بسمه تعالی

پاسخ سری ششم تمرینات درس یادگیری ماشین

امیرحسین رمضانی بناب (۹۹۲۱۰۲۹۴)

۱ سوال ۱

۱.۱ آ

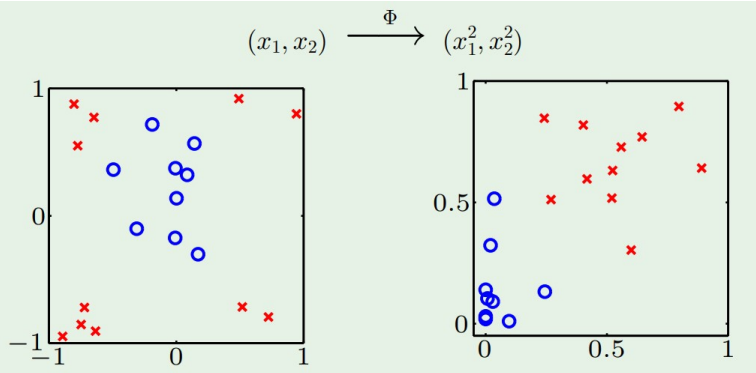
۱. اصل *Occam's Razor* : این اصل بیان می‌کند که ساده‌ترین مدل برای مدل کردن یک مجموعه‌ی داده، محتمل‌ترین مدل است. یعنی مدل‌هایی که $\mathcal{H}$ ساده‌تری دارند، دقت‌های *Out Of Sample* مناسبی خواهند داشت. شهود این اصل هم به این صورت است که مدل‌های ساده‌تر به دلیل اینکه تعداد کمتری نسبت به مدل‌های پیچیده‌تر دارند، احتمال کمتری دارد که به یک داده فیت شوند. چون فیت شدن به داده سخت‌تر است، پس موقعی که یک مدل ساده به یک مجموعه‌ی داده فیت می‌شود ارزش بالاتری دارد و نسبت به مدل پیچیده ارجح است.

۲. اصل *Sampling Bias* : طبق این اصل، داده ها باید به شکل عادلانه‌ای جمع‌آوری شده باشند و نمونه‌برداری آن‌ها به درستی انجام شده باشد. به عنوان مثال اگر جمع‌آوری داده‌ها به شکل *biased* انجام شده باشد، موقع عملکرد مدل در دنیای واقعی، داده ها از توزیع دیگری خواهند بود که با توزیع داده‌های آموزشی تفاوت دارد. این تفاوت منجر به شکست مدل در دنیای واقعی مسئله می‌شود.

۳. اصل *Data Snooping* : طبق این اصل، اگر دیتاست به هر نحوی روی هر مرحله‌ای از فرایند یادگیری تاثیر گذاشته باشد، ارزیابی خروجی الگوریتم روی این دیتاست باعث تولید دقت‌های ناعادلانه می‌گردد. زیرا تاثیر پذیرفتن بدون حسابرسی بخش‌هایی از فرایند یادگیری از روی داده به این معناست که بخشی از فضای فرضیه‌ی مسئله با دیدن این داده‌ها کنار گذاشته می‌شود و در نهایت مدل دقیقی به دست می‌آید که فقط روی داده‌های آموزشی خوب عمل می‌کند. در حالی که یک مدل مناسب باید بتواند قابلیت تعمیم‌پذیری داشته باشد.

۲.۱ ب

خیر. فرض کنید تبدیلی انجام داده‌ایم که داده‌ها را از فضای $\mathcal{X}$ به فضای $\mathcal{Z}$ برده است. ممکن است داده‌ها در فضای $\mathcal{X}$ دارای مزر تصمیم‌گیری بسیار پیچیده‌ای با یک الگوریتم خطی باشند ولی در فضای $\mathcal{Z}$ از هم خطی‌جداپذیر باشند. به عنوان مثال شکل زیر که از اسلاید درس وام گرفته شده است را آورده‌ام.



همانطور که مشخص است مرز تصمیم‌گیری تصویر سمت چپ یک مرز پیچیده خواهد بود. در حالی که با تصویر کردن داده‌ها در فضای جدید به یک مرز ساده‌ی خطی خواهیم رسید. پس اینکه یک مرز پیچیده داشته باشیم لزوما به این معنا نیست که اصل *Occam's Razer* نقض می‌شود.

