24th European Conference on Artificial Intelligence - ECAI 2020
Santiago de Compostela, Spain

Contextual Q-Learning

Tiago Pinto[1] and **Zita Vale**[2]

Abstract. This paper highlights a new learning model that introduces a contextual dimension to the well-known Q-Learning algorithm. Through the identification of different contexts, the learning process is adapted accordingly, thus converging to enhanced results. The proposed learning model includes a simulated annealing (SA) process that accelerates the convergence process. The model is integrated in a multi-agent decision support system for electricity market players negotiations, enabling the experimentation of results using real electricity market data.

1 CONTEXTUAL Q-LEARNING

Let's consider an agent that operates in an environment conceptualized by a set of possible states, in which the agent can choose actions from a set of possible actions. Each time that the player performs an action, a reinforcement value is received, indicating the immediate value of the resulting state transition. Thus, the only learning source is the agents' own experience, whose goal is to acquire an actions policy that maximizes its overall performance.

The methodology introduced in [3] proposes an adaptation of the Q-Learning algorithm [4] to undertake the learning process. Q-Learning is a very popular reinforcement learning method. It allows the autonomous establishment of an interactive action policy. It is demonstrated that the Q-Learning algorithm converges to the optimal proceeding when the learning Q state-action pairs is represented in a table containing the full information of each pair value [4]. The basic concept behind the proposed Q-Learning adaption is that the learning algorithm can learn a function of optimal evaluation over the whole space of context-scenario pairs (*c x s)*. This evaluation defines the Q confidence value that each scenario can represent the actual encountered negotiation scenario *s* in context *c*. The Q function performs the mapping as in (1):

$$Q: c\ x\ s \rightarrow U \tag{1}$$

where *U* is the expected utility value when selecting a scenario *s* in context *c*. The expected future reward, when choosing the scenario *s* in context *c,* is learned through trial and error according to (2):

$$Q_{t+1}(c_t, s_t) = Q_t(c_t, s_t) + \alpha(c_t, s_t)[r_{s.c.t} + \gamma \cdot U_t(c_{t+1}) - Q_t(c_t, s_t)] \tag{2}$$

where c_t is the kind of context when performing under scenario s_t at time *t:*

- $Q_t(c_t, s_t)$ represents the value of the previous iteration (each iteration represents each new contract established in the given scenario and context). Generally, the Q value is initialized to 0.
- $\alpha(c_t, s_t)$ $(0 < \alpha \leq 1)$ is the learning rate which determines the extent to which the newly acquired information will replace the old information (e.g. assuming a value of 0 learns nothing; on the other hand, a value of 1 represents a fully deterministic environment).
- $r_{s.c.t}$ is the reward, which represent the quality of the pair context-scenario *(c x s)*. It appreciates the positive actions with high values and negative with low values, all of them are normalized on a scale from 0 to 1. The reward *r* is defined in (3):

$$r_{s.c.t} = 1 - |RP_{c.t.a.p} - EP_{s.c.t.a.p}| \tag{3}$$

where $RP_{c.t.a.p}$ represents the real price that has been established in a contract with an opponent *p*, in context *c*, in time *t*, referring to an amount of power a; and $EP_{s.c.t.a.p}$ is the estimation price of scenario that corresponds to the same player, amount of power and context in time t. All *r* values are normalized in a scale from 0 to 1.

- γ $(0 \leq \gamma \leq 1)$ is the discount factor which determines the importance of future rewards. A value of 0 only evaluates current rewards, and higher values than 0 takes into account future rewards.
- $U_t(c_{t+1})$ is the estimation of the optimal future value which determines the utility of scenario *s*, resultant in context *c*. U_t is calculated as in (4):

$$U_t(c_{t+1}) = \max Q(c_{t+1}, s) \tag{4}$$

The Q-Learning algorithm is executed as follows:

- For each c and s, initialize $Q(c, s) = 0$;
- Observe new event (new established contract);
- Repeat until the stopping criterion is satisfied:
  - Select new scenario for current context;
  - Receive immediate reward $r_{s.c.t}$;
  - Update $Q(c, s)$ according to (2);
  - Observe new context c';
  - $c \rightarrow c'$.

After each update, all Q values are normalized according to the equation (5), to facilitate the interpretation of values of each scenario in a range from 0 to 1.

$$Q'(c, s) = \frac{Q(c, s)}{max[Q(c, s)]} \tag{5}$$

The proposed learning model assumes the confidence of Q values as the probability of a scenario in a given context. $Q(c, s)$ learns by treating a forecast error, updating each time a new observation (new established contract) is available again. Once all pairs context-scenario have been visited, the scenario that presents the highest Q value, in the last update, is chosen by the learning algorithm, to identify the most likely scenario to occur in actual negotiation.

[1] GECAD Research group, Polytechnic Institute of Porto, Portugal, email: tcp@isep.ipp.pt
[2] Polytechnic Institute of Porto, Portugal, email: zav@isep.ipp.pt

2 SIMULATED ANNEALING PROCESS

SA is an optimization method that imitates the annealing process used in metallurgy. The final properties of this substance depend strongly on the cooling schedule applied, i.e. if it cools down quickly the resulting substance will be easily broken due to an imperfect structure, if it cools down slowly the resulting structure will be well organized and strong. When solving an optimization problem using SA the structure of the substance represents a codified solution of the problem, and the temperature is used to determine how and when new solutions are perturbed and accepted. The algorithm is basically a three steps process: perturb the solution, evaluate the quality of the solution, and accept the solution if it is better than the previous one [1].

