بسمه تعالی

مقدمه‌ای بر یادگیری ماشین

نیمسال اول ۹۸-۹۹

تمرین عملی سری دوم

مدرس: صابر صالح

● مهلت تحویل تمرین‌ها: ۱۳۹۸/۰۸/۲۱ ●

در این تمرین مجاز به استفاده از کتاب‌خانه‌های موجود در پایتون برای یادگیری ماشین نیستید.

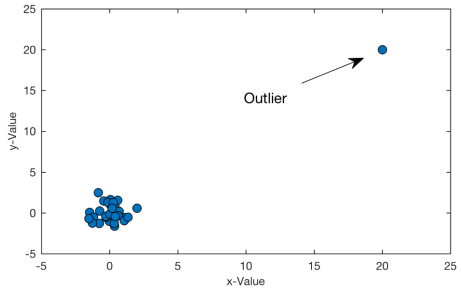
تمرین‌های برنامه‌نویسی

○ مقدمه و توضیحات

هدف از این تمرین پیاده‌سازی برخی الگوریتم‌های مهم یادگیری ماشین و آشنایی با روش‌های آماده‌سازی داده‌ها می‌باشد. در بخش اول تمرین نحوه مقابله با داده های از دست رفته و ویژگی های کیفی مورد بررسی قرار می گیرد و در بخش دوم الگوریتم های Linear regression و Logistic regression پیاده سازی خواهد شد.

▯ داده های پرت

داده‌های پرت داده‌هایی هستند که فاصله‌ی قابل توجهی از توزیع داده‌های ورودی دارند. علت وجود این داده‌ها در مسائل یادگیری ماشین ناشی از خطاهایی است که به هنگام جمع آوردی داده‌ها به وجود آمده است و گاهی می‌تواند ناشی از توزیع خود داده‌ها نیز باشد. در هر دو صورت، حذف کردن این نوع داده از دیتاست می‌تواند بر روی دقت یادگیری و همگرایی سریع‌تر تاثیر مثبتی بگذارد. در شکل زیر مثالی از آن را مشاهده می‌کنید.



▯ داده‌های از دست رفته

در دیتاست‌های گوناگون، با نمونه‌هایی روبرو خواهید شد که در یک یا چند ویژگی مقدار آن‌ها مشخص نیست. در این حالت‌ها به جای مشاهده کمیت مربوط به آن داده، عبارت nan را مشاهده می‌کنید. روش‌های مختلفی برای مقابله با این مشکل وجود دارد. یکی از این روش‌ها، حذف تمام نمونه‌ها و یا ویژگی‌هایی است که تعداد زیادی مقادیر نامشخص دارند. این روش به دلیل این‌که ممکن است بخش مهمی از داده‌ها از بین رود، چندان پیشنهاد نمی‌شود. در روش دوم، می‌توان ویژگی‌های متعلق به یک نمونه را با میانگین آن ویژگی(میانگین ستون داده‌ها) جایگذاری کرد. همچنین روش‌هایی دیگری مانند جایگذاری داده‌ها با استفاده از regression هم وجود دارد که در تمرین‌های آینده بررسی خواهند شد.

▯ گرادیان کاهشی[1] (GD)

با روش گرادیان کاهشی برای پیدا کردنِ کمینه‌ی یک تابع محدب آشنایی دارید. به طور خلاصه در این روش، تخمینِ محلِ کمینه شدنِ تابع در مرحله‌ی فعلی، به صورت زیر از تخمین مربوط به مرحله قبل محاسبه می‌شود:

$$\mathbf{x}^{(t+1)} = \mathbf{x}^{(t)} - \eta \nabla L(\mathbf{x}^{(t)})$$

که معمولاً η نرخ یادگیری نامیده می‌شود؛ به طور ویژه در بحث یادگیری ماشین. خوب است به این فکر کنید که با زیاد کردنِ η و با کم کردن آن به ترتیب چه اتفاقاتی ممکن است رخ دهد.

