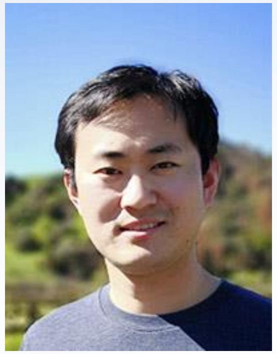
JD.COM

Data Science in Retail-as-a-Service

KDD 2018
August 2018
London, UK

Tutorial Contributors

Zuo-Jun (Max) Shen
Chief Supply Chain Scientist
JD.COM



Rong Yuan
Research Scientist
Silicon Valley R&D Center
JD.COM

Di Wu
Principal Scientist
Silicon Valley R&D Center
JD.COM

Jian Pei
Vice President
Head of Big Data and Smart Supply Chain
JD.COM

Other Contributors

- Yi Pan, Data Scientist, Silicon Valley R&D Center, JD.COM
- Dongyang Wang, Data Scientist, Silicon Valley R&D Center, JD.COM
- Chenyou Fan, Research Scientist, Silicon Valley R&D Center, JD.COM
- Yan Yan, Senior Research Scientist, Silicon Valley R&D Center, JD.COM
- Titouan Jehl, Research Scientist Intern, JD.COM, University of California, Berkeley
- Mei Wang, Research Scientist Intern, Silicon Valley R&D Center, JD.COM
- Yuze Zhang, Research Scientist Intern, JD.COM, Georgia Institute of Technology
- Xiaoyue Li, Research Scientist Intern, JD.COM, University of California at Davis
- Shaojie Cai, Senior Program Manager, Silicon Valley R&D Center, JD.COM
- Yulun Wu, Program Manager Intern, JD.COM, Georgia Institute of Technology

Outline

- Introduction
- Understand Customers
- Connect To Customers
- Serve Customers

Introduction

Presenter: Jian Pei

About JD.COM

JD.COM

Largest
retailer in China

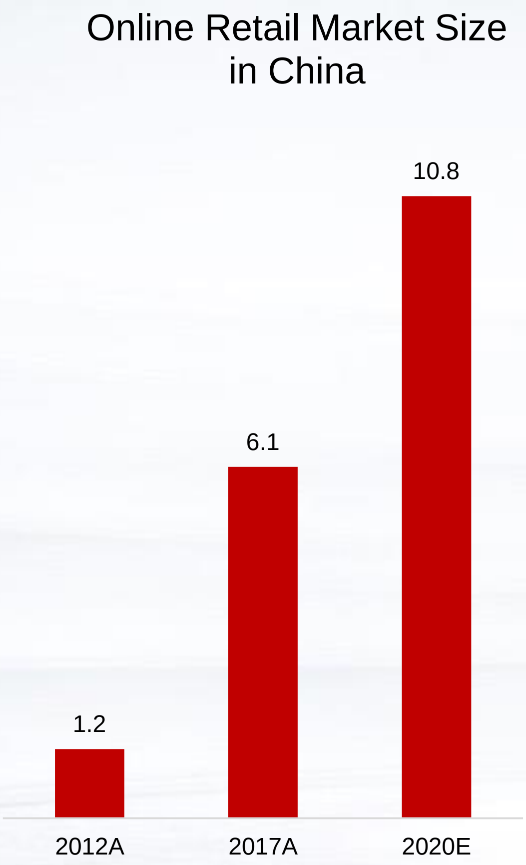
3rd Largest
internet company globally

301.8 million
active customers

90%
orders fulfilled same- or next-day

Tencent, Walmart and Google
are strategic partners

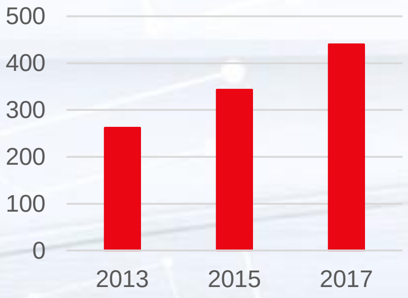
Growth and Scale



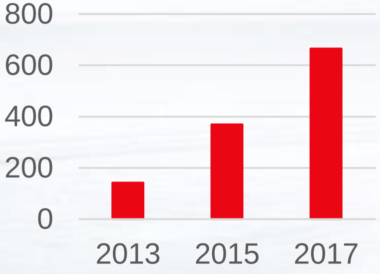
301.8 million	Customers
Millions of	Owned SKUs
170,000+	Merchants
99%	Population coverage
90%	Orders delivered in 24 hours
515	Warehouses

Growth and Scale

US Retail Ecommerce Sales 2013-2017 (Billion Dollars)



China Retail Ecommerce Sales 2013-2017 (Billion Dollars)



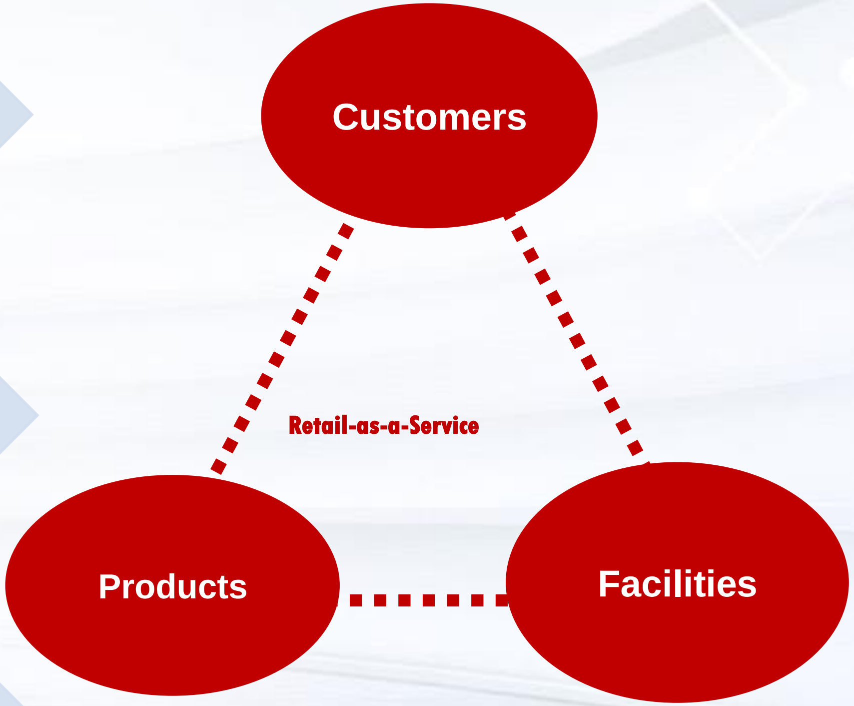
Worldwide Retail E-Commerce Sales 2012-2017 (Billion Dollars)

Year/Country	2012	2013	2014	2015	2016	2017	CAGR
US	225.3	262.3	300.6	343.3	390	440	14.30%
China	70.88	141.6	249.4	369.7	506.3	665.1	56.50%
UK	60.16	68.88	77.84	86.4	94.17	101.7	11.10%
Japan	77.6	70.75	76.85	83.3	89.26	95.08	4.10%
Germany	38.13	42.66	46.69	50.53	54.45	58.38	8.90%
France	30.22	34.21	38.36	42.62	46.41	50.25	10.70%
Canada	18.36	21.61	25.37	29.63	34.04	38.74	16.10%
Australia	18.07	19.16	20.32	21.44	22.6	23.61	5.50%
Russia	12.12	14.65	17.36	19.23	20.57	21.64	12.30%
South Korea	14.4	15.64	16.84	17.67	18.46	19.15	5.90%

Retail is Changing

JD.COM

	Now	Future
Customization	Physical products	Product and services catered to specific needs
Access	Mobile or PC	Any time, any place
Integration	Products → Customers	Products ⇄ Customers



Challenges in Retail

Competition for customers' time

"Open-shelf" stores versus scenario / theme-based stores

Innovations in marketing: search/recommendation model versus social content based marketing

Creating retail scenarios

A View of the Future

JD.COM

E-Commerce Solutions: Boundaryless Retail

JD.COM

Shop any time, any place

Order products & services

Customize design, manufacturing, fulfillment, and marketing with user input

Understand customer needs with big data analytics

Connect customers directly to products and services seamlessly

Serve customer requirements with a shorter and more efficient supply chain

Data-Driven Boundaryless Retail

JD.COM

Application	Understand Customers	Connect To Customers	Serve Customers
Technology	IoT	Automations/Robotics	Block Chain	Cloud Computing
Algorithm	Operation Research, Optimization, Probability	Statistics , Machine Learning, Deep Learning	Simulation, A/B Testing Platform
Data	Customers	Products	Suppliers	Suppliers	Metadata	Feedback	Competitor

Retail-as-a-Service

- Retail-as-a-Service
- New opportunities and challenges for modeling/algorithm design and data analytics

Understand Customers	Connect To Customers	Serve Customers
• **Demand forecasting** • Customer behavior analysis • Detect customer needs	• **Product recommendation** • Personalized coupon via all channels • Real-time on-demand services	• **Supply chain management** • Fulfillment/Delivery network • Warehouse/store operations

Review historical path → **Explain state-of-the-art practice** → **Discuss future development**

Understand Customers

Presenter: Rong Yuan

Understand Customers with Big Data

Data

- Online transactions
- Click-Stream
- Customer feedback
- Offline shopping
- Webpage browsing
- Social media
- App usage

Big Data Analysis

Application

- **Demand Forecasting**
- **Behavior Description**
- **Intention Detection**

Focus of this tutorial

Time Series Forecasting

JD.COM

- Customer demand on a product forms a time series

Model-Driven	Data-Driven	Big Data Enrichment
Stochastic Time Series Models	**Machine Learning**	**Deep Learning**

Stochastic Time Series Models

- Linear Models:
  - ARIMA: Box-Jenkins methodology (1970)
  - AR,MA,ARMA,SARMA
  - VAR
- Non-linear Models:
  - ARCH (1982)
  - GARCH (1986)

Machine Learning

- Linear Regression
- Support Vector Regression (1996)
- Gaussian Process
- Tree-based Models
  - Regression Tress (1984)
  - Random Forest (1995)
  - Gradient Boosting
    - AdaBoost (1997)
    - XGBoost (2014)
    - LightGBM (2017)
    - CatBoost (2017)

Deep Learning

- MLP (<1965)
- RNN (1980s)
- LSTM (1997)
- Seq2seq (2014)

Forecasting Metrics

Metric	Formula	Strengths
MAD (mean absolute deviation)	$\frac{1}{N}\sum_i \lvert \hat{Y}_i - Y_i \rvert$	Most intuitive
MAPE (mean absolute percentage error)	$\frac{1}{N}\sum_i \left\lvert \frac{Y_i - \hat{Y}_i}{Y_i} \right\rvert$	Independent of the scale of measurement
SMAPE (symmetric mean absolute percentage error)	$\frac{1}{N}\sum_i \frac{2\lvert Y_i - \hat{Y}_i \rvert}{\lvert Y_i \rvert + \lvert \hat{Y}_i \rvert}$	Avoid asymmetry of MAPE
MSE (mean squared error)	$\frac{1}{N}\sum_i (Y_i - \hat{Y}_i)^2$	Panelize extreme errors
Quantile Loss	$\frac{1}{N}\sum_i q(Y_i - \hat{Y}_i)^+ + (1-q)(\hat{Y}_i - Y_i)^+$	Measure distribution

Other common measure -- Root Mean Squared Error (RMSE), Mean Forecast Error (MFE), Mean Percentage Error (MPE), Sum of Squared Error (SSE), Signed Mean Squared Error (SMSE), Normalized Mean Squared Error (NMSE), Mean Absolute Scaled Error (MASE), Overall Weighted Average (OWA)

Variable Customers Requirement

- Retailing is about getting the right products to the right people in the right place at the right time.
- Customers requirement vary by

Location
(e.g. stationery sales near a school)

Time
(e.g. ice-cream sales on sunny days)

Special Event
(e.g. toy sales after movie is released)

Personal Preference
(e.g. different fashion styles)

Demand Forecasting in E-Commerce

JD.COM

Highly variable customers needs

Stock inventory to provide buffer against demand variability

Millions of products (not to mention product-region pairs)

Supply chain issue like vendor lead time

Implications to Modelling

Highly variable customers needs	Highly non-stationary demand time series	Stock inventory	Probabilistic forecast
Millions of products	Multiple time series	Vendor lead time	Multi-horizon forecast

Forecasting Methods to be Covered

- Stochastic time-series models
  - ARIMA
- Machine learning
  - Tree-based
- Deep learning
  - Seq2Seq

	Scorecard
Highly non-stationary	-
Multiple time series	-
Multi-horizon forecast	-
Probabilistic forecast	-

ARIMA

- Auto-Regressive Integrated Moving Average
- George Box and Gwilym Jenkins developed in 1970s
- ARIMA(p,d,q)

$$y_t = \delta + \underbrace{\phi_1 y_{t-1} + \phi_2 y_{t-2} + \cdots + \phi_p y_{t-p}}_{\text{AR(p) terms regress against past values}} + \underbrace{\theta_1 e_{t-1} + \theta_2 e_{t-2} + \cdots + \theta_q e_{t-q}}_{\text{MA(q) terms regress against past errors}}$$

- ARMA models can only be used for stationary time series
- Use finite differencing to 'stationarize' time series

$$y_t{}' = y_t - y_{t-d}$$

Level of differencing

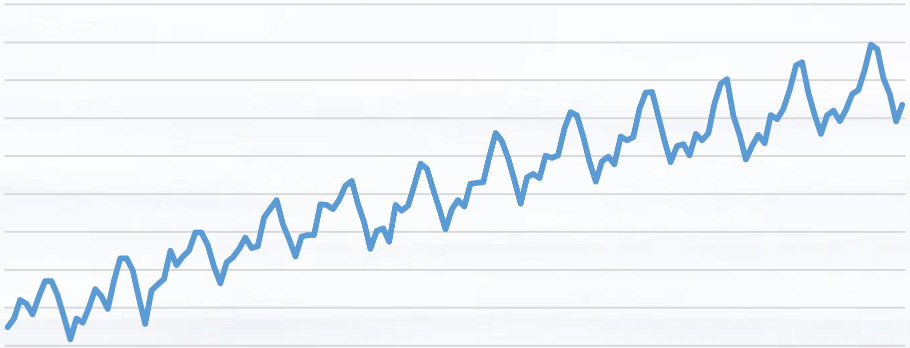
$$y_t' = \delta + \phi_1 y_{t-1}' + \phi_2 y_{t-2}' + \cdots + \phi_p y_{t-p}' + \theta_1 e_{t-1}' + \theta_2 e_{t-2}' + \cdots + \theta_q e_{t-q}'$$

ARIMA Example – Airline Passenger Dataset

Original Time Series

Time series with trend, seasonality, and non-constant variance

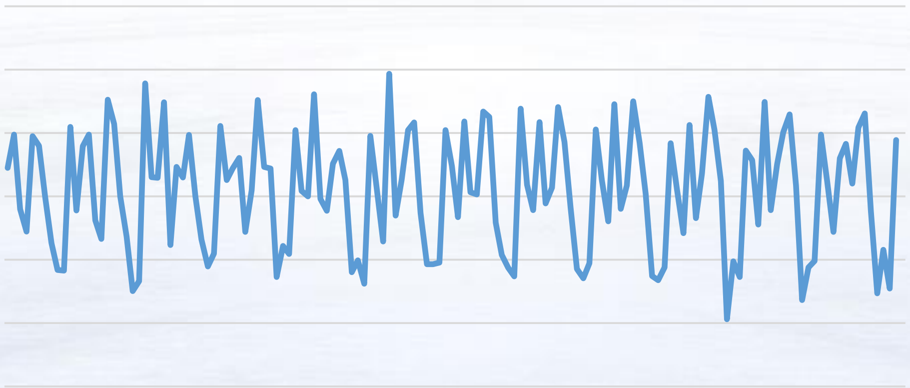
Take $log(y)$ to remove non-constant variance



Time series with trend and seasonality

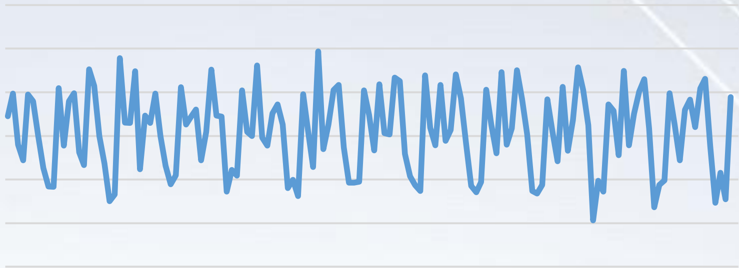
Differencing to remove trend
Level of differencing = 1

$y_t{}' = y_t - y_{t-1}$



Stationary time series with seasonality

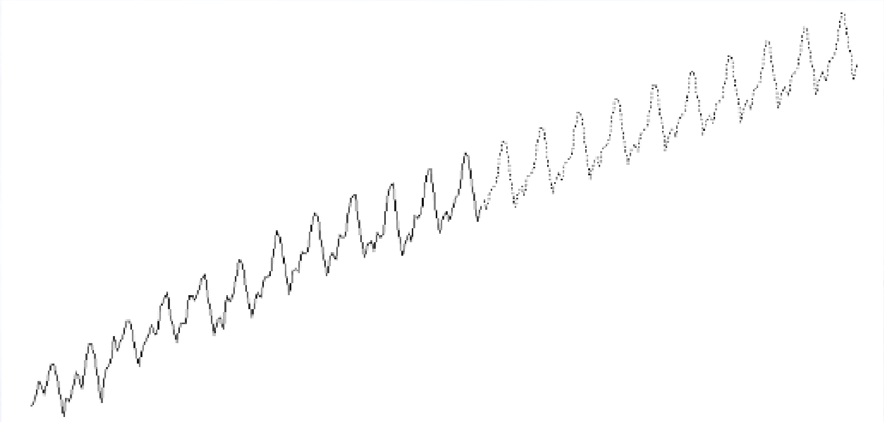
ARIMA Example – Fit Model



- Study autocorrelation and partial autocorrelation (ACF/PACF) charts to determine
  - Seasonal pattern: observe strong correlation between y_t' and y_{t-12}'
  - AR parameters: no strong correlation between y_t' and other y's
  - MA parameters: error terms at t – 1 and t – 12 are useful
  - ARIMA(0,1,1)x(0,1,1)^{12}
  - Automated in R/Python

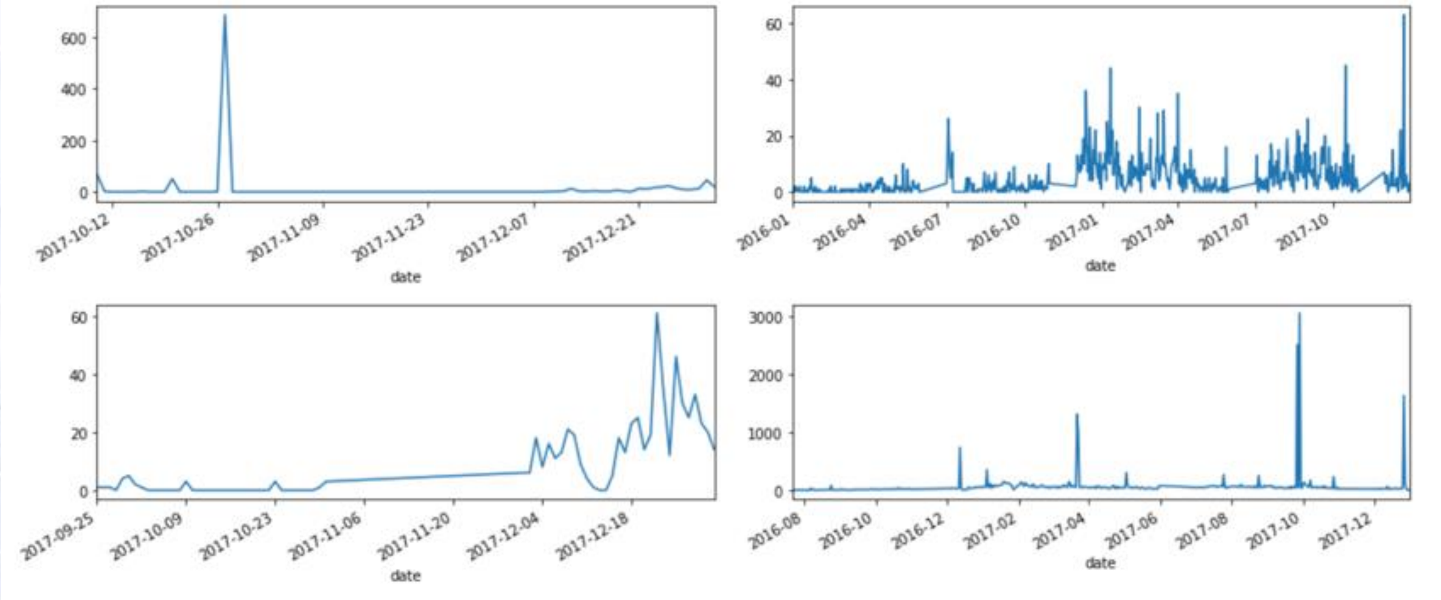
Expanded formula

$y_t = y_{t-12} + y_{t-1} - y_{t-13} - \theta_1 e_{t-1} - \theta_{12} e_{t-12} + \theta_{13} e_{t-13}$



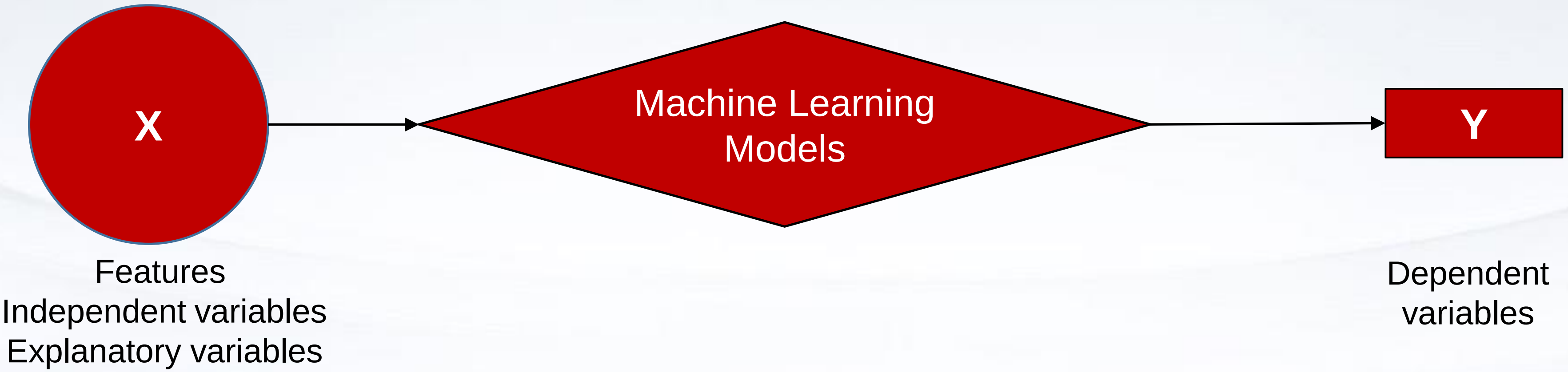
Limitations of ARIMA

- ARIMA assumes the underlying time series is linear
- Difficult to fit highly non-stationary time series
- Cannot deal with multiple time series at the same time
- What does demand look like in real business?



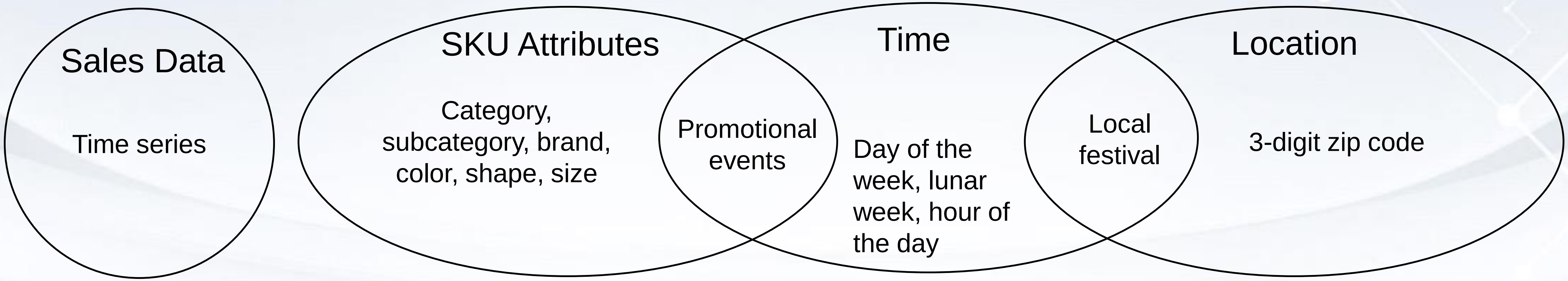
	Scorecard
Highly non-stationary	Limited
Multiple time series	Limited
Multi-horizon forecast	Yes
Probabilistic forecast	Yes

Machine Learning Model



- Flexible in having more features (X) in the model
- No assumption w.r.t the demand distribution
- One model for all time series
- Feature engineering is important

Feature Engineering



Human Expert | **One Hot Encoding** | **Feature Hashing** | **Embedding**

Features

<u>Example features</u>

Mean of the past 7-day sales
Variance of the past 7-day sales
Max sales of the past 14 days
Sales of the 7th day in the past
90% sales quantile of last month

<u>Example features</u>

Festival encoding ([0,0,0,1,0,0])
Percentage of discount
Promotional type (hash id)
Category (hash id)
SKU Name (embedding vector)

Popular Machine Learning Models

Linear Regression (ARIMA + X)

- Estimate independent variable as a linear expression of the dependent variables

Tree-based models

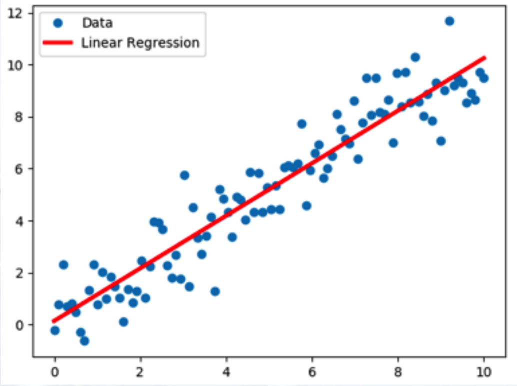
- Use decision trees to classify the dependent variables in order to make prediction

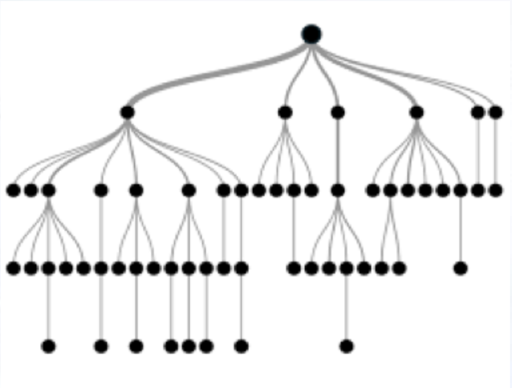
Support Vector Regression

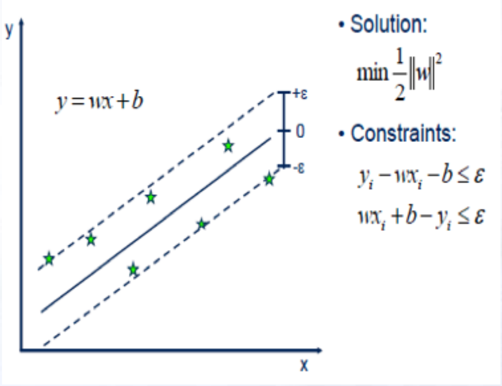
- A hyperplane is selected to separate the dependent variables
- Use a subset of the training data to draw a margin of tolerance around the hyperplane
- Use kernel function to model nonlinear relationships

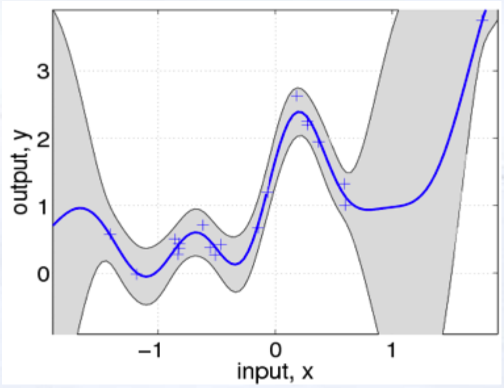
Gaussian Processes

- Assume the covariance between dependent variables is multivariant Gaussian
- Use kernel function to explore the relationship between the variables close to each other









Popular Machine Learning Models

JD.COM

Linear Regression (ARIMA + X)

- Estimate independent variable as a linear expression of the dependent variables

Tree-based models

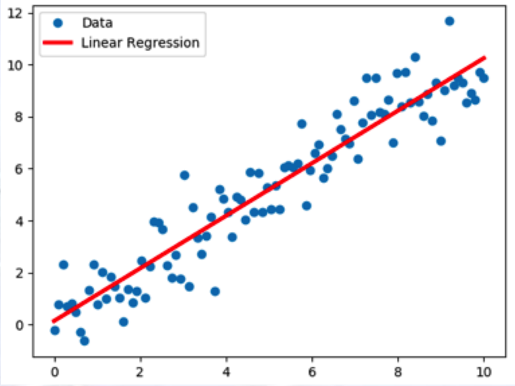
- Use decision trees to classify the dependent variables in order to make prediction

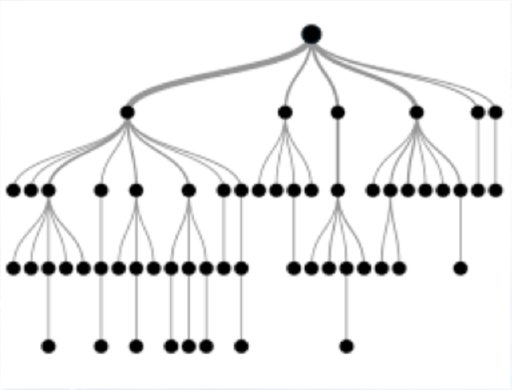
Support Vector Regression

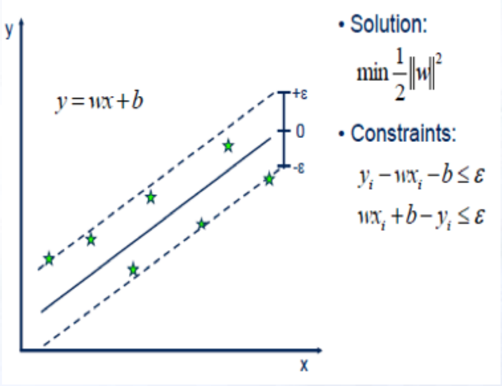
- A hyperplane is selected to separate the dependent variables
- Use a subset of the training data to draw a margin of tolerance around the hyperplane
- Use kernel function to model nonlinear relationships

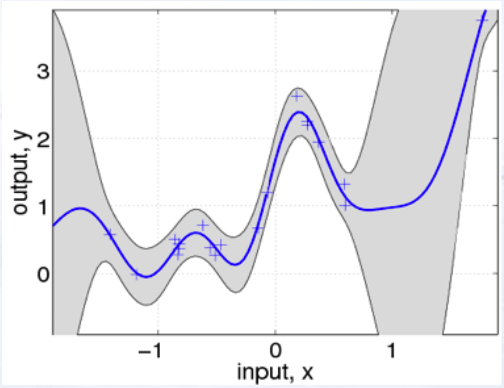
Gaussian Processes

- Assume the covariance between dependent variables is multivariant Gaussian
- Use kernel function to explore the relationship between the variables close to each other



