

# مقدمه‌ای بر روش ماشین بردار پشتیبان (SVM)

یک راهنمای ساده و جامع برای پژوهشگران علوم اعصاب

محمد مهدی شریف بیگی

MohammadMahdi Sharifbeigy

۲۲ شهریور ۱۴۰۴

## فهرست مطالب

۳	۱ ماشین بردار پشتیبان (SVM) چیست؟
۳	۲ جداکننده خطی و مفهوم حاشیه بیشینه
۴	۳ مقابله با داده‌های جداناپذیر: حاشیه نرم
۵	۱.۳ تابع هزینه Loss Hinge . . . . .
۶	۴ ریاضیات بهینه‌سازی: لاگرانژ، دوگانگی و KKT
۶	۱.۴ شرایط کاروش-کوهن-تاکر (KKT) . . . . .
۷	۵ ترفند کرنل: حرکت به ابعاد بالاتر
۸	۱.۵ انواع کرنل‌های متداول . . . . .
۸	۶ پرسش و پاسخ‌های عملی
۸	۱.۶ چرا SVM به مقیاس‌بندی ویژگی‌ها حساس است؟ . . . . .
۸	۲.۶ SVM چگونه مسائل چندکلاسه را حل می‌کند؟ . . . . .
۸	۳.۶ تأثیر پارامترهای C و Gamma چگونه است؟ . . . . .
۹	۴.۶ مقایسه فلسفه کرنل RBF و کرنل کسینوسی . . . . .
۹	۵.۶ SVM در برابر نویز و داده‌های پرت چگونه عمل می‌کند؟ . . . . .
۹	۶.۶ نفرین ابعاد و بُعد VC چه هستند؟ . . . . .
۹	۱.۶.۶ نفرین ابعاد (Curse of Dimensionality) . . . . .
۱۰	۲.۶.۶ بُعد VC (Vapnik-Chervonenkis Dimension) . . . . .
۱۰	۷ کاربرد ویژه: تشخیص داده پرت در الکتروفیزیولوژی
۱۰	۱.۷ الگوریتم One-Class SVM . . . . .
۱۱	۱.۱.۷ تفاوت ریاضی با SVM استاندارد . . . . .
۱۲	۲.۱.۷ مثال‌های عملی با Scikit-learn . . . . .
۱۴	۳.۱.۷ مراحل عملی شناسایی آرتیفکت EEG با One-Class SVM . . . . .
۱۵	۸ جمع‌بندی: مزایا و معایب SVM
۱۵	۱.۸ مزایا . . . . .
۱۵	۲.۸ معایب . . . . .

## ۱ ماشین بردار پشتیبان (SVM) چیست؟

ماشین بردار پشتیبان یا (Support Vector Machine - SVM) یکی از قدرتمندترین و پرکاربردترین الگوریتم‌های یادگیری ماشین با نظارت (Supervised Learning) است که عمدتاً برای مسائل طبقه‌بندی (Classification) و همچنین رگرسیون (Regression) به کار می‌رود. ایده اصلی SVM بسیار هوشمندانه و در عین حال ساده است: پیدا کردن یک خط یا یک صفحه (که به آن ابرصفحه یا Hyperplane می‌گوییم) که به بهترین شکل ممکن داده‌های مربوط به کلاس‌های مختلف را از یکدیگر جدا کند.

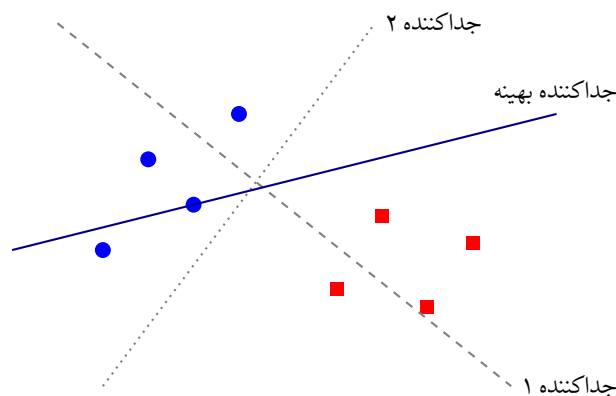
«بهترین شکل ممکن» در اینجا به معنای پیدا کردن ابرصفحه‌ای است که بیشترین فاصله را با نزدیک‌ترین نمونه‌های هر کلاس داشته باشد. این فاصله، حاشیه (Margin) نامیده می‌شود. SVM با بیشینه‌سازی این حاشیه، یک مرز تصمیم‌گیری قدرتمند و قابل تعمیم ایجاد می‌کند که در مقابل نویز و داده‌های جدید عملکرد خوبی از خود نشان می‌دهد.

در حوزه علوم اعصاب و تحلیل سیگنال‌های الکتروفیزیولوژی، از SVM می‌توان برای کاربردهای متنوعی استفاده کرد، مانند:

- واسطه‌های مغز و کامپیوتر (BCI): تشخیص و طبقه‌بندی الگوهای مغزی مرتبط با تصور حرکات مختلف (مثلاً حرکت دست راست در مقابل دست چپ).
- تشخیص بیماری: تفکیک سیگنال‌های EEG افراد سالم از بیماران مبتلا به صرع یا آلزایمر.
- تحلیل حالت‌های شناختی: طبقه‌بندی وضعیت‌های ذهنی مختلف مانند تمرکز، آرامش یا خواب‌آلودگی بر اساس ویژگی‌های استخراج‌شده از سیگنال‌های مغزی.

## ۲ جداکننده خطی و مفهوم حاشیه بیشینه

بیایید با یک مثال ساده شروع کنیم. فرض کنید داده‌های ما در یک فضای دو بعدی قرار دارند و به دو کلاس مختلف (مثلاً دایره‌های آبی و مربع‌های قرمز) تعلق دارند. هدف ما پیدا کردن یک خط است که این دو کلاس را از هم جدا کند.

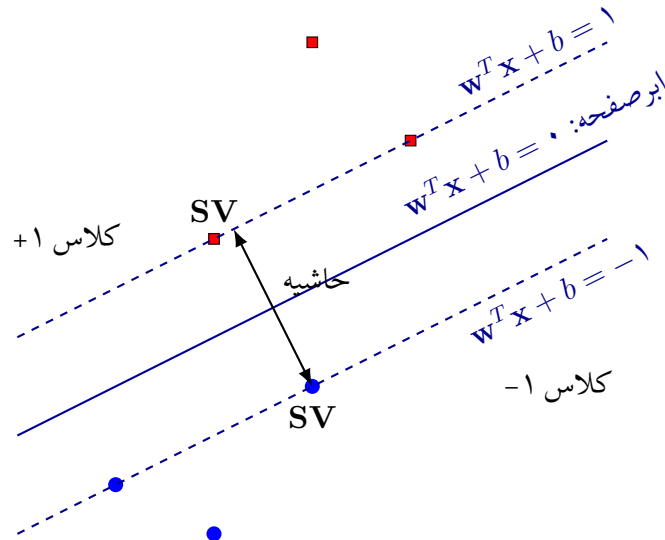


شکل ۱: چندین خط می‌توانند دو کلاس را از هم جدا کنند، اما کدام یک بهترین است؟

همانطور که در شکل ۱ می‌بینید، بی‌نهایت خط می‌توان برای جداسازی این دو کلاس رسم کرد. اما SVM به دنبال خطی است که بیشترین «فضای خالی» یا «حاشیه امن» را بین خود و نزدیک‌ترین نقاط هر دو کلاس ایجاد کند. این خط را ابرصفحه با حاشیه بیشینه (Maximal Margin Hyperplane) می‌نامیم.

