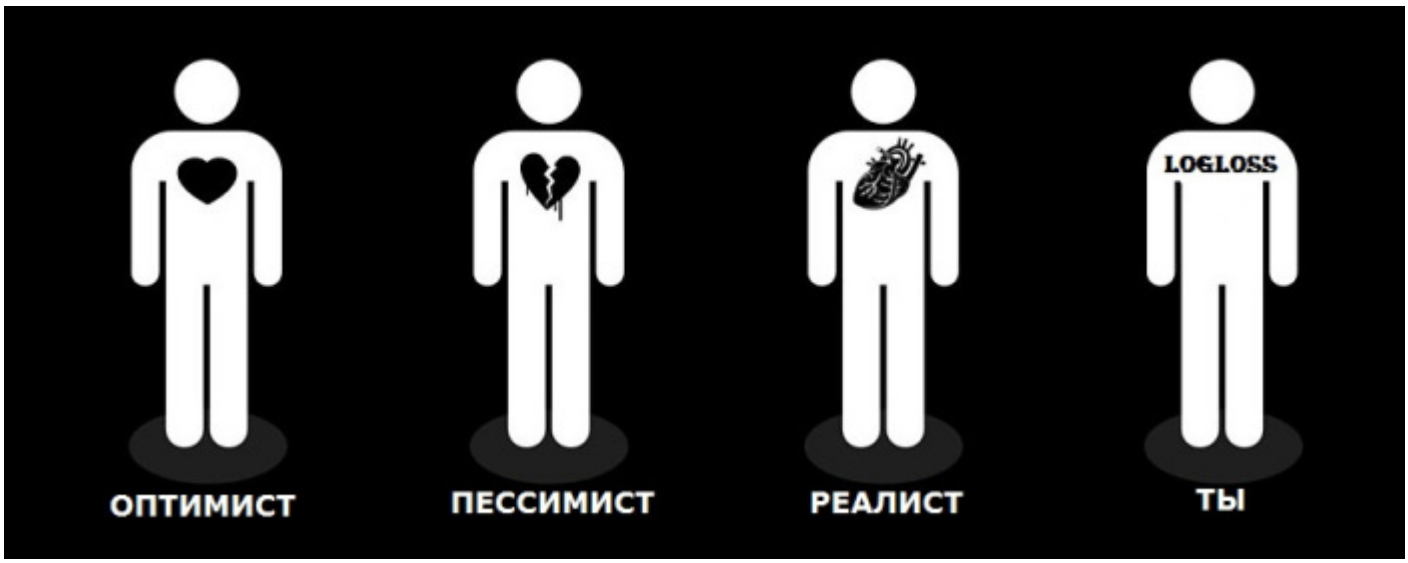
Анализ малых данных

КвазиНаучный блог Александра Дьяконова

Логистическая функция ошибки

2018/03/122018/03/13 alexanderdyakonov образование классификация, метрика качества, оценка качества, ошибка, logloss, регрессия, функция ошибки

Эту функцию называют также «логлосс» (logloss / log_loss (http://scikit-learn.org/stable/modules/generated/sklearn.metrics.log_loss.html)), перекрёстной / кросс-энтропией (https://en.wikipedia.org/wiki/Cross_entropy) (Cross Entropy) и часто используют в задачах классификации. Разберёмся, почему её используют и какой смысл она имеет. Для чтения поста нужна неплохая ML-математическая подготовка, но даже новичкам я бы рекомендовал почитать (хотя я не очень заботился, чтобы «всё объяснялось на пальцах»).



Начнём издалека…

Вспомним, как решается задача линейной регрессии (http://www.machinelearning.ru/wiki/index.php?title=%D0%9B%D0%B8%D0%BD%D0%B5%D0%B9%D0%BD%D0%B0%D1%8F_%D1%80%D0%B5%D0%B3%D1%80%D0%B5%D1%81%D1%81%D0%B8%D1%8F_(%D0%BF%D1%80%D0%B8%D0%BC%D0%B5%D1%80)). Итак, мы хотим получить линейную функцию (т.е. веса ***w***), которая приближает целевое значение с точностью до ошибки:

$$y = w^{\mathrm{T}} x + \varepsilon$$
$$\varepsilon \sim \mathrm{norm}(0, \sigma^2)$$