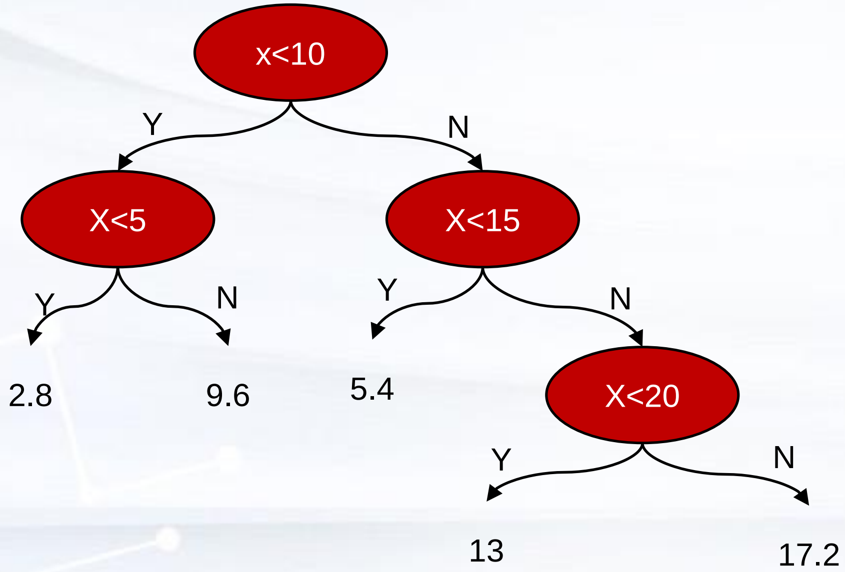


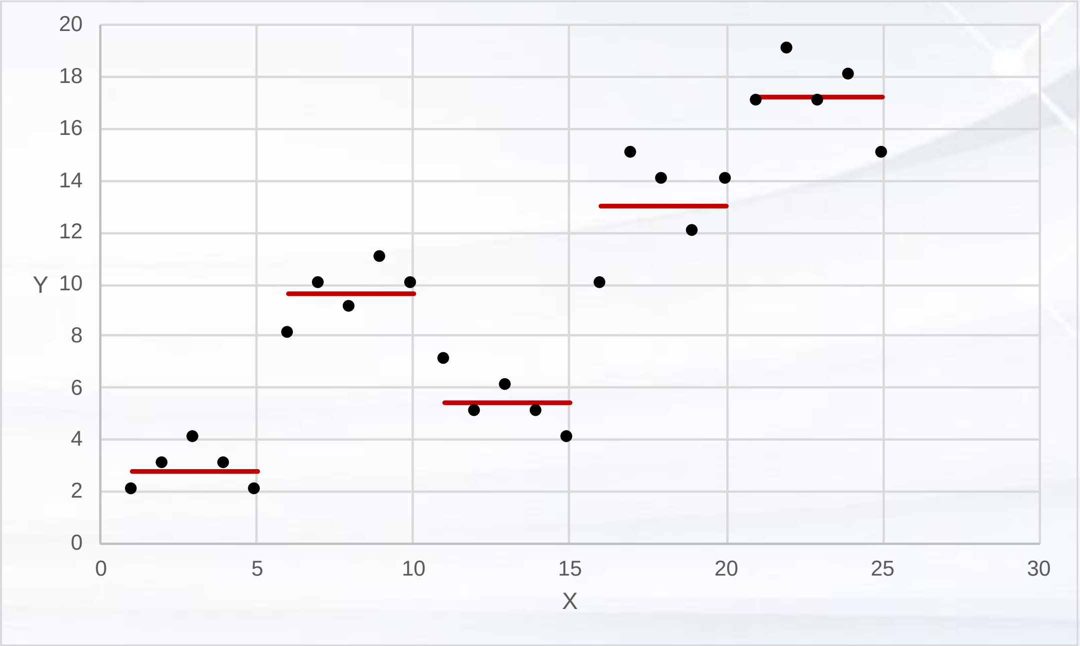




JD.COM

- Regression tree





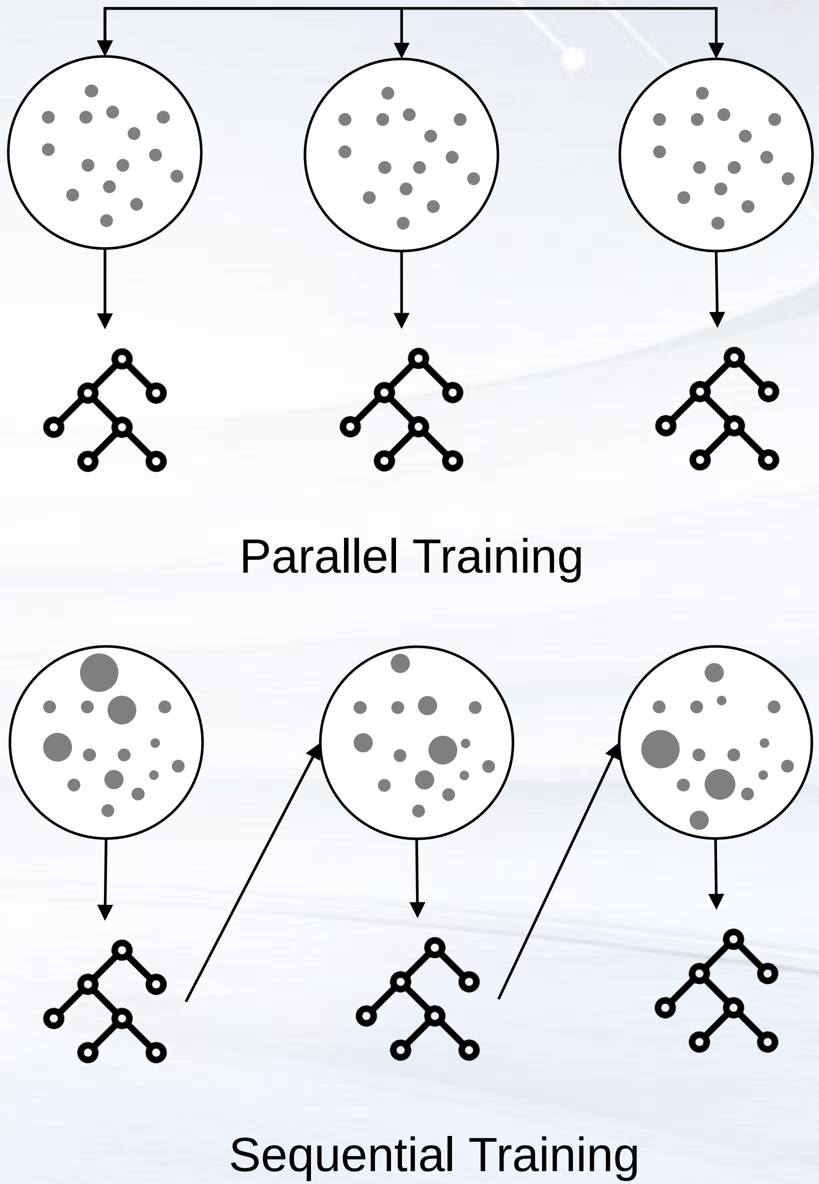
- Top-down approach – choose a variable at each step to best split the sample set;

Tree-Based Models

JD.COM

- Random Forest
  - Bagging
  - Independent classifiers
  - Random resample with replacement
  - Random feature selection at split

- Gradient Boosting
  - Boosting
  - Sequential classifiers
  - Resample with weights

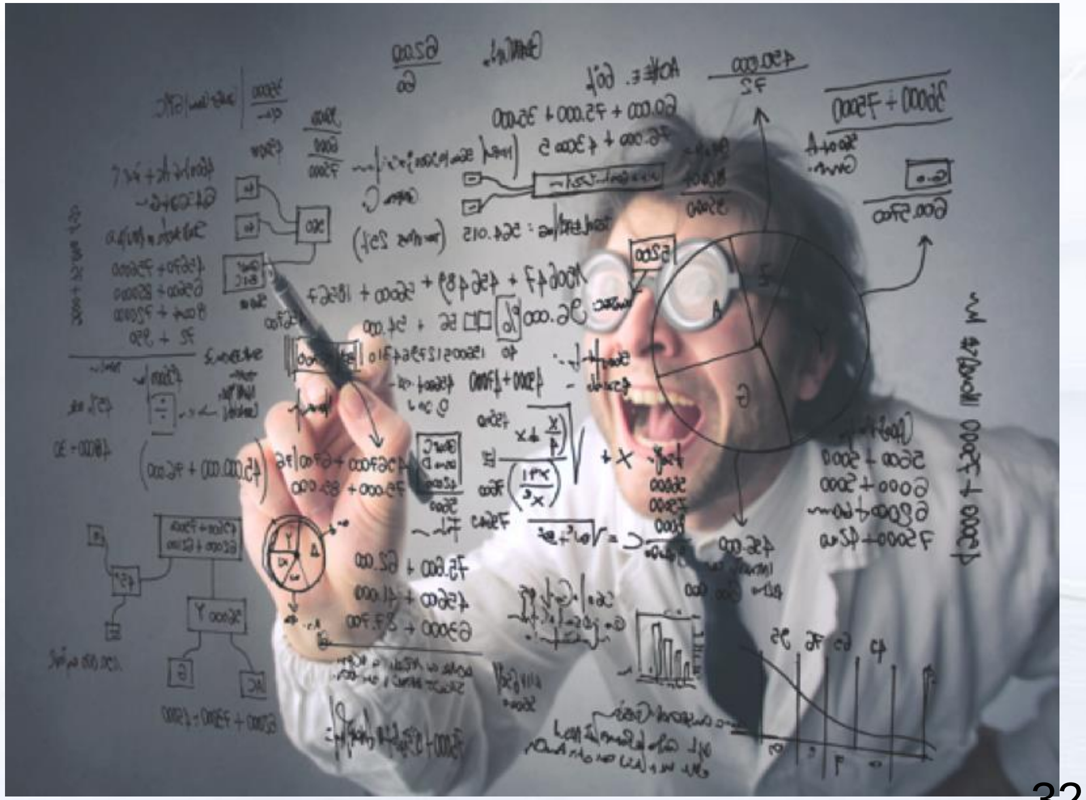


Limitations of Machine Learning

JD.COM

- Incorporating features requires manual work
  - Requires human expertise
  - Some features are difficult to capture
  - Time consuming

	Scorecard
Highly non-stationary	Yes
Multiple time series	Yes
Multi-horizon forecast	Yes
Probabilistic forecast	Yes



- Goal
  - Quick review of the deep learning techniques that empowers powerful applications in time series forecasting
- Content
  - Neuron
  - Multi-layer perceptron
  - RNN
  - LSTM
  - Seq2Seq

Neuron

Input vector:

$x = (x_1, x_2, \dots, x_n)$

Neuron j

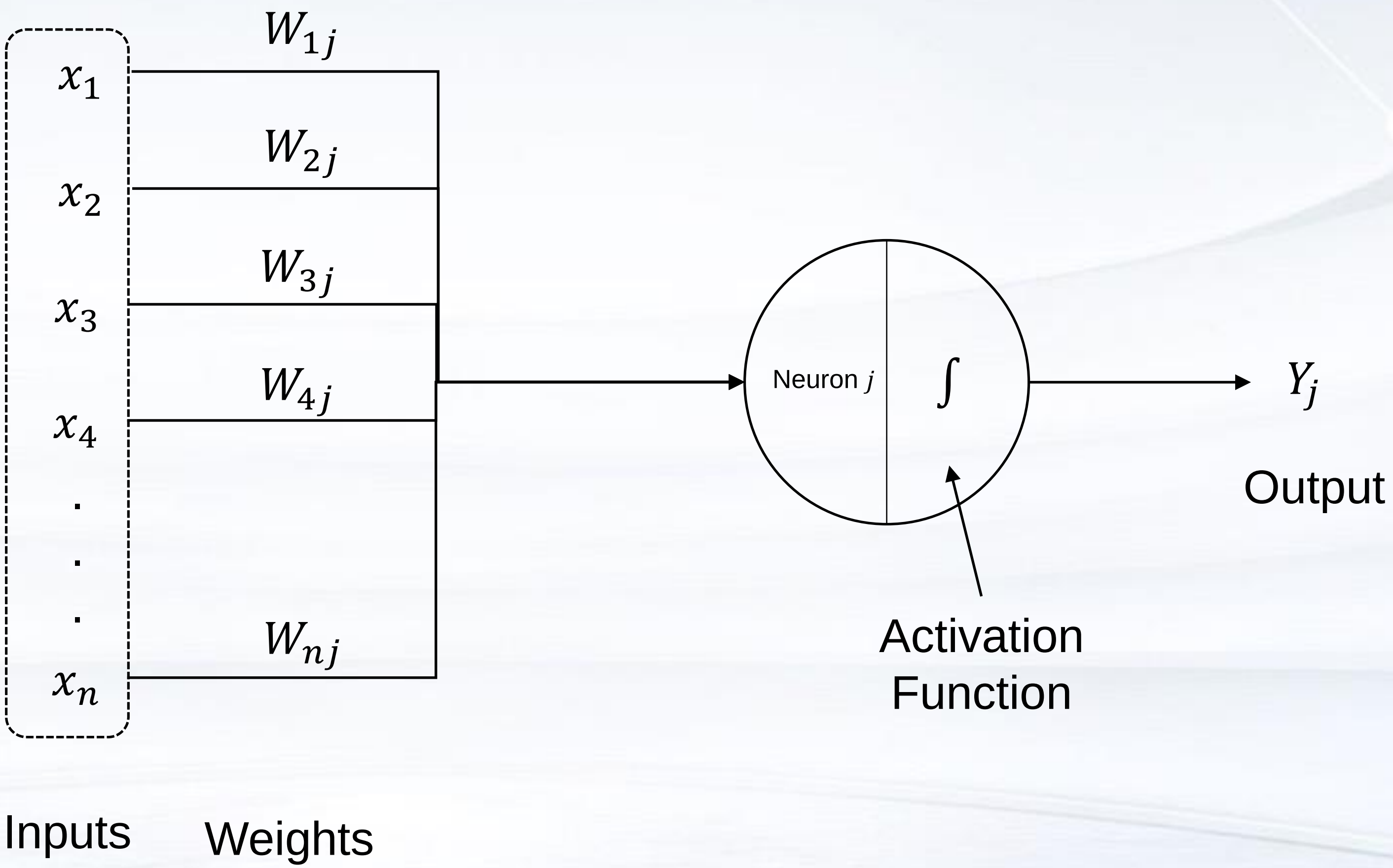
Weight vector: W_j

Bias: b_j

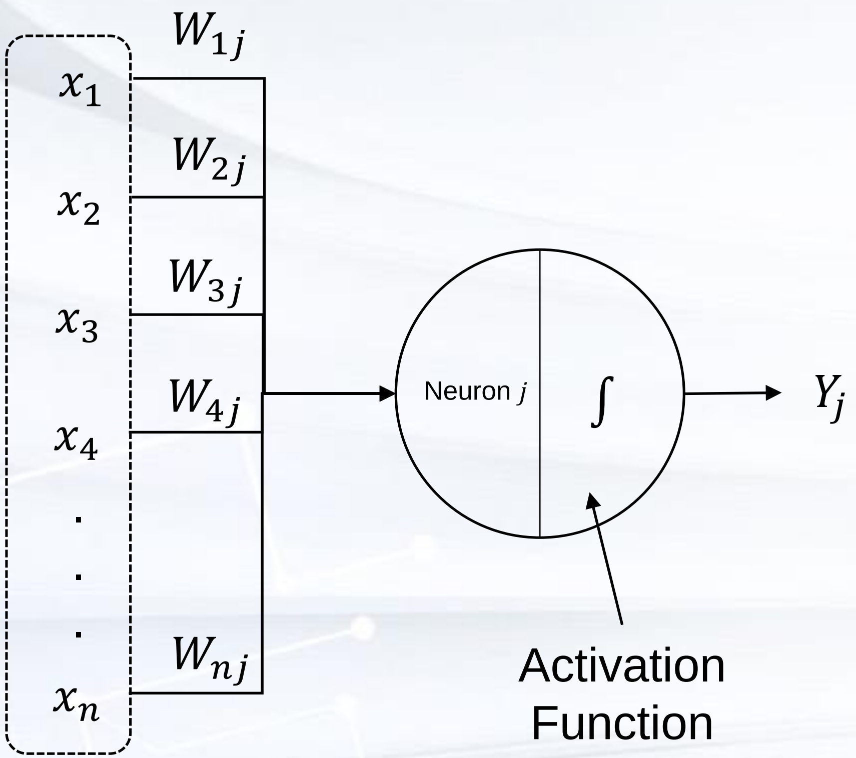
Activation function: f

Output:

$Y_j = f(W_j^T x + b_j)$

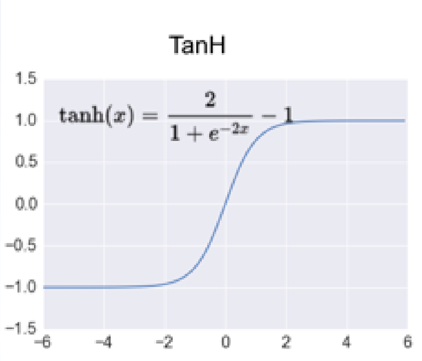


Type of Activation Functions



Tanh: $\mathbb{R} \rightarrow (-1,1)$

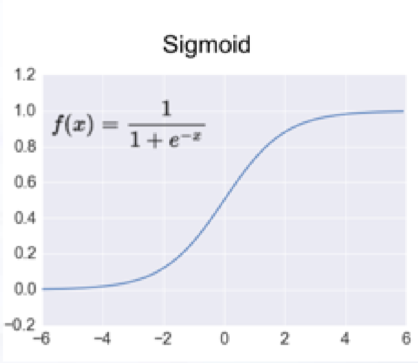
$$tanh(\mathrm{x}) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$$



Maintain negativity; model two-class classification

Sigmoid: $\mathbb{R} \rightarrow (0,1)$

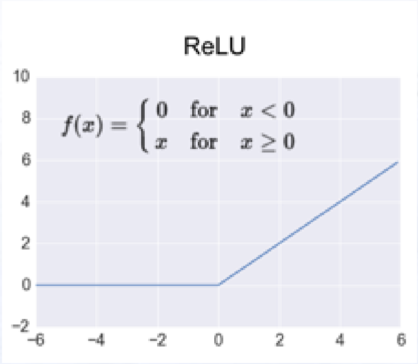
$$\sigma(x) = \frac{1}{1 + e^{-x}}$$



Predict probability; model multi-class classification

ReLU: $\mathbb{R} \rightarrow [0, \infty)$

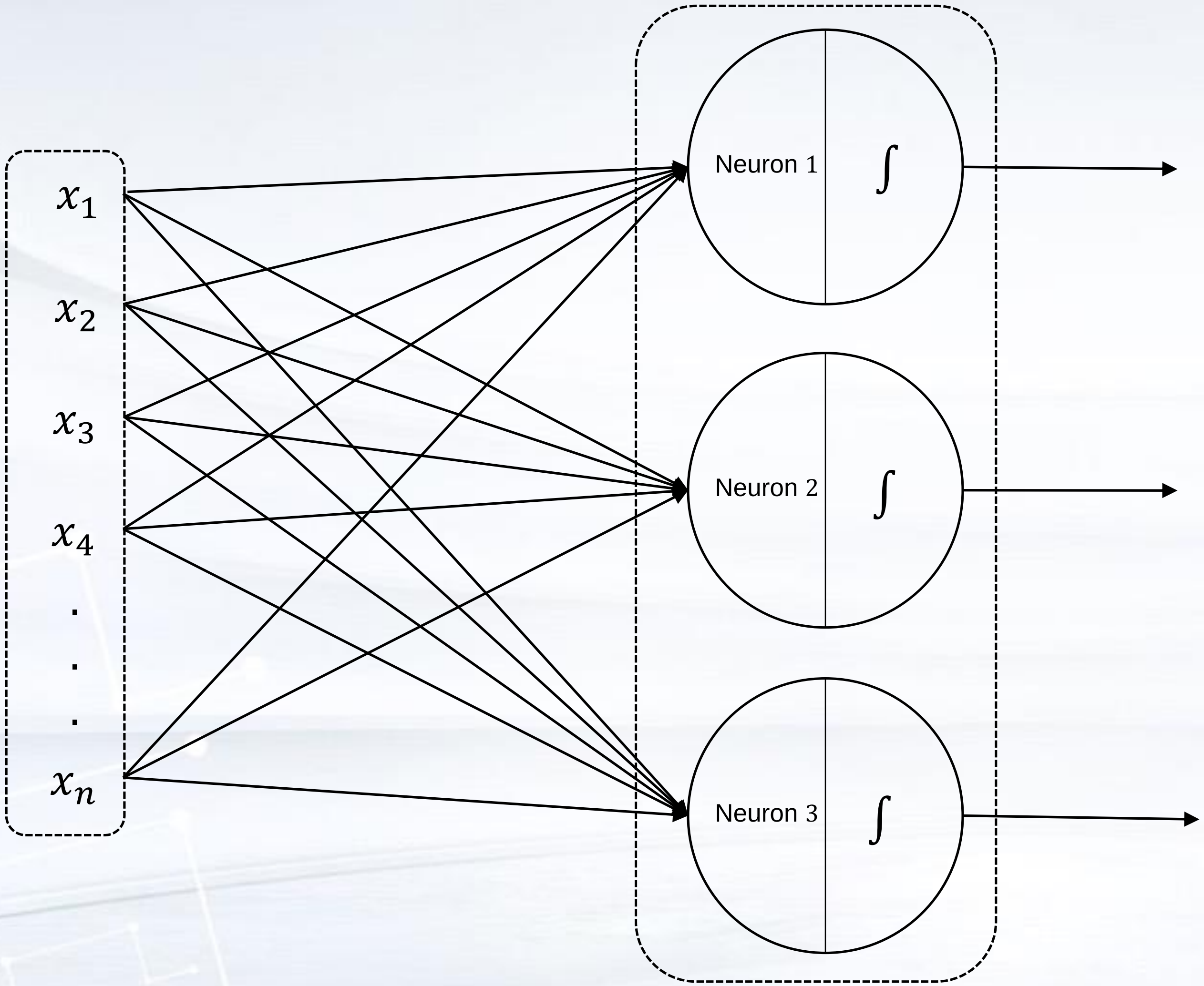
$$ReLU(x) = \max(0, x)$$



Only keep positive values;
gradient is constantly;
more computationally efficient

Single Layer

JD.COM



$$Y_1 = f(W_1^T x + b_1)$$

$$Y_2 = f(W_2^T x + b_2)$$

$$Y_3 = f(W_3^T x + b_3)$$

$$Y = f(W^T x + b)$$

Matrix Notation

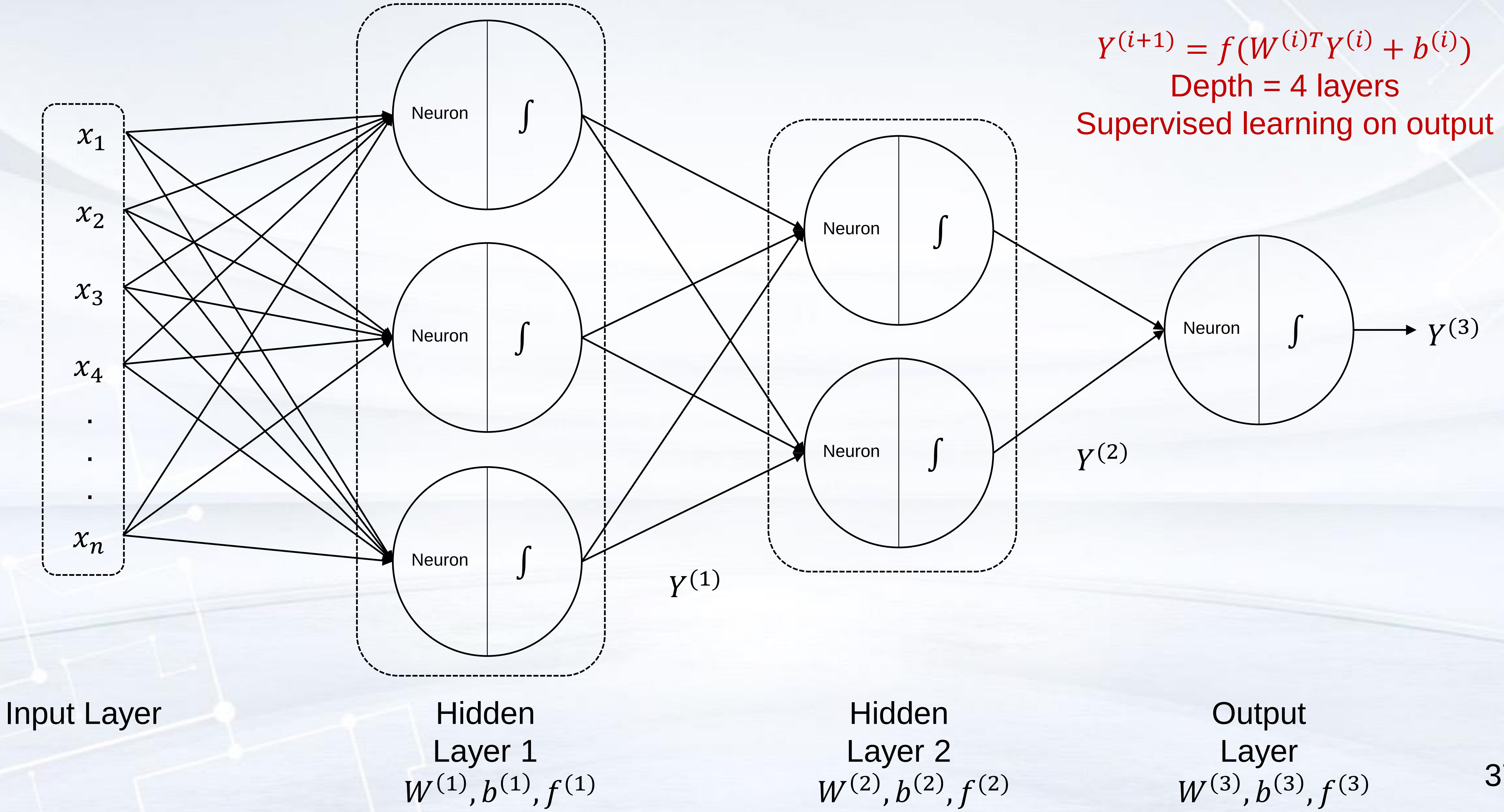
Inputs Layer Outputs

Multi-Layer Perceptron (MLP)

$$Y^{(i+1)} = f(W^{(i)T}Y^{(i)} + b^{(i)})$$

Depth = 4 layers

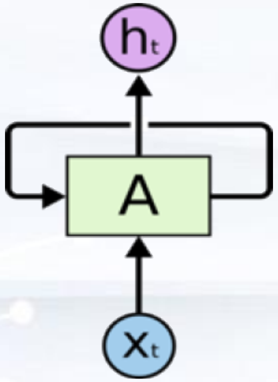
Supervised learning on output



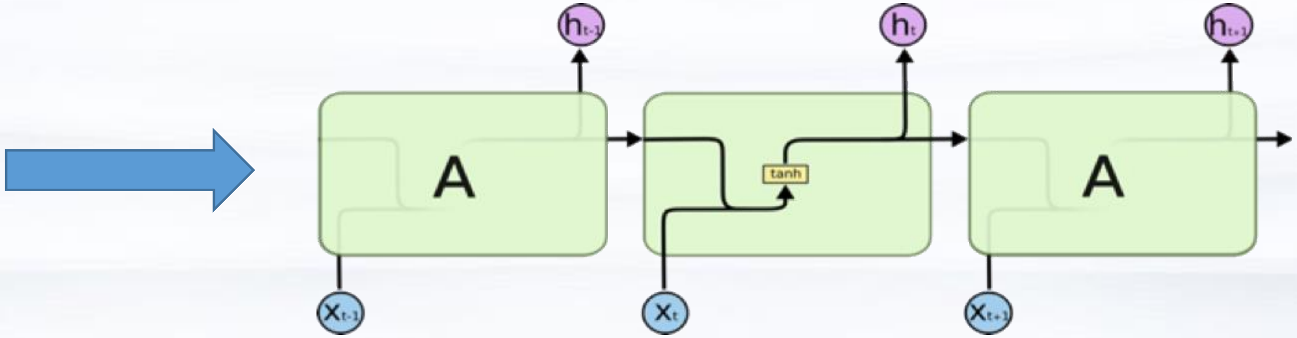
Recurrent Neural Network (RNN)

JD.COM

- Pass on information through feedback loop
- Parameter sharing across time indices



RNN



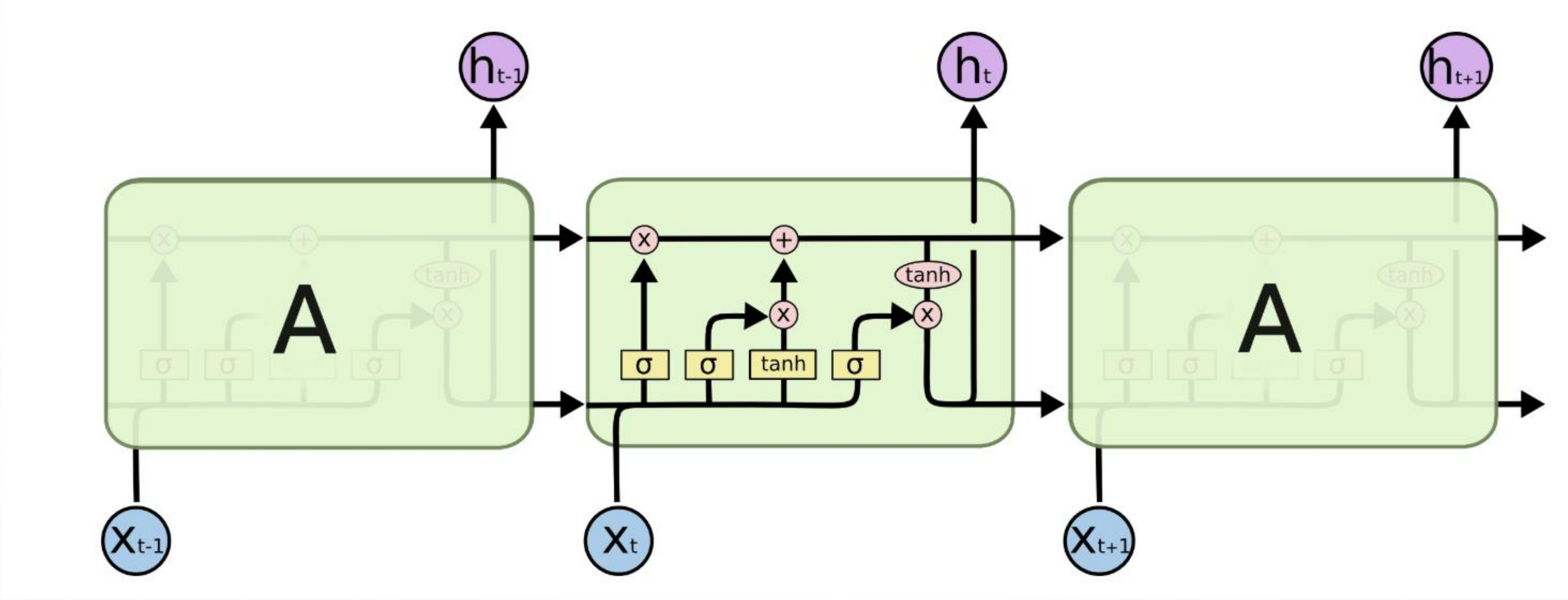
RNN Unfolded

$$h_t = \sigma_h(W_h x_t + U_h h_{t-1} + b_h)$$

Picture credits to colah.github.io

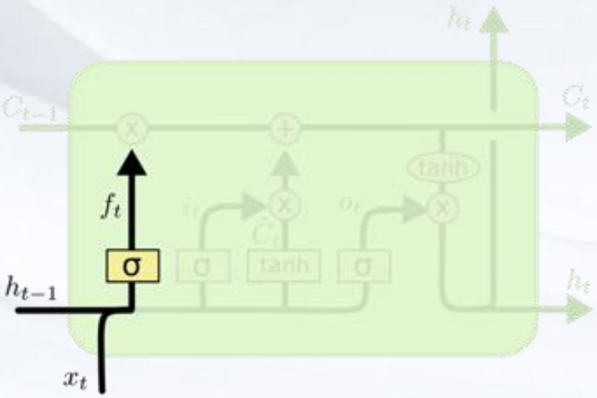
Long Short Term Memory (LSTM)

- Long term memory via cell state
- Cell state updates regulated by gates
- Applications : speech recognition, language modeling, translation, image captioning



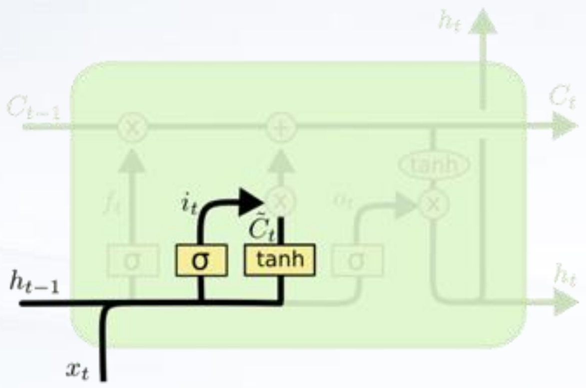
Picture credits to colah.github.io

Long Short Term Memory (LSTM)



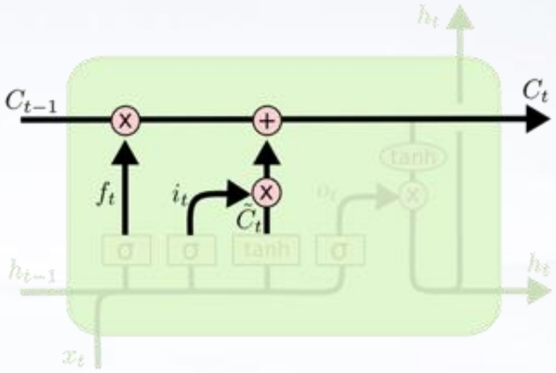
$$f_t = \sigma\left(W_f \cdot [h_{t-1}, x_t] \ + \ b_f\right)$$

Forget gate: when we see new input x_t, we want to forget "memorized" info in cell state C.



