یک روش جدید مبتنی بر معیارهای آماری توزیع برای تنظیم خودکار نرخ یادگیری اتوماتای یادگیر در محیط‌های پویا

محمدرضا ملاخلیلی میبدی و محمدرضا میبدی

چکیده: یکی از مسایل مطرح در ساخت سیستم‌های یادگیر نظیر شبکه‌های عصبی و یا اتوماتای یادگیر، تعیین نرخ یادگیری است. در اکثر موارد از یک الگوریتم کاهش‌یابنده در طول زمان برای تنظیم نرخ یادگیری استفاده می‌شود. در این مقاله یک روش جدید برای تغییر نرخ یادگیری و انطباق سیستم یادگیرنده با وضعیت محیط، برای استفاده در اتوماتای یادگیر پیشنهاد شده است. این روش جدید از برخی معیارهای آماری مربوط به توزیع فعلی به دست آمده برای بردار احتمالات متناظر با اقدام‌های اتوماتا به منظور تعیین افزایش یا کاهش نرخ یادگیری استفاده می‌کند. مزیت این روش در آن است که بر خلاف روش‌های موجود فعلی، در طول فرایند یادگیری هم افزایش و هم کاهش مقدار نرخ یادگیری را- بسته به نتایج مقایسه معیارهای آماری- انجام می‌دهد و به صورت خودکار نرخ یادگیری را تنظیم می‌کند.

ضمن تشریح مبانی ریاضی این الگوریتم جدید، عملکرد این الگوریتم را در محیط‌های تصادفی نمونه بررسی کرده و با مقایسه نتایج به دست آمده نشان داده‌ایم روش پیشنهادی جدید به دلیل این که در طول زمان یادگیری، هم‌زمان و بر اساس معیارهای تعیین‌شده، افزایش و کاهش نرخ یادگیری را انجام می‌دهد، از انعطاف‌پذیری بیشتری نسبت به روش‌های قبلی برای انطباق با محیط‌های تصادفی پویا برخوردار است و مقادیر یاد گرفته شده به مقادیر حقیقی نزدیک‌تر هستند.

کلید واژه: **اتوماتای یادگیر، نرخ یادگیری پویا، تنظیم نرخ یادگیری، نابرابری چبیشف.**

۱- مقدمه

بحث تنظیم خودکار نرخ یادگیری یکی از مسایلی است که در مباحث مرتبط با الگوریتم‌ها و سیستم‌های یادگیری و خصوصاً در مباحث مربوط به شبکه‌های عصبی به کرات مورد بررسی قرار گرفته است [۱] تا [۳]. اکثر الگوریتم‌های موجود، از یک نرخ یادگیری با مقدار بالا شروع کرده و در حین فرایند آموزش به کمک یک تابع کاهش‌یابنده با زمان (مثلاً $\alpha(t)=\alpha(\cdot)e^{-t/T}$) آن را کاهش می‌دهند. به این ترتیب پس از مدتی نرخ یادگیری مقدار معمولاً کوچکی دارد [۲].

مشکل از اینجاست که اگر از این مرحله به بعد، نمونه‌هایی وارد شوند که مشخصه‌های آماری متفاوتی با مشخصه‌های آماری نمونه‌های قبلی داشته باشند، سیستم یادگیر قادر به یادگیری آنها نیست. بنابراین نرخ یادگیری کاهش‌یابنده با زمان با وجود مزیت‌هایی که در همگرایی سریع دارد، باعث می‌شود سیستم یادگیر از قدرت تطابق‌پذیری و یادگیری خوبی در برخورد با محیط‌های تصادفی گوناگون برخوردار نباشد [۳].

کارهایی که در زمینه انطباقی‌کردن نرخ یادگیری در اتوماتای یادگیر انجام شده است را می‌توان به دو گروه تقسیم کرد: وابسته به مسأله و مستقل از مسأله. در روش‌های پیشنهادی وابسته به مسأله، بر اساس نوع مسأله‌ای که اتوماتای یادگیر و یا اجتماعی از اتوماتاهای یادگیر در صدد حل آن هستند روش‌هایی برای تطبیقی‌کردن نرخ یادگیری پیشنهاد شده است [۴] تا [۷]. در تمام این نمونه‌ها بر حسب نوع مسأله، روشی برای به روز رسانی نرخ یادگیری پیشنهاد شده است که سرعت همگرایی و حل مسأله را افزایش داده است. رده وسیعی از این روش‌های انطباقی را در [۴] می‌توانید مشاهده کنید که در آن نویسنده با توجه به مسأله یافتن درخت پوشای کمینه در گراف‌های تصادفی، از پاره‌ای ویژگی‌های توزیع احتمال اقدام‌ها به شکل مستقیم برای تطبیق‌کردن نرخ یادگیری و رسیدن به همگرایی سریع‌تر استفاده کرده است. مشابه همین ایده نیز در [۸] و [۶] مورد استفاده قرار گرفته است. اما در گروه دوم، مستقل از مسأله یا ساختار اجتماع اتوماتاهای یادگیر، روش‌هایی برای این کار پیشنهاد می‌شود.

از آنجا که غالباً استفاده از اتوماتاهای یادگیر به شکل شبکه‌ای ساختارمند صورت می‌گیرد [۹]، بنابراین روش‌های پیشنهادی برای انطباقی‌کردن نرخ یادگیری غالباً وابسته به مسأله و ساختار بوده و از مسأله‌ای به مسأله دیگر و از ساختاری به ساختار دیگر متفاوت هستند. پژوهش مستقلی که به بررسی تنظیم خودکار نرخ یادگیری در اتوماتای یادگیر (فارغ از مسأله‌ای که اتوماتا در صدد حل آن است و یا ساختار شبکه‌ای از اتوماتاها که برای حل مسأله مورد استفاده قرار می‌گیرد) پرداخته باشد، مشاهده نشده است. عموم پژوهش‌های این بخش مربوط به شبکه‌های عصبی می‌باشند که برخی از آنها به لحاظ عمومیتی که در مبانی دارند- نظیر نرخ یادگیر کاهش‌یابنده با زمان- برای سایر سیستم‌های یادگیر و از جمله اتوماتای یادگیر نیز قابل استفاده هستند.

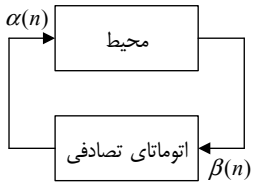
به نظر می‌رسد ایده‌های تطبیقی‌کردن نرخ یادگیری با هدف تسریع در همگرایی فرایند یادگیری صورت گرفته است و مسأله انطباقی‌کردن نرخ یادگیری در محیط‌های پویا چندان مد نظر قرار نگرفته است.

ادامه مقاله بدین صورت سازمان‌دهی شده است. در بخش دوم ضمن بررسی مختصر اتوماتای یادگیر، نامساوی چبیشف را به عنوان یک نامساوی کاربردی در توزیع‌های تصادفی که مستقل از توزیع متغیر تصادفی، رابطه‌ای را میان مقدار متغیر تصادفی، مقدار میانگین و واریانس آن وضع می‌کند، بررسی خواهیم کرد. در ادامه همین بخش ایده به کارگیری این نامساوی به عنوان یک معیار تشخیص پویایی محیط مورد بررسی قرار می‌گیرد. در بخش سوم الگوریتم پیشنهادی مبتنی بر نامساوی چبیشف ارائه خواهد شد. بخش چهارم به بررسی تجربی این الگوریتم در تعدادی محیط پویا اختصاص داده شده است و بخش پنجم به جمع‌بندی مطالب ارائه‌شده در مقاله اختصاص یافته است.

این مقاله در تاریخ ۳۰ خرداد ماه ۱۳۹۱ دریافت و در تاریخ ۲۵ بهمن ماه ۱۳۹۱ بازنگری شد.

محمدرضا ملاخلیلی میبدی، دانشکده مهندسی کامپیوتر، دانشگاه آزاد اسلامی واحد علوم و تحقیقات تهران، حصارک تهران، (email: m.meybodi@srbiau.ac.ir).

محمدرضا میبدی، دانشکده مهندسی کامپیوتر و فن‌آوری اطلاعات، دانشگاه صنعتی امیرکبیر، تهران، (email: mmeybodi@aut.ac.ir).



شکل ۱: اتوماتای یادگیر و نحوه تعامل آن با محیط.

۲- مبانی روش پیشنهادی

۲-۱ بررسی نحوه یادگیری در اتوماتای یادگیر

یک اتوماتای یادگیر به عنوان مدلی از یک سیستم یادگیر است که در محیط‌های تصادفی ناشناخته عمل می‌کند. اتوماتا در هر دور یک اقدام از میان مجموعه محدود اقدام‌های خود انتخاب کرده و با بررسی عکس‌العمل محیط نسبت به این اقدام، احتمال انتخاب اقدام‌های بعدی را بهبود می‌بخشد [۹].

