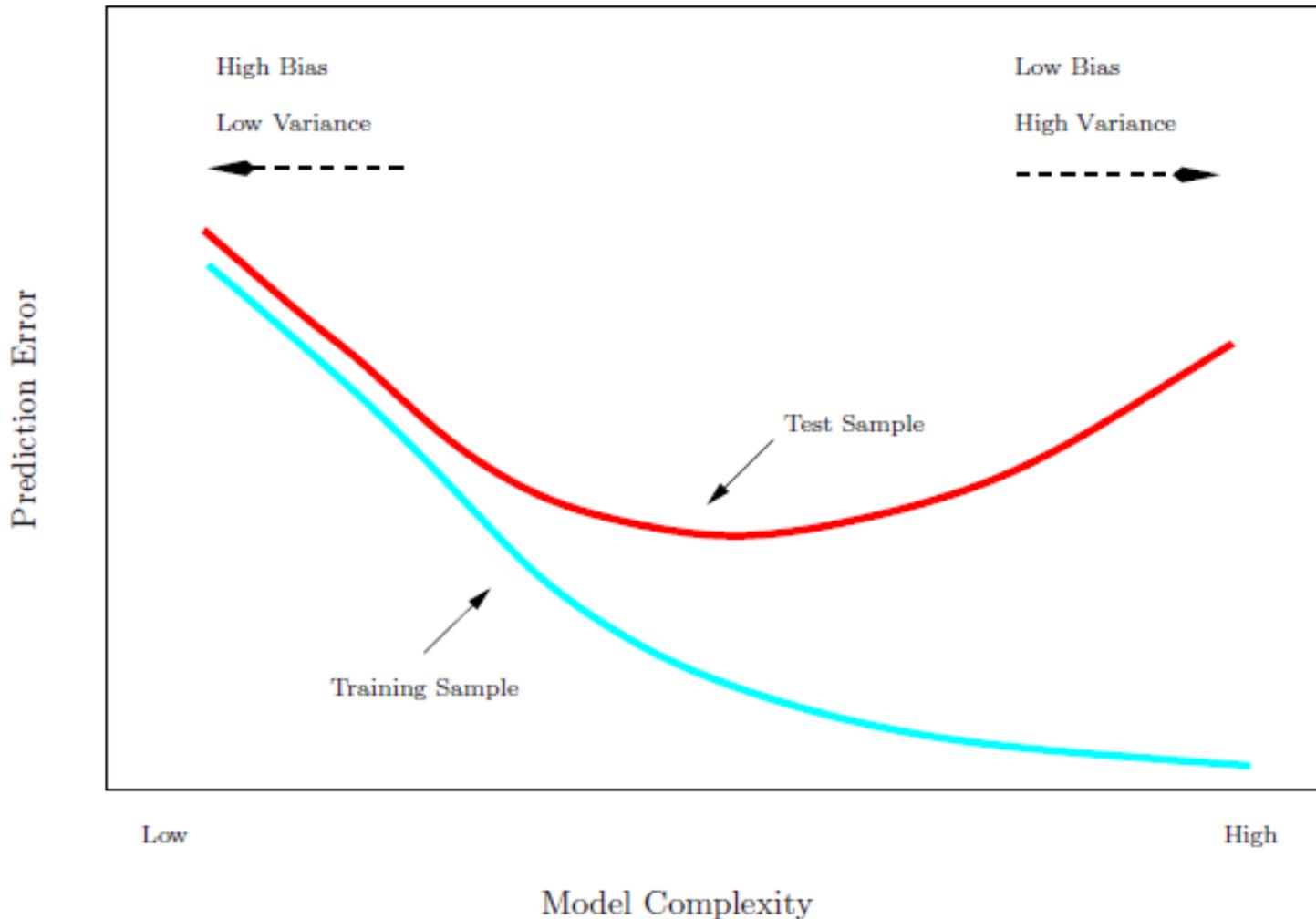




Лекция 6: оценка и выбор моделей

Качество на обучающем и тестовом наборе



Валидация, кросс-валидация и бутстреппинг

- Эти методы позволяют:
 - оценить ошибки прогнозирования тестового набора
 - стандартное отклонение и смещения оценок параметров модели
 - выбрать лучшую модель
- Различия между *ошибкой тестирования* и *ошибкой обучения*:
 - Ошибка тестирования - это усредненная ошибка, которая возникает в результате применения метода статистического обучения для прогнозирования отклика на новом наблюдении, которое не было задействовано в процессе обучения.
 - Ошибка обучения вычисляется после применения метода статистического обучения к наблюдениям, используемым в обучении.

Применение валидационного набора

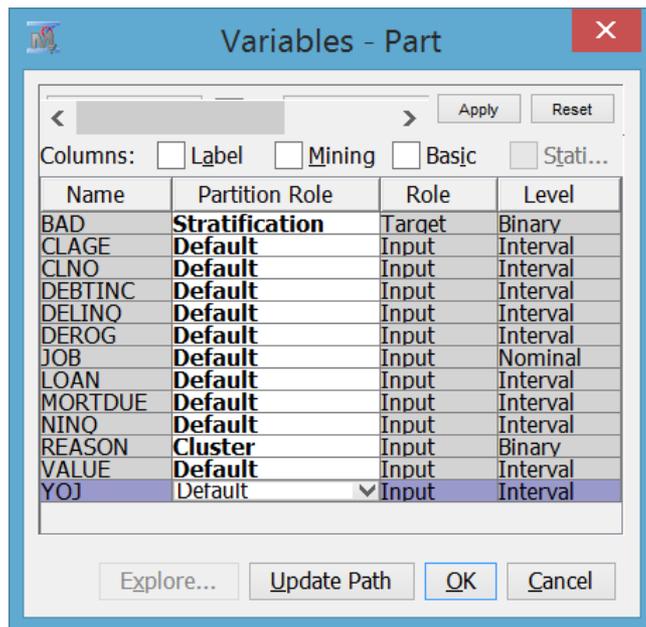
- Разделим случайным образом имеющийся набор образцов на две части: *обучающую и валидационную выборки.*



- Построим модель на обучающем наборе и используем ее для прогнозирования откликов наблюдений в валидационном наборе.
- Полученная ошибка на валидационном множестве дает оценку тестовой ошибки. Ошибка, как правило, оценивается с использованием MSE в случае количественного отклика и ошибки неправильной классификации в случае категориального отклика.

Узел Data Partition в EM

- Разбивает входной Raw набор на Train/Test/Validate наборы в заданной пропорции
- Поддерживает стратификацию по категориальному отклику и/или по категориальным или кластерным (номер кластера) переменным, сохраняя таким образом распределение



Использование валидационного набора данных

Training Data

	<i>inputs</i>			<i>target</i>

Validation Data

	<i>inputs</i>			<i>target</i>

Основные методы генерации валидационного набора как и в sampling:

- Случайная выборка
- Стратифицированная выборка (сохраняем распределение выбранных переменных)
- Кластерная выборка (сохраняем пропорции кластеров)

Оценка моделей

Training Data

	<i>inputs</i>		<i>target</i>

Validation Data

	<i>inputs</i>		<i>target</i>



**Оценка качества моделей
на валидационном наборе**

Сложность модели *Валидационная оценка*

Выбор модели

Training Data

	<i>inputs</i>			<i>target</i>

Validation Data

	<i>inputs</i>			<i>target</i>



**Самая простая модель среди
самых лучших на
валидационном наборе**

Сложность модели *Валидационная оценка*

Оценки ошибки прогнозирования

- Лучшее решение: большой валидационный набор. Часто не доступен или тяжело выделить.
- Некоторые методы дают *математическую корректировку* частоты ошибок обучения с целью оценки частоты ошибок тестирования. К ним относятся *Ср статистика*, *AIC* и *BIC*:
 - $n \log(SSE/n) + \text{Штраф}$ (у каждого свой, см. таблицу):

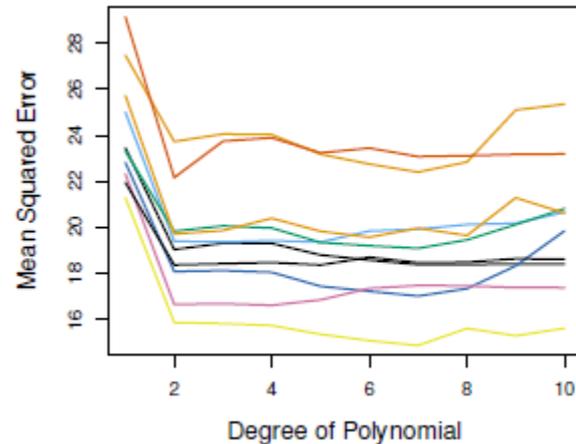
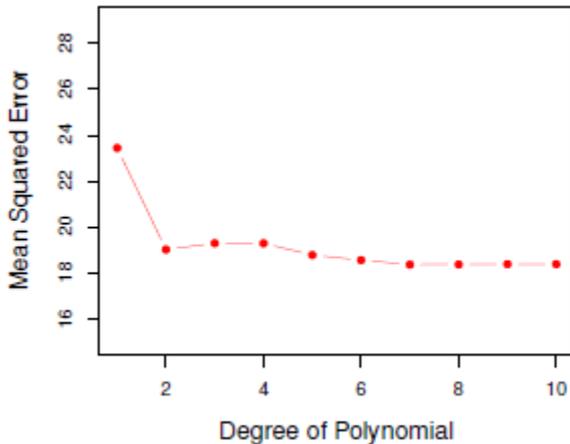
Information Criteria	Penalty Component
AIC	$2p + n + 2$
AICC	$\frac{n(n+p)}{n-p-2}$
BIC	$2(p+2)q - 2q^2$
SBC	$p \log(n)$

- p – число параметров, n – число наблюдений,
- SSE – сумма квадратов ошибок,
- σ^2 – оценка дисперсии для полной модели

$$q = \frac{n\hat{\sigma}^2}{SSE}$$

Пример

- Хотим сравнить регрессионные модели с разными степенями полинома
- Разделим случайным образом 392 наблюдения на две группы: обучающий набор, содержащий 196 объектов и валидационный набор, содержащий оставшиеся 196 объектов.



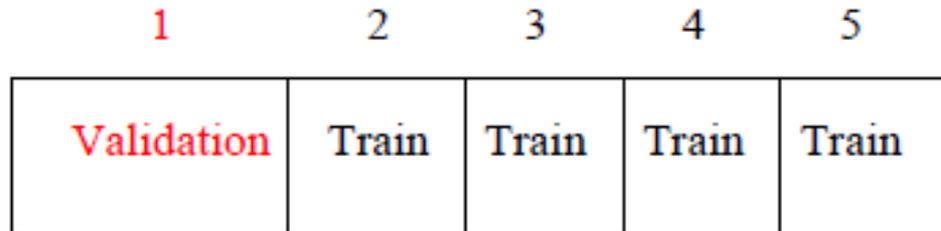
Слева показано одиночное разбиение, справа - множественное

Недостатки подхода применения валидационного набора

- Если плохое разбиение:
 - Валидационная оценка ошибки тестирования может сильно варьироваться в зависимости от того, какие именно наблюдения включены в обучающий набор, а какие в валидационный.
- Не вся информация используется при обучении:
 - При валидационный подходе только подмножество наблюдений (те, которые включены в обучающий набора, а не в валидационный) используются для построения модели.
- Чрезмерный оптимизм:
 - Ошибка на валидационном наборе может иметь тенденцию *переоценивать* ошибку тестирования

Кросс-валидация

- Широко используемый подход для оценки ошибки тестирования.
- Оценки могут быть использованы для:
 - выбора оптимальной модели,
 - оценки тестовой ошибки результирующей выбранной модели.
- Идея - разделить данные на K частей равного размера. Мы удаляем часть k , строим модель на оставшихся частях, а затем получаем прогнозы для удаленной k -ой части.

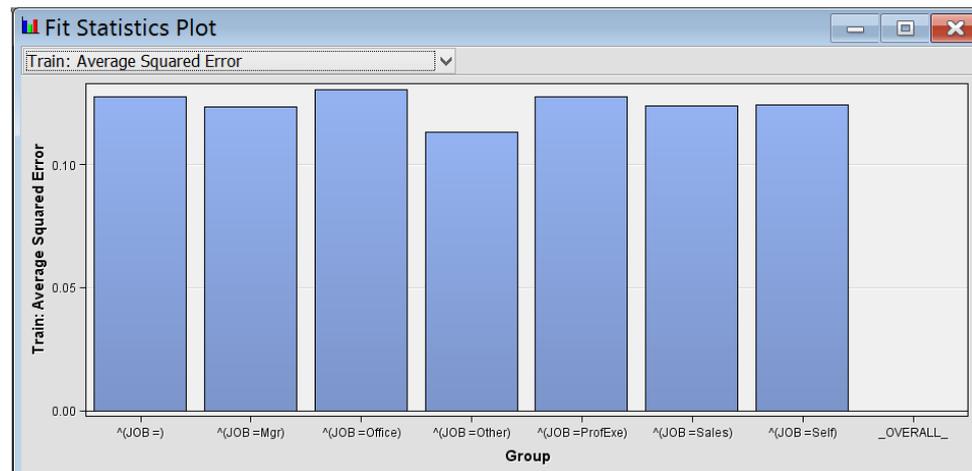
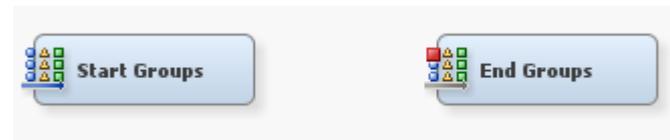


- Это делается в свою очередь для каждой части $k = 1, 2, \dots, K$, а затем результаты объединяются.

Кросс валидация в EM

- Многие алгоритмы имеют встроенные процедуры кросс-валидации метопараметров (в основном сложность модели)
- Узлы Start Group – End Group позволяют делать кроссвалиадционную оценку ошибки для любых моделей

Property	Value
General	
Node ID	Grp2
Imported Data	...
Exported Data	...
Notes	...
Train	
Variables	...
Rerun	No
General	
Mode	Cross-Validation
Target Group	No
Index Count	10
Minimum Group	10
Bagging	
Type	Percentage
Observations	.
Percentage	10.0
Random Seed	12345
Status	
Create Time	23.03.17 1:50
Run ID	

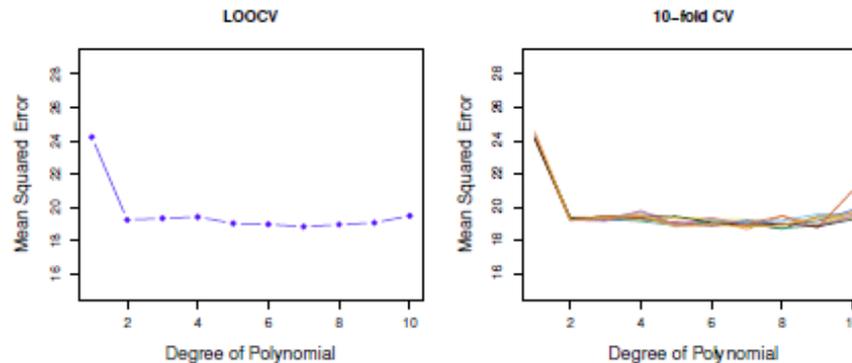


Кросс-валидация для оценки ошибки

- Обозначим K частей как C_1, C_2, \dots, C_K , где C_k - это индексы наблюдений в части k . Есть n_k наблюдения в части k : если N кратно K , то $n_k = n/K$.

- Вычислим
$$CV_{(K)} = \sum_{k=1}^K \frac{n_k}{n} \text{MSE}_k$$

где $\text{MSE}_k = \sum_{i \in C_k} (y_i - \hat{y}_i)^2 / n_k$ и \hat{y}_i - подгонка для наблюдения i , полученная на данных с удаленной частью k .

