دانشگاه صنعتی شریف
دانشکده مهندسی کامپیوتر

تمرین تئوری اول

پارسا ملکیان - ۴۰۲۱۷۱۰۷۵

یادگیری ماشین

دکتر شریفی

۱. (الف) نادرست

SVM soft-margin دقیقاً برای همین حالت طراحی شده است. وقتی داده‌ها خطی قابل تفکیک نیستند، soft-margin با استفاده از پارامتر C و متغیرهای slack (ξ_i) اجازه می‌دهد برخی نقاط داخل margin یا سمت اشتباه قرار بگیرند. هدف soft-margin دقیقاً رسیدگی به داده‌های غیرخطی یا نویزدار است.

(ب) درست

Forest Random به دلیل استفاده از bagging، در برابر نویز مقاوم‌تر است. چون از میانگین‌گیری روی چندین درخت مستقل استفاده می‌کند، نویزها خنثی می‌شوند. اما boosted trees روی اشتباهات تمرکز می‌کند و وزن داده‌های اشتباه را افزایش می‌دهد، بنابراین ممکن است روی داده‌های نویزدار overfit کند.

(پ) درست

در bagging معمولاً از درخت‌های عمیق (fully grown) استفاده می‌کنیم که واریانس بالا و بایاس کم دارند. سپس با ترکیب نتایج n درخت از طریق voting یا averaging، واریانس کل کاهش می‌یابد در حالی که بایاس تقریباً ثابت می‌ماند.

(ت) درست

استفاده از مدل بسیار پیچیده روی داده محدود منجر به overfitting می‌شود. چنین مدلی توانایی fit شدن کامل روی داده‌های train را دارد (بایاس کم) اما روی داده‌های جدید عملکرد ضعیفی دارد (واریانس بالا). این همان trade-off کلاسیک bias-variance است.

(ث) درست

طبق قضیه همگرایی پرسپترون (Perceptron Convergence Theorem)، اگر مجموعه داده به صورت خطی قابل تفکیک باشد، الگوریتم پرسپترون تضمین می‌کند که پس از تعداد محدودی iteration (حداکثر $\frac{R^2}{\gamma^2}$ که R شعاع داده‌ها و γ حاشیه است) به یک راه‌حل همگرا می‌شود.

۲. مدل چگالی kNN:

در روش kNN، برای تخمین چگالی نقطه x، کره‌ای با مرکز x و شعاع $r(x)$ در نظر می‌گیریم که دقیقاً k نقطه از داده‌ها را شامل شود. تخمین چگالی به صورت زیر است:

$$P(x) = \frac{k}{n \cdot V(x)}$$

که در آن:

- k تعداد همسایگان نزدیک

- n تعداد کل نمونه‌های آموزشی
- $V(x)$ حجم کره با شعاع $r(x)$ که k نقطه را در بر می‌گیرد

برای فضای d بعدی: $V(x) = C_d \cdot r(x)^d$ که C_d ثابت هندسی است.

اثبات نامناسب بودن:

فرض کنید کل داده‌های ما در یک ناحیه محدود $\mathcal{X}$ قرار دارند. حالا انتگرال کل چگالی را حساب می‌کنیم:

$$\int_{R^d} P(x)dx$$

این انتگرال را به دو قسمت تقسیم می‌کنیم:

$$\int_{\mathcal{X}} P(x)dx + \int_{R^d \setminus \mathcal{X}} P(x)dx$$

قسمت اول (داخل ناحیه محدود):

$$\int_{\mathcal{X}} \frac{k}{n \cdot V(x)}dx < \infty$$

این قسمت محدود است چون روی ناحیه محدود انتگرال می‌گیریم.

قسمت دوم (بیرون از ناحیه داده‌ها):

برای نقاط x که بسیار دور از داده‌ها هستند، شعاع $r(x)$ باید خیلی بزرگ شود تا k نقطه را پوشش دهد.

در واقع $r(x) \sim ||x||$ وقتی $||x|| \to \infty$.

بنابراین: $V(x) \sim C_d \cdot ||x||^d$

و چگالی: $P(x) \sim \frac{k}{n \cdot C_d \cdot ||x||^d}$

حالا انتگرال قسمت دوم:

$$\int_{R^d \setminus \mathcal{X}} \frac{k}{n \cdot C_d \cdot ||x||^d}dx$$

با تبدیل به مختصات کروی ($x = r\theta$، $dx = r^{d-1}drd\theta$):

$$\int_R^\infty \int_{S^{d-1}} \frac{k}{n \cdot C_d \cdot r^d} \cdot r^{d-1}drd\theta = \text{const} \cdot \int_R^\infty \frac{1}{r}dr = \infty$$

نتیجه: انتگرال کل واگرا است، پس $P(x)$ یک توزیع احتمال معتبر نیست.