یک اتوماتای تصادفی را می‌توان به صورت یک ماشین حالت متناهی در نظر گرفت. به بیان ریاضی نیز می‌توان آن را به صورت یک پنج‌تایی مانند زیر نشان داد

$$SA \equiv \{\alpha, \beta, F, G, \phi\} \qquad (۱)$$

که در آن r تعداد اقدام‌های اتوماتا، $\alpha \equiv \{\alpha_1, \alpha_2, \ldots, \alpha_r\}$ مجموعه اقدام‌های اتوماتا، $\beta \equiv \{\beta_1, \beta_2, \ldots, \beta_r\}$ مجموعه ورودی‌های اتوماتا، $F \equiv \phi \times \beta \to \phi$ تابع نگاشت وضعیت فعلی و ورودی به وضعیت بعدی، $G \equiv \phi \to \alpha$ تابع خروجی که وضعیت فعلی را به خروجی بعدی نگاشت می‌کند و $\phi(n) \equiv \{\phi_1, \phi_2, \ldots, \phi_k\}$ مجموعه وضعیت‌های داخلی اتوماتا در زمان n است.

مجموعه α شامل خروجی‌های (اقدام‌های) اتوماتا است که اتوماتا در هر گام یک اقدام از r اقدام این مجموعه را برای اِعمال بر محیط انتخاب می‌نماید. مجموعه ورودی‌ها (β) ورودی‌های اتوماتا را مشخص می‌کند و توابع F و G وضعیت فعلی ورودی را به خروجی بعدی (اقدام بعدی) اتوماتا نگاشت می‌کنند. اگر نگاشت‌های F و G قطعی باشند، اتوماتا یک اتوماتای قطعی نامیده می‌شود. در چنین حالتی با فرض یک وضعیت اولیه و ورودی مشخص، حالت بعدی و خروجی به صورت یکتا مشخص شده‌اند. در حالتی که نگاشت‌های F و G تصادفی باشند، اتوماتا به عنوان اتوماتای تصادفی معرفی می‌شود.

اتوماتای تصادفی را می‌توان به دو دسته اتوماتای تصادفی با ساختار ثابت و اتوماتای تصادفی با ساختار متغیر تقسیم‌بندی کرد. در اتوماتای تصادفی با ساختار ثابت احتمال اقدام‌های اتوماتا ثابت هستند در حالی که در اتوماتای تصادفی با ساختار متغیر احتمالات اقدام‌های اتوماتا در هر تکرار به روز می‌شوند (تغییر احتمال‌های اقدام‌ها بر اساس الگوریتم یادگیری انجام می‌شود). وضعیت داخلی اتوماتا ϕ توسط احتمالات اقدام‌های اتوماتا بازنمایی می‌شوند. برای سادگی هر وضعیت داخلی اتوماتا را مطابق با یک اقدام مشخص اتوماتا در نظر می‌گیرند، بنابراین می‌توان وضعیت داخلی اتوماتا ϕ را با بردار احتمال اقدام‌های اتوماتا P که به صورت زیر نشان داده می‌شود، جایگزین نمود

$$P(n) \equiv \{p_1(n), p_2(n), \ldots, p_r(n)\} \qquad (۲)$$

به گونه‌ای که

$$\sum_{i=1}^{r} p_i(n) = 1 \quad , \quad \forall n \quad , \quad p_i(n) = \text{Prob}[\alpha(n) = \alpha_i] \qquad (۳)$$

در آغاز فعالیت اتوماتا، احتمال اقدام‌های آن به صورت مساوی با هم برابر با $p_i = 1/r$ قرار داده می‌شوند (که r تعداد اقدام‌های اتوماتا می‌باشد).

محیط: محیط تصادفی را به طور ریاضی می‌توان به صورت سه‌تایی $E \equiv \{\alpha, \beta, c\}$ توصیف کرد که $\alpha \equiv \{\alpha_1, \alpha_2, \ldots, \alpha_r\}$ مجموعه ورودی‌های محیط، $\beta \equiv \{\beta_1, \beta_2, \ldots, \beta_r\}$ مجموعه خروجی‌های محیط و $c \equiv \{c_1, c_2, \ldots, c_r\}$ مجموعه احتمال‌های جریمه است.

ورودی محیط یکی از r اقدام انتخاب‌شده اتوماتا است و خروجی (پاسخ) محیط به هر اقدام i توسط β_i مشخص می‌شود [۱۰].

ارتباط اتوماتای تصادفی با محیط در شکل ۱ نشان داده شده است. از این مجموعه به همراه الگوریتم یادگیری تحت عنوان اتوماتای یادگیر تصادفی نام برده می‌شود. به این ترتیب اتوماتای یادگیر تصادفی را می‌توان با (۴) تعریف کرد

$$SLA \equiv \{\alpha, \beta, p, T, c\} \qquad (۴)$$

به طوری که $\alpha \equiv \{\alpha_1, \alpha_2, \ldots, \alpha_r\}$ مجموعه اقدام‌های اتوماتا/ مجموعه ورودی‌های محیط، $\beta \equiv \{\beta_1, \beta_2, \ldots, \beta_r\}$ مجموعه ورودی‌های اتوماتا/ مجموعه خروجی‌های محیط، $p \equiv \{p_1, p_2, \ldots, p_r\}$ بردار احتمال اقدام‌های اتوماتا، $T \equiv p(n+1) = T[\alpha(n), \beta(n), p(n)]$ الگوریتم یادگیری و $c \equiv \{c_1, c_2, \ldots, c_r\}$ مجموعه احتمالات جریمه که معرف محیط می‌باشند، است.

الگوریتم یادگیری خطی: الگوریتم یادگیری یک رابطه بازگشتی است که برای انجام تغییرات و به روز رسانی در بردار احتمال اقدام‌های اتوماتا در یک اتوماتای یادگیر تصادفی با ساختار متغیر مورد استفاده قرار می‌گیرد. فرض کنید یک اتوماتای یادگیر تصادفی ساختار متغیر در زمان k از میان مجموعه اقدام‌های α عمل $\alpha_i(k)$ را انتخاب کرده باشد. همچنین فرض کنید بردار احتمال انتخاب اقدام‌های اتوماتا را با $p(k)$ نمایش داده‌ایم. اگر a و b پارامترهایی باشند که به ترتیب میزان افزایش یا کاهش احتمالات اقدام‌ها را مشخص می‌کنند و r تعداد اقدام‌های قابل انجام توسط اتوماتای یادگیر باشد، بردار $p(k)$ توسط الگوریتم یادگیری خطی ارائه‌شده در روابط زیر به روز رسانی می‌شود (مقدار a را پارامتر پاداش و b را پارامتر جریمه می‌نامند)

$$p_j(k+1) = \begin{cases} (1-a)p_j(k) + a & , \ j = i \\ (1-a)p_j(k) & , \ \forall j \neq i \end{cases} \qquad (۵)$$

$$p_j(k+1) = \begin{cases} (1-b)p_j(k) & , \ j = i \\ (1-b)p_j(k) + \dfrac{b}{r-1} & , \ \forall j \neq i \end{cases} \qquad (۶)$$

رابطه (۵) زمانی مورد استفاده قرار می‌گیرد که عمل $\alpha_i(k)$ منجر به دریافت پاداش از محیط شده باشد و (۶) زمانی مورد استفاده قرار می‌گیرد که این عمل به دریافت جریمه از محیط منجر شده باشد. اگر $a = b$ باشد، روابط یادگیری خطی (معادله‌های (۵) و (۶)) را الگوریتم L_{R-P} می‌نامند. اگر $a >> b$ باشد آن را $L_{R-\varepsilon P}$ و اگر $b = 0$ باشد آن را L_{R-I} می‌نامند [۱۱].

عامل مؤثر در کارایی اتوماتای ساختار متغیر، الگوریتم‌های یادگیری هستند که برای به روز رسانی احتمال اقدام‌ها استفاده می‌شود.

با این مقدمات فرض کنید یک اتوماتای یادگیر با r اقدام در یک محیط تصادفی فعالیت می‌کند. محیط تصادفی اقدام انجام‌شده توسط اتوماتا را ارزیابی می‌کند و اتوماتا بر اساس این ارزیابی، بردار احتمالات اقدام‌های خود را به روز رسانی می‌کند. فرض کنید P_i^t مقدار احتمال انتخاب اقدام iام اتوماتا را در زمان t نشان دهد. ضمناً

$$\sum_{i=1}^{r} P_i^t = 1 \ , \ \forall t \in \{0,1,\ldots\} \qquad (۷)$$

بردار احتمال انتخاب اقدام‌های اتوماتا در زمان t را با $\overline{P(t)}$ نشان می‌دهیم که $\overline{P(t)} = [P_1^t, P_2^t, \ldots, P_r^t]$. علاوه بر این محیط تصادفی بر اساس بردار احتمالات $\overline{C} = [C_1, C_2, \ldots, C_r]$ اقدام‌های محیط را پاداش می‌دهد که در آن $\sum_{i=1}^{r} C_i = 1$ است.

