

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

مؤسسه آموزش عالی اشراق

واحد بجنورد

# یادگیری ماشین

ارائه شده به:

استاد سرگلزایی

توسط:

یونس خرامان

شیوه ارائه مطالب علمی - فنی

کارشناسی ناپیوسته کامپیوتر

ترم اول ۹۰-۸۹

## فهرست

صفحه	عنوان
<b>فصل اول : یادگیری تقویتی</b>	
۵	مقدمه
۵	تاریخچه یادگیری تقویتی
۵	تعریف یادگیری تقویتی
۶	الگوریتم یادگیری تقویتی
۶	توضیح الگوریتم
۷	استراتژیهای الگوریتم تقویتی
۸	برنامه نویسی پویا
۸	اصل بهینگی Bellman
۸	الگوریتم ژنتیک
۹	کاربردهای الگوریتم ژنتیک
۹	اجزای اصلی یادگیری تقویتی
۱۱	کاربردهای یادگیری تقویتی
<b>فصل دوم (یادگیری بیزی)</b>	
۱۴	مقدمه
۱۴	نگرش بیزی به یادگیری ماشین
۱۴	ویژگیهای یادگیری بیزی
۱۵	ایراد روش یادگیری بیزی
۱۵	روشهای یادگیری بیزی
۱۵	خلاصه

## فصل اول

# یادگیری تقویتی

بشر برای سازگار کردن خود با محیط اطرافش قادر است به یادگیری مهارت‌های جدید بپردازد. اگرچه یادگیری و سازگاری هر دو فرآیندهایی پیچیده اند. به اعتقاد روانشناسان هر رفتاری که از ماسر میزند معلول یادگیری است و انسان برای غلبه بر این دگرگونی‌ها ناچار به یادگیری است. به عبارت دیگر یادگیری توانایی بهبود رفتار براساس تجربیات و مشاهدات قبلی است. پس یادگیری ماشین بطور گسترده در هوش مصنوعی مطرح شد که سعی بر این است با یادگیری بتوان ربات‌های قابل انعطاف تر و هوشمندتر ایجاد کرد. به همین دلیل در یادگیری ماشین هدف برنامه ریزی کردن رایانه‌ها به صورتی که بتوانند از تجربیات گذشته برای حل یک مسئله داده شده استفاده کنند.

به طور معمول یادگیری به سامانه‌ای که قادر است به طور خودکار یاد بگیرد و دانش‌هایش را یکپارچه کند بر می‌گردد. با مجهز شدن به ابزاری مثل یادگیری، یک سامانه می‌تواند به طور پیوسته عملکردش را بهینه کند و کارایی آن بیشتر شود که در تحقیق در مورد یادگیری تقویتی<sup>۱</sup> که یکی از پرکاربردترین زمینه‌های تحقیقاتی در زمینه یادگیری ماشین است سخن خواهیم گفت.

## تاریخچه یادگیری تقویتی

تاریخچه یادگیری تقویتی دو ریشه اصلی دارد که هر دو قدیمی و معتبرند و قبل از یادگیری تقویتی جدید مورد بحث بوده است. یک ریشه آن که یادگیری با سعی و خطا را بیان میکند و منشأ آن روانشناسی یادگیری حیوانات است که در سال ۱۹۱۱ مطرح شد و در نخستین کارها در هوش مصنوعی به کار گرفته شد تا در اوایل سال ۱۹۸۰ باعث ایجاد یادگیری تقویتی شد.

ریشه دیگر آن مسائلی از کنترل بهینه<sup>۲</sup> را بیان میکند. واژه «کنترل بهینه» که در اواخر ۱۹۵۰ با مسائلی برای طراحی یک کنترلر برای به حداقل رساندن اندازه رفتار دینامیکی سامانه بیان شد. یکی از دیدگاه‌های که برای این مسئله در اواسط سال ۱۹۵۰ توسط بلمن مطرح شد به رابطه بلمن معروف است.

