نقاط ضعف و قوت ماشین SVM

بخاطر کرنل غیرخطی هم داده جدا پذیرخطی و جداپذیرغیر غیرخطی رو خوب دسته بندی میکنه. قوی، عملکرد خوب - برای کار بر روی دادگان کوچکتر مناسب است - برای کرنل خطی بلک باکس نیست - در مصرف حافظه بهینه عمل می کند - برای داده های اسپارس مناسب ترین است - چالش در اینجا تنظیم پارامتر هزینه **Cost** است - بالا بردن **cost** خطر **overfit** ایجاد می کند - پایین آوردن **cost** خطر **underfit** ایجاد می کند - برای کرنل غیر خطی خطی بلک باکس است - اگر مجموعه داده نویزی و یا بزرگ باشد عملکرد خوبی ندارد. - بطور مستقیم تخمین تابع احتمالاتی ندارد و تخمین با استفاده **CV** انجام می شود. داده زیاد پیچیده باشه **overfit** میشه.

تابع هدف در SVM و راهکار جریمه کردن و دوری از overfit

۲ جز دارد: ۱. پهنای مارجین(حاشیه) ۲ جریمه کردن داده هایی که خارج از کلاسشون قرار گرفتند و یا خارج از مرز هستند (یعنی اومد داخل مارجین). برای جریمه کردن داده ها باید بتونن برچسب غلط بگیرن و جریمه بشن و در تابع هدف اضافه بشن. همانطور که مارجین رو زیاد می کنیم جریمه رو سعی میک کنیم کم کنیم. هرقدر **C** بزرگتر داده ها با خطای کمتری دسته بندی میشن، هرچقدر **C** کوچکتر **SVM** پهنای حاشیه رو بزرگتر نگه میداره.

مزایای درخت تصمیم فهم ساده، هزینه ساخت ارزان، بشدت سریع در طبقه بندی رکوردهای نامعلوم،توانایی کار با داده های بزرگ و پیچیده، قابلیت ترکیب با سایر دسته بندها،ساده سازی تفسیر برای درخت های کوچک،استفاده مجددآسان، توانایی کار با داده های پیوسته و گسسته، عدم نیاز به تخمین تابع توزیع، سازگاری با داده های **Null**، یک مثل جعبه سفید، مناسب برای جداپذیرخطی وغیرخطی(با کاهش بُعد)

معایب درخت تصمیم مصرف زیاد حافظه، هزینه محاسباتی زیاد،بازنمایی دشوار، بزرگ شدن بصورت نمایی با بزرگ شدن مساله، احتمال تولید روابط نادرست، احتمال خطای بالا با تعداد نمونه آموزشی کم و دسته های زیاد، احتمال **Overfit** بالا

مواردکاربرد درخت تصمیم: داده جداپذیرخطی باشد.

پیش بینی لینک در شبکه، تشخیص بیماری +-، بهینه سازی مشارکت اوراق بهادار،تشخیص تقلب، پولشویی،اهدای وام

درخت تصمیم وقتی بجایی رسیدیم که داده هاش یه ویژگی دارن تقسیم متوقف میشه، برای جلوگیری از **overfit** حدآستانه قرارمیدیم. ترجیحا تعداد تقسیماتش کم باشه و عمق درخت از یه حدی بیشتر نشه. مهم است که کدام یک از ویژگی‌ها را در سطوح بالاتری از درخت انتخاب کنیم تا به طبقه‌بندی کمک کند.

Gain و Entropy، بهتر است شاخص جینی کمتر باشد. **Entropy** در واقع نشان دهنده کم بودن اطلاعات است. یعنی در مجموعه‌ی داده‌ی شما، از روی یک ویژگی(بُعد) چقدر می‌توانید تشخیص دهید که کلاسِ نهایی چیست.اگر ویژگی دارای مجموع **Entropy** بالا است و در نتیجه اطلاعات کمتری دارد. اگر ویژگی دارای مجموع **Entropy** پایین است و در نتیجه اطلاعات بیشتری دارد و بهتر است.**Gain** که در واقع همان **Information Gain** می باشد، از **Entropy** هر مقدار از ویژگی‌ها کمک گرفته و به میزانِ اطلاعاتی که می‌توان از یک ویژگی(بُعد) به دست آورد، گفته می شود.

الگوریتم CART: Classification And Regression Tree

بر اساس درخت های دودویی(باینری) بنا نهاده شده، داده‌ها را به قسمت‌های دوتایی تقسیم کرده و بر اساس آن‌ها درخت دودویی(باینری) را می‌سازد. از معیاری به نام معیار شاخص **Gini** استفاده می کند. برای هر ویژگی(بُعد) هر چقدر شاخص **Gini** کمتر باشد، یعنی آن ویژگی اطلاعات بیشتری را به ما می دهد. جلوگیری از **overfit** شدن درخت: شرط توقف(حدآستانه)

طبقه بند ترکیبی Ensemble

فایده اصلی: کاهش نرخ خطا، شرط استقلال مدلها مانع از همبسته شدن خطای مدلها خواهد شد. در نهایت برای تشخیص دسته یا جایگاه نمونه آزمایشی، خروجی همه مدلها با یکدیگر تجمیع میشوند. نکات مهم: چگونگی ایجاد دسته بندهای پایه، چگونگی ادغام خروجی های یادگیرنده های پایه، موفقیت سیستم **ensemble**، تکیه داشتن آن بر تنوع طبقه بندی کننده هایی که آن را تشکیل می دهند، می باشد. اگر هر طبقه بندی کننده خطای مختلفی ارائه دهد، پس از ترکیب استراتژیک آن ها می توانید کل خطا را کاهش دهد.

