Машинное обучение в финансах

Лекция 9: Линейная регрессия. Лассо. Ридж. Нелинейная регрессия

Роман В. Литвинов*

*CRO
Финансовая Группа БКС

Высшая школа экономики, Апрель 2021

Содержание

Содержание

Простая линейная регрессия. RSS. RSE. R^2

На шестой лекции мы говорили с вами о том, что задачи ML, связанные с регрессией, представляют собой поиск функции $f : \mathbb{R}^n \to \mathbb{R}$.

Сегодня мы обсудим одну из таких регрессионных моделей, которая базируется на предположении о том, что f представляет собой линейную функцию (от одной или нескольких переменных). Такая модель называется линейной регрессией (linear regression).

Линейная регрессия представляет собой, пожалуй, самый простой пример обучения на размеченных данных (supervised learning).

При этом это достаточно мощная и хорошо интерпретируемая модель, которая довольно часто используется в финансовых задачах.

В сочетание с техниками лассо и ридж, она используется (довольно успешно) и в задачах связанных с прогнозирование форвардной доходности акций (Zoonekynd, et al, 2016).

Простая линейная регрессия. RSS. RSE. R^2

Применительно к задачам, стоящим перед нами в рамках этого курса, линейная регрессия подразуемевает, что форвардная доходность (r) равна:

$$r = \beta_0 + \sum_{i=1}^{n} \beta_i Factor_i + \epsilon$$

ϵ - представляет собой остаточный шум, нормальную случайную переменную со средним ноль.

В случае, если мы имеем дело с простой регрессией (simple linear regression), то предсказываемая переменная линейно зависит от одного предиктора/фактора и задача сводится к:

$$r = \beta_0 + \beta_1 Factor + \epsilon$$

Простая линейная регрессия. RSS. RSE. R^2

Наш прогноз доходности с использованием такой модели при заданном значении фактора равном x выглядит следующим образом:

$$\hat{r} = \beta_0 + \beta_1 x$$

ϵ - в таком случае, ни что иное как:

$$\epsilon = r - \hat{r}$$

На седьмой лекции мы с вами говорили о том, что оценку производительности модели, можно базировать на том, насколько сильно прогноз отклоняется от реально наблюдаемой величины (тогда мы рассматривали MSE в качестве критерия).

Сейчас идея останется той же (оценка 'расстояния'), а ее реализация/воплощение будет еще проще.

Простая линейная регрессия. RSS. RSE. R^2

Пусть ϵ_i представляет собой остаточный ошибочный член (residual error) - разницу между i-тым наблюдением прогнозируемой величины и ее прогнозом:

$$\epsilon_i = r_i - \hat{r}_i$$

В качестве простой меры производительности нашей модели мы можем использовать сумму квадратов этих ошибок (residual sum of squares, RSS):

$$RSS = \sum_{i=1}^{n} \epsilon_i^2 = \sum_{i=1}^{n} (r_i - \hat{r}_i)^2 = \sum_{i=1}^{n} (r_i - \beta_0 + \beta_1 x)^2$$

Понятно, что чем меньше RSS тем качественнее модель (меньше ошибки). По мере приближения к 'идеальной модели', RSS будет стремиться к нулю.

Простая линейная регрессия. RSS. RSE. R^2

Наша задача подобрать такие коэффициенты β_0 и β_1, которые бы минимизировали RSS.

Чем показательна простая линейная регрессия, так этот тем, что для нее существуют решение в закрытой форме для этой задачи (чем не всегда могут похвастаться более сложные методы). Минимум достигается при следующих значениях коэффициентов:

$$\beta_0 = \bar{y} - \beta_1 \bar{x}$$

$$\beta_1 = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^{n}(x_i - \bar{x})^2}$$

где $\bar{x}$ и $\bar{y}$ среднее значение y и x, соответственно.

Простая линейная регрессия. RSS. RSE. R^2

Как мы с вами говорили выше, для хорошей модели RSS будем небольшим числом и для 'идеальной' стремиться к нулю. Но в целом, однозначно интерпретировать качество модели с помощью RSS иногда бывает затруднительно. Понятно, что 'много' это плохо, но насколько плохо? И действительно ли мы уже приблизились к отметке 'много'?