اگرچه این دوریشه کاملاً مستقل از هم بودن، باعث ایجاد ریشه دیگری گشتند که در سال ۱۹۵۹ مطرح شد. و آن را "TD-Learning" و سرانجام با توسعه الگوریتم "Q-Learning" در سال ۱۹۸۹ ایده‌های تفاضل زمانی و کنترل بهینه با یکدیگر بطور کامل آورده شد. این کار که توسعه یافته تمام کارهای قبلی در سه ریشه از یادگیری تقویتی است رشد زیادی در تحقیقات یادگیری ماشین و شبکه‌های عصبی داشت.

## تعریف یادگیری تقویتی

در یادگیری تقویتی عامل‌ها به حسگرهای مجهز شده اند که میتوانند مشخصه‌های قطعی محیط را مشاهده کنند که این مشخصه‌ها فضای حالت<sup>۳</sup> عامل یادگیرنده رو تشکیل میدهند. سپس در هر بازه زمانی عامل با انجام عملیاتی، محیط رو تحت تأثیر قرار میدهد. بنابراین عامل ورودیهای مختلفی رو در بازه زمانی بعدی با توجه به عمل قبلی در

<sup>۱</sup> -Reinforcement Learning

<sup>۲</sup> -Optimal Control

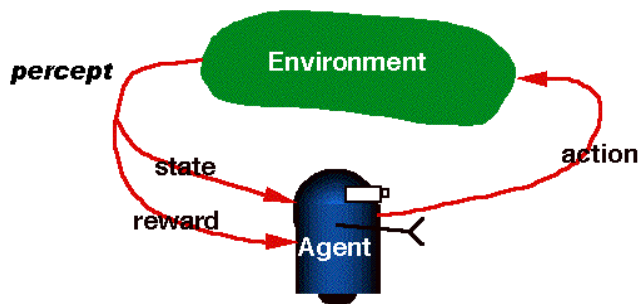
<sup>۳</sup> -State Space

یافت میکند. علاوه بر ورودی جدید، عامل یادگیرنده در هر زمان یک سیگنال تقویتی که میزان مطلوبیت کنش قبلی رو نشان می دهد دریافت میکند که به آن پاداش می گویند و باتوجه به این که کنش مناسب بوده یا نه این مقدار می تواند مثبت یا منفی باشد.

به طور کلی در در یک مسئله یادگیری تقویتی با عاملی<sup>۱</sup> روبرو هستیم که از طریق سعی و خطا با محیط تعامل کرده و یاد میگیرد تا عملی بهینه را برای رسیدن به هدف انتخاب نماید.

### الگوریتم یادگیری تقویتی

۱. مشاهده  $S_t$
۲. تصمیم گرفتن برای انجام کنش  $act$
۳. انجام کنش
۴. مشاهده حالت جدید  $S_{t+1}$
۵. مشاهده  $R_{t+1}$
۶. یادگیری از تجربیات
۷. تکرار



### توضیح الگوریتم

۱. محیط مجموعه ی از  $S$  حالت ممکن است.
۲. در هر لحظه  $t$  عامل میتواند یکی از  $A$  عمل ممکن را انجام دهد.
۳. عامل در محیط حرکت کرده و حالتها و پاداشهای مربوطه را به خاطر می سپارد
۴. عامل ممکن است در مقابل عمل و یا مجموعه ی از اعمالی که انجام میدهد پاداش  $r$  را دریافت کند. این پاداش ممکن است مثبت و یا منفی (تنبيه) باشد
۵. پاداش  $R_t$  مجموع پاداشی است که عامل با گذشت زمان  $t$  جمع کرده است.
۶. عامل سعی میکند طوری رفتار کند که تابع پاداش را ماکزیمم نماید.

$$R_t = r_{t+1} + \gamma r_{t+2} + \gamma^2 r_{t+3} + \dots = \sum_{k=0}^{\infty} \gamma^k r_{t+k+1} \quad 0 \leq \gamma \leq 1$$

<sup>۱</sup>-Agent

فرض می کنیم که اعمال عامل از قانونی مثل «پی» تبعیت میکنند که آنرا خط مشی و یا policy می نامیم. از آنجائیکه  $R_t$  یک متغیر تصادفی است لذا امید ریاضی آن تحت یک خط مشی خاص و بری یک حالت معین برابر خواهد بود با:

$$V^{\pi}(S_t) = E\{R_t | S_t, \pi\} = E\left\{\sum_{k=0}^{\infty} \gamma^k r_{t+k+1} | S_t, \pi\right\}$$

