Покоординатный спуск для задач с L1-регуляризацией

Илья Трофимов

13.10.2016

Машинное обучение и большие данные
ФИВТ, осень 2016

Негладкая регуляризация

- Задача LASSO

$$\underset{\beta \in \mathbb{R}^p}{\operatorname{argmin}} \left(\frac{1}{2} \sum_{i=1}^{n} (\beta^T \mathbf{x}_i - y_i)^2 + \lambda \|\beta\|_1 \right)$$

Негладкая регуляризация

- Задача LASSO

$$\underset{\beta \in \mathbb{R}^p}{\operatorname{argmin}} \left(\frac{1}{2} \sum_{i=1}^{n} (\beta^T \mathbf{x}_i - y_i)^2 + \lambda \|\beta\|_1 \right)$$

$$\|\beta\|_1 = \sum_{k=1}^{p} |\beta_k|$$

Негладкая регуляризация

- Задача LASSO

$$\underset{\boldsymbol{\beta}\in\mathbb{R}^p}{\operatorname{argmin}} \left(\frac{1}{2} \sum_{i=1}^{n} (\boldsymbol{\beta}^T \mathbf{x}_i - y_i)^2 + \lambda \|\boldsymbol{\beta}\|_1 \right)$$

$$\|\boldsymbol{\beta}\|_1 = \sum_{k=1}^{p} |\beta_k|$$

- Метод стохастического градиента, L-BFGS хорошо работают на практике, но требуют, чтобы целевая функция была дифференцируемой.
- Качество работы метода стохастического градиента существенно зависит от выбора параметров (темп обучения, скорость затухания темпа обучения)

Покоординатный спуск для LASSO

Shooting: Покоординатный спуск для LASSO

Вход: обучающая выборка $\{\mathbf{x}_i, y_i\}_{i=1}^n$, нач. приближение $\boldsymbol{\beta}$

Повторять, пока не выполнено условие останова:

1. for k = 1 ... p
2. $\beta_k \leftarrow \operatorname{argmin}_{\beta_k} \left(\frac{1}{2} \sum_{i=1}^n (\boldsymbol{\beta}^T \mathbf{x}_i - y_i)^2 + \lambda \|\boldsymbol{\beta}\|_1 \right)$

Вернуть $\boldsymbol{\beta}$

Покоординатный спуск для LASSO

Shooting: Покоординатный спуск для LASSO

Вход: обучающая выборка $\{\mathbf{x}_i, y_i\}_{i=1}^n$, нач. приближение $\boldsymbol{\beta}$

Повторять, пока не выполнено условие останова:

1. for k = 1 ... p
2. $$\beta_k \leftarrow S\left(\frac{\sum_{i=1}^n x_{ik}(y_i - (s_i - \beta_k x_{ik}))}{\sum_{i=1}^n x_{ik}^2}, \frac{\lambda}{\sum_{i=1}^n x_{ik}^2}\right), \text{ где } s_i = \boldsymbol{\beta}^T x_i$$

Вернуть $\boldsymbol{\beta}$

Покоординатный спуск для LASSO

Shooting: Покоординатный спуск для LASSO

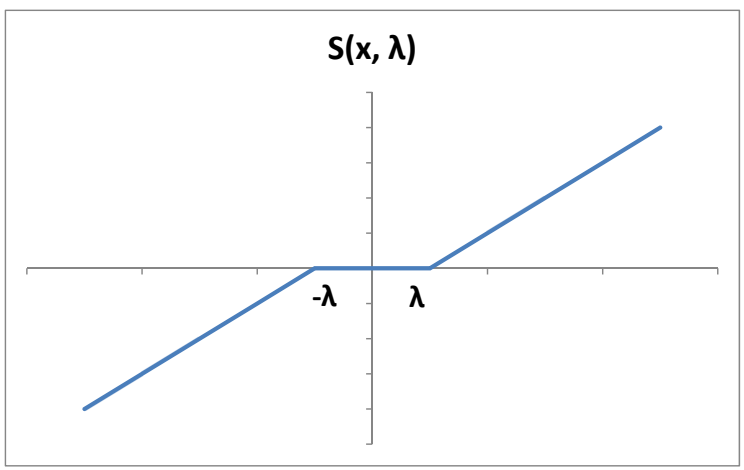
Вход: обучающая выборка $\{\mathbf{x}_i, y_i\}_{i=1}^n$, нач. приближение $\boldsymbol{\beta}$

