

An approach based on Reinforcement Learning and Self-Organizing Maps to design a marketing campaign.

Gabriel Gomez-Perez*, Jose D. Martin-Guerrero*, Emilio Soria-Olivas*,
Emili Balaguer-Ballester†, Alberto Palomares†,
Nicolas Casariego‡ and Daniel Paglialunga‡

*Grup de Processament Digital de Senyals, Universitat de València
C/ Dr. Moliner, 50. 46100 Burjassot, Spain

†Tissat, Inc., R & D Department, Spain

‡Espirus Europa, S.L., Spain

Abstract—Reinforcement learning (RL) algorithms have shown to be a useful method to carry out many marketing applications such as the mailing problem [8], [1], [6]. However, some problems related to high-dimensional data spaces make RL algorithms difficult to implement without using function regressors [9]. In this context, the use of clustering algorithms becomes a great help in the task of aggregating states [7]. In this work, we will show how the combination of RL and Self-Organizing Maps (SOM) enables to avoid the curse of dimensionality and also provides a better explanation of results. Our proposal has been tested with real data from a marketing campaign, achieving results much better than those obtained by the policy followed by the company so far.

I. INTRODUCTION

The field in marketing science which studies the connections between marketing actions and client's response is called targeted marketing. This kind of applications can be viewed as a Markov chain problems, in which a company decides what action to take once known customer properties in time t , which is the current state of the customer. In [8] the problem of mailing is analysed by studying how an action in time t influences following times. In [1] and [6], several reinforcement learning (RL) algorithms are benchmarked in mailing problems. In [2], RL is used to optimise the cross channel marketing.

Therefore, despite RL has already been used in targeted marketing, only the mailing problem has been exhaustively studied. In this work, RL is applied to design optimal policies for bonus assignment in order to maximise company sales, using all the information available to define the state of the customer.

In addition, marketing problems tend to have a very complex characterisation of the transactions involved in them. This highly dimensional attribute make RL algorithms difficult to implement because the use function regressors is needed, and this may lead to convergence problems [3], [10]. This drawback will be resolved by using a vectorial quantization carried out by Self-Organizing Maps (SOM) algorithms.

The rest of the paper is outlined as follows. A description of RL and SOM is shown in Section II and III, respectively. Problem modelling by using a combination of RL and SOM is presented in Section IV. Section V shows results achieved, ending up the paper with some conclusions and proposals for further work in Section VI.

II. REINFORCEMENT LEARNING

The two main components in an RL algorithm are the agent and the environment, as shown in Fig. 1. The agent is the learner that makes decisions according to the state of the environment, while the environment is every external condition which cannot be modified by the agent.

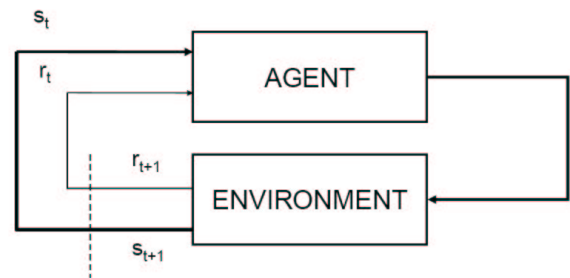


Fig. 1. Characterisation of the RL model. r_t denotes the immediate reward while s_t is the state of the environment.

The task of the learning agent is to optimise a certain objective function; this optimisation is carried out using only information from the state of the environment, i.e. without any external advisor. In particular, the task of the learning agent can be accomplished by modelling the system as a *Markov finite decisional process* (MDP). The following terms should be define as they will be used through the paper:

- **State of the environment (s_t):** Available information to define the behaviour of the environment which the agent is in.
- **Action (a_t):** Action taken by the agent.

