فصل ششم: یادگیری بیزی

استدلال بیز روشی احتمالی برای استنتاج ارائه می‌کند. این روش بر اساس این فرض است که کمیت‌های مورد نظر از توزیع‌های احتمال پیروی می‌کنند و تصمیم‌گیری بهینه را می‌توان با استدلال بر این توزیع‌های احتمال و داده‌های مشاهده شده انجام داد. اهمیت این روش در یادگیری ماشین این است که روشی کمّی برای ارزیابی مدارک فرضیه‌ها ارائه می‌کند. استدلال بیزی اساس الگوریتم‌های یادگیری‌ای که با استفاده از احتمالات کار می‌کنند است. همچنین استدلال بیز چارچوبی[1] برای بررسی عملیات دیگر الگوریتم‌هایی که از احتمالات استفاده نمی‌کنند ایجاد می‌کند.

٦,١ معرفی

متدهای یادگیری بیزی به دو دلیل به مطالعه‌ی ما در مورد یادگیری ماشین مربوط می‌شود. اول اینکه الگوریتم‌های یادگیری بیزی احتمال صریح هر فرضیه، مثل دسته‌بندی کننده‌ی ساده‌ی بیز[2]، را محاسبه می‌کنند، این نوع روش‌ها از پرکاربردترین روش‌ها در حل بعضی از مسائل یادگیری هستند. برای مثال، (Michie 1994) تحقیقی در مورد تفاوت‌های دسته‌بندی کننده‌ی ساده‌ی بیز و دیگر الگوریتم‌های یادگیری، از جمله یادگیری درختی و شبکه‌های عصبی، ارائه می‌کند. این تحقیقات نشان می‌دهد که کارایی دسته‌بندی کننده‌ی ساده‌ی بیز در بعضی ویژگی‌ها ضعیف‌تر از دیگر الگوریتم‌ها و در بعضی ویژگی‌ها بهتر است. در این فصل دسته‌بندی کننده‌ی ساده‌ی بیز را به همراه چند مثال بررسی می‌کنیم. در کل، کاربرد این الگوریتم را روی مسئله‌ی دسته‌بندی متونی مثل مقالات خبری الکترونیک بررسی می‌کنیم. برای چنین کارهای یادگیری‌ای دسته‌بندی کننده‌ی ساده‌ی بیز یکی از بهترین الگوریتم شناخته شده است.

دلیل دوم اهمیت متدهای بیز در مطالعه‌ی ما در یادگیری ماشین زمینه‌ی مساعدی است که این متدها برای درک الگوریتم‌های یادگیری‌ای که مستقیماً با احتمالات کار نمی‌کنند ایجاد می‌کند. برای مثال، در این فصل، ما الگوریتم‌هایی چون Find-S و Candidate-Elimination،

[1] framework
[2] naive Bayes classifier

را که در فصل ۲ آمده بود، برای مشخص کردن شروطی که خروجی، محتمل‌ترین فرضیه‌ی سازگار با نمونه‌های آموزشی باشد را بررسی خواهیم کرد. همچنین با بررسی‌ای بیزی توجیهی برای یکی از انتخاب‌های کلیدی الگوریتم‌های یادگیری شبکه‌های عصبی (انتخاب تابع خطای مجموع مربعات خطا برای جستجوی فضای شبکه‌های عصبی ممکن) ارائه خواهیم کرد. همچنین در این بخش اشتقاق تابع خطای جایگزینی را محاسبه می‌کنیم، cross-entropy، این معیار زمانی که تابع هدف احتمالات را پیش‌بینی می‌کند از معیار مجموع خطاهای مربعی کارآمدتر است. از نظری بیزی بایاس استقرایی الگوریتم‌های درختی که به درخت‌های کوچک‌تر علاقه دارند و قانون کوتاه‌ترین طول توضیح را بررسی خواهیم کرد. آشنایی پایه‌ای با روش‌های بیزی برای درک بسیاری از الگوریتم‌های یادگیری ماشین اهمیت بسزایی دارد. ویژگی‌های متدهای یادگیری بیزی شامل موارد زیر است:

- هر نمونه‌ی آموزشی می‌تواند احتمال تخمینی اینکه فرضیه درست است را کم یا زیاد کند. این حقیقت باعث می‌شود که روش‌های بیزی نسبت به الگوریتم‌هایی که کاملاً فرضیه را با نمونه‌های غیر سازگار رد می‌کنند انعطاف‌تر پذیر تر باشند.
- می‌توان از دانش قبلی به همراه داده‌های مشاهده شده برای تعیین احتمال نهایی درستی فرضیه‌ها استفاده کرد. در یادگیری بیزی، دانش قبلی با (۱) احتمال اولیه‌ی هر فرضیه و (۲) توزیع احتمال روی داده‌های تعیین شده برای هر فرضیه‌ی ممکن تعیین می‌شود.
- متدهای بیزی می‌توانند برای فرضیه‌ها احتمالاتی را پیش‌بینی کنند (برای مثال، فرضیه‌ی "این بیمار ذات‌الریه با احتمال ۹۳٪ کاملاً بهبود خواهد یافت).
- نمونه‌های جدید را می‌توان با ترکیب پیش‌بینی‌های چندین فرضیه، (هر کدام با وزن احتمالشان) دسته‌بندی کرد.
- حتی هنگامی که اثبات می‌شود که متدهای بیزی محاسباتی غیرقابل‌پیش‌بینی انجام می‌دهند، با این حال معیار استانداردی برای دیگر متدهای عملی یادگیری مطرح می‌کنند.

یکی از مشکلات عملی کاربرد متدهای بیزی نیاز آن‌ها به داشتن دانش اولیه از بسیاری از احتمالات است. هنگامی که این اطلاعات به طور دقیق در دسترس نیست، گاهی آن‌ها را با استفاده از دانش قبلی، داده‌های موجود قبلی، و فرض‌هایی درباره‌ی فرم توزیع تخمین می‌زنیم. دومین مشکل عملی کاربرد این متدها هزینه‌ی محاسباتی قابل‌توجه آن‌ها برای تعیین فرضیه‌ی بهینه‌ی بیز در حالت کلی است (که رابطه خطی با تعداد فرضیه‌های ممکن دارد). در حالت‌های خاص خاصی این هزینه‌ی محاسباتی به طور قابل‌توجهی کاهش می‌یابد.

ادامه‌ی این فصل به شکل زیر ساختاربندی شده است. بخش ۶٫۲ قضیه‌ی بیز را معرفی کرده و محتمل‌ترین[۱] و فرضیه‌ای با حداکثر احتمال ثانویه[۲] را تعریف خواهد کرد. چهار زیر بخش این بخش این چارچوب[۳] را برای بررسی چندین مشکل و الگوریتم یادگیری که در فصل‌های گذشته مطرح شد به کار می‌برند. برای مثال، نشان می‌دهیم که چندین الگوریتم مطرح شده با چه فرض‌هایی محتمل‌ترین فرضیه را خروجی می‌دهند. بخش‌های بعدی تعدادی از الگوریتم‌های یادگیری که منحصر با احتمالات کار می‌کنند را معرفی خواهند کرد. این الگوریتم‌ها شامل دسته‌بندی کننده‌ی بهینه‌ی بیز، الگوریتم گیبز و دسته‌بندی کننده‌ی ساده‌ی بیز می‌شود. بالاخره درباره‌ی شبکه‌ی باور بیز[۴] بحث خواهیم کرد و روشی جدید برای یادگیری بر اساس استدلال احتمالی و الگوریتم EM که الگوریتمی پرکاربرد در یادگیری در حضور متغیرهای غیرقابل‌مشاهده است را بررسی خواهیم کرد.

---

۱ maximum likelihood

۲ maximum a posteriori probability hypotheses

۳ framework

۴ Bayesian belief network

۶,۲ قضیه‌ی بیز

در یادگیری ماشین، گاهی سعی داریم که از میان فضای فرضیه‌های H بهترین فرضیه‌ی سازگار با نمونه‌های آموزشی D را پیدا کنیم. چندین راه برای تعریف "بهترین" در این جمله وجود دارد، یکی از این تعاریف "محتمل‌ترین" است، با در دست داشتن داده‌های D بدون نیاز به هیچ اطلاعات اولیه‌ی دیگر نمی‌توان محتمل‌ترین فرضیه را انتخاب کرد. قضیه‌ی بیز متدی مستقیم برای محاسبه‌ی احتمالات فرضیه‌های موجود در H ارائه می‌کند. به عبارت دیگر، قضیه‌ی بیز روشی برای محاسبه‌ی احتمال یک فرضیه بر اساس احتمال قبلی‌اش، احتمال مشاهده‌ی داده‌های سازگار با فرض درستی این فرضیه و احتمال خود داده‌های مشاهده شده ارائه می‌کند.

برای تعریف دقیق قضیه‌ی بیز، ابتدا بیایید نشانه‌گذاری‌ها را معرفی کنیم. برای نشان دادن احتمال اولیه‌ی فرضیه‌ی h، احتمال قبل از مشاهده‌ی داده‌های آموزشی، از P(h) استفاده می‌کنیم. به P(h) احتمال اولیه‌ی[1] h نیز می‌گویند، این احتمال از اطلاعات قبلی‌ای که در مورد احتمال درستی فرضیه‌ی h داریم تأثیر می‌پذیرد. به طور مشابه از P(D) برای نمایش احتمال اولیه‌ی مشاهده‌ی نمونه‌های آموزشی D استفاده می‌کنیم (مثلاً احتمال مشاهده‌ی D بدون داشتن هیچ اطلاعات قبلی در مورد اینکه با چه فرضیه‌هایی سازگار است). برای نشان دادن احتمال مشاهده‌ی D در جایی که فرضیه‌ی h درست است از P(D|h) استفاده می‌کنیم. در حالت کلی، از p(x|y) برای نشان دادن احتمال x با فرض وقوع y استفاده می‌کنیم. در مسائل یادگیری ماشین، علاقه‌ی ما به احتمال P(h|D) است که در آن h یک فرضیه و D نمونه‌های آموزشی مشاهده شده هستند. به P(h|D) احتمال ثانویه‌ی[2] h نیز می‌گویند، زیرا که اطمینان ما به فرضیه‌ی h بعد از مشاهده‌ی نمونه‌های آموزشی D را نشان می‌دهد. توجه داشته باشید که احتمال ثانویه P(h|D) بر خلاف احتمال اولیه P(h) که از نمونه‌های آموزشی مستقل است، از نمونه‌های آموزشی D تأثیر می‌پذیرد.

قضیه بیز، اساس متدهای یادگیری بیز است زیرا که راهی برای محاسبه‌ی احتمال ثانویه P(h|D) از P(h)، P(D) و P(D|h).

قضیه‌ی بیز:

$$p(\mathrm{h}|\mathrm{D}) = \frac{p(\mathrm{D}|\mathrm{h})P(\mathrm{h})}{p(\mathrm{D})} \qquad (6.1)$$

همان طور که انتظار می‌رود، بر اساس قضیه‌ی بیز P(h|D) با افزایش P(h) و P(D|h) افزایش می‌یابد. همچنین منطقی است که P(h|D)، با افزایش P(D) کاهش بیابد، زیرا که هر چه که احتمال مشاهده‌ی D به طور مستقل از h بالاتر رود دیگر D مدرکی برای درستی h نخواهد بود.

در بسیاری از مسائل یادگیری، یادگیر مجموعه‌ی فرضیه‌هایی مثل H را در نظر می‌گیرد و در بین آن‌ها به دنبال محتمل‌ترین فرضیه‌ی $h \in H$ با توجه به نمونه‌های آموزشی D می‌گردد (یا حداقل یکی از محتمل‌ترین فرضیه‌ها). هر کدام از این محتمل‌ترین، فرضیه با حداکثر احتمال

[1] prior probability
[2] posterior probability

ثانویه[1] یا (MAP) نامیده می‌شود. فرضیه‌های MAP را می‌توان با استفاده از قضیه‌ی بیز برای محاسبه‌ی احتمال ثانویه‌ی هر فرضیه مشخص کرد. به صورت دقیق‌تر، زمانی می‌گوییم که فرضیه‌ی h_{MAP} یک فرضیه‌ی MAP است که

$$\begin{aligned} h_{MAP} &= \arg\max_{h\in H} P(h|D) \\ &= \arg\max_{h\in H} \frac{P(D|h)P(h)}{P(D)} \\ &= \arg\max_{h\in H} P(h|D)\,P(h) \qquad (6.2) \end{aligned}$$

توجه داشته باشید که مرحله آخر عبارات بالا P(D) چون ثابتی است و h بر آن تأثیری ندارد حذف می‌شود.

در بعضی موارد، فرض می‌کنیم که هر فرضیه در H احتمال اولیه‌ی مساوی‌ای دارد (برای هر h_j و h_i در H داریم که $P(h_i) = P(h_j)$. در این شرایط می‌توان رابطه‌ی ۶٫۲ را بیشتر ساده کرد و کافی است که فقط عبارت P(D|h) را برای پیدا کردن محتمل‌ترین فرضیه در نظر بگیریم. P(D|h) گاهی محتمل بودن داده‌های D برای h نیز نامیده می‌شود و هر فرضیه‌ای که P(D|h) را ماکزیمم کند (ML)[2] ، h_{ML} نامیده می‌شود.

$$h_{ML} \equiv \arg\max_{\mathrm{h}\in\mathrm{H}} P(D|h) \qquad (6.3)$$

برای مشخص شدن رابطه با مسائل یادگیری ماشین، ابتدا قضیه‌ی بیز را با توجه به نمونه‌های D و فضای فرضیه‌ای H معرفی کردیم. در واقع قضیه‌ی بیز کلی‌تر از آنچه در بالا گفته شد است. از قضیه‌ی بیز می‌توان برای هر زیرمجموعه‌ی H که ناسازگارند (اشتراک ندارند) استفاده کرد (مثل "آسمان آبی است" و "آسمان آبی نیست"). در این فصل، در اکثر موارد فرض خواهیم کرد که H فضای فرضیه‌ای که تابع هدف را شامل می‌شود است و D نمونه‌های آموزشی هستند. در مواقع دیگر فرض می‌کنیم که H مجموعه‌ی دیگر ناسازگاری با یکدیگر از فرضیه‌هاست و D نیز مجموعه‌ی دیگری از داده‌هاست.

۶٫۲٫۱ یک مثال

برای تصور قانون بیز، فرض کنید که مسئله‌ای برای تشخیص بیماری داریم، دو فرضیه‌ی ممکن برای بیماری وجود دارد: (۱) بیمار نوع خاصی از سرطان دارد و (۲) بیمار آن نوع سرطان را ندارد. داده‌های موجود یک تست آزمایشگاهی است که دو خروجی ممکن دارد: $\oplus$ (مثبت) و $\ominus$ (منفی). دانش قبلی داریم در کل جمعیت حاضر در آزمایش فقط 008. این بیماری را دارند. علاوه بر آن نتیجه‌ی آزمایش همیشه قطعی نیست و احتمال خطا وجود دارد. تست آزمایشگاهی در ۹۸٪ مواردی که بیمار بیماری را دارد نتیجه‌ی مثبت درست می‌دهد و در ۹۷٪ مواردی که بیمار بیماری را ندارد نتیجه‌ی منفی درست می‌دهد. در بقیه‌ی موارد آزمایش نتیجه‌ی اشتباه می‌دهد. آنچه در بالا گفته شد را خلاصه‌وار می‌توان به صورت زیر نشان داد:

$$P(cancer) = .008, \qquad P(\neg cancer) = .992$$

[1] Maximum A Posteriori
[2] maximum likelihood

$$P(\oplus|cancer) = .98, \quad P(\ominus|\neg cancer) = .02$$

$$P(\oplus|\neg cancer) = .03, \quad P(\ominus|\neg cancer) = .97$$

فرض کنید که بیمار جدیدی پذیرش می‌شود و نتیجه‌ی آزمایشش مثبت است. حال با چه احتمالی می‌توان گفت که بیمار سرطان دارد؟ فرضیه با حداکثر احتمال را می‌توان از رابطه‌ی ۶٫۲ پیدا کرد:

$$P(\oplus|cancer)P(cancer) = (.98).008 = .0078$$

$$P(\oplus|\neg cancer)P(\neg cancer) = (.03).992 = .0298$$

پس، داریم $h_{MAP} = \neg cancer$. احتمال ثانویه‌ی فرضیه‌ها را می‌توان با رساندن مجموع دو احتمال به ۱ پیدا کرد ($P(\oplus|cancer)P(cancer) =.\frac{0078}{.0078+.0298} = .21$). این مرحله برای این درست است که قضیه‌ی بیز احتمال‌های ثانویه مجموعه‌ی تمامی داده را بدون اشتراک می‌پوشانند (افراز می‌کنند) بیان می‌کند. با وجود اینکه $P(\oplus)$ مستقیماً توسط مسئله داده نشده است، اما می‌توان آن را محاسبه کرد زیرا که می‌دانیم مجموع دو احتمال $P(cancer|\oplus)$ و $P(\neg cancer|\oplus)$ یک است (هر بیمار یا سرطان دارد یا سرطان ندارد). توجه داشته باشید که احتمال ثانویه‌ی سرطان نسبت به احتمال اولیه‌ی آن به طور قابل‌توجهی زیادتر است، اما با این حال محتمل‌ترین فرضیه این است که بیمار سرطان ندارد.

همان طور که در مثال بالا نشان داده شد، نتیجه‌ی تأثیر بیز به شدت به احتمال اولیه وابسته است، برای اینکه بتوان قضیه را به طور مستقیم به کار برد باید احتمالات اولیه معلوم باشند. توجه داشته باشید که در این مثال فرضیه‌ها کاملاً پذیرفته شده یا رد شده نیستند بلکه هر کدام با افزایش داده‌های مشاهده شده احتمالی پیدا می‌کنند. فرمول اصلی محاسبه‌ی احتمالات در جدول ۶٫۱ خلاصه شده است.

٦٫٣ قضیه‌ی بیز و یادگیری مفهوم

ارتباط بین قضیه‌ی بیز و مسائل یادگیری مفهوم چیست؟ از آنجایی که قضیه‌ی بیز راهی اصولی برای محاسبه‌ی احتمالات ثانویه‌ی هر یک از فرضیه‌ها بعد از مشاهده‌ی داده‌های آموزشی ارائه می‌کند، می‌توانیم از آن برای پایه‌ی یک الگوریتم یادگیری ساده استفاده کنیم، الگوریتمی که احتمال هر یک از فرضیه‌ها را محاسبه کرده و محتمل‌ترین فرضیه‌ها را خروجی می‌دهد. در این بخش چنین الگوریتم‌های بدون شعور[۱] یادگیری مفهوم بیز را بررسی و با الگوریتم‌های یادگیری مفهوم مقایسه می‌کنیم. همان طور که بعداً نیز خواهیم دید، یکی از نتایج جالب این مقایسه این است که تحت شرایط خاصی چندین الگوریتمی که در فصل‌های گذشته بررسی شدند همان فرضیه‌ای که یادگیری بدون شعور بیز خروجی می‌دهد را خروجی می‌دهند، با این تفاوت که آن‌ها احتمالات فرضیه‌ها را مشخص نمی‌کنند و فقط محتمل‌ترین را مشخص می‌کنند.

- قانون ضرب[۲]: احتمال $P(A \wedge B)$ که احتمال عطف دو اتفاق A و B است را محاسبه کن

$P(A \wedge B)=P(A|B)P(B)=P(B|A)P(A)$

- قانون جمع[۱]: احتمال فصل دو اتفاق A و B را محاسبه کن

[۱] brute-force
[۲] Product rule

P(A⋁B)=P(A)+P(B)-P(A⋀B)

- قضیه بیز[۲]: احتمال ثانویه‌ی P(h|D) را محاسبه کن

$$P(h|D) = \frac{P(D|h)P(h)}{P(D)}$$

- قضیه‌ی مجموع احتمالات[۳]: اگر اتفاق‌های $A_1, \dots, A_n$ دوبه‌دو ناسازگار باشند و $\sum_{i=1}^{n} P(A_i) = 1$ خواهیم داشت

$$P(B) = \sum_{i=1}^{n} P(B|A_i)P(A_i)$$

جدول ۶٫۱ خلاصه‌ی فرمول‌های پایه‌ای احتمال.

۶٫۳٫۱ یادگیری مفهوم بدون شعور بیز

مسئله‌ی یادگیری مفهومی را که در ابتدای فصل ۲ معرفی شد را در نظر بگیرید. در کل، فرض کنید که یادگیر فضای فرضیه‌ای محدود H را که شامل فرضیه‌هایی که بر فضای نمونه‌ای X تعریف شده‌اند است در نظر می‌گیرد و تابع هدف نیز مفهومی به فرم c:X→{0,1} است. مثل معمول، فرض می‌کنیم که به یادگیر دسته‌ای از نمونه‌های آموزشی مثل $\{< x_1, d_1 > \cdots < x_m, d_m >\}$ داده می‌شود، در این نمونه‌های آموزشی x_i عضوی از X و d_i نیز مقدار هدف برای آن x_i است $(d_i = c(x_i))$. برای ساده‌سازی بحث در این بخش، فرض می‌کنیم که ترتیب نمونه‌های نمونه‌ها $< x_1 \dots x_m >$ ثابت است پس می‌توان نمونه‌های آموزشی D را به فرم ساده به صورت $D =< d_1 \dots d_m >$ نوشت. می‌توان نشان داد که این ساده‌نویسی تأثیری بر نتایج به دست آمده از این قسمت ندارد (تمرین ۶٫۴).

می‌توان با استفاده از قضیه‌ی بیز الگوریتم یادگیری مفهوم مستقیمی طراحی کرد که فرضیه با حداکثر احتمال ثانویه را خروجی دهد:

الگوریتم یادگیری بدون شعور MAP [۴]

۱. برای هر فرضیه h در H احتمال ثانویه را محاسبه کن،

$$\mathrm{p(h|D)} = \frac{\mathrm{p(D|h)P(h)}}{\mathrm{p(D)}}$$

۲. فرضیه h_{MAP} را که بیشترین احتمال ثانویه را دارد خروجی بده

$$h_{MAP} = \arg\max_{h \in H} P(h|D)$$

[۱] Sum rule

[۲] Bayes theorem

[۳] Theorem of total probabily

[۴] Brute-Force MAP Learning

این الگوریتم ممکن است محاسبات قابل‌توجهی نیاز داشته باشد، زیرا که قانون بیز را برای تمامی فرضیه‌های H برای محاسبه‌ی P(h|D) بـه کار می‌برد. چنین حجم محاسباتی‌ای برای فضای فرضیه‌هایی با اندازه‌ی بالا غیرعملی است، با این وجود الگوریتم هنوز مورد توجه است زیرا که معیاری ارائه می‌کند در حالی که دیگر الگوریتم‌های یادگیری مفهوم هیچ معیاری ارائه نمی‌کنند.

