Techniki Optymalizacji: Stochastyczny spadek wzdłuż gradientu I

Wojciech Kotłowski

Instytut Informatyki Politechniki Poznańskiej
email: imię.nazwisko@cs.put.poznan.pl

pok. 2 (CW) tel. (61)665-2936 konsultacje: wtorek 15:00-16:30
Slajdy dostępne pod adresem: `http://www.cs.put.poznan.pl/wkotlowski/`

02.12.2012

Spis treści

Spis treści

Metoda spadku wzdłuż gradientu (Cauchy'ego)

Minimalizacja funkcji $L(\boldsymbol{w})$:

1. Zaczynamy od wybranego rozwiązania startowego, np. $\boldsymbol{w}_0 = \mathbf{0}$.
2. Dla $k = 1, 2, \ldots$ aż do zbieżności
   - Wyznaczamy gradient w punkcie $\boldsymbol{w}_{k-1}$, $\nabla_L(\boldsymbol{w}_{k-1})$.
   - Robimy krok wzdłuż negatywnego gradientu:

   $$\boldsymbol{w}_k = \boldsymbol{w}_{k-1} - \alpha_k \nabla_L(\boldsymbol{w}_{k-1}),$$

   gdzie α_k jest długością kroku ustaloną np. przez przeszukiwanie liniowe.

Specyficzna postać funkcji celu

W większości problemów uczenia maszynowego funkcja celu ma następującą postać:

$$L(\boldsymbol{w}) = \sum_{i=1}^{n} \ell_i(\boldsymbol{w})$$

Specyficzna postać funkcji celu

W większości problemów uczenia maszynowego funkcja celu ma następującą postać:

$$L(\boldsymbol{w}) = \sum_{i=1}^{n} \ell_i(\boldsymbol{w})$$

- L jest sumarycznym błędem na zbiorze uczącym.

Specyficzna postać funkcji celu

W większości problemów uczenia maszynowego funkcja celu ma następującą postać:

$$L(\boldsymbol{w}) = \sum_{i=1}^{n} \ell_i(\boldsymbol{w})$$

- L jest sumarycznym błędem na zbiorze uczącym.
- ℓ_i to błędy na poszczególnych obserwacjach.

Specyficzna postać funkcji celu – przykłady

Specyficzna postać funkcji celu – przykłady

- Regresja liniowa – metoda najmniejszych kwadratów

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

Specyficzna postać funkcji celu – przykłady

- Regresja liniowa – metoda najmniejszych kwadratów

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

- Regresja liniowa – min. wartości bezwzględnych błędów

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

Specyficzna postać funkcji celu – przykłady

- Regresja liniowa – metoda najmniejszych kwadratów

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

- Regresja liniowa – min. wartości bezwzględnych błędów

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

- Klasyfikacja liniowa – regresja logistyczna:

$$L(\boldsymbol{w}) = \sum_{i=1}^{n} \underbrace{\log\Big(1 + \exp(-y_i \boldsymbol{w}^\top \boldsymbol{x}_i)\Big)}_{\ell_i(\boldsymbol{w})}$$

Specyficzna postać funkcji celu – przykłady

- Regresja liniowa – metoda najmniejszych kwadratów

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

- Regresja liniowa – min. wartości bezwzględnych błędów

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

- Klasyfikacja liniowa – regresja logistyczna:

$$L(\boldsymbol{w}) = \sum_{i=1}^{n} \underbrace{\log\left(1 + \exp(-y_i \boldsymbol{w}^\top \boldsymbol{x}_i)\right)}_{\ell_i(\boldsymbol{w})}$$

- Klasyfikacja liniowa – funkcja zawiasowa:

$$L(\boldsymbol{w}) = \sum_{i=1}^{n} \underbrace{\left(1 - y_i \boldsymbol{w}^\top \boldsymbol{x}_i\right)_+}_{\ell_i(\boldsymbol{w})}$$

Metoda stochastycznego spadku wzdłuż gradientu

Idea

Zamiast obliczać gradient na całej funkcji L, w danym kroku oblicz gradient tylko na pojedynczym elemencie ℓ_i.

Metoda stochastycznego spadku wzdłuż gradientu

Idea

Zamiast obliczać gradient na całej funkcji L, w danym kroku oblicz gradient tylko na pojedynczym elemencie ℓ_i.

- Skąd nazwa „stochastyczny"? Ponieważ oryginalnie wybiera się element ℓ_i losowo...

Metoda stochastycznego spadku wzdłuż gradientu

Idea

Zamiast obliczać gradient na całej funkcji L, w danym kroku oblicz gradient tylko na pojedynczym elemencie ℓ_i.

