همانطور که پیش تر گفته شد برای min یا max تعیین کردن یک تابع نیاز ا ست که م شتق مرتبه دوم آن نیز برر سی شود. در حالت ماتریس مشتق مرتبه دوم با نام ماتریس Hessian شناخته می شود. به این ترتیب نیاز است مثبت یا منفی بودن یک ماتریس تعریف شود. اگر A یک ماتریس با ابعاد P×P و بردار x با بعد P داشته باشیم آنگاه :

$A \in R^{P \times P}$, $x \in R^{P}$

یک عدد است

A is positive definite(PD) if $x^T A x > 0$ for all nonzero x

A is semi positive definite if $x^T A x \geq 0$ for all nonzero x

A is negative definite if $x^T A x < 0$ for all nonzero x

A is semi negative definite if $x^T A x \leq 0$ for all nonzero x

A is indefinite if $\exists x_1, x_2 \Rightarrow \begin{cases} x_1^T A x_1 > 0 \\ x_2^T A x_2 < 0 \end{cases}$

* می توان نشان داد که اگر یک ماتریس قطری تمام عناصرش مثبت باشد می توان گفت که مثبت معین است.

به عنوان چند تعریف از جبر خطی داریم (البته اگه A و B مربعی باشند)

$$(AB)^T = B^T A^T$$

$$(AB)^{-1} = B^{-1} A^{-1}$$

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

یک مجموعه بردار وابسته خطی هستند اگر :

a set of vectors $\{x_1, \dots, x_n\} \in R^P$ is said to linearly dependent

if $\exists \alpha = (\alpha_1, \dots, \alpha_n) \neq 0 \Rightarrow \alpha_1 x_1 + \alpha_2 x_2 + \dots + \alpha_n x_n = 0$

در جبر خطی linear combination به فرم زیر تعریف می شود :

$$\text{if } X = \begin{bmatrix} | & | & & | \\ x^0 & x^1 & \cdots & x^p \\ | & | & & | \end{bmatrix} \quad \text{and} \quad \beta = \begin{bmatrix} \beta_0 \\ \vdots \\ \beta_p \end{bmatrix}$$

$$X_{(i)}\beta = \beta_0 x^0 + \beta_1 x^1 + \cdots + \beta_p x^p = \sum \beta_i x^i = \text{Linear combination}$$

i → سطر ام

به عنوان یک مثال از تعریف بالا می توان به regression ا شاره کرد. به این ترتیب ستون ها در واقع ویژگی ها هستند و سطرها مشاهدات هستند. به این ترتیب ترکیب خطی یک مشاهده ، حاصلضرب آن سطر در بردار بتاها خواهد بود.

Rank یک ماتریس برابر است با تعداد سطرها یا ستون های آن ماتریس است که مستقل خطی هستند.

Multi Linear Regression

در بحث رگر سیون خطی دیدم که در حالتی که نمونه ها multi variable می توان محا سبات را به فرم ماتریسی ن شان داد. در این حالت نیز برای کمینه کردن تابع loss نیاز اســت ابتدا نقاط بحرانی را با محاســبه gradient محاســبه کرد و برای اطمینان از نقطه کمینه ماتریس Hessian باید در این نقطه positive definite باشد. به این ترتیب می توان محاسبات را به فرم زیر ادامه داد :

$$\mathcal{L}(\beta) = \frac{1}{2}(Y - X\beta)^T(Y - X\beta)$$

$$\nabla_\beta \mathcal{L}(\beta) = \frac{\partial \mathcal{L}(\beta)}{\partial \beta} = 0 \quad \& \quad \nabla_\beta^2 \mathcal{L}(\beta) > 0$$

$$\mathcal{L}(\beta) = \frac{1}{2}\left(Y^TY - \underbrace{Y^TX\beta}_{1\times n \; n\times p \; p\times 1} - \underbrace{(X\beta)^TY}_{\beta^TX^TY} + \underbrace{(X\beta)^TX\beta}_{\beta^TX^TX\beta}\right)$$

