Uniwersytet Wrocławski
Wydział Matematyki i Informatyki
Instytut Matematyczny
specjalność: analiza danych

Wybór optymalnego podzbioru w analizie dużych zbiorów danych

Aleksander Milach

Praca magisterska
napisana pod kierunkiem
profesor Małgorzaty Bogdan

Wrocław 2021

Spis treści

Wstęp i streszczenie

Rozważmy model liniowy

$$\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \varepsilon,$$

gdzie $\mathbf{y}$ to wartości zmiennej objaśnianej, macierz $\mathbf{X}$ jest znana, ε to nieznany wektor szumu, a β to nieznany wektor współczynników, który chcemy estymować. W wielu istotnych zastosowaniach statystyki będziemy poszukiwali modelu najlepiej opisującego $\mathbf{y}$ przy pomocy co najwyżej k z p zmiennych objaśniających. Niewielka rzadkość wektora współczynników prowadzi do modelu, który łatwiej interpretować i stosować, ma ona szczególne znaczenie w sytuacjach, gdy liczba predyktorów przywyższa liczbę obserwacji n. Rozważane zagadnienie prowadzi do problemu optymalnego podzbioru, danego następująco:

$$\min_{\boldsymbol{\beta}} \frac{1}{2}||\mathbf{y} - \mathbf{X}\boldsymbol{\beta}||_2^2 \text{ takie, że } ||\boldsymbol{\beta}||_0 \leq k. \tag{0.1}$$

Ogranicznie na ilość niezerowych współrzędnych wektora $\boldsymbol{\beta}$ czyni powyższy problem NP-trudnym. Z tego powodu przez wiele lat uznawano, że w rozwiązaniu problemu optymalnego podzbioru w powyższej postaci nie można poczynić postępów.

Rozważany problem można sprowadzić do postaci ogólniejszego problemu mieszanej optymalizacji kwadratowej (MIO), jednak ze względu na trudność i długi czas wykonania obliczeń nie było to przydatne. W ciągu ostatnich 25 lat moc obliczeniowa oprogramowania rozwiązującego MIO znacząco wzrosła. Porównania ze sobą szybkości rozwiązania tych samych problemów przez kolejne wersje programów prowadzą do różnicy na poziomie 780 tysięcy razy, a jest to przyspieszenie niezależne od czasu wykonywania obliczeń - wykorzystujące głównie wiedzą teoretyczną. W połączeniu z rosnącą szybkością wykonywania obliczeń przez superkomputery szacowaną na 570 tysięcy razy w ostatnich 20 latach otrzymamy, że w tym okresie skumulowane przyspieszenie wynosi 450 bilionów razy.

Dotychczasowe przeświadczenie, że MIO jest tylko teoretycznie przydatne w rozwiązywaniu problemów statystyki wykształciło się w latach 80-tych było wówczas uzasadnione, jednak nadzwyczajny rozwój programów analizujących MIO każe poddawać to pod wątpliwość. Następujące rozważania są przykładem ich użyteczności w kontekście zagadnienia wyboru optymalnego podzbioru.

Praca poświęcona jest nowoczesnemu podejściu do problemu optymalnego podzbioru przy pomocy mieszanej optymalizacji kwadratowej oraz metod optymalizacyjnych pierwszego rzędu. W pierwszym rozdziale przedstawimy podejście do problemu optymalnego podzbioru przy pomocy kryteriów informacyjnych, przedstawimy ich

własności oraz podamy ich istotne wyniki. W rozdziale drugim przedstawimy sformułowania problemu optymalnego podzbioru jako MIO oraz wprowadzimy pojęcia konieczne do oszacowania parametrów sformułowań. W dalszej kolejności korzystając z metod optymalizacji wypukłej podamy algorytm pierwszego rzędu prowadzący do rozwiązania problemu powiązanego z zagadnieniem optymalnego podzbioru oraz zbadamy jego własności. Porównaniem rozważanych wcześniej metod w symulacjach oraz analizie rzeczywistego zbioru danych zajmujemy się w rozdziale czwartym.

1 Kryteria informacyjne

1.1 Geneza kryteriów informacyjnych [2]

Standardowym podejściem do problemu regresji liniowej jest metoda najmniejszych kwadratów, która w tym przypadku daje ten sam wynik, co estymacja metodą największej wiarogodności. Prowadzi ona do wzoru

$$\hat{\boldsymbol{\beta}} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y}.$$

Ten znany estymator ma wiele własności, w szczególności jest on nieobciążony i ma rozkład normalny $\hat{\boldsymbol{\beta}} \sim N(\boldsymbol{\beta}, \sigma^2(\mathbf{X}^T\mathbf{X})^{-1})$.

Takie podejście wystarcza w przypadku, gdy liczba zmiennych objaśniających p jest dużo mniejsza od liczby obserwacji n. W obecnie dostępnych zbiorach danych możemy mieć do czynienia z przypadkami, gdzie p jest porównywalne z n, wówczas metoda najmniejszych kwadratów ma spore wady i jest właściwie niemożliwa do użytku.

Zauważmy najpierw, że gdy p zbliża się do n, rośnie wariancja estymatora. Za jej wartość będą odpowiadały wartości na przekątnej macierzy $(\mathbf{X}^T\mathbf{X})^{-1}$, zatem w przypadku, gdy wartości na przekątnej macierzy $\mathbf{X}^T\mathbf{X}$ będa bliskie zeru, wariancja estymatora może bardzo wzrosnąć. Skutkować to będzie spadkiem mocy testu na istotność współczynnika dla danej zmiennej, nawet do ustalonej wartości prawdopodobieństwa błędu pierwszego rodzaju - nasz model będzie wykrywał bardzo mało zmiennych i wiele z nich będzie fałszywymi odkryciami, czyli statystycznie nieznaczącymi zmiennymi uznanymi przez nasz model za istotne.

Kolejny problem zauważamy w funkcji straty. Minimalizacja residualnej sumy kwadratów $RSS = ||\mathbf{y} - \mathbf{X}\hat{\boldsymbol{\beta}}||^2$, nie radzi sobie z dużą ilością zmiennych objaśniających. Wśród wszystkich modeli wolelibyśmy te, które zawierają mniej zmiennych. Jednak dla kwadratowej funkcji straty dodanie dowolnej zmiennej do modelu, poza rzadkimi przypadkami, gdy nowa zmienna jest linowo zależna od pozostałych lub jest

ortogonalna do wektora $\mathbf{y}$, spowoduje zmniejszenie wartości RSS, a zatem według reguły poprawę modelu. Nie jesteśmy w stanie określić, jak duży musi być spadek RSS, aby były podstawy do uznania zmiennej za istotną.

Pewnym rozwiązaniem powyższych problemów są kryteria informacyjne. Stosujemy je w następujący sposób. Dla ustalonego zbioru modeli obliczamy wartość kryterium, poczym wybieramy model, dla którego wartość kryterium jest najmniejsza. Dwa najstarsze i najbardziej znane kryteria to AIC i BIC.

Kryterium informacyjne Akaikego (AIC) w swoim uzasadnieniu wykorzystuje metody teorii informacji prowadzi do wzoru

$$AIC(\hat{\boldsymbol{\beta}}) = -2\log(L(\hat{\boldsymbol{\beta}})) + 2k,$$

gdzie k to ilość niezerowych współrzędnych wektora $\hat{\boldsymbol{\beta}}$.

BIC, czyli bayesowskie kryterium informacyjne Schwarza korzysta z estymacji bayesowskiej i wyraża się wzorem

$$BIC(\hat{\boldsymbol{\beta}}) = -2\log(L(\hat{\boldsymbol{\beta}})) + k\log(n).$$

Wzory obu kryteriów są podobne i nie jest to przypadek. Większość kryteriów informacyjnych można przedstwić wzorem

$$-2\log(L(\hat{\boldsymbol{\beta}})) + Pen(\hat{\boldsymbol{\beta}}).$$

Pierwszy składnik - logarytm funkcji wiarogodności dla estymatora - odpowiada za jakość estymacji, natomiast drugi ma za zadanie ograniczać ilość zmiennych w modelu.

1.2 Wady kryteriów AIC i BIC

Przypuśćmy, że macierz planu $\mathbf{X}$ jest ortogonalna i przeskalowana tak, że $\mathbf{X}^T\mathbf{X} = I_{p\times p}$. Pomimo tego, że szanse, że w rzeczywistych danych do tego dojdzie są bardzo niewielkie, na tym prostym przykładzie najłatwiej pokazać wady AIC i BIC, które występują w ogólniejszych przypadkach. Dzięki ortogonalności zmienne objaśniające są od siebie niezależne, a wzór na wartość współczynnika odpowiadającego zmiennej obliczonego przy pomocy metody najmniejszych kwadratów sprowadza się do $\hat{\beta}_j = \mathbf{X}_j^T\mathbf{Y}$, gdzie $\mathbf{X}_j$ jest kolumną macierzy $\mathbf{X}$. Wówczas powyższe estymatory mają rozkład normalny $N(\beta_j,\, \sigma^2)$, a hipotezy $\beta_j = 0$ przeciwko $\beta_j \neq 0$ testujemy przy pomocy statystyki $Z_j = \sqrt{n}\hat{\beta}_j/\sigma$, która przy hipotezie ma rozkład standardowy normalny.

W tej sytuacji składnik log-wiarogodności ma postać:

$$-2\log(L(\hat{\boldsymbol{\beta}})) = c + ||\mathbf{y} - \sum_{j=1}^{p} \mathbf{X}_j \hat{\beta}_j||^2/\sigma^2 = c + (\mathbf{y} - \mathbf{X}\hat{\boldsymbol{\beta}})^T(\mathbf{y} - \mathbf{X}\hat{\boldsymbol{\beta}})/\sigma^2 =$$

$$= c + (\mathbf{y}^T\mathbf{y} + \hat{\boldsymbol{\beta}}^T\mathbf{X}^T\mathbf{X}\hat{\boldsymbol{\beta}} - 2\mathbf{y}^T\mathbf{X}\hat{\boldsymbol{\beta}})/\sigma^2 = c + (\mathbf{y}^T\mathbf{y} + \hat{\boldsymbol{\beta}}^T\hat{\boldsymbol{\beta}} - 2\hat{\boldsymbol{\beta}}^T\hat{\boldsymbol{\beta}})/\sigma^2 =$$

$$= c + (\mathbf{y}^T\mathbf{y} - \hat{\boldsymbol{\beta}}^T\hat{\boldsymbol{\beta}})/\sigma^2 = c + (\mathbf{y}^T\mathbf{y} - ||\hat{\boldsymbol{\beta}}||^2)/\sigma^2,$$

zatem dodanie do modelu zmiennej $\mathbf{X}_j$ zmniejsza składnik log-wiarogodności o $\hat{\beta}_j^2/\sigma^2$, a zwiększa karę o 2. Korzystając z rozkładu estymatora mamy, że wartość kryterium AIC zmaleje, gdy jego wartość bezwzględna będzie większa niż $\sqrt{2}\sigma$. Oznacza to, że w kryterium prawdopodobieństwo błędu pierwszego rodzaju wynosi $2(1 - \Phi(\sqrt{2})) = 0,157$. Dodanie zmiennej do modelu odbywa się niezależnie od tego, ile zmiennych już zostało do niego wybranych, a ilość fałszywych odkryć rośnie liniowo z p.

Podobnie przy ortogonalności macierzy planu zachowuje się BIC, tutaj jednak kara zależy od wielkości próby n. Z analogicznych rozważań otrzymamy, że wartość kryterium zmaleje, gdy wartość bezwzględna estymatora będzie większa niż $\sqrt{\log(n)}\,\sigma$. Prowadzi to do prawdopodobieństwa błędu pierwszego rodzaju na poziomie $2(1 - \Phi(\sqrt{\log(n)}))$. Przykładowo dla $n = 100$ jest to około $0,032$, o wiele mniej niż w przypadku AIC, jednak nadal nie rozwiązuje to naszych problemów - co trzydziesta fałszywa zmienna zostaje uznana za istotną. Mimo tego kryterium BIC ma pewną przewagę, jest zgodne, to jest, przy $n \to \infty$ prawdopodobieństwo błędu maleje do zera.

Kolejne trudności pojawiają się, gdy wartość p jest duża. O ile powyższe kryteria sprawdzają się w przypadku, gdy chcemy porównać ze sobą małą liczbę modeli i wybrać z nich najlepszy, to w zupełnej ogólności, w sytuacji gdy mamy p zmiennych możemy z nich wybrać 2^p modeli, ponieważ każdą ze zmiennych możemy włączyć do modelu bądź nie. W związku z tym ilość modeli bardzo szybko rośnie wraz z p i dla $p > 50$ obliczenie wartośći kryterium dla każdego z nich jest właściwie niemożliwe, przez czas wykonania odpowiednich obliczeń.

