Blue Yonder
Fulfill your potential™

Cyclic Boosting

A Pure-Python, Explainable, and Efficient ML Method

Felix Wick, June 2024

This is old stuff.

But it just works.

Cyclic Boosting ML Methods

family of off-the-shelf, general-purpose supervised machine learning methods for both regression and classification tasks (focus on structured data)

closest relatives: Generalized Additive Models (not a deep learning approach)

main difference: estimation of factors for each bin of the different features (instead of estimation of parameters like coefficients in linear regression or weights in neural networks)

→ individual explainability

scientific papers describing the methods: Cyclic Boosting, Demand Forecasting with Cyclic Boosting

Cyclic Boosting Library

scikit-learn-like usage of library

open source: https://github.com/Blue-Yonder-OSS/cyclic-boosting

Python package: pypi

documentation: readthedocs

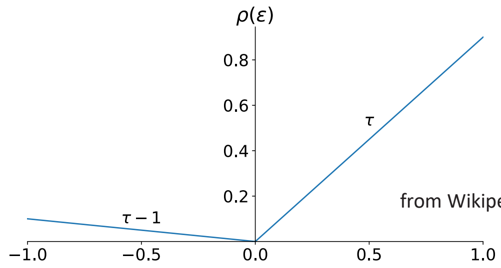
Cyclic Boosting | Different Modes/Scenarios

Mulitplicative Regression	Additive Regression	Classification	Negative Binomial Width	Exponential Price Elasticity	Background Subtraction
(conditional mean)	(conditional mean)	(probability)	(dispersion parameter)	(elasticity parameter)	(conditional mean)
$Y \in [0, \infty)$	$Y \in (-\infty, \infty)$	$Y \in [0, 1]$	$Y \in [0, 1]$	$Y \in [0, \infty)$	$Y \in (-\infty, \infty)$
Poisson / Negative Binomial distribution (link function *ln*)	Gaussian distribution (link function *identity*)	Bernoulli distribution (link function *logit*)	Negative Binomial distribution (link function *logit*)	exponential distribution (link function *ln*)	Gaussian distribution (link function *identity*)
example	*example*	*example*	*example*	*example*	*example*
demand forecasts (mean)	profit predictions	churn probability	demand forecasts as full probability distributions	individual price-demand elasticities	individual causal effects, e.g., customer targeting

Also Possible: Quantile Regression

quantile regression: estimate quantile τ of distribution instead of conditional mean by minimizing pinball loss

$$(1-\tau)\sum_{y_i<\hat{q}_i}(\hat{q}_i - y_i) + \tau\sum_{y_i\geq\hat{q}_i}(y_i - \hat{q}_i)$$



from Wikipedia

instead of squared error loss (choice of loss function defines point estimate)

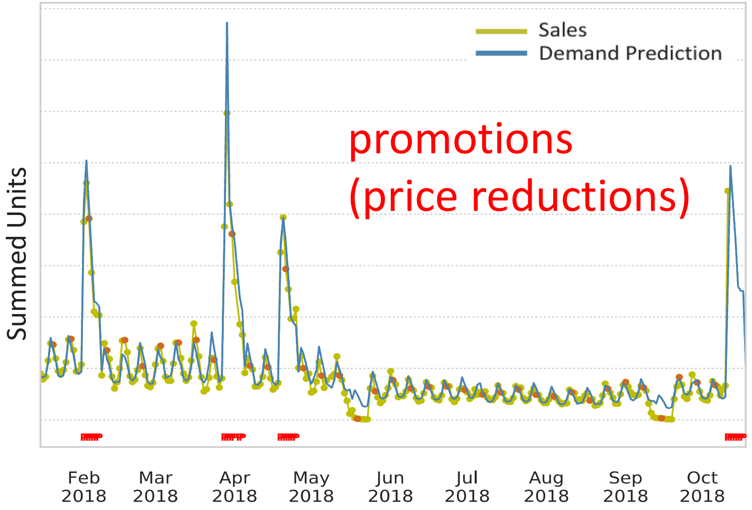
possible with various ML methods, including neural networks, tree-based methods (like random forests or gradient boosting), and Cyclic Boosting (minimized in each feature bin)

also: generic loss mode (e.g., maximum likelihood)

Example Use Case: Demand Forecasting

many individual time series to consider

typical retail grocery chain:
- products (items): ~20k
- locations (stores): ~500
- daily/hourly aggregated sales



advantages of machine learning over traditional univariate time series forecasting

combined learning on all time series of product-location combinations (rather than separately optimizing individual time series)
→ **reduces variance** by exploiting commonalities

natural consideration of many exogenous variables (prices, promotions, holidays, weather, ...)
→ **reduces bias**

to be noted:
- categorical features important (products and locations → high cardinality)
- mainly multiplicative effects
- demand (approximately) following Poisson (or rather negative binomial) distribution