اندازه است پس ترا نهاده آن با خودش برابر است

$$\downarrow \quad (Y^TX\beta)^T = Y^TX\beta$$

$$\mathcal{L}(\beta) = \frac{1}{2}\left(y^T y - 2\beta^T X^T y + \beta^T (X^T X)\beta \right)$$

symetric

$$\Rightarrow$$

$$(X^T X)^T = X^T (X^T)^T = X^T X \Rightarrow \text{symetric}$$

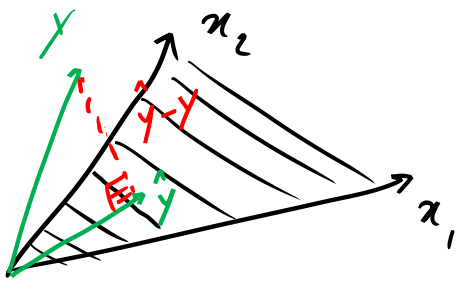
$$\frac{\partial \mathcal{L}(\beta)}{\partial \beta} = \frac{1}{2}(-2X^T y + 2X^T X\beta) = 0 \Rightarrow X^T X\beta = X^T y$$

$$\text{if } (X^T X) \equiv \underbrace{\text{non singular}}_{\text{معکوس پذیر باشد}} \Rightarrow \boxed{\hat{\beta} = (X^T X)^{-1} X^T y}$$

به این ترتیب این معادلات زمانی برقرار هستند که X^TX وارون پذیر باشد و ماتریس معکوس آن را بتوان حساب کرد. این در حالی است که می دانیم همیشه این وضعیت ممکن نیست.

معادلات مطرح شده را می توان از نگاه هندسی هم بیان کرد. در این دیدگاه می دانیم که $\hat{Y}$ باید در فضای محاسباتی و پوشاننده X یا همان span(X) یا column space (X) قرار گیرد. این درحالی است که Y می تواند خارج از این فضا باشد. به این ترتیب می توان نشان داده که $\hat{Y}$ در واقع تصویر Y در span(X) است. این موضوع به خوبی از normal equation که به فرم زیر تعریف می شود قابل برداشت است:

$$\nabla_\beta \mathcal{L}(\beta) = 0 \Rightarrow X^T y - X^T X\beta = 0 \Rightarrow \underbrace{X^T(y - X\beta) = 0}_{\text{normal equation}}$$



در مرحله بعد لازم است که ماتریس Hessian را حساب کنیم ببینم که positive definite هست یا نه. تا متوجه شویم که نقطه کمینه را بدست آورده ایم یا نه. بنابراین داریم :

$$\nabla_{\beta}^{2} \mathcal{L}(\beta) = X^{T} X$$

با یک مثال ساده می توان متوجه شد که این ماتریس همیشه positive definite ا ست! فرض کنیم که داده ها تنها دو ویژگی دارند. به این ترتیب داریم :

$$X^{T}X = \begin{bmatrix} x_0[1] & \cdots & x_0[n] \\ x_1[1] & \cdots & x_1[n] \end{bmatrix} \begin{bmatrix} x_0[1] & x_1[1] \\ \vdots & \vdots \\ x_0[n] & x_1[n] \end{bmatrix} =$$

$$\begin{bmatrix} \sum_{i=1}^{n} x_0^2[i] & \sum_{i=1}^{n} x_0[i]\,x_1[i] \\ \sum_{i=1}^{n} x_0[i]\,x_1[i] & \sum_{i=1}^{n} x_1^2[i] \end{bmatrix}$$

if $A \equiv PD$ then $\forall z \Rightarrow z^{T} A z > 0$ so,

$z^{T} = (z_0, z_1) \Rightarrow$

$$z^T(x^Tx)z = \begin{bmatrix} \sum_{i=1}^{n} z_0 x_0^2[i] + z_1 x_0[i]x_1[i] & \sum_{i=1}^{n} z_0 x_0[i]x_1[i] + z_1 x_1^2[i] \end{bmatrix} \begin{bmatrix} z_0 \\ z_1 \end{bmatrix}$$

$$\Rightarrow z^T(x^Tx)z = \sum_{i=1}^{n} \left(z_0^2 x_0^2[i] + z_1^2 x_1^2[i] + 2 z_1 z_0 x_0[i] x_1[i] \right) \Rightarrow$$