ابرفاصله به صورت ریاضی با معادله  $f(\mathbf{x}) = \mathbf{w}^T \mathbf{x} + b = 0$  تعریف می‌شود، که در آن:

- $\mathbf{w}$  یک بردار عمود بر ابرصفحه است (بردار وزن‌ها). جهت این بردار، جهت جداسازی را نشان می‌دهد.



شکل ۲: ابرصفحه با حاشیه بیشینه. نقاطی که روی خطوط حاشیه قرار دارند، بردارهای پشتیبان (Support Vectors) نامیده می‌شوند.

•  $x$  بردار ویژگی‌های یک نقطه داده است.

•  $b$  یک مقدار بایاس است که موقعیت ابرصفحه را نسبت به مبدأ مختصات مشخص می‌کند.

دو خط موازی دیگر که حاشیه را تعریف می‌کنند، با معادلات  $w^T x + b = 1$  و  $w^T x + b = -1$  مشخص می‌شوند. فاصله بین این دو خط برابر با  $\frac{2}{\|w\|}$  است. هدف SVM بیشینه‌سازی این فاصله است که معادل با کمینه‌سازی  $\|w\|$  (یا به طور معادل  $\frac{1}{2}\|w\|^2$  برای سادگی محاسبات) می‌باشد.

#### مسئله بهینه‌سازی (حاشیه سخت)

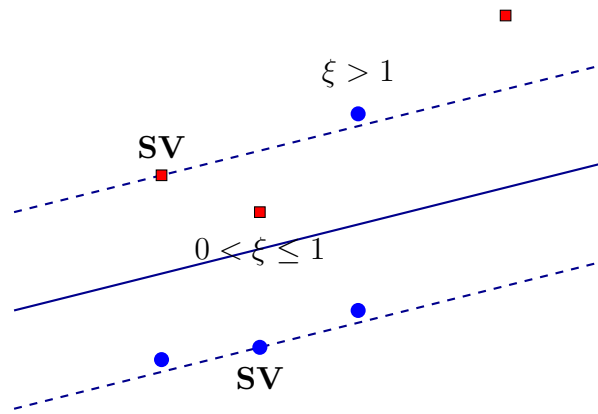
$$\underset{w, b}{\text{minimize}} \quad \frac{1}{2} \|w\|^2 \quad (1)$$

$$\text{subject to} \quad y_i(w^T x_i + b) \geq 1, \quad \forall i = 1, \dots, n \quad (2)$$

که در آن  $y_i$  برچسب کلاس نمونه  $i$ -ام است (با مقدار +1 یا -1). قید (۲) تضمین می‌کند که تمام نقاط داده به درستی و خارج از حاشیه طبقه‌بندی شوند.

### ۳ مقابله با داده‌های جداناپذیر: حاشیه نرم

مدل «حاشیه سخت» (Hard Margin) تنها زمانی کار می‌کند که داده‌ها کاملاً به صورت خطی جداپذیر باشند. اما در دنیای واقعی، داده‌ها اغلب دارای نویز هستند یا با یکدیگر همپوشانی دارند. راه حل این مشکل، استفاده از حاشیه نرم (Soft Margin) است. در این روش، به مدل اجازه می‌دهیم که برخی از نقاط را به اشتباه طبقه‌بندی کند یا آن‌ها را درون حاشیه قرار دهد. برای هر نقطه، یک متغیر کندی (Slack Variable) با نماد  $\xi_i \geq 0$  تعریف می‌کنیم. مسئله بهینه‌سازی به شکل زیر تغییر می‌کند:



شکل ۳: در روش حاشیه نرم، به برخی نقاط اجازه داده می‌شود که حاشیه را نقض کنند ( $\xi_i > 0$ ). این نقاط متخلف نیز بردار پشتیبان محسوب می‌شوند.

#### مسئله بهینه‌سازی (حاشیه نرم)

$$\underset{\mathbf{w}, b, \xi}{\text{minimize}} \quad \frac{1}{2} \|\mathbf{w}\|^2 + C \sum_{i=1}^n \xi_i \quad (3)$$

$$\text{subject to} \quad y_i(\mathbf{w}^T \mathbf{x}_i + b) \geq 1 - \xi_i, \quad \text{and} \quad \xi_i \geq 0, \quad \forall i \quad (4)$$

در اینجا پارامتر جدید  $C > 0$  پارامتر تنظیم یا Regularization Parameter نام دارد. این پارامتر یک توازن (Trade-off) بین دو هدف متضاد برقرار می‌کند:

- بیشینه‌سازی حاشیه: که با کمینه کردن جمله  $\frac{1}{2} \|\mathbf{w}\|^2$  (مدل ساده) به دست می‌آید.
- کمینه‌سازی خطای طبقه‌بندی: که با کمینه کردن جمله  $\sum \xi_i$  (دقت بالا روی داده‌های آموزش) حاصل می‌شود.

مقدار بزرگ  $C$  به معنای جریمه سنگین برای خطاها است و مدل را به سمت یک حاشیه باریک‌تر با خطای کمتر سوق می‌دهد (ریسک بیش‌برازش یا Overfitting). مقدار کوچک  $C$  به مدل اجازه می‌دهد حاشیه پهن‌تری داشته باشد، حتی اگر به قیمت چند خطای طبقه‌بندی بیشتر تمام شود (ریسک کم‌برازش یا Underfitting).

### ۱.۳ تابع هزینه Loss Hinge

می‌توان فرمول‌بندی حاشیه نرم را از دیدگاه دیگری نیز مشاهده کرد: با استفاده از یک تابع هزینه (Loss Function). تابع هزینه‌ای که SVM از آن استفاده می‌کند Loss Hinge نام دارد و به صورت زیر تعریف می‌شود:

## تابع هزینه Loss Hinge

$$L(y_i, f(\mathbf{x}_i)) = \max(0, 1 - y_i f(\mathbf{x}_i)) \quad (5)$$

که در آن  $f(\mathbf{x}_i) = \mathbf{w}^T \mathbf{x}_i + b$  خروجی مدل است. این تابع هزینه دارای ویژگی‌های زیر است:

- اگر یک نمونه به درستی و خارج از حاشیه طبقه‌بندی شود ( $y_i f(\mathbf{x}_i) \geq 1$ )، هزینه آن صفر است.
- اگر یک نمونه درون حاشیه قرار گیرد یا به اشتباه طبقه‌بندی شود ( $y_i f(\mathbf{x}_i) < 1$ )، هزینه آن به صورت خطی با فاصله از مرز حاشیه افزایش می‌یابد.

