Kryteria wyboru modelu

Piotr J. Sobczyk

8 grudnia 2016

Na dzisiejszych zajęciach:

1. Dowiemy się na dlaczego nie zawsze chcemy używać wszystkich dostępnych zmiennych w modelu
2. Poznamy dwie metody wyboru modelu AIC i BIC, które oparte są na penalizowanej funkcji wiarogodności
3. Poznamy jak wybierać model przy pomocy walidacji krzyżowej (CV)

Dotychczas w danych z jakimi się stykaliśmy było sporo obserwacji i mało zmiennych. Jest jednak wiele zbiorów danych, gdzie liczba zmiennych jest porównywalna z liczbą obserwacji.

Przykład 1

Analiza danych genetycznych, dla każdego pacjenta mamy 500K zmiennych opisujących jego genom. Każdy podzbiór zmiennych daje nam inny model. Który z nich jest najlepszy?

Przykład 2

Chcemy dopasować model regresji wielomianowej

$$m(x) = E(Y|X = x) = \beta_0 + \beta_1 x + \cdots + \beta_p x^p$$

Potrzebujemy wybrać rząd modelu p. Możemy myśleć o modelach $M_1 \dots M_p$, które odpowiadają odpowiednim rzędom wielomianów.

Przykład 3

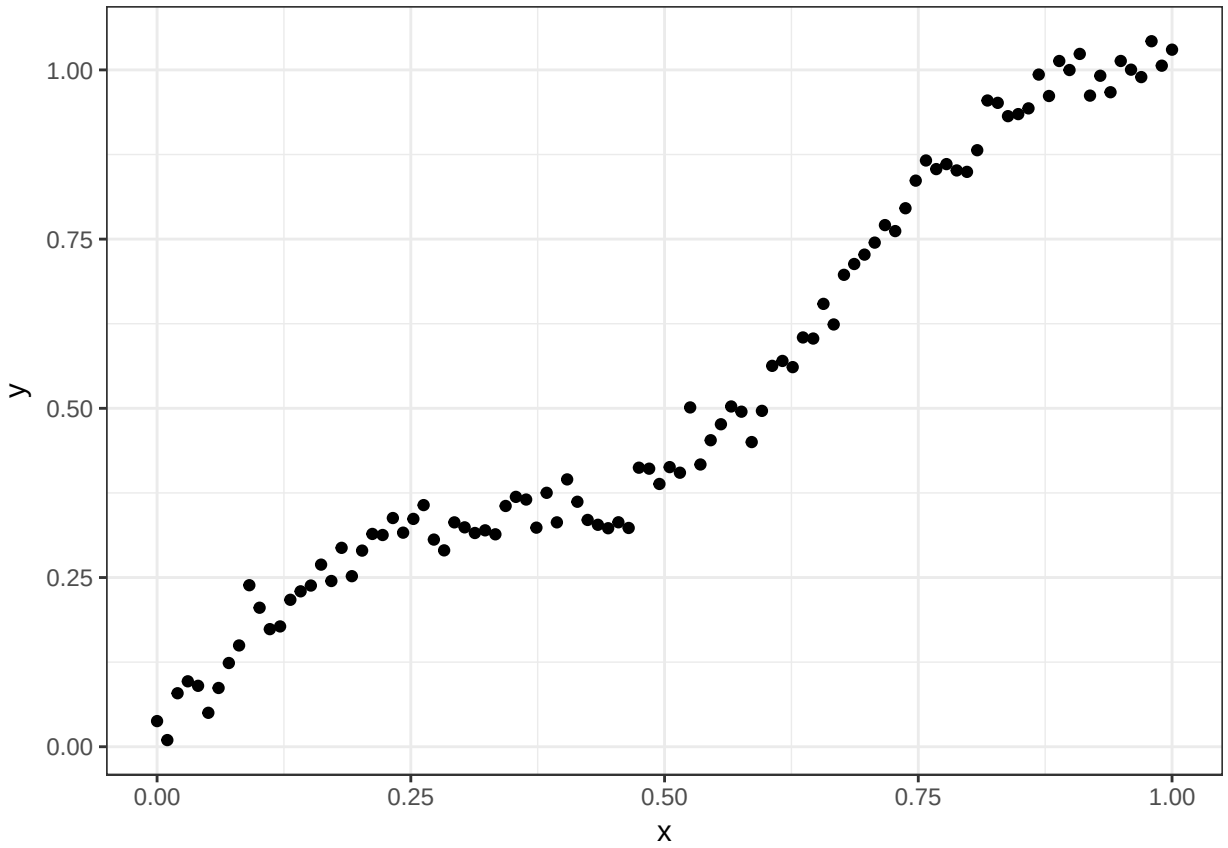
Mamy dane dotyczące zliczenia pewnych zdarzeń, na przykład, w eksperymencie biologicznym, patrzymy ile razy dany fragment RNA pojawia się w próbce. Tę liczbę możemy modelować za pomocą rozkładu Poisssona, rozkładu ujmenego dwumianowego lub np. rozkładu Poissona typu „zero-inflated“. Który model wybrać?

Wybór modelu

W przykładach powyżej zbudowanie modelu na wszystkich dostępnych zmiennych mija się z celem modelowania. Nie chodzi o to, żeby mieć cokolwiek, chcemy mieć model, który będzie przydatny. Wrócmy na chwilę do cytatu George Boxa:

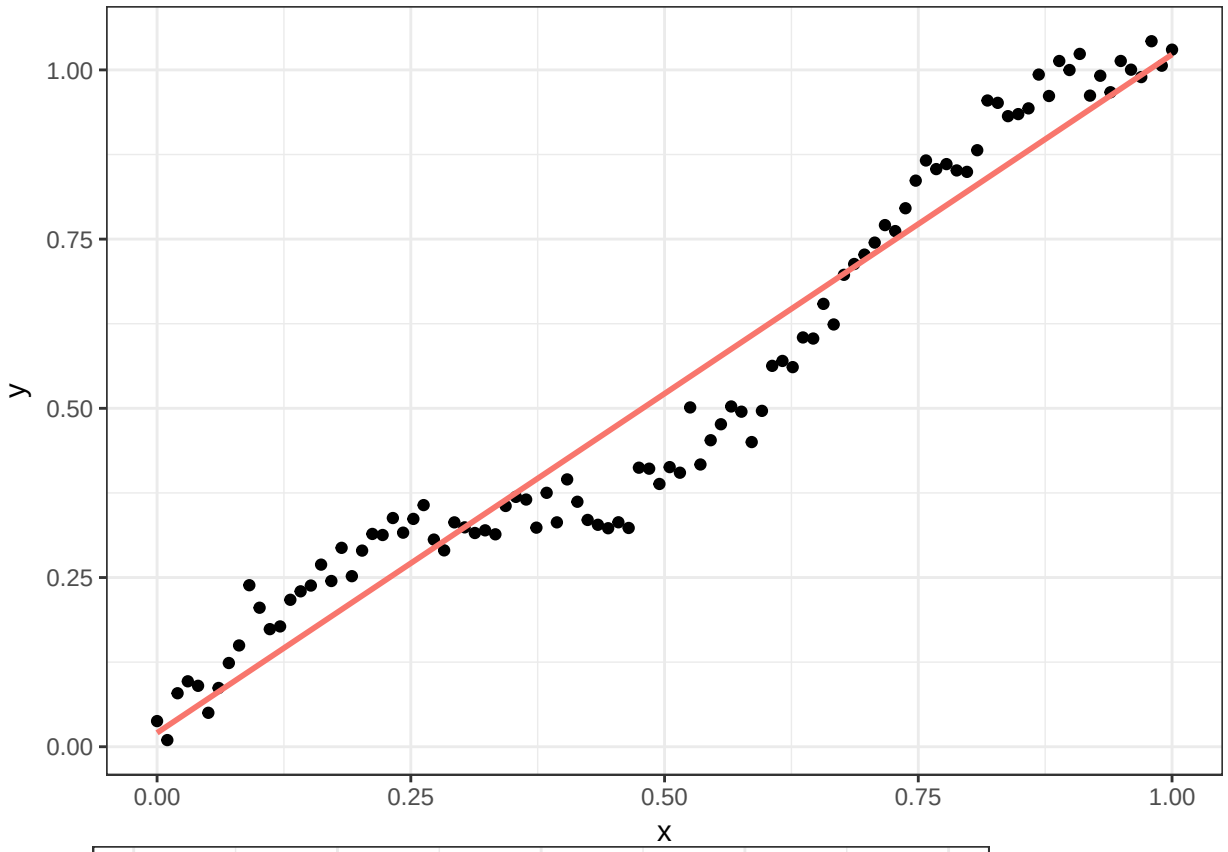
> Now it would be very remarkable if any system existing in the real world could be exactly represented by any simple model. However, cunningly chosen parsimonious models often do provide remarkably useful approximations. For example, the law PV = RT relating pressure P, volume V and temperature T of an “ideal” gas via a constant R is not exactly true for any real gas, but it frequently provides a useful approximation and furthermore its structure is informative since it springs from a physical view of the behavior of gas molecules. For such a model there is no need to ask the question “Is the model true?”. If “truth” is to be the “whole truth” the answer must be “No”. The only question of interest is “Is the model illuminating and useful?”.