Для целей более простой интрепретации придуманы показатели RSE (residual standard error) и R^2.

RSE характеризует среднее значение на которое прогноз отличается от результата и вычисляется по формуле:

$$RSE = \sqrt{\frac{1}{n-2}RSS} = \sqrt{\frac{1}{n-2}\sum_{i=1}^{n}(r_i - \hat{r}_i)^2}$$

Простая линейная регрессия. RSS. RSE. R^2

Пусть TSS (total sum of squares) представляет собой следующую величину, которая характеризует разброс данных вокруг среднего:

$$TSS = \sum_{i=1}^{n}(y_i - \hat{y})$$

Разницу между TSS и RSS можно трактовать, как часть информации о разбросе предсказываемой переменной, которую 'объясняет' модель.

Коэффициент R^2 будет тогда представлять собой пропорцию информации (о разбросе/дисперсии), которая объяснена моделью:

$$R^2 = \frac{TSS - RSS}{TSS} = 1 - \frac{RSS}{TSS}$$

R^2 (обычно!!) принимает значение от 0 до 1, где 1 соотвествует 'идеальной' модели. Значение коэффициента равное 0.4 означает, что модель объясняет только 40 процентов разброса/ дисперсии данных.

Содержание

Множественная линейная регрессия

На практике мы часто сталкиваемся с истуацией, когда предсказываемая переменная зависит (линейно) более чем от одного предиктора (фактора). В таком случае, мы имеем дело с множественой линейной регрессией (multiple linear regression).

В общем виде множественная линейно-регрессионная модель имеет вид:

$$r = \beta_0 + \sum_{i=1}^{n} \beta_i Factor_i + \epsilon$$

Соотвествующий коэффициент β_i можно интрпретировать как 'вес', с которым входит в модель фактор F_i, - средний эффект, который оказывает изменение фактора F_i на прогнозируемую переменную при условии того, что значения остальных предикторов не изменяются.

Множественная линейная регрессия

Значение RSS в таком случае будет равно:

$$RSS = \sum_{j=1}^{m} \epsilon_j^2 = \sum_{j=1}^{m} (r_j - \hat{r}_j)^2 = \sum_{j=1}^{m} (r_j - \beta_0 - \sum_{i=1}^{n} \beta_i Factor_i)^2$$

Так же как и в задаче простой линейной регрессии нам требуется подобрать такие значение коэффициентов β_i, которые бы минимизировали значение RSS.

На языке линейной алгебры эта задача сводится к решению следующей проблемы (более подробно см. Deisenroth, 2020):

$$\beta = (X^T X)^{-1} X^T r$$

где X - матрица, содержащая значения предикторов/факторов, r - вектор предсказываемых значений и β - вектор, содержащий оптимальные значения весов.

Множественная линейная регрессия

Основной проблемной точкой этой системы является существование и единственность (!) обратной матрицы $(X^T X)^{-1}$.

Такая обратная матрица будет уникальной только в случае, если:

- никакой из предикторов/факторов не является линейной комбинацией других факторов;
- количество предикторов меньше чем количество элементов выборки предсказываемых значений.

На практике п.1 релизуется за счет удаления коллинеарных (коррелированных) предикторов. П.2 за счет сокращения количества/ компрессии факторов.

NB: линейную регрессию можно использовать для коррелированных факторов, просто при этом решение не будет единственным (!) (лишь одним из нескольких).

Множественная линейная регрессия

Что касательно оценки производительности, то коэффициенты RSE и R^2 имеют для множественной регрессии тот же смысл и считаются по аналогичным формулам.

Единственное уточнение это обобщение формулы RSE для случая, когда имеются m-предикторов (При m = 1 формула имеет вид, который мы уже видели выше):

$$RSE = \sqrt{\frac{1}{n-m-1}RSS}$$

Важно отметить следующий эффект: модель с бОльшим количеством переменных будет иметь более высокое значение RSE в случаях, когда добавление этих дополнительных факторов, дает незначительное сокращение RSS (лучше это представить на примере моделей с одинаковым значением RSS, но разным количеством факторов).