الگوریتم Kmeans جزو الگوریتم های افرازی هستش.تلاش می‌شود تا مراکز دسته‌ای یافت شوند که نماینده ناحیه خاصی از داده‌ها هستند. هر نقطه داده به نزدیک‌ترین مرکز خوشه نسبت به خودش، تخصیص داده می‌شود. سپس، مرکز خوشه‌ها بر اساس میانگین نقاط داده‌ای که به آن خوشه تخصیص داده شده‌اند مجددا محاسبه و تعیین می‌شوند. شرط اتمام: همگرایی خوشه ها

نقاط قوت و ضعف Kmeansکارایی، مناسب برای داده های حجیم. فقط با مقادیر عددی کار می کند چون باید **avg** بگیرد. داده حتما باید نرمال شده باشد. **K** را حتما باید از قبل تعیین کرد. به داده نویز و پرت حساس است. در فضای ۲بُعدی خوشه بصورت دایره نمی تواند تولید کند و در ۳ بُعدی هم کروی. پیشنهاد: ابتدا داده ها مصوربشن ببینیم خوشه ها محدب است یا خیر!

الگوریتم K-medoidsگونه ای **K-means** که در برابر داده نویز و پرت مقاوم تر هست. مرکزی ترین عنصر خوشه به عنوان مرکز ثقل خوشه هستش. **Medoid** هر خوشه داده ای هست که مجموع فاصله اش با داده ها از هر داده دیگه ای کمتر باشه.

خوشه بندی سلسله مراتبی

خروجی: دندوگرام، ۲نوع: تجمعی، تجزیه ای، تجمعی سریعتر و پراستفاده تر است. (مشهور در تجمعی: الگوریتم **BRICH**)

محاسبه فاصله درون خوشه ای و بین خوشه ای

Single link: کوچکترین فاصله بین یک عنصر در یک خوشه و خوشه دیگر، **Complete link**: بزرگترین فاصله ...، نکته مهم: **single link** تمایل دارد تک خوشه تشکیل دهد. **Average**: متوسط فاصله ...، **centroid**: فاصله بین مرکز ۲ خوشه، **medoid**: فاصله بین **medoid** ۲ خوشه.

الگوریتم Knnیادگیرنده تنبل(منتظر داده ورودی میمونه تا طبقه بندی کنه). ساده است، از تابع اقلیدسی واسه فاصله استفاده می کنه.

یک دسته بند **ensemble** با ۱۰ درخت تصمیم و مکانیزم رای گیری اکثریت عملکرد خوب نیست! پیشنهاد؟ افزایش تعداد دسته بندها، تغییرنوع دسته بندهای پایه، تبدیل رای گیری از اکثریت **bagging** به وزن دار **boosting**، تغییر و تنظیم پارامتر دسته بندهای پایه مثلا کاهش عمق درخت برای وقتی که بیش برازش می شود

برای تشخیص و درمان بیماری کشنده کدام یک از معیارهای دسته بندها مناسب تر است؟

Accuracy خیلی اینجا مناسب نیست چون تمایزی قائل نمی شود، **recall** اینجا خیلی مفید است (اگر هیچ داده ای از کلاس + نره تو منفی میشه ۱۰۰٪)، **precession** و **recall** سعی می کنند مرزها قاطی نشوند، **f-major** میانگین هر۲است، زمانی بالاست که **precession** و **recall** بالا باشند. **Precession** (افرادی که سکته نکردن رو اشتباها نگیم سکته کردن)، **fmajor** (سکته ای و غیرسکته ای رو درست تشخیص بدیم).

Boosting و Bagging

در این روش مجموعه داده اصلی با استفاده از روش نمونه برداری با جایگذاری به تعدادی مجموعه داده تقسیم بندی می شود . در این ایده چون از روش نمونه برداری با جایگذاری برای نمونه برداری استفاده می شود در نتیجه برای مجموعه داده های با تعداد رکوردهای کم نیز مناسب است در نهایت بر اساس هر کدام از نمونه ها دسته بند ساخته می شود . این روش از یک الگوریتم تکرار شونده استفاده می کند تا به طور تطبیقی توزیع نمونه های آموزشی را تغییر دهد و در فرآیند یادگیری بیشتر بر روی رکوردهایی که در مراحل قبلی به اشتباه دسته بندی شده اند تمرکز دارد .در این ایده در انتهای هر مرحله ممکن است وزن نمونه ها تغییر کند به این صورت که وزن رکورد هایی که به اشتباه دسته بندی شده اند افزایش یافته و وزن رکوردهایی که به درستی دسته بندی شده اند کاهش می یابد

روش های ارزیابی دسته بندی

دقت، زمان برای ساخت و استفاده از مدل، پایداری، قابلیت تفسیر، جمع و جور بودن، توانایی مواجهه با داده نویزی و مفقوده

نقاط ضعف خوشه بندی انتخاب اندازه دقیق فواصل و وزن ها آسان نیست. به پارامترهای اولیه: **k**، حداقل نزدیکی، خوشه های اولیه حساس است. تفسیر نتایج نیازمند خبره است، ذات بدون ناظربودن الگوریتم ها، مشکل بودن تعریف تابع هدف.

