Postprint: Joint Inventory Control and Dynamic Pricing for Fresh Products Based on Deep Reinforcement Learning

Authors: Bi Wenjie, Yubing Zhou

Date: 2022-05-10T00:00:00+00:00

Abstract

To address the challenge in obtaining optimal ordering and pricing strategies for perishable goods due to their perishability characteristics and complex, dynamic real-world environments, a joint inventory control and dynamic pricing method based on deep reinforcement learning is proposed. By incorporating the characteristics of perishable goods, the problem is modeled and formulated as a Markov decision process, and subsequently, a joint inventory control and dynamic pricing algorithm for perishable goods is designed based on deep reinforcement learning. Experimental results demonstrate that the joint inventory control and dynamic pricing strategy based on deep reinforcement learning achieves the best revenue performance. Therefore, research on joint inventory control and dynamic pricing based on deep reinforcement learning can enhance corporate revenue, effectively promote the practical implementation of reinforcement learning in the revenue management domain, and demonstrates practical application value.

Full Text

Preamble

Vol. 39 No. 9
Application Research of Computers
ChinaXiv Partner Journal

Research on Joint Inventory Control and Dynamic Pricing of Fresh Produce Based on Deep Reinforcement Learning

Bi Wenjie, Zhou Yubing
(Business School, Central South University, Changsha 410083, China)

Abstract: Due to the perishable nature of fresh produce and the complex, dynamic real-world environment, obtaining optimal ordering and pricing strategies for fresh products is extremely challenging. This paper proposes a deep reinforcement learning approach for joint inventory control and dynamic pricing of fresh produce. The problem is modeled by incorporating the characteristics of fresh products and formulated as a Markov decision process, followed by the design of a joint inventory control and dynamic pricing algorithm based on deep reinforcement learning. Experimental results demonstrate that the joint inventory control and dynamic pricing strategy based on deep reinforcement learning achieves the best revenue performance. Therefore, this research can effectively improve enterprise profitability and promote the practical implementation of reinforcement learning in revenue management, offering significant real-world application value.

Keywords: deep reinforcement learning; revenue management; fresh produce; inventory control; dynamic pricing

0 Introduction

In recent years, rising living standards have led consumers to increasingly pursue quality lifestyles, resulting in growing demand for fresh produce. The rapid development of e-commerce and continuous improvement of logistics systems have also made consumers more stringent about product quality. According to 2020 statistical yearbook data from the National Bureau of Statistics, fresh produce accounts for over half of the per capita consumption of major food categories among national residents, with demand showing a year-over-year increasing trend.

Given the broad market prospects for fresh produce and the continuous development of e-commerce and cold chain logistics in recent years, the "fresh produce e-commerce" model has emerged. Businesses sell fresh fruits, vegetables, and meat online, developing diverse operational models including "in-store pickup," "in-store + home delivery," community group buying, and "locker pickup." Fresh produce exhibits tremendous market potential. However, in 2018, only 1% of national fresh produce e-commerce companies achieved profitability; in 2019, numerous small and medium-sized enterprises closed down; in 2020, the COVID-19 pandemic brought a spring to the fresh produce e-commerce industry while simultaneously intensifying competition. Consequently, fresh produce e-commerce urgently needs solutions to break through these challenges.

Fresh produce belongs to the category of perishable goods, exhibiting distinct perishability characteristics: seasonality, short life cycles, low salvage value for unsold products at the end of the period, and strong demand randomness [1, 2]. These features increase the risk of product spoilage and elevate sales risks. Therefore, enterprises must consider product perishability when formulating reasonable inventory and pricing strategies to maximize revenue. Additionally, product price affects demand, which in turn influences optimal inventory strategies. This interdependent relationship between price and inventory implies that pricing and inventory control decisions should be made jointly, as joint decision-making can effectively enhance enterprise revenue. Fang [3] found that compared to optimizing inventory strategy alone, joint pricing and inventory decisions can increase revenue for high-fashion enterprises by 12.36%, while compared to optimizing pricing strategy alone, joint optimization can yield a 9.78% revenue improvement.