- Skąd nazwa „stochastyczny"? Ponieważ oryginalnie wybiera się element ℓ_i losowo...
- ...ale w praktyce zwykle przechodzi się po całym zbiorze danych w losowej kolejności.

Metoda stochastycznego spadku wzdłuż gradientu

Minimalizacja funkcji $L(\boldsymbol{w})$:

Metoda stochastycznego spadku wzdłuż gradientu

Minimalizacja funkcji $L(\boldsymbol{w})$:

1. Zaczynamy od wybranego rozwiązania startowego, np. $\boldsymbol{w}_0 = \mathbf{0}$.

Metoda stochastycznego spadku wzdłuż gradientu

Minimalizacja funkcji $L(\boldsymbol{w})$:

1. Zaczynamy od wybranego rozwiązania startowego, np. $\boldsymbol{w}_0 = \mathbf{0}$.
2. Dla $k = 1, 2, \ldots$ aż do zbieżności

Metoda stochastycznego spadku wzdłuż gradientu

Minimalizacja funkcji $L(\boldsymbol{w})$:

1. Zaczynamy od wybranego rozwiązania startowego, np. $\boldsymbol{w}_0 = \mathbf{0}$.
2. Dla $k = 1, 2, \ldots$ aż do zbieżności
   - Wylosuj $i \in \{1, \ldots, n\}$.

Metoda stochastycznego spadku wzdłuż gradientu

Minimalizacja funkcji $L(\boldsymbol{w})$:

1. Zaczynamy od wybranego rozwiązania startowego, np. $\boldsymbol{w}_0 = \mathbf{0}$.
2. Dla $k = 1, 2, \ldots$ aż do zbieżności
   - Wylosuj $i \in \{1, \ldots, n\}$.
   - Wyznaczamy gradient funkcji ℓ_i w punkcie $\boldsymbol{w}_{k-1}$, $\nabla_{\ell_i}(\boldsymbol{w}_{k-1})$.

Metoda stochastycznego spadku wzdłuż gradientu

Minimalizacja funkcji $L(\boldsymbol{w})$:

1. Zaczynamy od wybranego rozwiązania startowego, np. $\boldsymbol{w}_0 = \mathbf{0}$.
2. Dla $k = 1, 2, \ldots$ aż do zbieżności
   - Wylosuj $i \in \{1, \ldots, n\}$.
   - Wyznaczamy gradient funkcji ℓ_i w punkcie $\boldsymbol{w}_{k-1}$, $\nabla_{\ell_i}(\boldsymbol{w}_{k-1})$.
   - Robimy krok wzdłuż negatywnego gradientu:

   $$\boldsymbol{w}_k = \boldsymbol{w}_{k-1} - \alpha_k \nabla_{\ell_i}(\boldsymbol{w}_{k-1}),$$

   gdzie α_k jest długością kroku.

Zalety metody stochastycznego spadku wzdłuż gradientu

Zalety metody stochastycznego spadku wzdłuż gradientu

- Szybkość: obliczenie gradientu wymaga wzięcia tylko jednej obserwacji.

Zalety metody stochastycznego spadku wzdłuż gradientu

- Szybkość: obliczenie gradientu wymaga wzięcia tylko jednej obserwacji.
- Skalowalność: cały zbiór danych nie musi nawet znajdować się w pamięci operacyjnej.

Zalety metody stochastycznego spadku wzdłuż gradientu

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

Zalety metody stochastycznego spadku wzdłuż gradientu

- Szybkość: obliczenie gradientu wymaga wzięcia tylko jednej obserwacji.
- Skalowalność: cały zbiór danych nie musi nawet znajdować się w pamięci operacyjnej.
- Prostota: gradient funkcji ℓ_i daje bardzo prosty wzór na modyfikację wag.
- Stochastyczny gradient jest obecnie najbardziej popularną metodą stosowaną w uczeniu maszynowym! (*stochastic gradient descent* – SGD).

Wady metody stochastycznego spadku wzdłuż gradientu

Wady metody stochastycznego spadku wzdłuż gradientu

- Wolna zbieżność: czasem gradient stochastyczny zbiega wolno i wymaga wielu iteracji po zbiorze uczącym.

Wady metody stochastycznego spadku wzdłuż gradientu

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

Wady metody stochastycznego spadku wzdłuż gradientu

- Wolna zbieżność: czasem gradient stochastyczny zbiega wolno i wymaga wielu iteracji po zbiorze uczącym.
- Problem z ustaleniem długości kroku α_k: wyznaczenie α_k przez przeszukiwanie liniowe nie przynosi dobrych rezultatów, ponieważ optymalizujemy oryginalnej funkcji L tylko jej jeden składnik ℓ_i.
- Zalety znacznie przewyższają wady!

