APML
שיעור 13

Reinforcement Learning

המטרה: ללמוד policy : מיפוי מרחב המצבים S למרחב הפעולות A.

כדי לעשות את זה, נגדיר מה זה policy טוב ע"י פונקציית reward

עבור $\bar{s}=((s_1,a_1),\dots,(s_T,a_T))$: $R: \bigcup_T (S\times A)^T \longrightarrow \mathbb{R}$

ואז נרצה לפתור את הבעיה הבאה: $\max_{\pi} \mathbb{E}_{\bar{s}}[R(\bar{s})]$

policy → π ; $\mathbb{E}_{\bar{s}}$ → expectation over episodes $\bar{s}$

אפשר לייחס בעיות SL כמקרה פרטי של RL. אבל עדיין יש כמה הבדלים:

1. ב-SL, פעולות לא משפיעות על הסביבה

⇐ אפשר לאסוף מלא דוגמאות מראש ואז ללמוד מהן

2. ב-SL, ההשפעה של פעולות היא מקומית בלבד.

3. ב-SL, ניתנת לנו התשובה הנכונה (ב-RL אותנו רק מקבלים reward מעיד על הטיבות שלנו.

פעולים בו יש לנו מומחה (expert):

אפשר להתייחס אליה כ- supervised ולאמן מדגמאות של המומחה...

בעיות: 1. אי אפשר להיות יותר טובים מהמומחה ממנו למדים

2. distribution drift:

follow vs predict

אימון על התפלגות מסוימת אבל בריצה

בריצה בפועל, אנחנו יצאנו מחוץ לאוסף המצבים

שלנו בכלל ונגיע למקומות שאותנו לא מכירים בכלל!

פעולים בו יודעים את המודל ורוצים להבין איך לפעול בו:

MDP - מניחים שהמידע כולל בעבר בהנתן ההווה.

* ואז מגדירים פונקציות V, Q עליהם מנסים אופטימלי

Q: expected reward starting from current state with current action

V: expected reward starting from current state on

ואז משתמשים באלגוריתם איטרטיבי Value Iteration

מה הבעיה? צריך לעבור על כל המצבים האפשריים באלגוריתם הזה,

ולא אם מרחב המצבים גדול נהיה בבעיה.

⇐ Approximate Value Iteration

מערכת על המודל - המשך

* Tree Search: אלא לבל עבור אבל לא קלן.

נסרוש את כל מרחב האפשרויות עד לאופן מסוים ונקבה לחפש
את המסלול הכי טוב. ואם העולם לא דטרמיניסטי, כמעבר קיימת
בשתי של Monte-Carlo-Tree-Search עדיין אפשר לעדכן חיפוש כדה.

* AlphaZero - אימון של שחקן בו רק מישחק מול עצמו שיגיע את ההצלחה
של DeepMind עם Alpha-Go.