$(a+b)^2$ یا

$$\text{if } \begin{cases} w_0[i] = z_0 x_0[i] \\ w_1[i] = z_1 z_1[i] \end{cases} \Rightarrow z^T(x^Tx)z = \sum_{i=1}^{n} \left(w_0[i] + w_1[i] \right)^2$$

به این ترتیب جواب نهایی همواره بزرگتر یا مساوی صفر است. می توان نشان داد که اگر x یک ماتریس full rank باشد آن گاه جواب معادله هیچگاه صـفر نخواهد بود و به این ترتیب بردار بتا تخمین زده شـده نقطه کمینه در loss function را نمایش می دهد.

$$\text{if } w_0[i] \neq -w_1[i] \Rightarrow z^T(x^Tx)z = \sum_{i=1}^{n} \left(w_0[i] + w_1[i] \right)^2 > 0 \Rightarrow$$

$$\text{if } w_0[i] = -w_1[i] \Rightarrow z^T(x^Tx)z = 0$$

$$w_0[i] = -w_1[i] \Rightarrow z_0 x_0[i] = -z_1 x_1[i] \Rightarrow x_0[i] = \alpha x_1[i]$$

$$\Rightarrow X \neq \text{full Rank matrix}$$

این اثبات را در حالت کلی هم می توان نشان داد :

$$z^T(x^Tx)z = \underbrace{(xz)^T}_{a} xz = a^Ta = \sum_{i=1}^{p} a_i^2 \geq 0 \Rightarrow$$

$$\text{for } a^Ta = 0 \rightarrow xz = 0 \longrightarrow X \neq \text{full Rank} \Rightarrow$$

$$\text{if } X = \text{full Rank} \Rightarrow x^Tx > 0$$

پس در کل برای آنکه بردار بتا تخمین زده شده نقطه کمینه در loss function را نمایش دهد ، باید دو شرط معکوس پذیر بودن x^Tx و full rank بودن ماتریس x برقرار باشد.
نزدیک ترین نقطه به y را برمی گرداند

پس در کل برای آنکه بردار بتا تخمین زده شده نقطه کمینه در loss function را نمایش دهد ، باید دو شرط معکوس پذیر بودن x^Tx و full rank بودن ماتریس x برقرار باشد.

به این ترتیب نشان دادیم که می توان بردار بتا را با استفاده از فرمول زیر تخمین زد :

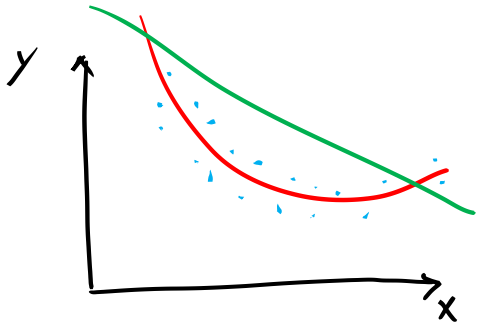
$$\hat{\beta} = (x^T x)^{-1} x^T y \Rightarrow \hat{y} = x\hat{\beta} = \underbrace{x(x^T x)^{-1} x^T}_{H} y = Hy$$

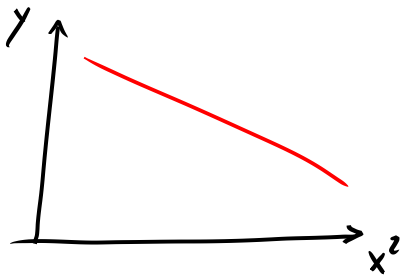
با ا ستفاده از این تعریف می توان ماتریس H را که با نام های Hat Matrix و Projection Matrix شناخته می شود معرفی کرد. علاوه بر این چون این ماتریس نزدیک ترین نقطه به y را در span(x) برمی گرداند (یعنی عمود اسـت) به آن Ortogonal projection matrix هم گفته می شود. به این ترتیب می توان خواص زیر را برای آن برشمرد:

$$\hat{y} = Hy \Rightarrow H\hat{y} = \hat{y} \Rightarrow \begin{cases} H^2 = H \\ H = H^T \end{cases}$$
$$H^2 y = Hy$$

