Opracowanie algorytmu ewolucyjnego do wyboru modelu normalnego

Dokumentacja wstępna

Chojecki Przemysław, Przybyłek Paulina
Numery indeksu 298814, 298837

23.04.2022

Dokumentacja wstępna projektu w ramach przedmiotu
Wstęp do Algorytmów Ewolucyjnych
na wydziale **Matematyki i Nauk Informacyjnych**
Politechniki Warszawskiej

Spis treści

1 Wprowadzenie **1**
1.1 Streszczenie zawartości dokumentu 1
1.2 Cel projektu 1

2 Opis problemu **1**

3 Proponowane rozwiązanie **2**

4 Hipotezy **2**

5 Metodologia **2**

6 Technologia **3**

A Dodatek - Opis rodziny funkcji celu **4**

1 Wprowadzenie

Niniejszy dokument jest częścią projektu o charakterze badawczym, na który składają się jeszcze kod rozwiązujący problem oraz dokumentacja końcowa. Repozytorium projektu umieszczone zostanie na stronie GitHub ([4]), gdzie znajdować się będzie implementacja proponowanego rozwiązania oraz najnowsze wersje wszystkich przygotowywanych dokumentów w ramach tego projektu.

1.1 Streszczenie zawartości dokumentu

Dokument ten zawiera opis problemu wraz z propozycją algorytmu go rozwiązującego. Przedstawiono w nim również testy, które zostaną przeprowadzone, aby sprawdzić poprawność i jakość zaimplementowanych algorytmów, a także postawione hipotezy, które zostaną sprawdzone w czasie realizacji projektu. Dodatkowo na końcu dokumentu załączono dodatek A opisujący rodzinę funkcji rozważanych w pracy.

1.2 Cel projektu

Znajdowanie maksimum funkcji celu z rodziny opisanej w [1] jest przydatne przy modelowaniu danych rozkładem normalnym, co jest często spotykanym problemem w uczeniu maszynowym. Celem projektu jest opracowanie algorytmu ewolucyjnego dobrze radzącego sobie w szukaniu maksimum tych funkcji celu. Autorzy wspomnianej pracy ([1]) próbowali znajdować maksimum funkcji za pomocą algorytmu Metropolisa-Hastingsa. Zaproponowany w projekcie algorytm powinien działać lepiej niż opisany przez tych autorów.

2 Opis problemu

Każda funkcja z rodziny rozważanych funkcji celu przyjmuje jako argument permutację, a zwraca dodatnią liczbę rzeczywistą. Rodzina ta jest parametryzowana kilkoma parametrami: p, n, U, δ oraz D, gdzie $p \in \mathbb{N}$ to długość permutacji przyjmowanych jako argument funkcji, $n \in \mathbb{N}$, $\delta \geqslant 3$, U oraz D są kwadratowymi macierzami symetrycznymi wymiaru $p \times p$.

Parametry δ oraz D parametryzują rozkład a priori i zgodnie z koncepcją statystyki Bayesowskiej, nie powinny być optymalizowane, gdyż reprezentują założenia przyjęte przed przystąpieniem do eksperymentów. Pozostałe parametry są szczegółowo opisane w dodatku A.

Warto jednak wspomnieć, że po ustaleniu parametrów p, n, U, δ oraz D funkcja celu jest proporcjonalna do funkcji wiarogodności permutacji przy zaobserwowanych danych. Oznacza to, że znalezienie maksimum tej funkcji jest równoznaczne ze znalezieniem maksimum funkcji wiarogodności, a co za tym idzie, ze znalezieniem estymatora największej wiarogodności tej permutacji. Wybór tej permutacji można utożsamić z **wyborem rozkładu normalnego** modelującego zaobserwowane dane.

Należy zauważyć, że już dla wartości $p = 20$ dziedzina funkcji celu jest niemożliwa do przeszukania ze względu na swoją wielkość. Dlatego algorytmy ewolucyjne wydają się adekwatnym wyborem do rozwiązywania tego problemu.

3 Proponowane rozwiązanie

Na początek rozważony zostanie podstawowy algorytm ewolucyjny, bez krzyżowania, z mutacją będącą jedną losową transpozycją i selekcją turniejową.

