کتاب راهنمای یادگیری ماشین

پردراگ رادیووژاک و مارتا وایت

فهرست

مرجع علامت گذاری

مجموعه نمادگذاری‌ها

$\mathcal{X}$ مجموعه‌ای عمومی از مقادیر. به عنوان مثال، $\mathcal{X} = \{0, 1\}$ مجموعه‌ای است که فقط شامل 0 و 1 است، $\mathcal{X} = [0, 1]$ فاصله بین 0 تا 1 و $\mathcal{X} = \mathbb{R}$ مجموعه اعداد حقیقی است. بسته به مکان استفاده آن، نمادهایی مانند A، B، Ω و موارد دیگر نیز به عنوان مجموعه استفاده خواهند شد.

$P(\mathcal{X})$ مجموعه توان $\mathcal{X}$، مجموعه‌ای شامل تمام زیر مجموعه‌های ممکن $\mathcal{X}$.

$[a, b]$ بازه بسته با شرط $a < b$، شامل a و b.

(a, b) بازه باز با شرط $a < b$، بدون a و b در مجموعه.

$(a, b]$ بازه نیم‌باز با شرط $a < b$، شامل b اما نه a.

$[a, b)$ بازه نیم‌باز با شرط $a < b$، شامل a اما نه b.

نماد برداری و ماتریس

x متغیرهای کوچک کم‌رنگ معمولاً اسکالر هستند. با این حال، هنگامی که $x \in \mathcal{X}$، جایی که $\mathcal{X}$ مشخص نشده است، x ممکن است بردار، یک شی ساختار یافته مانند گراف و غیره را نشان دهد.

$\boldsymbol{x}$ متغیرهای با حروف کوچک پررنگ بردار هستند. به طور پیش فرض، بردارها بردارهای ستونی هستند.

$\mathbf{X}$ متغیرهای با حروف بزرگ پررنگ ماتریس هستند. به نظر می‌رسد یک متغیر تصادفی چند متغیره، $\boldsymbol{X}$، اما متغیر تصادفی مورب است. اغلب از زمینه مشخص می‌شود که چه زمانی این یک متغیر تصادفی چند متغیره و چه زمانی یک ماتریس است.

$\mathbf{X}^{\mathrm{T}}$ جابجایی ماتریس. برای دو ماتریس $\boldsymbol{A}$ و $\boldsymbol{B}$، آن را برقرار می‌کند

$$(\mathbf{AB})^{\mathrm{T}} = \mathbf{B}^{\mathrm{T}}\mathbf{A}^{\mathrm{T}}$$

یک ماتریس $n \times d$ متشکل از n بردار هر یک از ابعاد d را می‌توان به صورت بیان کرد

$$\mathbf{X} = [\mathbf{x}_1\mathbf{x}_2 \ldots \mathbf{x}_n]^{\mathrm{T}}$$

$X_{i:}$ ردیف i-ام ماتریس. یک بردار سطری.

$X_{:j}$ ستون j ماتریس. یک بردار ستونی

تاپل‌ها، بردارها و دنباله‌ها

$(x_1, x_2, \ldots, x_d)$ یک تاپل; به عنوان مثال، یک لیست مرتب از عناصر d. وقتی $(x_1, x_2, \ldots, x_d) \in \mathbb{R}^d$، تاپل به عنوان بردار ستون $\mathbf{x} = [\mathrm{x_1 x_2 \ldots x_d}]^{\mathrm{T}}$ تلقی می‌شود.

$a_1, \ldots, a_m$ دنباله‌ای از m آیتم‌ها. متغیرهای شاخص روی این دنباله‌ها معمولاً متغیرهای i، j یا k هستند. به عنوان مثال، $\sum_{i=1}^{m} a_i$ یا اگر هر a_i بردار بعد d باشد، شاخص دوگانه $\sum_{i=1}^{m} \sum_{j=1}^{d} a_{ij}$ است.

نماد توابع

$f : \mathcal{X} \to \mathcal{Y}$ تابع در دامنه $\mathcal{X}$ به هم دامنه $\mathcal{Y}$ تعریف می‌شود و مقادیر $x \in \mathcal{X}$ را می‌گیرد و آنها را به $f(x) \in \mathcal{Y}$ ارسال می‌کند.

$\frac{\mathrm{dx}}{\mathrm{dx}}(\mathrm{x})$ مشتق تابع در $\mathrm{x} \in \mathcal{X}$، که در آن $\mathrm{f} : \mathcal{X} \to \mathbb{R}$ برای $\mathcal{X} \subset \mathbb{R}$.

$\nabla \mathrm{f}(\mathrm{x})$ گرادیان یک تابع در $\mathrm{x} \in \mathcal{X}$، که در آن $\mathrm{f} : \mathcal{X} \to \mathbb{R}$ برای $\mathcal{X} \subset \mathbb{R}^{\mathrm{d}}$. آن را نگه می‌دارد

$$\nabla \mathrm{f}(\mathrm{x}) = \left(\frac{\partial \mathrm{f}}{\partial \mathrm{x}_1}, \frac{\partial \mathrm{f}}{\partial \mathrm{x}_2}, \ldots, \frac{\partial \mathrm{f}}{\partial \mathrm{x}_{\mathrm{d}}}\right).$$

$H_{f(x)}$ ماتریس هسین[1] یک تابع در $\boldsymbol{x} \in \mathcal{X}$، که در آن $f : \mathcal{X} \to \mathbb{R}$ برای $\mathcal{X} \subset \mathbb{R}^d$. آن را نگه می‌دارد

$$H_{f(x)} = \begin{bmatrix} \frac{\partial^2 \mathrm{f}}{\partial \mathrm{x}_1^2} & \frac{\partial^2 f}{\partial \mathrm{x}_1 \, \partial \mathrm{x}_2} & \cdots & \frac{\partial^2 f}{\partial x_1 \partial x_{\mathrm{d}}} \\ \frac{\partial^2 f}{\partial x_2 \partial x_1} & \frac{\partial^2 f}{\partial x_2^2} & & \\ \vdots & & \ddots & \\ \frac{\partial^2 f}{\partial x_{\mathrm{d}} \partial x_1} & & \cdots & \frac{\partial^2 f}{\partial x_{\mathrm{d}}^2} \end{bmatrix}$$

$\ell: \mathbb{R}^d \to \mathbb{R}$ یک تابع هزینه که نشان دهنده خطای پیش‌بینی رخ داده توسط وزن‌های داده شده، $\ell(\boldsymbol{w})$. اگر مشترک باشد، ℓ_i معمولاً هزینه را در نمونه i-ام با $\frac{1}{n}\sum_{i=1}^{n} \ell_i(\boldsymbol{w})$ برای n نمونه نشان می‌دهد،

[1] Hessian matrix

$c: \mathbb{R}^d \to \mathbb{R}$ یک تابع هدف عمومی که می‌خواهیم آن را برای متغیر آموخته شده $\boldsymbol{w}$ کمینه کنیم. این می‌تواند، برای مثال، یک هزینه به علاوه یک تنظیم کننده باشد.

متغیرهای تصادفی و احتمالات

X یک متغیر تصادفی تک متغیره با حروف بزرگ نوشته می‌شود.

$\mathcal{X}$ فضای مقادیر برای متغیر تصادفی.

x متغیر کوچک یک نمونه یا نتیجه است، $x \in \mathcal{X}$.

$\mathbf{X}$ یک متغیر تصادفی چند متغیره با حروف بزرگ پررنگ نوشته می‌شود.

$\boldsymbol{x}$ متغیر پررنگ با حروف کوچک یک نمونه چند متغیره است. در موارد خاص، زمانی که مقدار متغیر به عنوان یک بردار در نظر گرفته می‌شود، از $\mathbf{x}$ استفاده می‌کنیم.

$\mathcal{N}(\mu, \sigma^2)$ یک توزیع گاوسی تک متغیره، با پارامترهای μ, σ_2.

$\sim$ نشان می‌دهد که یک متغیر به عنوان یک توزیع بیان میشود. مثال، $X \sim \mathcal{N}(\mu, \sigma_2)$.

پارامترها و برآورد

$\mathcal{D}$ یک مجموعه داده، معمولاً از n عنصر ورودی‌های چند متغیره $\boldsymbol{X} \in \mathbb{R}^{n \times d}$ و خروجی‌های تک متغیره $y \in \mathbb{R}^n$ یا خروجی‌های چند متغیره $Y \in \mathbb{R}^{n \times m}$ تشکیل شده است. مجموعه داده همچنین به عنوان مجموعه‌ای از تاپل‌های نمایه شده نامیده می‌شود. به عنوان مثال، $\mathcal{D} = \{(\boldsymbol{x}_1, \mathrm{y}_1), (\boldsymbol{x}_2, \mathrm{y}_2), \ldots, (\boldsymbol{x}_\mathrm{n}, \mathrm{y}_\mathrm{n})\}$.

M یک مدل عمومی را برای بحث در مورد تخمین پارامتر کلی نشان می‌دهد. به عنوان مثال، $M = \theta$ برای برخی از پارامترهای θ، مانند میانگین توزیع گاوسی.

ω پارامترهای حقیقی برای مدل‌های رگرسیون خطی (تعمیم یافته) و طبقه بندی، معمولاً با $\omega \in \mathbb{R}^d$

$\boldsymbol{w}$ پارامترهای تقریبی برای مدل‌های رگرسیون خطی (تعمیم یافته) و طبقه بندی، معمولاً با $\boldsymbol{w} \in \mathbb{R}^d$. هنگامی که $\boldsymbol{w}$ را به عنوان راه حل ماکسیمم احتمال در برخی از داده‌ها مورد بحث قرار می‌دهیم، $\boldsymbol{w}_{ML}(\mathcal{D})$ را می‌نویسیم تا نشان دهیم که تغییرپذیری از $\mathcal{D}$ ناشی می‌شود.

$max_{a \in \mathcal{B}} f(a)$ ماکسیمم مقدار یک تابع f در بین مقادیر a در مجموعه $\mathcal{B}$.

$\mathrm{argmax}_{\mathrm{a} \in \mathcal{B}} \mathrm{f(a)}$ مورد a در مجموعه $\mathcal{B}$ که ماکسیمم مقدار $f(a)$ را تولید می‌کند.

نرم‌ها

$||x||$ یک نرم روی x.

$||x||_2$ نرم ℓ_2 در یک بردار، $||x||_2 = \sqrt{\sum_{i=1}^{d} x_i^2}$. این نرم فاصله اقلیدسی را از مبدأ دستگاه مختصات به x می‌دهد. یعنی طول بردار x است.

$||x||_2^2$ نرم مجذور[1] ℓ_2در یک بردار، $||x||_2^2 = \sum_{i=1}^{d} x_i^2$

$||x||_p$ نرم کلی[2] ℓ_pدر یک بردار، $||x||_p = \left(\sum_{i=1}^{d} \{|x|_i^p\}^{1/p}\right)$

$||X||_F$ نرم فروبنیوس[3] یک ماتریس n در d است. یعنی

$$||\mathbf{X}||_{\mathbf{F}} = \sqrt{\sum_{i=1}^{n}\sum_{j=1}^{d} X_{ij}^2} = \sqrt{\sum_{i=1}^{n} ||X_{i:}||_2^2} = \sqrt{\sum_{j=1}^{d} ||X_{:j}||_2^2}$$

فرمول‌ها و قوانین مفید

$$\log\left(\frac{x}{y}\right) = \log(x) - \log(y)$$

$$\log(x^y) = y\log(x)$$

$$\sum_{i=1}^{m} a_i \int_{\mathcal{X}} f_i(x)p(x)dx = \int_{\mathcal{X}} \sum_{i=1}^{m} a_i f_i(x)p(x)dx \qquad \triangleright \text{ Can bring sum into integral}$$

$$\frac{d}{dx}\int_{X} f(x)p(x)dx = \int_{X} \frac{d}{dx} f(x)p(x)dx \qquad \triangleright \text{ Can (almost always) bring derivative into integral}$$

[1] squared
[2] general
[3] Frobenius

پیشگفتار: شروع با یک مثال رگرسیون خطی

مفهوم یادگیری ماشین شامل طیف وسیعی از تکنیک‌ها برای عمل یادگیری از داده‌ها است. یک هدف اصلی (و هدف ما عمدتا در این کتاب) پیش بینی است. بسیاری از تکنیک‌ها یک تابع $(f: \mathbb{R}^d \to \mathbb{R})$ را یاد می‌گیرند که یک ورودی‌های آن ویژگی‌ها یا نشانه‌هایی در مورد یک آیتم را وارد میکند و خروجی یک پیش بینی در مورد آن آیتم است. برای مثال، در نظر بگیرید که میخواهید قیمت یک خانه را بر اساس اطلاعات مربوط به آن خانه حدس بزنید یا پیش بینی کنید. این ویژگی‌ها ممکن است عبارت باشد از قدمت خانه، مساحت آن و فاصله آن تا نزدیک ترین بازار. بدون هیچ نمونه قبلی از هزینه‌های خانه، یعنی بدون هیچ داده‌ای، ممکن است حدس زدن این قیمت دشوار باشد. با این حال، تصور کنید مجموعه‌ای از ویژگی‌های چند خانه و هزینه‌های فروش مربوطه را برای خانه‌هایی که امسال فروخته شده‌اند به شما داده می‌شود. فرض کنید $\mathbf{x} \in \mathbb{R}^d$ بردار ویژگی‌های یک خانه باشد و در این مورد (قدمت خانه، مساحت و فاصله خانه تا نزدیک ترین نانوایی) $\mathrm{x} = [x_1 x_2 x_3] =$ [age, size, distancetobakery] باشد و هدف ما پیدا کردن قیمت $(y = price)$ باشد. اگر ما از قبل ۱۰ نمونه از قیمت خانه‌ها را داشته باشیم، یک مجموعه داده: $(\mathbf{x}_1, \mathrm{y}_1), \ldots, (\mathbf{x}_{10}, \mathrm{y}_{10})$، که در آن $(\mathbf{x}_i, \mathrm{y}_i)$ جفت $\left(\text{ویژگی،قیمت}\right)$ برای خانه $\mathbf{i}$ در مجموعه نمونه‌های شما است. یک هدف طبیعی یافتن تابعی است که داده‌ها را به دقت بازآفرینی کند، به‌عنوان مثال با تلاش برای یافتن تابع f که منجر به تفاوت کوچکی بین پیش‌بینی، $\mathrm{f}(\mathbf{x}_i)$ و قیمت واقعی، y_i، برای هر خانه شود. ما می‌توانیم این را به عنوان یک مشکل بهینه‌سازی در نظر گرفته و فرمول‌بندی کنیم. تصور کنید فضایی از توابع ممکن $\mathcal{H}$ داریم که می‌توانیم تابع f را از بین آن‌ها انتخاب کنیم. برای یک مورد ساده، اجازه دهید تصور کنیم که تابع خطی است: $\mathrm{f(x)} = w_0 + x_1 w_1 + x_2 w_2 + x_3 w_3$ که برای هر $\mathbf{w} = [w_0, w_1, w_2, w_3] \in R^d$ که در آن w_0 همان نقطه قطع تابع خطی، یا همان عرض از مبدا است. می‌توانیم سعی کنیم تابعی از کلاس توابع خطی پیدا کنیم که مجموع مربع اختلاف‌ها را به مقدار مینیمم برساند یعنی:

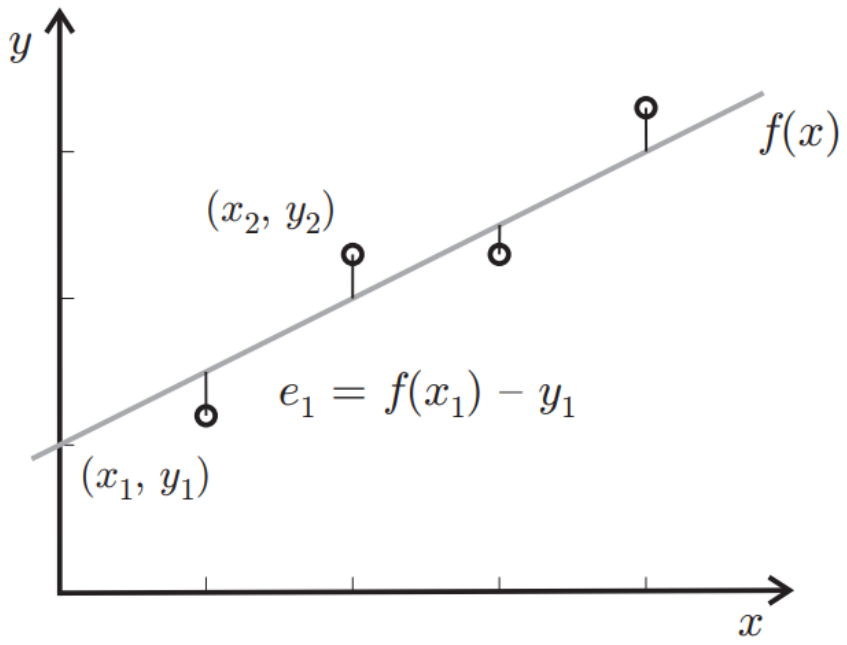
$$\min_{f \in H} \sum_{i=1}^{10} (f(\mathbf{x}_i) - y_i)^2$$

همانطور که بعداً در فصل ۵ خواهیم دید، حل این مسئله بهینه‌سازی برای توابع خطی ساده است. این روش شامل نوشتن صریح بهینه‌سازی بر حسب پارامترهای w و حل w بهینه است تا اندازه اختلاف‌ها را تا حد امکان کوچک می‌کند.

$$\mathrm{Err} \stackrel{\text{def}}{=} \sum_{i=1}^{10} (f(x_i) - y_i)^2 = \sum_{i=1}^{10} \left(w_0 + \sum_{j=1}^{d} w_j x_{ij} - y_i \right)^2$$

راه حل یک خط مستقیم است که سعی می‌کند به بهترین نحو با اهداف مشاهده شده y مطابقت داشته باشد. یک تصویر ساده از چنین تابعی، تنها برای یک ویژگی، در شکل ۱ نشان داده شده است. هنگامی که این تابع را داریم، وقتی خانه جدیدی را می‌بینیم، امیدواریم که به اندازه کافی شبیه خانه‌های قبلی باشد تا این تابع به اندازه کافی آن را پیش بینی کند. قیمت خانه

تابع آموخته شده f بین این 10 نقطه درون‌یابی می‌شود تا نقاط نادیده را پیش بینی کند. اما، یک سوال طبیعی این است که آیا ما به خوبی درون‌یابی کردیم و آیا f آموخته شده می‌تواند پیش‌بینی دقیقی در مورد خانه‌های جدید ایجاد کند؟ اگر می‌خواهید از این تابع آموخته شده f در عمل استفاده کنید، می‌خواهید چنین خصوصیاتی داشته باشید.



شکل ۱: نمونه ای از یک رگرسیون خطی مناسب در مجموعه داده $D = (1,1.2), (2,2.3), (3,2.3), (4,3.3)$ *که وظیفه فرایند بهینه سازی برای یافتن بهترین تابع خطی* $f(x) = w_0 + w_1 x$ *است به طوری که مجموع مجذور خطاها* $(e_1^2 + e_2^2 + e_3^2 + e_4^2)$ *به مقدار مینیمم میرسد*

در مورد درستی ادعای بالا، پاسخ به چنین سوالاتی بسیار دشوار است. ما می‌توانیم اصلاحات بصری انجام دهیم تا امیدوار باشیم پیش‌بینی‌های دقیق‌تری را ارائه دهیم، مانند گسترش کلاس توابع خطی به توابع غیر خطی پیچیده. اما، این تغییرات عملکردی هنوز به مشخص کردن صحت پیش‌بینی در خانه‌های جدید کمک نمی‌کند. در عوض، چیزی که از دست رفته مفهوم اعتماد به پیش‌بینی است. چقدر به پیش‌بینی‌ها اطمینان داریم؟ آیا خانه‌های قبلی را به اندازه کافی دیدیم که از این پیش‌بینی مطمئن باشیم؟ اعتبارسنجی منابع این داده‌ها چیست؟ همه این نوع سوالات نیاز به محاسبات احتمالی دارند. در این کتاب، ما با ارائه مقدمه‌ای بر احتمال شروع می‌کنیم تا مبنایی برای مقابله با عدم قطعیت در یادگیری ماشین فراهم کنیم. پس از آن که ابزارهای احتمالی را برای درک بهتر نحوه نزدیک شدن به پاسخ به این سؤالات در اختیار داشتیم، به یادگیری این توابع باز می‌گردیم. بسیاری از پیشینه ریاضی مورد نیاز برای این کتاب، شامل درک اولیه احتمال و بهینه سازی است. البته این کتاب تلاش خواهد کرد تا بیشتر پیشینه مورد نیاز را در سراسر آن ارائه کند.

فصل ۱

مقدمه‌ای بر مدل‌سازی احتمالی

مدل‌سازی جهان پیرامون و پیش‌بینی وقوع رویدادها، یک تلاش چند رشته‌ای است که بر پایه‌های محکم نظریه احتمال، آمار و علوم رایانه قرار دارد. اگرچه این زمینه‌ها در فرآیند مدل‌سازی در هم تنیده شده‌اند، اما نقش‌های نسبتاً قابل تشخیصی دارند و تا حدی می‌توان آنها را به صورت جداگانه مورد مطالعه قرار داد. تئوری احتمالات زیرساخت‌های ریاضی را برای دستکاری احتمالات به ارمغان می آورد و ما را با طیف وسیعی از مدل‌ها با ویژگی‌های نظری کاملاً درک شده مجهز می کند. آمار چارچوب‌هایی را برای فرمول‌بندی استنتاج و فرآیند محدود کردن فضای مدل بر اساس داده‌های مشاهده شده و تجربه ما به منظور یافتن و سپس تحلیل راه‌حل‌ها فراهم می‌کند. علوم کامپیوتر نظریه‌ها، الگوریتم‌ها و نرم‌افزارهایی را برای مدیریت داده‌ها، محاسبه راه حل‌ها و مطالعه رابطه بین راه‌حل‌ها و منابع موجود (زمان، مکان، معماری کامپیوتر و غیره) در اختیار ما قرار می‌دهد. به این ترتیب، این سه رشته چارچوب کمی اصلی را برای همه علوم تجربی و فراتر از آن تشکیل می‌دهند. تئوری احتمالات و آمار سابقه نسبتا طولانی دارند. ریشه شکل‌گیری هر دو را به طور می‌توان در قرن هفدهم دنبال کرد. نظریه احتمال از تلاش برای درک بازی‌های شانسی و قمار توسعه یافته است. مکاتبات بین بلز پاسکال و پیر دو فرما در سال ۱۶۵۴ به عنوان قدیمی ترین سابقه نظریه احتمالات مدرن است. از سوی دیگر، آمار از ابتکارات جمع‌آوری داده‌ها و تلاش‌ها برای درک روندها در جامعه (به عنوان مثال، تولید، علل مرگ و میر، قیمت زمین) و امور سیاسی (مانند درآمدهای عمومی، مالیات، ارتش) سرچشمه می‌گیرد. این دو رشته در قرن هجدهم با استفاده از داده‌ها برای اهداف استنتاجی در نجوم، جغرافیا و علوم اجتماعی شروع به ادغام کردند. افزایش پیچیدگی مدل‌ها و در دسترس بودن داده‌ها در قرن نوزدهم بر اهمیت ماشین‌های محاسباتی تاکید کرد. این به ایجاد پایه‌های حوزه علوم کامپیوتر در قرن بیستم کمک کرد، که عموماً به معرفی معماری فون نویمان و رسمی‌سازی مفهوم یک الگوریتم نسبت داده می‌شود. همگرایی این سه رشته در حال حاضر به جایگاه یک نظریه اصولی استنتاج احتمالی با کاربردهای گسترده در علم، تجارت، پزشکی، نظامی، مبارزات سیاسی و غیره رسیده است. مفاهیمی مانند توزیع بولتزمن، الگوریتم ژنتیک یا شبکه عصبی تأثیر فیزیک، زیست‌شناسی، روان‌شناسی و مهندسی را نشان می‌دهند. ما به فرآیند مدل‌سازی، استنتاج و تصمیم‌گیری بر اساس مدل‌های احتمالی به عنوان استدلال احتمالی یا استدلال تحت عدم قطعیت اشاره خواهیم کرد. نوعی استدلال در شرایط عدم قطعیت جزء ضروری زندگی روزمره است. به عنوان مثال، هنگام رانندگی، ما اغلب بر اساس انتظارات خود در مورد بهترین راه تصمیم می‌گیریم. در حالی که این موقعیت‌ها معمولاً مستلزم استفاده صریح از احتمالات و مدل‌های احتمالی نیستند، یک خودروی بدون راننده مانند شوفر گوگل[1] باید از آنها استفاده کند. و همینطور یک نرم افزار تشخیص هرزنامه در یک اکانت ایمیل، یک سیستم تشخیص تقلب در کارت اعتباری، یا الگوریتمی که استنباط می‌کند که آیا یک جهش ژنتیکی خاص منجر به بیماری می‌شود یا خیر. بنابراین، ابتدا باید مفهوم احتمال را درک

[1] Google Chauffeur

کنیم و سپس یک نظریه رسمی برای ترکیب شواهد (مثلاً داده‌های جمع‌آوری‌شده از ابزارها) به منظور اتخاذ تصمیم‌های خوب در طیف وسیعی از موقعیت‌ها معرفی کنیم.

در سطح پایه، از احتمالات برای تعیین کمیت احتمال وقوع رویدادها استفاده می‌شود. همانطور که ژاکوب برنولی در کار خود به نام هنر حدس زدن (۱۷۱۳) به طرز درخشانی بیان می‌کند: «حدس زدن [پیش‌بینی] درباره چیزی مانند اندازه‌گیری احتمال آن است. بنابراین، ما هنر حدس زدن[علم پیش‌بینی] اتفاقی را به عنوان هنر اندازه‌گیری احتمالات اشیاء با دقت هر چه تمام‌تر تعریف می‌کنیم تا در قضاوت‌ها و اعمال، همیشه آنچه را که پیدا شده است انتخاب کنیم یا دنبال کنیم. و نتایج بهتر، رضایت بخش تر، ایمن تر یا با دقت بیشتری مورد توجه قرار گیرد.» تکنیک‌های مدل سازی احتمالی بسیاری از مفاهیم شهودی را رسمیت می‌بخشد. به طور خلاصه، آنها ابزارهایی را برای تجزیه و تحلیل دقیق ریاضی و استنتاج، اغلب در حضور شواهد، در مورد رویدادهایی که تحت تأثیر عواملی هستند که ما یا به طور کامل درک نمی‌کنیم یا کنترلی بر آنها نداریم، ارائه می‌دهند. برای ارائه بینشی سریع به مفهوم عدم قطعیت و مدل سازی، پرتاب یک تاس شش وجهی مناسب را در نظر بگیرید. اگر موقعیت اولیه، نیرو، اصطکاک، تو رفتگی‌های شکل و سایر عوامل فیزیکی را با دقت در نظر بگیریم و محاسبه کنیم، سپس آزمایش را اجرا کنیم، می‌توانیم به‌طور دقیق (یا اینطور فکر می‌کنیم) نتیجه را پیش‌بینی کنیم. اما قوانین فیزیکی ممکن است شناخته شده نباشند، ممکن است ترکیب آنها دشوار باشد یا حتی ممکن است چنین اقداماتی توسط قوانین آزمایش مجاز نباشد. بنابراین، عملاً مفید است که به سادگی فرض کنیم که هر نتیجه به یک اندازه محتمل است. در واقع، اگر تاس را بارها پرتاب کنیم، در واقع مشاهده می‌کنیم که هر عدد تقریباً به یک اندازه مشاهده می‌شود. تخصیص یک شانس (احتمال) برابر برای هر نتیجه از چرخش تاس، روشی کارآمد و ظریف برای مدل‌سازی عدم قطعیت‌های ذاتی آزمایش فراهم می‌کند.

مثال واقعی‌تر دیگری که در آن جمع‌آوری داده‌ها مبنایی را برای مدل‌سازی احتمالی ساده فراهم می‌کند، وضعیت رانندگی هر روز به محل کار و پیش‌بینی مدت زمانی است که فردا به مقصد می‌رسیم. اگر «زمان برای کار» را برای چند ماه ثبت می‌کردیم، مشاهده می‌کردیم که سفرها معمولاً بسته به عوامل داخلی (مثلاً سرعت ترجیحی ما برای روز) و همچنین عوامل خارجی (مانند آب‌وهوا، شلوغی جاده‌ها، تصادفات، یک راننده کند.) زمان‌های متفاوتی طول می‌کشد». در حالی که این رویدادها، در صورت شناخته شدن، می‌توانند برای پیش‌بینی مدت زمان دقیق رفت و آمد مورد استفاده قرار گیرند، انتظار داشتن اطلاعات کامل نه تنها غیرواقعی است بلکه ما قابلیت مشاهده جزئی نداریم. ارائه راه‌هایی برای جمع‌آوری عوامل خارجی از طریق جمع آوری داده‌ها در یک دوره زمانی و ارائه توزیع زمان رفت و آمد مفید است. چنین توزیعی، در غیاب هر گونه اطلاعات دیگری، پس از آن استدلال در مورد رویدادهایی مانند رسیدن به موقع به یک جلسه مهم در ساعت ۹ صبح را تسهیل می‌کند.

تکنیک‌های مدل‌سازی احتمالی، یک چارچوب را برای برخورد با چنین آزمایش‌های تکراری تحت تأثیر تعدادی از عوامل خارجی که کنترل یا دانش کمی روی آن‌ها داریم، ارائه می‌کند. با چنین چارچوب‌هایی، می‌توانیم نحوه پیش‌بینی‌های خود را بهتر درک کرده و بهبود ببخشیم، زیرا می‌توانیم پیش‌فرض‌های خود را در مورد عدم قطعیت خود با وضوح بیشتری مشخص کنیم و به صراحت درباره نتایج احتمالی استدلال کنیم. در این فصل به معرفی احتمالات و نظریه احتمالات، از ابتدا می‌پردازیم. از آنجایی که احتمال یک مفهوم اساسی در یادگیری ماشینی است، ارزش آن را دارد که بفهمیم از کجا آمده است. با این وجود، با پیروی از محتوای این یادداشت‌ها، یک درمان مختصر خواهد بود و بیشتر بر آنچه برای درک مطالب در فصل‌های بعدی نیاز است تمرکز خواهد کرد.

۱-۱ نظریه احتمال و متغیرهای تصادفی

نظریه احتمال به عنوان شاخه ای از ریاضیات است که به اندازه‌گیری احتمال وقوع رویدادها می‌پردازد. در قلب نظریه احتمال، مفهوم آزمایش وجود دارد. یک آزمایش می‌تواند فرآیند پرتاب کردن یک سکه، چرخاندن تاس، بررسی دمای فردا یا تعیین محل کلیدها باشد. وقتی انجام می‌شود، هر آزمایش یک نتیجه دارد، که عنصری است که از مجموعه‌ای از گزینه‌های از پیش تعریف‌شده، به‌طور بالقوه بی‌نهایت اندازه گرفته شده است. نتیجه یک چرخش تاس عددی بین یک تا شش است. دمای فردا ممکن است یک عدد حقیقی باشد. نتیجه مکان کلیدها می‌تواند مجموعه‌ای مجزا از مکان‌ها مانند میز آشپزخانه، زیر کاناپه، دفتر کار و غیره باشد. از بسیاری جهات، هدف اصلی مدل‌سازی احتمالی، فرمول‌بندی یک سوال خاص یا یک فرضیه مربوط به دنیای فیزیکی به عنوان یک آزمایش این است که داده‌ها را جمع آوری کرده و سپس یک مدل بسازید. هنگامی که یک مدل ایجاد شد، می‌توانیم معیارهای کمی مجموعه‌ای از نتایج مورد علاقه خود را محاسبه کنیم و اعتمادی را که باید به این معیارها داشته باشیم، ارزیابی کنیم. ما می‌توانیم قواعد احتمال را بر اساس مجموعه‌ای ساده از اصول به نام اصول احتمال بسازیم. اجازه دهید فضای نمونه (Ω) مجموعه ای ناتهی از نتایج و فضای رویداد $(\mathcal{E})$ مجموعه ای ناتهی از زیر مجموعه‌های Ω باشد. به عنوان مثال، $\Omega = \{1,2,3\}$ و یک رویداد ممکن است $A = \{1,3\} \in \mathcal{E}$، که در آن فضای رویداد این است که 1 یا 3 مشاهده می‌شود. فضای رویداد $\mathcal{E}$ باید ویژگی‌های زیر را برآورده کند

۱. $A \in \mathcal{E} \Rightarrow A^c \in \mathcal{E}$ {به طوری که A^c مکمل مجموعه A است $A^c = \Omega - A$:A}

۲. $A_1, A_2, \ldots \in \mathcal{E} \Rightarrow \cup_{i=1}^{\infty} A_i \in \mathcal{E}$

۳. $\mathcal{E}$ یک مجموعه ناتهی است {در حقیقت از قسمت 1و 2 میدانیم که $\emptyset \in \mathcal{E}$ و $\Omega \in \mathcal{E}$}

اگر $\mathcal{E}$ این سه ویژگی را برآورده کند، آنگاه به $(\Omega, \mathcal{E})$ فضای قابل اندازه گیری گفته می‌شود. اکنون می‌توانیم اصول احتمال را تعریف کنیم، که روشن‌تر می‌کند که چرا این دو شرط برای فضای رویداد ما برای تعریف احتمالات معنادار بر روی رویدادها مورد نیاز است. تابع $P : \mathcal{E} \rightarrow [0,1]$ اصول احتمال را برآورده می‌کند اگر

۱. $P(\Omega) = 1$

۲. $A_1, A_2, \ldots \epsilon\, \mathcal{E}, A_i \cap A_j = \emptyset\ \forall i,j \Rightarrow P(\cup_{i=1}^{\infty} A_i) = \sum_{i=1}^{\infty} P(A_i)$

که به آن اندازه‌گیری احتمال یا توزیع احتمال گفته می‌شود. چندتایی (Ω, A, P) فضای احتمال نامیده می‌شود. زیبایی این اصول در فشردگی و ظرافت آنها نهفته است. بسیاری از عبارات مورد استفاده را می‌توان از بدیهیات احتمال استخراج کرد. برای مثال، واضح است که $P(A^c) = 1 - P(A)$. این امر روشن‌تر می‌کند که چرا ما نیاز داریم که اگر یک رویداد در فضای رویداد باشد، مکمل آن نیز باید در فضای رویداد باشد: اگر بتوانیم احتمال یک رویداد را اندازه‌گیری کنیم، می‌دانیم که احتمال رخ ندادن آن رویداد به طور مشابه 1 منهای این احتمال است. ما مستلزم این هستیم که اگر دو رویداد A_1, A_2 بدون اشتراک باشند، آنگاه: $P(A_1 \cup A_2) = P(A_1) + P(A_2)$ احتمال وقوع هر یک از این رویدادها برابر مجموع احتمالات آنها است، زیرا هیچ همپوشانی وجود ندارد. در نتیجه در رویدادها ویژگی دیگری که می‌توانیم استنباط کنیم این است که ما همیشه $\Omega, \emptyset \in \mathcal{E}$ داریم، جایی که $\emptyset$ مربوط به رویدادی است که در آن هیچ اتفاقی نمی‌افتد – که باید احتمال آن صفر باشد. اگر به قوانین دیگری که می‌توان استخراج کرد علاقه‌مند هستید، به پیوست A.1 مراجعه کنید.

مثال ۱: [متغیرهای گسسته (قابل شمارش)] احتمالات ریختن یک تاس را مدلسازی کنید.

فضای نتیجه مجموعه محدود $\Omega = \{1,2,3,4,5,6\}$ و فضای رویداد $\mathcal{E}$ مجموعه همه زیر مجموعه‌ها است، یعنی $\mathrm{P}(\Omega) = \{\emptyset, \{1\}, \{2\}, \ldots, \{2,3,4,5,6\}, \Omega\}$ که از همه زیر مجموعه‌های ممکن Ω تشکیل شده است. یک توزیع احتمال طبیعی روی $(\Omega, \mathcal{E})$به هر تاس $\frac{1}{6}$ شانس وقوع می‌دهد، که به صورت $\mathrm{P}(\{\mathrm{x}\}) = \frac{1}{6}$ برای $\mathrm{x} \in \Omega, \mathrm{P}(\{1,2\}) = \frac{1}{3}$ تعریف می‌شود. و غیره.

مثال ۲: [متغیرهای پیوسته (غیر قابل شمارش)] احتمال‌های زمان توقف خودرو را در محدوده ۳ تا ۶ ثانیه مدل کنید.

فضای نتیجه بازه پیوسته $\Omega = [3,6]$ است. یک رویداد می‌تواند این باشد که خودرو در عرض ۳ تا ۳,۱ ثانیه متوقف شود و به صورت $\mathrm{A} = [3,3.1] \in \mathcal{E}$ در نظر گرفته شود. احتمال $P(A)$ چنین رویدادی کم است، زیرا زمان توقف بسیار سریع خواهد بود. سپس می‌توانیم تمام فواصل زمانی ممکن برای فضای رویداد و احتمالات مربوطه را در نظر بگیریم. ما قبلاً دیدیم که این برای متغیرهای پیوسته کمی پیچیده‌تر خواهد بود، و بنابراین ما با دقت بیشتری نحوه تعریف $\mathcal{E}$ و P را در زیر در بخش ۱,۲,۲ نشان می‌دهیم.

این دو مثال دو مورد متداول را که با آن مواجه خواهیم شد نشان می‌دهد: متغیرهای گسسته و متغیرهای پیوسته. عبارات فوق - قابل شمارش و غیرقابل شمارش - نشان می‌دهد که آیا یک مجموعه قابل شمارش است یا خیر. به عنوان مثال، مجموعه اعداد طبیعی را می‌توان شمارش کرد، و بنابراین قابل شمارش است، در حالی که مجموعه اعداد حقیقی را نمی‌توان شمارش کرد - همیشه یک عدد حقیقی دیگر بین هر دو عدد حقیقی وجود دارد - و بنابراین غیرقابل شمارش است. اگرچه این تمایز منجر به تفاوت‌های واقعی می‌شود - مانند استفاده از مجموع برای مجموعه‌های قابل شمارش و انتگرال‌ها برای مجموعه‌های غیرقابل شمارش - قالب و شهود تا حد زیادی بین این دو دسته منتقل می‌شوند. ما بیشتر بر روی متغیرهای گسسته و پیوسته تمرکز خواهیم کرد. بسیاری از ایده‌های مشابه به متغیرهای مختلط نیز منتقل می‌شوند، جایی که فضاهای نتیجه از هر دو مجموعه گسسته و پیوسته مانند $\Omega = [0,1] \cup \{2\}$ تشکیل شده‌اند. علاوه بر این، برای تنظیم غیرقابل شمارش، ما به طور خاص مجموعه‌های پیوسته را مورد بحث قرار می‌دهیم، به عنوان مثال، آن‌ها اتحادیه‌های بازه‌های پیوسته مانند $\Omega = [0,1] \cup [5,10]$ هستند. از آنجا که تقریباً تمام مجموعه‌های غیرقابل شمارش که می‌خواهیم در نظر بگیریم پیوسته هستند، برای تعیین چنین فضاهایی از اصطلاحات پیوسته و غیرقابل شمارش استفاده می‌کنیم. در نهایت، مجموعه‌های گسسته می‌توانند متناهی باشند، مانند $\{1,2,3\}$، یا قابل شمارش بی نهایت، مانند اعداد طبیعی. مجموعه‌های پیوسته به وضوح نامتناهی هستند و گفته می‌شود که به طور غیرقابل شمارش نامتناهی هستند.

قبل از توضیح بیشتر در مورد چگونگی تعریف توزیع احتمال، ابتدا متغیرهای تصادفی را معرفی می‌کنیم و از اینجا به بعد به طور دقیق به متغیرهای تصادفی می‌پردازیم. یک متغیر تصادفی به ما اجازه می‌دهد تا تبدیل‌های فضاهای احتمال را با دقت بیشتری تعریف کنیم. هنگامی که آن تبدیل را اجرا می‌کنیم، می‌توانیم فضای احتمال زیربنایی را فراموش کنیم و می‌توانیم روی رویدادها و توزیع فقط بر روی متغیر تصادفی تمرکز کنیم. این در واقع همان کاری است که شما به طور طبیعی هنگام تعریف احتمالات بر روی متغیرها انجام می‌دهید، بدون اینکه نیازی به فرمول‌بندی کردن آن به صورت ریاضی باشد. البته در اینجا آن را به فرمول‌های ریاضی تبدیل می‌کنیم.

دوباره مثال تاس را در نظر بگیرید، جایی که اکنون در عوض ممکن است بخواهید بدانید: احتمال دیدن یک عدد کوچک در بازه $(1-3)$ یا یک عدد بزرگ در بازه $(4-6)$ چقدر است؟ می‌توانیم فضای احتمال جدیدی را با $\Omega_x = \{\text{high}, \text{low}\}$،

$\mathcal{E}_x = \{\text{high}, \text{low}\}$ و $P_x(\{\text{hifg}\}) = P_x(\{\text{low}\}) = \frac{1}{2}$ تعریف کنیم. تابع تبدیل $X = \Omega \to \Omega_x$ به این صورت تعریف میشود:

$$X(\omega) \overset{def}{=} \begin{cases} \text{low} & \text{if } \omega \in \{1,2,3\} \\ \text{high} & \text{if } \omega \in \{4,5,6\} \end{cases}$$

$$X(\omega) \overset{\text{def}}{=} \begin{cases} \text{low} & \text{if } \omega \in \{1,2,3\} \\ \text{high} & \text{if } \omega \in \{4,5,6\} \end{cases}$$

توزیع P_x بلافاصله از این تبدیل تعیین می‌شود. به عنوان مثال $P_x(\{\text{low}\}) = P(\{\omega : X(\omega) = \text{low}\})$ زیرا فضای احتمال low احتمال دیدن 1، 2 یا 3 را نشان می‌دهد. اکنون می‌توانیم به سؤالاتی در مورد احتمال پاسخ دهیم. دیدن یک عدد کوچک از یک عدد عضو مجموع high.

این تابع X یک متغیر تصادفی نامیده می‌شود. این کمی گیج کننده است که نه تصادفی است و نه یک متغیر، زیرا X یک تابع است. با این حال، از اینجا به بعد، برای تابع X به جای نوشتن $P(\{\omega : X(\omega)\})$ با نوشتن عباراتی مانند $P_x(X = x)$ یا $P_x(X \in A)$ مانند یک متغیر تصادفی رفتار می‌کنیم $P(\{\omega : X(\omega) = x\})$ یا $P(\{\omega : X(\omega) \in A\})$. برای صحت موضوع، می‌توانیم به یاد داشته باشیم که تابعی است که در فضای احتمالی زیربنایی پیچیده‌تر تعریف شده است. اما، در عمل، می‌توانیم مستقیماً بر حسب متغیر تصادفی X و احتمالات مرتبط با آن فکر کنیم. به طور مشابه، حتی برای نقش تاس، می‌توانیم تصدیق کنیم که فضای احتمالی پیچیده‌تری وجود دارد که توسط پویایی تاس تعریف می‌شود. هنگامی که فقط احتمالات نتایج گسسته از 1-6 را در نظر می‌گیریم، ما قبلاً به طور ضمنی تغییری را در بالای احتمالات سیستم فیزیکی اعمال کرده ایم. هنگامی که یک متغیر تصادفی داریم، یک فضای احتمال معتبر $(\Omega_x, \mathcal{E}_x, P_x)$ را تعریف می‌کند. بنابراین، همه قواعد احتمال یکسان اعمال می‌شوند، درک یکسانی از نحوه تعریف توزیع‌ها، و غیره. در واقع، ما همیشه می‌توانیم یک متغیر تصادفی X را که مطابق با هیچ تبدیلی نیست، تعریف کنیم تا فضای احتمال اصلی را بدست آوریم. به همین دلیل، با فرض اینکه همیشه با متغیرهای تصادفی سروکار داریم، بدون از دست دادن کلیت، می‌توانیم جلو برویم. ما اندیس‌ها را حذف می‌کنیم و $(\Omega, \mathcal{E}, P)$را برای X تعریف می‌کنیم. برای بحث عمیق‌تر در مورد متغیرهای تصادفی، به پیوست A.4 مراجعه کنید.

۲-۱ تعریف توزیع

اکنون می‌خواهیم بدانیم که چگونه P را برای اصول احتمال مشخص کنیم، تا احتمال X را برای یک رویداد A، $P(X \in A)$ با فضای نتیجه Ω مدل کنیم. این کار دلهره‌آور به نظر می‌رسد، زیرا به نظر می‌رسد ما باید احتمال هر رویداد ممکن را تعریف کنیم - مجموعه ای از نتایج - و به گونه‌ای که اصول احتمال را برآورده کند، نه! خوشبختانه، در عوض می‌توانیم توزیع را با استفاده از تابعی تعریف کنیم که مستقیماً روی نمونه‌های $x \in \Omega$ تعریف شده است. در نظر گرفتن جداگانه فضاهای نمونه گسسته (قابل شمارش) و پیوسته (غیر قابل شمارش) راحت است. برای حالت گسسته، توابع جرم احتمال و برای حالت پیوسته، توابع چگالی احتمال را تعریف می‌کنیم.

۱-۲-۱ توابع جرم احتمال

فرض کنید Ω یک فضای نمونه گسسته و $\mathcal{E} = \mathrm{P}(\Omega)$، مجموعه زیر مجموعه‌های Ω باشد. تابع $p : \Omega \to [0,1]$ تابع جرم احتمال[1] به اختصار (pmf) نامیده می‌شود اگر

$$\sum_{\omega\in\Omega} \mathrm{p}(\omega) = 1$$

احتمال هر رویداد $\mathrm{A} \in \mathcal{E}$ به صورت تعریف شده است

$$P(A) \stackrel{\text{def}}{=} \sum_{\omega\in\Omega} \mathrm{p}(\omega) = 1$$

راستی آزمایی اینکه P اصول احتمال را برآورده می‌کند و بنابراین یک توزیع احتمال است ساده است. بنابراین، برای متغیرهای تصادفی گسسته، ما اغلب $\mathrm{P}(\mathrm{X} = \mathrm{x})$ را می‌نویسیم، به این معنی که $\mathrm{P}(\mathrm{X} = \mathrm{x}) = \mathrm{p}(\mathrm{x})$ برای هر نتیجه $\mathrm{x} \in \Omega$. ما به ندرت توزیع را مستقیماً تعریف می‌کنیم، و به جای آن، $\mathrm{pmf}\ \mathrm{p}$ را که توزیع P را القا می‌کند، تعریف می‌کنیم.

مثال ۳: یک پرتاب از تاس شش وجهی منصفانه را در نظر بگیرید. یعنی $\Omega = \{1,2,3,4,5,6\}$ و فضای رویداد $\mathcal{E} = P(\Omega)$. احتمال اینکه نتیجه عددی بزرگتر از 4 باشد چقدر است؟

اول، چون تاس منصفانه است، می‌دانیم که $p(\omega) = \frac{1}{6}$ برای $\forall\omega \in \Omega$. حال، اجازه دهید A یک رویداد در $\mathcal{E}$ باشد که نتیجه آن بزرگتر از 4 باشد. یعنی $A = \{5,6\}$. بدین ترتیب،

$$P(A) \stackrel{\text{def}}{=} \sum_{\omega\in A} \mathrm{p}(\omega) = \frac{1}{3}$$

توجه داشته باشید که توزیع P بر روی عناصر $\mathcal{E}$ تعریف شده است، در حالی که p بر روی عناصر Ω تعریف شده است. یعنی $P(\{1\}) = p(1),\ P(\{2\}) = p(2),\ P(\{1،2\}) = p(1) + p(۲)$,. . . .

بنابراین، برای مشخص کردن P، باید نحوه تعیین pmf، یعنی احتمال هر نتیجه گسسته را تعیین کنیم. pmf اغلب به عنوان جدولی از مقادیر احتمال مشخص می‌شود. برای مثال، برای مدل‌سازی احتمال تولد برای هر روز در سال، می‌توان جدولی از ۳۶۵ مقدار بین صفر و یک داشت، تا زمانی که مجموع احتمالات برابر با 1 باشد. این احتمالات را می‌توان از داده‌های مربوط به تولد افراد محاسبه کرد با استفاده از شمارش برای هر روز و عادی سازی تعداد کل افراد در جمعیت برای تخمین احتمال دیدن تولد در یک روز معین. چنین جدول مقادیر بسیار منعطف است و امکان تعیین مقادیر احتمال دقیق برای هر نتیجه را فراهم می‌کند. با این حال، چند pmf مفید وجود دارد که شکل عملکردی (محدودتر) دارند. ما سه pmf از این قبیل را در اینجا شرح می‌دهیم که در سراسر این کتاب از آنها استفاده خواهیم کرد. برای مثال‌های بیشتر از pmf، به پیوست A.2 مراجعه کنید.

توزیع برنولی از مفهوم آزمایش برنولی ناشی می‌شود، آزمایشی که دو نتیجه ممکن دارد: موفقیت و شکست. در آزمایش برنولی، موفقیت با احتمال $\alpha \in [0،1]$ و بنابراین، شکست با احتمال $(1-\alpha)$ رخ می‌دهد. پرتاب یک سکه (سر/دم)، یک بازی

[1] Probability mass functions

بسکتبال (برد/باخت)، یا یک چرخش قالب (زوج/فرد) همگی به عنوان آزمایش برنولی دیده می‌شوند. ما این توزیع را با تنظیم فضای نمونه روی دو عنصر و تعریف احتمال یکی از آنها به عنوان α مدل می‌کنیم. به طور خاص، $\Omega = \{\text{success}، \text{failure}\}$ و داریم:

$$p(\omega) = \begin{cases} \alpha & \omega = \text{success} \\ 1 - \alpha & \omega = \text{failure} \end{cases}$$

که $\alpha \in (0 و 1)$ یک پارامتر است. اگر به جای آن $\Omega = \{0 و 1\}$ را در نظر بگیریم، می‌توانیم توزیع برنولی را به طور فشرده به صورت $p(\omega) = \alpha^{\omega}(1-\alpha)^{1-\omega}$ برای $\omega \in \Omega$ بنویسیم. توزیع برنولی اغلب به صورت $Bernoulli(\alpha)$ نوشته می‌شود. همانطور که خواهیم دید، یک نگاشت رایج که در آن از برنولی استفاده می‌کنیم برای طبقه بندی دوتایی است، مثلاً جایی که سعی می‌کنیم پیش‌بینی کنیم که آیا بیمار آنفولانزا دارد (نتیجه 0) یا آنفولانزا ندارد (نتیجه 1).

توزیع یکنواخت برای فضاهای نمونه گسسته بر روی مجموعه محدودی از نتایج تعریف می‌شود که احتمال وقوع هر کدام به یک اندازه است. اجازه دهید $\Omega = \{1، \ldots, n\}$؛ سپس برای $\forall \omega \in \Omega$ قرار دهیم

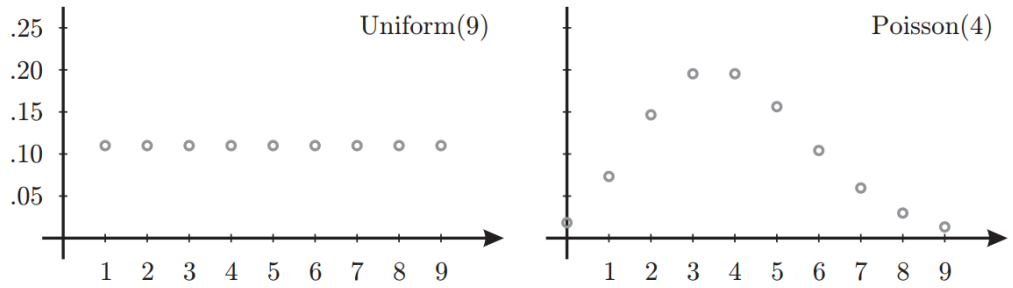
$$P(\omega) = \frac{1}{n}$$

توزیع یکنواخت شامل پارامترها نیست. با اندازه فضای نمونه تعریف می‌شود. ما به این توزیع به عنوان Uniform(n) اشاره می‌کنیم. بعداً خواهیم دید که توزیع یکنواخت را می‌توان در فواصل محدود در فضاهای پیوسته نیز تعریف کرد.

توزیع پواسون منعکس کننده احتمال وقوع چند حادثه است (به طور ضمنی در یک بازه زمانی ثابت). به عنوان مثال، یک مرکز تماس که به احتمال زیاد ۵۰ تماس در ساعت دریافت می‌کند، با احتمال بسیار کمتری ۵ تماس یا ۱۰۰۰ تماس دریافت می‌کند. این را می‌توان با $\text{Poisson}(\lambda)$ مدل کرد، که در آن λ تعداد تماس‌های مورد انتظار را نشان می‌دهد. به طور رسمی‌تر، $\Omega = \{0 و 1 و \ldots\}$ و $\forall \omega \in \Omega$

$$p(\omega) = \frac{\lambda^{\omega} e^{-\lambda}}{\omega!}$$

این تابع توده‌ای به شکل تپه است، جایی که بالای تپه بیشتر در مرکز λ قرار دارد و سمت چپ تپه کوتاه و پر شیب و یک دم سمت راست بلند و کم شیب وجود دارد. توزیع پواسون بر روی یک فضای نمونه بی‌نهایت تعریف شده است، اما همچنان قابل شمارش است. این در شکل ۱-۱ نشان داده شده است.



شکل ۱-۱: دو تابع جرم احتمالی، برای متغیرهای تصادفی گسسته. توزیع پواسون بیشتر روی محور x ادامه مییابد (برای متغیر $\omega \in N$)، با احتمال کاهش به صفر به صورت $\omega \to \infty$

تمرین ۱: ثابت کنید که $\sum_{\omega \in N} p(\omega) = 1$ برای توزیع پواسون.

مثال ۴: به عنوان مقدمه‌ای برای تخمین پارامترهای توزیع‌ها، مثالی از نحوه استفاده از توزیع برنولی و تعیین پارامتر α برای برنولی را در نظر بگیرید. یک مثال متعارف برای توزیع‌های برنولی، پرتاب سکه است که در آن نتایج سر (H) یا دم (T) هستند. $P(X = H) = \alpha$ احتمال دیدن H و $P(X = T) = 1 - \alpha$ احتمال دیدن T است. ما معمولا $\alpha = 0.5$ را فرض می‌کنیم. این سکه منصفانه (بی طرفانه) نامیده می‌شود. اگر سکه را بارها پرتاب کنیم، انتظار داریم تقریباً تعداد برابر H و T را ببینیم. با این حال، یک سکه ناهمگن ممکن است مقداری انحراف به سمت H یا T داشته باشد. اگر سکه را بارها پرتاب کنیم، اگر $\alpha > 0.5$ باشد، باید در نهایت ببینیم که H بیشتری مشاهده میشود و اگر $\alpha < 0.5$ باشد، باید T بیشتری را ببینیم

چگونه ممکن است واقعاً مقدار α را تعیین کنیم؟ یک ایده شهودی این است که از آزمایش‌های مکرر (داده‌ها) استفاده کنید، درست همانطور که در بالا توضیح داده شد: سکه را بارها پرتاب کنید تا ببینید آیا می‌توانید انحراف را اندازه بگیرید. اگر تعداد 1000 تا H و 50 تا T را ببینید، یک حدس طبیعی برای انحراف $\alpha = \frac{1000}{1000+50} \approx 0.95$ است. چقدر به این راه حل اطمینان دارید؟ آیا قطعا 0.95 است؟ و چگونه می‌توانیم به طور رسمی‌تر تعریف کنیم که چرا این باید راه حل باشد؟ این در واقع یک راه حل معقول است و مطابق با راه‌حل ماکسیمم احتمال است، همانطور که در فصل ۳ بحث خواهیم کرد.

۲-۲-۱ توابع چگالی احتمال

محاسبات فضاهای احتمال پیوسته مشابه فضاهای گسسته است، با توابع چگالی احتمال[1] یا به اختصار pdf که جایگزین توابع جرم احتمال و انتگرال‌ها جایگزین مجموع می‌شوند. با این حال، در تعریف pdfها، ما نمی‌توانیم از جداول مقادیر استفاده کنیم و به فرم‌های تابعی محدود می‌شویم. دلیل اصلی این تفاوت از این واقعیت ناشی می‌شود که دیگر منطقی نیست که احتمال یک رویداد تکی را اندازه گیری کنیم. دوباره زمان توقف خودرو را در نظر بگیرید، که در مثال ۲ مورد بحث قرار گرفت. اینکه احتمال توقف خودرو را دقیقاً در 3.14159625 ثانیه بپرسید، چندان منطقی نیست. به طور واقع بینانه، احتمال چنین رویداد دقیقی بسیار کم است. در واقع، احتمال مشاهده دقیق زمان توقف صفر است، زیرا مجموعه $\{3.14159625\}$ به عنوان زیر مجموعه $[3, 6]$ مجموعه‌ای با اندازه صفر است. اساساً، جرم صفر را در داخل بازه $[3, 6]$ می‌گیرد که در نهایت به طور غیرقابل شمارش بی‌نهایت است. در عوض، ما باید احتمالات فواصل، مانند $[4, 5]$ یا $[5.667, 5.668]$ را در نظر بگیریم.

برای فضاهای پیوسته، فرض می‌کنیم که مجموعه رویدادهای ε شامل تمام فواصل ممکن است که به آن میدان بورل $B(\Omega)$ می‌گویند. به عنوان مثال، اگر $\Omega = R$، میدان بورل $B(\mathbb{R})$ شامل تمام بازه‌های باز (به عنوان مثال $(0, 1)$)، بازه‌های بسته (مثلاً $[0, 1]$) و بازه‌های نیمه باز (مثلا $[0, 1)$) در $\mathbb{R}$، و همچنین مجموعه‌هایی که می‌توان با تعداد قابل شمارشی از عملیات‌های مجموعه پایه روی آنها، مانند اجتماع، به دست آورد. این منجر به مجموعه‌ای محدودتر از رویدادها نسبت به مجموعه توان Ω می‌شود، که برای مثال شامل مجموعه‌هایی با یک رویداد تکی می‌شود. $B(\mathbb{R})$ هنوز یک مجموعه عظیم است - یک مجموعه نامتناهی غیرقابل شمارش - اما کوچکتر از $P(\mathbb{R})$ است. با این حال، به خوبی، $B(\mathbb{R})$ هنوز هم شامل تمام مجموعه‌هایی است که می‌خواهیم بتوانیم اندازه‌گیری کنیم است. میدان BoIRel را می‌توان برای هر فضای قابل اندازه گیری، مانند فضاهای

[1] Probability density functions

با ابعاد بالاتر مانند $\Omega = \mathbb{R}^2$، با رویدادهایی مانند $A = [0، 1] \times [0، 1] \subset \Omega$ یا $A = [1, 2] \times [-1, 4] \cup [0, 0.1] \times [10, 1000]$ تعریف کرد.

اکنون Ω یک فضای نمونه پیوسته و $\mathcal{E} = B(\Omega)$ باشد. تابع $p : \Omega \to [0, \infty)$ تابع چگالی احتمال (pdf) نامیده می‌شود اگر

$$\int_{\Omega} p(\omega) d\omega = 1$$

احتمال یک رویداد $A \in B(\Omega)$ به صورت تعریف می‌شود

$$P(A) \stackrel{\text{def}}{=} \int_{A} p(\omega) d\omega$$

توجه داشته باشید که تعریف pdf تنها به داشتن محدوده $[0, 1]$ محدود نمی‌شود، بلکه به $(0, \infty]$ محدود می‌شود. برای pmfها، احتمال یک رویداد تکی $\{\omega\}$ مقدار pmf در نقطه نمونه ω است. یعنی $P(\{\omega\}) = p(\omega)$. از آنجایی که توزیع‌های احتمال P به محدوده $[0, 1]$ محدود می‌شوند، این بدان معناست که pmfها نیز باید به آن محدوده محدود شود. در مقابل، مقدار pdf در نقطه ω یک احتمال نیست. در واقع می‌تواند بزرگ‌تر از 1 باشد. اگر چه زمانی که به طور غیررسمی صحبت می‌شود معمولاً $p(x)$ را احتمال x می‌نامیم، به‌طور دقیق‌تر آن را چگالی در x می‌نامیم، زیرا قطعاً یک احتمال نیست. در واقع، همانطور که در بالا گفته شد، احتمال در هر نقطه منفرد 0 است (یعنی یک زیر مجموعه قابل شمارش از Ω مجموعه ای از اندازه‌گیری صفر است).

یک سردرگمی طبیعی این است که چگونه p می‌تواند با 1 ادغام شود، اما در واقع مقادیری بزرگتر از 1 داشته باشد. بازه کوچک $A = [x, x + \Delta x]$ را در نظر بگیرید با احتمال

$$P(A) = \int_{x}^{x+\Delta x} p(\omega) d\omega \approx p(x)\Delta x$$

مقدار بالقوه بزرگ تابع چگالی با بازه کوچک Δx جبران می‌شود تا عددی بین 0 و 1 به دست آید. بنابراین، حتی اگر $p(x)$ یک میلیون باشد، چگالی نقاط در یک بازه کوچک می‌باشد. احتمال یک رویداد باید هنوز ≤ 1 باشد. چگالی نشان می‌دهد که احتمال بالایی در اطراف آن نقطه وجود دارد. با داشتن یک چگالی بزرگ در اطراف x، این نشان می‌دهد که چگالی برای نقاط دیگر صفر یا نزدیک به صفر است و pdf به شدت حول x به اوج خود رسیده است.

برخلاف pmf، ما نمی‌توانیم pdf p را به این راحتی تعریف کنیم تا به‌طور انعطاف‌پذیر احتمالات خاصی را برای هر نتیجه با جدولی از احتمالات ارائه کنیم. بلکه برای pdf معمولا از pdf شناخته شده‌ای استفاده می‌کنیم که ویژگی‌های مورد نیاز را برآورده کند. علاوه بر این، بر خلاف حالت گسسته، ما هرگز $P(X = x)$ را نخواهیم نوشت، زیرا این عدد صفر خواهد بود. در عوض، ما معمولاً $P(X \in A)$ یا سوالات احتمالی صریح تر مانند $P(X \leq 5)$ را می‌نویسیم. ما در اینجا چهار pdf را نشان می‌دهیم که در سراسر این کتاب استفاده می‌شود. برای نمونه‌های بیشتر از PDFها ، به پیوست A.3 مراجعه کنید.

توزیع یکنواخت با مقدار مساوی یک تابع چگالی احتمال در یک بازه محدود در $\mathbb{R}$ تعریف می‌شود. بنابراین، برای $\Omega = [a, b]$ تابع چگالی احتمال یکنواخت $\forall \omega \in [a, b]$ به این صورت تعریف می‌شود.

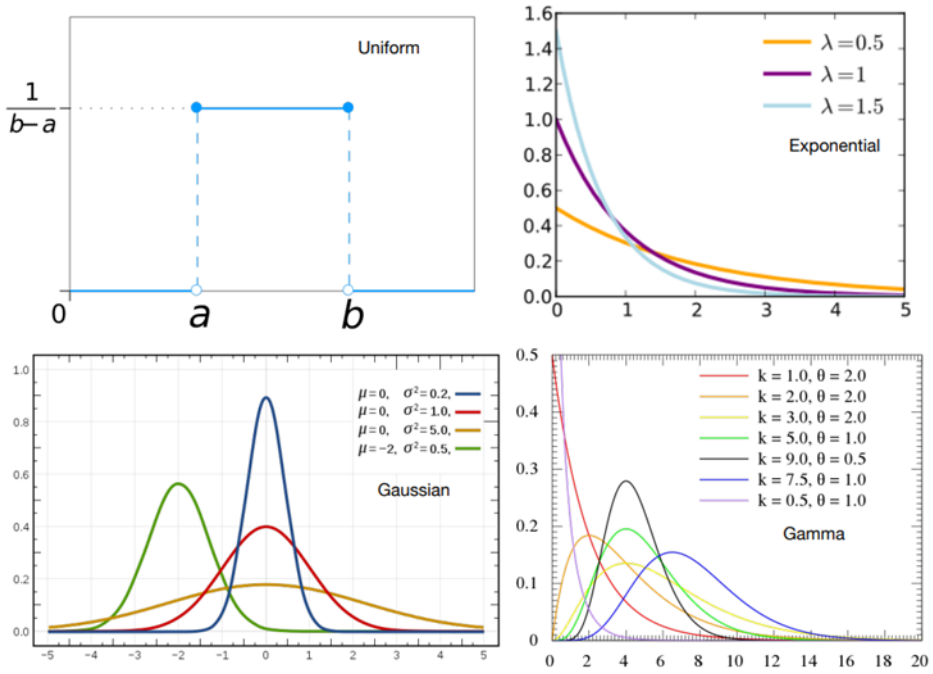
$$p(\omega) \stackrel{\text{def}}{=} \frac{1}{b - a}$$

همچنین می‌توان با درنظر گرفتن $\Omega = \mathbb{R}$ و تنظیم $p(\omega) = 0$ هر زمان که ω خارج از $[a, b]$ باشد، $\text{Uniform}(a, b)$ را تعریف کرد. این فرم مناسب است زیرا $\Omega = \mathbb{R}$ می‌تواند به طور مداوم برای همه توزیع‌های احتمال یک بعدی استفاده شود. وقتی این کار را انجام دادیم، به زیرمجموعه $\mathbb{R}$ اشاره خواهیم کرد که در آن $p(\omega) > 0$ به عنوان پشتیبان تابع چگالی است.

توزیع نمایی بر روی مجموعه‌ای از اعداد غیر منفی تعریف می‌شود. یعنی $\Omega = [0, \infty)$. برای پارامتر $\lambda > 0$، pdf آن به صورت:

$$p(\omega) = \lambda e^{-\lambda\omega}$$

همانطور که از نام آن پیداست، این pdf شکل نمایی دارد و با افزایش بزرگی مقادیر x، احتمال به شدت کاهش می‌یابد. مانند قبل، فضای نمونه را می‌توان به همه اعداد حقیقی گسترش داد، در این صورت برای $\omega < 0$، $p(\omega) = 0$ را قرار می‌دهیم.



شکل ۲-۱: چهار تابع چگالی احتمال، برای متغیرهای تصادفی پیوسته. تصاویر برگرفته از ویکی پدیا

توزیع گاوسی یا **توزیع نرمال** یکی از پرکاربردترین توزیع‌های احتمال است. بر روی $\Omega = \mathbb{R}$، با دو پارامتر $\mu \in \mathbb{R}$ و $\sigma > 0$ و pdf تعریف شده است.

توزیع نرمال دارای خواص زیر است:

- میانگین = میانه = مد
- خط تقارن در وسط قرار می‌گیرید
- نیمی از داده‌ها کوچکتر از میانگین و نیمی دیگر بزرگتر از میانگین

$$p(\omega) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{1}{2\sigma^2}(\omega-\mu)^2}$$

همانطور که در ادامه بحث خواهیم کرد، برای یک متغیر تصادفی که گاوسی توزیع شده است، پارامتر μ میانگین یا مقدار مورد انتظار و σ^2 واریانس است. ما به این توزیع به عنوان $\text{Gaussian}(\mu, \sigma^2)$ یا $N(\mu, \sigma^2)$ اشاره خواهیم کرد. وقتی میانگین صفر و واریانس یک باشد (واریانس واحد)، این گاوسی نرمال استاندارد نامیده می‌شود. نام این تابع گاوسی خاص به این دلیل نام دارد که بسیار مورد استفاده قرار می‌گیرد. هر دو توزیع گاوسی و نمایی اعضای خانواده وسیع‌تری از توزیع‌ها به نام خانواده نمایی طبیعی هستند. تعریف کلی این خانواده را بعداً در بخش ۷,۲ خواهیم دید.

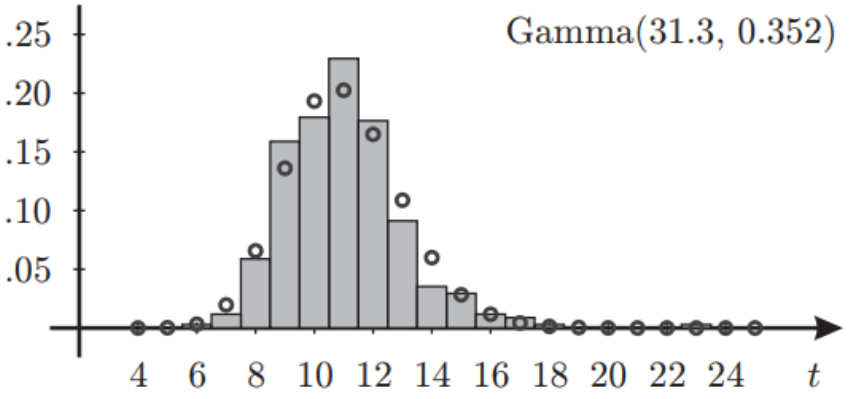
توزیع لاپلاس شبیه به توزیع گاوسی است، اما در حدوده میانگین اوج بیشتری دارد. همچنین بر روی $\Omega = \mathbb{R}$، با دو پارامتر، $\mu \in \mathbb{R}$ و $b > 0$ و pdf تعریف شده است.

$$p(\omega) = \frac{1}{2b} e^{-\frac{1}{b}|\omega-\mu|}$$

توزیع گاما برای مدل‌سازی زمان‌های انتظار استفاده می‌شود و مشابه توزیع پواسون است اما برای متغیرهای پیوسته. بر روی $\Omega = (0, \infty)$، با پارامتر شکل $\alpha > 0$ و پارامتر نرخ $\beta > 0$ و pdf تعریف شده است.

$$p(\omega) = \frac{\beta^{\alpha}}{\Gamma(\alpha)} \omega^{\alpha-1} e^{-\beta\omega}$$

شکل ۳-۱: انتخاب یک عدد تصادفی (x) از بازه واحد $[0, 1]$



شکل ۴-۱: هیستوگرام ضبط شده از زمان رفت و آمد (بر حسب دقیقه) تا محل کار. مجموعه داده شامل ۳۴۰ اندازه گیری است که در طی یک سال، برای مسافتی تقریباً ۳/۱ مایلی جمع آوری شده است. داده‌ها با استفاده از یک خانواده گاما از توزیع‌های احتمال، با پارامترهای مکان و

جایی که $\Gamma(\alpha)$ تابع گاما نامیده می‌شود. یک متغیر تصادفی که گاما توزیع شده است به صورت $X \sim \text{Gamma}(\alpha, \beta)$ نشان داده می‌شود.

مثال ۵: انتخاب یک عدد (x) بین 0 و 1 به طور یکنواخت و به طور تصادفی را در نظر بگیرید (شکل ۱,۳). احتمال اینکه عدد بزرگتر یا مساوی $\frac{3}{4}$ باشد یا کمتر مساوی $\frac{1}{4}$ چقدر است؟

ما می‌دانیم که $\Omega = [0,1]$. توزیع با pdf یکنواخت تعریف می‌شود، $p(\omega) = \frac{1}{b-a} = 1$ که $a = 0$، $b = 1$ بازه فضای نتیجه را تعریف می‌کند. ما رویداد مورد علاقه را به صورت $A = [0,\frac{1}{4}] \cup [\frac{3}{4},1]$ تعریف می‌کنیم و احتمال آن را به این صورت محاسبه می‌کنیم

$$P(A) = \int_0^{\frac{1}{4}} p(\omega)d\omega + \int_{\frac{1}{4}}^{1} p(\omega)d\omega = (\frac{1}{4} - 0) + (1 - \frac{3}{4}) = \frac{1}{2}$$

اگر در عوض احتمال اینکه عدد بزرگتر از $\frac{3}{4}$ یا کمتر از $\frac{1}{4}$ باشد را بپرسیم چه؟ از آنجایی که احتمال هر رویداد فردی در حالت پیوسته 0 است، اگر بازه‌های باز یا بسته را در نظر بگیریم، تفاوتی در ادغام وجود ندارد. بنابراین، احتمال همچنان $\frac{1}{2}$ خواهد بود

مثال ۶: بیایید تصور کنیم زمان رفت و آمد خود را برای سال جمع‌آوری کرده‌اید و می‌خواهید احتمال زمان رفت و آمد خود را مدل کنید تا به شما کمک کند بتوانید زمان رفت و آمد فردا را پیش بینی کنید. برای این تنظیم، متغیر تصادفی X شما مطابق با زمان رفت و آمد است و شما باید احتمالاتی را برای این متغیر تصادفی تعریف کنید. این داده‌ها را می‌توان گسسته در نظر گرفت و مقادیر را در دقیقه، $\{4,5,6,\ldots,26\}$ میگیرد. سپس می‌توانید هیستوگرام‌هایی از این داده‌ها ایجاد کنید (جدول مقادیر احتمال)، همانطور که در شکل ۱,۴ نشان داده شده است، تا احتمال زمان رفت و آمد را نشان دهید.

با این حال، زمان رفت و آمد در واقع مجزا نیست، بنابراین شما می‌خواهید به عنوان یک مدل پیوسته مدل‌سازی کنید. یک انتخاب معقول توزیع گاما است. با این حال، چگونه می‌توان داده‌های ثبت شده را گرفته و پارامترهای α, β را در توزیع گاما تعیین کرد؟ تخمین این پارامترها در واقع کاملاً ساده است، اگرچه به اندازه تخمین جداول مقادیر احتمال آشکار نیست. نحوه

انجام این کار را در فصل ۳ مورد بحث قرار می‌دهیم. توزیع گامای آموخته شده نیز در شکل ۱,۴ نشان داده شده است. با توجه به توزیع گاما، اکنون می‌توان این سوال را مطرح کرد: محتمل ترین زمان رفت و آمد امروز چقدر است؟ این مربوط به $\max_{\omega} p(\omega)$ است که به آن حالت توزیع می‌گویند. سوال طبیعی دیگر میانگین یا زمان مورد انتظار رفت و آمد است. برای به دست آوردن این، به مقدار مورد انتظار (میانگین) این توزیع گاما نیاز دارید که در زیر در بخش ۱,۴ تعریف می‌کنیم.

۱-۳ متغیرهای تصادفی چند متغیره

توسعه بسیاری از مفاهیم فوق به متغیرهای تصادفی چند متغیره - بردار متغیرهای تصادفی - گسترش می‌یابد، زیرا تعریف فضاهای نتیجه و احتمالات عمومی است. با این حال، مثال‌هایی که تاکنون ارائه شده‌اند، با متغیرهای تصادفی اسکالر سروکار داشته‌اند، زیرا برای متغیرهای تصادفی چند متغیره، باید نحوه تعامل متغیرها را درک کنیم. در این بخش، به چندین مفهوم جدید می‌پردازیم که تنها زمانی به وجود می‌آیند که متغیرهای تصادفی متعددی از جمله توزیع‌های مشترک، توزیع‌های شرطی، حاشیه‌ها و وابستگی بین متغیرها وجود داشته باشد.

اجازه دهید با یک مثال ساده‌تر شروع کنیم، با دو متغیر تصادفی گسسته X و Y با فضاهای نتیجه x و y. یک تابع جرم احتمال مشترک p وجود دارد: $X \times Y \rightarrow [0,1]$ و توزیع احتمال مشترک P، مانند که

$$p(x,y) \stackrel{\text{def}}{=} P(X = x, Y = y)$$

جایی که pmf باید انجام شود

$$\sum_{x \in X} \sum_{y \in Y} p(x,y) = 1$$

به عنوان مثال، اگر $X = \{young, old\}$ و $Y = \{no\ arthritis, arthritis\}$ آنگاه pmf می‌تواند جدول احتمالات مشترک باشد که در تعریف فضاهای احتمال مطابقت دارد، زیرا

		Y	
		0	1
X	0	$P(X=0, Y=0) = 1/2$	$P(X=0, Y=1) = 1/100$
	1	$P(X=1, Y=0) = 1/10$	$P(X=1, Y=1) = 39/100$

جدول ۱-۱: جدول احتمال مشترک برای متغیرهای تصادفی X و Y.

$\Omega = X \times Y$ یک فضای معتبر است و $\sum_{\omega \in \Omega} p(\omega) = \sum_{(x,y) \in \Omega} p(x,y) = \sum_{x \in X} \sum_{y \in Y} p(x,y)$. متغیر تصادفی $Z = (X, Y)$ یک متغیر تصادفی چند متغیره است که دارای دو بعد است.

