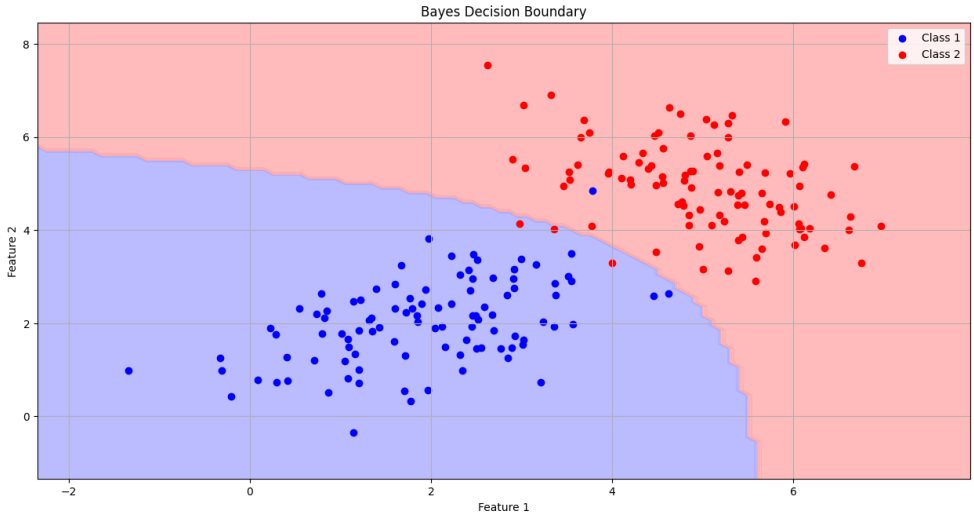
به نام خدا

نوید زارع

40435071

شناسایی آماری الگو



قانون تصمیم‌گیری

در Bayes classifier ، قانون تصمیم بر اساس نسبت likelihood است:

$$\Lambda(x) = \frac{P(x|\omega_1)}{P(x|\omega_2)}$$

اگر $\Lambda(x) > \frac{P(\omega_2)}{P(\omega_1)}$ باشد، نقطه را به ω_1 نسبت می‌دهیم، وگرنه به ω_2. چون در اینجا prior ها برابرند، این نسبت برابر یک می‌شود. پس قانون ساده می‌شود: اگر $\Lambda(x) > 1$ یعنی $P(x|\omega_1) > P(x|\omega_2)$ ، کلاس اول را انتخاب کن، وگرنه کلاس دوم.

