**4- ماشین بردار پشتیبان**

**1-4- مقدمه­ای بر نظریة یادگیری آماری**

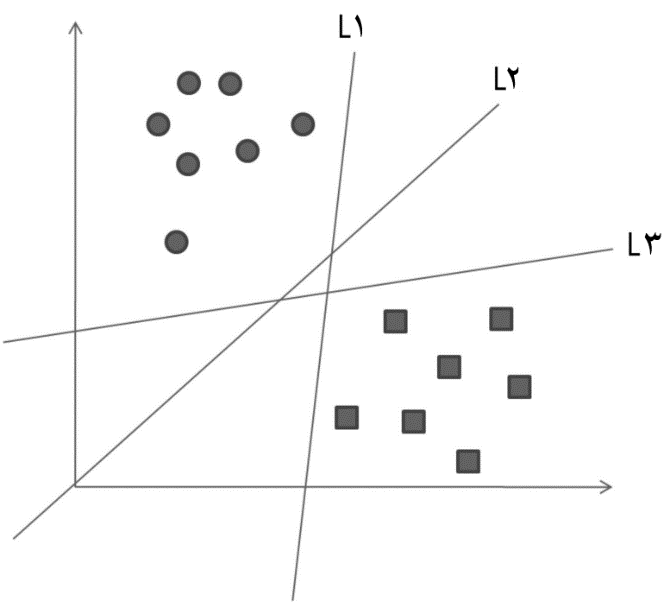
یادگیری آماری به معنای مجموعه­ای وسیع از کاربردهای ریاضی برای فهم داده­ها است. این کاربردها می­توانند به­عنوان نظارت­شده یا بدون نظارت دسته­بندی شوند. در بیانی کلی، یادگیری آماری نظارت­شده شامل ساخت یک مدل آماری برای پیش­بینی، یا تخمین یک خروجی براساس یک یا چند ورودی می­شود. مسائلی از این دست در زمینه­های گوناگونی مانند تجارت، داروسازی، اخترفیزیک[[1]](#footnote-1) و سیاست­گذاری اجتماعی پیش می­آیند. در یادگیری آماری بدون نظارت، ورودی­هایی وجود دارند اما هیچ خروجی نظارت­شده­ای وجود ندارد؛ با این حال می­توان به­کمک چنین داده­هایی، روابط و ساختار را یاد گرفت. هدف اصلی نظریة یادگیری آماری، ایجاد چارچوبی برای مطالعۀ مسئلة استنتاج است (به عبارت دیگر، کسب دانش، پیش­بینی کردن، تصمیم­گیری و ساخت مدل­ها از یک مجموعة داده­ها). مطالعه در یک چارچوب آماری صورت می­گیرد؛ که یعنی فرضیاتی با ماهیت آماری درباره پدیده­های اساسی وجود دارند (به­گونه­ای که داده تولید می­شود). در واقع، یک نظریة استنتاجی بایستی قادر باشد تا تعریفی صوری[[2]](#footnote-2) از کلماتی مانند یادگیری، تعمیم و بیش­برازش[[3]](#footnote-3) ارائه کند و همچنین عملکرد الگوریتم­های یادگیری را به­گونه­ای توصیف کند که در نهایت، بتواند به طراحی بهتر الگوریتم­های یادگیری کمک کند. دو هدف وجود دارد: دقیق­تر کردن و به­دست آوردن الگوریتم­های جدید یا بهبودیافته.

ماشین بردار پشتیبان[[4]](#footnote-4) (SVM) برای دسته­بندی الگوهای متنوع و مسائل تقریب توابع معرفی می­شود. دسته­بندی الگو برای دسته­بندی برخی اشیاء در یک دستة مشخص به نام کلاس­ها به­کار می­رود. دسته­بندی­کننده، که نرم­افزاری کامپیوتری است، به­گونه­ای توسعه یافته است که اشیاء به­طرز صحیحی با دقتی قابل­قبول دسته­بندی شوند. داده­های وارد شده به یک دسته­بندی­کننده، ویژگی­ها نامیده می­شوند و آن­ها هر کلاس را به­خوبی نشان می­دهند و یا داده­های متعلق به کلاس­های گوناگون را به­خوبی در فضای ورودی تفکیک می­کنند.

**2-4- دسته­بندی بردارهای پشتیبان**

**1-2-4- SVM سخت­حاشیه[[5]](#footnote-5)**

SVM سخت­حاشیه تنها زمانی می­تواند کار کند که داده­ها بدون هیچ خطایی (اختلال یا دادة پرت) از لحاظ خطی کاملاً تفکیک­پذیر باشند. در مورد خطاها، یا حاشیه باید کوچکتر باشد یا SVM سخت­حاشیه شکست می­خورد.( In case of errors, either the margin is smaller or the hard margin SVM fails.) SVM نرم­حاشیه[[6]](#footnote-6) توسط وپنیک[[7]](#footnote-7) برای حل این مسئله با معرفی متغیرهای کمکی پیشنهاد شد. روش SVM، یک دسته­بندی­کننده است که یک ابرصفحه[[8]](#footnote-8) یا تابعی که به­شکل صحیحی دو کلاس را با حاشیه­ای حداکثری تفکیک کند، پیدا می­کند. روشی دیگر برای افزایش قدرت SVM، استفاده از خطای حاشیة سخت است، که در آن تعداد دسته­بندی نادرست کمینه شده است. پیچیدگی محاسباتی استفاده از خطای حاشیة سخت، اغلب به­عنوان توجیهی برای اندازه­گیری پیوستة خطا استفاده شده است. با استفاده از فرایند­های کاوشی[[9]](#footnote-9) برای حل نمونه­هایی که بهینگی سراسری را تضمین نمی­کنند، می­توان SVM گسسته[[10]](#footnote-10) (DSVM) که از خطای حاشیة سخت برای SVM با هستة خطی و عبارت حاشیة خطی­شده استفاده می­کند را فرمول­بندی کرد. محققین اندکی فرمول­بندی و تکنیک خود را برای SVM نرم­حاشیه و SVM فازی بسط داده­اند. بروکس[[11]](#footnote-11) در سال 2011، فقدان حاشیة سخت برای SVM را به­کمک توابع پیوسته تقریب زده است و از روش بازوزنی­شده و تکرارشوندة حداقل مربعات برای حل نمونه­هایی که بهینگی سراسری را تضمین نمی­کنند، استفاده کرد. شکل 4-1، دسته­بندی­کننده­های خطی (ابرصفحه) را در فضاهای دو بعدی نمایش می­دهد.



شکل 4-1: دسته­بندی­کننده­های خطی (ابرصفحه) در فضای دو بعدی.

**2-2-4- SVM نرم­حاشیه**

دسته­بندی­کننده با حاشیة سخت و حداکثری[[12]](#footnote-12)، مفهومی مهم است، اما دو مشکل دارد. اولاً، دسته­بندی­کنندة سخت­حاشیه می­تواند بسیار انعطاف­ناپذیر باشد، زیرا هرگونه اشتباه در علامت­گذاری روی بردارهای پشتیبان منجر به تغییر قابل­ملاحظه­ای در ابرصفحة تصمیم­گیری می­شود؛ ثانیاً،­ داده­های آموزشی همیشه به­طور خطی تفکیک­پذیر نیستند و در این صورت، مجبور به استفاده از هستة قوی­تری که منجر به بیش­برازش می­شود، هستید. برای اینکه بتوان اختلالات و داده­های خارج از محدوده را تحمل کرد، بایستی موقعیت­های نمونه­های آموزشی بیشتری از آن­هایی که در نزدیک­ترین موقعیت نسبت به مرز قرار دارند را در نظر بگیرید. این کار معمولاً با معرفی متغیرهای کمکی و یک دسته­بندی­کنندة نرم حاشیه انجام می­شود.

**3-2-4- نگاشت به فضاهایی با­ ابعاد بالا**

*1-3-2-4- ترفندهای هسته[[13]](#footnote-13)*

یکی از اجزای اساسی SVM، به اصطلاح ترفند هسته برای محاسبة ضرب­های نقطه­ای در فضاهای ویژگی با ابعاد بالا با استفاده از توابعی ساده که روی جفت الگوهای ورودی تعریف شده­اند، است. این ترفند اجازة فرمول­بندی انواع گوناگون غیرخطی از هر الگوریتمی که بتواند در قالب عبارات ضرب نقطه­ای قرار گیرد را می­دهد؛ اما SVMها برجسته­ترین مثال هستند. اگرچه تقریباً یک قرن است که نتیجة ریاضی نهفته در ترفند هسته شناخته شده است، اما چندی پیش بود که برای جامعة یادگیری ماشین ثمربخش محسوب شد. از آن زمان، روش­های هسته منجر به تعمیم­های جالب توجهی از الگوریتم­های یادگیری و کاربردهای موفق آن در دنیای واقعی شده است (شولکوپف[[14]](#footnote-14)، 2001). شکل 4-2، تفکیک کلاس را نمایش می­دهد.