نقاط ضعف خوشه بندی

انتخاب اندازه دقیق فواصل و وزن ها آسان نیست. به پارامترهای اولیه: **k**، حداقل نزدیکی، خوشه های اولیه حساس است. تفسیر نتایج نیازمند خبره است، ذات بدون ناظربودن الگوریتم ها، مشکل بودن تعریف تابع هدف، یک مساله سخت است.

خوشه بندی افرازی

مجموعه داده را به**k** افراز که هر افراز نماینده یک خوشه میباشد تقسیم میکنند که این افرازبندی بر حسب یک تابع هدف صورت میپذیرد. کمینه سازی مجموع مربعات خطای فاصله هر نقطه تا مرکز خوشه، نمونه ای از تابع هدف بکاررفته در روشهای افرازی میباشد. در اینگونه روشها هر خوشه باید حداقل شامل یک داده باشد و هر داده هم فقط باید به یک خوشه تعلق داشته باشد. از معایب اینگونه روشها میتوان به کارایی ضعیف آن در خوشه های همپوشان اشاره کرد.

اگر ساختارداده غیرخطی باشد خوشه بندی کلاسیک با شکست مواجه می شود، راهکارپیشنهادی چیست؟

در این حالت خوشه بندی طیفی روشی قدرتمند برای دسته بندی داده ها محسوب می شود. این تکنیک با تبدیل ورودی، فضای جدیدی با قابلیت توصیف مناسب تر از داده ها را در اختیار ما قرار می دهد **(Spectral Clustering)**

درست+	بیمار دیابت دارد و درست پیش بینی شده
نادرست+	بیماردیابت ندارد اما پیش بینی ما غلطا میگه داره
نادرست -	بیماردیابت دارد اما پیش بینی ما غلطا میگه نداره
درست -	بیمار دیابت ندارد و درست پیش بینی شده

ایده آل: نادرست – و + ۰ باشد.

دقت: **TP/TN/N**، بازخوانی:**TP/TP+FN**، صحت: **TP/TP+FP**

معیارهای بازخوانی و صحت به جای معیار اولیه دقت، کاربرد وسیع تری در دنیای امروز یادگیری ماشین پیدا کرده است. در اغلب موارد، این دو معیار با هم رشد و حرکت نمی کنند. گر بتوانیم معیاری ترکیبی از این دو معیار برای سنجش الگوریتم های دسته‌بندی به دست آوریم، تمرکز بر آن معیار به جای بررسی همزمان این دو، مناسب‌تر خواهد بود مثلا از میانگین این دو به عنوان یک معیار جدید استفاده کنیم و سعی در بالا بردن میانگین حسابی این دو داشته باشیم.

$$\text{F1-Score} = \frac{2}{\frac{1}{Recall}+\frac{1}{Precision}} = 2 \times \frac{Precision*Recall}{Precision+Recall}$$

در تشخیص ایدز یا تشخیص کلاه برداری در تراکنش های بانکی، ما نیاز به شناسایی تمامی موارد ایدز و کلاه‌برداری داریم یعنی نیاز داریم که بازخوانی ما بسیار بالا باشد و اگر خطایی هم تولید شد مثلاً بیماری به اشتباه ایدزی تشخیص داده شد و یا یک تراکنش سالم، متهم به کلاه برداری شد، کافی است با کمی آزمایش بیشتر، نتایج را بهبود خواهیم بخشید و موارد خطا را از لیست تشخیص داده شده‌ها حذف خواهیم کرد.

در مواردی که دسته‌ها، متعادل هستند، مثلاً تعیین جنسیت ارسال کننده یک توئیت، می‌توانیم همان معیار دقت را به کار ببریم ولی وقتی دسته ها متعادل نیستند معیاردقت مناسب نیست.

معیار بازخوانی یا همان **Sensitivity** (حساسیت) نشان می‌دهد چقدر از بیماران واقعی (دسته مثبت) را نسبت به کل جامعه بیماران، شناسایی کرده‌ایم. یعنی نسبت آنهایی که درست شناسایی شده‌اند به مجموع تمام بیماران (آنهایی که به درستی بیمار شناخته شده اند + آنهایی که اشتباهاً سالم تشخیص داده شده‌اند). هدف ما این است که حساسیت مدل ما بالا باشد یعنی تعداد بیشتری از بیماران را شناسایی کند.

معیار **Specificity** همین مفهوم را برای افراد سالم (یا دسته منفی) نشان می‌دهد یعنی چند نفر از افراد واقعا سالم را از کل افراد سالم، درست تشخیص داده‌ایم **TN/TN+FP** میزان افرادی که بیمار نیستند (درست منفی – **TN**) به کل افراد سالم (آنهایی که سالم تشخیص داده شده‌اند و آنهایی که اشتباهاً بیمار فرض شده‌اند)، **Specificity** مدل را تشکیل می‌دهد

هدف اصلی ما در یک مدل دسته بندی چیست؟

افزایش **FScore**

دقت **Accuracy**: نسبت تعداد کل پیش‌بینی‌هایی است که توسط دسته‌بند به درستی برچسب خورده است.

