Введение в анализ данных, майнор
Семинар №6

За основу взят аналогичный семинар с курса МО-1 ФКН НИУ ВШЭ

1 Линейные модели классификации

Мы начнём с задачи бинарной классификации, а многоклассовый случай обсудим позже. Пусть $\mathbb{X} = \mathbb{R}^d$ — пространство объектов, $Y = \{-1, +1\}$ — множество допустимых ответов, $X = \{(x_i, y_i)\}_{i=1}^{\ell}$ — обучающая выборка. Иногда мы будем класс «+1» называть положительным, а класс «−1» — отрицательным.

Линейная модель классификации определяется следующим образом:

$$a(x) = \text{sign}\left(\langle w, x\rangle + w_0\right) = \text{sign}\left(\sum_{j=1}^{d} w_j x_j + w_0\right),$$

где $w \in \mathbb{R}^d$ — вектор весов, $w_0 \in \mathbb{R}$ — сдвиг (bias).

Если не сказано иначе, мы будем считать, что среди признаков есть константа, $x_{d+1} = 1$. В этом случае нет необходимости вводить сдвиг w_0, и линейный классификатор можно задавать как

$$a(x) = \text{sign}\langle w, x\rangle.$$

Геометрически линейный классификатор соответствует гиперплоскости с вектором нормали w. Величина скалярного произведения $\langle w, x\rangle$ пропорциональна расстоянию от гиперплоскости до точки x, а его знак показывает, с какой стороны от гиперплоскости находится данная точка. Таким образом, линейный классификатор разделяет пространство на две части с помощью гиперплоскости, и при этом одно полупространство относит к положительному классу, а другое — к отрицательному.

§1.1 Обучение линейных классификаторов

В задаче регрессии имеется континуум возможных ответов, и при таких условиях достаточно странно требовать полного совпадения ответов модели и истинных ответов — гораздо логичнее говорить об их близости. Более того, как мы выяснили, попытка провести функцию через все обучающие точки легко может привести к переобучению. Способов посчитать близость двух чисел (прогноза и истинного ответа) достаточно много, и поэтому при обсуждении регрессии у нас возникло большое количество функционалов ошибки.

В случае с бинарной классификацией всё гораздо проще: у нас всего два возможных ответа алгоритма и, очевидно, мы хотим видеть как можно больше правильных ответов. Соответствующий функционал называется *долей правильных ответов* (accuracy):

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

Нам будет удобнее решать задачу минимизации, поэтому будем вместо этого использовать долю неправильных ответов:

$$Q(a, X) = \frac{1}{\ell}\sum_{i=1}^{\ell}[a(x_i) \neq y_i] = \frac{1}{\ell}\sum_{i=1}^{\ell}[\operatorname{sign}\langle w, x_i\rangle \neq y_i] \to \min_w \tag{1.1}$$

Этот функционал является дискретным относительно весов, и поэтому искать его минимум с помощью градиентных методов мы не сможем. Более того, у данного функционала может быть много глобальных минимумов — вполне может оказаться, что существует много способов добиться оптимального количества ошибок. Чтобы не пытаться решать все эти проблемы, попытаемся свести задачу к минимизации гладкого функционала.

Отступы. Заметим, что функционал (1.1) можно несколько видоизменить:

$$Q(a, X) = \frac{1}{\ell}\sum_{i=1}^{\ell}[\underbrace{y_i\langle w, x_i\rangle}_{M_i} < 0] \to \min_w$$

Здесь возникла очень важная величина $M_i = y_i\langle w, x_i\rangle$, называемая *отступом* (margin). Знак отступа говорит о корректности ответа классификатора (положительный отступ соответствует правильному ответу, отрицательный — неправильному), а его абсолютная величина характеризует степень уверенности классификатора в своём ответе. Напомним, что скалярное произведение $\langle w, x\rangle$ пропорционально расстоянию от разделяющей гиперплоскости до объекта; соответственно, чем ближе отступ к нулю, тем ближе объект к границе классов, тем ниже уверенность в его принадлежности.

