

فصل دوم: یادگیری مفهوم و ترتیب کل به جزء

مسئله‌ی استقرای توابع کلی با داشتن تعدادی نمونه خاص، هدف اصلی یادگیری است. در این فصل به یادگیری مفهوم^۱ (پی بردن به تعریف یک رسته^۲ از اشیا یا اتفاقات با داشتن تعداد محدودی نمونه مثبت و منفی) می‌پردازیم. یادگیری مفهوم را می‌توان جستجو میان فرضیه‌های از پیش تعریف شده برای پیدا کردن مطابق‌ترین فرضیه با نمونه‌ها دانست. در بسیاری موارد این جستجو را می‌توان با بهره‌گیری از خاصیتی ذاتی در فضای فرضیه‌ای (ترتیب جز به کل فرضیه‌ها) سازمان دهی کرد. این فصل الگوریتم‌های یادگیری بسیاری را معرفی خواهد کرد و همچنین شرایطی را که این جواب این الگوریتم‌ها به فرضیه‌ی درست میل می‌کنند را بررسی می‌کند. در ادامه به یادگیری استقرایی می‌پردازیم و توجیه اینکه چگونه ممکن است برنامه‌ها بتوانند داده‌ها را برای نمونه‌های دیگر تعمیم دهند را بررسی خواهیم کرد.

۲.۱ مقدمه

قسمت عمده‌ای از یادگیری منوط به یادگیری مفهومی کلی از روی نمونه‌های آموزشی محدود است. مردم، به عنوان مثال، مفاهیم کلی‌ای مثل "پرنده"، "ماشین" و "وضعیتی که برای قبولی نیاز به بیشتر درس خواندن دارم" و ... را یاد می‌گیرند. هر مفهوم را می‌توان به عنوان زیر مجموعه‌ای از یک مفهوم کلی‌تر از اشیا یا اتفاقات در نظر گرفت. (برای مثال، مجموعه‌ی پرندگان زیر مجموعه‌ی حیوانات قرار می‌گیرد). همچنین، هر مفهوم را می‌توان به عنوان تابعی منطقی مقدار بر روی مجموعه‌ی بزرگ‌تر در نظر گرفت (برای مثال، روی مجموعه‌ی حیوانات مقادیر تابع برای پرندگان درست و برای دیگر حیوانات غلط است).

در این فصل، به استنتاج تعریف کلی یک مفهوم با استفاده از نمونه‌های موجود (که بعضی عضو مفهومند و بعضی دیگر عضو مفهوم نیستند) می‌پردازیم. به این کار در حالت کلی **یادگیری مفهوم** یا تخمین تابع منطقی از نمونه‌ها می‌گویند.

¹ concept

² category

یادگیری مفهوم: استنتاج مقادیر تابع منطقی با یادگیری از نمونه‌هایی از ورودی و خروجی تابع.

۲.۲ کار یادگیری مفهوم

برای درک بهتر، مفهوم "روزهایی که الدو از آبتنی لذت می‌برد" را در نظر بگیرید. جدول ۲.۱ چند روز مختلف را با ویژگی‌هایشان^۱ نشان می‌دهد. ویژگی EnjoySport اینکه الدو از آبتنی در آن روز از آبتنی لذت برده یا خیر را مشخص می‌کند. هدف یادگیری پیش‌بینی مقدار ویژگی EnjoySport با دانستن دیگر ویژگی‌ها یک روز است.

در این یادگیری یادگیر از چه نمایشی برای فرضیه‌ها باید استفاده کند؟ بیاپید برای فرضیه‌ها یک نمایش ساده در نظر بگیریم؛ فرض می‌کنیم که هر فرضیه عطفی از قید روی چندین ویژگی از ویژگی‌های موجود باشد. به طور دقیق‌تر ویژگی‌ها را به صورت شش تایی مرتبی در نظر می‌گیریم که مقادیر شش ویژگی ما را معلوم کنند. این شش ویژگی به ترتیب: Water, Wind, Humidity, AirTemp, Sky و Forecast هستند. هر یک ویژگی‌ها می‌توانند یکی از حالات ممکن زیر را داشته باشد:

مقدار "؟": یعنی هر مقداری را می‌تواند داشته باشد

مقدار مشخص خاصی داشته باشد (برای مثال: AirTemp ممکن است Warm باشد)

مقدار "0": یعنی هیچ مقداری برای این ویژگی قابل قبول نیست.

اگر نمونه x تمام قیود فرضیه‌ی h را تأمین کند، می‌توان گفت h, x را به عنوان یک نمونه‌ی مثبت دسته بندی می‌کند ($h(x) = 1$). برای درک بهتر، فرض اینکه الدو فقط در روزهای Cold و مرطوب از آبتنی لذت می‌برد (مجزا از اینکه بقیه ویژگی‌ها چه باشند) به صورت زیر نمایش داده می‌شود:

<?,Cold,High,?,?,?>

شماره‌ی نمونه	Sky	AirTemp	Humidity	Wind	Water	Forecast	EnjoySport
۱	Sunny	Warm	Normal	Strong	Warm	Same	Yes
۲	Sunny	Warm	High	Strong	Warm	Same	Yes
۳	Rainy	Cold	High	Strong	Warm	Change	No
۴	Sunny	Warm	High	Strong	Cool	Change	Yes

جدول ۲.۱ نمونه‌های مثبت و منفی یادگیری مفهوم EnjoySport

کلی‌ترین فرضیه، اینکه وی هر روز از آبتنی لذت می‌برد، به صورت زیر نمایش داده می‌شود:

<?,?,?,?,?,?>

¹ attribute

و در نقطه‌ی مقابل فرضیه‌ی اینکه در هیچ روزی مقدار EnjoySport بله نیست به صورت زیر نمایش داده می‌شود:

$\langle 0,0,0,0,0,0 \rangle$

به طور خلاصه، عمل یادگیری مفهوم EnjoySport پیدا کردن و توصیف روزهایی (در قالب گفته شده) که $\text{EnjoySport} = \text{Yes}$ است. در کل، هر گونه مسئله‌ی یادگیری مفهوم را می‌توان با نمونه‌هایی که تابع هدف بر روی آن‌ها تعریف شده است، تابع هدف، دسته‌ای از فرضیه‌های موجود که یادگیر^۱ در نظر می‌گیرد و مجموعه‌ی نمونه‌های آموزشی موجود مشخص کرد. مسئله‌ی کلی یادگیری مفهوم EnjoySport با این شکل کلی مذکور در جدول ۲.۲ آمده است.

۲.۲.۱ نماد گذاری

در تمام کتاب، زمانی که بحث، مسایل یادگیری مفهوم است، از نماد گذاری‌ای که توضیح داده می‌شود استفاده خواهد شد. دسته اجسامی که مفهوم روی آن تعریف می‌شود "مجموعه‌ی نمونه‌ها"^۲ نامیده می‌شود و با نماد X مشخص می‌شود. در مثال مذکور، X تمام روزهای ممکن است با ویژگی‌های Sky, AirTemp, Humidity, Wind, Water, Forecast مشخص می‌شود. مفهومی که به دنبال یادگیری آن هستیم "مفهوم هدف"^۳ نامیده و با C نمایش داده می‌شود. در کل، C هر مقدار منطقی است که X به عنوان خروجی می‌دهد و به زبان ریاضی داریم: $C: X \rightarrow \{0,1\}$. در مثال مذکور مقدار C همان مقدار EnjoySport است ($c(x)=1$) اگر مقدار $\text{EnjoySport}=\text{Yes}$ باشد و $c(x)=0$ اگر مقدار $\text{EnjoySport}=\text{No}$ باشد).

• معلومات:

- نمونه‌های X : تمامی حالت‌های روزهای ممکن
 - Sky (مقادیر ممکن: Sunny, Cloudy و Rainy)
 - AirTemp (مقادیر ممکن: Warm و Cold)
 - Humidity (مقادیر ممکن: Normal و High)
 - Wind (مقادیر ممکن: Strong و Weak)
 - Water (مقادیر ممکن: Warm و Cool)
 - Forecast (مقادیر ممکن: Same و Change)
- مجموعه فرضیه‌های H : هر فرضیه با یک شش تایی مرتب ازم تغییرهای Sky, AirTemp, Humidity, Wind, Water Forecast توصیف می‌شود. مقدار هر متغیر می‌تواند "?" (هر مقدار ممکن)، "0" (مقداری ممکن نیست) و یا یک مقدار خاص باشد.
- مفهوم هدف: $C: X \rightarrow \{0,1\}$: EnjoySport
- نمونه‌های یادگیری: نمونه‌های مثبت و منفی تابع هدف. (جدول ۲.۱)

¹ learner

² set of instances

³ target concept

○ فرضیه‌ی h عضوی از H است اگر که برای تمامی x ما داشته باشیم $h(x)=c(x)$

جدول ۲.۲ کار یادگیری مفهوم EnjoySport

در هنگام یادگیری، به یادگیر مجموعه‌ای از نمونه‌های آموزشی با مقدار تابع هدفشان ارائه می‌شود (x و $c(x)$) (جدول ۲.۱) که x ها عضو X هستند. نمونه‌هایی که در آن‌ها مقدار $c(x)=1$ نمونه مثبت^۱ یا عضو مفهوم هدف^۲ نامیده می‌شود. در مقابل، نمونه‌هایی که در آن‌ها مقدار $c(x)=0$ نمونه منفی^۳ یا غیر عضو مفهوم هدف^۴ نامیده می‌شود. گاهی برای راحتی کار از زوج مرتب $\langle x, c(x) \rangle$ برای نمایش نمونه‌های آموزشی استفاده می‌شود. مجموعه‌ی نمونه‌های آموزشی را با حرف D نشان می‌دهیم.

از یادگیر انتظار می‌رود که با داشتن نمونه‌هایی از عملکرد C که آن‌را فرضیه‌سازی^۵ کند یا تخمین بزند. مجموعه‌ی تمامی فرضیه‌های ممکن را با حرف H نشان می‌دهیم. H معمولاً توسط کاربر انسانی و در انتخاب نوع نمایش فرضیه تعیین می‌گردد. در کل، هر فرضیه‌ی h در H یک تابع منطقی مقدار است که $h: X \rightarrow \{0,1\}$. هدف یادگیر پیدا کردن h ای است که برای تمام مقادیر x در X ، $h(x)=c(x)$.

۲.۲.۲ یادگیری استقرایی فرضیه

باید توجه داشت که کار یادگیری پیدا کردن فرضیه h ای است که برای تمامی x های X مشابه مفهوم هدف کار کند در حالی که تنها اطلاعات موجود در مورد C فقط تعداد محدودی نمونه است که در اختیار یادگیر قرار گرفته می‌شود. بنابراین استفاده از الگوریتم‌های یادگیری استقرایی حداکثر تضمین می‌کنند که در نمونه‌های آموزشی مقدار فرضیه با مقدار تابع هدف یکی است. کمبود اطلاعات باعث می‌شود که فرض کنیم بهترین فرضیه همان فرضیه است که به بهترین شکل با نمونه‌های موجود مطابقت دارد^۶. این فرض، فرض اساسی یادگیری استقرایی است، در ادامه کتاب مفصلاً درباره‌ی این فرض بحث خواهیم کرد. فعلاً در این قسمت به طور غیر رسمی این فرض را تعریف می‌کنیم اما در فصل‌های ۵، ۶ و ۷ رسمی‌تر این فرض را بررسی می‌کنیم.

یادگیری استقرایی فرضیه: هر فرضیه‌ای که در مجموعه‌ای به اندازه‌ی کافی بزرگ از نمونه‌های آموزشی تابع هدف را خوب تخمین بزند می‌تواند در نقاط دیگر نیز تابع هدف را خوب تخمین می‌زند.

۲.۳ یادگیری مفهوم با دید جستجو

از نظری می‌توان یادگیری مفهوم را جستجویی بین تمام فرضیه‌های موجود (H) دانست. هدف از این جستجو پیدا کردن فرضیه‌ای است که به بهترین وجه ممکن رفتار تابع هدف را در نمونه‌های موجود تخمین بزند. مهم است که بدانیم که طراح با انتخاب نوع نمایش فرضیه، تمام فرضیه‌های موجود (H)، فرضیه‌هایی را که الگوریتم می‌تواند نمایش بدهد و در کل یاد بگیرد را نیز محدود می‌کند. دوباره به مثال EnjoySport برمی‌گردیم، تمام فرضیه‌های موجود H و دسته نمونه‌های ممکن X را در نظر بگیرید. Sky^۳ حالت ممکن و بقیه ویژگی‌ها

¹ positive example

² member of target concept

³ negative example

⁴ nonmember of target concept

⁵ hypothesize

⁶ best fit

هر کدام دو حالت ممکن دارند، پس در کل تعداد حالات ممکن برای X ، $3 \times 2 \times 2 \times 2 \times 2 \times 2 = 96$ خواهد بود. با محاسباتی ساده می‌توان (با در نظر گرفتن "?" و "0") گفت که کل تعداد فرضیات ممکن $5 \times 4 \times 4 \times 4 \times 4 \times 4 = 5120$ خواهد بود. اما می‌دانیم فرضیاتی که یک یا چند "0" دارند منتفی هستند چون همه‌ی مقادیر را 0 پیش بینی می‌کنند ("0" بودن یک مقدار به معنی این است که در هیچ حالتی از این ویژگی مقدار تابع ۱ نمی‌شود). پس تعداد واقعی کل فرضیات برابر است با $1 + (4 \times 3 \times 3 \times 3 \times 3 \times 3) = 973$. توجه داشته باشید که مثال EnjoySport، یک مثال بسیار ساده از یادگیری مفهوم است و متناسباً تعداد کل فرضیات ممکن کمی نیز دارد در حالی که در اکثر مثال‌های واقعی این تعداد بسیار زیاد و گاهی اوقات نیز نامتناهی است.

با دید جستجو به مسئله، طبیعی است که مطالعه بر روی الگوریتم‌های یادگیری مفهوم به مطالعه بر روی الگوریتم‌های جستجو بر روی فرضیه‌ها تبدیل شود. علاقه‌ی ما به الگوریتم‌هایی خواهد بود که به طور موثر و سریع بتوانند تعداد زیاد و حتی نامتناهی از فرضیه‌ها را بررسی کنند تا بهترین فرضیه را برای نمونه‌های موجود پیدا کنند.