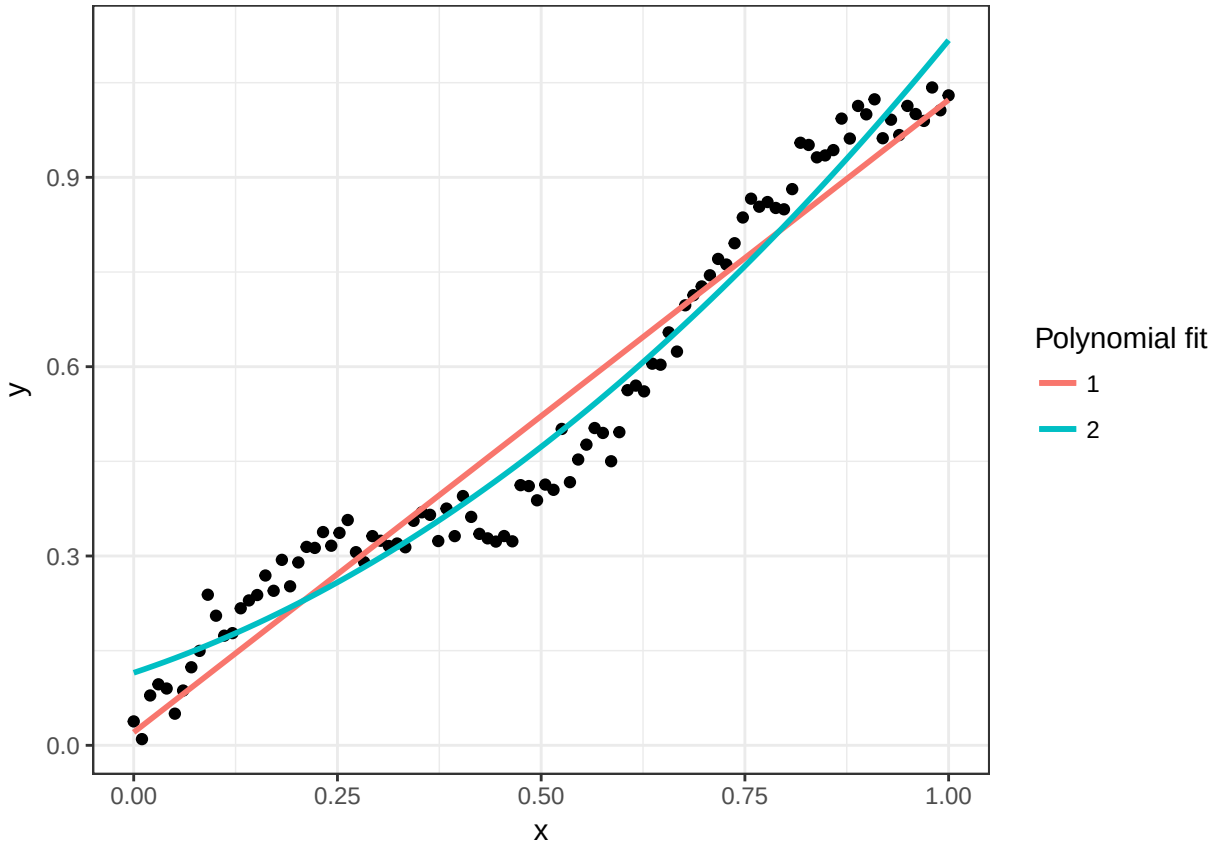
Jeśli liczba zmiennych jest porównywalna z liczbą obserwacji, to możemy być pewni, że nasz model nadmiernie dopasuje się do danych. Co to oznacza? Najłatwiej jest to zobaczyć na przykładzie.

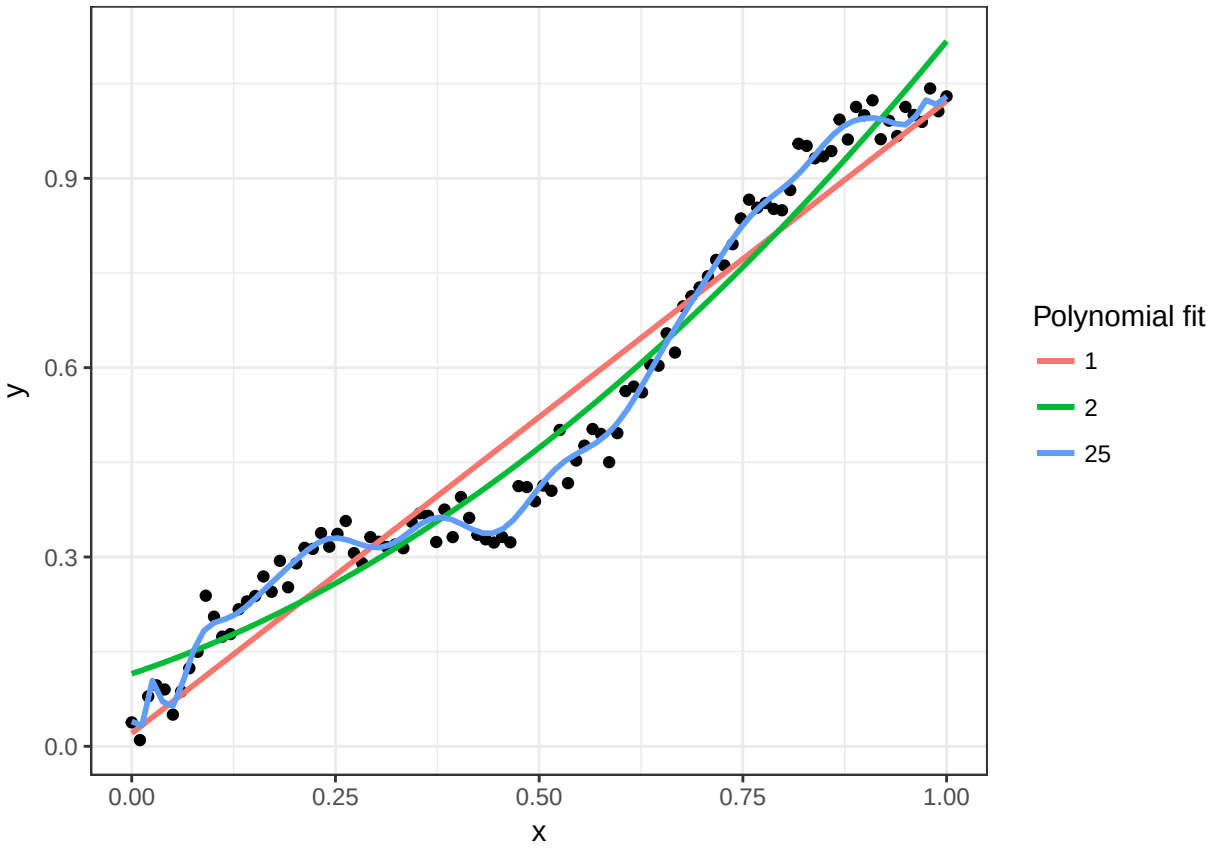


Jak dopasować krzywą opisującą te dane? Chcemy dopasować jakiś wielomian. Zauważmy najpierw, że jest to problem regresji liniowej.

y	x	x^2	$\dots$	x^p
y_1	x_1	x_1^2	$\dots$	x_1^p
y_2	x_2	x_2^2	$\dots$	x_2^p
$\vdots$	$\vdots$	$\vdots$	$\dots$	$\vdots$
y_n	x_n	x_n^2	$\dots$	x_n^p


1.00
0.75
0.50
0.25
0.00
y
0.00
0.25
0.50
0.75
1.00
x


0.9
0.6
0.3
0.0
y
0.00
0.25
0.50
0.75
1.00
x
Polynomial fit
1
2



W przypadku ostatnim mamy do czynienia z overfittingiem. Nadmiernym dopasowaniem do danych.