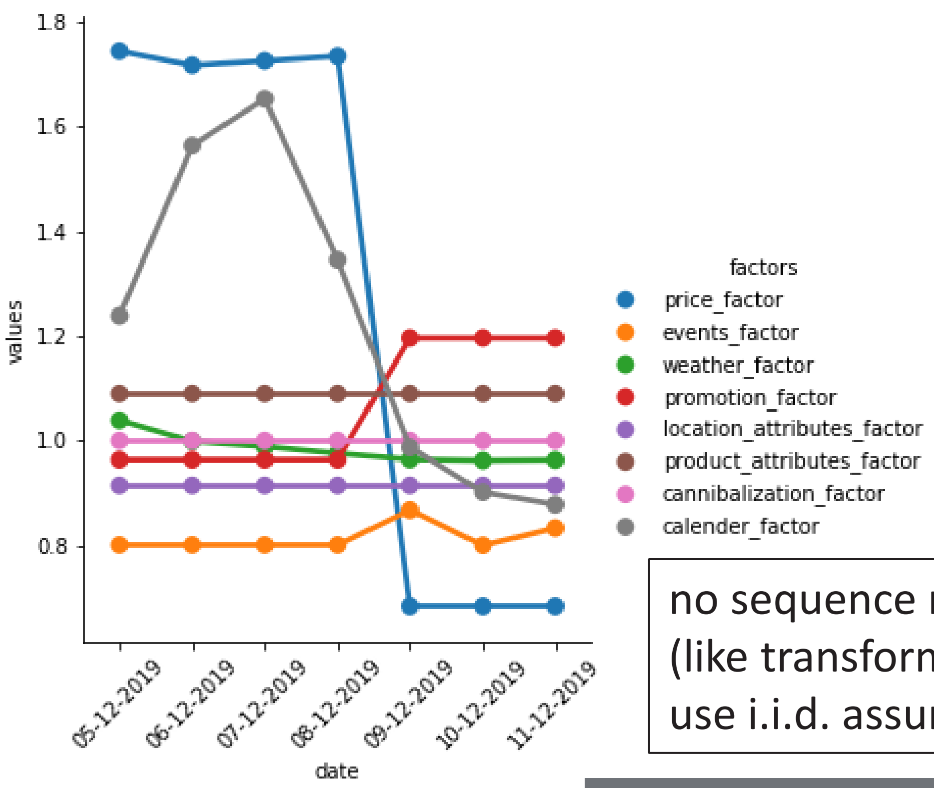
Cyclic Boosting - Prediction View | Individual Explainability

multiplicative model

variation proportional to level

Cyclic Boosting in multiplicative regression mode

individual item-store-day predictions



no sequence model (like transformers), use i.i.d. assumption

Cyclic Boosting allows for detailed explanation of each individual prediction by means of contributions (in form of factors) of each feature in the model.

prediction: look up learned factors of relevant bin for each feature

$$\hat{y}_i = \mu \cdot \prod_{j=1}^{p} f_j^k \quad \text{with} \quad k = \{x_{j,i} \in b_j^k\}$$

global target average

product over factors for all p features in corresponding bins of sample i

bin k of feature j

data binning of features (think of histograms)

do not confuse explainability with causality though
→ need for causal assumptions (e.g., specific smoothing)

Cyclic Boosting Training

Coordinate Descent: Boosting-like Update of Factors

Cyclic Boosting fitting procedure with three loops

```
while (stop criteria)        iterations
…
    for (features)           sequential
    …
        for (samples)        parallel
```

1. calculate global average μ, initialize all factors to 1
2. cyclically iterate through features and calculate factors for each feature bin (corresponding to minimization of quadratic loss)

multiplicative regression mode (other modes work accordingly)

for simplicity: show only non-aggregated mode

$$f_j^k = \sum_{x_{j,i} \in b_j^k} y_i \Big/ \sum_{x_{j,i} \in b_j^k} \hat{y}_{j,i} \qquad \Big| \qquad \hat{y}_{j,i} = \mu \cdot \prod_{l \neq j} f_l^k$$

