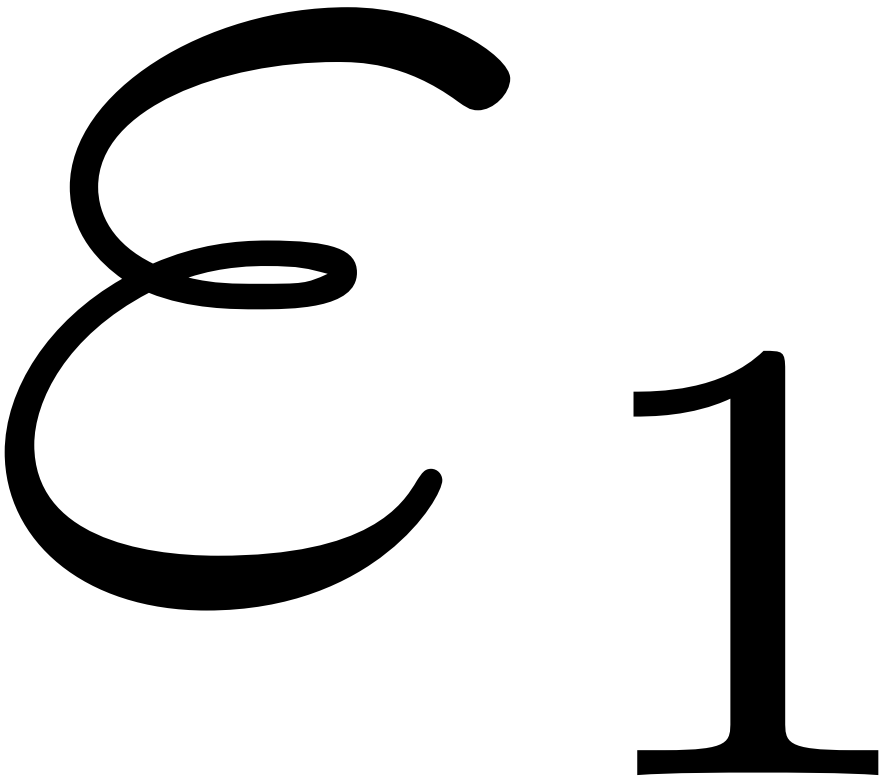


Э П С И Л О Н

E P S I L O N

журнал об эконометрике
и не только о ней

Вступительное слово

Замечательные, но пока ещё малознакомые читатели!

Вы, наверное, не раз заглядывали в научные журналы и могли заметить особенности академического языка. Статьи в такие журналы принято писать обезличенно, в страдательном залоге: было сделано, было проведено. Если вы ещё и писали такие статьи, то знаете, как хочется иногда выражаться хоть немножко человечнее: «я сделал», «мы провели», «а давайте-ка попробуем вот это». В прошлом году я попробовал (весьма сдержанно — куда сдержаннее, чем пишу сейчас) и получил ответ рецензента: стиль не соответствует нормам академического языка.

Ещё от научных статей требуют новизну. Например, если вы напишете про коэффициент ранговой корреляции Спирмена, то вам могут сказать: незачем, про это уже Спирмен написал. А если хочется?

Мы сделали журнал, в котором можно опубликовать статью без строгого академического стиля и научной новизны — лишь бы хорошая была. Как мы будем определять, хорошая ли статья? Попробуем как-нибудь разобраться. Если случилось так, что автор ошибся, мы можем предложить ему исправить ошибку, можем вставить своё примечание, а можем просто этого не заметить. Если заметите вы — пишите нам.

Собственно, статьи в этом журнале вообще не обязаны быть научными — приветствуются и методические, и дидактические работы. Конечно, это чревато тем, что наши авторы будут терять умеренность и писать всякую ерунду — так ведь чуточку побаловаться можно!

Архив журнала доступен на страничке http://bdemeshev.github.io/epsilon/. Замечательные читатели, пишите нам смелее по электронному адресу kuznesashka@gmail.com! Предлагайте свои статьи, комментируйте чужие, мы будет очень рады :)

Как выбрать функциональную форму уравнения регрессии

Кирилл Фурманов*

21 июня 2015 г.

Аннотация

Иногда при оценивании регрессии возникают вопросы: брать переменные в логарифмах или нет? Как узнать, нужно ли изменить свой выбор? Какие существуют в данном случае диагностические тесты? Данная статья предлагает несколько способов принятия решения до и после оценивания.

Ключевые слова: функциональная форма, диагностика остатков, нормальность остатков.

Если человек, начинающий изучать статистику, задаст вопрос, что же такое регрессионный анализ и чем он отличается от анализа корреляционного, он может получить такой ответ: и то и другое связано с изучением статистических связей, но задача корреляционного анализа — измерение тесноты связи между признаками, а задача регрессионного анализа — определение формы этой связи. По мере продолжения обучения этот человек начнёт, вероятно, подозревать, что форма связи практически одна — линейная. Иногда какие-нибудь переменные логарифмируются — то ли для разнообразия, то ли для приличия.

Более того, изучение статей с применением регрессионных моделей может подтвердить эти подозрения. В них исследователи излагают истории своей борьбы с гетероскедастичностью, мультиколлинеарностью, эндогенностью, но редко внимание сосредотачивается на определении функциональной формы зависимости. И даже когда встречаются существенно нелинейные модели, часто остаётся вопрос: почему у зависимости именно такой вид? Кажется, они приходят авторам откуда-то свыше — с потолка, что ли.

А ведь задача определения вида функции регрессии — самая существенная, если отрешиться от содержательной стороны дела и рассматривать только

*Кафедра математической экономики и эконометрики, НИУ ВШЭ, Москва.

статистические проблемы. Все статистические выводы имеют смысл, только если форма зависимости выбрана правильно — без этого сомнительна польза дальнейших изысканий, не спасают ни бутстрап, ни самый обобщённый метод моментов.

Итак, есть повод задуматься над двумя вопросами. Первый: как определить функциональную форму зависимости? Второй: почему так популярны линейные и линейные в логарифмах модели? Должно быть, когда-то они были популярны из-за простоты, но сейчас нет особых проблем в подгонке нелинейных зависимостей: процедуры подгонки реализованы в доступных статистических программах.

Пока что оставим эти вопросы читателю на обдумывание и возьмёмся за проблему с другого конца. Давайте сначала представим, что у нас уже есть полностью специфицированное уравнение регрессии. Более того, оно уже оценено, и теперь стоит вопрос: было ли оно правильным? Может, мы пытались подогнать линейную зависимость под данные, порождённые нелинейным процессом? Может, мы прологарифмировали все переменные, хотя в этом не было нужды? Как это выяснить?

1 Визуальная диагностика

Если оценивается парная регрессия, ответ очевиден: нужно построить график (пожалуй, это стоило сделать ещё до оценивания). По диаграмме рассеяния часто легко догадаться, какую зависимость стоит подгонять. Может быть, график нас всё же смущает — например, там недостаточно наблюдений, чтобы уверенно сделать выбор. Тогда на помощь полезно призвать теорию изучаемого явления, устоявшиеся традиции моделирования и здравый смысл. Если и это не помогло, стоит выбрать вариант попроще. Кроме того, если у нас есть две модели, из которых сложно выбрать одну, то мы можем сравнить выводы, полученные из обеих. Если они схожи — прекрасно, не надо и выбирать. Если они различаются, то мы хотя бы знаем, насколько результаты зависят от выбора модели.

Для множественной регрессии уже не получится на одном графике представить все переменные, но унывать не стоит. Есть несколько полезных способов проверки выбранной функциональной формы.

1.1 График «остатки — прогнозы»

После оценивания модели постройте график зависимости остатков от модельных (прогнозных) значений объясняемой величины. В идеальном случае график не выявит никакой зависимости (см. рис. 1.1). При правильной спецификации остатки должны вести себя беспорядочно — если мы обнаружим какую-то связь между прогнозными значениями и остатками, то её можно

использовать для улучшения функциональной формы. На рис. 1.2 показан пример графика, полученного при подгонке линейной зависимости под данные, порождённые квадратичной моделью $y = \beta_1 + \beta_2 x + \beta_3 x^2 + \varepsilon$. Видно, что остатки разбросаны вдоль параболы. Для усиления наглядности можно дополнить график сглаженной зависимостью остатков от прогнозов — линией, полученной методом lowess или ядерной регрессией. Можно вместо остатков откладывать по вертикальной оси наблюдаемые значения объясняемой переменной — тогда в идеальном случае график будет выглядеть как облако точек, рассеянное вдоль биссектрисы прямого угла — линии $y = \hat{y}$.

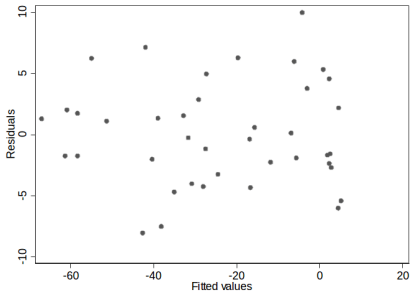


Рис. 1.1. Верная спецификация

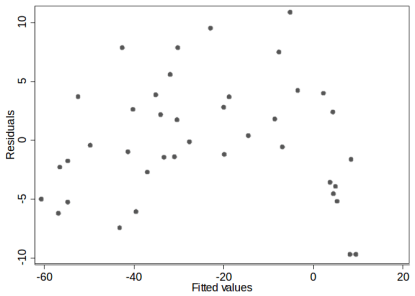


Рис. 1.2. Пропущенный квадратичный член

Рис. 1. График «остатки — прогнозы» при пропуске переменной

Есть также смысл смотреть на график зависимости остатков от каждого из регрессоров: это может натолкнуть на мысль, где́ именно кроется ошибка. Стоит только помнить об опасности множественных проверок: при большом количестве объясняющих переменных график для какой-нибудь из них может оказаться подозрительным просто по случайности.

Часто на графике «остатки — прогнозы» отражается не только рассеяние остатков вдоль кривой, но и усиление или уменьшение их разброса. Рис. 2.1 соответствует случаю, в котором линейная модель $y = \beta_1 + \beta_2 x + \varepsilon$ подгонялась под данные, порождённые логарифмической моделью $\ln y = \beta_1 + \beta_2 \ln x + \varepsilon$. Стоит отличать этот случай от того, в котором остатки имеют непостоянный разброс, но при этом рассеяны вдоль прямой — горизонтальной оси (рис. 2.2). Изменение разброса без изменения среднего уровня остатков свидетельствует о гетероскедастичности (непостоянстве дисперсии случайной составляющей), а не об ошибке функциональной формы. Это тоже проблема, но последствия и способы решения у неё другие.

Возможно, посмотрев на рис. 2, вы не найдёте явных различий между двумя графиками. Увы, понять, чем же именно грешит оценённая модель, часто нелегко.

Основной недостаток всех визуальных тестов — это то, что результат за-

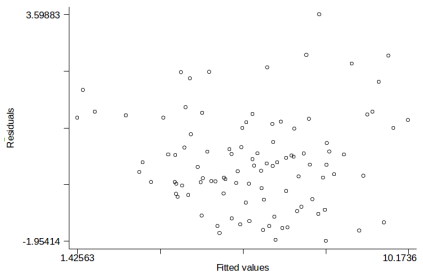


Рис. 2.1. Неверная функциональная форма

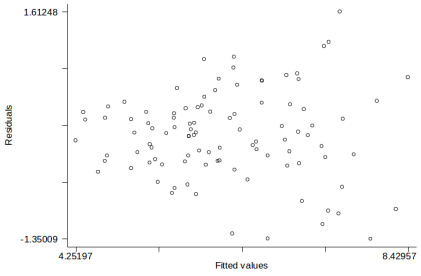


Рис. 2.2. Гетероскедастичность

Рис. 2. График «остатки — прогнозы» при непостоянном разбросе

висит от воображения смотрящего. Посмотрите на отсканированный лист из экзаменационной работы по эконометрике Дмитрия Арсютина (рис. 3). Отмеченные экзаменуемым проблемные явления вовсе не были задуманы автором задания; не заметил их и никто другой из студентов, писавших тот же экзамен.

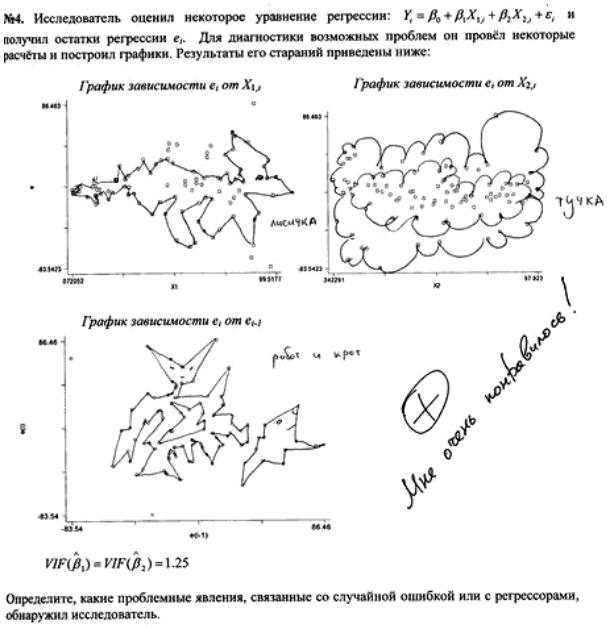


Рис. 3. Пример вольной интерпретации графика «остатки — прогнозы».

2 Диагностические тесты

2.1 Тест Рамсея

Он же RESET — Regression Equation Specification Error Test[1]. Рассчитайте несколько степеней прогнозных значений $\hat{y}$ и проверьте, не улучшает ли результат их включение в изначальную модель. Например, можно включить квадрат и куб прогнозов:

$$y = \beta_1 + \beta_2 x + \ldots + \beta_k x_k + \gamma_2 \hat{y}^2 + \gamma_3 \hat{y}^3 + \varepsilon$$

Если первоначальная модель верно специфицирована, то добавленные нелинейные члены должны быть незначимы, что соответствует основной гипотезе теста.

$$\mathcal{H}_0 \colon \gamma_2 = \gamma_3 = 0, \quad \mathcal{H}_A \colon \gamma_2^2 + \gamma_3^2 > 0 \text{ (ошибка спецификации)}.$$

Это гипотеза о линейном ограничении, которая проверяется обычной F-статистикой. Выбор числа степеней прогнозов $\hat{y}$ остаётся за исследователем.

Результаты теста Рамсея не зависят от воображения, но куда менее информативны, чем график. При выявленной ошибке они не дают подсказки, какую функциональную форму стоит выбрать и насколько сильно отклонение от первоначально оценённой зависимости. При большом числе наблюдений основная гипотеза может отвергаться и в том случае, когда обнаруженная ошибка спецификации незначительна, так что вы, возможно, предпочли бы ей пренебречь («ловушка большой выборки»). Это касается и другого популярного способа проверки функциональной формы — теста Бокса—Кокса[2] — да и любого статистического критерия.

Иногда тест Рамсея называют тестом на пропущенную переменную (omitted variable test) — кажется, это не очень удачное название. Отвержение основной гипотезы не говорит о необходимости добавлять в модель *содержательно новую* переменную, отражающую неучтённый статистический признак. Речь, скорее, о том, что функциональная форма может исправиться при включении в модель преобразованных значений тех величин, что уже есть в уравнении — например, добавлении квадрата или логарифма какого-либо регрессора.

2.2 Проверка нормальности остатков

По какой-то причине многие статистики верят, что при правильной спецификации остатки имеют распределение, близкое к нормальному (исключение —

[1] В статье Ramsey (1969) можно ещё почитать про три теста с названиями RASET, KOMSET и BAMSET.

[2] Статья-первоисточник [Box, G. E. P., Cox, D. R., 1964] может оказаться нелёгким чтивом, но можно ознакомиться с сутью преобразования Бокса—Кокса и его использованием для определения формы зависимости по работе Sakia (1992).

модели с дискретной объясняемой переменной и модели времени жизни). Имеет смысл построить для остатков гистограмму или график на вероятностной бумаге. В некоторых случаях гистограмма может подсказать правильное преобразование переменных в модели. На рис. 4 изображены гистограмма и график на вероятностной бумаге (график «квантиль — квантиль») для случая, когда линейная зависимость подгонялась вместо логарифмической.

