Семинары по выбору моделей

Евгений Соколов
sokolov.evg@gmail.com

15 апреля 2015 г.

1 Выбор моделей и критерии качества

§1.1 Критерии качества в задаче регрессии

1.1.1 Стандартные функционалы

Наиболее типичными мерами качества в задачах регрессии являются *средняя квадратичная* (Mean Squared Error, MSE) и *средняя абсолютная* (Mean Absolute Error, MAE) ошибки:

$$\text{MSE} = \frac{1}{\ell}\sum_{i=1}^{\ell}(a(x_i) - y_i)^2;$$

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

Среднеквадратичный функционал сильнее штрафует за большие отклонения по сравнению со среднеабсолютным, и поэтому более чувствителен к выбросам. При использовании любого из этих двух функционалов может быть полезно проанализировать, какие объекты вносят наибольший вклад в общую ошибку — не исключено, что на этих объектах была допущена ошибка при вычислении признаков или целевой величины.

Среднеквадратичная ошибка подходит для сравнения двух моделей или для контроля качества во время обучения, но не позволяет сделать выводов о том, насколько хорошо данная модель решает задачу. Например, MSE = 10 является очень плохим показателем, если целевая переменная принимает значения от 0 до 1, и очень хорошим, если целевая переменная лежит в интервале $(10000, 100000)$. В таких ситуациях вместо среднеквадратичной ошибки полезно использовать *коэффициент детерминации*, или коэффициент R^2:

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

где $\bar{y} = \frac{1}{\ell}\sum_{i=1}^{\ell} y_i$ — среднее значение целевой переменной. Коэффициент детерминации измеряет долю дисперсии, объясненную моделью, в общей дисперсии целевой

переменной. Фактически, данная мера качества — это нормированная среднеквадратичная ошибка. Если она близка к единице, то модель хорошо объясняет данные, если же она близка к нулю, то прогнозы сопоставимы по качеству с константным предсказанием.

1.1.2 Квантильная регрессия

В некоторых задачах цены занижения и завышения прогнозов могут отличаться друг от друга. Например, при прогнозировании спроса на товары интернет-магазина гораздо опаснее заниженные предсказания, поскольку они могут привести к потере клиентов. Завышенные же прогнозы приводят лишь к издержкам на хранение товара на складе. Функционал в этом случае можно записать как

$$Q(a, X^\ell) = \sum_{i=1}^{\ell} \rho_\tau(y_i - a(x_i)),$$

где

$$\rho_\tau(z) = (\tau - 1)[z < 0]z + \tau[z \geqslant 0]z,$$

а параметр τ лежит на отрезке $[0, 1]$ и определяет соотношение важности занижения и завышения прогноза.

Попробуем понять, какой вероятностный смысл несет оптимизация такого функционала. Пусть имеется некоторое распределение $p(x, y)$ на декартовом произведении объектов и ответов $\mathbb{X} \times \mathbb{Y}$. Известно, что при оптимизации квадратичного функционала алгоритм $a(x)$ будет приближать условное матожидание ответа в каждой точке пространства объектов: $a(x) \approx \mathbb{E}[y \,|\, x]$; если же оптимизировать среднее абсолютное отклонение, то итоговый алгоритм будет приближать медиану распределения: $a(x) \approx \text{median}[p(y \,|\, x)]$. Рассмотрим теперь некоторый объект x и условное распределение $p(y \,|\, x)$. Найдем наиболее оптимальное число q, которым можно приблизить данное распределение с точки зрения функционала

$$Q = \int_{\mathbb{Y}} \rho_\tau(y - q)p(y \,|\, x)dy.$$

Продифференцируем его (при этом необходимо воспользоваться правилами дифференцирования интегралов, зависящих от параметра):

$$\frac{\partial Q}{\partial q} = (1 - \tau) \int_{-\infty}^{q} p(y \,|\, x)dy - \tau \int_{q}^{\infty} p(y \,|\, x)dy = 0.$$

Получаем, что

$$\frac{\tau}{1 - \tau} = \frac{\int_{-\infty}^{q} p(y \,|\, x)dy}{\int_{q}^{\infty} p(y \,|\, x)dy}.$$

Данное уравнение будет верно, если q будет равно τ-квантили распределения $p(y \,|\, x)$. Таким образом, использование функции потерь $\rho_\tau(z)$ приводит к тому, что алгоритм $a(x)$ будет приближать τ-квантиль распределения в каждой точке пространства объектов.

§1.2 Критерии качества в задаче классификации

Начнем с задачи бинарной классификации, в которой метки принадлежат множеству $\{0, 1\}$. Объекты с меткой 1 будем называть положительными, а с меткой 0 — отрицательными. Алгоритм при этом может возвращать произвольное вещественное число, которое с помощью порога переводится в бинарный ответ: $a(x) = [b(x) > t]$.

1.2.1 Доля правильных ответов

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

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

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.

1.2.2 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 качества.

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:

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

	$y = 1$	$y = 0$
$a(x) = 1$	True Positive (TP)	False Positive (FP)
$a(x) = 0$	False negative (FN)	True Negative (TN)

Таблица 1. Матрица ошибок

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

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

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 $y = 0$ — 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 $a(x) = [b(x) > t]$. 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).

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

равна точности:

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

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 часто 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.

1.2.3 Lift

На практике часто возникают задачи, связанные с выбором подмножества: выделение лояльных клиентов банка, выделение уходящих пользователей мобильного оператора и т.д. Заказчика может интересовать вопрос, насколько выгоднее работать с этим подмножеством по сравнению со всем множеством. Если при рассылке предложений о кредите клиентам из подмножества и всем клиентам будет получаться одна и та же доля откликнувшихся, то подмножество не будет представлять особой ценности. Формально это измеряется с помощью *прироста концентрации* (lift), который равен отношению точности к доле положительных объектов в выборке:

$$\text{lift} = \frac{\text{precision}}{(\text{TP} + \text{FN})/\ell}.$$

Эту величину можно интерпретировать как улучшение доли положительных объектов в данном подмножестве относительно доли в случайно выбранном подмножестве такого же размера.

1.2.4 Area Under Curve

Выше мы изучили точность, полноту и F-меру, которые характеризуют качество работы алгоритма $a(x) = [b(x) > t]$ 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 $\{a(x) = [b(x) > t] \,|\, t \in \mathbb{R}\}$.

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 случайно выбранный положительный объект окажется позже случайно выбранного отрицательного объекта в ранжированном списке, порожденном $b(x)$. Разберем подробнее немного другую формулировку.

Задача 1.1. *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 классам: чем больше ответ классификатора, тем более он уверен в том, что данный объект относится к классу «+1». Отсортируем все объекты по неубыванию ответа классификатора b: $x_{(1)}, \ldots, x_{(\ell)}$. Обозначим истинные ответы на этих объектах через $y_{(1)}, \ldots, y_{(\ell)}$. Тогда доля дефектных пар записывается как

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

Как данный функционал связан с AUC-ROC?

Решение. Разберем процедуру построения AUC-ROC. Сначала все объекты сортируются по оценке классификатора. Стартуем из точки $(0, 0)$ — она соответствует порогу $y_{(\ell)}$. Начинаем идти от большей оценки к меньшей. Если текущий объект

имеет класс «1», то у алгоритма увеличивается TPR; значит, ROC-кривая сдвигается вверх на $1/\ell_+$ (ℓ_+ — число объектов положительного класса). Если у текущего объекта класс «0», то алгоритм допускает на одну ошибку больше, чем предыдущий — значит, ROC-кривая сдвигается вправо на $1/\ell_-$ (ℓ_- — число объектов отрицательного класса), а к AUC надо прибавить $\sum_{j=i+1}^{\ell}[y_{(j)}=+1]/(\ell_+\ell_-)$. Получаем:

$$\begin{aligned}
\text{AUC} &= \frac{1}{\ell_+\ell_-}\sum_{i=1}^{\ell}[y_{(i)}=0]\sum_{j=i+1}^{\ell}[y_{(j)}=+1] = \\
&= \frac{1}{\ell_+\ell_-}\sum_{i=1}^{\ell}\sum_{j=i+1}^{\ell}[y_{(i)}<y_{(j)}] = \\
&= \frac{1}{\ell_+\ell_-}\sum_{i<j}^{\ell}(1-[y_{(i)}>y_{(j)}]-[y_{(j)}=y_{(i)}]) = \\
&= \frac{1}{\ell_+\ell_-}\sum_{i<j}^{\ell}(1-[y_{(j)}=y_{(i)}]) - \frac{1}{\ell_+\ell_-}\sum_{i<j}^{\ell}[y_{(i)}>y_{(j)}] = \\
&= \frac{1}{\ell_+\ell_-}\frac{\ell(\ell-1)}{2} - \frac{\ell_+(\ell_+-1)}{2\ell_+\ell_-} - \frac{\ell_-(\ell_--1)}{2\ell_+\ell_-} - \frac{1}{\ell_+\ell_-}\sum_{i<j}^{\ell}[y_{(i)}>y_{(j)}] = \\
&= 1 - \frac{1}{\ell_+\ell_-}\sum_{i<j}[y_{(i)}>y_{(j)}].
\end{aligned}$$

Получаем, что AUC связан с числом дефектных пар DP формулой

$$\text{DP} = \frac{2\ell_-\ell_+}{\ell(\ell-1)}(1-\text{AUC}).$$

■

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

Чувствительность к соотношению классов. Рассмотрим задачу выделения математических статей из множества научных статей. Допустим, что всего имеется 1.000.100 статей, из которых лишь 100 относятся к математике. Если нам удастся построить алгоритм $a(x)$, идеально решающий задачу, то его TPR будет равен единице, а FPR — нулю. Рассмотрим теперь плохой алгоритм, дающий положительный

ответ на 95 математических и 50.000 нематематических статьях. Такой алгоритм совершенно бесполезен, но при этом имеет TPR = 0.95 и FPR = 0.05, что крайне близко к показателям идеального алгоритма.

Таким образом, если положительный класс существенно меньше по размеру, то AUC-ROC может давать неадекватную оценку качества работы алгоритма, поскольку измеряет долю неверно принятых объектов относительно общего числа отрицательных. Так, алгоритм $b(x)$, помещающий 100 релевантных документов на позиции с 50.001-й по 50.101-ю, будет иметь AUC-ROC 0.95.

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 $(0, 0)$. Будем идти по ранжированной выборке, начиная с первого объекта; пусть текущий объект находится на позиции k. Если он относится к классу «0», 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:

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

Отметим, что AUC-PR дает правильные 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].

1.2.5 Многоклассовый случай

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 $a^k(x) = [a(x) = k]$. При микро-усреднении сначала эти характеристики усредняются по всем классам, а затем вычисляется итоговая двухклассовая метрика — например, точность, полнота или F-мера. При макро-усреднении сначала вычисляется итоговая метрика для каждого класса, а затем результаты усредняются по всем классам.

Если классы отличаются по мощности, то при микро-усреднении они практически никак не будут влиять на результат, поскольку их 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, и поэтому каждый класс внесет равный вклад в итоговую метрику.

§1.3 Обучение на несбалансированных выборках

Выше мы изучили проблемы, возникающие при вычислении качества алгоритмов классификации на выборках с несбалансированными классами, а также способы борьбы с ними. Следует помнить, что различия в мощностях классов могут привести и к проблемам с обучением. В данном разделе мы рассмотрим некоторые подходы к улучшению методов обучения в таких ситуациях.

Будем вести речь только о задачах бинарной классификации. Считается, что выборка несбалансированная, если количество объектов в одном из классов превосходит количество объектов в другом в 10 раз или больше. При этом меньший класс называется минорным, больший — доминирующим. Самые простые методы борьбы с несбалансированностью — *undersampling* и *oversampling*. Первый из них удаляет случайные объекты доминирующего класса до тех пор, пока соотношение классов не станет приемлемым; второй дублирует случайные объекты минорного класса. Оптимальное число объектов для удаления или дублирования следует подбирать с помощью кросс-валидации. Отметим, что данные методы применяются лишь к обучающей выборке, а контрольная выборка остается без изменений.

Более сложный метод SMOTE [2] заключается в дополнении минорного класса синтетическими объектами. Генерация нового объекта производится следующим образом. Выбирается случайный объект x_1 минорного класса, для него выделяются k ближайших соседей из этого же класса (k — настраиваемый параметр), из этих соседей выбирается один случайный x_2. Новый объект вычисляется как точка на отрезке между x_1 и x_2: $\alpha x_1 + (1 - \alpha)x_2$, для случайного $\alpha \in (0, 1)$.

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.

[2] *Chawla N., Bowyer K., Hall L., Kegelmeyer W.* (2002). SMOTE: Synthetic Minority Over-sampling Technique. // Journal of Artificial Intelligence Research, Vol. 16, Pp. 321–357.