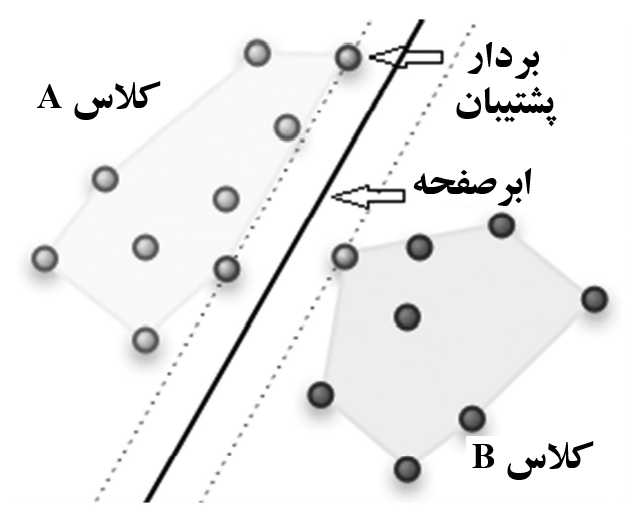
**توابع هسته:** ایدة تابع هسته، ممکن کردن عملیات­هایی است که قرار است ترجیحاً به­جای فضای ویژگی با ابعاد بالا، در فضای ورودی صورت بگیرند. بنابراین، ضرب داخلی نیازی به ارزیابی در فضای ویژگی ندارد. تابع مورد نظر می­بایست نگاشت خصوصیات فضای ورودی به فضای ویژگی را انجام دهد. تابع هسته نقشی حیاتی در SVM و عملکردش ایفا می­کند و بر مبنای بازتولید فضاهای هستة هیلبرت[[15]](#footnote-15) است.

|  |  |
| --- | --- |
| (1-4) |  |

|  |  |
| --- | --- |
| (2-4) |  |

|  |  |
| --- | --- |
| (3-4) |  |

اگر K تابع همیشه مثبت متقارن باشد، که شرایط مرسر[[16]](#footnote-16) را برآورده می­کند، آنگاه هسته، یک ضرب داخلی منطقی در فضای ویژگی را ارائه می­کند. مجموعة آموزشی به­طور خطی در فضای ورودی تفکیک­پذیر نیست؛ اما این مجموعه در فضای ویژگی تفکیک­پذیر است. این موضوع "ترفند هسته" نامیده می­شود.



شکل 4-2: تفکیک کلاس­ها.

**هسته­ها:** مزایای اصلی SVM این است که عملکرد تعمیم می­تواند با انتخاب هسته­های مناسب بهبود یابد. بنابراین، انتخاب هسته­ها برای کاربردهای ویژه، حیاتی است.

**هسته­های خطی:** اگر مسئلة دسته­بندی در فضای ورودی به­طور خطی تفکیک­پذیر باشد، آنگاه نگاشت فضای ورودی در فضای ابعاد بالا مورد نیاز نخواهد بود.

|  |  |
| --- | --- |
| (4-4) |  |

**هستة چندجمله­ای:** هستة چندجمله­ای از درجة d، که d عددی طبیعی است، به­صورت زیر داده شده است:

|  |  |
| --- | --- |
| (5-4) |  |

در این­جا، عدد یک اضافه می­شود تا همة عبارات ضرب­شده با درجه­های کوچکتر یا مساوی d را شامل شود.

هنگامی­که باشد، هسته برابر هستة خطی به علاوه یک است. بنابراین با تنظیم b در تابع تصمیم، هم­ارز هستة خطی می­شود.

1. چندجمله­ای: نگاشت چندجمله­ای، روشی متداول در مدل­سازی غیرخطی است. معمولاً هستة دوم ترجیح داده می­شود، زیرا از مشکلات مربوط به صفر شدن هسین[[17]](#footnote-17) اجتناب می­کند.

|  |  |
| --- | --- |
| (4-6) |  |

|  |  |
| --- | --- |
| (7-4) |  |

1. تابع اساس شعاعی گاوسی: توابع اساس شعاعی که متداول­ترین حالت آن در فرم گاوسی است.

|  |  |
| --- | --- |
| (4-8) |  |

1. تابع اساس شعاعی نمایی: یک تابع اساس شعاعی[[18]](#footnote-18) (RBF)، یک پاسخ خطی تکه­ای ایجاد می­کند که می­تواند هنگامی­که ناپیوستگی­ها قابل­قبول هستند، جالب توجه باشد.

|  |  |
| --- | --- |
| (4-9) |  |

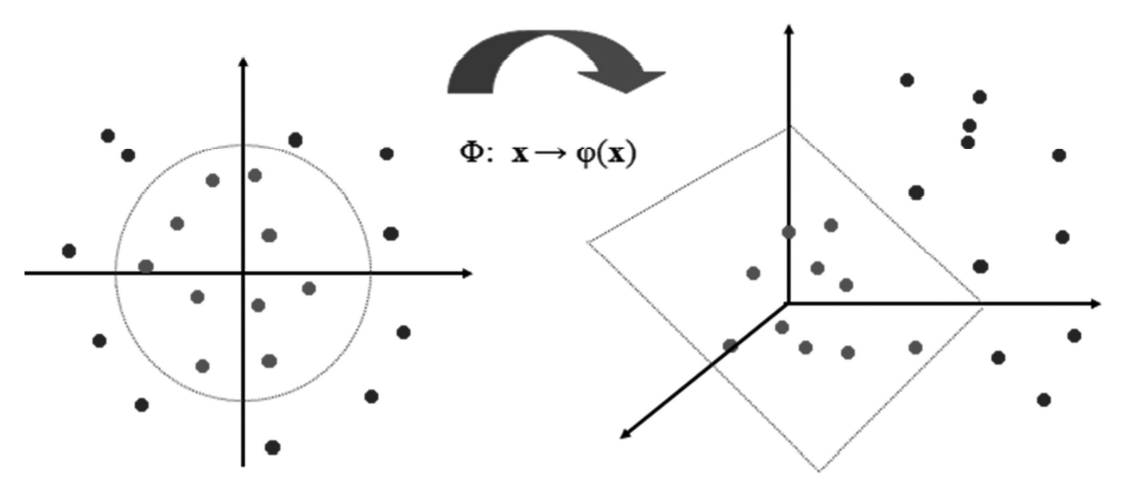
1. پرسپترون چندلایه: پرسپترون چندلایه[[19]](#footnote-19) (MLP) با یک لایة منفرد پنهان که مدت­ها پابرجا بوده، نیز نمایش هسته­ای معتبری دارد.

|  |  |
| --- | --- |
| (10-4) |  |

روش­های بسیار دیگری از جمله فوریه[[20]](#footnote-20)، اسپلاین­ها[[21]](#footnote-21)، بی-اسپلاین­ها[[22]](#footnote-22)، هسته­های افزودنی[[23]](#footnote-23) و ضرب تانسوری[[24]](#footnote-24) وجود دارند. اگر می­خواهید مطالب بیشتری را دربارة توابع هسته مطالعه کنید، می­توانید به هر کتابی در رابطه با SVM مراجعه کنید. شکل 4-3، ترفند هسته را نمایش می­دهد.

*2-3-2-4- هسته­های نرمال­ساز[[25]](#footnote-25)*

هر متغیر ورودی معنای فیزیکی متفاوتی دارد و بنابراین محدودة متفاوتی دارد. در محدودة ورودی اولیه و بدون استفاده از هسته­های مناسب، ممکن است نتیجه بهینه نباشد، زیرا SVMها برای انتقال خطی ورودی­ها یکسان و ثابت نیستند. بنابراین، برای انجام کاری که باعث شود هر متغیر ورودی به­طور برابری در دسته­بندی ایفای نقش کند، نرمال­سازی متغیرهای ورودی چه با مقیاس­بندی محدوده و چه با سفیدسازی[[26]](#footnote-26) الزامی است.



شکل 4-3: ترفند هسته.

اگر در جاهایی که عدد بسیار بزرگ است، متغیرهای ورودی نرمال­سازی شوند، مقدار هسته بسیار کوچک یا بسیار بزرگ می­شود که دسته­بندی SVM را دشوار می­کند. برای غلبه بر این مشکل، نرمال­سازی هسته­ها بسیار توصیه شده است.

**4-2-4- خواص توابع نگاشت مرتبط با هسته­ها**

برای شناخت روابط همسایگی میان داده­های آموزشی نگاشت­شده، بایستی فواصل اقلیدسی[[27]](#footnote-27) در فضای ورودی به­جای فضای ویژگی محاسبه شوند. همچنین، نزدیک­ترین همسایگی­ها به *k* در فضای ویژگی معادل با آن­ها در فضای ورودی است.

**5-2-4- خلاصه**

روابط همسایگی حفظ­کننده[[28]](#footnote-28)، نگاشت یک به یک، نامحدب بودن نواحی نگاشت­شده، عبارات سوگیری ضمنی[[29]](#footnote-29) و فضای ویژگی تجربی برخی از مؤلفه­های وابسته به توابع نگاشت و مرتبط با هسته­ها هستند.

