Урок 4

Метрики качества

4.1. Метрики качества в задачах регрессии

4.1.1. Применение метрик качества в машинном обучении

Метрики качества могут использоваться:

- Для задания функционала ошибки (используется при обучении).
- Для подбора гиперпараметров (используется при измерении качества на кросс-валидации). В том числе можно использовать другую метрику, которая отличается от метрики, с помощью которой построен функционал ошибки.
- Для оценивания итоговой модели: пригодна ли модель для решения задачи.

Далее мы рассмотрим, какие метрики можно использовать в задачах регрессии.

4.1.2. Среднеквадратичная ошибка

Первая метрика, о которой уже шла речь — среднеквадратичная ошибка:

$$MSE(a, X) = \frac{1}{\ell} \sum_{i=1}^{\ell} (a(x_i) - y_i)^2 .$$

Такой функционал легко оптимизировать, используя, например, метод градиентного спуска.

Этот функционал сильно штрафует за большие ошибки, так как отклонения возводятся в квадрат. Это приводит к тому, что штраф на выбросе будет очень сильным, и алгоритм будет настраиваться на выбросы. Другими словами, алгоритм будет настраиваться на такие объекты, на которые не имеет смысл настраиваться.

4.1.3. Средняя абсолютная ошибка

Похожий на предыдущий функционал качества — средняя абсолютная ошибка:

$$MAE(a, X) = \frac{1}{\ell} \sum_{i=1}^{\ell} |a(x_i) - y_i| .$$

Этот функционал сложнее минимизировать, так как у модуля производная не существует в нуле. Но у такого функционала больше устойчивость к выбросам, так как штраф за сильное отклонение гораздо меньше.

4.1.4. Коэффициент детерминации

Коэффициент детерминации $R^2(a, X)$:

$$R^2(a, X) = 1 - \frac{\sum_{i=1}^{\ell} \big(a(x_i) - y_i\big)^2}{\sum_{i=1}^{\ell} (y_i - \bar{y})}, \qquad \bar{y} = \frac{1}{\ell} \sum_{i=1}^{\ell} y_i,$$

позволяет интерпретировать значение среднеквадратичной ошибки. Этот коэффициент показывает, какую долю дисперсии (разнообразия ответов) во всем целевом векторе y модель смогла объяснить.

Для разумных моделей коэффициент детерминации лежит в следующих пределах:

$$0 \leq R^2 \leq 1,$$

причем случай $R^2 = 1$ соответствует случаю идеальной модели, $R^2 = 0$ — модели на уровне оптимальной «константной», а $R^2 < 0$ — модели хуже «константной» (такие алгоритмы никогда не нужно рассматривать). Оптимальным константым алгоритмом называется такой алгоритм, который возвращает всегда среднее значение ответов $\bar{y}$ для объектов обучающей выборки.

4.1.5. Несимметричные потери

До этого рассматривались симметричные модели, то есть такие, которые штрафуют как за недопрогноз, так и за перепрогноз. Но существуют такие задачи, в которых эти ошибки имеют разную цену.

Пусть, например, требуется оценить спрос на ноутбуки. В этом случае заниженный прогноз приведет к потере лояльности покупателей и потенциальной прибыли (будет закуплено недостаточное количество ноутбуков), а завышенный — только к не очень большим дополнительным расходам на хранение непроданных ноутбуков. Чтобы учесть это, функция потерь должна быть несимметричной и сильнее штрафовать за недопрогноз, чем за перепрогноз.

4.1.6. Квантильная ошибка

В таких случаях хорошо подходит квантильная ошибка или квантильная функция потерь:

$$\rho_\tau(a, X) = \frac{1}{\ell} \sum_{i=1}^{\ell} \Big((\tau - 1)\big[y_i < a(x_i)\big] + \tau\big[y_i \geq a(x_i)\big] \Big) \left(y_i - a(x_i)\right).$$

