سوال ۱ –

قسمت آ)

$$E_{out}(g^{(D)}) = E_X\left[\left(g^{(D)}(x) - f(x)\right)^2\right]$$

$$\rightarrow E_D[E_{out}(g^{(D)})] = E_D\left[E_X\left[\left(g^{(D)}(x) - f(x)\right)^2\right]\right] = E_X\left[E_D\left[\left(g^{(D)}(x) - f(x)\right)^2\right]\right]$$

$$\rightarrow if\ average\ hypothesis: \bar{g}(x) = E_D[g^{(D)}(x)]$$

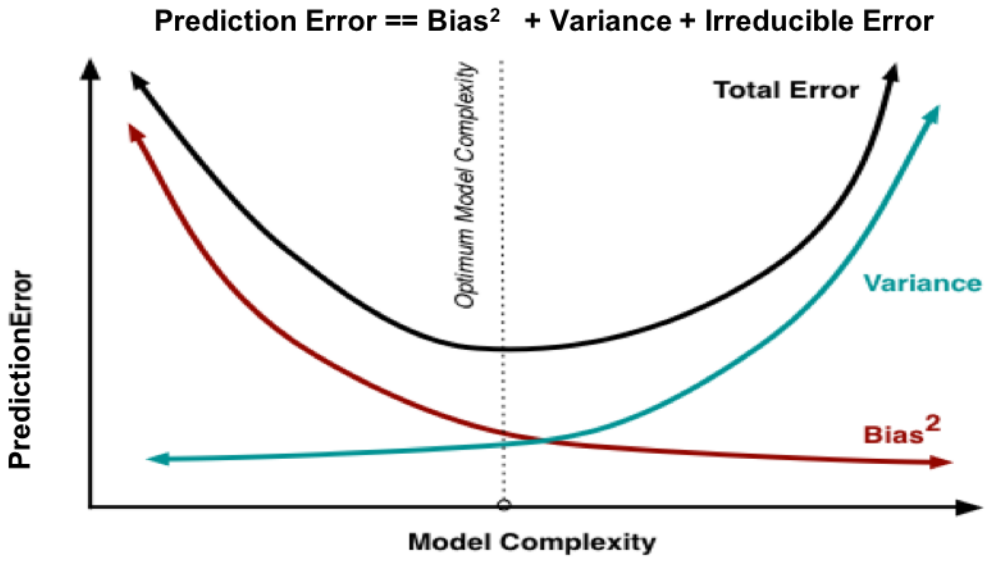
$$\rightarrow E_D\left[\left(g^{(D)}(x) - f(x)\right)^2\right] = E_D\left[\left(g^{(D)}(x) - \bar{g}(x) + \bar{g}(x) - f(x)\right)^2\right]$$

$$= E_D\left[\left(g^{(D)}(x) - \bar{g}(x)\right)^2 + \left(\bar{g}(x) - f(x)\right)^2 + 2\left(g^{(D)}(x) - \bar{g}(x)\right)\left(\bar{g}(x) - f(x)\right)\right]$$

$$= E_D\left[\left(g^{(D)}(x) - \bar{g}(x)\right)^2\right] + \left(\bar{g}(x) - f(x)\right)^2 + 0$$

$$\rightarrow E_D[E_{out}(g^{(D)})] = E_D\left[\left(g^{(D)}(x) - \bar{g}(x)\right)^2\right] + \left(\bar{g}(x) - f(x)\right)^2 = Variance + Bias$$

قسمت ب) منبع : تصاویر گوگل



قسمت پ) همانطور که میدانیم، بایاس و واریانس رابطه معکوس دارند. به مدلی که بایاس زیاد و واریانس کم دارد اصطلاحا مدلی میگویند که underfitting شده. برای جلوگیری از underfitting :

راه اول: اضافه کردن ویژگی های مرتبط(relevant feature) و داده های آموزشی(training data points) باعث افزایش پیچیدگی مدل و در نتیجه کاهش بایاس میشود. یعنی بهترین مدلی که میتوانیم یاد بگیریم را به تابع هدف نزدیک تر میکند. زیرا میدانیم فضای فرضیه کوچک باعث افزایش بایاس و کاهش واریانس مدل میشود و با افزایش فضای فرضیه و داده هایمان، این اثر را معکوس میکنیم.

راه دوم: حذف نویز از داده ها و کاهش outlier ها

راه سوم: افزایش epoch و یا افزایش زمان training

راه چهارم: کاهش regularization میتواند باعث کم شدن generalization مدل و افزایش overfitting و در نتیجه کاهش underfitting شود.