1.3 Wprowadzenie mBIC i mBIC2

W związku w wadami standardowych metod zaszła potrzeba znalezienia nowych kryteriów o użytecznych własnościach w przypadkach, gdy $n \sim p$. Pierwszym rozwiązaniem było kryterium RIC postaci

$$RIC(\hat{\boldsymbol{\beta}}) = -2\log(L(\hat{\boldsymbol{\beta}})) + 2k\log(p)$$

Sprawdzając, podobnie jak dla powyższych kryteriów, prawdopodobieństwo błędu pierwszego rodzaju w sytuacji ortogonalnej dochodzimy do zależności $|\hat{\beta}_j| > \sigma\sqrt{2\log p}$. Dla rosnącego p wartość bezwzględna estymatora musi być coraz większa, aby zmienna została włączona do modelu, równoważnie prawdopodobieństwo błędu pierwszego rodzaju maleje. Oznacza to, że frakcja fałszywych odkryć również maleje, jednak bardzo wolno ze względu na pojawiający się czynnik $\log p$. W praktyce dla $p = 10$ FWER, czyli prawdopodobieństwo uzyskania niezerowej liczby fałszywych odkryć wynosi 0,35, a dla $p = 1000$ nadal około 0,2.

Użyteczna w okazała się dopiero modyfikacja BIC wprowadzona w [3]. Wyprowadzenie kryterium BIC zakłada, że wszystkie modele są tak samo prawdopodobne. Zauważmy jednak, że modeli rzadkości 1 jest p, rzadkości 2 - $\binom{p}{2}$, ale rzadkości $p/2$ już $\binom{p}{p/2}$. Oznacza to, że BIC będzie często wybierać modele o rzadkości bliskiej $p/2$, w konsekwencji często będzie przeszacowywało rzadkość modelu. Wprowadzenie dwumianowego rozkładu oczekiwanej liczby zmiennych istotnych prowadzi do kryterium postaci

$$mBIC(\hat{\boldsymbol{\beta}}) = -2\log(L(\hat{\boldsymbol{\beta}})) + k\log n + 2k\log(p/E)$$

mBIC łączy kary BIC i RIC. Dzięki temu dla dużych wartości n czynnik z kryterium BIC będzie odpowiadał za minimalizowanie prawdopodobieństwa błędu pierwszego rodzaju, zaś dla dużego p będzie to kara z kryterium RIC. Stała E jest oczekiwaną liczbą istotnych zmiennych w naszym modelu - wartością oczekiwaną dla dwumianowego rozkładu a priori. W przypadku, gdy nie zakładamy żadnej konkretnej ilości istotnych zmiennych należy przyjąć $E = 4$. Już dla tego ogólnego przypadku kryterium osiąga lepsze wyniki w symulacjach. Dla $n = 150$ już dla $p \geq 10$ FWER jest kontrolowane na poziomie 0,1, dla $p = 1000$ otrzymujemy ograniczenie na poziomie 0,065, natomiast dla $n = 500$ FWER jest mniejsze niż 0,05 dla $p \geq 10$ i mniejsze niż 0,035 dla $p = 1000$.

Powyższe kryterium można zmodyfikować, aby osiągało minimalne wartości frakcji fałszywych odkryć, nie zaś FWER w przypadku mBIC. Kryterium realizującym tę własność jest mBIC2.

$$mBIC2(\hat{\boldsymbol{\beta}}) = -2\log(L(\hat{\boldsymbol{\beta}})) + k\log n + 2k\log(p/E) - 2\log k!$$

Zauważmy, że w tym przypadku zmniejszamy karę w stosunku do mBIC, właśnie ze względu na osiągnięcie jak najlepszej kontroli FDR. Działanie to znajduje uzasadnienie teoretyczne w związku z korektą Benjaminiego - Hochberga na wielokrotne testowanie. Kryterium mBIC2 osiąga kontrolę FDR na poziomie zależnym od n - $FDR_n \sim (n\log n)^{-1/2}$. Zatem dla $n = 100$ kontroluje FDR na poziomie około 0,1, dla $n = 1000$ na poziomie 0,031.

2 Mieszana całkowitoliczbowa optymalizacja kwadratowa [1]

Mieszana optymalizacja kwadratowa (ang. mixed quadratic optimization - MIO) jest zbiorem problemów opisanych następującym wzorem:

$$\min \ \alpha^T \mathbf{Q}\alpha + \alpha^T a$$

$$\text{takie, że} \quad A\alpha \leq b$$

$$\alpha_i \in \{0,1\}, \quad i \in \mathcal{I},$$

$$\alpha_j \geq 0, \quad j \notin \mathcal{I},$$

gdzie $a \in \mathbb{R}^m,\ A \in \mathbb{R}^{k\times m},\ b \in \mathbb{R}^k$ oraz $\mathbf{Q} \in \mathbb{R}^{m\times m}$ jest nieujemnie określona są danymi parametrami. Wektor $\alpha \in \mathbb{R}^m$ zawiera wartości dyskretne dla $\alpha_i,\ i \in \mathcal{I}$ i ciągłe dla $\alpha \notin \mathcal{I}$, przy $\mathcal{I} \subset \{1,...,m\}$. Niektórymi podproblamami MIO są problemy kwadratowej optymalizacji wypukłej ($\mathcal{I} = \emptyset$), mieszane całkowitoliczbowe i liniowe problemy optymalizacyjne ($\mathbf{Q} = 0_{m\times m}$). Przykładowe zaawansowane programy analizujące problemy tej postaci to CPLEX, Gurobi i Xpress.

2.1 Sposoby wyrażenia problemu optymalnego podzbioru przez MIO

Proste sformułowanie problemu (0.1) może być następujące:

$$Z_1 = \min_{\boldsymbol{\beta},\mathbf{z}} \frac{1}{2}||\mathbf{y} - \mathbf{X}\boldsymbol{\beta}||_2^2 \tag{2.1}$$

$$\text{takie, że:} \quad -\mathcal{M}_U z_i \leq \beta_i \leq \mathcal{M}_U z_i, \quad i = 1,...,p,$$

$$z_i \in \{0,1\}$$

$$\sum_{i=1}^{p} z_i \leq k,$$

gdzie $\mathbf{z} \in \{0,1\}^p$ oraz $\mathcal{M}_U$ jest stałą taką, że rozwiązanie $\hat{\boldsymbol{\beta}}$ spełnia $||\hat{\boldsymbol{\beta}}||_\infty \leq \mathcal{M}_U$. Jeśli $z_i = 1$, to $|\beta_i| \leq \mathcal{M}_U$, jeśli $z_i = 0$, to $\beta_i = 0$, skąd $\sum_{i=1}^{p} z_i$ jest ilością niezerowych współrzędnych w wektorze $\hat{\boldsymbol{\beta}}$.

Powyższe sformułowanie jest powiązane z problemem LASSO. Rozważmy uwypuklenie zbioru rozwiązań z ich ograniczeniami

$$\text{Conv}\left(\left\{\boldsymbol{\beta} : |\beta_i| \leq \mathcal{M}_U z_i, z_i \in \{0,1\},\ i = 1, ..., p,\ \sum_{i=1}^{p} z_i \leq k\right\}\right)$$

$$= \{\boldsymbol{\beta} : ||\boldsymbol{\beta}||_\infty \leq \mathcal{M}_U,\ ||\boldsymbol{\beta}||_1 \leq k\mathcal{M}_U\} \subseteq \{\boldsymbol{\beta} : ||\boldsymbol{\beta}||_1 \leq k\mathcal{M}_U\}.$$

Zatem rozwiązanie problemu (2.1) jest ograniczone z dołu przez rozwiązania następujących problemów optymalizacji wypukłej:

$$Z_2 = \min_{\boldsymbol{\beta}} \frac{1}{2}||\mathbf{y} - \mathbf{X}\boldsymbol{\beta}||_2^2 \text{ takie, że } ||\boldsymbol{\beta}||_\infty \leq \mathcal{M}_U,\ ||\boldsymbol{\beta}||_1 \leq k\mathcal{M}_U, \tag{2.2}$$

$$Z_3 = \min_{\boldsymbol{\beta}} \frac{1}{2}||\mathbf{y} - \mathbf{X}\boldsymbol{\beta}||_2^2 \text{ takie, że } ||\boldsymbol{\beta}||_1 \leq k\mathcal{M}_U, \tag{2.3}$$

gdzie (2.3) jest dokładnie problemem LASSO. Jest to uogólnienie problemu (2.2), w którym dodatkowo kontrolujemy poszczególne wartości β_i. W związku z tym zachodzą następujące nierówności $Z_3 \leq Z_2 \leq Z_1$, gdzie w praktyce są to nierówności ostre. Okazuje się to istotne w kontekście szacowania rozwiązania przez MIO, ponieważ programy najpierw wykorzystują ciągłe uogólnienie (2.1). Powyższe sformułowanie LASSO jest słabsze od tego uogólnienia.

Będziemy rozważali również następujące sformułowanie:

$$Z_4 = \min_{\boldsymbol{\beta}} \frac{1}{2}||\mathbf{y} - \mathbf{X}\boldsymbol{\beta}||_2^2 \tag{2.4}$$

$$\begin{gathered}\text{takie, że: } ||\boldsymbol{\beta}||_0 \leq k,\\ ||\boldsymbol{\beta}||_\infty \leq \mathcal{M}_U, \quad ||\boldsymbol{\beta}||_1 \leq \mathcal{M}_\ell,\\ ||\mathbf{X}\boldsymbol{\beta}||_\infty \leq \mathcal{M}_U^\xi, \quad ||\mathbf{X}\boldsymbol{\beta}||_1 \leq \mathcal{M}_\ell^\xi.\end{gathered}$$

Ograniczenia na $\boldsymbol{\beta}$ oraz $\mathbf{X}\boldsymbol{\beta}$ nie są konieczne, ale poprawiają sformułowanie problemu jako MIO. Wyrażenia z silniejszymi ograniczeniami prowadzą do szybszego osiągnięcia lepszych wyników, niż wyrażenia ze słabszymi.

Dla dostatecznie dużej wartości stałej $\mathcal{M}_U$ dla (2.1) lub stałych $\mathcal{M}_U, \mathcal{M}_\ell, \mathcal{M}_U^\xi, \mathcal{M}_\ell^\xi$, dla (2.4) rozwiązanie problemu będzie rozwiązaniem wyjściowego problemu. Kiedy ich wartość będzie zbyt mała, możemy otrzymać rozwiązanie inne niż dla (0.1). Pokażemy teraz jak obliczyć ich oszacowania z analizowanych danych.

2.2 Specyfikacja parametrów przez spójność i ograniczone wartości własne

Następująca metoda jest wiąże parametry problemu (2.4) ze spójnością i pojęciami pokrewnymi. Niech $\mathbf{X}$ będzie ustaloną macierzą planu. [7] definiuje funkcję kumulowanej spójności

$$\mu[k] = \max_{|I|=k} \max_{j \notin I} \sum_{i \in I} |\langle \mathbf{X}_j, \mathbf{X}_i \rangle|, \tag{2.5}$$

gdzie $\mathbf{X}_j, j = 1, ..., p$ są kolumnami macierzy $\mathbf{X}$. W szczególności dla $k = 1$ otrzymujemy spójność w rozumieniu [4] jako miarą maksymalnej korelacji między kolumnami macierzy $\mathbf{X}$: $\mu = \mu[1] = \max_{i \neq j} |\langle \mathbf{X}_j, \mathbf{X}_i \rangle|$. Skorzystamy również z warunku ograniczonych wartości własnych tj.

$$\lambda_{min}(\mathbf{X}_I^T \mathbf{X}_I) \geq \gamma_k \text{ dla dowolnego } I \subset \{1, ..., p\} \text{ takiego, że } |I| \leq k, \tag{2.6}$$

gdzie $\lambda_{min}(\mathbf{X}_I^T \mathbf{X}_I)$ jest najmniejszą wartością własną macierzy $\mathbf{X}_I^T \mathbf{X}_I$. Następujące nierówności łączą wprowadzone pojęcia.

Fakt 2.1. *[7] Mamy następujące ograniczenia:*
(a) $\mu[k] \leq \mu k$,
(b) $\gamma_k \geq 1 - \mu[k-1] \geq 1 - \mu(k-1)$.

Obliczenie $\mu[k]$ lub γ_k wymaga wielu pomocniczych obliczeń, dużo szybciej możemy obliczyć μ. Powyższy fakt podaje ograniczenia na $\mu[k]$ oraz γ_k wyrażone przez μ.

Użyteczne będzie również pojęcie normy operatorowej macierzy. Dla ustalonych (p, q) normę operatorową macierzy $\mathbf{A}$ definiujemy jako $||\mathbf{A}|| := \max_{||\mathbf{u}||_q=1} ||\mathbf{Au}||_p$. Będziemy używali jedynie $(1, 1)$ normy operatorowej.

Fakt 2.2. *Niech kolumny macierzy* $\mathbf{X}$ *mają normę* ℓ_2 *równą 1. Dla dowolnego* $I \subset \{1, ..., p\}$*, gdzie* $|I| = k$*, mamy:*
(a) $||\mathbf{X}_I^T \mathbf{X}_I - \mathbf{I}||_{1,1} \leq \mu[k-1]$,
(b) Jeśli $\mathbf{X}_I^T \mathbf{X}_I$ *jest odwracalna oraz* $||\mathbf{X}_I^T \mathbf{X}_I - \mathbf{I}||_{1,1} \leq 1$*, to* $||(\mathbf{X}_I^T \mathbf{X}_I)^{-1}||_{1,1} \leq \frac{1}{1-\mu[k-1]}$.