Big data technology has gradually become a competitive advantage for enterprises, yet traditional research has lacked utilization and mining of the big data resources stored by companies. Consequently, relevant theoretical research has gained new directions and developments. Some scholars have applied reinforcement learning techniques to revenue management, discovering that reinforcement learning algorithms such as Q-learning can better control inventory costs when applied to inventory control strategies for perishable products [4]. However, many studies using Q-learning struggle to solve the curse of dimensionality problem brought by big data. In real-world scenarios, the environment is not only highly complex but also changes over time. Deep reinforcement learning methods can solve various problems without making assumptions about demand, offering stronger adaptability and generality that better suits complex and dynamic real-world environments.

Currently, deep reinforcement learning methods are rarely applied in revenue management. This paper investigates the joint inventory control and dynamic pricing problem for fresh produce using deep reinforcement learning. Furthermore, while many studies examine perishable product pricing based on single-batch products, few simultaneously investigate inventory control and dynamic pricing issues. This paper considers the coexistence of multi-age products and incorporates the randomness of product shelf life. Additionally, this study introduces deterioration rate as a random variable in the state transition function, making the time-varying state transition function more realistic, albeit more challenging to study.

Finally, this paper considers the impact of inventory status on customer reservation prices, modeling demand as a non-homogeneous Poisson process to simulate the complex and dynamic real-world environment. Building upon existing research, this study is more realistic and provides managerial insights for enterprise decision-making.

1.1 Research on Inventory Control and Pricing Strategies for Fresh Produce

Since fresh goods inherently possess perishable and easily damaged properties, they constitute a major category in perishable inventory management and dynamic pricing research [5], holding certain representative significance. Numerous papers recognize the importance of perishable goods research and specifically study inventory decisions for perishables. Inventory control problems for perishable goods have been extensively studied based on factors such as deterioration rate [6–8], price discounts [1, 9], supplier deferred payment [10], and limited storage capacity [11]. Early researchers set the deterioration rate as a fixed constant. However, in real life, the deterioration rate of perishable goods is not static. Therefore, researchers gradually began to consider time-varying deterioration rates, with some assuming it follows a Weibull distribution. Mishra and Umakanta [7] considered a time-dependent deterioration rate, assuming it follows a Weibull distribution, and proposed an inventory control system.

In the field of dynamic pricing for perishables, some scholars have adopted the WTP (Willingness To Pay) model to measure the impact of pricing strategies on consumer purchasing decisions [12, 13]. Product scarcity has also attracted attention from many researchers [14]. Tunuguntla V et al. [15] argued that available inventory quantity can also affect customers' reservation prices, thereby influencing their willingness to pay. Herbon et al. [16] constructed a dynamic pricing model for perishable products, analyzing the impact of consumer sensitivity to product freshness on pricing policies by studying how product price and shelf life affect demand.

Furthermore, some scholars believe that jointly studying optimal pricing and inventory will be a new research direction in the perishable goods domain [17]. In recent years, numerous studies have investigated joint inventory and pricing strategies for perishables [18–21]. Reference [18] examined joint pricing, advertising, and inventory control for perishables with psychological inventory effects, treating inventory level, advertising goodwill, selling price, and freshness index as demand influencing factors. Tang Yuewu et al. [19] studied single-stage and two-stage pricing and inventory decision models based on consumer strategic behavior.

1.2 Research on Inventory Control and Dynamic Pricing Based on Reinforcement Learning

Traditional inventory and pricing strategies require assumptions about market demand to simplify problems, which differs from the complex and dynamic real-world environment. Many scholars have applied artificial intelligence techniques to solve revenue management problems, proposing inventory control research based on reinforcement learning methods. Dogan et al. [22] used reinforcement learning algorithms to analyze an infinite-horizon joint ordering and pricing problem for retailers considering multi-retailer competition. Rana and Oliveira [23] considered revenue management for multiple interdependent perishable products within a finite horizon, using Q-learning with eligibility traces to formulate optimal pricing policies when demand is random and its functional form is unknown.

Reinforcement learning methods often perform poorly when problems have very large state and action spaces and unknown state transition probabilities, primarily due to two unresolved issues: large state and action spaces make it difficult to store value functions or optimal actions for each state, and limited training data cannot provide sufficient experience for every state [24]. The Q-learning algorithm encounters the curse of dimensionality when facing complex real-world problems, as the Q-table becomes extremely large, causing storage and retrieval difficulties. Consequently, reinforcement learning has been limited to low-dimensional problems with small action and sample spaces due to memory complexity, computational complexity, and sample complexity [25, 26].