Non-linear Regression

تمامی محاسـباتی که تاکنون بررسـی شـد مربوط به رگرسـیون خطی بود. حالت هایی وجود دارد که در آن ها یک رگرسـیون غیر خطی می تواند تخمین بهتر و خطای کمتری دا شته با شد. در شکل زیر م شخص ا ست که رگر سیون غیر خطی که با رنگ قرمز نشان داده شده است ، از رگرسیون خطی که با رنگ سبز مشخص شده است ، تخمین بهتری را برمی گرداند. به صورت شهودی می توان نتیجه گرفت که اگر در معادلات رگر سیون خطی جملات با درجه بالاتر به کار برده شود می توان مدل های منعطف تر و پیچیده تری تولید کرد. برای مثال در این شـکل با اضـافه کردن جمله درجه دو می توان رگرسـیون قرمز رنگ را تولید کرد. این درحالی است که در فضای درجه دو هنوز رابطه میان y و ترم درجه دو خطی است.





$$Y \approx \beta_0 + \beta_1 X + \beta_2 X^2 \Rightarrow X = \begin{bmatrix} 1 & x[1] & x^2[1] \\ \vdots & \vdots & \\ 1 & x[n] & x^2[n] \end{bmatrix}$$

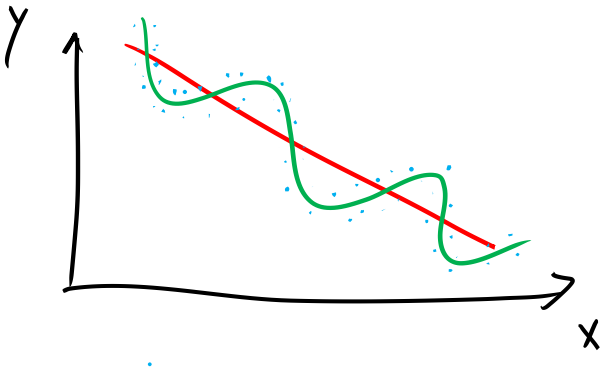
به این ترتیب می توانیم در حالت کلی یک polynomial regression را به فرم زیر تعریف کرد:

$$Y \approx \beta_0 + \beta_1 X + \beta_2 X^2 + \dots + \beta_p X^p$$

مشخص است که اگر این مدل را M_2 بنامیم می توان به خطای کمتری از حالت M_1 بدست آورد.

$$L(M_2) < L(M_1)$$

نکته ای که در اینجا وجود دارد این اســـت که اگر P را افزایش بدهیم ، انعطاف پذیری مدل در تطبیق ما داده های train بالا می رود و training error کاهش می یابد؛ اما ممکن است این موضوع برای prediction اصلا مناسب نباشد.



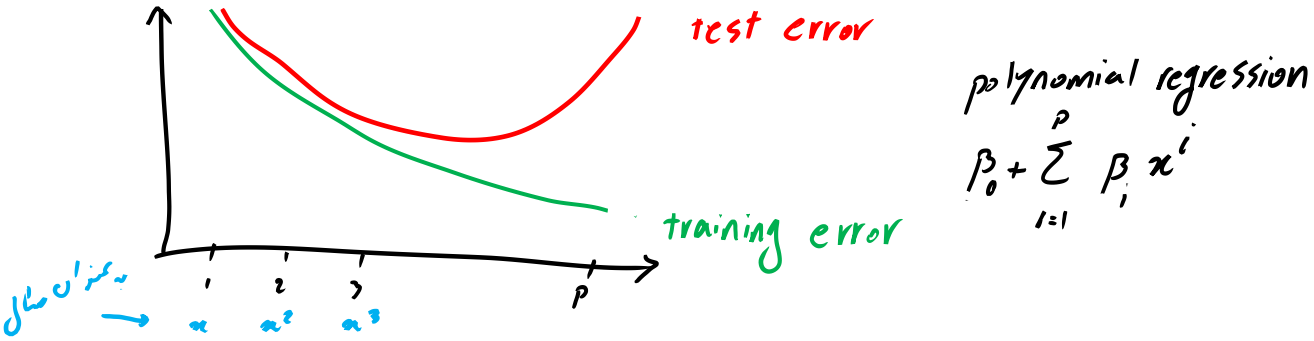
به عبارت دیگر ممکن است با پیچیده شدن مدل مشکل overfitting اتفاق بیافتد.
آنچه که بیان شد را می توان به صورت کلی تر نیز نمایش داد. به عبارت دیگر از basis function ها برای ترم های دیگر استفاده کنیم می توانیم یک non-linear regression را در حالت کلی تر داشـــته باشـــیم که ترم های آن در واقع توابعی از بردارها هستند. بنابراین می توان رگرسیون را به شکل زیر تعریف کرد:

$$Y \approx \beta_0 + \beta_1 h_1(X) + \beta_2 h_2(X) + \dots + \beta_m h_m(X)$$

$$X \to R^p (x_1, \dots, x_p)$$

موضوع بعدی مربوط به overfitting و generalization است. در واقع آنچه که برای ما مهم است ، پیدا کردن مدلی است که بتواند prediction بهتری روی داد هایی که هنوز دیده نشده انجام دهد و نیز تعمیم بهتری پیدا کند به گونه ای که خطای تعمیم کمی داشته باشد. این موضوع وقتی به وقوع می پیوندد که مدل به اندازه کافی پیچیده باشد تا شرط لازم برای تعمیم یافتگی وجود

داشته باشد. اما در این حالت مسئله overfitting می تواند مشکل ساز باشد. پس کاری که در این شرایط باید انجام داد ، ایجاد ساز و کارهایی است که از overfitting جلوگیری کند. از آنجا که خطای تعمیم روی داده های هنوز دیده نشده اتفاق می افتد راه کار دقیقی برای بد ست آوردن آن وجود ندارد و تنها می توان آن را تخمین زد. این کار می تواند با ا ستفاده از validation data صورت گیرد؛ که یعنی بخشی از داده ها را برای تست نگه داریم. راه کار دیگر استفاده از روش cross validation است که ساختار پیچیده تری دارد. شکل زیر خطای تعمیم و مفهوم overfitting را نمایش می دهد.



یکی دیگر از راه کارهای جلوگیری از overfitting استفاده از مفهوم regularized least squares کردن است. فرض کنید فرم زیر معادله رگرسیون ما را تشکیل دهد :

$$Y = \beta_0 + \beta_1 X + \beta_2 X^2 + \cdots + \beta_p X^p$$

بدیهی است که دراین حالت رگرسیون بسیار پیچیده خواهد بود و امکان overfitting بسیار محتمل می باشد. اما اگر ترم های بتا را صفر کنیم (به غیر از intercept) ، آن گاه مسئله دوباره به حالت بسیار ساده و حتی خطی کاهش پیدا می کند. به این ترتیب یک راه کار جالب می تواند کوچک نگه دا شتن ترم های بتا (مثلا تا نزدیک صفر) با شد. در این شرایط می توان از پیچیده شدن رگرسیون جلوگیری کرد. برای این کار کافی است با استفاده از یک متغیر trade off یک loss function جدید تعریف کرد که میزان پیچیدگی رگرسیون به صورت یک پنالتی در loss function ظهور می کند. تنها نکته مهم این است که در ترم پنالتی نباید هیچ پنالتی متوجه intercept با شد. چرا که در غیر این صورت خط رگرسیون همیشه از مبدا خواهد گذشت و این چیزی نیست که ما به دنبال آن باشیم.