هدف یادگیری تقویتی این است که یک خط مشی بهینه ای مثل پی\* پیدا نماید به نحویکه مقدار امید ریاضی فوق را برای تمامی حالات ماکزیمم کند

### دو استراتژی اصلی در یادگیری تقویتی بکار می روند:

یادگیری تقویتی از اینرو مورد توجه است که راهی برای آموزش عاملها برای انجام یک عمل از طریق دادن پاداش و تنبیه است بدون اینکه لازم باشد نحوه انجام عمل را برای عامل مشخص نمائیم.

دو استراتژی اصلی برای اینکار وجود دارد:

یکی استفاده از الگوریتم های ژنتیکی

و دیگری استفاده از روشهای آماری و dynamic programming

در یادگیری تقویتی روش دوم مد نظر است.

### تفاوت یادگیری تقویتی با یادگیری نظارت شده

یادگیری تقویتی از دو جنبه با یادگیری با ناظر تفاوت دارد:

مثالهایی یادگیری بصورت زوج <ورودی خروجی> مطرح نمیشوند. بلکه بعد از اینکه عامل عملی را انجام داد پاداشی را دریافت میکند و به مرحله بعدی میرود. عامل هیچ گونه اطلاعی در مورد اینکه در هر حالت بهترین عمل چیست را ندارد. بلکه این وظیفه عامل است که در طول زمان تجربه کافی در مورد حالتها، عمل های ممکن، انتقال و پاداش جمع آوری نموده و عملکرد بهینه را یاد بگیرد.

تفاوت دیگر در اینجاست که سیستم باید کارائی آنلاین بالائی داشته باشد. زیرا اغلب ارزیابی سیستم با عمل یادگیری بطور همزمان صورت میپذیرد.

## برنامه نویسی پویا

همانگونه که گفته شد یک تفاوت اساسی میان یادگیری تقویتی و یادگیری با ناظر در این است که در یادگیری تقویتی عامل مجبور به جستجوی محیط است. در اینگونه مسائل با این سوال روبرو هستیم که وقتی که عامل در حالتی قرار میگیرد که پاداش مناسبی دریافت میکند آیا باید جستجو برای حالتی بهتر را ادامه دهد و یا باید در همان مرحله متوقف شود؟

تکنیکهای متفاوتی برای اینکار معرفی شده است که یکی از آنها برنامه نویسی پویاست.

✓ یادگیری تقویتی با ترکیب تکنیک *Dynamic Programming* با یادگیری با کمک ناظر به حل مسئله میپردازد.

✓ بطور کلی کاری که *Dynamic programming* انجام میدهد عبارت است از حل یک مسئله چند متغیره از طریق حل مجموعه ای مسائل تک متغیره

مبنای *dynamic programming* بر پایه اصل بهینگی *Bellman* بنا شده است

### اصل بهینگی *Bellman*

این اصل بسادگی بیان میکند که یک خط مشی بهینه باید دارای این خاصیت باشد که بدون توجه به حالت اولیه و تصمیمات اولیه گرفته شده، باقی تصمیمات باید با در نظر گرفتن حالت ایجاد شده از تصمیمات اولیه به خط مشی بهینه برسند.

در واقع *Dynamic programming* روشی است که برای حل یک مسئله از آخرین حالت ممکن شروع کرده و آنچه را که در آن حالت امکان پذیر است را بررسی مینماید، سپس با استفاده از اطلاعات بدست آمده از فرض بودن در آخرین حالت به حل حالت ماقبل آخر میپردازد و اینکار برای حالت های قبل از آن ادامه می یابد

### الگوریتم ژنتیک

الگوریتم ژنتیک<sup>1</sup> روش یادگیری بر پایه تکامل بیولوژیک است.