Dowód. (a) Dla ustalonego I, niech $\mathbf{G} := \mathbf{X}_I^T \mathbf{X}_I - \mathbf{I}$. Dla dowolnego $\mathbf{u} \in \mathbb{R}^k$ mamy:

$$\max_{||\mathbf{u}||_1=1} ||\mathbf{G}\mathbf{u}||_1 = \max_{||\mathbf{u}||_1=1} \left(\sum_{i=1}^{k} |\sum_{j=1}^{k} g_{ij}u_j| \right)$$
$$\leq \max_{||\mathbf{u}||_1=1} \left(\sum_{i=1}^{k} \sum_{j=1}^{k} |g_{ij}||u_j| \right) = \max_{||\mathbf{u}||_1=1} \left(\sum_{i=1}^{k} |u_j| \sum_{i\neq j}^{k} |g_{ij}| \right)$$
$$\leq \max_{||\mathbf{u}||_1=1} (\mu[k-1]||\mathbf{u}||_1) = \mu[k-1].$$

(b) Zauważmy, że $\mathbf{X}_I^T\mathbf{X}_I = \mathbf{I} + \mathbf{G}$, zatem korzystając z własności normy oraz zbieżności szeregu geometrycznego (ponieważ zakładamy $||\mathbf{G}||_{1,1}|| < 1$) mamy

$$||(\mathbf{X}_I^T\mathbf{X}_I)^{-1}||_{1,1} = ||(\mathbf{I}+\mathbf{G})^{-1}||_{1,1}|| \leq \sum_{i=0}^{\infty} ||\mathbf{G}||_{1,1}^i \leq \frac{1}{1-||\mathbf{G}||_{1,1}} \leq \frac{1}{1-\mu[k-1]}.$$

□

Ustalmy $I \subset \{1,...,p\}$, gdzie $|I| = k$. Przez $\hat{\boldsymbol{\beta}}_I$ będziemy oznaczać estymator metody najmniejszych kwadratów otrzymany dla macierzy $\mathbf{X}_I$: $\hat{\boldsymbol{\beta}}_I = (\mathbf{X}_I^T\mathbf{X}_I)^{-1}\mathbf{X}_I^T\mathbf{y}$. Po uzupełnieniu go zerami na pozostałych współrzędnych otrzymamy pełny wektor $\hat{\boldsymbol{\beta}}$, którego zależność od I będziemy pomijali dla uproszczenia notacji.

Pamiętając, że $\mathbf{X}_j, j = 1,...,p$ oznaczają kolumny $\mathbf{X}$ oraz że $||\mathbf{X}_j||_2 = 1$, wprowadźmy sortowanie korelacji:

$$|\langle \mathbf{X}_{(1)}, \mathbf{y}\rangle| \geq |\langle \mathbf{X}_{(2)}, \mathbf{y}\rangle| \geq ... \geq |\langle \mathbf{X}_{(p)}, \mathbf{y}\rangle|$$

Dodatkowo, przez $\mathbf{x}_i, i = 1,...,n$ będziemy oznaczali wiersze macierzy $\mathbf{X}$. Niech wówczas $||\mathbf{x}_i||_{1:k}$ oznacza sumę k największych co do wartości bezwzględnej elementów $x_{ij}, j = 1,...,p$. Ograniczenia na stałe $\mathcal{M}_U, \mathcal{M}_\ell, \mathcal{M}_U^\xi, \mathcal{M}_\ell^\xi$ wyraża następujące twierdzenie:

Twierdzenie 2.3. *Dla dowolnego $k \geq 1$ takiego, że $\mu[k-1] < 1$ rozwiązanie $\hat{\boldsymbol{\beta}}$ problemu* (0.1) *spełnia:*

a)
$$||\hat{\boldsymbol{\beta}}||_1 \leq \frac{1}{1-\mu[k-1]} \sum_{j=1}^{k} |\langle \mathbf{X}_{(j)}, \mathbf{y}\rangle|,$$

b)

$$||\hat{\boldsymbol{\beta}}||_\infty \leq \min\left\{\frac{1}{\gamma_k}\sqrt{\sum_{j=1}^{k}|\langle \mathbf{X}_{(j)}, \mathbf{y}\rangle|^2}, \frac{1}{\sqrt{\gamma_k}}||\mathbf{y}||_2\right\},$$

c)

$$||\mathbf{X}\hat{\boldsymbol{\beta}}||_1 \leq \min\left\{\sum_{i=1}^{n}||\mathbf{x}_i||_\infty||\hat{\boldsymbol{\beta}}||_1, \sqrt{k}||\mathbf{y}||_2\right\},$$

d)

$$||\mathbf{X}\hat{\boldsymbol{\beta}}||_\infty \leq \left(\max_{i=1,...,n}||\mathbf{x}_i||_{1:k}\right)||\hat{\boldsymbol{\beta}}||_\infty.$$

Dowód. (a) Mamy $\hat{\boldsymbol{\beta}}_I = (\mathbf{X}_I^T\mathbf{X}_I)^{-1}\mathbf{X}_I^T\mathbf{y}$, zatem

$$||\hat{\boldsymbol{\beta}}||_1 = ||\hat{\boldsymbol{\beta}}_I||_1 \leq ||(\mathbf{X}_I^T\mathbf{X}_I)^{-1}||_{1,1}||\mathbf{X}_I^T\mathbf{y}||_1, \tag{2.7}$$

oraz

$$||\mathbf{X}_I^T\mathbf{y}||_1 = \sum_{j\in I}|\langle \mathbf{X}_j, \mathbf{y}\rangle| \leq \max_{I,|I|=k}\sum_{j\in I}|\langle \mathbf{X}_j, \mathbf{y}\rangle| \leq \sum_{j=1}^{k}|\langle \mathbf{X}_{(j)}, \mathbf{y}\rangle|.$$

Skąd korzystając z punktu (b) faktu (2.2) w (2.7) otrzymujemy tezę:

$$||\hat{\boldsymbol{\beta}}||_1 \leq ||(\mathbf{X}_I^T\mathbf{X}_I)^{-1}||_{1,1}||\mathbf{X}_I^T\mathbf{y}||_1 \leq \frac{1}{1-\mu[k-1]}\sum_{j=1}^{k}|\langle \mathbf{X}_{(j)}, \mathbf{y}\rangle|.$$

(b) Oznaczmy $\mathbf{A} := (\mathbf{X}_I^T\mathbf{X}_I)^{-1}\mathbf{X}_I^T, \mathbf{C} := (\mathbf{X}_I^T\mathbf{X}_I)^{-1}$ i niech $\mathbf{a}_i, \mathbf{c}_i, i = 1, ..., k$ będą wierszami macierzy $\mathbf{A}$ i $\mathbf{C}$ odpowiednio. Z nierówności Schwarza otrzymujemy:

$$||\hat{\boldsymbol{\beta}}||_\infty = \max_{i=1,...,k}|\langle \mathbf{a}_i, \mathbf{y}\rangle| \leq \left(\max_{i=1,...,k}||\mathbf{a}_i||_2\right)||\mathbf{y}||_2. \tag{2.8}$$

Dalej, dla każdego $i = 1, ..., k$ zachodzi

$$||\mathbf{a}_i||_2 \leq \max_{||\mathbf{u}||_2=1}||\mathbf{A}\mathbf{u}||_2 = \max_{||\mathbf{u}||_2=1}||(\mathbf{X}_I^T\mathbf{X}_I)^{-1}\mathbf{X}_I^T\mathbf{u}||_2$$

$$= \lambda_{max}\left((\mathbf{X}_I^T\mathbf{X}_I)^{-1}\mathbf{X}_I^T\right) = \max\left\{\frac{1}{d_1}, ..., \frac{1}{d_k}\right\}, \tag{2.9}$$

gdzie $d_1, ..., d_k$ to wartości osobliwe macierzy $\mathbf{X}_I$. Skorzystaliśmy z rozkładu na wartości osobliwe $\mathbf{X}_I = \mathbf{U}\mathbf{D}\mathbf{V}^T$, gdzie $\mathbf{D} = diag(d_1, ..., d_k)$. Rzeczywiście

$$(\mathbf{X}_I^T\mathbf{X}_I)^{-1}\mathbf{X}_I^T = (\mathbf{V}\mathbf{D}\mathbf{U}^T\mathbf{U}\mathbf{D}\mathbf{V}^T)^{-1}\mathbf{V}\mathbf{D}\mathbf{U}^T = (\mathbf{V}\mathbf{D}^2\mathbf{V}^T)^{-1}\mathbf{V}\mathbf{D}\mathbf{U}^T$$

$$= \mathbf{V}\mathbf{D}^{-2}\mathbf{V}^T\mathbf{V}\mathbf{D}\mathbf{U}^T = \mathbf{V}\mathbf{D}^{-1}\mathbf{U}^T,$$

skąd wartościami osobliwymi $(\mathbf{X}_I^T\mathbf{X}_I)^{-1}\mathbf{X}_I^T$ są $1/d_i, i = 1, ..., k$. Wiedząc, że wartościami własnymi $\mathbf{X}_I^T\mathbf{X}_I$ są d_i^2 oraz korzystając z ograniczenia zdefiniowanego w (2.6) w (2.9) mamy:

$$\max_{i=1,...,k} ||\mathbf{a_i}||_2 \leq \frac{1}{\sqrt{\gamma_k}}.$$

Stosując powyższą nierówność do (2.8) otrzymujemy pierwsze ograniczenie:

$$||\hat{\boldsymbol{\beta}}||_\infty \leq \frac{1}{\sqrt{\gamma_k}}||\mathbf{y}||_2.$$

Dla drugiego ograniczenia wykorzystamy macierz $\mathbf{C}$.

$$||\hat{\boldsymbol{\beta}}||_\infty = \max_{i=1,...k} |\langle \mathbf{c}_i, \mathbf{X}^T\mathbf{y}\rangle| \leq \left(\max_{i=1,...k} ||\mathbf{c}_i||_2\right) ||\mathbf{X}^T\mathbf{y}||_2 \leq \lambda_{max}\left(\left(\mathbf{X}_I^T\mathbf{X}_I\right)^{-1}\right) ||\mathbf{X}^T\mathbf{y}||_2$$
$$= \left(\max_{i=1,...k} \frac{1}{d_i^2}\right) \sqrt{\sum_{j\in I} |\langle \mathbf{X}_j, \mathbf{y}\rangle|^2} \leq \frac{1}{\gamma_k}\sqrt{\sum_{j\in I} |\langle \mathbf{X}_{(j)}, \mathbf{y}\rangle|^2}.$$

(c) Zauważmy najpierw, że

$$||\mathbf{X}_I\hat{\boldsymbol{\beta}}_I||_1 \leq \sum_{i=1}^n |\langle \mathbf{x}_i, \hat{\boldsymbol{\beta}}_I\rangle| \leq \sum_{i=1}^n ||\mathbf{x}_i||_\infty ||\hat{\boldsymbol{\beta}}_I||_1.$$

Przy dowodzie drugiego ograniczenia skorzystamy z macierzy $\mathbf{H}_I := \mathbf{X}_I(\mathbf{X}_I^T\mathbf{X}_I)^{-1}\mathbf{X}_I^T$. Ponieważ jest to macierz rzutu $\mathbf{y}$ na mniejszą podprzestrzeń zachodzi $||\mathbf{H}_I\mathbf{y}||_2 \leq ||\mathbf{y}||_2$, skąd pozostaje skorzystać z prostego wniosku z nierówności Schwarza:

$$||\mathbf{X}_I\hat{\boldsymbol{\beta}}_I||_1 = ||\mathbf{H}_I\mathbf{y}||_1 \leq \sqrt{k}||\mathbf{H}_I\mathbf{y}||_2 \leq \sqrt{k}||\mathbf{y}||_2.$$

(d) Dla dowolnego $\hat{\boldsymbol{\beta}}_I$:

$$||\mathbf{X}\hat{\boldsymbol{\beta}}_I||_\infty \leq \max_{i=1,...,n} |\langle \mathbf{x}_i, \hat{\boldsymbol{\beta}}_I\rangle| \leq \max_{i=1,...,n} ||\mathbf{x}_i||_{1:k}||\hat{\boldsymbol{\beta}}_I||_\infty.$$

□

Zauważmy, że założenia $\mu[k-1] < 1$ używamy tylko dla punktu (a). Gdy nie jest ono spełnione należy użyć innej metody oszacowania $||\hat{\boldsymbol{\beta}}||_1$. Wszystkie wartości parametrów oszacowaliśmy przy pomocy wyrażeń dostępnych wprost z $\mathbf{X}$ i $\mathbf{y}$ oraz γ_k i $\mu[k-1]$. W praktyce wystarcza oszacowanie dwóch ostatnich parametrów w myśl faktu 1. Dodatkowo oszacowania (c) i (d) wykorzystują oszacowania (a) i (b) odpowiednio, zatem można wykorzystać je po oszacowaniu $||\hat{\boldsymbol{\beta}}||_1$ i $||\hat{\boldsymbol{\beta}}||_\infty$ w inny sposób. W następującym podrozdziale wprowadzimy metodę podającą ograniczenia wykorzystującą algorytmy pierwszego rzędu.