در این اتوماتای یادگیر هر اقدام اتوماتا دارای یک نرخ یادگیری است. این بردار را با $\overline{\alpha} \equiv \{\alpha_1, \alpha_2, \ldots, \alpha_r\}$ نشان می‌دهیم. در این مقاله فرض می‌کنیم اتوماتای یادگیر برای به روز رسانی بردار احتمال انتخاب‌های خود از الگوریتم L_{R-I} استفاده می‌کند.

گزاره ۱: رابطه به روز رسانی بردار احتمال انتخاب اقدام اتوماتای یادگیر برای زمانی که اقدام kام توسط اتوماتا صورت گرفته و اتوماتا از الگوریتم یادگیر خطی L_{R-I} استفاده می‌کند را می‌توان به صورت (۸) نوشت

$$P_k^{t+1} = P_k^t \times (1-\alpha_k) + \alpha_k R^{t+1} \qquad (۸)$$

در (۸) R پاسخ محیط به اقدام انجام‌شده توسط اتوماتا است و داریم

$$R^{t+1} = \begin{cases} 1 & \text{with probability} \quad C_k \\ 0 & \text{with probability} \quad 1-C_k \end{cases} \qquad (۹)$$

رابطه یادگیری اتوماتا در این حالت را به شکل برداری (۱۰) می‌توان نشان داد

$$\vec{P}^{t+1} = \vec{P}^t \times (1-\alpha) + \alpha \vec{R}^{t+1} \qquad (۱۰)$$

رابطه (۱۰) در حقیقت نشان می‌دهد که بردار احتمال انتخاب اقدام‌های اتوماتای یادگیر در طول فرایند یادگیری، به بردار میانگین پاسخ‌های محیط همگرا می‌شود.

۲-۲ محیط‌های پویا

موفقیت اتوماتای یادگیر در محیط‌های پویا بستگی به تغییرات محیط و اطلاعاتی که توسط اتوماتای یادگیر از محیط قابل جمع‌آوری است، دارد. در [۹] نویسنده یک تقسیم‌بندی مفصل از محیط‌های پویا ارائه داده است اما از یک دیدگاه کلی می‌توان محیط‌های پویا را به دو گروه کلی تقسیم کرد: محیط‌های پویای MSE و محیط‌های پویا با تابع احتمال جریمه متغیر با زمان.

در MSEها فرض بر این است که محیط پویای واقعی خود از چند محیط ایستا[1] تشکیل شده است و پویایی محیط ناشی از یک توالی از فرایندهای جابه‌جایی بین این محیط‌های ایستا است [۱۲]. به بیان ریاضی، محیط پویای ξ توسط یک مجموعه $\{E_1, E_2, \ldots, E_H\}$ از محیط‌های ایستا و یک ماتریس جابه‌جایی T تعریف می‌شود. در این ماتریس T، عنصر $T_{i,j}$ نشان‌دهنده احتمال آن است که اگر اتوماتای یادگیر در حال حاضر با محیط تصادفی E_i در تعامل است، در گام بعدی با محیط تصادفی E_j در تعامل باشد. در حقیقت در MSEها محیط تصادفی پویا از مجموعه‌ای از محیط‌های تصادفی ایستا تشکیل شده است که مجموعه حالات یک زنجیره مارکوف را تشکیل می‌دهند [۹]، [۱۲] و [۱۳].

مدل دیگری که در متون مربوطه برای محیط‌های پویا ارائه شده است، مدلی است که احتمالات جریمه اقدام‌ها را غیر ثابت و متغیر با زمان فرض کرده است [۱۳] و [۱۴]. یعنی احتمال جریمه اقدام در عین این که به اقدام انجام‌شده بستگی دارد، به زمان انجام آن نیز بستگی دارد.

مقالاتی که به ارائه الگوریتم‌های یادگیر مبتنی بر اتوماتای یادگیر برای محیط‌های MSE پرداخته‌اند به دلیل تنوع این محیط‌ها بسیار زیاد هستند. ایده غالب آنها تشکیل سلسله مراتبی از اتوماتاهای یادگیر است که برخی سطوح وظیفه تعیین محیط ایستا را بر عهده دارند و برخی سطوح دیگر فرایند یادگیری در آن محیط ایستا را بر عهده دارند [۱۳].

همان طور که ذکر شد، هیچ پژوهش مستقلی که فرایند یادگیری در اتوماتای یادگیر را از طریق تطبیقی‌کردن نرخ یادگیری دنبال کرده باشد، در متون و مقالات تخصصی این حوزه یافت نشد و آنچه که از طریق تطبیقی‌کردن نرخ یادگیری صورت می‌گیرد غالباً وابسته به مسأله و ساختاری از اتوماتاها که برای حل مسأله مورد استفاده قرار گرفته، می‌باشد.

۲-۳ نابرابری چبیشف

برای تنظیم پویای نرخ یادگیری به منظور انطباق در محیط‌های پویا، نیاز به معیاری داریم تا بر اساس آن معیار میزان توانمندی نرخ یادگیری فعلی سیستم یادگیر را در رصدکردن تغییرات محیط بسنجیم. یکی از ابزارهای آماری مناسب در این مورد نابرابری موسوم به نابرابری چبیشف است. در نظریه احتمالات نابرابری چبیشف تضمین می‌کند که در هر نمونه تصادفی یا در هر توزیع احتمال، "تقریباً تمامی" مقادیر در نزدیکی میانگین خواهند بود. به طور دقیق‌تر این قضیه بیان می‌کند که حداکثر مقادیری که در هر توزیع می‌توانند بیش از k برابر انحراف معیار با میانگین فاصله داشته باشند $1/k^2$ است [۱۵].

قضیه (نامساوی مارکوف): اگر X یک متغیر تصادفی و a یک عدد حقیقی مثبت باشد، در این صورت

$$\Pr(|X| > a) \le \frac{E(X^2)}{a^2} \qquad (۱۱)$$

برای اثبات این قضیه می‌توانید به [۱۶] مراجعه کنید.

با استفاده از نامساوی مارکوف می‌توان به نامساوی چبیشف رسید. اگر m نشان‌دهنده میانگین متغیر تصادفی X باشد، با جایگذاری $X-m$ در رابطه بالا به نامساوی موسوم به چبیشف خواهیم رسید

$$\Pr(|X-m| \ge a) \le \frac{Var(X)}{a^2} \qquad (۱۲)$$

اگر $Var(X)$ را با δ نشان دهیم تفسیر دیگری از نامساوی چبیشف به دست می‌آید

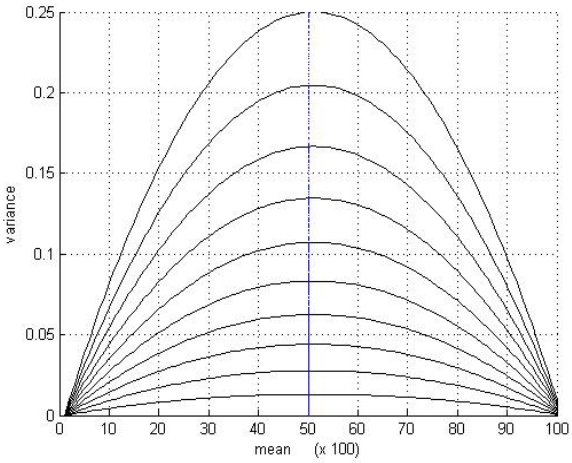
$$\Pr(|X-m| \ge a\delta) \le \frac{1}{a^2} \qquad (۱۳)$$

رابطه (۱۳) در حقیقت بیان می‌کند احتمال این که یک متغیر تصادفی در خارج از بازه‌ای حول میانگین به شعاع a برابر واریانس باشد از $1/a^2$ کمتر است. این رابطه برای مقادیر $a > 1$ واجد اطلاعات مفید است.

نابرابری چندبعدی چبیشف تعمیمی از نابرابری چبیشف است که به کمک آن می‌توان مرزی را برای این که یک بردار تصادفی از بردار مقادیر میانگینش بیش از یک مقدار معین فاصله داشته باشد را تعیین کرد. این نابرابری بدین شکل فرمول‌بندی می‌شود.

نابرابری چبیشف (تعمیم‌یافته): فرض کنید X یک بردار تصادفی با میانگین $\mu = E[X]$ و ماتریس کوواریانس $V = E[(X-\mu)(X-\mu)^T]$ باشد. اگر V یک ماتریس تعریف‌شده مثبت باشد در این صورت برای هر عدد حقیقی $t > 0$ داریم

۱. در این مقاله محیط ایستا را معادل با Stationary Environment و محیط پویا را برای Non-Stationary Environment در نظر گرفته‌ایم.



