Zrównoleglona optymalizacja stochastyczna na dużych zbiorach danych

mgr inż. C. Dendek prof. nzw. dr hab. J. Mańdziuk

Politechnika Warszawska,
Wydział Matematyki i Nauk Informacyjnych

Outline

Outline

Typowy przypadek zastosowania optymalizacji w uczeniu maszynowym

Cel

Minimalizacja funkcji F zależnej od elementów zbioru uczącego $(x_i, y_i)_{i=1}^n$ na zbiorze parametrów W.

Przykłady

- $F(w) = \sum_{i=1}^n (y_i - H(x_i, w))^2$
- $F(w) = \lambda\Omega(w) + \sum_{i=1}^n (y_i - H(x_i, w))^2$
- $F(w) = \sum_{i=1}^n \log(1 + e^{y_i H(x_i, w)})$

Typowy przypadek zastosowania optymalizacji w uczeniu maszynowym

Cel

Minimalizacja funkcji F zależnej od elementów zbioru uczącego $(x_i, y_i)_{i=1}^{n}$ na zbiorze parametrów W.

Strategie

- metody pierwszego i drugiego rzędu (skalowalność względem liczby zmiennych)
- metody quasi–newtonowskie
- on–line i batch learning

Strategie

Metody pierwszego rzędu

- przybliżenie funkcji F funkcją liniową (rozwinięcie w szereg Taylora)
- typowe wymagania pamięciowe: $o(d)$
- brak modelowania zależności pomiędzy zmiennymi

Strategie

Metody drugiego rzędu

- przybliżenie funkcji F funkcją “kwadratową“ (rozwinięcie w szereg Taylora)
- typowe wymagania pamięciowe: $o(d^2)$
- modelowanie zależności (kowariancji) pomiędzy zmiennymi
- dodatkowe wymagania na funkcję F

Strategie

Batch learning

- dostosowanie wag po przetworzeniu pełnego zbioru
- możliwość pełnego zrównoleglenia obliczeń dla ustalonego wektora wag
- możliwość ”utknięcia“ w lokalnym minimum

On–line learning

- dostosowanie wag po każdym elemencie zbioru uczącego
- problemy ze zrównolegleniem obliczeń
- zmniejszona możliwość ”utknięcia“ w lokalnym minimum

Outline

Gradient Descent

Cel

Minimalizacja funkcji F (wypukłej, różniczkowalnej i lipschitzowsko ciągłej) na zbiorze W.

Algorytm

- wybierz $w_0 \in W$ jako punkt startowy
- Iteruj:

$$w^{(k+1)} := \Pi_W(w^{(k)} - \alpha^{(k)} \nabla F(w^{(k)})),$$

gdzie $\Pi_W(z)$ stanowi projekcję z w zbiór W:
$\Pi_W(v) = \arg\min_{w \in W}(||w - v||)$

Subgradient Descent

Cel

[Gradient Descent] bez założenia o różniczkowalności

Algorytm

[Gradient Descent], zamiast gradientu należy użyć podgradientu:

$$g^{(k)} \in \nabla F(w^{(k)}) = \{g : \forall_{v \in w} F(v) \geq F(w^{(k)}) \langle v - w^{(k)}, g \rangle\},$$

równego gradientowi w punktach różniczkowalności.

Outline

Stochastic Gradient Descent

Cel

Minimalizacja *wartości oczekiwanej* funkcji F (wypukłej i lipschitzowsko cg ze stałą G) na zbiorze W.

Algorytm

- wybierz $w_0 \in W$ jako punkt startowy
- Iteruj:
  - wybierz *estymator* podgradientu $g^{(k)}$, tak, aby $E[g^{(k)}] \in \nabla F(w^{(k)})$
  - wykonaj krok

$$w^{(k+1)} := \Pi_W(w^{(k)} - \alpha^{(k)} g^{(k)})$$

- $\bar{w}^{(k)} = \frac{1}{k} \sum_{i=1}^{k} w^{(k)}$ LUB losowo wybierz $w^{(i)}$

Stochastic Gradient Descent

Algorytm: porównanie z wersją niestochastyczną

- ...
- Iteruj:
  - wybierz *estymator* podgradientu $g^{(k)}$, tak, aby $E[g^{(k)}] \in \nabla F(w^{(k)})$
    komentarz: gdy F zależne od zbioru uczącego *estymator* na podstawie 1 obserwacji (i.i.d.)
  - ...
- $\bar{w}^{(k)} = \frac{1}{k} \sum_{i=1}^{k} w^{(i)}$
  komentarz: Uśrednianie redukuje wariancję ale wymaga wypukłości względem W

Stochastic Gradient Descent jest metodą aproksymacji stochastycznej (SA).

Stochastic Gradient Descent: oszacowania błędów

Główne równanie

$$E[F(\bar{w}^{(k)})] - F(w^*) \leq \frac{||w^0 - w^*|| + \sum_{i=1}^{k} (\alpha^i)^2 E[||g^{(k)}||^2]}{\sum_{i=1}^{k} \alpha^i}$$

Przypadek ustalonych (niezmiennych) kroków gradientowych

- $\alpha = \frac{||w^*||}{G\sqrt{K}}$ w kroku K: $E[F(\bar{w}^{(K)})] - F(w^*) \leq \frac{G||w^*||}{\sqrt{K}}$
- $\alpha = \frac{\epsilon}{G^2}$ s.t. $E[F(\bar{w}^{(k)})] - F(w^*) \leq \epsilon$ with $k = \frac{G^2||w^*||^2}{\epsilon^2}$

Outline

Założenia

Cel

$$\arg\min_{w \in W} E_z[f(w, z)] = \arg\min_{w \in W} F(w)$$

na podstawie nieobciążonych estymatorów $F(w)$ oraz $\nabla F(w)$.