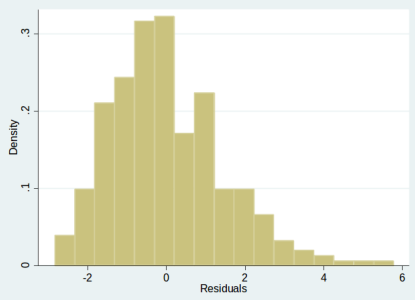


Рис. 4.1. Гистограмма

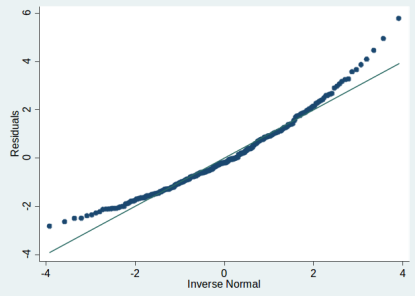


Рис. 4.2. График «квантиль — квантиль»

Рис. 4. Распределение остатков при неправильной функциональной форме

К сожалению, график остатков может искажать не только ошибка спецификации, но и гетероскедастичность; впрочем, она не должна приводить к несимметричной гистограмме. От формальных тестов на нормальность (критериев Харке—Бера, Шапиро—Уилка и т. д.) толку меньше. Они не дают представления о характере отклонений от нормальности и могут отвергать основную гипотезу, когда выявленные отклонения не представляются практически значительными.

3 Функциональная зависимость

А теперь вернёмся к вопросу, с которого начиналась эта статья: как выбрать функциональную форму? Тесты позволяют понять, нет ли ошибки в уже оценённом уравнении, но с какого уравнения разумно начать?

Во-первых, стоит посмотреть, что делали до вас. Для решения многих задач есть уже готовые шаблоны: есть готовые функциональные формы для производственных функций, функций издержек, уравнений заработной платы и т. п. Они могут не подходить идеально для какого-то конкретного случая, но быть разумной точкой отсчёта.

Во-вторых, стоит подумать. Почему только «во-вторых»? Потому что не стоит слишком полагаться на собственный ум: он иногда подводит. Однако ум (особенно при наличии опыта) может подсказать разумную форму зависимости или хотя бы отбраковать неразумные. Подумайте: могут ли потребление C

и доход Y быть связанными уравнением типа $\ln C = \alpha + \beta Y + \varepsilon$? Пожалуй, не стоит полагаться на такой вид зависимости, ведь он предполагает, что по мере роста дохода потребление растёт возрастающими темпами (либо падает при $\beta < 0$, что не менее странно). Уравнения $C = \alpha + \beta Y + \varepsilon$ и $\ln C = \alpha + \beta \ln Y + \varepsilon$ представляются более разумными.

В предыдущем абзаце как раз перечислены основные формы зависимости в эконометрике, но, прежде чем рассмотреть их подробнее, вернёмся к вопросу из начала статьи: почему оцениваются именно линейные зависимости? Можно назвать разные причины, но сейчас рассмотрим такую: параметры линейной модели легко понять. Одна из целей построения статистической модели — сведение большого и трудноосмыслимого объёма данных к нескольким интерпретируемым параметрам. Если из неосмысливаемого набора чисел в нашей выборке мы получили неосмысливаемый набор оценок, это может быть вовсе и не достижением. Именно интерпретируемость делает привлекательными наиболее популярные функциональные формы уравнений регрессии, к которым мы как раз переходим.

Линейная зависимость: $y = \beta_1 + \beta_2 x_2 + \ldots + \beta_k x_k + \varepsilon$. Интерпретация коэффициентов такова: увеличение x_j на единицу соответствует увеличению y на β_j при прочих равных условиях (то есть при неизменных значениях всех остальных регрессоров и случайной составляющей).

Логарифмическая зависимость: $\ln y = \beta_1 + \beta_2 \ln x_2 + \ldots + \beta_k \ln x_k + \varepsilon$. Увеличение x_j на один процент приблизительно соответствует увеличению y на β_j процентов при прочих равных условиях (точнее, в $1{,}01^{\beta_j}$ раз, но приближение очень хорошее). Иначе говоря, коэффициент β_j есть частная эластичность y по x_j.

Полулогарифмическая зависимость: $\ln y = \beta_1 + \beta_2 x_2 + \ldots + \beta_k x_k + \varepsilon$. Увеличение x_j на единицу соответствует при прочих равных условиях увеличению y в e^{β_j} раз, или на $(\mathrm{e}^{\beta_j} - 1) \cdot 100\,\%$. В этой модели интерпретируются потенцированные коэффициенты, но можно пользоваться тем, что $\mathrm{e}^{\beta_j} - 1 \sim \beta_j$ по базе $\beta_j \to 0$. Так что если значение коэффициента невелико, то увеличение x_j на единицу соответствует увеличению y на $\approx \beta_j \cdot 100\,\%$.

4 Заключение

В заключение статьи — пара задач на функциональную форму.[3]

Задание № 1. Сотрудники НИИ размышляют над тем, как оценивать экспоненциальную зависимость y от x. Старший научный сотрудник предлагает свести зависимость к линейной: $\ln y_i = \beta_1 + \beta_2 x_i + \varepsilon_i$ — и оценить её с помощью МНК. Его молодой коллега предлагает применить метод макси-

[3] Задачи взяты из задачника Фурманов, Чернышёва (2014), который можно найти на странице автора: http://www.hse.ru/org/persons/503346.

мального правдоподобия для оценивания нелинейной модели $y_i \sim \mathcal{N}(\mu_i, \sigma^2)$, $\mu_i = \exp(\beta_1 + \beta_2 x_i)$. Посмотрите на две возможные диаграммы рассеяния признаков y и x (рис. 5). Скажите, в каком случае вы бы поддержали старшего, а в каком — младшего научного сотрудника.

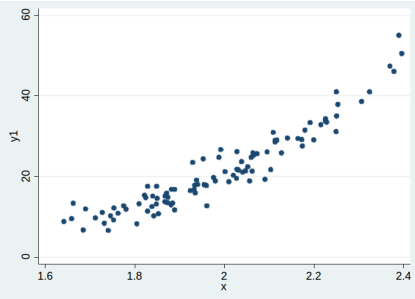


Рис. 5.1. Рассеяние 1

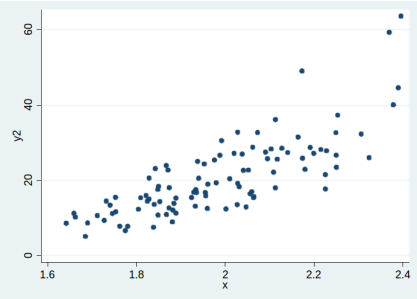


Рис. 5.2. Рассеяние 2

Рис. 5. Диаграммы к заданию 1

Задание № 2. Однажды любознательные барышни Оля и Маша оценили зависимость $y_i = \beta_1 + \beta_2 x_i + \varepsilon_i$. Оля провела тест, выявивший наличие гетероскедастичности. Маша решила, что результаты теста могут быть следствием ошибки спецификации. Оля, чтобы одолеть гетероскедастичность, оценила модель $\frac{y_i}{\sqrt{x_i}} = \alpha_1 \frac{1}{\sqrt{x_i}} + \alpha_2 \sqrt{x_i} + \nu_i$. Маша, чтобы устранить ошибку спецификации, перешла к логарифмам: $\ln y_i = \gamma_1 + \gamma_2 \ln x_i + u_i$. И тот, и другой подход дали хорошие результаты: в новых моделях тесты не выявили ни гетероскедастичности, ни ошибки спецификации. Представьте себя на месте любознательных барышень: какими соображениями вы бы руководствовались при выборе одной из этих двух моделей?

Автор говорит спасибо Дмитрию Арсютину, без чьей работы эта статья была бы беднее.

Список литературы

Box, G. E. P., Cox, D. R. An Analysis of Transformations // Journal of the Royal Statistical Society. Series B (Methodological). — 1964. — Т. 26, № 2. — С. 211—252.

Ramsey J. B. Tests for specification errors in classical linear least-squares regression analysis // Journal of the Royal Statistical Society. Series B (Methodological). — 1969. — Т. 31, № 2. — С. 350—371.

Sakia R. The Box-Cox transformation technique: a review // The statistician. — 1992. — Т. 41, № 2. — С. 169—178.

Фурманов К. К., *Чернышёва И. К.* Сборник задач по эконометрике и моделям времени жизни / НИУ ВШЭ. — 5 авг. 2014. — URL: http://www.hse.ru/data/2014/08/05/1314145909/%C3%90%C2%97%C3%90%C2%B0%C3%90%C2%B4%C3%90%C2%B0%C3%91%C2%87%C3%90%C2%BD%C3%90%C2%B8%C3%90%C2%BA%20%C3%90%C2%BD%C3%90%C2%B0%20%C3%91%C2%81%C3%90%C2%B0%C3%90%C2%B9%C3%91%C2%82.pdf (дата обр. 05.02.2015).

Эконометрика: типичные ошибки студентов и аспирантов

Борис Демешев, Кирилл Фурманов*

21 июня 2015 г.

Аннотация

В этой короткой статье перечислены ошибки, наиболее часто допускаемые студентами и аспирантами при интерпретации эконометрических моделей, написании работ и презентации своих результатов окружающим.

Ключевые слова: эконометрическая культура, статистическая мудрость, довольная комиссия, распространённые ошибки.

1 Советы Кирилла Фурманова

1. Коэффициенты в регрессии показывают наличие лишь статистической взаимосвязи. Причинно-следственная интерпретация часто ошибочна. Она возможна в некоторых случаях — например, когда данные получены в результате эксперимента.
2. Если нулевая гипотеза не отвергается, это не означает, что она верна. Корректно говорить, что недостаточно данных, чтобы отвергнуть $\mathcal{H}_0$, или что данные не противоречат $\mathcal{H}_0$. При этом они ещё много чему могут не противоречить.
3. Нужно понимать соответствие между содержательной гипотезой, формулируемой без статистических терминов, и формулировкой в терминах нулевой-альтернативной гипотезы.
4. «Мы говорим — Ленин, подразумеваем — партия, мы говорим — партия, подразумеваем — Ленин». (Маяковский, 1957.) Мы говорим, что проверяем гипотезу о значимости регрессии, хотя на самом деле проверяем гипотезу о незначимости регрессии, то есть $\mathcal{H}_0\colon \beta_2 = \beta_3 = \ldots = \beta_k = 0$.

*НИУ ВШЭ, Москва.

5. Если правильно интерпретировать коэффициенты, то разумными и верными могут оказаться одновременно несколько разных моделей. При этом все оценки коэффициентов будут несмещёнными! Например, и регрессия игрека на икс, и регрессия игрека на икс и зет могут быть осмысленны и интересны. Коэффициент при иксе в первой модели показывает, на сколько в среднем изменяется игрек, когда икс меняется на единицу, а во второй — насколько в среднем изменяется игрек, когда икс меняется на единицу, а зет не изменяется.
6. Значимость — это не то же самое, что существенность. Коэффициент может быть значимым, но совершенно бесполезным. Если месячная зарплата мужчин и женщин значимо отличается, но это отличие составляет два рубля, то можно считать, что его нет. Возможно домножать коэффициент на стандартную ошибку регрессора или на квантили.

2 Советы Бориса Демешева

1. Больше графиков! Работа по эконометрике без картинок скучна и бессмысленна. Если сомневаешься, нужно ли построить ещё один график, значит, нужно. Графиков много разных, не бойся экспериментировать! Например, в R пакет **mvtsplot** позволяет на одном графике осмысленно изобразить 60 временных рядов (см. Peng, 2012).
2. Рассказывай презентацию для идиотов. Методов и моделей слишком много. Во время презентации исходи из предположения, что комиссия не знает эту тему. Выбирай простые примеры, а не рассказывай про общий случай. Меньше буковок на слайдах!
3. «Ларису Ивановну хочу!» — «Хочу построить модель, которая описывает. . . » — это бяка. Так бессмысленно ставить вопрос! Хочу спасти город Энск от бедности! Это — хорошо! Хочу заработать миллион баксов — чуть менее прикольно, но тоже неплохо. Модель не строят просто так, модель строят для того, чтобы ответить на содержательный вопрос или чтобы использовать для некоторого действия.
4. Зачастую не нужны сложные модели. Всегда проверяй сложную модель против самой простейшей, которая приходит в голову. Например, качество прогнозов во временных рядах стоит проверить против модели «завтра будет так же, как сегодня».
5. После написания курсовой, ВКР, диссертации напиши для себя мораль. Там, конечно, аршинными буквами будет «в следующем году я начну писать до Нового года». Чему тебя научила ВКР? Ещё можно написать протокол воспроизведения всех регрессий и обработки данных. Полезно выложить в публичный доступ.
6. Осваивай открытый софт: R, gretl, LaTeX, Markdown, Python, SQL и ещё куча страшных слов! Это бесплатно, и сообщество просто огромное.

Модно, стильно, молодёжно!

7. А ты знаешь, что такое p-value? Учи матстат!
8. Мелочи по представлению результатов, способные вызвать праведный гнев членов комиссии:
   а) *Техническая подготовка.* Проверь флешку, выложи файл в интернет заранее, чтобы, если флешка не сработает, его можно было быстро скачать. Будь готов к мелочам вроде проектора, не отличающего красного от розового, яркого солнца, и полям слайда, вылезающим за доску.
   б) *Отсутствие номера на слайде.* А покажи-ка мне слайд, ну, этот… на котором… Делай номер внизу слайда.
   в) *Неподписанные оси на графике.* Принцип идеального графика: идеальный график можно понять, не читая оставшуюся часть работы. Подписывай оси, приводи единицы измерения, расшифровывай названия переменных. Чем больше нужно устных комментариев к графику, чтобы понять его, тем хуже график.
   г) *Семь знаков после запятой.* По некоторым данным (Frank [и др.], 2008), люди народности пираха (Бразилия) считают так: один, два и много. Пираха знают толк в знаках после запятой!

Список литературы

Frank M. C. [и др.] Number as a cognitive technology: Evidence from Pirahã language and cognition // Cognition. — 2008. — Т. 108, № 3. — С. 819—824.

Peng R. D. mvtsplot: Multivariate Time Series Plot. — 2012. — R package version 1.0-1.

Маяковский В. В. Полное собрание сочинений в тринадцати томах : в 13 т. Т. 6. — М. : Государственное издательство художественной литературы, 1957.

Метод k-средних и распознавание рукописных цифр

Саша Кузнецова*

21 июня 2015 г.

Аннотация

Задача распознавания рукописного текста — одна из классических задач машинного обучения, к решению которой применялось такое количество алгоритмов, что она успешно может быть использована в качестве учебной.

В этой заметке мы будем учиться распознавать написанные от руки цифры.

Ключевые слова: k ближайших соседей, R, машинное обучение, распознавание образов.

Мы обратимся к одному из наиболее известных хранилищ, в котором собраны обработанные изображения цифр, разберём, как эти данные оттуда извлекать, а затем применим к ним алгоритм k ближайших соседей, используя R.

1 База данных MNIST

Источником данных нам послужит база «MNIST» (LeCun, Cortes, Burges, 2011), http://yann.lecun.com/exdb/mnist/, в которой хранятся 70 000 изображений цифр, написанных несколькими сотнями разных людей. Каждая картинка в этой базе обработана так, чтобы поместиться в квадратик размером 28×28 пикселей. Каждый пиксель представлен числом от 0 до 255, где 0 соответствует белому цвету, а 255 — чёрному.

Все изображения поделены на две части: 60 000 относятся к учебной выборке, а 10 000 — к тестовой. По учебной выборке наш алгоритм будет настраивать свои параметры, а по тестовой мы будем оценивать качество классификации.

*НИУ ВШЭ, Москва.

Такое разбиение нужно для того, чтобы убедиться, что алгоритм не переобучился, то есть не настроен исключительно на учебные примеры и способен правильно классифицировать новые для него изображения.

Каждое из изображений в нашей задаче должно быть отнесено к одному из десяти классов — это цифры от 0 до 9. Для элементов обучающей выборки известно, к какому классу они принадлежат, поэтому настройка параметров классифицирующего алгоритма относится к области *обучения с учителем*. Это значит, что процесс обучения использует известные ответы и стремится за счёт выбора параметров сделать предсказания алгоритма максимально близкими к правильным (кстати, таким образом мы обучали все наши алгоритмы в курсе эконометрики).