۲.۳.۱ ترتیب کل به جزء فرضیه‌ها

بسیاری از الگوریتم‌های یادگیری مفهوم جستجوی بین فرضیه‌ها را با یک ساختار مفید سازمان دهی می‌کنند. این سازمان دهی برای تمام مسایل یادگیری مفهوم به کار می‌رود: ترتیب کل به جزء فرضیه‌ها. با استفاده از این ساختار طبیعی می‌توان الگوریتم‌هایی طراحی کرد که بدون بررسی تک تک فرضیه‌ها می‌توانند تمام فرضیه‌ها را بررسی کنند. با این نوع الگوریتم‌ها می‌توان حتی هنگامی که اندازه‌ی H نامتناهی است به جواب رسید. برای مثال دو فرضیه زیر را در نظر بگیرید:

$$h_1 = \langle \text{Sunny}, ?, ?, \text{Strong}, ?, ? \rangle$$

$$h_2 = \langle \text{Sunny}, ?, ?, ?, ?, ? \rangle$$

حال نمونه‌های درون این دو فرضیه در نظر بگیرید. چون h_2 قیود کمتری دارد پس تعداد بیشتری نمونه درون آن وجود خواهد داشت در واقع هر نمونه‌ای که در h_1 باشد در h_2 نیز هست. بنابراین می‌گوییم h_2 از h_1 کلی‌تر^۱ است.

رابطه‌ی ذاتی "کلی‌تر یا مساوی بودن" بین فرضیه‌ها را می‌توان به صورت دقیق‌تر نیز تعریف کرد. اول، برای هر نمونه x در X و هر فرضیه‌ی h می‌گوییم x را h راضی می‌کند اگر و فقط اگر $h(x)=1$. حال رابطه‌ی "کلی‌تر یا مساوی بودن" را بر اساس نمونه‌هایی که فرضیه‌ها را راضی می‌کنند تعریف می‌کنیم. برای دو فرضیه‌ی h_j و h_k داریم: h_j کلی‌تر از h_k است اگر و فقط اگر هر نمونه‌ای که h_k را راضی کرد h_j را نیز راضی کند.

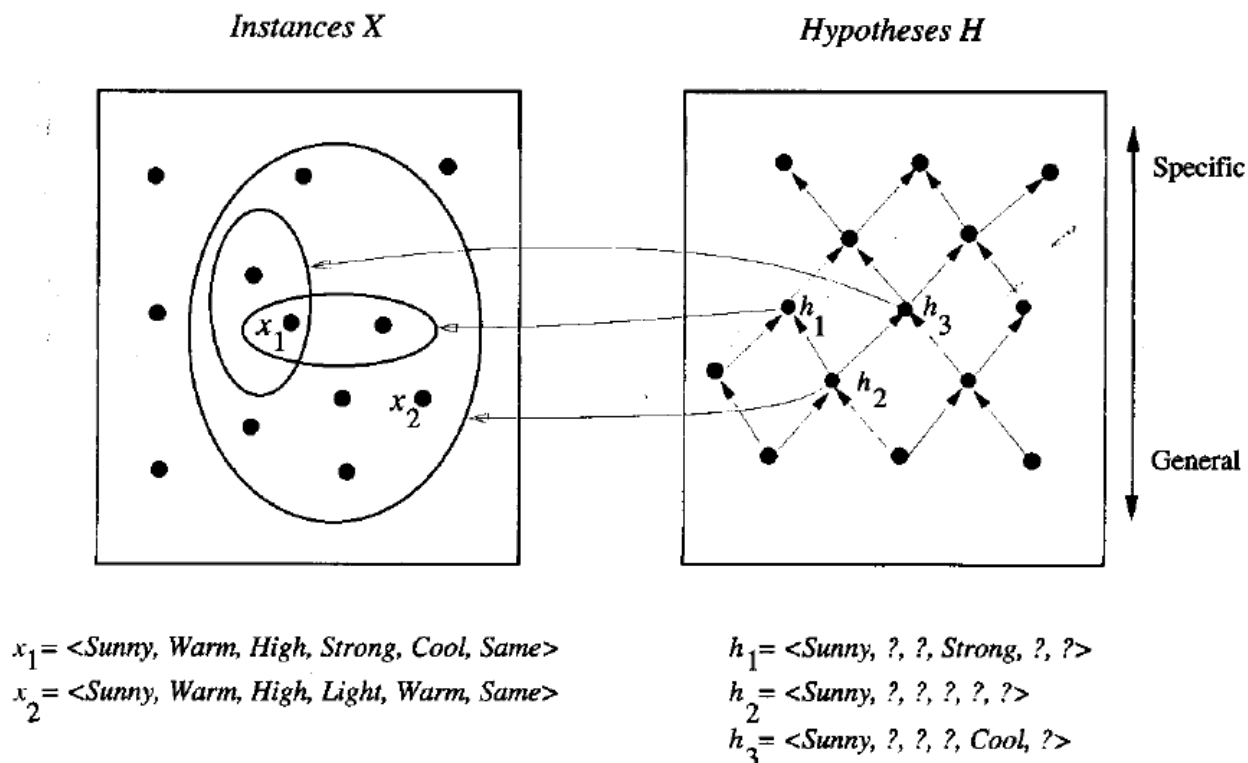
تعریف: اگر h_j و h_k دو تابع منطقی مقدار تعریف شده روی X باشند h_j کلی‌تر یا مساوی است با h_k اگر و فقط اگر

$$(h_j \geq_g h_k)$$

$$(\forall x \in X)[(h_k(x) = 1) \rightarrow (h_j(x) = 1)]$$

¹ more general

بعضی مواقع لازم است فرضیه ای به طور اکید از فرضیه دیگر کلی تر باشد. فرضیه h_j (اکیداً) کلی تر است از h_k اگر و فقط اگر $(h_j \geq_g h_k) \wedge (h_k \not\geq_g h_j)$. بعضی مواقع نیز لازم است برعکس بالا بگوییم که h_k خاص تر^۱ از h_j است اگر و تنها اگر $(h_j >_g h_k)$.



شکل ۲.۱ نمونه‌ها و فرضیه‌ها و رابطه‌ی کلی‌تری.

مربع سمت چپ نشان دهنده‌ی X یا همان تمامی نمونه‌هاست. و مربع سمت راست نشان دهنده‌ی H یا همان تمامی فرضیه‌هاست. هر فرضیه متناسب با زیر مجموعه‌ای از X است (همان زیر مجموعه‌ای که آن را راضی می‌کند). فلش‌های بین فرضیه‌ها رابطه‌ی خاص‌تر بودن را نشان می‌دهد (سراشتهایی فلش‌ها خاص‌ترند). توجه داشته باشید که مجموعه‌ی متناسب با h_2 مجموعه‌ی متناسب با h_1 را شامل می‌شود پس فرضیه‌ی h_2 از فرضیه‌ی h_1 کلی‌تر است.

برای درک بهتر، سه فرضیه‌ی h_1 ، h_2 و h_3 را در همان مثال EnjoySport در نظر بگیرید (شکل ۲.۱). رابطه‌ی \geq_g بین این سه فرضیه چگونه است؟ همان طور که قبلاً نیز گفته شد h_2 از h_1 کلی‌تر است زیرا هر نمونه‌ای که h_2 را راضی کند h_1 را نیز راضی می‌کند. به طور مشابه h_2 از h_3 نیز کلی‌تر است. توجه داشته باشید که هیچ کدام از فرضیه‌های h_1 و h_3 کلی‌تر از دیگری نیست. با وجود اینکه در نمونه‌هایی اشتراک دارند اما هیچ کدام دیگری را شامل نمی‌شود. توجه داشته باشید که دو رابطه‌ی \geq_g و $>_g$ مستقل از اینکه مفهوم هدف چه باشد تعریف شده‌اند و فقط بر اساس اینکه کدام نمونه‌ها در درون فرضیه قرار می‌گیرند تعریف شده‌اند و نه بر اساس تابع هدف. به طور رسمی، رابطه‌ی \geq_g ترتیب خاصی را در درون فضای فرضیه‌ها H ایجاد می‌کند (این رابطه، بازتابی، پاد متقارن و انتقالی است). به طور غیر رسمی،

¹ more specific

زمانی که می‌گوییم یک ساختار جزئی مرتب^۱ (در مقابل کلی^۲) است، منظورمان این است که ممکن است جفت فرضیه‌هایی مثل h_1 و h_3 وجود داشته باشند که $h_1 \not\geq_g h_3$ و $h_3 \not\geq_g h_1$.

اهمیت رابطه‌ی \geq_g در این است که ساختار مفیدی برای هر مسئله‌ی یادگیری مفهوم بر روی فضای فرضیه‌ها (H) ایجاد می‌کند. قسمت بعدی به الگوریتمی که با استفاده از این ساختار جستجو را سازمان دهی می‌کند، می‌پردازد.

۱. h را خاص‌ترین فرضیه‌ی H در نظر بگیر

۲. برای هر نمونه مثبت x

• برای هر ویژگی a_i در h

اگر x ، a_i را راضی می‌کند کاری انجام نده

در غیر این صورت در h از خاصیت a_i به سمت کلی‌تر شدن برو (قید کلی‌تری که نمونه را شامل می‌شود را در این ویژگی جایگزین کن).

۳. فرضیه h را خروجی بده.

جدول ۲.۳ الگوریتم FIND-S

۲.۴ FIND-S : پیدا کردن خاص‌ترین فرضیه

چگونه می‌توان از رابطه‌ی کلی‌تری برای سازمان دهی جستجوی بین فرضیه‌ها استفاده کرد؟ یک راه شروع کردن از خاص‌ترین فرضیه درون فضای فرضیه‌ها H و کلی‌تر کردن آن در مراحل که نمی‌تواند نمونه‌ها را بپوشاند است. (زمانی که یک فرضیه یک نمونه مثبت را می‌پوشاند که آن را شامل شود). برای بهتر روشن شدن این مطلب الگوریتم FIND-S (جدول ۲.۳) در نظر بگیرید.

برای تصور بهتر فرض کنیم که به یادگیر مقادیر جدول ۲.۱ داده شده تا مفهوم EnjoySport را یاد بگیرد. گام اول الگوریتم مقدار دهی اولیه‌ی h با خاص‌ترین فرضیه است.

$h \leftarrow \langle 0,0,0,0,0,0 \rangle$

بعد به سراغ اولین داده‌ی جدول ۲.۱ می‌رود، چون این داده مقدار مثبت این فرضیه نیست معلوم می‌شود که فرضیه بیش از حد خاص است. چون نمونه هیچ یک از مقادیر "0" فرضیه را راضی نمی‌کند پس این مقادیر با مقادیر کلی‌تری جایگزین می‌شوند.

$h \leftarrow \langle \text{Same}, \text{Warm}, \text{High}, \text{Normal}, \text{Warm}, \text{Sunny} \rangle$

¹ partial order

² total

اما با این حال این h بیش از حد خاص است زیرا جز به همان نمونه اول، نمونه مثبت دیگری ندارد. زمانی که به نمونه آموزشی دوم می‌رسد (که در اینجا یک نمونه مثبت است)، دوباره الگوریتم مجبور می‌شود که h را کلی‌تر کند و ویژگی سوم را "?" قرار دهد (ویژگی‌ای که در این نمونه پوشانده نشده بود) پس:

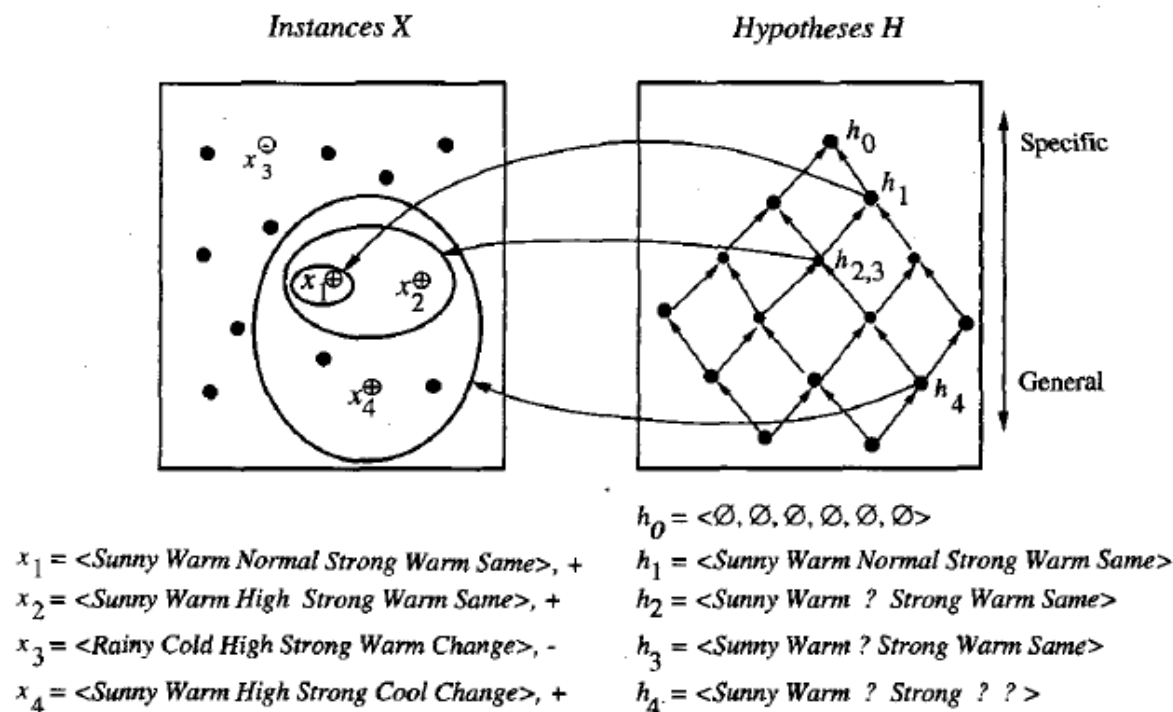
$h \leftarrow \langle \text{Same}, \text{Warm}, \text{High}, ?, \text{Warm}, \text{Sunny} \rangle$

با رسیدن به نمونه سوم (که یک نمونه منفی است) نیاز نیست که تغییر خاصی به h بدهد. در واقع الگوریتم FIND-S هیچ کاری در قبال نمونه‌های منفی نمی‌کند! با وجود اینکه این رفتار الگوریتم کمی عجیب به نظر می‌آید توجه داشته باشید که الآن این نمونه از h جواب منفی می‌گیرد (h این نمونه را درست دسته‌بندی کرده) پس به هر حال تغییری لازم نیست. در حالت کلی، تا زمانی که فرض کنیم در فضای فرضیه‌ها (H)، فرضیه‌ای وجود دارد که C را به طور کامل توصیف می‌کند و نمونه‌های آموزشی درست هستند نیازی به تغییر در نمونه‌های منفی نیست. دلیل این امر آن است که فرضیه‌ی h خاص‌ترین فرضیه‌ی ممکن در H است که بر اساس نمونه‌های مثبت ساخته شده، و چون فرض می‌کنیم که C نیز در H وجود دارد و نمونه‌های مثبت را نیز در بر دارد پس مطمئناً C کلی‌تر یا مساوی h است. و چون C هیچ نمونه‌ی منفی‌ای را شامل نمی‌شود، h نیز آن را شامل نخواهد شد. پس هیچ نیازی به تغییر در نمونه‌های منفی نداریم.