**3-4- SVM چندکلاسه**

**1-3-4- مقدمه**

ماشین­های بردار پشتیبان که یک بسط از مسائل چندکلاسه هستند، از توابع تصمیم مستقیم استفاده می­کنند و ساده نیستند. انواع گوناگونی از SVMهایی که به مسائل چندکلاسه می­پردازند، در ذیل آورده شده است.

1. SVM یک در مقابل همه[[30]](#footnote-30)
2. SVM دوبه­دو[[31]](#footnote-31)
3. SVM کد خروجی تصحیح خطا[[32]](#footnote-32)
4. SVM همه به یکباره[[33]](#footnote-33)

در SVM یک در مقابل همه، یک مسئله با n کلاس به دو مسئلة n کلاسه تبدیل می­شود؛ در مسائل دوکلاسة i ام، کلاس i از کلاس­های باقی­مانده جدا می­شود.

**2-3-4- SVM قراردادی[[34]](#footnote-34)**

SVM در مسئلة پیش­بینی­ها به­کار برده شده و خود را بالاتر از روش­های رقابتی مانند شبکة عصبی، روش­های چندافتراقی خطی[[35]](#footnote-35) و رگرسیون لجستیک اثبات کرده است. در هر حال، SVM قراردادی، اصل کمینه­سازی ریسک ساختاری را به­کار می­گیرد، به این ترتیب ممکن است ریسک تجربی دسته­بندی نادرست بالا باشد؛ خصوصا هنگامی­که نقطه­ای که قرار است دسته­بندی شود، به ابرصفحه نزدیک باشد.

**3-3-4- SVM مبتنی بر درخت تصمیم­گیری[[36]](#footnote-36)**

SVM مبتنی بر درخت تصمیم­گیری که ماشین­های بردار پشتیبان و درخت تصمیم­گیری را ترکیب می­کند، روشی کارآمد برای حل مسائل چندکلاسه است. یک مشکل این روش این است که تقسیم­بندی فضای ویژگی وابسته به ساختار درخت تصمیم­گیری است و ساختار درخت به­طور تنگاتنگی با عملکرد دسته­بندی­کننده مرتبط است. برای حفظ قابلیت تعمیم سطح بالا، تفکیک­پذیرترین کلاس­ها باید در گره­های بالایی درخت تصمیم­گیری تفکیک شوند. فاصله اغلب به­عنوان معیار تفکیک­پذیری میان کلاس­ها استفاده می­شود، اما فاصلة میان مراکز کلاس­ها نمی­تواند توزیع کلاس­ها را انعکاس دهد. پس از تحلیل ساختار درخت و عملکرد دسته­بندی SVM مبتنی بر درخت تصمیم­گیری، یک معیار تفکیک­پذیری جدید براساس توزیع نمونه­های آموزشی در فضای ویژگی تعریف می­شود. معیار تفکیک­پذیری تعریف­شده در شکل­گیری درخت تصمیم استفاده می­شود و الگوریتمی بهبودیافته برای ماشین بردار پشتیبان مبتنی بر درخت تصمیم­گیری، توسعه داده می­شود.

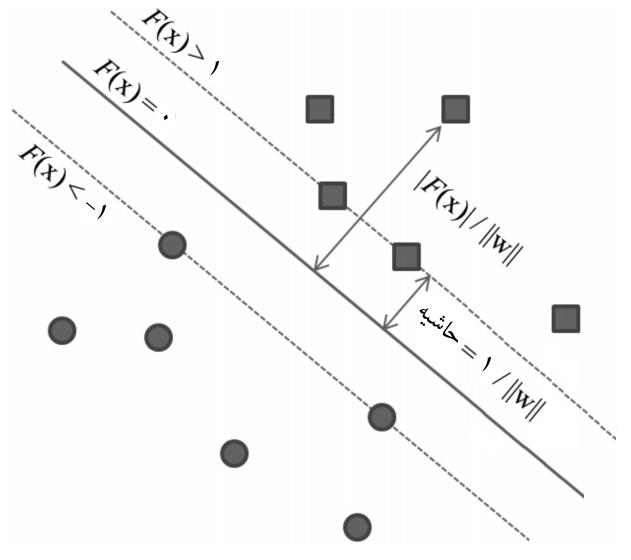
SVM مبتنی بر درخت تصمیم­گیری که بردار ماشین پشتیبان و درخت تصمیم­گیری را ترکیب می­کند، می­تواند روشی کارآمد برای حل مسائل چندکلاسه باشد. این روش می­تواند زمان آموزش و ارزیابی را کاهش دهد و به این ترتیب، کارآمدی سیستم را افزایش دهد. روش­های مختلف ساخت درخت­های باینری، مجموعة داده را به دو زیرمجموعه از ریشه تا برگ تقسیم می­کند تا جایی که هر زیرمجموعه تنها از یک کلاس تشکیل شود. ترتیب ساخت درخت باینری تاثیری شگرف بر عملکرد دسته­بندی دارد. SVM مبتنی بر درخت تصمیم­گیری نیز روشی مناسب برای حل مسائل چندکلاسه است که روش­های SVM و درخت تصمیم­گیری را برای آماده­سازی مدل­های تصمیم­گیری ترکیب می­کند. یکپارچه­سازی مدل­های مختلف، عملکرد بهتری نسبت به مدل­های یادگیری انفرادی یا تصمیم­گیری حاصل می­کند. یکپارچه­سازی، محدودیت­های مدل منفرد را کاهش می­دهد.

**4-3-4- SVM دوبه­دو**

برای بسط دسته­بندی­کننده­های باینری به دسته­بندی چندکلاسه، چندین اصلاح پیشنهاد شده­اند. رویکرد تازه­تری که در زمینة دسته­بندی چندکلاسه و باینری استفاده شده، دسته­بندی دوبه­دو است. این نوع از دسته­بندی به جای یک نمونه، بر دو نمونة ورودی متکی است و پیش­بینی می­کند که آیا دو نمونة ورودی متعلق یه یک کلاس یا کلاس­های متفاوتی هستند؛ این موضوع می­تواند یک مزیت ویژه باشد، اگر تنها یک زیرمجموعه از کلاس­ها برای آموزش شناخته شده باشد. یک ماشین بردار پشتیبان که قادر به انجام وظایف دسته­بندی دوبه­دو است، SVM دوبه­دو نامیده می­شود؛ که در ادامه از این تعریف استفاده خواهد شد. یکی از مقتضیات بدیهی یک دسته­بندی­کننده دوبه­دو این است که ترتیب دو نمونة ورودی نباید بر نتیجة دسته­بندی تأثیر بگذارد (تقارن). یک رویکرد رایج برای اعمال این تقارن، استفاده از هسته­های برگزیده است. در SVMهای دوبه­دو، رویکرد دیگری پیشنهاد شده است. برخی از متخصصان، به­کارگیری مجموعه­های آموزشی با ساختاری متقارن را پیشنهاد می­کنند. در این کتاب، هر دو رویکرد بررسی می­شود تا بتوان به روشی کلی به تقارن دست پیدا کرد. بر این اساس، هنگامی­که این رویکردها به دسته­بندی­کنندة یکسانی منجر شوند، شرایطی قید می­شوند. به علاوه، به­طور تجربی نشان داده می­شود که رویکرد استفاده از هسته­های برگزیده، سه الی چهار برابر در آموزش سریع­تر است. یکی از کارکردهای دسته­بندی دوبه­دوی رایج در مبحث تشخیص چهره پدید می­آید؛ که در آن اغلب اوقات، افراد به تعمیم میان ­کلاس­ها علاقه­مند­ هستند، درحالی­که هیچکدام از اشخاص حاضر در مجموعة آموزش، بخشی از مجموعة آزمایش نیستند. در ادامه نشان داده خواهد شد که مجموعه­های آموزشی با کلاس­های (اشخاص) بسیاری نیاز دارند تا بتوان به عملکردی مناسب در تعمیم میان ­کلاس­ها دست یافت. آموزش چنین داده­هایی از لحاظ محاسباتی پرهزینه است؛ بنابراین پیاده­سازی کارآمدی از SVMهای دوبه­دو در این مبحث مطرح خواهد شد. این کار، آموزش SVMهای دوبه­دو با چندین میلیون جفت را ممکن می­کند. این­گونه برای چهره­های برچسب­گذاری­شده در یک پایگاه دادة نامرتب، عملکردی حاصل می­شود که برتر از به­روزترین مدل­های کنونی است (برونر و همکاران، 2012[[37]](#footnote-37)). شکل 4-4، تابع دسته­بندی SVM را نشان می­دهد: ابرصفحة بیشینه­کنندة حاشیه در یک فضای دو بعدی.

**5-3-4- خلاصه**

در این بخش، کلاس­های متنوعی از SVM شرح داده شدند. SVM قراردادی، SVM مبتنی بر درخت تصمیم­گیری و SVM دوبه­دو کلاس­هایی هستند که مطابق با سازگاری آن­ها مورد استفاده قرار می­گیرند. دسته­بندی دوبه­دو، به­جای یک نمونه، بر دو نمونة ورودی متکی است و پیش­بینی می­کند که آیا دو نمونة ورودی متعلق به یک کلاس هستند یا کلاس­هایی مختلف. برای حفظ توانایی تعمیم سطح بالا، تفکیک­پذیرترین کلاس­ها بایستی در گره­های بالایی یک درخت تصمیم­گیری جدا شوند.