Deep learning is well-suited for handling high-dimensional continuous problems, leading scholars to combine reinforcement learning with deep neural networks to form deep reinforcement learning. In recent years, researchers have begun using deep reinforcement learning to study inventory problems. Oroojlooyjadid et al. [27] proposed a multi-agent deep reinforcement learning algorithm and transfer learning method for the beer game, testing it with a real-world dataset and demonstrating that the algorithm significantly outperforms base-stock policies when other agents use realistic behavioral models. Reinforcement learning theory has also been extensively applied to dynamic pricing research, with scholars using Q-learning [28], policy gradients [29], and other reinforcement learning algorithms to study multi-agent dynamic pricing strategies. Wang [30] further used deep reinforcement learning to investigate joint inventory and pricing problems, focusing on the more challenging perishable goods domain. They used neural networks to avoid the curse of dimensionality, with results showing that deep reinforcement learning models outperform traditional reinforcement learning models without neural networks.

In summary, while many studies have applied reinforcement learning methods to inventory or pricing problems for perishable products separately, few simultaneously consider ordering and inventory for fresh produce, and even fewer incorporate product deterioration characteristics and demand complexity—precisely the real-world problems enterprises currently face. This research addresses this gap by modeling and solving the joint inventory and pricing problem for fresh produce using deep reinforcement learning to maximize expected total revenue.

2.1 Mathematical Model

Fresh produce e-commerce typically adopts "home delivery" or "in-store + home delivery" models, where platforms set up stores or front warehouses near communities or partner with nearby supermarkets and retail shops to provide integrated online-offline services. After consumers place orders on the platform, logistics quickly deliver goods to their homes or consumers choose in-store pickup. Large fresh produce e-commerce platforms often operate in rapidly changing complex environments, leading to high uncertainty in consumer demand. Mean-while, the perishable nature of fresh produce causes deterioration, damage, and spoilage during storage and sales, affecting product inventory status. Therefore, utilizing the vast amounts of data stored on platforms to continuously adjust and optimize inventory and pricing strategies is the only viable solution.

Based on the above real-world scenario, this paper considers the joint inventory and pricing problem for large fresh produce e-commerce platforms such as JD Daojia and Hema Fresh. Retail stores require daily morning delivery from distribution centers, and retailers forecast and determine daily order quantities and prices based on previous day' s sales.

When customers visit fresh produce e-commerce platforms or physical retail stores, they decide whether to purchase a product based on their reservation price versus the actual price. Research has shown that product inventory levels also affect customers'reservation price settings [15]. Currently, some e-commerce platforms display current inventory status on product detail pages, such as Taobao and Amazon. When customers discover low inventory and anticipate imminent stockouts, they experience a sense of urgency, believing any delay might cause them to miss the product, thereby increasing purchase desire and raising reservation prices. Therefore, this paper assumes that both the mean and standard deviation of reservation prices increase as inventory decreases.

The product life cycle is , and products require a lead time L to reach the retail store warehouse. This paper introduces the following assumptions:

a) The distribution center is assumed to have unlimited supply capacity. Products enter upon allocation.

b) Based on Mishra and Umakanta [7], this paper uses the Weibull function to represent product perishability characteristics, following a two-parameter Weibull distribution: is the scale parameter, is the shape parameter.

c) Due to the perishable nature of fresh produce, products will be discontinued and destroyed after expiration, with zero salvage value. No replenishment occurs if products sell out early in the day.

d) Sales are assumed to follow a First-In-First-Out (FIFO) strategy, meaning products with shorter remaining shelf life are sold first.

e) Customer arrivals follow a non-homogeneous Poisson process with intensity [31]. Heterogeneous customer purchase probability depends on reservation price . When product price is lower than the customer' s reservation price, the customer purchases immediately. This paper also considers customer price sensitivity coefficient . Thus, the demand process can be expressed as a non-homogeneous Poisson process with parameter .

f) Customer reservation price is influenced by instantaneous inventory , where parameter k adjusts the inventory range.

In summary, the mathematical model for the joint inventory and pricing problem in this paper is as follows:

where the mathematical symbols are defined in Table 1 .

2.2 Markov Decision Process

The joint inventory and pricing problem for fresh produce described above is a sequential decision problem. Reinforcement learning has proven effective in solving complex sequential decision problems across various scenarios. Its characteristic lies in learning a mapping from environment to behavior for intelligent systems. Agents interact with the environment through behavioral policies and receive feedback. If a policy yields positive reward feedback, the agent strengthens its tendency to select that policy, gradually learning the optimal strategy through iterative updates.

