شماره دانشجویی: ۹۹۱۰۲۲۰۷

نام: ایمان محمدی

سوال ۱)

معادلات بلمن(Boolean equations) ، یک نوع از معادلات ریاضی هستند که برای بیان منطق عبارات با استفاده از متغیرهای منطقی استفاده می‌شوند. در این نوع معادلات، متغیرهای منطقی می‌توانند دو مقدار "صحیح" یا "غلط" داشته باشند و عملیات منطقی مثل "و"، "یا" و "نه" برای ترکیب آن‌ها استفاده می‌شود. به عنوان مثال، یک معادله بلمن برای یک دروازه منطقی "و" به صورت زیر است:

A AND B = C

در اینجا A و B دو متغیر منطقی هستند که می‌توانند مقدار "صحیح" یا "غلط" داشته باشند، و C نیز خروجی دروازه منطقی است که به MDP یا فرآیند تصمیم گیری مارکوف، یکی از مدل هایی است که برای مدل کردن فرآیندهای تصمیم گیری با عدم قطعیت استفاده می شود. یکی از روش های حلMDP، استفاده از الگوریتم‌هایی مانند Q-learning و policy iteration است.

در این الگوریتم ها، برای به دست آوردن بهترین عمل برای یک وضعیت خاص، از معادلات بلمن استفاده می شود. به این شکل که معادلات بلمن برای تعیین مقدار Q-function به صورت بازگشتی استفاده می‌شوند.

Q-function تابعی از وضعیت و عمل است و نشان دهنده ارزشی است که می توان بدست آورد با انجام یک عمل در یک وضعیت خاص.

به طور خاص، معادلات بلمن برای به دست آوردن Q-function به صورت زیر است:

$$Q(s,a) = R(s,a) + \gamma * \Sigma P(s'|s,a) * \max(Q(s',a'))$$

در اینجا Q(s,a) نشان دهنده مقدار Q-function برای وضعیت s و عمل a است، R(s,a)نشان دهنده پاداشی است که در اثر انجام عمل a در وضعیت s بدست می‌آید، γیک پارامتر ارزش‌گذاری آینده است، P(s'|s,a) احتمال رسیدن به وضعیت 's به ازای انجام عمل a در وضعیت s و max(Q(s',a')) مقدار Q-function بیشینه برای وضعیت 's است که از آن عمل a انجام داده می شود.

با استفاده از این معادلات بلمن و الگوریتم هایی مانند Q-learning وpolicy iteration ، می‌توان به صورت بهینه عمل کرد و بهترین تصمیمات را در فرآیند تصمیم گیری با عدم قطعیت اتخاذ کرد.

ما در مسائلMDP ، به دنبال یافتن یک راهبرد بهینه برای عامل در محیط عمل هستیم. با استفاده از اطلاعات موجود در MDP شامل حالت‌ها، اقدامات، پاداش‌ها و احتمالات انتقال، می‌توانیم بهینه‌ترین راهبرد را پیدا کنیم که عامل باید در هر حالت بگیرد. برای پیدا کردن این راهبرد، می‌توان از الگوریتم‌هایی مانند Q-Learning و SARSA استفاده کرد که با آپدیت مقدارQ-function ، به صورت تقریبی بهینه‌ترین راهبرد را یاد می‌گیرند. هدف نهایی در مسائل MDP این است که عامل بتواند بیشترین پاداش ممکن را در طول زمان دریافت کند.

معادله بهینگی بلمن (Bellman Optimality Equation) یکی از معادلات اساسی در حوزه یادگیری تقویتی است که برای یافتن راهبرد بهینه در مسائل MDP (Markov Decision Process) استفاده می‌شود. این معادله به شکل یک تعریف بازگشتی برای تابع ارزش بهینه (Optimal Value Function) sنوشته می‌شود:

$$V^{*}(s) = \max_a \sum_{s'} P(s'|s,a) [R(s,a,s') + \gamma V^{*}(s')]$$

در این معادله، $V^{*}(s)$تابع ارزش بهینه در حالت s است که به عنوان بهترین تابع ارزش ممکن برای هر حالت تعریف می‌شود. همچنین، aحرکتی است که عامل در حالت s انجام می‌دهد، $P(s'|s,a)$ احتمال انتقال از حالت s به حالت 's با انجام حرکت a و $R(s,a,s')$ پاداش دریافتی از حرکت a در حالت s با انتقال به حالت 's است. همچنین، γ(عامل کاهنده تخفیف) پارامتری است که به اندازه‌گیری اهمیت پاداش‌های آینده نسبت به پاداش‌های فعلی کمک می‌کند.