شکل ۲: رابطه میان واریانس با میانگین و نرخ یادگیری.

$$\Pr(((X-\mu)V^{-1}(X-\mu)^T) > qN) \le \frac{1}{q^2} \qquad (۱۴)$$

که در آن $N = trace(V^{-1}V)$ است.

۲-٤ استفاده از نابرابری چبیشف به عنوان یک معیار در تطبیقی‌کردن نرخ یادگیری

در این قسمت با توجه به نابرابری چبیشف و گزاره ۱ که نحوه تغییرات بردار احتمال انتخاب اقدام‌های اتوماتا را نشان می‌دهد، به ارائه معیاری برای کاهش یا افزایش نرخ یادگیری اتوماتای یادگیر خواهیم پرداخت.

گزاره ۲: فرض کنید X متغیری است که بر اساس (۱۵) به روز رسانی می‌شود

$$X^{t+1} = X^t \times (1-\alpha) + \alpha R^{t+1} \qquad (۱۵)$$

که در آن R^{t+1} طبق (۱۶) محاسبه می‌شود

$$R^{t+1} = \begin{cases} 1 & \text{with probability } C^* \\ 0 & \text{with probability } 1-C^* \end{cases} \qquad (۱۶)$$

ثابت می‌شود [۱۶]:

الف) $\lim_{t\to\infty} E[X^t] = \mu = C^*$

ب) واریانس X^t محدود و دارای مقدار حدی زیر است

$$Var(X^t) = \delta^2 = \frac{\alpha}{2-\alpha} C^*(1-C^*)$$

ج) مطابق با نامساوی چبیشف داریم

$$\forall q > 0 \quad P(|X^t - C^*| \ge q\delta) \le \frac{1}{q^2}$$

اثبات:

الف) R^{t+1} یک فرایند تصادفی برنولی با پارامتر C^* است. علاوه بر این داریم

$$\begin{aligned} E[X^{t+1}] &= (1-\alpha)E[X^t] + \alpha E[R^{t+1}] \\ &= (1-\alpha)E[X^t] + \alpha C^* \end{aligned} \qquad (۱۷)$$

از رابطه بازگشتی (۱۷) داریم

$$E[X^{t+1}] = (1-\alpha)^t E[X^0] + (1-(1-\alpha)^t).C^* \qquad (۱۸)$$

با توجه به این که $0 < \alpha < 1$، (۱۸) نشان می‌دهد

$$\lim_{t\to\infty} E[X^{t+1}] = C^* \qquad (۱۹)$$

ب) به طریق مشابه می‌توان نشان داد

$$Var[X^{t+1}] = \frac{\alpha}{2-\alpha} C^*(1-C^*)(1-(1-\alpha)^{2t+2}) \qquad (۲۰)$$

با توجه به این که $0 < \alpha < 1$، (۲۰) نشان می‌دهد

$$\lim_{t\to\infty} Var[X^{t+1}] = \frac{\alpha}{2-\alpha} C^*(1-C^*) \qquad (۲۱)$$

ج) با فرض $m = E[X]$ برای متغیر تصادفی X نامساوی مارکوف، $P(|X| \ge a) \le E(X^2)/a^2$ به (۲۲) تبدیل خواهد شد

$$P(|X-m| \ge a) \le \frac{Var(X)}{a^2} \qquad (۲۲)$$

با جایگذاری $a = q.\sqrt{Var(X^t)} = q.\delta$ و $m = E[X^t] = C^*$ در مورد متغیر تصادفی $X = X^t$ در (۲۲)، نتیجه ج حاصل خواهد شد.

گزاره ۲ چند نکته را نشان می‌دهد:

اولاً مقدار α کنترل‌کننده نرخ همگرایی است. روابط بازگشتی مربوط به میانگین و واریانس نشان می‌دهند که اگر α برابر با ۱ باشد سرعت همگرایی بیشینه است. اگرچه مقدار α برابر با ۱ سرعت همگرایی بالایی دارد اما منجر به واریانس بزرگ‌تری می‌شود (شکل ۲). مقادیر α کوچک‌تر گرچه همگرایی کندتری را سبب می‌شوند، اما جواب نزدیک‌تری به مقدار احتمال واقعی دارند. نکته حائز اهمیت در آن است که واریانس به صفر نمی‌رسد.

دوم این که گرچه وجود واریانس ناخوشایند به نظر می‌رسد، اما یک مقدار محدود و کوچک واریانس باعث می‌شود با مشاهده داده‌های جدید، الگوریتم از گیرافتادن در نقاط بیشینه محلی نجات یابد. علاوه بر این که باعث می‌شود قابلیت انطباق با محیط‌های متغیر را نیز پیدا کند. سرعت این انطباق‌پذیری به کمک α یا همان نرخ یادگیری قابل کنترل است. ضمناً گیرنیفتادن در نقاط بیشینه محلی، تضمین نشده است و به مقدار α و شکل تابع توزیع بستگی دارد.

سوم این که نامساوی چبیشف یک بازه اطمینان از مقدار احتمالی تخمین زده شده توسط اتوماتا نسبت به واریانس تخمین به دست می‌دهد. این ویژگی می‌تواند به انتخاب یک α مناسب کمک کند.

تمام الگوریتم‌هایی که سعی می‌کنند به گونه‌ای پویا نرخ یادگیری را تنظیم کنند، از این نکته استفاده می‌کنند که در زمانی که همگرایی رخ می‌دهد، نرخ یادگیری را کاهش می‌دهند. به بیان دیگر چنانچه اختلاف میان احتمال تخمین جدید و میانگین تخمین‌های قبلی بزرگ باشد، نرخ یادگیری افزایش می‌یابد. این حالت زمانی رخ می‌دهد که یک بیشینه محلی وجود داشته باشد و یا محیط تصادفی تغییر کند. در الگوریتم پیشنهادی جدید از ویژگی سوم مطرح در گزاره ۲ به عنوان عامل تشخیص‌دهنده اختلاف میان تخمین جدید و میانگین تخمین‌های قبلی استفاده می‌کنیم و به کمک آن در موارد لازم، نرخ یادگیری را افزایش می‌دهیم.

در اکثر سیستم‌های یادگیر و از جمله در اتوماتای یادگیر، نرخ یادگیری در مدت آموزش، به شکل پویا- کاهش‌یابنده با زمان- تغییر می‌کند. دلیل این امر هم واضح است. در ابتدای یادگیری، مقادیر بزرگ‌تر نرخ یادگیری، باعث آموزش سریع‌تر می‌گردند. به تدریج و با افزایش تعداد نمونه‌ها، سیستم یادگیرنده سعی می‌کند بیشتر متکی بر تجربیات آموزشی گذشته باقی بماند تا این که سعی کند از نمونه‌های جدید برای یادگیری استفاده

Proposed Algorithm L_{R-I} (1)

Parameters: Real $TSH \ll 1$, $q > 1$, $\bar{\alpha}$ Learning Rate Vector,
Initialization:
  $p_j \leftarrow 1/K$, $\mu_j = p_j$, $\delta_j \leftarrow 0$, $t_j \leftarrow 0$ **for** $j \leftarrow 1$ **to** K
loop
  Draw randomly an action i according to probabilities $p_0, \cdots, p_K$
  Receive either *reward* or *penalty*
  if *reward* **then**
    for $j \leftarrow 1$ **to** K **do**
      if $j \neq i$ **then** $p_j \leftarrow (1-\alpha_i)p_j$
      else $p_i \leftarrow p_i + \alpha_i(1-p_i)$
      end if
    end for
    for $j \leftarrow 1$ **to** K **do**
      update (μ_j)
      update (δ_j)
      if $|p_j - \mu_j| > q\delta_j$
        increment (α_j) ; $t_j \leftarrow 0$; reset μ_j
        else if $(1-\alpha_j)^{t_j} < TSH$
          decrement (α_j) ; $t_j \leftarrow 0$
        else $t_j \leftarrow t_j + 1$
        end if
      end for
    end if
end loop

شکل ۳: الگوریتم شماره ۱ مبتنی بر نامساوی چبیشف در حالت ساده.

کند. این منطق باعث می‌شود در ابتدای فرایند یادگیری، سیستم یادگیرنده جسورانه‌تر و با گذر زمان محافظه‌کارانه‌تر عمل کند.

اکثر سیستم‌های یادگیری از یک نرخ یادگیر پویای کاهش‌یابنده در حین فرایند یادگیری استفاده می‌کنند. نرخ یادگیری با این مفهوم پارامتری است که میزان فراموش‌کاری سیستم را تعریف می‌کند. مقادیر کوچک‌تر این پارامتر، یعنی اتکای بیشتر سیستم به تجربیات گذشته و مقادیر بزرگ‌تر به معنای نادیده‌گرفتن تجربیات گذشته است.