Как говорится, «наивные студенты думали, что .csv-файлы на деревьях, как булки, растут», но всё оказалось совсем не так. Для того чтобы прочитать IDX-файл, в котором хранятся данные с изображениями цифр в базе «MNIST», мне потребовались Google и кандидат физико-математических наук. Проблема заключается в том, что данные хранятся в этой базе в бинарном виде, и их не открыть в привычных нам приложениях. Вот как это можно сделать в R.

(1) Скачав данные с сайта, открываем файл с учебной выборкой на чтение командой `file`:

```
to.read <- file("train-images.idx3-ubyte", "rb")
```

(2) Данные устроены таким образом, что в самом начале мы читаем заголовок из четырёх чисел. Эти первые четыре числа содержат информацию о размерах выборки, упомянутых выше. Считываем их из полученного объекта `to.read` с помощью функции для чтения бинарных данных `readBin`.

```
readBin(to.read, integer(), n = 4, endian="big")

## [1]  2051 60000    28    28
```

(3) Далее для каждой картинки следуют 28×28 байт, содержащие информацию о цвете каждого пикселя (числа от 0 до 255) и записанные из изображения построчно. Получаем большой массив `TRAIN` из 60 000 таблиц размером 28×28: берём первые 28 чисел и кладём их в первый столбец первой таблицы, продолжаем, пока не дойдём до 28-го столбца, затем приступаем к следующей таблице — и так до конца.

Нужно обратить внимание, что изначально данные были разложены в строку, но мы только что разложили их по столбцам, для того чтобы позже было удобнее выводить рисунок.

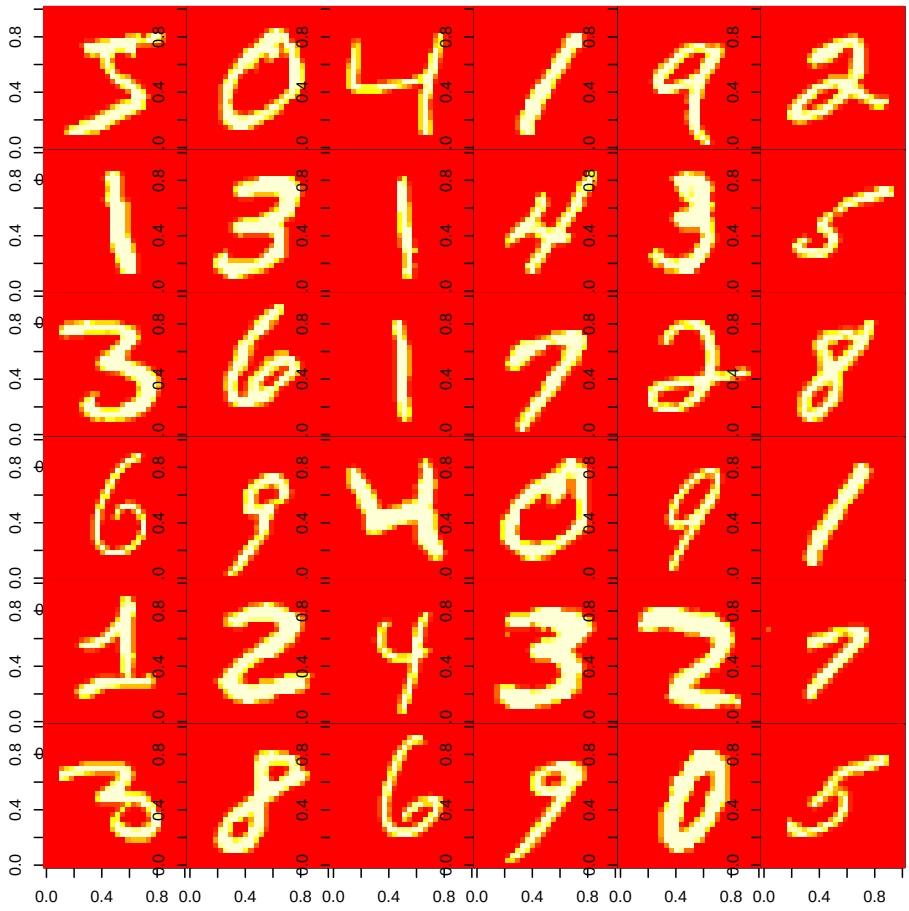
```
TRAIN <- array(data = NA, dim = c(28,28,60000))

for(i in 1:60000)
  {
TRAIN[,,i] <- matrix(readBin(to.read,integer(), size = 1,
    n = 28*28, signed = FALSE, endian = "big"), 28, 28);
  }
close(to.read)
```

(4) Перед тем как начинать работать с данными и оценивать по ним какие-либо модели, обычно бывает полезно взглянуть на них и построить описательные статистики. В данном случае для этой цели может послужить сама картинка.

```
layout(matrix(c(1:36), 6, 6, byrow = TRUE),
   widths = lcm(rep(2.5,36)), heights = lcm(rep(2.5,36)))
par(mar=c(0,0,0,0))

for(i in 1:36)
  {
image(TRAIN[,28:1,i])
  }
```



Нарисуем цифры по первым 36 таблицам из массива `TRAIN`. Нужно обратить внимание, что, нарисуй мы сейчас всё как есть, мы бы получили перевёрнутые цифры (это можно проверить), потому что функция `image` будет соотносить первый столбец с нулевой ординатой на картинке, но мы помним, что первый столбец соответствовал верхней строке рисунка. Для того чтобы избежать этой проблемы, будем рисовать столбцы в обратном порядке, от 28-го к 1-му: `TRAIN[,28:1,]`.

(5) Описанную выше процедуру повторяем, чтобы считать метки классов для учебной выборки. Заголовок в данном случае состоит из двух чисел, а данные должны восприниматься R'ом не как числа, а как категории: этого можно добиться с помощью команды `as.factor`.

```
to.read <- file("train-labels.idx1-ubyte", "rb")
header <- readBin(to.read, integer(), n=2, endian="big")
Train_labels <- readBin(to.read, integer(), size = 1,
            n = 60000, signed = FALSE, endian="big")
Train_labels <- as.factor(Train_labels)
close(to.read)
```

Итак, на данном этапе мы являемся обладателями учебной выборки (но умеем читать и многое другое) и великолепной картинки, что означает, что пора бы с ними что-нибудь да сделать.

2 Алгоритм k ближайших соседей

Одним из простейших алгоритмов классификации является метод k ближайших соседей (k Nearest Neighbours, kNN). Основной идеей данного алгоритма является то, что объект, как правило, находится в окружении объектов своего же класса. Таким образом, чтобы классифицировать новый объект, мы должны посмотреть на k ближайших к нему объектов из учебной выборки и отнести его к тому классу, который среди них чаще встретился.

Первый вопрос, который перед нами встаёт, — это выбор функции расстояния между объектами выборки. В простом случае, когда объекты представлены в виде числовых векторов, пользуются простой евклидовой метрикой, то есть $\rho(a,b) = \sqrt{\sum_{k=1}^{m}(a_k - b_k)^2}$.

Второй проблемой является выбор значения k: сколько объектов нам нужно посмотреть, чтобы наиболее точно классифицировать наш? Единого правила для того, чтобы выбрать k, не существует, однако нужно учитывать, что при слишком маленьких k алгоритм будет неустойчивым, а при слишком больших начнёт подстраиваться к шуму и терять обобщающую способность. Конкретное значение k можно определить опытным путём: попробовать диапазон значений и посмотреть, какое подходит лучше.

2.1 Существующие реализации

Для реализации алгоритма k ближайших соседей в R есть как минимум три пакета: **kknn**, **FNN** и **RWeka**, причём в двух последних есть ещё и готовые базы данных.

Пакет **kknn** реализует алгоритм kNN с весами, где голоса «соседей» не равнозначны, а входят в общую сумму с определёнными весами. Мы его реализовывать не будем.

Воспользуемся функцией `install.packages(c("FNN", "RWeka"))`, чтобы загрузить пакеты.

2.2 Пакет FNN

Воспользуемся теперь пакетом **FNN** (Beygelzimer [и др.], 2013), предварительно проделав с данными для тестовой выборки то же, что и с учебными, в итоге получив два больших массива, `TRAIN` и `TEST`, а также два вектора ответов, `Trainlabels` и `Testlabels`.

Функция `knn` пакета **FNN** в качестве аргументов принимает матрицы учебных и тестовых данных, а также вектор ответов для тренировочной выборки, а на выходе отдаёт предсказанную классификацию для тестовой выборки. Эта функция сразу и обучается, и предсказывает — очень удобно.

Сейчас наши данные выглядят не так, как нужно этой функции, поэтому преобразуем их в матрицу, содержащую значения цвета всех пикселей, размещённых в одну строку для каждой картинки. Воспользуемся функцией `as.vector`, которая будет по очереди брать столбцы из матрицы и выкладывать их в одну строку. Кроме того, для примера мы будем использовать только первую тысячу наблюдений из учебной выборки и пятьсот из тестовой. Если читатель желает получить более точный классификатор, то в этом месте следует использовать все 60 000 и 10 000 соответственно.

```
n <- 1000
train = matrix(data = NA, nrow = n, ncol = 28*28 )

for (i in 1:n)
  {
train[i,] <- as.vector(TRAIN[,,i])
  }
train_labels <- Train_labels[1:n]
```

Теперь воспользуемся функцией `knn`, чтобы классифицировать объекты тестовой выборки, и запишем результаты в вектор `results`. Возьмём, например, $k = 10$. Преимуществом функции является высокая скорость подсчёта предсказаний.

```
library(FNN)
results <- (0:9)[knn(train, test, train_labels, k = 10,
                     algorithm = "cover_tree")]
```

Посмотрим на долю ошибок, которые мы допустили при классификации. Она достаточно высока, но это можно объяснить тем, что мы использовали слишком маленькое для такого алгоритма число наблюдений.

```
errors <- (sum(results != test_labels))/m
errors

## [1] 0.2
```

2.3 Пакет RWeka

Пакет **RWeka** (Hornik, Buchta, Zeileis, 2009) позволяет использовать из R все возможности среды для машинного обучения Weka. Среда Weka, http://www.cs.waikato.ac.nz/ml/weka, содержит целый набор алгоритмов машинного обучения, среди которых есть и нужный нам. В Weka замечательно то, что алгоритм сам подбирает оптимальное в нашем случае значение k из заданного диапазона, а потом оценивает полученный классификатор с помощью пятикратной кросс-валидации. Это значит, что он поделит выборку на пять кусочков и по очереди будет использовать их в качестве тестовых, чтобы надёжнее оценить классификацию, исключив возможность того, что хорошие или плохие предсказания получились под влиянием какого-либо конкретного набора данных.

Устанавливать отдельно Weka не нужно, необходимые файлы пакет **RWeka** установит сам. Кстати, Weka разработана в Новой Зеландии, на родине R, и вообще, Weka — это птичка:

Рис. 1. Новозеландская курица Weka

Можно послушать, как она поёт, http://www.cs.waikato.ac.nz/ml/weka/sounds/weka-long.au.

Однако при загрузке пакета нужно убедиться, что на компьютере установлена среда Java, и отдавать себе отчёт, что поиск подходящих параметров

и пятикратная оценка классификатора будет требовать значительного времени.

Используем функцию `IBk`. Для этого добавим слева к нашей матрице `train` столбец с ответами и преобразуем матрицу в объект `data.frame`. Выражение `K = 10` указывает на то, что алгоритм будет перебирать все значения от 1 до 10. Функция `evaluate_weka_classifier` оценивает качество результатов и показывает, какие значения были классифицированы правильно в результате пятикратного разбиения выборки на кусочки и последующего усреднения, а какие — нет.

Обратите внимание на Confusion matrix, которая показывает, с какими именно классами возникали ошибки: можно заметить, что пятёрки и восьмёрки классифицировались как всё что угодно.

```
library(RWeka)

train_weka <- data.frame(train_labels, train)
train_weka[,1] <- as.factor(train_weka[,1])
classifier <- IBk(train_labels~., data = train_weka,
                  control = Weka_control(K = 10, X=TRUE))
classifier

## IB1 instance-based classifier
## using 1 nearest neighbour(s) for classification

evaluate_Weka_classifier(classifier, numFolds = 5)

## === 5 Fold Cross Validation ===
##
## === Summary ===
##
## Correctly Classified Instances         869               86.9    %
## Incorrectly Classified Instances       131               13.1    %
## Kappa statistic                          0.8542
## Mean absolute error                      0.0281
## Root mean squared error                  0.1609
## Relative absolute error                 15.6256 %
## Root relative squared error             53.6609 %
## Coverage of cases (0.95 level)          86.9    %
## Mean rel. region size (0.95 level)      10      %
## Total Number of Instances             1000
##
## === Confusion Matrix ===
##
```

```
##    a   b   c   d   e   f   g   h   i   j   <-- classified as
##   93   0   0   0   0   0   3   0   0   1 |   a = 0
##    0 113   1   1   0   0   0   1   0   0 |   b = 1
##    2   7  79   1   0   1   2   4   1   2 |   c = 2
##    1   2   2  82   0   2   1   1   1   1 |   d = 3
##    0   3   0   0  86   0   2   1   0  13 |   e = 4
##    0   1   1   2   0  80   4   1   1   2 |   f = 5
##    1   3   0   0   1   0  87   0   1   1 |   g = 6
##    0   4   1   0   2   2   0 101   1   6 |   h = 7
##    0   3   4   2   1   6   2   1  66   2 |   i = 8
##    1   1   0   0  11   2   1   2   0  82 |   j = 9
```

Теперь предскажем значения для тестовой выборки с помощью функции `predict` и посмотрим на долю ошибок.

```
test_weka = data.frame(test)
names(test_weka) = names(train_weka)[-1]

predictions <- predict(classifier, newdata = test_weka)

errors <- (sum(predictions != test_labels))/m
errors

## [1] 0.174
```

3 Заключение

Таким образом, мы рассмотрели интересную задачу и отличную, заботливо собранную базу данных, применили к ним один из самых первых и простых алгоритмов машинного обучения, который, однако, часто показывает очень хорошие результаты. Но это был один только пример, а в качестве инструментов в данном случае могут быть использованы разнообразнейшие алгоритмы. Кроме того, можно подумать над тем, как преобразовать сами изображения для улучшения качества классификации. Удачи!

Список литературы

Beygelzimer A. [и др.] FNN: Fast Nearest Neighbor Search Algorithms and Applications. — 2013. — R package version 1.1.

Hornik K., *Buchta C.*, *Zeileis A.* Open-Source Machine Learning: R Meets Weka // Computational Statistics. — 2009. — Т. 24, № 2. — С. 225—232.

LeCun Y., *Cortes C.*, *Burges C. J. C.* The MNIST database of handwritten digits. — 25 дек. 2011. — URL: http://yann.lecun.com/exdb/mnist/index.html.

Кто выигрывает рэп-баттлы?

Андрей Зубанов*

21 июня 2015 г.

Аннотация

Статистика является мощным инструментом при изучении не только экономических и социальных, но и культурных явлений. Даже простые статистические методы могут оказаться полезными при изучении таких событий, как различные соревнования, телешоу, прослушивания.

Ключевые слова: рэп, баттл, жеребьёвка.

1 Влияние позиции участника на результат

Данная статья анализирует рэп-баттлы на примере популярного российского «Versus Battle». В исследование вошли данные, собранные по выпускам «Первый сезон», «Второй сезон» и «Межсезонье». В баттле участвуют два человека (реже — две команды). Сущность рэп-баттла «Versus» состоит в том, чтобы посредством заранее подготовленного речитатива высказаться о себе и своём сопернике. Баттл проходит без музыкального сопровождения. Победитель определяется тремя судьями простым большинством голосов. На протяжении рассматриваемых баттлов участники стоят лицом друг к другу, сбоку от них стоят судьи, вокруг располагаются немногочисленные болельщики, приглашённые на мероприятие. Баттл состоит из трёх раундов, каждый из которых начинает участник, выбранный жеребьёвкой первым.

Согласно данным, всего победил первый по очереди игрок 6 раз, второй игрок — 22 раза. Всего победил игрок слева 18 раз, справа — 10 раз.

