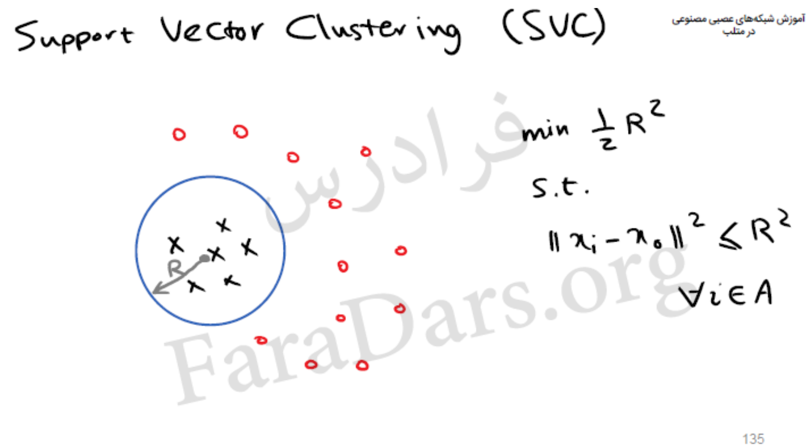
در این جلسه

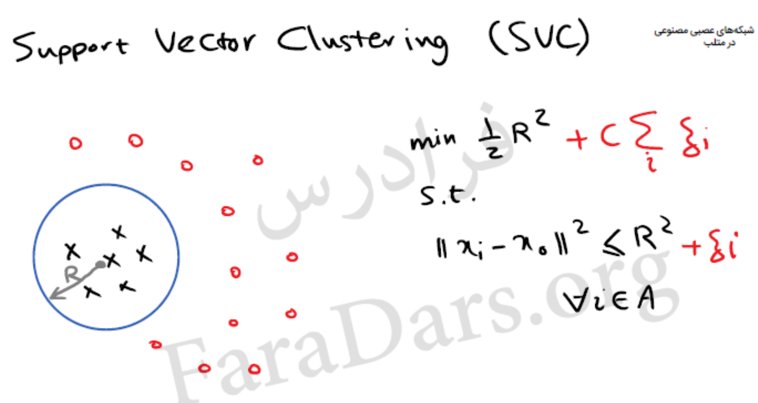
(SVC) or Support vector clustering and Support vector Regression(SVR)

خب در یک اسلاید برای شما میاورم یک سری توضیحات بر روی آن میدهیم .



اگر دقت کنید ما به دنبال این هد ف هستیم که یک مسئله ای را بهینه سازی کنیم که اطلاعات ما درون آن دایره قرار دارد و قیود آن این است که دیتاهای ما باید کم یا مساوی شعاع شوند به ازای هر مقداری که عضو کلاس A است .

حال آمدیم که تمامی اطلاعاتی که مدنظر ما است داخل این دایره نبود حال چه رویکردی داریم اینجا هم همانند SVM میاییم یک پارامتری تعریف میکنیم که باعث شود این مقدار نرم تر عمل کند چی نرم تر عمل کند تابع عزینه و قید های ما .



این همان زتا یا پارامتری است که دیتا ها تخطی کرده باشند از مرز ما با این پارامتر آن ها را درست میکنیم.

در نهایت به یک مسئله کلی میرسیم که شکل زیر را دارد :

یک کاره خوبی که این روش دارد این است که نیازی به این نداریم که دیتاهایی بیرون از این دایره را دسته بندی کنیم، و ادین باعهث میشود که کار ما خیلی آسان تر شود .

$$\min \sum_i \sum_j \alpha_i \alpha_j x_i^T x_j - \sum_i \alpha_i x_i^T x_i$$

$$\text{s.t.} \quad \sum_i \alpha_i = 1$$

$$0 \leqslant \alpha_i \leqslant C$$

$$S = \{ i \mid 0 < \alpha_i < C \}$$

حال اگر فرض شود یا گفته شود ما دیتا هایی که داریم به شکل یک دایره در نمیاید خب چیکار باید کنیم برای این کار در یک ماتریس وزنی ضرب میکنیم که در شکل زیر برای این موضع یک شکل آورده ام :