Повторять, пока не выполнено условие останова:

1. for k = 1 ... p
2. $$\beta_k \leftarrow S\left(\frac{\sum_{i=1}^n x_{ik}(y_i - (s_i - \beta_k x_{ik}))}{\sum_{i=1}^n x_{ik}^2}, \frac{\lambda}{\sum_{i=1}^n x_{ik}^2}\right), \text{ где } s_i = \boldsymbol{\beta}^T x_i$$

Вернуть $\boldsymbol{\beta}$

$$S(x, \lambda) = \operatorname{sgn}(x)\max(|x| - \lambda, 0) \qquad \text{функция soft-threshold}$$



Покоординатный спуск для LASSO

Эффективная реализация покоординатного спуска для LASSO

Вход: обучающая выборка X, начальное приближение β.

$s \leftarrow 0$ (длина s равна n)

Повторять, пока не выполнено условие останова:

1. for k = 1 ... p
2. $\beta_k^* \leftarrow S\left(\frac{\sum_{i=1}^n x_{ik}(y_i - (s_i - \beta_k x_{ik}))}{\sum_{i=1}^n x_{ik}^2}, \frac{\lambda}{\sum_{i=1}^n x_{ik}^2}\right)$, где $s_i = \beta^T \mathbf{x}_i$
3. Обновить компоненты s_i, такие что $x_{ik} \neq 0$:

$$s_i \leftarrow s_i + (\beta_k^* - \beta_k)x_{ik}$$

4. $\beta_k \leftarrow \beta_k^*$

Вернуть β

Покоординатный спуск для LASSO

Эффективная реализация покоординатного спуска для LASSO

Вход: обучающая выборка X, начальное приближение β.

$s \leftarrow 0$ (длина s равна n)

Повторять, пока не выполнено условие останова:

1. for k = 1 ... p
2. $\beta_k^* \leftarrow S\left(\frac{\sum_{i=1}^n x_{ik}(y_i-(s_i-\beta_k x_{ik}))}{\sum_{i=1}^n x_{ik}^2}, \frac{\lambda}{\sum_{i=1}^n x_{ik}^2}\right)$, где $s_i = \beta^T \mathbf{x}_i$
3. Обновить компоненты s_i, такие что $x_{ik} \neq 0$:

$$s_i \leftarrow s_i + (\beta_k^* - \beta_k)x_{ik}$$

4. $\beta_k \leftarrow \beta_k^*$

Вернуть β

Если поддерживать в RAM вектора $\beta, s = (\beta^T \mathbf{x}_i)$ то сложность одной итерации будет равна

Покоординатный спуск для LASSO

Эффективная реализация покоординатного спуска для LASSO

Вход: обучающая выборка X, начальное приближение β.

$s \leftarrow 0$ (длина s равна n)

Повторять, пока не выполнено условие останова:

1. for k = 1 ... p
2. $\beta_k^* \leftarrow S\left(\frac{\sum_{i=1}^n x_{ik}(y_i-(s_i-\beta_k x_{ik}))}{\sum_{i=1}^n x_{ik}^2}, \frac{\lambda}{\sum_{i=1}^n x_{ik}^2}\right)$, где $s_i = \beta^T \mathbf{x}_i$
3. Обновить компоненты s_i, такие что $x_{ik} \neq 0$:

$$s_i \leftarrow s_i + (\beta_k^* - \beta_k)x_{ik}$$

4. $\beta_k \leftarrow \beta_k^*$

Вернуть β

Если поддерживать в RAM вектора $\beta, s = (\beta^T \mathbf{x}_i)$ то сложность одной итерации будет равна $\underline{O(nnz)}$

