**بخش ۱۷ ماشین‌های بردار پشتیبان**

**17 مقدمه**

برای درک ماشین‌های بردار پشتیبان، باید ابتدا مفهوم ابرصفحه‌ها را درک کنیم. به طور رسمی، ابرصفحه یک زیرفضای n - 1 بعدی در یک فضای n بعدی است. اگرچه این ممکن است پیچیده به نظر برسد، در واقع به سادگی قابل فهم است. به عنوان مثال، اگر بخواهیم یک فضای دو بعدی را تقسیم کنیم، از یک ابرصفحه یک بعدی (یعنی یک خط) استفاده می‌کنیم. اگر بخواهیم یک فضای سه بعدی را تقسیم کنیم، از یک ابرصفحه دو بعدی (به عبارت دیگر، یک قطعه کاغذ صاف یا تخته) استفاده خواهیم کرد. ابرصفحه به اصطلاح یک تعمیم از این مفهوم به n بعد است.

ماشین‌های بردار پشتیبان با یافتن ابرصفحه‌ای که حاشیه بین دسته‌ها در داده‌های آموزش را حداکثر می‌کند، داده‌ها را دسته‌بندی می‌کنند. در یک مثال دوبعدی با دو دسته، می‌توانیم ابرصفحه را به عنوان «باند» (یعنی یک خط عریض) که دو دسته را از یکدیگر جدا می‌کند، در نظر بگیریم.

در این فصل، ما آموزش ماشین‌های بردار پشتیبان را در موقعیت‌های مختلف پوشش می‌دهیم و به تفصیل به بررسی نحوه‌ی گسترش این رویکرد برای مقابله با مسائل رایج می‌پردازیم.

**17.1 آموزش یک دسته‌بند خطی**

**مشکل:**

شما باید یک مدل را برای دسته‌بندی مشاهدات آموزش دهید.

**راه‌حل:**

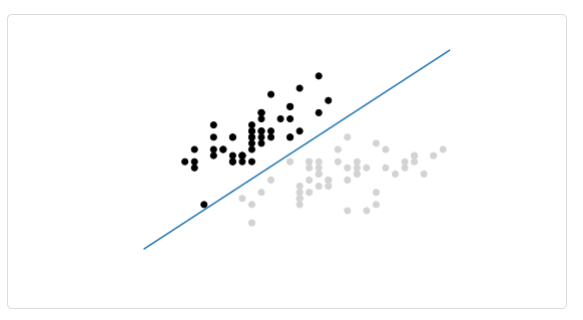
از یک دسته‌بند ماشین بردار پشتیبان (SVC) استفاده کنید تا ابرصفحه‌ای را که حاشیه بین دسته‌ها را حداکثر می‌کند، پیدا کنید.



**بحث:**

کتابخانه‌ی scikit-learn یک LinearSVC را که یک SVC ساده است پیاده‌سازی می‌کند. برای درک بهتر از اینکه یک SVC چه کاری انجام می‌دهد، داده‌ها و ابرصفحه را با نمودار کشیده و تجسم کنیم. در حالی که ماشین‌های بردار پشتیبان در ابعاد بالا به خوبی عمل می‌کنند، در راه‌حل ما فقط دو ویژگی را بارگذاری کرده و یک زیرمجموعه از مشاهدات را انتخاب کرده‌ایم تا داده‌ها تنها دارای دو کلاس باشند. این به ما اجازه می‌دهد که مدل را تجسم کنیم. به خاطر داشته باشید که SVC سعی می‌کند ابرصفحه‌ای را پیدا کند - یک خط در مواقعی که تنها دو بعد داریم - که حاشیه بیشینه بین کلاس‌ها را فراهم می‌کند. در کد زیر، ما دو کلاس را در یک فضای دو بعدی نمودار نمایش می دهیم، سپس ابرصفحه را رسم می‌کنیم:





در این تصویرسازی، تمام مشاهدات دسته 0 به رنگ سیاه و مشاهدات دسته 1 به رنگ خاکستری روشن است. ابرصفحه مرز تصمیم‌گیری است که نشان می‌دهد چگونه مشاهدات جدید دسته‌بندی می‌شوند. به طور خاص، هر مشاهده‌ای که بالای خط قرار داشته باشد به عنوان دسته 0 دسته‌بندی می‌شود در حالی که هر مشاهده‌ای که پایین خط قرار داشته باشد به عنوان دسته 1 دسته‌بندی می‌شود. ما می‌توانیم این را با ایجاد یک مشاهده جدید در گوشه بالا و چپ تصویرسازی‌مان اثبات کنیم، به این معنا که باید پیش‌بینی شود که به عنوان دسته 0 دسته‌بندی شود:



چند نکته درباره ماشین‌های بردار پشتیبان وجود دارد. اولاً، به منظور تصویرسازی، ما مثال خود را به یک مثال دودویی محدود کردیم (به عبارت دیگر، تنها دو دسته)؛ با این حال، ماشین‌های بردار پشتیبان به خوبی با چندین دسته کار می‌کنند. ثانیاً، همانطور که تصویرسازی ما نشان می‌دهد، ابرصفحه به شکل یک خطی است (یعنی منحنی ندارد). این موضوع در این مثال قابل قبول بود زیرا داده‌ها خطی قابل جداشدن بودند، به این معنا که یک ابرصفحه وجود داشت که به طور کامل دو دسته را از یکدیگر جدا می‌کرد. متأسفانه، در دنیای واقعی این موارد بسیار کم رخ می‌دهد.

به طور معمول، ما قادر به جدا کردن کلاس‌ها به صورت کامل نخواهیم بود. در این مواقع، توازنی بین حاشیه‌ بیشینه ابرصفحه توسط SVC و کمینه کردن اشتباه‌ها وجود دارد. در ماشین‌های بردار پشتیبان، این موضوع با استفاده از هایپرپارامتر C که جریمه‌ی خطاها را مشخص می‌کند، کنترل می‌شود. C یک پارامتر از یادگیرنده SVC است و جریمه‌ی دسته‌بندی یک نقطه‌ی داده را مشخص می‌کند. وقتی مقدار C کوچک است، دسته‌بند با داده‏های اشتباه دسته‌بندی‌‌شده مشکل ندارد (بیشترین انحراف و کمترین واریانس). وقتی مقدار C بزرگ است، دسته‌بند به شدت به دلیل اشتباه‌ دسته‏بندی شدن داده‌ها مجازات می‌شود و به همین دلیل تلاش می‌کند تا از هر داده اشتباه دسته‌بندی شده اجتناب کند (کمترین انحراف و بیشترین واریانس).

در scikit-learn، مقدار C با استفاده از پارامتر C تعیین می‌شود و مقدار پیش‌فرض آن C=1.0 است. ما باید C را به عنوان یک هایپرپارامتر الگوریتم یادگیری‌مان در نظر بگیریم، که می‌توانیم از تکنیک‌های انتخاب مدل در فصل 12 برای تنظیم آن استفاده کنیم.

**17.2 مدیریت کلاس‌های خطی تفکیک‌ناپذیر با استفاده از هسته‌ها**

**مشکل:**

شما نیاز به آموزش یک دسته‏بند بردار پشتیبان دارید، اما کلاس‌های شما قابل تفکیک خطی نیستند.

**راه‌حل:**

یک افزونه از ماشین بردار پشتیبان را با استفاده از توابع هسته آموزش دهید تا مرزهای تصمیم غیرخطی ایجاد کنید.



**بحث:**

توضیح کاملی از ماشین‌های بردار پشتیبان خارج از دامنه این کتاب است. با این حال، یک توضیح کوتاه احتمالاً برای درک ماشین‌های بردار پشتیبان و هسته‌ها مفید است. به دلایلی که بهتر است در جاهای دیگر یاد گرفته شوند، یک دسته‌بند بردار پشتیبان می‌تواند به شکل زیر نمایش داده شود:

که در آن پارامتر تعیین کننده بایاس، S مجموعه‌ای از تمام مشاهدات بردار پشتیبان، α پارامترهای مدل برای یادگیری هستند و هر دو مشاهده بردار پشتیبان و هستند. مهم‌ترین نکته این است که K یک تابع هسته است که شباهت بین و را مقایسه می‌کند. اگر توابع هسته را درک نمی‌کنید نگران نباشید. برای هدف‌های ما، فقط بفهمید که K1) نوع ابرصفحه مورد استفاده برای جدا کردن دسته‌ها را تعیین می‌کند و 2) با استفاده از هسته‌های مختلف ابرصفحه‌های مختلف ایجاد می‌کنیم. به عنوان مثال، اگر می‌خواستیم ابرصفحه خطی ابتدایی مشابه آنچه که در [روش 17.1](#روش۱۷۱) ایجاد کردیم را داشته باشیم، می‌توانیم از هسته خطی استفاده کنیم:

که در آن p تعداد ویژگی‌هاست. در هرحال اگر می‌خواستیم یک مرز تصمیم غیرخطی داشته باشیم، هسته خطی را با یک هسته چندجمله‌ای جایگزین می‌کنیم:

که در آن d درجه تابع هسته چندجمله‌ای است.

همچنین، می‌توانیم از یکی از معمول‌ترین هسته‌ها در ماشین‌های بردار پشتیبان استفاده کنیم، که هسته تابع توابع پایه شعاعی (RBF) است:

که γ یک هایپرپارامتر است و باید بیشتر از صفر باشد. نکته اصلی توضیحات قبلی این است که اگر داده‌هایی داریم که به صورت خطی قابل جداسازی نیستند، می‌توانیم یک هسته خطی را با یک هسته جایگزین جهت ایجاد یک مرز تصمیم غیرخطی با ابرصفحه تصمیم جایگزین کنیم.

می‌توانیم از طریق تصویرسازی یک مثال ساده، به اندیشه‌های پشت توابع هسته پی ببریم. این تابع که بر اساس یک تابع از Sebastian Raschka بوده، مشاهدات و ابرصفحه مرز تصمیم در یک فضای دو بعدی را نمایش می‌دهد. شما نیازی به درک دقیق از نحوه عملکرد این تابع ندارید؛ من آن را اینجا ضمیمه کرده‌ام تا خودتان بتوانید تجربه کنید.

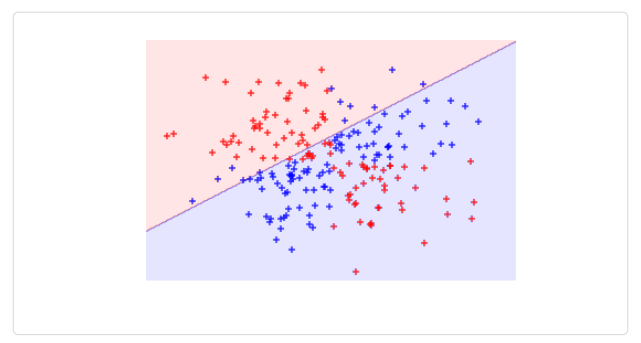


در راه‌حل ما، داده‌هایی داریم که دارای دو ویژگی (یعنی دو بعد) هستند و یک بردار هدف با کلاس هر مشاهده را داریم. نکته مهم این است که کلاس‌ها به نحوی تخصیص داده شده‌اند که به صورت خطی قابل جداسازی نیستند. به عبارت دیگر، خط ساده‌ای وجود ندارد که بتواند دو کلاس را از یکدیگر جدا کند. ابتدا، بیایید یک دسته‌بند ماشین بردار پشتیبان با هسته خطی ایجاد کنیم:



سپس، از آنجایی که تنها دو ویژگی داریم، در یک فضای دو بعدی کار می‌کنیم و می‌توانیم مشاهدات، کلاس‌هایشان و ابرصفحه خطی مدل خود را تصویرسازی کنیم:



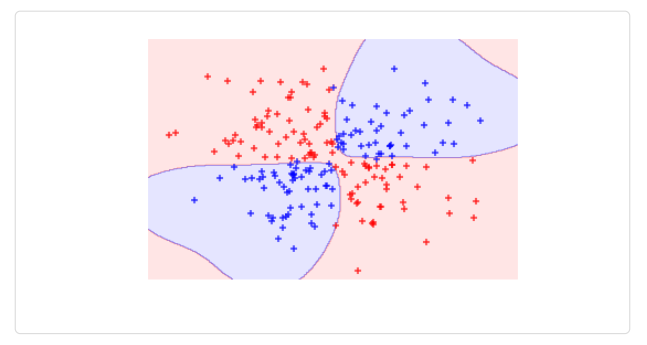


همانطور که می‌بینیم، ابرصفحه خطی ما در تقسیم دو کلاس به خوبی عمل نکرد! حالا، بیایید هسته خطی را با هسته تابع توابع پایه شعاعی (RBF) جایگزین کنیم و از آن برای آموزش مدل جدید استفاده کنیم:



و سپس مشاهدات و ابرصفحه را تصویرسازی کنیم:





با استفاده از هسته تابع توابع پایه شعاع (RBF)، ما می‌توانیم یک مرز تصمیم‏گیری ایجاد کنیم که نسبت به هسته خطی، نتیجه بهتری در جداسازی دو کلاس ارائه می‌دهد. این انگیزه‏ی استفاده از هسته‌ها در ماشین‌های بردار پشتیبان است.

در scikit-learn، می‌توانیم هسته‌ای که می‌خواهیم استفاده کنیم را با استفاده از پارامتر kernel انتخاب کنیم. پس از انتخاب هسته، باید تنظیمات مناسب هسته را مشخص کنیم، مانند مقدار d (با استفاده از پارامتر degree) در هسته‌های چندجمله‌ای و مقدار γ (با استفاده از پارامتر gamma) در هسته‌های تابع توابع پایه شعاعی. همچنین، باید پارامتر جریمه C را تنظیم کنیم. هنگام آموزش مدل، در اکثر موارد باید همه این پارامترها را به عنوان هایپرپارامترها در نظر بگیریم و از تکنیک‌های انتخاب مدل برای شناسایی ترکیب مقادیری که مدل با بهترین عملکرد را تولید می‌کند، استفاده کنیم.

**17.3 ایجاد احتمالات پیش‌بینی شده**

**مشکل:**

شما به احتمالات کلاس پیش‌بینی شده برای یک مشاهده نیاز دارید.

**راه حل:**

وقتی از SVC در scikit-learn استفاده می‌کنید، پارامتر probability را برابر با True قرار دهید، مدل را آموزش دهید، سپس از تابع predict\_proba برای مشاهده احتمالات تنظیم‌شده استفاده کنید:



**بحث:**

بسیاری از الگوریتم‌های یادگیری نظارت شده که مورد بحث قرار گرفته‌اند، از تخمین‌های احتمالی برای پیش‌بینی کلاس‌ها استفاده می‌کنند. به عنوان مثال، در مدل k نزدیکترین همسایه، کلاس‌های k همسایه یک مشاهده به عنوان رای‌ها تلقی می‌شوند تا احتمالی که مشاهده در آن کلاس قرار دارد، ایجاد شود. سپس کلاس با بیشترین احتمال پیش‌بینی می‌شود. استفاده از ابرصفحه در SVC برای ایجاد مناطق تصمیم، به طور طبیعی تخمینی از احتمال برای عضویت مشاهده در یک کلاس معین خروجی نمی‌دهد. با این حال، در واقع می‌توانیم با توجه به توضیحاتی که آمده، احتمالات کلاس تنظیم‌شده را با مقداری خطا در خروجی ارائه دهیم. در یک SVC با دو کلاس، می‌توان از مقیاس‌بندی پلات (Platt Scaling) استفاده کرد. در این روش، ابتدا SVC آموزش داده می‌شود، سپس یک رگرسیون لجستیک اعتبارسنجی شده جداگانه آموزش داده می‌شود تا خروجی‌های SVC را به احتمالات تجسمی تبدیل تصویر کند:

جایی که A و B بردارهای پارامتر هستند و f فاصله علامت زده شده برای مشاهده i ام از ابرصفحه است. وقتی بیش از دو کلاس داریم، یک نوع توسعه‏ یافته از مقیاس‌بندی پلات استفاده می‌شود.

در اصطلاحات عملی‌تر، ایجاد احتمالات پیش‌بینی دو مشکل اصلی دارد. اولاً، زیرا ما داریم یک مدل دوم با استفاده از تصویر کردن آموزش می‌دهیم، تولید احتمالات پیش‌بینی می‌تواند زمان زیادی را برای آموزش مدل اختصاص دهد. ثانیاً، به دلیل اینکه احتمالات پیش‌بینی با استفاده از تصویر کردن ایجاد می‌شوند، ممکن است همیشه با کلاس‌های پیش‌بینی شده مطابقت نداشته باشند. به عبارت دیگر، ممکن است مشاهده‌ای به عنوان کلاس 1 پیش‌بینی شود، اما احتمال پیش‌بینی شده برای کلاس 1 بودن کمتر از 0.5 باشد.