Stochastyczny gradient w praktyce

Stochastyczny gradient w praktyce

- Zwykle nie losuje się obserwacji, ale przechodzi się po zbiorze danych w losowej kolejności.

Stochastyczny gradient w praktyce

- Zwykle nie losuje się obserwacji, ale przechodzi się po zbiorze danych w losowej kolejności.
- Zbieżność wymaga często przejścia parokrotnie po całym zbiorze danych (jednokrotne przejście nazywa się epoką).

Stochastyczny gradient w praktyce

- Zwykle nie losuje się obserwacji, ale przechodzi się po zbiorze danych w losowej kolejności.
- Zbieżność wymaga często przejścia parokrotnie po całym zbiorze danych (jednokrotne przejście nazywa się epoką).
- Metody ustalania współczynników długości kroku α_k:

Stochastyczny gradient w praktyce

- Zwykle nie losuje się obserwacji, ale przechodzi się po zbiorze danych w losowej kolejności.
- Zbieżność wymaga często przejścia parokrotnie po całym zbiorze danych (jednokrotne przejście nazywa się epoką).
- Metody ustalania współczynników długości kroku α_k:
  - Ustalamy stałą wartość $\alpha_k = \alpha$

Stochastyczny gradient w praktyce

- Zwykle nie losuje się obserwacji, ale przechodzi się po zbiorze danych w losowej kolejności.
- Zbieżność wymaga często przejścia parokrotnie po całym zbiorze danych (jednokrotne przejście nazywa się epoką).
- Metody ustalania współczynników długości kroku α_k:
  - Ustalamy stałą wartość $\alpha_k = \alpha$
    $\Longrightarrow$ Zwykle tak się robi w praktyce, działa dobrze ale wymaga ustalenia α metodą prób i błędów

Stochastyczny gradient w praktyce

- Zwykle nie losuje się obserwacji, ale przechodzi się po zbiorze danych w losowej kolejności.
- Zbieżność wymaga często przejścia parokrotnie po całym zbiorze danych (jednokrotne przejście nazywa się epoką).
- Metody ustalania współczynników długości kroku α_k:
  - Ustalamy stałą wartość $\alpha_k = \alpha$
    $\Longrightarrow$ Zwykle tak się robi w praktyce, działa dobrze ale wymaga ustalenia α metodą prób i błędów
  - Bierzemy wartość kroku malejącą jak $\sim \frac{1}{\sqrt{k}}$: $\alpha_k = \alpha/\sqrt{k}$

Stochastyczny gradient w praktyce

- Zwykle nie losuje się obserwacji, ale przechodzi się po zbiorze danych w losowej kolejności.
- Zbieżność wymaga często przejścia parokrotnie po całym zbiorze danych (jednokrotne przejście nazywa się epoką).
- Metody ustalania współczynników długości kroku α_k:
  - Ustalamy stałą wartość $\alpha_k = \alpha$
    $\Longrightarrow$ Zwykle tak się robi w praktyce, działa dobrze ale wymaga ustalenia α metodą prób i błędów
  - Bierzemy wartość kroku malejącą jak $\sim \frac{1}{\sqrt{k}}$: $\alpha_k = \alpha/\sqrt{k}$
    $\Longrightarrow$ Zapewniona zbieżność, ale czasem może zbiegać zbyt wolno.

Spis treści

Regresja liniowa – metoda najmniejszych kwadratów (LS)

Regresja liniowa – metoda najmniejszych kwadratów (LS)

- Funkcja celu:

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

Regresja liniowa – metoda najmniejszych kwadratów (LS)

- Funkcja celu:

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

- Pochodne funkcji $\ell_i(\boldsymbol{w})$:

$$\frac{\partial \ell_i(\boldsymbol{w})}{\partial w_j} = -2(y_i - \boldsymbol{w}^\top \boldsymbol{x}_i) x_{ij}$$

Regresja liniowa – metoda najmniejszych kwadratów (LS)

- Funkcja celu:

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

- Pochodne funkcji $\ell_i(\boldsymbol{w})$:

$$\frac{\partial \ell_i(\boldsymbol{w})}{\partial w_j} = -2(y_i - \boldsymbol{w}^\top \boldsymbol{x}_i) x_{ij}$$

- Gradient funkcji $\ell_i(\boldsymbol{w})$:

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

SGD dla regresji liniowej LS

1. Zaczynamy od $\boldsymbol{w}_0 = \mathbf{0}$.
2. Dla $k = 1, 2, \ldots$ aż do zbieżności
   - Wylosuj $i \in \{1, \ldots, n\}$.
   - Wyznaczamy gradient ℓ_i w punkcie $\boldsymbol{w}_{k-1}$, $-2(y_i - \boldsymbol{w}^\top \boldsymbol{x}_i)\boldsymbol{x}_i$.
   - Robimy krok wzdłuż negatywnego gradientu:

$$\boldsymbol{w}_k = \boldsymbol{w}_{k-1} + 2\alpha_k (y_i - \boldsymbol{w}^\top \boldsymbol{x}_i)\boldsymbol{x}_i$$

SGD dla regresji liniowej LS

1. Zaczynamy od $\boldsymbol{w}_0 = \mathbf{0}$.
2. Dla $k = 1, 2, \ldots$ aż do zbieżności
   - Wylosuj $i \in \{1, \ldots, n\}$.
   - Wyznaczamy gradient ℓ_i w punkcie $\boldsymbol{w}_{k-1}$, $-2(y_i - \boldsymbol{w}^\top \boldsymbol{x}_i)\boldsymbol{x}_i$.
   - Robimy krok wzdłuż negatywnego gradientu:

$$\boldsymbol{w}_k = \boldsymbol{w}_{k-1} + 2\alpha_k(y_i - \boldsymbol{w}^\top \boldsymbol{x}_i)\boldsymbol{x}_i$$

- Krok w kierunku wektora $\boldsymbol{x}_i$, z długością kroku równą $2\alpha_k(y_i - \boldsymbol{w}^\top \boldsymbol{x}_i)$.

SGD dla regresji liniowej LS

1. Zaczynamy od $\boldsymbol{w}_0 = \mathbf{0}$.
2. Dla $k = 1, 2, \ldots$ aż do zbieżności
   - Wylosuj $i \in \{1, \ldots, n\}$.
   - Wyznaczamy gradient ℓ_i w punkcie $\boldsymbol{w}_{k-1}$, $-2(y_i - \boldsymbol{w}^\top \boldsymbol{x}_i)\boldsymbol{x}_i$.
   - Robimy krok wzdłuż negatywnego gradientu:

$$\boldsymbol{w}_k = \boldsymbol{w}_{k-1} + 2\alpha_k (y_i - \boldsymbol{w}^\top \boldsymbol{x}_i)\boldsymbol{x}_i$$

- Krok w kierunku wektora $\boldsymbol{x}_i$, z długością kroku równą $2\alpha_k (y_i - \boldsymbol{w}^\top \boldsymbol{x}_i)$.
- Długość kroku proporcjonalna do „przeszacowania predykcji”.

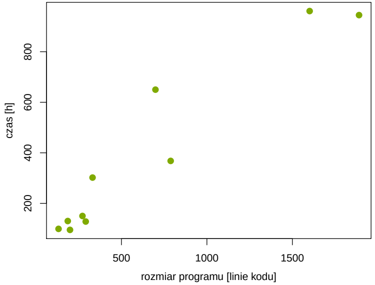
Spis treści

Przykład: szacowanie czasu pracy programistów

X	Y
Rozmiar programu	Oszacowany czas
186	130
699	650
132	99
272	150
291	128
331	302
199	95
1890	945
788	368
1601	961

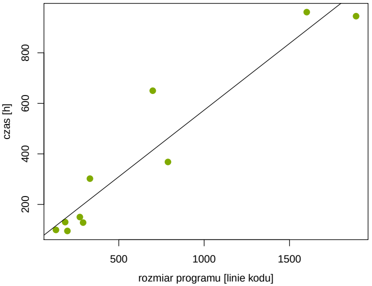
Przykład: szacowanie czasu pracy programistów

X	Y
Rozmiar programu	Oszacowany czas
186	130
699	650
132	99
272	150
291	128
331	302
199	95
1890	945
788	368
1601	961



Przykład: szacowanie czasu pracy programistów

X	Y
Rozmiar programu	Oszacowany czas
186	130
699	650
132	99
272	150
291	128
331	302
199	95
1890	945
788	368
1601	961



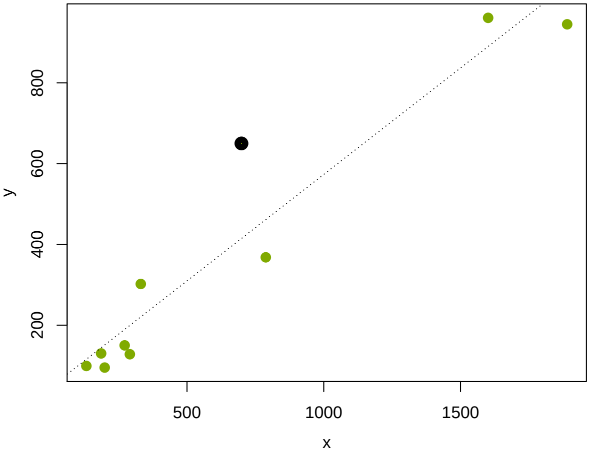
$$a_0 = 45.93, a_1 = 0.5273$$

Szacowanie Czasu programistów

- Zaczynamy z rozwiązaniem $\boldsymbol{w} = (w_0, w_1) = \mathbf{0}$.
- Krok: $\alpha_k = 0.5/\sqrt{k}$.
- Jednostki zostały zamienione na 1000min i 1000 lini kodu, aby uniknąć dużych liczb.
  (ujednolicenie skali jest istotne dla zbieżności metody!)