این روش در سال 1970 توسط *John Holland* معرفی گردید

یک الگوریتم ژنتیک برای حل یک مسئله مجموعه بسیار بزرگی از راه حلها را تولید میکند. هر یک از این راه حلها با استفاده از یک "تابع تناسب" مورد ارزیابی قرار میگیرد. آنگاه تعدادی از بهترین راه حلها باعث تولید راه حلهای جدیدی میشوند. که اینکار باعث تکامل راه حلها میگردد. بدین ترتیب فضای جستجو در جهتی تکامل پیدا میکند که به راه حل مطلوب برسد

---

<sup>1</sup> -Genetic Algorithm



الگوریتم های ژنتیک در مسائلی که فضای جستجوی بزرگی داشته باشند میتواند بکار گرفته شود.

همچنین در مسایلی با فضای فرضیه پیچیده که تاثیر اجرا آن در فرضیه کلی ناشناخته باشند میتوان از الگوریتم ژنتیک برای جستجو استفاده نمود.

الگوریتم های ژنتیک را میتوان براحتی بصورت موازی اجرا نمود از اینرو میتوان کامپیوترهای ارزان قیمت تری را بصورت موازی مورد استفاده قرار داد.

امکان به تله افتادن این الگوریتم در مینیمم محلی کمتر از سایر روشهاست.

از لحاظ محاسباتی پرهزینه هستند.

تضمینی برای رسیدن به جواب بهینه وجود ندارد.

### کاربردهای الگوریتم ژنتیک

الگوریتم ژنتیک در زمینه های مختلفی کاربرد دارد از جمله:

- optimization (بهینه سازی)
- automatic programming (برنامه نویسی خودکار)
- machine learning (یادگیری ماشین)
- economics (اقتصاد)
- operations research (پژوهش های عملیاتی)
- ecology (بوم شناسی)
- social systems (نظام اجتماعی)

### اجزای اصلی یادگیری تقویتی

- **عامل<sup>۱</sup>:** همان یادگیرنده است که از طریق محرکها و حسگرهایش با محیط در ارتباط است
- **محیط<sup>۲</sup>:** فضای که عامل کنشهای مربوطه را در آن انجام می دهد و روی آن تأثیر می گذارد
- **سیاست<sup>۳</sup>:** نگاشتی از مجموعه حالتها به کنشهاست و مشخص می کند چه کنشی را در هر حالت انجام دهیم.
- **تابع پاداش<sup>۴</sup>:** نگاشتی از زوج های حالت-کنش به یک عدد اسکالر با توجه به هدف یادگیری است و به عبارتی میزان مطلوبیت کنش را مشخص می کند.

---

<sup>۱</sup> -Agent

<sup>۲</sup> -Environment

<sup>۳</sup> - Policy

<sup>۴</sup> -Reward function

- ✓ در یادگیری تقویتی وقتی عامل در یک حالت خاص عملی را انجام میدهد، در مقابل پاداش (reward or reinforcement) دریافت میکند.
- ✓ یکی از نکات طراحی یک سیستم یادگیری تقویتی تعریف یک reinforcement function مناسب با اهداف عامل است. اینکار به طرق مختلف انجام میشود.
- ✓ در مسائل MDP با شرایطی مواجه هستیم که عامل میتواند S حالت مجزا را در محیط تشخیص دهد. این عامل قادر به انجام A عمل مجزا میباشد.
- ✓ در هر لحظه t عامل حالت  $s_t$  را تشخیص داده و عمل  $a_t$  را انجام میدهد.
- ✓ محیط در پاسخ به این عمل پاداش  $r_t = (s_t, a_t)$  را به عامل میدهد و به حالت بعدی  $s_{t+1} = p(s_t, a_t)$  میرود.
- ✓ توابع  $r, p$  جزئی از محیط بوده و برای عامل ناشناخته هستند.

