Математические методы распознавания образов, ВМК МГУ, Семинар №8 Метрики качества для бинарных классификаторов.

Брескану Никита

Октябрь, 2024

1 Общие сведения о бинарных классификаторах

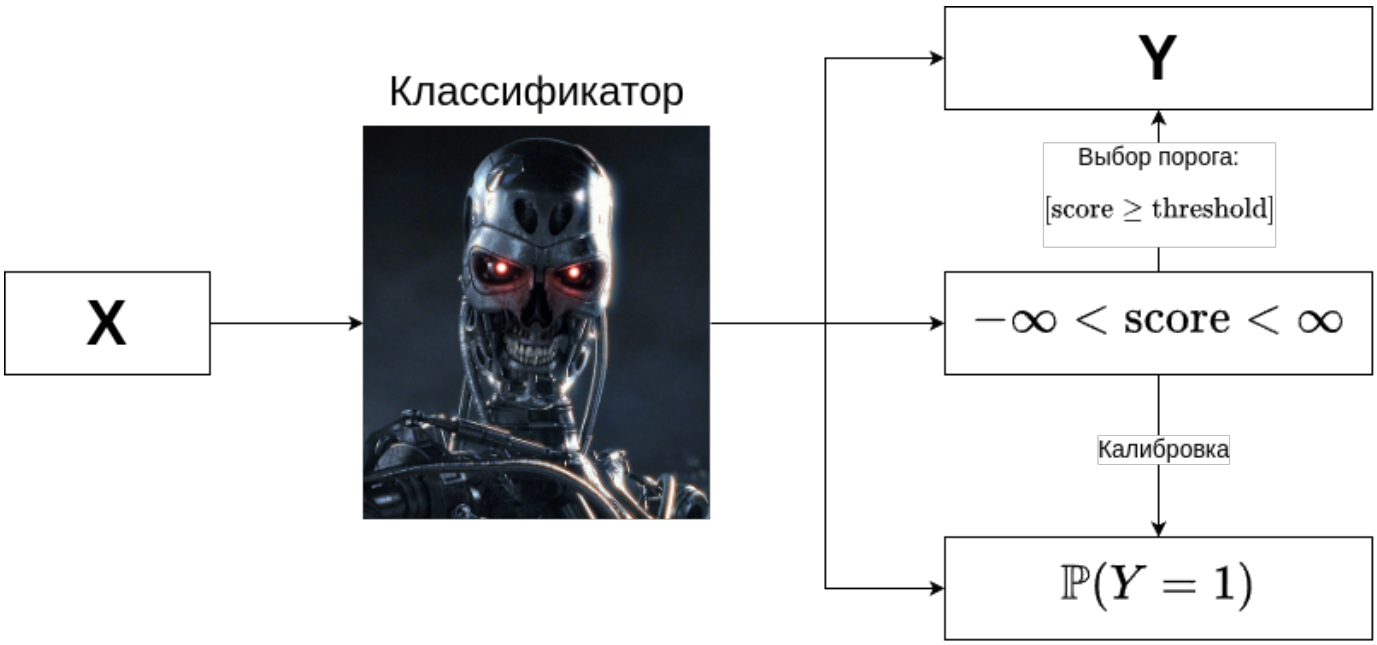
Вначале поговорим о бинарной классификации в целом. Что вы понимаете ввиду под бинарным классификатором? Что ему поступает на вход, а что он выдаёт на выходе?

На вход всегда подаётся объект, условно обозначенный за X. А вот выходное значение может быть разным. Чаще всего, оно бывает одним из 3 видов:

- метка класса, условно обозначенная за Y. Например, **KNN**.
- какую-то оценку (score), которая может быть любым числом. После этого обычно фиксируют порог (threshold), и класс выбирается следующим образом: $Y = [\text{score} \geq \text{threshold}]$. Например, **SVM**.
- вероятность положительного класса. Стоит заметить, что классификаторы предыдущего типа можно калибровать - тогда Например, **логистическая регрессия**, или **скалиброванный SVM**.

Для удобства будем называть их соотвественно **label-based** (или пороговый классификатор), **score-based** (ранговый) и **probability-based** (вероятностный).

В классической валидации ничего больше знать о модели не нужно. Фактически, на этой стадии она является «чёрным ящиком».



Также в бинарной классификации часто говорят о положительных и отрицательных классах. Это деление достаточно субъективно. Обычно положительным классом называют самый «интересный» случай. Например, для полиграфов - это присутствие лжи в ответе; для кредитования - дефолт; для пациентов - наличие заболевания.

2 Основные показатели качества

Пока что будем говорить о простых классификаторах, возвращающих только метку класса. Для валидации классификаторов обычно заранее отбирается отдельная выборка, отличная от тренировочной. Будем называть её **валидационной** (в контексте этой пары тестовая и валидационная выборка это одно и тоже; но обычно валидационная используется на стадии обучения, а тестовая - в самом конце).

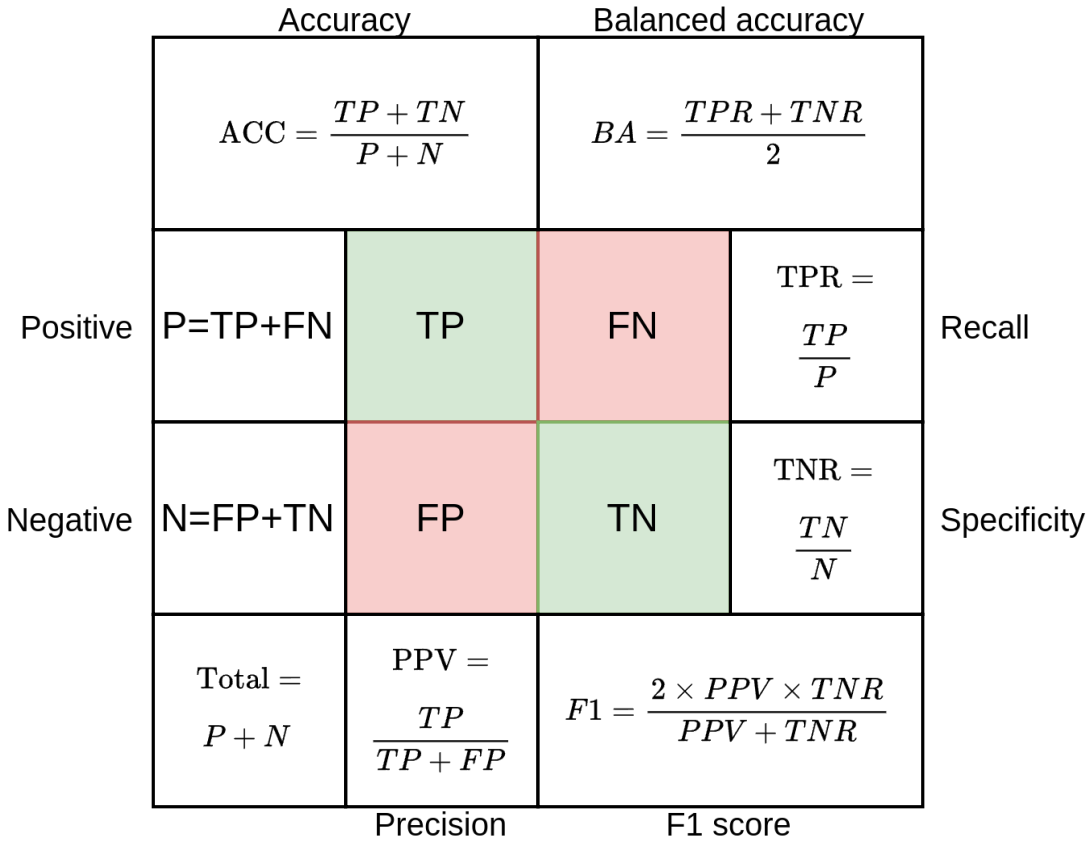
Нужно понимать, что валидационная выборка - это только очень малая часть из всех возможных объектов, с которыми может столкнуться модель. Все такие возможные объекты будем называть заумным словом - **генеральной совокупностью**.