برای آماده‌سازی یک مسئله برای حل با الگوریتم یادگیری بدون شعور MAP لازم است که مقادیر P(h) و P(D|h) را مشخص کنیم (همان طور که بعداً هم خواهیم دید با مشخص کردن مقادیر ذکر شده مقدار P(D) نیز مشخص می‌شود). اطلاعات اولیه‌ی ما در مورد مسئله بـا تعیـین دو توزیع P(h) و P(D|h) به طور دلخواه مشخص می‌شود. بیایید ابتدا با فرض‌های زیر شروع کنیم :

۱. نمونه‌های آموزشی D خطا ندارند ($d_i = c(x_i)$)
۲. مفهوم هدف c در فضای فرضیه‌ای H موجود است.
۳. هیچ مدرکی بر برتری یک فرضیه بر فرضیه‌ی دیگر وجود ندارد.

با فرض‌های بالا، چه مقداری باید برای P(h) تعیین شود؟ بدون هیچ اطلاعات قبلی، برتری فرضیه‌ها بر یکدیگر بی‌دلیل خواهد بود، مـی‌تـوانیم احتمال تمامی آن‌ها را مساوی قرار دهیم. علاوه بر آن، چون فرض کرده‌ایم تابع هدف c در H موجود است باید طوری احتمـال را پخـش کنـیم که مجموع احتمال کل H یک باشد. پس خواهیم داشت:

$$P(h) = \frac{1}{|H|} \ \ for\ all\ h\ in\ H$$

اما P(D|h) چه احتمالی باید داشته باشد؟ P(D|h) احتمال مشاهده‌ی مقادیر هدف $D =< d_1 \ldots d_n >$ است که بـرای دسـته‌ی ثابـت نمونه‌ها زمانی که h درست است می‌باشد.(مثلاً زمانی که h همان مفهوم هدف c است). از آنجایی که فـرض کـردیم داده‌هـای آموزشـی خطـا ندارند، احتمال دیدن d_i اگر $d_i = h(x_i)$ باشد ۱ و اگر $d_i \neq h(x_i)$ باشد۰ است. بنابراین،

$$P(D|h) = \begin{cases} 1\ \ if\ d_i = h(x_i)\ \ for\ all\ d_i\ in\ D \\ 0\ \ otherwise \end{cases} \qquad (6.4)$$

به عبارت دیگر احتمال مشاهده‌ی D با داشتن h، ۱ است اگر D با h سازگار باشد و در غیر این صورت ۰ است.

با این نوع انتخاب P(h) و P(D|h) حال مسئله را کاملاً برای الگوریتم یادگیری بدون شعور MAP آماده کرده‌ایم. مرحله‌ی اول این الگـوریتم که در آن با استفاده از قضیه‌ی بیز احتمال ثانویه‌ی P(h|D) برای تمامی hها با توجه به نمونه‌های آموزشی D محاسـبه مـی‌شـود را در نظـر بگیرید. با توجه به قضیه‌ی بیز داریم،

$$\mathrm{p(h|D)} = \frac{\mathrm{p(D|h)P(h)}}{\mathrm{p(D)}}$$

ابتدا فرض کنید که h با نمونه‌های آموزشی ناسازگار است. از رابطه‌ی ۶,۴ داریم که P(D|h) صفر است زیرا که h با D ناسـازگار اسـت پـس داریم که:

$$P(h|D) = \frac{0 \cdot P(h)}{P(D)} = 0\ if\ \ h\ is\ inconsistent\ with\ D$$

پس احتمال ثانویه‌ی فرضیه‌ی ناسازگار با D صفر خواهد بود.

حال فرض کنید که فرضیه‌ی h با D سازگار است. از رابطه‌ی ۶,۴ داریم که P(D|h) یک فرض شده است زیـرا کـه h بـا D سـازگار اسـت. داریم،

$$P(h|D) = \frac{1 \cdot \frac{1}{|H|}}{P(D)}$$

$$= \frac{1 \cdot \frac{1}{|H|}}{\frac{|VS_{H,D}|}{|H|}}$$

$$= \frac{1}{|VS_{H,D}|} \; if\ h\ is\ consistent\ with\ D$$

در این رابطه $VS_{H,D}$ زیرمجموعه‌ای از H است که با D سازگار است (مثلاً $VS_{H,D}$ می‌تواند همان فضای ویژه‌ای فصل ۲ باشد که با توجه به D به دست آمده). تشخیص اینکه $P(D) = \frac{VS_{H,D}}{|H|}$ کار ساده‌ای است زیرا که مجموع P(h|D) بـرای تمـامی فرضـیه‌ها بایـد ۱ باشـد و از طرفی تعداد کل فرضیه‌های سازگار با D در H طبق تعریف $|VS_{H,D}|$ است. می‌توان مقدار P(D) را از قضـیه‌ی مجمـوع احتمـال (در جـدول ۶,۱) و این حقیقت که فرضیه‌ها دوبه‌دو ناسازگارند $((\forall i \neq j)(P(h_i \wedge h_j) = 0))$ به دست آورد.

$$P(D) = \sum_{h_i \in H} P(D|h_i)P(h_i)$$

$$= \sum_{h_i \in VS_{H,D}} 1 \cdot \frac{1}{|H|} + \sum_{h_i \notin VS_{H,D}} 0 \cdot \frac{1}{|H|}$$

$$= \sum_{h_i \in VS_{H,D}} 1 \cdot \frac{1}{|H|}$$

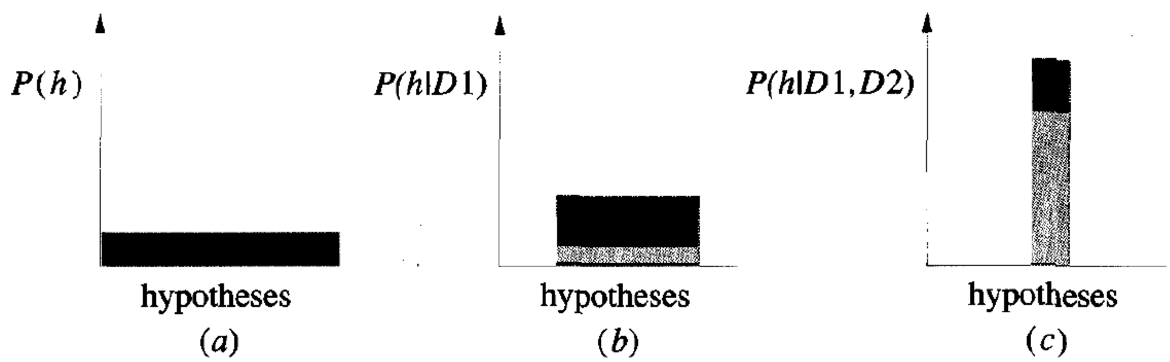
$$= \frac{|VS_{H,D}|}{|H|}$$

به طور خلاصه اینکه با فرض‌هایی که در مورد P(h) و P(D|h) کردیم قضیه‌ی بیز ایجاب می‌کند که P(h|D) به صورت زیر باشد:

$$P(h|D) = \begin{cases} \frac{1}{|VS_{H,D}|} \; if\ h\ is\ consistent\ with\ D \\ 0\ otherwise \end{cases} \qquad (6.5)$$

در این رابطه $|VS_{H,D}|$ تعداد فرضیه‌های H که با D سازگارند است. شکل ۶٫۱ سیر تکامل احتمالات را با نمودار نشان می‌دهد. ابتدا (شکل 6.1 (a)) تمامی فرضیه‌ها احتمال یکسانی دارند. با افزایش داده‌های آموزشی (شکل‌های 6.1 (b) و 6.1 (c)) احتمال ثانویه‌ی فرضیه‌های ناسازگار صفر می‌شود اما مجموع کل احتمالات ۱ باقی می‌ماند، یعنی احتمال فرضیه‌هایی که صفر می‌شود به طور مساوی در بین فرضیه‌های دیگر تقسیم می‌شود.

بررسی بالا نشان داد با انتخاب P(h) و P(D|h) تمامی فرضیه‌های سازگار احتمال ثانویه‌ی مساوی ($1/|VS_{H,D}|$) خواهند داشت و احتمال فرضیه‌های ناسازگار صفر خواهد شد. پس با توجه به این بررسی هر فرضیه‌ی سازگار یک MAP (فرضیه با حداکثر احتمال) است.



شکل ۶٫۱ تکامل احتمال ثانویه‌ی P(h|D) با افزایش داده‌های آموزشی.
(a) اولویت یکسان به تمامی فرضیه‌ها داده می‌شود. با افزایش داده‌ها به D1 (b) و سپس به D1⋀D2 (c)، احتمال ثانویه‌ی فرضیه‌های ناسازگار به صفر می‌رسد در حالی که احتمال ثانویه برای فرضیه‌های فضای ویژه افزایش می‌یابد.

۶٫۳٫۲ فرضیه‌های MAP و یادگیرهای سازگار

بررسی‌های بالا نشان می‌دهد که با مفروضات مذکور تمامی فرضیه‌های سازگار با D فرضیه‌ای MAP هستند. این عبارت را می‌توان مستقیماً به عبارتی جالب در مورد دسته‌ای از یادگیرها که یادگیرهای سازگار[1] می‌نامیم تفسیر کرد. زمانی می‌گوییم که یک الگوریتم یادگیری یادگیر سازگار است که فرضیه‌ی خروجی هیچ خطایی بر روی داده‌های آموزشی نداشته باشد. بر اساس بررسی بالا، می‌توان گفت تمامی یادگیرهای سازگار فرضیه‌ی خروجی‌شان یک فرضیه‌ی MAP است، به شرطی که فرض کنیم که توزیع اولیه احتمال روی H یکنواخت باشد ($(\forall i,j)P(h_i) = P(h_j)$) و همچنین فرض کنیم که داده‌های آموزشی قطعی و بدون خطا هستند (P(D|H)=1 اگر D با h سازگار باشد و در غیر این صورت صفر است).

برای مثال، الگوریتم یادگیری مفهوم Find-S را که در فصل ۲ بررسی شد را در نظر بگیرید. Find-S فضای فرضیه‌ای H را از فرضیه‌های جزئی‌تر به کلی‌تر جستجو می‌کند تا جزئی‌ترین فرضیه‌ی سازگار را پیدا کند (جزئی‌ترین عضو فضای ویژه). چون Find-S فرضیه‌ای سازگار را خروجی می‌دهد پس طبق احتمالات مفروض بالا برای P(h) و P(D|h) فرضیه‌ای MAP را خروجی خواهد داد. البته Find-S هیچ احتمالی

[1] consistent learner

را محاسبه و ارائه نمی‌کند و فقط خاص‌ترین فرضیه‌ی فضای ویژه را پیدا می‌کند. با این وجود، با مشخص کردن توزیع‌های P(h) و P(D|h) به صورتی که فرضیه‌ی خروجی MAP باشد، روشی مفید برای مشخص کردن رفتار Find-S داریم.

آیا توزیع احتمال‌های دیگری برای P(h) و P(D|h) وجود دارد که خروجی FIND-S فرضیه‌ی MAP باشد؟ بله، چون FIND-S خاص‌ترین فرضیه‌ی فضای ویژه را پیدا می‌کند فرضیه‌ی خروجی‌اش با اختصاص توزیع احتمال‌هایی که به سمت فرضیه‌های خاص‌تر تمایل دارند MAP خواهد بود. به عبارت دقیق‌تر، فرض کنید که $\mathcal{H}$ تمامی توزیع احتمال P(h) روی H است که در آن‌ها داریم $P(h_1) \geq P(h_2)$ اگر h_1 خاص‌تر از h_2 باشد. می‌توان نشان داد با چنین توزیع احتمال‌هایی و توزیع احتمال مذکور برای P(D|h) فرضیه‌ی خروجی FINS-S یک فرضیه‌ی MAP خواهد بود.

خلاصه بحث بالا بدین شکل است، چارچوب بیزی به ما اجازه می‌دهد تا ویژگی‌های رفتاری الگوریتم‌های یادگیری (حتی الگوریتم‌هایی که مقدار احتمال فرضیه را مشخص نمی‌کنند، مثل FIND-S) را مشخص کنیم. با مشخص کردن توزیع احتمال‌های P(h) و P(D|h) به صورتی که فرضیه‌ی خروجی الگوریتم بهینه، MAP، شود، پیش‌فرض‌هایی که الگوریتم برای نتیجه‌گیری انجام می‌دهد را می‌توان پیدا کرد.

استفاده از دیدگاه بیزی برای بررسی ویژگی‌های الگوریتم‌های یادگیری بدین صورت عملاً مشابه بررسی بایاس استقرایی یادگیرهاست. در فصل ۲ ما بایاس استقرایی یک الگوریتم را دسته پیش‌فرض‌هایی مثل B تعریف کردیم که نحوه‌ی استقرای یادگیر را توجیه می‌کند. برای مثال، گفته شد که بایاس استقرایی الگوریتم Candidate-Elimination وجود مفهوم هدف c در مجموعه‌ی فرضیه‌ای H است. علاوه بر آن نشان دادیم که خروجی این الگوریتم یادگیری را می‌توان از ورودی‌هایش و این پیش‌فرض استقرایی ضمنی نتیجه گرفت. تفسیری بیزی بالا می‌تواند جایگزینی برای بررسی ویژگی‌های این پیش‌فرض‌های الگوریتم‌های یادگیری باشد. با این تفاوت که در اینجا به جای مدل کردن الگوریتم با یک سیستم معادل استقرایی، الگوریتم را با سیستم معادل استدلال احتمالی[۲] که بر اساس قضیه‌ی بیز کار می‌کند، مدل‌سازی می‌کنیم. و در اینجا پیش‌فرض‌هایی که یادگیر فرض می‌کند به فرم "احتمال اولیه‌های فرضیه‌ها P(h) و قدرت داده‌ها در قبول یا رد فرضیه‌ها P(D|h)" است. تعریف P(h) و P(D|h) که در این قسمت معرفی شد مربوط به دو الگوریتم Candidate-Elimination و FIND-S بود. سیستم استدلال احتمالی‌ای که بر اساس قضیه‌ی بیز کار می‌کند، با این توزیع‌ها در ورودی و خروجی رفتاری مشابه این الگوریتم‌ها از خود نشان خواهد داد.

بحثی که در این بخش انجام شد حالت خاصی از استدلال بیزی بود زیرا که فرض کردیم داده‌های آموزشی بدون خطایند و فرضیه‌ها نیز قطعی‌اند، یعنی P(D|h) حتماً یکی از دو مقدار ۱ یا ۰ را دارد. همان طور که در قسمت بعدی نیز خواهیم دید، می‌توان یادگیری از نمونه‌های آموزشی خطادار را شبیه‌سازی کرد، فقط کافی است که مقدار P(D|h) مقادیری غیر ۰ و ۱ را نیز داشته باشد، با این تغییر توزیع احتمال P(D|h) خطا را کنترل خواهد کرد.

---

[۲] probabilistic reasoning system

۶,۴ محتمل‌ترین فرضیه‌ها[3] و فرضیه‌هایی که کمترین خطای مربعی[4] را دارند

همان طور که در بالا نیز نشان داده شد تحت شرایطی یک الگوریتم یادگیری فرضیه‌های MAP را خروجی می‌دهد حتی اگر این الگوریتم از روش بیز یا حتی از محاسبه‌ی احتمال‌ها استفاده نکند.

در این بخش، به مسئله‌ای یادگیری توابع هدف پیوسته مقدار می‌پردازیم، مسئله‌ای که راه‌های زیادی برای آن مثل شبکه‌های عصبی، تقریب خطی، و تقریب چندجمله‌ای ارائه شده است. یک بررسی مستقیم بیزی نشان می‌دهد که در شرایط خاصی هر الگوریتم یادگیری که خطای مربعی بین تخمین و خروجی داده‌های آموزشی را مینیمم کند یک محتمل‌ترین فرضیه[5] را خروجی می‌دهد. اهمیت این نتیجه در استفاده از این استدلال بیزی (تحت شرایط خاص) برای توجیه بسیاری از شبکه‌های عصبی و دیگر متدهایی که مجموع خطای مربعی را مینیمم می‌کنند است.

شرایط مسئله‌ی یادگیری تابع هدف پیوسته را در نظر بگیرید، یادگیر L که از فضای نمونه‌ای X و فضای فرضیه‌ای H که مجموعه‌ای از توابع حقیقی مقدار روی نمونه‌های X استفاده می‌کند (هر h در H تابعی است به فرم $h: X \rightarrow \mathfrak{R}$، که در آن $\mathfrak{R}$ مجموعه‌ی اعداد حقیقی است). مسئله‌ای که یادگیر L با آن مواجه است یادگیری تابع هدف مجهول $f: X \rightarrow \mathfrak{R}$ از مجموعه‌ی فرضیه‌های H است. مجموعه‌ای از m نمونه‌ی آموزشی در دسترس است، در این مجموعه مقدار تابع هدف هر یک از نمونه‌ها با یک مقدار تصادفی خطا که توزیع نرمال دارد معلوم است. به عبارت دقیق‌تر، هر نمونه‌ی آموزشی زوج مرتبی به فرم $< x_i, d_i >$ است که در آن $d_i = f(x_i) + e_i$. در اینجا $f(x_i)$ خود تابع هدف و مقدار e_i متغیر تصادفی خطاست. فرض می‌شود که مقدار e_i مستقل و دارای توزیع نرمال با میانگین صفر است. هدف یادگیر نیز پیدا کردن محتمل‌ترین فرضیه، یا به صورت معادل، یک فرضیه‌ی MAP است با این فرض که تمامی فرضیه‌ها احتمال اولیه‌ی یکسانی دارند.

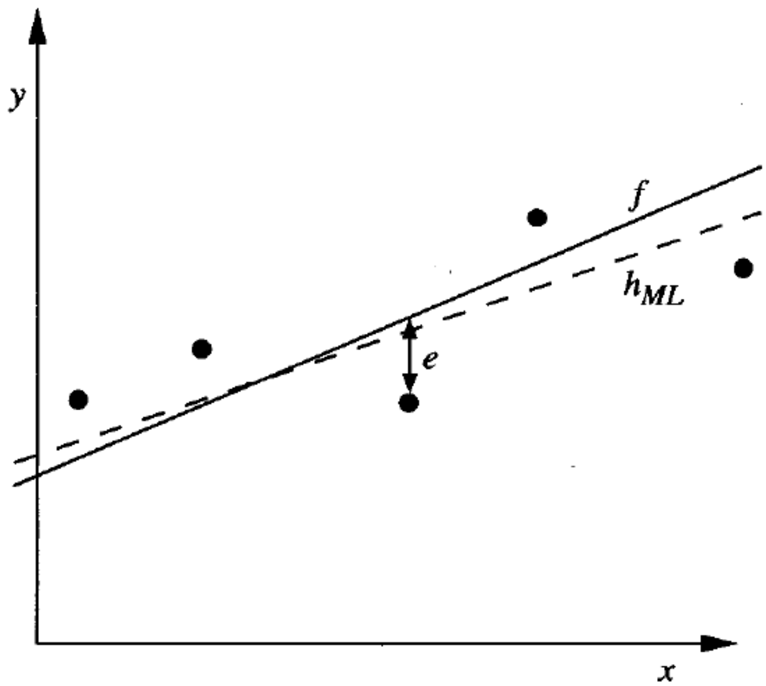
با وجود اینکه بررسی‌هایمان را برای یادگیری توابع دلخواه حقیقی مقدار انجام می‌دهیم، مسئله‌ی یادگیری تابع خطی نمونه‌ای از چنین مسائلی است. شکل ۶,۲ شکل تابع هدف خطی f را به همراه چندین نمونه‌ی آموزشی نشان داده است. خط‌چین فرضیه‌ی h_{ML} است که کمترین خطای مربعی را دارد، پس محتمل‌ترین فرضیه است. توجه داشته باشید که محتمل‌ترین فرضیه حتماً فرضیه‌ی درست نیست، زیرا که مجموعه‌های آموزشی محدود و خطادار هستند.

---

[3] maximum likelihood

[4] least squared error

[5] maximum likelihood



شکل ۶٫۲ یادگیری تابع حقیقی مقدار.
تابع هدف f با خط نشان داده شده است. نمونه‌های آموزشی $< x_i, d_i >$ با فرض اینکه خطایی با توزیع نرمال با میانگین صفر دارند در نظر گرفته شده‌اند. خط‌چین تابعی خطی را نشان می‌دهد که میزان خطای مربعی را مینیمم می‌کند. بنابراین این فرضیه محتمل‌ترین فرضیه، h_{ML}، بر اساس ۵ نمونه‌ی آموزشی موجود است.

قبل از اینکه به اثبات محتمل‌ترین بودن فرضیه‌هایی که خطای مربعی را مینیمم می‌کنند در شرایط مذکور بپردازیم؛ ابتدا بیایید دو مفهوم را از تئوری احتمال مرور کنیم: چگالی احتمال و توزیع نرمال. ابتدا برای بحث روی متغیرهای تصادفی پیوسته مثل e، ابتدا باید چگالی احتمال را معرفی کنیم. دلیل اولیه این پیش‌زمینه‌ها این است که می‌خواهیم مجموع احتمالات روی تمامی مقادیر ممکن متغیر تصادفی یک باشد. در این حالت که متغیرهای تصادفی پیوسته هستند، تعیین احتمال را نمی‌توان با نسبت دادن یک احتمال به هر یک از مقادیر ممکن متغیر تصادفی انجام داد. به جای آن، از چگالی احتمال[۶] برای مقادیر تصادفی حقیقی مثل e استفاده می‌کنیم و انتگرال روی کل چگالی احتمال را مساوی یک قرار می‌دهیم. در کل از حرف کوچک p برای نشان دادن تابع چگالی احتمال استفاده می‌کنیم و احتمال را با حرف بزرگ P نشان می‌دهیم (گاهی اوقات این مقدار جرم احتمال[۷] نیز نامیده می‌شود). چگالی احتمال $p(x_0)$ ، $\frac{1}{\varepsilon}$ برابر مقدار احتمال اینکه متغیر تصادفی در بازه‌ای $[x_0, x_0 + \varepsilon)$ زمانی که $\varepsilon \to 0$ قرار بگیرد است.