در بیشتر سیستم‌ها، منطق بالا پاسخگوی نیازها می‌باشد. مسأله اینجاست که کاهش نرخ یادگیری در طول زمان باعث می‌شود به مرور زمان، انطباق‌پذیری سیستم یادگیر کاهش یابد و نسبت به تغییرات در محیط عکس‌العمل مناسب نشان ندهد. بنابراین برای تنظیم اتوماتیک نرخ یادگیری در طول مدت آموزش بایستی معیاری داشته باشیم تا بر اساس آن نسبت به افزایش (در صورت بروز تغییرات جدی در محیط) یا کاهش (در صورت یکنواخت بودن پاسخ محیط و نزدیک‌شدن به همگرایی) نرخ یادگیری اقدام کنیم. این معیار بایستی قادر به تعیین میزان اهمیت تغییرات در محیط باشد.

۳- الگوریتم پیشنهادی

با توجه به رابطه یادگیری مورد استفاده توسط اتوماتای یادگیر برای هر اقدام و تفسیر برداری آن، می‌توان از نابرابری چبیشف به شکل ساده یا برداری آن برای تنظیم اتوماتیک نرخ یادگیری استفاده کرد. منطق کار بدین صورت است که:

الف) در حالت ساده: در این حالت برای هر یک از اقدام‌های اتوماتا یک نرخ یادگیری در نظر می‌گیریم. بر اساس نامساوی چبیشف (یا همان مارکوف (۱۳))، چنانچه اختلاف مقدار احتمال انتخاب یک اقدام اتوماتا با میانگین مقادیر قبلی آن از یک مقدار آستانه بیشتر باشد (نامساوی چبیشف)، به منزله تغییرات زیاد در محیط بوده و نرخ یادگیری را افزایش می‌دهد. هم‌زمان با افزایش نرخ یادگیری در هر بار، مقدار میانگین با صفر مقداردهی شده تا الگوریتم قابلیت گریز از بیشینه‌های محلی را داشته باشد (گرچه همان طور که توضیح دادیم، این فرار از بیشینه‌های محلی تضمین‌شده نیست).

در این الگوریتم برای اجتناب از سربار محاسباتی، از یک کران بالا برای مقدار واریانس استفاده می‌کنیم. برای رسیدن به این کران بالا قسمت ب گزاره ۲ و اثبات آن را در نظر بگیرید. نشان داده شد که مقدار واریانس در زمان $t+1$ از (۲۰) به دست می‌آید و (۲۱) نشان می‌دهد که واریانس تابعی از میانگین (C^*) و نرخ یادگیری (α) است. شکل ۲ رابطه میان واریانس و میانگین را به ازای نرخ‌های مختلف یادگیری نشان می‌دهد. همان گونه که این شکل نیز نشان می‌دهد، هرچه میانگین به صفر یا یک نزدیک‌تر باشد، واریانس کوچک‌تر است. بر عکس در میانگین برابر با ۰/۵ واریانس، بیشینه مقدار را دارد. علاوه بر این می‌توان دید که هرچه نرخ یادگیری به ۱ نزدیک‌تر باشد (نمودار بالایی) واریانس مقدار بیشتری دارد و بالعکس، در مقادیر کوچک‌تر نرخ یادگیری (پایین‌ترین نمودار)، واریانس کمتری داریم. برای سهولت در محاسبه واریانس از یک کران بالا برای آن استفاده می‌کنیم. بدین صورت که مقدار واریانس به ازای میانگین ۰/۵ را در محاسبات در نظر می‌گیریم. بدین ترتیب محاسبه واریانس در هر دور تنها تابعی از نرخ یادگیری خواهد بود. این کران بالا با مقدار q کوچک‌تر در الگوریتم جبران می‌شود.

برای محاسبه میانگین نیز از یک میانگین‌گیری روی مقادیر مربوط به احتمالات استفاده می‌کنیم. هر زمان که نرخ یادگیری افزایش می‌یابد، میانگین‌های قبلی را در نظر نگرفته و میانگین‌گیری را روی مقادیر جدید آغاز می‌کنیم. این کار باعث گریز از بیشینه‌های محلی می‌شود. برای افزایش و کاهش مقدار نرخ یادگیری آنها را در یک مقدار ثابت بزرگ‌تر از ۱ ضرب (برای افزایش) یا تقسیم (برای کاهش) می‌کنیم.

با این اصلاحات الگوریتم نهایی پیشنهادی، در حالت ساده آن به شکل ۳ خواهد بود.

ب) در حالت برداری: در این حالت، بردار احتمال انتخاب اقدام‌های اتوماتای یادگیر را در نظر گرفته و به این صورت بر خلاف حالت قبل، اتوماتای یادگیر یک نرخ یادگیری دارد که بر اساس نامساوی چبیشف در حالت برداری، مقدار آن تنظیم می‌شود. هر زمان که بردار جدید یاد گرفته شده توسط اتوماتای یادگیر، بیش از یک میزان مشخص از بردار میانگین مقادیر قبلی فرا گرفته شده توسط اتوماتا فاصله داشته باشد، به معنای وجود تغییرات گسترده در محیط است. برای این که بتوان اثر این تغییرات را منعکس کرد اتوماتا بایستی نرخ یادگیری را افزایش دهد. ملاک سنجش میزان تغییرات نابرابری چبیشف برداری است. همانند حالت ساده، در اینجا نیز با هر بار افزایش مقدار نرخ یادگیری برای گریز از بیشینه محلی، مقدار میانگین با صفر مقداردهی می‌شود.

به این ترتیب الگوریتم پیشنهادی جدید در حالت برداری آن به شکل ۴ خواهد بود.

در هر دو الگوریتم پیشنهادی مبتنی بر نابرابری چبیشف، اگر با نرخ یادگیری α پس از t بار، مقدار $(1-\alpha)^t$ از یک مقدار آستانه ثابت کمتر باشد، بیانگر آن است که واریانس به مقدار حدی خود- به ازای آن

Proposed Algorithm L_{R-I} (2)

Parameters: Real $q>1$, $\alpha<1$ Learning Rate
Initialization:
  $p_j \leftarrow 1/K$, $\mu_j = p_j$, $t \leftarrow 0$, $d_t \leftarrow 0$ **for** $j \leftarrow 1$ **to** K
loop
  Draw randomly an action i according to probabilities $p_0,\dots,p_K$
  Receive either *reward* or *penalty*
  if *reward* **then**
    for $j \leftarrow 1$ **to** K **do**
      if $j \neq i$ **then**
        $p_j \leftarrow (1-\alpha)p_j$
      else
        $p_i \leftarrow p_i + \alpha(1-p_i)$
      end if
    end for
    $t \leftarrow t+1$;
    $dt \leftarrow dt+1$;
    $\bar{\mu} = \text{mean}(\vec{P}(i-dt-1),\dots,\vec{P}(i))$
    $\vec{V} = \text{Cov}(\bar{\mu},\vec{P})$
    $N = \text{trace}(\vec{V}\vec{V}^{-1})$
    if $(\vec{P}-\vec{V})^T\vec{V}^{-1}(\vec{P}-\vec{V}) > qN$
      increment (α) ; $t \leftarrow 0$; $dt \leftarrow 0$;
    else if $(1-\alpha)^t < \text{TSH}$
      decrement (α) ; $t \leftarrow 0$;
    end if
  end if
end loop

شکل ۴: الگوریتم پیشنهادی جدید مبتنی بر نامساوی چبیشف به شکل برداری.

نرخ یادگیری خاص- نزدیک شده است. این نشان‌دهنده ثبات در محیط است و می‌توان با کاهش نرخ یادگیری، اثرپذیری اتوماتا از نمونه‌های ورودی جدید را کاهش داد. دلیل انتخاب این ملاک برای کاهش نرخ یادگیری به قسمت ب اثبات گزاره ۲ برمی‌گردد که واریانس را به شکل تابعی از میانگین واقعی نشان می‌دهد. بر اساس آنچه که در قسمت ب گزاره ۲ دیدیم مقدار واریانس در زمان تابعی از $(1-\alpha)^{2t}$ است که در آن α نرخ یادگیری و t تعداد دفعاتی است که نرخ یادگیری بدون تغییر برای آموزش مورد استفاده قرار گرفته است. بدین ترتیب الگوریتم می‌تواند از $(1-\alpha)^t$ به عنوان معیاری استفاده کند که میزان ثبات در محیط را نشان می‌دهد و در حقیقت به جای آن که منتظر بماند تا بر اثر افزایش مقدار t مقدار $Var[X^{t+1}]$ به مقدار حدی $(\alpha/(2-\alpha))C^*(1-C^*)$ نزدیک شود، با کاهش مقدار نرخ یادگیری α این فرایند تسریع شود.