A Markov decision process provides the mathematical description of a reinforcement learning problem. This paper constructs the reinforcement learning quadruple to solve the joint inventory and pricing problem for fresh produce using reinforcement learning algorithms. The Markov decision process is defined as follows:

State Space: The state variable at period t is , representing the inventory quantity of products with remaining shelf life i at period t. L is the order lead time, is the product shelf life, and represents products in transit. When lead time L is 0, indicates the retailer's available inventory. If , then current available inventory .

Action Space: The decision variable at period t is product price . Order quantity follows the rule: the retailer observes demand at period t and decides order quantity based on the previous period' s demand plus/minus a quantity. is the order quantity, meaning the retailer decides while simultaneously determining .

State Transition: The deterioration rate represents the perishability characteristics of fresh produce at time t. Assuming when retailers have insufficient inventory to fulfill an order, the order disappears. The inventory with remaining shelf life after fulfilling orders transfers to the next period, while expired products are disposed of.

Reward Function: After observing current state and making decision , the agent receives corresponding reward , which measures action value. From state variable , available inventory can be determined, enabling calculation of holding or shortage costs and disposal costs for expired products. Order quantity from the decision variable determines ordering costs, while price and demand generate current period revenue . Thus, the reward function can be expressed as: . When retailer' s available inventory exceeds demand, sales volume ; when stockouts occur, unmet orders do not carry over to the next period.

The sequence of events in period t is as follows:

a) At period beginning, update state to for period t.

b) Agent makes decision , determining order quantity and product price . Products arrive after L periods. When , products arrive immediately.

c) Demand arrives. Under stockout conditions, unmet orders disappear. After fulfilling orders, inventory within remaining shelf life transfers to the next period, while expired products are disposed of.

d) At period end, settle revenue and costs for the period. Agent receives reward and updates state to .

Bellman' s principle and Markov decision processes, described by the Bellman expectation equation relating state value functions and action value functions, form the theoretical foundation of reinforcement learning methods. However, directly solving the Bellman expectation equation is impractical. Reinforcement learning has evolved into value-based methods (Value-Based RL) and policy-based methods (Policy-Based RL). Representative algorithms for the former include the online SARSA algorithm and the off-policy Q-learning algorithm; the latter includes Policy Gradients, which targets continuous action spaces by directly outputting next action probabilities. Additionally, these two categories merge to form actor-critic algorithms, which combine advantages of both by using a Critic to learn the reward mechanism and an Actor to output action probabilities.

The goal of reinforcement learning is to obtain optimal rewards through learned policies. As a value function iteration method, the Q-learning algorithm records state-action pair values in a Q-table and updates it based on the Bellman equation:

Q-learning has obvious limitations: when state or action spaces are extremely large, the algorithm struggles to establish and maintain a massive Q-table, making it impossible to record all states and actions. To address the curse of dimensionality, Mnih et al. [32] combined Q-learning with deep learning to develop Deep Q-Network (DQN), which uses neural networks to approximate the Q-table —that is, using a function to represent state-action pairs. DQN can estimate Q-target values from current rewards and next-state Q estimates, minimizing the difference between Q-estimates and Q-targets to update neural network parameters. The loss function uses mean squared error between estimates and targets:

Training data collected in reinforcement learning often exhibits correlation, causing neural network instability. DQN uses an experience replay mechanism (replay buffer) to avoid this issue by storing each exploration step' s data in an experience pool and sampling from it to calculate Q-target values, then updating current neural network parameters.

To solve DQN' s overestimation issues and optimize algorithm performance, scholars have proposed Double DQN, DQN with Prioritized Replay, and Dueling DQN. Dueling DQN modifies DQN' s neural network architecture, significantly improving algorithm efficiency. Unlike DQN, which uses neural networks to directly output Q-values for each action, Dueling DQN defines a state value estimate and an advantage function estimate , where the Q-value is composed of these components.

