geoconvention
Calgary • Canada • May 15-17 2023

Optimal OBN acquisition design based on reinforcement learning approach

Yi Guo, Rongzhi Lin and Mauricio Sacchi

Signal Analysis and Imaging Group (SAIG), Department of Physics, University of Alberta

Summary

Seismic acquisition costs are directly associated with the number of sensors used in the survey. This work explores an optimal design method for ocean bottom node (OBN) detector deployment. The proposed method is based on a reinforcement learning approach. A Markov Decision Process (MDP) is formulated following reinforcement learning theory with the sensor placement configuration. Then, the sensor selection procedure entails using Q-learning to find the sensor configuration that maximizes the reconstruction quality. The optimal policy learned by the Q-learning achieves more rewards than the random policy. Further, the RL-based sensor placement method performs better than random sensor selection methods.

Introduction

Optimal sensor deployment entails finding the best spatial distribution of sensors to adequately sample a physical process. For instance, in seismic data acquisition, we distribute receivers on the earth's surface to measure seismic wavefields generated by sources. The optimal sensor deployment problem is an NP-hard problem. Thus, approaches based on a brute-force search that examines every sensor position are impractical (Wang et al., 2019). Specifically, many researchers attempt to solve this NP-hard optimal survey design problem in seismic data acquisition by different means. For instance, Nakayama et al. (2019) propose integrating a genetic algorithm (GA) and a convolutional neural network (CNN) to automatically provide acquisition parameters that define the source blending and the spatial sampling of the sensors. Similarly, Guo and Sacchi (2020) suggest a QR decomposition with the column pivoting method based on a pre-learned basis library to find the optimal sensor locations in a time-lapse seismic application. These approaches provide feasible alternatives to reduce the number of sensors one should deploy in seismic acquisition.

Reinforcement learning (RL) is a subfield of machine learning. It is concerned with deciding how agents should take action in a given environment to learn an optimal or nearly-optimal policy that leads to the most significant long-term rewards. RL permits making optimal decisions using experience and trial-and-error; it assigns rewards or punishments fed back into the agent, and this way, RL can learn the best way the agent can navigate an environment (Sutton and Barto, 2018; Mnih et al., 2015; Silver et al., 2016). A few attempts have been made to apply RL in geophysics. For example, Ma et al. (2019) investigate automatic first-arrival picking based on RL. Sun and Alkhalifah (2020) use RL to determine the proper time to switch between different misfit functions for Full-Waveform Inversion (FWI). More recently, Dell'Aversana (2022) combines geophysical inversion with RL using Q-learning. RL has recently been investigated in the previously mentioned fields but has yet to be tested for a seismic sensor placement application.

Our problem tests the idea of adopting RL for seismic sensor deployments. In essence, RL is used to find the optimal receiver configuration that maximizes the quality of the reconstruction. Our problem poses RL via a particular algorithm named Q-learning (Watkins and Dayan, 1992), and we provide numerical experiments to evaluate the feasibility of RL for seismic sensor deployment.

Theory

The core problem we want to tackle is the sensor selection problem. For this purpose, we adopt RL to find the optimal receiver configuration. RL is theoretically based on the Markov Decision Process (MDP) framework (Sutton and Barto, 2018). An MDP is a problem formulation that defines how an agent takes sequential actions from states in its environment, guided by rewards – using uncertainty in how it transitions from state to state. An MDP is denoted as a tuple (S,A,R,P), where S is the state space, A is the action space, R is the reward function, and P is the state transition probability. The policy of MDP is defined as a function π, that is, a probability distribution in the state-action space. The objective is to estimate the optimal policy π^* that satisfies

$$J_{\pi^*} = \max_{\pi} J_\pi = \max_{\pi} E_\pi \left[\sum_{t=0}^{\infty} \gamma^t r_t \right], \tag{1}$$

where $\gamma \in [0,1)$ is the discount factor, r_t is the reward at time-step t, $E_\pi[.]$ stands for the expectation under policy π, and J_π is the expected cumulative reward. The following illustrates how to mathematically design the optimal sensor placement as a formulation of an MDP concerning state space, action space, and reward.

For the state space, considering the methodology we adopted is classic RL rather than deep RL, the state space fed to the algorithm is limited. So, a criterion arises from making the possible combination reduced to a smaller number. For the optimal receiver selection problem, we first divide all the candidate positions into K ranges, where K is the number of sensors we want to select. The selected sensors can only be chosen in predefined ranges and cannot cross boundaries. This is a practical restriction because all the selected sensors are assumed equal, so there is no difference if two sensors interchange their positions. Strictly speaking, this criteria is beneficial in that not only the NP-hard optimization problem size becomes smaller, but also significant design gaps are avoided, similar to the sampling strategy called jittered sampling (Hennenfent and Herrmann, 2008).