الگوریتم گرادیان کاهشی می‌تواند پس از تعداد معینی گام یا در صورت برقراریِ $\|\mathbf{x}^{(t+1)} - \mathbf{x}^{(t)}\|_2 \leq \delta$ در یک گام t متوقف شود. در این‌جا δ یک عدد اختیاری و اصولاً کوچک است.

---

[1] Gradient Descent

گرادیان کاهشی تصادفی[۲] (SGD)

در روش گرادیان کاهشی تصادفی به جای استفاده‌ی کامل از بردار گرادیان روش کم‌هزینه‌تری را در پیش می‌گیریم. تعداد مشخصی تکرار یا iteration انجام می‌شود که با اندیس‌های $t = 1, 2, ..., T$ نمایش می‌دهیم. با شروع از یک نقطه‌ی اولیه، در هر مرحله بروزرسانیِ تقریبِ محل بهینه به شکل زیر انجام می‌شود:

$$\mathbf{x}^{(t+1)} = \mathbf{x}^{(t)} - \eta \mathbf{v}_t,$$

که لازم است امید ریاضی بردار $\mathbf{v}_t$ برابر با بردار گرادیان در $\mathbf{x}^{(t)}$ باشد. یعنی باید: $\mathbb{E}[\mathbf{v}_t|\mathbf{x}^{(t)}] = \nabla L(\mathbf{x}^{(t)})$. در نهایت هم خروجی به صورت زیر مشخص می‌شود:

$$\bar{\mathbf{x}} = \frac{1}{T}\sum_{t=1}^{T}\mathbf{x}^{(t)}$$

برای آشنایی بیش‌تر با این روش، می‌توانید به فصل ۱۴ از کتاب درسی مراجعه کنید.

Logistic regression

در این طبقه بند، خروجی با این عبارت مشخص می‌شود $P(Y = 1|X = x_i, \omega) = \sigma(\omega \cdot x_i)$ که $\sigma(x) = \frac{1}{1+\exp(-x)}$ و y_i, x_i به ترتیب نمونه ی i ام ورودی و برچسب صفر و یا یک متناظر به آن می‌باشند. این طبقه بند در واقع توزیع شرطی برچسب خروجی بر حسب بردار ورودی را پیش‌بینی می‌کند. برای تابع هدف این الگوریتم از توابع MSE استفاده نشده و به جای آن از تابع Cross-Entropy استفاده می‌کنیم. در نهایت تابع هدف و گرادیان آن به صورت زیر بدست می‌آید. p_i همان احتمال بدست آمده برای داده‌ی x_i میباشد. توجه کنید که در رابطه‌ی Cross-Entropy فرض می‌شود توزیع q ، توزیع واقعی داده‌ها و توزیع p خروجی تابع sigmoid می‌باشد($q(Y = 1|X = x_i) = y_i$). در گزارش تمرین ذکر کنید که چرا از این تابع به جایMSE برای تابع loss استفاده می‌شود.

$$Cross - Entropy : H(q, p) = E_q[-\log p]$$

$$L(\omega) = -\frac{1}{N}\sum_{i=1}^{N}[y_i ln(p_i) + (1 - y_i)ln(1 - p_i)]$$

$$\nabla L(\omega) = -\frac{1}{N}\sum_{i=1}^{N}((p_i - y_i)x_i)$$

Linear regression

این الگوریتم، برای تخمین توابعی که خروجی پیوسته دارند، مورد استفاده قرار میگیرد. در واقع هدف تخمین تابع $f : \mathbb{R}^n \to \mathbb{R}$ به صورت $f(\mathbf{x}; \mathbf{w}) = \mathbf{w}^T\mathbf{x} + b$ است. توابع هدف آن میتواند MSE باشد که به صورت زیر تعریف می‌شود. در این تمرین ما از MSE استفاده می‌کنیم:

$$\text{MSE: } L(\mathbf{w}, b) = \frac{1}{m}\sum_{i=1}^{m} l_i(\mathbf{w}, b) = \frac{1}{m}\sum_{i=1}^{m}(y_i - \mathbf{w}^T\mathbf{x}_i - b)^2$$

یادگیری برخط (Online Learning)

