Dynamic Pricing with Online Data Aggregation and Learning

Gianmarco Genalti[1], Marco Mussi[1], Alessandro Nuara[2] and Nicola Gatti[1]

[1] Politecnico di Milano [2] ML cube

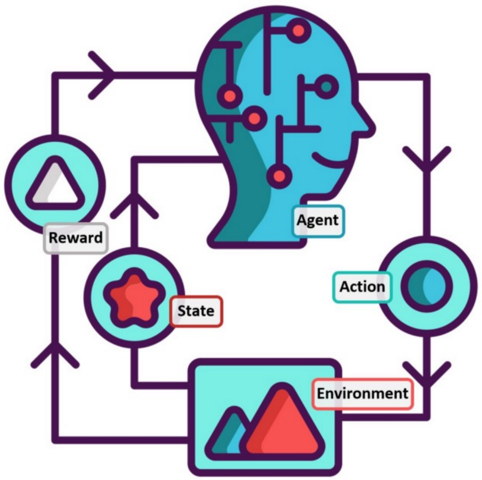
15th European Workshop on Reinforcement Learning, Milan

Sequential decision processes in real-world applications

Advertising optimization

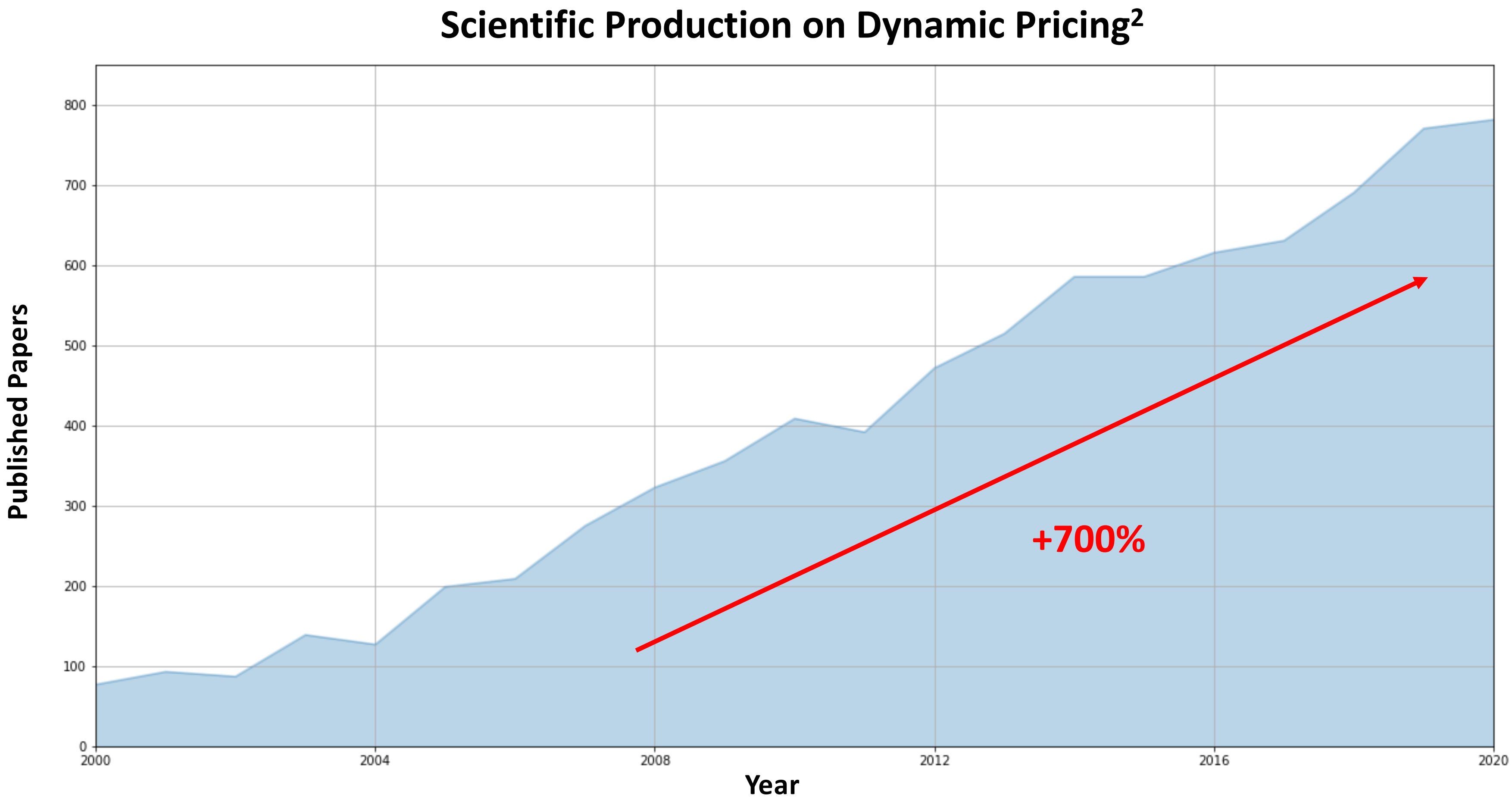
Dynamic pricing

Content recommendation



Medical trials

Dynamic Pricing in Research