LASSO с весами

Если объекты обучающей выборки имеют веса w_i, то фунция эмпирического риска будет равна

$$\underset{\boldsymbol{\beta}}{\operatorname{argmin}} \frac{1}{2} \sum_{i=1}^{n} w_i (y_i - \boldsymbol{\beta}^T \mathbf{x}_i)^2 + \lambda \|\boldsymbol{\beta}\|_1$$

В этом случае шаг алгоритма Shooting приобретает вид:

$$\beta_k \leftarrow S\left(\frac{\sum_{i=1}^{n} w_i x_{ik} (y_i - (s_i - \beta_k x_{ik}))}{\sum_{i=1}^{n} w_i x_{ik}^2}, \frac{\lambda}{\sum_{i=1}^{n} w_i x_{ik}^2} \right) \qquad (1)$$

Обобщенные линейные модели

Моделируем $y \sim f(\beta^T \mathbf{x})$

Примеры: линейная регрессия, логистическая регрессия, пуассоновская регрессия, пробит-регрессия.

$$L(\beta) = \sum_{i=1}^{n} \ell(\beta^T \mathbf{x}_i, y_i)$$

$$\underset{\beta}{\operatorname{argmin}} \left(L(\beta) + \lambda \|\beta\|_1 \right)$$

Метод Ньютона?

Хочется делать оптимизацию, используя информацию о Гессиане

$$L(\beta + \Delta\beta) \approx L(\beta) + L'(\beta)^T \Delta\beta + \frac{1}{2}\Delta\beta^T \nabla^2 L(\beta)\Delta\beta$$

Метод Ньютона?

Хочется делать оптимизацию, используя информацию о Гессиане

$$L(\beta + \Delta\beta) \approx L(\beta) + L'(\beta)^T \Delta\beta + \frac{1}{2}\Delta\beta^T \nabla^2 L(\beta)\Delta\beta$$

$$\begin{aligned}\Delta\beta^* &= \underset{\Delta\beta}{\operatorname{argmin}} \left(L(\beta) + L'(\beta)^T \Delta\beta + \frac{1}{2}\Delta\beta^T \nabla^2 L(\beta)\Delta\beta \right) \\ &= -(\nabla^2 L(\beta))^{-1} L'(\beta)\end{aligned}$$

Метод Ньютона?

Хочется делать оптимизацию, используя информацию о Гессиане

$$L(\beta + \Delta\beta) \approx L(\beta) + L'(\beta)^T \Delta\beta + \frac{1}{2}\Delta\beta^T \nabla^2 L(\beta)\Delta\beta$$

$$\begin{aligned}\Delta\beta^* &= \operatorname*{argmin}_{\Delta\beta}\left(L(\beta) + L'(\beta)^T \Delta\beta + \frac{1}{2}\Delta\beta^T \nabla^2 L(\beta)\Delta\beta\right) \\ &= -(\nabla^2 L(\beta))^{-1} L'(\beta)\end{aligned}$$

$$\beta \leftarrow \beta + \Delta\beta^*$$

Метод Ньютона?

Нужно найти: $\operatorname{argmin}_{\boldsymbol{\beta}} \left(L(\boldsymbol{\beta}) + \lambda \|\boldsymbol{\beta}\|_1 \right)$

Метод Ньютона?