Основной гипотезой исследования является гипотеза о том, что второй участник вследствие очерёдности хода имеет преимущество, предположительно из-за возможности ответить в своём выступлении на выступление соперника. Также это возможно и в силу того, что выступающий последним лучше запоминается судьям и поэтому выше оценивается. Вторая гипотеза — предположение о том, что из-за своего положения относительно судей участник слева имеет преимущество.

Если результат баттла не зависит от очерёдности, то при любом назначенном номере участник должен выигрывать с вероятностью, близкой к 1/2. Проверим эту гипотезу, учитывая, что номер выигравшего участника — случайная величина, распределённая биномиально.

$$\frac{|\hat{p}_n - p_0|\sqrt{n}}{\sqrt{\hat{p}_n(1-\hat{p}_n)}} \sim \mathsf{t}_{28} \tag{1.1}$$

*НИУ ВШЭ, Москва.

Рис. 1. Рэп-баттл

Доля выигравших вторых игроков составляет 22/28, всего наблюдений в выборке 28. Значит,

$$z_{\text{расч}} = \frac{|22/28 - 1/2|\sqrt{28}}{\sqrt{22/28 \cdot (1 - 22/28)}} = 8{,}98 > 2{,}05 = \mathsf{t}_{0{,}975;28} \tag{1.2}$$

Таким образом, выступающие вторыми участники значимо чаще выигрывают рэп-баттлы.

Также и участники слева чаще выигрывают баттлы (вероятность 18/28).

$$z_{\text{расч}} = \frac{|18/28 - 1/2|\sqrt{28}}{\sqrt{18/28 \cdot (1 - 18/28)}} = 3{,}29 > 2{,}05 = \mathsf{t}_{0{,}975;28} \tag{1.3}$$

Однако при уровне значимости в 0,001 критическая статистика составляет $\mathsf{t}_{0{,}0005;28} = 3{,}67$, и значимого различия не выявляется.

2 Выводы

Вопрос о зависимости между позицией участника и очерёдностью остаётся открытым, однако первое не должно влиять на второе, так как расстановка происходит после жеребьёвки.

Несмотря на то что жеребьёвка не показывается зрителю, а организаторы не раскрывают процедуры, допустимо считать её случайной, так как это соответствует традициям рэп-баттлов.

В итоге нельзя с уверенностью сказать о причинно-следственных связях между очерёдностью участников в баттлах и их результатами, ведь может существовать третий фактор, влияющий и на то, и на другое. Однако можно утверждать, что вторые участники действительно значимо чаще выигрывают рэп-баттлы.

Победитель по номеру	Положение первого относительно судей	Выиграл ли участник справа
1	справа	1
1	слева	0
2	слева	1
2	слева	1
1	слева	0
1	слева	0
2	слева	1
2	слева	1
2	слева	1
2	слева	1
2	справа	0
2	справа	0
2	слева	1
2	слева	1
2	слева	1
2	слева	1
2	слева	1
2	справа	0
2	справа	0
1	справа	1
2	слева	1
2	слева	1
1	слева	0
2	слева	1
2	справа	0
2	слева	1
2	справа	0
2	слева	1

Таблица 1. Данные о победах первого/второго по очереди участника и участника, стоящего слева/справа от судей

Весёлый уголок

1 Дорожный знак

Уважаемые читатели, перед вами изображён дорожный знак, который не оставит равнодушным любого человека, изучавшего математическую статистику. Этот, например, расположен на улице Jean-Jacques Rousseau в Иври (пригород Парижа, Франция).

Объявляется конкурс на лучшее (самое смешное, остроумное, оригинальное) название для этого дорожного знака! Пожалуйста, присылайте свои варианты ответа в редакцию! Лучшие варианты будут напечатаны в следующем номере вместе с именами авторов.

2 Как отцы науки коэффициенты подбирали

«

Мои данныя состояли изъ 930 наблюденій роста совершеннолѣтнихъ дѣтей и ихъ прямыхъ восходящихъ родственниковъ, числомъ 205 составлявшихъ. Всякій разъ я превращалъ высоту женскаго стана въ эквивалентъ мужескаго и использовалъ ихъ въ преобразованномъ видѣ, дабы не вызвать упрековъ, происходящихъ отъ наличествованія разницы въ ростахъ половой природы, буде я говорилъ-бы о среднихъ. Множитель, использованный мною, составлялъ 1,08, что равносильно прибавленію чуть менѣе чѣмъ одной двѣнадцатой части къ росту каждой женщины. Сей множитель ненамного отличается отъ оныхъ, использованныхъ иными антропологами, кои къ тому-же сами малости въ различіяхъ множителей имѣютъ; какъ-бы то ни было, онъ подходитъ къ моимъ даннымъ лучше, нежели 1,07 или 1,09. Итоговый результатъ никоимъ образомъ не относится къ тѣмъ, что зависятъ отъ этихъ минутныхъ деталей, ибо такъ сталось, что изъ-за ошибочнаго указанія расчетчикъ, которому я попервости ввѣрилъ цифры, использовалъ немного другой множитель, однако результатъ вышелъ практически одинъ въ одинъ.

»

Фрэнсисъ Гальтонъ,
«Регрессированіе къ посредственному при наслѣдованіи ростовъ», 1886.

Опубликовано в «Эпсилоне»

Новости CrossValidated: выпуск 1

Андрей Костырка

Аннотация. В этом разделе представлены компиляции ответов на наиболее интересные вопросы, задававшиеся на сайте StackExchange в разделе «Статистика» (stats). К каждому вопросу приводятся один или несколько ответов, получивших наибольшее количество пользовательских голосов. Мнение авторов ответов может не совпадать со мнением редакции.

Ключевые слова: статистика, вопросы, ответы, интернет.

1 Что такое over-fitting?

Вопрос. Где в реальной жизни чаще всего возникает проблема *переподгонки* (переобучения — не в смысле «обучения заново», а в смысле «чрезмерного обучения»)? Чем плоха чрезмерно точная подгонка модели под данные?

Исходный вопрос: http://stats.stackexchange.com/q/128616.

1.1 Подгонка под особенности шума

Зачастую набор данных является слишком простым, а модель — слишком «продвинутой», из-за чего оценивание даёт ложные либо нестабильные результаты оценивания. Дополнительные параметры сложных моделей иногда оцениваются по особенностям случайного шума, который совершенно не связан с самой структурой данных, но может образовывать статистические артефакты в единичной реализации.

1.2 Модель плохо работает за пределами выборки

Несмотря на свою примитивность, линейные модели довольно неплохо дают общее представление об устройстве данных. Если же точки располагаются вдоль воображаемого нелинейного облака точек, то тогда специфические математические функции (экспонента, логарифм, полином k-й степени, синус и проч.) принимают значения, более близкие к значениям набора данных, однако за границами определённого диапазона их поведение теряет адекватность интерпретации.

Рассмотрим динамику населения США в XX веке (рис. 1). Линейная модель довольно хорошо описывает данные. Полином шестой степени более точно проходит через имеющиеся точки, однако даёт прогноз, согласно которому к 2050 году всё население США загадочным образом исчезнет. Подобное экстраполирование абсурдно, поэтому этот пример — классический случай переподгонки.

НИУ ВШЭ, Москва.

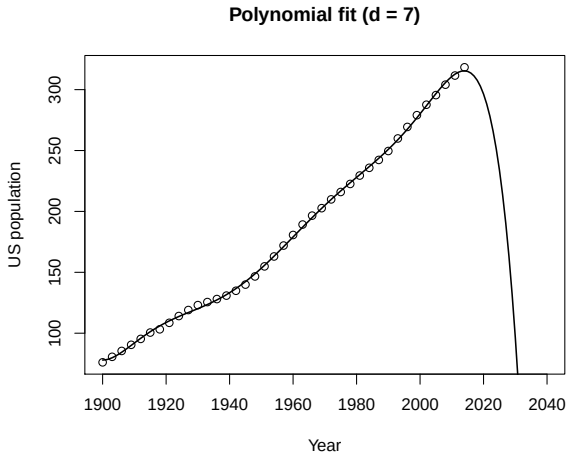


Рис. 1. Over-fitting динамики населения США по данным multpl.com

1.3 Система Птолемея

Птолемей полагал, что Земля находится в центре Вселенной, и вывел громоздкую систему вложенных сферических орбит, хорошо объяснявшую движение небесных тел. Однако реальные измерения систематически отличались от прогнозов, реализуемых в рамках системы Птолемея, поэтому астрономам-геоцентристам приходилось придумывать всё больше дополнительных сфер, пока, наконец, модель не стала настолько запутанной, что начала вызывать подозрения в истинности предпосылок, на которых основывалась.

1.4 Не бывает переподгонки процесса, порождающего данные

Сам по себе DGP (data-generating process) переподогнать невозможно. Мы не знаем истинного DGP, но лишь пытаемся его моделировать. Так как DGP один, то и истинная модель может быть только одна, а все остальные содержат ошибки спецификации того или иного уровня. В социальных науках большинство моделей принципиально неполные или неправильно специфицированные, поэтому усложнённые модели захватывают особенности доступного набора данных, а не процесса, стоящего за ними.

1.5 Боязнь пропущенных переменных

Многие эконометристы полагают, что пропущенные переменные — это опаснейшая проблема, куда более острая, чем избыточные переменные. Чтобы выбрать из двух зол наименьшее, некоторые из этих эконометристов

добавляют в модель степени регрессоров, а также всевозможные пересечения (кросс-произведения) и иррелевантные переменные. В самом общем случае добавление в уравнение множественной регрессии всех доступных в наборе переменных, которые могут потенциально обладать объясняющей силой, является переподгонкой: исследователь наблюдает не генеральную совокупность, а только выборку, поэтому он не может знать, какая из всех возможных спецификаций является верной.

Как водится, есть две новости: хорошая и плохая. Хорошая новость: включение лишних переменных не приводит к смещению оценок коэффициентов при значимых. Плохая новость: точность оценивания релевантных коэффициентов падает, ошибка регрессии растёт, доверительный интервал прогноза расширяется.

1.6 Мнение редактора

Два критерия качества модели — goodness of fit (подгонка) и goodness of forecast (прогноз) — зачастую бывают недостижимы одновременно. Представьте себе отличную базу панельных данных с одним миллионом индивидов, каждый из которых наблюдается в течение пяти-десяти лет (такие базы, например, имеются в распоряжении у французских статистических органов, а также у сотрудников Национальной школы статистики). Представим, что мы строим зарплатное уравнение в зависимости от стажа, пола, образования, возраста с квадратом и других стандартных факторов. Каждый индивид обладает своим индивидуальным эффектом, который легче всего измерить при помощи дамми-переменной для этого самого индивида. Предположим, у нас есть 8 000 000 наблюдений и 1 000 005 оцениваемых при помощи МНК коэффициентов (стаж, пол, возраст с квадратом, образование и миллион дамми). Оценённая модель будет обладать фантастически высоким показателем R^2 и отлично объяснять устройство данных; почти все коэффициенты при этом будут значимы! Однако представим, что в выборку попадает новый индивид — одна новая точка. Оценки индивидуального эффекта для него у нас попросту нет. Каким будет прогноз его заработной платы? Как в старом анекдоте: «Насчёт СССР мы не знаем, но на китайско-финской границе всё будет спокойно!»

Ещё более гипертрофированный пример: есть срез из 1 000 000 индивидов в один момент времени, и для каждого индивида в уравнение регрессии добавляется его дамми. Такая модель объяснит 100 % вариабельности переменной дохода, однако будет неспособна предсказать доход новых респондентов.

1.7 Вопрос читателям

Придумайте для какого-либо набора данных модель, которая обладает очень хорошей прогнозной силой (довольно точно предсказывает значения для точек как внутри диапазона значений — где модель оценивалась, — так и

за его пределами — где необходимо угадать свойства объекта, не участвовавшего в обучении), однако скверной объясняющей способностью.

2 Следует ли из причинно-следственной связи корреляция?

Вопрос. Многие студенты второго курса бывали биты за то, что утверждали, будто бы из корреляции следует каузальность, т. е. «они коррелируют, следовательно, одно является причиной другого». Любой зарубежный студент-отличник на устном экзамене повторяет мантру «correlation does not imply causation». Предположение о независимости случайных величин всегда сильнее предположения об их некоррелированности. Однако верно ли обратное? Обязательно ли из причинно-следственной связи следует корреляция?

Исходный вопрос: http://stats.stackexchange.com/q/26300.

2.1 Линейная корреляция

Если понимать под корреляцией общую корреляцию Пирсона, рассчитываемую по формуле

$$\operatorname{Corr}(X,Y) = \frac{\operatorname{Cov}(X,Y)}{\sqrt{\operatorname{Var}(X)\operatorname{Var}(Y)}},$$

то причинно-следственная связь двух переменных не всегда приводит к возникновению линейной корреляции. Достаточно рассмотреть такой набор данных, как точки, лежащие на прямой $y = x^2$ в любом симметричном относительно начала координат диапазоне. В данном случае зависимость будет функциональной, однако коэффициент корреляции Пирсона будет равен нулю. Во многих справочниках присутствует изображение (рис. 2[1]), на котором показано, что у многих наборов данных две переменные явно связаны некоторой зависимостью, однако линейная корреляция равна нулю.

Более подходящий термин для данного вопроса — это «взаимная информация». Справедливо утверждение о том, что причинно-следственная связь влечёт высокую *взаимную информацию*. Последняя имеет более сложное определение, чем корреляция, и измеряет уменьшение неопределённости относительно одной случайной величины при поступлении информации о другой случайной величине.

Следует помнить, что если некоторая причина влечёт некоторое следствие и при этом эта причина идеально коррелирует с другой причиной, вызывающей противоположный результат, то корреляция этих двух причин и следствия равна нулю. Однако в действительности такой идеальной линейной зависивости между двумя причинами со строго противоположными эффектами не существует, поэтому не следует полагать, что такие причины

[1] Исходный примет взят с en.wikipedia.org/wiki/File:Correlation_examples2.svg.

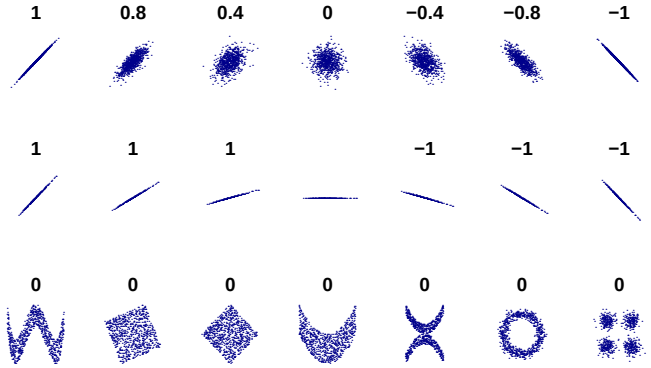


Рис. 2. Коэффициенты корреляции в различных наборах данных

существуют вообще. Пример: единороги вызывают некоторое явление, гномы оказывают точно такое же противодействие, а посему итоговый эффект равен нулю.

2.2 Теоретический контрпример

Рассмотрим две случайные величины: $X \sim \mathcal{N}(0;1)$, $Y = X^2$. По определению $Y \sim \chi^2_1$. Трудно придумать более сильную причинно-следственную связь: X полностью определяет Y. Мы видим, что $\mathbb{E}(X) = 0$, $\mathbb{E}(Y) = 1$. Вычислим значение коэффициента ковариации.

$$\begin{aligned}\operatorname{Cov}(X,Y) &= \mathbb{E}\big((X - \mathbb{E}(X))(Y - \mathbb{E}(Y))\big) = \\ &= \mathbb{E}\big((X-0)(Y-1)\big) = \mathbb{E}(XY) - \mathbb{E}(X) = \mathbb{E}(X^3) - 0 = 0\end{aligned}$$

Мы воспользовались тем свойством, что нечётные центральные моменты стандартной нормальной случайной величины равны нулю. Следовательно, корреляция между ними также равна нулю.

Данное доказательство работает не только для стандартного нормального, но и вообще для любого симметричного относительно нуля распределения (равномерного от $-a$ до a, Лапласа, Стьюдента и проч.), у которого существуют хотя бы три центральных момента. Если каждое положительное значение величины X так же вероятно, как и противоположное ему, то при возведении в квадрат мы не можем сказать, связаны ли бóльшие значения квадрата случайной величины с положительными или отрицательными X.