۳.۱ پ

(ایده گرفته شده از کتاب آقای ابومصطفی)

از آنجایی که *Sampling Bias* مربوط به جمع‌آوری داده و *Data Snooping* مربوط به انتخاب مدل یادگیری است، این دو مفهوم با هم تفاوت دارند و به فازهای متفاوتی از آموزش ارتباط دارند. ولی مواقعی پیش می‌آید که بین این دو مفهوم، ارتباط برقرار می‌شود. مثلا فرض کنید یک بانک فقط با یک مجموعه از شرکت‌ها سر و کار دارد که هر سال به هر یک از این شرکت‌ها میزان مشخصی وام تخصیص می‌دهد. بانک می‌خواهد برای برنامه‌ریزی میزان وام‌های سال جاری، از یادگیری ماشین کمک بگیرد. بدین منظور داده‌های ۲۰ سال اخیر شرکت‌هایی که در حال حاضر با بانک کار می‌کنند را در نظر می‌گیرد و میزان وامی که به آن شرکت‌ها تخصیص خواهد داد را پیش‌بینی می‌کند. اما این الگوریتم به احتمال قوی دقت مناسبی نخواهد داشت. زیرا شرکت‌هایی که بانک در ۲۰ سال اخیر با آن‌ها قطع رابطه کرده‌است در نظر گرفته نشده‌اند و تنها شرکت‌هایی لحاظ شده‌اند که در ۲۰ سال اخیر با بانک در ارتباط بوده‌اند. این مشکل همان *Sample Bias* است. اگر به ریشه‌ی آن توجه کنیم، متوجه می‌شویم که نباید داده‌های بقیه‌ی شرکت‌ها را حذف می‌کردیم و با گرفتن زیرمجموعه‌ای از داده شرکت‌ها از روی تاریخچه‌ی آن‌ها به نوعی دچار مشکل *Data Snooping* شده‌ایم.
پس در مواقعی *Data Snooping* می‌تواند منجر به *Sample Bias* شود.

۲ سوال ۲

ابتدا مسئله‌ی *Locally Weighted Linear Regression* را بیان می‌کنیم. تنها تفاوت این مسئله با مسئله‌ی رگرسیون خطی ساده این است که معیار خطای استفاده شده در آن به شکل زیر است

$$J(\boldsymbol{\theta}) = \frac{1}{2}\sum_{n=1}^{N} w_n(\boldsymbol{\theta}^T\mathbf{x}_n - y_n)^2 = \frac{1}{2}(\mathbf{X}\boldsymbol{\theta} - Y)^\intercal\mathbf{W}(\mathbf{X}\boldsymbol{\theta} - Y)$$

که $\mathbf{w}_n$ یک عدد نامنفی است که میزان تاثیری که برچسب داده‌ایی که می‌خواهیم پیش‌بینی کنیم از برچسب داده‌ی آموزشی nام میپذیرد را نشان می‌دهد. $\mathbf{W}$ ماترس قطری $n \times n$ است که

$$\mathbf{W}[i,i] = \mathbf{w}_i$$

همچنین چون ماتریس قطری است پس

$$\mathbf{W}^\intercal = \mathbf{W}$$

۱.۲ آ

برای به دست آوردن $\boldsymbol{\theta}$ بهینه، از J نسبت به $\boldsymbol{\theta}$ مشتق می‌گیریم و برابر صفر قرار می‌دهیم.