در MDP توابع فقط به حالت و عمل فعلی بستگی داشته و از حالت و عمل های قبلی مستقل است

برای اینکه یک عامل بتواند تاثیر رخدادهای آینده در انتخاب عمل مناسب برای حالت فعلی را در نظر بگیرد مدل های مختلفی پیشنهاد شده است:

❖ Discounted cumulative reward

$$V^{\pi}(S_t) = r_t + \gamma r_{t+1} + \gamma^2 r_{t+2} + \dots = \sum_{k=0}^{\infty} \gamma^k r_{t+k} \quad 0 \leq \gamma \leq 1$$

❖ Finite horizon

$$V^{\pi}(S_t) = \sum_{k=0}^h r_{t+k}$$

❖ Average reward

$$V^{\pi}(S_t) = \lim_{h \rightarrow \infty} \frac{1}{h} \sum_{k=0}^h r_{t+k}$$

- تابع برگشت<sup>1</sup>: مقادیر پاداش یک عامل که از زمان شروع عملکرد آن تا آینده را جمع آوری می کند.

---

<sup>1</sup> -Return function

● **تابع ارزش** : مقدار یک حالت عبارت است از مجموع مقدار پاداشی که با شروع از آن حالت و پیروی از خط مشی مشخصی که به حالت نهایی ختم شود، دریافت می‌گردد.

تابع مقدار یا Value Function عبارت است از نگاشتی از states به state values که میتواند توسط هر تقریب زننده تابع نظیر یک شبکه عصبی تخمین زده شود.

**مثال:** یک مسئله MDP با 16 حالت را در نظر بگیرید:

عامل دارای 4 عمل مختلف است: حرکت به چپ، به راست، به بالا و به پایین پاداش برای تمامی حرکتهای برابر 1- است.

هدف رسیدن به دو گوشه سمت راست پائین و یا گوشه سمت چپ بالاست

مقادیر نشان داده شده مقدار مورد انتظار برای هر حالت در صورت انجام یک حرکت تصادفی برای رسیدن به هدف است

	0	-1	-2	-3	0	-14	-20	-22
	-1	-2	-3	-2	-14	-18	-22	-20
	-2	-3	-2	-1	-20	-22	-18	-14
	-3	-2	-1	0	-22	-20	-14	0

● **مدل** : که رفتار محیط رو به کمک آن شبیه سازی می کنند و با آن می توانند برای پیشگویی انتقال حالت قبل از اجرای یک کنش خاص استفاده شود و یک جزء اختیاری است

### کاربردهای یادگیری تقویتی

➤ **سامانه های چند عامله<sup>1</sup>** عامل هر چیزی است که می تواند محیطش را به کمک حسگرهایش مشاهده کند و سپس با محرکهایش<sup>2</sup> رور محیط تأثیر گذارد. عاملی در هوش مصنوعی مورد نظر است عامل هوشمند است. عاملهای هوشمند، دانش محیطی اضافه ای دارند که آنها را قادر میسازد حتی وقتی که پارامتری یک وظیفه نیز تغییر می کند و یا وقتی که شرایط غیر منتظره ای رخ می دهد بتوانند وظایفشان را به انجام برسانند. علاوه بر هوشمندی، خصوصیت مهم دیگر در عامل ها خودمختاری است. خودمختاری توانای عمل و اخذ تصمیم مستقل از برنامه نویس یا کاربر عامل است. همچنین با اضافه شدن خصوصیت همکاری در بین عامل ها، سامانه های چند عامله به عنوان الگویی از هوش مصنوعی توزیع شده مطرح شد.

این سامانه ها بیشتر با ساختاری متشکل از چند عامل که رفتارهای عامل های مختلف باید در یک محیط به سمت یک هدف جامع با هم هماهنگ شوند، بیان می شود. پس در این سامانه ها از چند عامل در کنارهم برای حل یک مسئله استفاده می شود و همچنین کنترل، سامانه های چند عامله به نوعی هم

<sup>1</sup> -Multi-Agent System

<sup>2</sup> -Actuator

متمرکز شده و هم توزیع شده است. به عبارتی هیچ پردازش مرکزی وجود ندارد که سامانه آن را در نظر بگیرد و سپس تصمیم گیری کند که چه کنش های باید انجام دهد و فرآیند تصمیم گیری هر عامل باید به دست خودش انجام شود که البته این تصمیم گیری روی مشاهداتش و دانش هایی که درباره محیط و عملهای دیگر بنا شده است، انجام میگیرد.