- **Policy** ($\pi(s, a)$): Mapping from the state space to the action space which assigns an action a_t given some state s_t .
- **Immediate reward** (r_t): Signal returned by the environment to the agent depending on the taken action.
- **Long-term reward** (R_t): Sum of all the immediate rewards through a complete decisional process. It is the objective function which the agent is interested in maximising by taking optimal actions. Its mathematical expression is:

$$R_t = \sum_{k=0}^{\infty} \gamma^k r_{t+k+1} \quad (1)$$

where parameter γ is called the *discount rate*, and it ranges between 0 and 1. This factor stands for the relative weight between present and future rewards. Small values of γ indicate that only next rewards are taken into account, i.e., the agent maximises r_t . However, as γ is getting close to 1, further future rewards become more relevant, thus making the agent be further-sighted.

The goal is maximising R_t by means of an optimal behaviour policy. Hence, an estimation of the expected R_t as the result of an action a_t from a state s_t , is needed. This estimation is called the *action-value function*:

$$Q(s, a) = E_{\pi}[R_t | s_t = s, a_t = a] \quad (2)$$

where $Q(s, a)$ is the expected R_t starting at state s and taking action a and thereafter following policy $\pi(s, a)$. $Q(s, a)$ is a sort of register in which the expected cumulative rewards for all state-action pairs are stored. Therefore, a first approximation to the optimal policy is given by [9]:

$$\pi(s) = \arg \max Q(s, a) \quad (3)$$

Eq. (3) shows a deterministic policy, that is, given a fixed state only one action can be taken. Another approach is that of stochastic policies $\pi(s, a)$ where more than one action can be taken according to their probabilities, given by $\pi(s, a)$ [9]. This work is focused on deterministic policies.

The key issue on RL algorithms is computing the action-value function $Q(s, a)$ for a given arbitrary policy in order to obtain the optimal policy by Eq. (3). Many methods to compute $Q(s, a)$ are proposed in the bibliography, but the most widely used ones are *Temporal Difference methods (TD)* [9]. This kind of techniques does not require a model of the environment like Dynamic Programming, because the updates of the values in $Q_{t+1}(s, a)$ are made up by information from the interaction with the environment (r_t and s_{t+1}) and previous estimations $Q_t(s, a)$:

$$Q_{t+1}(s, a) \Leftarrow Q_t + \alpha[r_t + Q_t(s', a') - Q_t(s, a)] \quad (4)$$

being α the step-size of the update, and s' and a' state and action in time $t + 1$, respectively.

Sarsa and *Q-learning* are the most known TD methods. *Sarsa* is an on-line algorithm which modifies the starting policy towards the optimal one, being its update rule given by (4). Q-learning computes the optimal policy while the agent is interacting with the environment by another arbitrary policy. Since Q-learning is easy to implement and enables early convergence, this algorithm is the one used in this work. Its update rule is:

$$Q_{t+1}(s, a) \Leftarrow Q_t + \alpha[r_t + \max(Q_t(s', a')) - Q_t(s, a)] \quad (5)$$

III. SELF-ORGANIZING MAPS

Neural models based on SOM represent the input space into a lower dimensional space, usually one-dimensional or two-dimensional [5]. The main advantage of these models stems from the fact that this dimensionality reduction does not involve any change in the topological relation of input data.

The operation of SOM can be summarized in two steps:

- 1) For a given input vector $\mathbf{x} = [x_1, x_2, \dots, x_n]$, the Euclidean distance between this vector and all the set of weights $\mathbf{w} = [w_1, w_2, \dots, w_n]$ is computed as stated in Eq. 6. The weight w_i which is closer to the input pattern is the so-called *best matching unit*, (BMU).

$$d(\mathbf{x}, \mathbf{w}_i) = \|\mathbf{x} - \mathbf{w}_i\| \quad (6)$$

- 2) The weight vector values w_i are updated so that the match of the new set of weights in the active area (defined in the neighbourhood of the BMU) and the input patterns \mathbf{x} , is improved, Eq. 7.

$$\mathbf{w}_i = \mathbf{w}_i + \alpha N(i, BMU)(\mathbf{x} - \mathbf{w}_i) \quad (7)$$

where N_0 stands for the center of the area, $N(i, BMU)$ stands for the active area inside which weight vectors are modified. Several neighbourhood functions have been proposed in previous works, [4], but the one used in this work is the Gaussian neighbourhood function, defined as follows:

$$N(i, BMU) = N_0 \exp \frac{-\|r_i - r_{BMU}\|^2}{\sigma^2} \quad (8)$$

where $\|r_i - r_{BMU}\|$ is the distance between an input pattern and its BMU, and σ is the standard deviation.