شکل 4-4: تابع دسته­بندی SVM: ابرصفحة بیشینه­کنندة حاشیه در یک فضای دو بعدی.

**4-4- SVMهای متنوع**

**1-4-4- مقدمه**

در بخش­های بعدی، SVMهای متنوعی معرفی می­شوند: SVMهای حداقل مربعات، برنامه­نویسی خطی، تُنک[[38]](#footnote-38)، قدرتمند[[39]](#footnote-39) و بیزی به­طور اجمالی شرح داده خواهند شد که مطابق با سازگاری با مسئله به­کار می­روند.

**2-4-4- SVM حداقل مربعات**

یک نسخة SVM اصلاح­شده، معروف به SVM حداقل مربعات[[40]](#footnote-40) (LSSVM)، تنها محدودیت­های مساوی را به­جای غیرمساوی­ها در نظر می­گیرد. برای اینکه تصمیم­گیری را برای ویژگی­های استخراج­شده از روش مبتنی بر موجک[[41]](#footnote-41) اجرا کرد، روش LSSVM در این­جا به­عنوان یک دسته­بندی­کننده به­کار گرفته شده است. به سبب محدودیت­های مساوی در فرمول­بندی، LSSVM به­جای حل یک مسئلة برنامه­نویسی چهارگانه مانند SVM استاندارد، مجموعه­ای از معادلات خطی را در فضای دوگانه حل می­کند. این موضوع، محاسبه را ساده می­کند و سرعت را به میزان قابل­توجهی افزایش می­دهد. عملکرد SVM عمدتاً وابسته به تابع هسته و بردارهای وزن تنظیم­پذیر است. در هر صورت، هیچ روشی وجود ندارد که به یک فرد اجازه دهد تا تابع هستة مناسبی به­شکلی مستقل از داده­ها برگزیند. هستة RFB به­طور تجربی در این کاربرد انتخاب شده است. روش پیشنهاد شده بر این ایدة اساسی استوار است که برای بهبود عملکرد LSSVM، تفکیک­پذیری الگو یا حاشیة میان خوشه­ها نیاز به تقویت دارد. در این­جا هدف به­روزرسانی بردارهای وزن تنظیم­پذیر در فاز آموزش است، به­گونه­ای که همة نقاط داده به خارج از ناحیة جدایی انتقال یابند و عرض ناحیة تفکیک­پذیر افزایش یابد (آری و همکاران، 2010[[42]](#footnote-42)).

**3-4-4- SVM برنامه­نویسی خطی**

براساس تحلیل نتایج در نظریة یادگیری آماری، خصوصاً بُعد وپنیک-چرووننکیس[[43]](#footnote-43) (VC) از توابع خطی، ماشین­های بردار پشتیبان برنامه­نویسی خطی از جمله برنامه­نویسی خطی، SVMهای خطی و غیرخطی ارائه شده­اند. در SVMهای برنامه­نویسی خطی، برای اینکه سرعت آموزش را بهبود بخشید، مرز بُعد VC به­طور مناسبی آزاد می­شود. نتایج شبیه­سازی برای هر دو دادة مصنوعی و واقعی نشان می­دهند که عملکرد تعمیم در این روش، تقریب خوبی از SVMها است و پیچیدگی محاسبات به مقدار زیادی با بهره­گیری از روش مورد نظر، کاهش می­یابد.

دسته­بندی­کننده­های نرم­حاشیه ماشین بردار پشتیبان، الگوریتم­های یادگیری مهمی برای مسائل دسته­بندی هستند؛ آن­ها می­توانند به­عنوان مسائل بهینه­سازی محدب بیان شوند که برای مجموعة داده­های بزرگ مناسب هستند. دسته­بندی­کنندة SVM برنامه­نویسی خطی، خصوصاً برای نمونه­هایی با اندازة بسیار بزرگ کارآمد است؛ اما در رابطه با همگرایی آن در مقایسه با دسته­بندی­کنندة SVM برنامه­نویسی چهارگانه­ای که به­خوبی درک­ شده باشد، اطلاعات کمی در دست است. مدل کلاسیک SVM، اصطلاحاً SVM نرم­حاشیه با نُرم 1[[44]](#footnote-44)، همراه با هسته­های چندجمله­ای و هسته­های کلی توسط کورتس[[45]](#footnote-45) و وپنیک در سال 1995 معرفی شد. از آن زمان تاکنون، فرم­های متفاوت بسیاری از الگوریتم­های SVM با اهداف گوناگون معرفی شده­اند؛ که SVM برنامه­نویسی خطی[[46]](#footnote-46) (LPSVM) به­علت خطی بودن و انعطاف­پذیری آن در تنظیم داده­های بزرگ از مهم­ترین آن فرم­ها است. اصطلاح "برنامه­نویسی خطی" به این معنا است که الگوریتم براساس بهینه­سازی برنامه­نویسی خطی است. به همین ترتیب، SVM نرم­حاشیه با نُرم 1، به­علت اینکه براساس بهینه­سازی برنامه­نویسی چهارگانه است، SVM برنامه­نویسی چهارگانه[[47]](#footnote-47) (QPSVM) نامیده می­شود (وپنیک، 1995). بسیاری از آزمایش­ها نشان می­دهند که LPSVM کارآمد است و در برخی موارد حتی بهتر از QPSVM عمل می­کند. برای مثال، قادر به حل مسائل با نمونه­های بسیار بزرگ، بهبود سرعت محاسباتی و کاهش تعداد بردارهای پشتیبان است (وو و ژو، 2005[[48]](#footnote-48)).

**4-4-4- SVM تُنک**

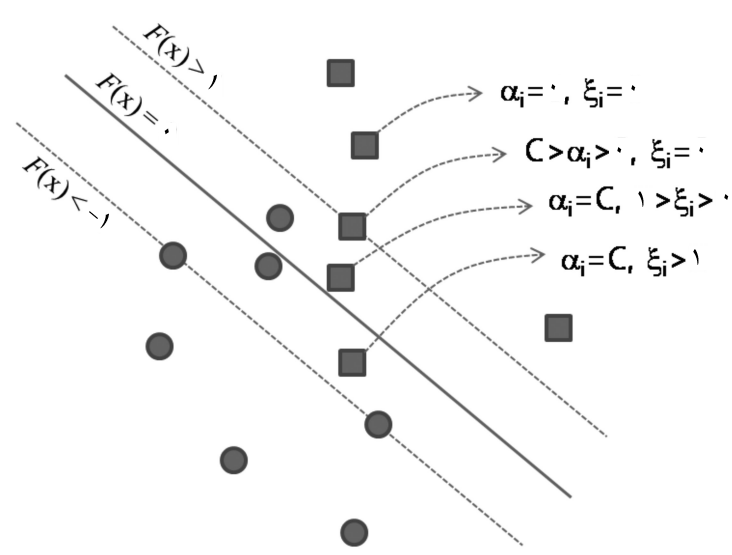
نمایش تُنک (پراکنده) از ماشین­های بردار پشتیبان (SVMها) با توجه به ویژگی­های ورودی برای بسیاری از کاربردها مطلوب است. در بسیاری از کاربردهای یادگیری ماشین، تمایل شدیدی به پراکندگی با در نظر گرفتن ویژگی­های ورودی وجود دارد. چندین عامل موجب این موضوع می­شوند؛ نخست، بسیاری از مجموعة داده­های واقعی مانند داده­های متنی یا داده­های ریزآرایه[[49]](#footnote-49) به­عنوان بردارهایی با ابعاد بسیار بالا نمایش داده می­شوند که منجر به چالش­های بزرگی در پردازش بیشتر می­شود. ثانیاً، معمولاً اکثر ویژگی­ها در بردارهای با ابعاد بالا غیرآگاهی­بخش یا پراختلال هستند و می­توانند به­شدت بر عملکرد تعمیم تأثیر بگذارند. ثالثاً، یک دسته­بندی­کنندة تُنک می­تواند منجر به یک قانون تصمیم­گیری ساده­شده برای پیش­بینی سریع­تر در مسائل بزرگ­مقیاس شود.