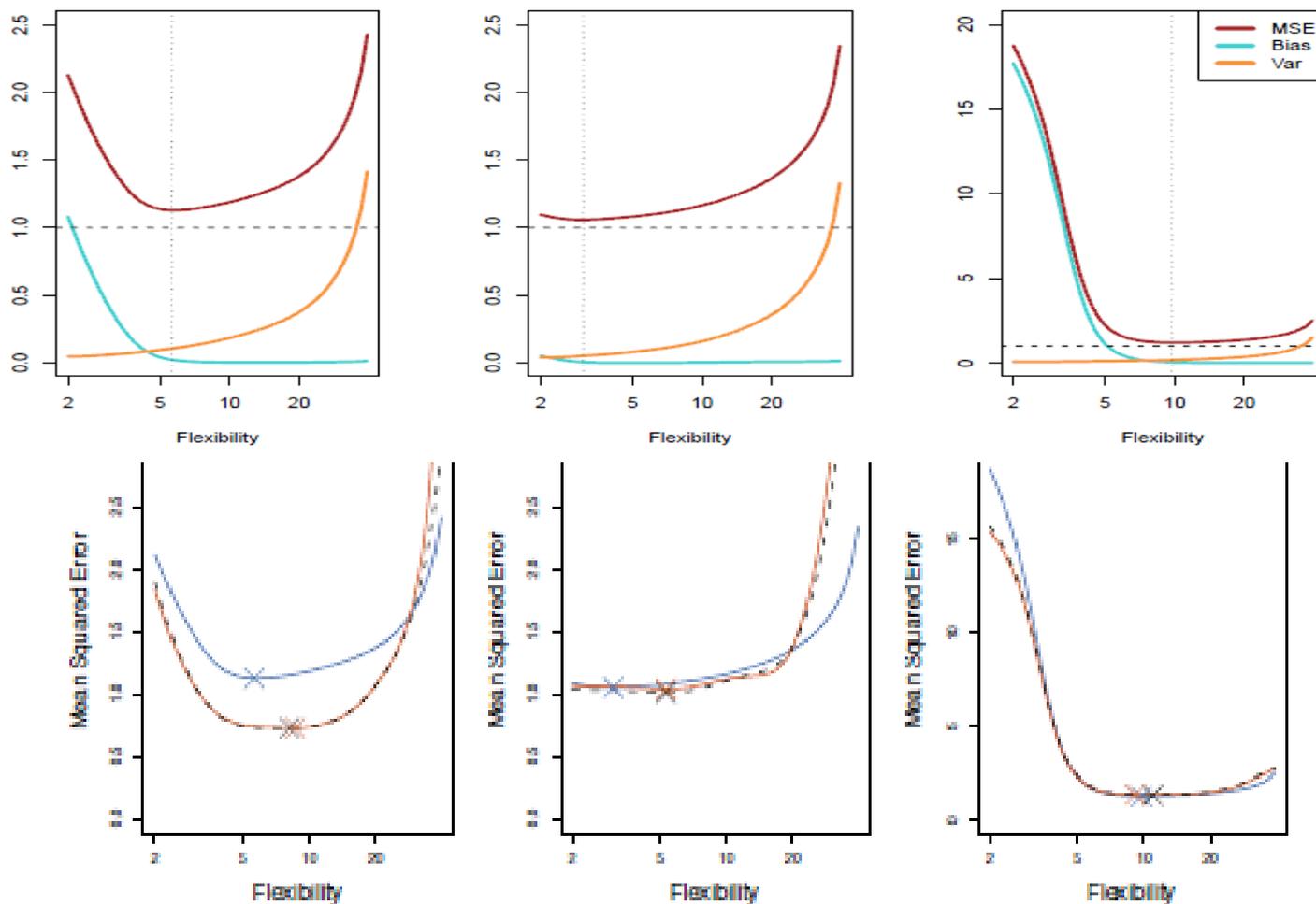


- При $K = n$ имеем n папок или кросс-валидацию с попеременным исключением одной из частей (*leave-one out cross-validation*, LOOCV).

Кросс-валидация для оценки метопараметров и выбора модели

- Зачастую кросс-валидацию используют не для оценки ошибки, а для выбора метопараметров
 - Запускают кросс-валидацию для разных значений метопараметров
 - Рассчитывают кросс-валидационные ошибки для каждого варианта
 - Выбирают лучшее значение метопараметра по кросс-валидационной ошибке
 - Перестраивают модель на всей выборке с этим значением метопараметра

Кросс-валидация для оценки метопараметров модели



Бутсреппинг

- *Бутстреппинг* представляет собой гибкий и мощный статистический инструмент, который может быть использован для количественной оценки неопределенности, связанной с данным методом статистического обучения.
- Например, он может позволить произвести оценку стандартной ошибки коэффициента или доверительного интервала для этого коэффициента.
- Использование термина бутстреппинг происходит от фразы, чтобы *to pull oneself up by one's bootstraps*, - цитата из книги «Удивительные приключения барона Мюнхгаузена»

Барон упал на дно глубокого озера. Когда казалось, что все было потеряно, он решил вытащить себя своими собственными силами.

Простой пример из другой области 😊

- Предположим, что мы хотим вложить определенную сумму денег в два финансовых актива, которые дают доход X и Y соответственно, где X и Y являются случайными величинами.
- Мы будем инвестировать часть α наших денег в X , и оставшиеся $1-\alpha$ в Y .
- Мы хотим, выбрать такое α , чтобы минимизировать общий риск, или дисперсию, наших инвестиций. Другими словами, мы хотим минимизировать

$$\text{Var}(\alpha X + (1 - \alpha)Y).$$

- Можно показать, что $\alpha = \frac{\sigma_Y^2 - \sigma_{XY}}{\sigma_X^2 + \sigma_Y^2 - 2\sigma_{XY}}$, минимизирует риск

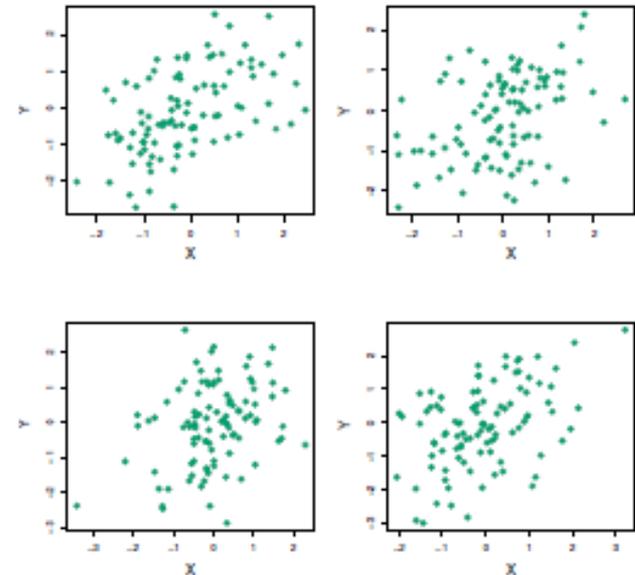
$$\sigma_X^2 = \text{Var}(X), \sigma_Y^2 = \text{Var}(Y) \quad \sigma_{XY} = \text{Cov}(X, Y).$$

Продолжение примера

- Но значения σ_X^2 , σ_Y^2 и σ_{XY} неизвестны.
- Можем вычислить оценки этих величин $\hat{\sigma}_X^2$, $\hat{\sigma}_Y^2$, и $\hat{\sigma}_{XY}$, используя набор данных, который содержит X и Y.
- Затем мы можем оценить значение того, что сводит к минимуму дисперсию наших инвестиций с использованием:

$$\hat{\alpha} = \frac{\hat{\sigma}_Y^2 - \hat{\sigma}_{XY}}{\hat{\sigma}_X^2 + \hat{\sigma}_Y^2 - 2\hat{\sigma}_{XY}}$$

Каждый рисунок отображает 100 смоделированных данных для инвестиций X и Y. Слева направо и сверху вниз полученные в результате оценки для α равны 0,576, 0,532, 0,657 и 0,651.



Продолжение примера

- Для оценки стандартного отклонения $\hat{\alpha}$ мы повторили 1000 раз процесс имитационного моделирования 100 парных наблюдений X и Y .
- Таким образом, мы получили 1000 оценок для α , которые обозначим

$$\hat{\alpha}_1, \hat{\alpha}_2, \dots, \hat{\alpha}_{1000}.$$

- Для моделирования параметры были заданы (допустим мы их знали) как:

$$\sigma_X^2 = 1, \sigma_Y^2 = 1.25, \quad \sigma_{XY} = 0.5$$

- таким образом, мы знаем, что истинное значение α равно 0,6

Продолжение примера

- Среднее значение по всем 1000 оценкам для α

$$\bar{\hat{\alpha}} = \frac{1}{1000} \sum_{r=1}^{1000} \hat{\alpha}_r = 0.5996,$$

очень близко к $\alpha = 0.6$, а стандартное отклонение оценок $\hat{\alpha}$

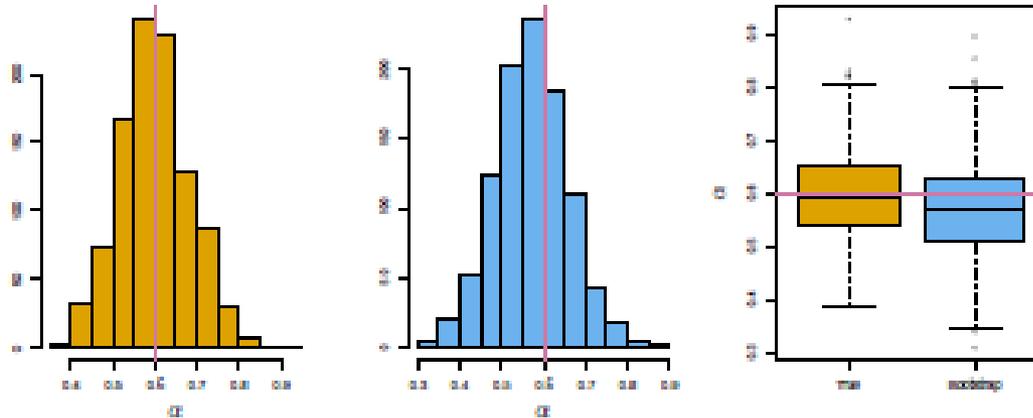
$$\sqrt{\frac{1}{1000 - 1} \sum_{r=1}^{1000} (\hat{\alpha}_r - \bar{\hat{\alpha}})^2} = 0.083.$$

- Это дает нам очень хорошее представление о точности

$$SE(\bar{\hat{\alpha}}) \approx 0.083.$$

- Так что, грубо говоря, для случайной выборки можно было бы ожидать, что $\hat{\alpha}$ в среднем отличается от α примерно на 0.08.

Результат

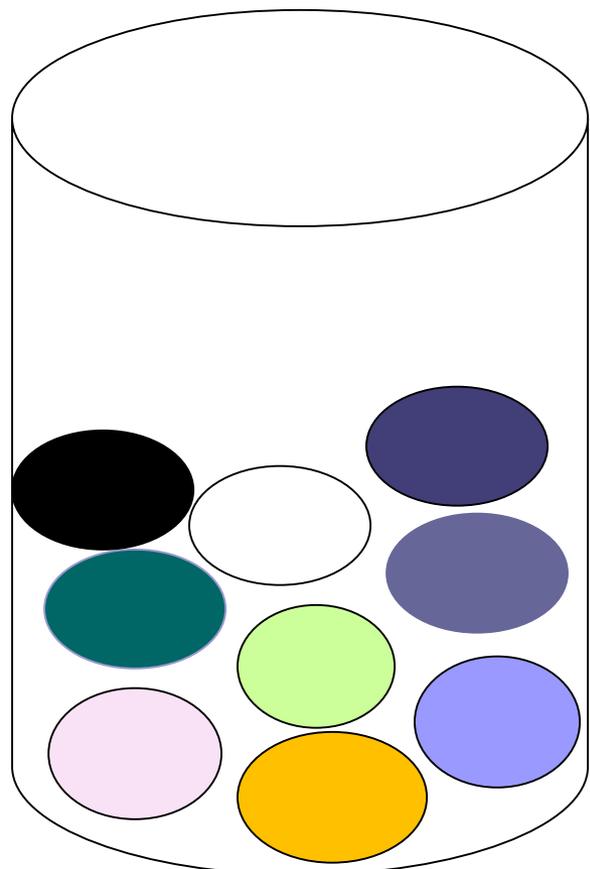


- Слева: Гистограмма оценок, полученных путем генерации 1000 смоделированных наборов данных
- В центре: Гистограмма оценок α , полученных на основе 1000 бутстреппинговых образцов из одного набора данных.
- Справа: Оценки α , изображенные в левой и центральной части, показаны как boxplot. В каждой части розовая линия показывает истинное значение α .

Вернемся к реальности

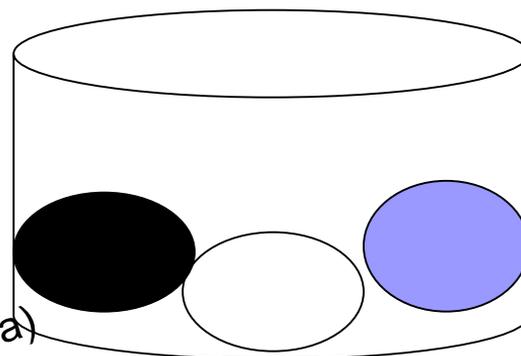
- Описанная выше процедура не может быть применена, потому что для реальных данных мы не можем генерировать новые образцы из исходной выборки (не знаем истинные характеристик распределения)
- Тем не менее, подход бутстреппинга позволяет имитировать процесс получения новых случайных наборов данных, так что мы можем оценить дисперсию нашей оценки, не создавая дополнительных образцов.
- Вместо того, чтобы постоянно получать независимые наборы данных, мы получаем различные наборы путем многократной выборки наблюдений из исходного набора *с замещением* (или *с возвращением*).
- Каждый из этих "наборов данных" создается путем выборки *с замещением* и имеет *такой же размер* как наш исходный набор данных. В результате некоторые наблюдения могут появляться более одного раза в наборе данных бутстреппинга, а некоторые нет вообще.

Случайная выборка с возвратом и без

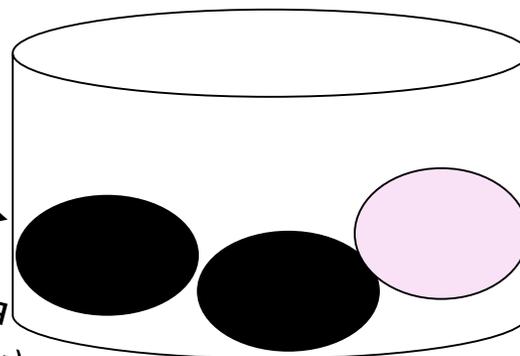


«Сырые» данные

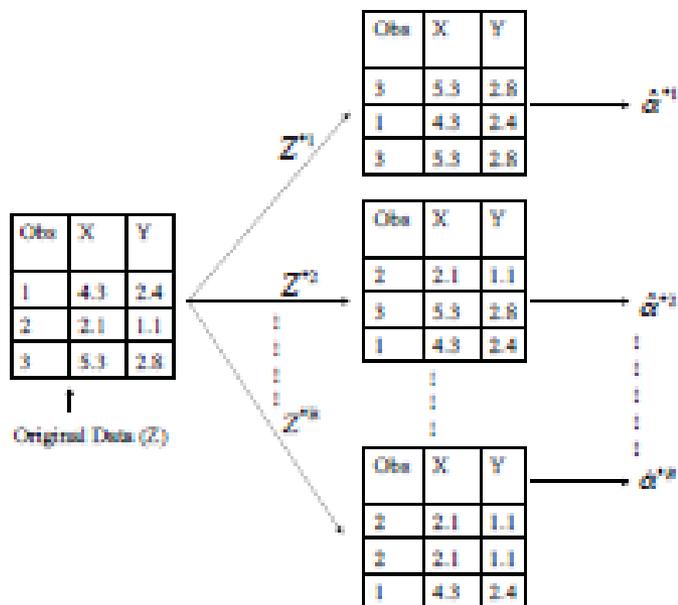
SRSWOR
(простая случайная
выборка без возврата)



SRSWR
(простая случайная
выборка с возвратом)



Демонстрационный пример с тремя наблюдениями



- Графическая иллюстрация бутстреппингового подхода на маленькой выборке, содержащей из $N = 3$ наблюдений.
- Каждый бутстреппинговый набор данных содержит n наблюдений, отобранных с заменой из исходного набора.
- Каждый такой набор данных начальной используется для получения оценки α

Бутстрепинг

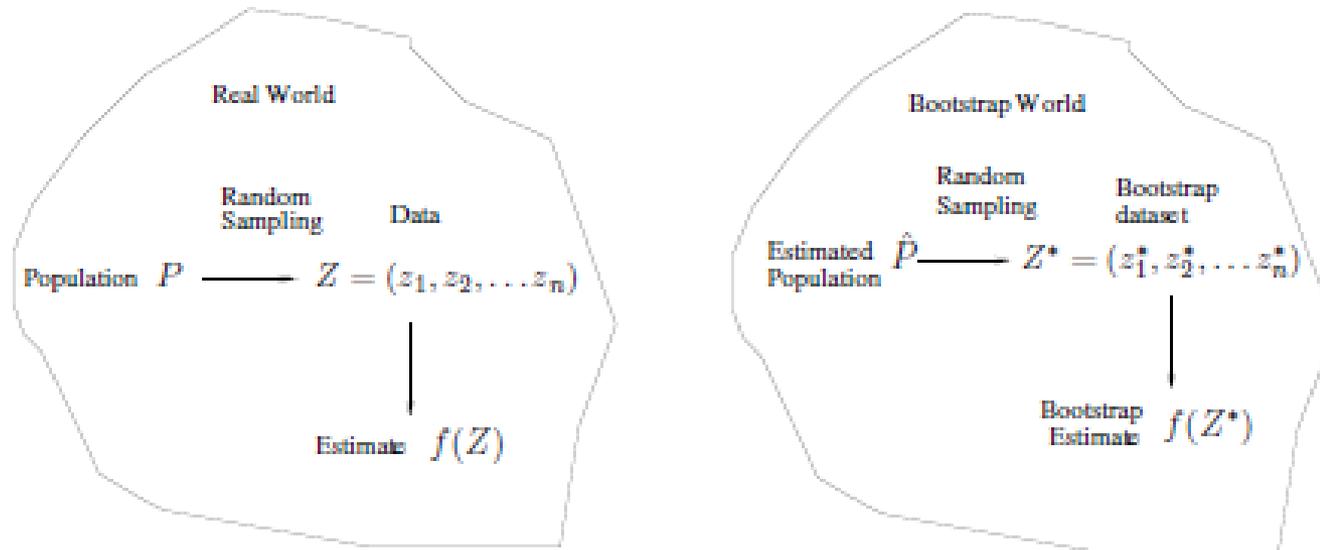
- Обозначая первый набор данных бутстреппинга как Z^{*1} , мы используем Z^{*1} , чтобы выполнить новую оценку для α , которую обозначим $\hat{\alpha}^{*1}$
- Эта процедура повторяется B раз для некоторого большого значения B (например, 100 или 1000), чтобы получить B различных наборов данных бутстреппинга $Z^{*1}, Z^{*2}, \dots, Z^{*B}$, и B соответствующих оценок α : $\hat{\alpha}^{*1}, \hat{\alpha}^{*2}, \dots, \hat{\alpha}^{*B}$.