Thus, the weights of the network form a new data space where close weights have similar properties according to the original distribution of data. This can be viewed as a vector quantisation, where the number of features required to identify an input pattern, is reduced by choosing some representative models which are tuned by the SOM.

This ability of SOM algorithms to cluster data preserving topological relations by unsupervised learning permits to reduce the dimensionality of the input space without any prior information about that space. For this reason, SOM has been used to quantise input spaces in RL frameworks [7].

In this work, the whole transaction set is used as input patterns to train an SOM network. This way, similar behaviours

of customers are located in close areas in the map, turning out to be in the same BMU or in a neighbouring one.

IV. PROBLEM MODELLING

A. Data collection and setup

Data were collected from a company which is interested in designing a campaign focused on encouraging their clients to buy more articles. Although a confidentiality agreement prevents a number of details of the campaign to be released, the main characteristics of the campaign can be made known:

- 1) The company assigns some virtual credits to clients according to the articles bought by them. When clients have enough credits they can exchange credits by gifts.
- 2) Clients can get virtual credits by buying specific articles which are indicated as “encouraged”. The company selects these promoted articles monthly (according to internal criteria).
- 3) Since the assignment of this bonus involves a cost for the company, immediate profits decrease as a direct consequence of the campaign.

The credit assignment takes place at the end of every month. The amount of bonus is computed taking into account how many “encouraged” articles were bought by the client during last month. The final reward for each client is obtained as follows:

$$LTV_t = \sum_i P_i \cdot C_i - K_c V_C \quad (9)$$

where C_i is the purchased amount of articles type i , P_i means the price of articles type i , V_C stands for the amount of virtual credits assigned and K_c is a coefficient which reports about the cost involved by credits to the company. In this work, we use the *Life Time Value (LTV)* as a representation of the total profits expected for each client in a month¹. The aim of this work is to increase LTV for each client by using RL as the strategy to achieve an optimal policy.

B. Action and state spaces

As it was stated in Section II the first task to tackle in an RL algorithm is the design of state and action spaces. This task requires an exhaustive study about characterisation of clients and actions.

The company which is developing the marketing campaign has a lot of stored information about its business transactions. There are many features which define the behaviour of a certain client. In particular, collected data register the following features:

- 1) Shop which carries out the sale.
- 2) Region which the client belongs to.
- 3) Amount of articles bought by the client.
- 4) Whether or not the articles are encouraged.
- 5) Price of the article.
- 6) Date.

¹In marketing literature LTV stands for the expected profits in long-term.

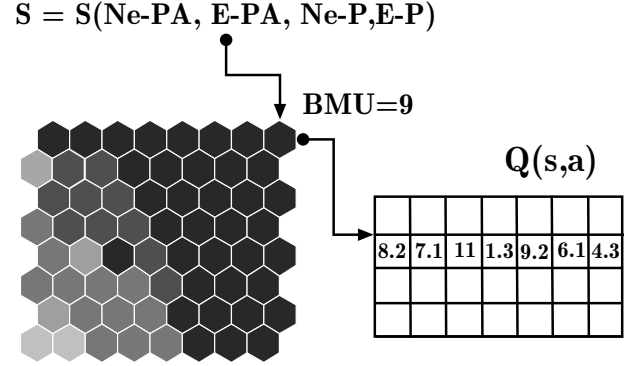


Fig. 2. Procedure of state computation and $Q(s,a)$ updates by means of SOM.

7) Type of article sold.