٤- بررسی نتایج شبیه‌سازی

برای بررسی نحوه عملکرد روش پیشنهادی جدید، یک اتوماتای تصادفی با ۲ اقدام در نظر گرفته‌ایم که توسط یک محیط تصادفی مورد ارزیابی قرار می‌گیرد. محیط تصادفی، یک محیط پویا در نظر گرفته شده است. بدین صورت که تابع ارزیابی اقدام اتوماتا توسط محیط یک مقدار ثابت فرض نشده و در طول زمان آموزش تغییر می‌کند. الگوریتم مورد استفاده توسط اتوماتای یادگیر الگوریتم L_{R-I} و $L_{R-\varepsilon}$ های پیشنهادی است.

برای انجام شبیه‌سازی، توابع مختلفی را بر حسب زمان به عنوان تابع پاداش یا جریمه محیط در نظر گرفته‌ایم. برای هر محیط، یک اتوماتا را هم‌زمان به دو شیوه آموزش داده‌ایم. روش اول همان روش معمول مورد استفاده در اتوماتای یادگیر است، به این صورت که با توجه به پویایی محیط، اتوماتای یادگیر از یک نرخ یادگیری کوچک- بین ۰/۱ و ۰/۲- آغاز کرده و با کاهش آن به اندازه ۰/۰۱ در هر دور اجرای الگوریتم (تا رسیدن به آستانه ۰/۰۱) فرایند یادگیری را انجام داده است. روش دوم مبتنی بر الگوریتم‌های پیشنهادی جدید است. الگوریتم‌های پیشنهادی جدید نیز از یک نرخ یادگیری دلخواه (و غالباً بزرگ نزدیک به ۱) آغاز می‌کنند. نتایج هر دو الگوریتم پیشنهادی در مقایسه با روش معمول در ادامه آورده شده و نتایج آن شرح داده شده است. معیار مقایسه، میزان انطباق بردار آموزش داده شده در روش جدید پیشنهادی و روش معمول با بردار ارزیابی محیط یا همان احتمال جریمه اقدام انجام‌شده توسط اتوماتا است (که قاعدتاً مقدار ثابتی نداشته و بر حسب زمان در تغییر است). از نرم[1] بردار احتمال انتخاب اقدام‌های یاد گرفته شده و بردار احتمالی واقعی به عنوان معیاری برای مقایسه میزان انطباق‌پذیری اتوماتا با محیط استفاده کرده‌ایم. نتایج را برای محیط‌های پویای مثال در ادامه مشاهده می‌کنید.

٤-۱ بررسی نتایج الگوریتم پیشنهادی ۱

در این گروه از آزمایش‌ها از الگوریتم شماره ۱ (شکل ۳) استفاده شده است.

آزمایش ۱: در اولین نمونه از تابع (۲۳) به عنوان تابع ارزیابی اقدام شماره ۱ اتوماتا استفاده کرده‌ایم

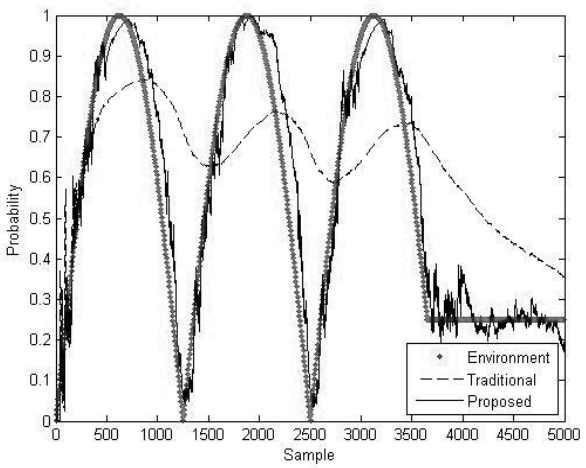
$$f(i)=\begin{cases}\left|\sin\dfrac{4i\pi}{n}\right| & , \quad i<0.73n \\ \left|\sin\dfrac{4t\pi}{n}\right| & , \quad t=0.73, i\geq 0.73n\end{cases} \qquad (۲۳)$$

در (۲۳)، i معرف iامین نمونه و n نشان‌دهنده تعداد کل نمونه‌ها است. برای بررسی عملکرد روش جدید و مقایسه آن، هم‌زمان اتوماتا را با الگوریتم ۱ و نیز الگوریتم L_{RI} با نرخ یادگیری کوچک $\alpha=0.002$ و تکنیک سردکردن تدریجی با ضریب کاهش ۰/۰۰۱ در هر دور، آموزش داده‌ایم $(\alpha^{t+1}=0.999\times\alpha^t)$. نتیجه مقایسه این دو در شکل ۵ آورده شده است. شکل ۶ مقدار نرم (فاصله) هر یک از دو بردار آموزش داده شده به روش معمول و روش پیشنهادی با بردار واقعی را نشان می‌دهد و در شکل ۷ نحوه تغییر (افزایش یا کاهش) نرخ یادگیری در الگوریتم پیشنهادی آمده است.

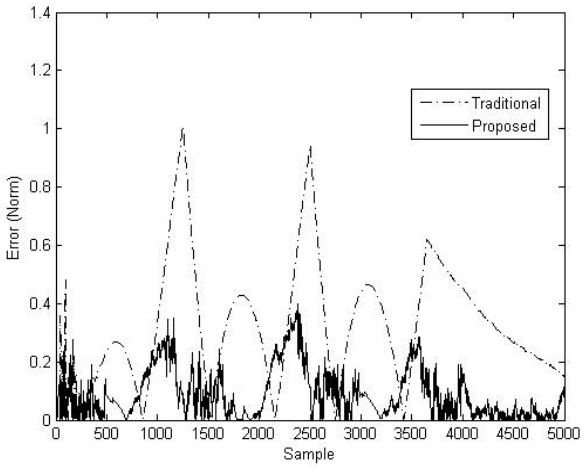
همچنان که شکل ۷ نشان می‌دهد، افزایش مقدار نرخ یادگیری در موقعیت‌هایی رخ داده که تغییرات بزرگی در محیط ایجاد شده است و در نتیجه، بردار احتمال انتخاب اقدام‌های اتوماتا به بردار واقعی محیط نزدیک‌تر است.

آزمایش ۲: در این آزمایش از یک محیط تصادفی استفاده کرده‌ایم که تابع ارزیابی آن در طول زمان به شکل پله‌ای تغییر می‌کند. این تابع $f(i)=0.2\times[5i/n]$ است که i معرف iامین نمونه و n نشان‌دهنده تعداد کل نمونه‌ها می‌باشد.

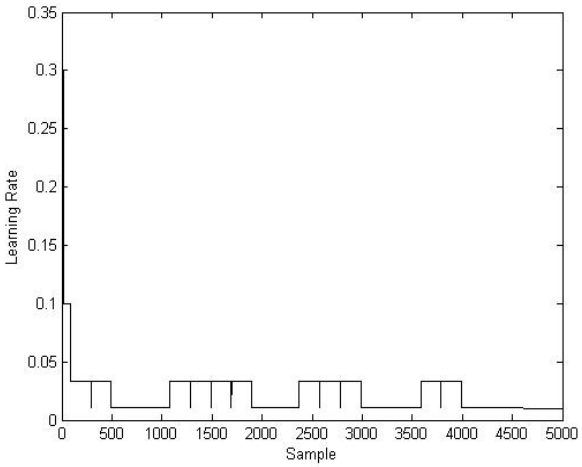
1. Norm



شکل ۵: خط تیره درشت تابع ارزیابی اقدام اتوماتا توسط محیط را نشان می‌دهد. خط تیره نازک نحوه تغییر بردار احتمال مربوط به اقدام اتوماتای آموزش داده شده به شیوه جدید و خطوط نقطه‌چین، همان بردار، آموزش داده شده به شیوه معمول را نشان می‌دهد (آزمایش ۱).

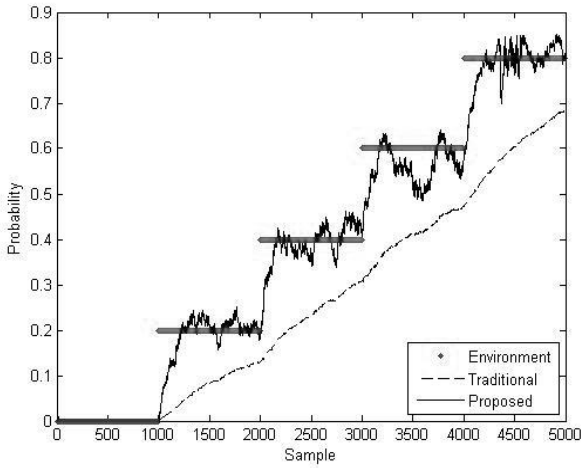


شکل ۶: فاصله بردارهای آموزش داده شده به هر یک از دو شیوه با بردار واقعی در آزمایش شماره ۱.