- Оценим стандартную ошибку этих оценок бутстреппинга, используя формулу:

$$SE_B(\hat{\alpha}) = \sqrt{\frac{1}{B-1} \sum_{r=1}^B (\hat{\alpha}^{*r} - \bar{\hat{\alpha}}^*)^2}.$$

- Она служит в качестве оценки стандартной ошибки, полученной на исходном наборе данных.

Общая схема бутстреппинга



- В более сложных ситуациях, определение подходящего способа для получения выборок бутстреппинга может потребовать значительных усилий.
- Например, если данные представляют собой временные ряды, мы не можем просто выбирать наблюдения с замещением

Другие применения бутстрепинга

- В основном используется для получения оценки стандартных ошибок.
- Также обеспечивает приближенные доверительные интервалы для параметра генеральной совокупности.
- Вышеуказанный интервал называется доверительный интервал *перцентиля бутстреппинга*. Это самый простой способ (среди многих подходов) для получения доверительного интервала бутстрепинга.
- Используется в *багинг ансамблях* моделей (BAG = Bootstrap Aggregation)

Как бутстрепинг оценивает ошибку прогнозирования

- При кросс-валидации каждая из K папок валидации отличается от других $K - 1$ папок, используемых для обучения: *перекрытия нет*. Это очень важно для получения хороших результатов.
- Для оценки ошибки прогнозирования с помощью бутстреппинга мы могли бы подумать об использовании каждого набора данных бутстреппинга в нашей обучающей выборке и исходного набора данных как валидационного набора (или наоборот).
 - Но каждая выборка бутстреппинга имеет значительное перекрытие с исходными данными. Около двух третей исходных точек данных появляются в каждой выборке бутстреппинга.
 - Это приведет бутстреппинг к существенному недооцениванию истинной ошибки прогнозирования
- Удаление перекрытия (*out of bag*)- можно частично решить эту проблему, используя для оценки только те наблюдения, которые не появились (случайно) в текущей выборке бутстреппинга.

Бутстреппинг в EM

- Некоторые алгоритмы имеют встроенные процедуры бутстреппинга (например, случайный лес)
- Узлы Start Group – End Group позволяют делать Bagging модель для любых моделей

Property	Value
General	
Node ID	Grp
Imported Data	...
Exported Data	...
Notes	...
Train	
Variables	...
Rerun	No
General	
Mode	Bagging
Target Group	No
Index Count	10
Minimum Group Size	10
Bagging	
Type	Percentage
Observations	.
Percentage	10.0
Random Seed	12345
Status	

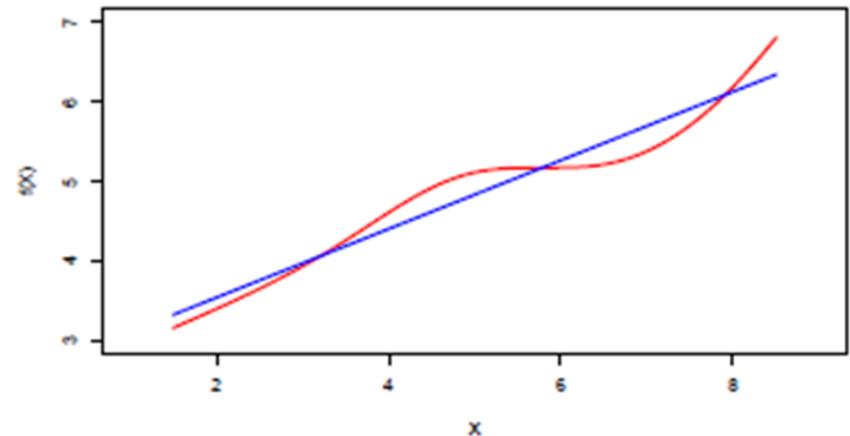


Выбор модели и регуляризация на примере линейных моделей регрессии

- Отклик линейной модели

$$Y = \beta_0 + \beta_1 X_1 + \dots + \beta_p X_p + \epsilon.$$

- Истинные зависимости редко бывают линейными
- Но: *интерпретируемость* и часто хорошее *качество прогнозирования*.
- Рассмотрим способы, при которых простая линейная модель может быть улучшена путем замены МНК некоторыми альтернативными методами подгонки.



Оценка точности расчета коэффициентов

- Стандартная ошибка оценки отражает, какие изменения происходят при повторной выборке

$$SE(\hat{\beta}_1)^2 = \frac{\sigma^2}{\sum_{i=1}^n (x_i - \bar{x})^2}, \quad SE(\hat{\beta}_0)^2 = \sigma^2 \left[\frac{1}{n} + \frac{\bar{x}^2}{\sum_{i=1}^n (x_i - \bar{x})^2} \right]$$

где $\sigma^2 = \text{Var}(\epsilon)$

- Эти стандартные ошибки могут быть использованы для вычисления *доверительных интервалов*.
- 95%-ый доверительный интервал определяется как диапазон значений, таких что с вероятностью 95% диапазон будет содержать истинное неизвестное значение параметра. Он имеет вид

$$\hat{\beta}_1 \pm 2 \cdot SE(\hat{\beta}_1).$$

Проверка гипотез

- Стандартные ошибки также могут использоваться для *проверки гипотез* о коэффициентах.
- Наиболее распространенный вариант проверки гипотез заключается в проверке *нулевой гипотезы*

H_0 : Нет никакой связи между X и Y

против *альтернативной гипотезы*

H_A : Между X и Y существует некоторая взаимосвязь.

- Математически это соответствует проверке

$$H_0 : \beta_1 = 0 \quad \text{vs} \quad H_A : \beta_1 \neq 0,$$

Проверка гипотез - продолжение

- Чтобы проверить нулевую гипотезу (о равенстве нулю коэффициента), мы вычисляем *t*-статистику

$$t = \frac{\hat{\beta}_1 - 0}{SE(\hat{\beta}_1)},$$

- Она будет иметь *t*-распределение с $n-2$ степенями свободы, предполагая $\beta_1 = 0$.
- Можно вычислить вероятность наблюдения любого значения, большего или равного $|t|$. это *p*-value.

	Coefficient	Std. Error	t-statistic	p-value
Intercept	7.0325	0.4578	15.36	< 0.0001
TV	0.0475	0.0027	17.67	< 0.0001

Оценка общей точности модели

- Мы вычисляем *стандартную ошибку невязок*

$$\text{RSE} = \sqrt{\frac{1}{n-2} \text{RSS}} = \sqrt{\frac{1}{n-2} \sum_{i=1}^n (y_i - \hat{y}_i)^2},$$

где сумма квадратов невязок: $\text{RSS} = \sum_{i=1}^n (y_i - \hat{y}_i)^2$.

- *R-квадрат* или доля объясненной дисперсии

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

где $\text{TSS} = \sum_{i=1}^n (y_i - \bar{y})^2$ - общая сумма квадратов.

- Можно показать, что в простой
- линейной регрессии $R^2 = r^2$,
- где r - корреляция между X и Y :

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

$$F = \frac{(\text{TSS} - \text{RSS})/p}{\text{RSS}/(n-p-1)} \sim F_{p, n-p-1}$$

Quantity	Value
Residual Standard Error	1.69
R^2	0.897
F-statistic	570

Интерпретация коэффициентов регрессии

- Идеальный сценарий: предикторы некоррелированы:
 - Каждый коэффициент можно оценить и тестировать отдельно.
 - Интерпретации такие как *«единичного изменение X_j связано с β_j -ым изменением в значении Y , тогда как все остальные переменные остаются фиксированными»*.
- Корреляции между переменными вызывают проблемы:
 - Дисперсия всех коэффициентов имеет тенденцию к увеличению, иногда резкому
 - Интерпретации становятся непредсказуемыми - когда X_j меняется, все остальное тоже меняется.
- *«По сути, все модели ошибочны, но некоторые из них полезны»* (George Box)

Качественные (категориальные) признаки

Пример: исследовать различия в балансе кредитных карт между мужчинами и женщинами, не учитывая другие переменные.

Создается новая переменная

$$x_i = \begin{cases} 1 & \text{if } i\text{th person is female} \\ 0 & \text{if } i\text{th person is male} \end{cases}$$

Итоговая модель имеет вид:

$$y_i = \beta_0 + \beta_1 x_i + \epsilon_i = \begin{cases} \beta_0 + \beta_1 + \epsilon_i & \text{if } i\text{th person is female} \\ \beta_0 + \epsilon_i & \text{if } i\text{th person is male.} \end{cases}$$

Интерпретация:

	Coefficient	Std. Error	t-statistic	p-value
Intercept	509.80	33.13	15.389	< 0.0001
gender [Female]	19.73	46.05	0.429	0.6690

Качественные признаки с более чем двумя возможными значениями

- Для признаков с несколькими возможными значениями создаются дополнительные фиктивные переменные. Например, для переменной ethnicity :

$$x_{i1} = \begin{cases} 1 & \text{if } i\text{th person is Asian} \\ 0 & \text{if } i\text{th person is not Asian,} \end{cases}$$

а вторая:

$$x_{i2} = \begin{cases} 1 & \text{if } i\text{th person is Caucasian} \\ 0 & \text{if } i\text{th person is not Caucasian.} \end{cases}$$

- Тогда обе эти переменные могут быть использованы в формуле регрессии и модель будет иметь вид

$$y_i = \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \epsilon_i = \begin{cases} \beta_0 + \beta_1 + \epsilon_i & \text{if } i\text{th person is Asian} \\ \beta_0 + \beta_2 + \epsilon_i & \text{if } i\text{th person is Caucasian} \\ \beta_0 + \epsilon_i & \text{if } i\text{th person is AA.} \end{cases}$$

Качественные признаки с более чем двумя ВОЗМОЖНЫМИ значениями

- Число фиктивных переменных будет на единицу меньше, чем количество возможных различных значений, есть специальное *базовое значение*.

<i>Level</i>	D_A	D_B	D_C	D_D	D_E	D_F	D_G	D_H	D_I
A	1	0	0	0	0	0	0	0	0
B	0	1	0	0	0	0	0	0	0
C	0	0	1	0	0	0	0	0	0
D	0	0	0	1	0	0	0	0	0
E	0	0	0	0	1	0	0	0	0
F	0	0	0	0	0	1	0	0	0
G	0	0	0	0	0	0	1	0	0
H	0	0	0	0	0	0	0	1	0
I	0	0	0	0	0	0	0	0	1

	Coefficient	Std. Error	t-statistic	p-value
Intercept	531.00	46.32	11.464	< 0.0001
ethnicity[Asian]	-18.69	65.02	-0.287	0.7740
ethnicity[Caucasian]	-12.50	56.68	-0.221	0.8260

Расширения линейной модели

Удаление предположения аддитивности: *взаимосвязи* и *нелинейность*

Взаимосвязи:

- Раньше мы предполагали, что влияние на предикторов на отклик независимо, например:

$$\widehat{\text{sales}} = \beta_0 + \beta_1 \times \text{TV} + \beta_2 \times \text{radio} + \beta_3 \times \text{newspaper}$$

- Модель с взаимосвязями имеет вид

$$\begin{aligned} \text{sales} &= \beta_0 + \beta_1 \times \text{TV} + \beta_2 \times \text{radio} + \beta_3 \times (\text{radio} \times \text{TV}) + \epsilon \\ &= \beta_0 + (\beta_1 + \beta_3 \times \text{radio}) \times \text{TV} + \beta_2 \times \text{radio} + \epsilon. \end{aligned}$$

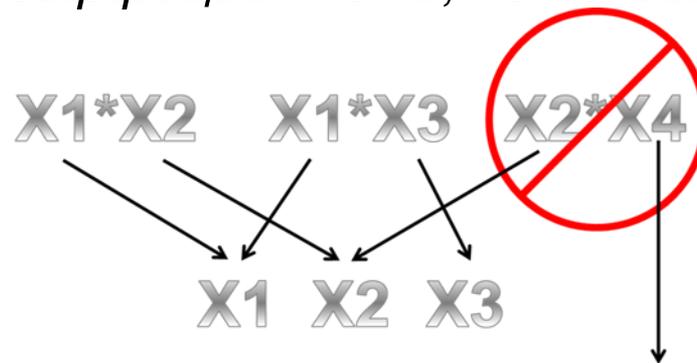
Интерпретация примера

- Результаты в этой таблице показывают, что взаимосвязи важны.
- Величина p -value для члена TV*radio, отражающего взаимосвязь, чрезвычайно мала, что свидетельствует о наличии достоверных доказательств для гипотезы $H_A : \beta_3 \neq 0$.
- R^2 для модели с учетом взаимосвязей составляет 96.8%, по сравнению с только лишь 89.7% для модели, которая прогнозирует значение sales, используя значения TV и radio без учета взаимосвязей между ними.
- Это означает, что $(96.8 - 89.7)/(100 - 89.7) = 69\%$ дисперсии для sales, которая остается после построения аддитивной модели, объясняется эффектом, отражающим взаимосвязь

	Coefficient	Std. Error	t-statistic	p-value
Intercept	6.7502	0.248	27.23	< 0.0001
TV	0.0191	0.002	12.70	< 0.0001
radio	0.0289	0.009	3.24	0.0014
TV×radio	0.0011	0.000	20.73	< 0.0001

Иерархия

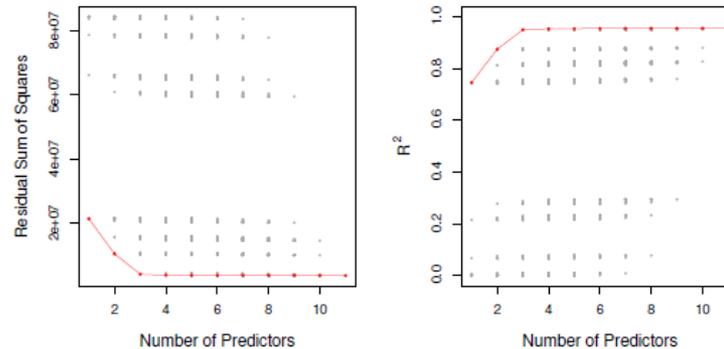
- Иногда может иметь место ситуация, что член, отражающий взаимосвязь, имеет очень маленькое p -value, но связанные с ним базовые признаки (в примере, **TV** и **radio**) не проявляют аналогичные свойства.
- Принцип иерархии:
Если мы включаем взаимосвязь в модель, мы должны также включать базовые признаки, даже если значения p -value, связанные с их коэффициентами, не показывают значимость.



Мотивация – улучшение интерпретируемости

Выбор важных переменных

- Наиболее очевидный подход называется регрессия *всех подмножеств* или регрессия *наилучших подмножеств* (МНК для всех комбинаций и выбор лучшего варианта по некоторому критерию)

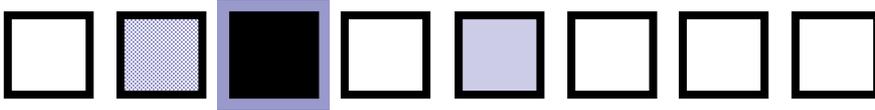


- На практике – не всегда применимо для значительного числа переменных, Поэтому существуют методы:
 1. Пошаговые методы (прямые, обратные, комбинированные)
 2. Методы с регуляризацией (штраф за сложность)
 3. Методы преобразования пространства признаков (регрессия главных компонент и PLS)

Прямой отбор

- Начинаем с *нулевой модели* содержащей только константу.
- Строим p простых линейных регрессий и добавляем к нулевой модели переменную, которая дает наименьшее значение RSS.
- Добавляем к этой модели переменную, которая дает наименьшее значение RSS среди всех моделей с двумя переменными.
- Продолжаем процесс до тех пор, пока не сработает какое-либо правило останова,
- Например, в майнере на каждом шаге рассчитывается p -value гипотезы о том, что улучшение целевой функции по сравнению с нулевой моделью существенно.