در واقع، متغیرهای  $\xi_i$  دقیقاً برابر با مقدار Hinge Loss برای هر نمونه هستند:  $\xi_i = \max(0, 1 - y_i f(\mathbf{x}_i))$ . بنابراین، مسئله بهینه‌سازی حاشیه نرم را می‌توان به صورت زیر نیز نوشت که به آن فرمول‌بندی اولیه نام‌نقید می‌گویند:

$$\underset{\mathbf{w}, b}{\text{minimize}} \quad \frac{1}{2} \|\mathbf{w}\|^2 + C \sum_{i=1}^n \max(0, 1 - y_i(\mathbf{w}^T \mathbf{x}_i + b)) \quad (6)$$

## ۴ ریاضیات بهینه‌سازی: لاگرانژ، دوگانگی و KKT

مسئله بهینه‌سازی SVM یک مسئله برنامه‌نویسی درجه دوم (Quadratic Programming - QP) است. برای حل این مسئله، از روش قدرتمند مضارب لاگرانژ استفاده می‌کنیم. تابع لاگرانژین اولیه به صورت زیر تعریف می‌شود:

$$L_P(\mathbf{w}, b, \boldsymbol{\xi}, \boldsymbol{\alpha}, \boldsymbol{\mu}) = \frac{1}{2} \|\mathbf{w}\|^2 + C \sum \xi_i - \sum \alpha_i [y_i(\mathbf{w}^T \mathbf{x}_i + b) - 1 + \xi_i] - \sum \mu_i \xi_i \quad (7)$$

که در آن  $\alpha_i \geq 0$  و  $\mu_i \geq 0$  مضارب لاگرانژ هستند. با مشتق‌گیری و ساده‌سازی، به مسئله دوگان (Dual Problem) می‌رسیم:

## مسئله بهینه‌سازی دوگان

$$\underset{\boldsymbol{\alpha}}{\text{maximize}} \quad W(\boldsymbol{\alpha}) = \sum_{i=1}^n \alpha_i - \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n \alpha_i \alpha_j y_i y_j (\mathbf{x}_i^T \mathbf{x}_j) \quad (8)$$

$$\text{subject to} \quad \sum_{i=1}^n \alpha_i y_i = 0, \quad \text{and} \quad 0 \leq \alpha_i \leq C, \quad \forall i \quad (9)$$

نکته کلیدی این است که مسئله دوگان فقط به متغیرهای  $\alpha_i$  بستگی دارد و داده‌ها تنها از طریق ضرب داخلی  $(\mathbf{x}_i^T \mathbf{x}_j)$  در آن ظاهر می‌شوند. این ویژگی اساس کار ترفند کرنل است.

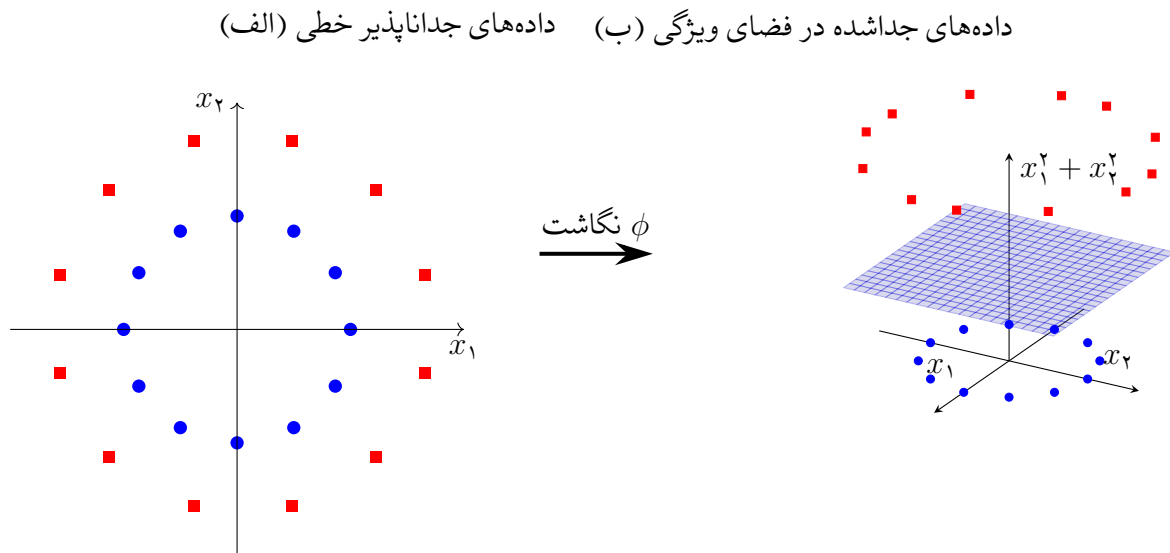
## ۱.۴ شرایط کاروش-کوهن-تاکر (KKT)

شرایط Karush-Kuhn-Tucker (KKT) ارتباط بین مسئله اولیه و دوگان را برقرار کرده و تفسیر جالبی از متغیرهای  $\alpha_i$  ارائه می‌دهند:

- اگر  $\alpha_i = 0$  باشد، نقطه  $\mathbf{x}_i$  به درستی و خارج از حاشیه طبقه‌بندی شده است. این نقطه در تعیین مرز نقشی ندارد.
  - اگر  $0 < \alpha_i < C$  باشد، آنگاه  $\xi_i = 0$  و نقطه  $\mathbf{x}_i$  دقیقاً روی حاشیه قرار دارد. این نقطه یک بردار پشتیبان است.
  - اگر  $\alpha_i = C$  باشد، آنگاه  $\xi_i > 0$  و نقطه  $\mathbf{x}_i$  یا درون حاشیه قرار دارد یا به اشتباه طبقه‌بندی شده است. این نقطه نیز یک بردار پشتیبان است.
- این ویژگی به این معناست که مرز تصمیم‌گیری تنها توسط زیرمجموعه کوچکی از داده‌ها (بردارهای پشتیبان) تعریف می‌شود.

## ۵ ترفند کرنل: حرکت به ابعاد بالاتر

تا اینجا فرض کردیم که داده‌ها به صورت خطی جداپذیر هستند. اما اگر مرز بین کلاس‌ها یک منحنی پیچیده باشد، چه باید کرد؟



شکل ۴: ایده اصلی ترفند کرنل: نگاشت داده‌ها به یک فضای با ابعاد بالاتر که در آن به صورت خطی جداپذیر شوند.

ایده ترفند کرنل (Kernel Trick) این است که به جای محاسبه پرهزینه نگاشت  $\phi(\mathbf{x})$ ، مستقیماً ضرب داخلی  $\phi(\mathbf{x}_i)^T \phi(\mathbf{x}_j)$  را با استفاده از یک تابع کرنل  $K(\mathbf{x}_i, \mathbf{x}_j)$  محاسبه کنیم.