Содержание

Ридж-регрессия и регуляризация

Очень часто основной проблемой, связанной со сложностью интерпретации модели и нестабильностью ее работы, является наличие большого количества переменных в финальном регрессионном уравнении, которое мы получили после обучения модели. Например, порядка 17 коэффициентов с ненулевыми весами.

При этом, многие из этих коэффициентов будут иметь веса близкие к нулю и оказывать незначительное влияние на прогнозируемую переменную, но существенно затруднять интерпретацию модели.

Для борьбы с подобными проблемами придуманы техники регуляризации (regularization) или, как их еще называют, сжатия (shrinkage). Общий смысл подобных техник - это введение некого механизма штрафов на количество ненулевых переменных в модели.

Основные представители подобного класса моделей - ридж-регрессия и лассо-регрессия.

Ридж-регрессия и регуляризация

Ридж-регрессия или, как ее еще называют, гребневая регрессия представляет собой модель линейной регрессии, базирующуюся на механизме наименьших квадратов с одной небольшой модификацией.

Как мы с вами помним поиск оптимальных весов в линейной регрессии базируется на минимизации суммы наименьших квадратов ошибок (RSS).

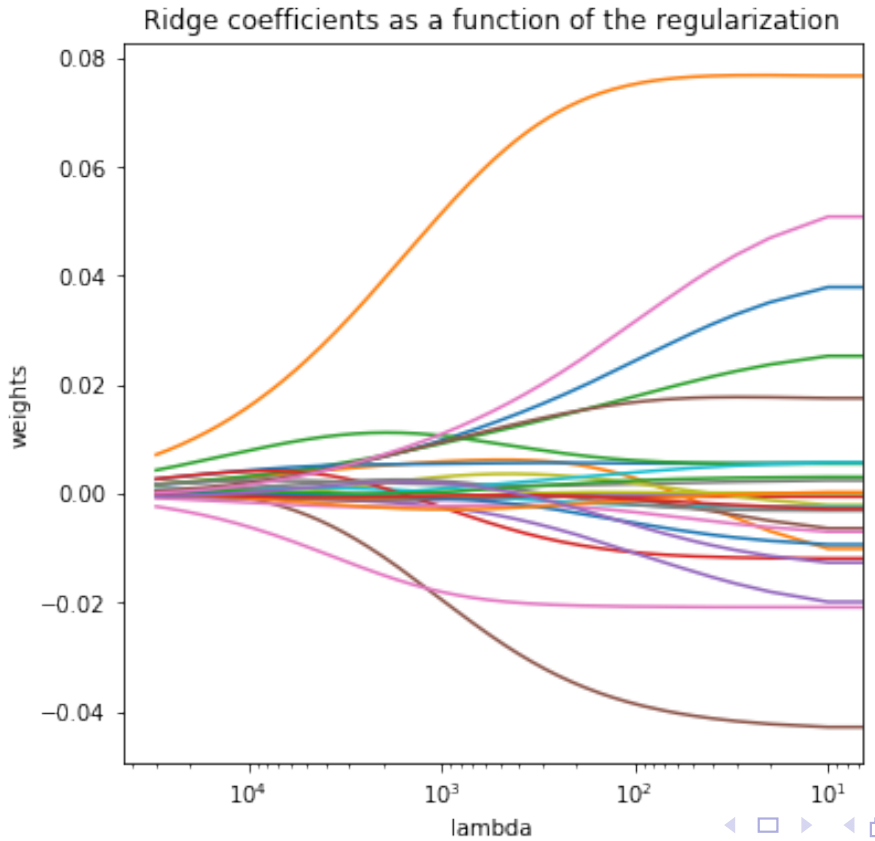
Ридж использует тот же критерий, но добавляет в уравнение второй член, который штрафует модель за каждый ненулевой коэффициент β (вес переменной).

$$RSS_{ridge} = \sum_{j=1}^{m}(r_j - \hat{r}_j)^2 + \lambda \sum_{j=1}^{m} \beta_j^2 = RSS + \lambda \sum_{j=1}^{m} \beta_j^2$$

Параметр λ называется параметром регуляризации, а член уравнения $\lambda \sum_{j=1}^{m} \beta_j^2$ - штрафом за сжатие (shrinkage penalty).