برای کامل کردن الگوریتم FIND-S چهارمین داده را نیز بررسی می‌کنیم و داریم که:

$h \leftarrow \{ "?", "?", \text{High}, "?", \text{Warm}, \text{Sunny} \}$

الگوریتم FIND-S نمونه‌ای از استفاده‌ی از ویژگی کلی‌تری برای جستجوی فضای فرضیه‌ها در پیدا کردن فرضیه مطلوب است. جستجو از فرضیه‌ای خیلی خاص شروع و با کلی‌تر کردن فرضیه‌ها در یک زنجیره‌ی کلی‌تر شدن ادامه پیدا می‌کند. شکل ۲.۲ این جستجو را در درون فضای فرضیه‌ای و فضای نمونه‌ها نشان می‌دهد. در هر مرحله فقط به اندازه‌ی لازم برای پوشش نمونه جدید فرضیه کلی‌تر می‌شود. پس در هر مرحله فرضیه‌ی h خاص‌ترین فرضیه‌ی ساخته شده بر روی نمونه‌های مثبت قبلی است. (S در نام FIND-S نیز از همان کلمه‌ی specific گرفته شده). تحقیقات یادگیری مفهوم پر از الگوریتم‌هایی مشابهی است که ترتیب کلی‌تری را به سبکی برای سازمان دهی جستجو استفاده کرده‌اند. تعدادی از این الگوریتم‌ها را در این فصل و تعدادی دیگر را در فصل ۱۰ بررسی می‌کنیم.



شکل ۲.۲ فرضیه‌هایی که در طی مراحل الگوریتم FIND-S به دست آمد.

جستجو از فرضیه‌ی h_0 شروع می‌شود که خاص‌ترین فرضیه در H است سپس پله به پله تحت تاثیر نمونه‌ها کلی‌تر می‌شود (از h_1 تا h_4). در نمودار نمونه‌ها (سمت چپ) نمونه‌های مثبت با علامت "+"، نمونه‌های منفی با علامت "-" و نمونه‌هایی که جزو نمونه‌های آموزشی نبوده‌اند با دایره‌های توپر نشان داده شده‌اند.

ویژگی کلیدی الگوریتم FIND-S این است که در میان فضای فرضیه‌ها (H) تضمین می‌کند که خاص‌ترین فرضیه را بر اساس نمونه‌های مثبت ارائه دهد. با فرض اینکه نمونه‌ها درست باشند و C نیز در H موجود باشد، خروجی الگوریتم FIND-S برای نمونه‌های منفی مقدار صفر می‌دهد. با این وجود تعدادی از سؤالات موجود بی‌جواب می‌مانند:

- آیا یادگیر به سمت مفهوم هدف همگرا شده؟ با وجود اینکه الگوریتم FIND-S فرضیه‌ای را پیدا می‌کند که با تمام نمونه‌های آموزشی مطابقت داشته باشد، اما تضمین نمی‌کند که فرضیه پیدا شده یکتا باشد و ممکن است فرضیه‌های دیگری در H موجود باشند که با نمونه‌ها مطابقت داشته باشند. ترجیح ما بر این است که از الگوریتم‌هایی استفاده کنیم که مشخص کنند آیا به فرضیه مشخصی همگرا شده‌اند و اگر نه، مشخص کنند که میزان عدم قطعیت چقدر و چگونه است.
- چرا دنبال خاص‌ترین فرضیه هستیم؟ زمانی که به الگوریتم FIND-S نمونه‌هایی داده می‌شود خروجی خاص‌ترین فرضیه‌ی ممکن خواهد بود. معلوم نیست که چرا دنبال کلی‌ترین فرضیه یا چیزی بینابین نمی‌گردیم و فقط دنبال خاص‌ترین فرضیه هستیم.
- آیا همیشه نمونه‌های یادگیری بدون خطا هستند؟ در بسیاری از مسایل یادگیری مفهوم امکان وجود خطا یا نویز^۱ در نمونه‌های یادگیری وجود دارد. نمونه‌هایی که خطا دارند کاملاً FIND-S را به اشتباه می‌اندازند، مخصوصاً اینکه FIND-

¹ noise

S در قبال نمونه های منفی هیچ عکس العملی انجام نمی دهد. ما ترجیح می دهیم از الگوریتم هایی استفاده کنیم که خطا داشتن نمونه ها را تشخیص بدهند و ترجیحاً بتوانند خود را با این خطاها تطبیق دهند.

- اگر خاص ترین فرضیه یکتا نبود چه؟ در مثال EnjoySport همیشه خاص ترین فرضیه ی ساخته شده روی نمونه ها یکتا بود. با این حال، در فضا های فرضیه ای دیگر (که بعداً درباره ی آن ها بحث خواهیم کرد) ممکن است خاص ترین فرضیه یکتا نباشد. در چنین شرایطی، الگوریتم FIND-S باید تصحیح شود تا بتواند گزینه های دیگر موجود در کلی تر سازی و احتمال این را که آیا می شود از شاخه ای دیگر از روند کلی سازی به مفهوم هدف رسید بررسی کند. در آینده، فضا های فرضیه ای را معرفی خواهیم کرد که در آن ها همیشه خاص ترین فرضیه ی موجود یکتا نیست، با این حال این نوع فضا های فرضیه ای بیشتر تئوری اند تا عملی.

۲.۵ فضا های ویژه و الگوریتم Candidate-Elimination

در این بخش به الگوریتم دیگری در یادگیری مفهوم به نام الگوریتم Candidate-Elimination می پردازیم که ضعف های FIND-S را ندارند. توجه داشته باشید که خروجی الگوریتم FIND-S فقط یکی از فرضیه های داخل H است که با نمونه ها مطابقت دارد. نکته ی کلیدی الگوریتم Candidate-Elimination هم همین است. این الگوریتم توصیفی از تمامی فرضیه های مطابق با نمونه ها می دهد. نکته ی جالب تر این است که الگوریتم Candidate-Elimination برای پیدا کردن دسته فرضیه ی مطابق با نمونه ها تمامی فرضیه ها را بررسی نمی کند. این کار بر اساس همان ترتیب کلی تری و با استفاده از توصیفی برای مجموعه ی شامل تمامی فرضیه های سازگار با نمونه ها انجام می گردد.

الگوریتم Candidate-Elimination در گذشته برای پیدا کردن رابطه ی بین طیف سنجی جرمی در شیمی (Mitchell 1979) و در یادگیری قوانین جستجوی اکتشافی^۱ (Mitchell 1983) استفاده شده است. اما با این حال در کاربرد های واقعی، به دلیل اینکه هر دو الگوریتم FIND-S و Candidate-Elimination در مواجهه با نمونه هایی که خطا و نویز دارند عملکرد ضعیفی دارند، کاربرد زیادی ندارند. از آن مهم تر، برای هدف ما، الگوریتم Candidate-Elimination قالبی مفهومی را برای معرفی بسیاری از مطالب پایه ای در یادگیری ماشین معرفی می کند. در ادامه به این الگوریتم و این مطالب خواهیم پرداخت. در فصل های بعدی به این الگوریتم هایی که با داده های نویز دار نیز درست کار می کنند خواهیم پرداخت.

۲.۵.۱ معرفی

الگوریتم Candidate-Elimination تمامی فرضیه های قابل توصیف که در نمونه ها صدق می کنند را پیدا می کند. برای تعریف دقیق الگوریتم، با چند تعریف اولیه شروع می کنیم. اول، یک فرضیه با نمونه های آموزشی سازگار است، اگر آن نمونه ها را به درستی دسته بندی کند.

تعریف: فرضیه ی h با نمونه های D سازگار^۲ است اگر و فقط اگر برای هر زوج مرتب $\langle x, c(x) \rangle$ در D داشته باشیم $h(x) = c(x)$.

$$\text{Consistent}(h, D) \equiv (\forall \langle x, c(x) \rangle \in D) h(x) = c(x)$$

¹ heuristic search

² consistent

توجه داشته باشید راضی کردن و سازگاری یکی نیستند. برای مثال: x فرضیه‌ی h را راضی می‌کند اگر $h(x)=1$ در حالی که فرقی نمی‌کند که x یک نمونه مثبت یا منفی باشد. در حالی که نمونه x زمانی با h سازگار است که $h(x)=c(x)$ باشد.

الگوریتم Candidate-Elimination تمامی دسته فرضیه‌های سازگار با نمونه‌های آموزشی را خروجی می‌دهد. این دسته فرضیه‌ها فضای ویژه^۱ نامیده می‌شود چون تمامی نسخه‌های قابل قبول مفهوم هدف را شامل می‌شود. فضای ویژه وابسته به فضای فرضیه‌ها (H) و نمونه‌های آموزشی (D) است.

تعریف: فضای ویژه، که با $VS_{H,D}$ نمایش داده می‌شود، با توجه به فضای فرضیه‌ها (H) و نمونه‌های آموزشی (D)، مجموعه‌ی فرضیه‌هایی از H است که با مثال‌های D سازگار است.

$$VS_{D,H} \equiv \{h \in H | \text{Consistent}(h, D)\}$$

۲.۵.۲ الگوریتم List-Then-Eliminate

ساده‌ترین راه ممکن برای معرفی فضای ویژه معرفی تک تک عضوهای آن است. این نوع معرفی به یک الگوریتم به نام List-Then-Eliminate می‌انجامد (جدول ۲.۴).

الگوریتم List-Then-Eliminate ابتدا فرض می‌کند که تمامی فرضیه‌ها سازگار با نمونه‌ها هستند. یعنی فضای ویژه را با H مقدار دهی اولیه می‌کند. سپس هر فرضیه‌ای را که با مثال‌ها سازگاری نداشته باشد حذف می‌کند. با بررسی سازگاری تک تک فرضیه‌ها با تک تک نمونه‌ها، فرضیه‌ها از فضای ویژه حذف می‌شوند و در آخر فقط یک فرضیه در فضای ویژه باقی می‌ماند (که همان مفهوم هدف است). اگر تعداد داده‌ها کافی نباشد، در فضای ویژه بیشتر از یک عضو باقی می‌ماند و آن هم دسته فرضیه‌های سازگار با نمونه‌هاست.

اصولاً، فقط زمانی که H متناهی^۲ است می‌توان از الگوریتم List-Then-Eliminate استفاده کرد. این الگوریتم مزیت‌های بسیاری شامل تضمین اینکه که تمامی فرضیه‌های سازگار با نمونه‌ها را پیدا کند دارد. اما در مقابل بسیار زمان‌گیر است چون باید سازگاری تمامی اعضای H را با تمامی داده‌ها بررسی کرد که جز در فضاهای فرضیه‌ای بسیار ساده شرطی غیر عملی است.

۲.۵.۳ نمایش فشرده تری از فضاهای ویژه

الگوریتم Candidate-Elimination مشابه الگوریتم List-Then-Eliminate عمل می‌کند. با این تفاوت که از نمایشی دیگر برای فضای ویژه استفاده می‌کند. در این نمایش فضای ویژه با کلی‌ترین و خاص‌ترین فرضیه‌هایش نمایش داده می‌شود. این اعضای مرزی فضای ویژه، نشان دهنده‌ی مکان فضای ویژه در ترتیب کلی‌تری هستند.

الگوریتم List-Then-Eliminate

۱. تمامی فرضیه‌های $H \leftarrow \text{VersionSpace}$

۲. برای هر نمونه آموزشی $\langle x, c(x) \rangle$

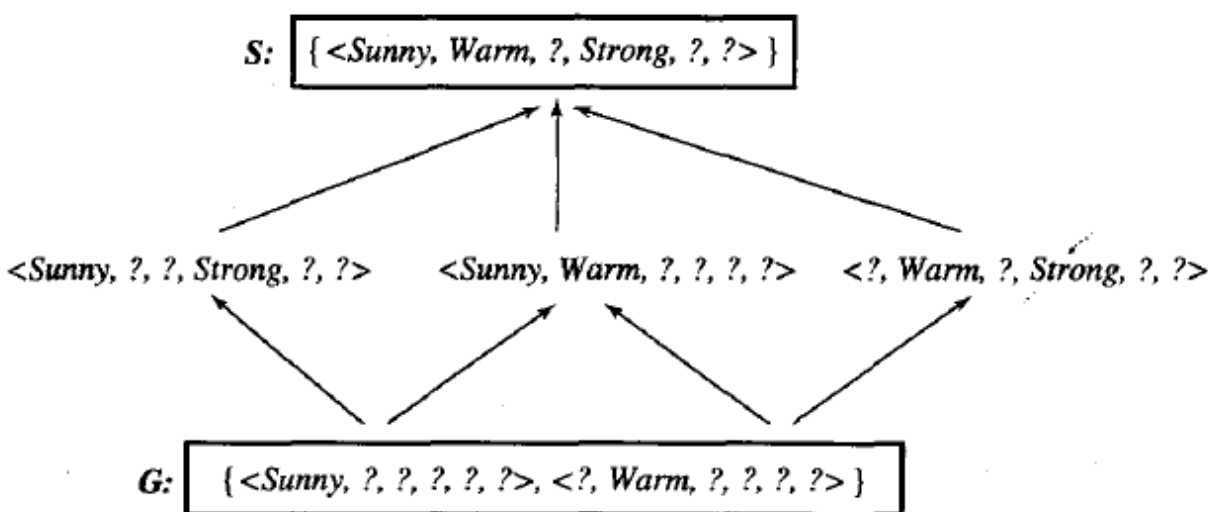
¹ version space

² finite

هر فرضیه ای در VersionSpace مثل h که $h(x) \neq c(x)$ را از VersionSpace حذف کن

۳. لیست باقی مانده در VersionSpace را چاپ کن

جدول ۲.۴ الگوریتم List-Then-Eliminate



شکل ۲.۳ یک فضای ویژه با مرزهای خاص‌ترین و کلی‌ترین فرضیه‌ها.