- **بازی ها** از یادگیری تقویتی برای چندین بازی اصلی که در جهان مطرح است استفاده شده است. در سال ۱۹۵۹ یک سامانه ارائه شد که به چکرزها یاد داده می شد که از راه بازی کردن با خودشان بازی کنند
- **هدایت کردن رباتها**<sup>۱</sup> هدایت کردن رباتها در محیط های ناشناخته برای انجام عملیات تحقیقاتی. در سال ۱۹۸۳ یک ربات قادر شد به کمک این یادگیری به اهداف تعیین شده با دوری از موانع برسد.
- **زمانبندی**<sup>۲</sup> یک استفاده دیگر از یادگیری تقویتی زمانبندی در کارهاست که در ابتدای سال ۱۹۹۶ تحقیق شد و با موفقیت دیدگاه TD-Learning در مسائل زمانبندی، این روش یادگیری در این زمینه هم مطرح شد.
- **مدیریت وزارت** سامانه های پشتیبان تصمیم برای سرمایه گذاران و مشتری ها به شدت در سالهای پیش مورد مطالعه بود. این سامانه ها سعی دارند که سود برگشتی برای سرمایه گذاران را بیشینه کند. دیدگاه اخیر از یادگیری تقویتی استفاده کرد که بطور کامل ریسک سرمایه گذاری را از بین برد.
- **کنترل ورودی و مسیریابی**<sup>۳</sup> مسیریابی در شبکه هایی با شرایط پویا می تواند با Q-Routing به خوبی انجام گیرد که یک روش یادگیری تقویتی توزیع شده است که از Q-Learning ناشی میشود. توسعه این الگوریتم در سالهای ۱۹۹۶ تا ۱۹۹۸ بود. مسئله ای که برای تهیه کنندگان ارتباط راه دور مهم است تصمیم گیری روی این است که کدام نوع از فراخوانی ها پذیرفته شوند و کدام یک برگشت داده شوند که دیدگاه یادگیری تقویتی در سال ۲۰۰۰ توانست ای مسئله را حل کند.

---

<sup>1</sup> - Robot navigation

<sup>2</sup> -Scheduling

<sup>3</sup> -Routing/Call admission control

## **فصل دوم**

# **یادگیری بیزی**

## یادگیری بیزی

### ۱-۲ : مقدمه

- استدلال بیزی روشی بر پایه احتمالات برای استنتاج کردن است
- اساس این روش بر این اصل استوار است که برای هر کمیتی یک توزیع احتمال وجود دارد که با مشاهده یک داده جدید و استدلال در مورد توزیع احتمال آن میتوان تصمیمات بهینه ای اتخاذ کرد.
- در برخی کاربردها (نظیر دسته بندی متن) استفاده از روشهای یادگیری بیزی (نظیر دسته بندی کننده بیزی ساده) توانسته است راه حلهای عملی مفیدی را ارائه کند
- مطالعه یادگیری بیزی به فهم سایر روشهای یادگیری که بطور مستقیم از احتمالات استفاده نمیکند کمک میکند.

### ۲-۲ : نگرش بیزی به یادگیری ماشین

نگرش بیزی به یادگیری ماشین ( و یا هر فرایند دیگر) بصورت زیر است:

دانش موجود در باره موضوع را بصورت احتمالاتی فرموله میکنیم

برای اینکار مقادیر کیفی دانش را بصورت توزیع احتمال، فرضیات استقلال و غیره مدل مینمائیم. این مدل دارای پارامترهای ناشناخته ای خواهد بود.

برای هر یک از مقادیر ناشناخته، توزیع احتمال اولیه ای در نظر گرفته میشود که بازگو کننده باور ما به مورد احتمال بودن هر یک از این مقادیر بدون دیدن داده است.

داده را جمع آوری مینمائیم

با مشاهده داده ها مقدار توزیع احتمال ثانویه را محاسبه میکنیم

با استفاده از این احتمال ثانویه:

به یک نتیجه گیری در مورد عدم قطعیت میرسیم

با میانگین گیری روی مقادیر احتمال ثانویه پیش بینی انجام میدهم

برای کاهش خطای ثانویه مورد انتظار تصمیم گیری میکنیم

### ۳-۲ : ویژگیهای یادگیری بیزی

مشاهده هر مثال میتواند بصورت جزئی باعث افزایش و یا کاهش احتمال درست بودن یک فرضیه گردد.

