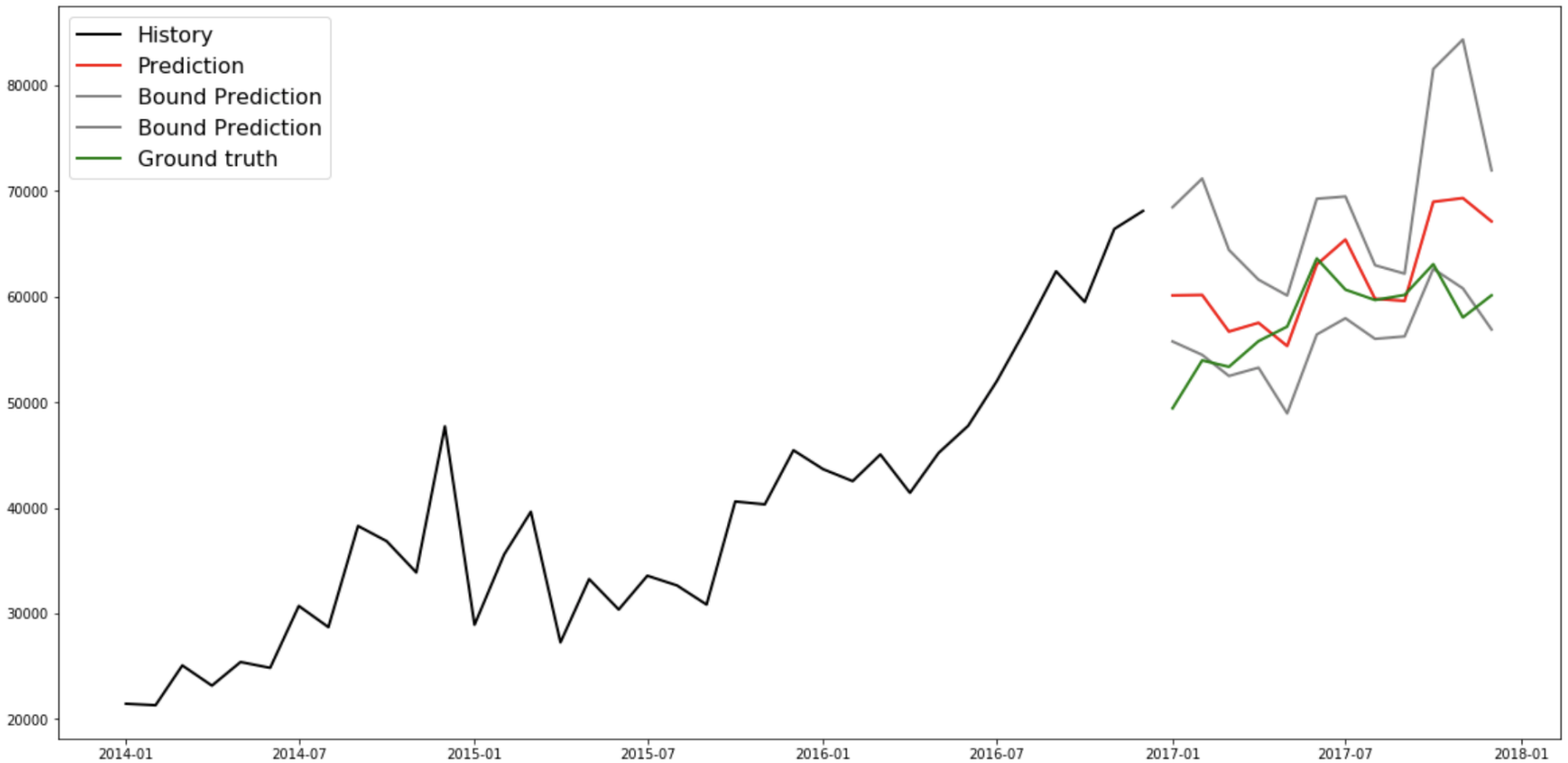


Figure – Revenue forecast for a single vintage

Forecast on a vintage level