Балансом классов будем называть соотношение между числом объектов положительного и отрицательного класса. Чаще всего при валидации делается предположение, что баланс классов в валидационной выборке совпадает с балансом классов на генеральной совокупности. Простыми словами это значит, что в продакшене положительные и отрицательные классы будут встречаться примерно в тех же пропорциях, что на стадии валидации. Будем называть выборку с таким балансом классов **репрезентативной**.

В основе валидации классификаторов лежит **матрица ошибок**. Она представляет собой матрицу 2x2. Названия ячеек можно запомнить следующим образом:

true / false - модель предсказала правильно / неправильно positive / negative - модель посчитала класс положительным / отрицательным.

Для краткости будем обозначать их: TP, FP, FN, TN.



Из построенной матрицы ошибок вытекают все показатели качества. Самые распространённые из них - это Accuracy, Precision, Recall, Specificity, F1 score, Balanced accuracy. Поговорим о каждой из них.

Accuracy (ACC) является самой интуитивно понятной метрикой - это число объектов, которые модель правильно предсказала. С точки зрения интерпретации, это частотная оценка вероятности модели выбрать правильный класс.

$$ACC = \frac{TP+TN}{P+N}$$

Однако надо понимать, что Accuracy смещена в сторону превалирующего класса. Например, если баланс классов 9:1, и модель предсказывает только 1ый класс, Accuracy $= 90\%$, но модель, очевидно, плохая. Поэтому рекомендуется пользоваться этим показателем только при близком к равному балансе классов.

$$PPV = \frac{TP}{TP+FP}$$

Precision (Positive predictive value, PPV) показывает, насколько модель уверена в предсказании положительного класса. Интерпретировать её можно как вероятность модели не ошибиться, если она предсказала положительный класс.

Например, в одном сериале главный герой сказал: «The problem with all polygraph tests is false positives». Это означает, что у полиграфов большое число FP, то есть они часто называют ложью то, что на самом деле правда. Следовательно, у них малый Precision. Также Precision сильно зависит от баланса классов.

$$TPR = \frac{TP}{P}$$

Recall (True positive rate, TPR) отвечает за качество работы модели только на положительных классах. Интерпретация: вероятность объявить класс положительным, если он на самом деле такой. Например, в детекции лжи Recall — это вероятность обнаружить ложь, если человек на самом деле лжёт.

Эта метрика не зависит от баланса классов, и её можно использовать, даже если вы не уверены, что выборка репрезентативна.

$$TNR = \frac{TN}{N}$$

Specificity (True negative rate, TNR) - это по сути Recall для другого класса. Если бы вы поменяли местами положительный и отрицательный класс, то Recall и Specificity перейдёт друг в друга. Поэтому иногда их называют Recall для класса 1 и Recall для класса 2 (аналогичное можно сказать и о Precision).

Balanced accuracy (BA) является аналогом Accuracy. Его отличие заключается в том, что он абсолютно не зависит от баланса классов, т.к. является средним двух Recall для обоих классов. Его можно применять, как общий показатель качества при сильном дисбалансе классов, либо при отсутсвии уверенности в репрезентативности выборки.

$$BA = \frac{TPR + TNR}{2}$$

2.1 F-score

F-мера [4] берёт начало из книги Van Rijsbergen'a «Information Retrieval». Он изучал нахождение релевантных документов по поисковому запросу. Автор показал, что Precision (доля релевантных документов из всех найденных) и Recall (доля найденных документов из всех релевантных) являются полезными метриками в этой задаче, и изъявлял желание объединить их в единый показатель качества алгоритма поиска. Изначально он назвал это эффективностью поиска, но по какой-то причине эту метрику спутали с другой F функцией из его книги, и на практике прижилось название $F - score$.

$$F_1 = \frac{2}{\frac{1}{PPV} + \frac{1}{TPR}} = \frac{2 \cdot PPV \cdot TPR}{PPV + TPR}$$

Автор аргументирует выбор среднего гармонического тем, что при Precision либо Recall, близких к нулю F1-score будет близок к нулю, что является разумным требованием. Напротив, среднее арифметическое позволяло бы иметь приличное значение метрики при малом Precision и большом Recall, или наоборот, т.е. при классификаторах, близких к константному предсказанию.

F1-score является одной из самых частоиспользуемых метрик, наряду с Accuracy. При этом можно заметить, что F1-score ценит Precision и Recall одинаково. Если же один этих 2 показателей важнее? чем другой (т.е. False positive и False negative случаи не равнозначны), то можно использовать обобщение Fβ-score, или просто F-score.

$$F_\beta = \frac{(1 + \beta^2) \cdot PPV \cdot TPR}{\beta^2 \cdot PPV + TPR}$$

Число β соответсвует тому, во сколько раз Precision важнее, чем Recall. Если $\beta >> 1$, то мера в основном зависит от Recall. Если $\beta << 1$, то - от Precision.

F1 score является частным случаем, если $\beta = 1$, т.е. Precision и Recall «равноправны».

2.2 Статистическая интерпретация задачи классификации

В курсе математической статистики вы, наверное, встречались с такими понятиями, как нулевая и альтернативная гипотеза. Если применить эти понятия к задаче классификации, мы получим следующее:

$$H_0 : Y = 0$$

$$H_1 : Y = 1$$

Где H_0 - нулевая гипотеза, а H_1 - альтернативная; 1 - положительный класс, 0 - отрицательный. В этой интерпретации классификатор является критерием (критерий - это вероятность отвергнуть нулевую гипотезу, т.е. посчитать класс положительным). Label-based - нерандомизированный критерий, а probability-based - рандомизированный.

При таком переносе понятий можно увидеть следующее соответсвие:

$$FP = \text{Ошибка 1 рода}$$

$$FN = \text{Ошибка 2 рода}$$

В статистике ошибка 1 рода является более затратной, чем 2 рода. Это ещё одно объяснение разделения классов на положительный и отрицательный. На нашем языке, FP дороже обойдётся, чем FN. Например, посчитать больного пациента здоровым хуже, чем посчитать здорового больным.