قسمت ت) با کم کردن feature ها ، پیچیدگی مدل کم میشود و مدلی که یاد گرفته میشود به نوعی smooth میشود لذا واریانس کم میشود(جلوگیری از overfit). از طرفی همانطور که گفته شد، بخاطر کم بودن تعداد feature ها ، بهترین مدلی که میتوانستیم یاد بگیریم از تابع هدف واقعی دور میشود یعنی $\left(\bar{g}(x) - f(x)\right)^2$ که همان بایاس است، زیاد میشود. یعنی با کم کردن تعداد feature ها، underfitting رخ میدهد(بهتر است بگوییم جلوگیری از overfitting و حرکت به سمت underfitting در صورت حذف تعداد زیادی feature رخ میدهد).

قسمت ث) افزایش بیش از اندازه واریانس یعنی overfitting شدن مدل. برای کاهش آن:

راه اول: استفاده از cross-validation که باعث tune شدن hyperparameter ها شده و در نتیجه از overfitting جلوگیری میشود یعنی به کم شدن واریانس کمک میکند.

راه دوم: افزایش مقدار داده های آموزشی: توجه شود در قسمت های قبل، برای کاهش underfitting هم این مورد بیان شد. دلیل آن است که افزایش داده ها به بهتر fit شدن مدل کمک کرده و در نتیجه مدل را به سمتی میکشاند که هم از underfitting و هم از overfitting دوری میکند.

راه سوم: همان طور که قبلا گفته شد، کم کردن تعدادfeature ها به کم کردن پیچیدگی مدل و در نتیجه کاهش واریانس و دوری از overfitting کمک میکند.

راه چهارم: استفاده از تکنیک early stopping که کمک میکند آموزش در بهترین لحظه متوقف شود و از افزایش واریانس جلوگیری شود.

راه پنجم: استفاده از تکنیک regularization که به ساده تر شدن مدل و در نتیجه کاهش واریانس کمک میکند.

قسمت ج) برای مثال مسئله رگراسیون یعنی یادگیری یک بردار وزن از داده های آموزشی و استفاده از آن برای پیش بینی داده های جدید است. فرمول بدست آوردن این بردار برابر است با:

$$W = (X^T X)^{-1} X^T y$$

فرض عمومی برای y این است که متغیری باتوزیع نرمال و واریانس σ^2 است، لذا واریانس این بردار وزن برابر است با:

$$\sigma^2 (X^T X)^{-1}$$

برای اینکه مدل به اندازه کافی پایدار باشد باید این مقدار واریانس کم باشد. با افزایش این مقدار واریانس، اتفاقی که می افتد این است که مدل به داده های ورودی بسیار حساس میشود. طبق چیزی که از جبر خطی میدانیم(قاعده singular value decomposition):

$$X = USV^T$$

ماتریس S یک ماتریس نامنفی قطری است. با استفاده از این قاعده داریم:

$$var[w] = \sigma^2 (X^T X)^{-1} = \sigma^2 V S^{-2} V^T$$

وقتی داده های دو به دو هم‌بسته داشته باشیم، مقادیر درون ماتریس S کوچک میشوند و بخاطر توان منفی آن، باعث افزایش واریانس بردار وزن میشوند.

پس پاسخ نهایی به سوال میشود: داده های هم‌بسته باعث افزایش واریانس مدل و ناپایداری آن میشود.

سوال ۲ – با توجه به جلسات کلاس، میدانیم که:

$$w_{MLE} = (X^T X)^{-1} X^T y \quad ,$$
$$E_{out} = (h(x) - E[f(x|D)])^2 + E[(f(x|D) - E[f(x|D])^2] = bias + variance$$

هم چنین میدانیم که $h(x) = X^T W$ و $y = XW$. لذا برای تحلیل بایاس داریم:

$$bias = (h(x) - E[f(x|D)])^2 = (X^T W - E_D[X^T W_{MLE}])^2$$
$$= (X^T W - E_D[X^T (X^T X)^{-1} X^T X W])^2 = 0$$

اثبات میشود که حتی اگر نویزی به y اضافه کنیم، بایاس باز هم صفر میشود. لذا وزن ***MLE*** به عبارتی ***unbiased*** است.