Optymalizacja na podstawie próbki

$z_1 \ldots z_n$ i.i.d.

$$g^{(k)} = \nabla_w f(w^{(k)}, z^{(k)})$$

Zawężenie: optymalizacja wypukła

- f jest f. wypukłą
- W jest zbiorem wypukłym

Stochastyczna optymalizacja wypukła – przykłady

- L2–regularyzowana regresja logistyczna
  np. wykrywanie spamu, klasyfikacja
- LASSO, LASSO grupowane
  np. klasyfikacja z wyborem zmiennych
- liniowy SVM
- niskorzędowa dekompozycja macierzy np. systemy rekomendacji filmów (Netflix)

Podejścia do problemu optymalizacji stochastycznej

Sample Average Approximation

- minimalizacja $\hat{F}(w) = \frac{1}{m} \sum_{i=1}^{m} f(w, z_i)$
- podstawia: kryterium minimalizacji ryzyka empirycznego (Empirical Risk Minimization)
- wykorzystuje skończoną próbkę

Stochastic Approximation

- parametry zmieniane na podstawie estymatorów
- w każdej iteracji wymaga ”świeżej próbki“
- najprostsza metoda: *jednoprzebiegowy* SGD
- inne metody: quasi–drugiego rzędu (quasi–Newton SGD), Stochastic Mirror Descent

Podejście łączone: SA w kroku SAA

Podejście łączone

- iteracyjna minimalizacja $\hat{F}(w) = \frac{1}{m} \sum_{i=1}^{m} f(w, z_i)$
- w ”środku“ SAA: metoda SA (np. SGD) stosowana do celu empirycznego z użyciem przykładów uczących z_i wypróbkowanych ze zb. uczącego
- używa skończonej próbki, pozwalając jednocześnie na większą ilość iteracji niż mamy obserwacji.

Intuicyjne pytanie...

Skoro wynik SGD jest zdefiniowany jako $\bar{w}^{(k)} = \frac{1}{k} \sum_{i=1}^{k} w^{(i)}$ czy można próbować dzielić go na podproblemy?

Parallel Stochastic Gradient Descent, [Zinkevich 2010]

Założenia

- $c^1 ... c^n$ (wypukła) ”strata“ związana z obserwacjami 1...n.
- stały wsp. kroku gradientu α
- p komputerów

Cel

Minimalizacja ryzyka całkowitego $\hat{c}(w) = \frac{1}{m} \sum_{i=1}^{m} c^i(w)$

Własności c^i

$$c^i(w) = \frac{\lambda}{2}||w||^2 + L(x^i, y^i, \langle w, x^i \rangle),$$

gdzie L jest

- wypukła
- lipschitzowsko cg względem w ($||L_{(x,y)}(x, y, y')||_L \leq G$)
- z lipschitzowsko ograniczonym gradientem ($||\nabla_{y'} L_{(x,y)}(x, y, y')||_L \leq c^*$)

Parallel Stochastic Gradient Descent, [Zinkevich 2010]

Założenia

- $c^1 ... c^n$ (wypukła) ”strata“ związana z obserwacjami 1...n.
- stały wsp. kroku gradientu α
- p komputerów
- minimalizacja komunikacji pomiędzy maszynami

Parallel Stochastic Gradient Descent, [Zinkevich 2010]

Algorytm

- Niech $T = \lfloor \frac{m}{p} \rfloor$
- Losowy podział i mieszanie przykładów uczących, tak aby na każdej maszynie było ich przynajmniej T
- Równolegle na każdej maszynie i:
  - inicjalizacja $w^{i,0} = 0$
  - iteracja przez zbiór $c^{i,1} ... c^{i,T}$ z każdorazową modyfikacją wag $w^{i,k+1} = w^{i,k} - \alpha \nabla_w c^{i,k}(w^{i,k})$
- $\bar{w} = \frac{1}{k} \sum_{i=1}^{p} w^{(i,T)}$

Wybór kroku α

Twierdzenie

Niech c^* t.ż. $||\nabla_{y'} L_{(x,y)}(x, y, y')||_L \leq c^*$.
Jeśli $\alpha \leq (||x^i||^2 c^* + \lambda)^{-1}$ wtedy krok gradientowy dla c^i jest odwzorowaniem zwężającym ze stałą lipschitzowską $1 - \alpha\lambda$.

Jednorodne zwężenie

α powinno generować zwężenie względem wszystkich c^i:

$$\alpha^* = \min_i (||x^i||^2 c^* + \lambda)^{-1}$$

Oszacowanie błędu

Twierdzenie

Niech $\alpha \leq \alpha^*$ i $T = \frac{\ln p - \ln \alpha\lambda}{2\alpha\lambda}$. Wtedy:

$$E_w[c(w)] - \min_w c(w) \leq \frac{8\alpha G^2}{\sqrt{p\lambda}} \sqrt{||\nabla c||_L} + \frac{8\alpha G^2}{p\lambda} ||\nabla c||_L + (2\alpha G^2)$$

Jak praktycznie kontrolować średni błąd?

$$\alpha \mapsto \frac{\alpha}{2} \Rightarrow T \mapsto 2T \wedge E_w[c(w)] - \min_w c(w) \mapsto \frac{E_w[c(w)] - \min_w c(w)}{2}$$

Dziękuję za uwagę

Dziękuję za uwagę