The action space in the sensor placement problem is defined as selecting the location of a sensor from all the candidate positions. More specifically, how to choose an action a comprised of two steps: activate a sensor and move left or right. Moving left or right means selecting the activated sensor' left or right candidate position. Notice in each action, we only change one sensor's position. After executing action a_t in state s_t, the system transits to a new state s_{t+1}. Afterward, by comparing the reconstructed data of the new state s_{t+1} with the previous state s_t according to the objective function, the algorithm can judge whether an action is appropriate.

The reward is the value received after completing a specific action a_t at a given state s_t, which is the immediate reward obtained for a single action. In the seismic sensor placement problem, we want to find the best possible sensor locations combination to obtain as much reward as possible. The one-step reward $r(s_t,a_t)$ is evaluated based on the reconstruction quality, which is compared between state s_t and state s_{t+1}. If the new data reconstruction quality corresponding to a new state is better than the previous state, the reward is $+10$. Otherwise, the reward (or the penalty) is -2. As the agent executes time steps, it accumulates a reward at each time step. Additionally, the algorithm cares about cumulative rewards rather than individual rewards, and the discount factor γ controls how myopic the agent is in its decision-making.

Q-learning is the classic and most adopted RL method suitable for the sensor placement problem. In general, Q-learning belongs to Temporal Difference (TD) learning that can achieve optimal policies from delayed rewards when the agent has no prior knowledge of the unknown environment. At a specific time step t, the agent observes the state s_t and then chooses an action a_t according to the ε-greedy strategy, which executes an intensified search by exploitation and a diversified search by exploration, making it an efficient method for this NP-hard problem.

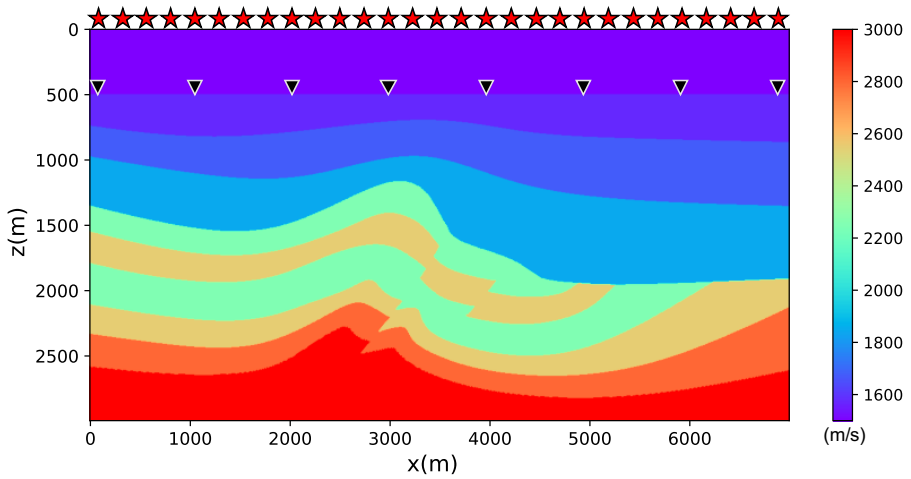

Figure 1: The velocity model was utilized to simulate ocean bottom node data via finite difference modeling. Red stars represent the sources, and black triangles represent the receivers.

The core algorithm of Q-learning is a Bellman equation as a value iteration update, using the weighted average of the old and new information:

$$Q^*(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha(r_{t+1} + \gamma \max_a Q(s_{t+1}, a) - Q(s_t, a_t)), \tag{2}$$

where $Q^*(s_t, a_t)$ is the expected value (cumulative discounted reward) of doing action a in state s at step t and then following the optimal policy. γ is the discount factor, and α is the learning rate.

Example

We use the acoustic finite-difference modeling from the SeismicJulia package (Stanton and Sacchi, 2016) to simulate a prestack marine OBN dataset for the SAIG velocity model. The reason why we use the OBN setting is that the price of each node is costly. Thus, finding an optimal location for the available nodes could be economically beneficial. The velocity model and the source-receiver geometry are shown in Figure 1. A total number of 350 sources (red stars) were simulated. Each source fires into a fixed array of 70 receivers (black triangles). The receivers are situated at 500-meter depth to simulate ocean bottom nodes. We select 24 out of the original 70 nodes for the optimal setting to test the proposed method. The parameter settings for the Q-learning iteration are as follows: learning rate $\alpha = 0.1$ and discount factor $\gamma = 0.7$. The ε-greedy strategy is applied to investigate the performance of the proposed algorithm. The exploration rate ε is set as 1.0 at the beginning and decreases gradually to 0.1. We run 1000 episodes while we iterate 100 time steps in each episode. The algorithm is converged, and we can find the optimal locations of the sensors following the learned policy.