bin k

feature j

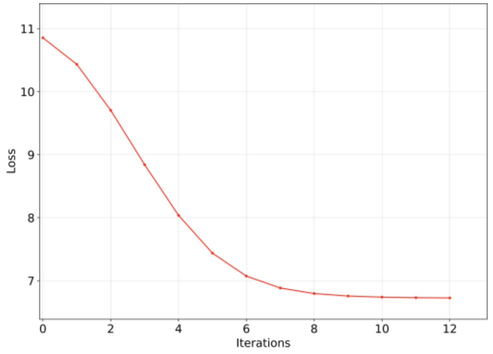
numerator: target values

sum over all samples i in bin k of feature j

denominator: predictions excluding factor from current feature

factors for corresponding feature bins of sample i

product over all features excluding j



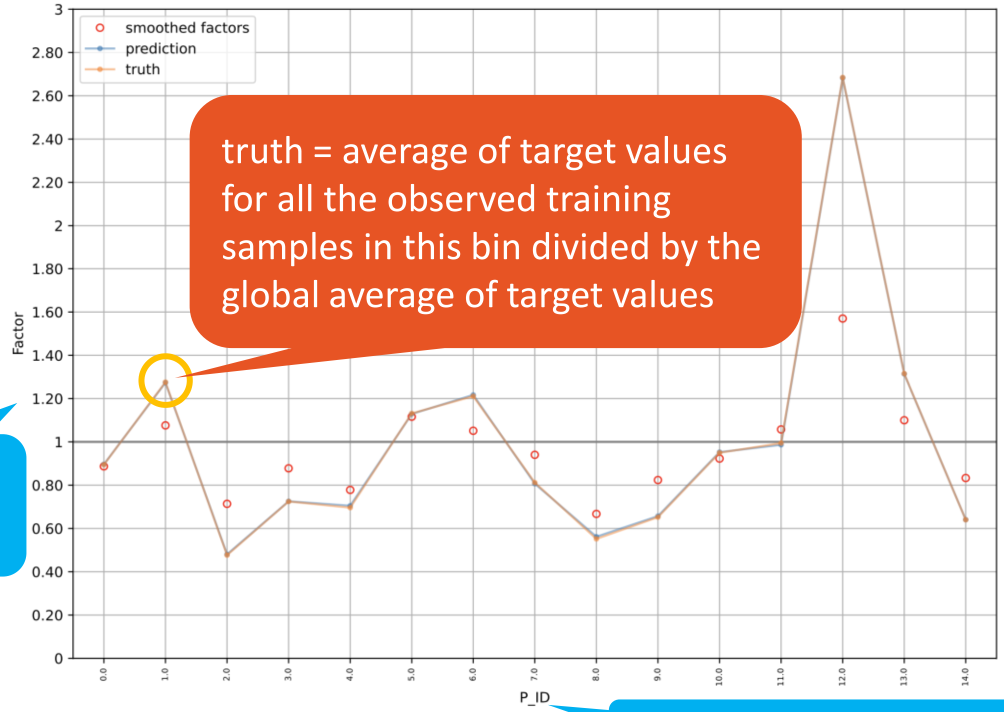
3. stop according to MAD or MSE criteria at end of iterations (full feature cycles) or when reaching given maximal number of iterations

Binning

categorical features retain original categories (learning of specific factor for each of the bins)
→ supporting categorical features with high cardinality

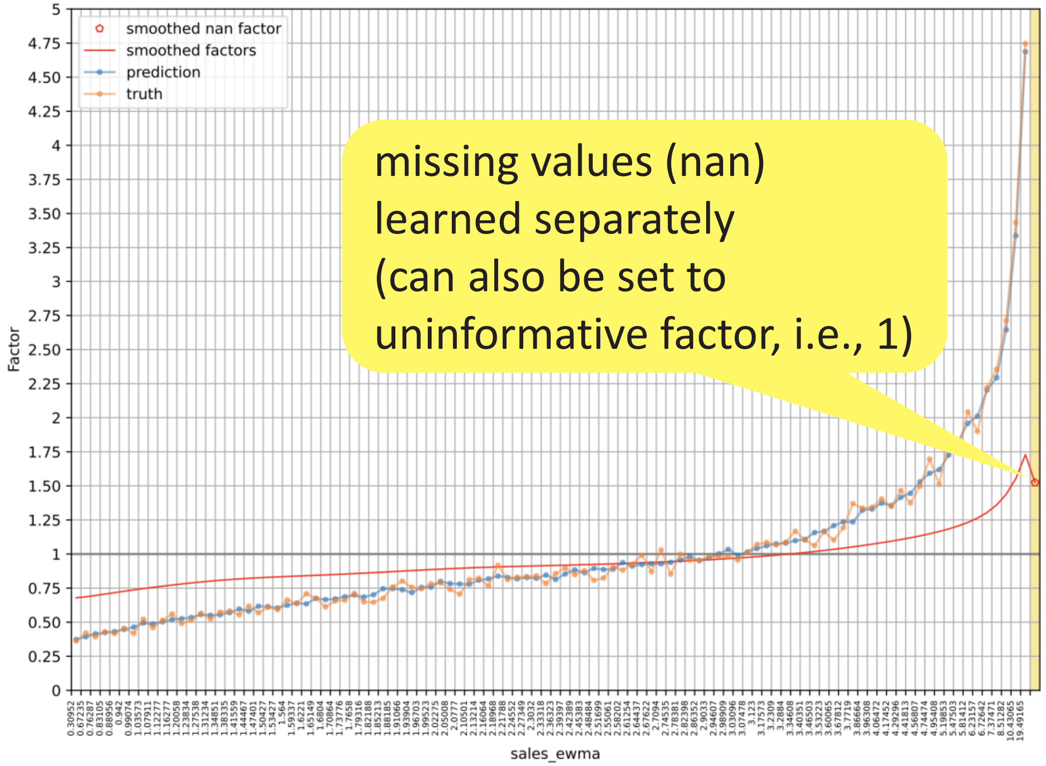
continuous features discretized to:

- either having same bin width (equidistant binning)
- or containing approximately same number of observations (equistatistics binning) with different bin widths



y axis: factors for each bin

x axis: e.g., product IDs as bins



local optimization in each bin: allows learning of rare effects with low bias

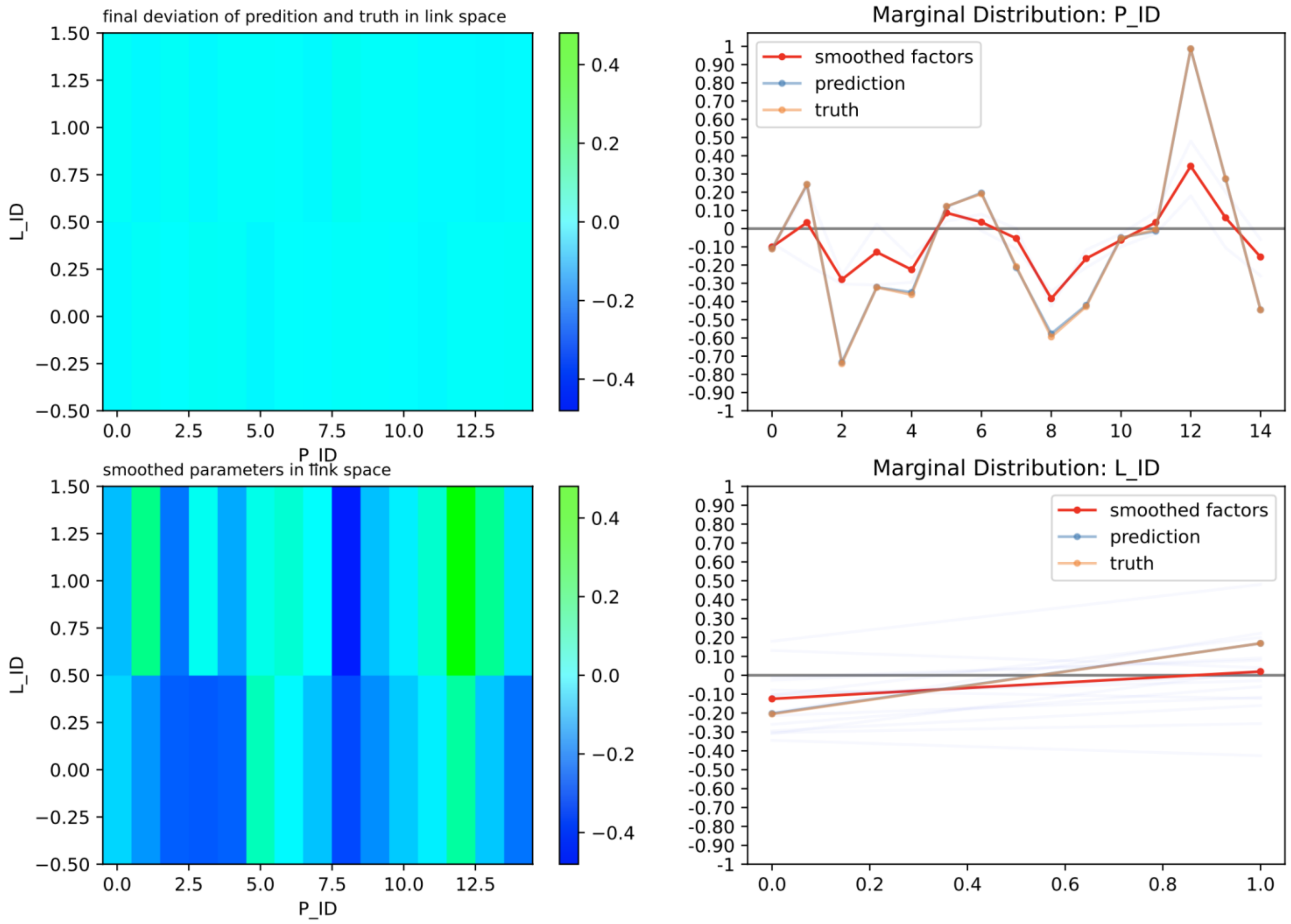
Interaction Terms

e.g., different holiday effects for different products

require even more local optimization (rare effects)

→ include binned interaction terms (e.g., 2D or 3D)

can (partly) enable hierarchical model structure (in combination with coordinate descent): interaction terms with product groups, products, locations



Smoothing

to avoid overfitting:
regularization (smoothing) across bins
→ **drastic reduction of variance** by ignoring fluctuations

