یادگیری ماشین — دکتر منصور رزقی

تمرین اول — دانشکده علوم ریاضی ، گروه علوم کامپیوتر، گرایش داده‌کاوی

سبحان عیوضی پور کفشدوزی — ۴۰۴۵۲۵۲۱۰۰۴

تمرین ۱:

می‌دانیم برای متغیر تصادفی Z خاصیت زیر برقرار است:

$$Var(Z) = E[(Z - E[Z])^2] \quad (۱)$$

خاصیت (۱) را بر $Var(X+Y)$ اعمال می‌کنیم:

$$Var(X+Y) = E[((X+Y) - E[X+Y])^2] \quad (۲)$$

از طرفی، طبق خاصیت خطی بودن امید ریاضی داریم:

$$E[X+Y] = E[X] + E[Y] \quad (۳)$$

با جای‌گذاری معادله (۳) در (۲) می‌توان نوشت:

$$Var(X+Y) = E[((X+Y) - (E[X] + E[Y]))^2] \quad (۴)$$

ترتیب عبارات درون امید ریاضی سمت راست معادله (۴) را بازنویسی می‌کنیم، به گونه‌ای که به جای کسر مجموع امیدریاضی تک تک متغیرها از مجموع متغیرها، مجموع تفاضل امیدریاضی هر متغیر از آن متغیر محاسبه گردد:

$$Var(X+Y) = E[((X - E[X]) + (Y - E[Y]))^2] \quad (۵)$$

طبق اتحاد مربع مجموع دو جمله‌ای، عبارت درون امید ریاضی سمت راست معادله (۵) را بسط می‌دهیم:

$$Var(X+Y) = E[(X - E[X])^2 + (Y - E[Y])^2 + 2(X - E[X])(Y - E[Y])] \quad (۶)$$

چون امید ریاضی مجموع چندین مولفه، برابر مجموع امیدریاضی هر مولفه است، پس برای معادله (۶) داریم:

$$Var(X+Y) = E[(X - E[X])^2] + E[(Y - E[Y])^2] + E[2(X - E[X])(Y - E[Y])] \quad (۷)$$

تساوی های زیر را در نظر بگیرید:

$$E[(X - E[X])^2] = Var(X) \quad (۸)$$

$$E[(Y - E[Y])^2] = Var(Y) \quad (۹)$$

$$E[2(X - E[X])(Y - E[Y])] = 2E[(X - E[X])(Y - E[Y])] = Cov(X,Y) \quad (۱۰)$$

با اعمال معادلات (۸)، (۹) و (۱۰) بر معادله (۷) به تساوی زیر دست می‌یابیم:

$$Var(X+Y) = Var(X) + Var(Y) + 2Cov(X,Y) \quad (۱۱)$$

بدین ترتیب، حکم ثابت می‌گردد. □

تمرین ۲:

ابتدا محاسبه زیر را انجام می‌دهیم:

$$||\tilde{y} - \tilde{X}w||_2^2 = || \begin{pmatrix} y \\ 0 \end{pmatrix} - c \begin{pmatrix} X \\ \sqrt{\lambda_2} I_d \end{pmatrix} w||_2^2 = ||y - cXw||_2^2 + ||0 - c\sqrt{\lambda_2} w||_2^2$$

بنابراین:

$$||\tilde{y} - \tilde{X}w||_2^2 = ||y - cXw||_2^2 + c^2\lambda_2 ||w||_2^2$$

حال $J_2(w)$ را تشکیل می‌دهیم:

$$J_2(w) = ||\tilde{y} - \tilde{X}w||_2^2 + c\lambda_1 ||w||_1 = ||y - cXw||_2^2 + c^2\lambda_2 ||w||_2^2 + c\lambda_1 ||w||_1$$

سپس J_1 را در cw محاسبه می‌کنیم:

$$J_1(cw) = ||y - X(cw)||_2^2 + \lambda_2 ||cw||_2^2 + \lambda_1 ||cw||_1 = ||y - cXw||_2^2 + c^2\lambda_2 ||w||_2^2 + c\lambda_1 ||w||_1$$

بنابراین برای هر w داریم:

$$J_1(cw) = J_2(w)$$

در نتیجه برای هر w می‌توان نوشت:

$$argmin\ J_1(w) = c\, argmin\ J_2(w)$$

بدین ترتیب حکم ثابت می‌گردد. □

تمرین ۳

در مدل برنولی_ گاوسی، این ایده مطرح می‌گردد که بیشتر وزن‌ها دقیقاً صفر هستند و وزن‌های غیرصفر از یک توزیع گاوسی می‌آیند.

