فصلنامه علمی- پژوهشی
فناوری اطلاعات و ارتباطات ایران

سال هفتم، شماره‌های ۲۵ و ۲۶، پاییز و زمستان ۱۳۹۴
صص: ۱- ۱۴

دسته‌بندی داده‌های دو رده‌ای با ابرمستطیل موازی محورهای مختصات

زهرا مصلحی* مازیار پالهنگ**
* دانشجوی دکتری، دانشکده مهندسی برق و کامپیوتر، دانشگاه صنعتی اصفهان، اصفهان
** دانشیار، دانشکده مهندسی برق و کامپیوتر، دانشگاه صنعتی اصفهان، اصفهان
تاریخ دریافت: ۹۲/۱۱/۱۶ تاریخ پذیرش: ۹۴/۱۱/۲۰

چکیده
یکی از روش‌های یادگیری در یادگیری ماشین و شناسایی الگو، یادگیری با ناظر است. در یادگیری با ناظر و در مسایل دو رده‌ای، برچسب داده‌های آموزشی موجود و شامل دو رده مثبت و منفی می‌باشند. هدف الگوریتم یادگیری با ناظر، محاسبه فرضیه‌ای است که بتواند با کمترین مقدار خطا، داده‌های مثبت و منفی را از یکدیگر جدا کند. در این مقاله، از بین کلیه الگوریتم‌های یادگیری با ناظر، بر عملکرد درخت‌های تصمیم متمرکز می‌شویم. دیدگاه هندسی درخت تصمیم ما را به مفهوم تفکیک‌پذیری در هندسه محاسباتی نزدیک می‌کند. از بین کلیه الگوریتم‌های تفکیک‌پذیری موجود و مرتبط با درخت تصمیم، مساله محاسبه مستطیل با حداکثر اختلاف دو رنگ را مطرح می‌کنیم و الگوریتم را در یک، دو، سه و m بعد پیاده‌سازی می‌کنیم که m تعداد ویژگی‌های داده‌ها را نشان می‌دهد. نتیجه پیاده‌سازی نشان‌دهنده آن است که این الگوریتم، الگوریتمی قابل رقابت با الگوریتم شناخته شده C4.5 است.

واژه‌های کلیدی : یادگیری ماشین، دسته‌بندی، درخت تصمیم، هندسه محاسباتی، تفکیک‌پذیری، مستطیل

۱- مقدمه

یکی از زمینه‌های فعالیت در یادگیری ماشین و شناسایی الگو یادگیری با ناظر می‌باشد. مهمترین ویژگی الگوریتم‌های یادگیری با ناظر آن است که در آن برچسب داده‌های آموزشی موجود است. تاکنون انواع روش‌های یادگیری با ناظر پیشنهاد شده است. به عنوان مثال می‌توان به درخت‌های تصمیم، یادگیر SVM و روش‌های نزدیکترین همسایه اشاره کرد. در این مقاله بر عملکرد درخت‌های تصمیم متمرکز می‌شویم. عملکرد هندسی درخت‌های تصمیم با مفهوم تفکیک‌پذیری در هندسه محاسباتی ارتباط نزدیکی دارد. در ابتدا در مورد عملکرد درخت تصمیم صحبت می‌کنیم و پس از آن در مورد مساله تفکیک‌پذیری در هندسه محاسباتی صحبت خواهیم کرد و ارتباط بین درخت‌های تصمیم و مساله تفکیک‌پذیری را مطرح می‌کنیم.

درخت‌های تصمیم یکی از روش‌های یادگیری با ناظر برای رده‌بندی داده‌های پیوسته و یا داده‌های گسسته است. نحوه‌ی ساخت فرضیه متناظر با درخت‌های تصمیم به این صورت است که ابتدا بر اساس یک معیار مشخص یکی از ویژگی‌های متناظر با داده‌های ورودی انتخاب می‌شود و این ویژگی به عنوان ریشه در نظر گرفته می‌شود. تعداد زیرشاخه‌های متناظر با ریشه برابر با مقادیر مختلف ویژگی

نویسنده عهده‌دار مکاتبات: زهرا مصلحی z.moslehi@ec.iut.ac.ir

انتخاب شده خواهد بود. اگر داده‌های آموزشی پیوسته-مقدار باشند، زیر شاخه‌های متناظر با ریشه با انتخاب یک خصیصه و تقسیم محدوده مربوط به آن به دو و یا چند قسمت بدست می‌آید. حال می‌توان خصیصه دوم را انتخاب کرد و همین روند را مجددا تکرار کرد. در نهایت برگ‌های درخت، برچسب متناظر با رده داده‌ها را مشخص می‌کند. یکی از مهمترین ملاک‌های کارایی این درخت‌ها آن است که آن‌ها قابلیت تعمیم بالایی داشته باشند. به همین منظور لازم است در هر مرحله از ساخت، خصیصه‌ای به عنوان ریشه انتخاب شود که به ازای هریک از مقادیر آن بیشترین خلوص ایجاد گردد. ارزیابی میزان خلوص با معیار آنتروپی قابل اندازه‌گیری است [1]. حال برای انتساب یک برچسب مشخص به داده ورودی لازم است بر اساس مقادیر متناظر با ویژگی‌های آن داده یک مسیر از ریشه تا یکی از برگهای درخت پیموده شود و برچسب متناظر با داده آزمون محاسبه گردد.

حال به تعریف مساله تفکیک‌پذیری می‌پردازیم. در مساله تفکیک‌پذیری در هندسه محاسباتی، دو مجموعه نقطه قرمز و آبی داده می‌شود. هدف مساله تفکیک‌پذیری پایه، به دست آوردن شکل هندسی C به گونه‌ای است که بتواند مجموعه نقاط قرمز و آبی را از یکدیگر جدا کند. به عبارتی قصد داریم شکل هندسی C را به گونه‌ای در فضا قرار دهیم بطوریکه کلیه نقاط آبی (قرمز) داخل آن و کلیه نقاط قرمز (آبی) خارج از آن قرار گیرند. تنوعات زیادی برای مساله تفکیک‌پذیری مطرح شده است. تفاوت الگوریتم‌های ارائه شده در انتخاب شکل هندسی C و تابع هدف تعریف شده است.

تاکنون کارهای زیادی در زمینه تفکیک با مستطیل، دایره، مثلث، خط و گوه ارائه شده است. هدف مساله بهینه‌سازی، گاهی کم کردن حجم، محیط و مساحت شکل هندسی تفکیک‌کننده است. همچنین گاهی هدف، حداکثر کردن تعداد نقاط همرنگ داخل C است. این مساله زمانی مطرح می‌شود که مجموعه نقاط قرمز و آبی به طور کامل تفکیک‌پذیر نباشند.ارتباط روشنی بین مساله درخت تصمیم در یادگیری ماشین و مساله تفکیک‌پذیری نقاط در هندسه محاسباتی وجود دارد. تعبیر هندسی عملکرد درخت تصمیم به صورت زیر است: اگر تعداد ویژگی‌های داده‌ها برابر m باشد، کلیه داده‌ها، نقاطی در فضای m بعدی هستند که هر بعد بیانگر یک ویژگی از داده‌ها است. عملکرد درخت تصمیم مشابه پیدا کردن ابرصفحه‌های تقسیم‌کننده در فضای m بعدی است، بطوریکه داده‌های موجود را به درستی رده‌بندی کند و تا حد ممکن رده‌بندی صحیح داده‌های آینده را نیز بدست آورد. برای آنکه بتوانیم این مساله را به مساله تفکیک‌پذیری ارتباط دهیم کافی است به داده‌های با برچسب مثبت رنگ آبی و به داده‌های با برچسب منفی رنگ قرمز انتساب دهیم. عملکرد درخت تصمیم مشابه آن است که در فضای m بعدی بهترین تفکیک‌کننده ممکن را بیابیم بطوریکه فضا به چندین بخش افراز گردد و هر بخش دارای بیشترین تعداد نقاط همرنگ باشد.

۱-۱ کارهای مرتبط

در مسایل تفکیک‌پذیری، مطالعه وسیعی برای انتخاب مفیدترین الگو برای تفکیک مجموعه نقاط قرمز و آبی در [2] انجام شده است. در همین مرجع نشان داده شده است که یکی از الگوهای مفید مستطیل است. تاکنون، مقالات زیادی در زمینه تفکیک‌پذیری برای مستطیل ارائه شده است.تفکیک کامل با یک مستطیل در [3] مورد بررسی قرار گرفته است. در این مساله، مستطیل فقط دارای نقاط از یک رنگ است و نقاط از رنگ دیگر، خارج از مستطیل واقع می‌شوند. ممکن است تفکیک با یک مستطیل بطور کامل امکان‌پذیر نباشد. تفکیک با یک مستطیل با هدف حداکثر تعداد نقاط از یک رنگ داخل مستطیل، به نحوی که نقطه‌ای از رنگ دیگر، داخل مستطیل واقع نباشد در [4] مورد بررسی قرار گرفته است. تفکیک با یک مستطیل با هدف حداکثر اختلاف دو رنگ، بین نقاط از یک رنگ و رنگ دیگر داخل مستطیل نیز در [5,6] بررسی گردیده است (در این حالت مستطیل دارای نقاط از هر دو رنگ است).

تفکیک با دو مستطیل بطوریکه دو مستطیل به گونه‌ای در صفحه قرار گیرند که اضلاع آنها موازی یکدیگر بوده و نقاطی که در یک مستطیل قرار می‌گیرند (که در مستطیل دیگر قرار ندارند) تک رنگ باشند، مساله دیگری است که در [7,8] بررسی شده است. به عبارتی اگر R را مستطیل دربرگیرنده نقاط قرمز و B را مستطیل در برگیرنده نقاط

آبی تعریف کنیم، نقاط داخل $B-R$ باید فقط از رنگ آبی و نقاط داخل $R-B$ فقط از رنگ قرمز باشند. در این مساله هدف بیشینه کردن تعداد نقاط داخل $(R\cup B)-(R\cap B)$ است. یعنی نقاطی که در فصل مشترک دو مستطیل و خارج از دو مستطیل واقع می‌شوند از مجموعه نقاط حذف خواهند شد. همچنین، الگوریتم محاسبه دو مستطیل مجزا و موازی محورهای مختصات بطوریکه کلیه نقاط آبی داخل آن دو مستطیل و کلیه نقاط قرمز خارج از آن‌ها واقع شوند در [9,10] ارائه شده است. تفکیک با دو مربع واحد مجزا و تک رنگ موازی با محورهای مختصات، با هدف حفظ بزرگترین زیرمجموعه از نقاط در [11,12] بررسی گردیده است. در کل تفاوت الگوریتم‌های گوناگون برای تفکیک‌پذیری نقاط با مستطیل بر اساس تعداد مستطیل‌های استفاده شده، زاویه این مستطیل‌ها نسبت به یکدیگر و نسبت به محورهای مختصات و همچنین گاهی تابع بهینه‌سازی مطرح شده می‌باشد. توابع مختلف بر اساس مقدار خطای موجود بر اساس تعداد نقاطی که به اشتباه رده‌بندی می‌شوند، مساحت و یا محیط مستطیل‌ها تعریف می‌گردند.

تاکنون کارهای بسیار کمی از لحاظ کاربردی و در حوزه یادگیری ماشین و دسته‌بندی برای مسایل تفکیک‌پذیری در هندسه محاسباتی ارائه شده است. از بین کارهای موجود تنها می‌توان به کارهای انجام شده توسط دابکین[1] و همکارانش اشاره کرد [5,13]. در [13] تنها وابستگی برخی از این مسایل به مساله درخت تصمیم شرح داده می‌شود. به عنوان مثال دابکین عملکرد یک درخت تصمیم $\tau(1,k)$ را به این صورت تعریف می‌کند که کلیه مثال‌های آموزشی را در یک بعد k قسمت کنیم، بطوریکه در هر قسمت بیشترین تعداد نقاط همرنگ وجود داشته باشند. به عنوان مثالی دیگر، ابرمستطیل با حداکثر اختلاف دو رنگ به گونه‌ای متفاوت درخت تصمیم مربوطه را ایجاد می‌کند [5,13]. در اینجا هر بعد دقیقا به سه قسمت تقسیم می‌شود و در نهایت ابرمستطیل ایجاد شده بیشترین تعداد نقاط همرنگ را دارا خواهد بود.

1. Dobkin

۱-۲- نتایج بدست آمده

مهم‌ترین ویژگی این مقاله، توجه به مسایل هندسه محاسباتی در حوزه کاربردی می‌باشد. تاکنون در هیچ‌یک از کارهای انجام شده در زمینه تفکیک‌پذیری نقاط در هندسه محاسباتی، به پیاده‌سازی الگوریتم‌های موجود بر روی داده‌های واقعی و کاربرد عینی این مسایل در یادگیری ماشین پرداخته نشده است. نزدیکترین کارهای موجود به این مقاله، کارهای ارائه شده توسط دابکین و همکارانش می‌باشد که در آن چندین مساله تفکیک‌پذیری ارائه و تنها به کاربرد آن‌ها در زمینه درخت‌های تصمیم اشاره می‌گردد [5,13]. در هیچ‌یک از کارهای موجود در زمینه تفکیک‌پذیری به تحلیل الگوریتم از نظر دقت و سرعت و مقایسه با الگوریتم‌هایی که در دنیای واقعی اجرا می‌شوند پرداخته نمی‌شود. بنابراین، یکی از نقاط ضعف کارهای موجود عدم اجرای این الگوریتم‌ها بر روی داده‌های واقعی می‌باشد که در این مقاله به شدت مورد توجه می‌باشد. در این مقاله محاسبه ابرمستطیل با حداکثر اختلاف دو رنگ مورد توجه قرار می‌گیرد. این مساله را در یک، دو، سه و m بعد پیاده‌سازی می‌کنیم. هنگام پیاده‌سازی الگوریتم در بیش از یک بعد به گونه‌ای متفاوت از الگوریتم اصلی رفتار می‌کنیم تا بتوانیم یک ابرمستطیل جداکننده تقریبی که زمان ساخت آن پیچیدگی محاسباتی کمتری داشته باشد ایجاد کنیم. از بین کلیه آزمایش‌های انجام شده، بهترین نتیجه با پیاده‌سازی این مساله در یک بعد حاصل شد. نتایج پیاده‌سازی این مساله در یک بعد و تحلیل سرعت اجرای الگوریتم، در مقایسه با الگوریتم C4.5 بیانگر آن است که الگوریتم ارائه شده، الگوریتمی قابل رقابت با الگوریتم C4.5 است.

در ادامه، ابتدا در بخش ۲ الگوریتم C4.5 را معرفی می‌کنیم. پس از آن در بخش ۳ به معرفی ابرمستطیل با حداکثر اختلاف دو رنگ، ارتباط آن با درخت تصمیم و الگوریتم‌های مربوط به آن می‌پردازیم. پیاده‌سازی آزمایش‌های مختلف و نتایج آن‌ها در بخش ۴ آورده می‌شود. این نتایج با نتایج بدست آمده توسط الگوریتم C4.5 مقایسه می‌گردد. در آخر در بخش ۵ نیز نتیجه‌گیری و مسایل باز آورده می‌شود.

۲- الگوریتم C4.5

در این بخش به معرفی الگوریتم یادگیری درخت تصمیم C4.5 می‌پردازیم. محاسبه فرضیه متناظر با الگوریتم C4.5 مشابه چیزی است که در قسمت مقدمه به آن اشاره شد. به عبارتی برای ساخت درخت C4.5 مشابه درخت‌های تصمیم پایه، به ترتیب یک ویژگی انتخاب می‌کنیم و انشعاب‌های متناظر با آن را ایجاد می‌کنیم و اینکار را تا زمانی ادامه می‌دهیم که کلیه داده‌های متناظر با هر برگ به خلوص کافی دست یابند. منتها الگوریتم‌های درخت تصمیم پایه، فاقد برخی از ویژگی‌ها از قبیل قابلیت برخورد با ویژگی‌های پیوسته- مقدار، قابلیت برخورد با مشکل بیش پوشش[۲]، انتخاب یک ویژگی مناسب در هر سطح از درخت، قابلیت برخورد با داده‌های آموزشی با برخی مقادیر ویژگی‌های نامعلوم[۳] و قابلیت توسعه کارایی محاسباتی الگوریتم می‌باشند. اغلب این موارد در الگوریتم C4.5 لحاظ شده است [1]. از این الگوریتم در انجام آزمایش‌ها، به عنوان یک معیار مقایسه استفاده می‌گردد.

۳- ابرمستطیل با حداکثر اختلاف دو رنگ

در این بخش با مفهوم ابرمستطیل با حداکثر اختلاف دو رنگ آشنا می‌شویم. ابتدا در یک زیر بخش به تعریف رسمی ابرمستطیل با حداکثر اختلاف دو رنگ می‌پردازیم. پس از آن در زیر بخش‌های جداگانه به معرفی الگوریتم در ابعاد مختلف خواهیم پرداخت.

۳-۱- تعریف و ارتباط با درخت‌های تصمیم

ابتدا مفهوم ابرمستطیل با حداکثر اختلاف دو رنگ را به صورت رسمی مطرح می‌کنیم و سپس ارتباط آن با درخت تصمیم را با رسم شکل نشان می‌دهیم.

فرض کنید داده‌های آموزشی به همراه برچسب‌های آن‌ها (برچسب‌های مثبت و منفی) داده شده است. این داده‌ها، مجموعه نقاط S در فضای m بعدی را تشکیل می‌دهند. برچسب هریک از داده‌ها به کمک رنگ آن‌ها نشان داده می‌شود. بنابراین، می‌توان داده‌های آموزشی را با رنگ‌های قرمز و آبی مجزا کرد. در مساله محاسبه مستطیل با حداکثر اختلاف دو رنگ، هدف، حداکثرسازی درجه خلوص داده‌های آموزشی پوشش داده شده با یک فرضیه مستطیل‌شکل و موازی محورهای مختصات است. منظور از حداکثر خلوص، محاسبه مستطیلی است که بیشترین نقاط آبی و کمترین نقاط قرمز را داراست. درنتیجه ابرمستطیل بدست آمده دارای بیشترین اختلاف دو رنگ است. به شکل ۱ توجه کنید. در این شکل نقاط قرمز و آبی به ترتیب با علامت × و ● مشخص شده‌اند.

چنانچه بخواهیم این مساله را با عبارات ریاضی مدل کنیم به روابط زیر می‌رسیم. ابتدا از تابع نگاشت رابطه (۱) برای مشخص کردن هریک از داده‌های مثبت و منفی استفاده می‌کنیم.

$$X : S \rightarrow \{-1,+1\} \quad (1)$$

سپس $\Delta(B)$ به صورت زیر تعریف می‌شود:

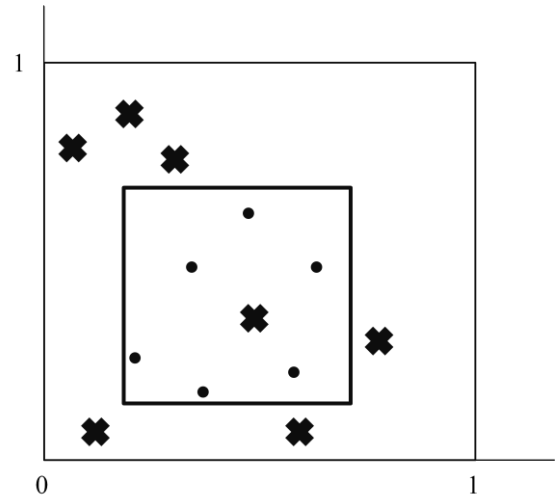
$$\Delta(B) = \sum_{x \in (B \cap S)} X(x) \quad (2)$$

در واقع $\Delta(B)$ به نوعی بیانگر اختلاف تعداد داده‌های مثبت پوشیده شده توسط فرضیه مستطیل- شکل B و داده‌های منفی به اشتباه پوشیده شده توسط این فرضیه است. در مساله محاسبه ابرمستطیل با حداکثر تمایز دو رنگ به دنبال برآورده کردن رابطه (۳) هستیم.

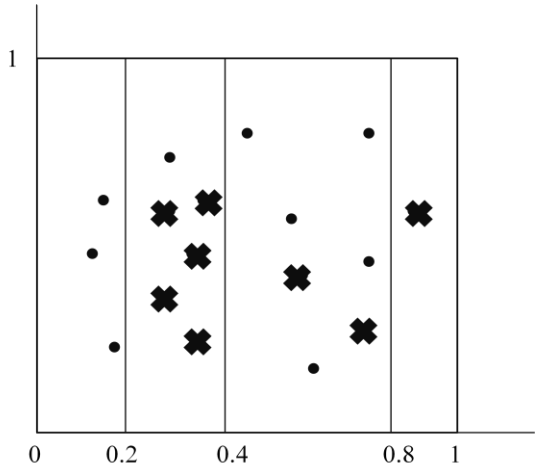
$$\arg\max_{B} |\Delta(B)| \quad (3)$$

ابرمستطیلی را در نظر بگیرید که در رابطه بالا صدق می‌کند. یعنی از بین کلیه ابرمستطیل‌ها، ابرمستطیلی داریم که بیشترین تعداد نقاط آبی و کمترین تعداد نقاط قرمز را پوشش می‌دهد. ابرمستطیلی که دارای بیشترین تمایز دو رنگ است، تعداد نقاط قرمز داخل ابرمستطیل و نقاط آبی خارج از آن را کمینه می‌کند. نقاط قرمز داخل ابرمستطیل به همراه نقاط آبی خارج از آن، معادل $Err_s(h)$ است. منظور از $Err_s(h)$ خطای نمونه برای کلیه فرضیه‌های مستطیل- شکل است. بنابراین، بدست آوردن ابرمستطیل با حداکثر تمایز دو رنگ معادل پیداکردن فرضیه‌ای با حداقل خطای $Err_s(h)$ است.

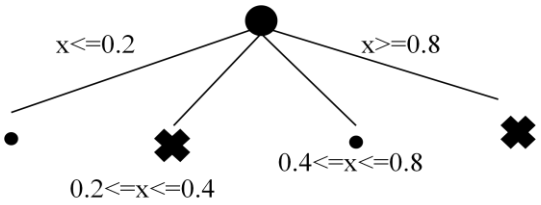
2. Overfitting
3. Missing value



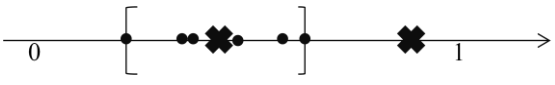
شکل ۱: محاسبه مستطیل با حداکثر تمایز دو رنگ (در این شکل نقاط قرمز با علامت × مشخص شده‌اند) [5]



شکل ۲: مرزهای جداکننده درخت تصمیم $\tau(1,k)$



شکل ۳: درخت تصمیم متناظر با شکل ۲



شکل ۴: نگاشت نقاط مابین دو خط افقی روی یک خط

قضیه ۱- از بین کلیه فرضیه‌های مستطیل- شکل که داده‌های آموزشی را رده‌بندی می‌کند، مستطیلی که دارای بیشترین تمایز دو رنگ است دارای کمترین مقدار $Err_S(h)$ است [5].لازم است اشاره کنیم اگر داده‌های آزمون به خوبی از توزیع جامعه انتخاب شوند، $Err_S(h)$ تقریب خوبی از $Err_D(h)$ خواهد بود. $Err_D(h)$ خطای واقعی فرضیه h را نشان می‌دهد. بنابراین، از بین کلیه فرضیه‌های مستطیل- شکل به دنبال مستطیل با بیشترین تمایز دو رنگ خواهیم بود. برای روشن شدن ارتباط بین درخت تصمیم و ابرمستطیل با حداکثر اختلاف دو رنگ ابتدا به شکل ۲ توجه کنید. در این شکل(شماره ۲) مرزهای جداکننده درخت تصمیم $\tau(1,k)$ نشان داده شده است.درخت تصمیم یک‌سطحی معادل آن در شکل ۳ آورده شده است.به صورت مشابه، بازه با حداکثر اختلاف دو رنگ را می‌توان معادل یک درخت تصمیم یک‌سطحی دانست بطوریکه ریشه دارای سه انشعاب برای مقادیر کمتر از ابتدای بازه، مقادیر مابین ابتدا و انتهای بازه و مقادیر بیشتر از انتهای بازه است. به همین ترتیب مستطیل، معادل یک درخت تصمیم دوسطحی و ابرمستطیل m بعدی، معادل یک درخت تصمیم m سطحی با حداکثر سه انشعاب در هر گره است. پیدا کردن ابرمستطیل با حداکثر اختلاف دو رنگ معادل بدست آوردن هریک از انشعاب‌های درخت تصمیم متناظر آن است.

۳-۲- الگوریتم در حالت یک بعدی

در حالت یک‌بعدی فرض می‌شود کلیه داده‌های آموزشی تنها دارای یک خصیصه هستند. در غیر اینصورت می‌توان یکی از خصایص آن‌ها را با استفاده از روش‌های انتخاب خصیصه[۴] انتخاب کرد. در هر حال شناسایی رده مربوط به هر داده، تنها با استفاده از یکی از خصایص آن‌ها انجام می‌شود. در اینجا ابتدا کلیه مثال‌های آموزشی روی محور متناظر با خصیصه انتخابی تصویر می‌شوند. منظور از ابرمستطیل با حداکثر اختلاف دو رنگ، بازه‌ای است که بیشترین تعداد نقاط آبی و کمترین تعداد نقاط قرمز را داراست. به شکل ۴ توجه کنید.

الگوریتم در حالت یک‌بعدی به زمان پیش‌پردازش $O(n\log n)$ نیاز دارد که در آن n تعداد کل داده‌های آموزشی است [5]. این مرتبه زمانی مربوط به مرتب‌سازی کلیه داده‌های آموزشی روی محور با خصیصه انتخابی است. فرض کنید پس از تصویر کلیه داده‌های آموزشی روی محور مربوطه، کلیه داده‌ها در بازه بسته $[0,1]$ قرار گیرند.

حال تعاریف زیر را در نظر بگیرید:

- $A_{\max}$: زیربازه‌ای از A است که از بین کلیه زیربازه‌های A، دارای بیشترین مقدار Δ است. بنابراین، اگر کلیه نقاط در بازه $[0,1]$ محدود شده باشند ما به دنبال $[0,1]_{\max}$ هستیم.
- A_{left}: زیربازه‌ای از A است که از میان کلیه زیربازه‌های A که نقطه ابتدای بازه آن‌ها با نقطه ابتدای بازه A برابر است، دارای بیشترین مقدار Δ است.
- A_{right}: زیربازه‌ای است که از میان کلیه زیربازه‌های A که نقطه انتهای آن‌ها با نقطه انتهای بازه A برابر است دارای بیشترین مقدار Δ است.

مساله محاسبه بازه با بیشترین اختلاف دو رنگ به کمک یک درخت و با استفاده از روش تقسیم و حل پیاده‌سازی می‌شود. ابتدا در مورد راهکار تقسیم و حل بکار گرفته شده صحبت می‌کنیم و سپس اجرای الگوریتم به کمک درخت ذکر شده در الگوریتم یک آورده می‌شود.

در ابتدا بازه $[0,1]$ را به $\frac{n}{2}$ زیربازه افراز می‌کنیم که در آن n تعداد کل داده‌های آموزشی است و در هر زیربازه حداکثر دو داده قرار می‌گیرد. بنابراین، زیربازه‌های ایجاد شده همه داده‌های آموزشی را پوشش می‌دهند و اشتراک هر دو زیربازه با یکدیگر تهی است. برای هر زیربازه ایجاد شده، سه زیربازه بیشینه[۵] گفته شده را محاسبه می‌کنیم. سپس، از چپ به راست L و R را دو زیربازه متوالی غیرهمپوشان تعریف می‌کنیم. اگر این دو زیربازه را با یکدیگر ترکیب کنیم، بازه LR بدست می‌آید. واضح است که $LR_{\max}$ برابر

4. Feature selection

5. Maximal

$L_{\max}$ یا $R_{\max}$ و یا $L_{right} \cup R_{left}$ خواهد بود. اشاره اختلاف دو رنگ پس از ترکیب L و R است. از طرفی LR_{left} برابر L_{left} و یا $L \cup R_{left}$ است. همچنین، LR_{right} برابر R_{right} و یا $L_{right} \cup R$ است. حال L و R جدید را دو زیر بازه متوالی بعد از L و R قبلی در نظر می‌گیریم و روند بالا را برای آن‌ها تکرار می‌کنیم. این کار را تا زمانی که تمامی $\frac{n}{2}$ زیر بازه ایجاد شده پیمایش شوند تکرار می‌کنیم.

می‌کنیم منظور از $LR_{\max}$ بهترین بازه موجود با بیشترین به این ترتیب به تعداد $\frac{n}{4}$ زیر بازه LR به همراه زیربازه‌های بیشینه آن‌ها محاسبه می‌گردد. حال $\frac{n}{4}$ زیر بازه جدید را در نظر می‌گیریم و کل مراحل الگوریتم گفته شده را بر روی آن‌ها تکرار می‌کنیم. این روند تا زمانی ادامه می‌یابد که تنها بازه $[0,1]$ مشاهده شود و سه زیربازه بیشنه متناظر با آن محاسبه گردد.

الگوریتم ۱- محاسبه بازه با حداکثر اختلاف دو رنگ

ورودی: تصویر کلیه داده‌های آموزشی روی محور متناظر با خصیصه انتخابی و مقیاس شده در بازه $[0,1]$

خروجی: بازه با حداکثر اختلاف دورنگ $[0,1]_{\max}$

۱- یک درخت دودویی به ارتفاع $\log_2(n)$ ایجاد کن. این درخت شامل $\frac{n}{2}$ برگ می‌باشد.

۲- بازه $[0,1]$ را به $\frac{n}{2}$ زیر بازه افراز کن. هر زیربازه دارای حداکثر دو نقطه از مجموعه S خواهد بود. از طرفی افراز به نحوی انجام می‌شود که نقاط ابتدا و انتهای بازه روی نقاط متناظر با مجموعه S واقع نشود. زیربازه‌های بدست آمده را در برگ‌های درخت درج کن. برای هر زیربازه A، سه زیربازه بیشینه A_{left} ، A_{right} و $A_{\max}$ را محاسبه کن و آن‌ها را در گره متناظر با زیربازه A در درخت ذخیره کن. بنابراین، هر گره از درخت شامل زیر بازه A و زیر بازه‌های A_{left} ، A_{right} و $A_{\max}$ می‌باشد.

۳- هر دو زیربازه متوالی را با یکدیگر ادغام کن. به این ترتیب گره والد مربوط به دو زیر درخت متناظر با آن دو زیربازه مقداردهی می‌شود. پس از عملیات ادغام، سه بیشینه گفته شده را به کمک بیشینه‌های ذخیره شده در فرزندان آن بدست آور و آن‌ها را در گره مربوطه ذخیره کن. واضح است که تعداد گره‌های این سطح از درخت نصف تعداد گره‌های سطح پایینی است.

۴- پس از ادغام زیربازه‌ها اگر بازه $[0,1]$ بدست آمد $[0,1]_{\max}$ را بازگردان. درغیراینصورت به مرحله ۳ برو.

الگوریتم ۱: محاسبه بازه با حداکثر اختلاف دو رنگ. اقتباس شده از [5]

حال به بیان شبه کد مربوط به الگوریتم محاسبه $[0,1]_{max}$ می‌پردازیم. در این شبه کد از یک درخت دودویی برای اجرای الگوریتم استفاده می‌کنیم. به الگوریتم ۱ توجه کنید.

قضیه ۲- اگر کلیه نقاط در مرحله پیش‌پردازش الگوریتم مرتب شده باشند، بدست آوردن بازه‌ای که دارای بیشترین اختلاف دو رنگ است در زمان خطی امکان‌پذیر است [5].

۳-۳- الگوریتم در حالت ۱ بعدی پویا

الگوریتم به سادگی می‌تواند با کمی تغییرات در حالت پویا نیز استفاده گردد. یعنی به سادگی می‌تواند درج و حذف را نیز پشتیبانی کند. کافی است یک درخت دودویی روی کلیه بازه‌ها مشابه قبل ساخته شود. برای حذف یک داده از مجموعه داده‌ها لازم است یک مسیر از ریشه به برگ پیموده شده و برگ متناظر با آن داده را بدست آوریم. سپس به ازای کلیه گره‌های موجود در مسیر پیموده شده، لازم است سه بیشینه مربوط به آن را بروز کنیم.

قضیه ۳- بدست آوردن زیربازه با حداکثر اختلاف دو رنگ، پس از هر عمل درج و یا حذف یک نقطه از مجموعه نقاط در زمان $O(\log n)$ امکان‌پذیر است [5].

پیچیدگی زمانی پایین الگوریتم برای حالت ۱بعدی پویا به ما کمک می‌کند که بتوانیم از این الگوریتم در الگوریتم‌های یادگیری برخط[6] نیز استفاده کنیم.

۳-۴- الگوریتم در حالت d بعدی

الگوریتم در حالت یک‌بعدی قابل تعمیم به ابعاد بالاتر نیز خواهد بود. محاسبه مستطیل با حداکثر اختلاف دورنگ در حالت دوبعدی با استفاده از تکنیک خط جاروب [14] و الگوریتم در حالت ۱بعدی پویا امکان‌پذیر است.

برای محاسبه مستطیل با حداکثر اختلاف دورنگ، هر بار اضلاع بالایی و پایینی مستطیل را ثابت فرض می‌کنیم و کلیه نقاط مابین اضلاع بالایی و پایینی را روی محور x نگاشت می‌کنیم. حال با محاسبه بازه با حداکثر اختلاف دو رنگ می‌توان اضلاع چپ و راست این مستطیل را محاسبه کرد. اگر Y_{coard} شامل مختصه y همه نقاط باشد، بنابراین، $O(|y_{coard}|^2) = O(n^2)$ جفت مقدار مختلف برای اضلاع بالایی و پایینی مستطیل وجود دارد. حال با استفاده از تکنیک خط جاروب که از بالا تا پایین، مختصه y نقاط را ملاقات می‌کند و الگوریتم ۱بعدی پویا می‌توان الگوریتمی کارا برای محاسبه مستطیل با حداکثر اختلاف دورنگ ارائه کرد.

قضیه ۴- فرض کنید مجموعه S دارای n عضو است. همچنین از تابع نگاشت X استفاده کرده‌ایم. بدست آوردن مستطیلی موازی محورهای مختصات که دارای بیشترین اختلاف دو رنگ است در زمان $O(n^2 \log n)$ و حافظه $O(n)$ امکان‌پذیر است [5].الگوریتم مستطیل با حداکثر اختلاف دو رنگ به سادگی می‌تواند برای حالت d بعدی نیز بکار برده شود. مساله پیداکردن ابرمستطیل با حداکثر اختلاف دو رنگ در d بعد به زمان $O(n^{2d-2} \log n)$ و حافظه $O(n)$ نیاز دارد [5].

۴- پیاده‌سازی آزمایش‌های گوناگون و نتایج

در این بخش به معرفی آزمایش‌های گوناگون با استفاده از الگوریتم‌های معرفی شده در بخش‌های قبل و پیاده‌سازی آن‌ها می‌پردازیم.

۴-۱- معرفی مجموعه داده‌های فراهم شده برای پیاده‌سازی

ابتدا مجموعه‌داده‌هایی را که برای ارزیابی آزمایش‌ها از آنها استفاده کرده‌ایم را معرفی می‌کنیم. ما برای پیاده‌سازی الگوریتم از ۹ مجموعه داده استفاده کردیم. مجموعه داده‌ها از پایگاه داده UCI [15] انتخاب گردید. مشخصات مجموعه داده‌های انتخابی در جدول ۱ آورده شده است. در این جدول ستون مقادیر نامشخص[7] نشان می‌دهد آیا از بین کلیه مثال‌های یک مجموعه، مثالی با یک مقدار ویژگی نامعلوم وجود دارد یا خیر. منظور از خط مبنا[8] در این جدول بیشترین درصد مثال‌ها از یک رده است. به عبارتی اگر یک الگوریتم کاملا تصادفی به هریک از داده‌های آموزشی یک برچسب انتساب دهد به دقتی برابر با خط مبنا خواهد رسید. بنابراین، لازم است الگوریتم یادگیر، به دقتی قابل توجه

6. Online learning

7. Missing value
8. Baseline

جدول۱: معرفی مجموعه داده‌ها.

نام	سال	نوع خصایص	تعداد داده	تعداد خصایص	مقادیر نامشخص	خط مبنا
BL	۲۰۰۸	حقیقی	۷۴۸	۴	خیر	۷۶,۲
HA	۱۹۹۹	صحیح	۳۰۶	۳	خیر	۷۳,۵۲
BCW	۱۹۹۲	صحیح	۶۹۹	۱۰	بله	۶۵,۵۲
IO	۱۹۸۹	صحیح/حقیقی	۳۵۱	۳۴	خیر	۶۴,۱
MA	۲۰۰۷	حقیقی	۱۹۰۲۰	۱۰	خیر	۶۴,۸
PI	۱۹۹۰	صحیح/حقیقی	۷۶۸	۸	بله	۶۵,۱
PA	۲۰۰۸	حقیقی	۱۹۵	۲۲	خیر	۷۵,۳۸
CO		حقیقی	۲۰۸	۶۰	خیر	۵۳,۳۶
G2	۱۹۸۷	حقیقی	۱۶۳	۹	خیر	۵۳,۴

نسبت به خط مبنا دست یابد. سایر ستون‌ها نیز سایر خصایص داده‌ها را معین می‌سازد. هنگام انتخاب از میان مجموعه داده‌های حقیقی- مقدار یا صحیح- مقدار تلاش کردیم به تعداد خصایص، تعداد داده‌ها و خط مبنا توجه کنیم بطوریکه تنوع در میان مجموعه‌های انتخابی موجود باشد.

۴-۲- آزمایش‌های انجام شده

ارزیابی الگوریتم‌های پیاده‌سازی شده بر اساس معیار دقت[9] و و پیچیدگی زمانی آن‌ها صورت می‌گیرد. رابطه مربوط به محاسبه دقت به صورت زیر است:

$$Accuracy = \frac{(TP+TN)}{(C^{+}+C^{-})} \qquad (4)$$

در این رابطه، منظور از TP (TN) داده‌های مثبت (منفی) است که توسط فرضیه موجود، مثبت (منفی) رده‌بندی می‌شوند. همچنین C^{+} تعداد کل داده‌های مثبت و C^{-} تعداد کل داده‌های منفی است. در واقع، به کمک رابطه بالا نسبت داده‌هایی که به درستی رده‌بندی شده‌اند به کل داده‌ها محاسبه می‌شود.

آزمایش ۱- پیاده‌سازی الگوریتم C4.5

برای ارزیابی عملکرد الگوریتم ابرمستطیل با حداکثر اختلاف دو رنگ لازم است این الگوریتم با یک الگوریتم پایه مقایسه گردد. به همین دلیل الگوریتم C4.5 را به عنوان معیار مقایسه در نظر می‌گیریم. ابتدا به کمک نرم‌افزار وکا[10] [16] الگوریتم j.48 (الگوریتم C4.5) را روی تک‌تک مجموعه داده‌ها اجرا می‌کنیم. در اجرای الگوریتم از روش تصدیق-متقاطع[11] با ۳ بخش[12] استفاده کردیم. همچنین، در این پیاده‌سازی درخت‌های ایجاد شده درخت‌هایی دودویی است که در آن در هر مسیر از ریشه به برگ ممکن است چندین بار یک خصیصه مشاهده گردد. نتایج این الگوریتم در جدول ۲ در سطر دوم آورده شده است.

اشاره می‌کنیم برای پیاده‌سازی کلیه آزمایش‌هایی که در ادامه آورده می‌شوند از زبان برنامه‌سازی C# و پایگاه داده SQL استفاده کردیم. همچنین، از آنجا که در مجموعه داده‌های خود مجموعه داده‌هایی با مقادیر نامعلوم به ازای برخی خصایص وجود دارد در کلیه آزمایش‌ها با این داده‌ها کاملا بدبینانه رفتار می‌کنیم. یعنی اگر خصیصه انتخاب شده دارای برخی مقادیر نامشخص باشد فرض می‌کنیم داده متناظر با آن، توسط فرضیه بدست آمده اشتباه رده‌بندی می‌شود.

آزمایش ۲- ابرمستطیل با حداکثر اختلاف دو رنگ (MBD[13]-*d)

9. Accuracy

10. Weka
11. Cross- validation
12. Fold
13. Maximum bichromatic discrepancy

در اینجا الگوریتم محاسبه ابرمستطیل dبعدی با حداکثر اختلاف دو رنگ را به گونه‌ای متفاوت اجرا می‌کنیم. مراحل اجرای این آزمایش در آزمایش ۲ آورده شده است.

جدول ۲: نمایش معیار دقت در آزمایش‌های مختلف

	BCW	BL	G2	HA	IO	MA	PI	PA	CO
C4.5	۹۴,۲۷	۷۶,۷۳	۸۰,۳۶	۷۰,۲۶	۹۰,۸۸	۸۵,۰۱	۷۳,۳	۸۴,۱	۷۲,۱۱
MBD-1d	۹۲,۱۵	۷۴,۱۴	۷۳,۷۶	۷۲,۷۳	۸۱,۲۷	۷۳,۵۲	۷۳,۶۲	۸۳,۱۳	۷۰,۷۲
MBD-2d	۹۳,۳۷	۷۴,۹۸	۷۰,۳۶	۷۳,۳۳	۸۷,۰۸	۷۵,۶۴	۶۷,۸۵	۸۲,۹۳	۶۹,۴۱
MBD-3d	۹۳,۶	۷۶,۱۹	۶۶,۷	۷۳,۲۵	۸۶,۸۴	۷۵,۳۲	۶۵,۹۸	۷۸,۹۳	۶۶,۷۴
MBD-4d	۹۴,۰۱	۷۶,۲۳	۶۳,۲۷	-	۸۶,۴۳	۷۴,۷۴	۶۵,۴۷	۷۷,۰۳	۶۳,۴
MBD-5d	۹۳,۹	-	۵۹,۳۸	-	۸۳,۰۳	۷۴,۰۶	۶۵,۵۲	۷۰,۵۶	۶۰,۷۵
MBD-md	۹۱,۹۶	-	۴۹,۴۹	-	۵۹,۲۶	۷۲	۶۵,۲۶	۵۰,۸	۴۷,۰۵
Hi-MBD-1d	۹۱,۵۴	۷۲,۹۲	۷۱,۹	۷۰,۲	۸۰,۶۹	۷۳,۶	۷۲,۵۷	۸۳,۰۳	۶۹,۲

۱- $\frac{1}{3}$ داده‌ها به صورت تصادفی به مجموعه آزمون و $\frac{2}{3}$ داده‌ها به مجموعه آموزشی تخصیص می‌یابد.

۲- یک خصیصه انتخاب می‌شود و بازه با حداکثر اختلاف دو رنگ به ازای آن خصیصه روی مثال‌های آموزشی به همراه دقت رده‌بندی آن، محاسبه می‌شود.

۳- مرحله ۲ را به تعداد خصایص تکرار می‌کنیم. بهترین خصیصه را از نظر دقت رده‌بندی بدست می‌آوریم. کلیه داده‌های خارج از بازه متناظر با بهترین خصیصه را از مجموعه داده‌ها حذف می‌کنیم. همچنین خصیصه انتخابی از مجموعه خصایص حذف می‌شود.

۴- اگر $d \succ 1$ باشد به تعداد d بار هریک از مراحل ۲ و ۳ روی خصایص باقیمانده و همچنین داده‌های باقیمانده تکرار می‌کنیم. به این ترتیب بازه با حداکثر اختلاف دو رنگ به ازای d خصیصه محاسبه می‌گردد. این d بازه را به عنوان تقریبی از ابرمستطیل با حداکثر اختلاف دو رنگ می‌شناسیم. ابر مستطیل بدست آمده، فرضیه مورد نظر ما است. دقت این فرضیه بر روی داده‌های آزمون مورد ارزیابی قرار می‌گیرد. برای پیشگویی رده داده آزمون با استفاده از فرضیه محاسبه شده، کلیه نقاط خارج از ابرمستطیل، منفی و کلیه نقاط داخل آن مثبت رده‌بندی می‌شوند.

۵- هریک از مراحل ۱ تا ۴ را ۱۰۰ بار تکرار کرده و نتایج بدست آمده را میانگین‌گیری می‌کنیم.

آزمایش ۲: محاسبه ابرمستطیل با حداکثر اختلاف دو رنگ (MBD-*d)

چنانچه در آزمایش ۲ گفته شد، ما به جای محاسبه دقیق ابرمستطیل با حداکثر اختلاف دو رنگ و با زمان اجرای $O(n^{2d-2}\log n)$، d بار الگوریتم بازه با حداکثر اختلاف دو رنگ را اجرا می‌کنیم. در هر بار، برای انتخاب بهترین خصیصه الگوریتم بازه با حداکثر اختلاف دورنگ را به تعداد خصایص تکرار می‌کنیم. به این ترتیب اجرای آزمایش ۲ در زمان $O(dmn\log n)$ امکان‌پذیر است. با این تقریب به جای الگوریتم ابرمستطیل با حداکثر اختلاف دو رنگ، می‌توان به پیاده‌سازی سرراست و با پیچیدگی محاسباتی کمتری نسبت به پیاده‌سازی الگوریتم اصلی دست یافت.

نتایج بدست آمده با این آزمایش در جدول ۲ آورده شده است. در پیاده‌سازی آزمایش MBD-*d، علامت * محاسبه ابرمستطیل d بعدی را نشان می‌دهد. به عنوان مثال، MBD-3d ابرمستطیل سه بعدی و MBD-md ابرمستطیل m بعدی را نشان می‌دهد، که در آن m تعداد خصایص مجموعه داده متناظر می‌باشد. به عنوان مثال مجموعه داده HA دارای سه خصیصه است. بنابراین، مقادیر MBD-4d و MBD-5d برای آن‌ها بی‌معنا است و در جدول به جای آن‌ها خط تیره گذاشته شده است. توجه به مقادیر به دست آمده برای مجموعه داده‌های گوناگون، نشان می‌دهد که در اکثر موارد با افزایش تعداد ابعاد پیاده‌سازی، دقت فرضیه محاسبه شده کاهش می‌یابد.

آزمایش ۳- بازه با حداکثر اختلاف دو رنگ سلسله مراتبی (Hi-MBD-1d)

با اجرای آزمایش ۲ دیدیم که هرقدر تعداد ابعاد افزایش می‌یابد دقت فرضیه محاسبه شده کاهش می‌یابد. بنابراین، نتیجه گرفتیم یک خصیصه تکی نسبت به مجموعه‌ای از خصایص نماینده بهتری از کل داده‌ها برای رده‌بندی است. به همین دلیل تلاش کردیم الگوریتم یک بعدی را مقداری بهبود بخشیم. به همین دلیل از روش بازه با حداکثر اختلاف دو رنگ سلسله مراتبی استفاده کردیم. درواقع پس از محاسبه بازه با بیشترین تعداد نقاط آبی و کمترین نقاط قرمز، کلیهٔ کلیه نقاط خارج از بازه را از مجموعه نقاط حذف می‌کنیم. سپس از مجموعه نقاط باقیمانده، بازه با بیشترین نقاط قرمز و کمترین نقاط آبی را محاسبه می‌کنیم. در واقع محور مورد نظر به ۵ قسمت تقسیم می‌شود. این پنج بازه را به عنوان فرضیه دلخواه درنظر می‌گیریم. به ترتیب داده‌های موجود در هریک از بازه‌ها، برچسب قرمز، آبی، قرمز، آبی و قرمز خواهند گرفت. نتایج اجرای این الگوریتم نیز در جدول ۲ آورده شده است.

۴-۳- تحلیل نتایج

با توجه به داده‌های جدول ۱، میانگین خط مبنا برای ۹ مجموعه داده استفاده شده برابر با ۶۵,۷ درصد است. میانگین دقت الگوریتم C4.5 برابر با ۸۰,۷۸ درصد، الگوریتم MBD-1d دارای میانگین دقت ۷۷,۲۲ درصد و MBD-2d میانگین دقت ۷۷,۲۱ درصد است. خواهیم دید با افزایش d، میانگین دقت مرتب کاهش می‌یابد بطوریکه میانگین دقت الگوریتم MBD-md به ۶۵,۰۳ درصد می‌رسد که در آن m تعداد ابعاد داده‌ها را مشخص می‌سازد.

میانگین دقت الگوریتم Hi-MBD-1d نیز برابر ۷۶,۱۸ است. بنابراین، پس از آزمایش یک (الگوریتم C4.5)، بهترین آزمایش انجام شده، آزمایش ۲ با $d=1$ است. یعنی از بین پیاده‌سازی‌های موجود برای محاسبه ابرمستطیل با حداکثر اختلاف دو رنگ، بازه با حداکثر اختلاف دو رنگ بیشترین دقت را داراست.

دقت این آزمایش ۱۱,۵ درصد از میانگین خط مبنا بالاتر و ۳,۵۶ درصد از آزمایش ۱ یعنی الگوریتم شناخته‌شده C4.5 کمتر است.

چنانچه به نتایج جدول ۲ توجه کنیم، می‌بینیم کارایی الگوریتم مطرح شده به ازای سه مجموعه داده IO، G2 و MA ضعیف است.

اگر این سه مجموعه داده را از مجموعه داده‌های خود حذف کنیم، می‌بینیم دقت آزمایش ۲ به اندازه ۰,۷ درصد از دقت الگوریتم C4.5 کمتر خواهد شد.

از طرفی می‌دانیم در پیاده‌سازی مستطیل با حداکثر اختلاف

دورنگ، ابتدا یک خصیصه انتخاب می‌کنیم و بهترین بازه متناظر با آن را بدست می‌آوریم. سپس، کلیه نقاط خارج از بازه را حذف می‌کنیم.

به ازای نقاط داخل بازه همین‌کار را به ازای ابعاد دیگر تکرار می‌کنیم. این در حالی است که در الگوریتم C4.5 نقاط موجود در هیچ یک از نواحی دور ریخته نمی‌شود و تقسیم‌بندی به ازای کلیه نواحی تا رسیدن به بیشترین خلوص ادامه می‌یابد. بنابراین، تا حدی دقت کمتر الگوریتم مطرح شده نسبت به الگوریتم C4.5 قابل توجیه است. به عبارتی در الگوریتم C4.5، کلیه ندهای درخت تا رسیدن به بیشترین خلوص بسط می‌یابند.

در صورتی که در الگوریتم بازه با حداکثر اختلاف دورنگ تنها یک مسیر از ریشه به برگ بسط می‌یابد. واضح است که بسط تنها یک مسیر از درخت از ریشه تا برگ در زمان کمتری انجام می‌شود.

حال چنانچه $d=1$ انتخاب شود در واقع تنها با یک درخت تصمیم یک سطحی و با سه انشعاب روبرو هستیم که از نظر پیچیدگی محاسباتی بسیار کاراتر از الگوریتم درخت تصمیم C4.5 خواهد بود. این در حالی است که دقت الگوریتم جاری تقریبا مشابه الگوریتم C4.5 است. بنابراین، الگوریتم ارائه شده در این مقاله الگوریتمی قابل رقابت با الگوریتم C4.5 معرفی می‌شود.

۵- نتیجه‌گیری و مسایل باز

در این مقاله الگوریتم محاسبه ابرمستطیل با حداکثر اختلاف دو رنگ که یکی از مسایل شناخته شده در هندسه محاسباتی است را معرفی کردیم. پس از بیان تئوری کافی در این زمینه، آزمایش‌های متعدد با استفاده از این الگوریتم روی ۹ مجموعه داده انجام شد.

سپس، کارایی این الگوریتم‌ها را با الگوریتم شناخته‌شده C4.5 مقایسه کردیم. چنانچه دیدیم کارایی الگوریتم بازه با حداکثر اختلاف دو رنگ به اندازه ۳,۵۶ درصد از الگوریتم C4.5 کمتر است. از طرفی محاسبات انجام شده برای بدست آوردن ابرمستطیل با حداکثر اختلاف دو رنگ در مقایسه با الگوریتم C4.5 کمتر خواهد بود. بنابراین، می‌توان این الگوریتم را به عنوان یک الگوریتم قابل رقابت با الگوریتم C4.5 معرفی کرد. تحلیل بیشتر هریک از مجموعه داده‌ها و بررسی عدم کارایی این الگوریتم روی برخی از مجموعه داده‌ها، همچنین بهبود عملکرد این الگوریتم به عنوان یک مساله باز پیشنهاد می‌گردد.

به عنوان یکی از کارهایی که می‌توان برای بهبود عملکرد این الگوریتم پیشنهاد کرد آن است که ابتدا بر روی داده‌ها PCA اعمال گردد. به این ترتیب می‌توان محورهایی که داده‌ها دارای بیشترین پراش هستند را شناسایی کرد. حال اگر ابتدا داده‌ها را روی این محور نگاشت کنیم و سپس بازه با حداکثر اختلاف دو رنگ را محاسبه کنیم به کارایی بیشتری نسبت به الگوریتم مطرح شده دست خواهیم یافت. همچنین می‌توان این روش را به عنوان یک روش برای عمل گسسته‌سازی روی محورهای پیوسته- مقدار قبل از اعمال الگوریتم درخت‌های تصمیم معرفی کرد. چرا که چنانچه می‌دانیم یکی از مهمترین چالش‌های درخت تصمیم در برخورد با داده‌های پیوسته- مقدار، گسسته‌سازی آن‌ها با تقسیم محورهای مختصات به چندین بازه است. تصور می‌شود اگر این روش اعمال شده و سپس الگوریتم C4.5 را بر روی بازه‌های بدست آمده اجرا کنیم کارایی الگوریتم C4.5 به مقدار قابل توجهی افزایش یابد. پیاده‌سازی هریک از ایده‌های مطرح شده به عنوان کارهای آینده پیشنهاد می‌گردد.

منابع

1.T. M. Mitchell, *Machine learning*, McGraw-Hill, 1997.

2.P.L. Hammer, A. Kogan, B. Simeone, and S. Szedmak, "Pareto-optimal patterns in logical analysis of data," *Discrete Applied Mathematics*, vol.144, pp.79-102, 2004.

3.M. Kreveld, T. Lankveld, and R. Veltkamp, "Identifying well-covered minimal bounding rectangles in 2D point data," in *25th European Workshop on Computational Geometry*, EWCG, 2009, pp.277-280.

4.J. Eckstein, P. Hammer, Y. Liu, M. Nediak, and B. Simeone, "The maximum box problem and its application to data analysis," *Journal of Computational Optimization and Application*, vol.23, pp. 85-98, 2002.

5.D. P. Dobkin, D. Gunopulos, and W. Maass, "Computing the maximum bichromatic discrepancy with applications to computer graphics and machine learning," *Journal of Computer and System Science*, vol.52, pp. 453–470, 1996.

6.Y. Liu, and M. Nediak, "Planar case of the maximum box and related problems," in *Proceedings 15th Canadian Conferance of Computational Geometry*, CCCG, 2003, pp.14-18.

7.C. Cortés, J. Díaz-Báòez, P. Pérez-Lantero, C. Seara, J. Urrutia, and I. Ventura, "Bichromatic separability with two boxes: A general approach," *Journal of Algorithms*, vol.64, pp.79-88. 2009.

8.C. Cortés, J. Díaz-Báòez, and J. Urrutia, "Finding enclosing boxes with empty intersection," in *Proceedings 23rd European Workshop on Computational Geometry*, EWCG, 2006, pp.185-188.

۹.ز. مصلحی، " تفکیک‌پذیری نقاط با اشیای هندسی در فضای دوبعدی، "پایان نامه کارشناسی ارشد دانشکده مهندسی کامپیوتر و فناوری اطلاعات، دانشگاه صنعتی امیرکبیر، ۱۳۹۱.

۱۰.ز. مصلحی، ع. باقری، " تفکیک پذیری سری نقاط دو رنگ با دو مستطیل مجزا و موازی محورهای مختصات، "*مجله علمی پژوهشی رایانش نرم و فناوری اطلاعات*، جلد ۱، شماره ۲، صفحه ۳۵-۴۲، ۱۳۹۱.

11.S. Cabello, J. M. Díaz-Báñez, C. Seara, J. A. Sellarès, J. Urrutia, and I. Ventura, "Covering point sets with two convex objects," in *21st European Workshop on Computational Geometry*, EWCG, 2005, pp. 195-206.

12.S. Cabello, J. M. Díaz-Báñez, C. Seara, J. Urrutia, and I. Ventura, "Covering point sets with two disjoint disks or squares," *Computational Geometry: Theory and Application*, vol.40, pp. 195-206, 2008.

13.D.P. Dobkin, and D. Gunopulos, "Geometric problems in machine learning, in *Lecture Notes in Computer Science*, *LNCS*, 1996, vol.1148, pp.121-132.

14.M. D. Berg, O. Cheong, M. V. Kreveld, and M. Overmars, *Computational geometry: algorithms and applications*, 3rd Edition, TELOS, Santa Clara, CA, USA, 2008.

15.K. Bache and M. Lichman. (2013). *UCI Machine Learning Repository* [Online]. Available: http://archive.ics.uci.edu/ml

16.I. H. Witten, E. Frank, and M. A. Hall, *Data Mining: Practical machine learning tools and techniques*, 3rd Edition, Morgan Kaufmann, San Francisco, 2011.

ضمیمه الف

در اینجا نام کامل هریک از مجموعه داده‌های بکار گرفته شده آورده شده است. در آخر نیز اشاره می‌کنیم برای دسترسی به هریک از مجموعه داده‌ها کافی است هریک از نام‌های گفته شده را در مرجع [15] جستجو کنید.

BL: Blood transfusion service center
HA: Haberman's survival
BCW: Breast cancer wisconsin (original)
IO: Ionosphere
MA: Magic gamma telescope
PI: Pima indians diabets
PA: parkinson
CO: Connectionist bench
G2: Glass identification