حساسیت «**Sensitivity**» یا «**Recall**»: نسبت موارد مثبت که به درستی شناسایی شدند.

وضوح **Specificity**: نسبت موارد منفی واقعی که به درستی شناسایی می شوند.

دقت **Precision**: نسبت رکوردهایی که مثبت برچسب‌گذاری می‌شوند و واقعا کلاس آن‌ها مثبت است.

اگربجای ماتریس داده فقط ماتریس شباهت زوجی را داشته باشیم و بخواهیم داده ها را به ۳ گروه خوشه بندی کنیم ولی ویژگی ها را نداریم چطور خوشه بندی کنیم ؟

ابتدا شباهت و فاصله را با ماتریس زوجی در میآوریم (با کمک روش سلسله مراتبی که ورودی اش ماتریس فاصله است). ما ۳ گروه غیر همپوشان می خواهیم اما چطور سلسله مراتبی رو به خوشه بندی افزاری تبدیل کنیم؟ یک دندوگرام رسم می کنیم، یک خط می کشیدیم، تعداد مولفه های باقیمانده ۳ تا میشد و تعداد مولفه های باقیمانده ۳ تا می شد. راه دوم: استفاده از **DBScan**

الگوریتم DBScan؟ بدون نظارت برای خوشه بندی، می تواند با ماتریس فاصله و روش خوشه بندی افرازی کارکند اما تعداد خوشه ها را نمی تواند تعیین کند، تعدادی از نقاط هم بدون خوشه به عنوان نقطه مرزی باقی می مانند. نیازی به این نیست که تعداد خوشه‌ها از ابتدا تعیین شود. می‌تواند خوشه‌های دارای اشکال پیچیده را کشف کند. نقاط دورافتاده را می تواند شناسایی کند. با شناسایی نقاطی که در نواحی شلوغ (چگال) از «فضای ویژگی» (**Feature Space**) قرار دارند کار می‌کند. منظور از نواحی چگال، قسمت‌هایی است که نقاط داده بسیار به یکدیگر نزدیک هستند. دو پارامتر **min_samples** و **eps** در الگوریتم **DBSCAN** وجود دارد. هر نقطه داده، از دیگر نقاط داده فاصله‌ای دارد. هر نقطه‌ای که فاصله‌اش با یک نقطه مفروض کمتر از **eps** باشد، به عنوان همسایه آن نقطه در نظر گرفت می‌شود. هر نقطه داده مفروضی که **min_samples** همسایه داشته باشد، یک نقطه «مرکزی» (**Core**) محسوب می‌شود. «نمونه‌های مرکزی» (**core samples**) که نسبت به یکدیگر نزدیک‌تر از فاصله **eps** هستند، در خوشه مشابهی قرار می‌گیرند. مزایا: سریع برای داده‌های با بعد کم یافتن خوشه‌ها برای اشکال نا منظم و کروی تشخیص نقاط نویز، معایب: نقاط مرزی که می‌توانند در دو خوشه نیز باشند، ممکن است به هریک از خوشه‌ها تعلق گیرند.

تفاوت KMeans و DBScan الگوریتم **DBSCAN** نیاز به تعیین تعدادِ خوشه توسط کاربر ندارد و خودِ الگوریتم می‌تواند خوشه‌ها را مبتنی بر غلظتِ آن‌ها شناسایی کند. گروه بندی بر اساس تراکم و غلظت. **DBSCAN** علاوه بر پیدا کردنِ خوشه‌ها، می‌تواند داده‌هایی را که در هیچ خوشه‌ای قرار نمی‌گیرند نیز کشف کند.

شاخص ارزیابی دیویس-بولدین (Davies-Bouldin)

$$R_{ij} = \frac{S_i + S_j}{d_{ij}}, i \neq j$$

مجموع فاصله درون خوشه ای خوشه **i** و خوشه **j** / فاصله بین خوشه ای خوشه **i** و خوشه **j** پس **Rij** بهتره کم باشه (یعنی فاصله درون خوشه ای کم باشه بهتره)

بهمین دلیل میاد میگه **Ri = max Rij** (یعنی برای هر خوشه **i**ام با همه خوشه های دیگه تک تک **Rij**ش رو حساب کن، بدترین خوشه رو پیدا کن نسبت به خوشه **i**، اونی که **Rij** ماکزیمم هستش اونو بزار توی **Ri**، پس برای هر خوشه باید ببینیم با چه خوشه ای بدترین وضعیت رو داره یعنی بیشترین شباهت و بین خود دو خوشه کمترین شباهت، اونو میایم ملاک قرار میدیم و متوسط اینا میشه معیار **.Davies & Bouldin Index**

پس هدف **Davies & Bouldin Index** کم کردن فاصله درون خوشه ای و بیشتر کردن فاصله بین خوشه ای هستش، منتهی سخت گیرانه است و میاد بدترین رو جریمه می کنه، بدترین ها رو شرکت میده.