**5-4-4- SVM قدرتمند**

SVM قدرتمند، قصد دارد که مسئلة بیش­برازش را با داده­های پرتی که دو کلاس را تفکیک­ناپذیر می­کنند، حل کند. اخیراً، بسیاری از تحقیقات بر روی ردیابی بصری[[50]](#footnote-50) معطوف شده­اند. این موضوع به­دلیل مشکلاتی از قبیل ظاهر پیچیده و تغییر روشنایی، انسداد[[51]](#footnote-51)، پس­زمینة به­هم­ریخته و غیره، یک مسئلة چالشی باقی می­ماند. برای ساخت یک ردیاب قدرتمند، طیف متنوعی از مدل­های نمایش که از روش­های یادگیری مختلفی استفاده می­کنند، در تحقیقات پیشنهاد شده­اند. مطابق با روش­های یادگیری، این مدل­های نمایش می­توانند تقریباً در دو گروه دسته­بندی شوند: حالت نمایش یادگیری براساس مولد و یادگیری براساس افتراق[[52]](#footnote-52). مدل­های نمایش یادگیری براساس مولد[[53]](#footnote-53) (GLMها) عمدتاً بر چگونگی ساخت نمایش جسم قدرتمند در فضاهای ویژگی مشخص تمرکز می­کنند که شامل هیستوگرام انتگرال[[54]](#footnote-54)، تخمین تراکم هسته[[55]](#footnote-55)، یک ترکیب رنگ فضایی[[56]](#footnote-56) از یادگیری زیرفضایی گاوسی[[57]](#footnote-57)، نمایش تُنک (پراکندگی)، تجزیة ردیابی بصری و به همین ترتیب موارد دیگر، می­شود. یک اشکال این روش­ها این است که آن­ها اغلب تأثیر پس­زمینه را نادیده می­گیرند و به­دنبال آن از سردرگمی­های ناشی از نواحی پیش­زمینه با ظاهری مشابه با اشیاء در پیش­زمینه رنج می­برند.

**6-4-4- SVM بیزی**

یک روش زیرفضایی مهم دیگر، الگوریتم بیزی با استفاده از زیرفضای احتمالاتی است. متفاوت از دیگر روش­های زیرفضایی، که تصویر چهرة آزمایشی را به M کلاس M نفری دسته­بندی می­کنند، الگوریتم بیزی مسئلة تشخیص چهره را به­شکل یک مسئلة دسته­بندی الگوی باینری درمی­آورد، که هر کدام از دو کلاس، دگرگونی درون­فردی و برون­فردی، توسط یک توزیع گاوسی مدل­سازی شده­اند. پس از محاسبة ویژگی­های زیرفضایی، اکثر روش­ها برای دسته­بندی تصاویر چهره از فاصلة اقلیدسی سادة ویژگی­های زیرفضایی استفاده می­کنند. اخیراً دسته­بندی­کننده­های هوشمندتری، مانند ماشین­های بردار پشتیبان، نشان داده­اند که می­توانند عملکرد دسته­بندی ویژگی­های زیرفضایی تحلیل مؤلفة اصلی[[58]](#footnote-58) (PCA) و تحلیل تفکیک­کنندة خطی[[59]](#footnote-59) (LDA) را بیشتر ارتقاء دهند. به ازای هر دو کلاس از بردارها، هدف SVMها این است که یک ابرصفحه بیابند که دو کلاس بردار را به­گونه­ای جدا کند که فاصله از ابرصفحه تا نزدیک­ترین بردار هر دو کلاس، بیشینه باشد. این ابرصفحه، به­عنوان ابرصفحة جداکنندة بهینه شناخته می­شود. ماشین­های بردار پشتیبان در مسائل تشخیص دوکلاسه سرآمد هستند و از بسیاری از دسته­بندی­کننده­های خطی و غیرخطی دیگر، عملکرد بهتری دارند. شکل 4-5، روابط گرافیکی میان،و را نمایش می­دهد.



شکل 4-5: روابط گرافیکی میان،و.

**7-4-4- خلاصه**

SVMهای مختلفی در این بخش مورد بحث قرار گرفتند: حداقل مربعات، برنامه­نویسی خطی، تُنک، قدرتمند و بیزی. به ازای هر دو کلاس از بردارها، هدف SVMها این است که یک ابرصفحه بیابند که دو کلاس بردار را به­گونه­ای جدا کند که فاصله از ابرصفحه تا نزدیک­ترین بردار هر دو کلاس بیشینه باشد.

**5-4- روش­های هسته­محور**

**1-5-4- مقدمه**

روش­های دسته­بندی­کنندة الگوی رایج، به­علت موفقیت SVMها در بهبود قابلیت­های تعمیم و دسته­بندی بسط داده شده­اند تا حاشیه­های بیشینه و نگاشت را در یک فضای ویژگی ادغام کنند.

**2-5-4- حداقل مربعات هسته**

روش­های حداقل مربعات در فضای ورودی می­توانند به­آسانی و با استفاده از روش­های هسته به فضای ویژگی توسعه یابند.

**3-5-4- تحلیل مؤلفة اصلی هسته**

تحلیل مؤلفة اصلی (PCA) روشی کارآمد برای یافتن مؤلفه­های ژن بنیادی مهیا کرده و ابعاد ورودی را کاهش می­دهد. این تبدیل خطی به­طور گسترده­ای در روش تحلیل و فشرده­سازی دادة بیان ژن[[60]](#footnote-60) استفاده شده است. اگر داده­ها در زیرفضایی خطی متمرکز شده باشند، PCA راهی برای فشرده کردن داده­ها و ساده­سازی نمایش بدون از دست رفتن اطلاعات بسیاری را فراهم می­کند. با این حال، اگر داده­ها در زیرفضایی غیرخطی متمرکز شده باشند، PCA به­خوبی کار نمی­کند. در این صورت، ممکن است نیاز به در نظر گرفتن تحلیل مؤلفة اصلی هسته[[61]](#footnote-61) (KCPA) باشد. KCPA نسخه­ای غیرخطی از PCA است؛ این روش در چندین سال گذشته در زمینة یادگیری ماشین به­شدت بررسی و در بسیاری از کاربردها ادعای موفقیت کرده است. در دسته­بندی و دیگر وظایف تحلیلی داده، اغلب ضروری است که پردازش داده­ها را پیش از اعمال الگوریتم مورد نظر انجام داد و رایج است که ابتدا ویژگی­های مناسب برای حل کردن وظیفه را استخراج کرد. استخراج ویژگی­ها برای دسته­بندی به­طرز چشمگیری با استخراج ویژگی برای توصیف داده­ها تفاوت می­کند. به­عنوان مثال، PCA جهت­هایی را می­یابد که با توصیف بیشترین واریانس داده­های ممکن با جهت­های m تعامدی[[62]](#footnote-62)، کمترین خطای بازسازی را داشته باشند. با در نظر گرفتن جهت­های اولیه، آن­ها نباید (و در عمل اغلب نخواهند توانست) ساختار کلاسی که برای دسته­بندی مناسب نیاز است را آشکار کنند. تحلیل تفکیک­کننده به این سؤال می­پردازد: در یک مجموعة دادة مشخص با دو کلاس، کدام یک (خطی یا غیرخطی) بهترین ویژگی یا مجموعة ویژگی برای تفکیک این دو کلاس است؟ رویکردهای کلاسیک در ابتدا با دسته­بندی­کنندة بیزی بهینه (از لحاظ تئوری) و با فرض توزیع­های نرمال برای این کلاس­ها، پاسخ به این سؤال را بر عهده می­گیرند. الگوریتم­های استانداردی مانند تحلیل­های تفکیک­کنندة مرتبه دوم یا خطی، که تفکیک­کنندة شناخته­شدة فیشر[[63]](#footnote-63) در میان آن­ها قرار می­گیرد، می­توانند استنتاج شوند. البته هر مدل دیگری غیر از مدل گاوسی را می­توان برای توزیع کلاس­ها فرض کرد؛ اما این کار، اغلب منجر به استفاده نکردن از راه­حل سادة فرم بسته می­شود.

**4-5-4- تحلیل تفکیک­کنندة هسته**

تحلیل تفکیک­کنندة خطی (LDA) یک روش آماری سنتی است که موفقیت خود را در مسائل دسته­بندی اثبات کرده است. این روش براساس یک تحلیل مقدار ویژه است و پاسخی دقیق با اینرسی بیشینه ارائه می­کند؛ اما این روش برای یک مسئلة غیرخطی کارآمد نیست. در کارهای بسیاری، LDA تعمیم­یافته برای مسائل غیرخطی اعلام شده­اند و تلاش می­شود که یک تحلیل تفکیک­کنندة تعمیم­یافته[[64]](#footnote-64) (GDA) به­وسیلة نگاشت فضای ورودی به یک فضای ویژگی با ابعاد بالا و خواص خطی تهیه شود. در فضای جدید، می­توان مسئله را به روشی کلاسیک مانند روش LDA حل کرد. ایدة اصلی این است که فضای ورودی را به یک فضای ویژگی مناسب که در آن متغیرها به­طور غیرخطی با فضای ورودی مرتبط هستند، نگاشت کرد. این نکته در برخی الگوریتم­ها از قبیل الگوریتم­های یادگیری بدون نظارت استفاده شده است. LDA یک ابزار استاندارد برای دسته­بندی بر مبنای تبدیل فضای ورودی به یک فضای جدید است. داده­ها به­عنوان ترکیبی خطی از مقادیر مختصاتی جدید که مؤلفه­های اصلی نامیده می­شوند و محور تفکیک­کننده را نمایش می­دهند، تعریف شده­اند. مشابه LDA، هدف روش GDA، بیشینه کردن اینرسی میان­کلاسی و کمینه کردن اینرسی درون­کلاسی است.