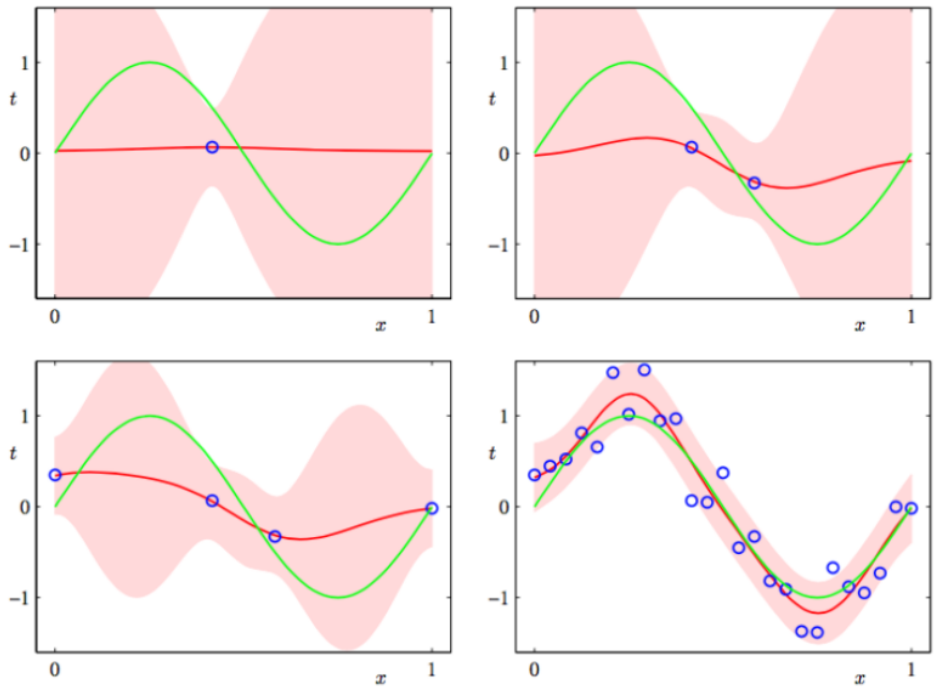
اما در حالتی که نویز اپسیلون با توزیع گاوسین با میانگین صفر و واریانس ۱ داشته باشیم، واریانس دیگر صفر نمیشود و داریم:

$$Variance = E[(X^T(X^TX)^{-1}X^T\varepsilon)^2] = E[X^T(X^TX)^{-1}X^T\varepsilon\varepsilon^T(X^T(X^TX)^{-1}X^T)^T]$$

$$= X^T(X^TX)^{-1}X^TE[\varepsilon\varepsilon^T](X^T(X^TX)^{-1}X^T)^T = var(\varepsilon)X^T(X^T(X^TX)^{-1})^T$$

$$= \frac{var(\varepsilon) * dimension}{number\ of\ data}$$

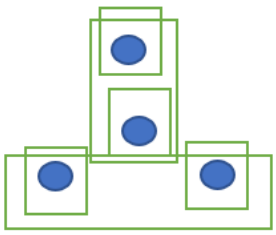
لذا استفاده از روش گاوسین یک روش unbiased است و واریانس آن نیز به تعداد داده ها بستگی دارد و با افزایش تعداد داده، واریانس کاهش می یابد. با استفاده از شبیه سازی ای که در کتاب bishop انجام شده است، میتوان به این امر نیز پی برد:



سوال ۳ – میدانیم که بعد VC یکی کمتر از نقطه شکست یک مدل است یعنی VC-dimension = breakpoint -1

لذا در این مسئله سعی در بدست آوردن نقطه شکست(breakpoint) میکنیم و سپس از روی آن، VC را بدست می آوریم.

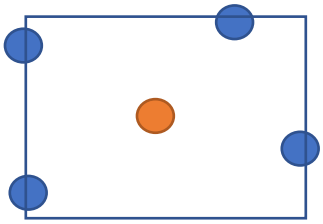
قسمت آ – ۱ : با توجه به تعریف بالا، میگوییم وجود دارد ۴ نقطه ای که بتوانیم همه حالت های آن را ایجاد کنیم و دسته بند آن هارا دسته بندی کند. برای مثال:



میبینیم که هر حالتی از برچسب گذاری آن هارا بخواهیم میتوانیم توسط این دسته بند ایجاد کنیم.

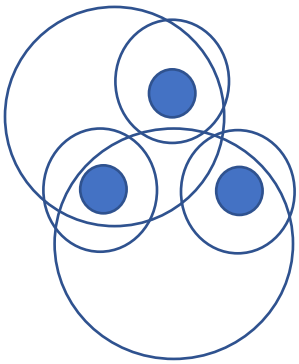
اما هیچ ۵ نقطه ای وجود ندارد که بتوان آن هارا در صفحه چید و دسته بند بتواند همه حالت های آن را دسته بندی کند. به این صورت اثبات شهودی میکنیم که با ۴ نقطه که از مرکز ثقل نقاط دور تر هستند، یک مستطیل با شرایط مسئله میسازیم(نقاط روی ضلع ها) و نقطه بعدی که نزدیک ترین به مرکز ثقل هست را برچسب مخالف با بقیه نقاط میزنیم.

