Université de Strasbourg

Faculté des sciences économiques et de gestion
Université de Strasbourg

UE - Architectures, modèles et langages de données

Big data project : Reinforcement learning

Hotelling's model of spatial competition

Moritz Müller, Professeur en Statistique & Econométrie
et Chercheur affilié au Bureau d'Economie Théorique et Appliqué
(BETA) à l'Université de Strasbourg

KHERMOUCHE Chafi
JANATI Zachary

SUMMARY :

1. Introduction

Spatial competition plays a crucial role in economics and game theory. The Hotelling model, developed by Harold Hotelling in the 1920s, provides a theoretical framework for analyzing competitive dynamics between companies located along a one-dimensional spatial line (generally between 0 and 1). This model has proved to be fundamental in understanding corporate location strategies and consumer decisions influenced by geographical proximity.

1.1 Background and significance

The contemporary economic context highlights the growing importance of strategic business location, whether in retail, foodservice or other physically rooted industries. Understanding how companies position their establishments, and how consumers react to these choices, is essential for predicting market trends, optimizing marketing strategies, and informing management decisions.

1.2 Study objectives

In this project, we aim to implement Hotelling's model using reinforcement learning agents in Python. We will explore the main underlying concepts, the model's assumptions and its practical applications.

1.3 Structure of the Project

The present study will be organized to provide an in-depth understanding of Hotelling's model. In the following section, we will examine the theoretical foundations of the model, highlighting its main elements. We will then detail the extensions to the model that we have chosen to incorporate into our project. We will then move on to the implementation part, in which we will detail the structure of the model, the simulation process, the reinforcement learning algorithm etc.... Finally, we'll provide our analysis of the results and conclude by summarizing the main limitations.

2. Basic model

Hotelling's general model is a duopoly model with two companies in a context of spatial competition. The two companies are located on a road, represented by a straight line graduated from 0 to 1, and the objective is to maximize profits, taking into account the reaction of the other company:

$$| - -^{a} - - A\overbrace{- - - - - -}^{x} - B - -^{b} - -|$$

Initially, the first company is located at a distance (a) from the beginning of the road, and the second is located at a distance (b) from the end.

Hotelling assumes that the good is homogeneous (there is no differentiation by quality of the good) and that consumers are evenly distributed across the market (the road). Consumers are assumed to buy a single unit of a good, and to make their choice according to the following criteria: the selling price and the transport cost linked to the distance between the consumer and the company.

In the end, there is indeed product differentiation, not by quality, but rather by the transport costs associated with the purchase of the goods.

In the general model, prices are assumed to be different. This means that market shares are determined by the prices set by A and B. However, if this had not been the case, A's market share would be the consumers to its left (a) and half the consumers between A and B (see image above).

Hotelling's solution is as follows:

$$p_A = u + c[L + \frac{1}{3}(a-b)]$$
$$p_B = u + c[L - \frac{1}{3}(a-b)]$$

<u>With :</u>

- u = cost price (= 0 for Hotelling)
- L = Road length (equal to 1 in our case)

Hotelling finds that A's profit increases if it moves closer to B (i.e. when A increases). It is therefore in both companies' interests to move closer together. This is why Hotelling's model is also known as the principle of minimal differentiation. Indeed, the principle asserts that in most markets, competition leads producers to reduce the difference between their products (by reducing distance). On the other hand, if we assume that transport costs are now of the second order, then we would observe an opposite trend (differentiation).

In our case, we maintain the following hypothesis: transport costs are fixed and equal to c. We must be careful, however, because if the two companies are too close, then the equilibrium price will be equal to the unit cost and the profit will be canceled out, which will not benefit the companies.

3. Extension/Our version

We've just announced the general Hotelling model for space competition. As part of our project, we decided to implement the model in a slightly more simplified format. It's based on a famous game-theoretic example: **The ice-cream seller's problem.**

3.1 Principle

The principle is as follows: Two ice cream vendors must select a location on a beach where customers are evenly distributed.
It is assumed that the prices and products of the two vendors are identical (the distinction is made only on the location of the vendors, i.e. the products differ only in transport costs), so that each customer systematically heads for the nearest vendor.

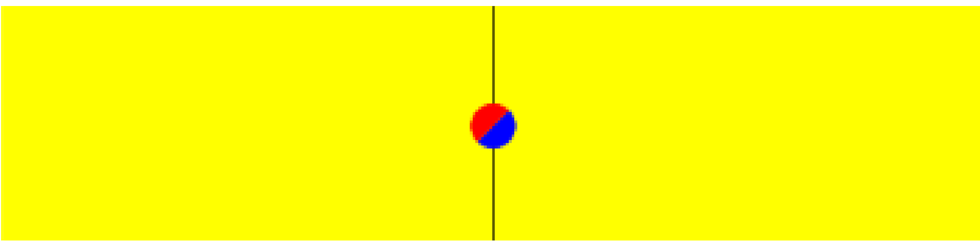
There are two aspects to this problem. On the one hand, we need to define the equilibrium position of this game, i.e. how the sellers will position themselves on the beach, assuming that each one seeks to maximize its profits. Secondly, we need to assess the optimality of this equilibrium, from the point of view of both sellers and customers.