$$\begin{aligned}
\frac{\partial J}{\partial \boldsymbol{\theta}} &= \frac{1}{2}\frac{\partial}{\partial \boldsymbol{\theta}}(\mathbf{X}\boldsymbol{\theta} - Y)^\intercal\mathbf{W}(\mathbf{X}\boldsymbol{\theta} - Y) \\
&= \frac{1}{2}\frac{\partial}{\partial \boldsymbol{\theta}}[(\boldsymbol{\theta}^\intercal\mathbf{X}^\intercal - Y^\intercal)\mathbf{W}(\mathbf{X}\boldsymbol{\theta} - Y)] \\
&= \frac{1}{2}\frac{\partial}{\partial \boldsymbol{\theta}}[\boldsymbol{\theta}^\intercal\mathbf{X}^\intercal\mathbf{W}\mathbf{X}\boldsymbol{\theta} - \boldsymbol{\theta}^\intercal\mathbf{X}^\intercal\mathbf{W}Y - Y^\intercal\mathbf{W}\mathbf{X}\boldsymbol{\theta} + Y^\intercal\mathbf{W}Y] \\
&= \frac{1}{2}\frac{\partial}{\partial \boldsymbol{\theta}}[\boldsymbol{\theta}^\intercal\mathbf{X}^\intercal\mathbf{W}\mathbf{X}\boldsymbol{\theta} - \boldsymbol{\theta}^\intercal\mathbf{X}^\intercal\mathbf{W}Y - Y^\intercal\mathbf{W}\mathbf{X}\boldsymbol{\theta}]
\end{aligned}$$

حال اگر عبارت بالا را برابر صفر قرار دهیم داریم:

$$\begin{aligned}
&\frac{1}{2}\frac{\partial}{\partial \boldsymbol{\theta}}[\boldsymbol{\theta}^\intercal\mathbf{X}^\intercal\mathbf{W}\mathbf{X}\boldsymbol{\theta} - \boldsymbol{\theta}^\intercal\mathbf{X}^\intercal\mathbf{W}Y - Y^\intercal\mathbf{W}\mathbf{X}\boldsymbol{\theta}] = 0 \\
&\Rightarrow \frac{1}{2}[2\mathbf{X}^\intercal\mathbf{W}\mathbf{X}\boldsymbol{\theta} - \mathbf{X}^\intercal\mathbf{W}Y - \mathbf{X}^\intercal\mathbf{W}^\intercal Y] = 0 \\
&\Rightarrow \frac{1}{2}[2\mathbf{X}^\intercal\mathbf{W}\mathbf{X}\boldsymbol{\theta} - \mathbf{X}^\intercal\mathbf{W}Y - \mathbf{X}^\intercal\mathbf{W}Y] = 0 \\
&\Rightarrow \frac{1}{2}[2\mathbf{X}^\intercal\mathbf{W}\mathbf{X}\boldsymbol{\theta} - 2\mathbf{X}^\intercal\mathbf{W}Y] = 0 \\
&\Rightarrow \mathbf{X}^\intercal\mathbf{W}\mathbf{X}\boldsymbol{\theta} - \mathbf{X}^\intercal\mathbf{W}Y = 0 \\
&\Rightarrow \mathbf{X}^\intercal\mathbf{W}\mathbf{X}\boldsymbol{\theta} = \mathbf{X}^\intercal\mathbf{W}Y
\end{aligned}$$

حال برای به دست آوردن مقدار $\boldsymbol{\theta}$ دو طرف معادله‌ی بالا را [با فرض معکوس‌پذیر بودن] از سمت چپ در $(\mathbf{X}^\intercal\mathbf{W}\mathbf{X})^{-1}$ ضرب می‌کنیم.داریم

$$\boldsymbol{\theta} = (\mathbf{X}^\intercal\mathbf{W}\mathbf{X})^{-1}\mathbf{X}^\intercal\mathbf{W}Y$$

۲.۲ ب

۱. وزن‌های $\mathbf{W}$ باعث می‌شود که برای پیش‌بینی برچسب یک داده، از همه‌ی داده‌های آموزشی به شکل یکسان بهره نبریم. به این شکل که هر داده‌ی آموزشی، براساس شباهتی که با داده‌ی آزمون دارد، در انتخاب برچسب برای آن دخالت می‌کند. برای محاسبه‌ی شباهت نیز می‌توان از معیارهای متفاوتی استفاده کرد و طبق معیار انتخاب شده، شباهت داده‌ی آزمون با داده‌های آموزش در $\mathbf{W}$ قرار داد.