$$\sum_{i=1}^{p} \beta_i^2 = \beta^T \beta \equiv \text{complexity of model} \qquad \beta_0 \text{ is not included}$$

$$L'(\beta) = \underbrace{(Y - X\beta)^T (Y - X\beta)}_{\text{fitting the data}} + \underbrace{\lambda \beta^T \beta}_{\text{penalize complexity}}$$

centered data

control trade off between fitness & complexity of model

Ridge Regression

به این ترتیب هدف ما عبارتند از :

$$\hat{\beta} = \arg\min_{\beta} \mathcal{L}(\beta) \qquad \beta^T = [\beta_1, \dots, \beta_p]$$

به این ترتیب می توان برای تابع loss جدید محاسبات را تکرار کرد :

$$\nabla_\beta \mathcal{L}(\beta) = \frac{\partial}{\partial \beta}\left(y^T y - 2\beta^T x^T y + \beta^T x^T x \beta + \lambda \beta^T \beta \right) = 0$$

$$\Rightarrow \quad -2x^T y + 2x^T x\beta + 2\lambda\beta = 0 \quad \Rightarrow$$

$$(x^T x + \lambda I)\beta = x^T y \Rightarrow \hat{\beta}_{ridge} = (x^T x + \lambda I)^{-1} x^T y$$

در این حالت دیگر لازم نیست که عبارت داخل پرانتز non-singular یا همان معکوس پذیر باشد. به عبارت دیگر می توان مقدار لاندا را به گونه تعیین کرد (به مقدار لازم بزرگ در نظر گرفت) که ماتریس حاصل همواره مشتق پذیر و معکوس پذیر باشد. به عبارت دیگر اگر تعداد مشاهدات از تعداد ویژگی ها خیلی کمتر باشد ، می توان نشان داد که ماتریس x^Tx مشتق پذیر نیست. علاوه براین می توان نشان داد که اگر مقادیر قطر اصلی یک ماتریس را از حد مشخصی بیشتر کنیم آن ماتریس مشتق پذیر خواهد شد. بنابراین با انتخاب درست مقدار لاندا می توان ماتریس مذکور را مشتق پذیر کرد.

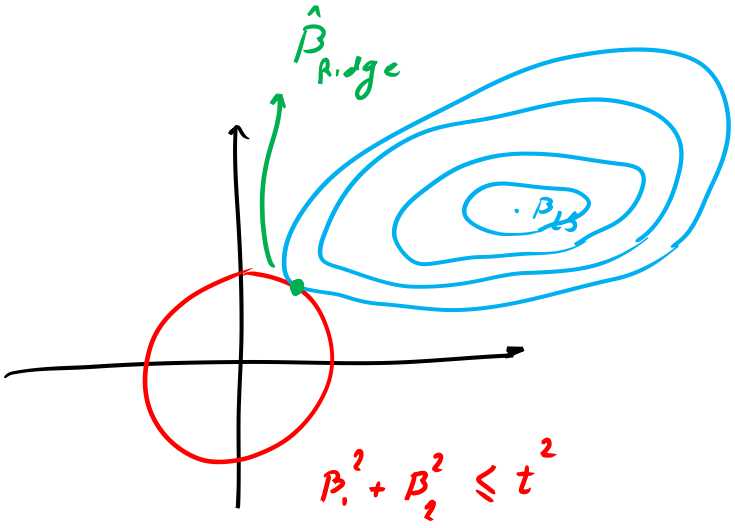
پس مرحله بعدی انتخاب مقدار لاندا است. مقدار لاندا عموما از طریق cross validation مشخص می شود. اما به صورت شهودی می توان مشاهده کرد که اگر مقدار آن را صفر در نظر بگیریم ، مقدار پنالتی را از بین برده و همان فرم linear regression را قبل را خواهیم داشت. در حالتی هم که مقدار آن را بسیار بزرگ در نظر بگیریم (بی نهایت) چون ترم پنالتی باعث صفر شدن تمام ترم های بتا می شود پس تخمین ما از y ($\hat{y}$) همان میانگین y ها ($\bar{y}$)می شود.