**5-5-4- خلاصه**

در این بخش، به­طور مختصر تحلیل تفکیک­کنندة هسته، حداقل مربعات هسته و PCA هسته بیان شد. PCA هسته، نسخه­ای غیرخطی از PCA است. این مبحث به­شدت در چند سال گذشته در حوزة یادگیری ماشین بررسی شده و در بسیاری از کاربردها موفق ظاهر شده است. تحلیل تفکیک­کننده می­تواند خطی و تعمیم­یافته باشد، که ابزارهای استانداردی برای دسته­بندی هستند. این روش بر مبنای تبدیل فضای ورودی به یک فضای جدید است.

**6-4- انتخاب و استخراج ویژگی**

**1-6-4- مقدمه**

این بخش به بررسی اینکه چگونه ویژگی­های نامرتبط منجر به اختلال، محاسبات سنگین و غیره می­شوند، می­پردازد. همچنین اینکه چگونه ویژگی­های وابسته به یکدیگر، به ویژگی­های مازاد[[65]](#footnote-65) متفاوتی وارد می­شوند و برای مدل بهتر[[66]](#footnote-66) هدف­گذاری شده­اند، بحث خواهند شد.

روش­های انتخاب ویژگی عبارتند از:

* روش فیلتر[[67]](#footnote-67): یک مقیاس آماری را به­کار می­گیرد تا به هر ویژگی یک امتیازدهی اختصاص دهد. به­عنوان مثال، درآزمون مربع کای[[68]](#footnote-68)، به­دست آوردن اطلاعات و ضرایب همبستگی، امتیازدهی می­شوند.
* روش دسته­بند[[69]](#footnote-69): انتخاب مجموعه­ای از ویژگی­ها را به­عنوان یک مسئلة جستجو در نظر می­گیرد.
* روش توکار[[70]](#footnote-70): درحالی­که مدل در حال ساخته شدن است، یاد می­گیرد که چه ویژگی­هایی بیشترین کمک را به دقت مدل می­کنند (برای مثال، رگرسیون لسو[[71]](#footnote-71)، رگرسیون خالص الاستیک[[72]](#footnote-72) و رگرسیون ریج[[73]](#footnote-73)).

کنترل مرزهای کلاس، اغلب دور از دسترس دسته­بندی­کننده­های متداول است، چرا که آن­ها سازوکار­هایی برای کنار آمدن با آن را ندارند. بنابراین، اگر تعداد ویژگی­هایی مانند متغیرهای ورودی نسبت به داده­های آموزشی بیشتر باشند، ممکن است همپوشانی مرزهای کلاس رخ ندهد. در این نوع از سناریوها، قابلیت تعمیم دسته­بندی­کننده­های متداول شاید کارآمد نباشد. برای بهبود قابلیت تعمیم، ایجاد مجموعه­ای کوچک از ویژگی­ها از متغیرهای ورودی اصلی چه با انتخاب ویژگی و چه با استخراج ویژگی توصیه می­شود. از آن­جا که SVMها مرزهای کلاس را مستقیماً از آموزش تعیین می­کنند، اگرچه متغیرهای ورودی بزرگ هستند، قابلیت تعمیم به مقدار زیادی کاهش نمی­یابد. در بخش بعد، روش­های متنوع انتخاب ویژگی با استفاده از SVM تحلیل خواهد شد. چگونگی اثرگذاری انتخاب ویژگی بر قابلیت تعمیم یک SVM و استخراج ویژگی با روش­هایی متنوع نیز بحث خواهند شد.

**2-6-4- مجموعة اولیة ویژگی­ها**

مجموعة ویژگی استفاده شده، اثرگذارترین عامل در درک یک دسته­بندی­کننده با قابلیت تعمیم بالا است. از آن­جا که قانون ثابتی برای تعیین مجموعة اولیه­ای از ویژگی­ها در یک مسئلة دسته­بندی مشخص وجود ندارد، تنها راه تعیین مجموعة اولیة ویژگی­ها، استفاده از سعی و خطا است.

اگر تعداد ویژگی­ها بسیار زیاد باشد و هر ویژگی توان دسته­بندی پایینی داشته باشد، بهتر است که تبدیل خطی یا غیرخطی صورت گیرد تا مجموعه­ای از ویژگی­ها ایجاد شود. برای توان دسته­بندی بالا، کاهش مجموعه می­تواند با انتخاب یا استخراج ویژگی حاصل شود. انتخاب ویژگی، ویژگی­های مازاد یا بی­معنا را کاهش می­دهد تا عملکرد تعمیم بهتر و دسته­بندی سریع­تری تحقق یابد.

**3-6-4- روند انتخاب ویژگی**

هدف انتخاب ویژگی این است که حداقل زیرمجموعة ویژگی­ها را از مجموعة اصلی ویژگی­ها انتخاب کند که قابلیت تعمیم حداکثر را فراهم خواهد کرد. در طی فرایند انتخاب ویژگی، قابلیت تعمیم زیرمجموعه­ای از ویژگی­ها می­بایست تخمین زده شود؛ به این روند، روش دسته­بند گفته می­شود. اما این یک روش زمانبر است و نیاز است که روش­های دیگری استفاده شوند، مانند روش فیلتر که با پیدایش معیارهای انتخاب متنوع در ارتباط است.

روش­های انتخاب روبه­جلو یا روبه­عقبی وجود دارند که به­طور گسترده در انتخاب معیارها استفاده می­شوند. در انتخاب روبه­عقب، ویژگی­ها آزمایش شده و یکی یکی حذف می­شوند، بسته به اینکه کدام یک منجر به کمترین خرابی می­شود. این روند حذف ادامه پیدا می­کند تا زمانی­که معیار انتخاب به مقدار مشخصی برسد.

در انتخاب روبه­جلو، ویژگی­ها یکی یکی به مجموعه­ای خالی افزوده می­شوند، بسته به اینکه کدام یک معیار انتخاب را بیشتر از بقیه بهبود می­بخشد؛ این روند تا هنگامی ادامه پیدا می­کند که معیار انتخاب به مقداری معین برسد. از آن­جا که انتخاب روبه­جلو یا روبه­عقب کم است، افزودن یا حذف بیش از یک ویژگی به­طور همزمان براساس رتبه­بندی ویژگی شاید ممکن باشد. همچنین ترکیب انتخاب روبه­جلو و روبه­عقب ممکن است.

چون این روش­های انتخاب، تکنیک­های بهینه­سازی محلی هستند؛ بهینه­سازی سراسری انتخاب ویژگی تأیید نشده است. با معرفی SVM، روش­های انتخاب گوناگونی که برای SVM مناسب باشند توسعه داده شده­اند؛ در اکثر موارد، SVM خطی استفاده می­شود. اگر برخی المان­های بردار ضریب ابرصفحه صفر باشند، حذف متغیرهای ورودی مرتبط، ابرصفحة بهینه برای متغیرهای باقی­مانده را تغییر نمی­دهد. اگر متغیرهای مرتبط با المان­های غیرصفر حذف شوند، پاسخ بهینه تغییر می­کند؛ بنابراین اندازة حاشیه افزایش می­یابد. علاوه بر روش­های دسته­بند و فیلتر، روش­های توکار آموزش و انتخاب ویژگی را ترکیب می­کنند. در این­جا آموزش SVM منجر به حل یک مسئلة بهینه­سازی مرتبه دوم می­شود و انتخاب ویژگی می­تواند با اصلاح تابع هدف انجام شود. برای برنامه­نویسی خطی، مشابه SVM همراه با هسته­های خطی، متغیرهای مرتبط با ضرایب صفر ابرصفحه­های جداساز، مازاد هستند.

**4-6-4- استخراج ویژگی**

به­منظور استخرج ویژگی، تحلیل مؤلفة اصلی به­طور گسترده­ای استفاده می­شود. PCA هسته در حال کسب پذیرش گسترده­تری است. نشان داده شده که ترکیب KCPA و SVM خطی قابلیت تعمیم بهتری نسبت به SVM غیرخطی تولید می­کند. KCPA ترکیب شده با حداقل مربعات نیز برای استخراج ویژگی استفاده می­شود. از آن­جا که PCA از اطلاعات کلاس استفاده نمی­کند، مؤلفة اصلی اولیه لزوماً برای جدایی کلاس سودمند نیست و همچنین تحلیل تفکیک­کنندة خطی برای یک مسئلة دوکلاسه استفاده شده است؛ اما کاربرد آن محدود به مواردی است که هر کلاس از یک خوشه تشکیل شود و آن­ها همپوشانی شدیدی نداشته باشند. با انتخاب بهینة هسته­ها و مقادیر پارامتری آن­ها، تحلیل تفکیک­کنندة هسته، محدودیت تحلیل تفکیک­کنندة خطی را از بین می­برد. این موضوع برای مسائل چندکلاسه نیز بسط داده شده است. برای انتخاب ویژگی، تحلیل تفکیک­کنندة هسته در کنار انتخاب هسته و استخراج ویژگی به­عنوان معیارها به­کار می­روند.