فضای ویژه‌ی فوق هر چهار فرضیه‌ی نشان داده شده را در بر می‌گیرد اما به طور خیلی ساده تر می‌توان آن را فقط با S و G نشان داده شده در شکل نمایش داد. فلش‌های شکل فرایند خاص‌تر شدن را نشان می‌دهد (پیکان فلش‌ها به سمت فرضیه‌های خاص‌تر است). این فضای ویژه برای مفهوم EnjoySport است و نمونه‌های آموزشی نیز همان نمونه‌های جدول ۲.۱ است.

برای تصور بهتر از این نمایش جدید فضاهای ویژه دوباره به سراغ مسئله‌ی EnjoySport می‌رویم (جدول ۲.۲). الگوریتم FIND-S برای این مسئله خروجی زیر را داده است:

$h = \langle \text{Sunny}, \text{Warm}, ?, \text{Strong}, ?, ? \rangle$

در واقع این فرضیه فقط یکی از ۶ فرضیه‌ی سازگار موجود در H است. تمامی این ۶ فرضیه در شکل ۲.۳ نشان داده شده‌اند. مجموعه‌ی ۶ فرضیه‌ی سازگار با این نمونه‌ها فضای ویژه است. فلش‌های شکل فرایند خاص‌تر شدن را نشان می‌دهد (پیکان فلش‌ها به سمت فرضیه‌های خاص‌تر است). الگوریتم Candidate-Elimination فضای ویژه را با تشخیص مرزهای کلی‌تر (که در شکل با حرف G مشخص شده) و مرزهای خاص‌تر (که در شکل با حرف S مشخص شده) مشخص می‌کند. با داشتن این دو دسته فرضیه می‌توان تمامی فرضیه‌های فضای ویژه را با ترتیب کلی‌تری مشخص کرد.

بدیهی است که می‌توان هر فضای ویژه را با خاص‌ترین و کلی‌ترین عضوهایش مشخص کرد. در ادامه دو مرز کلی‌تر و خاص‌تر را تعریف کرده و ثابت می‌کنیم که می‌توان فقط با استفاده از این دو مرز کل فضای ویژه را مشخص کرد.

تعریف: مرز کلی G ، با توجه به فضای فرضیه‌ها (H) و نمونه‌های آموزشی (D) مجموعه کلی‌ترین فرضیه‌ها در H و سازگار با D است.

$$G \equiv \{g \in H \mid \text{Consistent}(g, D) \wedge (\neg \exists g' \in H)[g' >_g g] \wedge \text{Consistent}(g', D)\}$$

تعریف: مرز خاص S ، با توجه به فضای فرضیه‌ها (H) و نمونه‌های آموزشی (D) مجموعه خاص‌ترین فرضیه‌ها در H و سازگار با D است.

$$S \equiv \{s \in H | \text{Consistent}(s, D) \wedge (\neg \exists s' \in H)[s \geq_g s'] \wedge \text{Consistent}(s', D)\}$$

تا زمانی که دو دسته‌ی G و S خوش تعریف^۱ باشند (تمرین ۲.۷)، فضای ویژه را به درستی و کاملاً مشخص می‌کنند. به عبارت دیگر، می‌توان نشان داد که فضای ویژه اجتماع سه مجموعه‌ی G ، S و مجموعه‌ی بین آن‌ها در ترتیب کلی‌تری است. این اثبات به طور کامل در قضیه‌ی ۲.۱ آمده.

قضیه‌ی ۲.۱. قضیه‌ی نمایش فضای ویژه. اگر X تمام نمونه‌ها، H تمام فرضیه‌های روی X ، $X \rightarrow \{0,1\}$ یک مفهوم هدف دلخواه روی X ، D نمونه‌های آموزشی موجود باشد $\{ \langle x, c(x) \rangle \}$ و S و G خوش تعریف باشند داریم:

$$VS_{H,D} = \{h \in H | (\exists s \in S)(\exists g \in G)(g \geq_g h \geq_g s)\}$$

اثبات. برای اثبات این قضیه کافی است نشان دهیم که (۱) هر h که در قسمت سمت راست تساوی بالا صدق می‌کند عضو $VS_{H,D}$ است و (۲) هر عضو $VS_{H,D}$ در سمت راست تساوی صدق می‌کند. اثبات قسمت (۱):

فرض کنیم g عضوی از G و s عضوی از S و h عضوی از H باشد به صورتی که $g \geq_g h \geq_g s$. از روی تعریف S داریم که، s توسط تمامی نمونه‌های D را راضی می‌شود و طبق فرض $s \geq_g h$ پس h نیز توسط تمامی نمونه‌های D را راضی می‌شود.

به طور مشابه طبق تعریف G ، g توسط هیچ یک از نمونه‌های منفی D را راضی نمی‌شود و طبق فرض $g \geq_g h$ پس h نیز توسط هیچ یک از نمونه‌های منفی D را راضی نمی‌شود.

با توجه به دو قسمت بالا پس h با D سازگار است پس h نیز عضو $VS_{H,D}$ است.

اثبات قسمت دوم کمی پیچیده‌تر است. باید از برهان خلف استفاده کرد و فرض کرد که فرضیه‌ای در مثل h در $VS_{H,D}$ وجود دارد که در قسمت سمت راست تساوی صدق نمی‌کند و به تناقض رسید (تمرین ۲.۶).

۲.۵.۴ الگوریتم یادگیری Candidate-Elimination

الگوریتم یادگیری Candidate-Elimination فضای ویژه‌ای را محاسبه می‌کند که با تمامی نمونه‌های آموزشی موجود سازگار باشد. مثل الگوریتم List-Then-Eliminate در ابتدای این الگوریتم فضای ویژه را کل H در نظر می‌گیریم. پس G را کلی‌ترین فرضیه در نظر می‌گیریم:

$$G_0 \leftarrow \{ \langle ?, ?, ?, ?, ?, ?, ? \rangle \}$$

در مقابل نیز S را خاص‌ترین فرضیه در نظر می‌گیریم:

$$S_0 \leftarrow \{ \langle 0, 0, 0, 0, 0, 0, 0 \rangle \}$$

¹ Well-defined

دو مرز تعیین شده تمامی فرضیه های موجود در H را در بر می گیرند، چون تمامی فرضیه ها از S_0 کلی تر و از G_0 خاص ترند. سپس تک تک نمونه ها بررسی می شوند و S و G ، کلی تر و خاص تر می گردند تا فرضیه های ناسازگار با نمونه ها را از فضای ویژه حذف کنند. بعد از بررسی کل نمونه ها فضای ویژه مشخص می شود. خلاصه ای این الگوریتم در جدول ۲.۵ آمده.

مقدار اولیه ی G را کلی ترین فرضیه در H قرار بده

مقدار اولیه ی S را خاص ترین فرضیه در H قرار بده

برای هر نمونه آموزشی d مراحل زیر را انجام بده

- اگر d نمونه ای مثبت بود
 - هر فرضیه ای که در G با d مطابقت نداشت را حذف کن
 - برای هر فرضیه ی S در S که سازگار با d نیست
 - s را از S حذف کن
 - تمامی خاص ترین کلی سازی h از s را که در شرط زیر صدق می کنند به S اضافه کن
 - h در آن با d سازگار است و حداقل یکی از اعضای G از آن کلی تر است
 - هر فرضیه ای که از فرضیه ی دیگری در S کلی تر بود از آن حذف کن
- اگر d نمونه ای منفی بود
 - هر مثال ناسازگار با d در S را حذف کن
 - برای هر فرضیه ی g در G که با d سازگار نیست
 - g را از G حذف کن
 - تمامی کلی ترین خاص سازی h از g را که در شرط زیر صدق می کنند به G اضافه کن
 - h در آن با d سازگار است و حداقل یکی از اعضای S از آن خاص تر است
 - هر فرضیه ای که از فرضیه ی دیگری در G خاص تر بود از آن حذف کن

جدول ۲.۵ الگوریتم Candidate-Elimination با استفاده از فضای ویژه.

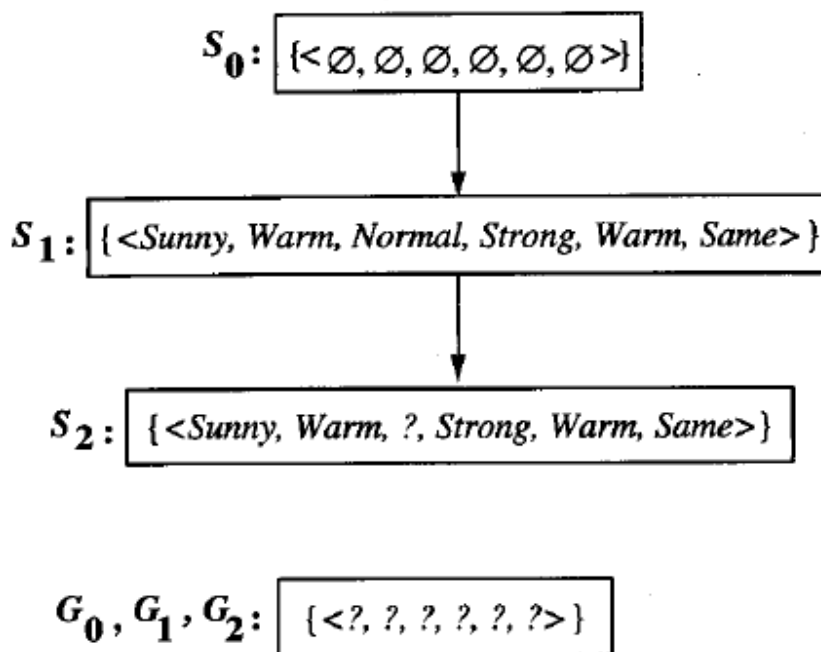
توجه داشته باشید که نمونه های مثبت و منفی به روش های متفاوتی روی S و G اثر می گذارند (به طور قرینه).

توجه داشته باشید که در شرح الگوریتم از عملیات هایی از جمله پیدا کردن خاص ترین کلی سازی، کلی ترین خاص سازی، خاص ترین، کلی ترین استفاده شده است. اطلاعات جزئی تر در مورد این عملیات به نمونه ها و فرضیه ها بستگی دارد. با این وجود، خود الگوریتم را می توان برای هر مسئله ی یادگیری مفهوم و هر فضای فرضیه ای که این عملیات ها را بشود رویش تعریف کرد، به کار برد. در ادامه دوباره همان مسئله ی EnjoySport را این بار از دید این الگوریتم بررسی می کنیم.

۲.۵.۵ یک مثال شهودی

در شکل ۲.۴ حاصل اجرا کردن الگوریتم Candidate-Elimination برای دو مثال اول جدول ۲.۱ آمده است. همان طور که پیش تر نیز اشاره شد، در ابتدا فرضیه های مرزی ابتدایی G_0 و S_0 به ترتیب کلی ترین و خاص ترین فرضیه ها هستند.

زمانی که الگوریتم به نمونه اول می‌رسد (یک نمونه مثبت)، الگوریتم S را چک می‌کند و متوجه می‌شود که S بیش از حد خاص است (مثال را پوشش نمی‌دهد). پس مرز تا خاص‌ترین فرضیه‌ای که نمونه را پوشش دهد کلی می‌شود. این تغییر مرز در شکل ۲.۴ با S_1 نشان داده شده. در این مرحله هیچ تغییری در G لازم نیست، چون G_0 مثال را پوشش می‌دهد. در نمونه آموزشی بعدی (باز هم یک نمونه مثبت)، دوباره S تغییر می‌کند و به S_2 تبدیل می‌شود در حالی که G همچنان بدون تغییر می‌ماند. توجه می‌کنید که تأثیر این دو نمونه‌ی مثبت در الگوریتم Candidate-Elimination مشابه تأثیر آن‌ها در دو مرحله‌ی اول در الگوریتم FIND-S است.

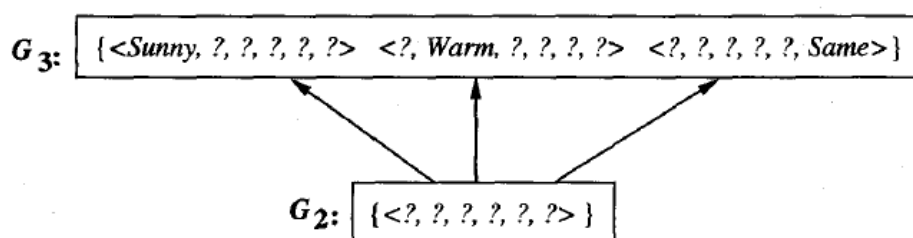


شکل ۲.۴ عملکرد الگوریتم Candidate-Elimination در دو گام اول. در ابتدا G_0 و S_0 دو مقدار اولیه‌ی G و S به ترتیب کلی‌ترین و خاص‌ترین فرضیه‌ها هستند. دو نمونه اول S را وادار می‌کنند که کلی‌تر شود (درست مثل الگوریتم FIND-S). این دو مثال تأثیری بر G ندارند. همان طور که در شکل نیز معلوم است دو نمونه اول مرز S را وادار می‌کنند که کلی‌تر شود و فضای ویژه را مشخص‌تر کند. در نقطه‌ی مقابل مثال‌های منفی G را مجبور می‌کنند که تا جای لازم خاص شود. نمونه آموزشی سوم را در نظر بگیرید (شکل ۲.۵). نمونه منفی نشان می‌دهد که G بیش از حد کلی است. چون بدون این خاص‌سازی پیش‌بینی می‌شود که این مثال یک نمونه مثبت است! پس باید G تا جایی که این نمونه را درست تشخیص دهد خاص‌تر شود. همان طور که در شکل ۲.۵ نیز نشان داده شده کلی‌ترین خاص‌سازی‌های بسیاری وجود دارد. همه این فرضیه‌ها عضو مرز جدید G_3 خواهند بود.