چگالی احتمال:

$$p(x_0) \equiv \lim_{\varepsilon \to 0} \frac{1}{\varepsilon} P(x_0 \leq x < x_0 + \varepsilon)$$

---

[۶] probability density
[۷] probability mass

دوم اینکه e را در مسئله طوری تعریف کردیم که از توزیع احتمال نرمال پیروی می‌کند. توزیع احتمال نرمـال، توزیـع احتمـالی همـوار و زنگـی شکل است که می‌توان آن را با میانگین μ و انحراف معیار σ کاملاً مشخص کرد. برای تعریف دقیق‌تر به جدول ۵,۴ مراجعه کنید.

حال با داشتن این پیش‌زمینه‌ها می‌توانیم به موضوع اصلی بر گردیم: در شرایط مذکور، نشان می‌دهـیم کـه فرضـیه‌هایی کـه خطـای مربعـی را مینیمم می‌کنند در واقع همان محتمل‌ترین فرضیه‌ها هستند. ابتدا محتمل‌ترین تابع را با استفاده از رابطه‌ی 6.3 مشخص می‌کنیم، با این تفاوت که توزیع احتمال در این رابطه را با p نشان می‌دهیم.

$$h_{ML} = \arg\max_{h\in H} p(D|h)$$

مثل قبل، فرض می‌کنیم که مجموعه‌ای از نمونه‌های آموزشی مثل $< x_1 \dots x_n >$ با مقـدار تـابع هدفشـان D، $D = < d_1 \dots d_2 >$ داریم. در اینجا $d_i = f(x_i) + e_i$. با فرض اینکه نمونه‌های آموزشی کاملاً مستقل از فرضیه‌ی h هستند P(D|h) را می‌توان بـر حسـب $p(d_i|h)$ ها نوشت

$$h_{ML} = \arg\max_{h\in H} \prod_{i=1}^{m} p(d_i|h)$$

با دانستن اینکه e_i ها از توزیع نرمال با میانگین صفر و واریانس مجهول σ^2 پیروی می‌کنند، هر d_i نیز باید از توزیع نرمالی با واریانس σ^2 و میانگین $f(x_i)$، به جای صفر، پیروی کند. بنابراین $p(d_i|h)$ را می‌توان به صورت توزیع نرمالی با واریانس σ^2 و میـانگین $\mu = f(x_i)$ نوشت. بیایید فرمول این توزیع نرمال را که $p(d_i|h)$ را توصیف می‌کند بنویسیم، ابتدا فرمولی که در جدول ۵,۴ آمده را می‌نویسـیم و مقـادیر μ و σ^2 را جایگزین می‌کنیم. چون رابطه‌ای برای d_i با فرض اینکه فرضیه‌ی h توصیف درست از تابع هـدف f اسـت مـی‌نویسـیم، خـواهیم داشت که $\mu = f(x_i) = h(x_i)$

$$h_{ML} = \arg\max_{h\in H} \prod_{i=1}^{m} \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{1}{2\sigma^2}(d_i-\mu)^2}$$

$$= \arg\max_{h\in H} \prod_{i=1}^{m} \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{1}{2\sigma^2}(d_i-h(x_i))^2}$$

حال از تبدیلی استفاده می‌کنیم که در اکثر محاسبات محتمل‌ترین‌ها متداول است: به جای ماکزیمم کردن مقدار کل عبارت، لگـاریتم آن را کـه بسیار ساده‌تر است ماکزیمم می‌کنیم. زیرا که تابع ln p تابعی یکنواخت و صعودی از p است. بنابراین مـاکزیمم کـردن ln p باعـث مـاکزیمم شدن خود p می‌شود.

$$h_{ML} = \arg\max_{h\in H} \sum_{i=1}^{m} ln\frac{1}{\sqrt{2\pi\sigma^2}} - \frac{1}{2\sigma^2}\left(d_i - h(x_i)\right)^2$$

جمله‌ی اول مستقل از h است و بنابراین می‌توان آن را حذف کرد،

$$h_{ML} = \arg\max_{h \in H} \sum_{i=1}^{m} -\frac{1}{2\sigma^2}\left(d_i - h(x_i)\right)^2$$

ماکزیمم کردن این کمیت منفی مشابه مینیمم کردن مقدار مثبت آن است،

$$h_{ML} = \arg\min_{h \in H} \sum_{i=1}^{m} \frac{1}{2\sigma^2}\left(d_i - h(x_i)\right)^2$$

و دوباره می‌توان ثابتی که مستقل از h است را حذف کرد و داریم:

$$h_{ML} = \arg\min_{h \in H} \sum_{i=1}^{m} \left(d_i - h(x_i)\right)^2 \qquad (6.6)$$

رابطه‌ی ۶٫۶ نشان می‌دهد که محتمل‌ترین فرضیه h_{ML} فرضیه‌ای است که مجموع خطاهای مربعی بین مقادیر هدف نمونه‌های آموزشی d_i و پیش‌بینی فرضیه $h(x_i)$ را مینیمم کند. این نتیجه‌گیری‌ها با این فرض بود که نمونه‌های آموزشی، d_i ها، مقادیر تابع هدف به اضافه‌ی مقدار خطای تصادفی با توزیع نرمال و میانگین صفر هستند. همان طور که استخراج عبارت بالا نیز نشان می‌دهد، مقدار جمله‌ی مربعی خطای $\left(d_i - h(x_i)\right)^2$ مستقیماً از توزیع نرمال ناشی شده است. با استفاده از دیگر توزیع‌های خطا می‌توان تعریف‌های دیگری برای خطا به دست آورد.

توجه داشته باشید که ساختار اشتقاق بالا شامل انتخاب فرضیه‌ای که لگاریتم محتمل بودن (ln p(D|h)) را حداکثر می‌کند به عنوان محتمل‌ترین فرضیه نیز می‌شود. همان طور که پیش‌تر نیز گفته شد، این مشابه این است که محتمل بودن (ln p(D|h)) را حداکثر کنیم. این روش کار با لگاریتم محتمل بودن[۸] در بسیار از بررسی‌های بیزی مورد استفاده قرار می‌گیرد، زیرا که کار با لگاریتم محتمل بودن بسیار ساده‌تر از کار با خود محتمل بودن است. البته، همان طور که قبلاً هم گفته شد، محتمل‌ترین فرضیه همیشه فرضیه‌ی *MAP* نیست مگر اینکه احتمال اولیه‌ی تمامی فرضیه‌ها مساوی فرض شود.

چرا استفاده از توزیع نرمال برای مدل‌سازی نویز یا همان خطای نمونه‌ها استفاده می‌کنیم؟ یکی از دلایلی که لازم است حتماً ذکر شود، این است که بررسی را از نظر ریاضی بسیار ساده‌تر می‌کند. دلیل دوم این است که این توزیع توزیعی هموار است و توزیع‌های زنگی شکل تخمین خوبی برای بسیاری از انواع خطاها در سیستم‌های فیزیکی هستند. در واقع، طبق قضیه‌ی حد مرکزی که در فصل ۵ توضیح داده شد، مجموع تعداد زیادی از متغیرهای مستقل و هم توزیع بدون توجه به نوع توزیع از توزیع نرمال پیروی می‌کند. این ثابت می‌کند که خطا که خود از مجموع تعداد زیادی متغیر مستقل و با ضریب توزیع یکسان تولید می‌شود از توزیع نرمال پیروی خواهد می‌کند. البته در واقعیت، مؤلفه‌های مختلفی که در نویز تأثیرگذارند همگی از یک توزیع پیروی نمی‌کنند، که در این شرایط این قضیه توجیهی برای استفاده از این توزیع نیست.

---

[۸] likelihood

مینیمم کردن مجموع خطای مربعی روشی متداول در بسیاری از شبکه‌های عصبی، منحنی‌های تخمین و ... در تخمین توابع حقیقی مقدار است. فصل ۴ روش شیب نزول را که مینیمم کردن خطای مربعی در شبکه‌های عصبی را با آن انجام می‌دهیم مفصلاً توضیح داده است.

بد نیست که قبل از اتمام بحث رابطه‌ی بین محتمل‌ترین فرضیه و فرضیه‌ای که خطای مربعی را مینیمم می‌کند، بعضی محدودیت‌ها این شرایط مسئله را ذکر کنیم. بررسی بالا فقط خطا در تابع هدف نمونه‌های آموزشی در نظر گرفته شده است و از خطای خود ویژگی‌هایی که نمونه را توصیف می‌کنند صرف‌نظر شده بود. برای مثال، اگر مسئله یادگیری پیش‌بینی وزن افراد بر اساس سن و قدشان باشد، در شرایط ذکر شده فقط می‌توان خطا را برای وزن در نظر گرفت و مقادیر سن و قد دقیق فرض می‌شوند. بررسی زمانی پیچیده‌تر می‌شود که فرض‌های ساده کننده حذف شوند.

۶,۵ محتمل‌ترین فرضیه برای مسائل پیش‌بینی

در تعریف مسئله‌ی قسمت قبلی محتمل‌ترین فرضیه را فرضیه‌ای مشخص کردیم که مجموع خطای مربعی را بر روی نمونه‌های آموزشی مینیمم می‌کند. در این بخش معیاری مشابه برای تعریف مسئله‌ی دیگری که در شبکه‌های عصبی متداول است بیان می‌کنیم: یادگیری پیش‌بینی احتمالات.

حالتی را در نظر بگیرید که در آن می‌خواهیم تابعی غیرقطعی (احتمالی) $f: X \to \{0,1\}$ را یاد بگیریم که دو خروجی گسسته دارد. برای مثال، فضای نمونه‌ای X ممکن است توصیف بیماران با علائم بیماری‌شان باشد، و تابع هدف f(x) زمانی که بیمار زنده بماند ۱ و در غیر این صورت ۰ باشد. یا به طور مشابه X می‌تواند توصیف مراجعین دریافت وام با وضعیت حسابشان در گذشته باشد و f(x) زمانی که وام بعدی کامل پرداخت می‌شود ۱ و در غیر این صورت ۰ باشد. برای مثال، در مجموعه‌ای از بیماران که علائم مشترکی دارند ۹۲٪ درصد زنده می‌مانند و ۸٪ جان سالم به در نمی‌برند. این عدم قطعیت ممکن است ناشی از ناتوانی ما را در مشاهده‌ی تمامی علائم مهم بیمار باشد یا ممکن است ناشی از یک فرایند تصادفی در پیشرفت بیماری باشد. جدا از اینکه منشأ مشکل چیست، ما تابع هدف f(x) را داریم که به صورت احتمالی روی این ورودی عمل می‌کند.

با این تعریف مسئله ممکن است از یک شبکه‌ی عصبی (یا تخمین زننده‌ی توابع حقیقی مقدار دیگر) که خروجی‌اش احتمال f(x)=1 باشد استفاده کنیم. به عبارت دیگر، ما دنبال یادگیری تابع هدف $f': X \to [0,1]$ هستیم که در آن $f'(x)=P(f(x)=1)$. در مثال بالا، اگر آن علائم غیرقابل تمیز را داشته باشیم به احتمال ۹۲٪ بیمار زنده می‌ماند، پس $f'(x)=0.92$ که یعنی احتمال اینکه f(x) برابر با ۱ باشد ۹۲٪ است، و احتمال اینکه f(x) برابر با ۰ باشد، ۸٪ است.

چگونه می‌توان f' را با روشی مثل شبکه‌های عصبی یاد گرفت؟ یکی از راه‌های غیرهوشمندانه جمع کردن تعداد تکرار ۱ها و ۰های تابع برای هر نمونه‌ی ممکن x و آموزش شبکه‌ی عصبی با نسبت این تعداد تکرارهاست. همان طور که در ادامه نیز خواهیم دید، به جای این کار می‌توان از خود نمونه‌های آموزشی f برای آموزش شبکه‌ی عصبی استفاده کرد و محتمل‌ترین فرضیه برای f' را به دست آورد.

چه معیاری را بهینه می‌کنیم تا محتمل‌ترین فرضیه در این تعریف مسئله را بیابیم؟ برای جواب این سؤال ابتدا باید رابطه‌ای برای P(D|h) پیدا کنیم. بیایید فرض کنیم که نمونه‌های آموزش D به فرم $D = \{< x_1, d_1 > \cdots < x_m, d_m >\}$ هستند که در آن d_i ها مقدار مشاهده‌شده‌ی $f(x_i)$ است.

با توجه به آنچه درباره‌ی محتمل‌ترین فرضیه گفته شد، مینیمم خطای مربعی قسمت قبل، فرض کردیم که نمونه‌های $< x_1 \dots x_m >$ ثابت‌اند. تا بتوان داده‌ها را فقط با مقدار هدفشان، d_i بررسی کرد. با وجود اینکه می‌توانستیم فرض دیگری در این تعریف مسئله‌ی جدید داشته باشیم، بیایید با همین فرض قبلی ادامه دهیم تا نشان دهیم این چنین فرض‌هایی در نتیجه‌ی حاصل اثری ندارند. بنابراین فرض می‌کنیم که x_i و d_i متغیرهای تصادفی هستند و هر نمونه‌ی آموزشی مستقل ایجاد شده است پس می‌توانیم P(D|h) را به صورت زیر بنویسیم:

$$P(D|h) = \prod_{i=1}^{m} P(x_i, d_i|h) \tag{6.7}$$

باز هم فرض می‌کنیم که احتمال مواجهه با هر نمونه مثل x_i مستقل از h است. برای مثال، احتمال اینکه در مجموعه‌ی آموزشی بیمار x_i را داشته باشیم مستقل از فرضیه‌ی ما درباره‌ی احتمال زنده ماندن است (با این وجود البته احتمال زنده ماندن d_i خیلی به h مربوط نیست، ارتباط بین مجموعه‌ی آموزشی و فرضیه انکار ناشدنی است). زمانی که x از h مستقل باشد می‌توانیم رابطه‌ی بالا به رابطه‌ی زیر ساده کنیم، (با استفاده از قانون جدول ۶,۱)،

$$P(D|h) = \prod_{i=1}^{m} P(x_i, d_i|h) = \prod_{i=1}^{m} P(d_i|h, x_i)P(x_i) \tag{6.8}$$

حال احتمال $P(d_i|h, x_i)$ یا احتمال مشاهده‌ی $d_i = 1$ برای تک نمونه‌ی x_i با فرض اینکه فرضیه‌ی h درست است چیست؟ با توجه به اینکه h فرضیه‌ی ما از تابع هدفی است که احتمالات را محاسبه می‌کند، $P(d_i = 1|h, x_i) = h(x_i)$ و در کل،

$$P(d_i|h, x_i) = \begin{cases} h(x_i) & if\ d_i = 1 \\ (1 - h(x_i)) & if\ d_i = 0 \end{cases} \tag{6.9}$$

برای جایگزینی این رابطه در رابطه‌ی ۶,۸ برای P(D|h) بیایید ابتدا این رابطه را به فرم ریاضی‌وار تری بنویسیم،

$$P(d_i|h, x_i) = h(x_i)^{d_i}\big(1 - h(x_i)\big)^{1-d_i} \tag{6.10}$$

به سادگی می‌توان نشان داد که دو رابطه‌ی ۶,۹ و ۶,۱۰ هم‌ارزند. توجه داشته باشید که زمانی که $d_i = 1$ عبارت دوم رابطه‌ی ۶,۱۰، $\big(1 - h(x_i)\big)^{1-d_i}$ مساوی یک می‌شود بنابراین خواهیم داشت که $P(d_i = 1|h, x_i) = h(x_i)^{d_i}$ که هم‌ارز حالت اول رابطه‌ی ۶,۹ است، به طور مشابه می‌توان نشان داد که برای $d_i = 0$ نیز دو رابطه با هم، هم‌ارزند.

می‌توان از رابطه‌ی ۶,۱۰ برای جایگزینی $P(d_i|h, x_i)$ در رابطه‌ی ۶,۸ استفاده کرد،

$$P(D|h) = \prod_{i=1}^{m} h(x_i)^{d_i}\big(1 - h(x_i)\big)^{1-d_i} P(x_i) \tag{6.11}$$

حال می‌توانیم رابطه‌ی محتمل‌ترین فرضیه را بنویسیم،

$$h_{ML} = \arg\max_{h \in H} \prod_{i=1}^{m} h(x_i)^{d_i}\big(1-h(x_i)\big)^{1-d_i} P(x_i)$$

جمله‌ی آخری مستقل از h است و می‌توان آن را حذف کرد،

$$h_{ML} = \arg\max_{h \in H} \prod_{i=1}^{m} h(x_i)^{d_i}\big(1-h(x_i)\big)^{1-d_i} \qquad (6.12)$$

عبارت سمت راست رابطه‌ی ۶,۱۲ را می‌توان در تعمیم توزیع دوجمله‌ای در جدول ۵,۳ دید. عبارت رابطه‌ی 6.12 احتمال ظهور برآمد $< d_1 \dots d_m >$ را با فرض اینکه هر سکه‌ی x_i احتمال شیر آمدن $h(x_i)$ را داشته باشد را نشان می‌دهد. توجه داشته باشید که توزیع دوجمله‌ای که در جدول ۵,۳ آمد مشابه این رابطه است، اما فرض دیگری نیز دارد، احتمال شیر آمدن برای تمامی سکه‌ها را مساوی فرض می‌کند $\big(h(x_i) = h(x_j), \forall i, j\big)$. اما در هر دو حالت فرض می‌کنیم که برآمد پرتاب سکه‌ها ناسازگارند، فرضی که در تعریف مسئله فعلی ما نیز صدق می‌کند.

مشابه گذشته، کار با لگاریتم محتمل بودن راحت‌تر از خود محتمل بودن است پس داریم:

$$h_{ML} = \arg\max_{h \in H} \sum_{i=1}^{m} d_i lnh(x_i) + (1-d_i)\ln\big(1-h(x_i)\big) \qquad (6.13)$$

رابطه‌ی ۶,۱۳ کمیتی را نشان می‌دهد که برای پیدا کردن محتمل‌ترین فرضیه در تعریف مسئله‌ی فعلی ماکزیمم می‌کنیم. این نتیجه مشابه نتیجه‌ی قبلی ما در مینیمم کردن مجموع خطای مربعی محتمل‌ترین فرضیه در تعریف مسئله‌ی قبلی است. به شباهت بین رابطه‌ی 6.13 و فرم کلی تابع آنتروپی $-\sum_i p_i logp_i$ که در فصل ۳ آمد توجه کنید. به خاطر این شباهت، قرینه‌ی عبارت بالا گاهی آنتروپی دورگه[۹] نامیده می‌شود.

۶,۵,۱ شیب نزول برای پیدا کردن محتمل‌ترین فرضیه در یک شبکه‌ی عصبی

در بالا نشان دادیم که با ماکزیمم کردن کمیت رابطه‌ی ۶,۱۳ محتمل‌ترین فرضیه به دست خواهد آمد. بیایید این کمیت را با اختصار G(h,D) نشان دهیم. در این بخش قانونی برای آموزش وزن‌ها[۱۰] برای شبکه‌های عصبی به دست خواهیم آورد که G(h,D) را توسط روش شیب نزول ماکزیمم می‌کند.

همان طور که در فصل ۴ نیز بحث شد، گرادیان G(h,D) توسط بردار مشتق‌های جزئی G(h,D) نسبت به وزن‌های مختلف شبکه که فرضیه‌ی h را مشخص می‌کند ایجاد می‌شود (برای توضیح کامل درباره‌ی جزئیات جستجوی شیب نزول و واژگان بکار رفته به فصل ۴ مراجعه کنید). در این قسمت، مشتق جزئی G(h,D) نسبت به وزن w_{jk} که از واحد kام به واحد jام است به فرم زیر است:

---

[۹] cross entropy

[۱۰] weight training

$$\frac{\partial G(h,D)}{\partial \mathrm{w_{jk}}} = \sum_{i=1}^{m} \frac{\partial G(h,D)}{\partial h(x_i)} \frac{\partial h(x_i)}{\partial w_{jk}}$$

$$= \sum_{i=1}^{m} \frac{\partial (d_i lnh(x_i) + (1-d_i)\ln\big(1-h(x_i)\big))}{\partial h(x_i)} \frac{\partial h(x_i)}{\partial w_{jk}}$$

$$= \sum_{i=1}^{m} \frac{d_i - h(x_i)}{\partial h(x_i)(1-h(x_i))} \frac{\partial h(x_i)}{\partial w_{jk}} \quad (6.14)$$

برای ساده نگه داشتن محاسبات، فرض کنید که شبکه‌ی عصبی ما از یک لایه واحد سیگموید تشکیل شده و در این حالت داریم که

$$\frac{\partial h(x_i)}{\partial w_{jk}} = \sigma'(x')x_{ijk} = h(x_i)\big(1-h(x_i)\big)x_{ijk}$$

در این رابطه x_{ijk}، k امین ورودی به واحد j برای i امین نمونه‌ی آموزشی است، و σ’(x) مشتق تابع سیگموید است (به فصل ۴ رجوع کنید). بالاخره، این رابطه را در رابطه‌ی ۶,۱۴ جایگذاری می‌کنیم و رابطه‌ای برای مؤلفه‌های گرادیان به دست می‌آوریم،

$$\frac{\partial G(h,D)}{\partial \mathrm{w_{jk}}} = \sum_{\mathrm{i=1}}^{\mathrm{m}} \big(\mathrm{d_i} - \mathrm{h(x_i)}\big)\mathrm{x_{ijk}}$$

چون بیشتر به دنبال ماکزیمم P(D|h) هستیم تا مینیمم به جای شیب نزول از جستجوی شیب صعود[11] استفاده می‌کنیم. در هر حلقه جستجو بردار توسط قانون زیر به سمت گرادیان تصحیح می‌شود.

$$w_{jk} \leftarrow w_{jk} + \Delta w_{jk}$$

که داریم،

$$\Delta w_{jk} = \eta \sum_{i=1}^{m} \big(d_i - h(x_i)\big)x_{ijk} \quad (6.15)$$

و در این رابطه نیز η مقدار کوچک و مثبت است که اندازه‌ی قدم‌ها در جستجوی شیب صعود را مشخص می‌کند.

جالب است که این قانون تغییر وزن‌ها را با قانون تغییر وزن الگوریتم Backpropagation که مجموع خطای مربعی بین پیش‌بینی و مقدار اصلی را مینیمم می‌کرد مقایسه کنیم. قانون تغییر وزن برای واحدهای خروجی در Backpropagation با نشانه‌گذاری این فصل بـه شـکل زیر است،

$$w_{jk} \leftarrow w_{jk} + \Delta w_{jk}$$

---

[11] gradient ascent

که در آن

$$\Delta w_{jk} = \eta \sum_{i=1}^{m} h(x_i)(1-h(x_i))\big(d_i - h(x_i)\big)x_{ijk}$$

توجه دارید که این رابطه جز در جمله‌ی $h(x_i)(1-h(x_i))$ که از تابع سیگموید ناشی شده کاملاً شبیه رابطه‌ی ۶,۱۵ است.