متد Silhouette

این معیار هم به پیوستگی (**Cohesion**) درون خوشه‌ها و هم به میزان تفکیک‌پذیری آن‌ها بستگی دارد. مقدار نیم‌رخ برای هر نقطه، میزان تعلق آن را به خوشه‌اش در مقایسه با خوشه مجاور اندازه می‌گیرد. فرض کنید نقطه‌ای مانند **x i** در میان داده‌هایی که خوشه‌بندی کرده‌اید وجود دارد و در طی مراحل خوشه‌بندی نیز **k** خوشه (**C k ۲, C ۱C , ... , C**) ایجاد شده است. برای محاسبه معیار نیم‌رخ احتیاج به آشنایی با دو مفهوم اصلی داریم:

میانگین فاصله یک نقطه از خوشه با نقاط دیگر آن خوشه: این مقدار را با **a (i)** نشان داده و به صورت زیر محاسبه می‌کنیم.

$$a(i) = \frac{1}{n_i} \sum_{l=1}^{n_i} d(x_i, x_l)$$

این معیار را می‌توان ملاکی برای ارزیابی تعلق نقطه **x i** در خوشه‌اش در نظر گرفت. هر چه مقدار **a (i)** کوچکتر باشد، میزان تعلق این نقطه به خوشه‌اش بیشتر است. نکته: این معیار می‌تواند براساس بیشتر توابع فاصله، مانند فاصله اقلیدسی و منهتن نیز محاسبه شود.

حداقل میانگین فاصله نقطه با خوشه‌های دیگر: فرض کنید نقطه **x i** به خوشه **C j** تعلق دارد. حال میانگین فاصله این نقطه را با نقاط خوشه‌های دیگر (مثلا **C k**) اندازه می‌گیریم. خوشه‌ای که دارای کمترین میانگین فاصله برای نقطه **x i** باشد، به عنوان خوشه مجاور با این نقطه نامیده می‌شود. مقدار میانگین فاصله نقطه **x i** با نقاط خوشه مجاور را با **b (i)** نشان می‌دهیم.

$$b(i) = \min_{1 \le l \le k} \frac{1}{n_l} \sum_{y_m \in C_l} (d(x_i, y_m))$$

به این ترتیب میزان معیار نیم‌رخ برای نقطه **x i** بوسیله رابطه زیر اندازه‌گیری می‌شود:

$$s(i) = \frac{b(i) - a(i)}{max(b(i), a(i))}$$

در نتیجه اگر **a (i)** کوچکتر از **b (i)** باشد، مقدار شاخص نیم‌رخ مثبت می‌شود و برعکس اگر **b (i)** کوچکتر از **a (i)** باشد، مقدار شاخص نیم‌رخ منفی شده و نشانگر خوشه‌بندی ضعیف است زیرا نقطه **x i** بیش از آنکه شبیه خوشه خودش باشد به خوشه مجاور شباهت دارد. با توجه به رابطه بالا مقدار این شاخص بین ۱- تا ۱+ تغییر می‌کند. مقدار نزدیک به ۱ بیانگر انطباق خوب بین نقطه و خوشه‌اش نسبت به خوشه مجاور است. اگر معیار نیم‌رخ برای همه نقاط درون خوشه‌ها نزدیک به ۱ باشد، عمل خوشه‌بندی به درستی انجام شده است. در حالیکه کوچک بودن مقدار نیم‌رخ برای خوشه‌ها، بیانگر ضعیف بودن نتایج خوشه‌بندی است که ممکن است به علت انتخاب نامناسب تعداد خوشه‌ها (**k**) نیز باشد.

خوشه بندی با شبکه های خودسازمان ده SOM

۲ روش: ۱)SOM رو خوشه بندی کن، ۲) داده رو با استفاده از SOM خوشه بندی کن (اینا فرق دارند)

توضیح روش ۲): داده رو به SOM میدیم، SOM اونو Train میکنه بصورتی که همگرا شده، بعدش بردار داده رو میدیم به SOM معلوم میشه نورون برنده ش کدومه، حالا بجای مقدارهای اصلی متغیرها در داده، میایم وزن نورون برنده هر داده رو میزاریم، یه داده جدید می سازیم (مثل کاهش بُعد)، حالا داده رو میدیم KMeans خوشه بندی می کنیم. (اینجا نورون خالی شرکت نمیکنه).

میشه از ماتریس فاصله یکسان استفاده کرد تا مرزهای خوشه های مستقل رو برجسته و شناسایی کنه و از یه الگوریتم به اسم watershedding استفاده کرد تا مولفه ها رو شناسایی کنه. واسه همین باید مناطق تو ماتریس مقعر باشند. (Concave)

میشه یک نقشه کوچک نسبی استفاده کرد و هرگره را به عنوان یک خوشه مدنظرگرفت، اول اینکه som رو بسازی و اونو train کنی. بعدش som رو خوشه بندی کنی، (پویا یا ایستا).

SOM

قبل از مصورسازی داده ها باید نرمال و عددی باشن، داده با ابعاد بالا رو به ۲ یا ۳بُعد تبدیل می کنه.میتونه با ترکیبی از ویژگی ها داده رو گسسته کنه.

