Методы оптимизации в машинном обучении

Метод сопряженных градиентов (семинар 4)

Родоманов А. О.

ВМК МГУ

26 сентября 2016

CG и предобуславливание

- Решаемая задача: $Ax = b$, где $A = A^\top \succ 0$.
- Скорость работы метода определяется спектром матрицы A.
- Спектр можно улучшить с помощью эквивалентного преобразования системы:

$$Ax = b \quad \Leftrightarrow \quad (S^{-\top} A S^{-1})(Sx) = S^{-\top} b,$$

  где $S \in \mathbb{R}^{n \times n}$ — невырожденная матрица.
- Новая система: $\tilde{A}\tilde{x} = \tilde{b}$, где

$$\tilde{A} := S^{-\top} A S^{-1}, \qquad \tilde{b} := S^{-\top} b.$$

  Решение исходной системы: $x = S^{-1}\tilde{x}$.
- Матрица $M := S^\top S$ называется предобуславливателем.
- Если $M \approx A$, то $\tilde{A} \approx I \Rightarrow$ сходимость $\approx$ за одну итерацию.

CG vs PCG

Обычный CG:

$r_0 := Ax_0 - b$;

$d_0 := -r_0$;

$k := 0$;

while $\|r_k\| > \varepsilon$ **do**

$\alpha_k := \frac{r_k^\top r_k}{d_k^\top A d_k}$;

$x_{k+1} := x_k + \alpha_k d_k$;

$r_{k+1} := r_k + \alpha_k A d_k$;

$\beta_k := \frac{r_{k+1}^\top r_{k+1}}{r_k^\top r_k}$;

$d_{k+1} := -r_{k+1} + \beta_k d_k$;

$k := k + 1$;

end

CG vs PCG

Обычный CG:

$r_0 := Ax_0 - b;$
$d_0 := -r_0;$
$k := 0;$
while $\|r_k\| > \varepsilon$ **do**
  $\alpha_k := \frac{r_k^\top r_k}{d_k^\top A d_k};$
  $x_{k+1} := x_k + \alpha_k d_k;$
  $r_{k+1} := r_k + \alpha_k A d_k;$
  $\beta_k := \frac{r_{k+1}^\top r_{k+1}}{r_k^\top r_k};$
  $d_{k+1} := -r_{k+1} + \beta_k d_k;$
  $k := k + 1;$
end

CG с предобуславливателем:

$r_0 := Ax_0 - b;$
$d_0 := -M^{-1} r_0;$
$k := 0;$
while $\|r_k\| > \varepsilon$ **do**
  $\alpha_k := \frac{r_k^\top M^{-1} r_k}{d_k^\top A d_k};$
  $x_{k+1} := x_k + \alpha_k d_k;$
  $r_{k+1} := r_k + \alpha_k A d_k;$
  $\beta_k := \frac{r_{k+1}^\top M^{-1} r_{k+1}}{r_k^\top M^{-1} r_k};$
  $d_{k+1} := -M^{-1} r_{k+1} + \beta_k d_k;$
  $k := k + 1;$
end