Здесь мы предположили, что ошибка нормально распределена (https://ru.wikipedia.org/wiki/%D0%9D%D0%BE%D1%80%D0%BC%D0%B0%D0%BB%D1%8C%D0%BD%D0%BE%D0%B5_%D1%80%D0%B0%D1%81%D0%BF%D1%80%D0%B5%D0%B4%D0%B5%D0%BB%D0%B5%D0%BD%D0%B8%D0%B5), ***x*** – признаковое описание объекта (возможно, в нём есть и фиктивный константный признак, чтобы в линейной функции был свободный член). Тогда мы знаем как распределены ответы нашей функции и можем записать функцию правдоподобия выборки (https://ru.wikipedia.org/wiki/%D0%A4%D1%83%D0%BD%D0%BA%D1%86%D0%B8%D1%8F_%D0%BF%D1%80%D0%B0%D0%B2%D0%B4%D0%BE%D0%BF%D0%BE%D0%B4%D0%BE%D0%B1%D0%B8%D1%8F) (т.е. произведение плотностей, в которые подставлены значения из обучающей выборки) и воспользоваться методом максимального правдоподобия (https://ru.wikipedia.org/wiki/%D0%9C%D0%B5%D1%82%D0%BE%D0%B4_%D0%BC%D0%B0%D0%BA%D1%81%D0%B8%D0%BC%D0%B0%D0%BB%D1%8C%D0%BD%D0%BE%D0%B3%D0%BE_%D0%BF%D1%80%D0%B0%D0%B2%D0%B4%D0%BE%D0%BF%D0%BE%D0%B4%D0%BE%D0%B1%D0%B8%D1%8F) (в котором для определения значений параметров берётся максимум правдоподобия, а чаще – его логарифма):

$$p(y \mid x, w) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left[-\frac{(y - w^{\mathrm{T}} x)^2}{2\sigma^2} \right]$$

$$L(w) = \log \prod_{i=1}^{m} p(y_i \mid x_i, w) = \sum_{i=1}^{m} \left[-\frac{1}{2} \log(2\pi\sigma^2) - \frac{(y_i - w^{\mathrm{T}} x_i)^2}{2\sigma^2} \right] \to \max$$

$$\frac{1}{2\sigma^2} \sum_{i=1}^{m} (y_i - w^{\mathrm{T}} x_i)^2 \to \min$$

В итоге оказывается, что максимизация правдоподобия эквивалентна минимизации среднеквадратичной ошибки (MSE) (https://en.wikipedia.org/wiki/Mean_squared_error), т.е. эта функция ошибки не зря широко используется в задачах регрессии. Кроме того, что она вполне логична, легко дифференцируема по параметрам и легко минимизируется, она ещё и теоретически обосновывается с помощью метода максимального правдоподобия в случае, если линейная модель соответствует данным с точностью до нормального шума.

Давайте ещё посмотрим, как реализуется метод стохастического градиента (http://www.machinelearning.ru/wiki/index.php?title=%D0%9C%D0%B5%D1%82%D0%BE%D0%B4_%D1%81%D1%82%D0%BE%D1%85%D0%B0%D1%81%D1%82%D0%B8%D1%87%D0%B5%D1%81%D0%BA%D0%BE%D0%B3%D0%BE_%D0%B3%D1%80%D0%B0%D0%B4%D0%B8%D0%B5%D0%BD%D1%82%D0%B0) (SGD) для минимизации MSE: надо взять производную функции ошибки для конкретного объекта и записать формулу коррекции весов в виде «шага в сторону антиградиента»:

$$J(w) = \frac{1}{2}(w^{\mathrm{T}} \cdot x - y)^2 \to \min$$

$$\frac{\partial J}{\partial w} = (w^{\mathrm{T}} \cdot x - y) \cdot x$$

$$w := w - \alpha \cdot (w^{\mathrm{T}} \cdot x - y) \cdot x$$

Получили, что веса линейной модели при её обучении методом SGD корректируются с помощью добавки вектора признаков. Коэффициент, с которым добавляют, зависит от «агрессивности алгоритма» (параметр альфа, который называют темпом обучения (http://www.machinelearning.ru

/wiki/index.php?title=%D0%9C%D0%B5%D1%82%D0%BE%D0%B4_%D1%81%D1%82%D0%BE%D1%85%D0%B0%D1%81%D1%82%D0%B8%D1%87%D0%B5%D1%81%D0%BA%D0%BE%D0%B3%D0%BE_%D0%B3%D1%80%D0%B0%D0%B4%D0%B8%D0%B5%D0%BD%D1%82%D0%B0)) и разности «ответ алгоритма – правильный ответ». Кстати, если разница нулевая (т.е. на данном объекте алгоритм выдаёт точный ответ), то коррекция весов не производится.

Log Loss

Теперь давайте, наконец, поговорим о «логлоссе». Рассматриваем задачу классификации с двумя классами: 0 и 1. Обучающую выборку можно рассматривать, как реализацию обобщённой схемы Бернулли: для каждого объекта генерируется случайная величина, которая с вероятностью ***p*** (своей для каждого объекта) принимает значение 1 и с вероятностью (1–***p***) – 0. Предположим, что мы как раз и строим нашу модель так, чтобы она генерировала правильные вероятности, но тогда можно записать функцию правдоподобия:

$a_i = a(x_i \mid w)$ – ответ алгоритма, зависящего от параметров w, на i-м объекте

$$p(y \mid X, w) = \prod_i p(y_i \mid x_i, w) = \prod_i a_i^{y_i} (1 - a_i)^{1-y_i} \to \max$$

$$\sum_i (-y_i \log a_i - (1 - y_i) \log(1 - a_i)) \to \min$$

После логарифмирования правдоподобия получили, что его максимизация эквивалентна минимизации последнего записанного выражения. Именно его и называют **«логистической функции ошибки»**. Для задачи бинарной классификации, в которой алгоритм должен выдать вероятность принадлежности классу 1, она логична ровно настолько, насколько логична MSE в задаче линейной регрессии с нормальным шумом (поскольку **обе функции ошибки выводятся из метода максимального правдоподобия**).