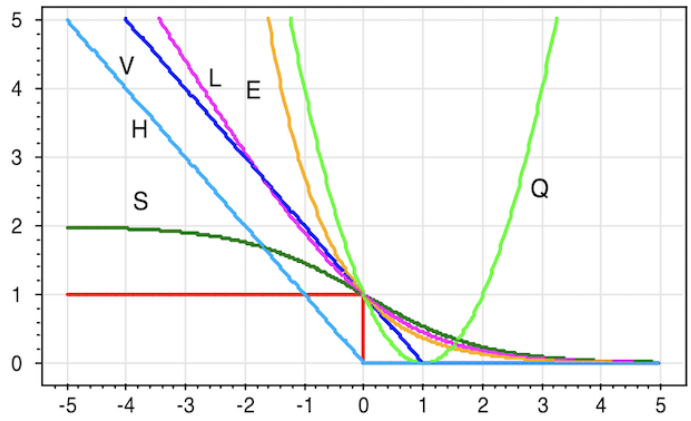

Рис. 1. Верхние оценки на пороговую функцию потерь.

Верхние оценки. Функционал (1.1) оценивает ошибку алгоритма на объекте x с помощью пороговой функции потерь $L(M) = [M < 0]$, где аргументом функции является отступ $M = y\langle w, x\rangle$. Оценим эту функцию сверху (см. рис. 1):

$$L(M) \leqslant \tilde{L}(M).$$

После этого можно получить верхнюю оценку на функционал (1.1):

$$Q(a, X) \leqslant \frac{1}{\ell} \sum_{i=1}^{\ell} \tilde{L}(y_i\langle w, x_i\rangle) \to \min_w$$

Если верхняя оценка $\tilde{L}(M)$ является гладкой, то и данная верхняя оценка будет гладкой. В этом случае её можно будет минимизировать с помощью, например, градиентного спуска. Если верхнюю оценку удастся приблизить к нулю, то и доля неправильных ответов тоже будет близка к нулю.

Приведём несколько примеров верхних оценок:

1. $\tilde{L}(M) = \log\left(1 + e^{-M}\right)$ — логистическая функция потерь
2. $\tilde{L}(M) = (1 - M)_+ = \max(0, 1 - M)$ — кусочно-линейная функция потерь (используется в методе опорных векторов)
3. $\tilde{L}(M) = (-M)_+ = \max(0, -M)$ — кусочно-линейная функция потерь (соответствует персептрону Розенблатта)
4. $\tilde{L}(M) = e^{-M}$ — экспоненциальная функция потерь
5. $\tilde{L}(M) = 2/(1 + e^{M})$ — сигмоидная функция потерь

Любая из них подойдёт для обучения линейного классификатора. Позже мы подробно изучим некоторые из них и выясним, какими свойствами они обладают.

2 Логистическая регрессия

§2.1 Правдоподобие и логистические потери

Хотя квадратичная функция потерь и приводит к корректному оцениванию вероятностей, она не очень хорошо подходит для решения задачи классификации. Причиной этому в том числе являются и слишком низкие штрафы за ошибку — так, если объект положительный, а модель выдаёт для него вероятность первого класса $b(x) = 0$, то штраф за это равен всего лишь единице: $(1 - 0)^2 = 1$.