این حالت را می توان از نگاه هندسی هم مورد بحث قرار داد. اگر فرمول های بیان شده را به فرم زیر بنویسیم :

$$\hat{\beta}_{Ridge} = \arg\min_{s.t} (y - X\beta)^T (y - X\beta)$$

$$\text{subject to} : \sum_{j=1}^{p} \beta_j^2 \leq t$$

در این حالت می توان نشان داد که نقطه مورد نظر ما نقطه تلاقی contour plot و دایره constraint است. به عبارت دیگر ما به دنبال نقطه کمینه در تابع loss نیستیم. چرا که این نقطه ممکن است نقطه ای باشد که over fit در آن اتفاق افتاده باشد. به این ترتیب ما با اعمال یک محدودیت یا پنالتی سعی می کنیم از آن نقطه فا صله بگیریم و شهودی هند سی آن را می توان به فرم زیر نشان داد.



تا کنون سعی کردیم راه هایی را معرفی کنیم که پیچیدگی مدل را کنترل کنیم. موضوع بعدی بررسی درجه آزادی موثر مدل است که در واقع همان پیچیدگی مدل را نشان می دهد. می توان نشان داد که درجه آزادی یک مدل برابر است با تریس Hat matrix.
به این ترتیب داریم :

$$\hat{y} = Hy \Rightarrow \text{Degree of freedom (Dof)} = tr(H)$$

linear regression

$$Dof = tr(H) = tr\left(X(X^TX)^{-1}X^T\right)$$

if AB = squar matrix then $tr(AB) = tr(BA)$

$$\Rightarrow tr\left((X^TX)^{-1}X^TX\right) = tr(I_{p\times p}) = p$$

در این تعریف intercept لحاظ نشده است که با احتساب آن درجه آزادی P+1 می شود.
مشابه همین محاسبات را می توان برای Ridge در نظر گرفت که به این ترتیب داریم :

$$Dof(\lambda) = tr\left(X\left(X^TX + \lambda I\right)^{-1}X^T\right)$$

برای حساب کردن تریس در این حالت نیاز است که مفهوم singular value decomposition را بررسی کنیم. این مفهوم بیان می کند که هر ماتریس را می توان توسط حاصلضرب سه ماتریس به گونه ای نشان داد که :

$$X_{n\times p} = U_{n\times n}\ D_{n\times p}\ V_{p\times p} \qquad U^TU = I \qquad V^TV = I \qquad D = \begin{bmatrix} d_{11} & & \\ & d_{22} & \\ & & d_{pp} \end{bmatrix}$$

(البته فرمول های نوشته شده مال حالت خاصی است که در آن ها ماتریس U مربعی نیست ولی ماتریس های D و V مربعی هستند)

در این تعریف ماتریس D یک ماتریس قطری است که به مقادیر روی قطر آن singular value گفته می شود. می توان نشان داد که با استفاده از مفهوم SVD ، DOf(λ) برابر است با :

$$Dof(\lambda) = \sum_{i=1}^{p} \frac{d_{ii}^2}{d_{ii}^2 + \lambda}$$

این تعریف نشان می دهد که اگر مقدار لاندا را صفر در نظر بگیریم درجه آزادی همان P می شود و اگر مقدار آن را خیلی زیاد در نظر بگیریم درجه آزادی به سمت صفر میل می کند.

یکی دیگر از مدل هایی که به دلیل خیلی بزرگ بودن مقدار ویژگی ها از تعداد مشاهدات مورد بحث قرار می گیرد مدل lasso است. در این مدل ترم پنالتی نرم اول بردار بتا است. چون نرم اول را می توان معادل قدر مطلق در نظر گرفت ، مشتق نمی توان برای آن محاسبه کرد. به این ترتیب در این مدل راه حل analytical وجود ندارد و از راه حل های convex optimization استفاده می کنیم.

