Programowanie Matematyczne
Raport z laboratorium 5

Adam Przemysław Chojecki
298814, Grupa lab B

2023.12.08

1 Zadanie

Celem zadania jest rozwiązanie problemu Ridge na wiele sposobów.

1.1 Opis problemu Ridge

Problemem Ridge jest:

$$n, m \in \mathbb{N}, b \in \mathbb{R}^m, A \in \mathbb{R}^{m\times n}, \alpha \in \mathbb{R}^+$$

$$\min_{x\in\mathbb{R}^n}\{\frac{1}{2}||A\cdot x - b||^2 + \alpha\cdot||x||^2\},$$

Problem tan jest często spotykaną regularyzacją w statystyce obliczeniowej. Zwany jest regularyzacją L_2 i powoduje zbliżenie rozwiązania do wektora zerowego. Dzięki tej regularyzacji rozwiązania nie wybuchają np. gdy kolumny macierzy A są bliskie współliniowości. Regularyzacja typu L_2 jest jedną z najczęściej używanych regularyzacji. Z powodzeniem może być używana w regularyzowaniu sieci neuronowych.

Niewątpliwą zaletą Ridge jest fakt, że można wyznaczyć analityczny wzór na jego rozwiązanie (co stoi w kontrze dla np. regularyzacji typu L_1, czyli LASSO). Dzięki temu szybko można znaleźć rozwiązanie i użyć go np. jako iterację innego algorytmu. Aby znaleźć analityczne rozwiązanie wystarczy policzyć gradient optymalizowanej funkcji względem x:

$$F(x) = \frac{1}{2}||A\cdot x - b||^2 + \alpha\cdot||x||^2 = \frac{1}{2}(Ax-b)^\top\cdot(Ax-b) + \alpha x^\top x =$$

$$= \frac{1}{2}(Ax)^\top(Ax) - b^\top Ax + \frac{1}{2}b^\top b + \alpha x^\top x$$

$$\nabla_x F(x) = A^\top Ax - A^\top b + 2\alpha x$$

Po przyrównaniu gradientu do 0 widzimy jedno jedyne rozwiązanie:

$$\nabla_x F(x) = 0 \iff A^\top b = A^\top Ax + 2\alpha x = (A^\top A + 2\alpha I)x \iff$$

$$\iff (A^\top A + 2\alpha I)^{-1} A^\top b = x$$

Warto przy tym zauważyć, że macierz $(A^\top A + 2\alpha I)$ zawsze jest odwracalna. Jest tak, gdyż macierz $A^\top A$ jest nieujemnie określona. Oznacza to, że jej wartości własne są nieujemne. Być może jakaś jej wartość własna jest równa 0. Gdy dodamy do tej macierzy na przekątnej wartości 2α, to do wszystkich wartości własnych będzie również dodane 2α. A skoro $\alpha > 0$, to macierz $(A^\top A + 2\alpha I)$ ma wszystkie wartości własne dodatnie. A więc jest dodatnio określona.

Inną rzeczą wartą odnotowania jest, że $(A^\top A + 2\alpha I)^{-1} A^\top b$ jest jedynym punktem w którym gradient się zeruje. Czemu jednak musi to być minimum? W ogólności z zerowania się gradientu wynika jedynie, że jest to punkt stacjonarny - być może maksimum, być może minimum, być może punkt siodłowy. W tym przypadku wiemy, że funkcja $F(x)$ musi mieć jakieś minimum gdzieś. Wynika to z tych faktów:

1. $\lim_{||x|| \to \infty} F(x) = \infty$
2. $\forall_x F(x) >= 0$
3. $F(x)$ jest ciągła

Wiemy, że w tym minimum będziemy mieli zerujący się gradient. Ale gradient zeruje sie tylko w jednym punkcie, więc to musi być te właśnie minimum.

1.2 Algorytmy optymalizacyjne

Poza rozwiązaniem analitycznym, wykorzystane zostaną trzy algorytmy optymalizacji numerycznej:

1. Wbudowana w oprogramowanie MATLAB funkcja ‘fminunc()‘ używająca algorytmu quasi-Newtona. Do swojego działania estymuje ona numerycznie wartości pochodnej oraz Hessianu.
2. Algorytm gradientu sprzężonego korzystającego z analitycznych wzorów na gradient oraz Hessian. Długość kroku w danym kierunku obliczana jest z dokładnej metody Newtona.
3. Algorytm gradientu sprzężonego korzystającego z analitycznego wzorów na gradient. Długość kroku w danym kierunku obliczana metodą złotego podziału.