۲. به واریانس $\mathbf{W}$ گوسی، پهنای باند می‌گویند زیرا هر چقدر مقدار واریانس بیشتر شود، داده‌های دورتر با وزن بیشتری در انتخاب برچسب داده‌ی جدید دخیل هستند و واریانس کوچک تر باعث می‌شود که داده‌های نزدیک‌تر اهمیت بیشتری پیدا کنند و داده‌های دورتر اهمیت کمتر. پس واریانس، سرعت افت مقادیر $\mathbf{W}$ را بر حسب فاصله از داده‌ی آزمون نمایش می‌دهد. داده‌هایی که وزن بیشتری بگیرند در تعیین fit تاثیر بیشتری خواهند گذاشت.

۳. واریانس این توزیع می‌تواند مقادیر متمایزی برای هر داده‌ی آزمون داشته باشد. این متغیر کردن توزیع می‌تواند به طرق مختلف انجام شود. به عنوان مثال می‌توان ابتدا یک الگوریتم خوشه بندی روی داده‌های آموزش اجرا کرد و سپس واریانس همه‌ی داده‌های آزمونی که به یک خوشه‌ی داده‌های آموزشی تعلق می‌گیرند را یکسان در نظر گرفت و آن مقدار را به نحوی تنظیم کرد که داده‌های آزمون داخل هر خوشه، بیشترین اثر را از داده‌های آموزش داخل آن خوشه بگیرند. به این ترتیب احتمالا عملکرد الگوریتم بهبود می‌یابد.

۳ سوال ۳

۱.۳ آ

برای این بخش ابتدا الگوریتم کلی حالت *Regression* بیان می‌شود و سپس با اعمال تغییراتی برای *Classification* نیز استفاده می‌گردد.

۱. حالت *Regression* :

- فرض می‌شود ورودی مسئله دادگان آموزشی $\{(\mathbf{x}_i, \mathbf{y}_i)\}_{i=1}^{n}$ باشند. همچنین یک تابع *loss* مشتق‌پذیر $L(\mathbf{y}, f(\mathbf{x}))$ داریم.
- ابتدا یک مقدار اولیه ثابت برای پیش‌بینی انتخاب می‌کنیم. این مقدار به شکلی انتخاب می‌شود که معیار *loss* را کمینه کند.

$$F_0(\mathbf{x}) = \underset{\gamma}{argmin} \sum_{i=1}^{n} L(\mathbf{y}_i, \gamma)$$

با این کار یک نقطه‌ی شروع خوب برای الگوریتم داریم.

- حال به سراغ ساخت یادگیرنده‌های جدید می‌رویم. اگر در مرحله‌ی ساخت یادگیرنده‌ی mام باشیم این مراحل را پیش می‌بریم
  - ابتدا *residual* ها را محاسبه می‌کنیم. به این معنا که خطای الگوریتم روی داده‌های مختلف را اندازه می‌گیریم.

$$r_{im} = -\Big[\frac{\partial L(\mathbf{y}_i, F(\mathbf{x}_i))}{\partial F(\mathbf{x}_i)}\Big]_{F(\mathbf{x})=F_{m-1}(\mathbf{x})}$$

به این شکل *residual* الگوریتم در مرحله‌ی m روی داده‌ی i محاسبه می‌شود.

  - حال سعی می‌کنیم توسط یک یادگیرنده‌ی پایه، این مقادیر *residual* را یادبگیریم و با افزودن این یادگیرنده، بتوانیم در راستای کاهش خطا حرکت کنیم. نام تابع به دست آمده از این یادگیرنده را $h_m(\mathbf{x})$ می‌گذاریم.
  - در قدم بعدی سعی می‌کنیم وزنی که یادگیرنده‌ی جدید در کل مدل خواهد داشت را انتخاب کنیم .این وزن طوری انتخاب می‌شود که در راستای کاهش خطای مدل حرکت کند.

$$\gamma_m = \underset{\gamma}{argmin} \sum_{i=1}^{n} L(\mathbf{y}_i, F_{m-1}(\mathbf{x}_i) + \gamma h_m(\mathbf{x}_i))$$

