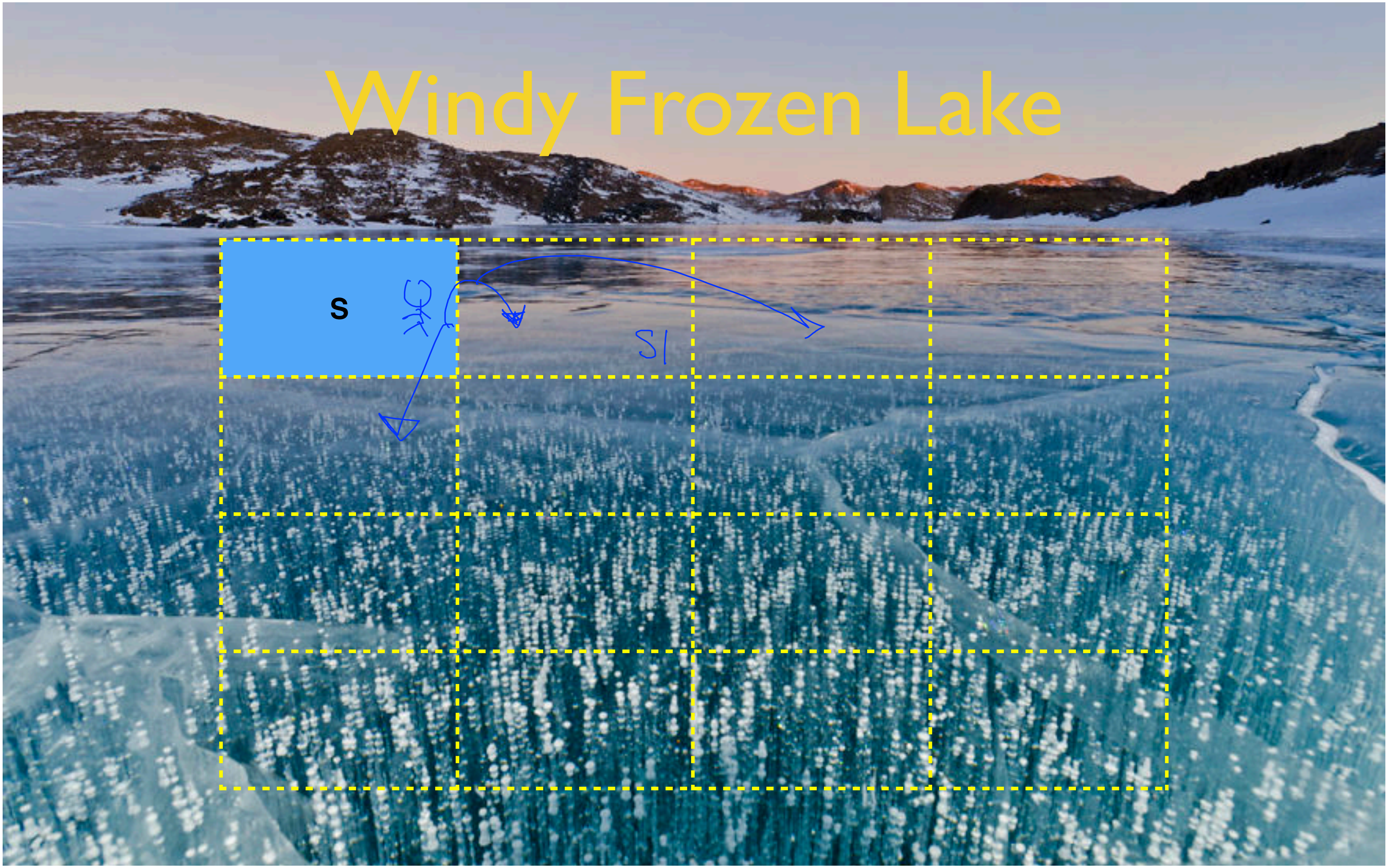
Lecture 5: Windy Frozen Lake Nondeterministic world!