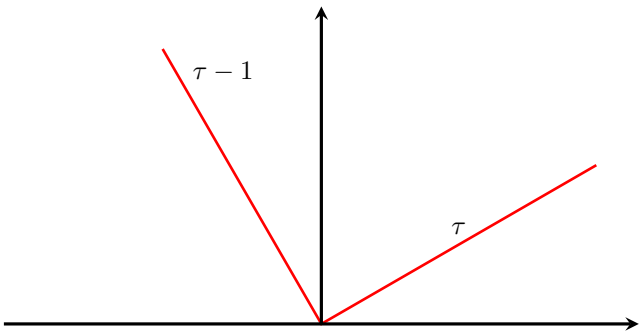

Рис. 4.1: График квантильной функции потерь

Параметр $\tau \in [0, 1]$ определяет то, за что нужно штрафовать сильнее — за недопрогноз или перепрогноз. Если τ ближе к 1, штраф будет больше за недопрогноз, а если, наоборот, ближе к 0 — за перепрогноз.

4.1.7. Вероятностный смысл квантильной ошибки

Чтобы разобраться, почему такая функция потерь называется квантильной, нужно разобраться с ее вероятностным смыслом. Пусть один и тот же объект x с одним и тем же признаковым описанием повторяется в выборке n раз, но на каждом из повторов — свой ответ $y_1, ..., y_n$.

Такое может возникнуть при измерении роста человека. Измерения роста одного и того же человека могут отличаться ввиду ошибки прибора, а также зависеть от самого человека (может сгорбиться или выпрямиться).

При этом алгоритм должен для одного и того же признакового описания возвращать одинаковый прогноз. Другими словами, необходимо решить, какой прогноз оптимален для x с точки зрения различных функционалов ошибки.

Оказывается, что если используется квадратичный функционал ошибки, то наиболее оптимальным прогнозом будет средний ответ на объектах, если абсолютный, то медиана ответов. Если же будет использоваться квантильная функция потерь, наиболее оптимальным прогнозом, будет τ-квантиль. В этом и состоит вероятностный смысл квантильной ошибки.

4.2. Метрика качества классификации

В этом блоке речь пойдет о том, как измерять качество в задачах классификации.

4.2.1. Доля правильных ответов

Как меру качества в задачах классификации естественно использовать долю неправильных ответов:

$$\frac{1}{\ell}\sum_{i=1}^{\ell}[a(x_i) \neq y_i]$$

Однако в задачах классификации принято выбирать метрики таким образом, чтобы их нужно было максимизировать, тогда как в задачах регрессии — так, чтобы их нужно было минимизировать. Поэтому определяют:

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

Эта метрика качества проста и широко используется, однако имеет несколько существенных недостатков.

4.2.2. Несбалансированные выборки

Первая проблема связана с несбалансированными выборками. Показателен следующий пример. Пусть в выборке 1000 объектов, из которых 950 относятся к классу -1 и 50 — к классу $+1$. Рассматривается бесполезный (поскольку не восстанавливает никаких закономерностей в данных) константный классификатор, который на всех объектах возвращает ответ -1. Но доля правильных ответов на этих данных будет равна 0.95, что несколько много для бесполезного классификатора.

Чтобы «бороться» с этой проблемой, используется следующий факт. Пусть q_0 — доля объектов самого крупного класса, тогда доля правильных ответов для разумных алгоритмов $accuracy \in [q_0, 1]$, а не $[1/2, 1]$, как это можно было бы ожидать. Поэтому, если получается высокий процент правильных ответов, это может быть связано не с тем, что построен хороший классификатор, а с тем, что какого-то класса сильно больше, чем остальных.

4.2.3. Цены ошибок

Вторая проблема с долей верных ответов состоит в том, что она никак не учитывает разные цены разных типов ошибок. Тогда как цены действительно могут быть разными.

Например, в задаче кредитного скоринга, то есть в задаче принятия решения относительно выдачи кредита, сравниваются две модели. При использовании первой модели кредит будет выдан 100 клиентам, 80 из которых его вернут. Во второй модели, более консервативной, кредит был выдан только 50 клиентам, причем вернули его в 48 случаях. То, какая из двух моделей лучше, зависит от того, цена какой из ошибок выше: не дать кредит клиенту, который мог бы его вернуть, или выдать кредит клиенту, который его не вернет. Таким образом, нужны дополнительные метрики качества, которые учитывают цены той или иной ошибки.