3 ROC-анализ

Теперь, имея базовое представление о пороговых (label-based) классификаторах, можно перейти к рассмотрению случая, когда можно выбирать порог (ранговые). Порог даёт свободу выбора человеку, применяющему модель. Однако теперь сравнение моделей стало ещё более сложным, т.к. показатели, обсуждённые ранее, меняется в зависимости от порога.

Вообще говоря, score может быть любым числом от $-\infty$ до ∞. Но далее будем считать его принадлежащим отрезку $[0, 1]$. Чаще всего, значение сводят от $(-\infty, \infty)$ к $(0, 1)$ с помощью сигмоиды.

Итак, мы хотим найти способ сравнения классификаторов, выдающих оценку (score). Самым простым способом будет выбрать порог, и далее уже смотреть на привычные показатели, например, Accuracy. Но порог сам по себе, строго говоря, нельзя интерпретировать (пока он не откалиброван).

3.1 Equal error rate

Рассмотрим один из способов сравнения, берущий начало из биометрических систем контроля. Например, когда вы открываете айфон, вам надо посмотреть в камеру, чтобы он разблокировался. В этот момент айфон решает задачу бинарной классификации:

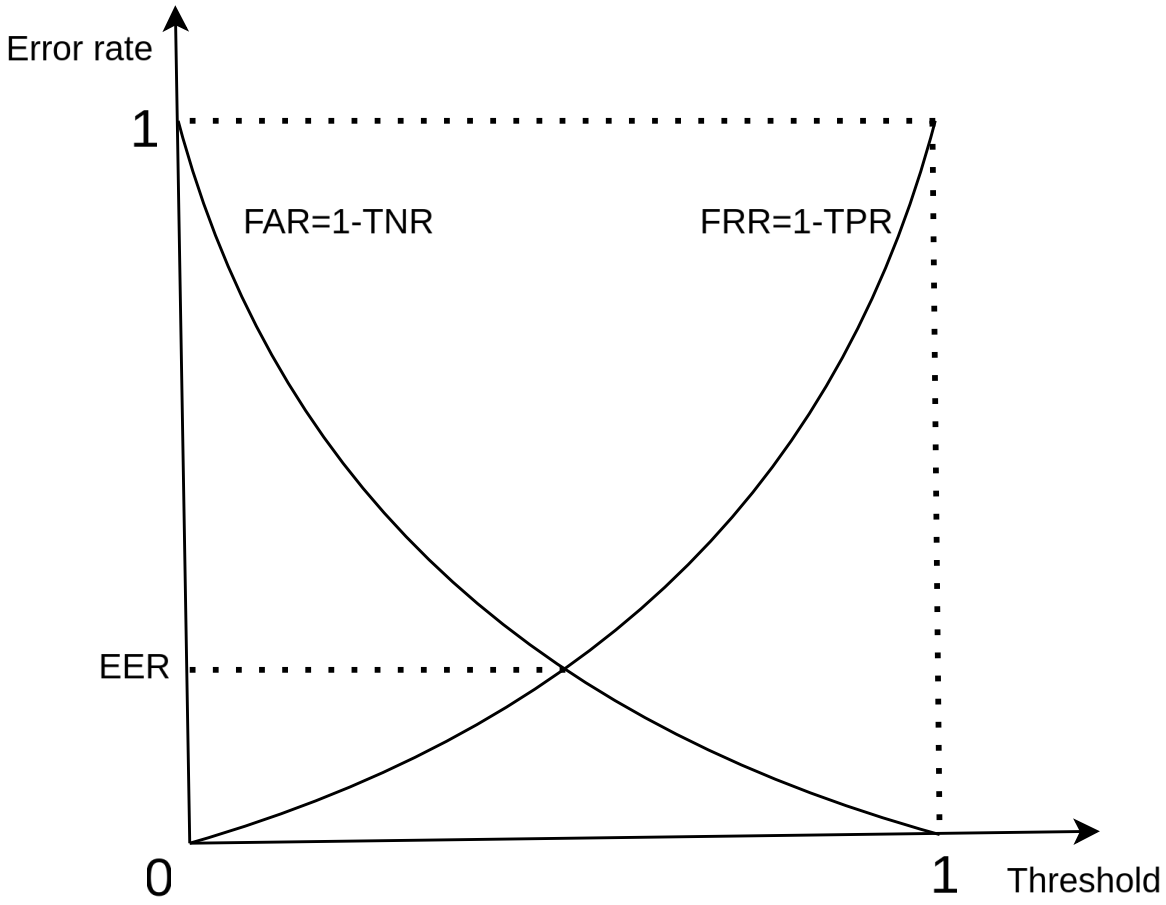
- P: Человек, смотрящий в камеру, является владельцем айфона.
- N: Не является.

В контексте биометрических систем используются свои термины:
False accept rate (FAR) - вероятность неавторизованного доступа к телефону.
False reject rate (FRR) - вероятность не предоставления доступа настоящему хозяину.
Эти величины тесно связаны с Recallами:

$$FAR = 1 - TNR$$

$$FRR = 1 - TPR$$

Нарисуем график, показывающий зависимость введённых 2 величин от порога.



Смотря на этот график, самой интересной для рассмотрения является точка пересечения двух кривых. В ней $FAR = FRR$, и следовательно, $TPR = TNR$. Поэтому значение error rate, соответсвующее этой точке, называют **Equal error rate** (ERR), $ERR = FAR = FRR$.

Классификаторы можно сравнивать с помощью ERR: чем она меньше, тем классификатор более качественный. В идеальном случае, $TPR = TNR = 1$, и $ERR = 0$, а кривые FAR и FRR как бы «сдавливаются» вниз к горизонтальной прямой.

3.2 ROC-кривая

ROC-кривая берёт начало во Второй мировой войне, в задаче детекции вражеских объектов на поле боя с помощью радара. Отсюда и её название: **Receiver Operating Characteristic (ROC) кривая** [1]. Дословно переводится, как «кривая, характеризующая работу приёмника». Приёмник собирает сигналы, которые по сути являются вольтажом, меняющимся со временем. Можно сказать, что в каждый момент времени решается следующая задача классификации:

- P: Вражеский объект обнаружен
- N: Сигнал является шумом

Вольтаж является результатом работы классификатора. Требуется выделить какой-то порог, чтобы уметь автоматически находить промежутки времени, когда объект был обнаружен.

Для наглядого изображения работы классификатора построим следующую кривую: На оси X будет $FPR = 1 - TNR$, на Y - TPR. То есть, это Recall1 vs 1 - Recall2 - взаимодействие двух Recallов в зависимости от порога.

