Семинар
Метрики качества классификации

Е. А. Соколов
ФКН ВШЭ

24 октября 2023 г.

1 Функция потерь $\neq$ метрика качества

Как мы узнали ранее, методы обучения реализуют разные подходы к обучению:

1. обучение на основе прироста информации (как в деревьях решений)
2. обучение на основе сходства (как в методах ближайших соседей)
3. обучение на основе вероятностной модели данных (например, максимизацией правдоподобия)
4. обучение на основе ошибок (минимизация эмпирического риска)

Во время сведения задачи о построении решающего правила к задаче численной оптимизации, мы вводили понятие функции потерь и, обычно, объявляли целевой функцией сумму потерь от предсказаний на всех объектах обучающей выборке.

Важно понимать разницу между функцией потерь и метрикой качества. Её можно сформулировать следующим образом: Функция потерь возникает в тот момент, когда мы сводим задачу построения модели к задаче оптимизации. Обычно требуется, чтобы она обладала хорошими свойствами (например, дифференцируемостью). Метрика – внешний, объективный критерий качества, обычно зависящий не от параметров модели, а только от предсказанных меток.

В некоторых случаях метрика может совпадать с функцией потерь. Например, в задаче регрессии MSE играет роль как функции потерь, так и метрики. Но, скажем, в задаче бинарной классификации они почти всегда различаются: в качестве функции потерь может выступать кросс-энтропия, а в качестве метрики – число верно угаданных меток (accuracy). Отметим, что в последнем примере у них различные аргументы: на вход кросс-энтропии нужно подавать логиты, а на вход accuracy – предсказанные метки (то есть по сути argmax логитов).

2 Метрики качества классификации

Будем считать, что классификатор имеет вид $a(x) = \text{sign}(b(x) - t) = 2[b(x) > t] - 1$. Линейная модель имеет именно такую форму, если положить $b(x) = \langle w, x \rangle$ и $t = 0$.

§2.1 Доля правильных ответов

Наиболее очевидной мерой качества в задаче классификации является доля правильных ответов (accuracy):

$$\text{accuracy}(a, x) = \frac{1}{\ell}\sum_{i=1}^{\ell}[a(x_i) = y_i].$$

Данная метрика, однако, имеет существенный недостаток. Если взять порог t меньше минимального значения прогноза $b(x)$ на выборке или больше максимального значения, то доля правильных ответов будет равна доле положительных и отрицательных ответов соответственно. Таким образом, если в выборке 950 отрицательных и 50 положительных объектов, то при тривиальном пороге $t = \max_i b(x_i)$ мы получим долю правильных ответов 0.95. Это означает, что доля положительных ответов сама по себе не несет никакой информации о качестве работы алгоритма $a(x)$, и вместе с ней следует анализировать соотношение классов в выборке. Также полезно вместе с долей правильных ответов вычислять *базовую долю* — долю правильных ответов алгоритма, всегда выдающего наиболее мощный класс.

Отметим, что при сравнении различных методов машинного обучения принято сообщать относительное уменьшение ошибки. Рассмотрим два алгоритма a_1 и a_2 с долями правильных ответов r_1 и r_2 соответственно, причем $r_2 > r_1$. Относительным уменьшением ошибки алгоритма a_2 называется величина

$$\frac{(1 - r_1) - (1 - r_2)}{1 - r_1}.$$

Если доля ошибок была улучшена с 20% до 10%, то относительное улучшение составляет 50%. Если доля ошибок была улучшена с 50% до 25%, то относительное улучшение также равно 50%, хотя данный прирост кажется более существенным. Если же доля ошибок была улучшена с 0.1% до 0.01%, то относительное улучшение составляет 90%, что совершенно не соответствует здравому смыслу.

§2.2 Матрица ошибок

Выше мы убедились, что в случае с несбалансированными классами одной доли правильных ответов недостаточно — необходима еще одна метрика качества. В данном разделе мы рассмотрим другую, более информативную пару критериев.

Введем сначала понятие матрицы ошибок. Это способ разбить объекты на четыре категории в зависимости от комбинации истинного ответа и ответа алгоритма (см. таблицу 1). Через элементы этой матрицы можно, например, выразить долю правильных ответов:

$$\text{accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{FP} + \text{FN} + \text{TN}}.$$

Гораздо более информативными критериями являются *точность* (precision) и *полнота* (recall):

$$\text{precision} = \frac{\text{TP}}{\text{TP} + \text{FP}};$$

$$\text{recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}.$$

	$y = 1$	$y = -1$
$a(x) = 1$	True Positive (TP)	False Positive (FP)
$a(x) = -1$	False negative (FN)	True Negative (TN)

Таблица 1. Матрица ошибок

Точность показывает, какая доля объектов, выделенных классификатором как положительные, действительно является положительными. Полнота показывает, какая часть положительных объектов была выделена классификатором.