اهمیت و نتایج:

۱. عدم نرمال‌سازی:

$$\int_{R^d} P(x)dx \neq 1$$

این یعنی نمی‌توانیم از آن به عنوان توزیع احتمال واقعی استفاده کنیم.
۲. مشکل در استنتاج بیزی:
نمی‌توانیم مستقیماً از این چگالی در فرمول بیز استفاده کنیم:

$$P(y|x) = \frac{P(x|y)P(y)}{P(x)}$$

۳. محدودیت در مقایسه مدل‌ها:
نمی‌توانیم likelihood یا log-likelihood را به درستی محاسبه کنیم.
۴. کاربرد عملی همچنان ممکن:
با وجود این مشکل، kNN برای طبقه‌بندی همچنان مفید است چون فقط به نسبت چگالی‌ها نیاز داریم:

$$\frac{P(x|y=1)}{P(x|y=0)}$$

که در آن مخرج‌های $V(x)$ ساده می‌شوند.

۳. حل سوال: اثبات نامساوی خطای Ensemble

تعاریف:

- $\hat{h}_i(x)$: مدل i-ام
- $\hat{H}_m(x) = \frac{1}{m}\sum_{i=1}^{m} \hat{h}_i(x)$: میانگین مدل‌ها
- $h(x)$: تابع واقعی

اثبات:

گام ۱: بازنویسی خطا
خطای مدل ensemble را می‌نویسیم:

$$E_x\left[\left(h(x) - \hat{H}_m(x)\right)^2\right] = E_x\left[\left(h(x) - \frac{1}{m}\sum_{i=1}^{m}\hat{h}_i(x)\right)^2\right]$$

$$= E_x\left[\left(\frac{1}{m}\sum_{i=1}^{m}\left(h(x) - \hat{h}_i(x)\right)\right)^2\right]$$

گام ۲: استفاده از نامساوی جنسن

از آنجایی که تابع $f(t) = t^2$ تابعی محدب است، از نامساوی جنسن برای میانگین‌ها داریم:

$$\left(\frac{1}{m}\sum_{i=1}^{m} a_i\right)^2 \leq \frac{1}{m}\sum_{i=1}^{m} a_i^2$$

با قرار دادن $a_i = h(x) - \hat{h}_i(x)$:

$$\left(\frac{1}{m}\sum_{i=1}^{m} \left(h(x) - \hat{h}_i(x)\right)\right)^2 \leq \frac{1}{m}\sum_{i=1}^{m} \left(h(x) - \hat{h}_i(x)\right)^2$$

گام ۳: اعمال امید ریاضی

با گرفتن امید ریاضی از دو طرف:

$$E_x\left[\left(\frac{1}{m}\sum_{i=1}^{m} \left(h(x) - \hat{h}_i(x)\right)\right)^2\right] \leq E_x\left[\frac{1}{m}\sum_{i=1}^{m} \left(h(x) - \hat{h}_i(x)\right)^2\right]$$

$$E_x\left[\left(h(x) - \hat{H}_m(x)\right)^2\right] \leq \frac{1}{m}\sum_{i=1}^{m} E_x\left[\left(h(x) - \hat{h}_i(x)\right)^2\right]$$

که همان نامساوی مورد نظر است.

مفهوم این حکم:

۱. کاهش خطا از طریق میانگین‌گیری:

خطای مدل ensemble (سمت راست) کمتر یا مساوی میانگین خطاهای تک تک مدل‌هاست. این یعنی ترکیب مدل‌ها همیشه بهتر یا حداقل به خوبی بهترین مدل فردی عمل می‌کند.

۲. کاهش واریانس:

میانگین‌گیری باعث کاهش واریانس پیش‌بینی‌ها می‌شود. اگر مدل‌ها مستقل و بی‌طرف باشند:

$$\text{error ensemble} = \frac{1}{m} \times \text{error personal of mean}$$

یعنی با افزایش m، خطا به نسبت $1/m$ کاهش می‌یابد.

۳. اصل **ensemble learning**:

این اساس روش‌هایی مثل Bagging، Random Forest و مدل‌های ensemble است. با آموزش چند مدل مختلف و میانگین‌گیری از آنها، عملکرد بهتری نسبت به تک تک مدل‌ها داریم.

۴. شرط مهم:

این کاهش خطا زمانی مؤثرتر است که مدل‌ها متنوع و خطاهایشان ناهمبسته باشد. اگر همه مدل‌ها یکسان باشند، هیچ بهبودی حاصل نمی‌شود.

۴. سوال ۴: Random Forest و نمونه‌گیری

(الف) دلایل خطای بالای Random Forest روی test:

مشکل: خطای پایین روی train و خطای بالا روی test $\rightarrow$ Overfitting

دلیل ۱: عمق زیاد درخت‌ها

در Random Forest، اگر درخت‌ها خیلی عمیق باشند، هر درخت می‌تواند داده‌های train را کاملاً حفظ کند (memorize).