3.2 Model equilibrium

Once the two sellers are set up, they naturally divide the beach into two zones: one seller's zone consists of all the points on the beach that are closer to him than to the other seller.
If one of the sellers' two zones is smaller than the other (which happens if one seller is further from the center of the beach), he can extend his zone by moving. Consequently, there is no equilibrium. Thus, equilibrium can only be achieved if the two zones have the same size, i.e. if the sellers are located on either side of the middle of the beach, at an equal distance.

However, if one of the sellers moves closer to the middle of the beach, he will increase his area at the expense of the other, who will also have to move closer to the middle of the beach to preserve his half of the beach. As a result, both sellers spontaneously move towards the middle of the beach, until they are both present.

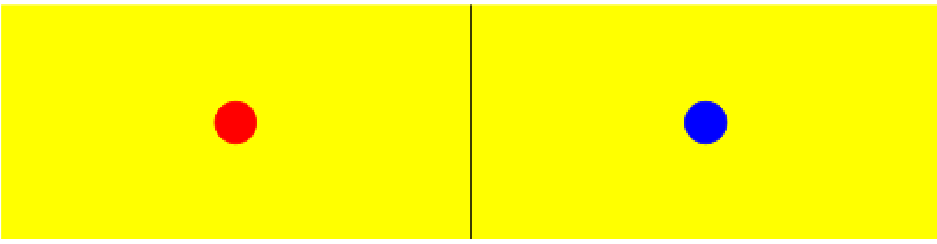
At this point, an equilibrium is reached: each seller occupies half the beach, and any slight movement to one side or the other results in a reduction of his area to the benefit of his competitor. This is the Nash equilibrium in this game (see image below).



3.3 The socially optimal situation

This game generates externalities, as customers are not indifferent to the position of the merchants, which has an impact on their walk. In particular, the equilibrium position, with the two merchants at the center of the beach, is not ideal, as customers at the far right have to walk halfway across the beach to buy their ice cream.

A more favorable distribution of vendors (from the point of view of the social optimum) would be to have one vendor in the middle of each half of the beach (see image below).



In this case, each vendor would still occupy an area equal to half the beach, and customers would only have to cross a quarter of the beach at most to buy their ice cream. However, this does not constitute an equilibrium, as it is possible for both vendors to move around while increasing their profit.

So, from the customer's point of view, the equilibrium of this game is not optimal. It is possible to make this equilibrium non-optimal for the merchants too, by assuming that a customer would rather give up his ice cream than cross more than a third of the beach. In this case, the equilibrium for the merchants would be to each position themselves at one-third of the beach, because if a merchant wishes to move closer to the middle to attract a customer from his competitor, he will then have to lose two who would come from the end of the beach (on average or assuming a homogeneous distribution of customers on the beach). This new equilibrium improves the condition of customers, but is not yet optimal for them.

4. Implementation

In this fourth part, we're going to present the coding part of our project and how we've implemented our model in Python. The code we've used obviously respects the structure you requested, i.e. comprising a Buffer, Policy, Trainer class and an environment (included in the trainer) for our model..

Our code is divided into two parts: the first corresponds to the model without including any hypothesis regarding consumer laziness, and in the second part, we include laziness in the sense that individuals won't buy the goods if the distance is high, even if the seller is the closest. We'll see that this has a fairly satisfactory impact on equilibrium.

4.1 Structure

As I said earlier, the architecture of the model is based on three fundamental classes: Buffer, Policy, and Trainer, each dedicated to specific aspects of modeling.
The Buffer class is instantiated with the size of the space (set to 250) and the number of random consumers/consumers (set to 5000). Random locations are generated for consumers.

The Policy class is initiated with the size of the space (250). It creates Q tables for two agents. We used the "update_q_table" method, which updates the Q table according to the current state, the next state, the reward, the learning rate (set to 0.85) and the reduction factor (set to 0.9).

The Trainer is instantiated with parameters such as the number of agents/sellers (set to 2), the size of the space (250), the learning rate (0.85), the reduction factor (0.9), the exploration decay (0.995), and the number of random consumers (5000). It also initializes the buffer and the policy.

The "consumer_preference" method was used to calculate consumer preferences as a function of distances between consumers and agents.