In summary, for the joint pricing and inventory control problem in this paper, the agent' s state is set as , representing current inventory status. This paper employs experience replay and fixed target network mechanisms, setting up two neural networks with identical structures but different parameters: an evaluation network and a target network. Additionally, an advantage function is defined for Q-values to optimize algorithm performance. Specific neural network details will be described in the next section. In each period, the agent makes decisions following an epsilon-greedy policy, selecting an action from the ordering and pricing action spaces. After taking action, the agent receives environmental feedback to estimate target Q-values and update evaluation network parameters. After a fixed number of steps, evaluation network parameters are assigned to the target network. The specific algorithm for fresh produce inventory control and pricing is as follows.

Algorithm: Fresh Produce Inventory Control and Pricing Algorithm

Input: Environment
Output: Action value function

a) (Initialization) Initialize evaluation network parameters and target network parameters ; initialize experience pool with size set to N.

b) Episode loop: For each episode, perform:

   1. Initialize environment and state ;
   2. For each time step until episode termination:
      - (Greedy policy) With probability , randomly select an action from ; otherwise select ;
      - Execute action , observe reward and next state from environment;
      - (Experience storage) Store quadruple in experience pool ;
      - (Experience replay) Sample a batch of data from experience pool;
      - Calculate return estimates: ;
      - Update neural network parameters approximating the action value function;
      - Update state variable ;
      - Every C steps, update target network parameters .

---

3 Algorithm Design for Joint Pricing and Inventory Control of Fresh Produce

The history of reinforcement learning can be traced back to Bellman' s foundational work. The Bellman optimality condition and Markov decision processes describe the relationship between state value functions and action value functions through the Bellman expectation equation, forming the theoretical basis of reinforcement learning methods. However, directly solving the Bellman expectation equation is impractical. Reinforcement learning has gradually developed into value-based reinforcement learning methods (Value-Based RL) and policy-based reinforcement learning methods (Policy-Based RL). Representative algorithms for the former include the online-updating SARSA algorithm and the off-policy Q-learning algorithm; the latter includes policy iteration algorithms (Policy Gradients), which target continuous action spaces by directly outputting the probability of the next action. In addition, the merger of these two algorithm categories forms actor-critic algorithms, which combine the advantages of both by using a Critic to learn the reward mechanism and an Actor to output action probabilities.

The purpose of reinforcement learning is to obtain optimal rewards through learned policies. As a value function iteration reinforcement learning method, the Q-learning algorithm records the values of state-action pairs in a Q-table and updates the Q-table based on the Bellman equation:

Q-learning has obvious limitations: when the state space or action space is particularly large, the Q-learning algorithm struggles to establish and maintain a massive Q-value table, making it difficult to record all states and actions. To solve the curse of dimensionality problem, Mnih et al. [32] combined the Q-learning algorithm with deep learning, extending it into Deep Q-Network (DQN), which uses neural networks to approximate the Q-value table—that is, using a function to represent state-action pairs. DQN can estimate Q-target values from current rewards and Q-estimates of the next state, and minimize the difference between Q-estimates and Q-targets to update the neural network parameters. The loss function uses the mean squared error between estimates and targets:

Training data collected by reinforcement learning often exhibits correlation, which can lead to neural network instability. DQN uses an experience replay mechanism (replay buffer) to avoid this problem. Specifically, it employs an experience pool to store data from each exploration step , samples from it, and calculates Q-target values based on the sampled data.

To address DQN' s overestimation issues and optimize algorithm performance, scholars have gradually proposed methods such as Double DQN, DQN with Prioritized Replay, and Dueling DQN. In the Dueling DQN algorithm, researchers modified DQN' s neural network architecture, significantly improving algorithm efficiency. Unlike DQN, which uses neural networks to directly output Q-values for each action, Dueling DQN defines a state value estimate and an advantage function estimate . The Q-value is composed of these components.

In summary, for the joint pricing and inventory control problem in this paper, the agent' s state is set as , representing the inventory status in the current period. This paper adopts experience replay and fixed target network mechanisms, setting up two neural networks with identical structures but different parameters: an evaluation network and a target network. Simultaneously, an advantage function is defined for Q-values to optimize algorithm performance. Specific details of the neural network settings will be described in the next section. In each period, the agent makes decisions according to an epsilon-greedy policy, selecting an action from the ordering and pricing action spaces. After the agent takes action, it receives environmental feedback to estimate the target Q-value and update the evaluation network parameters. After a fixed number of steps, the evaluation network' s parameter values are assigned to the target network. The specific algorithm for fresh produce inventory control and pricing is shown below.