3 Estymacja metodami pierwszego rzędu

Następujące rozwinięcie pewnej metody optymalizacji wypukłej pierwszego rzędu umożliwia otrzymanie niemal optymalnych rozwiązań problemu najlepszego podzbioru. Rozważmy najpierw następujące zagadnienie:

$$\min_{\boldsymbol{\beta}} g(\boldsymbol{\beta}) \ \text{ takie, że } ||\boldsymbol{\beta}|| \leq k, \tag{3.1}$$

gdzie $g(\boldsymbol{\beta}) \geq 0$ jest wypukła i jej gradient spełnia warunek Lipschitza

$$||\nabla g(\boldsymbol{\beta}) - \nabla g(\mathbf{a})|| \leq \ell ||\boldsymbol{\beta} - \mathbf{a}||. \tag{3.2}$$

Fakt 3.1. *Niech $g(\boldsymbol{\beta}) = ||\boldsymbol{\beta} - \mathbf{c}||_2^2$, dla ustalonego $\mathbf{c}$. Wówczas rozwiązanie problemu* (3.1) *ma postać*

$$\hat{\beta}_i = \begin{cases} c_i, & \text{gdy } i \in \{(1), ..., (k)\} \\ 0, & \text{w przeciwnym przypadku} \end{cases} \tag{3.3}$$

gdzie $|c_{(1)}| \geq |c_{(2)}| \geq ... \geq |c_{(p)}|$ są posortowanymi elementami wektora $\mathbf{c} \in \mathbb{R}^p$ i $\hat{\beta}_i$ oznacza i-tą współrzędną wektora $\hat{\boldsymbol{\beta}}$. Zbiór rozwiązań (3.1) będziemy oznaczali przez $\mathbf{H}_k(\mathbf{c})$.

Następnie zastosujemy ograniczenie funkcji $g(\boldsymbol{\eta})$ wokół $g(\boldsymbol{\beta})$ korzystające z metody gradientu prostego dla metod optymalizacji wypukłej pierwszego rzędu wprowadzone w [6] i [5].

Fakt 3.2. *[6] Dla ustalonej wypukłej funkcji $g(\boldsymbol{\beta})$ spełniającej* (3.2) *i dla każdego $L \geq \ell$, dla dowolnych $\boldsymbol{\eta}, \boldsymbol{\beta}$ mamy*

$$g(\boldsymbol{\eta}) \leq Q_L(\boldsymbol{\eta}, \boldsymbol{\beta}) := g(\boldsymbol{\beta}) + \frac{L}{2}||\boldsymbol{\eta} - \boldsymbol{\beta}||_2^2 + \langle \nabla g(\boldsymbol{\beta}), \boldsymbol{\eta} - \boldsymbol{\beta} \rangle, \tag{3.4}$$

gdzie równość zachodzi dla $\boldsymbol{\eta} = \boldsymbol{\beta}$.

Korzystając z faktu (3.1) w ograniczeniu $Q_L(\boldsymbol{\eta}, \boldsymbol{\beta})$ z faktu (3.2) otrzymamy

$$\begin{aligned} \arg\min_{||\boldsymbol{\eta}||_0 \leq k} Q_L(\boldsymbol{\eta}, \boldsymbol{\beta}) &= \arg\min_{||\boldsymbol{\eta}||_0 \leq k} \left(g(\boldsymbol{\beta}) + \frac{L}{2} \left\| \boldsymbol{\eta} - \left(\boldsymbol{\beta} - \frac{1}{L} \nabla g(\boldsymbol{\beta}) \right) \right\|_2^2 - \frac{1}{2L} \left\| \nabla g(\boldsymbol{\beta}) \right\|_2^2 \right) \\ &= \arg\min_{||\boldsymbol{\eta}||_0 \leq k} \left\| \boldsymbol{\eta} - \left(\boldsymbol{\beta} - \frac{1}{L} \nabla g(\boldsymbol{\beta}) \right) \right\|_2^2 = \mathbf{H}_k \left(\boldsymbol{\beta} - \frac{1}{L} \nabla g(\boldsymbol{\beta}) \right). \end{aligned} \tag{3.5}$$

Stosując powyższe równanie możemy zapisać następujący algorytm:

Algorytm 3.3. *Wejście: funkcja $g(\boldsymbol{\beta})$, parametr L taki jak w fakcie* (3.2)*, parametr tolerancji zbieżności ε. Wyjście: Rozwiązanie stacjonarne pierwszego rzędu $\boldsymbol{\beta}^*$.*

1. *Ustal $\boldsymbol{\beta}_1 \in \mathbb{R}^p$ takie, że $||\boldsymbol{\beta}_1||_0 \leq k$.*
2. *Dla $m \geq 1$ wykonuj* (3.5)*, gdzie $\boldsymbol{\beta} = \boldsymbol{\beta}_m$ a $\boldsymbol{\beta}_{m+1}$ jest postaci*

$$\boldsymbol{\beta}_{m+1} \in \mathbf{H}_k \left(\boldsymbol{\beta}_m - \frac{1}{L} \nabla g(\boldsymbol{\beta}_m) \right). \tag{3.6}$$

3. *Powtarzaj 2. tak długo, aż $g(\boldsymbol{\beta}_m) - g(\boldsymbol{\beta}_{m+1}) \leq \varepsilon$.*

3.1 Analiza zbieżności algorytmu

Przy analizie zbieżności algorytmu wprowadzimy pojęcia związane z optymalnością rozwiązania problemu (3.1).

Definicja 3.4. *Dla ustalonego $L \geq \ell$ wektor $\boldsymbol{\eta} \in \mathbb{R}^p$ nazywamy rozwiązaniem stacjonarnym pierwszego rzędu problemu* (3.1)*, gdy $||\boldsymbol{\eta}||_0 \leq k$ i spełnia następujące równanie:*

$$\boldsymbol{\eta} \in \mathbf{H}_k \left(\boldsymbol{\eta} - \frac{1}{L} \nabla g(\boldsymbol{\eta} \right). \tag{3.7}$$

Zbadajmy własności następującej definicji. Niech $\boldsymbol{\eta}$ spełnia definicję (3.4). Wówczas $||\boldsymbol{\eta}|| \leq k$, zatem istnieje zbiór $I \subset \{1, ..., p\}$ taki, że $\eta_i = 0, i \in I$ oraz zbiór I^c mocy k. Skoro $\boldsymbol{\eta} \in \mathbf{H}_k(\boldsymbol{\eta} - \frac{1}{L}\nabla g(\boldsymbol{\eta})$, to dla $i \notin I$ (równoważnie $i \in I^c$) mamy: $\eta_i = \eta_i - \frac{1}{L}\nabla_i g(\boldsymbol{\eta})$, gdzie $\nabla_i g(\boldsymbol{\eta})$ oznacza i-tą współrzędną $\nabla g(\boldsymbol{\eta})$. Stąd $\nabla_i g(\boldsymbol{\eta}) = 0, i \notin I$. Skoro $g(\boldsymbol{\eta})$ jest wypukłą funkcją $\boldsymbol{\eta}$, to oznacza, że $\boldsymbol{\eta}$ jest rozwiązaniem problemu optymalizacji wypukłej:

$$\min_{\boldsymbol{\eta}} g(\boldsymbol{\eta}) \text{ takie, że } \eta_i = 0, \ i \in I. \tag{3.8}$$

Zauważmy, że odwrotna implikacja nie zachodzi, to jest dla ustalonego $I' \subset \{1, ..., p\}$ mocy k rozwiązanie problemu (3.8) z $I = I'$ nie musi odpowiadać punktowi stacjonarnemu w rozumieniu definicji (3.4). Pokażemy za to później, że dowolne globalne rozwiązanie (3.1) jest również punktem stacjonarnym.

Wprowadźmy następujące oznaczenia związane z algorytmem. Niech $\boldsymbol{\beta}_m = (\beta_{m1}, ..., \beta_{mp})$ będzie elementem ciągu generowanego przez algorytm i $\mathbf{1}_m = (e_1, ..., e_p)$ z $e_j = 1$, gdy $\beta_{mj} \neq 0$ i $e_j = 0$, gdy $\beta_{mj} = 0$, dla $j = 1, ..., p$ - $\mathbf{1}_m$ jest wektorem

odpowiadającym rzadkości wektora $\boldsymbol{\beta}_m$. Rozważmy sortowanie współrzędnych wektora $\boldsymbol{\beta}_m$ względem ich wartości bezwzględnych: $|\beta_{(1),m}| \geq |\beta_{(2),m}| \geq ... \geq |\beta_{(p),m}|$. Zgodnie z (3.6) $\beta_{(i),m} = 0$, gdy $i > k$ i $m \geq 2$. Przez $\alpha_{k,m} = |\beta_{(k),m}|$ oznaczamy k-tą największą współrzędną wektora $\boldsymbol{\beta}_m$ co do wartości bezwzględnej. Podobnie jak powyżej, dla $m \geq 2$ gdy $\alpha_{k,m} > 0$, to $||\boldsymbol{\beta}_m||_0 = k$ oraz gdy $\alpha_{k,m} = 0$, to $||\boldsymbol{\beta}_m||_0 < k$. Niech dodatkowo $\bar{\boldsymbol{\alpha}}_{\boldsymbol{k}} := \limsup_{m\to\infty} \alpha_{k,m}$ oraz $\underline{\boldsymbol{\alpha}}_{\boldsymbol{k}} := \liminf_{m\to\infty} \alpha_{k,m}$.

Możemy teraz sformułować następujące twierdzenie.

Twierdzenie 3.5. *Rozważmy $g(\boldsymbol{\beta})$ i ℓ takie jak w* (3.2) *oraz $\boldsymbol{\beta}_m, m \geq 1$ takie jak w* (3.6).

a) Dla każdego $L \geq \ell$ ciąg $g(\boldsymbol{\beta}_m)$ jest malejący, zbieżny i spełnia

$$g(\boldsymbol{\beta}_m) - g(\boldsymbol{\beta}_{m+1}) \geq \frac{L-\ell}{2} \left\|\boldsymbol{\beta}_{m+1} - \boldsymbol{\beta}_m\right\|_2^2.$$

b) Jeśli $L > \ell$, to $\boldsymbol{\beta}_{m+1} - \boldsymbol{\beta}_m \to 0$, gdy $m \to \infty$.

c) Jeśli $L > \ell$ i $\underline{\boldsymbol{\alpha}}_{\boldsymbol{k}} > 0$, to ciąg $\mathbf{1}_m$ zbiega po skończenie wielu iteracjach, to znaczy istnieje indeks $\boldsymbol{M}^*$ taki, że $\mathbf{1}_m = \mathbf{1}_{m+1}$ dla każdego $m \geq \boldsymbol{M}^*$. Dodatkowo ciąg $\boldsymbol{\beta}_m$ jest ograniczony i zbieżny do punktu stacjonarnego pierwszego rzędu.

d) Jeśli $L > \ell$ i $\underline{\boldsymbol{\alpha}}_{\boldsymbol{k}} = 0$, to $\liminf_{m\to\infty} ||\nabla g(\boldsymbol{\beta}_m)||_\infty = 0$.

e) Jeśli $L > \ell$, $\bar{\boldsymbol{\alpha}}_{\boldsymbol{k}} = 0$ i ciąg $\boldsymbol{\beta}_m$ ma punkt skupienia, to $g(\boldsymbol{\beta}_m) \to \min_\beta g(\boldsymbol{\beta})$.

Dowód. (a)
Niech $\boldsymbol{\beta} \in \mathbb{R}^p$ taki, że $||\boldsymbol{\beta}||_0 \leq k$ i niech $\boldsymbol{\eta}' \in \mathbf{H}_k(\boldsymbol{\beta} - \frac{1}{L}\nabla g(\boldsymbol{\beta}))$. Korzystając z faktu (3.2) i równości (3.5) mamy następujący ciąg nierówności:

$$\begin{aligned}
g(\boldsymbol{\beta}) = Q_L(\boldsymbol{\beta}, \boldsymbol{\beta}) &\geq \inf_{||\boldsymbol{\eta}||_0 \leq k} Q_L(\boldsymbol{\eta}, \boldsymbol{\beta}) \\
&= \inf_{||\boldsymbol{\eta}||_0 \leq k} \left(g(\boldsymbol{\beta}) + \frac{L}{2}||\boldsymbol{\eta} - \boldsymbol{\beta}||_2^2 + \langle \nabla g(\boldsymbol{\beta}), \boldsymbol{\eta} - \boldsymbol{\beta} \rangle \right) \\
&= \inf_{||\boldsymbol{\eta}||_0 \leq k} \left(g(\boldsymbol{\beta}) + \frac{L}{2} \left\| \boldsymbol{\eta} - \left(\boldsymbol{\beta} - \frac{1}{L}\nabla g(\boldsymbol{\beta}) \right) \right\|_2^2 - \frac{1}{2L} \left\| \nabla g(\boldsymbol{\beta}) \right\|_2^2 \right) \\
&= \left(g(\boldsymbol{\beta}) + \frac{L}{2} \left\| \boldsymbol{\eta}' - \left(\boldsymbol{\beta} - \frac{1}{L}\nabla g(\boldsymbol{\beta}) \right) \right\|_2^2 - \frac{1}{2L} \left\| \nabla g(\boldsymbol{\beta}) \right\|_2^2 \right)
\end{aligned}$$

$$= \left(g(\boldsymbol{\beta}) + \frac{L}{2}||\boldsymbol{\eta}' - \boldsymbol{\beta}||_2^2 + \langle \nabla g(\boldsymbol{\beta}), \boldsymbol{\eta}' - \boldsymbol{\beta} \rangle \right)$$

$$= \left(\frac{L-\ell}{2}||\boldsymbol{\eta}' - \boldsymbol{\beta}||_2^2 + g(\boldsymbol{\beta}) + \frac{\ell}{2}||\boldsymbol{\eta}' - \boldsymbol{\beta}||_2^2 + \langle \nabla g(\boldsymbol{\beta}), \boldsymbol{\eta}' - \boldsymbol{\beta} \rangle \right)$$

$$= \frac{L-\ell}{2}||\boldsymbol{\eta}' - \boldsymbol{\beta}||_2^2 + Q_\ell(\boldsymbol{\eta}', \boldsymbol{\beta}) \geq \frac{L-\ell}{2}||\boldsymbol{\eta}' - \boldsymbol{\beta}||_2^2 + g(\boldsymbol{\eta}'). \tag{3.9}$$

Zatem ostatecznie

$$g(\boldsymbol{\beta}) - g(\boldsymbol{\eta}') \geq \frac{L-\ell}{2}||\boldsymbol{\eta}' - \boldsymbol{\beta}||_2^2. \tag{3.10}$$

Po podstawieniu $\boldsymbol{\beta} = \boldsymbol{\beta}_m$ i $\boldsymbol{\eta}' = \boldsymbol{\beta}_{m+1}$ otrzymujemy zadane ograniczenie. Z (3.10) wynika również, że ciąg $g(\boldsymbol{\beta}_m)$ jest malejący, a ponieważ jest ograniczony z dołu, bo $g(\boldsymbol{\beta}) \geq 0$, to jest zbieżny.