Часто гораздо более понятна такая запись logloss-ошибки на одном объекте:

$$-\begin{cases} \log a_i, & y_i = 1, \\ \log(1-a_i), & y_i = 0. \end{cases}$$

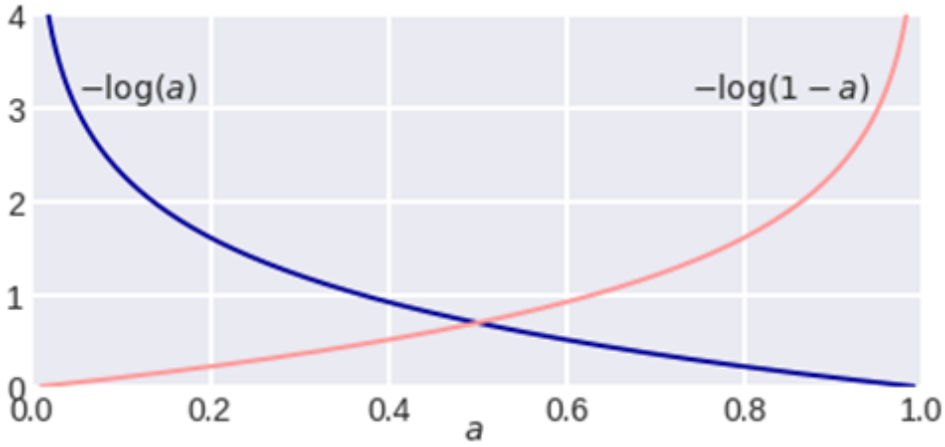


Рис. 1. logloss-ошибка на одном объекте.

Отметим неприятное свойство логосса: если для объекта 1го класса мы предсказываем нулевую вероятность принадлежности к этому классу или, наоборот, для объекта 0го – единичную вероятность принадлежности к классу 1, то ошибка равна бесконечности! Таким образом, **грубая ошибка на одном объекте сразу делает алгоритм бесполезным.** На практике часто логлосс ограничивают каким-то большим числом (чтобы не связываться с бесконечностями).

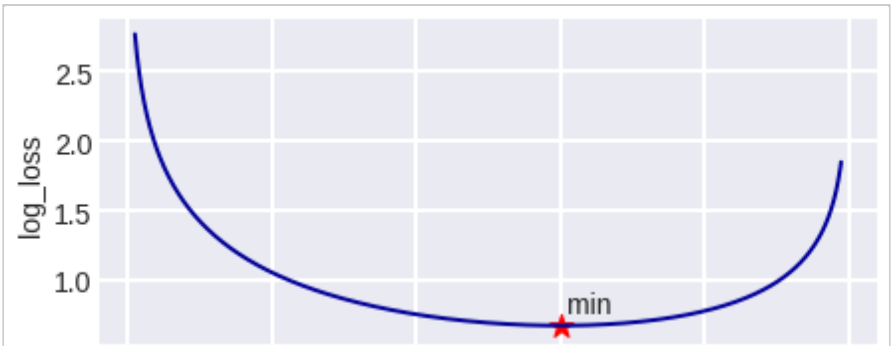
Если задаться вопросом, какой константный алгоритм оптимален для выборки из ***q_1*** представителей класса 1 и ***q_0*** представителей класса 0, ***q_1*** + ***q_0*** = ***q*** , то получим

$$-\frac{1}{q}\sum_{i=1}^{q}(y_i \log a + (1-y_i)\log(1-a)) \to \min_a$$

$$-\frac{q_1}{q}\log a - \frac{q_0}{q}\log(1-a) \to \min_a$$

$$a = \frac{q_1}{q}$$

Последний ответ получается взятием производной и приравниванием её к нулю. Описанную задачу приходится решать, например, при построении решающих деревьев (какую метку приписывать листу, если в него попали представители разных классов). На рис. 2 изображён график log_loss-ошибки константного алгоритма для выборки из четырёх объектов класса 0 и 6 объектов класса 1.



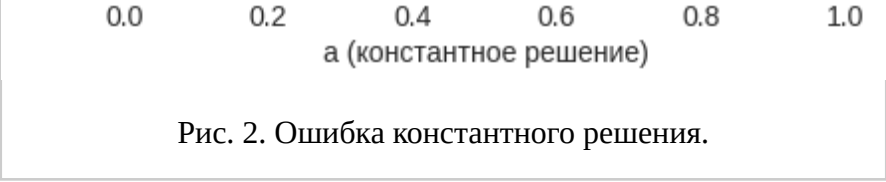


Рис. 2. Ошибка константного решения.

Представим теперь, что мы знаем, что объект принадлежит к классу 1 вероятностью ***p***, посмотрим, какой ответ оптимален на этом объекте с точки зрения log_loss: матожидание нашей ошибки

$$-p\log(a_i)-(1-p)\log(1-a_i)$$

$$\frac{p}{a_i}-\frac{1-p}{1-a_i}=0$$

$$a_i=p$$

Для минимизации ошибки мы опять взяли производную и приравняли к нулю. Мы получили, что оптимально для каждого объекта выдавать его вероятность принадлежности к классу 1! Таким образом, для минимизации log_loss надо уметь вычислять (оценивать) вероятности принадлежности классам!

Если подставить полученное оптимальное решение в минимизируемый функционал, то получим энтропию:

$$-p\log(p)-(1-p)\log(1-p).$$

Это объясняет, почему при построении решающих деревьев в задачах классификации (а также случайных лесов и деревьях в бустингах) применяют энтропийный критерий расщепления (https://www.google.ru/url?sa=t&rct=j&q=&esrc=s&source=web&cd=4&cad=rja&uact=8&ved=0ahUKEwjJ9MO8-OTZAhUBIpoKHcccBvIQFghNMAM&url=http%3A%2F%2Fjmlda.org%2Fpapers%2Fdoc%2F2014%2Fno8%2FGenrikhov2014Criteria.pdf&usg=AOvVaw0xROJPSoV2U39BcJUETxo4) (ветвления). Дело в том, что оценка принадлежности к классу 1 часто производится с помощью среднего арифметического меток в листе. В любом случае, для конкретного дерева эта вероятность будет одинакова для всех объектов в листе, т.е. константой. Таким образом, энтропия в листе примерно равна логлосс-ошибке константного решения. Используя энтропийный критерий мы неявно оптимизируем логлосс!

В каких пределах может варьироваться logloss? Ясно, что минимальное значение 0, максимальное – +∞, но эффективным максимальным можно считать ошибку при использовании константного алгоритма (вряд же мы в итоге решения задачи придумаем алгоритм хуже константы?!), т.е.

$$\left[0,\ -\frac{q_1}{q}\log\frac{q_1}{q}-\frac{q_0}{q}\log\frac{q_0}{q}\right]$$

Интересно, что если брать логарифм по основанию 2, то на сбалансированной выборке это отрезок [0, 1].

Связь с логистической регрессией

Слово «логистическая» в названии ошибки намекает на связь с логистической регрессией (https://ru.wikipedia.org/wiki/%D0%9B%D0%BE%D0%B3%D0%B8%D1%81%D1%82%D0%B8%D1%87%D0%B5%D1%81%D0%BA%D0%B0%D1%8F_%D1%80%D0%B5%D0%B3%D1%80%D0%B5%D1%81%D1%81%D0%B8%D1%8F) – это как раз метод для решения задачи бинарной классификации, который получает вероятность принадлежности к классу 1. Но пока мы исходили из общих предположений, что наш алгоритм генерирует эту вероятность (алгоритмом может быть, например, случайный лес или бустинг над деревьями). Покажем, что тесная связь с логистической регрессией всё-таки есть… посмотрим, как настраивается логистическая регрессия (т.е. сигмоида от линейной комбинации) на эту функцию ошибки методом SGD.

$$\text{logloss}(a, y) = -y \log a - (1 - y) \log(1 - a)$$

$$a = \text{sigmoid}(w^{\mathrm{T}} x) \equiv \frac{1}{1 + e^{-w^{\mathrm{T}} x}}$$

$$\frac{\partial \text{logloss}}{\partial w} = (a - y)x$$

$$w := w - \alpha(a - y)x$$

Как видим, корректировка весов точно такая же, как и при настройке линейной регрессии! На самом деле, это говорит о родстве разных регрессий: линейной и логистической, а точнее, о родстве распределений: нормального и Бернулли. Желающие могут внимательно почитать лекцию Эндрю Ына (http://cs229.stanford.edu/notes/cs229-notes1.pdf).

Во многих книгах логистической функцией ошибки (т.е. именно «logistic loss») называется другое выражение, которое мы сейчас получим, подставив выражение для сигмоиды в logloss и сделав переобозначение: считаем, что метки классов теперь –1 и +1, тогда

$$\text{logloss}(a, y) = \log(1 + \exp(-y \cdot w^{\mathrm{T}} x))$$

Полезно посмотреть на график функции, центральной в этом представлении:

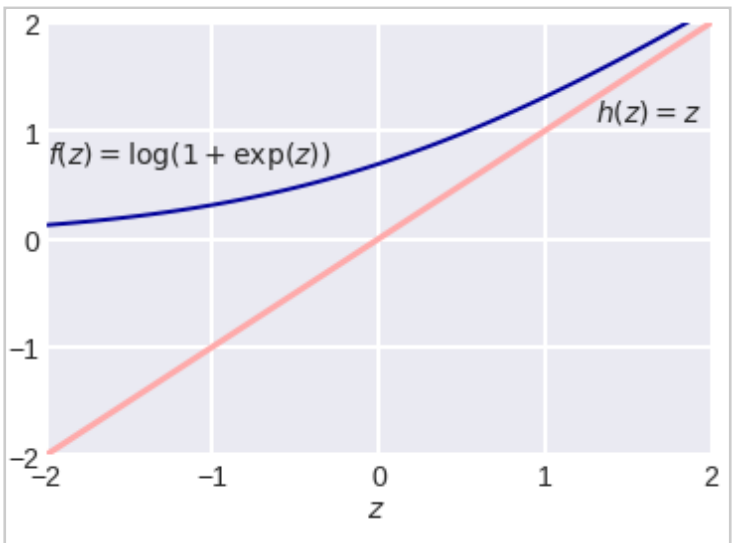


Рис. 3. Графики нескольких функций.

Как видно, это сглаженный (всюду дифференцируемый) аналог функции **max(0, *x*)**, которую в глубоком обучении принято называть **ReLu** (Rectified Linear Unit) (https://en.wikipedia.org/wiki/Rectifier_(neural_networks)). Если при настройке весов минимизировать logloss, то таким образом мы настраиваем классическую логистическую регрессию, если же использовать ReLu, чуть-чуть подправить аргумент и добавить регуляризацию, то получаем классическую настройку SVM (https://svmtutorial.online/):

$$\sum_i \max[1 - y_i w^T x, 0] + \alpha w^T w \to \min,$$

выражение под знаком суммы принято называть Hinge loss (https://en.wikipedia.org/wiki/Hinge_loss). Как видим, часто с виду совсем разные методы можно получать «немного подправив» оптимизируемые функции на похожие. Между прочим, при обучении RVM (https://en.wikipedia.org/wiki/Relevance_vector_machine) (Relevance vector machine) используется тоже очень похожий функционал:

$$\sum_i \log(1 + \exp(-y_i w^T x)) + w^T \operatorname{diag}(\alpha) w \to \min.$$

Связь с расхождением Кульбака-Лейблера

Расхождение (дивергенцию) Кульбака-Лейблера (https://ru.wikipedia.org/wiki/%D0%A0%D0%B0%D1%81%D1%81%D1%82%D0%BE%D1%8F%D0%BD%D0%B8%D0%B5_%D0%9A%D1%83%D0%BB%D1%8C%D0%B1%D0%B0%D0%BA%D0%B0_%E2%80%94_%D0%9B%D0%B5%D0%B9%D0%B1%D0%BB%D0%B5%D1%80%D0%B0) (KL, Kullback–Leibler divergence) часто используют (особенно в машинном обучении, байесовском подходе и теории информации) для вычисления непохожести двух распределений. Оно определяется по следующей формуле:

$$D_{\mathrm{KL}}(P \parallel Q) = \int p(z) \log \frac{p(z)}{q(z)} \partial z,$$

где ***P*** и ***Q*** – распределения (первое обычно «истинное», а второе – то, про которое нам интересно, насколько оно похоже на истинное), ***p*** и ***q*** – плотности этих распределений. Часто KL-расхождение называют расстоянием, хотя оно не является симметричным и не удовлетворяет неравенству треугольника. Для дискретных распределений формулу записывают так:

$$D_{\mathrm{KL}}(P \parallel Q) = \sum_i P_i \log \frac{P_i}{Q_i}$$

P_i, ***Q_i*** – вероятности дискретных событий. Давайте рассмотрим конкретный объект ***x*** с меткой ***y***. Если алгоритм выдаёт вероятность принадлежности первому классу – ***a***, то предполагаемое распределение на событиях «класс 0», «класс 1» – **(1–*a*, *a*)**, а истинное – **(1–*y*, *y*)**, поэтому расхождение Кульбака-Лейблера между ними

$$(1-y)\log\frac{(1-y)}{(1-a)} + y\log\frac{y}{a} = -(1-y)\log(1-a) - y\log a,$$

что в точности совпадает с logloss.

Настройка на logloss

Один из методов «подгонки» ответов алгоритма под logloss – **калибровка Платта (https://en.wikipedia.org/wiki/Platt_scaling) (Platt calibration)**. Идея очень простая. Пусть алгоритм порождает некоторые оценки принадлежности к 1му классу – ***a***. Метод изначально разрабатывался для калибровки ответов алгоритма опорных векторов (https://svmtutorial.online/) (SVM), этот алгоритм в простейшей реализации разделяет объекты гиперплоскостью и просто выдаёт номер класса 0 или 1, в зависимости от того, с какой стороны гиперплоскости объект расположен. Но если мы построили гиперплоскость, то для любого объекта можем вычислить расстояние до неё (со знаком минус, если объект лежит в полуплоскости нулевого класса). Именно эти расстояния со знаком ***r*** мы будем превращать в вероятности по следующей формуле:

$$a(x) = \text{sigmoid}(\alpha \cdot r(x) + \beta),$$

неизвестные параметры ***α, β*** обычно определяются методом максимального правдоподобия на отложенной выборке (calibration set).