Reinforcement Learning with TensorFlow&OpenAI Gym

Sung Kim <hunkim+ml@gmail.com>

Windy Frozen Lake



Deterministic VS Stochastic (nondeterministic)

- **In deterministic models** the output of the model is fully determined by the parameter values and the initial conditions initial conditions
- **Stochastic models** possess some inherent randomness.
  - The same set of parameter values and initial conditions will lead to an ensemble of different outputs.

Deterministic

```python
# Register FrozenLake with is_slippery False
register(
    id='FrozenLake-v3',
    entry_point='gym.envs.toy_text:FrozenLakeEnv',
    kwargs={'map_name' : '4x4', 'is_slippery': False}
)

env = gym.make('FrozenLake-v3')
```

S

```
SFFF
FHFH
FFFH
HFFG

SFFF
FHFH
FFFH
HFFG
  (Right)
('State: ', 1, 'Action: ', 2,
SFFF
FHFH
FFFH
HFFG
  (Right)
('State: ', 2, 'Action: ', 2,
SFFF
FHFH
FFFH
HFFG
```

```
  (Down)
('State: ', 6, 'Action: ', 1,
SFFF
FHFH
FFFH
HFFG
  (Down)
('State: ', 10, 'Action: ', 1,
SFFF
FHFH
FFFH
HFFG
  (Down)
('State: ', 14, 'Action: ', 1,
SFFF
FHFH
FFFH
HFFG
  (Right)
('State: ', 15, 'Action: ', 2,
('Finished with reward', 1.0)
```

Stochastic (non-deterministic)

```
# is_slippery True
env = gym.make('FrozenLake-v0')
```

S

```
SFFF
FHFH
FFFH
HFFG

SFFF
FHFH
FFFH
HFFG
  (Right)
('State: ', 0, 'Action: ', 2,
SFFF
FHFH
FFFH
HFFG
  (Right)
('State: ', 4, 'Action: ', 2,
SFFF
FHFH
FFFH
HFFG
  (Down)
('State: ', 5, 'Action: ', 1,
('Finished with reward', 0.0)
```

S

```
SFFF
FHFH
FFFH
HFFG
  (Right)
('State: ', 0, 'Action: ', 2,
SFFF
FHFH
FFFH
HFFG
  (Right)
('State: ', 1, 'Action: ', 2,
SFFF
FHFH
FFFH
HFFG
  (Right)
('State: ', 1, 'Action: ', 2,
SFFF
FHFH
FFFH
HFFG
  (Right)
('State: ', 5, 'Action: ', 2,
('Finished with reward', 0.0)
```

O — X

Stochastic (non-deterministic) worlds

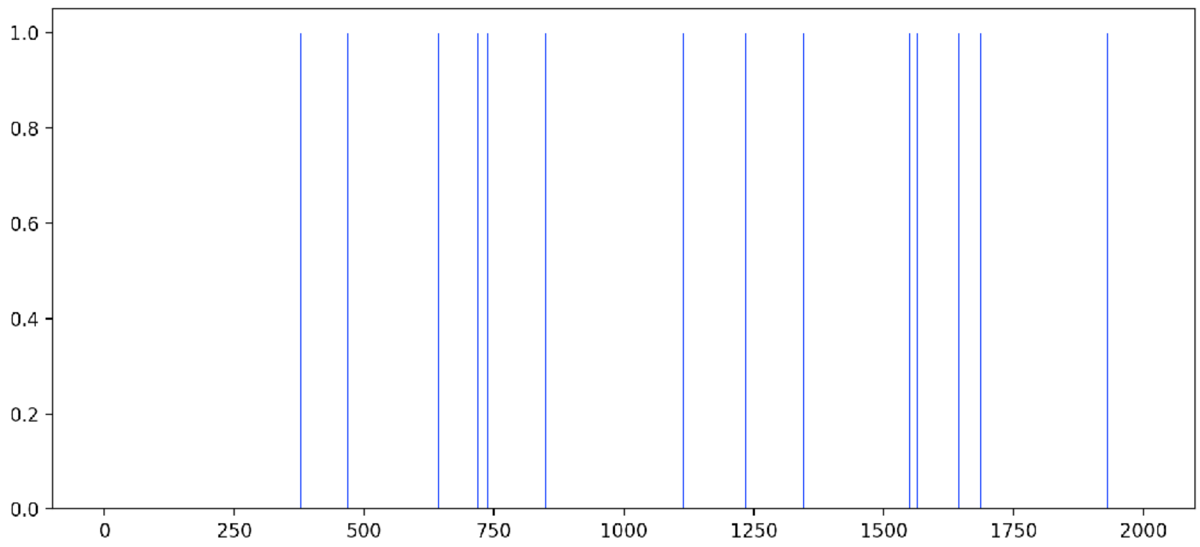
- Unfortunately, our Q-learning (for deterministic worlds) does not work anymore
- Why not?