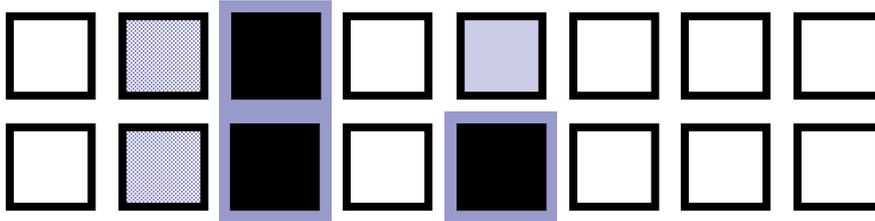
Input p -value



Entry Cutoff



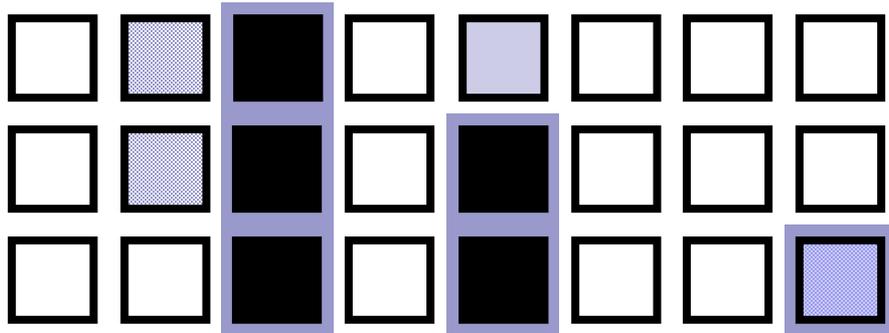
Input p -value



Entry Cutoff



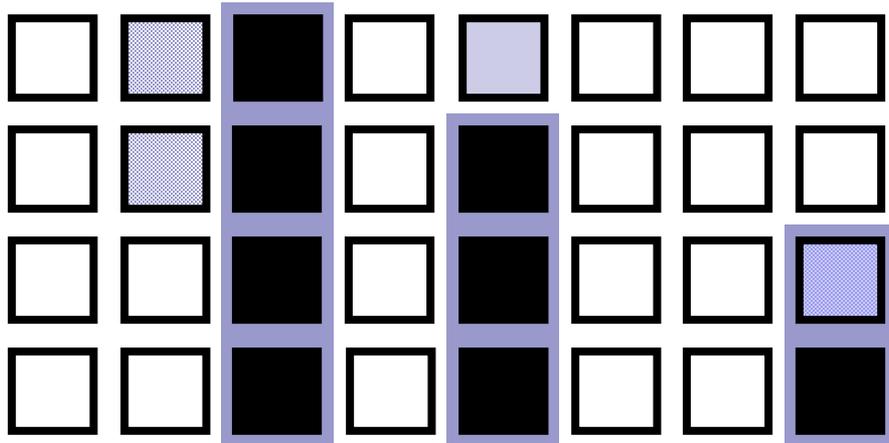
Input p -value



Entry Cutoff



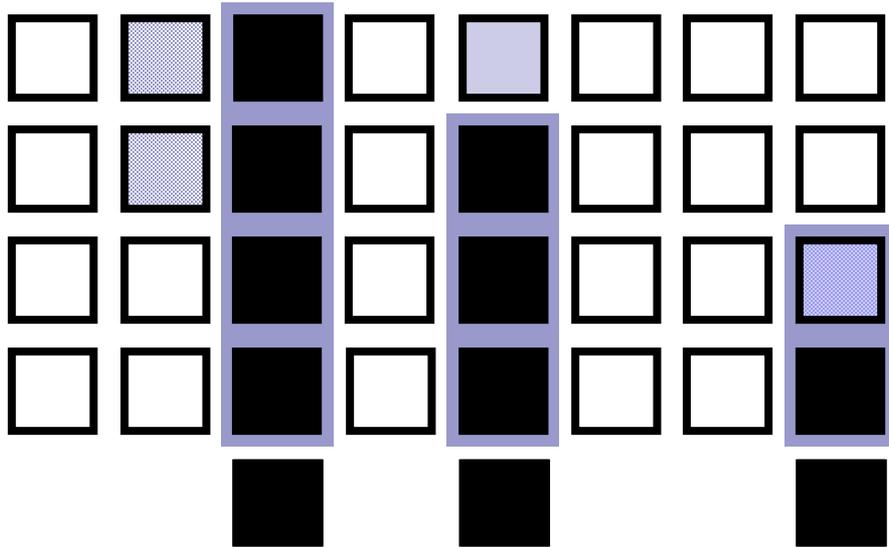
Input p -value



Entry Cutoff



Input p -value



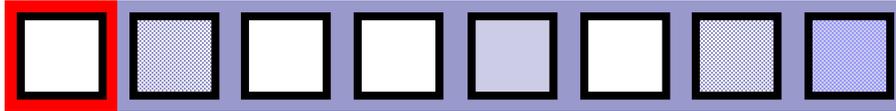
Entry Cutoff



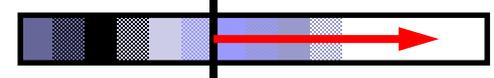
Обратный отбор

- Начинаем с модели, содержащей все переменные.
- Удаляем переменную с наибольшим p -value, то есть переменную, которая является наименее статистически значимой.
- Строим новую модель с $(p-1)$ -ой переменными и удаляем переменную с наибольшим p -value.
- Продолжаем до тех пор, пока не сработает правило останова.
- Например, в майнере на каждом шаге рассчитывается p -value гипотезы о том, что ухудшение целевой функции по сравнению с нулевой моделью существенно.

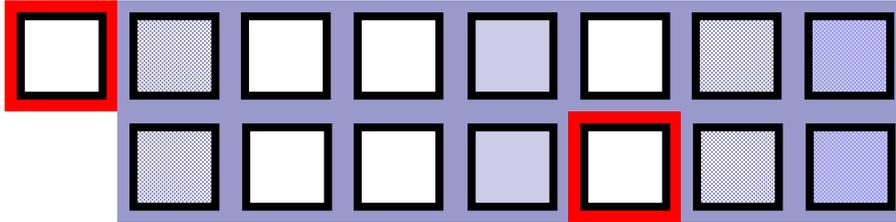
Input p -value



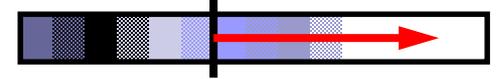
Stay Cutoff



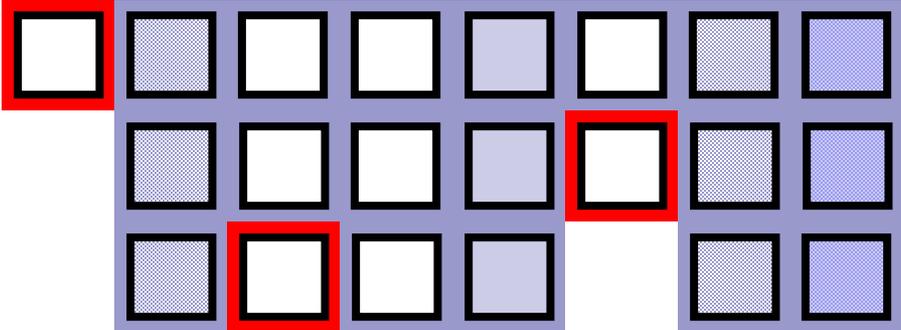
Input p -value



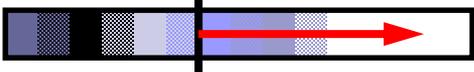
Stay Cutoff



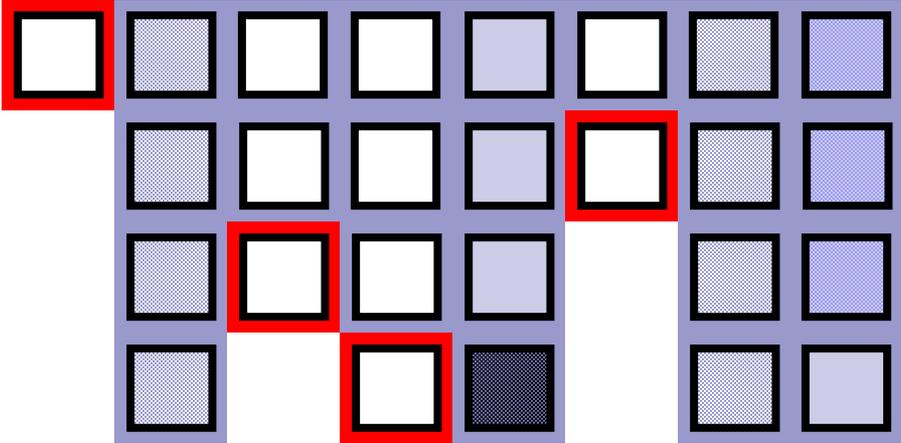
Input p -value



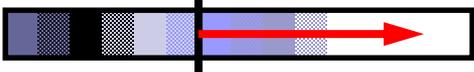
Stay Cutoff



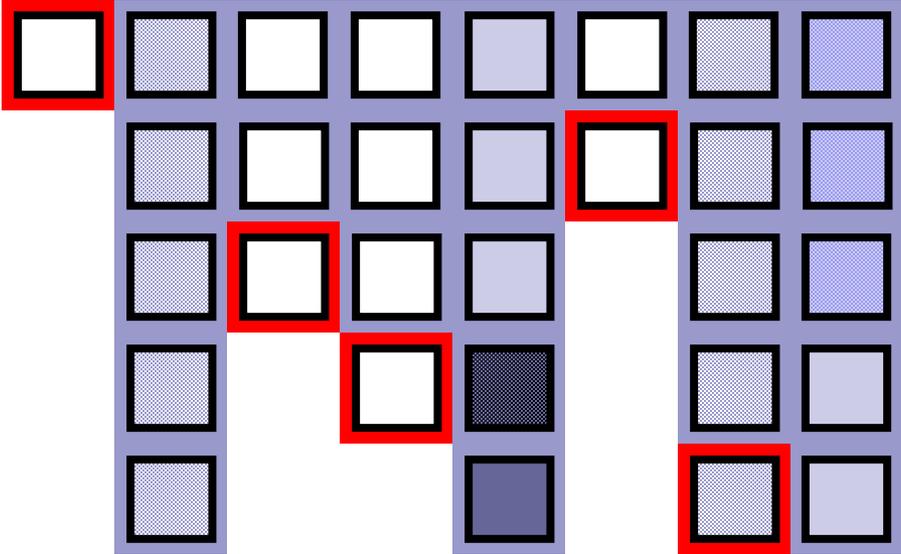
Input p -value



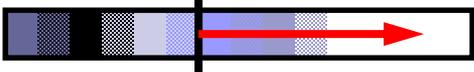
Stay Cutoff



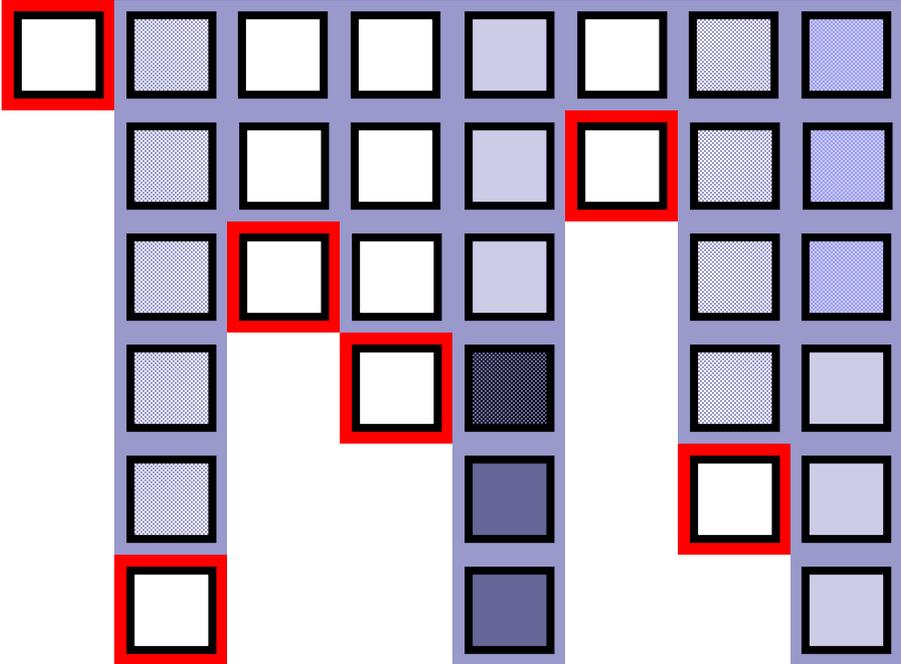
Input p -value



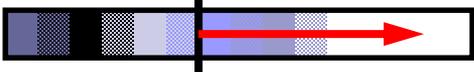
Stay Cutoff



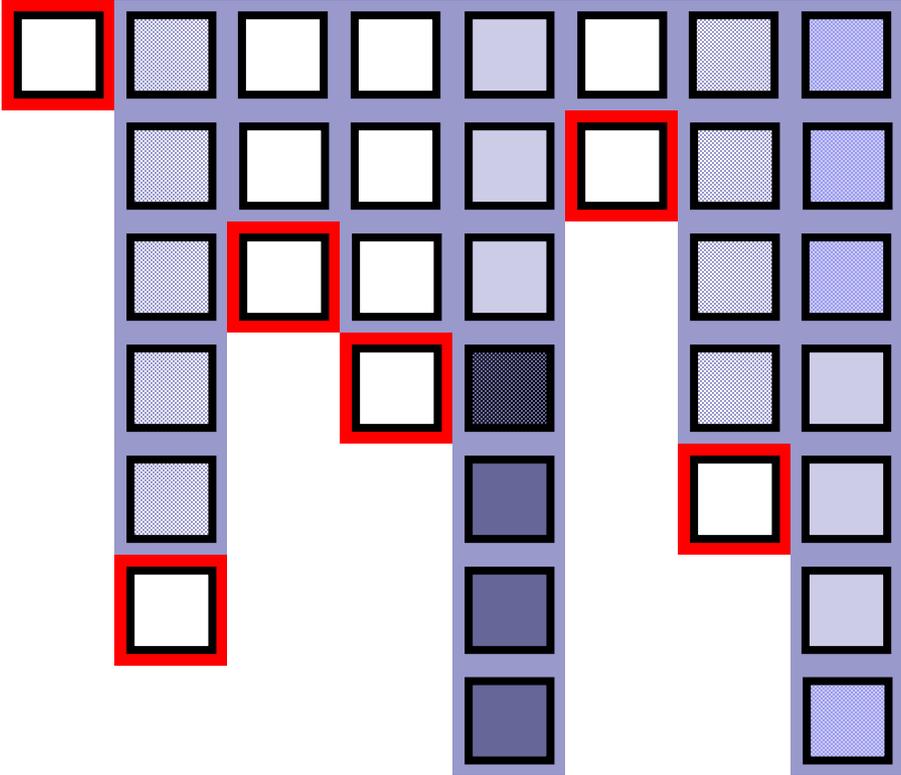
Input p -value



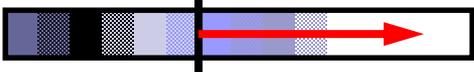
Stay Cutoff



Input p -value



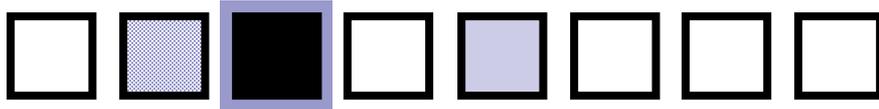
Stay Cutoff



Комбинированный отбор

- Прямой и обратный методы – жадные (как и комбинированный) с не сильно гибким перебором
- Комбинированный метод (более гибкий перебор):
 1. Пытаемся сделать шаг вперед (сначала из нулевой модели)
 2. Затем пытаемся сделать шаг назад
 3. Продолжаем 1 и 2 до тех пор, пока не сработает правило останова и для добавления и для удаления переменных или пока не попали в «цикл» (последовательное добавление и удаление одной и той же переменной)