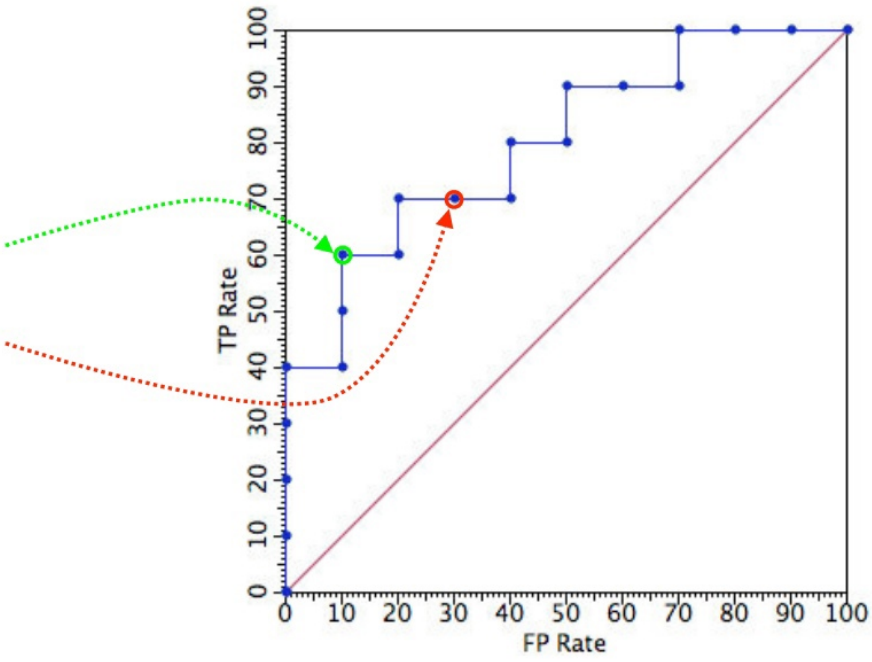
Этот график удобно строить следующим образом: расположим объекты по убыванию Score сверху вниз, обозначая положительный класс за $+$, и отрицательный за -. Далее будем фиксировать все возможные пороги, которые будут выглядеть, как «перегородки» между строками.

Для каждой перегородки можно легко посчитать TPR и FPR:

TPR - это число «+» сверху от неё, поделённое на общее число «+». FPR - это число «-» сверху, поделённое на общее число «-».

Соединяя эти точки, получаем ROC-кривую.

Class	Score
+	0.98
+	0.93
+	0.87
+	0.84
−	0.79
+	0.73
+	0.67
−	0.62
+	0.57
−	0.54
−	0.48
+	0.43
−	0.37
+	0.34
−	0.28
−	0.24
+	0.18
−	0.12
−	0.09
−	0.03



Проанализируем подробнее ROC-кривую: как видно на примера, она начинается из точки (0, 0) и заканчивается в (1, 1) - это крайние случае, для которых пороги 0 и 1.

Вырожденный классификатор (предсказывающий всегда один и тот же класс), будет иметь сторого диагональную ROC-кривую. В идеальном случае кривая должна заполнять весь квадрат и проходить через точку (0, 1). Чем «ближе» точка на кривой к точке (0, 1), тем она лучше. Но понятие «ближе» можно воспринимать по-разному.

Рассмотренний ранее EER можно легко найти по ROC-кривой: $EER = TPR = TNR = 1 - FPR$, т.е. на графике: $y = 1 - x$ - побочная диагональ в квадрате.

Полезно узнать, какой порог даст наибольший Accuracy. Для этого попытаемся нарисовать линии Accuracy на кривой:

$$ACC = \frac{TP + TN}{P + N} = \text{const}$$

$$ACC = \frac{P \cdot TPR + N \cdot (1 - FPR)}{P + N} = \text{const}$$

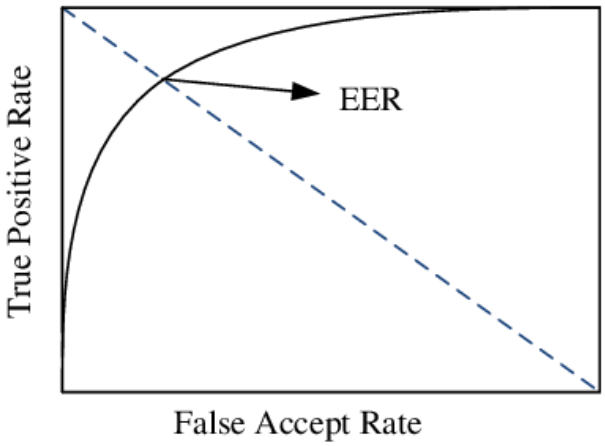
$$P \cdot TPR - N\dot{F}PR = \text{const}$$

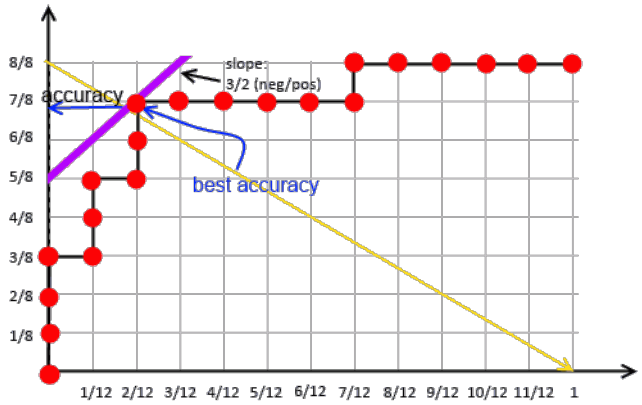
$$TPR = \frac{N}{P} FPR + \text{const}$$

Таким образом, линии уровня Accuracy являются прямыми с коэффициентом наклона $\alpha = \frac{N}{P}$.

Чем выше эта прямая (т.е. больше **const**), тем больше Accuracy. Таким образом, можно найти порог, дающий наибольший Accuracy.

Стоит заметить, что такую процедуру можно выполнять и с любой другой функцией от TPR и FPR . Пользователь модели может сам выбрать, какую функцию максимировать.





3.3 ROC-AUC

До этого момента мы познакомились с несколькими способами валидации и сравнения между собой score-based классификаторов:

- EER (чем меньше, тем лучше)
- Максимально возможный Accuracy (чем больше, тем лучше)
- Аналогично с другой функцией, например, F1-score

Все эти методы выбирают какой-то один порог, являющийся оптимальным в каком-то смысле, и далее смотрят на классические метрики в этом пороге. Помимо этого, существует семейство метрик, которые вычисляются по всей ROC-кривой, и оценивают классификатор в целом по всем порогам, а не только в какой-то области. Самым распространённым из них является **Area Under ROC Curve (ROC-AUC, AUC)** - площадь под ROC-кривой.