راه‌حل:

- محدود کردن عمق درخت‌ها: تنظیم `max_depth`
- افزایش حداقل نمونه در هر برگ: `min_samples_leaf`
- محدود کردن تعداد نمونه برای split: `min_samples_split`

دلیل ۲: تعداد کم درخت‌ها یا کمبود تنوع

اگر تعداد درخت‌ها کم باشد یا همبستگی بین آنها زیاد باشد، قدرت ensemble کاهش می‌یابد و مدل به راحتی overfit می‌کند.

راه‌حل:

- افزایش تعداد درخت‌ها: `n_estimators`
- کاهش `max_features` برای افزایش تنوع بین درخت‌ها
- استفاده از bootstrap sampling متنوع‌تر

(ب) محاسبه میانگین و واریانس نمونه‌گیری:

داده شده:

- داده‌های اصلی: $X = \{x_i\}_{i=1}^{n}$ با $E[x_i] = \mu$ و $\text{Var}(x_i) = \sigma^2$
- نمونه‌گیری: $z_m = \frac{1}{m}\sum_{i=1}^{m} a_i$ که a_i به صورت تصادفی و با جایگذاری از X انتخاب می‌شوند

محاسبه میانگین z_m:

$$E[z_m] = E\left[\frac{1}{m}\sum_{i=1}^{m} a_i\right] = \frac{1}{m}\sum_{i=1}^{m} E[a_i] = \frac{1}{m} \cdot m\mu = \mu$$

نتیجه: $E[z_m] = \mu$ (نمونه بی‌طرف است)

محاسبه واریانس z_m:

چون a_i ها مستقل هستند (نمونه‌گیری با جایگذاری):

$$\text{Var}(z_m) = \text{Var}\left(\frac{1}{m}\sum_{i=1}^{m} a_i\right) = \frac{1}{m^2}\sum_{i=1}^{m} \text{Var}(a_i)$$

$$= \frac{1}{m^2} \cdot m\sigma^2 = \frac{\sigma^2}{m}$$

نتیجه: $\text{Var}(z_m) = \frac{\sigma^2}{m}$

مقایسه $\{z_m\}$ با X:

خوبی‌ها:

۱. کاهش واریانس:

$$\text{Var}(z_m) = \frac{\sigma^2}{m} < \sigma^2 = \text{Var}(x_i)$$

با افزایش m، واریانس به 0 میل می‌کند. این باعث پایداری بیشتر تخمین می‌شود.

۲. بی‌طرفی:

$$E[z_m] = \mu$$

میانگین نمونه، تخمین بی‌طرفی از میانگین جامعه است.

۳. کاهش نویز: با میانگین‌گیری، اثر نقاط پرت (outlier) کاهش می‌یابد.

بدی‌ها:

۱. از دست دادن تنوع: همه z_m ها نزدیک μ هستند، در حالی که X تنوع بیشتری دارد.

۲. کاهش اطلاعات: m نمونه اصلی را به یک عدد تبدیل می‌کنیم، اطلاعات زیادی از دست می‌رود.

۳. همبستگی در bootstrap: اگر از نمونه‌گیری با جایگذاری استفاده کنیم، برخی نمونه‌ها تکرار می‌شوند و برخی دیگر حذف می‌شوند.

شرایط استفاده مناسب:

مناسب است زمانی که:

۱. هدف تخمین میانگین است: وقتی می‌خواهیم μ را تخمین بزنیم، z_m تخمین بهتری نسبت به تک نمونه است.

۲. داده دارای نویز زیاد است: میانگین‌گیری باعث کاهش اثر نویز می‌شود.

۳. در **ensemble methods**: در Bagging و Random Forest، این نمونه‌گیری برای ایجاد تنوع بین مدل‌ها استفاده می‌شود.

۴. برای برآورد عدم قطعیت: با ایجاد چندین z_m، می‌توانیم فاصله اطمینان محاسبه کنیم.

نامناسب است زمانی که:

۱. نیاز به حفظ تنوع داده‌ها: اگر تنوع و توزیع کامل داده مهم باشد.

۲. m خیلی کم است: واریانس هنوز زیاد است و تخمین قابل اعتماد نیست.

۳. نمونه اصلی کم است (n کوچک): نمونه‌گیری مجدد ممکن است به overfitting منجر شود.