Metody wyboru modelu

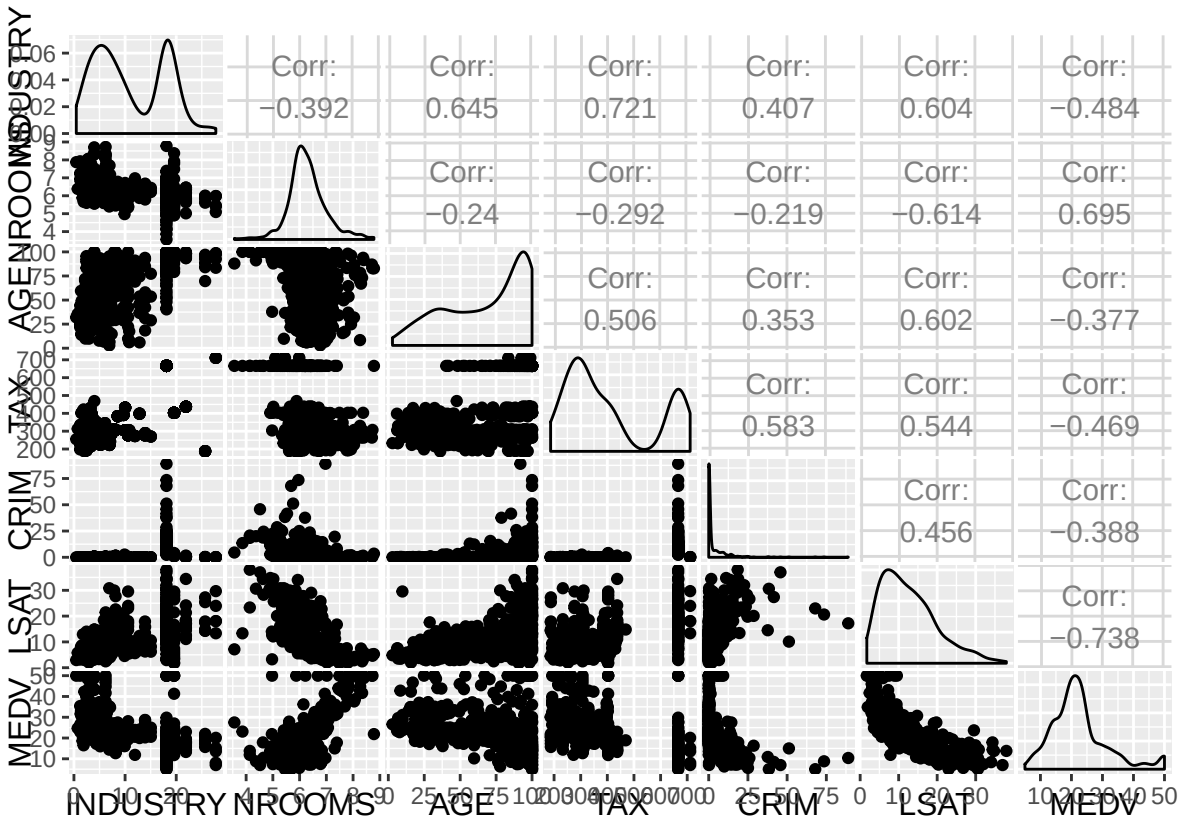
Przykład

Przykład pochodzi z wykładu Hyunseung Kang'a.

```
housing = read.csv("http://stat.wharton.upenn.edu/~khyuns/stat431/BostonHousing.txt")
str(housing)
```

```
## 'data.frame':    506 obs. of  14 variables:
##  $ CRIM    : num  0.00632 0.02731 0.02729 0.03237 0.06905 ...
##  $ ZN      : num  18 0 0 0 0 0 12.5 12.5 12.5 12.5 ...
##  $ INDUSTRY: num  2.31 7.07 7.07 2.18 2.18 2.18 7.87 7.87 7.87 7.87 ...
##  $ CHAR    : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ NOX     : num  0.538 0.469 0.469 0.458 0.458 0.458 0.524 0.524 0.524 0.524 ...
##  $ NROOMS  : num  6.58 6.42 7.18 7 7.15 ...
##  $ AGE     : num  65.2 78.9 61.1 45.8 54.2 58.7 66.6 96.1 100 85.9 ...
##  $ DIS     : num  4.09 4.97 4.97 6.06 6.06 ...
##  $ RAD     : int  1 2 2 3 3 3 5 5 5 5 ...
##  $ TAX     : int  296 242 242 222 222 222 311 311 311 311 ...
##  $ PTRATIO : num  15.3 17.8 17.8 18.7 18.7 18.7 15.2 15.2 15.2 15.2 ...
##  $ B       : num  397 397 393 395 397 ...
##  $ LSAT    : num  4.98 9.14 4.03 2.94 5.33 ...
##  $ MEDV    : num  24 21.6 34.7 33.4 36.2 28.7 22.9 27.1 16.5 18.9 ...
```

```
housing %>% select(INDUSTRY, NROOMS, AGE, TAX, CRIM, LSAT, MEDV) -> housing
ggpairs(housing)
```



BIC

Bayesian Information Criterion zostało zaproponowane przez Gideona Schwarza w latach 70-tych. Poniżej szkic wyprowadzenie, który jest dosyć prosty. Myślimy po Bayesowsku. Jakie jest prawdopodobieństwo modelu, gdy mamy dane?

$$P(M_j|Y) \propto P(Y|M_j)P(M_j),$$

gdzie $P(M_j)$ jest prawodopobieństwem apriori dla modelu j-tego, a $Y = (Y_1, \ldots, Y_n)$ to dane. Wybieramy ten model M_j, który maksymalizuje powyższą wartość. Równoważnie, możemy rozważać

$$\log P(M_j|Y) \propto logP(Y|M_j) + \log P(M_j),$$

Drugi element możemy rozpisać jako

$$P(Y|M_j) = \int_{\Theta_j} p(Y|M_j, \theta_j)\pi_j(\theta_j)d\theta_j = \int_{\Theta_j} L(\theta_j)\pi_j(\theta_j)d\theta_j,$$

gdzie Θ_j jest przestrzenią parametrów w modelu M_j, a L jest funkcją wiarogodności. Niestety najczęściej nie potrafimy tej całki policzyć. Pozostają nam przybliżenia.

Możemy rozwinąć całkę wokół θ_j maksymalizującego L, w tym celu stosujemy przybliżenie Laplace'a.

$$\log \int_{\Theta_j} L(\theta_j)\pi_j(\theta_j)d\theta_j \approx l(\hat{\theta_j}) - \frac{d_j}{2}\log n,$$

gdzie $l = \log L$, d_j jest wymiarem przestrzeni parametrów dla modelu M_j.