### تابع کرنل و شرط مرسر

$$K(\mathbf{x}_i, \mathbf{x}_j) = \phi(\mathbf{x}_i)^T \phi(\mathbf{x}_j) \quad (10)$$

یک تابع برای اینکه بتواند به عنوان یک کرنل معتبر استفاده شود، باید شرط مرسر را برآورده کند. این شرط تضمین می‌کند که ماتریس حاصل از اعمال کرنل بر روی تمام زوج داده‌ها، یک ماتریس نیمه معین مثبت (Positive Semi-definite) باشد.

## ۱.۵ انواع کرنل‌های متداول

- کرنل خطی:  $K(\mathbf{x}_i, \mathbf{x}_j) = \mathbf{x}_i^T \mathbf{x}_j$ .
- کرنل چندجمله‌ای:  $K(\mathbf{x}_i, \mathbf{x}_j) = (\gamma \mathbf{x}_i^T \mathbf{x}_j + r)^d$ .
- کرنل تابع پایه شعاعی (RBF):  $K(\mathbf{x}_i, \mathbf{x}_j) = \exp(-\gamma \|\mathbf{x}_i - \mathbf{x}_j\|^2)$ . این کرنل محبوب‌ترین انتخاب است.
- کرنل کسینوسی:  $K(\mathbf{x}_i, \mathbf{x}_j) = \frac{\mathbf{x}_i^T \mathbf{x}_j}{\|\mathbf{x}_i\| \cdot \|\mathbf{x}_j\|}$ .

## ۶ پرسش و پاسخ‌های عملی

### ۱.۶ چرا SVM به مقیاس‌بندی ویژگی‌ها حساس است؟

تابع هزینه SVM تلاش می‌کند تا فاصله‌ها را بیشینه کند. اگر یک ویژگی دارای مقیاس بسیار بزرگتری نسبت به بقیه باشد، فاصله در آن بُعد بر محاسبات غلبه خواهد کرد. به همین دلیل، استانداردسازی یا نرمال‌سازی داده‌ها قبل از آموزش SVM یک مرحله ضروری است.

### ۲.۶ SVM چگونه مسائل چندکلاسه را حل می‌کند؟

SVM ذاتاً یک طبقه‌بند دودویی است. برای حل مسائل با بیش از دو کلاس، از دو استراتژی رایج استفاده می‌شود:

- یک در مقابل بقیه (One-vs-Rest - OvR): برای  $K$  کلاس،  $K$  طبقه‌بند SVM آموزش داده می‌شود.
- یک در مقابل یک (One-vs-One - OvO): برای  $K$  کلاس، تعداد  $\frac{K(K-1)}{2}$  طبقه‌بند آموزش داده می‌شود. این روش معمولاً نتایج بهتری دارد اما از نظر محاسباتی سنگین‌تر است.

### ۳.۶ تأثیر پارامترهای $C$ و $\gamma$ چگونه است؟

دو پارامتر  $C$  و  $\gamma$  (در کرنل RBF نقش حیاتی در کنترل توازن بایاس-واریانس (Bias-Variance Trade-off) دارند.

- پارامتر  $C$  (هزینه): همانطور که قبلاً گفته شد،  $C$  میزان جریمه برای طبقه‌بندی اشتباه را کنترل می‌کند.  $C$  بالا به معنای جریمه سنگین (مدل پیچیده، واریانس بالا) و  $C$  پایین به معنای جریمه سبک (مدل ساده، بایاس بالا) است.
- پارامتر  $\gamma$  (گاما): این پارامتر در کرنل RBF، میزان تأثیر یک نمونه آموزشی را مشخص می‌کند. مقدار کم  $\gamma$  به معنای تأثیر زیاد (شعاع بزرگ) است، که منجر به یک مرز تصمیم‌گیری نرم و کلی می‌شود (بایاس بالا). مقدار زیاد  $\gamma$  به معنای تأثیر کم (شعاع کوچک) است، که باعث می‌شود مرز تصمیم‌گیری به شدت به نمونه‌های آموزشی نزدیک شود و پیچیده گردد (واریانس بالا و ریسک بیش‌برازش).

پیدا کردن ترکیب بهینه از این دو پارامتر معمولاً با روش‌هایی مانند جستجوی شبکه‌ای (Grid Search) و اعتبارسنجی متقابل انجام می‌شود.



## ۴.۶ مقایسه فلسفه کرنل RBF و کرنل کسینوسی

این دو کرنل دو دیدگاه متفاوت برای سنجش «شباهت» بین نقاط دارند:

- کرنل RBF (مبتنی بر فاصله): این کرنل شباهت را بر اساس نزدیکی در فضای اقلیدسی می‌سنجد. مرزهای تصمیم آن محلی و منحنی هستند و خوشه‌های داده را احاطه می‌کنند. این کرنل به اندازه (magnitude) بردارها حساس است.
- کرنل کسینوسی (مبتنی بر جهت): این کرنل شباهت را بر اساس زاویه (جهت) بین بردارها نسبت به مبدأ مختصات می‌سنجد و اندازه بردارها را نادیده می‌گیرد. مرزهای تصمیم آن به صورت خطوط صاف و شعاعی هستند که از مبدأ سرچشمه می‌گیرند. این کرنل در حوزه‌هایی مانند پردازش متن که جهت بردار (کلمات کلیدی) مهم‌تر از طول آن (تعداد کلمات) است، کاربرد دارد.

## ۵.۶ SVM در برابر نویز و داده‌های پرت چگونه عمل می‌کند؟

این سؤال مستقیماً به مفهوم حاشیه نرم و نقش پارامتر تنظیم  $C$  بازمی‌گردد. مکانیزم اصلی SVM برای مقابله با نویز و داده‌های پرت، «نادیده گرفتن» هوشمندانه آن‌هاست.

- داده پرت (Outlier): یک داده پرت نقطه‌ای است که بسیار دور از سایر داده‌های کلاس خود قرار گرفته است. در مدل حاشیه سخت، تنها یک داده پرت می‌تواند موقعیت ابرصفحه را به کلی تغییر داده و حاشیه را بسیار باریک کند. اما در مدل حاشیه نرم، اگر هزینه طبقه‌بندی اشتباه آن نقطه (که توسط  $C$  کنترل می‌شود) کمتر از هزینه کاهش شدید عرض حاشیه باشد، الگوریتم ترجیح می‌دهد آن نقطه را اشتباه طبقه‌بندی کند (یعنی  $\xi_i > 0$ ) تا یک حاشیه پهن و سالم برای بقیه داده‌ها حفظ شود.

- نویز (Noise): نویز به نقاطی اطلاق می‌شود که در منطقه متعلق به کلاس دیگر ظاهر می‌شوند. SVM با حاشیه نرم به سادگی به این نقاط اجازه می‌دهد که در سمت اشتباه مرز قرار بگیرند (تبدیل به خطای طبقه‌بندی شوند) تا از ایجاد یک مرز تصمیم‌گیری بیش از حد پیچیده و پرپیچ‌وخم (بیش‌برازش) که سعی در جدا کردن تک‌تک نقاط دارد، جلوگیری کند.

به طور خلاصه، با تنظیم مناسب پارامتر  $C$  (معمولاً با اعتبارسنجی متقابل)، SVM می‌تواند بین حفظ یک حاشیه بزرگ (که قدرت تعمیم را بالا می‌برد) و طبقه‌بندی صحیح تمام نقاط (که به نویز حساس است) تعادل برقرار کند و در نتیجه نسبت به داده‌های پرت مقاوم باشد.