۵. (الف) مشتق تابع سیگموئید و مشکل gradient vanishing

تابع سیگموئید:

$$\sigma(z) = \frac{1}{1 + e^{-z}}$$

محاسبه مشتق (گام به گام):

گام ۱: از قاعده کسر استفاده می‌کنیم:

$$\frac{d\sigma}{dz} = \frac{d}{dz}\left(\frac{1}{1 + e^{-z}}\right)$$

گام ۲: فرض کنیم $u = 1 + e^{-z}$، پس $\sigma = \frac{1}{u}$:

$$\frac{d\sigma}{dz} = -\frac{1}{u^2} \cdot \frac{du}{dz}$$

گام ۳: محاسبه $\frac{du}{dz}$:

$$\frac{du}{dz} = \frac{d}{dz}(1 + e^{-z}) = -e^{-z}$$

گام ۴: ترکیب:

$$\frac{d\sigma}{dz} = -\frac{1}{(1 + e^{-z})^2} \cdot (-e^{-z}) = \frac{e^{-z}}{(1 + e^{-z})^2}$$

گام ۵: ساده‌سازی با استفاده از $\sigma(z) = \frac{1}{1+e^{-z}}$:

$$\frac{e^{-z}}{(1 + e^{-z})^2} = \frac{1}{1 + e^{-z}} \cdot \frac{e^{-z}}{1 + e^{-z}}$$

$$= \frac{1}{1 + e^{-z}} \cdot \frac{1 + e^{-z} - 1}{1 + e^{-z}} = \sigma(z) \cdot (1 - \sigma(z))$$

نتیجه نهایی:

$$\boxed{\frac{d\sigma(z)}{dz} = \sigma(z)(1 - \sigma(z))}$$

رسم نمودارها و تحلیل:

نمودار تابع سیگموئید:

- برای $z \to -\infty$: $\sigma(z) \to 0$
- برای $z = 0$: $\sigma(0) = 0.5$
- برای $z \to +\infty$: $\sigma(z) \to 1$

نمودار مشتق $\sigma'(z) = \sigma(1 - \sigma)$:

- بیشترین مقدار در $z = 0$: $\sigma'(0) = 0.25$

- برای $\sigma'(z) \approx 0$:$|z| > 3$

مشکل gradient: vanishing

وقتی $|z|$ بزرگ باشد (مثلاً $z > 5$ یا $z < -5$):

$$\sigma'(z) \approx 0$$

در بروزرسانی وزن‌ها با gradient: descent

$$\theta_{new} = \theta_{old} - \alpha \frac{\partial L}{\partial \theta}$$

که:

$$\frac{\partial L}{\partial \theta} = \frac{\partial L}{\partial \sigma} \cdot \frac{\partial \sigma}{\partial z} \cdot \frac{\partial z}{\partial \theta}$$

مشکل: وقتی $\sigma'(z) \approx 0$، گرادیان خیلی کوچک می‌شود ⇒ بروزرسانی وزن‌ها خیلی کند ⇒ یادگیری متوقف می‌شود.

این مشکل در لایه‌های اولیه شبکه‌های عمیق شدیدتر است gradient) به صورت ضربی کاهش می‌یابد).

(ب) همگرایی کند رگرسیون لجستیک

تابع هزینه رگرسیون لجستیک:

$$L(\theta) = -\frac{1}{n} \sum_{i=1}^{n} [y_i \log(\sigma(z_i)) + (1 - y_i) \log(1 - \sigma(z_i))]$$

که $z_i = \theta^T x_i$.

دلیل همگرایی کند برای داده‌های خطی قابل تفکیک:

وقتی داده‌ها کاملاً قابل تفکیک هستند، مدل تلاش می‌کند احتمال $P(y = 1|x)$ را برای کلاس مثبت به 1 و برای کلاس منفی به 0 برساند.

برای این کار، $|\theta^T x|$ باید به ∞ برود:

$$\theta^T x \to +\infty \Rightarrow \sigma(\theta^T x) \to 1$$

$$\theta^T x \to -\infty \Rightarrow \sigma(\theta^T x) \to 0$$

مشکل:

هیچ مقدار محدودی از θ نمی‌تواند σ را دقیقاً به 0 یا 1 برساند، پس الگوریتم همیشه تلاش می‌کند $||\theta||$ را افزایش دهد.

اما وقتی $|\theta^T x|$ خیلی بزرگ شود:

$$\sigma'(\theta^T x) \approx 0$$

گرادیان خیلی کوچک می‌شود:

$$\nabla_\theta L = \frac{1}{n}\sum_{i=1}^{n}(\sigma(z_i) - y_i)x_i \approx 0$$

نتیجه: بروزرسانی‌ها خیلی کوچک ⇒ همگرایی خیلی کند ⇒ $||\theta|| \to \infty$ به آرامی.
راه‌حل: استفاده از regularization (مثل L_2) که $||\theta||$ را محدود می‌کند.