The two main factors of SA are the decrease of the temperature and the probability of acceptance. The temperature only decreases when the acceptance value is greater than a stipulated maximum. This acceptance number is only incremented when the probability of acceptance is higher than a random number, which allows some solutions to be accepted even if their quality is lower than the previous. When the condition of acceptance is not satisfied, the solution is compared to the previous one, and if it is better, the best solution is updated. At high temperatures, the simulated annealing method searches for the global optimum in a wide region; on the contrary, when the temperature decreases the method reduces the search area. This is done to try to refine the solution found in high temperatures. This is a good quality that makes the simulated annealing a good approach for problems with multiple local optima. SA, thereby, does not easily converge to solutions near the global optimum; instead this algorithm seeks a wide area always trying to optimize the solution. Thus, it is important to note that the temperature should decrease slowly to enable exploring a large part of the search space. The considered stopping criteria are: the current temperature and the maximum number of iterations. In each iteration is necessary to seek a new solution, this solution is calculated according to (6).

$$new\ solution = solution + S \times N(0,1) \quad (6)$$

solution in (1) refers to the previous solution, because this may not be the best found so far. $N(0,1)$ is a random number with a normal distribution, the variable S is obtained through (7).

$$S = 0.01 \times (upbound - lwbound) \quad (7)$$

$upbound$ and $lwbound$ are the limits of each variable, which prevent from getting out of the limits of the search problem.

The decisive parameters in SA's research are the decrease of temperature and the likelihood of acceptance. 4 variations of the SA algorithm have been implemented, combining different approaches for calculating these two components. It is expected that this will bring different results for different groups, as these components introduce a strong randomness in SA, which makes them reflect in the final results.

Table 1. Temperature and probability of acceptance calculation

group	temperature decreasing	probability of acceptance
1	$T_i = T_{i-1} \times \alpha$	$P = (2\pi T)^{-\frac{D}{2}} e^{\left(\frac{-\Delta x}{K \times T}\right)}$
2	$T_i = \frac{T_0}{i}$	$P = \frac{T_0}{(\Delta x^2 + T^2)^{\frac{(D+1)}{2}}}$
3	$T_i = T_0 e^{-ci^{\frac{1}{D}}}$	$P = \prod_{d=1}^{D} \frac{1}{2(\lvert y_d \rvert + Ti)\ln\left(1+\frac{1}{T_i}\right)}$
4	$T_i = T_0 \times \alpha^i$	$T_i = \frac{1}{1 + e^{\frac{\Delta x}{T_{max}}}}$

where:

$\alpha = 0.95$; i is the current iteration; $\Delta x = y(x^{max} - x^i)$ is the difference between best solution and current solution; $K = 1$ is the Boltzmann constant ; $T_0 = 1$ is the initial temperature; D is the number of variables; $c = 0.1$; $|y_d|$ is the abs of solution current; $T_{min} = 1 \times 10^{-10}$; $acceptance_{max} = 15$.

3 RESULTS

A historical database based on real data extracted from MIBEL - the Iberian Electricity Market is used to assess the results of the proposed model. The dataset is composed by the bilateral contracts declared in MIBEL, between 1 July 2007 and 31 October 2008 [2].

Table 2 shows the learning results of the proposed model against several benchmark reinforcement learning algorithms considering two distinct negotiation contexts. Results show that the proposed method achieves the lowest prediction errors in all contexts, resulting from the context aware learning capability.

Table 2: Average prediction errors in different contexts

Context	Algorithm	MAE	MAPE (%)	STD
1	Proposed Model	7.45	9.89	8.98
	Std. Q-Learning	11.24	16.28	14.04
	Roth-Erev	10.49	14.87	13.41
	UCB1	15.36	21.04	18.93
	EXP3	18.56	24.90	21.39
2	Proposed Model	4.28	6.46	5.89
	Std.Q-Learning	4.88	7.23	6.68
	Roth-Erev	4.46	6.83	6.03
	UCB1	5.89	9.31	8.72
	EXP3	6.53	10.85	9.38

4 CONCLUSIONS

The proposed model improves the standard Q-Learning algorithm by including a contextual dimension, thus providing a contextual aware learning model. A simulated annealing process is also included in order to enable accelerating the convergence process when needed, especially when the number of observations is low. Results show that the proposed model is able to undertake context-aware learning, surpassing the results of several benchmark learning algorithms when it comes to contextual learning.

ACKNOWLEDGEMENTS

This work has received funding from the EU Horizon 2020 research and innovation program under project DOMINOES (grant agreement No 771066) and from FEDER Funds through COMPETE program and from National Funds through FCT under projects CEECIND/01811/2017 and UIDB/00760/2020

REFERENCES

[1] Chen, C. Xudiera, and J. Montgomery, “Simulated annealing with thresheld convergence,” Evolutionary Computation (CEC), 2012 IEEE Congress on. pp. 1–7, 2012

[2] OMIE.. http://www.omie.es/files/flash/ResultadosMercado.html/, ejecucioncbfom 2018. [Online; accessed March-2019].

[3] Pinto, T., Vale, Z. “Contextual simulated annealing Q-learning for pre-negotiation of agent-based bilateral negotiations”, Proc. EPIA 2019

[4] Watkins, P. Dayan. 1992. “Q-learning. Machine Learning”. Machine Learning