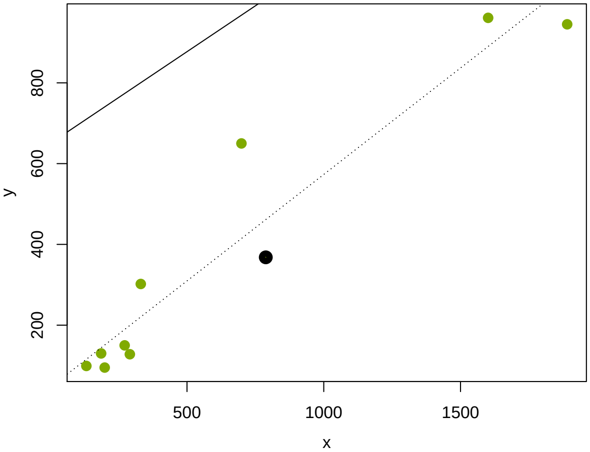
Szacowanie Czasu programistów

iteracja 1



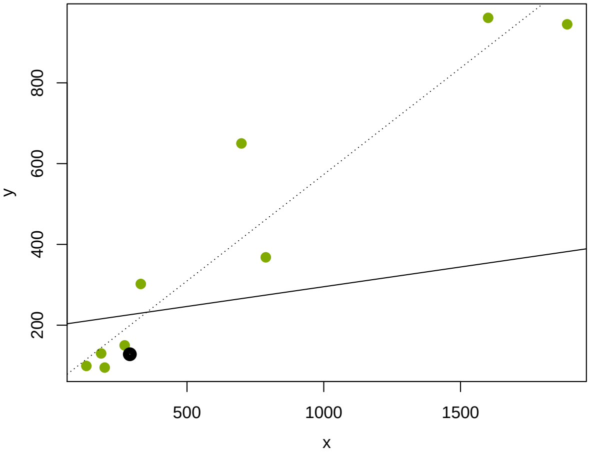
Szacowanie Czasu programistów

iteracja 2



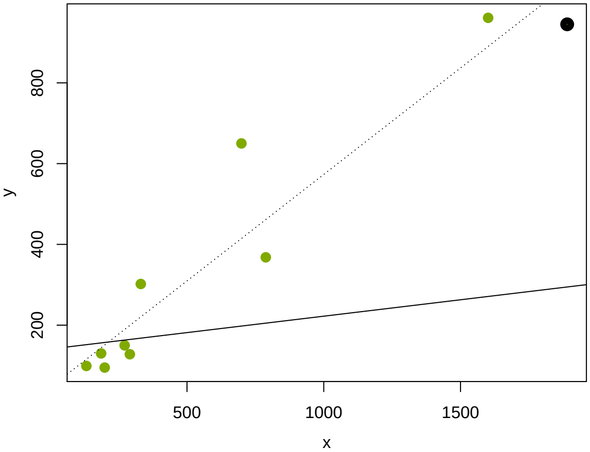
Szacowanie Czasu programistów

iteracja 3



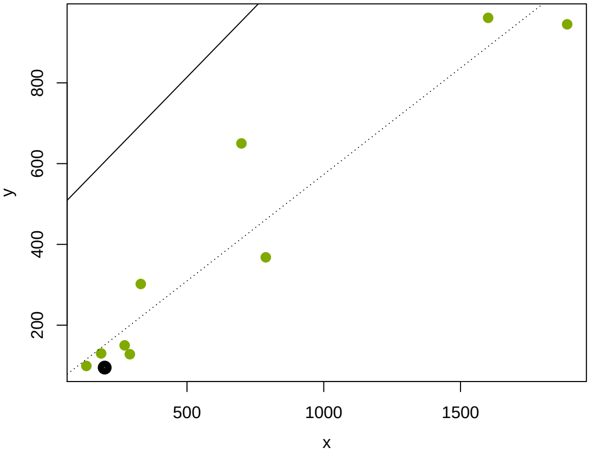
Szacowanie Czasu programistów

iteracja 4



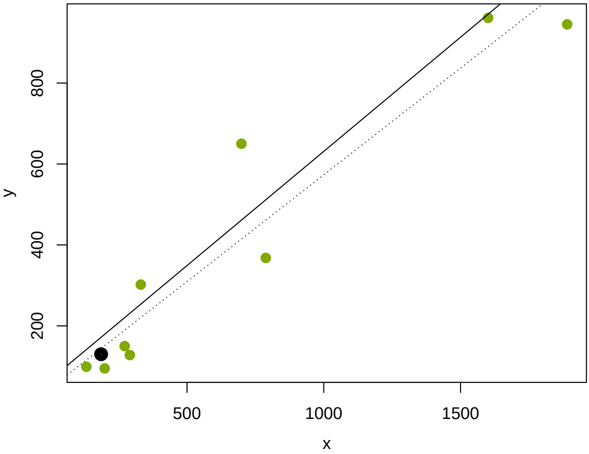
