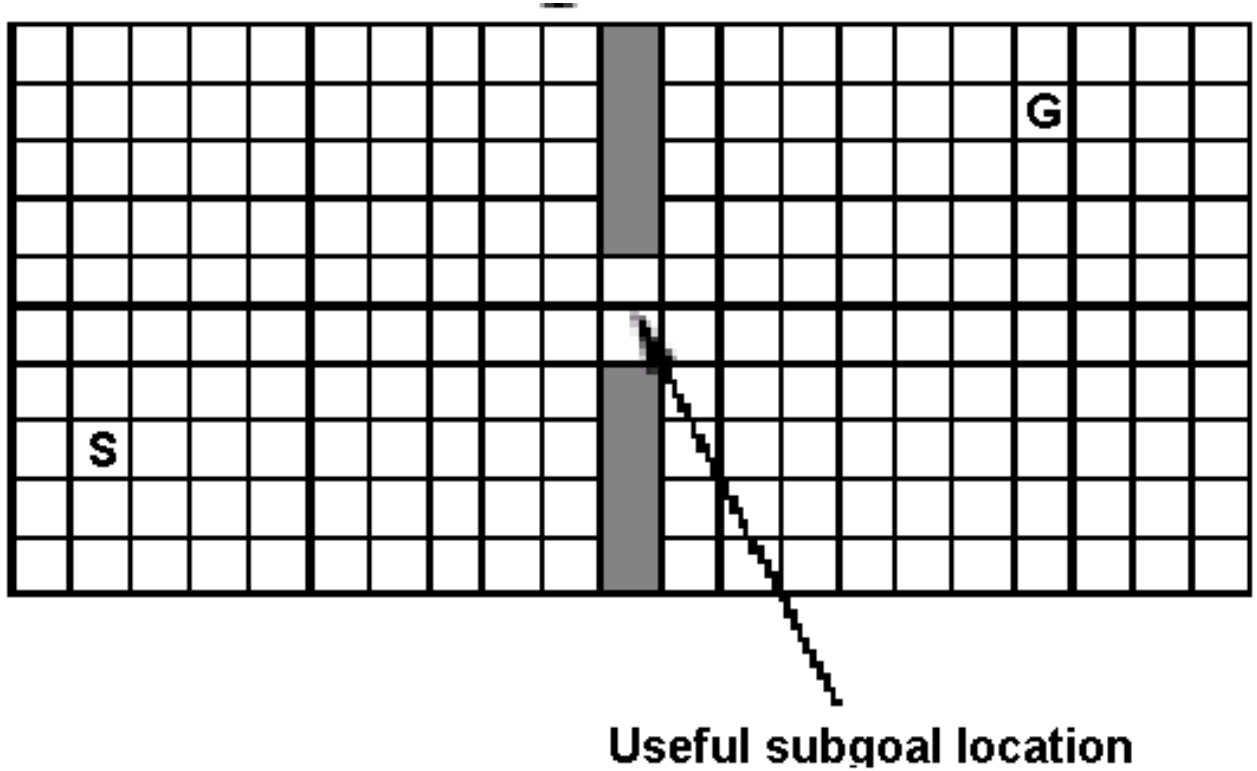
Automatically Learning Useful Subgoals/Options

Why Learn Subgoals?

- Hand defined options/subgoals may fail if:
  - there is no a priori knowledge to define options/subgoals
  - improperly designed (e.g. designer lends bias to suboptimal options)
  - environment changes to make subgoal improper

What are possible ways to learn where to place subgoals in order to learn useful options?

Two Room Gridworld

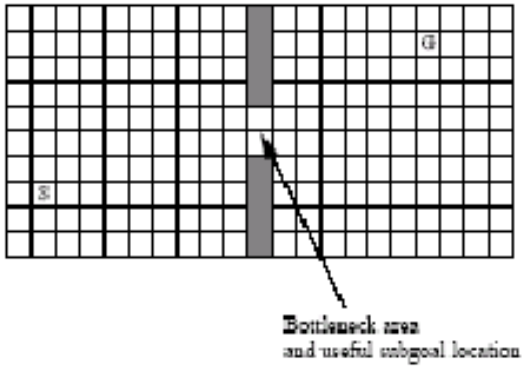


Source: McGovern01

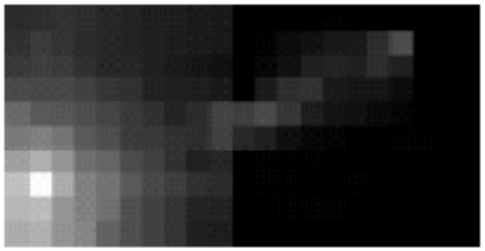
Simple approaches

- Approach I:
  - Keep track of how many times a state is visited over several episodes
  - Those visited the most are where we should place subgoals
- Approach II:
  - Keep track of whether or not a state is visited several episodes
  - Those states visited most often are where we should place subgoals