معادله بهینگی بلمن به عنوان یک اصل اساسی در یادگیری تقویتی، در الگوریتم‌های مختلفی مانندQ-Learning و SARSA برای یادگیری راهبردهای بهینه در مسائل MDP به کار می‌رود.

سوال ۲)

الگوریتم SARSA یک الگوریتم یادگیری تقویتی مبتنی بر مدل است که برای یادگیری راهبرد بهینه در مسائل MDP (Markov Decision Process) به کار می‌رود. نام این الگوریتم از حروف اول کلمات State، Action، Reward، State و Action گرفته شده است.

در الگوریتم SARSA ، عامل در هر گام از بازی، یک اقدام انجام می‌دهد و پاداش دریافتی را دریافت می‌کند. با استفاده از اطلاعات موجود در MDP شامل حالت‌ها، اقدامات، پاداش‌ها و احتمالات انتقال، مقدار Q-function برای حرکت انجام شده در حالت فعلی به روزرسانی می‌شود. برای این کار از یک جفت حالت-عمل به عنوان ورودی استفاده می‌شود که با S و A نشان داده می‌شوند.

الگوریتم SARSA به شکل زیر است:

۱. مقدارα، γ و ε را تعیین کنید.

۲. مقدار اولیه برای Q-function را مقداردهی کنید.

۳. شروع یک بازی و تعیین حالت فعلی s

۴. انتخاب عمل فعلی a با استفاده از روش ε-greedy برای حالت s

۵. انجام عمل a و دریافت پاداش r و تعیین حالت جدید s'

۶. انتخاب عمل بعدی a' با استفاده از روش ε-greedy برای حالت s'

۷. به روزرسانی مقدار Q-function برای جفت حالت-عمل (s, a) با استفاده از فرمول زیر : $Q(s, a) \leftarrow Q(s, a) + \alpha(r + \gamma Q(s', a') - Q(s, a))$

۸. تعیین حالت فعلی $s \leftarrow s'$ و عمل فعلی $a \leftarrow a'$.

۹. تکرار مراحل ۴ تا ۸ تا زمانی که بازی به پایان برسد.

در این الگوریتم، پارامتر α نرخ یادگیری، پارامتر γ عامل کاهنده تخفیف و پارامتر ε شیب‌دهی در روش ε-greedy است. هدف این الگوریتم پیدا کردن یک سیاست بهینه با استفاده از یک جدول Q-function است که برای هر جفت حالت-عمل، مقدار Q-function مربوط به آن بهینه شده باشد.

در هنگام اجرای الگوریتم SARSA، عامل در هر گام، بر اساس جدول Q-function عملی را انتخاب می‌کند که بیشترین مقدار Q-function را دارد. اما در مواردی که با احتمال ε، عامل یک عمل تصادفی انتخاب می‌کند و به این صورت می‌تواند کاوش بیشتری در فضای حالت و عمل داشته باشد. این روش به عنوان روش ε-greedy شناخته می‌شود.

مزیت این الگوریتم نسبت به روش Q-Learning این است که در روش SARSA، عامل در هر گام از بازی عملی را انتخاب می‌کند و با استفاده از آن در جدول Q-function، مقدار Q-function را به‌روزرسانی می‌کند. این موضوع باعث می‌شود که الگوریتم SARSA بهتر بتواند با مسائلی که انتخاب عمل‌های یکپارچه‌تری نسبت به روش Q-Learning دارند، سازگار باشد.

الگوریتم Q-Learning یک الگوریتم یادگیری تقویتی بدون مدل است که برای یادگیری یک سیاست بهینه در یک مسئله MDP (فرایند تصمیم‌گیری مارکوف) استفاده می‌شود. در این الگوریتم، یک جدول Q-function برای هر جفت حالت-عمل تعریف می‌شود که مقدار Q-function مربوط به آن، انتظار بیشینه مقدار بازگشتی را که در صورت انجام آن عمل در حالت مربوطه انتظار داریم، نشان می‌دهد.

در ابتدا، جدول Q-function به صورت تصادفی مقداردهی می‌شود. سپس با شروع از یک حالت اولیه، عامل در هر گام با استفاده از روش ε-greedy یک عمل را انتخاب می‌کند و سپس با انجام آن عمل، به حالت بعدی می‌رود. در این مرحله، با استفاده از جدولQ-function ، مقدار Q-function مربوط به جفت حالت-عمل فعلی و حالت-عمل بعدی را به روزرسانی می‌کنیم. این به روزرسانی، بر اساس یک رابطه بلمن انجام می‌شود که به شکل زیر است:

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

در این رابطه، s و a به ترتیب حالت و عمل فعلی را نشان می‌دهند، s' و a' حالت و عمل بعدی را نشان می‌دهند، r پاداش دریافتی در حین انجام عمل فعلی را نشان می‌دهد، α ضریب یادگیری را نشان می‌دهد و γ ضریب تخفیف را نشان می‌دهد.