Ostatecznie, wybieramy model, który maksymalizuje:

$$arg\max_j \; l(\hat{\theta_j}) - \frac{d_j}{2}\log n + \log P(M_j)$$

Najczęściej zakłada się rozkład apriori jednostajny dla wszystkich modeli i BIC ma wtedy postać

$$BIC_{M_j} := arg\max_j \; l(\hat{\theta_j}) - \frac{d_j}{2}\log n$$

```
n = nrow(housing)
step(lm(log(MEDV)~INDUSTRY + NROOMS + AGE + TAX + CRIM, data=housing), direction="both", k=log(n))
```

```
## Start:  AIC=-1355.85
## log(MEDV) ~ INDUSTRY + NROOMS + AGE + TAX + CRIM
##
##            Df Sum of Sq    RSS     AIC
## - INDUSTRY  1    0.0319 32.270 -1361.6
## <none>                  32.238 -1355.8
## - AGE       1    0.8919 33.130 -1348.3
## - TAX       1    1.0323 33.270 -1346.1
## - CRIM      1    3.6262 35.864 -1308.1
## - NROOMS    1   16.1974 48.436 -1156.1
##
## Step:  AIC=-1361.58
## log(MEDV) ~ NROOMS + AGE + TAX + CRIM
##
##            Df Sum of Sq    RSS     AIC
## <none>                  32.270 -1361.6
## + INDUSTRY  1    0.0319 32.238 -1355.8
## - AGE       1    1.3420 33.612 -1347.2
## - TAX       1    1.7801 34.050 -1340.6
## - CRIM      1    3.5944 35.864 -1314.4
## - NROOMS    1   17.7340 50.004 -1146.2
##
## Call:
## lm(formula = log(MEDV) ~ NROOMS + AGE + TAX + CRIM, data = housing)
##
## Coefficients:
## (Intercept)       NROOMS          AGE          TAX         CRIM
##   1.6539739    0.2810492   -0.0021435   -0.0004775   -0.0121286
```

Jaka różnica w BIC jest istotna? Zauważmy, że eksponenta z różnicy w BIC to iloraz prawdopodobieństw aposteriori. Za publikacją Kass and Raftery (1995), przyjmuje się następujące wartości:

$BIC_{M_j} - BIC_{min}$	Evidence Against Model
0-2	Not worth more than a bare mention
2-6	Positive

$BIC_{M_j} - BIC_{min}$	Evidence Against Model
6-10	Strong
>10	Very Strong

Notabene, jest to naprawdę świetna praca i warto ją przeczytać jeśli jest się zainteresowanym myśleniem Bayesowskim.

AIC

Formuła tego kryterium to jest podobne do BIC:

$$AIC_{M_j} := arg\max_j \ l(\hat{\theta_j}) - 2d_j$$

Ponieważ zwykle $\frac{n}{2} > 2$, AIC ma tendencję do wybierania większych modeli. Kara za wprowadzenie dodatkowej zmiennej jest mniejsza niż w przypadku BIC.

```
step(lm(log(MEDV)~., data=housing) ,direction="both")
```

```
## Start:  AIC=-1546.49
## log(MEDV) ~ INDUSTRY + NROOMS + AGE + TAX + CRIM + LSAT
##
##            Df Sum of Sq    RSS     AIC
## - INDUSTRY  1    0.0033 23.167 -1548.4
## <none>                  23.163 -1546.5
## - AGE       1    0.1166 23.280 -1545.9
## - TAX       1    0.5003 23.664 -1537.7
## - CRIM      1    1.6177 24.781 -1514.3
## - NROOMS    1    2.7111 25.874 -1492.5
## - LSAT      1    9.0748 32.238 -1381.2
##
## Step:  AIC=-1548.41
## log(MEDV) ~ NROOMS + AGE + TAX + CRIM + LSAT
##
##            Df Sum of Sq    RSS     AIC
## <none>                  23.167 -1548.4
## - AGE       1    0.1565 23.323 -1547.0
## + INDUSTRY  1    0.0033 23.163 -1546.5
## - TAX       1    0.6435 23.810 -1536.5
## - CRIM      1    1.6493 24.816 -1515.6
## - NROOMS    1    2.7571 25.924 -1493.5
## - LSAT      1    9.1034 32.270 -1382.7
##
## Call:
## lm(formula = log(MEDV) ~ NROOMS + AGE + TAX + CRIM + LSAT, data = housing)
##
## Coefficients:
## (Intercept)       NROOMS          AGE          TAX         CRIM
##   2.6676472    0.1364344    0.0008293   -0.0002913   -0.0083699
##        LSAT
##  -0.0314759
```

Dla kryteriów AIC i BIC mamy sporo ciekawych twierdzeń, mówiących o ich skuteczności. Problem z ich używaniem jest bardzo praktyczny. Aby wybrać model maksymalizujący BIC należy sprawdzić wszystkie możliwe modele. Ich liczba jest wykładnicza w stosunku do liczby zmiennych. A jak coś rośnie wykładniczo, to oznacza to, że nie da się tego policzyć w skończonym czasie dla rozsądnie dużych danych. Już przy 20 zmiennych mamy do porównania ponad milion modeli!

Zamiast przeglądać wszystkie modele, stosuje się heurystyki oparte o zachłanne (greedy) przeczesywanie przestrzeni wszystkich modeli. Do tego zadania służy eRowa funkcja step. Wychodząc od modelu pustego (pełnego), dodaje (odejmuje) zmienne i patrzy czy BIC się zwiększy. Jeśli tak, idzie dalej, jeśli nie zatrzymuje się i nie szuka dalej. Niektóre strategie są bardziej inteligentne, ale zasada pozostaje taka sama.

Walidacja krzyżowa (cross-validation)

Walidacja krzyżowa (Cross-Validation, CV) jest metodą, która służy do wyboru parametrów modelu, i oparta jest na poznanym na zeszłym wykładzie frameworku podziału danych na zbiór treningowy i testowy.

K-krotna walidacja krzyżowa (k-fold CV) polega na losowym podziale danych na k części, z których każda po kolei służy za zbiór testowy.
Schemat jest następujący:

1. Podziel zbiór danych na k-podzbiorów o mniej więcej równej wielkości $\frac{n}{k}$.
2. Dla każdej podzbioru:

- Oznacz go jako zbiór testowy, a pozostałe pozdbiory jako zbiór treningowy
- Dopasuj model w oparciu jedynie o zbiór treningowy
- Oblicz błąd średniokwadratowy predykcji (Prediction Mean Squared Error, PMSE),

$$PMSE = \frac{1}{\#\text{ próbek testowych}} \sum_{\text{i - testowe}} (Y_i - \hat{Y}_i)^2$$

3. Policz średnią po wszystkich PMSE.

Spośród wszystkich modeli wybieramy ten, który daje najniższe MSE. Oczywiście zamiast MSE możemy użyć innej miary dopasowanie np. R^2. W przypadku klasyfikacji AUC, MCC lub jakiejkolwiek innej miary, która będzie odpowiadała temu, czego oczekujemy po modelu.

CV skupia się na błędzie/dokładności predykcji. Służy do wyboru optymalnego pod tym względem. W szczególności niekoniecznie da nam „prawdziwy" model o małej liczbie zmiennych. W przypadku małego zbioru danych CV może okazać się bardzo niestabilna, MSE może się bardzo mocno różnić pomiędzy „foldami". Jeśli chcemy upewnić się co do jakości predkcji można zastosować wielokrotną CV, czyli powtarzamy wielokrotnie podział na podzbiory. W szczególności możemy wtedy dostać coś w rodzaju bootstrapowych (a właściwie bardziej permutacyjnych) przedziałów ufności dla mierzonej cechy - MSE, R^2 czy AUC.

Jak wybrać parametr k? Standardowo bierzemy $k = 10$. Szczególnym przypadkiem jest leave-one-out CV, w której $k = n$.

