Лекция 3
Линейная регрессия

Е. А. Соколов
ФКН ВШЭ

20 сентября 2016 г.

1 Переобучение

Мы выработали достаточно общий метод обучения линейных регрессионных моделей, основанный на градиентных методах оптимизации. При этом модель может оказаться *переобученной* — её качество на новых данных может быть существенно хуже качества на обучающей выборке. Действительно, при обучении мы требуем от модели лишь хорошего качества на обучающей выборки, и совершенно не очевидно, почему она должна при этом хорошо *обобщать* эти результаты на новые объекты.

В следующем разделе мы обсудим подходы к оцениванию обобщающей способности, а пока разберём явление переобучения на простом примере. Рассмотрим некоторую одномерную выборку, значения единственного признака x в которой генерируются равномерно на отрезке $[0, 1]$, а значения целевой переменной выбираются по формуле $y = \cos(1.5\pi x) + \mathcal{N}(0, 0.01)$, где $\mathcal{N}(\mu, \sigma^2)$ — нормальное распределение со средним μ и дисперсией σ^2. Попробуем восстановить зависимость с помощью линейных моделей над тремя наборами признаков: $\{x\}$, $\{x, x^2, x^3, x^4\}$ и $\{x, x^2, \ldots, x^{15}\}$. Соответствующие результаты представлены на рис. 1.

Видно, что при использовании признаков высоких степеней модель получает возможность слишком хорошо подогнаться под выборку, из-за чего становится непригодной для дальнейшего использования. Эту проблему можно решать многими способами — например, использовать более узкий класс моделей или штрафовать за излишнюю сложность полученной модели. Так, можно заметить, что у переобученной модели, полученной на третьем наборе признаков, получаются очень большие коэффициенты при признаках. Как правило, именно норма вектора коэффициентов используется как величина, которая штрафуется для контроля сложности модели. Такой подход называется *регуляризацией*, речь о нём пойдёт ниже.

2 Оценивание качества моделей

В примере, о котором только что шла речь, мы не можем обнаружить переобученность модели по обучающей выборке [1]. С другой стороны, если бы у нас были

[1] Конечно, это можно было бы заметить по большим весам в модели, но связь между нормой весов и обобщающей способностью алгоритма неочевидна.

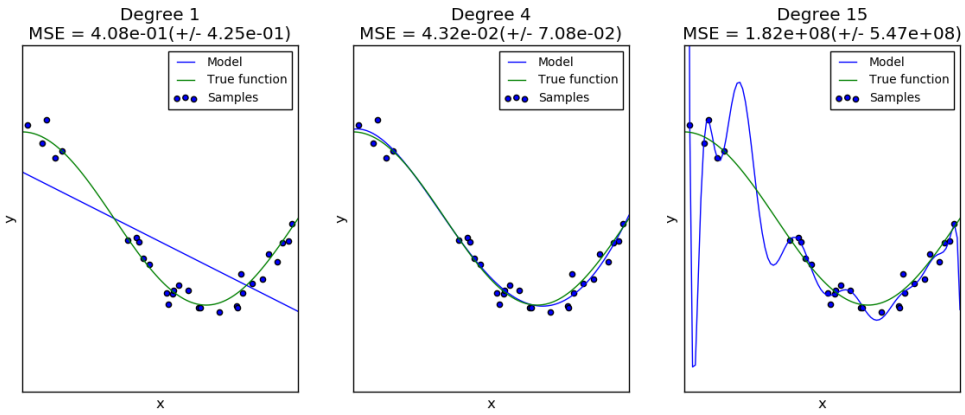

Рис. 1. Регрессионные кривые для признаковых наборов различной сложности.

дополнительные объекты с известными ответами, то по ним заметить низкое качество модели было бы довольно легко.

На данной идее основан подход с *отложенной выборкой*. Имеющиеся размеченные данные (т.е. данные с известными ответами) разделяются на две части: обучающую и контрольную. На обучающей выборке, как это следует из названия, модель обучается, а на контрольной выборке проверяется её качество. Если значение функционала на контрольной выборке оказалось удовлетворительным, то можно считать, что модель смогла извлечь закономерности при обучении.

Использование отложенной выборки приводит к одной существенной проблеме: результат существенно зависит от конкретного разбиения данных на обучение и контроль. Мы не знаем, какое качество получилось бы, если бы объекты из данного контроля оказались в обучении. Решить эту проблему можно с помощью *кросс-валидации*. Размеченные данные разбиваются на k блоков $X_1, \ldots, X_k$ примерно одинакового размера. Затем обучается k моделей $a_1(x), \ldots, a_k(x)$, причём i-я модель обучается на объектах из всех блоков, кроме блока i. После этого качество каждой модели оценивается по тому блоку, который не участвовал в её обучении, и результаты усредняются:

$$\text{CV} = \frac{1}{k}\sum_{i=1}^{k} Q\left(a_i(x), X_i\right).$$

3 Регуляризация

Выше мы упоминали, что если матрица $X^T X$ не является обратимой, то с оптимизацией среднеквадратичной ошибки могут возникнуть некоторые трудности. Действительно, 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.

