

Stochastyczny spadek wzdłuż gradientu

Techniki Optymalizacji

27.11.2018

Dziś zajmiemy się algorytmem stochastycznego spadku wzdłuż gradientu (*Stochastic Gradient Descent* = *SGD*) dla problemu regresji z błędem kwadratowym. Celem będzie więc optymalizacja funkcji celu:

$$L(\mathbf{w}) = \sum_{i=1}^n \underbrace{(y_i - \mathbf{w}^\top \mathbf{x}_i)^2}_{\ell_i(\mathbf{w})},$$

gdzie przez $\ell_i(\mathbf{w})$ oznaczyliśmy błąd kwadratowy na i -tej obserwacji. Gradient $\ell_i(\mathbf{w})$ ma postać:

$$\nabla_{\ell_i}(\mathbf{w}) = -2(y_i - \mathbf{w}^\top \mathbf{x}_i)\mathbf{x}_i,$$

jest więc proporcjonalny do wektora $\mathbf{x}_i$. Algorytm stochastycznego spadku wzdłuż gradientu (SGD) działa zgodnie z następującym schematem:

1. Zaczynaj od wektora początkowego $\mathbf{w}_0 = \mathbf{0}$.
2. W kolejnych iteracjach $i = 1, 2, \dots$:
 - (a) Wybierz kolejną obserwację $(\mathbf{x}_i, y_i)$,
 - (b) Modyfikuj wagi robiąc krok wzdłuż gradientu:

$$\mathbf{w}_i = \mathbf{w}_{i-1} - \alpha_i \nabla_{\ell_i}(\mathbf{w}_{i-1}) = \mathbf{w}_{i-1} + 2\alpha_i(y_i - \mathbf{w}_{i-1}^\top \mathbf{x}_i)\mathbf{x}_i.$$

Waszym celem będzie implementacja tego algorytmu i zastosowanie do problemu wyceny domów, który można pobrać [tutaj](#). Jest to ten sam zbiór, który był używany na pierwszych laboratoriach, tyle że teraz część treningowa i testowa została połączona w jedną.

Należy w kolejności wykonać następujące zadania:

1. Wczytać plik i losowo przetasować wiersze (wykonać losową permutację). Zapewni to, że dane nie będą ułożone w żadnej specjalnej kolejności, która mogłaby potencjalnie pogorszyć zbieżność metody SGD. Można to zrobić za pomocą komend:

```
rp = randperm(n)
y = y(rp)
X = X(rp, :)
```

gdzie n to liczba obserwacji, a y i X to wektor zmiennych wyjściowych i macierz danych wczytane funkcją `readdata`.

2. Kluczowe dla zbieżności algorytmu jest zapewnienie, że wszystkie cechy są w tej samej skali. W innym przypadku gradient zostanie zdominowany przez cechę z największymi wartościami. Dlatego należy wszystkie cechy przeskalować. Można to zrobić na dwa sposoby: (1) *normalizacja* – skalujemy wszystkie wartości na danej osi do przedziału $[0, 1]$; (2) *standaryzacja* – wyznaczamy dla każdej cechy wartość średnią i odchylenie standardowe, a następnie od każdej wartości odejmujemy średnią i dzielimy przez odchylenie standardowe.

Standaryzacja jest preferowana. Można wyznaczyć średnią i odchylenie standardowe za pomocą funkcji Octave'a `mean` oraz `std` (uwaga: te funkcje jako argument przyjmują całą macierz X i policzą średnie i odchylenia dla każdej kolumny!). Przykładowo:

```
Xst = (X - repmat(mean(X), rows(X), 1)) ./ repmat(std(X), rows(X), 1);
```

Podobnie trzeba postąpić z wektorem y :

```
Yst = (Y - repmat(mean(Y), rows(Y), 1)) ./ repmat(std(Y), rows(Y), 1);
```

Uwaga: jeśli została do macierzy X dodana kolumna jedynek, nie należy jej standaryzować, gdyż ta operacja da dzielenie przez zero!

3. Aby zbadać zbieżność algorytmu, należy wyznaczyć rozwiązanie optymalne za pomocą metody najmniejszych kwadratów. Dzięki temu będzie wiadomo, jak daleko od optimum stochastyczny gradient znajduje się w danej chwili. Można wyznaczyć średni błąd kwadratowy takiego wektora wag używając:

```
mse = mean((y - X*w).^2)
```

(powinien dać wartość ok. 0.225 na ustandaryzowanych danych)

4. Uruchomić algorytm SGD dla 50 epok (czyli pełnych przejść po zbiorze danych), pod koniec każdej epoki wyznaczać średni błąd kwadratowy aktualnego rozwiązania i sprawdzać, czy ten błąd zbiega do błędu rozwiązania optymalnego (wyznaczonego w poprzednim punkcie).

Uwaga: wektor x_i można odczytać jako $X(i,:)$ (prym daje wektor kolumnowy).

Uwaga: bardzo istotna jest długość kroku. Proponuję wyznaczyć ją jako $c/\sqrt{i}$, gdzie c jest stałą dobraną tak, aby zbieżność była jak najszybsza (można wybrać $c = 0.1$ na początek), a i jest numerem iteracji (ważne: i rośnie w każdej iteracji o 1, a nie w każdej epoce!).

Dla chętnych:

- Algorytm stochastycznego spadku wzdłuż gradientu może się w pewnych specyficznych przypadkach rozbiegać (wektor wag zaczyna niekontrolowanie wzrastać). Potrzebna jest wtedy tzw. *renormalizacja* wag. Najprościej jest przyjąć pewną maksymalną normę wag W (np. $W = 10$ powinno wystarczyć dla znormalizowanego pliku) i sprawdzać w każdym kroku, czy norma aktualnego wektora wag nie przekroczyła W . Jeśli to nastąpi, tzn. jeśli $\|w_i\| > W$, to dzielimy każdą współrzędną wektora przez jego normę i mnożymy przez W :

$$w_i := \frac{w_i}{\|w_i\|} \cdot W.$$

Uwaga: wykonujemy to tylko wtedy, gdy $\|w_i\| > W$. Zwykle renormalizacja wag nie jest potrzebna (o ile dobierzemy właściwy krok α_i), ale należy o tym pamiętać.