4.3. Точность и полнота

4.3.1. Цены ошибок

В этом разделе пойдет речь о метриках качества классификации, которые позволяют учитывать разные цены ошибок. В конце предыдущего разделе уже было сказано, что цены ошибок действительно могут быть разными. Так, в задаче банковского скоринга необходимо принять решение, что хуже: выдать кредит «плохому» клиенту или не выдать кредит «хорошему» клиенту. Доля верных ответов не способна учитывать цены разных ошибок и поэтому не может дать ответа на этот вопрос.

4.3.2. Матрица ошибок

Удобно классифицировать различные случаи, как соотносятся между собой результат работы алгоритма и истинный ответ, с помощью так называемой матрицы ошибок.

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

Когда алгоритм относит объект к классу $+1$, говорят, что алгоритм срабатывает. Если алгоритм сработал и объект действительно относится к классу $+1$, имеет место верное срабатывание (true positive), а если объект на самом деле относится к классу -1, имеет место ложное срабатывание (false positive).

Если алгоритм дает ответ -1, говорят, что он пропускает объект. Если имеет место пропуск объекта класса $+1$, то это ложный пропуск (false negative). Если же алгоритм пропускает объект класса -1, имеет место истинный пропуск (true negative).

Таким образом, существуют два вида ошибок: ложные срабатывания и ложные пропуски. Для каждого из них нужна своя метрика качества, чтобы измерить, какое количество ошибок какого типа совершается.

4.3.3. Точность и полнота

Пусть для примера рассматриваются две модели $a_1(x)$ и $a_2(x)$. Выборка состоит из 200 объектов, из которых 100 относятся к классу 1 и 100 — к классу -1. Матрицы ошибок имеют вид:

	$y = 1$	$y = -1$
$a_1(x) = 1$	80	20
$a_1(x) = -1$	20	80

	$y = 1$	$y = -1$
$a_2(x) = 1$	48	2
$a_2(x) = -1$	52	98

Введем две метрики. Первая метрика, точность (precision), показывает, насколько можно доверять классификатору в случае срабатывания:

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

Вторая метрика, полнота (recall), показывает, на какой доле истинных объектов первого класса алгоритм срабатывает:

$$recall(a, X) = \frac{TP}{TP + FN}.$$

В примере выше точность и полнота первого алгоритма оказываются равными:

$$\begin{aligned} precision(a_1, X) &= 0.8, & precision(a_2, X) &= 0.96 \\ recall(a_1, X) &= 0.8, & recall(a_2, X) &= 0.48 \end{aligned}$$

Вторая модель является очень точной, но в ущерб полноте.

4.3.4. Примеры использования точности и полноты

Первый пример — использование в задаче кредитного скоринга. Пусть в задаче кредитного скоринга ставится условие, что неудачных кредитов должно быть не больше 5%. В таком случае задача является задачей максимизации полноты при условии $\text{precision}(a, X) \geq 0.95$.

Второй пример — использование в медицинской диагностике. Необходимо построить модель, которая определяет, есть или нет определенное заболевание у пациента. При этом требуется, чтобы были выявлены как минимум 80% пациентов, которые действительно имеют данное заболевание. Тогда ставят задачу максимизации точности при условии $\text{recall}(a, X) \geq 0.8$.

Следует особо обратить внимание на то, как точность и полнота работают в случае несбалансированных выборок. Пусть рассматривается выборка со следующей матрицей ошибок:

	$y = 1$	$y = -1$
$a(x) = 1$	10	20
$a(x) = -1$	90	10000

Доля верных ответов (accuracy), точность(precision) и полнота(recall) для данного случая:

$$accuracy(a, X) = 0.99, \qquad precision(a, X) = 0.33, \qquad recall(a, X) = 0.1.$$

То, что доля верных ответов равняется 0.99, ни о чем не говорит: алгоритм все равно делает 66% ложных срабатываний и выявляет только 10% положительных случаев. Благодаря введению точности и полноты становится понятно, что алгоритм нужно улучшать.

4.4. Объединение точности и полноты

В этом разделе пойдет речь о том, как соединить точность и полноту в одну метрику качества классификации.