  - در نهایت یادگیرنده‌ی ساخته‌شده را با وزن به دست آمده به کل مدل اضافه می‌کنیم

$$F_{m-1}(\mathbf{x}) = F_{m-1}(\mathbf{x}) + \gamma h_m(\mathbf{x})$$

۲. حالت *Classification* : در این حالت فرض کنید که M کلاس داریم. در آن‌صورت به جای یک تابع F، به تعداد M تابع خواهیم داشت که هریک از این توابع امتیاز حضور یک داده در هر کلاس را مشخص می‌کنند. یعنی توابع $F_1, F_2, ..., F_M$ را داریم که $F_i(\mathbf{x}_j)$ نشان‌دهنده‌ی امتیاز حضور داده‌ی j در کلاس i است. سپس از روی این مقادیر امتیاز، می‌توانیم توسط تابع Softmax احتمال حضور داده در هر کلاس را مشخص کرده و داده را به کلاسی تخصیص دهیم که دارای بیشترین احتمال باشد. به این شکل که برای داده‌ی $\mathbf{x}$، احتمال حضور آن در هر کلاس را به شکل زیر محاسبه می‌کنیم.

$$P_m(\mathbf{x}) = \frac{e^{F_m(\mathbf{x})}}{\sum_{c=1}^{M} e^{F_c(\mathbf{x})}} \quad , \; m = 1, ..., M$$

و برچسب داده برابر $\underset{m\in\{1,..,M\}}{argmax}\, P_m(\mathbf{x})$ خواهد بود.

مراحل ابتدایی الگوریتم به شکل زیر است.

- ابتدا هر برچسب را به یک توزیع احتمال تبدیل می‌کنیم. یعنی برای داده‌ی iام

$$Y_1(\mathbf{x}_i) = 0, Y_2(\mathbf{x}_i) = 0 \; ... \; Y_{\mathbf{y}_i}(\mathbf{x}_i) = 1 \; .... \; Y_M(\mathbf{x}_i) = 0$$

- طبق مقادیر بالا، احتمال عضویت داده در هر کلاس($P_1(\mathbf{x}_i), ..., P_M(\mathbf{x}_i)$) را طبق فرمول ارائه شده محاسبه می‌کنیم.
- اختلاف مقادیر پیش‌بینی شده و مقادیر واقعی را به عنوان *residual* در نظر می‌گیریم.

حال به تفاوت مراحل اصلی الگوریتم می‌پردازیم

- در حالت *Regression*،یک بردار F برای داده ها داشتیم که می‌خواستیم مقدار آن را بهینه کنیم. در حالت *Classification*، با یک ماتریس طرف هستیم

$$\begin{bmatrix} F_1(\mathbf{x}_1) & F_2(\mathbf{x}_1) & ... & F_M(\mathbf{x}_1) \\ F_1(\mathbf{x}_2) & F_2(\mathbf{x}_2) & ... & F_M(\mathbf{x}_2) \\ ... & ... & ... & ... \\ F_1(\mathbf{x}_n) & F_2(\mathbf{x}_n) & ... & F_M(\mathbf{x}_n) \end{bmatrix}$$

- در نتیجه برای fit کردن یک مدل روی هاresidual نیز باید از یک ماتریس برای نشان داده گرادیان‌ها استفاده کنیم.

$$\begin{bmatrix} \frac{\partial L}{\partial F_1(\mathbf{x}_1)} & \frac{\partial L}{\partial F_2(\mathbf{x}_1)} & \cdots & \frac{\partial L}{\partial F_M(\mathbf{x}_1)} \\ \frac{\partial L}{\partial F_1(\mathbf{x}_2)} & \frac{\partial L}{\partial F_2(\mathbf{x}_2)} & \cdots & \frac{\partial L}{\partial F_M(\mathbf{x}_2)} \\ \cdots & \cdots & \cdots & \cdots \\ \frac{\partial L}{\partial F_1(\mathbf{x}_n)} & \frac{\partial L}{\partial F_2(\mathbf{x}_n)} & \cdots & \frac{\partial L}{\partial F_M(\mathbf{x}_n)} \end{bmatrix}$$

- پس برای هر iteration به جای مقدار γ، به تعداد کلاس‌ها پارامتر خواهیم داشت تا همه‌ی توابع را در راستای عکس گرادیان حرکت دهیم.
- در نهایت، در پایان هر iteration تمام توابع با وزن‌های مربوط به فرضیه‌های جدید بروزرسانی خواهند شد.

$$F_i = F_i + \gamma_i h_i$$

روش‌های دیگری نیز برای فرموله‌بندی مسئله‌ی دسته‌بندی وجود دارد. روشی که در این پاسخ ارائه شد، برگرفته از این لینک است.