Szacowanie Czasu programistów

iteracja 5



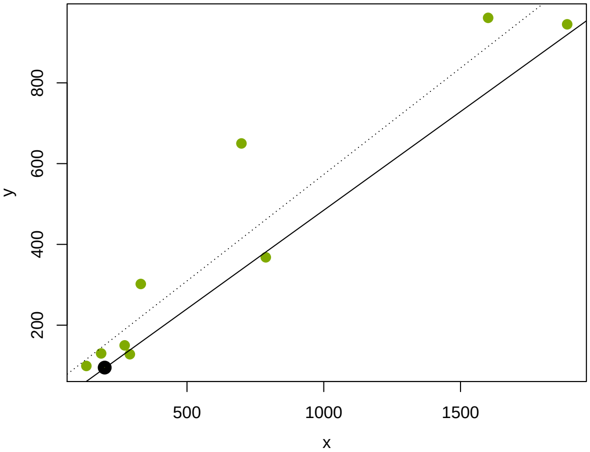
Szacowanie Czasu programistów

iteracja 10

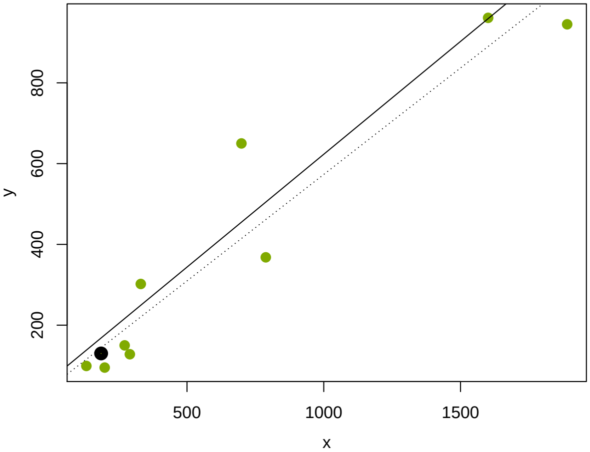


Szacowanie Czasu programistów

iteracja 15

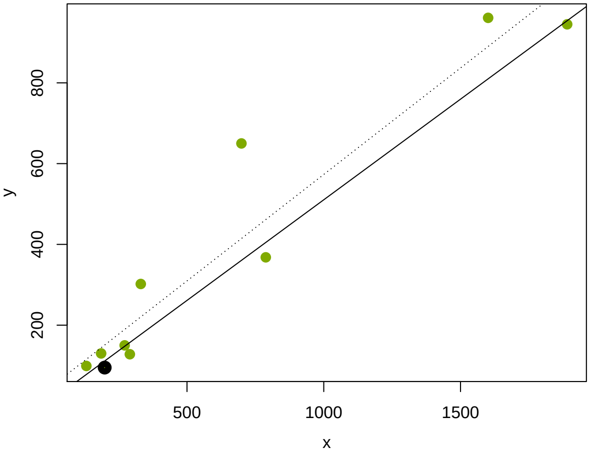


iteracja 20

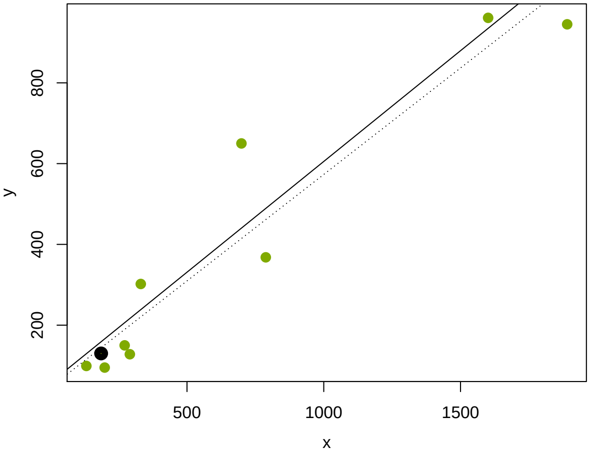


Szacowanie Czasu programistów

iteracja 25

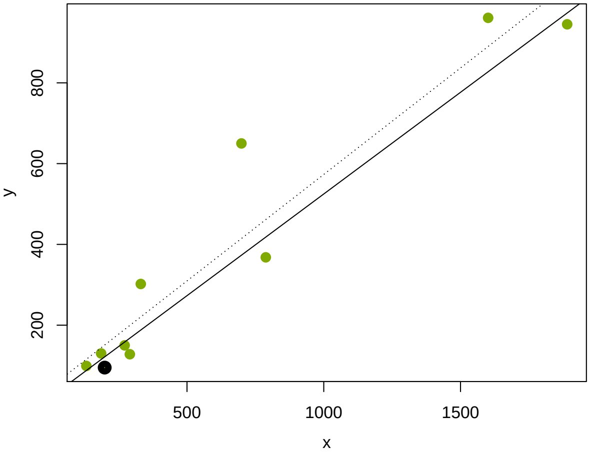


