# Two-Part Inventions by SARSA

Logo

Author's name(s)
Media Informatics Special Interest Group,
Affiliations

E-mail

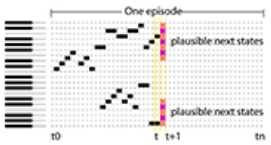
## A musical agent

learns to generate a two-part invention using SARSA. SARSA is a reinforcement learning technique that learns an optimal policy by sampling the state space to estimate the utility of state-action pairs Q(s,a) where s denotes a state , a denotes an action, r denotes a reward, alpha denotes the learning rate and gamma denotes the discount rate.

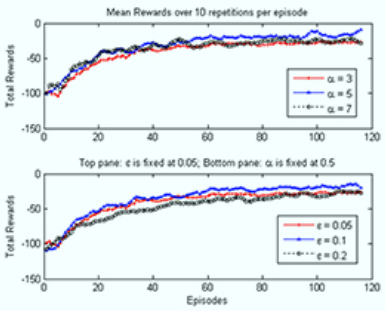
$$Q(s,a) \leftarrow Q(s,a) + \alpha[r + \gamma Q(s',a') - Q(s,a)]$$

- First, the policy was learned using hand-crafted rules describing the desired characteristics of two-part invections. These rulees could also be discovered using data mining techniques.
- Then, the rules acted as a critic's comments to the generated music. The musical agent would amend its policy based on these comments.

In our approach, each episode was a complete 32-bar two-part counterpoint. Form and other contexts were incorporated into the system via the critic's rules and the usage of context dependent Q-tables.

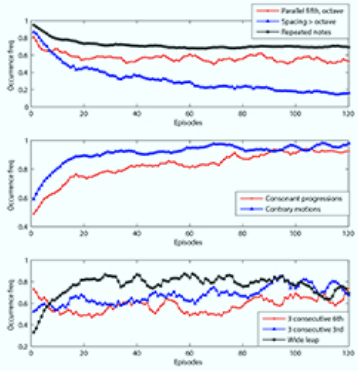



## Rewarding scheme , Parameters & Analysis




| Criteria | Reward value |
|---|---|
| Parallel fifth, octave | -0.1 |
| Crossing between parts | -0.1 |
| Spacing between voice more than one octave | -0.1 |
| Repeated notes | -0.1 |
| Repeated consonant major, minor third | -0.1 |
| Repeated consonant major, minor sixth | -0.1 |
| Wide leap interval | -0.1 |
| Dissonant progression second, tritone | -0.1 |
| Consonant progression major, minor third | 0.1 |
| Contrary motion | 0.1 |

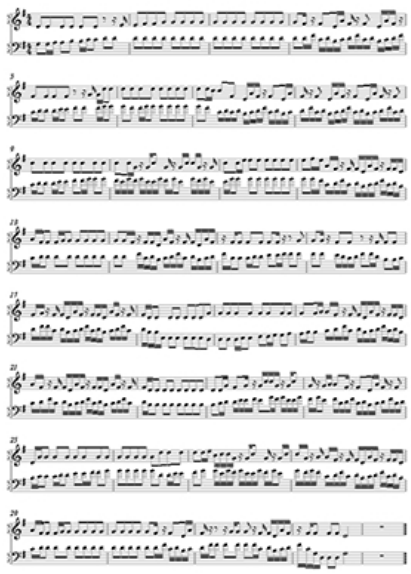
| SARSA Parameter Settings | | |
|---|---|---|
| Learning rate ($\alpha$) | 0.3, 0.5, 0.7 | 0.5 |
| Discount rate ($\gamma$) | 0.9 | 0.9 |
| ($\epsilon$)-greedy probability | 0.1 | 0.05, 0.1, 0.2 |
| Max-iteration | 512 | 512 |
| Max-episode | 120 | 120 |




## In this work,

we employed SARSA to generate 32-bar two-part invention pieces. By carefully selecting the representation of states, actions, rules and contexts, a complex problem such as algorithmic composition could be dealt with and reasonable output could be obtained with comparatively less effort.

**Tonal Counterpoint #1**



## SARSA

**SARSA**

Initialise $Q_c(s,a)$ for all possible contexts $C$ arbitrary
Repeat for each episode:
  Initialise $s$ for each $Q_c$
  Repeat for each step of episode:
    Choose $a$ according to policy $\pi(s)$ and context $c$
    Agent takes action $a$
    Observe $r$ from $s', a', c'$
    Update value function:
    $Q_c(s,a) \leftarrow Q_c(s,a) + \alpha[r + \gamma Q_c(s',a') - Q_c(s,a)]$
    $s \leftarrow s', a \leftarrow a', c \leftarrow c'$
  Until max step or until termination
Until max episode

## Policy learning

in RL is a powerful concept. An agent explores a partially observable environment until it learns a policy (i.e., how it should react to the environment) that maximises its return, $R$. The representation of the state space, S, and actions, A, are critical since they are the abstraction of behaviours to be learned. In further work, the following directions could be pursued:

(i) to improve the handcrafted rules for different composition,
(ii) to automate rules-acquisition process, and
(iii) to apply the approach to other genres (e.g., four part writing, jazz, etc).

## References

- Sutton, R.S. and Barto, A.G.: Reinforcement Learning: An Introduction. A Bradford Book, The MIT Press, 1998.
- Watkins, C.J. and Dayan, P.: Q-learning. Machine Learning 8:279:292, 1992.
- Somnuk Phon-Amnuaisuk: Generating Tonal Counterpoint Using Reinforcement Learning. ICONIP (1) 2009: 580-589