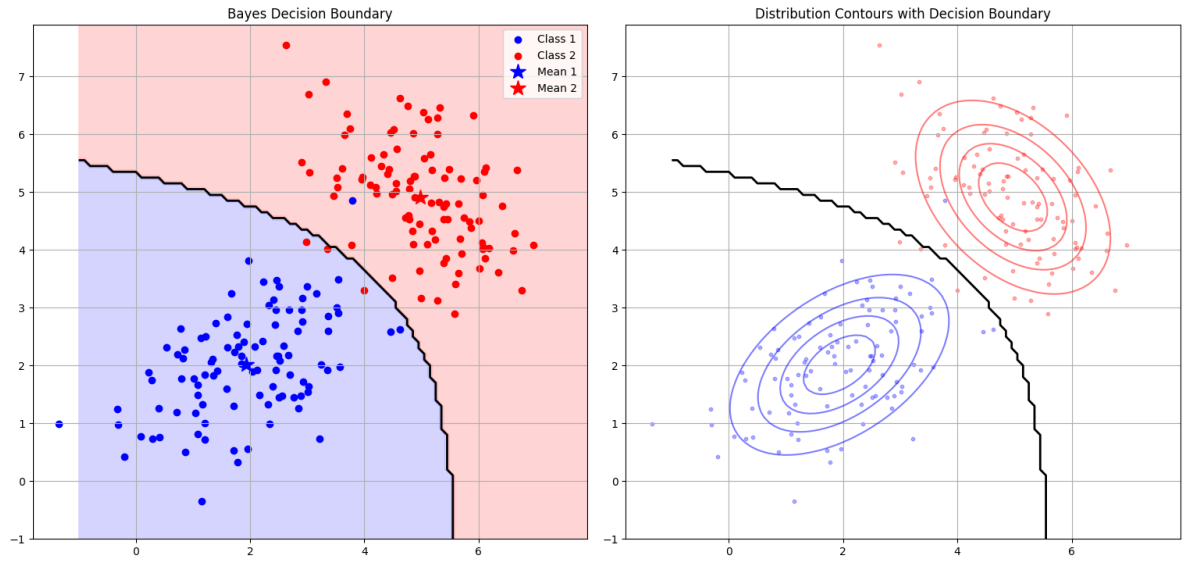
تقسیم فضا

ناحیه آبی (پایین چپ) به کلاس اول تعلق و ناحیه قرمز (بالا راست) به کلاس دوم تعلق دارد چون نقاط نزدیک به میانگین کلاس دوم هستند.

مرز تصمیم

مرز مشکی دقیقاً جایی است که $\Lambda(x) = 1$ یا به عبارت دیگر $P(x|\omega_1) = P(x|\omega_2)$. روی این خط، هر دو کلاس به طور مساوی محتمل هستند.

مرز به شکل منحنی است نه خط مستقیم، چون دو کلاس ماتریس کوواریانس متفاوت دارند.



1. Eigenvalues and Eigenvectors (eigen_vals_vects.py)

کلاس اول

در این تجزیه و تحلیل ما سعی می‌کنیم داده‌ها را از زاویه‌ای نگاه کنیم که بهترین تفکیک و پراکندگی را نشان دهد . PC1 حدود ۷۶٪ (دقیق ۷۵.۸۵٪) از واریانس را توضیح می‌دهد. این یعنی اگر فقط PC1 را نگه داریم و کلاس اول را یک‌بعدی کنیم، بیشتر اطلاعات حفظ می‌شود اما همچنان مقداری اطلاعات از دست می‌رود. وقتی بخواهیم دوباره به فضای دوبعدی برگردیم، خطای بازسازی دقیقاً برابر با lambda_2 خواهد بود یا به عبارتی بهتر برابر با نقطه ها به اندازه واریانس دوم که حذف کردیم ممکن است از مکان اصلی خود فاصله داشته باشند.

در خروجی مشاهده می شود که خطای بازسازی با یک PC برابر ۰.۴۷۴۵ است که تقریباً برابر با lambda_2 یعنی ۰.۴۷۴ می‌باشد. این تطابق نشان می‌دهد که دقیقاً به اندازه واریانسی که در مؤلفه دوم بود، اطلاعات از دست رفته است.

PC2 زاویه بهتری نسبت به PC1 ارائه نمی‌دهد چون بر اساس شکل و توزیع نقاط داده، پراکندگی بیشتری در جهت PC1 وجود دارد. eigenvalue اول (۱.۴۹۰) تقریباً سه برابر eigenvalue دوم (۰.۴۷۴) است که نشان می‌دهد داده‌ها بیشتر در یک جهت کشیده شده‌اند.

عمود بودن دو eigenvector ها به این معنی است که دو جهت مؤلفه‌های اصلی مستقل از هم هستند.

وقتی از هر دو مؤلفه اصلی استفاده می‌کنیم، خطای بازسازی صفر می‌شود چون ما با داده دوبعدی کار می‌کنیم و هر دو بعد را حفظ کرده‌ایم. تمام اطلاعات باقی مانده است.