iteracja 30



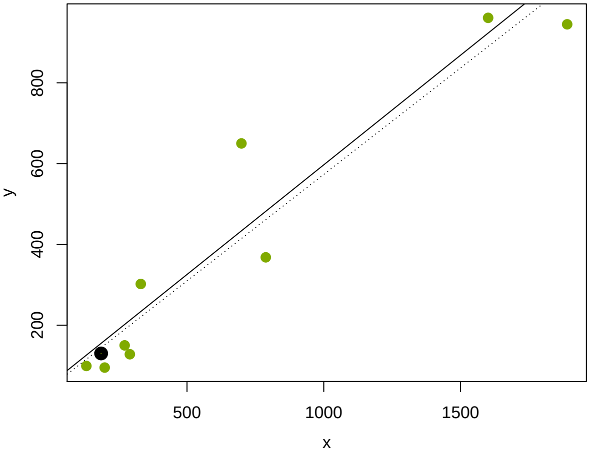
Szacowanie Czasu programistów

iteracja 35



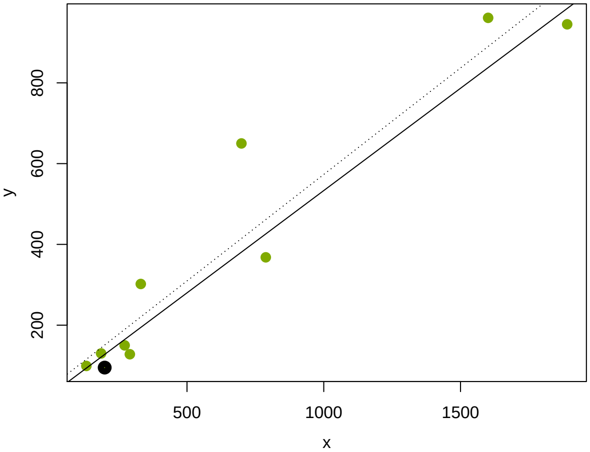
Szacowanie Czasu programistów

iteracja 40



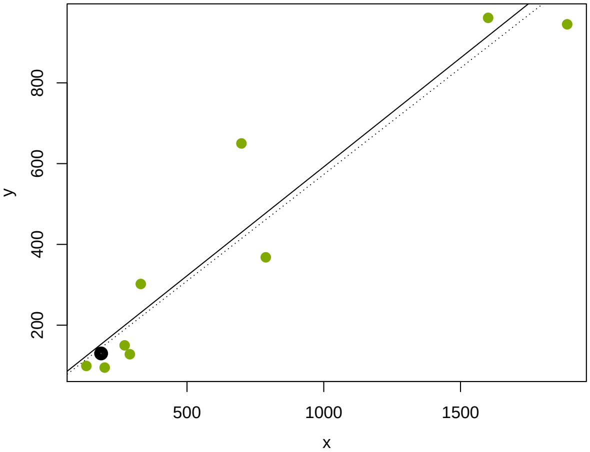
Szacowanie Czasu programistów

iteracja 45



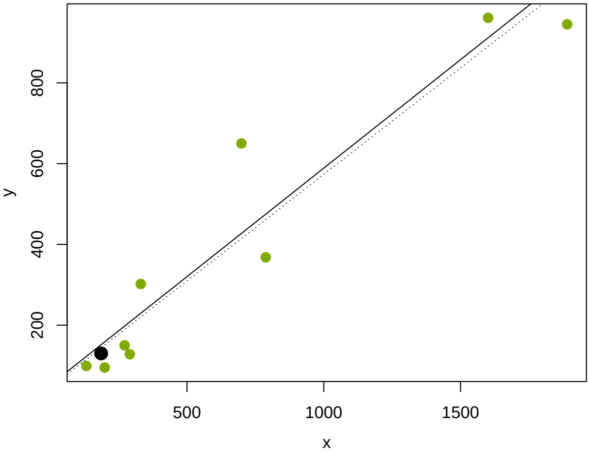
Szacowanie Czasu programistów

iteracja 50