We compare the performance of the RL-based optimization method and the state-of-the-art jittered sampling protocol as a benchmark. The jittered sampling we mention here only concerns the sampling strategy. Both sampling paradigms use the same number of sensors. Figure 2 shows a comparison of one CSG at position x = 3.5 km. The four CSGs shown in Figure 2 are the original CSG, optimal decimated CSG, reconstructed CSG with 24 receivers selected by jittered sampling, and the reconstructed CSG with 24 receivers located at the optimal positions selected by the RL algorithm, respectively. Note that the jittered sampling result shown in Figure 2(c) is the best-resulting protocol chosen from ten randomly initialized settings following the jittered sampling scheme. The RL-based method provides a better sensor placement scheme for reconstruction than jittered sampling because RL-based sampling always reaches the same optimal solution. In contrast, jittered sampling is a form of random sampling; hence, we cannot confirm when a jittered sampling realization is the best-stable

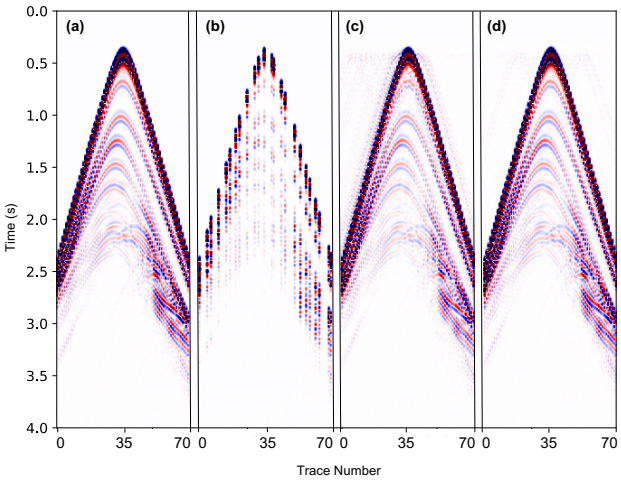

Figure 2: Comparison between the original data with the reconstructed CSG by jittered sampling and the proposed method. (a) Original CSG. (b) Optimal decimated CSG via RL-based method. (c) Reconstructed CSG via jittered sampling. (d) Reconstructed CSG via RL-based method.

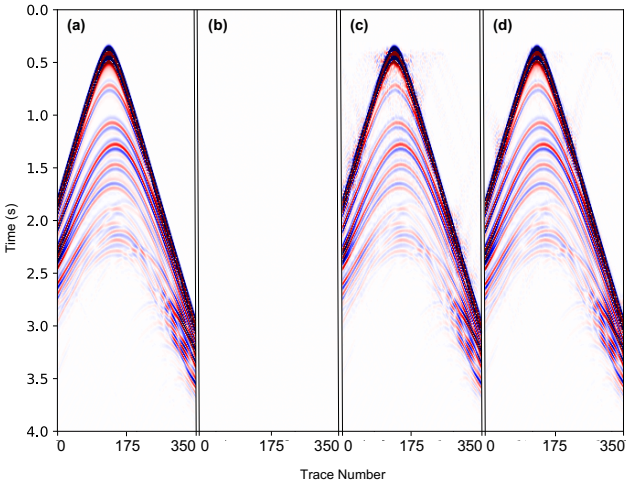

Figure 3: Comparison between the original data with the reconstructed CRG by jittered sampling and the proposed method. (a) Original CRG. (b) Optimal decimated CRG via RL-based method. (c) Reconstructed CRG via jittered sampling. (d) Reconstructed CRG via RL-based method.

option. Similarly, Figure 3 highlights a comparison of one CRG at position x = 2.7 km. The four CRGs shown in Figure 3 are the original CRG, optimal decimated CRG, reconstructed CRG with jittered sampling, and the reconstructed CRG with the proposed sampling, respectively. It is noticeable that even though we do not record the data at position x = 2.7 km with the optimal survey, we still can recover the data. What's more, the CRG recovered with the proposed sampling in Figure 3(d) shows better reconstruction quality than the jittered sampling scheme shown in Figure 3(c).