Рассмотрим, например, задачу предсказания реакции клиента банка на звонок с предложением кредита. Ответ $y = 1$ означает, что клиент возьмет кредит после рекламного звонка, ответ $y = -1$ — что не возьмет. Соответственно, планируется обзванивать только тех клиентов, для которых классификатор $a(x)$ вернет ответ 1. Если классификатор имеет высокую точность, то практически все клиенты, которым будет сделано предложение, откликнутся на него. Если классификатор имеет высокую полноту, то предложение будет сделано практически всем клиентам, которые готовы откликнуться на него. Как правило, можно регулировать точность и полноту, изменяя порог t в классификаторе $a(x) = \text{sign}(b(x) - t) = 2[b(x) > t] - 1$. Если выбрать t большим, то классификатор будет относить к положительному классу небольшое число объектов, что приведет к высокой точности и низкой полноте. По мере уменьшения t точность будет падать, а полнота увеличиваться. Конкретное значение порога выбирается согласно пожеланиям заказчика.

Отметим, что точность и полнота не зависят от соотношения размеров классов. Даже если объектов положительного класса на порядки меньше, чем объектов отрицательного класса, данные показатели будут корректно отражать качество работы алгоритма.

Существует несколько способов получить один критерий качества на основе точности и полноты. Один из них — F-мера, гармоническое среднее точности и полноты:

$$F = \frac{2 * \text{precision} * \text{recall}}{\text{precision} + \text{recall}}.$$

Среднее гармоническое обладает важным свойством — оно близко к нулю, если хотя бы один из аргументов близок к нулю. Именно поэтому оно является более предпочтительным, чем среднее арифметическое (если алгоритм будет относить все объекты к положительному классу, то он будет иметь recall = 1 и precision $\ll 1$, а их среднее арифметическое будет больше 1/2, что недопустимо). Можно заметить, что F-мера является сглаженной версией минимума из точности и полноты. Отметим, что геометрическое среднее также похоже на сглаженный вариант минимума, но при этом оно менее устойчиво к «выбросам» — например, для точности 0.9 и полноты 0.1 гармоническое среднее будет равно 0.18, а геометрическое 0.3.

Другим агрегированным критерием является *R-точность*, или точка баланса (breakeven point). Она вычисляется как точность при таком t, при котором полнота равна точности:

$$\text{R-precision} = \text{precision}(\text{sign}(b(x) - t^*)),$$
$$t^* = \arg\min_t |\text{precision}(\text{sign}(b(x) - t)) - \text{recall}(\text{sign}(b(x) - t))|.$$

Можно показать, что R-точность равна точности при таком пороге, при котором количество отнесённых к положительному классу объектов равно количеству положительных объектов в выборке.

Часто встречаются задачи, в которых целевой признак по-прежнему бинарный, но при этом необходимо ранжировать объекты, а не просто предсказывать их класс. Например, в задаче предсказания реакции клиента можно выдавать сортированный список, чтобы оператор мог в первую очередь позвонить клиентам с наибольшей вероятностью положительного отклика. Поскольку многие алгоритмы возвращают вещественный ответ $b(x)$, который затем бинаризуется по порогу t, то можно просто сортировать объекты по значению $b(x)$. Для измерения качества ранжирования нередко используют *среднюю точность* (average precision, AP):

$$\text{AP} = \frac{1}{\ell_+}\sum_{k=1}^{\ell}[y_{(k)} = 1]\text{precision@}k,$$

где $y_{(k)}$ — ответ k-го по порядку объекта, ℓ_+ — число положительных объектов в выборке, а precision@k — точность среди первых k в списке объектов. Если алгоритм $b(x)$ так ранжирует объекты, что сначала идут все положительные, а затем все отрицательные, то средняя точность будет равна единице; соответственно, чем сильнее положительные документы концентрируются в верхней части списка, тем ближе к единице будет данный показатель.

Связь точности, полноты и доли правильных ответов Выше мы отмечали, что высокие значения доли правильных ответов вовсе не влекут за собой высокое качество работы классификатора, и ввели точность и полноту как способ решения этой проблемы. Тем не менее, при выборе точности и полноты в качестве основных метрик, следует соблюдать осторожность при выборе требований к их значениям — как мы увидим из примера, независимость данных метрик от соотношения классов может привести к неочевидным последствиям.