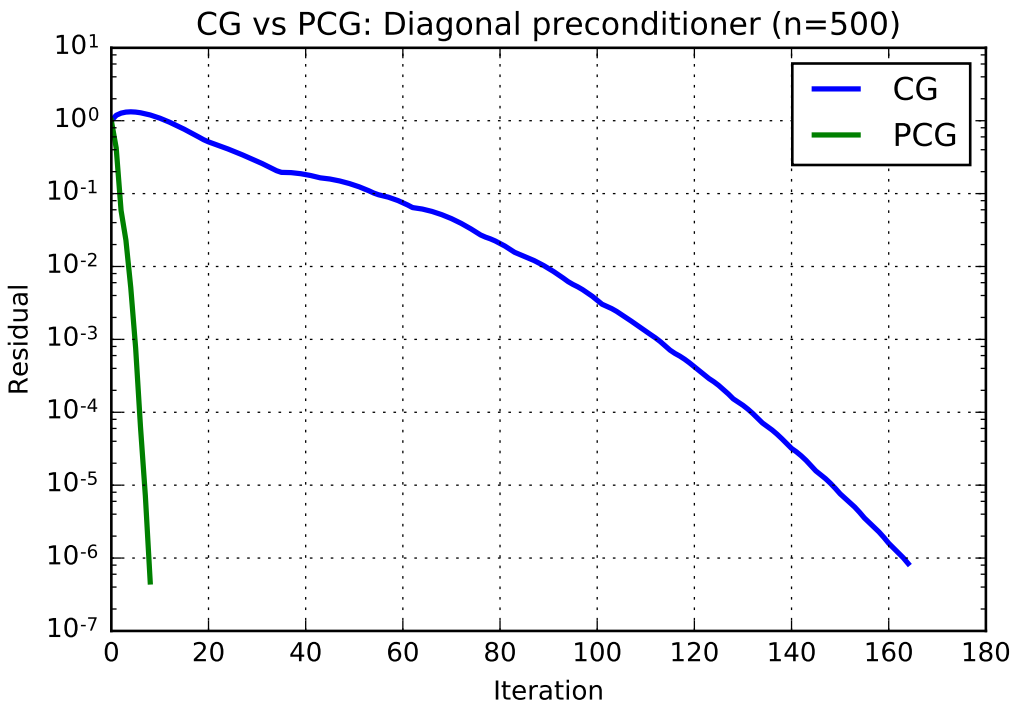
- Дополнительно нужна процедура решения системы $Mz_k = r_k$.

Предобуславливание: пример

- Система $Ax = b$ размера $n = 500$, где $b = (1, \ldots, 1)$ и

$$a_{ij} = \begin{cases} 1 + i^{1.2} & \text{если } i = j \\ 1 & \text{если } |i - j| = 1 \text{ или } |i - j| = 100 \\ 0 & \text{иначе} \end{cases}$$

- Диагональный предобуславливатель: $M = \mathrm{Diag}(A)$.



Нелинейный CG. Метод Флетчера–Ривса

Линейный CG:

$r_0 := Ax_0 - b;$
$d_0 := -r_0;$
$k := 0;$
while $\|r_k\| > \varepsilon$ **do**
  $\alpha_k := \frac{r_k^\top r_k}{d_k^\top A d_k};$
  $x_{k+1} := x_k + \alpha_k d_k;$
  $r_{k+1} := r_k + \alpha_k A d_k;$
  $\beta_k := \frac{r_{k+1}^\top r_{k+1}}{r_k^\top r_k};$
  $d_{k+1} := -r_{k+1} + \beta_k d_k;$
  $k := k + 1;$
end

Флетчер–Ривс:

$g_0 := \nabla f(x_0);$
$d_0 := -g_0;$
$k := 0;$
while $\|g_k\| > \varepsilon$ **do**
  $\alpha_k := \{\text{поиск по } d_k\};$
  $x_{k+1} := x_k + \alpha_k d_k;$
  $g_{k+1} := \nabla f(x_{k+1});$
  $\beta_k := \frac{g_{k+1}^\top g_{k+1}}{g_k^\top g_k};$
  $d_{k+1} := -g_{k+1} + \beta_k d_k;$
  $k := k + 1;$
end

Другие схемы

- Существует много вариантов нелинейного CG.
- Они все совпадают на строго выпуклой квадратичной функции.

Полак–Рибье: $$\beta_k^{\mathrm{PR}} := \frac{g_{k+1}^\top y_k}{\|g_k\|_2^2}$$

Хестинс–Штифель: $$\beta_k^{\mathrm{HS}} := \frac{g_{k+1}^\top y_k}{d_k^\top y_k}$$

Полак–Рибье+: $$\beta_k^{\mathrm{PR}+} := \max\{0, \beta_k^{\mathrm{PR}}\}$$

Гильберт-Ноусидаль: $$\beta_k^{\mathrm{GN}} := \max\{-\beta_k^{\mathrm{FR}}, \min\{\beta_k^{\mathrm{PR}}, \beta_k^{\mathrm{FR}}\}\}$$