Furthermore, we also compare the optimal policy learned by the RL algorithm with the random policy. Random policy means that at each state, the agent selects actions randomly. Figure 4 shows the total reward gained per episode, and the blue and red dots represent the reward obtained by the Q-learning policy and random policy, respectively. We iterate 100 episodes to see the difference, and the mean reward for the Q-learning policy is 29.62, while the one for the random policy is 6.23. It is clear that in most cases, the optimal policy learned by Q-learning reaches a higher reward than the random policy, which verifies the effectiveness of the proposed method.

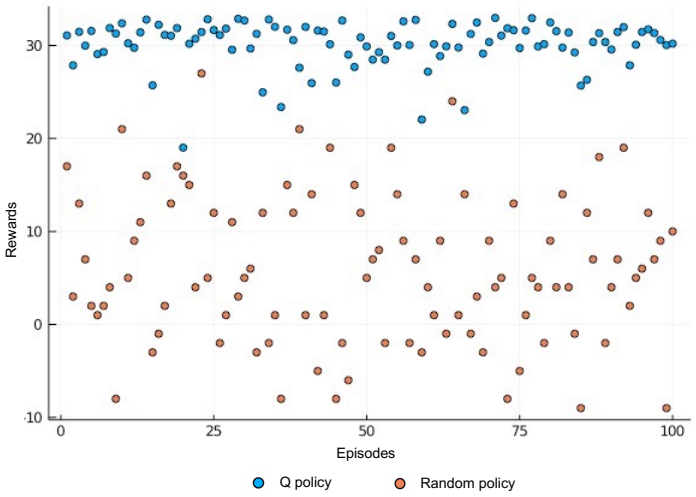

Figure 4: Comparison of total reward gained per episode between optimal policy and random policy.

Conclusion

This paper presents a new integrated RL-based optimal sensor placement method for seismic acquisition design. The sensor placement problem with the proposed objective function falls under the MDP formulation mathematically, where state space with a practical boundary restriction, action space, and reward is mainly defined for the seismic receiver selection problem. The Q-learning result is converged, and optimal sensors can be selected following the learned policy. The resultant synthetic data scenarios are shown in CSGs and CRGs demonstrating that the proposed method provides better sensor locations for reconstructing data than the jittered sampling scheme under the same circumstances.

Acknowledgements

The authors thank the Signal Analysis and Imaging Group sponsors at the University of Alberta.

References

Dell'Aversana, P., 2022, Combining geophysical inversion with reinforcement learning: 83rd EAGE Annual Conference & Exhibition, 1–5.

Guo, Y., and M. D. Sacchi, 2020, Data-driven time-lapse acquisition design via optimal receiver-source placement and reconstruction, *in* SEG Technical Program Expanded Abstracts 2020: Society of Exploration Geophysicists, 66–70.

Hennenfent, G., and F. J. Herrmann, 2008, Simply denoise: Wavefield reconstruction via jittered undersampling: Geophysics, **73**, V19–V28.

Ma, Y., T. Fei, and Y. Luo, 2019, A new insight into automatic first-arrival picking based on reinforcement learning: 81st EAGE Conference and Exhibition 2019, European Association of Geoscientists & Engineers, 1–5.

Mnih, V., K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski, et al., 2015, Human-level control through deep reinforcement learning: nature, **518**, 529–533.

Nakayama, S., G. Blacquière, and T. Ishiyama, 2019, Automated survey design for blended acquisition with irregular spatial sampling via the integration of a metaheuristic and deep learning: Geophysics, **84**, P47–P60.

Silver, D., A. Huang, C. J. Maddison, A. Guez, L. Sifre, G. Van Den Driessche, J. Schrittwieser, I. Antonoglou, V. Panneershelvam, M. Lanctot, et al., 2016, Mastering the game of go with deep neural networks and tree search: nature, **529**, 484–489.

Stanton, A., and M. D. Sacchi, 2016, Efficient geophysical research in julia: CSEG GeoConvention 2016, 1–3.

Sun, B., and T. Alkhalifah, 2020, A data-driven choice of misfit function for fwi using reinforcement learning: EAGE 2020 Annual Conference & Exhibition Online, European Association of Geoscientists & Engineers, 1–5.

Sutton, R. S., and A. G. Barto, 2018, Reinforcement learning: An introduction: MIT press.

Wang, Z., H.-X. Li, and C. Chen, 2019, Reinforcement learning-based optimal sensor placement for spatiotemporal modeling: IEEE transactions on cybernetics, **50**, 2861–2871.

Watkins, C. J., and P. Dayan, 1992, Q-learning: Machine learning, **8**, 279–292.