Рассмотрим задачу бинарной классификации с миллионом объектов ($\ell = = 1.000.000$), где доля объектов первого класса составляет 1% ($\ell_+ = 10.000$). Мы знаем, что доля правильных ответов будет вести себя не вполне интуитивно на данной несбалансированной выборке, и поэтому выберем точность и полноту для измерения качества классификаторов. Поскольку мы хотим решить задачу хорошо, то введём требования, что и точность, и полнота должны быть не менее 90%. Эти требования кажутся вполне разумными, если забыть о соотношении классов. Попробуем теперь оценить, какая доля правильных ответов должна быть у классификатора, удовлетворяющего нашим требованиям.

Всего в выборке 10.000 положительных объектов, и для достижения полноты 90% мы должны отнести как минимум 9.000 к положительному классу. Получаем TP = 9000, FN = 1000. Так как точность тоже должна быть не меньше 90%, получаем FP = 1000. Отсюда получаем, что доля правильных ответов должна быть равна $(1 - 2.000/1.000.000) = 99.8\%$! Это крайне высокий показатель, и его редко удаётся достичь на таких выборках во многих предметных областях.

§2.3 Area Under Curve

Выше мы изучили точность, полноту и F-меру, которые характеризуют качество работы алгоритма $a(x) = \text{sign}(b(x) - t)$ при конкретном выборе порога t. Однако зачастую интерес представляет лишь вещественнозначный алгоритм $b(x)$, а порог будет выбираться позже в зависимости от требований к точности и полноте. В таком случае возникает потребность в измерении качества семейства моделей $\{a(x) = \text{sign}(b(x) - t) \mid t \in \mathbb{R}\}$.

Можно измерять качество этого множества на основе качества лучшего (в некотором смысле) алгоритма. Для этого подходит упомянутая ранее точка баланса (breakeven point). В идеальном семействе алгоритмов она будет равна единице, поскольку найдется алгоритм со стопроцентной точностью и полнотой. Данная метрика, однако, основывается лишь на качестве одного алгоритма, и не характеризует вариативность семейства.

Широко используется такая интегральная метрика качества семейства, как *площадь под ROC-кривой* (Area Under ROC Curve, AUC-ROC). Рассмотрим двумерное пространство, одна из координат которого соответствует доле неверно принятых объектов (False Positive Rate, FPR), а другая — доле верно принятых объектов (True Positive Rate, TPR):

$$\text{FPR} = \frac{\text{FP}}{\text{FP} + \text{TN}};$$
$$\text{TPR} = \frac{\text{TP}}{\text{TP} + \text{FN}}.$$

Каждый возможный выбор порога t соответствует точке в этом пространстве. Всего различных порогов имеется $\ell + 1$. Максимальный порог $t_{\max} = \max_i b(x_i)$ даст классификатор с TPR $= 0$, FPR $= 0$. Минимальный порог $t_{\min} = \min_i b(x_i) - \varepsilon$ даст TPR $= 1$ и FPR $= 1$. ROC-кривая — это кривая с концами в точках $(0, 0)$ и $(1, 1)$, которая последовательно соединяет точки, соответствующие порогам $b(x_{(1)}) - \varepsilon, b(x_{(1)}), b(x_{(2)}), \ldots, b(x_{(\ell)})$. Площадь под данной кривой называется AUC-ROC, и принимает значения от 0 до 1. Если порог t может быть подобран так, что алгоритм $a(x)$ не будет допускать ошибок, то AUC-ROC будет равен единице; если же $b(x)$ ранжирует объекты случайным образом, то AUC-ROC будет близок к 0.5.

Критерий AUC-ROC имеет большое число интерпретаций — например, он равен вероятности того, что для случайно выбранных положительного объекта x_+ и отрицательного объекта x_- будет выполнено[1] $b(x_+) > b(x_-)$.

Индекс Джини. В задачах кредитного скоринга вместо AUC-ROC часто используется пропорциональная метрика, называемая индексом Джини (Gini index):

$$\text{Gini} = 2\text{AUC} - 1.$$

По сути это умноженная на два площадь между ROC-кривой и диагональю, соединяющей точки $(0, 0)$ и $(1, 1)$.

Отметим, что переход от AUC к индексу Джини приводит к увеличению относительных разниц. Если мы смогли улучшить AUC с 0.8 до 0.9, то это соответствует относительному улучшению в 12.5%. В то же время соответствующие индексы

[1] При условии, что прогнозы на всех объектах выборки попарно различны.

Джини были улучшены с 0.6 до 0.8, то есть на 33.3% — относительное улучшение повысилось почти в три раза!