Następnie wprowadzona zostanie modyfikacja polegająca na umożliwieniu algorytmowi większej mutacji niż pojedyncza transpozycja. Modyfikacja ta będzie polegać na tym, że na początku działania algorytmu mutacja będzie połączeniem k transpozycji, gdzie k jest dla każdej mutacji losowane jednostajnie z przedziału $\{1, 2, ..., k_{max}\}$, gdzie k_{max} jest parametrem modyfikacji. Następnie w czasie działania algorytmu te k, które częściej dawały poprawy, będą miały większe prawdopodobieństwo bycia wybranym.

Można zauważyć, iż algorytm podstawowy jest szczególnym przypadkiem algorytmu zmodyfikowanego, gdzie $k_{max} = 1$.

W przypadku, jeśli powyższe algorytmy nie uzyskają satysfakcjonujących rezultatów, a czas na to pozwoli, rozważony zostanie algorytm ewolucyjny z **krzyżowaniem**. Potomek będzie obliczany jako połączenie cykli rozłącznych z osobników rodziców. Autorzy dokumentu wierzą, że jest to krzyżowanie adekwatne do natury problemu, gdyż cykle rozłączne rozważanych permutacji stanowią centralną część rozważanej funkcji celu.

4 Hipotezy

W tej sekcji podano hipotezy do niniejszej pracy. Hipotezy te zostaną w czasie pracy sprawdzone i przetestowane sposobami opisanymi w sekcji 5.

1. Opracowany algorytm ewolucyjny będzie znajdował satysfakcjonującą wartość funkcji celu szybciej od algorytmu 'Metropolis_Hastings' z pakietu R *gips* ([3]).

2. Algorytm ewolucyjny z większymi mutacjami będzie dawał lepsze wyniki w porównaniu do podstawowego.

5 Metodologia

Zgodnie z artykułem [1], w pracy również przyjęto stałe wartości hiperparametrów $D = I_p$ oraz $\delta = 3$.

W celu sprawdzenia poprawności oraz jakości proponowanych rozwiązań przeprowadzone zostaną następujące eksperymenty:

- Poprawność algorytmów sprawdzona zostanie na małym wymiarze $p = 6$.

- Zaproponowane algorytmy będą porównane ze sobą oraz z algorytmem 'Metropolis_Hastings' z pakietu R *gips*. Porównanie algorytmów będzie polegało na porównaniu ze sobą parami dystrybuant empirycznych rozwiązań uzyskanych po założonym czasie. Porównanie dokonane będzie testem Wilcoxon'a ([8], [9]).

- Porównanie najlepszego znalezionego algorytmu z oryginalnymi wynikami z sekcji 5.1 oraz 5.2 artykułu [1] przy zachowaniu tych samych wartości hiperparametrów $\Sigma, p, n, U, \delta, D$ oraz liczby wywołań funkcji celu.

Algorytmy te będą przetestowane na kilku funkcjach z rozważanej rodziny funkcji celu. Hiperparametry algorytmu będą dostrojone metodą „random search".

6 Technologia

Projekt zostanie zaimplementowany w języku programowania R ([5]) i będzie używał funkcji celu zdefiniowanej w pakiecie *gips* możliwym do pobrania ze strony [3].

Dodatkowo w projekcie zostaną wykorzystane jeszcze następujące biblioteki R:

- wykresy będą wykonane z użyciem pakietu *ggplot2* ([7]),
- argumenty funkcji celu zapisane będą przy użyciu pakietu *permutations* ([2]).

Implementacja zostanie zrealizowana z wykorzystaniem środowiska RStudio ([6]), a wszystkie użyte skrypty będą udostępnione w repozytorium Github. Skrypty napisane będą w taki sposób, aby wyniki były reprodukowalne, a co za tym idzie ziarna losowości zostaną ustalone i jawnie podane. Ponadto, kod będzie możliwy do wykonania na systemach MacOS, Windows oraz Linux.

Literatura

[1] Piotr Graczyk, Hideyuki Ishi, Bartosz Kołodziejek, and Hélène Massam. Model selection in the space of gaussian models invariant by symmetry. *arXiv*, 04 2020. `https://www.researchgate.net/publication/340500495_Model_selection_in_the_space_of_Gaussian_models_invariant_by_symmetry`.

[2] Robin K.S. Hankin. Introducing the permutations package. *SoftwareX*, 11, 2020. `https://CRAN.R-project.org/package=permutations`.

[3] Chojecki Przemysław. Strona GitHub z kodem pakietu 'gips'. `https://github.com/PrzeChoj/gips/`. [Online; accessed 15-06-2022].

[4] Chojecki Przemysław. Strona GitHub z kodem tego projektu. `https://github.com/PrzeChoj/WAE/`. [Online; accessed 15-06-2022].

[5] R Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2020.

[6] RStudio Team. *RStudio: Integrated Development Environment for R*. RStudio, PBC., Boston, MA, 2020.

[7] Hadley Wickham. *ggplot2: Elegant Graphics for Data Analysis*. Springer-Verlag New York, 2016.

[8] Frank Wilcoxon. Individual comparisons by ranking methods. *Biometrics Bulletin*, 1(6):80–83, 1945.

[9] Frank Wilcoxon. Some rapid approximate statistical procedures. american cyanamid co. *Pearl River, NY, USA*, 1964.

A Dodatek - Opis rodziny funkcji celu

Z punktu widzenia głównej części dokumentacji wstępnej, funkcja celu jest czarną skrzynką. W tym dodatku opisano, skąd wzięła się rodzina rozważanych funkcji celu. Krótko wytłumaczono, czemu znalezienie argumentu z jak największą wartością tej funkcji jest istotne dla zadania uczenia maszynowego.

Skąd się wzięła funkcja celu?

Zakładamy, że wektory losowe są kolumnowe.

Załóżmy, że $Z \sim \mathcal{N}_p(0, \Sigma)$ jest wektorem losowym długości p o rozkładzie normalnym. Załóżmy, że $(Z_1, Z_2, ..., Z_n)$ jest ciągiem n zmiennych losowych iid o takim samym rozkładzie jak Z.

Jeśli $n \geqslant p$, to istnieje estymator największej wiarogodności Σ i wynosi on:

$$U = \frac{1}{n} \sum_{i=1}^{n} Z_i \cdot Z_i^T$$

Jeśli jednak $n < p$, to nie istnieje estymator największej wiarogodności Σ. Innymi słowy, nie da się sensownie estymować macierzy kowariancji.

Można jednak założyć coś więcej o macierzy Σ. Jeśli założymy na przykład, że przenumerowanie wierszy i kolumn $1 \to 2, 2 \to 3, ..., p-1 \to p, p \to 1$ nie zmienia macierzy, to można ją estymować, posiadając tylko jedną obserwację, $n = 1$. Widzimy jednak, że jest to bardzo mocne założenie.

Jeśli przenumerowanie wierszy i kolumn zgodnie z permutacją σ nie zmienia macierzy Σ, to znaczy $\Sigma = \sigma^{-1} \cdot \Sigma \cdot \sigma$, to mówimy, że „$\Sigma$ jest niezmiennicza względem permutacji σ".

W wielu praktycznych zadaniach modelowania możemy założyć niezmienniczość macierzy kowariancji względem jakiejś permutacji. Na przykład, jeśli pomiary dwóch kolumn wykonywane były tym samym urządzeniem. Trudno jest jednak ekspertsko zdecydować o tego typu założeniach.

Autorzy artykułu [1] podali pod numerem (41) wzór na funkcję, która jest proporcjonalna do prawdopodobieństwa a posteriori, że dana obserwacja została wygenerowana przez rozkład niezmienniczy względem permutacji c.

Jeśli dla konkretnej obserwacji $(Z_1, ..., Z_n)$ znalezione zostanie c, dla którego prawdopodobieństwo to będzie największe spośród wszystkich permutacji, to znalezione c będzie estymatorem największej wiarogodności prawdziwej permutacji, z której pochodzi dany rozkład. Sensownie więc byłoby założyć, że obserwacje te pochodzą z tego rozkładu i można dzięki temu założeniu estymować macierz Σ nawet w sytuacjach, gdy $n < p$.