First of all a classical data mining study is carried out. By means of frequency analysis and cross-correlations computations it is concluded that the geographical and temporal information are not relevant. Thereby these two features are not considered in following discussions.

However, a high number of features is still present in the data set. If state vectors are produced taking into account all this information, the dimensionality of the problem becomes considerably large, and then the use of function approximation is required [9]. However, it is also possible to use tabular RL algorithms by means of reducing the number of features used.

In this work, as it was stated in Section III, the modelling of state space is made by SOM, which allows to take into account all features present in the input data without tackling a high dimensional problem. An SOM is trained with all the input patterns from the data collection. After that, the BMU for a particular input vector can be computed. This BMU is the state of the client. This approach locates similar customers in the same state, i.e., an aggregation of states is held.

Thus, we arrange variables for each customer in each temporal step in four features:

- 1) Non-encouraged amount of purchases: NE-Pa.
- 2) Encouraged amount of purchases: E-Pa.
- 3) Non-encouraged averaged prices: NE-P.
- 4) Encouraged averaged prices: E-P.

This four artificially produced features are the ones which form the input data set. However, the values that take these variables are in a wide range which makes a quantisation process useful for avoiding an extra computational cost. This task is carried out using a discretisation based on percentiles to ensure a uniform population of each space.

The process explained above can be viewed in Fig. 3. Given an input vector $s_t = s_t(\text{Ne} - \text{PA}, \text{E} - \text{PA}, \text{Ne} - \text{P}, \text{E} - \text{P})$ its BMU is computed. This value is used as an entry for the Q-table.

The main advantage of our approach is that there is no need to use function regressors avoiding the danger of non-convergent policies [3], [10]. The use of SOM as a discrete

Model

- 1) State. $s_t = BMU_i$, for a customer transaction.
- 2) Action. $a_t = a_t(V_C)$ where V_C stands for the virtual credits assigned.
- 3) Immediate reward. $r_t = LTV$ (for a month)

Training of the algorithm

- 1) Training an SOM.
- 2) The state of a customer is computed by looking for his/her BMU at time t .
- 3) Action carried out by the company.
- 4) Once reward is observed, the following state is obtained.
- 5) Computation of $Q(s, a)$.
- 6) Estimation of the optimal policy $\pi^*(s)$.

Fig. 3. Procedure of problem modelling and RL algorithm.

state representation is very helpful in the implementation of the RL algorithm and later, in the interpretation of the solutions achieved.

The action to take is the assignment of a certain amount of virtual credits to clients depending on their purchases. Since there is a wide range of credits, a quantisation is required to avoid the curse of dimensionality again. But now, this quantisation process is carried out taking into account the gift value for a certain amount of credits. The company establishes 10 categories of gifts according to their price so that the action space is divided into 10 categories.

The whole process explained in this section is summarized in Fig. 2.

V. RESULTS

The company splits customers into two main groups: VIP² or non-VIP; VIP customers bought almost all articles of the company whereas non-VIP customers only bought a small set of articles. Transactions involving each kind of customers have very different features, whereby the algorithm is used separately in the two data sets.

The results achieved are analysed from three points of view. First, the use of SOM for aggregating states is studied. Second, an exhaustive study of optimal policies and comparisons with real policies are carried out. Finally, profits involved by the use of optimal policies are calculated.

A. Aggregating states by SOM

The use of SOM in this problem is dedicated to the task of aggregating states. Thus, a great dimension reduction is reached. Given the four features used, a four-dimension space is made, and supposing only ten levels in each dimension the number of states would be 10^4 , which is much bigger than that recommended for tabular methods in RL [9]. However, the quantisation process achieved by SOM has reduced the number of states to 152 for non-VIP customers and to 253 for VIP ones.

²A customer is considered to be VIP when it belongs to a private group, which is supported by the company itself. VIP customers have a number of advantages and privileges in their interaction with the company.