Попробуем сконструировать функцию потерь из других соображений. Если алгоритм $b(x) \in [0, 1]$ действительно выдает вероятности, то они должны согласовываться с выборкой. С точки зрения алгоритма вероятность того, что в выборке встретится объект x_i с классом y_i, равна $b(x_i)^{[y_i=+1]}(1 - b(x_i))^{[y_i=-1]}$. Исходя из этого, можно записать правдоподобие выборки (т.е. вероятность получить такую выборку с точки зрения алгоритма):

$$Q(a, X) = \prod_{i=1}^{\ell} b(x_i)^{[y_i=+1]}(1 - b(x_i))^{[y_i=-1]}.$$

Данное правдоподобие можно использовать как функционал для обучения алгоритма — с той лишь оговоркой, что удобнее оптимизировать его логарифм:

$$-\sum_{i=1}^{\ell} ([y_i = +1] \log b(x_i) + [y_i = -1] \log(1 - b(x_i))) \to \min$$

Данная функция потерь называется логарифмической (log-loss).

3 Регуляризация

В ряде случаев (признаков больше чем объектов, коррелирующие признаки) оптимизационная задача $Q(w) \to \min$ может иметь бесконечное число решений, большинство которых являются переобученными и плохо работают на тестовых данных. Покажем это.

Пусть в выборке есть линейно зависимые признаки. Это по определению означает, что существует такой вектор v, что для любого объекта x выполнено $\langle v, x\rangle = 0$. Допустим, мы нашли оптимальный вектор весов w для линейного классификатора. Но тогда классификаторы с векторами $w + \alpha v$ будут давать *точно такие же* ответы на всех объектах, поскольку

$$\langle w + \alpha v, x\rangle = \langle w, x\rangle + \alpha \underbrace{\langle v, x\rangle}_{=0} = \langle w, x\rangle.$$

Это значит, что метод оптимизации может найти решение со сколько угодно большими весами. Такие решения не очень хороши, поскольку классификатор будет чувствителен к крайне маленьким изменениям в признаках объекта, а значит, переобучен.

Мы уже знаем, что переобучение нередко приводит к большим значениям коэффициентов. Чтобы решить проблему, добавим к функционалу *регуляризатор*, который штрафует за слишком большую норму вектора весов:

$$Q_\alpha(w) = Q(w) + \alpha R(w).$$

Наиболее распространенными являются L_2 и L_1-регуляризаторы:

$$R(w) = \|w\|_2 = \sum_{i=1}^{d} w_i^2,$$

$$R(w) = \|w\|_1 = \sum_{i=1}^{d} |w_i|.$$

Коэффициент α называется параметром регуляризации и контролирует баланс между подгонкой под обучающую выборку и штрафом за излишнюю сложность. Разумеется, значение данного параметра следует подбирать под каждую задачу.

Отметим, что свободный коэффициент w_0 нет смысла регуляризовывать — если мы будем штрафовать за его величину, то получится, что мы учитываем некие априорные представления о близости целевой переменной к нулю и отсутствии необходимости в учёте её смещения. Такое предположение является достаточно странным. Особенно об этом следует помнить, если в выборке есть константный признак и коэффициент w_0 обучается наряду с остальными весами; в этом случае следует исключить слагаемое, соответствующее константному признаку, из регуляризатора.

Квадратичный (или L_2) регуляризатор достаточно прост в использовании в отличие от L_1-регуляризатора, у которого нет производной в нуле. При этом L_1-регуляризатор имеет интересную особенность: его использование приводит к занулению части весов. Позже мы подробно обсудим это явление.

Обратим внимание на вид решения при использовании L_2-регуляризации вместе со среднеквадратичной ошибкой. В этом случае формулу для оптимального вектора весов можно записать в явном виде:

$$w = (X^T X + \alpha I)^{-1} X^T y.$$

Напомним, что аналитическое решение без L_2-регуляризации выглядит следующим образом:

$$w = (X^T X)^{-1} X^T y.$$

Благодаря добавлению диагональной матрицы к $X^T X$ данная матрица оказывается положительно определённой, и поэтому её можно обратить. Таким образом, при использовании L_2 регуляризации решение всегда будет единственным.

4 Разреженные модели

В процессе обсуждения регуляризации мы упомянули, что использование L_1-регуляризатора приводит к обнулению части весов в модели. Обсудим подробнее, зачем это может понадобиться и почему так происходит.

