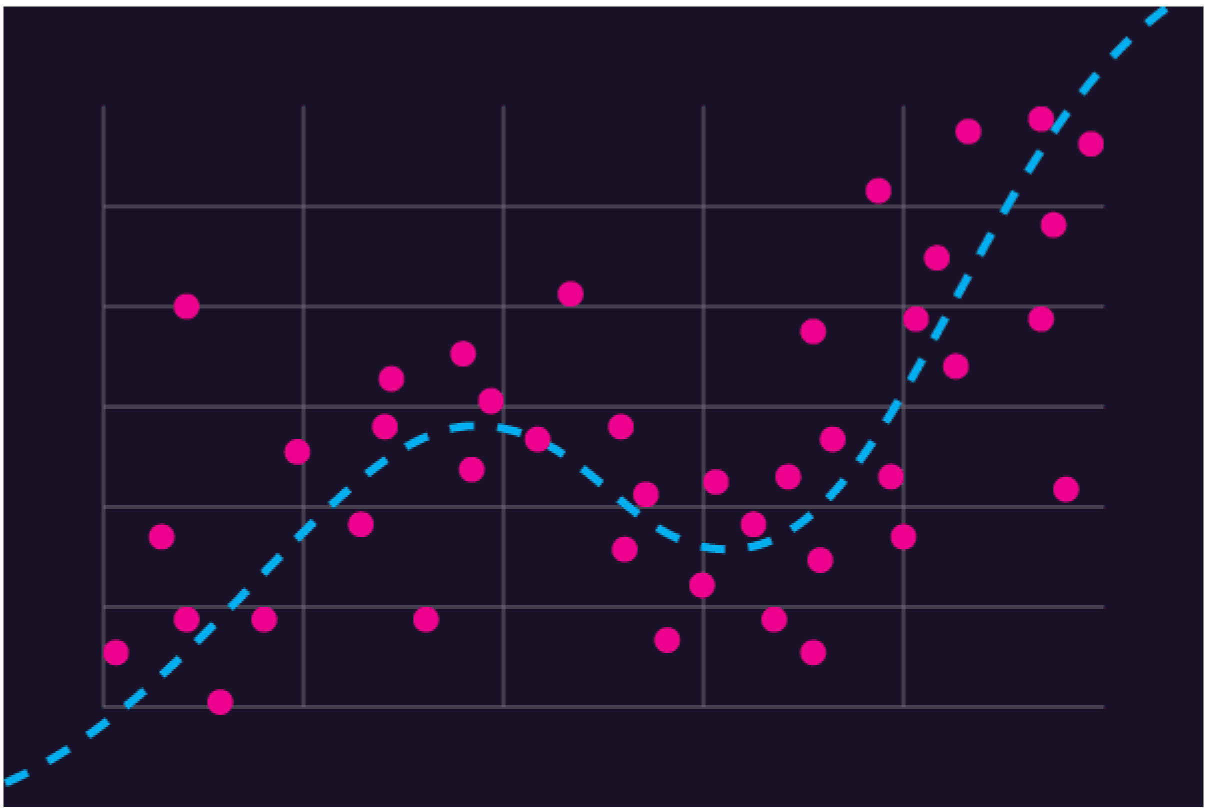
مساله کمترین مربعات[1]



[1] Least Squares Problem

فهرست مطالب

مقدمه

امروزه، آدمای زیادی رو می بینیم که وقتی صحبت از رمزارزها و دنیای کریپتو میشه، میگن که اگه اون اوایل که بیت کوین اومده بود، یه دونه بیت کوین می خریدیم، الان میلیاردر می شدیم...

اما سوالی که اینجا پیش میاد و به شدت جذابه اینه که "واقعا راهی وجود داره که از قبل بتونیم تشخیص بدیم، در آینده چه اتفاقی میفته؟" جوابش قطعیش، یه **نه بزرگه** چون جهانی که توش زندگی می کنیم به شدت پیچیدست و خب همون طور که میدونیم نمیشه آینده رو از قبل به طور قطعی پیش بینی کرد.

اما نکته ای که وجود داره اینه که، اگه بخوایم یه پیش بینی احتمالی از آینده داشته باشیم چی؟ اینجاست که میشه گفت **جواب مثبته :)**

یکی از راه‌ها، برای اینکه بتونیم اتفاقایی که تو آینده میتونه بیفته رو تحلیل کنیم، استفاده از داده‌هاییه که مربوط به گذشته است. ما می تونیم با استفاده از داده‌هایی که تا الان داشتیم و استفاده از یک سری الگوریتم خاص، این کارو انجام بدیم.