Furthermore, from Fig. 4 and Fig. 5 it can be derived the defining features of each state, i.e., the amount of purchased products and their average prices. In both figures dark colours mean high amount of purchases and high prices whereas light areas mean the opposite. This is very useful for the further study of the policies.

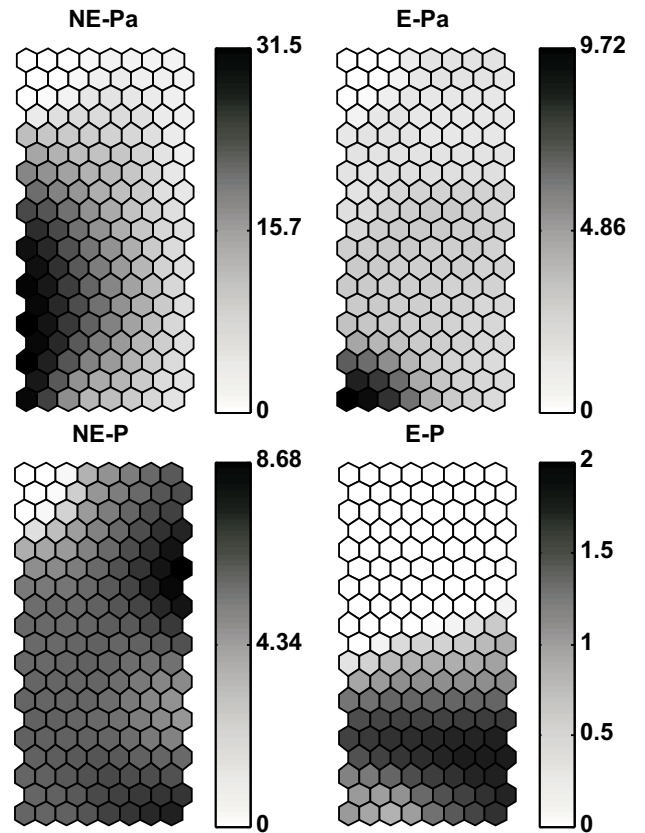


Fig. 4. Results achieved when training an SOM with non-VIP customers.

B. Optimal policies

The data set used in this work was obtained from real transactions carried out by the company. This involves a handicap because the data set is biased by the real policy which

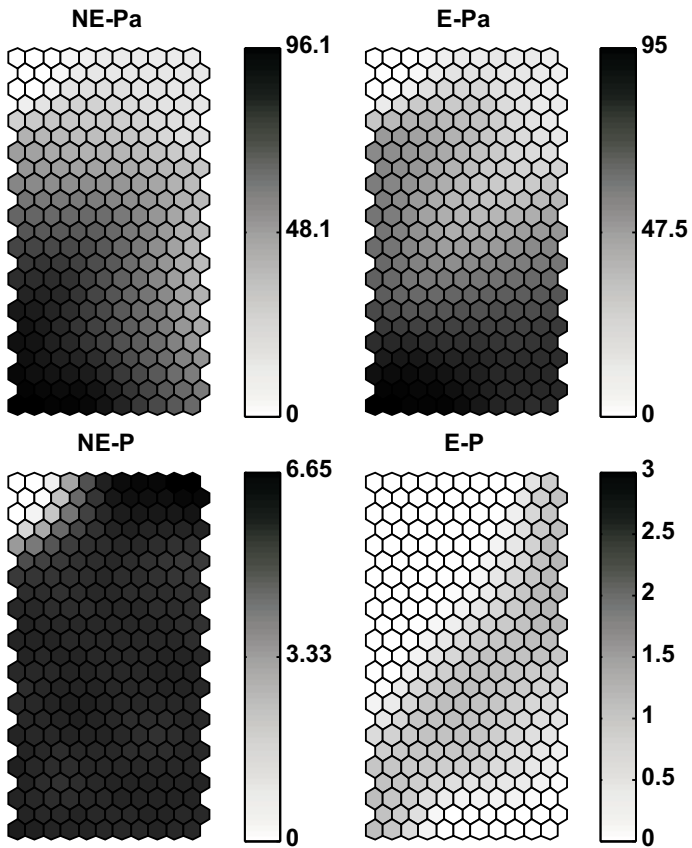


Fig. 5. Results achieved when training an SOM with VIP customers.

is very deterministic. However, some exploratory actions can be extracted from the data in order to obtain a better policy.