(ج) invariance در طبقه‌بندی چندکلاسه

مدل :softmax

$$P(y = k|x) = \frac{e^{\theta_k^T x}}{\sum_{j=1}^{K} e^{\theta_j^T x}}$$

هدف: نشان دهیم اگر به همه θ_k یک بردار ثابت c اضافه کنیم، احتمالات تغییر نمی‌کنند.
اثبات:
فرض کنیم $\theta'_k = \theta_k + c$ برای همه $k = 1, \ldots, K$.
احتمال جدید:

$$P'(y = k|x) = \frac{e^{(\theta_k+c)^T x}}{\sum_{j=1}^{K} e^{(\theta_j+c)^T x}}$$

$$= \frac{e^{\theta_k^T x + c^T x}}{\sum_{j=1}^{K} e^{\theta_j^T x + c^T x}}$$

$$= \frac{e^{\theta_k^T x} \cdot e^{c^T x}}{\sum_{j=1}^{K} e^{\theta_j^T x} \cdot e^{c^T x}}$$

$$= \frac{e^{\theta_k^T x} \cdot e^{c^T x}}{e^{c^T x} \cdot \sum_{j=1}^{K} e^{\theta_j^T x}}$$

$$= \frac{e^{\theta_k^T x}}{\sum_{j=1}^{K} e^{\theta_j^T x}} = P(y = k|x)$$

نتیجه: $P'(y = k|x) = P(y = k|x)$ در نتیجه softmax فقط به تفاوت نسبی بین θ_k ها وابسته است، نه مقادیر مطلق آنها.
این یعنی پارامترها unique نیستند و بی‌نهایت راه‌حل وجود دارد که همه یک پیش‌بینی می‌دهند.
راه‌حل عملی: معمولاً $\theta_K = 0$ قرار می‌دهیم (یک کلاس را reference می‌گیریم).

۶. (الف) Geometric Margin در SVM

بخش ۱: چرا geometric margin برابر است با $\gamma = \min_i \frac{y_i(w^T x_i + b)}{\|w\|}$؟

فاصله یک نقطه x_i از ابرصفحه $w^T x + b = 0$ به صورت زیر محاسبه می‌شود:

$$d_i = \frac{|w^T x_i + b|}{\|w\|}$$

برای طبقه‌بندی صحیح، می‌خواهیم $y_i(w^T x_i + b) > 0$ (نقطه در سمت درست باشد).

فاصله signed (با علامت):

$$d_i^{\text{signed}} = \frac{y_i(w^T x_i + b)}{\|w\|}$$

Geometric margin: کمترین فاصله از ابرصفحه تا نزدیک‌ترین نقطه:

$$\gamma = \min_i d_i^{\text{signed}} = \min_i \frac{y_i(w^T x_i + b)}{\|w\|}$$

توضیح:

- صورت کسر $y_i(w^T x_i + b)$ فاصله functional است
- مخرج $\|w\|$ برای نرمال‌سازی و تبدیل به فاصله واقعی هندسی
- حداقل روی همه نقاط $\Rightarrow$ نزدیک‌ترین نقطه به مرز

بخش ۲: چرا کاهش $\frac{1}{2}\|w\|^2$ باعث افزایش margin می‌شود؟

از شرط مسئله: $y_i(w^T x_i + b) \geq 1$ برای همه i.

پس:

$$\gamma = \min_i \frac{y_i(w^T x_i + b)}{\|w\|} \geq \frac{1}{\|w\|}$$

رابطه معکوس:

$$\gamma \geq \frac{1}{\|w\|} \quad \Rightarrow \quad \gamma \text{افزایش} \Rightarrow \|w\| \text{کاهش}$$

استدلال هندسی:

مسئله بهینه‌سازی:

$$\min_{w,b} \frac{1}{2}\|w\|^2 \quad \text{s.t.} \quad y_i(w^T x_i + b) \geq 1$$

شرط $y_i(w^T x_i + b) \geq 1$ یعنی functional margin حداقل 1 است.

چون $\gamma = \frac{\text{functional margin}}{\|w\|}$، با ثابت نگه داشتن صورت کسر (functional margin ≥ 1) و کاهش دادن مخرج ($\|w\|$)، margin افزایش می‌یابد:

$$\gamma \text{بیشینه‌سازی} \quad \Leftrightarrow \quad \frac{1}{\|w\|} \text{بیشینه‌سازی} \quad \Leftrightarrow \quad \|w\| \text{کمینه‌سازی}$$

نتیجه: با کمینه کردن $\frac{1}{2}\|w\|^2$ (که معادل کمینه کردن $\|w\|$ است)، geometric margin را بیشینه می‌کنیم.

(ب) تخمین‌گر Least Squares

مدل:

$$y = X\beta + \epsilon, \quad \epsilon \sim N(0, \sigma^2 I)$$

که $y \in R^n$، $X \in R^{n \times p}$، $\beta \in R^p$.