Дай–Юань: $$\beta_k^{\mathrm{DY}} := \frac{\|g_{k+1}\|_2^2}{d_k^\top y_k}$$

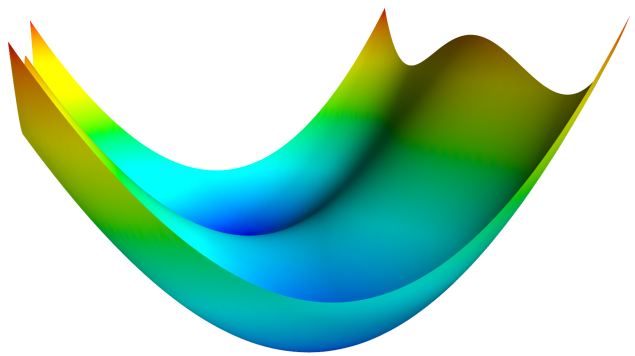
Агер–Джан: $$\beta_k^{\mathrm{HZ}} := \frac{g_{k+1}^\top}{d_k^\top y_k}\left(y_k - \frac{2\|y_k\|_2^2}{d_k^\top y_k} d_k\right)$$

Во всех формулах $y_k := g_{k+1} - g_k$.

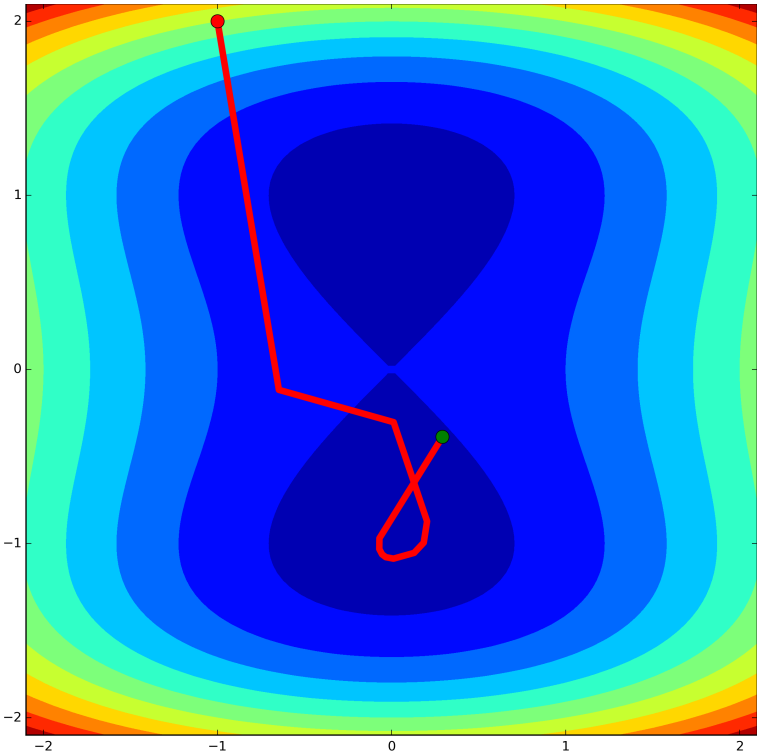
Флетчер–Ривс и направление спуска

- В общем случае метод Флетчера–Ривса не гарантирует того, что направление d_k является направлением спуска.
- Однако можно доказать, что если на этапе поиска по прямой использовать сильные условия Вульфа с константой $c_2 \in (0, 0.5)$, то d_k будет направлением спуска (т. е. поиск должен быть точнее, чем обычно).
- Пример: рассмотрим функцию