در اینجا W همان ماتریس وزنی است که گفتیم که به جای نرم فاصله از مرکز سمت راست تساوی یک ماتریس W یا وزن ضرب کرده ایم ولی بازم باید شرط کوچکتر مساوی از شعاع واحد را داشته باشد.

$$\| x_i - x_0 \|^2 = (x_i - x_0)^T (x_i - x_0)$$

$$\| x_i - x_0 \|_W^2 = (x_i - x_0)^T W (x_i - x_0)$$

$$\| x_i - x_0 \|_W^2 \leq 1 + \xi_i$$

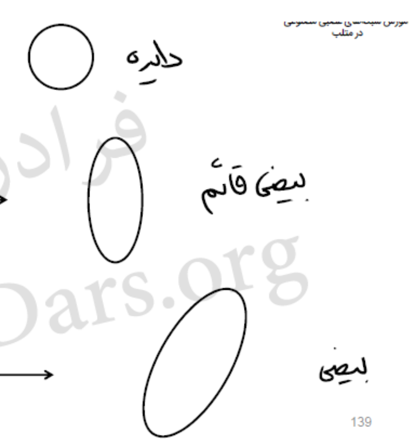
منظور ما در شکل زیر که م.یاورم کلا بیان میشود :

میبینید که حالت اول دایره داریم ماتریس همانی W اگر a,b باشد بیضی قائم و در نهایت بیضی عمومی داریم که تمامی مقادیر این ماتریس دارای عدد هستند.

$$W = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \longrightarrow$$

$$W = \begin{bmatrix} a & 0 \\ 0 & b \end{bmatrix} \longrightarrow$$

$$W = \begin{bmatrix} w_{11} & w_{12} \\ w_{21} & w_{22} \end{bmatrix} \longrightarrow$$



در ادامه میخواهیم Support vector Regression(SVR) بپردازیم .

خب در ابتدای کار باید مشخص کنیم که در ایمن روند کاری که داریم چیست برای این روند کاری ابتد ا دیتا های ما باید مشخص شود که به صورت زیر است :

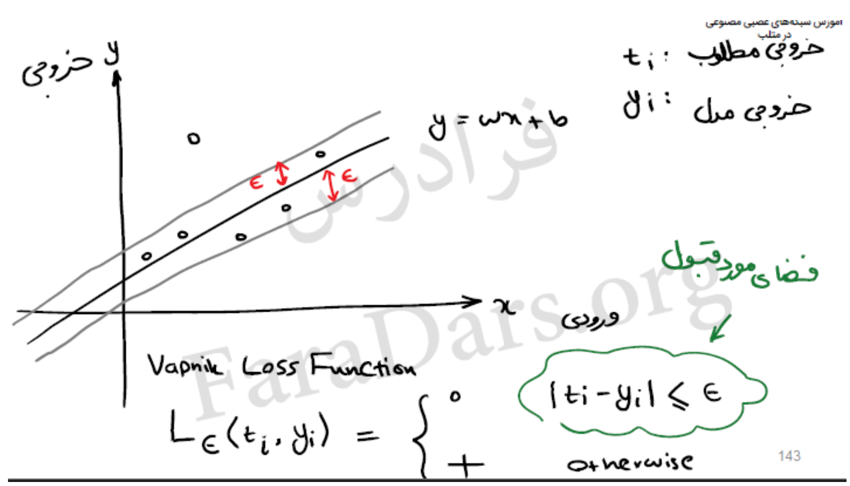
$$x \in \mathbb{R}^m$$

$$t \in \mathbb{R} \quad Data = \{ x_i , t_i \}$$

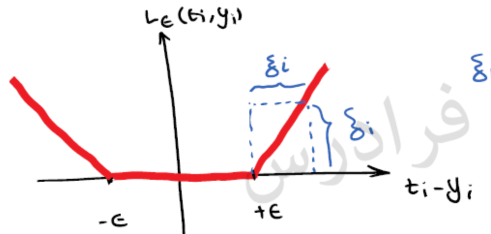
حال میخواهیم چیکار کنیم کار ما این است که یک خط از دیتا ها عبود دهیم که نماینده خو بی از تمامی دیتاها است :

$$t_i \approx y_i = w^T x_i + b$$

خب بعد از این که ما یک رگرسیون را انجام دادیم در این روش دوتا باند برای خود مشخص میکنیم که این دو باند یا پهنا دیتا های ما نباید از آن تخطی کنند برای ایمن منظور یک تابع جریمه تعریف میکنیم که دیتا های خارج از باند به آنها تعلق میگیرد ولی ما یک محدوده معین داریم که مدنظر ما است یعنی بعد از این که سیستم یک سری خروجی داد ما هدف این داریم یا Target ما این است که دیتا های ما فی ما بین این دو باندی باشد که تعین میکنیم که برای فهم بیشتر شکل زیر آورده ام :