2.3 Эмпирический контрпример

Если одно случайное событие является причиной другого случайного события, между ними обязана существовать некоторая взаимосвязь (одно-

сторонняя или двухсторонняя), которая может выражаться в нелинейной корреляции.

При проведении выборочных исследований домохозяйств иногда в качестве средства опроса используется телефон. При этом вероятность ответа на звонок максимальна у людей, относящихся к среднему классу в терминах дохода, и значительно ниже у очень состоятельных или, наоборот, социально незащищённых граждан. На рис. 2 видно, что параболическая зависимость переменных (нижний ряд, центральное изображение, ветви параболы могут быть направлены вверх или вниз) влечёт нулевую корреляцию. Причинно-следственная связь вида «доход влияет на вероятность ответа по телефону» заключается в том, что на верхнем конце распределения люди предпочитают не выдавать информации о себе (в том числе и потому, что к ним зачастую обращаются по телефону с просьбой «поделиться доходами»), а на нижнем конце распределения индивиды чаще являются должниками, имеющими непогашенный долг перед банком или знакомыми, и это является причиной их более осторожного поведения.

3 Почему вариабельность измеряют именно квадратами?

Вопрос. Почему в статистике для определения меры разброса случайной величины берутся *квадраты* отклонений от среднего, и почему стандартное отклонение считается как корень из мат. ожидания квадрата? Разве мат. ожидание модуля отклонений не покажет вариабельность данных?

Почему вместо	не используют
$\sigma = \sqrt{\mathbb{E}\big((X-\mu_X)^2\big)}$	$\sigma = \mathbb{E}(\lvert X-\mu_X\rvert)$?

Исходный вопрос: http://stats.stackexchange.com/q/118.

3.1 Квадрат не единственная используемая функция

В некоторых моделях используется среднее абсолютное отклонение для определения меры разброса случайных величин. Так, при диагностике моделей временных рядов качество прогноза оценивается при помощи нескольких мер: среднее относительное отклонение, среднее абсолютное отклонение, среднее квадратичное отклонение.

Кроме того, стандартное отклонение, определяемое как корень из дисперсии, не является «стандартным» в статистической науке. Точно так же и главные компоненты в одноимённом методе не являются главным инструментом учёного: это всего лишь название.

3.2 Преимущества стандартного отклонения

Если стандартное отклонение призвано измерить разброс данных, то стоит сперва определить этот самый разброс. Функция квадратов отклонений обладает несколькими важными свойствами:

1. Функция квадрата непрерывно дифференцируема;
2. Она является достаточной статистикой для распределения Гаусса;
3. Она является разновидностью L^2-нормы, которая полезна при доказательствах сходимости;
4. Возведение в квадрат смещает вес в сторону бóльших отклонений — со всеми полезными и негативными последствиями.

Обе штрафные функции — квадрат и модуль — всегда возвращают неотрицательную величину, поэтому (за исключением вырожденного случая) сумма штрафов будет положительной.

Само по себе возведение в квадрат трудно интерпретируется, так как некоторые единицы измерения в квадрате (доллары, дни, станки) лишены физического смысла. Для возврата к оригинальным единицам считается квадратный корень из суммы.

Абсолютные отклонения (модули) назначают равные веса наблюдениям из всего диапазона значений, в то время как квадраты усиливают влияние крайних наблюдений. С алгебраической точки зрения работать намного удобнее с квадратами, в то время как модули не дают некоторых свойств (например, дисперсия равна разности мат. ожидания квадрата и квадрата мат. ожидания).

Использование квадратов хорошо интерпретируется через статистический аналог теоремы Пифагора: $c = \sqrt{a^2 + b^2}$. Из неё также следует, что дисперсии независимых случайных величин складываются, а стандартные отклонения — нет.

3.3 Недостатки абсолютного отклонения

Если функция модуля непрерывна всюду на $\mathbb{R}$, то её первая производная — нет (в нуле). Это усложняет аналитическое решение многих задач.

Если в линейной регрессии используется штрафная функция $L(e) = |e|$, то тогда полученная регрессия называется медианной, а в более общем случае ($(1-\alpha)|e|$ для $e < 0$ и $\alpha|e|$ для $e \geq 0$) — квантильной. Вычисление квантилей связано с задачами линейного программирования, которые могут становиться сложнее на порядок. При наличии n точек задача минимизации суммы квадратов решается за время $O(n)$, а суммы модулей — за $O(n \ln n)$, так как самый общий алгоритм подразумевает поиск решения.

3.4 Иные случаи

Предположим, исследователю необходимо измерить очень маленькие величины сравнительно неточным прибором (например линейкой). Так как длина не может быть отрицательной, то распределение будет асимметричным, следовательно, имеет смысл оценивание параметров распределения, дающего положительные значения (бета-, гамма-, Пуассона, Фишера—Снедекора и проч.). Среднеквадратичный разброс будет связан с параметрами этих распределений, однако будет хуже содержательно интерпретироваться.

3.5 Вопрос читателям

Чем бóльшие значения принимает штрафная функция при больших значениях аргумента, тем чувствительнее регрессия к большим выбросам, так как по сути происходит минимизация суммы с большой долей функции от максимальной компоненты. Рассмотрите два примера штрафной функции от остатков:

1. $L(e) = x \cdot \ln(|e| + 1)$;
2. $L(e) = x \cdot \ln^2(|e| + 1)$;
3. $L(e) = \sqrt{|e|}$.

Решите нормальные уравнения для задачи $\min \sum_i L(e_i)$ во всех трёх случаях для парной регрессии вида $y_i = \alpha + \beta x_i + \varepsilon_i$.

Сгенерируйте в любой эконометрической программной среде набор данных с известными свойствами и проведите серию экспериментов Монте-Карло, оценив в каждом случае уравнение регрессии методов наименьших штрафных функций, предложенных выше. Изучите распределение коэффициентов. Измерьте чувствительность коэффициентов к статистическим выбросам. Сравните эти оценки с оценками методов наименьших квадратов и наименьших модулей.

4 Как преобразовывать неотрицательные данные с нулями?

Если данные строго положительные, то в таком случае переменные иногда логарифмируют. Но что делать с неотрицательными данными, в которых присутствуют нули? Если рассматривать преобразование вида $\ln(x + c)$, то следует ли использовать $c = 1$, чтобы нули обратились в нули, либо оценивать $\hat{c}$, либо брать очень малое положительное значение? Есть ли другие преобразования?

Исходный вопрос: http://stats.stackexchange.com/q/1444.

4.1 Причины возникновения нулей

В первую очередь необходимо изучить природу данных и понять, почему некоторые наблюдения содержат нулевые значения. Каждую из нижеслеюующих причин необходимо рассматривать в отдельности:

1. Усечение или цензурирование данных (исследователь не располагает отрицательными наблюдениями, хотя они могли бы быть);
2. Пропущенные наблюдения (если переменная содержательно должна быть положительной: цена, длительность и проч.);
3. Естественный ноль (доход индивида может быть равен нулю, если он безработный);
4. Специфика чувствительности средства измерения переменной (инструмент не реагирует на количества, меньшие определённого порога).

Предлагаемые решения:

1. Использование моделей, учитывающих информацию об ограниченных значениях переменной (модель Хекмана, интервальная регрессия, модели времени жизни);
2. Выбрасывание из модели наблюдений, содержащих пропуски, если учтены все возможные последствия этого решения;
3. Числовое преобразование, упоминавшееся в вопросе;
4. Использование специальных LOD-моделей (Limit of Detection), непараметрических методов, а в первую очередь — изучение книги [Helsel, 2005].

С точки зрения регрессионного анализа может быть полезно добавить дамми-переменную для наблюдений с нулями. Последний случай является самым тяжёлым, причём в эконометрике он практически не встречается (этот вопрос более актуален для специалистов, снимающих показания с реальных датчиков, обладающих порогом чувствительности).

4.2 Числовое преобразование

В работе [Smithson, Verkuilen, 2006] предлагается использовать преобразование

$$x' = \frac{x(N-1)+s}{N}, \tag{4.1}$$

где N — число наблюдений, а s — априорная вероятность для бета-распределения (зачастую наиболее практично использовать $s = 0{,}5$).

Кроме того, не следует забывать о преобразованиях Бокса—Кокса:

$$y^*(\lambda_1) = \begin{cases} \frac{y^{\lambda_1}-1}{\lambda_1}, & \lambda_1 \neq 0, \\ \ln y, & \lambda_1 = 0; \end{cases} \qquad y^*(\lambda_1,\lambda_2) = \begin{cases} \frac{(y+\lambda_2)^{\lambda_1}-1}{\lambda_1}, & \lambda_1 \neq 0, \\ \ln(y+\lambda_2), & \lambda_1 = 0. \end{cases}$$

Сами Бокс и Кокс в статье [Box, G. E. P., Cox, D. R., 1964] приводят алгоритм, позволяющий численно найти значения $\hat{\lambda}_1$ и $\hat{\lambda}_2$, максимизирующие значение

функции правдоподобия, которая зависит не только от вектора зависимой переменной $\boldsymbol{y}$, но и от матрицы наблюдений $\boldsymbol{X}$. Существует пакет geoR для **R**, позволяющий оценить оба параметра, хотя при отсутствии возможности подогнать двухпараметрическое преобразование под данные можно воспользоваться значением $\lambda_2 = 0{,}5 \min\{y_i \colon y_i \geq 0\}$ или рекомендацией «первый квартиль в квадрате делить на третий квартиль» [Stahel, 2013].

Преобразование Бокса—Кокса позволяет не только решить проблему нулей в данных, но и приблизить спецификацию к линейной.

Кроме того, существует альтернатива преобразованию Бокса—Кокса — гиперболический арксинус:

$$y^*(\theta) = \frac{\operatorname{arsh} \theta y}{\theta} = \frac{\ln(\theta y + \sqrt{\theta^2 y^2 + 1})}{\theta},$$

где $\theta > 0$. В статье [Burbidge, Magee, Robb, 1988] приводится метод оценивания параметра $\hat{\theta}$, а также предлагается (без формул) преобразование со сдвигом, похожим на сдвиг в преобразовании Бокса—Кокса, т. е. $y^*(\theta, \omega) = \frac{\operatorname{arsh} \theta(y+\omega)}{\theta}$.

4.3 Дешёвые и сердитые приёмы

Если нулей совсем немного и можно предположить, что они возникли в данных случайно, то можно оценить модель с нулями, затем удалить проблемные точки и взять логарифмы. Также можно добавить небольшое $\hat{c}$, рекомендуемые значения для которого приведены в предыдущем пункте. Если результаты оценивания моделей почти не отличается, следовательно, модель довольно устойчива, поэтому следует выбирать тот набор данных, который позволяет сделать хорошо интерпретируемый вывод (пример: «в данных присутствовало 2 % безработных с нулевым доходом, однако их исключение не повлияло на значимость коэффициентов и не изменило двух первых значащих цифр оценок»).

Наконец, если требуется сделать монотонное преобразование $f(x)$ с $f' > 0$, $f'' < 0$, почему бы не рассмотреть квадратный корень? Если есть отрицательные значения, то можно использовать кубический корень.

4.4 Более сложные модели

Если данные непрерывные, то при наличии в них нулей следует обратить внимание на распределение значений: дискретный пик означает, что по той или иной причине малые значения были округлены до нуля. В любом случае, наиболее подходящие модели для данных, в которых присутствует большое количество нулей, должны учитывать вероятность того, что наблюдение будет нулём, а также условное распределение ненулевых значений (например, марковские модели для смешанных распределений, обобщённые линейные

модели для дискретных данных, байесовские методы, симуляции Монте-Карло и проч.).

Список литературы

Box, G. E. P., *Cox, D. R.* An Analysis of Transformations // Journal of the Royal Statistical Society. Series B (Methodological). — 1964. — Т. 26, № 2. — С. 211—252.

Burbidge J. B., *Magee L.*, *Robb A. L.* Alternative Transformations to Handle Extreme Values of the Dependent Variable // Journal of the American Statistical Association. — 1988. — Т. 83, № 401. — С. 123—127.

Helsel D. R. Nondetects and data analysis: statistics for censored environmental data. — Wiley-Interscience, 2005. — (Statistics in practice).

Smithson M., *Verkuilen J.* A better lemon squeezer? Maxium-likelihood regression with beta-distributed dependent variables // Psychological Methods. — 2006. — Т. 11, № 1. — С. 54—71.

Stahel W. A. Statistische Datenanalyse: Eine Einführung für Naturwissenschaftler. — Springer-Verlag, 2013.

QR-код, или немного о дополненной реальности

Анастасия Игнатьева*

21 июня 2015 г.

Аннотация

Статья знакомит читателей с одной из наиболее известных разновидностей дополненной реальности — QR-кодами. Помимо основных технических характеристик, вы сможете узнать, как устроен QR-код, понять алгоритмы, которые используются при шифровании информации, а также, как декодировать QR-код вручную.

Ключевые слова: дополненная реальность, QR-код, распознавание образов.

Что позволяет порядочному исследователю творить? Совесть? Возможно, но здесь не об этом, здесь всё серьёзно. Это *данные*. Сегодня они нужны всем, ведь любой эконометрический анализ, как и множество исследований, без этого ключевого элемента становится невозможным или остаётся узником чистой и неприменимой теории. Вот так в поисках релевантных временных рядов, всевозможных панелей и просто данных мы скитаемся по интернету: fira.ru, незаменимый gks.ru, различные базы OECD, RUSLANA, СПАРК… Впрочем, это не новость, собственно, и заметка не совсем об этом. Всё дело в том, что мало кто замечает: данные повсюду, нужно только заглянуть несколько глубже, заглянуть в дополненную реальность.

После того как необходимая, хоть и минимальная, отсылка к эконометрике, ввиду тематики журнала, была соблюдена, самое время поговорить об этой самой дополненной реальности, где окружающий нас мир соприкасается с миром виртуальным.

Почему именно дополненная реальность? Просто мне всегда казалось, что это слишком сложно, чтобы быть правдой. Возможно, для девушек это вполне нормально, когда компьютер почти как магия.

Всё началось с машинного зрения. Для человека зрение настолько естественно, что большинство просто не задумывается, что кого-то, в данном случае что-то, нужно этому учить. Хотя многие современные компьютеры выглядят совсем не глупее людей, научить их видеть — чрезвычайно непростая задача. Они должны не только уметь различать цвета, идентифицировать предметы, определять их границы и классифицировать, но и учитывать контекст, внутриклассовую изменчивость, масштаб, освещение, возможную деформацию при движении, изменении ракурса и положения, отличать, к примеру, отбрасываемую тень от самого предмета.

Однако мир не без умных людей, и современные алгоритмы так или иначе позволяют решать эти проблемы. Сфера применения компьютерного зрения весьма обширна: системы моделирования объектов и окружающей среды, медицинских изображений

*НИУ ВШЭ, Москва.

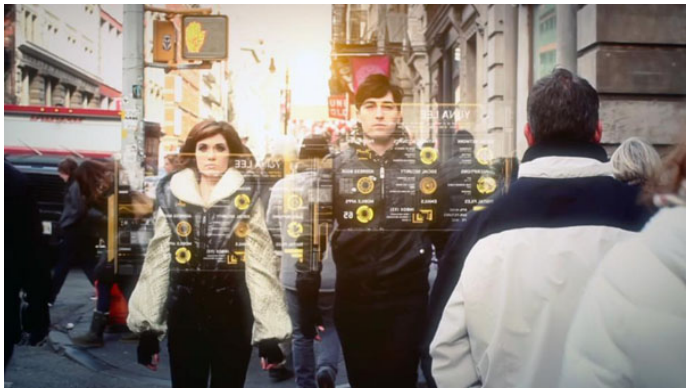


Рис. 1. Дополненная реальность

(рентген, томография), системы видеонаблюдения и организации информации, а также системы дополненной реальности и проч. С дополненной реальностью, несмотря на всю загадочность названия, сталкивался каждый, кто хоть раз смотрел различные спортивные мероприятия, будь то теннис, когда в случае спорных моментов моделируется траектория полёта мяча, или футбол, где при определении офсайда возникает линия, параллельная лицевой, которая позволяет определить ближайшего к воротам игрока.