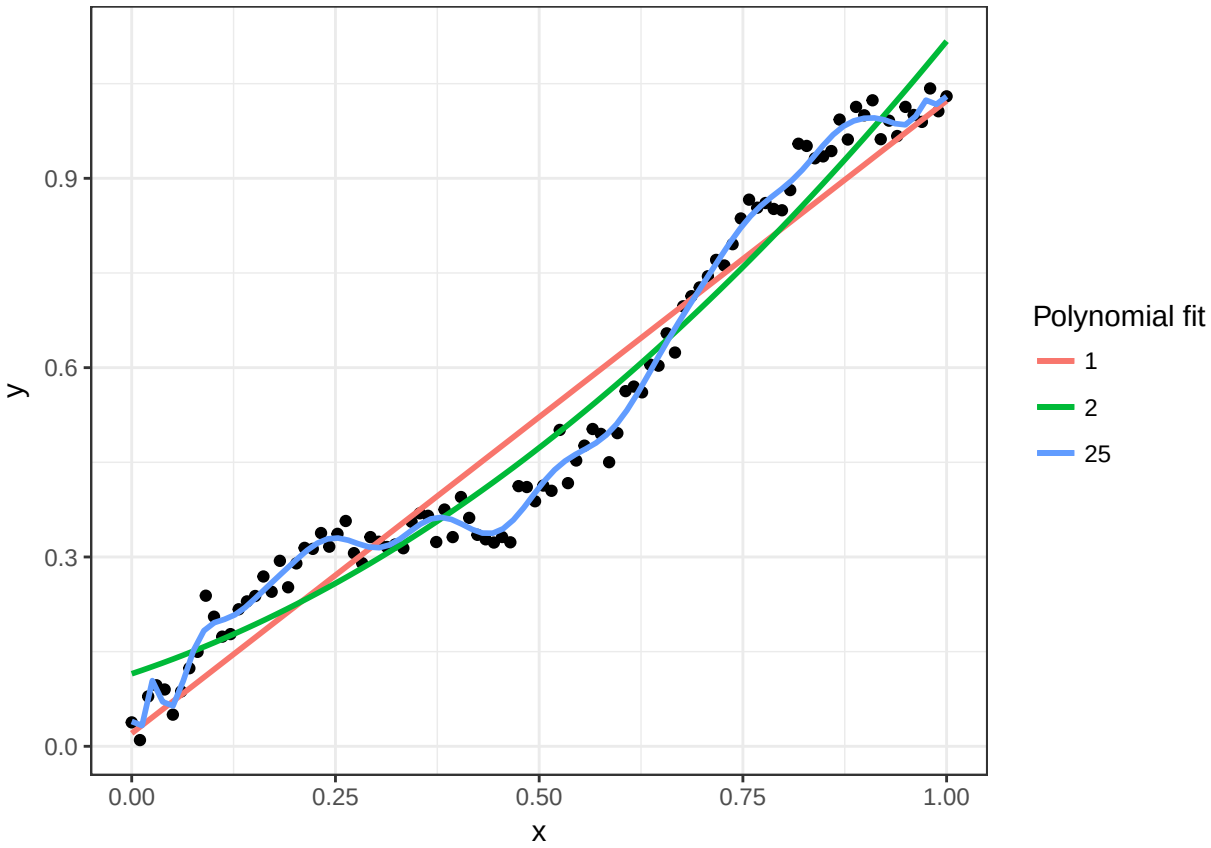
Regresja z karą (penalized regression)

Przypomnijmy sobie na czym polega budowanie regresji metodą największej wiarogodności. Minimalizowaliśmy logarytm z funkcji wiarogodności po parametrach β i σ^2.

$$l(\beta, \sigma^2|y) = -\frac{n}{2}\log\sigma^2 - \frac{n}{2}\log(2\pi) - \frac{1}{2\sigma^2}(Y - \mathbf{X}\beta)^T(Y - \mathbf{X}\beta)$$

Stąd $\hat{\beta} = (X^T X)^{-1} X^T y$ i $\hat{\sigma^2} = RSS(\hat{\beta})/n$.

Wracając do przykładu z dopasowaniem krzywej



Jak uniknąć nadmiernego dopasowania? Nakładając ograniczenia na współczynniki β.

$$\hat{\beta} = arg\min_{\beta} \underbrace{l(\beta|y)}_{likelihood} + \underbrace{\lambda pen(\beta)}_{\text{kara}}$$

W przypadku AIC i BIC nakładamy karę na liczbę niezerowych współczynników, czyli funkcja pen oparta jest o normę l_0.

$$BIC_M := l(\hat{\beta}) - \frac{\|\beta\|_0}{2} \log n$$

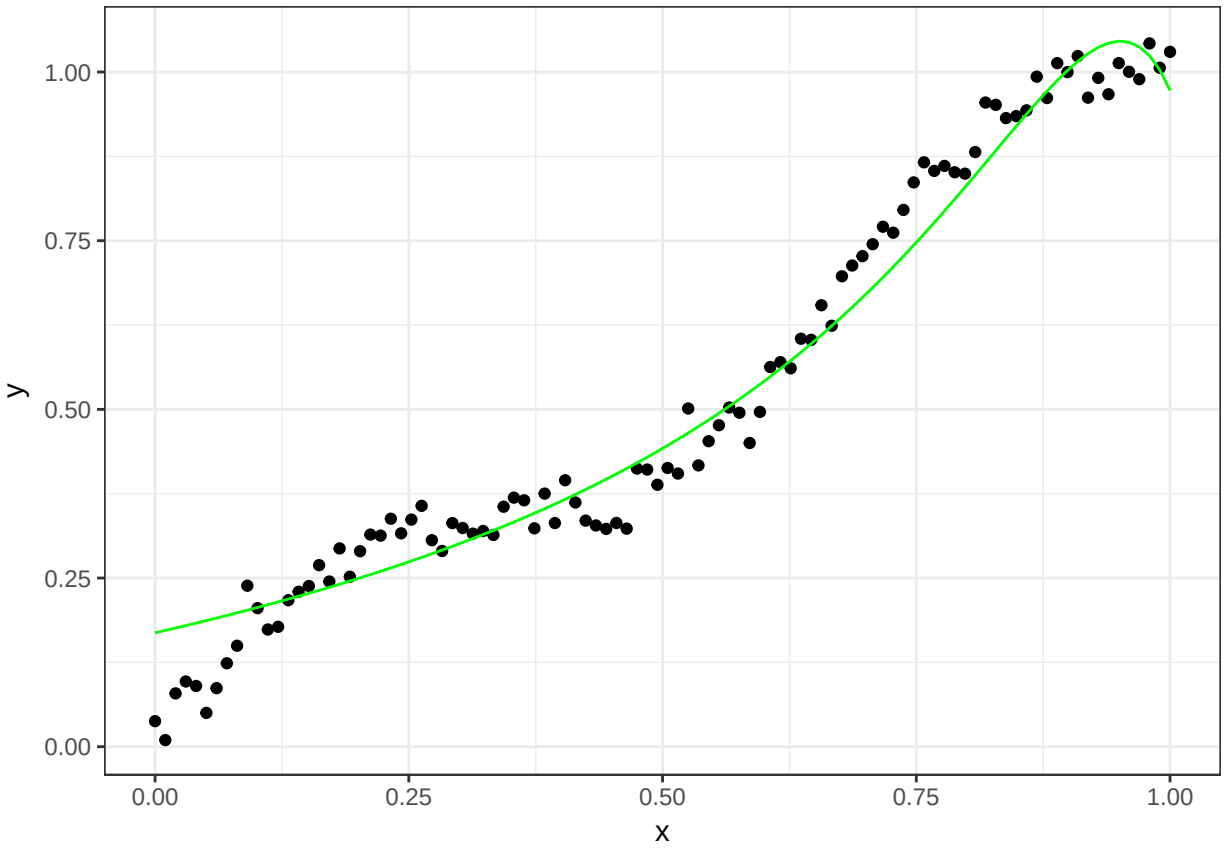
Jak łatwo się domyślić l_0 nie jest jedyną normą jaką możemy wykorzystać.