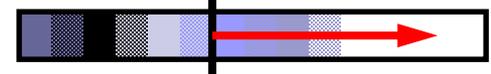
Input p -value



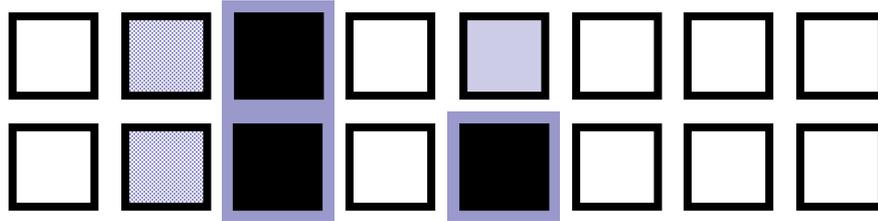
Entry Cutoff



Stay Cutoff



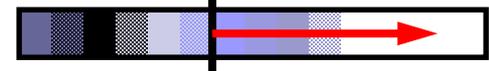
Input p -value

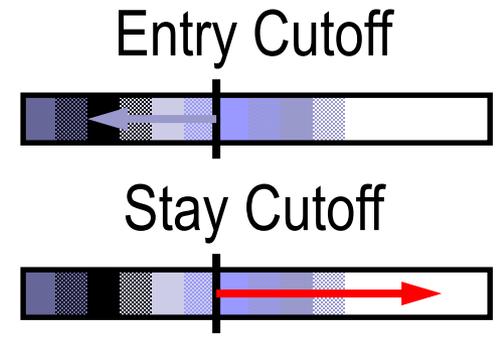
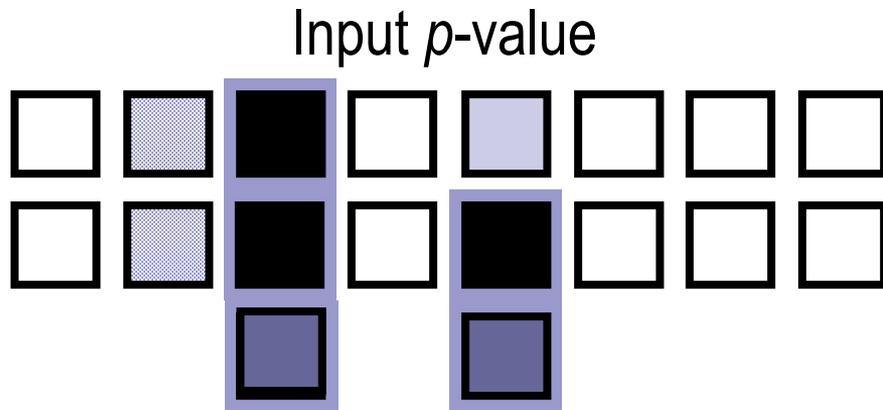


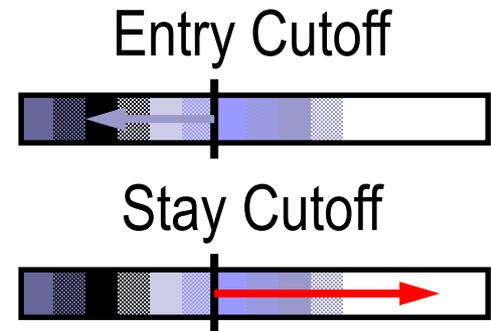
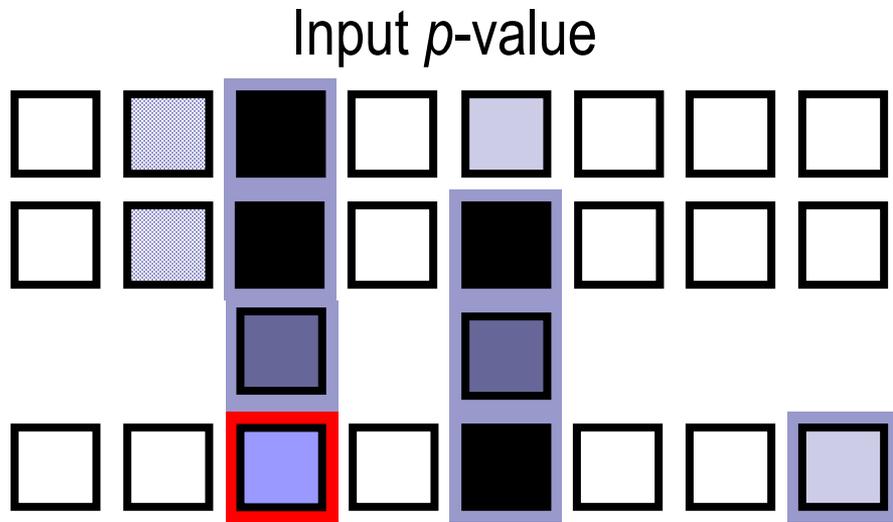
Entry Cutoff

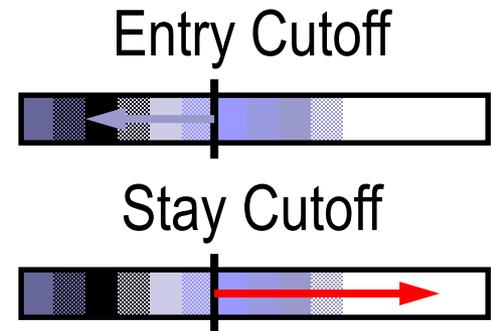
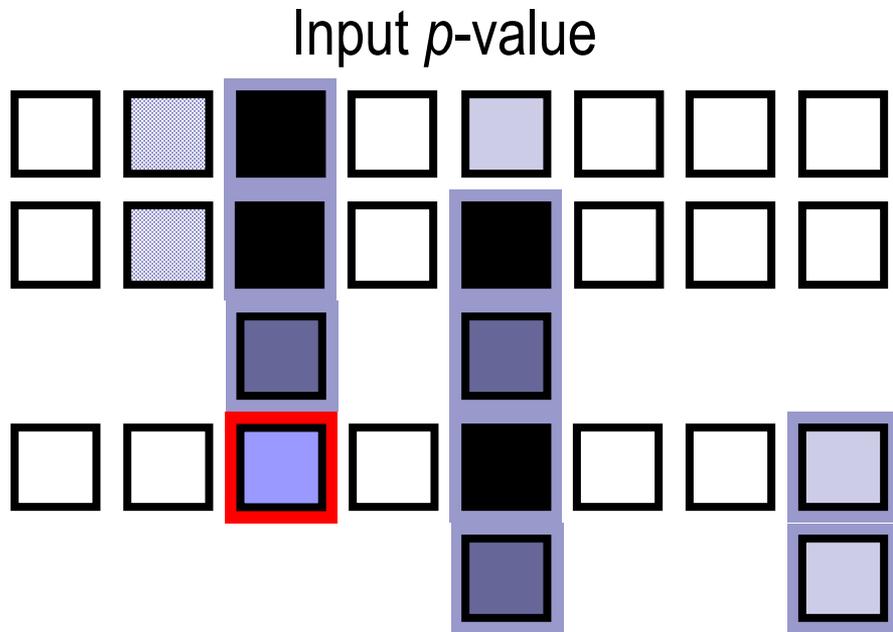


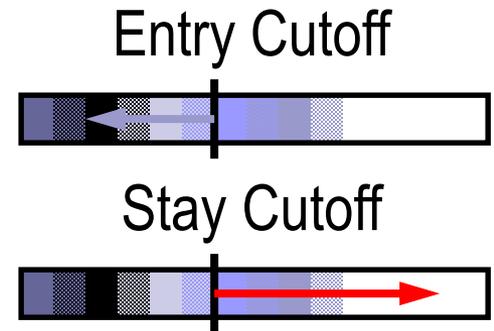
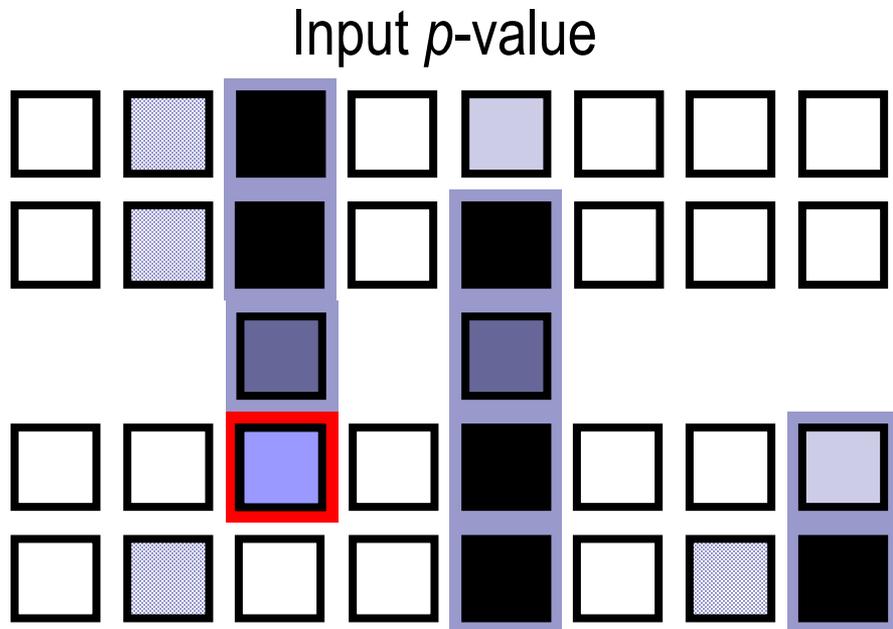
Stay Cutoff

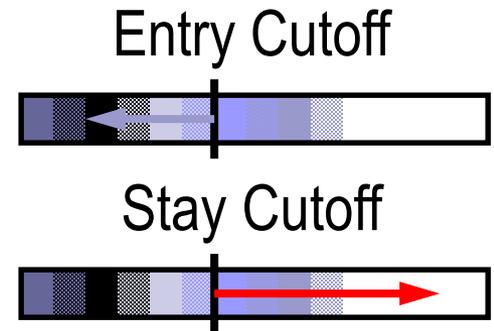
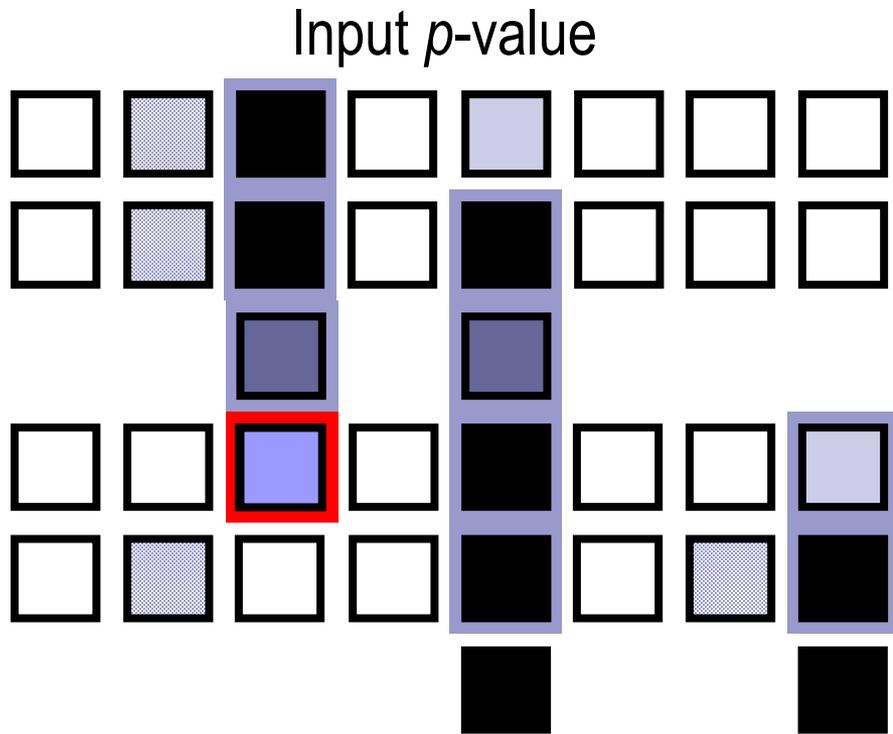










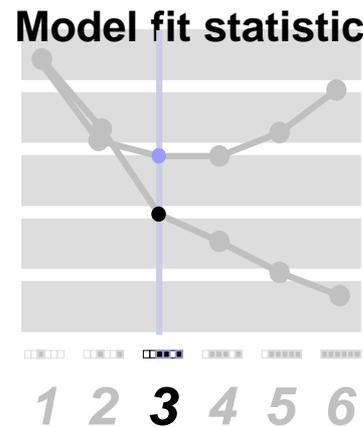
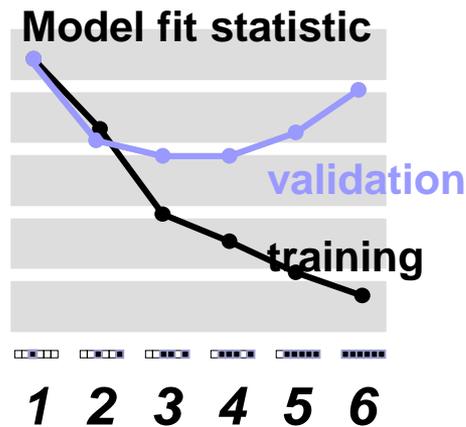


Замечания по пошаговым методам

- Автоматический выбор переменных приводит к:
 - Смещенным оценкам параметров и стандартных ошибок
 - Некорректным оценкам числа степеней свободы
 - p -values «переоценивают» значимость параметров, увеличивая вероятность ошибки первого рода
- Совет – строить модель на одном наборе, оценивать на другом.

Пошаговая регрессия

1. Пусть M_0 – начальная модель, которая не содержит предикторов.
2. Для $k = 0, \dots, p - 1$: все $p - k$ моделей, которые дают улучшение по сравнению с предикторами в M_k с добавлением одного дополнительного предиктора.



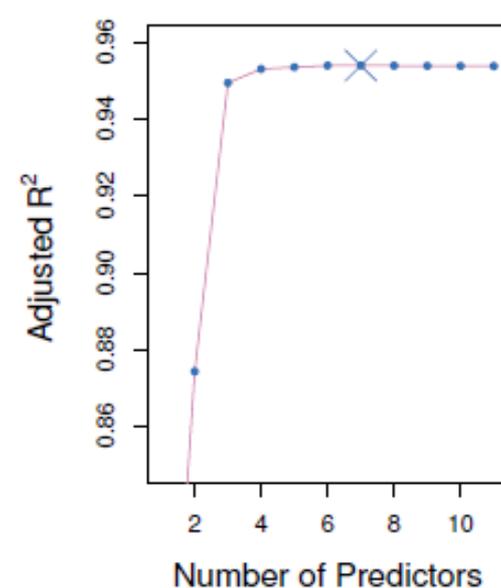
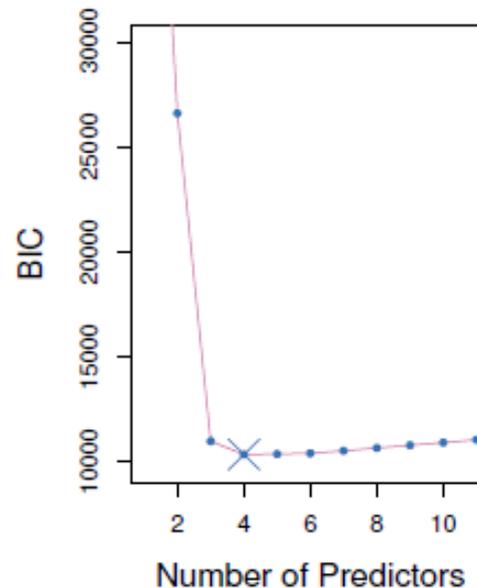
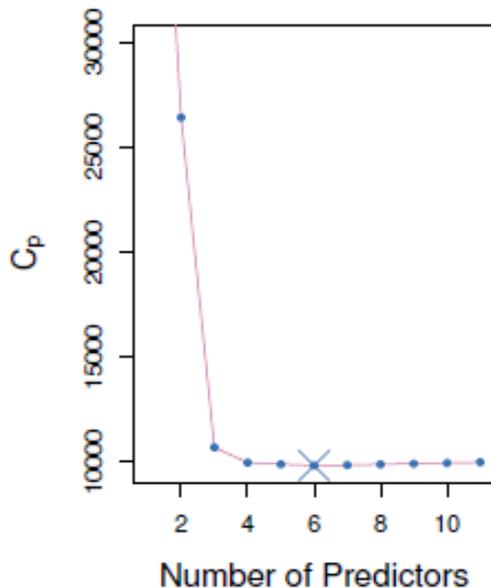
3. Выбираем одну наилучшую модель среди M_0, \dots, M_p , используя валидационную или кросс-валидированную ошибку прогнозирования, C_p (AIC), BIC или скорректированный коэффициент детерминации R^2 .

Выбор оптимальной модели

- Модель, содержащая все предикторы, всегда будет иметь наименьшее RSS и наибольший R^2 , так как эти величины связаны с ошибкой обучения.
- Нужно выбрать модель с низкой тестовой ошибкой, а не с малой ошибкой обучения. Таким образом, RSS и R^2 не подходят для выбора лучшей модели среди набора моделей с различным количеством предикторов.
- Мы можем косвенно оценить ошибку тестирования, делая *поправку* на ошибку обучения, чтобы учесть смещения из-за *переобучения*.
- Мы можем *непосредственно* оценить погрешность тестирования, используя либо подход использования валидационного набора, либо подход кросс-валидации.