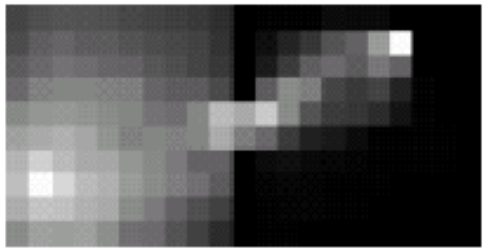
A: The two-room gridworld environment



B: Every-visit histogram



C: First-visit histogram



Source: McGovern01

Approach in McGovern01

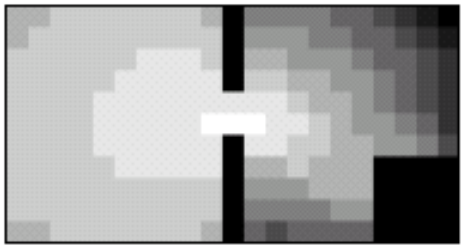
- For each new trajectory:
  - Classify trajectory as “positive bag” or “negative bag”
  - “Positive bags” are successful trajectories
  - “Negative bags” are unsuccessful trajectories
  - Try to find states that appear often in positive bags, but not in negative bags (i.e. states with high “Diverse Density”)
  - Pick state with highest average Diverse Density as potential location for subgoals
- Model for new options learned like in assigned reading, except with “experience replay”

Assumptions in McGovern01

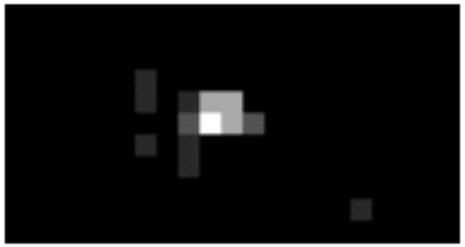
- There exists some sort of bottleneck that typically needs to be done to accomplish goal
- Negative bags cannot contain any positive instances
- Agent needs to initially reach goal state using only primitive actions
- States near the start and goal state can be filtered out when looking for highest diverse density

Results

A: Average Diverse Density

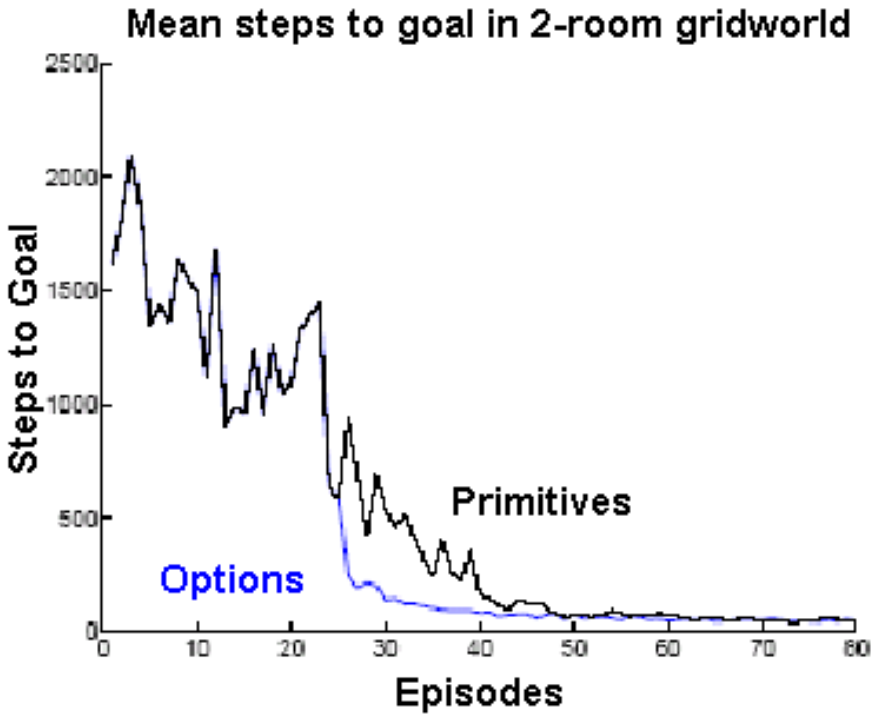


B: Subgoals Discovered



Source: McGovern01

Results



Source: McGovern01

Drawbacks of McGovern01

- Initial experimentation does not use negative bags
- In Kretchmar03, experiments done on 2 room gridworld with negative bags; lots of parameters to tune
- If a state appears in just one negative trajectory, it cannot be a subgoal
- Still uses some a priori knowledge to eliminate states around start and goal (which have higher Diverse Density than states in doorway)
- What happens when the bottleneck assumption is violated?
  - e.g. multiple valid paths to goal

References

- McGovern (2001): Automatic Discovery of Subgoals in RL using Diverse Density
  - http://www-anw.cs.umass.edu/~amy/pubs/mcgovern_barto_isairs2001.pdf

- Kretchmar (2003): Improved Automatic Discovery of Subgoals for Options in Hierarchical Reinforcement Learning
  - http://journal.info.unlp.edu.ar/journal/journal9/papers/JCST-Oct03-2.pdf