Проиллюстрируем применение метода на реальной задаче, которую автор решал недавно. На рис. показаны ответы (в виде вероятностей) двух алгоритмов: градиентного бустинга (lightgbm (https://alexanderdyakonov.wordpress.com/2017/06/09/%d0%b3%d1%80%d0%b0%d0%b4%d0%b8%d0%b5%d0%bd%d1%82%d0%bd%d1%8b%d0%b9-%d0%b1%d1%83%d1%81%d1%82%d0%b8%d0%bd%d0%b3/)) и случайного леса (random forest (https://alexanderdyakonov.wordpress.com/2016/11/14/%d1%81%d0%bb%d1%83%d1%87%d0%b0%d0%b9%d0%bd%d1%8b%d0%b9-%d0%bb%d0%b5%d1%81-random-forest/)).

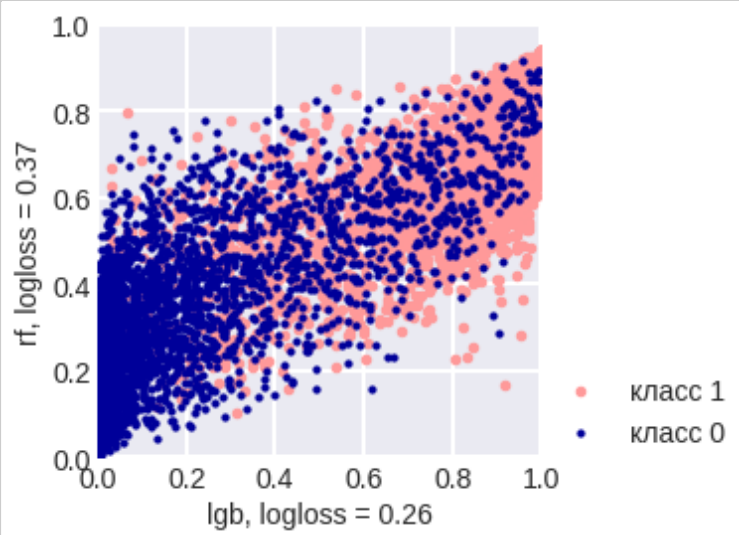
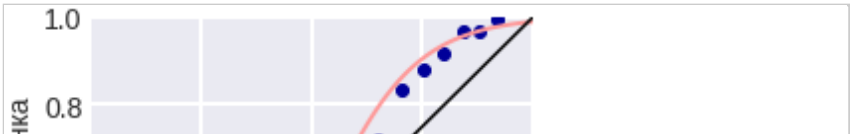


Рис. 4. Ответы двух алгоритмов на всех объектах выборки.

Видно, что качество леса намного ниже и он довольно осторожен: занижает вероятности у объектов класса 1 и завышает у объектов класса 0. Упорядочим все объекты по возрастанию разобьем на ***k*** равных частей и для каждой части вычислим среднее всех ответов алгоритма и среднее всех правильных ответов. Результат показан на рис. 5 – точки изображены как раз в этих двух координатах.



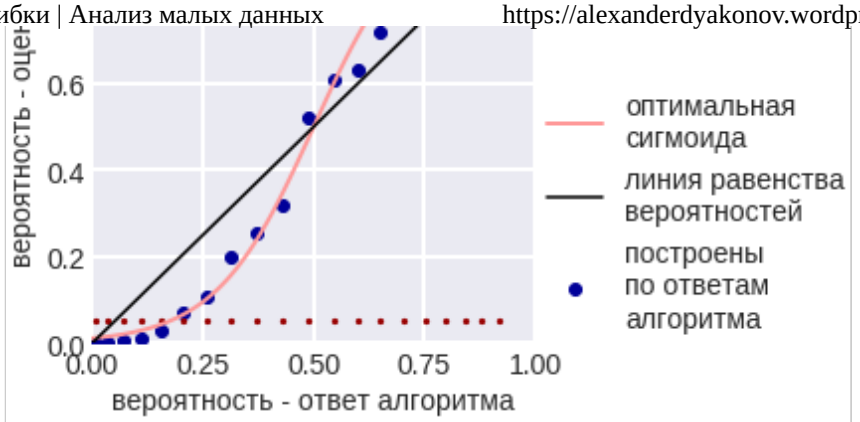


Рис. 5. Соотношения вероятностей: оценённой алгоритмом и усреднением.

Нетрудно видеть, что точки располагаются на линии, похожей на сигмоиду – можно оценить параметр сжатия-растяжения в ней, см. рис. 6. Оптимальная сигмоида показана розовым цветом на рис. 5. Если подвергать ответы такой сигмоидной деформации, то логлосс-ошибка случайного леса снижается с 0.37 до 0.33.

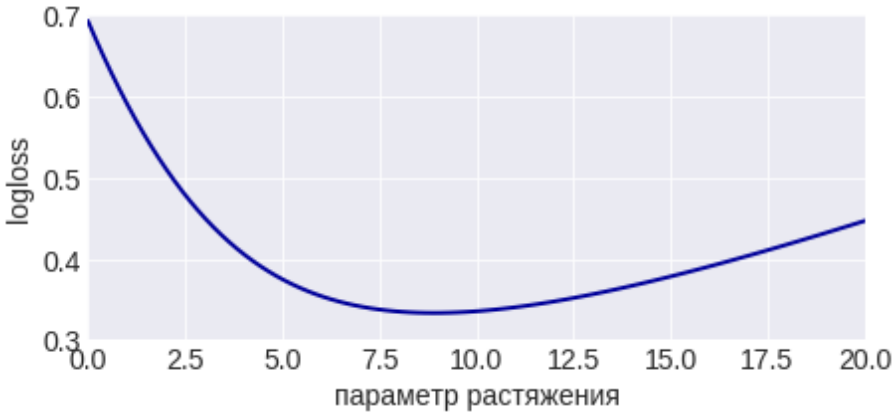


Рис. 6. Ошибка в зависимости от коэффициента сжатия аргумента сигмоиды.