یعنی همچین شکلی میشه:

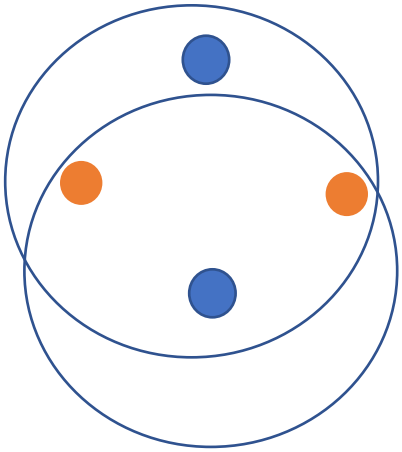


پس میفهمیم break point = 5 است. لذا VC-dimension = 5 – 1 = 4

قسمت آ – ۲ : در مورد دایره ها، واضح است که سه نقطه را میتوانیم داشته باشیم که هر دایکوتومی اش را بتوانیم داشته باشیم:



اما هیچ ۴ نقطه ای را نمیتوانیم تمام دایکوتومی هایش را داشته باشیم:



لذا *VC-dimension=3* است.

قسمت آ – ۳ : چون فضا d بعدی است، فرض کنید d نقطه داریم که هر کدام درایه متناظر با یک بعد آن ها ۱ و در بقیه بعد ها صفر باشند. یعنی داده x_j در درایه jام ۱ داشته باشد و مابقی صفر باشند. این نقاط توسط ابرصفحه های گذرنده از مبدا مانند دسته بند زیر ***shattered*** میشوند:

$$f(x) = sign\left(\sum_{i=1}^{d} y_i x_i^T x\right)$$

تابع وزن این دسته بند نیز $w = \sum_{i=1}^{d} y_i x_i$ است. زیرا مقدار $x_i^T x_j$ وقتی i و j برابر نباشند صفر میشود و وقتی برابر باشند ۱ میشود. لذا نتیجه میشود که حداقل ***vc-dimension*** برابر است با d .

حال از یک برهان خلف استفاده میکنیم. فرض میکنیم $d+1$ نقطه داریم که توسط یک دسته بند بتوانند در این فضا ***shattered*** شوند یعنی تمام برچسب گذاری های ممکن این $d+1$ نقطه بتوانند تولید شوند یعنی بردار وزنی مانند w_l وجود دارد که $f_l(x) = sign(w_l^T x)$ میتواند این برچسب هارا تولید کند. حال یک ماتریس برای تجمیع خروجی های این دسته بند را در نظر میگیریم:

$$H = \begin{bmatrix} w_1^T x_1 & \cdots & w_{2^{d+1}}^T x_1 \\ \vdots & \ddots & \vdots \\ w_1^T x_{d+1} & \cdots & w_{2^{d+1}}^T x_{d+1} \end{bmatrix} = XW$$

که $X = [x_1, \dots, x_{d+1}]^T$ و $W = [w_1, \dots, w_2^{d+1}]$ است.

از طرفی هر حالت برچسب گذاری توسط ستون های ***sign(H)*** نمایش داده میشوند و از طرف دیگر سطر های H مستقل خطی هستند چون هیچ بردار a غیر صفری وجود ندارد که باعث $a^T H = 0^T$ شود. وقتی میگوییم $d+1$ سطر H مستقل خطی هستند یعنی رنک این ماتریس $d+1$ است و از طرف دیگر گفته شد که $H = XW$ لذا رنک H باید کوچک تر مساوی مینیموم رنک X و W باشد لذا باید کوچک تر مساوی d باشد. لذا به تناقض رسیدیم. پس نتیجه میگیریم که $d+1$ نقطه نمیتوانند ***shattered*** شوند پس ***vc-dimension*** باید کوچک تر مساوی d باشد. از تمام این حرفها میشود نتیجه گرفت که ***VC-dimension = d*** است.