Одна из разновидностей дополненной реальности — это всем известные QR-коды и штрихкоды. Несложно догадаться, где́ именно черпал вдохновение создатель штрихкодов: да, вы правы, в азбуке Морзе. Однако линейные штрихкоды вмещают слишком мало информации — по этой причине в 1994 году в Японии и были изобретены двухмерные, или матричные коды, самым популярным из которых и стал QR-код, что означает «быстрый отклик» (Quick Response). Если у обычных штрихкодов объём памяти не превышал 100 байт, то у матричных кодов данный показатель значительно выше — до 2048 байт (Википедия, 2014a); более того, информация может быть считана даже при 30%-м повреждении метки!

Сегодня QR-коды можно встретить где угодно, даже на кладбищах, где данные коды используются для хранения информации об усопшем. Такое нестандартное решение относительно применения QR-кодов было найдено в Японии и Австралии, впрочем, вернемся к основным техническим характеристикам. Самый маленький QR-код имеет размер 21×21 пиксель, в то время как самый большой (версия 40) — 177×177 пикселей. Что касается кодировки QR-кодов, то существует 4 основных типа: цифровая (до 7089 цифр), алфавитно-цифровая (до 4296 символов), байтовая (до 2953 байт), кандзи (до 1817 иероглифов) (Википедия, 2014b).

Каждому не раз приходилось сталкиваться с QR-кодами в жизни, некоторые даже использовали смартфоны, чтобы считать код. Но вряд ли кому-то приходило в голову делать это вручную. А так как в жизни бывают разные ситуации, почему бы не восполнить данный пробел?

Для начала разберёмся, как устроен QR-код. Изначально данные, которые нужно закодировать, разбиваются на блоки в зависимости от режима кодирования; далее прибавляется заголовок, указывающий режим и количество блоков. Безусловно, существуют режимы с более сложной структурой кодирования, однако из них весьма

проблематично извлекать информацию вручную, поэтому остановимся на более простых случаях. После того как записаны все информационные данные, к ним добавляются корректирующие ошибки коды Рида—Соломона (RS), которые позволяют исправлять недочёты при чтении. Именно эти коды и занимают большую часть QR-матрицы. Перед записью в картинку данные с RS-кодами перемешиваются, для чего используются специальные маски. Среди имеющихся восьми алгоритмов, которые представлены на рис. 2, выбирается наилучший, который определяется за счёт системы штрафов. После этой процедуры перемешанные данные записываются на шаблонную картинку, к которой добавляется техническая информация для декодирующих устройств (Хабрахабр, 2011).

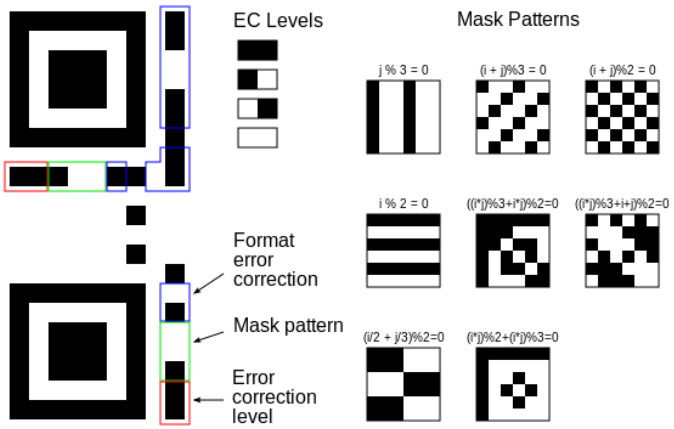


Рис. 2. Восемь алгоритмов масок

Возможно, многие замечали, что QR-код можно разбить на несколько областей, у каждой из которых индивидуальные функции (показано на рис. 3). Так вот, три квадрата в углах изображения и меньшие синхронизирующие квадратики по всему коду — техническая информация для декодирующих устройств, которая позволяет нормализовать размер изображения и его ориентацию, а также угол, под которым сенсор расположен к поверхности изображения. Таким образом, как вы можете догадаться, эта область абсолютно неинтересна для нас, так как не содержит никакой информации о скрывающемся за кодом послании. Что касается полезной части кода, то её можно разделить на две области: область, отвечающая за системную информацию, и непосредственно данные. Также в матрице содержится информация о версии кода, от которой зависит ёмкость последнего. Так, при повышении версии добавляются специальные блоки; при высоких версиях кода не рекомендуется считывать его вручную.

Системная информация представляет собой 15 бит данных, из которых только 5 бит для нас значимы: 2 бита отвечают за уровень коррекции ошибок, а оставшиеся 3 — за применяемую к данным маску. Ещё 10 бит данных — это BCH-код, так называемый код Боуза-Чоудхури-Хоквингема, который относится к широкому классу циклических кодов и позволяет исправлять ошибки в системных данных. Упомянутые ранее RS-коды, коды Рида—Соломона, также относятся к классу BCH. Помимо всего прочего, для дополнительной защиты системной информации используется статическая маска, применяемая к любой системной информации. Она имеет запись

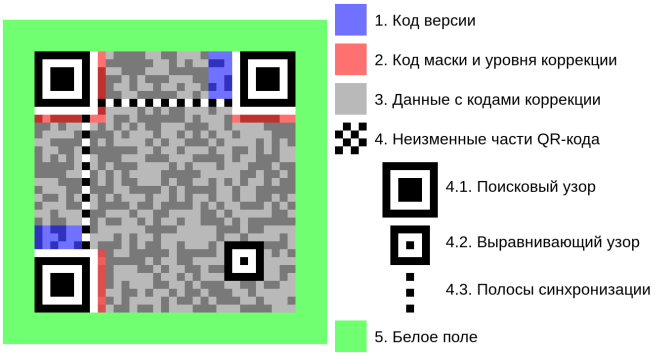


Рис. 3. Области QR-кода

101010000010010. Так как нам интересны только первые 5 бит, то маску можно сократить, и её уже не так сложно запомнить: 10101. Как видно из рисунка, системная информация, отмеченная красным цветом, дублируется, что позволяет значительно понизить вероятность возникновения ошибок.

Таким образом, **первый шаг — чтение первых 5 бит системной информации** (рис. 4).

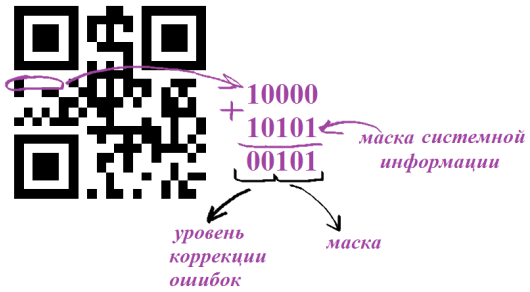


Рис. 4. Чтение первых 5 бит информации

Для лучшего понимания воспользуемся кодом, сгенерированным на qrcoder.ru. Получили, что для нашего кода уровень коррекции ошибок — 00 — уровень M, позволяющий скорректировать до 15 % ошибок, маска — 101, соответствующая 8-й схеме на рисунке 2. Все возможные варианты масок и уровней коррекции представлены в таблице 1.

Шаг второй — определение режима кодировки.

Чтобы понять, с какими данными предстоит иметь дело, необходимо изначально прочитать 4-битный заголовок, который содержит в себе информацию о режиме. Заголовок находится в правом нижнем углу матрицы, причём читать его надо змейкой, начиная справа. После извлечения четырёх бит, описывающих режим, необходимо применить к ним маску. Маска определяется выражением, приведённым в таблице — в нашем случае

$$(ij) \bmod 2 + (ij) \bmod 3 = 0.$$

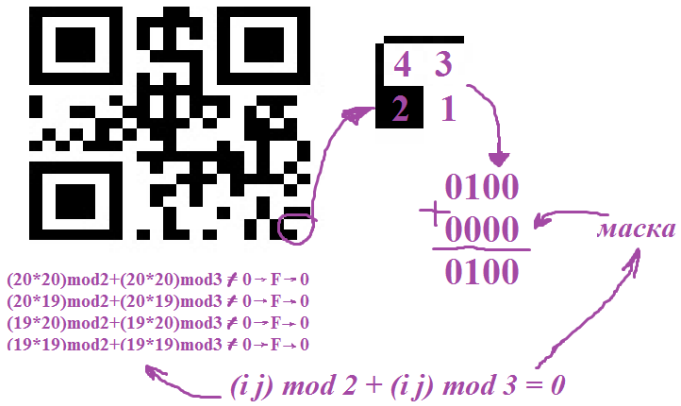


Рис. 5. Определение режима кодировки

Если данное выражение сводится к TRUE для бита с координатами $(i; j)$, то бит инвертируется, иначе всё остаётся без изменений. Начало координат — в левом верхнем углу, $(0; 0)$; в матрице 21×21 бит, т. е. квадрат; таким образом, бит, находящийся в правом нижнем углу, имеет координаты $(20; 20)$. Получили 0100, что соответствует 8-битному режиму.

Шаг третий — чтение данных.

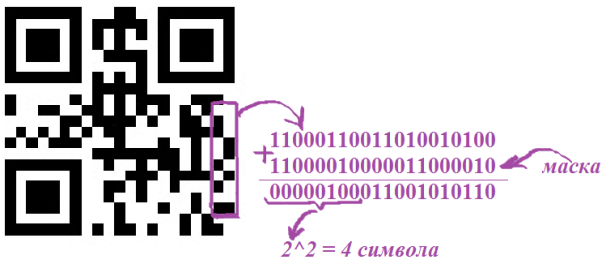


Рис. 6. Чтение данных

Необходимость определения режима кодирования обуславливается тем, что от него зависит длина блоков данных, которая также варьируется для различных версий кода. Для версий кода 1–9 в числовом режиме используются 10- или 4-битные блоки (последние — если в 10-битном объёме нет необходимости); в буквенно-числовом режиме — 9-битные блоки; в 8-битном (байтном) режиме — 8-битные блоки. Первый блок после указателя режима — это количество символов. Таким образом, для определения количества символов расшифровываем следующие 8 бит кода (змейкой, начиная справа) и применяем маску. Видно, что в коде зашифровано 4 символа, поэтому необходимо перейти к чтению следующего столбца для извлечения всех четырёх блоков информации.

Снова считываем данные по такому же алгоритму. Главное отличие: биты надо отсчитывать змейкой, но с правого верхнего угла. Далее полученный набор из нулей и единиц делим на 4 блока по 8 бит в каждом. Текст в 8-битном режиме многие онлайн-генераторы QR-кодов кодируют, используя ASCII. Таким образом, первые

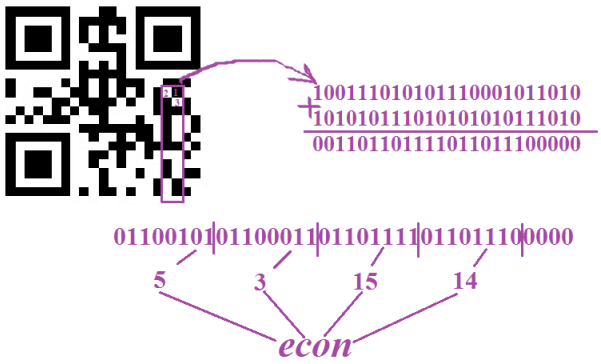


Рис. 7. Расшифровка ASCII-кода

4 элемента блока указывают на регистр, 0110 соответствует нижнему регистру букв, 0100 — верхнему, вторые 4 элемента — это число, равное номеру буквы в алфавите. Как можно заметить, все наши буквы строчные. Получаем числа 5, 3, 15 и 14 и находим нужные нам буквы. Так, в коде была зашифрована часть слова, а именно **econ**, а какого именно слова — economics, или econometrics, или что-то ещё — пусть каждый решает для себя сам!

Список литературы

Википедия Data Matrix. — 28 сент. 2014a. — URL: http://ru.wikipedia.org/?oldid=65832511 (дата обр. 07.12.2014).

Википедия QR-код. — 9 дек. 2014b. — URL: http://ru.wikipedia.org/?oldid=67249340 (дата обр. 11.12.2014).

Хабрахабр Читаем QR-код. — 28 авг. 2011. — URL: http://habrahabr.ru/post/127197/ (дата обр. 11.12.2014).

Возможные маски	
000	$(i + j) \bmod 2 = 0$
001	$i \bmod 2 = 0$
010	$j \bmod 3 = 0$
011	$4(i + j) \bmod 3 = 0$
100	$((i \operatorname{div} 2) + (j \operatorname{div} 3)) \bmod 2 = 0$
101	$(ij) \bmod 2 + (ij) \bmod 3 = 0$
110	$((ij) \bmod 2 + (ij) \bmod 3) \bmod 2 = 0$
111	$((i + j) \bmod 2 + (ij) \bmod 3) \bmod 2 = 0$
Возможные уровни коррекции ошибок	
01	L
00	M
11	Q
10	H
Возможные режимы	
0111	ECI
0001	Числовые
0010	Буквенно-числовые
0100	8-битный (байтный)
1000	Kanji
0011	Структурированное дополнение
0101	FNC1 (1-я позиция)
1001	FNC1 (2-я позиция)

Таблица 1. Варианты масок и уровней коррекции

Сгладить, нельзя не сгладить! Пара слов о методах сезонной корректировки

Иван Станкевич*

21 июня 2015 г.

Аннотация

В статье приводится обзор двух самых популярных на сегодняшний день процедур корректировки сезонности — методов семейства X-11 и TRAMO/SEATS. Описывается история развития методов, механизм их работы, указываются их основные достоинства и недостатки. Рассматривается реализация этих методов в компьютерных пакетах, в том числе R.

Ключевые слова: сезонность, R, X-11, X-12, TRAMO/SEATS.

1 В качестве введения

1.1 Зачем нужна сезонная корректировка

Сезонность — очень распространенное в экономических (и не только) временных рядах явление, которому, увы, часто не уделяется достаточного внимания. Сезонность представляет собой регулярные колебания в данных, повторяющиеся из года в года примерно в одно и то же время. Что же является причиной сезонности? Прежде всего, это климатические и институциональные факторы, такие как выходные и праздничные дни (вспомним наши 10-дневные новогодние праздники в начале января, или 10 дней в начале мая, которые значимая часть населения страны проводит не на работе, а на даче, занимаясь сельскохозяйственными работами). Климат влияет на сельское хозяйство (тривиальным образом), потребление и производство электроэнергии (отопление зимой, в жарких и не только странах — кондиционирование летом), туризм, строительство; в меньшей степени — на добычу полезных ископаемых и отчасти на отдельные виды торговли и услуг. Некоторые авторы также

*НИУ ВШЭ, Москва.

выделяют как отдельную причину ожидания сезонности, которые в свою очередь провоцируют колебания, либо служат их усилению.

Может возникнуть резонный вопрос, ответ на который не очевиден с первого взгляда, — зачем с ней бороться? На то есть целый ряд причин:

- Невозможность наблюдать и изучать тренды (особенно короткие, локальные тренды) в рядах с сильной сезонной компонентой. Проблема здесь в том, что сезонность, особенно сезонность с достаточно большой амплитудой, зачастую просто «забивает» тренд и рост или падение, к примеру, сельскохозяйственного производства на пару-тройку миллиардов рублей может просто пропасть на фоне сезонных колебаний размахом в сотни миллиардов.
- Высокая вероятность получения кажущихся зависимостей при оценке связи между рядами с сильной сезонностью (аналогично кажущимся регрессиям для нестационарных рядов). Если пики двух показателей приходятся на один квартал — отлично, корреляция будет большая и положительная, на разные — вполне возможно, что не менее большая и отрицательная. Но это, понятное дело, не говорит о наличии какой-то реальной зависимости между рядами.
- Проблемы с оценками коэффициентов из-за высокой дополнительной дисперсии. Чем выше дисперсия, тем, как мы знаем, ниже точность получаемых оценок, а чем ниже точность, тем хуже работают наши модели!
- Трудности с обнаружением, анализом и предсказанием изломов в трендах и длинных циклах, особенно при использовании высокочастотных (месячных и чаще) данных. Проблема та же самая, что и с наблюдением коротких трендов, но в контексте не трендов, а их изломов. Кризис случился, а никто его и не заметил...