4 Гиперпараметры

В машинном обучении принято разделять подлежащие настройке величины на *параметры* и *гиперпараметры*. Параметрами называют величины, которые настраиваются по обучающей выборки — например, веса в линейной регрессии. К гиперпараметрам относят величины, которые контролируют сам процесс обучения и не могут быть подобраны по обучающей выборке.

Хорошим примером гиперпараметра является коэффициент регуляризации α. Введение регуляризации мешает модели подгоняться под обучающие данные, и с

точки зрения среднеквадратичной ошибки выгодно всегда брать $\alpha = 0$. Разумеется, такой выбор не будет оптимальным с точки зрения качества на новых данных, и поэтому коэффициент регуляризации (как и другие гиперпараметры) следует настраивать по отложенной выборке или с помощью кросс-валидации.

При подборе гиперпараметров по кросс-валидации возникает проблема: мы используем отложенные данные, чтобы выбрать лучший набор гиперпараметров. По сути, отложенная выборка тоже становится обучающей, и показатели качества на ней перестают характеризовать обобщающую способность модели. В таких случаях выборку, на которой настраиваются гиперпараметры, называют валидационной, и при этом выделяют третий, тестовый набор данных, на которых оценивается качество итоговой модели.

5 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.

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

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

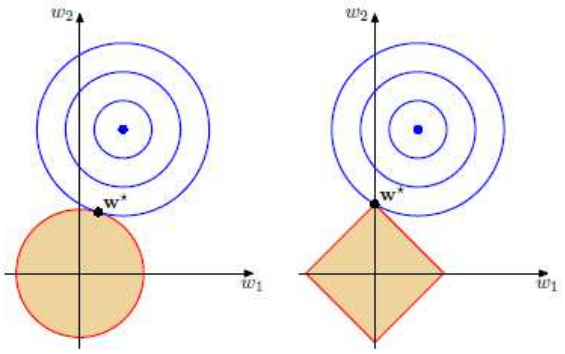

Рис. 2. Линии уровня функционала качества, а также ограничения, задаваемые L_2 и L_1-регуляризаторами.

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.

Проксимальный шаг. *Проксимальные методы* — это класс методов оптимизации, которые хорошо подходят для функционалов с негладкими слагаемыми. Не будем сейчас останавливаться на принципах их работы, а приведём лишь формулу для шага проксимального метода в применении к линейной регрессии с квадратичным функционалом ошибки и L_1-регуляризатором:

$$w^{(k)} = S_{\eta\alpha}\left(w^{(k-1)} - \eta\nabla_w F(w^{(k-1)})\right),$$

где $F(w) = \|Xw - y\|^2$ — функционал ошибки без регуляризатора, η — длина шага, α — коэффициент регуляризации, а функция $S_{\eta\alpha}(w)$ применяется к вектору весов

покомпонентно, и для одного элемента выглядит как

$$S_{\eta\alpha}(w_i) = \begin{cases} w_i - \eta\alpha, & w_i > \eta\alpha \\ 0, & |w_i| < \eta\alpha \\ w_i + \eta\alpha, & w_i < -\eta\alpha \end{cases}$$

Из формулы видно, что если на данном шаге значение некоторого веса не очень большое, то на следующем шаге этот вес будет обнулён, причём чем больше коэффициент регуляризации, тем больше весов будут обнуляться.

6 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. Чем больше здесь τ, тем выше штраф за завышение прогноза.

На семинарах будет показано, что если алгоритм должен выдавать константу, то при таком функционале оптимально будет возвращать τ-квантиль по всем ответам. Сейчас же обсудим вероятностный смысл данного функционала. Будем считать, что в каждой точке $x \in \mathbb{X}$ пространства объектов задано вероятностное распределение $p(y \mid x)$ на возможных ответах для данного объекта. Такое распределение может возникать, например, в задаче предсказания кликов по рекламным баннерам: один и тот же пользователь может много раз заходить на один и тот же сайт и видеть данный баннер; при этом некоторые посещения закончатся кликом, а некоторые — нет.