Обратите внимание, что здесь мы деформировали ответы случайного леса (это были оценки вероятности – и все они лежали на отрезке [0, 1]), но из рис. 5 видно, что для деформации нужна именно сигмоида. Практика показывает, что **в 80% ситуаций для улучшения logloss-ошибки надо деформировать ответы именно с помощью сигмоиды** (для меня это также часть объяснения, почему именно такие функции успешно используются в качестве функций активаций в нейронных сетях).

Ещё один вариант калибровки – монотонная регрессия (Isotonic regression (http://scikit-learn.org/stable/auto_examples/plot_isotonic_regression.html)).

Многоклассовый logloss

Для полноты картины отметим, что logloss обобщается и на случай нескольких классов естественным образом:

$$\text{logloss} = -\frac{1}{q}\sum_{i}^{q}\sum_{j}^{l} y_{ij} \log a_{ij}$$

$$\frac{1}{q}\sum_{i=1}^{q}\sum_{j=1}^{l}$$

здесь ***q*** – число элементов в выборке, ***l*** – число классов, ***a_ij*** – ответ (вероятность) алгоритма на ***i***-м объекте на вопрос принадлежности его к ***j***-му классу, ***y_ij*=1** если ***i***-й объект принадлежит ***j***-му классу, в противном случае ***y_ij*=0**.

На посошок…

В каждом подобном посте я стараюсь написать что-то из мира машинного обучения, что, с одной стороны, просто и понятно, а с другой – изложение этого не встречается больше нигде. Например, есть такой естественный вопрос: **почему в задачах классификации при построении решающих деревьев используют энтропийный критерий расщепления**? Во всех курсах его (критерий) преподносят либо как эвристику, которую «вполне естественно использовать», либо говорят, что «энтропия похожа на кросс-энтропию». Сейчас стоимость некоторых курсов по машинному обучению достигает нескольких сотен тысяч рублей, но «профессиональные инструкторы» не могут донести простую цепочку:

- в статистической теории обучения настройка алгоритма производится максимизацией правдоподобия,
- в задаче бинарной классификации это эквивалентно минимизации логлосса, а сам минимум как раз равен энтропии,
- поэтому использование энтропийного критерия фактически эквивалентно выбору расщепления, минимизирующего логлосс.

Если Вы всё-таки отдали несколько сотен тысяч рублей, то можете проверить «профессиональность инструктора» следующими вопросами:

- Энтропия в листе примерно равна logloss-ошибке константного решения. Почему не использовать саму ошибку, а не приближённое значение? Или, как часто происходит в задачах оптимизации, её верхнюю оценку?
- Минимизации какой ошибки соответствует критерий расщепления Джини?
- Можно показать, что если в задаче бинарной классификации использовать в качестве функции ошибки среднеквадратичное отклонение, то также, как и для логлосса, оптимальным ответом на объекте будет вероятность его принадлежности к классу 1. Почему тогда не использовать такую функцию ошибки?

Ответы типа «так принято», «такой функции не существует», «это только для регрессии», естественно, заведомо неправильные. Если Вам не ответят с такой же степенью подробности, как в этом посте, то Вы точно переплатили;)

П.С. Что ещё почитать…

В этом блоге я публиковал уже несколько постов по метрикам качества…

- AUC ROC (площадь под кривой ошибок) (https://alexanderdyakonov.wordpress.com/2017/07/28/auc-roc-%d0%bf%d0%bb%d0%be%d1%89%d0%b0%d0%b4%d1%8c-%d0%bf%d0%be%d0%b4-%d0%ba%d1%80%d0%b8%d0%b2%d0%be%d0%b9-%d0%be%d1%88%d0%b8%d0%b1%d0%be%d0%ba/)
- Задачки про AUC (ROC) (https://alexanderdyakonov.wordpress.com/2015/10/09/%d0%b7%d0%b0%d0%b4%d0%b0%d1%87%d0%ba%d0%b8-%d0%bf%d1%80%d0%be-auc-roc/)
- Знакомьтесь, Джини (https://alexanderdyakonov.wordpress.com/2015/12/15/%d0%b7%d0%bd%d0%b0%d0%ba%d0%be%d0%bc%d1%8c%d1%82%d0%b5%d1%81%d1%8c-%d0%b4%d0%b6%d0

%b8%d0%bd%d0%b8/)

И буквально на днях вышла классная статья Дмитрия Петухова про коэффициент Джини, читать обязательно:

- Коэффициент Джини. Из экономики в машинное обучение (https://habrahabr.ru/company/ods/blog/350440/)

Логистическая функция ошибки: 6 комментариев

1. **Дмитрий**:
   2018/03/12 в 12:53

   Интересно было бы добавить еще часть про вывод лог. регрессии из байесовского вывода для экспоненциального семейства распределений. Там и интуиция сигмоиды проясняется.