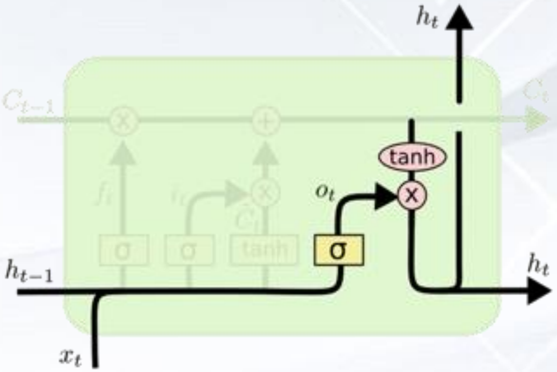
$$i_t = \sigma\left(W_i \cdot [h_{t-1}, x_t] \ + \ b_i\right)$$
$$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] \ + \ b_C)$$

Input gate: add new info to the cell state C.



$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t$$

Combine 'forget' and 'update' info to the cell state C

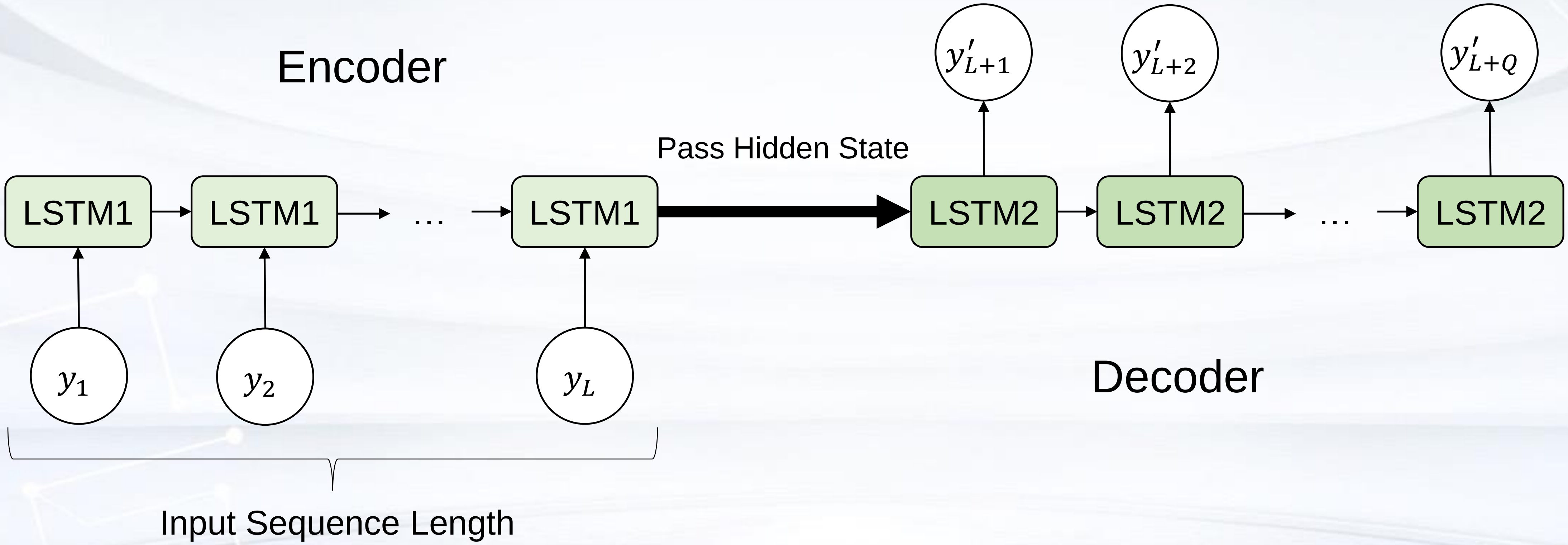


$$o_t = \sigma\left(W_o \ [h_{t-1}, x_t] \ + \ b_o\right)$$
$$h_t = o_t * \tanh\left(C_t\right)$$

Output gate: decide what we are going to output. Output will be based on current cell state C, and direct input x_t.

Picture credits to colah.github.io

Seq2Seq Model for Time Series



Related literature on e-commerce demand forecast with Seq2Seq model: DeepAR (Flunkert et al., 2017), MQ-RNN (Wen et al., 2017)

Methods Comparison

- Stochastic time-series models
  - Good model interpretability
  - Limited model complexity to handle non-linearity
  - Difficult to incorporate cross features among multiple time series
- Machine learning
  - Flexible and can incorporate any feature explicitly
  - Heavy workload in terms of feature engineering
- Deep learning
  - Very flexible and automated feature detection
  - Poor model interpretability

	Stochastic Time Series	Machine Learning	Deep Learning
Highly non-stationary	Limited	Yes	Yes
Multiple time series	Limited	Yes	Yes
Multi-horizon forecast	Yes	Yes	Yes
Probabilistic forecast	Yes	Yes	Yes

JD Practice

JD.COM

- Data
- Problem definition
- Model/Methodology
- Some highlights
- Numerical results

JD Practice : Data

JD.COM

SKU Attributes Table

item_sku_id
item_first_cate
item_second_cate
item_third_cate
brand_code
attr_cd
attr_value_cd

Historical Sales Table

item_sku_id
dc_id
date
quantity
vendibility

Promotional Events Table

item_sku_id
date
promotion_type

Attribute COLOR can have multiple values such as RED, GREEN, BLUE, etc.

Vendibility measures if on-hand inventory is positive at the end of the day

Promotion type can be direct sales, group rate discount, gift, etc.

Example data (1000 SKUs over 2 years) available on JD Global Optimization Challenge (GOC)

JD Practice: Probabilistic + Multi-Horizon Forecast Output

JD.COM

	1-day	2-day	3-day	…	31-day
80%	-	-	-	-	-
81%	-	-	-	-	-
82%	-	-	-	-	-
…	-	-	-	-	-
97%	-	-	-	-	-
98%	-	-	-	-	-
99%	-	-	-	-	-

Predict this table for each SKU every day

JD Practice: Model Framework

JD.COM

Data Analysis
- Data Collection
- Exploratory Data Analysis

Data Preprocessing
- Data Transformation
- Filling Missing Data

Feature Engineering
- Features Identification
- Embedding on Categorical Data

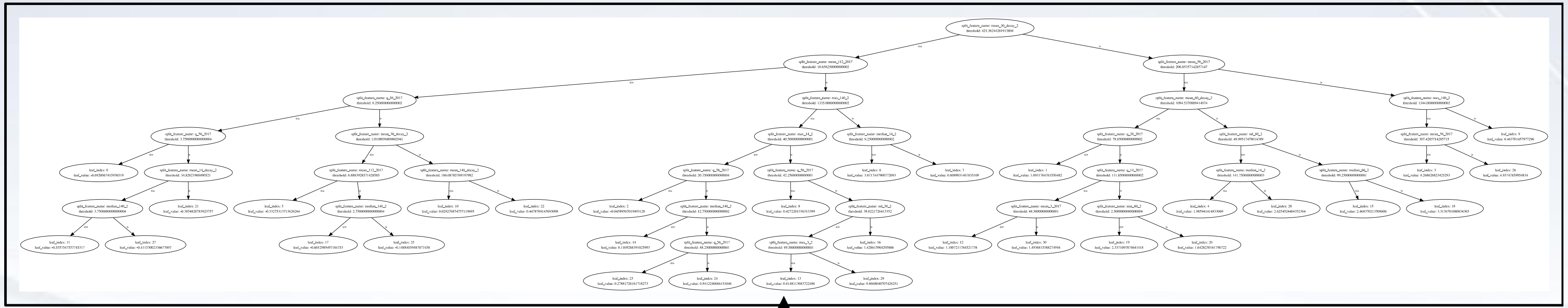
Training Mechanism
- Sample Selection
- Augmentation
- Cross Validation

Model Evaluation
- Parameter Tuning
- Adjustment for Overfitting
- Model Selection

Model

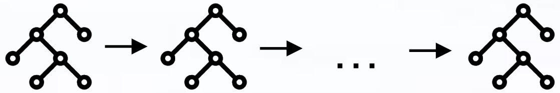
Gradient Boosting Machine Learning
Seq2Seq Deeping Learning

JD Practice : Gradient Boosting Machine Learning Model



Data processing and feature engineering

Gradient Boosting

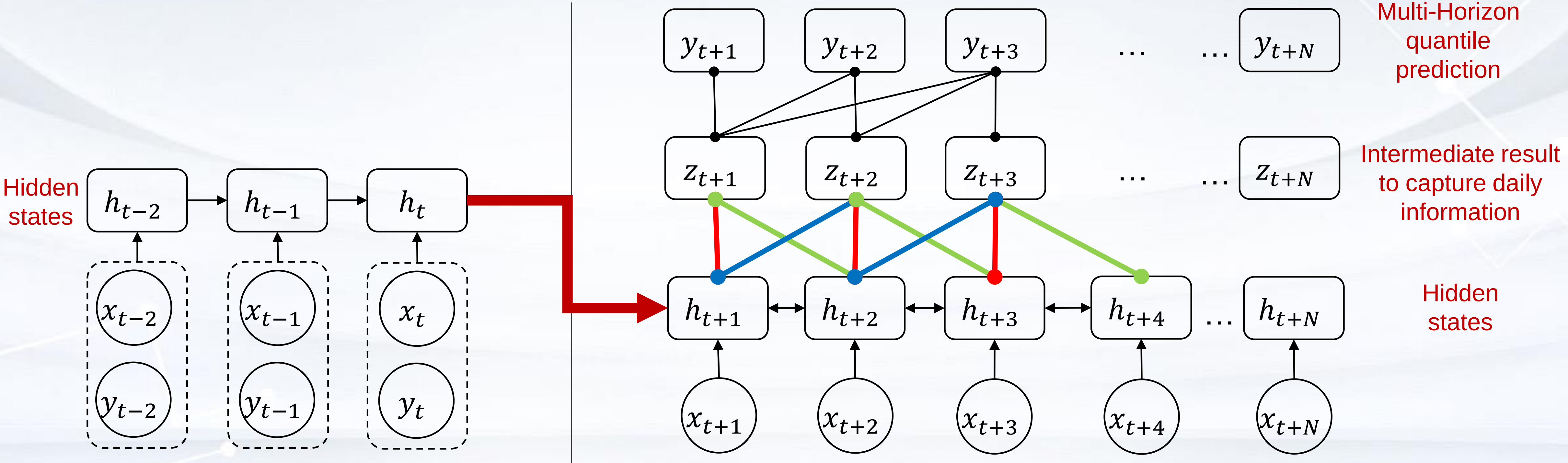


- LightGBM
- Sequential classifier
- **Parameter tuning for better accuracy**
  - Number of leaves
  - Num_iterations
  - Learning rate
- **Regularization/over-fitting**
  - Lambda_l1/Lambda_l2
  - Max depth
  - Min_data_in_leaves

Prediction

JD Practice : Seq2Seq Deep Learning Model

JD.COM



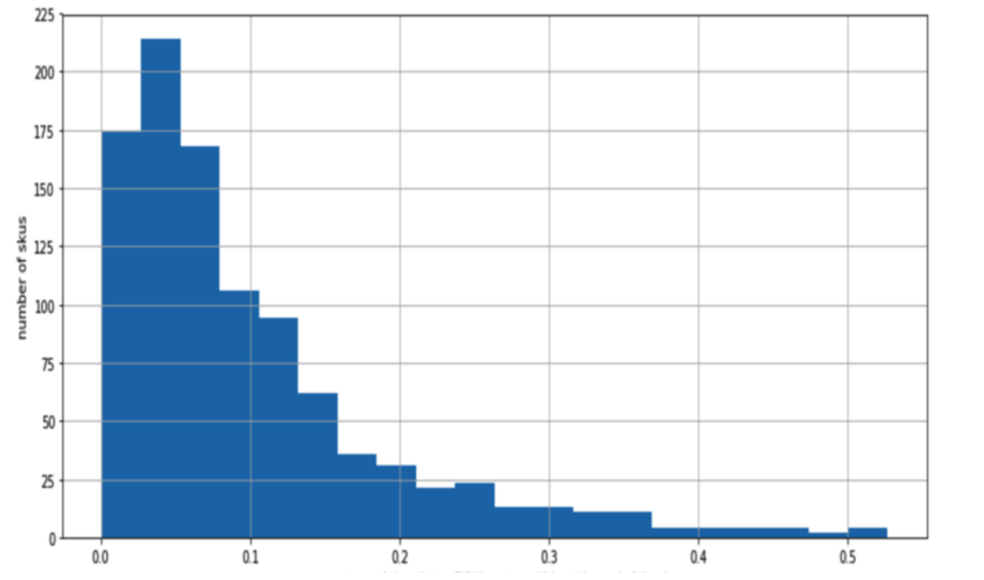
LSTM Encoder for Historical Sales

Decoder - global MLP applied at every local sliding window, capturing both global and local information.

JD Practice : Exploratory Data Analysis

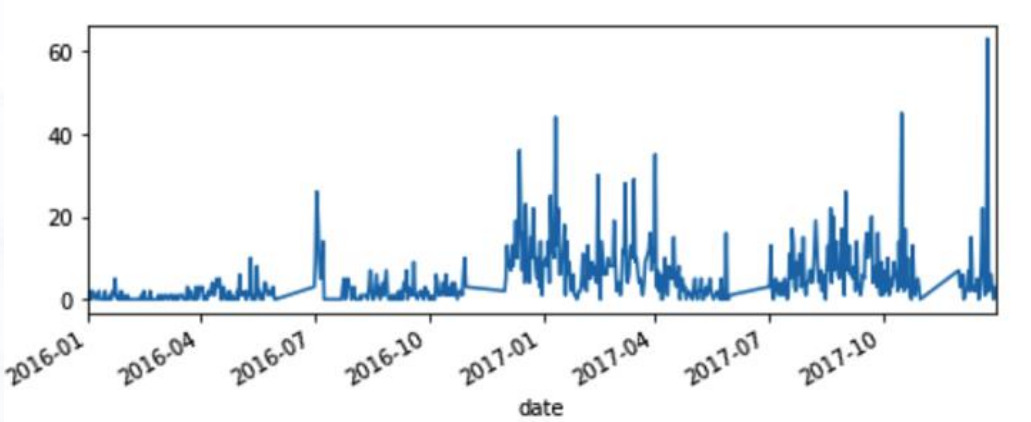
JD.COM

Censored Demand Observation



Percentage of time SKUs are out of stock

Both Seasonality and correlation in sales data

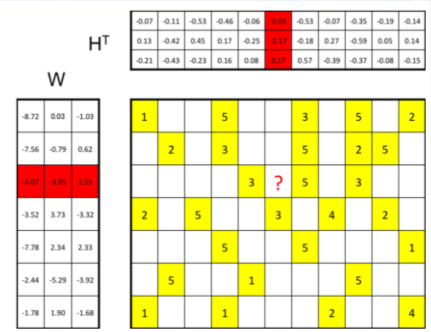


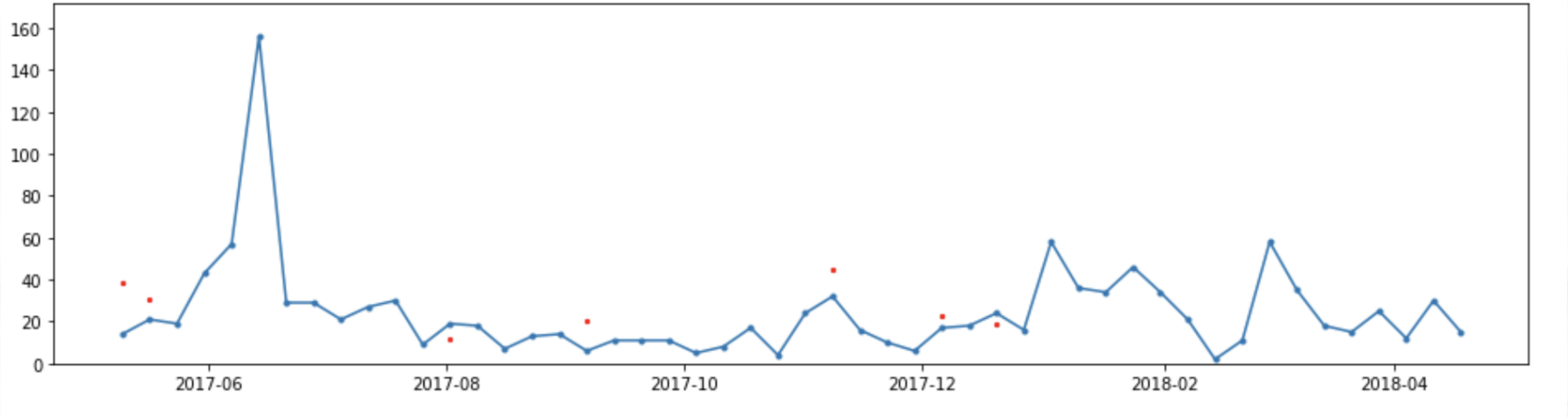
Data Analysis | Data Preprocessing | Feature Engineering | Training Mechanism | Model Evaluation

JD Practice : Filling Missing Data

JD.COM

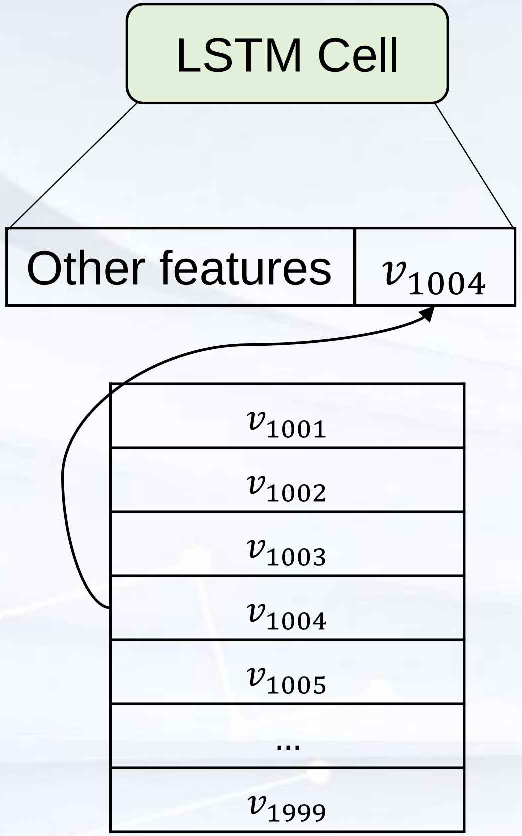
- Matrix Factorization
- Improve forecast accuracy by 5~15%



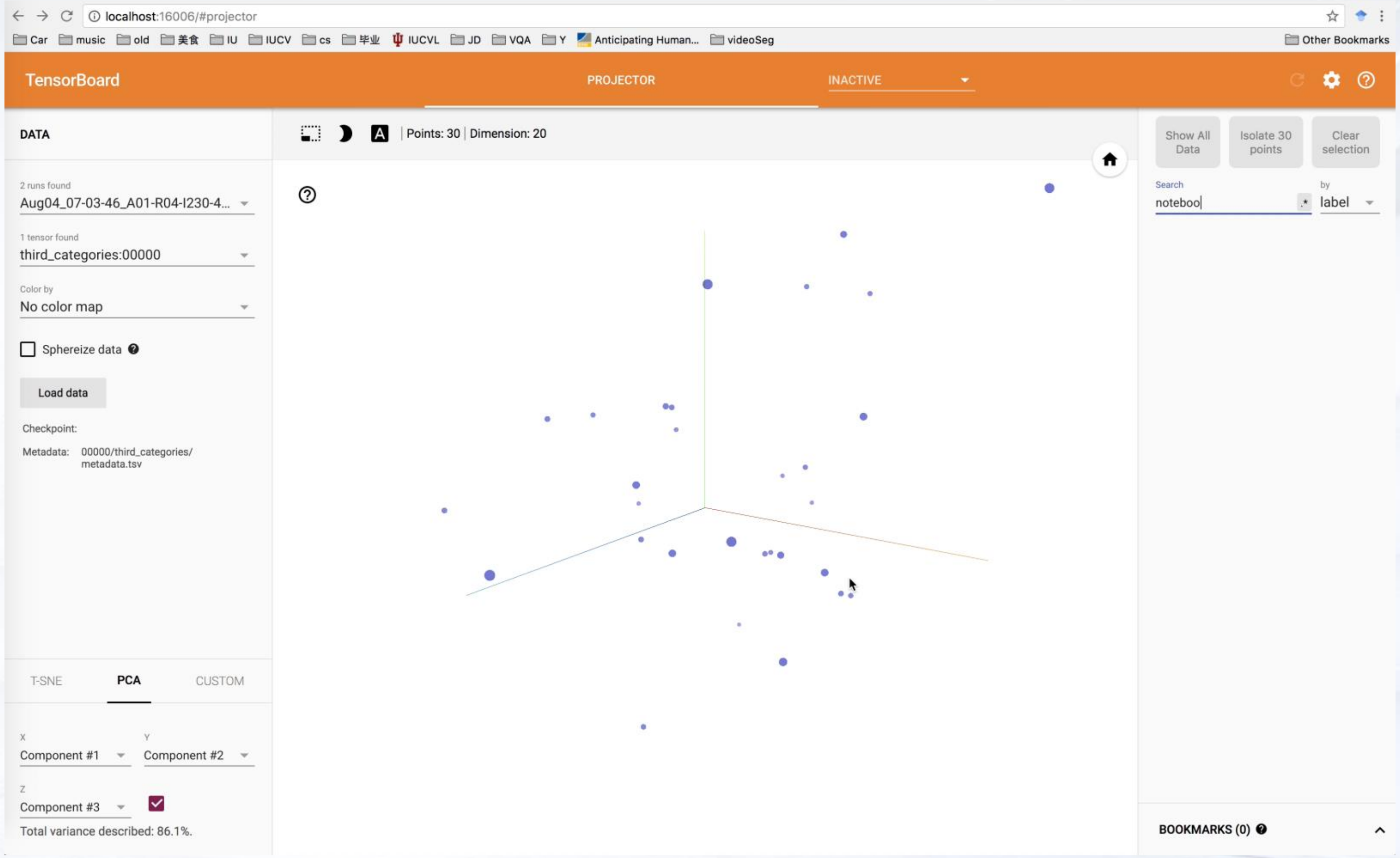


Data Analysis → **Data Preprocessing** → Feature Engineering → Training Mechanism → Model Evaluation

JD Practice : Embedding



Category Embedding Vectors



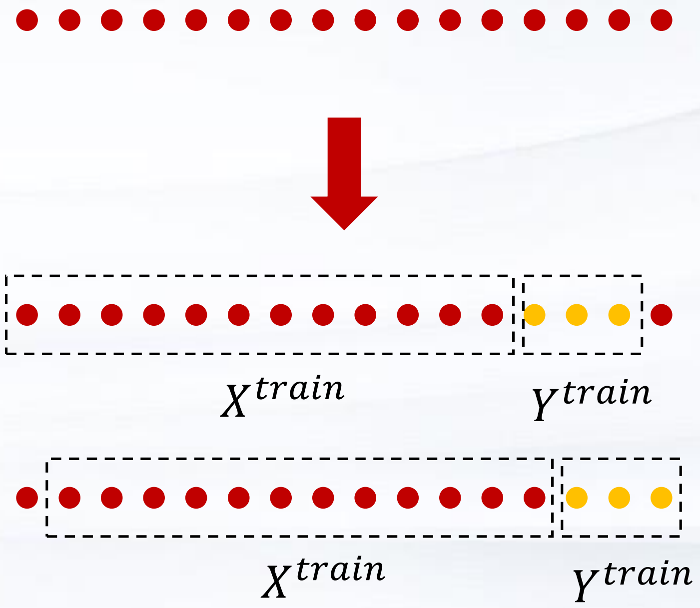
Data Analysis → Data Preprocessing → **Feature Engineering** → Training Mechanism → Model Evaluation

JD Practice : Sample Selection

In order to capture yearly seasonality…

In order to capture influence from near past…

Data Augmentation

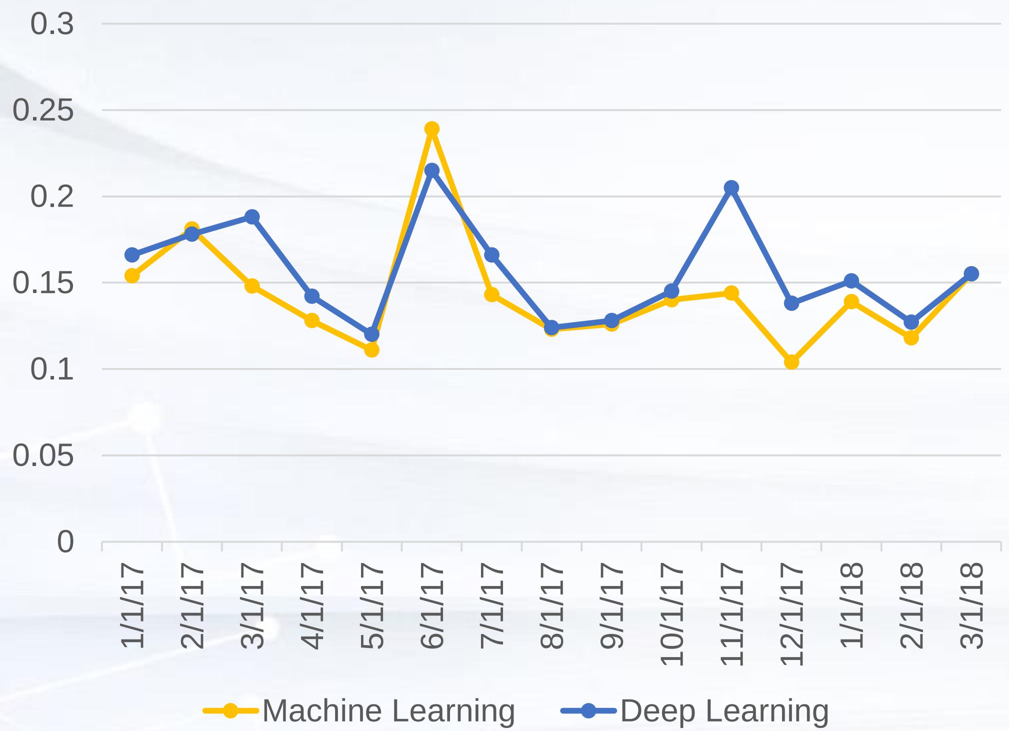


Data Analysis → Data Preprocessing → Feature Engineering → **Training Mechanism** → Model Evaluation

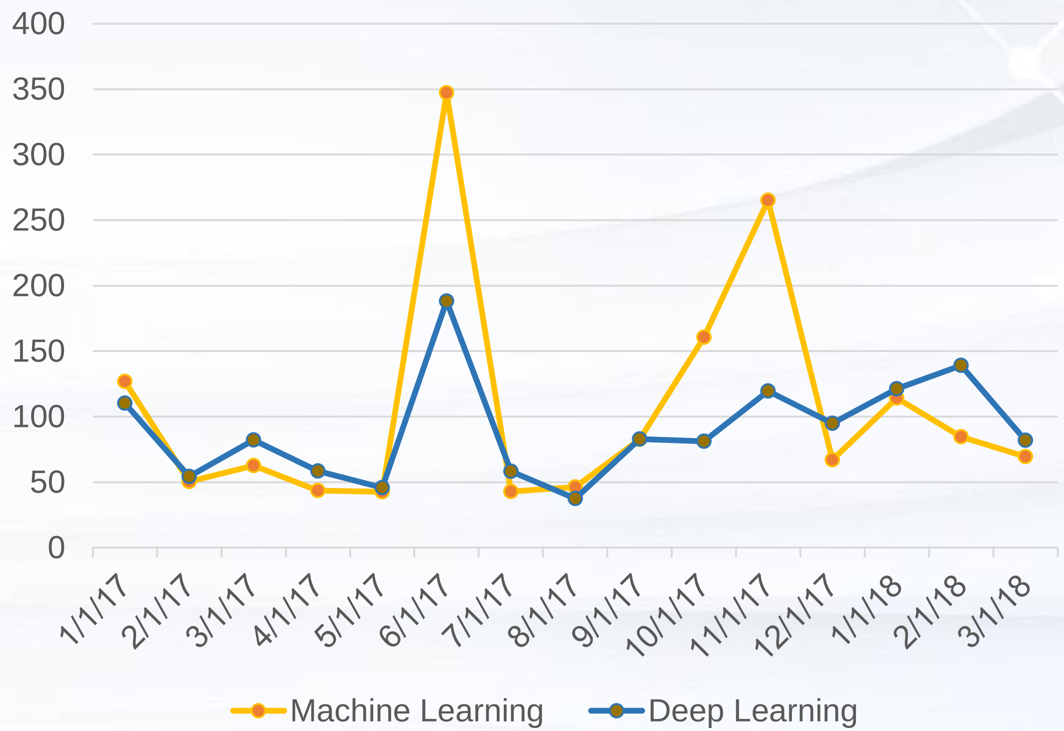
JD Practice : Numerical Result

JD.COM

95% quantile with metric $\frac{1}{N}\sum_i \frac{q(Y_i-\hat{Y}_i)^+ + (1-q)(\hat{Y}_i-Y_i)^+}{Y_i}$