Идейно есть два основных подхода к тому, что делать с сезонностью в данных (вариант «не обращать внимание» не рассматривается). Первый — это сезонная корректировка. При помощи тех или иных процедур из ряда явным образом выделяется сезонная компонента, которую после этого можно удалить и работать с данными, как обычно (к примеру, практически все статистические службы мира, в том числе и Росстат[1], публикуют основные макроэкономические ряды не только в исходном виде, но и в сглаженном). Второй подход — это моделирование сезонности, когда сезонные колебания учитываются при построении модели явным образом. К сожалению, сделать это получается не для всех моделей.

Сезонная корректировка — самый простой способ работы с сезонностью в данных (особенно если корректировку проводит статистическая служба, а не

[1] Правда, Росстат публикует в сглаженном виде не так много рядов и обновляет их не каждый квартал, а раз в год

исследователь), потому что не требует никаких дополнительных усилий со стороны исследователя (изучение структуры сезонности, изменение формы модели для учета сезонного фактора и т.д.). Оставляя в стороне самые простые варианты (классическую сезонную декомпозицию скользящими средними и взятие сезонных разностей — методы простые и хорошо работающие в некоторых приложениях, но не учитывающие многие нюансы, часто возникающие при работе с реальными данными разной природы), рассмотрим подробнее более современные методы.

1.2 Общие предпосылки

Основное предположение об устройстве данных, которое делается процедурами сезонной корректировки и которое лежит, в конечном итоге, в основе всех методов удаления сезонности, заключается в следующем. Возьмём ряд данных Y_t. Тогда при использовании аддитивной модели сезонности верно равенство:

$$Y_t = U_t + S_t + E_t$$

Либо, при использовании мультипликативной модели сезонности:

$$Y_t = U_t S_t E_t$$

Где U_t — трендовая компонента данных (долгосрочные тенденции), S_t — сезонная компонента (циклы с периодом в год), E_t — случайная составляющая (просто шум). Иногда в эту схему ещё добавляется четвертая компонента, отвечающая за более длинные циклы.

В целом, такая схема предполагает, что данные можно разбить на три компоненты либо мультипликативно, либо аддитивно. Другие формы включения сезонности, как правило, не рассматриваются. В такой постановке задачи цель любой процедуры выделения сезонности — максимально точно оценить компоненту S_t, чтобы потом удалить её из данных. Надо понимать, что удаление сезонности, которое также иногда называют сглаживанием сезонности, — не совсем то же самое, что сглаживание данных. Сглаживание осуществляется с целью получения максимального гладкого ряда, где не будет не только периодических колебаний (хотя они как раз могут и остаться, в зависимости от того, как именно осуществляется сглаживание), но и шума (прежде всего шума). Сезонная корректировка же выделяет и удаляет только сезонную компоненту, случайную же она сохраняет, хотя современные процедуры, конечно, дают оценку и трендовой, и случайной составляющей по отдельности, что позволяет получать в том числе и гладкий тренд без шума.

Теперь, поняв зачем нужна сезонная корректировка и в чем она, в конечном счете, заключается, можно переходить к описанию основных процедур, использующихся на практике. В этом небольшом обзоре сосредоточимся

на двух наиболее часто использующихся: методах семейства X-11 и методе TRAMO/SEATS.

2 Методы семейства X-11

2.1 Предыстория и X-11

Методы семейства X-11 начали разрабатываться в US Bureau of Census ещё в далёкие 1950-ые годы, когда компьютеров было мало, а вычислительные мощности были серьёзно ограничены, что, в общем-то, наложило свой отпечаток на всё семейство процедур. Долгое время эти методы удаления сезонности из данных оставались основным инструментом и, во многом, остаются и по сей день.

Общая идея процедуры достаточно проста - за несколько шагов, используя специальный набор фильтров и достаточно простой алгоритм, выделить оценки трендовой и сезонной компоненты ряда. Предлагались следующие шаги:

- На первом шаге осуществляется сглаживание для получения первоначальной оценки тренда. Здесь и далее — при помощи скользящей средней Хендерсона, про неё подробнее ниже. Количество точек, по которому осуществляется сглаживание, зависит от частоты ряда (квартальный, месячный) и его характеристик. Полученная первоначальная оценка тренда удаляется из данных (в форме для мультипликативной сезонности): $\frac{Y_t}{\hat{U}_t} = \frac{U_t S_t E_t}{\hat{U}_t} \approx S_t E_t$
- На втором шаге полученный на первом шаге ряд сглаживается для получения первоначальной оценки сезонности. Она удаляется из данных, получаем ряд: $\frac{Y_t}{\hat{S}_t} = \frac{U_t S_t E_t}{\hat{S}_t} \approx U_t E_t$
- На третьем шаге ряд, очищенный от первоначальной сезонности, сглаживается для получения второй оценки тренда, которая также удаляется из данных
- Четвертый шаг — аналог второго, полученная оценка сезонности рассматривается как финальная
- На пятом шаге из данных удаляется финальная оценка сезонности. Ряд сглаживается для получения финальной оценки тренда.

В итоге получается разбиение ряда на трендовую компоненту, сезонную и нерегулярную. Шум — то, что осталось после удаления из данных сезонности и тренда.

Есть ряд технических нюансов, которые следует помнить при работе с процедурами семейства X-11.

Прежде всего, скользящие средние. В фильтрах X-11 используются сколь-

зящие средние Хендерсона[2], смысл которых в том, что если применить их к кубическому полиному (сгладить кубический полином при помощи скользящих средних Хендерсона), он не поменяется (сглаженный скользящими средними Хендерсона кубический полином должен быть идентичен тому же самому полиному, но не сглаженному). При этом веса у дальних наблюдений оказываются отрицательными (сумма всех весов, разумеется, равна единице). Как правило, считается, что кубического полинома (локального, конечно же, не в применении ко всему, возможно очень длинному, ряду) достаточно для описания экономических временных рядов, что оправдывает использование именно такого набора скользящих средних, и одновременно с этим скользящие средние Хендерсона достаточно эффективно удаляют высокочастотные шумы. Стоит заметить и ещё одну очень важную вещь - сглаживание на краях ряда осуществляется с использованием несимметричных (за отсутствием данных за пределами периода наблюдения) фильтров, что может порождать некоторые смещения (решение этой проблемы было предложено в версии X-11-ARIMA, о которой будет рассказано чуть ниже).

Помимо собственно фильтра, в X-11 уже были встроены механизмы для борьбы с эффектом числа торговых дней (разное количество рабочих дней в разные месяцы и кварталы), с эффектами начала года и рядом других календарных эффектов, которые также оказывают заметное влияние на структуру сезонности.

2.2 X-11-ARIMA

Метод X-11-ARIMA, предложенный Statistics Canada в 1980 году, предложил решение одной из самых серьёзных проблем исходного X-11 — проблемы краевых точек. Вместо использования несимметричных фильтров предлагалось достраивать ряд с концов при помощи оцененной по имеющимся данным ARIMA-модели и использовать обычные симметричные фильтры.[3]

Подбор модели (заметим, что 1980 год был достаточно давно и вычислительные мощности по-прежнему были довольно ограниченными) осуществлялся достаточно любопытным образом. В процессе разработки метода, исследователи применяли большое количество моделей к большому количеству макроэкономических рядов, и выбрали среди этих моделей 5 наиболее универсальных, хорошо работающих в большинстве случаев (хотя бы одна из них хорошо работала в подавляющем большинстве случаев). Вот эти модели:

[2] Henderson, R. (1916). Note on Graduation by Adjusted Average. Transactions of the American Society of Actuaries, 17, 43-48

[3] Можно, конечно, заметить, что все достроенные точки представляют собой не более чем линейную комбинацию старых, а значит и «симметричный» фильтр оказывается на самом деле несимметричным, но с весами, определяемыми по модели, а не заданными свыше

ARIMA(0,1,1) x SARIMA(0,1,1)
ARIMA(0,1,2) x SARIMA(0,1,1)
ARIMA(2,1,0) x SARIMA(0,1,1)
ARIMA(0,2,2) x SARIMA(0,1,1)
ARIMA(2,1,2) x SARIMA(0,1,1)

Где в записи ARIMA(p,d,q) x SARIMA(P,D,Q): p — количество обычных AR лагов, P — количество сезонных AR лагов (первый сезонный лаг — 4-ый лаг для квартальных данных, 12-ый для месячных и так далее); аналогично d и D — разности обычные и сезонные; q и Q — MA лаги. К примеру, модели ARIMA(0,1,1) x SARIMA(0,1,1) для месячных данных будут записываться следующим образом:

$$(1-\Delta)(1-\Delta^{12})Y_t = (1-\beta_1\Delta)(1-\beta_2\Delta^{12})\varepsilon_t$$

Метод X-11-ARIMA использует следующий алгоритм получения скорректированного ряда:

- На первом шаге очищает данные от календарных эффектов — торговые дни, начало года, «гуляющие» праздники типа Пасхи и т.д.
- На втором — строит по данным ARIMA-модели. При этом процедура оценивает все пять моделей из списка, оставляет те из них, которые удовлетворяют минимальным критериям по точности, значимости статистик Бокса-Льюнга (они должны быть незначимыми для лагов, соответствующих 2 годам — 8 кварталов или 24 месяца) и не передифференцируют (не берут слишком много разностей) ряды. Из моделей, прошедших отбор, выбирается лучшая на основе критерия MAPE (Mean Absolute Percent Error): $MAPE = 100\%\frac{1}{n}\sum_{t=1}^{n}\frac{|y_t-\hat{y}_t|}{|y_t|}$. При этом прогнозы каждый раз строятся на 1 шаг вперед, потом ряд дополняется одной точкой из реальных данных и опять строится прогноз на 1 шаг вперед.
- Ряд достраивается с краев до длины, достаточной для использования обычного симметричного фильтра X-11
- Наконец, фильтр X-11 применяется к достроенному ряду, получаются оценки трендовой, сезонной и нерегулярной компонент.

2.3 X-12-ARIMA и дальнейшее развитие

Следующей важной вехой на пути развития методов семейства X-11 был метод X-12-ARIMA (который используется и по сей день), предложенный уже в 1990 году. Главным нововведением было использование моделей regARIMA:

$$\ln\left(\frac{Y_t}{D_t}\right) = \beta' X_t + Z_t$$

Где Y_t — ряд данных, D_t — преобразования в данных (какие-то априорные сведения о структуре сезонности, которыми мы располагаем), X_t - матрица регрессоров (количество торговых дней, дамми на праздники, в том числе плавающие, и другие календарные эффекты — использование этих переменных внутри модели и есть главное нововведение моделей regARIMA), Z_t — ARIMA процесс.

Модели regARIMA - логичное продолжение идей метода X-11-ARIMA, позволяющее в рамках одной конструкции учесть и календарные эффекты, и оценить по данным ARIMA (SARIMA) модель. При помощи этой же модели осуществляется поиск выбросов, достраиваются пропуски в данных (возможность работать с данными с пропусками также является важным достижением метода), явно оцениваются календарные эффекты, что также может представлять интерес. При этом выбор модели может осуществляться как аналогично модели X-11-ARIMA, из списка (правда, здесь он заметно больше), так и автоматически — начиная с самой простой модели с постепенным добавлением лагов и разностей, пока это улучшает качество модели. Такой подход позволяет не ограничивать выбор моделей каким-то списком, что потенциально делает метод применимым ещё в большем количестве ситуаций, в том числе достаточно экзотических.

Помимо этого, в пакет встроено большое количество дополнительных функций для диагностики данных (спектральный анализ, средства для анализа стабильности сезонности и качества работы сезонной корректировки), что также можно отнести к достоинствам пакета, потому что это экономит время и силы исследователя.

Последняя на сегодня версия пакета — X-13-ARIMA-SEATS[4]. Если не получается на него зайти (из России иногда возникают проблемы с доступом к сайту census.gov) — можно попробовать воспользоваться прокси[5] или Tor — в целом сохраняет функционал X-12-ARIMA, но добавляет к нему возможность пользоваться методом SEATS (о нём чуть ниже) из того же пакета.

3 TRAMO/SEATS

Процедура TRAMO/SEATS (строго говоря, не столько процедура, сколько набор из двух программ — собственно TRAMO и SEATS, — которые предлагается использовать в комплексе) была предложена центральным банком Испании (Bank of Spain) в 1996 году. Она основана на идеях, заметно отличающихся от использующихся в методах семейства X-11. Прежде всего — это подбор фильтра на основе данных, а не использование готового универсального решения, как в X-11. Две части метода решают две разные задачи.

[4] Сама программа, её документация, новости и много чего ещё доступно на сайте US Bureau of Census, https://www.census.gov/srd/www/x13as/

[5] Например, https://hide.me/en/proxy или https://www.hidemyass.com/proxy

TRAMO (Time Series Regression with ARIMA Noise, Missing Observations and Outliers) по сути своей аналогичен моделям regARIMA. Здесь так же строится регрессия исходных данных на календарные переменные с остатками в форме ARIMA, так же определяются и корректируются выбросы в данных.

SEATS (Signal Extraction in ARIMA Times Series) значительно интереснее. Она рассматривает остатки Z_t из regARIMA (TRAMO) как сумму трёх компонент (вспомним аддитивную модель сезонности):

$$Z_t = U_t + S_t + E_t$$

Как и раньше, U_t — тренд, S_t — сезонность, E_t — нерегулярная компонента.

И вводит ряд дополнительных предпосылок (ортогональность U_t и S_t, белошумность E_t). Тогда, зная модель для Z_t (из оценок TRAMO), процедура получает в явном виде модели для U_t и S_t. Это делается не единственным образом, но можно отобрать «лучшую» по какому-то критерию — здесь минимизируется шум в S_t. Полученные таким образом модели можно использовать для фильтрации рядов.

Более подробно описывать устройство TRAMO/SEATS в рамках обзорной статьи смысла нет, но если возникнет необходимость более глубоко погрузиться в нюансы его работы — стоит ознакомиться с документацией на сайте Банка Испании[6], и со статьями, посвященными этому вопросу: хорошая подборка есть там же, на сайте Банка Испании[7].

4 Общие слова

4.1 О преимуществах и недостатках методов

Методы семейства X-11 и TRAMO/SEATS представляют два основных направления в области сезонной корректировки — использование наборов готовых фильров и определение оптимального фильтра по данным. Каждый из подходов имеет свои достоинства и недостатки.

Методы X-11 считаются более гибкими, потому что используют непараметрический подход, не специфицируя форму сезонности ни в какой форме, тогда как SEATS предполагает, что сезонность задаётся линейной ARIMA моделью. Поэтому метод TRAMO/SEATS, как раз благодаря сравнительно жёсткой параметризации зависимостей, может оказаться более точным в некоторых ситуациях. Последние версии американской процедуры также предлагают большое количество дополнительных функций для диагностики сезонности в

[6] http://www.bde.es/bde/en/secciones/servicios/Profesionales/Programas_estadi/Notas_introduct_3638497004e2e21.html, правда, стоит заметить, что качество этой документации оставляет желать лучшего

[7] http://www.bde.es/bde/en/secciones/servicios/Profesionales/Programas_estadi/Trabajos_sobre__0ac07f3710fd821.html

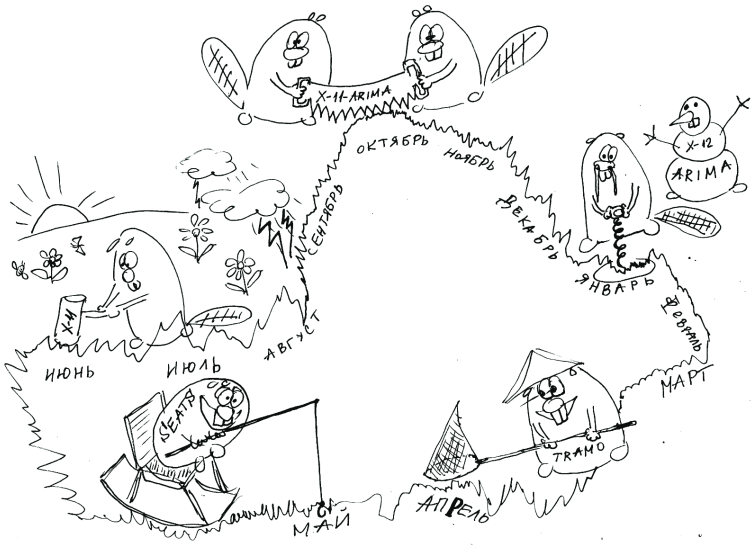


Рис. 1. Бобры выполняют сезонную корректировку

данных, чем не может похвастаться процедура европейская. При этом считается, что на практике TRAMO/SEATS иногда даёт более гладкие результаты, и более устойчивые к добавлению новых точек, а X-11 лучше работает на сравнительно коротких (до 5-6 лет) временных рядах.