4.4.1. Точность и полнота (напоминание)

Точность показывает, насколько можно доверять классификатору в случае срабатывания:

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

Полнота показывает, на какой доле истинных объектов первого класса алгоритм срабатывает:

$$recall(a, X) = \frac{TP}{TP + FN}.$$

В некоторых задачах есть ограничения на одну из этих метрик, тогда как по второй метрике будет производиться оптимизация. Но в некоторых случаях хочется максимизировать и точность, и полноту одновременно. Встает вопрос об объединении этих двух метрик.

4.4.2. Арифметическое среднее

Единая метрика может быть получена как арифметическое среднее точности и полноты:

$$A = \frac{1}{2}(precision + recall)$$

Пусть есть алгоритм, точность которого равна 10%, а полнота — 100%:

$$precision = 0.1 \qquad recall = 1.$$

Это может быть случай, когда в выборке всего 10% объектов класса +1, а алгоритм является константным и всегда возвращает +1. Очевидно, что этот алгоритм плохой, но введенная выше метрика для него равна $A = 0.55$. В свою очередь другой, гораздо более лучший алгоритм, с $precision = 0.55$ и $recall = 55$ также характеризуется $A = 0.55$.

Ситуация, когда константный и разумный алгоритмы могут лежать на одной линии, является недопустимой, поэтому следует искать другой способ построения единой метрики.

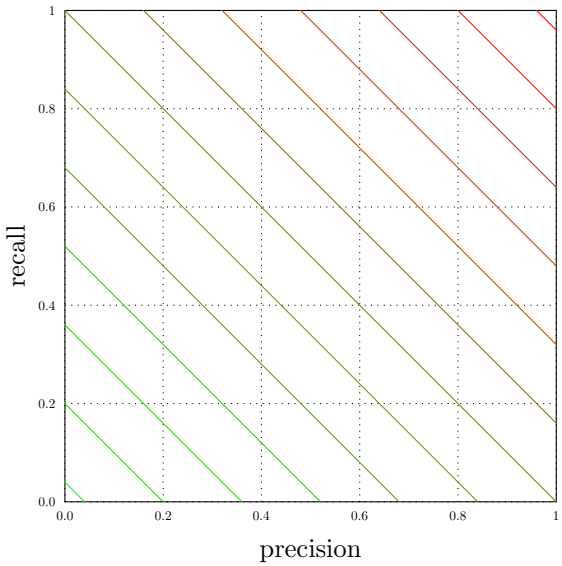

Рис. 4.2: Линии $A = \frac{1}{2}(precision + recall) = const$ в координатах precision–recall

4.4.3. Минимум

Чтобы констрантный и разумный алгоритмы не лежали на одной линии уровня, можно рассматривать:

$$M = \min(precision, recall).$$

Данный подход решает вышеупомянутую проблему, например:

$$precision = 0.05, \quad recall = 1 \qquad \Longrightarrow \qquad M = 0.05.$$

Но есть другой нюанс: два алгоритма, для которых точности одинаковы, но отличаются значения полноты, будут лежать на одной линии уровня M:

$$\begin{aligned} precision = 0.4, \quad recall = 0.5 \qquad &\Longrightarrow \qquad M = 0.4, \\ precision = 0.4, \quad recall = 0.9 \qquad &\Longrightarrow \qquad M = 0.4. \end{aligned}$$

Такое тоже недопустимо, так как второй алгоритм существенно лучше первого.

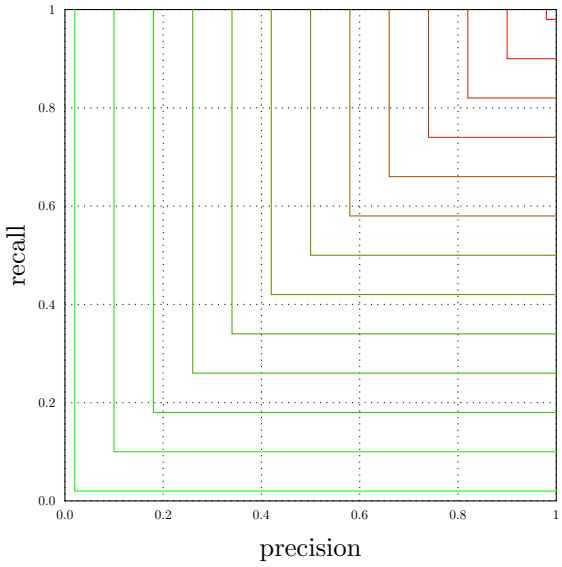

Рис. 4.3: Линии $M = \min(precision, recall) = const$ в координатах precision–recall