Ridge regression (regresja grzbietowa), Hoerl & Kennard, Technometrics, 1970

Powodem powstania tej metody są problemy numeryczne. Zauważmy, że gdy $n \approx p$, macierz $X^T X$ jest bliska nieodwracalnej. Jednak jeśli nieco zwiększy się przekątną dodając λI do $X^T X$, to macierz staje się odwracalna i ma większy wyznacznik. Okazuje się, że ma to dobre własności z punktu widzenia statystycznego. Kiedy macierz jest bliska osobliwej, wariancja estymatora β jest bardzo duża (wystarczy popatrzeć na wzór).

$$\hat{\beta}_{\text{ridge}} := arg\min_{\beta} l(\hat{\beta}) - \lambda\|\beta\|_2$$

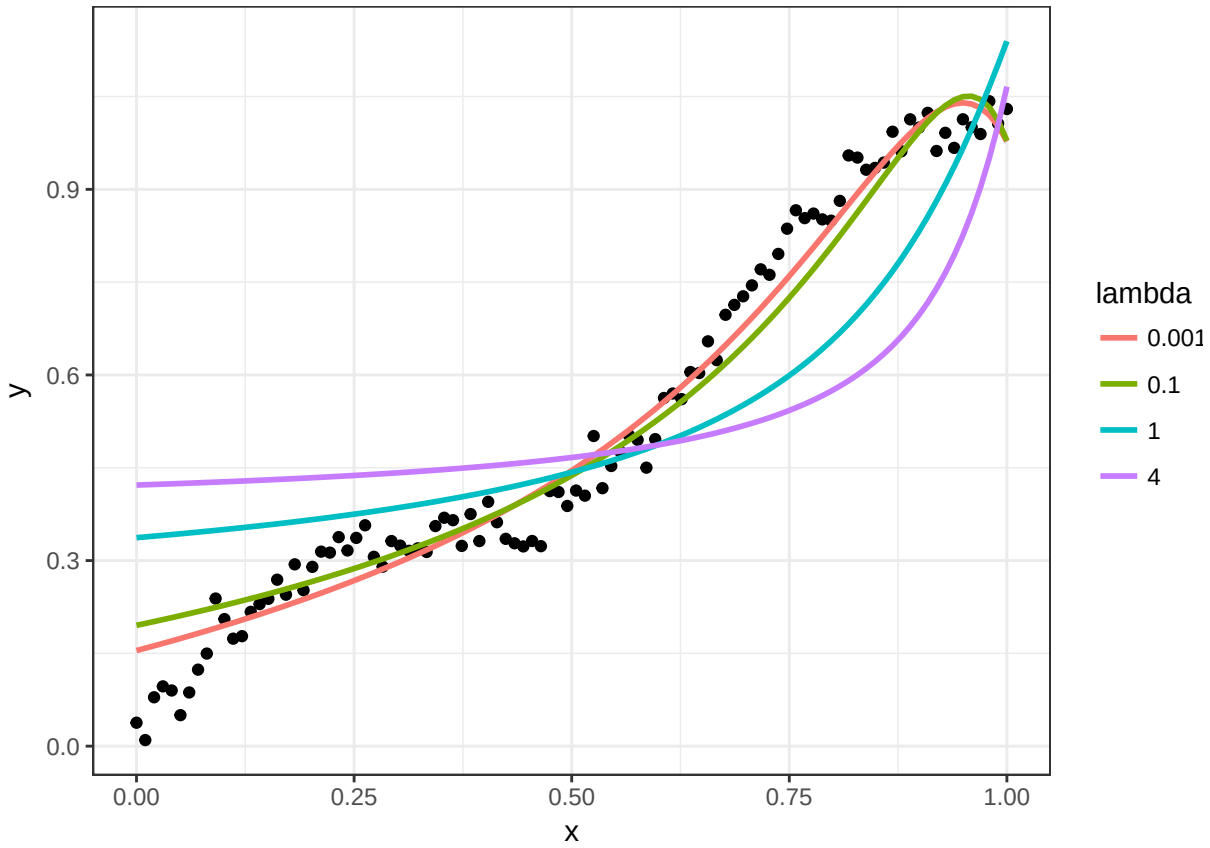
Co ciekawe te same estymatory dostaniemy jako MAP (maximum a posteriori) rozważając zwykłą regresję z normalnym rozkładem apriori dla parametrów β.



Ćwiczenia

1. Co się dzieje z $\hat{\beta}_{\text{ridge}}$ gdy $\lambda \to 0$?
2. Co się dzieje z $\hat{\beta}_{\text{ridge}}$ gdy $\lambda \to \infty$?

Osobnym problemem jest wybór parametru λ. Ten wybór ma bardzo poważne konsekwencje jeśli chodzi o jakość dopasowania! Jedną z możliwości jest wykorzystanie cross-validacji.



Regresja grzbietowa nie pozwala na wybór liczby zmiennych. Parametry β są bliższe zeru, ale pozostają niezerowe. Rozwiązaliśmy zatem problem nadmiernego dopasowania do danych, ale nie mamy możliwości wyboru zmiennych do modelu, jak to było w przypadku AIC i BIC.

Lasso

Tibshirani (Journal of the Royal Statistical Society 1996) LASSO: least absolute shrinkage and selection operator.

$$\hat{\beta}_{\text{lasso}} := arg \min_{\beta} l(\hat{\beta}) - \lambda \|\beta\|_1$$

Lub w innym zapisie:

$$\hat{\beta}_{\text{lasso}} := arg \min_{\beta} \sum_{i=1}^{n} (y_i - \beta_0 - \sum_j x_{i,j}\beta_j)^2 + \lambda \sum_j |\beta_j|$$

Lub jeszcze inaczej (przekształcone przez warunki Karusha-Kuhn-Tuckera KKT):

$$\hat{\beta}_{\text{lasso}} := arg \min_{\beta} \sum_{i=1}^{n} (y_i - \beta_0 - \sum_j x_{i,j}\beta_j)^2, \text{ pod warunkiem } \sum_j |\beta_j| \leq t$$

Mamy odpowiedniość 1-1 między t i λ, ale nie jest ona dana żadną prostą zależnością funkcyjną. Ostatnia formuła na estymator Lasso daje nam intuicję dotyczącą jego działania i różnicy z regresją grzbietową. Poniżej znajduje się rysunek, notabene jeden z najbardziej znanych i wpływowych w nowoczesnej statystyce,

pochodzący z książki Elements of Statistical Learning Hastiego, Tibshiraniego i Friedmana. Przedstawia dopasowanie regresji w przypadku gdy mamy dwa parametry β_1 i β_2. $\hat{\beta}$ to estymator NW. Czerwone kręgi to poziomice dla wartości log-likelihood. Im dalej jesteśmy od $\hat{\beta}$ tym likelihood jest mniejszy. Dla regresji ridge (z prawej strony) ograniczenie ma postać warunku na kulę w normie l_2, zaś dla lasso na kulę w normie l_1. Estymatorem jest przecięcie kuli w odpowiedniej normie z poziomicą odpowiadającą największej możliwej wartości log-likelihoodu. Dzięki temu, że norma l_1 jest „kanciasta“, współrzędne wektora β mogą się zerować.

