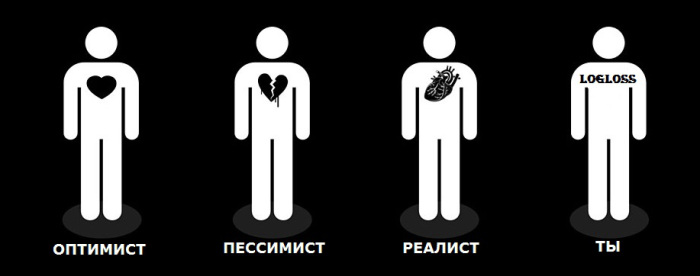
Эту функцию называют также «логлосс» (logloss / [log\_loss](http://scikit-learn.org/stable/modules/generated/sklearn.metrics.log_loss.html" \t "_blank)), [перекрёстной / кросс-энтропией](https://en.wikipedia.org/wiki/Cross_entropy) (Cross Entropy) и часто используют в задачах классификации. Разберёмся, 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»).

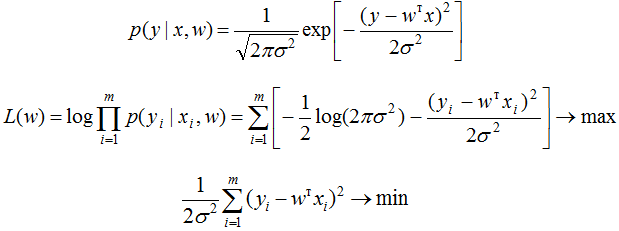


**Начнём издалека…**

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:

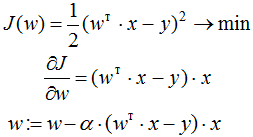
log_loss_01

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):



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.

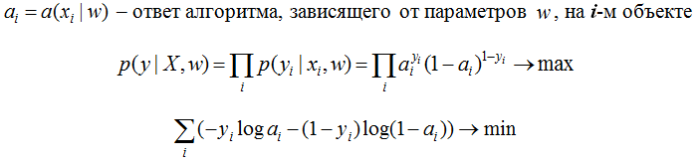
Давайте ещё посмотрим, как реализуется [метод стохастического градиента](http://www.machinelearning.ru/wiki/index.php?title=%D0%9C%D0%B5%D1%82%D0%BE%D0%B4_%D1%81%D1%82%D0%BE%D1%85%D0%B0%D1%81%D1%82%D0%B8%D1%87%D0%B5%D1%81%D0%BA%D0%BE%D0%B3%D0%BE_%D0%B3%D1%80%D0%B0%D0%B4%D0%B8%D0%B5%D0%BD%D1%82%D0%B0)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



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.

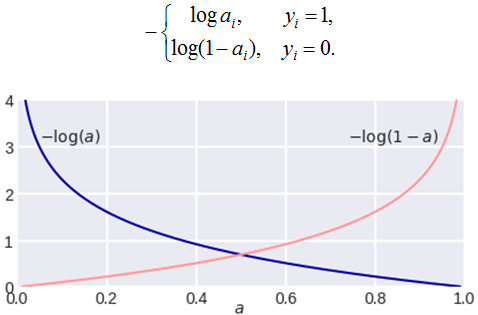
**Log Loss**

**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:



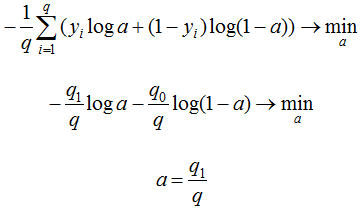
После логарифмирования правдоподобия получили, что его максимизация эквивалентна минимизации последнего записанного выражения. Именно его и называют **«логистической функции ошибки»**. Для задачи бинарной классификации, в которой алгоритм должен выдать вероятность принадлежности классу 1, она логична 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:

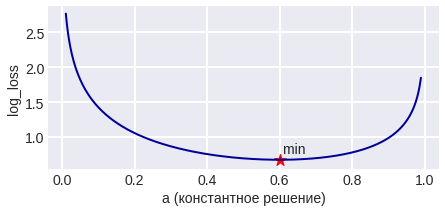
Рис. 1. logloss-ошибка на одном объекте.

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).