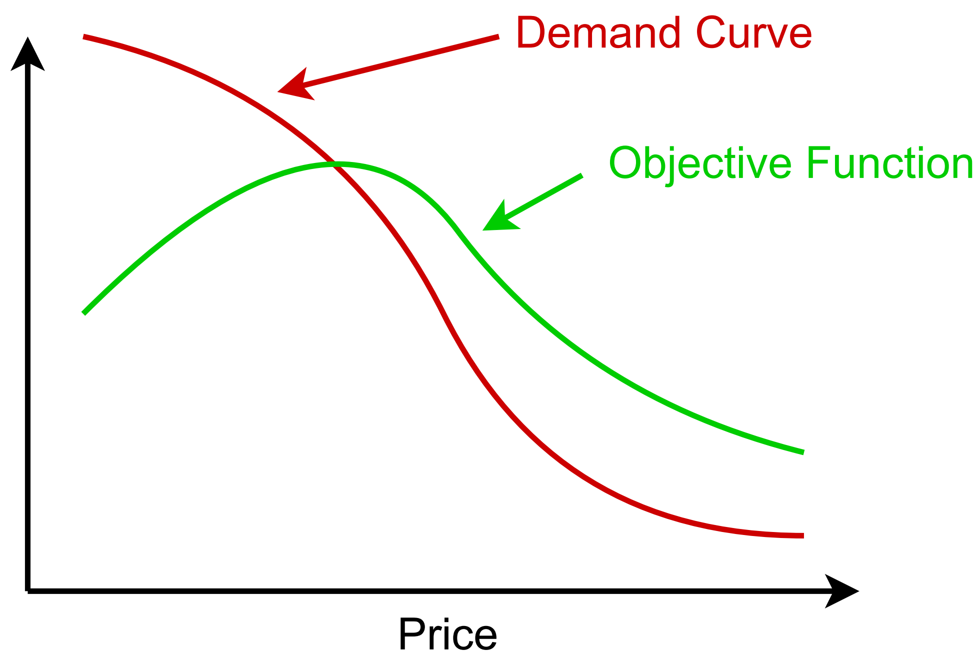


[2] www.scopus.com

Dynamic pricing

Consider an e-commerce which sells a product

- Customers **visit** the product page and **decide** whether to buy or not
- By aggregating users choices, we are able to build a **demand curve**



[3] Arnoud V Den Boer. Dynamic pricing and learning: historical origins, current research, and new directions. Surveys in operations research and management science, 20(1):1–18, 201

The Long Tail Paradigm

Real-World scenario:

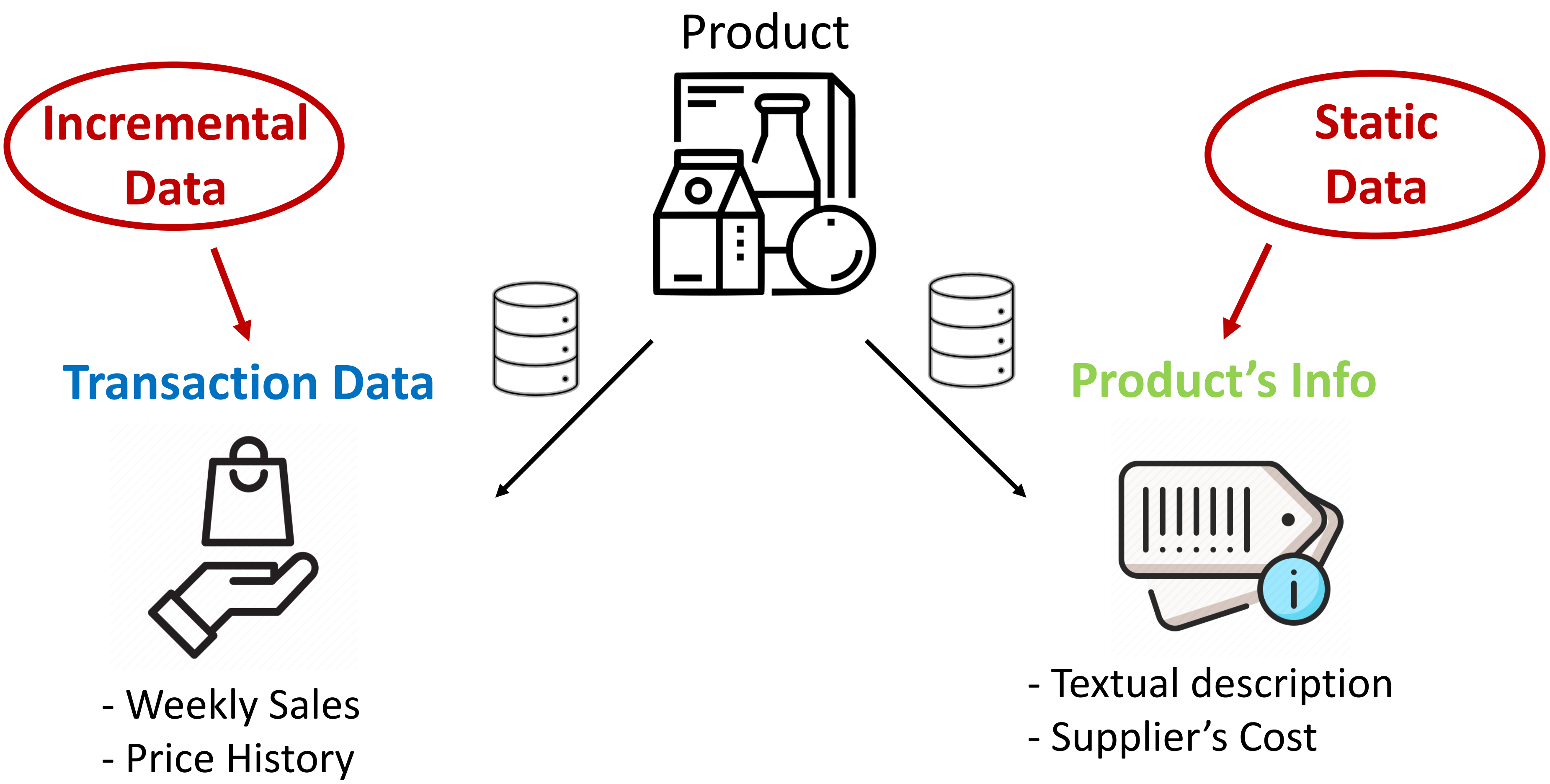
≈ **20000** products in total

- ≈ **1000** best sellers
- ≈ **12000** long tail
- ≈ **7000** never sold before

[4] Chris Anderson. The long tail: Why the future of business is selling less of more. Hachette Books, 2006

Setting and Goal

Available Data



Problem Formulation

- The e-commerce sells N products with **unlimited availability**

- At every time t, for every product j, we aim to set a **percentage margin** m_{jt} defined as

$$m_{jt} := \frac{p_{jt} - c_j}{c_j}$$

  - p_{jt} is the selling price of product j at time t
  - c_i is the acquisition cost for product j

Objective Function

Goal: Maximize *total profit*

$$f_{jt}(m_{jt}) \coloneqq m_{jt}\, c_j\, v_{jt}(m_{jt})$$

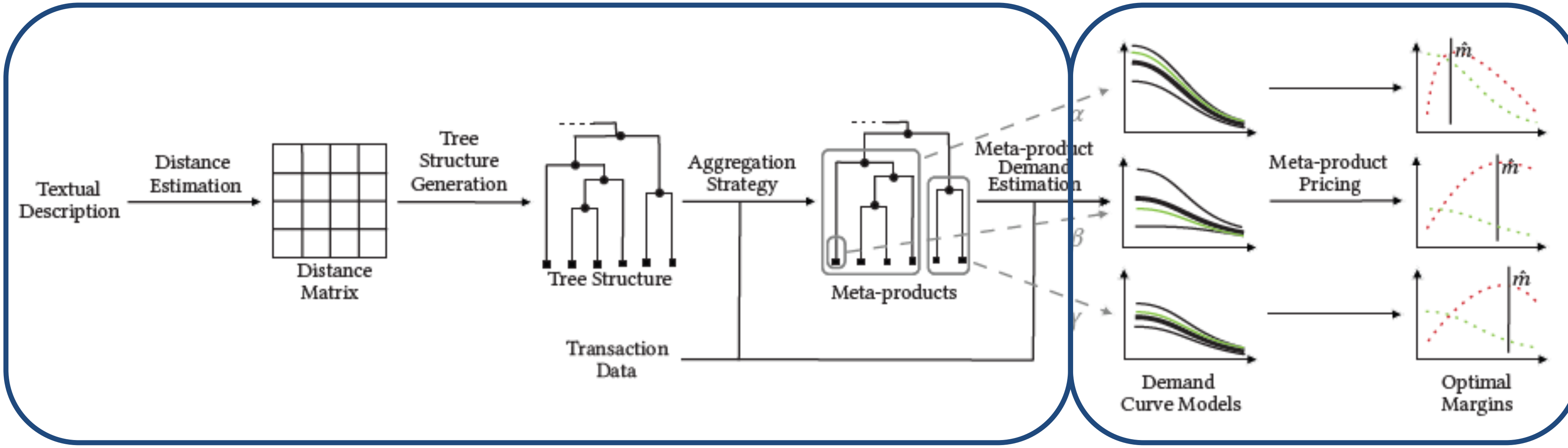
- $v_{jt}(m_{jt})$ is the **unknown** number of units products j would sold at time t by setting the margin as m_{jt}

This implies find the **optimal pricing strategy**:

$$\boldsymbol{m}^* \coloneqq \underset{\{\boldsymbol{m}_1, \boldsymbol{m}_2, \ldots\}}{\operatorname{argmax}} \sum_{t=1}^{T} \sum_{j=1}^{N} f_{jt}(m_{jt})$$