Algorithm: Fresh Produce Inventory Control and Pricing Algorithm

Input: Environment
Output: Action value function

a) (Initialization) Initialize evaluation network parameters and target network parameters ; initialize experience pool with size set to N.

b) Execute episode by episode:

  1. Initialize environment and state ;
  2. While and , perform:
     - (Greedy policy) With probability , randomly select an action from ; otherwise, select the action with the maximum Q-value;
     - Execute action , observe reward and next state from environment;
     - (Experience storage) Store quadruple in experience pool ;
     - (Experience replay) Sample a batch of data from experience pool;
     - Calculate return estimates: ;
     - Update neural network parameters of the action value function approximation;
     - Update state variable ;
     - Every C steps, update target network parameters .

4.1 Selection and Testing of Deep Learning Hyperparameters

Since the algorithm introduces deep neural networks and model hyperparameters generally require manual setting (unlike parameters estimated from data), hyperparameter selection directly affects the stability and convergence of agent training, influencing training effectiveness. Therefore, how to select hyperparameters to ensure policy effectiveness is a critical issue in practical reinforcement learning applications. This paper compares learning rates and gamma values to analyze their impact on training effectiveness.

a) With other parameters held constant, we tested learning rates α=0.005, α=0.001, and α=0.005. Experimental results are shown in Figure 1 [Figure 1: see original paper]. The learning rate is a crucial parameter in deep learning model training, affecting the degree of neural network parameter updates and consequently model convergence. Setting the learning rate too high or too low leads to suboptimal model performance: an excessively high learning rate may cause model divergence—when α=0.005, the algorithm exhibits large oscillations in early stages and underperforms compared to α=0.001 in later stages; an excessively low learning rate results in prolonged training time, requiring more time to converge.

b) With other parameters fixed, we tested gamma values γ=0.9, γ=0.95, and γ=0.99. Experimental results are shown in Figure 2 [Figure 2: see original paper]. Regarding gamma values, higher parameter settings cause the agent to focus more on future total rewards rather than immediate short-term rewards, making training more difficult and slower.

The experiment sets specific parameters and obtains numerical results through interaction between the deep reinforcement learning algorithm and the designed simulation environment. This analyzes the algorithm' s performance in the simulated application environment and determines whether the algorithm can be applied to real-world environments. The experiment uses the control variable method to compare the impact of learning rate and gamma value parameters on results.

Based on the above model and algorithm analysis, this paper first configures the algorithm' s neural networks. The algorithm has two neural networks with identical structures, with parameters and respectively. Each neural network has two hidden layers using ReLU activation functions. The experience pool capacity N is set to 10,000, with random sampling each episode. The target network update interval C is set to 300 steps.

4.2 Performance Experiments of Deep Reinforcement Learning Algorithms

Reinforcement learning methods have been applied to revenue management with some research foundation. Q-learning and SARSA algorithms are widely used [33, 34]; the DDPG algorithm from deep reinforcement learning methods has also been applied in e-commerce [35]. Based on the above parameter settings, this paper conducts comparative experiments with the following baseline models:

a) DDPG algorithm incorporates deep neural networks into deterministic policy gradient methods. This paper sets both Actor networks and both Critic networks in DDPG as two-layer fully connected networks with ReLU activation functions, learning rate α=0.001, gamma value γ=0.95, and exploration value using the previously mentioned decay strategy.

b) Tabular reinforcement learning methods: SARSA algorithm solves optimal policies through on-policy temporal difference updates, while Q-learning is an off-policy algorithm using different methods to update optimal action value estimates. This paper sets the learning rate, gamma value, and other hyperparameters for both SARSA and Q-learning algorithms consistent with the above settings. Algorithm results are shown in Figure 3 [Figure 3: see original paper].

As shown in Table 2 , the DQN algorithm achieves the best revenue performance, followed by DDPG and Q-learning algorithms, with SARSA showing the lowest revenue. In terms of stability, DQN also outperforms other algorithms—compared to DQN, both DDPG and Q-learning exhibit more volatility.

Table 2: Performance Comparison Between Models

Model	Median	Average Revenue	Revenue Upper Bound	Revenue Lower Bound
DQN				
DDPG				
Q-learning				
SARSA				

Real-world demand changes are extremely complex, with various factors causing demand to exhibit strong randomness and non-stationary fluctuations. The DQN inventory control and dynamic pricing algorithm can solve the curse of dimensionality problem while providing near-optimal ordering and pricing strategies for enterprises. This demonstrates that the joint inventory control and dynamic pricing model based on DQN methods has broad application value.