4.4.4. F-мера

«Сгладить» минимум можно с помощью гармонического среднего, или F-меры:

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

Для двух упомянутых выше алгоритма значения F-меры, в отличие от M, будут отличаться:

$$precision = 0.4, \quad recall = 0.5 \quad \Longrightarrow \quad F = 0.44,$$
$$precision = 0.4, \quad recall = 0.9 \quad \Longrightarrow \quad F = 0.55.$$

Если необходимо отдать предпочтение точности или полноте, следует использовать расширенную F-меру, в которой есть параметр β:

$$F = (1 + \beta^2)\frac{precision \cdot recall}{\beta^2 \cdot precision + recall}.$$

Например, при $\beta = 0.5$ важнее оказывается полнота, а в случае $\beta = 2$, наоборот, важнее оказывается точность.

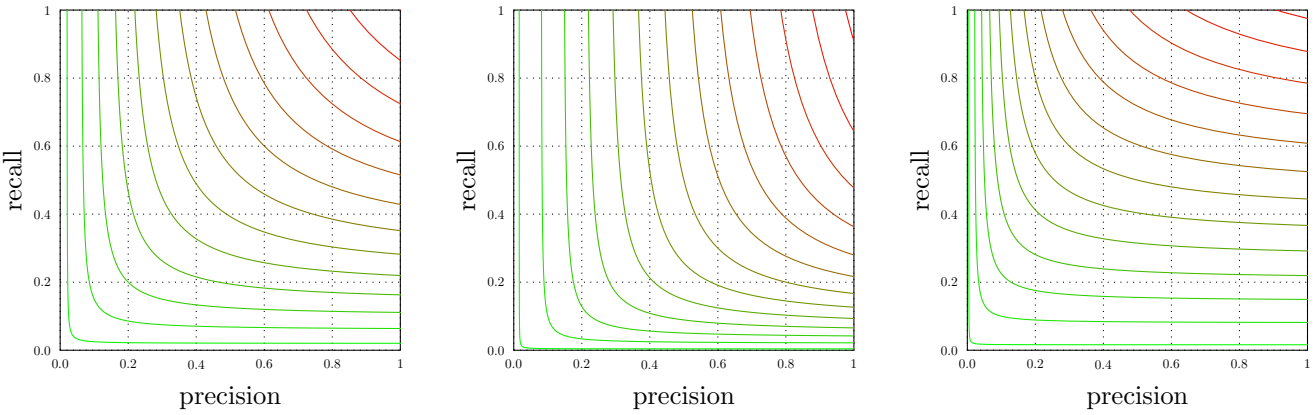

Рис. 4.4: Линии $F = const$ в координатах precision–recall при значениях $\beta = 1$, $\beta = 0.5$ и $\beta = 2$ соответственно

4.5. Качество оценок принадлежности классу

4.5.1. Оценка принадлежности

Многие алгоритмы бинарной классификации устроены следующим образом: сначала вычисляется некоторое вещественное число $b(x)$, которое сравнивается с порогом t.

$$a(x) = [b(x) > t],$$

где $b(x)$ — оценка принадлежности классу $+1$. Другими словами, $b(x)$ выступает в роли некоторой оценки уверенности, что x принадлежит классу $+1$.

В случае линейного классификатора $a(x) = [\langle w, x\rangle > t]$ оценка принадлежности классу $+1$ имеет вид $b(x) = \langle w, x\rangle$.

Часто бывает необходимо оценить качество именно оценки принадлежности, а порог выбирается позже из соображений на точность или полноту.

4.5.2. Оценка принадлежности в задаче кредитного скоринга

Пусть рассматривается задачного кредитного скоринга и была построена некоторая функция $b(x)$, которая оценивает вероятность возврата кредита клиентом x. Далее классификатор строится следующим образом:

$$a(x) = [b(x) > 0.5]$$

При этом получилось, что точность (precision) равна 10%, а полнота (recall) — 70%. Это очень плохой алгоритм, так как 90% клиентов, которым будет выдан кредит, не вернут его.