برای بدست آوردن احتمال یک فرضیه میتوان دانش قبلی را با مثال مشاهده شده ترکیب کرد. این دانش قبلی به دو طریق بدست میآید:

۱. احتمال قبلی برای هر فرضیه موجود باشد
۲. برای داده مشاهده شده توزیع احتمال هر فرضیه ممکن موجود باشد

روشهای بیزی فرضیه هائی ارائه میدهند که قادر به پیش بینی احتمالی هستند (مثل بیمار به احتمال 93% بهبود می یابد)

مثالهای جدید را میتوان با ترکیب وزنی چندین فرضیه دسته بندی نمود.

حتی در مواردی که روشهای بیزی قابل محاسبه نباشند، میتوان از آنها به عنوان معیاری برای ارزیابی روشهای دیگر استفاده کرد

## ۲-۴: ایراد روش بیزی

نیاز به دانش اولیه در مورد تعداد زیادی مقادیر احتمال دارد. وقتی که این اطلاعات موجود نباشند اغلب ناگزیر به تخمین زدن آن هستیم. برای این کار از اطلاعات زمینه، داده هائیکه قبلا جمع آوری شده اند، و فرضیاتی در مورد توزیع احتمال استفاده میشود.

محاسبه فرضیات بهینه بیزی بسیار هزینه بر است (تعداد فرضیه های کاندید خطی است)

## ۲-۵: روشهای یادگیری بیزی شامل:

### ۲-۵-۱: Optimal classifier

استفاده از این روش برای فضاهای فرضیه های بزرگ غیرعملی است

### ۲-۵-۲: Naive Bayes learning

یک روش یادگیری بسیار عملی روش Naive Bayes learner است. در کاربردهائی نظیر دسته بندی متن و تشخیص پزشکی این روش کارائی قابل مقایسه ای با شبکه های عصبی و درخت تصمیم دارد. این روش در مسایلی کاربرد دارد که:

نمونه  $X$  توسط ترکیب عطفی ویژگیها قابل توصیف بوده و این ویژگیها بصورت شرطی مستقل از یکدیگر باشند.

تابع هدف  $f(x)$  بتواند هر مقداری را از مجموعه محدود  $V$  داشته باشد.

مجموعه مثالهای آموزشی نسبتا زیادی در دست باشد

تابع هدف زیر را در نظر بگیرید  $f: X \rightarrow V$  که در آن هر نمونه  $X$  توسط ویژگی زیر مشخص میشود  $(a_1, \dots, a_n)$

**صورت مسئله:** برای یک نمونه مشاهده شده مقدار تابع هدف یا عبارت دیگر دسته بندی آنرا مشخص کنید.

در روش بیزی برای حل مسئله محتملترین مقدار هدف  $v_{map}$  محاسبه میشود:

$$v_{MAP} = \arg \max_{v_j \in V} P(v_j | a_1, \dots, a_n)$$

این رابطه با استفاده از تئوری بیز بصورت زیر نوشته میشود :

$$\begin{aligned} v_{MAP} &= \arg \max_{v_j \in V} \frac{P(a_1, \dots, a_n | v_j) P(v_j)}{P(a_1, \dots, a_n)} \\ &= \arg \max_{v_j \in V} P(a_1, \dots, a_n | v_j) P(v_j) \end{aligned}$$

در رابطه زیر مقدار  $P(v_j)$  با شمارش دفعاتی که  $v_j$  در مثالهای آموزشی مشاهده شده است محاسبه میشود.

$$v_{MAP} = \arg \max_{v_j \in V} P(v_j | a_1, \dots, a_n)$$

اما محاسبه  $P(a_1, \dots, a_n | v_j)$  چندان عملی نیست مگر اینکه مجموعه داده آموزشی بسیار بسیار بزرگی در دست باشد .