بخش ۱: فرم بسته تخمین‌گر Least Squares

تابع هزینه:

$$L(\beta) = \|y - X\beta\|^2 = (y - X\beta)^T (y - X\beta)$$

بازکردن:

$$L(\beta) = y^T y - 2\beta^T X^T y + \beta^T X^T X \beta$$

گرفتن مشتق نسبت به β:

$$\frac{\partial L}{\partial \beta} = -2X^T y + 2X^T X \beta$$

صفر قرار دادن برای یافتن نقطه بهینه:

$$-2X^T y + 2X^T X \beta = 0$$

$$X^T X \beta = X^T y$$

فرم بسته (اگر $X^T X$ معکوس‌پذیر باشد):

$$\boxed{\hat{\beta} = (X^T X)^{-1} X^T y}$$

بخش ۲: اثبات unbiased بودن

امید ریاضی از $\hat{\beta}$:

$$E[\hat{\beta}] = E[(X^T X)^{-1} X^T y]$$

جایگذاری $y = X\beta + \epsilon$:

$$E[\hat{\beta}] = E[(X^T X)^{-1} X^T (X\beta + \epsilon)]$$

$$= E[(X^T X)^{-1} X^T X \beta + (X^T X)^{-1} X^T \epsilon]$$

$$= E[\beta + (X^T X)^{-1} X^T \epsilon]$$

چون X deterministic است و $E[\epsilon] = 0$:

$$E[\hat{\beta}] = \beta + (X^T X)^{-1} X^T E[\epsilon] = \beta + 0 = \beta$$

$\boxed{E[\hat{\beta}] = \beta \quad \Rightarrow \quad \text{تخمین‌گر unbiased است}}$

بخش ۳: ماتریس Covariance

$$\text{Cov}(\hat{\beta}) = E[(\hat{\beta} - E[\hat{\beta}])(\hat{\beta} - E[\hat{\beta}])^T]$$

چون $E[\hat{\beta}] = \beta$:

$$\text{Cov}(\hat{\beta}) = E[(\hat{\beta} - \beta)(\hat{\beta} - \beta)^T]$$

محاسبه $\hat{\beta} - \beta$:

$$\hat{\beta} - \beta = (X^T X)^{-1} X^T y - \beta$$

$$= (X^T X)^{-1} X^T (X\beta + \epsilon) - \beta$$

$$= \beta + (X^T X)^{-1} X^T \epsilon - \beta = (X^T X)^{-1} X^T \epsilon$$

پس:

$$\text{Cov}(\hat{\beta}) = E[(X^T X)^{-1} X^T \epsilon \epsilon^T X (X^T X)^{-1}]$$

چون X ثابت است:

$$= (X^T X)^{-1} X^T E[\epsilon \epsilon^T] X (X^T X)^{-1}$$

و $E[\epsilon \epsilon^T] = \text{Cov}(\epsilon) = \sigma^2 I$:

$$= (X^T X)^{-1} X^T (\sigma^2 I) X (X^T X)^{-1}$$

$$= \sigma^2 (X^T X)^{-1} X^T X (X^T X)^{-1}$$

$$= \sigma^2 (X^T X)^{-1}$$

$$\boxed{\text{Cov}(\hat{\beta}) = \sigma^2 (X^T X)^{-1}}$$

نتیجه: واریانس تخمین‌گر به σ^2 (واریانس نویز) و $(X^T X)^{-1}$ (هندسه داده‌ها) بستگی دارد.

۷. (الف) اثبات برای نمونه اول

هدف: نشان دهیم:

$$E[e_1(S_n)] = E\left[(y - \hat{f}_{S_{n-1}}(x))^2\right]$$

که در آن (x, y) یک نمونه تصادفی از توزیع داده است.

اثبات:

خطای نمونه اول:

$$e_1(S_n) = (y_1 - \hat{f}_{S_{n-1}^{-1}}(x_1))^2$$

امید ریاضی از e_1:

$$E[e_1(S_n)] = E\left[(y_1 - \hat{f}_{S_{n-1}^{-1}}(x_1))^2\right]$$

نکته کلیدی: امید ریاضی روی همه نمونه‌های تصادفی ممکن S_n است.
برای محاسبه این امید، می‌توانیم آن را به صورت زیر بنویسیم:

$$E[e_1(S_n)] = E_{S_n}\left[(y_1 - \hat{f}_{S_{n-1}^{-1}}(x_1))^2\right]$$

این را می‌توان به صورت امید مشروط نوشت:

$$= E_{(x_1,y_1)}\left[E_{S_{n-1}^{-1}|(x_1,y_1)}\left[(y_1 - \hat{f}_{S_{n-1}^{-1}}(x_1))^2\right]\right]$$

اما چون $\hat{f}_{S_{n-1}^{-1}}$ تابعی از S_{n-1}^{-1} است (که شامل (x_1, y_1) نیست):

$$= E_{(x_1,y_1)}\left[E_{S_{n-1}^{-1}}\left[(y_1 - \hat{f}_{S_{n-1}^{-1}}(x_1))^2 \middle| (x_1, y_1)\right]\right]$$

چون (x_1, y_1) و S_{n-1}^{-1} مستقل هستند:

$$= E_{(x_1,y_1),S_{n-1}^{-1}}\left[(y_1 - \hat{f}_{S_{n-1}^{-1}}(x_1))^2\right]$$