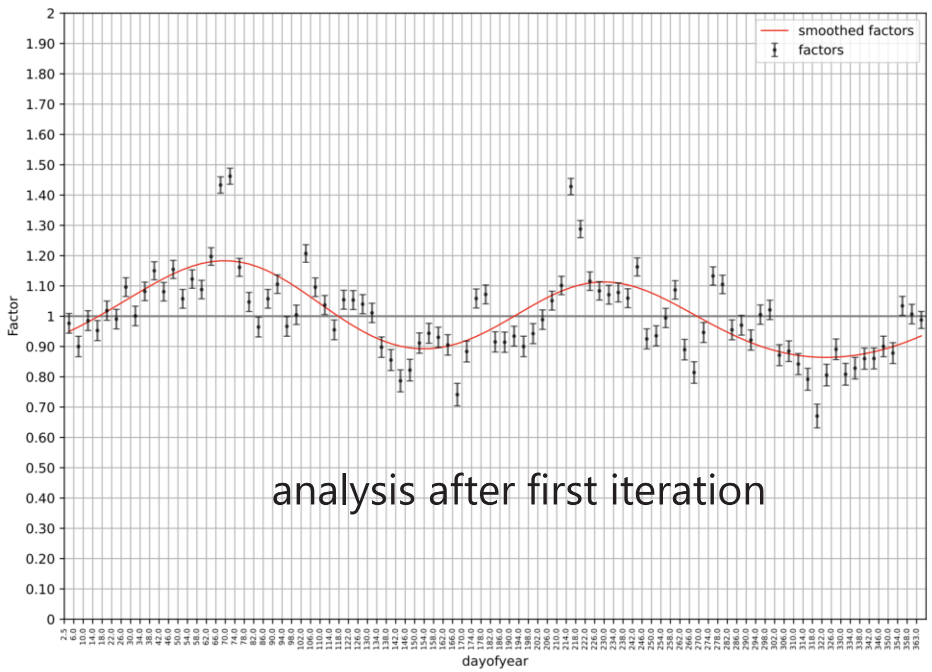
separate smoothings in each iteration

in general: orthogonal polynomials

another way to include *prior knowledge* via

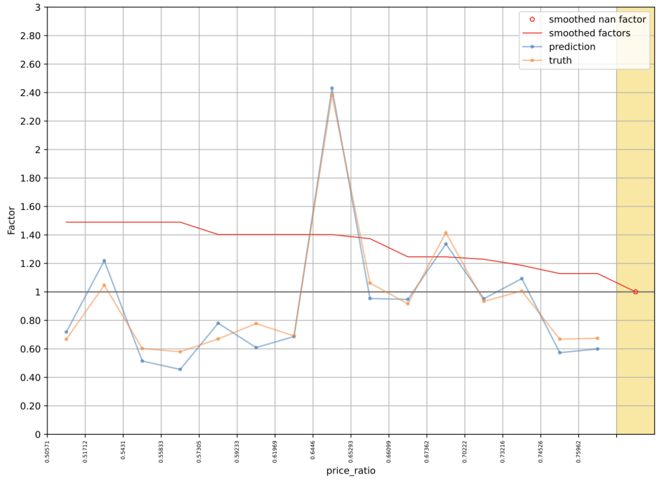
- monotonic requirements
- sinusoidal functions
- (piecewise) linear

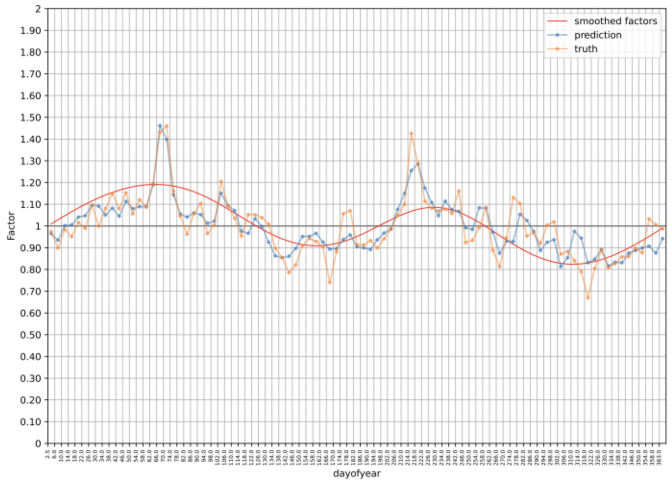
use fitted functions (smoothed factors) instead of original factors



analysis after first iteration

analysis after last iteration:





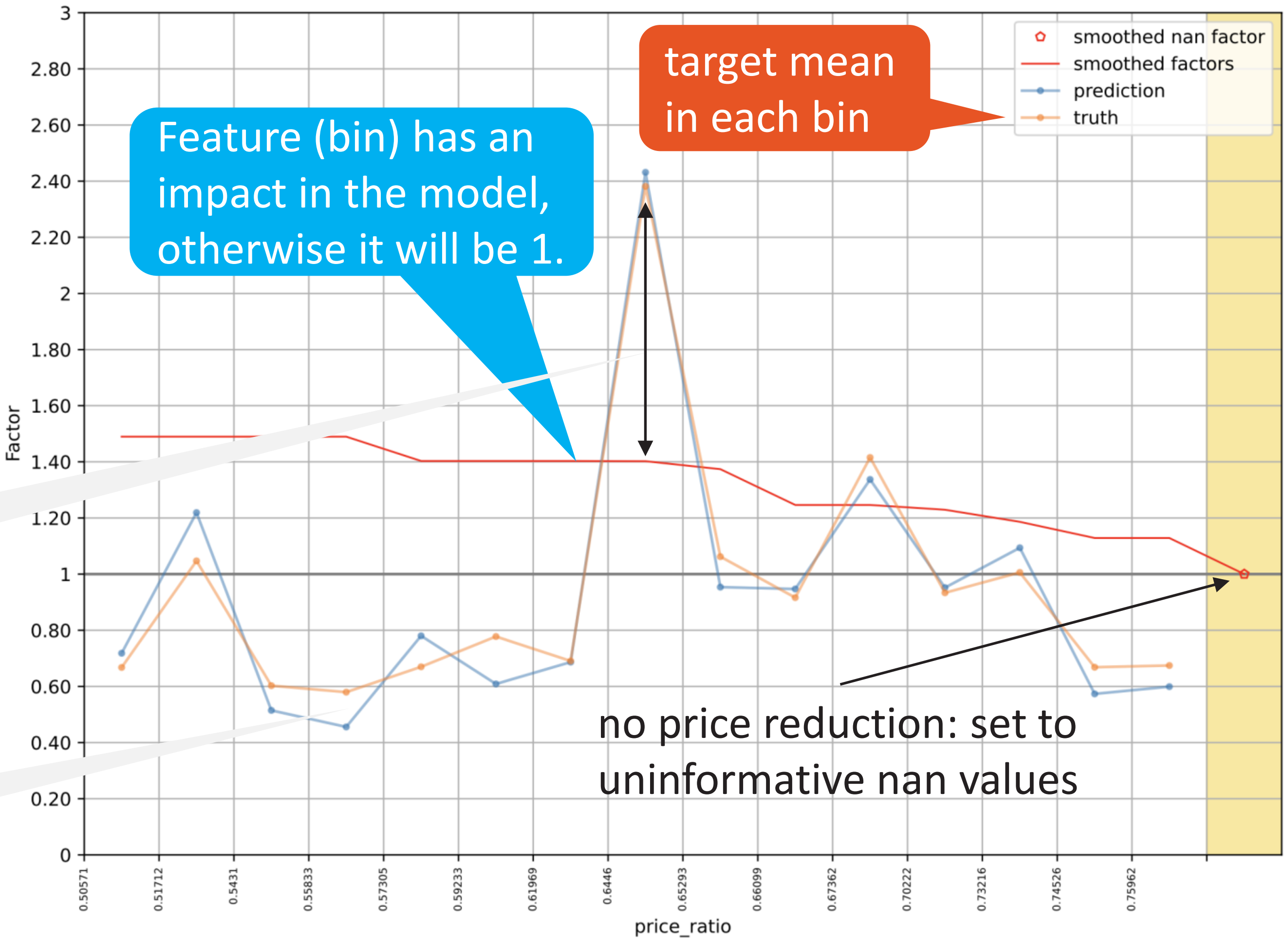
Analysis Plots

analysis after last iteration:

support EDA and modeling

model transparency → ease of development

automatically generated



Deviations between smoothed factors and predictions come from correlations with other features.

Deviations of predictions from truths show potential model weaknesses (biases) in different bins.

Cyclic Boosting | Characteristics

TOPOLOGY

Generalized Additive Model

OPTIMIZATION

cyclic coordinate descent

PREREQUISITE

binning of features

REGULARIZATION

Bayesian updates and smoothing of factor distributions

similarities to:	backfitting, forward-stagewise modeling (aka boosting)	LightGBM

Off-The-Shelf Method for Structured Data

- “simple” algorithm, but robust and fast
- few hyperparameters to be tuned
- not much data pre-processing needed
- easily configurable for different data types
- supporting missing values in input data
- assisting model development with individual analysis plots for features
- allowing buiding of complex models by means of interaction terms