   Ответить (https://alexanderdyakonov.wordpress.com/2018/03/12/%d0%bb%d0%be%d0%b3%d0%b8%d1%81%d1%82%d0%b8%d1%87%d0%b5%d1%81%d0%ba%d0%b0%d1%8f-%d1%84%d1%83%d0%bd%d0%ba%d1%86%d0%b8%d1%8f-%d0%be%d1%88%d0%b8%d0%b1%d0%ba%d0%b8/?replytocom=1603#respond)
   - **alexanderdyakonov**:
     2018/03/13 в 00:48

     Ну, это уже больше про лог.регрессию, а не про логлосс. Про байесовский вывод я хотел написать отдельный пост… но, вообще, большой вопрос, насколько и куда надо углубляться при изучении машинного обучения (если не готовить учёных-исследователей в этой области). Например, Энрю Ын много говорит про обобщённые линейные модели, экспоненциальные семейства и т.п. И совсем мало про кластеризацию. На мой взгляд, это совсем непонятный перекос. Про первое можно вообще ничего не говорить (тем более, в курсе для «начинающих»), и странно, что кроме k-means и EM-алгоритма начинающему ничего не рассказывается про кластеризацию.

     Ответить (https://alexanderdyakonov.wordpress.com/2018/03/12/%d0%bb%d0%be%d0%b3%d0%b8%d1%81%d1%82%d0%b8%d1%87%d0%b5%d1%81%d0%ba%d0%b0%d1%8f-%d1%84%d1%83%d0%bd%d0%ba%d1%86%d0%b8%d1%8f-%d0%be%d1%88%d0%b8%d0%b1%d0%ba%d0%b8/?replytocom=1606#respond)
2. **Rustam Guliev**:
   2018/03/12 в 15:41

   Спасибо большое, за интересный пост!
   Связь с логистической регрессией: logloss(y,a)=-y*log(a) + (1-y)*log(1-a). Тут, кажется опечатка. Вроде, минус должно быть, не?

   Ответить (https://alexanderdyakonov.wordpress.com/2018/03/12/%d0%bb%d0%be%d0%b3%d0%b8%d1%81%d1%82%d0%b8%d1%87%d0%b5%d1%81%d0%ba%d0%b0%d1%8f-%d1%84%d1%83%d0%bd%d0%ba%d1%86%d0%b8%d1%8f-%d0%be%d1%88%d0%b8%d0%b1%d0%ba%d0%b8/?replytocom=1604#respond)
   - **alexanderdyakonov**:
     2018/03/13 в 00:41

     Да, спасибо. Исправил.

     Ответить (https://alexanderdyakonov.wordpress.com/2018/03/12/%d0%bb%d0%be%d0%b3%d0%b8

%b8%d1%81%d1%82%d0%b8%d1%87%d0%b5%d1%81%d0%ba%d0%b0%d1%8f-%d1%84%d1%83%d0%bd%d0%ba%d1%86%d0%b8%d1%8f-%d0%be%d1%88%d0%b8%d0%b1%d0%ba%d0%b8/?replytocom=1605#respond)

3. **Андрюха**:
   2018/03/13 в 05:29

   Дивергенция Кульбака-Лейблера это кросс энтропия плюс энтропия, а написано равенство лог лоссу, стоит написать что при оптимизации энтропия — константа для ответов.

   Ответить (https://alexanderdyakonov.wordpress.com/2018/03/12/%d0%bb%d0%be%d0%b3%d0%b8%d1%81%d1%82%d0%b8%d1%87%d0%b5%d1%81%d0%ba%d0%b0%d1%8f-%d1%84%d1%83%d0%bd%d0%ba%d1%86%d0%b8%d1%8f-%d0%be%d1%88%d0%b8%d0%b1%d0%ba%d0%b8/?replytocom=1607#respond)

   - **alexanderdyakonov**:
     2018/03/13 в 12:02

     Ну, не плюс, ДКЛ = перекрёстная энтропия МИНУС энтропия, но последняя вычисляется для вырожденного распределения, поэтому равна нулю! У меня не просто написано, что ДКЛ = логлоссу, у меня это доказывается с указанием, какие распределения имеются в виду!

     А вот конец предложения я не понял…

     Ответить (https://alexanderdyakonov.wordpress.com/2018/03/12/%d0%bb%d0%be%d0%b3%d0%b8%d1%81%d1%82%d0%b8%d1%87%d0%b5%d1%81%d0%ba%d0%b0%d1%8f-%d1%84%d1%83%d0%bd%d0%ba%d1%86%d0%b8%d1%8f-%d0%be%d1%88%d0%b8%d0%b1%d0%ba%d0%b8/?replytocom=1608#respond)