تهیه و تنظیم: محمد حیدری

ارشد مهندسی فناوری اطلاعات

گرایش شبکه های پیچیده

دانشکده مهندسی سیستم

دانشگاه تربیت مدرس تهران

M_Heydari@Modares.ac.ir

منابع: چیستیو، فرادرس، ویکی پدیا، کلاس درس داده کاوی دکترخطیبی در تربیت مدرس تهران

خوشه با تراکم بالا، شکل و قطردلخواه، کدام شاخص اعتبارسنجی کمک می کند؟

استفاده از شاخص های سیلوهت: single link، وقتی تراکم بالا هستش یعنی شباهت درون خوشه زیاده، complete link کمک نمیکنه

ارزیابی مقایسه تطابق نتایج خوشه بندی با برچسب کلاس داده با چه روش های صورت می گیرد؟

با استفاده از روش External Index، کلا کاربرد روش External دو تا است: ۱)مقایسه دو خوشه بندی با هم ۲) مقایسه خوشه بندی با برچسب کلاس

شاخص ارزیابی بیرونی External Index

برای همه نقاط یک برچسب Benchmark وجود دارد و نشان می دهد تعلق نقاط به کدام دسته هاست.

شاخص ها: ۱) خلوص : درصد مطابقت بین برچسب های واقعی و خوشه بندی ۲)شاخص رند، نمایش میزان شباهت بین ۲ روش برچسب گذاری، معمولا به برچسب‌های واقعی، «استاندارد طلایی» نیز می‌گویند. از طرفی «برچسب‌های خوشه‌بندی» نیز کد مربوط به خوشه‌ای است که یک نقطه درون آن قرار دارد. در روش ارزیابی بیرونی، مطابقت این دو گونه برچسب انجام می‌پذیرد. باید توجه داشت که ممکن است کدهای برچسب‌های حاصل از خوشه‌بندی با برچسب‌های واقعی یکسان نباشند. به این معنی که برچسب واقعی ۱ برای یک نقطه بیانگر متعلق بودن آن به دسته شماره ۱ است در حالیکه ممکن است شماره برچسب برای این نقطه در خوشه‌بندی برابر با ۴ باشد.

شاخص ارزیابی بیرونی External Index

شاخص خلوص: در این حالت برچسب هر خوشه با برچسب واقعی دسته‌ای که بیشترین اشتراک را دارد مطابقت پیدا کرده و تعداد نقاطی از خوشه که در دسته صحیح طبقه‌بندی شده‌اند شمارش می‌شوند. نسبت این تعداد به تعداد کل نقاط شاخص خلوص را می‌سازد. در تطابق کامل شاخص ۱ و عدم کمال شاخص خلوص: ۰،

خصوصیات شاخص خلوص

۱)سادگی در محاسبات۲)مستقل از تعداد خوشه‌ها:
شاخص خلوص به تعداد خوشه‌ها توجه ندارد. در نتیجه نمی‌توان این شاخص را به عنوان معیاری برای سنجش مطابقت تعداد خوشه‌ها نیز در نظر گرفت.
۳)کاهش کارایی با افزایش تعداد خوشه‌ها:
اگر تعداد خوشه‌ها زیاد باشد و هیچ هماهنگی نیز بین برچسب‌های واقعی و خوشه‌ای وجود نداشته باشد ممکن است شاخص خلوص به ۱ نزدیک شود که یک عیب برای چنین شاخصی است.

شاخص ارزیابی بیرونی External Index شاخص رند اصلاح شده:

نشان دادن میزان شباهت بین دو شیوه برچسب‌گذاری
۲ پارامتر دارد:
۱) تعداد زوج‌هایی که هم در خوشه‌ها و هم در دسته‌ها در کنار هم هستند. (یکسانی برچسب خوشه ها و برچسب دسته آنها)
۲) تعداد زوج‌هایی که هم در خوشه‌ها و هم در دسته‌ها از یکدیگر جدا هستند. (تفاوت برچسب خوشه ها و برچسب دسته آنها)

خصوصیات شاخص رند اصلاح شده:

۱)شاخص کارا برای مقایسه چندین روش۲)بدون وابستگی به تعداد خوشه‌ها۳)عدم حساسیت به تغییر برچسب‌ها

الگوریتم جنگل تصادفی Random Forest

توضیح استاد: همان bagging درخت تصمیم است فقط توی هر گره همه ویژگی ها رو برای اینکه کدام ویژگی رو ملاک تصمیم قرار بده شرکت نمیده، معمولا یه زیرمجموعه تصادفی میگیره به اندازه رایدکال n که nتعداد خود ویژگی هاست و از بین شان بهترین را انتخاب می کند.
در نهایت هم می تواند داده رو با bagging درخت تصمیم تقسیم بندی کنه هم اینکه به هر ویژگی یک وزنی میده.
فرمولش: اولا متوسط وزن این ویژگی تو تمام درخت ها
چطور وزنش رو حساب میکنه؟ این ویژگی در چند گره استفاده شده، در هرگره ای که استفاده شده چقدر عدم خلوص رو بهبود داده.

اگر حجم داده زیاد باشد و در حافظه جا نشود الگوریتمی برای آن پیشنهاد دهید.

باید Sampling (نمونه برداری) انجام داد که بتوان برایش مدل ساخت. نمونه ها می تواند غیرهمپوشان باشند. برای هر نمونه یک دسته بند جدا می سازیم که هیچ اشتراک و همپوشانی ندارند.
سپس داده تست رو به همین مدل و تک تک دسته بندها می فرستیم (مثلا رای گیری اکثریت می زنیم)، در آخر هم تست و ارزیابی مدل. البته از Map Reduce هم می توانیم استفاده کنیم.