با این کار، مقدار Q-function برای جفت حالت-عمل فعلی و حالت-عمل بعدی بهبود می‌یابد و با تکرار این عمل برای گام‌های بعدی، جدول Q-function بهبود پیدا می‌کند. در نهایت، با استفاده از هر دو الگوریتم SARSA و Q-Learning الگوریتم‌های یادگیری تقویتی بدون مدل هستند که برای یادگیری سیاست بهینه در یک مسئله MDP (فرایند تصمیم‌گیری مارکوف) استفاده می‌شوند، اما تفاوت‌های مهمی نیز دارند.

تفاوت اصلی بین الگوریتم SARSA و Q-Learning در روش به روز رسانی جدول Q-function است. درQ-Learning ، در هر گام از آموزش، مقدار Q-function برای حالت و عمل فعلی با توجه به بهترین عمل ممکن در حالت بعدی به روز می‌شود. به عبارت دیگر، در هر مرحله از آموزش، Q-Learning از تجربه بهترین عمل برای هر حالت بهره می‌برد.

اما در الگوریتم SARSA، مقدار Q-function برای جفت حالت-عمل فعلی و حالت-عمل بعدی با توجه به عملی که عامل در حالت بعدی خواهد انجام داد به‌روز می‌شود. یعنی در هر گام از آموزش، SARSA از تجربه عمل فعلی خود بهره می‌برد.

به طور خلاصه، Q-Learning از تجربه بهترین عمل در حالت بعدی بهره می‌برد، در حالی که SARSA از تجربه عمل فعلی خود استفاده می‌کند. به همین دلیل، SARSA در برخی مسائل بهتر عمل می‌کند که برای رسیدن به سیاست بهینه، باید عمل‌های بیشتری را بررسی کرد. از طرف دیگر، Q-Learning برای برخی مسائل، می‌تواند به سرعت به سیاست بهینه برسد.

سوال ۳)

در یادگیری تقویتی، تریدآف بین استخراج و اکتشاف یکی از مهم‌ترین چالش‌هاست. به طور کلی، استخراج (Exploitation) به معنای انتخاب عملی است که بهترین پاداش را در حال حاضر به ما می‌دهد، و اکتشاف (Exploration) به معنای انتخاب عمل‌هایی است که در گذشته به ما پاداش کمتری داده‌اند، اما ممکن است در آینده باعث بهبود عملکرد ما شوند.

اگر تنها به استخراج عمل‌ها بپردازیم، ممکن است در یک حفره کوچک گیر کنیم و از بهترین عمل‌های فعلی هیچ چیز بیشتر یاد نگیریم. از سوی دیگر، اگر تنها به اکتشاف عمل‌ها بپردازیم، ممکن است در پرتاب بین گزینه‌های مختلف گرفتار شویم و پاداش کمتری دریافت کنیم.

برای حل این چالش، الگوریتم‌های یادگیری تقویتی از روش‌های متنوعی برای تریدآف استخراج و اکتشاف استفاده می‌کنند. برخی از این روش‌ها عبارتند از:

1- Epsilon-Greedy:

این روش، یکی از ساده‌ترین روش‌های تریدآف استخراج و اکتشاف است. در این روش، با احتمال ε، یک عمل تصادفی انتخاب می‌شود و با احتمال ۱ε-، عمل بهینه انتخاب می‌شود.

2- Upper Confidence Bound (UCB):

در این روش، به هر عمل یک بالاترین مرز (Upper Confidence Bound) نسبت داده می‌شود و در هر گام از آموزش، عملی با بیشترین UCB انتخاب می‌شود.

در یادگیری تقویتی، هدف پیدا کردن بهترین راهبرد برای رسیدن به حالت هدف است. برای دستیابی به این هدف، عامل باید هم زمان به استخراج از تجربیات قبلی خود و هم به اکتشاف برای یافتن تجربیات جدید توجه کند. روش های مختلفی برای حل مشکل تریدآف وجود دارد، مانند استفاده از روش های اکتشافی مثل ε-greedy و Upper Confidence Bound (UCB) که بالاتر گفته شد؛ استفاده از تابع ارزش مطمئن‌تر (Optimistic Initial Values) و استفاده از تکنیک های مانیتورینگ کارایی عامل در طول زمان(Monitoring Performance Over Time)