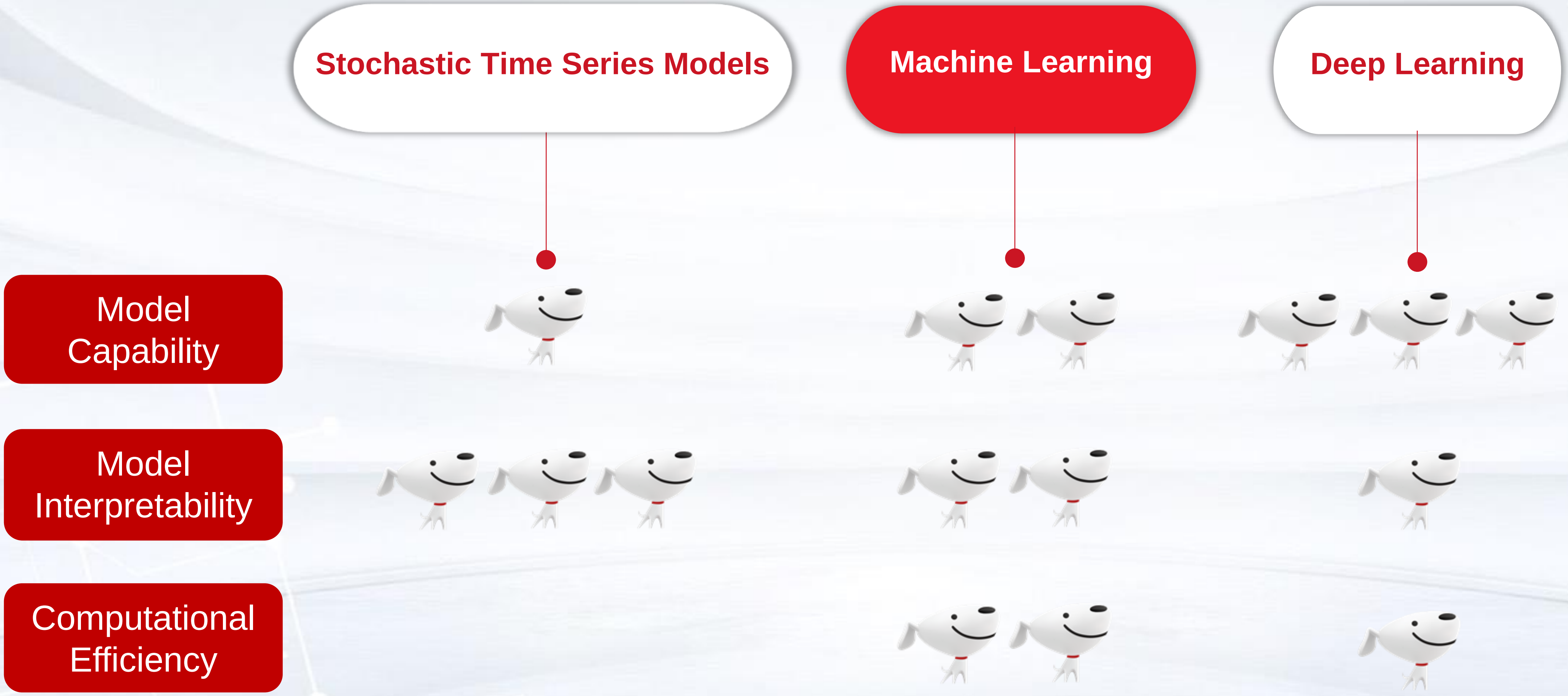


95% quantile with metric $\sum_i q(Y_i - \hat{Y}_i)^+ + (1-q)(\hat{Y}_i - Y_i)^+$



- Comparable results between ML and DL on 20000 products with historical sales over 2 years
- DL is in general better at handling sales peaks

Summary



Stochastic Time Series Models
Machine Learning
Deep Learning
Model Capability
Model Interpretability
Computational Efficiency

Reference

•Adhikari, R., & R.K., A. (2013). An Introductory Study on Time Series Modeling and Forecasting. LAP Lambert Academic Publishing, 18.
•Dickey, D. A., & Fuller, W. A. (1979). Distribution of the Estimators for Autoregressive Time Series With a Unit Root. Journal of the American Statistical Association, 427-431.
•Engle, R. (2001). GARCH 101: The Use of ARCH/GARCH Models in Applied Econometrics. Journal of Economic Perspectives, 157-168.
•Ahmed, Nesreen K. , Atiya, Amir F. , Gayar, Neamat El and El-Shishiny, Hisham (2010) An Empirical Comparison of Machine Learning Models for Time Series Forecasting', Econometric Reviews, 29: 5, 594-621.
•Bishop, C. M. (1995). Neural Networks for Pattern Recognition. Oxford, UK: Oxford University Press.
•Moody, J. E., Darken, C. (1989). Fast learning in networks of locally-tuned processing units. Neural Computation 1:281–294.
•Powell, M. J. D. (1987). Radial basis functions for multivariable interpolation: In: Mason, J. C., Cox, M. G., eds. A Review in Algorithms for Approximation. Oxford: Clarendon, pp. 143–168.
•MacKay, D. J. C. (1992a). Bayesian interpolation. Neural Computation 4:415–447.
•MacKay, D. J. C. (1992b). A practical Bayesian framework for backpropagation networks. Neural Computation 4:448–472.
•Hardle, W. (1990). Applied Nonparametric Regression, Econometric Society Monographs, 19. Cambridge, UK: Cambridge University Press.
•Hastie, T. Tibshirani, R., Friedman, J. (2001). The Elements of Statistical Learning. Springer Series in Statistics. Springer-Verlag.
•Breiman, L. (1993). Classification and Regression Trees. Boca Raton, FL: Chapman & Hall.
•Scornet, E., Biau, G., Vert, J.P. (2015) Consistency of random forests. Ann. Stat., 43, 1716–1741.
•Biau, G., Scornet, E. (2016) A random forest guided tour. Test, 25, 197–227.
•Tyralis, H., Papacharalampous, G. (2017) Variable Selection in Time Series Forecasting Using Random Forests. Algorithms 2017, 10, 114; doi:10.3390/a10040114
•Dudek, G. (2015) Short-Term Load Forecasting using Random Forests. Intelligent Systems 2014: Proceedings of the 7th IEEE International Conference Intelligent vol 2: pp.821-828.
•Nedelleca, R., Cugliarib, J., Goudea, Y. (2012) GEFCom2012: Electric load forecasting and backcasting with semi-parametric models. International Journal of Forecasting 30 (2014) 375–381.
•Kusiak, A., Zheng, H., Song, Z. (2009) Short-Term Prediction of Wind Farm Power: A Data Mining Approach. IEEE Transactions On Energy Conversion, 24 (1)
•Scholkopf, B., Smola, A. J. (2001). Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond. Cambridge, MA: MIT Press.
•Rasmussen, C. E., Williams, C. K. L. (2006). Gaussian Processes for Machine Learning. Cambridge, MA: MIT Press.
•Elman, JL. (1990). Finding structure in time. Cognitive Science 14(2):179–211.
•Connor, J., Atlas, L.E., Martin, D.R. (1991) Recurrent networks and NARMA modeling. NIPS. pp. 301–308
•Makridakis, S., Spiliotis, E., Assimakopoulos, V. (2018) Statistical and Machine Learning forecasting methods: Concerns and ways forward. PLoS ONE 13(3): e0194889.
•Hochreiter, S, Schmidhuber, J. Long Short-Term Memory. Neural Computation. 1997; 9(8):1735–1780.
•Hamid SA, Habib A. Financial Forecasting with Neural Networks. Academy of Accounting and Financial Studies Journal. 2014; 18(4):37–55.
•Qiu M, Song Y. Predicting the Direction of Stock Market Index Movement Using an Optimized Artificial Neural Network Model. PLOS ONE. 2016; 11(5):1–11.
•Graves, A. (2013) Generating sequences with recurrent neural networks. arXiv preprint arXiv:1308.0850
•Bengio, Samy, Oriol Vinyals, Navdeep Jaitly, and Noam Shazeer. (2015) Scheduled sampling for sequence prediction with recurrent neural networks. In Advances in Neural Information Processing Systems, pp. 1171-1179.
•Lamb, Alex M., Anirudh Goyal, Ying Zhang, Saizheng Zhang, Aaron C. Courville, and Yoshua Bengio. (2016) Professor forcing: A new algorithm for training recurrent networks. In Advances In Neural Information Processing Systems, pp. 4601-4609.
•Cinar, Y. G., Mirisaee, H., Goswami, P., Gaussier, Ali Ait-Bachir, A., Strijov, V. (2017) Position-based Content Attention for Time Series Forecasting with Sequence-to-sequence RNNs. arXiv preprint arXiv:1703.10089.
•Chevillon, Guillaume. (2007) Direct Multi-step Estimation and Forecasting. Journal of Economic Surveys 21, no. 4: 746-785.
•Taieb, Souhaib Ben, and Amir F. Atiya. (2016) A bias and variance analysis for multistep-ahead time series forecasting." IEEE transactions on neural networks and learning systems 27, no. 1: 62-76.
•Flunkert, V., Salinas, D., and Gasthaus, J. (2017) DeepAR: Probabilistic Forecasting with Autoregressive Recurrent Networks. arXiv preprint arXiv:1704.04110.
•Wen, R., Torkkola, K., Narayanaswamy, B. A Multi-Horizon Quantile Recurrent Forecaster. (2017) arXiv preprint arXiv:1711.11053.
•Cho, Kyunghyun, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. (2014) Learning phrase representations using RNN encoder-decoder for statistical machine translation. arXiv preprint arXiv:1406.1078.
•De Gooijer, J. G. & Hyndman, R. J. (2006) 25 years of time series forecasting. Interna- tional Journal of Forecasting 22(3), 443–473.
•Pascual, L., Romo, J., & Ruiz, E. (2004) Bootstrap predictive inference for ARIMA processes. Journal of Time Series Analysis, 25, 449–465.
•Taylor, S.J., Letham, B. (2017) Forecasting at Scale. PeerJ Preprints.
•Makridakis, S. (1993) Accuracy measures: theoretical and practical concerns. International Journal of Forecasting. 9 (4): 527-529.
•Goodwin P, Lawton R. (1999) On the asymmetry of the symmetric MAPE. International Journal of Forecasting. 15(4):405–408.
•Hyndman RJ, Koehler AB. (2006) Another look at measures of forecast accuracy. International Journal of Forecasting. 22(4):679–688.
•Kim, S., Kim, H. (2016) A new metric of absolute percentage error for intermittent demand forecasts. International Journal of Forecasting. 32: 669–679.
•Chen C, Twycross J, Garibaldi JM (2017) A new accuracy measure based on bounded relative error for time series forecasting. PLoS ONE 12(3): e0174202.
•Matthias W Seeger, David Salinas, and Valentin Flunkert. (2016) Bayesian intermittent demand fore- casting for large inventories. In Advances in Neural Information Processing Systems, pages 4646–4654.
•Vasile, F., Smirnova, E., Conneau, A. (2016) Meta-Prod2Vec - Product Embeddings Using Side-Information for Recommendation. arXiv preprint arXiv:1607.07326.
•M. Grbovic, V. Radosavljevic, N. Djuric, N. Bhamidipati, J. Savla, V. Bhagwan, and D. Sharp. (2015) E-commerce in your inbox: Product recommendations at scale. arXiv preprint arXiv:1606.07154.
•T. Mikolov, K. Chen, G. Corrado, and J. Dean. (2013) Efficient estimation of word representations in vector space. arXiv preprint arXiv:1301.3781.
•Zolna, K., Romanski, B. (2016) user2vec: user modeling using LSTM networks. Proceedings of the 33rd International Conference on Machine Learning, New York, NY, USA.

Connect to Customers

Presenter: Di Wu

Connect To Customers

- Connecting products to customers seamlessly in all scenarios.
- People are different in many ways

Background

Locations

Activities

Social Connections

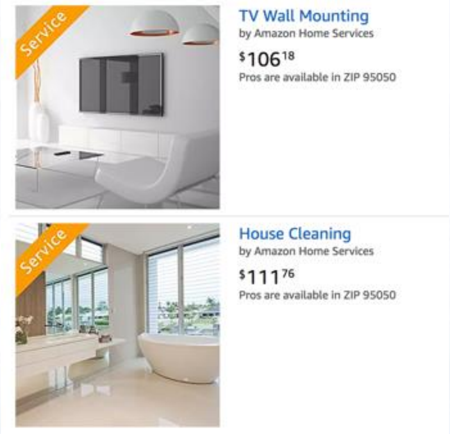
Connect To Customers

- Connecting products to customers seamlessly in all scenarios.
- Products are different in many ways

Physical Products

Digital Goods

Service



Content

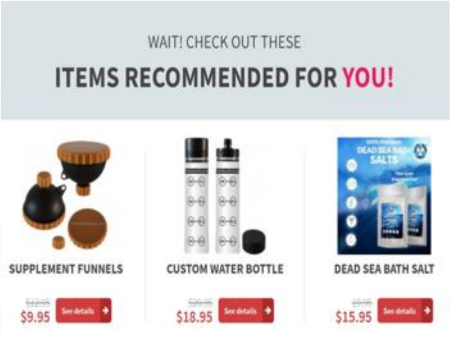
Connect To Customers

- Delivering the right products to the right customers at the right place and right time

Browsing

Search

Recommendation



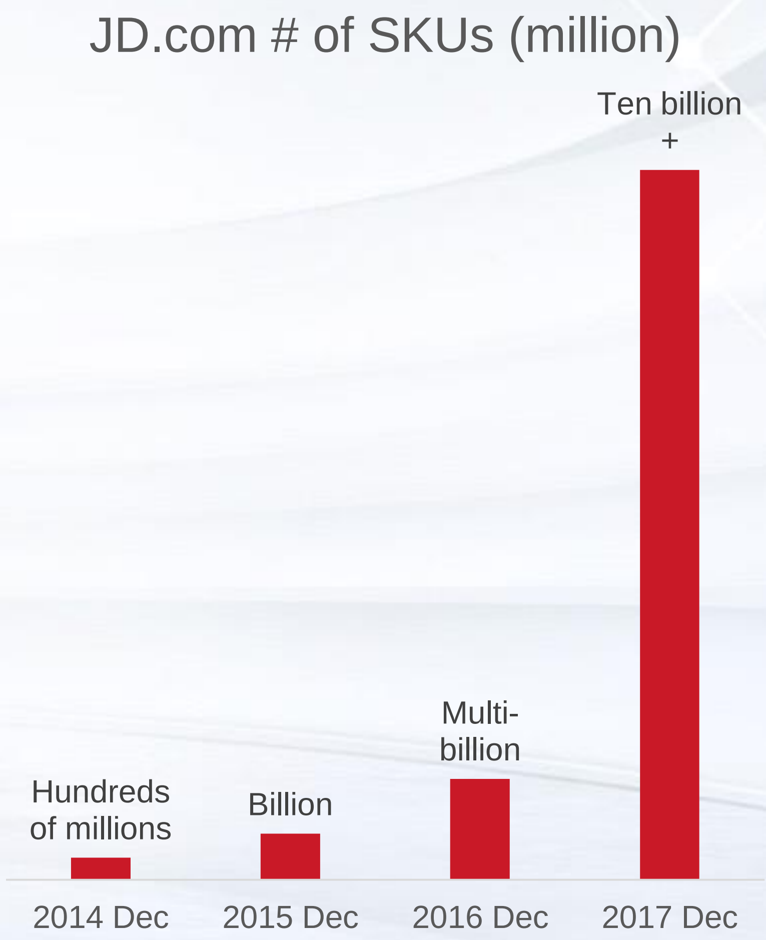
Advertising

Social Network

C2M

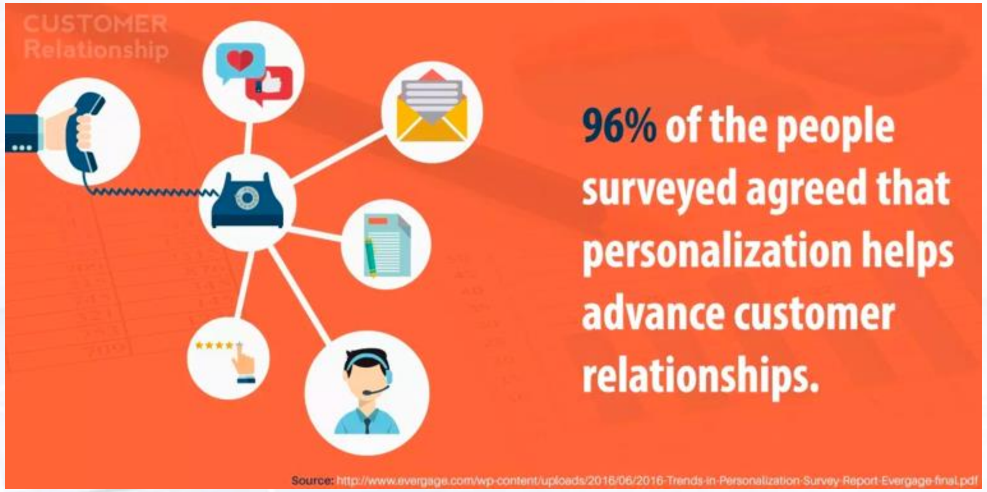
Product Recommendation in E-Commerce

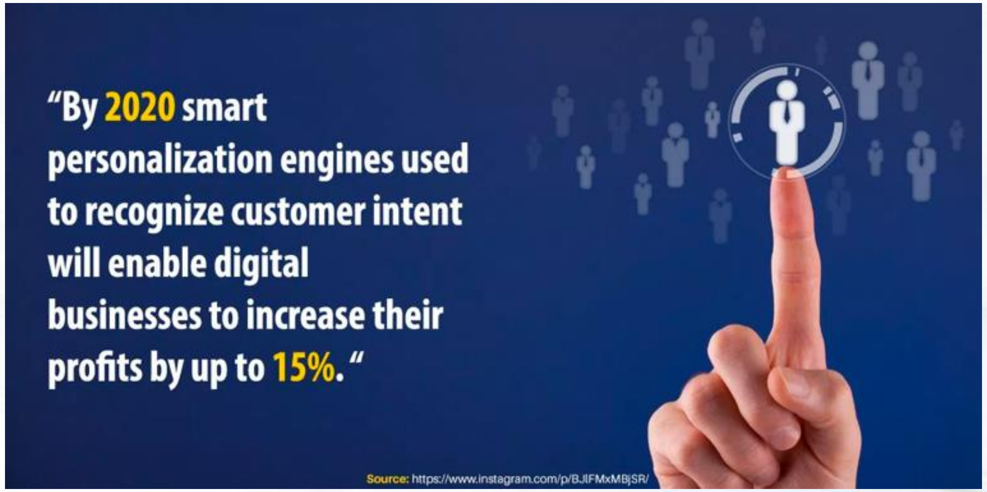
- With an ever increasing number of products available to customers, delivering the most appropriate products to customers has become a core functionality of retail platforms.
- Naturally, product recommendation has now become a centerpiece of e-commerce platforms.



Product Recommendation in E-Commerce

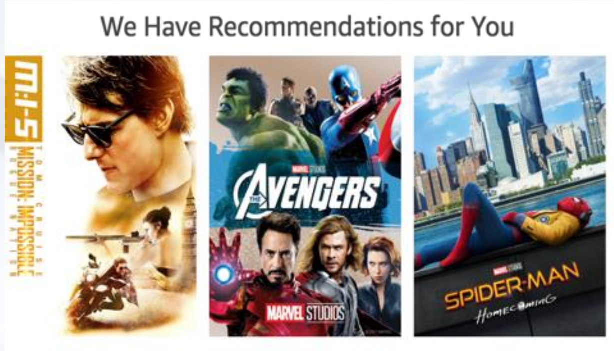
- 35% of goods purchased on Amazon and 75% of content watched on Netflix come about as a result of product recommendations.

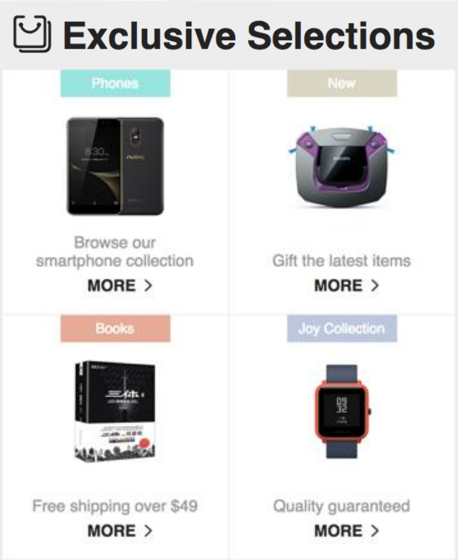


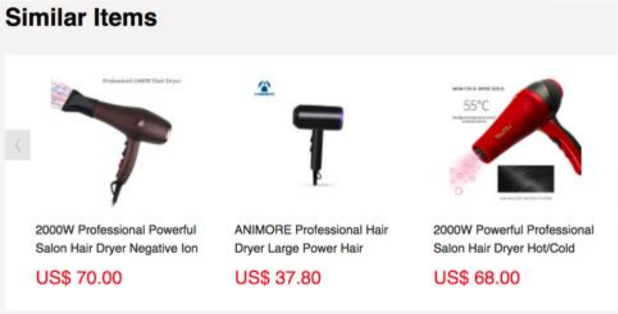


(www.intelliverse.com/blog)

Product Recommendation Example





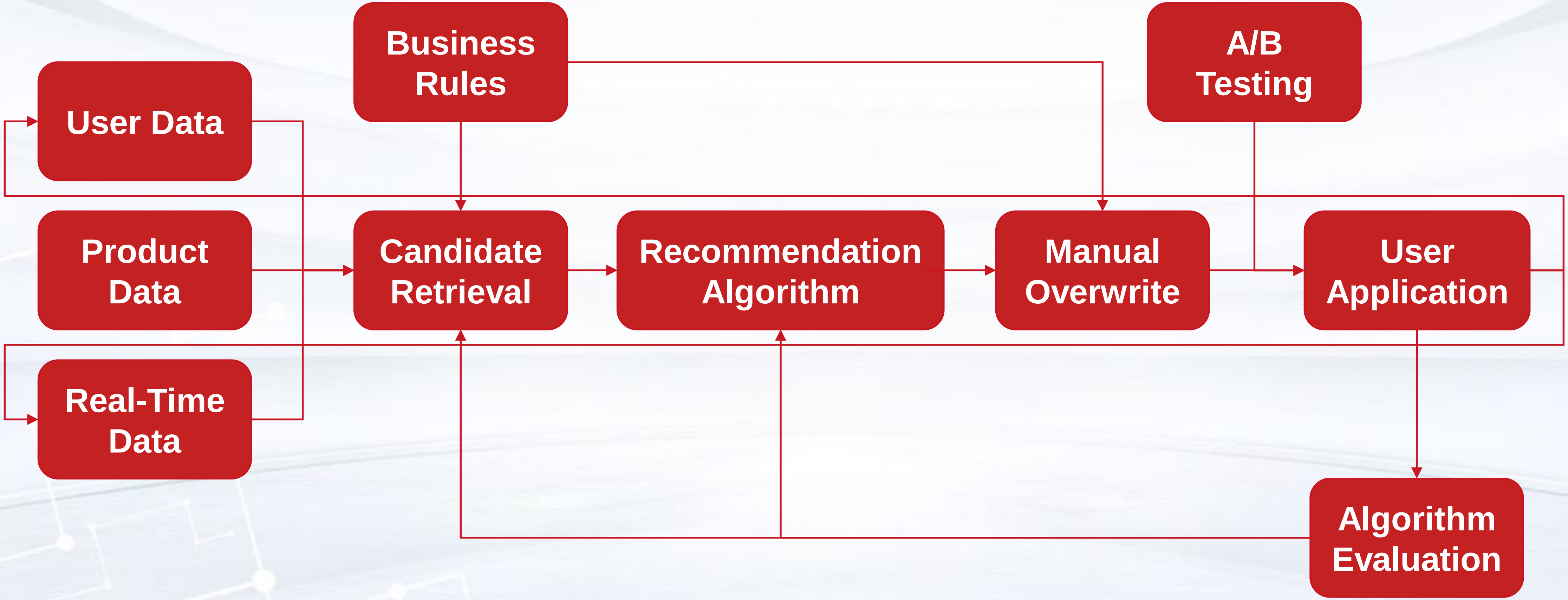


Uniqueness of Product Recommendation

JD.COM

- Other common recommendation systems
  - Video and music: Netflix, Pandora, Tik Tok (Douyin), etc.
  - News and information: Google News, Facebook, Toutiao, etc.
- Key differences between common recommendation and product recommendation in e-commerce.
  - Different objectives
    - Strong preference to business metrics such as revenue or profit, vs. indirect measurement such as browsing time, click count, etc.
    - Need to balance multiple objectives (GMV, order value, conversion, ads revenue, etc.)
  - Varying customer intentions
    - Shopping customers have stronger and changing short-term intention vs. common recommendations' main objective is to capture customers' long-term preference.
  - Complicated product relationships
    - Complementarity and substitutability of products can be hard to estimate, while document/video similarity is relatively well-defined.
  - Different repurchase behaviors

Recommendation System Framework



Inputs

User Data	Product Data	Real-Time Data
User Identifier	Category	Location
Demographic Information	Brand/Manufacture	Time
Shopping Habit	Origin	Device
Shopping History	Rating	Session Information
Browse History	Product Price	Product Searches
Favorite/Disliked Items	Product Description	Product Impressions
Devices	Product Images	Product Browses
…	…	…
Describe users, their preferences, their histories, etc.	Describe the all things related to the products and all product-related user interactions.	Describe the shopping scenario and users' interaction with the shopping scenario

Evaluation Metrics

JD.COM

True North Metrics

Total Order Numbers

Total Visit Duration

Gross Merchandise Value (GMV)

Total Gross Profit

Total Net Profit

Objective Metrics

Click-Through Rate

$$CTR = \frac{|\{clicked\ items\}|}{|\{retrieved\ items\}|}$$

Conversion Rate

$$CVR = \frac{|\{purchased\ items\}|}{|\{retrieved\ items\}|}$$

Precision

$$precision = \frac{|\{relevant\ items\} \cap \{retrieved\ items\}|}{|\{retrieved\ items\}|}$$

Recall

$$recall = \frac{|\{relevant\ items\} \cap \{retrieved\ items\}|}{|\{relevant\ items\}|}$$

F-Score

$$F = \frac{2 \cdot precision \cdot recall}{precision + recall}$$

Area under the ROC Curve (AUROC)

Average Precision

$$AveP = \frac{\sum_{k=1}^{n}\left(precision(k) \cdot rel(k)\right)}{|\{relevant\ items\}|}$$

Half Life Utility

$$R_a = \sum_k \frac{u(a,j)}{2^{\frac{(idx(k)-1)}{\alpha-1}}}, \qquad R = \frac{R_a}{R_{max}}$$

Discounted Cumulative Gain (DCG)

$$DCG_p = \sum_{i=1}^{p} \frac{u(i)}{\log_2(i+1)}$$

Normalized Discounted Cumulative Gain (nDCG)

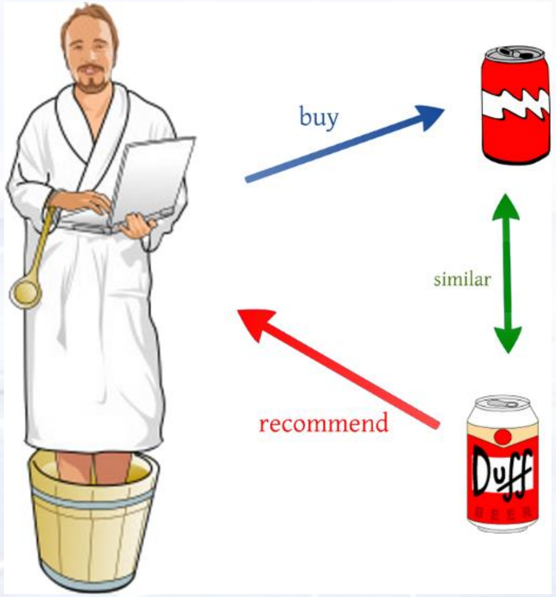
$$\mathrm{n}DCG_p = \frac{DCG_p}{IDCG_p} = \frac{\sum_{i=1}^{p} \frac{u(i)}{\log_2(i+1)}}{\sum_{i=1}^{REL} \frac{u(i)}{\log_2(i+1)}}$$

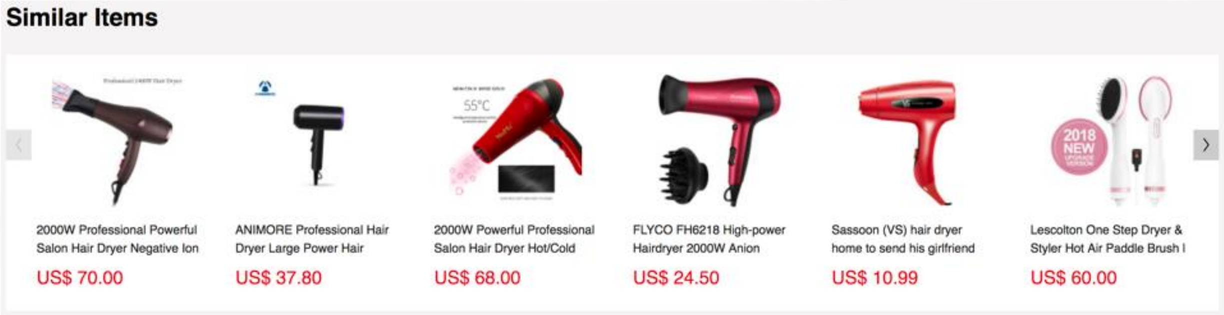
Types of Product Recommendation Algorithms