Results obtained in this section show the real policy followed by the company and the modifications suggested by the Q-learning algorithm. These results are presented in a hexagonal topology map to be able to infer relationships between the improvements in policies and the features of the states. The performance of the algorithm is shown for both VIP and non-VIP customers.

Fig. 6 (a) and Fig. 7 (a) show the policy followed by the company; the darker the colour, the lower the credit assignment. Fig. 6 (b) and Fig. 7 (b) show the difference in credit assignment between the optimal policy and the one followed by the company; in particular, dark colours mean that the optimal policy give less credits than the real one, whereas light colours stand for a larger credit assignment by the optimal policy than the real one.

A brief study of the real policies confirms that the guidelines of the campaign stated at the beginning of Section IV are right. VIP customers are rewarded with more credits since they buy more articles than non-VIP customers (see the values in the colorbars in Fig. 4 and Fig. 5). Nevertheless, it is more interesting to study the differences between both policies, thus finding a relationship between these differences and the features of each state.

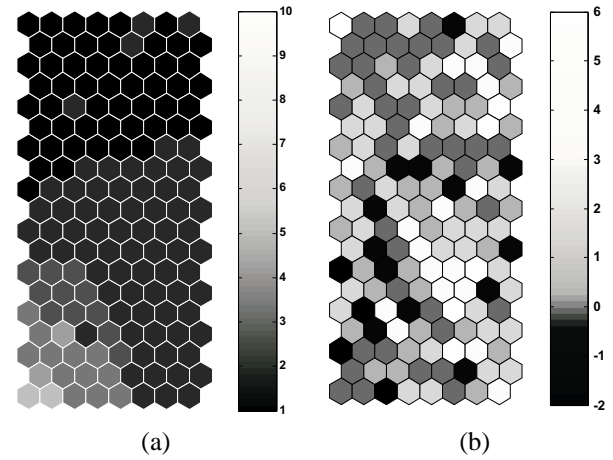


Fig. 6. Policy followed by the company for non-VIP customers (a) and modifications suggested by the optimal policy (b). It can be viewed in (a) that a poorer credit assignment is achieved in almost all states.

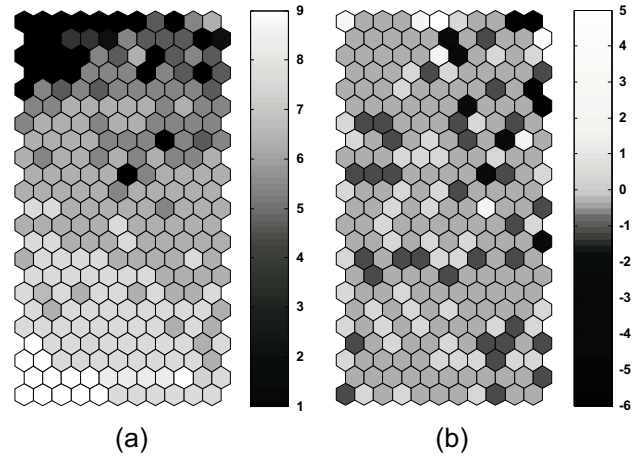


Fig. 7. Policy followed by the company for VIP customers (a) and modifications suggested by the optimal policy (b). Here, the policy followed by the company gives more credits than in the case of Fig. 6

Some conclusions could be extracted from Figs. 6 and 7:

- 1) **Non-VIP Clients:** Two dark areas can be distinguished in Fig. 6 (b). These zones are located in the upper left corner and lower left corner of the map. The former is related to sporadic customers that buy few articles and just the cheaper ones. While the real policy gives them a small amount of credits, the optimal policy advises the company to give them no credit because it will not produce any effect to the customer. The lower-left corner stands for customers that purchase many products with many different prices. Now the optimal policy advises to give less credits because these customers are already loyal and there is no need to give great rewards to them. On the right side of the map, it is observed a zone with light colours, i.e., more credit assignment is suggested. This area is related with the SOM feature of the encouraged price (E-P) feature and belongs to states

with high averaged prices. The optimal policy suggests to assign more credits to customers in these states.