۲.۳ ب

در حالت دو کلاسه، به جای استفاده از *One Hot*، می‌توانیم از تابع هزینه‌ی *Cross Entropy* استفاده کنیم. اما به دلیل اینکه ورودی دوم تابع هزینه از جنس احتمال نمی‌باشد، باید تغییراتی در تابع هزینه اعمال شود. در صورتی که فرض کنیم "امتیاز" محاسبه شده تا این لحظه برای تخصیص داده‌ی i ام به کلاس ۱ برابر s_i باشد. در این صورت تابع sigmoid را روی این مقدار اعمال کرده و آن را برابر "احتمال" تخصیص داده‌ی i به کلاس ۱ در نظر می‌گیریم و با p_i نمایش می‌دهیم. پس تابع هزینه به شکل زیر می‌باشد

$$L(\mathbf{y}, s) = -\sum_{i=1}^{n} \mathbf{y}_i log(\sigma(s_i)) + (1 - \mathbf{y}_i) log(1 - \sigma(s_i))$$

به عبارتی برای داده‌ی iام تابع هزینه برابر است با

$$-[\mathbf{y}_i log(\sigma(s_i)) + (1 - \mathbf{y}_i) log(1 - \sigma(s_i))]$$

رابطه‌ی بالا را کمی ساده تر می‌کنیم.

$$\begin{aligned} -[\mathbf{y}_i log(\sigma(s_i)) + (1 - \mathbf{y}_i) log(1 - \sigma(s_i))] &= -\mathbf{y}_i log(\sigma(s_i)) - (1 - \mathbf{y}_i) log(1 - \sigma(s_i)) \\ &= -\mathbf{y}_i log(\frac{e^{s_i}}{1 + e^{s_i}}) - (1 - \mathbf{y}_i) log(\frac{1}{1 + e^{s_i}}) \\ &= -\mathbf{y}_i [log(e^{s_i}) - log(1 + e^{s_i})] - (1 - \mathbf{y}_i)[log(1) - log(1 + e^{s_i})] \\ &= -\mathbf{y}_i [s_i - log(1 + e^{s_i})] + (1 - \mathbf{y}_i)[log(1 + e^{s_i})] \\ &= -\mathbf{y}_i s_i + log(1 + e^{s_i}) \end{aligned}$$

اگر مشتق این تابع را نسبت به s_i محاسبه کنیم داریم:

$$\begin{aligned}\frac{\mathrm{d}}{\mathrm{d}s_i}[-\mathbf{y}_i s_i + log(1+e^{s_i})] &= -\mathbf{y}_i + \sigma(s_i) \\ &= -\mathbf{y}_i + p_i\end{aligned}$$

و اگر مشتق دوم این تابع را نسبت به s_i محاسبه کنیم داریم:

$$\begin{aligned}\frac{\mathrm{d}^2}{\mathrm{d}s_i^2}[-\mathbf{y}_i s_i + log(1+e^{s_i})] &= \frac{\mathrm{d}}{\mathrm{d}s_i}[-\mathbf{y}_i + \sigma(s_i)] \\ &= \frac{e^{s_i}}{(1+e^{s_i})^2} \\ &= \frac{e^{s_i}}{1+e^{s_i}} \times \frac{1}{1+e^{s_i}} \\ &= \sigma(s_i)(1-\sigma(s_i)) \\ &= p_i(1-p_i)\end{aligned}$$

حال با استفاده از بسط تیلور عبارت داده‌شده را ساده می‌کنیم.