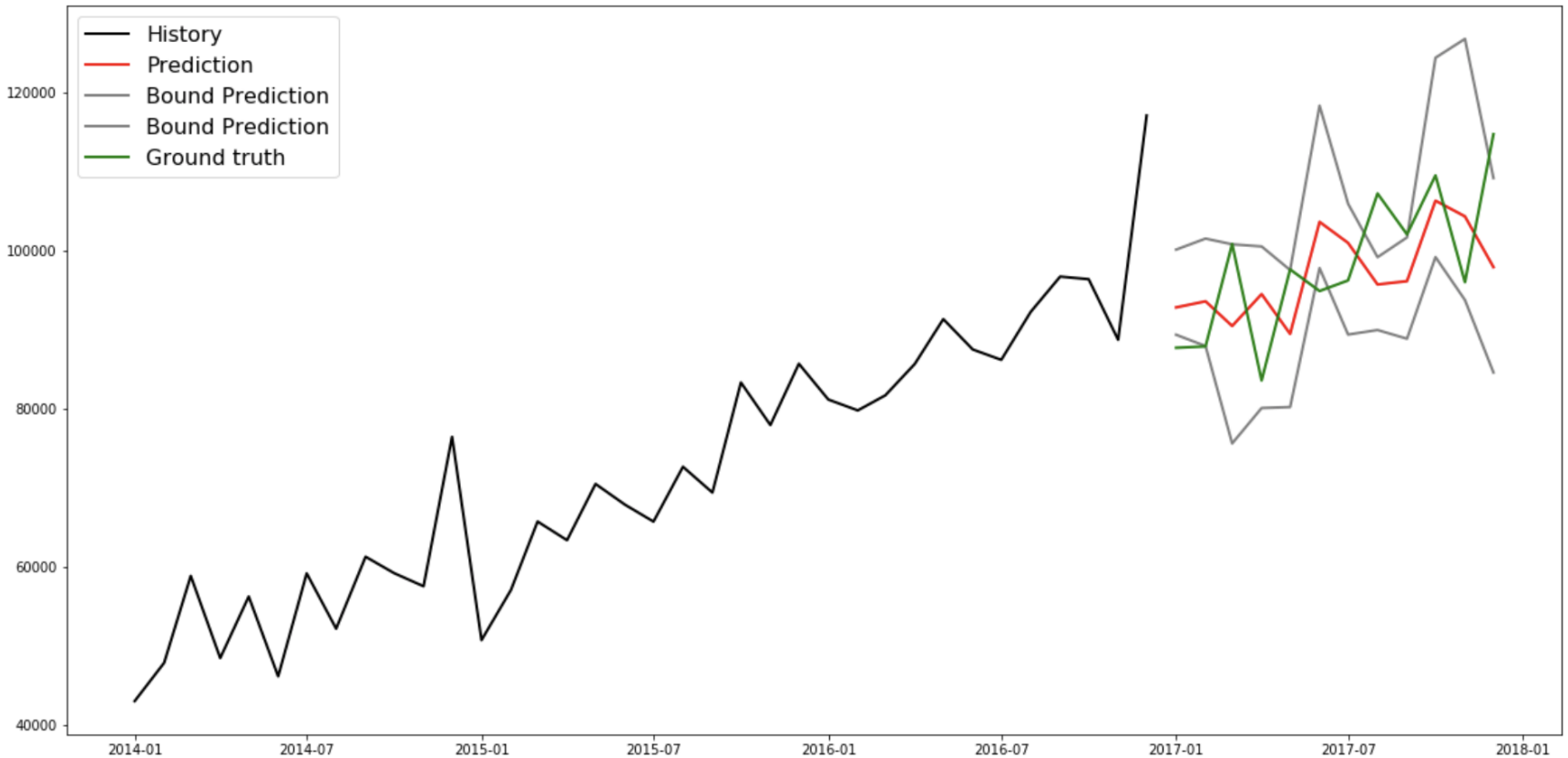


Figure – Revenue forecast for a single vintage

Forecast of a total