4.2 Simulation process (Initialization, Tournament process)

The simulation process begins with the initialization of the Trainer object, followed by the execution of a simulation loop. During each episode, agent positions are updated, rewards are calculated according to consumer preferences, and policy Q-tables are updated. A tournament-like interaction emerges as agents compete for consumers' attention based on their spatial positions.

4.3 Reinforcement Learning Algorithm

The reinforcement learning algorithm is based on the Q-learning method. The "update_q_table" method of the Policy class applies the Q update rule, taking into account the current state, the next state, the reward, the learning rate and the reduction factor. Exploration and exploitation strategies guide the updating of agent positions in the environment.

4.4 Parameters, Training process

The Trainer class is initialized with various parameters, including the number of agents (set to 2), the size of the space, the learning rate, the reduction factor (0.9), the exploration probability (0.85), the exploration decay (0.995), and the maximum number of episodes (set to 8000).

Training runs through a simulation loop for a defined number of episodes. At each episode, agent positions are updated according to exploration or exploitation. Rewards are calculated according to consumer preferences, and Q-Tables are updated using the Q-learning algorithm. The simulation includes a graphical representation of consumer and agent positions. The probability of exploration decreases progressively with each episode.

To summarize, this implementation models Hotelling's spatial competition scenario, where agents strategically position themselves to attract consumers in a spatial environment. The simulation offers insights into the convergence of agents towards the center of the space over time.

4.5 Introducing consumer laziness

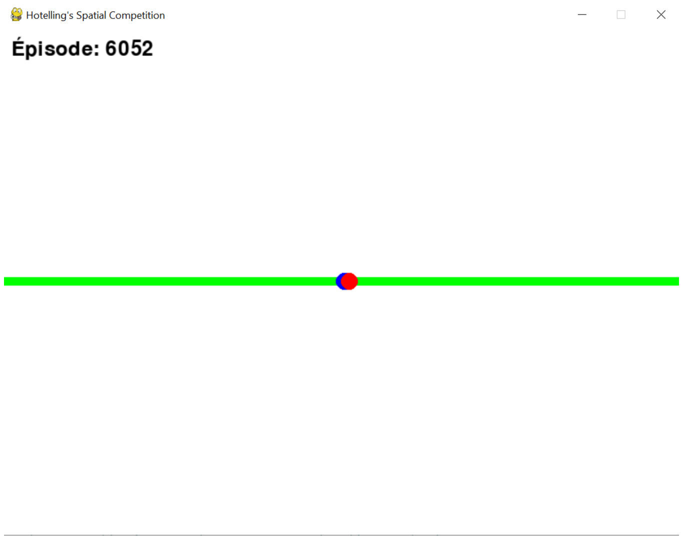
To take laziness into account, a new parameter ('laziness_threshold') has been introduced for the Trainer class. It represents the laziness threshold of consumers, an additional consideration in the calculation of consumer preferences.

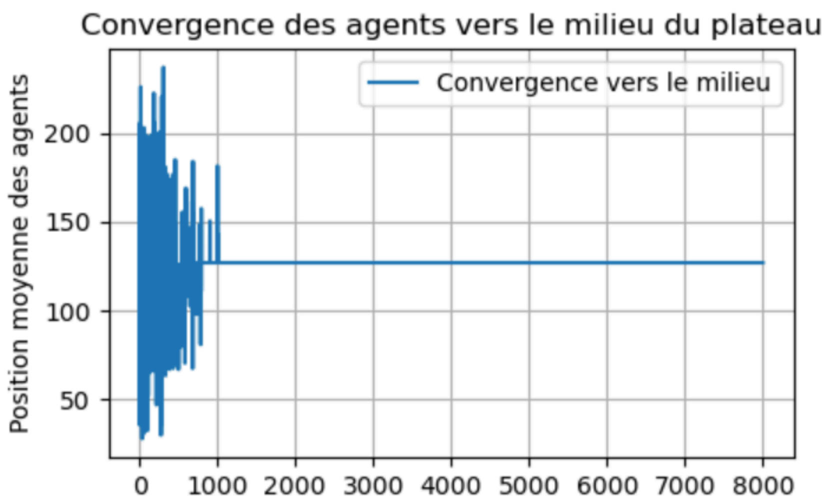
The second code includes a check in the "consumer_preference" method to exclude consumers whose distance to all agents is greater than the laziness threshold (set at 50). These consumers are ignored in the reward calculation.

5. Analysis

In this section, we'll be highlighting our results following the model's launch. As I said earlier, we have been involved in the ice-cream seller problem with and without lazy consumers.