احتمالا همون طور که حدس می زنین، شما همین الانشم با یکی از این الگوریتم‌ها آشنا هستین و اون الگوریتم چیزی نیست به جز الگوریتم حل مساله Least Square :)

بریم که ببینیم اصلا داستان این مساله چیه. تو قسمت بعد می‌بینمتون :)

توضیح مساله کمترین مربعات

خب، حالا که تا حدی متوجه شدین که داستانی که امروز باهاش سر و کار داریم چیه، می‌خوایم یه خورده بیشتر در رابطه با تئوری این الگوریتم صحبت کنیم.

مساله Least Square به این شکله که ما یک زیرفضای برداری داریم که اون رو می‌تونیم به صورت فضای ستونی از یک ماتریس مثل ماتریس A نمایش بدیم. این جا نقاطی که در کل زیرفضای ما وجود دارند، دو حالت میشن.

1) اون نقطه‌ها در فضای ستونی ماتریس A وجود داره پس ما طبق اون چیزهایی که تا الان یاد گرفتیم، می‌تونیم یه نقطه ای مثل x پیدا بکنیم که با اعمال ضرب ماتریسی Ax به اون نقطه برسیم.
2) اون نقطه‌ها در فضای ستونی ماتریس A وجود نداره پس ما نمیتونیم هیچ نقطه x رو پیدا کنیم تا اون رو بسازیم.

حالا فرض کنید که ما می خواهیم کل فضای برداریمون رو صرفا با استفاده از زیرفضای $Col\ A$ تقریب بزنیم. تو این حالت که تکلیف اون نقاطی که جزو حالت (1) بودن کاملا مشخصه چون تو این حالت تقریبا میشه گفت تقریب زدن معنی نداره چون ما میتونیم این مقدار رو دقیق محاسبه کنیم.

اما تو حالت دوم چی؟

تو این حالت درسته که مقدار دقیقی براش وجود نداره، اما ما برای اینکه خطامون رو حداقل کنیم، می‌تونیم به جای x، نقطه x' رو انتخاب کنیم که نزدیک ترین نقطه روی زیرفضای $Col\ A$ نسبت به نقطه x هست.

حالا این جاست که باید یه لامپ بقل سرتون روشن بشه :)

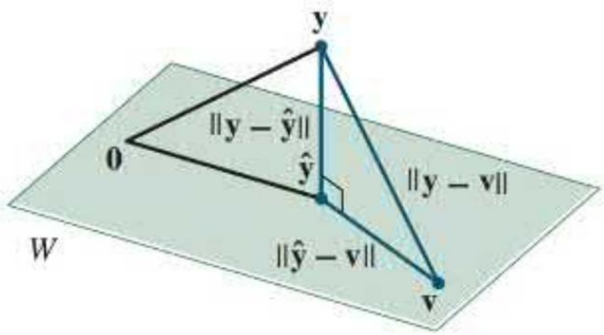
قضیه 9 (قضیه بهترین تخمین):

فرض کنید W یک زیرفضایی از $\mathbb{R}^n$، y هر بردار دلخواهی در $\mathbb{R}^n$ و در نهایت $\hat{y}$ تصویر عمودی y بر روی زیرفضای W ($\hat{y} = proj_W y$)باشد. در این صورت، $\hat{y}$ نزدیک ترین نقطه در W به y است. به گونه ای که رابطه زیر

$$||y - \hat{y}|| < ||y - v||$$

به ازای هر بردار v دلخواه و متمایز از $\hat{y}$ در W برقرار است.

برای اینکه قضیه بالارو بهتر درک کنین، همانطور که در شکل پایین مشاهده می‌کنین، نزدیک ترین نقطه در زیر فضای W نسبت به y، تصویر عمودی y بر W است.



همونطور که توی قضیه 9 کتاب دیدیم، در اصل نزدیک ترین نقطه روی $Col\ A$ به این نقطه اولیه ما (یا به عبارت بهتر، بهترین تخمینی که با $Col\ A$ می‌تونیم ازش بزنیم)، همون تصویر عمودی (orthogonal projection) نقطه اولیه روی $Col\ A$.