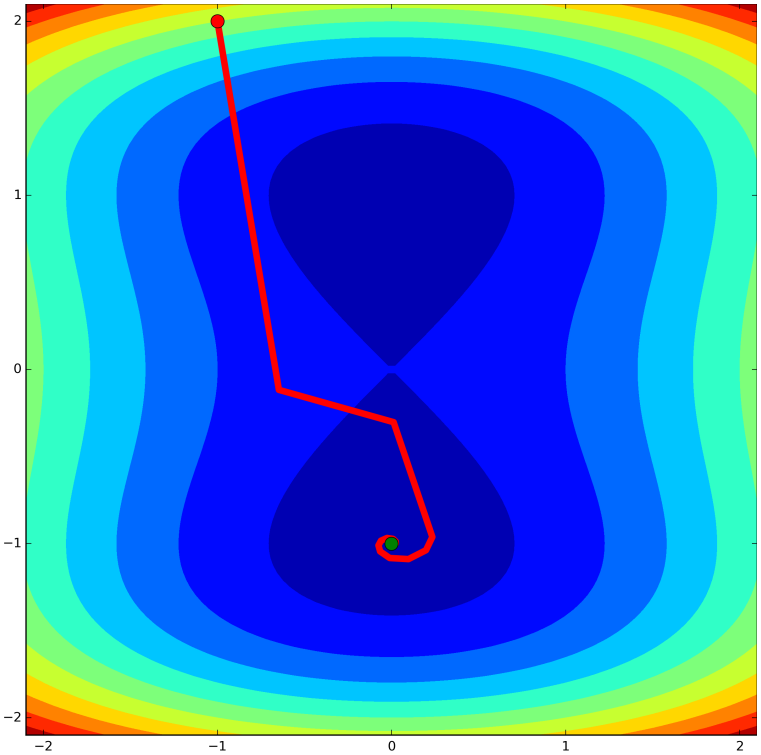
$$f(x, y) := \frac{1}{2}x^2 + \frac{1}{4}y^4 - \frac{1}{2}y^2,$$



Флетчер–Ривс и направление спуска #2



$c_2 = 0.9$



$c_2 = 0.2$

- При $c_2 > 0.5$ метод может не сходиться.
- При $c_2 < 0.5$ метод сходится.

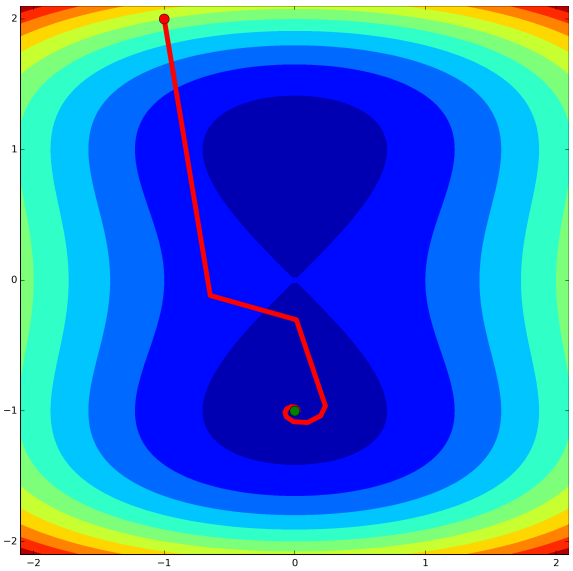
NCG и рестарт

- На квадратичной функции CG сходится $\leq$ за n итераций.
- Для функций общего вида это необязательно так.
- Многие функции вблизи оптимума очень близки к квадратичной.
- Как гарантировать быструю сходимость в окрестности оптимума?
- Для линейного CG очень важно, что $d_0 = -g_0$.
- Этого можно добиться, если делать рестарт каждые n итераций.
- Такое условие неэффективное, т.к. на практике метод обычно запускается на менее чем n итераций (например, $n \sim 10^6$).
- Более эффективным условием является условие Пауэлла:

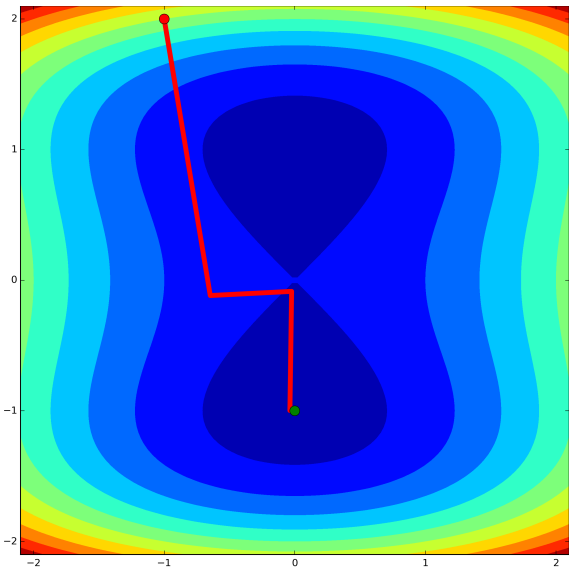
$$\text{рестарт} \Leftrightarrow \frac{|g_k^\top g_{k+1}|}{\|g_{k+1}\|_2^2} \geq \nu.$$

  Обычно берут $\nu = 0.1$.
- Основано на том, что в CG соседние градиенты ортогональны.