با نگاه کردن به احتمالات مشترک در جدول، می‌توانیم ببینیم که دو متغیر تصادفی با هم تعامل دارند. به عنوان مثال، احتمال مفاصل برای افراد جوان و مبتلا به آرتریت کم است. علاوه بر این، به نظر می‌رسد بزرگی بیشتری در ردیف‌های مربوط به جوان

وجود دارد، که نشان می‌دهد احتمالات تحت تأثیر نسبت افراد پیر یا جوان در جمعیت است. در واقع، می‌توان پرسید که آیا می‌توانیم این نسبت را فقط از این جدول دریابیم.

پاسخ کاملاً مثبت است، و ما را به توزیع‌های حاشیه‌ای و اینکه چرا ممکن است به توزیع‌های حاشیه‌ای اهمیت دهیم، هدایت می‌کند. با توجه به توزیع مشترک بر روی متغیرهای تصادفی، می‌توان امیدوار بود که بتوانیم احتمالات خاص‌تری را استخراج کنیم، مانند توزیع فقط روی یکی از آن متغیرها، که توزیع حاشیه‌ای نام دارد. حاشیه را می‌توان به سادگی با جمع کردن تمام مقادیر متغیر دیگر محاسبه کرد

$$\mathrm{P(X = young)} = \mathrm{p(young, no\ arthritis)} + \mathrm{p(young, arthritis)} = \frac{51}{100}$$

یک فرد جوان یا بیماری آرتریت دارد یا ندارد، بنابراین جمع‌بندی این دو مورد احتمالی آن متغیر را مشخص می‌کند. بنابراین، با استفاده از داده‌های جمع‌آوری‌شده برای متغیر تصادفی $\mathrm{Z = (X, Y)}$، می‌توان نسبت جمعیت جوان و نسبت پیر را تعیین کرد.

به طور کلی، می‌توانیم متغیر تصادفی d بعدی $\mathbf{X} = (\mathrm{X}_1, \mathrm{X}_2, \ldots, \mathrm{X}_d)$ را با نتایج بردار $\boldsymbol{x} = (x_1, x_2, \ldots\ldots, x_d)$ در نظر بگیریم، به طوری که هر x_i از بین برخی از اشیا انتخاب شود سپس، برای حالت گسسته، هر تابع $p : X_1 \times X_2 \times \ldots \times X_d \rightarrow [0, 1]$ تابع جرم احتمال چند بعدی نامیده می‌شود اگر

$$\sum_{\mathrm{x}_1 \in \mathrm{X}_1} \sum_{\mathrm{x}_2 \in \mathrm{X}_2} \cdots \sum_{\mathrm{x}_d \in \mathrm{X}_d} \mathrm{p}\,(\mathrm{x}_1, \mathrm{x}_2, \ldots, \mathrm{x}_d) = 1$$

یا، برای حالت پیوسته، $p: X_1 \times X_2 \times .. \times X_d \rightarrow [1, \infty]$ یک تابع چگالی احتمال چند بعدی است اگر

$$\int_{\mathrm{x}_1} \int_{x_2} \ldots \int_{x_d} p(x_1, x_2, \ldots, x_d)\, dx_1 dx_2 \ldots dx_d = 1$$

یک توزیع حاشیه‌ای برای زیرمجموعه‌ای از $\mathrm{X} = (\mathrm{X}_1, \mathrm{X}_2, \ldots\ldots, \mathrm{X}_d)$ با جمع یا ادغام بر روی متغیرهای باقی مانده تعریف می‌شود. برای حالت گسسته، توزیع حاشیه‌ای $p(x_i)$ به این صورت تعریف شده است

$$\mathrm{p}\,(\mathrm{x}_i) \triangleq \sum_{\mathrm{x}_1 \in \mathrm{X}_1} \cdots \sum_{\mathrm{x}_{i-1} \in \mathrm{X}_{i-1}} \sum_{\mathrm{x}_{i+1} \in \mathrm{X}_{i+1}} \cdots \sum_{\mathrm{x}_d \in \mathrm{X}_d} p(x_1, \ldots, x_{i-1}, x_i, x_{i+1}, \ldots, x_d)$$

که در آن متغیر x_i روی مقداری ثابت است و همه مقادیر ممکن متغیرهای دیگر را جمع می‌کنیم. به طور مشابه، برای حالت پیوسته، توزیع حاشیه‌ای $p\,(x_i)$ به این صورت تعریف شده است

$$\mathrm{p}\,(\mathrm{x}_i) \triangleq \int_{\mathrm{x}_1} \ldots \int_{\mathrm{x}_{i-1}} \int_{\mathrm{x}_{i+1}} \ldots \int_{\mathrm{x}_d} p(x_1, \ldots, x_{i-1}, x_i, x_{i+1}, \ldots, x_d) \mathrm{dx}_1 \ldots \mathrm{dx}_{i-1} \mathrm{dx}_{i+1} \ldots \mathrm{dx}_d$$

توجه داشته باشید که ما از p برای تعریف چگالی روی $\mathbf{x}$ استفاده می‌کنیم، اما سپس این اصطلاح را بارگذاری می‌کنیم و همچنین از p برای چگالی فقط روی x_i استفاده می‌کنیم. برای نتایج دقیق‌تر، باید دو تابع مجزا (pdf) تعریف کنیم، مثلا p_x برای چگالی روی متغیر تصادفی چند متغیره و p_{x_i} برای حاشیه. با این حال، استفاده ساده از p و استنتاج متغیر تصادفی از متن معمول است. در بیشتر موارد، واضح است; اگر اینطور نیست، ما به صراحت pdfها را با زیرنویس‌های اضافی برجسته می‌کنیم.

ما می‌توانیم pmf و pdfهای چند متغیره رایج را تعریف کنیم که پسوند pmf و pdf اسکالر هستند. برخی از پسوندها - مانند جداول مقادیر احتمال و توزیع‌های یکنواخت واضح تر هستند، در حالی که برخی دیگر نیاز به اندیس‌های بیشتر دارند، مانند گاوسی. برای دیگران، مانند لاپلاس، پسوند ممکن است منحصر به فرد نباشد و چندین گزینه امکان پذیر است. ما در بخش پایانی این فصل، بخش ۱,۵، برای مرجع، افزونه‌هایی را که به آن نیاز خواهیم داشت، تعریف می‌کنیم. با این حال، ابتدا درک چگونگی تعامل چندین متغیر مفید خواهد بود، زیرا این امر بر گسترش از تک متغیره به چند متغیره تأثیر می‌گذارد. به ویژه، درک توزیع‌های شرطی و وابستگی مفید خواهد بود، که در ادامه به آن می‌پردازیم.

۱-۳-۱ توزیع‌های مشروط

احتمالات شرطی احتمالات یک متغیر تصادفی مانند X، اطلاعاتی در مورد مقدار متغیر تصادفی دیگر Y به ما میدهد. به طور رسمی‌تر، احتمال شرطی $p(y|x)$ برای دو متغیر تصادفی X و Y به صورت تعریف می‌شود.

$$p(y|x) \stackrel{\text{def}}{=} \frac{p(x,y)}{p(x)} \qquad (1.1)$$

که $p(x) > 0$

تمرین ۲: بررسی کنید که $p(y|x)$ بر روی تمام مقادیر $y \in Y$ برای یک $x \in X$ معین ثابت، با 1 ادغام می‌شود، و بنابراین شرایط یک تابع جرم احتمالی (چگالی) را برآورده می‌کند.

معادله (۱,۱) اکنون به ما امکان می‌دهد تا با توجه به مشاهدات x، احتمال پیشین یک رویداد A را محاسبه کنیم.

$$P(Y \in A|X = x) = \begin{cases} \sum_{y \in A} p(y|x) & Y: discrete \\ \int_A p(y|x)dy & Y: continuous \end{cases}$$

نوشتن $p(x,y) = p(x|y)p(y) = p(y|x)p(x)$ قاعده ضرب نامیده می‌شود. گسترش بیش از دو متغیر ساده است. ما میتوانیم بنویسیم

$$p(x_1, \ldots, x_d) = p(x_d|x_1, \ldots, x_{d-1})p(x_1, \ldots, x_{d-1})$$

با استفاده تابع بازگشتی از قاعده ضرب، به دست می‌آوریم

$$p(x_1, \ldots, x_d) = p(x_d|x_1, \ldots, x_{d-1})p(x_1, \ldots, x_{d-1})$$
$$= p(x_d|x_1, \ldots, x_{d-1})p(x_{d-1}|x_1, \ldots, x_{d-2})p(x_1, \ldots, x_{d-2})$$
$$.$$
$$.$$
$$.$$
$$p(x_d|x_1, \ldots, x_{d-1})p(x_{d-1}|x_1, \ldots, x_{d-2}) \ldots p(x_2|x_1)p(x_1)$$

به طور فشرده تر

$$\mathrm{p}(\mathrm{x}_1,\ldots,\mathrm{x}_d) \;=\; \mathrm{p}(\mathrm{x}_1)\prod_{i=2}^{d} p(x_i|x_1,\ldots,x_{i-1}) \qquad (1.2)$$

که از آن به عنوان قانون زنجیره‌ای یا قاعده کلی ضرب یاد می‌شود. به عنوان مثال، برای سه متغیر، قاعده ضرب می‌دهد

$$\mathrm{p}(\mathrm{x}_1, x_2, \mathrm{x}_3) \;=\; \mathrm{p}(\mathrm{x}_3|\mathrm{x}_{2,}\mathrm{x}_1)p(\mathrm{x}_2|\mathrm{x}_1)p(\mathrm{x}_1)$$

این قاعده برای مجموعه‌ای از متغیرهای تصادفی نیز اعمال می‌شود، جایی که یک مجموعه را می‌توان به عنوان یک متغیر تصادفی در نظر گرفت. مثلا،

$$\mathrm{p}(\mathrm{x}_1, x_2, \mathrm{x}_3) = \mathrm{p}(\mathrm{x}_3|\mathrm{x}_{2,}\mathrm{x}_1)p(\mathrm{x}_2|\mathrm{x}_1)p(\mathrm{x}_1)$$

این به این دلیل است که $(\mathrm{x}_2، \mathrm{x}_3)$ یک فضای احتمال معتبر دارند، بنابراین می‌توانیم از قاعده ضرب برای دو متغیر استفاده کنیم: x_1 و $(x_2، x_3)$. با استفاده از قاعده ضرب، با دادن $p(x,y) \;=\; p(x|y)p(y) \;=\; p(y|x)p(x)$ می‌توانیم قانون بیز را نیز استخراج کنیم:

$$\mathrm{p}(\mathrm{x}|\mathrm{y}) \;=\; \frac{\mathrm{p}(\mathrm{y}|\mathrm{x})\mathrm{p}(\mathrm{x})}{\mathrm{p}(\mathrm{y})} \qquad (1.3)$$

بنابراین، واقعاً فقط باید قاعده ضرب را به خاطر بسپارید تا به راحتی قانون بیز را به خاطر بیاورید

ممکن است متوجه شوید که ترتیب متغیرها در قاعده ضرب اهمیتی ندارد. در واقع تا حدودی جالب است که ما می‌توانیم توزیع شرطی $p(x|y)$ و $p(y)$ حاشیه‌ای را تعریف کنیم یا می‌توانیم $p(y|x)$ و $p(x)$ را تعریف کنیم و هر دو به طور معادل توزیع مشترک $p(\,x,y)$. این ویژگی به سادگی یک واقعیت از تعریف توزیع‌های شرطی است و در هنگام تخمین توزیع‌ها انعطاف‌پذیری را فراهم می‌کند. ما بیشتر از این هم‌ارزی در قالب قانون بیز، هنگام انجام تخمین پارامتر و ماکسیمم احتمال استفاده خواهیم کرد. برای کار در مدل‌های گرافیکی، که در اینجا مورد بحث قرار نمی‌گیرد، این انعطاف‌پذیری از اهمیت بیشتری برخوردار است.

۲-۳-۱ متغیرهای تصادفی مستقل

دو متغیر تصادفی مستقل هستند اگر عوامل توزیع احتمال مشترک آنها، حاصل ضرب حاشیه‌ها باشد

$$\mathrm{p}\,(\mathrm{x},\mathrm{y}) \;=\; \mathrm{p}(\mathrm{x})\mathrm{p}(\mathrm{y})$$

یکی از دلایل شهودی این تعریف را می‌توان با در نظر گرفتن X به شرط Y مشاهده کرد. اگر $\mathrm{p}(\mathrm{x}|\mathrm{y}) \;=\; \mathrm{p}(\mathrm{x})$، این بدان معناست که مقدار Y هیچ تأثیری بر توزیع روی X ندارد و بنابراین آنها مستقل هستند. از قاعده حاصلضرب، $\mathrm{p}(\mathrm{x},\mathrm{y}) \;=\; \mathrm{p}(\mathrm{x}|\mathrm{y})\mathrm{p}(\mathrm{y})$ را می‌دانیم و از آنجایی که $\mathrm{p}(\mathrm{x}|\mathrm{y}) \;=\; \mathrm{p}(\mathrm{x})$ $\mathrm{p}(\mathrm{x},\mathrm{y}) \;=\; \mathrm{p}(\mathrm{x})$ را بدست می‌آوریم. $\mathrm{p}(\mathrm{y})$ همانطور که در بالا تعریف شد.

مفهوم استقلال را می‌توان به بیش از دو متغیر تصادفی تعمیم داد. به طور کلی‌تر، اگر بتوان توزیع احتمال مشترک هر زیرمجموعه‌ای از متغیرها را به‌عنوان حاصلضرب توزیع‌های احتمال حاشیه‌ای اجزای آن بیان کرد، به d متغیر تصادفی مستقل یا مشترکاً مستقل گفته می‌شود.

$$\mathrm{p}\,(\mathrm{x}_1,\mathrm{x}_2,\ldots,\mathrm{x}_d) \;=\; \mathrm{p}(\mathrm{x}_1)\mathrm{p}(\mathrm{x}_2)\ldots\mathrm{p}(\mathrm{x}_d)$$

شکل دیگری از استقلال، به نام استقلال شرطی، حتی بیشتر در یادگیری ماشین استفاده می‌شود. این نشان دهنده استقلال بین متغیرها در حضور برخی متغیرهای تصادفی دیگر (شواهد) است. به عنوان مثال،

$$\mathrm{p(x, y|z) = p(x|z)p(y|z)}$$

جالب اینجاست که این دو شکل استقلال با هم ارتباطی ندارند: هیچ کدام متضمن دیگری نیست. X و Y می‌توانند مستقل باشند، اما نه به طور مشروط مستقل با توجه به Z. X و Y می‌توانند به صورت شرطی مستقل باشند، اما مستقل نیستند. ما این را در دو مثال ساده در شکل A.1 در پیوست نشان می‌دهیم.

در اینجا، مثالی ارائه می‌دهیم که مستقیماً با یادگیری ماشین مرتبط است، در مورد اینکه چرا به استقلال و استقلال مشروط اهمیت می‌دهیم. اگر دو متغیر مستقل باشند، این مفاهیم مدلسازی مهمی دارد. برای مثال، اگر ویژگی X و هدف Y مستقل باشند، X برای پیش‌بینی Y مفید نیست و بنابراین ویژگی مفیدی نیست. اگر دو متغیر با توجه به متغیر دیگری به صورت شرطی مستقل باشند، این نیز می‌تواند مفاهیم مدل‌سازی مهمی داشته باشد. به عنوان مثال، اگر ما دو ویژگی $\mathrm{X_1}$ و $\mathrm{X_2}$ با هدف Y داشته باشیم، که در آن X_2 و Y به طور مشروط مستقل از X_1 هستند، ویژگی X_2 اضافی است و می‌تواند به طور بالقوه کنار گذاشته شود.

به عنوان یک مثال عینی، اجازه دهید $\mathrm{X_1 = }$ temperature in Celcius و $\mathrm{X_2 = }$ temperature in Fahrenheit ، با $\mathrm{Y = }$ plants need watering. Y قطعا مستقل از $\mathrm{X_2}$ نیست. با این حال، هنگامی که X_1 شناخته شد (یا داده شد)، دیگر اطلاعات اضافی از X_2 به دست نمی‌آید و بنابراین $\mathrm{p(y|x_1, x_2) = p(y|x_1) = p(y|x_2)}$. به طور کلی، شناخت استقلال‌ها و استقلال‌های مشروط می‌تواند روند مدل‌سازی را اطلاع‌رسانی و ساده‌سازی کند و نمونه‌های متعددی را از نظر ساده‌سازی ماکسیمم احتمال برای متغیرهای تصادفی مستقل و با توزیع یکسان[1] یا به اختصار $\mathrm{i.i.d}$ را برای داده‌ها و در بیز ساده برای طبقه بندی خواهیم دید. ما این بخش را با یک مثال دیگر، با استفاده از یک سکه مغرضانه، به پایان می‌بریم تا تمایز بین استقلال و استقلال مشروط را برجسته کنیم.

مثال ۷: [سکه مغرضانه و استقلال مشروط] فرض کنید یک تولیدکننده یک سکه مغرضانه تولید کرده است، جایی که به طور تصادفی سر (H) یا دم (T) را نمی‌دهد. در عوض، در واقع مقداری احتمال ناشناخته α برای دیدن H در هنگام چرخاندن سکه را دارد. از آنجایی که این سوگیری ناشناخته است، ما عدم قطعیت خود را با تعریف یک Z تصادفی (بایاس سکه) رمزگذاری می‌کنیم. به طور کلی، این متغیر تصادفی می‌تواند مقادیر $[0, 1]$را بگیرد. برای اهداف این مثال، اجازه دهید این را کمی ساده‌تر کنیم و فرض کنیم که می‌دانیم بایاس یکی از $Z = \{0.1, 0.5, 0.8\}$ است. اگر سوگیری 0.5 باشد، به این معنی است که این یک سکه بی طرفانه (منصفانه) است. اجازه دهید فرض کنیم که احتمال هر سوگیری به یک اندازه محتمل است، یعنی $P(Z = z) = \frac{1}{3}$، زیرا سازنده هیچ دلیلی به ما نداده است که هر یک از 0.1، 0.5 یا 0.8 را محتمل‌تر بدانیم. حال تصور کنید که سکه را دو بار برگردانید و دو نتیجه x_1 و x_2 را ثبت کنید. این دو تلنگر جداگانه با دو متغیر تصادفی X_1 و X_2 مطابقت دارند. فضای نتیجه برای $\mathrm{X_1}$ و $\mathrm{X_2}$، $\{\mathrm{H, T}\}$ است. با توجه به سوگیری واقعی سکه، α، توزیع واقعی برنولی $\mathrm{P(X_i = H|Z = \alpha) = \alpha}$ است. با این حال، ما سوگیری α را نمی‌دانیم. در عوض، ما آن را با یک متغیر تصادفی Z مدل می‌کنیم، به طوری که برای هر $Z = z$ داده شده، می‌دانیم که $P(X_i|Z = z) = z$، یعنی X_i یک متغیر تصادفی

[1] Independent and identically distributed random variables

برنولی با پارامتر z است. از آنجایی که ما سوگیری واقعی را نمی‌دانیم، باید روی Z به حاشیه برسیم تا توزیع حاشیه ای را روی X_1 بدست آوریم.

$$P(X_1 = x) = \sum_{z \in Z} P(X_1 = x, Z = z)$$

$$= \sum_{z \in Z} P(X_1 = x | Z = z) P(Z = z)$$

$$= P(X_1 = x | Z = 0.1) P(Z = 0.1) + P(X_1 = x | Z = 0.5) P(Z = 0.5) + P(X_1 = x | Z = 0.8) P(Z = 0.8)$$

آیا X_1 و X_2 به صورت مشروط مستقل از Z هستند؟ پاسخ مثبت است، زیرا با توجه به سوگیری سکه، دانستن نتیجه X_2 بر توزیع بر روی X_1 تأثیری ندارد، به عنوان مثال،

$$P(X_1 = x_1, X_2 = x_2 | Z = z) = P(X_1 = x_1 | Z = z) P(X_2 = x_2 | Z = z)$$

صرف نظر از آنچه برای X_2 مشاهده می‌کنیم، می‌دانیم که توزیع بر روی X_1 برنولی با بایاس z داده شده است.

آیا X_1 و X_2 مستقل هستند؟ پاسخ منفی است، زیرا بدون دانستن سوگیری سکه، دانستن نتیجه X_2 چیزی در مورد توزیع برنولی بر روی X_2 به ما می‌گوید. برای مثال، اگر $X_1 = T$ و $X_2 = H$، نتیجه دوم نشان می‌دهد که سوگیری ممکن است کاملاً به سمت T منحرف نشود.

$$P(X_1 = x_1, X_2 = x_2) = \sum_{z \in Z} P(X_1 = x_1, X_2 = x_2 | Z = z) P(Z = z)$$

$$= \sum_{z \in Z} P(X_1 = x_1 | Z = z) P(X_2 = x_2 | Z = z) P(Z = z)$$

که برابر با $P(X_1 = x_1) P(X_2 = x_2)$ تضمین نشده است، که در آن

$$P(X_1 = x_1) P(X_2 = x_2)$$

$$= \left(\sum_{z_1 \in Z} P(X_1 = x_1 | Z = z_1) P(Z = z_1) \right) \left(\sum_{z_2 \in Z} P(X_2 = x_2 | Z = z_2) P(Z = z_2) \right)$$

۱-۴ امیدهای ریاضی و گشتاور

مقدار مورد امید ریاضی، یا میانگین، متغیر تصادفی X، میانگین x نمونه برداری مکرر در محدوده نمونه گیری است. این لزوماً مقداری نیست که ما امید داریم اغلب آن را ببینیم - که حالت نامیده می‌شود. به طور دقیق تر، با توجه به pmf p یا pdf pبرای فضای نتیجه X، امید X است

$$\mathbb{E}[X] \stackrel{\text{def}}{=} \begin{cases} \sum_{x \in X} xp(x) & if\ X\ is discrete \\ \int_x xp(x)dx & if\ X\ is\ continuous \end{cases}$$

برای تاس انداختن، که در آن هر عدد از 1 تا 6 دارای احتمال یکنواخت است، مقدار مورد انتظار 3.5 است و حالت برای همه اعداد گره خورده است (یعنی چند وجهی است). برای توزیع برنولی، که در آن $X = \{0, 1\}$، مقدار مورد انتظار α است، که حتی نتیجه‌ای نیست که مشاهده شود، اما اگر سکه را بی‌نهایت بار برگردانیم، میانگین 0 و 1 است. حالت در این مورد به α بستگی دارد: اگر $\alpha > 0.5$ باشد، احتمال 1 بیشتر است، آنگاه حالت 1 است. اگر $\alpha < 0.5$ باشد، حالت 0 است. در غیر این صورت، دو وجهی با مدهای 0 و 1 است. برای توزیع گاوسی، مقدار مورد انتظار پارامتر μ است و مد نیز برابر μ است. به طور کلی، ممکن است به مقدار مورد انتظار توابع متغیر تصادفی X علاقه‌مند باشیم. برای مثال، ممکن است بخواهیم $E[X^2]$ یا به طور کلی $E[X^k]$ را برای مقداری $k > 1$ بدانیم. یا ممکن است بخواهیم بدانیم $E[(X - c)^k]$ برای مقداری $k > 1$ و ثابت c. به این گشتاور X می‌گویند. به طور کلی، برای تابع $f : X \to R$، می‌توانیم $f(X)$ را یک متغیر تصادفی تبدیل شده در نظر بگیریم و امید آن را به صورت تعریف کنیم.

$$\mathbb{E}[f(X)] = \mathbb{E}[f(X)] = \begin{cases} \sum_{x \in X} f(x)p(x) & if\ X\ is\ discrete \\ \int_X f(x)p(x)dx & if\ X\ is\ continuous \end{cases}$$

اگر $\mathbb{E}[f(X)] = \pm\infty$، می‌گوییم که امید ریاضی وجود ندارد یا به خوبی تعریف نشده است.

یک گشتاور مفید، واریانس است: گشتاور دوم گرایش به مرکز، که در آن گرایش به مرکز نشان دهنده $c = \mathbb{E}[X]$ است. واریانس مقداری را نشان می‌دهد که متغیر تصادفی حول میانگین آن تغییر می‌کند. به عنوان مثال، برای توزیع گاوسی، اگر واریانس σ^2 بزرگ باشد، گاوس بسیار گسترده است، که نشان دهنده چگالی غیر قابل اغماض برای محدوده وسیع‌تری از نقاط x در اطراف μ است. متناوبا، اگر σ^2 تقریباً صفر باشد، گاوسی به شدت در اطراف μ متمرکز می‌شود. همچنین می‌توانیم امیدهای شرطی و امیدها را برای متغیرهای تصادفی چند متغیره در نظر بگیریم. برای دو متغیر تصادفی X و Y و تابع $f : Y \to \mathbb{R}$، امید شرطی است

$$\mathbb{E}[f(Y)|X = x] = \begin{cases} \sum_{y \in Y} f(y)p(y|x) & if\ Y\ is\ discrete \\ \int_y f(y)p(y|x)dy & if\ Y\ is\ continuous \end{cases}$$

استفاده از تابع همانی $f(y) = y$ به امید شرطی استاندارد $\mathbb{E}[Y|x]$ منجر می‌شود.

تمرین ۳: قانون کل امید ریاضی را نشان دهید: $\mathbb{E}[Y] = \mathbb{E}[\mathbb{E}[Y|X]]$ ، که در آن امید بیرونی بیش از X و امید درونی بیش از Y است. برای مثال، اگر Y و X هر دو گسسته باشند

$$\mathbb{E}[\mathbb{E}[Y|X]] = \sum_{x \in X} p(x)\mathbb{E}[Y|X = x]$$

$$= \sum_{x \in X} p(x) \sum_{y \in Y} yp(y|x)$$

برای دو متغیر تصادفی X و Y و $f: X \times Y \to \mathbb{R}$، همچنین می‌توانیم امید را بر روی توزیع مشترک تعریف کنیم، با یک متغیر ثابت

$$\mathbb{E}\,[f(X,y)] \;=\; \begin{cases} \sum_{x\in\mathcal{X}} f(x,\,y)p(x|y) & \text{if X is discrete} \\ \int_{\mathcal{X}} f(x,\,y)p(x|y)dx & \text{if X is continuous} \end{cases}$$

یا بیش از هر دو متغیر

$$\mathbb{E}\,[f(X,Y)] \;=\; \begin{cases} \sum_{y\in\mathcal{Y}} p(y)E[f(X,y)] & \text{if Y isdiscrete} \\ \int_{y} p(y)E[f(X,y)]dy & \text{if Y iscontinuous} \end{cases}$$

به عنوان مثال، اگر X پیوسته و Y گسسته باشد، این نشان می‌دهد

$$\mathbb{E}[f(X,Y)] = y \in Y\; p(y)\mathbb{E}[f(X,y)]$$

تمرین ۴: نشان دهید $\int_{\mathcal{X}}\left(\sum_{y\in\mathcal{Y}}\{f(x,\,y)p(x,\,y)\right)dx \;=\; \int_{\mathcal{X}} \mathbb{E}[f(x,Y)]p(x)dx$

همانطور که در بالا در مورد واریانس، کوواریانس یک نمونه مهم از این مقادیر مورد انتظار است، با $f(x,y) \;=\; (x-\mathbb{E}[X])(y-\mathbb{E}[Y])$. مقدار مورد انتظار در این تابع نشان می‌دهد که چگونه دو متغیر با هم متفاوت هستند. ما از علامت گذاری خاص برای کوواریانس استفاده می‌کنیم، که اغلب استفاده می‌شود

$$\mathrm{Cov}[X,Y] = \mathbb{E}[(X-\mathbb{E}[X])(Y-\mathbb{E}[Y])]$$

$$=\; \mathbb{E}\,[XY\,] \;-\; \mathbb{E}\,[X]\,\mathbb{E}\,[Y\,]$$

با $Cov[X,X] \;=\; V\,[X]$ واریانس متغیر تصادفی X است. همبستگی کوواریانس است که با انحراف استاندارد - ریشه دوم واریانس - هر متغیر تصادفی نرمال شده است.

$$\mathrm{Corr}[X,Y\,] \;=\; \frac{\mathrm{Cov}[X,Y\,]}{\sqrt{V\,[X]}\,\sqrt{V\,[Y\,]}}$$

اگر X و Y خود واریانس زیادی داشته باشند، کوواریانس می‌تواند بزرگتر شود. از سوی دیگر، همبستگی بین **1-** و **1** تضمین شده است، و به همین ترتیب یک معیار متغیر مقیاس از نحوه تغییر متغیرها با هم است. در بسیاری از شرایط ما نیاز به تجزیه و تحلیل بیش از دو متغیر تصادفی داریم. یک خلاصه دو بعدی ساده از تمام مقادیر کوواریانس زوجی شامل d متغیرهای تصادفی $X_1، X_2، \ldots، X_d$ را ماتریس کوواریانس می‌گویند. ماتریس کوواریانس $\Sigma \;\in\; \mathbb{R}^{d\times d}$ دارای ورودی (i,j) است که به صورت تعریف شده است.

$$\Sigma_{\mathrm{ij}} \;=\; \mathrm{Cov}[\mathrm{Xi}\,,\mathrm{X}_j]$$

$$=\; \mathbb{E}\,[(\mathrm{X}_i \;-\; \mathbb{E}\,[\mathrm{X}_i])\;(\mathrm{X}_j \;-\; \mathbb{E}\,[\mathrm{X}_j])]$$

با ماتریس کامل به صورت نوشته شده است

$$\mathbf{\Sigma} = \mathrm{Cov}[\mathbf{X},\mathbf{X}]$$

$$=\; \mathbb{E}[(\mathbf{X} \;-\; \mathbb{E}[\mathbf{X}])(\mathbf{X} \;-\; \mathbb{E}[\mathbf{X}])^T]$$

$$= \mathbb{E}[\mathbf{X}\mathbf{X}^T] - \mathbb{E}[\mathbf{X}]\mathbb{E}[\mathbf{X}]^T$$

خط دوم شامل حاصلضرب بیرونی بردار $v = \boldsymbol{X} - E[\boldsymbol{X}] \in \mathbb{R}^d$ است تا $\boldsymbol{A} = vv^T \in \mathbb{R}^{d\times d}$ را تولید کند. این حاصل ضرب بیرونی یک ضرب ماتریس است که ماتریس اول $d \times 1$ و دومی $1 \times d$ است. با استفاده از قوانین ضرب ماتریس، $\boldsymbol{A_j} = v_i v_j$ به دست می‌آید. توجه داشته باشید که عناصر قطری ماتریس کوواریانس d × d واریانس برای هر متغیر X_i و عناصر خارج از مورب مقادیر کوواریانس بین جفت متغیرها هستند. ماتریس کوواریانس متقارن و مثبت نیمه معین است. به یاد بیاورید که اگر $\boldsymbol{z}^T \Sigma \boldsymbol{z} \geq 0$ برای همه بردارها $z \neq 0$ یک ماتریس نیمه معین مثبت (که با $\Sigma \succcurlyeq 0$ نشان داده می‌شود) گفته می‌شود. به طور معادل، مقادیر ویژه ماتریس همگی بزرگتر یا مساوی صفر هستند. اگر ماتریس مثبت قطعی به جای مثبت نیمه قطعی ($\Sigma \succcurlyeq 0$) باشد، آنگاه مقادیر ویژه کاملا مثبت هستند و ماتریس کوواریانس رتبه کاملی دارد. یک ماتریس نیمه معین مثبت می‌تواند مقادیر ویژه ای داشته باشد که صفر هستند و بنابراین دارای رتبه کوچکتر از d هستند. برای d = 1، این خاصیت نیمه معین مثبت ماتریس با خاصیت $\sigma^2 \geq 0$ برای واریانس اسکالر مطابقت دارد.

خواص امید ریاضی

در اینجا به بررسی برخی از خواص مفید انتظارات می‌پردازیم. متغیرهای تصادفی چند متغیره، $\mathbf{X} \in \mathbb{R}^d$ و $\mathbf{Y} \in \mathbb{R}^m$ را برای $d, m \in N$، با متغیرهای تصادفی تک متغیره به عنوان یک مورد خاص در نظر بگیرید. برای ثابت $c \in \mathbb{R}$، به این صورت است که:

۱. $\mathrm{E}[c\mathbf{X}] = c\mathrm{E}[\mathbf{X}] \in \mathbb{R}^d$

۲. $\mathrm{E}[\mathbf{X} + \mathbf{Y}] = \mathrm{E}[\mathbf{X}] + \mathrm{E}[\mathbf{Y}]$ وقتی که d = m

۳. $\mathrm{V}[c] = 0$ واریانس یک ثابت صفر است

۴. $\mathrm{V}[\mathbf{X}] \stackrel{\text{def}}{=} \mathrm{Cov}[\mathbf{X}, \mathbf{X}] \geq 0$ که در آن برای $d = 1$، $\mathrm{V}[\mathbf{X}] \geq 0$ یک اسکالر است.

ما از $\mathrm{V}[\mathbf{X}]$ به عنوان مخفف $\mathrm{Cov}[\mathbf{X}, \mathbf{X}]$ استفاده می‌کنیم.

۵. $\mathrm{V}[c\mathbf{X}] = c^2\,\mathrm{V}[\mathbf{X}] \in \mathbb{R}^{d\times d}$

۶. $\mathrm{Cov}[\mathbf{X}, \mathbf{Y}] = \mathrm{E}[(\mathbf{X} - \mathrm{E}[\mathbf{X}])(\mathbf{Y} - \mathrm{E}(\mathbf{Y})^T] = \mathrm{E}[\mathbf{X}\mathbf{Y}^T] - \mathrm{E}[\mathbf{X}]\mathrm{E}[\mathbf{Y}]^T \in \mathbb{R}^{d\times m}$

۷. $\mathrm{V}[\mathbf{X} + \mathbf{Y}] = \mathrm{V}[\mathbf{X}] + \mathrm{V}[\mathbf{Y}] + 2\mathrm{Cov}[\mathbf{X}, \mathbf{Y}]$ وقتی که d = m

علاوه بر این، اگر $\boldsymbol{X}$ و $\boldsymbol{Y}$ متغیرهای تصادفی مستقل باشند، چنین است که:

۸. $\mathrm{E}[\mathrm{X}_i\mathrm{Y}_j] = \mathrm{E}[\mathrm{X}_i]\,\mathrm{E}[\mathrm{Y}_j]$ برای هر i, j

۹. $\mathrm{V}[\mathbf{X} + \mathbf{Y}] = \mathrm{V}[\mathbf{X}] + \mathrm{V}[\mathbf{Y}]$ وقتی که d = m

۱۰. $\mathrm{Cov}[\mathbf{X}, \mathbf{Y}] = 0$

در نهایت، برای هر متغیر تصادفی d بعدی، $\boldsymbol{X_1} + \boldsymbol{X_2} + \ldots + \boldsymbol{X_m}$

۱۱. $Cov[\boldsymbol{X}_1 + \boldsymbol{X}_2 + \ldots + \boldsymbol{X}_m] = \sum_{i=1}^{m}\sum_{j=1}^{m} Cov[\boldsymbol{X}_i, \boldsymbol{X}_j] = \sum_{i=1}^{m} \mathrm{V}[\mathbf{X}_i] + 2\sum_{1\leq i<j\leq m} Cov[\boldsymbol{X}_i, \boldsymbol{X}_j]$

۱-۵ چند متغیره PMF و PDF

ما اکنون افزونه‌هایی را برای تعاریف pmf و pdf برای دسته بندی چند متغیره‌ها در نظر می‌گیریم. توزیع گاوسی چند متغیره تعمیم توزیع گاوسی یا نرمال به حالت d-بعدی با $\Omega = \mathbb{R}^d$ است. به عنوان تعریف شده است

$$\mathrm{p}(\omega) \stackrel{\text{def}}{=} \frac{1}{\sqrt{(2\pi)^d|\Sigma|}} exp(-\frac{1}{2}(\omega - \mu)^T \Sigma^{-1}(\omega - \mu))$$

با پارامترهای $\mu \in \mathbb{R}^d$ و ماتریس مثبت- معین Σ که ماتریس کوواریانس است. این تعریف نحوه تغییر متغیرها با هم را در نظر می‌گیرد که توسط ماتریس کوواریانس Σ ارائه شده است. به عنوان مثال، اگر متغیرها مستقل باشند، ماتریس کوواریانس مورب است. علاوه بر این، اگر هر متغیر دارای واریانس واحد باشد، گاوسی کروی است. اگر برخی از ابعاد واریانس بالاتری داشته باشند، گاوسی بیضی شکل است. اگر متغیرها مستقل نباشند، گاوسی برش می‌شود و از میانگین آن به طور متفاوتی منحرف می‌شود که صرفاً با واریانس متغیرها قابل محاسبه است. ما به این توزیع به عنوان $N(\mu, \Sigma)$ اشاره می‌کنیم و گاهی اوقات از $N(\omega|\mu, \Sigma)$ برای نشان دادن pdf برای μ و Σ داده شده استفاده می‌کنیم.

چند اصطلاح جدید در این pdf وجود دارد، بنابراین اجازه دهید این فرمول را بررسی کنیم. اصطلاح $|\Sigma|$ تعیین کننده Σ است. دترمینان برابر حاصل ضرب مقادیر ویژه Σ است. عرض توزیع را منعکس می‌کند. هنگامی که $d = 1$، تعیین کننده به سادگی برابر با σ^2 است، و ما طبق معمول با واریانس توزیع مقیاس می‌کنیم. به عنوان مثال دیگر، موردی را در نظر بگیرید که در آن کوواریانس بین همه متغیرها صفر است. تصور کنید یک تابع گاوسی همه یک‌ها را در مورب برای کوواریانس داشته باشد، و دیگری دارای واریانس 5 برای اولین عنصر قطری و $1s$ در غیر این صورت. تعیین کننده اولی 1 و دومی 5 است. منطقی است که برای گاوسی دوم که واریانس بیشتری در یک بعد دارد، مقیاس بیشتری داشته باشیم. دومین عبارت جدید $(\omega - \mu) > \Sigma - 1(\omega - \mu)$ است. این در واقع با یک محصول نقطه وزنی مطابقت دارد. دوباره یک ماتریس کوواریانس مورب را در نظر بگیرید، با $d = 3$ با 5، 0.5 و 2 در قطر. سپس، برای $z = \omega - \mu$

$$z^T\Sigma^{-1}z = \frac{{z_1}^2}{5} + \frac{{z_2}^2}{0.5} + \frac{{z_3}^2}{2}$$

برای متغیر تصادفی گسسته، تعمیم به ابعاد چندگانه برای pmf مستقیم است: pmf چند بعدی به سادگی با جداول احتمال چند بعدی مطابقت دارد. نمونه ای از آن را در جدول ۱،۱ دیدیم. با این حال، مانند حالت تک متغیره، چند pmf با نام وجود دارد، زیرا اغلب از آنها استفاده می‌شود.

یک مثال از یک pmf چند بعدی نامگذاری شده، توزیع طبقه‌ای است که نمونه‌ای از توزیع چندجمله‌ای است. توزیع طبقه‌ای برای مدل‌سازی یک متغیر تصادفی d-بعدی استفاده می‌شود که در آن هر عنصر می‌تواند 0 یا 1 باشد. این توزیع می‌تواند به طور معادل برای مدل‌سازی یک متغیر تصادفی اسکالر با d نتایج ممکن استفاده شود. با این حال، وقتی از آن در بخش ۸،۳ استفاده می‌کنیم، مفید خواهد بود که توزیع طبقه ای را به عنوان یک pmf برای یک متغیر تصادفی d بعدی در نظر بگیریم. هر نقطه $(\mathrm{k}_1, \mathrm{k}_2, \ldots, \mathrm{k}_d)$ در فضای نتیجه یک بردار باینری است که دقیقاً یک عنصر $\mathrm{k}_i = 1$ و بقیه صفر دارد، که نشان می‌دهد که نتیجه i رخ داده است. pmf طبقه بندی شده به این صورت تعریف می‌شود

$$\mathrm{p}(\mathrm{k}_1, \mathrm{k}_2, \ldots, \mathrm{k}_d) \stackrel{\text{def}}{=} \begin{cases} {\propto_1}^{k_1}{\propto_2}^{k_2}{\propto_d}^{k_d} & if\ k_1 + k_2 + \ldots + k_d = n \\ 0 & \text{otherwise} \end{cases}$$

که در آن α_iها ضرایب مثبت هستند به طوری که $\sum_{i=1}^{d} \propto_i = 1$. یعنی هر ضریب α_i احتمال نتیجه i را می‌دهد.

مثال ۸: [مشتق ابعاد برای متغیرهای تصادفی گسسته چند بعدی] یکی از راه‌های اجتناب از pdf گسسته کردن متغیرها و سپس تعریف یک pmf است که جدولی از مقادیر احتمال است. اگرچه در برخی موارد معقول است، اما به طور کلی این می‌تواند به طور تصاعدی تعداد پارامترهای توزیع احتمال را افزایش دهد و نمونه ای از مشتق ابعاد است.

برای اینکه بفهمید چرا، مثال زیر را در نظر بگیرید. فرض کنید یک متغیر تصادفی d بعدی داریم که هر ورودی مقادیری بین 0 و 1 دارد (یعنی $Xi = [0,1]$). شما تصمیم می‌گیرید این را گسسته کنید تا هر ورودی در یکی از سه سطل قرار گیرد و $X_i = \{1,2,3\}$ را تغییر دهید. اکنون شما انعطاف زیادی در تعیین این احتمالات در pmf خود دارید، برخلاف گوسی که فرم عملکردی سخت تری دارد. با این حال، متأسفانه، این جدول مقادیر می‌تواند بسیار بزرگ باشد، با ورودی‌های 3^d. برای این کار باید مقادیر احتمالی $3^d - 1$ را مشخص کنید، جایی که یکی از مقادیر به طور خودکار روی یک منهای مجموع همه احتمالات دیگر تنظیم می‌شود تا اطمینان حاصل شود که یک pmf معتبر دارید.

اگر در عوض، یک توزیع گاوسی روی این متغیرها مشخص کرده بودید، تعداد پارامترهای مورد نیاز برای تعریف توزیع فقط $d + d^2$ است که احتمالاً بسیار کمتر است.

فصل ۲

مقدمه‌ای بر بهینه‌سازی

بسیاری از مسائل یادگیری ماشین با یافتن تابع بهینه مطابق با یک هدف، با توابع یادگیری سروکار دارند. به عنوان مثال، ممکن است کسی علاقه‌مند به یافتن تابعی باشد $f : \mathbb{R}^d \rightarrow \mathbb{R}$ که تفاوت‌های مجذور برخی از اهداف را برای همه نمونه‌ها به مقدار مینیمم می‌رساند: $\sum_{i=1}^{n}(f(x_i) - y_i)^2$. برای یافتن چنین تابعی، باید درک اولیه‌ای از تکنیک‌های بهینه‌سازی داشته باشید. در این فصل، ابزارهای بهینه‌سازی اساسی را برای اهداف هموار عمومی مورد بحث قرار می‌دهیم. بسیاری از الگوریتم‌ها در یادگیری ماشین بر یک رویکرد ساده تکیه دارند: گرادیان کاهشی. ابتدا در مورد چگونگی به مقدار مینیمم رساندن اهداف با استفاده از گرادیان کاهشی مرتبه اول و دوم بحث می‌کنیم. این نمای کلی تنها بخش کوچکی از بهینه‌سازی را پوشش می‌دهد، اما خوشبختانه، بسیاری از الگوریتم‌های یادگیری ماشین مبتنی بر این رویکردهای بهینه‌سازی ساده هستند. ما بعداً، در فصل ۶، پس‌زمینه‌های بهینه‌سازی بیشتری را ارائه می‌کنیم، زمانی که فرصت استفاده از این دانش اولیه بهینه‌سازی را در دو فصل بعدی داشته باشید.

۲-۱ مسئله بهینه سازی اساسی و نقاط ثابت

یک هدف اصلی بهینه سازی انتخاب مجموعه ای از پارامترهای $w \in \mathbb{R}^d$ برای به مقدار مینیمم رساندن تابع هدف داده شده $c : \mathbb{R}^d \rightarrow \mathbb{R}$ است.

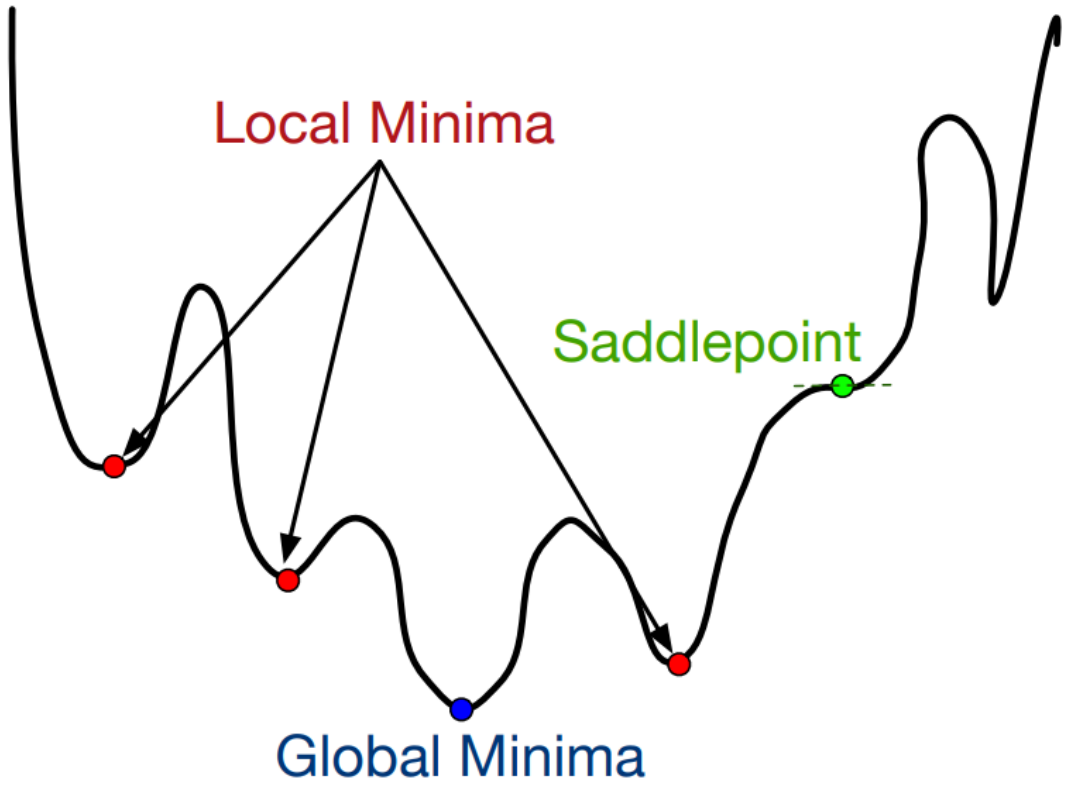
$$\min_{w \in \mathbb{R}^d} c(w)$$

به عنوان مثال، برای به دست آوردن پارامترهای w برای رگرسیون خطی که مجموع مجذور اختلافات را به مقدار مینیمم می‌رساند، از $c(w) = \sum_{i=1}^{n}(x_i, w_i - y_i)^2$ برای حاصل ضرب نقطه استفاده می‌کنیم.

$$< x_i, w_i > = \sum_{j=1}^{d} x_{ij} w_j$$

ما در اینجا به جای خطا از اصطلاح هدف استفاده می‌کنیم، زیرا خطا مفهوم صریحی دارد که مفهوم آن تابع نادرست است. بعداً خواهیم دید که اهداف شامل هر دو معنای عبارت خطا می‌شوند - که نشان می‌دهد با چه دقتی داده‌ها را بازآفرینی می‌کنند - و همچنین عبارت‌هایی که اولویت‌های دیگر را در عملکرد ارائه می‌دهند. ترکیب این عبارات با خطا هدف نهایی را ایجاد می‌کند که مایلیم آن را به مقدار مینیمم برسانیم. به عنوان مثال، برای رگرسیون خطی، ما یک هدف منظم را بهینه می‌کنیم، $c(w) =$

$\sum_{i=1}^{n}((x_i, w_i) - y_i)^2 + \sum_{j=1}^{d} {w_j}^2$ که در آن جمله دوم ترجیحی برای ضرایب کوچکتر w_j را رمزگزاری می‌کند. پس هدف یافتن w است که آن را به مقدار مینیمم برساند. ساده ترین راه حل می‌تواند انجام یک جستجوی تصادفی باشد: $\boldsymbol{w}$ تصادفی تولید کنید و $c(\boldsymbol{w})$ را بررسی کنید. اگر هر $\boldsymbol{w_t}$ جدید تولید شده در تکرار t عملکرد بهتری از بهترین راه حل قبلی $\boldsymbol{w}$ داشته باشد، که در آن $c(\boldsymbol{w_t}) < c(\boldsymbol{w})$ آنگاه می‌توانیم $\boldsymbol{w_t}$ را به عنوان راه حل بهینه جدید تنظیم کنیم. ما فرض می‌کنیم که اهداف ما پیوسته هستند و بنابراین می‌توانیم از این مزیت برای طراحی استراتژی‌های جستجوی بهتر استفاده کنیم. به طور خاص، برای توابع هموار، ما قادر خواهیم بود از گرادیان کاهشی استفاده کنیم که در قسمت بعدی توضیح می‌دهیم.



شکل ۱-۲: نقاط ثابت روی یک سطح عملکرد صاف: مینیمم‌های محلی، مینیمم‌های مطلق و نقاط زین.

گرادیان کاهشی ما را قادر می‌سازد به نقاط ثابت برسیم: نقاط w که در آن شیب صفر است. ابتدا حالت تک متغیره را در نظر بگیرید. مشتق میزان تغییر سطح تابع در نقطه w را به ما می گوید. هنگامی که مشتق هدف در $w \in \mathbb{R}$ صفر باشد، یعنی $\frac{d}{dw} c(\mathbf{w}) = 0$، به این معنی است که سطح تابع به صورت محلی صاف است. چنین نقاطی مطابق شکل ۲,۱ با مینیمم‌های محلی، ماکزیمم‌های محلی و نقاط زینی است.

به عنوان مثال، دوباره فرض کنید که ما در حال انجام رگرسیون خطی هستیم، تنها با یک ویژگی و بنابراین فقط یک وزن $\mathbf{w} \in \mathbb{R}$. مشتق هدف $c(w) = \sum_{i=1}^{n}(x_i \mathbf{w} - y_i)^2$ به این صورت است.

$$\frac{d}{dw} c(w) = \frac{d}{dw} \sum_{i=1}^{n} (x_i w - y_i)^2$$

$$= \sum_{i=1}^{n} \frac{d}{dw}(x_i w - y_i)^2$$

$$= \sum_{i=1}^{n} 2(x_i w - y_i)x_i$$

جایی که آخرین گام از قانون زنجیره پیروی می‌کند. هدف ما یافتن w به گونه‌ای است که $\frac{d}{dw}\ c(w) = 0$؛ هنگامی که چنین نقطه ثابتی را پیدا کردیم، می‌توانیم تعیین کنیم که آیا آن یک مینیمم محلی، ماکسیمم محلی یا نقطه زینی است. از آنجایی که این هدف محدب است، ما در واقع می‌دانیم که تمام نقاط ثابت باید مینیمم جهانی باشند و بنابراین نیازی به انجام این بررسی نداریم. ما در بخش آخر در این مورد بیشتر بحث می‌کنیم، جایی که در مورد برخی از ویژگی‌های اهداف بحث می‌کنیم.

برای حالت چند متغیره، به جای مشتقات، باید گرادیان‌ها را در نظر بگیریم. برای $w \in \mathbb{R}^d$ که $d > 1$ است، باید بپرسیم: بسته به اینکه هر عنصر w چگونه تغییر می‌کند، تابع به صورت محلی چگونه تغییر می‌کند؟ برای تعیین این کمیت، از گرادیان استفاده می‌کنیم که از مشتقات جزئی تشکیل شده است.

$$\nabla \mathrm{c}(\mathbf{w}) = [\frac{\partial \mathrm{c}}{\partial w_1}(\mathbf{w})\ \frac{\partial \mathrm{c}}{\partial w_2}(\mathbf{w})\ \ldots\ldots \frac{\partial \mathrm{c}}{\partial w_d}(\mathbf{w})\ ]$$

هر مشتق جزئی $\frac{\partial \mathrm{c}}{\partial w_j}(\mathbf{w})$ نحوه تغییر تابع c را نشان می‌دهد، زمانی که فقط w_j تغییر می‌کند و دیگری $\mathrm{w}_1, \ldots, \mathrm{w}_{\mathrm{j}-1}, \mathrm{w}_{\mathrm{j}+1}, \ldots, \mathrm{w}_{\mathrm{d}}$ ثابت نگه داشته می‌شوند. به عنوان مثال، برای $c(w = (w_1, w_2)) = \frac{1}{2}(x_1 w_1 + x_2 w_2 - y)^2$، مشتقات جزئی هستند

$$\frac{\partial \mathrm{c}}{\partial \mathrm{w}_1}(w) = (x_1 w_1 + x_2 w_2 - y)x_1$$

$$\frac{\partial \mathrm{c}}{\partial \mathrm{w}_2}(w) = (x_1 w_1 + x_2 w_2 - y)x_2$$

به طور مفید، ما مجبور نیستیم در نظر بگیریم که چگونه کل بردار به طور مشترک در همه متغیرها تغییر می‌کند. بلکه کافی است نقاط ثابت را با یافتن $\mathbf{w}$ در جایی که مشتقات جزئی صفر هستند پیدا کنیم.

۲-۲ گرادیان کاهشی

ایده اصلی پشت شیب نزول، تقریب تابع با تقریب سری تیلور است. این تقریب محاسبه جهت نزول را به صورت محلی روی سطح تابع تسهیل می‌کند. ما با در نظر گرفتن تنظیم تک متغیره، با $\mathrm{w} \in \mathbb{R}$ شروع می‌کنیم. یک تابع $\mathrm{c}(\mathrm{w})$ در همسایگی نقطه w_0، می‌تواند با استفاده از سری تیلور به صورت تقریبی باشد.

$$\mathrm{c}(w) = \sum_{n=0}^{\infty} \frac{c^{(n)}(w_0)}{n!}(w - w_0)^n$$

که $c^{(n)}(w_0)$ آن n-امین مشتق تابع $\mathrm{c}(w)$ است که در نقطه w_0 ارزیابی شده است. این فرض می‌کند که $\mathrm{c}(w)$ بی‌نهایت قابل تفکیک است، اما در عمل ما چنین تصاویر تقریبی چند جمله‌ای را برای n محدود می‌گیریم. یک تقریب مرتبه دوم برای این تابع از سه عبارت اول سری به استفاده می‌کند

$$\mathrm{c}(w) \approx \hat{c}(w) = c(w_0) + (w - w_0)\acute{c}(w_0) + \frac{1}{2}(w - w_0)^2\acute{c}(w_0)$$

یک نقطه ثابت از این $\hat{c}(w)$ را می‌توان با پیدا کردن اولین مشتق و صفر کردن آن به راحتی پیدا کرد.

$$\acute{c}(w) \approx \acute{c}(w_0) + (w - w_0)\acute{c}(w_0) = 0$$

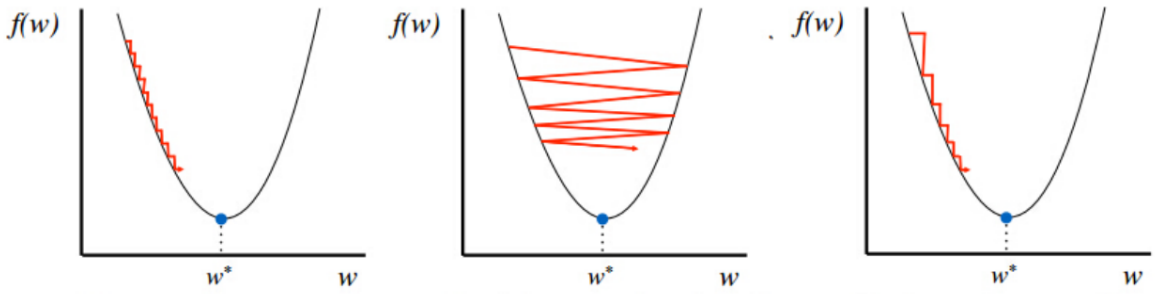
حل این معادله برای w به ما می‌دهد

$$w_1 = w_0 - \frac{c(w_0)}{\acute{c}(w_0)}$$

به صورت محلی، این w_1 جدید یک پیشرفت در w_0 خواهد بود و یک نقطه ثابت از این تقریب محلی $\hat{\mathrm{c}}$ خواهد بود. با این حال، حرکت (به اندازه کافی دور) از w_0 باعث می‌شود این سری محلی تیلور مرتبه دوم نادرست باشد. ما باید تقریب محلی را در این نقطه جدید w_1 بررسی کنیم تا مشخص کنیم که آیا می‌توانیم به صورت محلی بهبود بیشتری داشته باشیم. بنابراین، برای یافتن w بهینه، می‌توانیم به طور مکرر این روش را اعمال کنیم

$$w_{t+1} = w_t - \frac{c(w_0)}{\acute{c}(w_0)} \qquad (2.1)$$

دائماً w_i را بهبود می‌بخشد تا زمانی که به نقطه ای برسیم که مشتق صفر یا تقریباً صفر باشد. این روش را روش نیوتن رافسون یا شیب نزول مرتبه دوم می‌نامند.



(a) Step-size too small (b) Step-size too big (c) Adaptive step-size

شکل ۲٫۲ : مسیرهای بهینه سازی مختلف، به دلیل انتخاب های مختلف اندازه.

در گرادیان کاهشی مرتبه اول، تقریب بد است، جایی که ما دیگر از مشتق دوم استفاده نمی‌کنیم. درعوض، هنگام گرفتن تقریب مرتبه اول، می‌دانیم که عبارت‌های $\mathrm{O}((w - w_0)^2)$ را نادیده می‌گیریم و بنابراین تقریب محلی تبدیل می‌شود به:

$$\mathrm{c}(w) \approx \hat{c}(w) = c(w_0) + (w - w_0)\acute{c}(w_0) + \frac{1}{2\eta}(w - w_0)^2$$

برای مقدار ثابت $\frac{1}{\eta}$ که بزرگی عبارات $O((w - w_0)^2)$ نادیده گرفته شده را منعکس می‌کند. سپس به روزرسانی حاصل برای گام به اندازه η_t است انجام می‌شود

$$w_{t+1} = w_t - \eta_t \acute{c}(w_t) \qquad (2.2)$$

از این، می‌توان دریافت که با توجه به دسترسی به مشتق دوم، یک انتخاب معقول برای اندازه مراحل $\eta_t = \frac{1}{\acute{c}(w_t)}$ است. ما می‌توانیم به طور مشابه چنین قوانینی را برای متغیرهای چند متغیره بدست آوریم. به عنوان مثال، شیب نزول برای $c : \mathbb{R}^d \to \mathbb{R}$ شامل به روزرسانی است

$$\mathbf{w}_{t+1} = \mathbf{w}_t - \eta_t \nabla c(\mathbf{w}_t).$$

به طوری که

$$\nabla c(\mathbf{w}_t) = \left(\frac{\partial c}{\partial w_1}(\mathbf{w}_1), \frac{\partial c}{\partial w_2}(\mathbf{w}_2), \ldots, \frac{\partial c}{\partial w_d}(\mathbf{w}_t)\right) \in \mathbb{R}^d$$

گرادیان تابع c است که در w_t ارزیابی می‌شود. ما در مورد نحوه استخراج این به روز رسانی در تنظیمات چند متغیره در فصل ۶ بحث خواهیم کرد.

۲-۳ انتخاب اندازه گام

بخش مهمی از گرادیان کاهشی (مرتبه اول) انتخاب اندازه گام است. اگر اندازه گام خیلی کوچک باشد، برای رسیدن به یک نقطه ثابت نیاز به تکرارهای زیادی است (شکل ۲,۲(a)). اگر اندازه گام خیلی بزرگ باشد، احتمالاً حول مینیمم نوسان خواهید داشت (شکل ۲,۲ (ب)). چیزی که ما واقعاً می‌خواهیم یک اندازه گام تطبیقی است (شکل ۲,۲ (c))، که احتمالاً بزرگ‌تر شروع می‌شود و سپس با نزدیک شدن به یک نقطه ثابت به آرامی در طول زمان کاهش می‌یابد.

روش اصلی برای به دست آوردن اندازه‌های گام تطبیقی استفاده از جستجوی خط است. این ایده از هدف زیر سرچشمه می‌گیرد: ما می‌خواهیم اندازه گام بهینه را مطابق با آن به دست آوریم

$$\min_{\eta \in \mathbb{R}^+} c(\boldsymbol{w}_t - \eta \nabla c(\boldsymbol{w}_t))$$

راه‌حل این بهینه‌سازی مربوط به بهترین اندازه مقیاس اسکالر است که می‌توانیم برای نقطه $\mathrm{w_t}$ فعلی با جهت نزول $-\nabla c(\mathrm{w_t})$ انتخاب کنیم. حل این بهینه سازی بسیار پرهزینه خواهد بود. با این حال، ما می‌توانیم به سرعت راه حل‌های تقریبی پیدا کنیم. یک انتخاب طبیعی استفاده از یک جستجوی خط عقبگرد است که بزرگترین اندازه گام معقول $\eta_{\max}$ را امتحان می‌کند و سپس آن را کاهش می‌دهد تا هدف کاهش یابد. ایده این است که در امتداد خط ممکن $\eta \in (0, \eta_{\max}]$ جستجو کنید، با این تصور که یک گام بزرگ خوب است - تا زمانی که بیش از حد نباشد. اگر بیش از حد بالا برود و اندازه گام خیلی بزرگ شود، و باید کاهش معمولاً طبق قانون $\tau\eta$ برای مقداری $\tau \in [0.5, 0.9]$ است. برای $\tau = 0.5$، اندازه گام سریعتر کاهش می‌یابد - در هر مرحله از جستجوی خط عقب نشینی به نصف می‌رسد؛ برای $\tau = 0.9$، جستجو آهسته تر از η_{max} عقب می‌نشیند. به محض اینکه اندازه گامی پیدا شد که هدف را کاهش می‌دهد، پذیرفته می‌شود. سپس یک w_t جدید به دست می‌آوریم، دوباره گرادیان را محاسبه می‌کنیم و یک بار دیگر جستجوی خط را از η_{max} شروع می‌کنیم.

می‌توان استراتژی‌های بهتری را برای انتخاب اندازه گام نسبت به این جستجوی ساده تصور کرد. ما در واقع برخی از این موارد را در بخش ۶٫۵ مورد بحث قرار خواهیم داد. با این وجود، این جستجوی خط اولیه، شهودی را برای هدف ما در تطبیق اندازه مراحل فراهم می‌کند.

Algorithm 1: Line Search($w_t, c, g = \nabla c(w_t)$)

1: Optimization parameters: $\eta_{max} = 1.0, \tau = 0.7, tolerance \leftarrow 10e^{-4}$
2: $\eta \leftarrow \eta_{max}$
3: $w \leftarrow w_t$
4: $obj \leftarrow c(w)$
5: while number of backtracking iterations is less than maximum iterations **do**
6: $w \leftarrow w_t - \eta g$
7: // Ensure improvement is at least as much as tolerance
8: **If** $c(w) < obj - tolerance$ **then break**
9: // Else, the objective is worse and so we decrease stepsize
10: $\eta \leftarrow \tau\eta$
11: **if** maximum number of iterations reached **then**
12: // Could not improve solution
13: **return** $w_t, \eta = 0$
14: **return** w, η

۲-۴ خواص بهینه سازی

چندین ویژگی بهینه سازی وجود دارد که باید هنگام مطالعه این کتاب در نظر داشت که در اینجا به آنها اشاره می‌کنیم.

به ماکسیمم رساندن در مقابل به مینیمم رساندن[1] ما تاکنون در مورد هدف به مینیمم رساندن یک هدف بحث کرده‌ایم. یک جایگزین معادل، به ماکسیمم رساندن منفی این هدف است.

$$\underset{\mathbf{w} \in \mathbb{R}^d}{\operatorname{argmin}}\, c(\boldsymbol{w}) = \underset{\mathbf{w} \in \mathbb{R}^d}{\operatorname{argmax}} - c(\boldsymbol{w})$$

جایی که argmin، w را برمی‌گرداند که مینیمم مقدار $\mathrm{c(w)}$ را تولید می‌کند و argmax، w را برمی‌گرداند که ماکسیمم مقدار $-\mathrm{c(w)}$ را تولید می‌کند. مقادیر واقعی مینیمم و ماکسیمم یکسان نیستند، زیرا برای یک جواب بهینه داده شده، $\mathrm{c(w)} \neq -\mathrm{c(w)}$ است. ما تصمیم می‌گیریم هر یک از بهینه‌سازی‌هایمان را به‌عنوان کمینه‌سازی فرمول‌بندی کنیم و شیب نزول را انجام دهیم. با این حال، فرمول‌بندی بهینه‌سازی‌ها به‌عنوان بیشینه‌سازی و انجام صعود گرادیان به همان اندازه معتبر است.

تابع محدب: به تابع $\mathrm{c} : \mathbb{R}^{\mathrm{d}} \rightarrow \mathbb{R}$ محدب گفته می‌شود اگر برای هر $\mathbf{w}_1, \mathbf{w}_2 \in \mathbb{R}^{\mathrm{d}}$ و $\mathrm{t} \in [0,1]$

$$\mathrm{c}(\mathrm{t}\mathbf{w}_1 + (1-\mathrm{t})\mathbf{w}_2) \leq \mathrm{t}\mathrm{c}(\mathbf{w}_1) + (1-\mathrm{t})\mathrm{c}(\mathbf{w}_2) \qquad (2.4)$$

[1] Maximizing versus minimizing

این تعریف به این معنی است که وقتی بین هر دو نقطه در سطح تابع خطی می‌کشیم، مقادیر تابع بین این دو نقطه همگی زیر این خط قرار می‌گیرند. تحدب یک ویژگی مهم است، زیرا به این معنی است که هر نقطه ثابت یک مینیمم مطلق است. بنابراین، صرف نظر از اینکه نزول شیب خود را از کجا شروع می‌کنیم، با اندازه گام‌های مناسب انتخاب شده و تکرارهای کافی، به یک راه حل بهینه خواهیم رسید.

یک تعریف مربوطه یک تابع مقعر است که دقیقاً برعکس است: همه نقاط بالای خط قرار دارند. برای هر تابع محدب c، منفی آن تابع $\mathrm{c}-$ یک تابع مقعر است.

منحصر به فرد بودن راه حل ما اغلب اهمیت می‌دهیم که بیش از یک راه حل برای مشکل بهینه‌سازی ما وجود داشته باشد. در برخی موارد، ما به قابلیت شناسایی اهمیت می‌دهیم، به این معنی که می‌توانیم راه‌حل واقعی را شناسایی کنیم. اگر بیش از یک راه‌حل وجود داشته باشد، ممکن است تصور شود که مشکل دقیقاً مطرح نشده است. برای برخی از مشکلات، مهم یا حتی ضروری است که قابلیت شناسایی داشته باشیم (به عنوان مثال، تخمین درصد افراد مبتلا به یک بیماری) در حالی که برای برخی دیگر ما صرفاً به یافتن یک عملکرد مناسب (پیش‌بینی‌کننده) اهمیت می‌دهیم که به طور منطقی و دقیق اهداف را پیش‌بینی کند، حتی اگر آن را پیش‌بینی کند. چنین عملکرد منحصر به فردی نیست. ما قابلیت شناسایی را در این سند بیشتر از این در نظر نخواهیم گرفت، اما مهم است که بدانیم آیا هدف شما راه حل‌های متعددی دارد یا خیر.

هم ارزی تحت یک جابجایی ثابت جمع یا ضرب در یک ثابت $\mathrm{a} \neq 0$ جواب را تغییر نمی‌دهد

$$\underset{\mathbf{w} \in \mathbb{R}^d}{\operatorname{argmin}}\, c(\boldsymbol{w}) = \underset{\mathbf{w} \in \mathbb{R}^d}{\operatorname{argmin}}\, a\, c(\boldsymbol{w}) = \underset{\mathbf{w} \in \mathbb{R}^d}{\operatorname{argmin}}\, c(\boldsymbol{w}) + a$$

با گرفتن گرادیان هر سه هدف و مشاهده صفر بودن گرادیان در شرایط یکسان می‌توانید دلیل آن را ببینید.

$$\nabla \mathrm{a}\, \mathrm{c}(\mathbf{w}) = 0 \Leftrightarrow \mathrm{a} \nabla \mathrm{c}(\mathbf{w}) = 0 \Leftrightarrow \nabla \mathrm{c}(\mathbf{w}) = 0$$

و

$$\nabla(\mathrm{c}(\mathbf{w}) + \mathrm{a}) = 0 \Leftrightarrow \nabla \mathrm{c}(\mathbf{w}) = 0$$

فصل ۳

اصول اولیه تخمین پارامتر

در مدل‌سازی احتمالی، معمولاً مجموعه‌ای از مشاهدات به ما ارائه می‌شود و هدف یافتن مدل یا تابعی است که مطابقت خوبی با داده‌ها نشان می‌دهد و الزامات اضافی خاصی را رعایت می‌کند. ما تقریباً این الزامات را به سه گروه دسته‌بندی می‌کنیم: (۱) توانایی تعمیم خوب، (۲) توانایی ترکیب دانش و فرضیات قبلی در مدل‌سازی، و (۳) مقیاس‌پذیری. اول، مدل باید بتواند در آزمون زمان مقاومت کند. یعنی عملکرد آن روی داده‌های دیده نشده قبلی نباید پس از ارائه این داده‌های جدید بدتر شود. گفته می‌شود مدل‌هایی با چنین عملکردی به خوبی تعمیم می‌یابند. دوم، $\hat{f}$ باید بتواند اطلاعات مربوط به فضای مدل $\mathcal{F}$ را که از آن انتخاب شده است، ترکیب کند و فرآیند انتخاب یک مدل باید بتواند "مشاوره" آموزشی را از یک تحلیلگر بپذیرد. در نهایت، زمانی که حجم زیادی از داده در دسترس است، الگوریتم‌های یادگیری باید بتوانند با توجه به منابعی مانند حافظه یا قدرت CPU، راه‌حل‌هایی را در زمان معقول ارائه دهند. به طور خلاصه، انتخاب یک مدل در نهایت به مشاهدات در دست، تجربه ما در مدل‌سازی پدیده‌های زندگی واقعی، و توانایی الگوریتم‌ها برای یافتن راه‌حل‌های خوب با توجه به منابع محدود بستگی دارد.

یک راه آسان برای فکر کردن در مورد یافتن "بهترین" مدل از طریق یادگیری پارامترهای یک توزیع است. فرض کنید به ما مجموعه‌ای از مشاهدات داده شده است $\mathcal{D} = \{x_i\}_{i=1}^{n}$، جایی که $x_i \in R$ و می‌دانیم که x_iها $i.i.d$[1] هستند. از توزیع گاوسی. در این مورد، مشکل یافتن بهترین مدل را می‌توان به عنوان یافتن بهترین پارامترهای μ^* و σ^* مشاهده کرد: مشکل را می‌توان به عنوان تخمین پارامتر مشاهده کرد. ما به این فرآیند، تخمین می‌گوییم زیرا فرض معمول این است که داده‌ها توسط یک مدل ناشناخته از $\mathcal{F}$ تولید شده است که پارامترهای آن را می‌خواهیم از داده‌ها بازیابی کنیم. ما تخمین پارامتر را با استفاده از تکنیک‌های احتمالی فرمول‌بندی می‌کنیم و متعاقباً راه‌حل‌هایی را از طریق بهینه‌سازی پیدا می‌کنیم، گاهی اوقات با محدودیت‌هایی در فضای پارامتر.

۳-۱ نقشه و برآورد ماکسیمم احتمال

تصور کنید مجموعه داده‌ای از مشاهدات را مشاهده می‌کنید $\mathcal{D} = \{x_i\}_{i=1}^{n}$. داده‌ها از مقداری توزیع واقعی p^* گرفته می‌شوند، اما این توزیع برای شما ناشناخته است. در عوض، تنها چیزی که می‌دانید این است که توزیع در مجموعه‌ای از توزیع‌های ممکن، $\mathcal{F}$ است که گاهی اوقات فضای فرضی یا کلاس تابع نامیده می‌شود. به عنوان مثال، $\mathcal{F}$ می‌تواند خانواده همه توزیع‌های گاوسی تک متغیره باشد:

$$\mathcal{F} = \{\mathcal{N}(\mu, \sigma^2) \mid \text{for any } \mu \in \mathbb{R} \text{ and } \sigma \in R^+\}$$

[1] Independent and identically distributed

توزیع واقعی دارای پارامترهای μ^* و σ^* است. با استفاده از داده‌ها، می‌خواهیم μ و σ را تا حد امکان به تابع هدف نزدیک کنیم. ایده پشت آن ماکسیمم تخمین پسینی[1] (MAP) یافتن محتمل‌ترین مدل برای داده‌های مشاهده شده است. با توجه به مجموعه داده‌های $\mathcal{D}$، راه‌حل MAP را اینگونه فرمول‌بندی می‌کنیم

$$f_{\text{MAP}} = \underset{\text{f} \in \mathcal{F}}{\text{argmax}}\ \text{p}(\text{f}|\mathcal{D})$$

که در آن $\text{p}(\text{f}|\mathcal{D})$ توزیع پسین مدل با توجه به داده‌ها نامیده می‌شود. در فضاهای مدل گسسته، $\text{p}(\text{f}|\mathcal{D})$ تابع جرم احتمال و تخمین MAP دقیقا محتمل‌ترین مدل است. همتای آن در فضاهای پیوسته مدلی است که بیشترین مقدار تابع چگالی پسین را دارد. توجه داشته باشید که ما از مدل کلمات که یک تابع است و پارامترهای آن که ضرائب آن تابع هستند تا حدودی به جای هم استفاده می‌کنیم. به عنوان مثال، در بالا، می‌توانیم به طور معادل $\mathcal{F} = \{\mu \in \mathbb{R}, \sigma \in \text{R}^+\}$ را در نظر بگیریم. ما معمولاً به‌جای اینکه به‌طور غیرمستقیم در مورد مدل‌ها یا احتمالاتی که آنها پارامتر را انتخاب می‌کنند، به طور مستقیم درباره فضای پارامتر یا فضای تابع استدلال می‌کنیم.

برای محاسبه توزیع پسین ما با اعمال قانون بیز شروع می‌کنیم

$$\text{p}(\text{f}|\mathcal{D}) = \frac{\text{p}(\mathcal{D}|\text{f})\text{p}(\text{f})}{\text{p}(\mathcal{D})} \qquad (3.1)$$

که در آن $p(\mathcal{D}|f)$ تابع درست‌نمایی نامیده می‌شود، $p(f)$ توزیع قبلی مدل و $p(\mathcal{D})$ توزیع حاشیه‌ای داده‌ها است. توجه داشته باشید که ما از $\mathcal{D}$ برای مجموعه داده‌های مشاهده شده استفاده می‌کنیم، اما معمولاً آن را به‌عنوان تحقق یک متغیر تصادفی چند بعدی $\mathcal{D}$ که بر اساس توزیع $p(\mathcal{D})$ ترسیم می‌شود، در نظر می‌گیریم. با استفاده از فرمول احتمال کل می‌توانیم $p(\mathcal{D})$ را به این صورت بیان کنیم.

$$\text{p}(\mathcal{D}) = \begin{cases} \sum_{\text{f}\in\mathcal{F}} \text{p}(\mathcal{D}|\text{f})\text{p}(\text{f}) & \text{f: discrete} \\ \int_{\mathcal{F}} \text{p}(\mathcal{D}|\text{f})\text{p}(\text{f})\text{df} & \text{f: continuous} \end{cases}$$

بنابراین، توزیع پسین را می‌توان به طور کامل با استفاده از احتمال و پیشین توصیف کرد. حوزه تحقیق و عملی که شامل روش‌های تعیین این توزیع و مدل‌های بهینه است، آمار استنباطی نامیده می‌شود.