اینم همان چیزی است که به دنبال آن هستیم .خب برای این که بتوانیم بهتر کار های عملیاتی خودمان جلو ببریم میبایست یک تابع جریمه خود را به نحوی تعریف کنیم که در شکل بعدی آورده شده است که اگر مقدار خطا یا اختلاف مابین همان باند باشد صفر است بین مثبت epsilon و منفی epsilon که ما بین این صفر است مابقی با یک خط مشخص میشود :



$$\xi_i = |t_i - y_i| - \epsilon$$

$$L_\epsilon(t_i,y_i) = \begin{cases} 0 & |t_i - y_i| \leqslant \epsilon \\ |t_i - y_i| - \epsilon & \text{otherwise} \end{cases}$$

این همان تابع جریمه است که گفتم باید تعریف شود .

در ادامه میخواهیم یک ریسک عملیاتی را تا حد ممکن کاهش دهیم برای این منظور میبایست همانند روش های قبلی بردار پشتیبان عمل کنیم که به شکل زیر عمل میکنیم :

$$R_{emp} = \frac{1}{N}\sum_{i=1}^{N} L_\epsilon(t_i,y_i)$$ ریسک عملیاتی

$$\min \|\omega\| \rightsquigarrow \min \tfrac{1}{2}\omega^T\omega$$

ساده سازی مدل

در این تصویر به خوبی نمایش داده شده است که ریسک عملیاتی را به حداقل میخواهیم برسانیم و نحوه کاری این است که تابع جریمه ای که ساختیم را به عنوان ریسک بگیریم سپس داده های خود را ساده سازی کنیم و به فرم زیر در بیاید و آنها را کمینه کنیم .

حال باید برای این کار میزان تخطی را نرم کنیم که مثل قبلا پارامتر زتا را بربای این منظور استفاده میکنیم که برای فهم بیشتر به شکل زیر ارجاع میدهم :

$$-\epsilon \leqslant t_i - y_i \leqslant +\epsilon$$

$$\Downarrow$$

$$-\epsilon - \xi_i^- \leqslant t_i - y_i \leqslant \epsilon + \xi_i^+$$

$$\xi_i^+, \xi_i^- \geqslant 0$$

تخطی تخطی

این کلا هدف ما است که به دنبال آن هستیم که Target و خروجی بین منفی epsilon .و مثبت epsilon قرار بگیرند و آن پارامتر Zeta هم برای میزان تخطی که دیتا ها دارند نرم کنیم تابع جریمه خودمان را .

حال در ادامه میخواهیم بر اساس پارامتر Zeta کلا کار کنیم که کارمان راحتر شود یعنی دیتاهایی که از این پهنا باندی که داشتیم تخطی کرده اند ما میاییم یک تابع برای ریسک عملیاتی جدید براساس آنها میسازیم :

$$-\epsilon - \xi_i^- \leq t_i - y_i \leq \epsilon + \xi_i^+$$

$$\begin{cases} \xi_i^+ + \xi_i^- = L_\epsilon(t_i, y_i) \\ \xi_i^+ \xi_i^- = 0 \end{cases}$$

ابتدای امر مجموع این Zeta ها میشود همان تابع جریمه ما که به عنوان ریسک عملیاتی درنظر گرفتیم.

حال در ادامه کار میاییم این را بهینه سازی میکنیم.