چرا با این که می‌دانیم ۶ کلی‌ترین خاص‌سازی برای G وجود داشت اما با این حال چرا فقط ۳ فرضیه به G_3 اضافه شد؟ برای مثال $h = \langle ?, ?, \text{Normal}, ?, ?, ? \rangle$ که نمونه منفی را به درستی منفی می‌داند و یک کلی‌ترین خاص‌سازی است اضافه نشده؟ جواب در این نهفته است که این فرضیه با نمونه‌های مثبت قبلی سازگار نیست. الگوریتم این نکته را با مقایسه‌ی کلی‌ترین خاص‌سازی‌ها با S می‌فهمد زیرا که S خلاصه‌ای از نمونه‌های مثبت قبلی را در خود ذخیره کرده و می‌توان با آن مشخص کرد که آیا کلی‌ترین خاص‌سازی‌ها با نمونه‌های مثبت

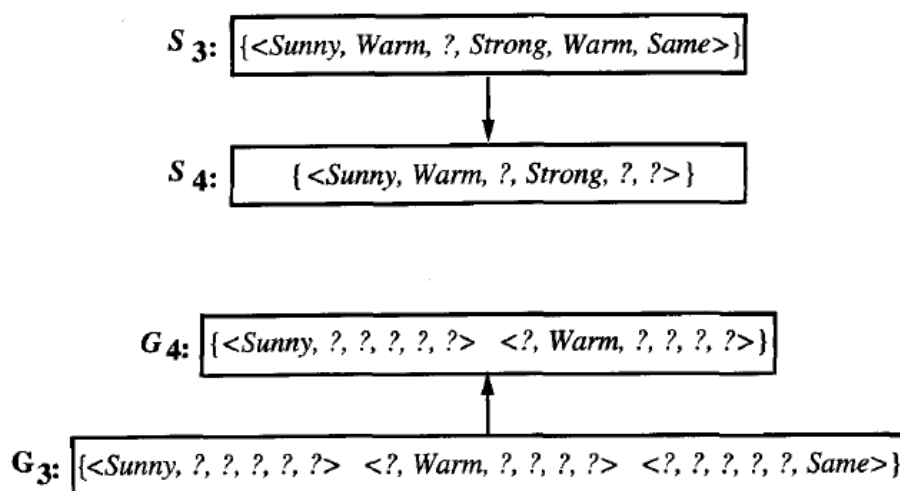
قبلی سازگار هستند یا نه. هر فرضیه ای که کلی تر از S باشد، طبق تعریف، با هر نمونه مثبت قبلی ای سازگار است. در نقطه‌ی مقابل نیز G خلاصه ای از نمونه های منفی را ذخیره می کند. و هر فرضیه ای که از G خاص تر باشد با نمونه های منفی قبلی سازگار نیست. چون طبق تعریف چنین فرضیه ای نمونه های منفی G را به عنوان نمونه منفی در خود دارد.

$S_2, S_3: \{ \langle \text{Sunny, Warm, ?, Strong, Warm, Same} \rangle \}$



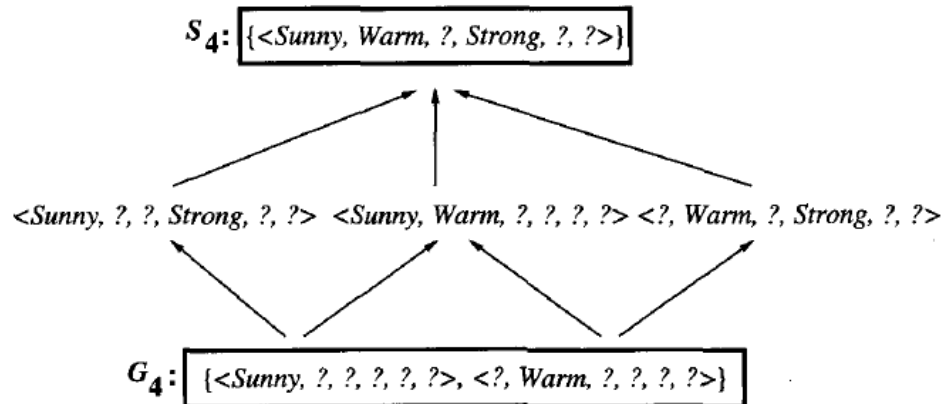
شکل ۲.۵ عملیات الگوریتم Candidate-Elimination در نمونه سوم.

نمونه سوم یک نمونه منفی است و باعث می شود که G_2 به G_3 تغییر کند. توجه داشته باشید که کلی ترین خاص سازی ها در G_3 قرار داده شده. در نمونه آموزشی پنجم (شکل ۲.۶) باز S کلی تر می شود و در طرف دیگر نیز یکی از اعضای G بخاطر نپوشاندن این نمونه جدید حذف می گردد. این حذف در خط اول الگوریتم جدول ۲.۵ نوشته شده. این سؤال به جا است که چرا باید فرضیه های اضافی حذف شود. جواب در این نکته است که خاص تر کردن باعث پوشش نمونه نخواهد شد. کلی تر کردن نیز باعث می شود که فرضیه با حداقل یکی از نمونه های منفی قبلی ناسازگار شود (طبق تعریف: هر فرضیه کلی تر حداقل یکی از نمونه های منفی را داراست). پس، ناچار فرضیه ی مذکور از G حذف می شود و با این حذف شدن شاخه ای از فضای ویژه کم می گردد.



شکل ۲.۶ قدم سوم در الگوریتم *Candidate-Elimination*.

نمونه سوم (نمونه مثبت) مرز S_3 را به S_4 تبدیل می‌کند. و همچنین یکی از اعضای G_3 در این فرایند حذف می‌شود، زیرا که کلی‌تر از S_4 نیست. بعد از انجام مراحل برای هر چهار نمونه، دو مرز S_4 و G_4 فضای ویژه و متعاقباً تمامی فرضیه سازگار با آن‌ها چهار نمونه را شامل می‌شود. کل فضای ویژه در شکل ۲.۷ نشان داده شده است. فضای ویژه‌ی پیدا شده به ترتیب نمونه‌ها وابسته نیست (زیرا که در انتها تمامی فرضیه‌های سازگار با نمونه‌ها را در بر خواهد گرفت). با بیشتر شدن تعداد نمونه‌ها کم کم دو مرز S و G به سوی یکدیگر می‌روند و فضای ویژه‌ی کوچک‌تری را تشکیل می‌دهند.



شکل ۲.۷ فضای ویژه‌ی مشخص شده برای مفهوم *EnjoySport* برای نمونه‌های آموزشی داده شده.

۲.۶ نکاتی چند در مورد فضای ویژه و الگوریتم *Candidate-Elimination*

۲.۶.۱ آیا الگوریتم *Candidate-Elimination* به سمت فرضیه‌ی درست می‌رود؟

اگر (۱) خطایی در فرضیه‌ها نباشد و (۲) فرضیه‌ای که درست مفهوم هدف را توصیف کند در H باشد، فضای ویژه‌ی خروجی الگوریتم *Candidate-Elimination* به سمت فرضیه‌ای میل می‌کند که مفهوم هدف را به درستی توصیف می‌کند. در واقع، می‌توان بعد از هر مثال بررسی کرد که آیا تعداد نمونه‌های آموزشی کافی بوده (فضای ویژه به فرضیه‌ی خاصی میل کرده؟ و ابهامات را در مورد مفهوم هدف از بین برده؟). زمانی که دو مرز S و G به یک مجموعه‌ی واحد و یکی برسند فرضیه هدف به طور کامل یاد گرفته شده است.

اما اگر نمونه‌های آموزشی خطا داشته باشد چه اتفاقی می‌افتد؟ برای مثال فرض کنیم که نمونه دوم در *EnjoySport* به جای نمونه مثبت به عنوان نمونه منفی ارائه شده بود. متأسفانه در چنین مثال‌هایی الگوریتم، مفهوم هدف را از فضای ویژه حذف می‌کرد! زیرا که به محض مواجهه با نمونه دوم تمام فرضیه‌هایی که آن‌را شامل می‌شد را حذف می‌کرد! البته در چنین شرایطی با دادن نمونه‌های کافی معلوم می‌شد که S و G به جایی میل می‌کنند که فضای ویژه تهی می‌شود. و این به این معناست که هیچ فرضیه‌ای در H وجود ندارد که با تمامی نمونه‌های آموزشی مطابقت داشته باشد. یکی از حالات ممکن این است که نمونه درست باشد اما مفهوم هدف در H وجود نداشته باشد (مثلاً مواقعی که تابع هدف یک تابع فصلی است و ما H را به صورت توابع عطفی در نظر گرفته‌ایم). چنین احتمالاتی را بعداً مفصلاً بررسی خواهیم کرد. اما در حال حاضر بنا بر این فرض است که تمامی مثال‌ها درستند و مفهوم هدف نیز در H وجود دارد.

۲.۶.۲ یادگیر چه مثال‌هایی را باید در خواست کند؟

تا الآن فرض می‌کردیم که مثال‌ها از معلم به یادگیر داده می‌شود و یادگیر هیچ حق انتخابی ندارد. حال فرض کنیم که یادگیر حق انتخاب داشته باشد. یعنی بتواند ویژگی‌هایی را در نظر بگیرد و از طریقی (آزمایش، طبیعت و یا معلم) مقدار تابع هدف را برای آن ویژگی‌ها پیدا کند. در چنین شرایطی وضع یادگیر بسیار متفاوت‌تر خواهد بود و یادگیر می‌تواند به انجام آزمایش بپردازد (مثلاً برای مفهوم "استحکام پل" می‌تواند پل‌های جدیدی با خواص دلخواه خود بسازد و از روی طبیعت بفهمد که پل مستحکم هست یا نه؟) یا زمانی که معلمی حاضر است می‌تواند از معلم سؤال کند (پلی طراحی کند و از معلم سؤال کند که آیا پل مستحکم است یا نه؟). به چنین نمونه‌هایی که ویژگی‌هایشان را یادگیر تعیین می‌کند آزمایش^۱ می‌گوییم.

دوباره فضای ویژه‌ی بدست آمده برای مثال EnjoySport را در نظر بگیرید (شکل ۲.۳). حال فرض کنیم می‌توانیم آزمایش کنیم. چه آزمایشی، آزمایش خوبی محسوب می‌شود؟ در کل با چه استراتژی کلی‌ای باید آزمایش کنیم؟ واضح است که یادگیر باید آزمایش‌هایی را امتحان کند که تفاوت بین فرضیه‌ها را نشان بدهند. چنین آزمایش‌هایی، آزمایش‌هایی هستند که در بعضی فرضیه‌های در فضای ویژه، نمونه مثبت و در بعضی دیگر نمونه منفی باشند. برای مثال:

<Sunny,Warm,Normal,Light,Warm,Same>

توجه می‌کنید که این نمونه سه فرضیه از ۶ فرضیه‌ی موجود (شکل ۲.۳) را راضی می‌کند. حال اگر این نمونه با مفهوم هدف سازگار بود S را کلی‌تر می‌کنیم. در مقابل اگر این مثال با مفهوم هدف ناسازگار بود مرز G را خاص‌تر می‌کنیم. در هر صورت اطلاعات مفیدی در مورد مفهوم هدف بدست می‌آید و تعداد اعضای فضای ویژه نیز نصف می‌شود.

در حالت کلی آزمایش بهینه، آزمایشی است که با نصف اعضای فضای ویژه سازگار و با نصف دیگر ناسازگار باشد. اگر آزمایش‌ها چنین شرایطی را داشته باشند در هر آزمایش تعداد اعضای فضای ویژه نصف می‌شود تا در آخر در $\log_2 |VS|$ آزمایش مفهوم هدف معلوم می‌شود. درست مثل بازی ۲۰ سؤالی^۲ که هر دفعه یک دسته فرضیه رد می‌شوند. در بازی ۲۰ سؤالی نیز بهترین استراتژی بازی پرسیدن سؤال‌هایی است که دقیقاً نصف فرضیه‌ها را در بر بگیرد است. از طرفی دیگر، همان طور که در شکل ۲.۳ نیز نشان داده شده می‌توان آزمایش‌هایی را ایجاد کرد که دقیقاً با نصف فضای ویژه سازگار باشد. در کل ممکن است این کار ممکن نباشد، در چنین شرایطی تعداد آزمایش‌ها متعاقباً بیش از $\log_2 |VS|$ خواهد شد.

۲.۶.۲ چگونه می‌توان از فضای ویژه برای تشخیص سازگاری نمونه‌ها با مفهوم هدف استفاده کرد؟

فرض کنید که فقط همان چهار مثال در اختیار یادگیر قرار گرفته و یادگیر هیچ گونه حق آزمایشی نیز ندارد، اما از یادگیر انتظار می‌رود که پیش‌بینی کند کدام مثال‌ها با مفهوم هدف سازگارند. با وجود اینکه فضای ویژه‌ی بدست آمده هنوز مفهوم هدف را دقیقاً نمی‌توان مشخص کرد. اما با این وجود می‌توان با استفاده از آن قطعاً گفت که مفهوم هدف بعضی نمونه‌ها را چگونه دسته بندی می‌کند. برای مثال فرض کنید که از یادگیر خواسته می‌شود تا مثال‌های جدول ۲.۶ را دسته بندی کند.

¹ query

² Twenty questions

توجه می‌کنید که با وجود اینکه نمونه A جزو نمونه های آموزشی نبوده اما هر ۶ فرضیه‌ی موجود در فضای ویژه آنرا مثبت دسته بندی می‌کنند (شکل ۲.۳). و چون تمامی فرضیه‌ها به اتفاق این نمونه را مثبت دسته بندی می‌کنند، یادگیر با اطمینان می‌تواند پیش بینی کند که این نمونه برای مفهوم هدف مثبت است (چون مفهوم هدف یکی از همین فرضیه‌هاست). پس جدا از اینکه کدام فرضیه مفهوم هدف است، نمونه A برای آن مثبت است. توجه دارید که لازم نیست برای تشخیص این که A برای تمامی فرضیه‌ها مثبت است آنرا با تک تک فرضیه‌ها تطبیق دهیم. فقط کافی است که نمونه A در تک تک اعضای S مثبت باشد (چرا؟). دلیل اینجاست که همه‌ی اعضای فضای ویژه حداقل از یکی از اعضای S کلی‌تر هستند و طبق تعریف کلی‌تری هر نمونه ای که اعضای S را راضی کند، تمامی فرضیه‌ها را راضی خواهد کرد.

نمونه	Sky	AirTemp	Humidity	Wind	Water	Forecast	EnjoySport
A	Sunny	Warm	Normal	Strong	Cool	Change	?
B	Rainy	Cold	Normal	Light	Warm	Same	?
C	Sunny	Warm	Normal	Light	Warm	Same	?
D	Sunny	Cold	Normal	Strong	Warm	Same	?