Модели, в которых некоторые веса равны нулю, называют *разреженными*, поскольку прогноз в них зависит лишь от части признаков. Потребность в таких моделях можно возникнуть по многим причинам. Несколько примеров:

1. Может быть заведомо известно, что релевантными являются не все признаки. Очевидно, что признаки, которые не имеют отношения к задаче, надо исключать из данных, то есть производить *отбор признаков*. Есть много способов решения этой задачи, и L_1-регуляризация — один из них.

2. К модели могут выдвигаться ограничения по скорости построения предсказаний. В этом случае модель должна зависеть от небольшого количества наиболее важных признаков, и тут тоже оказывается полезной L_1-регуляризация.

3. В обучающей выборке объектов может быть существенно меньше, чем признаков (так называемая «проблема $N \ll p$»). Поскольку параметров линейной модели при этом тоже больше, чем объектов, задача обучения оказывается некорректной — решений много, и сложно выбрать из них то, которое обладает хорошей обобщающей способностью. Решить эту проблему можно путём внедрения в процесс обучения априорного знания о том, что целевая переменная зависит от небольшого количества признаков. Такая модификация как раз может быть сделана с помощью L_1-регуляризатора.

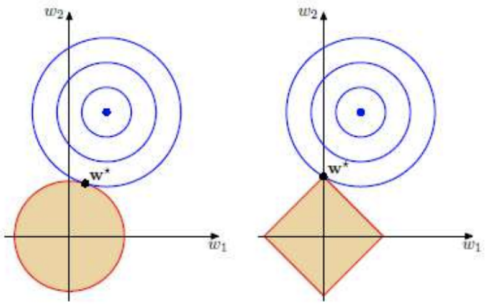

Рис. 2. Линии уровня функционала качества, а также ограничения, задаваемые L_2 и L_1-регуляризаторами.

Теперь, когда мы представляем некоторые области применения разреженных моделей, попробуем понять, почему L_1 регуляризатор позволяет их обучать. Этому есть несколько объяснений.

Угловые точки. Можно показать, что если функционал $Q(w)$ является выпуклым, то задача безусловной минимизации функции $Q(w) + \alpha\|w\|_1$ эквивалентна задаче условной оптимизации

$$\begin{cases} Q(w) \to \min\limits_{w} \\ \|w\|_1 \leqslant C \end{cases}$$

для некоторого C. На рис. 2 изображены линии уровня функционала $Q(w)$, а также множество, определяемое ограничением $\|w\|_1 \leqslant C$. Решение определяется точкой пересечения допустимого множества с линией уровня, ближайшей к безусловному минимуму. Из изображения можно предположить, что в большинстве случаев эта точка будет лежать на одной из вершин ромба, что соответствует решению с одной зануленной компонентой.

Штрафы при малых весах. Предположим, что текущий вектор весов состоит из двух элементов $w = (1, \varepsilon)$, где ε близко к нулю, и мы хотим немного изменить данный вектор по одной из координат. Найдём изменение L_2- и L_1-норм вектора при уменьшении первой компоненты на некоторое положительное число $\delta < \varepsilon$:

$$\|w - (\delta, 0)\|_2^2 = 1 - 2\delta + \delta^2 + \varepsilon^2$$
$$\|w - (\delta, 0)\|_1 = 1 - \delta + \varepsilon$$

Вычислим то же самое для изменения второй компоненты:

$$\|w - (0, \delta)\|_2^2 = 1 - 2\varepsilon\delta + \delta^2 + \varepsilon^2$$
$$\|w - (0, \delta)\|_1 = 1 - \delta + \varepsilon$$

Видно, что с точки зрения L_2-нормы выгоднее уменьшать первую компоненту, а для L_1-нормы оба изменения равноценны. Таким образом, при выборе L_2-регуляризации гораздо меньше шансов, что маленькие веса будут окончательно обнулены.