خلاصه اینکه، این دو قانون تغییر وزن هر دو در تعریف مسئله‌ی خودشان به سمت محتمل‌ترین فرضیه همگرا می‌شوند. قانونی که مجموع خطاهای مربعی را مینیمم می‌کند با فرض اینکه خطاهای داده‌های آموزشی را می‌توان با توزیع نرمال مدل‌سازی کرد به دنبال محتمل‌ترین فرضیه می‌گردد. قانونی که آنتروپی دورگه را مینیمم می‌کند با فرض اینکه مقادیر منطقی مشاهده شده احتمالی (و نه قطعی) هستند به دنبال محتمل‌ترین فرضیه برای تابع پیش‌بینی احتمال بر حسب نمونه‌ها می‌گردد.

٦,٦ قانون کمترین طول توضیح[۱۲]

با توجه به آنچه در فصل ۳ درباره‌ی تیغ Ocam گفته شد، یک بایاس استقرایی متداول، به فرم "توضیحی که کوتاه‌تر است را در مورد داده‌های مشاهده شده قبول کن" است. در آن فصل درباره‌ی ضررهای توضیحات بلند با توجه به تیغ Ocam استدلال کردیم. در اینجا با دیدی بیزی به این موضوع می‌پردازیم و قانونی مشابه به نام قانون کمترین طول توضیح (MDL) را بررسی خواهیم کرد.

انگیزه‌ی ایجاد قانون کمترین طول توضیح تفسیر تعریف h_{MAP} با مفاهیم اولیه‌ی تئوری اطلاعات است. دوباره تعریف نه چندان ناآشنای h_{MAP} را به خاطر بیاورید.

$$h_{MAP} = \arg\max_{h \in H} P(D|h)P(h)$$

این رابطه را می‌توان به صورت معادل با log_2 آن نیز نشان داد،

$$h_{MAP} = \arg\min_{h \in H} -\log_2 P(D|h) - \log_2 P(h) \qquad (6.16)$$

جالب است که رابطه‌ی ۶,۱۶ را می‌توان طوری تفسیر کرد که فرضیه‌های کوتاه‌تر ارجح‌ترند، با فرض اینکه یک طرح نمایش خاص برای کد کردن فرضیه‌ها و داده‌ها استفاده کنیم. برای توضیح این، بیایید ابتدا یک نتیجه‌ی اساسی تئوری اطلاعات را معرفی کنیم: مسئله‌ی طراحی کدی برای ارسال پیام‌های تصادفی، را که در آن احتمال ارسال پیام i مقدار p_i است را در نظر بگیرید. در اینجا علاقه‌ی ما به فشرده‌ترین کد ممکن است؛ به عبارت دیگر علاقه‌ی ما به کدی است که امید تعداد بیت‌هایی که باید ارسال شوند تا یک پیام تصادفی فرستاده شود مینیمم کند. واضح است که برای مینیمم کردن امید طول کد ارسالی باید کدهای کوتاه‌تر را به پیام‌هایی اختصاص دهیم که احتمال بیشتری دارند. (Shannon and Weaver 1949) نشان دادند که کد بهینه (کدی که امید تعداد بیت‌های ارسالی را مینیمم می‌کند) به پیام i، $\log_2 p_i$ بیت برای کد کردن اختصاص می‌دهد. به این تعداد بیت که برای کد کردن پیام i توسط کد C لازم است طول توضیح پیام i بر اساس C نیز می‌گویند و با $L_C(i)$ آن را نشان می‌دهند.

---

[۱۲] Minimum description length

بیایید حالا رابطه‌ی ۶,۱۶ را با توجه به نتیجه‌ی بالا از تئوری کد سازی بررسی کنیم.

- $-\log_2 P(h)$ اندازه‌ی توضیح h بر اساس کد بهینه‌ی تمامی فضای فرضیه‌ای H است. به عبارت دیگر، این مقدار اندازه‌ی توضیحات فرضیه‌ی h با استفاده از نمایش بهینه است. در نمادگذاری فعلی $L_{C_H}(h) = -\log_2 P(h)$، که در آن C_H کد بهینه برای کد کردن فضای فرضیه‌ای H است.
- $-\log_2 P(D|h)$ اندازه‌ی توضیح داده‌های آموزشی D با معلوم بودن h توسط کد بهینه است. در نمادگذاری فعلی $L_{C_{D|h}}(D|h) = -\log_2 P(D|h)$، که در آن $C_{D|h}$ کد بهینه برای توضیح داده‌های D با فرض اینکه فرستنده و گیرنده هر دو مطلع از h هستند است.
- بنابراین می‌توانیم رابطه‌ی ۶,۱۶ را برای تعریف h_{MAP} بازنویسی کنیم و بگوییم h_{MAP} فرضیه‌ای مثل h است که مجموع طول توضیحات فرضیه‌ها به علاوه‌ی طول توضیحات داده‌ها با معلوم بودن فرضیه را مینیمم می‌کند.

$$h_{MAP} = \arg\min_{h} L_{C_H}(h) + L_{C_{D|h}}(D|h)$$

در این رابطه C_H و $C_{D|h}$ به ترتیب کدهای بهینه برای H و D با معلوم بودن h هستند.

قانون کمترین طول توضیح (MDL) توصیه می‌کند که فرضیه‌هایی را انتخاب کنیم که مجموع این دو طول توضیح را حداقل کنند. البته برای بکار بردن این قانون در عمل باید کد سازی یا نمایش خاصی را که با عمل یادگیری متناسب است انتخاب کنیم. با فرض اینکه ما از کدهای C_1 و C_2 برای نمایش فرضیه‌ها و داده‌ها با معلوم بودن فرضیه استفاده می‌کنیم، می‌توان MDL را به صورت زیر بیان کرد،

قانون کمترین طول توضیح: فرضیه‌ی h_{MDL} را انتخاب کن،

$$h_{MDL} = \arg\min_{h\in H} L_{C_1}(h) + L_{C_2}(D|h) \qquad (6.17)$$

بررسی بالا نشان می‌دهد که اگر ما C_1 را برای کد سازی بهینه‌ی فرضیه‌ها، C_H، و C_2 را برای کد سازی بهینه‌ی داده‌ها، $C_{D|h}$، انتخاب کنیم داریم $h_{MDL} = H_{MAP}$.

به صورت مفهومی، می‌توان به قانون MDL به فرم ترجیح متدهای کوتاه‌تر برای کد سازی دوباره‌ی داده‌های آموزشی نگاه کرد که در آن هر دو معیار اندازه‌ی فرضیه و هزینه‌ی اضافی کد سازی داده‌ها به شرط معلوم بودن فرضیه در نظر گرفته می‌شود.

بیایید مثالی را در نظر بگیریم. فرض کنید قصد داریم از قانون MDL برای مسئله‌ی یادگیری درخت‌های تصمیم از داده‌های آموزشی‌ای استفاده کنیم. برای نمایش فرضیه C_1 و داده‌های C_2 چه نمایشی را باید در نظر بگیریم؟ برای C_1 می‌توان به طور طبیعی یکی از کد سازی‌های واضح درخت تصمیم، که در آن طول توضیح با افزایش تعداد گره‌های درخت و تعداد یال‌ها افزایش می‌یابد را انتخاب کرد. اما چگونه باید با معلوم بودن یک درخت فرضیه‌ی خاص مجموعه‌ی داده‌های C_1 را کد کرد. برای ساده نگه داشتن موضوع، فرض کنید که سری نمونه‌های $< x_1 \dots x_n >$ برای فرستنده و گیرنده معلوم باشد، پس تنها چیز باقیمانده برای ارسال دسته‌بندی‌های $< f(x_1) \dots f(x_n) >$ است. (توجه دارید که هزینه‌ی ارسال خود نمونه‌ها از درستی فرضیه مستقل است، پس به هر حال تأثیری بر انتخاب h_{MDL} ندارد). حال اگر دسته‌بندی‌های $< f(x_1) \dots f(x_n) >$ همان پیش‌بینی‌های فرضیه باشد، دیگر نیازی به ارسال اطلاعات در مورد نمونه‌ها نیست (گیرنده می‌تواند این مقادیر را با فرضیه‌ای که دریافت کرده محاسبه کند). پس بنابراین طول توضیحات لازم با داشتن فرضیه در این حالت صفر است. در

چنین شرایطی اگر نمونه‌هایی توسط h اشتباه دسته‌بندی شده باشند، لازم است پیغامی مبنی بر دسته‌بندی اشتباه این نمونه‌ها (طول این پیغام حداکثر $\log_2 n$ بیت خواهد بود) را به همراه دسته‌بندی درست آن‌ها ارسال کنیم (این کار را می‌توان با پیغامی با حداکثر طول $\log_2 k$ انجام داد که در آن k تعداد دسته‌بندی‌های ممکن هر نمونه است). در چنین شرایطی فرضیه‌ی h_{MDL} تحت کد سازی C_1 و C_2 فرضیه‌ای است که کمترین مجموع طول توضیح را لازم داشته باشد.

بنابراین قانون MDL راهی برای ارزیابی پیچیدگی فرضیه‌ها با تعداد اشتباه‌های فرضیه ارائه می‌کند. ممکن است این معیار فرضیه‌ای کوتاه‌تر را که اشتباهات کمی دارد را نسبت به یک فرضیه بلندتر که اشتباهی ندارد ترجیح دهد. MDL از این نظر، متدی مناسب برای برخورد با مسئله‌ی overfit است.

(Quinlar and Rivest 1989) آزمایشاتی را با استفاده از قانون MDL برای تشخیص بهترین اندازه‌ی درخت تصمیم انجام داده‌اند. آن‌ها گزارش داده‌اند که متد مبتنی بر MDL درخت‌هایی را ایجاد می‌کند که دقتی قابل‌مقایسه با درخت‌های خروجی الگوریتم‌های فصل ۳ دارند. (Mehta et al. 1995) نیز روش دیگری مبتنی بر MDL برای هرس درخت تصمیم ارائه می‌کند و آزمایش‌هایی را تشریح کرده که در آن روش مبتنی بر MDL نتایج قابل‌مقایسه‌ای با روش‌های معمول را می‌دهد.

چه نتیجه‌گیری‌ای را باید از بررسی قانون کمترین طول توضیح بگیریم؟ آیا این اثباتی بر این که تمامی فرضیه‌های کوتاه‌تر ارجح‌اند است؟ خیر. بلکه ما اثبات کردیم که اگر نمایش فرضیه طوری انتخاب شود که کد سازی فرضیه‌ی h، $-\log_2 P(h)$ باشد و اگر کد سازی استثنا به گونه‌ای باشد که طول کد D با شرط معلوم بود h، $-\log_2 P(h|D)$، آنگاه قانون MDL فرضیه‌ای MAP خروجی خواهد داد. با این وجود، برای نشان دادن برقراری چنین شرطی باید تمامی احتمالات اولیه‌ی P(h) و P(D|h) را داشته باشیم. هیچ دلیلی برای این وجود ندارد که باور داشته باشیم که MDL برای هر کد سازی دلخواه C_1 و C_2 بر قرار است. ممکن گاهی برای طراح انسانی مشخص کردن نمایشی خاص برای دانش در مورد احتمالات نسبی فرضیه‌ها راحت‌تر از نمایش کامل احتمال دقیق هر یک از فرضیه‌ها باشد. توصیفات به کار رفته در ادبیات کاربرد MDL در مسائل یادگیری کاربردی گاهی شامل معیارهایی می‌شود که فرم خاصی از کد سازی C_1 و C_2 را توجیه می‌کند.

٦,٧ دسته‌بندی کننده‌ی بهینه‌ی بیز[۱۳]

تا اینجا به سؤال "محتمل‌ترین فرضیه با داشتن داده‌های آموزشی کدام است؟" پرداختیم، در واقع، این سؤال بیشتر شبیه این سؤال است که "محتمل‌ترین دسته‌بندی کننده نمونه‌های جدید با داشتن داده‌های آموزشی کدام است؟". با وجود اینکه ممکن است به نظر برسد که این سؤال دوم را می‌توان با اعمال فرضیه‌ی MAP به نمونه‌های جدید جواب داد، کاری بهتر ممکن است.

برای ایجاد شهود فضای فرضیه‌ای را در نظر بگیرید که سه فرضیه‌ی h_1، h_2 و h_3 را شامل می‌شود. فرض کنید که احتمال ثانویه‌ی این فرضیه با داده‌های آموزشی به ترتیب 4. و 3. و 3. است. بنابراین، h_1 فرضیه‌ی MAP است. حال فرض کنید که نمونه‌ی جدید x به ما داده می‌شود که توسط h_1 مثبت و توسط دو فرضیه‌ی h_2 و h_3 منفی دسته‌بندی می‌شود. با در نظر گرفتن تمامی فرضیه‌ها نمونه‌ی x به احتمال 4. مثبت است (احتمال مربوط به فرضیه‌ی h_1)، و به احتمال 6. منفی است. محتمل‌ترین دسته‌بندی (منفی) در این مثال با دسته‌بندی MAP متفاوت است.

---

۱۳ bayes optimal classifier

در کل محتمل‌ترین دسته‌بندی نمونه‌ی جدید از ترکیب پیش‌بینی‌های همه‌ی فرضیه‌ها به دست می‌آید، فقط هـر فرضـیه بـه انـدازه‌ی احتمـال ثانویه‌اش در این دسته‌بندی تأثیرگذار است. اگر دسته‌بندی ممکن نمونه‌ی جدید v_j عضـو مجموعـه‌ی V باشـد، $P(v_j|D)$، احتمـال اینکـه دسته‌بندی v_j برای نمونه‌ی جدید درست باشد به صورت زیر است،

$$P(v_j|D) = \sum_{h_j \in H} P(v_j|h_i) P(h_i|D)$$

دسته‌بندی ی بهینه‌ی نمونه‌ی جدید مقدار v_j است که با آن $P(v_j|D)$ ماکزیمم می شود،

دسته‌بندی ی بهینه‌ی بیز:

$$\arg\max_{v_j \in V} \sum_{h_i \in H} P(v_j|h_i) P(h_i|D) \qquad (6.18)$$

برای شهود در مثال بالا، مجموعه‌ی دسته‌بندی‌های نمونه‌ی جدید $V = \{\oplus, \ominus\}$ است و

$$P(h_1|D) = .4\,, P(\ominus\,|h_1) = 0, P(\oplus\,|h_1) = 1$$

$$P(h_2|D) = .3\,, P(\ominus\,|h_2) = 1, P(\oplus\,|h_2) = 0$$

$$P(h_3|D) = .3\,, P(\ominus\,|h_3) = 1, P(\oplus\,|h_3) = 0$$

بنابراین،

$$\sum_{h_i \in H} P(\oplus\,|h_i) P(h_i|D) = .4$$

$$\sum_{h_i \in H} P(\ominus\,|h_i) P(h_i|D) = .6$$

و

$$\arg\max_{v_j \in \{\oplus, \ominus\}} \sum_{h_i \in H} P(v_j|h_i) P(h_i|D) = \ominus$$

هر سیستمی که نمونه‌های جدید را با رابطه‌ی ۶,۱۸ دسته‌بندی کند دسته‌بندی کننده‌ی بهینه‌ی بیز[۱۴] یا یادگیر بهینه‌ی بیـز[۱۵] نامیـده مـی‌شـود. هیچ متد دسته‌بندی دیگری با همان فضای فرضیه‌ای و همان دانش اولیه نمی‌تواند به طور متوسط بازده بهتری داشته باشد. ایـن متـد احتمـال اینکه نمونه‌ی جدید درست دسته‌بندی شود را با معلوم بودن داده‌های موجود و فضای فرضیه‌ای احتمالات اولیه‌ی فرضیه‌ها حداکثر می‌کند.

---

[۱۴] Bayes optimal classifier

برای مثال در یادگیری مفاهیم حقیقی مقدار با استفاده از فضای ویژه، همان طور که در قسمت قبلی هم گفته شد، دسته‌بندی بهینه‌ی بیز نمونه‌های جدید با دادن وزن (احتمال ثانویه‌ی فرضیه) و رأی‌گیری بین اعضای فضای ویژه انجام می‌گرفت.

یکی از ویژگی‌های عجیب دسته‌بندی کننده‌ی بهینه‌ی بیز این است که پیش‌بینی‌هایی که انجام می‌دهد ممکن است فرضیه‌ای را تشکیل دهد که حتی در H موجود نیست. تصور کنید که از رابطه‌ی ۶،۱۸ برای دسته‌بندی تمامی نمونه‌های X استفاده کرده‌ایم. این دسته‌بندی نمونه‌ها که بدین صورت تعریف می‌شود الزاماً با فرضیه‌ای مثل h در H سازگار نیست. یکی از روش‌های نگاه به این وضعیت تصور دسته‌بندی کننده‌ی بهینه‌ی بیز به عنوان عاملی است که فضای فرضیه‌ای H′ را به طرز مؤثری، که با فضای فرضیه‌ای H (که قضیه بیز روی آن اعمال شده) فرق دارد، در نظر می‌گیرد. در کل، H′ به صورت مؤثر فرضیه‌هایی که مقایسه‌ای خطی بین ترکیبات پیش‌بینی‌های فرضیه‌های مختلف H می‌کنند را شامل می‌شود.

۶،۸ الگوریتم گیبس

با وجود اینکه دسته‌بندی کننده‌ی بهینه‌ی بیز بهترین عملکرد ممکن را با داشتن داده‌های آموزشی دارد، اما اعمال آن هزینه‌بر است. این هزینه در محاسبه‌ی احتمال ثانویه‌ی تمامی فرضیه‌های H و ترکیب پیش‌بینی‌هایشان برای هر نمونه‌ی جدید است.

یک روش جایگزین، ولی کمتر بهینه الگوریتم گیبس (رجوع کنید به Opper and Haussler 1991) است، که به صورت زیر تعریف می‌شود:

۱. فرضیه‌ای مثل h از H به طور تصادفی و با توزیع احتمالات ثانویه انتخاب کن.
۲. از h برای دسته‌بندی نمونه‌ی جدید بعدی استفاده کن.

زمانی که نمونه‌ی جدیدی برای دسته‌بندی ارائه می‌شود، الگوریتم گیبس به سادگی فرضیه‌ای به طور تصادفی و با توزیع احتمالات ثانویه انتخاب می‌کند و دسته‌بندی آن را به عنوان خروجی می‌دهد. جالب‌تر اینکه، می‌توان نشان داد که در شرایطی امید تعداد دسته‌بندی‌های غلط این الگوریتم حداکثر دو برابر امید خطای دسته‌بندی کننده‌ی بهینه‌ی بیز است (Haussler 1994). به عبارت دقیق‌تر، مقدار امید برای تمامی مفاهیم هدف تصادفی و توزیع احتمال اولیه‌ی یادگیر محاسبه شده. در چنین شرایطی، مقدار امید خطای الگوریتم گیبس دو برابر بدتر از مقدار امید خطای دسته‌بندی کننده‌ی بهینه‌ی بیز است.

این نتیجه معنای جالبی در مسائل یادگیری مفهوم که قبلاً در موردشان بحث کردیم دارد. در کل، این نتیجه نشان می‌دهد که اگر یادگیر احتمالات اولیه H را یکسان فرض کند، و مفاهیم هدف نیز در واقع با چنین احتمالی انتخاب شوند، آنگاه دسته‌بندی نمونه‌ی بعدی با فرضیه‌ای که به طور تصادفی از فضای ویژه انتخاب می‌شود (با توزیعی یکنواخت)، حداکثر دو برابر امید خطای دسته‌بندی کننده‌ی بهینه‌ی بیز، امید خطا خواهد داشت. دوباره، با نمونه‌ای از بررسی بیزی یک الگوریتم غیر بیزی طرف هستیم که این بررسی میزان کارایی آن الگوریتم را مشخص می‌کند.

---

[۱۵] Bayes optimal learner

۶٫۹ دسته‌بندی کننده‌ی ساده‌ی بیز

یکی از متدهای پرکاربرد یادگیری بیزی، یادگیر ساده‌ی بیز[۱۶] است که معمولاً دسته‌بندی کننده‌ی ساده‌ی بیز[۱۷] نیز نامیده می‌شود. در بعضی کاربردها کارایی این متد قابل‌مقایسه با شبکه‌های عصبی و یادگیری درختی است. در این بخش دسته‌بندی کننده‌ی ساده‌ی بیز را موردبحث و بررسی قرار می‌دهیم و در بخش بعدی آن را در مسئله یادگیری‌ای واقعی دسته‌بندی متون زبان‌های طبیعی به کار می‌بریم.

دسته‌بندی کننده‌ی ساده‌ی بیز در کارهای یادگیری‌ای به کار می‌رود که در آن x با عطفی از مقادیر ویژگی‌ها مشخص می‌شود و تابع هدف f(x) می‌تواند هر مقدار از مجموعه‌ی V باشد. مجموعه‌ای از نمونه‌های آموزشی تابع هدف و نمونه‌ای جدید که با ویژگی‌هایش توصیف شده به یادگیر داده می‌شود، $< a_1, a_2 \dots a_n >$ و از آن خواسته می‌شود که مقدار تابع هدف یا دسته‌بندی تابع هدف را برای این نمونه‌ی جدید پیش‌بینی کند.

روش بیزی برای دسته‌بندی نمونه‌ی جدید، دسته‌بندی آن بر اساس محتمل‌ترین مقدار تابع هدف، v_{MAP} است با داشتن نمونه‌های $< a_1, a_2 \dots a_n >$ است.

$$v_{MAP} = \arg\max_{v_j \in V} P(v_j | a_1, a_2 \dots a_n)$$

با استفاده از قضیه‌ی بیز این رابطه را بازنویسی می‌کنیم،

$$v_{MAP} = \arg\max_{v_j \in V} \frac{P(a_1, a_2 \dots a_n | v_j) P(v_j)}{P(a_1, a_2 \dots a_n)}$$

$$= \arg\max_{v_j \in V} P(a_1, a_2 \dots a_n | v_j)\, P(v_j) \qquad (6.19)$$

حال می‌توانیم دو عبارت رابطه‌ی ۶٫۱۹ را بر اساس داده‌های آموزشی تخمین بزنیم. تخمین مقادیر $P(v_j)$ با شمارش تعداد تکرار مقدارهای ویژگی هدف در بین داده‌های آموزشی بسیار ساده است. با این وجود، تخمین عبارتی با فرم $P(a_1, a_2 \dots a_n | v_j)$ بدین صورت ممکن نیست، مگر اینکه مجموعه‌ی داده‌های آموزشی‌مان بسیار بزرگ باشد. مشکل اینجاست که تعداد این چنین عبارت‌هایی مساوی تعداد نمونه‌های ممکن ضربدر تعداد مقادیر ممکن تابع هدف است. بنابراین لازم است که هر نمونه ممکن در فضای نمونه‌ای چندین بار مشاهده شود تا تخمین احتمال قابل‌اطمینان باشد.