## ۶.۶ نفرین ابعاد و بُعد VC چه هستند؟

این دو مفهوم نظری از حوزه یادگیری آماری به ما کمک می‌کنند تا بفهمیم چرا SVM با وجود قدرت بالایش، دچار بیش‌برازش نمی‌شود.

### ۱.۶.۶ نفرین ابعاد (Curse of Dimensionality)

این اصطلاح به مشکلات متعددی اشاره دارد که هنگام کار با داده‌ها در فضاهای با ابعاد بسیار بالا رخ می‌دهد. با افزایش تعداد ویژگی‌ها (ابعاد)، حجم فضا به صورت نمایی افزایش می‌یابد. در نتیجه، داده‌ها به شدت پراکنده و دور از هم می‌شوند. این پراکندگی باعث می‌شود الگوریتم‌هایی که بر اساس فاصله یا همسایگی کار می‌کنند (مانند  $k$ -NN) کارایی خود را از دست بدهند، زیرا مفهوم «نزدیکی» بی‌معنا می‌شود.

SVM چگونه با این مشکل مقابله می‌کند؟ به طور شگفت‌انگیزی، SVM نه تنها از نفرین ابعاد رنج نمی‌برد، بلکه اغلب از آن سود می‌برد. دلیل این امر آن است که مرز تصمیم‌گیری در SVM تنها توسط بردارهای پشتیبان تعریف می‌شود، نه تمام داده‌ها. در یک فضای با ابعاد بالا، احتمال اینکه داده‌ها به صورت خطی جداپذیر شوند، بیشتر می‌شود. ترفند کرنل این ایده را به نهایت خود می‌رساند و داده‌ها را به یک فضای با ابعاد بسیار بالا (حتی بی‌نهایت) می‌برد تا یک ابرصفحه جداکننده پیدا کند، بدون آنکه هزینه محاسباتی این فضا را متحمل شود.

## ۲.۶.۶ بُعد VC (Vapnik-Chervonenkis Dimension)

بُعد VC یک معیار برای سنجش ظرفیت، پیچیدگی یا قدرت یک مدل طبقه‌بندی است. به طور شهودی، بُعد VC برابر با بیشترین تعداد نقاطی است که مدل می‌تواند آن‌ها را به تمام ترکیبات ممکن برچسب‌گذاری کند (به اصطلاح shatter کند).

- یک مدل با بُعد VC بسیار بالا، مدلی بسیار انعطاف‌پذیر است که می‌تواند داده‌های پیچیده را یاد بگیرد، اما در معرض خطر بیش‌برازش قرار دارد.
- یک مدل با بُعد VC پایین، مدل ساده‌تری است که قدرت تعمیم‌پذیری بالاتری دارد.

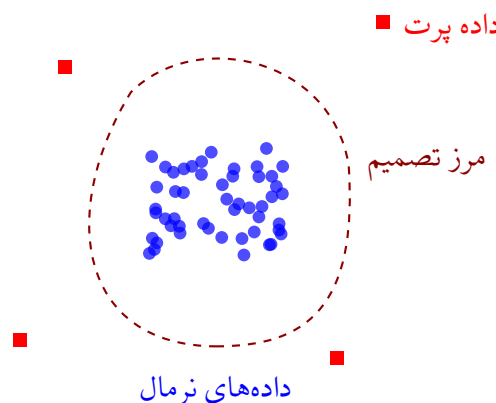
ارتباط آن با SVM چیست؟ یکی از مهم‌ترین نتایج نظری در پشت SVM این است که خطای تعمیم آن (خطا روی داده‌های جدید) نه به بُعد فضا، بلکه به نسبت بین شعاع کره دربرگیرنده داده‌ها و عرض حاشیه بستگی دارد. به عبارت دیگر، با بیشینه‌سازی حاشیه، SVM به طور مؤثری بُعد VC خود را کنترل کرده و یک مدل با کمترین پیچیدگی ممکن که قادر به جداسازی داده‌هاست، پیدا می‌کند. این دلیل اصلی قدرت تعمیم بالای SVM است، حتی زمانی که با کرنل RBF در یک فضای با ابعاد بی‌نهایت کار می‌کند.

## ۷ کاربرد ویژه: تشخیص داده پرت در الکتروفیزیولوژی

یکی از چالش‌های بزرگ در تحلیل سیگنال‌های الکتروفیزیولوژی مانند EEG و MEG، وجود آرتیفکت‌ها یا داده‌های پرت (Outliers) است. SVM یک ابزار بسیار کارآمد برای شناسایی این ناهنجاری‌ها ارائه می‌دهد که به آن One-Class SVM می‌گویند.

### ۱.۷ الگوریتم One-Class SVM

برخلاف SVM استاندارد که به دنبال مرزی بین دو کلاس است، One-Class SVM به دنبال یادگیری یک مرز فشرده در اطراف یک کلاس از داده‌ها (داده‌های نرمال) است. این رویکرد را می‌توان به کشیدن یک حصار دور گله‌ای از گوسفندان تشبیه کرد؛ هر حیوانی که بیرون این حصار باشد، به احتمال زیاد گوسفند نیست (گرگ یا همان داده پرت است). الگوریتم تلاش می‌کند تا با کمترین محیط ممکن، بیشترین تعداد گوسفندان را در بر بگیرد. هر نقطه‌ای که خارج از این مرز قرار گیرد، به عنوان یک داده پرت یا ناهنجاری شناسایی می‌شود.



شکل ۵: نمایش شماتیک عملکرد One-Class SVM. الگوریتم مرزی را دور داده‌های نرمال می‌آموزد تا نقاط خارج از آن را به عنوان ناهنجاری شناسایی کند.

در این روش نیز می‌توان از ترفند کرنل برای یادگیری مرزهای غیرخطی و پیچیده حول داده‌های نرمال استفاده کرد. برای مثال، کرنل RBF می‌تواند مرزهایی با اشکال دلخواه ایجاد کند که برای توصیف توزیع‌های پیچیده داده‌های فیزیولوژیک بسیار مناسب است.

## مسئله بهینه‌سازی One-Class SVM

ایده اصلی، پیدا کردن ابرصفحه‌ای است که داده‌های نرمال را با بیشترین حاشیه از مبدأ مختصات جدا کند.

$$\underset{\mathbf{w}, b, \xi}{\text{minimize}} \quad \frac{1}{2} \|\mathbf{w}\|^2 + \frac{1}{\nu n} \sum_{i=1}^n \xi_i - \rho \quad (11)$$

$$\text{subject to} \quad \mathbf{w}^T \phi(\mathbf{x}_i) \geq \rho - \xi_i, \quad \text{and} \quad \xi_i \geq 0, \quad \forall i \quad (12)$$

در اینجا،  $\rho$  متغیری است که موقعیت مرز را تعیین می‌کند. پارامتر کلیدی در این مدل  $\nu$  (نو) است (یک عدد بین ۰ و ۱) که یک توازن را کنترل می‌کند:

- این پارامتر یک حد بالا برای کسر داده‌های پرتی که مدل مجاز به شناسایی آن‌هاست، تعیین می‌کند.
- همچنین یک حد پایین برای کسر بردارهای پشتیبانی که برای تعریف مرز استفاده می‌شوند، مشخص می‌کند.