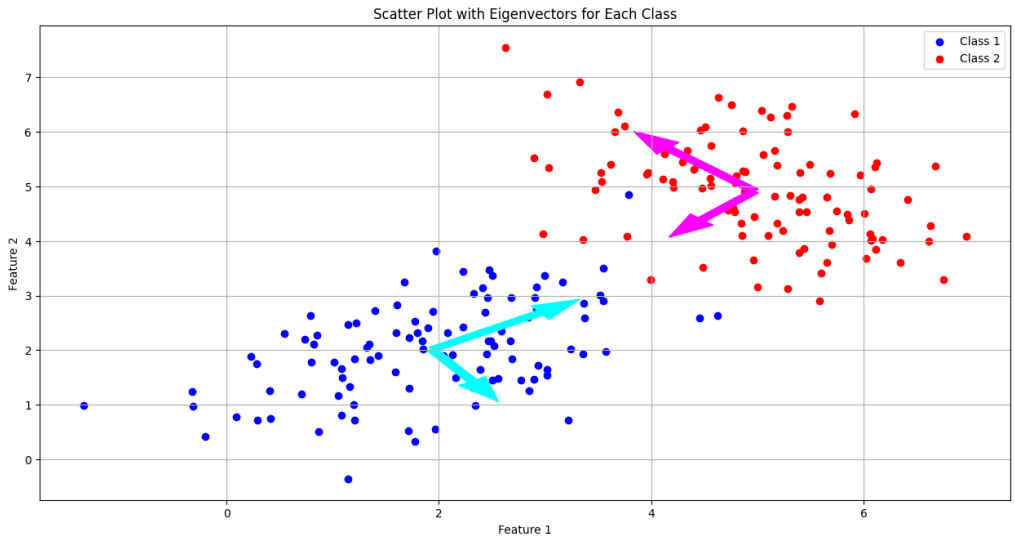
کلاس دوم

برای کلاس دوم، PC1 حدود ۷۱٪ (دقیق ۷۱.۱۷٪) از واریانس را توضیح می‌دهد. همانند کلاس اول، اگر فقط PC1 را نگه داریم، اطلاعاتی از دست می‌رود و خطای بازسازی برابر با lambda_2 خواهد بود. در این مورد، خطای بازسازی ۰.۵۱۹۲ است که تقریباً برابر با lambda_2 یعنی ۰.۵۱۹ می‌باشد.

نکته جالب اینجاست که خطای بازسازی کلاس دوم (۰.۵۱۹۲) بیشتر از کلاس اول (۰.۴۷۴۵) است. دلیلش این است که lambda_2در کلاس دوم بزرگ‌تر است، یعنی واریانس بیشتری در جهت مؤلفه دوم وجود دارد که با حذف آن از دست می‌رود.

eigenvalueها نشان‌دهنده واریانس واقعی در هر جهت مؤلفه اصلی هستند. برای کلاس اول، lambda_1 برابر ۱.۴۹۰ یعنی پراکندگی داده‌ها در جهت مؤلفه اول برابر ۱.۴۹۰ است و در جهت دوم فقط ۰.۴۷۴. این نسبت تقریباً سه به یک است.

ها ۱.۲۸۲ و ۰.۵۱۹ هستند که نسبتی حدود دو و نیم به یک می‌دهد. این یعنی کلاس eigenvalueبرای کلاس دوم، اول بیشتر کشیده شده است، در حالی که کلاس دوم نسبت به اولی کمتر کشیده است. این تفاوت در شکل را می‌توان برای بنفش برای کلاس اول کشیده‌تر هستند و فلش‌های آبی در نمودار پراکندگی هم دید، جایی که فلش‌های رنگی کلاس دوم متوازن‌تر به نظر می‌رسند.



چند مؤلفه اصلی کافی است؟

کلاس	مؤلفه اصلی	Eigenvalue	Explained Variance
کلاس ۱	PC1	1.490	75.85%
کلاس ۱	PC2	0.474	24.15%
کلاس ۲	PC1	1.282	71.17%
کلاس ۲	PC2	0.519	28.83%

این جدول اطلاعات مهمی درباره ساختار داده‌ها به ما می‌دهد. برای تصمیم‌گیری درباره تعداد مؤلفه‌های اصلی مورد نیاز، باید به دو معیار نگاه کنیم: درصد واریانس توضیح داده شده توسط هر مؤلفه.