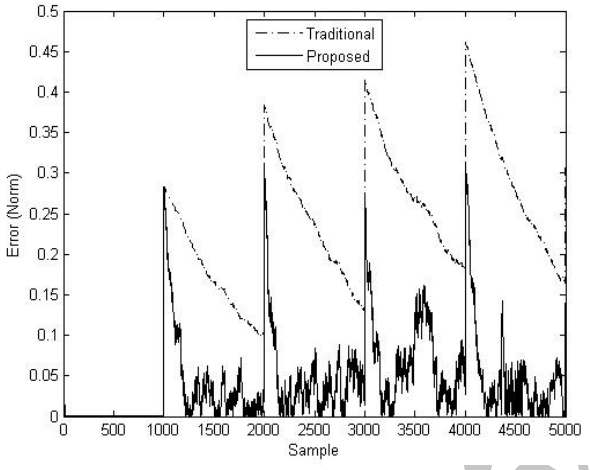


شکل ۷: تغییرات نرخ یادگیری در آزمایش ۱.

نتایج بررسی عملکرد مقایسه‌ای الگوریتم پیشنهادی و الگوریتم معمول (بهینه‌شده) در تنظیم نرخ یادگیری را در شکل ۸ و شکل ۹ مشاهده می‌کنید. شکل ۸ نشان می‌دهد که الگوریتم پیشنهادی انطباق بیشتری با رفتار پویای محیط داشته و شکل ۹ نیز مؤید میزان خطای کمتر روش جدید در مقایسه با روش معمول است.



شکل ۸: مقایسه روش معمول (خطوط نازک بریده) در مقایسه با روش پیشنهادی (خطوط تیره پیوسته) در یک محیط پویا (خطوط تیره پلکانی) در آزمایش ۲.



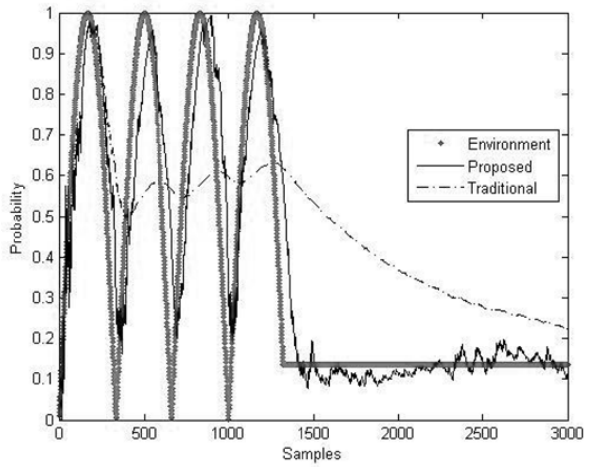
شکل ۹: فاصله بردارهای آموزش داده شده به هر یک از دو شیوه با بردار واقعی در آزمایش شماره ۲.

٤-۲ بررسی نتایج الگوریتم پیشنهادی ۲

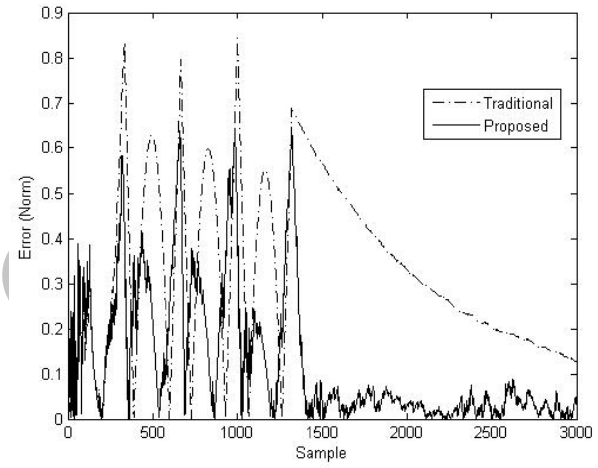
در این گروه از آزمایش‌ها از الگوریتم ۲ (شکل ۴) استفاده کرده‌ایم. الگوریتم جدید از یک نرخ یادگیری دلخواه بزرگ آغاز کرده. و در هر دور اجرا بر اساس میزان فاصله بردار جدید احتمال انتخاب به دست آمده برای اقدام‌های اتوماتا از بردار میانگین، اقدام به کاهش یا افزایش نرخ یادگیری نموده است.

در این الگوریتم، اتوماتای یادگیر یک نرخ یادگیری دارد و نامساوی چبیشف در حالت برداری آن به عنوان معیار تشخیصی برای تعیین آن که آیا تغییر بزرگی در محیط رخ داده است یا نه مورد استفاده قرار می‌گیرد. نتیجه را در مورد نمونه‌هایی از محیط‌های پویا در ادامه بررسی کرده‌ایم.

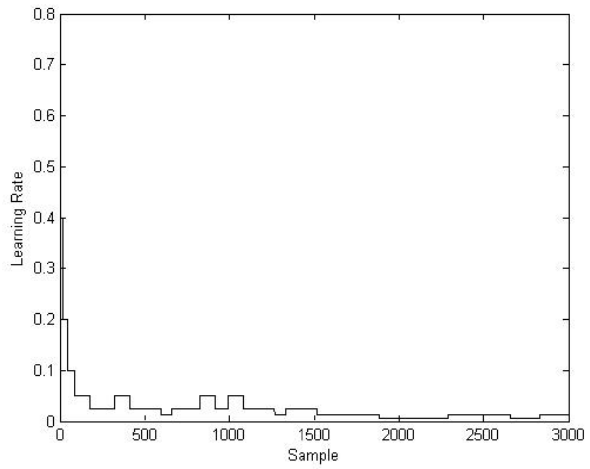
آزمایش ۳: محیط پویایی که برای این بررسی انتخاب شده است تا حدود ۴۰٪ زمان آموزش از یک تابع پویا به صورت $\left|\sin(9i\pi/n)\right|$ برای ارزیابی اقدام اتوماتا استفاده می‌کند (در iامین نمونه که n تعداد کل نمونه‌ها است). اما از این مرحله به بعد، تابع به شکل خطی ثابت و بدون تغییر، اقدام اتوماتا را مورد ارزیابی قرار می‌دهد. نتایج مقایسه‌ای را در شکل ۱۰ و شکل ۱۱ مشاهده می‌کنید. در شکل ۱۲ نیز نحوه تغییرات نرخ یادگیری نشان داده شده است. تلاش اتوماتا برای انطباق با محیط در بخش اول نمونه‌ها که محیط از پویایی بالایی برخوردار است، از طریق افزایش نرخ یادگیری قابل مشاهده است.



شکل ۱۰: خط تیره‌رنگ درشت تابع ارزیابی اقدام اتوماتا توسط محیط را نشان می‌دهد. خط تیره‌رنگ نازک نحوه تغییر بردار احتمال مربوط به اقدام اتوماتای آموزش داده شده به شیوه جدید و خطوط نقطه‌چین، همان بردار، آموزش داده شده به شیوه معمول را نشان می‌دهد.

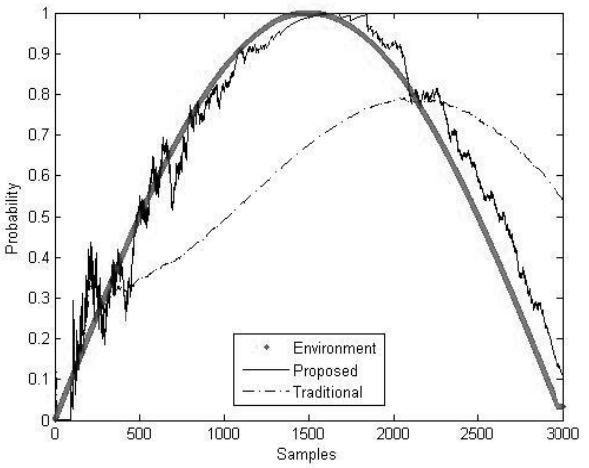


شکل ۱۱: فاصله بردارهای آموزش داده شده به هر یک از دو شیوه با بردار واقعی در آزمایش شماره ۳.

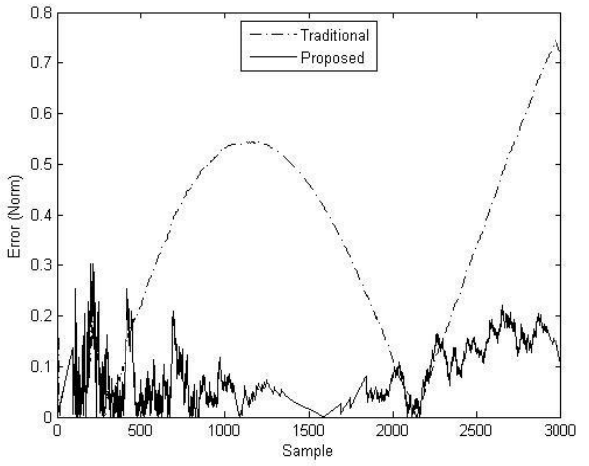


شکل ۱۲: تغییرات نرخ یادگیری در آزمایش ۳.

