Politechnika Warszawska
Wydział Matematyki i Nauk Informacyjnych

PRACA DYPLOMOWA MAGISTERSKA
NA KIERUNKU MATEMATYKA
SPECJALNOŚĆ STATYSTYKA MATEMATYCZNA I ANALIZA DANYCH

ESTYMACJA W MODELU COXA METODĄ STOCHASTYCZNEGO SPADKU GRADIENTU Z PRZYKŁADAMI ZASTOSOWAŃ W ANALIZIE DANYCH Z THE CANCER GENOME ATLAS

Autor:
Marcin Piotr Kosiński

Promotor:
prof. ndzw. dr hab. inż Przemysław Biecek

Warszawa, lipiec 2015

.....

podpis promotora

.....

podpis autora

Spis treści

Wprowadzenie **5**

Podstawy modelu statystycznego **7**

1. Estymacja metodą największej wiarogodności **9**
1.1. Estymacja 9
1.2. Metoda największej wiarogodności 11
1.3. Asymptotyczne własności estymatora największej wiarogodności 11

2. Model Coxa **17**
2.1. Wprowadzenie do modelu Coxa i nomenklatura 17
2.2. Założenia modelu proporcjonalnego ryzyka Coxa 18
2.3. Estymacja w modelu Coxa 20

3. Numeryczne metody estymacji **23**
3.1. Algorytmy spadku wzdłuż gradientu 24
3.2. Algorytm stochastycznego spadku wzdłuż gradientu I 25
3.3. Porównanie algorytmów spadku wzdłuż gradientu 26

4. Estymacja w modelu Coxa metodą stochastycznego spadku gradientu **29**
4.1. Porównanie z innymi modelami 30
4.2. Implementacja 30
4.3. Dwuwymiarowy przykład 34
4.4. Porównanie z innymi algorytmami spadku gradientu 34

5. Analiza danych genomicznych **35**
5.1. Opis i pobranie danych 35
5.2. Analiza 35

A. Wykorzystane narzędzia **37**

B. Kody w R **39**

C. Dokumentacja pakietu RTCGA **41**

Literatura **53**

Wprowadzenie

+++ Analiza przeżycia. +++

Najbardziej charakterystyczną cechą typowych danych, jakimi posługuje się w analizie przeżycia, jest obecność obiektów, w których końcowe zdarzenie nastąpiło (wówczas ma się do czynienia z obserwacjami *kompletnymi*), oraz obiektów, w których to zdarzenie (jeszcze) nie nastąpiło (obserwacja *ucięta*). Ta specyficzna postać danych statystycznych doprowadziła do powstania specjalnych metod stosowanych tylko w analizie czasu trwania zjawisk. Jednym z takich modeli jest model proporcjonalnych hazardów Coxa. Jak podaje [2], model proporcjonalnych hazardów Coxa jest jednym z najszerzej stosowanych modeli w onkologicznych publikacjach naukowych, ale także jedną z najmniej rozumianych metod statystycznych. Wynika to z łatwego dostępu do pakietów statystycznych zawierających programy do analizy przeżyć, modeli regresji i analiz wielowariantowych, ale prawie nigdy nie zawierających dobrego opisu podstawowych zasad działania modelu Coxa. Dostarczają one wyłącznie instrukcje, jak wprowadzić dane i uruchomić odpowiednie procedury w celu uzyskania wyniku. Poniższy praca zawiera pełny opis metodologii modelu proporcjonalnych hazardów Coxa, w tym wyjaśnienie najważniejszych pojęć.

++++ Stochastyczny Spadek Gradientu ++++ W przeciągu ostatniej dekady, rozmiary danych rosły szybciej niż prędkość procesorów. W tej sytuacji możliwości statystycznych metod uczenia maszynowego stały się ograniczone bardziej przez czas obliczeń niż przez rozmiary zbiorów danych. Jak podaje [4], bardziej szczegółowa analiza wykazuje jakościowo różne kompromisy w przypadkach problemów uczenia maszynowego na małą i na dużą skalę. Rozwiązania kompromisowe w przypadku dużej skali danych związane są ze złożonością obliczeniową zasadniczych algorytmów optymalizacyjnych, których należy dokonywać w nietrywialny sposób. Jednym z takich rozwiązań są algorytmy optymalizacyjne oparte o stochastyczny spadek gradientu, które wykazują niesamowitą wydajność dla problemów wielkiej skali.

Podstawy modelu statystycznego

W pracy zakłada się znajomość podstaw statystyki matematycznej. Aby ujednolicić oznaczenia w niniejszym rozdziale wprowadzona została klasyczna nomenklatura oparta o [27].

Definicja 0.1. ***Model statystyczny*** *określamy przez podanie rodziny* $\{\mathbb{P}_\theta : \theta \in \Theta\}$ *rozkładów prawdopodobieństwa na przestrzeni próbkowej* Ω *oraz zmiennej losowej* $X : \Omega \to \mathcal{X}$*, którą traktujemy jako obserwację. Zbiór* $\mathcal{X}$ *nazywamy przestrzenią obserwacji, zaś* Θ *nazywamy przestrzenią parametrów.*

Symbol θ jest nieznanym parametrem opisującym rozkład badanego zjawiska. Może być jednowymiarowy lub wielowymiarowy. Determinując opis zjawiska poprzez podanie parametru θ, jednoznacznie wyznaczany jest rozkład rozważanego zjawiska spośród całej rodziny rozkładów prawdopodobieństwa $\{\mathbb{P}_\theta : \theta \in \Theta\}$, co umożliwia określenie prawdziwości tezy.

Zakłada się, że przestrzeń próbkowa Ω jest wyposażona w σ-ciało $\mathcal{F}$. Wtedy:

Definicja 0.2. ***Przestrzenią statystyczną*** *nazywa się trójkę* $(\mathcal{X}, \mathcal{F}, \{\mathbb{P}_\theta : \theta \in \Theta\})$.

Wprowadzenie σ-ciała $\mathcal{F}$ sprawia, że przestrzeń statystyczna staje się przestrzenią mierzalną, a więc można na niej określić rodzinę miar $\{\mathbb{P}_\theta : \theta \in \Theta\}$, dzięki której da się ustalić prawdopodobieństwa zajścia wszystkich zjawisk w rozważanej teorii.

W celu budowania niezbędnych pojęć potrzebna jest również definicja losowej próby statystycznej, zazwyczaj nazywanej *próbką*.

Definicja 0.3. ***Losową próba statystyczną*** *nazywamy zbiór obserwacji statystycznych wylosowanych z populacji, które są realizacjami ciągu zmiennych losowych o rozkładzie takim jak rozkład populacji.*

Rozdział 1

Estymacja metodą największej wiarogodności

The making of maximum likelihood was one of
the most important developments in 20th century statistics.
It was the work of one man but it was no simple process (...).
John Aldrich o R. A. Fisher'ze

1.1. Estymacja

Estymacja to dział wnioskowania statystycznego będący zbiorem metod pozwalających na uogólnianie wyników badania próby losowej na nieznaną postać i parametry rozkładu zmiennej losowej całej populacji oraz szacowanie błędów wynikających z tego uogólnienia [42].

W statystyce parametrycznej zakłada się, że rozkład prawdopodobieństwa opisujący doświadczenie należy do rodziny $\{\mathbb{P}_\theta : \theta \in \Theta\}$, ale nie zna się parametru θ. Można go jednak szacować dzięki estymatorom opartym na statystykach.

Definicja 1.1. ***Statystyka****, dla $X = (X_1, \ldots, X_n)$, to odwzorowanie mierzalne $T : \mathcal{X} \to \mathcal{R}$.*

Definicja 1.2. ***Estymatorem*** *parametru θ nazywamy dowolną statystykę $T = T(X)$, gdzie X to próba z badanego rozkładu, o wartościach w zbiorze Θ.*

Interpretuje się T jako przybliżenie θ i często estymator θ oznacza symbolem $\hat{\theta}$. Niekiedy w kręgu zainteresowań jest również estymacja $g(\theta)$, gdzie g to ustalona funkcja.

Pewne estymatory mające odpowiednie własności są preferowane nad inne ze względu na większą precyzję bądź ufność oszacowania danego estymatora. Poniżej przedstawione są 2 ważne definicje związane z jakością estymatorów [27], gdy rozmiar próbki $X_1, \ldots, X_n$ jest duży. Mówi się wtedy o własnościach asymptotycznych estymatorów, które z matematycznego punktu widzenia są twierdzeniami granicznymi, w których n dąży do nieskończoności. Dzięki tym twierdzeniom możliwe jest opisane w przybliżeniu zachowania estymatorów dla dostatecznie dużych próbek. Niestety, teoria asymptotyczna nie dostarcza informacji o tym, jak duża powinna być próbka, żeby przybliżenie było dostatecznie dobre.

Definicja 1.3. *Estymator $\hat{g}(X_1, \ldots, X_n)$ wielkości $g(\theta)$ jest* ***nieobciążony****, jeśli dla każdego n*

$$\mathbb{E}\hat{g}(X_1, \ldots, X_n) = g(\theta).$$

Definicja 1.4. *Estymator $\hat{g}(X_1, \ldots, X_n)$ wielkości $g(\theta)$ jest* ***zgodny****, jeśli dla każdego $\theta \in \Theta$*

$$\lim_{n\to\infty} \mathbb{P}_\theta(|\hat{g}(X_1, \ldots, X_n) - g(\theta)| \leq \varepsilon) = 1,$$

dla każdego $\varepsilon > 0$.

Definicja 1.5. *Estymator $\hat{g}(X_1, \ldots, X_n)$ wielkości $g(\theta)$ jest* ***mocno zgodny****, jeśli*

$$\mathbb{P}_\theta\Big(\lim_{n\to\infty} \hat{g}(X_1, \ldots, X_n) = g(\theta)\Big) = 1.$$

Zgodność (mocna zgodność) znaczy tyle, że

$$\hat{g}(X_1, \ldots, X_n) \to g(\theta), \qquad (n \to \infty)$$

według prawdopodobieństwa (prawie na pewno). Interpretacja jest taka: estymator jest uznany za zgodny, jeśli zmierza do estymowanej wielkości przy nieograniczonym powiększaniu badanej próbki.

Jednak zgodność (nawet w mocnym sensie) nie jest specjalnie satysfakcjonującą własnością estymatora, a zaledwie minimalnym żądaniem, które powinien spełniać każdy przyzwoity estymator. Dlatego od niektórych estymatorów żąda się silniejszych właściwości, takich jak asymptotyczna normalność.

Definicja 1.6. *Estymator $\hat{g}(X_1, \ldots, X_n)$ wielkości $g(\theta)$ jest* ***asymptotycznie normalny****, jeśli dla każdego $\theta \in \Theta$ istnieje funkcja $\sigma^2(\theta)$, zwana asymptotyczną wariancją, taka że*

$\xrightarrow{d}$ *oznacza zbieżność wg rozkładu.*

$$\sqrt{n}(\hat{g}(X_1, \ldots, X_n) - g(\theta)) \xrightarrow{d} \mathcal{N}(0, \sigma^2(\theta)), \quad (n \to \infty).$$

Oznacza to, że rozkład prawdopodobieństwa statystyki $\hat{g}(X_1, \ldots, X_n)$ jest dla dużych n zbliżony do rozkładu

$$\mathcal{N}\Big(g(\theta), \frac{\sigma^2(\theta)}{n}\Big).$$

Inaczej mówiąc, estymator jest asymptotycznie normalny, gdy:

$$\lim_{n\to\infty} \mathbb{P}_\theta\Big(\frac{\sqrt{n}}{\sigma(\theta)}(\hat{g}(X_1, \ldots, X_n) - g(\theta) \leq a\Big) = \Phi(a),$$

gdzie $\Phi(x)$ to dystrybuanta standardowego rozkładu normalnego $\mathcal{N}(0, 1)$.

Asymptotyczna normalność mówi, że estymator nie tylko zbiega do nieznanego parametru, ale również że zbiega wystarczająco szybko, jak $\frac{1}{\sqrt{n}}$, czyli, że

$$\mathbb{P}_\theta\Big(\frac{\sqrt{n}}{\sigma(\theta)}(\hat{g}(X_1, \ldots, X_n) - g(\theta) \leq a\Big) - \Phi(a) = f(n) \in \eth(\frac{1}{\sqrt{n}}).$$

Jeśli estymator jest asymptotycznie normalny, to jest zgodny, choć nie musi być *mocno zgodny.*

W dalszej części tego rozdziału zostanie wprowadzone pojęcie estymatora największej wiarogodności oraz zostaną udowodnione dla niego jego właściwości, co utwierdzi w przekonaniu, że metoda największej wiarogodności, przy odpowiednich założeniach, jest metodą konstrukcji rozsądnych estymatorów.

1.2. Metoda największej wiarogodności

Metodę największej wiarogodności wprowadził R. A. Fisher w 1922 r. [15], dla której po raz pierwszy procedurę numeryczną zaproponował już w 1912 r. [14]. O burzliwym procesie powstawania metody, o zmianach w jej uzasadnieniu, o koncepcjach, które powstały w obrębie tej metody takich jak parametr, statystyka, wiarogodność, dostateczność czy efektywność oraz o podejściach, które Fisher odrzucił tworząc podstawy pod nową teorię można przeczytać w obszernej pracy dokumentalnej [1].

Metoda ta, jako alternatywa dla metody najmniejszych kwadratów [23], [17], była rozwijana i szeroko stosowana później przez wielu statystyków, i wciąż znajduje obszerne zastosowania w wielu obszarach estymacji statystycznej, np. [20], [21], [25].

Aby zdefiniować estymator oparty o metodę największej wiarogodności, należy najpierw wprowadzić pojęcie funkcji wiarogodności.

Definicja 1.7. ***Funkcją wiarogodności*** *nazywamy funkcję* $L : \Theta \to \mathbb{R}$ *daną wzorem*

$$L(\theta) = L(\theta; x_1, \ldots, x_n) = f(\theta; x_1, \ldots, x_n),$$

którą rozważamy jako funkcję parametru θ *przy ustalonych wartościach obserwacji* $x_1, \ldots, x_n$*, gdzie*

$$f(\theta; x_1, \ldots, x_n) = \begin{cases} \mathbb{P}_\theta(X_1 = x_1, \ldots, X_n = x_n), & \textit{dla rozkładów dyskretnych,} \\ f_\theta(x_1, \ldots, x_n), & \textit{dla rozkładów absolutnie ciągłych.} \end{cases}$$

Oznacza to, że wiarogodność jest właściwie tym samym, co gęstość prawdopodobieństwa, ale rozważana jako funkcja parametru θ, przy ustalonych wartościach obserwacji $x = X(\omega)$.

Definicja 1.8. ***Estymatorem największej wiarogodności*** *parametru* θ*, oznaczanym ENW(*θ*), nazywamy wartość parametru, w której funkcja wiarogodności przyjmuje supremum*

$$L(\hat{\theta}) = \sup_{\theta \in \Theta} L(\theta).$$

Takie supremum może nie istnieć, dlatego niektóre pozycje w literaturze, w definicji estymatora największej wiarogodności, supremum zastępują wartością największą [33], [43], [29].

1.3. Asymptotyczne własności estymatora największej wiarogodności

W tym podrozdziale zostanie wykazane, że estymator największej wiarygodności jest

i) zgodny,

ii) asymptotycznie normalny,

iii) asymptotycznie nieobciążony.

Asymptotyczna nieobciążoność wynika z asymptotycznej normalności.
Dowody w tym rozdziale są znane w literaturze i opierają się o [30] i [43].

Zgodność estymatora największej wiarogodności

Chcąc wykazać zgodność estymatora największej wiarogodności *przy pewnych warunkach regularności* przydatna będzie poniższa definicja i następujący Lemat.

Definicja 1.9. ***Funkcja log-wiarogodności*** *to funkcja spełniająca równanie*

$$\ell(\theta) = \log(L(\theta)),$$

gdzie przyjmuje się $\ell(\theta) = -\infty$ *jeśli* $L(\theta) = 0$.

Lemat 1.1. *Gdy* θ_0 *to prawdziwe maksimum funkcji wiarogodności, to dla każdego* $\theta \in \Theta$

$$\mathbb{E}_{\theta_0}\ell(\theta) \leq \mathbb{E}_{\theta_0}\ell(\theta_0).$$

Dowód. Rozważając różnicę, przy założeniu ciągłości rozkładu:

$$\begin{aligned}\mathbb{E}_{\theta_0}\ell(\theta) - \mathbb{E}_{\theta_0}\ell(\theta_0) &= \mathbb{E}_{\theta_0}(\ell(\theta) - \ell(\theta_0)) = \mathbb{E}_{\theta_0}(\log f(\theta; X) - \log f(\theta_0; X)) \\ &= \mathbb{E}_{\theta_0} \log \frac{f(\theta; X)}{f(\theta_0; X)},\end{aligned}$$

i pamiętając o tym, że $\log t \leq t - 1$, można dojść do

$\mathbb{E}_{\theta_0}$ oznacza wartość oczekiwaną względem rozkładu parametryzowanego przez θ_0.

$$\begin{aligned}\mathbb{E}_{\theta_0} \log \frac{f(\theta; X)}{f(\theta_0; X)} &\leq \mathbb{E}_{\theta_0}\Big(\frac{f(\theta; X)}{f(\theta_0; X)} - 1\Big) = \int \Big(\frac{f(\theta; x)}{f(\theta_0; x)} - 1\Big) f(\theta_0; x)dx \\ &= \int f(\theta; x)dx - \int f(\theta_0; x)dx = 1 - 1 = 0.\end{aligned}$$

Obie całki równają się 1 jako, że są całkami z funkcji gęstości, zaś równość w nierówności zachodzi tylko wtedy, gdy $\mathbb{P}_\theta = \mathbb{P}_{\theta_0}$. ■

Dzięki temu wynikowi możliwe jest udowodnienie poniższego Twierdzenia.

Twierdzenie 1.2. *Pod pewnymi* ***warunkami regularności*** *nałożonymi na rodzinę rozkładów prawdopodobieństwa, estymator największej wiarogodności* $ENW(\theta)$ *jest zgodny, tzn.*

$$ENW(\theta) \to \theta \quad dla \quad n \to \infty.$$

Dowód.
1) Z definicji w $ENW(\theta)$ przyjmowana jest wartość największa funkcji $L(\theta)$, a więc tym bardziej funkcji $\ell(\theta) = \log L(\theta)$ oraz funkcji

$$\ell_n(\theta) = \frac{1}{n}\ell(\theta) = \frac{1}{n}\log L(\theta) = \frac{1}{n}\sum_{i=1}^{n} \log f(\theta; X_i)$$

(zakładając ciągłość rozkładu i niezależność $X_1, \ldots, X_n$), gdyż ekstremum jest niezmiennicze ze względu na monotoniczną transformację i liniowe przekształcenie jakim jest dzielenie.

2) Z Lematu 1.1 wynika, że θ_0 maksymalizuje $\mathbb{E}_{\theta_0}\ell(\theta)$.

3) Z Prawa Wielkich Liczb, które jest spełnione gdy założy się, że X_i to realizacje ciągu zmiennych losowych o skończonych wartościach oczekiwanych, wynika, że

$$\ell_n(\theta) = \frac{1}{n}\sum_{i=1}^{n} \log f(\theta; X_i) \to \mathbb{E}_{\theta_0}\ell(\theta),$$

co ostatecznie oznacza, że $ENW(\theta)$ jest zgodny. ■

Asymptotyczna normalność estymatora największej wiarygodności

Fisher w swojej karierze wprowadził wiele pożytecznych pojęć stosowanych do dziś. Jednym z nich jest Informacja Fishera, która zostanie wykorzystana w dowodzie asymptotycznej normalności estymatora największej wiarogodności.

Definicja 1.10. *Niech X będzie zmienną losową o gęstości f_θ, zależnej od jednowymiarowego parametru $\theta \in \Theta \subset \mathbb{R}$.* ***Informacją Fishera*** *zawartą w obserwacji X nazywa się funkcję*

$$\mathcal{I}(\theta) = \mathbb{E}_\theta(\ell'(\theta; X))^2 = \mathbb{E}_\theta\Big(\frac{\partial}{\partial\theta}\log f_\theta(X)\Big)^2, \tag{1.1}$$

gdzie odpowiednio

$$\mathcal{I}(\theta) = \int \Big(\frac{\partial}{\partial\theta}\log f_\theta(x)\Big)^2 f_\theta(x)dx \qquad \textit{dla zmiennej ciągłej;}$$
$$\mathcal{I}(\theta) = \sum_x \Big(\frac{\partial}{\partial\theta}\log f_\theta(x)\Big)^2 \mathbb{P}_\theta(X = x) \;\; \textit{dla zmiennej dyskretnej.}$$

W dowodzie asymptotycznej normalności estymatora największej wiarogodności kluczowymi założeniami są poniższe warunki regularności. Rodzina gęstości musi być dostatecznie regularna aby pewne kroki rachunkowe w dalszych rozumowaniach były poprawne.

Definicja 1.11. ***Warunki regularności.***

i) Informacja Fishera jest dobrze określona. Zakłada się, że Θ jest przedziałem otwartym, istnieje pochodna $\frac{\partial}{\partial\theta}\log f_\theta$, całka/suma we wzorze (1.1) jest bezwzględnie zbieżna (po obłożeniu funkcji podcałkowej modułem całka istnieje i jest skończona) i $0 < \mathcal{I}(\theta) < \infty$.

ii) Wszystkie gęstości f_θ mają jeden nośnik, tzn. zbiór $\{x \in X : f_\theta(x) > 0\}$ nie zależy od θ.

iii) Można przenosić pochodną przed znak całki, czyli zamienić kolejność operacji różniczkowania $\frac{\partial}{\partial\theta}$ i całkowania $\int \ldots dx$.

Wprowadzając takie założenia, otrzymano przydatne właściwości Informacji Fishera.

Stwierdzenie 1.3. *Jeśli spełnione są warunki regularności (1.11) to:*

$$(i)\quad \mathbb{E}_\theta\frac{\partial}{\partial\theta}\log f_\theta(x) = 0,$$
$$(ii)\quad \mathcal{I}(\theta) = \mathbb{V}ar_\theta\Big(\frac{\partial}{\partial\theta}\log f_\theta(X)\Big),$$
$$(iii)\quad \mathcal{I}(\theta) = -\mathbb{E}_\theta\Big(\frac{\partial^2}{\partial\theta^2}\log f_\theta(X)\Big).$$

Dowód tego stwierdzenia można znaleźć w [27].

Patrząc na postać pochodnej funkcji log-wiarogodności

$$\ell'(\theta_0; X) = (\log f(\theta_0; X))' = \frac{f'(\theta_0; X)}{f(\theta_0; X)},$$

można wywnioskować, że nieformalnie interpretacja Informacji Fishera jest miarą tego jak szybko zmieni się funkcja gęstości jeśli delikatnie zmieni się parametr θ w okolicach θ_0. Biorąc kwadrat i wartość oczekiwaną, innymi słowy uśredniając po X, otrzymuje się uśrednioną wersję tej miary. Jeżeli Informacja Fishera jest duża, oznacza to, że gęstość zmieni się szybko gdyby poruszyć parametr θ_0, innymi słowy - gęstość z parametrem θ_0 jest znacząco inna i może zostać łatwo odróżniona od gęstości z parametrami nie tak bliskimi θ_0. Stąd wiemy, że możliwa estymacja θ_0 oparta o takie dane jest dobra. Z drugiej strony, jeżeli Informacja Fishera jest mała, oznacza to, że gęstość dla θ_0 jest bardzo podobna do gęstości z parametrami nie tak bliski do θ_0, a co za tym idzie, dużo ciężej będzie odróżnić tę gęstość, czyli estymacja będzie słabsza.

Dzięki pojęciu Informacji Fishera i warunkom regularności, możliwe jest udowodnienie poniższego Twierdzenia.

Twierdzenie 1.4. *Pod pewnymi* ***warunkami regularności*** *nałożonymi na rodzinę rozkładów prawdopodobieństwa, estymator największej wiarogodności jest asymptotycznie normalny,*

$$\sqrt{n}(ENW(\theta) - \theta_0) \to \mathcal{N}\Big(0, \frac{1}{\mathcal{I}(\theta_0)}\Big).$$

Z Twierdzenia widać, że im większa Informacja Fishera tym mniejsza asymptotyczna wariancja estymatora prawdziwego parametru θ_0.

Dowód. Ponieważ $ENW(\theta)$ maksymalizuje $\ell_n(\theta) = \frac{1}{n}\sum_{i=1}^{n} \log f(\theta; X)$, to $\ell_n'(\theta) = 0$.
Dalej, korzystając z Twierdzenia o Wartości Średniej:

$$\frac{g(a) - g(b)}{a - b} = g'(c) \text{ albo } g(a) = g(b) + g'(c)(a - b), \text{ dla } c \in [a, b],$$

gdzie $g(\theta) = \ell_n'(\theta), a = ENW(\theta), b = \theta_0$, można zapisać równość

$$0 = \ell_n'(ENW(\theta)) = \ell_n'(\theta_0) + \ell_n''(\theta_1)(ENW(\theta) - \theta_0), \text{ dla } \theta_1 \in [ENW(\theta), \theta_0],$$

a z niej przejść do postaci

$$\sqrt{n}(ENW(\theta) - \theta_0) = -\frac{\sqrt{n}\ell_n'(\theta_0)}{\ell_n''(\theta_1)}. \tag{1.2}$$

Z Lematu (1.1) wynika, że θ_0 maksymalizuje $\mathbb{E}_{\theta_0}\ell(\theta_0)$ czyli

$$\mathbb{E}_{\theta_0}\ell'(\theta_0) = 0, \tag{1.3}$$

a to można wstawić do licznika w równaniu (1.2)

$$\begin{aligned}\sqrt{n}\ell_n'(\theta_0) &= \sqrt{n}\Big(\frac{1}{n}\sum_{i=1}^{n}\ell'(\theta_0) - 0\Big) \\ &= \sqrt{n}\Big(\frac{1}{n}\sum_{i=1}^{n}\ell'(\theta_0) - \mathbb{E}_{\theta_0}\ell'(\theta_0)\Big) \to \mathcal{N}\Big(0, \mathbb{V}ar_{\theta_0}(\ell'(\theta_0))\Big),\end{aligned} \tag{1.4}$$

gdzie zbieżność wynika z Centralnego Twierdzenia Granicznego.

Następnie można rozważyć mianownik w równaniu (1.2). Dla wszystkich θ wynika

$$\ell''(\theta) = \frac{1}{n}\sum_{i=1}^{n} \ell''(\theta) \to \mathbb{E}_{\theta_0}\ell''(\theta)$$

z Prawa Wielkich Liczb.

Dodatkowo, ponieważ $\theta_1 \in [ENW(\theta), \theta_0]$ a $ENW(\theta)$ jest zgodny (poprzedni podrozdział), to ponieważ $ENW(\theta) \to \theta_0$, to też $\theta_1 \to \theta_0$, a wtedy

$$\ell_n''(\theta_1) \to \mathbb{E}_{\theta_0}\ell''(\theta_0) = -\mathcal{I}(\theta_0)$$

z punktu (*iii*) ze Stwierdzenia (1.3).

Wtedy prawa strona równania (1.2), dzięki (1.4)

$$-\frac{\sqrt{n}\ell_n'(\theta_0)}{\ell_n''(\theta_1)} \xrightarrow{d} \mathcal{N}\Big(0, \frac{\mathbb{V}ar_{\theta_0}(\ell'(\theta_0))}{(\mathcal{I}(\theta_0))^2}\Big).$$

Ostatecznie wariancja

$$\mathbb{V}ar_{\theta_0}(\ell'(\theta_0)) \overset{z\ def}{=} \mathbb{E}_{\theta_0}(\ell'(\theta_0))^2 - (\mathbb{E}_{\theta_0}\ell'(\theta_0))^2 = \mathcal{I}(\theta_0) - 0,$$

co wynika z definicji Informacji Fishera i (1.3).

■

Rozdział 2

Model Coxa

W tym rozdziale zostanie przedstawiony model proporcjonalnych hazardów Coxa. Głównym celem tej pracy jest wykorzystanie numerycznej metody estymacji współczynników metodą stochastycznego spadku gradientu w omawianym modelu. Więcej o estymacji metodą stochastycznego spadku gradientu napisane jest w rozdziale 3.2. Definicje i twierdzenia w tym rozdziale oparte są o [9], [37], [2] i [7].

2.1. Wprowadzenie do modelu Coxa i nomenklatura

Analiza przeżycia, w której model Coxa znalazł największe zastosowania, polega na modelowaniu wpływu czynników na czas do wystąpienia pewnego zdarzenia. Zdarzeniem może być np. śmierć pacjenta, awaria urządzenia, zerwanie umowy przez klienta, odejście z pracy pracownika lub deaktywacja pewnej usługi. Analizując czasy do wystąpienia zdarzenia wykorzystuje się funkcję przeżycia bądź niosącą równoważną informację funkcję hazardu.

Definicja 2.1. ***Funkcja przeżycia*** *w grupie j, to funkcja, która spełnia*

$$S_j(t) = \mathbb{P}(T^*_{j,i} \geq t) = 1 - F_j(t), t \in \mathbb{R} \tag{2.1}$$

gdzie $F_j(t)$ to dystrybuanta rozkładu zadanego gęstością $f_j(t)$.

Definicja 2.2. ***Funkcja hazardu*** *to funkcja, która wyraża się wzorem*

$$\begin{aligned} \lambda_j(t) &= \lim_{h\to 0} \frac{\mathbb{P}(t \leq T^* \leq t+h | T^* \geq t)}{h} \\ &= \lim_{h\to 0} \frac{\mathbb{P}(t \leq T^* \leq t+h)}{h} \cdot \frac{1}{\mathbb{P}(T^* \geq t)} \\ &= \lim_{h\to 0} \frac{F_j(t+h) - F_j(t)}{h} \cdot \frac{1}{S_j(t)} \qquad = \frac{f_j(t)}{S_j(t)}. \end{aligned} \tag{2.2}$$

W tych definicjach T^* oznacza czas do wystąpienia zdarzenia. Zakłada się, że wewnątrz każdej grupy $j = 1, 2, \ldots, m$ wyznaczonej przez poziomy zmiennych objaśniających, czasy $T^*_{j,i}$ dla $i = 1, \ldots, n_j$ to niezależne zmienne losowe z tego samego rozkładu o gęstości $f_j(t)$.

Wartość funkcji hazardu w momencie t traktuje się jako chwilowy potencjał pojawiającego się zdarzenia (np. śmierci lub choroby), pod warunkiem że osoba dożyła czasu t. Funkcja hazardu nazywana jest również funkcją ryzyka, intensywnością umieralności (*force of mortality*), umieralnością chwilową (*instantaneous death rate*) lub chwilową częstością niepowodzeń (awarii) (*failure rate*). Ostatniego określenia używa się w teorii odnowy [8], w której analizuje się awaryjność elementów przemysłowych.

Model proporcjonalnych hazardów Coxa [9] jest obecnie najczęściej stosowaną procedurą do modelowania relacji pomiędzy zmiennymi objaśniającymi a przeżyciem, lub innym cenzurowanym zdarzeniem. Model ten umożliwia analizę wpływu czynników prognostycznych na przeżycie. Sir David Cox opracował tego typu model dla tabeli przeżyć i zilustrował zastosowanie modelu dla przypadku białaczki, ale model może być stosowany do obliczania przeżyć w odniesieniu do innych chorób, jak w przypadku przeżyć w chorobach nowotworowych lub kardiologicznych po transplantacji serca lub zawałach serca [28].

Definicja 2.3. ***Model Coxa*** *zakłada postać funkcji hazardu dla i-tej obserwacji X_i jako*

$$\lambda_i(t) = \lambda_0(t)e^{X_i(t)'\beta}, \tag{2.3}$$

gdzie λ_0 to niesprecyzowana nieujemna funkcja nazywana bazowym hazardem, a β to wektor współczynników rozmiaru p, co odpowiada liczbie zmiennych objaśniających w modelu Coxa.

Takie sformułowanie modelu gwarantuje, że funkcja hazardu jest nieujemna.

Model Coxa, dla wersji kiedy współczynniki są stałe w czasie, nazywany jest **modelem proporcjonalnych hazardów**, gdyż stosunek (proporcja) hazardów dla dwóch obserwacji X_i oraz X_j jest stały w czasie:

$$\frac{\lambda_i(t)}{\lambda_j(t)} = \frac{\lambda_0(t)e^{X_i\beta}}{\lambda_0(t)e^{X_j\beta}} = \frac{e^{X_i\beta}}{e^{X_j\beta}} = e^{(X_i - X_j)\beta}.$$

Oznacza to, że hazard dla jednej obserwacji można uzyskać poprzez przemnożenie hazardu dla innej obserwacji przez pewną stałą c_{ij}:

$$\lambda_i(t) = \frac{e^{X_i'\beta}}{e^{X_j'\beta}} \cdot \lambda_j(t) = c_{ij} \cdot \lambda_j(t).$$

W modelu proporcjonalnych hazardów istotnym elementem jest estymacja stałych c_{ij}.

2.2. Założenia modelu proporcjonalnego ryzyka Coxa

Model Coxa znalazł szerokie zastosowanie w sytuacjach, gdy analiza wymaga wykorzystania cenzurowanych danych. Model Coxa jest w stanie wykorzystać je do estymacji współczynników w modelu, przekładających się na proporcje hazardów. Z uwagi na aspekt praktyczny podyktowany warunkami technicznymi prób klinicznych i badań biologicznych, zbiory danych klinicznych zawierają cenzurowane czasy zdarzeń. Oznacza to, że w wielu przypadkach niemożliwe jest obserwowanie czasu zdarzeń dla wszystkich obserwacji w zbiorze. Niekiedy jest to uwarunkowane zbyt długim czasem do wystąpienia zdarzenia. Czasem jest to związane z zaplanowanym okresem próby klinicznej, który jest krótszy niż czas do zdarzenia dla pacjentów, którzy mogli zostać włączeni do próby klinicznej pod koniec jej trwania i nie

udało się dla nich zaobserwować czasów zdarzeń. W wielu przypadkach pacjenci, traktowani jako obserwacje w zbiorze, znikają z pola widzenia w momencie, gdy np. przestają pojawiać się na wizytach kontrolnych. Może być to spowodowane negatywnymi relacjami z lekarzem prowadzącym lub przeprowadzką. W takich sytuacjach wykorzystuje się daną obserwację do momentu jej ostatniej kontroli. Nie rezygnuje się z tej obserwacji w analizie i wykorzystuje się o niej informacje w pełni dla czasu, w którym przebywała pod obserwacją. Jest to ogromna zaleta modelu Coxa.

Z przyczyny cenzurowanych danych potrzebne są założenia modelu dotyczące cenzurowania czasów, które opierają się o następujące definicje.

Definicja 2.4. ***Cenzuorwanie prawostronne*** *polega na zaobserwowaniu czasu*

$$T = \min(T^*, C),$$

gdzie T^* to prawdziwy czas zdarzenia, zaś C jest nieujemną zmienną losową.

Definicja 2.5. ***Cenzurowanie jest niezależne*** *jeśli zachodzi*

$$\lim_{h\to 0} \frac{\mathbb{P}(t \le T^* \le t+h | T^* \ge t)}{h} = \lim_{h\to 0} \frac{\mathbb{P}(t \le T^* \le t+h | T^* \ge t, Y(t)=1)}{h},$$

gdzie $Y(t) = 1$ jeśli do chwili t nie wystąpiło zdarzenie ani cenzurowanie, czyli jednostka pozostaje narażona na ryzyko zdarzenia oraz $Y(t) = 0$ w przeciwnym wypadku.

Interpretacja tej definicji jest następująca: jednostka cenzurowana w chwili t jest reprezentatywna dla wszystkich innych narażonych na ryzyko zdarzenia w chwili t. Innymi słowy cenzurowanie nie wybiera z populacji osobników bardziej albo mniej narażonych na zdarzenie. Cenzurowanie działa niezależnie od mechanizmu występowania zdarzenia.

Definicja 2.6. ***Cenzurowanie jest nie-informatywne*** *jeśli zachodzi*

$$g(t;\theta,\phi) \equiv g(t;\phi), \tag{2.4}$$

gdzie $g(t;\theta,\phi)$ jest funkcją gęstości dla cenzurowań C_i wyrażonych jako niezależne zmienne losowe o jednakowym rozkładzie, zaś prawdzie czasy T_i^* są interpretowane jako niezależne zmienne losowe o jednakowym rozkładzie i funkcji gęstości $f(t;\theta)$, czyli θ parametryzuje jedynie rozkład czasów zdarzeń.

Oznacza to, że cenzurowanie nie daje informacji o parametrach rozkładu czasów zdarzeń, ponieaż nie zależy od parametrów od których zależny jest hazard.

Model proporcjonalnych hazardów Coxa oparty jest na założeniach:

i) Współczynniki modelu $\beta_k, k = 1, \cdots, p$ są stałe w czasie, co przekłada się na to, że stosunek hazardów dla dwóch obserwacji jest stały w czasie.

ii) Postać funkcjonalna efektu zmiennej niezależnej, czyli postać modelu $\lambda_i(t) = \lambda_0(t)e^{X_i(t)'\beta}$.

iii) Obserwacje są niezależne.

iv) Cenzurowanie czasów jest nie-informatywne.

v) Cenzurowanie czasów jest niezależne (od mechanizmu występowania zdarzenia).

2.3. Estymacja w modelu Coxa

Funkcja hazardu jest wykładniczą funkcją zmiennych objaśniających, nieznana jest natomiast postać bazowej funkcji hazardu, co bez dalszych założeń uniemożliwia estymację standardową metodą największej wiarygodności. Rozwiązaniem Cox'a jest maksymalizacja tylko tego fragmentu funkcji wiarygodności, który zależy jedynie od estymowanych parametrów. W modelu Coxa proporcjonalnych hazardów estymacja współczynników β oparta jest o częściową funkcję wiarogodności, którą wprowadził Cox w 1972 r. [9].

Dla konkretnego czasu zdarzenia t_i, gdzie w zbiorze obserwowanych jest K czasów zdarzeń, prawdopodobieństwo warunkowe ze względu na liczność zbioru ryzyka w czasie t_i, że czas zdarzenia dotyczy i-tej jednostki spośród wciąż obserwowanych jest równe

$$\frac{e^{X_i'\beta}}{\sum\limits_{l\in\mathscr{R}(t_i)} e^{X_l'\beta}}, \tag{2.5}$$

gdzie *zbiór ryzyka* $\mathscr{R}(t_i)$, w chwili t_i, rozumiany jest jako zbiór indeksów obserwacji, które są w danym czasie t_i pod obserwacją.

Chcąc estymować współczynniki metodą największej wiarogodności należy rozważyć funkcję wiarogodności, która dla niezależnego cenzurowania prawostronnego ma postać:

$$L(\beta,\varphi) = L_p(\beta)\cdot L^*(\beta,\varphi), \tag{2.6}$$

gdzie

$$L_p(\beta) = \prod_{i=1}^{n} f(t_j;\beta)^{\delta_j} S(t_j;\beta)^{1-\delta_j} = \prod_{i=1}^{n} \lambda(t_j;\beta)^{\delta_j} S(t_j;\beta) \tag{2.7}$$

to częściowa funkcja wiarogodności, a $L^*(\beta,\varphi)$ zależy od cenzurowania (parametr φ).

Wtedy dla niezależnego cenzurowania i dla czasów zdarzeń, które nie zaszły jednocześnie **częściowa funkcja wiarogodności w modelu Coxa** ma postać:

$$L_p(\beta) = \prod_{i=1}^{K} \frac{e^{X_i'\beta}}{\sum\limits_{l=1}^{n} Y_l(t_i) e^{X_l'\beta}}, \tag{2.8}$$

gdzie $Y_l(t_i) = 1$, gdy obserwacja X_l jest w zbiorze ryzyka w czasie t_i, i $Y_l(t_i) = 0$ w przeciwnym przypadku, n to liczba obserwacji w zbiorze, a K to wspomniana wyżej liczba zaobserwowanych czasów zdarzeń. Zaletą takiej postaci funkcji częściowej wiarogodności jest to, że w jej wzorze nie występuje funkcja bazowego hazardu, zatem estymacja współczynników może odbywać się bez znajomości jej postaci.

Jeśli dodatkowo cenzurowanie jest nie-informatywne, to $L_p(\beta)$ jest **pełną** funkcją wiarogodności, bowiem wówczas

$$L^*(\beta,\varphi) \propto L^*(\varphi)$$

co bierze się z definicji cenzurowania nie-informatywnego (2.4)

$$g(t;\theta,\phi) \equiv g(t;\phi).$$

Ponieważ model proporcjonalnych hazardów Coxa zakłada niezależność i nie-informatywność cenzurowania zatem można uważać, że częściowa funkcja wiarogodności daje pełną informację o współczynnikach i wnioskowanie w oparciu o nią jest uzasadnione i poprawne.

W sytuacjach, gdy nie jest spełnione założenie nie-informatywności cenzurowania i częściowa funkcja wiarogodności nie jest funkcją wiarogodności w sensie bycia proporcjonalną do prawdopodobieństwa obserwowanego zbioru, można ją traktować jako funkcję wiarogodności dla celów asymptotycznego wnioskowania o współczynnikach modelu, zobacz [**?**].

Analityczna estymacja współczynników

Standardowo w celu znalezienia maximum, aby ułatwić obliczenia, można rozważaną funkcję obłożyć monotoniczną transformacją jaką jest logarytm, tak aby w konsekwencji otrzymać **częściową funkcję log-wiarogodności**

$$\ell_p(\beta) = \sum_{i=1}^{K} X_i'\beta - \sum_{i=1}^{K} \log\Big(\sum_{l\in\mathscr{R}(t_i)} e^{X_l'\beta}\Big). \tag{2.9}$$

Bezpośrednie kalkulacje dają p-wymiarowy wektor pochodnych, dla $k = 1, \ldots, p$

$$U_k(\beta) = \frac{\partial \ell_k(\beta)}{\partial \beta_k} = \sum_{i=1}^{K} \Big(X_{ik} - A_{ik}\Big), \tag{2.10}$$

X_{ik} to i ta obserwacja i k ta zmienna.

gdzie czynnik

$$A_{ik} = \frac{\sum\limits_{l\in\mathscr{R}(t_i)} X_{lk} e^{X_l'\beta}}{\sum\limits_{l\in\mathscr{R}(t_i)} e^{X_l'\beta}} \tag{2.11}$$

to średnia z $X_{.k}$ (k-tych zmiennych) po skończonej populacji $\mathscr{R}(t_i)$, z wykorzystaniem *ważonej eksponencjalnie* formy próbkowania.

Z kolei drugie pochodne cząstkowe, jak podaje [9], mają postać dla $k_1, k_2 = 1, \ldots, p$

$$\mathscr{I}_{k_1k_2}(\beta) = -\frac{\partial^2 L_p(\beta)}{\partial \beta_{k_1} \partial \beta_{k_2}} = \sum_{i=1}^{K} C_{ik_1k_2}(\beta), \tag{2.12}$$

gdzie

$$C_{ik_1k_2}(\beta) = \frac{\sum\limits_{l\in\mathscr{R}(t_i)} X_{lk_1} X_{lk_2} e^{X_l'\beta}}{\sum\limits_{l\in\mathscr{R}(t_i)} e^{X_l'\beta}} - A_{ik_1}(\beta) A_{ik_2}(\beta) \tag{2.13}$$

to kowariancja pomiędzy $X_{.k_1}$ (k_1-tymi zmiennymi) a $X_{.k_2}$ (k_2-tymi zmiennymi) przy tej formie ważonego próbkowania.

Estymator największej wiarogodności β można uzyskać poprzez przyrównanie (2.10) do 0, a numerycznie poprzez iteracyjne wykorzystanie (2.10) oraz (2.12) w algorytmie spadku gradientu rzędu II nazywanego również algorytmem Raphsona-Newtona, który jest opisany w podrozdziale 3.1. Jest to tradycyjne i szeroko stosowane podejście do estymacji współczynników w modelu proporcjonalnych hazardów Coxa. Niniejsza praca skupia się na wykorzystaniu metody estymacji współczynników jaką jest metoda stochastycznego spadku wzdłuż gradientu, która jest szerzej opisana w następnym rozdziale w podrozdziale 3.2.

Rozdział 3

Numeryczne metody estymacji

Przez **numerykę** rozumie się dziedzinę matematyki zajmującą się rozwiązywaniem przybliżonych zagadnień algebraicznych. Odkąd zjawiska przyrodnicze zaczęto opisywać przy użyciu formalizmu matematycznego, pojawiła się potrzeba rozwiązywania zadań analizy matematycznej czy algebry. Dopóki były one nieskomplikowane, dawały się rozwiązywać analitycznie, tzn. z użyciem pewnych przekształceń algebraicznych prowadzących do otrzymywania rozwiązań ścisłych danych problemów. Z czasem jednak, przy powstawaniu coraz to bardziej skomplikowanych teorii opisujących zjawiska, problemy te stawały się na tyle złożone, iż ich rozwiązywanie ścisłe było albo bardzo czasochłonne albo też zgoła niemożliwe. Numeryka pozwalała znajdywać przybliżone rozwiązania z żądaną dokładnością. Ich podstawową zaletą była ogólność tak formułowanych algorytmów, tzn. w ramach danego zagadnienia nie miało znaczenia czy było ono proste czy też bardzo skomplikowane (najwyżej wiązało się z większym nakładem pracy obliczeniowej). Natomiast wadą była czasochłonność. Stąd prawdziwy renesans metod numerycznych nastąpił wraz z powszechnym użyciem w pracy naukowej maszyn cyfrowych, a w szczególności mikrokomputerów [24]. Dziś dziesiątki żmudnych dla człowieka operacji arytmetycznych wykonuje komputer, jednak złożoność obliczeniowa algorytmów uczących i modeli statystycznych stała się krytycznym czynnikiem ograniczającym w sytuacjach, gdy rozważane są duże zbiory danych. Te ograniczenia spowodowały, że w uczeniu maszynowym i modelowaniu statystycznym wielkiej skali zaczęto wykorzystywać algorytmy **stochastycznego spadku gradientu**. W poniższym rozdziale przedstawione są klasyczne algorytmy spadku wzdłuż gradientu Cauchy'ego oraz Raphsona-Newtona. Następnie omówiony jest algorytm stochastycznego spadku wzdłuż gradientu, którego wykorzystanie do estymacji współczynników w modelu Coxa jest kluczowym celem tej pracy. Algorytm stochastycznego spadku gradientu to metoda optymalizacji wzdłuż spadku gradientu wykorzystywana w sytuacjach, gdy rozważaną funkcję można zapisać jako sumę różniczkowalnych składników. Ponadto przedstawiono również zalety algorytmów stochastycznego spadku gradientu, które przemawiają za atrakcyjnością i popularnością tego typu rozwiązania. Ostatecznie przedyskutowano asymptotyczną efektywność estymatorów uzyskanych dzięki jednemu przejściu po zbiorze, zwanym *epoką*. Definicje i pojęcia w tym rozdziale pochodzą z [4], [6], [22] i [16].

3.1. Algorytmy spadku wzdłuż gradientu

Poniższy rozdział przedstawia popularne iteracyjne algorytmy wyznaczania przybliżonej wartości miejsca zerowego funkcji oraz rozważaną w pracy metodę stochastycznego spadku gradientu. Szukanie miejsc zerowych funkcji jest przydatne w problemach optymalizacyjnych, gdy celem jest znalezienie pierwiastka pochodnych badanej funkcji. Dodatkowo takie algorytmy wykorzystywane są do rozwiązywania (nieliniowych) układów równań. Metody iteracyjne składają się zazwyczaj z k kroków bądź są zatrzymywane, gdy osiągnięty zostanie warunek stopu, czyli gdy odległość pomiędzy kolejnymi przybliżeniami jest dość mała $\| w_{k+1} - w_k \| < \epsilon$ lub wartość gradientu funkcji w wyznaczonym punkcie jest bliska wektorowymi zerowemu $\| \nabla_Q(\mathbf{w_k}) \| \leqslant \epsilon$ (test stacjonarności), gdzie ϵ to zadana z góry precyzja. Metoda stochastycznego spadku wzdłuż gradientu zakłada, że minimalizowaną funkcję $Q(w)$ można przedstawić jako różniczkowalną sumę jej składników $Q(w) = \sum_{i=1}^{n} Q_i(w)$. W poniższych algorytmach α_k oznacza długość kroku algorytmu.

Metoda spadku wzdłuż gradientu I (Cauchy'ego)

Minimalizacja funkcji $Q(w)$:

- Zaczynamy od wybranego rozwiązania startowego, np. $w_0 = 0$.
- Dla $k = 1, 2, \ldots$ aż do zbieżności
  - Wyznaczamy gradient w punkcie $w_{k-1}, \nabla_Q(w_{k-1})$.
  - Robimy krok wzdłuż negatywnego gradientu:

$$w_k = w_{k-1} - \alpha_k \nabla_Q(w_{k-1}).$$

Metoda spadku wzdłuż gradientu II (Newtona-Raphsona)

Minimalizacja funkcji $Q(w)$:

- Zaczynamy od wybranego rozwiązania startowego, np. $w_0 = 0$.
- Dla $k = 1, 2, \ldots$ aż do zbieżności
  - Wyznaczamy gradient w punkcie $w_{k-1}, \nabla_Q(w_{k-1})$ i odwrotność Hesjanu $(D^2_Q(w_{k-1}))^{-1}$.
  - Robimy krok wzdłuż negatywnego gradientu z zadanym krokiem przez Hesjan:

$$w_k = w_{k-1} - (D^2_Q(w_{k-1}))^{-1} \nabla_Q(w_{k-1}).$$

Metoda stochastycznego spadku wzdłuż gradientu I

Minimalizacja funkcji $Q(w)$:

- Zaczynamy od wybranego rozwiązania startowego, np. $w_0 = 0$.
- Dla $k = 1, 2, \ldots$ aż do zbieżności
  - Wylosuj $i \in \{1, \ldots, n\}$
  - Wyznaczamy gradient funkcji Q_i w punkcie $w_{k-1}, \nabla_{Q_i}(w_{k-1})$.
  - Robimy krok wzdłuż negatywnego gradientu:

$$w_k = w_{k-1} - \alpha_k \nabla_{Q_i}(w_{k-1}). \tag{3.1}$$

3.2. Algorytm stochastycznego spadku wzdłuż gradientu I

Stochastyczny spadek gradientu to popularny algorytm wykorzystywany do estymacji współczynników w szerokiej gamie modeli uczenia maszynowego takich jak maszyny wektorów podpierających (*ang. Support Vector Machines*), regresja logistyczna czy modele graficzne [13]. W połączeniu z algorytmem propagacji wstecznej jest standardowym algorytmem w trenowaniu sztucznych sieci neuronowych. Algorytm stochastycznego spadku gradientu był używany już od 1960 przy estymacji współczynników w modelu regresji liniowej, oryginalnie znanym jako *ADALINE* [39]. Kolejnym algorytmem wykorzystującym stochastyczny spadek gradientu jest filtr adaptacyjny najmniejszych średnich kwadratów [41] (*ang. least mean squares (LMS) adaptive filter*), który został wynaleziony przez Bernarda Widrowa, twórcę *ADALINE*.

Idea algorytmu stochastycznego spadku gradientu jest następująca: zamiast obliczać gradient na całej funkcji L, w danym kroku oblicz gradient tylko na pojedynczym elemencie ℓ_i. Nazwa *stochastyczny* bierze się stąd, iż oryginalnie wybiera się element ℓ_i losowo. W praktyce zwykle przechodzi się po całym zbiorze danych w losowej kolejności.

Właściwości stochastycznego spadku wzdłuż gradientu

Zbieżność algorytmu stochastycznego spadku gradientu była szeroko badana w literaturze aproksymacji stochastycznych. Aby uzyskać zbieżność zazwyczaj wymaga się aby ciąg kroków algorytmu α_k był malejący i spełniał poniższe warunki $\sum_k \alpha_k = \infty$ oraz $\sum_k \alpha_k^2 < \infty$ [4]. Twierdzenie Robbinsa-Siegmunda [32] przy łagodnych warunkach zapewnia zbieżność prawie na pewno [5], nawet gdy optymalizowana funkcja nie jest wszędzie różniczkowalna.

Prędkość zbieżności stochastycznego spadku gradientu jest w rzeczywistości ograniczana przez zgrubną (*ang. noisy*) aproksymacje prawdziwego gradientu. Gdy długości kroków algorytmu maleją zbyt wolno, wariancja estymatorów parametrów w_k maleje równie wolno. Gdy kroki algorytmu maleją zbyt szybko, oczekiwane estymatory parametrów w_k potrzebują więcej czasu by osiągnąć optimum [4]. Pod pewnymi warunkami regularności [26], najlepsza prędkość zbieżności jest uzyskana dla kroków algorytmu $\alpha_k \sim k^{-1}$.

Jak wykazano w [11] pod pewnymi odpowiednimi warunkami regularności, gdy zainicjowany współczynnik początkowy w_0 jest wystarczająco blisko optimum i krok algorytmu jest odpowiednio mały, algorytm stochastycznego spadku gradientu osiąga liniową zbieżność. Oznacza to, iż przy spełnieniu założeń metody, odległości pomiędzy kolejnymi przybliżeniami a minimum funkcji $\mathbf{w}^*$ maleją liniowo: $\| \mathbf{w}^* - \mathbf{w_{k+1}} \| \leqslant c \| \mathbf{w}^* - \mathbf{w_k} \|$. Zbieżność wymaga często przejścia parokrotnie po całym zbiorze danych. Wady i zalety algorytmu wymienione są poniżej. Zalety zdecydowanie przewyższają wady.

Zalety

- Szybkość: obliczenie gradientu wymaga wzięcia tylko jednej obserwacji.
- Skalowalność: cały zbiór danych nie musi nawet znajdować się w pamięci operacyjnej.
- Prostota: gradient funkcji Q_i daje bardzo prosty wzór na modyfikacje wag.

Wady

- Wolna zbieżność: czasem gradient stochastyczny zbiega wolno i wymaga wielu iteracji po zbiorze uczącym.
- Problem z ustaleniem długości kroku k: wyznaczenie k przez przeszukiwanie liniowe nie przynosi dobrych rezultatów, ponieważ nie optymalizujemy oryginalnej funkcji Q tylko jej jeden składnik Q_i.

3.3. Porównanie algorytmów spadku wzdłuż gradientu

W niniejszym podrozdziale przedstawiono graficznie różnice w wyborze kolejnych punktów w trakcie optymalizacji między omawianymi w poprzedniej części pracy algorytmami spadku wzdłuż gradientu I (Cauchy'ego), spadku wzdłuż gradientu II (Newtona-Raphsona) oraz stochastycznego spadku wzdłuż gradientu I. W celu zobrazowania przykładu na dwuwymiarowym wykresie, postanowiono ograniczyć się do modelu z jedną zmienną objaśniającą i wyrazem wolnym. Do przykładu wybrano model regresji logistycznej, z racji na prostotę przedstawienia funkcji log-wiarogodności jako sumy różniczkowalnych składników.

Funkcja log-wiarogodności dla modelu regresji logistycznej, za [10] i [12], ma postać

$\beta = (\beta_1, \beta_2)$

$$\ell(\beta) = \sum_{i=1}^{N} \Big(y_i(\beta_1 + \beta_2 x_i) - \log(1 + \exp(\beta_1 + \beta_2 x_i)) \Big) = \sum_{i=1}^{N} Q_i(\beta_1, \beta_2), \tag{3.2}$$

$$Q_i(\beta_1, \beta_2) = y_i(\beta_1 + \beta_2 x_i) - \log(1 + \exp(\beta_1 + \beta_2 x_i)). \tag{3.3}$$

Dla tak skonstruowanej funkcji wiarogodności, współrzędne gradientu to odpowiednio

$$\frac{\partial \ell(\beta)}{\partial \beta_1} = \sum_{i=1}^{N} \Big(y_i - \pi_i(\beta) \Big), \qquad \frac{\partial \ell(\beta)}{\partial \beta_2} = \sum_{i=1}^{N} x_i \Big(y_i - \pi_i(\beta) \Big),$$

zaś macierz informacji wyraża się jak następuje

$$\mathscr{I}(\beta) = \begin{bmatrix} \sum\limits_{i=1}^{N} \pi_i(\beta)(1 - \pi_i(\beta)) & \sum\limits_{i=1}^{N} x_i \pi_i(\beta)(1 - \pi_i(\beta)) \\ \sum\limits_{i=1}^{N} x_i \pi_i(\beta)(1 - \pi_i(\beta)) & \sum\limits_{i=1}^{N} x_i^2 \pi_i(\beta)(1 - \pi_i(\beta)) \end{bmatrix}, \tag{3.4}$$

gdzie $\pi_i(\beta) = \frac{\exp(\beta_1 + \beta_2 x_i)}{1 + \exp(\beta_1 + \beta_2 x_i)}$, a N to liczba obserwacji.

Wtedy aktualizacja kandydata na miejsce zerowe w k-tym kroku algorytmu optymalizacyjnego dla kolejnych metod omówionych w rozdziale (3.1) wyraża się poniższymi wzorami

Metoda spadku wzdłuż gradientu I (Cauchy'ego)

$\alpha_k \in \mathbb{R}$

$$\beta_k = \beta_{k-1} + \alpha_k \cdot \Big(\sum_{i=1}^{N} (y_i - \pi_i(\beta_{k-1})), \sum_{i=1}^{N} x_i (y_i - \pi_i(\beta_{k-1})) \Big).$$

Metoda spadku wzdłuż gradientu II (Newtona-Raphsona)

$$\beta_k = \beta_{k-1} - \mathscr{I}(\beta_{k-1})^{-1} \cdot \Big(\sum_{i=1}^{N} (y_i - \pi_i(\beta_{k-1})), \sum_{i=1}^{N} x_i (y_i - \pi_i(\beta_{k-1})) \Big),$$

gdzie $\mathscr{I}(\beta_{k-1})$ zdefiniowane jest we wzorze 3.4.

Metoda stochastycznego spadku wzdłuż gradientu I

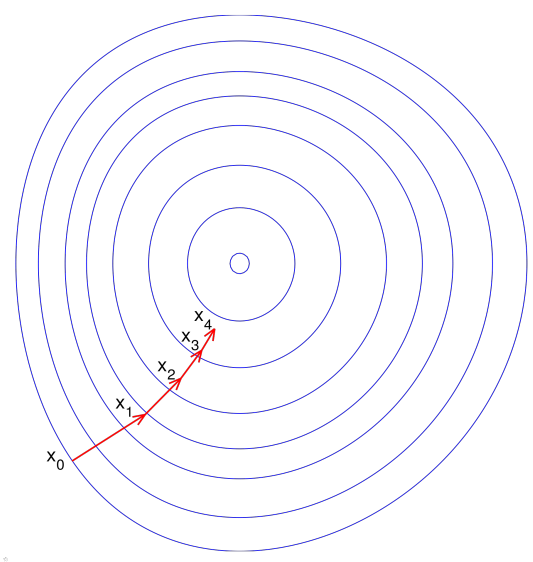
$$\beta_k = \beta_{k-1} + \alpha_k \cdot \Big(y_i - \pi_i(\beta_{k-1}), x_i (y_i - \pi_i(\beta_{k-1})) \Big),$$

dla wylosowanego w danym kroku i.

Ponieważ algorytmy te znajdują minimum funkcji, a docelowo szukane jest maksimum, stąd wykorzystano przeciwieństwo funkcji log-wiarogodności, dlatego w wyżej wymienionych wzorach zmieniono znaki przed pochodnymi na przeciwne.

Poniższymi wywołaniami kodów z pakietu $\mathcal{R}$ [31] można otrzymać wartości ekstremum w danym kroku kolejnych algorytmów. Kody funkcji `logitGD()` dostępne są w Dodatku B.

```
logitGD(mtcars$vs, mtcars$mpg, optim.method = "GDI", beta_0 = c(-8,0),
        alpha=function(t){1/3})$steps
logitGD(mtcars$vs, dd, optim.method = "GDII",
        beta_0 = c(0.43, -8.8))$steps
logitGD(mtcars$vs[sample(nrow(mtcars))],
        mtcars$mpg[sample(nrow(mtcars))],
        optim.method = "SGDI", eps=1e-7, beta_0 = c(0,0),
        alpha=function(t){1/nrow(mtcars)})$steps
```



Rysunek 3.1: Porównanie algorytmów spadku gradientu.

Rozdział 4

Estymacja w modelu Coxa metodą stochastycznego spadku gradientu

Poniższy rozdział przedstawia implementację oraz zastosowanie metody stochastycznego spadku gradientu do estymacji współczynników w modelu proporcjonalnych hazardów Coxa. Jest to główny cel pracy. Poniższe rozważania odnośnie podejścia do stosowania tej metody w tym konkretnym modelu nie są oparte na żadnej literaturze ze względu na jej brak.

W czasie powstawania pracy nawiązano jedynie nieznaczną wymianę informacji z pracownikami *Harvard Laboratory for Applied Statistical Methodology & Data Science*, dzięki której dowiedziano się że podjęte zostały kroki w kierunku stworzenia podwalin pod teorię do omawianego zagadnienia, jednak ewentualne publikacje nie zostały jeszcze dokończone. Przedsmak niedokończonej implementacji algorytmu przez pracowników wyżej wymienionego laboratorium można znaleźć w [36].

W procesie estymacji współczynników w omawianym modelu wykorzystano pochodne cząstkowe częściowej funkcji log-wiarogodności (2.10)

$$U_k(\beta) = \frac{\partial \ell_k(\beta)}{\partial \beta_k} = \sum_{i=1}^{K} \Big(X_{ik} - \frac{\sum\limits_{l \in \mathscr{R}(t_i)} X_{lk} e^{X_l' \beta}}{\sum\limits_{l \in \mathscr{R}(t_i)} e^{X_l' \beta}} \Big) = \sum_{i=1}^{n} Y_i \Big(X_{ik} - \frac{\sum\limits_{l \in \mathscr{R}(t_i)} X_{lk} e^{X_l' \beta}}{\sum\limits_{l \in \mathscr{R}(t_i)} e^{X_l' \beta}} \Big) = \sum_{i=1}^{n} U_{k_i}(\beta),$$

(dla $Y_i = 1$, gdy obserwacja nie była cenzurowana i $Y_i = 0$, gdy obserwacja była cenzurowana; K to liczba zdarzeń; n to liczba obserwacji) oraz wzór (3.1), którego wersja z adekwatnymi oznaczeniami dla powyższej funkcji wygląda następująco

$$\beta_{k_{j+1}} = \beta_{k_j} - \alpha_j U_{k_i}(\beta_{k_j}), \tag{4.1}$$

gdzie j oznacza krok algorytmu, i iteruje składniki U_k, α_j to długość j-tego kroku algorytmu, zaś U_k to k-ta pochodna cząstkowa gradientu U oraz β_k to k-ta współrzędna estymowanego wektora współczynników β o rozmiarze p, czyli $k = 1, \ldots, p$.

Bez straty ogólności, w celu uproszczenia oznaczeń, można założyć, że obserwacje są uporządkowane rosnąco względem czasu zdarzeń. Ostatecznie z tego nie korzystam.

4.1. Porównanie z innymi modelami

Matematycy i informatycy często uważają, że idea stochastycznego spadku gradientu polega na losowaniu składnika optymalizowanej funkcji. Jednak ze statystycznego punktu widzenia, metoda stochastycznego spadku gradientu opiera się o losowanie indeksu obserwacji ze zbioru, z którego uczony jest algorytm, zanim postanowi się w jakikolwiek sposób przedstawić konkretną funkcję wiarogodności. Zatem w celu estymacji w oparciu o stochastyczny spadek gradientu w modelu Coxa konstruując metodę należy najpierw losować obserwacje a następnie dopiero wyznaczać formę optymalizowanej funkcji częściowej log-wiarogodności.

Dla wielu modeli opierających się o funkcje wiarogodności te dwa punkty widzenia są równoważne, jednak dla modelu Coxa nie. W przypadku modelu ADALINE sformułowanego jak w [40], opartego na minimalizowaniu funkcji kosztu w postaci błędu najmniejszych kwadratów, w [6] podano postać funkcji straty oraz równanie algorytmu stochastycznego spadku gradientu jak poniżej

$$Q_{adaline} = \frac{1}{2}(y - w'\Phi(x))^2,$$
$$w \leftarrow w + \alpha_k(y_k - w'\Phi(x_t))\Phi(x_t),$$
$$\Phi(x) \in \mathbb{R}^d, y \in \{-1, 1\},$$

dla których widać, że w kolejnych krokach algorytmu t wystarczy tylko jedna obserwacja $z_t = (x_t, y_t)$ aby poprawić oszacowanie parametru w.

W modelu proporcjonalnych hazardów Coxa jest to bardziej skomplikowane.

4.2. Implementacja

Dla zadanej z góry funkcji hazardu, częściowa funkcja wiarogodności odpowiada prawdopodobieństwu tego, że obserwowane zdarzenia zdarzyłyby się dokładnie w tej kolejności w jakiej się pojawiły. To prawdopodobieństwo zależy od wszystkich obserwacji w zbiorze. Niemożliwe jest wyliczenie tego poprzez obliczenie wartości funkcji częściowej wiarogodności oddzielnie dla obserwacji o numerach od 1 do 5 i oddzielnie dla obserwacji od numerach 6 do 10, a następnie przemnożeniu wyników przez siebie. Z tej przyczyny niemożliwe jest losowanie czynników optymalizowanej funkcji przy użyciu metody stochastycznego spadku gradientu do estymacji współczynników w tym modelu. Jednak możliwe jest losowanie podzbioru obserwacji a następnie konstruowanie funkcji wiarogodności dla zaobserwowanego zredukowanego zbioru.

Taki sposób wprowadzania obserwacji do estymacji można wykorzystać w sytuacjach, gdy mamy do czynienia z nieskończonym napływem nowych obserwacji a interesują nas oszacowania estymowanych parametrów modelu $\beta_k, k = 1, \ldots, p$ dla obecnie zaobserwowanych i wykorzystanych obserwacji. Proces ten ma dwie zalety: nie dość, że dla nowych obserwacji model jest w stanie na bazie obecnych oszacowań parametrów dokonać predykcji proporcji hazardów, to dodatkowo po każdej porcji obserwacji aktualizuje parametry modelu.

Omówiona w tym rozdziale metoda estymacji metodą stochastycznego spadku gradientu dla modelu proporcjonalnych hazardów Coxa została zaimplementowana w języku $\mathcal{R}$ [31]. Kod wraz z dokumentacją stworzonej funkcji w języku angielskim oraz opisem poszczególnych kroków algorytmu zawarty jest poniżej.

Algorytm jest dostępny w specjalnie przygotowanym pakiecie o nazwie `coxphSGD` napisanym w języku $\mathcal{R}$, który można pobrać z internetu i zainstalować poleceniem

```
if (packageVersion("devtools") < 1.6) {
  install.packages("devtools")
}
devtools::install_github("MarcinKosinski/Cox-SGD")
```

Poniżej przedstawione są argumenty, które przyjmuje funkcja `coxphSGD()`, która estymuje współczynniki w modelu proporcjonalnych hazardów Coxa metodą stochastycznego spadku gradientu. Starano zachować jednorodność kolejności i nazewnictwa parametrów z funkcją `coxph` z pakietu `survival` [37], [38].

```
#' Stochastic Gradient Descent log-likelihood estimation in
#' Cox proportional hazards model
#'
#' Function \code{coxphSGD} estimates coefficients using stochastic
#' gradient descent algorithm in Cox proportional hazards model.
#'
#' @param formula a formula object, with the response on the left of a ~ operator,
#' and the terms on the right. The response must be a survival object as returned by
#' the Surv function.
#' @param data a data.frame in which to interpret the variables named in the \code{formul
#' @param reorderObs a logical value telling whether reorder observations at each epoch.
#' when order of observations in estimation should be randomly generated.
#' @param learningRates a function specifing how to define learning rates in
#' steps of the algorithm. By default the \code{f(t)=1/t} is used, where \code{t} is
#' the number of algorithm's step.
#' @param beta_0 a numeric vector (if of length 1 then will be replicated) of length
#' equal to the number of variables after using \code{formula} in the \code{model.matrix}
#' function
#' @param epsilon a numeric value with the stop condition of the estimation algorithm.
#' @param epoch a numeric value declaring the number of epoches to run for the
#' estimation algorithm in the stochastic gradient descent.
#' @param batchSize a numeric value specifing the size of a batch set to take from
#' the reordered dataset to update the coefficients in one step of an algorithm.
#'
#' @note If one of the conditions is fullfiled
#' \itemize{
#'  \item \eqn{||\beta_{j+1}-\beta_{j}|| <}\code{epsilon} parameter for any \eqn{j}
#'  \item \eqn{#epochs >} \code{epochs} parameter
#' }
#' the estimation process is stopped.
#' @export
#' @importFrom survival Surv
#' @importFrom assertthat assert_that
#' @examples
#' library(survival)
#' \dontrun{
#' coxphSGD(Surv(time, status) ~ ph.ecog + age, data=lung)
```

```
#' }
#'
```

Sprawdzenie parametrów na wejściu funkcji.

```
coxphSGD = function(formula, data, reorderObs = TRUE,
                    learningRates = function(x) 1/x,
                    beta_0 = 0, epsilon = 1e-5,
                    batchSize = 10, epoch = 20 ) {

  assert_that(is.data.frame(data))
  assert_that(is.logical(reorderObs))
  assert_that(is.function(learningRates))
  assert_that(is.numeric(epsilon))
  assert_that(is.numeric(epoch) & epoch > 0)
```

Identyfikacja przekazanych parametrów. Poniższa identyfikacja bazuje na kodzie funkcji `coxph()`.

```
  Call <- match.call()
  indx <- match(c("formula", "data", "order", "learningRates",
                  "epsilon", "batchsize", "epoch"),
                names(Call), nomatch = 0)
  if (indx[1] == 0)
      stop("A formula argument is required")
  temp <- Call[c(1, indx)]
  temp[[1]] <- as.name("model.frame")

  mf <- eval(temp, parent.frame())
  Y <- model.extract(mf, "response")

  if (!inherits(Y, "Surv"))
      stop("Response must be a survival object")
  type <- attr(Y, "type")

  if (type != "right" && type != "counting")
      stop(paste("Cox model doesn't support \"", type, "\" survival data",
          sep = ""))
  if (length(beta_0) == 1) {
    beta_0 <- rep(beta_0, ncol(mf)-1)
  }
```

Początkowa zamiana kolejności obserwacji w wejściowym zbiorze.

```
  if (reorderObs) {
    obsOrder <- sample(1:nrow(data))
    mf <- mf[obsOrder, ]
    Y <- Y[obsOrder, ]
  }
```

Wprowadzenie zmiennych pomocniczych.

```
  j <- 0 # number of an algorithm's step
  diff <- 0 # differences between estimates along steps
```

```
i <- 0 # indicator of a batch sample
n <- nrow(data)
batchSamplesStarts <- seq(1,n, batchSize) # indexes of starts of batch samples
epochs_n <- 1# indicator of the present epochs number
beta_j <- beta_0
```

gdzie

$$\text{diff}_j = \| \beta_{j+1} - \beta_j \| < \epsilon.$$

Sprawdzenie warunku zbieżności algorytmu.

```
while ( j == 0 | (diff < eps & epochs_n <= epoch) ){
  j <- j+1
  i <- i+1
```

Rozpoczęcie algorytmu. Dla losowej kolejności obserwacji, weź pierwszą porcję obserwacji, której wielkość ustawiona jest dzięki parametrowi `batchSize`. Tak powstaje podzbiór obserwacji oznaczany przez $\mathcal{B}$, dla którego zachodzi $|\mathcal{B}| = b$, a b odpowiada wartości ustawionej w parametrze `batchSize`. Indeksy obserwacji należace do zbioru $\mathcal{B}$ zdefiniujmy jako $\mathcal{B}_{\text{ind}} = \{i : X_i \in \mathcal{B}\}$.

```
if (i < length(batchSamplesStarts)-1){
  batchSample_variables <- mf[batchSamplesStarts[i]:(batchSamplesStarts[i]+batchSize-1)
  batchSample_response <- Y[batchSamplesStarts[i]:(batchSamplesStarts[i]+batchSize-1),
} else {
  if (i == length(batchSamplesStarts)-1) {
    # last batch sample can me shorter than all others
    batchSample_variables <- mf[batchSamplesStarts[i]:(n), ]
    batchSample_response <- Y[batchSamplesStarts[i]:(n), ]
  } else {
    i <- 1
    batchSample_variables <- mf[batchSamplesStarts[i]:(batchSamplesStarts[i]+batchSize-
    batchSample_response <- Y[batchSamplesStarts[i]:(batchSamplesStarts[i]+batchSize-1)
    epochs_n <- epochs_n + 1 # epoch has passed
    # so reorder samples
      if (reorderObs) {
        obsOrder <- sample(1:nrow(data))
        mf <- mf[obsOrder, ]
        Y <- Y[obsOrder, ]
      }
  }
}
```

Dla danego podzbioru wyznacz odpowiadającą jemu część pochodnej częściowej funkcji log-wiarogoności ze zmienionym znakiem. Ponieważ omawiane algorytmy rozwiązuja problem minimalizacji badanej funkcji, zaś celem estymacji w modelu Coxa jest znalezienie parametrów modelu maksymalizujących funkcję częściowej log-wiarogodności, zatem wzięcie do minimalizacji funkcji z przeciwnym znakiem doprowadzi do wykorzystania metod znajdujących minimum do znalezienia maksimum. Dla j-ego kroku algorytmu i k-tej pochodnej cząstkowej dysponuje się podzbiorem $\mathcal{B}$ o liczności b (parametr `batchSize`), wtedy

$$-U_k^{\mathcal{B}}(\beta_j) = -\sum_{i \in \mathcal{B}_{\text{ind}}} U_{k_i}^{\mathcal{B}}(\beta_j) = -\sum_{i \in \mathcal{B}_{\text{ind}}} Y_i \Big(X_{ik} - \frac{\sum\limits_{l \in \mathscr{R}_{\mathcal{B}}(t_i)} X_{lk} e^{X_l' \beta_j}}{\sum\limits_{l \in \mathscr{R}_{\mathcal{B}}(t_i)} e^{X_l' \beta_j}} \Big),$$

gdzie b oznacza wielkość podzbioru $\mathcal{B}$, zaś $\mathscr{R}_{\mathcal{B}}(t_i)$ to zbiór ryzyka dla podzbioru $\mathcal{B}$ w czasie t_i.

```
  U_ik <- matrix(0, ncol = ncol(mf)-1,
                 nrow = nrow(batchSample_variables))
  U_k <- numeric(ncol(mf)-1)
  for ( k in 2:ncol(mf)) { # 1st dimension is Y
    for (i in 1:nrow(batchSample_variables)){
      l <- which(batchSample_response[, 1] <= batchSample_response[i, 1])
      U_ik[i,k] <- -batchSample_variables[i, k] + sum(batchSample_variables[l, k]*
                                              exp(batchSample_variables[l, ]%*%beta_j)/
            sum(exp(batchSample_variables[l, ]%*%beta_j))

    }
  U_k[k] <- sum(U_ik[, k])
  }
```

Następnie zaktualizuj parametry modelu.

$$\beta_{k_{j+1}} = \beta_{k_j} - \alpha_j U_k^{\mathcal{B}}(\beta_{k_j})$$

```
   beta_j <- beta_j - learningRates(j)*U_k
```

Przypisz nowy warunek stopu.

```
   diff <- sqrt(sum(learningRates(j)*U_k))
  }
```

Zwrócenie parametrów funkcji, gdy spełniony chociaż jeden warunek stopu.

```
  fit <- list()
  fit$Call <- Call
  fit$mf <- mf
  fit$coeff <- beta_j
  fit$epochs_n <- epochs_n
  fit
}
#coxphSGD(Surv(time, status) ~ ph.ecog + age, data=lung)
```

4.3. Dwuwymiarowy przykład

4.4. Porównanie z innymi algorytmami spadku gradientu

Rozdział 5

Analiza danych genomicznych

5.1. Opis i pobranie danych

5.2. Analiza

Dodatek A

Wykorzystane narzędzia

Dodatek B

Kody w R

```
logitGD <- function(y, x, optim.method = "GDI", eps = 10e-4,
                    max.iter = 100, alpha = function(t){1/t}, beta_0 = c(0,0)){
  stopifnot(length(y) == length(x) & optim.method %in% c("GDI", "GDII", "SGDI")
            & is.numeric(c(max.iter, eps, x)) & all(c(eps, max.iter) > 0) &
              is.function(alpha))
  iter <- 0
  err <- list()
  err[[iter+1]] <- eps+1
  w_old <- beta_0

  res <-list()
  while(iter < max.iter && (abs(err[[ifelse(iter==0,1,iter)]]) > eps)){

    iter <- iter + 1
    if (optim.method == "GDI"){
      w_new <- w_old + alpha(iter)*updateWeightsGDI(y, x, w_old)
    }
    if (optim.method == "GDII"){
      w_new <- w_old - as.vector(inverseHessianGDII(x, w_old)%*%
                                   updateWeightsGDI(y, x, w_old))
    }
    if (optim.method == "SGDI"){
      w_new <- w_old + alpha(iter)*updateWeightsSGDI(y[iter], x[iter], w_old)
    }
    res[[iter]] <- w_new
    err[[iter]] <- sqrt(sum((w_new - w_old)^2))

    w_old <- w_new

  }
  return(list(steps = c(list(beta_0),res), errors = c(list(c(0,0)),err)))
}

updateWeightsGDI <- function(y, x, w_old){
  (1/length(y))*c(sum(y-p(w_old, x)), sum(x*(y-p(w_old, x))))
```

```
}

updateWeightsSGDI <- function(y_i, x_i, w_old){
  c(y_i-p(w_old, x_i), x_i*(y_i-p(w_old, x_i)))
}

p <- function(w_old, x_i){
  1/(1+exp(-w_old[1]-w_old[2]*x_i))
}

inverseHessianGDII <- function(x, w_old){
  solve(
    matrix(c(
      sum(p(w_old, x)*(1-p(w_old, x))),
      sum(x*p(w_old, x)*(1-p(w_old, x))),
      sum(x*p(w_old, x)*(1-p(w_old, x))),
      sum(x*x*p(w_old, x)*(1-p(w_old, x)))
    ),
    nrow =2 )
  )
}
```

Dodatek C

Dokumentacja pakietu RTCGA

Package 'RTCGA'

October 25, 2015

Title The Cancer Genome Atlas Data Integration

Version 1.0.0

Date 2015-08-05

Author Marcin Kosinski <m.p.kosinski@gmail.com>, Przemyslaw Biecek <przemyslaw.biecek@gmail.com>

Maintainer Marcin Kosinski <m.p.kosinski@gmail.com>

Description The Cancer Genome Atlas (TCGA) Data Portal provides a platform for researchers to search, download, and analyze data sets generated by TCGA. It contains clinical information, genomic characterization data, and high level sequence analysis of the tumor genomes. The key is to understand genomics to improve cancer care. RTCGA package offers download and integration of the variety and volume of TCGA data using patient barcode key, what enables easier data possession. This may have an benefcial infuence on impact on development of science and improvement of patients' treatment. Furthermore, RTCGA package transforms TCGA data to tidy form which is convenient to use.

BugReports https://github.com/MarcinKosinski/RTCGA/issues

License GPL-2

LazyLoad yes

LazyData yes

Depends R (>= 3.2.0), knitr

Imports XML, assertthat, stringi, rvest, data.table, magrittr, xml2

Suggests testthat, pander

Repository Bioconductor

biocViews Software

VignetteBuilder knitr

NeedsCompilation no

R topics documented:

RTCGA-package 2
checkTCGA 3

datasetsTCGA 5
downloadTCGA 6
infoTCGA 7
readTCGA 7

Index **10**

RTCGA-package *The Caner Genome Atlas data integration*

Description

The Cancer Genome Atlas (TCGA) Data Portal provides a platform for researchers to search, download, and analyze data sets generated by TCGA. It contains clinical information, genomic characterization data, and high level sequence analysis of the tumor genomes. The key is to understand genomics to improve cancer care. RTCGA package offers download and integration of the variety and volume of TCGA data using patient barcode key, what enables easier data possession. This may have an benefcial infuence on impact on development of science and improvement of patients' treatment. Furthermore, RTCGA package transforms TCGA data to form which is convenient to use in R statistical package. Those data transformations can be a part of statistical analysis pipeline which can be more reproducible with RTCGA

Details

For more detailed information visit **RTCGA** `wiki` on Github.

Author(s)

Marcin Kosinski [aut, cre] < m.p.kosinski@gmail.com >
Przemyslaw Biecek [aut] < przemyslaw.biecek@gmail.com >

See Also

Other RTCGA: checkTCGA; datasetsTCGA; downloadTCGA; infoTCGA; readTCGA

Examples

```
## Not run:
browseVignettes('RTCGA')

## End(Not run)
```

checkTCGA *Information about datasets from TCGA project*

Description

The checkTCGA function let's to check

- DataSets: TCGA datasets' names for current release date and cohort.
- Dates: TCGA datasets' dates of release.

Usage

```
checkTCGA(what, cancerType, date = NULL)
```

Arguments

what	One of DataSets or Dates.
cancerType	A character of length 1 containing abbreviation (Cohort code - http://gdac.broadinstitute.org/) of types of cancers to check for.
date	A NULL or character specifying from which date informations should be checked. By default (date = NULL) the newest available date is used. All available dates can be checked on http://gdac.broadinstitute.org/runs/ or by using checkTCGA('Dates') function. Required format 'YYYY-MM-DD'.

Details

- If what='DataSets' enables to check TCGA datasets' names for current release date and cohort.
- If what='Dates' enables to check dates of TCGA datasets' releases.

Value

- If what='DataSets' a vector of available datasets' names to pass to the downloadTCGA function.
- If what='Dates' a vector of available dates to pass to the downloadTCGA function.

See Also

Other RTCGA: RTCGA, RTCGA-package; datasetsTCGA; downloadTCGA; infoTCGA; readTCGA

Examples

```
## Not run:

##############################

# names for current release date and cohort
checkTCGA('DataSets', 'BRCA' )
checkTCGA('DataSets', 'OV', tail(checkTCGA('Dates'))[1] )
#checkTCGA('DataSets', 'OV', checkTCGA('Dates')[5] ) # error
```

```
# dates of TCGA datasets' releases.
checkTCGA('Dates')

############################

# TCGA datasets' names availability for
# current release date and cancer type.

releaseDate <- '2015-06-01'
cancerTypes <- c('OV', 'BRCA')

cancerTypes %>% sapply(function(element){
  grep(x = checkTCGA('DataSets', element, releaseDate),
      pattern = 'humanmethylation450', value = TRUE) %>%
       as.vector()
       })

############################

# TCGA genes' names and availability
# in Merge_rnaseqv2__... dataset
dir.create('data2')
sapply( cancerTypes, function(element){
tryCatch({
    downloadTCGA( cancerTypes = element,
                  dataSet = paste0('rnaseqv2__illuminahiseq_rnaseqv2__unc',
                  '_edu__Level_3__RSEM_genes_normalized__data.Level'),
                  destDir = 'data2',
                  date = releaseDate )},
    error = function(cond){
        cat('Error: Maybe there weren't rnaseq data for ', element, ' cancer.\n')
    }
)
})

# Paths to rna-seq data

sapply( cohorts, function( element ){
folder <- grep( paste0( '(_',element,'\\.', '|','_',element,'-FFPE)', '.*rnaseqv2'),
               list.files('data2/'), value = TRUE)
file <- grep( paste0(element, '.*rnaseqv2'), list.files( paste0( 'data2/',folder ) ),
              value = TRUE)
path <- paste0( 'data2/', folder, '/', file )
assign( value = path, x = paste0(element, '.rnaseq.path'), envir = .GlobalEnv)
})

rnaseqDir <- 'OV.rnaseq'


fread(rnaseqDir, select = c(1),
      data.table = FALSE,
      colClasses = 'character')[-1, 1]


## End(Not run)
```

datasetsTCGA *RTCGA.data - The Family of R Packages with Data from The Cancer Genome Atlas Study*

Description

Snapshots of the clinical, mutations, cnvs and rnaseq datasets from the 2015-06-01 release date are included in the RTCGA.data family that contains 4 packages:

- **RTCGA.rnaseq**
- **RTCGA.clinical**
- **RTCGA.mutations**
- **RTCGA.cnv**

Details

For more detailed information visit **RTCGA.data** website.

Author(s)

Marcin Kosinski [aut, cre] < m.p.kosinski@gmail.com >
Przemyslaw Biecek [aut] < przemyslaw.biecek@gmail.com >

See Also

Other RTCGA: RTCGA, RTCGA-package; checkTCGA; downloadTCGA; infoTCGA; readTCGA

Examples

```
# installation of packages containing snapshots
# of TCGA project's datasets

## Not run:
source('http://bioconductor.org/biocLite.R')
biocLite(RTCGA.clinical)
biocLite(RTCGA.mutations)
biocLite(RTCGA.rnaseq)
biocLite(RTCGA.cnv)

# use cases and examples + more data info
browseVignettes('RTCGA')

## End(Not run)
```

downloadTCGA *Download TCGA data*

Description

Enables to download TCGA data from specified dates of releases of concrete Cohorts of cancer types. Pass a name of required dataset to the dataSet parameter. By default the Merged Clinical dataSet is downloaded (value dataSet = 'Merge_Clinical.Level_1') from the newest available date of the release.

Usage

```
downloadTCGA(cancerTypes, dataSet = "Merge_Clinical.Level_1", destDir,
  date = NULL, untarFile = TRUE, removeTar = TRUE)
```

Arguments

cancerTypes	A character vector containing abbreviations (Cohort code) of types of cancers to download from http://gdac.broadinstitute.org/. For easy access from R check details below.
dataSet	A part of the name of dataSet to be downloaded from http://gdac.broadinstitute.org/runs/. By default the Merged Clinical dataSet is downloaded (value dataSet = 'Merge_Clinical.Level_1 Available datasets' names can be checked using checkTCGA function.
destDir	A character specifying a directory into which dataSets will be downloaded.
date	A NULL or character specifying from which date dataSets should be downloaded. By default (date = NULL) the newest available date is used. All available dates can be checked on http://gdac.broadinstitute.org/runs/ or by using checkTCGA function. Required format 'YYYY-MM-DD'.
untarFile	Logical - should the downloaded file be untarred. Default is TRUE.
removeTar	Logical - should the downloaded .tar file be removed after untarring. Default is TRUE.

Details

All cohort names can be checked using: sub(x = names(infoTCGA()), '-counts', '').

Value

No values. It only downloads files.

See Also

Other RTCGA: RTCGA, RTCGA-package; checkTCGA; datasetsTCGA; infoTCGA; readTCGA

Examples

```
## Not run:
dir.create( 'hre')

downloadTCGA( cancerTypes = 'ACC', dataSet = 'miR_gene_expression',
destDir = 'hre', date =  tail( checkTCGA('Dates'), 2 )[1] )


downloadTCGA( cancerTypes = c('BRCA', 'OV'), destDir = 'hre',
 date = tail( checkTCGA('Dates'), 2 )[1] )

## End(Not run)
```

`infoTCGA` *Information about cohorts from TCGA project*

Description

Function restores codes and counts for each cohort from TCGA project.

Usage

```
infoTCGA()
```

Value

A list with a tabular information from http://gdac.broadinstitute.org/.

See Also

Other RTCGA: `RTCGA`, `RTCGA-package`; `checkTCGA`; `datasetsTCGA`; `downloadTCGA`; `readTCGA`

Examples

```
infoTCGA()

(cohorts <- infoTCGA() %>%
rownames() %>%
   sub('-counts', '', x=.))
```

`readTCGA` *Read TCGA data to the tidy format*

Description

`readTCGA` function allows to read unzipped files:

- clinical data - `Merge_Clinical.Level_1`
- rnaseq data (genes' expressions) - `Mutation_Packager_Calls.Level`
- genes' mutations data - `rnaseqv2__illuminahiseq_rnaseqv2`

from TCGA project. Those files can be easily downloded with downloadTCGA function. See examples.

Usage

```
readTCGA(path, dataType, ...)
```

Arguments

path	If `dataType = 'clinical'` a directory to a `cancerType.clin.merged.txt` file. If `dataType = 'mutations'` a directory to the unzziped folder `Mutation_Packager_Calls.Lev` containing `.maf` files. If `dataType = 'rnaseq'` a directory to the uzziped file `rnaseqv2__illuminahiseq_rnaseqv2__unc_edu__Level_3__RSEM_genes_normalized__data.` See examples.
dataType	One of `'clinical'`, `'rnaseq'`, `'mutations'` depending on which type of data users is trying to read in the tidy format.
...	Further arguments passed to the as.data.frame.

Details

All cohort names can be checked using: `sub( x = names( infoTCGA() ), '-counts', '')`.

Value

An output:

- If `dataType = 'clinical'` a `data.frame` with clinical data.
- If `dataType = 'rnaseq'` a `data.frame` with rnaseq data.
- If `dataType = 'mutations'` a `data.frame` with mutations data.

See Also

Other RTCGA: RTCGA, RTCGA-package; checkTCGA; datasetsTCGA; downloadTCGA; infoTCGA

Examples

```
## Not run:

##############
##### clinical
##############

dir.create('hre')

# downloading clinical data
downloadTCGA( cancerTypes = c('BRCA', 'OV'), destDir = 'hre' )


# reading datasets
sapply( c('BRCA', 'OV'), function( element ){
folder <- grep( paste0( '(_',element,'\\.', '|','_',element,'-FFPE)', '.*Clinical'),
      list.files('hre/'),value = TRUE)
path <- paste0( 'hre/', folder, '/', element, '.clin.merged.txt')
assign( value = readTCGA( path, 'clinical' ),
         x = paste0(element, '.clin.data'), envir = .GlobalEnv)
         })

##############
```

```
##### rnaseq
##############

dir.create('data2')

# downloading rnaseq data
downloadTCGA( cancerTypes = 'BRCA',
dataSet = 'rnaseqv2__illuminahiseq_rnaseqv2__unc_edu__Level_3__RSEM_genes_normalized__data.Level',
destDir = 'data2' )

# shortening paths and directories
list.files( 'data2/') %>%
    file.path( 'data2', .) %>%
    file.rename( to = substr(.,start=1,stop=50))

# reading data
list.files( 'data2/') %>%
    file.path( 'data2', .) -> folder

folder %>%
    list.files %>%
    file.path( folder, .) %>%
    grep( pattern = 'illuminahiseq', x = ., value = TRUE) -> pathRNA
readTCGA( path = pathRNA, dataType = 'rnaseq' ) -> my_data


##############
##### mutations
##############

dir.create('data3')


downloadTCGA( cancerTypes = 'OV',
              dataSet = 'Mutation_Packager_Calls.Level',
              destDir = 'data3' )

# reading data
list.files( 'data3/') %>%
    file.path( 'data3', .) -> folder

readTCGA(folder, 'mutations') -> mut_file


## End(Not run)
```

Index

as.data.frame, *8*

checkTCGA, *2*, 3, *5–8*

datasetsTCGA, *2, 3*, 5, *6–8*
downloadTCGA, *2, 3*, *5*, 6, *7, 8*

infoTCGA, *2, 3*, *5, 6*, 7, *8*

readTCGA, *2, 3*, *5–7*, 7
RTCGA, *3*, *5–8*
RTCGA *(*RTCGA-package*)*, 2
RTCGA-package, 2

Literatura

[1] Aldrich J., (1997) *R. A. Fisher and the Making of Maximum Likelihood 1912 – 1922*, Statistical Science 1997, Vol. 12, No. 3, 162-176.

[2] Asselain B., Mould R. F., (2010) *Methodology of the Cox proportional hazards model*, Journal of Oncology 2010, volume 60, Number 5, 403–409.

[3] Biecek P., (2011) *Przewodnik po pakiecie R*, Rozprawa doktorska, Oficyna Wydawnicza GiS, wydanie II.

[4] Bottou L., (2010) *Large-Scale Machine Learning with Stochastic Gradient Descent.*

[5] Bottou L., (1998), *Online Learning and Stochastic Approximations.*

[6] Bottou L., (2012) *Stochastic Gradient Descent Tricks.*

[7] Burzykowski T., (2015?) *Notatki do przedmiotu Biostatystyka*, `https://e.mini.pw.edu.pl/sites/default/files/biostatystyka.pdf`.

[8] Cox D. R. (1962) *Renewal Theory. Methuen Monograph on Applied Probability & Statistics*, London: Methuen.

[9] Cox D. R., (1972) *Regression models and life-tables (with discussion)*, Journal of the Royal Statistical Society Series B 34:187-220..

[10] Czepiel S. A., *Maximum Likelihood Estimation of Logistic Regression Models: Theory and Implementation*, `http://czep.net/stat/mlelr.pdf`.

[11] Dennis J. E. Jr., Schnabel R. B., (1983), *Numerical Methods For Un- constrained Optimization and Nonlinear Equations*, Prentice-Hall.

[12] Dobson A. J., (2002) *An Introduction to Generalized Linear Models*, Wydanie II, Chapman & Hall/CRC.

[13] Finkel J. R., Kleeman A., Manning C. D., (2008), *Efficient, Feature-based, Conditional Random Field Parsing*, Proc. Annual Meeting of the ACL.

[14] Fisher R. A., (1912) *An absolute criterion for fitting frequency curves.*

[15] Fisher R. A., (1922) *On the mathematical foundations of theoretical statistics*, Philos. Trans. Roy. Soc. London Ser. A 222 309-368.

[16] Fortuna Z., Macukow B., Wąsowski J., (2006), *Metody Numeryczne*, Wydawnictwa Naukowo-Techniczne.

[17] Gauss C. F., (1809) *Theoria Motus Corporum Coelestium.*

[18] Gągolewski M., (2014) *Programowanie w języku R*, Wydawnictwo Naukowe PWN.

[19] Hald A., (1949) *Maximum likelihood estimation of the parameters of a normal distribution which is truncated at a known point*, Skandinavisk Aktuarietidskrift, 119-134.

[20] Hutchinson J. B., (1928) *The Application of the "Method of Maximum Likelihood" to the Estimation of Linkage*, Genetics. 1929 Nov; 14(6): 519–537.

[21] Kenward M. G., Lesaffre E. and Molenberghs G., (1994) *An Application of Maximum Likelihood and Generalized Estimating Equations to the Analysis of Ordinal Data from a Longitudinal Study with Cases Missing at Random*, Biometrics Vol. 50, No. 4 (Dec., 1994), pp. 945-953.

[22] Kotłowski W., (2012), Notatki do przedmiotu *Techniki Optymalizacji* prowadzonego na Politechnice Poznańskiej,
`http://www.cs.put.poznan.pl/wkotlowski/teaching/wyklad3b.pdf`

[23] Legendre A. M., (1804) *Nouvelles mèthods pour la dètermination des orbites des comètes.*

[24] Milewski S., (2006) Konspekt do przedmiotu *Metody Numeryczne* prowadzonego na Politechnice Krakowskiej,
`http://l5.pk.edu.pl/images/skrypty/Metody_numeryczne_1`

[25] Millar R. B., (2011) *Maximum Likelihood Estimation and Inference: With Examples in R, SAS and ADMB, chapter 6. Some Widely Used Applications of Maximum Likelihood*, John Wiley & Sons, Ltd.

[26] Murata N., (1998), *A Statistical Study of On-line Learning. In Online Learning and Neural Networks*, Cambridge University Press.

[27] Niemiro W., (2011) Skrypt do przedmiotu *Statystyka* prowadzonego na Uniwersytecie Warszawskim,
`http://www-users.mat.umk.pl/~wniem/Statystyka/Statystyka.pdf`

[28] Norwegian Multicentre Study Group, (1981) *Timolol-induced reduction in mortality and reinfarction*, The New England Journal of Medicine; 304: 801-7.

[29] Panchenko D., (2006), Notatki do otwartego kursu MIT *Statistics for Applications, Lecture 2: Maximum Likelihood Estimators.*,
`http://ocw.mit.edu/courses/mathematics/18-443-statistics-for-applications-fall-2006/`

[30] Panchenko D., (2006), Notatki do otwartego kursu MIT *Statistics for Applications, Lecture 3: Properties of MLE: consistency, asymptotic normality. Fisher information.*,
`http://ocw.mit.edu/courses/mathematics/18-443-statistics-for-applications-fall-2006/`

[31] R Core Team, (2013) *R: A language and environment for statistical computing.* R Foundation for Statistical Computing, Wiedeń , ISBN 3-900051-07-0, `http://www.R-project.org/`.

[32] Robbins H. E., Siegmund. D. O., (1971), *A convergence theorem for non negative almost supermartingales and some applications*, In Proc. Sympos. Optimizing Methods in Statistics, pages 233–257, Ohio State University. Academic Press, New York.

[33] Rydlewski J., (2009) *Estymatory Największej Wiarogodności w Uogólnionych Modelach Regresji Nieliniowej*, Rozprawa doktorska.

[34] Sokołowski A., (2010) *Jak rozumieć i wykonywać analizę przeżycia http://www.statsoft.pl/Portals/0/Downloads/Jak_rozumiec_i_wykonac_analize_przezycia.pdf*

[35] Statistics Views, (2014) *"I would like to think of myself as a scientist, who happens largely to specialise in the use of statistics"– An interview with Sir David Cox.*

[36] Tran D., Lan T., Toulis P., (2015), *sgd: Stochastic Gradient Descent for Scalable Estimation. R package version 0.1*, `https://github.com/airoldilab/sgd`.

[37] Therneau T. M., Grambsch P. M., (2000), *Modeling Survival Data: Extending the Cox Model*, Springer.

[38] Therneau T. M., (2015), *A Package for Survival Analysis in S. version 2.38*, `http://CRAN.R-project.org/package=survival`.

[39] Widrow B., (1960), *An adaptive "ADALINE" neuron using chemical "memistors"*, Technical Report No. 1553-2, Stanford University.

[40] Widrow B., Ho M.E., (1960), *Adaptive switching circuits*, In: IRE WESCON Conv. Record, Part 4. pp. 96-104.

[41] Widrow B., Stearns S. D., (1985), *Adaptive Signal Processing, Prentice Hall.*

[42] Wikipedia, encyklopedia wolnego dostępu `wikipedia.pl`

[43] Woodcock S., (2014), Notatki do otwartego kursu Uniwersytetu Simona Frasera *ECON 837, Lecture 11 Asymptotic Properties of Maximum Likelihood Estimators*, `http://www.sfu.ca/ swoodcoc/teaching/sp2014/econ837/11.mle.pdf`

[44] Zieliński R., (1990) *Siedem wykładów wprowadzających do statystyki matematycznej*, Warszawa, Wydawnictwo Naukowe PWN.

Marcin Piotr Kosiński
Nr albumu 265361

Warszawa, 25 października 2015

Oświadczenie

Oświadczam, że pracę magisterską pod tytułem „Estymacja w modelu Coxa metodą stochastycznego spadku gradientu z przykładami zastosowań w analizie danych z The Cancer Genome Atlas", której promotorem jest prof. ndzw. dr hab. inż Przemysław Biecek wykonałem samodzielnie, co poświadczam własnoręcznym podpisem.

.....
Marcin Piotr Kosiński