قسمت ب – میدانیم که:

$$\boldsymbol{N \geq \frac{8}{\mathcal{E}^2} ln\left(\frac{4\mathcal{M}_{\mathcal{H}}(2N)}{\delta}\right)}$$

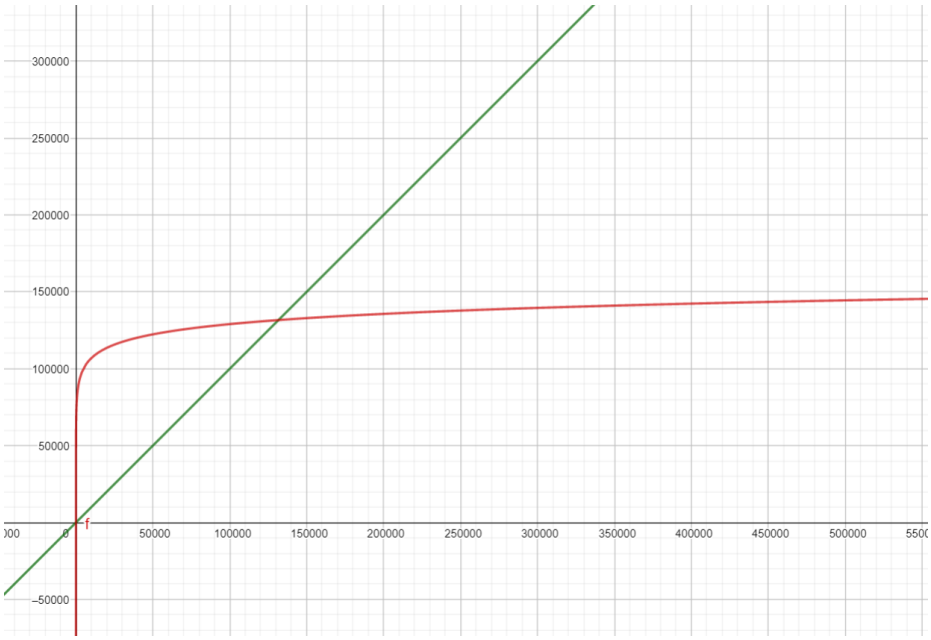
هم چنین میدانیم که:

$$\mathcal{M}_{\mathcal{H}}(2N) \in O((2N)^{d_{vc}} + 1)$$

لذا با جایگذاری $\varepsilon = 0.05 , \delta = 0.1$ داریم:

$$N \geq \frac{8}{0.05^2} ln \left(\frac{4((2N)^3 + 1)}{0.1} \right) \rightarrow N \geq 132000$$

برای بدست آوردن جواب نامساوی بالا نیز از رسم توابع استفاده کردم:



قسمت پ – وقتی میگوییم با احتمال ۹۰ درصد فلان دقت را داشته باشیم، منظور از احتمال همان اصطلاح *confidence* است یعنی اگر آزمایش را به دفعات زیاد انجام دهیم(با ۱۳۲ هزار نمونه حداقل)، در چند آزمایش(در چه درصدی از تعداد آزمایش ها) این دقت بدست نمی آید و به عبارتی *violate* میشود. پس برای نشان دادن آن، به تعداد بالا آزمایش را تکرار میکنیم و *generalization error* را محاسبه میکنیم. سپس درصدی از آزمایش هارا که این *error* بیشتر از ۵ درصد اتفاق افتاده را بدست می آوریم. در صورت صحت مسئله، باید این درصد، کمتر مساوی ۱۰ درصد باشد که *confidence* هه ۹۰ درصد را داشته باشیم. در مورد خود آزمایش هم اینکار را انجام میدهیم که همان تعداد گفته شده نمونه آزمایشی رندم را توسط یک مدل دایره ای(وقتی مدل یاد گرفته میشود، طبیعتا بهترین دایره ممکن در فضای فرضیه انتخاب میشود) در فضای دو بعدی دسته بندی میکنیم و سپس کار های گفته شده قبلی را روی برچسب های دسته بند و برچسب های واقعی انجام میدهیم.