**5-6-4- خوشه­بندی**

SVM می­تواند برای مسائل یک­کلاسه فرمول­بندی شود؛ که به آن دسته­بندی یک­کلاسه گفته می­شود و در خوشه­بندی و شناسایی داده­های پرت هم در الگوی دسته­بندی و هم در تقریب تابع به­کار می­رود. روش­های متداول خوشه­بندی مانند الگوریتم خوشه­بندی k-میانگین و الگوریتم خوشه­بندی c-میانگین فازی[[74]](#footnote-74) می­توانند به فضای ویژگی بسط داده شوند. توصیف دامنه، ناحیه­ای از داده را به­وسیلة یک ابرصفحه در فضای ویژگی تعریف می­کند. ابرصفحه در فضای ویژگی متناظر با نواحی خوشه­بندی­شده در فضای ورودی است. بنابراین، توصیف دامنه می­تواند برای خوشه­بندی استفاده شود. اگر هیچ دادة پرتی وجود نداشته باشد، که یعنی همة داده­ها در داخل یا روی ابرصفحه هستند؛ آنگاه مسئله خوشه­ها را در فضای ورودی تعیین خواهد کرد.

**6-6-4- خلاصه**

در این بخش، انتخاب و استخراج ویژگی مورد بحث قرار گرفتند. مقداردهی اولیه، انتخاب و خوشه­بندی در استخراج ویژگی تحلیل و بررسی شد. روش­های متداول خوشه­بندی مانند الگوریتم خوشه­بندی k-میانگین و الگوریتم خوشه­بندی c-میانگین فازی می­توانند به فضای ویژگی بسط داده شوند. توصیف دامنه، ناحیه­ای از داده را به­وسیلة یک ابرصفحه در فضای ویژگی تعریف می­کند. برای استخراج ویژگی، تحلیل مؤلفة اصلی به­طور گسترده­ای استفاده می­شود.

**7-4- تقریب تابع**

**1-7-4- مقدمه**

مطابق با یافته­ها، رگرسیون بردار پشتیبان[[75]](#footnote-75) (SVR)، یک بسط از SVM، قابلیت تعمیم بالایی برای تقریب توابع مختلف و مسائل پیش­بینی سری­های زمانی دارد. بسط­های SVM مانند LP-SVR، v-SVR و LS-SVR برای دسته­بندی الگو و تقریب توابع قابل استفاده هستند.

**2-7-4- ابرصفحه­های بهینه**

گسترش روابط ورودی-خروجی با استفاده از جفت­های ورودی-خروجی، هدف تقریب تابع است. در SVR، نگاشت فضای ورودی به یک فضای ویژگی با ابعاد بالا رخ می­دهد. ابرصفحه بایستی در فضای ویژگی مشخص شود.

**3-7-4- رگرسیون بردار پشتیبان نرم­حاشیه[[76]](#footnote-76)**

مشابه این موضوع در SVM خطی را به خاطر بیاورید؛ می­بایست یک ابرصفحه با حاشیة بیشینة به همراه بهینه­سازی زیر مشخص گردد:



در SVM نرم­حاشیه، روش بهینه­سازی مشابهی را در نظر بگیرید، جز در واهلش نامساوی­ها[[77]](#footnote-77)، به­گونه­ای که حالت داده­های تفکیک­ناپذیر خطی را هم برآورده کند.

برای انجام این کار SVM نرم­حاشیه متغیرهای کمکی، مقداری مثبت، را به قید یک مسئلة بهینه­سازی معرفی می­کند.

بنابراین برای نرم­حاشیه، مسئلة بهینه­سازی را مطابق زیر می­توان نوشت:



بنابراین در SVM نرم­حاشیه، می­بایست W و b حساب شوند و پاسخی برای یادگیری SVM است.

**4-7-4- انتخاب مدل**

در SVM، انتخاب مدل به معنی انتخاب مقادیر پارامتری بهینه مانند، و برای هسته­های RFB است. (رگهوندرا[[78]](#footnote-78) و دکا[[79]](#footnote-79)، 2014). بهینه کردن این مقادیر در SVR دشوار است. یکی از قابل­اعتمادترین روش­ها برای انجام این کار، اعتبارسنجی متقابل[[80]](#footnote-80) است. همچنین در مقایسه با SVM، یک مقدار پارامتر بیشتر باید در SVR علاوه بر مقادیر دو پارامتر تعیین شود. روش­های مختلفی برای سرعت­دهی به انتخاب مدل پیشنهاد داده شده­اند. کمینه کردن کران­های خطای یک­طرفه[[81]](#footnote-81) در SVR، یک روش مقایسه با آن­ها در دسته­بندی است. تعیین این پارامترها با جایگزینی آموزش رگرسورهای SV[[82]](#footnote-82) و بهینه­سازی پارامترها با تندترین کاهش در اعتبارسنجی متقابل داده­ها، روشی دیگر است. براساس ارزیابی عملکردهای مختلف از انتخاب مدل، مشخص شده است که اعتبارسنجی متقابل k-fold[[83]](#footnote-83)، یک­طرفه و قید محدوده[[84]](#footnote-84) هر سه با مقادیر انتخاب­شدة مختلف به یک اندازه خوب کار کردند.

**5-7-4- روش­های آموزش**

هنگامی­که یک رگرسور بردار پشتیبان توسط یک مسئلة بهینه­سازی مرتبه دوم بیان می­شود، پاسخ به­طور سراسری بهینه است. استفاده از هسته­های غیرخطی باید مسئلة بهینه­سازی دوگانه­ای را حل کند که در آن تعداد متغیرها دو برابر تعداد داده­های آموزشی است. بنابراین، آموزش برای تعداد زیادی از داده­های آموزشی، دشوار می­شود. برای حل این مسئله، می­توان از تکنیک تجزیه استفاده کرد. تکنیک تجزیه می­تواند در تکه تکه کردن[[85]](#footnote-85) با اندازه ثابت دسته­بندی شود. در این­جا، کاندیدهای بردار پشتیبان ممکن است از اندازة مجموعة کاری تجاوز کنند و ممکن است برخی المان­های غیرصفر از مجموعة کاری خارج شوند. مسائل اندازه­بزرگ می­توانند آموزش داده شوند، اما برای همگرایی آن­ها تکرارهای زیادی مورد نیاز است.

در تکه تکه کردن با اندازة متغیر، به­علت المان­های صفر حاضر در مجموعه، معمولاً تعداد تکرارها در مقایسه با تکه تکه کردن با اندازة ثابت کمتر است. مجموعة کاری شامل متغیرهای مرتبط با قیدهای فعال که در آن­ها تساوی­ها برآورده گشته­اند، می­شود. روش­های آموزشی که از تکه تکه کردن با اندازة متغیر توسعه داده شده برای دسته­بندی الگو استفاده می­کنند، می­توانند برای تقریب تابع هم بسط داده شوند.

تکنیک فیلتر کالمان[[86]](#footnote-86) معمولاً برای آموزش ترتیبی رگرسورهای بردار پشتیبان استفاده می­شود. همچنین ورودی­های آموزشی می­توانند روی یک شبکه قرار بگیرند که در آن برای آموزش سریع­تر، یک ماتریس مرتبط با هسته توسط یک ماتریس بلوکی بیان شده است.

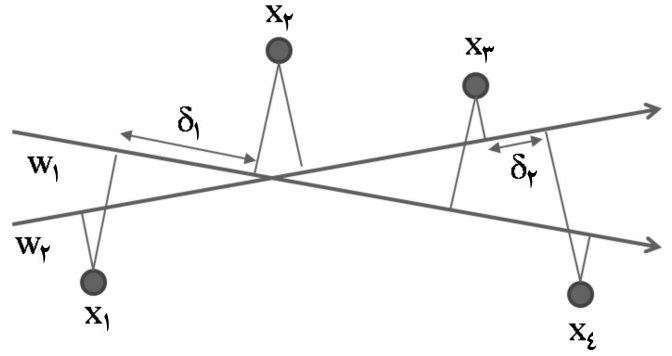
**6-7-4- انواع SVR**

انواع مختلف SVR، برنامه­نویسی خطی، v-SVR، حداقل مربعات و تُنک هستند. معمولاً برای بهبود قابلیت تعمیم از طریق هسته­های چندگانه و شناسایی داده­های پرت، بهبود یا افزودن توابع در نظر گرفته می­شود. برای ارتقاء توانایی تقریب، مخلوطی از هسته­های مختلف استفاده می­شود. برای پیشرفت در توانایی­های درون­یابی و برون­یابی، مخلوطی از هسته­های سراسری مانند چندجمله­ای­ها و هسته­های محلی پیشنهاد شده­اند. SVR چندپاسخی نیز براساس تحلیل سیگنال چندپاسخی برای بهبود عملکرد پیشنهاد شده است. به­جای هسته­های منفرد، هسته­های چندگانه با پاسخ­های متفاوت با یکدیگر ترکیب شده­اند. در توابع متعامد اساسی، اجزای پاسخ­های گوناگون می­توانند به­طور جداگانه محاسبه شوند. با این حال، هسته­ها در SVR متعامد نیستند، پس آموزش بردار پشتیبان مجدداً فرمول­بندی می­شود تا ضرایب لاگرانژی برای دو نوع هسته را به­طور همزمان تعیین کند.