Precison-recall кривая. Избавиться от указанной проблемы с несбалансированными классами можно, перейдя от ROC-кривой к Precision-Recall кривой. Она определяется аналогично ROC-кривой, только по осям откладываются не FPR и TPR, а полнота (по оси абсцисс) и точность (по оси ординат). Критерием качества семейства алгоритмов выступает площадь под PR-кривой (AUC-PR). Данную величину можно аппроксимировать следующим образом. Стартуем из точки $(0, 1)$. Будем идти по ранжированной выборке, начиная с первого объекта; пусть текущий объект находится на позиции k. Если он относится к классу «-1», то полнота не меняется, точность падает — соответственно, кривая опускается строго вниз. Если же объект относится к классу «1», то полнота увеличивается на $1/\ell_+$, точность растет, и кривая поднимается вправо и вверх. Площадь под этим участком можно аппроксимировать площадью прямоугольника с высотой, равной precision@k и шириной $1/\ell_+$. При таком способе подсчета площадь под PR-кривой будет совпадать со средней точностью:

$$\text{AUC-PR} = \frac{1}{\ell_+}\sum_{k=1}^{\ell}[y_k = 1]\text{precision@}k.$$

Отметим, что AUC-PR дает разумные результаты в рассмотренном выше примере с классификацией статей. Так, при размещении 100 релевантных документов на позициях 50.001-50.101 в ранжированном списке, AUC-PR будет равен 0.001.

Несмотря на указанные различия, между ROC- и PR-кривой имеется тесная связь. Так, можно показать, что если ROC-кривая одного алгоритма лежит полностью над ROC-кривой другого алгоритма, то и PR-кривая одного лежит над PR-кривой другого [1].

§2.4 Метрики качества многоклассовой классификации

В многоклассовых задачах, как правило, стараются свести подсчет качества к вычислению одной из рассмотренных выше двухклассовых метрик. Выделяют два подхода к такому сведению: микро- и макро-усреднение.

Пусть выборка состоит из K классов. Рассмотрим K двухклассовых задач, каждая из которых заключается в отделении своего класса от остальных, то есть целевые значения для k-й задаче вычисляются как $y_i^k = [y_i = k]$. Для каждой из них можно вычислить различные характеристики (TP, FP, и т.д.) алгоритма $a^k(x) = [a(x) = k]$; будем обозначать эти величины как $\text{TP}_k, \text{FP}_k, \text{FN}_k, \text{TN}_k$. Заметим, что в двухклассовом случае все метрики качества, которые мы изучали, выражались через эти элементы матрицы ошибок.

При микро-усреднении сначала эти характеристики усредняются по всем классам, а затем вычисляется итоговая двухклассовая метрика — например, точность, полнота или F-мера. Например, точность будет вычисляться по формуле

$$\text{precision}(a, X) = \frac{\overline{\text{TP}}}{\overline{\text{TP}} + \overline{\text{FP}}},$$

где, например, $\overline{\text{TP}}$ вычисляется по формуле

$$\overline{\text{TP}} = \frac{1}{K}\sum_{k=1}^{K}\text{TP}_k.$$

При макро-усреднении сначала вычисляется итоговая метрика для каждого класса, а затем результаты усредняются по всем классам. Например, точность будет вычислена как

$$\text{precision}(a, X) = \frac{1}{K}\sum_{k=1}^{K}\text{precision}_k(a, X); \qquad \text{precision}_k(a, X) = \frac{\text{TP}_k}{\text{TP}_k + \text{FP}_k}.$$

Если какой-то класс имеет очень маленькую мощность, то при микро-усреднении он практически никак не будет влиять на результат, поскольку его вклад в средние TP, FP, FN и TN будет незначителен. В случае же с макро-вариантом усреднение проводится для величин, которые уже не чувствительны к соотношению размеров классов (если мы используем, например, точность или полноту), и поэтому каждый класс внесет равный вклад в итоговую метрику.

3 AUC-ROC

Рассмотрим задачу бинарной классификации с метками классов $\mathbb{Y} = \{-1, +1\}$, и пусть задан некоторый алгоритм $b(x)$, позволяющий вычислять оценку принадлежности объекта x положительному классу. AUC-ROC позволяет оценивать качество классификации для семейства алгоритмов следующего вида:

$$a(x; t) = \begin{cases} -1, \; b(x) \le t, \\ +1, \; b(x) > t, \end{cases}$$

т.е. алгоритмов, присваивающих метки объектам в соответствии с оценками $b(x)$, отсекая их по некоторому порогу t. Каждый алгоритм (получающийся при фиксации значения порога t) представляется точкой на плоскости (FPR, TPR), где

$$\text{FPR} = \frac{\text{FP}}{\text{FP} + \text{TN}} = \frac{\text{FP}}{\ell_-},$$

$$\text{TPR} = \frac{\text{TP}}{\text{TP} + \text{FN}} = \frac{\text{TP}}{\ell_+},$$

ℓ_-, ℓ_+ — количество объектов отрицательного и положительного классов соответственно. AUC-ROC, в свою очередь, является площадью под получившейся кривой.