revenue on a group level

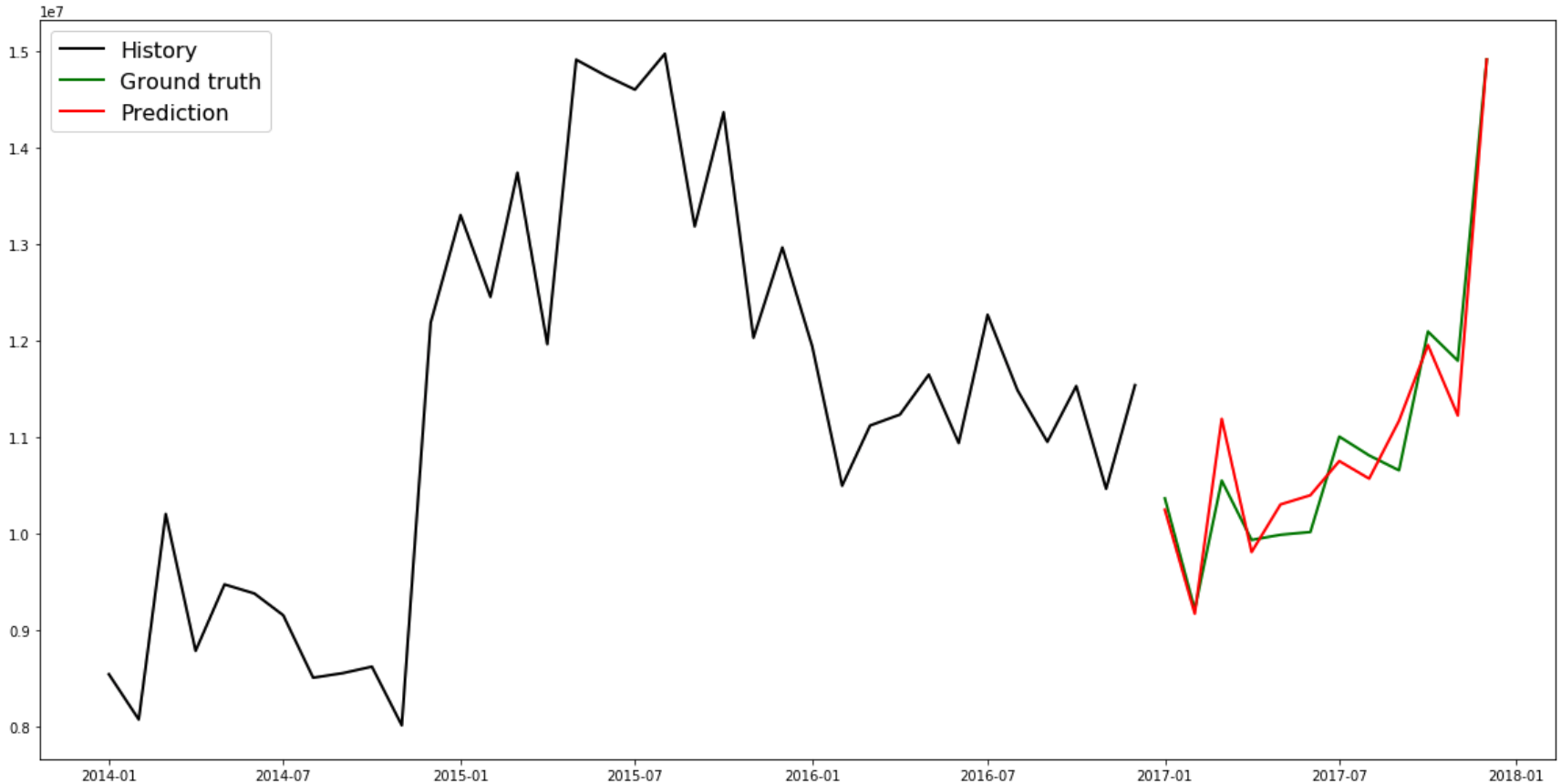


Figure – Total revenue forecast on a group level