امروزه بسیاری از کاربردهای پایش و کنترل سیستم‌ها وابسته به تحلیل‌های لحظه‌ای براساس روش‌های یادگیری ماشین و اتخاذ تصمیم‌های تطبیقی به صورت خودکار می‌باشند. به این ترتیب برای سادگی فرض کنید که ما یک مدلِ پیش‌بینی کننده برای یک کاربردی نیاز داریم و یک جریان پیوسته‌ای از اطلاعات به ما داده می‌شود و ما پیوسته باید بر اساس اطلاعات جدید مدل را بروزرسانی کنیم. در کل به این روش یادگیری ماشین Online Learning یا در مواردی Stream Processing گفته می‌شود که در مقابلِ Batch Processing قرار دارد. در روش Batch یک مجموعه‌ی نسبتاً بزرگ دادگان که از پیش آماده شده است به ما داده می‌شود. دیدیم Batch Processing در بعضی از کاربردهای دنیای واقعی عملاً معنایی ندارد؛ ضمن این‌که اصولاً ممکن است Online Learning در کاهش پیچیدگی محاسباتی و حافظه‌ی مورد استفاده هم مطلوب ما باشد. در سوال بخش Linear Regression مدلی که استفاده می‌کنیم مبتنی بر مسائل Online Learning است.

تمارین

۱. آماده سازی داده‌ها

دیتاست مورد نظر، متشکل از ۱۲ ویژگی و یک لیبل با نام quality است. در ابتدا باید با دو روش ارائه شده مشکل داده‌های از دست رفته را برطرف کنید. در روش اول، داده‌ها با مقادیر نامشخص را حذف و در روش دوم، با استفاده از میانگین ویژگی‌ها، آن‌ها را جایگذاری کنید. پس از آماده‌سازی داده‌ها، ویژگی‌های توصیفی را با روش دلخواه، به ویژگی‌های عددی تبدیل کنید. دقت داشته باشید که لیبل نیز مقدار توصیفی داشته و برای طبقه بندی با استفاده از Logistic regression باید آنها را به لیبل‌های باینری تبدیل کنید.

۲. طبقه بند Logistic regression

در این بخش الگوریتم Logistic regression را پیاده سازی و آن را بر روی داده‌ها اجرا می‌کنید. با استفاده از تنها دو ویژگی اول، دقت الگوریتم را گزارش کنید و داده‌ها به همراه مرز طبقه‌بندیِ به دست آمده توسط الگوریتم را برای هر دو داده‌ی حاصل از روش های ارائه شده در بخشِ "آماده سازی داده‌ها" رسم کنید. دقت کنید که تعدادی داده‌ی پرت در ورودی وجود دارد. یک‌بار الگوریتم را با حذف داده های پرت و بار دیگر با وجود آن ها اجرا کنید. نمودار ها باید به ازای گام های زمانی مختلف و نرخ های یادگیری متفاوت رسم شود.

---

[۲] Stochastic Gradient Descent

(آ) مجموعه دادگان اصلی را در نظر بگیرید. با دو روشِ مواجهه با Missing Value آشنا شدید؛ برای این بخش، باید نمونه‌های دارای Missing Value حذف شوند به طوری که ترتیب نقاط داده حفظ بشود. به این صورت که اگر در فایل دادگان اصلی نمونه‌ی a قبل از نمونه‌ی b دیده می‌شود، پس از حذف نمونه‌های دارای Missing Value هم نمونه‌ی a قبل از نمونه‌ی b دیده شود. هم‌چنین ما ویژگی A indx را به عنوان پاسخ (y) و ویژگی‌هایِ [fixed acidity,volatile acidity,citric acid] را به‌عنوان $\mathbf{x} = [\mathbf{x(1)}, \mathbf{x(2)}, \mathbf{x(3)}]$ در نظر می‌گیریم.
پس از آن‌که نمونه‌های دارایMissing Value را حذف کردید، ۳۷۰ نمونه‌ی اول از نتیجه‌ی حاصل را ذخیره کنید. طبعاً کافی است که ستون‌های مذکور یعنی [fixed acidity,volatile acidity,citric acid, A indx] را نگهداری کنید.