## ۱.۱.۷ تفاوت ریاضی با SVM استاندارد

تفاوت اصلی در هدف و فرمول‌بندی مسئله بهینه‌سازی نهفته است. SVM استاندارد یک الگوریتم با نظارت است که از برچسب‌ها ( $y_i$ ) برای یافتن یک ابرصفحه جداکننده بین کلاس‌ها استفاده می‌کند. در مقابل، SVM One-Class یک الگوریتم ذاتاً بی‌نظارت (یا نیمه‌نظارت) است که به دنبال یافتن یک مرز در اطراف یک کلاس از داده‌هاست.

- SVM استاندارد (حاشیه نرم):

- هدف: بیشینه‌سازی حاشیه بین دو کلاس.
- ورودی: داده‌های  $\mathbf{x}_i$  و برچسب‌های آن‌ها  $y_i \in \{+1, -1\}$ .
- فرمول‌بندی:

$$\underset{\mathbf{w}, b, \xi}{\text{minimize}} \quad \frac{1}{2} \|\mathbf{w}\|^2 + C \sum \xi_i$$

$$\text{subject to} \quad y_i (\mathbf{w}^T \mathbf{x}_i + b) \geq 1 - \xi_i$$

- SVM: One-Class

- هدف: پیدا کردن یک مرز فشرده که داده‌های نرمال را از مبدأ مختصات جدا کند.
- ورودی: فقط داده‌های نرمال  $\mathbf{x}_i$ . برچسبی وجود ندارد.
- فرمول‌بندی:

$$\underset{\mathbf{w}, \rho, \xi}{\text{minimize}} \quad \frac{1}{2} \|\mathbf{w}\|^2 + \frac{1}{\nu n} \sum \xi_i - \rho$$

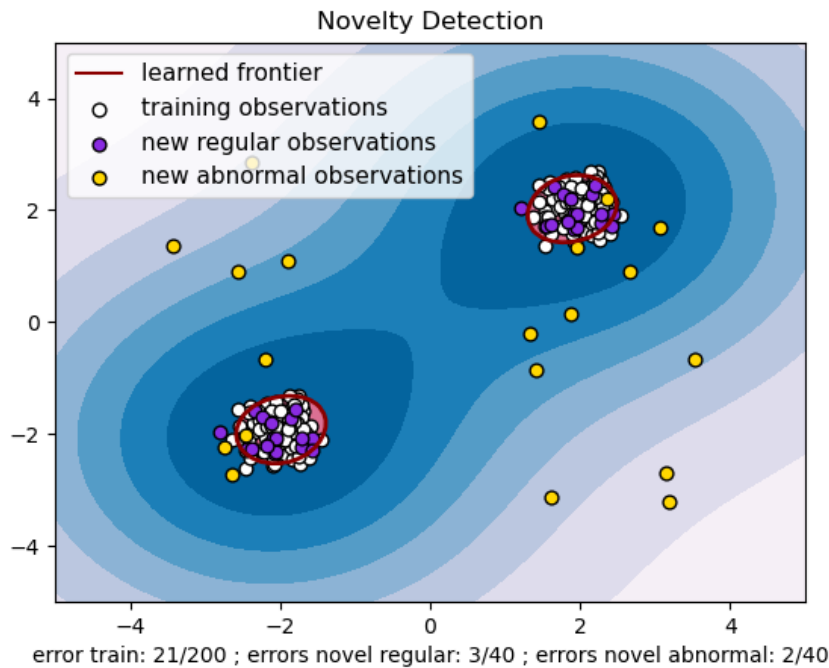
$$\text{subject to} \quad \mathbf{w}^T \phi(\mathbf{x}_i) \geq \rho - \xi_i$$

همانطور که مشاهده می‌شود، در SVM One-Class برچسب  $y_i$  حذف شده و دو پارامتر جدید معرفی می‌شوند:  $\rho$  که آفست یا بایاس برای ابرصفحه نسبت به مبدأ است و  $\nu$  که جایگزین پارامتر  $C$  شده و توازن بین حجم مرز و تعداد خطاهای مجاز را کنترل می‌کند.

## ۲.۱.۷ مثال‌های عملی با Scikit-learn

برای درک بهتر رفتار این الگوریتم، دو مثال از کتابخانه Scikit-learn را بررسی می‌کنیم.

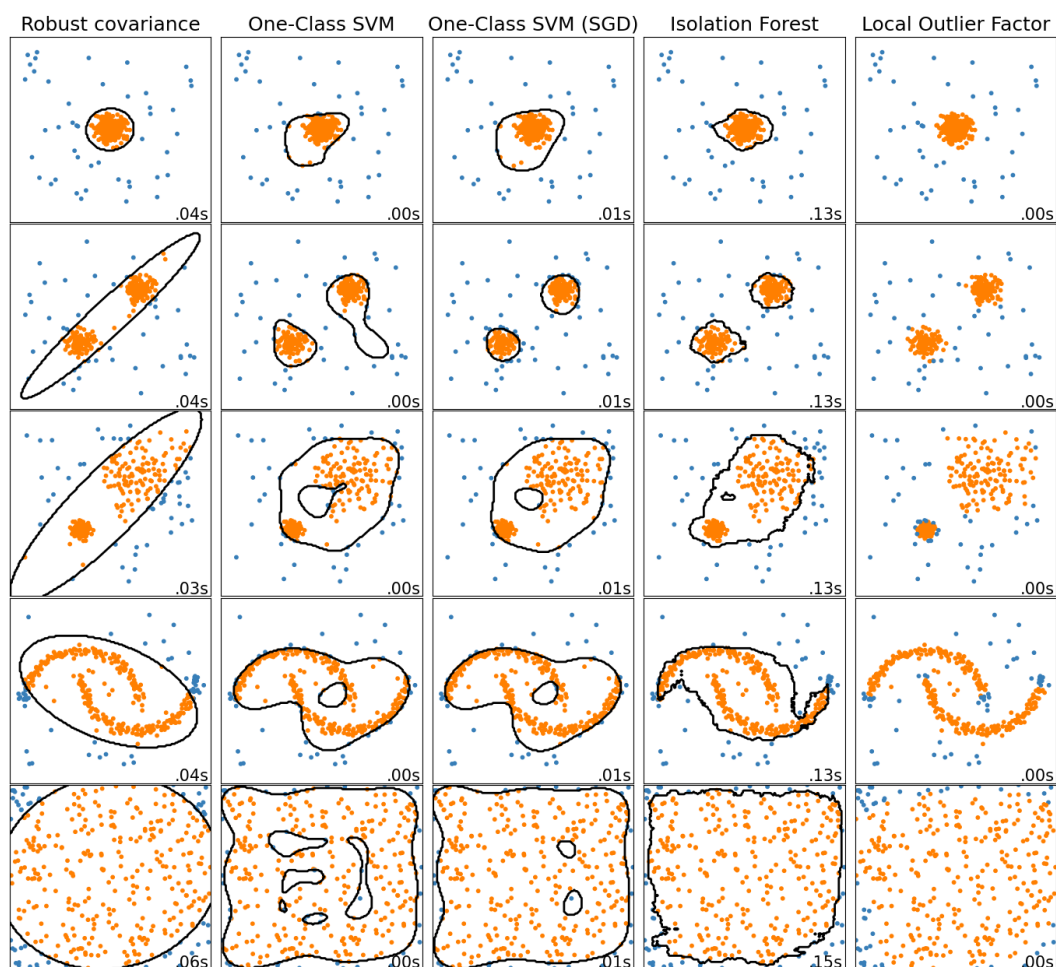
مثال ۱: تشخیص داده‌های پرت ساختگی. این مثال به خوبی عملکرد پایه‌ای الگوریتم را نشان می‌دهد. در این کد، مجموعه‌ای از داده‌های «نرمال» (inliers) که از یک توزیع مشخص پیروی می‌کنند، به همراه تعدادی داده «پرت» (outliers) که به صورت تصادفی تولید شده‌اند، ایجاد می‌شوند. سپس مدل One-Class SVM روی این داده‌ها آموزش داده می‌شود.