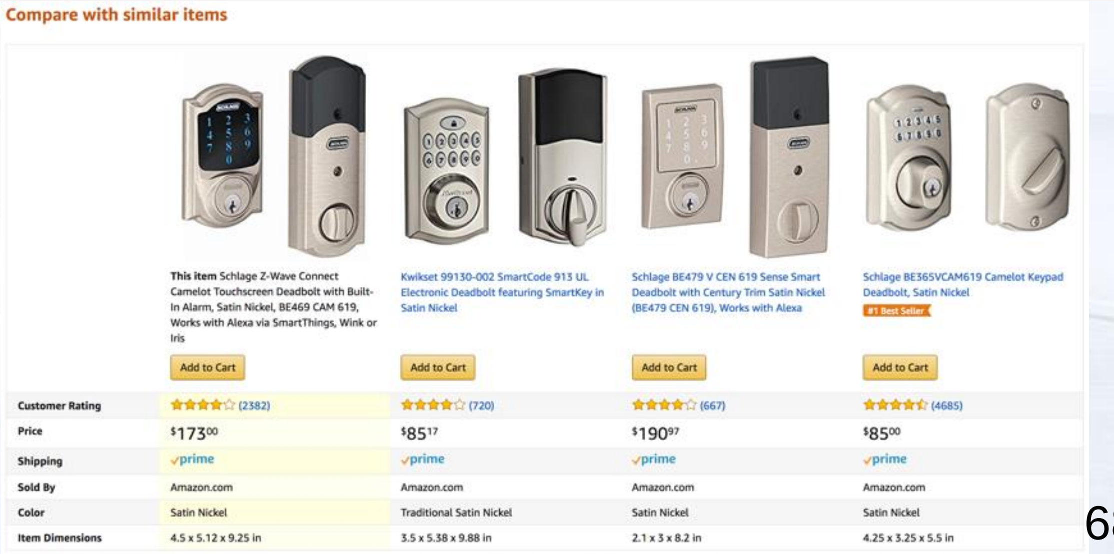
- Content Based Methods (Ricci et al., 2015; Pazzani and Billsus, 2007)
  - Recommends items similar to those liked/purchased by the customer in the past
  - Use attributes of items/customers
- Collaborative Filtering Based Methods (Goldberg et al., 1992; Linden et al., 2003; Schafer et al., 2007)
  - Recommends items liked or purchased by similar customers
  - Enable exploration of diverse content
- Hybrid Methods (Burke, 2002; Zhang, et al., 2017)
  - A combination of both methods
  - Deep Learning techniques have been proven to be effective
  - Recommends items by embedding features in different levels
  - Enable exploration of context, time and sequence

Content Based Recommendation

- Based on similarity of item attributes
  - Item name, categorical information, price, description, technical specs, etc.
- Challenges:
  - Vague definition of similarity
  - Cannot provide diverse content



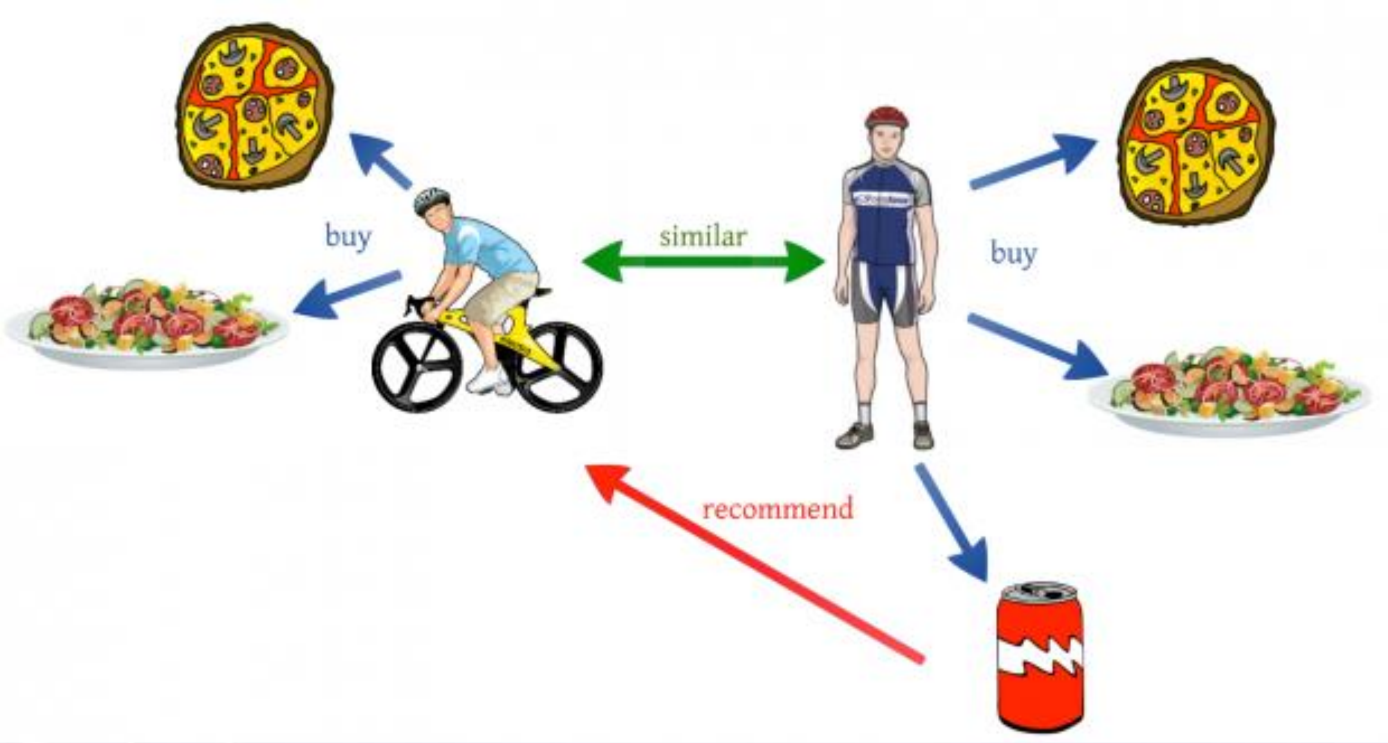




(https://d4datascience.wordpress.com/2016/07/22/recommender-systems-101/)

Collaborative Filtering

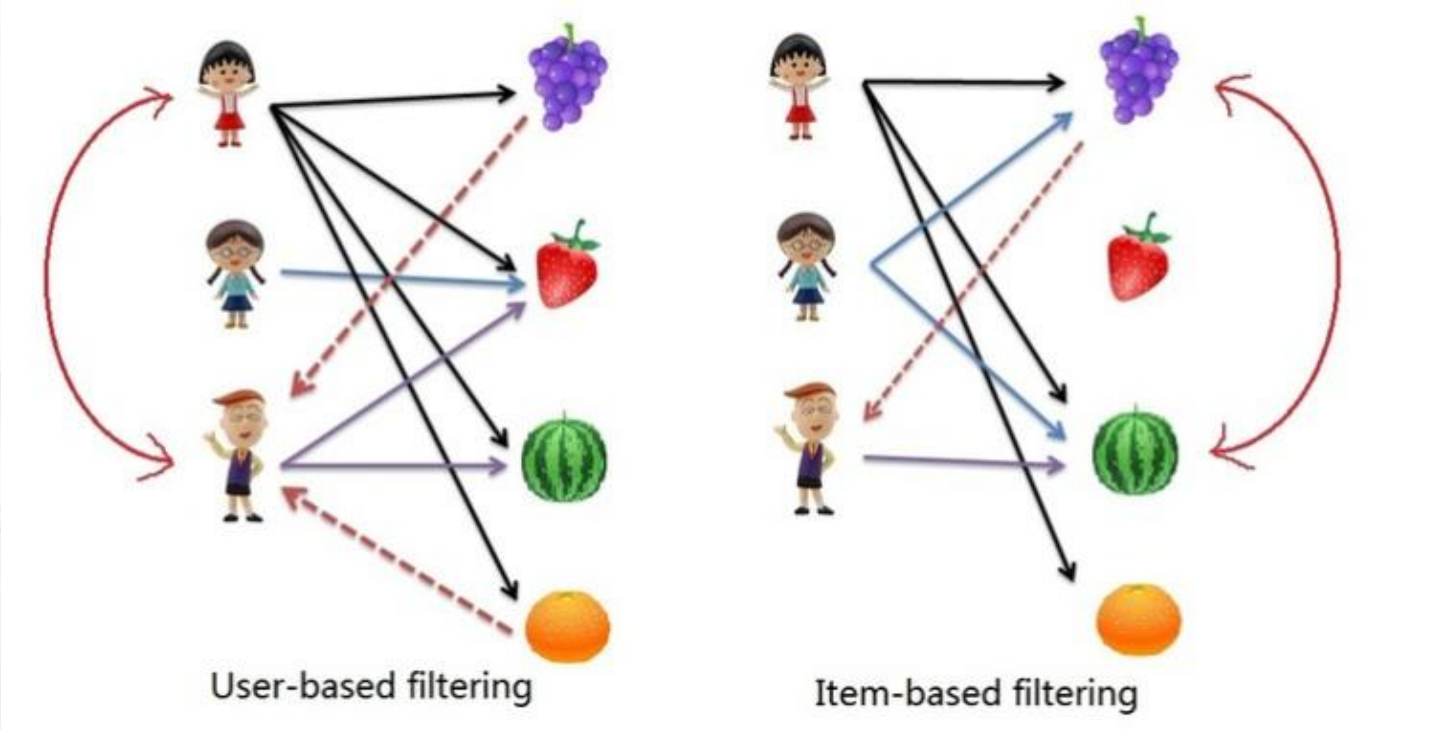
- Collaborative Filtering is the process of filtering or evaluating items using the opinions of other people.



(https://d4datascience.wordpress.com/2016/07/22/recommender-systems-101/)

Collaborative Filtering

- User-User Collaborative Filtering
- Item-Item Collaborative Filtering



(https://d4datascience.wordpress.com/2016/07/22/recommender-systems-101/)

Collaborative Filtering

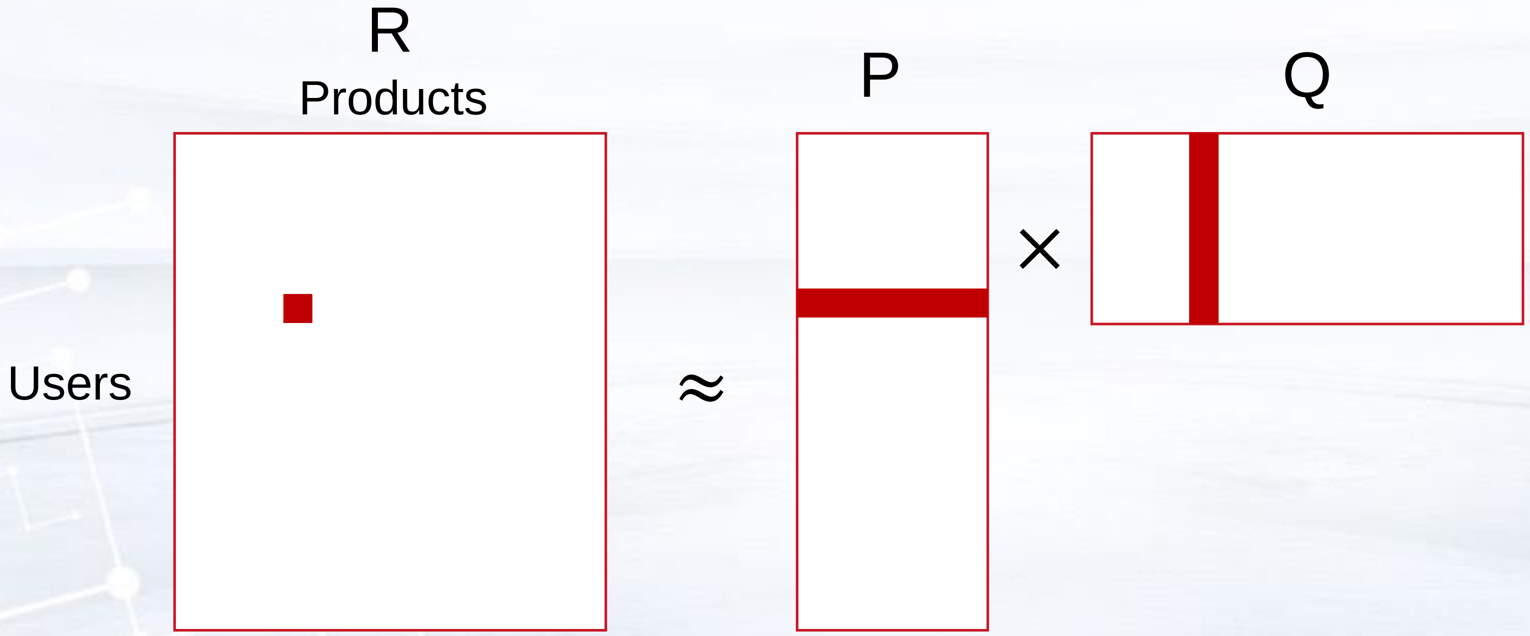
- Association Rule Mining Algorithm
  - Agrawal, et al., 1993; Mobasher, et al., 2001; Lin, et al., 2002
- Probabilistic (model-based) Algorithm
  - Breese, et al., 1998; Rendle, et al., 2009
- Nearest-Neighbor (memory-based) Algorithm
  - Sarwar, et al., 2001; Deshpande and Karypis, 2004
- Dimensionality Reduction (matrix factorization) Algorithm
  - Koren, et al., 2009; Sarwar, et al., 2000; Paterek, 2007

Matrix Factorization Based Recommender

- Populated by the winning algorithms of Netflix Prize (Koren et al., 2009)

$$r_{ui} \sim \hat{r}_{ui} = q_i^T \cdot p_u$$

$$\min_{q,p} \sum_{u,i} \left(r_{ui} - q_i^T \cdot p_u\right)^2 + \lambda \cdot \left(\|q_i\|^2 + \|p_u\|^2\right)$$



Collaborative Filtering Based Recommendation

JD.COM

CF-Based Methods
- Collaborative Filtering (CF, Schafer et al, AdaptiveWeb'07)
- Matrix Factorization (MF, Koren et al, Computer'09)

Context-Aware
- SVD++ model (Koren et al, KDD'08)
- Behavior Factorization (Zhao et al, WWW'17)

Time-Aware
- TimeSVD++ model (Koren et al, KDD'09)
- Personalized Recommendation (STAR, Song et al, CIKM'15)

Sequence-Aware
- Factorizing Personalized Markov Chain (FPMC, Rendle et al, WWW'10)

- Challenges:
  - Cold-start problem
  - Inflexibility in adding new features
  - Time consuming calculation
  - Hard to scale

Hybrid Recommendation System

- Content based recommendation and collaborative filtering are complementary.
- Hybrid systems combine collaborative and content-based methods
  - Combining separate recommenders
  - Adding content-based characteristics to collaborative models
  - Adding collaborative characteristics to content-based models
  - **Developing a single unifying recommendation model**

(Adomavicius and Tuzhilin, 2005)

Hybrid Recommendation System

- Machine learning based recommendation

User Features

User-Product Interaction Features

$$r_{ui} \sim \alpha + \beta \cdot v_u + \gamma \cdot p_i + \delta \cdot x_{u,i}$$

Product Features

- Challenges:
  - Huge (sparse) feature space
  - Requires a lot of feature engineering

Hybrid Recommendation System

JD.COM

- Machine learning based recommendation

$$\underbrace{r_{ui}\sim\alpha+\beta\cdot v_u}_{\text{User Features}}+\gamma\cdot p_i+\delta\cdot x_{u,i}$$

User Features

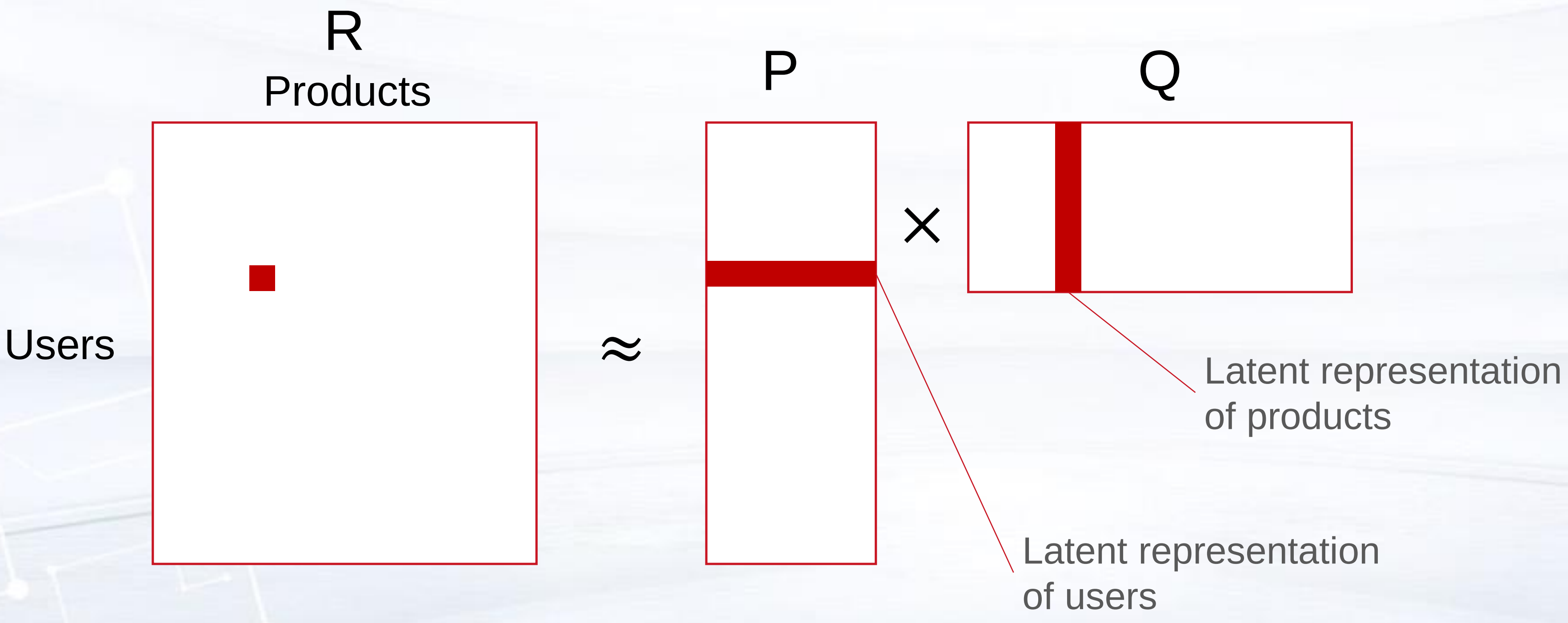
User-Product Interaction Features

Product Features

- Challenges:
  - Huge (sparse) feature space
  - Requires a lot of feature engineering

→ Deep Learning

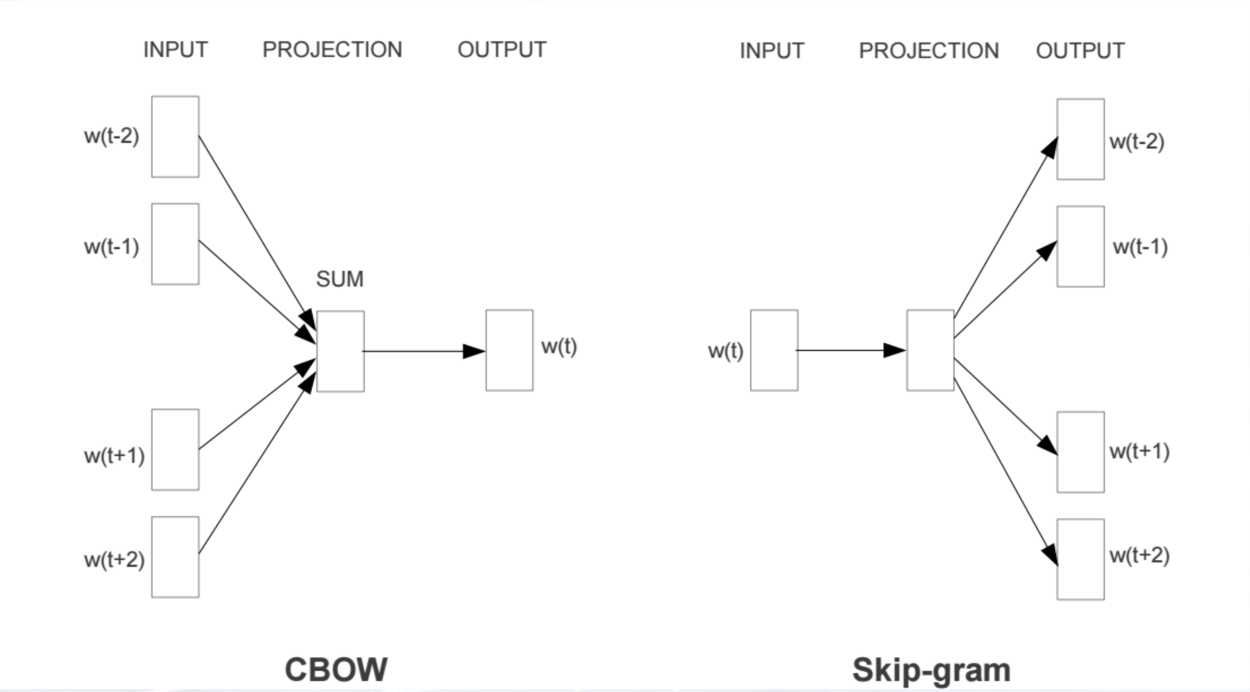
- Recap: Matrix Factorization

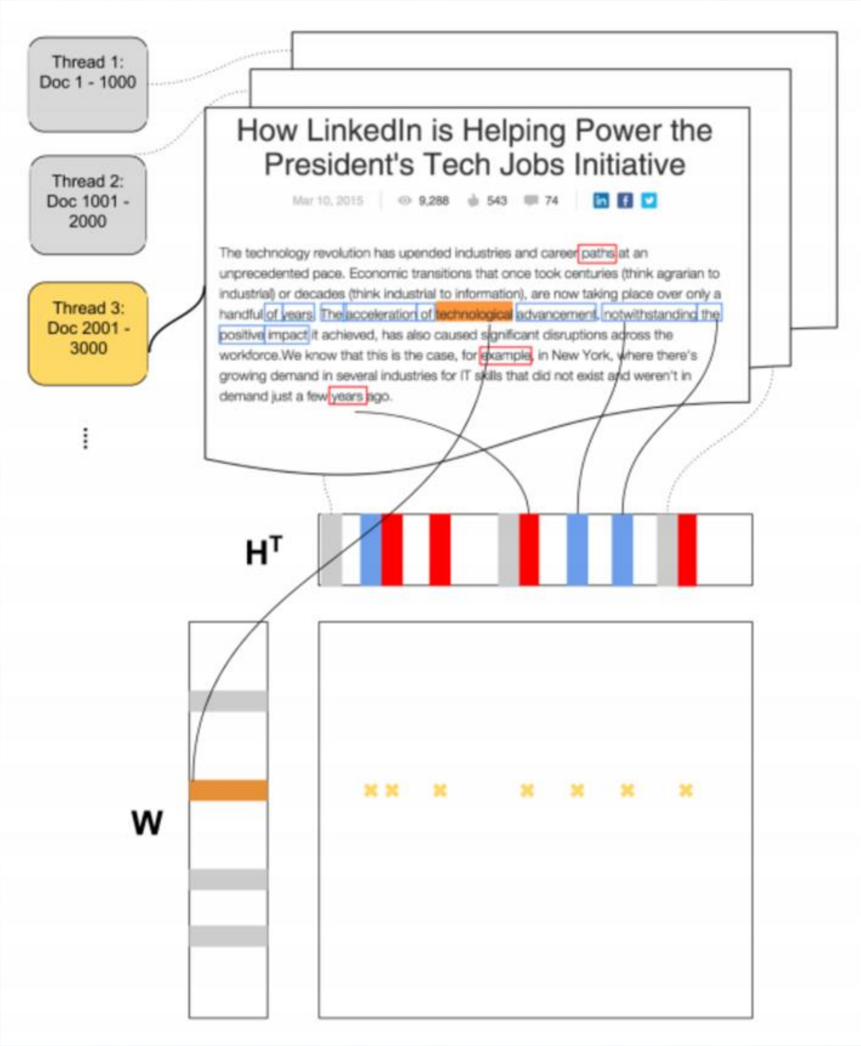


Embedding

Embedding - Word2vec

- For each word i, learns the low dimensional embedding $w_i \in \mathbb{R}^k$ (Mikolov et al., 2013)
- “Shallow” neutral networks

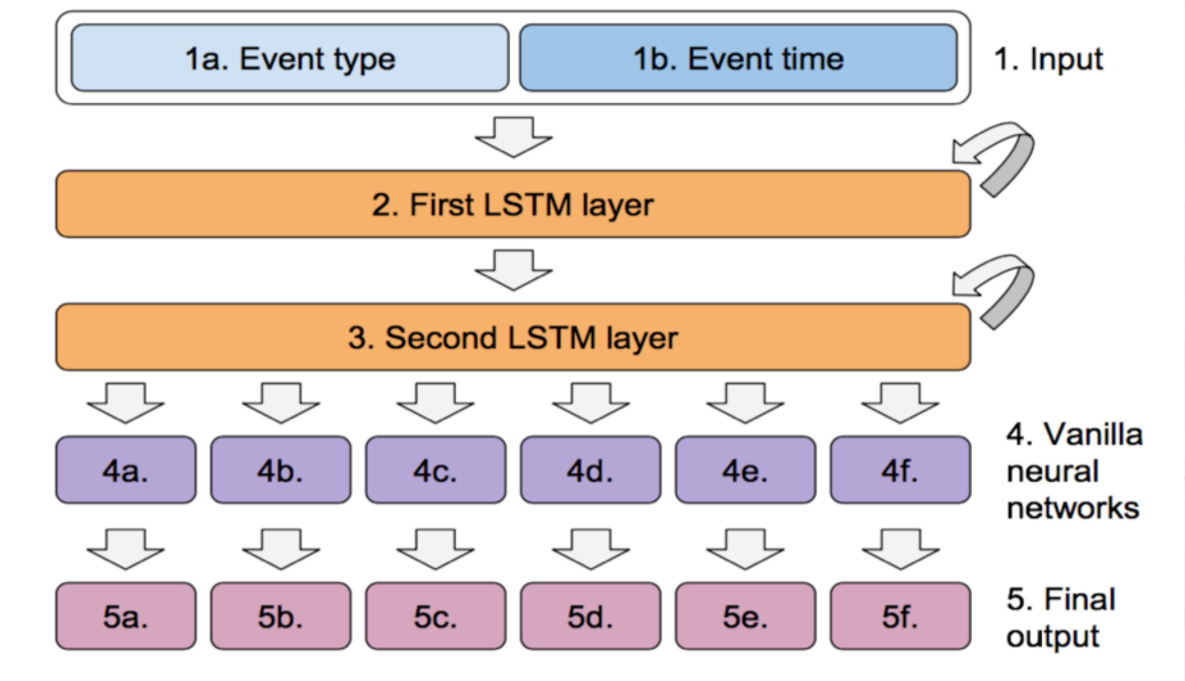




(Mikolov et al., 2013)

Embedding - User2vec

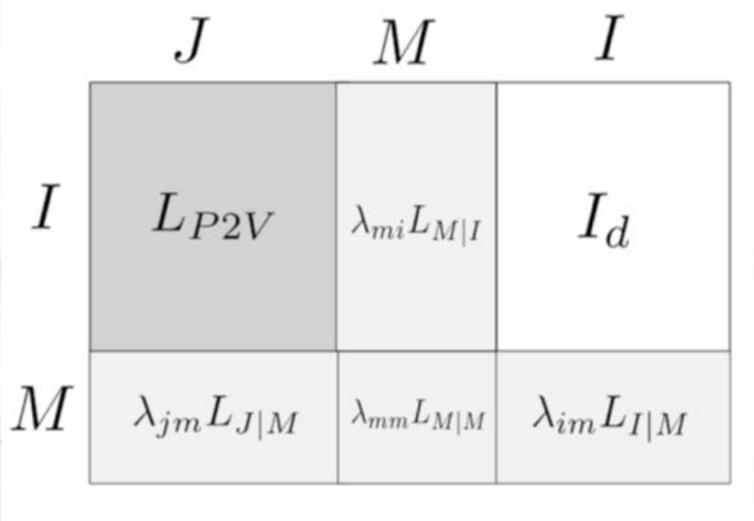
- Input: the sequence of all the events from the customer's activity.
- Targets: answers to a fixed list of questions asked at every event.