При этом не понятно, в чем дело: был плохо выбран порог или алгоритм не подходит для решения данной задачи. Именно для этого необходимо измерять качество самих оценок $b(x)$.

4.5.3. PR-кривая

Первый способ оценки принадлежности классу основан на использовании кривой точности-полноты. По оси X откладывается полнота, а по оси Y — точность. Каждой точке на этой кривой будет соответствовать классификатор с некоторым значением порога.

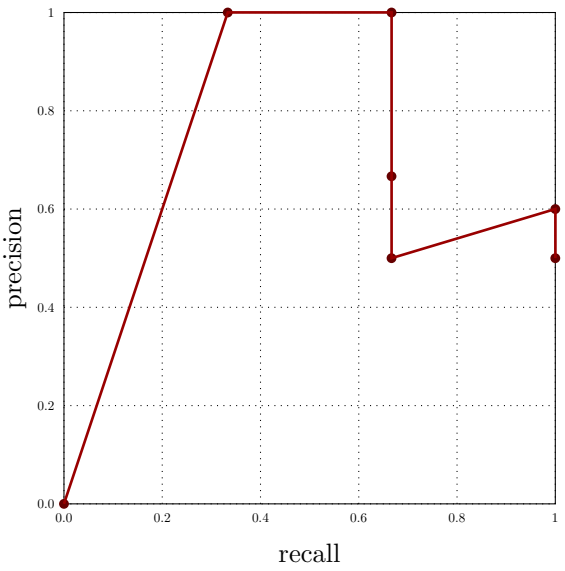

Рис. 4.5: Кривая полноты–точности

Для примера будет приведено построение PR-кривой для выборки из 6 объектов, три из которых относятся к классу 1 и 3 — к классу 0. Соответствующий ей график изображен выше.

b(x)	0.14	0.23	0.39	0.54	0.73	0.90
y	0	1	0	0	1	1

1. При достаточно большом пороге ни один объект не будет отнесен к классу 1. В этом случае и точность и полнота равны 0.
2. При таком пороге, что ровно один объект отнесен к классу 1, точность будет 100% (поскольку этот объект действительно из 1 класса), а полнота — 1/3 (поскольку всего 3 объекта 1 класса).
3. При дальнейшем уменьшении порога уже два объекта отнесены к классу 1, точность также остается 100%, а полнота становится равной 2/3.
4. При таком пороге, что уже три объекта будут отнесены к классу 1, точность становится равной 2/3, а полнота остается такой же.
5. При таком пороге, что четыре объекта отнесены к классу 1, точность уменьшится до 0.5, а полнота опять не изменится.
6. При дальнейшем уменьшении порога уже 5 объектов будут отнесены к 1 классу, полнота станет равной 100%, а точность — 3/5.

В реальных задачах с числом объектов порядка нескольких тысяч или десятков тысяч, кривая точности-полноты выглядит примерно следующим образом.

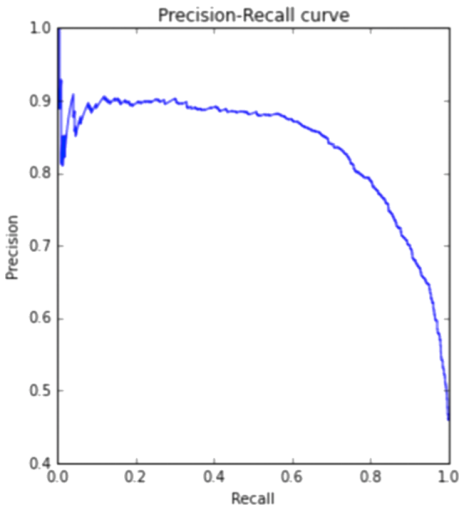

Рис. 4.6: Кривая точности-полноты в реальных задачах с десятками тысяч объектов

Следует отметить, что начинается PR-кривая всегда из точки $(0, 0)$, а заканчивается точной $(1, r)$, где r — доля объектов класса 1.

В случае идеального классификатора, то есть если существует такой порог, что и точность, и полнота равны 100%, кривая будет проходить через точку $(1, 1)$. Таким образом, чем ближе кривая пройдет к этой точке, тем лучше оценки. Площадь под этой кривой может быть хорошей мерой качества оценок принадлежности к классу 1. Такая метрика называется AUC–PRC, или площадь под PR-кривой.