(ب) تابعی بنویسید که با دریافت نقاطِ داده در ورودی یعنی $(X_i, y_i), i \in [m]$ ها، نرخ یادگیری η، نقطه‌ی اولیه برای شروع بهینه‌سازی و احیاناً حداکثر تعدادِ تکرار مجاز در روند بهینه‌سازی یا δ، مسئله‌ی کمینه کردن خطای MSE را به کمک گرادیان کاهشی حل کند و جوابِ مسئله‌ی بهینه‌سازی و احیاناً موارد دیگری را در خروجی ایجاد کند. دقت کنید جزئیات ورودی و خروجی تابعی که می‌نویسید در اختیار شماست و این توضیحات صرفاً جهت تعیین چارچوب کلی می‌باشد. به همین خاطر بهتر است بخش بعدی سوال را هم بخوانید و سپس تابع را پیاده کنید.

(ج) در قسمت (آ) ۳۷۰ نقطه‌ی اول را جدا کردید. از ۳۰۰ نقطه‌ی اول این مجموعه استفاده کنید تا تخمینی از $[\mathbf{w(1)}, \mathbf{w(2)}, \mathbf{w(3)}, b]$ به دست آورید. باید از تابعِ بخش (ب) استفاده کنید. هم چنین از بین پارامترهای ورودی تابع، قیدی که وجود دارد این است که باید نقطه‌ی شروع بهینه‌سازی را به صورتِ زیر انتخاب کنید:

$$[\mathbf{w}^{(0)}(1), \mathbf{w}^{(0)}(2), \mathbf{w}^{(0)}(3), b^{(0)}] = [5, -5, 5, -5]$$

در نهایت، پس از گردن کردن تا سه رقم اعشار، باید مقدار تابع در نقطه‌ی بهینه برابر با ۰.۰۹۵ باشد. نقطه‌ی بهینه و مقدار تابع در نقطه‌ی بهینه را گزارش کنید. نرخ یادگیری مورد استفاده را نیز گزارش کنید.

در روش GD (گرادیان کاهشی) کاری که انجام می شد، بروزرسانی به صورت زیر بود:

$$(\mathbf{w}, b)^{(t+1)} = (\mathbf{w}, b)^{(t)} - \eta \frac{1}{m} \sum_{i=1}^{m} \nabla l_i((\mathbf{w}, b)^{(t)})$$

هم‌چنین با روش SGD (گرادیان کاهشی تصادفی) آشنا شدید که در آن، در هر مرحله به جای گرادیان از برداری مانند $\mathbf{v}_t$ استفاده می‌کردیم که امید ریاضی آن برابر با گرادیان باشد.

(د) نشان دهید اگر بردار $\mathbf{v}_t$ به صورت زیر انتخاب بشود که در آن r مقادیر 1 تا m را با احتمال برابر قبول می‌کند، آن‌گاه $\mathbf{v}_t$ قابل استفاده در SGD است.

$$\mathbf{v}_t = \nabla l_r((\mathbf{w}, b)^{(t)})$$

حال فرض کنید که نقاطِ ۳۰۱ تا ۳۷۰ از مجموعه‌ی حاصله در بخش (آ) به صورت پی در پی و دنبال هم وارد سیستم می‌شوند؛ مثلاً وقتی داده‌ی ۳۰۵ وارد سیستم شده است هنوز از داده‌ی ۳۱۰ خبری نداریم. هدف این است که در هر مرحله، با دیدنِ هر کدام از آن‌ها با یک روشی پیش‌بینی کننده‌ی خطی را بر اساس مشاهدات جدید بروز کنیم. در این‌جا یک فرضی انجام می‌دهیم و آن هم این‌که به تعدادِ M_{large} که عددی بسیار بزرگ است نمونه وجود داشته است و این نمونه‌ها به طور کاملاً تصادفی مرتب شده‌اند به طوری که در جریانِ داده‌ای که ما داریم نمونه‌ای که در هر جایگاه قرار می‌گیرد با احتمال کاملاً برابر (یکنواخت) یکی از M_{large} نمونه است. می‌خواهیم از ایده‌ی SGD برای بروزرسانی تخمین خود کمک بگیریم. به این ترتیب، با دیدن هر یک از داده‌های ۳۰۱ تا ۳۷۰ مانند $(\mathbf{x}, y)$ بروزرسانی را به شکل زیر انجام می‌دهیم.