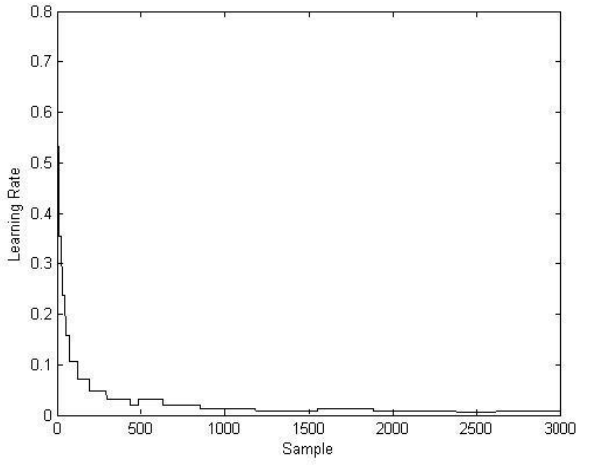
آزمایش ٤: برای این نمونه، محیط دیگری در نظر گرفته‌ایم که تابع ارزیابی محیط (احتمال پاداش به اقدام) از یک رفتار شبه‌سینوسی برخوردار است. نتیجه مقایسه‌ای عملکرد الگوریتم یادگیری L_{R-I} و الگوریتم جدید را در شکل‌های ۱۳ و ۱۴ مشاهده می‌کنید. نحوه تغییرات افزایشی و کاهشی نرخ یادگیری در شکل ۱۵ نشان داده شده است.



شکل ۱۳: مقایسه روش معمول (خطوط نازک بریده) در مقایسه با روش پیشنهادی (خطوط تیره پیوسته) در یک محیط پویا (شبه‌سینوسی) در آزمایش ۴.



شکل ۱۴: فاصله بردارهای آموزش داده شده به هر یک از دو شیوه با بردار واقعی در آزمایش شماره ۴.



شکل ۱۵: تغییرات نرخ یادگیری در آزمایش ۴.

۵- نتیجه‌گیری

در این مقاله به کمک شاخص‌های آماری حاصل از توزیع احتمال بردار انتخاب اقدام‌های اتوماتا و با کمک نابرابری چبیشف، روش‌های جدیدی پیشنهاد شد که به کمک آن اتوماتای یادگیر ضمن تنظیم خودکار نرخ یادگیری، قادر به یادگیری در محیط‌های با پویایی بالا است.

این الگوریتم‌های جدید قادر به تنظیم اتوماتیک نرخ یادگیری بوده و

می‌توانند بر حسب میزان تغییرات در محیط و به منظور انطباق با آن، اقدام به کاهش یا افزایش نرخ یادگیری نمایند. استفاده از این الگوریتم‌های جدید در محیط‌های تصادفی پویا که پاسخ محیط به اقدام انجام‌شده توسط اتوماتا غیر ثابت و تابعی از زمان است مورد بررسی قرار گرفت و نشان داده شد که الگوریتم‌های جدید مبتنی بر نامساوی چبیشف از عملکرد بهتری نسبت به روش‌های یادگیری معمول برخوردارند.

مراجع

[1] H. Beigy, M. R. Meybodi, and M. B. Menhaj, "Adaptation of learning rate in back propagation algorithm using fixed structure learning automata," in *Proc. 6th Iranian Conf. on Electrical Engineering*, pp. 117-123, Tehran, Iran, 1998.

[2] H. Shah-Hosseini and R. Safabaksh, "Automatic adjustment of learning rates of the self-organizing feature map," *Scientia Iranica*, vol. 8, no. 4, pp. 277-286, Oct. 2001.

[3] C. Chinrungrueng and C. H. Sequin, "Optimal adaptive k - means algorithm with dynamic adjustment of learnig rate," *IEEE Trans. on Neural Networks*, vol. 6, no. 1, pp. 157-169, Jan. 1995.

[4] J. Akbari Torkestani and M. R. Meybodi, "Learning automata - based algorithms for solving stochastic minimum spanning tree problem," *Applied Soft Computing*, vol. 11, no. 6, pp. 4064-4077, Sep. 2011.

[5] J. Akbari Torkestani and M. R. Meybodi, "Learning automata - based algorithms for finding minimum weakly connected dominating set in stochastic graphs," *International J. of Uncertainty, Fuzziness, and Knowledge - Based Systems*, vol. 18, no. 6, pp. 721-758, Dec. 2010.

[6] H. Beigy and M. R. Meybodi, "Utilizing distributed learning automata to solve stochastic shortest path problems," *International J. of Uncertainty,* vol. 14, no. 5, pp. 591-615, Oct. 2006.

[7] M. R. Mollakhalili Meybodi and M. R. Meybodi, "A new distributed learning automata based algorithm for solving stochastic shortest path," in *Proc. 6th Conf. on Intelligent Systems*, Kerman, Iran, 26-27 Nov. 2004.

[8] M. R. Meybodi and H. Beigy, "Solving stochastic shortest path problem using distributed learning automata," in *Proc. 6th Annual CSI Computer Conf., CSICC 2001*, pp. 70-86, Isfehan, Iran, Feb. 2001.

[9] H. Beigy Intelligent Channel Assignment in Cellular Networks: A Learning Automata Approach, Ph.D. Thesis, Amirkabir University of Technology, 2004.

[10] M. L. Thathachar and P. S. Sastry, "Varieties of learning automata: an overview," *IEEE Trans. on Systems, Man, and Cybernetics, Part B, Cybernetics*, vol. 32, no. 6, pp. 711-22, Jan. 2002.

[11] K. S. Narenda and M. A. L. Thatacher, *Learning Automata*, Prentice-Hall, 1989.

[12] M. L. Tsetlin, "On the behaviour of finite automata in random media," *Automata., Telemekh.*, vol. 22, pp. 1345-1354, Oct. 1961.

[13] B. J. Oomen and H. Masum, "Switching models for nonstationary random environments," *IEEE Trans. on Systems, Man, and Cybernetics.*, vol. 25, no. 9, pp. 1334-1347, Sep. 1995.

[14] K. S. Narendra and M. A. L. Thathachar, "On the behavior of a learning automaton in a changing environment with application to telephone traffic routing," *Systems, Man, and Cybernetics, IEEE Trans. on*, vol. 10, no. 5, pp. 262-269, May 1980.

[15] S. M. Ross, Introduction to Probability and Statistics dor Engineers and Scientists, 3rd. Elsevier Academic Press, 2004.

[16] A. Papoulis, *Probability, Random Variables, and Stochastic Processes*, 3rd Ed. New York, USA: McGrawHill, 1991.

محمدرضا ملاخلیلی میبدی تحصیلات خود را در مقاطع کارشناسی و کارشناسی ارشد مهندسی کامپیوتر به‌ترتیب در سال‌های ۱۳۸۰ و ۱۳۸۲ از دانشگاه‌های شهید بهشتی تهران و صنعتی امیرکبیر به پایان رسانده است. وی از سال ۱۳۸۲ به عضویت هیأت علمی دانشگاه آزاد اسلامی واحد میبد درآمده و در حال حاضر مشغول به تحصیل در دوره دکتری کامپیوتر در واحد علوم و تحقیقات دانشگاه آزاد اسلامی می‌باشد. زمینه‌های تحقیقاتی مورد علاقه ایشان عبارتند از: شبکه‌های کامپیوتری و وب، محاسبات نرم و کاربردهای آن، یادگیری والگوریتم‌ها، گراف‌های تصادفی و شبکه‌های پیچیده.

محمدرضا میبدی تحصیلات خود را در مقاطع کارشناسی و کارشناسی ارشد اقتصاد به‌ترتیب در سال‌های ۱۳۵۲ و ۱۳۵۶ از دانشگاه شهید بهشتی و در مقاطع کارشناسی ارشد و دکتری علوم کامپیوتر به‌ترتیب در سال‌های ۱۳۵۹ و ۱۳۶۲ از دانشگاه اوکلاهمای آمریکا به پایان رسانده است و هم‌اکنون استاد دانشکده مهندسی کامپیوتر دانشگاه صنعتی امیرکبیر می‌باشد. نام‌برده قبل از پیوستنش به دانشگاه صنعتی امیرکبیر در سال‌های ۱۳۶۲ الی ۱۳۶۴ استادیار دانشگاه میشیگان غربی و در سال‌های ۱۳۶۴ الی ۱۳۷۰ دانشیار دانشگاه اوهایو در ایالات متحده آمریکا بوده است. زمینه‌های تحقیقاتی مورد علاقه ایشان عبارتند از: الگوریتم‌های موازی، پردازش موازی، محاسبات نرم و کاربردهای آن، شبکه‌های کامپیوتری و مهندسی نرم افزار.