جدول ۲.۶ نمونه های جدید

به طور مشابه نمونه B یک نمونه منفی دسته بندی می‌شود زیرا که همه‌ی فرضیه های فضای ویژه آنرا نمونه منفی دسته بندی می‌کنند. و دوباره به طور مشابه کافی است فقط نمونه منفی اعضای G باشد (چرا؟).

در مورد نمونه C قضیه کاملاً متفاوت است. نیمی از فرضیه های فضای ویژه آنرا نمونه مثبت و نیمی دیگر آنرا نمونه منفی دسته بندی می‌کنند، پس تا زمانی که یادگیر با نمونه های آموزشی بیشتری مواجه نشده دسته بندی این مثال ممکن نیست. توجه دارید که نمونه C یک نمونه خوب برای آزمایش است. این اتفاق کاملاً قابل پیش بینی بود زیرا نمونه‌هایی که دسته بندیشان غیرممکن است حاوی اطلاعاتی هستند که ما نداریم (و برای همین نمی‌توانیم دسته بندیشان کنیم).

نمونه D توسط ۲ فرضیه نمونه مثبت و توسط ۴ فرضیه‌ی دیگر نمونه منفی دسته بندی می‌شود. در این حالت بر خلاف مثال‌های A و B نمی‌توان با اطمینان نمونه D را دسته بندی کرد. یک روش پذیرفتن رای اکثریت و دسته بندی D به عنوان نمونه منفی با درصد اطمینان (میزان رای اکثریت به رای کل) است. اگر همان طور که در فصل ۶ هم بحث شده احتمال درست بودن هر فرضیه را مساوی در نظر بگیریم چنین دسته بندی‌ای بهترین خروجی خواهد بود. با این حال همچنان احتمال مثبت بودن نمونه وجود دارد.

۲.۷ بایاس استقرایی

همان طور که قبلاً نیز بحث شد الگوریتم Candidate-Elimination در صورت وجود تابع هدف در H و درست بودن نمونه‌ها به تابع هدف میل می‌کند. اما اگر تابع هدف در H نبود چه؟ آیا می‌توان با در نظر گرفتن H به صورت تمامی فرضیه‌ها از این مشکل پرهیز کرد؟ تأثیر اندازه‌ی فضای فرضیه‌ها بر توانایی الگوریتم برای دسته بندی نمونه های جدید چگونه است؟ تأثیر اندازه‌ی فضای فرضیه‌ها بر تعداد نمونه های آموزشی لازم چگونه است؟ این سؤال‌های اساسی در مورد استقرا در کل مطرح هستند. در اینجا این سؤال‌ها را در متن Candidate-

Elimination بررسی خواهیم کرد. اما با این حال، می‌بینید که نتایج بدست آمده از این بررسی برای تمامی سیستم‌های یادگیری مفهوم قابل اجراست.

۲.۷.۱ فضای فرضیه ای بایاس دار

فرض کنید که می‌خواهیم فضای فرضیه‌ها، مفهوم هدف که نا معلوم است را در بر بگیرد. اولین و ساده‌ترین راه این است که تمامی فرضیه‌های ممکن را در فضای فرضیه ای قرار دهیم. دوباره مثال EnjoySport را در نظر بگیرید. ما در آنجا فضای فرضیه ای را تمام فرضیه‌های عطفی ممکن فرض کردیم. بخاطر این محدودیت، فضای فرضیه ای فرضیه‌ی بسیار ساده‌ی غیر عطفی “Sky=Sunny or Sky=Cloudy” را در بر نمی‌گیرد. در واقع در فضای فرضیه ای فعلی با دادن سه نمونه زیر به عنوان نمونه‌های آموزشی، خروجی الگوریتم برای فضای ویژه تهی می‌شود.

مثال	Sky	AirTemp	Humidity	Wind	Water	Forecast	EnjoySport
۱	Sunny	Warm	Normal	Strong	Cool	Change	Yes
۲	Cloudy	Warm	Normal	Strong	Cool	Change	Yes
۳	Rainy	Warm	Normal	Strong	Cool	Change	No

برای این که ثابت شود که فرضیه ای در H وجود ندارد که با سه نمونه بالا سازگار باشد، به این توجه کنید که خاص‌ترین فرضیه سازگار با دو و فرضیه‌ی اول به شکل زیر است:

$$S_2: \langle ?, Warm, Normal, Strong, Cool, Change \rangle$$

با این که این فرضیه خاص‌ترین فرضیه ممکن در H ، اما با این حال بیش از حد کلی است زیرا که نمونه سوم را مثبت دسته بندی می‌کند. مشکل در این جاست که ما H را با بایاس^۱ (محدود) کرده‌ایم و فقط ترکیب‌های عطفی را در نظر گرفته‌ایم. برای حل این مشکل لازم است که از فضای فرضیه ای شامل‌تری استفاده کنیم.

۲.۷.۲ یادگیر بدون بایاس

راه حل این مشکل که ممکن است مفهوم هدف در H نباشد این است که H را مجموعه‌ی تمامی فرضیه‌های قابل یادگیری در نظر بگیریم. و این به معنای پیدا کردن تمام زیر مجموعه‌های مجموعه‌ی X است. در کل، به مجموعه ای که تمامی این زیر مجموعه‌ها را در بر بگیرد مجموعه‌ی توانی^۲ X می‌گوییم.

در مثال EnjoySport تمامی حالت‌های یک روز که توسط شش ویژگی آن مشخص می‌شد ۹۶ حالت است. چند مفهوم روی این مجموعه می‌توان تعریف کرد؟ به عبارت دیگر تعداد اعضای مجموعه‌ی توانی X چقدر است؟ در کل تعداد زیر مجموعه‌های مجموعه‌ی X که $|X|$ عضو

¹ bias

² Power set

دارد $2^{|X|}$ است (اندازه‌ی مجموعه‌ی توانی X). 2^{96} مفهوم یا حدود 10^{28} مفهوم می‌توان روی X تعریف کرد. با توجه به آنچه در قسمت ۲.۳ گفته شد فضای فرضیه‌ی ای عطفی فقط ۹۷۳ عضو دارد، که واقعاً فضای بایاس داری بوده!

بیباید دوباره عمل یادگیری مفهوم EnjoySport را این دفعه با تعریف فضای فرضیه‌ی ای بدون بایاس H' (که H' همان مجموعه‌ی توانی X است) انجام دهیم. یک راه برای تعریف چنین H' ی اضافه کردن نقیض، عطف و فصل به تعداد دلخواه فرضیه‌هایی که قبلی به H است. برای مثال مفهوم هدف “Sky=Sunny or Sky=Cloudy” را می‌توان به صورت زیر نشان داد:

$$\langle \text{Sunny}, ?, ?, ?, ?, ? \rangle \vee \langle \text{Cloudy}, ?, ?, ?, ?, ? \rangle$$

با اضافه کردن چنین فرضیه‌هایی، می‌توانیم با اطمینان به سراغ الگوریتم Candidate-Elimination برویم و مطمئن باشیم که فضای فرضیه‌ها مفهوم هدف را در بر می‌گیرد. با این وجود حل کردن مشکل بدین شکل خود یک مشکل دیگر به وجود می‌آورد، با این فضای فرضیه‌ی ای پیش بینی نمونه‌های جدید غیرممکن می‌شود! برای معلوم شدن دلیل این مشکل فرض کنید که ما سه نمونه مثبت (x_1, x_2, x_3) و دو نمونه منفی (x_4, x_5) را به یادگیر می‌دهیم. بعد از بررسی هر پنج مثال مرز S فصل بین سه نمونه مثبت خواهد بود:

$$S: \{ (x_1 \vee x_2 \vee x_3) \}$$

زیرا که این خاص‌ترین فرضیه‌ی ممکن سازگار با سه مثال است. به طور مشابه نیز G به صورت زیر خواهد بود (کلی‌ترین فرضیه‌ی سازگار با نمونه‌های منفی)

$$G: \{ \neg(x_4 \vee x_5) \}$$

مشکل اینجاست که با این فضای فرضیه‌ی ای شامل، مرز S همواره فصل نمونه‌های مثبت و مرز G همواره نقیض فصل نمونه‌های منفی خواهد بود. پس تنها مثال‌هایی که توسط S و G حذف می‌شوند خود نمونه‌های آموزشی خواهند بود و برای میل کردن به یک فرضیه باید روی تک تک اعضای X آزمایش انجام دهیم.

در ابتدا ممکن است به نظر برسد که می‌توان با استفاده از آنچه در قسمت ۲.۳ گفته شد با رای گیری روی کلیه‌ی فرضیه‌های فضای ویژه حداقل به درصدی قابل قبول رسید، اما متأسفانه فقط برای نمونه‌های آموزشی نتیجه قطعی خواهد بود و برای تمامی مثال‌های غیر آموزشی این درصد ۵۰-۵۰ خواهد بود (نیمی از فرضیه‌ها نمونه را مثبت و نیمی دیگر نمونه را منفی دسته بندی می‌کنند(چرا؟)). برای معلوم شدن دلیل این امر توجه داشته باشید که زمانی که H مجموعه‌ی توانی X است و x نیز یک نمونه غیر آموزشی، در مقابل هر فرضیه‌ی ای در H که x را می‌پوشاند فرضیه‌ی ای دیگر مثل h' در مجموعه توانی وجود دارد که در روی تمامی اعضای X مشابه h است و تنها x را نمی‌پوشاند. و البته اگر h در فضای ویژه باشد مسلماً h' نیز در فضای ویژه خواهد بود، زیرا که در تمامی نمونه‌های آموزشی مشابه h رفتار می‌کند.

۲.۷.۳ بهبودگی یادگیری بدون بایاس

بحث بالا یک خاصیت بنیادی یادگیری استقرایی را مشخص کرد: یادگیری که هیچ پیش قضاوتی در مورد ماهیت مفهوم هدف نمی‌کند نمی‌تواند نمونه‌های جدید را دسته بندی کند. در واقع تنها دلیلی که باعث می‌شود الگوریتم Candidate-Elimination در مثال EnjoySport بتواند نمونه‌های جدید را دسته بندی کند این بود که در تعیین H بایاس شده است و فرض شده تابع هدف به صورت عطف ویژگی‌های موجود بیان شود. در واقع زمانی که این فرض درست است (و نمونه‌ها نیز خطا ندارند) این الگوریتم می‌تواند نمونه‌های جدید را

درست نیز دسته بندی کند. و زمانی که این فرض غلط باشد، الگوریتم Candidate-Elimination حداقل برای تعدادی از اعضای X دسته بندی اشتباه انجام می‌دهد.

چون یادگیری استقرایی نیاز به حداقل نوعی پیش فرض، بایاس در استقرا (بایاس نباید با بایاس تخمینی که در فصل ۵ آمده اشتباه گرفته شود) دارد، دسته بندی ی روش‌های مختلف بایاس مفید خواهد بود. نکته‌ی مهم اینجا این است که چگونه یادگیر نمونه‌های آموزشی را برای دسته بندی نمونه‌های دیگر تعمیم می‌دهد. اگر فرض کنیم که به الگوریتم یادگیری L ، نمونه‌های آموزشی دلخواه $D_c = \{ \langle x, c(x) \rangle \}$ برای یادگیری مفهوم دلخواه C داده شود و بعد از آموزش از L خواسته می‌شود تا نمونه جدید x_i را دسته بندی کند و $L(x_i, D_c)$ تشخیص الگوریتم L درباره‌ی نمونه جدید باشد (مثبت یا منفی) می‌توان این استنباط L را توسط عبارت زیر توصیف کنیم:

$$(D_c \wedge x_i) > L(x_i, D_c)$$

عبارت $y > z$ بدین معناست که z از روی y استنباط^۱ شده است. برای مثال اگر L همان الگوریتم Candidate-Elimination باشد و D_c نیز همان نمونه‌های جدول ۲.۱ باشد و x_i نیز همان نمونه اول جدول ۲.۶ باشد پس خواهیم داشت که $L(x_i, D_c) = (\text{EnjoySport} = \text{Yes})$.

چون L یک الگوریتم یادگیری استقرایی است، نمی‌توان ثابت کرد که حکم $L(x_i, D_c)$ درست یا غلط است. این فقط استنباطی است که L از نمونه‌های آموزشی D_c در مورد نمونه جدید x_i می‌کند. با این حال این سؤال جالب است که بپرسیم که چه پیش فرض‌های دیگری را می‌توان به $D_c \wedge x_i$ اضافه کرد تا همچنان $L(x_i, D_c)$ درست بماند. این مجموعه پیش فرض‌های جدید را بایاس استقرایی ی L ^۲ می‌نامیم. به طور دقیق‌تر بایاس استقرایی L را به صورت دسته پیش فرض‌های B تعریف می‌کنند که برای تمامی نمونه‌های x_i داشته باشیم:

$$(B \wedge D_c \wedge x_i) \vdash L(x_i, D_c)$$

عبارت $y \vdash z$ به این معناست که z قابل نتیجه گیری از y است^۳ (مثلاً: z از y اثبات می‌شود). پس بایاس استقرایی را به صورت دسته پیش فرض‌های کافی B تعریف می‌کنیم تا بتوان استقرا را با استنتاج (نتیجه گیری) توجیه کرد. به طور خلاصه،

تعریف: فرض کنید که الگوریتم یادگیری مفهوم L برای فضای نمونه‌های X تعریف شده است و C نیز یک مفهوم دلخواه روی X و $D_c = \{ \langle x, c(x) \rangle \}$ نیز نمونه‌های آموزشی دلخواهی از C باشند. اگر $L(x_i, D_c)$ تشخیص استقرایی L از x_i بر اساس D_c باشد. بایاس استقرایی L هر دسته فرضیاتی مثل B است که برای هر مفهوم هدف C و نمونه‌های آموزشی مربوطه D_c داشته باشیم:

$$(\forall x_i \in X) [(B \wedge D_c \wedge x_i) \vdash L(x_i, D_c)] \quad (2.1)$$

پس با این تعریف، بایاس استقرایی الگوریتم Candidate-Elimination چیست؟ برای جواب این سؤال، باید اول $L(x_i, D_c)$ را دقیقاً برای این الگوریتم مشخص کنیم: برای هر دسته نمونه آموزشی D_c الگوریتم Candidate-Elimination اول فضای ویژه‌ی متناسب با آن VS_{H, D_c} را محاسبه می‌کند و سپس x_i را از رای گیری روی آن دسته بندی می‌کند. بیایید فرض کنیم که الگوریتم فقط زمانی که کل

¹ infer

² inductive bias

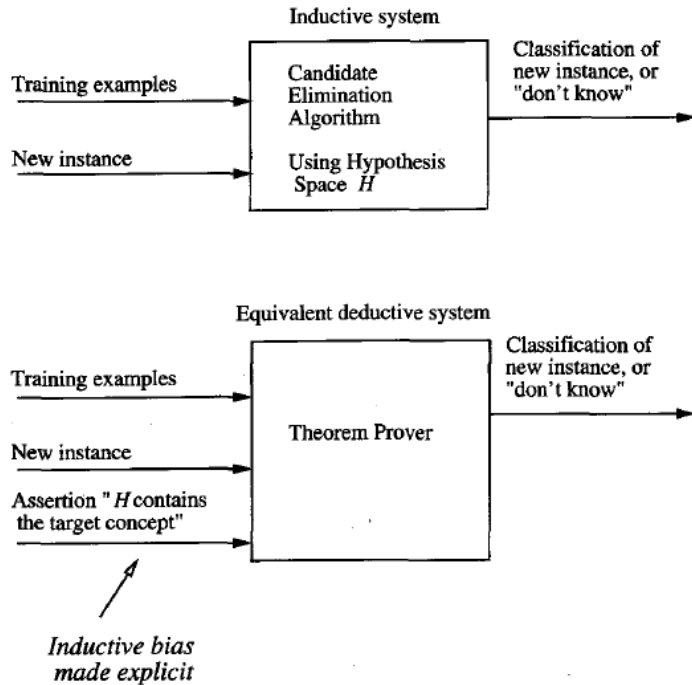
³ Follows deductively

فرضیه های فضای ویژه به طور صد در صدی رای دهند خروجی بدهد. حال با معلوم شدن تعریف $L(x_i, D_c)$ برای الگوریتم Candidate-Elimination، بایاس استقرایی آن چیست؟ جواب واضح است، طبق تعریف پیش فرض $c \in H$ است. با داشتن این پیش فرض هر استنباط استقرایی توسط Candidate-Elimination درست خواهد بود.