Solution

Algorithm - Overview



Data Aggregation

Online Optimization Step

Demand Curve Model

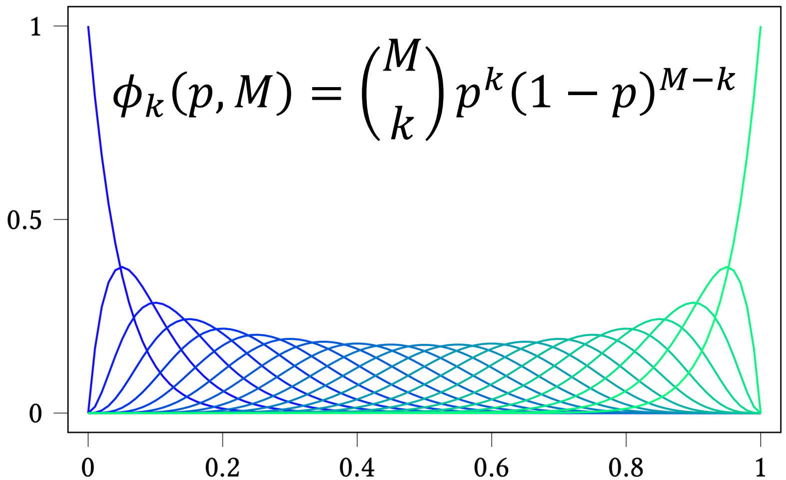
- To estimate the volume curve v_i we resort to *Bayesian Linear Regression* (BLR) with a basis function expansion:

$$\hat{v}_{jt}(m, \boldsymbol{w}_{jt}) = w_{0,\mathrm{jt}} + \sum_{i=1}^{M-1} w_{i,jt}\phi_i(m)$$

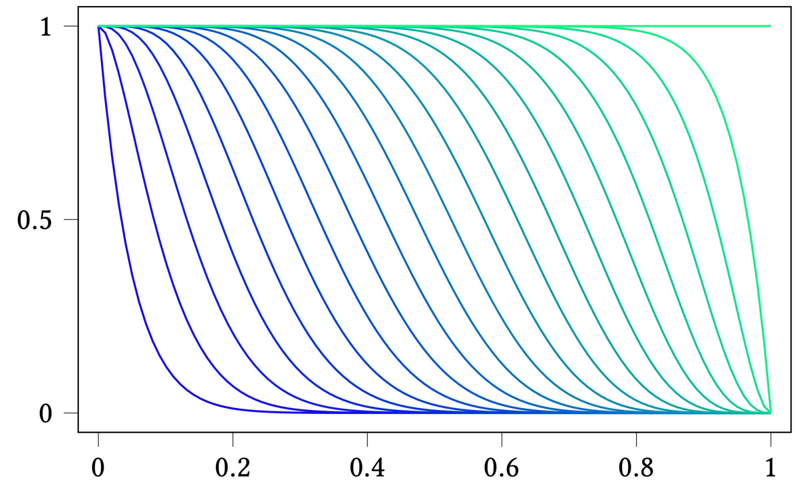
- We assume that the true volume curve is **non-increasing in price**. This assumption:
  - Is realistic in our setting since this goods are **non-luxury**
  - Alleviates the problem of **learning from noisy samples**

Bernstein Polynomials

Price values are expanded using transformed **Bernstein Polynomial**



Bernstein Polynomials



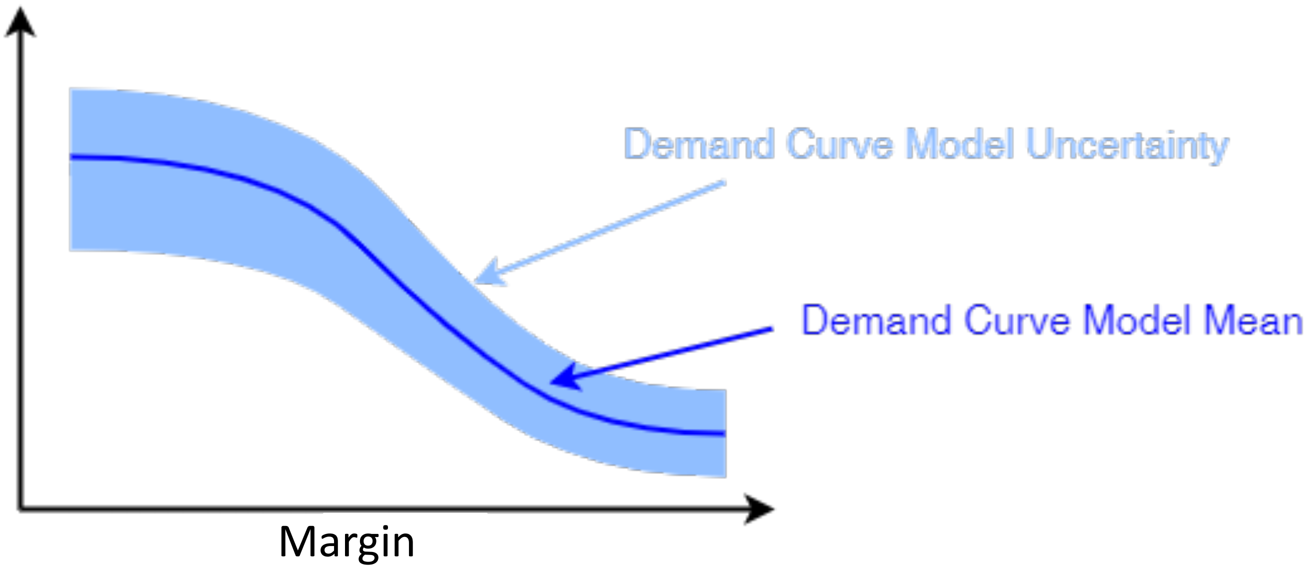
Transformed Bernstein Polynomials Expansion