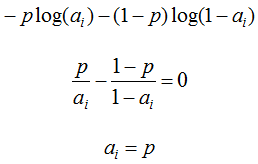
Если задаться вопросом, какой константный алгоритм оптимален для выборки из ***q\_*1**  представителей класса 1 и  ***q\_*0** представителей класса 0, ***q\_*1**+ ***q\_*0*= q*** , то получим



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



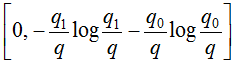
Для минимизации ошибки мы опять взяли производную и приравняли к нулю. Мы получили, что оптимально для каждого объекта выдавать его вероятность принадлежности к классу 1! Таким образом, для минимизации log\_loss надо уметь вычислять (оценивать) вероятности принадлежности классам!

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

log_loss_09.png

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" \t "_blank) (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!

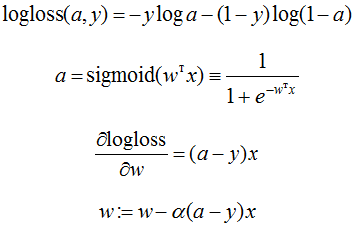
В каких пределах может варьироваться logloss? Ясно, что минимальное значение 0, максимальное – **+∞**, но эффективным максимальным можно считать ошибку при использовании константного алгоритма (вряд же мы в итоге решения задачи придумаем алгоритм хуже константы?!), т.е.



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" \t "_blank) – 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.

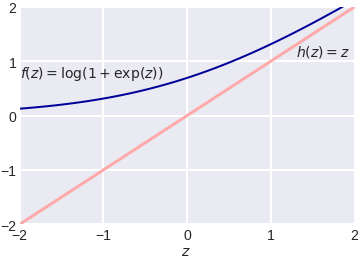


Как видим, корректировка весов точно такая же, как и при настройке линейной регрессии! На самом деле, это говорит о родстве разных регрессий: линейной и логистической, а точнее, о родстве распределений: нормального и Бернулли. Желающие могут внимательно почитать [лекцию Эндрю Ына](http://cs229.stanford.edu/notes/cs229-notes1.pdf).

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

log_loss_12.png

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](https://en.wikipedia.org/wiki/Rectifier_(neural_networks)" \t "_blank)**[(Rectified Linear Unit)](https://en.wikipedia.org/wiki/Rectifier_(neural_networks)" \t "_blank). 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/):

log_loss_14.png

выражение под знаком суммы принято называть [Hinge loss](https://en.wikipedia.org/wiki/Hinge_loss" \t "_blank). 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:

log_loss_15

**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:

log_loss_16

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:

log_loss_17.png

***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

log_loss_18.png

что в точности совпадает с logloss.

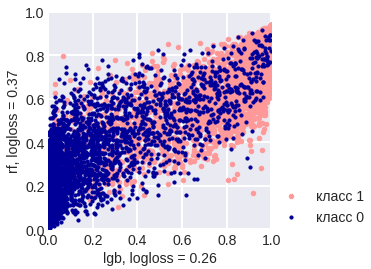
**Настройка на logloss**

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:

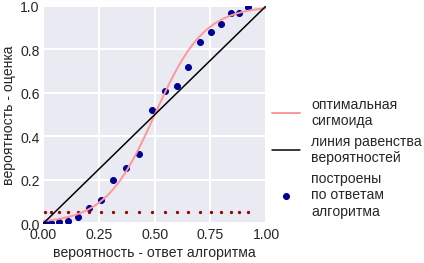
log_loss_19.png

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

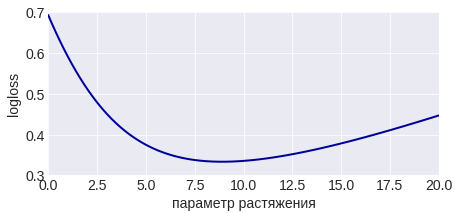
Проиллюстрируем применение метода на реальной задаче, которую автор решал недавно. На рис. показаны ответы (в виде вероятностей) двух алгоритмов: градиентного бустинга ([lightgbm](https://alexanderdyakonov.wordpress.com/2017/06/09/%d0%b3%d1%80%d0%b0%d0%b4%d0%b8%d0%b5%d0%bd%d1%82%d0%bd%d1%8b%d0%b9-%d0%b1%d1%83%d1%81%d1%82%d0%b8%d0%bd%d0%b3/" \t "_blank)) 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.

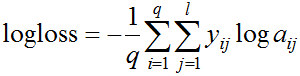
Рис. 6. Ошибка в зависимости от коэффициента сжатия аргумента сигмоиды.

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" \t "_blank)).

**Многоклассовый logloss**

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



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;)