Рассмотрим подробнее ROC-AUC. Он принимает значения от 0 до 1, при этом для вырожденного классификатора (диагональная линия): $\frac{1}{2}$. Вначале, аналитически представим его в удобной для нас форме. Для удобства обозначим общее число объектов через $L = P + N$, а сами упорядоченные по возврастанию Score объекты (их классы) через $y_{(i)}$

$$
\begin{aligned}
ROCAUC &= \int_0^1 TPR(FPR)d(FPR) \\
&= \{\text{суммируем по всем порогам, в обратном порядке}\} = \sum_{t=1}^{L} TPR_t(FPR_t - FPR_{t+1}) \\
&= \sum_{t=1}^{L} \frac{\sum_{i=t}^{L}[y_{(i)}=1]}{P}\left(\frac{\sum_{i=t}^{L}[y_{(i)}=0] - \sum_{i=t+1}^{L}[y_{(i)}=0]}{N}\right) \\
&= \frac{1}{PN}\sum_{t=1}^{L}[y_{(t)}=0]\sum_{i=t}^{N}[y_{(i)}=1] \\
&= \{\text{слагаемое i=t всегда обращается в 0}\} = \frac{1}{PN}\sum_{t=1}^{L}[y_{(t)}=0]\sum_{i=t+1}^{N}[y_{(i)}=1] \\
&= \{\text{перестановка}\} = \frac{1}{PN}\sum_{i<j}^{L}[y_{(i)} < y_{(j)}]
\end{aligned}
\tag{1}
$$

В итоге мы получили, что ROC-AUC - это пропорция числа согласованных пар. В этой формуле не учитывается ситуация, когда у двух объектов разных классов одинаковый Score. В этом случае такие пары нужно добавить к результату, как $\frac{1}{2}\frac{1}{PN}\sum_{y_i<y_j}[s_i = s_j]$, где s_i - Score.

Можно заметить, что для ROC-AUC безразличны сами значения Score, а важно только взаимное расположение объектов (порядок) относительно этого Score.

Exercise 1:

Пусть валидационная выборка состоит из 4 положительных и 4 отрицательных объектов. Скоры для объектов приведены в таблице справа для 2 классификаторов. Построить для них ROC-кривую. Чему равны их ROC-AUC? Чему равны их наилучшие Accuracy? Чему равны EER? Какой из классификаторов лучше?

+	9	0.7
+	10	0.3
+	-7	0.2
+	2	1
-	4	0.1
-	-6	0.35
-	5	0.15
-	-8	0.9

3.4 Вероятностная интерпретация

Если перейти к генеральной совокупности, то финальная формула перейдёт в вероятность:

$$
ROCAUC = \mathbb{P}\left(s(X_1) < s(X_2) \mid y(X_1) < y(X_2)\right) + \frac{1}{2}\mathbb{P}\left(s(X_1) = s(X_2) \mid y(X_1) < y(X_2)\right)
$$

где X - случайная величина, объект; $y(X)$ - истинный класс объекта; $s(X)$ - Score.

Второе слагаемое почти всегда равно нулю, потому что модель чаще всего выдаёт непрерывные Score, а вероятность двух континуальных точек совпасть нулевая. Оно добавлено только ради симметрии.

Эмпирический ROC-AUC является статистической оценкой для вероятностного, где выборкой в пространстве объектов (генеральной совокупности) является валидационная выборка.

Интерпретировать это можно так: если выбрать случайный положительный и случайный отрицательный объект, с какой вероятностью модель поставит положительный над отрицательным.

Заметим, что если поменять местами классы: $y = 0 \longleftrightarrow y = 1$, при этом не трогая Score, то ROCAUC изменится следующим образом:

$$
\begin{aligned}
ROCAUC' &= \mathbb{P}\left(s(X_1) < s(X_2) \mid y'(X_1) < y'(X_2)\right) + \frac{1}{2}\mathbb{P}\left(s(X_1) = s(X_2) \mid y'(X_1) < y'(X_2)\right) \\
&= \mathbb{P}\left(s(X_1) < s(X_2) \mid y(X_1) > y(X_2)\right) + \frac{1}{2}\mathbb{P}\left(s(X_1) = s(X_2) \mid y(X_1) > y(X_2)\right) \\
&= \{X_1 \longleftrightarrow X_2\} = \mathbb{P}\left(s(X_1) > s(X_2) \mid y(X_1) < y(X_2)\right) + \frac{1}{2}\mathbb{P}\left(s(X_1) = s(X_2) \mid y(X_1) < y(X_2)\right) \\
&= \mathbb{P}\left(s(X_1) \geq s(X_2) \mid y(X_1) < y(X_2)\right) - \frac{1}{2}\mathbb{P}\left(s(X_1) = s(X_2) \mid y(X_1) < y(X_2)\right) \\
&= 1 - \mathbb{P}\left(s(X_1) < s(X_2) \mid y(X_1) < y(X_2)\right) - \frac{1}{2}\mathbb{P}\left(s(X_1) = s(X_2) \mid y(X_1) < y(X_2)\right) \\
&= 1 - ROCAUC
\end{aligned}
\tag{2}
$$

Таким образом, ROC-AUC симметричен относительно главной диагонали квадрата. При этом мы самим имеет право выбора положительного и отрицательного класса (например, мы можем заменить Score на отрицательный, что будет эквивалентно этому). Поэтому если ROC-AUC $< \frac{1}{2}$, то мы можем сделать такую замену, получив 1 - ROC-AUC, который станет $> \frac{1}{2}$.

Посмотрим, что произойдёт с ROC-AUC, если классификатор не зависит от объекта (в какой-то степени является вырожденным). Тогда при смене классов ничего не поменяется, ведь зависимости от объектов нет (в вероятностях условность пропадает). Мы получим:

$$
ROCAUC = 1 - ROCAUC \implies ROCAUC = \frac{1}{2}
$$

Таким образом вырожденные и случайные классификаторы всегда имеют вероятностный ROC-AUC $\frac{1}{2}$.

Exercise 2:

Что произойдёт с ROC-AUC классификатора, если его выходные скоры умножить на 2? умножить на 0.5? умножить на -1?

3.5 PR-AUC

Помимо ROC-кривой, для score-based классификаторов можно построить ещё одну кривую: по оси x откладывается Recall, а по y — Precision. При этом перебираются всевозможные пороги.

Таким образом, образуется непрерывная кривая, называемая **Precision-Recall кривой** (PR-кривой). В отличие от ROC-кривой, здесь кривая начинается из точки (0, 1) — предсказания только отрицательных классов с порогом $-\infty$ и заканчивается в точке (1, 0) — предсказание только положительных классов с порогом $+\infty$. Однако для левого порога $-\infty$ следует понимать, что Precision вообще говоря, является делением $\frac{0}{0}$ и не определён, поэтому имеется ввиду левый предел.

Очевидно, что функция $PPV = f(TPR)$ является монотонно невозрастающей, т.к. при увеличении порога Precision может только уменьшиться, а Recall увеличиться.

Идеальный классификатор имеет при каком-нибудь пороге $PPV = 1$ и $TPR = 1$, поэтому кривая проходит через точку (1, 1) и описывает весь единичный квадрат. Интуитивно, чем «выше» кривая, тем она лучше.

Подумаем над тем, что происходит у вырожденного (константного) классификатора, предсказывающего только какое-нибудь константное значение s:

- $t \leq s \implies \hat{y} = [a(x) \geq t] \equiv 1 \implies PPV = \frac{P}{P+N}, TPR = 1$, т.е. проходит через точку (1, $\frac{P}{P+N}$)
- $t > s \implies \hat{y} \equiv 0 \implies$ Precision не определён, поэтому эту точку не рисуем.