(b) Gdy $L > \ell$, tezę otrzymujemy wprost z rozumowania dla (a).

(c) Warunek $\underline{\boldsymbol{\alpha}}_k > 0$ pociąga, że od pewnego m elementy postaci $|\beta_{(k),m}|$ będą oddzielone od zera. Przypuśćmy niewprost, że ciąg $\mathbf{1}_m$ nie jest zbieżny. Wówczas istnieje nieskończenie wiele wartości m' takich, że $\mathbf{1}_m \neq \mathbf{1}_{m+1}$. Korzystając z faktu, że dla $||\boldsymbol{\beta}||_0 = k$ i prostych nierówności między normami dla dostatecznie dużych m' mamy:

$$||\boldsymbol{\beta}_{m'} - \boldsymbol{\beta}_{m'+1}|| \geq \sqrt{\beta_{m',i}^2 + \beta_{m'+1,j}^2} \geq \frac{|\beta_{m',i}| + |\beta_{m'+1,j}|}{2},$$

gdzie i, j wybieramy tak, aby $\beta_{m'+1,i} = \beta_{m',j} = 0$. Dla dużych m' wielkość po prawej stronie jest ściśle większa od zera, ponieważ $\underline{\boldsymbol{\alpha}}_k > 0$. Otrzymujemy sprzeczność, ponieważ na mocy (b) wielkość po lewej stronie zbiega do zera. Uzyskana sprzeczność dowodzi, że ciąg $\mathbf{1}_m$ jest zbieżny. Ponieważ może on przyjąć skończenie wiele wartości zbiegnie po skończenie wielu iteracjach algorytmu, innymi słowy, $\mathbf{1}_m = \mathbf{1}_{m+1}$ dla wszystkich $m \geq M^*$. Wówczas algorytm redukuje się do zwykłej metody gradientu prostego na ustalonym zbiorze $\mathbf{1}_m$. Korzystając z faktu z [5], że metoda gradientu prostego minimalizacji wypukłej funkcji po zbiorze domkniętym prowadzi do ciągu zbieżnego otrzymujemy, że algorytm zbiega. Oznacza to, że ciąg $\boldsymbol{\beta}_m$ zbiega do $\boldsymbol{\beta}^*$ - punktu stacjonarnego pierwszego rodzaju. Ograniczność wynika ze zbieżności $\boldsymbol{\beta}_m$.

(d) Niech $\mathcal{I}_{max} \subset \{1, ..., p\}$ będzie zbiorem k największych wartości wektora $(\boldsymbol{\beta}_m - \frac{1}{L}\nabla g(\boldsymbol{\beta}_m))$ co do wartości bezwzględnej. Wobec definicji $\mathbf{H}_k(\boldsymbol{\beta}_m - \frac{1}{L}\nabla g(\boldsymbol{\beta}_m))$ mamy

$$\left\| \left(\boldsymbol{\beta}_m - \frac{1}{L}\nabla g(\boldsymbol{\beta}_m) \right)_i \right\| \geq \left\| \left(\boldsymbol{\beta}_m - \frac{1}{L}\nabla g(\boldsymbol{\beta}_m) \right)_j \right\|,$$

dla każdego i, j takich, że $i \in \mathcal{I}_{max}, j \notin \mathcal{I}_{max}$. Dalej

$$\liminf_{m\to\infty} \min_{i\in\mathcal{I}_{max}} \left\| \left(\boldsymbol{\beta}_m - \frac{1}{L}\nabla g(\boldsymbol{\beta}_m) \right)_i \right\| \geq \liminf_{m\to\infty} \max_{j\notin\mathcal{I}_{max}} \left\| \left(\boldsymbol{\beta}_m - \frac{1}{L}\nabla g(\boldsymbol{\beta}_m) \right)_j \right\|. \quad (3.11)$$

Zauważmy, że w myśl faktu (3.1)

$$\left(\boldsymbol{\beta}_m - \mathbf{H}_k \left(\boldsymbol{\beta}_m - \frac{1}{L}\nabla g(\boldsymbol{\beta}_m) \right) \right)_i = \begin{cases} \frac{1}{L}(\nabla g(\boldsymbol{\beta}_m))_i, & \text{gdy } i \in \mathcal{I}_{max} \\ \beta_{m,i}, & \text{w przeciwnym przypadku.} \end{cases}$$

Korzystając z $\boldsymbol{\beta}_{m+1} - \boldsymbol{\beta}_m \to \mathbf{0}$ mamy

$$(\nabla g(\boldsymbol{\beta}_m))_i \to 0 \text{ dla } i \in \mathcal{I}_{max} \text{ i } \beta_{m,j} \to 0 \text{ dla } j \notin \mathcal{I}_{max},$$

gdy $m \to \infty$. Podstawiając do (3.11) otrzymujemy

$$\liminf_{m\to\infty} \min_{i\in\mathcal{I}_{max}} |\boldsymbol{\beta}_{mi}| \geq \liminf_{m\to\infty} \max_{j\notin\mathcal{I}_{max}} \frac{1}{L}|(\nabla g(\boldsymbol{\beta}_m))_j| = \frac{1}{L}\liminf_{m\to\infty} ||\nabla g(\boldsymbol{\beta}_m)||_\infty.$$

Z założenia $\liminf_{m\to\infty} \min_{i\in\mathcal{I}_{max}} |\boldsymbol{\beta}_{mi}| = \underline{\boldsymbol{\alpha}}_k = 0$, zatem lewa strona powyższego równania wynosi zero, skąd $\liminf_{m\to\infty} ||\nabla g(\boldsymbol{\beta}_m)||_\infty = 0$.

(e) Skorzystamy z części dowodu (d). Odpowiednio przepisując w (3.11) granice dolne na granice górne otrzymujemy:

$$\limsup_{m\to\infty} \min_{i\in\mathcal{I}_{max}} |\boldsymbol{\beta}_{mi}| \geq \limsup_{m\to\infty} \max_{j\notin\mathcal{I}_{max}} \frac{1}{L}|(\nabla g(\boldsymbol{\beta}_m))_j| = \frac{1}{L}\limsup_{m\to\infty} ||\nabla g(\boldsymbol{\beta}_m)||_\infty.$$

Lewa strona równania to dokładnie $\bar{\boldsymbol{\alpha}}_k$, która wynosi zero z założenia, skąd $||\nabla g(\boldsymbol{\beta}_m)||_\infty \to 0$, gdy $m \to \infty$.

Niech $\boldsymbol{\beta}_\infty$ będzie punktem skupienia ciągu $\boldsymbol{\beta}_m$. Wówczas istnieje podciąg $m' \subset \{1, 2, ..., \}$ taki, że $\boldsymbol{\beta}_{m'} \to \boldsymbol{\beta}_\infty$ i $g(\boldsymbol{\beta}_{m'}) \to g(\boldsymbol{\beta}_\infty)$. Korzystając z ciągłości gradientu, a w konsekwencji funkcji $\cdot \to ||\nabla g(\cdot)||_\infty$ mamy, że $||\nabla g(\boldsymbol{\beta}_{m'})||_\infty \to ||\nabla g(\boldsymbol{\beta}_\infty)||_\infty = 0$, gdy $m' \to \infty$. Stąd $\boldsymbol{\beta}_\infty$ jest rozwiązaniem problemu $\min g(\boldsymbol{\beta})$ bez ograniczenia na rzadkość wektora $\boldsymbol{\beta}$. Ponieważ $g(\boldsymbol{\beta}_m)$ jest ciągiem malejącym, zbiega on do minimum funkcji $g(\boldsymbol{\beta})$. □

Zauważmy, że do istnienia punktu skupienia w punkcie (e) twierdzenia wystarczą łatwo osiągalne założenia, przykładowo $\sup\{\boldsymbol{\beta} : ||\boldsymbol{\beta}||_0 \leq k, f(\boldsymbol{\beta}) \leq E\} < \infty$ dla pewnego skończonego E. Powyższy warunek możemy odczytywać jako ograniczoność

zbioru rzadkości k dla funkcji $g(\boldsymbol{\beta})$. W szczególności, gdy $g(\boldsymbol{\beta})$ jest kwadratową funkcją straty, jest to równoważne warunkowi, że każda podmacierz złożona z k kolumn macierzy $\mathbf{X}$ jest pełnego rzędu. W przypadku, gdy elementy macierzy $\mathbf{X}$ są z ciągłego rozkładu, dla $k < n$, warunek ten zachodzi z prawdopodobieństwem 1.

W następującym twierdzeniu podamy własności rozwiązania spełniającego równanie (3.7).

Twierdzenie 3.6. *Niech $L > \ell$. Wówczas mamy:*

a) Jeśli $\boldsymbol{\eta}$ spełnia definicję (3.4)*, to zbiór $\mathbf{H}_k(\boldsymbol{\eta} - \frac{1}{L}\nabla g(\boldsymbol{\beta}))$ zawiera dokładnie jeden element: $\boldsymbol{\eta}$.*

b) Jeśli $\hat{\boldsymbol{\beta}}$ jest globalnym minimum problemu (3.1)*, to jest również punktem stacjonarnym pierwszego rzędu w myśl definicji* (3.4)*.*

Dowód. (a) Przypomnijmy punkt (a) twierdzenia (3.5), z którego wprost wynika, że:

$$g(\boldsymbol{\eta}) - g(\boldsymbol{\eta'}) \geq \frac{L-\ell}{2} \left\|\boldsymbol{\eta'} - \boldsymbol{\eta}\right\|_2^2,$$

dla dowolnego $\boldsymbol{\eta'} \in \mathbf{H}_k(\boldsymbol{\eta} - \frac{1}{L}\nabla g(\boldsymbol{\eta}))$. Z definicji $\mathbf{H}_k(\cdot)$ mamy $g(\boldsymbol{\eta}) = g(\boldsymbol{\eta'})$, skąd lewa strona nierówności wynosi zero, zatem przy $L > \ell, ||\boldsymbol{\eta'} - \boldsymbol{\eta}_2 = 0$, czyli $\boldsymbol{\eta} = \boldsymbol{\eta'}$. Wobec dowolności $\boldsymbol{\eta'}$ zbiór $\mathbf{H}_k(\boldsymbol{\eta} - \frac{1}{L}\nabla g(\boldsymbol{\eta}))$ zawiera tylko element $\boldsymbol{\eta}$.

(b) Skorzystamy z faktu, że wektor $\hat{\boldsymbol{\beta}}$ jest rzadkości k oraz ponownie odpowiedniego sformułowania punktu (a) twierdzenia (3.5)

$$g(\hat{\boldsymbol{\beta}}) - g(\boldsymbol{\eta'}) \geq \frac{L-\ell}{2} \left\|\boldsymbol{\eta'} - \hat{\boldsymbol{\beta}}\right\|_2^2,$$

dla dowolnego $\boldsymbol{\eta'} \in \mathbf{H}_k(\hat{\boldsymbol{\beta}} - \frac{1}{L}\nabla g(\hat{\boldsymbol{\beta}}))$. Korzystając z własności definicji (3.7) mamy, że $g(\hat{\boldsymbol{\beta}}) = g(\boldsymbol{\eta'})$, zatem na mocy analogicznego rozumowania jak dla punktu (a) otrzymujemy, że $\hat{\boldsymbol{\beta}} \in \mathbf{H}_k(\boldsymbol{\eta} - \frac{1}{L}\nabla g(\boldsymbol{\eta}))$, skąd otrzymujemy tezę.

□

W twierdzeniu (3.5) opisaliśmy zbieżność algorytmu, natomiast w poniższym twierdzeniu podamy szybkość tej zbieżności do punktu stacjonarnego pierwszego rzędu.

Twierdzenie 3.7. *Niech $L > \ell$ i niech $\boldsymbol{\beta}^*$ będzie punktem stacjonarnym pierwszego rzędu. Wówczas po M iteracjach algorytm spełnia*

$$\min_{m=1,\ldots,M} \left\|\boldsymbol{\beta}_{m+1} - \boldsymbol{\beta}_m\right\|_2^2 \leq \frac{2(g(\boldsymbol{\beta}_1) - g(\boldsymbol{\beta}^*))}{M(L-\ell)}, \tag{3.12}$$

gdzie ciąg $g(\boldsymbol{\beta}_m)$ zbiega do $\boldsymbol{\beta}^*$, gdy $m \to \infty$.