Нужно найти: $\operatorname{argmin}_{\beta}\left(L(\beta)+\lambda\|\beta\|_1\right)$

$$L(\beta+\Delta\beta)=\sum_{i=1}^{n}\ell(y_i,(\beta+\Delta\beta)^T x_i)\approx$$

$$\approx\sum_{i=1}^{n}\left\{\ell(y_i,\beta^T x_i)+\frac{\partial\ell(y_i,\beta^T x_i)}{\partial\hat{y}}\Delta\beta^T\mathbf{x}_i+\frac{1}{2}(\Delta\beta^T\mathbf{x}_i)\frac{\partial^2\ell(y_i,\beta^T x_i)}{\partial\hat{y}^2}(\Delta\beta^T\mathbf{x}_i)\right\}$$

$$=C(\beta)+\frac{1}{2}\sum_{i=1}^{n}w_i(z_i-\Delta\beta^T\mathbf{x}_i)^2$$

$$w_i=\frac{\partial^2\ell(y_i,\beta^T x_i)}{\partial\hat{y}^2}$$

$$z_i=-\frac{\partial\ell(y_i,\beta^T x_i)/\partial\hat{y}}{\partial^2\ell(y_i,\beta^T x_i)/\partial\hat{y}^2}$$

Метод Ньютона?

Нужно найти: $\operatorname{argmin}_{\beta}\left(L(\beta)+\lambda\|\beta\|_1\right)$

$$L(\beta+\Delta\beta)=\sum_{i=1}^{n}\ell(y_i,(\beta+\Delta\beta)^T x_i)\approx$$

$$\approx\sum_{i=1}^{n}\left\{\ell(y_i,\beta^T x_i)+\frac{\partial\ell(y_i,\beta^T x_i)}{\partial\hat{y}}\Delta\beta^T\mathbf{x}_i+\frac{1}{2}(\Delta\beta^T\mathbf{x}_i)\frac{\partial^2\ell(y_i,\beta^T x_i)}{\partial\hat{y}^2}(\Delta\beta^T\mathbf{x}_i)\right\}$$

$$=C(\beta)+\frac{1}{2}\sum_{i=1}^{n}w_i(z_i-\Delta\beta^T\mathbf{x}_i)^2$$

$$w_i=\frac{\partial^2\ell(y_i,\beta^T x_i)}{\partial\hat{y}^2}$$

$$z_i=-\frac{\partial\ell(y_i,\beta^T x_i)/\partial\hat{y}}{\partial^2\ell(y_i,\beta^T x_i)/\partial\hat{y}^2}$$

$$\operatorname*{argmin}_{\Delta\beta}\left(\frac{1}{2}\sum_{i=1}^{n}w_i(z_i-\Delta\beta^T\mathbf{x}_i)^2+\lambda\|\beta+\Delta\beta\|_1\right)$$

Логистическая регрессия

$$P(y = +1|x) = \frac{1}{1 + \exp(-\beta^T \mathbf{x})}$$

Минус лог-правдоподобие (эмпирический риск) $L(\beta)$

$$L(\beta) = \sum_{i=1}^{n} \log(1 + \exp(-y_i \beta^T \mathbf{x}_i))$$

$$z_i = \frac{(y_i + 1)/2 - p(\mathbf{x}_i)}{p(\mathbf{x}_i)(1 - p(\mathbf{x}_i))}$$

$$w_i = p(\mathbf{x}_i)(1 - p(\mathbf{x}_i))$$

$$p(\mathbf{x}_i) = \frac{1}{1 + e^{-\beta^T \mathbf{x}_i}}$$

Алгоритм GLMNET

Алгоритм GLMNET

Вход: обучающая выборка $\{\mathbf{x}_i, y_i\}_{i=1}^n$, нач. приближение $\boldsymbol{\beta}$, параметр регуляризации λ

$s_i \leftarrow \boldsymbol{\beta}^T \mathbf{x}_i$, $\Delta s_i \leftarrow 0$ (вектора размера n)

Повторять, пока не выполено условие останова:

1. for k = 1 ... p
2. $$\Delta\beta_k \leftarrow \underset{\Delta\beta_k}{\operatorname{argmin}} \left(\frac{1}{2} \sum_{i=1}^{n} w_i (z_i - \Delta\boldsymbol{\beta}^T \mathbf{x}_i)^2 + \lambda \|\boldsymbol{\beta} + \Delta\boldsymbol{\beta}\|_1 \right)$$
3. Обновить компоненты Δs_i, такие что $x_{ik} \neq 0$:
$$\Delta s_i \leftarrow \Delta s_i + \Delta\beta_k x_{ik}$$
4. Найти α с помощью линейного поиска по правилу Armijo
5. $\boldsymbol{\beta} \leftarrow \boldsymbol{\beta} + \alpha\Delta\boldsymbol{\beta}$
6. $s \leftarrow s + \alpha\Delta s$

Вернуть $\boldsymbol{\beta}$

Алгоритм GLMNET

Целевая функция оптимизации (регуляризованное минус лог-правдободобие):

$$f(\beta) = \sum_{i=1}^{n} \ell(\beta^T x_i, y_i) + \lambda \|\beta\|_1$$

Линейный поиск по правилу Armijo

Вход α_{init}, $0 < b < 1, 0 < \sigma < 1, 0 \leq \gamma < 1$

1. Параметр α должен быть равен максимальному элементу последовательности $\{\alpha_{init}, \alpha_{init} b, \alpha_{init} b^2, ...\}$, такому что

$$f(\beta + \alpha \Delta\beta) \leq f(\beta) + \alpha\sigma D$$

$$D = f'(\beta)^T \Delta\beta + \gamma \Delta\beta^T \nabla^2 f(\beta) \Delta\beta + \lambda \left(\|\beta + \Delta\beta\|_1 - \|\beta\|_1 \right)$$

Вернуть α

Алгоритм GLMNET

Целевая функция оптимизации (регуляризованное минус лог-правдободобие):

$$f(\beta) = \sum_{i=1}^{n} \ell(\beta^T x_i, y_i) + \lambda \|\beta\|_1$$

Линейный поиск по правилу Armijo

Вход α_{init}, $0 < b < 1, 0 < \sigma < 1, 0 \leq \gamma < 1$

1. Параметр α должен быть равен максимальному элементу последовательности $\{\alpha_{init}, \alpha_{init} b, \alpha_{init} b^2, ...\}$, такому что

$$f(\beta + \alpha \Delta\beta) \leq f(\beta) + \alpha\sigma D$$

$$D = f'(\beta)^T \Delta\beta + \gamma \Delta\beta^T \nabla^2 f(\beta) \Delta\beta + \lambda \left(\|\beta + \Delta\beta\|_1 - \|\beta\|_1 \right)$$

Вернуть α

Примечание. В исходном алгоритме GLMNET не было линейного поиска, он необходим для гарантии сходимости алгоритма

Regularization path

На практике обычно нужно найти оптимальную регуляризацию λ, чтобы качество на тестовом множестве было максимальным.

Нахождение regularization path с помощью GLMNET
Вход: обучающая выборка X, убывающий список $\{\lambda_t\}_1^T$
$\beta \leftarrow 0$
for t = 1 ... T

1. Использовать β в качестве начального приближения
2. $\beta \leftarrow$ результат обучения с помощью GLMNET, регуляризация λ_t
3. Вернуть (λ_t, β)

