



دانشگاه آزاد اسلامی

واحد تهران شمال

دانشکده مهندسی برق و کامپیوتر گروه مهندسی کامپیوتر

پایان نامه برای دریافت درجه کارشناسی ارشد «M.Sc»

رشته مهندسی کامپیوتر، گرایش هوش مصنوعی

عنوان

مقاوم سازی دسته بند ماشین بردار پشتیبان دوقلو در برابر داده های نویزی

استاد راهنما

جلال الدین نصیری

استادان مشاور

سمیه فتاحی

نگارش

سید امیر محمود میر

زمستان ۱۳۹۷

الله اکرم

۱۴۳۱۹۴
۹۷/۱۱/۲۳

به نام خدا



صورتجلسه دفاع از پایان نامه کارشناسی ارشد

با تأیید خداوند متعال جلسه دفاع از پایان نامه کارشناسی ارشد آقای سید امیرمحمود میر ورودی ۹۴ با شماره دانشجویی ۹۴۰۰۱۸۱۱۷ در رشته مهندسی کامپیوتر گرایش موش مصنوعی تحت عنوان مقام‌سازی دسته‌بند ماشین بردار پشتیبان دوقلو در برابر داده‌های نویزی با حضور استادان راهنما، مشاور (مشاوران)، داور (داوران) و مدیر یا نماینده گروه در دانشکده برق و کامپیوتر در تاریخ ۹۷/۱۱/۰۳ تشکیل گردید.

نظرهای داوران پس از استماع بیانات و نحوه ارایه نامبرده به شرح ذیل می‌باشد:

<input checked="" type="checkbox"/> قبول	<input type="checkbox"/> دفاع مجدد	<input type="checkbox"/> مردود	defense of the final thesis
<input checked="" type="checkbox"/> عالی (۱۸/۱-۲۰)	<input type="checkbox"/> بسیار خوب (۱۶/۱-۱۸)	<input type="checkbox"/> خوب (۱۴/۱-۱۶)	Grade of the final thesis

نمره پایان نامه بد عدد: ۲۰ / ۲ - نمره پایان نامه به حروف: هشت هزار

همایه داوران	نام و نام خانوادگی	مرتبه علمی	امضاء
استاد راهنما	دکتر جلال الدین نصیری	اساتذه ایار	
استاد مشاور ۱	دکتر سمیه فتاحی	استاد رایر مرید	
استاد مشاور ۲			
استاد داور ۱	دکتر سمانه یزدانی	استاد داور	
استاد داور ۲			

تأیید مدیر گروه یا مدیر تحصیلات تکمیلی واحد:	ججه داری	امضا	تاریخ
تأیید معاون پژوهشی واحد:		امضا	تاریخ

احسال: معاونت آموزشی

روزنگار: ۱- اداره آموزش دانشکده

۲- اداره پژوهش دانشکده

۳- خدمات کامپیوتری دانشکده

۴- مدیر گروه کارشناسی ارشد

تعهدنامه اصالت پایان نامه

اینجانب سید امیر محمود میر دانش آموخته مقطع کارشناسی ارشد ناپیوسته در رشته مهندسی کامپیوتر گرایش هوش مصنوعی که در تاریخ ۹۷/۱۱/۰۳ از پایان نامه خود تحت عنوان "مقاآم سازی دسته بند ماشین بردار پشتیبان دو قلو در برابر داده های نویزی" با کسب نمره ۲۰ و درجه عالی دفاع نموده ام بدین وسیله متعهد می شوم:

۱. این پایان نامه / رساله حاصل تحقیق و پژوهش انجام شده توسط اینجانب بوده و در مواردی که از دستاوردهای علمی و پژوهشی دیگران (اعم از پایان نامه، کتاب، مقاله و....) استفاده نموده ام، مطابق ضوابط و رویه موجود، نام منبع مورد استفاده و سایر مشخصات آن را در فهرست مربوطه ذکر و درج کرده ام.
۲. این پایان نامه / رساله قبل از دریافت هیچ مدرک تحصیلی (هم سطح، پایین تر یا بالاتر) در سایر دانشگاهها و موسسات آموزشی عالی ارائه نشده است.
۳. چنانچه بعد از فراغت تحصیل، قصد استفاده و هرگونه بهره برداری اعم از چاپ کتاب، ثبت اختراع و... از این پایان نامه داشته باشم، از حوزه معاونت پژوهشی واحد مجوزهای مربوطه را اخذ نمایم.
۴. چنانچه در هر مقطعي زمانی خلاف موارد فوق ثابت شود، عواقب ناشی از آن را می پذيرم و واحد دانشگاهی مجاز است با اينجانب مطابق ضوابط و مقررات رفتار نموده و در صورت ابطال مدرک تحصيلی ام هيچگونه ادعائي نخواهم داشت.

نام و نام خانوادگی: سید امیر محمود میر



معاونت پژوهش و فناوری

به نام خدا

شور اخلاق پژوهش

بیاری از خداوند بجان و اعتماد به این که عالم محض خدا است و هماره ناظر بر اعمال انسان و به مطهور پاس داشت مقام بلند دانش پژوهش و نظریه ایست
جایگاه دانشگاه در اعلای فرهنگ و تمدن بشری، ماد انجیان و اعضا هایی دانشگاه آزاد اسلامی متهم می کردیم اصول زیر را در انجام
فایت های پژوهشی مد نظر قرار داده و از آن تحقیقی گنیم:

- ۱- اصل برآنت: اثرا نام ببرآنت جویی از هرگونه رفتار غیر حرفه ای و اعلام موضع نسبت به کسانی که حوزه علم پژوهش را باشند با غیر علمی می آینند.
- ۲- اصل رعایت انصاف و امانت: تهدید به اختیاب از هرگونه جانب داری غیر علمی و حافظت از اموال، تجزیرات و منابع در اختیار.
- ۳- اصل ترویج: تهدید به رواج دانش و اساس نتایج تحقیقات و انتقال آن به هکاران علمی و دانشجویان به غیر از مواردی که منع قانونی دارد.
- ۴- اصل احترام: تهدید به رعایت حریم ناد مرمت ناد انجام تحقیقات و رعایت جانب تقدیم خود را از هرگونه حرمت نکنی.
- ۵- اصل رعایت حقوق: اثرا نام برعایت کامل حقوق پژوهشگران و پژوهیدگان (انسان، حیوان و نبات) و سایر صاحبان حق.
- ۶- اصل رازداری: تهدید به میانت از اسرار و اطلاعات محیمان افراد، سازمان ها و کشور و کلیه افراد و نهادهای مرتبط با تحقیق.
- ۷- اصل تحقیقت جویی: تلاش در راستای پی جویی تحقیقت و وفاداری به آن و دوری از هرگونه پنهان سازی تحقیقت.
- ۸- اصل مالکیت مادی و معنوی: تهدید به رعایت کامل حقوق مادی و معنوی دانشگاه و کلیه هکاران پژوهش.
- ۹- اصل منافع ملی: تهدید به رعایت مصالح ملی و در نظر داشتن پیشبرد و توسعه کشور دلکیه مراحل پژوهش.

تقدیم به:

خدایی که آفرید

جهان را، انسان را، عقل را، علم را، معرفت را، عشق را

سپاسگزاری

در آغاز سپاسگزار پدر و مادرم هستم که همواره در زندگی و دوران تحصیلم حامی معنوی و مالی من بوده‌اند. وظیفه خود می‌دانم که از استاد راهنمای خود، جناب آقای دکتر جلال‌الدین نصیری، تشکر و قدردانی کنم. ایشان با راهنمایی، حمایت و توصیه‌های ارزنده‌شان، بنده را در انجام این پژوهش بسیار یاری کردند. همچنین از سرکار خانم دکتر فتاحی جهت مشاوره و راهنمایی در زمینه‌ی داده‌کاوی و روش تحقیق سپاسگزارم. در آخر لازم به ذکر است که پیاده‌سازی و ارزیابی الگوریتم‌ها در آزمایشگاه متن‌کاوی و یادگیری ماشین پژوهشگاه ایراندак صورت گرفته است. لذا از ریاست محترم وقت پژوهشگاه ایراندак، جناب آقای دکتر علیدوستی و کارکنان محترم این مجموعه برای فراهم کردن امکانات مورد نیاز پژوهش تشکر می‌نمایم.

فهرست مطالب

عنوان	صفحه
چکیده	۱
۱ کلیات	۲
۱-۱ مقدمه	۲
۱-۲ تعریف مسئله	۴
۱-۳ نوآوری‌های پژوهش	۵
۱-۴ ساختار کاری پایان‌نامه	۷
۲ پژوهشیه پژوهش	۸
۲-۱ ماشین بردار پشتیبان	۸
۲-۱-۱ ماشین بردار پشتیبان با حاشیه سخت	۸
۲-۱-۲ ماشین بردار پشتیبان با حاشیه نرم	۱۰
۲-۱-۳ ماشین بردار پشتیبان با هسته غیر خطی	۱۲
۲-۱-۴ ماشین بردار پشتیبان چند کلاسه	۱۳
۲-۱-۴-۱ یک-در مقابل-بقیه	۱۳
۲-۱-۴-۲ یک-در مقابل-یک	۱۴
۲-۱-۴-۳ گراف جهت دار یک-در مقابل-یک	۱۵
۲-۱-۴-۵ پژوهشیه پژوهش در ماشین بردار پشتیبان	۱۶
۲-۲ ماشین بردار پشتیبان دو قلو	۱۸
۲-۲-۱ ماشین بردار پشتیبان دو قلو خطی	۱۸

الف

۲۱	۲-۲-۲ ماشین بردار پشتیبان دو قلو غیر خطی
۲۳	۳-۲-۲ پیشینه پژوهش در ماشین بردار پشتیبان دو قلو
۲۳	۱-۳-۲-۲ ماشین بردار پشتیبان دو قلو کمترین مربعات
۲۴	۲-۳-۲-۲ ماشین بردار پشتیبان دو قلو مبتنی بر مرز
۲۵	۳-۳-۲-۲ ماشین بردار پشتیبان دو قلو وزن دار با اطلاعات محلی
۲۸	۴-۳-۲-۲ سایر گسترش‌های ماشین بردار پشتیبان دو قلو
۳۰	۳ ماشین بردار پشتیبان دو قلو کمترین مربعات مبتنی بر نزدیک‌ترین همسایه
۳۰	۱-۳ مقدمه
۳۱	۲-۳ ساخت ماتریس وزن‌ها
۳۳	۳-۳ نسخه خطی
۳۶	۴-۳ نسخه غیر خطی
۳۸	۵-۳ تحلیل روش KNN-LSTSVM
۳۹	۶-۳ جمع‌بندی
۴۰	۴ ماشین بردار پشتیبان دو قلو مبتنی بر رگولارسیون و نزدیک‌ترین همسایه
۴۰	۱-۴ مقدمه
۴۲	۲-۴ الگوریتم نزدیک‌ترین همسایه مبتنی بر تفاوت مکانی فاصله‌ها
۴۴	۳-۴ تعریف ماتریس وزن‌ها
۴۶	۴-۴ نسخه خطی
۵۱	۵-۴ نسخه غیر خطی
۵۳	۶-۴ تحلیل روش RKNN-TSVM
۵۴	۱-۶-۴ پیچیدگی محاسباتی روش RKNN-TSVM
۵۵	۲-۶-۴ مقایسه با سایر دسته‌بندهای مشابه
۵۵	۱-۲-۶-۴ مقایسه با روش TSVM
۵۵	۲-۲-۶-۴ مقایسه با روش WLTSVM
۵۶	۳-۲-۶-۴ مقایسه با روش KNN-STSV

۵۶ ۳-۶-۴ محدودیت‌های روش RKNN-TSVM
۵۶ ۴-۶-۴ مقیاس‌پذیری روش RKNN-TSVM
۵۷ ۷-۴ الگوریتم بهینه‌سازی clipDCD
۵۹ ۸-۴ جمع‌بندی
۶۰ ۵ نتایج و ارزیابی
۶۰ ۱-۵ مقدمه
۶۰ ۲-۵ ارزیابی روش KNN-LSTSVM
۶۱ ۱-۲-۵ نحوه پیاده‌سازی و اجرای الگوریتم‌ها
۶۱ ۲-۲-۵ مجموعه داده مصنوعی
۶۲ ۳-۲-۵ نتایج ارزیابی بر روی مجموعه داده UCI
۶۵ ۱-۳-۲-۵ بررسی آماری
۶۶ ۴-۲-۵ مجموعه داده NDC
۶۹ ۳-۵ ارزیابی روش RKNN-TSVM
۶۹ ۱-۳-۵ نحوه پیاده‌سازی و اجرای الگوریتم‌ها
۶۹ ۲-۳-۵ نحوه انتخاب پارامترها
۷۰ ۳-۳-۵ نتایج ارزیابی و بحث
۷۰ ۱-۳-۳-۵ مجموعه داده مصنوعی
۷۱ ۲-۳-۳-۵ مجموعه داده UCI
۷۴ ۳-۳-۳-۵ بررسی آماری
۷۵ ۴-۳-۳-۵ بررسی حساسیت روش RKNN-TSVM به پارامترها
۷۷ ۵-۳-۳-۵ آزمایش با مجموعه داده NDC
۷۹ ۴-۵ جمع‌بندی
۸۰ ۶ نتیجه‌گیری و پژوهش‌های آینده
۸۰ ۱-۶ مقدمه
۸۰ ۲-۶ مروری بر دسته‌بندهای پیشنهادی

۸۱	۳-۶ مروری بر یافته‌های این پژوهش
۸۳	۴-۶ پیشنهادها
۸۴	مقالات‌های مستخرج از پایان‌نامه
۸۵	مراجع
۸۸	واژه نامه انگلیسی به فارسی
۹۰	واژه نامه فارسی به انگلیسی
۹۲	فهرست اختصارات
۹۴	چکیده انگلیسی

فهرست جداول

عنوان	صفحه
۱-۲ مرور کلی گسترش‌های مبتنی بر روش TSVM	۲۹
۱-۵ مشخصات مجموعه داده‌ها برای ارزیابی روش KNN-LSTSVM	۶۳
۲-۵ مقایسه دقت و زمان آموزش دسته‌بندهای LSTSVM، WLTSVM، TSVM و KNN-LSTSVM	۶۴
۳-۵ میانگین رتبه براساس دقت (ارزیابی KNN-LSTSVM)	۶۶
۴-۵ مشخصات مجموعه داده NDC	۶۷
۵-۵ مقایسه زمان آموزش روش KNN-LSTSVM و سایر روش‌ها بر روی مجموعه داده NDC	۶۸
۶-۵ کتابخانه و نرم افزارهای استفاده شده برای ارزیابی روش RKNN-TSVM	۷۰
۷-۵ مشخصات مجموعه داده‌ها برای ارزیابی روش RKNN-TSVM	۷۲
۸-۵ مقایسه روش‌های RKNN-TSVM، TSVM، TBSVM و WLTSVM بر روی مجموعه داده‌های UCI با تابع هسته RBF	۷۳
۹-۵ میانگین رتبه براساس دقت (ارزیابی RKNN-TSVM)	۷۵
۱۰-۵ مقایسه زمان آموزش روش RKNN-TSVM با سایر روش‌روی مجموعه داده NDC با تابع هسته خطی	۷۸
۱۱-۵ مقایسه زمان آموزش روش RKNN-TSVM با سایر روش‌روی مجموعه داده NDC با تابع هسته RBF	۷۸

فهرست شکل‌ها

عنوان	صفحه
۱-۲ مسئله حاشیه سخت در ماشین بردار پشتیبان	۹
۲-۲ مسئله حاشیه نرم در ماشین بردار پشتیبان	۱۱
۳-۲ نگاشت نمونه‌ها با تکنیک حقه‌ی هسته	۱۳
۴-۲ نواحی غیر قابل دسته‌بندی در روش یک-در مقابل-بقیه	۱۴
۵-۲ نواحی غیر قابل دسته‌بندی در روش یک-در مقابل-یک	۱۵
۶-۲ گراف دودویی DAG برای سه کلاس	۱۶
۷-۲ تفسیر هندسی روش ماشین بردار پشتیبان دو قلو خطی	۱۸
۸-۲ مقایسه هندسی روش WLTSVM با روش TSVM	۲۶
۱-۳ تفسیر هندسی روش KNN-LSTSVM و LS-TSVM	۳۱
۴-۴ ایده اصلی روش LDMDBA	۴۲
۴-۴ ایده اصلی روش RKNN-TSVM	۴۶
۵-۱ عملکرد و ناحیه تصمیم روش KNN-LSTSVM و LS-TSVM را برای روی داده Ripley	۶۲
۵-۲ عملکرد و ناحیه تصمیم روش KNN-LSTSVM و LS-TSVM را برای روی داده Checkerboard	۶۲
۵-۳ اثر افزایش پارامتر k روی دقت دسته‌بند KNN-LSTSVM	۶۵
۵-۴ تاثیر پارامتر k روی زمان آموزش روش KNN-LSTSVM	۶۹
۵-۵ ناحیه تصمیم روش WLTSVM و RKNN-TSVM روی مجموعه داده Ripley با تابع هسته خطی	۷۱

- ۶-۵ ناحیه تصمیم روشن RKNN-TSVM و WLTSVM روی مجموعه داده با Checkerboard تابع هسته RBF
- ۷۱
- ۷-۵ تاثیر پارامتر k روی زمان آموزش روشن RKNN-TSVM روی مجموعه داده Pima-Indian
- ۷۶
- ۸-۵ عملکرد نسخه خطی روشن RKNN-TSVM روی پارامترهای مختلف c_1 و c_2
- ۷۶
- ۹-۵ عملکرد نسخه خطی روشن RKNN-TSVM روی پارامترهای مختلف c_1 و k

فهرست الگوریتم‌ها

عنوان	صفحه
۱-۳ ایجاد مدل خطی دسته‌بند KNN-LSTSVM	۳۵
۲-۳ ایجاد مدل غیر خطی دسته‌بند KNN-LSTSVM	۳۸
۱-۴ الگوریتم نزدیکترین همسایه مبتنی بر تفاوت مکانی فاصله‌ها (LDMDBA)	۴۴
۲-۴ ایجاد مدل خطی دسته‌بند RKNN-TSVM	۵۰
۳-۴ ایجاد مدل غیر خطی دسته‌بند RKNN-TSVM	۵۴
۴-۴ الگوریتم بهینه‌سازی clipDCD	۵۸

فهرست علائم و نمادها

T	مجموعه داده آموزشی
$\mathcal{O}(\cdot)$	نماد مجانبی
m	تعداد نمونه‌های آموزشی
n	تعداد ویژگی‌ها
$x_i \in \mathbb{R}^n$	نمونه نام
y_i	برچسب نمونه نام
$w \in \mathbb{R}^n$	بردار وزن
e	بردار همانی
$w^T x = \sum_{i=1}^n w_i x_i$	ضرب داخلی دو بردار
A^T	ترانهاده ماتریس
A^{-1}	معکوس ماتریس
I	ماتریس همانی
$[A \ B]$	الحق عمودی دو ماتریس
$\ w\ = \sqrt{\sum_{i=1}^n w_i^2}$	نرم اقلیدسی
α, β	ضرایب لاغرانژ
η, y, ξ	بردارهای لغزش
$\frac{\partial f}{\partial w}$	مشتق جزئی تابع f نسبت به w

چکیده

در دهه اخیر، یادگیری ماشین برای حل کردن مسائل با الگوهای پیچیده استفاده شده است. دسته‌بندی یکی از روش‌های اصلی یادگیری است که مسائلی نظری تشخیص چهره، تشخیص متون و تشخیص بیماری‌ها را حل می‌کند. ماشین بردار پشتیبان (SVM) یکی از روش‌های شناخته شده دسته‌بندی است که دقیق و تعمیم‌پذیری خوبی دارد. دسته‌بندهای مختلفی بر پایه SVM در سال‌های اخیر ارائه شده است. در میان آن‌ها ماشین بردار پشتیبان دو قلو (TSVM) بیشتر مورد توجه بوده است.

ایده اصلی روش TSVM پیدا کردن دو ابرصفحه غیرموازی برای دسته‌بندی داده‌ها است. بطوریکه دو مسئله بهینه‌سازی با اندازه کوچک‌تر از مسئله SVM حل می‌شود. از این رو دسته‌بند TSVM در تئوری 4 برابر سریع‌تر از SVM است. با وجود اینکه دسته‌بند TSVM دقیق و مرتبه زمانی بهتری نسبت به SVM دارد، نقاط ضعفی مانند حساسیت به نمونه‌های پرت و نویزی، برازش بیش از حد (Overfitting) و پیچیدگی محاسباتی بالا برای مجموعه داده‌های بزرگ را دارد. در این پژوهش با هدف برطرف کردن نقطه ضعف بیان شده در TSVM، دو دسته‌بند ماشین بردار پشتیبان دو قلو کمترین مربعات مبتنی بر نزدیک‌ترین همسایه (KNN-LSTSVM) و ماشین بردار پشتیبان دو قلو مبتنی بر رگولارسیون و نزدیک‌ترین همسایه (RKNN-TSVM) ارائه می‌شود.

دسته‌بندهای پیشنهادی KNN-LSTSVM و RKNN-TSVM با ساخت گراف نزدیک‌ترین همسایه به نمونه‌های آموزشی وزن می‌دهند و همچنین نمونه‌های حاشیه‌ای هر کلاس را مشخص می‌کنند. این مشخصه دسته‌بندهای پیشنهادی را نسبت به نمونه‌های نویزی و پرت مقاوم‌تر از روش TSVM می‌کند. روش‌های پیشنهادی بر روی مجموعه داده‌های مصنوعی و واقعی به طور جامع ارزیابی شده است. نتایج ارزیابی نشان می‌دهد که دسته‌بندهای پیشنهادی KNN-LSTSVM و RKNN-TSVM نسبت به سایر دسته‌بندهای مشابه از نظر دقیق و سرعت یادگیری بهتر عمل کرده‌اند.

واژه‌های کلیدی: ماشین بردار پشتیبان دو قلو، گراف نزدیک‌ترین همسایه، کمترین مربعات، ریسک ساختاری، دسته‌بندی

فصل ۱

کلیات

۱-۱ مقدمه

یادگیری ماشین به عنوان یکی از شاخه‌های پرکاربرد هوش مصنوعی در چند دهه اخیر بسیار مورد توجه دانشگاه‌ها و صنعت بوده است [۱]. یادگیری ماشین به دنبال شناسایی خودکار الگوهای معنادار از داده‌ها است. در سال‌های اخیر، فناوری‌های مبتنی بر یادگیری ماشین گسترش و توسعه یافته‌اند. به طور مثال، نتایج موتورهای جستجو با روش‌های یادگیری ماشین بهبود یافته است. سامانه‌های تشخیص تقلب در تراکنش‌های بانکی و تشخیص چهره در دوربین‌های دیجیتال از دیگر کاربردهای یادگیری ماشین هستند.

الگوها در مسائل اشاره شده مانند تشخیص چهره بسیار پیچیده هستند. بطوریکه یک برنامه نویس نمی‌تواند با دستورالعمل‌های صریح الگوهای پیچیده را تشخیص دهد. با این حال روش‌های یادگیری ماشین با داشتن داده‌های فراوان می‌توانند الگوهای پیچیده را شناسایی کنند. مزیت اصلی روش‌های یادگیری ماشین سازگاری با تغییرات در محیط است [۲]. به عنوان مثال، اعداد دستنوشته برای هر فرد دارای الگوهای متفاوت می‌باشد.

به طور کلی روش‌های یادگیری ماشین به دو دسته بانظارت^۱ و بی‌نظرارت^۲ تقسیم می‌شوند [۲]. روش‌های بانظارت یا دسته‌بندی^۳ توسط نمونه‌های آموزشی با برچسب آموزش داده می‌شوند. برای مثال، در تشخیص خودکار بیماری قلب، نمونه‌های آموزشی شامل ویژگی‌ها و علائم بیمار است و برچسب‌ها نیز نشان می‌دهد که آیا یک شخص بیماری قلبی دارد یا خیر. بعد از اتمام فرآیند یادگیری با نمونه‌های آموزشی، روش بانظارت باید نمونه‌های بدون برچسب را تشخیص دهد. از طرف دیگر، نمونه‌های

¹Supervised

²Unsupervised

³Classification

آموزشی در یادگیری بدون ناظارت فاقد برچسب هستند. روش‌های بیناظارت یا خوشه‌بندی^۴ معمولاً نمونه‌ها را به گروه‌های مشابه تقسیم می‌کنند. روش‌های دسته‌بندی در بسیاری از مسائل مختلف استفاده شده‌اند. زیرا مسائل زیادی به صورت ویژگی‌ها و متغیر هدف بیان می‌شوند. برای مثال، می‌توان به مسائل تشخیص چهره، تشخیص هرزنامه،^۵ تشخیص بیماری‌ها، دسته‌بندی متن و تشخیص حمله به شبکه‌های کامپیوتری اشاره کرد. به طور کلی الگوریتم‌های دسته‌بندی دارای دو مرحله هستند:

۱. مرحله آموزش: در این مرحله، یک مدل خروجی از داده‌های آموزشی ساخته می‌شود.

۲. مرحله تست: در این مرحله، مدل ساخته شده برای پیش‌بینی برچسب یک نمونه تست استفاده می‌شود.

الگوریتم‌های بسیار زیادی برای دسته‌بندی داده‌ها توسعه و گسترش یافته‌اند. برخی از مشهورترین آن‌ها شامل درخت تصمیم^۶، گراف نزدیک‌ترین همسایه^۷، بیز^۸، شبکه‌های عصبی^۹ و ماشین بردار پشتیبان (SVM^{۱۰}) هستند [۳]. هر کدام از این الگوریتم‌های دسته‌بندی نقاط قوت و ضعفی دارند و برای مسائل مشخصی بهتر عمل می‌کنند. در میان روش‌های دسته‌بندی اشاره شده، ماشین بردار پشتیبان دقیق و تعمیم‌پذیری بهتری دارد [۳]. این روش دسته‌بندی توسط وپنیک^{۱۱} و کورتس^{۱۲} در سال ۱۹۹۵ ارائه شد [۴].

ماشین بردار پشتیبان بر پایه کمینه کردن ریسک ساختاری^{۱۳} طراحی شده است [۵]. ایده اصلی SVM پیدا کردن یک ابرصفحه^{۱۴} با بیشترین فاصله ممکن از داده‌های دو کلاس می‌باشد. بطوریکه یک مسئله بهینه‌سازی^{۱۵} از نوع برنامه‌ریزی درجه دو (QPP^{۱۶}) برای بدست آوردن چنین ابرصفحه‌ای حل می‌شود. این روش یادگیری در مسائل مختلف مانند تشخیص آریتمی‌های قلبی [۶]، شناسایی نفوذ به شبکه‌های کامپیوتری [۷]، دسته‌بندی متن [۸] و شناسایی هرزنامه [۹] مورد استفاده قرار گرفته است.

⁴Clustering

⁵Spam

⁶Decision Tree

⁷Nearest Neighbor

⁸Bayes

⁹Neural Networks

¹⁰Support Vector Machine

¹¹Vapnik

¹²Cortes

¹³Structural Risk

¹⁴Hyperplane

¹⁵Optimization

¹⁶Quadratic Programming Problem

در دو دهه گذشته، پژوهشگران دسته‌بندهایی مبتنی بر روش ماشین بردار پشتیبان ارائه کردند [۱۰]. در میان گسترش‌های روش SVM، ماشین بردار پشتیبان دو قلو (TSVM^{۱۷}) بیشتر از سایرین مورد توجه پژوهشگران قرار گرفته است. روش TSVM با هدف بهبود پیچیدگی زمانی SVM در سال ۲۰۰۷ ارائه گردید [۱۱]. ایده اصلی این روش یادگیری، بدست آوردن دو ابرصفحه غیر موازی است. بطوریکه هر ابرصفحه غیر موازی به نمونه‌های کلاس خود نزدیک است و نمونه‌های کلاس مقابله دور می‌شود. دو مسئله بهینه‌سازی کوچک از نوع برنامه‌ریزی درجه دو برای بدست آوردن این دو ابرصفحه غیر موازی حل می‌گردد. در حالی‌که در روش SVM یک مسئله بهینه‌سازی بزرگ حل می‌شود. در نتیجه، روش ماشین بردار پشتیبان دو قلو در تئوری 4 برابر سریع‌تر از روش SVM است.

۲-۱ تعریف مسئله

اگرچه ماشین بردار پشتیبان دو قلو نسبت به SVM اصلی سریع‌تر است و داده‌های نامتوزان را بهتر دسته‌بندی می‌کند. با این حال، این روش یادگیری نقاط ضعفی نیز دارد که عبارتند از:

۱. در دسته‌بند TSVM، دو مسئله بهینه‌سازی از نوع برنامه‌ریزی درجه دو باید حل گردد. چنانچه نمونه‌های آموزشی بسیار زیاد باشد، سرعت یادگیری این روش به شدت کند می‌شود. زیرا مرتبه زمانی حل کردن یک مسئله بهینه‌سازی درجه دو برابر با $O(n^3)$ است. بطوریکه n نشان دهنده تعداد نمونه‌های آموزشی می‌باشد. برای رفع کردن این مشکل، از روش کمترین مربعات [۱۲]^{۱۸} استفاده می‌گردد. در نتیجه دو دستگاه معادلات خطی به جای دو مسئله بهینه‌سازی درجه دو حل می‌شود. بنابراین سرعت آموزش دسته‌بند بر روی مجموعه داده‌های بزرگ به طور قابل توجه‌ای افزایش می‌یابد.

۲. برخلاف روش SVM اصلی، ماشین بردار پشتیبان دو قلو ریسک ساختاری^{۱۹} را در مسئله بهینه‌سازی خود کمینه می‌کند. [۱۳] این مسئله، موجب پدیده برازش بیش از حد^{۲۰} می‌شود. به عبارت دیگر، مدل خروجی تمام نمونه‌های آموزشی را به خوبی دسته‌بندی می‌کند. بطوریکه دقت مدل خروجی روی نمونه‌های تست کاهش می‌یابد. این مشکل، قدرت تعمیم‌پذیری^{۲۱} روش ماشین بردار پشتیبان دو قلو را کم می‌کند. برای برطرف کردن این مشکل، در سال ۲۰۱۱، شائو و همکاران، ماشین بردار

¹⁷Twin Support Vector Machine

¹⁸Least Squares

¹⁹Empirical Risk

²⁰Overfitting

²¹Generalization

۱-۳. نوآوری‌های پژوهش

پشتیبان دو قلو مبتنی بر مرز (TBSVM²²) را ارائه کردند [۱۳]. این روش، به مسئله بهینه‌سازی روش TSVM اصلی، یک جمله رگولارسیون^{۲۳} اضافه می‌کند تا مانند روش SVM اصلی حاشیه بیشینه گردد.

۳. ماشین بردار پشتیبان دو قلو به تمام نمونه‌های آموزشی اهمیت یکسانی می‌دهد. در نتیجه ابرصفحه غیر موازی به نمونه‌های نویزی و پرت^{۲۴} نیز نزدیک می‌شود. بنابراین دقت و تعمیم‌پذیری مدل ایجاد شده روی نمونه‌های تست کاهش می‌یابد. ایده اصلی این تحقیق، حل کردن این مسئله است. در پژوهش‌های پیشین نیز به این مسئله پرداخته شده است. برای مثال، روش ماشین بردار پشتیبان دو قلو وزن دار با اطلاعات محلی (WLTSVM²⁵) [۱۴] با ایجاد گراف نزدیک‌ترین همسایه، به نمونه‌های آموزشی وزن نسبت می‌دهد تا اثر نویز و نمونه‌های پرت در ایجاد مدل برای دسته‌بندی کاهش یابد.

۱-۳ نوآوری‌های پژوهش

این پژوهش با ایده گرفتن از روش یادگیری WLTSVM [۱۴]، دو دسته‌بند جدید ارائه می‌کند. بطوريکه نقاط ضعف بیان شده در بخش تعریف مسئله را حل می‌کند. دستاورد این پژوهش شامل معرفی دسته‌بند ماشین بردار پشتیبان دو قلو کمترین مربuat مبتنی بر نزدیک‌ترین همسایه [۱۵] (KNN-LSTSVM²⁶) که مزیت‌های زیر را دارد:

۱. به منظور کاهش اثر نمونه‌های نویزی و پرت، دسته‌بند پیشنهادی KNN-LSTSVM همانند روش WLTSVM از گراف نزدیک‌ترین همسایه بهره می‌گیرد. بطوريکه در مسئله بهینه‌سازی به نمونه‌های آموزشی وزن داده می‌شود و نمونه‌های حاشیه‌ای کلاس مقابله مشخص می‌گردد. این ویژگی باعث بهبود دقت دسته‌بندی مدل خروجی می‌شود.

۲. دسته‌بند پیشنهادی همانند روش WLTSVM باید گراف نزدیک‌ترین همسایه را محاسبه کند تا بتواند وزن تمام نمونه‌های آموزشی را بدست آورد. مرتبه زمانی ایجاد این گراف و ماتریس وزن برابر با $\mathcal{O}(n^2 \log n)$ است. جهت بهبود سرعت یادگیری، دو دستگاه معادلات خطی به جای دو مسئله

²²Twin Bounded Support Vector Machine

²³Regularization

²⁴Outlier

²⁵Weighted Twin Support Vector Machine with Local Information

²⁶KNN-based Least Squares Twin Support Vector Machine

بهینه‌سازی درجه دو حل می‌شود. از این رو سرعت آموزش روش پیشنهادی به طور قابل توجه‌ای بیشتر از روش WLTSVM است و پیاده‌سازی آن نیز ساده‌تر می‌باشد. زیرا دسته‌بند KNN-LSTSVM با استفاده از روش کمترین مربعات، نیازی به الگوریتم‌های حل مسائل بهینه‌سازی ندارد.

۳. با وزن دهی به نمونه‌ها و در نظر گرفتن نمونه‌های حاشیه‌ای، مدل خروجی در دسته‌بند پیشنهادی نسبت به نمونه‌های نویزی و پرت حساسیت کمتری دارد. زیرا این نمونه‌ها وزن بسیار کمتری دارند.

در ادامه با هدف بهبود روش WLTSVM، دسته‌بند ماشین بردار پشتیبان دو قلو مبتنی بر رگولارسیون و نزدیک‌ترین همسایه (RKNN-TSVM²⁷) ارائه شده است که ویژگی‌های زیر را دارد:

۱. با وجود اینکه روش WLTSVM با استفاده از گراف نزدیک‌ترین همسایه، به نمونه‌های آموزشی وزن می‌دهد، روش وزن دهی صرفا بر اساس شمارش تعداد همسایه‌های یک نمونه از طریق گراف نزدیک‌ترین همسایه است. جهت بهبود شیوه وزن دهی به نمونه‌ها، دسته‌بند (RKNN-TSVM) به یک نمونه آموزشی بر اساس فاصله آن نمونه با همسایه‌های نزدیک خود وزن می‌دهد. مزیت این روش وزن دهی این است که نمونه‌های با چگالی بالا^{۲۸} بهتر از نمونه‌های نویزی و پرت تفکیک و شناسایی می‌شود. همچنین به نمونه‌هایی که همسایه‌شان نزدیک‌تر است، وزن بیشتری نسبت داده می‌شود.

۲. دسته‌بند پیشنهادی (RKNN-TSVM) برخلاف روش WLTSVM و TSVM اصلی، ریسک ساختاری را کمینه می‌کند. بدین منظور یک جمله رگولارسیون به مسئله بهینه‌سازی روش پیشنهادی اضافه شده است. هدف مانند SVM اصلی، بیشینه کردن مرز یا حاشیه است. در مجموع، مدل خروجی دچار پدیده برآذش بیش از حد نمی‌شود و از تعمیم‌پذیری بهتری برخوردار است.

۳. روش WLTSVM برای ساخت گراف نزدیک‌ترین همسایه از الگوریتم جستجوی کامل (FSA²⁹) استفاده می‌کند که مرتبه زمانی آن برابر با $O(n^2)$ است. با این حال روش پیشنهادی (-RKNN-TSVM) از الگوریتم نزدیک‌ترین همسایه مبتنی بر تفاوت مکانی فاصله‌ها (LDMDBA³⁰) بهره می‌برد [۱۶]. مرتبه زمانی این الگوریتم برابر با $O(logdnlogn)$ می‌باشد که از الگوریتم FSA کمتر

²⁷Regularized KNN-based Twin Support Vector Machine

²⁸High-density samples

²⁹Full Search Algorithm

³⁰Location difference of multiple distances based k-nearest neighbors algorithm

است. همچنین الگوریتم LDMDBA برای نسخه غیرخطی روش پیشنهادی موثرتر است. زیرا پیدا کردن نزدیک‌ترین همسایه‌های یک نمونه در فضای ویژگی با ابعاد بالا با این الگوریتم سریع‌تر از روش FSA می‌باشد.

۴-۱ ساختار کلی پایاننامه

در فصل ۲، ماشین بردار پشتیبان و گسترش‌های آن از جمله روش TSVM بررسی و تشریح شده است. همچنین در این فصل پیشینه پژوهش SVM مرور شده است و گسترش‌های این روش نیز معرفی شده‌اند.

دسته‌بند پیشنهادی KNN-LSTSVM در فصل ۳ ارائه شده است. نسخه خطی و غیرخطی این دسته‌بند در این فصل تشریح شده است. روش پیشنهادی علاوه بر داشتن مزایای روش WLTSVM، سرعت یادگیری آن نیز به‌وسیله روش کمترین مربعات بهبود یافته است.