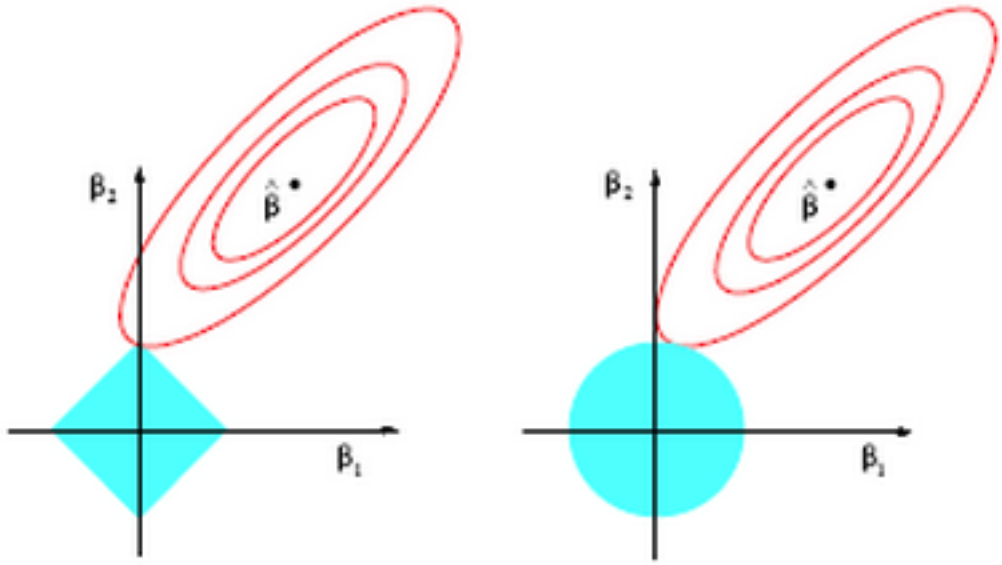
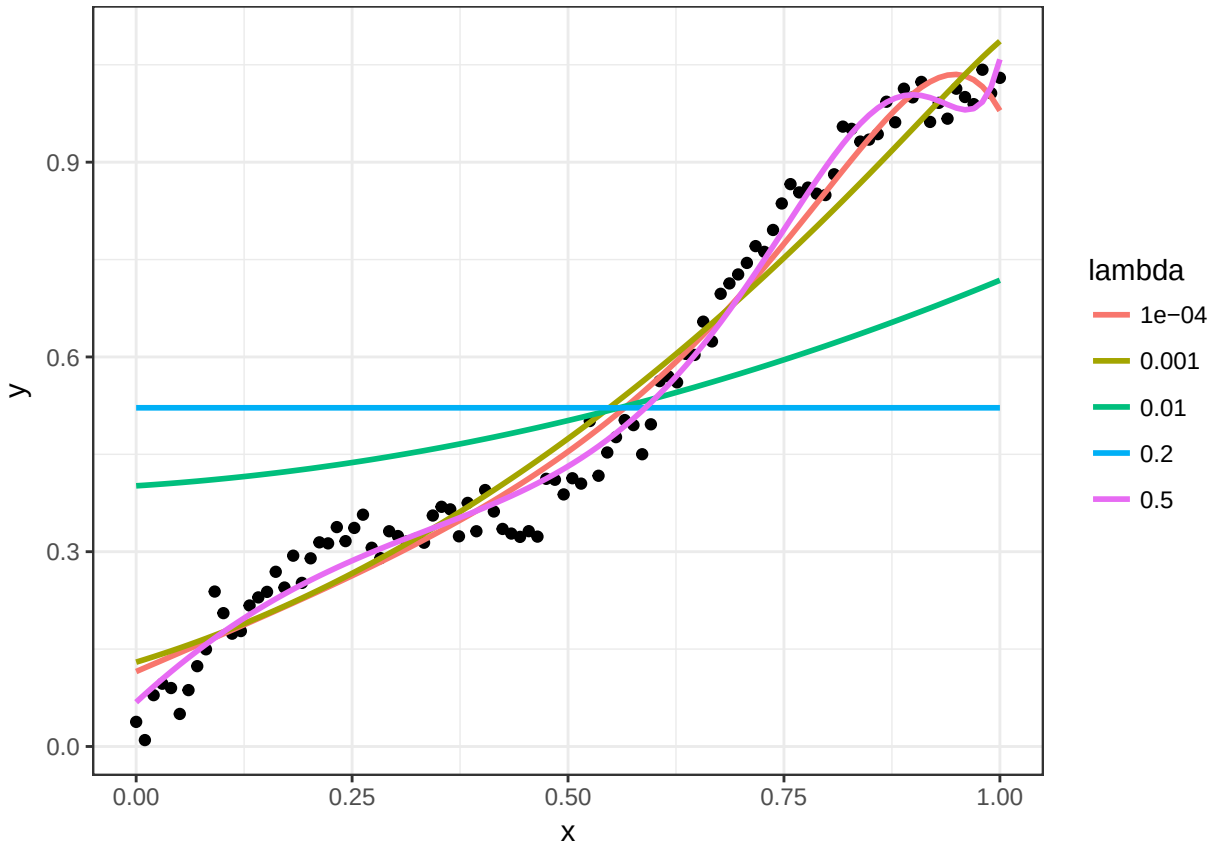


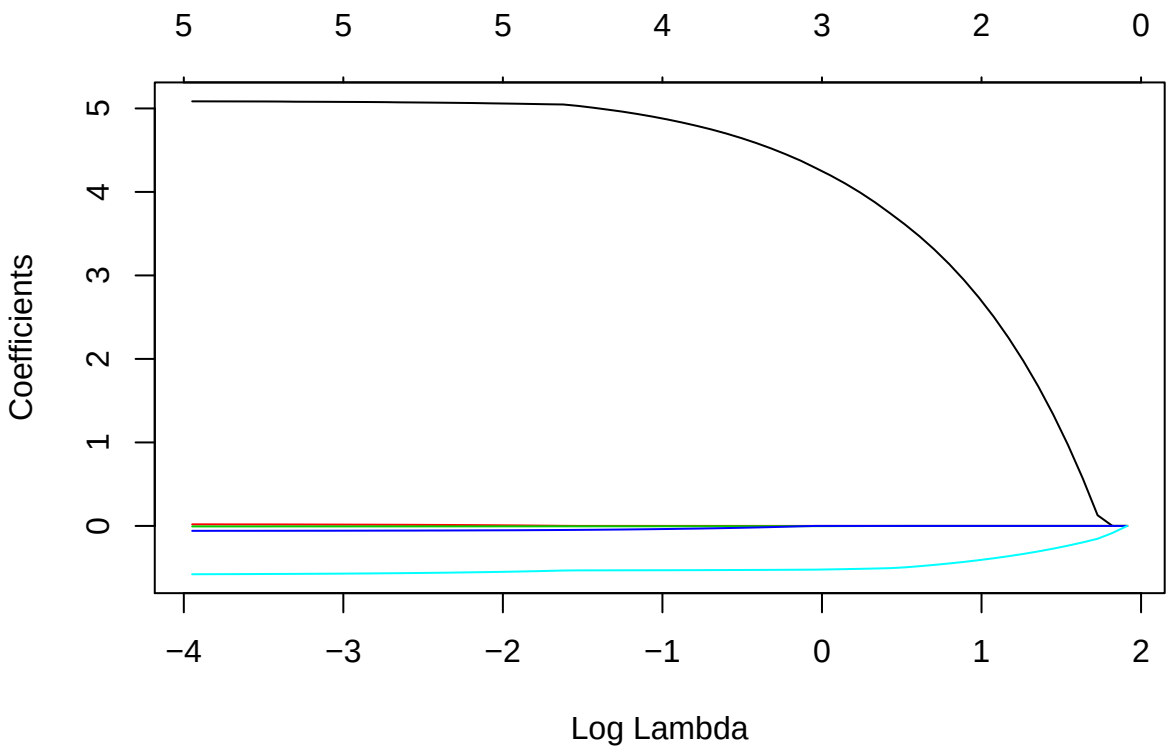
Figure 1:



Jak korzystać z Lasso w R?

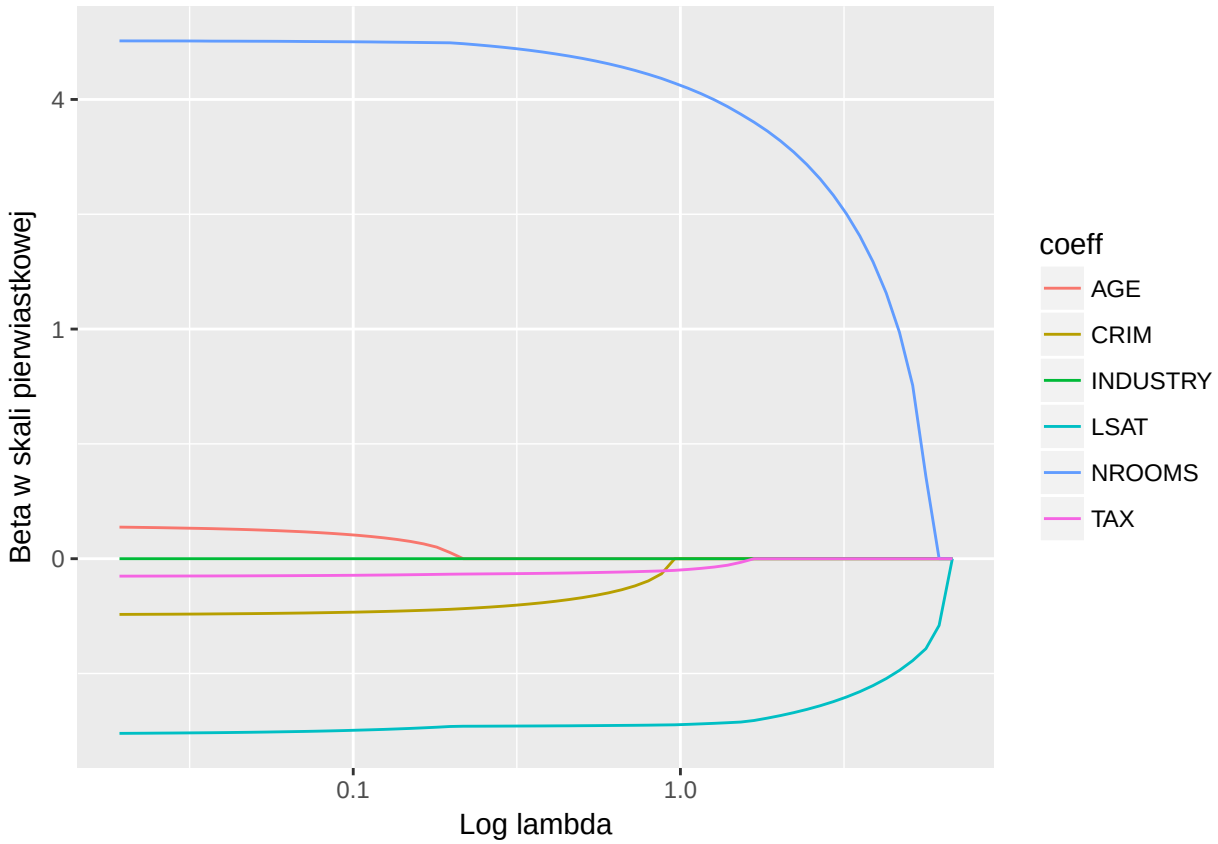
Najlepiej użyć biblioteki **glmnet**. Pozwala ona na dopasowanie penalizowanych regresji liniowej i logistycznej oraz modelu hazardu Coxa. Dzięki wykorzystaniu biblioteki do macierzy rzadkich jest rozsądnie szybka.

```
housing.lasso=glmnet(x = as.matrix(housing[,-ncol(housing)]), y = housing$MEDV, alpha = 1)
plot(housing.lasso, xvar = "lambda")
```