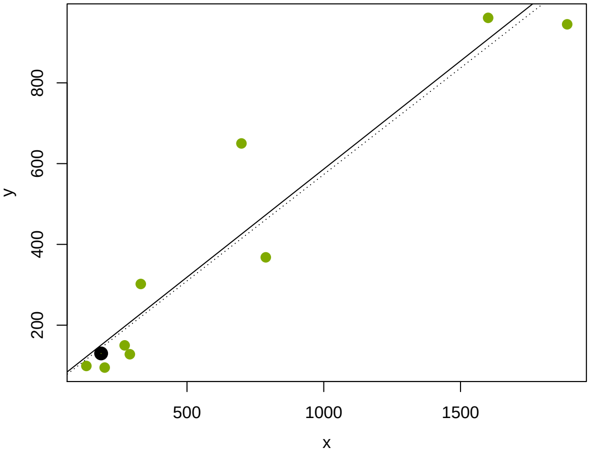
Szacowanie Czasu programistów

iteracja 60



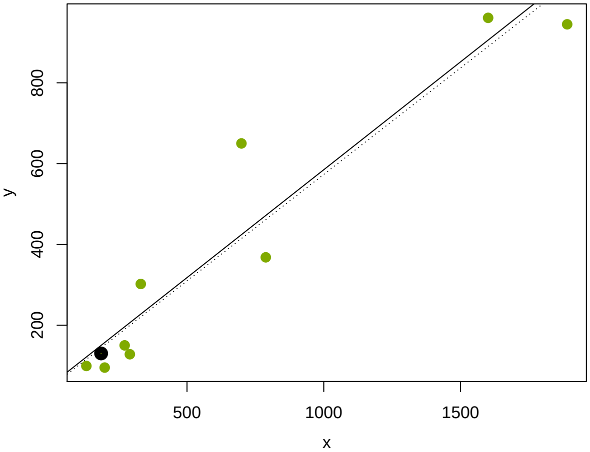
Szacowanie Czasu programistów

iteracja 70



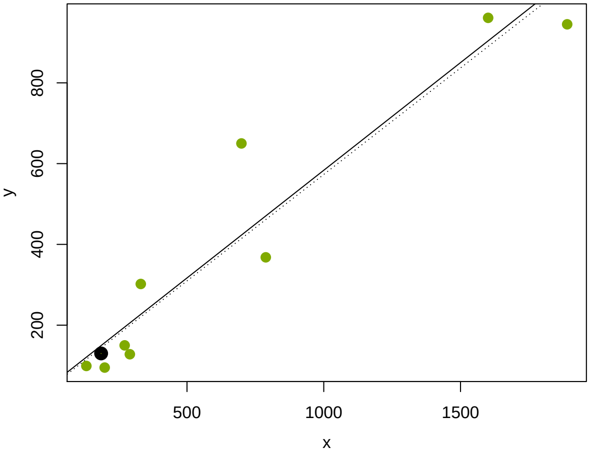
Szacowanie Czasu programistów

iteracja 80

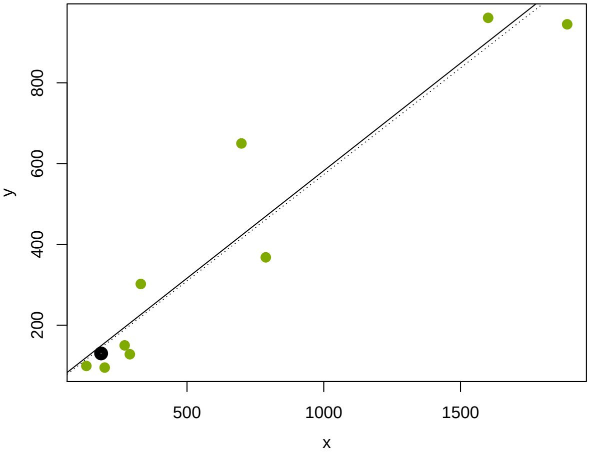


Szacowanie Czasu programistów

iteracja 90



iteracja 100



Koniec na dzisiaj :)