دسته‌بندی کننده‌ی ساده‌ی بیز بر اساس یک فرض ساده‌سازی است، مقدار ویژگی‌ها با معلوم بودن مقدار هدف مستقل‌اند. به عبارت دیگر، فرض‌هایی که با داشتن مقدار هدف نمونه می‌توان زد، احتمال مشاهده‌ی عطف $a_1, a_2 \dots a_n$ فقط وابسته به احتمال تک‌تک این نمونه‌هاست: $P(a_1, a_2 \dots a_b | v_j) = \prod_i P(a_i | v_i)$. با جایگذاری این رابطه در رابطه‌ی ۶٫۱۹ به دسته‌بندی کننده‌ی ساده‌ی بیز می‌رسیم.

---

[۱۶] Naïve bayes learner

[۱۷] Naïve bayes classifier

دسته‌بندی کننده‌ی ساده بیز:

$$v_{NB} = \arg\max_{v_j \in V} P(v_j) \prod_i P(a_i|v_i) \qquad (6.20)$$

در این رابطه v_{NB} نماد مقدار هدفی خروجی دسته‌بندی کننده‌ی ساده‌ی بیز است. توجه دارید که در یک دسته‌بندی کننده‌ی ساده‌ی بیز تعـداد جملات متمایز $P(a_i|v_i)$ موجود، که باید بر اساس داده‌های آموزشی تخمین زده شود، ضرب تعداد مقادیر ویژگـی‌هـا و تعـداد مقـادیر هـدف است، این عدد در نگاه اول، نسبت به تعداد جملات ممکن $P(a_1, a_2 \ldots a_b|v_j)$ بسیار کوچکتر است.

به طور خلاصه، متد یادگیری ساده‌ی بیز مرحله‌ای دارد که در آن جملات مختلف $P(a_i|v_i)$ و $P(v_j)$ بر اساس تعـداد تکرارشـان در میـان نمونه‌های آموزشی تخمین زده می‌شوند. مجموعه‌ی این تخمین‌ها تعیین‌کننده‌ی فرضیه‌ی تخمینی خواهد بود. ایـن فرضـیه، بـرای دسـته‌بندی نمونه‌های جدید رابطه‌ی ۶,۲۰ را بکار خواهد بست. هرگاه که فرض استقلال شـرطی ارضـا مـی‌شـود، و دسـته‌بندی سـاده‌ی بیـز v_{NB} همـان دسته‌بندی MAP خواهد بود.

یکی از تفاوت‌های جالب دسته‌بندی کننده‌ی ساده‌ی بیز و دیگر متدهای یادگیری بحث شده، این است که این روش جستجویی صریح در میان فرضیه‌های ممکن انجام نمی‌دهد (در چنین شرایطی، فضای فرضیه‌ها همان فضای مقادیر ممکن قابل نسبت به متغیـر $P(v_j)$ و $P(a_i|v_j)$ است). در مقابل، فرضیه‌ها بدون جستجو و فقط با شمارش تعداد تکرار ترکیب‌های مختلف داده در میان نمونه‌های آموزشی ایجاد می‌شوند.

۶,۹,۱ مثالی توضیحی

بیایید دسته‌بندی کننده‌ی ساده‌ی بیز را به مسئله یادگیری مفهومی که در فصل یادگیری درختی مطرح شـد بکـار بـریم: دسـته‌بندی روزهـا بـر اساس اینکه کسی تنیس بازی خواهد کرد یا خیر. جدول ۳,۲ مجموعه‌ای از ۱۴ نمونه‌ی آموزشی را برای مفهوم PlayTennis نشان می‌دهد، در اینجا روزها با ویژگی‌های Outlook، Tempereture، Humidity، و Wind توصیف مـی‌شـوند. در اینجـا از دسـته‌بندی کننـده‌ی ساده‌ی بیز و داده‌های آموزشی این جدول برای دسته‌بندی نمونه‌ی جدید زیر استفاده می‌کنیم:

<Outlook=sunny,Temperature=cool,Humidity=high,Wind=strong>

هدف در اینجا پیش‌بینی مقدار هدف (Yes یا No) مفهوم هدف PlayTennis برای نمونه‌ی جدید است. با مقدار گذاری رابطه‌ی ۶,۲۰ برای این کار مقدار v_{NB} به صورت زیر محاسبه می شود.

$$v_{NB} = \arg\max_{v_j \in \{yes,no\}} P(v_j) \prod_i P(a_i|v_j)$$

$$= \arg\max_{v_j \in \{yes,no\}} \begin{matrix} P(v_j)\, P(\text{Outlook} = \text{sunny}|v_j) P(\text{Temperature} = \text{cool}|v_j) \\ P(\text{Humidity} = \text{High}|v_j) P(\text{Wind} = \text{strong}|v_j) \end{matrix} \qquad (6.21)$$

توجه دارید که در عبارت آخری a_i با استفاده از مقادیر ویژگی‌های نمونه‌ی جدید نوشته شده است. برای محاسبه‌ی v_{NB} بـه ۱۰ احتمـال نیـاز داریم که از روی داده‌های آموزشی تخمین زده می‌شوند. ابتدا، احتمال مقادیر مختلف هدف، که می‌توان آن را به سادگی با شمارش تکرار مقادیر از نمونه‌های آموزشی استخراج کرد.

$$P(PlayTennis = yes) = \frac{9}{14} = .64$$

$$P(PlayTennis = no) = \frac{5}{14} = .36$$

به طور مشابه می‌توان احتمالات شرطی را تخمین زد. برای مثال، برای Wind = strong داریم،

$$P(Wind = strong|PlayTennis = yes) = \frac{3}{9} = .33$$

$$P(Wind = strong|PlayTennis = no) = \frac{3}{5} = .60$$

با استفاده از تخمین‌های احتمالات مذکور و تخمین مشابه دیگر ویژگی‌ها، v_{NB} را بر اساس رابطه‌ی ۶,۲۱ به صورت زیر محاسبه می‌کنیم،

$$P(yes)\ P(sunny|yes)\ P(cool|yes)\ P(high|yes)\ P(strong|yes) = .0053$$

$$P(no)\ P(sunny|no)\ P(cool|no)\ P(high|no)\ P(strong|no) = .0053$$

دسته‌بندی کننده‌ی ساده‌ی بیز احتمالات تخمینی بر اساس داده‌های آموزشی موجود مقدار PlayTennis = no را به این نمونه‌ی جدید اختصاص می‌دهد. علاوه بر این، با نرمالیزه کردن کمیت‌های بالا (طوری که جمعشان یک شود) می‌توان احتمال شرطی اینکه مقدار تابع هدف no باشد را حساب کرد. برای مثال فعلی، این احتمال مقدار $.795 = \frac{0206}{.0206+.0053}$. است.

٦,٩,١,١ تخمین احتمالات

تا به حال، احتمالات را با نسبت تعداد مشاهده‌ی اتفاق به کل حالات را تخمین زدیم. برای مثال، در مثال بالا مقدار P(Wind=strong|PlayTennis=no) را با نسبت $\frac{n_c}{n}$ تخمین زدیم، در این نسبت n=5 تعداد نمونه‌های آموزشی PlayTennis=no و $n_c = 3$ تعداد نمونه‌هایی بود که در آن Wind=strong بود.

با وجود اینکه در بسیاری از موارد این نسبت تخمین خوبی از احتمال به ما می‌دهد، اما زمانی که n_c بسیار کوچک است تخمین ضعیف خواهد بود. برای درک این مشکل، فرض کنید که در حقیقت مقدار احتمال P(Wind=strong|PlayTennis=no) برابر با 08. باشد و در مجموعه‌ی نمونه‌های ما فقط ۵ نمونه مقدار PlayTennis=no را داشته باشند. با این فرض‌ها، n_c به احتمال زیادی صفر خواهد بود. این حقیقت دو مشکل ایجاد می‌کند. ابتدا اینکه $\frac{n_c}{n}$ تخمینی بایاس دار و دست کم گیرنده از مقدار احتمال خواهد بود. دوم اینکه زمانی که تخمین این احتمال صفر است باعث می‌شود که تمامی نمونه‌هایی که در آن‌ها Wind=strong است جزو دسته‌ی دیگر اطلاق شوند.

برای پرهیز از این مشکل می‌توان از روش بیزی برای تخمین احتمالات استفاده کرد، برای این کار تخمین m [۱۸] را به فرم زیر تعریف می‌کنیم.

تخمین m احتمالات:

[۱۸] m-estimate

$$\frac{n_c + mp}{n + m} \tag{6.22}$$

در این رابطه n_c و n همان مقادیر رابطه‌ی قبلی‌اند و p احتمال اولیه‌ی مقدار تخمینی است. m ثابتی که اندازه‌ی نمونه‌ی معادل[۱۹] نامیده می‌شود. این ثابت مشخص می‌کند که مقدار احتمال به چه میزان به نمونه‌های آموزشی وابسته باشد. یکی از روش‌های متداول انتخاب p، بدون داشتن هیچ اطلاعات قبلی‌ای، یکنواخت گرفتن تمامی احتمالات اولیه است؛ بدین معنا که اگر ویژگی‌ای k مقدار ممکن دارد خواهیم داشت که $p = \frac{1}{k}$. برای مثال، در تخمین P(Wind=strong|PlayTennis=no) می‌دانیم که ویژگی Wind دو مقدار ممکن دارد، پس با احتمال اولیه‌ی یکنواخت خواهیم داشت که p=.5. توجه دارید که اگر m را صفر انتخاب کنیم، تخمین m معادل همان کسر ساده‌ی $\frac{n_c}{n}$ می‌شود. اگر مقادیر m و n هر دو غیر صفر باشند، حاصل تخمین m میانگین دو مقدار با وزن m خواهد بود. m ثابت اندازه‌ی نمونه‌ی معادل نامیده می‌شود از این رو که رابطه‌ی ۶،۲۲ را می‌توان به صورت ترکیب n مشاهده‌ی واقعی و m مشاهده‌ی مجازی (با احتمال p) در نظر گرفت.

۶،۱۰ یک مثال: یادگیری دسته‌بندی متون

برای تصور اهمیت کاربردی متدهای یادگیری بیز، مسئله‌ی یادگیری‌ای را در نظر بگیرید که در آن نمونه‌ها متنند. برای مثال، شاید بخواهیم مفهوم هدف "مقالات خبری الکترونیکی جالب برای من" یا "صفحاتی از Web که یادگیری ماشین در آن‌ها بحث شده" را یاد بگیریم. در هر دو حالت، اگر یک کامپیوتر بتواند چنین کاری را انجام دهد می‌تواند به جای تعداد بسیاری زیادی از متون وب فقط مربوط‌ترین نتیجه جستجو روی وب را به کاربر ارائه کند.

در اینجا الگوریتمی کلی بر اساس دسته‌بندی کننده‌ی ساده‌ی بیز برای یادگیری دسته‌بندی متون ارائه می‌کنیم. جالب است که روش‌های احتمالی مثل آنچه پیش‌تر توضیح دادیم یکی از مؤثرترین الگوریتم‌های شناخته شده برای دسته‌بندی متون هستند. مثال‌هایی از چنین سیستم‌هایی در (Lewis 1991)، (Lang 1995) و (Joachims 1996) توصیف شده‌اند.

الگوریتم دسته‌بندی کننده‌ی ساده بیز که توضیح خواهیم داد با تعریف مسئله‌ای کلی تطابق دارد. فضای نمونه‌ای X را که شامل تمامی مستندات متنی (تمامی رشته کلمات و علامات با طول دلخواه) است در نظر بگیرید. به ما نمونه‌های آموزشی تابع هدف مجهول f(x) داده شده است، این تابع مجهول ممکن است هر یک از اعضای V باشد. هدف ما یادگیری از این نمونه‌های آموزشی برای پیش‌بینی مقدار هدف متنی جدید است. برای تصور، تابع هدف دسته‌بندی متون به دو دسته‌ی جذاب و غیر جذاب برای فرد بخصوص است، برای این تابع هدف مقادیر like (جذاب) و dislike (غیر جذاب) برای دسته‌بندی این دو مجموعه تعریف می‌شود.

دو مشکل اصلی برای کاربرد دسته‌بندی کننده‌ی ساده‌ی بیز در مسائل دسته‌بندی متن وجود دارد. اول اینکه با چه روشی یک متن دلخواه را با مقدار ویژگی‌هایی نمایش داد و دوم اینکه احتمالات لازم برای دسته‌بندی کننده‌ی ساده‌ی بیز را با چه روشی تخمین زد.

روش ما در نمایش متن دلخواه به طرز مشکل‌سازی ساده است: با داشتن یک متن، مثل همین پاراگراف، باید یک ویژگی برای هر مکان کلمه در متن تعریف کنیم و مقدار ویژگی‌ها را هم کلمات آن مکان‌ها در نظر بگیریم. بنابراین این پاراگراف ۹۷ مقدار ویژگی خواهد داشت که متناسب

---

[۱۹] equivalent sample size

با ۹۷ کلمه‌ی این پاراگراف است. مقدار ویژگی اول کلمه‌ی "روش" و مقدار ویژگی دوم کلمه‌ی "ما" خواهد بود و ... توجه دارید که با این روش متون بلند تعداد بیشتری ویژگی خواهند داشت. همان طور که بعداً نیز خواهیم دید، این تفاوت هیچ مشکلی ایجاد نمی‌کند.

با این نمایش برای متون، حال می‌توانیم دسته‌بندی کننده‌ی ساده‌ی بیز را به مسئله اعمال کنیم. بیایید به خاطر حفظ سادگی، فرض کنیم که ۷۰۰ متن آموزشی که فردی dislike دسته‌بندی کرده به همراه ۳۰۰ متن دیگر که like دسته‌بندی شده در اختیار است. حال متن جدیدی در اختیار یادگیر قرار گرفته و از وی دسته‌بندی این متن سؤال می‌شود. دوباره به خاطر سادگی، بیایید فرض کنیم که متن جدید پاراگراف قبلی باشد. در چنین شرایطی، اگر رابطه‌ی ۶,۲۰ را برای دسته‌بندی مقداردهی کنیم خواهیم داشت که،

$$v_{NB} = \arg\max_{v_j \in \{like, dislike\}} P(v_j) \prod_{i=1}^{97} P(a_i | v_j)$$

$$\arg\max_{v_j \in \{like, dislike\}} P(v_j) P\left(a_1 = \text{"روش"} \middle| v_j\right) P\left(a_2 = \text{"ما"} \middle| v_j\right)$$

$$\dots P\left(a_{97} = \text{"نمیکند"} \middle| v_j\right)$$

به طور خلاصه، دسته‌بندی ساده‌ی بیز v_{NB} دسته‌بندی‌ای است که احتمال مشاهده‌ی کلماتی را که واقعاً در متن بوده‌اند را با توجه به فرض مستقل بودن ساده‌ی بیز ماکزیمم می‌کند. فرض مستقل بودن $P(a_1, \dots a_{97} | v_j) = \prod_1^{97} P(a_i | v_j)$ در این تعریف مسئله فرض می‌کند که احتمال هر کلمه با داشتن دسته‌بندی متن v_j، برای هر مکان در متن مستقل از دیگر کلمات دیگر مکان‌هاست. توجه می‌کنید که این فرض به وضوح غلت است. برای مثال، در متون ممکن است احتمال آمدن کلمه‌ی "ماشین" بعد از کلمه‌ی "یادگیری" بسیار بیشتر از دیگر کلمات باشد. با وجود این نقص مشهود فرض مستقل بودن، انتخاب دیگری جز این نداریم، زیرا که بدون این شرط تعداد جملات احتمالی‌ای که باید محاسبه شوند به شدت زیاد می‌شوند. خوشبختانه در یادگیر ساده‌ی بیز در بسیاری از موارد در مسائل دسته‌بندی متون بر خلاف غلت بودن فرض استقلال نتایج خوبی به دست می‌آید. (Domingos and Pazzani 1996) بررسی جالبی از این پدیده‌ی تصادفی ارائه می‌کند.

برای محاسبه‌ی v_{NB} با استفاده از رابطه‌ی بالا، نیاز داریم که احتمال جمله‌های $P(v_j)$ و $P(a_i = w_k | v_j)$ را محاسبه کنیم (در این مسئله w_k، kامین واژه‌ی واژگان زبان فارسی است). احتمالات جمله‌ی اول را می‌توان به سادگی و با یک نسبت ساده از نمونه‌های آموزشی محاسبه کرد، (در مثال فعلی P(like)=.3 و P(dislike)=.7). مثل همیشه تخمین دسته‌بندی احتمالات شرطی (مثل $P(a_1 = \text{"روش"} | dislike)$) مشکل‌سازتر خواهد بود زیرا که باید چنین جمله‌ی احتمالی را برای هر ترکیب از متن ممکن کلمات فارسی و تابع هدف محاسبه کنیم. متأسفانه با 50,000 کلمه در واژگان زبان و ۲ حالت مقادیر هدف و ۹۷ کلمه‌ی نمونه‌ی فعلی به تقریباً محاسبه‌ی 2·97·50,000≈10,000,000 حالت نیاز خواهیم داشت.

خوشبختانه، می‌توانیم فرض استدلالی دیگری نیز که تعداد احتمالات را کم بکند به فرض‌ها پیشین اضافه کنیم. در کل، می‌توانیم احتمال برخورد با کلمه‌ی خاص w_k (مثل "شکلات") را مستقل از مکان حضورش (مثل a_{23} یا a_{95}) در نظر بگیریم. به عبارت رسمی‌تر، ویژگی‌ها از هم مستقل‌اند و توزیع یکسان نیز دارند، با معلوم بودن دسته‌بندی هدف؛ برای تمامی m،k،j،i داریم $P(a_i = w_k | v_j) = P(a_m = w_k | v_j)$. بنابراین تخمین می‌زنیم که کل مجموعه احتمالات $P(a_1 = w_k | v_j)$ ، $P(a_2 = w_k | v_j)$ ... برابر با یک مقدار مستقل مثبت $P(w_k | v_j)$ باشد، یعنی این مقدار احتمال به مکان کلمه بستگی ندارد. تأثیر این فرض این است که حال فقط نیاز به محاسبه‌ی

$2 \cdot 50,000$ جمله‌ی مستقل به فرم $P(w_k|v_j)$ داریم. این مقدار هنوز زیاد است اما دیگر در حد کنترل است. توجه دارید که در شرایطی که داده‌های آموزشی محدود باشند، مزیت اولیه‌ی این فرض افزایش تعداد نمونه‌های موجود بـرای تخمـین هـر یـک از احتمالـات و متعاقبـاً دقـت دسته‌بندی است.

برای کامل کردن طراحی الگوریتم یادگیری‌مان، هنوز باید متدی برای تخمین جملات احتمالات پیدا کنیم. از تخمین m، کـه در رابطـه‌ی ۶,۲۲ آمد، و احتمالات اولیه‌ی یکنواخت و اندازه‌ی واژگان موجود برای $P(w_k, v_j)$ داریم،

$$\frac{n_k + 1}{n + |Vocabulary|}$$

در این رابطه n کل تعداد کلمات ممکن در نمونه‌های آموزشی است با مقدار تابع هدف v_j اسـت، n_k تعـداد تکـرار کلمـه‌ی w_k در میـان n کلمه‌ی ممکن است و $|Vocabulary|$ نیز تعداد خالص کل کلمات (و دیگر نشانه‌های) موجود در نمونه‌های آموزشی است.

به طور خلاصه اینکه الگوریتم نهایی از دسته‌بندی کننده‌ی ساده‌ی بیز به همراه فرض استقلال کلمات از مکانشـان اسـتفاده مـی‌کنـد. الگـوریتم نهایی در جدول ۶,۲ آورده شده است. توجه می‌کنید که این الگوریتم به نسـبت سـاده اسـت. در طـول یـادگیری، زیـر روال Learn-Naive-bayes-text تمامی متون آموزشی را برای استخراج تمام کلمات و نشانه‌های موجـود در متـون بررسـی مـی‌کنـد و تعـداد تکرارشـان را در دسته‌بندی‌های مختلف تابع می‌شمرد تا تخمین‌های لازم را به دست آورد. سپس، برای یک متن جدید (که لازم اسـت دسـته‌بندی شـود) فرآینـد Classify-naive-bayes-text با توجه به رابطه‌ی ۶,۲۰ از این تخمین احتمالات برای محاسبه‌ی v_{NB} استفاده می‌کند. توجه داریـد کـه کلمه که در متن جدید ظاهر شده که در متون قبلی نبوده‌اند توسط Classify-naive-bayes-text نادیده گرفته می‌شود. کد و مجموعه‌ی داده‌های آموزشی در آدرس http://www.cs.cmu.edu/~tom/book.html موجود است.

٦,١٠,١ نتیجه‌های تجربی

الگوریتم جدول ۶,۲ به چه میزان کارایی دارد؟ در یک آزمایش (Joachims 1996)، الگوریتم بسیار مشابهی برای دسته‌بندی مقالات خبـری یوزنت[۲۰] بکار رفت. دسته‌بندی مقاله در این مثال اسم گروه خبری مقاله در یوزنت بود. الگوریتمی که هر مقالـه را پـس از دسـته‌بندی در جـای اصلی خود قرار می‌دهد. در این آزمایش ۲۰ گروه خبری الکترونیکی در نظر گرفته شد (که در جدول ۶,۳ نیز آمده‌اند)، سـپس 1,000 مقالـه از هر گروه خبری جمع شد تا تعداد نمونه‌ها به 20,000 برسد. الگوریتم ساده‌ی بیز دوسوم از این 20,000 متن به عنوان نمونـه‌های آموزشـی آموزش داده شد و سپس کارایی الگوریتم برای یک‌سوم باقیمانده ارزیابی شد. از ۲۰ گروه خبری ممکن، حداکثر مقدار دسته‌بندی درست اتفـاقی ۵٪ خواهد بود، اما دقت دسته‌بندی الگوریتم ۸۹٪ اندازه‌گیری شد. الگوریتم به کار رفته در این آزمایش فقط یـک تفـاوت کوچـک بـا الگـوریتم جدول ۶,۲ داشت، یک زیرمجموعه از کلمات متون به عنوان واژگان[۲۱] در نظر گرفته شده بود. به عبارت دقیق‌تر، ۱۰۰ کلمه‌ی پرکاربردتر واژگان در آن در نظر گرفته نشده بود (کلماتی مثل "این")، و همچنین تمامی کلماتی که کمتر از ۳ بار ظاهر شده بودند نادیده گرفته شدند. واژگان بـه دست آمده بدین ترتیب تقریباً 38,500 کلمه داشت.

