Combining Reinforcement Learning and Constraint Programming for Sequence-Generation Tasks with Hard Constraints - An Extended Abstract

Daphné Lafleur
Polytechnique Montréal, Canada
Quebec Artificial Intelligence Institute (Mila), Canada

Sarath Chandar
Polytechnique Montréal, Canada
Quebec Artificial Intelligence Institute (Mila), Canada
Canada CIFAR AI Chair

Gilles Pesant
Polytechnique Montréal, Canada

Abstract

In this work, we use Reinforcement Learning to combine Machine Learning (ML) and Constraint Programming (CP). We show that combining ML and CP allows the agent to reflect a pretrained network while taking into account constraints, leading to melodic lines that respect both the corpus' style and the music theory constraints.

2012 ACM Subject Classification Software and its engineering → Constraint and logic languages; Theory of computation → Reinforcement learning; Applied computing → Sound and music computing

Keywords and phrases Constraint programming, reinforcement learning, RNN, music generation

Digital Object Identifier 10.4230/LIPIcs.CP.2022.21

1 Introduction

Recurrent Neural Networks (RNNs) [3] are a class of Machine Learning (ML) algorithms renowned for their ability to extract structural information from a corpus in order to generate sequences that mimic said corpus' style. However, some sequence-generation tasks require the final output to respect a set of rules. Music generation, especially applied to Renaissance music, is a perfect example of these kinds of tasks.

2 Methodology

RL-Tuner [1] is an algorithm that uses Reinforcement Learning (RL) to bridge the gap between RNNs and hard constraints, resulting in samples that both better respect the constraints and are representative of the data the algorithm was trained on. The agent tries to maximize a reward function composed of (1) the RNN reward indicating the probability of a value returned by an RNN and (2) a hard constraint reward tuned manually for each rule. In this work we use Constraint Programming (CP) with Belief Propagation (BP) to learn better how to satisfy constraints. By using CP, we benefit from the interactions between all the constraints and may anticipate violations. Furthermore, the addition of BP is a recent advance in CP that provides marginal probabilities for every selected value in the sequence [2]. These marginals take into account the entirety of the sequence and act as a metric to determine if choosing an action could potentially lead to a complete sequence with no added constraint violations.

Table 1 Constraint satisfaction (sat) and average RNN reward with different reward functions. *Takeaway:* RNN + marginals + violations yields the best constraint satisfaction and its RNN reward is still higher than without using CP.

Reward function	sat (%)	RNN reward
1. violations	60.1	-4.9
2. marginals	75.1	-2.1
3. marginals + violations	76.9	-1.8
4. RNN + violations (similar to RL-Tuner)	77.2	-1.8
5. RNN + marginals	77.4	**-1.4**
6. RNN + marginals + violations	**81.6**	-1.9
7. RNN	74.5	-2.2

We illustrate our framework by applying it to contrapuntal music writing in the style of the Renaissance period. We design two similar CP models implementing the rules of counterpoint for melody writing according to a standard textbook [4] (one obtaining the marginals with BP, the other counting the number of violations). We train our RNN model using Bach chorales, a widely available corpus that is compatible to some degree with the constraints we enforce. We combine our CP and RNN models within RL-Tuner and analyze the evolution of two metrics to show that our algorithm retains its acquired knowledge from the Bach corpus while more-closely following the rules of counterpoint we added.

3 Experiments

Table 1 presents the final constraint satisfaction and average RNN reward for each of our reward functions. Since we have a different corpus and different constraints, we cannot compare our results directly with those in the RL-Tuner paper. However, line 4 allows us to compare to the RL-Tuner approach because we only count the number of violations (similar to summing rewards for each rule). We can see here that line 4 is better than line 7. However, we can also see that adding the marginals in the mix offers another improvement of 4%.

We wondered if CP and the RNN would counteract each other, the RNN wanting to stay close to the corpus and CP aiming to constrain the algorithm. On the contrary, combining them yields better constraint satisfaction than using them separately (lines 4 to 6 vs 1 to 3). The same can be said for the RNN reward. Yet again, it seems like the CP model generally helps the RNN to stylistically represent the corpus (lines 4 to 6 vs 1 to 3). This is due to concordance between our corpus and our constraints.

The reward function yielding the best constraint satisfaction is the `RNN + marginals + violations` function (line 6). Both CP models are necessary because the marginals provide information about how good the chosen note is to respect the constraints in the long run and the violations provide information about how big the constraint violation is.

4 Conclusion

In this work, we presented an extension of the RL-Tuner algorithm: adding the output of CP models to the reward function in order to learn hard constraints. We showed that combining ML and CP yields generated music sequences that are better at both reflecting the corpus and respecting constraints than using only ML (or CP). We hope that this method could be applied to other sequence generation tasks.

5 References

References

1 Natasha Jaques, Shixiang Gu, Richard E. Turner, and Douglas Eck. Tuning recurrent neural networks with reinforcement learning. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Workshop Track Proceedings*. OpenReview.net, 2017. URL: https://openreview.net/forum?id=Syyv2e-Kx.

2 Gilles Pesant. From support propagation to belief propagation in constraint programming. *J. Artif. Intell. Res.*, 66:123–150, 2019. doi:10.1613/jair.1.11487.

3 Robin M. Schmidt. Recurrent neural networks (rnns): A gentle introduction and overview. *CoRR*, abs/1912.05911, 2019. URL: http://arxiv.org/abs/1912.05911, arXiv:1912.05911.

4 Peter Schubert. *Modal Counterpoint, Renaissance Style*. Oxford University Press, 2nd edition, 2008.