سوال ۴ –

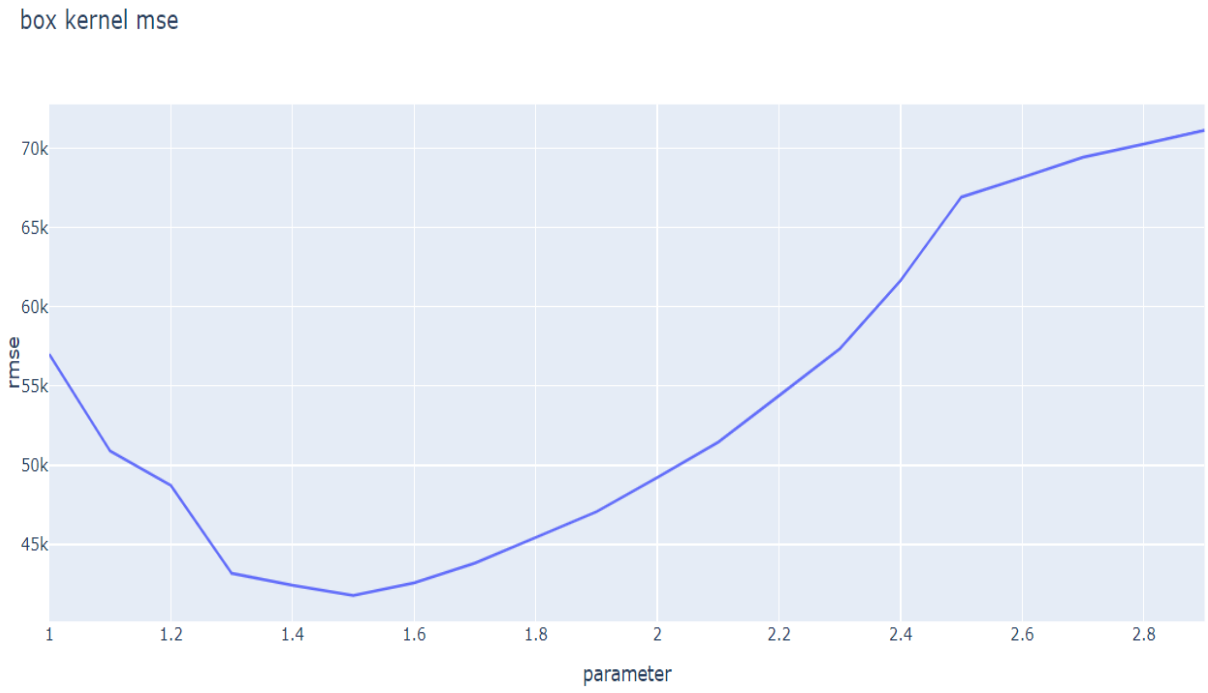
قسمت آ – به ازای بردار x_j که میخواهیم برچسب آن را پیش بینی کنیم، کرنل های $K(x_j , x_i)$ را محاسبه کرده که در این رابطه x_i ها همان سطر های داده آموزشی ما هستند. سپس وزن متناظر با آن سطر آموزشی برابر است با:

$$K(x_j , x_i) = \begin{cases} 1 & if \ \|x_j , x_i\| \leq h \\ 0 & otherwise \end{cases} \qquad w_i = \frac{K(x_j , x_i)}{\sum_{n=1}^{N} K(x_j , x_n)}$$

سپس برچسب تخمین زده شده برابر است با:

$$y_j = \sum_{i=1}^{N} x_i w_i$$

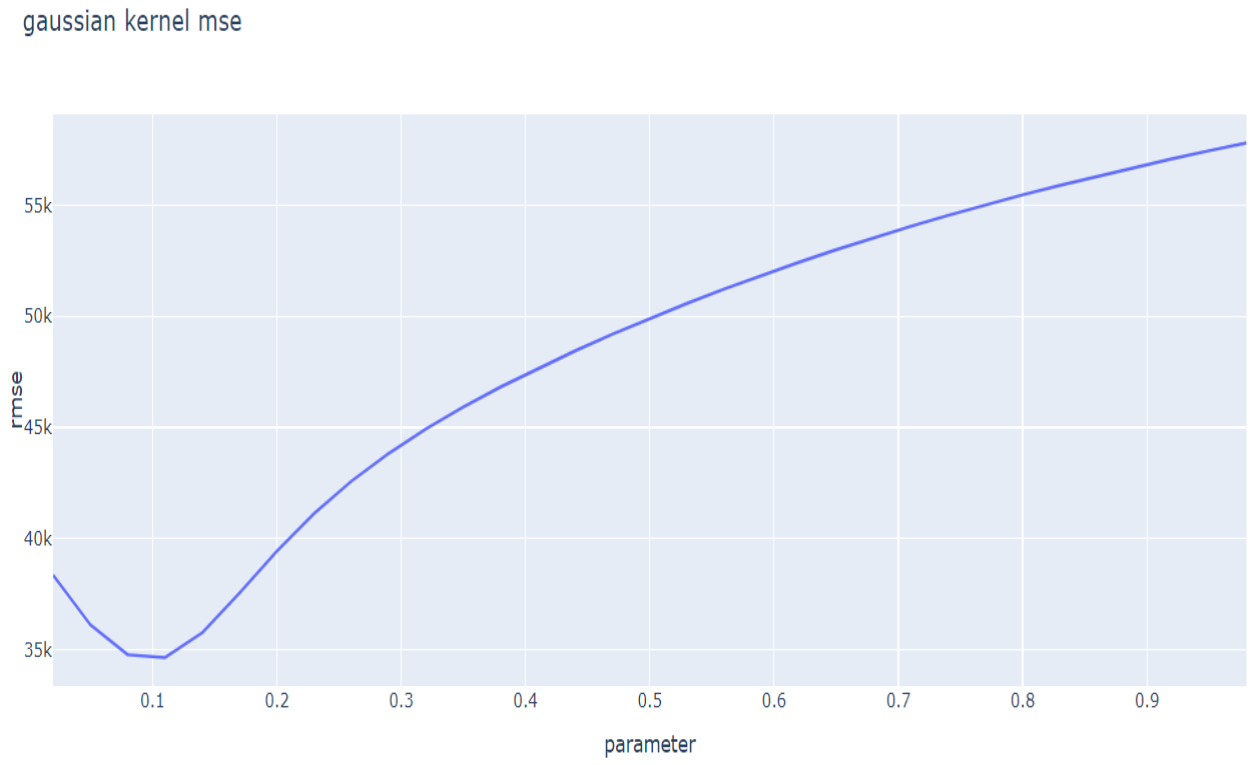
با پیاده سازی این روش با h های مختلف(کد های زده شده در فایل زیپ موجود است)، به نمودار زیر رسیدیم. همانطور که دیده میشود، با افزایش h از مقدار ۱، ابتدا $RMSE$ کم شده تا به یک نقطه کمینه میرسد و سپس دوباره افزایش پیدا میکند. پس افزایش مقدار h تا یک جایی اثر مثبتی دارد اما از یک جایی به بعد، اثر منفی میگذارد:



قسمت ب – ۱ : در این قسمت، تنها چیزی که تغییر میکند، رابطه کرنل ما است که هسته گوسی دارد یعنی همه چیز با قسمت آ برابر است به جز:

$$K(x_j, x_i) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}*\frac{\|x_j, x_i\|}{\sigma}}$$

با پیاده سازی این روش با سیگما های مختلف ، به نمودار زیر رسیدیم. همانطور که دیده میشود مانند قسمت قبل، با افزایش مقدار سیگما از صفر، ابتدا *RMSE* کم شده تا به یک نقطه کمینه میرسد و سپس دوباره افزایش پیدا میکند. پس افزایش مقدار سیگما تا یک جایی اثر مثبتی دارد اما از یک جایی به بعد، اثر منفی میگذارد. و همچنین با توجه به مقدار *RMSE* در نقطه بهینه و با مقایسه این مقدار در قسمت قبل، میبینیم که هسته گوسی به خطای کمتری دست پیدا میکند، لذا برای مسئله ما، بهتر عمل میکند:



قسمت ب – ۲ : در این هسته ، سیگما همان *standard deviation* کرنل است و از آنجایی که با افزایش تعداد ویژگی ها، پیچیدگی مدل و لذا واریانس مدل افزایش پیدا میکند، پس هرچه تعداد ویژگی بیشتری داشته باشیم ، سیگمای بهینه عدد بزرگ تری میشود و بالعکس.پس تعداد ویژگی با سیگما رابطه مستقیم دارد.طبق منبع *36-402/36-608* بهترین مقدار سیگما طبق رابطه زیر بدست می آید:

$$\sigma^* = \frac{C_1}{2C_2 n^{1/3}}$$

قسمت ب – ۳ (منبع لکچر ۷ *STAT/Q SCI 403* از *Yen-chi chen*) ابتدا یک سری تعریف اولیه انجام میدهیم و بایاس و واریانس کلی را(فارغ از نوع هسته) بدست می آوریم و در پایان با جایگذاری پارامتر های مربوطه، بایاس و واریانس هسته گوسی را بدست می آوریم.

در این نوع مسئله ها ما باید توزیع داده های آموزشی را بدست آوریم و از روی آن برچسب داده آزمایشی را پیش بینی کنیم. به این نوع مسئله ها *KDE : kernel density estimator* میگوییم. فرمول کلی *KDE* برابر است با :

$$\widehat{p_n}(x) = \frac{1}{nh}\sum_{i=1}^{n} K\left(\frac{X_i - x}{h}\right)$$

به تابع *K* تابع هسته میگویند و *h* نیز *smoothing bandwidth* نام دارد. X_i ها نیز همان سطر های دیتاست آموزشی و *x* نیز سطری از دیتاست آزمایشی است که قرار است برچسب آن را پیش بینی کنیم. *n* یک عبارت بسته به نوع هسته است که در هسته های مختلف متفاوت است.