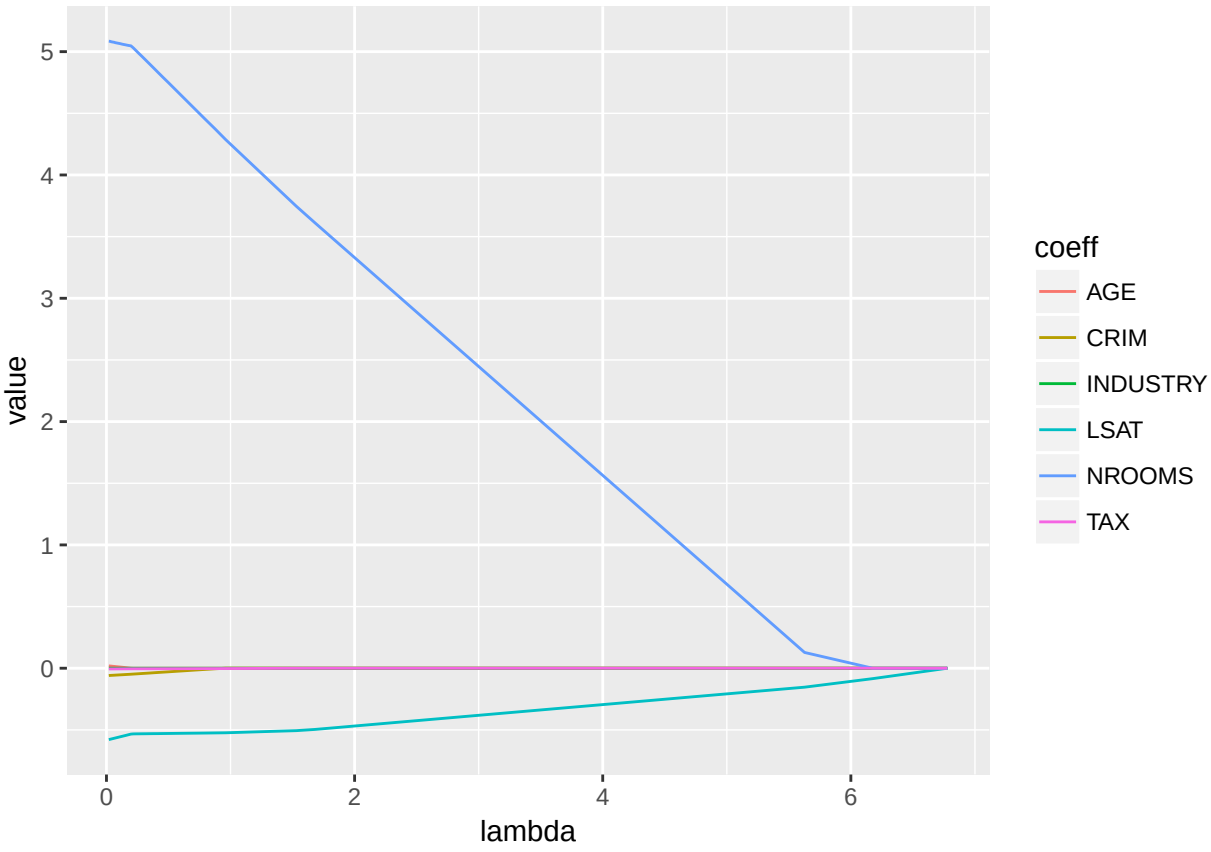
W tym przykładzie nie widać dobrze zmian współczynników, stwórzmy własny wykres w skali pierwiastkowej

```
betas=data.frame(lambda=housing.lasso$lambda, as.matrix(t(housing.lasso$beta))) %>%
  gather(coeff, value, -lambda)
ggplot(betas, aes(x=lambda, y=sign(value)*sqrt(abs(value)), group=coeff, color=coeff)) +
  geom_line() + scale_y_continuous("Beta w skali pierwiastkowej", labels=function(x) sign(x)*x^2) +
  scale_x_log10("Log lambda")
```



Powyższy wykres nazywamy ścieżkami lasso (lasso paths). Warto zauważyć, że w skali liniowej, ścieżki lasso są krzywymi łamanymi (piece-wise linear).

```
ggplot(betas, aes(x=lambda, y=value, group=coeff, color=coeff)) +
  geom_line()
```



Dlaczego warto korzystać z penalizowanej regresji?

Omówiliśmy kilka metod regresji penalizowanej (BIC, ridge, lasso). Powiedzieliśmy o własnościach, w szczególności odnośnie wyboru zmiennych (zerowanie współczynników β). Ale powód jest dużo bardziej konkretny. Chcielibyśmy żeby nasz estymator $\hat{\beta}$ był blisko prawdziwego β_{true}

$$MSE(\hat{\beta}) = E\|\hat{\beta} - \beta_{true}\|^2 = E\|\hat{\beta} - E\hat{\beta}\|^2 + (E\|\hat{\beta} - \beta_{true}\|)^2 = Var(\hat{\beta}) + bias(\hat{\beta})$$

Estymator NW w modelu liniowym jest zawsze nieobciążony. Niestety opłacamy to czasami bardzo wysoką wariancją (pamiętamy, że wariancja zależy od macierzy $X^T X$). Czasem opłaca się popełnić systematycznym błąd (estymator jest obciążony), a w zamian dostać mniejsze MSE.

Jednym z celów budowy modelu liniowego jest predykcja. Chcemy dobrze przewidywać wartości nowych obserwacji. Podobnie rzecz ma się również w tym przypadku.

$$PE(x_o) = E\{(Y - \hat{f}(X))^2 | X = x_0\} = \sigma^2 + Bias^2(\hat{f}(x_0)) + Var(\hat{f}(x_0))$$

Tak jak poprzednio, nawet jeśli mamy estymator nieobiążony, to nasz błąd predykcji może być duży ze względu na dużą wariancję.

Referencje:

1. http://www.stat.cmu.edu/~larry/=stat705/Lecture16.pdf
2. https://lagunita.stanford.edu/c4x/HumanitiesScience/StatLearning/asset/model_selection.pdf
3. http://web.stanford.edu/~hskang/stat431/ModelSelection.pdf
4. https://www.stat.washington.edu/raftery/Research/PDF/kass1995.pdf

5. http://statweb.stanford.edu/~tibs/ElemStatLearn/
6. http://www.stat.umn.edu/geyer/5931/mle/sel.pdf