مزایای انتخاب ویژگی

بهبود کارایی الگوریتم‌های یادگیری ماشین، درک داده، کاهش داده کلی، کاهش مجموعه ویژگی‌ها، سادگی و قابلیت استفاده از مدل‌های ساده‌تر و کسب سرعت

فیلتر را در یک ستون اعمال می کنیم و آنقدر این کار را انجام می دهیم تا به مجموعه ویژگی که می خواهیم برسیم.

فیلتر:سریع، تعمیم خوب، گاهی اوقات تعمیم دردسرساز می شود و مجموعه ویژگی ها برای دسته بندها بهینه نخواهد بود، به عنوان فاز پیش پردازش استفاده می شود.

Wrapper: یادگیرنده بعنوان جعبه سیاه درنظرگرفته می شود، رابط جعبه سیاه به منظور امتیازدهی به زیرمجموعه ای از متغیرها مطابق با قدرت پیش بینی یادگیرنده ها به هنگام استفاده از زیرمجموعه ها استفاده می شود، نتایج برای یادگیرنده های مختلف متفاوت است، نیاز به تعریف ۲ مورد داریم: ۱)چطور فضای زیرمجموعه های متغیر ممکنه را جستجو کنیم؟ ۲) چطور عملکرد پیش بینی یادگیرنده را ارزیابی کنیم؟

Embedded: انتخاب متغیر را در فاز آموزش انجام می دهد. خاص یک ماشین یادگیری است که بهش داده میشه. مثال: الگوریتم WINNOW

«فیلترها» (Filters) بر ویژگی‌های کلی مجموعه داده آموزش تکیه دارند و فرآیند انتخاب ویژگی را به عنوان یک گام پیش پردازش با استقلال از الگوریتم استقرایی انجام می‌دهند. مزیت این مدل‌ها هزینه محاسباتی پایین و توانایی تعمیم خوب آن‌ها محسوب می‌شود.

«بسته‌بندها» (Wrappers) شامل یک الگوریتم یادگیری به عنوان جعبه سیاه هستند و از کارایی پیش‌بینی آن برای ارزیابی مفید بودن زیرمجموعه‌ای از متغیرها استفاده می‌کنند. به عبارت دیگر، الگوریتم انتخاب ویژگی از روش یادگیری به عنوان یک زیرمجموعه با بار محاسباتی استفاده می‌کند که از فراخوانی الگوریتم برای ارزیابی هر زیرمجموعه از ویژگی‌ها نشات می‌گیرد. با این حال، این تعامل با دسته‌بند منجر به نتایج کارایی بهتری نسبت به فیلترها می‌شود.

«روش‌های توکار» (Embedded) انتخاب ویژگی را در فرآیند آموزش انجام می‌دهند و معمولا برای ماشین‌های یادگیری خاصی مورد استفاده قرار می‌گیرند. در این روش‌ها، جست‌وجو برای یک زیرمجموعه بهینه از ویژگی‌ها در مرحله ساخت دسته‌بند انجام می‌شود و می‌توان آن را به عنوان جست‌وجویی در فضای ترکیبی از زیر مجموعه‌ها و فرضیه‌ها دید. این روش‌ها قادر به ثبت وابستگی‌ها با هزینه‌های محاسباتی پایین‌تر نسبت به بسته‌بندها هستند.

کلا فیلتر از همه سریعتر است.

۵ نوع اصلی توابع ارزیابی بر اساس فیلتر و wrapper

- فیلتر: فاصله، اطلاعات(انتروپی، info gain)، همبستگی : ضریب همبستگی، سازگاری
- Wrapper: نرخ خطای دسته بند

مقایسه روش های ارزیابی متدهای مختلف انتخاب ویژگی

روش	عمومیت	پیچیدگیt	دقت
فاصله	بله	کم	–
اطلاعات	بله	کم	–
وابستگی	بله	کم	
سازگاری	بله	متوسط	–
نرخ خطا	–	بالا	عالی

رتبه بندی بر اساس سرعت
فاصله و سازگاری سریع نیستند.
۱)وابستگی، ۲)اطلاعات، ۳)فاصله، ۴)سازگاری ۵)نرخ خطا
در ناسازگاری ترجیح ما وجود چند متغیر است، برعکسش در information و dependency تکی تکی چک می کنیم. بهمین دلیل ناسازگاری کُندتر است.

ادامه سیلوئت

حال اگر میانگین مقدار نیم‌رخ برای نقطه‌های هر خوشه را محاسبه کنیم، معیاری برای ارزیابی هر خوشه بدست می‌آید. همچنین میانگین کل مقدارهای نیم‌رخ نیز معیاری برای ارزیابی عملیات خوشه‌بندی محسوب می‌شود. برای تفسیر این معیار، از نموداری استفاده می‌شود که میزان انطباق هر نقطه را با خوشه خودش نمایش می‌دهد. در تصویر زیر این نمودار دیده می‌شود. محور افقی نقطه‌ها و ستون‌ها، مقدار معیار نیم‌رخ برای آن نقطه است. همچنین میانگین شاخص نیم‌رخ برای همه نقاط نیز در نمودار مشخص می‌شود. همانطور که در نمودار دیده می‌شود، برای خوشه شماره ۲ بعضی نقاط دارای مقدار نیم‌رخ منفی هستند که نشان می‌دهد ممکن است به درستی خوشه‌بندی نشده باشند و به خوشه مجاور تعلق داشته باشند. همچنین میانگین کل شاخص نیم‌رخ نیز برابر با ۰,۴۶ محاسبه شده است.