[۲۰] use net
[۲۱] Vocabulary

نتایج چشم‌گیر دیگری نیز توسط دیگر روش‌های یادگیری آماری متون به دست آمده است. برای مثال، (Lang 1995) نسخه‌ای دیگر از الگوریتم ساده‌ی بیز را توصیف کرده و آن را در یادگیری مفهوم هدف "مقالات یوزنتی که من به آن‌ها علاقه دارم" به کار می‌برد. وی سیستم NewsWeeder را معرفی می‌کند، برنامه‌ای که به کاربران اجازه می‌دهد تا متون را بعد از خواند ارزیابی[۲۲] کنند. سیستم NewsWeeder از این ارزیابی‌ها به عنوان نمونه‌های آموزشی برای پیش‌بینی اینکه مقاله‌ای برای کاربر جالب است یا خیر استفاده می‌کند، پس برنامه می‌تواند مقالاتی که پیش‌بینی می‌کند کاربر به خواندن آن‌ها علاقه دارد را به وی پیشنهاد کند. (Lang 1995) آزمایشی را گزارش می‌کند که در آن NewsWeeder از اطلاعات یاد گرفته خود بر اساس علاقه‌ی کاربر، مقاله‌ای که بالاترین مقدار پیش‌بینی ارزیابی را دارد به کاربر ارائه می‌دهد. با ذخیره‌ی ۱۰٪ اول این مقالات اتوماتیک ارزیابی شده، برنامه مجموعه‌ای از مقالات خواهد داشت که نسبت به مجموعه‌ی کل مقالات سه تا چهار برابر برای کاربر جالب‌ترند. برای مثال، برای یک کاربر نسبت مقالاتی که "جالب[۲۳]" دسته‌بندی می‌کند در کل ۱۶٪ است اما در میان این مقالات ۵۹٪ توصیه‌ی NewsWeeder بوده.

تعداد زیاد روش‌های غیر بیزی آماری برای یادگیری متون متداول‌اند، بسیاری از این روش‌ها بر اساس معیارهای مشابه استخراج اطلاعات هستند. (Rocchio 1971; Salton). الگوریتم‌های یادگیری متون دیگر در (Hearst and Hirsh 1996) آورده شده است.

۶٫۱۱ شبکه‌های باور بیزی

همان طور که در دو قسمت قبلی نیز گفته شد، دسته‌بندی کننده‌ی ساده‌ی بیز از فرض اینکه احتمالات شرطی $a_1 \dots a_2$ با داشتن مقدار تابع هدف v مستقل‌اند استفاده‌ی شدیدی می‌کند. این فرض به طور قابل‌توجهی میزان پیچیدگی یادگیری تابع هدف را کاهش می‌دهد. با این فرض، دسته‌بندی کننده‌ی ساده‌ی بیز دسته‌بندی بهینه‌ی بیز را خروجی می‌دهد. با این وجود، در بسیاری از موارد این شرط مستقل بودن به شدت محدود کننده است.

شبکه‌های باور بیزی[۲۴] توزیع احتمالات حاکم بر مجموعه‌ی متغیرهایی که با دسته‌ای از فرض استقلال احتمالات شرطی مشخص می‌شوند را توصیف می‌کنند. بر خلاف دسته‌بندی کننده‌ی ساده‌ی بیز که فرض می‌کرد تمامی متغیرهای به طور شرطی با معلوم بودن فرضیه‌ی h مستقل‌اند، شبکه‌های باور بیزی فرض‌های استقلال احتمالات را در زیرمجموعه‌های متغیرها درست می‌دانند. بنابراین، شبکه‌های باور بیزی، روشی میانی که شرطی آزادتر از فرض مستقل بودن تمامی متغیرهای دسته‌بندی کننده‌ی ساده‌ی بیز و محدود کننده تر از پرهیز از هرگونه شرط استقلال است، ارائه می‌کنند. شبکه‌های باور بیز یکی از موضوعات مورد توجه تحقیقات فعلی هستند، و دامنه‌ی وسیعی از الگوریتم‌ها برای یادگیری و استنتاج از آن‌ها ارائه شده است. در این بخش مفاهیم کلیدی و نحوه‌ی نمایش شبکه‌های باور بیزی را معرفی خواهیم کرد. اطلاعات دقیق‌تر در این زمینه (Pearl 1988) و (Russell and Norving 1995) و (Heckerman 1995) و (Jensen 1996) آمده است.

---

۲۲ rate
۲۳ interesting
۲۴ bayesian belief networks

در کل، یک شبکه‌ی باور بیز توزیع‌های احتمال دسته‌ای از متغیرها را توصیف می‌کند. مجموعه‌ی دلخواهی از متغیرهای تصادفی $Y_1 \ldots Y_n$ را در نظر بگیرید که هر Y_i می‌تواند هر یک از مقادیر مجموعه‌ی $V(Y_i)$ را داشته باشد. فضای توأم[۲۵] را مجموعه‌ی متغیرهای Y که از ضرب خارجی $V(Y_1) \times V(Y_2) \ldots V(Y_n)$ به دست می‌آید تعریف می‌کنیم. به عبارت دیگر، هر عضو فضای توأم متناسب با یکی از مقادیر ممکن متغیرهای $< Y_1 \ldots Y_n >$ است. توزیع احتمال این فضای توأم، توزیع احتمال توأم[۲۶] نامیده می‌شود. توزیع احتمال توأم، احتمال مشاهده‌ی هر یک از نمونه‌های $< Y_1 \ldots Y_n >$ را مشخص می‌کند. یک شبکه‌ی باور بیز توزیع احتمال توأم یک مجموعه از متغیرها را توصیف می‌کند.

۶,۱۱,۱ شرط استقلال

بیایید بحثمان درباره‌ی شبکه‌های باور بیزی را با تعریف دقیق مفهوم استقلال آغاز کنیم. فرض کنید X، Y و Z سه متغیر تصادفی گسسته مقدار باشند، زمانی می‌گوییم که X از Y به شرط Z مستقل است که توزیع احتمال حاکم بر X با فرض داشتن مقدار Z مستقل از مقدار Y باشد؛ به عبارت دیگر،

$$(\forall x_i, y_j, z_k)\ P\left(X = x_i \middle| Y = y_j, Z = z_k\right) = P(X = x_i | Z = z_k)$$

در این رابطه $x_i \in V(X)$، $y_i \in V(Y)$ و $z_i \in V(Z)$ است. معمولاً عبارت بالا را به طور خلاصه به فرم P(X|Y,Z)=P(X|Z) می‌نویسیم. تعریف استقلال شرطی را می‌توان برای مجموعه‌ای از متغیرها تعمیم داد. می‌گوییم که مجموعه متغیرهای $X_1 \ldots X_l$ مستقل از مجموعه متغیرهای $Y_1 \ldots Y_m$ به شرط متغیرهای $Z_1 \ldots Z_l$ هستند اگر

$$P(X_1 \ldots X_l | Y_1 \ldots Y_m, Z_1 \ldots Z_l) = P(X_1 \ldots X_l | Z_1 \ldots Z_l)$$

به رابطه‌ی این تعریف و تعریفمان از استقلال شرطی در دسته‌بندی کننده‌ی ساده‌ی بیز توجه کنید. دسته‌بندی کننده‌ی ساده بیز به طور شرطی مستقل بودن ویژگی A_1 از ویژگی A_2 را تعریف می‌کند. این تعریف به دسته‌بندی کننده‌ی ساده‌ی بیز اجازه می‌دهد که مقدار $P(A_1, A_2 | V)$ را که در رابطه‌ی ۶,۲۰ آمده با استفاده از رابطه‌ی زیر محاسبه کند،

$$P(A_1, A_2 | V) = P(A_1 | A_2, V) P(A_2 | V) \qquad (6.23)$$

$$= P(A_1 | V) P(A_2 | V) \qquad (6.24)$$

رابطه‌ی ۶,۲۳ فقط فرم کلی حاصل از قانون احتمال جدول ۶,۱ است. رابطه‌ی ۲,۲۴ نیز از آن نتیجه شده است، زیرا که اگر A_1 با معلوم بودن V از A_2 مستقل باشد، پس طبق تعریف مستقل شرطی خواهیم داشت که، $P(A_1 | A_2, V) = P(A_1 | V)$.

۶,۱۱,۲ نمایش

یک شبکه‌ی باور بیزی (که معمولاً به اختصار شبکه بیزی نامیده می‌شود) با توزیع احتمالات توأم مجموعه‌ای از متغیرها نمایش داده می‌شود. برای مثال، شبکه بیزی شکل ۶,۳ توزیع احتمال توأم متغیرهای منطقی Storm، Lightning، Thunder، ForestFire، Campfire

---

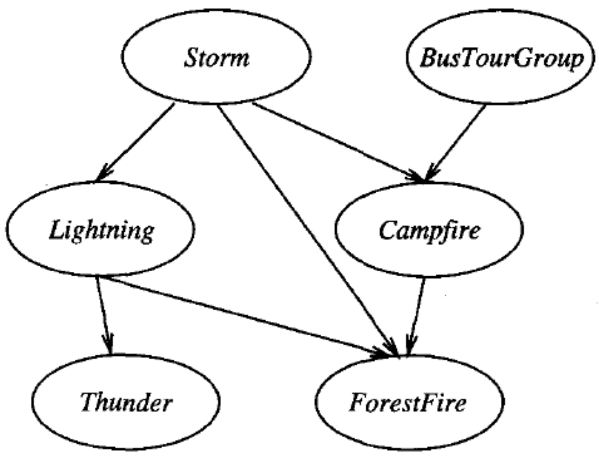
[۲۵] joint space

[۲۶] Joint probability distribution

و BusTourGroup را نشان می‌دهد. در کل، یک شبکه‌ی بیزی توزیع احتمال توأم را با استفاده از مشخص کردن مجموعه‌ای از فرض‌های استقلال شرطی (که با یک گراف بدون دور نمایش داده می‌شود) و مجموعه‌های از احتمالات شرطی هر کدام مشخص می‌کند. هر متغیر فضـای توأم با یک گره در شبکه بیزی نشان داده می‌شود. برای هر متغیر دو نوع اطلاعات ذکر می‌شود، اول اینکه با فـرض داشـتن والـدین (در گـراف) متغیر از متغیرهای غیر زیرینش مستقل شرطی است. زمانی می‌گوییم که X زیرینی[۲۷] برای Y است که مسیری مسـتقیم از Y بـه X باشـد. دوم اینکه جدولی از احتمالات شرطی برای هر متغیر داده می‌شود که توزیع احتمال را برای مقدار متغیرهای بالایی[۲۸] مشخص می‌کنـد. احتمـال تـوأم هر یک از مقادیر $< y_1, \dots , y_n >$ که مقداری از $< Y_1 \dots Y_n >$ است را می‌توان با استفاده از رابطه‌ی زیر محاسبه کرد،

$$P(y_1, \dots , y_n) = \prod_{i=1}^{n} P(y_i|Parnets(Y_i))$$

در این رابطه $Parents(Y_i)$ نماد مجموعه‌ای از بالایی‌های مستقیم Y_i در شبکه است. توجه داشته باشید کـه $P(y_i|Parents(Y_i))$ دقیقاً مقدارهای ذخیره شده در جدول احتمالات شرطی مربوط به گره Y_i است.



	S,B	S,¬B	¬S,B	¬S,¬B
C	0.4	0.1	0.8	0.2
¬C	0.6	0.9	0.2	0.8

شکل ۶,۳ یک شبکه‌ی باور بیزی.
شبکه‌ی سمت چپ مجموعه‌ای از فرض‌های استقلال شروط را نشان می‌دهد. در کل، هر گره مستقل شرطی است از شروط غیر زیرینش[۲۹] بـا معلـوم بـودن شروط والدش مستقل است. برای هر گره جدول مقادیر شرطی‌ای وجود دارد که توزیع احتمال شرطی متغیرها را با معلوم بـودن شـروط والـدینش در گـراف مشخص می‌کند. جدول احتمال شرطی مربوط به گره Campfire که به طور خلاصه با C نمایش داده شده در سمت راست شکل آورده شده است، گره‌های Storm و BusTourGroup نیز به ترتیب به طور خلاصه با S و B نمایش داده شده‌اند.

برای تصور، شبکه‌ی بیزی شکل ۶,۳ توزیع احتمال تـوأم را بـرای متغیرهـای منطقـی Storm، Lightning، Thunder، ForestFire، Campfire و BusTourGroup نشان می‌دهد. گره‌ها و یال‌های[۳۰] شبکه نشان مـی‌دهـد کـه Campfire بـا معلـوم بـودن والـدینش، Storm و BusTourGroup، از Lightning و Thunder مستقل شرطی است. این بدین معناست کـه زمـانی کـه مقـدار Storm و

[۲۷] descendant
[۲۸] predecessors
[۲۹] nondescendants
[۳۰] arc

BusTourGroup مشخص است متغیرهای Lightning و Thunder هیچ اطلاعات اضافه‌ای در مورد متغیر Campfire به ما نخواهند داد. برای مثال، سه داده‌ی اول سمت چپ جدول نشان می‌دهند که،

$$P(Campfire = True|Storm = True, BusTourGroup = True) = 0.4$$

توجه دارید که این جدول فقط مقادیر احتمال شرطی Campfire را با معلوم بودن مقادیر متغیرهای Storm و BusTourGroup می‌دهد. مجموعه‌ی موضعی جدول احتمالات شرطی برای تمامی متغیرها و مجموعه‌ای از فرض‌های استقلال شرطی که شبکه می‌گذارد، با هم توزیع احتمال شبکه روی کل فضای توأم را مشخص می‌کنند.

یکی از ویژگی‌های جذاب شبکه‌های باور بیزی این است که اجازه‌ی نمایش ساده‌ی اطلاعات علی[۳۱]، مثل این حقیقت که رعدوبرق (lightning) باعث طوفان (Thunder) می‌شود، را به ما می‌دهد. در واژگان استقلال شرطی، این حقیقت را در شبکه با اینکه احتمال Thunder با معلوم بودن مقدار Lightning از بقیه‌ی متغیرها مستقل است نشان می‌دهیم. توجه داشته باشید که این فرض استقلال شرطی با یال‌های شبکه‌ی بیزی شکل ۶٫۳ نشان داده شده است.

۶٫۱۱٫۳ استنتاج[۳۲]

ممکن است بخواهیم از شبکه‌های بیزی برای استنتاج مقدار چند متغیر (مثل ForestFire) با داشتن چند متغیر دیگر استفاده کنیم. البته، با دانستن اینکه کار ما با متغیرهای تصادفی است، در کل نیز نسبت دادن یک مقدار به متغیر هدف صحیح نخواهد بود. در اینجا ما بیشتر به استنتاج توزیع احتمال متغیر هدف علاقه داریم، توزیع احتمالی که مشخص می‌کند مقدار هدف با معلوم بودن مقادیر مفروض با چه احتمالی کدام مقدارش را می‌تواند داشته باشد. اگر مقادیر تمامی متغیرهای دیگر شبکه معلوم باشند مرحله‌ی استنتاج خیلی ساده خواهد شد. در حالت کلی‌تر ممکن است بخواهیم توزیع احتمال یک متغیر را با داشتن فقط زیرمجموعه‌ای از تمامی متغیرها (مثل ForestFire) استنتاج کنیم (ممکن است دو مقدار Thunder و BusTourGroup تنها مقادیر مشاهده شده‌ی ما باشند). در کل، یک شبکه‌ی بیزی را می‌توان برای محاسبه‌ی توزیع احتمال هر زیرمجموعه‌ای از متغیرهای شبکه با استفاده از معلوم بودن مقادیر هر زیرمجموعه‌ی دیگری از متغیرهای شبکه استفاده کرد.

استنتاج دقیق احتمالات در حالت کلی برای هر شبکه‌ی بیز دلخواه NP-hard است (Cooper 1990). متدهای عددی‌ای نیز برای استنتاج احتمالات در شبکه‌های بیزی، شامل متدهای استنتاج دقیق[۳۳] و متدهای تخمین استنتاج که دقت را فدای بازده می‌کنند ارائه شده‌اند. برای مثال، متدهای Monte Carlo راه‌حل‌های تخمینی را با استفاده از نمونه‌برداری تصادفی از توزیع احتمال متغیرهای مورد نظر را پیشنهاد می‌کنند (Pradham and Dagum 1996). در تئوری، حتی تخمین استنتاجی احتمالات شبکه‌ی بیزی را می‌توان NP-hard دانست (Dagum and Luby 1993). خوشبختانه در عمل، متدهای تخمینی در بسیاری از موارد مفید از آب در آمده‌اند. بحث متدهای استنتاج شبکه‌های بیزی در (Russell and Norvig 1995) و (Jensen 1996) آمده است.

---

[۳۱] causal knowledge
[۳۲] inference
[۳۳] exact inference

۶,۱۱,۴ یادگیری شبکه‌های باور بیزی

آیا می‌توانیم الگوریتمی مؤثر برای یادگیری شبکه‌های باور بیزی از داده‌های آموزشی پیدا کنیم؟ این سؤال، زمینه‌ی مورد توجه اکثر تحقیقات فعلی است. تعریف مسئله‌های مختلفی را می‌توان برای این سؤال در نظر گرفت. ابتدا اینکه ساختار شبکه ممکن است دقیق مشخص باشد، یا ممکن است ساختار شبکه با توجه به داده‌های آموزشی انتخاب شود. دوم اینکه تمامی متغیرهای شبکه ممکن است در هر نمونه‌ی آموزشی مشهود و معلوم باشد یا بلعکس بعضی ممکن است بعضی متغیرها غیرقابل‌مشاهده باشند.

در حالتی که ساختار شبکه دقیق مشخص است، و فقط بعضی از مقادیر متغیرهای آن قابل‌مشاهده‌اند، مسئله‌ی یادگیری بسیار سخت‌تر خواهد بود. این مسئله به نحوی مشابه یادگیری وزن‌های واحدهای پنهان شبکه‌های عصبی است، جایی که ورودی و خروجی شبکه مشخص است اما اطلاعاتی در مورد لایه‌ی پنهان شبکه در نمونه‌های آموزشی نیست. در واقع، (Russell 1995) فرایندی مشابه با شیب صعود ارائه داده است که احتمالات شرطی جدول را یاد می‌گیرد. این فرایند شیب صعود در فضایی از فرضیه‌ها که متناسب با مجموعه‌ای از تمامی حالت‌های ممکن احتمالات شرطی است برای یافتن مقادیر جدول احتمالات شرطی جستجو می‌کند. تابع ای که در طول شیب صعود ماکزیمم می‌شود احتمال P(D|h) داده‌های آموزشی D است با معلوم بودن فرضیه‌ی h است. طبق تعریف، این جستجو معادل جستجو برای محتمل‌ترین فرضیه برای مقادیر جدول است.

۶,۱۱,۵ آموزش شیب صعود برای شبکه‌های بیزی

قانون شیب صعودی که (Russell 1995) معرفی کرد، با حرکت به سمت افزایش ln P(D|h) مقدار P(D|h) را با توجه به مقادیر جدول احتمالات شرطی شبکه‌ی بیز ماکزیمم می‌کند. فرض کنید w_{ijk} نماد تک داده‌ی ای از جدول‌های احتمالات شرطی شبکه باشد. در کل فرض کنید که w_{ijk} نماد احتمال شرطی اینکه متغیر شبکه‌ی Y_i مقدار y_{ij} را داشته باشد با معلوم بودن اینکه متغیر U_i مقدار u_{ij} را داشته باشد. برای مثال، اگر w_{ijk} داده‌ی گوشه‌ی بالا و سمت راست جدول احتمالات شرطی ۶,۳ باشد و Y_i نیز متغیر Campfire باشد و U_i نیز والدین آن یعنی <Storm,BusTourGroup> و $y_{ij} = True$ و $u_{ik} =< False, False >$ است. گرادیان ln P(D|h) که با $\frac{\partial \ln P(D|h)}{\partial w_{ijk}}$ برای هر w_{ijk} نشان داده می‌شود را همان طور که بعداً نیز نشان خواهیم داد می‌توان به صورت زیر محاسبه کرد،

$$\frac{\partial lnP(D|h)}{\partial \mathrm{w_{ij}}} = \sum_{d \in D} \frac{\mathrm{P}\big(\mathrm{Y_i = y_{ij}, U_i = u_{ik}} \big| \mathrm{d}\big)}{\mathrm{w_{ijk}}} \qquad (6.25)$$

برای مثال، برای محاسبه‌ی هر یک از مقادیر مشتق ln P(D|h) نسبت به داده‌ی گوشه‌ی بالا و راست جدول ۶,۳ باید مقدار P(Campfire=True,Storm=False,BusTourGroup=False|d) را برای هر یک از نمونه‌های آموزشی d در D محاسبه کنیم. زمانی که این متغیرهای برای نمونه‌ای مثل d مجهول است، لازم است که این احتمال را با استنتاج از متغیرهای دیگر آموزشی موجود d محاسبه کنیم. در واقع، این کمیت‌ها به راحتی از محاسبات استنتاجی انجام شده در اکثر شبکه‌های بیزی استخراجی می‌شود، بنابراین یادگیری را می‌توان با هزینه‌ای کمی بیشتر، که از شبکه‌ی بیزی برای استنتاج و مدارک جدید متعاقباً به دست می‌آید.

در زیر از رابطه‌ی ۶,۲۵ که (Russell 1995) معرفی کرده به دست می‌آوریم. ادامه‌ی این قسمت را می‌توانید بدون از دست دادن پیوستگی قسمت‌ها در اولین خواند کتاب نخوانید. برای ساده‌سازی نماد، در این مشتق‌گیری از نماد $\mathrm{P_h(d)}$ برای نمایش P(D|h) استفاده خواهیم کرد.