روش یادگیری Naive Bayes Classifier بر پایه این فرض ساده (Naive) عمل میکند که:

مقادیر ویژگیها بصورت شرطی مستقل هستند

در اینصورت برای یک مقدار هدف مشخص احتمال مشاهده ترکیب عطفی  $(a_1, \dots, a_n)$  برابر است با حاصلضرب احتمال تک تک ویژگیها در اینصورت رابطه فوق بصورت زیر در میآید

$$v_{NB} = \arg \max_{v_j \in V} P(v_j) \prod_{i=1}^n P(a_i | v_j)$$

خلاصه:

در روش یادگیری Naive Bayes Classifier مقادیر مختلف  $P(v_j)$  و  $P(a_i | v_j)$  با استفاده از دفعات تکرار آنها تخمین زده میشود

مجموعه این تخمین ها فرضیه ای را تشکیل میدهد که با استفاده از رابطه زیر برای دسته بندی داده جدید بکار میرود:

در این روش هیچگونه عمل جستجوی آشکاری وجود ندارد.

## ۲-۵-۳ : Bayesian belief network learning

شبکه های باور بیزی یا Bayesian Belief Networks که Bayes Nets هم نامیده میشود روشی

است برای توصیف توزیع احتمال توام با مجموعه ای از متغیرها.

BBN استقلال شرطی زیر مجموعه ای از متغیرها را قابل توصیف کرده و امکان ترکیب دانش قبلی درباره وابستگی متغیرها را با داده های آموزشی فراهم میآورد



# خلاصه:

یادگیری تقویتی که یکی از پرکاربردترین زمینه های تحقیقاتی در زمینه یادگیری ماشین است یادگیری تقویتی دو ریشه اصلی دارد :

۱ - یادگیری با سعی و خطا که در سال 1911 مطرح شد.

۲ - کنترل بهینه که در اواخر سال 1950 مطرح شد.

میگیرد تا یاد و کرده تعامل محیط با خطا و سعی طریق از که هستیم روبرو عاملی با تقویتی یادگیری مسئله یک در نماید انتخاب هدف به برای رسیدن را بهینه عملی

دو استراتژی اصلی در یادگیری تقویتی به کار میروند:

یکی استفاده از الگوریتم های ژنتیکی

و دیگری استفاده از روشهای آماری و dynamic programming در یادگیری تقویتی روش دوم مد نظر است.

در واقع Dynamic programming روشی است که برای حل یک مسئله از آخرین حالت ممکن شروع کرده و آنچه را که در آن حالت امکان پذیر است را بررسی مینماید، سپس با استفاده از اطلاعات بدست آمده از فرض بودن در آخرین حالت به حل حالت ماقبل آخر میپردازد و اینکار برای حالت های قبل از آن ادامه می یابد.

اجزای اصلی یادگیری تقویتی:

۱ - عامل

۲ - محیط

۳ - سیاست

۴ - تابع پاداش

۵ - تابع برگشت

۶ - تابع ارزش و مدل

از کاربردهای یادگیری تقویتی نیز میتوانیم به، سامانه های چند عامله، بازیها، هدایت رباتها و... اشاره کرد.

استدلال بیزی روشی بر پایه احتمالات برای استنتاج کردن است

اساس روش بیزی بر این اصل استوار است که برای هر کمیتی یک توزیع احتمال وجود دارد

برای بدست آوردن احتمال یک فرضیه میتوان دانش قبلی را با مثال مشاهده شده ترکیب کرد. این دانش قبلی به دو طریق بدست میاید:

۳. احتمال قبلی برای هر فرضیه موجود باشد

۴. برای داده مشاهده شده توزیع احتمال هر فرضیه ممکن موجود باشد

محاسبه فرضیات بهینه بیزی بسیار هزینه بر است.

روشهای یادگیری بیزی شامل سه روش:

- Optimal classifier
- Naive Bayes learning
- Bayesian belief network learning

## فهرست منابع

- [1]- <http://ceit.aut.ac.ir/~shiry/lecture/machine-learning>
- [2]- <http://fa.wikipedia.org>
- [3]- <http://artificial.ir>
- [4]- Tom M.Mitchell , "Machine Learning" , Publisher: "MC Graw – Hill science/Engineering/Match;(march 1,1997)" , ISBN:0070428077