شکل ۶: خروجی مثال Scikit-learn برای One-Class SVM. شکل سمت چپ داده‌های نرمال (سفید) و پرت (نارنجی) شناسایی شده را نشان می‌دهد. شکل سمت راست، مرز تصمیم (خط سیاه) و ناحیه داده‌های نرمال (ناحیه رنگی) را به تصویر می‌کشد.

در نمودار ۶، ناحیه رنگی نشان‌دهنده منطقه‌ای است که توسط مدل به عنوان «نرمال» شناخته می‌شود و خط سیاه مرز این ناحیه است. رنگ پس‌زمینه در واقع مقدار تابع تصمیم را نشان می‌دهد؛ هرچه یک نقطه در عمق بیشتری از ناحیه رنگی باشد، مقدار این تابع برای آن مثبت‌تر است و با اطمینان بیشتری نرمال تلقی می‌شود. نقاط سفید، داده‌های آموزشی نرمال (inliers) هستند که به درستی درون مرز قرار گرفته‌اند. نقاط نارنجی، داده‌های پرتی (outliers) هستند که مدل به درستی آن‌ها را خارج از مرز تشخیص داده است. نکته مهم این است که مدل توانسته یک مرز غیرخطی و بسته در اطراف توده اصلی داده‌ها ایجاد کند و نقاط پراکنده را به عنوان ناهنجاری شناسایی نماید. پارامتر 'nu' در این مدل (که در مثال برابر ۱.۰ تنظیم شده) یک حد بالا برای کسر خطاهای آموزشی (داده‌های نرمالی که اشتباهاً پرت تشخیص داده شوند) و یک حد پایین برای کسر بردارهای پشتیبانی که برای تعریف این مرز استفاده شده‌اند، تعیین می‌کند.

مثال ۲: مقایسه عملکرد روی توزیع‌های مختلف داده. این مثال نشان می‌دهد که One-Class SVM چگونه با توزیع‌های مختلف داده سازگار می‌شود. این امر به ویژه در تحلیل داده‌های نوروفیزیولوژیک که ممکن است حالت «نرمال» شامل چندین زیرحالت باشد، اهمیت دارد. شکل ۷ به زیبایی قدرت و انعطاف‌پذیری کرنل RBF را به نمایش می‌گذارد:



شکل ۷: مقایسه عملکرد One-Class SVM روی دو مجموعه داده متفاوت. در داده‌های سمت چپ که یک خوشه کلی را تشکیل می‌دهد، یک مرز کلی ایجاد می‌شود. در داده‌های سمت راست که از دو خوشه مجزا تشکیل شده‌اند، مدل دو مرز جداگانه برای هر خوشه می‌آموزد.

- نمودار سمت چپ: داده‌ها از یک توزیع گوسی واحد و یکپارچه آمده‌اند. مدل به درستی یک مرز کلی و محدب در اطراف این خوشه واحد ایجاد کرده است. این رفتار مورد انتظار برای داده‌های با توزیع ساده است.
- نمودار سمت راست: در اینجا، داده‌های نرمال از دو خوشه مجزا و دور از هم تشکیل شده‌اند. یک الگوریتم ساده ممکن است تلاش کند یک دایره بزرگ پیرامون هر دو خوشه بکشد که این کار باعث می‌شود فضای خالی بین دو خوشه نیز به اشتباه به عنوان ناحیه نرمال در نظر گرفته شود. اما One-Class SVM با کرنل RBF، به دلیل ماهیت «محلی» خود (که شباهت را بر اساس نزدیکی می‌سنجد)، قادر است این ساختار چندوجهی را تشخیص داده و دو مرز تصمیم جداگانه برای هر خوشه بیاموزد. این قابلیت در کاربردهای عملی بسیار ارزشمند است، زیرا حالت «نرمال» سیگنال EEG ممکن است شامل چندین الگوی متمایز باشد (مثلاً حالت چشم بسته در مقابل چشم باز در وضعیت استراحت) و الگوریتم باید بتواند تمام این زیرحالت‌ها را به عنوان نرمال شناسایی کند.

در نمودار ۷، هر سطر یک الگوریتم متفاوت را نشان می‌دهد. تحلیل هر یک از این الگوریتم‌ها به شرح زیر است:

- **Robust Covariance** (کواریانس مقاوم): این یک روش آماری است که فرض می‌کند داده‌های نرمال از یک توزیع گوسی پیروی می‌کنند. الگوریتم تلاش می‌کند تا یک بیضی (که نمایانگر توزیع گوسی است) را به داده‌ها برازش دهد، اما به گونه‌ای که داده‌های پرت تأثیر کمی روی شکل بیضی داشته باشند. همانطور که دیده می‌شود، این روش برای داده‌های خوشه‌ای و گوسی شکل (ستون چپ) مناسب است، اما در مدل‌سازی

داده‌های با ساختار پیچیده‌تر و غیرمحدب (مانند دو خوشه در ستون راست) شکست می‌خورد و یک بیضی بزرگ پیرامون هر دو می‌کشد.

● **One-Class SVM:** این همان الگوریتم مبتنی بر کرنل است که پیشتر به تفصیل شرح داده شد. با استفاده از کرنل RBF، این الگوریتم مرزهای بسیار انعطاف‌پذیر و نرمی را می‌آموزد. در ستون چپ، یک مرز فشرده و دقیق دور خوشه اصلی ایجاد کرده است. نقطه قوت اصلی آن در ستون راست نمایان می‌شود، جایی که به درستی دو مرز جداگانه برای هر خوشه آموخته و فضای خالی بین آن‌ها را به عنوان ناحیه ناهنجاری در نظر گرفته است. این قابلیت، آن را برای داده‌های با ساختار پیچیده بسیار قدرتمند می‌سازد.