در کلاس اول، مؤلفه اصلی اول تنها ۷۶٪ (دقیق ۷۵.۸۵٪) از کل تغییرات را توضیح می‌دهد. این یعنی اگر فقط یک مؤلفه را نگه داریم، حدود یک چهارم از اطلاعات از دست می‌رود. این مقدار قابل توجه است و نمی‌توان آن را نادیده گرفت. در کلاس دوم وضعیت حتی بارزتر است، چون مؤلفه اول فقط ۷۱٪ (دقیق ۷۱.۱۷٪) واریانس را پوشش می‌دهد و تقریباً ۲۹٪ در مؤلفه دوم باقی می‌ماند

هیچ کدام از دو کلاس یک مؤلفه اصلی غالب و قوی ندارند. در بسیاری از کاربردهای PCA که موفق هستند، مؤلفه اول ممکن است ۹۰ درصد یا بیشتر از واریانس را توضیح دهد، و در آن صورت می‌توان با اطمینان گفت که یک مؤلفه کافی است. اما در این داده‌ها، نسبت eigenvalue ها برای کلاس اول حدود سه به یک و برای کلاس دوم حدود دو و نیم به یک است، که نشان می‌دهد مؤلفه دوم هنوز سهم قابل توجهی دارد!

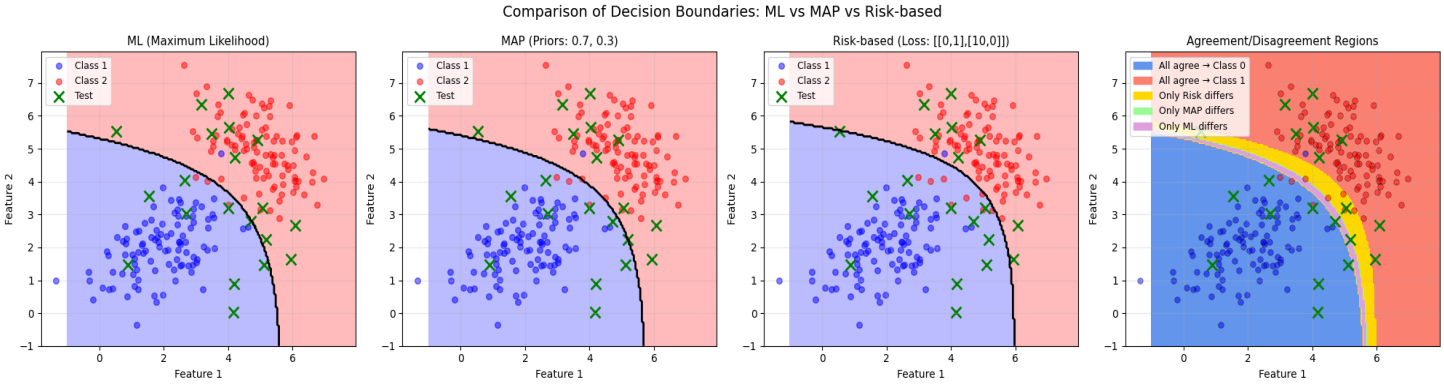
2 .Bayesian Decision Rules and Decision Boundaries (bayes_decision_boundary.py, ml_mlp_risk.py)

درک مرزهای تصمیم‌گیری:

در این بخش سه روش مختلف برای تصمیم‌گیری را بررسی کردیم

- ML (Maximum Likelihood)
- MAP (Maximum A Posteriori)
- Risk-based MAP classifier

هر کدام از این روش‌ها مرز تصمیم‌گیری را به شکل متفاوتی تعریف می‌کنند.



ML (Maximum Likelihood) Classifier

طبقه‌بند ML تصمیم خود را زمانی می‌گیرد که log p(x|Class1) - log p(x|Class2) >= 0 باشد. این روش فقط بر اساس اینکه کدام توزیع احتمال بیشتری دارد که نقطه مشاهده شده را تولید کرده باشد، عمل می‌کند. در این روش فرض می‌شود که هر دو کلاس از قبل احتمال یکسانی دارند.

مرز تصمیم‌گیری ML دقیقاً جایی قرار می‌گیرد که دو توزیع Gaussian چگالی احتمال برابری دارند. در شکل Distribution Contours می‌توان دید که مرز مشکی بین contour های آبی و قرمز قرار گرفته، جایی که توزیع‌ها با هم برابر می‌شوند.

این شکل Bayes Decision Boundary با prior های برابر در واقع معادل ML است، چون وقتی prior ها برابر باشند، تأثیری در تصمیم‌گیری ندارند.

MAP (Maximum A Posteriori) Classifier

طبقه‌بند MAP دانش قبلی ما را با اضافه کردن log(prior1/prior2) به قانون تصمیم‌گیری وارد می‌کند. بیایید این را فقط از دید ریاضی نگاه کنیم: ما یک مقدار ثابت به اختلاف log pdf ها اضافه می‌کنیم. در واقع داریم می‌گوییم که مرز تصمیم‌گیری باید کمی به بالا یا پایین جابجا شود تا با واقعیتی که ما باور داریم، هم‌راستا باشد.

در این مسئله، prior1 برابر ۰.۷ و prior2 برابر ۰.۳ است. نسبت prior ها برابر ۰.۳/۰.۷ = ۲.۳۳ می‌شود و log این مقدار یک عدد مثبت است. این عدد مثبت باعث می‌شود مرز از مرکز کلاس اول دور شده و به سمت ناحیه کلاس دوم حرکت کند.

این جابجایی یک نوع bias ایجاد می‌کند که بسته به مسئله و نحوه اعمال آن می‌تواند خوب یا بد باشد. در این مثال، چون ما باور داریم کلاس اول رایج‌تر است، مرز را طوری جابجا می‌کنیم که راحت‌تر نقاط را به کلاس اول نسبت دهیم.

Risk-based Classifier

طبقه‌بند مبتنی بر ریسک یک قدم جلوتر می‌رود و عواقب اشتباهات را در نظر می‌گیرد. ما داریم می‌گوییم که هر prior اهمیت متفاوتی دارد. اشتباه در طبقه‌بندی کلاس دوم به عنوان کلاس اول ممکن است در دنیای واقعی عواقب بیشتری داشته باشد نسبت به اشتباه در جهت عکس.

در این مسئله، ماتریس loss برابر [[0, 1], [10, 0]] است. این یعنی اگر یک نقطه واقعاً از کلاس اول باشد و ما آن را کلاس دوم طبقه‌بندی کنیم، هزینه ۱۰ واحد می دهیم، اما اگر اشتباه معکوس رخ دهد فقط ۰.۱ واحد هزینه داریم.

این ریسک‌ها به نوعی وزن به prior ها اضافه می‌کنند. ما می‌گوییم که prior کلاس اول برای ما مهم‌تر است که اشتباه طبقه‌بندی نشود.

تفسیر جابجایی مرز تصمیم

مرز تصمیم‌گیری را می‌توان به عنوان یک خط (یا منحنی) در نظر گرفت که فضای ویژگی‌ها را تقسیم می‌کند. مرز ML جایی قرار می‌گیرد که دو توزیع Gaussian چگالی احتمال برابری دارند. وقتی در MAP مقدار prior ها را اضافه می‌کنیم، این مرز را به سمت کلاسی که احتمال کمتری دارد هل می‌دهیم که در اینجا کلاس دوم است .

در قسمت Agreement/Disagreement Regions مشخص است که طبقه‌بند Risk مرز را بیشتر از همه به سمت کلاس دوم هل داده است. نواحی زرد نشان می‌دهند جایی که فقط Risk با دو روش دیگر اختلاف دارد.