در فصل ۴، دسته‌بند RKNN-TSVM معرفی شده است و در ادامه نسخه خطی و غیرخطی آن بیان شده است. این دسته‌بند نقاط ضعف روش WLTSVM را حل می‌کند. بطوریکه مدل خروجی با روش جدید وزن‌دهی حساسیت کمتری نسبت به داده‌های نویزی و پرت خواهد داشت.

دو دسته‌بند پیشنهادی KNN-LSTSVM و RKNN-TSVM در فصل ۵ به طور جامع مورد بررسی و ارزیابی قرار گرفته‌اند. سنجش عملکرد دسته‌بندها با مجموعه داده‌های مصنوعی و واقعی انجام شده است. همچنین سرعت یادگیری نیز با مجموعه داده‌های بزرگ بررسی شده است.

در فصل آخر، ویژگی‌های دسته‌بندهای پیشنهادی مرور شده است. همچنین یافته‌های اصلی این پژوهش نیز در این فصل ذکر شده است. در نهایت پیشنهادهایی برای پژوهش‌های آینده ارائه شده است.

فصل ۲

پژوهش

قبل از معرفی دسته‌بند پیشنهادی، شناخت ماشین بردار پشتیبان، گسترش دو قلو آن و ویژگی‌های این گونه دسته‌بند ضروری است. بدین جهت ماشین بردار پشتیبان و کارهای پیشین آن در این فصل به طور کامل بررسی و تشریح می‌شود.

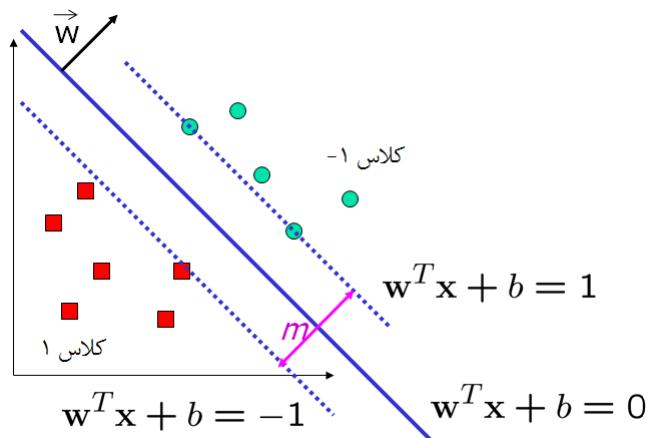
۱-۲ ماشین بردار پشتیبان

در این بخش، ابتدا ماشین بردار پشتیبان خطی و غیر خطی شرح داده می‌شود. سپس پیشینه پژوهش در زمینه ماشین بردار پشتیبان بررسی می‌گردد.

۱-۱-۲ ماشین بردار پشتیبان با حاشیه سخت

ماشین بردار پشتیبان با هدف جداسازی نمونه‌های دو کلاس در سال ۱۹۹۵ معرفی گردید [۴]. ایده اصلی این روش یادگیری، بدست آوردن ابرصفحه بهینه‌ای است که از نمونه‌های دو کلاس تا جای ممکن بیشترین فاصله را داشته باشد. به عبارت دیگر، این روش یادگیری حاشیه دو کلاس را بیشینه می‌کند. برای فهم بهتر ایده این روش، فرض کنید مجموعه داده‌ی $\{(x_1, y_1), (x_2, y_2), \dots, (x_m, y_m)\}$ را در اختیار داریم که شامل m تا نمونه آموزشی است. هر نمونه آموزشی $x_i \in \mathbb{R}^n$ با n ویژگی در فضای ورودی و $\{-1, 1\} \ni y_i$ برحسب نمونه‌ی x_i می‌باشد. در ساده‌ترین حالت، نمونه‌های دو کلاس در مجموعه داده T با یک ابرصفحه $w^T x + b$ بدون خطای دسته‌بندی می‌شود. این حالت مسئله حاشیه سخت^۱ نامیده می‌شود. شکل ۱-۲ مسئله حاشیه سخت در روش SVM را نشان می‌دهد. (لازم به ذکر است، خطوط نقطه‌چین در شکل ۱-۲ نشان دهنده حاشیه است). در این حالت، یک مسئله بهینه‌سازی برای بدست آوردن ابرصفحه باید حل گردد.

¹Hard margin



شکل ۱-۲: مسئله حاشیه سخت در ماشین بردار پشتیبان

$$\begin{aligned} & \min_w \frac{1}{2} \|w\|^2 \\ \text{s.t. } & y_i(w^T x_i + b) \geq 1, \forall i \end{aligned} \quad (1-2)$$

در رابطه ۱-۲، بردار w مختصات ابرصفحه و b بایاس است. قید این مسئله بهینه‌سازی بیان می‌کند که تمام نمونه‌های آموزشی باید از ابرصفحه به مقدار ۱ یا بیشتر فاصله داشته باشند. به عبارت دیگر، تمام نمونه‌های آموزشی باید روی خط حاشیه یا قبل از آن قرار بگیرند. برای حل کردن مسئله بهینه‌سازی ۱-۲ از تابع لاغرانژ^۲ استفاده می‌شود که در رابطه زیر تعریف شده است.

$$L(w, b, \alpha) = \frac{1}{2} \|w\|^2 - \sum_{i=1}^m \alpha_i (y_i(w^T x_i + b) - 1) \quad (2-2)$$

در رابطه ۲-۲، بردار α نشان دهنده ضرایب لاغرانژ است. برای حل کردن رابطه ۲-۲، از تابع لاغرانژ نسبت به w و b مشتق می‌گیریم.

$$\frac{\partial L}{\partial b} = 0 \Rightarrow \sum_{i=1}^m \alpha_i y_i = 0 \quad (3-2)$$

$$\frac{\partial L}{\partial w} = 0 \Rightarrow w = \sum_{i=1}^m \alpha_i y_i x_i. \quad (4-2)$$

با استفاده از اصل دوگانگی لاغرانژ، مسئله اصلی^۳ در رابطه ۱-۲ می‌تواند به مسئله دوگان^۴ تبدیل شود که حل کردن آن آسان‌تر از مسئله اصلی خواهد بود. با در نظر گرفتن روابط ۳-۲ و ۴-۲، مسئله

²Lagrange³Primal problem⁴Dual problem

دوگان برای حالت حاشیه سخت به صورت زیر تعریف می‌شود.

$$\begin{aligned} \min_{\alpha} & \frac{1}{2} \sum_{i=1}^m \sum_{j=1}^m \alpha_i \alpha_j y_i y_j x_i^T x_j - \sum_{k=1}^m \alpha_k \\ \text{s.t.} & \sum_{i=1}^m \alpha_i y_i = 0, \\ & \alpha_i \geq 0, \quad \forall i \end{aligned} \tag{5-2}$$

بعد از حل کردن مسئله دوگان ۵-۲، اکثر مقادیر بردار α برابر با صفر خواهد. با این حال ضرایب لاغرانژ α_i متناظر با نمونه‌های x_i بزرگتر از صفر خواهد بود، اگر معادله زیر برقرار باشد.

$$y_i(w^T x_i + b) = 1 \tag{6-2}$$

نمونه‌های که رابطه ۶-۲ برای آنها برقرار باشد، به اصطلاح بردار پشتیبان^۵ نامیده می‌شود. ضرایب لاغرانژ متناظر با بردارهای پشتیبان بزرگ‌تر از صفر است. همچنین این بردارها روی حاشیه قرار می‌گیرند. بردار w و بایاس b از طریق رابطه بدست می‌آید.

$$\begin{aligned} w &= \sum_{i=1}^m \alpha_i y_i x_i \\ b &= y_i - w^T x_i \end{aligned} \tag{7-2}$$

یک نمونه جدید یا تست بر اساستابع تصمیم در رابطه زیر به یکی از کلاس‌های ۱ و ۰-۱- تعلق می‌گیرد.

$$D(x) = sign(w^T x + b) = sign\left(\sum_{i=1}^m \alpha_i y_i x_i^T x + b\right) \tag{8-2}$$

در مسئله حاشیه سخت، فرض گرفته می‌شود که تمام نمونه‌های دو کلاس به صورت خطی جدا پذیر هستند. با این حال در دنیای واقعی داده‌ها اغلب به صورت خطی جدا پذیر نیستند. در حالتی که داده‌ها به صورتی خطی جداپذیر نیستند، مسئله حاشیه نرم^۶ در ماشین بردار پشتیبان مطرح شده است.

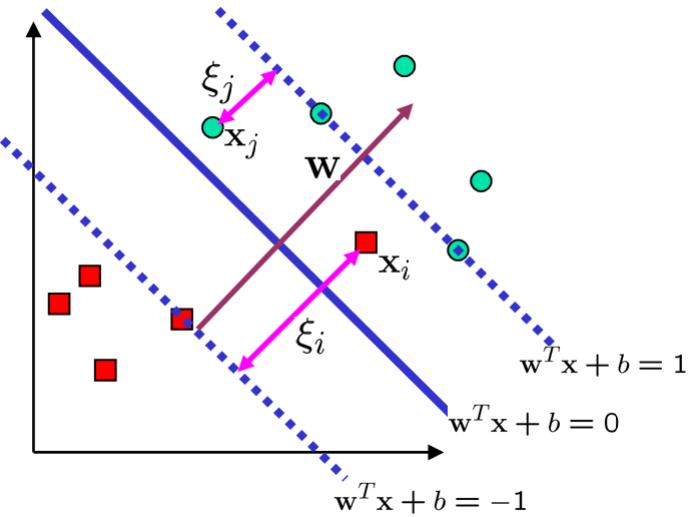
۲-۱-۲ ماشین بردار پشتیبان با حاشیه نرم

در این حالت اجازه خطأ در دسته‌بندی نمونه‌ها داده می‌شود. به عبارت دیگر، با معرفی متغیر کمکی^۷ ξ ، تاثیر بعضی از نمونه‌های آموزشی در ایجاد حاشیه و ابرصفحه کم می‌شود. برای درک بهتر، مسئله حاشیه نرم در ماشین بردار پشتیبان در شکل ۲-۲ نشان داده شده است. در حالت حاشیه نرم، برای بدست آوردن ابرصفحه مسئله بهینه‌سازی به صورت زیر تعریف می‌شود.

⁵Support Vector

⁶Soft margin

⁷Slack variable



شکل ۲-۲: مسئله حاشیه نرم در ماشین بردار پشتیبان

$$\begin{aligned} \min_w \quad & \frac{1}{2} \|w\|^2 + C \sum_{i=1}^n \xi_i \\ \text{s.t.} \quad & y_i(w^T x_i + b) + \xi_i \geq 1, \forall i \end{aligned} \quad (9-2)$$

در رابطه ۹-۲، متغیر $\xi_i \geq 0$ خطای متناظر برای هر نمونه x_i است. پارامتر C بیانگر تعادل بین بیشینه کردن حاشیه و خطای دسته‌بندی است. مقدار این پارامتر قبل از آموزش دسته‌بند باید مشخص گردد. مانند مسئله حاشیه سخت، برای حل کردن مسئله اصلی از تابع لاگرانژ استفاده می‌شود که در رابطه زیر تعریف شده است.

$$L(w, b, \xi, \alpha, \beta) = \frac{1}{2} \|w\|^2 + C \sum_{i=1}^m \xi_i - \sum_{i=1}^m \alpha_i (y_i(w^T x_i + b) - 1 + \xi_i) - \sum_{i=1}^m \beta_i \xi_i \quad (10-2)$$

در رابطه ۱۰-۲، بردار w و ضرایب α و β ضرایب لاگرانژ هستند. برای حل کردن رابطه ۱۰-۲، نسبت به w و b مشتق می‌گیریم.

$$\frac{\partial L}{\partial b} = 0 \Rightarrow \sum_{i=1}^m \alpha_i y_i = 0 \quad (11-2)$$

$$\frac{\partial L}{\partial w} = 0 \Rightarrow w = \sum_{i=1}^m \alpha_i y_i x_i. \quad (12-2)$$

$$\frac{\partial L}{\partial \xi} = 0 \Rightarrow \alpha_i + \beta_i = C \quad (13-2)$$

با در نظر گرفتن روابط بالا، مسئله دوگان برای حالت حاشیه نرم به صورت زیر تعریف می‌شود.

$$\begin{aligned} \min_{\alpha} \quad & \frac{1}{2} \sum_{i=1}^m \sum_{j=1}^m \alpha_i \alpha_j y_i y_j x_i^T x_j - \sum_{k=1}^m \alpha_k \\ \text{s.t.} \quad & \sum_{i=1}^m \alpha_i y_i = 0, \\ & 0 \leq \alpha_i \leq C, \quad \forall i \end{aligned} \quad (14-2)$$

راه حل مسئله ۱۴-۲ شبیه به مسئله ۵-۲ در حالت حاشیه سخت است. با این تفاوت که برای ضرایب لاگرانژ یک حد بین صفر تا پارامتر تعریف شده است. پارامتر انعطاف‌پذیری دسته‌بند را افزایش می‌دهد. مقدار این پارامتر با داشتن دانش قبلی از مسئله و یا جستجوی شبکه‌ای تعیین می‌شود. همانطور که در شکل ۲-۲ نشان داده شده است، بردارهای پشتیبان لزوماً روی خط حاشیه قرار ندارند. در مسئله حاشیه نرم،تابع تصمیم مشابه حالت حاشیه سخت به صورت زیر تعریف می‌شود.

$$D(x) = sign(w^T x + b) = sign\left(\sum_{i \in S} \alpha_i y_i x_i^T x + b\right) \quad (15-2)$$

در رابطه ۱۵-۲، مجموعه S نشان دهنده بردارهای پشتیبان است.

۳-۱-۲ ماشین بردار پشتیبان با هسته غیر خطی

در مسائل حاشیه سخت و نرم، فرض گرفته می‌شود که نمونه‌ها در فضای ورودی به صورت خطی جدا پذیر هستند. با این حال در مواردی که نمونه‌ها به صورت خطی جدا پذیر نیستند، نمونه‌ها x_i به فضای ویژگی^۸ با ابعاد بیشتر با تکنیک حقه‌ی هسته^۹ نگاشت می‌شود. در این فضا، ماشین بردار پشتیبان یک ابرصفحه جدا کننده بهینه برای جداسازی نمونه‌ها پیدا می‌کند. شکل ۳-۲ تکنیک حقه‌ی هسته را نشان می‌دهد.

در نسخه غیر خطی ماشین بردار پشتیبان، مسئله ۱۴-۲ به صورت زیر تعریف می‌شود.

$$\begin{aligned} \min_{\alpha} \quad & \frac{1}{2} \sum_{i=1}^m \sum_{j=1}^m \alpha_i \alpha_j y_i y_j K(x_i, x_j) - \sum_{k=1}^m \alpha_k \\ \text{s.t.} \quad & \sum_{i=1}^m \alpha_i y_i = 0, \\ & 0 \leq \alpha_i \leq C, \quad \forall i \end{aligned} \quad (16-2)$$

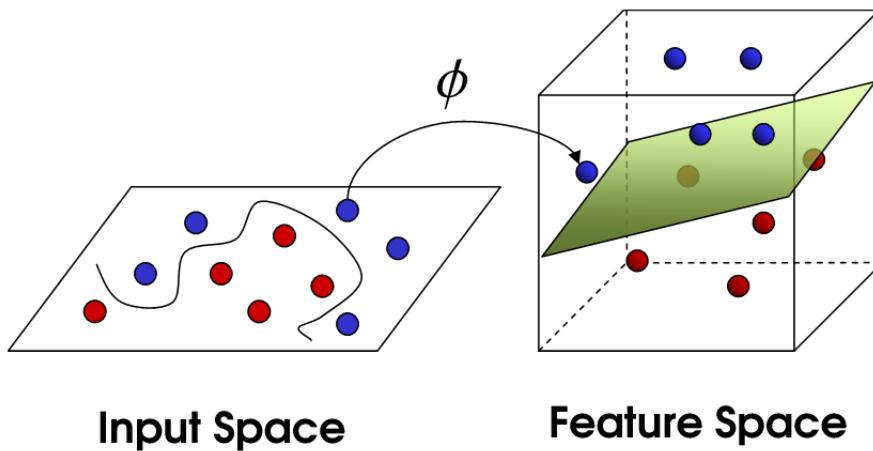
در رابطه ۱۶-۲، تابع هسته $K(x_i, x_j)$ نمونه‌ها را به فضای ویژگی نگاشت می‌کند. تابع‌های هسته متداول شامل^{۱۰} RBF، چند جمله‌ای و سیگموئید^{۱۱} هستند. تابع تصمیم در نسخه غیر خطی به صورت

⁸Feature space

⁹Kernel trick

¹⁰Radial basis function

¹¹Sigmoid



شکل ۲-۳: نگاشت نمونه‌ها با تکنیک حقه‌ی هسته

زیر تعریف می‌شود.

$$D(x) = \text{sign}(\sum_{i \in S} \alpha_i y_i K(x_i, x) + b) \quad (17-2)$$

۴-۱-۲ ماشین بردار پشتیبان چند کلاسه

ماشین بردار پشتیبان برای دسته‌بندی دو کلاس ارائه شده است. به منظور تعمیم ماشین بردار پشتیبان به مسائل چند کلاسه، به طور کلی دو روش وجود دارد [۱۷]:

۱. ایجاد و ترکیب چندین دسته‌بند دو کلاسه برای حل کردن مسائل چند کلاسه

۲. لحاظ کردن تمام نمونه‌های آموزشی در یک مسئله بهینه‌سازی بزرگ

روش اول برای حل مسائل چند کلاسه بیشتر مورد توجه گرفته است. بطوریکه سه نوع روش ایجاد و ترکیب چندین دسته‌بند شامل یک-در مقابل-بقیه^{۱۲}، یک-در مقابل-یک^{۱۳} و گراف جهت دار یک-در مقابل-یک ارائه شده است [۱۸، ۱۷]. در ادامه هر یک از این روش‌های چند کلاسه توضیح داده می‌شود.

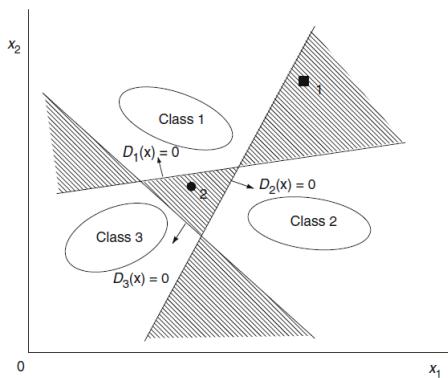
۱-۴-۱-۲ یک-در مقابل-بقیه

روش یک-در مقابل-بقیه قدیمی‌ترین روش چند کلاسه برای دسته‌بند SVM است. برای تشریح این روش

چند کلاسه، ابتدا یک مسئله دسته‌بندی k کلاسه با مجموعه داده $\{(x_1, y_1), (x_2, y_2) \dots, (x_m, y_m)\}$

¹²One-Against-All

¹³One-Against-One



شکل ۴-۲: نواحی غیر قابل دسته‌بندی در روش یک-در مقابل-بقیه [۱۸]

را در نظر می‌گیریم. $x_i \in \mathbb{R}^n$ نشان‌دهنده نمونه آموزشی و $y_i \in \{1, \dots, k\}$ کلاس نمونه x_i را نشان می‌دهد. دسته‌بند SVM دو دسته‌ای در روش یک-در مقابل-بقیه آموزش داده می‌شود. بطوریکه k برابر با تعداد کلاس‌ها است. در اینجا نمین تابع تصمیم $D_i(x) = w_i^T \varphi(x) + b_i$ کلاس i را از سایر کلاس‌ها جدا می‌کند.

به منظور پیدا کردن کلاس داده تست، ابتدا مقادیر تابع تصمیم $D_i(x)$ برای k دسته‌بند محاسبه می‌شود. چنانچه به ازای فقط یک i مقدار $0 < D_i(x) <$ برقرار باشد، آنگاه داده تست به کلاس i تعلق دارد. با این حال امکان دارد که برای یک داده تست چندین مقدار D_i مثبت یا مقادیر تمامی توابع تصمیم منفی شوند. در این حالت، امکان دسته‌بندی داده تست وجود ندارد. این مشکل تحت عنوان نواحی غیر قابل دسته‌بندی^{۱۴} شناخته می‌شود. شکل ۴-۲ نواحی غیر قابل دسته‌بندی در روش یک-در مقابل-بقیه را نشان می‌دهد. این نواحی در شکل با هاشور مشخص شده‌اند.

برای مثال، نمونه ۱ در شکل ۴-۲ در ناحیه تصمیم دو کلاس ۱ و ۲ قرار دارد. بطوریکه مقدار $D_1(x_1) > 0$ و $D_2(x_1) < 0$ برقرار است. بنابراین کلاس نمونه ۱ غیر قابل تشخیص است.

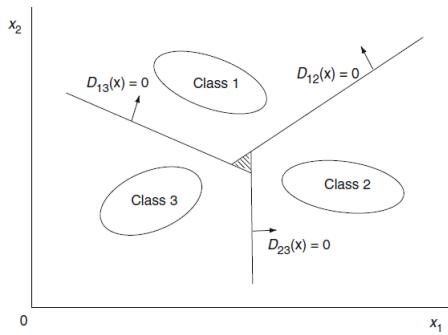
۲-۴-۱-۲ یک-در مقابل-یک

این روش برای یک مسئله دسته‌بندی با k کلاس، $k/(k-1)/2$ دسته‌بند دو دسته‌ای ایجاد می‌کند [۱۷]. در اینجا هر دسته‌بند فقط با نمونه‌های دو کلاس آموزش داده می‌شود. تابع تصمیم $D_{ij}(x) = w_{ij}^T \varphi(x) + b_{ij}$ در روش یک-در مقابل-یک کلاس i را از کلاس j جدا می‌کند.

به منظور تعیین کلاس داده تست در روش یک-در مقابل-یک، از رای‌گیری^{۱۵} استفاده می‌شود.

¹⁴Unclassifiable

¹⁵Voting



شکل ۲-۵: نواحی غیر قابل دسته‌بندی در روش یک-در مقابل-یک [۱۸]

چنانچه $\text{sign}(w_{ij}^T \varphi(x) + b_{ij})$ مشخص کند که نمونه تست x به کلاس i ام تعلق دارد، آنگاه رای برای کلاس i ام یکی افزایش پیدا می‌کند. در غیر این صورت، رای کلاس j ام یکی افزایش می‌یابد. در آخر نمونه تست x به کلاسی با بیشترین رای تعلق می‌گیرد. شکل ۲-۵ نواحی غیر قابل دسته‌بندی در روش یک-در مقابل-یک را نشان می‌دهد.

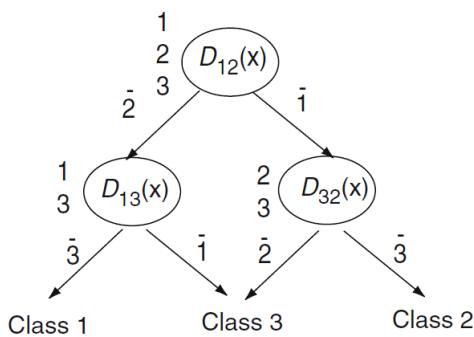
همانطور که در شکل ۲-۵ مشخص است، نواحی غیر قابل دسته‌بندی در روش یک-در مقابل-یک بسیار کوچکتر از روش یک-در مقابل-بقیه است. روش یک-در مقابل-یک با وجود داشتن دقت دسته‌بندی بهتر، پیچیدگی محاسباتی زیادی دارد. برای مثال، یک مسئله با ۱۰ کلاس نیاز به آموزش ۴۵ دسته‌بند دودسته‌ای دارد.

۳-۴-۱-۲ گراف جهت‌دار یک-در مقابل-یک

به منظور برطرف کردن نواحی غیر قابل دسته‌بندی در روش یک-در مقابل-یک، گراف تصمیم جهت‌دار غیر چرخه‌ای (DAG^{۱۶}) ارائه شده است [۱۹]. فرآیند آموزش این روش همانند روش یک-در مقابل-یک است و $k(k-1)/2$ دسته‌بند دودسته‌ای را ایجاد می‌کند. با این حال بدست آوردن کلاس نمونه تست با استفاده از یک گراف دودویی DAG با $k(k-1)/2$ گره میانی و k برگ انجام می‌شود. در واقع هر گره در این گراف، یک دسته‌بند دودسته‌ای برای کلاس i و j است. شکل ۶-۲ گراف دودویی DAG برای سه کلاس نشان می‌دهد.

پیدا کردن کلاس داده تست از گره ریشه آغاز می‌شود و سپس با توجه به خروجیتابع تصمیم حرکت به سمت چپ یا راست گراف صورت می‌گیرد. حرکت تا رسیدن به گره برگ ادامه می‌یابد. گره برگ نشان‌دهنده کلاس داده تست است. دیگر مزیت روش DAG نسبت به روش یک-در مقابل-یک

^{۱۶}Directed Acyclic Graph



شکل ۲-۶: گراف دودویی DAG برای سه کلاس [۱۸]

کمتر بودن زمان تست می‌باشد.

۵-۱-۲ پیشینه پژوهش در ماشین بردار پشتیبان

ماشین بردار پشتیبان بیان ریاضی قوی‌ای دارد و دارای قدرت تعمیم‌پذیری بالایی است. از این رو در مسائل مختلف مانند تشخیص آریتمی‌های قلبی [۶]، شناسایی نفوذ به شبکه‌های کامپیوتری [۷]، دسته‌بندی متن [۸] و شناسایی هرزنامه [۹] مورد استفاده قرار گرفته است. با این حال ماشین بردار پشتیبان نقاط ضعفی نیز دارد که مهم‌ترین آن‌ها عبارتند از:

۱. ماشین بردار پشتیبان برای بدست آوردن ابرصفحه بهینه یک مسئله بهینه‌سازی از نوع برنامه‌ریزی درجه دو حل می‌کند. مرتبه زمانی حل کردن چنین مسئله‌ای برابر با $\mathcal{O}(m^3)$ است. m تعداد نمونه‌های آموزشی می‌باشد. این مرتبه زمانی، آموزش روش ماشین بردار پشتیبان را برای مجموعه داده‌های بزرگ به طور قابل توجهی کند می‌کند.
۲. در روش SVM، بردارهای پشتیبان نقش مهمی در ایجاد مدل دارند. در صورتی که این بردارها از نمونه‌های پرت یا نویزی باشد، دقت و تعمیم‌پذیری مدل خروجی کاهش می‌یابد. در نتیجه روش SVM به نمونه‌های پرت و نویزی حساس است.
۳. چنانچه نمونه‌های یک کلاس از کلاس دیگر بسیار بیشتر باشد (مجموعه داده نامتوازن باشد)، مدل ایجاد شده توسط SVM به سمت کلاس اکثریت گرایش پیدا می‌کند. در نتیجه مدل در تشخیص کلاس اقلیت ناتوان است.

در دو دهه اخیر، روش‌های یادگیری جدیدی مبنی SVM ارائه شده است که برخی از آن‌ها نقاط

ضعف بالا را حل می‌کند. در سال ۱۹۹۹، ماشین بردار پشتیبان کمترین مربعات (LS-SVM^{۱۷}) ارائه شد [۲۰]. در این روش قید در مسئله بهینه‌سازی از نامساوی به مساوی تبدیل شده است. بطوریکه به جای حل کردن مسئله بهینه‌سازی از نوع برنامه‌ریزی درجه دو، دستگاه معادلات خطی حل می‌گردد. در نتیجه سرعت یادگیری برای مجموعه داده‌های بزرگ بسیار بیشتر می‌شود و نقطه ضعف مورد اول تا حد زیادی حل شده است.

در سال ۲۰۰۱، ماشین بردار پشتیبان مبتنی بر مفهوم نزدیکی (PSVM^{۱۸}) ارائه شد [۲۱]. در این روش دو ابرصفحه موازی برای دسته‌بندی نمونه‌ها ایجاد می‌شود. در سال ۲۰۰۲، ماشین بردار پشتیبان فازی (FSVM^{۱۹}) ارائه گردید [۲۲]. در این روش به هر یک از نمونه‌های هر دو کلاس، تعلق فازی داده می‌شود. بطوریکه اثر نمونه‌های نویزی و پرت در ایجاد مدل خروجی کم خواهد شد. در سال ۲۰۰۶، ماشین بردار پشتیبان با رویکرد مقدار ویژه تعمیم یافته (GEPSVM^{۲۰}) ارائه شد [۲۳]. برخلاف روش PSVM، این روش دو ابرصفحه غیر موازی ایجاد می‌کند که هر یک از این ابرصفحه‌ها به نمونه‌های کلاس خود نزدیک است و از نمونه‌های کلاس مقابل تا جای ممکن فاصله می‌گیرد. همچنین روش PSVM بر روی مسئله XOR عملکرد بهتری نسبت به روش SVM اصلی دارد.

در سال ۲۰۰۷، ماشین بردار پشتیبان دو قلو (TSVM) با هدف بهبود پیچیدگی زمانی SVM ارائه گردید [۱۱]. ایده اصلی این روش یادگیری، بدست آوردن دو ابرصفحه غیر موازی است. بطوریکه هر ابرصفحه غیر موازی به نمونه‌های کلاس خود نزدیک است و نمونه‌های کلاس مقابل دور می‌شود. دو مسئله بهینه‌سازی کوچک از نوع برنامه‌ریزی درجه دو برای بدست آوردن این دو ابرصفحه غیر موازی حل می‌گردد. در حالی‌که در روش SVM یک مسئله بهینه‌سازی بزرگ حل می‌شود. در نتیجه، روش ماشین بردار پشتیبان دو قلو در تئوری ^۴ برابر سریعتر از روش SVM است. در بخش ۲-۲ ماشین بردار پشتیبان دو قلو و پیشینه پژوهش آن به طور کامل بررسی می‌شود.

در سال‌های اخیر نیز، روش‌های جدید مبتنی بر SVM ارائه شده است. در سال ۲۰۱۸، می‌توان به روش یادگیری برخط مبتنی بر بدترین نمونه‌ی نقض‌کننده (OLLAWV^{۲۱}) اشاره کرد [۲۴]. این الگوریتم بر اساس گرادیان نزولی ^{۲۲} تصادفی طراحی شده است و مسئله اصلی در روش SVM را به جای مسئله

¹⁷Least squares support vector machines

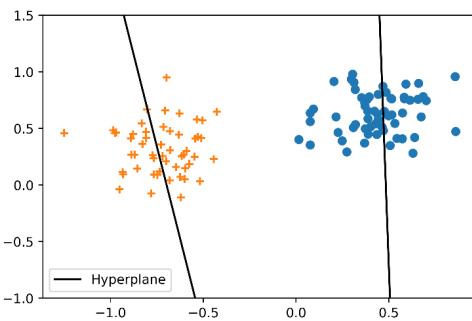
¹⁸Proximal Support Vector Machine

¹⁹Fuzzy Support Vector Machine

²⁰Generalized Eigenvalue Proximal Support Vector Machine

²¹Online Learning Algorithm using Worst-Violators

²²Gradient Descent



شکل ۷-۲: تفسیر هندسی روش ماشین بردار پشتیبان دو قلو خطی

دوگان حل می‌کند. مزایای این روش، دقیق‌تر، سرعت یادگیری بیشتر برای مجموعه داده‌های بزرگ و همچنین مدل خلوت^{۲۳} می‌باشد.

۲-۲ ماشین بردار پشتیبان دو قلو

در این بخش، ابتدا نسخه خطی ماشین بردار پشتیبان دو قلو شرح داده می‌شود. سپس نسخه غیر خطی این روش توضیح داده شده و در آخر پیشینه پژوهش آن و روش‌های مبتنی بر TSVM بررسی می‌شود.

۲-۲-۱ ماشین بردار پشتیبان دو قلو خطی

ایده اصلی روش TSVM، ایجاد دو ابرصفحه غیر موازی است [۱۱]. بطوریکه هر ابرصفحه غیر موازی از نمونه‌های کلاس خود کمترین فاصله را دارد و از نمونه‌های کلاس مقابله حداکثر فاصله ممکن را خواهد داشت. برای تشریح بهتر این روش، یک مسئله دسته‌بندی دوکلاسه با m_1 نمونه آموزشی کلاس مثبت و m_2 نمونه آموزشی کلاس منفی را در نظر می‌گیریم. همچنین فرض می‌کنیم که ماتریس $A \in \mathbb{R}^{m_1 \times n}$ بیانگر نمونه‌های کلاس مثبت و ماتریس $B \in \mathbb{R}^{m_2 \times n}$ بیانگر نمونه‌های کلاس منفی است. روش TSVM در حالت خطی به دنبال دو ابرصفحه غیر موازی در فضای \mathbb{R}^n است که در رابطه زیر تعریف شده است.

$$x^T w_1 + b_1 = 0 \quad \text{and} \quad x^T w_2 + b_2 = 0 \quad (18-2)$$

در رابطه ۱۸-۲، $w^{(1)} \in \mathbb{R}^n$ نشان دهنده مختصات دو ابرصفحه و $w^{(2)} \in \mathbb{R}^n$ نشان دهنده بایاس است. شکل ۷-۲ تفسیر هندسی روش ماشین بردار پشتیبان دو قلو خطی را نشان می‌دهد. در روش TSVM، دو مسئله بهینه‌سازی از نوع برنامه‌ریزی درجه دو برای بدست آوردن دو ابرصفحه غیر موازی به صورت زیر تعریف می‌شود.

²³Sparse

$$\min_{w^{(1)}, b^{(1)}} \quad \frac{1}{2} \|Aw^{(1)} + e_1 b^{(1)}\|^2 + C_1 e_1^T y_2 \quad (19-2)$$

$$\text{s.t.} \quad -(Bw^{(1)} + e_2 b^{(1)}) + y_2 \geq e_2, y_2 \geq 0$$

$$\min_{w^{(2)}, b^{(2)}} \quad \frac{1}{2} \|Bw^{(2)} + e_2 b^{(2)}\|^2 + C_2 e_2^T y_1 \quad (20-2)$$

$$\text{s.t.} \quad (Aw^{(2)} + e_1 b^{(2)}) + y_1 \geq e_1, y_1 \geq 0$$

در روابط ۱۹-۲ و ۲۰-۲، e_1 و e_2 پارامترهای خطای A و B بردار با درایه‌های یک در ابعاد متناسب، y_1 و y_2 متغیر کمکی هستند. لازم به ذکر است که تعداد نمونه‌ها در مسائل بهینه‌سازی روش SVM تقریباً برابر با $m/2$ در نظر گرفته می‌شود. با این حال در روش SVM، در قید مسئله بهینه‌سازی تمام نمونه‌های آموزشی نقش دارند. بنابراین زمان اجرای روش SVM از روش TSVM حدود ۴ برابر سریع‌تر است که زیر نشان داده شده است.

$$\left[(m^3) / (2 \times (\frac{m}{2})^3) \right] = 4. \quad (21-2)$$

مانند روش SVM، برای حل کردن مسائل بهینه‌سازی ۱۹-۲، از تابع لاگرانژ استفاده می‌کنیم.

$$L(w_1, b_1, y_2, \alpha, \beta) = \frac{1}{2} (Aw^{(1)} + e_1 b^{(1)})^T (Aw^{(1)} + e_1 b^{(1)}) + C_1 e_1^T y_2 \quad (22-2)$$

$$- \alpha^T (-Bw^{(1)} + e_2 b^{(1)}) + y_2 - e_2 - \beta^T y_2$$

در رابطه ۲۲-۲، $\alpha = (\alpha_1, \alpha_2, \dots, \alpha_m)^T$ و $\beta = (\beta_1, \beta_2, \dots, \beta_m)^T$ بردارهای ضرایب لاگرانژ هستند. با مشتق‌گیری از رابطه ۲۲-۲، شرایط KKT²⁴ زیر برقرار می‌شود.

$$A^T (Aw^{(1)} + e_1 b^{(1)}) + B^T \alpha = 0. \quad (23-2)$$

$$e_1^T (Aw^{(1)} + e_1 b^{(1)}) + e_2^T \alpha = 0. \quad (24-2)$$

$$C_1 e_2 - \alpha - \beta = 0. \quad (25-2)$$

با توجه اینکه $0 \leq \beta \leq C_2$ ، از رابطه ۲۵-۲ خواهیم داشت:

$$0 \leq \alpha \leq C_1 \quad (26-2)$$

سپس با ترکیب کردن ۲۳-۲ و ۲۴-۲، رابطه زیر بدست می‌آید.

$$[A^T \ e_1^T][A \ e_1][w^{(1)} \ b^{(1)}]^T + [B^T \ e_2^T]\alpha = 0. \quad (27-2)$$

²⁴Karush-Kuhn-Tucker

با تعریف کردن ماتریس H و $G = [B \ e]$ رابطه ۲۷-۲ به صورت زیر بازنویسی می‌شود.

$$\begin{bmatrix} w^{(1)} \\ b^{(1)} \end{bmatrix} = -(H^T H)^{-1} G^T \alpha \quad (28-2)$$

برای جلوگیری از شرایط ماتریس منفرد^{۲۵} در ماتریس $H^T H$ ، یک عدد ثابت بسیار کوچک $\varepsilon I > 0$ به عناصر قطر این ماتریس اضافه می‌شود. بنابراین دستگاه معادلات در رابطه ۲۸-۲ به صورت زیر تغییر می‌یابد.

$$\begin{bmatrix} w^{(1)} \\ b^{(1)} \end{bmatrix} = -(H^T H + \varepsilon I)^{-1} G^T \alpha \quad (29-2)$$

با توجه به شرایط KKT و مسئله اصلی ۱۹-۲، مسئله دوگان برای ۱۹-۲ به صورت زیر تعریف می‌گردد.

$$\begin{aligned} \min_{\alpha} \quad & \frac{1}{2} \alpha^T G (H^T H)^{-1} G^T \alpha - e_1^T \alpha \\ \text{s.t.} \quad & e_2 \leq \alpha \leq C_1 e_2 \end{aligned} \quad (30-2)$$

مشابه راه حل کلاس مثبت، مسئله دوگان برای کلاس منفی ۲۰-۲ به صورت زیر تعریف می‌شود.

$$\begin{aligned} \min_{\alpha} \quad & \frac{1}{2} \alpha^T G (H^T H)^{-1} G^T \alpha - e_2^T \alpha \\ \text{s.t.} \quad & e_2 \leq \alpha \leq C_1 e_2 \end{aligned} \quad (31-2)$$

بعد از حل کردن مسئله دوگان ۳۱-۲، ابرصفحه کلاس منفی از طریق رابطه زیر بدست می‌آید.

$$\begin{bmatrix} w^{(2)} \\ b^{(2)} \end{bmatrix} = (G^T G)^{-1} H^T \beta \quad (32-2)$$

در نسخه خطی روش TSVM، برای بدست آوردن مدل خروجی دو گام محاسباتی وجود دارد که عبارتند از:

۱. ضرایب لاغرانژ α و β به ترتیب با حل کردن مسائل دوگان ۳۱-۲ و ۳۲-۲ بدست می‌آید. مرتبه زمانی این گام برابر با $O(1/4m^3)$ می‌باشد.