При этом для многих сравнительно простых случаев разница между X-12 и TRAMO/SEATS остается практически незаметной. К примеру, на графике 1 приведен квартальный ВВП России в текущих ценах в исходном виде и скорректированный X-12-ARIMA и TRAMO/SEATS, а на рисунке 2 — расхождение в рядах, скорректированных этими двумя методами. Как видно, расхождения очень невелики, хотя и носят очевидно неслучайный характер, обусловленный разными свойствами методов. Следовательно, и больших различий в выводах, полученных по этим данным, скорее всего, не будет.

В целом же, выбор лучшего метода для сезонной корректировки остаётся процессом во многом творческим и требующим как понимания методов, так и вдумчивой работы с данными.

4.2 Реализация в компьютерных пакетах

Из коммерческих пакетов в EViews 8 доступны и X-13, и TRAMO/SEATS; в STATA, насколько известно автору, доступность ограничивается версией X-12-ARIMA, TRAMO/SEATS также доступен.

В R доступны все методы и в разных версиях. Есть отдельный пакет **x12** для X-12-ARIMA, достаточно удобный, правда, не так давно разработчики достаточно сильно поменяли синтаксис, из-за чего автору пришлось перепи-

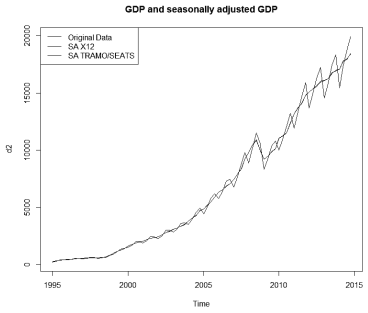


Рис. 2.1. Динамика ВВП в исходном виде и скорректированного X-12 и TRAMO/SEATS

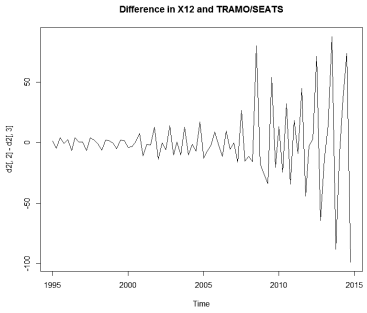


Рис. 2.2. Расхождение в рядах, скорректированных X-12 и TRAMO/SEATS

сывать много кода, и, если они сделали это один раз, ничего не мешает им сделать это ещё раз[8].

Для нового метода X-13 также есть отдельный пакет **seasonal**, тоже новый. Он позиционируется как простой и удобный в использовании.

Важно отметить, что все пакеты для методов семейства X-11 требуют установки исходных двоичных файлов (binaries), которые можно бесплатно и без регистрации скачать на сайте Bureau of Census, www.census.gov/srd/www/x12a/. После установки, в R надо будет просто указать путь к этим файлам.

Пакет для работы с TRAMO/SEATS в R (и не только) есть на сайте Банка Испании, `http://www.bde.es/bde/en/secciones/servicios/Profesionales/Programas_estadi/Interfaces.html`. Представлено как описание пакета, так и простейшие примеры работы с ним.

4.3 Почему надо быть осторожными при использовании сезонной корректировки

Несмотря на все достоинства методов сезонной корректировки с точки зрения простоты использования и минимизации дополнительных усилий исследователя, стоит заметить, что у них есть и ряд серьёзных недостатоков, исследованию которых посвящен большой пласт литературы. Рассмотрим некоторые из них.

Самая, пожалуй, неожиданная черта процедур сезонной корректировки — это их способность создавать фиктивную сезонность.[9] Смысл в том, что если пропустить через алгоритм сезонной корректировки ряд без сезонности, но с

[8] В таких ситуациях можно ставить пакеты R на определённую дату с помощью пакетов **packrat** или **checkpoint**.

[9] Бессонов В. А., Петроневич А. В. Сезонная корректировка как источник ложных сигналов1 //Экономический журнал ВШЭ. – 2013. – Т. 17. – №. 4. – С. 554-584

выбросом, на «очищенных» от сезонности рядах в окрестности этого выброса методы (это свойственно как семейству X-11, так и TRAMO/SEATS) покажут ложные волны, очень похожие на неучтенную сезонность. В контексте анализа реальных данных эта проблема становится ещё серьезнее — на скорректированных рядах, содержащих, к примеру, кризисный период, в окрестности кризиса (самое главное — перед кризисом) появятся какие-то волны, которые можно трактовать в том числе и как предвестники кризиса. После чего следует вывод, что кризис приближался, это было заметно, но никто не обратил на него внимания, тогда как на самом деле эти «предвестники кризиса» — не более чем артефакты процедур сезонной корректировки.

Ещё одна достаточно серьёзная проблема процедур удаления сезонности — это создаваемые ими смещения в тестах на единичные корни. Было аналитически доказано,[10] что в случае отсутствия единичного корня в данных сезонная корректировка создает смещение в тестах в сторону наличия единичного корня. То есть, стационарный ряд после сезонной корректировки можно принять за нестационарный! И чем дальше ряд от нестационарного - тем больше смещение. Та же самая проблема получается и при тестировании на коинтеграцию при помощи теста Энгла-Грейнджера. При этом для нестационарных рядов никаких дополнительных эффектов не возникает.

Большое количество проблем также возникает в связи с шоками, выбросами и сдвигами в данных. К примеру, было показано,[11] что сезонная корректировка при наличии в данных шока длиной в несколько точек (чего-то наподобие кризиса) уменьшает величину шока, но увеличивает его продолжительность и сдвигает точку окончания спада (а это уже действительно серьёзная с точки зрения, к примеру, проведения государственной политики, проблема). X-12-ARIMA часто оказывается неустойчивым к выбросам и структурным сдвигам,[12] на коротких рядах может создавать смещения в начале ряда, которые растут с ростом волатильности сезонной компоненты.[13]

И этим список проблем, порождаемых сезонной корректировкой, не исчерпывается. Разумеется, это не значит, что сезонную корректировку нельзя использовать ни в коем случае и ни под каким предлогом, но равно не стоит и применять её бездумно ко всем рядам и безоглядно доверять получившимся результатам. Во многих ситуациях может оказаться разумнее вместо сезонной корректировки просто построить модель, в явном виде учитывающую сезон-

[10] Ghysels E., Perron P. The effect of seasonal adjustment filters on tests for a unit root //Journal of Econometrics. – 1993. – Т. 55. – №. 1. – С. 57-98.

[11] Matas-Mir A., Osborn D. R., Lombardi M. J. The effect of seasonal adjustment on the properties of business cycle regimes //Journal of Applied Econometrics. – 2008. – Т. 23. – №. 2. – С. 257-278.

[12] Bruce A. G., Jurke S. R. Non-Gaussian seasonal adjustment: X-12-ARIMA versus robust structural models //Journal of Forecasting. – 1996. – Т. 15. – №. 4. – С. 305-328

[13] Mir A. M., Rondonotti V. The performance of X-12 in the seasonal adjustment of short time series //Seasonal Adjustment. – 2003. – С. 149.

ность, либо же внимательнее изучить и сравнить свойства разных процедур сезонной корректировки в контексте тех проблем и особенностей, которые свойственны изучаемым рядам.

5 Небольшая инструкция по установке

5.1 Установка на windows

1. Скачайте готовый архив `x13ashtmlall.zip` с http://www.census.gov/srd/www/x13as/x13down_pc.html. Если буржуйский сайт блокируется доблестными силами добра (например, у меня блокируется втихаря, как будто такого сайта нет, без предупреждения «Ресурс заблокирован...»), то можно воспользоваться сервисами веб-прокси. Например, https://hide.me/en/proxy, https://www.hidemyass.com/proxy и т.д.
2. Распакуйте архив и запомните адрес папки с файлом `x13ashtml.exe`.
3. Установите в R пакет **seasonal**.

5.2 Установка на macos/linux

Подробная инструкция есть на странице https://github.com/christophsax/seasonal/wiki/. Борьба с блокировкой буржуйского сайта также может потребоваться :)

5.3 Простой пример

В начале скрипта R мы пишем

```
Sys.setenv(X13_PATH ="путь к папке с файлами")
library("seasonal")
```

Если есть желание проверить, что всё ок, то это можно сделать командой `checkX13()`. И простой-простой пример автоматической борьбы с сезонностью в ряду `y`:

```
y_sa_model <- seas(y)
y_sa <- final(y_sa_model)
```

Список `y_sa_model` будет содержать модель, которая использовалась для корректировки сезонности, а ряд `y_sa` будет очищенным от сезонности.

На сайте https://github.com/christophsax/seasonal/wiki/ есть куча примеров! Прочитайте также замечательную виньетку к пакету **seasonal**, http://cran.r-project.org/web/packages/seasonal/vignettes/seas.pdf, там есть и инструкция по установке, и графический интерфейс для подбора модели корректировки сезонности!

6 Выводы

Сезонная корректировка — один из основных инструментов работы с сезонностью в данных. Она позволяет быстро и удобно устранить сезонность и продолжать работать с данными. Два основных метода устранения сезонности — методы семейства X-11 и TRAMO/SEATS — применяют несколько отличные подходы (которые описаны выше и которые полезно хотя бы в общих чертах понимать, чтобы не работать с процедурами как с «чёрными ящиками», как это, увы, чаще всего происходит). Оба метода представлены во всех основных статистических пакетах, как коммерческих, так и открытых. Оба же метода иногда приводят к появлению дополнительных проблем в данных, о которых следует помнить и которые следует учитывать при использовании сезонно скорректированных данных.

Задачи

Фольклор и коллектив кафедры*

21 июня 2015 г.

Аннотация

Прикольные задачи по теории вероятностей, теории игр, динамической оптимизации, а заодно и здравому смыслу!

Ключевые слова: взрыв мозга, только хардкор, слабо.

Задача 1. Злобный Дракон поймал принцесс Настю и Сашу и посадил в разные башни. Перед каждой из принцесс Злобный Дракон подбрасывает один раз правильную монетку. А дальше даёт каждой из них шанс угадать, как выпала монетка у её подруги. Если хотя бы одна из принцесс угадает, то Злобный Дракон отпустит принцесс на волю. Если обе принцессы ошибутся, то они навсегда останутся у него в заточении.

Подобная практика у Злобного Дракона исследователями была отмечена уже давно, поэтому принцессы имели достаточно времени договориться на случай вероятного похищения.

Как следует поступать принцессам при подобных похищениях?

Задача 2. Удав-Пустынник любит программировать на python и есть французские багеты[1]. Длина французского багета равна 1 метру. За один заглот Удав-Пустынник заглатывает кусок случайной длины равномерно распределенной на отрезке $[0;1]$. Для того, чтобы съесть весь багет удаву потребуется случайное количество N заглотов.

1. Найдите $\mathbb{E}(N)$ и $\mathrm{Var}(N)$
2. Как поменяются ответы, если багет имеет длину 2 метра?

Задача 3. Ефросинья подкидывают правильную монетку неограниченное количество раз.

---

*

[1] «Удав из которого говорит кролик, — это не тот удав, который нам нужен». [Искандер, 1982]

1. Сколько в среднем нужно сделать бросков до появления последовательности OPOP?
2. А до появления последовательности POPP?
3. Какова вероятность того, что OPOP появится раньше POPP?

Задача 4. Эконометресса Барбара оценивает с помощью МНК модель $y_t = \beta x_t + \varepsilon_t$. Ошибки ε_t независимы, имеют нулевое среднее и постоянную дисперсию, регрессоры известны и равны $x_t = 1/2^t$.

1. Получит ли Барбара состоятельную оценку для β?
2. Эконометресса Виолетта оценивает с помощью взвешенного МНК ту же модель, однако ошибочно предполагает, что имеет место гетероскедастичность вида $\mathrm{Var}(\varepsilon_t) = \sigma^2 x_t^2$. Получит ли Виолетта состоятельную оценку для β?

Задача 5. Трое заядлых игроков в покер сидят в чате. Предложите процедуру раздачи карт, при которой каждый игрок знает свои карты и не знает карт соперника. Игроки абсолютно рациональны и обладают безграничными вычислительными возможностями, поэтому использование кодов с открытым ключом (типа RSA) недопустимо. В чате можно посылать сообщения, адресованные как всем сразу, так и конкретному лицу.

Список литературы

Искандер Ф. Кролики и удавы. — Ann Arbor, 1982.

Стол заказов

Борис Демешев*

21 июня 2015 г.

Для того, чтобы выходили новые номера «Эпсилон», нужны новые статьи. И побольше, побольше... Присоединяйтесь к нам, пишите как мы, пишите лучше нас!

1 Смысловая часть

Статья должна быть интересной и не содержать отклонений от здравомыслия! Она может относиться к эконометрике, теории игр, динамической оптимизации, анализу данных, теории вероятностей или математической статистике, а также всему, что понадобится впредь. Хорошо бы, чтобы в статье были красивые картинки! Если у вас есть свои идеи — предлагайте, обсудим. Если вы хотите, чтобы мы написали про что-то статью, спросите! Если есть желание написать статью для «Эпсилон», а идей нет, то мы предлагаем готовые сюжеты для методических статей-проектов и готовы оказать поддержку в написании:

- Векторно-матричное дифференциирование с примерами.
  Как взять производную по вектору? По матрице? Среди примеров могут быть: оценка ковариационной матрицы с помощью ML, МНК, LDA, PCA, канонические корреляции.
- Линейный дискриминантный анализ по чесноку.
  Что это за зверь? Какова его связь с логит-регрессией? Примеры задач. Честный вывод формул.
- Как мы порвали всех во втором туре универсиады по метрике.
  Условие задачи и решение с кодом R.
- Частная корреляция.
  Отличия от обычной. Два подхода к расчёту: через проецирование и уравнение регрессии. Эквивалентность через теорему Фриша-Вау.

*НИУ ВШЭ, Москва.

- Теорема Фриша-Вау с примерами.
  Геометрическое доказательство Фриша-Вау. Примеры: регрессия на константу, частная корреляция
- Как строить карты России в R?
  Больше примеров карт, хороших и разных!
- Пакет для обработки данных RLMS в R.
  Написать функцию для автоматического объединения данных по индивидам из разных волн на основании родственных связей. И описать с примерами
- Пакет со списком источников экономических данных по России и вспомогательными функциями для работы с ними.
  Смутная идея, но вдруг кто вдохновится?
- Ещё раз подчеркнём: это идеи сюжетов для тех, кто хочет написать статью, но не знает за что взяться. Мы только «за» другие смелые идеи!

В методических статьях главное — чтобы и ежу было понятно! Разумно включать простые примеры и упражнения для читателя. Если дело касается работы с реальными данными, то очень желателен код R.

2 Техническая часть

Мы принимаем статьи, написанные с помощью LaTeX, языка разметки маркдаун и грамотного программирования (literate programming) с использованием R (форматы `Rmd`, `Rnw`). Если у вас есть особо ценная статья написанная в другом формате, пишите нам, обсудим. С исходными текстами статей первого номера можно ознакомиться, глянув репозиторий https://github.com/bdemeshev/epsilon/tree/master/e_001.

Пожалуйста, используйте LaTeX правильно. Сейчас мы ведём переговоры с Биллом Гейтсом, Карлосом Слимом и Уорреном Баффетом о спонсировании работы штата корректоров, а пока переговоры продолжаются мы просим будущих авторов:

1. Для выносных формул не используйте окружение `$$ $$`. Используйте `\[ \]`.
2. Не забывайте неразрывный пробел, `~`, после короткого предлога. Например, `в~Москве`.
3. Не забывайте длинное тире. Не забывайте, что оно пишется с помощью трёх коротких чёрточек, `---`.
4. Языки программирования или статистический софт записывайте с помощью `\proglang{}`. Например, `\proglang{R}`.
5. Пакеты для R или другой среды записывайте с помощью `\pkg{}`. Например, `\pkg{dplyr}`.

6. Для отдельно стоящих команд R или другого пакета используйте `\code||`. Например, `\code|model <- lm(data=cars, dist~speed)|`.
7. Любите и уважайте букву «ё».