Ридж-регрессия и регуляризация

Теперь поиск оптимальных весов ведется не только с прицелом на минимизацию отклонений между прогнозом и результатом, но и на количество ненулевых весов в модели. Чем больше параметр λ, тем больше модель ориентируется на штраф связанный с весами переменных, нежели на ошибки в прогнозе.



Содержание

Лассо-регрессия

Лассо-регрессия представляет собой похожую на ридж схему, в которой используется следующий вид штрафа:

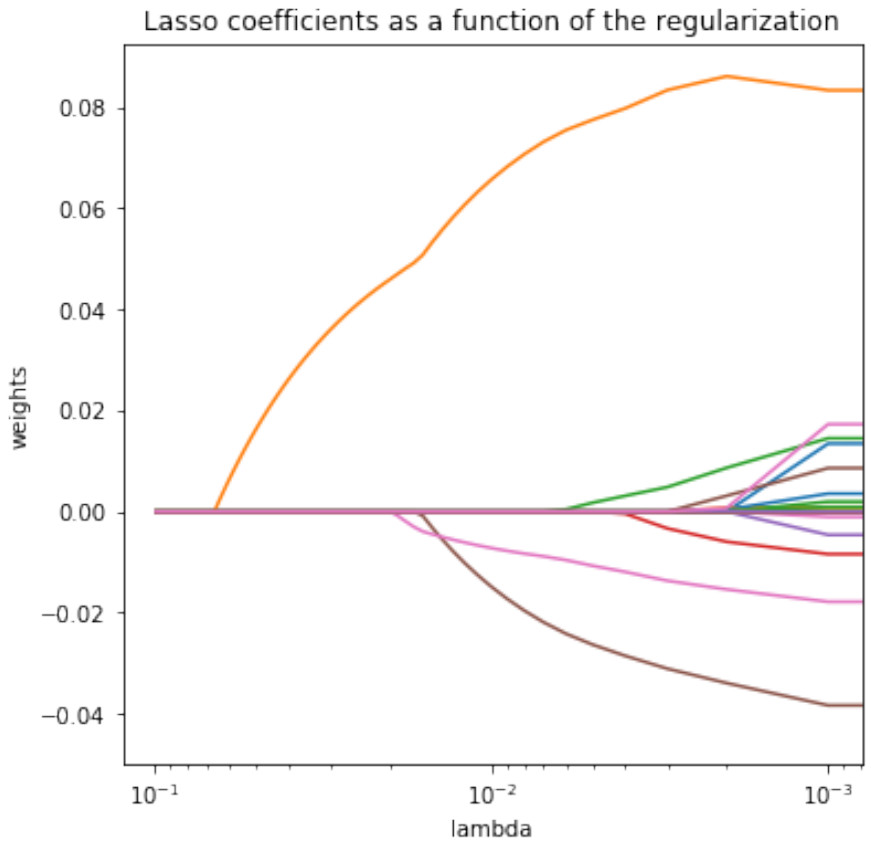
$$RSS_{ridge} = \sum_{j=1}^{m}(r_j - \hat{r}_j)^2 + \lambda \sum_{j=1}^{m} |\beta_j| = RSS + \lambda \sum_{j=1}^{m} |\beta_j|$$

Важно помнить следующее основное отличие между ридж и лассо. По техническим причинам, в которые мы не будем погружаться сейчас, ридж не присваивает нулевые веса предикторам (формально, нулевые веса возможны только при $\lambda = \infty$, что на практике невозможно т.к. λ может принимать большие, но все же не бесконечные значения). Веса в ридж-регрессии будут стремиться к нулю по мере роста λ.

Что касательно лассо, то такой вид регрессии допускает нулевые значения переменных по мере увеличения параметра регуляризации. Что приводит к т.н. разряженным (sparse) моделям - хорошо интерпретируемым за счет небольшого количества переменных.