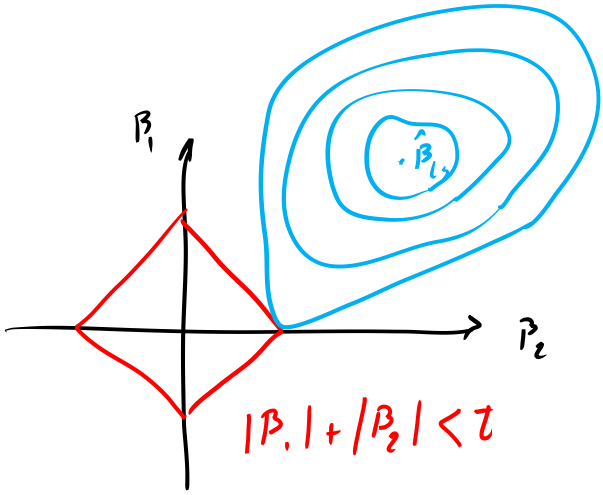
$$\mathcal{L}(\beta) = (y - X\beta)^T(y - X\beta) + \lambda \|\beta\|_1 \qquad \|\beta\|_1 = \sum_{i=1}^{p} |\beta_i|$$

در این حالت هم می توان تفسیر هندسی برای این مدل در نظر گرفت که در شکل نشان داده شده است. نکته ای که در این مدل وجود دارد این ا ست که می توان ن شان داد نقطه مورد نظر در گو شه های لوزی بد ست می آید. این یعنی اینکه یکی از ویژگی ها

صفر خواهد شد. از این نکته می توان در PCA ا ستفاده کرد. در این مدل درجه آزادی برابر ا ست با تعداد ترم های non zoro بردار بتا.

$$\arg\min_{\beta} (y - X\beta)^T (y - X\beta)$$

$$s.t \quad \|\beta\| < t \equiv |\beta_1| + |\beta_2| < t$$



بحث بعدی در ارتباط با خطای تعمیم است. آنچه که ما داریم یک مدل محاسباتی است که بر مبنای داده های train آموزش داده شده ا ست. اما آنچه که برای ما اهمیت دارد این ا ست که این داده ها چقدر روی داده های دیده نشده خوب می توانند عمل کنند. مشکل در اینجاست که خروجی ما روی داده های دیده نشده حکم یک پیش بینی را دارد. به این ترتیب نمی توانیم خطای داده های دیده نشده را حساب کنیم. کاری که می شود کرد این است که روی بخشی از داده های موجود مرحله یادگیری را انجام داد و روی بخش دیگری از داده ها که باقی مانده است، خطای مرحله پیش بینی را اندازه گرفت. بدیهی است که چون داده ها و نوع توزیع آن ها را نمی دانیم ، باز هم خطای در نظر گرفته شده در این حالت نمی تواند معادل خطای واقعی باشد. به همین دلیل خطای حاصل یک میانگین یا امید ریاضی خطای اصلی است.

کاری که در این مرحله می توان انجام داد این ا ست که داده ها را به چند د سته یا fold تقسیم کنیم. فرض کنید داده ها را به n دسته تقسیم کردیم. سپس در هر بار از اجرا یک دسته را برای تست و n-1 دسته باقی مانده را برای آموزش مورد استفاده قرار می گیریم. به این ترتیب ما n خطای متفاوت خواهیم داشت. اگر میانگین خطا های بدست آمده را در نظر بگیریم ، آنگاه cross validation انجام داده ایم.

یک نکته بسیار مهم در ا ستفاده از CV نوع ا ستفاده از داده ها ا ست. به عبارت دیگر وقتی ما داده ها را به چند د سته تقسیم می کنیم و یک د سته را برای تست در نظر می گیریم. این د سته نباید در هیچ یک از مراحل آموزش و ساخت مدل مورد استفاده قرار بگیرند. به عبارت دیگر باید کاملا برای ما ناشناخته باشند. مثلا اگر بخواهیم از داده های خود ویژگی استخراج کنیم ، ابتدا باید

دســته مربوط به تســت را خارج کرده و از دســته های باقی مانده ویژگی اســتخراج کنیم. ســپس میزان خطای مدل را از داده های دسته خارج شده حساب نماییم. به این ترتیب اگر ویژگی ها را با استفاده از تمام داده ها انجام دهیم و بعد دسته ها را تشکیل داده و مراحل تست و یادگیری را انجام دهیم ، خطای بدست آمده ، دارای دقت و اعتبار علمی نخواهد بود.