Regularization path

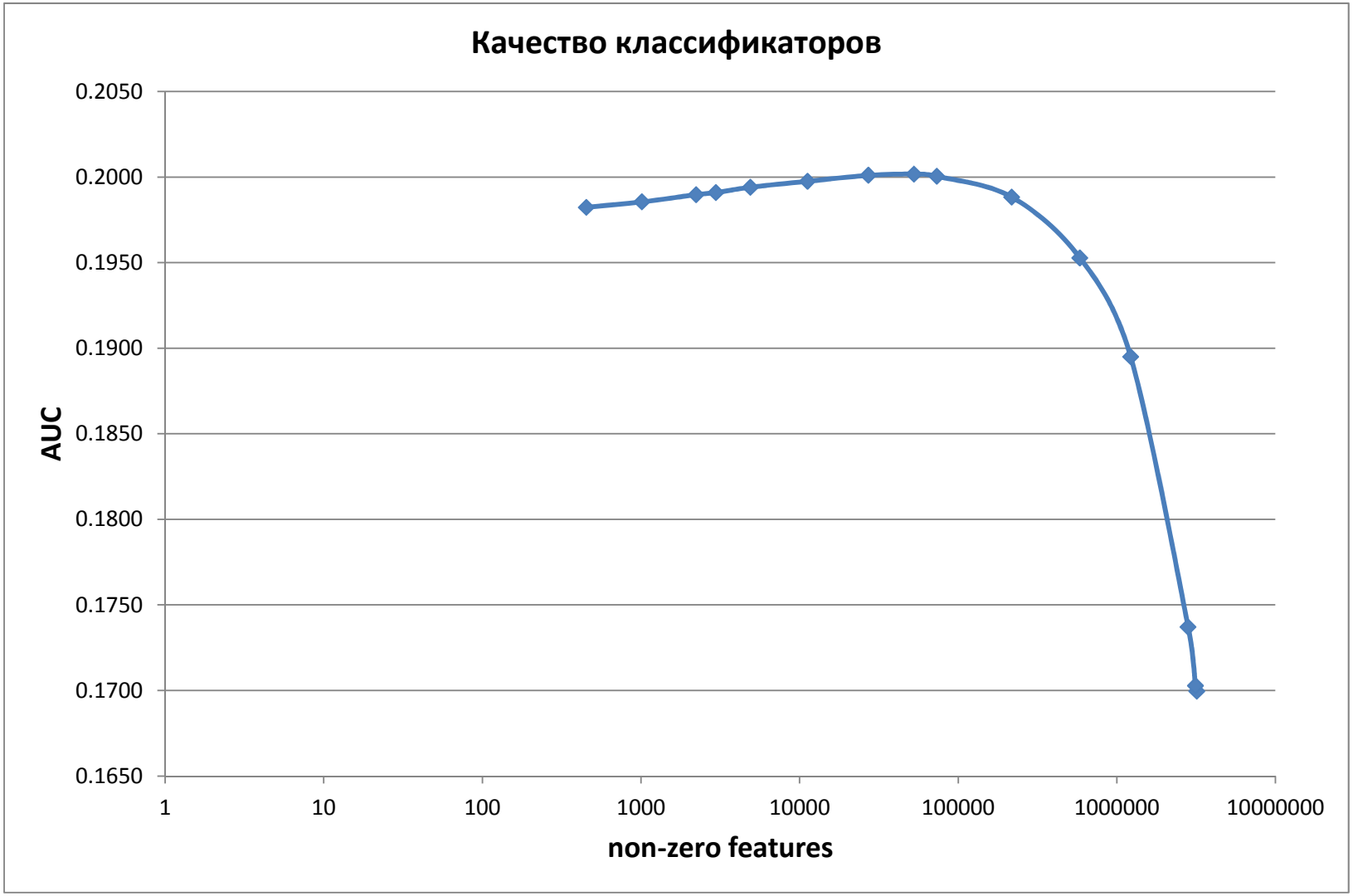
На практике обычно нужно найти оптимальную регуляризацию λ, чтобы качество на тестовом множестве было максимальным.

Нахождение regularization path с помощью GLMNET
Вход: обучающая выборка X, убывающий список $\{\lambda_t\}_1^T$
$\beta \leftarrow 0$
for t = 1 ... T

1. Использовать β в качестве начального приближения
2. $\beta \leftarrow$ результат обучения с помощью GLMNET, регуляризация λ_t
3. Вернуть (λ_t, β)

Обычно берут $\lambda_t = \lambda_0 \times a^{-t}$, $a < 1$, где λ_0 - это минимальная регуляризация, при которой $\beta = 0$.

Regularization path



Как распараллелить методы покоординатного спуска?