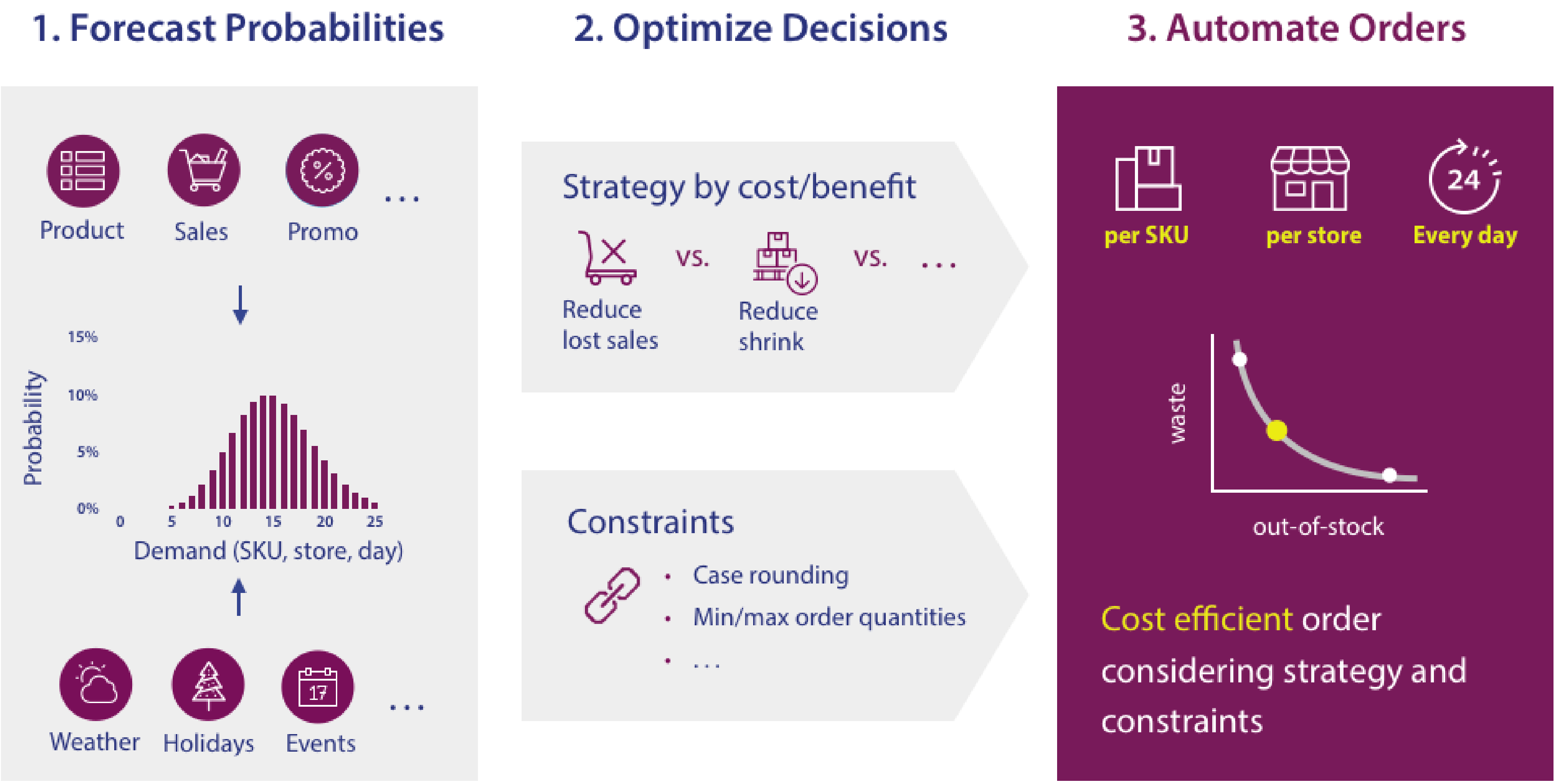
- (multiplicative or additive) regression (location parameter)
- classification

Other Modes: Width Prediction (Scale)

important busines application: automated replenishment

full, individual PDF predictions (e.g., probability distributions for each product-location-day combination)

by means of separate ML models for mean and variance (actually, indirect prediction of variance via dispersion parameter), assuming negative binomial distribution of target (e.g., demand) in maximum likelihood estimation



We understand internal & external factors. By forecasting the probability density, we know the risks of e.g. lost sales vs. waste.

Knowing these risks we calculate the order, which minimizes these risks and balances them according to strategy set by the retailer.

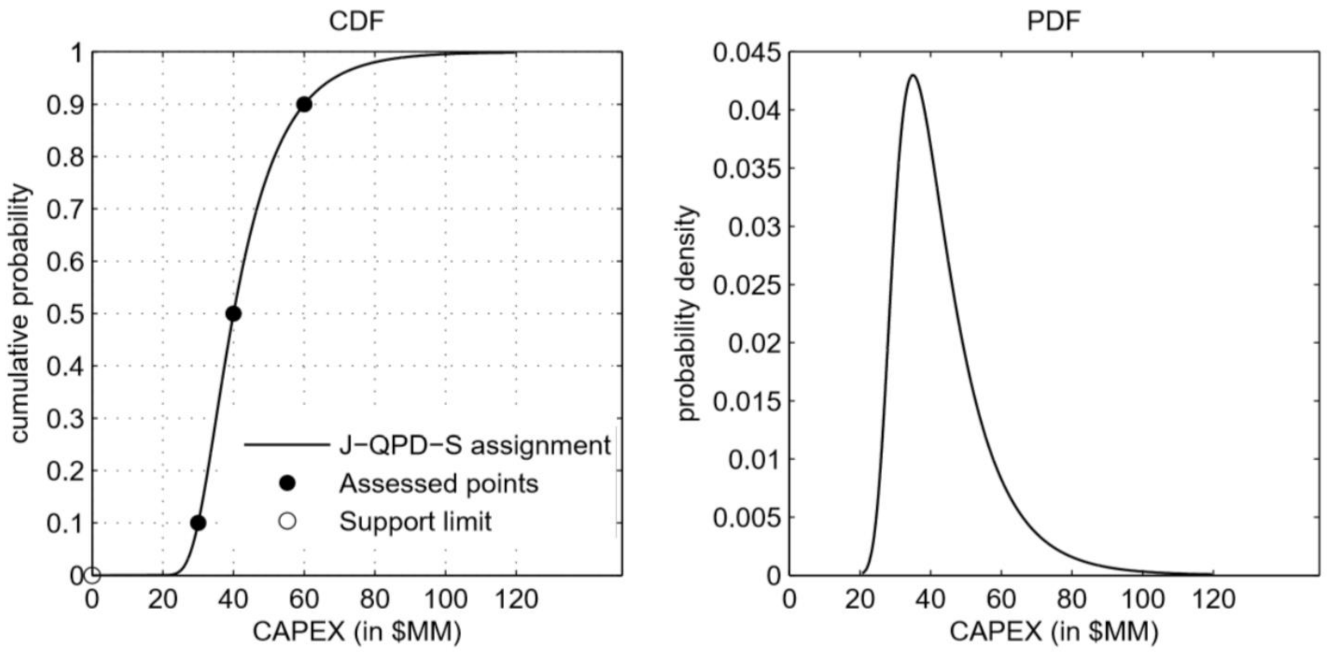
Alternative: Quantile-Parameterized Distributions (QPD)