یافتن f_{MAP} را می‌توان تا حد زیادی ساده کرد زیرا $p(\mathcal{D})$ در مخرج بر جواب تاثیر نمی‌گذارد. ما باید معادله (۳,۱) را دوباره بنویسیم

$$\text{p}(\text{f}|\mathcal{D}) = \frac{\text{p}(\mathcal{D}|\text{f})\text{p}(\text{f})}{\text{p}(\mathcal{D})}$$

$$\propto \text{p}(\mathcal{D}|\text{f}) \cdot \text{p}(\text{f})$$

که در آن $\propto$ نماد تناسب است. بنابراین، با حل مسئله بهینه سازی زیر می‌توانیم راه‌حل MAP را پیدا کنیم

$$f_{\text{MAP}} = \underset{\text{f} \in \mathcal{F}}{\text{argmax}}\ \text{p}(\mathcal{D}|\text{f})\text{p}(\text{f})$$

[1] maximum a posteriori

در برخی شرایط ممکن است دلیلی برای ترجیح یک مدل بر مدل دیگر نداشته باشیم و می‌توانیم $\mathrm{p(f)}$ را به عنوان یک ثابت نسبت به فضای مدل $\mathcal{F}$ در نظر بگیریم. سپس، MAP به ماکسیمم کردن تابع درست‌نمایی کاهش می‌دهد:

$$f_{MLE} = \underset{\mathrm{f} \in \mathcal{F}}{\operatorname{argmax}} \ \mathrm{p}(\mathcal{D}|\mathrm{f})$$

این راه‌حل را راه‌حل ماکسیمم احتمال[1] (MLE) می‌نامند. به طور رسمی، فرض ثابت بودن $p(f)$ مشکل‌ساز است زیرا یک توزیع یکنواخت را نمی‌توان همیشه تعریف کرد (مثلاً روی $\mathbb{R}$)، اگرچه راه‌حل‌هایی برای این موضوع با استفاده از پیشین‌های نامناسب وجود دارد. با این وجود، فکر کردن به MLE به عنوان یک مورد خاص از تخمین MAP مفید است.

مثال ۹: فرض کنید مجموعه داده $D = \{2, 5, 9, 5, 4, 8\}$ یک i.i.d است. نمونه‌ای از توزیع پواسون با پارامتر ثابت اما ناشناخته λ_0. تخمین ماکسیمم درست‌نمایی λ_0 را پیدا کنید.

تابع جرم احتمال توزیع پواسون به صورت $\mathrm{p}(x|\lambda) = \frac{\lambda^x \ \mathrm{e}^{-\lambda}}{x!}$ با پارامتر $\lambda \in \mathrm{R}^+$ بیان می‌شود. ما این پارامتر را به این صورت تخمین می‌زنیم

$$\lambda_{MLE} = \underset{\lambda \in (0,\infty)}{\operatorname{argmax}} \ \mathrm{p}(\mathcal{D}|\lambda) \qquad (3.2)$$

می‌توانیم تابع احتمال را به صورت زیر بنویسیم

$$\mathrm{p}(\mathcal{D}|\lambda) = \mathrm{p}(\{x_\mathrm{i}\}_{\mathrm{i}=1}^{\mathrm{n}}|\lambda)$$

$$\prod_{\mathrm{i}=1}^{n} \mathrm{p}(x_\mathrm{i}|\lambda)$$

که در آن احتمال به احتمالات فردی هر x_i تقسیم می‌شود زیرا داده‌ها $\mathrm{i.i.d}$ هستند. (متغیرهای تصادفی مستقل هستند). برای یافتن λ که احتمال را به ماکسیمم می‌رساند، ابتدا از یک لگاریتم (یک تابع یکنواخت) برای ساده کردن محاسبه استفاده می‌کنیم. سپس اولین مشتق آن را نسبت به λ پیدا کنید. و در نهایت آن را برابر با صفر کنید تا ماکسیمم را بدست آورید. به طور خاص، ما احتمال ورود به سیستم $\ln \mathrm{p}(\mathrm{D}|\lambda)$ را به این صورت بیان می‌کنیم

$$\ln \mathrm{p}(\mathcal{D}|\lambda) = \ln \prod_{\mathrm{i}=1}^{n} \mathrm{p}(x_\mathrm{i}|\lambda)$$

$$= \sum_{\mathrm{i}=1}^{n} \ln \mathrm{p}(x_\mathrm{i}|\lambda)$$

$$= \ln \lambda \sum_{\mathrm{i}=1}^{n} \mathrm{x_i} - \mathrm{n}\lambda - \sum_{\mathrm{i}=1}^{n} \ln (x_\mathrm{i}!)$$

زیرا

$$\ln \mathrm{p}(x_\mathrm{i}|\lambda) = \frac{\ln \lambda^{x_\mathrm{i}} \ \mathrm{e}^{-\lambda}}{(x_\mathrm{i})!}$$

[1] maximum likelihood

$$= \ln \lambda^{x_i} + \ln e^{-\lambda} - \ln x_i !$$

$$= x_i \ln \lambda - \lambda - \ln x_i !$$

اکنون در این فرم، ساده‌تر به محاسبه مشتق می‌پردازیم

$$\frac{\partial \ln p(\mathcal{D}|\lambda)}{\partial \lambda} = \frac{1}{\lambda}\sum_{i=1}^{n} x_i - n$$

حل λ به گونه‌ای که $\frac{\partial \ln p(D|\lambda)}{\partial \lambda} = 0$ یک نقطه ثابت از این مسئله را به ما می‌دهد، و ما $\lambda_{MLE} = \frac{1}{n}\sum_{i=1}^{n} x_i$ را دریافت می‌کنیم. می‌توانیم $n = 6$ و مقادیر $\mathcal{D}$ را جایگزین کنیم تا جواب را به این صورت محاسبه کنیم

$$\lambda_{MLE} = \frac{1}{n}\sum_{i=1}^{n} x_i = 5.5$$

که به سادگی یک میانگین نمونه است.

مشتق دوم این احتمال log همیشه منفی است زیرا λ باید مثبت باشد. مشتق دوم $-\lambda^{-2}\sum_{i=1}^{n} x_i$ است که برای این λ_{MLE} که $0 <$ است. بنابراین، عبارت قبلی در واقع احتمال را به مقدار ماکسیمم می‌رساند. توجه داشته باشید که برای به ماکسیمم رساندن درست این هزینه‌ها، ما همچنین باید از اعمال محدودیت $\lambda \in (0, \infty)$ اطمینان حاصل کنیم. از آنجایی که راه‌حل بالا در مجموعه محدودیت‌ها قرار دارد، می‌دانیم که جواب صحیح معادله (۳,۲) را داریم. با این حال، در موقعیت‌های دیگر، همانطور که بعداً بحث خواهیم کرد، باید به صراحت محدودیت‌هایی را در بهینه‌سازی اعمال کنیم.

مثال ۱۰: فرض کنید $\mathcal{D} = \{2, 5, 9, 5, 4, 8\}$ دوباره یک i. i. d باشد. نمونه‌ای از پواسون (λ_0)، اما اکنون اطلاعات اضافی نیز به ما داده می‌شود. فرض کنید دانش قبلی در مورد λ_0 را می‌توان با استفاده از توزیع گاما با پارامترهای $k = 3$ و $\theta = 1$ بیان کرد. پیدا کردن λ_0 برای تخمین MAP را بیابید.

ابتدا تابع چگالی احتمال توزیع گاما را برای قبلی خود می‌نویسیم

$$p(\lambda) = \frac{\lambda^{k-1} e^{-\frac{\lambda}{\theta}}}{\theta^k \Gamma(k)}$$

که برای $\lambda > 0$ ، $\Gamma(k)$ تابع گامایی است که تابع فاکتوریل را تعمیم می‌دهد. وقتی k یک عدد صحیح است، $\Gamma(k) = (k-1)!$ داریم. تخمین MAP پارامترها را می‌توان به صورت زیر پیدا کرد

$$\lambda_{MAP} = \underset{\lambda \in (0,\infty)}{\operatorname{argmax}} p(\mathcal{D}|\lambda)p(\lambda)$$

مانند قبل، ما log را برای ساده کردن محاسبات می‌گیریم تا به دست آوریم

$$\ln p(D|\lambda)p(\lambda) = \ln p(D|\lambda) + \ln p(\lambda)$$

$$= \sum_{i=1}^{n} \ln p(x_i|\lambda) + \ln p(\lambda).$$

ما قبلاً عبارت اول را در مثال قبلی ساده کرده‌ایم. برای ورود به سیستم توزیع قبلی، ما داریم

$$\ln p(\lambda) = \ln(\lambda^{k-1} e^{-\frac{\lambda}{\theta}}) - \ln(\theta^k \Gamma(k))$$

$$= (k-1)\ln\lambda - \frac{\lambda}{\theta} - \ln(\theta^k \Gamma(k)).$$

جمله آخر با توجه به λ ثابت است. بنابراین وقتی مشتق را می‌گیریم ناپدید می‌شود و می‌توانیم از محاسبه آن اجتناب کنیم. دوباره همه چیز را در جای خود قرار میدهیم

$$\ln p(D|\lambda)p(\lambda) = \ln\lambda \sum_{i=1}^{n} xi - n\lambda - \sum_{i=1}^{n} \ln(x_i!) + (k-1)\ln\lambda - \frac{\lambda}{\theta} - \ln(\theta^k \Gamma(k))$$

و گرفتن مشتق آن می‌دهد

$$\frac{\partial \ln p(D|\lambda)p(\lambda)}{\partial\lambda} = \frac{1}{\lambda}\sum_{i=1}^{n} x_i - n + \frac{k-1}{\lambda} - \frac{1}{\theta}$$

زیرا

$$\frac{\partial \ln p(\lambda)}{\partial\lambda} = \frac{k-1}{\lambda} - \frac{1}{\theta}$$

بار دیگر با صفر کردن مشتق و حل λ به دست می‌آید

$$\lambda_{MAP} = \frac{k - 1 + \sum_{i=1}^{n} x_i}{n + \frac{1}{\theta}} = 5$$

برای مجموعه داده $\mathcal{D}$

نگاهی گذرا به λ_{MAP} و λ_{MLE} نشان می‌دهد که با رشد n، هم اعداد و هم مخرج‌ها در عبارات بالا به طور فزاینده‌ای شبیه‌تر می‌شوند. در واقع، این یک نتیجه شناخته شده است که در حد نمونه‌های نامتناهی، هر دو MAP و MLE به یک مدل، f، همگرا می‌شوند، تا زمانی که پیشین احتمال (یا چگالی) روی f را نداشته باشد. این نتیجه نشان می‌دهد که برآورد MAP به راه‌حل MLE برای مجموعه داده‌های بزرگ نزدیک می‌شود. به عبارت دیگر، داده‌های بزرگ از اهمیت دانش قبلی می‌کاهد. این یک نتیجه‌گیری مهم است زیرا دستگاه‌های ریاضی لازم برای استنتاج عملی را ساده می‌کند.

برای بدست آوردن مقداری شهود برای این نتیجه، نشان خواهیم داد که برآوردهای MAP و MLE برای مثال بالا با توزیع پواسون به یک راه‌حل همگرا می‌شوند. فرض کنید $s_n = \sum_{i=1}^{n} x_i$، که نمونه‌ای از متغیر تصادفی $S_n = \sum_{i=1}^{n} X_i$ است. اگر $\lim_{n\to\infty} \frac{s_n}{n^2} = 0$ (یعنی s_n سریعتر از n^2 رشد نمی‌کند)،آنگاه

$$|\lambda_{MAP} - \lambda_{MLE}| = \left|\frac{k - 1 + s_n}{n + \frac{1}{\theta}} - \frac{s_n}{n}\right|$$

$$= \left|\frac{k-1}{n + \frac{1}{\theta}} - \frac{s_n}{n\left(n + \frac{1}{\theta}\right)}\right|$$

$$\leq \frac{|k-1|}{n+\frac{1}{\theta}} + \frac{s_n}{n\left(n+\frac{1}{\theta}\right)} \xrightarrow[n\to\infty]{} 0$$

توجه داشته باشید که اگر $\lim_{n\to\infty} \frac{s_n}{n^2} \neq 0$، هر دو برآوردگر به ∞ می‌روند. با این حال، چنین دنباله‌ای از مقادیر احتمال وقوع اساساً صفر است. قضایای سازگاری برای تخمین MLE و MAP بیان می‌کنند که همگرایی با پارامترهای واقعی "قریب به یقین" یا "با احتمال 1" رخ می‌دهد تا نشان‌دهد که این توالی‌های نامحدود مجموعه‌ای از اندازه‌گیری-صفر را تحت شرایط معقول معینی تشکیل می‌دهند (برای اطلاعات بیشتر ببینید: [۱۹ قضیه ۹,۱۳]).

مثال ۱۱: فرض کنید $\mathcal{D} = \{x_i\}_{i=1}^n$ یک $i.i.d$ باشد. نمونه از یک توزیع گاوسی تک متغیره. هدف ما یافتن برآوردهای ماکسیمم احتمال پارامترها است. با تشکیل تابع $log - likelihood$ شروع می‌کنیم

$$\ln p(\mathcal{D}|\mu,\sigma) = \ln \prod_{i=1}^{n} p(x_i|\mu,\sigma)$$

$$n\, ln\frac{1}{\sqrt{2\pi}} + n\ln\frac{1}{\sigma} - \frac{\sum_{i=1}^{n}(x_i-\mu)^2}{2\sigma^2}$$

ما مشتقات جزئی $\log - \text{lihood}$ را با توجه به تمام پارامترها محاسبه می‌کنیم

$$\frac{\partial}{\partial\mu} \ln p(\mathcal{D}|\mu,\sigma) = \frac{\sum_{i=1}^{n}(x_i-\mu)}{\sigma^2}$$

و

$$\frac{\partial}{\partial\sigma} \ln p(\mathcal{D}|\mu,\sigma) = -\frac{n}{\sigma} + \frac{\sum_{i=1}^{n}(x_i-\mu)^2}{\sigma^3}$$

از اینجا می‌توانیم استخراج کنیم:

$$\mu_{MLE} = \frac{1}{n}\sum_{i=1}^{n} x_i$$

و

$$\sigma^2_{MLE} = \frac{1}{n}\sum_{i=1}^{n}\left(x_i-\mu_{MLE}\right)^2$$

برآوردهای MAP و MLE را تخمین نقطه‌ای می‌نامند. این تخمین‌ها با تخمین‌های بیزی، که کل توزیع پسین یا فواصل اطمینان را برای پارامترها تخمین می‌زنند، در تضاد هستند. ما در این کتاب اساساً بر تخمین‌های نقطه‌ای تمرکز می‌کنیم و تنها به اختصار رویکردهای بیزی را تقریبا تا پایان بررسی می‌کنیم.

۳-۲ ماکسیمم احتمال برای توزیع‌های شرطی

همچنین می‌توانیم مشکلات ماکسیمم احتمال را برای توزیع‌های شرطی فرموله کنیم. به یاد بیاورید که یک توزیع شرطی به شکل $p(y|x)$ برای دو متغیر تصادفی Y و X است، که در بالا توزیع حاشیه‌ای $p(x)$ یا $p(y)$ را در نظر گرفتیم. برای توزیع‌های بالا، پرسیدیم: توزیع روی این متغیر چگونه است؟ برای توزیع شرطی، در عوض می‌پرسیم: با توجه به برخی اطلاعات کمکی، اکنون توزیع روی این متغیر چگونه است؟ هنگامی که اطلاعات کمکی تغییر می‌کند، توزیع روی متغیر نیز تغییر می‌کند. برای مثال، ممکن است بخواهیم توزیع را بر فروش یک محصول خاص (Y) با توجه به ماه جاری (X) شرط کنیم. ما انتظار داریم که توزیع بر روی Y بسته به ماه متفاوت باشد.

توزیع‌های شرطی می‌توانند از هر یک از خانواده‌های توزیع مورد بحث در بالا باشند، و ما می‌توانیم به طور مشابه مسائل تخمین پارامتر را فرموله کنیم. با این حال، پارامترها معمولاً به متغیر X مرتبط هستند. ما یک مثال ساده برای نشان دادن این موضوع در زیر ارائه می‌دهیم. بسیاری از فرمول‌های تخمین پارامتری که در ادامه کتاب در نظر می‌گیریم، برای توزیع‌های شرطی هستند، زیرا در یادگیری ماشین معمولاً تعداد زیادی متغیر کمکی (ویژگی‌ها) داریم و سعی می‌کنیم تا اهداف را پیش‌بینی کنیم (یا نوع توزیع را یاد بگیریم). در فصل‌های مربوط به رگرسیون و طبقه‌بندی، نشان خواهیم داد که چند مدل می‌توانند به عنوان ماکسیمم احتمال برای توزیع‌های شرطی $p(y|x)$ فرموله شوند.

مثال ۱۲: فرض کنید به شما دو متغیر تصادفی X و Y داده شده است و معتقدید $p(y|x) = N(\mu = x, \sigma^2)$ برای برخی σ ناشناخته. هدف ما تخمین این پارامتر ناشناخته σ است. توجه داشته باشید که توزیع بر روی Y متفاوت است، بسته به اینکه کدام مقدار X مشاهده یا داده شود.

ما دوباره با تشکیل تابع $\log - \text{likelihood}$ شروع می‌کنیم، اکنون برای جفت n نمونه $\mathcal{D} = (x_1, y_1), \ldots, (x_n, y_n)$. ما از قانون زنجیره استفاده خواهیم کرد: $p(x_i, y_i) = p(y_i|x_i)p(x_i)$

$$\ln p(\mathcal{D}|\sigma) = \ln \prod_{i=1}^{n} p(x_i, y_i|\sigma)$$

$$= \ln \prod_{i=1}^{n} p(y_i|x_i, \sigma)p(x_i)$$

$$\sum_{i=1}^{n} \ln p(y_i|x_i, \sigma) + \ln p(x_i)$$

$$\sum_{i=1}^{n} \ln \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(y_i - x_i)^2}{2\sigma^2}\right) + \ln p(x_i)$$

$$= n \ln \frac{1}{\sqrt{2\pi}} + n \ln \frac{1}{\sigma} - \frac{\sum_{i=1}^{n}(y_i - x_i)^2}{2\sigma^2} + \sum_{i=1}^{n} \ln p(x_i)$$

توجه داشته باشید که ما از $\mu = x_i$ برای هر توزیع نرمال $p(y_i|x_i, \sigma)$ استفاده می‌کنیم. اکنون مشتقات جزئی $\log - \text{likelihood}$ را با توجه به پارامتر σ محاسبه می‌کنیم

$$\frac{\partial}{\partial\sigma}\ln p(\mathcal{D}|\sigma) = -\frac{n}{\sigma} + \frac{\sum_{i=1}^{n}(y_i - x_i)^2}{\sigma^3}$$

توجه کنید که $\frac{\partial}{\partial\sigma}\sum_{i=1}^{n}\ln p(x_i) = 0$، زیرا σ احتمال $p(x_i)$ پارامتری نمی‌کند. بنابراین برای بدست آوردن σ بهینه، نیازی به دانستن یا تعیین توزیع بر روی متغیر تصادفی X نیست. با قرار دادن مشتق بر روی صفر، برای به دست آوردن یک نقطه ثابت، به دست می‌آوریم.

$$\sigma^2_{MLE} = \frac{1}{n}\sum_{i=1}^{n}(y_i - x_i)^2$$

۳-۳ [پیشرفته] رابطه بین به ماکسیمم رساندن احتمال و واگرایی Kullback – Leibler

ما اکنون رابطه بین تخمین ماکسیمم درست‌نمایی و واگرایی کول بک لایبلر را بررسی می‌کنیم. واگرایی Kullback – Leibler بین دو توزیع احتمال $p(x)$ و $q(x)$ در $X = R$ به این صورت تعریف شده است.

$$D_{KL}(p||q) = \int_{-\infty}^{\infty}(x)\log\frac{p(x)}{q(x)}\,dx$$

در تئوری اطلاعات، واگرایی Kullback – Leibler تفسیری طبیعی از بازده فشرده‌سازی سیگنال دارد، زمانی که کد با استفاده از توزیع غیربهینه $q(x)$ به جای توزیع صحیح (اما ناشناخته) $p(x)$ ساخته می‌شود. داده تولید شده است. با این حال، اغلب، واگرایی کولبک-لایبلر به سادگی به عنوان معیاری از واگرایی بین دو توزیع احتمال در نظر گرفته می‌شود. اگرچه این واگرایی یک متریک نیست (متقارن نیست و نابرابری مثلث را برآورده نمی‌کند) اما دارای ویژگی‌های نظری مهمی است که (۱) همیشه غیر منفی است و (۲) برابر با صفر است اگر و تنها اگر $p(x) = q(x)$

اکنون یک واگرایی بین یک توزیع احتمال تخمینی $p(x|\theta)$ و یک توزیع اساسی (درست) $p(x|\theta_0)$ را در نظر بگیرید که بر اساس آن مجموعه داده $\mathcal{D} = \{x_i\}_{i=1}^{n}$ ایجاد شد. واگرایی کولبک-لایبلر[1] بین $p(x|\theta)$ و $p(x|\theta_0)$ است.

$$D_{KL}(p(x|\theta_0)||p(x|\theta)) = \int_{-\infty}^{\infty}p(x|\theta_0)\log\frac{p(x|\theta_0)}{p(x|\theta)}\,dx$$

$$= \int_{-\infty}^{\infty}p(x|\theta_0)\log\frac{1}{p(x|\theta)}\,dx - \int_{-\infty}^{\infty}p(x|\theta_0)\log\frac{1}{p(x|\theta_0)}\,dx$$

عبارت دوم در معادله فوق صرفاً آنتروپی (دیفرانسیل) توزیع واقعی است و تحت تأثیر انتخاب ما از مدل θ نیست. از طرف دیگر اصطلاح اول را می‌توان به این صورت بیان کرد

$$\int_{-\infty}^{\infty}p(x|\theta_0)\log\frac{1}{p(x|\theta)}\,dx = \mathbb{E}[log\ p(X|\theta)]$$

بنابراین، به مقدار ماکسیمم رساندن $\mathbb{E}[\log p(X|\theta)$ واگرایی کولبک-لایبلر بین $p(x|\theta)$ و $p(x|\theta_0)$ را به مقدار مینیمم می‌رساند. با استفاده از قانون قوی اعداد بزرگ، این را می‌دانیم که

[1] Kullback-Leibler (KL)

$$\frac{1}{\mathrm{n}}\sum_{\mathrm{i}=1}^{\mathrm{n}}\log \mathrm{p}(x_{\mathrm{i}}\,|\theta)\ \xrightarrow{\text{a.s.}}\ \mathbb{E}[\log \mathrm{p}(\mathrm{X}|\theta)]$$

وقتی $n \to \infty$. بنابراین، زمانی که مجموعه داده به اندازه کافی بزرگ باشد، به ماکسیمم رساندن تابع احتمال، واگرایی $Kullback - Leibler$ را به مینیمم می‌رساند و به این نتیجه می‌رسد که $p(x|\theta_{MLE}) = p(x|\theta_0)$، اگر مفروضات اساسی برآورده شوند. تحت شرایط معقول، می‌توانیم از آن استنباط کنیم که $\theta_{MLE} = \theta_0$. این برای خانواده‌هایی از توزیع‌ها که مجموعه‌ای از پارامترها به طور منحصربه‌فرد توزیع احتمال را تعیین می‌کنند، صادق است. به عنوان مثال، به طور کلی برای مخلوطی از توزیع‌ها صدق نمی‌کند، اما ما بعداً در مورد این وضعیت بحث خواهیم کرد. این نتیجه تنها یکی از بسیاری از ارتباطات بین آمار و نظریه اطلاعات است.

فصل ۴

مقدمه‌ای بر مسائل پیش‌بینی

یادگیری ماشین به بسیاری از مسائل پاسخ می‌دهد، که گاهی اوقات ممکن است باعث ایجاد احساس مشکل شوند. به عنوان یک فهرست غیر جامع، این موارد شامل یادگیری تحت نظارت (با طبقه‌بندی[1] و رگرسیون[2]) می‌شود. یادگیری نیمه نظارتی؛ یادگیری بدون نظارت؛ تکمیل یادگیری فاقد دسته‌بندی. پیش‌بینی ساختار یافته؛ یادگیری رتبه‌بندی؛ یادگیری رابطه‌ای آماری; یادگیری فعال؛ و پیش‌بینی زمانی (با پیش‌بینی سری‌های زمانی و ارزیابی خط‌مشی در یادگیری تقویتی و یادگیری آنلاین). برای برخی از این مجموعه‌ها، مانند یادگیری فعال و یادگیری تقویتی، جمع‌آوری داده‌ها بخش مرکزی الگوریتم است و می‌تواند به طور قابل توجهی کیفیت مدل‌های پیش‌بینی آموخته شده را تعیین کند. اکثر مجموعه‌های دیگر فرض می‌کنند که داده‌ها جمع‌آوری شده‌اند بدون توانایی ما برای تأثیرگذاری بر آن مجموعه - و اکنون ما فقط باید آن داده‌ها را تجزیه و تحلیل کنیم و بهترین پیش‌بینی‌کننده‌ها را یاد بگیریم. در این مجموعه غیرفعال، می‌توانیم فرض کنیم که داده‌ها $\mathrm{i.i.d.}$ هستند - که رایج‌ترین حالت است - یا وابستگی‌هایی بین نقاط داده وجود دارد - مانند پیش‌بینی سری‌های زمانی یا یادگیری رابطه‌ای آماری. همچنین زمینه‌هایی وجود دارد که داده‌ها ناقص هستند، مثلاً به این دلیل که کاربر سن خود را پر نکرده است

بنابراین، یک هستی‌شناسی می‌تواند ابعاد زیر را برای دسته‌بندی مسائل یادگیری ماشین در نظر بگیرد:

۱. منفعل در مقابل فعال
۲. $i.i.d.$ در مقابل $non-i.i.d$
۳. کامل در مقابل ناقص

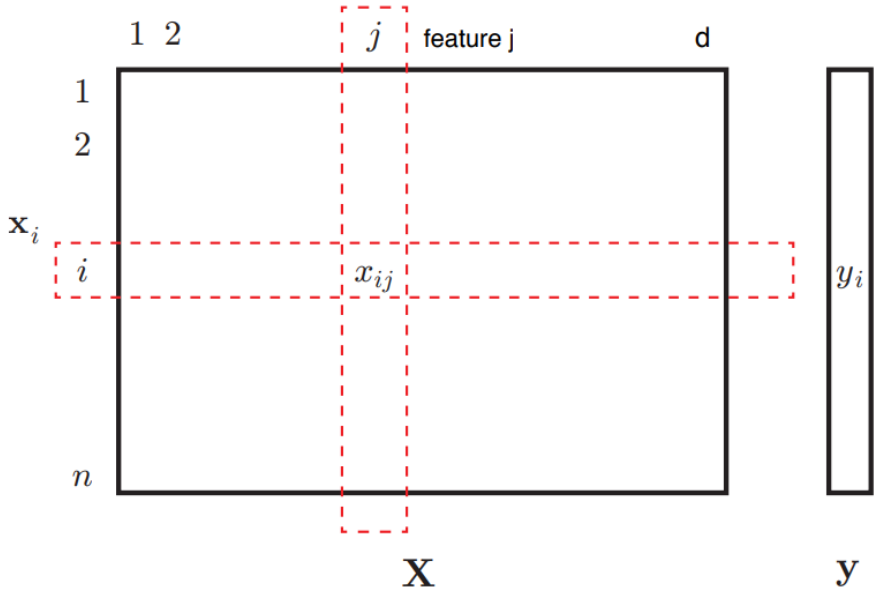
مانند همه هستی‌شناسی‌ها، هر مسئله کاملاً در این دسته‌بندی‌ها قرار نمی‌گیرد. علاوه بر این، این احتمال وجود دارد که اکثر جمع‌آوری داده‌ها کاملاً غیرفعال نباشند (حتی اگر فقط به این دلیل که مدل‌ساز انسانی بر جمع‌آوری داده‌ها تأثیر می‌گذارد)، احتمالاً $i.i.d.$ (حتی اگر قصد داشتیم باشد)، و احتمالاً برخی از اجزای گم شده را دارد. با این وجود، الگوریتم‌ها این مفروضات را به درجات مختلف انجام می‌دهند، حتی اگر داده‌ها آن مفروضات را برآورده نکنند. برای اکثر این یادداشت‌ها، ما روی ساده‌ترین مجموعه‌ها تمرکز می‌کنیم: منفعل، $i.i.d.$ و کامل در این فصل ابتدا طبقه‌بندی و رگرسیون را معرفی می‌کنیم و سپس معیارهای انتخاب توابع برای طبقه‌بندی و رگرسیون را مورد بحث قرار می‌دهیم تا الگوریتم‌های توسعه‌یافته در فصل‌های بعدی را ایجاد کنیم.

---

[1] Classification
[2] Regression

۴-۱ مسائل یادگیری تحت نظارت

ما با تعریف یک مجموعه داده شروع می‌کنیم $\mathcal{D} = \{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \ldots, (\mathbf{x}_n, y_n)\}$ جایی که $\mathbf{x}_i \in \mathcal{X}$ ورودی یا مشاهدات i-ام و $y_i \in Y$ هدف ما است. معمولاً فرض می‌کنیم که $\mathcal{X} = R^d$، در این صورت $\boldsymbol{x}_i = (x_{i1}, x_{i2}, \ldots\ldots, x_{id})$ یک بردار d بعدی است که یک نمونه نامیده می‌شود.



شکل ۴٫۱: علامت گذاری برای مجموعه داده. $\boldsymbol{X}$ یک ماتریس n در d است که ردیف‌هایی مربوط به نمونه‌ها و ستون‌های مربوط به ویژگی‌ها است. y یک بردار n در 1 از اهداف است.

یا یک نمونه هر بعد از x_j معمولاً یک ویژگی نامیده می‌شود. ما اغلب مجموعه داده را در یک ماتریس $X \in R^{n \times d}$ سازماندهی می‌کنیم که در آن هر ردیف با یک نمونه x_i و هر ستون مربوط به یک ویژگی است (شکل ۴٫۱ را ببینید).

تمایز بین $\mathbf{x}$ و y به این دلیل است که فرض می‌کنیم ویژگی‌ها برای هر شی نسبتاً آسان جمع‌آوری می‌شوند (مثلاً با اندازه‌گیری قد یک فرد یا فوت مربع یک خانه)، در حالی که مشاهده یا جمع‌آوری متغیر هدف دشوار یا هزینه بر است. (مثلاً وجود بیماری یا قیمت نهایی فروش خانه قبل از فروش آن). چنین موقعیت‌هایی معمولاً از ساخت یک مدل محاسباتی استفاده می‌کنند که اهداف را از روی مجموعه‌ای از مقادیر ورودی پیش‌بینی می‌کند. این مدل با استفاده از مجموعه‌ای از مشاهدات ورودی که مقادیر هدف قبلاً جمع‌آوری شده‌اند، آموزش داده می‌شود. در استقرار، می‌توانیم از این مدل برای اهداف پیش‌بینی‌هایی از اطلاعات به‌دست‌آمده-مشاهده-در مورد اطلاعاتی که به سختی به دست می‌آیند- استفاده کنیم.

۴-۱-۱ رگرسیون و طبقه‌بندی

تفاوت در الگوریتم‌ها برای مسائل پیش‌بینی، با داده‌های کامل $i.i.d.$، معمولاً از ویژگی‌های ورودی‌ها (مشاهدات) و ویژگی‌های اهداف ناشی می‌شوند. به عنوان مثال، ممکن است لازم باشد مشاهدات متنی - مانند مشاهدات مجموعه‌ای از اسناد - را متفاوت

از یک بردار مشاهداتی با ارزش واقعی ده بعدی از خوانش‌های حسگر که دما و فشار را در یک سیستم فیزیکی منعکس می‌کند، بررسی کنیم. یک استراتژی ساده و نسبتاً رایج برای رسیدگی به این تفاوت‌ها، ترسیم انواع مختلف مشاهدات - زبان، متغیرهای طبقه‌بندی و حتی داده‌های دنباله‌ای - در فضای اقلیدسی است که در آن مشاهدات دوباره به عنوان یک بردار با ارزش واقعی نشان داده می‌شود. بسیاری از الگوریتم‌های پیش‌بینی برای مشاهدات با ارزش واقعی طراحی شده‌اند، بنابراین الگوریتم‌های استاندارد را می‌توان اعمال کرد. این مسئله نمایش داده‌ها به خودی خود یک مشکل اساسی و دشوار است. در فصل ۹ بیشتر درباره آن بحث خواهیم کرد. در حال حاضر، فرض می‌کنیم مشاهدات از قبل به شکل مناسبی هستند، به عنوان یک بردار با ارزش واقعی d بعدی.

ویژگی‌های هدف نیز مهم هستند و منجر به دو تمایز معمولی برای مسائل پیش‌بینی می‌شوند: طبقه‌بندی و رگرسیون. به طور کلی، وقتی $\mathcal{Y}$ پیوسته است، یک مسئله از نوع رگرسیون داریم و اگر $\mathcal{Y}$ گسسته باشد، یک مسئله از نوع طبقه‌بندی. در رگرسیون مجموعه هدف ممکن، شامل $\mathcal{Y} = \mathbb{R}$ یا $\mathcal{Y} = [0, \infty)$ است. نمونه‌ای از مسئله رگرسیون در جدول ۴،۱ نشان داده شده است.

	size [sqft]	age [yr]	dist [mi]	inc [$]	dens [ppl/mi^2]	y
$\mathbf{x}_1$	1250	5	2.85	56,650	12.5	2.35
$\mathbf{x}_2$	3200	9	8.21	245,800	3.1	3.95
$\mathbf{x}_3$	825	12	0.34	61,050	112.5	5.10

جدول ۴،۱: مثالی از مسئله رگرسیون: پیش‌بینی قیمت یک خانه در یک منطقه خاص. در اینجا، ویژگی‌ها نشان دهنده اندازه خانه $(size)$ بر حسب فوت مربع، قدمت خانه (age) بر حسب سال، فاصله از مرکز شهر (dist) بر حسب مایل، متوسط درآمد در شعاع یک مایل مربع (inc) در صد هزار دلار ، و تراکم جمعیت در همان منطقه (dens). هدف نشان دهنده قیمتی است که یک خانه به آن قیمت فروخته می‌شود، به عنوان مثال.

در طبقه‌بندی، تابعی می‌سازیم که برچسب‌های کلاس گسسته را پیش‌بینی می‌کند. این تابع معمولاً طبقه‌بندی کننده[1] نامیده می‌شود. کاردینالیته $\mathcal{Y}$ در مسائل طبقه‌بندی معمولاً کوچک است، به عنوان مثال. $\mathcal{Y} = \{\text{healthy, diseased}\}$ نمونه‌ای از یک مجموعه داده برای طبقه‌بندی با تعداد $n = 3$ داده و $d = 5$ ویژگی در جدول ۴،۲ نشان داده شده است.

مسائل طبقه‌بندی را می‌توان بیشتر به مسائل چند کلاسه[2] و چند برچسبی تقسیم کرد. یک مسئله چند کلاسه شامل ارائه برچسب واحد برای یک ورودی است. به عنوان مثال، برای گروه خونی (ساده) با $Y = \{A, B, AB, O\}$، بیمار فقط می‌تواند با یکی از این برچسب‌ها برچسب گذاری شود. در مسائل چند کلاسه، اگر فقط دو کلاس وجود داشته باشد، طبقه‌بندی باینری نامیده می‌شود، مانند مثال در جدول ۴،۲. در چند برچسب، یک ورودی را می‌توان با بیش از یک برچسب مرتبط کرد. نمونه‌ای از مسئله چند برچسبی، طبقه‌بندی اسناد متنی به دسته‌هایی مانند {ورزش، پزشکی، مسافرت، سیاست} است. در اینجا، یک نمونه واحد ممکن است به بیش از یک مقدار در مجموعه مرتبط باشد. به عنوان مثال، مقاله‌ای در مورد پزشکی ورزشی تابع آموخته شده اکنون می‌تواند چندین خروجی را برگرداند.

[1] classifier
[2] classifier

به طور معمول، برای سازگاری بیشتر خروجی‌ها بین این دو مجموعه، خروجی برای چند کلاس و چند برچسب یک بردار نشانگر است. برای $m = |Y|$، پیش‌بینی گروه‌های خونی ممکن است $[0\ 1\ 0\ 0]$ باشد تا نشان‌دهنده گروه خونی B باشد و پیش‌بینی چهار برچسب مقاله می‌تواند $[1\ 1\ 0\ 0]$ باشد اگر هم مقاله‌ای مربوط به ورزش و هم مربوط به پزشکی باشد.

همانطور که با مشاهده، اهداف ممکن است خود پیچیده باشند، مانند اهداف متنی. یکی از حوزه‌هایی که با اهداف پیچیده‌تر سروکار دارد، پیش‌بینی خروجی ساختاریافته است، جایی که y می‌تواند مجموعه‌ای از خروجی‌های ساختاریافته باشد، به عنوان مثال. رشته‌ها، درختان یا نمودارها. کاردینالیته فضای خروجی در مسائل یادگیری ساختار یافته اغلب بسیار زیاد است. برای مثال، هنگام پیش‌بینی عملکرد یک پروتئین، کل درخت هستی‌شناسی باید پیش‌بینی شود، زیرا عملکرد خاصی زیرمجموعه‌ای از عملکردهای دیگر است. همانند مشاهدات، ممکن است بتوانیم بازنمودهای ساده‌تری را برای این اهداف پیدا کنیم تا روش‌های استانداردتری از رگرسیون و طبقه‌بندی را اعمال کنیم. مجدداً، در فصل ۹ بیشتر در این مورد بحث خواهیم کرد. در حال حاضر، اهداف نسبتاً ساده‌ای را فرض می‌کنیم، که بردارهای با ارزش واقعی m بعدی یا تعداد نسبتاً کمی از نتایج گسسته هستند.

	wt [kg]	ht [m]	T [°C]	sbp [mmHg]	dbp [mmHg]	y
$\mathbf{x}_1$	91	1.85	36.6	121	75	-1
$\mathbf{x}_2$	75	1.80	37.4	128	85	$+1$
$\mathbf{x}_3$	54	1.56	36.6	110	62	-1

جدول ۴/۲: مثالی از یک مسئله طبقه‌بندی دوتایی: پیش‌بینی وضعیت بیماری برای یک بیمار. در اینجا، ویژگی‌ها وزن (wt)، قد (ht)، دما (T)، فشار خون سیستولیک (sbp) و فشار خون دیاستولیک (dbp) را نشان می‌دهند. برچسب‌های کلاس به عنوان وجود یک بیماری خاص را نشان می‌دهد،

۲-۱-۴ تصمیم گیری در مورد نحوه فرمول‌بندی کردن مسئله

اگرچه ما مسائل یادگیری تحت نظارت را به دو دسته تقسیم می‌کنیم، اما همیشه مشخص نیست که چگونه یک مسئله باید فرمول‌بندی شود. به عنوان مثال، فضای خروجی $Y = \{0, 1, 2\}$ را در نظر بگیرید. می‌توانیم این مسئله را به‌عنوان یک مسئله طبقه‌بندی چند کلاسه در نظر بگیریم، یا می‌توانیم $Y = [0, 2]$ را فرض کنیم و یک مدل رگرسیون را یاد بگیریم. سپس می‌توانیم پیش‌بینی‌های برگردانده‌شده توسط مدل رگرسیون را با گرد کردن آن‌ها به نزدیک‌ترین عدد صحیح، آستانه‌ای کنیم.

چگونه تصمیم می‌گیرید که از کدام فرمول برای حل مسئله استفاده کنید؟ اگرچه رویه‌های ریاضی در یادگیری ماشین دقیق هستند، تصمیم‌گیری در مورد چگونگی فرمول‌بندی مسئله دنیای واقعی ظریف است و بنابراین ذاتاً واضح نیست. انتخاب یک روش خاص برای مدل‌سازی به تحلیلگر و دانش او از حوزه و همچنین جنبه‌های فنی یادگیری بستگی دارد. در این مثال، می‌توانید بپرسید: آیا به طور ذاتی ترتیبی برای خروجی‌های $\{0, 1, 2\}$ وجود دارد؟ اگر اینطور نیست، بگوییم که آنها سیب را ترجیح می‌دهند (در اولویت قرار میدهند)، پرتقال را ترجیح می‌دهند، یا موز را ترجیح می‌دهند مطابقت دارند، در این صورت ممکن است مدل کردن خروجی به عنوان یک بازه، که اغلب به سفارش دادن اشاره دارد، انتخاب ضعیفی باشد. از سوی دیگر، یادگیری توابع رگرسیون آسان‌تر است و اغلب پیش‌بینی‌های طبقه‌بندی به‌طور شگفت‌آور خوبی تولید می‌کنند. به‌علاوه، اگر برای این کلاس‌ها ، ترتیبی وجود داشته باشد ، مثلاً خوب، بهتر، بهترین آنگاه اکثر مدل‌های طبقه‌بندی - که ترتیبی را در خروجی‌ها فرض نمی‌کنند - نمی‌توانند از این ترتیب برای بهبود عملکرد پیش‌بینی استفاده کنند.

فرمول‌بندی کردن مسئله و انتخاب کلاس تابع و هدف گام مهمی در استفاده موثر از یادگیری ماشین است. خوشبختانه، دانش زیادی وجود دارد، به ویژه از نظر تجربی، که می‌تواند این انتخاب را هدایت کند. با یادگیری بیشتر در مورد روش‌ها، همراه با برخی اطلاعات در مورد ساختار دامنه خود، در تشخیص بهتر خواهد شد.

۴-۲ یادگیری بدون نظارت و یادگیری نیمه نظارت

مجموعه داده‌ها همیشه کامل نیستند: در برخی موارد، ما فقط می‌توانیم برچسب‌هایی را برای زیرمجموعه کوچکی از نمونه‌ها دریافت کنیم، یا اصلاً نمی‌توانیم برچسبی دریافت کنیم. برای مثال، هنگام پیش‌بینی اینکه یک گربه در یک تصویر است یا خیر، به یک انسان نیاز داریم تا هر تصویر را بگیرد و آن را با 0 یا 1 برچسب گذاری کند. همه عکس‌های گربه دارای چنین برچسب مرتبطی هستند. استفاده از یادگیری تحت نظارت فقط در این زیر مجموعه برچسب‌گذاری شده در صورتی که احتمال دارد به دلیل داده‌های محدود، پیش‌بینی‌کننده ضعیفی ایجاد کند. یادگیری نیمه نظارت شده با استفاده از تمام داده‌های بدون برچسب، برای تکمیل مجموعه داده کوچک برچسب گذاری شده، با یافتن ساختار در ویژگی‌ها سروکار دارد. برای مثال، ویژگی‌ها ممکن است روی یک منیفولد با ابعاد پایین‌تر قرار بگیرند. این ساختار را می‌توان از داده‌های بدون برچسب استنباط کرد و به طور بالقوه به طور موثرتری کلاس تابع را برای یادگیری در مجموعه داده برچسب گذاری شده محدود کرد. یادگیری بدون نظارت تنها در به دست آوردن این ساختار متمرکز می‌شود، بدون اینکه هدف یادگیری یک تابع برای پیش‌بینی اهداف باشد، زیرا هیچ هدفی ارائه نمی‌شود. یادگیری بدون نظارت در بخش ۹,۲,۲ به عنوان بخشی از یادگیری بازنمایی مورد بحث قرار خواهد گرفت

این دو مجموعه مسائل را می‌توان به عنوان نمونه‌ای از یک مجموعه بزرگتر یادگیری تحت داده‌های از دست رفته در نظر گرفت. به طور کلی، ممکن است نه تنها جمع‌آوری خروجی‌ها، بلکه برخی از ویژگی‌ها نیز دشوار باشد. به عنوان مثال، هنگام جمع‌آوری داده‌های بیمار، این احتمال وجود دارد که برخی از بیماران برخی از اطلاعات را حذف کنند. اگرچه جمع‌آوری اطلاعات «دارای بیماری» دشوار است، اما اطمینان از جمع‌آوری داده‌های ساده‌تر مانند «سن» یا «وزن» نیز می‌تواند دشوار باشد. علاوه بر این، حتی ممکن است بپرسیم که چرا بین ویژگی‌ها و اهداف تمایز وجود دارد: همه آنها اطلاعات مرتبط با یک مورد هستند، مانند یک بیمار. با توجه به اینکه یک بیمار واقعاً یک بیماری دارد، ممکن است بخواهید از این ویژگی و سن او برای پیش‌بینی وزن او استفاده کنید - چیزی که آنها خیلی راحت ترجیح دادند فاش نکنند. این روش کلی‌تر برای نزدیک شدن به مسئله می‌تواند زمانی مفید باشد که داده‌ها از دست رفته و منجر به مشکل تکمیل کلی شود. تکنیک‌های متفاوتی اغلب در چنین مجموعه‌هایی استفاده می‌شود، و ما تا بخش ۹,۲,۲ بیشتر از این، به آن نخواهیم پرداخت. در حال حاضر، ما تمرکز خود را بر یادگیری نظارت شده حفظ می‌کنیم، که همچنان می‌توانیم از آن استفاده کنیم، حتی زمانی که برخی ویژگی‌ها از دست رفته‌اند، با استفاده از برخی اکتشاف‌های ساده برای مقابله با این داده‌های از دست رفته.

۴-۳ طبقه‌بندی بهینه و مدل‌های رگرسیون

هدف ما اکنون ایجاد معیارهای عملکردی است که برای ارزیابی پیش‌بینی‌کننده‌های $f: \mathcal{X} \rightarrow \mathcal{Y}$ و متعاقباً تعریف مدل‌های طبقه‌بندی و رگرسیون بهینه استفاده می‌شود. برای انجام این کار، فرض می‌کنیم که به توزیع مشترک واقعی $p(\boldsymbol{x}, y)$ دسترسی داریم و می‌پرسیم که پیش‌بینی بهینه در این حالت ایده‌آل چه خواهد بود. پیش‌بینی‌کننده بهینه بر اساس هزینه تابع هزینه تعریف می‌شود: $[0, \infty) \mathcal{Y} \times \mathcal{Y} \rightarrow$، که در آن هزینه $(\hat{y}, y)$ هزینه یا جریمه را برای پیش‌بینی $\hat{y}$ زمانی که هدف واقعی y است منعکس می‌کند. از آنجایی که X، Y تصادفی هستند، هزینه $C = \text{cost}(f(X), Y)$ نیز یک متغیر تصادفی

است، زیرا تابعی از این متغیرهای تصادفی است. هدف ما به مینیمم رساندن هزینه‌های مورد انتظار است. ابتدا چند نمونه از هزینه‌ها را در نظر می‌گیریم و سپس پیش‌بینی کننده‌های بهینه را استخراج می‌کنیم.

۴-۳-۱ نمونه‌هایی از هزینه‌ها

$$cost(\hat{y}, y) = \begin{cases} 0 & when\ y = \hat{y} \\ 1 & when\ y \neq \hat{y} \end{cases} \qquad (4-1)$$

یک تابع هزینه پیچیده‌تر ممکن است در مجموعه‌هایی ایجاد شود که برخی از پیش‌بینی‌های نادرست مشکل‌سازتر از بقیه هستند. بیایید یک مثال عینی را در حوزه پزشکی در نظر بگیریم. فرض کنید هدف ما این است که تصمیم بگیریم که آیا یک بیمار با مجموعه‌ای از علائم خاص (x) باید برای یک آزمایش آزمایشگاهی اضافی ($y = 1$ اگر بله و $y = -1$ اگر نه) با هزینه c_{lab} فرستاده شود تا تشخیص را بهبود بخشد. با این حال، اگر آزمایش آزمایشگاهی انجام ندهیم و بعداً مشخص شود که بیمار برای درمان مناسب به آزمایش نیاز داشته است، ممکن است جریمه قابل توجهی متحمل شویم، مثلاً $c_{lawsuit}$. اگر $c_{lawsuit} \gg c_{lab}$، همانطور که انتظار می‌رود، طبقه‌بندی کننده باید خروجی‌های خود را به طور مناسب تنظیم کند تا تفاوت هزینه را در اشکال مختلف پیش‌بینی نادرست محاسبه کند. در اینجا، هزینه بهتر به صورت جدول نشان داده می‌شود. همیشه نمی‌توان هزینه معناداری را برای تابع تعریف کرد.

		Y	
		-1 (¬Has Disease)	1 (Has Disease)
$\hat{Y}$	-1 (¬Has Disease, No Test)	0	1000
	1 (Has Disease, Do Test)	1	1

جدول ۴٫۳: تابع هزینه برای آزمایشگاه پزشکی، هزینه $(\hat{y}, y)$، با $c_{lawsuit} = 1000$ و $c_{lab} = 1$.

و بنابراین، یک معیار معقول استفاده از هزینه $0,1$ پیش‌فرض در معادله (۴٫۱) است. در رگرسیون، هزینه‌های متداول مربع خطا است.

$$cost(\hat{y}, y) = (\hat{y} - y)^2 \qquad (4-2)$$

و قدر مطلق خطا

$$cost(\hat{y}, y) = |y - y| \qquad (4-3)$$

مربع خطا، نسبت به قدر مطلق خطا مقادیر دورتر از y را به شدت جریمه می‌کند. هزینه‌های بسیار دیگری نیز وجود دارد که در بزرگی اهداف نقش دارد، مانند درصد خطا.

۲-۳-۴ استخراج پیش‌بینی کننده‌های بهینه

ابتدا با استخراج طبقه‌بندی‌کننده بهینه شروع می‌کنیم. می‌توانیم هزینه مورد انتظار را به صورت زیر بیان کنیم، با فرض اینکه ورودی‌ها بردارهای با مقدار حقیقی پیوسته هستند و اهداف از یک مجموعه گسسته Y و $y = f(x)$ برای پیش‌بینی‌کننده f هستند.

$$\mathbb{E}[C] = \int_{\mathcal{X}} \sum_{y \in \mathcal{Y}} cost(f(\boldsymbol{x}), y) p(\boldsymbol{x}, y) d\boldsymbol{x}$$

$$= \int_{x} p(\boldsymbol{x}) \sum_{y \in \mathcal{Y}} cost(f(\boldsymbol{x}), y) p(y|\boldsymbol{x}) d\boldsymbol{x}$$

جایی که ادغام کل فضای ورودی $\mathcal{X} = \mathbb{R}^d$ است. توجه داشته باشید که ما باید یک کلاس را برای هر مشاهده پیش‌بینی کنیم: $f(x)$ تنها می‌تواند یک مقدار $\hat{y}$ را در $\mathcal{Y}$ ارائه دهد. اما مقدار هدف تصادفی است. به همین دلیل طبقه‌بندی کننده بهینه f^* ممکن است نتواند هزینه صفر را به دست آورد. با این حال، به سادگی با نگاه کردن به معادله بالا، می‌توانیم با انتخاب بهترین طبقه‌بندی کننده برای هر x به طور جداگانه، $f^* = \text{argmin } E[C]$ را بدست آوریم.

$$f^*(x) = \underset{\hat{y} \in \mathcal{Y}}{argmin}\ \mathbb{E}[C|X = x]$$

$$= \underset{y \in \mathcal{Y}}{\text{argmin}} \sum_{y \in Y} cost(\hat{y}, y) p(y|x)$$

این طبقه‌بندی کننده، طبقه‌بندی کننده ریسک Bayes نامیده می‌شود.

اگر از تابع هزینه $0 - 1$ استفاده کنیم، در معادله (۴,۱)، طبقه‌بندی کننده ریسک Bayes به سادگی تبدیل می‌شود به:

$$f^*(x) = \underset{\hat{y} \in \mathcal{Y}}{argmin} \sum_{y \in \mathcal{Y}} \text{cost}(\hat{y}, y) \text{p}(y|\mathbf{x})$$

$$= \underset{\hat{y} \in \mathcal{Y}}{argmax} \left(1 - \sum_{y \in \mathcal{Y}} \left(cost(\hat{y}, y) p(y|\boldsymbol{x})\right)\right)$$

$$= \underset{\hat{y} \in \mathcal{Y}}{\text{argmax}} \sum_{y \in \mathcal{Y}} (1 - \text{cost}(\hat{y}, y)) \text{p}(y|\mathbf{x}) \qquad \triangleright \text{ because } \sum_{y \in \mathcal{Y}} \text{p}(y|\mathbf{x}) = 1$$

$$\underset{\hat{y} \in \mathcal{Y}}{\text{argmax}} \sum_{y \in \mathcal{Y}, y \neq \hat{y}} 0 \cdot \text{p}(y|\mathbf{x}) + \sum_{y \in \mathcal{Y}, y = \hat{y}} 1 \cdot \text{p}(y|\mathbf{x})$$

$$= \underset{y \in \mathcal{Y}}{\text{argmax}}\ \text{p}(y|\mathbf{x})$$

بنابراین، اگر $\text{p}(y|\mathbf{x})$ شناخته شده باشد یا بتوان به طور دقیق آن را یاد گرفت، ما به طور کامل مجهز به پیش‌بینی هستیم که هزینه کل را به مینیمم برساند. به عبارت دیگر، ما مسئله به مینیمم رساندن هزینه طبقه‌بندی مورد انتظار یا احتمال خطا را به مسئله توابع یادگیری، به ویژه توزیع‌های احتمال یادگیری تبدیل کرده‌ایم.

تجزیه و تحلیل برای رگرسیون مشابه تجزیه و تحلیل برای طبقه‌بندی است. در اینجا نیز، ما علاقه‌مند به، به مینیمم رساندن هزینه مورد انتظار برای پیش‌بینی هدف واقعی y هستیم که از یک پیش‌بینی کننده $\mathrm{f(x)}$ استفاده می‌شود. هزینه مورد انتظار را می‌توان به این صورت بیان کرد:

$$\mathbb{E}[C] = \int_{\mathcal{X}} \int_{y} \mathrm{cost}(\mathrm{f}(\mathbf{x}), \mathrm{y}) \mathrm{p}(\mathbf{x}, \mathrm{y}) \mathrm{dy} \mathrm{d}\mathbf{x}$$

برای سادگی، مربع خطای معادله (۴,۲) را در نظر خواهیم گرفت.

$$\mathrm{cost}(\mathrm{f(x)}, \mathrm{y}) \;=\; (\mathrm{f(x)} \;-\; \mathrm{y})^2$$

که منجر به

$$\mathbb{E}[C] = \int_{\mathcal{X}} \int_{y} (\mathrm{f}(\mathbf{x}) - \mathrm{y})^2 \mathrm{p}(\mathbf{x}, \mathrm{y}) \mathrm{dy} \mathrm{d}\mathbf{x}$$

$$= \int_{\mathcal{X}} p(\boldsymbol{x}) \underbrace{\int_{y} (f(\boldsymbol{x}) - y)^2 p(y|\boldsymbol{x}) dy d\boldsymbol{x}}_{g(\overbrace{f(\boldsymbol{x})}^{\hat{y}})}$$

با فرض اینکه $\mathrm{f(x)}$ به اندازه کافی منعطف است که به طور جداگانه برای هر واحد حجم dx بهینه شود، می‌بینیم که به مینیمم رساندن $\mathbb{E}[\mathrm{C}]$ ما را به مسئله پیدا کردن $\hat{y}$ برای هر x برای کمینه کردن سوق می‌دهد.

$$\mathrm{g}(\hat{\mathrm{y}}) = \int_{y} (\hat{y} - y)^2 \, p(y|\boldsymbol{x}) dy$$

برای یافتن $\hat{y}$ بهینه، می‌توانیم این مسئله کمینه‌سازی را با یافتن یک نقطه ثابت، مینیمم مطلق، حل کنیم. برای این کار، از g نسبت به $\hat{y}$ مشتق می‌گیریم و نقطه‌ای را می‌یابیم که مشتق برابر با صفر است.

$$\frac{\partial \mathrm{g}(\hat{\mathrm{y}})}{\partial \hat{\mathrm{y}}} = 2 \int_{y} (\hat{y} - y)^2 \, P(y|\boldsymbol{x}) dy \;=\; 0$$

$$\Longrightarrow \hat{\mathrm{y}} \underbrace{\int_{y} p(y|\mathbf{x}) dy}_{=1} = \int_{y} p(y|\boldsymbol{x}) dy$$

$$\Longrightarrow \hat{y} \underbrace{\int_{y} p(y|\mathbf{x}) dy}_{=1} = \int_{y} p(y|\boldsymbol{x}) dy$$

$$\Longrightarrow \hat{\mathrm{y}} \int_{y} p(y|\boldsymbol{x}) dy \;=\; \mathbb{E}[\mathrm{Y}|\mathrm{x}]$$

بنابراین، پیش‌بینی بهینه اینگونه است

$$\mathrm{f}^*(\mathrm{x}) \;=\; \mathbb{E}[\mathrm{Y}\,|\mathrm{x}]$$

بنابراین، مدل رگرسیون بهینه به معنای به مینیمم رساندن مربع خطای بین پیش‌بینی و هدف واقعی، انتظار شرطی $\mathbb{E}[\mathrm{Y}\,|\mathrm{X} = \mathrm{x}]$ است.

تمرین ۵: ما می‌توانیم به طور مشابه پیش‌بینی کننده بهینه برای هزینه خطای مطلق را در رابطه (۴,۳) محاسبه کنیم. نشان دهید که پیش‌بینی کننده بهینه برای خطای مطلق، میانه شرطی، میانه$[Y|X = x]$ است.

موارد فوق باعث شده که یادگیری $p(y|x)$ برای طبقه‌بندی معقول باشد تا خطای طبقه‌بندی $0-1$ کاهش یابد. یک جایگزین برای یادگیری مستقیم $p(y|x)$ این است که به‌ترتیب توزیع‌های شرطی کلاس و پیشین، $p(x|y)$ و $p(y)$ را یاد بگیرید. با استفاده از:

$$p(y|\mathbf{x}) = \frac{p(\mathbf{x},y)}{p(\mathbf{x})}$$
$$= \frac{p(\mathbf{x}|y)p(y)}{p(\mathbf{x})}$$
$$\propto p(\mathbf{x}|y)p(y)$$

می‌توانیم ببینیم که این دو رویکرد یادگیری در تئوری معادل هستند تا $\hat{y}$ با بالاترین $p(y|\mathbf{x})$ تعیین شود. انتخاب به دانش و/یا ترجیحات قبلی ما بستگی دارد. مدل‌هایی که با تخمین مستقیم $p(y|\mathbf{x})$ به دست می‌آیند، مدل‌های متمایز و مدل‌هایی که با تخمین مستقیم $p(\mathbf{x}|y)$ و $p(y)$ به دست می‌آیند، مدل‌های مولد نامیده می‌شوند.

۳-۳-۴ خطای قابل کاهش و کاهش ناپذیر

پس از یافتن مدل رگرسیون بهینه، اکنون می‌توانیم هزینه مورد انتظار را در هر دو مدل بهینه و غیربهینه $f(x)$ بنویسیم. یعنی ما علاقه‌مندیم که $\mathbb{E}[C]$ را بیان کنیم وقتی که

۱. $f(x) = \mathbb{E}[Y|x]$
۲. $f(x) \neq \mathbb{E}[Y|x]$

وقتی $f(x) = \mathbb{E}[Y|x]$، هزینه مورد انتظار را می‌توان به سادگی به این صورت بیان کرد

$$\mathbb{E}[C] = \int_x p(x) \int_y (\mathbb{E}[Y|x] - y)^2 p(y|x) dy dx$$
$$= \int_x p(x) V[Y|X = x] dx$$

به یاد بیاورید که $V[Y|X = \boldsymbol{x}]$ واریانس Y برای x داده شده است. بنابراین هزینه مورد انتظار منعکس کننده هزینه‌های ناشی از نویز یا تغییر در اهداف است. این بهترین سناریو در رگرسیون برای مجذور هزینه خطا است. ما نمی‌توانیم به هزینه کمتر از هزینه مورد انتظار دست یابیم.

موقعیت بعدی زمانی است که $f(x) \neq \mathbb{E}[Y|\mathbf{x}]$ در اینجا، ما با تجزیه مربع خطا ادامه خواهیم داد

$$(f(\mathbf{x}) - y)^2 = (f(\mathbf{x}) - \mathbb{E}[Y|\mathbf{x}] + \mathbb{E}[Y|\mathbf{x}] - y)^2$$
$$= (f(\boldsymbol{x}) - \mathbb{E}[Y|\boldsymbol{x}])^2 + \underbrace{2(f(\boldsymbol{x}) - \mathbb{E}[Y|\boldsymbol{x}])(\mathbb{E}[Y|\boldsymbol{x}] - y)}_{g(\boldsymbol{x},y)} + (\mathbb{E}[Y|\boldsymbol{x}] - y)^2$$

توجه داشته باشید که مقدار مورد انتظار $g(\boldsymbol{x}, Y)$ برای هر $\mathbf{x}$ صفر است زیرا

$$\mathbb{E}[g(\boldsymbol{x},Y)] = \mathbb{E}[(f(\boldsymbol{x}) - \mathbb{E}[Y|\boldsymbol{x}])(\mathbb{E}[Y|\boldsymbol{x}] - Y)|\boldsymbol{x}]$$
$$= (f(\boldsymbol{x}) - \mathbb{E}[Y|\boldsymbol{x}])\mathbb{E}[(\mathbb{E}[Y|\boldsymbol{x}] - Y)|\boldsymbol{x}]$$
$$= (\mathrm{f}(\mathbf{x}) - \mathbb{E}[\mathrm{Y}|\mathbf{x}])\,(\mathbb{E}[\mathrm{Y}|\mathbf{x}] - \mathbb{E}[\mathrm{Y}|\mathbf{x}])$$
$$= 0$$

بنابراین، می‌توانیم نتیجه بگیریم که $\mathbb{E}[\mathrm{g}(\mathbf{X},\mathrm{Y})] = 0$، زمانی که انتظارات را بیش از $\mathbf{X}$ می‌گیریم. اکنون می‌توانیم هزینه مورد انتظار را به صورت این بیان کنیم.

$$\mathbb{E}[\mathrm{C}] = \mathbb{E}[(\mathrm{f}(\mathbf{X}) - \mathrm{Y})^2]$$
$$= \underbrace{\mathbb{E}[(f(\boldsymbol{X}) - \mathbb{E}[Y|\boldsymbol{X}])^2]}_{reducibleerror} + \underbrace{\mathbb{E}[(\mathbb{E}[Y|\boldsymbol{X}] - Y)^2]}_{irreducibleerror}$$

عبارت اول نشان می‌دهد که مدل آموزش دیده $\mathrm{f}(\mathrm{x})$ چقدر از مدل بهینه $\mathrm{E}[\mathrm{Y}|\mathrm{x}]$ فاصله دارد. عبارت دوم منعکس کننده تغییرپذیری ذاتی در Y با دادن $\mathbf{x}$ است، همانطور که در معادله (۴،۴) نوشته شده است. این اصطلاحات اغلب خطاهای تقلیل پذیر و غیر قابل تقلیل نیز نامیده می‌شوند. اگر کلاس توابع f را برای پیش‌بینی $\mathrm{E}[\mathrm{Y}|\mathrm{x}]$ گسترش دهیم، می‌توانیم اولین خطای مورد انتظار را کاهش دهیم. با این حال، خطای دوم ذاتی یا غیرقابل کاهش است به این معنا که هر چقدر عملکرد را بهبود بخشیم، نمی‌توانیم این عبارت را کاهش دهیم. این به مسئله مشاهده‌پذیری جزئی مربوط می‌شود، جایی که همیشه به دلیل کمبود اطلاعات مقداری تصادفی وجود دارد. این فاصله غیرقابل کاهش به طور بالقوه می‌تواند با ارائه اطلاعات ویژگی‌های بیشتر کاهش یابد (یعنی گسترش اطلاعات در x). با این حال، برای یک مجموعه داده معین، با ویژگی‌های داده شده، این خطا غیر قابل کاهش است.

به طور خلاصه، ما در اینجا استدلال کردیم که مدل‌های طبقه‌بندی و رگرسیون بهینه به طور انتقادی به دانستن یا یادگیری دقیق توزیع پسین $\mathrm{p}(\mathrm{y}|\mathrm{x})$ بستگی دارد. این کار را می‌توان به روش‌های مختلفی حل کرد، اما یک رویکرد ساده این است که یک شکل تابعی برای $\mathrm{p}(\mathrm{y}|\mathrm{x})$ فرض کنیم، مثلاً $\mathrm{p}(\mathrm{y}|\mathrm{x},\theta)$، که θ مجموعه‌ای از وزن‌ها یا پارامترهایی است که باید از داده‌ها یاد گرفت شوند.

۴-۴ [پیشرفته] مدل‌های بهینه بیز

قبلاً دیدیم که مدل‌های پیش‌بینی بهینه به یادگیری توزیع پسین $\mathrm{p}(\mathrm{y}|\mathrm{x})$ کاهش می‌یابد که سپس برای به مینیمم رساندن هزینه مورد انتظار (ریسک، هزینه) استفاده می‌شود. با این حال، در عمل، توزیع احتمال $\mathrm{p}(\mathrm{y}|\mathrm{x})$ باید با استفاده از یک فرم تابعی خاص و مجموعه‌ای از ضرایب قابل تنظیم مدل‌سازی شود. هنگامی که $\mathcal{X} = \mathbb{R}^{\mathrm{d}}$ و $\mathcal{Y} = \{0,1\}$، یکی از این مثالها در رگرسیون لجستیک استفاده می‌شود، که در آن

$$\mathrm{p}(1|\mathrm{x}) = \frac{1}{1 + \mathrm{e}^{-(\mathrm{w}_0 + \sum_{j=1}^{d} w_j x_j)}}$$

و $p(0|x) = 1 - p(1|x)$. در اینجا $(w_0, w_1, \ldots, w_d) \in R^d + 1$ مجموعه‌ای از وزن‌ها است که باید از مجموعه داده‌های داده شده $\mathcal{D}$ استنباط شوند و $\boldsymbol{x} \in \mathbb{R}^d$ یک نقطه داده ورودی است. تعدادی از انواع دیگر روابط عملکردی نیز می‌تواند مورد استفاده قرار گیرد که مجموعه وسیعی از امکانات را برای مدل سازی توزیع‌ها فراهم می‌کند.

برای دقیق تر بودن در مورد این اشکال عملکردی، باید نماد خود را طوری تنظیم کنیم که توزیع بر روی y را با x به عنوان نشان دهیم.

$$\mathrm{p}(\mathrm{y}|\mathrm{x}) = \mathrm{p}(\mathrm{y}|\mathrm{x},\mathrm{f})$$

که در آن f یک تابع خاص از یک تابع (فرضیه) فضای $\mathcal{F}$ است. ما می‌توانیم $\mathcal{F}$ را به عنوان مجموعه‌ای از تمام توابع از یک کلاس مشخص، مثلاً برای همه $(w_0, w_1, \ldots, w_d) \in \mathbb{R}^{k+1}$ در نظر بگیریم در مثال بالا، اما ما همچنین می‌توانیم کلاس عملکردی را فراتر از تغییرات ساده پارامتر گسترش دهیم تا سطوح تصمیم‌گیری غیرخطی را بگنجانیم. ما معمولاً یک تابع را با توجه به داده‌ها انتخاب می‌کنیم - مثلاً ماکسیمم احتمال یا راه حل MAP

در عوض می‌توانیم توزیع $Y \mid x$ را روی همه توابع قابل قبول f در نظر بگیریم. در یک مسئله یادگیری معمولی، یک مجموعه داده $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^{n}$ به ما داده می‌شود و از ما خواسته می‌شود که $p(y|x)$ را مدل کنیم. برای این منظور، $\mathcal{D}$ را به عنوان تحقق یک متغیر تصادفی $\mathcal{D}$ در نظر می‌گیریم و فرض می‌کنیم که $\mathcal{D}$ مطابق با توزیع زیربنایی واقعی $p(x, y)$ ترسیم شده است. بنابراین وظیفه ما بیان $p(y|x, \mathcal{D})$ است. با استفاده از قواعد مجموع و محصول، وظیفه اصلی ما برای تخمین $p(y|x)$ به این صورت بازنویسی می‌شود.

$$p(y|x, \mathcal{D}) = \int_{\mathcal{F}} p(y|x, f, \mathcal{D}) p(f|x, \mathcal{D}) df$$

$$= \int_{\mathcal{F}} p(y|x, f) p(f|x, \mathcal{D}) df$$

در اینجا ما از استقلال شرطی بین خروجی Y و مجموعه داده $\mathcal{D}$ استفاده کردیم که یک مدل خاص f بر اساس $\mathcal{D}$ انتخاب شد. بنابراین $p(y|x, f, D) = p(y|x, f)$. این معادله به ما این احساس را می‌دهد که تصمیم بهینه را می‌توان از طریق مخلوطی از توزیع‌های $p(y|x, f)$، که در آن وزن‌ها به صورت چگالی پسینی $p(f|x, D)$ داده می‌شود، اتخاذ کرد. در فضاهای فرضی متناهی $\mathcal{F}$ که داریم

$$p(y|x, \mathcal{D}) = \sum_{f \in \mathcal{F}} p(y|x, f) p(f|x, \mathcal{D})$$

و $p(f|x, \mathcal{D})$ احتمالات پسینی هستند. همچنین ممکن است فرض کنیم که $p(f|x, \mathcal{D}) = p(f|\mathcal{D})$، در این صورت وزن‌ها را می‌توان بر اساس مجموعه داده‌های $\mathcal{D}$ از قبل محاسبه کرد. این منجر به محاسبات کارآمدتر احتمالات پسین می‌شود.

در طبقه‌بندی، می‌توانیم طبقه‌بندی کننده بهینه خود را به این صورت بازنویسی کنیم

$$f^*(x, \mathcal{D}) = \underset{y \in \mathcal{Y}}{\operatorname{argmax}} \; p(y|x, \mathcal{D})$$

که به آسانی به فرمول زیر منجر می‌شود

$$f^*(x, \mathcal{D}) = \underset{y \in \mathcal{Y}}{\operatorname{argmax}} \int_F p(y|x, f, \mathcal{D}) p(f|\mathcal{D}) df$$

$$\underset{y \in \mathcal{Y}}{\operatorname{argmax}} \int_F p(y|x, f) p(\mathcal{D}|f) df$$

می‌توان نشان داد که هیچ طبقه‌بندی کننده‌ای نمی‌تواند از طبقه‌بندی کننده بهینه بیز بهتر عمل کند. جالب توجه است، مدل بهینه بیز همچنین اشاره می‌کند که عملکرد پیش‌بینی بهتری را می‌توان با ترکیب چندین مدل و میانگین‌گیری خروجی‌های آنها به دست آورد. این امر پشتیبانی تئوریکی را برای یادگیری گروه و روش‌هایی مانند بسته‌بندی و تقویت فراهم می‌کند.

با توجه به اینکه تابع (فرضیه) فضای $\mathcal{F}$ به طور کلی غیرقابل شمارش است، یک مسئله در طبقه‌بندی بهینه بیز، محاسبه کارآمد $f^*(x, \mathcal{D})$ است. یک روش برای این کار نمونه برداری از توابع از $\mathcal{F}$ با توجه به $p(f)$ و سپس محاسبه $p(f|\mathcal{D})$ یا $p(\mathcal{D}|f)$ است. این را می‌توان تا زمانی محاسبه کرد که $p(y|x, \mathcal{D})$ همگرا شود.

فصل ۵

رگرسیون خطی

با توجه به مجموعه داده $\mathcal{D} = \{(\mathrm{x_i}\ ,\ \mathrm{y_i})\}_{\mathrm{i=1}}^{\mathrm{n}}$ هدف یادگیری رابطه بین ویژگی‌ها و هدف است. ما معمولاً با فرضیه‌سازی شکل عملکردی این رابطه شروع می‌کنیم. مثلا،

$$\mathrm{f(x)} = w_0 + w_1 x_1 + w_2 x_2$$

که در آن $w = (w_0, w_1, w_2)$ مجموعه‌ای از پارامترهایی است که باید تعیین شوند (آموخته شوند) و $\boldsymbol{x} = (x_1, x_2)$. از طرف دیگر، ممکن است فرض کنیم که $f(x) = \alpha + \beta x_1 x_2$، که در آن $\theta = (\alpha, \beta)$، مجموعه دیگری از پارامترها است که باید آموخته شود. در مورد اول، تابع هدف به عنوان ترکیبی خطی از ویژگی‌ها و پارامترها مدل می‌شود، به عنوان مثال.

$$f(x) = \sum_{j=0}^{d} w_j x_j$$

که $\boldsymbol{x}$ را به صورت $(x_0 = 1, x_1, x_2, \dots, x_d)$ گسترش دادیم. یافتن بهترین پارامترهای $\boldsymbol{w}$ به عنوان مسئله رگرسیون خطی یاد می‌شود، در حالی که همه انواع دیگر روابط بین ویژگی‌ها و هدف در دسته‌ای از رگرسیون غیر خطی قرار می‌گیرند. در هر یک از شرایط، مسئله رگرسیون را می‌توان به عنوان یک رویکرد مدل‌سازی احتمالی ارائه کرد که به تخمین پارامتر کاهش می‌یابد: به یک مسئله بهینه‌سازی با هدف به ماکسیمم رساندن یا به مینیمم رساندن برخی از معیارهای عملکرد بین مقادیر هدف $\{y_i\}_{i=1}^{n}$ و پیش‌بینی‌ها $\{\mathrm{f(x_i)}\}_{\mathrm{i=1}}^{\mathrm{n}}$ ما می‌توانیم یک الگوریتم بهینه‌سازی خاص را به عنوان الگوریتم یادگیری یا آموزش در نظر بگیریم.

۵-۱ فرمول ماکسیمم احتمال

اکنون فرمول آماری رگرسیون خطی را در نظر می‌گیریم. ابتدا مفروضات پشت این فرآیند را بیان می‌کنیم و متعاقباً مسئله را از طریق ماکسیمم کردن تابع احتمال شرطی فرموله می‌کنیم. در بخش بعدی نحوه حل بهینه‌سازی و تجزیه و تحلیل راه‌حل و ویژگی‌های آماری اساسی آن را نشان خواهیم داد.

فرض کنید که مجموعه داده مشاهده شده $\mathcal{D}$ محصول یک فرآیند تولید داده است که در آن n نقطه داده به طور مستقل و بر اساس توزیع یکسان $p(x)$ ترسیم شده است. همچنین فرض کنید که متغیر هدف Y یک رابطه خطی زیربنایی با ویژگی‌های $X = (X_1, X_2, \dots, X_d)$ دارد که با مقداری خطای ε اصلاح شده است که از توزیع گاوسی میانگین صفر پیروی می‌کند،

یعنی $\varepsilon: \mathcal{N}(0, \sigma^2)$. یعنی برای یک ورودی $\boldsymbol{x}$، هدف y تحقق یک متغیر تصادفی Y است که به این صورت تعریف شده است.

$$Y = \sum_{j=0}^{d} \omega_j X_j + \varepsilon$$

که در آن $\boldsymbol{\omega} = (\omega_0, \omega_1, \dots, \omega_d)$ مجموعه‌ای از ضرائب ناشناخته است که ما به دنبال بازیابی آن از طریق تخمین هستیم. به طور کلی، فرض نرمال بودن عبارت خطا معقول است (قضیه حد مرکزی را به یاد بیاورید!)، اگرچه ممکن است استقلال بین ε و $\boldsymbol{X}$ در عمل برقرار نباشد. با استفاده از چند ویژگی ساده انتظارات، می‌توانیم ببینیم که Y نیز از توزیع گاوسی پیروی می‌کند، یعنی چگالی شرطی آن $p(y|\boldsymbol{x},\boldsymbol{\omega}) = \mathcal{N}(\omega^T \boldsymbol{x}, \sigma^2)$ است.

در رگرسیون خطی، ما به دنبال تقریب هدف به صورت $f(x) = \boldsymbol{w^T x}$ هستیم، جایی که وزن‌های $\boldsymbol{w}$ تعیین می‌شوند. ابتدا تابع درستنمایی شرطی را برای یک جفت واحد $(\boldsymbol{x}, \boldsymbol{y})$ به این صورت می‌نویسیم

$$\mathrm{p}(y|\boldsymbol{x},\boldsymbol{\omega}) = \frac{1}{\sqrt{2\pi\,\sigma^2}} exp\left(-\frac{\left(y - \sum_{j=0}^{d} w_j x_j\right)^2}{2\sigma^2}\right)$$

که در آن از نماد $exp(a) = e^a$ استفاده می‌کنیم تا خوانش توان را آسان‌تر کنیم. توجه کنید که تنها تغییر تابع چگالی شرطی Y این است که به جای ω از ضرائب $\boldsymbol{w}$ استفاده می‌شود. با ترکیب کل مجموعه داده $D = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^{n}$، اکنون می‌توانیم تابع درستنمایی شرطی را به صورت $p(\boldsymbol{y}|\boldsymbol{X}, \boldsymbol{w})$ بنویسیم و وزن‌ها را به صورت

$$\mathbf{w}_{\mathrm{MLE}} = \underset{\mathrm{w}}{\arg\max}\{\mathrm{p}(\mathrm{y}|\mathbf{X}, \mathrm{w})\}.$$

از آنجایی که n نمونه مستقل هستند و به طور یکسان توزیع شده‌اند $(i.i.d.)$، ما داریم

$$p(\mathbf{y}|\mathbf{X}, \mathrm{w}) = \prod_{i=1}^{n} \mathrm{p}\left(\mathrm{y_i}|\mathbf{x_i}, \mathbf{w}\right)$$

$$= \prod_{i=1}^{n} \frac{1}{\sqrt{2\pi\sigma^2}} exp\left(-\frac{\left(y_i - \sum_{j=0}^{d} w_{\_i}\, x_{ij}\right)^2}{2\sigma^2}\right)$$

به دلایل راحتی ریاضی، به لگاریتم (تابع یکنواخت) تابع درستنمایی نگاه می‌کنیم و لگاریتم احتمال را به این صورت بیان می‌کنیم.

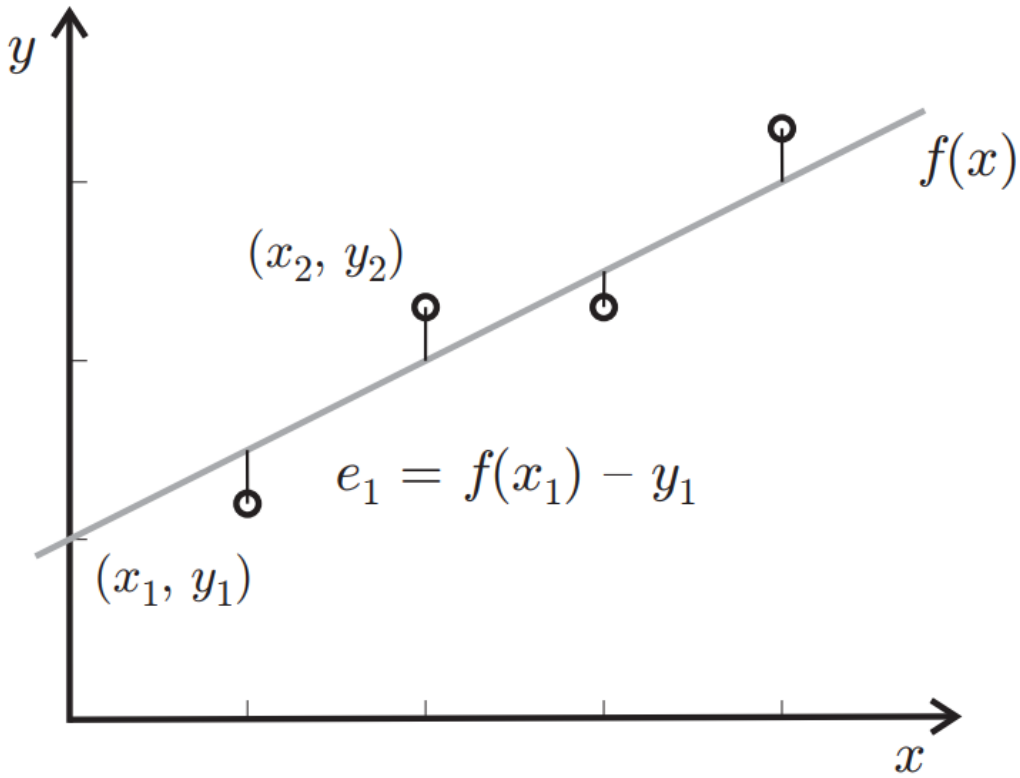
$$\ln\left(\mathrm{p}(\mathrm{y}|\mathbf{X}, \mathrm{w})\right) = -\sum_{\mathrm{i=1}}^{\mathrm{n}} \log\left(\sqrt{2\pi\sigma^2}\right) - \frac{1}{2\sigma^2}\sum_{i=1}^{n}\left(y_i - \sum_{j=0}^{d} w_j\, xij\right)^2$$

با توجه به اینکه جمله اول در سمت راست مستقل از $\mathbf{w}$ است، ماکسیمم کردن تابع درستنمایی دقیقاً با مینیمم کردن مجموع مجذور خطاها مطابقت دارد.

$$Err(w) = \sum_{i=1}^{n} (f(x_i) - y_i)^2 \qquad \rhd\ f(x_i) = \sum_{j=0}^{d} w_j x_{ij} = \hat{y}_i$$

از نظر هندسی، این خطا مربع فاصله اقلیدسی بین بردار پیش‌بینی‌های $\hat{\boldsymbol{y}} = \big(f(x_1), f(x_2), \ldots, f(x_n)\big)$ و بردار مقادیر هدف مشاهده شده $\boldsymbol{y} = (y_1, y_2, .., y_n)$ یک مثال ساده که مسئله رگرسیون خطی را نشان می‌دهد در شکل ۵,۱ نشان داده شده است.

برای اینکه واضح‌تر ببینید که چرا راه‌حل ماکسیمم درستنمایی با کمینه کردن $Err(w)$ مطابقت دارد، توجه کنید که به ماکسیمم رساندن احتمال معادل به ماکسیمم رساندن $log - relihood$ است.



شکل ۵,۱: یک راه‌حل رگرسیون خطی در مجموعه داده $\mathcal{D} = \{(1, 1.2), (2, 2.3), (3, 23), (4, 3.3)\}$ وظیفه فرآیند بهینه‌سازی یافتن بهترین تابع خطی $f(x) = \omega_0 + \omega_1 x$ است به طوری که مجموع مربعات خطاهای $e_1^2 + e_2^2 + e_3^2 + e_4^2$ به مینیمم برسد.

(چون log یکنواخت است) که معادل به مینیمم رساندن احتمال $log - likelihood$ منفی است. بنابراین، ماکسیمم احتمال $\boldsymbol{w}_{MLE}$ می‌شود

$$\mathbf{w}_{\mathrm{MLE}} = \underset{\mathbf{w} \in \mathbb{R}^{\mathrm{d}}}{\operatorname{argmin}} - \ln\big(\mathrm{p}(\mathbf{y}|\mathbf{X}, \mathbf{w})\big)$$

$$\underset{\mathbf{w} \in \mathbb{R}^{\mathrm{d}}}{\operatorname{argmin}} \sum_{\mathrm{i}=1}^{\mathrm{n}} log\left(\sqrt{2\pi\sigma^2}\right) + \frac{1}{2\sigma^2} \sum_{i=1}^{n} \left(y_i - \sum_{j=0}^{d} w_j\, x_{ij}\right)^2$$

$$= \underset{\mathbf{w} \in \mathbb{R}^{\mathrm{d}}}{\operatorname{argmin}} \sum_{i=1}^{n} \left(y_i - \sum_{j=0}^{d} w_j\, x_{ij}\right)^2$$

$$\underset{\mathbf{w} \in \mathbb{R}^{\mathrm{d}}}{\operatorname{argmin}} \; Err(\boldsymbol{w})$$

در بخش‌های بعدی به چگونگی حل این بهینه‌سازی و خواص راه‌حل می‌پردازیم.

توجه داشته باشید که ما می‌توانستیم به سادگی با برخی از توابع خطا (تعریف شده توسط متخصص) شروع کنیم، همانطور که در ابتدا برای OLS و با استفاده از $Err(w)$ انجام شد. با این حال، چارچوب آماری بینش‌هایی را در مورد مفروضات پشت رگرسیون OLS ارائه می‌دهد. به طور خاص، مفروضات شامل این است که داده $\mathcal{D}$ به صورت $\mathrm{i.i.d.}$ ترسیم شده است. یک رابطه خطی اساسی بین ویژگی‌ها و هدف وجود دارد. که نویز (اصطلاح خطا) گاوسی صفر و مستقل از ویژگی‌ها و عدم وجود نویز در مجموعه ویژگی‌ها است.

۲-۵ رگرسیون مینیمم مربعات معمولی[1] (OLS).

برای به مینیمم رساندن مجموع مجذور خطاها، ابتدا $Err(\boldsymbol{w})$ را دوباره می‌نویسیم

$$\mathrm{Err}(\mathbf{w}) = \sum_{i=1}^{n} (\mathrm{f}(\mathbf{x}_\mathrm{i}) - \mathrm{y}_\mathrm{i})^2$$

$$= \sum_{i=1}^{n} \left(\sum_{j=0}^{d} w_j \, x_{ij} - y_i \right)^2$$

جایی که، دوباره، هر نقطه داده $\boldsymbol{x}_i$ را با $x_{i0} = 1$ گسترش دادیم تا عبارت را ساده کنیم.

اکنون گرادیان $\nabla Err(w)$ را محاسبه می‌کنیم. یافتن وزن‌هایی که $\nabla Err(w) = 0$ باعث ایجاد یک نقطه ثابت می‌شود. برای اطمینان از اینکه این نقطه ثابت یک مینیمم مطلق است، به اطلاعات بیشتری نیاز داریم. می‌توانیم به مشتق دوم نگاه کنیم. این مستلزم درک هسین[2] است، بنابراین ما آن را بعداً در یادداشت‌های مثال ۱۶ لحاظ می‌کنیم. اما خوشبختانه، در اینجا حتی ساده‌تر است، زیرا می‌دانیم که این هدف در $\boldsymbol{w}$ محدب است. بنابراین، هر نقطه ثابت یک میینمم مطلق خواهد بود.

اکنون مشتقات جزئی را برابر 0 قرار می‌دهیم و معادلات هر وزن w_j را حل می‌کنیم

$$\frac{\partial \mathrm{Err}}{\partial w_0} = 2 \sum_{i=1}^{n} \left(\sum_{j=0}^{d} w_j x_{ij} - y_i \right) x_{i0} = 0$$

$$\frac{\partial \mathrm{Err}}{\partial w_1} = 2 \sum_{i=1}^{n} \left(\sum_{j=0}^{d} w_j x_{ij} - y_i \right) x_{i1} = 0$$

$$\vdots$$

[1] Ordinary Least-Squares
[2] Hessian

$$\frac{\partial \mathrm{Err}}{\partial w_{\mathrm{d}}} = 2\sum_{i=1}^{n}\left(\sum_{j=0}^{d} w_j x_{ij} - y_i\right)x_{id} = 0$$

این منجر به سیستمی از معادلات خطی $\mathrm{d} + 1$ با مجهولات $\mathrm{d} + 1$ می‌شود که می‌توانند به طور معمول حل شوند (به عنوان مثال با استفاده از حذف گاوسی)

در حالی که این فرمول‌بندی مفید است، به ما اجازه نمی‌دهد که یک راه‌حل به شکل بسته برای $\mathbf{w}$ بدست آوریم یا در مورد وجود یا تعدد این ترکیبات بحث کنیم. برای پرداختن به اولین نکته، مقداری محاسبات ماتریسی را اعمال خواهیم کرد، در حالی که بقیه نکات بعداً مورد بحث قرار خواهند گرفت. ابتدا مجموع مربعات خطاها را با استفاده از نماد ماتریس به این صورت می‌نویسیم

$$\mathrm{Err}(\boldsymbol{w}) = (\mathbf{X}\mathbf{w} - \mathbf{y})^T(\mathbf{X}\mathbf{w} - \mathbf{y}) = \left||\mathbf{X}\mathbf{w} - \mathbf{y}|\right|_2^2$$

که در آن $\left||\boldsymbol{v}|\right|_2 = \sqrt{\boldsymbol{v}^T\boldsymbol{v}} = \sqrt{v_1^2 + v_2^2 + \cdots + v_3^2}$ طول بردار $\boldsymbol{v}$ است. به آن نرم[1] ℓ_2 نیز می‌گویند. اکنون می‌توانیم مسئله رگرسیون خطی مینیمم مربعات معمولی (OLS) را به این صورت فرمول‌بندی کنیم

$$\mathbf{w}_{\mathrm{MLE}} = \underset{\mathbf{w}}{\arg\min}\left||\mathbf{X}\mathbf{w} - \mathbf{y}|\right|_2^2$$

با یافتن $\nabla Err(\boldsymbol{w})$ کار را ادامه می‌دهیم. تابع گرادیان $\nabla Err(\boldsymbol{w})$ مشتقی از یک اسکالر با توجه به یک بردار است. با این حال، مراحل میانی محاسبه گرادیان به مشتقات بردارها با درنظر گرفتن بردارها نیاز دارد (برخی از قوانین چنین مشتقاتی در جدول B.1 نشان داده شده است). به کارگیری قوانین جدول B.1 نتیجه می‌دهد

$$\nabla\mathrm{Err}(\mathbf{w}) = 2\mathbf{X}^{\mathrm{T}}\mathbf{X}\mathbf{w} - 2\mathbf{X}^{\mathrm{T}}\mathbf{y}$$

و بنابراین، از $\nabla Err(\boldsymbol{w}) = 0$ متوجه می‌شویم که

$$\boldsymbol{w}_{MLE} = (\boldsymbol{X}^T\boldsymbol{X})^{-1}\boldsymbol{X}^T\boldsymbol{y}$$

اکنون می‌توانیم مقادیر هدف پیش‌بینی شده را به این صورت بیان کنیم

$$\hat{y} = \boldsymbol{X}\boldsymbol{w}_{MLE} = \boldsymbol{X}(\boldsymbol{X}^T\boldsymbol{X})^{-1}\boldsymbol{X}^T\boldsymbol{y}$$

ماتریس $\boldsymbol{X}(\boldsymbol{X}^T\boldsymbol{X})^{-1}\boldsymbol{X}^T$ ماتریس طرح‌ریزی نامیده می‌شود. به بخش C.1 مراجعه کنید تا بفهمید چگونه $\mathbf{y}$ را به فضای ستون $\mathbf{X}$ می‌فرستد

مثال ۱۳: دوباره مجموعه داده $\mathcal{D} = \{(1, 1.2), (2, 2.3), (3, 2.3), (4, 3.3)\}$ را از شکل ۵،۱ در نظر بگیرید. ما می‌خواهیم ضرائب بهینه مینیمم مربعات متناسب با $f(x) = w_0 + w_1 x$ را پیدا کنیم و سپس مجموع مربعات خطاهای $\mathcal{D}$ را پس از برازش محاسبه کنیم.

$$\boldsymbol{x} = \begin{bmatrix}1 & 1\\ 1 & 2\\ 1 & 3\\ 1 & 4\end{bmatrix}, w = \begin{bmatrix}w_0\\ w_1\end{bmatrix}, y = \begin{bmatrix}1.2\\ 2.3\\ 2.3\\ 3.3\end{bmatrix}$$

[1] norm

که در آن یک ستون از یک‌ها به $\mathbf{x}$ اضافه شد تا یک وقفه غیر صفر مجاز باشد ($y = w_0$ وقتی $x = 0$). جایگزینی $\mathbf{x}$ و $\mathbf{y}$ به معادله (۵،۱) به $w = (0.7, 0.63)$ می‌رسد و مجموع خطاهای مربع برابر $\mathrm{Err}(\mathbf{w}) = 0.223$ است.

همانطور که در مثال بالا مشاهده شد، اضافه کردن ستونی از یک‌ها به ماتریس داده $\mathbf{x}$ به منظور اطمینان از اینکه خط برازش شده، یا به طور کلی یک ابر صفحه، لازم نیست از مبدأ سیستم مختصات عبور کند، یک روش استاندارد است. اما این تأثیر را می‌توان از راه‌های دیگری نیز به دست آورد. اولین جزء بردار گرادیان را در نظر بگیرید

$$\frac{\partial \mathrm{Err}}{\partial \omega_0} = 2\sum_{i=1}^{n}\left(\sum_{j=0}^{d} w_j x_{ij} - y_i\right)x_{i0} = 0$$

جایی که طبق تعریف ، چون $x_{i0} = 1$ ، آن را اینگونه بدست می‌آوریم

$$0 = \sum_{i=1}^{n}\left(\sum_{j=0}^{d} w_j x_{ij} - y_i\right) = \sum_{i=1}^{n}\left(w_0 + \sum_{j=1}^{d} w_j x_{ij} - y_i\right)$$

که می‌دهد

$$\sum_{i=1}^{n} w_0 = \sum_{i=1}^{n} y_i - \sum_{j=1}^{d} w_j \sum_{i=1}^{n} x_{ij}$$

وقتی همه ویژگی‌ها (ستون‌های $\mathbf{X}$) نرمال می‌شوند تا میانگین صفر داشته باشند، یعنی وقتی $\sum_{i=1}^{n} x_{ij} = 0$ برای هر ستون j، نتیجه می‌شود که

$$w_0 = \frac{1}{n}\sum_{i=1}^{n} y_i$$

اکنون می‌بینیم که اگر متغیر هدف به میانگین صفر نیز نرمال شود، نتیجه آن این است که $\omega_0 = 0$ و ستون یک‌ها مورد نیاز نیست.

۵-۲-۱ تابع خطای وزنی

در برخی از کاربردها، به مینیمم رساندن تابع خطای وزنی مفید است

$$\mathrm{Err}(w) = \sum_{i=1}^{n} c_i\left(\sum_{j=0}^{d} w_j x_{ij} - y_i\right)^2$$

جایی که $c_i > 0$ هزینه‌ای برای نقطه داده i است. با بیان آن به صورت ماتریسی، هدف این است که $(\boldsymbol{Xw} - \boldsymbol{y})^T \boldsymbol{C}(\boldsymbol{Xw} - \boldsymbol{y})$ را به مینیمم برسانیم، جایی که $\mathbf{C} = \mathrm{diag}(c_1, c_2, \ldots, c_n)$. با استفاده از روشی مشابه با بالا، می‌توان نشان داد که راه‌حل مینیمم مربعات وزنی $\mathbf{w}_C$ را می‌توان به این صورت بیان کرد.

$$\mathbf{w}_C = \left(\mathbf{X}^T\mathbf{C}\mathbf{X}\right)^{-1}\mathbf{X}^T\mathbf{C}\mathbf{y}$$

علاوه بر این، می‌توان این را استخراج کرد که

$$\mathbf{w}_C = \mathbf{w}_{MLE} + (\mathbf{X}^T\mathbf{C}\mathbf{X})^{-1}\mathbf{X}^T(\mathbf{1}-\mathbf{C})(\mathbf{X}\mathbf{w}_{\mathbf{MLE}} - \mathbf{y})$$

جایی که $\mathbf{w}_{\mathrm{MLE}}$ توسط معادله ارائه شده است. (۵,۱). می‌توانیم ببینیم که راه‌حل‌ها زمانی که $C = I$، و همچنین زمانی که $\boldsymbol{X}\boldsymbol{w}_{MLE} = \boldsymbol{y}$ یکسان هستند، یکسان هستند.

۵-۲-۲ پیش‌بینی چندین خروجی به طور همزمان

گسترش چندین خروجی ساده است، جایی که اکنون هدف یک بردار m بعدی، $y \in \mathbb{R}^m$ است، نه یک اسکالر، که ماتریس هدف $\boldsymbol{Y} \in \mathbb{R}^{n\times m}$ را می‌دهد. به همین ترتیب، وزن‌های $\boldsymbol{W} \in \mathbb{R}^{d\times m}$ برای دادن $\boldsymbol{W}^T\boldsymbol{x} \in \mathbb{R}^m$ با خطا

$$\mathrm{Err}(\mathbf{W}) = \left|\left|\mathbf{X}\mathbf{W} - \mathbf{Y}\right|\right|_{\mathrm{F}}^2 = \sum_{\mathrm{i}=1}^{\mathrm{n}} ||\mathbf{X}_{\mathrm{i},:}\mathbf{W} - \mathbf{Y}_{\mathrm{i},:}||_2^2 \qquad \triangleright \text{Frobenius norm}$$

$$= \mathrm{trace}\left((\mathbf{X}\mathbf{W} - \mathbf{Y})^{\mathrm{T}}(\mathbf{X}\mathbf{W} - \mathbf{Y})\right)$$

و راه‌حل آن

$$\mathbf{W}_{\mathrm{MLE}} = \left(\mathbf{X}^{\mathrm{T}}\mathbf{X}\right)^{-1}\mathbf{X}^{\mathrm{T}}\mathbf{Y}$$

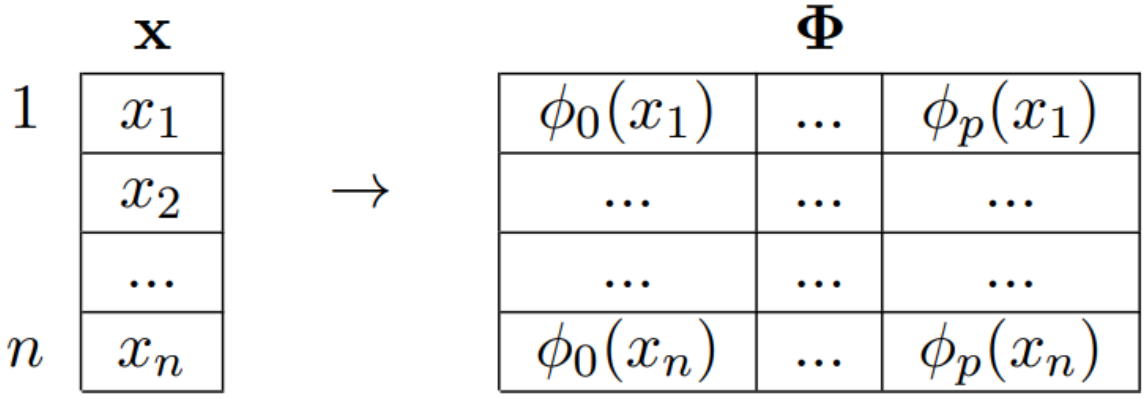
تمرین ۶: این راه‌حل را با گرفتن مشتقات جزئی یا ترجیحاً با استفاده از قوانین گرادیان برای متغیرهای ماتریس استخراج کنید. یک منبع خوب برای گرادیان‌های ماتریس، کتابچه راهنما ماتریس[1] [۱۶] است.

برای به دست آوردن بینش بیشتر در مورد راه‌حل رگرسیون خطی معمولی، چشم انداز جبری را در پیوست C.1 ببینید. بینش بیشتری در مورد منحصر به فرد بودن راه‌حل و فضای راه‌حل‌های ممکن ارائه می‌دهد و بهینه‌سازی رگرسیون خطی را به سیستم‌های حل متصل می‌کند.

۵-۳ رگرسیون خطی برای مسائل غیر خطی

در ابتدا، ممکن است به نظر برسد که کاربرد رگرسیون خطی برای مسائل زندگی واقعی بسیار محدود است. در کل، مشخص نیست که آیا آن را واقع‌بینانه (بیشتر اوقات) فرض کنیم

[1] matrix cookbook



شکل ۵٫۲: تبدیل یک ماتریس داده $\boldsymbol{x}$ با اندازه $n \times 1$ به یک ماتریس $\boldsymbol{\Phi}$ با اندازه $n \times (p+1)$ با استفاده از مجموعه ای از توابع پایه φ_j و $j = 0,1,\ldots,p$

که متغیر هدف ترکیبی خطی از ویژگی‌ها است. خوشبختانه، کاربرد رگرسیون خطی گسترده‌تر است زیرا می‌توانیم از آن برای به دست آوردن توابع غیر خطی استفاده کنیم. ایده اصلی این است که قبل از مرحله برازش، یک تبدیل غیر خطی به ماتریس داده $\mathbf{X}$ اعمال شود، که سپس یک برازش غیر خطی را امکان پذیر می‌کند. به دست آوردن چنین نمایش ویژگی مفیدی یک مشکل اصلی در یادگیری ماشین است. ما در این مورد در فصل ۹ به تفصیل بحث خواهیم کرد. در اینجا، ابتدا یک نمایش ساده‌تر را بررسی خواهیم کرد که یادگیری غیرخطی را امکان پذیر می‌کند: برازش منحنی چند جمله‌ای[1]

۵-۳-۱ برازش منحنی چند جمله‌ای

ما با داده‌های یک بعدی شروع می‌کنیم. در رگرسیون OLS، ما به دنبال تناسب در شکل زیر هستیم

$$f(x) = w_0 + w_1 x$$

که در آن x نقطه داده و $\mathbf{w} = (w_0, w_1)$ بردار وزن است. برای دستیابی به تناسب چند جمله‌ای درجه p، عبارت قبلی را به آن تغییر می‌دهیم

$$f(x) = \sum_{j=0}^{p} w_j x^j$$

که در آن p درجه چند جمله‌ای است. ما این عبارت را با استفاده از مجموعه‌ای از توابع پایه بازنویسی می‌کنیم

$$f(x) = \sum_{j=0}^{p} w_j \varphi^j(x) = \boldsymbol{w}^T \boldsymbol{\varphi}$$

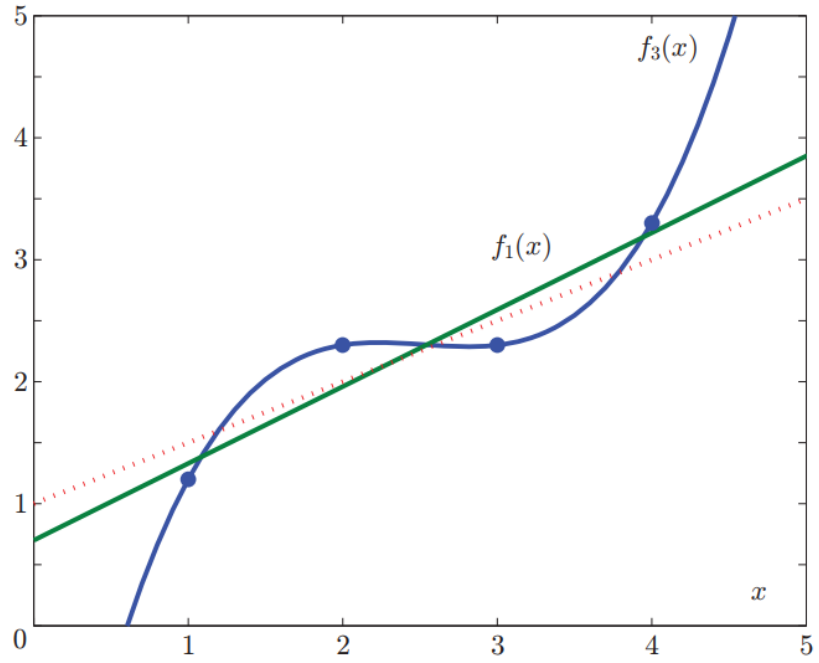
که در آن $\varphi_j(x) = x^j$ و $\varphi = \left(\varphi_0(x), \varphi_1(x), \ldots, \varphi_p(x)\right)$. اعمال این تبدیل برای هر نقطه داده در x منجر به ماتریس داده‌ای جدید Φ می‌شود، همانطور که در شکل ۵٫۲ نشان داده شده است.

[1] Polynomial curve fitting

در ادامه بحث از بخش ۵،۲، مجموعه بهینه وزن‌ها به صورت محاسبه می‌شود

$$\mathbf{w}_{\mathrm{MLE}} = \left(\mathbf{\Phi}^{\mathrm{T}}\mathbf{\Phi}\right)^{-1}\mathbf{\Phi}^{\mathrm{T}}\mathbf{y}$$

مثال ۱۴: در شکل ۵،۱ نمونه‌ای از یک مجموعه داده با چهار نقطه داده را ارائه کردیم. چیزی که ما ذکر نکردیم این بود که با توجه به مجموعه‌ای $\{\mathrm{x}_1, \mathrm{x}_2, \mathrm{x}_3, \mathrm{x}_4\}$، اهداف تولید شدند.



شکل ۵/۳: نمونه ای از تناسب خطی در مقابل چند جمله‌ای در مجموعه داده نشان داده شده در شکل ۵/۱. تناسب خطی، $f_1(x)$، به عنوان یک خط سبز ثابت نشان داده شده است، در حالی که تناسب چند جمله‌ای مکعبی، $f_3(x)$، به عنوان یک خط آبی ثابت نشان داده شده است. خط قرمز نقطه‌چین مفهوم خطی هدف را نشان می‌دهد

با استفاده از تابع $\frac{\mathrm{x}}{2} + ۱$ و سپس اضافه کردن یک خطای اندازه‌گیری $\mathrm{e} = (-0.3, 0.3, -0.2, 0.3)$. مشخص شد که ضرایب بهینه $\mathrm{w}_{\mathrm{MLE}} = (0.7, 0.63)$ به ضرایب واقعی $\omega = (1, 0.5)$ نزدیک است، حتی اگر عبارات خطا نسبتاً معنی‌دار بودند. اکنون سعی خواهیم کرد ضرائب یک تناسب چند جمله‌ای را با درجات $p = 2$ و $p = 3$ تخمین بزنیم. همچنین مجموع خطاهای مجذور $\mathcal{D}$ را پس از برازش و همچنین روی یک مجموعه گسسته بزرگ از مقادیر $\mathrm{x} \in \{0, 0.1, 0.2, \ldots, 10\}$ محاسبه خواهیم کرد. که در آن مقادیر هدف با استفاده از تابع واقعی $\frac{\mathrm{x}}{2} + ۱$ تولید می‌شوند.

استفاده از یک برازش چند جمله‌ای با درجات $\mathrm{p} = 2$ و $\mathrm{p} = 3$ به ترتیب به $\mathbf{w}_2 = (0.575, 0.755, -0.025)$ و $\mathbf{w}_3 = (-3.1, 6.6, -2.65, 0.35)$ منجر می‌شود. مجموع مجذور خطاها در $\mathcal{D}$ برابر است با $\mathrm{Err}(\mathbf{w}_2) = 0.221$ و $\mathrm{Err}(\mathbf{w}_3) \approx 0$. بنابراین، بهترین تناسب با چند جمله‌ای مکعبی به دست می‌آید. با این حال، مجموع خطاهای مجذور در مجموعه داده‌های بیرونی توانایی تعمیم ضعیف مدل مکعبی را نشان می‌دهد زیرا ما $\mathrm{Err}(\mathbf{w}) = 26.9$، $\mathrm{Err}(\mathbf{w}_2) = 3.9$، و $\mathrm{Err}(\mathbf{w}_3) = 22018.5$ به دست می‌آوریم. این اثر بیش از حد برازش[1] نامیده می‌شود. به طور کلی، برازش بیش از حد با تفاوت قابل توجهی در تناسب بین مجموعه داده‌ای که مدل بر روی آن آموزش داده شده است و مجموعه داده‌های خارجی که انتظار می‌رود مدل بر روی آن اعمال شود نشان داده می‌شود (شکل ۵،۳). در این مورد، بیش از حد برازش رخ داد زیرا پیچیدگی مدل به طور قابل توجهی افزایش یافت، در حالی که اندازه مجموعه داده‌ها کوچک باقی ماند.

[1] overfitting

یکی از نشانه‌های برازش بیش از حد، افزایش بزرگی ضرائب است. به عنوان مثال، در حالی که مقادیر مطلق همه ضرایب در $\mathbf{w}$ و $\boldsymbol{w}_2$ کمتر از یک بود، مقادیر ضرایب در $\boldsymbol{w}_3$ با علائم متناوب به طور قابل‌توجهی بزرگ‌تر شدند (که نشان دهنده جبران بیش از حد است). در بخش ۵,۴,۲ به عنوان رویکردی برای جلوگیری از این اثر، قانون‌گذاری را مورد بحث قرار خواهیم داد.

برازش منحنی چند جمله‌ای تنها یکی از راه‌های برازش غیرخطی است، زیرا انتخاب توابع پایه نباید به توان‌های x محدود شود. از جمله توابع پایه غیر خطی که معمولاً مورد استفاده قرار می‌گیرند تابع سیگموئید[1] هستند.

$$\varphi_j(x) = \frac{1}{1 + e^{-\frac{x-\mu_j}{s_j}}}$$

یا تابع نمایی به سبک گاوسی

$$\varphi_j(x) = e^{-\frac{(x-\mu_j)^2}{2\sigma_j^2}}$$

که در آن μ_j، s_j و σ_j ثابت‌هایی هستند که باید تعیین شوند. با این حال، این رویکرد فقط برای یک ورودی یک بعدی x کار می‌کند. برای ابعاد بالاتر، این رویکرد را می‌توان با استفاده از توابع پایه شعاعی تعمیم داد. برای جزئیات بیشتر به بخش ۹,۱ مراجعه کنید.

۵-۴ ثبات و مبادله[2] بایاس واریانس[3]

راه‌حل OLS می‌تواند ناپایدار باشد. در این بخش، ما نشان می‌دهیم که چرا، و بحث می‌کنیم که چگونه می‌توان از منظم‌سازی برای کاهش این مشکل استفاده کرد. سپس یک مفهوم اساسی در یادگیری ماشین را مورد بحث قرار خواهیم داد: مبادله بایاس واریانس.

۵-۴-۱ حساسیت راه‌حل OLS

راه‌حل OLS ناپایدار است اگر $\boldsymbol{X}^T\boldsymbol{X}$ معکوس‌پذیر نباشد. این می‌تواند به دو دلیل اصلی رخ دهد: ویژگی‌های وابسته به خطی و مجموعه داده‌های کوچک. مجموعه داده‌ها اغلب شامل تعداد زیادی ویژگی است که گاهی یکسان، مشابه یا تقریباً خطی وابسته هستند. اگر مجموعه داده کوچک باشد، ممکن است برخی از ویژگی‌ها در نمونه‌ها یکسان باشند، و دوباره منجر به رتبه پایین $\boldsymbol{X}$ می‌شود. وقتی $\boldsymbol{X}^T\boldsymbol{X}$ معکوس‌پذیر نیست یا شرایط نامناسبی دارد، راه‌حل OLS به اغتشاش‌های کوچک در y و $\boldsymbol{X}$ بسیار حساس است.

برای اینکه بفهمیم چرا، به تجزیه مقدار یکتای $\mathbf{X}$ نگاه خواهیم کرد. مانند ساختارهای جبر خطی قبلی، به ما این امکان را می‌دهد که به راحتی ویژگی‌های $\mathbf{X}$ را بررسی کنیم. بیایید حالت رایج را در نظر بگیریم، که در آن $n > d$: تعداد نمونه‌ها بیشتر است. نسبت به بعد ورودی تجزیه مقدار یکتا $\mathbf{X} = \mathbf{U\Sigma V^T}$ برای ماتریس‌های متعامد $\boldsymbol{U} \in \mathbb{R}^{n\times n}$، $V \in \mathbb{R}^{d\times d}$

[1] sigmoid function
[2] trade-of
[3] bias-variance

و ماتریس قطری غیر منفی (مستطیل شکل) $\boldsymbol{\Sigma} \in \mathbb{R}^{n\times d}$. ورودی‌های قطری در $\boldsymbol{\Sigma}$ مقادیر یکتای هستند که معمولاً آنها را به ترتیب نزولی $\sigma_1, \sigma_2, .., \sigma_d$, مرتب می‌کنیم. که میدهد:

$$\sum \overset{\text{def}}{=} \begin{bmatrix} \sigma_1 & 0 & 0 & \cdots & 0 \\ 0 & \sigma_2 & 0 & \cdots & 0 \\ & & \vdots & & \\ 0 & 0 & \cdots & 0 & \sigma_{\mathrm{d}} \\ 0 & 0 & \cdots & 0 & 0 \\ & & \vdots & (\mathrm{n}-\mathrm{d})\text{rows} & \text{of zeros} \\ 0 & 0 & \cdots & 0 & 0 \end{bmatrix} = \begin{bmatrix} \sum_d \\ 0 \end{bmatrix} where \sum_d \overset{\text{def}}{=} \begin{bmatrix} \sigma_1 & 0 & \cdots & 0 \\ 0 & \sigma_2 & \cdots & 0 \\ & & \vdots & \\ 0 & 0 & \cdots & \sigma_d \end{bmatrix}$$

هر ماتریس $\mathbf{X} \in \mathbb{R}^{\mathrm{n}\times\mathrm{d}}$ را می‌توان به تجزیه مقدار یکتای آن تجزیه کرد، زیرا هر تبدیل خطی را می‌توان به یک چرخش (ضرب در $\mathbf{V}^{\boldsymbol{T}}$) تجزیه کرد، به دنبال آن یک مقیاس (ضرب در $\boldsymbol{\Sigma}$) و به دنبال آن دوباره چرخش (ضرب در $\boldsymbol{U}$).

این تجزیه، تجزیه و تحلیل خواص یک ماتریس را ساده می‌کند. برای مثال، تعداد مقادیر غیرصفر یکتا، رتبه $\boldsymbol{X}$ را تشکیل می‌دهد. برای اینکه ببینید چرا، $\sigma_d = 0$ و $\sigma_{d-1} > 0$ را فرض کنید، به این معنی که $\mathbf{X}$ دارای رتبه $\mathrm{d} - 1$ است. هر بردار $\mathrm{w} \in \mathbb{R}^{\mathrm{d}}$ را در نظر بگیرید، و $\mathbf{Xw}$ را در نظر بگیرید. ما می‌توانیم این محصول را به صورت $\mathbf{U\Sigma V^{T}w} = \mathbf{U\Sigma\widetilde{w}}$ برای $\widetilde{\mathbf{w}} = \boldsymbol{V^{T}w}$ بنویسیم. محصول $\boldsymbol{\Sigma\widetilde{w}}$ را تنظیم می‌کند

آخرین بعد $\widetilde{\mathbf{w}}$ به صفر، به طور موثر آن بعد را حذف می‌کند و بنابراین $\widetilde{\mathbf{w}}$ را به فضایی با ابعاد پایین‌تر $(d - 1)$ نشان می‌دهد. سپس آن بردار پیش‌بینی شده را با استفاده از $\boldsymbol{U}$ می‌چرخاند، اما نمی‌تواند آن پیش‌بینی را در فضایی با ابعاد پایین‌تر خنثی کند. بنابراین، $\boldsymbol{Xw}$ فقط می‌تواند $\hat{y} = \boldsymbol{Xw}$ را حاصل کند که در یک صفحه $d - 1$ بعدی قرار دارد که در $\mathbb{R}^{d-1}$ چرخیده است. بنابراین، این تجزیه می‌تواند به ما در درک فضای پیش‌بینی‌های ممکن برای رگرسیون خطی $\boldsymbol{Xw}$ کمک کند.

اکنون می‌توانیم راه‌حل مینیمم مربعات را بر حسب تجزیه مقدار یکتای $\boldsymbol{X}$ مورد بحث قرار دهیم. توجه کنید که

$$\mathbf{X^{T}X} = \mathbf{V\Sigma^{T}U^{T}U\Sigma V^{T}} = \mathbf{V\Sigma_{d}^{2}V^{T}}$$

زیرا $\mathbf{U}$ متعامد است و بنابراین $\mathbf{U^{T}U} = \mathrm{I}$ ماتریس همانی است (I یک ماتریس قطری با یک‌هایی در قطر است). معکوس $\boldsymbol{X^{T}X}$ وجود دارد اگر $\boldsymbol{X}$ رتبه کامل باشد، یعنی Σ_d هیچ صفری در قطر نداشته باشد، زیرا $(\boldsymbol{X^{T}X})^{-1} = \boldsymbol{V\Sigma_{d}^{-2}V^{T}}$. راه‌حل حاصل برای w اینگونه به نظر می‌رسد

$$\mathbf{w} = \left(\mathbf{X^{T}X}\right)^{-1}\mathbf{X^{T}y} = \mathbf{V\Sigma^{-1}U^{T}y} = \sum_{\mathrm{j}=1}^{\mathrm{d}} \frac{\boldsymbol{u_j^{T}y}}{\sigma_{\mathrm{j}}}\boldsymbol{v_j} \qquad (5.2)$$

جایی که $U = [u_1, \ldots, u_n] \in \mathbb{R}^{n\times n}$ ماتریس متعامد متشکل از بردارهای یکتای سمت چپ است، $Ud =$ $[u_1, \ldots, u_d] \in \mathbb{R}^{n\times d}$ اولین d بردار یکتای سمت چپ است و $V = [v_1, \ldots, v_d] \in \mathbb{R}^{d\times d}$ ماتریس متعامد متشکل از بردارهای یکتای راست است.

راه‌حل در معادله (۵،۳) روشن می‌کند که چرا راه‌حل رگرسیون خطی می‌تواند به آشفتگی‌ها حساس باشد. برای مقادیر کوچک یکتا، σ_j^{-1} بزرگ است و هر گونه تغییر در y را تقویت می‌کند. به عنوان مثال، برای مولفه نویز کمی متفاوت $\in_i$ برای نمونه $\boldsymbol{i}$، بردار راه‌حل $\boldsymbol{w}$ می‌تواند بسیار متفاوت باشد. یک استراتژی رایج برای مقابله با این بی ثباتی، حذف یا کوتاه کردن مقادیر کوچک تکی است. این یک شکل منظم‌سازی است که در بخش بعدی به آن می‌پردازیم.

نکته: در حالت کلی، جایی که $\boldsymbol{X}$ رتبه کامل نیست، هنوز هم می‌توانیم یک راه‌حل مینیمم مربعی برای $\boldsymbol{X^T X w = X^T y}$ بدست آوریم. اکنون، راه‌حل‌های بی‌نهایت زیادی وجود دارد. انتخاب رایج انتخاب راه‌حل مینیمم واریانس است که مربوط به حذف مولفه‌ها (بردارهای یکتا) برای مقادیر صفر یکتا است:

$$\boldsymbol{w} = \sum_{j=1}^{rank\ of\ X} \frac{\boldsymbol{u_j^T y}}{\sigma_j} \boldsymbol{v_j} \quad (5.3)$$

مثال ۱۵: [تقریباً وابسته به خطی] بیایید به یک مثال ساده نگاه کنیم که چرا $X \in \mathbb{R}^{n \times d}$ ممکن است مقادیر تکی کوچکی داشته باشد. اول، $d = 2$ و $x_2 = x_1$ را فرض کنید، یعنی ویژگی دوم یک کپی از اولی و به سادگی یک زائد است. سپس $\mathbf{X} = \mathbf{U_2 \Sigma_2 V}^T$ یک SVD باریک است، جایی که $\mathbf{U}_2$ فقط دو ستون اول SVD کامل را دارد. ما می‌توانیم این SVD باریک را بنویسیم زیرا $\mathbf{X} = \mathbf{U_2 \Sigma_2 V}^T \mathbf{U \Sigma V}^T$، که در آن مقادیر یکتای صفر، ستون‌های باقی‌مانده $\mathbf{U}$ را صفر می‌کند.

SVD فقط اولین ستون $x_1 \in \mathbb{R}^{n \times 1}$ ساده است: $x_1 = \mathbf{u}_1 \sigma_1 v_1$، که در آن $\mathbf{u}_1 = x_1 / ||x_1||$، $\sigma_1 = ||x_1||$ و $v_1 = 1$. بنابراین، SVD هست $\mathbf{X} = [x_1 x_2]$ برای هر n است. بردار واحد بعدی $\mathbf{u}_2$ که متعامد به $\mathbf{u}_1$ است و بردارهای یکتا راست $v_1, v_2 \in \mathbb{R}^2$

$$\mathbf{X} = [\mathbf{u}_1\ \mathbf{u}_2] \Sigma [\mathbf{v}_1\ \mathbf{v}_2]^T = [\mathbf{u}_1 \mathbf{u}_2] \begin{bmatrix} 2\sigma_1 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} 0.5 & 0.5 \\ -0.5 & 0.5 \end{bmatrix} = u_1\ \sigma_1 [1.0 \quad 1.0]$$

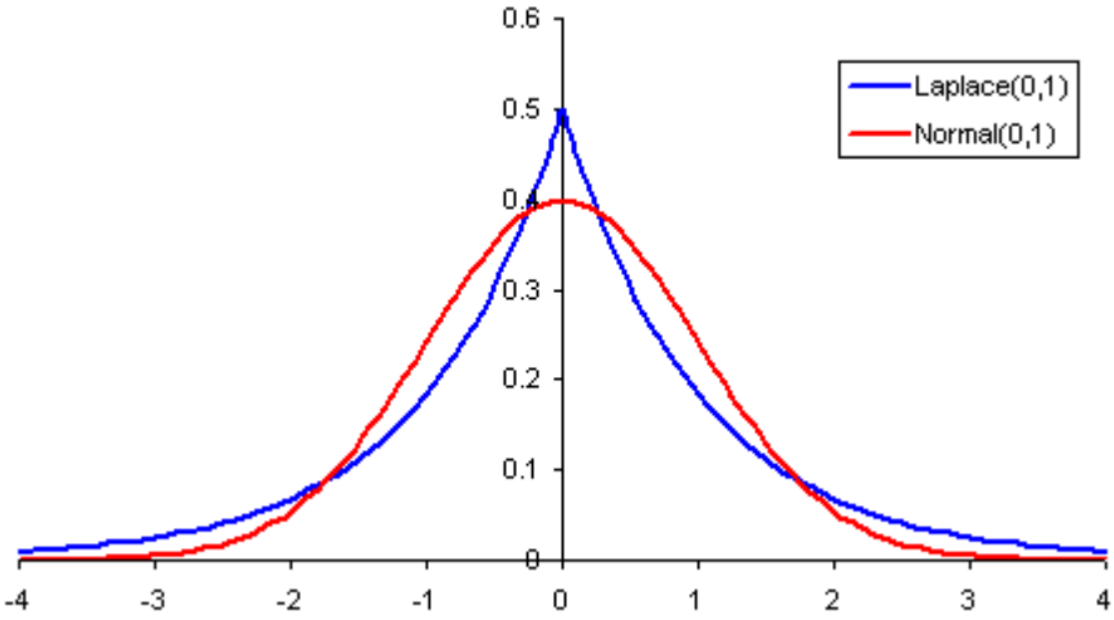


Figure ۱ شکل ۵/۴: مقایسه بین پیشین‌های گاوسی و لاپلاس. هر دو ترجیح می‌دهند مقادیر نزدیک به صفر باشند، اما لاپلاس قبلی به شدت ترجیح می‌دهد مقادیر برابر با صفر باشد.

که در آن ما v_1 را به دو بعدی گسترش دادیم (از آنجایی که $d = 2$)، و v_2 را متعامد به آن بردار تعریف کردیم، و مجبور بودیم σ_1 را تغییر مقیاس دهیم تا بردارهای واحد یکتا را حفظ کنیم. بنابراین چون x_2 به x_1 وابسته است، وقتی آن را به عنوان یک ستون و مقدار یکتای $\sigma_2 = 0$ را اضافه می‌کنیم، رتبه افزایش نمی‌یابد.

اگر به جای $x_2 = x_1 + \epsilon$ برای یک بردار نویز کوچک $\epsilon \in \mathbb{R}^n$ باشد، در عوض می‌بینیم که σ_2 دیگر صفر نخواهد بود، بلکه بسیار نزدیک به صفر خواهد بود، زیرا u_1 و اولین مقدار یکتای σ_1 تا حد زیادی برای بازسازی x_2 قادر خواهند بود.

۲-۴-۵ منظم سازی[1]

تا اینجا در مورد رگرسیون خطی از نظر ماکسیمم احتمال بحث کردیم. اما، مانند قبل، ما همچنین می‌توانیم یک هدف MAP را پیشنهاد کنیم. به جای تعیین هیچ پیش از $\boldsymbol{w}$، می‌توانیم یکی پیش از آن را انتخاب کنیم تا به تنظیم بیش از حد برازش داده‌های مشاهده‌شده کمک کند. ما دو پیشین متداول (تنظیم کننده) را مورد بحث قرار خواهیم داد: پیشین گاوسی (نرم ℓ_2) و پیشین لاپلاس (نرم ℓ_1)، که در شکل ۵,۴ نشان داده شده است.

با گرفتن گزارش گاوسی میانگین صفر قبل، $\mathcal{N}(0, \lambda^{-1}\boldsymbol{I})$، به دست می‌آوریم

$$-\ln p(\mathbf{w}) = \frac{1}{2}\ln(2\pi|\lambda^{-1}\mathbf{I}|) + \frac{\boldsymbol{w}^T\boldsymbol{w}}{2\lambda^{-1}} = \frac{1}{2}\ln(2\pi) - d\ln(\lambda) + \frac{\lambda}{2}\boldsymbol{w}^T\boldsymbol{w}$$

زیرا $|\lambda^{-1}\boldsymbol{I}| = \lambda^{-d}$، که در آن $|\mathbf{A}|$ تعیین کننده ماتریس $\mathbf{A}$ است. مانند قبل، می‌توانیم اولین ثابت را حذف کنیم که بر انتخاب $\mathbf{w}$ تأثیری ندارد.

اکنون می‌توانیم احتمال ورود منفی و گزارش منفی قبلی را با هم ترکیب کنیم. سپس با نادیده گرفتن ثابت‌ها، می‌توانیم log-likelihood منفی و log منفی را با قبل جمع کنیم تا به دست بیاوریم.

$$\underset{w}{\operatorname{argmin}} -\ln\big(p(\mathbf{y}|\mathbf{X}, w)\big) - \ln p(\mathbf{w}) = \underset{w}{\operatorname{argmin}} \frac{1}{2\sigma^2}\sum_{i=1}^{n}\left(y_i - \sum_{j=0}^{d}\omega_j x_{ij}\right)^2 + \frac{\lambda}{2}\boldsymbol{w}^T\boldsymbol{w}$$

$$= \underset{\mathbf{w}}{\operatorname{argmin}} \sum_{i=1}^{n}\left(y_i - \sum_{j=0}^{d}\omega_j x_{ij}\right)^2 + \frac{\lambda\sigma^2}{2}\boldsymbol{w}^T\boldsymbol{w}$$

بنابراین، اگر فرض کنیم که وزن‌ها دارای میانگین گاوسی صفر $\mathcal{N}\left(0, \lambda^{-1}\sigma^2\mathbf{I}\right)$، قبلی هستند آنگاه مسئله رگرسیون پشته زیر را دریافت می‌کنیم:

$$c(\mathbf{w}) = (\mathbf{X}\mathbf{w} - \mathbf{y})^{\mathrm{T}}(\mathbf{X}\mathbf{w} - \mathbf{y}) + \lambda\mathbf{w}^{\mathrm{T}}\mathbf{w} \qquad \triangleright\ ||\mathbf{w}||_2^2 = \mathbf{w}^{\mathrm{T}}\mathbf{w}$$

که در آن λ یک پارامتر انتخاب شده توسط کاربر است که به آن پارامتر منظم‌سازی می‌گویند. ایده این است که ضرائب وزنی بیش از حد بزرگ را جریمه کنیم. هرچه λ بزرگتر باشد، وزنه‌های بزرگ بیشتری جریمه می‌شوند. به همین ترتیب، λ بزرگتر مربوط به یک کوواریانس کوچکتر در قبلی است، که وزن‌ها را نزدیک به صفر می‌کند. بنابراین، برآورد MAP باید بین این قبل از وزن‌ها و برازش داده‌های مشاهده‌شده تعادل برقرار کند.

اگر این معادله را به روش قبلی حل کنیم، به دست می‌آوریم

[1] Regularization

$$\mathbf{w}_{\mathrm{MAP}} = \left(\mathbf{X}^{\mathrm{T}}\mathbf{X} + \lambda\mathbf{I}\right)^{-1}\mathbf{X}^{\mathrm{T}}\mathbf{y}$$

این اثر خوب جابجایی مقادیر مجذور یکتای در Σ_d^2 را λ دارد، تا زمانی که λ خود به اندازه کافی بزرگ باشد، مسائل پایداری را با تقسیم بر مقادیر کوچک حذف می‌کند.

اگر توزیع لاپلاس را انتخاب کنیم، هدف جریمه شده ℓ_1 دریافت می‌کنیم

$$\mathrm{c}(\mathbf{w}) = (\mathbf{X}\mathbf{w} - \mathbf{y})^{\mathrm{T}}(\mathbf{X}\mathbf{w} - \mathbf{y}) + \lambda\left|\left|\mathbf{w}\right|\right|_1$$

که اغلب به آن Lasso می‌گویند. این هدف را می‌توان به طور مشابه با هدف منظم شده ℓ_2 به دست آورد، اما در عوض از توزیع لاپلاس با پارامتر λ برای قبلی استفاده کرد. همانند تنظیم کننده ℓ_2 برای رگرسیون برآمدگی، این تنظیم کننده مقادیر بزرگ در $\mathbf{w}$ را جریمه می‌کند. با این حال، راه‌حل‌های پراکنده‌تری را نیز تولید می‌کند، جایی که ورودی‌های $\mathbf{w}$ صفر هستند. این ترجیح را می‌توان در شکل ۵,۴ مشاهده کرد، جایی که توزیع لاپلاس بیشتر حول صفر متمرکز شده است. با این حال، در عمل، این ترجیح حتی قوی‌تر از آن چیزی است که توزیع نشان می‌دهد، به دلیل اینکه چگونه از دست دادن مینیمم مربعات کروی و تنظیم‌کننده «۱» با هم تعامل دارند.

اجباری کردن ورودی‌های $\mathbf{w}$ به صفر تأثیر انتخاب ویژگی دارد، زیرا صفر کردن ورودی‌ها در $\mathbf{w}$ معادل حذف ویژگی مربوطه است. هر بار که یک پیش‌بینی انجام می‌شود، محصول نقطه‌ای را در نظر بگیرید،

$$\mathbf{x}^{\mathrm{T}}\mathbf{w} = \sum_{j=0}^{d} x_j w_j = \sum_{j:\omega_j \neq 0} x_j w_j$$

این معادل به سادگی حذف ورودی در x و w است که در آن $\mathrm{w_j} = 0$ است.

برای Lasso، ما دیگر راه‌حلی به شکل بسته نداریم. ما یک راه‌حل شکل بسته نداریم، زیرا نمی‌توانیم $\mathbf{w}$ را در فرم بسته‌ای که نقطه ثابتی را ارائه می‌دهد حل کنیم. در عوض، از گرادیان نزول برای محاسبه راه‌حل $\mathbf{w}$ استفاده می‌کنیم. با این حال، تنظیم‌کننده ℓ_1 در 0 غیر قابل تمایز است. درک چگونگی بهینه‌سازی این هدف به پیشینه بهینه‌سازی کمی بیشتری نیاز دارد، بنابراین ما این الگوریتم را در فصل بعدی، در الگوریتم ۴ ارائه می‌کنیم.

۳-۴-۵ انتظار و واریانس برای راه‌حل منظم

یک سوال طبیعی که پیش می‌آید این است که چگونه می‌توان این پارامتر منظم‌سازی را انتخاب کرد و تأثیر آن بر بردار حل نهایی. انتخاب این پارامتر منظم‌سازی منجر به مبادله بایاس واریانس می‌شود. برای درک این مبادله، باید بدانیم که منظور از بایاس دار بودن راه‌حل چیست و چگونه می‌توان واریانس راه‌حل را در مجموعه داده‌های ممکن مشخص کرد.

اجازه دهید با درک بایاس و واریانس با یک راه‌حل غیرقانونی شروع کنیم، با این فرض که مفروضات توزیعی پشت رگرسیون خطی درست هستند. این بدان معنی است که یک پارامتر واقعی ω وجود دارد به طوری که برای هر یک از نقاط داده $Y_i = \sum_{j=0}^{d} \omega_j X_{ij} + \varepsilon_i$، جایی که ε_j ، $i.i.d$ است. متغیرهای تصادفی بر اساس $\mathcal{N}(0, \sigma^2)$ ترسیم شده‌اند. ما می‌توانیم بردار حل (تخمین‌گر) $\mathrm{w_{MLE}}$ را به‌عنوان یک متغیر تصادفی مشخص کنیم، جایی که تصادفی بودن در میان مجموعه‌های داده ممکن است، که می‌توانستند مشاهده شوند. از این نظر، ما مجموعه داده $\mathcal{D}$ را یک متغیر تصادفی در نظر می‌گیریم و راه‌حل $w_{ML}(\mathcal{D})$ از آن مجموعه داده را به عنوان تابعی از این متغیر تصادفی در نظر می‌گیریم.

اجازه دهید اکنون به مقدار مورد انتظار (با توجه به مجموعه داده‌های آموزشی $\mathcal{D}$) برای بردار وزن w_{ML} با $\varepsilon = (\varepsilon_1, \varepsilon_2, \ldots, \varepsilon_n)$ نگاه کنیم:

$$\mathbb{E}[\mathrm{w_{ML}}(\mathcal{D})] = \mathbb{E}\left[\left(\mathbf{X}^{\mathrm{T}}\mathbf{X}\right)^{-1}\mathbf{X}^{\mathbf{T}}(\mathbf{X}\omega + \varepsilon)\right]$$

$$= \mathbb{E}[(\mathbf{X}^T\mathbf{X})^{-1}(X^TX)] \; + \; \mathbb{E}[(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^{\mathbf{T}}\varepsilon]$$

$$= \; \mathbb{E}\,[\omega] \; + \mathbb{E}[(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^{\mathbf{T}}]\;\mathbb{E}[\varepsilon]$$

$$= \; \omega$$

که در آن تساوی سوم از این واقعیت ناشی می‌شود که عبارت‌های نویز ε مستقل از ویژگی‌ها و آخرین تساوی هستند، زیرا ω یک بردار ثابت (غیر تصادفی) است و $\mathbb{E}[\varepsilon] \; = \; 0$. برآوردگر که مقدار مورد انتظار آن مقدار واقعی است. پارامتر را برآوردگر بی طرف می‌نامند. ماتریس کوواریانس برای مجموعه بهینه پارامترها را می‌توان به این صورت بیان کرد

$$\mathrm{Cov}[\mathrm{w_{ML}}(\mathcal{D})] = \mathbb{E}[(\mathrm{w_{ML}}(\mathcal{D}) - \omega)(\mathrm{w_{ML}}(\mathcal{D}) - \omega)^{\mathrm{T}}]$$

$$= \mathbb{E}[\mathbf{w}_{\mathrm{ML}}(\mathcal{D})\mathbf{w}_{\mathrm{ML}}(\mathcal{D})^{\mathrm{T}}] - \boldsymbol{\omega}\boldsymbol{\omega}^{\mathrm{T}}$$

ما $\mathbf{X}^{\dagger} = \left(\mathbf{X}^{\mathrm{T}}\mathbf{X}\right)^{-1}\mathbf{X}^{\mathrm{T}}$ را میگیریم و داریم $\mathbf{w}_{\mathrm{ML}}(\mathcal{D}) = \omega + \mathbf{X}^{\dagger}\varepsilon$ در نتیجه

$$\mathrm{Cov}[\mathbf{w}_{ML}(\mathcal{D})] \; = \; \mathbb{E}[(\omega + \; \mathbf{X}^{\dagger}\,\varepsilon)(\omega + \; \boldsymbol{X}^{\dagger}\,\varepsilon)^T \; - \; \boldsymbol{\omega}\boldsymbol{\omega}^T$$

$$\boldsymbol{\omega}\boldsymbol{\omega}^T \; + \; \mathbb{E}[\boldsymbol{X}^{\dagger}\,\varepsilon\varepsilon^T\boldsymbol{X}^{\dagger\mathrm{T}}] \; - \; \boldsymbol{\omega}\boldsymbol{\omega}^T$$

زیرا $\mathbb{E}[\omega + \; \mathbf{X}^{\dagger}\,\varepsilon] \; = \mathbb{E}[\mathbf{X}^{\dagger}]\mathbb{E}[\varepsilon]\;\boldsymbol{\omega}^T \; = 0$. اکنون چون عبارات نویز مستقل از ورودی‌ها هستند، یعنی $\mathbb{E}[\varepsilon\varepsilon^T|\boldsymbol{X}] = \mathbb{E}[\varepsilon\varepsilon^T] \; = \sigma^2\mathbf{I}$، می‌توانیم از قانون احتمال کل (که قانون برج نیز نامیده می‌شود) استفاده کنیم.

$$\mathbb{E}\left[\mathrm{X}^{\dagger}\varepsilon\varepsilon^{\mathrm{T}}\mathrm{X}^{\dagger\mathrm{T}}\right] = \mathbb{E}\left[\mathbb{E}\left[\mathrm{X}^{\dagger}\varepsilon\varepsilon^{\mathrm{T}}\mathrm{X}^{\dagger\mathrm{T}}|\mathrm{X}\right]\right]$$

$$= \mathbb{E}\left[\mathrm{X}^{\dagger}\mathbb{E}[\varepsilon\varepsilon^{\mathrm{T}}|\mathrm{X}]\mathrm{X}^{\dagger\mathrm{T}}\right]$$

$$= \; \sigma\text{^}2\;\mathbb{E}\;[\mathrm{X}^{\dagger}\mathrm{X}^{\dagger\mathrm{T}}]$$

بنابراین، ما داریم

$$\mathrm{Cov}[\mathbf{w}_{\mathrm{ML}}(\mathrm{D})] = \sigma^2\mathbb{E}\left[\left(\mathbf{X}^{\mathrm{T}}\mathbf{X}\right)^{-1}\right]$$

می‌توان نشان داد که برآوردگر $\mathrm{w_{ML}}(\mathcal{D}) = \mathbf{X}^{\dagger}\mathbf{y}$ دارای کمترین واریانس در بین همه برآوردگرهای بی طرف است (قضیه گاوس-مارکوف).

با این حال، متأسفانه، همانطور که در بالا توضیح داده شد، ماتریس $\boldsymbol{X}^T\boldsymbol{X} = \boldsymbol{V}\boldsymbol{\Sigma}\boldsymbol{V}^T$ می‌تواند شرایط ضعیفی داشته باشد، با مقادیری صفر یا نزدیک به صفر. در نتیجه، این ماتریس کوواریانس می‌تواند شرط بدی شود، با مقادیر کوواریانس با بزرگی بالا. این نشان می‌دهد که در بین مجموعه‌های داده، راه‌حل $w_{ML}(\mathcal{D})$ می‌تواند بسیار متفاوت باشد. این نوع رفتار حکایت از تناسب بیش از حد دارد و مطلوب نیست. اگر راه‌حل ما می‌تواند در چندین زیر مجموعه تصادفی مختلف داده بسیار متفاوت باشد، نمی‌توانیم به هیچ یک از این راه‌حل‌ها اطمینان داشته باشیم.

از سوی دیگر، راه‌حل منظم شده، بسیار کمتر احتمال دارد که کوواریانس بالایی داشته باشد، اما دیگر بی طرف نخواهد بود. اجازه دهید $\boldsymbol{w}_{MAP}(\mathcal{D})$ تخمین MAP برای مسئله منظم ℓ_2با مقدار $\lambda > 0$ باشد. با استفاده از تحلیلی مشابه با بالا، مقدار مورد انتظار $\boldsymbol{w}_{MAP}(\mathcal{D})$ برابر است با

$$\begin{aligned}\mathrm{E}[\mathbf{w}_{\mathrm{MAP}}(\mathcal{D})] &= \mathbb{E}\left[\left(\mathbf{X}^{\mathrm{T}}\mathbf{X}+\lambda\mathbf{I}\right)^{-1}\mathbf{X}^{\mathrm{T}}(\mathbf{X}\omega+\varepsilon)\right] \\ &= \mathbb{E}\left[\left(\mathbf{X}^{\mathrm{T}}\mathbf{X}+\lambda\mathbf{I}\right)^{-1}\left(\mathbf{X}^{\mathrm{T}}\mathbf{X}\right)\omega\right] \\ &\neq \omega\end{aligned}$$

همانطور که $\lambda \rightarrow 0$، راه‌حل MAP به بی طرف بودن نزدیک و نزدیکتر می‌شود. کوواریانس برابر است با

$$\mathrm{Cov}[\mathbf{w}_{\mathrm{MAP}}(\mathcal{D})] = \sigma^2\mathbb{E}\left[\left(\mathbf{X}^{\mathrm{T}}\mathbf{X}+\lambda\mathbf{I}\right)^{-1}\left(\mathbf{X}^{\mathrm{T}}\mathbf{X}\right)\left(\mathbf{X}^{\mathrm{T}}\mathbf{X}+\lambda\mathbf{I}\right)^{-1}\right]$$

این کوواریانس بسیار کمتر مستعد ابتلا به شرایط نامناسب $\boldsymbol{X}^T\boldsymbol{X}$ است، زیرا همانطور که در بالا بحث شد، تغییر λ شرایط را بهبود می‌بخشد. در نتیجه، ما انتظار داریم که $\boldsymbol{w}_{MAP}$ واریانس کمتری در بین مجموعه داده‌های مختلف داشته باشد که می‌توان مشاهده کرد. این به طور متناظر نشان می‌دهد که ما کمتر به مجموعه داده‌ای اضافه می‌کنیم. توجه داشته باشید که با $\lambda \rightarrow \infty$ ، واریانس به صفر کاهش می‌یابد، اما بایاس به مقدار ماکسیمم آن افزایش می‌یابد (یعنی نرم وزن‌های واقعی). همانطور که در شکل ۵،۵ نشان داده شده است، یک انتخاب بهینه از λ وجود دارد که این مبادله بایاس واریانس را به مینیمم می‌رساند - اگر بتوانیم آن را پیدا کنیم.

دلیل اینکه ما به بایاس و واریانس اهمیت می‌دهیم این است که میانگین مجذور خطای مورد انتظار نسبت به وزن‌های واقعی را می‌توان به بایاس و واریانس تجزیه کرد. تا ببینیم چرا

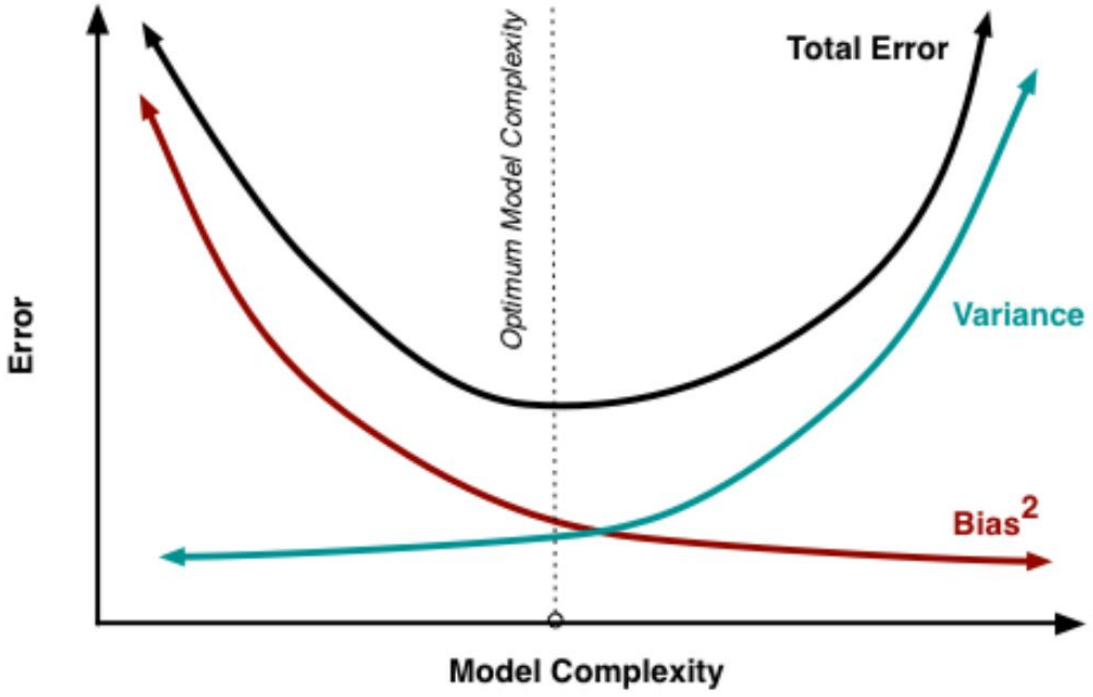
$$\begin{aligned}\mathbb{E}\left[\left|\left|\mathrm{w}(\mathcal{D})-\omega\right|\right|_2^2\right] &= \mathbb{E}\left[\sum_{\mathrm{j}=1}^{\mathrm{d}}\left\{w_{\mathrm{j}}(\mathcal{D})-w_{\mathrm{j}}\right)^2\right] \\ &= \sum_{j=1}^{d}\mathbb{E}\left[\left(w_j(\mathcal{D})-w_j\right)^2\right]\end{aligned}$$

که در آن ما می‌توانیم این اصطلاح درونی را بیشتر ساده کنیم

$$\begin{aligned}\mathrm{E}\left[\left(w_{\mathrm{j}}(\mathcal{D})-w_{\mathrm{j}}\right)^2\right] &= \mathrm{E}\left[w_j(\mathcal{D})-\mathbb{E}\left[\mathrm{w}_{\mathrm{j}}(\mathcal{D})\right]+\mathrm{E}\left[w_{\mathrm{j}}(\mathrm{D})\right]-\omega_{\mathrm{j}}\right)^2 \\ &= \mathbb{E}\left[\left(w_{\mathrm{j}}(\mathcal{D})-\mathbb{E}\left[w_{\mathrm{j}}(\mathcal{D})\right]\right)^2\right]+\mathbb{E}\left[\left(\mathbb{E}\left[w_{\mathrm{j}}(\mathcal{D})\right]-\omega_{\mathrm{j}}\right)^2\right]\end{aligned}$$

جایی که مرحله دوم از این واقعیت ناشی می‌شود که

$$\begin{aligned}-2\mathbb{E}\left[\left(w_{\mathrm{j}}(\mathcal{D})-\mathbb{E}\left[w_{\mathrm{j}}(\mathcal{D})\right]\right)\left(\mathbb{E}\left[w_{\mathrm{j}}(\mathcal{D})\right]-\omega_{\mathrm{j}}\right)\right] &= \left(\mathbb{E}\left[w_{\mathrm{j}}(\mathcal{D})\right]-\omega_{\mathrm{j}}\right)\mathbb{E}\left[w_{\mathrm{j}}(\mathcal{D})-\mathbb{E}\left[w_{\mathrm{j}}(\mathcal{D})\right]\right] \\ &= 0\end{aligned}$$



شکل ۵/۵: مبادله بایاس واریانس. منبع: http://scott.fortmann-roe. com/docs/BiasVariance.html

جمله اول بالا در $\mathbb{E}\left(w_j(\mathcal{D}) - \omega_j\right)^2$ واریانس وزن j و جمله دوم بایاس وزن j است که در آن $\mathbb{E}\left(\mathbb{E}\left[w_j(\mathcal{D})\right] - \omega_j\right)^2$ زیرا هیچ چیز در این عبارت تصادفی نیست بنابراین انتظار بیرونی حذف می‌شود. که یعنی

$$\mathbb{E}\left[\left(\mathrm{f}_{\mathcal{D}}(\mathbf{x}) - \mathrm{f}(\mathbf{x})\right)^2\right] = \left(\mathbb{E}\left[f_{\mathcal{D}}(\boldsymbol{x})\right] - \mathrm{f}(\mathbf{x})\right)^2 + \mathrm{V}\left[\mathrm{f}_{\mathcal{D}}(\mathbf{x})\right]$$

نشان می‌دهد که خطای میانگین مربعات مورد انتظار برای بردار وزن واقعی $\boldsymbol{\omega}$ به بایاس مربع تجزیه می‌شود - که در آن بایاس $\mathbb{E}\left[w_j(\mathcal{D})\right] - \omega_j$ است - و واریانس. مبادله واریانس بایاس این واقعیت را منعکس می‌کند که تا زمانی که واریانس بیشتر از بایاس مجذور کاهش یابد، می‌توانیم به طور بالقوه خطای میانگین مربع را با اعمال برخی سوگیری کاهش دهیم.

نکته: ما به طور مستقیم مبادله سوگیری-واریانس را بهینه نمی‌کنیم. ما در واقع نمی‌توانیم بایاس را اندازه‌گیری کنیم، بنابراین مستقیماً این اصطلاحات را به مینیمم نمی‌رسانیم. در عوض، این تجزیه نحوه انتخاب مدل‌ها را راهنمایی می‌کند.

تمرین ۷: فرمول کوواریانس را برای $w_{MAP}(\mathcal{D})$ استخراج کنید

۵-۵ مبادله بایاس واریانس[1]

در بالا فرض کردیم که مدل واقعی خطی است، و بنابراین تنها سوگیری معرفی شده از قانون‌گذاری بود. این فرض بر این بود که فضای فرضی توابع خطی شامل تابع واقعی است، و این که سوگیری معرفی شده تنها به دلیل قاعده‌سازی است. در واقع، هنگام استفاده از رگرسیون خطی با منظم‌سازی، هم از انتخاب یک کلاس تابع ساده‌تر و هم از نظم‌دهی، بایاس را معرفی می‌کنیم. اگر تابع درست خطی نباشد، نمی‌توان وزن‌های آموخته‌شده را برای یک تابع خطی مستقیماً با تابع واقعی مقایسه کرد.

[1] Bias-Variance

اگر از یک مبنای قدرتمند برای تبدیل داده‌ها استفاده شود، آنگاه می‌توانیم توابع غیرخطی را یاد بگیریم حتی اگر راه حل از رگرسیون خطی استفاده کند. در این حالت، امکان پذیر است که این کلاس تابع به اندازه کافی قدرتمند باشد و تابع واقعی را شامل شود، و این سوگیری بیشتر به دلیل منظم شدن است. اما، به طور کلی، تضمین اینکه ما یک کلاس تابعی را که شامل تابع $true$ باشد، سخت خواهد بود، و مقایسه مستقیم پارامترهای ما با پارامترهای واقعی (که ممکن است حتی از یک بعد هم نباشند) دشوار خواهد بود.

با در نظر گرفتن خطای تقلیل‌پذیر، می‌توانیم به طور کلی‌تر در مورد سوگیری و واریانس صحبت کنیم. در واقع، مبادله بایاس واریانس تماماً در مورد کاهش خطای قابل کاهش است. (به یاد داشته باشید، ما نمی‌توانیم خطای تقلیل‌ناپذیر را کاهش دهیم - نام همه چیز را می‌گوید - با بهبود نحوه تخمین تابع.) ما می‌توانیم یک تجزیه بایاس واریانس کلی‌تر تعریف کنیم که خروجی‌های تابع را به جای بردارهای پارامتر مقایسه می‌کند. معادله خطای تقلیل پذیر را به $\mathbb{E}\left[\left(f_D(\boldsymbol{X}) - f(\boldsymbol{X})\right)^2\right]$ به یاد بیاورید، که در آن $f(\boldsymbol{X})$ تابع بهینه است، یعنی $f(\boldsymbol{x}) = \mathbb{E}[Y\,|\boldsymbol{x}]$ برای مربع هزینه. ما قبلاً در مورد این خطای تقلیل‌پذیر برای یک تابع ثابت بحث کرده‌ایم، با انتظار فقط بیش از $\boldsymbol{X}$. اما اکنون به علاوه این واقعیت را در نظر می‌گیریم که f_D تصادفی است، و می‌توانیم در مورد انتظار و واریانس آن برای $\boldsymbol{x}$ معین استدلال کنیم.

بیایید فقط با در نظر گرفتن خطای میانگین مربع مورد انتظار، برای یک ورودی داده شده $\boldsymbol{x}$ شروع کنیم. با استفاده از مراحل مشابه تجزیه بالا، دریافت می‌کنیم

$$\mathbb{E}\left[\left(\mathrm{f}_{\mathcal{D}}(\mathrm{x}) - \mathrm{f}(\mathrm{x})\right)^2\right]$$

$$= \left(\mathbb{E}[\mathrm{f}_{\mathcal{D}}(\mathrm{x})] - \mathrm{f}(\mathrm{x})\right)^2 + \mathrm{V}[f_{\mathcal{D}}(x)]$$

توجه کنید که در خط دوم، انتظار کنونی در مجذور فاصله است. این عبارت با بایاس مربعی مطابقت دارد. بایاس در اینجا خروجی تابع تخمینی $f_D(x)$ را در تمام مجموعه‌های داده $\mathcal{D}$ منعکس می‌کند. اصطلاح واریانس نشان می‌دهد که پیش‌بینی $\boldsymbol{x}$ چقدر می‌تواند متفاوت باشد، اگر در مجموعه داده‌های $i.i.d.$ مختلف یاد بگیریم. این تجزیه خطای میانگین مربع به بایاس و واریانس مجذور آشکار نیست، اما مراحل مشابه بالا را دنبال می‌کند. به عنوان تمرین باقی می‌ماند

تعمیم بالا نشان می‌دهد که یکی از راه‌هایی که ما بایاس و واریانس را متعادل می‌کنیم، در واقع انتخاب کلاس تابع است. اگر یک کلاس تابع ساده را انتخاب کنیم، کلاس احتمالاً به اندازه کافی بزرگ نیست - به اندازه کافی قدرتمند نیست - تا تابع واقعی را نشان دهد. این مقداری بایاس را معرفی می‌کند، اما احتمالاً واریانس کمتری نیز دارد، زیرا آن کلاس تابع ساده‌تر احتمال کمتری دارد که به هر مجموعه داده اضافه شود. اگر این کلاس خیلی ساده باشد، می‌توانیم بگوییم که تابع ما زیر پارامتر است و زیر برازش است. از طرف دیگر، اگر یک کلاس تابع قدرتمندتر را انتخاب کنیم که حاوی تابع واقعی باشد، ممکن است هیچ گونه سوگیری نداشته باشیم، اما به دلیل توانایی یافتن تابعی در کلاس بزرگ شما که بیش از حد با یک مجموعه داده معین مطابقت دارد، واریانس بالایی داشته باشیم. در این تنظیمات، ممکن است بگوییم که تابع بیش از حد پارامتر شده است، و اگرچه ما توانایی یادگیری یک تابع بسیار دقیق را داریم، یافتن آن تابع در این کلاس بزرگ‌تر مشکل خواهد بود. در عوض، احتمالاً مدلی انتخاب می‌شود که با داده‌های داده‌شده بیش از حد برازش می‌کند و به داده‌های جدید تعمیم نمی‌یابد (یعنی در داده‌های جدید ضعیف عمل می‌کند).

یافتن تعادل بین بایاس و واریانس، و بین عدم تناسب و برازش بیش از حد، یک مشکل اصلی در یادگیری ماشین است. ما در فصل ۱۰ راه‌هایی را برای بررسی تئوری و تجربی این مبادله مورد بحث قرار می‌دهیم.

فصل ۶

اصول بهینه‌سازی پیشرفته‌تر

با توجه به پیشینه بهینه‌سازی در فصل ۲، و مشاهده چگونگی مفید بودن آن در فصل‌های بعدی، اکنون می‌توانیم به رویکردهای بهینه‌سازی پیشرفته‌تر روی آوریم. اکنون به طور عمیق‌تر بحث خواهیم کرد که چگونه به روزرسانی مرتبه دوم گرادیان کاهشی را برای موارد چند متغیره بدست آوریم. سپس برخی از پیشرفت‌های محاسباتی در این روش‌ها را مورد بحث قرار می‌دهیم، به‌ویژه از طریق استفاده از تکنیک‌های انتخاب اندازه گام بهبودیافته، با استفاده از گرادیان کاهشی تصادفی و برخی تغییرات کوچک برای مقابله با نقاط غیر قابل تمایز. در نهایت، ما همچنین برخی از اصول اولیه را در مورد بهینه‌سازی محدود ارائه خواهیم کرد. هنگام حرکت به حالت چند متغیره، عادت کردن به حساب چند متغیره مفید خواهد بود. ما برخی از قوانین اساسی را در بخش B.1 ارائه می‌کنیم. مرجع کامل‌تری برای این قوانین را می‌توان در کتابچه راهنمای ماتریسی (بسیار مفید) [۱۶] یافت.

۱-۶ گرادیان کاهشی در توابع چند متغیره

می‌توانیم بحث در مورد به‌روزرسانی گرادیان کاهشی در بخش ۲,۲ را از حالت تک متغیره به حالت چند متغیره با استفاده از تقریب سری تیلور چند متغیره تعمیم دهیم. تقریب تیلور مرتبه دوم برای یک تابع حقیقی از چندین متغیر می‌تواند به این صورت نوشته شود.

$$c(\boldsymbol{w}) \approx \hat{\mathrm{c}}(\boldsymbol{w}) = c(\boldsymbol{w}_0) + \nabla c(\boldsymbol{w}_0)^T(\boldsymbol{w} - \boldsymbol{w}_0) + \frac{1}{2}(\boldsymbol{w} - \boldsymbol{w}_0)^T H_{c(\boldsymbol{w}_0)}(\boldsymbol{w} - \boldsymbol{w}_0)$$

که

$$\nabla c(\boldsymbol{w}_0) = \left(\frac{\partial \mathrm{c}}{\partial w_1}(\boldsymbol{w}_0), \frac{\partial \mathrm{c}}{\partial w_2}(\boldsymbol{w}_0), \ldots, \frac{\partial \mathrm{c}}{\partial w_d}(\boldsymbol{w}_0)\right) \qquad \in \mathbb{R}^d$$

گرادیان تابع c است که در $\mathbf{w}_0$ و ارزیابی می‌شود

$$\mathbf{H}_{c(w0)} = \begin{bmatrix} \frac{\partial^2 c}{\partial w_1^2}(\mathbf{w}_0) & \frac{\partial^2 c}{\partial w_1 \partial w_2}(\mathbf{w}_0) & \cdots & \frac{\partial^2 c}{\partial w_1 \partial w_2}(\boldsymbol{w}_0) \\ \vdots & \frac{\partial^2 c}{\partial w_2^2}(\boldsymbol{w}_0) & & \\ \vdots & & \ddots & \\ \frac{\partial^2 c}{\partial w_{\mathbf{d}} \partial w_1}(\boldsymbol{w}_0) & \cdots & & \frac{\partial^2 c}{\partial w_{\mathrm{d}}^2}(\boldsymbol{w}_0) \end{bmatrix} \in \mathbb{R}^{\mathrm{d}\times\mathrm{d}}$$

ماتریس هسین[1] تابع c است که در $\mathbf{w}_0$ ارزیابی شده است. در بخش بعدی مقداری شهود برای هسین ارائه می‌کنیم، اما در اینجا می‌توان آن را به طور شهودی مشابه مشتق دوم در نظر گرفت. مانند مشتق دوم، اطلاعاتی در مورد انحنای تابع ارائه می‌دهد، و بنابراین اطلاعات مفیدی در مورد میزان گام برداشتن در جهت گرادیان برای هر w_i ارائه می‌دهد.

به عنوان یادآوری در مورد ضرب ماتریس-بردار، $\mathbf{H}$ حاصل ضرب یک ماتریس $d \times d$ و بردار $d \times 1$ بردار $\mathbf{w}$ یک بردار $d \times 1$ $\mathbf{Hw}$ است. سپس، گرفتن $\mathbf{w}^{\mathrm{T}}\mathbf{Hw}$ حاصل ضرب نقطه‌ای بین $\boldsymbol{w}$ بردار $1 \times d$ و $\mathrm{d} \times 1$ بردار $\mathbf{Hw}$ است که منجر به یک اسکالر می‌شود. برای ضرب ماتریس بردار داریم

$$\mathbf{Hw} = \begin{bmatrix} \boldsymbol{H}_{1:} \\ \boldsymbol{H}_{2:} \\ \vdots \\ \boldsymbol{H}_{d:} \end{bmatrix} \mathbf{w} = \begin{bmatrix} \boldsymbol{H}_{1:}\boldsymbol{w} \\ \boldsymbol{H}_{2:}\boldsymbol{w} \\ \vdots \\ \boldsymbol{H}_{d:}\boldsymbol{w} \end{bmatrix} = \begin{bmatrix} < \boldsymbol{H}_{1:}, \boldsymbol{w} > \\ < \boldsymbol{H}_{2:}, \boldsymbol{w} > \\ \vdots \\ < \boldsymbol{H}_{d:}, \boldsymbol{w} > \end{bmatrix}$$

هنگام انجام ضرب ماتریس-بردار، فقط می‌توانید تصور کنید که بردار w به طرفین بچرخد و هر ردیف $\mathbf{H}$ را ضرب کند. برای ضرب ماتریس-ماتریس، $\mathbf{AB}$، باید اطمینان حاصل کنید که بعد دوم ماتریس $\mathbf{A}$ برابر با بعد اول ماتریس $\mathbf{B}$ است. -ضرب ماتریس‌ها-بردار برای هر ستون $\mathbf{B}$ تجزیه می‌شود.

مانند قبل، برای دریافت به روزرسانی افزایشی، می‌توانیم گرادیان این تقریب را گرفته و نقطه ثابت (محلی) را بدست آوریم. با استفاده از قوانین اساسی خلاصه شده در زیر در بخش B.1، گرادیان $\hat{c}(\boldsymbol{w})$ برابر است با

$$\nabla \hat{\mathrm{c}}(\mathbf{w}) = \nabla \mathrm{c}(\mathbf{w}_0) + \mathbf{H}_{\mathrm{c}(\mathbf{w}_0)}(\mathbf{w} - \mathbf{w}_0)$$

باز هم می‌خواهیم $\mathbf{w}_1$ را به گونه‌ای پیدا کنیم که این گرادیان صفر باشد. اگر هنوز با معکوس یک ماتریس آشنا نیستید، در بخش‌های بعدی این یادداشت‌ها (به ویژه برای رگرسیون خطی در فصل ۵) بیشتر مورد بحث قرار خواهد گرفت. در حال حاضر، برای حل $\mathrm{H}_{\mathrm{c}(\mathbf{w}_0)}(\mathbf{w} - \mathbf{w}_0) = -\nabla \mathrm{c}(\mathbf{w}_0)$، می‌توان معکوس $H_{c(\boldsymbol{w}_0)}^{-1}$ را محاسبه کرد و هر دو طرف معادله را در این معکوس ضرب کرد. این کار دوباره مشابه معکوس یک اسکالر است: $h^{-1}\ h\ =\ 1$. به روزرسانی چند متغیره مربوطه، که فراتر از معادله (۲,۱) برای حالت اسکالر گسترش یافته است،

$$\mathbf{w}_{\mathrm{i}+1} = \mathbf{w}_{\mathrm{i}} - \mathrm{H}_{\mathrm{c}(\mathbf{w}_{\mathrm{i}})}{}^{-1}\, \nabla c(\mathbf{w}_{\mathrm{i}}) \qquad (6.1)$$

در معادله ۶,۱، هم گرادیان و هم هسین در نقطه $\mathbf{w}_{\mathrm{i}}$ ارزیابی می‌شوند

اندازه هسین‌باعث می‌شود که انتخاب بین گرادیان کاهشی مرتبه اول و مرتبه دوم در حالت چند متغیره کمتر آشکار شود. بر خلاف مجموعه اسکالر، محاسبه خود هسین سنگین است (در اندازه $\mathbf{w}$ درجه دوم) و محاسبه معکوس هسین حتی سنگین‌تر است. به همین دلیل، به‌روزرسانی‌های مرتبه اول سبک‌تر اغلب ترجیح داده می‌شوند. برای مثال، اگر محاسبه هسین مانند هدف

[1] Hessian

رگرسیون خطی هزینه $O(d^2n)$ داشته باشد، پیچیدگی محاسباتی گرادیان کاهشی مرتبه دوم $O(d^3 + d^2n)$ در هر تکرار است، با فرض زمان $O(d^3)$برای یافتن معکوس‌های ماتریس. از سوی دیگر، دوباره برای رگرسیون خطی، پیچیدگی محاسباتی برای گرادیان کاهشی مرتبه اول فقط $O(dn)$ در هر تکرار است.

به‌روزرسانی مرتبه اول برای حالت چند متغیره تقریبی حتی بزرگ‌تر است، زیرا کل هسین با یک عدد اسکالر $\frac{1}{\eta}$ تقریب می‌یابد (که تقریب هسین را به یک ماتریس مورب با $\frac{1}{\eta}$ در قطر تبدیل می‌کند). سپس گرادیان تقریب مرتبه اول تبدیل می‌شود به

$$\nabla\hat{c}(w) = \nabla c(w_0) + \frac{1}{\eta}(w - w_0)$$

و در نتیجه به روزرسانی مرتبه اول

$$w_{i+1} = w_i - \eta_i \nabla c(w_i)$$

انتخاب این اندازه گام یک ملاحظه مهم است. ما قبلاً یک استراتژی اساسی را برای انتخاب اندازه گام مورد بحث قرار داده‌ایم. در بخش ۶،۵، چند مورد دیگر را مورد بحث قرار می‌دهیم.

۶-۲ خواص هسین

مانند مشتق دوم، هسین انحنای تابع را در نقطه w_0 منعکس می‌کند. هر ورودی نشان می‌دهد که چگونه مشتق جزئی برای w_j با تغییر w_i تغییر می‌کند.

برای شهود بیشتر، مشتق جهت را در نظر بگیرید. مشتق جهتی نشان می‌دهد که چگونه یک تابع (چند متغیری) با گام برداشتن مقدار کوچک t در یک جهت ثابت $\boldsymbol{u}$ تغییر می‌کند.

$$\lim_{t\to 0} \frac{c(\mathbf{w} + t\mathbf{u}) - c(\mathbf{w})}{t}$$

هنگامی‌که خود را محدود به تغییر تابع در این یک جهت می‌کنیم، تصور آن آسان‌تر است و به ما اجازه می‌دهد از قوانین مشتق دوم آشنا برای تنظیم تک متغیره استفاده کنیم.

$$\mathbf{w}(t) = \mathbf{w} + t\mathbf{u}$$

$$g(t) = c(\mathbf{w}(t))$$

ما می‌توانیم از قانون زنجیره‌ای در $g(t)$ برای محاسبه مشتق با توجه به t استفاده کنیم.

$$\acute{g}(t) = \nabla c(\mathbf{w}(t))^T \frac{\partial(\mathbf{w}(t))}{\partial t}$$

$$= \nabla c\big(\mathbf{w}(t)\big)^{\mathrm{T}}\mathbf{u}$$

$$\acute{g}(0) = \nabla c(\mathbf{w}(t))^T \boldsymbol{u}$$

$$= \nabla c(\mathbf{w})\text{^T}\,\mathbf{u} = 0$$

جایی که آخرین تساوی رخ می‌دهد چون $\mathbf{w}$ یک نقطه ثابت است و بنابراین $\nabla c(\boldsymbol{w}) = 0$ مشتق دوم است

$$\acute{g}(t) = \frac{\partial(\mathbf{w}(t))^T}{\partial t}\mathbf{H}_{c(w(t))}\frac{\partial(\mathbf{w}(t))}{\partial t}$$

$$= \mathbf{u}^T\mathbf{H}_{c(w(t))}\mathbf{u}$$

$$\acute{g}(0) = \mathbf{u}^T\mathbf{H}_{c(w)}\mathbf{u}$$

برای اینکه این نقطه ثابت $\mathbf{w}$ (مطابق با $t = 0$) مینیمم محلی باشد، $\acute{g}(0)$ باید آزمون مشتق دوم را برآورده کند: $\acute{g}(0) > 0$. این آزمون تنها در صورتی انجام می‌شود که $\boldsymbol{H}_{c(w)}$ باشد. قطعی مثبت، با تعریف ماتریس قطعی مثبت. به یاد بیاورید که یک ماتریس قطعی مثبت H همانی است که با توجه به $u \neq 0$، $\boldsymbol{u^T H u} > 0$، یا معادل آن، همه مقادیر ویژه بزرگتر از صفر باشد. از آنجایی که u یک جهت دلخواه دور از w بود، هسین باید مثبت- معین باشد تا اطمینان حاصل شود که $\acute{g}(0) > 0$ برای همه $u \neq 0$.

بنابراین، مقادیر ویژه هسین، انحنای تابع را به صورت محلی منعکس می‌کند. اگر $H_{c(w)}$ مقدار ویژه λ بسیار کوچکی داشته باشد، آنگاه بردار ویژه مربوطه u-که $\boldsymbol{H}_{c(w)}\boldsymbol{u} = \lambda\boldsymbol{u}$ را برآورده می‌کند- جهتی دورتر از $\boldsymbol{w}$ جایی است که تابع تقریباً مسطح است. این به این دلیل است که $\acute{g}(0) = \boldsymbol{u^T H}_{c(w)}\boldsymbol{u} = \lambda||\boldsymbol{u}||_2^2 = \lambda$ بسیار کوچک است.

مثال ۱۶: اکنون می‌توانیم هسین $H_{c(w)}$ را برای حل رگرسیون خطی در نظر بگیریم. این هسین ما را قادر می‌سازد تا بررسی کنیم که آیا واقعاً یک مینیمم محلی پیدا کرده‌ایم یا نه، اگر در عوض یک نقطه ثابت پیدا کرده‌ایم که ماکسیمم محلی یا یک نقطه زینتی است. هسین است

$$\mathbf{H}_{c(w)} = 2\mathbf{X}^T\mathbf{X}$$

این هسین ماتریس نیمه معین مثبت است. برای اینکه ببینید چرا، برای هر بردار $w \neq 0$ در نظر بگیرید،

$$\mathbf{w}^T\mathbf{X}^T\mathbf{X}\mathbf{w} = (\mathbf{X}\mathbf{w})^T\mathbf{X}\mathbf{w} = ||\mathbf{X}\mathbf{w}||_2^2 \geq 0$$

که در آن فقط می‌تواند برابری اتفاق بیفتد - برای برخی از $\boldsymbol{w}$ - اگر ستون‌های $\boldsymbol{X}$ به صورت خطی وابسته باشند. از آنجایی که هسین برای هر $\boldsymbol{w}$ نیمه معین است، این امر تحدب $\boldsymbol{c(w)}$ را تأیید می‌کند. علاوه بر این، اگر ستون‌های $\boldsymbol{x}$ به صورت خطی مستقل باشند، هسین مثبت قطعی است، که نشان می‌دهد مینیمم مطلق منحصر به فرد است.

۳-۶ مدیریت مجموعه داده‌های بزرگ

یکی از رویکردهای رایج برای مدیریت مجموعه داده‌های بزرگ استفاده از تقریب تصادفی است که در آن نمونه‌ها به صورت تدریجی پردازش می‌شوند. برای اینکه ببینیم چگونه این کار انجام می‌شود، اجازه دهید گرادیان تابع هدف، $\nabla c(\mathbf{w})$ را دوباره بررسی کنیم. ما یک محلول شکل بسته برای $\nabla c(\mathbf{w}) = 0$ به دست آوردیم. با این حال، برای بسیاری از توابع هدف دیگر، حل $\nabla c(\mathbf{w}) = 0$ به صورت بسته امکان پذیر نیست. در عوض، از مقدار اولیه $\mathbf{w}_0$ (معمولاً تصادفی) شروع می‌کنیم و سپس در جهت منفی گرادیان قدم می‌گذاریم تا به مینیمم محلی برسیم. این رویکرد شیب نزولی نامیده می‌شود و در الگوریتم ۲ خلاصه می‌شود. توجه کنید که در اینجا گرادیان با تعداد نمونه‌های n نرمال می‌شود، زیرا $\mathbf{X}^T(\mathbf{X}\mathbf{w} - \mathbf{y})$ با تعداد نمونه‌ها رشد می‌کند و انتخاب اندازه مراحل را دشوارتر می‌کند.

Algorithm 2: Batch Gradient Descent(c, X, y)

1: $// \ A \ non-optimized, basic \ implementation \ of \ batch \ gradient \ descent$
2: $\boldsymbol{w} \leftarrow \ random \ vector \ in \ \mathbb{R}^d$
3: $err \leftarrow \infty$
4: $tolerance \leftarrow 10e^{-4}$
5: $max \ iterations \leftarrow 10e^5$
6: $\boldsymbol{while} |c(\boldsymbol{w}) - err| > tolerance and have not reached max iterations \boldsymbol{do}$
7: $err \leftarrow c(\boldsymbol{w})$ $\quad \triangleright \ for \ linear \ regression, c(\boldsymbol{w}) = \frac{1}{2n} \left|\left|\boldsymbol{X}\boldsymbol{w} - \boldsymbol{y}\right|\right|_2^2$
8: $g \leftarrow \nabla c(\boldsymbol{w})$ $\quad \triangleright \ for \ linear \ regression, \nabla c(\boldsymbol{w}) = \frac{1}{n} \boldsymbol{X}^T(\boldsymbol{X}\boldsymbol{w} - \boldsymbol{y})$
9: $// \ The \ step-size \ \eta \ could \ be \ chosen \ by \ line-search, as \ in \ Algorithm \ 1$
10: $\eta \leftarrow linesearch(\boldsymbol{w}, c, g)$
11: $\boldsymbol{w} \leftarrow \boldsymbol{w} - \eta \boldsymbol{g}$
12: $return \ \boldsymbol{w}$

با این حال، برای تعداد زیادی نمونه n، محاسبه گرادیان در تمام نمونه‌ها می‌تواند سنگین یا غیرممکن باشد. یک جایگزین این است که گرادیان را با دقت کمتری با نمونه‌های کمتر تقریبی کنیم. در تقریب تصادفی، ما معمولاً شیب را با یک نمونه تقریب می‌زنیم، مانند الگوریتم ۳. اگرچه این رویکرد ممکن است بیش از حد تقریبی به نظر برسد، یک تاریخچه نظری و تجربی طولانی وجود دارد که اثربخشی آن را نشان می‌دهد (برای مثال [۶، ۵ را ببینید]). با افزایش روزافزون اندازه مجموعه داده‌ها برای بسیاری از سناریوها، کلیت تقریب تصادفی آن را می‌توان به روشی مدرن برای برخورد با داده‌های بزرگ تبدیل کرد. برای سناریوهای تخصصی، البته رویکردهای دیگری نیز وجود دارد. برای مثال، [۱۷] را ببینید.

الگوریتم آموزشی برای گرادیان کاهشی تصادفی اکنون می‌تواند مورد بازنگری قرار گیرد تا به طور تصادفی یک نقطه داده را در هر زمان از $\mathcal{D}$ ترسیم کند و سپس وزن‌های فعلی را با استفاده از معادله قبلی به روز کند. به طور معمول، در عمل، این مستلزم تکرار یک یا چند بار در مجموعه داده به ترتیب (با فرض تصادفی بودن، با نمونه‌های $\mathrm{i.}\, i.\, d.$ است. هر تکرار بر روی مجموعه داده یک دوره نامیده می‌شود. شرایط همگرایی معمولاً شامل شرایط اندازه گام‌ها می‌شود که نیاز به کاهش آنها در طول زمان دارد. مانند نزول شیب دسته ای، این به روزرسانی‌های شیب نزولی تصادفی همگرا می‌شوند، هرچند با نوسان بیشتر حول بردار وزن حقیقی، با کاهش اندازه گام به تدریج این نوسانات را هموار می‌کند.

Algorithm 3: $Stochastic \ Gradient \ Descent(c, \boldsymbol{X}, y)$

1: $\boldsymbol{w} \leftarrow \ random \ vector \ in \ \mathbb{R}^d$
2: $\boldsymbol{for} \ i = 1, \ldots number \ of \ epochs \ \boldsymbol{do}$
3: $\quad Shuffle \ data \ points \ from \ 1, \ldots, n$
4: $\quad \boldsymbol{for} \ j = 1, \ldots, n \ \boldsymbol{do}$
5: $\quad\quad \boldsymbol{g} \leftarrow \nabla c_j(\boldsymbol{w})$ $\quad \triangleright \ for \ linear \ regression, \nabla c_{j(\boldsymbol{w})} = \left(x_j^T \boldsymbol{w} - \boldsymbol{y}_j\right)\boldsymbol{x}_j$
6: $\quad\quad // \ For \ convergence, the \ step-size \ \eta t \ needs \ to \ decrease \ with \ time, such \ as$
7: $\quad\quad //\eta_t = \eta_0 t^{-1/2} \quad or \ \eta_t = \eta_0 i^{-1} \ for \ an \ initial \ \eta_0 (e.g., \eta_0 = 1.0).$
8: $\quad\quad // \ In \ practice, it \ is \ common \ to \ pick \ a \ fixed, small \ stepsize$
9: $\quad\quad \eta_t \leftarrow i^{-1}$
10: $\quad\quad \boldsymbol{w} \leftarrow \boldsymbol{w} - \eta_t g$
11: $\boldsymbol{return} \ \boldsymbol{w}$

۶-۴ بهینه‌سازی غیر هموار اما همچنان مستمر

ما در سراسر این یادداشت‌ها فرض می‌کنیم که اهداف ما مستمر هستند. با این حال، این به معنای هموار بودن آنها نیست: در برخی موارد، این اهداف پیوسته ممکن است دارای نقاط غیر قابل تمایز باشند. به عنوان مثال، تنظیم کننده ℓ_1 در 0 غیر قابل تمایز است، و $||\boldsymbol{Xw}-\boldsymbol{y}||_2^2+\lambda||\boldsymbol{w}||_1$ را غیر قابل تمایز می‌کند. یک استراتژی استفاده از زیر گرادیان کاهشی است. این به معنای انتخاب یک انتخاب معقول برای گرادیان در نقطه غیر قابل تمایز است. برای مثال، در اینجا، می‌توانیم مشتق جزئی ℓ_1 برای w_j را صفر در صفر، -1 برای $w_j>0$ و 1 برای $w_j>0$ در نظر بگیریم. متأسفانه، این کاهشی است زیرا تمایل به پرش در اطراف صفر وجود دارد. برخلاف ℓ_2، گرادیان به تدریج نزدیک به صفر کاهش نمی‌یابد و به آرامی w_j را کاهش می‌دهد، بلکه بین دو مقدار بزرگ -1 و 1 می‌پرد. با چنین گرادیان بزرگی، کاهش تدریجی w_j به صفر دشوار است، حتی اگر این راه‌حل بهینه باشد.

یک جایگزین برای چنین اهداف غیر هموار، استفاده از روش‌های پروگزیمال است. ایده ساده است: از شیب نزول برای مولفه هموار بهینه‌سازی (اصطلاح خطای $||\boldsymbol{Xw}-\boldsymbol{y}||_2^2$) استفاده کنید، و سپس برای مقادیر $\boldsymbol{w}$ که نزدیک به صفر هستند، آنها را روی صفر قرار دهید. این ایده آستانه‌سازی، اگرچه ساده است، اما از لحاظ نظری یک رویکرد صحیح برای بهینه‌سازی با «۱» غیر هموار است. این عملگر آستانه‌ای، عملگر پروگزیمال[1] نامیده می‌شود و می‌توان آن را به عنوان یک عملگر پروجکشن[2] دید. هر بار که $\boldsymbol{w}$ با گرادیان به روز می‌شود، آن را از یک راه‌حل پراکنده دور می‌کند. سپس عملگر پروگزیمال $\mathbf{w}$ را به فضای راه‌حل‌های پراکنده باز می‌گرداند. عملگر پروگزیمال برای ℓ_1 از نظر عنصر به $\boldsymbol{w}$ اعمال می‌شود، و بنابراین در هر $\boldsymbol{w_i}$ به عنوان، با اندازه گام η و پارامتر تنظیم λ، تعریف می‌شود.

$$\text{prox}_{\eta\lambda\ell_1}(\mathbf{w_i})=\begin{cases}\mathbf{w_i}-\eta\lambda & \text{if}\mathbf{w_i}>\eta\lambda\\ 0 & \text{if}|\boldsymbol{w_i}|\leq\eta\lambda\\ \mathbf{w_i}+\eta\lambda & \text{if}\mathbf{w_i}<-\eta\lambda\end{cases}$$

عملگر پروگزیمال در کل بردار $\mathbf{w}$ از نظر عنصر اینگونه تعریف می‌شود: $\text{prox}_{\eta\lambda\ell_1}(\mathbf{w})=\left[\text{prox}_{\eta\lambda\ell_1}(\mathbf{w_1}),\ldots,\text{prox}_{\eta\lambda\ell_1}(\mathbf{w_d})\right]$. به خوبی، این تئوری بیان می‌کند که اندازه گام نباید بزرگ‌تر از معکوس ثابت لیپشیتز[3] برای بخش هموار هدف باشد، جایی که به طور شهودی ثابت‌های لیپشیتز نشان‌دهنده سرعت تغییر تابع است. در الگوریتم ۴، ما یک الگوریتم گرادیان کاهشی برای به روزرسانی افزایشی با منظم کننده ℓ_1 ارائه می‌دهیم که به عنوان الگوریتمی به نام $ISTA$ [۴] معرفی شده است. به طور کلی‌تر، روش‌های پروگزیمال برای اهداف غیر هموار دیگر استفاده می‌شود، اگرچه در این یادداشت‌ها ما فقط Lasso را در نظر می‌گیریم.

***Algorithm* 4**: *Batch gradient descent for ℓ_1 regularized linear regression* $(\boldsymbol{X}, y, \lambda)$

$\boldsymbol{w} \leftarrow 0 \in \mathbb{R}^d$
$err \leftarrow \infty$
$tolerance \leftarrow 10e^{-4}$
// *Precomputing these matrices, to avoid recomputing them in the loop*
$XX \leftarrow \frac{1}{n}\boldsymbol{X}^T\boldsymbol{X}$

---

[1] proximal
[2] projection
[3] Lipschitz

$\boldsymbol{Xy} \leftarrow \frac{1}{n}\boldsymbol{X}^T\boldsymbol{y}$
// $This\ stepsize\ is\ specific\ to\ the\ least-squares\ loss\ for\ linear\ regression$
$\eta \leftarrow 1/\left(2||\boldsymbol{XX}||_F\right)$
while $|c(\boldsymbol{w}) - err| > tolerance\ and\ have\ not\ reached\ max\ iterations$ **do**
    $err \leftarrow c(\boldsymbol{w})$
    // $Proximal\ operator\ projects\ back\ into\ the\ space\ of\ sparse\ solutions\ given\ by\ \ell_1$
    $w \leftarrow prox_{\eta\lambda\ell_1}(\boldsymbol{w} - \eta\boldsymbol{XXw} + \boldsymbol{\eta Xy})$
return $\boldsymbol{w}$

۶-۵ روش‌های بیشتر برای انتخاب اندازه گام‌ها

از آنجایی که انتخاب اندازه گام بخش مهمی از یک الگوریتم فرود موثر است، راه‌های زیادی برای انجام این کار وجود دارد. علاوه بر جستجوی خط، یکی از رایج‌ترین روش‌ها استفاده از روش‌های شبه مرتبه دوم (یا شبه نیوتن) است. همانطور که دیدیم، معکوس هسین راه خوبی برای انتخاب اندازه گام ارائه می‌دهد، اما معمولاً برای محاسبه بسیار سنگین است چه برسد به معکوس کردن. روش‌های شبه مرتبه دوم تقریباً هسین را با کمترین میزان ذخیره‌سازی و محاسبات ممکن انجام می‌دهند. یکی از ساده‌ترین این تقریب‌ها تقریب فقط قطر هسین و معکوس کردن آن است که فقط برای محاسبه $O(d)$ و فضا هزینه دارد. چنین تقریبی معمولاً حتی برای مورب هسین معکوس بسیار ضعیف است و بنابراین معمولاً استفاده نمی‌شود. در عوض، محبوب‌ترین روش‌ها عبارتند از LBFGS [14]، Adadelta [21] و Adam [11].

فصل ۷

مدل‌های خطی تعمیم یافته[1]

در بخش‌های قبلی، دیدیم که چارچوب آماری بینش‌های ارزشمندی را در مورد رگرسیون خطی ارائه می‌کند، به‌ویژه با توجه به بیان صریح بیشتر مفروضات در سیستم (تصویر کامل را تنها زمانی خواهیم دید که از فرمول بیزی استفاده شود). این مفروضات برای برآورد دقیق پارامترهای مدل ضروری بودند، که سپس می‌توان از آن برای پیش‌بینی نقاط داده‌ای که قبلاً دیده نشده بود استفاده کرد.

در این بخش، مدل‌های خطی تعمیم‌یافته (GLM) را معرفی می‌کنیم که رگرسیون مینیمم مربعات معمولی را فراتر از توزیع‌های احتمال گاوسی و وابستگی‌های خطی بین ویژگی‌ها و هدف گسترش می‌دهند. این تعمیم همچنین شما را با طیف وسیع‌تری از توابع از دست دادن، به نام واگرایی برگمن[2] آشنا می‌کند.

ابتدا نکات اصلی رگرسیون مینیمم مربعات معمولی را مرور خواهیم کرد. در آنجا، ما فرض کردیم که مجموعه‌ای از $i.i.d.$ نقاط داده با اهداف خود $\mathcal{D} = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^{n}$ بر اساس توزیع $p(x, y)$ ترسیم شدند. ما همچنین فرض کردیم که یک رابطه اساسی بین ویژگی‌ها و هدف خطی است، یعنی

$$\boldsymbol{Y} = \sum_{j=0}^{d} \omega_j \boldsymbol{X}_j + \varepsilon$$

که در آن ω مجموعه‌ای از وزن‌های مجهول بود و ε یک متغیر تصادفی با میانگین صفر با واریانس σ^2 بود. به منظور ساده‌سازی تعمیم، این مدل را کمی دوباره فرموله می‌کنیم. به ویژه، جدا کردن رابطه خطی اساسی بین ویژگی‌ها و هدف از این واقعیت که Y به طور معمول توزیع شده است مفید خواهد بود. یعنی خواهیم نوشت

$$1.\ \mathbb{E}[y|\boldsymbol{x}] = \boldsymbol{\omega}^{\mathrm{T}}\mathbf{x}$$

$$2.\ \mathrm{p}(\mathrm{y}|\mathbf{x}) = \mathcal{N}(\mu, \sigma^2)$$

با $\mu = \boldsymbol{\omega}^T \boldsymbol{x}$ دو عبارت را به هم متصل می‌کند. این روش فرمول‌بندی رگرسیون خطی به ما امکان می‌دهد (1) چارچوب را به روابط غیرخطی بین ویژگی‌ها و هدف تعمیم دهیم و همچنین (2) از توزیع‌های خطا به غیر از گاوسی استفاده کنیم.

---

[1] Generalized Linear Models

[2] Bregman divergences

۱-۷ انتقال نمایی و توزیع پواسون

ابتدا با مثالی از GLM شروع می‌کنیم، قبل از اینکه به کلاس عمومی و تعریف کلی برویم. فرض کنید که نقاط داده مطابق با شهرهای جهان است که با برخی از ویژگی‌های عددی توصیف شده‌اند - و متغیر هدف تعداد روزهای آفتابی مشاهده شده در یک سال خاص است. متغیر هدف y با توجه به ویژگی‌های x ممکن است شبیه توزیع پواسون باشد. بنابراین طبیعی‌تر است که $p(y|\boldsymbol{x}) = Poisson(\lambda)$ را مدل کنیم، که در آن $\lambda > 0$ پارامتر (میانگین) توزیع پواسون است: $\mathbb{E}[\mathbf{y}|\mathbf{x}] = \lambda$. با این حال، چون $\lambda \in \mathbb{R}^+$، مدل کردن λ با $\omega^{\mathrm{T}}\mathrm{x} \in \mathbb{R}$ مناسب نخواهد بود. بلکه می‌خواهیم پیش‌بینی خطی خود را با تابع f انتقال دهیم تا محدوده ترکیب خطی ویژگی‌ها را به دامنه پارامترهای توزیع احتمال تنظیم کنیم.

ما می‌توانیم این کار را با معرفی یک انتقال نمایی برای این توزیع پواسون و به طور کلی‌تر، بعداً هر تابع انتقال معکوس f انجام دهیم. اگر بجای آن بتوانیم ω را به گونه‌ای تخمین بزنیم که $\lambda = \mathrm{e}^{\boldsymbol{\omega}^T\boldsymbol{x}}$، آنگاه می‌توانیم تضمین کنیم که تخمین‌هایمان در محدوده صحیح هستند. از طرف دیگر، می‌توان در نظر گرفت که ما در حال یادگیری یک وزن دهی خطی از ویژگی‌ها برای یادگیری یک پارامتر تبدیل شده، $log(\lambda) = \boldsymbol{\omega}^T\boldsymbol{x}$ هستیم. این اصلاح ساده به همین دلیل است که این مدل‌ها را مدل‌های خطی کلی‌شده می‌نامند، زیرا مؤلفه اصلی وزن‌دهی خطی است. ما انواع توزیع‌ها و انتقال‌هایی را که می‌توان در بخش‌های زیر در نظر گرفت، فرمول‌بندی می‌کنیم، اما ابتدا این مثال را با رگرسیون پواسون برای ارائه یک مثال ملموس به پایان می‌رسانیم.

برای ایجاد مدل GLM برای رگرسیون پواسون، (۱) یک انتقال نمایی بین انتظار هدف و ترکیب خطی ویژگی‌ها، و (۲) توزیع پواسون برای متغیر هدف را فرض می‌کنیم.

1. $\mathbb{E}[\mathbf{y}|\mathbf{x}] = \exp(\boldsymbol{\omega}^T\mathbf{x})\mathrm{or}\log(\mathbb{E}[\mathbf{y}|\mathbf{x}]) = \boldsymbol{\omega}^{\mathrm{T}}\mathbf{x}$
2. $\mathrm{p}(\mathrm{y}|\mathbf{x}) = \mathrm{Poisson}(\lambda)$

با استفاده از این واقعیت که $\mathbb{E}[\boldsymbol{y}|\boldsymbol{x}] = \lambda$، دو فرمول را با استفاده از $\lambda = e^{\omega^T x}$ به هم وصل می‌کنیم. توزیع احتمال اینگونه حاصل میشود

$$\mathrm{p}(\mathrm{y}|\mathrm{x}) = \frac{\mathrm{e}^{\omega^{\mathrm{T}}\mathrm{xy}} \cdot \mathrm{e}^{-\mathrm{e}^{\omega^{\mathrm{T}}\mathrm{x}}}}{y!} \qquad \text{for any } \mathrm{y} \in \mathbb{N}$$

برای یافتن پارامترهای مدل رگرسیون می‌توانیم از تخمین ماکسیمم درستنمایی استفاده کنیم. تابع $log - likelihood$ که به شکل

$$\mathrm{ll}(\mathbf{w}) = \sum_{\mathrm{i}=1}^{\mathrm{n}} \mathrm{ll}_{\mathrm{i}}(\mathbf{w})$$

$$\mathrm{ll}_{\mathrm{i}}(\mathbf{w}) = \mathbf{w}^{\mathrm{T}}\mathbf{x}_{\mathrm{i}}\mathrm{y}_{\mathrm{i}} - \mathrm{e}^{\mathbf{w}^{\mathrm{T}}\mathbf{x}_{\mathrm{i}}} - \mathrm{lny}_{\mathrm{i}}!$$

هدف ما به مقدار مینیمم رساندن احتمال ورود به سیستم منفی است: $min_{\boldsymbol{w}} - ll(\boldsymbol{w})$. به راحتی می‌توان فهمید که $\nabla ll(\boldsymbol{w}) = 0$ راه‌حل یک شکل بسته ندارد. بنابراین، برخلاف رگرسیون خطی، باید از گرادیان کاهشی استفاده کنیم. می‌توانیم از شیب نزول مرتبه اول یا دوم و گرادیان کاهشی دسته‌ای یا تصادفی استفاده کنیم. مرحله کلیدی در هر یک از اینها این است که ابتدا گرادیان را برای یک نمونه محاسبه کنید. ما با استخراج مشتق جزئی لگاریتم احتمال منفی برای یک نمونه شروع می‌کنیم

$$-\frac{\partial \mathrm{ll_i}(\mathbf{w})}{\partial w_\mathrm{j}} = \mathrm{e}^{\mathbf{w}^\mathrm{T}\mathbf{x_i}}\mathrm{x_{ij}} - \mathrm{x_{ij}y_i}$$

$$= \mathrm{x_{ij}}\left(\mathrm{e}^{\mathrm{w}^\mathrm{T}\mathrm{x_i}} - \mathrm{y_i}\right)$$

گرادیان برای یک نمونه اینگونه است

$$-\nabla \mathrm{ll_i}(\mathbf{w}) = \mathrm{x_i} \cdot (\mathrm{p_i} - \mathrm{y_i})$$

که در آن $p_i = e^{w^T x_i}$ پیش‌بینی است. توجه داشته باشید که $\mathrm{p_i} - \mathrm{y_i}$ مربوط به یک خطای پیش‌بینی برای نمونه i است. گرادیان دسته‌ای است

$$-\nabla \mathrm{ll}(\mathbf{w}) = -\sum_{i=1}^{n} \nabla \mathrm{ll_i}(\mathbf{w})$$

$$= \sum_{i=1}^{n} \boldsymbol{x}_i(p_i - y_i)$$

$$= \mathbf{X}^\mathrm{T}(\mathbf{p} - \mathrm{y})$$

که در آن $\boldsymbol{p}$ یک بردار با عناصر $p_i = e^{w^T x_i}$ است، که در آن $\boldsymbol{p} - \boldsymbol{y}$ یک بردار خطا است.

معمولاً، اکنون فقط به صورت تصادفی یا شیب نزولی دسته‌ای انجام می‌شود. برای گرادیان کاهشی تصادفی، هر مرحله شامل استفاده از گرادیان برای یک نمونه (یعنی $-ll_i(wt)$) و برای شیب نزولی دسته‌ای، هر مرحله شامل استفاده از گرادیان برای همه نمونه‌ها (یعنی $-ll(w_t)$) است. علاوه بر این، می‌توانیم ماتریس هسین را در نظر بگیریم، هم برای ارزیابی ویژگی‌های نقاط ثابت و هم برای گرادیان کاهشی مرتبه دوم، اگرچه، اگر d بزرگ باشد، احتمالاً خیلی سنگین است. دومین مشتق جزئی تابع احتمال ورود به سیستم منفی برای یک نمونه است

$$-\frac{\partial^2 \mathrm{ll_i}(\mathbf{w})}{\partial w_\mathrm{j}\, \partial w_\mathrm{k}} = \mathrm{x_{ij}} \mathrm{e}^{\mathrm{w}^\mathrm{T}\mathrm{x_i}} \mathrm{x_{ik}}$$

$$= \mathrm{x_{ij}} \mathrm{p_i} \mathrm{x_{ik}}$$

با

$$-\frac{\partial^2 ll_i(\mathbf{w})}{\partial w_j \partial w_k} = \sum_{i=1}^{n} \frac{\partial^2 \mathrm{ll_i}(\mathbf{w})}{\partial w_j \partial w_k}$$

برای P یک ماتریس مورب $\mathrm{n} \times \mathrm{n}$ با p_i در قسمت مورب، ماتریس هسین اینگونه است

$$\mathbf{H}_{-\mathrm{ll}(\mathbf{w})} = \mathbf{X}^T \mathbf{P} \mathbf{X}$$

اگر $\mathbf{X}$ رتبه پایین نباشد، این ماتریس مثبت است، که به این معنی است که تنها یک نقطه ثابت وجود دارد و آن مینیمم مطلق است. در واقع، می‌دانیم که هدف رگرسیون پواسون محدب است، حتی اگر $\mathbf{X}$ رتبه کامل نباشد، و بنابراین همه نقاط ثابت مینیمم‌های مطلق هستند.

تمرین ۸: آیا اگر $\mathbf{X}$ رتبه کامل نباشد یک مینیمم مطلق وجود دارد؟

تمرین ۹: به روز رسانی مرتبه دوم برای رگرسیون پواسون چیست؟

۷-۲ توزیع‌های خانوادگی نمایی

در بخش قبل، از یک مثال خاص برای نشان دادن چگونگی تعمیم فراتر از توزیع‌های گاوسی استفاده کردیم. این رویکرد به طور کلی به هر خانواده توزیع نمایی گسترش می‌یابد. برای سادگی، در اینجا ما روی خانواده نمایی طبیعی تمرکز می‌کنیم که برای اکثر مدل‌های خطی تعمیم یافته کافی است. خانواده نمایی طبیعی کلاسی از توزیع‌های احتمال با شکل زیر است

$$p(x|\theta) = \exp(\theta x - a(\theta) + b(x))$$

در جایی که $\theta \in \mathbb{R}$ پارامتر توزیع است، $a : \mathbb{R} \to \mathbb{R}$ یک تابع نرمال ساز log است و $b: \mathbb{R} \to \mathbb{R}$ تابعی از x است که معمولاً در بهینه سازی ما نادیده گرفته می‌شود زیرا تابع θ نیست. بسیاری از توزیع‌ها (خانواده‌های) که اغلب با آن‌ها مواجه می‌شوند، اعضای خانواده نمایی هستند. به عنوان مثال، توزیع‌های نمایی، گاوسی، گاما، پواسون یا دوجمله‌ای. بنابراین، مطالعه کلی خانواده نمایی برای درک بهتر مشترکات و تفاوت‌های بین توابع تک تک اعضا مفید است.

مثال ۱۷: توزیع پواسون را می‌توان به صورت بیان کرد

$$p(x|\lambda) = \exp(x \log \lambda - \lambda - \log x!)$$

که $\lambda \in \mathbb{R}^+$ و $\mathcal{X} = \mathbb{N}_0$

اکنون اجازه دهید بینش بیشتری در مورد ویژگی‌های پارامترهای خانواده نمایی و اینکه چرا این کلاس برای تخمین راحت است، به دست آوریم. تابع $a(\theta)$ معمولاً تابع $\log - \text{partitioning}$ یا به سادگی یک $\log - \text{normalizer}$ نامیده می‌شود. به این دلیل نامیده می‌شود

$$a(\theta) = \log \int_{\mathcal{X}} \exp(\theta x + b(x)) dx$$

و بنابراین نقش اطمینان از داشتن چگالی معتبر را ایفا می‌کند: $\int_{\mathcal{X}} p(x)dx = 1$ نکته مهم این است که برای بسیاری از GLMهای رایج، مشتق a با تابع انتقال مطابقت دارد. برای مثال، برای رگرسیون پواسون، تابع انتقال $f(\theta) = exp(\theta)$، و مشتق e^{θ} برابر a است. بنابراین، همانطور که در پایین بحث می‌کنیم، $\log - \text{normalizer}$ برای یک خانواده نمایی نشان می‌دهد که چه انتقال f باید استفاده شود.

ویژگی‌های این $\log - \text{normalizer}$ برای تخمین مدل‌های خطی تعمیم‌یافته نیز کلیدی هستند. می‌توان آن را استخراج کرد که

$$\frac{\partial a(\theta)}{\partial \theta} = \mathbb{E}[X]$$

$$\frac{\partial^2 a(\theta)}{\partial \theta^2} = V[X]$$

۳-۷ فرمول‌بندی کردن مدل‌های خطی تعمیم یافته

اکنون مدل‌های خطی تعمیم یافته را فرمول‌بندی می‌کنیم. دو جزء کلیدی GLMها را می‌توان به صورت بیان کرد

1. $\mathbb{E}[\mathbf{y}|\mathbf{x}] = \mathrm{f}(\boldsymbol{\omega}^{\mathrm{T}}\mathbf{x})$ or $\mathrm{g}(\mathbb{E}[\mathbf{y}|\mathbf{x}]) = \boldsymbol{\omega}^{\mathrm{T}}\mathbf{x}$ where $\mathrm{g} = \mathrm{f}^{-1}$
2. $\mathrm{p}(\mathbf{y}|\mathbf{x})$ توزیع خانواده نمایی است

تابع f را تابع انتقال و g را تابع پیوند می‌نامند. برای رگرسیون پواسون، f تابع نمایی است، و همانطور که برای رگرسیون لجستیک[1] خواهیم دید، f تابع سیگموئید[2] است. تابع انتقال محدوده $\omega^T \mathrm{x}$ را به دامنه Y تنظیم می‌کند. به دلیل این رابطه، توابع پیوند معمولاً مستقل از توزیع Y انتخاب نمی‌شوند. تعمیم به خانواده نمایی از توزیع گاوسی که در رگرسیون مینیمم مربعات معمولی استفاده می‌شود، به ما امکان مدل‌سازی طیف وسیع‌تری از توابع هدف را می‌دهد. GLMها شامل سه مدل پرکاربرد می‌باشند: رگرسیون خطی، رگرسیون پواسون و رگرسیون لجستیک که در فصل بعدی در مورد طبقه‌بندی در مورد آنها صحبت خواهیم کرد.

برای ارتباط واضح‌تر اینها با توزیع‌های خانواده نمایی، باید توزیع‌های شرطی را در نظر بگیریم. هر $p(y|\boldsymbol{x})$ یک توزیع خانواده نمایی با پارامتر $\theta = \boldsymbol{x}^T \boldsymbol{w}$ است. هنگام یادگیری $\boldsymbol{w}$ - با به ماکسیمم رساندن احتمال - ما در حال یادگیری پارامتر θ_i برای هر نمونه هستیم (x_i, y_i). $\log - \text{likelihood}$ منفی کلی است

$$-ll(\mathbf{w}) = -log \prod_{i=1}^{n} e^{\theta_i y_i - a(\theta) + b(y_i)}$$

$$= -\sum_i \left(\theta_i y_i - a(\theta) + b(y_i)\right)$$

$$= -\sum_i \mathrm{ll\_i}(\mathbf{w})$$

با گرادیان‌های

$$-\frac{\partial ll_i(\mathbf{w})}{\partial w_j} = \frac{\partial a(\theta_i)}{\partial w_j} - \frac{\partial \theta_i}{\partial w_j y_i}$$

$$= \frac{\partial a(\theta_i)}{\partial \theta_i}\frac{\partial \theta_i}{\partial w_j} - \frac{\partial \theta_i}{\partial w_j} y_i$$

$$= \left(\frac{\partial a(\theta i)}{\partial \theta \mathrm{i}} - y_i\right)\frac{\partial \theta_i}{\partial w_j}$$

همانطور که برای رگرسیون پواسون واضح بود، هیچ تضمینی برای راه‌حل به شکل بسته برای w وجود ندارد. بنابراین، فرمول‌های GLM معمولاً از تکنیک‌های تکرار شونده مانند گرادیان کاهشی استفاده می‌کنند. از این رو، یک مکانیسم واحد را می‌توان برای طیف گسترده‌ای از توابع پیوند و توزیع احتمال، با استفاده از این گرادیان‌های بالا استفاده کرد.

---

[1] logistic
[2] sigmoid

این به روزرسانی را می‌توان با استفاده از رایج‌ترین تنظیمات برای GLMها ملموس‌تر کرد. نکته مهم این است که این تنظیم فقط به دانش تابع انتقال f نیاز دارد، بدون اینکه صریحاً نیاز به دانستن $\log - \text{normalized } a$ داشته باشد. این ساده سازی از ارتباط بین انتقال fو $log - normalizer\ a$ که در بالا اشاره شد ناشی می‌شود. ما بحث کردیم که تابع انتقال f برای منعکس کردن محدوده متغیر خروجی y انتخاب شده است. با این حال، انتخاب باید ویژگی‌های دیگری نیز داشته باشد. به طور خاص، ما می‌خواهیم اطمینان حاصل کنیم که g یک احتمال Log منفی محدب و هموار ارائه می‌کند تا بهینه‌سازی را ساده کنیم. به طور مفید، پارامتر a از توزیع خانواده نمایی دقیقاً چنین انتخابی را در اختیار ما قرار می‌دهد: $f = \nabla a$. از آنجایی که $\frac{\partial \theta_i}{\partial w_j} = x_{ij}$ برای $\theta_i = x_i^T w$، دریافت می‌کنیم که

$$-\frac{\partial ll_i(\mathbf{w})}{\partial w_j} = \left(\frac{\partial a(\theta i)}{\partial \theta \mathrm{i}} - y_i\right)\frac{\partial \theta_i}{\partial w_j}$$

$$= (f(\theta_i) - y_i)x_{ij}$$

$$= (f(\boldsymbol{x}_i^T \boldsymbol{w}) - y_i)x_{ij}$$

بنابراین، با توجه به انتقال مناسب f برای توزیع خانواده نمایی مورد نظر، به‌روزرسانی گرادیان کاهشی تصادفی به سادگی انجام می‌شود.

$$\mathbf{w}_{t+1} = \mathbf{w}_t - \eta_t\left(f\left(\mathbf{x}_i^T\mathbf{w}_t\right) - y_i\right)\mathbf{x}_i$$

و به روزرسانی گرادیان کاهشی دسته‌ای است

$$\mathbf{w}_{t+1} = \mathbf{w}_t - \eta_t \sum_{i=1}^{n}\left(f\left(\mathbf{x}_i^T\mathbf{w}_t\right) - y_i\right)\mathbf{x}_i$$

$$= \mathbf{w}_t - \eta_t \mathbf{X}^T(\mathbf{p} - \mathbf{y})$$

که در آن $\mathbf{p}_\mathbf{i} = \mathrm{f}\left(\mathrm{x}_\mathrm{i}^\mathrm{T}\mathbf{w}_\mathrm{t}\right)$. برای بررسی هسین، دومین مشتق جزئی تابع احتمال ورود به سیستم منفی برای یک نمونه است.

$$-\frac{\partial^2 ll_i(\mathbf{w})}{\partial w_j \partial w_k} = x_{ij}\frac{\partial f(\theta_i)}{\partial \theta_i}x_{ik}$$

برای $\boldsymbol{D}$ یک ماتریس مورب $n \times n$ با $\frac{\partial f(\theta_i)}{\partial \theta_i}$ در قطر، ماتریس هسی بنابراین

$$\mathbf{H}_{-\mathrm{ll(w)}} = -\mathbf{X}^\mathrm{T}\mathbf{D}\mathbf{X}. \qquad (7.3)$$

همانطور که در رگرسیون پواسون، این ماتریس تضمین شده است که مثبت نیمه معین است، و اگر $\mathbf{X}$ رتبه پایین نباشد، مثبت قطعی است.

نکته: تنظیم رایج $f = \nabla a$برای GLMها با اهداف پرکاربردی به نام واگرایی برگمن[1] ارتباط دارد. این واگرایی‌ها به صورت $D_a(\hat{y}||y)$ نوشته می‌شوند، که نشان دهنده تفاوت بین $\hat{y}$ و y است، جایی که واگرایی با a پارامتر می‌شود. به مینیمم رساندن این واگرایی برگمن با به مینیمم رساندن احتمال ورود به سیستم منفی خانواده نمایی طبیعی مربوطه مطابقت دارد:

[1] Bregman divergences

$$\underset{\theta}{\operatorname{argmin}}\ D_a(x||g(\theta)) = \underset{\theta}{\operatorname{argmin}}\ -\ln p(x|\theta)$$

برای جزئیات بیشتر در مورد این رابطه به [۲۰، بخش ۲,۲] و [۲] مراجعه کنید.

توجه داشته باشید که پیوند انتخاب شده لزوماً نباید با مشتق a مطابقت داشته باشد. در عوض، این مکانیسمی را برای اطمینان از یک تابع از دست دادن خوب فراهم می‌کند، زیرا واگرایی‌های برگمن ویژگی‌های خوبی دارند، از جمله محدب بودن در آرگومان اول. با این حال، این بدان معنا نیست که هر پیوند دیگری لزوماً منجر به عملکرد زیان نامطلوب می‌شود.

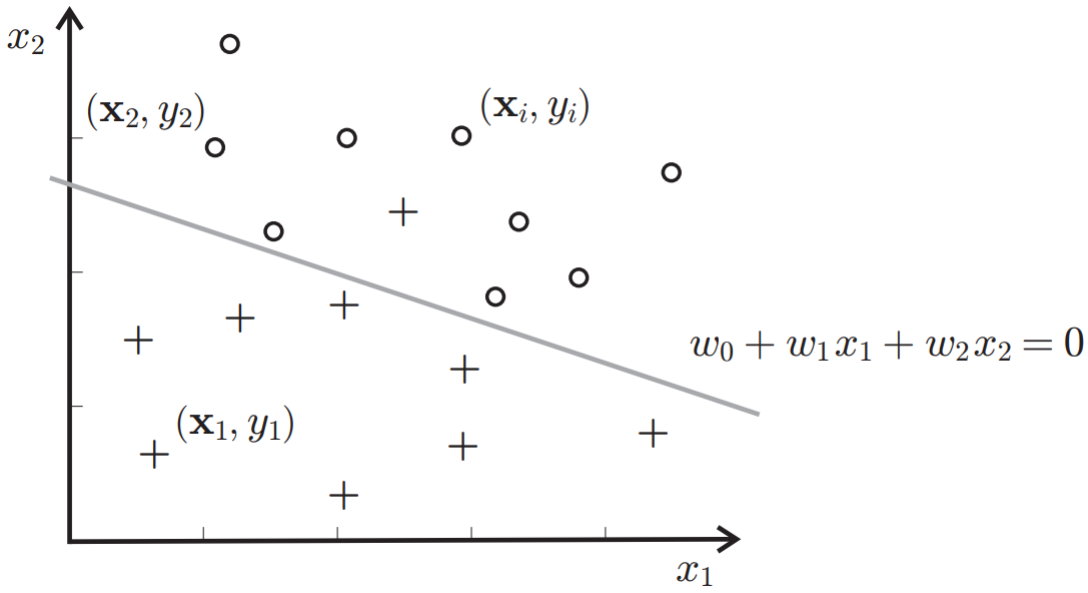
فصل ۸

طبقه‌بندی‌های خطی

فرض کنید ما علاقه‌مند به ساخت یک طبقه‌بندی کننده خطی $f : \mathbb{R}^d \rightarrow \{-1, +1\}$ هستیم. طبقه‌بندی خطی برای یافتن رابطه بین ورودی‌ها و خروجی‌ها با ساخت یک تابع خطی (نقطه، خط، صفحه یا ابر صفحه) که $\mathbb{R}^d$ را به دو نیم فاصله تقسیم می‌کند. دو نیمه فضا به ترتیب به عنوان مناطق تصمیم‌گیری برای مثال‌های مثبت و منفی عمل می‌کنند. با توجه به مجموعه داده‌ای $\mathcal{D} = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^n$ متشکل از مثال‌های مثبت و منفی، روش‌های زیادی وجود دارد که طبقه‌بندی‌کننده‌های خطی را می‌توان ساخت. به عنوان مثال، یک الگوریتم آموزشی ممکن است به طور صریح برای موقعیت‌یابی سطح تصمیم به منظور جداسازی مثال‌های مثبت و منفی بر اساس برخی معیارهای مرتبط با مشکل کار کند. به عنوان مثال، ممکن است سعی کند کسر نمونه‌ها را در سمت نادرست سطح تصمیم به حداقل برساند. از طرف دیگر، هدف الگوریتم آموزشی ممکن است تخمین مستقیم توزیع پسینی $p(y|\boldsymbol{x})$ باشد، در این صورت الگوریتم به احتمال زیاد بر اصول تخمین پارامتر رسمی تکیه می‌کند. به عنوان مثال، ممکن است احتمال را به حداکثر برساند. نمونه‌ای از یک طبقه‌بندی کننده با سطح تصمیم‌گیری خطی در شکل ۸,۱ نشان داده شده است

برای ساده‌سازی فرمالیسم در بخش‌های بعدی، یک جزء $x_0 = 1$ به هر ورودی اضافه می‌کنیم $(x_1, \ldots, x_d)$. این فضای ورودی را به $\mathcal{X} = \mathbb{R}^{d+1}$ گسترش می‌دهد، اما خوشبختانه ما را به یک نماد ساده‌شده نیز هدایت می‌کند که در آن مرز تصمیم‌گیری در $\mathbb{R}^d$ می‌تواند به صورت $\boldsymbol{w}^T\boldsymbol{x} = 0$ نوشته شود، که در آن $\boldsymbol{w} = (w_0, w_1, \ldots, w_d)$ مجموعه‌ای از وزن‌ها است و $\boldsymbol{x} = (x_0 = 1, x_1, \ldots, x_d)$ هر عنصر فضای ورودی است. با این وجود، باید به یاد داشته باشیم که ورودی‌های واقعی d-بعدی هستند.

قبلاً در اظهارات مقدماتی، ما یک طبقه‌بندی کننده را به عنوان تابع $f : \mathcal{X} \rightarrow \mathcal{Y}$ ارائه کردیم و مسئله یادگیری را به $p(y|\boldsymbol{x})$ تقریبی تبدیل کردیم. در شرایطی که طبقه‌بندی‌کننده‌های خطی، انعطاف‌پذیری ما محدود است، زیرا روش ما باید احتمالات پسینی $p(y|\boldsymbol{x})$ را بیاموزد و در عین حال سطح تصمیم‌گیری خطی در $\mathbb{R}^d$ داشته باشد. با این حال، اگر $p(y|\boldsymbol{x})$ به عنوان تابع یکنواخت $\boldsymbol{w}^T\boldsymbol{x}$ مدل‌سازی شود، می‌توان به این امر دست یافت. به عنوان مثال، $tanh(\boldsymbol{w}^T\boldsymbol{x})$ یا $\left(1 + e^{-\boldsymbol{w}^T x}\right)^{-1}$. البته، یک مدل آموزش دیده برای یادگیری احتمالات پسین $p(y|\boldsymbol{x})$ را می‌توان به عنوان یک پیش‌بینی کننده "نرم" یا یک تابع امتیاز دهی $s : \mathcal{X} \rightarrow [0, 1]$ مشاهده کرد. سپس، تبدیل از s به f یک کاربرد مستقیم از حداکثر اصل پسینی است: خروجی پیش‌بینی‌شده مثبت است اگر $\mathrm{s(x)} \geq 0.5$ باشد و اگر $s(x) < 0.5$ باشد، منفی است. به طور کلی، تابع امتیازدهی می‌تواند هر نقشه برداری باشد $s : \mathcal{X} \rightarrow \mathbb{R}$ ، با آستانه اعمال شده بر اساس هر مقدار خاص τ.



شکل ۸٫۱: مجموعه‌ای از داده‌ها در $\mathbb{R}^2$ شامل نه مثال مثبت و نه مثال منفی است. خط خاکستری نشان دهنده سطح تصمیم‌گیری خطی در $\mathbb{R}^2$ است. سطح تصمیم به طور کامل نکات مثبت را از منفی جدا نمی‌کند.

۸-۱ رگرسیون لجستیک[1]

اجازه دهید طبقه‌بندی باینری را در $\mathbb{R}^d$ در نظر بگیریم، که در آن $\mathcal{X} = \mathbb{R}^{d+1}$ و $Y = \{0, 1\}$. رگرسیون لجستیک یک مدل خطی تعمیم یافته است، که در آن توزیع بر روی Y داده شده $\boldsymbol{x}$ یک توزیع برنولی است و تابع انتقال تابع سیگموئید است که تابع لجستیک نیز نامیده می‌شود.

$$\sigma(t) = (1 + e^{-t})^{-1}$$

در شکل ۸,۲ ترسیم شده است. در اصطلاحات مشابه GLMها، تابع انتقال سیگموئید است و تابع پیوند - معکوس تابع انتقال - تابع logit است $\text{logit}(x) = \ln \frac{x}{1-x}$، با

1. $E[y|\mathbf{x}] = \sigma(\boldsymbol{\omega}^T \mathbf{x})$
2. $p(y|\mathbf{x}) = \text{Bernoulli}(\alpha) \text{ with } \alpha = E[y|\mathbf{x}]$

توزیع برنولی با α تابعی از x است

$$p(y|\mathbf{x}) = \begin{cases} \left(\frac{1}{1 + e^{-\omega^T x}}\right)^y & for\ y = 1 \\ \left(1 - \frac{1}{1 + e^{-\omega^T x}}\right)^{1-y} & for\ y = 0 \end{cases}$$

$$= \sigma(\mathbf{x}^T \mathbf{w})^y \left(1 - \sigma(\mathbf{x}^T \mathbf{w})\right)^{1-y}$$

[1] Logistic regression

که در آن $\boldsymbol{\omega} = (\omega_0, \omega_1, \ldots, \omega_d)$ مجموعه‌ای از ضرائب مجهول است که می‌خواهیم بازیابی کنیم (یا یاد بگیریم). توجه کنید که پیش‌بینی ما $\sigma(\boldsymbol{\omega}^T \mathbf{x})$ است و برآورده می‌شود

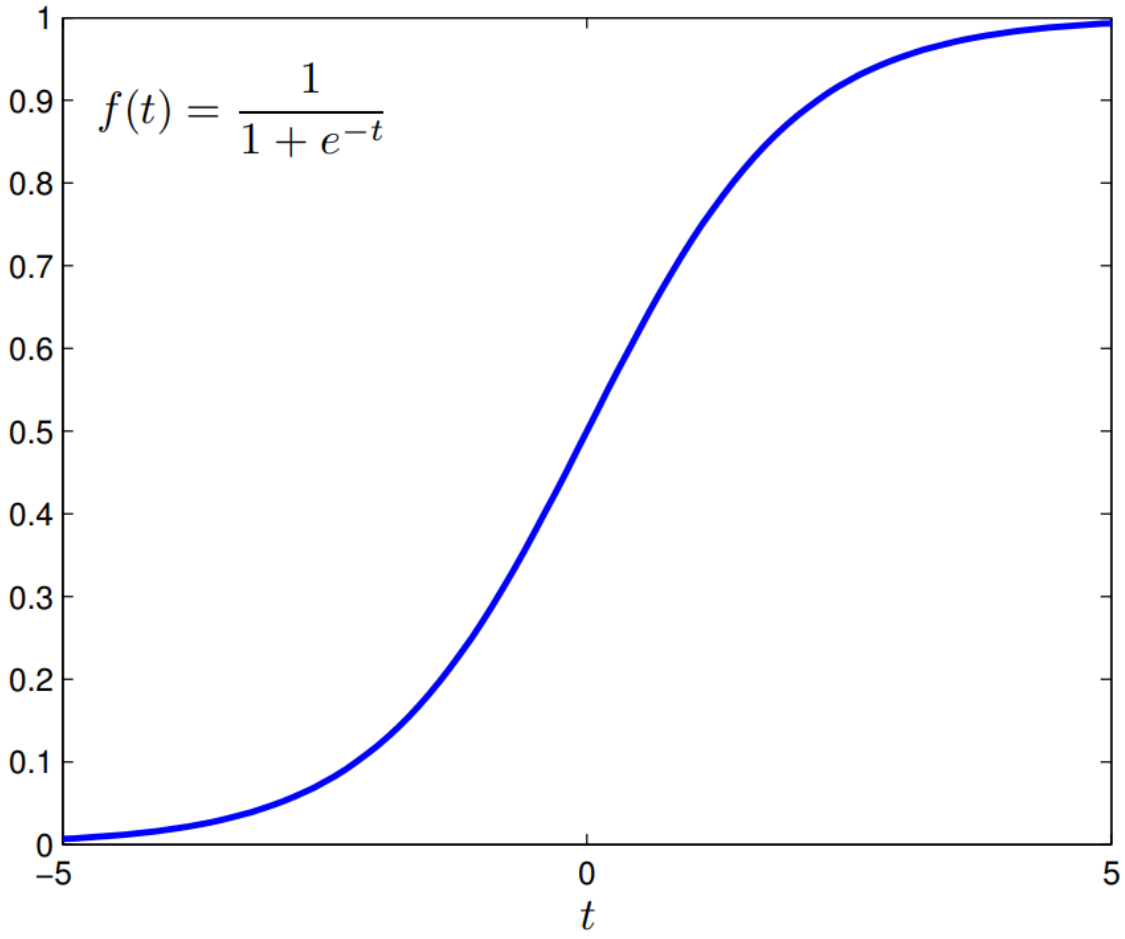
$$p(y = 1|\mathbf{x}) = \sigma(\boldsymbol{\omega}^T \mathbf{x})$$

بنابراین، مانند بسیاری از رویکردهای طبقه‌بندی باینری، هدف ما پیش‌بینی احتمال 1 بودن کلاس است. با توجه به این احتمال، می‌توانیم $p(y = 0|\boldsymbol{x}) = 1 - p(y = 1|\boldsymbol{x})$ را استنباط کنیم.

۱-۱-۸ پیش‌بینی برچسب‌های کلاس

تابعی که توسط رگرسیون لجستیک آموخته می‌شود، به جای یک پیش‌بینی صریح 0 یا 1، یک احتمال را بر می‌گرداند. بنابراین، ما باید این تخمین احتمال را بگیریم و آن را به یک پیش‌بینی مناسب کلاس تبدیل کنیم. برای یک نقطه داده‌ای که قبلاً دیده نشده بود $\boldsymbol{x}$ و مجموعه‌ای از ضرائب آموخته شده $\boldsymbol{w}$، به سادگی احتمال پسین را به این صورت محاسبه می‌کنیم.

$$P(Y = 1|\mathbf{x}, \mathbf{w}) = \frac{1}{1 + e^{-\mathbf{w}^T \mathbf{x}}}$$



شکل ۸/۲: تابع سیگموئید در بازه $[-5, 5]$.

اگر $P(Y = 1|\boldsymbol{x}, \boldsymbol{w}) \geq 0.5$ نتیجه می‌گیریم که نقطه داده $\boldsymbol{x}$ باید مثبت باشد $(\hat{y} = 1)$. در غیر این صورت، اگر $P(Y = 1|\boldsymbol{x}, \boldsymbol{w}^*) < 0.5$، نقطه داده را منفی می‌گذاریم $(\hat{y} = 0)$. پیش‌بینی کننده یک بردار $(d + 1)$بعدی $\boldsymbol{x} = (x_0 = 1, x_1, \ldots, x_d)$ را به صفر یا یک نگاشت می‌کند.

با این حال، همانطور که در فصل ۱۰ بحث می‌کنیم، این آستانه نباید 0.5 باشد. در برخی موارد، ممکن است فرد بیشتر به عدم شناسایی مثبت اهمیت دهد (به عنوان مثال، ناتوانی در شناسایی یک بیماری). در چنین حالتی، ممکن است اشتباه کردن در آستانه کوچکتر ایمن‌تر باشد، به طوری که نمونه‌های بیشتری به عنوان مثبت برچسب گذاری شوند. علاوه بر این، مقادیر احتمال خود می‌توانند منعکس کننده باشند: حتی اگر هر دو طبقه‌بندی کننده دقت خوبی داشته باشند، ترجیح داده می‌شود طبقه‌بندی کننده‌ای داشته باشیم که به طور مداوم احتمالات نزدیک به 0.9و 0.1 را تولید کند، به جای احتمالات با اطمینان کمتر که در اطراف 0.5 قرار دارند. دلیل این امر این است که انتظار می‌رود اغتشاش‌های کوچک تأثیر بیشتری بر طبقه‌بندی‌کننده دوم داشته باشند، که می‌تواند به طور ناگهانی برچسب‌گذاری را در یک نمونه تغییر دهد. برای تعیین کمیت این جنبه از پیش‌بینی، معیارهای دیگری مانند منحنی عملیاتی گزارش‌شده ترجیح داده می‌شوند که در بخش ۱۰,۴ در مورد آن بحث می‌کنیم. در حال حاضر، این آستانه ساده‌تر را فرض می‌کنیم و ارزیابی پیشرفته‌تر الگوریتم‌های طبقه‌بندی را به بعد واگذار می‌کنیم.

توجه داشته باشید که طبقه‌بندی کننده رگرسیون لجستیک یک طبقه‌بندی کننده خطی است، علیرغم اینکه سیگموئید غیرخطی است. این به این دلیل است که $P(Y = 1|\boldsymbol{x}, \boldsymbol{w}) \geq 0.5$ فقط زمانی که $\boldsymbol{w}^T\boldsymbol{x} \geq 0$ باشد. عبارت $\boldsymbol{w}^T\boldsymbol{x} = 0$ معادله یک ابر صفحه را نشان می‌دهد که مثال‌های مثبت و منفی را از هم جدا می‌کند.

۸-۱-۲ برآورد حداکثر احتمال برای رگرسیون لجستیک

از آنجایی که رگرسیون لجستیک یک GLM است، می‌توانیم از الگوریتم گرادیان کاهشی عمومی مشتق‌شده در بخش ۷,۳ با انتقال $\mathrm{f} = \sigma$، با گرادیان زیر از احتمال ورود به سیستم منفی در هر نمونه استفاده کنیم.

$$-\frac{\partial \mathrm{ll_i(w)}}{\partial w_\mathrm{j}} = \left(\sigma\left(\mathbf{x}_\mathrm{i}^\mathrm{T}\mathbf{w}\right) - \mathrm{y_i}\right)\mathrm{x_{ij}}$$

حتی اگر به‌روزرسانی نهایی را می‌دانیم، به عنوان یک تمرین، به صراحت راه‌حل حداکثر احتمال را استخراج خواهیم کرد. مانند قبل، فرض کنید که مجموعه داده $\mathcal{D} = \{(\mathrm{x_i}, \mathrm{y_i})\}_{\mathrm{i}=1}^{\mathrm{n}}$ یک $i.i.d.$ است. نمونه از یک توزیع احتمال ثابت اما ناشناخته $p(\boldsymbol{x}, y) = p(y|\boldsymbol{x})p(\boldsymbol{x})$. داده‌ها با رسم تصادفی یک نقطه $\boldsymbol{x}$ مطابق $p(\boldsymbol{x})$ تولید می‌شوند و سپس برچسب کلاس Y را مطابق توزیع برنولی در (۸,۱) تنظیم می‌کنند. احتمال ورود منفی برای این مجموعه داده $ll(\boldsymbol{w}) = \sum_{i=1}^{n} -ll_i(\boldsymbol{w})$ است که در آن

$$\mathrm{ll\_i(w)} = \log \mathrm{p(y\_i\,|x)}$$

$$= \mathrm{y_i}\log\sigma\left(\mathbf{w}^\mathrm{T}\mathbf{x_i}\right) + (1 - \mathrm{y_i})\log\left(1 - \sigma\left(\mathbf{w}^\mathrm{T}\mathrm{x_i}\right)\right)$$

$$= \left(y_i log\left(\frac{1}{1 + \mathrm{e}^{-\mathrm{w}^\mathrm{T}\mathrm{x_i}}}\right) + (1 + y_i) log\left(1 - \frac{1}{1 + \mathrm{e}^{-\mathrm{w}^\mathrm{T}\mathrm{x_i}}}\right)\right)$$

منفی $log - likelihood$ با این $\mathrm{ll_i}$ معمولاً به عنوان آنتروپی متقاطع[1] نامیده می‌شود

[1] cross-entropy

از اینجا، می‌توانید مشتق هر جزء را در این مجموع، با استفاده از قانون زنجیره‌ای برای سیگموئید، بگیرید. برای جزء اول، با $\mathrm{p_i} = \sigma(\theta_\mathrm{i})$

$$\frac{\partial \mathrm{y_i} \log\sigma(\mathbf{w}^\mathrm{T}\mathbf{x}_\mathrm{i})}{\partial w_\mathrm{j}} = y_i \frac{\partial \log \sigma(\theta_\mathrm{i})}{\partial \theta_\mathrm{i}} \frac{\partial \theta_\mathrm{i}}{w_j}$$

$$= \frac{\partial log p_i}{\partial p_i} \frac{\partial \mathrm{p_i}}{\partial \theta_\mathrm{i}} \frac{\partial \theta_i}{w_j}$$

$$= \mathrm{y_i} \frac{1}{\mathrm{p_i}} \frac{\partial \mathrm{p_i}}{\partial \theta_\mathrm{i}} \frac{\partial \theta_\mathrm{i}}{w_j}$$

$$= \mathrm{y_i} \frac{1}{\mathrm{p_i}} \sigma(\theta_\mathrm{i})\big(1 - \sigma(\theta_\mathrm{i})\big)\mathrm{x_{ij}}$$

$$= \mathrm{y_i}\big(1 - \sigma(\theta_\mathrm{i})\big)\mathrm{x_{ij}}$$

زیرا

$$\frac{\partial \sigma(\theta_\mathrm{i})}{\partial \theta_\mathrm{i}} = \sigma(\theta_\mathrm{i})\big(1 - \sigma(\theta_\mathrm{i})\big)$$

شما می‌توانید این مرحله را برای خودتان تأیید کنید، اما به صراحت تعریف σ را وصل کنید. برای جزء دوم، با دنبال کردن مراحل مشابه، دریافت می‌کنیم

$$\frac{\partial (1 - \mathrm{y_i}) \log\Big(1 - \sigma\big(\mathrm{w}^\mathrm{T}\mathrm{x_i}\big)\Big)}{\partial w_\mathrm{j}} = (\mathrm{y_i} - 1)\sigma(\theta_\mathrm{i})\mathrm{x_{ij}}$$

با جمع این موارد و گرفتن منفی، به گرادیان $(\mathrm{p_i} - \mathrm{y_i})\mathrm{x_{ij}}$ می‌رسیم.

برای تمرین بیشتر گرفتن شیب از این اهداف، می‌توانیم قبل از گرفتن گرادیان، هدف را کمی تغییر دهیم. این به مسیر دیگری برای استخراج قانون به‌روزرسانی برای رگرسیون لجستیک منجر می‌شود که اکنون از آن عبور می‌کنیم. ابتدا توجه کنید که

$$\left(1 - \frac{1}{1 + \mathrm{e}^{-\mathrm{w}^\mathrm{T}\mathrm{x_i}}}\right) = \frac{\mathrm{e}^{-\mathrm{w}^\mathrm{T}\mathrm{x_i}}}{1 + \mathrm{e}^{-\mathrm{w}^\mathrm{T}\mathrm{x_i}}}$$

که میدهد

$$\mathrm{ll}(\mathbf{w}) = \sum_{i=1}^{n} \left(-\mathrm{y_i} \cdot \log\left(1 + \mathrm{e}^{-\mathbf{w}^\mathrm{T}\mathbf{x_i}}\right) + (1 - \mathrm{y_i}) \cdot \log\left(\mathrm{e}^{-\mathbf{w}^\mathrm{T}\mathbf{x_i}}\right) - (1 - \mathrm{y_i}) \cdot \log\left(1 + \mathrm{e}^{-\mathbf{w}^\mathrm{T}\mathbf{x_i}}\right)\right)$$

$$= \sum_{i=1}^{n} \left((y_i - 1)\mathbf{w}^\mathrm{T}\mathbf{x_i} + \log\left(\frac{1}{1 + \mathrm{e}^{-\mathbf{w}^\mathrm{T}\mathbf{x_i}}}\right)\right)$$

باز هم، بر خلاف رگرسیون خطی، واضح است که هیچ راه‌حل بسته‌ای برای $\nabla \mathrm{ll}(\mathbf{w}) = 0$ وجود ندارد. بنابراین، ما باید با روش‌های بهینه‌سازی تکراری پیش برویم. ما $\mathbf{w}_0$ را معمولاً به یک بردار تصادفی یا به طور بالقوه با راه‌حل رگرسیون خطی که نقطه اولیه بسیار بهتری را ارائه می‌دهد مقداردهی اولیه می‌کنیم. از آنجایی که هدف محدب است، مقداردهی اولیه تنها بر تعداد مراحل تأثیر می‌گذارد، اما نباید مانع از همگرایی شیب نزولی به مینیمم مطلق شود. به روزرسانی گرادیان کاهشی تصادفی است

$$\mathbf{w}_{t+1} = \mathbf{w}_t - \eta_t\big(\sigma\big(\mathbf{x}_i^T\mathbf{w}_t\big) - y_i\big)\mathbf{x}_i$$

و به روزرسانی گرادیان کاهشی دسته‌ای است

$$\mathbf{w}_{t+1} = \mathbf{w}_t - \eta_t \sum_{i=1}^{n}\big(\sigma\big(\mathbf{x}_i^T\mathbf{w}_t\big) - y_i\big)\mathbf{x}_i$$

$$= \mathbf{w}_t - \eta_t\mathbf{X}^T(\sigma(\mathbf{X}\mathbf{w}_t) - \mathbf{y})$$

جایی که ما تعریف σ را هنگام اعمال بر یک بردار اضافه بارگذاری می‌کنیم به این معنی که به طور جداگانه برای هر عنصر در آن بردار اعمال می‌شود: $\sigma(v) = [\sigma(v_1),\ldots,\sigma(v_n)]$. هسین $\boldsymbol{H}_{-ll(w)} = -\boldsymbol{X}^T\boldsymbol{D}\boldsymbol{X}$ است که در آن $\mathcal{D}$ یک ماتریس مورب $n \times n$ با $p_i(1-p_i)$ در قطر است (نگاه کنید به (۷,۳))

تابع درستنمایی شرطی وزنی

در شرایط خاص، ممکن است توجیهی باشد که به شما اهمیت یکسانی برای هر نقطه داده، داده شود. این تابع احتمال شرطی را تغییر می‌دهد

$$\mathrm{l}(\mathbf{w}) = \prod_{i=1}^{n} \mathrm{p_i}^{\mathrm{c_i y_i}} \cdot (1-\mathrm{p_i})^{\mathrm{c_i}(1-\mathrm{y_i})}$$

که در آن $0 \le \mathrm{c_i} \le 1$ هزینه برای نقطه داده i است. با در نظر گرفتن $\mathrm{C} = \mathrm{diag}(\mathrm{c_1}, \mathrm{c_2}, \ldots, \mathrm{c_n})$ اکنون می‌توانیم گرادیان $\log - \mathrm{relihood}$ منفی را به این صورت بیان کنیم

$$-\nabla \mathrm{ll}(\mathbf{w}) = \mathrm{X}^T\mathbf{C}(\mathbf{p} - \mathbf{y})$$

و هسین را به عنوان

$$\mathbf{H}_{-\mathrm{ll}(\mathbf{w})} = -\mathbf{X}^{\mathrm{T}}\mathbf{C}\mathbf{P}(\mathbf{I} - \mathbf{P})\mathbf{X}$$

مشاهده این نکته جالب است که هسین به صورت نیمه قطعی مثبت باقی می‌ماند. بنابراین، انتظار می‌رود قانون به روزرسانی به مینیمم مطلق همگرا شود.

۳-۱-۸ مسائل مربوط به مینیمم کردن فاصله اقلیدسی

یک سوال طبیعی این است که چرا ما این مسیر را برای طبقه‌بندی خطی دنبال کردیم. به جای اینکه صریحا فرض کنیم $P(Y = 1|\boldsymbol{x}, \boldsymbol{w})$ یک توزیع برنولی است و راه‌حل حداکثر درستنمایی را برای $\sigma(x^T w) = E[Y|\boldsymbol{x}] =$ $P(Y = 1|\boldsymbol{x}, \boldsymbol{w})$ محاسبه کنیم. می‌توانستیم به سادگی تصمیم بگیریم از $\sigma(x^T w)$ برای پیش‌بینی اهداف $y \in \{0, 1\}$ استفاده کنیم و سپس سعی کنیم تفاوت آنها را با استفاده از هزینه مورد علاقه خود (هزینه مجذور) به حداقل برسانیم. متأسفانه، این مشخصات مسئله تصادفی‌تر منجر به بهینه‌سازی غیر محدب می‌شود. در واقع، نتیجه‌ای وجود دارد که استفاده از خطای اقلیدسی برای انتقال سیگموئید به طور تصاعدی مینیمم‌های محلی زیادی در تعداد ویژگی‌ها به دست می‌دهد [۱]. برای علاقه به این مسیر جایگزین، نشان خواهیم داد که این جهت به یک بهینه‌سازی غیر محدب منجر می‌شود

اجازه دهید تابع خطا با فاصله اقلیدسی اکنون به این صورت نوشته شود

$$\mathrm{Err}(\mathbf{w}) = \sum_{\mathrm{i}=1}^{\mathrm{n}}(\mathrm{y_i} - \mathrm{p_i})^2 \qquad \rhd\ \mathrm{p_i} = \sigma\big(\mathbf{x}_{\mathrm{i}}^{\mathrm{T}}\mathbf{w}\big), \mathrm{e_i} = \mathrm{y_i} - \mathrm{p_i}$$

کمینه سازی $Err(\boldsymbol{w})$ به طور فرمول‌بندی شده به این صورت بیان می‌شود

$$\mathbf{w}^* = \underset{\mathbf{w}}{\arg\min} \{\mathrm{Err}(\mathbf{w})\}$$

$$= \underset{\mathbf{w}}{\arg\min} \left\{ \sum_{i=1}^{n} (y_i - p_i)^2 \right\}$$

مشابه فرآیند حداکثر احتمال، هدف ما محاسبه بردار گرادیان و هسین تابع خطا خواهد بود. مشتقات جزئی تابع خطا را می‌توان به صورت زیر محاسبه کرد

$$\frac{\partial \mathrm{Err}(\mathbf{w})}{\partial w_{\mathrm{j}}} = \frac{\partial}{\partial w_{\mathrm{j}}} \sum_{i=1}^{n} e_i^2$$

$$= \sum_{i=1}^{n} 2 \cdot e_i \cdot \frac{\partial e_i}{\partial w_j}$$

$$= 2 \cdot \sum_{i=1}^{n} \left(\mathrm{y_i} - \frac{1}{1 + \mathrm{e}^{-\mathbf{w}^{\mathrm{T}}\mathbf{x_i}}} \right) \cdot \frac{1}{\left(1 + \mathrm{e}^{-\mathbf{w}^{\mathrm{T}}\mathbf{x_i}}\right)^2} \cdot \mathrm{e}^{-\mathbf{w}^{\mathrm{T}}\mathbf{x_i}} \cdot \left(-\mathrm{x_{ij}}\right)$$

$$= 2 \cdot \sum_{i=1}^{n} \mathrm{x_{ij}} \cdot \frac{1}{1 + e^{-\mathbf{w}^T\mathbf{x}_i}} \cdot \left(1 - \frac{1}{1 + e^{-\mathbf{w}^T\mathbf{x}_i}}\right) \cdot \left(\mathrm{y_i} - \frac{1}{1 + e^{-\mathbf{w}^T\mathbf{x}_i}}\right)$$

$$= -2\mathbf{f}_{\mathbf{j}}^{\mathrm{T}}\mathbf{P}(\mathbf{I} - \mathbf{P})(\mathrm{y} - \mathbf{p})$$

این بردار گرادیان را به شکل زیر ارائه می‌دهد

$$\nabla \mathrm{Err(w)} = -2\mathbf{X}\text{^T}\,\mathbf{P}\,(\mathbf{I} - \mathbf{P})\,(\mathrm{y} - \mathbf{p})$$

ماتریس $\boldsymbol{J} = \boldsymbol{P}\,(\boldsymbol{I} - \boldsymbol{P})\,\boldsymbol{X}$ به عنوان ژاکوبین[1] نامیده می‌شود. به طور کلی، ژاکوبین یک ماتریس $n \times d$ است که به صورت محاسبه می‌شود

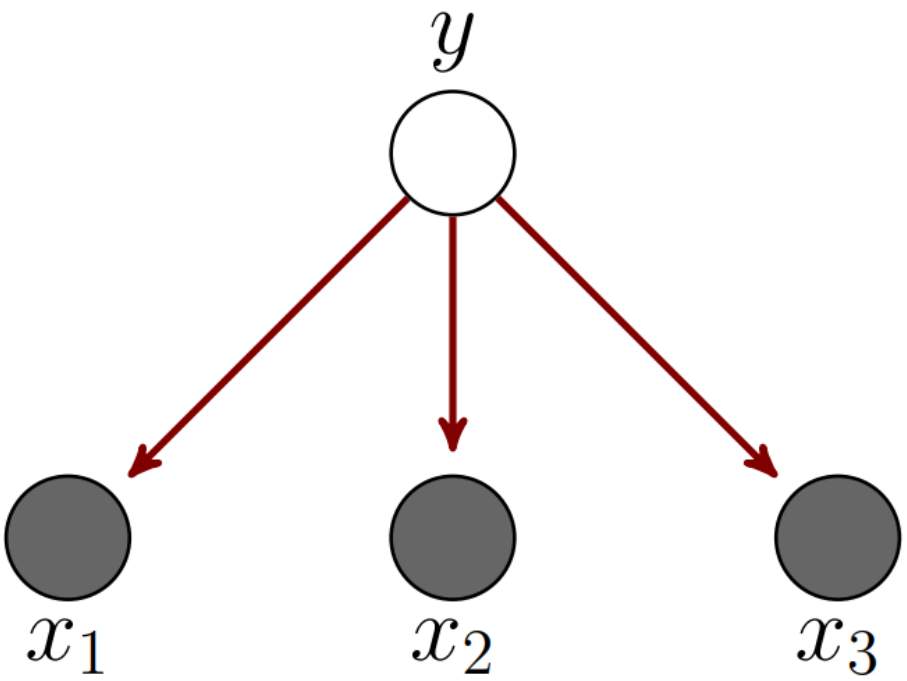
$$\begin{bmatrix} \frac{\partial \mathrm{e_1}}{\partial w_1} & \frac{\partial \mathrm{e_1}}{\partial w_2} & \cdots & \frac{\partial \mathrm{e_1}}{\partial w_{\mathrm{d}}} \\ \vdots & \ddots & & \\ \frac{\partial \mathrm{e_n}}{\partial w_1} & & & \frac{\partial \mathrm{e_n}}{\partial w_{\mathrm{d}}} \end{bmatrix}$$

دومین مشتق جزئی تابع خطا را می‌توان به این صورت پیدا کرد

$$\frac{\partial^2 \mathrm{Err}(\mathbf{w})}{\partial w_{\mathrm{j}}\, \partial w_{\mathrm{d}}} = 2 \cdot \sum_{i=1}^{n} \frac{\partial \mathrm{ei}}{\partial w_{\mathrm{d}}} \cdot \frac{\partial \mathrm{ei}}{\partial w_{\mathrm{j}}} + \mathrm{ei} \cdot \frac{\partial\, 2\ \mathrm{ei}}{\partial w_{\mathrm{j}}\, \partial w_{\mathrm{d}}}$$

$$= 2 \cdot \sum_{i=1}^{n} \mathrm{x_{ij}} \cdot \left(\mathrm{p_i^2}(1 - \mathrm{pi})^2 + \mathrm{p_i} \cdot \left(1 - \mathrm{p_i}\right) \cdot \left(2\mathrm{p_i} - 1\right) \cdot \left(\mathrm{y_i} - \mathrm{p_i}\right) \right) \cdot \mathrm{x_{ik}}$$

[1] Jacobian



شکل ۸/۳: مدل گرافیکی Naive Bayes، با سه ویژگی

بنابراین، هسین را می‌توان به این صورت محاسبه کرد

$$\mathbf{H}_{\mathrm{Err}(\mathbf{w})} = 2\mathbf{X}^{\mathrm{T}}(\mathbf{I}-\mathbf{P})^{\mathrm{T}}\mathbf{P}^{\mathrm{T}}\mathbf{P}(\mathbf{I}-\mathbf{P})\mathbf{X} + 2\mathbf{X}^{\mathrm{T}}(\mathbf{I}-\mathbf{P})^{\mathrm{T}}\mathbf{P}^{\mathbf{T}}\mathbf{E}(2\mathbf{P}-\mathbf{I})\mathbf{X} = 2\mathbf{J}^{\mathbf{T}}\mathbf{J} + 2\mathbf{J}^{\mathrm{T}}\mathbf{E}(2\mathbf{P}-\mathbf{I})\mathbf{X}$$

که در آن $\mathbf{P} = \mathrm{diag}\{\mathbf{p}\}, \mathbf{E} = \mathrm{diag}\{\mathrm{e}\}$ یک ماتریس مورب حاوی عناصر $\mathrm{E_{ii}} = \mathrm{e_i} = \mathrm{y_i} - \mathrm{p_i}$ و I یک ماتریس همانی است.

اکنون می‌بینیم که هسین تضمینی برای مثبت بودن نیمه قطعی نیست. این بدان معنی است که $Err(\boldsymbol{w})$ محدب نیست، یعنی باید مینیمم‌های متعدد با مقادیر مختلف تابع هدف داشته باشد. یافتن یک بهینه‌مطلق بستگی به این دارد که راه‌حل اولیه $\boldsymbol{w}^{(0)}$ چقدر مطلوب است و چگونه مرحله به روزرسانی وزن می‌تواند از مینیمم‌های محلی برای یافتن راه‌حل‌های بهتر فرار کند. با این حال، به حداقل رساندن این تابع غیر محدب، بسیار مشکل سازتر از آنتروپی متقابل محدب خواهد بود.

۸-۲ طبقه‌بندی کننده بیز ساده

طبقه‌بندی بیز ساده یک رویکرد مولد برای پیش‌بینی است. تا کنون، ما در مورد رویکردهای افتراقی[1] (رگرسیون خطی، رگرسیون لجستیک) بحث کرده‌ایم که سعی در یادگیری $p(y|\boldsymbol{x})$ دارند. برای یک تنظیم مولد، $p(\boldsymbol{x},y) = p(\boldsymbol{x}|y)p(y)$ را یاد می‌گیریم. همانطور که می‌توانید تصور کنید، این می‌تواند یک کار دشوارتر باشد، زیرا ما همچنین نیاز داریم که توزیع را روی

[1] discriminative

خود ویژگی‌ها بیاموزیم. برای بیز ساده ، ما به طور قابل توجهی به سادگی این توزیع مشترک را با ایجاد یک فرض قوی یاد می‌گیریم: ویژگی‌ها با توجه به برچسب مستقل هستند. این فرض با مدل گرافیکی در شکل ۸،۳ نشان داده شده است.

همانند طبقه‌بندی‌کننده‌های متمایز، مانند رگرسیون لجستیک، قانون تصمیم‌گیری برای برچسب‌گذاری یک نقطه به عنوان کلاس c (یعنی $y = 1$) است.

$$\underset{\mathrm{c} \in \{1,\ldots,\mathrm{k}\}}{\operatorname{argmax}} \mathrm{p}(\mathrm{y} = \mathrm{c}|\mathbf{x})$$

که در آن $\mathrm{p}(\mathrm{y} = 1|\mathbf{x}) \propto \mathrm{p}(\mathbf{x}|\mathrm{y} = 1)\mathrm{p}(\mathrm{y} = 1)$. برای دو کلاس، اگر $\mathrm{p}(\mathrm{y} = 1|\mathbf{x}) \geq \mathrm{p}(\mathrm{y} = 0|\mathbf{x})$ باشد، این به انتخاب کلاس 1 مربوط می‌شود.

برای شروع، تنظیمات ساده‌تری را با ویژگی‌های باینری در نظر می‌گیریم و سپس به ویژگی‌های پیوسته می‌پردازیم. توجه داشته باشید که بیز ساده یک طبقه‌بندی خطی برای ویژگی‌های باینری است. با این حال، به طور کلی‌تر، لزوماً یک طبقه‌بندی خطی نیست. توجه داشته باشید که یک طبقه‌بندی خطی طبقه‌بندی‌کننده‌ای است که در آن دو کلاس با یک صفحه خطی از هم جدا می‌شوند، یعنی مرز تصمیم‌گیری بر اساس ترکیب خطی ویژگی‌ها است.

۱-۲-۸ ویژگی‌های باینری و طبقه‌بندی خطی

اجازه دهید $\mathcal{D} = \{(xi, y)\}_{i=1}^{n}$ یک مجموعه داده ورودی باشد، که در آن $\mathcal{X} = \{0,1\}^d$ و $\mathcal{Y} = \{0,1\}$. تحت فرض بیز ساده ، ویژگی‌ها با توجه به برچسب کلاس مستقل هستند. بنابراین، ما می‌توانیم بنویسیم

$$\mathrm{p}(\mathbf{x}|\mathrm{y}) = \prod_{i=1}^{d} \mathrm{p}(\mathrm{x_i}|\mathrm{y})$$

یک انتخاب مناسب برای این احتمالات تک متغیره ساده‌تر، توزیع برنولی است، زیرا هر $\mathrm{x_j}$ باینری است،

$$\mathrm{p}(\mathrm{x_j}|\mathrm{y} = \mathrm{c}) = \mathrm{p}_{\mathrm{j,c}}^{\mathrm{x_j}}(1 - \mathrm{p}_{\mathrm{j,c}})^{1-\mathrm{x_j}}$$

پارامترهای توزیع برنولی $\mathrm{p_{j,c}} = \mathrm{p}(\mathrm{x_j} = 1|\mathrm{y} = \mathrm{c})$، با یک پارامتر مختلف $p_{j,c}$ برای هر کلاس c و برای هر ویژگی است. با محاسبه به راحتی می‌توانیم این پارامتر را از داده‌ها یاد بگیریم

$$p_{j,c} = \frac{number\ of\ times\ x_j = 1\ for\ class\ c}{number\ of\ datapoints\ labeled\ as\ class\ c}$$

به طور مشابه، ما می‌توانیم با استفاده از $p_c = p(y = c)$ قبلی یاد بگیریم

$$p_c = p(y = c) = \frac{number\ of\ datapoint\ labeled\ as\ class\ c}{total\ number\ of\ datapoints}$$

توجه داشته باشید که این رویکرد می‌تواند برای بیشتر از دو کلاس نیز انجام شود. پیش‌بینی در نقطه جدید $\mathbf{x}$ پس از آن اینگونه است

$$\max_{\mathrm{c} \in \mathcal{Y}} \mathrm{p}(\mathrm{y} = \mathrm{c}|\mathbf{x}) = \max_{\mathrm{c} \in \mathcal{Y}} \mathrm{p}(\mathbf{x}|\mathrm{y} = \mathrm{c})\mathrm{p}(\mathrm{y} = \mathrm{c})$$

$$= \max_{c \in y} \prod_{j=1}^{d} p(x_j|y = c)p(y = c)$$

$$= \max_{c \in y} \prod_{j=1}^{d} p_{j,c}p_c$$

تمرین ۱۰: راه‌حل بالا بصری است و از استخراج راه‌حل حداکثر احتمال به طور مشابه به دست می‌آید. با فرض اینکه شما n نقطه داده دارید و توزیع انتخاب شده $p(x_i|y)$ برنولی است همانطور که در بالا توضیح داده شد، حداکثر پارامترهای احتمال را برای $p_{j,c}, p_c for j = 1, \dots, d, c = 0,1$ استخراج کنید. برای ساده‌تر کردن کارها، از گزارش احتمال استفاده کنید

مرز طبقه‌بندی خطی با ویژگی‌ها و اهداف باینری

جالب اینجاست که طبقه‌بندی کننده بیز ساده با ویژگی‌های باینری و دو کلاس یک طبقه‌بندی کننده خطی است. این تا حدودی تعجب‌آور است، زیرا این رویکرد مولد بسیار متفاوت از آنچه قبلا انجام دادیم به نظر می‌رسد. برای اینکه ببینید چرا چنین است، توجه کنید که طبقه‌بندی کننده در چه زمانی تصمیم مثبت خواهد گرفت

$$p(y = 1|\mathbf{x}) \geq p(y = 0|\mathbf{x})$$

آن موقع است که

$$p(\mathbf{x}|y = 1)p(y = 1) \geq p(\mathbf{x}|y = 0)p(y = 0)$$

ما این نماد را با استفاده از $p(y = 0) = p(0)$، $p(\mathbf{x}|y = 0) = p(\mathbf{x}|0)$ کوتاه می‌کنیم. با استفاده از فرض بیز ساده، اکنون داریم

$$p(1) \prod_{j=1}^{d} p(x_j|1) \geq p(0) \prod_{j=1}^{d} p(x_j|0)$$

که پس از اعمال لگاریتم تبدیل می‌شود به

$$\log p(1) + \sum_{j=1}^{d} \log p(x_j|1) \geq \log p(0) + \sum_{j=1}^{d} \log p(x_j|0)$$

اجازه دهید اکنون احتمالات شرطی کلاس $p(x_j|y)$ را بررسی کنیم، زمانی که $y \in \{0, 1\}$ است. به یاد بیاورید که هر ویژگی برنولی توزیع شده است، یعنی.

$$p(x_j|1) = p_{j,1}^{x_j}(1 - p_{j,1})^{1-x_j}$$

و

$$p(x_j|0) = p_{j,0}^{x_j}(1 - p_{j,0})^{1-x_j}$$

که در آن پارامترهای $p_{j,c}$ از مجموعه آموزشی تخمین زده می‌شوند. با گرفتن $p(y = c) = p_c$، داریم

$$\sum_{j=1}^{d} \mathrm{x_j} \log \frac{\mathrm{p_{j,1}}\left(1-\mathrm{p_{j,0}}\right)}{\left(1-\mathrm{p_{j,1}}\right)\mathrm{p_{j,0}}} + \sum_{j=1}^{d} \log \frac{1-\mathrm{p_{j,1}}}{1-\mathrm{p_{j,0}}} + \log \frac{\mathrm{p_1}}{\mathrm{p_0}} \geq 0$$

می‌توانیم عبارت قبلی را به این صورت بنویسیم

$$w_0 + \sum_{j=1}^{d} w_j x_j \geq 0$$

که

$$w_0 = \log \frac{\mathrm{p_1}}{\mathrm{p_0}} + \sum_{j=1}^{d} \log \frac{1-\mathrm{p_{j,1}}}{1-\mathrm{p_{j,0}}}$$

$$w_j = \log \frac{\mathrm{p_{j,1}}\left(1-\mathrm{p_{j,0}}\right)}{\left(1-\mathrm{p_{j,1}}\right)\mathrm{p_{j,0}}} \qquad \mathrm{j} \in \{1, 2, \ldots, \mathrm{d}\}$$

بنابراین، در مورد ویژگی‌های باینری، بیز ساده یک طبقه‌بندی خطی است.

۲-۲-۸ بیز ساده پیوسته

برای ویژگی‌های پیوسته، توزیع برنولی دیگر برای $\mathrm{p(x_j|y)}$ مناسب نیست و باید توزیع شرطی متفاوتی $\mathrm{p(x|y)}$ انتخاب کنیم. یک انتخاب متداول توزیع گاوسی است، اکنون با میانگین و واریانس متفاوت برای هر ویژگی و کلاس، $\mu_{\mathrm{j,c}}, \sigma^2_{\mathrm{j,c}}$:

$$\mathrm{p(x_j|y=c)} = \left(2\pi\sigma^2_{\mathrm{j,c}}\right)^{-1/2} \exp\left(-\frac{\left(\mathrm{x_j}-\mu_{\mathrm{j,c}}\right)^2}{2\sigma^2_{\mathrm{j,c}}}\right)$$

از آنجایی که y هنوز گسسته است، می‌توانیم $p(y)$ را با استفاده از شمارش‌های قبلی تقریب کنیم. پارامترهای میانگین و واریانس حداکثر درستنمایی با میانگین نمونه و کوواریانس نمونه برای هر کلاس به طور جداگانه مطابقت دارد. این شامل محاسبه میانگین و واریانس ویژگی j در نقاط داده برچسب‌گذاری شده با کلاس c است:

$$\mu_{j,c} = \frac{\sum_{i=1}^{n} 1(y_i = c) x_j}{number\ of\ datapoints\ labeled\ as\ class\ c}$$

$$\sigma^2_{j,c} = \frac{\sum_{i=1}^{n} 1(y_i = c)\left(\mathrm{x_j}-\mu_{\mathrm{j,c}}\right)^2}{number\ of\ datapoints\ labeled\ as\ class\ c}$$

تمرین ۱۱: فرمول حداکثر احتمال را برای یک مدل بیز ساده گوسی استخراج کنید و بررسی کنید که آیا راه‌حل در واقع با میانگین نمونه و واریانس هر ویژگی و کلاس به طور جداگانه مطابق بالا مطابقت دارد.

۸-۳ رگرسیون لجستیک چند جمله ای

حال اجازه دهید طبقه‌بندی چندطبقه‌ای متمایز را در نظر بگیریم، که در آن $\mathcal{X} = \mathbb{R}^d$ و $\mathcal{Y} = \{1, 2, \ldots, k\}$. این تنظیم به طور طبیعی در یادگیری ماشینی ایجاد می‌شود، جایی که اغلب بیش از دو دسته وجود دارد. به عنوان مثال، اگر بخواهیم گروه خونی (A، B، AB و O) یک فرد را پیش‌بینی کنیم، چهار کلاس داریم. در اینجا ما در مورد طبقه‌بندی چند کلاسه بحث می‌کنیم که در آن فقط می‌خواهیم یک نقطه داده را با یک کلاس از k برچسب گذاری کنیم. در تنظیمات دیگر، ممکن است بخواهید یک نقطه داده را با چندین کلاس برچسب‌گذاری کنید. در پایان این بخش به اختصار به این موضوع اشاره شده است.

ما می‌توانیم با استفاده از ایده چندجمله‌ای و تابع پیوند مربوطه، مانند دیگر مدل‌های خطی تعمیم‌یافته، به این تنظیمات تعمیم دهیم. توزیع چند جمله‌ای عضوی از خانواده نمایی است. ما میتوانیم بنویسیم

$$\mathrm{p}(\mathbf{y}|\mathrm{x}) = \frac{1}{\mathrm{y}_1! \ldots \mathrm{y}_\mathrm{k}!} \mathrm{p}(\mathrm{y}_1 = 1|\mathrm{x})^{\mathrm{y}_1} \ldots, \mathrm{p}(\mathrm{y}_\mathrm{k} = 1|\mathrm{x})^{\mathrm{y}_\mathrm{k}} \qquad (8.3)$$

که در آن صورت معمولی $n! = 1$ زیرا $n = \sum_{j=1}^{k} y_j = 1$ زیرا ما فقط می‌توانیم یک مقدار کلاس داشته باشیم. مانند رگرسیون لجستیک، می‌توانیم $p(y_j = 1|\boldsymbol{x}) = \sigma(\boldsymbol{x}^T \boldsymbol{w}_j)$ را پارامتر کنیم. با این حال، ما همچنین باید اطمینان حاصل کنیم که $\sum_{j=1}^{k} \mathrm{p}\left(\mathrm{y}_\mathrm{j} = 1\middle|\mathbf{x}\right) = 1$. برای انجام این کار، حول کلاس نهایی، $p(y_k = 1|\boldsymbol{x}) \sum_{j=1}^{k} p(y_j = 1|\boldsymbol{x})$، «پیوت[1]» می‌کنیم. و فقط به طور صریح $\boldsymbol{w}_1, \ldots, \boldsymbol{w}_{k-1}$ را یاد می‌گیرد. توجه داشته باشید که این مدل‌ها به طور مستقل یاد نمی‌گیرند، زیرا آنها با احتمال برای آخرین کلاس گره خورده‌اند. پارامترها را می‌توان نمایش داد. به عنوان یک ماتریس $\boldsymbol{W} \in \mathbb{R}^{d \times k}$ که در آن $\boldsymbol{W} = [\boldsymbol{w}_1, \ldots, \boldsymbol{w}_k]$ از k بردار وزن با $\boldsymbol{w}_k = 0$ تشکیل شده است. خواهیم دید که چرا $\boldsymbol{w}_k = 0$ را ثابت می‌کنیم.

انتقال (معکوس پیوند) برای این تنظیم، انتقال سافت مکس است

$$\mathrm{softmax}(\mathrm{x}^T\mathrm{W}) = \left[\frac{\exp(\mathbf{x}^\mathrm{T}\mathbf{w}_1)}{\sum_{j=1}^{k}\exp(\mathbf{x}^\mathrm{T}\mathbf{w}_\mathrm{j})}, \ldots\ldots, \frac{\exp(\mathbf{x}^\mathrm{T}\mathbf{w}_\mathrm{k})}{\sum_{j=1}^{k}\exp(\mathbf{x}^\mathrm{T}\mathbf{w}_\mathrm{j})}\right]$$

$$= \left[\frac{\exp(\mathbf{x}^\mathrm{T}\mathbf{w}_1)}{1^\mathrm{T}\exp(\mathrm{x}^\mathrm{T}\mathbf{W})}, \ldots\ldots, \frac{\exp(\mathbf{x}^\mathrm{T}\mathbf{w}_\mathrm{k})}{1^\mathrm{T}\exp(\mathrm{x}^\mathrm{T}\mathbf{W})}\right]$$

و پیش‌بینی $softmax(x) = \hat{y} \in [0,1]^k$ است که در هر ورودی احتمال برچسب‌گذاری آن کلاس را می‌دهد، جایی که $\hat{\boldsymbol{y}}^T 1 = 1$ نشان می‌دهد که مجموع احتمالات برابر با 1 است. توجه داشته باشید که این مدل شامل تنظیمات باینری برای رگرسیون لجستیک است، زیرا $\sigma(\boldsymbol{x}^T\boldsymbol{w}) = \left(1 + exp(-\boldsymbol{x}^T\boldsymbol{w})\right)^{-1} = \frac{exp(\boldsymbol{x}^T\boldsymbol{w})}{1+exp(\boldsymbol{x}^T\boldsymbol{w})}$. وزن رگرسیون لجستیک چند جمله‌ای با دو کلاس $\boldsymbol{W} = [\boldsymbol{w}, 0]$ است.

$$\mathrm{p}(\mathrm{y} = 0|\mathrm{x}) = \frac{\exp(\mathbf{x}^\mathrm{T}\mathbf{w})}{1^\mathrm{T}\exp(\mathbf{x}^\mathrm{T}\mathbf{W})}$$

$$= \frac{\exp(\mathbf{x}^\mathrm{T}\mathbf{w})}{\exp(\mathbf{x}^\mathrm{T}\mathbf{w}) + \exp(\mathbf{x}^\mathrm{T}0)}$$

[1] pivot

$$= \frac{\exp(\mathbf{x}\text{^T}\,\mathbf{w})}{\exp(\mathbf{x}\text{^T}\,\mathbf{w}) \; + \; 1}$$

$$= \sigma\left(\mathbf{x}^{\mathrm{T}}\mathbf{w}\right)$$

به طور مشابه، برای $k \; > \; 2$، با ثابت کردن $w_k = 0$، وزن‌های دیگر $\mathbf{w_1}, \ldots, \mathbf{w_{k-1}}$ برای اطمینان از اینکه $p(y = k|x) =$ $\sum_{j=1}^{k} p(y = j|\boldsymbol{x}) \; = \; 1$ یاد گرفته می‌شود و آن $\frac{\exp(\mathbf{x}^{\mathrm{T}}\mathbf{w_k})}{1^{\mathrm{T}}\exp(\mathrm{x}^{\mathrm{T}}\mathbf{W})} = \frac{1}{1+\sum_{j=1}^{k-1} exp(\boldsymbol{x}^T \boldsymbol{w}_j)}$

با پارامترهای مدل با پارامتر $\mathbf{W}$ و انتقال softmax، می‌توانیم فرمول حداکثر درستنمایی را تعیین کنیم. با وارد کردن پارامتر به معادله (۸,۳)، با گرفتن log منفی آن احتمال و حذف ثابت‌ها، به از خطای زیر برای نمونه‌های $(\mathrm{x_1}, \mathrm{y_1}), \ldots, (\mathrm{x_n}, \mathrm{y_n})$ می‌رسیم

$$\min_{W \in \mathbb{R}^{d \times k}: \mathbf{W}_{:k=0}} \sum_{\mathrm{i}=1}^{\mathrm{n}} \log\left(1^{\mathrm{T}}\exp\left(\mathrm{x_i^T}\mathbf{W}\right)\right) - \mathbf{x_i^T}\mathbf{W}\mathbf{y_i}$$

با گرادیان

$$\nabla \sum_{i=1}^{n} \left(log\left(1^{T}\exp\left(\mathrm{x_i^T}\mathbf{W}\right)\right) - \mathrm{x_i^T}\mathbf{W}\mathbf{y_i}\right) = \sum_{i=1}^{n} \frac{\exp\left(\mathrm{x_i^T}\mathbf{W}\right)^{\mathrm{T}}\mathrm{x_i^T}}{1^{\mathrm{T}}\exp\left(\mathrm{x_i^T}\mathbf{W}\right)} - \mathbf{x_i}\mathbf{y_i}^{\mathrm{T}}$$

مانند قبل برای این گرادیان راه‌حل بسته‌ای نداریم و از روش‌های تکراری برای حل $\mathbf{W}$ استفاده می‌کنیم. توجه داشته باشید که در اینجا برخلاف روش‌های قبلی، روی بخشی از متغیر محدودیت داریم. با این حال، این فقط برای راحتی نوشته شده است. ما $\mathbf{W}_{:\mathrm{k}}$ را بهینه نمی‌کنیم، زیرا روی صفر ثابت شده است. می‌توان این کمینه‌سازی و گرادیان را بازنویسی کرد تا فقط برای $\mathrm{W}_{:(1:k-1)}$ اعمال شود. این مربوط به مقداردهی اولیه $\mathbf{W}_{:k} \; = \; 0$، و سپس تنها با استفاده از اولین ستون‌های $k - 1$ گرادیان در به روزرسانی به $\mathrm{W}_{:(1:k-1)}$ است.

پیش‌بینی نهایی $softmax(\boldsymbol{x}^T\boldsymbol{W}) \in [0,1]$ احتمال حضور در یک کلاس را نشان می‌دهد. همانند رگرسیون لجستیک، برای انتخاب یک کلاس، بالاترین مقدار احتمال انتخاب می‌شود. برای مثال، با $k \; = \; 4$، ممکن است [0.1 0.2 0.6 0.1] را پیش‌بینی کنیم و بنابراین تصمیم بگیریم که نقطه را در کلاس 3 واحد طبقه‌بندی کنیم.

نکته در مورد همپوشانی کلاس‌ها: اگر می‌خواهید چندین کلاس را برای نقطه داده $\mathbf{x}$ پیش‌بینی کنید، یک استراتژی رایج این است که پیش‌بینی‌های باینری جداگانه برای هر کلاس را یاد بگیرید. هر پیش‌بینی‌کننده به‌طور جداگانه مورد پرسش قرار می‌گیرد، و یک نقطه داده هر کلاس را با 0 یا 1 برچسب‌گذاری می‌کند، که احتمالاً بیش از یک کلاس دارای 1 است. در بالا، موردی را بررسی کردیم که در آن نقطه داده منحصراً در یکی از کلاس‌های ارائه شده قرار داشت، با تنظیم $n \; = \; 1$ در چند جمله‌ای.

فصل ۹

نمایش‌هایی برای یادگیری ماشینی

در ابتدا، ممکن است به نظر برسد که کاربرد رگرسیون خطی و طبقه‌بندی برای مسائل زندگی واقعی بسیار محدود است. و مشخص نیست که آیا درست است (بیشتر اوقات) فرض کنیم که متغیر هدف ترکیبی خطی از ویژگی‌ها است یا خیر. خوشبختانه، کاربرد رگرسیون خطی گسترده‌تر از آن چیزی است که در ابتدا تصور می‌شد. ایده اصلی این است که قبل از مرحله برازش، یک تبدیل غیر خطی به ماتریس داده X اعمال شود، که برازش غیر خطی را ممکن می‌کند. به دست آوردن چنین نمایش ویژگی مفیدی یک مشکل اساسی در یادگیری ماشین است.

ما ابتدا نمایش‌های ثابت را برای رگرسیون خطی بررسی می‌کنیم: شبکه‌های برازش منحنی چند جمله‌ای و تابع پایه شعاعی (RBF). سپس، در مورد بازنمایی‌های یادگیری بحث خواهیم کرد.

۹-۱ شبکه‌های تابع پایه شعاعی[1] و نمایش هسته

ایده شبکه‌های تابع پایه شعاعی (RBF) یک تعمیم طبیعی از برازش منحنی چند اسمی و رویکردهای بخش قبل است. با توجه به مجموعه داده $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^{n}$، با انتخاب p نقطه به عنوان "مراکز" در فضای ورودی X شروع می‌کنیم. ما آن مراکز را به صورت $\mathrm{c}_1, \mathrm{c}_2, \ldots, \mathrm{c}_\mathrm{p}$ نشان می‌دهیم. معمولاً، اینها را می‌توان از $\mathcal{D}$ انتخاب کرد یا با استفاده از برخی تکنیک‌های خوشه‌بندی (مانند الگوریتم EM، میانگین K) محاسبه کرد.

هنگامی که خوشه‌ها با استفاده از مدل مخلوط گاوسی تعیین می‌شوند، توابع پایه را می‌توان با این عنوان انتخاب کرد

$$\varphi_\mathrm{j}(\mathbf{x}) = \mathrm{e}^{-\frac{1}{2}(\mathbf{x}-\mathbf{c}_\mathrm{j})^\mathrm{T} \Sigma_\mathrm{j}^{-1}(\mathbf{x}-\mathbf{c}_\mathrm{j})}$$

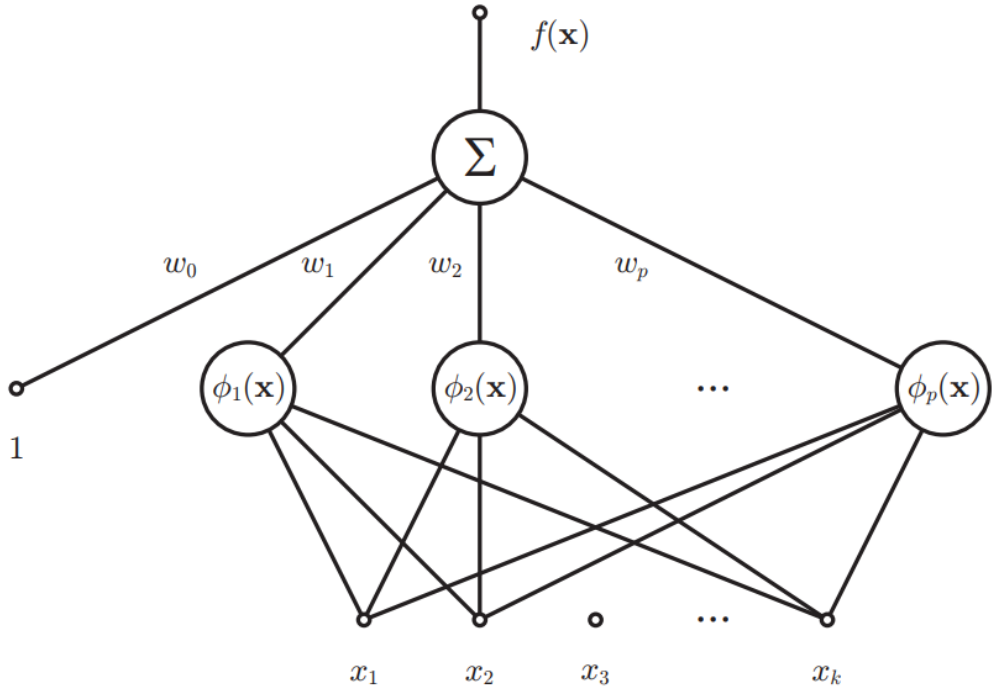
که در آن مراکز خوشه و ماتریس کوواریانس در طول خوشه‌بندی یافت می‌شوند. وقتی از $K-means$ یا خوشه‌بندی دیگر استفاده می‌شود، می‌توانیم از این فرمول استفاده کنیم

$$\varphi_\mathrm{j}(\mathrm{x}) = \mathrm{e}^{-\frac{||\mathrm{x}-\mathrm{cj}||^2}{2\sigma_\mathrm{j}^2}}$$

[1] Radial basis function

جایی که σ_j را می‌توان به طور جداگانه بهینه کرد. به عنوان مثال، با استفاده از مجموعه اعتبارسنجی در زمینه تبدیل‌های چند بعدی از x به $\boldsymbol{\Phi}$، توابع پایه را می‌توان به عنوان توابع هسته نیز نام برد، یعنی ماتریس $\varphi_j(\boldsymbol{x}) = k_j(x, \boldsymbol{c}_j)$

$$\boldsymbol{\Phi} = \begin{bmatrix} \varphi_0(\mathbf{x}_1) & \varphi_1(\mathbf{x}_1) & \cdots & \varphi_p(\mathbf{x}_1) \\ \varphi_0(\mathbf{x}_2) & \varphi_1(\mathbf{x}_2) & & \\ \vdots & & \ddots & \\ \varphi_0(\mathbf{x}_n) & & & \varphi_p(\mathbf{x}_n) \end{bmatrix}$$



شکل ۹/۱: شبکه تابع پایه شعاعی.

اکنون به عنوان یک ماتریس داده جدید استفاده می‌شود. برای یک ورودی داده شده $\mathbf{x}$، پیش‌بینی هدف y به این صورت محاسبه می‌شود

$$\mathrm{f}(\mathbf{x}) = w_0 + \sum_{j=1}^{p} w_j \varphi_j(\mathbf{x})$$

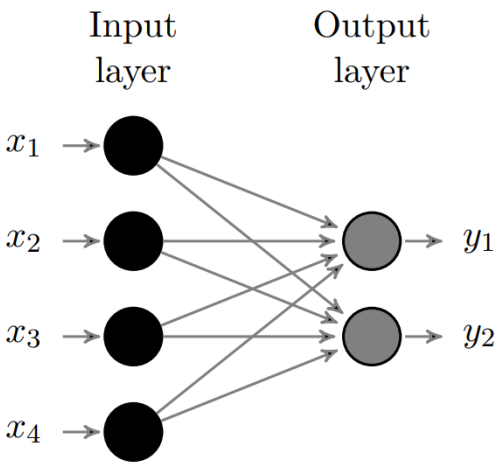
$$= \sum_{j=0}^{p} w_j \varphi_j(\mathbf{x})$$

جایی که $\varphi_0(\mathbf{x}) = 1$ و $\mathbf{w}$ یافت می‌شود. می‌توان ثابت کرد که با تعداد کافی از توابع پایه شعاعی می‌توانیم هر تابع را به طور دقیق تقریب کنیم. همانطور که در شکل ۹،۱ مشاهده می‌شود، می‌توانیم RBFها را به عنوان شبکه‌های عصبی در نظر بگیریم.

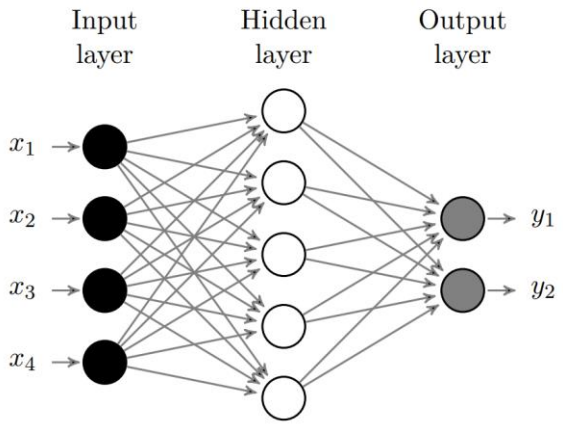
شبکه‌های RBF و نمایش‌های هسته بسیار مرتبط هستند. تمایز اصلی این است که نمایش‌های هسته از هر تابع هسته برای اندازه‌گیری شباهت $k(x, c_j) = \varphi_j(x)$ استفاده می‌کنند، که در آن توابع پایه شعاعی نمونه‌ای از یک هسته هستند. علاوه بر این، اگر یک هسته RBF انتخاب شود، برای بازنمایی هسته معمولی، مراکز از مجموعه داده آموزشی انتخاب می‌شوند. برای شبکه‌های RBF، انتخاب مراکز به طور کلی به عنوان یک مرحله مهم باقی مانده است، جایی که می‌توان آنها را از مجموعه آموزشی انتخاب کرد، اما همچنین می‌تواند به روش‌های دیگر انتخاب شود.

۹-۲ بازنمایی‌های یادگیری

رویکردهای زیادی برای بازنمایی یادگیری وجود دارد. دو رویکرد غالب، تکنیک‌های فاکتورسازی ماتریس (نیمه نظارت‌شده) و شبکه‌های عصبی هستند. شبکه‌های عصبی بر اساس مدل‌های خطی تعمیم‌یافته‌ای که مورد بحث قرار گرفتیم، ساخته می‌شوند و چندین مدل خطی تعمیم‌یافته را با هم ترکیب می‌کنند. تکنیک‌های فاکتورسازی ماتریسی (به عنوان مثال، کاهش ابعاد، کدگذاری پراکنده) معمولاً داده‌های ورودی را در یک فرهنگ لغت و یک نمایش جدید (پایه) فاکتور می‌کنند. ما ابتدا شبکه‌های عصبی را مورد بحث قرار می‌دهیم و سپس در مورد بسیاری از تکنیک‌های یادگیری بدون نظارت و نیمه نظارت که توسط فاکتورسازی‌های ماتریسی احاطه شده‌اند بحث خواهیم کرد.



شکل ۹/۲ : مدل خطی تعمیم یافته، مانند رگرسیون لجستیک.



شکل ۹/۳ : شبکه عصبی دو لایه استاندارد

۹-۲-۱ شبکه‌های عصبی

شبکه‌های عصبی شکلی از یادگیری بازنمایی نظارت شده هستند. مانند قبل، هدف یادگیری تابعی از ورودی‌ها، f، برای تولید یک پیش‌بینی از هدف است: $f(x)$. افزودن لایه‌های پنهان با توابع فعال‌سازی غیرخطی، یادگیری توابع غیرخطی f را امکان‌پذیر می‌سازد. برای برخی از شهود، می‌توان در نظر گرفت که اولین لایه‌های پنهان لایه بازنمایی را تشکیل می‌دهند، با یادگیری در آخرین لایه مربوط به قسمت پیش‌بینی نظارت شده است. شکل ۹,۲ مدل گرافیکی مدل‌های خطی تعمیم‌یافته‌ای را که در فصل‌های قبل مورد بحث قرار دادیم، نشان می‌دهد. جایی که وزن‌ها و انتقال مربوطه را می‌توان روی فلش‌ها در نظر گرفت (زیرا آنها متغیرهای تصادفی نیستند). شکل ۹,۳ یک شبکه عصبی با یک لایه پنهان را نشان می‌دهد. این شبکه عصبی دو لایه نامیده می‌شود، زیرا دو لایه وزن وجود دارد

در شکل، شبکه عصبی یک بردار ویژگی ۴ بعدی $x = [x_1, x_2, x_3, x_4]$ (یعنی $d = 4$) را وارد کرده و یک پیش‌بینی دو بعدی $y = [y_1, y_2]$ (یعنی $m = 2$). لایه پنهان شامل یک نقشه برداری از x به یک نمایش جدید است که ۵ بعدی است (یعنی $\mathrm{k}_1 = 5$ طبق نماد زیر). برای شبکه عصبی، اجازه دهید هر گره در این نمایش پنهان با $\mathrm{k} \in \{1, \ldots, 5\}$، نمایه شود. هر h_k از تبدیل وزن خطی x تشکیل شده است، مانند انتقال سیگموئید: $\mathrm{h_k} = \sigma\left(\sum_{j=1}^{d}\left(x_j w_{kj}\right) = \sigma(\boldsymbol{x}\boldsymbol{w_k})\right\}$ که در آن $\boldsymbol{w}_k \in \mathbb{R}^d$ وزن‌های روی اولین لایه است که برای تولید گره در نمایش پنهان شماره k استفاده می‌شود.

مثال ۱۸: برای مثال ساده، $d = 1$ (یعنی یک مشاهده ورودی)، $m = 1$ (یعنی یک خروجی)، $k_1 = 2$ (یعنی لایه پنهان 2 بعدی) و یک انتقال سیگموید را در نظر بگیرید تا اولین مورد را بدست آورید. لایه پنهان فرض کنید یک نمونه به ما داده شده است (x, y). سپس مشاهده ورودی x تبدیل به

$$\mathbf{h} = [\mathrm{h}_1, \mathrm{h}_2], \qquad \text{with } \mathrm{h}_1 = \sigma\left(\mathrm{x}w_{1_{(2)}}\right) \text{and } \mathrm{h}_2 = \sigma\left(\mathrm{x}w_2^{(2)}\right) \qquad \text{for } w_1^{(2)}, w_2^{(2)} \in \mathbb{R}.$$

برای جلوگیری از انتقال نماد، از $x \in \mathbb{R}^{1\times d}$ برای ارائه یک ردیف از ماتریس داده $x \in \mathbb{R}^{n\times d}$ و بردار ردیف $\boldsymbol{h} \in \mathbb{R}^{1\times k_1}$ استفاده کردیم. برای تمایز بین وزن‌های لایه اول و آخر از نماد بالانویس استفاده می‌کنیم. ممکن است غیرقابل درک به نظر برسد که چرا ما برچسب $\boldsymbol{w}^{(2)}$ را برای لایه ورودی و $\boldsymbol{w}^{(1)}$ برای لایه خروجی می‌زنیم، اما در زیر خواهید دید که نشان‌گذاری شروع نمایه‌سازی از لایه خروجی را ساده‌تر می‌کند.

هنگامی که $\boldsymbol{h}$ را داریم، می‌توانیم وانمود کنیم که $\boldsymbol{h}$ نمایش ورودی جدید است و ادامه می‌دهیم و یک مدل خطی (تعمیم‌یافته) در این لایه آخر یاد می‌گیریم. بیایید دو حالت را در نظر بگیریم: $y \in \mathbb{R}$ و $y \in \{0, 1\}$. اگر $y \in \mathbb{R}$ ، از رگرسیون خطی برای آخرین لایه استفاده می‌کنیم و بنابراین وزن‌های $w^{(2)} \in \mathbb{R}^2$ را طوری یاد می‌گیریم که $\boldsymbol{h}\boldsymbol{w}^{(2)}$خروجی واقعی y را تقریب می‌زند. اگر $y \in \{0, 1\}$ ، از رگرسیون لجستیک برای آخرین لایه استفاده می‌کنیم و بنابراین وزن‌های $\boldsymbol{w}^{(2)} \in \mathbb{R}^2$ را طوری یاد می‌گیریم که $\sigma\left(\boldsymbol{h}\boldsymbol{w}^{(2)}\right)$ خروجی واقعی y را تقریب می‌زند.

حال حالت کلی‌تر را با هر d، k_1، m در نظر می‌گیریم. برای ارائه شهودی برای این تنظیم کلی تر، ما با یک لایه پنهان، برای تابع انتقال سیگموئید و از دست دادن خروجی آنتروپی متقابل شروع می‌کنیم. برای رگرسیون لجستیک $\boldsymbol{W} \in \mathbb{R}^{d\times m}$ را با $f(\boldsymbol{x}\boldsymbol{W}) = \sigma(\boldsymbol{x}\boldsymbol{W}) \approx y$ برآورد کردیم. ما یک بردار خروجی $\mathrm{y} \in \mathbb{R}^{\mathrm{m}}$ را پیش‌بینی می‌کنیم، زیرا تعمیم‌های بعدی را واضح‌تر می‌کند و نماد وزن‌های هر لایه را یکنواخت‌تر می‌کند. وقتی یک لایه پنهان اضافه می‌کنیم، دو ماتریس پارامتر $W^{(1)} \in \mathbb{R}^{k_1\times m}$ و $\mathrm{W}^{(2)} \in \mathbb{R}^{\mathrm{d}\times \mathrm{k}_1}$ داریم که k_1 بعد لایه پنهان است.

$$\mathbf{h} = \sigma\left(\mathrm{x}\mathbf{W}^{(2)}\right) = \begin{bmatrix} \sigma\left(\mathbf{x}\mathbf{W}_{:1}^{(2)}\right) \\ \sigma\left(\mathbf{x}\mathbf{W}_{:2}^{(2)}\right) \\ \vdots \\ \sigma\left(\mathbf{x}\mathbf{W}_{:\mathrm{k}_1}^{(2)}\right) \end{bmatrix} \in \mathbb{R}^{\mathrm{k}_1}$$

که در آن تابع سیگموئید برای هر ورودی در $\boldsymbol{x}\boldsymbol{W}^{(2)}$ و $\mathbf{h}\boldsymbol{W}^{(2)}$ اعمال می‌شود. این لایه پنهان مجموعه جدیدی از ویژگی‌ها است و مجدداً بهینه‌سازی رگرسیون لجستیک معمولی را برای یادگیری وزن‌ها در $\mathbf{h}$ انجام خواهید داد:

$$\mathrm{p}(\mathrm{y} = 1|\mathbf{x}) = \sigma\left(\mathbf{h}\mathbf{W}^{(1)}\right) = \sigma\left(\sigma\left(\mathbf{x}\mathbf{W}^{(2)}\right)\mathbf{W}^{(1)}\right)$$

با مدل احتمالی و پارامتر مشخص شده، اکنون باید یک الگوریتم برای بدست آوردن آن پارامترها استخراج کنیم. مانند قبل، ما یک رویکرد ماکسیمم احتمال را اتخاذ کرده و به‌روزرسانی‌های شیب نزولی را استخراج می‌کنیم. به نظر می‌رسد این ترکیب

انتقال‌ها مسائل را پیچیده می‌کند، اما همچنان می‌توانیم شیب $w.r.t$ را بگیریم. ما اکنون پارامترهای بیشتری داریم: $W^{(2)} \in$ $\boldsymbol{W}^{(1)} \in \mathbb{R}^{1\times k_1}\mathbb{R}^{k_1\times d}$, . وقتی گرادیان $w.r.t$ را داریم. هر ماتریس پارامتر، طبق معمول به سادگی یک قدم در جهت منفی گرادیان برداریم. گرادیان برای این پارامترها اطلاعات را به اشتراک می‌گذارند. برای کارایی محاسباتی، گرادیان ابتدا برای $\boldsymbol{W}^{(1)}$ محاسبه می‌شود، و اطلاعات گرادیان تکراری برای محاسبه گرادیان برای $\boldsymbol{W}^{(2)}$ ارسال می‌شود. این الگوریتم معمولاً به نام انتشار برگشتی نامیده می‌شود که در ادامه به توضیح آن می‌پردازیم.

به طور کلی، ما می‌توانیم گرادیان را برای هر تعداد لایه پنهان محاسبه کنیم. هر تابع انتقال قابل تفکیک $\mathrm{f_1},\ldots,\mathrm{f_H}$ را ، نشان دهید. با f_1 به عنوان انتقال خروجی، و $k_1,\ldots,k_{H-1}$، مرتب شده است. به عنوان ابعاد پنهان با لایه‌های پنهان $H-1$. سپس خروجی از شبکه عصبی است

$$\mathrm{f_1}\big(\mathrm{f_2}\big(\ldots\mathrm{f_{H-1}}\big(\mathrm{f_H}\big(\mathbf{x}\mathbf{W}^{(\mathrm{H})}\big)\mathbf{W}^{(\mathrm{H-1})}\big)\ldots\big)\mathbf{W}^{(1)}\big)$$

که $\mathbf{W}^{(1)} \in \mathbb{R}^{\mathrm{k_1}\times\mathrm{m}}, \mathbf{W}^{(2)} \in \mathbb{R}^{\mathrm{k_2}\times\mathrm{k_1}},\ldots,\mathbf{W}^{(\mathrm{H})} \in \mathbb{R}^{\mathrm{d}\times\mathrm{k_{H-1}}}$

الگوریتم پس انتشار

ما با استخراج پس انتشار برای دو لایه شروع خواهیم کرد. گسترش چندین لایه با توجه به این اشتقاق واضح‌تر خواهد بود. با توجه به اندازه شبکه، ما اغلب با گرادیان کاهشی تصادفی یاد خواهیم گرفت. بنابراین، ابتدا این گرادیان را با فرض اینکه فقط یک نمونه (x,y) داریم محاسبه می‌کنیم.

الگوریتم انتشار برگشتی به سادگی گرادیان کاهشی بر روی یک هدف غیر محدب، با ترتیب دقیق محاسبات برای جلوگیری از تکرار محاسبات است. به ویژه، ابتدا به جلو منتشر می‌شود و متغیر $\boldsymbol{h} = f_2\big(\boldsymbol{x}\boldsymbol{W}^{(2)}\big) \in \mathbb{R}^{1\times k}$ و سپس $\hat{y} =$ $f_1\big(f_2\big(\boldsymbol{x}\boldsymbol{W}^{(2)}\big)\boldsymbol{W}^{\{(1)\}}\big) = f_1\big(\boldsymbol{h}\boldsymbol{W}^{(1)}\big)$ را محاسبه می‌کند. . سپس خطای بین پیش‌بینی y و برچسب واقعی را محاسبه می‌کنیم. گرادیان این خطا (از دست دادن) را $w.r.t$ می‌گیریم. به پارامترهای ما؛ در این حالت، برای محاسبه کارآمد، بهترین ترتیب، محاسبه گرادیان w.r.t است. ابتدا به آخرین پارامتر $\mathbf{W}^{(1)}$و سپس $\mathbf{W}^{(2)}$برسید. دلیل استفاده از اصطلاح $back-propagation$ به همین دلیل است، زیرا خطا ابتدا از آخرین لایه به عقب منتشر می‌شود.

سپس انتخاب‌ها شامل انتخاب انتقال‌ها در هر لایه، تعداد گره‌های پنهان و از دست دادن لایه آخر است. تطبیق محدب $\mathrm{L}(\cdot,\mathrm{y})$ به $\mathrm{p}(\mathrm{y}|\mathbf{x})$ انتخاب شده و تابع انتقال متناظر برای آخرین لایه شبکه عصبی بستگی دارد، درست مانند مدل‌های خطی تعمیم یافته. برای سهولت علامت‌گذاری، این تابع خطا را به صورت تعریف می‌کنیم

$$\mathrm{Err}\big(\mathbf{W}^{(1)},\mathbf{W}^{(2)}\big) \stackrel{\text{def}}{=} \sum_{\mathrm{k}=1}^{\mathrm{m}} \mathrm{L}\left(\mathrm{f_1}\left(\mathrm{f_2}\big(\mathbf{x}\mathbf{W}^{(2)}\big)\mathbf{W}^{(1)}_{:\mathrm{k}}\right),\mathbf{y}_\mathrm{k}\right)$$

برای یک نمونه $(\mathbf{x},\mathbf{y})$. به عنوان مثال، برای $\mathrm{p}(\mathrm{y}=1|\mathbf{x})$ گاوسی و انتقال هویت[1] f_2، $Err\big(\boldsymbol{W}^{(1)},\boldsymbol{W}^{(2)}\big) =$ $\big(f_2\big(\boldsymbol{x}\boldsymbol{W}^{(2)}\big)\boldsymbol{W}^{(1)}-y\big)^2$ را بدست می‌آوریم. اگر $p(y=1|\boldsymbol{x})$ یک توزیع برنولی باشد، آنگاه از دست دادن رگرسیون لجستیک (آنتروپی متقاطع) را انتخاب می‌کنیم.

مانند قبل، گرادیان‌های اتلاف $w.r.t$ را محاسبه می‌کنیم. پارامترهای ما ابتدا مشتق جزئی $w.r.t$ را می‌گیریم. پارامترهای $\boldsymbol{W}^{(1)}$ (با فرض اینکه $\boldsymbol{W}^{(2)}$ ثابت باشد).

[1] identity transfer

$$\frac{\partial \mathrm{Err}\left(\mathbf{W}^{(1)},\mathbf{W}^{(2)}\right)}{\partial \mathbf{W}_{\mathrm{jk}}^{(1)}}=\frac{\partial \mathrm{L}\left(\mathrm{f}_1\left(\mathrm{f}_2\left(\mathbf{x}\mathbf{W}^{(2)}\right)\mathbf{W}^{(1)}\right),\mathbf{y}\right)}{\partial \mathbf{W}_{\mathrm{jk}}^{(1)}} \qquad \triangleright\ \hat{y}_k=\mathrm{f}_1\left(\mathbf{h}\mathbf{W}_{:\mathrm{k}}^{(1)}\right)$$

$$=\left(\frac{\partial \mathrm{L}\left(\hat{\mathbf{y}}_{\mathrm{k}},\mathbf{y}_{\mathrm{k}}\right)}{\partial \hat{\mathbf{y}}_{\mathrm{k}}}\right)\frac{\partial \hat{\mathbf{y}}_{\mathrm{k}}}{\partial \mathbf{W}_{\mathrm{jk}}^{(1)}}$$

که در آن فقط $\hat{y}_k$ تحت تأثیر $\boldsymbol{W}_{jk}^{(1)}$ در از دست دادن قرار می‌گیرد، و بنابراین گرادیان برای بقیه صفر است. در ادامه،

$$\frac{\partial Err\left(\mathbf{W}^{(1)},\mathbf{W}^{(2)}\right)}{\partial \mathbf{W}_{jk}^{(1)}}=\left(\frac{\partial \mathrm{L}\left(\hat{\mathbf{y}}_{\mathrm{k}},\mathbf{y}_{\mathrm{k}}\right)}{\partial \hat{\mathbf{y}}_{\mathrm{k}}}\right)\frac{\partial \mathrm{f}_1\left(\theta_{\mathrm{k}}^{(1)}\right)}{\partial \theta_{\mathrm{k}}^{(1)}}\frac{\partial \theta_{\mathrm{k}}^{(1)}}{\partial \mathbf{W}_{\mathrm{jk}}^{(1)}} \qquad \triangleright\ \theta_k^{(1)}=\mathbf{h}\mathbf{W}_{:k}^{(1)}$$

$$=\left(\frac{\partial \mathrm{L}\left(\hat{\mathbf{y}}_{\mathrm{k}},\mathbf{y}_{\mathrm{k}}\right)}{\partial \hat{\mathbf{y}}_{k}}\right)\frac{\partial \mathrm{f}_1\left(\theta_{\mathrm{k}}^{(1)}\right)}{\partial \theta_{\mathrm{k}}^{(1)}}\boldsymbol{h}_j$$

در این مرحله این معادلات انتزاعی هستند. اما محاسبه آنها برای هزینه و انتقالاتی که ما بررسی کردیم ساده است. به عنوان مثال، برای $L(\hat{y}_k,y_k)=\frac{1}{2}(\hat{y}_k-y_k)^2$ و f_2 هویت، به دست می‌آوریم

$$\frac{\partial L\left(\hat{\mathbf{y}}_k,\mathbf{y}_k\right)}{\partial \hat{\mathbf{y}}_k}=\left(\hat{\mathbf{y}}_k-\mathbf{y}_k\right)$$

$$\frac{\partial \mathrm{f}_1\left(\theta_{\mathrm{k}}^{(1)}\right)}{\partial \theta_{\mathrm{k}}^{(1)}}=1$$

که می‌دهد

$$\frac{\partial \mathrm{Err}\left(\mathbf{W}^{(1)},\mathbf{W}^{(2)}\right)}{\partial \mathbf{W}_{\mathrm{jk}}^{(1)}}=\left(\frac{\partial L\left(\hat{\mathbf{y}}_k,\mathbf{y}_k\right)}{\partial \hat{\mathbf{y}}_k}\right)\frac{\partial \mathrm{f}_1\left(\theta_{\mathrm{k}}^{(1)}\right)}{\partial \theta_{\mathrm{k}}^{(1)}}\boldsymbol{h}_j=\left(\hat{\mathbf{y}}_k-\mathbf{y}_k\right)\boldsymbol{h}_j$$

به روز رسانی گرادیان طبق معمول با $W^{(1)}=W^{(1)}-\alpha(\hat{y}-y)h^T$ برای مقداری α اندازه گام است.

سپس، گرادیان جزئی را با توجه به $W^{(2)}$ محاسبه می‌کنیم. اما اکنون، کل متغیر خروجی $y\in\mathbb{R}^{1\times m}$ تحت تأثیر انتخاب $W_{ij}^{(2)}$ برای همه $i\in 1,\ldots,k_2$ و $\mathrm{j}\in 1,\ldots,\mathrm{k}_1$. بنابراین، باید مشتق جزئی $\mathrm{w.r.t}$ را برای هر y بگیریم.

$$\frac{\partial \mathrm{Err}\left(\mathbf{W}^{(1)},\mathbf{W}^{(2)}\right)}{\partial \mathbf{W}_{\mathrm{ij}}^{(2)}}=\frac{\partial \sum_{\mathrm{k}=1}^{\mathrm{m}}\mathrm{L}\left(\mathrm{f}_1\left(\mathrm{f}_2\left(\mathbf{x}\mathbf{W}^{(2)}\right)\mathbf{W}_{:\mathrm{k}}^{(1)}\right),\mathbf{y}_{\mathrm{k}}\right)}{\partial \mathbf{W}_{\mathrm{ij}}^{(2)}}$$

$$\sum_{k=1}^{m}\frac{\partial L(\hat{\mathbf{y}}_k,\mathbf{y}_k)}{\partial \hat{\mathbf{y}}_k}\frac{\partial \hat{\mathbf{y}}_{\mathrm{k}}}{\partial \mathbf{W}_{\mathrm{ij}}^{(2)}} \qquad \triangleright\ \hat{\mathrm{y}}_{\mathrm{k}}=\mathrm{f}_1\left(\mathbf{h}\mathbf{W}_{:\mathrm{k}}^{(1)}\right)=\mathrm{f}_1\left(\theta_{\mathrm{k}}^{(1)}\right)$$

$$=\sum_{k=1}^{m}\frac{\partial L\left(\hat{\mathbf{y}}_k,\mathbf{y}_k\right)}{\partial \hat{\mathbf{y}}_k}\frac{\partial \mathrm{f}_1\left(\theta_{\mathrm{k}}^{(1)}\right)}{\partial \theta_{\mathrm{k}}^{(1)}}\frac{\partial \theta_{\mathrm{k}}^{(1)}}{\partial \mathbf{W}_{\mathrm{ij}}^{(2)}}$$

در ادامه

$$\frac{\partial \theta_{\mathrm{k}}^{(1)}}{\partial \mathbf{W}_{\mathrm{ij}}^{(2)}} = \frac{\partial \mathbf{h}\mathbf{W}_{:\mathrm{k}}^{(1)}}{\partial \mathbf{W}_{\mathrm{ij}}^{(2)}} = \frac{\partial \sum_{\mathrm{l}=1}^{\mathrm{k}} \mathbf{h}_{\mathrm{l}} \mathbf{W}_{\mathrm{lk}}^{(1)}}{\partial \mathbf{W}^{(2)}}$$

$$\frac{\partial \sum_{l=1}^{k} \mathrm{f}_2\left(\mathbf{x}\mathbf{W}_{:\mathrm{l}}^{(2)}\right)\mathbf{W}_{\mathrm{lk}}^{(1)}}{\partial \mathbf{W}_{\mathrm{ij}}^{(2)}}$$

$$\sum_{l=1}^{k} \mathbf{W}_{lk}^{(1)} \frac{\partial \mathrm{f}_2\left(\mathbf{x}\mathbf{W}_{:\mathrm{l}}^{(2)}\right)}{\partial \mathbf{W}_{\mathrm{ij}}^{(2)}}$$

$$= \mathbf{W}_{\mathrm{jk}}^{(1)} \frac{\partial \mathrm{f}_2\left(\mathbf{x}\mathbf{W}_{:\mathrm{j}}^{(2)}\right)}{\partial \mathbf{W}_{\mathrm{ij}}^{(2)}}$$

زیرا $\frac{\partial \mathrm{f}_2\left(\mathbf{x}\mathbf{W}_{:\mathrm{l}}^{(2)}\right)}{\partial \mathbf{W}_{\mathrm{ij}}^{(2)}} = 0$ برای $\mathrm{l} \neq \mathrm{j}$. حالا قانون زنجیره‌ای را ادامه دهید

$$\frac{\partial \mathrm{f}_2\left(\mathbf{x}\mathbf{W}_{:\mathrm{j}}^{(2)}\right)}{\partial \mathbf{W}_{\mathrm{ij}}^{(2)}} = \frac{\partial \mathrm{f}_2\left(\theta_{\mathrm{j}}^{(2)}\right)}{\partial \theta_{\mathrm{j}}^{(2)}} \frac{\partial \theta_{\mathrm{j}}^{(2)}}{\partial \mathbf{W}_{\mathrm{ij}}^{(2)}} \qquad \triangleright\ \theta_{\mathrm{j}}^{(2)} = \mathbf{x}\mathbf{W}_{:\mathrm{j}}^{(2)}$$

$$= \frac{\partial \mathrm{f}_2\left(\theta_{\mathrm{j}}^{(2)}\right)}{\partial \theta_{\mathrm{j}}^{(2)}} \mathbf{x}_{\mathrm{i}}$$

با کنار هم قرار دادن اینها، می‌گیریم

$$\frac{\partial \mathrm{Err}\left(\mathbf{W}^{(1)}, \mathbf{W}^{(2)}\right)}{\partial \mathrm{W}_{\mathrm{ij}}^{(2)}} = \sum_{k=1}^{m} \frac{\partial L\left(\hat{\mathbf{y}}_k, \mathbf{y}_k\right)}{\partial \hat{\mathbf{y}}_k} \frac{\partial \mathrm{f}_1\left(\theta_{\mathrm{k}}^{(1)}\right)}{\partial \theta_{\mathrm{k}}^{(1)}} \frac{\partial \theta_{\mathrm{k}}^{(1)}}{\partial \mathbf{W}_{\mathrm{ij}}^{(2)}}$$

توجه داشته باشید که مقداری از گرادیان مانند $\boldsymbol{W}^{(1)}$ است، یعنی.

$$\delta_{\mathrm{k}}^{(1)} = \frac{\partial L(\hat{\mathbf{y}}_k, \mathbf{y}_k)}{\partial \hat{\mathbf{y}}_k} \frac{\partial \mathrm{f}_1\left(\theta_{\mathrm{k}}^{(1)}\right)}{\partial \theta_{\mathrm{k}}^{(1)}}$$

محاسبه این مؤلفه‌ها فقط باید یک بار برای $\boldsymbol{W}^{(1)}$ انجام شود و این اطلاعات دوباره منتشر شود تا گرادیان $\boldsymbol{W}^{(2)}$ بدست آید. تفاوت در گرادیان $\frac{\partial \theta^{(1)}}{\partial \mathbf{W}^{(2)}}$ است، زیرا $\boldsymbol{h}$ به $\mathbf{W}^{(2)}$ متکی است. برای $\boldsymbol{W}^{(1)}$، $\boldsymbol{h} = f_2\left(x_i \boldsymbol{W}^{(2)}\right)$ یک ثابت است، و بنابراین بر شیب $\boldsymbol{W}^{(1)}$ تأثیر نمی‌گذارد. گرادیان نهایی است

$$\frac{\partial \mathrm{Err}\left(\mathbf{W}^{(1)}, \mathbf{W}^{(2)}\right)}{\partial \mathbf{W}_{\mathrm{ij}}^{(2)}} = \left(\sum_{\mathrm{k}=1}^{\mathrm{m}} \delta_k^{(1)} \mathbf{W}_{jk}^{(1)}\right) \frac{\partial \mathrm{f}_2\left(\theta_{\mathrm{j}}^{(2)}\right)}{\partial \theta_{\mathrm{j}}^{(2)}} \mathbf{x}_i$$

$$= \left(\mathbf{W}_{\mathrm{j}:}^{(1)} \delta^{(1)}\right) \frac{\partial \mathrm{f}_2\left(\theta_{\mathrm{j}}^{(2)}\right)}{\partial \theta_{\mathrm{j}}^{(2)}} \mathbf{x}_{\mathrm{i}}$$

اگر لایه دیگری قبل از $\mathbf{W}^{(2)}$ اضافه شود، اطلاعات منتشر شده به عقب است

$$\delta_j^{(2)} = \left(W_{j:}^{(1)}\delta^{(1)}\right)\frac{\partial f_2\left(\theta_j^{(2)}\right)}{\partial\theta_j^{(2)}}$$

و x_i با $h_i^{(2)}$ جایگزین می‌شود. گرادیان برای $W_{ij}^{(3)}$ است

$$\left(\mathbf{W}_{j:}^{(2)}\delta^{(2)}\right)\frac{\partial f_3\left(\theta_j^{(3)}\right)}{\partial\theta_j^{(3)}}\mathbf{x}_i$$

مثال ۱۹: فرض کنید $p(y = 1|\boldsymbol{x})$ یک توزیع برنولی باشد، با f_1 و f_2 هر دو تابع سیگموئید. هزینه آنتروپی متقاطع است. می‌توانیم قانون به‌روزرسانی دو لایه را با این تنظیمات با وصل کردن در بالا استخراج کنیم.

$$L(\hat{y}, y) = -y\log(\hat{y}) - (1 - y)\log(1 - \hat{y}) \qquad \triangleright\ \text{cross} - \text{entropy}$$

$$\frac{\partial L(\hat{y}, y)}{\partial\hat{y}} = -\frac{y}{\hat{y}} + \frac{1 - y}{1 - \hat{y}}$$

$$f_2\left(\mathbf{x}\mathbf{W}_{:j}^{(2)}\right) = \sigma\left(\mathbf{x}\mathbf{W}_{:j}^{(2)}\right) = \frac{1}{1 + \exp\left(-\mathbf{x}\mathbf{W}_{:j}^{(2)}\right)}$$

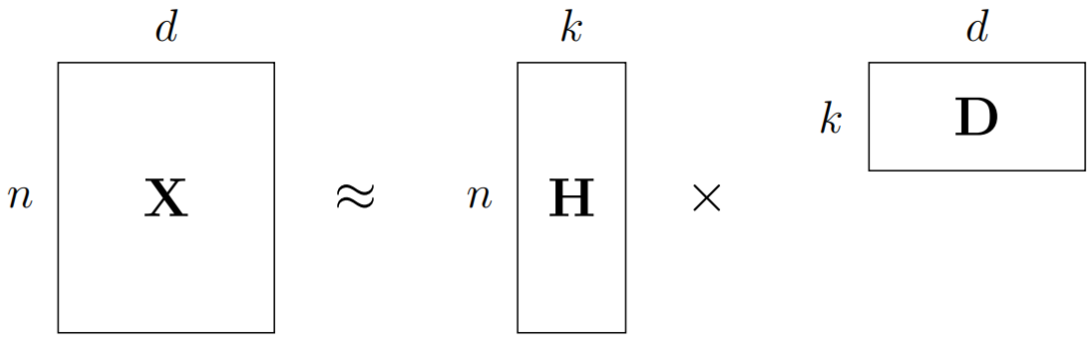
$$f_1\left(\mathbf{h}\mathbf{W}_{:k}^{(1)}\right) = \sigma\left(\mathbf{h}\mathbf{W}_{:k}^{(1)}\right) = \frac{1}{1 + \exp\left(-\mathbf{h}\mathbf{W}_{:k}^{(1)}\right)}$$

$$\partial\sigma(\theta) = \sigma(\theta)(1 - \sigma(\theta))$$

اکنون می‌توانیم به‌روزرسانی پس انتشار را با انتشار به جلو محاسبه کنیم

$$\mathbf{h} = \sigma\left(\mathbf{x}\mathbf{W}^{(2)}\right)$$

$$\hat{y} = \sigma\left(\mathbf{h}\mathbf{W}^{(1)}\right)$$



شکل ۹٫۴ : فاکتورسازی ماتریسی ماتریس داده $X \in \mathbb{R}^{n\times d}$.

و سپس انتشار گرادیان به عقب

$$\delta_k^{(1)} = \frac{\partial L(\hat{\mathbf{y}}_k, \mathbf{y}_k)}{\partial\hat{\mathbf{y}}_k}\frac{\partial f_1\left(\theta_k^{(1)}\right)}{\partial\theta_k^{(1)}}$$

$$= \left(-\frac{\mathbf{y}k}{\hat{\mathbf{y}}k} + \frac{1-\mathbf{y}_k}{1-\hat{\mathbf{y}}_k}\right)\hat{\mathbf{y}}_k(1-\hat{\mathbf{y}}_k)$$

$$= -\mathbf{y}_k(1-\hat{\mathbf{y}}_k) + (1-\mathbf{y}_k)\hat{\mathbf{y}}_k$$

$$= \hat{\mathbf{y}}_k - \mathbf{y}_k$$

$$\frac{\partial}{\partial \mathbf{W}_{jk}^{(1)}} = \delta_k^{(1)}\mathbf{h}_j$$

$$\delta_j^{(2)} = \left(\mathbf{W}_{j:}^{(1)}\delta^{(1)}\right)\mathbf{h}_j(1-\mathbf{h}_j)$$

$$\frac{\partial}{\partial \mathbf{W}_{ij}^{(2)}} = \delta_j^{(2)}\mathbf{x}_i$$

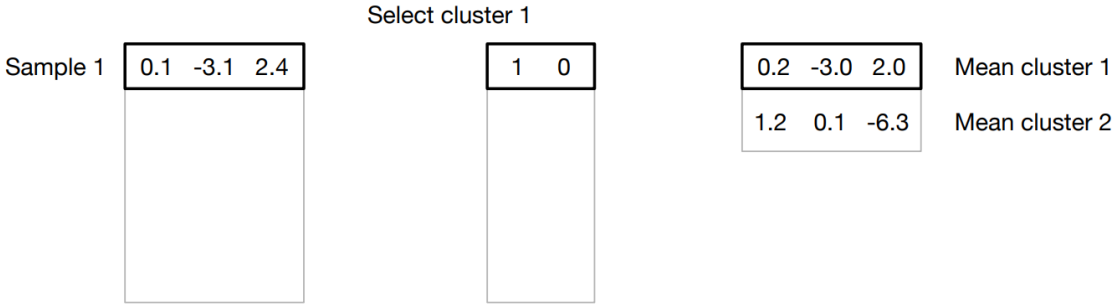
به روزرسانی به سادگی شامل گام برداشتن در جهت این شیب‌ها است، همانطور که برای گرادیان کاهشی معمول است. ما با برخی از $\mathbf{W}^{(1)}$ و $\mathbf{W}^{(2)}$ اولیه شروع می‌کنیم (مثلاً با مقادیر تصادفی پر شده است)، و سپس قوانین گرادیان کاهشی را با این گرادیان‌ها اعمال می‌کنیم.

۹-۲-۲ یادگیری بدون نظارت[1] و فاکتورسازی ماتریس[2]

استراتژی دیگر برای به دست آوردن یک نمایش جدید از طریق فاکتورسازی ماتریس است. ماتریس داده $\boldsymbol{X}$ به یک فرهنگ لغت $\boldsymbol{D}$ و یک پایه یا نمایش جدید $\boldsymbol{H}$ تبدیل می‌شود (شکل ۹,۴ را ببینید). در واقع، بسیاری از الگوریتم‌های یادگیری بدون نظارت (مانند کاهش ابعاد، کدگذاری پراکنده) و الگوریتم‌های یادگیری نیمه‌نظارت‌شده (مانند یادگیری فرهنگ لغت نظارت‌شده) در واقع می‌توانند به‌عنوان عامل‌بندی ماتریسی فرموله شوند. ما به عنوان مثال به خوشه‌بندی $\mathrm{k-means}$ و تجزیه و تحلیل اجزای اصلی نگاه خواهیم کرد. الگوریتم‌های باقی مانده به سادگی در جدول زیر خلاصه شده‌اند. این رویکرد کلی برای به دست آوردن یک نمایش جدید با استفاده از فاکتورسازی، **یادگیری فرهنگ لغت**[3] نامیده می‌شود

خوشه‌بندی $\mathbf{K-means}$ یک مشکل یادگیری بدون نظارت برای گروه‌بندی نقاط داده در k خوشه‌ها با به مینیمم رساندن فاصله تا میانگین هر خوشه است. این مشکل معمولاً به عنوان یک رویکرد یادگیری بازنمایی در نظر گرفته نمی‌شود، زیرا شماره خوشه معمولاً به عنوان یک نمایش استفاده نمی‌شود. با این حال، با این حال، ما با $\mathrm{k-means}$ شروع می‌کنیم زیرا این یک مثال شهودی از این است که چگونه این الگوریتم‌های یادگیری بدون نظارت را می‌توان به عنوان ماتریس در نظر گرفت.

[1] Unsupervised learning
[2] matrix factorization
[3] dictionary learning



شکل ۹٫۵: $K-means$ خوشه‌بندی به عنوان فاکتورسازی ماتریس برای ماتریس داده $X \in \mathbb{R}^{n\times d}$.

فاکتورسازی علاوه بر این، رویکرد خوشه‌بندی را می‌توان به عنوان یک رویکرد یادگیری بازنمایی در نظر گرفت، زیرا این یک گسسته‌سازی آموخته‌شده از فضا است. ما این دیدگاه $\mathrm{k-means}$ را پس از بحث در مورد آن به عنوان فاکتورسازی ماتریسی مورد بحث قرار خواهیم داد.

تصور کنید که شما دو خوشه دارید $(\mathrm{k}=2)$، با ابعاد داده $\mathbf{d}=3$. فرض کنید $\boldsymbol{d}_1$ میانگین برای خوشه 1 و $\boldsymbol{d}_2$ میانگین برای خوشه 2 باشد. هدف این است که فاصله ℓ_2 هر نقطه داده $\boldsymbol{x}$ را به مینیمم برسانید. به مرکز خوشه آن

$$\left\|\boldsymbol{x}-\sum_{\boldsymbol{i}=\mathbf{1}}^{\mathbf{2}}1(\mathbf{x}\text{ in cluster i})\mathrm{d_i}\right\|_2^2=\left|\left|\mathbf{x}-\mathbf{hD}\right|\right|_{\mathbf{2}}^{\mathbf{2}}$$

بردارهای خوشه‌های مختلف $\mathbf{h}$ برای هر $\mathbf{x}$ آموخته می‌شوند، اما فرهنگ لغت میانگین‌ها بین تمام نقاط داده مشترک است. بهینه‌سازی مشخص‌شده باید فرهنگ لغت $\mathbf{D}$ را انتخاب کند که کمترین فاصله را تا نقاط در مجموعه داده آموزشی فراهم می‌کند.

که در آن $\boldsymbol{h}=[1\ 0]$ یا $\mathbf{h}=[0\ 1]$ و $\boldsymbol{D}=[\boldsymbol{d}_1;\boldsymbol{d}_2]$. یک مثال در شکل ۹٫۵ نشان داده شده است. برای یک نقطه $\boldsymbol{x}=[0.1\ -\ 3.1\ 2.4]$، به این معنی که در خوشه 1 با میانگین $\mathbf{h}=[1\ 0]$ قرار می‌گیرد. $\mathbf{d}_1=[0.2-3.02.0]$ قرار دادن $\mathbf{x}$ در خوشه 2 که میانگین آن متفاوت‌تر است، خطای بیشتری دارد: $\boldsymbol{d}_2=[1.20.1-6.3]$.

$$\begin{matrix}min\\ \mathbf{H}\in\{0,1\}^{\mathrm{n\times k},\mathbf{1H=1}}\\ \mathbf{D}\in\mathbb{R}^{\mathrm{k\times d}}\end{matrix}\left|\left|\boldsymbol{X}-\boldsymbol{HD}\right|\right|_F^2$$

تجزیه و تحلیل مولفه‌های اصلی[1] (PCA) یک تکنیک کاهش ابعاد استاندارد است که در آن داده‌های ورودی $\mathbf{x}\in\mathbb{R}^{1\times\mathrm{d}}$ به ابعاد $\boldsymbol{h}\in\mathbb{R}^{1\times k}$ با فضای مولفه‌های اصلی پیش‌بینی می‌شود. این مولفه‌های اصلی جهت ماکسیمم واریانس در داده‌ها هستند. برای به دست آوردن این k مولفه‌های اصلی $\boldsymbol{D}\in\mathbb{R}^{k\times d}$، روش حل رایج به دست آوردن تجزیه ارزش منفرد ماتریس داده $\boldsymbol{X}=\boldsymbol{U\Sigma V}^T\in\mathbb{R}^{n\times d}$ است که

$$\mathbf{D}=\mathbf{V}_\mathrm{k}^\mathrm{T}\in\mathbb{R}^{\mathrm{k\times d}}$$

$$\mathbf{H}=\mathbf{U}_\mathrm{k}\mathbf{\Sigma}_\mathrm{k}\in\mathbb{R}^{\mathrm{n\times k}}$$

[1] Principal components analysis

جایی که $\boldsymbol{\Sigma}_k \in \mathbb{R}^{k\times k}$ از بزرگترین مقادیر k مفرد (به ترتیب نزولی) تشکیل شده است و $\boldsymbol{U}_k \in \mathbb{R}^{n\times k}$ و $\boldsymbol{V}_k \in \mathbb{R}^{k\times d}$ بردارهای منفرد متناظر هستند، یعنی $\boldsymbol{U}_k = \boldsymbol{U}_{:,1:k}$ و $\boldsymbol{V}_k = \boldsymbol{V}_{:1:k.}$ نمایش جدید برای $\boldsymbol{X}$ (با استفاده از PCA) این $\boldsymbol{H}$ است. توجه داشته باشید که PCA ویژگی‌های زیر را انتخاب نمی‌کند، بلکه ویژگی‌های جدیدی ایجاد می‌کند: $\boldsymbol{h}$ تولید شده زیرمجموعه‌ای از x اصلی نیست.

این تکنیک کاهش ابعاد را می‌توان به عنوان یک فاکتورسازی ماتریسی نیز فرموله کرد. بهینه‌سازی مربوطه نشان داده شده است

$$\underset{\mathbf{D}\in\mathbb{R}^{\mathrm{k}\times\mathrm{d}},\mathbf{H}\in\mathbb{R}^{\mathrm{n}\times\mathrm{k}}}{min}\left|\left|\boldsymbol{X}-\boldsymbol{HD}\right|\right|_{\boldsymbol{F}}^{\mathbf{2}}$$

یک راه ساده برای فهمیدن چرایی این امر، یادآوری قضیه معروف $Eckart-Young\ Mirsky$ است که ماتریس K رتبه ای $\widehat{\boldsymbol{X}}$ که بهترین تقریب $\boldsymbol{X}$ را بر حسب نرم فروبنیوس مینیمال[1] دارد، $\widehat{\boldsymbol{X}}=\boldsymbol{U}_k\boldsymbol{\Sigma}_k\boldsymbol{V}_k^T$ است.

مانند خوشه‌بندی $\mathrm{k}-\mathrm{means}$، ممکن است به سختی بفهمیم که چرا $\mathbf{h}$ تولید شده توسط PCA می‌تواند به عنوان یک نمایش مفید باشد. در واقع، PCA اغلب برای تجسم استفاده می‌شود، و بنابراین همیشه برای یادگیری بازنمایی استفاده نمی‌شود. برای تجسم، طرح ریزی اغلب به دو یا سه بعدی تهاجمی است. با این حال، به طور کلی، طرح ریزی به ابعاد پایین این خاصیت را دارد که نویز را حذف می‌کند و فقط معنی دارترین جهت‌ها را حفظ می‌کند. بنابراین، این پیش‌بینی با کاهش تعداد ویژگی‌ها و ترویج تعمیم، با جلوگیری از تطابق بیش از حد با نویز به سرعت یادگیری کمک می‌کند.

کدگذاری پراکنده رویکرد متفاوتی دارد، جایی که داده‌های ورودی به یک نمایش پراکنده گسترش می‌یابد. کدگذاری پراکنده از نظر بیولوژیکی [۱۵] بر اساس فعالیت‌های پراکنده برای حافظه در مغز پستانداران است. تفسیر دیگر این است که کدگذاری پراکنده به طور موثر فضا را گسسته می‌کند، مانند خوشه‌بندی $\mathrm{k}-\mathrm{means}$، اما با خوشه‌های همپوشانی و مقدار مربوط به اینکه چقدر یک نقطه به آن خوشه تعلق دارد.

یک استراتژی معمول برای به دست آوردن نمایش‌های پراکنده استفاده از تنظیم کننده پراکنده در نمایش آموخته شده $\mathbf{h}$ است. این مربوط به بهینه‌سازی است

$$\underset{\mathbf{D}\in\mathbb{R}^{k\times d},\mathbf{H}\in\mathbb{R}^{n\times k}}{min}\left|\left|\boldsymbol{X}-\boldsymbol{HD}\right|\right|_{\boldsymbol{F}}^{\mathbf{2}}+\lambda\sum_{\mathrm{i}=1}^{\mathrm{k}}\left|\left|\mathbf{H}_{:\mathrm{i}}\right|\right|_1+\lambda\sum_{\mathrm{i}=1}^{\mathrm{k}}\left|\left|\mathbf{D}_{\mathrm{i}:}\right|\right|_2^2$$

همانطور که در بخش ۵،۴،۲ مورد بحث قرار گرفت، تنظیم‌کننده ℓ_1 ورودی‌های صفر شده را گسترش می‌دهد، و بنابراین $\boldsymbol{H}$ را با ماکسیمم صفرهای ممکن ترجیح می‌دهد. یک تنظیم کننده نیز به $\boldsymbol{D}$ اضافه می‌شود تا اطمینان حاصل شود که $\boldsymbol{D}$ خیلی بزرگ نمی‌شود. در غیر این صورت، تمام وزن در $\boldsymbol{DH}$ به $\boldsymbol{D}$ منتقل می‌شود.

به طور کلی، انواع مختلفی از الگوریتم‌های یادگیری بدون نظارت وجود دارد که در واقع با فاکتورسازی ماتریس داده‌ها مطابقت دارند. جزئیات بیشتر در ضمیمه $\boldsymbol{D}$ داده شده است. ما همچنین جزئیاتی را در مورد نحوه یادگیری این فاکتورگیری‌ها در پیوست ارائه می‌دهیم. همانند الگوریتم‌های قبلی، آنها به سادگی بر روی متغیرهای (ماتریس) نزولی هستند. تنها تمایز در اینجا این است که استفاده از نزول مختصات بلوکی به جای الگوریتم گرادیان کاهشی استانداردتر رایج است. این تمایز جزئی است و استفاده از گرادیان کاهشی استاندارد کاملاً معتبر است.

[1] minimal Frobenius

فصل ۱۰

ارزیابی الگوریتم‌های یادگیری

اکثر این کتاب بر روی استخراج الگوریتم و به دست آوردن مدل‌ها متمرکز شده است، اما ما هنوز باید به نحوه ارزیابی این مدل‌ها بپردازیم. فرمالیسم حداکثر احتمال برای استخراج الگوریتم‌های یادگیری برخی از نتایج سازگاری را ارائه می‌دهد، که در حد نمونه‌ها می‌توانیم نقطه همگرایی یک برآوردگر را مورد بحث قرار دهیم. با این حال، در عمل، ما می‌خواهیم الگوریتم‌ها را بر اساس یک نمونه محدود ارزیابی کنیم. محیطی را تصور کنید که در آن دو مدل را یاد می‌گیرید، مثلاً با استفاده از رگرسیون لجستیک با دو پارامتر تنظیم متفاوت. کدام یک از این دو مدل "بهتر" است؟ حتی بهتر است بگوییم چه معنایی دارد؟ آیا می‌خواهید بگویید که مدل برای این مشکل (تنظیم داده‌ها) بهتر است یا برای مشکلات متعدد؟ آیا ما سعی می‌کنیم الگوریتم‌ها یا مدل‌های به دست آمده را با هم مقایسه کنیم. از یک نمونه خاص از یک الگوریتم؟ چگونه می‌توانیم مطمئن باشیم که عملکرد اندازه گیری شده دقیقاً عملکردی را که انتظار داریم در داده‌های جدید ببینیم منعکس می‌کند؟ سوالاتی در مورد ویژگی‌های آن هدف و ویژگی‌های تجربی مدل‌های آموخته شده.

در این فصل، ابزارهای نظری و تجربی را برای ارزیابی بهتر ویژگی‌های الگوریتم‌های یادگیری ارائه می‌کنیم. ما با برخی از نتایج نظری نمونه محدود اولیه شروع می‌کنیم، که پیچیدگی کلاس مدل را به تعداد نمونه‌های مورد نیاز برای به دست آوردن یک تخمین منطقی از خطای مورد انتظار (خطای تعمیم) مرتبط می‌کند. این بخش همچنین ایده‌های بهینه سازی در یک کلاس تابع و اهداف ما برای به دست آوردن بهترین مدل از نظر خطای تعمیم را معرفی می‌کند. حوزه ای که با این نوع خصوصیات نظری سروکار دارد، نظریه یادگیری آماری نامیده می‌شود. ما یک نتیجه را با استفاده از نابرابری‌های تمرکز و پیچیدگی $Rademacher$ برای توصیف پیچیدگی کلاس مدل مورد بحث قرار خواهیم داد. برای اطلاعات بیشتر، می‌توانید این آموزش را در مورد موضوع [۷] در نظر بگیرید.

سپس، نحوه مقایسه تجربی الگوریتم‌ها را مورد بحث قرار خواهیم داد. در اکثر تنظیمات دنیای واقعی، شما بین الگوریتم‌ها بر اساس عملکرد آنها در داده‌های موجود، یکی را انتخاب خواهید کرد. شما می‌خواهید این انتخاب منعکس کننده میزان عملکرد آن الگوریتم‌ها بر روی داده‌های جدید باشد. برای رسیدن به این هدف، ما در مورد چگونگی تقسیم داده‌ها و نحوه استفاده از آزمون‌های معناداری آماری برای ارائه سطحی از اطمینان به اینکه یک الگوریتم یا مدل از دیگری بهتر است، تحت برخی معیارهای خاص بحث خواهیم کرد. ما به ندرت قادر به نتیجه گیری قوی بر اساس آزمایش خواهیم بود، اما می‌توانیم شواهدی در مورد ویژگی‌های الگوریتم ایجاد کنیم.

این ابزارها مسلماً حیاتی ترین جنبه‌های استفاده صحیح از الگوریتم‌های یادگیری ماشین در عمل هستند. می‌توان یک مدل پیچیده را یاد گرفت، اما بدون درک نحوه عملکرد آن در عمل روی داده‌های جدید، استفاده واقعی از این مدل‌ها امکان‌پذیر

نیست. این که آیا یک الگوریتم برای مقاصد علمی استفاده می‌شود یا در سیستم‌های واقعی به کار می‌رود، داشتن درک درستی از ویژگی‌های آن چه از نظر تئوری و چه از لحاظ تجربی، کلیدی برای به دست آوردن نتایج مورد انتظار است. این فصل فقط شروع به خراش دادن سطح این ابزارها می‌کند، با این هدف که علاقه شما را برانگیزد و شما را به سمت مطالب بیشتری برای یادگیری در مورد ارزیابی هدایت کند.

۱-۱۰ مقدمه‌ای کوتاه بر مرزهای تعمیم

هدف ما در سراسر این کتاب این بوده است که تابعی را بر اساس مجموعه‌ای از مثال‌ها به دست آوریم که به دقت پیش‌بینی می‌کند: خطای کم مورد انتظار را در فضای نمونه‌های ممکن ایجاد می‌کند. با این حال، ما نمی‌توانیم خطای مورد انتظار را اندازه گیری کنیم. از نظر آماری می‌دانیم که با یک نمونه کافی می‌توانیم یک انتظار را تقریبی کنیم. در اینجا، ما این را با دقت بیشتری برای توابع آموخته شده تعیین می‌کنیم.

هدف ما به‌طور دقیق‌تر انتخاب تابعی از یک تابع کلاس H برای به حداقل رساندن یک تابع هزینه است: $\ell: \mathbb{R} \times \mathbb{R} \rightarrow [0, \infty)$ در انتظار روی همه جفت‌ها $(\mathbf{x}, \mathrm{y})$

$$\min_{\mathrm{f} \in \mathcal{H}} \mathbb{E}[\ell(\mathrm{f}(\mathbf{X}), \mathbf{Y})]$$

به عنوان مثال، در رگرسیون خطی، $\mathcal{H} = \{f : \mathbb{R}^d \rightarrow \mathbb{R} \mid f(x) = \boldsymbol{x}^T \boldsymbol{w}, \text{for any w} \in \mathbb{R}^d\}$. این فضای توابع $\mathcal{H}$ همه توابع خطی ممکن ورودی $x \in \mathbb{R}^d$ را برای تولید یک خروجی اسکالر نشان می‌دهد. هدف ما در رگرسیون خطی، به حداقل رساندن یک پروکسی برای خطای واقعی مورد انتظار، یعنی خطای نمونه بود: $\frac{1}{n} \sim_{i=1}^{n} \ell\left(\mathrm{f}(\mathbf{x}_\mathrm{i}), \mathrm{y}_\mathrm{i}\right)$ اکنون یک سوال طبیعی مطرح می‌شود: آیا این خطای نمونه تخمین دقیقی از خطای مورد انتظار واقعی ارائه می‌دهد؟ و در مورد عملکرد تعمیم واقعی، یعنی خطای واقعی مورد انتظار چه چیزی به ما می‌گوید؟

بیایید با یک مثال ساده با استفاده از رگرسیون خطی شروع کنیم. یک تابع محدود کلاس $\mathcal{H}$ را فرض کنید، که در آن $\mathcal{H} = \{\mathrm{f}: \mathbb{R}^\mathrm{d} \rightarrow \mathbb{R} | \mathrm{f}(\mathrm{x}) = \mathbf{x}^\mathrm{T}\mathbf{w}, \text{for any } \mathbf{w} \in \mathbb{R}^\mathrm{d} \text{ such that } ||\mathbf{w}||_2 \leq \mathbf{B}_w\}$ برای برخی از اسکالرهای محدود $\mathbf{B}_w > 0$. فرض کنید ویژگی‌های ورودی از یک فضای محدود می‌آیند، به طوری که برای همه $\mathbf{x}$، $||\mathbf{x}||_2 \leq \mathbf{B}_\mathrm{x}$ برای برخی اسکالر محدود $\mathbf{B}_x > 0$، و همچنین خروجی $y \in [-\mathrm{B}_\mathrm{y}, \mathrm{B}_\mathrm{y}]$ برای برخی از $B_y > 0$. فرض کنید از اتلاف $\ell(\hat{y}, y) = \frac{1}{2}(\hat{y}, y)^2$ استفاده می‌کنیم که (به صورت محلی) $Lipschitz$ پیوسته است. برای منطقه محدود ما، با ثابت $Lipschitz$ $c = B_y + B_x B_w$. این به این دلیل است که $|\hat{y}| \leq \mathrm{B_x B}_w$ و

$$\left|\frac{d\ell(\hat{y}, y)}{d\hat{y}}\right| = = |\hat{y} - \mathrm{y}| \leq |\hat{y}| + |\mathrm{y}| \leq \mathrm{B_y} + \mathrm{B_x B}_w$$

بعلاوه، چون $\mathrm{y} \in [-\mathrm{B_y}, \mathrm{B_y}]$، می‌دانیم که هزینه به صورت محدود می‌شود

$$\ell(\hat{y}, y) = \frac{1}{2}(\hat{y}, y)^2 \leq \frac{1}{2}(\mathrm{B_y^2} + \mathrm{B_x^2 B_w^2})$$

برای خطای تقریبی

$$\widehat{\text{Err}}\,\text{d(f)} = \frac{1}{\text{n}}\sum_{i=1}^{n} \ell(f(\boldsymbol{x_i}), \boldsymbol{y_i})$$

و خطای واقعی

$$\text{Err(f)} \;=\; \mathbb{E}[\ell(\text{f}(\mathbf{X}), \mathbf{Y}) = \int_{\boldsymbol{x}\times y} \text{p}(\mathbf{x}, \mathbf{y})\boldsymbol{\ell}(\mathbf{f}(\mathbf{x}), \boldsymbol{y})\boldsymbol{dxdy}$$

با استفاده از معادله ۱۰,۲ زیر، دریافت می‌کنیم که با احتمال $\delta \;-\; 1$، برای $\delta \in (0,\, 1]$

$$\text{Err(f)} \leq \widehat{\text{Err}}\text{d(f)} + \frac{2\text{cB}_\text{x}\text{B}_w}{\sqrt{n}} + \frac{1}{2}\left(\text{B}_\text{y}^2 + \text{B}_\text{x}^2\text{B}_w^2\right) \qquad (10.1)$$

با افزایش نمونه‌های n، دو عبارت دوم ناپدید می‌شوند و خطای نمونه به خطای واقعی مورد انتظار نزدیک می‌شود. این کران میزان ناپدید شدن این اختلاف را نشان می‌دهد. برای اطمینان بیشتر - δ کوچک که $\ln(1/\delta)$را بزرگتر می‌کند - به نمونه‌های بیشتری نیاز است تا ترم سوم کوچک باشد. این جمله سوم با استفاده از نابرابری‌های غلظت به دست می‌آید، که ما را قادر می‌سازد نرخی را بیان کنیم که میانگین نمونه به مقدار مورد انتظارش نزدیک می‌شود. برای مقادیر احتمالاً بزرگ ویژگی‌ها یا وزن‌های آموخته شده، عبارت دوم می‌تواند بزرگ باشد و دوباره به نمونه‌های بیشتری نیاز دارد. عبارت دوم ویژگی‌های کلاس تابع ما را منعکس می‌کند: یک کلاس ساده‌تر، با وزن‌های محدود کوچک، می‌تواند تخمین دقیق‌تری از هزینه در تعداد کمتری از نمونه‌ها داشته باشد. به طور کلی تر، این اندازه گیری پیچیدگی، پیچیدگی Rademacher نامیده می‌شود. 1 برای توابع خطی بالا، با نرم‌های 2 محدود برای $\boldsymbol{x}$، $\boldsymbol{w}$، پیچیدگی $Rademacher$ به صورت $R_n(\mathcal{H}) \leq B_x B_y/\sqrt{n}$ محدود می‌شود (به [۱۰، معادله ۳] مراجعه کنید).

در چند بخش بعدی، یک نتیجه تعمیم برای توابع کلی تر و همچنین پیش زمینه مورد نیاز برای تعیین آن نتیجه ارائه می‌دهیم.

۱-۱-۱۰ نابرابری‌های تمرکز

ما استفاده از نابرابری‌های تمرکز را با یک مثال رایج بررسی خواهیم کرد: نابرابری $Hoeffding$. برای تعمیم کران زیر، یک تعمیم به نام نابرابری $McDiarmid$ استفاده می‌شود

برای $i.i.d.$ متغیرهای تصادفی $X_1, \ldots, X_n$، به طوری که $0 \;\leq\; X_i \;\leq\; 1$، اجازه دهید $\bar{X} \;=\; \frac{1}{n}\sum_{i=1}^{n} X_i$ میانگین نمونه باشد. سپس نابرابری هوفدینگ بیان می‌کند که برای هر کدام

$$\Pr(\bar{\mathbf{X}} - \mathbb{E}[\bar{\mathbf{X}}] \geq \epsilon) \leq \exp(-2\text{n}\epsilon^2)$$

ما با تنظیم این مقدار احتمال روی δ شروع می‌کنیم، به طوری که می‌توانیم با احتمال δ، $\Pr(\bar{X} \;-\; \mathbb{E}[\bar{X}] \;\geq$ function$(\delta))$ بگوییم. ما می‌توانیم برای را بر حسب δ حل کنیم تا بدست آوریم

$$\delta \;=\; \exp(-2\text{n}\,\epsilon\text{^}2\,) \;\Longrightarrow \epsilon \;=\; \pm\sqrt{\frac{\ln(1/\delta)}{2\text{n}}}$$

ما می‌توانیم برای محدود کردن $\overline{\mathrm{X}}$ به نزدیک $\mathbb{E}[\overline{\mathrm{X}}]$ از بالا و پایین ϵ را روی $\sqrt{\frac{\ln(1/\delta)}{2n}}$ یا $-\sqrt{\frac{\ln(1/\delta)}{2n}}$ تنظیم کنیم، ما آن را با احتمال $1-\delta$ دریافت می‌کنیم، $|\bar{X} - \mathbb{E}[\overline{\mathrm{X}}]| \leq |\epsilon| = \sqrt{\frac{\ln(1/\delta)}{2n}}$

این نابرابری غلظت، مفروضات کمی در مورد متغیرهای تصادفی ایجاد می‌کند و به هیچ فرض توزیعی نیاز ندارد. در نتیجه، نرخ همگرایی به میانگین واقعی تنها $1/\sqrt{n}$ است. نرخ‌های سریع تری را می‌توان با فرضیات بیشتر به دست آورد.

۲-۱-۱۰ پیچیدگی یک کلاس تابع

پیچیدگی $Rademacher$ یک کلاس تابع، توانایی بیش از حد برازش توابع را در یک نمونه خاص مشخص می‌کند. کلاس توابعی که معمولاً پیچیده‌تر هستند، به احتمال زیاد می‌توانند نویز تصادفی را تطبیق دهند و بنابراین پیچیدگی $Rademacher$ بالاتری دارند. پیچیدگی تجربی $Rademacher$، برای نمونه $z_1، \ldots ،z_n$ - جایی که معمولا $z_i = (x_i, y_i)$ را در نظر می‌گیریم - به صورت تعریف می‌شود

$$\widehat{\mathrm{R}}_{\mathrm{n}}(\mathcal{H}) = \mathbb{E}\left[\max_{\mathrm{f} \in \mathcal{H}} \frac{1}{\mathrm{n}} \sum_{\mathrm{i}=1}^{\mathrm{n}} \sigma_{\mathrm{i}} \mathrm{f}(\mathbf{x}_{\mathrm{i}})\right]$$

جایی که انتظار به پایان رسیده است $\mathrm{i.i.d.}$ متغیرهای تصادفی $\sigma_1, \ldots, \sigma_n$ به طور یکنواخت از $\{-1, 1\}$ انتخاب شده است. این انتخاب نشان می‌دهد که چگونه کلاس تابع می‌تواند با این نویز تصادفی ارتباط داشته باشد. به عنوان مثال اگر $f(x)$ مقدار 1 یا -1 را پیش‌بینی می‌کند، مانند طبقه‌بندی باینری، در نظر بگیرید. اگر تابعی در کلاس توابع وجود داشته باشد که بتواند کاملاً با علامت σ_i نمونه برداری تصادفی مطابقت داشته باشد، آن تابع بالاترین مقدار $\sum_{i=1}^{n} \sigma_i f(\boldsymbol{x}_i)$ را تولید می‌کند. پیچیدگی تجربی $Rademacher$ برای یک کلاس تابع زیاد است، اگر برای هر $\sigma\mathrm{i}$ نمونه‌برداری تصادفی، چنین تابعی در کلاس تابع وجود داشته باشد (می‌تواند برای هر $\sigma_1, \ldots, \sigma_n$ یک تابع متفاوت باشد). پیچیدگی $Rademacher$ پیچیدگی تجربی $Rademacher$ مورد انتظار است، بیش از همه نمونه‌های احتمالاً از n نمونه

برای کلاس‌های تابع با پیچیدگی $Rademacher$ بالا، خطا در مجموعه آموزشی بعید است که منعکس کننده خطای تعمیم باشد، تا زمانی که تعداد نمونه کافی وجود داشته باشد. این در تعمیم محدود در بخش ۳ − ۱ − ۱۰ منعکس شده است.

اتصال به بعد VC: پیچیدگی یک کلاس تابع را می‌توان با بعد VC نیز مشخص کرد. ایده بعد VC برای مشخص کردن تعداد نقاطی که می‌توانند توسط یک کلاس تابع جدا شوند (یا خرد شوند). توابع ساده ابعاد VC پایینی دارند، زیرا به اندازه کافی پیچیده نیستند تا نقاط زیادی را از هم جدا کنند. توابع پیچیده تر، که مرزهای پیچیده را فعال می‌کنند، بعد VC بالاتری دارند. به عنوان مثال، برای توابع به شکل $\mathrm{f}\big((\mathrm{x}_1, \mathrm{x}_2)\big) = \mathrm{sign}(\mathrm{x}_1 w_1 + \mathrm{x}_2 w_2 + w_0)$ بعد VC برابر 3 است. به طور کلی، برای $x \in \mathbb{R}^d$، بعد VC برابر $\mathrm{d} + 1$ است. بعد VC ایده ای مشابه با پیچیدگی $\mathrm{Rademacher}$ است، اما به طبقه بندی کننده‌های باینری محدود می‌شود. به همین دلیل، ما به طور مستقیم پیچیدگی $\mathrm{Rademacher}$ را مورد بحث قرار می‌دهیم، که برای طبقه بندی کننده‌های باینری می‌تواند برحسب بعد VC محدود شود. با لمای Sauer، ما معمولاً می‌توانیم پیچیدگی $\mathrm{Rademacher}$ یک کلاس فرضیه را با $\sqrt{\frac{2\mathrm{VC-dimension}\ln \mathrm{n}}{\mathrm{n}}}$ محدود کنیم.

۱۰-۱-۳ مرزهای تعمیم

کران تعمیم برای یک کلاس از مدل‌ها را می‌توان با ترکیب نابرابری‌های غلظت به انحراف کران از میانگین برای نمونه‌های کمتر، و استفاده از پیچیدگی $Rademacher$ برای محدود کردن تفاوت بین خطای نمونه و خطای واقعی مورد انتظار در همه توابع در کلاس تابع ما علاوه بر این باید مجموعه زیان‌ها را محدود کنیم. ما فرض می‌کنیم که هزینه‌های $Lipschitz$ با ثابت c هستند، به این معنی که آنها خیلی سریع در یک منطقه تغییر نمی‌کنند، با c نشان دهنده نرخ تغییر است. علاوه بر این، ما همچنین فرض می‌کنیم که هزینه با b محدود می‌شود، یعنی به مقادیر $[-b, b]$ می‌رسد. همانطور که در بالا، اگر $z_1, \ldots, z_n$ یک $i.i.d$ باشد، سپس با احتمال $1 - \delta$، برای هر $f \in \mathcal{H}$،

$$\mathbb{E}[\ell(f(\mathbf{X}), Y)] \leq \frac{1}{n} \sum_{i = 1}^{n} \{\ell(f(x_i), y_i) + 2c R_n(\mathcal{H}) + b\sqrt{\frac{\ln(1/\delta)}{2n}} \qquad (10.2)$$

برای بیان قضیه و اثبات دقیق تر، به [۳، قضیه ۷] و [۱۰، قضیه ۱] مراجعه کنید.

۱۰-۲ مقایسه الگوریتم‌های یادگیری

برای ارزیابی تجربی الگوریتم‌ها، می‌توانیم تنظیماتی را با یک یا چند الگوریتم در یک یا چند مجموعه داده در نظر بگیریم. بسته به تنظیمات، ارزیابی‌های متفاوتی به کار گرفته می‌شود. برای یک نمای کلی خوب از ارزیابی الگوریتم‌های یادگیری ماشین، [۹] را ببینید.

در حال حاضر، اجازه دهید با یک مورد ساده شروع کنیم، که در آن دو الگوریتم را با هم مقایسه کرده و از تست دو جمله ای استفاده می‌کنیم. فرض کنید مجموعه ای از مسائل یادگیری $D_1, D_2, \ldots D_m$ و مایل به مقایسه الگوریتم‌های یادگیری a_1 و a_2هستند. ما می‌توانیم چنین مقایسه‌ای را با استفاده از آزمون شمارش به صورت زیر انجام دهیم: برای هر مجموعه داده، هر دو الگوریتم بر حسب معیار عملکرد انتخابی ارزیابی می‌شوند و الگوریتمی با دقت عملکرد بالاتر برنده می‌شود، در حالی که به دیگری هزینه داده می‌شود. (در صورت عملکرد دقیقاً یکسان، ما می‌توانیم یک برد / باخت به صورت تصادفی ارائه دهیم).

	$\mathcal{D}_1$	$\mathcal{D}_2$	$\mathcal{D}_3$	$\mathcal{D}_4$		$\mathcal{D}_{m-1}$	$\mathcal{D}_m$
a_1	1	0	1	1	$\cdots$	0	1
a_2	0	1	0	0		1	0

جدول ۱۰/۱: یک آزمون شمارش که در آن الگوریتم های یادگیری a_1 و a_2 بر روی مجموعه ای از m مجموعه داده های مستقل مقایسه می شوند. الگوریتمی با عملکرد بهتر در یک مجموعه داده خاص، برد (1) را جمع آوری می کند، در حالی که الگوریتم دیگر خزینه (0) را جمع آوری می کند.

ما اکنون علاقه مند به ارائه شواهد آماری هستیم که می‌گوید الگوریتم a_1 بهتر از الگوریتم a_2 است. فرض کنید a_1 تعداد kبرد از m داشته باشد و الگوریتم a_2 دارای $m - k$ برد باشد، همانطور که در جدول ۱۰،۱ نشان داده شده است. ما می‌خواهیم فرضیه صفر H_0 را ارزیابی کنیم که الگوریتم‌های a_1 و a_2 عملکرد یکسانی دارند با ارائه یک فرضیه جایگزین H_1 مبنی بر اینکه الگوریتم a_1 بهتر از a_2 است. به اختصار،

$$\mathrm{H0}: \mathrm{quality}(\mathrm{a}_1) = \mathrm{quality}(\mathrm{a}_2)$$

$$\mathrm{H1}: \mathrm{quality}(\mathrm{a}_1) > \mathrm{quality}(\mathrm{a}_2)$$

اگر فرضیه صفر درست باشد، برد/ باخت در هر مجموعه داده به همان اندازه محتمل خواهد بود و با تغییرات جزئی تعیین می‌شود. بنابراین، احتمال برد در هر مجموعه داده تقریباً برابر با $\mathrm{p} = 1/2$ خواهد بود. اکنون، می‌توانیم با استفاده از توزیع دوجمله‌ای، این احتمال را بیان کنیم که الگوریتم a_1دارای k برنده یا بیشتر را تحت فرضیه صفر می‌برد.

$$\mathrm{P} = \sum_{\mathrm{i=k}}^{\mathrm{m}} \binom{m}{i} \mathrm{p}^{\mathrm{i}}(1-\mathrm{p})^{\mathrm{m-i}}$$

و از آن به عنوان $\mathrm{P-value}$ یاد کنید. این مقدار احتمال پیروزی k، به اضافه احتمال برد $\mathrm{k}+1$، تا احتمال m برد، تحت فرضیه صفر است. یک رویکرد معمولی در این موارد، ایجاد یک مقدار معنی‌داری است، مثلاً $\alpha = 0.05$ و رد فرضیه صفر اگر $\mathrm{P} \leq \alpha$ باشد. اگر مقدار P بزرگتر از α باشد، می‌گوییم شواهد کافی برای رد H_0 وجود ندارد. برای مقادیر P به اندازه کافی پایین، ممکن است به این نتیجه برسیم که شواهد کافی وجود دارد که الگوریتم a_1 بهتر از الگوریتم a_2 است.

انتخاب آستانه اهمیت α تا حدودی دلخواه است. به طور معمول، 5% یک مقدار معقول است، اما مقادیر پایین تر نشان می‌دهد که وضعیت خاص k برد از m بسیار بعید است، که می‌توانیم شواهدی برای رد H_0 بسیار قوی در نظر بگیریم. توانایی رد فرضیه صفر تا حدی اطمینان می‌دهد که نتیجه تصادفی رخ نداده است.

به طور کلی تر، می‌توانیم آزمون‌های معناداری آماری دیگری را بر اساس توزیع معیارهای عملکرد در نظر بگیریم. در مثال بالا، توزیع دوجمله ای مناسب بود. اگر در عوض خطاهای واقعی مجموعه داده‌ها را در نظر بگیریم، آنگاه جفت مقادیر واقعی خواهیم داشت. در این مورد، اگر هر دو خطا به طور نرمال توزیع شده باشند و واریانس مشابهی داشته باشند، یک انتخاب رایج، آزمون tزوجی است. آزمون t زوجی تفاوت‌های نمونه بین الگوریتم‌ها را می‌گیرد (خط ۳ در جدول ۱۰،۲). $\mathrm{d}_1, \ldots, \mathrm{d}_\mathrm{m}$ از آنجایی که مجدداً فرضیه صفر ما این است که الگوریتم‌ها به یک اندازه عمل می‌کنند، در فرضیه صفر میانگین این تفاوت‌ها ۰ است. اگر تفاوت‌ها به طور نرمال توزیع شوند، برای میانگین نمونه $\bar{\mathrm{d}} = \frac{1}{\mathrm{m}}\sum_{\mathrm{i}=1}^{\mathrm{m}} \mathrm{d}_\mathrm{i}$ و انحراف استاندارد نمونه $S_d = \sqrt{\frac{1}{m-1}\sum_{i=1}^{m}(d_i - \bar{d})^2}$، متغیر تصادفی $t = \frac{\bar{d}-0}{Sd/\sqrt{m}}$ بر اساس توزیع tدانش‌آموز توزیع می‌شود. توزیع $t-$ $Student$ تقریباً مانند یک توزیع نرمال است، با پارامتر درجه آزادی $m - 1$ که باعث می‌شود با بزرگتر شدن m، توزیع بیشتر شبیه توزیع نرمال به نظر برسد.

اکنون می‌توانیم در مورد احتمال این متغیر تصادفی T نسبت به آمار محاسبه شده بپرسیم. اگر ما فقط به دانستن اینکه آیا الگوریتم ۱ بهتر از الگوریتم ۲ است اهمیت می‌دهیم، یک آزمایش یک دنباله انجام می‌دهیم. اگر احتمال اینکه T بزرگتر از tباشد، یعنی $p = Pr(T > t)$، کوچک باشد، شواهدی به دست می‌آوریم که نشان می‌دهد الگوریتم ۱ بهتر از الگوریتم ۲ است. ما می‌توانیم ترتیب تفاوت را مبادله کنیم. اگر $p = Pr(T > -t)$ کوچک باشد، شواهدی به دست می‌آوریم که نشان می‌دهد الگوریتم ۲ بهتر از الگوریتم ۱ است. در عوض یک تست دو طرفه می‌پرسد که آیا این دو الگوریتم متفاوت هستند یا خیر. در این مورد، از $p = Pr(T > |t|)$ استفاده می‌شود.

	$\mathcal{D}_1$	$\mathcal{D}_2$	$\mathcal{D}_3$	$\mathcal{D}_4$		$\mathcal{D}_{m-1}$	$\mathcal{D}_m$
a_1	0.11	0.08	0.15	0.12	$\cdots$	0.07	0.09
a_2	0.10	0.09	0.11	0.12	$\cdots$	0.10	0.09
d	0.01	-0.01	0.04	0.0	$\cdots$	-0.03	0.0

جدول ۱۰/۲: جدولی از خطاها برای دو الگوریتم یادگیری a_1 و a_2 بر روی مجموعه ای از m مجموعه داده های مستقل مقایسه شده است. سطر آخر شامل تفاوت‌هایی است که برای آزمون t زوجی استفاده می شود.

۱۰-۳ به دست آوردن نمونه‌های خطا

یک مرحله کلیدی در مقایسه الگوریتم‌ها، به دست آوردن معیارهای معتبر عملکرد برای مقایسه است. تا اینجا ما فرض کردیم که اینها داده شده است. یک روش برای به دست آوردن نمونه‌های بی طرفانه از خطا، نگه داشتن یک مجموعه تست نگهدارنده است. تصور کنید m نمونه در رزرو تنظیم شده است، که الگوریتم‌ها روی آن‌ها آموزش داده نشده‌اند و تا زمانی که آماده ارزیابی نباشیم نمی‌توانیم به آن‌ها نگاه کنیم. ما می‌توانیم دو مدل را در مجموعه آموزشی آموزش دهیم، و سپس m نمونه‌های جفتی از خطا را بدست آوریم. سپس می‌توانیم از آزمون t زوجی استفاده کنیم تا ادعا کنیم که آیا این دو مدل از نظر آماری به طور معنی‌داری برای مسئله متفاوت هستند یا خیر.

یک رویکرد جایگزین برای به دست آوردن تخمین‌های خطا، استفاده از تکنیک‌های نمونه گیری مجدد از کل مجموعه داده است. دو تکنیک نمونه‌گیری مجدد رایج عبارتند از اعتبارسنجی متقاطع k – fold و نمونه‌برداری مجدد $bootstrap$. در مرحله اول، داده‌ها به k مجموعه‌های ناهمگون (فولد) تقسیم می‌شوند. این مدل روی $k - 1$ چین آموزش داده می‌شود و روی چین دیگر آزمایش می‌شود. این k بار تکرار می‌شود که در آن هر فولد به عنوان تای تست عمل می‌کند. این رویکرد محیط یادگیری رایج را شبیه سازی می‌کند که در آن مجموعه‌های آموزشی و آزمون از هم جدا هستند. تخمین‌های عملکرد k به‌دست‌آمده عمدتاً مستقل هستند، با برخی وابستگی‌ها به دلیل وابستگی‌های بین مجموعه‌های آموزشی در سراسر k اجراها معرفی شده‌اند. برخی از تعصبات اضافی ارائه شده از این واقعیت است که ما مدل را در کل مجموعه آموزشی اجرا نمی‌کنیم، بلکه تخمینی از خطای الگوریتم آموزش داده شده بر روی $n - (n/k)$ دریافت می‌کنیم. برای هر مدل نهایی که پس از انجام این ارزیابی‌ها وارد تولید می‌شود، احتمالاً کل مجموعه n نمونه را آموزش خواهیم داد.

نمونه مجدد بوت استرپ با داده‌ها از ایده پشت بوت استرپ استفاده می‌کند: داده‌ها مدل معقولی از داده‌ها را تشکیل می‌دهند. با نمونه برداری از داده‌ها، مانند نمونه برداری از توزیعی است که داده‌ها را تولید کرده است. برای ایجاد تقسیم‌های آموزشی/آزمون، داده‌ها با جایگزینی برای ایجاد مجموعه آموزشی نمونه‌برداری می‌شوند و نمونه‌های استفاده نشده باقی‌مانده برای آزمایش استفاده می‌شوند. اگر k نمونه مجدد بدست آید، دوباره k معیارهای عملکرد را بدست می‌آوریم و می‌توانیم میانگین نمونه عملکرد را در تقسیمات مختلف بدست آوریم و از آزمون معناداری آماری استفاده کنیم.

برای درک بهتر خواص این دو رویکرد، به توضیح کامل و قابل دسترس در [۸، فصل ۵] مراجعه کنید.

۱۰-۴ معیارهای عملکرد برای مدل‌های طبقه بندی

در طبقه بندی، انواع معیارهای عملکردی وجود دارد که اهمیت نسبی پیش بینی‌های نادرست را برای هر یک از کلاس‌ها منعکس می‌کند. به عنوان مثال، پیش‌بینی اینکه بیمار واقعاً بیمار است (منفی کاذب) می‌تواند مضرتر باشد، که در نتیجه تصمیم به انجام ندادن تشخیص بیشتر می‌شود و در نتیجه باعث عوارض جدی ناشی از عدم درمان بیماری می‌شود. هنگام آموزش و ارزیابی الگوریتم‌های طبقه‌بندی، این اولویت‌ها باید کدگذاری شوند. جدول ۱۰،۳ برخی از اصطلاحات را برای بحث در مورد عملکرد مدل‌های طبقه بندی خلاصه می‌کند.

Name	Symbol	Definition
Classification error	$error$	$error = \frac{fp+fn}{tp+fp+tn+fn}$
Classification accuracy	$accuracy$	$accuracy = 1 - error$
True positive rate	tpr	$tpr = \frac{tp}{tp+fn}$
False negative rate	fnr	$fnr = \frac{fn}{tp+fn}$
True negative rate	tnr	$tnr = \frac{tn}{tn+fp}$
False positive rate	fpr	$fpr = \frac{fp}{tn+fp}$
Precision	pr	$pr = \frac{tp}{tp+fp}$
Recall	rc	$rc = \frac{tp}{tp+fn}$

جدول ۱۰/۳: برخی از معیارهای طبق بندی.

[1] P Auer, M Herbster, and Manfred K Warmuth. Exponentially many local minima for single neurons. In Advances in Neural Information Processing Systems, 1996.

[2] A Banerjee, S Merugu, I S Dhillon, and J Ghosh. Clustering with Bregman divergences. Journal of Machine Learning Research, 2005.

[3] Peter L Bartlett and Shahar Mendelson. Rademacher and Gaussian Complexities: Risk Bounds and Structural Results. Journal of Machine Learning Research, 2002.

[4] Amir Beck and Marc Teboulle. A Fast Iterative Shrinkage-Thresholding Algorithm for Linear Inverse Problems. SIAM J. Imaging Sciences, 2009.

[5] L on Bottou and Yann Le Cun. On-line learning for very large data sets. Applied Stochastic Models in Business and Industry, 2005.

[6] Léon Bottou. Online learning and stochastic approximations. Online Learning and Neural Networks, 1998.

[7] Olivier Bousquet, Stéphane Boucheron, and Gábor Lugosi. Introduction to Statistical Learning Theory. In Advanced Lectures on Machine Learning. Springer Berlin Heidelberg, 2004.

[8] Gareth James, Daniela Witten, Trevor Hastie, and Robert Tibshirani. An Introduction to Statistical Learning. Springer New York, 2013.

[9] Nathalie Japkowicz and Mohak Shah. Evaluating Learning Algorithms - A Classification Perspective. Cambridge University Press, 2011.

[10] Sham M Kakade, Karthik Sridharan, and Ambuj Tewari. On the Complexity of Linear Prediction: Risk Bounds, Margin Bounds, and Regularization. In Advances in Neural Information Processing Systems, 2008.

[11] Diederik P Kingma and Jimmy Ba. Adam: A Method for Stochastic Optimization. In International Conference on Learning Representations, 2015.

[12] Lei Le and Martha White. Global optimization of factor models using alternating minimization. arXiv.org, 2016.

[13] Julien Mairal, Francis Bach, Jean Ponce, Guillermo Sapiro, and Andrew Zisserman. Supervised dictionary learning. In Advances in Neural Information Processing Systems, 2009.

[14] Jorge Nocedal. Updating quasi-Newton matrices with limited storage. Mathematics of Computation, 1980.

[15] Bruno A Olshausen and David J Field. Sparse coding with an overcomplete basis set: a strategy employed by V1? Vision Research, 1997.

[16] K B Petersen. The matrix cookbook. Technical University of Denmark, 2004.

[17] Dinah Shender and John Lafferty. Computation-Risk Tradeoffs for Covariance-Thresholded Regression. International Conference on Machine Learning, 2013.

[18] Ajit P Singh and Geoffrey J Gordon. A unified view of matrix factorization models. In European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases, 2008.

[19] Larry Wasserman. All of Statistics: A Concise Course in Statistical Inference. Springer, 2004.

[20] Martha White. Regularized factor models. PhD thesis, University of Alberta, 2014.

[21] Matthew D Zeiler. ADADELTA: An Adaptive Learning Rate Method. arXiv.org, 2012.

پیوست A

مطالب اضافی برای نظریه احتمال

A.1 بدیهیات احتمال

ما می‌توانیم ویژگی‌های دیگری را از تعریف اولیه مجموعه رویدادهای قابل اندازه‌گیری (جبر سیگما) و توزیع‌های احتمال استخراج کنیم. تعریف میدان سیگما مستلزم آن است که $\mathcal{E}$ تحت هر دو تعداد محدود و قابل شمارش نامتناهی از عملیات مجموعه پایه (اتحاد، تقاطع، متمم و اختلاف مجموعه) بسته شود. اتحاد عملیات و تکمیل در تعریف است. برای تقاطع، می‌توانیم از قوانین دی مورگان[1] استفاده کنیم: $\cup A_i = (\cap A_i^c)^c$ و $\cap A_i = (\cup A_i^c)^c$. هر تقاطع مجموعه ها در $\mathcal{E}$ باید دوباره در $\mathcal{E}$ باشد زیرا $\mathcal{E}$ تحت اتحاد و مکمل بسته است. بنابراین، یک میدان سیگما نیز در زیر تقاطع بسته است. به طور مشابه برای اختلاف مجموعه، می‌توانیم $A_1 - A_2 = (A_1 \cap A_2)^c \cap A_1$ را بنویسیم، که سپس به $A\_1 - A\_2 \in \mathcal{E}$ اشاره می‌کند زیرا $A_1 \cap A_2 \in \mathcal{F} \Longrightarrow (A_1 \cap A_2)c \in \mathcal{E} \Longrightarrow (A_1 \cap A_2)c \cap A_1 \in \mathcal{F}$. از آنجایی که $\mathcal{E}$ خالی نیست، مشاهده می‌کنیم که همه شرایط فوق نشان می‌دهد که $\Omega \in \mathcal{E}$ و $\emptyset \in \mathcal{E}$، که در آن $\emptyset$ مجموعه خالی است.

برای توزیع احتمال $P : \mathcal{E} \to [0,1]$، ما نیاز داریم

$$1.\ \mathrm{P}(\Omega) = 1$$

$$2.\ \mathrm{A}_1, \mathrm{A}_2, \ldots \in \mathcal{E}, \mathrm{A_i} \cap \mathrm{A_j} = \emptyset \forall \mathrm{i,j} \Rightarrow \mathrm{P}\left(\bigcup_{\mathrm{i}=1}^{\infty} \mathrm{A_i}\right) = \sum_{\mathrm{i}=1}^{\infty} \mathrm{P}(\mathrm{A_i})$$

تاپل $(\Omega, \mathcal{E}, P)$ فضای احتمال نامیده می‌شود. به نظر می‌رسد بصری است که شرط دوم را می‌توان با اتحادیه‌ای از مجموعه های محدود (نیاز ساده‌تر افزایش به جای σ افزایشی) جایگزین کرد. با این حال، برای فیلدهای سیگما، بسته شدن تحت اتحادیه های محدود ممکن است منجر به بسته شدن تحت اتحادیه های نامحدود نشود.

زیبایی این بدیهیات در فشردگی و ظرافت آنها نهفته است. بسیاری از عبارات مفید را می‌توان از بدیهیات احتمال استخراج کرد. برای مثال، واضح است که $\mathrm{P}(\emptyset) = 0$ یا $\mathrm{P}(\mathrm{A_c}) = 1 - \mathrm{P}(\mathrm{A})$. به طور مشابه، بسته شدن تحت اتحادیه های نامتناهی مجموعه های غیرمتناسب (σ افزایشی) به بسته شدن محدود (افزودنی) دلالت دارد، زیرا مجموعه های باقیمانده را می‌توان به مجموعه تهی $\forall \mathrm{A_1, A_2} \in \mathcal{E}: \emptyset$ تنظیم کرد با: $A_1 \cap A_2 = \emptyset$ مجموعه در $\mathrm{A_i} = \emptyset$ برای بدست آوردن $\mathrm{i} > 2$. فرمول دیگری که اهمیت ویژه‌ای دارد را می‌توان با $\mathrm{P}(\mathrm{A_1} \cup \mathrm{A_2}) = \mathrm{P}(\mathrm{U}_{\mathrm{i}=1}^{\infty} \mathrm{A_i}) = \sum_{\mathrm{i}=1}^{\infty} \mathrm{P}(\mathrm{A_i}) = \mathrm{P}(\mathrm{A_1}) + \mathrm{P}(\mathrm{A_2})$

[1] De Morgan's laws

در نظر گرفتن پارتیشن[1] فضای نمونه بدست آورد. یعنی مجموعه‌ای از k مجموعه‌های غیر همپوشانی $\{Bi\}_{i=1}^{k}$ به طوری که $\Omega = \cup_{i=1}^{k} B_i$. یعنی اگر A هر مجموعه‌ای در Ω باشد و اگر $\{Bi\}_{i=1}^{k}$ پارتیشنی از Ω باشد نتیجه می‌شود که

$$\begin{aligned} \mathrm{P(A)} &= \mathrm{P(A \cap \Omega)} \\ &= P\left(A \cap \left(\bigcup_{i=1}^{k} B_i\right)\right) \\ &= P\left(\left(\bigcup_{i=1}^{k} A \cap B_i\right)\right) \qquad (A.1) \\ &= \sum_{i=1}^{k} p(A \cap \mathrm{B_i}) \end{aligned}$$

که در آن خط آخر از بدیهیات احتمال پیروی می‌کند. ما به این عبارت به عنوان قانون جمع اشاره خواهیم کرد. عبارت مهم دیگری که در اینجا بدون مشتق نشان داده شده است این است که $P(A \cup B) = P(A) + P(B) - P(A \cap B)$.

A.2 چند pmf مفید دیگر

ما چند نمونه دیگر از pmfها را در اینجا ارائه می‌کنیم، عمدتاً به این منظور که مثال‌های ملموس اضافی می‌توانند برای درک مفید باشند. ما بیشتر از این pmf ها در این یادداشت ها استفاده نخواهیم کرد.

توزیع دو جمله‌ای برای توصیف دنباله‌ای از n آزمایش برنولی مستقل و توزیع شده یکسان $(i.i.d.)$ استفاده می‌شود. در هر مقدار k در فضای نمونه، توزیع این احتمال را می‌دهد که موفقیت دقیقاً k بار از n آزمایش اتفاق افتاده است، که البته $0 \leq \mathrm{k} \leq \mathrm{n}$ به طور فرمول‌بندی، $\Omega = \{0, 1, \ldots, \mathrm{n}\}$ برای $\forall \mathrm{k} \in \Omega$ دو جمله‌ای pmf به این صورت تعریف می‌شود

$$p(k) = \binom{n}{k} \alpha^{\mathrm{k}} (1-\alpha)^{\mathrm{n-k}}$$

که در آن $\alpha \in (0,1)$، مانند قبل، پارامتری است که احتمال موفقیت در یک آزمایش را نشان می‌دهد. در اینجا، ضریب دو جمله ای

$$\binom{n}{k} = \frac{n!}{k!\,(n-k)!}$$

همه راه‌هایی را برمی‌شمارد که از طریق آن می‌توان k عنصر را از فهرستی از n عنصر انتخاب کرد (به عنوان مثال، ۳ روش مختلف وجود دارد که از طریق آنها می‌توان $k = 2$ عنصر را از یک گروه $n = 3$ عنصر انتخاب کرد). ما به توزیع دو جمله‌ای با پارامترهای n و α به عنوان $Binomial(n, \alpha)$ اشاره خواهیم کرد. آزمایشی که منجر به توزیع دو جمله‌ای می‌شود را می‌توان به موقعیتی با بیش از دو نتیجه ممکن تعمیم داد. این آزمایش منجر به یک تابع جرم احتمال چند بعدی (یک بعد در هر نتیجه ممکن) به نام توزیع چند جمله‌ای می‌شود.

[1] partition

توزیع هندسی همچنین برای مدل‌سازی دنباله‌ای از آزمایش‌های مستقل برنولی با احتمال موفقیت α استفاده می‌شود. در هر نقطه $k \in \Omega$، این احتمال را می‌دهد که اولین موفقیت دقیقاً در آزمایش k-ام اتفاق می‌افتد. در اینجا، $\Omega = \{1, 2, \dots\}$ و برای $\forall k \in \Omega$

$$\mathrm{p(k)} = (1-\alpha)^{\mathrm{k}-1}\alpha$$

که $\alpha \in (0, 1)$ یک پارامتر است. توزیع هندسی، $\mathrm{Geometric}(\alpha)$، بر روی یک فضای نمونه بی‌نهایت تعریف شده است. یعنی $\Omega = N$

برای توزیع فراهندسی، جمعیت محدودی از عناصر N از دو نوع (مثلاً موفقیت و شکست) را در نظر بگیرید که K از یک نوع (مثلاً موفقیت) هستند. این آزمایش شامل ترسیم n عنصر، بدون جایگزینی، از این جمعیت است، به طوری که عناصر باقی مانده در جمعیت از نظر انتخاب شدن در قرعه کشی بعدی، یکسان هستند. احتمال استخراج k موفقیت از n آزمایش را می‌توان به صورت توصیف کرد

$$p(k) = \frac{\binom{K}{\mathrm{k}} \cdot \binom{N-K}{n-k}}{\binom{N}{n}}$$

که در آن $0 \leq \mathrm{n} \leq \mathrm{N}$ و $k \leq n$. توزیع ابر هندسی ارتباط نزدیکی با توزیع دو جمله‌ای دارد که در آن عناصر با جایگزینی ترسیم می‌شوند ($\alpha = K/N$). در آنجا، احتمال ترسیم موفقیت در آزمایشات بعدی تغییر نمی‌کند. ما به توزیع فوق هندسی به عنوان $Hypergeometric\ (n, N, K)$ اشاره خواهیم کرد.

A.3 چند pdf مفید دیگر

همانطور که در بالا ذکر شد، ما چند فایل pdf دیگر را برای ارائه مثال‌های عینی ارائه می‌دهیم، اگرچه به صراحت از این pdf ها در یادداشت ها استفاده نخواهیم کرد.

توزیع log نرمال یک تغییر توزیع نرمال است. در اینجا، برای $\Omega = (0, \infty)$ چگالی log نرمال را می‌توان به این صورت بیان کرد

$$\mathrm{p}(\omega) = \frac{1}{\omega\ \sqrt{2\pi\sigma^2}}\, \mathrm{e}^{-\frac{1}{2\sigma^2}(\ln\omega-\mu)^2}$$

که در آن $\mu \in \mathbb{R}$ و $\sigma > 0$ پارامترها هستند. ما به این توزیع به عنوان $Lognormal(\mu, \sigma^2)$ یا $ln\ N(\mu, \sigma^2)$ اشاره خواهیم کرد.

توزیع $Gumbel$ متعلق به کلاس توزیع‌های ارزش شدید است. تابع چگالی احتمال آن بر روی $\Omega = \mathbb{R}$ به این صورت تعریف می‌شود

$$\mathrm{p}(\omega) = \frac{1}{\beta}\, \mathrm{e}^{-\frac{\omega-\alpha}{\beta}\mathrm{e}^{-\mathrm{e}^{\frac{-\omega-\alpha}{\beta}}}}$$

که $\alpha \in \mathbb{R}$ پارامتر مکان و $\beta > 0$ پارامتر مقیاس است. ما به این توزیع به عنوان $\mathrm{Gumbel}(\alpha, \beta)$ اشاره خواهیم کرد.

توزیع پارتو[1] برای مدل‌سازی رویدادهایی با موارد نادر مقادیر شدید مفید است. تابع چگالی احتمال آن بر روی $\Omega = [\omega_{\min}, \infty)$ به صورت تعریف شده است

$$p(\omega) = \frac{\alpha \omega_{\min}}{\omega^{\alpha+1}}$$

که $\alpha > 0$ یک پارامتر و $\omega_{min} > 0$ مینیمم مقدار مجاز برای ω است. ما به توزیع پارتو به عنوان $\text{Pareto}(\alpha, \omega_{min})$ اشاره خواهیم کرد. هنگامی که $\alpha \in (0, 2]$ به یک ویژگی بدون مقیاس منجر می‌شود.

A.4 متغیرهای تصادفی

در بسیاری از موقعیت‌ها، ما می‌خواهیم از مدل‌سازی احتمالی در مجموعه‌ها (به عنوان مثال، گروهی از افراد) استفاده کنیم که در آن عناصر می‌توانند با توصیفگرهای مختلف مرتبط شوند. به عنوان مثال، یک فرد ممکن است با سن، قد، شهروندی، ضریب هوشی یا وضعیت تأهل او مرتبط باشد و ممکن است ما به رویدادهای مرتبط با این توصیف کننده ها علاقه‌مند باشیم. در موقعیت‌های دیگر، ممکن است به تبدیل فضاهای نمونه مانند فضاهایی که مربوط به دیجیتالی کردن سیگنال آنالوگ از یک میکروفون به مجموعه‌ای از اعداد صحیح بر اساس مجموعه‌ای از آستانه‌های ولتاژ هستند، علاقه‌مند باشیم. مکانیسم یک متغیر تصادفی پرداختن به همه چنین موقعیت‌هایی را به روشی ساده، دقیق و یکپارچه تسهیل می‌کند.

متغیر تصادفی متغیری است که از دیدگاه ناظر، مقادیر را به صورت غیر قطعی، با ترجیحات کلی متفاوت برای نتایج متفاوت می‌گیرد. اما از نظر ریاضی، تابعی است که یک فضای نمونه را به فضای دیگر نگاشت می‌کند، با چند اخطار فنی که بعداً معرفی خواهیم کرد. اجازه دهید نیاز به متغیرهای تصادفی را تحریک کنیم. یک فضای احتمال (Ω, E, P) را در نظر بگیرید، که در آن Ω مجموعه‌ای از افراد است و اجازه دهید احتمال خوشحالی یک فرد تصادفی انتخاب شده $\omega \in \Omega$ را بررسی کنیم (ممکن است فرض کنیم یک روش تشخیصی برای ارزیابی وضعیت هر فرد داریم) . ما با تعریف یک رویداد A شروع می‌کنیم

$$A = \{\omega \in \Omega : \text{Status}(\omega) = \text{happy}\}$$

و به سادگی احتمال این رویداد را محاسبه کنید. این یک رویکرد کاملاً قانونی است، اما می‌توان آن را با استفاده از مکانیسم متغیر تصادفی بسیار ساده کرد. ابتدا توجه می‌کنیم که از نظر فنی، روش تشخیصی ما با یک تابع مطابقت دارد: $S \to \Omega$ که فضای نمونه Ω را به یک فضای نمونه باینری جدید $\mathcal{S} = \{\text{happy}, \text{not happy}\}$ نگاشت می‌کند.جالبتر اینکه رویکرد ما همچنین توزیع احتمال P را به یک توزیع احتمال P_{status} جدید که بر روی برخی از جبر سیگما S تعریف شده است، ترسیم می‌کند. مثلاً ES_{tatus} (برای اینکه نگاشت مطابق انتظار عمل کند، باید مجموعه توان S باشد). اکنون می‌توانیم ببینیم که می‌توانیم وضعیت $P(\{\text{happy}\})$ را از روی احتمال رویداد A محاسبه کنیم. به عنوان مثال، $P_{\text{Status}}(\text{happy}) = P(A)$. این یک نماد به هم ریخته است، بنابراین ممکن است بخواهیم آن را با استفاده از $P(\text{Status} = \text{happy})$، که در آن وضعیت یک "متغیر تصادفی" است، ساده کنیم.

از حروف بزرگ $X, Y, \ldots$ برای نشان دادن متغیرهای تصادفی (مانند وضعیت) و حروف کوچک $x, y, \ldots$ برای نشان دادن عناصر (مانند "happy") استفاده خواهیم کرد فضاهای جدید $X, Y \ldots$ به طور کلی، احتمالات را به صورت $P(X = x)$ می‌نویسیم، که یک ریلکسیشن[2] نمادین از $P(\{\omega : X(\omega) = x\})$ یا $P(X \leq x)$ برای $P(\{\omega : X(\omega) \leq x\})$ است. زمانی

[1] Pareto distribution
[2] relaxation

که هم دامنه X پیوسته باشد. همچنین زمانی که نیاز به توضیح بیشتر در مورد متغیر تصادفی داشته باشیم، به توابع جرم یا چگالی احتمال مربوطه به صورت $p(x)$ یا $pX(x)$ اشاره خواهیم کرد. این در واقع زمانی اتفاق می‌افتد که x یک مقدار خاص را بگیرد. مثلاً برای $pX(1)، x = 1$ را خواهیم نوشت. قبل از اینکه به تعریف رسمی متغیرهای تصادفی بپردازیم، به دو مثال گویا نگاه خواهیم کرد.

مثال ۲۰: [پرتاب‌های متوالی یک سکه منصفانه.] فرآیندی از سه پرتاب سکه و دو متغیر تصادفی X و Y را در نظر بگیرید که در فضای نمونه تعریف شده اند. X را به عنوان تعداد سرها در اولین پرتاب و Y را به عنوان تعداد سرهای روی هر سه پرتاب تعریف می‌کنیم. هدف ما یافتن فضاهای احتمالی است که پس از تبدیل ها ایجاد می‌شوند.

ابتدا $\Omega = \{HHH, HHT, HTH, HTT, THH, THT, TTH, TTT\}$ و

ω	HHH	HHT	HTH	HTT	THH	THT	TTH	TTT
$X(\omega)$	1	1	1	1	0	0	0	0
$Y(\omega)$	3	2	2	1	2	1	1	0

بگذارید فقط روی متغیر Y تمرکز کنیم. واضح است که $Y : \Omega \to \{0, 1, 2, 3\}$ اما ما همچنین باید $\mathcal{E}_Y$ و P_Y را پیدا کنیم. برای محاسبه P_Y، یک روش ساده یافتن $pmf\ p(y)$ آن است. برای مثال، اجازه دهید $p_Y(2) = P_Y(\{2\})$را به صورت محاسبه کنیم

$$P_Y(\{2\}) = P(Y = 2)$$
$$= P(\{\omega : Y(\omega) = 2\})$$
$$= P(\{HHT, HTH, THH\})$$
$$= \frac{3}{8}$$

به دلیل توزیع یکنواخت در فضای اصلی $(\Omega، \mathcal{E}، P)$. به روشی مشابه، می‌توانیم محاسبه کنیم که $P(Y = 0) = P(Y = 3) = 1/8$ و $P(Y = 1) = 3/8$. در این مثال، ما در نظر گرفتیم که $\mathcal{E} = P(\Omega)$ و $\mathcal{E}_Y = P(Y)$. به عنوان نکته پایانی، اشاره می‌کنیم که تمام تصادفی بودن در فضای احتمال اصلی $(\Omega، \mathcal{E}، P)$ تعریف می‌شود و فضای احتمال جدید $(Y، \mathcal{E}_Y، P_Y)$ به سادگی آن را از طریق یک تبدیل قطعی به ارث می‌برد.

مثال ۲۱: [کوانتیزاسیون] $(\Omega، \mathcal{E}، P)$ را در نظر بگیرید که در آن $\Omega = [0, 1]$، $\mathcal{E} = B(\Omega)$و P توسط یک pdf یکنواخت القا می‌شود. X را تعریف کنید: $X : \Omega \to \{0, 1\}$ به عنوان

$$X(\omega) = \begin{cases} 0 & \omega \leq 0.5 \\ 1 & \omega > 0.5 \end{cases}$$

و فضای احتمال تبدیل شده را پیدا کنید.

از نظر فنی، فضای نمونه را به $X = \{0, 1\}$ تغییر داده‌ایم. برای فضای رویداد $\mathcal{E}_X = P(\mathcal{X}) = \emptyset, 0, 1, 0, 1$ ما می‌خواهیم توزیع احتمال جدید P_X را درک کنیم. ما داریم

$$p_X(0) = PX(\{0\})$$
$$= P(X = 0)$$

$$= \mathrm{P}(\{\omega: \omega \in [0,0.5]\})$$

$$= \frac{1}{2}$$

و

$$\mathrm{p_X}(1) = \mathrm{P_X}(\{1\})$$

$$= \mathrm{P}(\mathrm{X} = 1)$$

$$= \mathrm{P}(\{\omega : \omega \in (0.5, 1]\})$$

$$= \frac{1}{2}$$

از اینجا به راحتی می‌توانیم ببینیم که $\mathrm{PX}(\{0,1\}) = 1$ و $\mathrm{PX}(\emptyset) = 0$، و بنابراین $\mathrm{P_X}$ در واقع یک توزیع احتمال است. دوباره، P_X به طور طبیعی با استفاده از P تعریف می‌شود. بنابراین، فضای احتمال $(\Omega، \mathcal{E}، P)$ را به $(X، \mathcal{E}_X، P_X)$ تبدیل کرده‌ایم.

A.4.1 تعریف رسمی متغیر تصادفی

اکنون به طور رسمی یک متغیر تصادفی تعریف می‌کنیم. با در نظر گرفتن یک فضای احتمال $(\Omega, \mathcal{E}, P)$، یک متغیر تصادفی X تابع $X : \Omega \to X$ است به طوری که برای هر $A \in B(\mathcal{X})$ این گونه است که $\{\omega : X(\omega) \in A\} \in \mathcal{E}$ است. نتیجه می‌شود که

$$\mathrm{P_X}(\mathrm{A}) = \mathrm{P}(\{\omega: \mathbf{X}(\omega) \in \mathrm{A}\})$$

ذکر این نکته ضروری است که به‌طور پیش‌فرض، فضای رویداد یک متغیر تصادفی را فیلد $Borel\ X$ تعریف کرده‌ایم. این راحت است زیرا یک میدان بورل از یک مجموعه قابل شمارش Ω مجموعه توان آن است. بنابراین، ما در حال کار با بزرگترین فضاهای رویداد ممکن برای متغیرهای تصادفی گسسته و پیوسته هستیم.

اکنون یک متغیر تصادفی گسسته X را در نظر بگیرید که روی $(\Omega، \mathcal{E}، P)$ تعریف شده است. همانطور که از مثال‌های قبلی می‌بینیم، توزیع احتمال $\boldsymbol{X}$ را می‌توان به صورت پیدا کرد

$$\mathrm{p}(\mathrm{x}) = \mathrm{P_{X(x)}}$$

$$= \mathrm{P}(\{\omega: \mathrm{X}(\omega) = \mathrm{x}\})$$

برای $\forall \mathrm{x} \in \mathcal{X}$. احتمال یک رویداد A را می‌توان به صورت پیدا کرد

$$\mathrm{P_X}(\mathrm{A}) = \mathrm{P}(\{\omega: \mathrm{X}(\omega) \in \mathrm{A}\})$$

$$= \sum_{x \in A} p(x)$$

برای $\forall \mathrm{A} \subseteq \mathcal{X}$

مورد متغیرهای تصادفی پیوسته پیچیده‌تر است، اما به رویکردی شبیه به متغیرهای تصادفی گسسته کاهش می‌یابد. در اینجا ابتدا یک تابع توزیع تجمعی (cdf) را به صورت تعریف می‌کنیم

$$F_X(t) = P_X(\{x: x \le t\})$$
$$= P(\{\omega : X(\omega) \le t\})$$
$$= P(X \le t)$$

که در آن $P(X \le t)$، مانند قبل، سوء استفاده جزئی از نماد را نشان می‌دهد. اگر تابع توزیع تجمعی قابل تمایز باشد، تابع چگالی احتمال یک متغیر تصادفی پیوسته به صورت تعریف می‌شود.

$$p(x) = \left.\frac{dF_X(t)}{d_t}\right|_{t=x}$$

متناوبا، اگر $p(x)$ وجود داشته باشد، آنگاه

$$F_X(t) = \int_{-\infty}^{t} p(x)dx$$

برای هر $t \in \mathbb{R}$. تمرکز ما منحصراً بر روی متغیرهای تصادفی است که توابع چگالی احتمال خود را دارند. با این حال، برای یک دید کلی تر، ما همیشه باید "اگر وجود دارد" را در هنگام مراجعه به فایل‌های pdf در نظر داشته باشیم.

احتمال اینکه یک متغیر تصادفی مقداری از بازه (a, b) بگیرد اکنون می‌تواند به صورت محاسبه شود.

$$PX((a,b]) = P(a < X \le b)$$
$$= \int_a^b p(x)\,dx$$
$$= F_X(b) - F_X(a)$$

که از ویژگی‌های یکپارچه سازی به دست می‌آید.

حالا فرض کنید که متغیر تصادفی X فضای احتمال $(\Omega, \mathcal{E}, P)$ را به $(X, B(\mathcal{X}), P_X)$ تبدیل می‌کند. برای توصیف فضای احتمال حاصل، معمولاً از توابع جرم و چگالی احتمال القاکننده P_X استفاده می‌کنیم. به عنوان مثال، اگر P_X توسط توزیع گاوسی با پارامترهای μ و σ^2 القا شود، از

$$X: \mathcal{N}(\mu, \sigma^2) \qquad \text{or} \qquad X \sim \mathcal{N}(\mu, \sigma^2)$$

هر دو نماد نشان می‌دهند که تابع چگالی احتمال برای متغیر تصادفی X است

$$p(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{1}{2\sigma^2}(x-\mu)^2}$$

چگالی گاوسی به طور ضمنی تعریف می‌کند که $X = R$. اما این نقطه سطحی است زیرا ما همیشه می‌توانیم دامنه یک تابع چگالی را به $\mathbb{R}$ گسترش دهیم و هر جا که تابع اصلی تعریف نشده است، $p(x) = 0$ را تنظیم کنیم.

گروهی از d متغیرهای تصادفی $\{X_i\}_{i=1}^{d}$ که در فضای احتمال یکسان $(\Omega, \mathcal{E}, P)$ تعریف شده‌اند، بردار تصادفی یا متغیر تصادفی چند متغیره (چند بعدی) نامیده می‌شوند. ما قبلاً نمونه‌ای از یک بردار تصادفی ارائه شده توسط متغیرهای تصادفی (X, Y) را در مثال ۲۰ دیده ایم. یعنی $\{X_i : i \in \mathcal{T}\}$، که در آن $\mathcal{T}$ یک مجموعه شاخص است که معمولاً به عنوان مجموعه‌ای از شاخص‌های زمانی تفسیر می‌شود. در مورد شاخص‌های زمان گسسته (به عنوان مثال، $\mathcal{T} = N$) فرآیند تصادفی یک فرآیند

تصادفی زمان گسسته نامیده می‌شود. در غیر این صورت (به عنوان مثال، $\mathcal{T} = \mathbb{R}$) به آن یک فرآیند تصادفی زمان پیوسته می‌گویند. مدل‌های زیادی در یادگیری ماشینی وجود دارد که با متغیرهای تصادفی مرتبط زمانی سروکار دارند (مانند مدل‌های خودبازگشتی برای سری‌های زمانی، زنجیره‌های مارکوف، مدل‌های مارکوف پنهان، شبکه‌های بیزی پویا). زبان متغیرهای تصادفی، از طریق فرآیندهای تصادفی، به خوبی امکان فرمول‌بندی این مدل‌ها را فراهم می‌کند. با این حال، بیشتر این یادداشت‌ها با تنظیمات ساده‌تر که فقط به متغیرهای تصادفی چند متغیره $(i.i.d.)$ نیاز دارند، سروکار دارند.

مثال ۲۲: [سه پرتاب یک سکه منصفانه برای احتمالات مشترک.] دو متغیر تصادفی از مثال ۲۰ را در نظر بگیرید و فضاهای احتمال، توزیع مشترک و حاشیه آنها را محاسبه کنید. $Recall\ X$ تعداد سرها در اولین پرتاب و Y تعداد سرهای روی هر سه پرتاب است.

تابع جرم احتمال مشترک $p(x,y) = P(X = x, Y = y)$ در زیر نشان داده شده است

X \ Y	0	1	2	3
0	$1/8$	$1/4$	$1/8$	0
1	0	$1/8$	$1/4$	$1/8$

اما اجازه دهید لحظه‌ای به عقب برگردیم و نشان دهیم که چگونه می‌توانیم آن را محاسبه کنیم. اجازه دهید دو مجموعه $A = \{HHH, HHT, HTH, HTT\}$ و $B = \{HHT, HTH, THH\}$ را در نظر بگیریم که به ترتیب مربوط به رویدادهایی است که اولین پرتاب سر بود و دقیقاً دو سر روی سه پرتاب وجود داشت. . حال، اجازه دهید به احتمال تقاطع A و B نگاه کنیم

$$P(A \cap B) = P(\{HHT, HTH\})$$
$$= \frac{1}{4}$$

می‌توانیم احتمال عبارت منطقی $X = 1 \wedge Y = 2$ را به عنوان نمایش دهیم

$$p_{XY}(1,2) = P(X = 1, Y = 2)$$
$$= P(A \cap B)$$
$$= P(HHT, HTH)$$
$$= \frac{1}{4}$$

توزیع احتمال حاشیه‌ای را می‌توان به روشی ساده پیدا کرد

$$p(x) = \sum_{y \in \mathcal{Y}} p(x,y)$$

که در آن $\mathcal{Y} = \{0,1,2,3\}$. بدین ترتیب

$$p_X(0) = \sum_{y \in \mathcal{Y}} p_{XY}(0, y)$$

برای پایان خاطرنشان می‌کنیم که در حالت $|\mathcal{X}| \cdot |\mathcal{Y}| - 1$ داریم پارامتر آزاد (زیرا مجموع باید برابر ۱ باشد) تا توزیع مشترک $p(x, y)$ را به طور کامل توصیف کند. به طور مجانبی، این مربوط به رشد تصاعدی تعداد ورودی‌های جدول با تعداد متغیرهای تصادفی (d) است. برای مثال، اگر $|X_i| = 2$ برای $\forall X_i$، ۲ $d - 1$ عنصر آزاد در توزیع احتمال مشترک وجود دارد. برآورد چنین توزیع‌هایی از داده‌ها غیرقابل حل است و یکی از اشکال لعنت ابعاد ابعادی است.

A.4.2 مثالی از استقلال مشروط

در دو مثال ساده از شکل A.1 نشان می‌دهیم که استقلال و استقلال مشروط بر یکدیگر دلالت ندارند. این مثال بیشتر جبری است و تمرین مفیدی برای نشان دادن این ویژگی است، اما شهود زیادی در مورد اینکه چرا این اتفاق می‌افتد ارائه نمی‌دهد. قوانین جداسازی d ارائه شده در بخش A.6 بیشتر توضیح می‌دهد که چرا می‌توانید این وابستگی‌های مختلف را داشته باشید. ما قبلاً در مثال ۷ مثالی از X و Y ارائه کرده‌ایم که به طور مشروط مستقل هستند، اما مستقل نیستند، در مثال ۷. ما یک مثال دیگر را در اینجا برای زمانی که دو متغیر X و Y مستقل هستند، اما با توجه به Z مستقل مشروط نیستند، ارائه می‌دهیم. این است که اطلاعات در Z زوج X و Y است، در حالی که در مثال ۷، دانستن Z (بایاس سکه) X و Y را جدا می‌کند (دو تلنگر از سکه بایاس)

مثال ۲۳: تنظیماتی را در نظر بگیرید که در آن سعی می‌کنید قیمت یک خانه را پیش بینی کنید. شما نمونه‌های زیادی از قیمت خانه‌های قبلی دارید، اما بدون هیچ ویژگی مرتبط. بگذارید X و Y قیمت دو خانه متفاوت باشد. بدون هیچ گونه اطلاعات اضافی، این دو متغیر مستقل هستند - در واقع، می‌توانیم آنها را به عنوان $i.i.d.$ در نظر بگیریم. نمونه‌هایی از برخی توزیع‌های اساسی بر روی قیمت مسکن. با این حال، اگر اکنون اطلاعات اضافی به ما داده شود که هر دو خانه مشترک هستند، آن‌ها وابسته می‌شوند. بگذارید Z با متغیری مطابقت داشته باشد که دو خانه در یک همسایگی قرار دارند (یعنی یک متغیر 0 یا 1). اگر $Z = 1$ باشد، آنگاه دانستن قیمت X قطعاً بر میزان توزیع بر قیمت ها برای Y تأثیر می‌گذارد. اضافه شدن این ویژگی این دو متغیر تصادفی را به صورت شرطی وابسته می‌کند.

A.4.3 اطلاعات اضافی برای انتظارات و لحظات

در این بخش، چند مثال اضافی از انتظارات توابع یک متغیر تصادفی ارائه می‌کنیم که اغلب در نظر گرفته می‌شوند - به اندازه‌ای که نام‌هایی برای آنها داده شود. به یاد بیاورید که برای یک تابع $f : \mathcal{X} \to \mathbb{R}$، مقدار مورد انتظار $\mathbb{E}[f(X)] = \sum f(x)p(x)$ را داریم. با استفاده از $f(x) = x^k$ در لحظه k-ام، $f(x) = log\ 1/p(x)$ تابع آنتروپی شناخته شده $H(X)$ یا آنتروپی دیفرانسیل برای متغیرهای تصادفی پیوسته و $f(x) = (x - \mathbb{E}[X])^2$ واریانس یک متغیر تصادفی X را که با $V[X]$ نشان داده می‌شود، ارائه می‌دهد. جالب توجه است که احتمال وقوع یک رویداد $A \subseteq \mathcal{X}$ نیز می‌تواند به شکل انتظار بیان شود. یعنی

$$P(A) = \mathbb{E}[1(\mathbf{X} \in A)]$$

A. Y و X مستقل هستند، اما با توجه به Z مستقل مشروط نیستند

$P(X=1)$
a

$$P(Y=y|X=x) = P(Y=y)$$

for example,

$$P(Y=1|X=x) = b$$
$$P(Y=1) = b$$

X	$P(Y=1\|X)$
0	b
1	b

X	Y	$P(Z=1\|X,Y)$
0	0	c
0	1	$1-c$
1	0	$1-c$
1	1	c

$$P(Y=y|X=x, Z=z) \neq P(Y=y|Z=z)$$

for example,

$$P(Y=1|X=1, Z=1) = bc/(1-c-b(1-2c))$$
$$P(Y=1|Z=1) = b(1-c-a(1-2c))/d$$
$$\text{where } d = P(Z=1)$$

B. X و Z با توجه به Y به صورت مشروط مستقل هستند، اما مستقل نیستند

$P(X=1)$
a

$$P(Z=z|X=x) \neq P(Z=z)$$

for example,

$$P(Z=1|X=1) = d + ce - cd$$
$$P(Z=1) = d + (e-d)(a(c-b)+b)$$

X	$P(Y=1\|X)$
0	b
1	c

X	Y	$P(Z=1\|X,Y)$
0	0	d
0	1	e
1	0	d
1	1	e

$$P(Z=z|X=x, Y=y) = P(Z=z|Y=y)$$

for example,

$$P(Z=1|X=x, Y=1) = e$$
$$P(Z=1|Y=1) = e$$

شکل A.1: استقلال در مقابل استقلال شرطی با استفاده از توزیع احتمال شامل سه متغیر تصادفی باینری. توزیع‌های احتمال با استفاده از فاکتورسازی $p(x,y,z) = p(x)p(y|x)p(z|x,y)$ ارائه می‌شوند که در آن همه ثابت‌های $a,b,c,d,e \in [0,1]$. (الف) متغیرهای X و Y مستقل هستند، اما با توجه به Z به صورت شرطی مستقل نیستند. هنگامی که $c = 0$، $Z = X \oplus Y$، که در آن $\oplus$ یک عملگر "انحصاری یا" است. (ب) متغیرهای X و Z با توجه به Y به صورت شرطی مستقل هستند، اما مستقل نیستند.

$f(x)$	Symbol	Name
x	$\mathbb{E}[X]$	Mean
$(x - \mathbb{E}[X])^2$	$V[X]$	Variance
x^k	$\mathbb{E}[X^k]$	k-th moment; $k \in \mathbb{N}$
$(x - \mathbb{E}[X])^k$	$\mathbb{E}[(X - \mathbb{E}[X])^k]$	k-th central moment; $k \in \mathbb{N}$
e^{tx}	$M_X(t)$	Moment generating function
e^{itx}	$\varphi_X(t)$	Characteristic function
$\log \frac{1}{p(x)}$	$H(X)$	(Differential) entropy
$\log \frac{p(x)}{q(x)}$	$D(p\|\|q)$	Kullback-Leibler divergence
$\left(\frac{\partial}{\partial\theta} \log p(x\|\theta)\right)^2$	$\mathcal{I}(\theta)$	Fisher information

جدول A.1: برخی از توابع مهم انتظار $\mathbb{E}[f(X)]$ برای متغیر تصادفی X که با توزیع آن $p(x)$ توصیف شده است. تابع $q(x)$ در تعریف واگرایی $Kullback - Leibler$ غیر منفی است و باید مجموع (ادغام) 1 شود. یعنی خود یک توزیع احتمال است. اطلاعات فیشر برای خانواده‌ای از توزیع‌های احتمالی تعریف شده توسط پارامتر θ تعریف شده است. توجه داشته باشید که تابع مولد لحظه ممکن است برای برخی از توزیع ها و همه مقادیر t وجود نداشته باشد. با این حال، تابع مشخصه همیشه وجود دارد، حتی زمانی که تابع چگالی وجود ندارد.

که

$$1(t) = \begin{cases} 1 \text{ t is true} \\ 0 \text{ t is false} \end{cases}$$

یک تابع نشانگر است. با این، می‌توان تابع توزیع تجمعی را به صورت $F\_X(t) = \mathbb{E}[1(\mathbf{X} \in (-\infty, t])]$ بیان کرد.

تابع $f(x)$ درون انتظار نیز می‌تواند با ارزش مختلط باشد. برای مثال، $\phi_X(t) = \mathbb{E}[e^{itX}]$، جایی که i واحد خیالی است، تابع مشخصه X را تعریف می‌کند. استنتاج آماری چندین تابع انتظار در جدول A.1 خلاصه شده است.

مثال ۲۴: [سه پرتاب یک سکه منصفانه (دوباره).] دو متغیر تصادفی از مثال‌های ۳ و ۵ را در نظر بگیرید و انتظار و واریانس را برای X و Y محاسبه کنید. سپس $\mathbb{E}[Y|X = 0]$ را محاسبه کنید. ما با محاسبه $\mathbb{E}[X] = 0 \cdot p_X(0) + 1 \cdot p_X(1) = \frac{1}{2}$ شروع می‌کنیم. به همین ترتیب

$$\mathbb{E}[Y] = \sum_{y=0}^{3} y \cdot p_Y(y)$$

$$= p_Y(1) + 2p_Y(2) + 3p_Y(3)$$

$$= \frac{3}{2}$$

$f(x,y)$	Symbol	Name
$(x-\mathbb{E}[X])(y-\mathbb{E}[Y])$	$\mathrm{Cov}[X,Y]$	Covariance
$\frac{(x-\mathbb{E}[X])(y-\mathbb{E}[Y])}{\sqrt{V[X]V[Y]}}$	$\mathrm{Corr}[X,Y]$	Correlation
$\log\frac{p(x,y)}{p(x)p(y)}$	$I(X;Y)$	Mutual information
$\log\frac{1}{p(x,y)}$	$H(X,Y)$	Joint entropy
$\log\frac{1}{p(x\|y)}$	$H(X\|Y)$	Conditional entropy

جدول A.2: برخی از توابع مهم انتظار $\mathbb{E}[f(X,Y)]$ برای دو متغیر تصادفی X و Y که با توزیع مشترک آنها $p(x,y)$ توضیح داده شده است. اطلاعات متقابل گاهی اوقات به عنوان اطلاعات متقابل متوسط نامیده می‌شود.

انتظار مشروط را می‌توان به صورت پیدا کرد

$$\mathbb{E}[\mathrm{Y}|\mathrm{X}=0]=\sum_{\mathrm{y}=0}^{3}\mathrm{y}\cdot\mathrm{pY}|\mathrm{X}(\mathrm{y}|0)$$

$$=\mathrm{p_{Y|X}}(1|0)+2p_{Y|X}(2|0)+3p_{Y|X}(3|0)$$

$$=1$$

که در آن $\mathrm{p}(\mathrm{y}|\mathrm{x})=\mathrm{p}(\mathrm{x},\mathrm{y})/\mathrm{p}(\mathrm{x})$.

A.5 مخلوط‌های توزیع

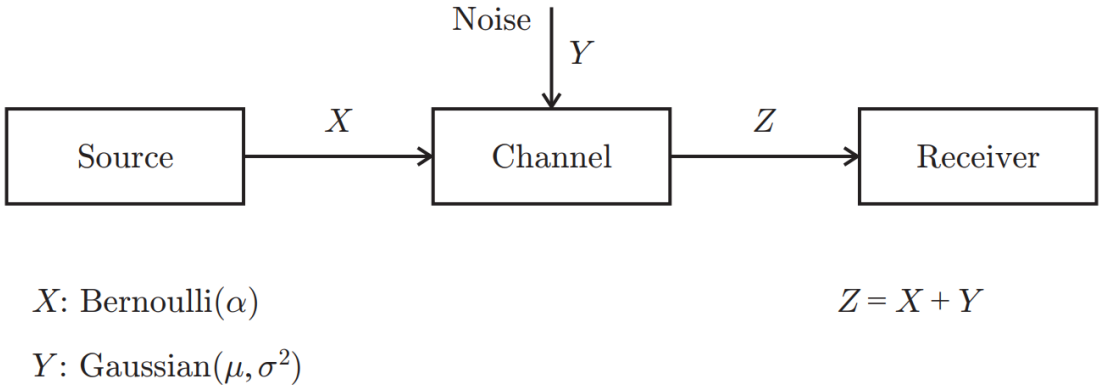
در بخش‌های قبلی دیدیم که متغیرهای تصادفی اغلب با استفاده از خانواده‌های خاصی از توزیع‌های احتمال توصیف می‌شوند. این رویکرد را می‌توان با در نظر گرفتن مخلوطی از توزیع ها تعمیم داد. به عنوان مثال، ترکیبات خطی سایر توزیع‌های احتمال. مانند قبل، ما فقط متغیرهای تصادفی را در نظر خواهیم گرفت که توابع جرم یا چگالی احتمالی خود را دارند

با توجه به مجموعه‌ای از توزیع‌های احتمال m، $\{p_i(x)\}_{i=1}^{m}$، یک تابع توزیع مخلوط محدود، یا مدل مخلوط، $p(x)$ به صورت تعریف می‌شود.

$$\mathrm{p}(\mathrm{x})=\sum_{\mathrm{i}=1}^{\mathrm{m}}w_{\mathrm{i}}\mathrm{p}_{\mathrm{i}}(\mathrm{x})$$

که در آن $\boldsymbol{w}=(w_1,w_2,\ldots,w_m)$ مجموعه‌ای از اعداد حقیقی غیر منفی است به طوری که $\sum_{i=1}^{m}w_i=1$. ما به w به عنوان ضرایب اختلاط یا گاهی اوقات به عنوان احتمالات اختلاط اشاره می‌کنیم. ترکیب خطی با چنین ضرایبی را ترکیب محدب می‌نامند. راستی آزمایی اینکه تابعی که به این روش تعریف شده است، در واقع یک توزیع احتمال است، ساده است.

در اینجا به طور مختصر به توابع اولیه انتظار از توزیع مخلوط می‌پردازیم. حالت فرضی $\{X_i\}_{i=1}^m$ مجموعه‌ای از m متغیرهای تصادفی است که با توزیع احتمال مربوطه آن‌ها توصیف می‌شوند $\{p_{X_i}(x)\}_{i=1}^m$. همچنین فرض کنید که یک متغیر تصادفی X با توزیع مخلوط با هم کارآمدی $\boldsymbol{w}$ و توزیع احتمال $\{p_{X_i}(x)\}_{i=1}^m$ توصیف شود. سپس با فرض متغیرهای تصادفی پیوسته



شکل A.2: یک سیستم ارتباطی سیگنال دیجیتال با نویز افزایشی.

در $\mathbb{R}$ تعریف شده است، تابع انتظار به صورت داده شده است

$$\mathbb{E}[f(X)] = \int_{-\infty}^{+\infty} f(x)p_X(x)dx$$

$$= \int_{-\infty}^{+\infty} f(x) \sum_{i=1}^{m} w_i pX_i(x)dx$$

$$= \sum_{i=1}^{m} w_i \int_{-\infty}^{+\infty} f(x)p_{X_i}(x)dx$$

$$= \sum_{i=1}^{m} w_i \mathbb{E}[f(X_i)]$$

اکنون می‌توانیم این فرمول را برای به دست آوردن میانگین، زمانی که $f(x) = x$ و واریانس، زمانی که $f(x) = (x - E[X])^2$ به‌دست آوریم، متغیر تصادفی X به عنوان

$$\mathbb{E}[X] = \sum_{i=1}^{m} w_i w[X_i]$$

و به این ترتیب

$$V[X] = \sum_{i=1}^{m} w_i V[X_i] + \sum_{i=1}^{m} w_i (\mathbb{E}[X_i] - \mathbb{E}[X])^2$$

مثال ۲۵: ارتباطات سیگنالی انتقال یک سیگنال دیجیتال باینری منفرد (بیت) را از طریق یک کانال ارتباطی نویز نشان داده شده در شکل A.2 در نظر بگیرید. بزرگی سیگنال X منتشر شده از منبع به همان اندازه 0 یا 1 ولت است. سیگنال از طریق

یک خط انتقال ارسال می‌شود (به عنوان مثال، ارتباطات رادیویی، فیبر نوری، نوار مغناطیسی) که در آن یک مولفه نویز معمولی توزیع شده صفر به X اضافه می‌شود. توزیع احتمال سیگنال $Z = X + Y$ را که وارد می‌شود استخراج کنید. گیرنده.

ما وضعیت کمی کلی‌تر را در نظر خواهیم گرفت که در آن: $X : \text{Bernoulli}(\alpha)$و Y گاوسی (μ, σ^2). برای یافتن $p(z)$ از توابع مشخصه متغیرهای تصادفی X، Y و Z استفاده خواهیم کرد که به صورت $\phi_X(t) = \mathbb{E}[e^{itX}]$، $\phi_Y(t) = \mathbb{E}[e^{itY}]$و $\phi_Z(t) = \mathbb{E}[e^{itZ}]$ نوشته می‌شود.. بدون اشتقاق می‌نویسیم

$$\phi_X(t) = 1 - \alpha + \alpha e^{it}$$

$$\phi_Y(t) = e^{it\mu - \frac{\sigma^2 t}{2}}$$

و پس از آن

$$\phi_Z(t) = \phi_{X+Y}(t)$$

$$= \phi_X(t) \cdot \phi_Y(t)$$

$$= (1 - \alpha + \alpha e^{it\mu - \frac{\sigma^2 t^2}{2}}$$

$$= \alpha e^{it(\mu+1) - \frac{\sigma^2 t^2}{2}} + (1 - \alpha) e^{it\mu - \frac{\sigma^2 t^2}{2}}$$

با انجام یکپارچه سازی روی $\phi_Z(t)$ می‌توانیم به راحتی آن را تأیید کنیم

$$p(z) = \alpha \cdot \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{1}{2\sigma^2}(z-\mu-1)^2} + (1 - \alpha) \cdot \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{1}{2\sigma^2}(z-\mu)^2}$$

که مخلوطی از دو توزیع نرمال $N(\mu + 1, \sigma^2)$ و $N(\mu, \sigma^2)$ با ضرایب $w_1 = \alpha$ و $w_2 = 1 - \alpha$ است. توجه کنید که ترکیب محدب متغیرهای تصادفی $Z = w_1 X + w_2 Y$ به معنای $p_Z(x) = w_1 p_X(x) + w2pY(x)$ نیست.

A.6 نمایش گرافیکی توزیع‌های احتمال

قبلاً دیدیم که یک توزیع احتمال مشترک را می‌توان با استفاده از قانون زنجیره‌ای از معادله (۱,۲) فاکتور گرفت. چنین فاکتورسازی‌هایی را می‌توان با استفاده از یک نمایش گراف جهت‌دار، که در آن گره‌ها متغیرهای تصادفی را نشان می‌دهند و یال‌ها وابستگی را نشان می‌دهند، تجسم کرد. مثلا،

$$p(x, y, z) = p(x)p(y|x)p(z|x, y)$$

در شکل A.3A نشان داده شده است. نمایش‌های گرافیکی توزیع‌های احتمال با استفاده از نمودارهای غیر چرخه‌ای جهت‌دار، همراه با توزیع‌های احتمال شرطی، شبکه‌های بیزی یا شبکه‌های باور نامیده می‌شوند. آنها تفسیر و همچنین استنتاج آماری مؤثر را تسهیل می‌کنند.

تجسم روابط بین متغیرها به ویژه زمانی راحت می‌شود که بخواهیم ویژگی‌های استقلال شرطی متغیرها را درک و تحلیل کنیم. شکل A.3B همان فاکتورگیری $p(x, y, z)$را نشان می‌دهد که در آن متغیر Z مستقل از X با توجه به Y است. با این حال، برای تعیین دقیق ویژگی‌های استقلال شرطی و وابستگی، معمولاً از قوانین جداسازی d برای شبکه‌های اعتقادی استفاده می‌شود. اگرچه روابط اغلب شهودی هستند، گاهی اوقات ویژگی‌های وابستگی به دلیل روابط متعدد بین گره‌ها پیچیده‌تر می‌شوند. به

عنوان مثال، در شکل A.4A، دو گره لبه ندارند، اما به طور مشروط از طریق گره دیگری وابسته هستند. از سوی دیگر، در شکل A.4B، فقدان لبه به معنای استقلال مشروط است. در حال حاضر قوانین جداسازی d را بیشتر بررسی نخواهیم کرد. آنها را می‌توان به راحتی در هر کتاب درسی استاندارد در مورد مدل‌های گرافیکی یافت.

شبکه‌های اعتقادی یک تعریف ساده و رسمی دارند. با توجه به مجموعه‌ای از d متغیرهای تصادفی $X = (X_1, \ldots, X_d)$ شبکه‌های اعتقادی توزیع احتمال مشترک X را به صورت فاکتور می‌گیرند.

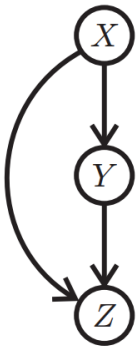
$$p(x) = \prod_{i=1}^{d} p\left(x_i | \boldsymbol{x}_{Parents(X_i)}\right)$$

که در آن $Parents(X)$ اجداد مستقیم گره X را در نمودار نشان می‌دهد. در شکل A.3B، گره Y والد Z است، اما گره X والد Z نیست.

ذکر این نکته ضروری است که چندین راه (چند؟) برای فاکتورگیری توزیع وجود دارد. برای مثال، با معکوس کردن ترتیب متغیرهای $p(x, y, z)$ را نیز می‌توان به صورت فاکتور گرفت

$$p(x, y, z) = p(z)p(y|z)p(x|y, z)$$

A. توزیع احتمال گسسته بدون استقلال شرطی



$P(X = 1)$
0.3

X	$P(Y = 1\|X)$
0	0.5
1	0.9

X	Y	$P(Z = 1\|X, Y)$
0	0	0.3
0	1	0.1
1	0	0.7
1	1	0.4

$$P(X = x, Y = y, Z = z) = P(X = x)P(Y = y|X = x)P(Z = z|X = x, Y = y)$$

B. توزیع احتمال گسسته؛ Z به طور مشروط مستقل از X با Y است

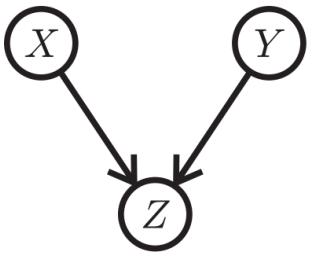
$P(X=1)$
0.3

X	$P(Y=1\|X)$
0	0.5
1	0.9

Y	$P(Z=1\|Y)$
0	0.2
1	0.7

$$P(X=x,Y=y,Z=z)=P(X=x)P(Y=y|X=x)P(Z=z|Y=y)$$

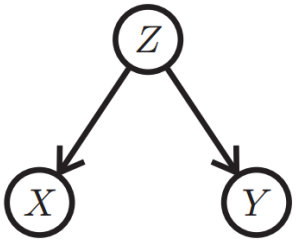
A. X مستقل از Y است، اما Z داده نشده است



$$P(X=x\,|\,Y=y)=P(X=x)$$

$$P(X=x\,|\,Y=y,Z=z)\neq P(X=x\,|\,Z=z)$$

B. X و Y وابسته هستند، اما با توجه به Z به طور مشروط مستقل هستند



$$P(X=x\,|\,Y=y,Z=z)=P(X=x\,|\,Z=z)$$

$$P(Y=y\,|\,X=x,Z=z)=P(Y=y\,|\,Z=z)$$

که نمایش گرافیکی متفاوتی دارد و توزیع‌های احتمال شرطی خاص خود را دارد، اما همان توزیع احتمال مشترک با فاکتورگیری قبلی را دارد. انتخاب فاکتورسازی مناسب و تخمین توزیع‌های احتمال شرطی از داده ها بعداً به تفصیل مورد بحث قرار خواهد گرفت.

از نمودارهای غیر جهت دار نیز می‌توان برای فاکتورسازی توزیع‌های احتمال استفاده کرد. ایده اصلی در اینجا این است که نمودارها را به دسته‌های حداکثر C (کوچکترین مجموعه‌ای از دسته‌هایی که نمودار را پوشش می‌دهد) تجزیه کنیم و توزیع را به شکل زیر بیان کنیم.

$$\mathrm{p(x)}=\frac{1}{\mathrm{Z}}\prod_{\mathrm{C}\in\mathcal{C}}\psi_{\mathrm{C}}(\mathbf{x}_{\mathrm{C}})$$

که در آن هر $\psi_{\mathrm{C}}(\mathrm{x}_{\mathrm{C}})\geq 0$ تابع پتانسیل دسته و نامیده می‌شود

$$Z = \int_x \prod_{C \in \mathcal{C}} \psi C(x_C)\, dx$$

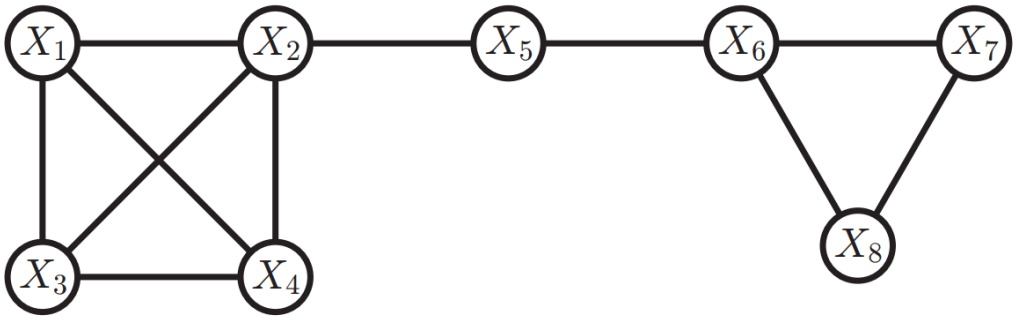
تابع پارتیشن نامیده می‌شود که صرفاً برای اهداف عادی سازی استفاده می‌شود. برخلاف توزیع‌های احتمال شرطی در نمودارهای غیر چرخه‌ای جهت‌دار، پتانسیل‌های دسته معمولاً تفسیر احتمال شرطی ندارند و بنابراین، نرمال‌سازی ضروری است. یک نمونه از تجزیه دسته حداکثر در شکل A.5 نشان داده شده است.

توابع بالقوه معمولاً کاملاً مثبت در نظر گرفته می‌شوند، $\psi_C(\mathbf{x}_C) > 0$، و به این صورت بیان می‌شوند.

$$\psi_C(\mathbf{x}_C) = \exp(-E(\mathbf{x}_C))$$

که در آن $E(\mathbf{x}_C)$ یک تابع انرژی مشخص شده توسط کاربر بر روی دسته متغیرهای تصادفی $\mathbf{X}_C$ است. این منجر به توزیع احتمال شکل زیر می‌شود

$$p(\mathbf{x}) = \frac{1}{Z} \exp\left(\sum_{C \in \mathcal{C}} \log \psi_C(\mathbf{x}_C)\right)$$



$$\boldsymbol{X}_{C_1} = \{X_1, X_2, X_3, X_4\} \qquad \boldsymbol{X}_{C_3} = \{X_5, X_6\}$$

$$\boldsymbol{X}_{C_2} = \{X_2, X_5\} \qquad \boldsymbol{X}_{C_4} = \{X_6, X_7, X_8\}$$

همانطور که فرمول بندی شد، این توزیع احتمال، توزیع بولتزمن یا توزیع گیبس نامیده می‌شود.

تابع انرژی $E(x)$ باید برای مقادیر x که احتمال بیشتری دارند کمتر باشد. همچنین ممکن است شامل پارامترهایی باشد که سپس از داده‌های آموزشی موجود تخمین زده می‌شود. البته، در یک مسئله پیش‌بینی، یک گراف بدون جهت باید ایجاد شود تا متغیرهای هدف را که در اینجا زیرمجموعه‌ای از X در نظر گرفته می‌شوند، نیز در بر گیرد.

اکنون هر توزیع احتمال را بر روی تمام پیکربندی‌های ممکن بردار تصادفی X با نمایش گرافیکی زیربنایی آن در نظر بگیرید. اگر اموال زیر

$$p(x_i | \mathbf{x}_{-X_i}) = p(x_i | \mathbf{x}_{N(X_i)}) \qquad (A.4)$$

راضی است، توزیع احتمال به عنوان شبکه مارکوف یا میدان تصادفی مارکوف نامیده می‌شود. در معادله بالا

$$\mathbf{X}_{-X_i} = (X_1, \ldots, X_{i-1}, X_{i+1}, \ldots, X_d)$$

و $N(X)$ مجموعه‌ای از متغیرهای تصادفی مجاور X در نمودار است. یعنی یک لبه بین X و هر گره در $N(X)$ وجود دارد. به مجموعه متغیرهای تصادفی در $N(X)$ پتوی مارکوف X نیز گفته می‌شود.

می‌توان نشان داد که هر توزیع گیبس ویژگی معادله (A.4) را برآورده می‌کند و برعکس، برای هر توزیع احتمالی که معادله (A.4) برای آن وجود دارد، می‌توان به عنوان یک توزیع گیبس با برخی از پارامترها نشان داد. این هم ارزی توزیع‌های گیبس و شبکه‌های مارکوف توسط قضیه هامرسلی-کلیفورد ایجاد شد.

پیوست B

پس زمینه بهینه سازی

B.1 قوانین اساسی برای گرادیان

برای مشتقات، قوانین مفیدی وجود دارد که با آنها آشنا هستید، مانند $\frac{d}{dw} aw = a$، $\frac{d}{dw} w^2 = 2w$ و $\frac{d}{dw} e^w = e^w$. ما می توانیم به طور مشابه چنین قوانینی را برای تنظیم چند متغیره بنویسیم تا محاسبه گرادیان ها را بدون نیاز به محاسبه هر مشتق جزئی ساده کنیم. هر یک از قوانین زیر را می توان با محاسبه مشتقات جزئی، با قوانینی که برای حالت تک متغیره به آن ها عادت دارید، تأیید کرد. ما قوانین کلیدی این سند را در اینجا خلاصه می کنیم. برای مرجع کامل تر، به کتاب آشپزی ماتریسی [۱۶] مراجعه کنید.

برخی از این قوانین در جدول B.1 خلاصه شده است. این فهرست جامع نیست، اما برخی از قوانین اضافی را امکان‌پذیر می‌سازد. به عنوان مثال، برای به دست آوردن مشتق تابع $f(x) = \boldsymbol{x}^T \boldsymbol{A}$، ابتدا می توان مشتق $f(x)^T = \boldsymbol{A}^T \boldsymbol{x}$ را بدست آورد و سپس جابجایی آن را گرفت زیرا

$$\left(\nabla \mathrm{f}(\mathrm{x})\right)^{\mathrm{T}} = \nabla\left(\mathrm{f}(\mathrm{x})^{\mathrm{T}}\right)$$

بنابراین، چون برای $\nabla \mathrm{f}(\mathbf{x})^T = \mathbf{A}^{\mathrm{T}}$ ما $\nabla f(\boldsymbol{x}) = \boldsymbol{A}$ را دریافت می کنیم

$f(\mathbf{x})$	$\frac{\partial f}{\partial \mathbf{x}}$
$\mathbf{x}^{\top}\mathbf{x}$	$2\mathbf{x}$
$\mathbf{A}\mathbf{x}$	$\mathbf{A}$
$\mathbf{x}^{\top}\mathbf{A}\mathbf{x}$	$\mathbf{A}\mathbf{x}+\mathbf{A}^{\top}\mathbf{x}$

جدول B.1: فرمول های مشتق مفید بردارها با توجه به بردارها. مشتق تابع با مقدار برداری $f: \mathbb{R}^{d\times 1} \rightarrow \mathbb{R}^{m\times 1}$ با توجه به بردار $\boldsymbol{x} \in \mathbb{R}^{d\times 1}$ یک ماتریس $m \times d$ به اندازه M با مولفه‌های $Mij = \partial y_j / \partial x_i, i \in \{1,2,\ldots,d\}$ است و $j \in \{1,2,\ldots,m\}$. یک مشتق از اسکالر با توجه به یک بردار، که در آن $m = 1$، یک مورد خاص از این وضعیت است که منجر به بردار ستونی $d \times 1$ می شود. توجه داشته باشید که در جدول، m برای هر ردیف یکسان نیست. به عنوان مثال، $f(x) = \boldsymbol{x}^T \boldsymbol{x}$ یک اسکالر است، در حالی که برای یک ماتریس عمومی $A \in \mathbb{R}^{m\times d}$، $f(\boldsymbol{x}) = \boldsymbol{Ax}$ یک بردار m بعدی است.

پیوست C

پس زمینه جبر خطی

c.1 دیدگاه جبری

ابزار قدرتمند دیگری برای تجزیه و تحلیل و درک رگرسیون خطی از جبر خطی خطی و کاربردی می آید. در این بخش ما مسیری انحرافی می‌کنیم تا به مبانی جبر خطی بپردازیم و سپس این مفاهیم را برای عمیق‌تر کردن درک خود از رگرسیون به کار ببریم. در جبر خطی، ما اغلب علاقه مند به حل مجموعه معادلات زیر هستیم که در زیر به صورت ماتریسی آورده شده است.

$$\boldsymbol{A}x = b(C.1)$$

در اینجا، $\boldsymbol{A}$ یک ماتریس $m \times n$، $\mathbf{b}$ یک بردار $m \times 1$، و x یک بردار $n \times 1$ است که باید پیدا شود. تمام عناصر $\boldsymbol{A}$، x و $\boldsymbol{b}$ به عنوان اعداد واقعی در نظر گرفته می شوند. ما با یک سناریوی ساده شروع می کنیم و فرض می کنیم $\boldsymbol{A}$ یک مربع، ماتریس 2×2 است. این مجموعه از معادلات را می توان به صورت بیان کرد

$$a_{11}x_1 + a_{12}x_2 = b_1$$

$$a_{21}x_1 + a_{22}x_2 = b_2$$

به عنوان مثال، ممکن است ما علاقه‌مند به حل باشیم

$$x_1 + 2x_2 = 3$$

$$x_1 + 3x_2 = 5$$

این یک فرمول مناسب زمانی است که ما می خواهیم سیستم را حل کنیم، به عنوان مثال. با حذف گاوسی با این حال، فرمول مناسبی برای درک مسئله وجود راه حل ها نیست. برای اینکه بتوانیم این کار را انجام دهیم، مفاهیم اساسی جبر خطی را به اختصار مرور می کنیم.

C.1.1 چهار زیرفضای اساسی

هدف این بخش بررسی مختصر چهار زیرفضای اساسی در جبر خطی (فضای ستون، فضای ردیف، خالی، فضای خالی سمت چپ) و رابطه متقابل آنهاست. ما با مثال خود از بالا شروع می کنیم و سیستم معادلات خطی را به صورت زیر می نویسیم

$$\begin{bmatrix}1\\1\end{bmatrix}x_1 + \begin{bmatrix}2\\3\end{bmatrix}x_2 = \begin{bmatrix}3\\5\end{bmatrix}$$

اکنون می بینیم که با حل $\boldsymbol{Ax} = \boldsymbol{b}$ به دنبال مقادیر مناسب بردارهای $(1,1)$ و $(2,3)$ هستیم تا ترکیب خطی آنها $(3,5)$ را ایجاد کند. این مقادیر $\mathrm{x}_1 = -1$ و $x_2 = 2$هستند. اجازه دهید $\mathbf{a}_1 = (1,1)$و $\mathbf{a}_2 = (2,3)$ را به عنوان بردارهای ستون $\mathbf{A}$ تعریف کنیم. یعنی $\mathbf{A} = [\mathbf{a}_1\mathbf{a}_2]$. بنابراین، هر زمان که $\mathbf{b}$ را بتوان به صورت ترکیب خطی از بردارهای ستون $\mathbf{a}_1$ و $\boldsymbol{a}_2$ بیان کرد، $\mathbf{Ax} = \mathbf{b}$ قابل حل خواهد بود.

تمام ترکیبات خطی ستون های ماتریس $\mathbf{A}$ فضای ستون $\boldsymbol{C(A)}$، را با بردارهای $\boldsymbol{a_1 \ldots a_n}$ تشکیل می دهنداساس این فضا است. هر دو $\boldsymbol{b}$ و $\boldsymbol{C(A)}$ در فضای m بعدی $\mathbb{R}^m$ قرار دارند. بنابراین، آنچه $\boldsymbol{Ax} = \boldsymbol{b}$ می گوید این است که $\boldsymbol{b}$ باید در فضای ستون $\boldsymbol{A}$ قرار داشته باشد تا معادله جواب داشته باشد. در مثال بالا، اگر ستون‌های $\boldsymbol{A}$ به‌طور خطی مستقل باشند، راه‌حل منحصربه‌فرد است، یعنی تنها یک ترکیب خطی از بردارهای ستون وجود دارد که $\boldsymbol{b}$ را نشان می‌دهد. در غیر این صورت، چون $\boldsymbol{A}$ یک ماتریس مربع است، سیستم هیچ راه حلی ندارد. نمونه ای از چنین وضعیتی است

$$\begin{bmatrix} 1 & 2 \\ 1 & 2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 3 \\ 5 \end{bmatrix}$$

که در آن $\mathbf{a}_1 = (1,1)$و$(2,2) = \mathbf{a}_2$. در اینجا، $\boldsymbol{a}_1$ و $\boldsymbol{a}_2$ (خطی) وابسته هستند زیرا$2\mathbf{a}_1 - \mathbf{a}_2 = 0$. بین فضاهای ایجاد شده توسط مجموعه ای از بردارها و ویژگی های ماتریس $\boldsymbol{A}$ ارتباط عمیقی وجود دارد. در حال حاضر، با استفاده از مثال بالا، کافی است بگوییم که اگر $\boldsymbol{a}_1$ و $\boldsymbol{a}_2$ مستقل باشند، ماتریس $\mathbf{A}$ غیر مفرد است (تفرد را فقط برای ماتریس های مربع می توان مورد بحث قرار داد)، که دارای رتبه کامل است.

به شیوه ای معادل با فضای ستون، تمام ترکیبات خطی ردیف های $\boldsymbol{A}$ فضای ردیف را تشکیل می دهند که با $\boldsymbol{C(A^T)}$ نشان داده می شود، که در آن هر دو x و $\boldsymbol{C(A^T)}$ در $\mathbb{R}^n$ هستند. تمام راه حل های $\boldsymbol{Ax} = 0$، فضای خالی ماتریس، $\boldsymbol{N(A)}$ را تشکیل می دهند، در حالی که تمام راه حل های $\boldsymbol{A\hat{}T}\ y = 0$ فضای خالی سمت چپ $\boldsymbol{A}$، $\boldsymbol{N(A\hat{}T)}$ را تشکیل می دهند. واضح است که $\boldsymbol{C(A)}$ و $\boldsymbol{N(A^T)}$ در $\mathbb{R}^m$ تعبیه شده اند، در حالی که $\boldsymbol{C(A^T)}$ و $\boldsymbol{N(A)}$ در $\mathbb{R}^n$ هستند. با این حال، جفت فضاهای فرعی متعامد هستند (بردارهای u و v متعامد هستند اگر $u^T v = 0$). یعنی هر بردار در $\boldsymbol{C(A)}$ با تمام بردارهای $\boldsymbol{N(A^T)}$ متعامد است و هر بردار در $\boldsymbol{C(A^T)}$ نسبت به همه بردارهای $\boldsymbol{N(A)}$ متعامد است. این به راحتی قابل مشاهده است: اگر $x \in N(A)$، پس طبق تعریف $\boldsymbol{Ax} = 0$، و بنابراین هر ردیف از $\boldsymbol{A}$ متعامد به $\boldsymbol{x}$ است. اگر هر سطر بر $\boldsymbol{x}$ متعامد باشد، تمام ترکیبات خطی سطرها نیز متعامد هستند.

متعامد بودن ویژگی کلیدی چهار زیرفضا است، زیرا تجزیه مفید بردارهای x و $\boldsymbol{b}$ را از معادله فراهم می کند. (C.1) با توجه به $\boldsymbol{A}$ (ما در بخش بعدی از آن استفاده خواهیم کرد). برای مثال، هر $x \in \mathbb{R}^n$ را می توان به صورت تجزیه کرد

که در آن $x_r \in \boldsymbol{C(A^T)}$ و $x_n \in \boldsymbol{N(A)}$، به طوری که $\left\|x\right\|_2^2 = \left\|x_r\right\|_2^2 + \left\|x_n\right\|_2^2$. به طور مشابه، هر $b \in \mathbb{R}^m$ را می توان به صورت تجزیه کرد

$$\mathbf{b} = \mathbf{b}_\mathrm{c} + \mathbf{b}_\mathrm{l}$$

که $\mathbf{b}_\mathrm{c} \in \mathrm{C}(\mathbf{A})$، $\mathbf{b}_\mathrm{l} \in \mathrm{N}(\mathbf{A}^T)$ و $\left\|\boldsymbol{b}\right\|_2^2 = \left\|\boldsymbol{b_c}\right\|_2^2 + \left\|\boldsymbol{b}\right\|_2^2$

در بالا اشاره کردیم که ویژگی‌های فضاهای بنیادی با ویژگی‌های ماتریس $\mathbf{A}$ ارتباط تنگاتنگی دارند. برای نتیجه‌گیری این بخش، اجازه دهید به طور خلاصه در مورد رتبه یک ماتریس و رابطه آن با ابعاد زیرفضاهای بنیادی بحث کنیم. اساس فضا کوچکترین مجموعه بردارهایی است که فضا را در بر می گیرد (این مجموعه بردارها منحصر به فرد نیستند). اندازه پایه را بعد فضا نیز می گویند. در مثال ابتدای این بخش، یک فضای ستون دو بعدی با بردارهای پایه $a1$ = (۱،۱) و $a2$ = (۲،۳) داشتیم. از سوی دیگر،

برای $a_1 = (1,1)$ و $a_2 = (2,3)$یک فضای ستون یک بعدی، یعنی یک خط، به طور کامل توسط هر یک از بردارهای پایه تعیین می شود. جای تعجب نیست که بعد فضای پوشیده شده توسط بردارهای ستون برابر با رتبه ماتریس $\mathbf{A}$ است. یکی از نتایج اساسی در جبر خطی این است که رتبه $\mathbf{A}$ با بعد $\mathbf{C(A)}$یکسان است که به نوبه خود با بعد یکسان است. بعد $\mathbf{C(A^T)}$.

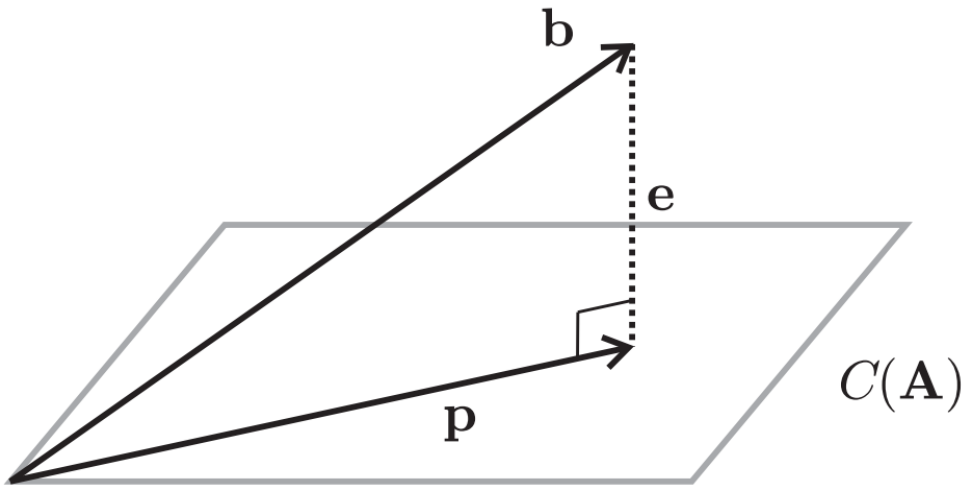
C.1.2 به حداقل رساندن $\mathbf{||Ax - b||_2^2}$

اجازه دهید اکنون دوباره به راه حل های $\boldsymbol{Ax = b}$ نگاه کنیم. به طور کلی، سه نتیجه متفاوت وجود دارد:

۱. هیچ راه حلی برای سیستم وجود ندارد

۲. یک راه حل منحصر به فرد برای سیستم وجود دارد، و

۳. بی نهایت راه حل وجود دارد.



شکل C.1: تصویر طرح ریزی بردار $\boldsymbol{b}$ به فضای ستون ماتریس $\boldsymbol{A}$. بردارهای $\boldsymbol{p}$ (b_c) و $\boldsymbol{e}(b_l)$ به ترتیب نقطه طرح و خطا را نشان می دهند.

این نتایج به رابطه بین رتبه (r)و مقدار $\mathbf{A}$ و ابعاد m و n بستگی دارد. ما قبلاً می دانیم که وقتی $r = m = n$ (ماتریس مربع، معکوس، رتبه کامل $\mathbf{A}$) یک راه حل منحصر به فرد برای سیستم وجود دارد، اما اجازه دهید موقعیت های دیگر را بررسی کنیم. به طور کلی، هنگامی که $r = n < m$ (رتبه ستون کامل)، سیستم یا یک راه حل دارد یا هیچ راه حلی ندارد، همانطور که به صورت لحظه ای خواهیم دید. وقتی $r = m < n$ (رتبه ردیف کامل)، سیستم بی نهایت راه حل دارد. در نهایت، در مواردی که $r < m$ و $r < n$، یا هیچ راه حلی وجود ندارد یا بی نهایت راه حل وجود دارد. از آنجا که $\mathbf{Ax = b}$ ممکن است قابل حل نباشد، ما حل $\mathbf{Ax = b}$ را به مینیمم کردن $\mathbf{||Ax - b||_2}$ تعمیم می دهیم. به این ترتیب می توان همه موقعیت ها را در یک چارچوب یکپارچه در نظر گرفت.

اجازه دهید مثال زیر را در نظر بگیریم

$$\mathbf{A} = \begin{bmatrix} 1 & 2 \\ 1 & 3 \\ 1 & 4 \end{bmatrix}, \boldsymbol{x} = \begin{bmatrix} \boldsymbol{x_1} \\ \boldsymbol{x_2} \end{bmatrix}, \boldsymbol{b} = \begin{bmatrix} \boldsymbol{b_1} \\ \boldsymbol{b_2} \\ \boldsymbol{b_3} \end{bmatrix}$$

که نمونه ای را نشان می دهد که بعید است راه حلی برای $\mathbf{Ax} = \mathbf{b}$ داشته باشیم، مگر اینکه محدودیتی در b_1، b_2 و b_3 وجود داشته باشد. در اینجا، محدودیت $b_3 = 2b_2 - b_1$ است. در این وضعیت، $C(\boldsymbol{A})$ یک صفحه دو بعدی در $\mathbb{R}^3$ است که توسط بردارهای ستون $\mathrm{a}_1 = (1,1,1)$ و $\mathrm{a}_2 = (2,3,4)$پوشیده شده است. اگر محدودیت عناصر $\boldsymbol{b}$ برآورده نشود، هدف ما این است که سعی کنیم نقطه ای را در $C(\boldsymbol{A})$ پیدا کنیم که نزدیکترین نقطه به $\boldsymbol{b}$ باشد. همانطور که در شکل $\boldsymbol{C.1}$ نشان داده شده است، این نقطه ای است که $\boldsymbol{b}$ به $C(\boldsymbol{A})$ پیش بینی می شود. ما به طرح ریزی $\boldsymbol{b}$ به $C(\boldsymbol{A})$ به صورت $\boldsymbol{p}$ اشاره خواهیم کرد. حال با استفاده از نماد جبری استاندارد معادلات زیر را داریم

$$\mathbf{b} = \mathbf{p} + \mathbf{e}$$

$$\mathbf{p} = \mathbf{Ax}$$

از آنجایی که $\mathbf{p}$ و $\mathbf{e}$ متعامد هستند، می دانیم که $\mathbf{p}^{\boldsymbol{T}}\mathbf{e} = 0$. اجازه دهید $\boldsymbol{x}$ را حل کنیم

$$(\mathbf{Ax})^{\mathrm{T}}(\mathbf{b} - \mathbf{Ax}) = 0$$

$$\mathbf{x}^{\mathrm{T}}\mathbf{A}^{\mathrm{T}}\mathbf{b} - \mathbf{x}^{\mathrm{T}}\mathbf{A}^{\mathrm{T}}\mathbf{Ax} = 0$$

$$\mathbf{x}^{\mathrm{T}}\left(\mathbf{A}^{\mathrm{T}}\mathbf{b} - \mathbf{A}^{\mathrm{T}}\mathbf{Ax}\right) = 0$$

و بنابراین

$$\mathbf{x}^{*} = \left(\mathbf{A}^{\mathrm{T}}\mathbf{A}\right)^{-1}\mathbf{A}^{\mathrm{T}}\mathbf{b}$$

این دقیقاً همان راه حلی است که مجموع مجذور خطاها را به حداقل رساند و احتمال را به حداکثر رساند. ماتریکس

$$\mathbf{A}^{\dagger} = \left(\mathbf{A}^{\mathrm{T}}\mathbf{A}\right)^{-1}\mathbf{A}^{\mathrm{T}}$$

شبه معکوس مور-پنروز یا به سادگی یک شبه معکوس نامیده می شود. این یک ماتریس مهم است زیرا همیشه وجود دارد و منحصر به فرد است، حتی در شرایطی که معکوس $\boldsymbol{A^TA}$ وجود ندارد.

این زمانی اتفاق می‌افتد که $\mathbf{A}$ دارای ستون‌های وابسته باشد (از لحاظ فنی، $\mathbf{A}$ و $\mathbf{A^TA}$فضای خالی یکسانی دارند که حاوی بیش از مبدأ سیستم مختصات است؛ بنابراین رتبه $\mathbf{A^TA}$ کمتر از n است). اجازه دهید برای لحظه ای به بردار طرح ریزی $\boldsymbol{p}$ نگاه کنیم. ما داریم

$$\mathbf{p} = \mathbf{Ax}$$

$$= \mathbf{A}\left(\mathbf{A}^{\mathrm{T}}\mathbf{A}\right)^{-1}\mathbf{A}^{\mathrm{T}}\mathbf{b}$$

که در آن $A(A^TA)^{-1A^T}$ماتریسی است که $\boldsymbol{b}$ را به فضای ستون $\boldsymbol{A}$ می فرستد

در حالی که ما به همان نتیجه ای که در بخش های قبلی بود رسیدیم، ابزار جبر خطی به ما اجازه می دهد تا رگرسیون OLS را در سطح عمیق تری مورد بحث قرار دهیم. اجازه دهید لحظه ای وجود و تعدد راه حل ها را بررسی کنیم

$$\underset{\mathrm{x}}{\arg\min} \left|\left|\mathbf{Ax} - \mathbf{b}\right|\right|_2 \qquad (C.2)$$

واضح است که راه حل این مشکل همیشه وجود دارد. با این حال، اکنون خواهیم دید که راه حل این مشکل به طور کلی منحصر به فرد نیست و به رتبه $\boldsymbol{A}$ بستگی دارد. $\boldsymbol{x}$ را یک راه حل برای معادله در نظر بگیرید. $(C.2)$. به یاد بیاورید که $x = x_r + x_n$ و در $\boldsymbol{A}$ ضرب می شود. بنابراین، هر بردار $\boldsymbol{x} = \boldsymbol{x_r} + \alpha \boldsymbol{x_n}$، که $\alpha \in \mathbb{R}$، نیز یک راه حل است. توجه داشته باشید که x_r در همه این راه حل ها مشترک است. اگر نمی توانید آن را ببینید، فرض کنید بردار دیگری از فضای ردیف وجود دارد و نشان دهید که امکان پذیر نیست. اگر ستون های $\boldsymbol{A}$ مستقل باشند، راه حل منحصر به فرد است زیرا فضای خالی فقط مبدا را در بر می گیرد. در غیر این صورت، بی نهایت راه حل وجود دارد. در چنین مواردی، با فرافکنی $\boldsymbol{b}$ به $\boldsymbol{C(A)}$ دقیقاً چه راه حلی پیدا می شود؟ اجازه دهید به آن نگاه کنیم:

$$\mathbf{x}^* = \mathbf{A}^\dagger \mathbf{b}$$

$$= \left(\mathbf{A}^{\mathrm{T}}\mathbf{A}\right)^{-\mathbf{1}}\mathbf{A}^{\mathrm{T}}(\mathbf{p} + \mathbf{e})$$

$$= (\mathbf{A}^T\mathbf{A})^{-\mathbf{1}}\mathbf{A}^T\mathbf{p}$$

$$= \boldsymbol{x_r}$$

به عنوان $\mathbf{p} = \mathbf{A}\mathbf{x}_{\mathrm{r}}$. با توجه به اینکه $\boldsymbol{x_r}$ منحصر به فرد است، راه حلی که با بهینه سازی مینیمم مربعات یافت می شود، راه حلی است که به طور همزمان $\left|\left|\boldsymbol{Ax} - \boldsymbol{b}\right|\right|_2$ و $\left|\left|\boldsymbol{x}\right|\right|_2$ را به حداقل می رساند (مشاهده کنید که $\left|\left|\boldsymbol{x}\right|\right|_2$ به حداقل می‌رسد زیرا راه حل هر جزء از فضای خالی را نادیده می گیرد). بنابراین، مشکل رگرسیون OLS گاهی اوقات به عنوان مسئله مینیمم نرم مینیمم مربع نامیده می‌شود

حال بیایید موقعیت هایی را در نظر بگیریم که در آن $\mathbf{Ax} = \mathbf{b}$ راه حل های بی نهایت زیادی دارد. یعنی وقتی $\mathrm{b} \in \mathrm{C}(\mathbf{A})$. این معمولاً زمانی رخ می دهد که $\mathrm{r} \leq \mathrm{m} < \mathrm{n}$ باشد. در اینجا، به دلیل اینکه $\mathbf{b}$ قبلاً در فضای ستون $\mathbf{A}$ قرار دارد، تنها سؤال این است که با روش کمینه‌سازی چه راه‌حل خاصی پیدا می‌شود. همانطور که در بالا دیدیم، نتیجه فرآیند کمینه سازی راه حلی با مینیمم نرم L_2 $\left|\left|\mathbf{x}\right|\right|_2$ است.

به طور خلاصه، اجازه دهید ابتدا به نماد اصلی خود بازگردیم که در آن $\mathbf{X}$ ماتریس و $\mathbf{w}$ وزن هایی هستند که باید پیدا شوند. هدف از مسئله رگرسیون OLS حل $\mathbf{Xw} = \mathbf{y}$ است، اگر قابل حل باشد. وقتی $\mathbf{d} < \mathbf{n}$ این یک سناریوی واقع بینانه در عمل نیست. بنابراین، ما نیاز را کاهش دادیم و سعی کردیم نقطه ای را در فضای ستون $\mathrm{C}(\mathbf{X})$ که نزدیکترین به $\mathbf{y}$ است را پیدا کنیم. معلوم شد که این معادل به حداقل رساندن مجموع خطاهای مربع (یا فاصله اقلیدسی) بین بردارهای n بعدی $\mathbf{Xw}$ و $\mathbf{y}$ است. همچنین مشخص شد که معادل راه حل حداکثر احتمال ارائه شده در بخش ۵,۱ است. وقتی $\mathrm{n} < \mathrm{d}$، یک وضعیت معمول در عمل این است که بی نهایت راه حل وجود دارد. در این شرایط، الگوریتم بهینه‌سازی ما الگوریتمی را با مینیمم نرم L_2 پیدا می‌کند.

پیوست D

جزئیات در مورد رویکردهای نمایندگی بدون نظارت با استفاده از فاکتورسازی

انواع مختلفی از الگوریتم های یادگیری بدون نظارت وجود دارد که در واقع با فاکتورسازی ماتریس داده ها مطابقت دارند که در جدول ***D.1*** خلاصه می کنیم. در بسیاری از موارد، آنها به سادگی با تعریف یک هسته جالب در ***X***، و سپس فاکتورگیری آن هسته (یعنی هسته PCA) مطابقت دارند. اگر ورودی خالی باشد، هیچ قاعده‌سازی و هیچ محدودیتی را مشخص نمی‌کند. برای فهرست کامل تر، [۱۸] و [۲۰] را ببینید. همانند تنظیمات رگرسیون، می‌توانیم این اتلاف اقلیدسی را به هر زیان محدب تعمیم دهیم

$$\mathrm{L_x}(\mathbf{H},\mathbf{D},\mathbf{X}) = \sum_{\mathrm{i}=1}^{\mathrm{n}} \mathrm{L_x}(\mathbf{H_i}:\mathbf{D},\mathbf{X_i}:)$$

جایی که در بالا استفاده کردیم

$$\mathrm{L_x}(\mathbf{H},\mathbf{D},\mathbf{X}) = \sum_{i=1}^{n} \left|\left|\mathbf{H}_{i:}\mathbf{D} - \mathbf{X}_{i:}\right|\right| = \left|\left|\mathbf{HD} - \mathbf{X}\right|\right|_{\mathrm{F}}^{2}$$

الگوریتم هایی برای یادگیری لغت نامه ها

تمرکز ما بر پیش‌بینی است، و بنابراین مایلیم از این نمایش‌ها برای یادگیری تحت نظارت (یا نیمه نظارت) استفاده کنیم. ما از یک رویکرد دو مرحله ای استفاده می کنیم، که در آن ابتدا نمایش جدید به روشی بدون نظارت یاد می شود و سپس با الگوریتم های یادگیری نظارت شده استفاده می شود. این دو مرحله را می توان با یادگیری فرهنگ لغت تحت نظارت در یک مرحله ترکیب کرد. برای بحث در مورد این رویکرد پیشرفته تر به [۱۳] مراجعه کنید.

رایج ترین استراتژی برای یادگیری این مدل های فرهنگ لغت، انجام یک کمینه سازی متناوب بر روی متغیرها است. بهینه سازی روی ***D*** و ***H*** به طور مشترک محدب نیست. با این حال، در هر متغیر به طور جداگانه محدب است. استراتژی این است که یک متغیر را ثابت کنیم، مثلا ***H***، و در دیگری نزول کنیم، مثلا ***D***، و سپس سوئیچ کنیم، ***D*** را ثابت کنیم و در ***H*** نزولی کنیم. این کمینه سازی متناوب تا زمان همگرایی ادامه می یابد. اگرچه این یک بهینه سازی غیر محدب است، شواهد اخیری وجود دارد که نشان می دهد این روش در واقع مینیمم مطلق را برمی گرداند (به عنوان مثال [۱۲] مراجعه کنید). ما این روش را در الگوریتم ۵ خلاصه می کنیم.

هنگامی که فرهنگ لغت $\mathbf{D}$ و نمایش جدید $\mathbf{H}$ یاد گرفتیم، می توانیم وزن های نظارت شده $\mathbf{W} \in \mathbb{R}^{k \times m}$ را برای به دست آوردن $\boldsymbol{HW} \approx \boldsymbol{Y}$ یاد بگیریم. این را می توان با هر یک از روش های رگرسیون خطی یا طبقه بندی که تاکنون آموخته ایم انجام داد.

در نهایت، ما باید بدانیم که چگونه از این مدل های آموخته شده برای پیش بینی خارج از نمونه (یعنی برای نمونه های جدید) استفاده کنیم. ماتریس‌های $\mathbf{D}$ و $\mathbf{W}$ حاوی تمام اطلاعات لازم برای انجام پیش‌بینی خارج از نمونه هستند و $\mathbf{H}$ نیازی به ذخیره‌سازی ندارد، زیرا این نمایش مختص داده‌های آموزشی بود. برای نمونه جدید $\boldsymbol{x}_{new}$، نمایش را می توان با استفاده از آن به دست آورد

$$\text{h\_\{new\}} = \underset{\text{h} \in \mathbb{R}\text{\^{}k}}{\text{argmin}} \ \text{L\_x}(\mathbf{hD}, \mathbf{x})$$

با نمایش این نمونه، می‌توانیم $f(h_{new}\boldsymbol{W})$را پیش‌بینی کنیم.

Algorithm	Loss and constraints
CCA ≡ orthonormal PLS	$\left\| \begin{bmatrix} \mathbf{X}(\mathbf{X}^\top\mathbf{X})^{-1} \\ \mathbf{Y}(\mathbf{Y}^\top\mathbf{Y})^{-1} \end{bmatrix} - \mathbf{HD} \right\|_F^2$
Isomap	$\|\mathbf{K} - \mathbf{HD}\|_F^2$ $\mathbf{K} = -\frac{1}{2}(\mathbf{I} - \mathbf{ee}')\mathbf{S}(\mathbf{I} - \mathbf{ee}')$ with $\mathbf{S}_{i,j} = \|\mathbf{X}_{i:} - \mathbf{X}_{j:}\|$
K-means clustering	$\|\mathbf{X} - \mathbf{HD}\|_F^2$ with $\mathbf{H} \in \{0,1\}^{n \times k}$, $\mathbf{H1} = \mathbf{1}$
K-medians clustering	$\|\mathbf{X} - \mathbf{HD}\|_{1,1}$ with $\mathbf{H} \in \{0,1\}^{n \times k}$, $\mathbf{H1} = \mathbf{1}$
Laplacian eigenmaps ≡ Kernel LPP	$\|\mathbf{K} - \mathbf{HD}\|_F^2$ for $\mathbf{K} = \mathbf{L}^\dagger$
Metric multi-dimensional scaling	$\|\mathbf{K} - \mathbf{HD}\|_F^2$ for isotropic kernel $\mathbf{K}$
Normalized-cut	$\left\|(\mathbf{\Lambda}^{-1}\mathbf{X} - \mathbf{HD})\mathbf{\Lambda}^{1/2}\right\|_F^2$ with $\mathbf{H} \in \{0,1\}^{n \times k}$, $\mathbf{H1} = \mathbf{1}$
Partial least squares	$\|\mathbf{XY}' - \mathbf{DH}\|_F^2$
PCA	$\|\mathbf{X} - \mathbf{HD}\|_F^2$
Kernel PCA	$\|\mathbf{K} - \mathbf{HD}\|_F^2$
Ratio cut	$\|\mathbf{K} - \mathbf{HD}\|_F^2$ for $\mathbf{K} = \mathbf{L}^\dagger$

شکل D.1: الگوریتم‌های یادگیری بدون نظارت که با فاکتورسازی ماتریسی مطابقت دارند.

Algorithm 5: Alternating minimization for dictionary learning

Input:

inner dimension k
loss L, where $L(\boldsymbol{HD}) = L_x(\boldsymbol{H}, \boldsymbol{D}, \boldsymbol{X})$
R_D, the regularizer on $\boldsymbol{D}$
R_H, the regularizer on $\boldsymbol{H}$
the regularization weight λ
convergence tolerance
fixed positive step-sizes η_D, η_H
dataset $x_1, \ldots, x_n$

Initialization:

$\boldsymbol{D}, \boldsymbol{H} \leftarrow$ full-rank random matrices with inner dimension k
prevobj $\leftarrow \infty$

Loop until convergence within tolerance or reach maximum number of iterations:

Update $\boldsymbol{D}$ using one step of gradient descent
Update $\boldsymbol{H}$ using one step of gradient descent
currentobj $\leftarrow L(\boldsymbol{HD}) + \lambda R_D(\boldsymbol{D}) + \frac{\lambda}{n} R_H(\boldsymbol{H})$
If |currentobj − prevobj| < tolerance, Then break
prevobj $\leftarrow$ currentobj

Output:

$\boldsymbol{D}$, $\boldsymbol{H}$

پیوست E

تخمین بیزی

رویکردهای حداکثر پسین و حداکثر درستنمایی راه حلی را گزارش می کنند که به ترتیب با حالت توزیع پسین و تابع درستنمایی مطابقت دارد. با این حال، این رویکرد امکان توزیع‌های اریب، توزیع‌های چندوجهی یا صرفاً مناطق بزرگ با مقادیر مشابه $p(f|D)$ را در نظر نمی‌گیرد. تخمین بیزی به این نگرانی ها می پردازد.

ایده اصلی در آمار بیزی، به حداقل رساندن ریسک پسین است

$$\mathrm{R} = \int_{\mathcal{F}} \ell(\mathrm{f}, \hat{\mathrm{f}}) \cdot \mathrm{p}(\mathrm{f}|\mathcal{D})\mathrm{df}$$

که در آن $\hat{f}$ تخمین ما و $\ell(\mathrm{f}, \hat{\mathrm{f}})$مقداری تابع هزینه بین دو مدل است. هنگامی که $\ell(f, \hat{f}) = (f - \hat{f})^2$ (سوء استفاده از علامت گذاری را نادیده بگیرید)، می توانیم خطر بعدی را به صورت زیر به حداقل برسانیم.

$$\frac{\partial}{\partial \hat{\mathrm{f}}}\mathrm{R} = 2\hat{\mathrm{f}} - 2\int_{\mathcal{F}} \mathrm{f} \cdot \mathrm{p}(\mathrm{f}|\mathcal{D})\mathrm{df} \; = \; 0$$

که از آن می‌توان به دست آورد که مینیمم کردن ریسک پسین، تابع میانگین پسین است. یعنی

$$\mathrm{f_B} = \int_{\mathcal{F}} \mathrm{f} \cdot \mathrm{p}(\mathrm{f}|\mathcal{D})\mathrm{df} \; = \; \mathbb{E}[F|\mathcal{D}]$$

که در آن F یک متغیر تصادفی است که مدل را نشان می دهد. ما باید به f_B به عنوان تخمینگر بیز اشاره کنیم. ذکر این نکته ضروری است که محاسبه میانگین پسین معمولاً شامل حل انتگرال های پیچیده است. در برخی شرایط، این انتگرال ها را می‌توان به صورت تحلیلی حل کرد. در برخی دیگر، ادغام عددی ضروری است.

مثال ۲۶: اجازه دهید $\mathcal{D} \; = \; \{2, 5, 9, 5, 4, 8\}$دوباره یک $i.i.d$ باشد. نمونه از پواسون (λ_0) فرض کنید دانش قبلی در مورد پارامتر توزیع را می توان با استفاده از توزیع گاما با پارامترهای $k \; = \; 3$ و $\theta \; = \; 1$بیان کرد. تخمین بیزی λ_0 را بیابید.

می خواهیم $\mathbb{E}[\Lambda|\mathcal{D}]$ را پیدا کنیم. اجازه دهید ابتدا توزیع پسین را به صورت بنویسیم

$$p(\lambda|\mathcal{D}) = \frac{p(\mathcal{D}|\lambda)p(\lambda)}{p(\mathcal{D})}$$

$$= \square\{\, \mathrm{p}(\mathcal{D}|\lambda)\mathrm{p}(\lambda)\}\{\int_0^{\infty} \mathrm{p}(\mathcal{D}|\lambda)\mathrm{p}(\lambda)\mathrm{d}\lambda$$

که در آن، همانطور که در مثال های قبلی نشان داده شد، آن را داریم

$$\mathrm{p}(\mathcal{D}|\lambda) = \frac{\lambda \sum_{\mathrm{i}=1}^{\mathrm{n}} \mathrm{x_i} \cdot \mathrm{e}^{-\mathrm{n}\lambda}}{\prod_{\mathrm{i}=1}^{\mathrm{n}} \mathrm{x_i}!}$$

و

$$\mathrm{p}(\lambda) \frac{\lambda^{k-1} e^{-\frac{\lambda}{\theta}}}{\theta^{\mathrm{k}} \Gamma(\mathrm{k})}$$

قبل از محاسبه $p(\mathcal{D})$، ابتدا به این نکته توجه کنیم

$$\int_0^{\infty} \mathrm{x}^{\alpha-1} \mathrm{e}^{-\beta \mathrm{x}} \mathrm{dx} = \frac{\Gamma(\alpha)}{\beta^{\alpha}}$$

اکنون، ما می توانیم آن را استخراج کنیم

$$\mathrm{p}(\mathcal{D}) = \int_0^{\infty} \mathrm{p}(\mathcal{D}|\lambda)\mathrm{p}(\lambda)\mathrm{d}\lambda$$

$$= \int_0^{\infty} \frac{\sum_{i=1}^{n} x_i \cdot e^{-n\lambda}}{\lambda \prod_{\mathrm{i}=1}^{\mathrm{n}} x_i!} \cdot \frac{\lambda^{\mathrm{k}-1} \mathrm{e}^{-\frac{\lambda}{\theta}}}{\theta^{\mathrm{k}} \Gamma(\mathrm{k})} \, \mathrm{d}\lambda$$

$$= \frac{\Gamma(\mathrm{k} + \sum_{\mathrm{i}=1}^{\mathrm{n}} \mathrm{x_i})}{\theta^{\mathrm{k}} \Gamma(\mathrm{k}) \prod_{\mathrm{i}=1}^{\mathrm{n}} \mathrm{x_i}! \left(\mathrm{n} + \frac{1}{\theta}\right) \sum_{\mathrm{i}=1}^{\mathrm{n}} \mathrm{x_i} + \mathrm{k}}$$

و متعاقبا آن

$$\mathrm{p}(\lambda|\mathcal{D}) = \frac{\mathrm{p}(\mathcal{D}|\lambda)\mathrm{p}(\lambda)}{\mathrm{p}(\mathcal{D})}$$

$$= \frac{\lambda \sum_{\mathrm{i}=1}^{\mathrm{n}} \mathrm{x_i} \cdot \mathrm{e}^{-\mathrm{n}\lambda}}{\prod_{\mathrm{i}=1}^{\mathrm{n}} \mathrm{x_i}!} \cdot \frac{\lambda^{k-1} e^{-\lambda\theta}}{\theta^k \Gamma(k)} \cdot \frac{\theta^k \Gamma(k) \prod_{i=1}^{n} x_i! \, (n+1\theta) \sum_{i=1}^{n} x_i + k}{\Gamma(k + \sum_{i=1}^{n} x_i)}$$

$$= \frac{\lambda^{\mathrm{k}-1} + \sum_{\mathrm{i}=1}^{\mathrm{n}} \mathrm{x_i} \cdot \mathrm{e}^{-\lambda(\mathrm{n}+1/\theta)} \cdot \left(\mathrm{n} + \frac{1}{\theta}\right) \sum_{\mathrm{i}=1}^{\mathrm{n}} \mathrm{x_i} + \mathrm{k}}{\Gamma(\mathrm{k} + \sum_{\mathrm{i}=1}^{\mathrm{n}} \mathrm{x_i})}$$

سرانجام،

$$\mathbb{E}[\Lambda|\mathcal{D}] = \int_0^{\infty} \lambda \mathrm{p}(\lambda|\mathcal{D})\mathrm{d}\lambda$$

$$= \frac{k + \sum_{i=1}^{n} x_i}{\mathrm{n} + \frac{1}{\theta}} = 5.14$$

که تقریباً همان راه حلی است که برآورد MAP در مثال ۹ یافت شد.

از مثال قبلی مشهود است که انتخاب توزیع قبلی پیامدهای مهمی در محاسبه میانگین پسین دارد. ما توزیع گاما را تصادفی انتخاب نکرده ایم. یعنی وقتی احتمال در قبلی ضرب شد، توزیع حاصل در همان کلاس توابع قبلی باقی می ماند. ما به چنین توزیع های قبلی به عنوان پیشین های مزدوج اشاره خواهیم کرد. پیشین های مزدوج نیز ریاضیات را ساده می کنند. در واقع، این دلیل اصلی توجه آنهاست. جالب اینجاست که علاوه بر توزیع پواسون، توزیع گاما مزدوج قبل از توزیع نمایی و همچنین خود توزیع گاما است.