Wyniki funkcji porównane będą z analitycznym rozwiązaniem. Celem jest porównanie metod optymalizacji numerycznej.

1.3 Przewidywane wyniki

Nie ulega wątpliwości, że podejście oparte na rozwiązaniu analitycznym stanowi najbardziej precyzyjną i efektywną metodę. Fascynującym aspektem będzie zestawienie różnych technik optymalizacji numerycznej. Wydaje się, że metoda nr 2 wyróżnia się pod względem efektywności, korzystając z analitycznego obliczania Hessianu. Kolejną obiecującą opcją zdaje się być metoda nr 3, opierająca się przynajmniej na analitycznym obliczaniu gradientu. Ostatnią w kolejności wydaje się być metoda nr 1, która wymaga estymacji wszystkich gradientów i Hessianów.

2 Krótki opis algorytmu

Optymalizacyjny algorytm gradientu sprzężonego można opisać w następujących krokach:

1. **Inicjalizacja:** Rozpoczynamy od wybrania punktu startowego w przestrzeni parametrów.
2. **Obliczenie gradientu:** Obliczamy gradient funkcji celu w danym punkcie. Jeśli mamy możliwość to analitycznie. W przeciwnym razie robimy to w sposób numeryczny.
3. **Kierunek sprzężony:** Określamy kierunek sprzężony do poprzedniego kierunku, który minimalizuje gradient w obecnym punkcie.
4. **Długość kroku:** Wybieramy optymalną długość kroku w wybranym kierunku, aby zminimalizować funkcję celu. Ta część algorytmu może być wykonywana na bardzo wiele sposobów.
5. **Aktualizacja parametrów:** Przesuwamy się w nowym kierunku o wybraną długość kroku, aktualizując parametry.
6. **Sprawdzenie warunku stopu:** Powtarzamy powyższe kroki, sprawdzając jednocześnie warunki stopu. W projekcie użyto maksymalną liczbę iteracji, małą wartość gradientu, małą zmianę wartości funkcji celu.

Algorytm ten ma zaletę skutecznej minimalizacji funkcji celu, zwłaszcza gdy ta funkcja jest trudna do zoptymalizowania za pomocą innych metod. Optymalizuje on parametry w sposób bardziej "inteligentny", dostosowując się do kształtu funkcji celu.

3 Metodologia

W celu porównania wykonano 100 testów na losowych danych. Policzono średni błąd od analitycznego rozwiązana (jako błąd średniokwadratowy między analitycznym x, a policzonym numerycznie) oraz czas potrzebny komputerowi na wykonanie algorytmu. Sprawdzono również liczbę iteracji potrzebną do zbiegnięcia.

W celach reprodukowalności wyników ustawiono źródło dla algorytmu losującego.

4 Wyniki

Wyniki mierzonych metryk przedstawiono w tabelce poniżej.

mierzona metryka	wbudowany ‘fminunc()‘	z Hessianem	ze złotym podziałem
czas (sumaryczna; sekundy)	0.021138	0.000792	0.271789
dokładność (średnia)	1e-10.706512	1e-27.125345	1e-13.409664
liczba interacji (średnia)	NA	1.00000	27.910000

Można z niej odczytać, że zgodnie z przewidywaniami algorytm z Hessianem był najszybszy i najdokładniejszy spośród metod numerycznych. Zaskakującym było dla autora raportu, że zbiegł już w pierwszej iteracji. Algorytm ze złotym podziałem był co prawda (zgodnie z przewidywaniami) o 3 rzędy wielkości dokładniejszy od wbudowanego w MATALB, jednak zajmował zdecydowanie więcej czasu. Można by się zastanowić nad modyfikacją parametru stopu w celu zwiększenia prędkości kosztem dokładności wyniku.

5 Szczegóły techniczne

Kod rozwiązania znajduje się w wielu plikach `.m`. Głównym jest `lab5.m`, gdzie wykonane są testy.

6 Podsumowanie

Udało się osiągnąć założenia projektu. Cel został osiągnięty, implementacja algorytmu działa.

Przewidywania odnośnie rezultatów potwierdziły się.

7 Oświadczenie o samodzielności

Oświadczam, że niniejsza praca stanowiąca podstawę do uznanie osiągnięcia efektów uczenia się z przedmiotu Programowanie Matematyczne została wykonana przeze mnie samodzielnie.

298814, Adam Przemysław Chojecki