Подход 1: задействуем несколько ядер CPU или GPU

Как распараллелить методы покоординатного спуска?

Подход 1: задействуем несколько ядер CPU или GPU

- Делать шаги независимо по разным координатам. Шаги могут кофликтовать и целевая функция не будет уменьшаться.

Как распараллелить методы покоординатного спуска?

Подход 1: задействуем несколько ядер CPU или GPU

- Делать шаги независимо по разным координатам. Шаги могут кофликтовать и целевая функция не будет уменьшаться.
- Распараллелить вычисление

$$\beta_k^* \leftarrow S\left(\frac{\sum_{i=1}^n x_{ik}(y_i - (s_i - \beta_k x_{ik}))}{\sum_{i=1}^n x_{ik}^2}, \frac{\lambda}{\sum_{i=1}^n x_{ik}^2}\right)$$

$$s_i \leftarrow s_i + \sum_{i:x_{ik}\neq 0} (\beta_k^* - \beta_k) x_{ik}$$

Как распараллелить методы покоординатного спуска?

Подход 2: используем несколько машин.

Как распараллелить методы покоординатного спуска?

Подход 2: используем несколько машин. Естественно, чтобы каждая машина отвечала за свое подмножество переменных

Как распараллелить методы покоординатного спуска?

Распределенный покоординатный спуск

Вход: Обучающая выборка $\{\mathbf{x}_i, y_i\}_{i=1}^n$, разделенная на M частей по переменным.

$\beta \leftarrow 0, \Delta\beta \leftarrow 0$, где m - номер машины

Пока не выполнено условие останова:

1. Выполнить параллельно на M машинах:
2. Выполнить шаги по переменным, сохранить $\Delta\beta^m$, $(\Delta(\beta^m)^T\mathbf{x}_i))$
3. Суммировать вектора $\Delta\beta^m$, $(\Delta(\beta^m)^T\mathbf{x}_i)$ с помощью `MPI_AllReduce`
4. $\Delta\beta \leftarrow \sum_{m=1}^M \Delta\beta^m$
5. $(\Delta\beta^T\mathbf{x}_i) \leftarrow \sum_{m=1}^M (\Delta(\beta^m)^T\mathbf{x}_i)$
6. Найти α с помощью алгоритма линейного поиска
7. $\beta \leftarrow \beta + \alpha\Delta\beta$,
8. $(\exp(\beta^T\mathbf{x}_i)) \leftarrow (\exp(\beta^T\mathbf{x}_i + \alpha\Delta\beta^T\mathbf{x}_i))$

Сходимость

Алгоритм GLMNET и его распределенный вариант сходятся для LASSO и логистической регресии с L1-регуляризацией. Вектор весов β сходится Q-линейно.

Реализации

- *sklearn.linear_model.Lasso*
- *sklearn.linear_model.ElasticNet*
- программа *liblinear*
- пакет *glmnet* в *R*

Библиография

LASSO FU, W. J. (1998). Penalized regressions: The bridge versus the lasso. J. Comput. Graph. Statist. 7 397–416. MR1646710

logistic Friedman, J., Hastie, T., Höfling, H., Tibshirani, R., Pathwise coordinate optimization, The Annals of Applied Statistics, 2007, Vol 1, 2, 302-332

GPU Suchard, M. E, Simpson, S. E., Zorych, I., Ryan, P., Madigan, D. (2013). Massive parallelization of serial inference algorithms for a complex generalized linear model. ACM Transactions on Modeling and Computer Simulation, 23(1), 1–18. doi:10.1145/2414416.2414791

Distr I. Trofimov, A. Genkin. Distributed Coordinate Descent for L1-regularized Logistic Regression. http://arxiv.org/abs/1411.6520

Converge Tseng, P., Yun, S. (2009). A coordinate gradient descent method for nonsmooth separable minimization. Mathematical Programming, 117(1-2), 387–423. doi:10.1007/s10107-007-0170-0