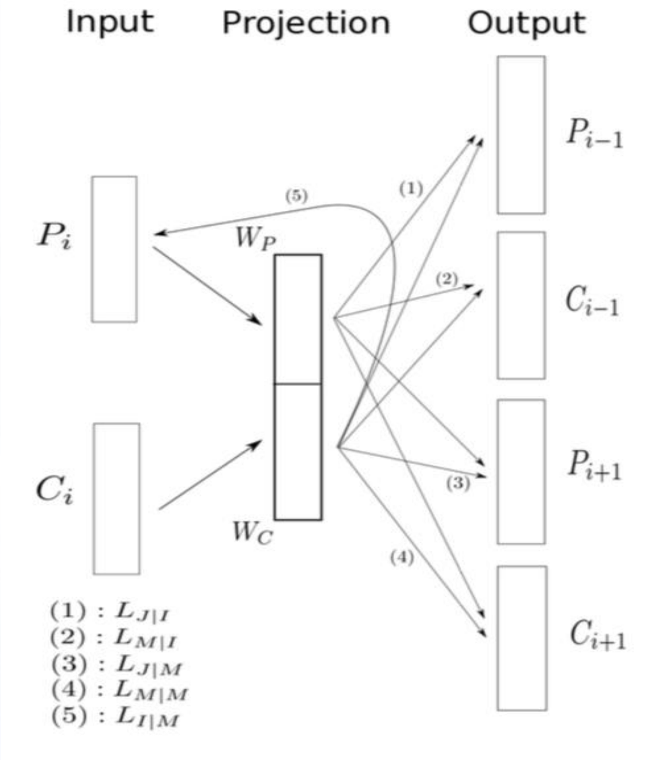


(Zolna and Romanski, 2016)

Embedding - Prod2vec & Meta-Prod2vec

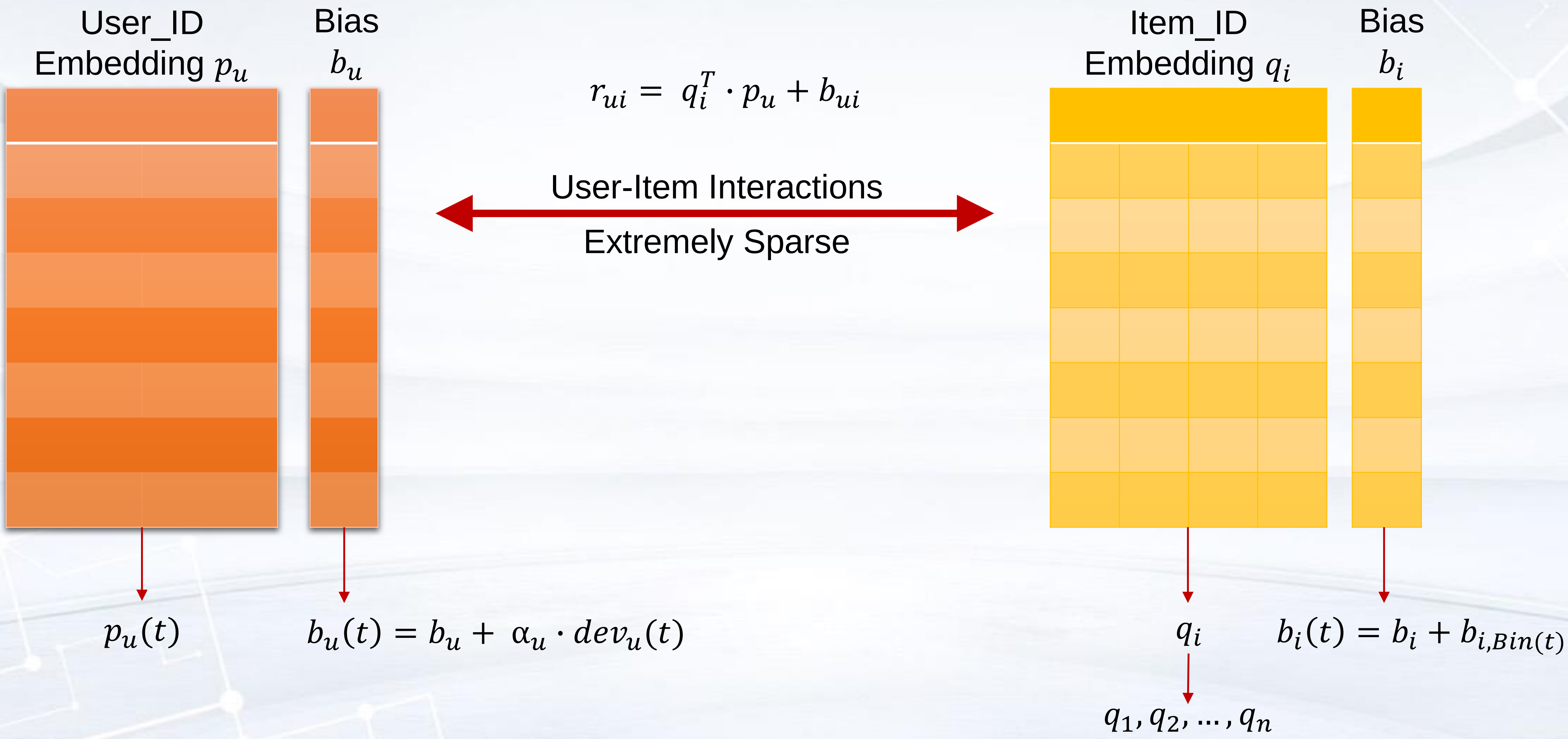
- Prod2vec: uses Word2Vec on sequences of product receipts
- Meta-Prod2vec: adds product metadata as side information



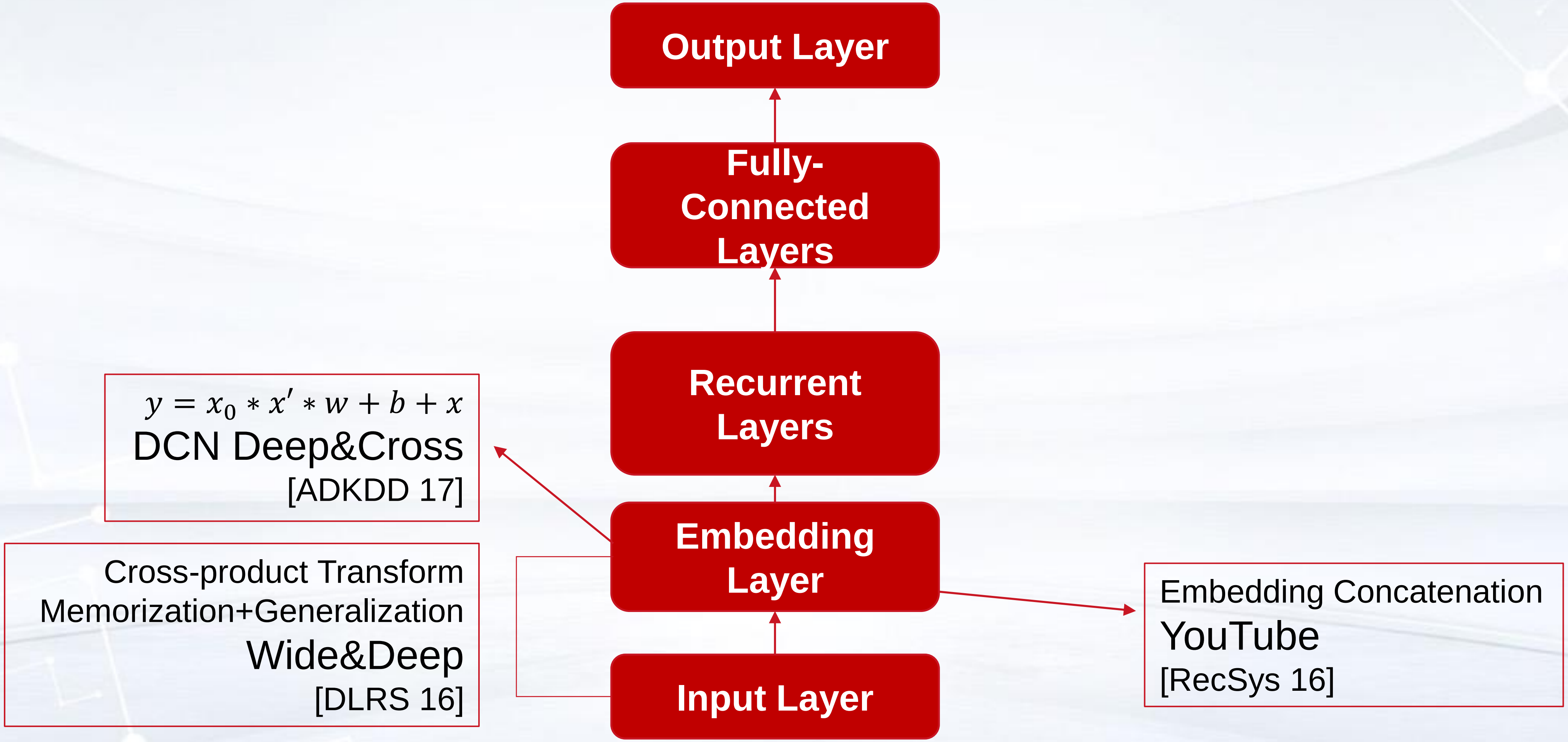


(Vasile et al., 2016)

Deep Learning Based Recommendation



DL-Based Recommendation + Context



DL-Based Recommendation + Time

JD.COM

Multi-task Training LSTM
Ob Func: negative Poisson log-likelihood
NSR
[WSDM 17]

Output Layer

Fully-Connected Layers

Recurrent Layers

Embedding Layer

Input Layer

Updating Gating Mechanism:
$h_t = LSTM(h_{t-1}, z_t)$
RRN
[WSDM 17]

Gating Mechanism:
+time gates(Phased-LSTM)
Time-LSTM
[IJCAI 17]

$y = x_0 * x' * w + b + x$
DCN Deep&Cross
[ADKDD 17]

$h^{(\tau)} = (1 + w_t + w_d) * h^{(\tau)}$
Latent Cross
[WSDM 18]

Cross-product Transform
Memorization+Generalization
Wide&Deep
[DLRS 16]

Embedding Concatenation
YouTube
[RecSys 16]

DL-Based Recommendation + Sequence

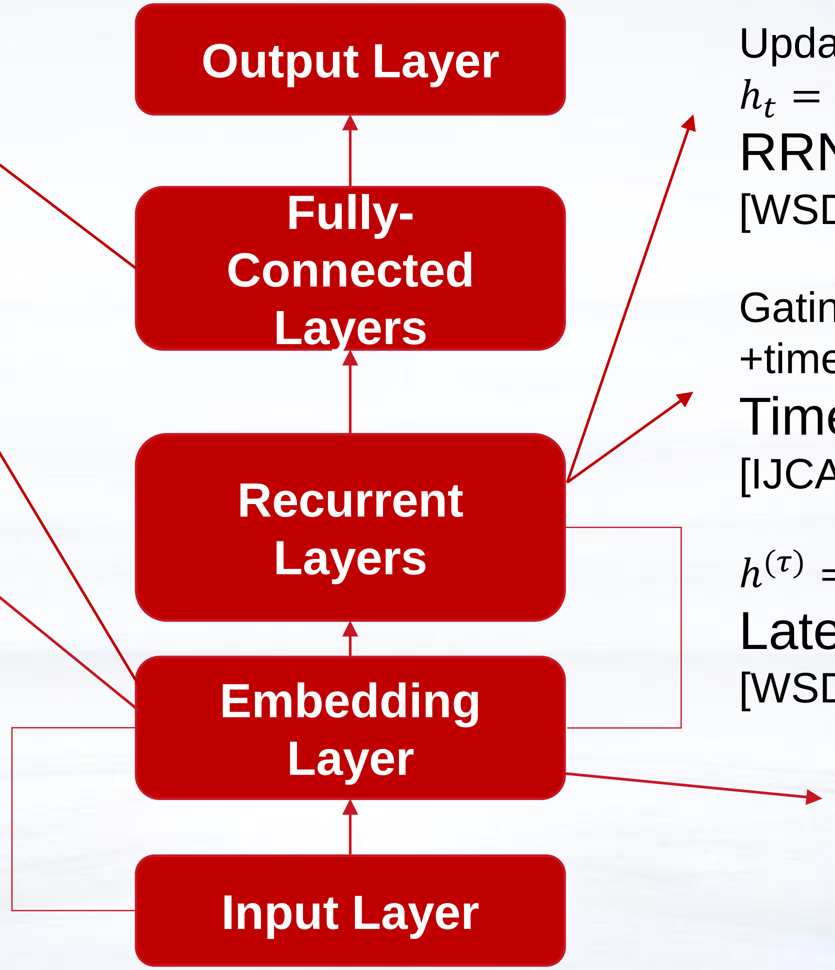
JD.COM

Multi-task Training LSTM
Ob Func: negative Poisson log-likelihood
NSR
[WSDM 17]

Session Sequence Embedding
GRU4REC
[ICLR 16]

$y = x_0 * x' * w + b + x$
DCN Deep&Cross
[ADKDD 17]

Cross-product Transform
Memorization+Generalization
Wide&Deep
[DLRS 16]



Updating Gating Mechanism:
$h_t = LSTM(h_{t-1}, z_t)$
RRN
[WSDM 17]

Gating Mechanism:
+time gates(Phased-LSTM)
Time-LSTM
[IJCAI 17]

$h^{(\tau)} = (1 + w_t + w_d) * h^{(\tau)}$
Latent Cross
[WSDM 18]

Embedding Concatenation
YouTube
[RecSys 16]

CF-Based Recommendation vs. DL-Based Recommendation

JD.COM

CF-Based Methods

- Collaborative Filtering (CF, Schafer et al, AdaptiveWeb'07)
- Matrix Factorization (MF, Koren et al, Computer'09)

DL-Based Methods

- YouTube DNN Model (Covington et al, RecSys'16)
- Neural Collaborative Filtering (NCF, He et al, WWW'17)

Context-Aware

- SVD++ model (Koren et al, KDD'08)
- Behavior Factorization (Zhao et al, WWW'17)
- Deep & Cross model (DCN, Wang et al, ADKDD'17)
- Latent Cross model (Beutel et al, WSDM'18)

Time-Aware

- TimeSVD++ model (Koren et al, KDD'09)
- Personalized Recommendation (STAR, Song et al, CIKM'15)
- Time-LSTM model (Zhu et al, IJCAI'17)
- Neural Survival Recommendation (NSR, Jing et al, WSDM'17)

Sequence-Aware

- Factorizing Personalized Markov Chain (FPMC, Rendle et al, WWW'10)
- Session-based RNN (GRU4REC, Hidasi , ICLR'16)

Case Study: JD.COM's DORS

JD.COM

- JD.COM’s Deep Online Ranking System (DORS)
  - Front page product recommendation
  - Endless item flow
  - Presented in page-wise fashion
  - Each page contains a fixed number of items

Case Study: JD.COM's DORS

JD.COM

- Evaluation Metrics
  - GMV
  - Order numbers
  - Overall and page-wise normalized discounted cumulative gains (nDCG)

$$DCG_{p,page-k} = \sum_{i=1,k\in page}^{p} \frac{gmv_{ki} I(x_{ki})}{\log_2(i+1)}$$

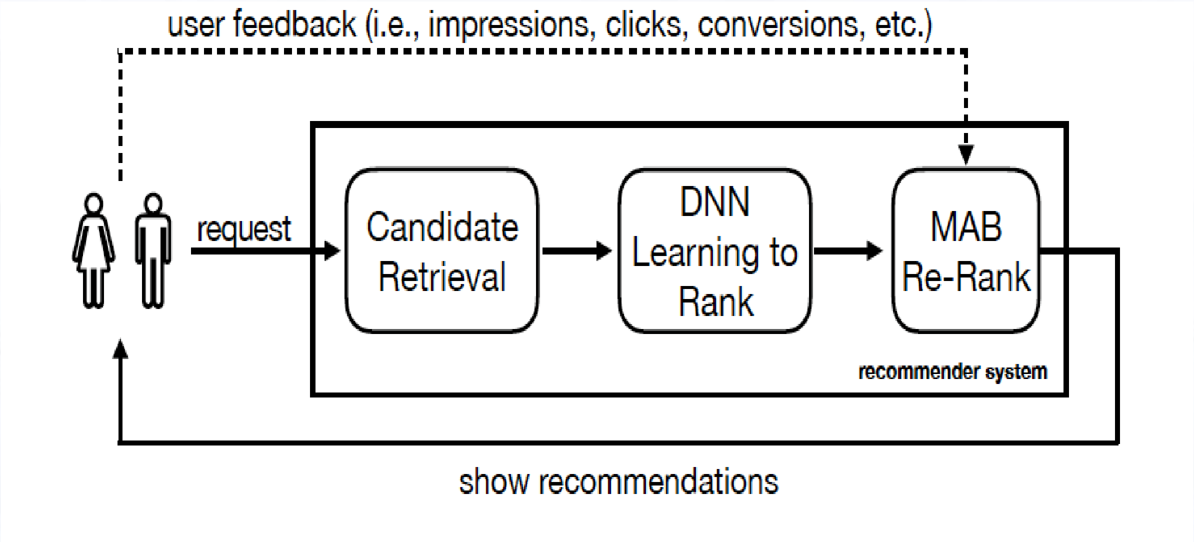
$$IDCG_{p,page-k} = \max_{H} DCG_{p,page-k}$$

$$NDCG_{p,page-k} = \frac{DCG_{p,page-k}}{IDCG_{p,page-k}}$$

$$\Delta NDCG_{p,page-k} = \left(\frac{NDCG_{p,page-k}^{test}}{NDCG_{p,page-k}^{control}} - 1.0 \right) \times 100.0\%$$

Case Study: JD.COM's DORS

- JD.COM's scalable deep online ranking system (DORS)
  - Presents a relevant, responsive, and scalable recommendation system.
  - Implemented in a three-level architecture.
  - Able to precisely capture users' real-time purchasing intents.



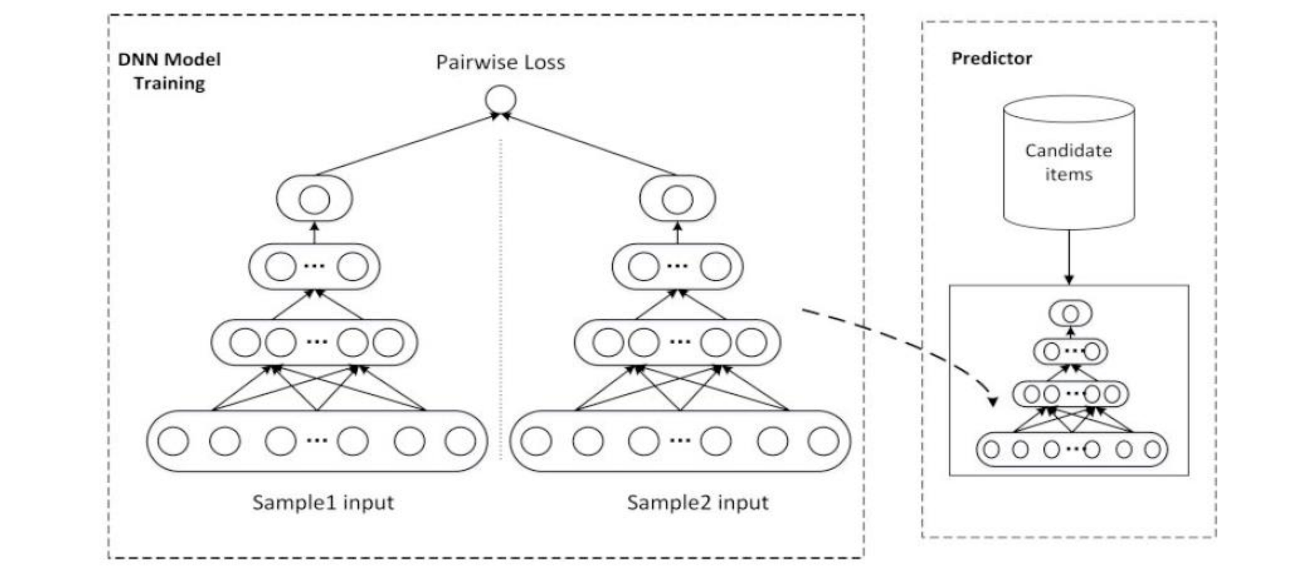
(Yan et al, 2018)

DORS - Learning-to-rank via DNN

- Data: full sets of both item and user offline features (Dimension: 5e10^8)
- Algorithm: pairwise architecture of learning-to-rank DNN
- Goal: scores all candidates for long-term user intents

$$y_i = \mathcal{Y}(gmv_{s_i}, \mathcal{I}(s_i)) = \frac{gmv_{s_i}}{\max_{s \in c_{s_i}} gmv_s} \times \mathcal{I}(s_i)$$

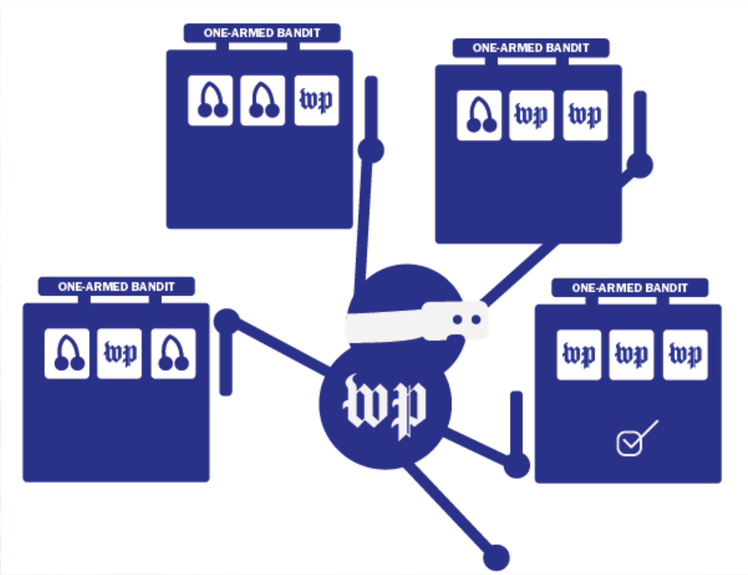
$$\mathcal{L} = \sum_j [(\hat{y}_{1j} - y_{1j})^2 + (\hat{y}_{2j} - y_{2j})^2 + \lambda \max(0, \gamma - (\hat{y}_{1j} - \hat{y}_{2j})(y_{1j} - y_{2j}))]$$



(Yan et al, 2018)

DORS - Online Re-ranking via Multi-Arm Bandits (MAB)

- Organizes items by categories (arms).
- Warm starts the MAB by initializing the algorithm with DNN scores.
- Uses customers real-time clicks and impressions as positive and negative responses.
- Re-ranks recommendations in real-time using MAB with real-time customer feedbacks (i.e. impressions, clicks)



(oukas.info)

(Yan et al, 2018)

Summary: Connect to Customers

- Connect Products to Customers
- Recommendation Systems Introduction
  - Content Based Method
  - Collaborative Method
  - Deep Learning Based Method
- Case Study: JD.COM’S Deep Online Ranking System (DORS)

Summary: Recommendation Algorithms

References

- Ricci, F., Rokach, L., Shapira, B. (2015) Recommender Systems: Introduction and Challenges. In: Ricci F., Rokach L., Shapira B. (eds) Recommender Systems Handbook. Springer, Boston, MA.
- Pazzani, M., Billsus, D. (2007) Content-based Recommendation Systems. In: Brusilovsky, P., Kobsa, A., Nejdl, W. (eds.) The Adaptive Web: Methods and Strategies of Web Personalization. LNCS, vol. 4321, pp. 325–341. Springer, Heidelberg.
- Goldberg, D., Nochols, D., Oki, B.M., Terry, D. (1992) Using collaborative filtering to weave an information tapestry, Communications of the ACM, 35(12), 61-70.
- Linden, G., Smith, B., York, J. (2003) Amazon.com recommendations: Item-to-item collaborative filtering. IEEE Internet Computing 7(1), 76–80.
- Schafer, B., Frankowski, D., Herlocker, J., Sen, S. (2007) Collaborative Filtering Recommender Systems. In: Brusilovsky, P., Kobsa, A., Nejdl, W. (eds.) The Adaptive Web: Methods and Strategies of Web Personalization. LNCS, vol. 4321, pp. 291–324. Springer, Heidelberg.
- Burke, R. (2002) Hybrid recommender systems: Survey and experiments. User modeling and user-adapted interaction 12 (4), 331-370.
- Zhang, S., Yao, L., Sun, A. (2017) Deep learning based recommender system: A survey and new perspectives." arXiv preprint arXiv:1707.07435.
- Agrawal, R., Imieliński, T., Swami, A. (1993) Mining association rules between sets of items in large databases. Acm sigmod record 22(2).
- Mobasher, B., Dai, H., Luo, T., Nakagawa, M. (2001) Effective personalization based on association rule discovery from web usage data. Proceedings of the 3rd international workshop on Web information and data management.
- Lin, W., Alvarez, S.A., Ruiz, C. (2002) Efficient adaptive-support association rule mining for recommender systems. Data mining and knowledge discovery 6(1): 83-105.
- Sarwar, B., Karypis, G., Konstan, J., Riedl, J. (2001) Item-based collaborative filtering recommendation algorithms. Proceedings of the 10th international conference on World Wide Web.
- Deshpande, M., Karypis, G. (2004) Item-based top-n recommendation algorithms. ACM Transactions on Information Systems (TOIS) 22(1): 143-177.
- Breese, J.S., Heckerman, D., Kadie, C. (1998) Empirical analysis of predictive algorithms for collaborative filtering. Proceedings of the Fourteenth conference on Uncertainty in artificial intelligence. Morgan Kaufmann Publishers Inc..
- Rendle, S., Freudenthaler, C., Gantner, Z., Schmidt-Thieme, L. (2009) BPR: Bayesian personalized ranking from implicit feedback. Proceedings of the twenty-fifth conference on uncertainty in artificial intelligence. AUAI Press.
- Koren, Y., Bell, R.M., Volinsky, C. (2009) Matrix factorization techniques for recommender systems. IEEE Computer 42(8), 30–37.
- Sarwar, B., Karypis, G., Konstan, J., Riedl, J. (2000) Application of dimensionality reduction in recommender system-a case study. No. TR-00-043. Minnesota Univ Minneapolis Dept of Computer Science.
- Paterek, A. (2007) Improving regularized singular value decomposition for collaborative filtering. Proceedings of KDD cup and workshop.
- Adomavicius, G., Tuzhilin, A. (2011) Context-Aware recommender Systems, in: F. Ricci, et al. (Ed.), Recommender Systems Handbook, pp. 217–253.
- Campos, P.G., Díez, F., Cantador, I. (2014) Time-aware recommender systems: a comprehensive survey and analysis of existing evaluation protocols. User Model User-Adap Inter 24:67–119.
- Quadrana, M., Cremonesi, P., Jannach, D. (2018) Sequence-Aware Recommender Systems. arXiv:1802.08452.
- Adomavicius, G., Tuzhilin, A. (2005) Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. IEEE Transactions on Knowledge & Data Engineering 6: 734-749.
- Mikolov, T., Chen, K., Corrado, G., Dean, J. (2013) Efficient estimation of word representations in vector space. arXiv preprint arXiv:1301.3781.
- Zolna, K., Romanski, B. (2016) user2vec: user modeling using LSTM networks. Proceedings of the 33rd International Conference on Machine Learning, New York, NY, USA.
- Vasile, F., Smirnova, E., Conneau, A. (2016) Meta-Prod2Vec - Product Embeddings Using Side-Information for Recommendation. arXiv preprint arXiv:1607.07326.
- Yan, Y. Liu, Z., Zhao, M. (2018) A Practical Deep Online Ranking System in E-commerce Recommendation.

Serve Customers

Presenter: Zuo-Jun (Max) Shen

Serve Customers

JD.COM

A transformed shopping experience driven by cutting-edge technologies in big data and operations research

Unlimited choices available online

Convenience

- Anywhere
- Anytime

Fast delivery

Serve Customers

JD.COM

Big data introduces new opportunities to better serve customers, as well as challenges to traditional solution methods

Unlimited choices available online

Convenience

Fast delivery

Challenges

Unlimited choices available online:
- Limited capacity at local warehouses
- Delivery speed
- Inventory placement
- …

Convenience:
- Local demand
- Inventory replenishment
- …

Fast delivery:
- Balance online and offline demand
- Omni-channel fulfillment
- …

Serve Customers

Inventory Placement

Inventory Replenishment

Order Fulfillment

Inventory Placement

JD’s nationwide convenience stores

Expanding nationally, especially in rural areas

Expected to reach 1M stores by 2023

Cater to local needs and support fulfilling online demand

Inventory Placement

Problem:

- How should inventory be allocated to JD's stores nationwide?

Goal:

- Delivery products to meet local needs
- Satisfactory fulfillment rate

Constraint:

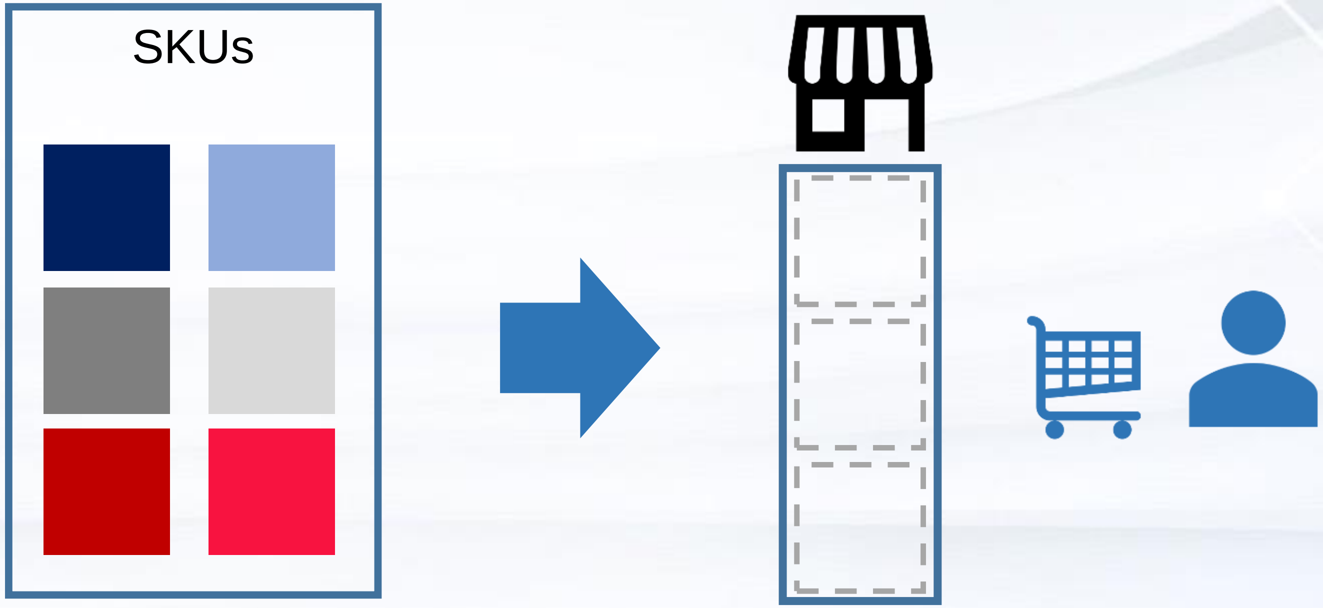
- Limited store capacity
- …

Inventory Placement – Offline Demand

JD's nationwide stores

Limited capacity per store

Only sellable if in-stock

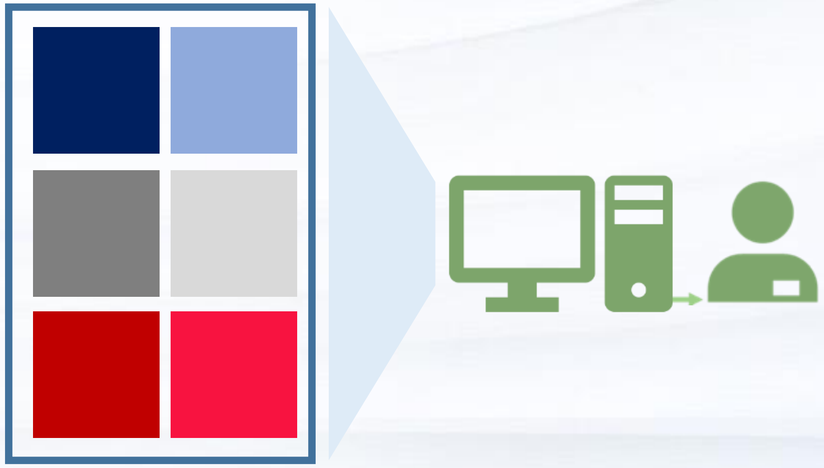


How to optimize profit and satisfy customer needs?

Inventory Placement – Online Demand

Limited assortment of SKUs at local stores …

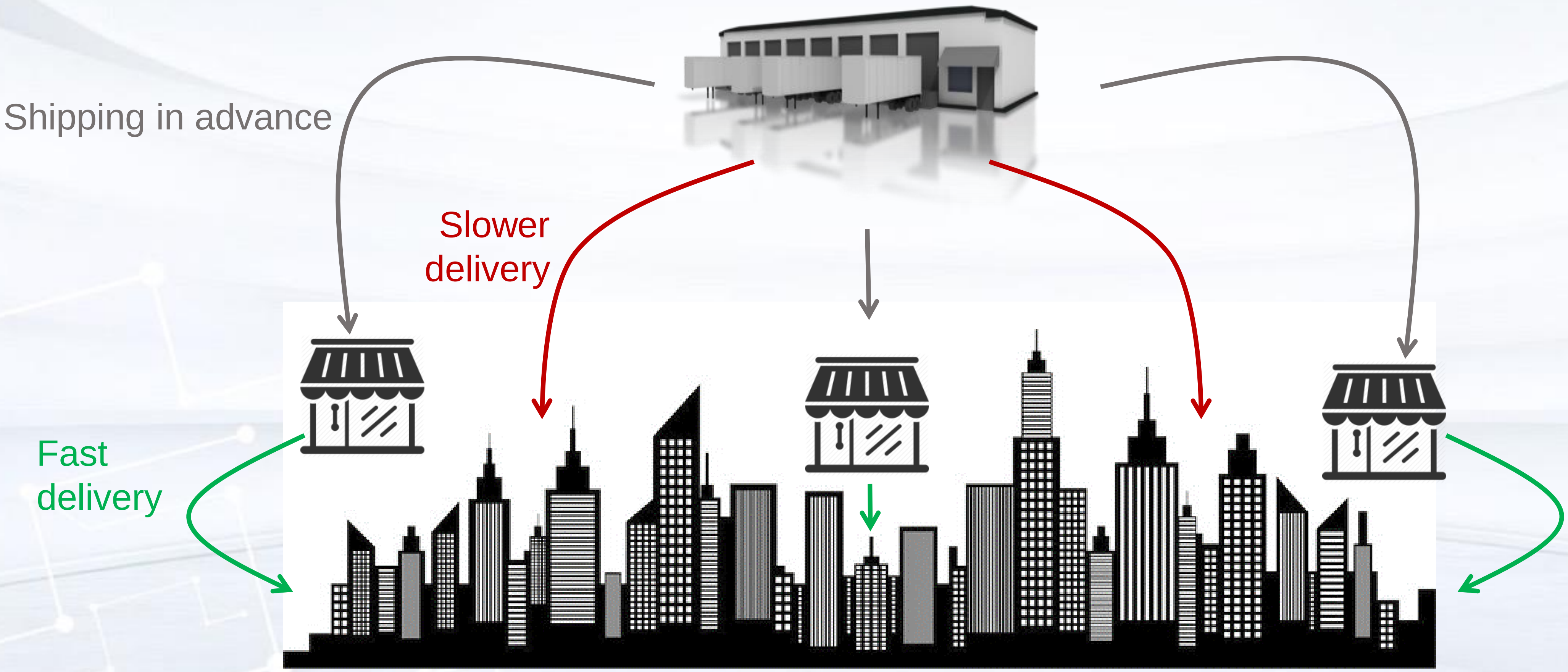
… while selection is unlimited online.