$$\begin{aligned}L(\mathbf{y}_i, F_{m-1}(\mathbf{x}_i) + \gamma h_m(\mathbf{x}_i)) \approx L(\mathbf{y}_i, F_{m-1}(\mathbf{x}_i)) &+ \frac{\mathrm{d}}{\mathrm{d}F_{m-1}(\mathbf{x}_i)} L(\mathbf{y}_i, F_{m-1}(\mathbf{x}_i))\gamma h_m(\mathbf{x}_i) \\ &+ \frac{1}{2}\frac{\mathrm{d}^2}{\mathrm{d}F_{m-1}(\mathbf{x}_i)^2} L(\mathbf{y}_i, F_{m-1}(\mathbf{x}_i))\gamma^2 h_m(\mathbf{x}_i)^2\end{aligned}$$

پس تابعی که می‌خواهیم نسبت به γ کمینه کنیم به شکل زیر است.

$$\sum_{i=1}^{n}\left[L(\mathbf{y}_i, F_{m-1}(\mathbf{x}_i)) + \frac{\mathrm{d}}{\mathrm{d}F_{m-1}(\mathbf{x}_i)} L(\mathbf{y}_i, F_{m-1}(\mathbf{x}_i))\gamma h_m(\mathbf{x}_i) + \frac{1}{2}\frac{\mathrm{d}^2}{\mathrm{d}F_{m-1}(\mathbf{x}_i)^2} L(\mathbf{y}_i, F_{m-1}(\mathbf{x}_i))\gamma^2 h_m(\mathbf{x}_i)^2\right]$$

که اگر کمی ساده‌سازی کنیم

$$\begin{aligned}\sum_{i=1}^{n}\left[L(\mathbf{y}_i, F_{m-1}(\mathbf{x}_i))\right] &+ \gamma\sum_{i=1}^{n}\left[\frac{\mathrm{d}}{\mathrm{d}F_{m-1}(\mathbf{x}_i)} L(\mathbf{y}_i, F_{m-1}(\mathbf{x}_i))h_m(\mathbf{x}_i)\right] \\ &+ \frac{1}{2}\gamma^2\sum_{i=1}^{n}\left[\frac{\mathrm{d}^2}{\mathrm{d}F_{m-1}(\mathbf{x}_i)^2} L(\mathbf{y}_i, F_{m-1}(\mathbf{x}_i))h_m(\mathbf{x}_i)^2\right]\end{aligned}$$

پس اگر نسبت به γ مشتق بگیریم و برابر صفر قرار دهیم داریم:

$$\sum_{i=1}^{n}\left[\frac{\mathrm{d}}{\mathrm{d}F_{m-1}(\mathbf{x}_i)} L(\mathbf{y}_i, F_{m-1}(\mathbf{x}_i))h_m(\mathbf{x}_i)\right] + \gamma\sum_{i=1}^{n}\left[\frac{\mathrm{d}^2}{\mathrm{d}F_{m-1}(\mathbf{x}_i)^2} L(\mathbf{y}_i, F_{m-1}(\mathbf{x}_i))h_m(\mathbf{x}_i)^2\right] = 0$$

حال اگر مشتقات را جایگذاری کنیم داریم

$$\sum_{i=1}^{n}\left[(-\mathbf{y}_i + \sigma(F_{m-1}(\mathbf{x}_i)))h_m(\mathbf{x}_i)\right] + \gamma\sum_{i=1}^{n}\left[\sigma(F_{m-1}(\mathbf{x}_i))(1-\sigma(F_{m-1}(\mathbf{x}_i)))h_m(\mathbf{x}_i)^2\right] = 0$$

پس

$$\gamma = \frac{-\sum_{i=1}^{n}\left[(-\mathbf{y}_i + \sigma(F_{m-1}(\mathbf{x}_i)))h_m(\mathbf{x}_i)\right]}{\sum_{i=1}^{n}\left[\sigma(F_{m-1}(\mathbf{x}_i))(1-\sigma(F_{m-1}(\mathbf{x}_i)))h_m(\mathbf{x}_i)^2\right]}$$

۳.۳ پ

مراحل اجرای الگوریتم خواسته شده در قالب اسلایدی در فایل *Q3p.pdf* به پیوست تقدیم شده است.

۴.۳ ت

مراحل اجرای الگوریتم خواسته شده در قالب اسلایدی در فایل *Q3t.pdf* به پیوست تقدیم شده است.

۴ سوال ۴

نوت‌بوک خواسته شده در قالب فایل *ML2021S_HW6* به پیوست تقدیم شده است.