حالا (x_1, y_1) یک نمونه تصادفی از توزیع است و S_{n-1}^{-1} مجموعه‌ای از $n-1$ نمونه مستقل دیگر است.
اگر (x, y) را یک نمونه تصادفی جدید از همان توزیع و S_{n-1} را مجموعه‌ای از $n-1$ نمونه بنامیم، داریم:

$$E[e_1(S_n)] = E_{(x,y),S_{n-1}}\left[(y - \hat{f}_{S_{n-1}}(x))^2\right]$$

$$\boxed{E[e_1(S_n)] = E\left[(y - \hat{f}_{S_{n-1}}(x))^2\right]}$$

تفسیر: خطای مورد انتظار برای نمونه اول برابر است با خطای مورد انتظار برای یک نمونه تصادفی جدید (x, y) وقتی مدل روی $n-1$ نمونه آموزش دیده است.

(ب) تعمیم به LOOCV کامل

هدف: نشان دهیم:

$$E[e(S_n)] = E\left[(y - \hat{f}_{S_{n-1}}(x))^2\right]$$

اثبات:
خطای LOOCV:

$$e(S_n) = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{f}_{S_{n-1}^{-i}}(x_i))^2 = \frac{1}{n}\sum_{i=1}^{n} e_i(S_n)$$

امید ریاضی:

$$E[e(S_n)] = E\left[\frac{1}{n}\sum_{i=1}^{n} e_i(S_n)\right]$$

با استفاده از خطی بودن امید ریاضی:

$$= \frac{1}{n}\sum_{i=1}^{n} E[e_i(S_n)]$$

نکته کلیدی: به دلیل تقارن ،(symmetry) همه نمونه‌ها از همان توزیع هستند و نقش یکسانی دارند:

$$E[e_1(S_n)] = E[e_2(S_n)] = \cdots = E[e_n(S_n)]$$

بنابراین:

$$E[e(S_n)] = \frac{1}{n}\sum_{i=1}^{n} E[e_i(S_n)] = \frac{1}{n} \cdot n \cdot E[e_1(S_n)]$$

$$= E[e_1(S_n)]$$

از نتیجه قسمت (الف):

$$E[e(S_n)] = E\left[(y - \hat{f}_{S_{n-1}}(x))^2\right]$$

$$\boxed{E[e(S_n)] = E\left[(y - \hat{f}_{S_{n-1}}(x))^2\right]}$$

تفسیر و اهمیت:

۱. تخمین بی‌طرف خطای تعمیم:

LOOCV یک تخمین‌گر (تقریباً) بی‌طرف از خطای تعمیم است. خطای مورد انتظار آن برابر است با خطای واقعی مدل آموزش دیده روی $n-1$ نمونه.

۲. مقایسه با error: test

اگر مدل روی n نمونه آموزش ببیند، خطای test برابر است با:

$$\text{Error Test} = E[(y - \hat{f}_{S_n}(x))^2]$$

خطای LOOCV کمی بدبینانه‌تر است (چون فقط $n-1$ نمونه استفاده می‌کند).

۳. واریانس بالا:

با اینکه LOOCV تخمین‌گر بی‌طرف است، اما واریانس بالایی دارد چون مدل‌های $\hat{f}_{S_{n-1}^{-i}}$ همبستگی زیادی با هم دارند (تقریباً روی همان داده‌ها آموزش دیده‌اند).

۴. cross-validation: k-fold

به همین دلیل معمولاً از k-fold CV (مثلاً $k=5$ یا $k=10$) استفاده می‌شود که trade-off بهتری بین bias و variance دارد.

۸. سوال ۸: درخت تصمیم و Bootstrap

داده‌ها:

دما	مزه	اندازه	خوشایندی
گرم	شور	کوچک	خیر
سرد	شیرین	بزرگ	خیر
سرد	شیرین	بزرگ	خیر
سرد	ترش	کوچک	بله
گرم	ترش	کوچک	بله
گرم	شور	بزرگ	خیر
گرم	ترش	بزرگ	بله
سرد	شیرین	کوچک	بله
سرد	شیرین	کوچک	بله
گرم	شور	بزرگ	خیر