۲. دو دستگاه معادلات خطی در روابط ۲۹-۲ و ۳۲-۲ باید حل گردد. در این گام، معکوس ماتریس های $G^T G$ و $H^T H$ باید محاسبه شود. ابعاد این دو ماتریس برابر $(n+1) \times (n+1)$ است. بطوریکه n بسیار کوچکتر از تعداد نمونه‌های کلاس مثبت و منفی می‌باشد.

²⁵Singular

بعد از بدست آمدن دو ابرصفحه غیرموازی ۲-۱۸، داده جدید باتابع تصمیم زیر به یکی از کلاس‌های مثبت یا منفی تعلق می‌گیرد.

$$\arg \min_{j=1,2} |x^T w^{(j)} + b^{(j)}| \quad (33-2)$$

۲-۲-۲ ماشین بردار پشتیبان دو قلو غیر خطی

مسائل یادگیری ماشین در دنیای واقعی غالباً به صورت خطی جدا پذیر نیستند. روش TSVM برای جدا سازی مسائل غیر خطی، نمونه‌ها را به فضای ویژگی با ابعاد بالاتر نگاشت می‌کند. بدین منظور دو ابرسطح^{۲۶} به صورت زیر تعریف می‌شود.

$$K(x^T, C^T)u^{(1)} + b^{(1)} = 0 \text{ and } K(x^T, C^T)u^{(2)} + b^{(2)} = 0 \quad (34-2)$$

در رابطه ۳۴-۲، ماتریس C برابر با $C = [A \ B]^T$ است و K تابع هسته دلخواه می‌باشد. در نسخه غیر خطی، مسائل بهینه‌سازی اصلی برای کلاس مثبت و منفی به ترتیب در روابط ۳۵-۲ و ۳۶-۲ تعریف شده است.

$$\begin{aligned} \min_{u^{(1)}, b^{(1)}} & \frac{1}{2} \|K(A, C^T)u^{(1)} + e_1 b^{(1)}\|^2 + C_1 e_1^T y_1 \\ \text{s.t.} & -(K(B, C^T)u^{(1)} + e_2 b^{(1)}) + y_2 \geq e_2, y_2 \geq 0 \end{aligned} \quad (35-2)$$

$$\begin{aligned} \min_{u^{(2)}, b^{(2)}} & \frac{1}{2} \|K(B, C^T)u^{(2)} + e_2 b^{(2)}\|^2 + C_2 e_2^T y_2 \\ \text{s.t.} & (K(A, C^T)u^{(2)} + e_1 b^{(2)}) + y_1 \geq e_1, y_1 \geq 0 \end{aligned} \quad (36-2)$$

تابع لاگرانژ برای مسئله اصلی در رابطه ۳۵-۲ به صورت زیر تعریف می‌شود.

$$\begin{aligned} L(u_1, b_1, y_2, \alpha, \beta) = & \frac{1}{2} \|K(A, C^T)u^{(1)} + e_1 b^{(1)}\|^2 + C_1 e_1^T y_2 \\ & - \alpha^T(-(K(B, C^T)u^{(1)} + e_2 b^{(1)}) + y_2 - e_2) - \beta^T y_2 \end{aligned} \quad (37-2)$$

مشابه نسخه خطی، شرایط KKT برای رابطه ۳۷-۲ زیر تعریف شده است.

$$K(A, C^T)^T (K(A, C^T)u^{(1)} + e_1 b^{(1)}) + K(B, C^T)^T \alpha = 0. \quad (38-2)$$

$$e_1^T (K(A, C^T)u^{(1)} + e_1 b^{(1)}) + e_2^T \alpha = 0. \quad (39-2)$$

$$C_1 e_2 - \alpha - \beta = 0. \quad (40-2)$$

²⁶Hypersurface

با ترکیب کردن روابط ۳۸-۲ و ۳۹-۲، رابطه زیر بدست می‌آید.

$$[K(A, C^T)^T e_1^T][K(A, C^T) e_1][u^{(1)} b^{(1)}]^T + [K(B, C^T) e_2^T]\alpha = 0. \quad (41-2)$$

با تعریف کردن ماتریس S و $R = [K(B, C^T) e_2]$ و $S = [K(A, C^T) e_1]$ به صورت $41-2$ به صورت بازنویسی می‌شود.

$$\begin{bmatrix} u^{(1)} \\ b^{(1)} \end{bmatrix} = -(S^T S)^{-1} R^T \alpha \quad (42-2)$$

با توجه به شرایط KKT و مسئله اصلی ۳۵-۲، مسئله دوگان برای بدست آوردن ضریب لاغرانژ α به صورت زیر تعریف می‌شود.

$$\begin{aligned} \min_{\alpha} \quad & \frac{1}{2} \alpha^T R (S^T S)^{-1} R^T \alpha - e_1^T \alpha \\ \text{s.t.} \quad & 0 e_2 \leq \alpha \leq C_1 e_2 \end{aligned} \quad (43-2)$$

به طرز مشابه‌ای، مسئله دوگان مسئله اصلی ۳۶-۲ در رابطه زیر تعریف شده است.

$$\begin{aligned} \min_{\beta} \quad & \frac{1}{2} \beta^T S (R^T R)^{-1} S^T \beta - e_2^T \beta \\ \text{s.t.} \quad & 0 e_1 \leq \beta \leq C_2 e_1 \end{aligned} \quad (44-2)$$

بعد از حل کردن مسئله دوگان و بدست آمدن ضرایب لاغرانژ β ، ابرسطح کلاس منفی از طریق رابطه زیر بدست می‌آید.

$$\begin{bmatrix} u^{(2)} \\ b^{(2)} \end{bmatrix} = (R^T R)^{-1} S^T \alpha \quad (45-2)$$

بعد از حل کردن مسائل دوگان ۴۳-۲ و ۴۴-۲، مشابه نسخه خطی یک نمونه جدید به کلاس مثبت یا منفی نسبت داده می‌شود. به عبارت دیگر، براساس فاصله عمودی نمونه جدید از دو ابرسطح، کلاس آن مشخص می‌گردد.

برخلاف نسخه خطی، حل کردن روابط ۴۲-۲ و ۴۵-۲ نیاز به محاسبه معکوس ماتریس در ابعاد $m \times m$ دارد. بطوریکه m برابر با کل نمونه‌های آموزشی است. زمانی که مجموعه داده‌ها بزرگ باشد، بدست آوردن مدل غیر خطی بسیار زمان بر می‌شود. تکنیک هسته‌ی مستطیلی^{۲۷} [۲۱] برای کاهش ابعاد مسئله و بهبود محاسبه استفاده می‌شود.

²⁷Rectangular kernel

۳-۲-۲ پیشینه پژوهش در ماشین بردار پشتیبان دو قلو

روش TSVM نسبت به روش SVM دو نقطه قوت مهم دارد که عبارتند از:

۱. حساسیت کمتری به مجموعه داده‌های نامتوزان دارد و دقت آن روی این گونه داده‌ها بیشتر است.

۲. دو مسئله دوگان کوچکتر برای بدست آوردن مدل خروجی حل می‌شود. به عبارتی هر مسئله دوگان فقط شامل نمونه‌های یک کلاس است. در حالی که در SVM تمام نمونه‌ها در مسئله دوگان نقش دارند.

با این حال ماشین بردار پشتیبان دو قلو نقاط ضعفی نیز دارد. در دهه اخیر، دسته‌بندهایی مبتنی بر TSVM ارائه شده است که نقاط ضعف TSVM را حل می‌کند [۲۵، ۲۶، ۲۷]. در ادامه برخی از گسترش‌های مهم روش TSVM در زیربخش‌های جداگانه توضیح داده شده است. در آخر سایر روش‌های مبتنی بر TSVM در یک زیربخش به طور خلاصه معرفی شده است.

۱-۳-۲-۲ ماشین بردار پشتیبان دو قلو کمترین مربعات

روش TSVM اصلی دو مسئله دوگان برای ایجاد مدل خروجی حل می‌کند. بطوریکه سرعت آموزش و ایجاد مدل در TSVM برای مجموعه داده‌های بزرگ به طور قابل توجهی کاهش می‌یابد. به منظور افزایش سرعت آموزش، ماشین بردار پشتیبان دو قلو کمترین مربعات (LS-TSVM²⁸) در سال ۲۰۰۹ ارائه گردید [۱۲]. در این روش، برای بدست آوردن دو ابرصفحه غیر موازی دو مسئله بهینه‌سازی به صورت زیر تعریف می‌شود.

$$\begin{aligned} \underset{w^{(1)}, b^{(1)}, y}{\text{Min}} \quad & \frac{1}{2} \|Aw^{(1)} + e_1 b^{(1)}\|^2 + \frac{C_1}{2} e_1^T y \\ \text{s.t.} \quad & -(Bw^{(1)} + e_2 b^{(1)}) + y = e_2 \end{aligned} \quad (46-2)$$

$$\begin{aligned} \underset{w^{(2)}, b^{(2)}, y}{\text{Min}} \quad & \frac{1}{2} \|Bw^{(2)} + e_2 b^{(2)}\|^2 + \frac{C_2}{2} e_2^T y \\ \text{s.t.} \quad & (Aw^{(2)} + e_1 b^{(2)}) + y = e_1 \end{aligned} \quad (47-2)$$

مسائل اصلی ۴۶-۲ و ۴۷-۲، یک تغییر مهم نسبت به مسائل اصلی در TSVM دارد. قید مسئله بهینه‌سازی به یک معادله تبدیل شده است. به عبارت دیگر، ابرصفحه غیرموازی باید به فاصله ۱ از کلاس مقابل دور شود. با جایگذاری قید درتابع هدف و گرفتن مشتق از $w^{(1)}$ و $b^{(1)}$ ، راه حل بدست

²⁸Least Squares Twin Support Vector Machine

می‌آید. در آخر مدل خروجی با حل کردن دو دستگاه معادلات خطی زیر ایجاد می‌شود.

$$\begin{bmatrix} w^{(1)} \\ b^{(1)} \end{bmatrix} = -\left(F^T F + \frac{1}{C_1} E^T E\right)^{-1} F^T e \quad (48-2)$$

$$\begin{bmatrix} w^{(2)} \\ b^{(2)} \end{bmatrix} = \left(E^T E + \frac{1}{C_2} F^T F\right)^{-1} E^T e \quad (49-2)$$

ماتریس E و F به صورت $F = [B \ e]$ و $E = [A \ e]$ تعریف می‌شود. روش LS-TSVM برای مسائل غیر خطی نیز استفاده می‌شود. همانند TSVM، نمونه‌ها با استفاده ازتابع هسته به فضای ویژگی با ابعاد بالا نگاشت می‌شود. جزئیات نسخه غیر خطی این روش در مقاله اصلی [۱۲] ذکر شده است. مزیت اصلی روش LS-TSVM، سرعت بسیار زیاد یادگیری و ایجاد مدل است. زیرا این روش به الگوریتم‌های حل مسائل بهینه‌سازی نیازی ندارد.

۲-۳-۲-۲ ماشین بردار پشتیبان دو قلو مبتنی بر مرز

روش TSVM ریسک تجربی را در مسئله بهینه‌سازی خود کمینه می‌کند. به عبارت دیگر، خط روى نمونه‌های آموزشی کمینه می‌شود. این موضوع باعث به وجود آمدن پدیده برآزش بیش از حد می‌گردد. در سال ۲۰۱۱، شانو و همکاران، ماشین بردار پشتیبان دو قلو مبتنی بر مرز (TBSVM) را ارائه کردند [۱۳]. روش TBSVM ریسک ساختاری را کمینه می‌کند و مانند SVM، حاشیه را بیشینه می‌کند. در این روش دو مسئله بهینه‌سازی اصلی به صورت زیر تعریف می‌شود.

$$\min_{w_1, b_1} \frac{1}{2} C_1 (\|w_1\|^2 + b_1^2) + \frac{1}{2} \|Aw_1 + e_1 b_1\|^2 + C_1 e_1^T y_1 \quad (50-2)$$

$$\text{s.t. } -(Bw_1 + e_1 b_1) + y_1 \geq e_1, y_1 \geq 0$$

$$\min_{w_2, b_2} \frac{1}{2} C_2 (\|w_2\|^2 + b_2^2) + \frac{1}{2} \|Bw_2 + e_2 b_2\|^2 + C_2 e_2^T y_2 \quad (51-2)$$

$$\text{s.t. } (Aw_2 + e_2 b_2) + y_2 \geq e_2, y_2 \geq 0$$

در رابطه ۵۰-۲ و ۵۱-۲، C_1, C_2, C_3 و C_4 پارامترهای مثبت هستند. این دو مسئله بهینه‌سازی اصلی شبیه به مسائل TSVM است. با این تفاوت که جمله رگولاریسون $(\frac{1}{2} C_1 (\|w_1\|^2 + b_1^2) + \frac{1}{2} \|Aw_1 + e_1 b_1\|^2 + C_1 e_1^T y_1)$ به مسئله اضافه شده است. بخارط این جمله، ریسک ساختاری در مسئله بهینه‌سازی ۵۰-۲ کمینه می‌شود. با استفاده ازتابع لاغرانژ و شرایط KKT، مسئله دوگان ۵۰-۲ و ۵۱-۲ در زیر تعریف شده است.

$$\min_{\alpha} \frac{1}{2} \alpha^T G (H^T H + C_1 I)^{-1} G^T \alpha - e_1^T \alpha \quad (52-2)$$

$$\text{s.t. } e_1 \leq \alpha \leq C_1 e_1$$

$$\begin{aligned} \min_{\beta} & \quad \frac{1}{2} \beta^T H (G^T G + C_4 I)^{-1} H^T \beta - e_1^T \beta \\ \text{s.t.} & \quad e_1 \leq \beta \leq C_2 e_1 \end{aligned} \quad (53-2)$$

در روابط ۵۲-۲ و ۵۳-۲، پارامتر C_4 و C_2 تعادل بین جمله رگولاریسون و ریسک تجربی است. نتایج در مقاله اصلی [۱۳] نشان می‌دهد که تنظیم کردن این پارامتر می‌تواند دقت دسته‌بندی را افزایش دهد. بعد از بدست آوردن ضرایب لاگرانژ، دو ابرصفحه غیر موازی از طریق روابط زیر بدست می‌آید.

$$\begin{bmatrix} w^{(1)} \\ b^{(1)} \end{bmatrix} = -(H^T H + C_4 I)^{-1} G^T \alpha \quad (54-2)$$

$$\begin{bmatrix} w^{(2)} \\ b^{(2)} \end{bmatrix} = (G^T G + C_4 I)^{-1} H^T \beta \quad (55-2)$$

نسخه غیر خطی روش TBSVM در مقاله اصلی توضیح داده شده است [۱۳].

۳-۳-۲-۲ ماشین بردار پشتیبان دو قلو وزن دار با اطلاعات محلی

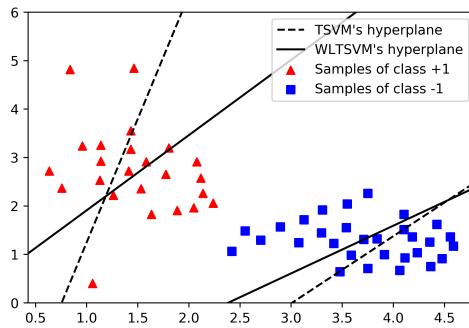
یکی از مشکلات روش TSVM اصلی این است که در ساخت مدل به تمام نمونه‌ها اهمیت یکسانی می‌دهد. بطوریکه نمونه‌های پرت و نویزی باعث کاهش دقت مدل می‌شوند. در سال ۲۰۱۲، ماشین بردار پشتیبان دو قلو وزن دار با اطلاعات محلی (WLTSVM) ارائه شد [۱۴]. این روش با ساخت گراف درون کلاسی^{۲۹} W_s به نمونه‌های هر کلاس بر اساس تعداد همسایه‌هایش وزن می‌دهد. همچنین نمونه‌های حاشیه‌ای^{۳۰} نیز توسط گراف برون کلاسی^{۳۱} W_d مشخص می‌شود. ایده اصلی روش WLTSVM این است که ابرصفحه غیر موازی باید به نمونه‌های با وزن بیشتر نزدیک شود و از نمونه‌های حاشیه‌ای حداقل فاصله را داشته باشد. ایده اصلی روش WLTSVM این است که ابرصفحه غیر موازی باید به نمونه‌های با وزن بیشتر نزدیک شود و از نمونه‌های حاشیه‌ای حداقل فاصله را داشته باشد. شکل ۸-۲ مقایسه هندسی روش WLTSVM با روش TSVM اصلی را نشان می‌دهد.

همانطور که شکل ۸-۲ در نشان داده شده است، ابرصفحه غیرموازی در روش WLTSVM به نمونه‌های پرتراکم نزدیک‌تر و از نمونه‌های پرت فاصله بیشتری دارد. در این روش دو مسئله بهینه‌سازی اصلی به

²⁹Intra-class

³⁰Margin point

³¹Inter-class



شکل ۲-۸: مقایسه هندسی روش WLTSVM با روش TSVM

صورت زیر تعریف می شود.

$$\begin{aligned} \min_{w_1, b_1} & \frac{1}{2} \sum_{i=1}^{m_1} \sum_{j=1}^{m_1} W_{s,ij}^{(1)} (w_1^T x_j^{(1)} + b_1)^2 + c \sum_{j=1}^{m_1} \xi_j \\ \text{s.t.} & -f_j^{(1)} (w_1^T x_j^{(1)} + b_1) + \xi_j \geq f_j^{(1)} \end{aligned} \quad (56-2)$$

$$\xi_j \geq 0, \quad j = 1, \dots, m_1$$

$$\begin{aligned} \min_{w_2, b_2} & \frac{1}{2} \sum_{i=1}^{m_2} \sum_{j=1}^{m_2} W_{s,ij}^{(2)} (w_2^T x_j^{(2)} + b_2)^2 + c \sum_{j=1}^{m_2} \eta_j \\ \text{s.t.} & f_j^{(2)} (w_2^T x_j^{(2)} + b_2) + \eta_j \geq f_j^{(2)} \end{aligned} \quad (57-2)$$

$$\eta_j \geq 0, \quad j = 1, \dots, m_2$$

در روابط ۵۶-۲ و ۵۷-۲، c پارامتر خطای ξ و η متغیر لغزش، $W_{s,ij}^{(1)}$ و $W_{s,ij}^{(2)}$ به ترتیب نشان دهنده گراف درون کلاسی کلاس مثبت و منفی است. همچنین $f_j^{(1)}$ و $f_j^{(2)}$ به ترتیب بیانگر نمونه های حاشیه ای کلاس مثبت و منفی است. در اینجا نکته حائز اهمیت این است که در قید مسئله بهینه سازی فقط نمونه های حاشیه ای کلاس مقابل در نظر گرفته می شود. در حالی که در روش TSVM اصلی، تمام نمونه های کلاس مقابل در قید مسئله بهینه سازی لحاظ شده است. در نتیجه با در نظر گرفتن نمونه های حاشیه ای، مرتبه زمانی روش WLTSVM نسبت به TSVM کاهش می یابد.

با استفاده ازتابع لاگرانژ و شرایط KKT حالت دوگان مسئله اصلی ۵۶-۲ و ۵۷-۲ در زیر تعریف شده است.

$$\begin{aligned} \min_{\alpha} & \frac{1}{2} \alpha^T (F^T G) (H^T D H)^{-1} (G^T F) \alpha - e_2^T F \alpha \\ \text{s.t.} & 0 \leq \alpha \leq ce_2 \end{aligned} \quad (58-2)$$

$$\min_{\beta} \quad \frac{1}{2} \beta^T (P^T H) (G^T Q G)^{-1} (H^T P) \beta - e_1^T P \beta \quad (59-2)$$

$$\text{s.t.} \quad e_1 \leq \beta \leq c e_1$$

در رابطه ۵۸-۲ و ۵۹-۲ نشان $Q = diag(d_1^{(2)}, d_2^{(2)}, \dots, d_{m_2}^{(2)})$ و $D = diag(d_1^{(1)}, d_2^{(1)}, \dots, d_{m_1}^{(1)})$ دهنده ماتریس وزن کلاس مثبت و منفی است. همچنین $F = P = diag(f_1^{(1)}, f_2^{(1)}, \dots, f_{m_1}^{(1)})$ دهنده نمونه‌های حاشیه‌ای کلاس مثبت و منفی است. بعد از حل $diag(f_1^{(2)}, f_2^{(2)}, \dots, f_{m_2}^{(2)})$ کردن مسئله بهینه‌سازی دوگان، مدل خروجی با حل کردن روابط زیر بدست می‌آید.

$$\begin{bmatrix} w_1 \\ b_1 \end{bmatrix} = -(H^T D H)^{-1} G^T F \alpha \quad (60-2)$$

$$\begin{bmatrix} w_2 \\ b_2 \end{bmatrix} = (G^T Q G)^{-1} H^T P \beta \quad (61-2)$$

نسخه غیر خطی روش WLTSVM در مقاله اصلی شرح داده شده است [۱۴]. روش WLTSVM نسبت به TSVM اصلی برتری‌های نظری دقت دسته‌بندی بیشتر و مرتبه‌زمانی بهتر را دارد. با این حال روش WLTSVM دارای نقاط ضعفی است که عبارتند از:

۱. روش WLTSVM به هر نمونه بر اساس تعداد نزدیک‌ترین همسایه‌هایش وزن نسبت می‌دهد. به عنوان مثال، وزن نمونه‌های کلاس مثبت از طریق رابطه زیر محاسبه می‌شود.

$$d_j^{(1)} = \sum_{i=1}^{m_1} W_{s,ij}, \quad j = 1, 2, \dots, m_1 \quad (62-2)$$

در رابطه ۶۲-۲، متغیر $d_j^{(1)}$ نشان دهنده وزن نمونه x_j است و $W_{s,ij}$ مقدار ۰ یا ۱ دارد. در حالی که می‌توان به همسایه‌های یک نمونه مقداری بین ۰ تا ۱ براساس فاصله شان نسبت داد.

۲. این روش نیز مانند TSVM ریسک تجربی را در مسائل اصلی ۵۶-۲ و ۵۷-۲ کمینه می‌کند. بطوریکه برای جلوگیری از شرایط ماتریس منفرد، معکوس ماتریس‌های $(H^T D H)^{-1}$ و $(G^T Q G)^{-1}$ به ترتیب با $(H^T D H + \varepsilon I)^{-1}$ و $(G^T Q G + \varepsilon I)^{-1}$ جایگزین می‌شود. بنابراین تنها راه حل تقریبی مسائل ۵۸-۲ و ۵۹-۲ بدست می‌آید.

۳. با وجود اینکه روش WLTSVM مرتبه زمانی را با لحاظ کردن نمونه‌های حاشیه‌ای در قید مسئله بهینه‌سازی کاهش می‌دهد. در این روش، k تا از نزدیک‌ترین همسایه‌های تمام نمونه‌های آموزشی باید محاسبه شود. بنابراین پیچیدگی محاسباتی کلی این روش برابر با $O(2m^3 + m^2 \log m)$ است.

با فرض اینکه $m_1 = m_2$ و $m \ll m_1, m_2$. روش‌های جدید و سریع KNN برای حل کردن این مشکل می‌تواند استفاده شود.

۴-۳-۲-۲ سایر گسترش‌های ماشین بردار پشتیبان دو قلو

در این زیربخش، سایر روش‌های مبتنی بر TSVM به طور خلاصه معرفی می‌شود. هر کدام از این از گسترش‌ها، یک نقطه ضعف روش TSVM را حل کرده‌اند [۲۵، ۲۶، ۲۷]. در سال ۲۰۱۳، ماشین بردار پشتیبان دو قلو ساختاری (STSVM³²) ارائه گردید [۲۸]. این روش یادگیری اطلاعات مفید ساختاری درون هر کلاس و توزیع نمونه‌ها را از طریق خوشبندی سلسله مراتبی در مدل خروجی لحاظ می‌کند. در سال ۲۰۱۴، ماشین بردار پشتیبان دو قلو مبتنی بر انرژی (ELS-TSVM³³) ارائه شد [۲۹]. در این روش، پارامتر انرژی برای دو ابرصفحه غیرموازی تعریف شده است. بطوریکه مقدار این پارامتر با توجه به دانش قبلی تعیین می‌شود تا اثر نمونه‌های نویزی و پرت کاهش یابد. در سال ۲۰۱۵، ماشین بردار پشتیبان دو قلو ساختاری با رویکرد گراف نزدیک‌ترین همسایه (KNN-STSVM³⁴) معرفی شد [۳۰]. در این روش، علاوه بر در نظر گرفتن اطلاعات ساختاری نمونه‌ها، با استفاده از گراف نزدیک‌ترین همسایه به نمونه‌ها وزن داده می‌شود. در نتیجه، دقت مدل خروجی افزایش می‌یابد.

در سال ۲۰۱۶، ماشین بردار پشتیبان دو قلو چند کلاسه با رویکرد مبتنی بر نزدیک‌ترین همسایه (KWM-TSVM³⁵) معرفی شد [۳۱]. این روش با استفاده از گراف نزدیک‌ترین همسایه، اطلاعات درون کلاسی و برون کلاسی را در تابع هدف مسئله بهینه‌سازی لحاظ می‌کند. در نتیجه پیچیدگی زمانی و دقیق مدل بهبود یافته است. در سال ۲۰۱۸، یک روش امن برای کاهش تعداد نمونه‌ها^{۳۶} در روش KWM-TSVM ارائه گردید [۳۲]. این روش بخش زیادی از نمونه‌های دو کلاس را قبل از آموزش مدل حذف می‌کند. بنابراین پیچیدگی محاسباتی به طور قابل توجه‌ای کاهش یافته است. گسترش‌های مبتنی بر TSVM که در بخش ۴-۲-۲ معرفی شده‌اند، در جدول ۱-۲ ایده اصلی، نقاط ضعف و محدودیت‌های آن‌ها به صورت خلاصه مرور شده است.

³²Structural Twin Support Vector Machine

³³Energy-based Model of Least Squares Twin Support Vector Machine

³⁴K-nearest neighbor based structural twin support vector machine

³⁵K-nearest neighbor-based weighted multi-class twin support vector machine

³⁶Safe instance reduction rule

جدول ۲-۱: مرور کلی گسترش‌های مبتنی بر روش TSVM

نقطه ضعف و محدودیت‌ها	ایده اصلی	سال معرفی	روش پادگیری
	این دسته‌بند به تمام نمونه‌های آموزشی اهمیت پیکاری	۲۰۰۹	حل دستگاه معادلات خطی به جای مسئله دوگان که باعث افزایش چشم گیر سرعت پادگیری شده است.
	می‌دهد. از این رو نمونه‌های پرت و نویزی، دقت مدل خروجی را کاهش می‌دهد،	۲۰۱۱	ریسک ساختاری را در مسئله بهینه‌سازی خود کمینه می‌کند.
	بنابراین جستجوی مقادیر بهینه پارامترها با محاسباتی بالای دارد.	۲۰۱۲	با استفاده از گراف نزدیک‌ترین همسایه به هر نمونه وزن همچینی از شرایط ماتریس منفرد جلوگیری می‌کند.
	بنابراین همسایه بهینه پارامترها با محاسباتی بالای دارد.	۲۰۱۳	از گراف نزدیک‌ترین همسایه به هر نمونه وزن وزدنی به نمونه‌ها تنها براساس شمارش تعداد همسایه‌های نزدیک صورت می‌گیرد.
	بنابراین همسایه سرعت پادگیری این دسته‌بند را براي نزدیک‌ترین همسایه دادهای بزرگ به طور قابل توجهی کاهش می‌دهد.	۲۰۱۴	اطلاعات ساختاری را از طریق خوشبتدی مسئله مرانی در مدل خروجی لحاظ می‌کند.
	مجموعه دادهای بزرگ به گرفته نمی‌شود. همچنین بدست آوردن خوشبندی مجموعه دادهای اهمیت نمونه های درون هر خوشبندی در نظر گرفته نمی‌شود.	۲۰۱۵	پیامتر انحری برای دو ابرصخه معرفی شده است که اثر نمونه‌های نویزی و پرت را در مدل خروجی کمتر می‌کند.
	بنابرگ، سرعت پادگیری این مدل را کاهش می‌دهد.	۲۰۱۶	اطلاعات ساختاری با گراف نزدیک‌ترین همسایه برای بهبود دقت مدل خروجی ترکیب شده است.
	بنابراین همانند روش TSVM، ریسک تجربی را کمینه می‌کند.	۲۰۱۷	روش TSVM چند کلاسه را با استفاده از گراف نزدیک‌ترین نزدیک‌ترین همسایه و خوشبندی برای تمام نمونه‌های آموزشی پایل بدلست آید.
	بنابراین همسایه از نظر دقت و پیچیدگی زمانی بهبود داده است.	۲۰۱۸	روش TSVM چند کلاسه را با استفاده از گراف نزدیک‌ترین داده‌های بزرگ بسیار زیاد است.
	در حافظه ذخیره گردید.	۲۰۱۹	معکوس ماتریس در این دسته‌بند اجتناب ناپذیر است. بنابراین سرعت پادگیری برای مجموعه دادهای بزرگ به طور چشم گیر کاهش می‌پاید.
	بنابراین همسایه بهینه پارامترها با محاسباتی بالای دارد.	۲۰۲۰	امن حذف می‌کند. بطوریکه پیچیدگی زمانی روش TSVM چند کلاسه بهبود یافته است.
	بنابراین همسایه بهینه پارامترها با محاسباتی بالای دارد.	۲۰۲۱	ذخیره کردن گراف نزدیک‌ترین همسایه بسیار زیاد است.

فصل ۳

ماشین بردار پیشتبان دو قلو کمترین مربعات مبتنی بر نزدیک ترین همسایه

۱-۳ مقدمه

نقطه ضعف بزرگ روش TSVM و LS-TSVM این است که این روش‌ها اطلاعات شباخت بین نمونه‌های آموزشی را در نظر نمی‌گیرند. به عبارت دیگر، به تمام نمونه‌های آموزشی اهمیت یکسانی داده می‌شود. بطوریکه نمونه‌های نویزی و پرت دقت مدل خروجی را روی داده‌های جدید کاهش می‌دهد. روش WLTSVM این نقطه ضعف مهم را حل کرده است. این روش با ساخت گراف نزدیک‌ترین همسایه، اطلاعات درون و برون کلاسی را در تابع هدف مسئله بهینه‌سازی لحاظ کرده است. بطوریکه به هر یک از نمونه‌های آموزشی وزن نسبت می‌دهد و همچنین نمونه‌های حاشیه‌ای هر کلاس را استخراج می‌کند. روش WLTSVM مانند TSVM اصلی، دو مسئله بهینه‌سازی دوگان از نوع برنامه‌ریزی درجه دو حل می‌کند. بطوریکه آموزش روش WLTSVM بر روی مجموعه داده‌های بزرگ کند و زمان برخواهد بود. در این فصل، با گرفتن ایده از روش‌های LS-TSVM و WLTSVM، روش جدید ماشین بردار پیشتبان دو قلو کمترین مربعات مبتنی بر رویکرد نزدیک‌ترین همسایه (KNN-LSTSVM) ارائه می‌شود [۱۵]. روش پیشنهادی دارای مزایای زیر است:

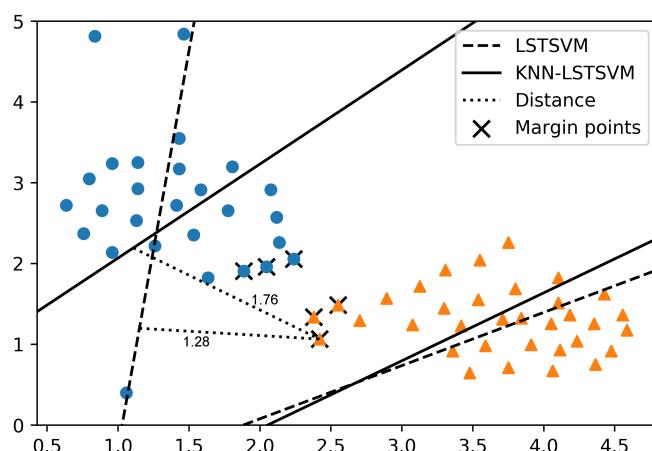
- روش پیشنهادی (KNN-LSTSVM)، مشابه روش WLTSVM به طور کامل از اطلاعات شباخت بین نمونه‌ها استفاده می‌کند. بطوریکه با ساخت گراف نزدیک‌ترین همسایه اطلاعات درون و برون کلاسی را در مسئله بهینه‌سازی لحاظ می‌کند. به عبارت دیگر، به هر نمونه آموزشی بر اساس شمارش تعداد نزدیک‌ترین همسایه‌هایش وزن داده می‌شود. همچنین نمونه‌های حاشیه‌ای هر کلاس نیز مشخص می‌گردد.

- روش پیشنهادی مشابه روش LS-TSVM، دو دستگاه معادلات خطی برای بدست آوردن مدل خروجی حل می‌کند. این مزیت روش پیشنهادی را به یک الگوریتم ساده با پیچیدگی محاسباتی کمتر از WLTSVM تبدیل می‌کند. به طور کلی، روش KNN-LSTSVM نیازی به الگوریتم‌های بهینه‌سازی برای حل مسائل دوگان ندارد.
- روش پیشنهادی برخلاف روش LS-TSVM نسبت به نمونه‌های پرت حساسیت کمتری دارد. زیرا با استفاده از گراف نزدیک‌ترین همسایه به نمونه‌های پرت و نویزی ورن کمتری نسبت داده می‌شود. بنابراین مدل خروجی دقت بهتری خواهد داشت.

در ادامه این فصل، ابتدا نحوه ساخت ماتریس وزن‌ها از طریق گراف k -نزدیک‌ترین همسایه شرح داده می‌شود. سپس نسخه خطی و غیر خطی روش KNN-LSTSVM توضیح داده شده است.

۲-۳ ساخت ماتریس وزن‌ها

ایده اصلی روش پیشنهادی (KNN-LSTSVM) و WLTSVM این است که به نمونه‌های با تراکم بیشتر وزن بیشتری بدهد و از کلاس مقابل نمونه‌های حاشیه‌ای را مشخص کند. به عبارت دیگر، ابرصفحه غیرموافق در روش KNN-LSTSVM به نمونه‌های پرترکم نزدیک‌تر است و از نمونه‌های حاشیه‌ای کلاس مقابل حداقل فاصله را می‌گیرد. شکل ۱-۳ تفسیر هندسی روش LS-TSVM و KNN-LSTSVM را نشان می‌دهد.



شکل ۱-۳: تفسیر هندسی روش LS-TSVM و KNN-LSTSVM

همانطور که در شکل ۱-۳ مشخص شده است، ابرصفحه غیرموازی در LS-TSVM به نمونه‌های پرت (کلاس دایره) بسیار نزدیک است. در حالی که روش پیشنهادی (KNN-LSTSVM) از نمونه‌های پرت فاصله قابل توجه‌ای دارد و به نواحی پرتراکم نزدیک‌تر است. همچنین فاصله عمودی یک نمونه حاشیه‌ای از ابرصفحه هر دو روش در شکل ۱-۳ محاسبه شده است. روش پیشنهادی از نمونه‌های حاشیه‌ای فاصله بیشتری دارد.

ابتدا گراف k نزدیک‌ترین همسایه جهت بدست آوردن ماتریس وزن‌ها به صورت زیر تعریف می‌شود.

$$W_{ij} = \begin{cases} 1, & \text{if } x_i \in N_{ea}(x_j) \text{ or } x_j \in N_{ea}(x_i), \\ 0, & \text{otherwise.} \end{cases} \quad (1-3)$$

در رابطه ۱-۳، مجموعه شامل k نزدیک‌ترین همسایه نمونه x_i است که در رابطه زیر تعریف شده است.

$$N(x_j) = \{x_j^1, x_j^2, \dots, x_j^k\} \quad (2-3)$$

گراف W به تنها یی نمی‌تواند تفکیک پذیری در داده‌ها را پیدا کند. در عوض یک گراف درون کلاسی $W_{w,ij}$ و برون کلاسی $W_{b,ij}$ ایجاد می‌شود تا به ترتیب فشردگی درون کلاسی^۱ و جداپذیری برون کلاسی^۲ مشخص شود. ماتریس وزن W_w و W_b به ترتیب در روابط ۳-۳ و ۴-۳ تعریف شده است.

$$W_{w,ij} = \begin{cases} 1, & \text{if } x_i \in N_w(x_j) \text{ or } x_j \in N_w(x_i) \\ 0, & \text{otherwise.} \end{cases} \quad (3-3)$$

$$W_{b,ij} = \begin{cases} 1, & \text{if } x_i \in N_b(x_j) \text{ or } x_j \in N_b(x_i) \\ 0, & \text{otherwise.} \end{cases} \quad (4-3)$$

در رابطه ۳-۳ و ۴-۳، مجموعه $N_w(x_j)$ نشان‌دهنده k نزدیک‌ترین همسایه نمونه x_j از کلاس خودش و $N_b(x_j)$ مجموعه شامل k نزدیک‌ترین همسایه نمونه x_j از کلاس مقابل است. دو مجموعه $N_w(x_j)$ و $N_b(x_j)$ به ترتیب در روابط ۵-۳ و ۶-۳ تعریف شده‌اند.

$$N_w(x_i) = \{x_i^j \mid l(x_i^j) = l(x_i), 1 \leq j \leq k\} \quad (5-3)$$

$$N_b(x_i) = \{x_i^j \mid l(x_i^j) \neq l(x_i), 1 \leq j \leq k\} \quad (6-3)$$

¹Intra-class compactness

²Inter-class separability

در رابطه ۳-۵ و ۳-۶، $N_w(x_i) \cap N_b(x_i) = \emptyset$ نشان دهنده برچسب نمونه x_i است. بدیهی است که $N_w(x_i) \cup N_b(x_i) = N(x_i)$. فاصله بین هر کدام از نمونه های آموزشی توسط قاعده اقلیدس محاسبه می شود. به منظور پیدا کردن نمونه های حاشیه ای کلاس منفی، ماتریس وزن W_b به صورت زیر باز تعریف می شود.

$$f_j = \begin{cases} 1, & \text{if } \exists i, W_{b,ij} \neq 0 \\ 0, & \text{otherwise.} \end{cases} \quad (7-3)$$

وزن هر کدام از نمونه های کلاس مثبت به صورت زیر محاسبه می شود.

$$d_j = \sum_{i=1}^{m_1} W_{w, ij}, \quad j = 1, 2, \dots, m_1 \quad (8-3)$$

در رابطه ۸-۳، d_j نشان دهنده وزن نمونه x_j است. در اینجا تعداد همسایه ها با برچسب یکسان، وزن یک نمونه را مشخص می کند.

۳-۳ نسخه خطی

روش پیشنهادی مشابه روش WLTSVM، دو ابر صفحه غیر موازی ایجاد می کند. بطوریکه هر کدام از این ابر صفحه ها به نمونه های پر تراکم نزدیک تر است و از نمونه های حاشیه ای کلاس مقابل حداقل فاصله را دارد. با این حال روش پیشنهادی (KNN-LSTSVM)، مسائل اصلی روش WLTSVM را تغییر می دهد. به این صورت که نامعادله در قید مسئله بهینه سازی به قید مساوی تغییر می یابد و همچنین متغیر لغزش به توان دو رسیده است.

راه حل دو مسئله اصلی تغییر یافته نیاز به حل کردن دو دستگاه معادلات خطی دارد. در حالی که در روش WLTSVM، دو مسئله دوگان باید حل شود. مسئله اصلی تغییر یافته کلاس مثبت به صورت زیر تعریف می شود.

$$\begin{aligned} \min_{w^{(1)}, b^{(1)}} \quad & \frac{1}{2} (Aw^{(1)} + eb^{(1)})^T D (Aw^{(1)} + eb^{(1)}) + \frac{C}{2} y^T y \\ \text{s.t.} \quad & -F(Bw^{(1)} + eb^{(1)}) + y = Fe \end{aligned} \quad (9-3)$$

در رابطه ۹-۳، $D = diag(d_1, \dots, d_{m_1})$ نشان دهنده ماتریس وزن کلاس مثبت و $F = diag(f_1, \dots, f_{m_1})$ نشان دهنده نمونه های حاشیه ای کلاس منفی است. مقدار f_j برابر با 0 یا 1 است و d_i بزرگتر یا مساوی با صفر است ($d_i \geq 0$).

مزیت مهم مسئله اصلی ۹-۳ این است که می‌توان قید مساوی را در آن در جایگذاری کرد.

بنابراین تابع لاگرانژ مسئله اصلی ۹-۳ به صورت تعریف می‌شود.

$$\min_{w^{(1)}, b^{(1)}} L = \frac{1}{2} \|D(Aw^{(1)} + eb^{(1)})\|^2 + \frac{C}{2} \|F(Bw^{(1)} + eb^{(1)}) + Fe\|^2 \quad (10-3)$$

با گرفتن مشتق‌گیری جزئی از رابطه ۱۰-۳ نسبت به $w^{(1)}$ و $b^{(1)}$ خواهیم داشت:

$$\frac{\partial L}{\partial w^{(1)}} = A^T D(Aw^{(1)} + eb^{(1)}) + CB^T F(Bw^{(1)} + eb^{(1)} + Fe) = \bullet e \quad (11-3)$$

$$\frac{\partial L}{\partial b^{(1)}} = e^T D(Aw^{(1)} + eb^{(1)}) + Ce^T F(Bw^{(1)} + eb^{(1)} + Fe) = \bullet \quad (12-3)$$

در ادامه با ترکیب کردن روابط ۱۱-۳ و ۱۲-۳ دستگاه معادلات خطی زیر بدست می‌آید.

$$\begin{bmatrix} B^T FB & B^T Fe \\ e^T FB & e^T Fe \end{bmatrix} \begin{bmatrix} w^{(1)} \\ b^{(1)} \end{bmatrix} + \frac{1}{C} \begin{bmatrix} A^T DA & A^T De \\ e^T DA & e^T De \end{bmatrix} \begin{bmatrix} w^{(1)} \\ b^{(1)} \end{bmatrix} + \begin{bmatrix} B^T Fe \\ e^T Fe \end{bmatrix} = \bullet e. \quad (13-3)$$

$$\begin{bmatrix} w^{(1)} \\ b^{(1)} \end{bmatrix} = \begin{bmatrix} B^T FB + \frac{1}{C} A^T DA & B^T Fe + \frac{1}{C} A^T De \\ e^T FB + \frac{1}{C} e^T DA & e^T Fe + \frac{1}{C} e^T De \end{bmatrix}^{-1} \times \begin{bmatrix} -B^T Fe \\ -e^T Fe \end{bmatrix} \quad (14-3)$$

$$\begin{bmatrix} w^{(1)} \\ b^{(1)} \end{bmatrix} = \begin{bmatrix} B^T \\ e^T \end{bmatrix} F \begin{bmatrix} B & e \end{bmatrix} + \frac{1}{C} \begin{bmatrix} A^T \\ e^T \end{bmatrix} D \begin{bmatrix} A & e \end{bmatrix}^{-1} \times \begin{bmatrix} -B^T \\ -e^T \end{bmatrix} Fe \quad (15-3)$$

با تعریف کردن ماتریس $H = [A \ e]$ و $G = [B \ e]$ راه حل مسئله بهینه‌سازی ۹-۳

به صورت زیر تعریف می‌شود.

$$\begin{bmatrix} w^{(1)} \\ b^{(1)} \end{bmatrix} = -(G^T FG + \frac{1}{C} H^T DH)^{-1} G^T Fe \quad (16-3)$$

مسئله اصلی تغییر یافته کلاس منفی به صورت زیر تعریف می‌شود.

$$\begin{array}{ll} \min_{w^{(1)}, b^{(1)}} & \frac{1}{2} (Bw^{(1)} + eb^{(1)})^T Q (Bw^{(1)} + eb^{(1)}) + \frac{C}{2} y^T y \\ \text{s.t.} & P(Aw^{(1)} + eb^{(1)}) + y = Pe \end{array} \quad (17-3)$$

در رابطه ۱۷-۳، $P = diag(p_1, \dots, p_{m_1})$ نشان‌دهنده ماتریس وزن کلاس منفی و $Q = diag(q_1, \dots, q_{m_2})$ نشان‌دهنده نمونه‌های حاشیه‌ای کلاس مثبت است. مانند ماتریس F ، مقدار p_j برابر \bullet یا 1 است.

راه حل مسئله اصلی ۱۷-۳ همانند کلاس مثبت با جایگذاری قید مساوی، گرفتن مشتق‌گیری جزئی

به صورت زیر تعریف می‌شود.

$$\begin{bmatrix} w^{(1)} \\ b^{(1)} \end{bmatrix} = (H^T PH + \frac{1}{C} G^T QG)^{-1} H^T Pe \quad (18-3)$$

راه حل‌های ۱۶-۳ و ۱۸-۳ شامل دو معکوس ماتریس با اندازه $(n+1) \times (n+1)$ است. بطوریکه n بسیار کوچکتر از تعداد نمونه‌های کلاس مثبت و منفی می‌باشد. بنابراین سرعت یادگیری نسخه خطی روش KNN-LSTSVM بسیار زیاد است.

تابع تصمیم نسخه خطی به صورت زیر تعریف می‌شود.

$$D(x_i) = \begin{cases} +1, & \text{if } |x^T w^{(1)} + b^{(1)}| < |x^T w^{(2)} + b^{(2)}| \\ -1, & \text{otherwise.} \end{cases} \quad (19-3)$$

مراحل ایجاد مدل خطی دسته‌بند KNN-LSTSVM در الگوریتم ۱-۳ خلاصه شده است.

الگوریتم ۱-۳: ایجاد مدل خطی دسته‌بند KNN-LSTSVM

ابتدا فرض می‌گیریم که نمونه‌های کلاس مثبت با ماتریس $A \in \mathbb{R}^{m_1 \times n}$ و نمونه‌های کلاس منفی با ماتریس $B \in \mathbb{R}^{m_2 \times n}$ نشان داده شده است.

۱. ابتدا پارامتر k برای ساخت گراف نزدیکترین همسایه تعیین می‌شود. سپس ماتریس وزن‌های W_b و W_w برای کلاس مثبت و منفی با استفاده از روابط ۳-۳ و ۴-۳ بدست می‌آید.

۲. ماتریس‌های قطری D, F, Q و P باید تعریف گردد. سپس ماتریس‌های H و G به صورت $G = [B \ e]$ و $H = [A \ e]$ تعریف می‌شود.

۳. مقدار پارامتر خطا C باید تعیین گردد. این پارامتر معمولاً براساس مجموعه داده صحت مشخص می‌شود.

۴. مختصات دو ابرصفحه غیرموازی از طریق روابط ۱۶-۳ و ۱۸-۳ بدست می‌آید.

۵. فاصله عمودی از $|x^T w^{(1)} + b^{(1)}|$ و $|x^T w^{(2)} + b^{(2)}|$ برای نمونه جدید $x \in \mathbb{R}^n$ محاسبه می‌شود.

۶. کلاس نمونه جدید از طریق تابع تصمیم ۱۹-۳ مشخص می‌گردد.

۴-۳ نسخه غیر خطی

به منظور حل کردن مسائل غیر خطی، نسخه خطی روش KNN-LSTSVM با در نظر گرفتن دو ابرسطح زیر می‌توان به نسخه غیر خطی گسترش داد.

$$K(x^T, C^T)u^{(1)} + b^{(1)} = 0 \text{ and } K(x^T, C^T)u^{(2)} + b^{(2)} = 0 \quad (20-3)$$

در رابطه ۲۰-۳، ماتریس برابر C با $C = [A \ B]^T$ است و K تابع هسته دلخواه می‌باشد. همانند نسخه خطی، مسائل بهینه‌سازی اصلی نسخه غیر خطی روش KNN-LSTSVM با مساوی قرار دادن قید برای کلاس مثبت و منفی به ترتیب در روابط ۲۱-۳ و ۲۲-۳ تعریف شده است.

$$\begin{aligned} \min_{u^{(1)}, b^{(1)}} \quad & \frac{1}{2} (K(A, C^T)u^{(1)} + eb^{(1)})^T D (K(A, C^T)u^{(1)} + eb^{(1)}) + \frac{C}{2} y^T y \\ \text{s.t.} \quad & -F(K(B, C^T)u^{(1)} + eb^{(1)}) + y = Fe \end{aligned} \quad (21-3)$$

$$\begin{aligned} \min_{u^{(2)}, b^{(2)}} \quad & \frac{1}{2} (K(B, C^T)u^{(2)} + eb^{(2)})^T Q (K(B, C^T)u^{(2)} + eb^{(2)}) + \frac{C}{2} y^T y \\ \text{s.t.} \quad & P(K(A, C^T)u^{(2)} + eb^{(2)}) + y = Pe \end{aligned} \quad (22-3)$$

در روابط ۲۱-۳ و ۲۲-۳، ماتریس $K(B, C^T)$ و $K(A, C^T)$ نشان‌دهنده ماتریس هسته به ترتیب با اندازه‌های $m_2 \times m$ و $m_1 \times m$ هستند ($m = m_1 + m_2$). مسائل بهینه‌سازی اصلی ۲۱-۳ و ۲۲-۳ با جایگذاری قید در تابع هدف به صورت زیر تعریف می‌شود.

$$\min_{u^{(1)}, b^{(1)}} \frac{1}{2} \|D(K(A, C^T)u^{(1)} + eb^{(1)})\|^2 + \frac{C}{2} \|F(K(B, C^T)u^{(1)} + eb^{(1)} + Fe)\|^2 \quad (23-3)$$

$$\min_{u^{(2)}, b^{(2)}} \frac{1}{2} \|Q(K(B, C^T)u^{(2)} + eb^{(2)})\|^2 + \frac{C}{2} \|P(K(A, C^T)u^{(2)} + eb^{(2)} + Pe)\|^2 \quad (24-3)$$

راه حل مسائل ۲۳-۳ و ۲۴-۳ به طرز مشابه نسخه خطی در زیر تعریف شده است.

$$\begin{bmatrix} u^{(1)} \\ b^{(1)} \end{bmatrix} = -(S^T FS + \frac{1}{C} R^T DR)^{-1} S^T Fe \quad (25-3)$$

$$\begin{bmatrix} u^{(2)} \\ b^{(2)} \end{bmatrix} = (R^T PR + \frac{1}{C} S^T QS)^{-1} R^T Pe \quad (26-3)$$

در رابطه ۲۵-۳ و ۲۶-۳، ماتریس S و R به صورت $S = [K(B, C^T) \quad e]$ و $R = [K(A, C^T) \quad e]$ تعریف می‌شود. کلاس یک نمونه جدید مشابه نسخه خطی تعیین می‌شود. بطوریکه فاصله عمودی از

دو ابرسطح محاسبه می‌گردد. تابع تصمیم در نسخه غیر خطی به صورت زیر تعریف شده است.

$$D(x) = \begin{cases} +1, & \text{if } |K(x, C^T)u^{(1)} + b^{(1)}| < |K(x, C^T)u^{(2)} + b^{(2)}| \\ -1, & \text{otherwise.} \end{cases} \quad (27-3)$$

لازم به ذکر است که راه حل نسخه غیر خطی روش KNN-LSTSVM شامل دو معکوس ماتریس با ابعاد $(1 \times (m+1) \times (m+1))$ است. بطوریکه m تعداد کل نمونه‌های آموزشی است. جهت کاهش پیچیدگی محاسباتی، می‌توان از فرمول^۳ SMW^۳ استفاده کرد [۳۳]. در این صورت راه حل‌های ۲۵-۳ و ۲۶-۳ را می‌توان با چهار عمل معکوس ماتریس با ابعاد کوچک‌تر از $(1 \times (m+1))$ محاسبه کرد. بنابراین راه حل‌های ۲۵-۳ و ۲۶-۳ به صورت زیر بازنویسی می‌شود.

$$\begin{bmatrix} u^{(1)} \\ b^{(1)} \end{bmatrix} = -(Y - YR^T D(CI + RYR^T D)^{-1} RY)S^T Fe \quad (28-3)$$

$$\begin{bmatrix} u^{(2)} \\ b^{(2)} \end{bmatrix} = (Z - ZS^T Q(CI + SZS^T Q)^{-1} SZ)R^T Pe \quad (29-3)$$

در رابطه ۲۸-۳ و ۲۹-۳، ماتریس‌های Y و Z به صورت $Y = (S^T FS)^{-1}$ و $Z = (R^T PR)^{-1}$ تعریف می‌شوند. با این حال ماتریس‌های $(S^T FS)$ و $(R^T PR)$ ممکن که دچار شرایط منفرد شوند. یک عدد ثابت بسیار کوچک εI ، $\varepsilon > 0$ به این ماتریس‌ها اضافه می‌شود تا از شرایط منفرد جلوگیری شود. بعد از اضافه شدن εI می‌توان از SMW برای پیدا کردن ماتریس‌های Y و Z به صورت زیر استفاده کرد.

$$Y = \frac{1}{\varepsilon}(I - S^T F(\varepsilon I + SS^T F)^{-1} S) \quad (30-3)$$

$$Z = \frac{1}{\varepsilon}(I - R^T P(\varepsilon I + RR^T P)^{-1} R) \quad (31-3)$$

بعد از استفاده کردن از فرمول SMW، راه حل نسخه غیر خطی شامل دو معکوس ماتریس به اندازه $(m_1 \times m_1)$ و دو معکوس ماتریس به اندازه $(m_2 \times m_2)$ می‌شود. مراحل ایجاد مدل غیر خطی دسته‌بند KNN-LSTSVM در الگوریتم ۲-۳ خلاصه شده است.

^۳Sherman-Morrison-Woodbury

الگوریتم ۲-۳: ایجاد مدل غیر خطی دسته‌بند KNN-LSTSVM

ابتدا فرض می‌گیریم که نمونه‌های کلاس مثبت با ماتریس $A \in \mathbb{R}^{m_1 \times n}$ و نمونه‌های کلاس منفی با ماتریس $B \in \mathbb{R}^{m_2 \times n}$ نشان داده شده است.

۱. ابتدا یکتابع هسته برای نگاشت نمونه‌های آموزشی به فضای ویژگی انتخاب می‌شود.

غالباً از تابع RBF در این مرحله استفاده می‌شود.

۲. ابتدا پارامتر k برای ساخت گراف نزدیک‌ترین همسایه تعیین می‌شود. سپس ماتریس وزن‌های W_b و W_w برای کلاس مثبت و منفی با استفاده از روابط ۳-۳ و ۴-۳ بدست می‌آید.

۳. ماتریس‌های قطری D ، F ، R و P باید تعریف گردد. سپس ماتریس‌های R و S به صورت

$$S = \begin{bmatrix} K(B, C^T) & e \end{bmatrix} \quad R = \begin{bmatrix} K(A, C^T) & e \end{bmatrix}$$

۴. مقدار پارامتر خطا C باید تعیین گردد. این پارامتر معمولاً براساس مجموعه داده صحت مشخص می‌شود.

۵. مختصات دو ابرسطح از طریق روابط ۲۸-۳ و ۲۹-۳ بدست می‌آید.

۶. فاصله عمودی از $|K(x, C^T)u^{(2)} + b^{(2)}|$ و $|K(x, C^T)u^{(1)} + b^{(1)}|$ برای نمونه جدید محاسبه می‌شود.

۷. کلاس نمونه جدید از طریق تابع تصمیم ۲۷-۳ مشخص می‌گردد.

۵-۳ تحلیل روش KNN-LSTSVM

در مقایسه با LSTSVM، روش پیشنهادی با استفاده از گراف نزدیک‌ترین همسایه اطلاعات درون کلاسی و بروون کلاسی را در مسئله بهینه‌سازی لحاظ می‌کند. بنابراین روش KNN-LSTSVM حساسیت کمتری نسبت به نمونه‌های نویزی و پرت دارد. همچنین روش پیشنهادی مانند LSTSVM دو دستگاه معادلات خطی حل می‌کند. در نتیجه سرعت یادگیری هر دو روش بسیار زیاد است.

دو روش KNN-LSTSVM و WLTSVM با ساخت گراف نزدیک‌ترین همسایه به نمونه‌های آموزشی

وزن می‌دهند و همچنین نمونه‌های حاشیه‌ای هر دو کلاس را مشخص می‌کنند. با این حال روش پیشنهادی دو دستگاه معادلات خطی حل می‌کند. در حالی که در WLTSVM دو مسئله دوگان حل می‌شود. بنابراین سرعت یادگیری روش پیشنهادی بسیار بیشتر از WLTSVM است. با وجود اینکه مدل خروجی در روش پیشنهادی دقیق‌تر نسبت به LSTSVM دارد، دو محدودیت مهم نیز دارد که عبارتند از:

۱. در راه‌های حل نسخه خطی و غیر خطی، عمل معکوس کردن ماتریس اجتناب ناپذیر است. از طرفی، مرتبه زمانی معکوس کردن ماتریس برابر با $O(m^3)$ است. بنابراین زمان محاسبه با افزایش ابعاد ماتریس، به طور قابل توجه‌ای بیشتر می‌شود.
۲. مصرف حافظه روش پیشنهادی برای مجموعه داده‌های بزرگ (بیش از ۵۰ هزار نمونه) بسیار زیاد است. زیرا دو گراف نزدیک‌ترین همسایه و ماتریس وزن‌ها برای ساخت مدل باید ذخیره گردد.

۳-۶ جمع‌بندی

در این فصل دسته‌بند KNN-LSTSVM ارائه شد. روش پیشنهادی همانند WLTSVM با ساخت گراف نزدیک‌ترین همسایه، به نمونه‌های آموزشی وزن می‌دهد و نمونه‌های حاشیه‌ای هر کلاس را مشخص می‌کند. ایده اصلی روش پیشنهادی این است که ابرصفحه غیر موازی به نمونه‌های پرتراکم کلاس خود نزدیک می‌شود و از نمونه‌های حاشیه‌ای کلاس مقابله حداکثر فاصله را می‌گیرد. با این حال دسته‌بند KNN-LSTSVM مزیت اصلی روش LSTSVM دارد. بطوریکه دو دستگاه معادلات خطی برای ساخت مدل خروجی حل می‌شود. همچنین نقطه ضعف روش LSTSVM یعنی حساسیت به نمونه‌های نویزی و پرت در روش پیشنهادی برطرف شده است. روش KNN-LSTSVM در بخش ۲-۵ به طور جامع ارزیابی شده است.

دسته‌بند KNN-LSTSVM در مجله زیر به چاپ رسیده است.

- KNN-based least squares twin support vector machine for pattern classification, Applied Intelligence, vol.48, pp.4551–4564, Dec 2018

فصل ۴

ماشین بردار پشتیبان دو قلو مبتنی بر رگولارسیون و نزدیک ترین همسایه

۱-۴ مقدمه

در بخش ۲-۳-۲ برخی از گسترش‌های TSVM معرفی شد. بعضی از این گسترش‌ها مبتنی بر رویکرد نزدیک ترین همسایه هستند. با وجود اینکه گسترش‌های با رویکرد نزدیک ترین همسایه مزیت‌هایی مانند دقت بیشتر را دارند، این روش‌ها سه نقطه ضعف دارند که عبارتند از:

۱. این روش‌ها به نمونه‌ها بر اساس شمارش تعداد همسایه‌های نزدیک‌شان وزن می‌دهند. بطوریکه فاصله بین نزدیک‌ترین همسایه‌های یک نمونه در نظر گرفته نمی‌شود. نسبت دادن وزن براساس فاصله می‌تواند باعث شناسایی بهتر نواحی پرتراکم شود. به عبارت دیگر، وزن بیشتری به یک نمونه با همسایه‌های نزدیک‌تر داده می‌شود تا به یک نمونه با همسایه‌های دورتر.
۲. این روش‌ها همانند TSVM خطای آموزشی (ریسک تجربی) را کمینه می‌کنند. از این رو امکان برآش بیش از حد وجود دارد و تعیین‌پذیری مدل خروجی کاهش می‌یابد. این نقطه ضعف با در نظر گرفتن ریسک ساختاری در تابع هدف مسئله بهینه‌سازی برطرف می‌شود.
۳. در این روش‌ها، گراف نزدیک‌ترین همسایه با الگوریتم جستجوی کامل^۱ (FSA) ساخته می‌شود. مرتبه زمانی این الگوریتم جستجو برابر با $O(m^2)$ است. بنابراین اجرای این الگوریتم روی مجموعه داده‌های بزرگ زمان برخواهد بود. اگرچه الگوریتم‌های جدیدی برای ساخت گراف نزدیک‌ترین همسایه ارائه شده است که مرتبه زمانی کمتر از روش FSA دارند. در سال ۲۰۱۵، الگوریتم گراف

^۱Full search algorithm

نزدیک‌ترین همسایه مبتنی بر تفاوت مکانی فاصله‌ها (LDMDBA) ارائه شد [۱۶]. مرتبه زمانی روش LDMDBA برابر با $\mathcal{O}(\log nm \log m)$ است. این روش جدید می‌تواند به منظور ساخت گراف نزدیک‌ترین همسایه استفاده شود. در نتیجه پیچیدگی محاسباتی کلی دسته بند بهبود می‌یابد.

در این فصل، با انگیزه‌ی برطرف کردن نقاط ضعف اشاره شده، دسته‌بند ماشین بردار پشتیبان دو قلو مبتنی بر رگولارسیون و نزدیک‌ترین همسایه (RKNN-TSVM) ارائه شده است. روش پیشنهادی برخلاف روش‌های WLTSVM و KNN-LSTSVM، به نمونه‌های آموزشی بر اساس فاصله بین نزدیک‌ترین همسایه‌هایش وزن می‌دهد. بطوریکه شناسایی نمونه‌های با تراکم بالا و فشردگی درون کلاسی بهبود می‌یابد. همچنین روش RKNN-TSVM ریسک ساختاری را کمینه می‌کند. بنابراین مسائل بهینه‌سازی در این روش معین مثبت^۲ هستند.

چالش اصلی روش RKNN-TSVM، پیچیدگی محاسباتی بالا برای مجموعه داده‌های بزرگ است. زیرا این روش دو مسئله بهینه‌سازی دوگان حل می‌کند و همچنین k نزدیک‌ترین همسایه برای تمام نمونه‌های آموزشی باید محاسبه شود. به منظور بهبود مرتبه زمانی گراف نزدیک‌ترین همسایه، روش‌هایی مانند درخت k بعدی (k-d tree³) [۳۴]، درخت LB^4 [۳۵] و روش LDMDBA ارائه شده است. روش LDMDBA دارای مرتبه زمانی $\mathcal{O}(\log nm \log m)$ است که از روش FSA و بیشتر الگوریتم‌های گراف نزدیک‌ترین همسایه بهتر است. روش RKNN-TSVM برای ساخت گراف نزدیک‌ترین همسایه از روش LDMDBA استفاده می‌کند.

دسته‌بند ارائه شده در این فصل، یعنی RKNN-TSVM دارای مزایای زیر است:

- روش RKNN-TSVM به نمونه‌ها بر اساس فاصله نزدیک‌ترین همسایه‌هایش وزن می‌دهد. به عبارت دیگر، نواحی پرترکم بهتر شناسایی می‌گردد و ابرصفحه به نمونه‌های با تراکم بالا نزدیک‌تر می‌شود. بطوریکه به نمونه‌های با همسایه‌های نزدیک‌تر وزن بیشتری نسبت داده می‌شود.
- برخلاف روش WLTSVM و KNN-LSTSVM، ریسک ساختاری در مسائل بهینه‌سازی روش-RKNN TSVM لحاظ شده است. بدین ترتیب دقیق و تعمیم‌پذیری مدل خروجی افزایش بهبود می‌یابد.
- روش LDMDBA جهت بهبود پیچیدگی محاسباتی دسته‌بند بکار گرفته شده است. همچنین این روش برای نسخه غیر خطی دسته‌بند RKNN-TSVM نیز موثر می‌باشد. بطوریکه پیدا کردن k

²Positive definite

³K-dimensional tree

⁴Lower bound tree

۲-۴. الگوریتم نزدیک‌ترین همسایه مبتنی بر تفاوت مکانی فاصله‌ها

نزدیک‌ترین همسایه در فضای ویژگی با ابعاد بسیار بالا توسط روش LMDDBA بسیار سریعتر از روش FSA است.

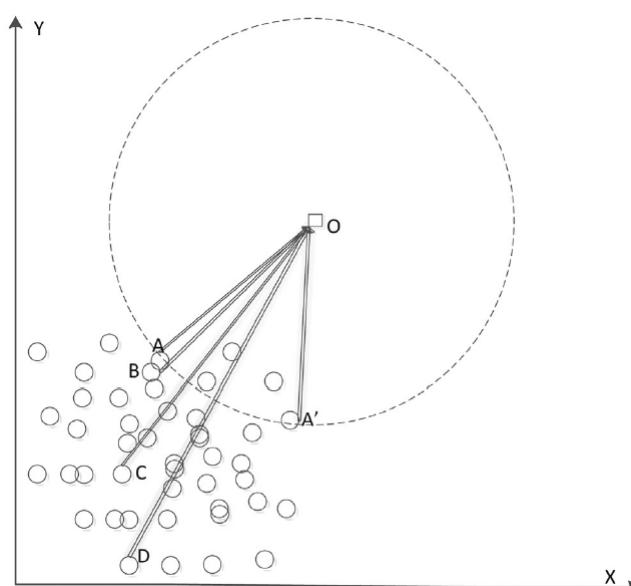
- شیوه وزن دهی به یک نمونه در روش RKNN-TSVM، براساس فاصله نمونه مورد نظر از نزدیک‌ترین همسایه‌هایش است. بطوریکه مدل خروجی حساسیت کمتری نسبت به نمونه‌های نویزی و پرت دارد.

در ادامه این فصل، ابتدا روش LMDDBA به طور خلاصه معرفی می‌شود. شیوه جدید وزن دهی در بخش ۳-۴ بیان شده است. نسخه خطی و غیر خطی دسته‌بند RKNN-TSVM به ترتیب در بخش‌های ۴-۴ و ۵-۴ شرح داده می‌شود. سپس دسته‌بند RKNN-TSVM در بخش ۶-۴ تحلیل و بررسی می‌شود.

۲-۴ الگوریتم نزدیک‌ترین همسایه مبتنی بر تفاوت مکانی فاصله‌ها

الگوریتم LMDDBA مفهوم تفاوت مکانی را معرفی کرد [۱۶]. ایده اصلی این است که همسایه‌ها راجع به مکان‌شان اطلاعات مشابه‌ای دارند. به منظور شرح بهتر، شکل ۱-۴ ایده روش LMDDBA را نشان می‌دهد.

همانطور که در شکل ۱-۴ نشان داده شده است، تفاوت مکانی همسایه‌ها از طریق فاصله آن‌ها از نقطه مرجع O اندازه‌گیری می‌شود. برای مثال، نمونه A همسایه نزدیک نمونه B است. زیرا فاصله هر



شکل ۱-۴: ایده اصلی روش LMDDBA [۱۶]

۲-۴. الگوریتم نزدیک‌ترین همسایه مبتنی بر تفاوت مکانی فاصله‌ها

دو نمونه از نقطه مرجع بسیار مشابه یکدیگر است. با این حال فاصله از یک نقطه مرجع برای پیدا کردن دقیق نزدیک‌ترین همسایه‌ها کافی نمی‌باشد. بنابراین فاصله از چندین نقطه مرجع محاسبه می‌شود. به منظور بیان الگوریتم LDMDBA، مسئله پیدا کردن نزدیک‌ترین همسایه با مجموعه داده $T = \{(x_1, y_1), \dots, (x_m, y_m)\}$ را در نظر می‌گیریم. فاصله یک نمونه $x_j \in T$ از نقطه مرجع O_1 به این صورت $\|x_j - O_1\| = \|x_j - O_1\|_{Dis_1(x_j)}$ نشان داده می‌شود. تعداد نقاط مرجع طبق مقاله اصلی [۱۶] برابر با $\log_2 n$ است (یعنی تعداد ویژگی‌ها است). مقادیر i بعد اول نمین نقطه مرجع O_i مساوی با ۱ و سایر مقادیر برابر با ۰ است. به عبارت دیگر، نقطه مرجع i ام به این صورت $(1, 1, \dots, -1, -1, \dots, -1)$ معرفی شده است. همچنین $Nea_i(x_j)$ نشان‌دهنده همسایه‌های نمونه x_j است که توسط نقطه مرجع i ام مشخص شده است.

به منظور بدست آوردن مجموعه $Nea_i(x_j)$ ، ابتدا فاصله نمونه x_j از تمام نقاط مرجع محاسبه می‌شود. بعد از مرتب‌سازی فاصله‌ها، یک دنباله مرتب شده بدست می‌آید. بطوریکه نزدیک‌ترین همسایه نمونه x_j در یک زیردنباله با مرکزیت نمونه x_j قرار دارد. طول زیردنباله برابر با $2k * \epsilon$ است. بطوریکه مقدار ϵ مساوی با $\log_2 \log_2 m$ می‌باشد (نحوه انتخاب مقدار ϵ در مقاله اصلی [۱۶] ذکر شده است). در آخر، فاصله اقلیدسی تمام نمونه‌ها در زیردنباله محاسبه می‌شود. آن دسته از نمونه‌های متناظر با k کوچک‌ترین فاصله در زیردنباله به عنوان نزدیک‌ترین همسایه نمونه x_j شناخته می‌شوند. در جمع‌بندی این بخش، الگوریتم LDMDBA به صورت گام به گام در الگوریتم ۱-۴ بیان می‌شود.

مرتبه زمانی الگوریتم LDMDBA توسط گام‌های ۳ و ۵ مشخص می‌شود. مرتبه زمانی الگوریتم مرتب‌سازی استفاده شده در این گام‌ها برابر با $\mathcal{O}(m \log_2 m)$ است. بنابراین پیچیدگی محاسباتی کلی الگوریتم LDMDBA مساوی با $\mathcal{O}(\log nm \log m)$ خواهد بود. مرتبه زمانی الگوریتم FSA برابر با $\mathcal{O}(m^2 \log_2 m)$ است که از الگوریتم LDMDBA بیشتر است.

مزیت مهم دیگر الگوریتم LDMDBA این است که از ساختار درختی استفاده نمی‌کند. درنتیجه این الگوریتم برای داده‌های با ابعاد بسیار بالا موثر است. نتایج ارزیابی در مقاله اصلی [۱۶]، برتری این الگوریتم را نسبت به FSA و سایر الگوریتم‌های نزدیک‌ترین همسایه نشان می‌دهد.

الگوریتم ۱-۴: الگوریتم نزدیک‌ترین همسایه مبتنی بر تفاوت مکانی فاصله‌ها (LDMDBA)

ابتدا مجموعه آموزشی T را در نظر می‌گیریم. سپس مقدار k یعنی تعداد نزدیک‌ترین همسایه‌ها مشخص می‌گردد. با در نظر گرفتن $a = \min_{j \in T} \text{dis}_i(x_j)$ با طی کردن گام‌های زیر بدست می‌آید.

۱. نمین نقطه مرجع O_i به عنوان یک بردار در نظر گرفته می‌شود که بعد اول آن برابر ۱-

است. سایر مقادیر این بردار را مقادیر ۱ تشکیل می‌دهد.

۲. فاصله تمام نمونه‌ها از نقطه مرجع O_i با استفاده از $\text{dis}_i(x_j) = \|x_j - O_i\|$ محاسبه می‌شود.

۳. تمام نمونه‌ها بر اساس مقدار dis_i مرتب می‌شود و یک دنباله مرتب شده بدست می‌آید.

۴. یک زیردنباله‌ای از نمونه‌ها با اندازه $2k * \log_2 \log_2 m$ و x_j به عنوان نمونه مرکز را در نظر گرفته می‌شود. سپس فاصله اقلیدسی نمونه x_j از تمام نمونه‌های زیردنباله محاسبه می‌گردد.

۵. فاصله‌های محاسبه شده در گام ۴ مرتب می‌شود.

۶. کوچک‌ترین فاصله در زیردنباله مرتب شده، a نزدیک‌ترین همسایه نمونه x_j هستند.

۷. چنانچه همسایه تمام نمونه‌ها با استفاده از همه نقاط مرجع محاسبه گردد، الگوریتم خاتمه می‌یابد. در غیر این صورت، $i + 1 = i$ و الگوریتم در گام ۱ ادامه می‌یابد.

۳-۴ تعریف ماتریس وزن‌ها

گسترش‌های TSVM مبتنی بر رویکرد نزدیک‌ترین همسایه [۱۴، ۳۰، ۳۱]، گراف G را برای مدل کردن شباهت نمونه‌ها به صورت زیر تعریف می‌کنند.

$$W_{ij} = \begin{cases} 1, & \text{if } x_i \in \text{Nea}(x_j) \text{ or } x_j \in \text{Nea}(x_i), \\ 0, & \text{otherwise.} \end{cases} \quad (1-4)$$

در رابطه ۱-۴، مجموعه $\text{Nea}(x_j)$ به صورت زیر تعریف می‌شود.

$$\text{Nea}(x_j) = \{x_j^i \mid \text{if } x_j^i \text{ is a knn of } x_j, 1 \leq i \leq k\} \quad (2-4)$$

مجموعه $Nea(x_j)$ بر اساس فاصله اقلیدسی $d(x_j, x_j^i)$ بین نمونه x_i و x_j مرتب است.

$$d(x_j, x_j^i) = \sqrt{(x_j - x_j^i)^T (x_j - x_j^i)} \quad (3-4)$$

با این حال گراف ۱-۴ یک نقطه ضعف دارد. مقدار W_{ij} برابر ۱ یا ۰ است. بطوريکه بین هر کدام از نزدیکترین همسایه‌های نمونه x_j تمایزی قائل نمی‌شود. به منظور بهبود شیوه وزن‌دهی، وزن به یک نمونه بر اساس فاصله بین نزدیکترین همسایه‌هایش نسبت داده می‌شود. گراف G در روش RKNN-TSVM با گرفتن ایده از [۳۶، ۳۷] به صورت زیر بازتعریف شده است.

$$W_{ij} = \begin{cases} w_{ij}, & \text{if } x_i \in Nea(x_j) \text{ or } x_j \in Nea(x_i), \\ 0, & \text{otherwise.} \end{cases} \quad (4-4)$$

در رابطه ۴-۴، w_{ij} نشان‌دهنده وزن i مین نزدیکترین همسایه نمونه x_j که به صورت زیر تعریف می‌شود.

$$w_{ij} = \begin{cases} \frac{d(x_i, x_j^k) - d(x_i, x_j)}{d(x_i, x_j^k) - d(x_i, x_j)}, & \text{if } d(x_i, x_j^k) \neq d(x_i, x_j), \\ 1, & \text{if } d(x_i, x_j^k) = d(x_i, x_j). \end{cases} \quad (5-4)$$

بر اساس رابطه ۵-۴، نمونه x_i با فاصله کمتر وزن بیشتری می‌گیرد نسبت به نمونه‌ای با فاصله بیشتر. بنابراین مقادیر w_{ij} به صورت خطی بین بازه [۰، ۱] قرار می‌گیرد.

به منظور مدل کردن اطلاعات درون و بروون کلاسی بر اساس فاصله بین نزدیکترین همسایه‌ها، گراف درون کلاسی G_s و گراف بروون کلاسی G_d به ترتیب در روابط ۶-۴ و ۷-۴ تعریف شده است.

$$W_{s,ij} = \begin{cases} w_{ij}, & \text{if } x_i \in Nea_s(x_j) \text{ or } x_j \in Nea_s(x_i), \\ 0, & \text{otherwise.} \end{cases} \quad (6-4)$$

$$W_{d,ij} = \begin{cases} w_{ij}, & \text{if } x_i \in Nea_d(x_j), \\ 0, & \text{otherwise.} \end{cases} \quad (7-4)$$

مجموعه‌های $Nea_d(x_j)$ و $Nea_s(x_j)$ به ترتیب نشان‌دهنده k نزدیکترین همسایه نمونه x_j در کلاس مثبت و منفی است. این دو مجموعه در رابطه زیر تعریف شده‌اند.

$$Nea_s(x_j) = \{x_j^i \mid l(x_j^i) = l(x_j), 1 \leq i \leq k\} \quad (8-4)$$

$$Nea_d(x_j) = \{x_j^i \mid l(x_j^i) \neq l(x_j), 1 \leq i \leq k\} \quad (9-4)$$

در رابطه ۸-۴ و ۹-۴، $l(x_j)$ نشان‌دهنده برچسب نمونه x_j است. بدیهی است که $Nea_s(x_j) \cap Nea_d(x_j) = \emptyset$ و $W_{s,ij} \neq 0$ برقرار است. زمانی که $W_{s,ij} = 0$ یا $W_{d,ij} \neq 0$ یک یال بی‌جهت به گره x_i و x_j در گراف متناظر ش اضافه می‌شود.

برخلاف روش TSVM، فقط بردارهای پشتیبان به جای کل نمونه‌های آموزشی در ایجاد ابرصفحه بهینه کلاس متناظر اهمیت دارند. به منظور استخراج بردارهای پشتیبان (نمونه‌های حاشیه‌ای) از نمونه‌های کلاس منفی، ماتریس وزن W_d به صورت زیر بازتعریف می‌شود.

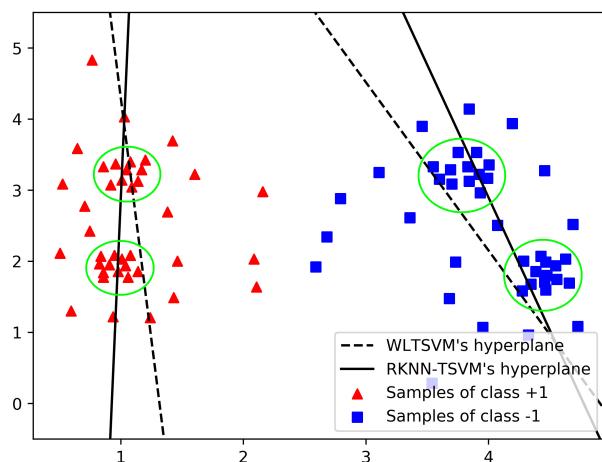
$$f_j = \begin{cases} 1, & \exists j, W_{d,ij} \neq 0, \\ 0, & \text{otherwise.} \end{cases} \quad (10-4)$$

۴-۴ نسخه خطی

روش پیشنهادی مشابه روش TSVM دو ابرصفحه غیر موازی را ایجاد می‌کند. با این حال هر ابرصفحه به نمونه‌های پرتراکم کلاس خودش نزدیک می‌شود. همچنین روش پیشنهادی ریسک ساختاری را کمینه می‌کند. در حالی که در روش TSVM خطای آموزشی یا ریسک تجربی کمینه می‌شود. شکل ۴-۴ ایده اصلی روش RKNN-TSVM را روی یک مجموعه داده مصنوعی نشان می‌دهد.

در شکل ۴-۴، دایره‌های سبز رنگ نواحی پرتراکم را نشان می‌دهد. ابرصفحه روش RKNN-TSVM نسبت به روش WLTSVM به نواحی پرتراکم نزدیکتر است. زیرا روش RKNN-TSVM به نمونه‌ها براساس فاصله‌شان از نزدیک‌ترین همسایه‌ها وزن می‌دهد. بطوریکه روش پیشنهادی حساسیت کمتری نسبت به نمونه‌های پرت و نویزی دارد.

بعد از محاسبه ماتریس وزن کلاس مثبت $W_{s;j}^{(1)}$ و نمونه‌های حاشیه‌ای کلاس منفی $f_j^{(2)}$ ، مسائل



شکل ۴-۴: ایده اصلی روش RKNN-TSVM

بهینه‌سازی اصلی روش پیشنهادی با در نظر گرفتن ریسک ساختاری به صورت زیر تعریف می‌شوند.

$$\begin{aligned} \min_{w_1, b_1} \quad & \frac{1}{2} \sum_{i=1}^{m_1} d_j^{(1)} (w_1^T x_j^{(1)} + b_1)^2 + c_1 e_1^T \xi + \frac{c_2}{2} (\|w_1\|^2 + b_1^2) \\ \text{s.t.} \quad & -f_j^{(2)} (w_1^T x_j^{(2)} + b_1) + \xi_j \geq f_j^{(2)} \end{aligned} \quad (11-4)$$

$$\xi_j \geq 0, \quad j = 1, \dots, m_1$$

$$\begin{aligned} \min_{w_2, b_2} \quad & \frac{1}{2} \sum_{i=1}^{m_2} d_j^{(2)} (w_2^T x_j^{(2)} + b_2)^2 + c_2 e_2^T \eta + \frac{c_3}{2} (\|w_2\|^2 + b_2^2) \\ \text{s.t.} \quad & f_j^{(1)} (w_2^T x_j^{(1)} + b_2) + \eta_j \geq f_j^{(1)} \end{aligned} \quad (12-4)$$

$$\eta_j \geq 0, \quad j = 1, \dots, m_2$$

در روابط ۱۱-۴ و ۱۲-۴، $c_1, c_2, c_3 \geq 0$ پارامترهای مثبت، ξ و η متغیر لغزش مثبت، e_1 و e_2 به ترتیب بردار ستونی با ابعاد m_1 و m_2 هستند. همچنین $d_j^{(1)}$ نشان‌دهنده وزن نمونه $x_j^{(1)}$ که در رابطه زیر تعریف شده است.

$$d_j^{(1)} = \sum_{i=1}^{n_1} W_{s,ij}^{(1)}, \quad j = 1, 2, \dots, n_1 \quad (13-4)$$

تفاوت بین مسائل بهینه‌سازی اصلی روش پیشنهادی و گسترش‌های روش TSVM مبتنی بر گراف نزدیک‌ترین همسایه عبارتند از:

۱. برخلاف روش WLTSVM، مقدار به $d_j^{(1)}$ فاصله میان n نزدیک‌ترین همسایه نمونه $x_j^{(1)}$ بستگی دارد. به عبارت دیگر، مقدار $d_j^{(1)}$ نشان‌دهنده تراکم نمونه $x_j^{(1)}$ است.

۲. در مسائل بهینه‌سازی اصلی ۱۱-۴ و ۱۲-۴، ریسک ساختاری با اضافه شدن جمله رگولارسیون $\frac{c_2}{2} (\|w_1\|^2 + b_1^2)$ کمینه می‌شود. در حالی که خطای نمونه آموزشی در گسترش‌های روش TSVM مبتنی بر گراف نزدیک‌ترین همسایه کمینه می‌گردد.

همچنین روش پیشنهادی (RKNN-TSVM) مزیت‌های گسترش‌های مبتنی بر گراف نزدیک‌ترین همسایه را دارد که عبارتند از:

۱. مسائل بهینه‌سازی ۱۱-۴ و ۱۲-۴ محدب^۵ و از نوع برنامه‌ریزی درجه دو هستند. بطوریکه این مسائل راه حل بهینه سراسری دارند.

⁵Convex

۲. همانند روش WLTSVM، پیچیدگی زمانی روش پیشنهادی با در نظر گرفتن نمونه‌های حاشیه‌ای در قيد مسئله بهینه‌سازی بهبود یافته است.

به منظور حل کردن مسئله بهینه‌سازی ۱۱-۴، تابع لاگرانژ به صورت زیر تعریف شده است.

$$L_1(w_1, b_1, \xi, \alpha, \gamma) = \frac{1}{2} \sum_{i=1}^{m_1} d_j^{(1)} (w_1^T x_j^{(1)} + b_1)^2 + c_1 e_1^T \xi + \frac{c_2}{2} (\|w_1\|^2 + b_1^2) - \sum_{i=1}^{m_2} \alpha_j (-f_j^{(2)} (w_1^T x_j^{(2)} + b_1) + \xi_j - f_j^{(2)}) - \gamma^T \xi \quad (14-4)$$

در رابطه ۱۴-۴، w_1, b_1 و ξ بردارهای ضرایب لاگرانژ هستند.

با مشتق‌گیری از تابع ۱۴-۴ نسبت به w_1, b_1 و ξ شرایط KKT زیر برقرار می‌شود.

$$\frac{\partial L_1}{\partial w_1} = \sum_{i=1}^{m_1} d_j^{(1)} x_j^{(1)} (w_1^T x_j^{(1)} + b_1) + c_1 w_1 + \sum_{i=1}^{m_2} \alpha_j f_j^{(2)} x_j^{(2)} = 0, \quad (15-4)$$

$$\frac{\partial L_1}{\partial b_1} = \sum_{i=1}^{m_1} d_j^{(1)} (w_1^T x_j^{(1)} + b_1) + c_1 b_1 + \sum_{i=1}^{m_2} \alpha_j f_j^{(2)} = 0, \quad (16-4)$$

$$\frac{\partial L_1}{\partial \xi} = c_1 e_1 - \alpha - \gamma = 0, \quad (17-4)$$

$$\alpha \geq 0, \gamma \geq 0. \quad (18-4)$$

با قرار دادن روابط ۱۵-۴ و ۱۶-۲ به صورت فرم ماتریسی، معادلات زیر بدست می‌آید.

$$A^T D (Aw_1 + e_1 b_1) + c_1 w_1 + B^T F \alpha = 0, \quad (19-4)$$

$$e_1^T D (Aw_1 + e_1 b_1) + c_1 b_1 + e_1^T F \alpha = 0, \quad (20-4)$$

در روابط ۱۹-۴ و ۲۰-۴ ماتریس‌های $F = diag(f_1^{(2)}, f_2^{(2)}, \dots, f_{m_2}^{(2)})$ و $D = diag(d_1^{(1)}, d_2^{(1)}, \dots, d_{m_1}^{(1)})$ قدری هستند. در اینجا $d_j^{(1)} \geq f_j^{(2)}$ مقدار برابر با ۰ یا ۱ است. با توجه به اینکه $\alpha \geq 0$ از رابطه ۱۷-۴ خواهیم داشت:

$$0 \leq \alpha \leq c_1 e_1 \quad (21-4)$$

با ترکیب کردن روابط ۱۹-۴ و ۲۰-۴ معادله زیر بدست می‌آید.

$$([A^T e_1^T] D [A e_1] + c_1 I) [w_1 \ b_1]^T + [B^T e_1^T] F \alpha = 0. \quad (22-4)$$

در رابطه ۲۲-۴، I ماتریس همانی با ابعاد مناسب است. با تعریف کردن ماتریس‌های $H = [A e_1]$ و

رابطه ۲۲-۴ به صورت زیر بازنویسی می‌شود.

$$(H^T D H + c_2 I) \begin{bmatrix} w_1 \\ b_1 \end{bmatrix} + G^T F \alpha = 0. \quad (23-4)$$

$$i.e., \begin{bmatrix} w_1 \\ b_1 \end{bmatrix} = -(H^T D H + c_2 I)^{-1} G^T F \alpha$$

با توجه به رابطه ۱۴-۴ و شرایط KKT ذکر شده، حالت دوگان مسئله ۱۱-۴ در رابطه زیر تعریف شده است.

$$\begin{aligned} \max_{\alpha} \quad & e_1^T F \alpha - \frac{1}{2} \alpha^T (F^T G) (H^T D H + c_2 I)^{-1} (G^T F) \alpha \\ \text{s.t.} \quad & 0 \leq \alpha \leq c_1 e_1 \end{aligned} \quad (24-4)$$

لازم به ذکر است که پارامتر c_2 در مسئله دوگان ۲۴-۴ می‌تواند با $0 < \varepsilon$ جایگزین شود. بطوریکه پارامتر ε یک عدد ثابت بسیار کوچک مانند $1e-8 = \varepsilon$ است. در حالی که پارامتر c_2 تعادل بین جمله رگولارسیون و ریسک تجربی را تعیین می‌کند [۱۳].

مشابه کلاس مثبت،تابع لاغرانژ مسئله اصلی کلاس منفی ۱۲-۴ به صورت زیر تعریف می‌شود.

$$\begin{aligned} L_2(w_2, b_2, \eta, \beta, \nu) = & \frac{1}{2} \sum_{i=1}^{n_2} d_j^{(2)} (w_2^T x_j^{(2)} + b_2)^2 + c_1 e_1^T \eta + \frac{c_2}{2} (\|w_2\|^2 + b_2^2) \\ & - \sum_{i=1}^{n_1} \beta_j (f_j^{(1)} (w_2^T x_j^{(2)} + b_2) + \eta_j - f_j^{(1)}) - \nu^T \eta \end{aligned} \quad (25-4)$$

در رابطه ۲۵-۴، $\nu = (\nu_1, \nu_2, \dots, \nu_{n_1})^T$ و $\beta = (\beta_1, \beta_2, \dots, \beta_{n_2})^T$ بردارهای ضرایب لاغرانژ هستند.

با مشتق‌گیری از تابع لاغرانژ ۲۵-۴ نسبت به w_2 , b_2 و η حالت دوگان مسئله اصلی ۱۲-۴ به صورت زیر تعریف می‌شود.

$$\begin{aligned} \max_{\beta} \quad & e_1^T P \beta - \frac{1}{2} \beta^T (P^T H) (G^T Q G + c_3 I)^{-1} (H^T P) \beta \\ \text{s.t.} \quad & 0 \leq \beta \leq c_1 e_1 \end{aligned} \quad (26-4)$$

در رابطه ۲۶-۴، $P = diag(f_1^{(1)}, f_2^{(1)}, \dots, f_{n_1}^{(1)})$ و $Q = diag(d_1^{(2)}, d_2^{(2)}, \dots, d_{n_2}^{(2)})$ به ترتیب ماتریس وزن کلاس منفی و نمونه‌های حاشیه‌ای کلاس مثبت هستند. همچنین مقدار $f_j^{(1)}$ برابر با 0 یا 1 است. بررسی مسئله دوگان ۲۴-۴ و ۲۶-۴ نشان می‌دهد که پیچیدگی محاسباتی روش RKNN-TSVM وابسته به نمونه‌های حاشیه‌ای است.

زمانی که مسئله دوگان ۲۶-۴ حل شود، ابرصفحه کلاس منفی از طریق معادله زیر بدست می‌آید.

$$\begin{bmatrix} w_2 \\ b_2 \end{bmatrix} = (G^T Q G + c_3 I)^{-1} H^T P \beta \quad (27-4)$$

کلاس نمونه جدید $x \in \mathbb{R}^d$ از طریقتابع تصمیم زیر مشخص می‌شود.

$$d(x) = \begin{cases} +1, & \text{if } \frac{|x^T w_1 + b_1|}{\|w_1\|} < \frac{|x^T w_2 + b_2|}{\|w_2\|} \\ -1, & \text{otherwise.} \end{cases} \quad (28-4)$$

مراحل ایجاد مدل خطی دسته‌بند RKNN-TSVM در الگوریتم ۲-۴ ذکر شده است.

الگوریتم ۲-۴: ایجاد مدل خطی دسته‌بند RKNN-TSVM

ابتدا فرض می‌گیریم که مجموعه آموزشی $T = \{(x_1, y_1), \dots, (x_m, y_m)\}$ را در اختیار داریم و k پارامتر یعنی تعداد نزدیک‌ترین همسایه‌ها مشخص شده است. سپس مدل خطی روش RKNN-TSVM با طی کردن گام‌های زیر بدست می‌آید.

۱. به منظور بدست آوردن مجموعه $Nea(x_j)$ ، نزدیک‌ترین همسایه هر نمونه $x_j \in T$ را با

استفاده الگوریتم FSA یا LDMDBA پیدا کنید.

۲. ماتریس وزن W_s و W_d برای کلاس مثبت و منفی با استفاده از روابط ۶-۴ و ۷-۴ تعریف

می‌شود.

۳. ماتریس‌های قطری D ، Q و P با استفاده از روابط ۱۰-۴ و ۱۳-۴ بدست می‌آید.

۴. ابتدا ماتریس‌های ورودی $B \in \mathbb{R}^{m_2 \times n}$ و $A \in \mathbb{R}^{m_1 \times n}$ معین می‌شود. سپس ماتریس‌های

G و H به این صورت $G = [Ae_1]$ و $H = [Be_2]$ تعریف می‌گردد.

۵. پارامترهای c_1 ، c_2 و c_3 تعیین می‌شود. این پارامتر معمولاً براساس مجموعه داده صحت

مشخص می‌شود.

۶. راه حل بهینه α و β به ترتیب با حل کردن مسائل ۲۴-۴ و ۲۶-۴ دوگان بدست می‌آید.

۷. مختصات دو ابرصفحه غیر موازی با حل کردن معادلات ۲۳-۴ و ۲۷-۴ تعیین می‌شود.

۸. کلاس نمونه جدید $x \in \mathbb{R}^n$ از طریقتابع تصمیم ۲۸-۴ مشخص می‌شود.

در جمع‌بندی نسخه خطی، ذکر دو نکته زیر ضروری است.

۱. راه حل‌های نسخه خطی ۲۳-۴ و ۲۷-۴ به ترتیب شامل دو معکوس ماتریس $(H^T DH + c_2 I)^{-1}$

و $(G^T QG + c_3 I)^{-1}$ با ابعاد $(n+1) \times (n+1)$ می‌باشد. بطوریکه n بسیار کوچکتر از نمونه‌های

آموزشی است ($n \ll m$).

۲. به دلیل اضافه شدن جمله رگولارسیون بهتابع هدف، ماتریس‌های $(G^T QG + c_3 I)$ و $(H^T DH + c_2 I)$ معین مثبت هستند. بنابراین روش پیشنهادی پایدار است و از شرایط منفرد ماتریس‌های $H^T DH$ و $G^T QG$ جلوگیری می‌کند.

۵-۴ نسخه غیر خطی

در دنیای واقعی، بسیاری از مسائل دسته‌بندی با تابع هسته خطی جدا پذیر نیستند. به منظور جدا سازی مسائل غیر خطی، نمونه‌ها به فضای ویژگی با ابعاد بیشتر نگاشت می‌شوند. بنابراین روش RKNN-TSVM با در نظر گرفتن دو ابرسطح زیر به نسخه غیر خطی گسترش می‌یابد.

$$K(x)\mu_1 + b_1 = 0, \quad \text{and} \quad K(x)\mu_2 + b_2 = 0 \quad (29-4)$$

در رابطه ۲۹-۴، $K(x)$ نشان دهنده تابع هسته دلخواه است که به صورت زیر تعریف می‌شود.

$$K(x) = [K(x_1, x), K(x_2, x), \dots, K(x_n, x)]^T \quad (30-4)$$

مسائل بهینه‌سازی اصلی نسخه غیر خطی روش RKNN-TSVM به صورت زیر تعریف می‌گردد.

$$\begin{aligned} \min_{\mu_1, b_1} \quad & \frac{1}{2} \sum_{i=1}^{m_1} d_j^{(1)} (\mu_1^T K(x_j^{(1)}) + b_1)^2 + c_1 e_1^T \xi + \frac{c_2}{2} (\|\mu_1\|^2 + b_1^2) \\ \text{s.t.} \quad & -f_j^{(1)} (\mu_1^T K(x_j^{(1)}) + b_1) + \xi_j \geq f_j^{(1)} \end{aligned} \quad (31-4)$$

$$\xi_j \geq 0, \quad j = 1, \dots, m_1$$

$$\begin{aligned} \min_{\mu_2, b_2} \quad & \frac{1}{2} \sum_{i=1}^{m_2} d_j^{(2)} (\mu_2^T K(x_j^{(2)}) + b_2)^2 + c_1 e_2^T \eta + \frac{c_3}{2} (\|\mu_2\|^2 + b_2^2) \\ \text{s.t.} \quad & f_j^{(2)} (\mu_2^T K(x_j^{(2)}) + b_2) + \eta_j \geq f_j^{(2)} \end{aligned} \quad (32-4)$$

$$\eta_j \geq 0, \quad j = 1, \dots, m_2$$

در روابط ۳۱-۴ و ۳۲-۴ و c_1, c_2, c_3 پارامترهای مثبت، ξ و η متغیر لغزش هستند. همچنین d_j و f_j همانند نسخه خطی تعریف می‌شوند. فاصله اقلیدسی نیز در فضای ویژگی با ابعاد بالا محاسبه می‌شود. مشابه نسخه خطی، تابع لاگرانژ مسئله بهینه‌سازی اصلی ۴ ۳۲-۴ به صورت زیر تعریف می‌شود.

$$L_1(\mu_1, b_1, \xi, \alpha, \gamma) = \frac{1}{2} \sum_{i=1}^{n_1} d_j^{(1)} (\mu_1^T K(x_j^{(1)}) + b_1)^2 + c_1 e_1^T \xi + \frac{c_2}{2} (\|\mu_1\|^2 + b_1^2) - \sum_{i=1}^{n_2} \alpha_j (-f_j^{(2)} (\mu_1^T K(x_j^{(2)}) + b_1) + \xi_j - f_j^{(2)}) - \gamma^T \xi \quad (33-4)$$

در رابطه ۳۳-۴، $\gamma = (\gamma_1, \gamma_2, \dots, \gamma_{n_2})^T$ و $\alpha = (\alpha_1, \alpha_2, \dots, \alpha_{n_2})^T$ نشان دهنده بردارهای ضرایب

لاگرانژ هستند. شرایط KKT برای μ_1, b_1, ξ و α, γ به صورت زیر تعریف شده است.

$$\frac{\partial L_1}{\partial \mu_1} = \sum_{i=1}^{n_1} d_j^{(1)} K(x_j^{(1)}) (\mu_1^T K(x_j^{(1)}) + b_1) + c_2 \mu_1 + \sum_{i=1}^{n_2} \alpha_j f_j^{(2)} K(x_j^{(2)}) = 0, \quad (34-4)$$

$$\frac{\partial L_1}{\partial b_1} = \sum_{i=1}^{n_1} d_j^{(1)} (w_1^T x_j^{(1)} + b_1) + c_2 b_1 + \sum_{i=1}^{n_2} \alpha_j f_j^{(2)} = 0, \quad (35-4)$$

$$\frac{\partial L_1}{\partial \xi} = c_1 e_1 - \alpha - \gamma = 0, \quad (36-4)$$

$$\alpha \geq 0, \gamma \geq 0. \quad (37-4)$$

با قرار دادن روابط ۳۴-۴ و ۳۵-۴ به صورت فرم ماتریسی، معادلات زیر بدست می‌آید.

$$K(A)^T D(K(A)\mu_1 + e_1 b_1) + c_2 \mu_1 + K(B)^T F \alpha = 0, \quad (38-4)$$

$$e_1^T D(K(A)\mu_1 + e_1 b_1) + c_2 b_1 + e_1^T F \alpha = 0, \quad (39-4)$$

در روابط ۳۸-۴ و ۳۹-۴ به ترتیب ماتریس‌های هسته با ابعاد $m_2 \times m$ و $m_1 \times m$ و $K(A)$ و $K(B)$ با ابعاد $m_1 + m_2$ داریم.

هر دو ماتریس‌ها هستند ($m = m_1 + m_2$). با توجه به اینکه $0 \leq \alpha \leq c_1 e_1$ از رابطه ۳۷-۲ خواهیم داشت:

$$0 \leq \alpha \leq c_1 e_1 \quad (40-4)$$

به طرز مشابه‌ای، با ترکیب روابط ۳۸-۴ و ۳۹-۴ معادله زیر بدست می‌آید.

$$([K(A)^T e_1^T] D [K(A) e_1] + c_2 I) [\mu_1 \ b_1]^T + [K(B)^T e_1^T] F \alpha = 0. \quad (41-4)$$

با تعریف ماتریس‌های $R = [K(A) e_1]$ و $S = [K(B) e_1]$ رابطه ۴۱-۴ به صورت زیر بازنویسی

می‌شود.

$$\begin{bmatrix} \mu_1 \\ b_1 \end{bmatrix} = -(R^T D R + c_2 I)^{-1} S^T F \alpha \quad (42-4)$$

سپس حالت دوگان مسئله اصلی ۳۱-۴ به صورت زیر تعریف می‌گردد.

$$\max_{\alpha} \quad e_1^T F \alpha - \frac{1}{2} \alpha^T (F^T S) (R^T D R + c_2 I)^{-1} (S^T F) \alpha \quad (43-4)$$

$$\text{s.t.} \quad 0 \leq \alpha \leq c_1 e_1$$

با جابه‌جا کردن نقش‌های $K(A)$ و $K(B)$ در رابطه ۴-۳، حالت دوگان مسئله اصلی ۴-۲ به صورت زیر بدست می‌آید.

$$\begin{aligned} \max_{\beta} \quad & e_1^T P \beta - \frac{1}{2} \beta^T (P^T R) (S^T Q S + c_1 I)^{-1} (R^T P) \beta \\ \text{s.t.} \quad & e_1 \leq \beta \leq c_1 e_1 \end{aligned} \quad (44-4)$$

زمانی که مسئله دوگان ۴-۴ حل شود، ابرصفحه کلاس منفی از طریق حل معادله زیر بدست می‌آید.

$$\begin{bmatrix} \mu_2 \\ b_2 \end{bmatrix} = (S^T Q S + c_1 I)^{-1} R^T P \beta \quad (45-4)$$

در اینجا ماتریس‌های D , F , P و Q مشابه نسخه خطی تعریف می‌شوند.تابع تصمیم در نسخه غیر خطی به صورت زیر تعریف می‌گردد.

$$d(x) = \begin{cases} +1, & \text{if } \frac{|K(x)\mu_2 + b_2|}{\|\mu_2\|} < \frac{|K(x)\mu_1 + b_1|}{\|\mu_1\|} \\ -1, & \text{otherwise.} \end{cases} \quad (46-4)$$

مراحل ایجاد مدل غیر خطی دسته‌بند RKNN-TSVM در الگوریتم ۴-۳ ذکر شده است. در پایان این زیر بخش، لازم به ذکر است که نسخه غیر خطی روش RKNN-TSVM نیاز به دو معکوس ماتریس با ابعاد $(1 \times m) \times (m \times 1)$ را دارد. به منظور کاهش پیچیدگی محاسباتی در نسخه غیر خطی، دو رویکرد زیر را می‌توان استفاده کرد.

۱. تابع هسته مستطیلی [۲۱] برای کاهش ابعاد نمونه‌ها می‌تواند استفاده شود.
۲. فرمول SMW برای محاسبه معکوس ماتریس با ابعاد کوچک‌تر از $(1 \times m) \times (m \times 1)$ قابل استفاده است.

۶-۴ تحلیل روش RKNN-TSVM

در این زیر بخش، ابتدا پیچیدگی محاسباتی روش RKNN-TSVM بررسی می‌شود. سپس روش پیشنهادی با سایر دسته‌بندهای مرتبط مقایسه شده است. همچنین محدودیت‌های روش RKNN-TSVM در این زیر بخش بیان شده است. در آخر مقایسه‌پذیری روش RKNN-TSVM توضیح داده شده است.

الگوریتم ۳-۴: ایجاد مدل غیر خطی دسته‌بند RKNN-TSVM

ابتدا فرض می‌گیریم که مجموعه آموزشی $\{(x_1, y_1), \dots, (x_m, y_m)\}$ را در اختیار داریم و k پارامتر یعنی تعداد نزدیکترین همسایه‌ها مشخص شده است. سپس مدل خطی روش RKNN-TSVM با طی کردن گام‌های زیر بدست می‌آید.

۱. ابتدا یکتابع هسته برای نگاشت نمونه‌های آموزشی به فضای ویژگی انتخاب می‌شود.

غالبا از تابع RBF در این مرحله استفاده می‌شود.

۲. به منظور بدست آوردن مجموعه $N_{ea}(x_j)$ ، k نزدیکترین همسایه هر نمونه $x_j \in T$ را با استفاده الگوریتم FSA یا LDMDBA پیدا کنید.

۳. ماتریس وزن W_s و W_d برای کلاس مثبت و منفی با استفاده از روابط ۴-۶ و ۷-۴ تعریف می‌شود.

۴. ماتریس‌های قطری D ، Q و P با استفاده از روابط ۱۰-۴ و ۱۳-۴ بدست می‌آید.

۵. ابتدا ماتریس‌های ورودی $A \in \mathbb{R}^{m_1 \times n}$ و $B \in \mathbb{R}^{m_2 \times n}$ معین می‌شود. سپس ماتریس‌های S و R به این صورت $S = [K(B)e_1]$ و $R = [K(A)e_2]$ تعریف می‌گردد.

۶. پارامترهای c_1 ، c_2 و c_3 تعیین می‌شود. این پارامتر معمولاً براساس مجموعه داده صحت مشخص می‌شود.

۷. راه حل بهینه α و β به ترتیب با حل کردن مسائل ۴-۴ و ۴۴-۴ دوگان بدست می‌آید.

۸. مختصات دو ابرصفحه غیر موازی با حل کردن معادلات ۴۲-۴ و ۴۵-۴ تعیین می‌شود.

۹. کلاس نمونه جدید $x \in \mathbb{R}^n$ از طریق تابع تصمیم ۴۶-۴ مشخص می‌شود.

۱-۶-۴ پیچیدگی محاسباتی روش RKNN-TSVM

به طور کلی محاسبات در روش RKNN-TSVM شامل دو مرحله است:

۱. همانند روش TSVM، دو مسئله دوگان باید حل شود. با فرض این‌که تعداد نمونه‌های هر دو کلاس تقریباً برابر هستند ($m_1 \approx m_2$). بنابراین مرتبه زمانی حل کردن مسائل بهینه‌سازی در روش

پیشنهادی برابر با $O(2m^3)$ است.

۲. به منظور محاسبه ماتریس وزن‌ها، نزدیک‌ترین همسایه تمام نمونه‌های آموزشی باید پیدا شود. مرتبه زمانی این مرحله با استفاده از الگوریتم FSA برابر با $O(m^2 \log m)$ می‌شود. با این حال روش پیشنهادی برای کاهش پیچیدگی محاسباتی این مرحله، از الگوریتم LDMDBA استفاده می‌کند که مرتبه زمانی آن برابر با $O(\log nm \log m)$ است.

۲-۶-۴ مقایسه با سایر دسته‌بندهای مشابه ۱-۲-۶-۴ مقایسه با روش TSVM

روش پیشنهادی RKNN-TSVM اطلاعات شباهت بین نمونه‌ها را با استفاده از ساخت گراف نزدیک‌ترین همسایه در تابع هدف لحاظ می‌کند. بطوریکه ابرصفحه غیر موازی به نمونه‌های پرتراکم کلاس خود نزدیک‌تر می‌شود و از نمونه‌های حاشیه‌ای کلاس مقابله حداکثر فاصله را دارد. همچنین روش پیشنهادی با در نظر گرفتن نمونه‌های حاشیه‌ای در قید مسئله بهینه‌سازی، مرتبه زمانی را کاهش می‌دهد. روش TSVM خطای نمونه‌های آموزشی را کمینه می‌کند. بطوریکه احتمال رخداد برآش بیش از حد وجود دارد. با این حال روش RKNN-TSVM ریسک ساختاری را کمینه می‌کند. بنابراین دقت و تعمیم‌پذیری روش RKNN-TSVM بهتر از روش TSVM است.

۲-۲-۶-۴ مقایسه با روش WLTSVM

روش پیشنهادی همانند روش WLTSVM، دو گراف درون کلاسی و برون کلاسی جهت مدل کردن اطلاعات شباهت نمونه‌ها ایجاد می‌کند. با این حال روش پیشنهادی به نمونه‌ها براساس فاصله نزدیک‌ترین همسایه‌هایشان وزن می‌دهد. به عبارت دیگر، یک نمونه با همسایه‌های نزدیک‌تر وزن بیشتری نسبت به یک نمونه با همسایه‌های دورتر دریافت می‌کند. همچنین روش پیشنهادی نزدیک‌ترین همسایه‌های نمونه‌های آموزشی را با استفاده از الگوریتم LDMDBA بدست می‌آورد. بطوریکه سرعت آموزش روش پیشنهادی به طور قابل توجه‌ای بیشتر از روش WLTSVM است.

روش WLTSVM همانند روش TSVM خطای آموزشی را کمینه می‌کند. با این حال روش پیشنهادی ریسک ساختاری را در نظر می‌گیرد. بنابراین سه پارامتر c_1 ، c_2 و c_3 باید تنظیم شود. در حالی که تنها یک پارامتر c در روش WLTSVM در نظر گرفته می‌شود. شیوه جدید وزن‌دهی و در نظر گرفتن ریسک ساختاری، دقت روش پیشنهادی را بهتر از روش WLTSVM می‌کند.

۳-۶-۴ مقایسه با روش KNN-STSVM

روش KNN-STSVM همانند روش WLTSVM بر اساس شمارش تعداد همسایه‌های نزدیک، به هر نمونه وزن می‌دهد. برخلاف روش پیشنهادی، روش KNN-STSVM خطای نمونه‌های آموزشی را کمینه می‌کند. همچنین روش KNN-STSVM با استفاده از خوشبندی سلسله مراتبی، توزیع داده‌ها را در تابع هدف لحاظ می‌کند. به طور کلی روش KNN-STSVM سه گام محاسباتی دارد:

۱. خوشها با استفاده از الگوریتم Ward مشخص می‌شوند.

۲. نزدیک‌ترین همسایه‌های تمام نمونه‌های آموزشی محاسبه می‌شود.

۳. دو مسئله بهینه‌سازی دوگان حل می‌گردد.

با در نظر گرفتن سه گام محاسباتی بالا، پیچیدگی زمانی کلی روش KNN-STSVM برابر با $O(1/4m^3 + n(m^{\frac{3}{2}} + m^{\frac{1}{2}}) + m^{\frac{1}{2}} \log m)$ است. بطوریکه این روش برای مجموعه داده‌های بزرگ مناسب نمی‌باشد.

۳-۶-۴ محدودیت‌های روش RKNN-TSVM

با وجود اینکه روش پیشنهادی دقیق و مرتبه زمانی بهتری نسبت به روش‌های مشابه مانند WLTSVM دارد، محدودیت‌های زیر را هم دارد:

۱. معکوس ماتریس در روش RKNN-TSVM اجتناب ناپذیر است. از طرفی مرتبه زمانی معکوس ماتریس برابر با $O(m^3)$ می‌باشد. بطوریکه پیچیدگی محاسباتی با افزایش ابعاد ماتریس به طور قابل توجهی زیاد می‌شود.

۲. مصرف حافظه روش RKNN-TSVM برای مجموعه داده‌های بزرگ بسیار زیاد است. زیرا دو گراف نزدیک‌ترین همسایه باید در حافظه ذخیره گردد.

۳. در روش RKNN-TSVM، چهار پارامتر c_1, c_2, c_3 و k جهت بهبود دقت مدل باید تنظیم شوند. بنابراین جستجو برای پیدا کردن پارامترهای بهینه در این روش هزینه محاسباتی زیادی دارد. به منظور کاهش بار محاسباتی، در بخش آزمایش‌ها پارامتر $c_3 = c_2$ شده است.

۴-۶-۴ مقیاس‌پذیری روش RKNN-TSVM

روش پیشنهادی همانند WLTSVM بردار f_j را در قید مسئله بهینه‌سازی لحاظ کرده است. بطوریکه فقط نمونه‌های حاشیه‌ای به جای کل نمونه‌های کلاس مقابل در قید ظاهر می‌شوند. بدین ترتیب مرتبه زمانی

حل کردن مسئله دوگان کاهش می‌یابد. با این حال روش پیشنهادی در مقایسه با روش WLTSVM، برای مجموعه داده‌های بزرگ مناسب‌تر است. زیرا روش پیشنهادی از الگوریتم LDMDBA برای پیدا کردن نزدیک‌ترین همسایه تمام نمونه‌ها استفاده می‌کند. این الگوریتم سرعت آموزش روش RKNN-TSVM را برای مجموعه داده‌های بزرگ به طور قابل توجه‌ای بهبود می‌دهد.

٧-٤ الگوریتم بهینه‌سازی clipDCD

در روش RKNN-TSVM، چهار مسئله دوگان محدب وجود دارد: ۴-۴، ۲۶-۴، ۲۴-۴ و ۴۴-۴. این مسائل بهینه‌سازی را می‌توان به صورت فرم دوگان زیر بازنویسی کرد.

$$\begin{aligned} \min_{\alpha} \quad & f(\alpha) = \frac{1}{2} \alpha^T Q \alpha - e^T \alpha, \\ \text{s.t.} \quad & 0 \leq \alpha \leq c. \end{aligned} \quad (47-4)$$

در رابطه ٤-٤، ماتریس $Q \in \mathbb{R}^{n \times n}$ معین مثبت است. برای مثال، این ماتریس می‌تواند با عبارت $(F^T S)(R^T D R + c_2 I)^{-1}(S^T F)$ جایگزین شود. به منظور حل کردن مسئله دوگان محدب، الگوریتم‌های مختلفی طراحی و معرفی شده است. برخی از این الگوریتم‌ها شامل روش‌های نقطه داخلی^٦، روش^٧ SOR [٣٩] و الگوریتم^٨ DCD^٩ [٤٠] هستند. در این پژوهش، الگوریتم clipDCD^٩ [٤١] جهت بهبود سرعت آموزش روش RKNN-TSVM بکار گرفته می‌شود. الگوریتم clipDCD براساس روش گرادیان نزولی می‌باشد. این الگوریتم یک مسئله تک متغیره را حل می‌کند. بطوریکه یک متغیر براساس رویکرد بیشترین کاهش ممکن^{١٠} تغییر می‌کند.

این الگوریتم برخلاف روش DCD، هیچ تکرار داخلی و خارجی^{١١} ندارد. بطوریکه در هر تکرار فقط یک عنصر بردار α تغییر می‌کند. نحوه تغییر این بردار به این صورت $\alpha_L \rightarrow \alpha_L + \lambda$ نشان داده می‌شود ($L \in \{1, \dots, m\}$). تابع هدف در الگوریتم clipDCD به صورت زیر تعریف می‌شود.

$$f(\lambda) = f(0) + \frac{1}{2} \lambda^2 Q_{LL} - \lambda(e_L - \alpha^T Q_{:,L}). \quad (48-4)$$

در رابطه ٤-٤، $Q_{:,L}$ نشان‌دهنده ستون L ماتریس است. با مشتق‌گیری نسبت به λ رابطه زیر

⁶Interior-point

⁷Successive Overrelaxation

⁸Dual Coordinate Descent

⁹Clipping Dual Coordinate Descent

¹⁰Maximal possibility-decrease

¹¹Inner and outer iteration

بدست می‌آید.

$$\frac{df(\lambda)}{d\lambda} = \bullet \Rightarrow \lambda = \frac{(e_L - \alpha^T Q_{.,L})}{Q_{LL}} \quad (49-4)$$

بیشترین کاهش ممکن روی تابع هدف با انتخاب اندیس L بدست می‌آید.

$$L = \arg \max_{i \in S} \left\{ \frac{(e_i - \alpha^T Q_{.,i})^2}{Q_{ii}} \right\}, \quad (50-4)$$

اندیس L از مجموعه S انتخاب می‌شود. این مجموعه به صورت زیر تعریف می‌گردد.

$$S = \left\{ i : \alpha_i > \bullet \text{ if } \frac{e_i - \alpha^T Q_{.,i}}{Q_{ii}} < \bullet \text{ or } \alpha_i < c \text{ if } \frac{e_i - \alpha^T Q_{.,i}}{Q_{ii}} > \bullet \right\}. \quad (51-4)$$

شرط توقف الگوریتم clipDCD به صورت زیر تعریف می‌شود.

$$\frac{(e_L - \alpha^T Q_{.,L})^2}{Q_{LL}} < \epsilon, \quad \epsilon > \bullet \quad (52-4)$$

در رابطه ۵۲-۴، ϵ نشان دهنده آستانه توقف که یک عدد کوچک مثبت $10^{-5} = \epsilon$ است. اثبات همگرایی الگوریتم clipDCD در مقاله اصلی [۴۱] ذکر شده است. مراحل حل کردن یک مسئله دوگان ۴۷-۴ در الگوریتم ۴-۴ بیان شده است.

الگوریتم ۴-۴: الگوریتم بهینه‌سازی clipDCD

ابتدا فرض گرفته می‌شود که ماتریس Q و پارامتر c را در اختیار داریم.

۱. مقادیر اولیه بردار α برابر با $\bullet = \alpha$ در نظر گرفته می‌شود.

۲. اندیس L با استفاده از رابطه ۵۰-۴ و ۵۱-۴ بدست می‌آید. سپس λ از طریق رابطه ۴۹-۴ محاسبه می‌شود.

۳. بردار α_L به صورت $\alpha_L^{new} \leftarrow \alpha_L + \max\{\bullet, \min\{\lambda, c\}\}$ تغییر می‌کند.

۴. در صورتی که شرط توقف یعنی $\epsilon < \frac{(e_L - \alpha^T Q_{.,L})^2}{Q_{LL}}$ برقرار باشد، الگوریتم خاتمه می‌یابد. در غیر این صورت الگوریتم از گام ۲ ادامه می‌یابد.

۸-۴ جمع‌بندی

در این فصل، روش ماشین بردار پشتیبان دو قلو مبتنی بر رگولارسیون و نزدیک‌ترین همسایه (RKNN-TSVM) ارائه شد. روش پیشنهادی سه مزیت نسبت به روش WLTSVM دارد:

۱. شیوه وزن‌دهی به نمونه‌ها براساس فاصله نزدیک‌ترین همسایه‌هایشان می‌باشد. بطوریکه ابرصفحه

غیر موازی به نمونه‌های پرتراکم نزدیک‌تر می‌شود. در نهایت مدل خروجی مقاومت بهتری نسبت به نمونه‌های نویزی و پرت دارد.

۲. روش RKNN-TSVM ریسک ساختاری را کمینه می‌کند. دو پارامتر جدید به مسائل بهینه‌سازی اضافه شده است. این پارامترها تعادل بین ریسک تجربی و جمله رگولارسیون را کنترل می‌کنند.

بطوریکه دقت و تعمیم‌پذیری مدل خروجی افزایش می‌یابد.

۳. روش پیشنهادی از الگوریتم LDMDBA برای ساخت گراف نزدیک‌ترین همسایه استفاده می‌کند.

نه تنها این الگوریتم سرعت آموزش روش پیشنهادی را به طور قابل توجه‌ای افزایش می‌دهد بلکه دقت مدل خروجی را هم بهبود داده است.

در بخش ۳-۵ روش RKNN-TSVM مورد ارزیابی و بررسی قرار گرفته است. همچنین مقاله مستخرج از دسته‌بند RKNN-TSVM برای مجله زیر ارسال شده است.

- An enhanced KNN-based twin support vector machine with stable learning rules, Neural Computing and Applications (Under review)

فصل ۵

نتایج و ارزیابی

۱-۵ مقدمه

در این فصل دو روش KNN-LSTSVM و RKNN-TSVM به ترتیب در بخش های ۲-۵ و ۳-۵ به صورت جامع بررسی و ارزیابی می شود. این روش ها در دو بخش جداگانه مورد بررسی قرار گرفته اند. زیرا ایده و هدف اصلی این دو روش با یکدیگر متفاوت است.

۱. هدف اصلی در روش KNN-LSTSVM ضمن حفظ مزیت اصلی روش LSTSVM، اضافه کردن گراف نزدیک ترین همسایه به منظور بهبود دقت این مدل است.

۲. روش RKNN-TSVM با کمینه کردن ریسک ساختاری و شیوه جدید وزن دهی، دقت روش WLTSVM را بهبود می دهد. با این حال در روش RKNN-TSVM، افزایش سرعت آموزش هم مد نظر است.

در ادامه عملکرد دو دسته بند پیشنهادی از نظر دقت و سرعت روی مجموعه داده های مختلف سنجیده می شود.

۲-۵ ارزیابی روش KNN-LSTSVM

ابتدا نحوه پیاده سازی و انتخاب پارامترهای بهینه توضیح داده شده است. سپس عملکرد روش-KNN LSTSVM روی مجموعه داده های مصنوعی بررسی می شود. بطوریکه ناحیه تصمیم خطی و غیر خطی این روش در فضای دو بعدی نشان داده شده است. در آخر نتایج روی مجموعه داده های UCI و NDC مورد بحث قرار می گیرد.

۱-۲-۵ نحوه پیاده‌سازی و اجرای الگوریتم‌ها

تمام روش‌ها در زبان برنامه‌نویسی پایتون^۱ [۴۲] پیاده‌سازی شده است. آزمایش‌ها روی یک کامپیوتر شخصی با پردازنده Core i7 6700K، سیستم عامل Windows 8 و ۳۲ گیگابایت حافظه صورت گرفته است. اعمال جبر خطی با کتابخانه NumPy [۴۳] انجام شده است. همچنین کتابخانه SciPy [۴۴] برای محاسبه فاصله و توابع آماری بکار گرفته شده است. همچنین الگوریتم بهینه‌سازی clipDCD جهت بهبود سرعت اجرا، در Cython [۴۵] پیاده‌سازی شده است.

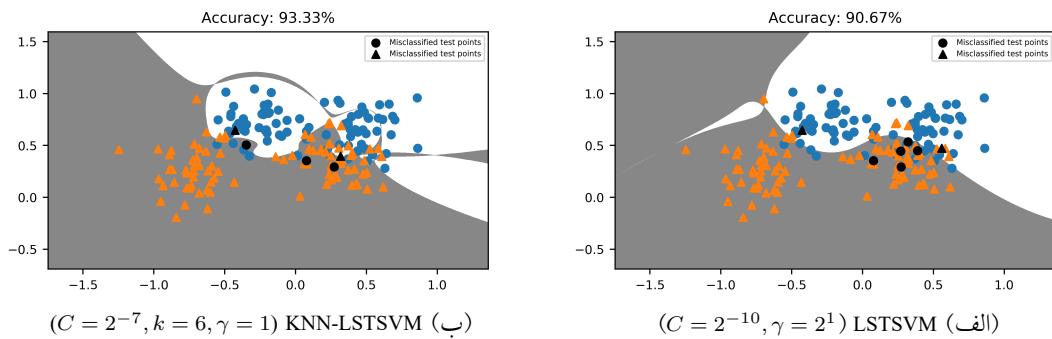
دقت روش TSVM و گسترش‌هایش به انتخاب پارامترهای بهینه بسیار وابسته است. بدین منظور جستجوی شبکه‌ای^۲ برای پیدا کردن پارامترهای بهینه استفاده می‌شود. همچنین تابع TSVM را به عنوان تابع هسته $k(x_i, x_j) = \exp(-\|x_i - x_j\|^2/\gamma^2)$ اغلب بکار می‌گیرند. پارامترهای خطا در روش‌های TSVM، WLTSVM و LSTSVM از مجموعه $\{2^i \mid i = -10, -9, \dots, 9, 10\}$ انتخاب شده است. پارامتر تابع هسته γ نیز از مجموعه $\{2^i \mid i = -15, -14, \dots, 5\}$ تعیین می‌شود. تعداد نزدیک‌ترین همسایه k از مجموعه $\{2, 3, \dots, 10\}$ انتخاب می‌گردد.

۲-۲-۵ مجموعه داده مصنوعی

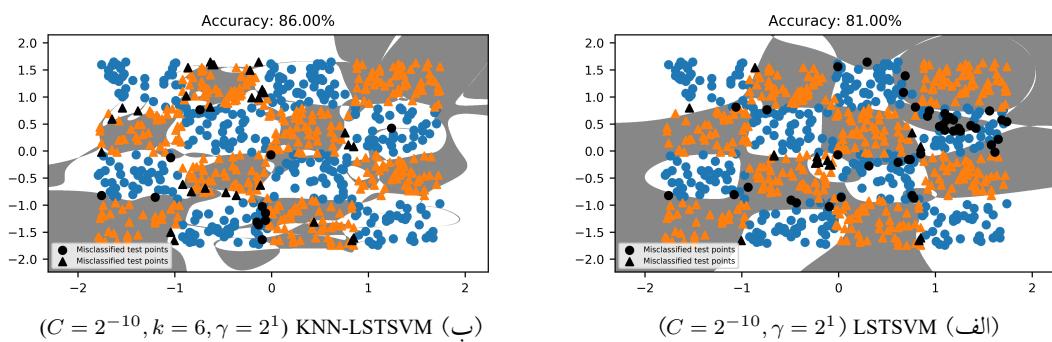
به منظور نشان دادن برتری روش KNN-LSTSVM نسبت به روش LSTSVM به صورت هندسی، آزمایش روی مجموعه داده Ripley [۴۶] و Checkerboard [۴۷] انجام گرفته است که به ترتیب شامل ۲۵۰ و ۱۰۰۰ نمونه آموزشی هستند. شکل ۱-۵ و شکل ۲-۵ عملکرد و ناحیه تصمیم روش LS-TSVM و KNN-LSTSVM را به ترتیب برای روی مجموعه داده Ripley و Checkerboard نشان می‌دهد. همانطور که در شکل ۱-۵ و شکل ۲-۵ نشان داده شده است، روش پیشنهادی KNN-LSTSVM دقت بیشتر و ناحیه تصمیم بهتری دارد. بطوریکه تعداد نمونه‌های تست کمتری به طور اشتباه دسته‌بندی شده‌اند. به عبارت دیگر، تعمیم‌پذیری روش KNN-LSTSVM بیشتر است.

¹Python

²Grid search



شکل ۱-۵: عملکرد و ناحیه تصمیم روشن KNN-LSTSVM و LS-TSVM را برای روی داده Ripley می‌نماید.



شکل ۲-۵: عملکرد و ناحیه تصمیم روشن KNN-LSTSVM و LS-TSVM را برای روی داده Checkerboard می‌نماید.

۳-۲-۵ نتایج ارزیابی بر روی مجموعه داده UCI

در این زیر بخش، عملکرد روش KNN-LSTSVM روی ۱۴ مجموعه داده از مخزن UCI^۳ ارزیابی و بررسی می‌شود. مشخصات این مجموعه داده‌ها در جدول ۱-۵ ذکر شده است.

دقت دسته‌بندی هر کدام از روش‌ها توسط معیار ارزیابی اعتبارسنج ضربدری ۱۰-اتایی سنجیده می‌شود. بطوریکه نمونه‌های آموزشی به صورت تصادفی به ۱۰ بخش تقسیم می‌شوند. یکی از این بخش‌ها به عنوان نمونه‌های تست در نظر گرفته می‌شود و سایر بخش‌ها برای آموزش دسته‌بند استفاده می‌گردد. این فرآیند ۱۰ بار تکرار می‌شود تا زمانی که تمام بخش‌ها به عنوان نمونه تست استفاده شود

.[۴۸]

جدول ۲-۵ میانگین و انحراف معیار دقت (به درصد) را برای دسته‌بندی‌های TSVM، WLTSVM و KNN-LSTSVM نشان می‌دهد. همچنین زمان آموزش دسته‌بندی‌ها (به ثانیه) و مقادیر بهینه پارامترها نیز در جدول ۲-۵ درج شده است. لازم به ذکر است که محاسبه گراف نزدیک‌ترین همسایه

³<https://archive.ics.uci.edu/ml/index.php>

جدول ۱-۵: مشخصات مجموعه داده‌ها برای ارزیابی روش KNN-LSTSVM

مجموعه داده	تعداد نمونه‌ها	نمونه‌های منفی	نمونه‌های مثبت	تعداد ویژگی‌ها
Austrailian	۶۹۰	۳۰۷	۳۸۳	۱۴
Bupa-Liver	۳۴۵	۱۴۵	۲۰۰	۶
Cleveland	۳۰۳	۱۳۹	۱۶۴	۱۳
Haber-Man	۳۰۶	۲۲۵	۸۱	۳
Heart-Statlog	۲۷۰	۱۲۰	۱۵۰	۱۳
Hepatitis	۱۵۵	۳۲	۱۲۳	۱۹
Ionsphere	۳۵۱	۲۲۵	۱۲۶	۳۴
Monk3	۵۵۴	۲۸۸	۲۶۶	۶
Pima-Indian	۷۶۸	۲۶۸	۵۰۰	۸
Sonar	۲۰۸	۹۷	۱۱۱	۶۰
Titanic	۸۹۱	۳۴۲	۵۴۹	۷
Votes	۴۳۵	۲۶۷	۱۶۸	۱۶
Wdbc	۵۶۹	۲۱۲	۳۵۷	۳۰
Wpbc	۱۹۸	۴۷	۱۵۱	۳۳

در زمان آموزش دسته‌بندی‌های WLTSVM و KNN-LSTSVM لحاظ شده است.

نتایج بدست آمده به این صورت تحلیل و بررسی می‌شود:

۱. از نظر دقیق دسته‌بندی، روش پیشنهادی (KNN-LSTSVM) از سه روش دیگر بهتر عمل کرده است.

زیرا روش KNN-LSTSVM با استفاده از گراف نزدیک‌ترین همسایه، اطلاعات شباهت نمونه‌ها را در تابع هدف لحاظ می‌کند. از این رو ابرصفحه غیر موازی در روش پیشنهادی به نمونه‌های پرترکم نزدیک‌تر و از نمونه‌های حاشیه‌ای حداقل فاصله را می‌گیرد. در حالی که روش‌های LSTSVM و TSVM نمونه‌های حاشیه‌ای را در نظر نمی‌گیرند.

۲. از نظر زمان آموزش و پیچیدگی محاسباتی، روش LSTSVM از سایر روش‌ها سریع‌تر است. زیرا

در این روش فقط دو دستگاه معادلات خطی حل می‌گردد. با این حال روش پیشنهادی (KNN-LSTSVM) از روش TSVM و WLTSVM به طور قابل توجه‌ای سریع‌تر است. زیرا روش پیشنهادی دو دستگاه معادلات خطی حل می‌کند، در حالی که روش TSVM و WLTSVM دو مسئله دوگان درجه دو را حل می‌کنند.

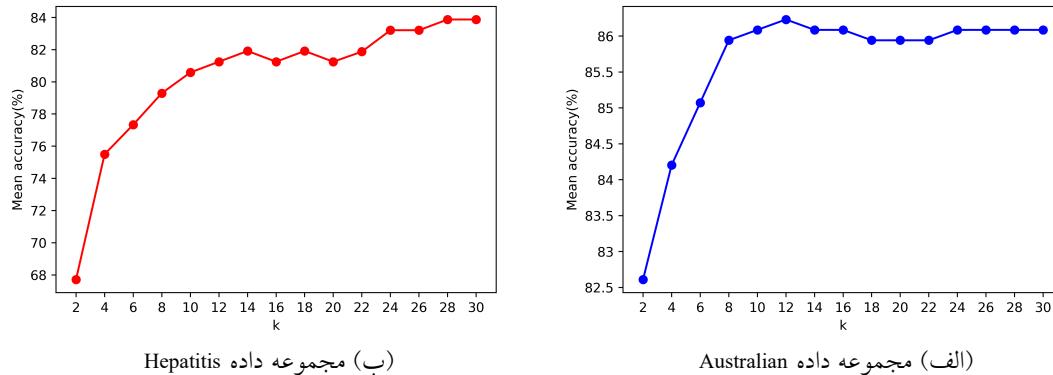
۳. به منظور بررسی اثر پارامتر k روی دقت روش KNN-LSTSVM، آزمایش روی مجموعه داده‌های

جدول ۲-۵: مقایسه دقت و زمان آموزش دسته‌بندهای LSTSVM، WLTSVM، TSVM و KNN-LSTSVM

KNN-LSTSVM			LSTSVM			WLTSVM			TSVM			Mجموعه داده
زمان اجرا	دقت (%)	(C, γ, k)	زمان اجرا	دقت (%)	(C_1, C_2, γ)	زمان اجرا	دقت (%)	(C, γ, k)	زمان اجرا	دقت (%)	(C_1, C_2, γ)	($m \times n$)
۰/۲۳۰	۸۷/۳۹±۳/۳۱	(۲۳, ۲-۱۳, ۷)	۰/۰۵۸	۸۷/۶۸±۳/۷۹	(۲-۷, ۲-۵, ۲-۶)	۰/۲۶۴	۸۷/۲۵±۳/۶۵	(۲-۱, ۲-۱۴, ۶)	۰/۰۷۰	۸۷/۹۸±۲/۷۵	(۲-۴, ۲-۴, ۲-۶)	Australian (۶۹۰ × ۱۴)
۰/۰۴۱	۷۵/۹۶±۵/۴۰	(۲۱, ۲-۶, ۷)	۰/۰۰۷	۷۴/۵۱±۶/۸۹	(۲۵, ۲۴, ۲-۶)	۰/۱۹۳	۷۴/۸۲±۳/۸۴	(۲۲, ۲-۷, ۴)	۰/۰۸۲	۷۳/۶۲±۴/۷۷	(۲-۳, ۲-۳, ۲-۶)	Bupa-Liver (۳۴۵ × ۶)
۰/۰۲۶	۸۵/۵۱±۶/۴۱	(۲-۲, ۲-۱۵, ۵)	۰/۰۰۵	۸۵/۴۹±۴/۹۲	(۲۲, ۲۱, ۲-۱۱)	۰/۰۸۲	۸۴/۴۸±۶/۲۹	(۲۳, ۲-۹, ۷)	۰/۰۳۷	۸۴/۸۶±۳/۷۶	(۲, ۲-۱, ۲-۱۱)	Cleveland (۳۰۳ × ۱۳)
۰/۰۳۱	۷۶/۸۱±۵/۸۲	(۲-۱, ۲-۶, ۹)	۰/۰۰۵	۷۶/۷۳±۷/۸۳	(۲۳, ۲۹, ۲-۷)	۰/۱۳۸	۷۵/۸۰±۵/۲۲	(۲۳, ۲-۴, ۲)	۰/۰۸۲	۷۶/۱۴±۳/۶۰	(۲۱, ۲-۱, ۲-۶)	Haber-Man (۳۰۶ × ۳)
۰/۰۲۲	۸۵/۱۹±۵/۷۴	(۲-۱, ۲-۱۴, ۷)	۰/۰۰۴	۸۵/۱۹±۶/۲۰	(۲۰, ۲۱, ۲-۱۱)	۰/۰۳۲	۸۵/۱۹±۵/۴۹	(۲۱, ۲-۱۳, ۶)	۰/۰۴۱	۸۵/۵۶±۵/۸۴	(۲۱, ۲۰, ۲-۱۱)	Heart-Statlog (۲۷۰ × ۱۳)
۰/۰۰۶	۸۷/۱۳±۷/۵۹	(۲-۱, ۲-۵, ۷)	۰/۰۰۱	۸۷/۷۹±۶/۵۷	(۲۳, ۲۵, ۲-۱۱)	۰/۰۴۰	۸۶/۵۰±۸/۷۹	(۲۸, ۲-۳, ۳)	۰/۰۰۳	۸۵/۷۹±۸/۶۵	(۲-۴, ۲-۲, ۲-۸)	Hepatitis (۱۵۵ × ۱۹)
۰/۰۴۰	۹۲/۵۹±۴/۴۷	(۲۱, ۲-۵, ۵)	۰/۰۰۶	۹۱/۷۴±۴/۳۲	(۲۳, ۲۱, ۲-۵)	۰/۱۲۶	۹۲/۰۴±۵/۵۰	(۲۱, ۲۱, ۳)	۰/۰۲۸	۹۲/۵۹±۳/۱۹	(۲-۱, ۲-۳, ۲-۵)	Ionsphere (۳۵۱ × ۳۴)
۰/۱۲۶	۹۸/۵۶±۱/۳۴	(۲۰, ۲-۳, ۵)	۰/۰۳۲	۹۸/۵۵±۱/۳۶	(۲-۶, ۲-۳, ۲-۳)	۰/۴۹۴	۹۸/۳۸±۱/۶۹	(۲۳, ۲-۶, ۲)	۰/۱۵۰	۹۸/۳۷±۱/۲۶	(۲-۴, ۲۲, ۲-۳)	Monk3 (۵۵۴ × ۶)
۰/۳۳۹	۷۸/۰۱±۳/۶۴	(۲-۱, ۲-۴, ۱۰)	۰/۰۷۳	۷۷/۶۱±۵/۸۹	(۲-۱, ۲-۱, ۲-۴)	۰/۵۶۶	۷۷/۸۶±۳/۴۹	(۲۴, ۲-۵, ۲)	۰/۱۴۷	۷۷/۸۷±۴/۷۳	(۲۱, ۲۱, ۲-۳)	Pima-Indian (۷۶۸ × ۸)
۰/۰۱۱	۸۷/۴۸±۶/۶۵	(۲-۴, ۲-۶, ۴)	۰/۰۰۲	۸۵/۵۵±۸/۳۱	(۲-۱, ۲۳, ۲-۳)	۰/۰۳۳	۸۷/۵۰±۳/۱۷	(۲۰, ۲-۴, ۵)	۰/۰۱۲	۸۶/۱۴±۸/۳۵	(۲-۵, ۲-۱, ۲-۳)	Sonar (۲۰۸ × ۶۰)
۰/۴۸۶	۸۲/۲۷±۳/۸۰	(۲۸, ۲-۵, ۱۰)	۰/۱۰۸	۸۲/۳۸±۴/۶۳	(۲۱, ۲۱, ۲-۵)	۰/۶۹۲	۸۱/۴۹±۴/۷۰	(۲۰, ۲-۶, ۷)	۰/۲۲۵	۸۱/۹۴±۳/۲۳	(۲-۴, ۲-۵, ۲-۳)	Titanic (۸۹۱ × ۷)
۰/۰۶۴	۹۷/۰۱±۳/۱۱	(۲۶, ۲-۹, ۳)	۰/۰۱۲	۹۷/۰۲±۲/۸۹	(۲-۶, ۲-۳, ۲-۹)	۰/۰۹۸	۹۷/۰۱±۱/۷۹	(۲۷, ۲-۱۳, ۷)	۰/۱۴۵	۹۶/۷۸±۲/۱۱	(۲-۶, ۲-۳, ۲-۶)	Votes (۴۳۵ × ۱۶)
۰/۱۵۴	۹۷/۷۲±۱/۱۲	(۲-۵, ۲-۷, ۵)	۰/۰۳۴	۹۸/۰۷±۲/۵۴	(۲-۴, ۲-۲, ۲-۸)	۰/۲۰۶	۹۷/۵۴±۲/۵۱	(۲-۳, ۲-۵, ۶)	۰/۰۰۳	۹۸/۴۲±۱/۲۳	(۲-۳, ۲-۱, ۲-۸)	Wdbc (۵۶۹ × ۳۰)
۰/۰۱۰	۸۲/۷۶±۵/۴۸	(۲-۳, ۲-۷, ۶)	۰/۰۰۲	۸۱/۳۲±۹/۰۱	(۲-۴, ۲-۴, ۲-۷)	۰/۰۱۹	۸۰/۸۴±۸/۹۴	(۲۱, ۲-۱۰, ۲)	۰/۰۰۶	۸۲/۳۹±۹/۲۰	(۲-۴, ۲-۵, ۲-۶)	Wpbc (۱۹۸ × ۳۳)
میانگین دقت			۸۶/۷۴	۸۶/۴۰		۸۶/۱۹			۸۶/۳۱			

Australian و Hepatitis صورت گرفته است. مقادیر پارامتر k بین ۲ تا ۳۰ با گام ۲ تعیین شده است.

شكل ۳-۵ اهمیت انتخاب پارامتر را نشان می‌دهد. بیشترین دقت برای مجموعه داده Australian با پارامتر $k = 12$ بدست آمده است. به عبارت دیگر، پارامتر بهینه برای مجموعه داده Australian از طرف دیگر، افزایش مقدار k برای مجموعه داده Hepatitis باعث بهبود دقت روش KNN-LSTSVM می‌شود. به طور کلی، انتخاب بهینه پارامتر k روی عملکرد روش پیشنهادی تاثیر زیادی دارد.

شکل ۳-۵: اثر افزایش پارامتر k روی دقت دسته‌بند KNN-LSTSVM

۱-۳-۲-۵ بررسی آماری

به منظور تحلیل بیشتر عملکرد چهار روش روی ۱۴ مجموعه داده (۲-۵)، آزمون‌های آماری طبق پیشنهاد دمسار^۴ [۴۹] استفاده می‌شود. بدین منظور آزمون آماری ساده و ناپارامتریک فریدمن^۵ مورد استفاده قرار گرفته است. به منظور انجام آزمون آماری، میانگین رتبه چهار روش بر اساس دقت محاسبه شده و در جدول ۳-۵ نشان داده شده است. ابتدا بر اساس فرض صفر، تمام روش‌ها را یکسان در نظر می‌گیریم. سپس آزمون فریدمن با رابطه زیر محاسبه می‌شود.

$$\chi_F^2 = \frac{12N}{k(k+1)} \left[\sum_j R_j^2 - \frac{k(k+1)^2}{4} \right], \quad (1-5)$$

در رابطه ۱-۵، r_i^j نشان‌دهنده رتبه زمین روش بر روی i -مین مجموعه داده از N است.

سپس مقدار F_F براساس χ_F^2 به صورت زیر محاسبه می‌شود. بطوريکه F_F از توزيع F با $(1-k)$ و درجه آزادی p می‌کند.

$$F_F = \frac{(N-1)\chi_F^2}{N(k-1) - \chi_F^2} \quad (2-5)$$

بر اساس روابط ۱-۵ و ۲-۵، مقادیر $F_F = ۳/۱۹۸$ و $\chi_F^2 = ۸/۲۹۳$ بدست آمده است. در اینجا F_F از توزيع F با $(3, 39)$ درجه آزادی پیروی می‌کند. مقادیر ویژه $F(3, 39)$ برای سطوح معناداری 0.05 و 0.01 به ترتیب برابر 2.23 و 4.2 است. مقدار F_F به طور قابل توجه‌ای بیشتر از مقدار

⁴Demsar⁵Friedman

جدول ۵-۳: میانگین رتبه براساس دقت (ارزیابی KNN-LSTSVM)

Mجموعه داده	KNN-LSTSVM	LSTSVM	WLTSVM	TSVM	
۳	۲	۴	۱	Australian	
۱	۳	۲	۴	Bupa-Liver	
۱	۲	۴	۳	Cleveland	
۱	۲	۴	۳	Haber-Man	
۳	۳	۳	۱	Heart-Statlog	
۲	۱	۳	۴	Hepatitis	
۱/۵	۴	۳	۱/۵	Ionsphere	
۱	۲	۳	۴	Monk3	
۱	۴	۳	۲	Pima-Indian	
۲	۴	۱	۳	Sonar	
۲	۱	۴	۳	Titanic	
۲/۵	۱	۲/۵	۴	Votes	
۳	۲	۴	۱	Wdbc	
۱	۳	۴	۲	Wpbc	
۱/۷۹	۲/۴۳	۳/۱۸	۲/۶۱	میانگین رتبه	
<hr/>					

ویژه است. بنابراین از آزمون آماری استنتاج می‌شود که تفاوت قابل توجه‌ای بین ۴ روش وجود دارد. همچنین جدول ۵-۵ نشان می‌دهد که روش پیشنهادی (KNN-LSTSVM) در مجموع عملکرد بهتری نسبت به سایر روش‌ها دارد. زیرا میانگین رتبه روش پیشنهادی در میان سایر روش‌ها کمترین است.

۴-۲-۵ مجموعه داده NDC

به منظور بررسی سرعت آموزش روش پیشنهادی (RKNN-TSVM) و مقایسه آن با سایر روش‌ها، آزمایش بر روی مجموعه داده‌های بزرگ صورت گرفته است. مجموعه داده NDC [۵۰] برای این منظور انتخاب شده است. جدول ۴-۵ مشخصات این مجموعه داده را نشان می‌دهد.

جهت آزمایش با مجموعه داده NDC، پارامتر C برای تمام روش برابر با یک است. در نسخه غیر خطی از تابع RBF با پارامتر $\gamma = 2^{-15}$ استفاده شده است. همچنین پارامتر k برای روش‌های WLTSVM و KNN-LSTSVM برابر با ۵ است. جدول ۵-۵ زمان آموزش چهار روش بر روی مجموعه داده NDC با تابع هسته خطی و RBF را نشان می‌دهد.

روش پیشنهادی (KNN-LSTSVM) چندین برابر سریع‌تر از روش WLTSVM روی تمام مجموعه داده‌های NDC ظاهر شده است. زیرا روش پیشنهادی نیازی به الگوریتم‌های حل مسائل بهینه‌سازی ندارد، در حالی‌که روش WLTSVM با استفاده از الگوریتم clipDCD پیاده‌سازی شده است. همانطور که

جدول ۵-۵: مشخصات مجموعه داده NDC

مجموعه داده	تعداد نمونه‌های آموزش	تعداد نمونه‌های تست	تعداد ویژگی‌ها
۳۲	۵۰	۵۰۰	NDC-500
۳۲	۷۰	۷۰۰	NDC-700
۳۲	۹۰	۹۰۰	NDC-900
۳۲	۱۰۰	۱۰۰۰	NDC-1K
۳۲	۲۰۰	۲۰۰۰	NDC-2K
۳۲	۳۰۰	۳۰۰۰	NDC-3K
۳۲	۴۰۰	۴۰۰۰	NDC-4K
۳۲	۵۰۰	۵۰۰۰	NDC-5K
۳۲	۱۰۰۰	۱۰۰۰۰	NDC-10K
۳۲	۲۵۰۰	۲۵۰۰۰	NDC-25K
۳۲	۵۰۰۰	۵۰۰۰۰	NDC-50K

در جدول ۵-۵ مشخص است، روش KNN-LSTSVM سریع‌تر از روش LSTSVM نمی‌باشد. چون در روش پیشنهادی علاوه بر حل کردن دو دستگاه معادلات خطی، گراف نزدیک‌ترین همسایه نیز برای تمام نمونه‌ها محاسبه می‌شود.

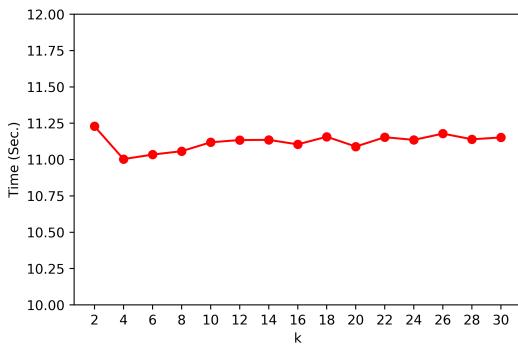
رویکرد تابع هسته مستطیلی با 10×10 درصد از نمونه‌ها در نسخه غیر خطی استفاده شده است. نتایج نسخه غیر خطی نشان می‌دهد که روش KNN-LSTSVM و LSTSVM از روش TSVM و WLTSVM بسیار سریع‌تر هستند. زیرا حتی با تابع هسته تقلیل یافته ($m \times \bar{m}$), در روش TSVM و WLTSVM دو مسئله دوگان حل می‌شود.

به منظور نشان دادن تاثیر پارامتر k بر روی زمان آموزش روش KNN-LSTSVM، یک آزمایش بر روی مجموعه داده بزرگ NDC-10K انجام گرفته است. همانطور که در شکل ۴-۵ نشان داده شده است، افزایش پارامتر k روی زمان آموزش روش پیشنهادی تاثیر بسیار اندکی دارد. به طور خلاصه، نتایج روی مجموعه داده NDC نشان می‌دهد که روش KNN-LSTSVM نسبت به TSVM و WLTSVM برای مجموعه داده‌های بزرگ مناسب‌تر است.

جدول ۵-۵: مقایسه زمان آموزش روش KNN-LSTSVM و سایر روش‌ها بر روی مجموعه داده NDC

NDC	KNN-LSTSVM			LSTSVM			WLTSVM			TSVM		
	زمان اجرا	خطی	RBF	زمان اجرا	خطی	RBF	زمان اجرا	خطی	RBF	زمان اجرا	خطی	RBF
۰/۳۹۹	۰/۰۰۳۱	۰/۰۳۴	۰/۰۰۰۴	۰/۰۸۳۸	۰/۰۵۸۳	۰/۰۰۰۷	۰/۰۲۲۶	۰/۰۳۰۷	۰/۰۲۲۶	۰/۰۲۲۶	۰/۰۳۰۷	NDC-500
۰/۷۵۱	۰/۰۰۵۳	۰/۰۶۴	۰/۰۰۰۴	۰/۰۷۴۱	۰/۱۱۵	۰/۰۰۰۷	۰/۰۷۴۴	۰/۰۴۵	۰/۰۷۴۴	۰/۰۷۴۴	۰/۰۷۴۴	NDC-700
۱/۳۶۲	۰/۰۰۸۴	۰/۰۲۲۳	۰/۰۰۰۴	۰/۰۷۶۱	۰/۱۸۸۴	۰/۰۰۰۷	۰/۱۱۱	۰/۰۸۳	۰/۰۸۳	۰/۰۸۳	۰/۰۸۳	NDC-900
۱/۶۵۱	۰/۰۱	۰/۰۲۶۹	۰/۰۰۰۴	۰/۰۴۹۹	۰/۰۳۱۵	۰/۰۰۰۷	۰/۰۵۹۹	۰/۰۹۷۶	۰/۰۹۷۶	۰/۰۹۷۶	۰/۰۹۷۶	NDC-1K
۰/۲۱	۰/۰۳۸۷	۰/۰۵۵	۰/۰۰۰۴	۰/۰۴۷۴	۰/۰۲۹۷	۰/۰۰۰۷	۰/۰۷۴۶	۰/۰۶۰۶	۰/۰۷۴۶	۰/۰۷۴۶	۰/۰۷۴۶	NDC-2K
۱/۴۰۴	۰/۰۹۰۴	۰/۰۵۸۸	۰/۰۰۰۴	۰/۰۲۱۳	۰/۰۴۵۱	۰/۰۰۰۷	۰/۰۱۰۸	۰/۰۳۸۷	۰/۰۱۰۸	۰/۰۱۰۸	۰/۰۱۰۸	NDC-3K
۰/۵۸۱۲	۰/۱۹۴۷	۰/۰۳۱۴	۰/۰۰۰۵	۰/۰۸۹۹	۰/۰۲۴۹	۰/۰۰۰۷	۰/۰۹۳۲	۰/۰۲۱۹	۰/۰۹۳۲	۰/۰۹۳۲	۰/۰۹۳۲	NDC-4K
۰/۳۹۷۸۸	۰/۰۹۱۸	۰/۰۰۰۵	۰/۰۰۰۵	۰/۰۵۷۷	۰/۰۱۹۷	۰/۰۰۰۷	۰/۰۶۶۳	۰/۰۹۰۹	۰/۰۶۶۳	۰/۰۹۰۹	۰/۰۹۰۹	bNDC-5K
۱/۶۲/۶۰۵	۱/۱۲۴۹	۰/۰۷۷۵	۰/۰۰۰۷	۰/۰۷۵	۰/۰۰۰۷	۰/۰۰۰۷	۰/۰۵۶۲	۰/۰۵۶۲	۰/۰۵۶۲	۰/۰۵۶۲	۰/۰۵۶۲	bNDC-10K
a	۰/۵۷۰۷	۰/۰۱۱	۰/۰۰۱۱	۰/۰۵۸	۰/۰۰۱۱	۰/۰۰۱۱	a	a	a	a	a	bNDC-25K
a	۰/۸۳/۸۲۹	۰/۰۰۲	۰/۰۰۲	a	a	a	a	a	a	a	a	bNDC-50K

^a اجرای روش به دلیل زیاد بودن زمان آزمایش خاتمه پذیر است.
^b از تابع هسته مستطبی با اندازه ۱۰ درصد نمونه‌های آموزشی استفاده شده است.

شکل ۴-۵: تاثیر پارامتر k روی زمان آموزش روش KNN-LSTSVM

۳-۵ ارزیابی روش RKNN-TSVM

در این زیر بخش، ابتدا نحوه پیاده‌سازی و اجرای الگوریتم‌ها شرح داده شده است. سپس نحوه انتخاب پارامترهای بهینه توضیح داده شده است. در زیر بخش ، روش RKNN-TSVM روی مجموعه داده‌های مصنوعی و واقعی به طور جامع بررسی می‌شود.

۱-۳-۵ نحوه پیاده‌سازی و اجرای الگوریتم‌ها

نرم افزار LightTwinSVM برای اجرای روش TSVM در بخش ۳-۵ مورد استفاده قرار گرفته است. این نرم افزار شامل پیاده‌سازی ساده و سریع روش TSVM می‌باشد که به عنوان بخشی از این پژوهش به طور رایگان در اختیار عموم قرار گرفته است.^۶ روش RKNN-TSVM و سایر روش‌ها در زبان برنامه‌نویسی C++ نسخه ۳.۵ پیاده‌سازی شده و همچنین الگوریتم clipDCD و LDMDBA در زبان برنامه‌نویسی پیاده‌سازی شده است. تمامی آزمایش‌ها روی یک کامپیوتر شخصی با پردازنده Core i7 6700K، سیستم عامل Ubuntu 16.04 LTS و ۳۲ گیگابایت حافظه انجام شده است. کتابخانه و نرم افزارهای استفاده شده برای آزمایش روش RKNN-TSVM را نشان می‌دهد. جدول ۶-۵ کتابخانه و نرم افزارهای استفاده شده برای آزمایش روش RKNN-TSVM را نشان می‌دهد.

۲-۳-۵ نحوه انتخاب پارامترها

همانطور که در زیر بخش ۱-۲-۵ ذکر شد، عملکرد روش TSVM و گسترش‌هایش بسیار وابسته به پارامترهای بهینه است. روش جستجوی شبکه‌ای برای انتخاب پارامترهای بهینه در آزمایش‌های بخش ۳-۵ استفاده شده است. تابع RBF به عنوان تابع هسته $k(x_i, x_j) = \exp(-\|x_i - x_j\|^2 / \sigma^2)$ در نسخه غیر

⁶<https://github.com/mir-am/LightTwinSVM>

جدول ۵-۶: کتابخانه و نرم افزارهای استفاده شده برای ارزیابی روش RKNN-TSVM

کتابخانه/نرم افزار	توضیح
[۴۳] NumPy	اعمال جبر خطی در پایتون مانند ضرب و معکوس ماتریس
[۴۴] SciPy	محاسبه فاصله و توابع آماری
[۵۱] Scikit-learn	یادگیری ماشین و ارزیابی دسته‌بندها
GCC	کامپایلر برای زبان برنامه‌نویسی C++
Pybind11	اجرای کد C++ در پایتون

خطی بکار گرفته شده است. پارامتر تابع هسته σ از مجموعه $\{2^i \mid i = -10, -9, \dots, 2\}$ انتخاب شده است. مقادیر بهینه پارامترهای c_1, c_2 و c_3 از مجموعه $\{2^i \mid i = -8, -7, \dots, 2\}$ تعیین شده است. به منظور کاهش بار محاسباتی انتخاب پارامترها، $c_2 = c_1$ و $c_3 = c_4$ در روش TBSVM و $c_2 = c_3$ در روش RKNN-TSVM مساوی قرار داده شده است. همچنین مقدار بهینه پارامتر k از مجموعه $\{2, 3, \dots, 15\}$ مشخص شده است.

۳-۳-۵ نتایج ارزیابی و بحث

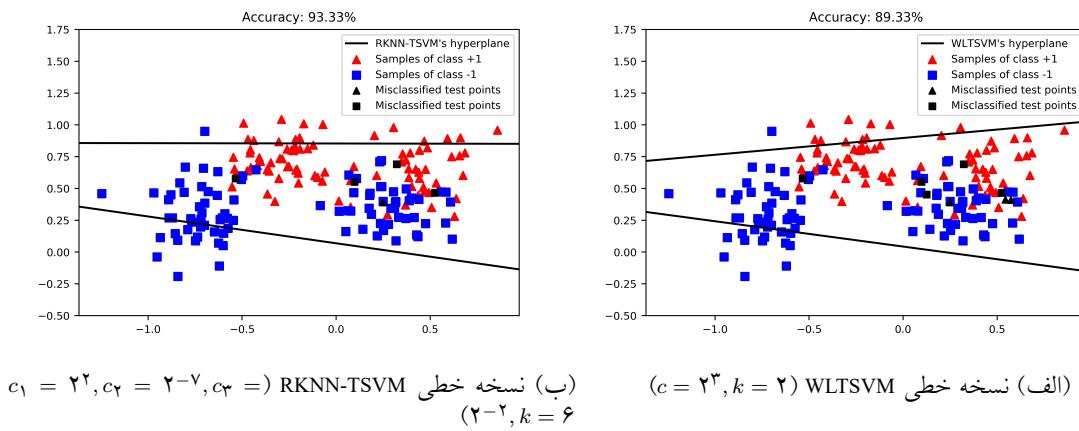
در این زیر بخش، روش پیشنهادی (RKNN-TSVM) روی مجموعه داده‌های مختلف مصنوعی و واقعی از نظر دقت و سرعت یادگیری بررسی می‌شود.

۱-۳-۵ مجموعه داده مصنوعی

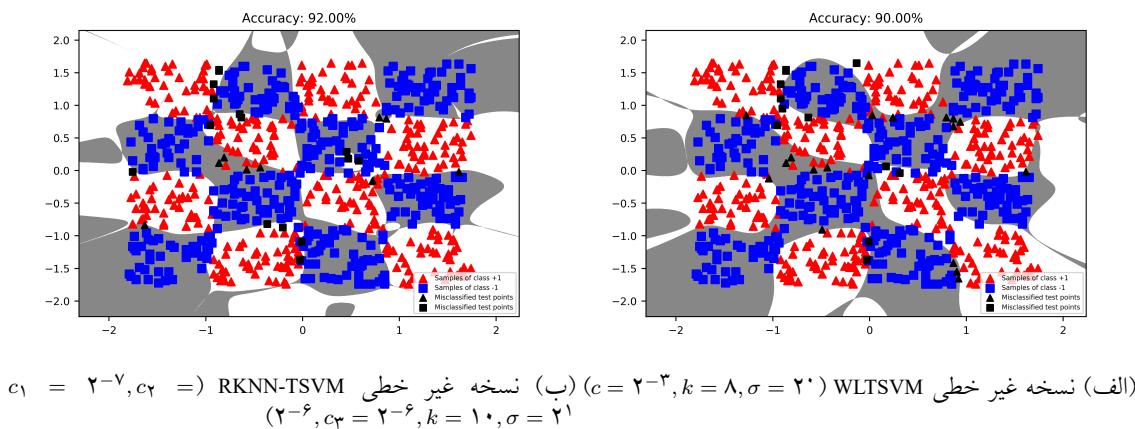
به منظور نشان دادن برتری روش پیشنهادی نسبت به روش WLTSVM به صورت هندسی، آزمایش روی دو مجموعه داده مصنوعی صورت گرفته است. ۷۰ درصد نمونه‌های این مجموعه داده‌ها برای آموزش دسته‌بند به صورت تصادفی انتخاب شده است.

در مثال اول، مجموعه داده مصنوعی Ripley با ۲۵۰ نمونه استفاده شده است. شکل ۵-۵ ناحیه تصمیم روش WLTSVM و RKNN-TSVM را روی مجموعه داده Ripley با تابع هسته خطی را نشان می‌دهد. همانطور که در شکل ۵-۵ مشخص است، روش پیشنهادی دقت بیشتری نسبت به روش WLTSVM دارد. زیرا روش پیشنهادی به نمونه‌ها براساس فاصله نزدیک‌ترین همسایه‌هایش وزن می‌دهد. همچنین ابرصفحه‌ها در روش RKNN-TSVM به نمونه‌های پرترکم نزدیک‌تر دارد.

در مثال دوم، مجموعه داده مصنوعی Checkerboard با ۱۰۰۰ نمونه استفاده شده است. شکل ۶-۵ ناحیه تصمیم روش WLTSVM و RKNN-TSVM را روی مجموعه داده Checkerboard با تابع هسته RBF



شکل ۵-۵: ناحیه تصمیم روش RKNN-TSVM و WLTSVM روی مجموعه داده Ripley با تابع هسته خطی



شکل ۵-۶: ناحیه تصمیم روش RKNN-TSVM و WLTSVM روی مجموعه داده Checkerboard با تابع هسته RBF

را نشان می‌دهد. دقیق نسخه غیر خطی روش پیشنهادی از نسخه غیر خطی روش WLTSVM روی این مجموعه داده بیشتر است. زیرا روش پیشنهادی ریسک ساختاری را کمینه می‌کند و همچنین وزن دهی به نمونه‌ها براساس فاصله‌شان از نزدیک همسایه‌ها می‌باشد.

۲-۳-۳-۵ مجموعه داده UCI

به منظور بررسی بیشتر عملکرد روش پیشنهادی (RKNN-TSVM)، مقایسه‌ای با روش‌های TSVM، WLTSVM و TBSVM روی مجموعه داده‌های UCI صورت گرفته است. لازم به ذکر است که تمامی مجموعه داده‌ها نرمال شده است. بطوریکه مقادیر ویژگی‌ها در بازه‌ی $[0, 1]$ قرار دارد. جدول ۷-۵ مشخصات این مجموعه داده‌ها را نشان می‌دهد.

ارزیابی عملکرد روش‌ها و تنظیم کردن پارامترها با اعتبارسنج ضربدری ۵ تایی انجام شده است. از

جدول ۵-۷: مشخصات مجموعه داده‌ها برای ارزیابی روش RKNN-TSVM

مجموعه داده تعداد نمونه‌ها نمونه‌های مثبت نمونه‌های منفی تعداد ویژگی‌ها				
۱۴	۳۸۳	۳۰۷	۶۹۰	Australian
۱۳	۱۵۰	۱۲۰	۲۷۰	Heart-Statlog
۶	۲۰۰	۱۴۵	۳۴۵	Bupa-Liver
۳۳	۱۵۱	۴۷	۱۹۸	WPBC
۳۰	۳۵۷	۲۱۲	۵۶۹	WDBC
۱۹	۱۲۳	۳۲	۱۵۵	Hepatitis
۳۴	۱۲۶	۲۲۵	۳۵۱	Ionosphere
۳	۸۱	۲۲۵	۳۰۶	Haberman
۸	۵۰۰	۲۶۸	۷۶۸	Pima-Indian
۹	۱۲	۸۸	۱۰۰	Fertility
۱۶	۱۶۸	۲۶۷	۴۳۵	Votes

این رو مجموعه داده به ۵ بخش به صورت تصادفی تقسیم می‌شود و یکی از این زیربخش‌ها به عنوان نمونه‌های تست استفاده می‌گردد. فرآیند تقسیم‌بندی داده‌ها ۵ بار تکرار می‌شود و میانگین این ۵ بار به عنوان دقت دسته‌بند بشمار می‌رود.

دقت دسته‌بندی و زمان اجرای روش‌های WLTSVM، TBSVM، TSVM و RKNN-TSVM در جدول ذکر شده است. در اینجا دقت نشان دهنده میانگین نتایج روی داده‌های تست (به درصد) می‌باشد. زمان اجرا نیز به ثانیه است.

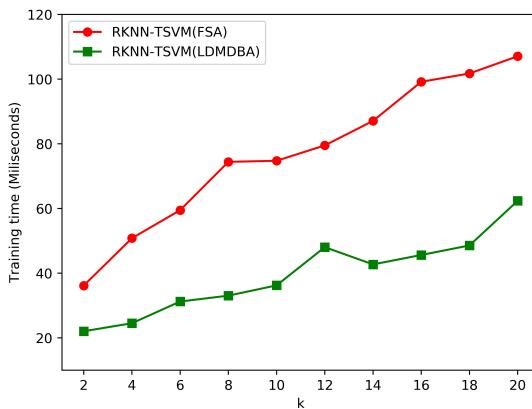
از جنبه دقت دسته‌بندی، روش پیشنهادی (RKNN-TSVM) بهتر از سایر روش‌ها مانند TSVM و WLTSVM روی بیشتر مجموعه داده‌ها عمل کرده است. ویژگی‌های روش پیشنهادی باعث بهبود دقت شده است که در زیر توضیح داده شده است.

۱. روش پیشنهادی به نمونه‌ها بر اساس فاصله شان از نزدیک‌ترین همسایه‌ها وزن می‌دهد. در حالی که در روش WLTSVM صرفا براساس شماره تعداد همسایه‌های نزدیک به نمونه وزن نسبت داده می‌شود. بطوریکه ماتریس وزن‌ها در روش WLTSVM شامل مقادیر دودویی است.

۲. مشابه روش TBSVM، روش پیشنهادی ریسک ساختاری را در مسئله بهینه سازی خود در نظر می‌گیرد. در حالی که روش‌های TSVM و WLTSVM ریسک تجربی را کمینه می‌کنند. به عبارت دیگر، مقادیر پارامترهای c_2 و c_3 دقت دسته‌بندی روش بهبود داده است. با این حال مقادیر این پارامتر در روش‌های TSVM و WLTSVM یک عدد ثابت بسیار کوچک است.

جدول ۵-۸: مقایسه روش‌های RBF با تابع هسته، مجموعه دادهای UCI و RKNN-TSVM و WLTSVM و TBSVM و TSVM

RKNN-TSVM(LLMDBA)		RKNN-TSVM(FSA)		WLTSVM		TBSVM		TSVM	
زمان	دقت (%)	زمان	دقت (%)	زمان	دقت (%)	زمان	دقت (%)	زمان	دقت (%)
اجرا	(c_1, c_r, σ, k)	اجرا	(c_1, c_r, σ, k)	اجرا	(c, σ, k)	اجرا	(c_1, c_r, σ)	اجرا	(c_1, c_r, σ)
۰/۰۲۶۴	۸۷/۹۷ ± ۳/۸۵	۸۷/۵۴ ± ۳/۹۵	۸۶/۵۲ ± ۳/۵۳	۸۷/۳۹ ± ۳/۳۹	۸۷/۶۶ ± ۰/۰۹	۸۷/۱۰ ± ۰/۰۹	۸۷/۳۰ ± ۰/۰۹	۸۷/۶۶ ± ۰/۰۹	Australian (۶۹۰ × ۱۴)
(۲-۳, ۲-۳, ۲-۲, ۰)	(۲-۳, ۲-۳, ۲-۴, ۲)	(۲-۳, ۲-۳, ۲-۴, ۱)	(۲-۳, ۲-۴, ۱, ۱)	(۲-۳, ۲-۴, ۰, ۱)	(۲-۳, ۲-۴, ۰, ۰)	(۲-۴, ۲-۴, ۰, ۰)	(۲-۴, ۲-۴, ۰, ۰)	(۲-۴, ۲-۴, ۰, ۰)	Australian (۶۹۰ × ۱۴)
۰/۰۲۸	۸۰/۵۹ ± ۲/۱۹	۸۰/۰۲۳	۸۳/۷۰ ± ۱/۳۹	۸۰/۰۱۳	۸۰/۰۱۳	۸۰/۰۱۰	۸۴/۸۱ ± ۰/۰۱	۸۰/۰۱۰	Heart-Statlog (۲۷۰ × ۱۳)
(۲-۳, ۲-۳, ۲-۱, ۰)	(۲-۳, ۲-۳, ۲-۱, ۱)	(۲-۳, ۲-۳, ۲-۱, ۲)	(۲-۳, ۲-۳, ۲-۱, ۳)	(۲-۳, ۲-۳, ۲-۱, ۴)	(۲-۳, ۲-۳, ۲-۱, ۵)	(۲-۳, ۲-۳, ۲-۱, ۶)	(۲-۳, ۲-۳, ۲-۱, ۷)	(۲-۳, ۲-۳, ۲-۱, ۸)	Bupa-Liver (۳۴۵ × ۹)
۰/۰۶۶	۷۳/۹۱ ± ۴/۵۸	۷۳/۰۴۹	۷۳/۹۱ ± ۴/۳۰	۰/۰۲۹	۷۳/۹۲ ± ۱/۱۳	۰/۰۱۶	۷۴/۷۸ ± ۰/۳۵	۰/۰۱۶	WDBC (۱۹۸ × ۲۲)
(۲-۳, ۲-۳, ۲-۵, ۰)	(۲-۳, ۲-۳, ۲-۵, ۱)	(۲-۳, ۲-۳, ۲-۵, ۲)	(۲-۳, ۲-۳, ۲-۵, ۳)	(۲-۳, ۲-۳, ۲-۵, ۴)	(۲-۳, ۲-۳, ۲-۵, ۵)	(۲-۳, ۲-۳, ۲-۵, ۶)	(۲-۳, ۲-۳, ۲-۵, ۷)	(۲-۳, ۲-۳, ۲-۵, ۸)	WDBC (۱۹۸ × ۲۲)
۰/۰۲۸	۸۰/۳۲ ± ۲/۹۸	۸۰/۰۱۶	۷۸/۸۲ ± ۸/۰۵	۰/۰۱۲	۷۸/۸۱ ± ۷/۴۷	۰/۰۱۷	۷۹/۲۷ ± ۰/۴۸	۰/۰۱۷	WDBC (۱۹۸ × ۲۲)
(۲-۳, ۲-۳, ۲-۴, ۱, ۰)	(۲-۳, ۲-۳, ۲-۴, ۱, ۱)	(۲-۳, ۲-۳, ۲-۴, ۱, ۲)	(۲-۳, ۲-۳, ۲-۴, ۱, ۳)	(۲-۳, ۲-۳, ۲-۴, ۱, ۴)	(۲-۳, ۲-۳, ۲-۴, ۱, ۵)	(۲-۳, ۲-۳, ۲-۴, ۱, ۶)	(۲-۳, ۲-۳, ۲-۴, ۱, ۷)	(۲-۳, ۲-۳, ۲-۴, ۱, ۸)	WDBC (۱۹۸ × ۲۲)
۰/۰۱۵	۹۸/۵۹ ± ۰/۷*	۹۸/۰۹	۹۷/۵۴ ± ۱/۰۲	۰/۰۰۵	۹۸/۲۴ ± ۰/۷۸	۰/۰۰۷۲	۹۸/۲۴ ± ۰/۳۶	۰/۰۰۷۲	WDBC (۱۹۸ × ۲۲)
(۲-۳, ۲-۳, ۲-۴, ۰)	(۲-۳, ۲-۳, ۲-۴, ۱)	(۲-۳, ۲-۳, ۲-۴, ۲)	(۲-۳, ۲-۳, ۲-۴, ۳)	(۲-۳, ۲-۳, ۲-۴, ۴)	(۲-۳, ۲-۳, ۲-۴, ۵)	(۲-۳, ۲-۳, ۲-۴, ۶)	(۲-۳, ۲-۳, ۲-۴, ۷)	(۲-۳, ۲-۳, ۲-۴, ۸)	WDBC (۱۹۸ × ۲۲)
۰/۰۱۵	۸۸/۳۹ ± ۶/۹۵	۸۷/۰۱۷	۸۷/۷۴ ± ۷/۱۸	۰/۰۱۲	۸۷/۱۶ ± ۵/۹۸	۰/۰۰۰۴	۸۵/۸۱ ± ۷/۷*	۰/۰۰۰۴	Hepatitis (۱۵۵ × ۱۹)
(۲-۳, ۲-۳, ۲-۴, ۳)	(۲-۳, ۲-۳, ۲-۴, ۴)	(۲-۳, ۲-۳, ۲-۴, ۵)	(۲-۳, ۲-۳, ۲-۴, ۶)	(۲-۳, ۲-۳, ۲-۴, ۷)	(۲-۳, ۲-۳, ۲-۴, ۸)	(۲-۳, ۲-۳, ۲-۴, ۹)	(۲-۳, ۲-۳, ۲-۴, ۱۰)	(۲-۳, ۲-۳, ۲-۴, ۱۱)	Hepatitis (۱۵۵ × ۱۹)
۰/۰۶۶	۹۳/۱۷ ± ۳/۸۷	۹۳/۰۴۷	۹۳/۷۳ ± ۳/۳۵	۰/۰۰۷	۹۲/۹۰ ± ۳/۹۷	۰/۰۰۰۱	۹۰/۸۹ ± ۴/۹۱	۰/۰۰۰۱	Ionosphere (۳۵۱ × ۳۴)
(۲-۳, ۲-۳, ۲-۴, ۱, ۱)	(۲-۳, ۲-۳, ۲-۴, ۱, ۲)	(۲-۳, ۲-۳, ۲-۴, ۱, ۳)	(۲-۳, ۲-۳, ۲-۴, ۱, ۴)	(۲-۳, ۲-۳, ۲-۴, ۱, ۵)	(۲-۳, ۲-۳, ۲-۴, ۱, ۶)	(۲-۳, ۲-۳, ۲-۴, ۱, ۷)	(۲-۳, ۲-۳, ۲-۴, ۱, ۸)	(۲-۳, ۲-۳, ۲-۴, ۱, ۹)	Haberman (۳۰۶ × ۳)
۰/۰۴۹	۷۹/۷۹ ± ۳/۹۷	۷۹/۰۳۱	۷۹/۷۷ ± ۰/۵۱*	۰/۰۲۷	۷۹/۱۱ ± ۰/۳۶	۰/۰۱۲	۷۵/۸۲ ± ۰/۱۷	۰/۰۱۵	Pima-Indian ($\nabla ۶۸ \times \lambda$)
(۲-۳, ۲-۳, ۲-۴, ۰)	(۲-۳, ۲-۳, ۲-۴, ۱)	(۲-۳, ۲-۳, ۲-۴, ۲)	(۲-۳, ۲-۳, ۲-۴, ۳)	(۲-۳, ۲-۳, ۲-۴, ۴)	(۲-۳, ۲-۳, ۲-۴, ۵)	(۲-۳, ۲-۳, ۲-۴, ۶)	(۲-۳, ۲-۳, ۲-۴, ۷)	(۲-۳, ۲-۳, ۲-۴, ۸)	Pima-Indian ($\nabla ۶۸ \times \lambda$)
۰/۰۹۲	۹۷/۰۱ ± ۱/۵۶	۹۷/۰۴۰	۹۷/۰۱ ± ۱/۳۸	۰/۰۰۴۲	۹۶/۰۵ ± ۰/۹۱	۰/۰۰۲۱	۹۷/۰۵ ± ۰/۹۱	۰/۰۰۲۱	Fertility (۱۰۰ × ۹)
(۲-۳, ۲-۳, ۲-۴, ۱, ۱)	(۲-۳, ۲-۳, ۲-۴, ۱, ۲)	(۲-۳, ۲-۳, ۲-۴, ۱, ۳)	(۲-۳, ۲-۳, ۲-۴, ۱, ۴)	(۲-۳, ۲-۳, ۲-۴, ۱, ۵)	(۲-۳, ۲-۳, ۲-۴, ۱, ۶)	(۲-۳, ۲-۳, ۲-۴, ۱, ۷)	(۲-۳, ۲-۳, ۲-۴, ۱, ۸)	(۲-۳, ۲-۳, ۲-۴, ۱, ۹)	Votes (۴۳۵ × ۱۶)
۰/۰۵۱	۸۶/۳۹	۸۶/۵۱	۸۵/۱۰	۸۵/۷۵	۸۵/۱۰	۸۵/۷۵	۸۵/۱۰	۸۵/۷۵	میانگین دقت



شکل ۷-۵: تاثیر پارامتر k روی زمان آموزش روش RKNN-TSVM روی مجموعه داده Pima-Indian

با توجه به جدول ۸-۵، نه تنها روش پیشنهادی با الگوریتم LDMDBA از سایر روش‌های TSVM و WLTSVM بهتر عمل کرده است، بلکه نسبت به روش پیشنهادی با الگوریتم FSA نیز دقیق‌تر است. هدف از بکارگیری الگوریتم LDMDBA بهبود سرعت آموزش روش پیشنهادی بوده است. با این حال این الگوریتم دقیق‌تر است.

از جنبه زمان اجرا و سرعت آموزش، روش TSVM از روش پیشنهادی و WLTSVM سریع‌تر است. زیرا روش فقط دو مسئله دوگان را حل می‌کند. در حالی‌که در روش پیشنهادی و WLTSVM علاوه بر حل کردن دو مسئله دوگان، نزدیک‌ترین همسایه‌های تمام نمونه‌ها باید محاسبه گردد. به منظور بهبود مرتبه زمانی روش پیشنهادی (RKNN-TSVM)، الگوریتم LDMDBA برای ساخت گراف نزدیک‌ترین همسایه استفاده شده است. در بخش ۳-۵، سرعت آموزش روش پیشنهادی بر روی مجموعه داده‌های بزرگ بررسی شده است.

شکل ۷-۵ تاثیر مقدار پارامتر k روی زمان آموزش روش پیشنهادی با الگوریتم FSA و LDMDBA روی مجموعه داده Pima-Indian را نشان می‌دهد. همانطور که در شکل ۷-۵ مشخص است، زمان آموزش روش پیشنهادی با افزایش مقدار k بیشتر می‌شود. با این حال روش پیشنهادی با الگوریتم LDMDBA به طور قابل توجه‌ای از الگوریتم FSA برای هر یک از مقادیر k سریع‌تر است.

۳-۳-۳-۵ بررسی آماری

به منظور تحلیل آماری عملکرد پنج دسته‌بند روی مجموعه داده‌های UCI، آزمون فریدمن بکار گرفته شده است (نحوه محاسبه این آزمون آماری در بخش ۲-۱-۳-۵ بیان شده است). بدین منظور میانگین

جدول ۹-۵: میانگین رتبه براساس داده (ارزیابی RKNN-TSVM)

RKNN-TSVM(LMDDBA)	RKNN-TSVM(FSA)	WLTSVM	TBSVM	TSVM	مجموعه داده
۱	۲	۵	۳	۴	Australian
۳	۱/۵	۵	۱/۵	۴	Heart-Statlog
۳	۳	۳	۵	۱	Bupa-Liver
۱	۲	۴	۵	۳	WPBC
۱/۵	۱/۵	۵	۳/۵	۳/۵	WDBC
۱	۲	۵	۳	۴	Hepatitis
۲	۱	۳	۴	۵	Ionosphere
۱	۲	۳	۴	۵	Haberman
۱	۲	۵	۴	۳	Pima-Indian
۱	۲	۴/۵	۳	۴/۵	Fertility
۲	۲	۴/۵	۲	۴/۵	Votes
۱/۵۹	۱/۹۱	۴/۲۷	۳/۴۵	۳/۷۷	میانگین رتبه

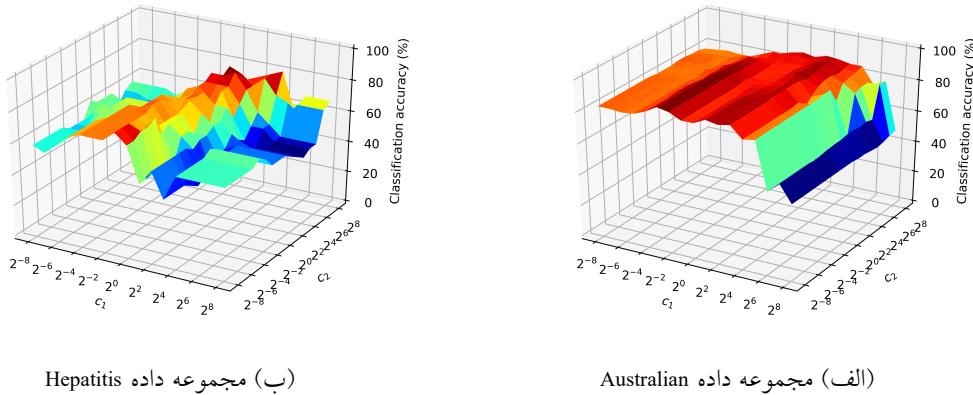
رتبه پنج روش براساس دقت محاسبه شده و در جدول نمایش داده شده است. ابتدا فرض گرفته می‌شود که بر اساس فرضیه صفر تمام دسته‌بندها یکسان هستند.

بعد از انجام آزمون آماری فریدمن، مقادیر $F_F = ۱۲/۷۲۳$ و $\chi_F^2 = ۲۴/۶۳۶$ بدست آمده است. با در نظر گرفتن ۵ دسته‌بند و ۱۱ مجموعه داده، F_F از توزیع F با $(4, 40)$ درجه آزادی پیروی می‌کند. مقادیر ویژه $(4, 40)$ برای سطوح معناداری $0/25$ ، $0/1$ و $0/05$ به ترتیب برابر با $1/40$ ، $2/61$ و $2/09$ است. مقدار F_F به طور قابل توجه‌ای بیشتر از مقدار ویژه است. بنابر با توجه به نتایج آزمون آماری، فرضیه صفر رد می‌شود. در نتیجه تفاوت معنادار و قابل توجه‌ای بین ۵ دسته‌بند وجود دارد. همچنین جدول ۹-۵ نشان می‌دهد که روش پیشنهادی (RKNN-TSVM) بهتر از سایر روش‌ها عمل کرده است. زیرا میانگین رتبه روش پیشنهادی در میان سایر روش‌ها کمترین است.

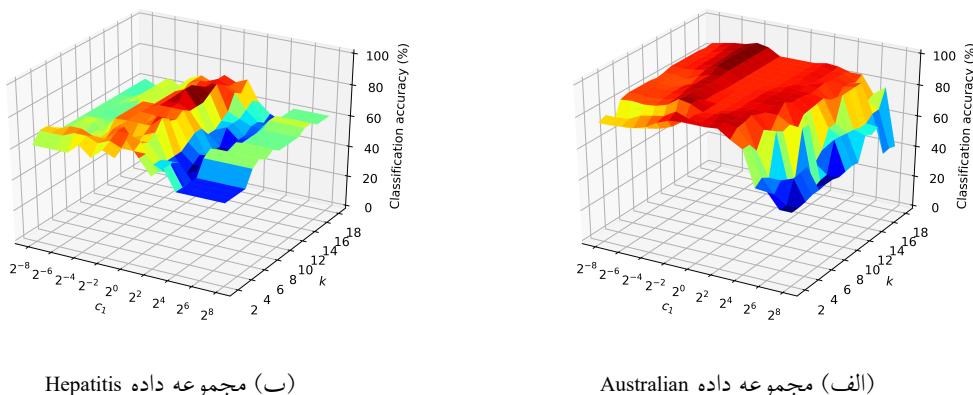
۴-۳-۵ بررسی حساسیت روش RKNN-TSVM به پارامترها

به منظور دست‌یابی به دقت بهتر، انتخاب پارامترهای بهینه برای روش پیشنهادی اهمیت زیادی دارد. آزمایش روی دو مجموعه داده Hepatitis و Australian صورت گرفته است تا حساسیت روش پیشنهادی به پارامترهای c_1 و c_2 و k بررسی شود.

برای هر مجموعه داده، در این آزمایش، تعداد حالت‌های پارامترهای c_1 و c_2 و k برابر با ۱۷ است. بطوریکه تعداد ترکیبات (c_1, c_2) و (c_1, k) مساوی با ۲۸۹ می‌باشد. شکل ۸-۵ عملکرد نسخه خطی روش روی پارامترهای مختلف c_1 و c_2 نشان می‌دهد. همانطور در این شکل نشان داده شده است،



شکل ۸-۵: عملکرد نسخه خطی روش RKNN-TSVM روی پارامترهای مختلف c_1 و c_2



شکل ۹-۵: عملکرد نسخه خطی روش RKNN-TSVM روی پارامترهای مختلف c_1 و k

مقادیر پارامتر c_2 دقت دسته‌بندی روش RKNN-TSVM را بهبود می‌دهد. بنابراین می‌توان نتیجه گرفت که ریسک ساختاری دقت روش پیشنهادی (RKNN-TSVM) را بهتر می‌کند.

شکل ۹-۵ عملکرد نسخه خطی روش RKNN-TSVM روی مقادیر پارامترهای c_1 و k را نشان می‌دهد. همانطور که در این شکل مشخص شده است، عملکرد روش RKNN-TSVM به پارامتر k نیز وابسته است. به عنوان مثال، افزایش مقدار پارامتر k منجر به بهبود دقت روش پیشنهادی روش مجموعه داده Hepatitis شده است. به طور خلاصه، آزمایش‌های این بخش نشان می‌دهد که دقت روش پیشنهادی (RKNN-TSVM) وابسته به انتخاب بهینه پارامترهایش است.

۵-۳-۳-۵ آزمایش با مجموعه داده NDC

به منظور بررسی سرعت آموزش روش RKNN-TSVM بر روی مجموعه داده‌های بزرگ، آزمایش بر روی مجموعه داده NDC [۵۰] صورت گرفته است. مشخصات مجموعه داده‌های NDC در جدول ۴-۵ نشان داده شده است. جهت آزمایش با مجموعه داده NDC، پارامتر C برای تمام روش برابر با یک است. در نسخه غیر خطی ازتابع RBF با پارامتر $\sigma = 2^{-15}$ استفاده شده است. همچنین پارامتر k برای روش‌های WLTSVM و RKNN-TSVM برابر با ۵ است.