Изучим подробнее некоторые важные свойства данной метрики.

Критерий AUC-ROC имеет большое число интерпретаций — например, он равен вероятности того, что случайно выбранный положительный объект окажется позже случайно выбранного отрицательного объекта в ранжированном списке, порожденном $b(x)$. Разберем подробнее немного другую формулировку.

Задача 3.1. *В ранжировании часто используется функционал «доля дефектных пар». Его можно определить и для задачи бинарной классификации.*

Пусть дан классификатор $b(x)$, который возвращает оценки принадлежности объектов классу $+1$, и пусть все значения $b(x_i)$, $i = \overline{1,\ell}$, для некоторой выборки $X = \{(x_i, y_i)\}_{i=1}^{\ell}$ различны. Отсортируем все объекты по возрастанию ответа классификатора: $b(x_{(1)}) < \cdots < b(x_{(\ell)})$. Обозначим истинные ответы на этих объектах через $y_{(1)}, \ldots, y_{(\ell)}$. Тогда доля дефектных пар записывается как

$$DP(b, X) = \frac{2}{\ell(\ell-1)} \sum_{i<j} [y_{(i)} > y_{(j)}].$$

Как данный функционал связан с AUC-ROC?

Решение. Для начала разберем процедуру построения ROC-кривой. Сперва все объекты сортируются по неубыванию оценки $b(x)$, тем самым формируя список $x_{(1)}, \ldots, x_{(\ell)}$. Заметим, что для построения ROC-кривой достаточно рассмотреть $(\ell + 1)$ различных значений порога t, соответствующих всем различным способам классификации выборки, порожденным алгоритмом $b(x)$, — например, в качестве таких порогов можно рассмотреть следующий набор:

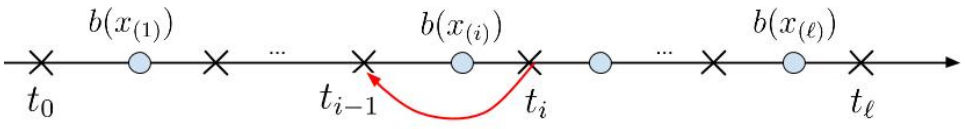
$$\begin{aligned} t_\ell &= b(x_{(\ell)}) + 1, \\ t_i &= \frac{b(x_{(i)}) + b(x_{(i+1)})}{2},\ i = \overline{1,\ell-1}, \\ t_0 &= b(x_{(1)}) - 1. \end{aligned}$$

Зафиксируем значение порога $t = t_\ell = b(x_{(\ell)}) + 1$, в этом случае имеем

$$\text{FPR} = \frac{\text{FP}}{\ell_-} = \frac{0}{\ell_-} = 0,$$

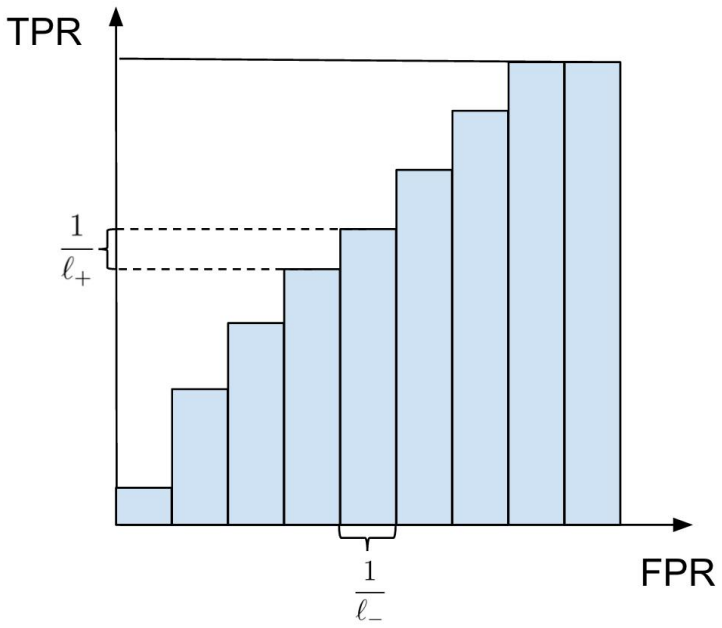
$$\text{TPR} = \frac{\text{TP}}{\ell_+} = \frac{0}{\ell_+} = 0.$$