idea: approximate full, individual probability distribution for each sample by using estimated quantiles

- **e.g., from Cyclic Boosting's quantile regression mode (or any other quantile regression method)**
- **quantiles as parameters of smooth distribution → no fitting, no strict functional assumption**

Johnson QPDs (J-QPD) are parameterized by symmetric quantile triplet

implemented in Cyclic Boosting



J-QPD

using factors in exponential terms

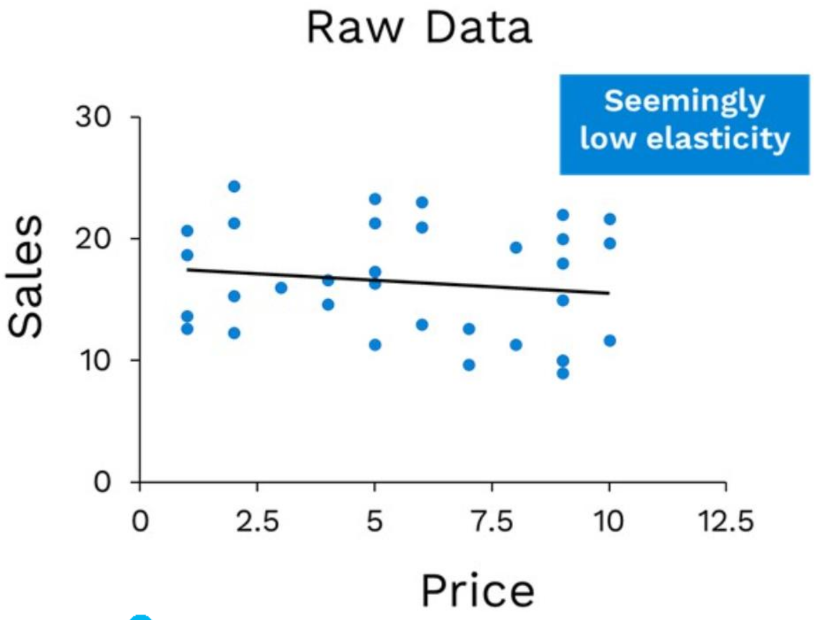
using negative sample weights

Other Modes: Elasticity Prediction (Shape), Background Subtraction

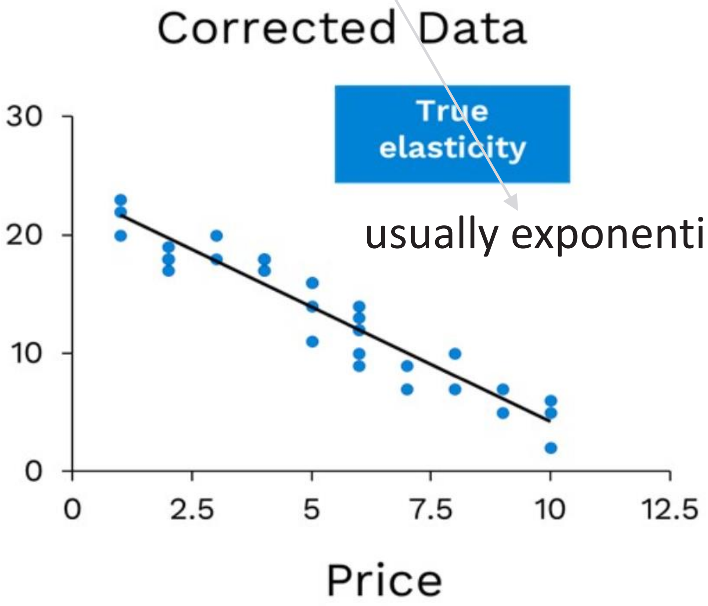
other business application: demand shaping by causal inference (mainly beyond ML/CB), examples:

- **dynamic pricing: influence demand of different products by price setting**
- **customer targeting: influence individual customer demand by couponing (subtract unaffected customers)**

confounded effect:



after de-confounding:



usually exponential

use for pricing policies: