Wydział Matematyki i Nauk Informacyjnych
Politechniki Warszawskiej

Metody optymalizacji w analizie danych

Integracja metody LARS z pakietem ROI i analiza jej wydajności

Kosma Grochowski, Szymon Kowalik

Spis treści

1 Wstęp

Niniejszy raport stanowi podsumowanie prac wykonanych w ramach projektu na przedmiot Metody optymalizacji. Celem projektu była rejestracja metody *Least-angle regression* (LARS) w infrastrukturze pakietu *R Optimization Infrastructure* (ROI) oraz weryfikacja skuteczności i wydajności tej metody w zastosowaniu do zadania optymalizacji (1) przy wykorzystaniu nietrywialnego (dużego) zbioru danych.

$$\min_{\beta}\left[\frac{1}{2}\|y - X\beta\|_2^2 + \lambda\|\beta\|_1\right] \tag{1}$$

gdzie $X \in \mathbb{R}^{m\times n}, y \in \mathbb{R}^m, \beta \in \mathbb{R}^n, 0 < \lambda \in \mathbb{R}$

1.1 R Optimization Infrastructure

R Optimization Infrastructure (ROI) [4] to pakiet języka R, który dostarcza spójny interfejs dostępu do wielu algorytmów stanowiących implementację różnych metod optymalizacyjnych. Podstawowym założeniem twórców tego pakietu było stworzenie infrastruktury pozwalającej na łatwe rozszerzanie bazy dostępnych metod przy zachowaniu spójności w dostępie do poszczególnych solverów. W związku z tym pakiet definiuje obiekt reprezentujący problem optymalizacyjny: liniowy, kwadratowy, stożkowy (ang. *conic*) lub nieliniowy wraz ze wszystkimi jego parametrami, takimi jak: funkcja celu, jej typ, ograniczenia zadania oraz jego dziedzina.

Biblioteka dostarcza wiele gotowych solverów, a także możliwość rejestracji nowych przez użytkownika przy pomocy specjalnych funkcji. Omawiana niżej metoda LARS nie jest domyślnie zaimplementowana w ROI. W związku z tym zachodzi potrzeba ręcznej integracji tej metody z ROI.

1.2 Least-angle regression

Least-angle regression (LARS) to metoda dopasowywania modeli regresji liniowej do danych o dużej liczbie wymiarów, stworzona przez Bradleya Efrona, Trevora Hastiego, Iaina Johnstone'a oraz Roberta Tibshiraniego [2].

1.2.1 Działanie algorytmu

Pierwszym krokiem algorytmu jest znalezienie predyktora o najwyższym współczynniku korelacji z wynikiem. Współczynnik tego predyktora jest stopniowo zwiększany, a z różnicy pomiędzy wynikiem faktycznym, a podawanym przez algorytm liczone są wartości

rezydualne. Postępowanie to należy kontynuować do momentu, w którym inny predyktor osiąga najwyższy ze wszystkich współczynnik korelacji z wartością rezydualną. W tym momencie, należy zwiększać oba predyktory w kierunku największego spadku rezydualnego, dopóki kolejny predyktor nie osiągnie największej korelacji z wartością rezydualną. Algorytm w analogiczny sposób dodaje wszystkie predyktory do modelu [5].

1.2.2 Powiązanie z LASSO

Okazuje się, że jeżeli do powyższego algorytmu wprowadzona zostanie następująca modyfikacja:

> jeżeli współczynnik niezerowego predyktora osiągnie wartość 0, to ten predyktor jest usuwany ze zbioru aktywnych predyktorów i kierunek największego spadku obliczany jest ponownie,

to tak zmodyfikowany algorytm generuje rozwiązanie LASSO jednocześnie dla wszystkich wartości lambda [5].

2 Integracja pakietu LARS z ROI

Jak opisano w [3] problem (1) można sprowadzić do kwadratowego problemu optymalizacyjnego

$$\begin{aligned} &\min_{(\beta,\gamma,t)} \left[\frac{1}{2}\gamma^T\gamma + \lambda \mathbf{1}^T t \right] \\ \text{przy ograniczeniach } & y - X\beta = \gamma \\ & -t \leq \beta \leq t \end{aligned} \tag{2}$$

gdzie $\gamma \in \mathbb{R}^n, t \in \mathbb{R}^n$.

Sprowadzenie problemu do postaci (2) pozwala na zapisanie go w formie obiektu problemu optymalizacyjnego ROI przy wykorzystaniu funkcji [3]:

- `Q_objective` definiującej kwadratową funkcję celu $\min_{(\beta,\gamma,t)}\left[\frac{1}{2}\gamma^T\gamma + \lambda \mathbf{1}^T t\right]$,
- `L_constraint` określającej liniowe ograniczenia $X\beta+\gamma = y$, $0 \leq \beta+t$ oraz $\beta-t \leq 0$,
- `V_bound` ustalającej dziedzinę zadania na $(-\infty, \infty)$.

Należy jednak pamiętać, że funkcja z pakietu LARS, która ma zostać zarejestrowana w ROI jako solver, na swoje wejście przyjmuje X, y oraz λ z wyrażenia (1), nie zaś

obiekt kwadratowego problemu optymalizacyjnego. Fakt ten sprawia, że konieczne jest przygotowanie odwrotnego przekształcenia z postaci (2) do postaci (1) (a dokładniej wystarczy uzyskać macierz X, wektor y oraz wartość λ).

Nie stanowi to jednak problemu od strony implementacyjnej ze względu na to, że wynikami działania funkcji `Q_objective` oraz `L_constraint` są macierze rzadkie, których konkretne podmacierze zawierają potrzebne wartości.

2.1 Możliwości dalszego rozwoju rozwiązania

Aktualnie konwersja z obiektu kwadratowego problemu optymalizacyjnego do X, y oraz λ została zrealizowana z założeniem konkretnej reprezentacji wewnętrznej problemu kwadratowego w macierzach opisujących funkcję celu oraz ograniczenia. Budowa tych macierzy została wykonana za pomocą opisanych wyżej funkcji, a cały ten proces został zamknięty w ramach funkcji `qp_lasso`.

Jedną z możliwości rozwoju istniejącego aktualnie rozwiązania jest zwiększenie dowolności w konstrukcji reprezentacji problemu optymalizacyjnego, co wiązałoby się z bardziej skomplikowaną analizą struktur omawianych macierzy. Uzyskanie uniwersalnego narzędzia nie było jednak celem niniejszego projektu, dlatego autorzy pozostawiają ten aspekt otwarty.

Kolejną możliwością rozszerzenia projektu jest rejestracja analogicznego solvera dla reprezentacji problemu (1) w postaci stożkowej (analogicznie – z konwersją z tej postaci do X, y oraz λ podawanych na wejściu dla metody LARS).

3 Wydajność omawianej metody

Drugim celem niniejszego projektu – po integracji pakietu LARS z ROI – była analiza porównawcza wydajności metody LARS. Wydajność pakietu LARS dla konkretnego, opisanego wyżej zadania minimalizacji została porównana z wydajnością pakietu qpOASES [**qpoases**], zdolnego do rozwiązania dowolnych problemów programowania kwadratowego o następującej formie:

$$\begin{aligned} &\min_{(x)} \left[\frac{1}{2} x^T H x + x^T g \right] \\ &\text{przy ograniczeniach } \mathrm{lb} \leq x \leq \mathrm{ub} \\ &\mathrm{lb}A \leq Ax \leq \mathrm{ub}A \end{aligned} \tag{3}$$

3.1 Zbiór diabetyków

W oryginalnej pracy [2], wyniki metody LARS przedstawiono na przykładzie zbioru 442 pomiarów pacjentów. Pacjentom zmierzono wiek, płeć, współczynnik BMI, ciśnienie krwi oraz sześć czynników we krwi, a także miarę postępu choroby w ciągu roku od przeprowadzenia pomiarów.

Dla współczynnika $\lambda = 50$:

Solver	**qpOASES**	**LARS**
Długość działania [s]	10.268	0.046
Uzyskane minimum	5 821 850	5 821 850

Tabela 1: Zestawienie wyników działania metod qpOASES i LARS dla zbioru diabetyków

Z wyników z tabeli 1 wynika jasno, że z problemem postawionym w formie (2) LARS radzi sobie zdecydowanie szybciej, niż metoda qpOASES. Jednocześnie zachowana jest poprawność rozwiązania - oba rozwiązania osiągają to samo minimum. Powyższa teza jest potwierdzona z pomocą analizy uzyskanych współczynników minimalizujących funkcję – obie metody uzyskały te same współczynniki z dokładnością do błędu numerycznego.

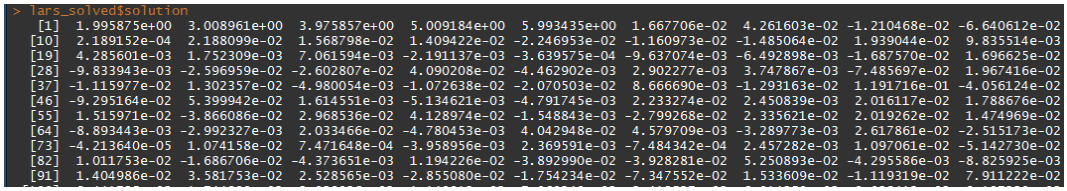
3.2 Zbiory nietrywialne

Jak wspomniano w podrozdziale 1.2, metoda LARS sprawdza się szczególnie dobrze dla danych o dużej liczbie wymiarów – została zaprojektowana z myślą o takim typie zbiorów. Poniższe podrozdziały – 3.2.1 i 3.2.2 – weryfikują jej wydajność w problemach o dużych rozmiarach w porównaniu z metodą qpOASES.

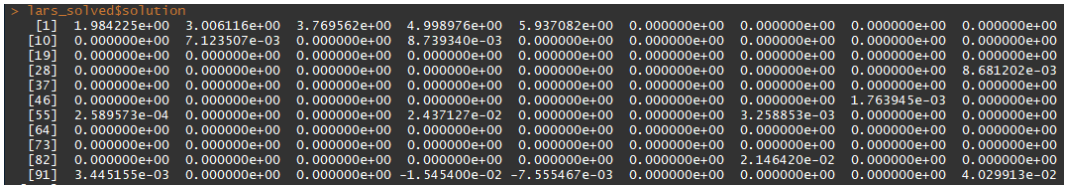
3.2.1 Zbiór generowany

W celu przetestowania poprawności działania metody na trudniejszym zbiorze, stworzony został sztuczny zbiór zawierający 10 000 próbek oraz 1000 predyktorów, spośród których zaledwie pierwsze 5 jest skorelowanych z wynikiem. Wynik został zaszumiony w celu utrudnienia zadania.

Zadanie rozwiązano dla wartości współczynnika $\lambda = 0$ oraz $\lambda = 2500$. W obu przypadkach rozwiązanie zadania metodą LARS trwało około 30 sekund.



Rysunek 1: Przedstawienie pierwszych 99 współczynników predyktorów dla $\lambda = 0$



Rysunek 2: Przedstawienie pierwszych 99 współczynników predyktorów dla $\lambda = 2500$

Rysunki 1 oraz 2 ilustrują działanie metody LARS.

Dla wartości współczynnika $\lambda = 0$, metoda jest równoważna z metodą regresji liniowej. Rozwiązanie pierwszych pięciu współczynników jest bliskie oczekiwanemu, jednakże metoda ulega zjawisku *overfittingu*, przez co nieskorelowanym predyktorom przypisano niezerowe współczynniki.

Dla wartości współczynnika $\lambda = 2500$, metoda wyzerowuje zdecydowaną większość nieskorelowanych predyktorów, jednocześnie zachowując te istotne. Rozwiązanie otrzymane w ten sposób jest zdecydowanie bardziej dopasowane do faktycznych korelacji i ignoruje ono większość szumu.

3.2.2 Million Song Dataset

Opisywany solver został także przetestowany na podzbiorze istniejącego zbioru *Million Song Dataset* [1]. Jest to zbiór składający się z ponad 500 000 próbek oraz 90 predyktorów. Zmienną objaśnianą jest rok wydania danego utworu muzycznego, a zmiennymi objaśniającymi są wartości liczbowe określające barwę dźwięku.

Zbiór został wybrany ze względu na pokaźną liczbą próbek oraz jedynie liczbowe wartości wszystkich predyktorów, co wykluczyło konieczność realizacji wstępnego przetwarzania danych.

Solver	qpOASES	LARS
Długość działania [s]	–	52.218
Uzyskane minimum	–	4 855 338 257

Tabela 2: Zestawienie wyników działania metod qpOASES i LARS dla zbioru Million Song Dataset. Wyniki metodą qpOASES nie zostały uzyskane ze względu na konieczność alokacji 1.98 TB pamięci, co nie było możliwe na dostępnym sprzęcie.

Tabela 2 prezentuje wyniki działania metody LARS dla współczynnika $\lambda = 0$. Jak widać, czas działania tej metody wydłużył się do około minuty, ale nadal pozostaje w akceptowalnym zakresie. Metoda qpOASES nie znalazła zastosowania dla tak obszernego zbioru danych ze względu na próbę alokacji gigantycznego obszaru pamięci roboczej. Taki problem nie dotknął metody LARS.

Zadanie rozwiązano dla wartości współczynnika $\lambda = 0$ oraz $\lambda = 10e8$. W obu przypadkach czas trwania rozwiązania zadania metodą LARS był podobny.

```
> lars_solved$solution
 [1] 37.3559561521 -1.6492771718 -0.0764143876  0.0260697824 -0.4862708458 -4.3935856715 -2.3670519570 -0.0641517548 -2.5538759843
[10]  2.2322219931 -7.9036106368  3.6006216701  1.1516981402  0.0121752567  0.0217104520  0.0011421394  0.0239042479  0.0761736594
[19]  0.0515965569 -0.0232086561  0.0727317738  0.0985964690  0.0008668678  0.1886522000 -0.2886676595  0.0005560200 -0.0128450373
[28]  0.0125541121  0.0126541251  0.0432787295  0.0480023564 -0.0003870457  0.0071250983  0.1168424827  0.1246476924  0.0115271259
[37] -0.0133575340 -0.0099549843  0.0110775638 -0.0036024597  0.0585208117  0.0285495880  0.0514585387  0.0016100830 -0.0444386952
[46]  0.2698362827  0.0011606104 -0.0349459764  0.0002122285  0.0499779112  0.0692937997  0.0214057775 -0.0478259350 -0.0028348575
[55]  0.0602524214 -0.0070885328 -0.0323221190 -0.0136372152  0.0301034178  0.0446487951 -0.0131394446 -0.0524214290  1.1118005321
[64]  0.0227850008 -0.0268747435 -0.0159599035 -0.0805296612 -0.0102387241 -0.0528189120 -0.0316184212 -0.0144420883 -0.0195038765
[73]  0.0159409156 -0.0602471592 -0.0063217442  0.5778799256  0.0327854095  0.0483628772  0.0389492687 -0.0055737319  0.5107689048
[82] -0.0323711377 -0.0849152856 -0.0599779835 -0.4040065907  0.1074068547  0.0091730145  0.9893902110 -0.0106387498  0.0351111141
```

Rysunek 3: Przedstawienie predyktorów dla $\lambda = 0$

```
> lars_solved$solution
 [1] 37.8422134050 -1.2455856329 -0.0963855568 -0.8958187010  0.0000000000 -0.7729077892  0.0000000000  0.0000000000  0.0000000000
[10]  0.0000000000  0.0000000000  0.0000000000  1.0075649415  0.0189157376  0.0150267264  0.0055158094  0.0322303557  0.0756651841
[19]  0.0464350044  0.0000000000  0.1159091095  0.0000000000  0.0000000000  0.2166998300 -0.3210318660  0.0000000000  0.0000000000
[28]  0.0000000000  0.0002353114  0.0138865961  0.0099284308  0.0000000000  0.0000000000  0.0000000000  0.0341434195  0.1100487605
[37] -0.0081218971 -0.0005829536  0.0122885031  0.0000000000  0.0000000000  0.0000000000  0.0000000000  0.0000000000  0.0000000000
[46]  0.2977076910  0.0001587481 -0.0008093558  0.0000000000  0.0474093119  0.0287685051  0.0577611153 -0.0503492363  0.0000000000
[55]  0.0000000000 -0.0255900964 -0.0202475280  0.0071626587  0.0000000000  0.0314986619 -0.0281474739 -0.0347120242  0.7506768756
[64]  0.0194019596  0.0000000000  0.0000000000 -0.0231306031 -0.0102399615 -0.0384741738 -0.0386408378 -0.0016141190  0.0023216355
[73]  0.0288944459 -0.0211423299  0.0000000000  0.0000000000  0.0431383444  0.0089855693  0.0408852188 -0.0206150917  0.1685361412
[82] -0.0589551274 -0.0193331268 -0.0459084594  0.0000000000  0.0516776937  0.0000000000  0.0000000000 -0.0092742202  0.0000000000
```

Rysunek 4: Przedstawienie predyktorów dla $\lambda = 10e8$

Rysunki 3 oraz 4 ilustrują działanie metody LARS. Podobnie jak w przypadku zbioru generowanego – dla dużej wartości parametru λ metoda wykrywa część predyktorów jako nieskorelowane ze zmienną objaśnianą.

4 Podsumowanie

W ramach niniejszego projektu z powodzeniem zarejestrowano istniejącą metodę LARS jako solver w pakiecie ROI. Ze względu na różny sposób opisu problemu optymalizacyjnego w pakietach LARS oraz ROI, nie było to zadanie trywialne polegające na prostym

mapowaniu funkcji. Konieczne było zastosowanie konwersji między obiektem reprezentującym kwadratowy problem optymalizacyjny a parametrami wejściowymi metody LARS - macierzą predyktorów, wektorem wartości zmiennej objaśnianej oraz parametrem λ.

Przeprowadzona analiza wydajności rozpatrywanego algorytmu wykazuje, że jest on szczególnie skuteczny dla nietrywialnych, bardzo obszernych zbiorów danych, przy których inne metody nie są w stanie zakończyć obliczeń ze względu na zapotrzebowanie pamięciowe lub zbyt dużą złożoność czasową.

Bibliografia

1. T. Bertin-Mahieux, D. P. Ellis, B. Whitman i P. Lamere. „The Million Song Dataset". W: *Proceedings of the 12th International Conference on Music Information Retrieval (ISMIR 2011)*. 2011. URL: https://archive.ics.uci.edu/ml/datasets/YearPredictionMSD (term. wiz. 26. 01. 2020).
2. B. Efron, T. Hastie, I. Johnstone i R. Tibshirani. „Least angle regression". *Ann. Statist.* 32:2, 2004, s. 407–499. DOI: 10.1214/009053604000000067. URL: https://doi.org/10.1214/009053604000000067.
3. *Lasso - ROI Use Case*. URL: http://roi.r-forge.r-project.org/use_case_lasso.html (term. wiz. 26. 01. 2020).
4. S. Theußl, F. Schwendinger i K. Hornik. *ROI: The R Optimization Infrastructure Package*. Research Report Series / Department of Statistics and Mathematics 133. Vienna: WU Vienna University of Economics i Business, 2017. URL: http://epub.wu.ac.at/5858/.
5. R. Tibshirani. *A simple explanation of the Lasso and Least Angle Regression*. URL: http://statweb.stanford.edu/~tibs/lasso/simple.html?fbclid=IwAR1EIaz5qZIpz074edwDoHYmNnh_VSZAy3SIrp_vXGG2vr8NZYaudOKyvxE.