Таким образом, алгоритму $a(x; t_\ell)$ соответствует точка (0; 0) на плоскости, откуда начинается построение ROC-кривой. Будем перебирать пороги в порядке невозрастания их значения, начиная с t_ℓ. Пусть мы хотим уменьшить значение порога с t_i до t_{i-1}. При этом классификация объекта $x_{(i)}$ (и только его) изменится с отрицательной на положительную. Рассмотрим 2 случая.



1. $y_{(i)} = +1$. В этом случае классификатор начнет верно классифицировать объект, на котором ранее допускал ошибку, при этом FPR не изменится, а TPR повысится на $\frac{1}{\ell_+}$.

2. $y_{(i)} = -1$. В этом случае классификатор начнет ошибаться на объекте, который ранее классифицировал верно, при этом TPR не изменится, а FPR повысится на $\frac{1}{\ell_-}$.



Теперь рассмотрим, как при этом изменяется AUC-ROC. Заметим, что область под ROC-кривой состоит из непересекающихся прямоугольников, каждый из которых снизу ограничен осью FPR, а сверху — одним из горизонтальных отрезков, соответствующих второму из рассмотренных случаев. Поэтому каждый раз, когда имеет место второй случай, к текущей накопленной площади под кривой (которая изначально в точке (0; 0) равна 0) добавляется площадь прямоугольника, горизонтальные стороны которого равны $\frac{1}{\ell_-}$, а вертикальные равны $\frac{1}{\ell_+}\sum_{j=i+1}^{\ell}[y_{(j)} = +1]$ (доля уже рассмотренных положительных объектов среди всех положительных), поэтому в этом случае текущее значение AUC-ROC увеличивается на $\frac{1}{\ell_-\ell_+}\sum_{j=i+1}^{\ell}[y_{(j)} = +1]$. Итого, финальное значение AUC-ROC можно посчитать следующим образом:

$$\begin{aligned}
\text{AUC} &= \frac{1}{\ell_+\ell_-}\sum_{i=1}^{\ell}[y_{(i)} = -1]\sum_{j=i+1}^{\ell}[y_{(j)} = +1] = \\
&= \frac{1}{\ell_+\ell_-}\sum_{i=1}^{\ell}\sum_{j=i+1}^{\ell}[y_{(i)} < y_{(j)}] = \\
&= \frac{1}{\ell_+\ell_-}\sum_{i<j}^{\ell}(1 - [y_{(i)} = y_{(j)}] - [y_{(i)} > y_{(j)}]5) = \\
&= \frac{1}{\ell_+\ell_-}\sum_{i<j}^{\ell}(1 - [y_{(i)} = y_{(j)}]) - \frac{1}{\ell_+\ell_-}\sum_{i<j}^{\ell}[y_{(i)} > y_{(j)}] = \\
&= \frac{1}{\ell_+\ell_-}\sum_{i<j}^{\ell}([y_{(i)} \neq y_{(j)}]) - \frac{1}{\ell_+\ell_-}\sum_{i<j}^{\ell}[y_{(i)} > y_{(j)}] = \\
&= \frac{\ell_+\ell_-}{\ell_+\ell_-} - \frac{1}{\ell_+\ell_-}\sum_{i<j}^{\ell}[y_{(i)} > y_{(j)}] = 1 - \frac{1}{\ell_+\ell_-}\sum_{i<j}[y_{(i)} > y_{(j)}].
\end{aligned}$$

Отсюда получаем, что AUC-ROC и доля дефектных пар связаны следующим соотношением:

$$\mathrm{DP}(b, X) = \frac{2\ell_- \ell_+}{\ell(\ell-1)}(1 - \mathrm{AUC}(b, X))).$$

■

Заметим, что в случае, когда несколько объектов выборки имеют равные значения $b(x)$, при уменьшении значения порога с $t_i > b(x)$ до $t_{i-1} < b(x)$, где x — один из таких объектов, изменение значений FPR и TPR происходит одновременно, поэтому соответствующий участок ROC-кривой будет наклонным, а не горизонтальным или вертикальным.