$\boldsymbol{w} \sim \mathcal{LN}(\mu, \sigma^2)$

We select a **positive support prior** distribution of BLR

Dealing with Price Exploration

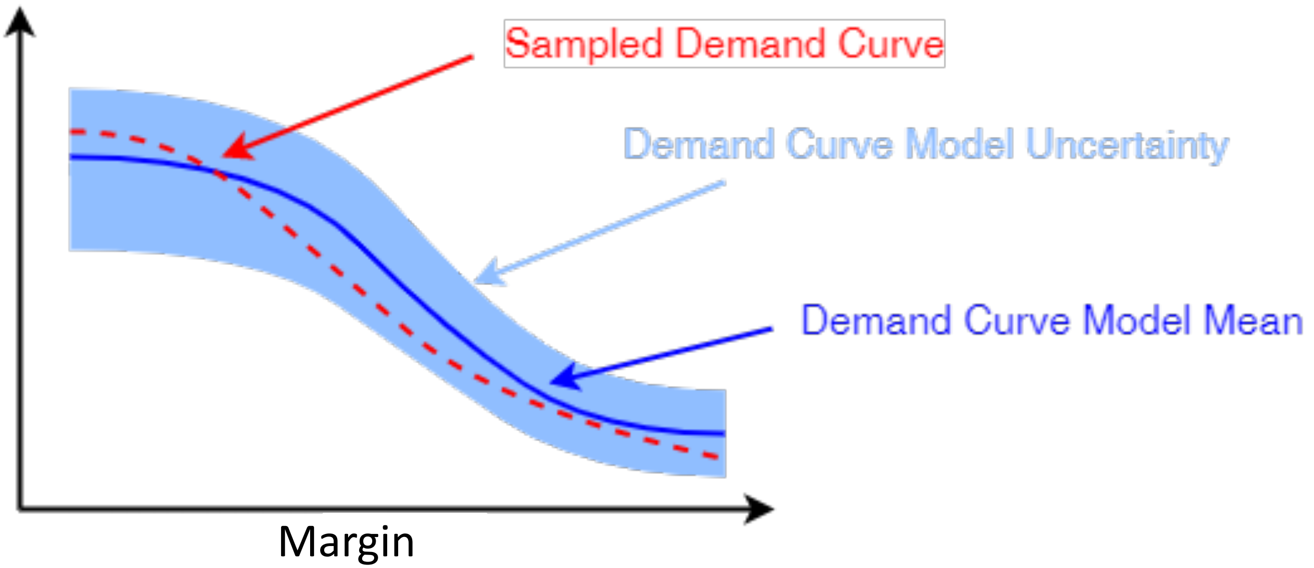
- From weights' posterior distribution we sample[3] a single curve $\hat{v}_{jt}(\cdot, \tilde{\boldsymbol{w}}_{jt})$
- Objective function $\hat{v}_{jt}(\cdot, \tilde{\boldsymbol{w}}_{jt})$ is then computed over the range of possible margins



[5] Nuara, Alessandro, et al. "A combinatorial-bandit algorithm for the online joint bid/budget optimization of pay-per-click advertising campaigns." *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 32. No. 1. 2018.

Dealing with Price Exploration

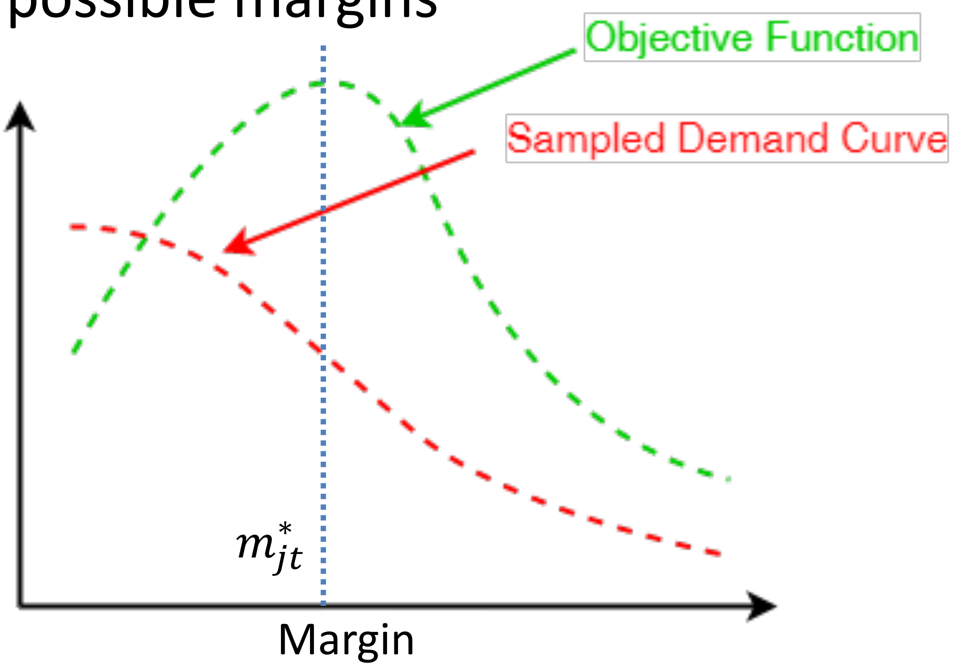
- From weights' posterior distribution we sample[3] a single curve $\hat{v}_{jt}(\cdot, \tilde{\boldsymbol{w}}_{jt})$
- Objective function $\hat{v}_{jt}(\cdot, \tilde{\boldsymbol{w}}_{jt})$ is then computed over the range of possible margins



[5] Nuara, Alessandro, et al. "A combinatorial-bandit algorithm for the online joint bid/budget optimization of pay-per-click advertising campaigns." *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 32. No. 1. 2018.