$$\boxed{\hat{Q}(s,a) \leftarrow r + \gamma \max_{a'} \hat{Q}(s',a')}$$

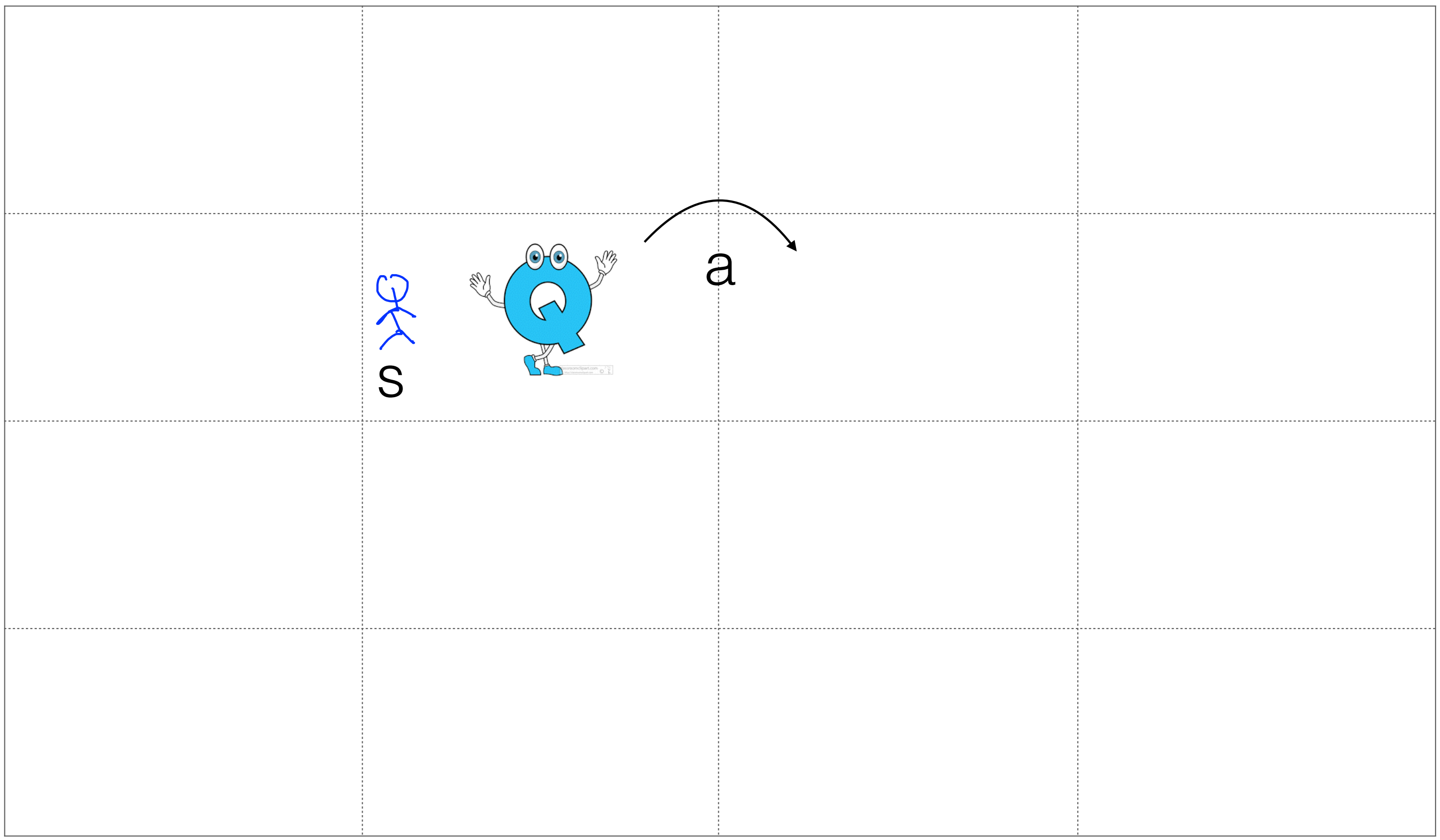
Our previous Q-learning does not work

```
env = gym.make('FrozenLake-v0')
```