$$p(w_i) = (1 - \pi)\delta(w_i) + \pi N(w_i; 0, \sigma_2)$$

در ادامه به نقش روش تخمین MAP می‌پردازیم که این عبارت را حل می‌کند:

$$w^* = argmax\ p(y|X, w)p(w)$$

با منفی لگاریتم گرفتن درمی‌یابیم که عبارت فوق معادل مینیمم‌سازی عبارت زیر است:

$$-log\ p(y|X, w) - log\ p(w)$$

عبارت $p(w)$ نقشی کلیدی ایفا می‌کند:

- اگر $w_i = 0$ باشد: $p(w_i = 0) = 1 - \pi$ و در نتیجه : $-log(p(w_i = 0)) = -log(1 - \pi)$
- اگر $w_i \neq 0$ باشد: $p(w_i \neq 0) = \pi N(w_i; 0, \sigma_2)$ برقرار است. اگر منفی لگاریتم بگیریم، خواهیم داشت:

$$-log\ p(w_i \neq 0) = \log\ \pi + \frac{w_i^2}{2\sigma^2} + \text{ثابت}$$

در اینجا دو بخش وجود دارد:

- یک جریمه ثابت $-log\ \pi$ که هرگاه وزن غیرصفر باشد، اعمال می‌گردد.
- یک جریمه درجه دوم $\frac{w_i^2}{2\sigma^2}$ (مشابه منظم‌کننده L_2).

اگر جمع را تحت همه وزن‌ها بر کل بردار وزن اعمال کنیم:

$$-log\ p(w) = \Sigma_{i;w_i=0}[-log(1-\pi)] + \Sigma_{i;w_i\neq 0}[-\log\ \pi + \frac{w_i^2}{2\sigma^2}]$$

عبارات را بر اساس تعداد $entry$ های غیرصفر، گروه‌بندی می‌کنیم. قرار می‌دهیم:

$$||w||_0 = \#\{i : w_i \neq 0\}$$

$$C_0 = -log\ 1-\pi$$

$$C_1 = -log\ \pi$$

آن‌گاه، می‌توان نوشت:

$$-log\ p(w) = (C_1 - C_0)||w||_0 + \frac{1}{2\sigma^2}]||w||_2^2 + \text{ثابت}$$

تعریف می‌کنیم:

$$\lambda_0 = C_1 - C_0 = -log\ \pi + \log\ (1-\pi)$$

$$\lambda_2 = \frac{1}{2\sigma^2}$$

در نهایت:

$$-log\ p(w) = \lambda_0||w||_0 + \lambda_2||w||_2^2 + \text{ثابت}$$

پس با روش تخمین MAP به این عبارت خواهیم رسید که شامل منظم‌کننده L_0 یعنی $||w||_0$ است:

$$min_w\ ||y - Xw||_2^2 + \lambda_0||w||_0 + \lambda_2||w||_2^2$$

می‌توان نتیجه گرفت:
این مدل، به صورت طبیعی، باعث تشویق $sparsity$ می‌شود؛ زیرا داشتن وزن غیر صفر هزینه مجزا دارد.

تمرین ۴

در مسئلهٔ یادگیری، پیدا کردن وزن‌ها یعنی پیدا کردن نقطه‌ای که کفِ منحنی خطا و ناحیهٔ محدودکنندهٔ منظم‌ساز به هم برخورد می‌کنند. شکل ناحیهٔ محدودکننده در نرم l_2 به صورت دایره/کره و در نرم l_1 لوزی با گوشه‌های تیز است و گوشه‌های لوزی دقیقاً روی محورهای مختصات قرار دارند. به همین دلیل، زمانی که منحنی خطا به لوزی برخورد می‌کند، بسیار احتمال دارد روی یکی از گوشه‌ها برخورد کند. برخورد روی گوشه یعنی یک یا چند وزن دقیقاً صفر شوند. به عبارت دیگر، گوشه‌های تیزِ لوزی، جاذب هستند و حلِ بهینه را وادار می‌کنند روی محور قرار بگیرد. این درحالی است که شکل ناحیه محدودکننده l_2 کاملاً گرد و بدون گوشه است. زمانی که منحنی خطا به این کره برخورد می‌کند، نقطهٔ برخورد تقریباً هیچ‌وقت دقیقاً روی یک محور نیست؛ بنابراین هیچ وزن شخصاً دقیقاً صفر نمی‌شود همهٔ وزن‌ها فقط کوچک می‌شوند ولی حذف نمی‌شوند پس، نرم l_2 حالت $shrinkage$ دارد نه حذف‌کننده.
به همین جهت، نرم l_1 در انتخاب ویژگی ، مدل‌های قابل تفسیر، داده‌های با ویژگی‌های بسیار زیاد و زمانی که می‌دانیم تعداد کمی ویژگی مهم‌اند کاربرد دارد و نرم l_2 در زمانی که همهٔ ویژگی‌ها کمابیش مؤثرند، زمانی که همبستگی بین ویژگی‌ها زیاد است به کار می‌رود.