Dealing with Price Exploration

- From weights' posterior distribution we sample[3] a single curve $\hat{v}_{jt}(\cdot, \tilde{\boldsymbol{w}}_{jt})$
- Objective function $\hat{v}_{jt}(\cdot, \tilde{\boldsymbol{w}}_{jt})$ is then computed over the range of possible margins

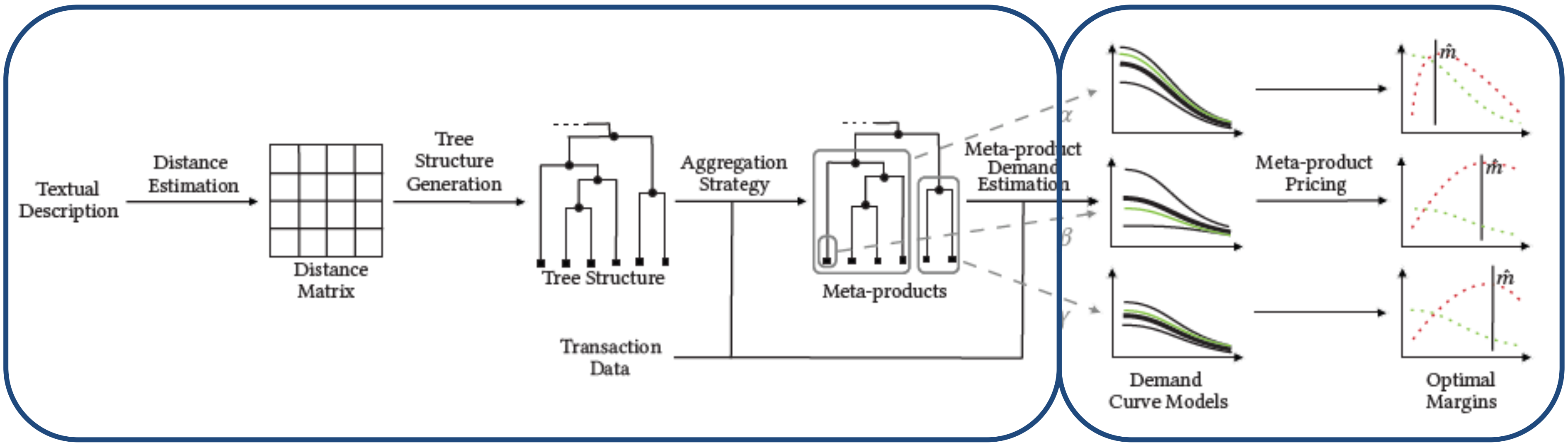


Chosen margin m^*_{jt} s.t.

$$m^*_{jt} = \underset{m}{\operatorname{argmax}}\ \hat{v}(m, \tilde{\boldsymbol{w}}_{jt})\, m\, c_j$$

[5] Nuara, Alessandro, et al. "A combinatorial-bandit algorithm for the online joint bid/budget optimization of pay-per-click advertising campaigns." *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 32. No. 1. 2018.

Algorithm - Overview



Data Aggregation

Online Optimization Step

Why Product Aggregation

- Most of the products register **less than 1 sale** per week

- This translates in an agent often receiving an **uninformative feedback** from the environment

- An aggregation procedure is required to generate informative samples

- Our goal is to design a way to aggregate data from **similar long-tail products**

Products Similarity

- The only technical data available are **products' textual descriptions**

$$\kappa_{ij} = tf_{ij} \log \frac{N}{df_i} \xrightarrow{\textit{Cosine similarity}} \mathcal{S} = \begin{bmatrix} 1 & \cdots & 0.3 \\ \vdots & \ddots & \vdots \\ 0.7 & \cdots & 1 \end{bmatrix}$$