Dowód. Sumując nierówności z punktu (a) twierdzenia (3.5) dla $m = 1, ..., M$ otrzymujemy:

$$\sum_{m=1}^{M} \Big(g(\boldsymbol{\beta}_m) - g(\boldsymbol{\beta}_{m+1})\Big) \geq \frac{L-\ell}{2} \sum_{m=1}^{M} ||\boldsymbol{\beta}_{m+1} - \boldsymbol{\beta}_m||_2^2,$$

skąd

$$g(\boldsymbol{\beta}_1) - g(\boldsymbol{\beta}_{M+1}) \geq \frac{M(L-\ell)}{2} \min_{m=1,\ldots,M} ||\boldsymbol{\beta}_{m+1} - \boldsymbol{\beta}_m||_2^2.$$

Zgodnie z twierdzeniem (3.5) ciąg $g(\boldsymbol{\beta}_m)$ jest malejący i zbieżny do $g(\boldsymbol{\beta}^*)$, zatem

$$\frac{g(\boldsymbol{\beta}_1) - g(\boldsymbol{\beta}^*)}{M} \geq \frac{g(\boldsymbol{\beta}_1) - g(\boldsymbol{\beta}_{M+1})}{M} \geq \frac{L-\ell}{2} \min_{m=1,\ldots,M} ||\boldsymbol{\beta}_{m+1} - \boldsymbol{\beta}_m||_2^2.$$

□

Powyższe twierdzenie mówi, że dla dowolnego ε istnieje M takie, że dla pewnego $1 \leq m^* \leq M$ zachodzi $\left\|\boldsymbol{\beta}_{m^*+1} - \boldsymbol{\beta}_{m^*}\right\|_2^2 \leq \varepsilon$. Dodatkowo powyższa szybkość zbieżności zachodzi dla dużej klasy problemów (3.1) dla pewnych wypukłych funkcji g spełniających (3.2). Z punktu (c) twierdzenia (3.5) nośnik $\boldsymbol{\beta}_m$ stabilizuje się po skończenie wielu iteracjach. Po tym czasie algorytm zachowuje się jak metoda gradientu prostego na ustalonym nośniku.

3.2 Zastosowanie dla kwadratowej funkcji straty

Dla problemu optymalnego podzbioru (0.1) i kwadratowej funkcji straty mamy $g(\boldsymbol{\beta}) = \frac{1}{2}||\mathbf{y} - \mathbf{X}\boldsymbol{\beta}||_2^2, \nabla g(\boldsymbol{\beta}) = -\mathbf{X}^T(\mathbf{y} - \mathbf{X}\boldsymbol{\beta})$ oraz $\ell = \lambda_{max}(\mathbf{X}^T\mathbf{X})$. Przy podstawieniu wymienionych wartości wyniki uzyskane w tym rozdziale możemy bezpośrednio zastosować.

4 Porównanie omawianych metod

4.1 Badanie symulacyjne

Między innymi z umówionych w powyższej części wyników korzysta pakiet *L0Learn* w R. W szczególności funkcja *L0Learn.fit* rozwiązuje problem optymalizayjny

$$\min_{\beta_0,\boldsymbol{\beta}} \sum_{i=1}^{n} \ell(y_i, \beta_0 + \langle x_i, \boldsymbol{\beta}\rangle) + \lambda||\boldsymbol{\beta}||_0,$$

gdzie ℓ jest funkcją straty. Jednym z argumentów funkcji jest maksymalna rzadkość wektora $\boldsymbol{\beta}$, zatem przy ustaleniu go na wartość $k \in \{0, 1, ...\}$ oraz wyborze kwadratowej funkcji straty funkcja *L0Learn.fit* rozwiązuje problem optymlanego podzbioru (0.1). Wynikiem działania funkcji jest kilkanaście modeli, co najwyżej po jednym dla każdej dopuszczalnej rzadkości wektora współczynników oraz związana z nimi wartość parametru λ.

Skorzystamy również z funkcji *L0Learn.cvfit*, która po ustaleniu dodatkowych parametrów zwraca błędy walidacji krzyżowej, obliczane dla każdego modelu, równoważnie każdej z wybranych przez *L0Learn.fit* wartości λ.

Poniższe wyniki dotyczą badania symulacyjnego porównującego kryterium mBIC2 z funkcjami z biblioteki *L0Learn*. We wszystkich symulacjach rozważamy model regresji liniowej dla $n = p = 500$, błąd ε pochodzi z rozkładu normalnego $N(0, I)$, k - liczba niezerowych współczynników wektora $\boldsymbol{\beta}$ przyjmuje wartości $k \in \{10, 20, 40, 60, 80, 100\}$. Wyróżniamy dwa poziomy siły sygnału. Słaby sygnał utożsamiamy z wektorem $\boldsymbol{\beta}$ postaci:

$$\beta_1 = \beta_2 = ... = \beta_k = 1.3\sqrt{2\log p}$$

natomiast silny sygnał z

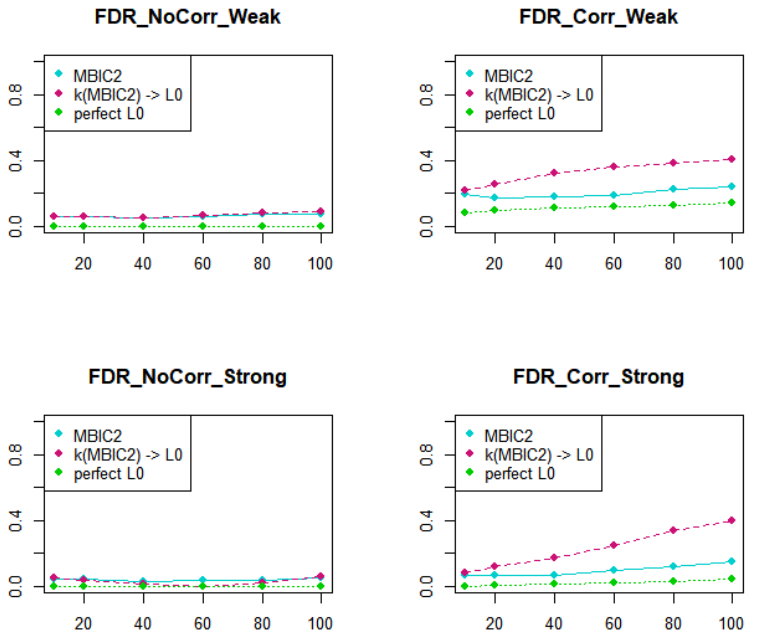
$$\beta_1 = \beta_2 = ... = \beta_k = 2\sqrt{2\log p},$$

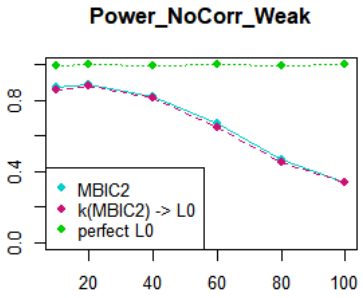
pozostałe współrzędne wektora β są zerami. Wiersze macierzy planu X są niezależnymi wektorami losowymi z wielowymiarowego rozkładu normalnego $N\left(0, \frac{1}{n}\Sigma\right)$. Rozważamy dwa przypadki: niezależnych zmiennych objaśniających, gdzie $\Sigma = I$ i skorelowanych zmiennych objaśniających, gdzie $\Sigma_{i,j} = 1$, gdy $i = j$ i $\Sigma_{i,j} = 0,5$, gdy $i \neq j$.

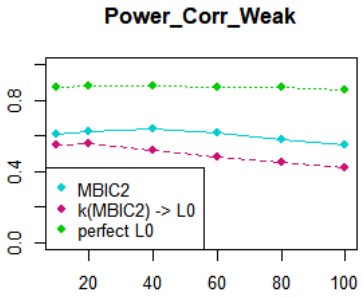
Dla każdej symulacji obliczamy wartości czterech statystyk: frakcja fałsywych odkryć (**FDR**), **moc**, **MSE** $= MSE(\hat{\beta}) \ / \ ||\beta||^2$ oraz **MSP** $= MSE(\mathbf{X}\hat{\beta}) \ / \ ||\mathbf{X}\beta||^2$. Wyniki oparte są na 200 symulacjach. W symulacji porównujemy następujące metody:

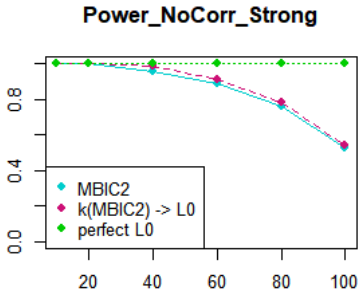
- mBIC2 z procedurą stepwise z biblioteki *bigstep*
- *L0Learn.fit* z karą L0 i optymalną rzadkością wektora współczynników
- *L0Learn.fit* z karą L0 i rzadkością wektora współczynników najbliższą tej, wybranej przez mBIC2.

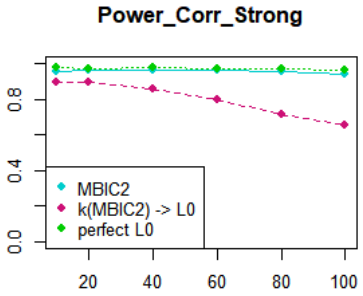
W tym przypadku koncentrujemy się na optymalizacji tylko z dodatkową karą L0. Przez wybór optymalnej rzadkości wektora współczynników rozumiemy wybranie z obiektu generowanego przez *L0Learn.fit* modelu, którego wielkość nośnika jest najbliższa k. Jest to bardzo teoretyczna reguła i silne założenie, ale jest intiucyjna, łatwa do interpretacji i pozwala na uzyskanie pewnych wyników. W eksperymencie trzecim stosujemy bardziej realistyczną regułę wyboru modelu spośród tych, proponowanych przez *L0Learn.fit*. Wybieramy wielkość nośnika najbliższą ilości zmiennych modelu wybranego przez kryterium mBIC2. Wyniki eksperymentów przedstawiają wykresy.

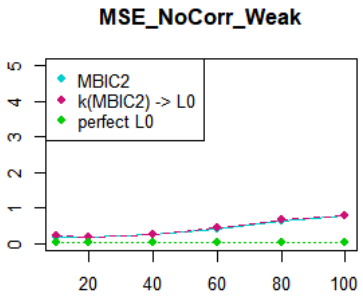


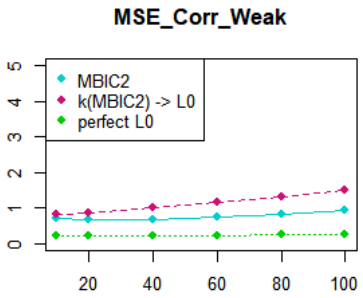

Power_NoCorr_Weak
MBIC2
k(MBIC2) -> L0
perfect L0
0.0
0.4
0.8
20
40
60
80
100

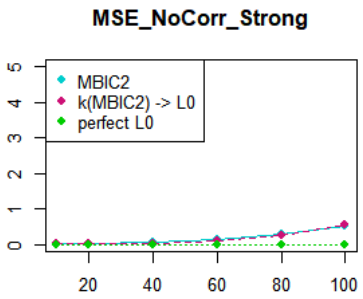

Power_Corr_Weak
MBIC2
k(MBIC2) -> L0
perfect L0
0.0
0.4
0.8
20
40
60
80
100

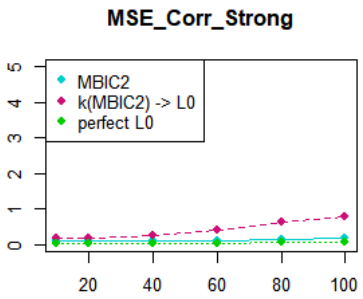

Power_NoCorr_Strong
MBIC2
k(MBIC2) -> L0
perfect L0
0.0
0.4
0.8
20
40
60
80
100

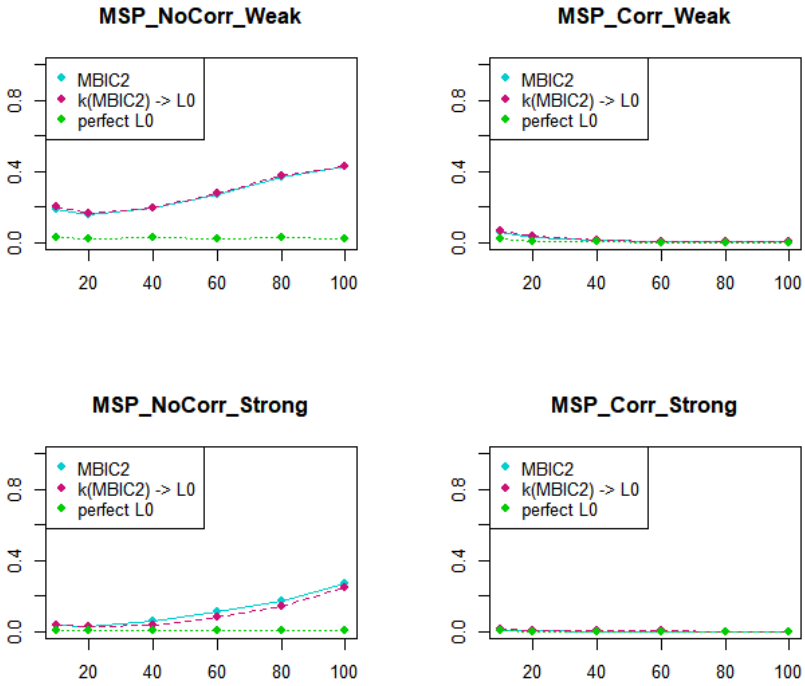

Power_Corr_Strong
MBIC2
k(MBIC2) -> L0
perfect L0
0.0
0.4
0.8
20
40
60
80
100


MSE_NoCorr_Weak
MBIC2
k(MBIC2) -> L0
perfect L0
0
1
2
3
4
5
20
40
60
80
100


MSE_Corr_Weak
MBIC2
k(MBIC2) -> L0
perfect L0
0
1
2
3
4
5
20
40
60
80
100


MSE_NoCorr_Strong
MBIC2
k(MBIC2) -> L0
perfect L0
0
1
2
3
4
5
20
40
60
80
100


MSE_Corr_Strong
MBIC2
k(MBIC2) -> L0
perfect L0
0
1
2
3
4
5
20
40
60
80
100



Zauważmy, że gdy potrafimy podać poprawną ilość istotnych zmiennych, otrzymujemy bardzo dobre wyniki. Wyjście funkcji *L0Learn.fit* zwykle zawiera bardzo dobry model, zatem przydatne będzie znalezienie odpowiedniej metody wyboru modelu spośród proponowanych.

