



درس : یادگیری ماشین

دانشجو : امیرمحمد خرازی

شماره دانشجویی : ۴۰۱۵۲۵۲۱۰۰۲

استاد درس : دکتر منصور رزقی آهق

دانشکده علوم ریاضی ، گروه علوم کامپیوتر، گرایش داده‌کاوی

پرسش‌های کلاسی سری دوم

گیت‌هاب این پرسش ( لینک )

گیت‌هاب درس ( لینک )

## هدف مسئله

تعدادی نقطه به تصادف در بازه  $[0, 1]$  در نظر گرفته شده است. می‌خواهیم تابع  $\sin(2\pi x)$  را بازسازی کنیم. بدین منظور ۵ دیتاست مختلف به ما داده شده است. این دیتاست‌ها هر کدام با هم تفاوتی دارند.

- دیتاست اول : ۱۰ نقطه تصادفی روی همان تابع سینوس است
- دیتاست دوم : ۱۰ نقطه تصادفی روی همان تابع سینوس است که کمی نویز دارند.
- دیتاست سوم : نمونه‌های تست ما هستند.
- دیتاست چهارم: ۱۰ نقطه تصادفی روی تابع سینوس به همراه ۲ نقطه دور افتاده .
- دیتاست پنجم : شامل ۵۰ نقطه مانند دیتاست دوم است.

برای هر دیتاست می‌خواهیم چند جمله‌ای رگرسیون را برای درجات  $M = 1, \dots, 9$  تشکیل داده و سپس مدل‌های بدست آمده را (۳۶ مدل را) روی نمونه‌های آزمون بررسی کنیم و نتایج را با توجه به  $MAE$  و  $RMSE$  ارزیابی کنیم.

## چند جمله رگرسیون

در رگرسیون چند جمله، یک تابع پایه داریم که هر  $x$  را به بعد دیگری می‌برد. این تابع پایه در این مسئله  $x^j$  است. مثلاً اگر داشته باشیم  $x_i$  و بخواهیم چند جمله درجه  $M$  آن را بسازیم، قرار می‌دهیم:

$$\Phi(X_i) = \begin{bmatrix} 1 \\ x_i^1 \\ x_i^2 \\ \vdots \\ x_i^M \end{bmatrix}$$

این  $\Phi$  داده ما را به بعدی  $M + 1$  برد. یعنی بجایی اینکه از روش‌های غیر خطی استفاده کنیم، فرض می‌کنیم داده ما در بعد  $M + 1$  قرار دارد و سپس با روش‌های خطی آن را حل می‌کنیم.  
یک چند جمله درجه  $M$  روی متغیر تک بعدی  $X$  به شکل زیر است:

$$W_0 + W_1 X_1 + W_2 X_1^2 + \dots + W_M X_1^M$$

واضح است که اگر برداری به  $X$  ها و  $Y$  ها نگاه کنیم، برای  $N$  داده خواهیم داشت  
که  $Y \approx \Phi X$

$$\Phi = \begin{bmatrix} \Phi^T(X_1) \\ \Phi^T(X_2) \\ \vdots \\ \Phi^T(X_N) \end{bmatrix}$$

## روش کار

میخواهیم تفاوت این  $y$  و تخمین زده شده‌ی آن یعنی  $\Phi W$  حداقل شود.  
می‌توانیم از نرم ۱ یا ۲ برای این کار استفاده کنیم.  
ما در اینجا با نرم ۲ می‌رویم. نرم ۲ توان ۲ فاصله‌ها را در نظر می‌گیرد. مشکل اینجاست که توان ۲ فاصله‌ها به نویز و نقاط پرت حساس می‌شود.

$$\min_W \{ \|\Phi W - Y\|_2^2 \}$$

## بهینه سازی

با کمک نرم ۲ می‌توانیم از روش‌های ماتریسی برای حل این مسئله بهینه سازی استفاده کنیم لذا جواب می‌شود :

$$W = (\Phi^T \Phi)^{-1} \Phi^T y = \Phi^\dagger y$$

## کد

با کمک این فرمول‌ها می‌توانیم برای وزن تمامی این ۳۶ مدل تصمیم بگیریم. عدد حالت  $\Phi^T \Phi$  می‌تواند روی تاثیر نویز بر  $W$  واقعی تاثیر داشته باشد. بدین صورت که اگر بد حالت باشد، نویز بیشتری را در نظر می‌گیرد و لذا مدل ما به احتمال زیاد روی داده‌های آزمون بیش‌برازش خواهد داشت. البته در کد ما از scikit-learn استفاده می‌کنیم.

## نتیجه

جزئیات کامل کد و نتیجه گیری‌های گرفته شده به همراه تصاویر در فایل Ipython. موجود است. لذا برای بررسی کد می‌توانید به لینک گیت‌هاب، یا گوگل مراجعه کنید. این کد در کنار این گزارش نیز ارسال می‌شود.  
در حالت کلی مدل‌های دیتاست‌های ۴ و ۲ بد عمل می‌کردند. دیتاست ۵ با داده‌های زیاد مشکل نویزی خود را تا حدی حل کرد، دیتاست اول که نسبتاً خیلی خوب بود.