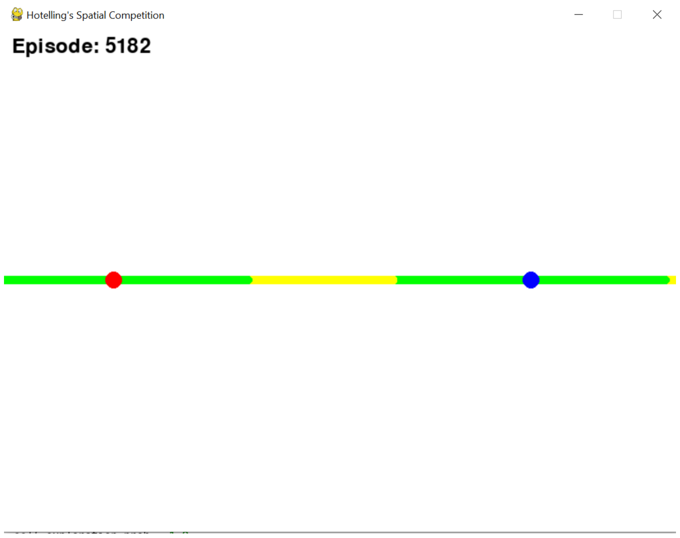
For the model without laziness, we obtain a convergence of positions in the middle of the straight line corresponding to the Nash equilibrium, as predicted by the model (see image below).



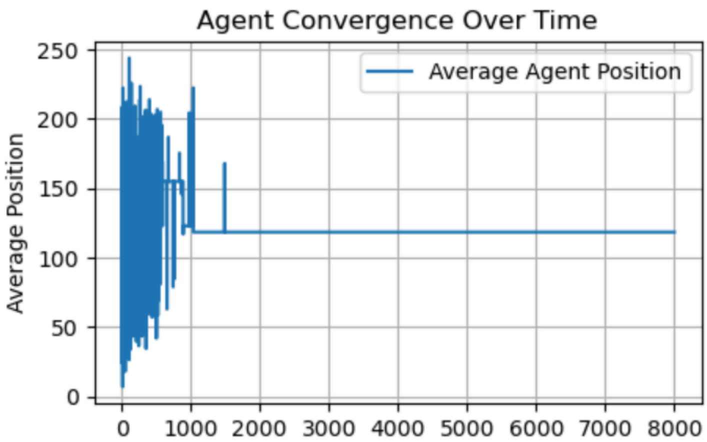


This graph shows that, from the 1000th episode onwards, the sellers' positions converge towards Nash equilibrium: A and B are positioned at 125. The average is therefore 125.

However, it's important to remember that, in section 3.3, we saw that equilibrium is not synonymous with the social optimum. Indeed, equilibrium does not benefit consumers at the ends of the range. The solution to this is to introduce the laziness of individuals, who will withdraw from the market if the distance to the nearest seller exceeds a certain threshold (set at 50). This brings us back to the social optimum in which sellers would be in the middle of each half of the beach (see image below).



Consumers shown in yellow are those who don't buy the goods..



This graph shows that, from the 1500th episode onwards, the sellers' positions converge towards the social optimum: A is at 62.5 and B is at 187.5. The average is therefore 125 ((62.5 + 187.5)/2).

But that's not all, laziness allows us two other different convergences (which don't seem to me to be equilibria).

The other two are convergences in which the two firms totally ignore the opposite side, left or right, which corresponds to the two cases observed. This can be explained by the calculation method used in the idea of consumer laziness, which probably should have included the idea of decreasing consumer laziness. In this case, the 3 convergences would not have been observed.

Another possible improvement is the idea of running a large number of parameter tests, automatically, in order to test the "best" values that the model could consider.

6. Limitations and Conclusion

The implemented model effectively captures the dynamics of spatial competition between agents in a resource allocation scenario. The use of Q-learning and spatial aspects of consumer preferences contributes to a realistic simulation. The integration of a laziness threshold for consumers adds a further layer of complexity, reflecting a more nuanced real-world scenario.

Despite the model's strengths, certain limitations remain. The current implementation assumes a discrete spatial environment and a uniform distribution of consumers, which may not fully reflect the complexity of real-world scenarios.
Furthermore, the simplicity of the spatial competition model may not take into account more complex market behaviors.

Several improvements could be considered. The transition to a continuous spatial environment could offer a more realistic representation. Improvements in the modeling of consumer preferences, by integrating diversified behaviors and preferences, could enrich the simulation. Refining learning parameters and exploring other reinforcement learning algorithms could optimize model performance.

Finally, the project has provided valuable insights into the dynamics of spatial competition and the application of reinforcement learning to simulate economic scenarios.

7. Source

Hotelling, Harold. "Stability in Competition." Economic Journal 39.153 (1929): 41-57.
Wikipédia | Wikiwand

"La règle de Hotelling." The Other Economy. Lien

d'Autume, Antoine, and Katheline Schubert. "Le modèle de Hotelling avec coûts de transport et concurrence imparfaite." HAL Archives Ouvertes (1991). Lien vers le PDF