در scikit-learn، احتمالات پیش‌بینی باید در زمان آموزش مدل تولید شوند. برای این کار، ما می‌توانیم با تنظیم مقدار True برای ویژگی probability در SVC، احتمالات پیش‌بینی را ایجاد کنیم. پس از آموزش مدل، می‌توانیم با استفاده از تابع predict\_proba، احتمالات تخمینی برای هر کلاس را خروجی دهیم.

**17.4 شناسایی بردارهای پشتیبان**

**مشکل:**

شما نیاز دارید تا تشخیص دهید که کدام مشاهدات، بردارهای پشتیبان از ابرصفحه تصمیم‏گیری هستند.

**راه‌حل:**

مدل را آموزش دهید، سپس از ویژگی support\_vectors\_ استفاده کنید.



**بحث:**

ماشین‌های بردار پشتیبان نام خود را به دلیل این واقعیت به‌دست آورده‌اند که ابرصفحه تصمیم‏گیرنده توسط تعداد نسبتاً کمی از مشاهدات، که به آن‌ها بردارهای پشتیبان می‌گویند، تعیین می‌شود. به طور استدلالی، ابرصفحه را می‌توان به عنوان یک «باروری» توسط این بردارهای پشتیبان تصور کرد. این بردارهای پشتیبان به همین دلیل برای مدل ما بسیار مهم هستند. به عنوان مثال، اگر یک مشاهده که بردار پشتیبان نیست را از داده حذف کنیم، مدل تغییر نمی‌کند؛ اما اگر یک بردار پشتیبان را حذف کنیم، ابرصفحه دیگر دارای حاشیه بیشینه نخواهد بود.

بعد از آموزش یک ماشین بردار پشتیبانی (SVC)، scikit-learn به ما چندین گزینه برای شناسایی بردارهای پشتیبان ارائه می‌دهد. در راه‌حل ما، ما از ویژگی support\_vectors\_ برای خروجی دادن ویژگی‌های واقعی مشاهدات چهار بردار پشتیبان در مدل خود استفاده کردیم. به طور جایگزین، ما می‌توانیم با استفاده از ویژگی support\_ شاخص‌های بردارهای پشتیبان را مشاهده کنیم.



در نهایت، می‌توانیم از n\_support\_ استفاده کنیم تا تعداد بردارهای پشتیبان متعلق به هر کلاس را پیدا کنیم:



**17.5 مدیریت کلاس‌های نامتوازن**

**مشکل:**

شما نیاز دارید که در حضور کلاس‌های نامتوازن، یک مدل دسته‌بندی ماشین بردار پشتیبان آموزش دهید.

**راه‌حل:**

با استفاده از class\_weight، جریمه را برای اشتباه ‌دسته‏بندی کردن کلاس کوچکتر افزایش دهید.



**بحث:**

در ماشین‌های بردار پشتیبان، C یک هایپرپارامتر است که جریمه اشتباه طبقه‏بندی‌کردن یک مشاهده را تعیین می‌کند. یکی از روش‌های مدیریت کلاس‌های نامتوازن در ماشین‌های بردار پشتیبان، تنظیم وزن C بر اساس کلاس‌ها است، به طوری که:

که C جریمه طبقه‏بندی اشتباه است، وزنی است که به طور معکوس به فراوانی کلاس j وابسته است، و  مقدار C برای کلاس j است. ایده کلی این است که جریمه اشتباه کلاس‌های با فراوانی کم را افزایش داده تا از غلبه کلاس اکثریت جلوگیری شود.

در scikit-learn، وقتی از SVC استفاده می‌کنیم، می‌توانیم مقادیر را به طور خودکار با تنظیم class\_weight='balanced' تنظیم کنیم. آرگومان balanced به طور خودکار وزن‌دهی به کلاس‌ها را انجام می‌دهد به نحوی که:

که وزن مرتبط با کلاس j، n تعداد مشاهدات، تعداد مشاهدات در کلاس j و k تعداد کل کلاس‌هاست.