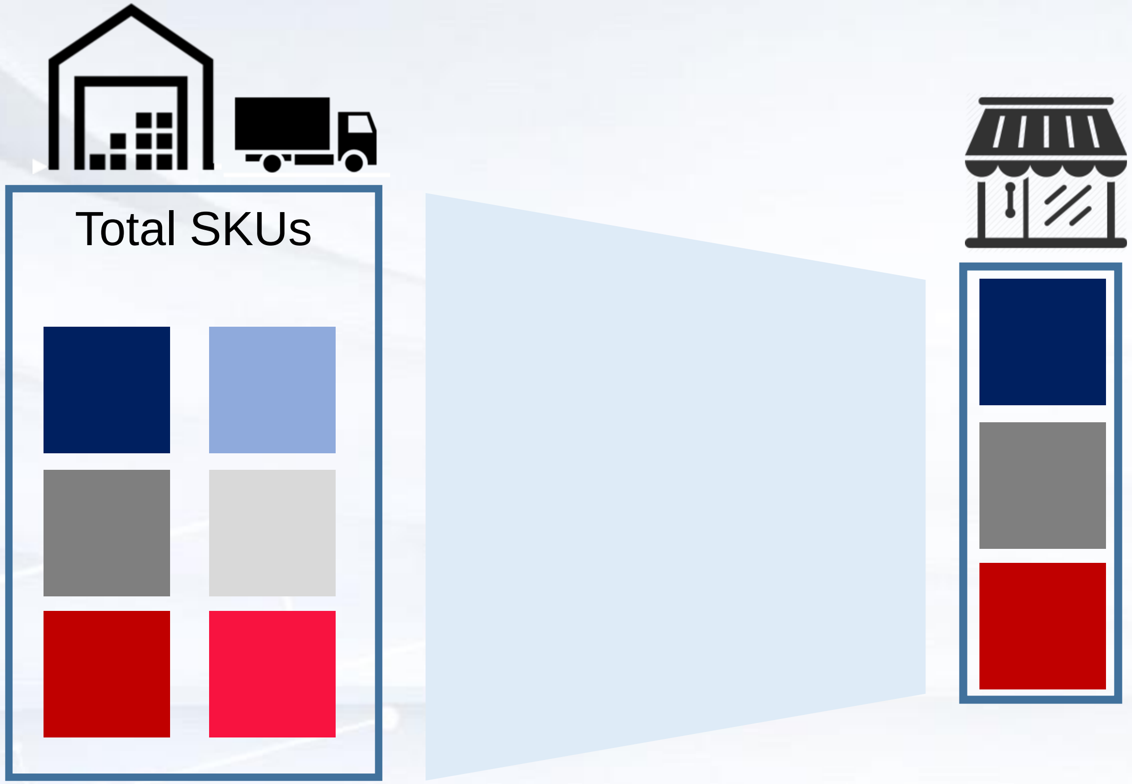
What SKUs to allocate to local stores to maximize order fulfillment?

Inventory Placement

What SKUs to allocate to local stores?



An Assortment Problem

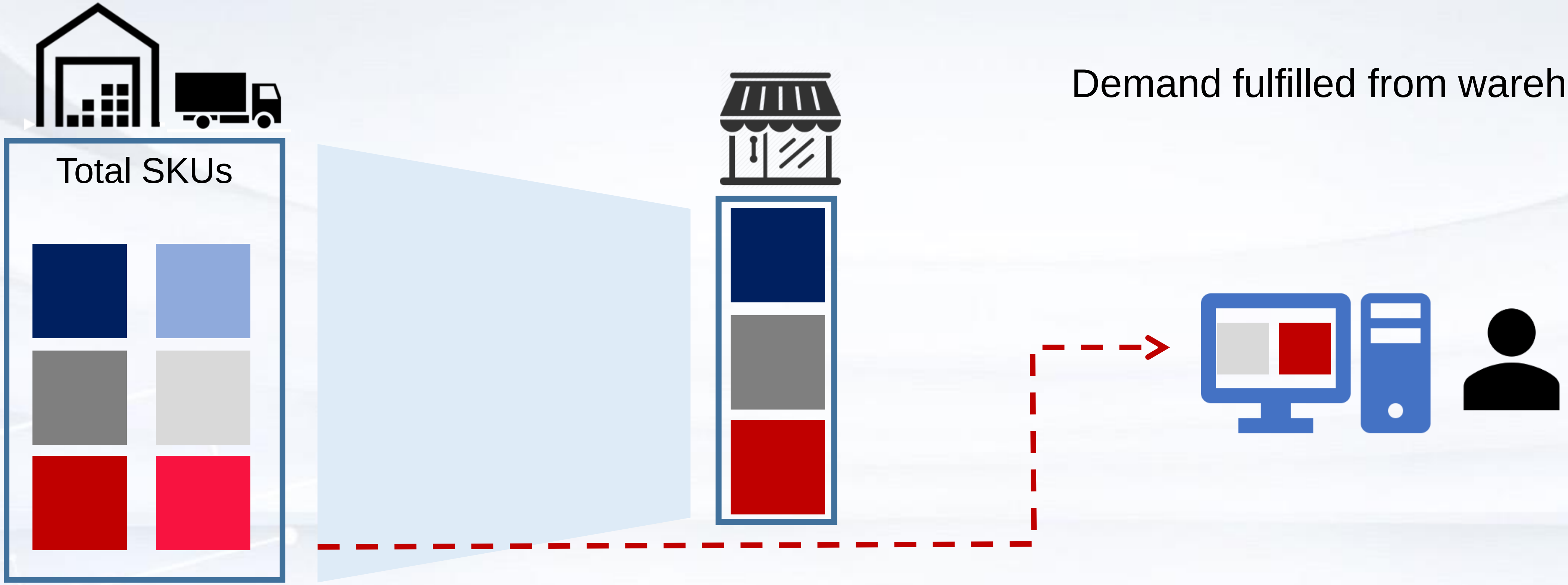


Demand fulfilled by local store

Local fulfillment enables expediated delivery that delights customers

An Assortment Problem

JD.COM

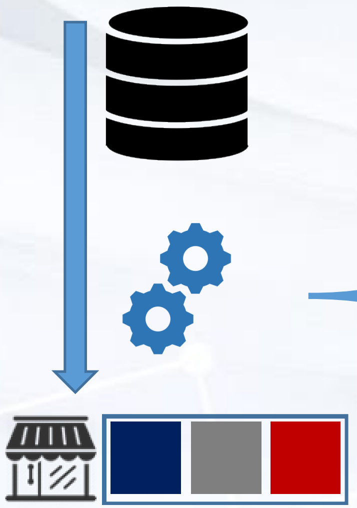


Longer delivery time and higher cost if fulfilled by remote warehouses

Assortment as a Classification Problem

$$\min \#LocallyMissedFutureOrders \quad s.t. \quad \#SKUInWarehouse \leq k$$

Past information

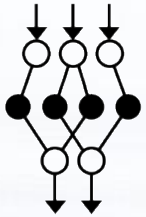


Serve future orders

Classify each SKU as in the assortment or out

$\Rightarrow$Train a supervised learning algorithm

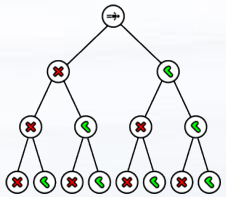
Artificial Neural Network



$\min \#LocalyMissedFutureOrders$

Non continuous loss ⇒ No gradient
Issues with enforcing constraint

Classifiers

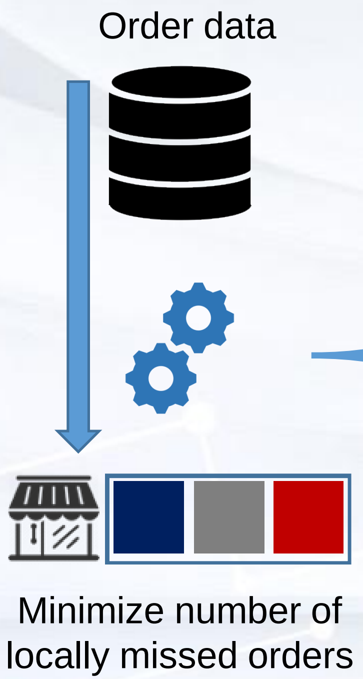


Needs training set to be labelled

To train a classifier we first need to label the training set

Labeling Training Set

$$\min \#LocallyMissedOrders \quad s.t. \quad \#SKUInWarehouse \leq k$$



Solve a deterministic discrete optimization problem

$\Longrightarrow$Reduces from k-densest graph problem

The Deterministic problem is NP-hard

Labeling the data is already a hard problem

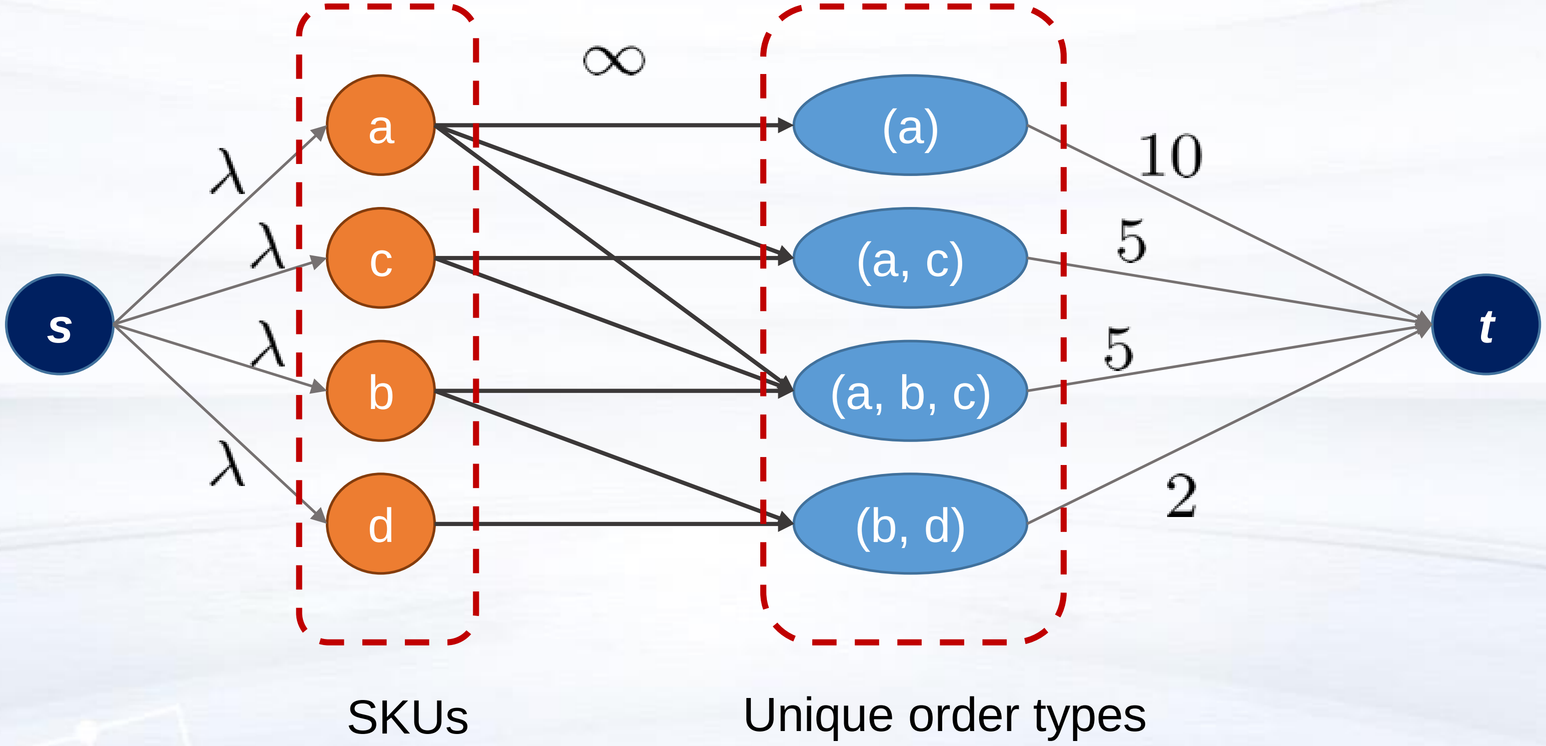
Bipartite Graph Representation of Orders

JD.COM

$$\min \#LocallyMissedOrders + \lambda * \#SKUInWarehouse$$

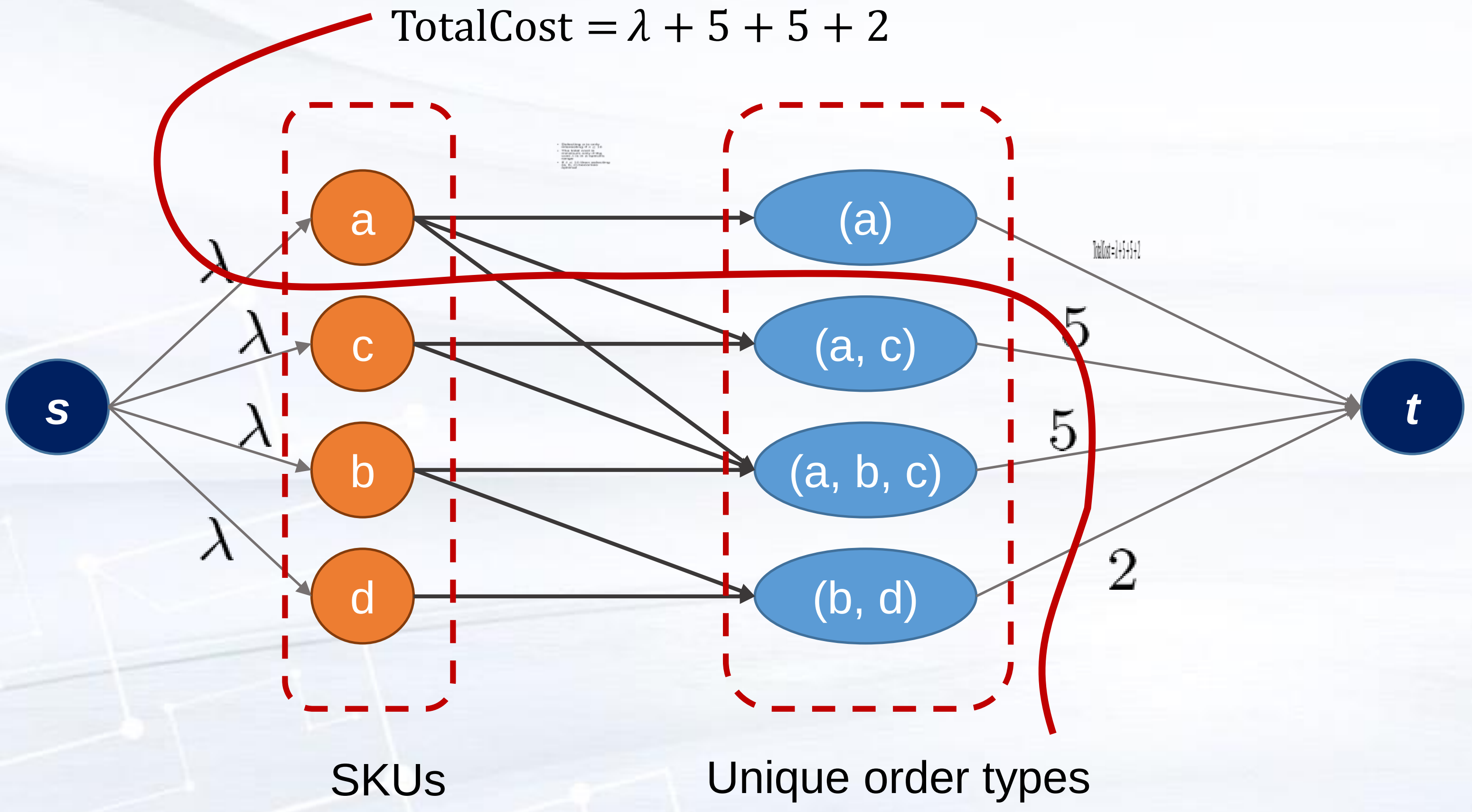
Order Set	
(a,b,c)	5
(a,c)	5
(b,d)	2
(a)	10

SKU Set
(a,b,c,d)



Bipartite Graph Representation of Orders

$$\text{TotalCost} = \min \#LocallyMissedOrders + \lambda * \#SKUInWarehouse$$

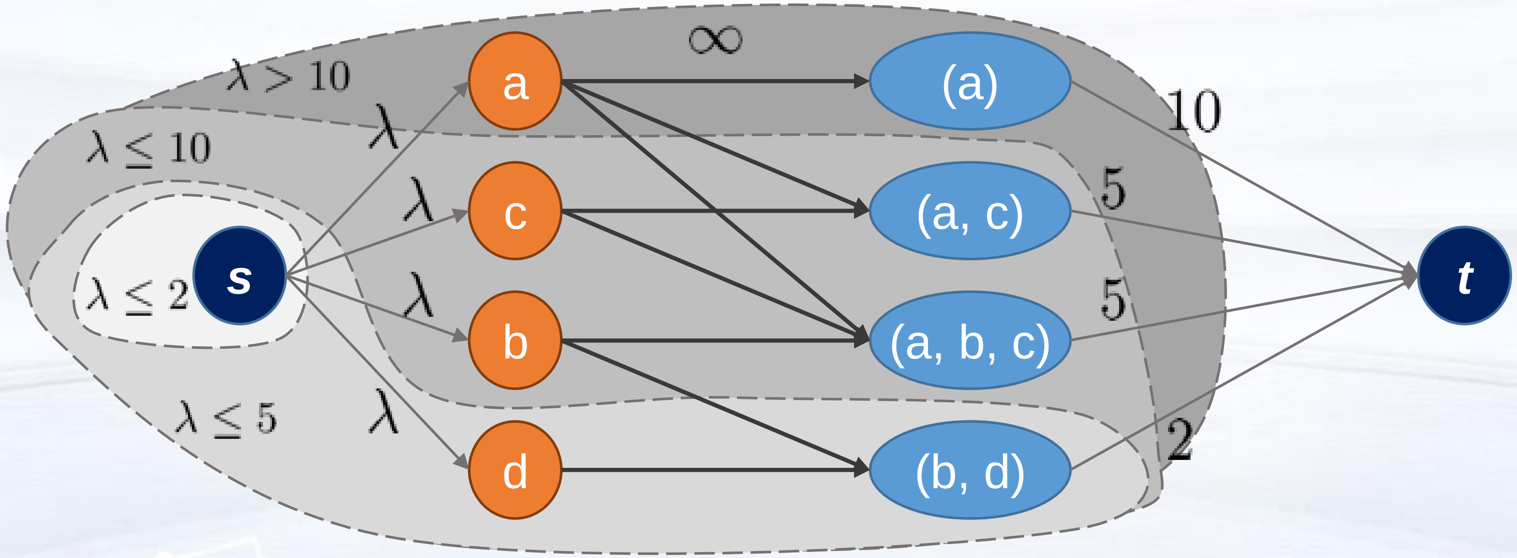


Example

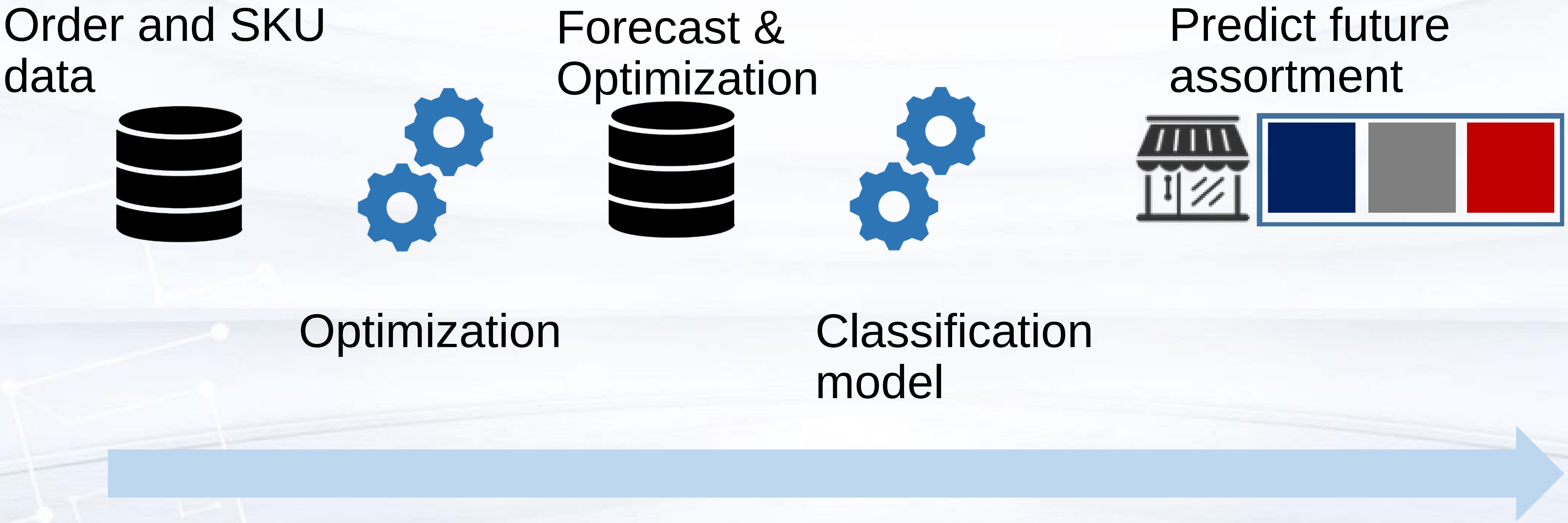
- Selecting if $\lambda > 10$
- If $\lambda \leq 10$ then selecting (a, b, c) becomes optimal

Parametric Cut Algorithm

- Using a graph method called Parametric Cut, we can efficiently identify every unique assortment that are optimal for a range of costs. The outputs are all NESTED assortments.
- We are not guaranteed to find an assortment of exactly the size we were looking for, however each assortment found is optimal for their cardinality.



Incorporate Future Demand



Inventory Placement

- Inventory placement
- Product assortment

Problem

Data

- Online & offline sales
- Demand forecast

- What SKUs to be carried by each store

Solution

Serve Customers

Inventory Placement

Inventory Replenishment

Order Fulfillment

Inventory Replenishment

JD.COM

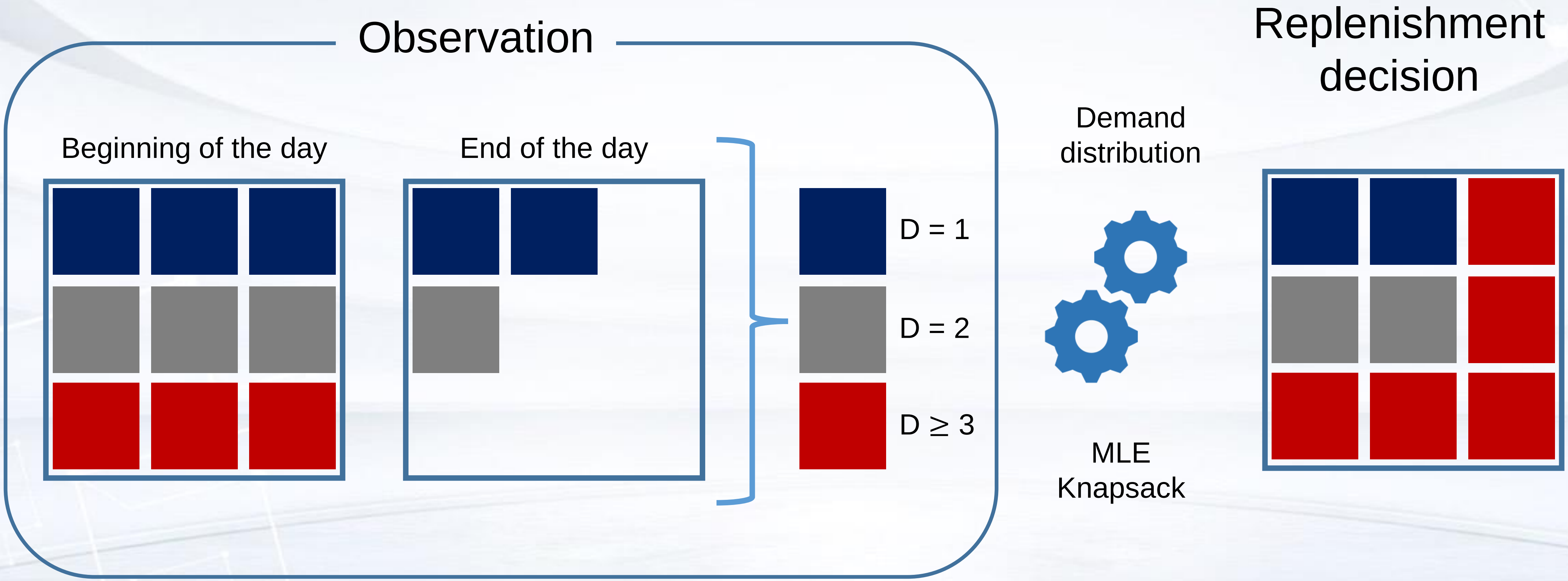
Smart vending machine

- Flexible shelf-space sharing
- Mobile log-in and payment
- Frequent replenishment
- Demand-driven selection

Challenge

- Demand uncertainty
- Censored sales

A Replenishment Problem

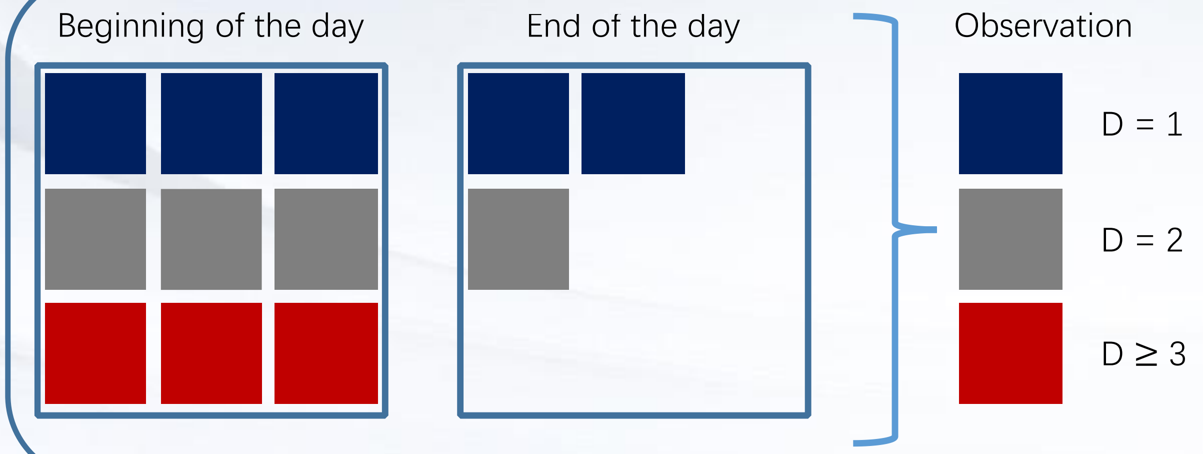


This differs from the multi-armed bandit problem as the reward yielded by 2 units of the same product is not independent

Demand Estimation with MLE

JD.COM

Observation



Assumptions

- Demand is independently distributed
- θ is the set of parameters of the distribution
- $\mathcal{L}(\theta;\ obs)$ is the likelihood function of θ given the observation obs

- $M[i,j]$ is the matrix that maps i and j to the probability that product i is bought more than j times during the next cycle

$$\mathcal{L}(\theta;\ obs) = P(D_{blue} = 1;\ \theta) \cdot P(D_{grey} = 2;\ \theta) \cdot P(D_{red} \geq 3;\ \theta)$$

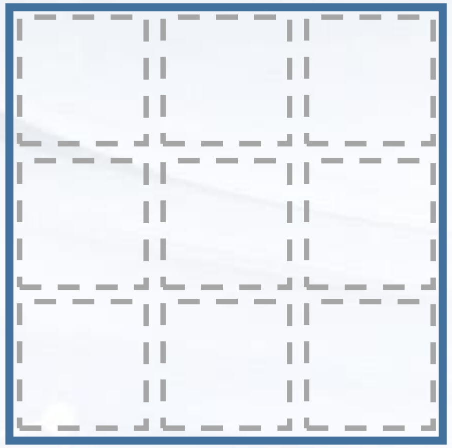
$$\mathcal{L}(\theta;\ obs) = \prod_{x_i\ uncensored} P(D_i = x_i;\ \theta) \cdot \prod_{x_j\ censored} P\big(D_j \geq x_j;\ \theta\big)$$

$$\hat{\theta} \in \{\arg\max \mathcal{L}(\theta;\ obs)\} \rightarrow M[i,j] = P(D_i \geq j; \hat{\theta})$$

Use Knapsack to Solve for Inventory Level

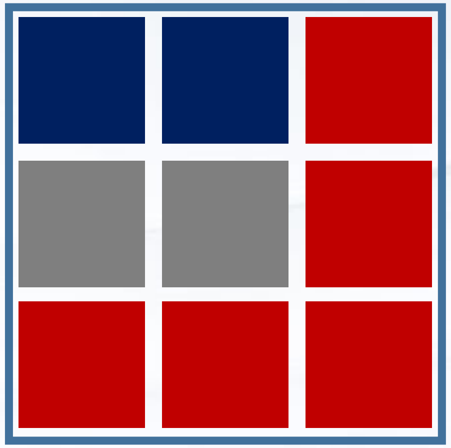
JD.COM

C *capacity of the container*



$$Max \sum_i \sum_j r_i \, M[i,j] \, X_i^j$$
$$s.t. \sum_i \sum_j v_i \, X_i^j \le C$$

Next day inventory level



Decision Variables

$X_i^j = 1$ *if at least j units of product i are selected*

Parameters

v_i *volume of one unit of product i*
r_i *revenue yielded by one unit of product i sold*
C *capacity of the container*

Maximize expected revenue under capacity constraint

Inventory Replenishment

Problem

- Inventory replenishment

Data

- Censored sales
- Demand distribution
- Sales imputation

Solution

- When and how many SKUs to be restocked

Serve Customers

Inventory Placement

Inventory Replenishment

Order Fulfillment

Order Fulfillment

Pros:

- Easy to manage

Cons:

- Inflexibility
- Limited products per FC/store

Need a more flexible fulfillment system

Order Fulfillment – Omni-Channel

Objective:

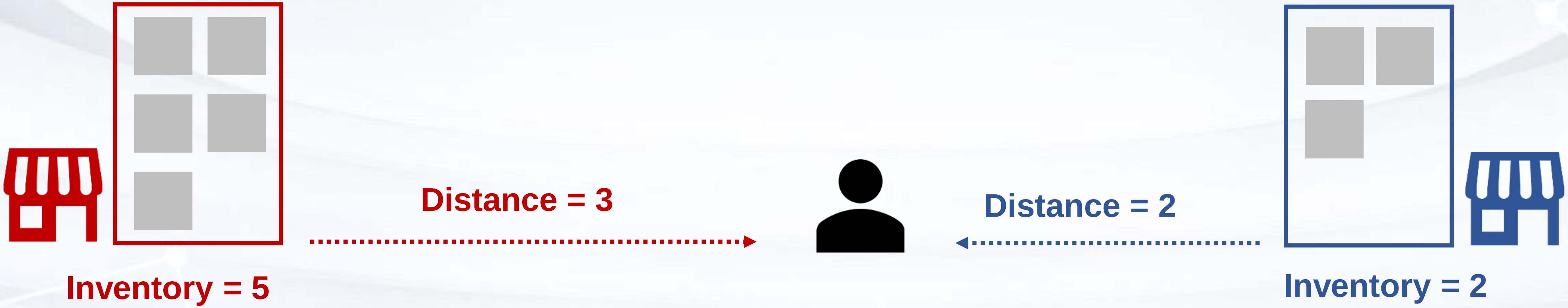
- Max delivery utilization
- Min cost of delivery

Constraint:

- Delivery deadline
- Inventory availability
- Delivery capacity
- …

Orders can be fulfilled more efficiently across FDC/stores

A Fulfillment Problem (1/2)

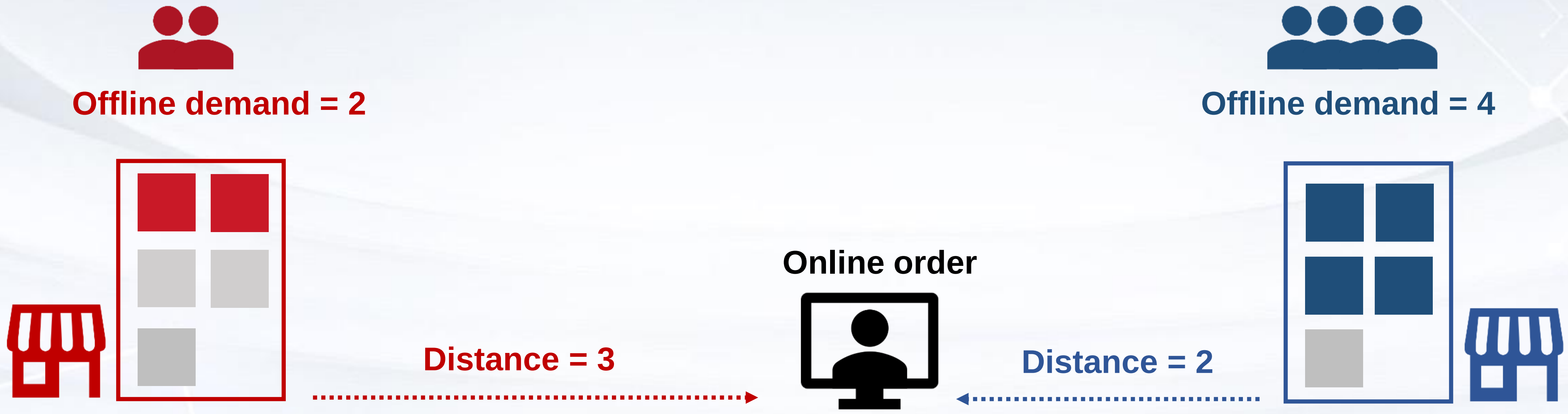


- Higher delivery cost
- Lower stock-out probability

- Lower delivery cost
- Higher stock-out probability

How to best fulfill demand while balancing cost of delivery and loss of sales?