W sytuacji, gdy zmienne objaśniające są niezależne wyniki porównywanych metod są niemal równe, z małymi różnicami dla FDR i mocy z przewagą *L0Learn.fit*. W przypadku korelacji kryterium mBIC2 osiąga lepsze wyniki, różnice są większe dla silniejszego sygnału.

4.2 Analiza rzeczywistego zbioru danych

Rozważymy zbiór poziomów ekspresji genów w limfoblastach u 210 osób z czterech różnych populacji. Celem analizy jest znalezienie genów, których poziom ekspresji istotnie wpływa na poziom ekspresji genu CCT8, odpowiedzialnego za cechy występujące przy zespole Downa.

Nazwa	p-wartość	R(**y**)	R(Z2465)	R(Z1876)	R(Z2516)	R(Z1753)
Z2465	$<$ 2e-16	0.62				
Z1876	4.57e-05	0.50	0.46			
Z2516	8.04e-07	0.30	0.00	0.26		
Z1753	6.83e-07	0.25	-0.01	0.01	0.17	
Z756	3.96e-06	0.05	-0.11	-0.15	-0.22	-0.1

Tablica 1: Własności istotnych zmiennych

Zbiór danych zawiera poziomy ekspresji 47293 genów. Podobnie jak w [2] we wstępnej fazie analizy usuniemy geny, których maksymalny poziom ekspresji wśród badanych był wśród 25% najniższych poziomów ekspresji, oraz te, których rozrzut poziomów ekspresji był mniejszy niż 2. W rezultacie otrzymamy 3220 zmiennych objaśniających.

Korzystając z funkcji *L0Learn.cvfit* wśród modeli proponowanych przez *L0Learn.fit* wybierzemy ten, który ma najmniejsze błędy predykcji obliczone przy pomocy walidacji krzyżowej. Wartość parametru λ dla optymalnego modelu wynosi 0,0145525 i prowadzi do pięciu istotnych zmiennych: Z756, Z1753, Z1876, Z2465, Z2516. Własności wybranych zmiennych przedstawia tabela (1):

Wartość R^2 dla tego modelu to 0,588, natomiast RSS wynosi 86,11. Wynik analizy tego samego zbioru danych przy pomocy kryterium mBIC2 z zaawansowaną procedurą stepwise z pakietu *bigstep* [2] również prowadzi do modelu z pięcioma zmiennymi. Dla niego z kolei wartości R^2 i RSS wynoszą odpowiednio 0,58 i 87,56. Pamiętając o tym, że dla modeli z taką samą ilością zmiennych porównywanie względem residualnej sumy kwadratów zasadne możemy stwierdzić, że wartości obu wyżej wymienionych wskaźników były o około 2% lepsze na korzyść *L0Learn*, przy nieporównywalnie krótszym czasie wykonania obliczeń.