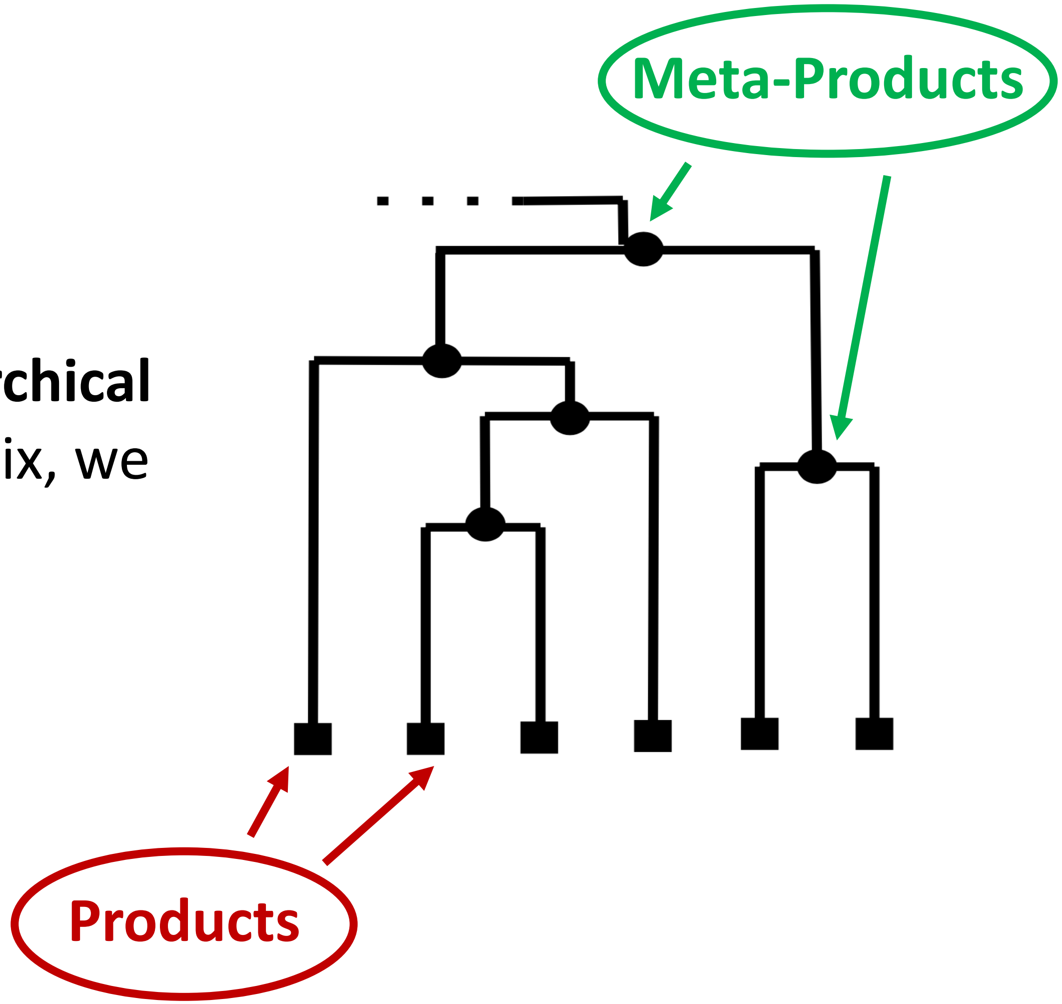
tf_{ij} = number of occurrences of word i in description of j

df_i = number of products containing i in description

- We resort to TF-IDF to **encode each product in a vector $\boldsymbol{\kappa}_j$**, then, **cosine similarity** is used to build a **matrix expressing the similarity** between each couple of products

Aggregation Structure

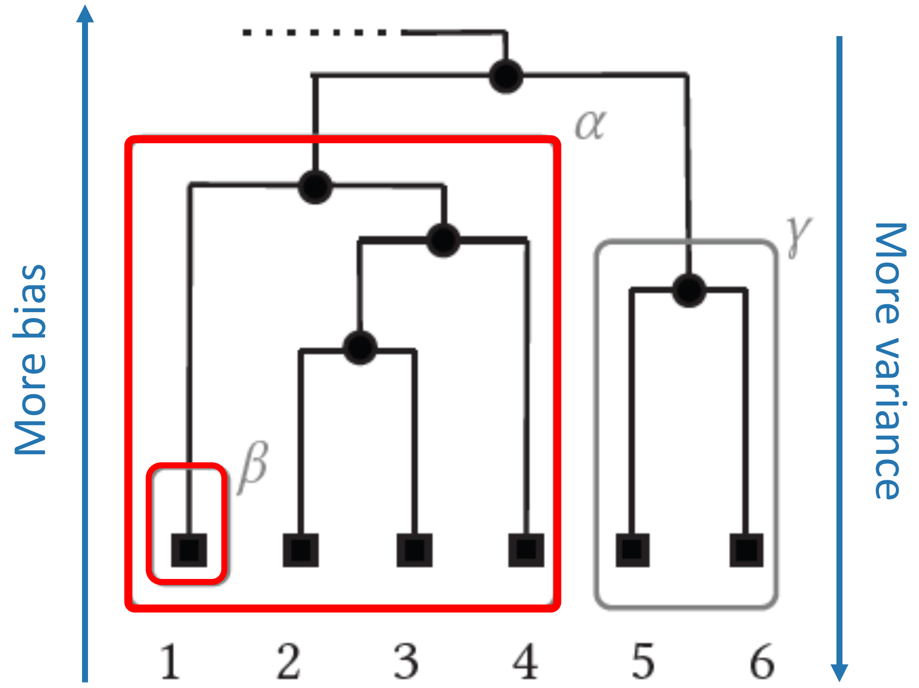
Using a **single linkage hierarchical clustering** on similarity matrix, we construct a **tree structure**



Product Aggregation Strategy

For every product:

- If its non-zero samples are **greater than a threshold τ**, the product can be priced alone
- If not, we **climb up a level** in the tree and **recursively perform this check** on the next meta-product



Data Sample Aggregation

- Once the meta-product α related to product j has been found, the meta-product data are estimated:

$$v_{\alpha t} \coloneqq \sum_{k \in \alpha} v_{kt}$$

$$m_{\alpha t} \coloneqq \sum_{k \in \alpha} m_{kt} \frac{v_{kt}}{v_{\alpha t}}$$

- Treating the meta-product as a single product, compute its **optimal margin,** and apply it **to product** j

Experimental Campaign

Experimental Setting

- The solution is tested in **both synthetic** and **real** environments
- An A/B test is performed on a real e-commerce over 7826 products with an yearly turnover of 2.5MEuros
- The test lasted $\boldsymbol{T}$ **= 8 weeks**
- The **same time period** $\boldsymbol{C}$ of the **previous year** is considered as benchmark

Configuration A

$N_A = 5694$ products

Priced by our algorithm

Configuration B

$N_B = 2132$ products

Priced by human specialist

Evaluation Metric

- The chosen metric is the **total profit** collected during the **period of 8 weeks**

$$M(A,T) := \sum_{t=1}^{T} \sum_{j \in A} v_{jt} m_{jt} c_j$$

- Due to the different overall magnitude in the two sets' weekly profits, we compare the **ratio of profits between the two periods**

$$G := \frac{M(A,T)}{M(A,C)} \frac{M(B,C)}{M(B,T)}$$

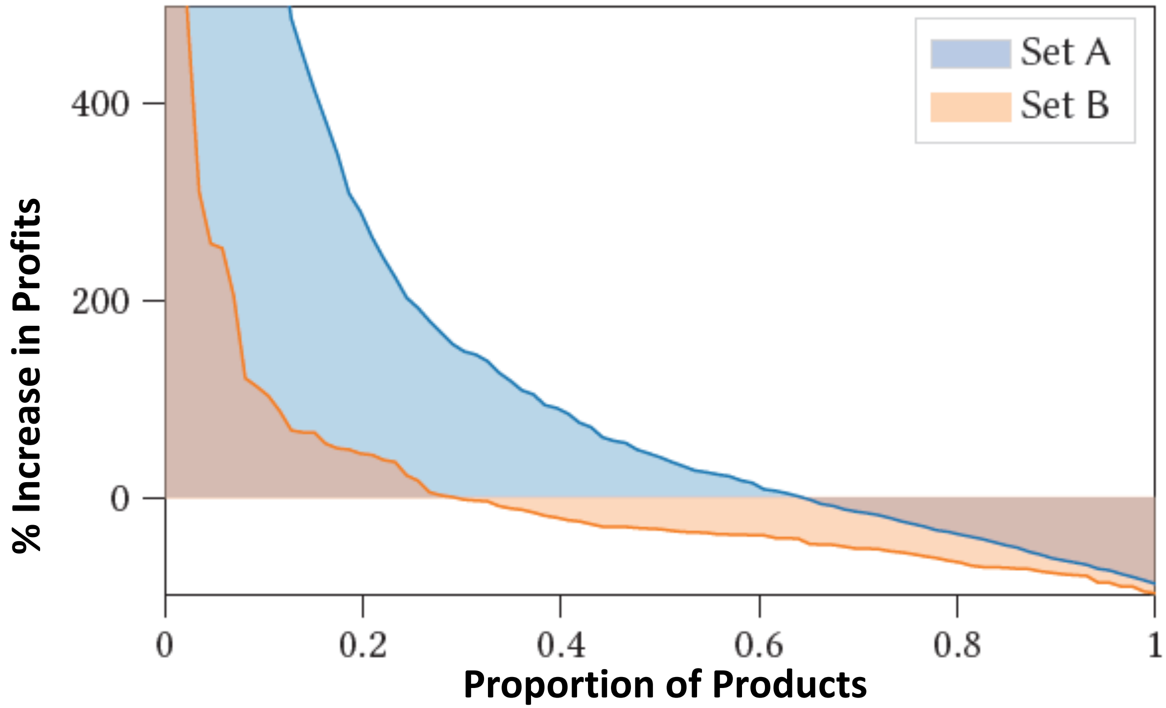
Overall Results

- We **separated the analysis** between bestsellers (or *popular*) products and long-tail ones

- The metric is **in favor of our algorithm**, attesting an increment in profits w.r.t previous year that is **40% higher in configuration A**

Popular G_P	Long tail G_{LT}	Total G
1.18	1.91	1.4

Results at Product Level



Configuration A had a positive increase in profits in 65% of the product. Configuration B had only 29% of the products improved from previous year

Conclusions

- We provided a novel dynamic pricing algorithm that can **face the exploration-exploitation dilemma** in long-tail markets

- We evaluated the methodology through a **real-world campaign**, obtaining results in favor of the algorithm

Future Works

- Integration of **advertising** and **recommendation** strategies in the long-tail framework
- Online learning of products' **interactions**
- Analyzing the performances under a more complex **user behavior model**

Thank you for the attention!

Take a look at our work

POLITECNICO
MILANO 1863