C_p , AIC, BIC и скорректированный коэффициент детерминации R^2

- Эти методы настраивают ошибку обучения для размера модели, и могут быть использованы для выбора среди множества моделей с различным числом переменных.
- На рисунке показаны C_p , BIC, и скорректированный R^2 для лучшей модели каждого размера.



Некоторые подробности

- *Mallow* C_p :

$$C_p = \frac{1}{n} (\text{RSS} + 2d\hat{\sigma}^2),$$

- где d – обобщенное значение используемых параметров и $\hat{\sigma}^2$ – ошибка оценки дисперсии ε , связанной с каждым измерением отклика.
- Критерий *AIC*, определяемый для широкого класса моделей, рассчитывается методом максимального правдоподобия:

$$AIC = -2 \log L + 2d$$

где L – значение функции логарифмического правдоподобия для оцениваемой модели.

- В случае линейной модели с гауссовскими ошибками, максимальное правдоподобие и наименьшие квадраты – это одно и то же, и C_p и *AIC* эквивалентны.

BIC

$$\text{BIC} = \frac{1}{n} (\text{RSS} + \log(n)d\hat{\sigma}^2).$$

- Аналогично C_p , BIC будет небольшой для моделей с низкой ошибкой тестирования, и поэтому мы выбираем модель, которая имеет самое низкое значение BIC .
- Обратите внимание на то, что BIC заменяет $2d\hat{\sigma}^2$, используемое C_p , на $\log(n)d\hat{\sigma}^2$, где n - число наблюдений.
- Так как $\log n > 2$ для любого $n > 7$, BIC статистики в целом сильнее штрафует модели со многими переменными, и, следовательно, приводит к выбору модели меньшего размера, чем C_p

Вычисление R^2

- Для модели наименьших квадратов с d переменными скорректированная R^2 статистика рассчитывается как:

$$\text{Adjusted } R^2 = 1 - \frac{\text{RSS}/(n - d - 1)}{\text{TSS}/(n - 1)}.$$

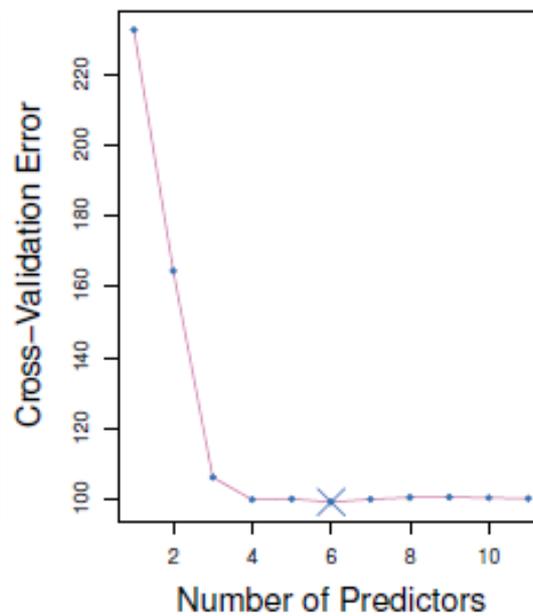
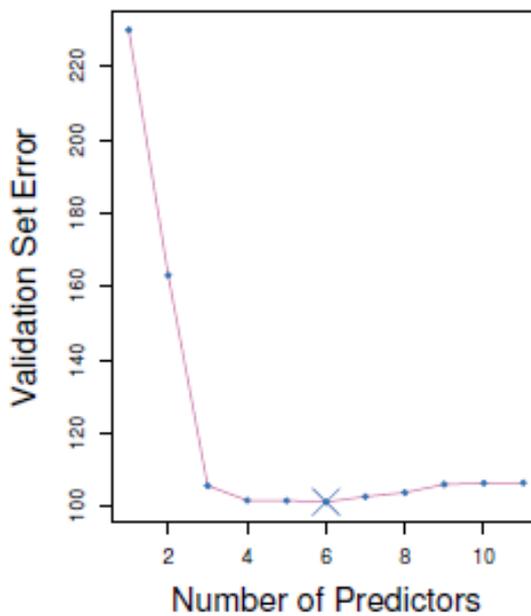
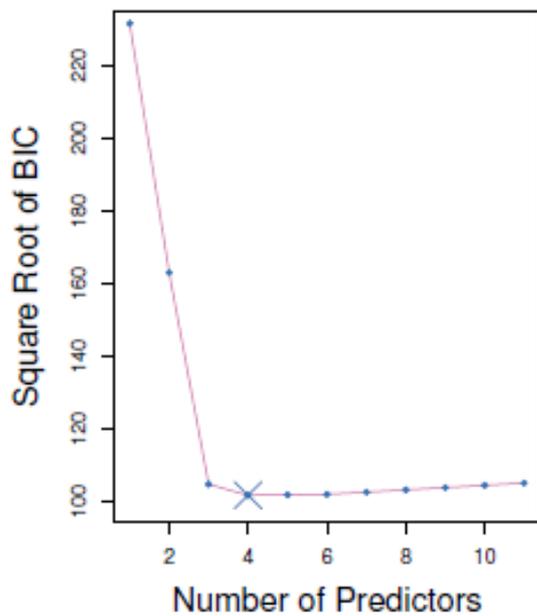
где TSS – полная сумма квадратов.

- В отличие от C_p , AIC и BIC, для которых *малое* значение указывает на модель с низкой ошибкой тестирования, *большое* значение скорректированного R^2 соответствует модели с небольшой ошибкой тестирования.
- В отличие от статистики R^2 , скорректированная R^2 статистика наказывает за включение ненужных переменных в модель.

Валидация и кросс-валидация

- Каждая из процедур возвращает последовательность моделей M_k , индексированная по размеру модели $k = 0, 1, 2, \dots$. Наша задача здесь состоит в выборе \hat{k} . После выбора, мы определим модель $M_{\hat{k}}$
- Вычислим ошибку на валидационном наборе или ошибку перекрестной проверки для каждой модели M_k , а затем выберем k , для которого полученная ошибка тестирования является наименьшей.
- Эта процедура имеет преимущество по сравнению с AIC, BIC, C_p и скорректированным R^2 , которое состоит в том, что оно обеспечивает непосредственную оценку тестовой ошибки, и *не требует оценки дисперсии ошибки*
- Она также может быть использована в более широком диапазоне задач выбора модели, даже в случаях, когда трудно точно определить число степеней свободы модели (например количество предикторов в модели) или трудно оценить дисперсию ошибок

Пример



Узел Regression в EM

Property	Value
General	
Node ID	Reg
Imported Data	...
Exported Data	...
Notes	...
Train	
Variables	...
Equation	
Main Effects	Yes
Two-Factor Inte	No
Polynomial Term	No
Polynomial Deg	2
User Terms	Yes
Term Editor	...
Class Targets	
Regression Type	Logistic Regress
Link Function	Logit
Model Options	
Suppress Inter	No
Input Coding	Deviation
Model Selection	
Selection Model	Backward
Selection Criteri	Validation Error
Use Selection D	No
Selection Option	...
Optimization Or	
Technique	Default
Default Optimizer	Yes
Max Iterations	0
Max Function Ca	0
Maximum Time	1 Hour
Convergence Cr	
Uses Defaults	Yes
Options	...
Output Options	
Confidence Limi	No
Save Covariance	No
Covariance	No

Настройка уравнения

Настройка пошагового метода и критерия выбора модели

Настройка отчетности (статистики, доверительные интервалы и т.д.)

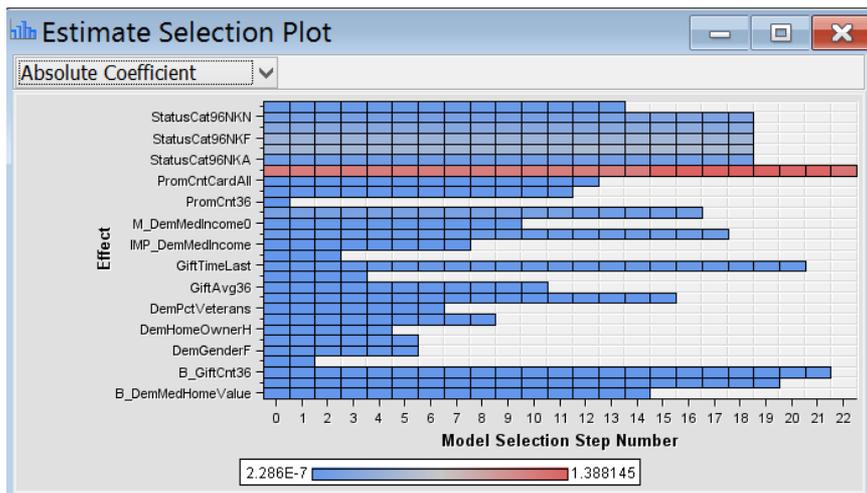
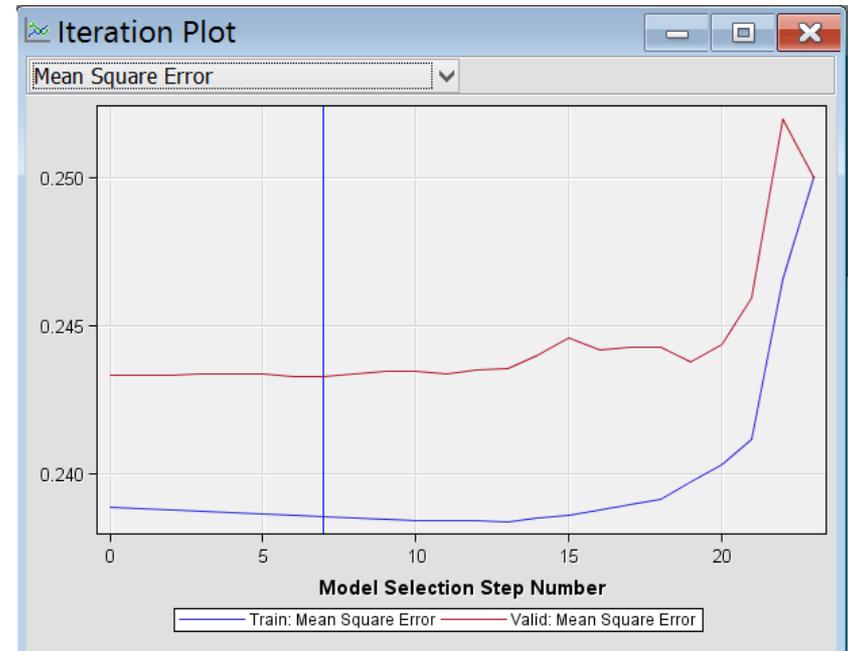
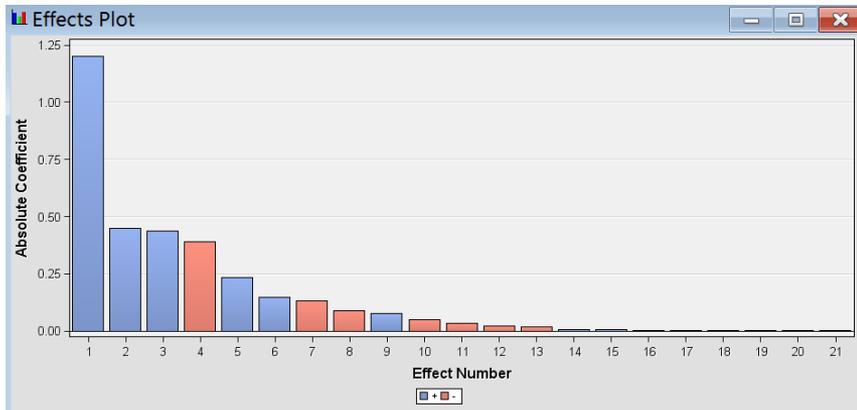
Property	Value
Sequential Order	No
Entry Significance Level	0.05
Stay Significance Level	0.0
Start Variable Number	0
Stop Variable Number	0
Force Candidate Effects	0
Hierarchy Effects	Class
Moving Effect Rule	None
Maximum Number of Steps	0

Entry Significance Level

Significance level for adding variables in forward or stepwise regression.

OK Cancel

Узел Regression в EM



Fit Statistics

Target	Target Label	Fit Statistics	Statistics Label	Train	Validation	Test
TargetB	Target Gift Flac	AIC	Akaike's Infor...	6507.798		
TargetB	Target Gift Flac	ASE	Average Squa...	0.237527	0.243299	
TargetB	Target Gift Flac	AVERR	Average Error...	0.667541	0.679739	
TargetB	Target Gift Flac	DFE	Degrees of Fr...	4822		
TargetB	Target Gift Flac	DFM	Model Degree...	21		
TargetB	Target Gift Flac	DFT	Total Degree...	4843		
TargetB	Target Gift Flac	DIV	Divisor for ASE	9686	9686	
TargetB	Target Gift Flac	ERR	Error Function	6465.798	6583.952	
TargetB	Target Gift Flac	FPE	Final Predict...	0.239595		
TargetB	Target Gift Flac	MAX	Maximum Abs...	0.836742	0.825488	
TargetB	Target Gift Flac	MSE	Mean Square ...	0.238561	0.243299	
TargetB	Target Gift Flac	NOBS	Sum of Frequ...	4843	4843	
TargetB	Target Gift Flac	NW	Number of Es...	21		
TargetB	Target Gift Flac	RASE	Root Average...	0.487367	0.493254	

Методы регуляризации (штраф за сложность)

Регрессия Ridge u Lasso

- Методы выбора подмножества используют метод наименьших квадратов для линейной модели, которая содержит подмножество предикторов.
- В качестве альтернативы, можно построить модель содержащую все p предикторов с использованием методики, которая *ограничивает или регуляризирует* оценки коэффициентов, или, что эквивалентно, что сводит некоторые оценки коэффициентов к нулю.
- Может быть не сразу понятно, почему такое ограничение должно улучшить построение модели, но оказывается, что сокращение количества коэффициентов может значительно уменьшить их дисперсию.

Гребневая регрессия

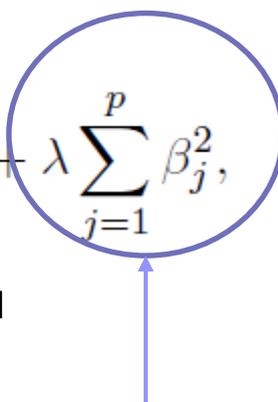
- Напомним, что процедура подгонки по методу наименьших квадратов оценивает коэффициенты $\beta_0, \beta_1, \dots, \beta_p$, минимизируя:

$$\text{RSS} = \sum_{i=1}^n \left(y_i - \beta_0 - \sum_{j=1}^p \beta_j x_{ij} \right)^2.$$

- Оценки коэффициентов $\hat{\beta}^R$ ridge-регрессии напротив являются значениями, которые надо минимизировать

$$\sum_{i=1}^n \left(y_i - \beta_0 - \sum_{j=1}^p \beta_j x_{ij} \right)^2 + \lambda \sum_{j=1}^p \beta_j^2 = \text{RSS} + \lambda \sum_{j=1}^p \beta_j^2,$$

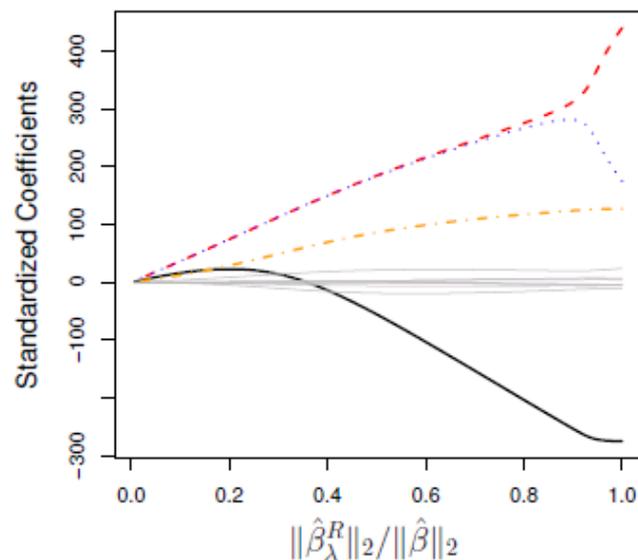
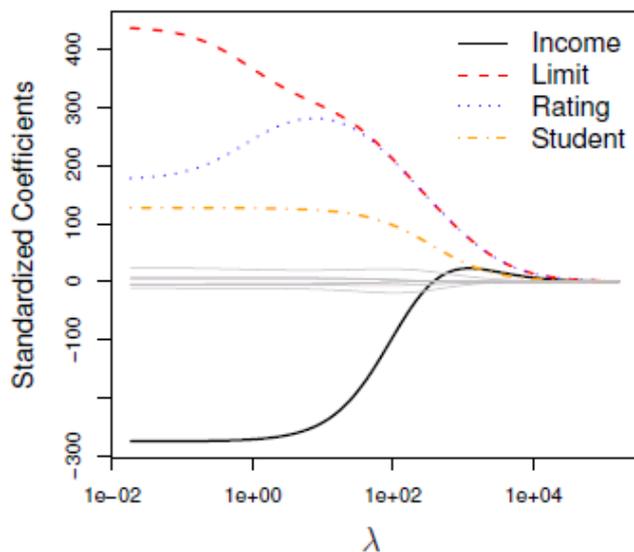
где $\lambda \geq 0$ - параметр настройки, который задается (рассчитывается) независимо.