Задача 3.2. *Пусть даны выборка X, состоящая из 5 объектов, и классификатор $b(x)$, предсказывающий оценку принадлежности объекта положительному классу. Предсказания $b(x)$ и реальные метки объектов приведены ниже:*

$$\begin{aligned}
&b(x_1) = 0.2, \quad y_1 = -1,\\
&b(x_2) = 0.4, \quad y_2 = +1,\\
&b(x_3) = 0.1, \quad y_3 = -1,\\
&b(x_4) = 0.7, \quad y_4 = +1,\\
&b(x_5) = 0.05, \quad y_5 = +1.
\end{aligned}$$

Вычислите AUC-ROC для множества классификаторов $a(x;t)$, порожденного $b(x)$, на выборке X.

Решение. В соответствии с процессом построения ROC-кривой, описанным в предыдущей задаче, отсортируем оценки $b(x_i)$ в порядке их неубывания: $(b(x_{(i)}))_{i=1}^{\ell} = (0.05, 0.1, 0.2, 0.4, 0.7)$. Также составим последовательность реальных меток объектов из этого упорядоченного списка: $(y_{(i)})_{i=1}^{\ell} = (+1, -1, -1, +1, +1)$.

Построим ROC-кривую (см. рис. 1), откуда AUC-ROC $= \frac{2}{3}$.

■

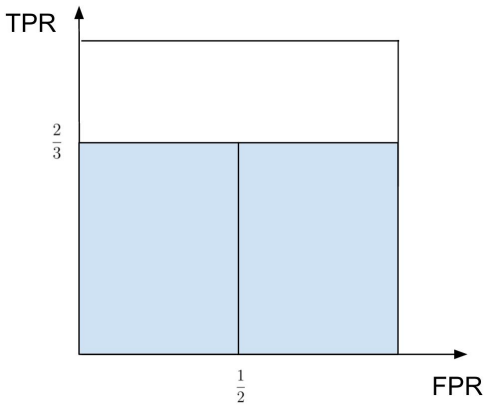


Рис. 1. Иллюстрация к задаче 1.2.

Заметим, что при вычислении AUC-ROC на некоторой выборке X для итогового классификатора $a(x;t)$ важны не конкретные значения $b(x_i)$, $i = \overline{1,\ell}$, а порядок расположения объектов в отсортированном по неубыванию списке $b(x_{(1)}), \ldots, b(x_{(\ell)})$, порожденным алгоритмом $b(x)$. Таким образом, для фиксированной выборки X алгоритм $b(x)$ задаёт перестановку на её объектах, которая в дальнейшем используется при расчёте AUC-ROC.

Задача 3.3. *Пусть $b(x)$ — классификатор, предсказывающий оценку принадлежности объекта x классу $+1$ таким образом, что для некоторой выборки X он равновероятно выдаёт на её объектах одну из всех возможных перестановок. Чему равно матожидание AUC-ROC этого классификатора?*

Решение. Как было показано в задаче 1.1, для AUC-ROC верно

$$\text{AUC} = \frac{1}{\ell_+\ell_-}\sum_{i<j}^{\ell}[y_{(i)} = -1][y_{(j)} = +1],$$

поэтому

$$\mathbb{E}\text{AUC} = \frac{1}{\ell_+\ell_-}\sum_{i<j}^{\ell}\mathbb{E}([y_{(i)} = -1][y_{(j)} = +1]).$$

Заметим, что величина $[y_{(i)} = -1][y_{(j)} = +1]$ принимает значения 0 и 1, поэтому

$$\mathbb{E}([y_{(i)} = -1][y_{(j)} = +1]) = \mathbb{P}(y_{(i)} = -1,\, y_{(j)} = +1) = \frac{\ell_-\ell_+(\ell-2)!}{\ell!} = \frac{\ell_-\ell_+}{\ell(\ell-1)}.$$

Отсюда имеем

$$\mathbb{E}AUC = \frac{1}{\ell_-\ell_+}\sum_{i<j}^{\ell}\frac{\ell_-\ell_+}{\ell(\ell-1)} = \frac{\ell(\ell-1)}{2}\frac{1}{\ell(\ell-1)} = \frac{1}{2}.$$

■

Итого, можем заметить, что значение AUC-ROC, близкое к $\frac{1}{2}$, означает, что классификатор близок к случайному, тогда как значение, равное 1, означает, что классификатор безошибочно классифицирует объекты при некотором значении порога.

Задача 3.4. *Пусть $b(x)$ — некоторый классификатор, предсказывающий оценку принадлежности объекта x положительному классу, и при этом AUC-ROC множества классификаторов $a(x;t)$, порожденных $b(x)$, на некоторой выборке X принимает значение, меньшее 0.5. Как можно скорректировать прогнозы классификаторов $a(x;t)$, чтобы они были более осмысленными по сравнению с прогнозами классификатора, выдающего случайные ответы?*

Решение.