Score over time: 0.0165



Why does not work in stochastic (non-deterministic) worlds?



Stochastic (non-deterministic) world

- Solution?
  - Listen to Q (s`) (just a little bit)
  - Update Q(s) little bit (learning rate)
- Like our life mentors
  - Don't just listen and follow one mentor
  - Need to listen from many mentors

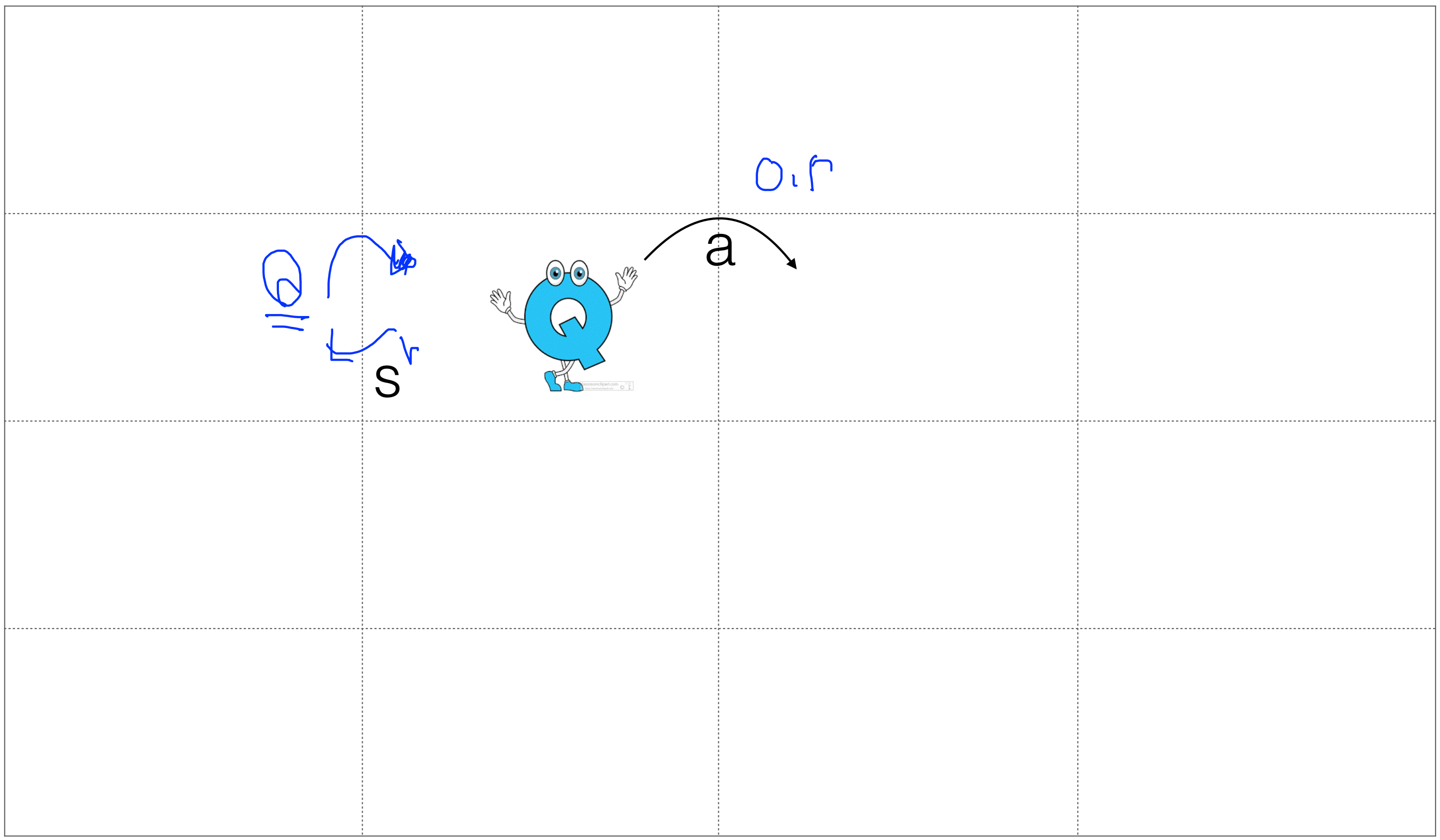
BC-HBR-WAKE-UP-CALL-MENTORS-NYTSF

Your Career Needs Many Mentors, Not Just One

20.1.2017 15.00

http://m.kauppalehti.fi/uutiset/your-career-needs-many-mentors--not-just-one/gp3Q4rTp

Stochastic (non-deterministic) world



Learning incrementally

$$Q(s,a) \leftarrow r + \gamma \max_{a'} Q(s',a')$$

α

- Learning rate, α
  - $\alpha = 0.1$

$$Q(s,a) \leftarrow (1-\alpha)\, Q(s,a) + \alpha\ [r + \gamma \max_{a'} Q(s',a')]$$

Learning with learning rate

$$Q(s,a) \leftarrow r + \gamma \max_{a'} Q(s',a')$$

$$\boxed{Q(s,a) \leftarrow (1-\alpha)Q(s,a) + \alpha[r + \gamma \max_{a'} Q(s',a')]}$$