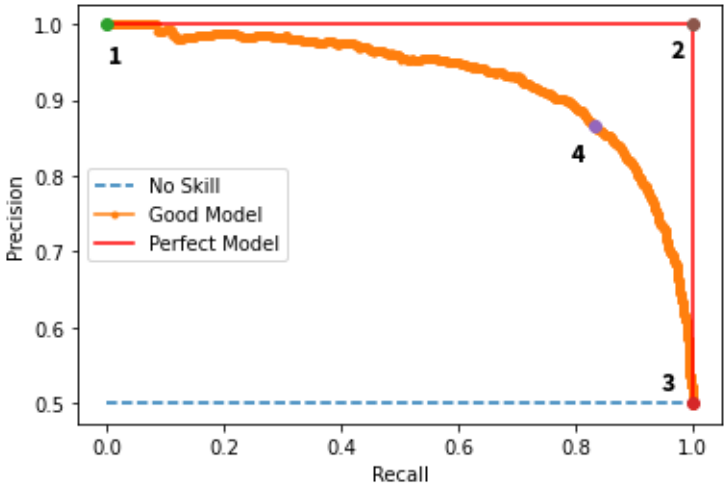


Рис. 2: PR-кривая

По сути перебор всех порогов даёт только одну точку, однако для наглядности в этом случае неопределённый Precision доопределяется значением $\frac{P}{P+N}$, и кривая принимает вид горизонтальной прямой.

Ранее уже было замечено, что чем выше кривая, тем лучше. Поэтому использование площади под PR-кривой в качестве метрики является разумным. Такая площадь называется **Precision-Recall AUC** (PR-AUC).

У идеального классификатора кривая описывает весь единичный квадрат, поэтому $PRAUC = 1$.

У вырожденного классификатора прямая описывает прямоугольник, и площадь равна: $PRAUC = \frac{P}{P+N}$. Если классификатор лучше, чем константный, его кривая будет находиться над горизонтальной прямой константного, а $PRAUC > \frac{P}{P+N}$.

В отличие от ROC-AUC, PR-AUC не имеет какого-то определённого вероятностного смысла, и просто является искуственно придуманной метрикой. Его часто используют в ситуациях, когда наблюдается существенный дизбаланс классов (например, 1:9), и положительный класс гораздо реже встречается, чем отрицательный класс. В этом случае, константный PR-AUC $\frac{P}{P+N}$ будет близок к нулю, что облегчает интерпретацию метрики.

Одним из примеров таких задач является прогноз обращения по страховому случаю. Очевидно, из всех, кто оплачивает страховку, только у очень малого количества людей возникнет страховой случай. Поэтому доля положительных классов мала: $\frac{P}{P+N} \approx 0$, и здесь может иметь смысл использование PR-AUC вместо ROC-AUC.

3.6 Сравнение PR-AUC и ROC-AUC

Многие считают, что PR-AUC лучше, чем ROC-AUC для сильно дизбалансированных классов. Одним из отличий PR-AUC от ROC-AUC является то, что PR-AUC не является симметричным. То есть, если начать считать отрицательный класс положительным, то PR-AUC поменяется, а ROC-AUC — нет. Вообще, вопрос о правильности использования PR-AUC вместо ROC-AUC очень сложный, и я не буду на него отвечать в этом семинаре, а только приведу статью [3], где читатель может ознакомиться с этим сам.

4 Многоклассовая классификация

Теперь, когда мы более-менее разобрали бинарную классификацию, можно перейти на случай, когда классов больше, чем 2. Пусть, их K штук: $1, 2, \ldots, K$.

Естественным образом обобщается матрица ошибок с 2×2 до $K \times K$ матрицы, где строки — это реальные классы, а столбцы — классы, предсказанные моделью. Верные предсказания будут также находиться на диагонали.

Для удобства обозначим через TP_i число правильно предсказанных объектов i-ого класса, а через E_{ij} — число ошибок путём предсказания класса i, когда на самом деле класс j.

$\text{F1}_{micro} = \frac{2 \times PPV_{micro} \times TPR_{micro}}{PPV_{micro} + TPR_{micro}}$	$FP_1 = E_{12} + \ldots + E_{1K}$	$FP_2 = E_{21} + \ldots + E_{2K}$	...	$FP_K = E_{K1} + E_{K2} + \ldots +$	$\text{TPR}_{micro} = \frac{\overline{TP}}{\overline{TP} + \overline{FN}}$
$\text{F1}_{macro} = \frac{1}{K} \sum_{i=1}^{K} \frac{2 \times PPV_i \times TPR_i}{PPV_i + TPR_i}$					$\text{TPR}_{macro} = \frac{TPR_1 + \ldots + TPR_K}{K}$
$\text{ACC} = \frac{TP_1 + \ldots + TP_K}{L_1 + \ldots + L_K}$					
$FN_1 = E_{21} + \ldots + E_{K1}$	TP_1	E_{21}	...	E_{K1}	$\text{TPR}_1 = \frac{TP_1}{TP_1 + FN_1}$
$FN_2 = E_{12} + \ldots + E_{K2}$	E_{12}	TP_2	...	E_{K2}	$\text{TPR}_2 = \frac{TP_2}{TP_2 + FN_2}$
...	...	...	...	...	...
$FN_K = E_{1K} + E_{2K} + \ldots$	E_{1K}	E_{2K}	...	TP_K	$\text{TPR}_K = \frac{TP_K}{TP_K + FN_K}$
$\overline{TP} = \frac{TP_1 + \ldots + TP_K}{K}$	$\text{PPV}_1 = \frac{TP_1}{TP_1 + FP_1}$	$\text{PPV}_2 = \frac{TP_2}{TP_2 + FP_2}$	...	$\text{PPV}_K = \frac{TP_K}{TP_K + FP_K}$	$\text{PPV}_{micro} = \frac{\overline{TP}}{\overline{TP} + \overline{FP}}$
$\overline{FP} = \frac{FP_1 + \ldots + FP_K}{K}$					$\text{PPV}_{macro} = \frac{PPV_1 + \ldots + PPV_K}{K}$
$\overline{FN} = \frac{FN_1 + \ldots + FN_K}{K}$					

Accuracy имеет такой же смысл, как и в бинарном случае — это вероятность предсказать правильный класс.

$$ACC = \frac{TP_1 + \ldots + TP_K}{\sum_i TP_i + \sum_{i \neq j} E_{ij}}$$

Если модель предсказывает вероятности принадлежности каждому классу, то по ним можно создать и другие варианты Accuracy:

Обозначим через $TP_i@k$ число объектов, для которых класс i попал в первые k классов после сортировки по вероятностям. Тогда вводится $ACC@k$ как доля таких попаданий.

$$ACC@k = \frac{\sum_i TP_i@k}{\sum_i TP_i + \sum_{i \neq j} E_{ij}}$$

Очевидно, что $ACC = ACC@1$, $ACC@K = 1$, и значение $ACC@k$ возрастает по k. Для задач, где много классов (например, 100), часто используются такие **top-k** варианты Accuracy.

Теперь перейдём к Precision и Recall. Для каждого класса i можно свести задачу к бинарной классификации (**One-vs-Rest** сведение): тогда бинарная матрица ошибок вводится естественным образом: отрицательные классы - это совокупность всех классов кроме i.

$$TP_i = TP_i$$
$$FP_i = \sum_{i \neq j} E_{ij}$$
$$FN_i = \sum_{i \neq j} E_{ji}$$
$$TN_i = \sum_{j \neq i} TP_j$$

Для таких бинарных матриц ошибок мы уже умеем считать Precision и Recall:

$$PPV_i = \frac{TP_i}{TP_i + FP_i}$$
$$TPR_i = \frac{TP_i}{TP_i + FN_i}$$

Таким образом мы получили K метрик Precision и Recall. Но хочется видеть какую-то одну метрику, а не смотреть на K разных, поэтому разумно их усреднить:

$$PPV_{\text{macro}} = \frac{1}{K} \sum_{i=1}^{K} PPV_i$$
$$TPR_{\text{macro}} = \frac{1}{K} \sum_{i=1}^{K} TPR_i$$

То есть, из бинарных метрик, оперирующих над бинарной матрицей ошибок, мы перешли к многоклассым обобщениям этих метрик, путём усреднения соответсвующих для каждого класса значений. Назовём такое усреднение **макро-усреднением** (макро, потому что усредняем сами метрики).

Другой способ усреднять метрики — вначале усреднить соответсвующие для каждого класса элементы бинарной матрицы ошибок:

$$\overline{TP} = \frac{1}{K} \sum_{i=1}^{K} TP_i$$
$$\overline{FP} = \frac{1}{K} \sum_{i=1}^{K} FP_i$$
$$\overline{FN} = \frac{1}{K} \sum_{i=1}^{K} FN_i$$
$$\overline{TN} = \frac{1}{K} \sum_{i=1}^{K} TN_i$$

А затем считать метрики от этих усреднённых значений, как будто мы имеем дело с бинарной матрицей ошибок из усреднений.

$$PPV_{\text{micro}} = \frac{\overline{TP}}{\overline{TP} + \overline{FP}}$$

$$TPR_{\text{micro}} = \frac{\overline{TP}}{\overline{TP} + \overline{FN}}$$

Назовём такое усреднение **микро-усреднением** (микро, потому что усредняем вначале сами значения).

Микро-усреднение

Усреднение $\Longrightarrow$ подсчёт метрик

Все объекты одинаково важны

$$\overline{TP} = \frac{1}{K}\sum_{i=1}^{K} TP_i \qquad \overline{FN} = \frac{1}{K}\sum_{i=1}^{K} FN_i$$

$$\overline{FP} = \frac{1}{K}\sum_{i=1}^{K} FP_i \qquad \overline{TN} = \frac{1}{K}\sum_{i=1}^{K} TN_i$$

$$f_{\text{micro}} = f(\overline{TP}, \overline{FN}, \overline{FP}, \overline{TN})$$

Макро-усреднение

Подсчёт метрик $\Longrightarrow$ усреднение

Все классы одинаково важны

$$f_i = f(TP_i, FN_i, FP_i, TN_i)$$

$$f_{\text{macro}} = \frac{1}{K}\sum_{i=1}^{K} f_i$$

Главным отличием микро и макро усреднения заключается в том, как эти усреднения ведут себя при сильном дизбалансе классов: макро-усреднение относится к каждому классу одинаково (независимо от числа объектов в нём), а микро-усреднение будет учитывать в большей степени наиболее большие классы.

Exercise 3:

Пусть имеется матрица ошибок для классификации с 4 классами. Найти микро и макро Precision. Почему они различаются?

A	1	20	0	1
B	0	10	1	0
C	1	40	1	0
D	0	30	0	1

Таблица 1: Матрица ошибок

Аппендикс для ботанов

A Matthew's correlation coefficient

Очень популярной метрикой, показывающей себя как достойную замену F1-score, является **Matthew's correlation coefficient** (MCC) [6]. Эта метрика так называется, т.к. была впервые введена Brian W. Matthews в 1975 году. В этой секции мы выведем её самостоятельно как корреляцию между предсказанием модели и реальным классом объекта.

Пусть ξ — случайная величина, отвечающая за класс объекта, предсказанный моделью; η — настоящий класс. ξ и η принимают 0, если класс отрицательный, и 1, если положительный.

Найдём коэффициент корреляции Пирсона между этими двумя случайными величинами. При этом матожидания и дисперсии будем заменять их эмпирическими аналогами.

$$\mathbb{E}\xi = \frac{TP + FP}{P + N}$$

$$\mathbb{E}\eta = \frac{P}{P + N}$$

$$\mathbb{E}\xi\eta = \frac{TP}{P + N}$$

Эти равенства являются оценками матожиданий. Можно сказать, что имеет выборка $\xi_1, \ldots, \xi_{P+N}$, и $\eta_1, \ldots, \eta_{P+N}$, и по ней высчитывается среднее арифметическое.

Заметим, что $\xi^2 = \xi$ и $\eta^2 = \eta$, поэтому $\mathbb{E}\xi^2 = \mathbb{E}\xi$, $\mathbb{E}\eta^2 = \mathbb{E}\eta$.

$$\mathbb{D}\xi = \mathbb{E}\xi - (\mathbb{E}\xi)^2 = (\mathbb{E}\xi)(1 - \mathbb{E}\xi) = \frac{TP + FP}{P + N} \times \frac{TN + FN}{P + N} = \frac{(TP + FP)(TN + FN)}{(P + N)^2}$$

$$\mathbb{D}\eta = \mathbb{E}\eta - (\mathbb{E}\eta)^2 = (\mathbb{E}\eta)(1 - \mathbb{E}\eta) = \frac{P}{P + N} \times \frac{N}{P + N} = \frac{PN}{(P + N)^2}$$

Вспомним определение корреляции Пирсона:

$$\phi = \frac{\operatorname{cov}(\xi, \eta)}{\sqrt{\mathbb{D}\xi \times \mathbb{D}\eta}} = \frac{\mathbb{E}\xi\eta - \mathbb{E}\xi\mathbb{E}\eta}{\sqrt{\mathbb{D}\xi \times \mathbb{D}\eta}}$$

Теперь, подставим найденные величины, и получим MCC:

$$MCC = \frac{\frac{TP}{P+N} - \frac{(TP+FP)P}{(P+N)^2}}{\frac{\sqrt{(TP+FP)(TN+FN)PN}}{(P+N)^2}} = \frac{TP(P + N) - (TP + FP)P}{\sqrt{(TP + FP)(TN + FN)PN}} = \frac{TP \cdot N - FP \cdot P}{\sqrt{(TP + FP)(TN + FN)PN}}$$

Теперь вспомним, что $P = TP + FN$, $N = TN + FP$, и подставим это в выражение, получив окончательную формулу:

$$MCC = \frac{TP \cdot TN - FP \cdot FN}{\sqrt{(TP + FP)(TN + FN)(TP + FN)(TN + FP)}}$$

Первое отличие MCC от других метрик заключается в том, что он, как корррреляция, принадлежит отрезку [-1, 1]. При этом случаи ± 1 означают линейную зависимость двух величин, а более точно:

$$MCC = 1 \iff \xi = \eta$$

$$MCC = -1 \iff \xi = 1 - \eta$$

То есть $MCC = 1$ — это идеальный классификатор, а -1 — классификатор, всегда предсказывающий противоположный класс. Поэтому, если $MCC < 0$, имеет смысл поменять местами классы в модели, чтобы получить число, большее 0.

Самый худший случай — это $MCC = 0$. Это означает, что предсказания модели и реальные классы независимы (если быть точнее, нескоррелированны, из нулевой корреляции не следует независимость, но всем всегда всё равно на этот факт).

B Оптимизация ROC-AUC

ROC-AUC очень часто используется на практике для оценки классификаторов. Например, многие соревнования Kaggle используют его как главный показатель, по которому выбирают победителей.

По этому имеет смысл идея о прямой оптимизации этой метрики. Для удобства эмпирический ROC-AUC можно переписать, как:

$$ROCAUC = \frac{1}{PN} \sum_{i:y_i=0} \sum_{j:y_j=1} [s(x_i) < s(x_j)] = \frac{1}{PN} \sum_{i:y_i=0} \sum_{j:y_j=1} [s(x_j) - s(x_i) > 0]$$

Такая функция не является гладкой, поэтому её сложно оптимизировать стандартными методами. Поэтому часто используется её аппроксимация, называемая **Soft-AUC** [2]:

$$SoftAUC = \frac{1}{PN} \sum_{i:y_i=0} \sum_{j:y_j=1} \sigma_\beta(s(x_j) - s(x_i))$$

где $\sigma_\beta(x) = \frac{1}{1+\exp(-\beta x)}$

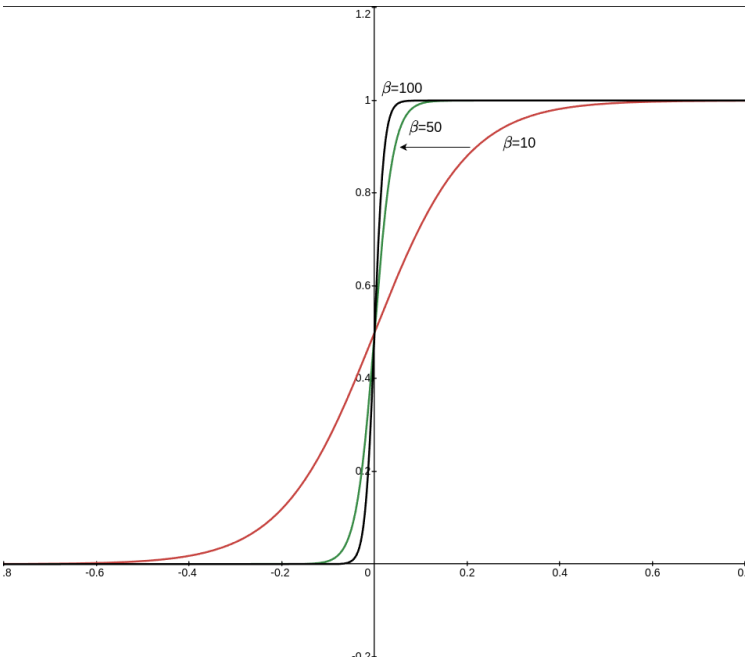


Рис. 3: Пояснение к аппроксимации индикатора

Однако у такой оптимизируемой функции есть недостаток: она требует прохождения по всем парам объектов разных классов, т.е. сложность O(PN), в то время как обычные функции потерь имеют O(P+N). Поэтому оптимизация может занять дольше времени.

C Многоклассовый ROC-AUC

В (4) мы поговорили про обобщения метрик для label-based классификаторов. Однако обычно модели возвращают **логиты** — значения для каждого класса. Если применить к ним softmax и откалибровать, можно получить вероятности принадлежности каждому классу.

Ввиду большой популярности ROC-AUC, имеет смысл попробовать обобщить его на многоклассовый случай. Для этого нужно как-то перейти к бинарной классификации, а затем считать ROC-AUC по

известной нам формуле (1), используя только скоры положительного класса, а затем усреднить эти значения. По сути, это является макро-усреднением.

Обозначим через $y_{j(n)}$ классы, упорядоченные по j-ой координате логитов.

One-vs-One ROC-AUC

Усреднение по всем $K(K-1)$ упорядоченным парам классов

Не чувствителен к балансу классов.

$$ROCAUC_{ij} = \frac{1}{L_i L_j} \sum_{m<n} [y_{j(m)} = i][y_{j(n)} = j]$$

$$ROCAUC_{\text{OvO}} = \frac{1}{K(K-1)} \sum_{i \neq j} ROCAUC_{ij}$$

One-vs-Rest ROC-AUC

Усреднение по K классам

Чувствителен к балансу классов.

$$ROCAUC_i = \frac{1}{L_i \sum_{j \neq i} L_j} \sum_{m<n} [y_{i(m)} \neq i][y_{i(n)} = i]$$

$$ROCAUC_{\text{OvR}} = \frac{1}{K} \sum_{i=1}^{K} ROCAUC_i$$

Различают 2 способа подсчёта многоклассового ROC-AUC: One-vs-One (OvO) и One-vs-Rest (OvR). В первом случае усреднение идёт по всем упорядоченным парам классов (порядок влияет на то, какой из классов будет положительным, т.е. по значению score которого будет считаться классический ROC-AUC), а во втором случае — по всем классам, где отрицательным классом считаются все классы, кроме положительного.

Список литературы

[1] Flach, Peter A. "ROC analysis."Encyclopedia of machine learning and data mining. Springer, 2016. 1-8.

[2] Calders, Toon, and Szymon Jaroszewicz. "Efficient AUC optimization for classification."European conference on principles of data mining and knowledge discovery. Berlin, Heidelberg: Springer Berlin Heidelberg, 2007.

[3] McDermott, Matthew, et al. "A closer look at auroc and auprc under class imbalance."arXiv preprint arXiv:2401.06091 (2024).

[4] Christen, Peter, David J. Hand, and Nishadi Kirielle. "A review of the F-measure: its history, properties, criticism, and alternatives."ACM Computing Surveys 56.3 (2023): 1-24.

[5] Flach, Peter A. "The geometry of ROC space: understanding machine learning metrics through ROC isometrics."Proceedings of the 20th international conference on machine learning (ICML-03). 2003.

[6] Chicco, Davide, and Giuseppe Jurman. "The advantages of the Matthews correlation coefficient (MCC) over F1 score and accuracy in binary classification evaluation."BMC genomics 21 (2020): 1-13.