برای روشن شدن دلیل اینکه چرا از D_c و مشخصات x_i و اینکه $B = \{c \in H\}$ دسته بندی $L(x_i, D_c)$ نتیجه گیری می شود، به بحث زیر توجه کنید. اول اینکه، توجه داشته باشید که پیش فرض $c \in H$ نتیجه می دهد که $c \in VS_{H,D_c}$ (زیرا که $c \in H$ و از تعریف فضای ویژه به عنوان مجموعه ای شامل تمام اعضای H که با نمونه های D_c مطابقت دارد و تعریف $D_c = \{ \langle x, c(x) \rangle \}$ به عنوان نمونه های آموزشی ای که با C سازگار است). دوم اینکه، توجه داشته باشید که فرض کردیم $L(x_i, D_c)$ رای صد درصدی اعضای فضای ویژه است. بنا برین، اگر $L(x_i, D_c)$ را به عنوان خروجی بدهد، حتما تمامی فرضیه های VS_{H,D_c} نیز همان تشخیص را داشته اند و چون $c \in VS_{H,D_c}$ بنابراین $c(x_i) = L(x_i, D_c)$. به طور خلاصه الگوریتم Candidate-Elimination تعریف شده با بایاس زیر مطابقت دارد:

بایاس استقرایی الگوریتم Candidate-Elimination. باید فضای فرضیه ها H مفهوم هدف C را شامل شود.

شکل ۲.۸ خلاصه ای وضع را به صورت نموداری نمایش می دهد. الگوریتم استقرایی Candidate-Elimination در بالای شکل دو ورودی دارد: نمونه های آموزشی و نمونه جدید. در پایین شکل ثابت کننده ی قضیه با همان دو ورودی و یک فرض سوم، "مفهوم هدف در H موجود است"، نشان داده شده است. به طور کلی، این دو سیستم برای هر دسته نمونه آموزشی و هر نمونه جدید خروجی یکسان می دهند. البته بایاس استقرایی که ورودی سوم ثابت کننده ی قضیه است به طور بالقوه در کد الگوریتم Candidate-Elimination وجود دارد. به عبارتی، این فقط در شکل نشان داده شده و تأثیری دیگر ندارد. از طرف دیگر این ورودی به ما اطمینان می دهد که خروجی حتماً درست است.



شکل ۲.۸ مدل سازی سیستم‌های استقرایی با سیستم‌های استقرایی هم‌ارز.

رفتار ورودی-خروجی الگوریتم *Candidate-Elimination* با استفاده از فضای فرضیه H درست مشابه رفتار استقرایی ثابت کننده‌ی قضیه‌ی ای است که از پیش فرض "مفهوم هدف در فضای فرضیه‌ی ای وجود دارد" استفاده می‌کند. از این رو پیش فرض اضافه شده، "مفهوم هدف در فضای فرضیه‌ی ای وجود دارد"، بایاس استقرایی *Candidate-Elimination* نامیده می‌شود. دسته بندی سیستم‌های استقرایی با بایاس‌های استقرایی‌شان با ما این اجازه را می‌دهد که آن‌ها را با سیستم‌های استقرایی نظیر مد سازی کنیم. این مدل سازی باعث می‌شود تا بتوانیم سیستم‌های استقرایی را بر اساس نحوه‌ی استقرار روی نمونه‌های آموزشی مقایسه کنیم.

یکی از مزیت‌های نگاه بایاس استقرایی به سیستم‌های استنباط استقرایی، پیدا کردن نحوه‌ی تأمین نمونه‌های آموزشی برای نمونه‌های جدید، بدون این که با مراحل الگوریتم‌شان درگیر شویم است. مزیت دوم اینکه، می‌توانیم قدرت یادگیرهای الگوریتم‌های مختلف را مقایسه کنیم. برای مثال سه الگوریتم زیر به ترتیب ضعیف‌ترین به قوی‌ترین بایاس ترتیب شده‌اند:

۱. *Rote-Learner*: هر نمونه آموزشی را در حافظه ذخیره سازی می‌کند. برای دسته بندی نمونه‌های جدید، فقط در حافظه جستجو می‌کند، اگر مثال میان نمونه‌های آموزشی بود جواب را خروجی می‌دهد، در غیر این صورت خروجی نمی‌دهد!

۲. الگوریتم *Candidate-Elimination*: همان طور که قبلاً نیز توضیح داده شد، این الگوریتم خاص‌ترین و کلی‌ترین فرضیه‌های سازگار با نمونه‌های آموزشی را پیدا می‌کند و با استفاده از فضای ویژه (رای صد درصدی) نمونه‌های جدید را دسته بندی می‌کند.

۳. *FIND-S*: این الگوریتم خاص‌ترین فرضیه‌ی ممکن سازگار با نمونه‌های آموزشی را پیدا کرده و از آن برای دسته بندی نمونه‌های جدید استفاده می‌کند.

الگوریتم *Rote-Learner* هیچ بایاس استقرایی‌ای ندارد و نمونه‌های جدید را فقط با توجه به نمونه‌های آموزشی دسته بندی می‌کند و هیچ فرض اضافه‌ای نمی‌کند. *Candidate-Elimination* بایاس استقرایی قوی‌تری دارد: مفهوم هدف در میان اعضای فضای فرضیه‌ها H موجود است. چون *Candidate-Elimination* بایاس استقرایی قوی‌تری نسبت به *Rote-learner* دارد مثال‌هایی را دسته بندی می‌کند که *Rote-Learner* از دسته بندی آن‌ها عاجز است. البته توجه دارید که درستی این دسته بندی‌ها کاملاً به درستی بایاس استقرایی وابسته است. بایاس استقرایی *FIND-S* حتی از این هم قوی‌تر است: علاوه بر این که فرض می‌کند که مفهوم هدف یکی از اعضای فضای فرضیه‌هاست، فرض می‌کند که تمامی نمونه‌های جدید نمونه منفی هستند مگر اینکه خلافش ثابت شود (با اطلاعات قبلی ثابت شده باشد که نمونه مثبت است).

بد نیست که در مواجهه با دیگر متد‌های استنباط استقرایی بایاس استقرایی و قدرت آن‌را نیز بررسی کنیم. متدهایی که بایاس استقرایی قوی‌تری دارند قدرت بیشتری نیز در دسته بندی نمونه‌های جدید دارند و نمونه‌های جدید بیشتری را می‌توانند دسته بندی کنند. بعضی بایاس‌های استقرایی پیش فرض‌های دسته بندی کننده‌ای هستند که دسته‌ای از فرضیه‌ها را به کلی کنار می‌گذارند، مثل این بایاس: "فضای فرضیه‌ها H باید شامل مفهوم هدف باشد". بعضی دیگر از بایاس‌ها فقط ترتیبی برای اولویت بین فرضیه‌ها می‌گذارند، مثل: "فرضیه‌های خاص‌تر بر فرضیه‌های کلی‌تر ارجحیت دارند". بعضی از بایاس‌ها نیز به طور بالقوه در یادگیر قرار داده شده‌اند، بایاس‌هایی که ذکر شد. در فصول ۱۱ و ۱۲ سیستم‌های دیگری را مورد بحث قرار خواهیم داد که بنا به خواست یادگیر بایاس تغییر می‌کند.

۲.۸ خلاصه و منابع برای مطالعه‌ی بیشتر

نکات اصلی این فصل:

- می‌توان به مسایل یادگیری مفهوم به دید جستجوی در میان تمامی فرضیه‌های ممکن نگاه کرد.
 - ترتیب کلی‌تری فرضیه‌ها می‌تواند به این جستجو سازمان ببخشد، این سازماندهی در جستجوی فضای فرضیه‌ها را می‌توان برای هر مسئله‌ی یادگیری مفهوم به کار برد.
 - الگوریتم FIND-S با استفاده از ترتیب کلی‌تری، یک جستجو در میان فرضیه‌ها را در یک شاخه از ترتیب کلی‌تری، ترتیب می‌دهد، تا در آخر خاص‌ترین فرضیه‌ی مطابق با نمونه‌های آموزشی را پیدا کند.
 - الگوریتم Candidate-Elimination با استفاده از ترتیب کلی‌تری فضای ویژه را (تمامی فرضیه‌های سازگار با نمونه‌های آموزشی) با استفاده از محاسبه‌ی خاص‌ترین (S) و کلی‌ترین (G) فرضیه‌های سازگار پیدا می‌کند.
 - چون S و G تمامی فرضیه‌های سازگار با نمونه‌ها را محدود می‌کنند، به یادگیر اطلاعاتی قطعی در مورد مفهوم هدف می‌دهند. این فضای ویژه دو کاربرد دارد: اول اینکه می‌توان بررسی کرد که آیا فرضیه‌های ممکن همگرا شده‌اند، تا مشخص شود که آیا نمونه‌های آموزشی کافی بوده و چه آزمایش‌هایی برای همگرا سازی فضای ویژه مناسب است. دوم اینکه می‌توان با استفاده از آن برای دسته‌بندی نمونه‌های جدید استفاده کرد.
 - با وجود اینکه فضای ویژه و الگوریتم Candidate-Elimination محیطی ادراکی برای درک یادگیری مفهوم ایجاد می‌کنند، دو مشکل عمده دارند: اول اینکه در مقابل خطا و داده‌های نویز دار کاملاً آسیب پذیرند و دوم اینکه اگر مفهوم هدف در H نباشد نمی‌توانند آن را پیدا کنند. در فصل ۱۰ با الگوریتم‌هایی که از ترتیب کلی‌تری استفاده می‌کنند و در مقابل نویز نیز مقاومند می‌پردازیم.
 - الگوریتم‌های استقرایی فقط زمانی می‌توانند نمونه‌های جدید را دسته‌بندی کنند که بایاسی داشته باشند. بایاس استقرایی باعث می‌شود که این الگوریتم‌ها فرضیه‌ای را از فرضیه‌ی دیگر مقدم‌تر بدانند و با آن نمونه‌های جدید را دسته‌بندی کنند. بایاس استقرایی در نظر گرفته شده در الگوریتم Candidate-Elimination این است که فرض شده مفهوم هدف در فضای فرضیه‌ها موجود است ($c \in H$). استنباط‌هایی که این الگوریتم انجام می‌دهد مبتنی بر نمونه‌های آموزشی و این بایاس استقرایی است.
 - اگر فضای فرضیه‌ها همه‌ی فرضیه‌های ممکن را داشته باشد (مجموعه‌ی توانی مثال‌ها) بایاس استقرایی Candidate-Elimination بر طرف می‌شود. اما متأسفانه هرگونه دسته‌بندی نمونه‌های جدید با این عمل از بین می‌رود. یادگیر بدون بایاس نمی‌تواند با استقرا نمونه‌های جدید را دسته‌بندی کند.
- ایده‌ی یادگیری مفهوم و ترتیب کلی‌تری خیلی جدید نیست. (Burner et al. 1957) اولین مطالعات را در مورد یادگیری مفهوم روی انسان‌ها انجام داد، (Hunt and Hovland 1963) (1963) اولین تلاش‌ها را برای الگوریتمیک کردن آن انجام دادند. تز دکترای معروف (Wiston) که یادگیری مفهوم را به صورت جستجویی با استفاده از عملیات‌های کلی سازی و جزیی سازی تعریف کرد. (Plotkin 1970, 1971) فرمولی اولیه از رابطه‌ی کلی‌تری وابسته به جانشینی θ ^۱ (که در فصل ۱۰ مطرح شده) بدست آورد. (Simon and Lea 1973) یادگیری مفهوم را به معنای جستجو در فضای فرضیه‌ای را مطرح کردند. بقیه‌ی سیستم‌های یادگیری مفهوم اولیه، (Popplestone 1969)، (Michalski 1973)، (Buchanan 1974)، (Vere 1975) و (Hayes-Roth 1974) هستند.
- تعداد زیادی از الگوریتم‌هایی که تا به حال برای یادگیری مفهوم طراحی شده مبتنی بر نمایش نمادین^۲ بوده است. در فصل ۱۰، تعداد بسیار دیگری از الگوریتم‌های یادگیری مفهوم از جمله الگوریتم‌هایی که از منطق مرتبه اول استفاده می‌کنند، الگوریتم‌هایی که نسبت به نویز و خطا مقاوم هستند و الگوریتم‌هایی که حتی اگر مفهوم هدف در میان فضای فرضیه‌ها نباشد باز هم درست کار می‌کنند را بررسی خواهیم کرد.