Learning with learning rate

$$\boxed{Q(s,a) \leftarrow r + \gamma \max_{a'} Q(s',a')}$$

$$Q(s,a) \leftarrow (1-\alpha)Q(s,a) + \alpha[r + \gamma \max_{a'} Q(s',a')]$$

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

Q-learning algorithm

For each s, a initialize table entry $\hat{Q}(s,a) \leftarrow 0$

Observe current state s

Do forever:

- Select an action a and execute it — E&E
- Receive immediate reward r
- Observe the new state s'
- Update the table entry for $\hat{Q}(s,a)$ as follows:

$$Q(s,a) \leftarrow (1-\alpha)Q(s,a) + \alpha[r + \gamma \max_{a'} Q(s',a')]$$

- $s \leftarrow s'$

Machine Learning, T. Mitchell, McGraw Hill, 1997

Convergence

$\hat{Q}$ denote learner's current approximation to Q.

$$\hat{Q}(s,a) \leftarrow (1-\alpha)\hat{Q}(s,a) + \alpha[r + \gamma \max_{a'} \hat{Q}(s',a')]$$

Can still prove convergence of $\hat{Q}$ to Q [Watkins and Dayan, 1992]

Machine Learning, Tom Mitchell, 1997

Next

Lab: Stochastic worlds