**7-7-4- انتخاب­های متغیر**

انتخاب متغیر یک فرایند مؤثر برای کاهش پیچیدگی محاسباتی و بهبود قابلیت تعمیم رگرسور است. هدف انتخاب متغیر، دستیابی به کوچکترین مجموعه از متغیرها است که بیشترین قابلیت تعمیم را ارائه می­دهند. روش دسته­بند زمانبر است؛ اگرچه روش فیلتر بار محاسباتی را کمتر می­کند، اما با دیگر مسائل از قبیل ریسک افت ناگهانی در قابلیت تعمیم در انتخاب زیرمجموعه­ای از متغیرهای ورودی مرتبط است. برای حداقل کردن این مشکلات، ترکیبی از هر دو روش برای انتخاب متغیرها در طی آموزش پیشنهاد شده که روش­های توکار نامیده می­شود.

یکی از مهم­ترین مسائل در انتخاب متغیر، معیار انتخاب-توقف متغیرها است. تعیین مجموعه­هایی از متغیرها با قابلیت تعمیم نظیر یا برتر از زیرمجموعة اولیة متغیرهای مربوط به معیار انتخاب است. هدف اصلی ایجاد کوچکترین مجموعه از متغیرها با قابلیت تعمیم در مقایسه با مجموعة اولیة متغیرها است. بنابراین، پیش از انتخاب متغیر باید یک مقدار آستانه برای معیارهای انتخاب قرار داده شود که با استفاده از تمام متغیرها خطای تقریب را ارزیابی کند. شکل 4-6، نمایشی خطی از چهار نقطة داده را نشان می­دهد.



شکل 4-6: نمایش خطی از چهار نقطة داده.

**8-7-4- خلاصه**

در این بخش، ابرصفحه­های بهینه، رگرسیون بردار پشتیبان نرم­حاشیه، انتخاب مدل، روش­های آموزش، انواع SVR و انتخاب متغیر به­طور خلاصه بحث شدند. انتخاب ابرصفحة مرتبط با نگاشت­های ورودی-خروجی شرح داده شدند. حاشیه برای SVR، خصوصیات مجموعة داده­ها و اندازه­ها در انتخاب مدل مورد بحث قرار گرفتند. روش­های آموزش گوناگون و انواع SVR با انتخاب متغیرها توضیح داده شدند. نهایتاً، دربارة قابلیت اعمال روش­های فیلتر و دسته­بند بحث گردید.

**منابع**

Ari, S., Hembram, K., and Saha, G. 2010. Detection of cardiac abnormality from PCG signal using LMS based least square SVM classifier, *Expert Systems with* *Applications*, 37(12), 8019–8026.

Brooks, J.P. 2011. Support vector machines with the ramp loss and the hard margin loss, *Operations Research*, 59(2), 467–479.

Brunner, C., Fischer, A., Luig, K., and Thies, T. 2012. Pairwise support vector machines and their application to large scale problems, *Journal of Machine* *Learning Research*, 13(Aug), 2279–2292.

Cortes, C. and Vapnik, V. 1995. Support-vector networks, Machine Learning, 20(3), 273–297.

Raghavendra, N.S., Deka, P.C. 2014. Support vector machine applications in the field of hydrology: a review, Applied Soft Computing 19, 372–386. doi:10.1016/j. asoc.2014.02.002.

Scholkopf, B. 2001. The kernel trick for distances, In Advances in Neural Information Processing Systems, Eds. T.K. Leen, T.G. Dietterich and V. Tresp, MIT Press, 14th Annual Neural Information Processing Systems Conference, Denver, CO pp. 301–307.

Vapnik, V.N. 1995. The Nature of Statistical Learning Theory, Springer Verlag, New York.

Wu, Q., and Zhou, D.X. 2005. SVM soft margin classifiers: linear programming versus quadratic programming, Neural Computation, 17(5), 1160–1187.

1. astrophysics [↑](#footnote-ref-1)
2. formal definition [↑](#footnote-ref-2)
3. overfitting [↑](#footnote-ref-3)
4. Support vector machine [↑](#footnote-ref-4)
5. Hard Margin SVM [↑](#footnote-ref-5)
6. Soft Margin SVM [↑](#footnote-ref-6)
7. Vapnik [↑](#footnote-ref-7)
8. hyperplane [↑](#footnote-ref-8)
9. heuristics [↑](#footnote-ref-9)
10. Discrete SVM [↑](#footnote-ref-10)
11. Brooks [↑](#footnote-ref-11)
12. hard-maximal margin classifier [↑](#footnote-ref-12)
13. kernel tricks [↑](#footnote-ref-13)
14. Schölkopf [↑](#footnote-ref-14)
15. Hilbert [↑](#footnote-ref-15)
16. Mercer [↑](#footnote-ref-16)
17. Hessian [↑](#footnote-ref-17)
18. Radial Basis Function [↑](#footnote-ref-18)
19. Multi-Layered Perceptron [↑](#footnote-ref-19)
20. Fourier [↑](#footnote-ref-20)
21. splines [↑](#footnote-ref-21)
22. B-splines [↑](#footnote-ref-22)
23. additive kernels [↑](#footnote-ref-23)
24. tensor products [↑](#footnote-ref-24)
25. normalizing kernels [↑](#footnote-ref-25)
26. فرایند سفیدسازی (Whitening)، فرایند تقلیل داده­ها به بردار نویز سفید است. [↑](#footnote-ref-26)
27. Euclidian distances [↑](#footnote-ref-27)
28. preserving neighborhood relations [↑](#footnote-ref-28)
29. implicit bias terms [↑](#footnote-ref-29)
30. One against all SVM [↑](#footnote-ref-30)
31. Pairwise SVM [↑](#footnote-ref-31)
32. Error correcting output code SVM [↑](#footnote-ref-32)
33. All at once SVM [↑](#footnote-ref-33)
34. conventional SVM [↑](#footnote-ref-34)
35. linear multiple discriminant [↑](#footnote-ref-35)
36. decision tree-based SVM [↑](#footnote-ref-36)
37. Brunner et al., 2012 [↑](#footnote-ref-37)
38. Sparse [↑](#footnote-ref-38)
39. robust [↑](#footnote-ref-39)
40. Least Square SVM [↑](#footnote-ref-40)
41. wavelet-based technique [↑](#footnote-ref-41)
42. Ari et al., 2010 [↑](#footnote-ref-42)
43. Vapnik-Chervonenkis dimension [↑](#footnote-ref-43)
44. 1- norm soft margin SVM [↑](#footnote-ref-44)
45. Cortes [↑](#footnote-ref-45)
46. Linear Programming SVM [↑](#footnote-ref-46)
47. Quadratic Programming SVM [↑](#footnote-ref-47)
48. Wu and Zhou, 2005 [↑](#footnote-ref-48)
49. microarray data [↑](#footnote-ref-49)
50. visual tracking [↑](#footnote-ref-50)
51. occlusion [↑](#footnote-ref-51)
52. discriminative learning-based [↑](#footnote-ref-52)
53. generative learning-based [↑](#footnote-ref-53)
54. integral histogram [↑](#footnote-ref-54)
55. kernel density estimation [↑](#footnote-ref-55)
56. spatial-color mixture [↑](#footnote-ref-56)
57. Gaussian subspace learning [↑](#footnote-ref-57)
58. Principal Component Analysis [↑](#footnote-ref-58)
59. Linear Discriminant Analysis [↑](#footnote-ref-59)
60. gene expression data [↑](#footnote-ref-60)
61. Kernel principal component analysis [↑](#footnote-ref-61)
62. m orthogonal directions [↑](#footnote-ref-62)
63. Fisher discriminant [↑](#footnote-ref-63)
64. Generalized Discriminant Analysis [↑](#footnote-ref-64)
65. redundant features [↑](#footnote-ref-65)
66. Better Model [↑](#footnote-ref-66)
67. Filter method [↑](#footnote-ref-67)
68. Chi-squared [↑](#footnote-ref-68)
69. Wrapper method [↑](#footnote-ref-69)
70. Embedded method [↑](#footnote-ref-70)
71. LASSO Regression [↑](#footnote-ref-71)
72. Elastic Net regression [↑](#footnote-ref-72)
73. Ridge Regression [↑](#footnote-ref-73)
74. fuzzy-c means [↑](#footnote-ref-74)
75. Support Vector Regression [↑](#footnote-ref-75)
76. soft margin Support Vector Regression [↑](#footnote-ref-76)
77. relaxation of inequalities [↑](#footnote-ref-77)
78. Raghavendra [↑](#footnote-ref-78)
79. Deka [↑](#footnote-ref-79)
80. cross-validation [↑](#footnote-ref-80)
81. leave-one-out error [↑](#footnote-ref-81)
82. SV regressors [↑](#footnote-ref-82)
83. k-fold cross-validation [↑](#footnote-ref-83)
84. span bound [↑](#footnote-ref-84)
85. chunking [↑](#footnote-ref-85)
86. Kalman filtering technique [↑](#footnote-ref-86)