تعداد کل: $n = 10$، بله: 5، خیر: 5

(الف) آنتروپی اولیه

$$H(S) = -\sum_i p_i \log_2(p_i)$$

$$H(S) = -\frac{5}{10}\log_2\left(\frac{5}{10}\right) - \frac{5}{10}\log_2\left(\frac{5}{10}\right)$$

$$= -0.5 \times (-1) - 0.5 \times (-1) = 0.5 + 0.5$$

$$\boxed{H(S) = 1}$$

(ب) Gain Information برای ویژگی مزه

تقسیم بر اساس مزه:
مزه = شور: نمونه‌های $\{1, 6, 10\} \Rightarrow$ همه خیر (۳ نمونه)

$$H(\text{salty}) = 0$$

مزه = شیرین: نمونه‌های $\{2, 3, 8, 9\} \Rightarrow$ ۳ بله، ۱ خیر (۴ نمونه)

$$H(\text{sweet}) = -\frac{3}{4}\log_2\left(\frac{3}{4}\right) - \frac{1}{4}\log_2\left(\frac{1}{4}\right)$$

$$= -0.75 \times (-0.415) - 0.25 \times (-2) = 0.311 + 0.5 = 0.811$$

مزه = ترش: نمونه‌های $\{4,5,7\} \Rightarrow$ همه بله (۳ نمونه)

$$H(\text{sour}) = 0$$

آنتروپی شرطی:

$$H(S|\text{taste}) = \frac{3}{10} \times 0 + \frac{4}{10} \times 0.811 + \frac{3}{10} \times 0$$
$$= 0.324$$

Gain: Information

$$\boxed{IG(\text{taste}) = H(S) - H(S|\text{taste}) = 1 - 0.324 = 0.676}$$

(پ) ساخت درخت تصمیم کامل

گام ۱: انتخاب ریشه

باید Gain Information برای همه ویژگی‌ها محاسبه کنیم:

دما: – گرم: $\{1,5,6,7,10\} \Rightarrow$ ۲ بله، ۳ خیر $\Rightarrow H = 0.971$ – سرد: $\{2,3,4,8,9\} \Rightarrow$ ۳ بله، ۲ خیر $\Rightarrow H = 0.971$

$$H(S|\text{temperature}) = \frac{5}{10} \times 0.971 + \frac{5}{10} \times 0.971 = 0.971$$
$$IG(\text{temperature}) = 1 - 0.971 = 0.029$$

اندازه: – کوچک: $\{1,4,5,8,9\} \Rightarrow$ ۴ بله، ۱ خیر $\Rightarrow H = 0.722$ – بزرگ: $\{2,3,6,7,10\} \Rightarrow$ ۱ بله، ۴ خیر $\Rightarrow H = 0.722$

$$H(S|\text{size}) = \frac{5}{10} \times 0.722 + \frac{5}{10} \times 0.722 = 0.722$$
$$IG(\text{size}) = 1 - 0.722 = 0.278$$

مزه: $IG = 0.676$ (از قسمت قبل)

نتیجه: بیشترین IG مربوط به مزه است $\Rightarrow$ ریشه = مزه

گام ۲: شاخه شور (pure) همه خیر $\Rightarrow$ برگ = خیر

گام ۳: شاخه ترش (pure) همه بله $\Rightarrow$ برگ = بله

گام ۴: شاخه شیرین نمونه‌های $\{2,3,8,9\}$: ۳ بله، ۱ خیر

محاسبه IG برای ویژگی‌های باقیمانده:

دما در شیرین: – سرد: $\{2,3,8,9\}$ همه سرد $\Rightarrow$ ۳ بله، ۱ خیر

اندازه در شیرین: – کوچک: $\{8,9\} \Rightarrow$ همه بله $\Rightarrow H = 0$ – بزرگ: $\{2,3\} \Rightarrow$ همه خیر $\Rightarrow H = 0$

$$H(S|\text{size}) = \frac{2}{4} \times 0 + \frac{2}{4} \times 0 = 0$$

$$IG(\text{size}) = 0.811 - 0 = 0.811$$

اندازه بهترین split است.
درخت نهایی:

$$\text{taste} \begin{cases} \text{salty} \to \text{NO} \\ \text{sour} \to \text{YES} \\ \text{sweet} \to \text{size} \begin{cases} \text{small} \to \text{YES} \\ \text{big} \to \text{NO} \end{cases} \end{cases}$$

(ت) پیش‌بینی برای غذای شیرین، کوچک، سرد

مسیر در درخت:

$$\text{sweet} = \text{taste} \to \text{small} = \text{size} \to \boxed{\text{YES}}$$

(ث) نمونه Bootstrap

نمونه‌گیری :Bootstrap با جایگذاری ۱۰ نمونه از داده اصلی انتخاب می‌کنیم.
فرض کنید نمونه bootstrap ما: $\{1, 1, 3, 4, 5, 6, 7, 8, 9, 10\}$ است.
(نمونه ۱ دوبار، نمونه ۲ حذف شده)
داده جدید:
- بله: ۵، خیر: ۵
- $H(S) = 1$

محاسبات مشابه قبل انجام می‌شود و درخت ساخته می‌شود.

نکته: به دلیل تکرار برخی نمونه‌ها و حذف برخی دیگر، ممکن است درخت متفاوتی به دست آید که تنوع را در Random Forest ایجاد می‌کند.

در این مثال خاص، چون نمونه ۲ و ۳ مشابه بودند، احتمالاً درخت مشابهی ساخته می‌شود، اما در حالت کلی تفاوت‌هایی خواهد داشت.