- 2) **VIP Clients:** In this case, optimal policy and company policy are more similar than for non-VIP clients. This happens because the company policy has a strongly deterministic behaviour with these costumers which involves only a few explorative actions. The differences between both policies are very smooth in the whole map with the exception of the upper-right corner, where a small dark area is shown. The reason of these slight difference stems from the E-P feature, that has very low values in that zone, thus making the optimal policy give no credit to customers in those states.

To sum up, it seems that an optimal policy has to take into account not only the amount of purchases but also the prices of the purchases carried out by the customers. This last property is not taken into consideration by the company policy, as can be verified by comparison of Figs. 4 and 5 with Figs. 6 and 7.

C. Profits increase

Finally, it must be checked that the suggested modifications in the policy lead to an improvement in profits for the company. This can be made by comparing the $Q(s, a)$ function computed for both real and optimal policies (Fig. 8).

Results achieved show how the increase in profits is more relevant for non-VIP customers. This is related with the improvements suggested by the optimal policies. For non-VIP customers more changes were carried out in the policy, which implied a greater improvement in the Q function. On the contrary, both policies were similar for VIP customers, which makes the Q values very similar, as well.

The histograms in Fig. 9 show how the most usual improvement in profits is about 30 % for non-VIP customers and about 20 % for VIP ones. Once again, it is shown how improvement in profits gets higher values for non-VIP customers than for VIP ones. Moreover, some states reach even to double its profits. The table I shows a segmentation of all the customers of the company depending on the profit improvements.

VI. CONCLUSIONS AND FURTHER WORK

A new model that combines SOM with RL has been developed to improve a marketing campaign. The use of SOM has allowed to avoid the drawback of dimensionality and moreover has shown to be very useful in order to explain the results achieved. This way, several improvements to the company policy have been suggested in order to increase overall profits. The differences between the company policy and the optimal one have been related to the specific features of the modified states given by SOM. As a result of this approach, policies which consider more features were developed.

These changes in the policies would lead to an increase in profits, because they get customers to purchase more articles, which is the main goal of the company. However, it has not been possible to carry out a validation of the model by the company, which is needed as a final test.