می‌خواهیم گرادیان این تابع را بیابیم پس باید رابطه‌ی $\frac{\partial lnP_h(D)}{\partial \mathrm{w_{ijk}}}$ را به ازای تمامی مقادیر i و j و k محاسبه کنیم. با فرض اینکـه نمونـه‌های آموزشی d در مجموعه‌ی داده‌های D مستقل باشند، می‌توان نوشت،

$$\frac{\partial lnP(D|h)}{\partial \mathrm{w_{ij}}k} = \frac{\partial}{\partial \mathrm{w_{ij}}k} \ln \prod_{\mathrm{d \in D}} \mathrm{P_h(d)}$$

$$= \sum_{\mathrm{d \in D}} \frac{\partial \ln \mathrm{P_h(d)}}{\partial \mathrm{w_{ij}}k}$$

$$= \sum_{\mathrm{d \in D}} \frac{1}{\mathrm{P_h(d)}} \frac{\partial \mathrm{P_h(d)}}{\partial \mathrm{w_{ij}}k}$$

مرحله‌ی آخر از رابطه‌ی $\frac{\partial \ln f(x)}{\partial x} = \frac{1}{f(x)} \frac{\partial f(x)}{\partial x}$ نتیجه‌گیری شده است. حال می‌توان مقادیر متغیرهای Y_i و $U_i = Parents(Y_i)$ را با استفاده از جمع روی مقادیر $\mathrm{y_{ij'}}$ و $\mathrm{u_{ik'}}$ معرفی کرد.

$$\frac{\partial lnP(D|h)}{\partial \mathrm{w_{ij}}k} = \sum_{\mathrm{d \in D}} \frac{1}{\mathrm{P_h(d)}} \frac{\partial}{\partial \mathrm{w_{ij}}k} \sum_{\mathrm{j',k'}} \mathrm{P_h}\big(\mathrm{d}\big|\mathrm{y_{ij'}}, \mathrm{u_{ik'}}\big) \mathrm{P_h}(\mathrm{y_{ij'}}, \mathrm{u_{ik'}})$$

$$= \sum_{\mathrm{d \in D}} \frac{1}{\mathrm{P_h(d)}} \frac{\partial}{\partial \mathrm{w_{ij}}k} \sum_{\mathrm{j',k'}} \mathrm{P_h}\big(\mathrm{d}\big|\mathrm{y_{ij'}}, \mathrm{u_{ik'}}\big) \mathrm{P_h}\big(\mathrm{y_{ij'}}\big|\mathrm{u_{ik'}}\big) \mathrm{P_h}(\mathrm{u_{ik'}})$$

مرحله‌ی آخر از قانون احتمال جدول ۶٫۱ نتیجه‌گیری شده است. حال جمع سمت راستی رابطه‌ی بالـا را در نظـر بگیریـد، بـا توجـه بـه تعریـف $w_{ijk} \equiv P_h(y_{ij}|u_{ik})$ خواهیم داشت که تمامی جملات جز جمله‌ی j'=j و i'=i صفر خواهند بود پس داریم،

$$\frac{\partial lnP(D|h)}{\partial \mathrm{w_{ij}}k} = \sum_{\mathrm{d \in D}} \frac{1}{\mathrm{P_h(d)}} \frac{\partial}{\partial \mathrm{w_{ij}}k} \sum_{\mathrm{j',k'}} \mathrm{P_h}\big(\mathrm{d}\big|\mathrm{y_{ij}}, \mathrm{u_{ik}}\big) \mathrm{P_h}\big(\mathrm{y_{ij}}\big|\mathrm{u_{ik}}\big) \mathrm{P_h}(\mathrm{u_{ik}})$$

$$= \sum_{\mathrm{d \in D}} \frac{1}{\mathrm{P_h(d)}} \frac{\partial}{\partial \mathrm{w_{ij}}k} \sum_{\mathrm{j',k'}} \mathrm{P_h}\big(\mathrm{d}\big|\mathrm{y_{ij}}, \mathrm{u_{ik}}\big) \mathrm{w_{ijk}} \mathrm{P_h}(\mathrm{u_{ik}})$$

$$= \sum_{\mathrm{d \in D}} \frac{1}{\mathrm{P_h(d)}} \mathrm{P_h}\big(\mathrm{d}\big|\mathrm{y_{ij}}, \mathrm{u_{ik}}\big) \mathrm{P_h}(\mathrm{u_{ik}})$$

با استفاده از قضیه‌ی بیز برای مقدار $\mathrm{P_h}\big(\mathrm{d}\big|\mathrm{y_{ij}}, \mathrm{u_{ik}}\big)$ داریم،

$$\frac{\partial lnP(D|h)}{\partial \mathrm{w_{ij}}k} = \sum_{\mathrm{d \in D}} \frac{1}{\mathrm{P_h(d)}} \frac{\mathrm{P_h}\big(\mathrm{y_{ij}}, \mathrm{u_{ik}}\big|\mathrm{d}\big) \mathrm{P_h(d)} \mathrm{P_h}(\mathrm{u_{ik}})}{\mathrm{P_h}\big(\mathrm{y_{ij}}, \mathrm{u_{ik}}\big)}$$

$$= \sum_{d \in D} \frac{P_h(y_{ij}, u_{ik}|d) P_h(u_{ik})}{P_h(y_{ij}, u_{ik})}$$

$$= \sum_{d \in D} \frac{P_h(y_{ij}, u_{ik}|d)}{P_h(y_{ij}|u_{ik})}$$

$$= \sum_{d \in D} \frac{P_h(y_{ij}, u_{ik}|d)}{w_{ijk}} \quad (6.26)$$

بدین صورت مشتق رابطه‌ی ۶٫۲۵ محاسبه می‌شود. قبل از رفتن به سراغ قانون فرایند شیب صعود باید در نظر گرفت که باید پس از تغییر مقادیر w_{ijk} آن‌ها همچنان در بازه‌ی [0,1] باقی بمانند تا احتمالات معتبری باشند. از طرف دیگر باید مقدار $\sum_j w_{ijk}$ برای تمامی مقادیر i و k، ۱ باقی بماند. این شروط را می‌توان با تغییر دومرحله‌ای وزن‌ها اعمال کرد، ابتدا هر w_{ijk} را با توجه به شیب صعود تغییر می دهیم،

$$w_{ijk} \leftarrow w_{ijk} + \eta \sum_{d \in D} \frac{P_h(y_{ij}, u_{ik}|d)}{w_{ijk}}$$

در این رابطه η ثابت کوچکی به نام ضریب یادگیری است. در مرحله‌ی دوم وزن‌ها را نرمالیزه می‌کنیم تا در شروط بالا صدق کنند. همان طور که Russell نیز توضیح داده است این فرایند به محتمل‌ترین فرضیه‌ی نسبی برای احتمالات شرطی در شبکه بیز میل خواهد کرد.

مثل دیگر روش‌های بر پایه‌ی شیب صعود، این الگوریتم نیز فقط تضمین می‌کند که راه‌حل بهینه‌ی موضعی پیدا کند. جایگزین دیگر موجود برای شیب صعود الگوریتم EM است که در قسمت ۶٫۱۲ توضیح داده می‌شود، این الگوریتم نیز راه‌حلی بهینه موضعی پیدا خواهد کرد.

٦,١١,٦ یادگیری ساختار شبکه‌ی بیزی

یادگیری شبکه‌های بیزی هنگامی که ساختار شبکه به دقت معلوم نیست نیز پیچیده است. (Cooper and Herskovits 1992) روشی Bayesian scoring metric برای انتخاب میان شبکه‌های مختلف ارائه می‌کنند. آن‌ها همچنین جستجویی ابتکاری به نام الگوریتم K2 برای یادگیری ساختار شبکه در شرایطی که داده‌ها به طور کامل قابل‌مشاهده‌اند ارائه می‌کنند. مشابه اکثر الگوریتم‌های یادگیری ساختار شبکه‌ی بیز، K2 نیز از جستجویی حریصانه که پیچیدگی فرضیه را فدای دقت روی داده‌های آموزشی می‌کند استفاده می‌کند. در آزمایشی به K2 مجموعه‌ای از 3,000 نمونه‌ی آموزشی تصادفی از شبکه بیزی‌ای معلومی با ۳۷ گره و ۴۶ یال داده شد. این شبکه‌ی خاص مشکلات بیهوشی را در یک اتاق جراحی بیمارستان توصیف می‌کرد. علاوه بر این داده‌ها، به برنامه ترتیبی اولیه‌ای از ۳۷ متغیری که سازگار با قسمتی از ترتیب وابستگی متغیرها در شبکه‌ی واقعی بود نیز داده شد. این برنامه در تشخیص شبکه‌ی بیزی درست تقریباً موفق شد، این شبکه یالی اضافه و یالی دیگر کمتر از شبکه‌ی اصلی داشت.

روش‌های مبتنی بر قیود[۳۴] نیز در یادگیری ساختار شبکه‌های بیزی نیز پیشنهاد شده است (Sprites et al. 1993). این روش‌ها روابط استقلال و وابستگی را از داده‌ها استنتاج کرده و از آن‌ها برای ساخت شبکه‌های بیزی استفاده می‌کنند. بررسی مربوطه‌ی روش‌های فعلی یادگیری ساختار شبکه‌های بیزی در (Heckerman 1995) و (Buntine 1994) آورده شده است.

٦,١٢ الگوریتم EM

در بسیاری از تعریف مسئله‌های کاربردی، فقط یک زیرمجموعه از ویژگی‌های نمونه‌ها قابل‌مشاهده است. برای مثال، در یادگیری یا استفاده‌ی شبکه‌ی باور بیزی‌ای که در جدول ۶,۳ آورده شد، ممکن است فقط داده‌های نظیر یک زیرمجموعه از متغیرهای شبکه مثل زیرمجموعه‌ی Storm, Lightning, Thunder, ForestFire, Campfire, BusTourGroup را داشته باشیم. روش‌های بسیاری برای کنترل این مشکل پیشنهاد شده است، همان طور که در فصل ۳ نیز دیدید، اگر بعضی متغیرها در بعضی موارد غیرقابل‌مشاهده و در بعضی موارد قابل‌مشاهده باشند، می‌توان از نمونه‌های آموزشی‌ای که این مقدار را دارند برای پیش‌بینی این ویژگی در دیگر نمونه‌ها استفاده کرد. در این بخش به الگوریتم EM که در یادگیری با وجود ویژگی‌های مجهول کاربرد زیاد دارد می‌پردازیم. از الگوریتم EM می‌توان حتی برای متغیرهایی که هیچ وقت به طور مستقیم قابل‌مشاهده نیستند نیز استفاده کرد، اما لازم است که فرم کلی توزیع احتمال حاکم بر این متغیرها معلوم باشد. الگوریتم EM برای آموزش شبکه‌های باور بیزی (برای اطلاعات بیشتر به Heckerman 1995 رجوع کنید) و شبکه‌های توابع شعاعی[۳۵] که در قسمت ۸,۴ توضیح داده شد به کار می‌روند. الگوریتم EM پایه‌ی بسیاری از الگوریتم‌های خوشه یابی[۳۶] (Cheeseman 1988) و همچنین پایه‌ی الگوریتم‌های پرکاربرد Baum-Welch forward-backward برای یادگیری مدل‌های نیمه مشهود مارکوف[۳۷] است (Rabiner 1989).

٦,١٢,١ تخمین میانگین k توزیع نرمال

راحت‌ترین راه معرفی الگوریتم EM از طریق یک مثال است. مسئله‌ای را در نظر بگیرید که در آن داده‌های آموزشی D مجموعه‌ای از نمونه‌هایی است که از طریق توزیعی که ترکیب k توزیع نرمال[۳۸] است به دست آمده‌اند. این تعریف مسئله برای k=2 در شکل ۶,۴ آمده است، در این شکل نمونه‌ها نقاط روی محور x هستند. هر نمونه از فرایندی دومرحله‌ای به دست می‌آید. ابتدا به تصادف یکی از k توزیع نرمال انتخاب می‌شود. سپس بر اساس آن توزیع نرمال نمونه‌ی x_i ایجاد می‌گردد. این فرایند برای ایجاد مجموعه‌ای از نمونه‌های آموزشی همان طور که در شکل نشان داده شده است تکرار خواهد شد. برای ساده‌سازی بحث، حالتی را بررسی می‌کنیم که احتمال تمامی توزیع‌های نرمال در مرحله‌ی اول یکسان است و تمامی توزیع‌های نرمال واریانس مشترک σ^2 دارند. هدف یادگیری پیدا کردن فرضیه‌ای به شکل $h = <\mu_1, \ldots, \mu_k>$ است که میانگین‌های k توضیح احتمال را توصیف کند. در کار یادگیری سعی می‌کنیم تا محتمل‌ترین فرضیه را پیدا کنیم؛ فرضیه‌ای که p(D|h) را ماکزیمم کند.

---

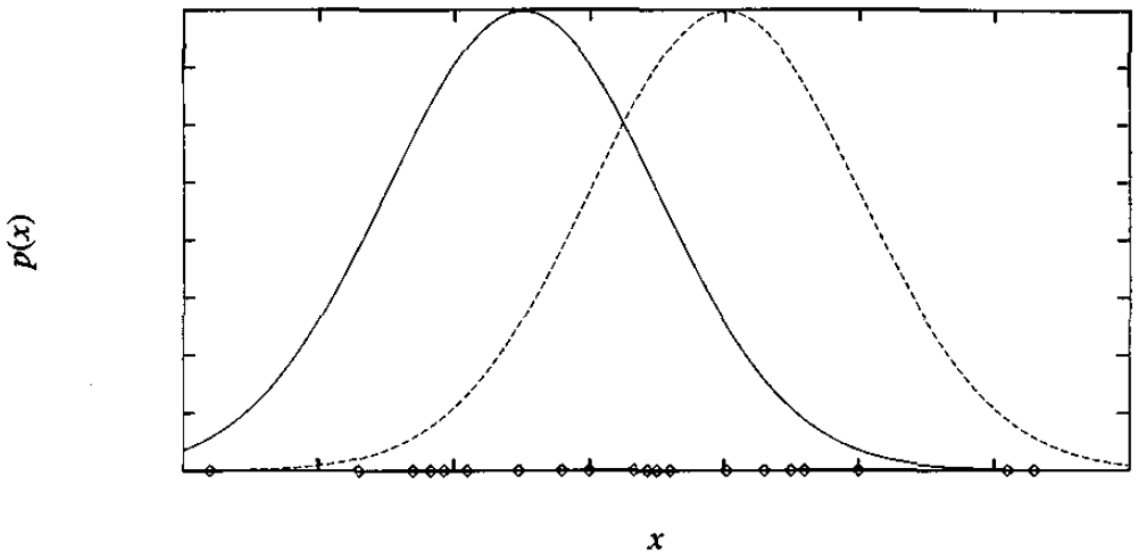
۳۴ constraint-based

۳۵ Radial basis function network

۳۶ clustering

۳۷ Partially Observable Markov Models

۳۸ Mixed Gaussian distribution



شکل ۶,۴ نمونه‌های حاصل از ترکیب دو توزیع نرمال با واریانس σ یکسان.
نمونه‌ها با نقاطی روی محور x نشان داده شده‌اند. اگر میانگین توزیع‌های نرمال نامعلوم باشد، الگوریتم EM را می‌توان برای جستجوی محتمل‌ترین مقدار تخمین آن‌ها به کاربرد.

توجه دارید که محاسبه‌ی محتمل‌ترین فرضیه برای میانگین یک توزیع نرمال با داشتن نمونه‌های $x_1, x_2, \dots, x_m$ فقط حالت خاصی از مسئله‌ای است که در قسمت ۶,۴ بحث شد، در رابطه‌ی ۶,۶ نشان دادیم که محتمل‌ترین فرضیه، فرضیه‌ای است که مجموع خطاهای مربعی را برای تمامی m نمونه مینیمم می‌کند. اگر رابطه‌ی ۶,۶ را با توجه به نمادگذاری جدید بازنویسی کنیم، خواهیم داشت،

$$\mu_{ML} = \arg\max_{\mu} \sum_{i=1}^{m} (x_i - \mu)^2 \tag{6.27}$$

در چنین شرایطی مجموع خطاهای مربعی با تساوی زیر مینیمم خواهد شد،

$$\mu_{ML} = \frac{1}{m} \sum_{i=1}^{m} x_i \tag{6.28}$$

با وجود این وجود مسئله‌ی ما درباره‌ی ترکیبی از توابع نرمال است و تشخیص اینکه نمونه‌ها از کدام توزیع به دست آمده‌اند نیز ممکن نیست. بنابراین، با صورت مثالی کلی از مسئله‌هایی که متغیرهای پنهان دارند مواجهیم. در مثال شکل ۶,۴ می‌توان توضیح کامل مربوطه‌ی هر نمونه را به شکل سه‌تایی مرتب $< x_i, z_{i1}, z_{i2} >$ در نظر گرفت، در این سه‌تایی مرتب x_i مقدار مشاهده شده‌ی iامین نمونه است و دو مقدار z_{i1}, z_{i2} مشخص می‌کند که کدام یک از دو توزیع نرمال برای تولید این نمونه به کار رفته‌اند. در کل، z_{ij} زمانی یک است که نمونه از توزیع نرمال jام به دست آمده است و در غیر این صورت صفر خواهد بود. در اینجا متغیر x_i قابل‌مشاهده و متغیرهای z_{i1}, z_{i1} متغیرهای پنهان هستند. اگر مقادیر متغیرهای z_{i1}, z_{i1} قابل‌مشاهده بودند می‌شد از رابطه‌ی ۶,۲۷ برای پیدا کرد μ_1 و μ_2 استفاده کرد، حال چون این متغیرها قابل‌مشاهده نیستند از الگوریتم EM استفاده خواهیم کرد.

در این مثال، پیدا کردن k میانگین، الگوریتم EM به تخمین مقادیر z_{ij} با معلوم بودن فرضیه‌ی فعلی $< \mu_1 \dots \mu_k >$ می‌پردازد و سپس مقادیر محتمل‌ترین فرضیه‌ها را با توجه به این مقادیر تصادفی برای متغیرهای پنهان دوباره محاسبه می‌کند. ابتدا این مثال را در الگوریتم EM توصیف حل می‌کنیم، الگوریتم EM را در حالت کلی بیان خواهیم کرد.

برای شکل ۶،۴ الگوریتم EM ابتدا مقدار اولیه‌ی فرضیه را به $h =< \mu_1, \mu_2 >$ که در آن μ_1 و μ_2 دو مقدار دلخواه هستند مقداردهی اولیه می‌کند. سپس فرضیه‌ی h را با تکرار حلقه‌ی دومرحله‌ای زیر ارزیابی می‌کند، این حلقه تا زمانی که فرایند به مقدار ثابتی از h همگرا شود حلقه تکرار خواهد شد.

مرحله‌ی ۱: مقدار امید $E[z_{ij}]$ را برای هر متغیر پنهان z_{ij} با فرض درستی فرضیه‌ی $h =< \mu_1, \mu_2 >$ محاسبه کن.

مرحله‌ی ۲: محتمل‌ترین فرضیه‌ی جدید $h' =< \mu_1', \mu_2' >$ را با فرض اینکه تمامی مقادیر z_{ij} مقدار امید $E[z_{ij}]$ که در مرحله‌ی ۱ محاسبه شد است را محاسبه کن. سپس فرضیه‌ی $h =< \mu_1, \mu_2 >$ را با فرضیه‌ی $h' =< \mu_1', \mu_2' >$ جایگزین کن.

بیایید نحوه‌ی پیاده‌سازی هر یک از مراحل را در عمل بررسی کنیم. مرحله‌ی اول باید مقدار امید هر یک از z_{ij} را محاسبه کند. این مقدار $E[z_{ij}]$ فقط احتمال نمونه‌ی x_i است که از طریق j امین توزیع نرمال ایجاد شده است.

$$E[z_{ij}] = \frac{p(x = x_i | \mu = \mu_j)}{\sum_{n=1}^{2} p(x = x_i | \mu = \mu_n)}$$

$$= \frac{e^{-\frac{1}{2\sigma^2}(x_i - \mu_j)^2}}{\sum_{n=1}^{2} e^{-\frac{1}{2\sigma^2}(x_i - \mu_n)^2}}$$

مرحله‌ی اول با بکار گیری مقادیر فعلی $< \mu_1, \mu_2 >$ و مقدار فعلی x_i نتیجه‌گیری شده است.

در مرحله‌ی دوم مقداری که برای $E[z_{ij}]$ در مرحله‌ی اول محاسبه شد استفاده می‌شود تا محتمل‌ترین فرضیه $h' =< \mu_1', \mu_2' >$ محاسبه شود. همان طور که بعداً نیز بحث خواهیم کرد، محتمل‌ترین فرضیه در این حالت با رابطه‌ی زیر محاسبه می‌شود،

$$\mu_j \leftarrow \frac{\sum_{i=1}^{m} E[z_{ij}] x_i}{\sum_{i=1}^{m} E[z_{ij}]}$$

توجه دارید که این رابطه تشابه بسیاری به رابطه‌ی ۶،۲۸ برای تخمین مقدار μ برای یک توزیع نرمال دارد. در رابطه‌ی جدید فقط میانگین وزن‌دار μ_j هاست، وزن هر μ_j مقدار $E[z_{ij}]$ که از j امین توزیع نرمال به دست آمده است.

الگوریتم بالا برای تخمین میانگین‌های ترکیب k توزیع نرمال با روش الگوریتم EM است: فرضیه‌ی فعلی برای تخمین متغیرهای نامشهود استفاده می‌شود، سپس مقدار امید این متغیرها برای محاسبه‌ی فرضیه‌ی بهتری به کار می‌رود. می‌توان اثبات کرد که با هر دور اجرای حلقه الگوریتم EM میزان محتمل بودن P(D|h) را بیشتر می‌کند، مگر اینکه آن یک ماکزیمم نسبی باشد. پس الگوریتم EM در انتها به یک ماکزیمم موضعی برای محتمل بودن $< \mu_1, \mu_2 >$ میل خواهد کرد.

۶,۱۲,۲ حالت کلی الگوریتم EM

در بالا الگوریتم EM را برای مسئله‌ی تخمین میانگین‌های ترکیب توزیع احتمال‌های نرمال بیان کـردیم. در حالـت کلـی‌تـر، الگـوریتم EM را می‌توان در بسیاری از تعریف مسئله‌ها که در آن‌ها بحث از تخمین مجموعه‌ای از پارامترهای θ که توضیح احتمال حاکم را توصیف می‌کننـد بـا استفاده از قسمتی از داده‌ها که قابل شهود است به کاربرد. در مثال دو میانگین بالا پارامترهای مورد علاقه $\theta =< \mu_1, \mu_2 >$ است، داده‌های کامل سه‌تایی مرتب‌های $< x_i, z_{i1}, z_{i2} >$ هستند که فقط x_i قابل‌مشاهده است. در کل اگـر $X = \{x_1, \dots, x_m\}$ داده‌هـای مشـاهده شده در مجموعه‌ای از m نمونه‌ی مستقل و $Z = \{z_1, \dots, z_m\}$ نیز داده‌های غیرقابل‌مشاهده باشد در ایـن نمونـه‌های آموزشـی Y=XUZ کل داده‌ها خواهد بود. توجه دارید که با Z می‌توان به دید متغیر تصادفی‌ای که توزیع احتمالش به پارامترهـای نـامعلوم θ و داده‌هـای مشـاهده شده‌ی X وابسته است نگاه کرد. به طور مشابه، Y نیز متغیری تصادفی است، زیرا که توسط متغیر تصادفی Z تعریف می‌شـود. در ادامـه‌ی ایـن بخش فرم کلی الگوریتم EM را توضیح خواهیم داد. از h برای نمایش مقادیر مفروض فعلی از پارامترهای θ و از 'h بـرای فرضـیه‌ی بـازبینی شده‌ی هر حلقه الگوریتم EM استفاده خواهیم کرد.