شاخص ارزیابی دیویس-بولدین (Davies-Bouldin)

وابسته به تعداد خوشه‌ها و یا الگوریتم خوشه‌بندی نیست. برای محاسبه این شاخص ابتدا باید با دو معیار «اندازه پراکندگی» (Dispersion measure) و «عدم شباهت بین خوشه‌ها» (Cluster dissimilarity) آشنا شویم.

اندازه پراکندگی درون خوشه

فرض کنید S i میزان پراکندگی مربوط به خوشه C i و d نیز یک تابع فاصله باشد. آنگاه میزان پراکندگی برای این خوشه توسط رابطه زیر قابل محاسبه است:

$$S_i = [\frac{1}{|C_i|} \sum_{x \in C_i \$\$} d^r(x, c_i)]^{(\frac{1}{r})}, \quad r > 0$$

این رابطه در حقیقت شبیه فاصله مینکوفسکی نقطه‌های هر خوشه از مراکز آن است

عدم شباهت (فاصله) بین خوشه‌ها

فاصله بین دو خوشه نیز بر اساس فاصله بین دو نقطه مرکزی آن‌ها سنجیده می‌شود. اگر V i و V j مراکز خوشه‌های i و j باشند، فاصله بین این دو خوشه با D i j نشان داده شده و توسط رابطه زیر بدست می‌آید:

$$D_{ij} = [\sum d(V_i, V_j)^t]^{\frac{1}{t}}$$

باز هم به نظر می‌رسد از فاصله مینکوفسکی برای سنجش فاصله بین دو خوشه استفاده شده است. حال با توجه به این دو مفهوم می‌توان میزان فاصله بین دو خوشه C i و C j را که با R i j نشان می‌دهیم به صورت زیر محاسبه کنیم:

$$R_{ij} = \frac{S_i + S_j}{D_{ij}}$$

همانطور که دیده می‌شود در صورت کسر، میزان پراکندگی دو خوشه با یکدیگر جمع شده و در مخرج نیز میزان عدم شباهت بین خوشه‌ها قرار گرفته است. هر چه خوشه‌ها دارای پراکندگی بیشتری باشند، مقدار R i j بزرگتر می‌شود. از طرفی اگر دو خوشه با یکدیگر فاصله کمتری داشته باشند باز هم R i j بزرگ می‌شود.
به این ترتیب برای محاسبه شاخص دیویس-بولدین برای یک روش خوشه‌بندی کافی است ابتدا بیشنیه فاصله هر خوشه را نسبت به خوشه‌های دیگر بدست آورد. یعنی برای خوشه iام خواهیم داشت:

$$R_i = \max_{j/i} Rij$$

سپس میانگین بیشینه فاصله‌های محاسبه شده برای همه خوشه‌های ایجاد شده توسط الگوریتم را محاسبه می‌کنیم. این شاخص را با V D B نشان می‌دهند.

$$V_{DB} = \frac{\sum_{i=1}^{k} R_i}{k}$$

در حقیقت این شاخص، میانگین حداکثر نسبت پراکندگی درون به پراکندگی بین خوشه‌ها را محاسبه می‌کند. هر چه مقدار شاخص V D B کمتر باشد، عمل خوشه‌بندی بهتر صورت گرفته است.

شاخص دان (Dunn's Index)

با دو معیار «فاصله» (Cluster Distance) و قطر (Diameter)، میزان فشردگی و تفکیک‌پذیری را محاسبه می کند.

حال اگر فاصله بین دو خوشه C i و C j را با D (C i , C j) نشان دهیم، می‌توانیم میزان تفکیک‌پذیری در خوشه‌بندی را به صورت زیر محاسبه کنیم:

$$D(C_i, C_j) = \min_{x \in C_i, y \in C_j} d(x, y)$$

همینطور برای اندازه‌گیری فشردگی خوشه‌ها، از قطر هر خوشه استفاده می‌شود. برای خوشه C l مقدار قطر توسط رابطه زیر بدست می‌آید

$$diam(C_l) = \max_{x,y \in C_l} d(x, y)$$

حال شاخص دان به صورت زیر تعریف می‌شود.

$$V_D = [\frac{\min_{i=1 \le j \le k} D(C_i, C_j)}{\max_{1 \le l \le k} diam(C_l)}]$$

در صورت این کسر، فاصله بین دو خوشه به عنوان معیاری برای تفکیک‌پذیری دیده می‌شود و در مخرج نیز قطر هر خوشه دیده می‌شود. نسبت این دو، مقیاسی برای سنجش فاصله بین دو خوشه خواهد بود. بنابراین کوچکترین مقدار این نسبت برای همه خوشه‌ها، می‌تواند شاخصی برای ارزیابی خوشه‌بندی باشد. هر چه مقدار این شاخص بزرگتر باشد، بیانگر تفکیک‌پذیری بهتر و در نتیجه خوشه‌بندی موثرتر است. بر همین اساس اگر نسبت میزان تفکیک‌پذیری به قطر خوشه‌ها مقدار بزرگی باشد، خوشه‌بندی به خوبی انجام شده است