Известно, что при оптимизации квадратичного функционала алгоритм $a(x)$ будет приближать условное матожидание ответа в каждой точке пространства объектов: $a(x) \approx \mathbb{E}[y \mid x]$; если же оптимизировать среднее абсолютное отклонение, то итоговый алгоритм будет приближать медиану распределения: $a(x) \approx \text{median}[p(y \mid x)]$. Рассмотрим теперь некоторый объект x и условное распределение $p(y \mid x)$. Найдем число q, которое будет оптимальным с точки зрения нашего функционала:

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

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

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

Получаем, что

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

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

7 Преобразования признаков

§7.1 Кодирование категориальных признаков

В линейных моделях предполагается, что признаки являются числовыми — без этого домножение на веса и суммирование не будут нести в себе никакого смысла. Для категориальных же признаков эти операции не определены, и поэтому требуется проводить преобразование таких признаков в числовые. Один из самых простых способов — это бинаризация, или one-hot-кодирование.

Допустим, категориальный признак $f_j(x)$ принимает значения из множества $C = \{c_1, \ldots, c_m\}$. Заменим его на m бинарных признаков $b_1(x), \ldots, b_m(x)$, каждый из которых является индикатором одного из возможных категориальных значений:

$$b_i(x) = [f_j(x) = c_i].$$

Отметим, что признаки $b_1(x), \ldots, b_m(x)$ являются линейно зависимыми: для любого объекта выполнено

$$b_1(x) + \cdots + b_m(x) = 1.$$

Чтобы избежать этого, можно выбрасывать один из бинарных признаков. Впрочем, такое решение имеет и недостатки — например, если на тестовой выборке появится новая категория, то её как раз можно закодировать с помощью нулевых бинарных признаков; при удалении одного из них это потеряет смысл.

§7.2 Нелинейные признаки

С помощью линейной регрессии можно восстанавливать нелинейные зависимости, если провести преобразование признакового пространства:

$$x = (x_1, \ldots, x_d) \to \varphi(x) = (\varphi_1(x), \ldots, \varphi_m(x)).$$

Например, можно перейти к квадратичным признакам:

$$\varphi(x) = (x_1, \ldots, x_d, x_1^2, \ldots, x_d^2, x_1x_2, \ldots, x_{d-1}x_d).$$

Линейная модель над новыми признаками уже сможет приближать любые квадратичные закономерности. Аналогично можно работать и с полиномиальными признаками более высоких порядков.

Возможны и другие преобразования:

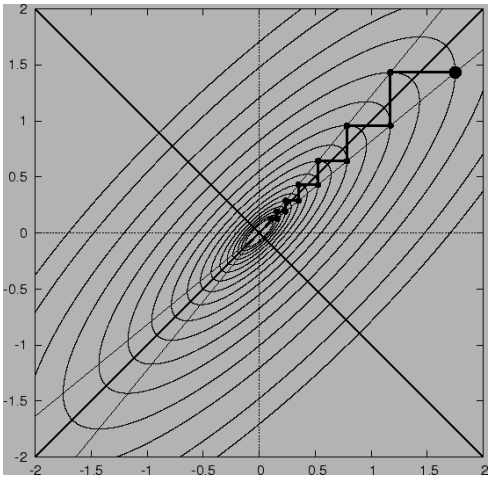

Рис. 3. Траектория градиентного спуска на функционале при признаках разного масштаба.

- $\log x_j$ — для признаков с тяжёлыми хвостами
- $\exp(\|x - \mu\|^2/\sigma)$ — для измерения близости до некоторой точки
- $\sin(x_j/T)$ — для задач с периодическими зависимостями

§7.3 Масштабирование

При обучении линейных моделей полезно масштабировать признаки, то есть приводить их к единой шкале. Разберёмся, зачем это нужно.

Рассмотрим функцию $f_1(x) = x_1^2 + x_2^2$, выберем начальное приближение $x^{(0)} = (1, 1)$ и запустим из него градиентный спуск. Окажется, что за один шаг мы сможем сразу попасть в точку минимума.

Теперь «растянем» функцию вдоль одной из осей: $f_2(x) = 100x_1^2 + x_2^2$. При таком же начальном приближении $x^{(0)}$ антиградиент на первой итерации будет равен $(-100, -1)$, и попасть по нему в минимум уже невозможно — более того, при неаккуратном выборе длины шага можно очень далеко уйти от минимума. Пример траектории градиентного спуска при такой форме функции можно найти на рис. 3.

Аналогичная проблема возникает с функционалом ошибки в линейной регрессии, если один из признаков существенно отличается по масштабу от остальных. Чтобы избежать этого, признаки следует масштабировать — например, путём стандартизации:

$$x_{ij} := \frac{x_{ij} - \mu_j}{\sigma_j},$$

где $\mu_j = \frac{1}{\ell}\sum_{i=1}^{\ell} x_{ij}$, $\sigma_j = \frac{1}{\ell}\sum_{i=1} \ell(x_{ij} - \mu_j)^2$. Или, например, можно масштабировать признаки на отрезок $[0, 1]$:

$$x_{ij} := \frac{x_{ij} - \min_i x_{ij}}{\max_j x_{ij} - \min_j x_{ij}}.$$