در حالت کلی مسئله ای که میماند این است باید به فرم زیر عمل کنیم برای بهینه سازی بر اساس قیودی که ایجاد میشود :

شبکه‌های عصبی مصنوعی در متلب

$$\min \quad \frac{1}{2} w^T w + C \sum_{i=1}^{N} (\xi_i^+ + \xi_i^-)$$

$$\text{s.t.} \quad \alpha_i^+ \leftarrow -t_i + y_i + \epsilon + \xi_i^+ \geq 0 \qquad \forall i$$

$$\alpha_i^- \leftarrow t_i - y_i + \epsilon + \xi_i^- \geq 0 \qquad \forall i$$

$$\mu_i^+ \leftarrow \xi_i^+ \geq 0$$

$$\mu_i^- \leftarrow \xi_i^- \geq 0$$

150

سپس برای این قیود یک ضریب لاگرانژ در نظر گرفت که برای اینهم کار Alpha ,mu برای تابع جریمه و میزان تخطی Zeta در نظر میگیریم حال در اسلاید بعدی که میاورم میبینید این ضرایب در تمامی جملاتی که از این مسئله بیرون میاید ضرب شده سپس نسبت به هر کدام مشتق گیری میکنیم که بهینه سازی را انجام دهیم :

خب این همان تابعی است که گفتم بعد از ضرب شدن ضرایب لاگرانژ در آنها تولید شد و به صورت sigma نوشته شده است چون ما مجموع مقادیر را داریم .

حال در ادامه همانند قبل باید مشتق گیری را از هر کدام پارامتر ها انجام دهیم و بعد از آن چیزی که حاصل میشود در تابع اصلی بگذاریم حال در اینجا مراحل مشتق گیری را برای شمال آورده ام :

$$L_p = \frac{1}{2} w^T w + C \sum_i (\xi_i^+ + \xi_i^-)$$
$$- \sum_i \alpha_i^+ (-t_i + y_i + \epsilon + \xi_i^+)$$
$$- \sum_i \alpha_i^- (t_i - y_i + \epsilon + \xi_i^-)$$
$$- \sum_i \mu_i^+ \xi_i^+ - \sum_i \mu_i^- \xi_i^-$$

نسبت به وزن و بایاس و دو پارامتر مرتبط به میزان تخطی را نسبت به آنهخا مشتق گیری کرده ایم .

آموزش شبکه‌های عصبی مصنوعی در متلب

$$\frac{\partial L_p}{\partial w} = 0 \longrightarrow w = \sum_i (\alpha_i^+ - \alpha_i^-) x_i$$

قید بایاس

$$\frac{\partial L_p}{\partial b} = 0 \longrightarrow \sum_i (\alpha_i^+ - \alpha_i^-) = 0$$

Box Constraint

$$\frac{\partial L_p}{\partial \xi_i^+} = 0 \longrightarrow \alpha_i^+ + \mu_i^+ = C$$
$$\frac{\partial L_p}{\partial \xi_i^-} = 0 \longrightarrow \alpha_i^- + \mu_i^- = C$$

152

بعد از این کار تابع ما میشود به فرم زیر :

که هدف اصلی ما این میباشد این تابع را کمینه سازی کنیم .

آموزش شبکه‌های عصبی مصنوعی در متلب

$$\min \quad \frac{1}{2} \sum_i \sum_j (\alpha_i^+ - \alpha_i^-)(\alpha_j^+ - \alpha_j^-) x_i^T x_j$$
$$- \sum_i (\alpha_i^+ - \alpha_i^-) t_i + \epsilon \sum_i (\alpha_i^+ + \alpha_i^-)$$

$$\sum_i (\alpha_i^+ - \alpha_i^-) = 0$$
$$0 \leq \alpha_i^+ \leq C \qquad 0 \leq \alpha_i^- \leq C$$

154

در نهایت برای بهینه سازی کار خود باید به فرم زیر برسیم که بدین منظور یک سری شرایط مرزی را تعین میکنیم و باید آنها را برسی کرد :

این تمام همان روند بهینه سازی ماست که برای این کار میبایست یک سری شرایط حدی یا مرزی را برسی کنیم که KKT نام دارد حال برای این کار در ادامه صحبت میکنیم .

$$\alpha_i^+ ,\ \alpha_i^- \longrightarrow \left.\begin{array}{l} w \\ \xi \end{array}\right\} \longrightarrow b \rightarrow y = w^T x + b$$