در نهایت اگه از این خاصیت‌هایی که تا به این جا توضیح دادیم، استفاده کنید، میرسیم به یه معادله خیلی خیلی معروف به اسم معادله نرمال (Normal Equation) که به شکل زیر هست:

$$A^T A x = A^T b$$

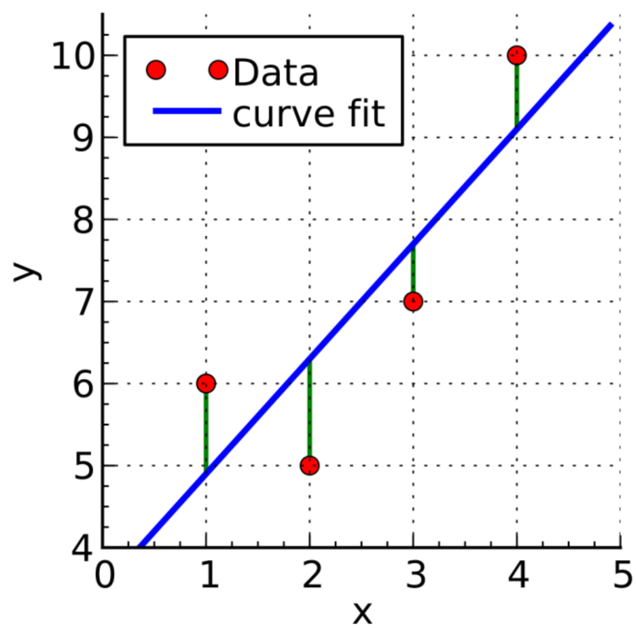
که خب توی این معادله، b همون برداریه که قصد داریم بهترین تخمینش رو روی $Col\ A$ پیدا کنیم و x مقداری است که به واسطه آن نزدیک ترین نقطه روی $Col\ A$ به b به دست میاد.

رگرسیون[2]

خب حالا که کامل با الگوریتمی که برای حل این مساله ارائه شده آشنا شدین، وقتشه که یه کمی در رابطه با رگرسیون خطی صبحت کنیم.

رگرسیون خطی[3]

برای سادگی فرض کنید که شما یه سری داده دو بعدی دارین (به این معنی که تمامی اون‌ها رو می‌تونیم به فرم $\begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$ بنویسیم.) مثل شکل زیر



همونطور که تو شکل بالا می‌بینین در کل چهار نقطه داده (datapoint) وجود داره.

$$(x, y): [\begin{bmatrix} 1 \\ 6 \end{bmatrix}, \begin{bmatrix} 2 \\ 5 \end{bmatrix}, \begin{bmatrix} 3 \\ 7 \end{bmatrix}, \begin{bmatrix} 4 \\ 10 \end{bmatrix}]$$

حالا ما قصد داریم که این فضای 2 بعدی رو با یک زیرفضای یک بعدی تخمین بزنیم. اما نکته ای که وجود داره اینه که ما دنبال هر زیرفضای برداری یک بعدی نیستیم، بلکه ما دنبال اون زیرفضای برداری یک بعدی هستیم که کمترین خطا رو داشته باشه.

[2] Regression
[3] Linear Regression

خب اما سوالی که پیش میاد اینجا اینه که خطا چیه؟

خطا در واقع اینجا همون فاصله بین datapoint های اصلی و تخمینیه که ما به واسطه زیرفضای یک بعدیمون به دست اوردیم.

$$error = y - y'$$

تا اینجا کلی گفتیم زیرفضای یک بعدی، اما می‌تونیم این رو خیلی ساده تر هم بگیم. این به این معنیه که ما یک خط پیدا کنیم که کمترین خطا را نتیجه بدهد. از قبل می‌دونیم که معادله خط به شکل زیر هست:

$$a_0 + a_1 x = y$$

که توی اینجا a_0 همون عرض از مبدا این خط و a_1 هم شیب این خط محسوب میشه. پس با این حساب می‌تونیم بگیم، وقتی که میگیم ما دنبال زیرفضای برداری یک بعدی هستیم که کمترین خطا رو داشته باشه، در اصل منظورمون پیدا کردن مقادیر مناسبی برای a_0 و a_1 هست.