گزارش تمرین عملی ۱

- روش‌های ساده مانند ($Mean$، $Gaussian$، $Uniform\ Noise$):

این روش‌ها پایین‌ترین عملکرد را داشتند. به‌ویژه نویز یکنواخت و نویز گاوسی که به دلیل برهم‌زدن ساختار محلی تصویر، SSIM بسیار پایینی تولید کردند. $Mean\ Imputation$ نسبت به نویزدهی نتایج بهتری دارد اما همچنان ساختارهای پیچیده تصویر را بازسازی نمی‌کند. در کل، روش‌های ساده با افزایش درصد حذف، افت SSIM قابل‌توجهی نشان دادند.

- روش‌های رگرسیونی ساده ($Linear$، $Ridge$، $Lasso$):

این مدل‌ها همگی عملکرد مشابهی داشتند (میانگین SSIM ≈ 0.476). دلیل این مسئله آن است که مدل‌های خطی تنها قادر به تخمین روابط خطی بین همسایگی پیکسل‌ها هستند و ساختارهای بافتی پیچیده مانند لبه‌ها را به‌خوبی بازسازی نمی‌کنند. افت SSIM در این روش‌ها از ٪۱۰ به ٪۳۰ حذف حدود 0.16 است که نشان می‌دهد این مدل‌ها نسبت به کمبود داده حساس هستند.

- رگرسیون چندجمله‌ای ($Polynomial\ Regression$): این خانواده‌ی مدل‌ها بهترین نتایج را تولید کردند.
  - افزایش درجه چندجمله‌ای تا حدود درجه ۸ باعث بهبود پیوسته کیفیت بازسازی می‌شود.
  - پس از آن (درجه ۹ و ۱۰)، نوسان‌هایی دیده می‌شود که می‌تواند ناشی از بیش‌برازش و حساسیت به نویز باشد.
  - بهترین نتیجه کلی مربوط به درجه ۹ است (میانگین $SSIM = 0.6867$).
  - بهترین نتیجه در حذف ٪۱۰ مربوط به درجه ۸ بوده است ($SSIM = 0.7657$).
  - این روش‌ها به دلیل توانایی در مدل‌کردن الگوهای غیرخطی، لبه‌ها، بافت‌ها و تغییرات شدت روشنایی، جزئیات بیشتری را در تصویر بازسازی می‌کنند و برتر از روش‌های خطی عمل کرده‌اند.

- نکته‌ای درباره اثر افزایش درصد حذف پیکسل‌ها: با افزایش نسبت پیکسل‌های حذف‌شده از ٪۱۰ به ٪۳۰، تمام روش‌ها افت کیفیت نشان دادند، اما تفاوت‌ها معنادار بود. روش‌های ساده افت شدید دارند. رگرسیون خطی و Ridge و Lasso افتی حدود 0.16 دارند. چندجمله‌ای‌ها به دلیل قدرت مدل‌سازی بیشتر، بهترین مقاومت را در برابر افزایش $Missing\ Rate$ دارند. به طور کلی، هرچه مدل توانایی بیشتری در یادگیری روابط محلی داشته باشد، عملکرد بهتری ارائه داده است.

گزارش تمرین عملی ۲

- عملکرد مدل‌ها در داده‌های بدون نویز و بدون outlier :
  - اندازه نمونه کوچک (۱۰۰ داده) :
    - رگرسیون خطی عملکرد متوسطی دارد ($MSE \approx 0.07$).
    - رگرسیون چندجمله‌ای به‌وضوح بهتر عمل می‌کند؛ بهترین عملکرد با درجه ۸ و $MSE \approx 0.035$.
    - روش‌های Ridge/Lasso/ElasticNet تفاوت چندانی با Linear ندارند چون مدل آن‌ها هنوز خطی است.