تحلیل توافق تصمیمات

در خروجی کد می‌بینیم که از ۲۰ نقطه تست:

- MLو MAP روی ۱۹ نقطه توافق دارند و فقط روی ۱ نقطه اختلاف دارند
- MLو Risk-MAP روی ۱۷ نقطه توافق دارند و روی ۳ نقطه اختلاف دارند
- MAPو Risk-MAP روی ۱۸ نقطه توافق دارند و روی ۲ نقطه اختلاف دارند
- هر سه روش روی ۱۷ نقطه کاملاً با هم موافق هستند

روش طبقه‌بندی	دقت کل	خطای کلاس ۱	خطای کلاس ۲
ML (Maximum Likelihood)	98.0%	1.0%	**3.0%**
MAP (Maximum A Posteriori)	98.0%	1.0%	**3.0%**
Risk-based	95.5%	0.65%	**3.0%**

دلیل اینکه ML و MAP فقط روی یک نقطه اختلاف دارند این است که prior های (۰.۷، ۰.۳) نسبتاً متعادل هستند و تأثیر زیادی روی مرز ندارند. اما Risk به خاطر نسبت شدید ۱۰ به ۱ در ماتریسloss ، روی ۳ نقطه با ML اختلاف پیدا می‌کند.

اختلافات بیشتر در نواحی نزدیک به مرز تصمیم رخ می‌دهند، جایی که عدم قطعیت بیشتر است. نقاطی که دور از مرز هستند و به وضوح به یکی از کلاس‌ها تعلق دارند، توسط هر سه روش یکسان طبقه‌بندی می‌شوند.

کاربرد در دنیای واقعی

این جابجایی‌های مرز تصمیم چیز مهمی درباره تصمیم‌گیری در دنیای واقعی نشان می‌دهند. مثلاً در تشخیص پزشکی، ممکن است از ماتریس loss مشابه همین مثال استفاده کنیم اگر کلاس یک نشان‌دهنده یک بیماری جدی باشد. از دست دادن بیماری (پیش‌بینی سالم بودن در حالی که فرد واقعاً بیمار است (False Negative)) می‌تواند کشنده باشد. در این صورت، ما ترجیح می‌دهیم که مرز را طوری جابجا کنیم که احتمال این نوع خطا را کاهش دهیم، حتی اگر به قیمت افزایش خطاهای جهت مخالف باشد.

خلاصه

ساختار داده‌ها

Eigenvalue ها مقدار واریانس در هر جهت اصلی را نشان می‌دهند و نسبت آن‌ها شکل داده را مشخص می‌کند. کلاس اول با نسبت ۳ به ۱ بیشتر کشیده است، کلاس دوم با نسبت ۲.۵ به ۱ کمتر کشیده تراست. Eigenvectorها جهت‌های اصلی را تعیین می‌کنند و خطای بازسازی با یک مؤلفه برابر eigenvalue مؤلفه دوم است. هر دو مؤلفه برای نمایش کامل داده لازم هستند.

مرزهای تصمیم‌گیری

مرز Bayes جایی است که نسبت likelihood برابر نسبت prior ها می‌شود. چون دو کلاس ماتریس کوواریانس متفاوت دارند، مرز منحنی است نه خطی ML .فقط روی توزیع‌های داده تمرکز دارد، MAPبا اضافه کردن prior ها مرز را جابجا می‌کند، و Risk-based با در نظر گرفتن هزینه خطاها مرز را بیشتر تغییر می‌دهد.

تأثیر Prior ها و ریسک

Priorهای (۰.۷، ۰.۳) تأثیر کمی داشتند، MLو MAP هر دو دقت ۹۸ درصد رسیدند. اما ماتریس loss دقت را به ۹۵.۵ درصد کاهش داد اما خطای کلاس ۲ را افزایش داد تا از خطای گران‌تر جلوگیری کند. این نشان می‌دهد وقتی هزینه خطاها متفاوت است، باید دقت کلی را قربانی کنیم تا ریسک را کاهش دهیم.