● **One-Class SVM (SGD):** این نسخه، یک پیاده‌سازی متفاوت از One-Class SVM است که برای کار با مجموعه داده‌های بسیار بزرگ طراحی شده است. تفاوت بنیادی آن در روش بهینه‌سازی است:

– روش حل: به جای حل مسئله بهینه‌سازی درجه دوم به صورت دقیق (که از نظر محاسباتی سنگین است)، این نسخه از روش نزول گرادیان تصادفی (Stochastic Gradient Descent - SGD) برای یافتن یک راه‌حل تقریبی استفاده می‌کند.

– ماهیت مدل: پیاده‌سازی SGD معمولاً روی یک مدل خطی اعمال می‌شود. این بدان معناست که حتی اگر داده‌ها در فضای کرنل باشند، بهینه‌سازی به صورت خطی انجام می‌شود. در نتیجه، مرز تصمیم نهایی معمولاً یک شکل ساده مانند دایره یا بیضی خواهد بود.

تحلیل نمودار: همانطور که در هر دو ستون چپ و راست مشاهده می‌شود، مرز تصمیم این مدل یک دایره ساده است. این مدل توانسته خوشه تکی در سمت چپ را به خوبی (هرچند نه به دقت نسخه کرنلی) محصور کند. اما در داده‌های دو خوشه‌ای سمت راست، کاملاً شکست خورده و یک دایره بزرگ پیرامون هر دو خوشه کشیده است که به هیچ وجه توصیف مناسبی از داده‌ها نیست. این نمودار به وضوح توازن بین سرعت و مقیاس‌پذیری (SGD) و دقت و انعطاف‌پذیری (نسخه کرنلی) را نشان می‌دهد.

● **Isolation Forest (جنگل ایزوله):** این الگوریتم بر اساس یک ایده کاملاً متفاوت کار می‌کند: داده‌های پرت نقاطی هستند که «راحت‌تر ایزوله می‌شوند». الگوریتم تعدادی درخت تصمیم تصادفی می‌سازد و میانگین عمقی که برای جداسازی هر نقطه لازم است را محاسبه می‌کند. نقاط پرت به عمق کمتری برای جداسازی نیاز دارند. مرزهای تصمیم این مدل معمولاً به صورت خطوط موازی با محورهای مختصات هستند که باعث ایجاد ظاهری «جعبه‌ای» یا «پلکانی» می‌شود. این روش بسیار سریع است و روی داده‌های با ابعاد بالا به خوبی کار می‌کند.

● **Local Outlier Factor (LOF):** این یک الگوریتم مبتنی بر چگالی است. این روش چگالی محلی یک نقطه را با چگالی محلی همسایگانش مقایسه می‌کند. نقاطی که در ناحیه‌ای با چگالی بسیار کمتر از همسایگان خود قرار دارند، به عنوان داده پرت شناسایی می‌شوند. این الگوریتم در شناسایی داده‌های پرتی که در خوشه‌های با چگالی‌های متفاوت قرار دارند بسیار خوب عمل می‌کند و می‌تواند مرزهای بسیار پیچیده‌ای را مدل کند. همانطور که در نمودارها مشخص است، مرزهای تصمیم آن به خوبی شکل خوشه‌ها را دنبال می‌کنند.

### ۳.۱.۷ مراحل عملی شناسایی آرتیفکت EEG با One-Class SVM

در عمل، می‌توان One-Class SVM را برای پاکسازی سیگنال‌های الکتروفیزیولوژیک به صورت زیر به کار برد:

۱. پیش‌پردازش و قطعه‌بندی: ابتدا سیگنال خام EEG با استفاده از فیلترهای مناسب (مثلاً فیلتر میان‌گذر) از نویزهای فرکانسی پاک شده و سپس به قطعات زمانی کوتاه و هم‌پوشان (که epoch نامیده می‌شوند) تقسیم می‌شود.

۲. استخراج ویژگی: از هر epoch مجموعه‌ای از ویژگی‌های آماری استخراج می‌شود. این ویژگی‌ها باید بتوانند تفاوت بین سیگنال مغزی نرمال و آرتیفکت را به خوبی نشان دهند. نمونه‌هایی از این ویژگی‌ها عبارتند از:



- واریانس یا توان سیگنال
- کجی (Skewness) و کشیدگی (Kurtosis) توزیع دامنه
- توان در باندهای فرکانسی خاص (دلتا، تتا، آلفا، بتا، گاما)

۳. آموزش مدل: مدل One-Class SVM بر روی مجموعه ویژگی‌های استخراج شده از epoch‌هایی که توسط یک متخصص به عنوان «پاک» و «بدون آرتیفکت» برچسب‌گذاری شده‌اند، آموزش داده می‌شود. در این مرحله، پارامترهای مدل مانند  $\nu$  و پارامترهای کرنل (مثلاً  $\gamma$  در RBF) با استفاده از روش‌های اعتبارسنجی متقابل بهینه می‌شوند.

۴. تشخیص آرتیفکت: پس از آموزش، مدل می‌تواند ویژگی‌های استخراج شده از epoch‌های جدید و دیده‌نشده را دریافت کند. اگر بردار ویژگی یک epoch درون مرز آموخته شده قرار گیرد، آن epoch به عنوان «پاک» طبقه‌بندی می‌شود. در غیر این صورت، به عنوان «آرتیفکت» (مانند پلک زدن، حرکت عضله یا نویز الکتریکی) برچسب‌گذاری شده و می‌تواند از تحلیل‌های بعدی حذف گردد.

## ۸ جمع‌بندی: مزایا و معایب SVM

### ۱.۸ مزایا

- عملکرد قوی در فضاهای با ابعاد بالا
- کارایی حافظه به دلیل استفاده از بردارهای پشتیبان
- تطبیق‌پذیری بالا با استفاده از کرنل‌ها

### ۲.۸ معایب

- حساسیت بالا به انتخاب پارامترها ( $\gamma$ ,  $C$ )
- سرعت آموزش پایین برای مجموعه داده‌های بسیار بزرگ
- تفسیرپذیری پایین مدل نهایی (جعبه سیاه)

## مراجع

- Scikit-learn User Guide. (2023). 1.4. Support Vector Machines. Retrieved from <https://scikit-learn.org/stable/modules/svm.html>
- Scikit-learn Examples. (2023). One-class SVM with non-linear kernel (RBF). Retrieved from [https://scikit-learn.org/stable/auto\\_examples/svm/plot\\_oneclass.html](https://scikit-learn.org/stable/auto_examples/svm/plot_oneclass.html)
- Scikit-learn Examples. (2023). Comparing anomaly detection algorithms for outlier detection on toy datasets. Retrieved from [https://scikit-learn.org/stable/auto\\_examples/miscellaneous/plot\\_anomaly\\_comparison.html](https://scikit-learn.org/stable/auto_examples/miscellaneous/plot_anomaly_comparison.html)
- Bishop, C. M. (2006). *Pattern Recognition and Machine Learning*. Springer.
- Vapnik, V. N. (1995). *The Nature of Statistical Learning Theory*. Springer-Verlag.