Лассо-регрессия

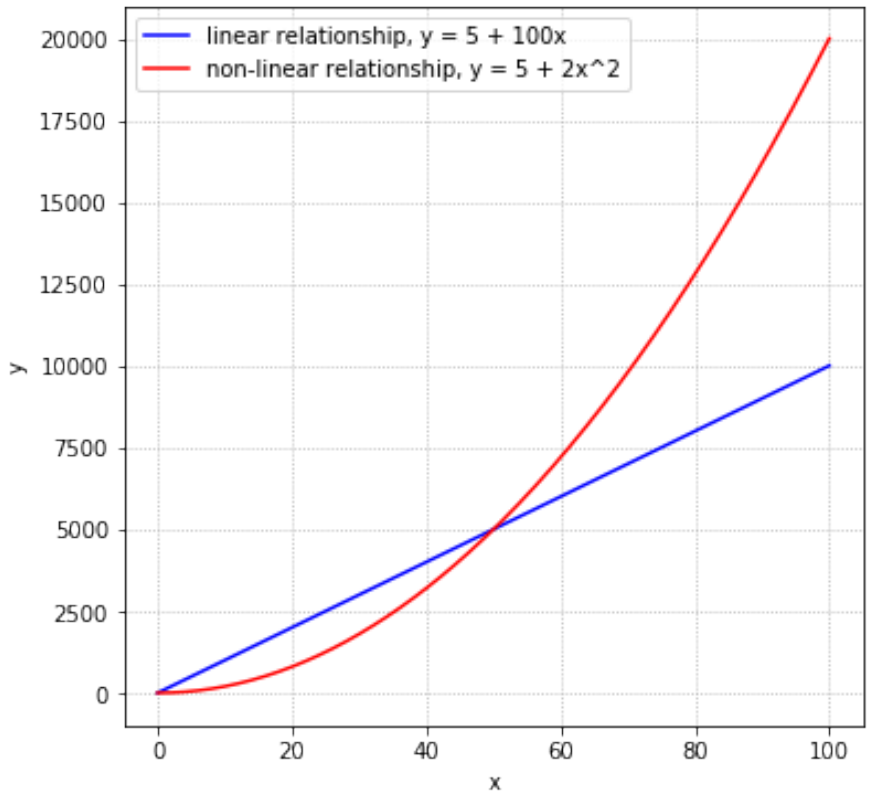
Ниже проиллюстрирован пример работы лассо. Как вы видите, по мере роста λ алгоритм довольно быстро и агрессивно присваивает нулевые веса переменным и выводит их из игры.



Содержание

Регрессия и нелинейные зависимости

Основной недостаток линейных регрессионных моделей - это невозможность построения моделей с нелинейными зависимостями переменных.



Регрессия и нелинейные зависимости

Однако, есть очень простой и зачастую эффективный способ это исправить - полиномиальная регрессия (polynomial regression).

Суть в том, что мы используем тоже самое уравнение линейной регресии, но трансформируем факторы. Для этого мы берем первоначальные факторы и возводим их в степень (квадрат, куб и тп).

Затем мы с ипользованием трансформированных факторов калибруем модель линейной регрессии.

На примере модели линейной регрессии с одним фактором и квадратичной зависимостью между прогнозируемой величиной и фактором, наше итоговое уравнение может иметь вид:

$$r = \beta_0 + \beta_1 Factor_1 + \beta_2 Factor_1^2$$

Литература

Дополнительная литература к сегодняшней лекции:

- Albon, C. (2018) Machine learning with Python cookbook. O'Reilly.
- Deisenroth, M. Fisal, A., Ong, C.S. (2020) Mathematics for machine learning. Cambridge.
- Hastie, T., Tibshirani, R., Friedman, J. (2017). The elements of statistical learning. Springer. (есть на русском)
- James, G., Witten, D., Hastie, T., Tibshirani, R. (2013). An introduction to statistical learning. Springer. (есть на русском)
- Kuhn, M., Johnson, K., (2016). Applied predictive modeling. Springer. (есть на русском)
- Zoonekynd, V., Lau, A., Sambatur, H., LeBinh, K. (2016) Machine learning in finance. Deutsche Bank Market Research.