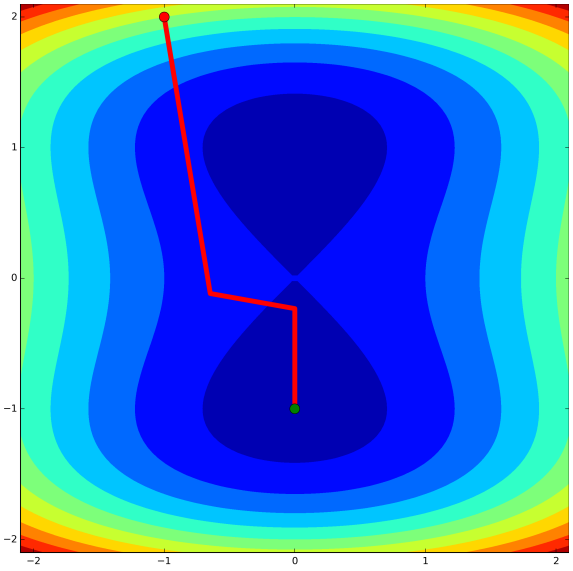
Пример: FR vs PR vs FR+restart



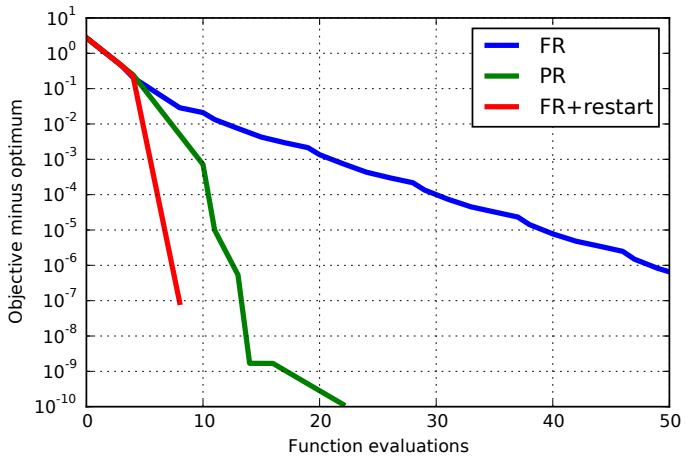
FR



PR



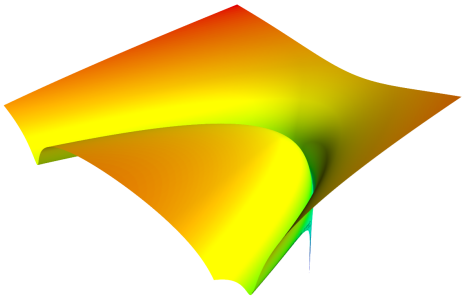
FR+restart

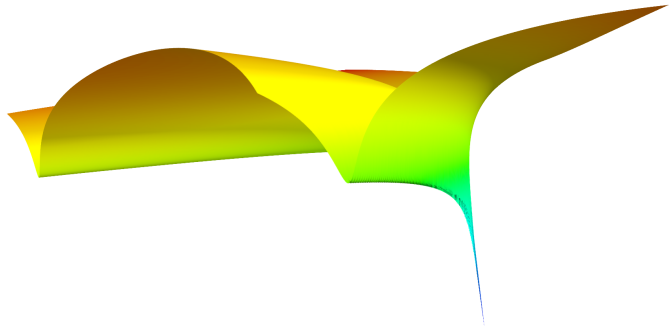


- Методы PR и FR+restart сходятся сильно быстрее, чем просто FR.