حال میتونیم معادله مربوط به این مساله رو به شکل زیر بنویسیم:

$$\begin{aligned} a_0 + a_1 x_1 &= y_1 \\ a_0 + a_1 x_2 &= y_2 \\ a_0 + a_1 x_3 &= y_3 \\ a_0 + a_1 x_4 &= y_4 \end{aligned}$$

که خب ما می‌تونیم این معادله رو به شکل یک معادله ماتریسی در بیاریم.

$$Ax = \begin{bmatrix} 1 & x_1 \\ 1 & x_2 \\ 1 & x_3 \\ 1 & x_4 \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \end{bmatrix} = \begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \end{bmatrix} = y$$

که در این حالت می‌بینیم معادله ای که به دست اوردیم، شبیه به معادله استاندارد مساله Least Square هست پس ما می‌تونیم از راه حلی که برای اون مساله به دست اوردیم برای این مساله نیز استفاده کنیم.

$$A^T A x = A^T y$$

اینجا به شرطی که $A^T A$ وارون پذیر باشه، میتونیم معادله رو به شکل زیر بنویسیم.

$$x = (A^T A)^{-1} A^T y$$

و خب به این شکل ما تونستیم عرض از مبدا و شیب خط مناسب برای تخمین datapoint های dataset رو به دست بیاریم و اگر در آینده لازم باشه که مقدار مربوط به داده‌هایی غیر از داده‌هایی رو که در اختیار داریم رو به دست بیاریم، به سادگی می‌تونیم این کار رو انجام بدیم.

رگرسیون چندجمله ای[4]

نکته ای که اینجا وجود داره اینه که لزوما تابعی که پیدا میکنم نیازی نیست خطی باشه. می‌تونه هر تابع چندجمله ای باشه اما برای این کار لازمه که یه خورده تغییراتی روی معادله اولیه اعمال کنیم و به جای اینکه از معادله خطی استفاده کنیم، از معادله درجه دو استفاده کنیم.

$$a_0 + a_1x + a_2x^2 = y$$

و ادامه کار مثل فرایندی هست که توی مرحله قبل برای رگرسیون خطی انجام دادیم. اینجا لازمه که دستگاه معادلات درجه دو رو تشکیل بدیم و بعد از اون ماتریس اصلی و بردار مورد نظرمون رو به دست بیاریم.

$$a_0 + a_1x_1 + a_2{x_1}^2 = y_1$$
$$a_0 + a_1x_2 + a_2{x_2}^2 = y_2$$
$$a_0 + a_1x_3 + a_2{x_3}^2 = y_3$$
$$a_0 + a_1x_4 + a_2{x_4}^2 = y_4$$

$$Ax = \begin{bmatrix} 1 & x_1 & x_1^2 \\ 1 & x_2 & x_2^2 \\ 1 & x_3 & x_3^2 \\ 1 & x_4 & x_4^2 \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ a_2 \end{bmatrix} = \begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \end{bmatrix} = y$$

در نهایت چون به فرم معادله مساله Least Square رسیدیم، می تونیم از قضیه نرمال استفاده کنیم و مقادیر a_0, a_1, a_2 را به دست بیاریم.

$$A^T Ax = A^T y$$

[4] Polynomial Regression

پیاده‌سازی

یک دیتاست (covid_cases.csv) در اختیار شما قرار داده شده است که میزان آمار مبتلایان کرونا در جهان را در خود دارد. شما می توانید با استفاده از تابع read_csv() موجود در کتابخانه pandas، محتویات این دیتاست را بخوانید.

ستون World از این دیتاست بیانگر میزان ابتلای به کرونا در مقیاس ۱۰۰۰۰۰ برابر در کل جهان است.(عدد ۱ به معنای ۱۰۰۰۰۰ مبتلای به کرونا است.)

به کمک این دیتاست و الگوریتم Least Square باید با آموزش مدل رگرسیون، به پیش‌بینی داده‌های جدید بپردازید.