فرض میکنیم *X1,…,Xn* نمونه های *IID* با توزیع *p* هستند. برای سادگی فرض میکنیم میخواهیم برچسب نقطه x_0 را به وسیله $\widehat{p_n}(x_0)$ تخمین بزنیم. ابتدا **بایاس** را تحلیل میکنیم:

$$E\left(\widehat{p_n}(x_0)\right) - p(x_0) = E\left(\frac{1}{nh}\sum_{i=1}^{n} K\left(\frac{X_i - x_0}{h}\right)\right) - \ p(x_0)$$

$$= \frac{1}{h}E\left(K\left(\frac{X_i - x_0}{h}\right)\right) - \ p(x_0)$$

$$= \frac{1}{h}\int K\left(\frac{x - x_0}{h}\right) p(x)dx - \ p(x_0)$$

حال تغییر متغیر $y = \frac{x-x_0}{h}$ میدهیم لذا $dy = dx/h$ پس:

$$E\left(\widehat{p_n}(x_0)\right) - p(x_0) = \int K(y)p(x_0 + hy)dy - \ p(x_0) \qquad (x = x_0 + hy)$$

حال با استفاده از بسط تیلور، وقتی h کوچک باشد داریم:

$$p(x_0 + hy) = p(x_0) - hy \cdot p'(x_0) + \frac{1}{2}h^2y^2p''(x_0) + o(h^2)$$

لذا داریم:

$$E\left(\widehat{p_n}(x_0)\right) - p(x_0) = \int K(y)p(x_0 + hy)dy - p(x_0)$$

$$= \int K(y)\left[p(x_0) - hy \cdot p'(x_0) + \frac{1}{2}h^2y^2p''(x_0) + o(h^2)\right]dy - p(x_0)$$

$$= \int K(y)p(x_0)dy + \int K(y)hy \cdot p'(x_0)dy + \int K(y)\frac{1}{2}h^2y^2p''(x_0)dy + o(h^2) - p(x_0)$$

$$= p(x_0)\int K(y)dy + hp'(x_0)\int y\,K(y)dy + \frac{1}{2}h^2p''(x_0)\int y^2K(y)dy + o(h^2) - p(x_0)$$

$$= p(x_0) + \frac{1}{2}h^2p''(x_0)\int y^2K(y)dy + o(h^2)$$

$$= \frac{1}{2}h^2p''(x_0)\int y^2K(y)dy + o(h^2)$$

حال با جایگذاری y داریم:

$$\boldsymbol{Bias\left(\widehat{p_n}(x_0)\right)} = \frac{1}{2}h^2p''(x_0)\int \left(\frac{x - x_0}{h}\right)^2 K\left(\frac{x - x_0}{h}\right) d\left(\frac{x - x_0}{h}\right) + o(h^2)$$

حال برای محاسبه **واریانس**، نیز با تحلیل های مشابه و دقیقا حتی تغییر متغیر های مشابه به مقدار زیر میرسیم:

$$\boldsymbol{Var\left(\widehat{p_n}(x_0)\right)} = \frac{1}{\sqrt{2\pi}h}p(x_0)\int K^2\left(\frac{x - x_0}{h}\right) d\left(\frac{x - x_0}{h}\right) + o\left(\frac{1}{\sqrt{2\pi}h}\right)$$

قسمت پ –

<u>مزایای هسته گوسی نسبت به هسته *box*</u>: ۱- توزیع گوسی بهتر از توزیع یونیفورم پدیده های طبیعی را مدل میکند ۲- دقت این هسته وابستگی کمتری نسبت به سایز هسته دارد ۳- اگر داده ها ***dense*** باشند، هسته گوسی بهتر عمل میکند ۴- حساسیت هسته گوسی به تعداد نمونه های ***training*** کمتر است.

<u>مزایای هسته box نسبت به هسته گوسی</u>: ۱- محاسبات ساده تر با پیچیدگی کمتر ۲- حساسیت کمتر نسبت به تراکم داده های آموزشی و توزیع آن ها

قسمت ت – داده ها sparse عموما توزیعی شبیه توزیع گوسی دارند لذا پیش بینی آن ها با هسته گوسی بهتر است. از طرفی bandwidth هسته گوسی بسیار قدرتمند است و با تغییر درست آن میتوانیم واریانس مدل را بسیار بسیار کم کنیم. حساسیت توزیع گوسی نسبت به توزیع یونیفرم نیز بسیار کمتر است چون ۰ و ۱ ای نیست و این باعث میشود که حساسیت به نویز نیز در این هسته کمتر هسته یونیفرم باشد. لذا با توجه به نکات فوق، هسته گوسی در این مواقع پیشنهاد میشود.