Штраф за сложность

Гребневая регрессия

- Гребневая регрессия (как и МНК) стремится найти коэффициенты, которые дают наименьшее RSS.
- Но, второй член, $\lambda \sum_j \beta_j^2$, называемый штрафом сокращения, мал при β_1, \dots, β_p близких к нулю, и поэтому имеет место эффект сведения оценок β_j к нулю. Параметр настройки λ (подбор кросс-валидацией) служит для управления относительным влиянием этих двух членов на оценки коэффициентов



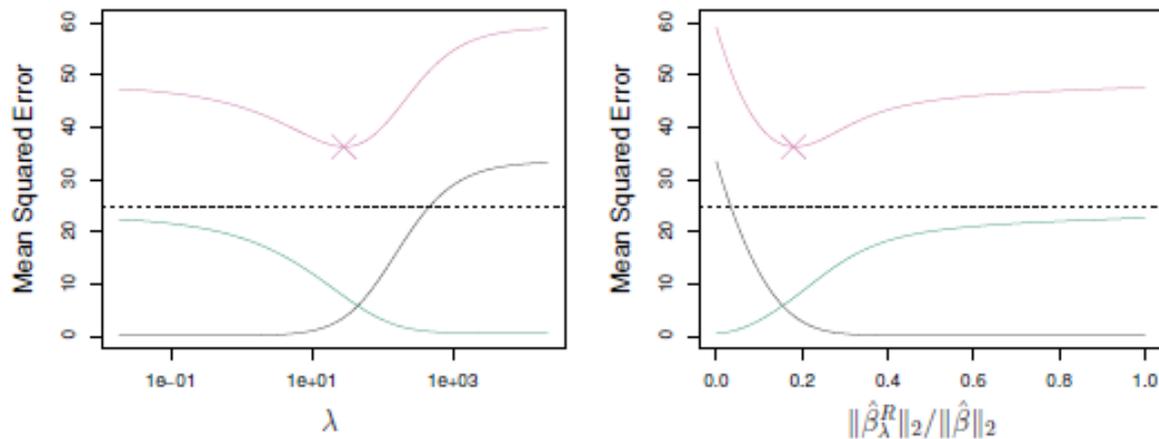
Гребневая регрессия: масштабирование предикторов

- Оценки коэффициентов стандартным методом наименьших квадратов являются масштабируемым: умножая X_j на константу c просто приводит к масштабированию оценок коэффициентов наименьших квадратов на коэффициент $1/c$. Другими словами, независимо от того, как масштабируется j -ый предиктор, $X_j \hat{\beta}_j$ останется прежним.
- Оценки коэффициентов гребневой регрессии наоборот могут *существенно* измениться при умножении заданного предиктора на константу, из-за суммы квадратов коэффициентов в штрафной части целевой функции регрессии.
- Поэтому, лучше всего применять гребневую-регрессию после *стандартизации предикторов*, используя формулу

$$\tilde{x}_{ij} = \frac{x_{ij}}{\sqrt{\frac{1}{n} \sum_{i=1}^n (x_{ij} - \bar{x}_j)^2}}$$

Почему Ridge регрессия дает улучшения по сравнению с методом наименьших квадратов?

Компромисс отклонение-дисперсия



Смоделированные данные с $n = 50$ наблюдениями, $p = 45$ предикторами все имеют отличные от нуля коэффициенты. Квадратичное смещение (черное), дисперсия (зеленая) и среднеквадратичная ошибка тестирования (фиолетовая) для предикторов *ridge*-регрессии на смоделированном наборе данных, в зависимости от λ и $\|\hat{\beta}_\lambda^R\|_2 / \|\hat{\beta}\|_2$. Горизонтальные пунктирные линии указывают на минимально возможное значение MSE.

Lasso

- Гребневая регрессия имеет один очевидный недостаток: в отличие от отбора подмножества, которое, как правило, выбирает модели, которые включают только подмножество переменных, гребневая регрессия будет включать в себя все p предикторов в конечной модели
- *Lasso* - относительно недавняя альтернатива, которая преодолевает этот недостаток. Коэффициенты *lasso* минимизируют величину

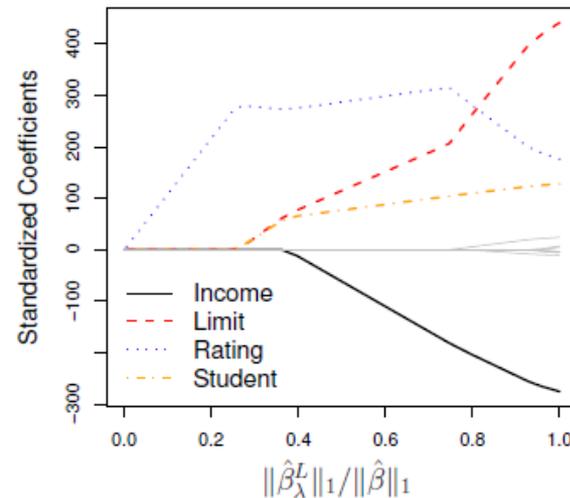
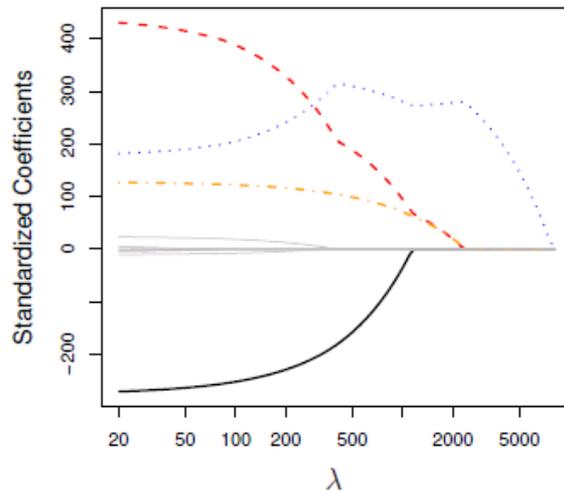
$$\sum_{i=1}^n \left(y_i - \beta_0 - \sum_{j=1}^p \beta_j x_{ij} \right)^2 + \lambda \sum_{j=1}^p |\beta_j| = \text{RSS} + \lambda \sum_{j=1}^p |\beta_j|.$$

ℓ_2 ℓ_1 (circled term)

Штраф за сложность

Lasso: продолжение

- В случае Lasso, штраф имеет эффект сведения некоторых оценок коэффициентов в точности к нулю, когда параметр настройки λ достаточно велик.
- Следовательно, так же, как выбор лучшего подмножества, lasso выполняет *отбор переменных*.
- lasso приводит к *разреженным* моделям - моделям, которые включают только подмножество переменных.
- Как и в ridge регрессии, выбирая хорошее значение λ для lasso имеет решающее значение



Выбор переменных для регрессии Lasso

Почему lasso, в отличие от ridge-регрессии, приводит к оценкам коэффициентов, которые в точности равны нулю?

Можно показать, что оценки коэффициентов lasso и ridge регрессии решают проблемы

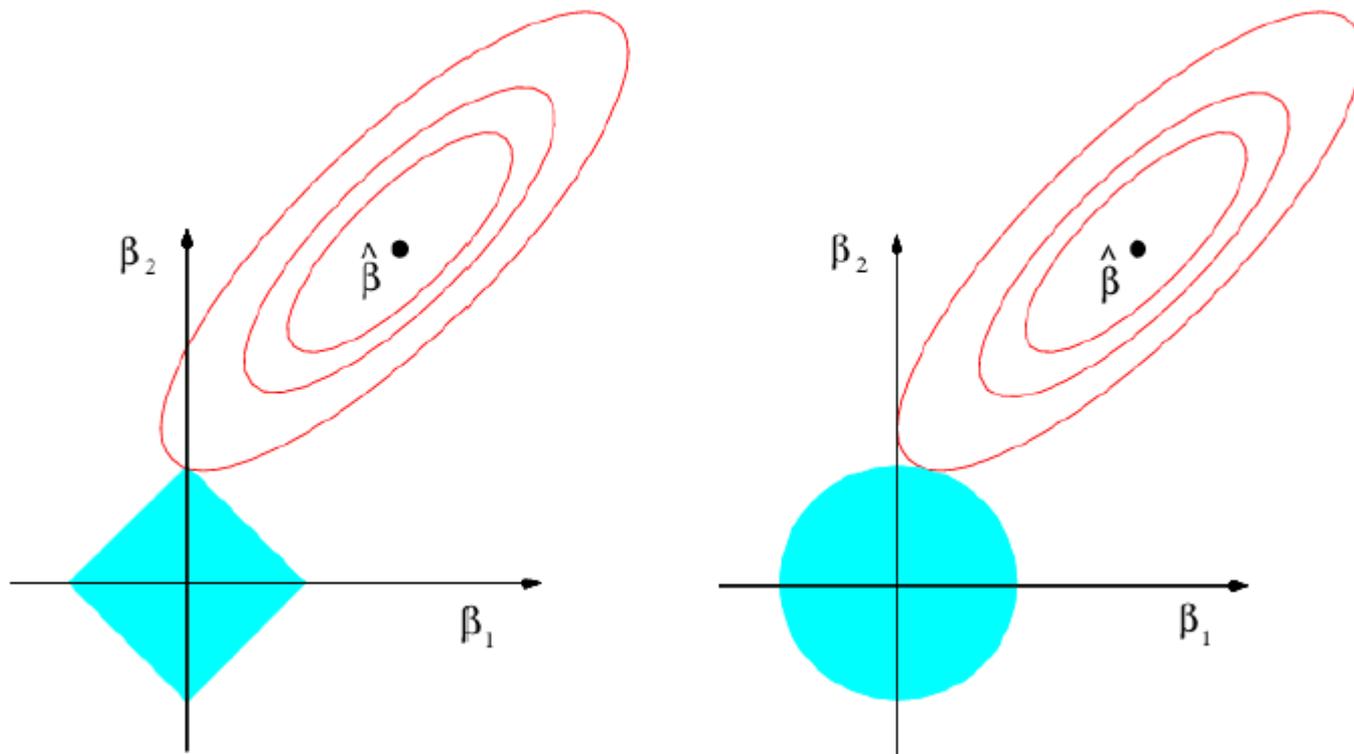
$$\underset{\beta}{\text{minimize}} \sum_{i=1}^n \left(y_i - \beta_0 - \sum_{j=1}^p \beta_j x_{ij} \right)^2 \quad \text{при условии} \quad \sum_{j=1}^p |\beta_j| \leq s$$

и

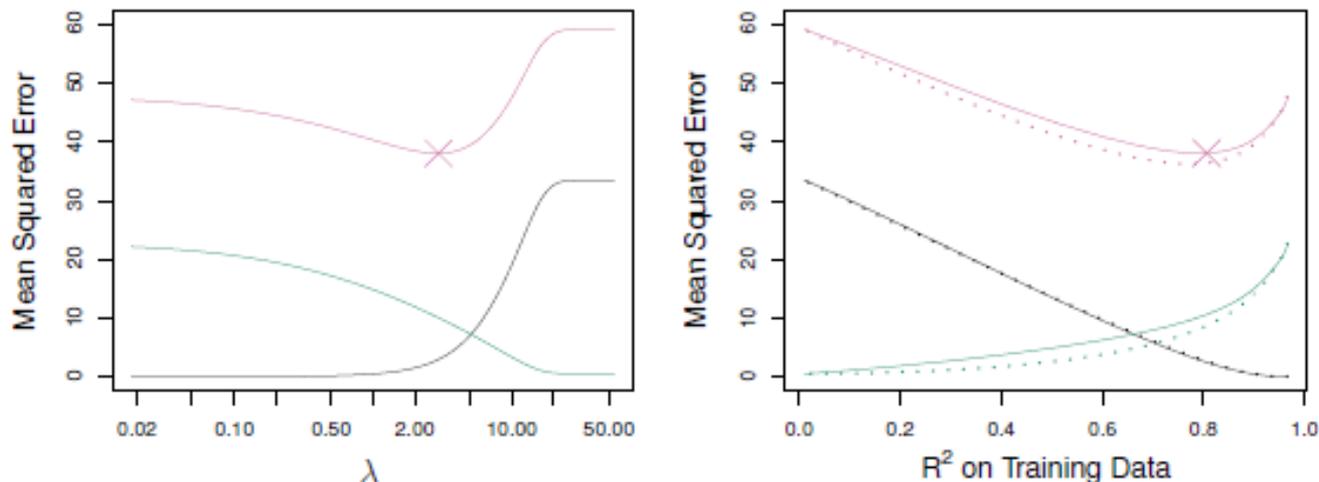
$$\underset{\beta}{\text{minimize}} \sum_{i=1}^n \left(y_i - \beta_0 - \sum_{j=1}^p \beta_j x_{ij} \right)^2 \quad \text{при условии} \quad \sum_{j=1}^p \beta_j^2 \leq s,$$

соответственно.

Иллюстрация регрессии Lasso



Сравнение Lasso и гребневой регрессии



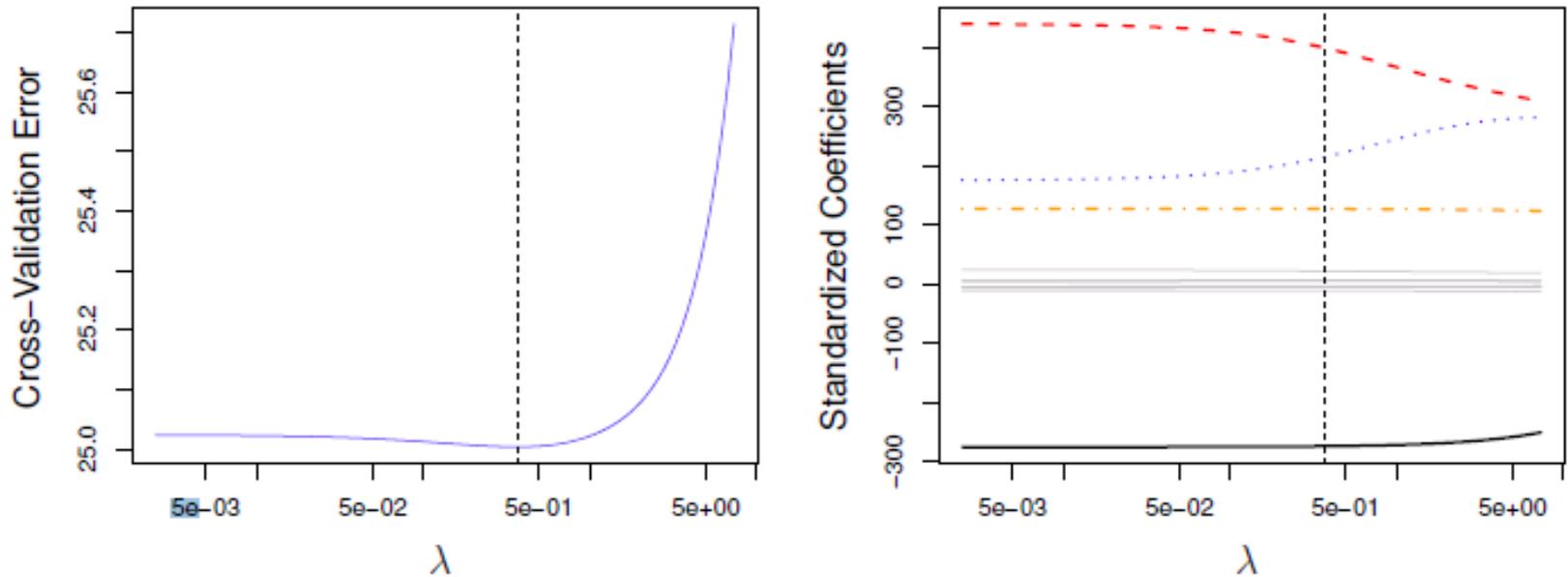
Слева: Графики квадрата смещения (черная), дисперсии (зеленая) и тестовой MSE (фиолетовая) для *lasso* на смоделированном наборе данных

Справа: Сравнение квадрата смещения, дисперсии и тестовой MSE между *lasso* (сплошная линия) и *ridge* (пунктирная). Оба построены относительно R^2 на обучающих данных. Крестиками на обоих графиках обозначена *lasso* модель, для которой MSE является наименьшим

Выбор параметров настройки для Ridge регрессии и Lasso

- Что же касается выбора подмножества, для ridge-регрессии и lasso нам нужен способ определения, какая из рассматриваемых моделей лучше.
- То есть нам нужен метод выбора значения для параметра настройки λ или, что эквивалентно, значение s .
- *Перекрестная проверка* обеспечивает простой способ решения этой проблемы. Выберем сетку значений λ и вычислим частоту ошибок кросс-валидации для каждого значения λ .
- Затем мы выбираем значение параметра настройки, для которого ошибка перекрестной проверки является наименьшей.
- И, наконец, модель перестраивается с использованием всех имеющихся объектов и выбранного значения параметра настройки.