تابع sinc رفتار نوسانی دارد و مدل خطی قادر به مدل‌سازی این رفتار نیست؛ اما مدل چندجمله‌ای با درجه مناسب ساختار تابع را به خوبی بازسازی می‌کند.

- عملکرد روی داده‌های نویزی : افزودن نویز Gaussian با واریانس کم باعث افت عملکرد مدل‌ها نمی‌شود. الگو تقریباً مشابه حالت بدون نویز است.
  - همچنان $Polynomial\ Regression$ $(degree \approx 8)$ بهترین است $(MSE \approx 0.0307)$.
  - مدل‌های Ridge/Lasso/ElasticNet روی داده‌های کوچک تمایلی به یادگیری غیرخطیت ندارند، چون پایه ویژگی‌ها خطی است.
- تغییر عملکرد با افزایش تعداد نمونه (۳۰۰ و ۱۰۰۰ داده) :
  - با افزایش تعداد نمونه‌ها، $Polynomial\ Regression$ عملکرد خارق‌العاده‌ای پیدا می‌کند.
  - در ۳۰۰ نمونه بهترین $MSE \approx 0.0038$
  - در ۱۰۰۰ نمونه بهترین $MSE \approx 0.0155$
  - درجه بهینه معمولاً با افزایش تعداد نمونه‌ها افزایش می‌یابد (برای ۳۰۰ نمونه درجه ۱۲، برای ۱۰۰۰ نمونه درجه ۱۰).

  هرچه داده بیشتر می‌شود، مدل چندجمله‌ای بهتر می‌تواند نوسانات sinc را یاد بگیرد و خطر overfitting کمتر می‌شود.
- اثر وجود Outliers : Outlier ها بیشترین تأثیر را روی مدل‌های حساس مانند $Polynomial\ Regression$ و $Linear\ Regression$ دارند.
  - تغییرات در نمونه ۱۰۰ :
    - MSE مدل چندجمله‌ای از 0.035 به 0.0568 افزایش می‌یابد (افت نسبی کیفیت).
    - Lasso بهترین مقاومت را نشان می‌دهد (کمترین افت MSE حدود صفر درصد).
  - در نمونه‌های ۳۰۰ و ۱۰۰۰ :
    - همچنان Polynomial بهترین مدل باقی می‌ماند ولی حساسیت به نقطه پرت دارد.
    - مدل‌های Lasso و ElasticNet پایدارتر هستند و افت عملکرد بسیار کم است.

  $Polynomial\ Regression$ دقیق‌ترین است ولی نسبت به outlier ها حساس است.
  Lasso و ElasticNet مقاوم‌ترین رفتار را دارند چون وزن ویژگی‌ها را محدود کرده و اثر شدید نقاط پرت را کاهش می‌دهند.

گزارش تمرین عملی ۳

در خروجی فایل پیوست شده، به وسیله نمودارها و جداول مختلف، بهترین مدلها، مقایسه عملکرد آنها، دسته‌بندی کیفی به طور کامل ارائه شده است. بر اساس نتایج حاصل‌شده:

- مدل $Gradient\ Boosting\ Regressor$ دقیق‌ترین پیش‌بینی را ارائه داد و از نظر قدرت کلی یادگیری بهترین عملکرد را داشت و از نظر میانگین $K-Fold\ R^2$ برترین بود.
- پایدارترین مدل از نظر واریانس پایین در K-Fold ، مدل XGBoost بود.
- بهترین مدل از نظر تعادل عملکرد + پایداری XGBoost بود.
- درخت تصمیم به شکل مشخص بیشترین شکاف را دارد و در کاربرد واقعی قابل اعتماد نیست.

- مدل‌هایی که بیشترین نشانه‌های Overfitting را داشتند : $Random\ Forest$ و Gradient Boosting
- دسته‌بندی کیفی:
  - عالی و پایدار :
    Gradient Boosting, XGBoost, Extra Trees, Random Forest,
    Bagging, AdaBoost, HistGradientBoosting
  - خوب و پایدار:
    Ridge, Linear Regression, Poisson, Tweedie, Kernel Ridge, Bayesian Ridge
  - متوسط :
    SVR Linear, ARD, Lasso, Gamma, Theil–Sen, Huber, KNN
- مدل‌های مبتنی بر Boosting و $Ensemble\ Methods$بهترین دقت و بیشترین پایداری را در پیش‌بینی قیمت خانه دارند.