---

5 Conclusion

Currently, demand for fresh produce is increasing year by year, with market scale expanding accordingly. For retailers, how to reasonably control inventory and pricing represents a crucial decision problem. This paper studies the joint inventory control and dynamic pricing problem for fresh produce, exploring optimal order quantities and pricing through deep reinforcement learning algorithms under continuously changing demand conditions to achieve enterprise revenue maximization.

First, this paper introduces time-varying deterioration rates as part of state transitions. As time progresses, fresh produce will deteriorate or spoil to varying degrees, and describing this phenomenon through changing deterioration rates better reflects reality. Reinforcement learning methods perform poorly with unknown state transition functions, and current revenue management research based on reinforcement learning rarely involves time-varying state transition equations. This paper addresses this limitation through deep reinforcement learning methods.

Additionally, when customers perceive low product inventory and imminent stockouts, the resulting sense of urgency increases their purchase desire, thereby affecting product demand. Considering this, retailers can strategically determine inventory and pricing strategies to maximize revenue. Based on inventory-influenced customer willingness to pay, this paper studies retailer inventory and pricing strategies for selling fresh produce within short sales horizons. Currently, few related studies exist in academia, and this research contributes to the perishable goods revenue management field.

In the joint inventory control and dynamic pricing problem for fresh produce, this paper designs a deep reinforcement learning algorithm that dynamically adjusts prices and order quantities based on current available inventory to maximize expected profit. This paper focuses on the joint pricing and inventory control problem for a single agent. Future research could delve deeper into inventory control and dynamic pricing under competitive conditions.

References

[1] Dan Bin, Chen Jun, Wu Qing. Optimal ordering policy for perishable product with progressive price discounts [J]. Chinese Journal of Management Science, 2006(3): 38-44.

[2] Zhao Quanwu, Xiong Zhongkai, Lin Ya, et al. Game analysis of two-stage supply chain for perishable goods under e-marketplace [J]. Chinese Journal of Management Science, 2004(3): 92-7.

[3] Fang F, Nguyen T-D, Currie C S. Joint pricing and inventory decisions for substitutable and perishable products under demand uncertainty [J]. European Journal of Operational Research, 2021, 293(2): 594-602.

[4] Kara A, Dogan I. Reinforcement learning approaches for specifying ordering policies of perishable inventory systems [J]. Expert Systems with Applications, 2018, 91: 150-8.

[5] Li R, Lan H, Mawhinney J R. A review on deteriorating inventory study [J]. Journal of Service Science and Management, 2010, 3(01): 117.

[6] Ferguson M, Ketzenberg M E. Information sharing to improve retail product freshness of perishables [J]. Production and Operations Management, 2006, 15(1): 57-73.

[7] Mishra U. An EOQ model with time dependent Weibull deterioration, quadratic demand and partial backlogging [J]. International Journal of Applied and Computational Mathematics, 2016, 2(4): 545-63.

[8] Li Yemei, Huang Shaoan. Advance payment strategy of non-instantaneous food based on EOQ model [J]. Chinese Journal of Management Science, 1-13.

[9] Viswanathan S, Wang Q. Discount pricing decisions in distribution channels with price-sensitive demand [J]. European Journal of Operational Research, 2003, 149(3): 571-87.

[10] Xu Xianhao, Wang Qian, Zeng Kuan, et al. Study on the optimal ordering policy of perishable products with delayed payment [J]. Chinese Journal of Management Science, 2021, 29(02): 108-16.

[11] Jing Fuying, Pan Yang. A two product dynamic lot size model with perishable inventory and joint ordering under bounded inventory [J]. Systems Engineering, 2018, 36(7): 47-54.

[12] Wertenbroch K, Skiera B. Measuring consumers' willingness to pay at the point of purchase [J]. Journal of marketing research, 2002, 39(2): 228-41.

[13] Kalish S. A new product adoption model with price, advertising, and uncertainty [J]. Management science, 1985, 31(12): 1569-85.

[14] Zhu M, Ratner R K. Scarcity polarizes preferences: The impact on choice among multiple items in a product class [J]. Journal of Marketing Research, 2015, 52(1): 13-26.