جدول ۱۰-۵ مقایسه زمان آموزش روش‌های WLTSVM, TSVM و RKNN-TSVM با تابع هسته خطی را نشان می‌دهد. مشابه روش TBSVM، روش TSVM دو مسئله دوگان را برای بدست آوردن مدل خروجی حل می‌کند. بنابراین زمان آموزش روش TBSVM در این آزمایش آورده نشده است. ستون آخر نسبت تسریع الگوریتم LDMDBA را نشان می‌دهد که به این صورت تعریف می‌شود:

$$\frac{\text{زمان آموزش روش (FSA)}}{\text{زمان آموزش روش (LDMDBA)}} = \text{نسبت تسریع}$$

نتایج در جدول ۱۰-۵ نشان می‌دهد که الگوریتم LDMDBA سرعت آموزش روش پیشنهادی را به طور قابل توجه‌ای بهبود داده است. بطوریکه با افزایش تعداد نمونه‌های آموزشی، نسبت تسریع الگوریتم LDMDBA بیشتر می‌شود. برای مثال، روش پیشنهادی با الگوریتم LDMDBA از روش پیشنهادی با الگوریتم FSA روی مجموعه داده NDC-25K حدود ۳/۲۵ سریع‌تر است. همچنین سرعت آموزش نسخه خطی روش RKNN-TSVM با الگوریتم LDMDBA بسیار نزدیک به نسخه خطی TSVM می‌باشد. بطوریکه نسخه خطی روش پیشنهادی از نسخه خطی TSVM روی مجموعه داده NDC-50K کمی سریع‌تر است. زیرا روش پیشنهادی دو مسئله دوگان با اندازه کوچک‌تر حل می‌کند. به عبارت دیگر فقط نمونه‌های حاشیه‌ای در مسئله دوگان نقش دارند.

جدول ۱۱-۵ مقایسه زمان آموزش روش‌های WLTSVM, TSVM و RKNN-TSVM با تابع هسته LDMDBA را نشان می‌دهد. نتایج با تابع هسته غیر خطی نشان می‌دهد که روش پیشنهادی با الگوریتم LDMDBA از روش WLTSVM و روش RKNN-TSVM با الگوریتم FSA بسیار سریع‌تر است. بطوریکه بیشترین نسبت تسریع با الگوریتم LDMDBA مساوی با ۱۴ برابر می‌باشد. با وجود اینکه از تابع هسته تقلیل یافته برای مجموعه داده‌های بزرگ از ۵ هزار نمونه استفاده شده است، روش TSVM همچنان حدود ۲ برابر سریع‌تر از روش پیشنهادی با الگوریتم LDMDBA است. زیرا روش پیشنهادی با الگوریتم LDMDBA

۳-۵. ارزیابی روش RKNN-TSVM

جدول ۱۰-۵: مقایسه زمان آموزش روش RKNN-TSVM با سایر روش روی مجموعه داده NDC با تابع هسته خطی

نسبت تسریع	RKNN-TSVM(LDMDBA)	RKNN-TSVM(FSA)	WLTSVM	TSVM	مجموعه داده
	زمان اجرا	زمان اجرا	زمان اجرا	زمان اجرا	
۱/۰۲	۰/۰۵۲	۰/۰۷۹	۰/۰۹۲	۰/۰۶۴	NDC-1K
۱/۰۴	۰/۱۹	۰/۲۹۲	۰/۳۶	۰/۱۲	NDC-2K
۲/۲۴	۰/۲۹۵	۰/۶۶۲	۰/۸۴	۰/۲۶	NDC-3K
۲/۱۲	۰/۵۶۲	۱/۱۹۲	۱/۴۷۶	۰/۴۲۲	NDC-4K
۲/۲۸	۰/۸۲۸	۱/۸۸۴	۲/۳۹۷	۰/۶۹۳	NDC-5K
۲/۸	۲/۷۲۷	۷/۶۲۸	۹/۸۷۲	۲/۵۵۶	NDC-10K
۳/۲۵	۱۶/۲۵	۵۲/۸۶۷	۶۸/۸۹۳	۱۷/۶۰۶	NDC-25K
-	۶۴/۴۳۳	a	a	۷۰/۱	NDC-50K

^a آزمایش به دلیل کمبود حافظه خاتمه یافته است.

جدول ۱۱-۵: مقایسه زمان آموزش روش RKNN-TSVM با سایر روش روی مجموعه داده NDC با تابع هسته RBF

نسبت تسریع	RKNN-TSVM(LDMDBA)	RKNN-TSVM(FSA)	WLTSVM	TSVM	مجموعه داده
	زمان اجرا	زمان اجرا	زمان اجرا	زمان اجرا	
۱/۴۵	۰/۵۵۵	۰/۸۰۷	۰/۸۰۳	۰/۲۰۳	NDC-1K
۲/۳۵	۲/۴۴۲	۵/۷۲۹	۵/۷۳۱	۰/۹۸۳	NDC-2K
۲/۸۸	۶/۴۶۵	۱۸/۰۹۹	۱۸/۲۲۵	۲/۷۴	NDC-3K
۳/۳۵	۱۲/۴۸۵	۴۱/۷۸۴	۴۲/۲۲۴	۰/۸۹۶	NDC-4K
۳/۹	۲۱/۱۴	۸۲/۵۰۷	۸۴/۱۸۸	۱۰/۳۲۸	NDC-5K
۷/۵۲	۸/۶۰۶	۶۴/۷۲۱	۶۷/۶۲۶	۴/۶۰۵	^b NDC-10K
۱۴/۲۷	۶۷/۴۸۵	۹۶۳/۳۴۱	۹۸۳/۶۷۸	۳۱/۴۵۹	^b NDC-25K
-	۳۵۷/۹۴۲	a	a	۱۸۶/۷۶۱	^b NDC-50K

^a اجرای روش به دلیل زیاد بودن زمان آزمایش خاتمه یافته است.

^b از تابع هسته مستطیلی با اندازه ۱۰ درصد نمونه‌های آموزشی استفاده شده است.

و تابع هسته تقلیل یافته ($n \times \bar{n}$) علاوه بر پیدا کردن نزدیکترین همسایه‌های تمام نمونه‌ها، دو مسئله دوگان نیز حل می‌کند.

نتایج روی مجموعه داده NDC با تابع هسته RBF این فرضیه را تایید می‌کند که الگوریتم LDMDBA برای نسخه غیر خطی روش RKNN-TSVM مناسب است. زمان اجرای این الگوریتم روی داده‌های با ابعاد بالا بسیار بهتر از الگوریتم FSA می‌باشد. به طور خلاصه، روش RKNN-TSVM با الگوریتم WLTSVM نسبت به روش LDMDBA روی مجموعه داده‌های بزرگ از نظر زمان اجرا بسیار بهتر عمل می‌کند.

۴-۵ جمع‌بندی

در این فصل، دو دسته‌بند پیشنهادی یعنی KNN-TSVM و RKNN-TSVM مورد بررسی و ارزیابی قرار گرفت. روش KNN-LSTSVM مزیت‌های اصلی دو روش WLTSVM و LSTSVM را دارد که عبارتند از:

۱. مشابه روش WLTSVM، اطلاعات شباهت نمونه‌ها را با ساخت گراف نزدیک‌ترین همسایه در مسئله بهینه‌سازی لحاظ می‌کند. بطوریکه به هر نمونه بر اساس شمارش تعداد همسایه‌های نزدیک‌ش وزن داده می‌شود. همچنین نمونه‌های حاشیه‌ای هر کلاس نیز با استفاده از گراف ساخته شده مشخص می‌گردد. نتایج بر روی مجموعه داده‌های مصنوعی (بخش ۲-۲-۵) واقعی (بخش ۳-۲-۵) نشان می‌دهد دقت مدل خروجی نسبت به روش LSTSVM بهبود یافته است.

۲. مشابه روش LSTSVM، قید مسئله بهینه‌سازی در تابع هدف جایگذاری می‌شود. بطوریکه مدل خروجی با حل کردن دو دستگاه معادلات خطی بدست می‌آید. در حالی‌که در روش WLTSVM دو مسئله دوگان حل می‌شود. نتایج ارزیابی روی مجموعه داده‌های بزرگ (بخش ۴-۲-۵) نشان می‌دهد که سرعت آموزش نسبت به روش WLTSVM بسیار افزایش یافته است.

روش پیشنهادی دوم یعنی WLTSVM روش RKNN-TSVM را از نظر دقت و سرعت آموزش بهبود داده است:

۱. برخلاف روش RKNN-TSVM به یک نمونه براساس فاصله نزدیک‌ترین همسایه‌هایش از نمونه مورد نظر وزن می‌دهد. همچنین روش پیشنهادی ریسک ساختاری را در مسئله بهینه‌سازی کمینه می‌کند. نتایج بر روی مجموعه داده‌های مصنوعی (بخش ۱-۳-۵) واقعی (بخش ۲-۳-۵) نشان می‌دهد که دقت دسته‌بندی و تعمیم‌پذیری روش پیشنهادی نسبت به روش WLTSVM بهبود یافته است.

۲. محاسبه گراف نزدیک‌ترین همسایه جهت وزن‌دهی به نمونه‌ها، یک چالش در روش پیشنهادی است. به منظور حل کردن این چالش، الگوریتم LDMDBA به منظور سرعت بخشیدن به فرآیند پیدا کردن نزدیک‌ترین همسایه‌های نمونه‌ها استفاده شده است. نتایج بر روی مجموعه داده‌های بزرگ نشان می‌دهد که الگوریتم LDMDBA سرعت آموزش روش پیشنهادی را برای هر دو نسخه خطی و غیر خطی به طور قابل توجه‌ای افزایش داده است. بطوریکه سرعت یادگیری تا ۱۴ برابر نسبت به روش WLTSVM سریع‌تر شده است.

فصل ۶

نتیجه‌گیری و پژوهش‌های آینده

۱-۶ مقدمه

در این فصل، ابتدا ویژگی‌های دسته‌بندهای پیشنهادی مرور می‌شود. سپس یافته‌های مهم این پژوهش به طور خلاصه بیان می‌شود. در آخر، پیشنهادهایی برای پژوهش‌های آینده بیان می‌گردد.

۲-۶ مروری بر دسته‌بندهای پیشنهادی

هدف اصلی این پژوهش، ارائه یک دسته‌بند مبتنی بر ماشین بردار پشتیبان دو قلو است که بتواند دقت و تعمیم‌پذیری مطلوبی در برابر داده‌های نویزی و پرت داشته باشد. بدین منظور دو دسته‌بند-KNN و LSTSVM و RKNN-TSVM به ترتیب در فصول ۳ و ۴ ارائه شد.

دسته‌بند KNN-LSTSVM [۱۵] با ایده گرفتن از روش WLTSVM گراف نزدیک‌ترین همسایه را برای تمام نمونه‌های آموزشی ایجاد می‌کند. سپس با استفاده از گراف درون کلاسی W_w به نمونه‌ها بر اساس شمارش تعداد همسایه‌های نزدیک‌شان وزن می‌دهد و همچنین با بکارگیری گراف برون کلاسی W_b نمونه‌های حاشیه‌ای هر کلاس را مشخص می‌کند. بطوریکه دقت و تعمیم‌پذیری روش KNN-LSTSVM نسبت به روش TSVM با وجود داده‌های نویزی بهتر است. به منظور افزایش سرعت یادگیری و آموزش، روش KNN-LSTSVM همانند روش LSTSVM قید مسئله بهینه‌سازی را در تابع هدف جایگذاری می‌کند. بطوریکه مدل خروجی با حل کردن دو دستگاه معادلات خطی بدست می‌آید. دسته‌بند پیشنهادی در بخش ۲-۵ مورد ارزیابی و بررسی قرار گرفت. نتایج نشان می‌دهد که روش پیشنهادی نسبت به روش TSVM از نظر دقت دسته‌بندی و سرعت یادگیری به طور قابل توجه‌ای بهتر می‌باشد.

در ادامه، دسته‌بند RKNN-TSVM با هدف برطرف کردن مشکلات روش WLTSVM ارائه شده است.

روش پیشنهادی سه مزیت نسبت به روش WLTSVM دارد که عبارتند از:

۱. روش پیشنهادی به نمونه‌ها بر اساس فاصله از نزدیکترین همسایه‌شان وزن می‌دهد. برای مثال، اگر فاصله نزدیکترین همسایه‌های یک نمونه از آن بسیار کم باشد، وزن بیشتری به این نمونه نسبت داده می‌شود. این شیوه وزن‌دهی نمونه‌های پرتراکم را بهتر از روش WLTSVM شناسایی می‌کند.

۲. روش پیشنهادی RKNN-TSVM برخلاف روش WLTSVM ریسک ساختاری را کمینه می‌کند. زیرا جمله رگولارسیون به مسائل بهینه‌سازی روش پیشنهادی اضافه شده است. بطوریکه مانند روش SVM حاشیه بیشینه می‌گردد. کمینه کردن ریسک ساختاری در روش پیشنهادی باعث بهبود تعمیم‌پذیری آن شده است.

۳. سرعت یادگیری روش WLTSVM به دلیل محاسبه گراف نزدیکترین همسایه برای مجموعه داده‌های بزرگ بسیار کاهش می‌یابد. با این حال روش پیشنهادی از الگوریتم LDMDBA برای ساخت گراف نزدیکترین همسایه استفاده می‌کند. مرتبه زمانی این الگوریتم برابر با $O(\log nm \log m)$ است. بطوریکه سرعت یادگیری روش پیشنهادی نسبت به روش WLTSVM برای مجموعه داده‌های بزرگ به طور قابل توجهی سریع‌تر است.

نتایج بدست آمده در بخش ۳-۵، مزیت‌های دسته‌بند RKNN-TSVM نسبت به روش WLTSVM را تایید می‌کند.

۴-۳ مروری بر یافته‌های این پژوهش

در فصل ۵، دو دسته‌بند پیشنهادی یعنی KNN-TSVM و RKNN-TSVM به طور جامع بررسی و ارزیابی شده است. در این بخش یافته‌های مهم و اصلی این پژوهش براساس نتایج ارزیابی به طور خلاصه در زیر بیان شده است.

- دسته‌بندهای KNN-TSVM و RKNN-TSVM با در نظر گرفتن گراف درون کلاسی در مسئله بهینه‌سازی‌شان به نمونه‌های آموزشی وزن می‌دهند. بطوریکه مدل خروجی نسبت به نمونه‌های پرت و نویزی حساسیت کمتری خواهد داشت. در نتیجه دقت و تعمیم‌پذیری مدل خروجی افزایش می‌یابد.

• روش‌های LSTSVM و KNN-LSTSVM با استفاده از تکنیک کمترین مربعات، قید مسئله بهینه‌سازی را در تابع هدف جایگذاری می‌کنند. بدین ترتیب مدل خروجی با حل کردن دو دستگاه معادلات خطی بدست می‌آید. فرآیند یادگیری در این گونه روش‌ها برای مجموعه داده‌های بزرگ بسیار سریع است.

• دسته‌بندهای WLTSVM و RKNN-TSVM با در نظر گرفتن گراف برون کلاسی در مسائل بهینه‌سازی شان، نمونه‌های حاشیه‌ای هر کلاس را مشخص می‌کنند. بطوریکه ابرصفحه به جای تمام نمونه‌های کلاس مقابله، فقط از نمونه‌های حاشیه‌ای کلاس مقابله حداکثر فاصله ممکن را می‌گیرد. در نتیجه ابعاد مسئله بهینه‌سازی دوگان کوچکتر می‌شود و مرتبه زمانی دسته‌بند بهبود می‌یابد.

• روش پیشنهادی RKNN-TSVM همانند روش TBSVM ریسک ساختاری را کمینه می‌کند. بطوریکه مسائل بهینه‌سازی در این روش‌ها مثبت معین است و از شرایط ماتریس منفرد جلوگیری شده است. با این حال کمینه کردن ریسک ساختاری منجر به معرفی یک پارامتر جدید به دسته‌بند می‌شود. از طرفی تنظیم کردن این پارامتر تعمیم‌پذیری مدل خروجی را بهبود داده است. از طرف دیگر، معرفی این پارامتر باعث افزایش بار محاسباتی جستجوی پارامترهای بهینه می‌شود.

• روش WLTSVM براساس شمارش تعداد همسایه‌های نزدیک به یک نمونه وزن می‌دهد. در حالی که روش پیشنهادی RKNN-TSVM به یک نمونه براساس فاصله بین آن نمونه و نزدیک‌ترین همسایه‌هایی وزن می‌دهد. بطوریکه ماتریس وزن شامل اعداد بین بازه $[0, 1]$ است. این نوع شیوه وزن‌دهی باعث شناسایی بهتر نمونه‌های پرتراکم در فضای ویژگی می‌شود و همچنین نمونه‌های پرت و نویزی بهتر از روش WLTSVM مشخص می‌شوند. نتایج نشان داده است که این نوع شیوه وزن‌دهی دقیق دسته‌بند را بهبود می‌دهد.

• با وجود اینکه روش پیشنهادی RKNN-TSVM با در نظر گرفتن نمونه‌های حاشیه‌ای، دو مسئله دوگان با ابعاد کوچک‌تر را حل می‌کند. محاسبه گراف نزدیک‌ترین همسایه سرعت یادگیری روش پیشنهادی را به طور قابل توجه‌ای کاهش می‌دهد. الگوریتم LDMDBA به منظور بهبود سرعت فرآیند ساخت گراف نزدیک‌ترین همسایه بکار گرفته شده است. نتایج آزمایش بر روی مجموعه داده‌های بزرگ نشان می‌دهد که این الگوریتم سرعت یادگیری روش پیشنهادی را تا ۱۴ برابر بهبود داده است. همچنین این الگوریتم دقیق دسته‌بند پیشنهادی RKNN-TSVM را در بعضی از مجموعه

داده‌ها افزایش داده است.

۴-۶ پیشنهادها

موارد زیر به عنوان موضوع پژوهش‌های آینده پیشنهاد می‌شود.

- دسته‌بند KNN-LSTSVM از الگوریتم FSA برای ساخت گراف نزدیک‌ترین همسایه استفاده می‌کند. بطوریکه سرعت یادگیری این دسته‌بند برای مجموعه داده‌های بزرگ به طور محسوسی کاهش می‌یابد. بکارگیری الگوریتم LDMDBA می‌تواند سرعت یادگیری دسته‌بند را به طور قابل توجه‌ای بهبود دهد.
- چهار پارامتر برای دست‌یابی به دقت مطلوب در نسخه خطی روش پیشنهادی RKNN-TSVM باید تنظیم گردد. بطوریکه تعداد پارامترها در نسخه غیر خطی به ۵ افزایش می‌یابد. بنابراین پیدا کردن پارامترهای بهینه بسیار زمان‌بر است. بکارگیری الگوریتم‌های جستجو مانند الگوریتم‌های تکاملی می‌تواند فرایند جستجوی پارامترهای بهینه را بهبود دهد.
- معکوس ماتریس برای بدست آوردن مدل خروجی در هر دو دسته‌بند پیشنهادی یعنی -KNN- LSTSVM و RKNN-TSVM اجتناب ناپذیر است. بطوریکه مرتبه زمانی معکوس ماتریس برابر با $O(m^3)$ است. بنابراین سرعت یادگیری دسته‌بند برای مجموعه داده‌های بسیار بزرگ به شدت کاهش می‌یابد. بدست آوردن مدل خروجی بدون عمل معکوس ماتریس می‌تواند موضوع پژوهش آینده باشد.
- دسته‌بندهای پیشنهادی را می‌توان برای مسائل مختلف مانند تشخیص بیماری‌ها و دسته‌بندی متن استفاده کرد. داده‌های واقعی در این گونه مسائل معمولاً دارای نویز و نمونه‌های پرت هستند. استفاده از دسته‌بندهای پیشنهادی می‌تواند دقت دسته‌بندی در مسائل ذکر شده را بهبود دهد.

مقالات‌های مستخرج از پایان‌نامه

مجله

- KNN-based least squares twin support vector machine for pattern classification, Applied Intelligence, vol.48, pp.4551–4564, Dec 2018
- An enhanced KNN-based twin support vector machine with stable learning rules, Neural Computing and Applications (Under review)
- LightTwinSVM: A simple and fast implementation of standard twin support vector machine classifier, The Journal of Open Source Software (Submitted).

• دسته‌بندی سبک‌های یادگیری با استفاده از ویژگی‌های رفتاری و ماشین بردار پشتیبان دو قلو، مجله فناوری آموزش (پذیرفته شده در آبان ۱۳۹۷)

کنفرانس

- نظرکاوی خودکار نقد فیلم‌ها با رویکرد مقاوم‌سازی ماشین بردار پشتیبان، چهارمین همایش ملی زبان‌شناسی رایانشی، تهران، پژوهشگاه علوم انسانی و مطالعات فرهنگی، بهمن ۱۳۹۶
- پیش‌بینی ابعاد شخصیتی یادگیرنده‌گان در محیط یادگیری الکترونیکی با استفاده از ماشین بردار پشتیبان دو قلو کمترین مربعات، دومین کنفرانس بین‌المللی پژوهش‌های دانش بنیان در مهندسی کامپیوتر و فناوری اطلاعات، تهران، دانشگاه علامه طباطبائی، شهریور ۱۳۹۶
- تحلیل احساس نظرات فیلم‌ها با استفاده از ماشین بردار پشتیبان دو قلو کمترین مربعات، اولین کنفرانس ملی اصول مهندسی برق و کامپیوتر، تهران، دانشگاه پیام نور، تیر ۱۳۹۶

مراجع

- [1] M. I. Jordan and T. M. Mitchell, "Machine learning: Trends, perspectives, and prospects," *Science*, vol.349, no.6245, pp.255–260, 2015.
- [2] S. Shalev-Shwartz and S. Ben-David. *Understanding machine learning: From theory to algorithms*. Cambridge university press, 2014.
- [3] S. B. Kotsiantis, I. Zaharakis, and P. Pintelas, "Supervised machine learning: A review of classification techniques," *Emerging artificial intelligence applications in computer engineering*, vol.160, pp.3–24, 2007.
- [4] C. Cortes and V. Vapnik, "Support-vector networks," *Machine learning*, vol.20, no.3, pp.273–297, 1995.
- [5] V. N. Vapnik and V. Vapnik. *Statistical learning theory*, vol.1. Wiley New York, 1998.
- [6] J. A. Nasiri, M. Naghibzadeh, H. S. Yazdi, and B. Naghibzadeh, "Ecg arrhythmia classification with support vector machines and genetic algorithm," in *Computer Modeling and Simulation, 2009. EMS'09. Third UKSim European Symposium on*, pp.187–192, IEEE.
- [7] M. G. Raman, N. Somu, K. Kirthivasan, R. Liscano, and V. S. Sriram, "An efficient intrusion detection system based on hypergraph-genetic algorithm for parameter optimization and feature selection in support vector machine," *Knowledge-Based Systems*, vol.134, pp.1–12, 2017.
- [8] L. H. Lee, C. H. Wan, R. Rajkumar, and D. Isa, "An enhanced support vector machine classification framework by using euclidean distance function for text document categorization," *Applied Intelligence*, vol.37, no.1, pp.80–99, 2012.
- [9] A.-Z. Ala'M, H. Faris, and M. A. Hassonah, "Evolving support vector machines using whale optimization algorithm for spam profiles detection on online social networks in different lingual contexts," *Knowledge-Based Systems*, vol.153, pp.91–104, 2018.
- [10] J. Nayak, B. Naik, and H. Behera, "A comprehensive survey on support vector machine in data mining tasks: applications and challenges," *International Journal of Database Theory and Application*, vol.8, no.1, pp.169–186, 2015.
- [11] Jayadeva, R. Khemchandani, and S. Chandra, "Twin support vector machines for pattern classification," *IEEE Transactions on pattern analysis and machine intelligence*, vol.29, no.5, 2007.
- [12] M. A. Kumar and M. Gopal, "Least squares twin support vector machines for pattern classification," *Expert Systems with Applications*, vol.36, no.4, pp.7535–7543, 2009.
- [13] Y.-H. Shao, C.-H. Zhang, X.-B. Wang, and N.-Y. Deng, "Improvements on twin support vector machines," *IEEE transactions on neural networks*, vol.22, no.6, pp.962–968, 2011.
- [14] Q. Ye, C. Zhao, S. Gao, and H. Zheng, "Weighted twin support vector machines with local information and its application," *Neural Networks*, vol.35, pp.31–39, 2012.

- [15] A. Mir and J. A. Nasiri, "Knn-based least squares twin support vector machine for pattern classification," *Applied Intelligence*, vol.48, pp.4551–4564, Dec 2018.
- [16] S. Xia, Z. Xiong, Y. Luo, L. Dong, and G. Zhang, "Location difference of multiple distances based k-nearest neighbors algorithm," *Knowledge-Based Systems*, vol.90, pp.99–110, 2015.
- [17] C.-W. Hsu and C.-J. Lin, "A comparison of methods for multiclass support vector machines," *IEEE transactions on Neural Networks*, vol.13, no.2, pp.415–425, 2002.
- [18] A. Shigeo, "Support vector machines for pattern classification," *Advances in Pattern Recognition*, Springer, Heidelberg, 2005.
- [19] J. C. Platt, N. Cristianini, and J. Shawe-Taylor, "Large margin dags for multiclass classification," in *Advances in neural information processing systems*, pp.547–553, 2000.
- [20] J. A. Suykens and J. Vandewalle, "Least squares support vector machine classifiers," *Neural processing letters*, vol.9, no.3, pp.293–300, 1999.
- [21] O. L. Mangasarian and E. W. Wild, "Proximal support vector machine classifiers," in *Proceedings KDD-2001: Knowledge Discovery and Data Mining*, Citeseer.
- [22] C.-F. Lin and S.-D. Wang, "Fuzzy support vector machines," *IEEE Transactions on neural networks*, vol.13, no.2, pp.464–471, 2002.
- [23] O. L. Mangasarian and E. W. Wild, "Multisurface proximal support vector machine classification via generalized eigenvalues," *IEEE transactions on pattern analysis and machine intelligence*, vol.28, no.1, pp.69–74, 2006.
- [24] G. Melki, V. Kecman, S. Ventura, and A. Cano, "Ollawv: Online learning algorithm using worst-violators," *Applied Soft Computing*, vol.66, pp.384–393, 2018.
- [25] S. Ding, J. Yu, B. Qi, and H. Huang, "An overview on twin support vector machines," *Artificial Intelligence Review*, vol.42, no.2, pp.245–252, 2014.
- [26] S. Ding, N. Zhang, X. Zhang, and F. Wu, "Twin support vector machine: theory, algorithm and applications," *Neural Computing and Applications*, vol.28, no.11, pp.3119–3130, 2017.
- [27] H. Huang, X. Wei, and Y. Zhou, "Twin support vector machines: A survey," *Neurocomputing*, vol.300, pp.34–43, 2018.
- [28] Z. Qi, Y. Tian, and Y. Shi, "Structural twin support vector machine for classification," *Knowledge-Based Systems*, vol.43, pp.74–81, 2013.
- [29] J. A. Nasiri, N. M. Charkari, and K. Mozafari, "Energy-based model of least squares twin support vector machines for human action recognition," *Signal Processing*, vol.104, pp.248–257, 2014.
- [30] X. Pan, Y. Luo, and Y. Xu, "K-nearest neighbor based structural twin support vector machine," *Knowledge-Based Systems*, vol.88, pp.34–44, 2015.
- [31] Y. Xu, "K-nearest neighbor-based weighted multi-class twin support vector machine," *Neurocomputing*, vol.205, pp.430–438, 2016.
- [32] X. Pang, C. Xu, and Y. Xu, "Scaling knn multi-class twin support vector machine via safe instance reduction," *Knowledge-Based Systems*, vol.148, pp.17–30, 2018.
- [33] G. H. Golub and C. F. Van Loan. *Matrix computations*, vol.3. JHU Press, 2012.
- [34] J. H. Friedman, J. L. Bentley, and R. A. Finkel, "An algorithm for finding best matches in logarithmic expected time," *ACM Transactions on Mathematical Software (TOMS)*, vol.3, no.3, pp.209–226, 1977.

- [35] Y.-S. Chen, Y.-P. Hung, T.-F. Yen, and C.-S. Fuh, “Fast and versatile algorithm for nearest neighbor search based on a lower bound tree,” *Pattern Recognition*, vol.40, no.2, pp.360–375, 2007.
- [36] S. A. Dudani, “The distance-weighted k-nearest-neighbor rule,” *IEEE Transactions on Systems, Man, and Cybernetics*, no.4, pp.325–327, 1976.
- [37] J. Gou, L. Du, Y. Zhang, and T. Xiong, “A new distance-weighted k-nearest neighbor classifier,” *J. Inf. Comput. Sci.*, vol.9, no.6, pp.1429–1436, 2012.
- [38] S. Sra, S. Nowozin, and S. J. Wright. *Optimization for machine learning*. Mit Press, 2012.
- [39] O. L. Mangasarian and D. R. Musicant, “Successive overrelaxation for support vector machines,” *IEEE Transactions on Neural Networks*, vol.10, no.5, pp.1032–1037, 1999.
- [40] C.-J. Hsieh, K.-W. Chang, C.-J. Lin, S. S. Keerthi, and S. Sundararajan, “A dual coordinate descent method for large-scale linear svm,” in *Proceedings of the 25th international conference on Machine learning*, pp.408–415, ACM, 2008.
- [41] X. Peng, D. Chen, and L. Kong, “A clipping dual coordinate descent algorithm for solving support vector machines,” *Knowledge-Based Systems*, vol.71, pp.266–278, 2014.
- [42] V. L. Ceder, K. McDonald, and D. D. Harms. *The quick Python book*. Manning, 2010.
- [43] S. v. d. Walt, S. C. Colbert, and G. Varoquaux, “The numpy array: a structure for efficient numerical computation,” *Computing in Science and Engineering*, vol.13, no.2, pp.22–30, 2011.
- [44] E. Jones, T. Oliphant, and P. Peterson, “SciPy: open source scientific tools for Python,” 2014.
- [45] S. Behnel, R. Bradshaw, C. Citro, L. Dalcin, D. S. Seljebotn, and K. Smith, “Cython: The best of both worlds,” *Computing in Science and Engineering*, vol.13, no.2, pp.31–39, 2011.
- [46] B. D. Ripley. *Pattern recognition and neural networks*. Cambridge: Cambridge university press, 1st ed. , 2007.
- [47] T. Ho and E. Kleinberg, “Checkerboard dataset,” 1996.
- [48] C. M. Bishop. *Pattern recognition and machine learning*. New York : Springer, [2006] ©2006, 2006.
- [49] J. Demšar, “Statistical comparisons of classifiers over multiple data sets,” *Journal of Machine learning research*, vol.7, no.Jan, pp.1–30, 2006.
- [50] D. Musicant, “Ndc: normally distributed clustered datasets,” *Computer Sciences Department, University of Wisconsin, Madison*, 1998.
- [51] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, and V. Dubourg, “Scikit-learn: Machine learning in python,” *Journal of machine learning research*, vol.12, no.Oct, pp.2825–2830, 2011.

واژه‌نامه انگلیسی به فارسی

G

Generalization	تعمیم‌پذیری
Gradient Descent	گرادیان نزولی
Grid search	جستجوی شبکه‌ای

B

Bayes	بیز
Gradient Descent	گرادیان نزولی
Grid search	جستجوی شبکه‌ای

C

Classification	دسته‌بندی
Clustering	خوشه‌بندی
Hard margin	حاشیه سخت
Hyperplane	ابرصفحه
Hypersurface	ابرسطح

D

Decision Tree	درخت تصمیم
Dual problem	مسئله دوگان
Inter-class	برون کلاسی
Interior-point	نقطه داخلی
Intra-class	درون کلاسی

E Empirical Risk

K

Kernel trick	حقه‌ی هسته
Lagrange	لاگرانژ

F Feature space

S Least Squares کمترین مربعات

Sigmoid سیگموئید

Singular منفرد

Slack variable متغیر کمکی

Soft margin حاشیه نرم

Spam هرزنامه

Sparse خلوت

Structural Risk ریسک ساختاری

Supervised بانظارت

M

Machine Learning یادگیری ماشین

Margin point نمونه حاشیه‌ای

N

Nearest Neighbor نزدیک‌ترین همسایه

Neural Networks شبکه‌های عصبی

U

Unclassifiable غیر قابل دسته‌بندی

Unsupervised بی‌نظرارت

O

Optimization بهینه‌سازی

Outlier پرت

Overfitting برآش بیش از حد

V

Validation set مجموعه داده صحبت

Voting رای‌گیری

P

Primal problem مسئله اصلی

R

Rectangular kernel هسته مستطیلی

Regularization رگولاریزیون

واژه‌نامه فارسی به انگلیسی

ح

Hard margin	حاشیه سخت	Hypersurface
Soft margin	حاشیه نرم	Hyperplane
Kernel trick	حقه‌ی هسته	

ا

خ

Sparse	خلوت	Overfitting
Clustering	خوشه‌بندی	Suppor Vector
Decision Tree	درخت تصمیم	Inter-class
Intra-class	درون کلاسی	Optimization
Classification	دسته‌بندی	Bayes

د

Voting	رأی‌گیری
Regularization	رگولارسیون
Structural Risk	ریسک ساختاری

ر

Grid search	جستجوی شبکه‌ای
Sigmoid	سیگموئید
Generalization	تعیین‌پذیری

س

ج

Interior-point	نقطه داخلی	ش
Margin point	نمونه حاشیه‌ای	شبکه‌های عصبی

ه

Spam	هرزنامه	غیر قابل دسته‌بندی
Rectangular kernel	هسته‌ی مستطیلی	

ف

Machine Learning	یادگیری ماشین	فضای ویژگی
		Feature space

ک

Least Squares	کمترین مربعات
---------------------	---------------------

گ

Gradient Descent	گرادیان نزولی
------------------------	---------------------

ل

Lagrange	لاگرانژ
----------------	---------------

م

Slack variable	متغیر کمکی
Validation set	مجموعه داده صحت
Convex	محدب
Primal problem	مسئله اصلی
Dual problem	مسئله دوگان
Singular	منفرد

ن

Nearest Neighbor	نزدیک‌ترین همسایه
------------------------	-------------------------

فهرست اختصارات

C

clipDCD Clipping Dual Coordinate Descent

D

DAG Directed Acyclic Graph
DCD Dual Coordinate Descent

E

ELS-TSVM Energy-based Model of Least Squares Twin Support Vector Machine

F

FSA Full Search Algorithm
FSVM Fuzzy Support Vector Machine

G

GCC GNU Compiler Collection
GEPSVM Generalized Eigenvalue Proximal Support Vector Machine

K

k-d tree K-dimensional tree
KKT Karush-Kuhn-Tucker
KNN-LSTSVM KNN-based Least Squares Twin Support Vector Machine
KNN-STSV K-nearest neighbor based structural twin support vector machine
KWM-TSVM K-nearest neighbor-based weighted multi-class twin support vector machine

L

LB Lower bound tree
LDMDBA Location difference of multiple distances based k-nearest neighbors algorithm
LS-SVM Least squares support vector machines
LS-TSVM Least Squares Twin Support Vector Machine

O

OLLA WV Online Learning Algorithm using Worst-Violators

P

PSVM Proximal Support Vector Machine

Q

QPP Quadratic Programming Problem

R

RBF Radial basis function

RKNN-TSVM Regularized KNN-based Twin Support Vector Machine

S

SMW Sherman-Morrison-Woodbury

SOR Successive Overrelaxation

STSVM Structural Twin Support Vector Machine

SVM Support Vector Machine

T

TBSVM Twin Bounded Support Vector Machine

TSVM Twin Support Vector Machine

W

WLTSVM Weighted Twin Support Vector Machine with Local Information

Abstract

In the past decade, machine learning has been used for solving problems with complex patterns. Classification is one of the main learning types which solves problems such as face recognition, text classification, and disease recognition. Support vector machine (SVM) is a state-of-the-art classification method which has good generalization and accuracy. In recent years, classifiers based on SVM have been proposed. Among these are twin support vector machine (TSVM) which has received more attention. The central idea of TSVM is to find two non-parallel hyperplanes for binary classification. Therefore, it solves two smaller-sized Quadratic Programming Problems (QPPs) as opposed to one large QPP in standard SVM. As a result, TSVM is four times faster than standard SVM in theory. Even though TSVM has better prediction accuracy and time complexity than SVM, it has several drawbacks such as high sensitivity to outliers and noise, overfitting, and high computational cost for large datasets. In this study with the aim of addressing the drawbacks, we propose two classifiers, called KNN-based least squares twin support vector machine (KNN-LSTSVM) and a regularized KNN-based twin support vector machine (RKNN-TSVM). The proposed KNN-LSTSVM and RKNN-TSVM classifiers construct K-nearest neighbor graph to give weight to each training samples. Also, margin points of each class are determined using these graphs. This further makes the proposed classifiers more robust to noise and outliers than standard TSVM. The proposed classifiers were comprehensively evaluated on synthetic and benchmark datasets. The experimental results validate the effectiveness of proposed KNN-LTSVM and RKNN-TSVM in terms of classification accuracy and computational time.

Keywords: Twin support vector machine, Nearest neighbor graph, Least squares, Structural risk, Classification

ISLAMIC AZAD UNIVERSITY

North Tehran Branch

”M.Sc” Thesis

Research Title

Robust Twin Support Vector Machine for Noisy Data

Supervisor

Jalal A. Nasiri

Consulting Supervisor

Somayeh Fatahi

By

A. Mir

Winter 2019