شرایط KKT

$$\alpha_i^+\left(-t_i + y_i + \epsilon + \xi_i^+\right) = 0$$

$$\alpha_i^-\left(t_i - y_i + \epsilon + \xi_i^-\right) = 0$$

$$\mu_i^+ \xi_i^+ = (C - \alpha_i^+)\xi_i^+ = 0$$

$$\mu_i^- \xi_i^- = (C - \alpha_i^-)\xi_i^- = 0$$

حال در نهایت به تحلیل میپردازیم که به شکل زیر در میاید :

$$\alpha_i^+ = 0 \rightarrow \mu_i^+ = C \rightarrow \xi_i^+ = 0$$

$$\hookrightarrow -t_i + y_i + \epsilon \geq 0 \rightarrow t_i - y_i \leq \epsilon$$

$$\alpha_i^- = 0 \rightarrow \mu_i^- = C \rightarrow \xi_i^- = 0$$

$$\hookrightarrow t_i - y_i + \epsilon \geq 0 \rightarrow t_i - y_i \geq -\epsilon$$

$$0 < \alpha_i^+ < C \rightarrow 0 < \mu_i^+ < C \rightarrow \xi_i^+ = 0$$

$$\hookrightarrow -t_i + y_i + \epsilon = 0 \rightarrow t_i - y_i = \epsilon$$

$$t_i - y_i \geq -\epsilon \rightarrow \xi_i^- = 0 \rightarrow \mu_i^- > 0$$

$$\left.\begin{array}{l} \alpha_i^- = 0 \\ \mu_i^- > 0 \end{array}\right\} \rightarrow \begin{array}{l} \alpha_i^- = 0 \\ \mu_i^- = C \end{array}$$

این از تحلیل و در نهایت جوابی که بدست میاید به شکل زیر است :

دو شرطی که قرمز رنگ هستند به عنوان ساپورت وکتور های ما در نظر میگیریم .

$$\alpha_i^+ \alpha_i^- = 0$$

1. $\alpha_i^+ = \alpha_i^- = 0$
2. $0 < \alpha_i^+ < C ,\ \alpha_i^- = 0$
3. $0 < \alpha_i^- < C ,\ \alpha_i^+ = 0$

S.V.

4. $\alpha_i^+ = C ,\ \alpha_i^- = 0$
5. $\alpha_i^- = C ,\ \alpha_i^+ = 0$

بعد از این کار ما به دنبال این هستیم که مسئله خود را کامل حل کنیم که باید مقدار بایاس را پیدا کنیم که به شکل کلی زیر میرسیم بعد از این کار هایی که انجام دادیم : پ

آموزش شبکه‌های عصبی مصنوعی
در متلب

$$t_i = y_i + \mathrm{sign}(\alpha_i^+ - \alpha_i^-)\epsilon$$

$$= w^T x_i + b + \mathrm{sign}(\alpha_i^+ - \alpha_i^-)\epsilon$$

$$b = t_i - w^T x_i - \mathrm{sign}(\alpha_i^+ - \alpha_i^-)\epsilon$$

$$b = \frac{1}{|S|}\sum_{i \in S}\left(t_i - w^T x_i - \mathrm{sign}(\alpha_i^+ - \alpha_i^-)\epsilon\right)$$

161

خب بعد همه این ها میخواهم در یک قاب تمامی روال کاری SVR را برای شما به نمایش بگذاریم :

آموزش شبکه‌های عصبی مصنوعی
در متلب

Inputs: $\{x_i, t_i\}$ $\quad i = 1,2,\dots,N$

Output: $y = w^T x + b$

$$\text{Q.P.} \longrightarrow \alpha_i^+, \alpha_i^- \longrightarrow S = \{i \mid 0 < \alpha_i^+ + \alpha_i^- < C\}$$

$$\longrightarrow w = \sum_i (\alpha_i^+ - \alpha_i^-) x_i$$

$$\longrightarrow b = \frac{1}{|S|}\sum_{i \in S}\left[t_i - w^T x_i - \mathrm{sign}(\alpha_i^+ - \alpha_i^-)\epsilon\right]$$

162

این تمامی مراحل کلی یک الکوریتم SVR است .

ادامه کار در دو کد در گیت هاب بنده است یکی با پارامتر خطی و یکی برای غیر خطی :

DBZ_NeuralNetwork_SVR.m

DBZ_NeuralNetwork_SVR_2.m