4.5.4. ROC-кривая

Второй способ измерить качество оценок принадлежности к классу 1 — ROC-кривая, которая строится в осях False Positive Rate (ось X) и True Positive Rate (ось Y):

$$FPR = \frac{FP}{FP + TN}, \qquad TPR = \frac{TP}{TP + FN}.$$

ROC-кривая строится аналогично PR-кривой: постепенно рассматриваются случаи различных значений порогов и отмечаются точки на графике. Для упомянутой выше выборки ROC-кривая имеет следующий вид:

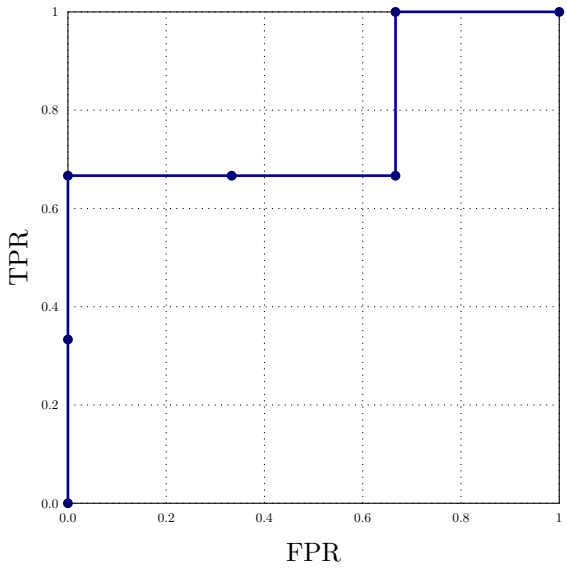

Рис. 4.7: ROC–кривая

В случае с большой выборкой ROC-кривая выглядит следующим образом:

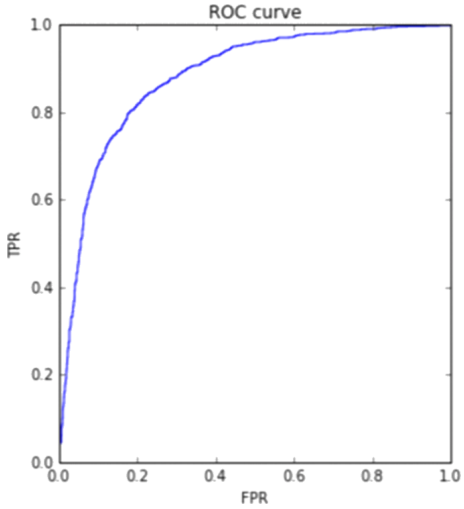

Рис. 4.8: Кривая ROC в реальных задачах с десятками тысяч объектов

Кривая стартует с точки $(0, 0)$ и приходит в точку $(1, 1)$. При этом, если существует идеальный классификатор, кривая должна пройти через точку $(0, 1)$. Чем ближе кривая к этой точке, тем лучше будут оценки, а площадь под кривой будет характеризовать качество оценок принадлежности к первому классу. Такая метрика называется AUC–ROC,или площадь под ROC-кривой.

4.5.5. Особенности AUC-ROC

Как было написано выше, ROC-кривая строится в осях FPR и TPR, которые нормируются на размеры классов:

$$FPR = \frac{FP}{FP + TN}, \qquad TPR = \frac{TP}{TP + FN}.$$

Следовательно, при изменении баланса классов величина AUC-ROC и неизменных свойствах объектов выборки площадь под ROC-кривой не изменится. В случае идеального алгоритма $AUC - ROC = 1$, а в случае худшего $AUC - ROC = \frac{1}{2}$.

Значение $AUC-ROC$ имеет смысл вероятности того, что если были выбраны случайный положительный и случайный отрицаельный объекты выборки, положительный объект получит оценку принадлежности выше, чем отрицательный объект.

4.5.6. Особенности AUC-PRC

PR-кривая строится в осях precision и recall:

$$precision = \frac{TP}{TP + FP}, \qquad recall = \frac{TP}{TP + FN},$$

а следовательно изменяется при изменении баланса классов.