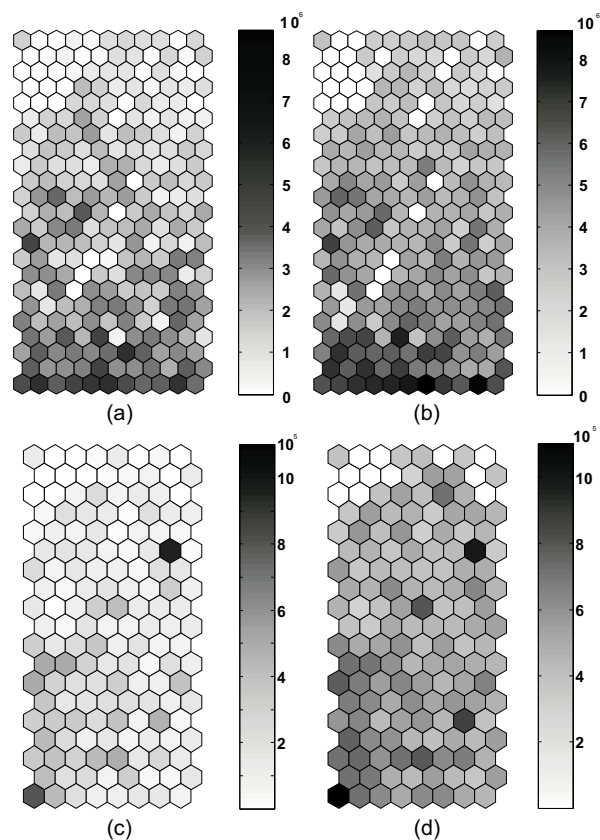


Fig. 8. Q values for real policies, (a) for VIP customers and (c) for non-VIP customers; and for optimal policies, (b) for VIP customers and (d) non-VIP customer. The darker the colour, the higher the value of Q function. High values of Q function involve more profits.

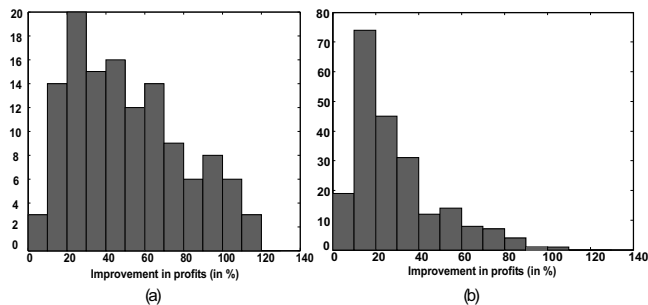


Fig. 9. Histograms showing relative profit increase in non-VIP customers (a) and VIP customers (b).

Further work will be dedicated to validate this model and use new features in the training of SOM. This task will require new marketing concepts which will allow the algorithm to learn more about the behaviour of several kinds of customers.

In addition, other RL algorithms can be used to solve problems like the one shown in this work. This way, it would be useful to test the performance of RL with classical function approximation for aggregating states [9], or other algorithms based on explorations in the policy space [10], [3].

TABLE I
DISTRIBUTION OF PROFIT IMPROVEMENT.

CLIENT	<20%	20-40 %	40-60 %	60-80 %	80-100 %	>100 %
NON-VIP	10.88 %	19.6 %	15.68 %	21.56 %	14.7 %	13.01 %
VIP	34.54 %	44.53 %	11.32 %	5 %	4.09 %	0.45 %

REFERENCES

- [1] N. Abe, E. Pednault, H. Wang, B. Zadrozny, F. Wei, and C. Apte. Empirical comparison of various reinforcement learning strategies for sequential targeted marketing. In *Proceedings of the ICDM*, pages 3–10, 2002.
- [2] N. Abe, N. Verma, R. Schroko, and C. Apte. Cross channel optimized marketing by reinforcement learning. In *Proceedings of the KDD*, pages 767–772, 2004.
- [3] Leemon Baird and Andrew Moore. Gradient descent for general reinforcement learning. In *Proceedings of the 1998 conference on Advances in neural information processing systems II*, pages 968–974, Cambridge, MA, USA, 1999. MIT Press.
- [4] T. Kohonen. The self-organizing map. In *Proceedings of IEEE*, number 78, pages 1464–1480, 1990.
- [5] T. Kohonen. *Self-Organizing Maps*. Springer, 1995.
- [6] E. Pednault, N. Abe, and B. Zadrozny. Sequential cost-sensitive decision making with reinforcement learning. In *Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 259–268. ACM, 2002.
- [7] Andrew James Smith. Applications of the self-organising map to reinforcement learning. *Neural Netw.*, 15(8-9):1107–1124, 2002.
- [8] P. Sun. *Constructing Learning Models from Data: The Dynamic Catalog Mailing Problem*. PhD thesis, Tsinghua University, May 2003.
- [9] Richard S. Sutton and Andrew G. Barto. *Reinforcement Learning: An Introduction*. MIT Press, 1998.
- [10] R.S. Sutton, S. McAllester, D. ansd Singh, and Y. Mansour. Policy gradient methods for reinforcement learning with function approximation. In *Proceedings of Advances in Neural Information Processing Systems 12*, pages 1057–1063. The MIT Press, 2000.