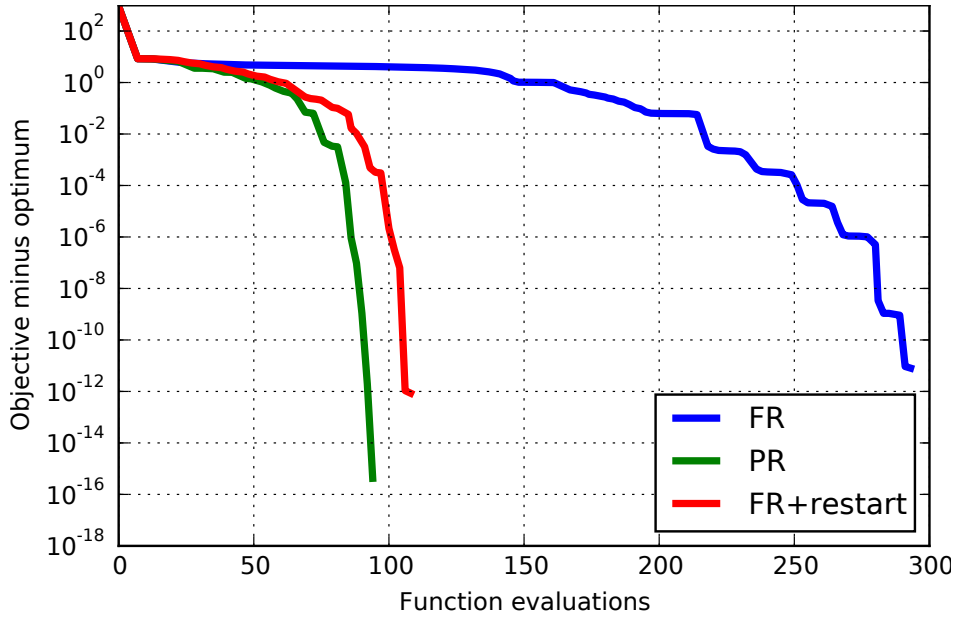
Пример #2: функция Розенброка

- Функция Розенброка: $f(x, y) := (1 - x)^2 + 100(y - x^2)^2$





- График сходимости:



Пример: логистическая регрессия

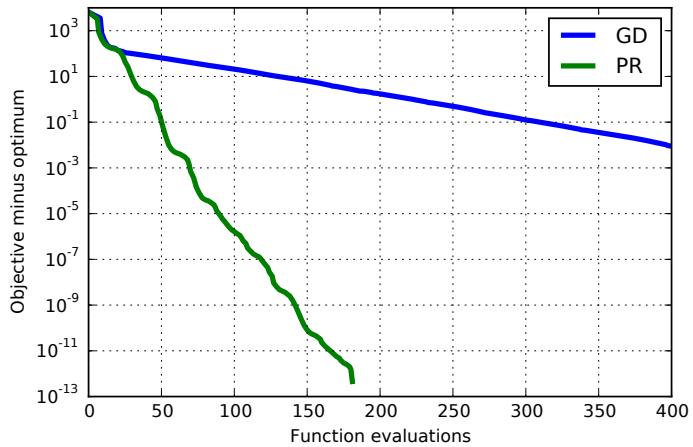
- Функция потерь:

$$f(w) := \sum_{i=1}^{N} \ln(1 + \exp(-y_i w^\top x_i)) + \frac{\lambda}{2} \|w\|_2^2$$

- Модельные данные:

```
N, D = 10000, 500
np.random.seed(0)
X = np.random.randn(N, D)
w = np.random.randn(D)
y = np.sign(X.dot(w) + np.random.randn(N))
reg_coef = 1
```

- PR vs GD:



Нелинейный CG: заключение

- Нелинейный CG является методом первого порядка.
- Существуют много разных вариантов, отличающихся выбором β_k.
- Используются сильные условия Вульфа с константой $c_2 \in (0, 0.5)$.
- Обычно присутствует рестарт.
- Метод Флетчера–Ривса без рестартов наименее эффективный.
- Наиболее популярным является метод Полака–Рибье.