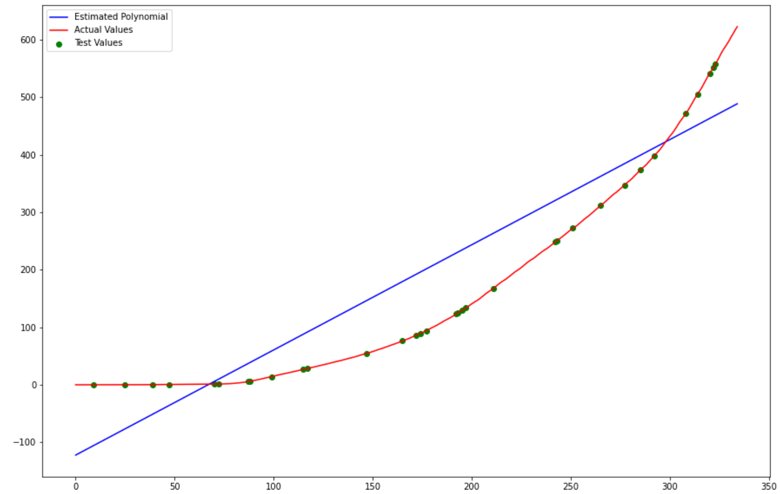
با استفاده از ۹۵ درصد از داده‌های موجود در دیتاست، یک مدل رگرسیون درجه۱ آموزش دهید. سپس ۵ درصد باقی‌مانده از دیتاست اولیه را به عنوان دیتاست آزمون در نظر بگیرید و خطای مدل خود را حساب کنید.

به ازای ۵ داده از داده های موجود در دیتاست آزمون، به صورت زیر گزارش بگیرید.

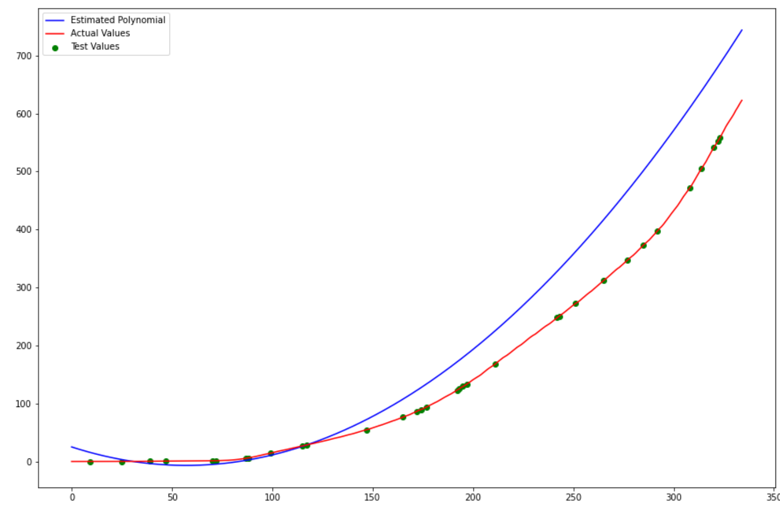
Real value:?
Estimated value:?
Error:?

مراحل گفته شده را برای آموزش یک مدل درجه ۲ تکرار کنید.

به ازای هر دو مدل، نموداری مانند زیر رسم کنید.



نمونه یک مدل درجه ۱



نمونه یک مدل درجه ۲

دقت کنید که دو نمودار بالا صرفاً به عنوان نمونه آورده شده است و ممکن است خروجی شما همانند بالا نشود.

تفکر : به نظر شما کدام مدل می تواند پیش بینی دقیق تری از داده های دیتاست ما ارائه دهد؟

قوانین و ددلاین

- ددلاین این پروژه، ساعت 23:59 روز 12 دی می‌باشد.
- تنها نیاز است کد پایتون خود را بعد از تکمیل، در صفحه کوئرا آپلود کنید.
- هر دانشجو می‌بایست پروژه را به صورت انفرادی انجام دهد. تقلب‌ها به صورت خودکار، توسط سامانه کوئرا بررسی خواهد شد.
- از آن جایی که زبان برنامه نویسی پایتون، یکی از زبان‌های پرکاربرد در حوزه جبر خطی است و آموزش‌های مربوط به این زبان و کتابخانه‌های آن، توسط تیم تدریس‌یاری در اختیار شما قرار گرفته است، بنابراین برای پیاده سازی این پروژه تنها مجاز به استفاده از زبان پایتون و کتابخانه های Matplotlib ،Numpy و Pandas در کنار توابع و کتابخانه‌های پیش فرض پایتون هستید. استفاده از هر زبان برنامه‌نویسی یا کتابخانه‌ای دیگر قابل قبول نبوده و در صورت استفاده، نمره‌ای به شما تعلق نخواهد گرفت.
- رعایت تمیزی کد، استفاده از توابع مختلف برای پیاده سازی پروژه به شدت استقبال می‌شود.

با آرزوی موفقیت و سلامتی

تیم تدریس یاری جبر خطی کاربردی، پاییز ۱۴۰۰