$$(\mathbf{w}, b)^{(t+1)} = (\mathbf{w}, b)^{(t)} - \eta \nabla_{(\mathbf{w},b)} l((\mathbf{x}, y); (\mathbf{w}, b)^{(t)})$$

از آن‌جا که ما ۷۰ داده دیگر داریم، اندیس مراحل به صورت $t = 1, 2, ..., 70$ می‌باشد. برای نقطه‌ی شروع، $(\mathbf{w}, b)^{(0)}$ هم از نقطه‌ای که در پایان قسمت (ج) به دست آوردید استفاده کنید. در این روش پیش‌بینی کننده‌ای که در پایان مرحله t داریم به صورت زیر در نظر گرفته می‌شود:

$$(\bar{\mathbf{w}}^{(t)}, \bar{b}^{(t)}) = \frac{1}{t} \sum_{i=1}^{t} (\mathbf{w}, b)^{(i)}$$

هم‌چنین فرض کنید که به عنوان یک معیار، $\mathbf{MSE}^{(t)}$ را به صورت زیر تعریف کنیم:

$$\mathbf{MSE}^{(t)} = \frac{1}{370} \sum_{i=1}^{370} (y_i - < \bar{\mathbf{w}}^{(t)}, \mathbf{x}_i > - \bar{b}^{(t)})^2$$

(ه) با پیاده‌سازی روش یادگیریِ ذکر شده، نرخ یادگیری را به نحوی تنظیم کنید که در پایانِ کار، $\mathbf{MSE}^{(70)}$ تا حد امکان کوچک باشد. در این حالت، مقدارِ نرخ یادگیریِ یافت شده، η_a را به همراه پیش‌بینی کننده‌ی یافت شده $(\bar{\mathbf{w}}^{(70)}, \bar{b}^{(70)})$ و $\mathbf{MSE}^{(70)}$ گزارش کنید.

(و) حال فرض کنید در یک نوعِ اندکی متفاوت از این الگوریتم، نرخ یادگیری ثابت نباشد و به صورت $\eta^{(t)} = \frac{\eta^{(1)}}{\sqrt{t}}$ تغییر کند. در این حالت هم مشابه بخش قبل بهترین انتخاب برای نرخ یادگیری در $t = 1$ را که با η_b نمایش می‌دهیم به همراه $(\bar{\mathbf{w}}^{(70)}, \bar{b}^{(70)})$ و $\mathbf{MSE}^{(70)}$ گزارش کنید. هم‌چنین ذکر کنید که به نظر شما، در نظر گرفتنِ نرخ یادگیریِ متغیر به شکل فعلی، چه مزیتی می‌تواند داشته باشد؟

(ز) بزرگ‌ترین نرخ یادگیری، η_* را بیابید و گزارش کنید که برای آن هر دو روش بخش‌های قبلی همگرا شوند. فرض کنید منظور از همگرایی در این‌جا این است که برای هر دو روش، $\mathbf{MSE}^{(70)} \leq 0.15$ باشد. با این مقدار برای نرخ یادگیری، نمودارِ $\mathbf{MSE}^{(t)}$ برحسب t را برای هر دو روش ترسیم کنید. رفتارهای این دو نمودار چه تفاوتی با هم دارند؟ این موضوع را چگونه توضیح می‌دهید؟

- در فایل starter_code.ipynb مربوطه، خواسته‌های مساله به صورت گام به گام ذکر شده است. نهایتا در گزارش تمرین، تحلیل خود را از نتایج حاصله بنویسید و سوال های بخش‌های قبل را پاسخ دهید. توجه کنید که برای پیاده‌سازی این الگوریتم‌ها مجاز به استفاده از کتابخانه‌های آماده مانند scikit learn نیستید.