5 Wykorzystywany kod w R

```
library(L0Learn)
library(bigstep)
library(mvtnorm)
library(magrittr)
```

```
arr_means = function(arr){
  b = dim(arr)[2]
  c = dim(arr)[3]
  x = matrix(0, b, c)
  for (i in 1:b){
    for (j in 1:c){
      x[i,j] = mean(arr[,i,j])
    }
  }
  return(x)
}

n = 500
p = 500

s_ncorr = matrix(0, n, p)
diag(s_ncorr) = rep(1/n, n)

s_corr = matrix(0.5/n, n, p)
diag(s_corr) = rep(1/n, n)

weak_signal = 1.3 * sqrt(2 * log(p))
strong_signal = 2 * sqrt(2 * log(p))

values = c(10, 20, 40, 60, 80, 100)

test_mbic2 = function(k, nonzero_coef, s) {

  eps = rnorm(n)
  beta = c(rep(nonzero_coef, k), rep(0, p - k))
  X = matrix(rmvnorm(n, sigma = s), n, p)
  Y = X %*% beta + eps

  data = prepare_data(Y, X, verbose = FALSE)
  res_mbic2 <- data %>%
    reduce_matrix() %>%
    fast_forward()%>%
```

```
    multi_backward(crit=mbic2)%>%
    stepwise(crit=mbic2)

  #d = prepare_data(Y, X, verbose = FALSE)
  #e = reduce_matrix(d)
  #f = fast_forward(e)
  #g = multi_backward(f, crit = mbic2)
  #res_mbic2 = stepwise(g, crit = mbic2)
  chosen_vars = res_mbic2$model
  #print(length(chosen_vars))
  est_FDR = mean(as.integer(chosen_vars) > k)
  est_power = sum(as.integer(chosen_vars) <= k)/k

  est_beta_prep = coef(summary(res_mbic2))[2:(length(chosen_vars) + 1)]
  aux = rbind(as.integer(chosen_vars), est_beta_prep)
  est_beta = numeric(p)
  est_beta[aux[1,]] = aux[2,]
  est_mse = sum((est_beta - beta)^2)/sum(beta^2)
  est_msp = sum((X %*% est_beta - X %*% beta)^2)/sum((X %*% beta)^2)
  v = c(est_FDR, est_power, est_mse, est_msp)
  v

return (v)
}

nr_of_repetitions = 200

#eksperyment MBIC2x
{

ncorr_weakx = function(k) {test_mbic2(k, weak_signal, s_ncorr)}
ncorr_strongx = function(k) {test_mbic2(k, strong_signal, s_ncorr)}
corr_weakx = function(k) {test_mbic2(k, weak_signal, s_corr)}
corr_strongx = function(k) {test_mbic2(k, strong_signal, s_corr)}

mbic2x_res_ncorr_weak = array(0, dim = c(nr_of_repetitions, 4, 6))
mbic2x_res_ncorr_strong = array(0, dim = c(nr_of_repetitions, 4, 6))
mbic2x_res_corr_weak = array(0, dim = c(nr_of_repetitions, 4, 6))
```

```
mbic2x_res_corr_strong = array(0, dim = c(nr_of_repetitions, 4, 6))


for (i in 1:nr_of_repetitions){
  mbic2x_res_ncorr_weak[i,,] = sapply(values, ncorr_weakx)
}

for (i in 1:nr_of_repetitions){
  mbic2x_res_ncorr_strong[i,,] = sapply(values, ncorr_strongx)
}

for (i in 1:nr_of_repetitions){
  mbic2x_res_corr_weak[i,,] = sapply(values, corr_weakx)
}

for (i in 1:nr_of_repetitions){
  mbic2x_res_corr_strong[i,,] = sapply(values, corr_strongx)
}

means_ncorr_weakx = arr_means(mbic2x_res_ncorr_weak)
means_ncorr_strongx = arr_means(mbic2x_res_ncorr_strong)
means_corr_weakx = arr_means(mbic2x_res_corr_weak)
means_corr_strongx = arr_means(mbic2x_res_corr_strong)

}


test_L0Learn0 = function(k, nonzero_coef, s) {

  eps = rnorm(n)
  beta = c(rep(nonzero_coef, k), rep(0, p - k))
  X = matrix(rmvnorm(n, sigma = s), n, p)
  Y = X %*% beta + eps

  fit = L0Learn.fit(X, Y, penalty = "L0", maxSuppSize = 1.5 * k)

  access_to_sols = cbind(fit$lambda[[1]], fit$gamma[1], fit$suppSize[[1]])
```

```
  dist_from_correct_p = abs(access_to_sols[,3] - k)
  best_sol = which.min(dist_from_correct_p)

  ats = access_to_sols

  not_est_beta_yet = coef(fit, lambda = ats[best_sol,1],
  gamma = ats[best_sol,2])[-1]
  ind_est_beta_l0 = which(as.vector(not_est_beta_yet) != 0)
  est_beta_l0 = numeric(p)
  est_beta_l0[ind_est_beta_l0] = as.vector(not_est_beta_yet
  [ind_est_beta_l0])

  chosen_vars = ind_est_beta_l0
  chosen_vars
  est_FDR = mean(as.integer(chosen_vars) > k)
  est_power = sum(as.integer(chosen_vars) <= k) / k

  est_mse = sum((est_beta_l0 - beta)^2) / sum(beta^2)
  est_msp = sum((X %*% est_beta_l0 - X %*% beta)^2) / sum((X %*% beta)^2)
  v = c(est_FDR, est_power, est_mse, est_msp)
  v
  return (v)

}

nr_of_repetitions = 200

#eksperyment L0
{
  ncorr_weak_l0 = function(k) {test_L0Learn0(k, weak_signal, s_ncorr)}
  ncorr_strong_l0 = function(k) {test_L0Learn0(k, strong_signal, s_ncorr)}
  corr_weak_l0 = function(k) {test_L0Learn0(k, weak_signal, s_corr)}
  corr_strong_l0 = function(k) {test_L0Learn0(k, strong_signal, s_corr)}


  l0_res_ncorr_weak = array(0, dim = c(nr_of_repetitions, 4, 6))
  l0_res_ncorr_strong = array(0, dim = c(nr_of_repetitions, 4, 6))
  l0_res_corr_weak = array(0, dim = c(nr_of_repetitions, 4, 6))
```

```
  l0_res_corr_strong = array(0, dim = c(nr_of_repetitions, 4, 6))


  for (i in 1:nr_of_repetitions){
    l0_res_ncorr_weak[i,,] = sapply(values, ncorr_weak_l0)
  }

  for (i in 1:nr_of_repetitions){
    l0_res_ncorr_strong[i,,] = sapply(values, ncorr_strong_l0)
  }

  for (i in 1:nr_of_repetitions){
    l0_res_corr_weak[i,,] = sapply(values, corr_weak_l0)
  }

  for (i in 1:nr_of_repetitions){
    l0_res_corr_strong[i,,] = sapply(values, corr_strong_l0)
  }

  means_ncorr_weak_l0 = matrix(0, 4, 6)
  means_ncorr_strong_l0 = matrix(0, 4, 6)
  means_corr_weak_l0 = matrix(0, 4, 6)
  means_corr_strong_l0 = matrix(0, 4, 6)

  for (i in 1:4){
    for (j in 1:6){
      means_ncorr_weak_l0[i, j] = mean(l0_res_ncorr_weak[,i,j])
      means_ncorr_strong_l0[i, j] = mean(l0_res_ncorr_strong[,i,j])
      means_corr_weak_l0[i, j] = mean(l0_res_corr_weak[,i,j])
      means_corr_strong_l0[i, j] = mean(l0_res_corr_strong[,i,j])
    }
  }

}


test_L0Learnmbic2_0 = function(k, nonzero_coef, s) {
```

```
eps = rnorm(n)
beta = c(rep(nonzero_coef, k), rep(0, p - k))
X = matrix(rmvnorm(n, sigma = s), n, p)
Y = X %*% beta + eps

data = prepare_data(Y, X, verbose = FALSE)
res_mbic2 <- data %>%
  reduce_matrix() %>%
  fast_forward()%>%
  multi_backward(crit=mbic2)%>%
  stepwise(crit=mbic2)

chosen_vars = res_mbic2$model
chosen_vars
mbic2_k = length(chosen_vars)
print(mbic2_k)
fit = L0Learn.fit(X, Y, penalty = "L0", maxSuppSize = 1.5 * mbic2_k + 1)


access_to_sols = cbind(fit$lambda[[1]], fit$gamma[1], fit $suppSize[[1]])

dist_from_correct_p = abs(access_to_sols[,3] - mbic2_k)
best_sol = which.min(dist_from_correct_p)
k_mBIC2_L0 = access_to_sols[best_sol,3]
k_mBIC2_L0
ats = access_to_sols

not_est_beta_yet = coef(fit, lambda = ats[best_sol,1],
gamma = ats[best_sol,2])[-1] #bez interceptu
ind_est_beta_l0 = which(as.vector(not_est_beta_yet) != 0)
est_beta_l0 = numeric(p)
est_beta_l0[ind_est_beta_l0] = as.vector(not_est_beta_yet
[ind_est_beta_l0])

chosen_vars = ind_est_beta_l0
est_FDR = mean(as.integer(chosen_vars) > k)
est_power = sum(as.integer(chosen_vars) <= k) / k
```

```
  est_mse = sum((est_beta_l0 - beta)^2) / sum(beta^2)
  est_msp = sum((X %*% est_beta_l0 - X %*% beta)^2) / sum((X %*% beta)^2)
  v = c(est_FDR, est_power, est_mse, est_msp)
  v
  return (v)

}

nr_of_repetitions = 200

#eksperyment k(mbic2)->L0
{
  ncorr_weak_mbic2l0 = function(k) {test_L0Learnmbic2_0
  (k, weak_signal, s_ncorr)}
  ncorr_strong_mbic2l0 = function(k) {test_L0Learnmbic2_0
  (k, strong_signal, s_ncorr)}
  corr_weak_mbic2l0 = function(k) {test_L0Learnmbic2_0
  (k, weak_signal, s_corr)}
  corr_strong_mbic2l0 = function(k) {test_L0Learnmbic2_0
  (k, strong_signal, s_corr)}


  mbic2l0_res_ncorr_weak = array(0, dim = c(nr_of_repetitions, 4, 6))
  mbic2l0_res_ncorr_strong = array(0, dim = c(nr_of_repetitions, 4, 6))
  mbic2l0_res_corr_weak = array(0, dim = c(nr_of_repetitions, 4, 6))
  mbic2l0_res_corr_strong = array(0, dim = c(nr_of_repetitions, 4, 6))


  for (i in 1:nr_of_repetitions){
    mbic2l0_res_ncorr_weak[i,,] = sapply(values, ncorr_weak_mbic2l0)
  }

  for (i in 1:nr_of_repetitions){
    mbic2l0_res_ncorr_strong[i,,] = sapply(values, ncorr_strong_mbic2l0)
  }

  for (i in 1:nr_of_repetitions){
    mbic2l0_res_corr_weak[i,,] = sapply(values, corr_weak_mbic2l0)
```

```
  }

  for (i in 1:nr_of_repetitions){
    mbic2l0_res_corr_strong[i,,] = sapply(values, corr_strong_mbic2l0)
  }


  means_ncorr_weak_mbic2l0 = arr_means(mbic2l0_res_ncorr_weak)
  means_ncorr_strong_mbic2l0 = arr_means(mbic2l0_res_ncorr_strong)
  means_corr_weak_mbic2l0 = arr_means(mbic2l0_res_corr_weak)
  means_corr_strong_mbic2l0 = arr_means(mbic2l0_res_corr_strong)

}


{
par(mfrow = c(2,2))
par(mar = c(3,3,3,3))
FDR_ncorr_weak = rbind(means_ncorr_weakx[1,], means_ncorr_weak_mbic2l0[1,],
means_ncorr_weak_l0[1,])
matplot(c(10, 20, 40, 60, 80, 100), t(FDR_ncorr_weak),
xlab = "", ylab = "",
ylim = c(0,1), main = "FDR_NoCorr_Weak", pch = 19, type = 'o', col = cl)
legend('topleft', legend2, pch = 19, col = cl)

FDR_corr_weak = rbind(means_corr_weakx[1,], means_corr_weak_mbic2l0[1,],
means_corr_weak_l0[1,])
matplot(c(10, 20, 40, 60, 80, 100), t(FDR_corr_weak),
xlab = "", ylab = "",
main = "FDR_Corr_Weak", pch = 19,ylim = c(0,1), type = 'o', col = cl)
legend('topleft', legend2, pch = 19, col = cl)

FDR_ncorr_strong = rbind(means_ncorr_strongx[1,],
means_ncorr_strong_mbic2l0[1,], means_ncorr_strong_l0[1,])
matplot(c(10, 20, 40, 60, 80, 100), t(FDR_ncorr_strong),
xlab = "", ylab = "",
main = "FDR_NoCorr_Strong",ylim = c(0,1), pch = 19, type = 'o', col = cl)
legend('topleft', legend2, pch = 19, col = cl)
```

```
FDR_corr_strong = rbind(means_corr_strongx[1,],
means_corr_strong_mbic2l0[1,], means_corr_strong_l0[1,])
matplot(c(10, 20, 40, 60, 80, 100), t(FDR_corr_strong),
xlab = "", ylab = "",
main = "FDR_Corr_Strong", ylim = c(0,1), pch = 19, type = 'o', col = cl)
legend('topleft', legend2, pch = 19, col = cl)

power_ncorr_weak = rbind(means_ncorr_weakx[2,],
means_ncorr_weak_mbic2l0[2,], means_ncorr_weak_l0[2,])
matplot(c(10, 20, 40, 60, 80, 100), t(power_ncorr_weak), xlab = "",
ylab = "", main = "Power_NoCorr_Weak", ylim = c(0,1),
pch = 19, type = 'o', col = cl)
legend('bottomleft', legend2, pch = 19, col = cl)

power_corr_weak = rbind(means_corr_weakx[2,],
means_corr_weak_mbic2l0[2,], means_corr_weak_l0[2,])
matplot(c(10, 20, 40, 60, 80, 100), t(power_corr_weak), xlab = "",
ylab = "", main = "Power_Corr_Weak", ylim = c(0,1),
pch = 19, type = 'o', col = cl)
legend('bottomleft', legend2, pch = 19, col = cl)

power_ncorr_strong = rbind(means_ncorr_strongx[2,],
means_ncorr_strong_mbic2l0[2,], means_ncorr_strong_l0[2,])
matplot(c(10, 20, 40, 60, 80, 100), t(power_ncorr_strong), xlab = "",
ylab = "", main = "Power_NoCorr_Strong", ylim = c(0,1),
pch = 19, type = 'o', col = cl)
legend('bottomleft', legend2, pch = 19, col = cl)

power_corr_strong = rbind(means_corr_strongx[2,],
means_corr_strong_mbic2l0[2,], means_corr_strong_l0[2,])
matplot(c(10, 20, 40, 60, 80, 100), t(power_corr_strong), xlab = "",
ylab = "", main = "Power_Corr_Strong", ylim = c(0,1),
pch = 19, type = 'o', col = cl)
legend('bottomleft', legend2, pch = 19, col = cl)

MSE_ncorr_weak = rbind(means_ncorr_weakx[3,],
means_ncorr_weak_mbic2l0[3,], means_ncorr_weak_l0[3,])
```

```
matplot(c(10, 20, 40, 60, 80, 100), t(MSE_ncorr_weak), xlab = "",
ylab = "", main = "MSE_NoCorr_Weak", pch = 19, type = 'o',
ylim = c(0,5), col = cl)
legend('topleft', legend2, pch = 19, col = cl)

MSE_corr_weak = rbind(means_corr_weakx[3,],
means_corr_weak_mbic2l0[3,], means_corr_weak_l0[3,])
matplot(c(10, 20, 40, 60, 80, 100), t(MSE_corr_weak), xlab = "",
ylab = "", main = "MSE_Corr_Weak", pch = 19,ylim = c(0,5),
type = 'o', col = cl)
legend('topleft', legend2, pch = 19, col = cl)

MSE_ncorr_strong = rbind(means_ncorr_strongx[3,],
means_ncorr_strong_mbic2l0[3,], means_ncorr_strong_l0[3,])
matplot(c(10, 20, 40, 60, 80, 100), t(MSE_ncorr_strong), xlab = "",
ylab = "", main = "MSE_NoCorr_Strong", pch = 19,ylim = c(0,5),
type = 'o', col = cl)
legend('topleft', legend2, pch = 19, col = cl)

MSE_corr_strong = rbind(means_corr_strongx[3,],
means_corr_strong_mbic2l0[3,], means_corr_strong_l0[3,])
matplot(c(10, 20, 40, 60, 80, 100), t(MSE_corr_strong), xlab = "",
ylab = "", main = "MSE_Corr_Strong", pch = 19,
ylim = c(0,5), type = 'o', col = cl)
legend('topleft', legend2, pch = 19, col = cl)

MSP_ncorr_weak = rbind(means_ncorr_weakx[4,],
means_ncorr_weak_mbic2l0[4,], means_ncorr_weak_l0[4,])
matplot(c(10, 20, 40, 60, 80, 100), t(MSP_ncorr_weak), xlab = "",
ylab = "",ylim = c(0,1), main = "MSP_NoCorr_Weak",
pch = 19, type = 'o', col = cl)
legend('topleft', legend2, pch = 19, col = cl)

MSP_corr_weak = rbind(means_corr_weakx[4,],
means_corr_weak_mbic2l0[4,], means_corr_weak_l0[4,])
matplot(c(10, 20, 40, 60, 80, 100), t(MSP_corr_weak), xlab = "",
ylab = "",ylim = c(0,1), main = "MSP_Corr_Weak",
pch = 19,type = 'o', col = cl)
```

```
legend('topleft', legend2, pch = 19, col = cl)

MSP_ncorr_strong = rbind(means_ncorr_strongx[4,],
means_ncorr_strong_mbic2l0[4,], means_ncorr_strong_l0[4,])
matplot(c(10, 20, 40, 60, 80, 100), t(MSP_ncorr_strong), xlab = "",
ylab = "",ylim = c(0,1), main = "MSP_NoCorr_Strong",
pch = 19, type = 'o', col = cl)
legend('topleft', legend2, pch = 19, col = cl)

MSP_corr_strong = rbind(means_corr_strongx[4,],
means_corr_strong_mbic2l0[4,], means_corr_strong_l0[4,])
matplot(c(10, 20, 40, 60, 80, 100), t(MSP_corr_strong), xlab = "",
ylab = "",ylim = c(0,1), main = "MSP_Corr_Strong", pch = 19,
type = 'o', col = cl)
legend('topleft', legend2, pch = 19, col = cl)

}


library(bigstep)
library(L0Learn)
library('Rcpp')

load("Sangerdata.Rdata")

y = as.numeric(data[24266,-1])
x = t(as.matrix(data[-24266,-1]))

quan = quantile(as.vector(x), probs = 0.25)
out1 = which(apply(x, 2, max) < quan)
length(out1)
x1 = x[, -out1]

range = apply(x1, 2, max) - apply(x1, 2, min)
out2 = which(range < 2)
length(out2)
x2 = x1[, -out2]
xx = x2
```

```
p = dim(xx)[2]
n = dim(xx)[1]
X = as.matrix(xx)
X = matrix(X, ncol = ncol(X), dimnames = NULL)
colnames(X) = colnames(xx)
rownames(X) = rownames(xx)

y = scale(y)
X = scale(X)/sqrt(n)

results.CD = L0Learn.fit(X, y, loss = "SquaredError", penalty = "L0",
                         algorithm = "CD", maxSuppSize = 30,
                         nLambda = 200, nGamma = 10, gammaMax = 10,
                         gammaMin = 1e-03, intercept = TRUE)

cvfit = L0Learn.cvfit(X, y, loss = "SquaredError", penalty = "L0",
                        nFolds = 10, algorithm = "CD", maxSuppSize = 30,
                        nLambda = 200, nGamma = 10, gammaMax = 10,
                        gammaMin = 1e-03, intercept = FALSE)

res.CD = print(results.CD)
res.CD

optimalLambdaIndex = which.min(as.vector(cvfit$cvMeans[[1]]))
optimalLambdaIndex
optimalLambda = cvfit$fit$lambda[[1]][optimalLambdaIndex]
optimalLambda

vcv = as.vector(coef(cvfit, lambda=optimalLambda, gamma=0))
significant = which(vcv != 0)
significant
vcv[significant]

Xdwa = X[,significant]#via L0Learn
summary(lm(y ~ Xdwa))
#str(lm(y ~ Xdwa))
sum(lm(y ~ Xdwa)$residuals^2)
cor(cbind(y, Xdwa))
```

```
Xjeden = X[,c(206, 682, 1004, 1354, 1370)]#via mbic2
summary(lm(y ~ Xjeden))
sum(lm(y ~ Xjeden)$residuals^2)
cor(cbind(y, Xjeden))
```

Literatura

[1] Bertsimas, D., King, A., and Mazumder, R. Best subset selection via a modern optimization lens. *Ann. Statist. 44*, 2 (2016), 813–852.

[2] Bogdan, M., and Frommlet, F. Identifying importatn predictors in large data bases - multiple testing and model selection.

[3] Bogdan, M., Frommlet, F., Biecek, P. a., Cheng, R., Ghosh, J. K., and Doerge, R. W. Extending the modified Bayesian information criterion (mBIC) to dense markers and multiple interval mapping. *Biometrics 64*, 4 (2008), 1162–1169.

[4] Donoho, D. L., and Elad, M. Optimally sparse representation in general (nonorthogonal) dictionaries via l^1 minimization. *Proc. Natl. Acad. Sci. USA 100*, 5 (2003), 2197–2202.

[5] Nesterov, Y. *Introductory lectures on convex optimization*, vol. 87 of *Applied Optimization*. Kluwer Academic Publishers, Boston, MA, 2004. A basic course.

[6] Nesterov, Y. Gradient methods for minimizing composite functions. *Math. Program. 140*, 1, Ser. B (2013), 125–161.

[7] Tropp, J. A. Just relax: convex programming methods for identifying sparse signals in noise. *IEEE Trans. Inform. Theory 52*, 3 (2006), 1030–1051.