الگوریتم EM فضای فرضیه‌ی محتمل‌ترین فرضیه‌ها 'h را برای پیدا کردن 'h ی که مقدار $E[\ln P(Y|h')|h]$ را ماکزیمم کنـد جسـتجو می‌کند. این مقدار امید برای تمامی توزیع احتمالات Y که توسط پارامترهای نامعلوم مشخص می‌گردد محاسبه می‌شود. بیایید مفهوم دقیق ایـن مقدار امید را با هم بررسی کنیم. ابتدا اینکه $P(Y|h')$ محتمل بودن داده‌های کامل Y را با شرط معلوم بودن 'h نشـان مـی‌دهـد. پـس پیـدا کردن فرضیه‌ی 'h به قسمی که تابعی از این معیار را ماکزیمم کنـد منطقـی خواهـد بـود. دوم اینکـه، مـاکزیمم کـردن لگـاریتم ایـن کمیـت، $ln\ P(Y|h')$ نیز $P(Y|h')$ را ماکزیمم می‌کند (همان طور که پیش‌تر نیز گفته بودیم). سوم اینکه ما مقـدار امیـد $E[ln\ P(Y|h')]$ را برای اینکه داده‌های کامل Y خود متغیری تصادفی است معرفی می‌کنیم. با دانستن اینکه داده‌های کامل Y ترکیبی از داده‌های مشاهده شـده‌ی X و داده‌های مشاهده نشده‌ی Z است، باید میانگین را برای مقادیر ممکن Zهای مشاهده نشده با وزن متناسب با احتمالشان محاسبه کنـیم. بـه عبارت دیگر، مقدار امید $E[ln\ P(Y|h')]$ بر روی توزیع احتمالات تصادفی Y محاسبه می‌شود. توزیع Y توسط مقادیر کاملاً معلوم X و توزیع احتمال حاکم بر Z تعیین می‌شود.

توزیع احتمال حاکم بر Y چیست؟ در کل این توزیع را نمی‌دانیم، زیرا که این توزیع با پارامترهای θ که می‌خواهیم تخمین بزنیم تعیین می‌شـود. بنابراین، الگوریتم EM از فرضیه‌ی فعلی h به جای پارامترهای واقعی θ برای تخمین توزیع احتمال حاکم بر Y استفاده می‌کند. بیایید تابعی به فرم $Q(h'|h)$ تعریف کنیم که مقدار $E[\ln P(Y|h')|h]$ را به عنوان تابعی از h' با فرض θ=h و داشتن قسـمت قابل‌مشـاهده‌ی X از کل داده‌های Y بیان کند.

$$Q(h'|h) = E[\ln P(Y|h')|h, X]$$

این تابع Q را به فرم $Q(h'|h)$ می‌نویسیم تا نشان دهد که این تابع با این فرض تعریف شده که فرضیه‌ی فعلی h با θ مساوی است. در فرم کلی، الگوریتم EM تا رسیدن به همگرایی دو مرحله‌ی زیر را تکرار می‌کند:

مرحله‌ی ۱: مرحله‌ی تخمین (E): مقدار Q(h'|h) را با استفاده از فرضیه‌ی فعلی h و داده‌هـای مشـاهده شـده‌ی X بـرای تخمـین توزیـع احتمال روی Y محاسبه کن.

$$Q(h'|h) = E[\ln p(Y|h')\ |h, X]$$

مرحله‌ی ۲: مرحله‌ی ماکزیمم سازی (M): فرضیه‌ی h را با فرضیه‌ی h' که مقدار Q را ماکزیمم می‌کند جایگزین کن.

$$h \leftarrow \arg\max_{h'} Q(h'|h)$$

اگر تابع Q پیوسته باشد، الگوریتم EM به نقطه تعادل محتمل‌ترین فرضیه‌ی P(Y|h') میل خواهد کرد. اگر این تابع محتمل بودن فقط یک ماکزیمم داشته باشد، EM نیز به همان تخمین همان ماکزیمم مطلق برای h' میل خواهد کرد. در غیر این صورت، این الگوریتم تضمین می‌کند تا به ماکزیممی موضعی میل کند. در چنین شرایطی، EM محدودیت‌های الگوریتم‌های دیگری که از جستجوی شیب نزول استفاده می‌کنند را خواهد داشت، در فصل ۴ توضیح کاملی در مورد این مشکلات و راه‌حل‌های آن‌ها آورده شده است.

۶,۱۲,۳ اشتقاق الگوریتم k میانگین

برای تصور بهتر کلی الگوریتم EM، بیایید مشتق الگوریتم آورده شده در قسمت ۶,۱۲,۱ برای تخمین میانگین‌های k توزیع نرمال را بررسی کنیم. همان طور که در بالا نیز توضیح داده شد، مسئله‌ی تخمین k میانگین، مسئله‌ی تخمین پارامترهای $\theta =< \mu_1 \ldots \mu_k >$ است که میانگین k توزیع نرمال تعریف می‌شوند. داده‌های مشاهده شده‌ی $X = \{< x_i >\}$ به ما داده شده‌اند. در این مسئله متغیرهای پنهان $Z = \{z_{i1}, \ldots, z_{ik}\}$ هستند که مشخص می‌کنند که نمونه با استفاده از کدام توزیع ایجاد شده است.

برای به کار بردن EM ابتدا باید مشتق Q(h|h') را برای این تخمین k میانگین پیدا کرد. بیایید ابتدا مشتق رابطه‌ی p(Y|h') را محاسبه کنیم. توجه دارید که احتمال $p(y_i|h')$ برای یک تک نمونه‌ی $y_i =< x_i, z_{i1}, \ldots, z_{ik} >$ از داده‌های کامل را می‌توان به صورت زیر دقیق محاسبه کرد.

$$p(y_i|h') = p(x_i, z_{i1}, \ldots, z_{ik}|h') = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{1}{2\sigma^2}\sum_{j=1}^{k} z_{ij}\left(x_i-\mu_j'\right)^2}$$

توجه داشته باشید که از تمامی z_{ij} ها فقط یکی ۱ است و بقیه ۰ هستند. بنابراین این رابطه توزیع احتمال x_i را بر اساس توزیع نرمال انتخابی نشان می‌دهد. با داشتن احتمال تک نمونه، $p(y_i|h')$، لگاریتم احتمال P(Y|h') برای تمامی m نمونه در داده‌ها به صورت زیر خواهد بود،

$$\ln P(Y|h') = ln \prod_{i=1}^{m} p(y_i|h')$$

$$= \sum_{i=1}^{m} \ln p(y_i|h')$$

$$= \sum_{i=1}^{m} \left(ln \frac{1}{\sqrt{2\pi\sigma^2}} - \frac{1}{2\sigma^2} \sum_{j=1}^{k} z_{ij}\left(x_i - \mu_j'\right)^2 \right)$$

حال می‌توان بالاخره مقدار امید ln P(Y|h') را بر اساس توزیع احتمال حاکم بر Y یا به طور مشابه توزیع احتمال حاکم بر متغیرهای غیرقابل‌مشاهده‌ی Y (z_{ij} ها) محاسبه کرد. توجه دارید که عبارت بالا برای ln P(Y|h') تابعی خطی از z_{ij}هاست. در کل برای هر تابع f(z) که تابعی خطی از z است رابطه‌ی زیر درست است،

$$E[f(z)] = f(E[z])$$

با استفاده از حقیقت بالا درباره‌ی توابع خطی می‌توان نوشت،

$$E[\ln P(Y|h')] = E\left[\sum_{i=1}^{m}\left(ln\frac{1}{\sqrt{2\pi\sigma^2}} - \frac{1}{2\sigma^2}\sum_{j=1}^{k} z_{ij}\left(x_i - \mu_j'\right)^2\right)\right]$$

$$= \sum_{i=1}^{m}\left(ln\frac{1}{\sqrt{2\pi\sigma^2}} - \frac{1}{2\sigma^2}\sum_{j=1}^{k} E[z_{ij}]\left(x_i - \mu_j'\right)^2\right)$$

برای خلاصه‌سازی تابع Q(h'|h) در مسئله‌ی k میانگین به صورت زیر است،

$$Q(h'|h) = \sum_{i=1}^{m}\left(ln\frac{1}{\sqrt{2\pi\sigma^2}} - \frac{1}{2\sigma^2}\sum_{j=1}^{k} E[z_{ij}]\left(x_i - \mu_j'\right)^2\right)$$

در این رابطه $h' =< \mu_1', \dots, \mu_k' >$ و $E[z_{ij}]$ نیز بر اساس فرضیه‌ی فعلی h و داده‌های مشاهده شده‌ی X محاسبه می‌شود. همان طور که قبلاً نیز نشان دادیم،

$$E\left[z_{ij}\right] = \frac{e^{-\frac{1}{2\sigma^2}\left(x_i - \mu_j\right)^2}}{\sum_{n=1}^{2} e^{-\frac{1}{2\sigma^2}(x_i - \mu_n)^2}} \qquad (6.29)$$

بنابراین مرحله اول (تخمین) الگوریتم EM تابع Q را بر اساس تخمین $E[z_{ij}]$ تعریف می‌کند. مرحله‌ی دوم (ماکزیمم سازی) نیز مقادیر $\mu_1', \dots, \mu_k'$ را پیدا خواهد کرد که این تابع Q را ماکزیمم کند. در مثال فعلی داریم که،

$$\arg\max_{h'} Q(h'|h) = \arg\max_{h'} \sum_{i=1}^{m}\left(ln\frac{1}{\sqrt{2\pi\sigma^2}} - \frac{1}{2\sigma^2}\sum_{j=1}^{k} E[z_{ij}]\left(x_i - \mu_j'\right)^2\right)$$

$$= \arg\min_{h'} \sum_{i=1}^{m}\sum_{j=1}^{k} E[z_{ij}]\left(x_i - \mu_j'\right)^2 \qquad (6.30)$$

محتمل‌ترین فرضیه اینجا مجموعی وزن‌دار از خطاهای مربعی را مینیمم می‌کند، در این خطا مربع اختلاف x_i ها با μ_j' با وزن $E[z_{ij}]$ مینیمم می‌شود. کمیت رابطه‌ی ۶,۳۰ با قرار دادن مقادیر μ_j' به صورت زیر مینیمم می‌شود،

$$\mu_j \leftarrow \frac{\sum_{i=1}^{m} E\left[z_{ij}\right] x_i}{\sum_{i=1}^{m} E\left[z_{ij}\right]} \qquad (6.31)$$

توجه دارید که روابط ۶,۲۹ و ۶,۳۱ دو مرحله‌ی الگوریتم k میانگین قسمت ۶,۱۲,۱ را توصیف می‌کنند.

۶٫۱۳ خلاصه و منابع برای مطالعه‌ی بیشتر

این فصل شامل موارد زیر می‌شود:

- متدهای بیزی پایه‌ای برای متدهای یادگیری احتمالی‌ای با دانش قبلی یا فـرض آن دربـاره‌ی احتمالـات ثانویـه فرضـیه‌ها و احتمـال مشاهده‌ی نمونه‌ها است. متدهای بیزی نسبت دادن احتمال ثانویه به هر فرضیه‌ی ممکن را بر اساس احتمالـات اولیـه‌ی مفـروض را ممکن می‌سازند.
- از متدهای بیزی می‌توان برای تعیین محتمل‌ترین فرضیه با فرض داشتن داده‌ها استفاده کرد، فرضیه‌ی MAP. این فرضیه‌ی از این جهت بهینه است که از تمامی فرضیه‌های دیگر محتمل‌تر است.
- دسته‌بندی کننده‌ی بهینه‌ی بیز پیش‌بینی تمامی فرضیه‌های ممکن را با احتمالات ثانویه‌شان ترکیب کرده محتمل‌ترین دسته‌بندی هر نمونه را به ما می‌دهد.
- دسته‌بندی کننده‌ی ساده‌ی بیز متد یادگیری بیزی است که در بسیاری از موارد کاربردی مفید شناخته شده است. به این الگوریتم بـه این دلیل "ساده" می‌گویند که شامل این فرض ساده کننده است که ویژگی‌های نمونه‌ها با فرض داشتن دسته‌بندی نمونه مستقل‌اند. با این فرض، دسته‌بندی کننده‌ی ساده‌ی بیز یک فرضیه MAP را خروجی خواهد داد. حتی زمانی که این فرض هم درسـت نیسـت هم، مثل حالتی که از این دسته‌بندی کننده برای دسته‌بندی متون استفاده کردیم، گاهی دسته‌بندی کننده‌ی ساده بیزی مـؤثر اسـت. شبکه‌های بیزی نمایش بهتری نسبت به مجموعه فرض‌ها استقلال در بین زیرمجموعه‌ای از ویژگی‌ها دارند.
- چهارچوب استدلال بیزی می‌تواند پایه مناسبی برای بررسی متدهای یادگیری بخصوص که مستقیماً از قضیه‌ی بیز استفاده نمی‌کننـد باشد. برای مثال در شرایط خاص می‌توان نشان داد که زمانی که تابع هدف حقیقی مقداری را بـا مینـیمم کـردن مجمـوع خطاهـای مربعی یاد می‌گیریم، محتمل‌ترین فرضیه را یاد می‌گیریم.
- قانون حداقل طول توضیح توصیه می‌کند که فرضیه‌هایی را انتخاب کنیم که کمترین طول توضیح برای فرضـیه بـه اضـافه‌ی طـول توضیحات همراه فرضیه را داشته باشد. قضیه‌ی بیز و نتایج پایه‌ای تئوری اطلاعات را می‌توان برای ایجاد دلیلی برای ایـن قـانون بـه کار برد.
- در بسیاری از کارهای یادگیری عملی، بعضی از ویژگی‌های نمونه‌های ممکن است قابل‌مشاهده نباشد. الگـوریتم EM روش کلـی‌ای برای یادگیری در حضور متغیرهای غیر مشهود ارائه می‌کند. این الگوریتم کار خود را با مجموعه‌ای از فرضیه‌های دلخواه آغاز می‌کند. سپس مقدار امید متغیر نامشهود را محاسبه کرده (با این فرض که فرضیه فعلی درست است). و سپس مقدار محتمـل‌ترین فرضـیه را محاسبه می‌کند (با فرض اینکه متغیرهای پنهان همان مقادیر امید محاسبه شده‌ی این مرحله هستند). تکـرار ایـن فراینـد بـه یـک ماکزیمم نسبی در احتمال درستی فرضیه میل می‌کند و مقادیر متغیرهای پنهان را نیز تقریب می‌زند.

کتب آموزشی ساده‌ی بسیاری درباره‌ی احتمالات و آمار مثل Casella and Berger (1990) نوشته شده است. کتب مرجع سریع بسـیاری نیز مثل Maisel (1971) و Speigel (1991) نوشته شده، این کتب نمادگذاری آمار و احتمال متناسب با یـادگیری ماشـین را نیـز ارائـه می‌کنند.

بسیاری از نمادگذاری‌های ابتدایی دسته‌بندی کننده‌های بیزی و دسته‌بندی کننده‌های مینیم خطای مربعـی در Duda and Hart (1973) بررسی شده است. Domingos and Pazzani (1996) شرایط اینکه دسته‌بندی کننـده‌ی سـاده‌ی بیـز، دسـته‌بندی بهینـه را خروجـی

می‌دهد تحلیل می‌کند، این بررسی در حالتی انجام شده که شرط استقلال دسته‌بندی کننده‌ی ساده‌ی بیز ممکن است درست نباشد (نکته در این است که شروطی وجود دارد که دسته‌بندی درست باشد اما احتمالات ثانویه درست نباشند.)

Cestnik (1990) بحث درباره‌ی استفاده از تخمین m برای دسته‌بندی احتمالات را مطرح می‌کند.

نتایج تجربی که از مقایسه‌ی روش‌های مختلف بیزی و درخت تصمیم و دیگر الگوریتم‌های یادگیری انجام شده در Michie et al. (1994) آورده شده است. Chauvin and Rumelhart (1995) بررسی بیزی شبکه‌های عصبی را که بر اساس الگوریتم backpropagation است را مطرح می‌کنند.

بحث بر روی قانون کمترین طول توضیح را می‌توانید در Rissanen (1983,1989) بیابید. Quinlan and Rivest (1989) نیز استفاده از این قانون را در اجتناب از overfit در درخت‌های تصمیم را بررسی می‌کنند.

تمرینات

۶,۱ دوباره مثال عملی قانون بیز در قسمت ۶,۲,۱ را در نظر بگیرید. فرض کنید که دکتر تصمیم می‌گیرد که دستور دهد که آزمایش دومی انجام شود، و فرض کنید که نتیجه‌ی آزمایش دوم نیز مثبت است. احتمال ثانویه‌ی cancer¬ و cancer را محاسبه کنید. فرض کنید که دو تست مستقل‌اند.

۶,۲ در مثال قسمت ۶,۲,۱ احتمال ثانویه‌ی cancer را با نرمالیزه کردن P(+|cancer).P(cancer) و P(+|¬cancer).P(¬cancer) به صورتی که مجموعشان یک شود محاسبه کردیم. از قضیه‌ی بیز قضیه مجموع احتمالات (با توجه به جدول ۶,۱) برای اثبات این متد استفاده کنید. (ثابت کنید که نرمالیزه کرده به این صورت مقدار درستی برای P(cancer|+) ارائه می‌کند).

۶,۳ الگوریتم یادگیری مفهوم FindG را در نظر بگیرید، که کلی‌ترین فرضیه‌ی ممکن ساخته شده از فرضیه‌ها را ارائه می‌کند (کلی‌ترین اعضای فضای فرضیه‌ای متناسب با نمونه‌های آموزشی).

(a) توزیعی برای P(h) و P(D|h) ارائه کنید (فرض کنید FindG تضمین می‌کند که همیشه فرضیه‌ای MAP خروجی دهد)

(b) توزیعی برای P(h) و P(D|h) ارائه کنید (فرض کنید FindG تضمین نمی‌کند که همیشه فرضیه‌ای MAP خروجی دهد)

(c) توزیعی برای P(h) و P(D|h) ارائه کنید (فرض کنید FindG تضمین نمی‌کند که همیشه فرضیه‌ای ML خروجی دهد)

۶,۴ در بررسی یادگیری مفهوم در بخش ۶,۳ فرض کردیم که ترتیب نمونه‌های $< x_1, \dots, x_m >$ همیشه ثابت است. بنابراین برای استخراج عبارتی P(D|h) فقط کافی است که احتمال مشاهده‌ی سری‌ای از مقادیر هدف $< d_1, \dots, d_m >$ را برای این سری ثابت نمونه‌ها بررسی کنیم. حالت کلی‌تری را که در آن نمونه‌ها ثابت نیستند را در نظر بگیرید، اما فرض کنید که تمامی نمونه‌ها با توزیع مشخصی روی X انتخاب می‌شوند. داده‌های D را باید اکنون به فرم زوج‌های مرتب $\{< x_i, d_i >\}$ نشان داده و در P(D|h) باید احتمال حضور x_i ها را علاوه بر d_i دخیل کرد. نشان دهید که رابطه‌ی ۶,۵ در این حالت کلی‌تر نیز درست است. (راهنمایی: بررسی رابطه‌ی ۶,۵ را نیز در نظر بگیرید)

۶,۵ قانون کمترین طول توضیح را در نظر بگیرید که به فضای فرضیه‌ای H ی که شامل عطف n متغیر منطقی است اعمال می‌شود. واضح است که هر فرضیه به سادگی با ویژگی‌های موجود در فرضیه توصیف می‌شود، اگر تعداد بیت‌های لازم برای هر یک از متغیرها $\log_2 n$ است. فرض کنید که کد سازی هر نمونه با داشتن فرضیه نیاز به صفر بیت دارد اگر نمونه سازگار با فرضیه باشد و نیاز به $\log_2 m$ بیت دارد اگر نمونه با فرضیه سازگار نباشد، m تعداد نمونه‌هایی است که اشتباه دسته‌بندی می‌شوند. (برای تعیین اینکه کدام یک از m نمونه‌ی اشتباه دسته‌بندی شده، دسته‌بندی درست را می‌توان به نقیض آنچه فرضیه دسته‌بندی می‌کند دانست)

(a) رابطه‌ی لازم برای کمیتی که باید بنا بر قانون کمترین طول مینیمم شود را بیابید.

(b) آیا ممکن است که دسته‌ای از داده‌های آموزشی موجود باشد که فرضیه‌ای سازگار با آن‌ها وجود داشته باشد اما MDL فرضیه‌ای با سازگاری کمتر را بر گزیند؟ اگر چنین است آن مجموعه را بیابید. اگر خیر، توضیح دهید چرا.

(c) توزیع احتمال P(h) و P(D|h) را برای اینکه الگوریتم MDL فوق فرضیه‌ای MAP را خروجی دهد بیابید.

۶,۶ شبکه‌ی باور بیزی‌ای را که فرض استقلال دسته‌بندی کننده‌ی ساده‌ی بیز را برای مفهوم PlayTennis در مسئله‌ی قسمت ۶,۹,۱ بکشید. جدول احتمال شرطی مربوطه‌ی گره باد را نیز رسم کنید.

فرهنگ لغات تخصصی فصل (فارسی به انگلیسی)

احتمال اولیه	prior probability
احتمال ثانویه	posterior probability
آنتروپی دورگه	cross entropy
اندازه‌ی نمونه‌ی معادل	equivalent sample size
بدون شعور	brute-force
برآمد	Outcome
تخمین m	m estimate
تعریف مسئله	problem setting
جرم احتمال	probability mass
چگالی احتمال	probability density
دسته‌بندی کننده‌ی بهینه‌ی بیز	bayes optimal classifier
دسته‌بندی کننده‌ی ساده‌ی بیز	naive Bayes classifier
زیرین	Descendant
شبکه‌های باور بیزی	bayesian belief networks
فرضیه با حداکثر احتمال	Maximum A Posteriori
فضای توأم	joint space
قانون کمترین طول توضیح	Minimum description length
الگوریتم گیبس	Gibbs algorithm
محتمل‌ترین	maximum likelihood

Criterion	معیار
consistent learner	یادگیر سازگار
Arc	یال