¹ θ -sumsumption

² symbolic representation

فضا های ویژه و الگوریتم Candidate-Elimination توسط میشل (Mitchell 1977,1982) طراحی و معرفی شد. کاربرد این الگوریتم برای استنباط قوانین طیف بینی جرمی نیز توسط وی انجام شد (1979). و همچنین کاربرد آن برای یادگیری قوانین کنترل جستجو نیز توسط وی در سال ۱۹۸۳ انجام شد. Haussler (1988) نشان داد که مرز کلی متناسب با تعداد نمونه های آموزشی می تواند، حتی زمانی که فضای فرضیه ای شامل عطف ویژگی های نمونه هاست، به صورت نمایی افزایش یابد. (Smith and Rosenbloom 1990) نشان دادند که تغییر کوچکی در نمایش مجموعه ی G می تواند در بعضی موارد پیچیدگی را بهبود بخشد و (Hirsh 1992) نشان داد که یادگیری می تواند در بعضی موارد که G مرتب^۱ متناسب با چند جمله ای از تعداد نمونه ها باشد. (Subramanian and Feigenbaum 1986) متدی را که می تواند نمونه های موثری در بعضی موارد با فاکتور گیری از فضای ویژه^۲ انجام دهد. یکی از بزرگ ترین محدودیت های عملی الگوریتم Candidate-Elimination نیاز آن به داده های آموزشی بدون خطاست. (Mitchell 1979) تعمیمی از این الگوریتم را ارائه می کند که می تواند با خطای دسته بندی محدود و از پیش تعیین شده کار کند و (Hirsh 1990,1994) تعمیم زیبایی برای کار با خطای محدود در ویژگی های حقیقی مقدار که نمونه های آموزشی را توصیف می کنند را ارائه می کند. (Hirsh 1990) الگوریتم Incremental Version Space Merging را که تعمیمی از الگوریتم Candidate-Elimination است را برای حالاتی که اطلاعات آموزشی حالات مختلفی از قیود توسط فضای ویژه بیان شده است را ارائه داد. اطلاعات هر یک از قیود توسط فضای ویژه ای بیان شده و سپس قیود با تقسیم فضای ویژه ترکیب می شوند. (Sebag 1994, 1996) روش یادگیری ای را که فضای ویژه ی فصلی می نامد را برای یادگیری فصل مفاهیم از داده های خطا دار ارائه می کند. در این روش، فضای ویژه ی مجزایی برای هر نمونه ی آموزشی یاد گرفته شده و نمونه های جدید با رای گیری میان این فضاها و ویژه دسته بندی می شوند. وی تحقیقات زیادی را در زمینه های مختلف انجام داده و برتری نتایج الگوریتم خود را نسبت به الگوریتم های مشابه مانند درخت تصمیم و k-nearest neighbor نشان می دهد.

تمارین

۲.۱ توضیح دهید که چرا اندازه ی فضای فرضیه ها در مثال EnjoySport ۹۷۳ شد. با اضافه کردن ویژگی دیگری مثل WaterCurrent که سه حالت Light، Moderate و Strong را داشته باشد، تعداد مثال ها و فرضیه های ممکن چگونه تغییر می کرد؟ در حالت کلی اگر ویژگی A را که k حالت دارد را اضافه کنیم تعداد مثال ها و فرضیه های ممکن چگونه تغییر خواهد کرد؟

۲.۲ برای الگوریتم Candidate-Elimination دو مرز S و G را برای جدول ۲.۱ با ترتیب عکس حساب کنید. با وجود اینکه فضای ویژه ی بدست آمده در انتها یکی است (چرا؟)، اما در مراحل میانی S و G متفاوتی بدست می آید. آیا می توانید با عوض کردن ترتیب کاری کنید که مجموع تعداد اعضای S و G در تمام مراحل کمینه شود؟

۲.۳ دوباره مسئله ی EnjoySport را با فضای فرضیه ای قسمت ۲.۲ در نظر بگیرید. حال اگر فضای فرضیه ای H' را تمام ترکیب های دو تایی فصلی H در نظر بگیریم الگوریتم Candidate-Elimination را برای این فضای فرضیه ای جدید و نمونه های جدول ۲.۱ انجام دهید. (سری S ها و G های مراحل را بدست آورید).

یک نمونه بسیار ساده از H':

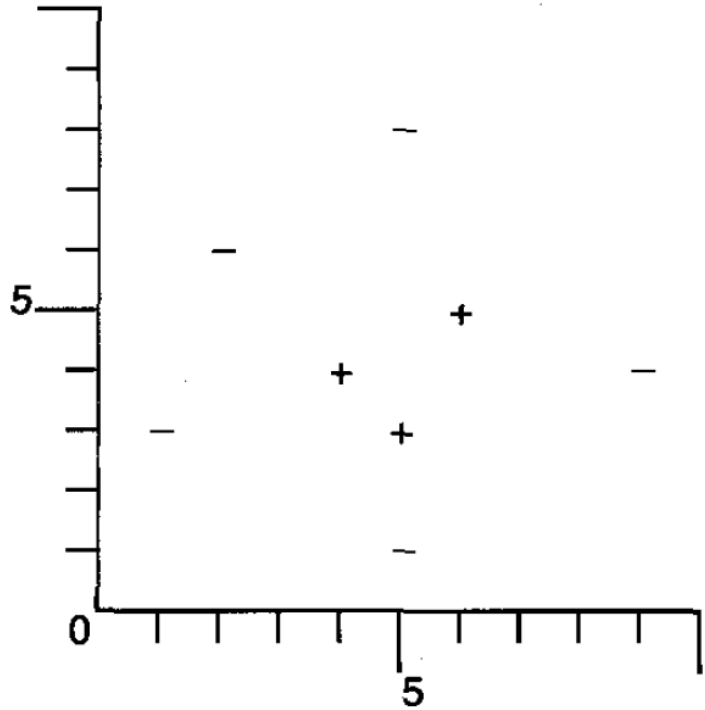
¹ sorted

² Factoring version space

<?,Cold,High,?,?,?>V<Sunny,?,High,?,?,Same>

۲.۴ اگر فضای نمونه ای نقاطی با x و y صحیح باشند و فضای فرضیه‌ها نیز مستطیل‌ها باشند. به عبارت دقیق‌تر فرضیه‌ها به فرم $a \leq x \leq b$ و $c \leq y \leq d$ هستند که a, b, c, d در آن اعداد صحیحند.

(a) فضای ویژه ای که با نمونه های آموزشی شکل زیر مطابقت دارد را در نظر بگیرید. مرز S در این فضای ویژه چیست؟ آن را بنویسید و در شکل نیز مشخص کنید.



(b) G را برای فضای ویژه‌ی مربوطه چیست؟ آن را بنویسید و در شکل نیز مشخص کنید.

(c) فرض کنید که حالا یادگیر حق آزمایش دارد. آزمایشی را که اندازه‌ی فضای ویژه را کم می‌کند پیدا کنید. یک آزمایش که اندازه‌ی فضای ویژه را کم نمی‌کند نیز پیدا کنید.

(d) حال فرض کنید که مفهوم هدف خاصی را برای تعلیم در نظر گرفته‌ایم مثل $3 \leq x \leq 5$ و $2 \leq y \leq 9$ کمترین تعداد نمونه آموزشی‌ای که لازم است به یادگیر الگوریتم Candidate-Elimination بدهیم تا این مفهوم را یاد بگیرد چقدر است؟

۲.۵ نمونه های آموزشی مثبت و منفی زیر را که برای آموزش مفهوم "جفت‌هایی که با هم در یک خانه زندگی می‌کنند" در نظر بگیرید. ویژگی‌ها به ترتیب جنسیت، رنگ مو (سیاه، قهوه ای و بور)، قد (بلند، متوسط و کوتاه) و ملیت (امریکایی، فرانسوی، آلمانی، ایرلندی، هندی، ژاپنی، پرتغالی) هستند.

+ (<مرد،قهوه ای،بلند،امریکایی>،<زن،سیاه،کوتاه،امریکایی>)

+ (<مرد،قهوه ای،کوتاه،فرانسوی>،<زن،سیاه،کوتاه،امریکایی>)

- (<زن،قهوه ای،بلند،آلمانی>،<زن،سیاه،کوتاه،هندی>)

+ (<مرد،قهوه ای،بلند،ایرلندی>،<زن،قهوه ای،کوتاه،ایرلندی>)

فرض کنید فضای فرضیه ای بر روی این مثال‌ها به صورت زیر تعریف شده که هر یک از ویژگی‌ها می‌تواند ؟ یا ۰ یا یک مقدار باشد (همیشه ۴ ویژگی معلوم است). برای مثال:

(<مرد،؟،بلند،؟>،<زن،؟،؟،ژاپنی>)

که این مثال تمامی زوج‌هایی را در بر می‌گیرد که نفر اول مردی قد بلند (از هر ملیتی با هر رنگ مویی) است و نفر دوم زنی ژاپنی (از هر قدی و هر رنگ مویی) است را در بر می‌گیرد

(a) فرایند الگوریتم Candidate-Elimination را برای مثال‌های فوق طی کنید و مرزهای S و G را برای فضای ویژه بعد از هر مثال بیابید.

(b) چند تا از فرضیه‌های فضای فرضیه ای تعریف شده با مثال زیر سازگارند؟

+ (<مرد،سیاه،کوتاه،پرتغالی>،<زن،بور،بلند،هندی>)

(c) فرض کنید که فقط نمونه مثبت قسمت b را به عنوان نمونه آموزشی داریم و حالا به یادگیر اجازه داده می‌شود که آزمایش انجام دهد. سری‌ای از آزمایش‌ها را ترتیب دهید که در هر صورت ما را به فرضیه‌ی درست برساند (با فرض اینکه مفهوم هدف در فضای فرضیه ای وجود دارد). کوتاه‌ترین سری آزمایش‌ها را انتخاب کنید. طول این سری چه ربطی به فرضیه جواب قسمت b دارد؟

(d) با توجه به این که فضای فرضیه‌های تعریف شده تمامی مفهوم‌های هدف قابل تعریف روی فضای مثال‌ها را در بر نمی‌گیرد، اگر H را طوری تعریف می‌کردیم که تمامی مفاهیم هدف ممکن را در بر بگیرد، جواب قسمت c چه تغییری می‌کرد؟

۲۶ اثبات قضیه‌ی ۲.۱ (ارایه فضای ویژه) را کامل کنید

۲.۷ مسئله‌ی یادگیری مفهومی را در نظر بگیرید که در آن هر مثال یک عدد حقیقی است و هر فرضیه نیز بازه ای روی اعداد حقیقی است. به طور دقیق‌تر، فضای فرضیه‌ها H به صورت $a < x < b$ در نظر گرفته می‌شود که در آن a, b اعداد حقیقی‌اند. برای مثال $4.5 < x < 6.1$ تمامی اعداد بین ۴.۵ و ۶.۱ را مثبت و بقیه‌ی اعداد حقیقی را منفی دسته بندی می‌کند. غیر رسمی، توضیح دهید که چرا برای تعدادی نمونه مثبت خاص‌ترین فرضیه موجود نیست. تغییری را در فضای فرضیه‌ها پیشنهاد کنید که این چنین فرضیه‌هایی موجود باشد.

۲.۸ در این فصل فضای فرضیه ای بدون بایاس (مجموعه‌ی توانی مثال‌ها) معرفی شد و گفته شد که با استفاده از آن دقیقاً نصف فرضیه‌ها با هر مثال را مثبت و نصفی دیگر منفی دسته بندی می‌کنند. این گزاره را اثبات کنید. به عبارت دیگر ثابت کنید برای هر فضای مثال‌های X و نمونه‌های آموزشی D و هر نمونه جدید c، اگر H مجموعه‌ی توانی X باشد دقیقاً نیمی از فرضیه‌های $VS_{H,D}$ ، x را مثبت و نیمی دیگر x را منفی دسته بندی می‌کنند.

۲.۹ مسئله ای از یادگیری مفهوم را در نظر بگیرید که هر مثال عطفی از n ویژگی منطقی a_1, \dots, a_n باشد. یک مثال ممکن است این باشد:

$$(a_1 = T) \wedge (a_2 = F) \wedge \dots \wedge (a_n = T)$$

حال فرض کنید که فضای فرضیه‌ها به صورت فصلی از ویژگی‌ها تعریف شود. مثال:

$$(a_1 = T) \vee (a_5 = F) \vee (a_7 = T)$$

الگوریتمی پیشنهاد کنید که نمونه‌های آموزشی را بگیرد و اگر فرضیه‌ی سازگاری با آن‌ها وجود داشت آن‌را خروجی دهد. الگوریتم باید در زمان متناسب با چند جمله‌ای از n و تعداد نمونه‌های آموزشی اجرا شود.

۲.۱۰ برنامه‌ای برای الگوریتم FIND-S بنویسید و آن‌را برای مثال EnjoySport اجرا کنید و نشان دهید که مراحل همان مراحل قسمت ۲.۴ است. با این برنامه تعیین کنید که چند نمونه تصادفی آموزشی برای تعیین دقیق مفهوم هدف لازم است. برنامه‌ای بنویسید که نمونه‌های آموزشی تصادفی متناسب با مفهوم زیر ایجاد کند:

<Warm.Sunny؟؟؟؟>

فرض کنید که خروجی برنامه‌ی ایجاد نمونه آموزشی را به برنامه‌ی اول بدهیم، آیا می‌توانید حدس بزنید که به طور متوسط چند نمونه آموزشی لازم است تا برنامه به مفهوم هدف پی ببرد؟ این کار را حداقل ۲۰ بار انجام دهید و متوسط آن‌را حساب کنید. فکر می‌کنید با عوض کردن تعداد ؟‌های مفهوم هدف این تعداد چگونه تغییر می‌کند؟ تعداد ویژگی‌ها چه تأثیری بر این مقدار (تعداد نمونه‌های آموزشی لازم) دارد؟

فرهنگ لغات تخصصی فصل (فارسی به انگلیسی)

Consistent	سازگار
space version	فضای ویژه
Hypothesis	فرضیه
Specific	خاص
General	کلی
training example	نمونه آموزشی
minimal specialization	کلی‌ترین خاص‌سازی‌ها
maximal generalization	خاص‌ترین کلی‌سازی‌ها
target concept	مفهوم هدف
Classify	دسته‌بندی
Bias	پایاس
Negation	نقیض
Expressive	شامل
unobserved instance	مثال غیر آموزشی
inductive bias	پایاس استقرایی