[15] Tunuguntla V, Basu P, Rakshit K, et al. Sponsored search advertising and dynamic pricing for perishable products under inventory-linked customer willingness to pay [J]. European Journal of Operational Research, 2019, 276(1): 119-32.

[16] Herbon A, Khmelnitsky E. Optimal dynamic pricing and ordering of a perishable product under additive effects of price and time on demand [J]. European Journal of Operational Research, 2017, 260(2): 546-56.

[17] Dye C-Y, Hsieh T-P, Ouyang L-Y. Determining optimal selling price and lot size with a varying rate of deterioration and exponential partial backlogging [J]. European Journal of Operational Research, 2007, 181(2): 668-78.

[18] Dye C-Y. Optimal joint dynamic pricing, advertising and inventory control model for perishable items with psychic stock effect [J]. European Journal of Operational Research, 2020, 283(2): 576-87.

[19] Tang Yuewu, Fan Tijun, Liu Sha. Pricing and inventory decision-making for fresh agricultural products with strategic consumers [J]. Chinese Journal of Management Science, 2018, 26(11): 105-13.

[20] Wang Shuyun, Jiang Yingmei, Mou Jinjin. Inventory and pricing decision of an integrated cold chain based on freshness [J]. Chinese Journal of Management

Science, 2018, 26(7): 132-41.

[21] Cao Yu, Yi Chaoqun, Wan Guangyu. Dynamic pricing, service level and production control strategy of stochastic production-inventory models with perishable products [J]. Systems Engineering—Theory & Practice, 2018, 38(7): 1717-31.

[22] Dogan I, Guener A R. A reinforcement learning approach to competitive ordering and pricing problem [J]. Expert Systems, 2015, 32(1): 39-48.

[23] Rana R, Oliveira F S. Dynamic pricing policies for interdependent perishable products or services using reinforcement learning [J]. Expert Systems with Applications, 2015, 42(1): 426-36.

[24] Zarandi M H F, Moosavi S V, Zarinbal M. A fuzzy reinforcement learning algorithm for inventory control in supply chains [J]. The International Journal of Advanced Manufacturing Technology, 2013, 65(1-4): 557-69.

[25] Arulkumaran K, Deisenroth M P, Brundage M, et al. A brief survey of deep reinforcement learning [J]. arXiv preprint arXiv: 170805866, 2017.

[26] Strehl A L, Li L, Wiewiora E, et al. PAC model-free reinforcement learning [C]// Proceedings of the 23rd international conference on Machine learning. 2006: 881-888.

[27] Oroojlooyjadid A, Nazari M, Snyder L V, et al. A Deep Q-Network for the Beer Game: Deep Reinforcement Learning for Inventory Optimization [J]. Manufacturing & Service Operations Management, 2022.

[28] Jintian W, Lei Z. Application of reinforcement learning in dynamic pricing algorithms [C]// 2009 IEEE International Conference on Automation and Logistics. IEEE, 2009: 419-423.

[29] Könönen V. Dynamic pricing based on asymmetric multiagent reinforcement learning [J]. International journal of intelligent systems, 2006, 21(1): 73-98.

[30] Wang R, Gan X, Li Q, et al. Solving a Joint Pricing and Inventory Control Problem for Perishables via Deep Reinforcement Learning [J]. Complexity, 2021, 2021.

[31] Zhao W, Zheng Y-S. Optimal dynamic pricing for perishable assets with nonhomogeneous demand [J]. Management science, 2000, 46(3): 375-88.

[32] Mnih V, Kavukcuoglu K, Silver D, et al. Human-level control through deep reinforcement learning [J]. Nature, 2015, 518(7540): 529-533.

[33] Wang Xin, Wang Fang. A review of dynamic pricing strategy based on reinforcement learning [J]. Computer Applications and Software, 2019, 36(12): 1-6.

[34] Gašperov B, Begušić S, Posedel Šimović P, et al. Reinforcement Learning Approaches to Optimal Market Making [J]. Mathematics, 2021, 9(21): 2689.

[35] Mosavi A, Faghan Y, Ghamisi P, et al. Comprehensive Review of Deep Reinforcement Learning Methods and Applications in Economics [J]. Mathematics, 2020, 8(10): 1640.

Note: Figure translations are in progress. See original paper for figures.