Для некоторого классификатора $a(x;t)$ рассмотрим классификатор $a^*(x;t)$, выдающий противоположные метки по сравнению с $a(x;t)$, т.е.:

$a^*(x;t) = -a(x;t)$.

При этом TP и FP на обучающей выборке для некоторого классификатора $a^*(x;t)$ будут принимать следующие значения:

$$\mathrm{TP}(a^*(x;t),X) = \sum_{i=1}^{\ell}[y_i=+1][a^*(x;t)=+1] = \\ = \sum_{i=1}^{\ell}[y_i=+1][a(x;t)=-1] = \mathrm{FN}(a(x;t),X),$$

$$\mathrm{FP}(a^*(x;t),X) = \sum_{i=1}^{\ell}[y_i=-1][a^*(x;t)=+1] = \\ = \sum_{i=1}^{\ell}[y_i=-1][a(x;t)=-1] = \mathrm{TN}(a(x;t),X).$$

Отсюда имеем

$$\mathrm{TPR}(a^*(x;t),X) = \frac{\mathrm{TP}(a^*(x;t),X)}{\ell_+} = \frac{\mathrm{FN}(a(x;t),X)}{\ell_+} = \\ = \frac{\ell_+ - \mathrm{TP}(a(x;t),X)}{\ell_+} = 1 - \mathrm{TPR}(a(x;t),X),$$

$$\mathrm{FPR}(a^*(x;t),X) = \frac{\mathrm{FP}(a^*(x;t),X)}{\ell_-} = \frac{\mathrm{TN}(a(x;t),X)}{\ell_-} = \\ = \frac{\ell_- - \mathrm{FP}(a(x;t),X)}{\ell_-} = 1 - \mathrm{FPR}(a(x;t),X),$$

поэтому классификатор $a^*(x;t)$ будет представлен на плоскости точкой, симметричной точке, отвечающей классификатору $a(x;t)$, относительно точки $(0.5;0.5)$.

Рассмотрим ROC-кривую для множества классификаторов $a(x;t)$. Пусть площадь областей единичного квадрата, находящихся между его диагональю и частями ROC-кривой, расположенных под ней, равна S_-, а между диагональю и частями ROC-кривой, расположенных над диагональю, — S_+. Тогда AUC-ROC для такой кривой принимает значение $0.5 + S_+ - S_- < 0.5$ (по условию), отсюда $S_+ - S_- < 0$.

Как было показано ранее, ROC-кривая для множества классификаторов $a^*(x;t)$ симметрична ROC-кривой для множества классификаторов $a(x;t)$, а потому для первой кривой область, соответствующая площади S_-, будет расположена над диагональю единичного квадрата, площади S_+ — под диагональю. Отсюда AUC-ROC для множества классификаторов $a^*(x;t)$ будет принимать значение $0.5 - S_+ + S_- > 0.5$, а потому прогнозы классификаторов из этого множества более осмысленны по сравнению со случайным классификатором.

■

§3.1 Прямая оптимизация AUC-ROC

При обучении модели в бинарной классификации чаще всего решается задача минимизации верхней оценки функционала ошибки:

$$Q(a,X) = \frac{1}{\ell}\sum_{i=1}^{\ell}[a(x_i)\neq y_i] \leq \frac{1}{\ell}\sum_{i=1}^{\ell}\tilde{L}(M_i) \to \min_w$$

Однако иногда возникает необходимость оптимизировать более сложные метрики — в частности, AUC-ROC. Напрямую оптимизировать подобные метрики не представляется возможным из-за их дискретной структуры, однако мы можем использовать трюк с верхней оценкой функционала ошибки и в этом случае. В задаче 1.1 мы показали, что AUC-ROC связан с долей дефектных пар в выборке, поэтому максимизация AUC-ROC равносильна минимизации доли дефектных пар.

$$DP(b, X) = \frac{2}{\ell(\ell-1)} \sum_{i<j}^{\ell} [y_i < y_j][b(x_i) > b(x_j)] =$$

$$\frac{2}{\ell(\ell-1)} \sum_{i<j}^{\ell} [y_i < y_j][b(x_j) - b(x_i) < 0] \leq \frac{2}{\ell(\ell-1)} \sum_{i<j}^{\ell} [y_i < y_j] \tilde{L}(b(x_j) - b(x_i)) \to \min_b$$

Список литературы

[1] *Davis J., Goadrich M.* (2006). The Relationship Between Precision-Recall and ROC Curves. // Proceedings of the 23rd International Conference on Machine Learning, Pittsburgh, PA.