Пример



Слева: ошибки перекрестной проверки, которые являются результатом применения ridge регрессии для различных значений λ .

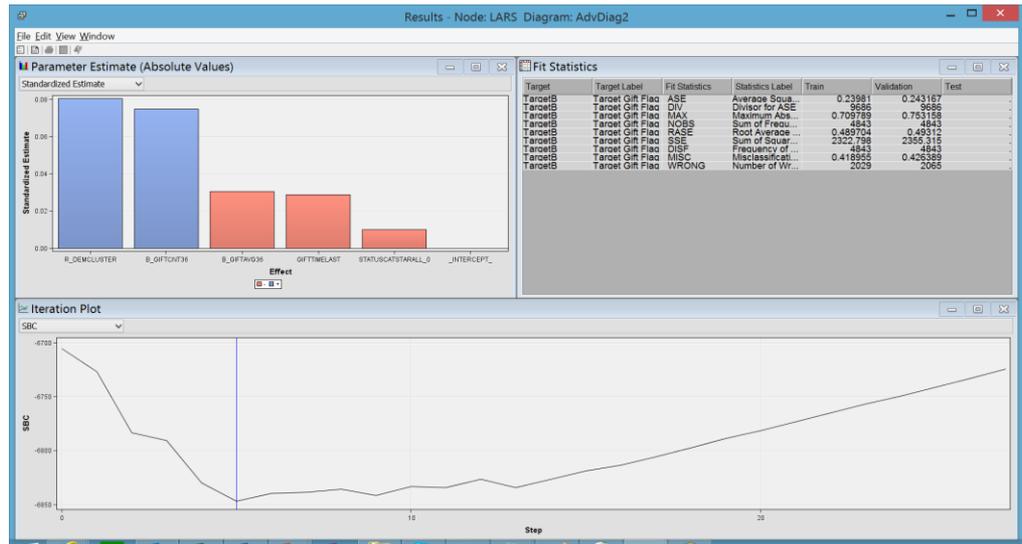
Справа: оценки коэффициентов в зависимости от λ . Вертикальные пунктирные линии обозначают значение λ , выбранное в результате перекрестной проверки.

Узел LARS в EM

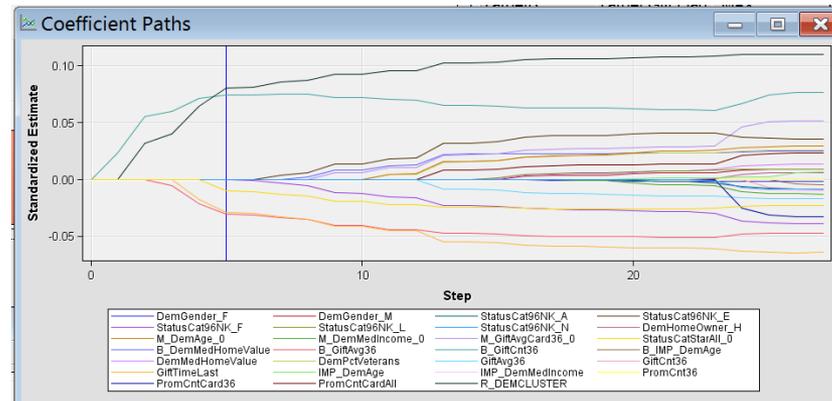
результаты



Настройка отбора переменных и выбор модели



Property	Value
General	
Node ID	LARS
Imported Data	
Exported Data	
Notes	
Train	
Variables	
<input type="checkbox"/> Modeling Techn	
Use Class Inputs	Yes
Intercept	Yes
Variable Selectio	LASSO
Model Selection	SBC
Path Stopping C	Maximum Steps
Maximum Steps	200
<input type="checkbox"/> Cross Validation	
Cross Validation	Random
CV Fold	5
Seed	12345
<input type="checkbox"/> Reports	
Details	Summary



Методы сокращения размерности

- Методы, которые мы обсуждали до сих пор в этой главе, были связаны с построением модели линейной регрессии по методу наименьших квадратов или подхода с использованием исходных предикторов X_1, X_2, \dots, X_p .
- Теперь мы будем исследовать класс подходов, которые *преобразуют* предикторы и строят модели методом наименьших квадратов с использованием преобразованных переменных. Мы будем называть эти методы методами *сокращения размерности*.

Методы сокращения размерности: подробности

- Пусть Z_1, Z_2, \dots, Z_M – M линейных комбинаций ($M < p$) наших исходных p предикторов. т.е.

$$Z_m = \sum_{j=1}^p \phi_{mj} X_j \quad (1)$$

для некоторых $\phi_{m1}, \dots, \phi_{mp}$.

- Затем мы можем построить модель линейной регрессии

$$y_i = \theta_0 + \sum_{m=1}^M \theta_m z_{im} + \epsilon_i, \quad i = 1, \dots, n, \quad (2)$$

используя МНК.

- Отметим, что в модели (2), коэффициенты регрессии заданы значениями $\theta_0, \theta_1, \dots, \theta_M$. Если $\phi_{m1}, \dots, \phi_{mp}$.
- подобраны хорошо, то такие подходы к снижению размерности могут быть лучше, чем МНК регрессия.

- Заметим, что из определения (1) следует,

$$\sum_{m=1}^M \theta_m z_{im} = \sum_{m=1}^M \theta_m \sum_{j=1}^p \phi_{mj} x_{ij} = \sum_{j=1}^p \sum_{m=1}^M \theta_m \phi_{mj} x_{ij} = \sum_{j=1}^p \beta_j x_{ij},$$

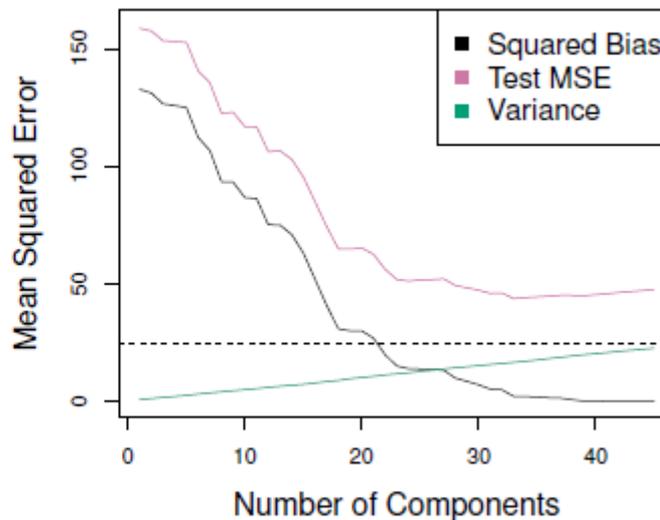
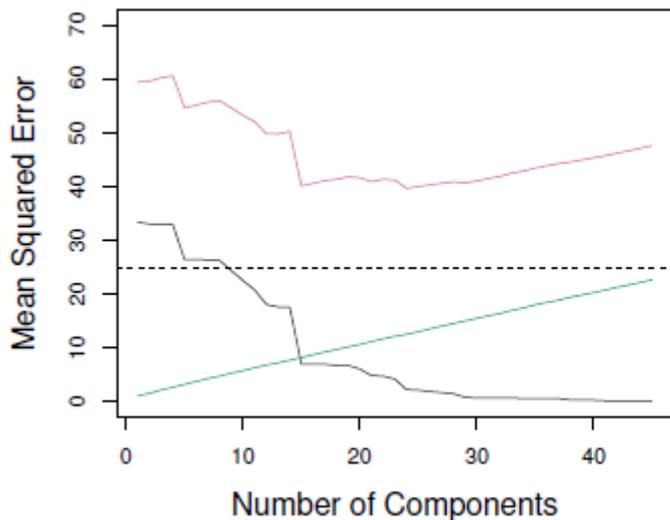
где
$$\beta_j = \sum_{m=1}^M \theta_m \phi_{mj}. \quad (3)$$

- Следовательно, модель (2) можно рассматривать как частный случай исходной модели линейной регрессии.
- Снижение размерности необходимо для ограничения коэффициентов β_j , так как теперь они должны принимать форму (3).
- Это может дать выигрыш в компромиссе дисперсии смещения.

Регрессия главных компонент

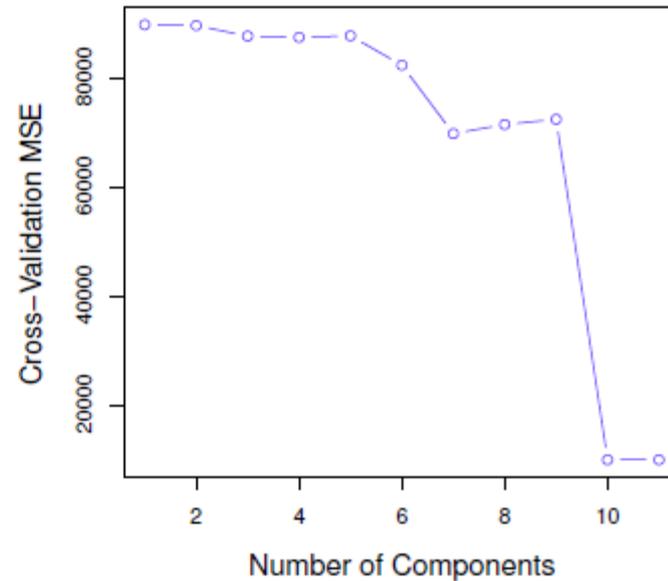
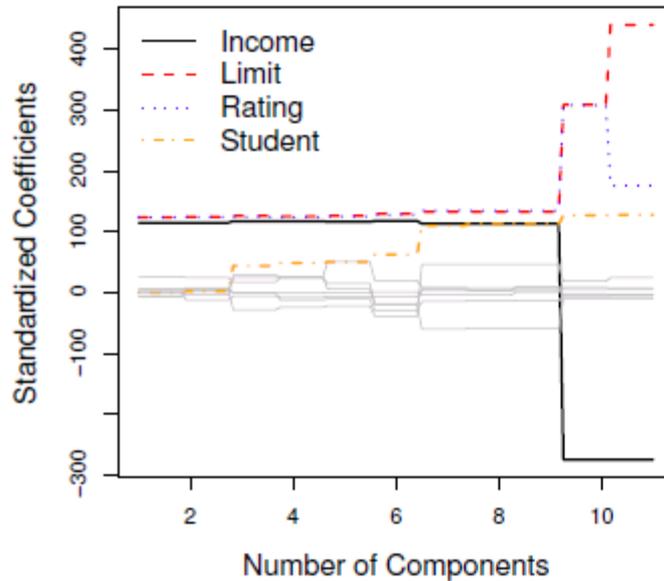
- Мы применяем анализ главных компонент (РСА), чтобы определить линейные комбинации предикторов для применения в регрессии.
- Первый главный компонент соответствует (нормализованной) линейной комбинации переменных с самой большой дисперсией.
- Второй главный компонент имеет самую большую дисперсию, при условии отсутствия корреляции с первым.
- И так далее.
- Поэтому если мы имеем много скоррелированных исходных переменных, мы заменим их с небольшим набором главных компонент, которые отражают их совместное изменение.

Применения регрессии главных КОМПОНЕНТ



PCR применена к двум наборам смоделированных данных. Черные, зеленые и фиолетовые линии соответствуют квадрату смещения, дисперсии и тестовой среднеквадратической ошибки соответственно.

Выбор количества компонент M



- Слева: Оценки стандартизованного PCR коэффициента для различных значений M.
- Справа: MSE кросс-валидации с десятью папками, используя PCR как функцию от M.

Метод частичных наименьших квадратов

- PCR определяет линейные комбинации, или *направления*, которые наилучшим образом представляют предикторы X_1, \dots, X_p .
- Эти направления определяются *обучением без учителя*, так как отклик Y не используется при определении направлений главных компонент.
- То есть отклик не *контролирует* определение главных компонент.
- Следовательно, PCR страдает от потенциально серьезного недостатка: нет никакой гарантии, что направления, которые наилучшим образом объясняют предикторы, также будут лучшими направлениями при использовании для прогнозирования отклика.

Метод частичных наименьших квадратов (PLS): продолжение

- Подобно PCR, PLS является метод снижения размерности, который сначала определяет новый набор признаков Z_1, \dots, Z_M , которые являются линейными комбинациями исходных признаков, а затем строит линейную модель с помощью OLS с использованием этих M новых признаков.
- Но в отличие от PCR, PLS определяет эти признаки на основе контролируемого обучения - то есть, он использует отклик Y с целью выявления новых признаков, которые не только хорошо аппроксимируют исходные признаки, но и *связаны с откликом*.
- Грубо говоря, подход PLS пытается определить направления, которые позволяют объяснить как отклики, так и предикторы.

$$\max_{|\alpha|=1, v_l^T S \alpha=0, l=1, \dots, m-1} \text{Corr}^2(y, X\alpha) \text{Var}(X\alpha)$$

Подробности о методе частичных наименьших квадратов

- После стандартизации p предикторов, PLS вычисляет первое направление Z_1 на основе установки каждого ϕ_{1j} в (1) равным коэффициенту простой линейной регрессии Y для X_j .
- Можно показать, что этот коэффициент пропорционален корреляции между Y и X_j .
- Следовательно, при вычислении $Z_1 = \sum_{j=1}^p \phi_{1j} X_j$, PLS устанавливает наибольший вес переменным, которые наиболее тесно связаны с откликом.
- Последующие направления определяются на основе расчета невязки, а затем повторения вышеописанного.

Узел PLS в EM

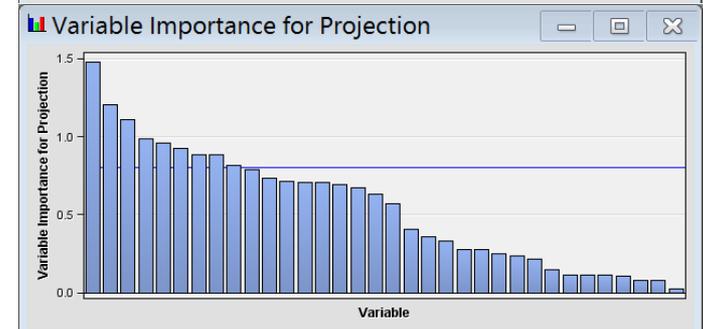
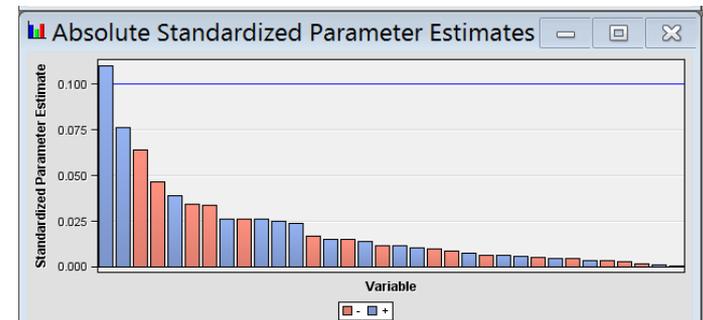
Property	Value
Train	
Variables	
Modeling Techn	
Regression Mod	PLS
PLS Algorithm	NIPALS
Maximum Iterat	200
Epsilon	1.0E-12
Number of Facto	
Default	Yes
Number of Facto	15
Cross Validation	
CV Method	None
CV N Parameter	7
Random CV Opt	
Number of Itera	10
Default No. of T	Yes
No. of Test Obs	100
Default Random	Yes
Random Seed	1234
Score	
Variable Selecti	
Variable Selecti	Both
Para. Est. Cutoff	0.1
VIP Cutoff	0.8
Export Selected	No
Hide Rejected V	Yes
Status	
Create Time	23.03.17 2:42

Выбор PLS или PCR



Настройка числа факторов

Настройка отбора переменных



Сравнение моделей в EM

■ Узел Model Comparison:

- Можно задавать целевой набор для оценки (train/test/validate)
- Критерий сравнения (ошибку, точность и т.д.)

Property	Value
General	
Node ID	MdlComp
Imported Data	...
Exported Data	...
Notes	...
Train	
Variables	...
Assessment Rep	
Number of Bins	20
ROC Chart	Yes
Recompute	No
Model Selection	
Selection Data	Default
Selection Statistic	ROC
Grid Selection Size	Default
Selection Table	Test
Selection Depth	10
Score	