A Fulfillment Problem (2/2)



- Higher delivery cost
- Lower demand offline

- Lower delivery cost
- Higher demand offline

How to design an optimal inventory holding policy?

Order Fulfillment

JD.COM

Problem

- Fulfill online and offline demand

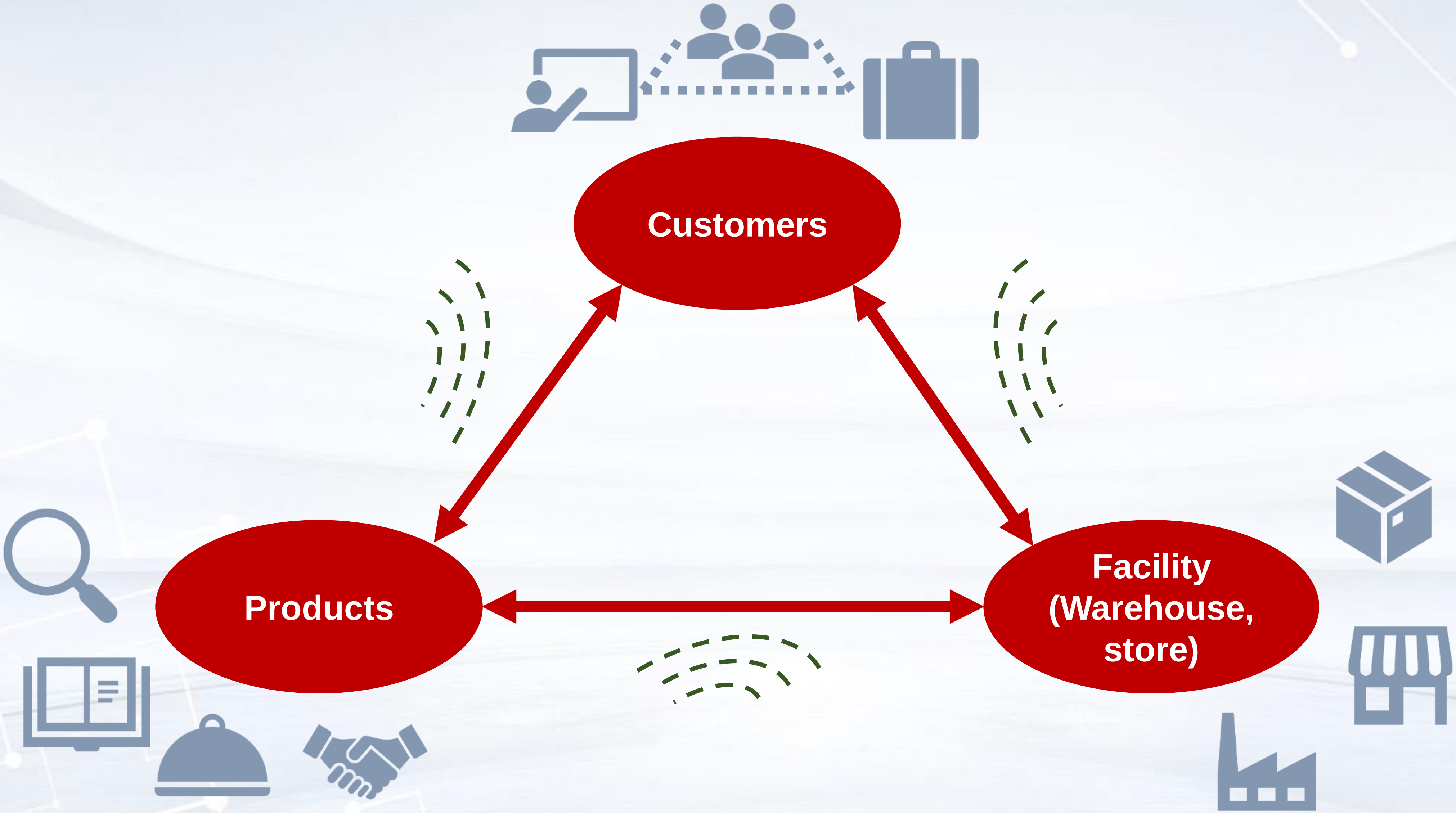
Data

- Online and offline sales
- Real-time inventory data

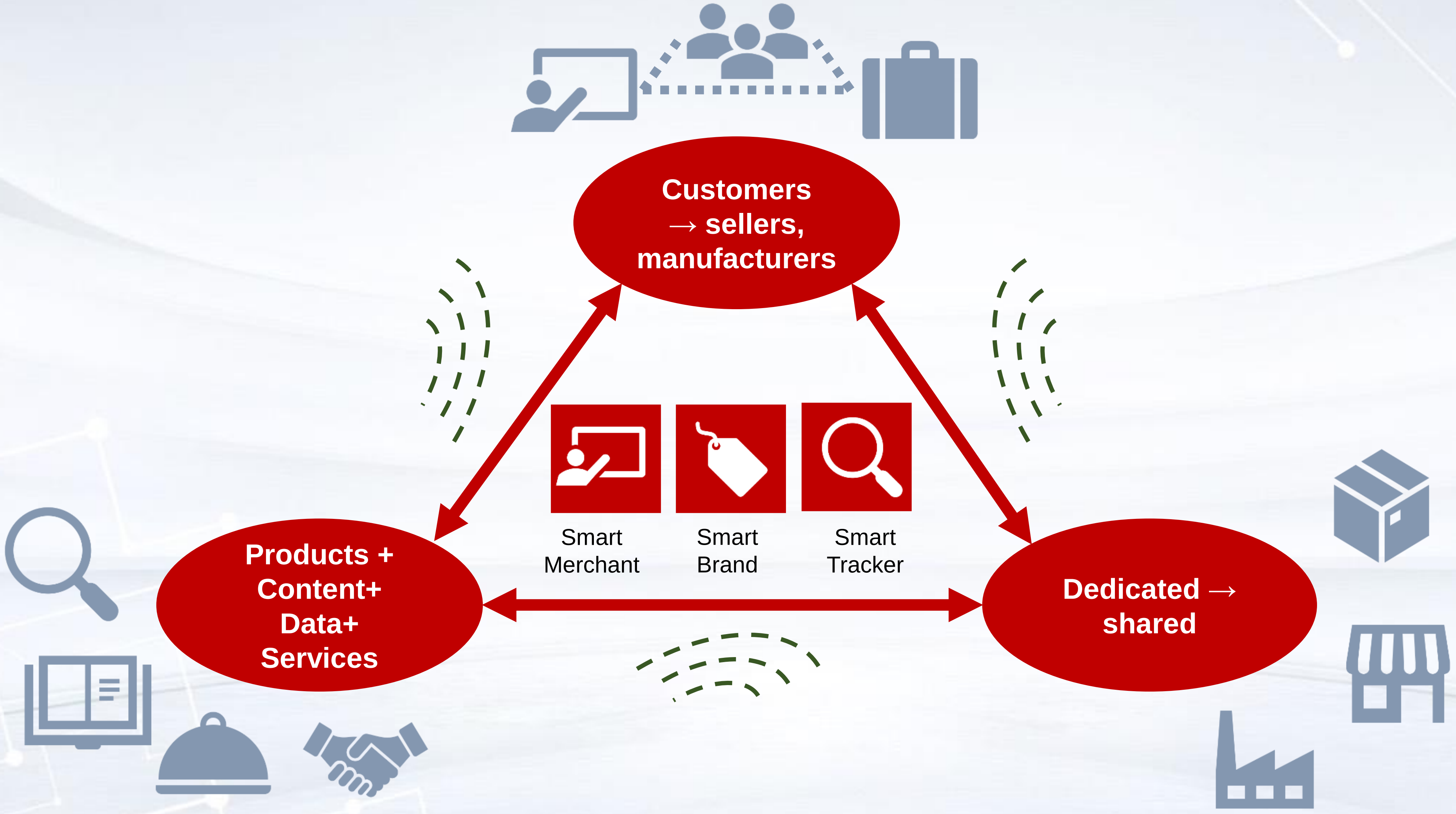
Solution

- Which store to fulfill an online order

Traditional Retailing



Data-Driven Retail-as-a-Service



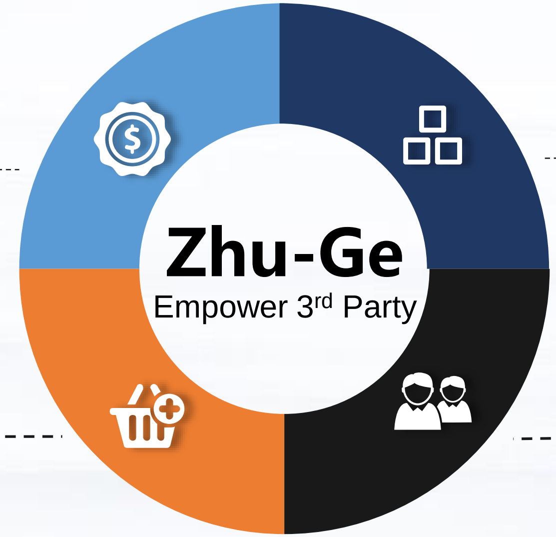
Smart Merchant - Zhu-Ge

Price Master

Support POP merchants with extensive pricing and market analysis

Smart Inventory

Optimize inventory decision based on big data and Operations Research



Assortment Guru

Explore market opportunities and industrial trends to boost sales volume

Marketing Advisor

Monitor market trend, public opinions on social media to improve PR

Smart Brands - Retail Intelligence Lab

JD.COM

Collaborative Forecasting

Sales Diagnose

Assortment

Pricing

Inventory Management

Analysis ▶ Prediction ▶ Decision

Business Management Platform

AI Open Platform Based on Business Scenarios

TOP Partners

P&G

Unilever

MARS

Improve supply chain efficiency based on AI and field experience;

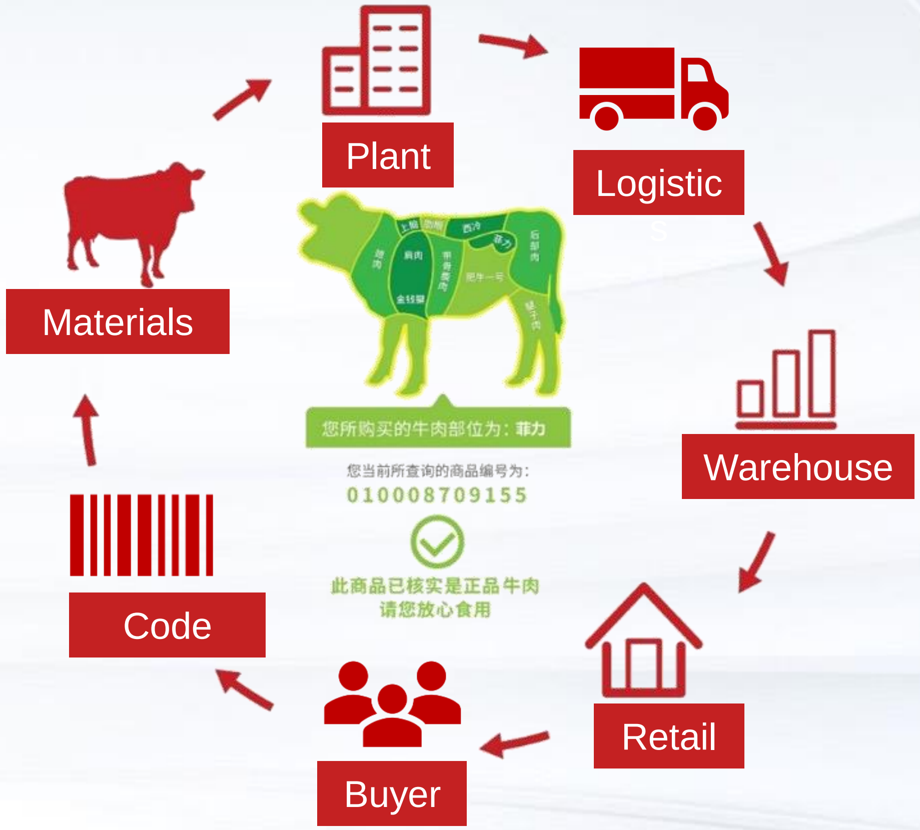
Uncover user insights to improve product designs and assortment

Smart Tracker Based on Block-Chain

JD.COM

56 Top Brands

1 million+ QR Code Scans



To establish a new traceable E-commerce business based on Block-Chain

Smart Tracker Based on Block-Chain

JD.COM

该追溯码已被查询 72 次，第一次查询时间：2018-03-09 19:29:17

全程品质追溯

京东区块链黑科技神器

您购买的产品是

京造 泰国天然乳胶枕头 波浪曲线枕芯 93%乳胶含量

2018-03-09 19:29:17

3446e261ca35110725348c271cd682efb3f892623c9a2215f4c9aa74320b6

区块链数字证书

371

2018年04月10日 22:44:28

3002500000001003

产品信息

产品名称：	泰国天然乳胶枕头
产地：	泰国
产品批次：	280036
保质期：	2020年06月30日
存储条件：	这里填写存储条件

*以上信息由 京造供应商 提供

生产商信息

生产商名称：	泰国橡泰集团(THAILAND THAI BURRER LATEX CO LET.)
生产商注册号：	A0A002

*以上信息由 京造供应商 提供

生产信息

橡胶林地址：	Thai Rubber Latex Corporation (Thailand) 29 Moo2, Ban Bueng - Klaeng road, Km. 56-57。Nong Yai district, Chonburi 20190
采收时间：	2017-08-01
搅拌时间：	2017-08-01
成型时间：	2017-08-03
检验时间：	2017-08-04
打包时间：	2017-08-10
发货时间：	2017-09-05

*以上信息由 京造供应商 提供

通关信息

出口许可证：	12345
原产地证：	12345
进口货物报关单：	12345

*以上信息由 京造供应商 提供

检验仓信息

入库检验仓：	检验仓名称
二次检验时间：	2018-02-01
打包时间：	2018-03-01
发货时间：	2018-03-23

*以上信息由 京造供应商 提供

Product Info

Manufacture Info

Trace Info

Declaration Info

Inspection Info

Block-chain technology enables customers to trace all phases from manufacturing to delivery of a product

Case Study: 7 Fresh Integrated Service

7Fresh

JD.COM

Grocery	Delivery	Dining	Integrated Data

7FRESH
FOOD MARKET

7Fresh-Grocery

JD.COM

Dynamic Pricing based on

- Product life cycle
- Inventory level
- Customer demand

Product Tracing

- Block chain
- Recognition technology
- Additional information

- origin,
- freshness
- cooking tips
- …

7Fresh-Delivery

JD.COM

7FRESH
FOOD MARKET

Omni-channel
Fulfillment

30-Minute
Ultra Fast Delivery

Online Offline Data
Integrated

7Fresh-Value-Added Service

JD.COM

7FRESH
FOOD MARKET

+

=

Fresh Ingredients

Cooking Service

7Fresh

JD.COM

Grocery

Delivery

Dining

Integrated Data

Facial recognition

Shopping path tracking

Online offline data integration

Gravity sensing shelf

7Fresh-Integrated Data

JD.COM

Shopping path tracking

Facial recognition

Online offline data

Gravity sensing shelf

Enriched Data Set

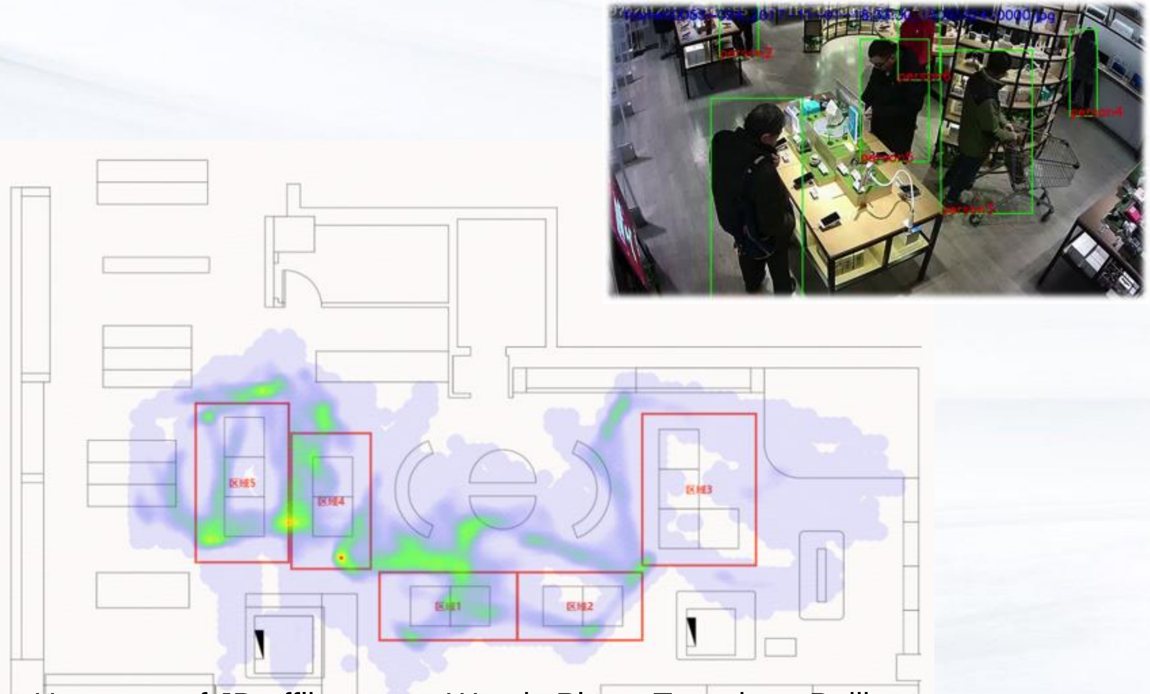
Customer profile | Customer behavior | Real-time inventory | …

7Fresh-Integrated Data

Face Recognition matches online user accounts with offline shoppers

7Fresh-Integrated Data

Shopping Path Tracking System tracks and records consumer movement within store; enables spatial-temporal data analysis



Heat map of JD offline store, Wanda Plaza, Tongzhou, Beijing

Consumer movement

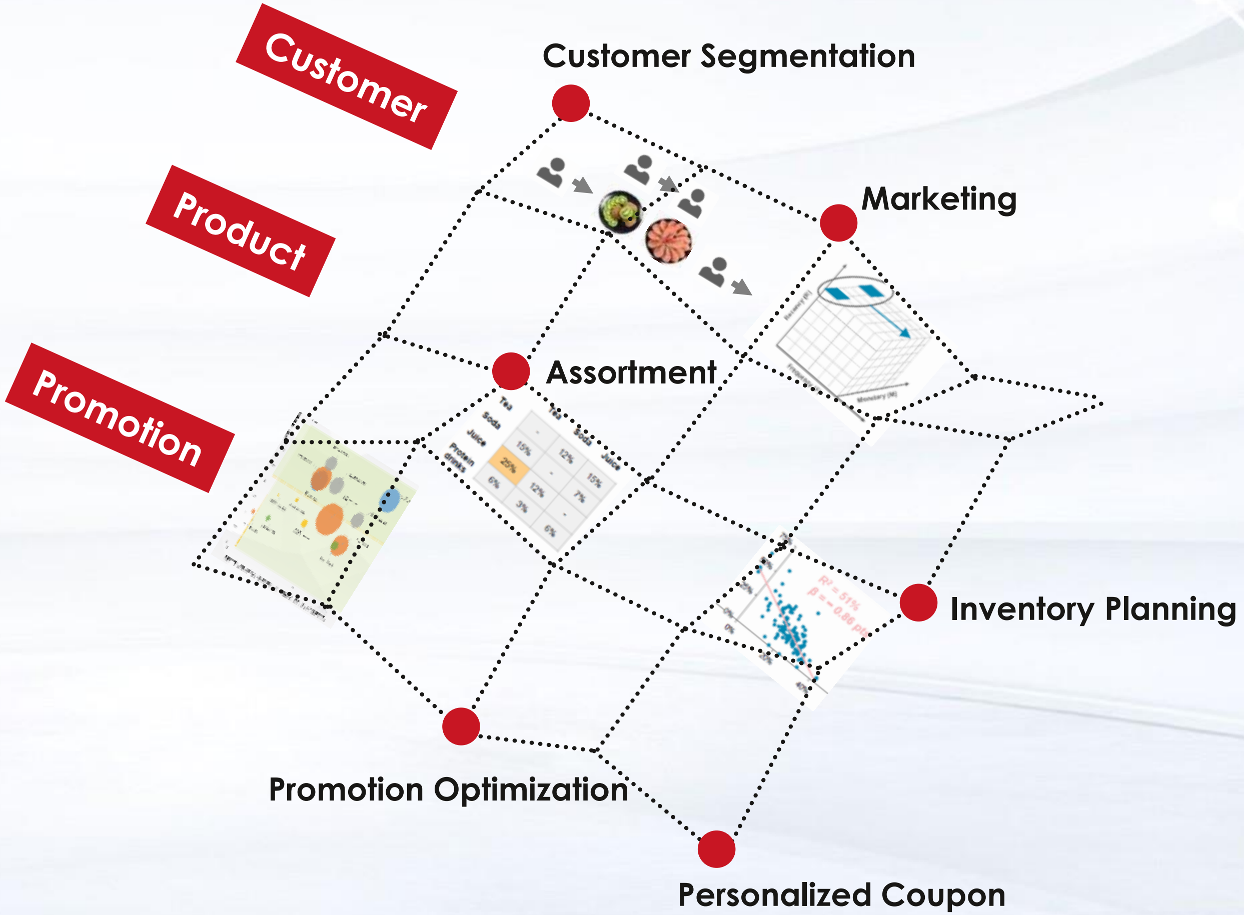
- Heatmap
- Popular shelf
- Hot movement path
- Regional population

7Fresh-Integrated Data

Online-Offline Data Integration: leveraging offline in-store data to inform online marketing efforts and personalize promotions

- 50% of offline customers are active online on JD.com

Personal coupons



7Fresh-Integrated Data

Online-Offline Data Integration: channels online customer shopping data to enhance offline shopping experience

Smart Shopping Cart

Coupon usable also offline

- Extensive online data enables data-driven localized assortment and highly personalized marketing from Day 1 of store opening

7Fresh-Integrated Data

Gravity Sensing Shelf : generates real-time inventory information to improve customer satisfactory and operational efficiency

- Real-time inventory
- Number of times a product got lifted
- Shelf placement

7Fresh

- Ultimate grocery store that breaks the boundary of online and offline shopping
- Offline store supported by latest retailing technologies
- Instore dining
- 30min delivery service to nearby areas

7Fresh Summary

JD.COM

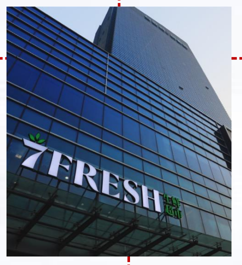
Quality Grocery | **Ultra Fast Delivery** | **Value-added Service**

- Data driven supply chain (customer profile, customer behavior, real-time data)
- Smart multi-connection to customers
- Flexible replenishment and fulfillment system

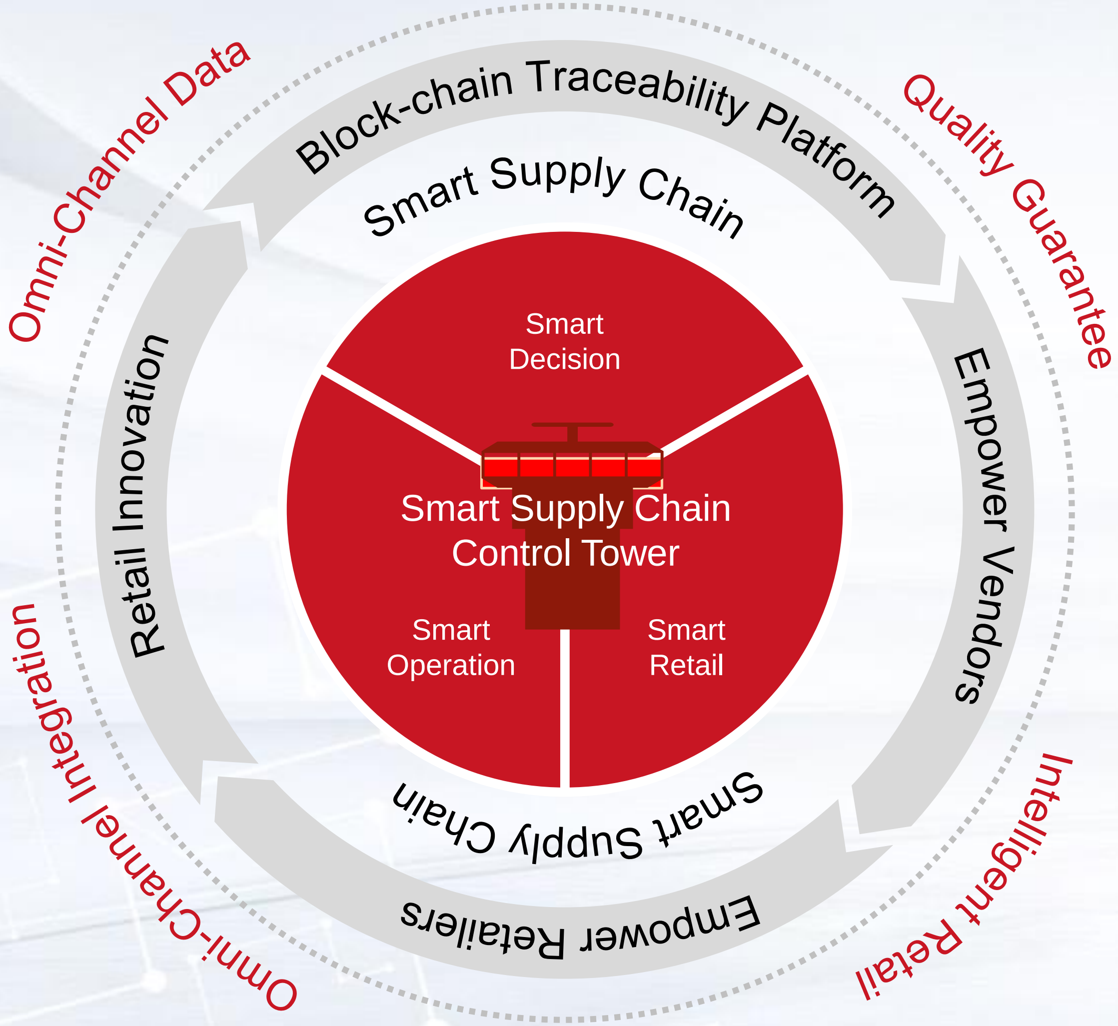
Understand Customers

Connect to Customers

Serve Customers



Data Science



Omni-Channel Data
Block-chain Traceability Platform
Quality Guarantee
Smart Supply Chain
Smart Decision
Smart Supply Chain Control Tower
Smart Operation
Smart Retail
Retail Innovation
Empower Vendors
Smart Supply Chain
Empower Retailers
Omni-Channel Integration
Intelligent Retail

Q&A