УДК 519.865

Применение алгоритма зеркального спуска для групповой обработки данных в случайной среде

Д.Н. Шиян[1]***, А.В. Колногоров***[1]

[1]Новгородский государственный университет имени Ярослава Мудрого

Рассматривается оптимизация алгоритма зеркального спуска для групповой обработки данных в двухальтернативной среде.

Пусть есть некоторая случайная среда, в которой $X = \{1,2\}$ это множество доступных действий. В каждый момент времени $t = 1,2,\ldots,$ можно выбрать только одно из доступных действий $x_t \in X$. Применение определенного действия может принести доход (1) или ничего не принести (0). Основная задача – максимизация дохода за фиксированный промежуток времени, при условии, что распределение вероятностей выигрыша заранее неизвестно, но не изменяется со временем.

Вводится функция потерь $L_T = \max(\rho_1, \rho_2) \cdot T - \eta_T$, где η_T - общий полученный доход за время T, ρ_1, ρ_2 - вероятности выигрыша при выборе первого и второго действия соответственно.

Исследование алгоритмов производится с помощью метода Монте-Карло с количеством итераций равном 10000. Вероятности выигрыша при моделировании имеют следующее распределение: $\overline{\rho} = (\rho_1; \rho_2) = (\rho + d\sqrt{D/T}; \rho - d\sqrt{D/T})$, где ρ - некоторая фиксированная вероятность, $D = 0.25$ – максимальная возможная дисперсия, T – горизонт управления, d – параметр, характеризующий величину различия между вероятностями.

Алгоритмы имеют настраиваемый параметр $\beta_0 > 0$, оптимальное значение которого вычисляется экспериментально. При моделировании используются значения нормализованной функции потерь, которая определена как $L'_T(\beta_0, \rho, d) = (DT)^{-1/2} L_T(\beta_0, \rho, d)$.

Базовый вариант алгоритма (Алгоритм 1), обрабатывающий данные по одному основан на оригинальном алгоритме зеркального спуска [1] с некоторыми модификациями.

Введем вектор вероятностей $\overline{p}_n = (p_1; p_2)$, $p_1, p_2 > 0$ и $p_1 + p_2 = 1$, двойственный вектор $\overline{\zeta}_n = (\zeta_1; \zeta_2)$ и вектор стохастического градиента $\overline{u}_n = (u_1; u_2)$. Для всех алгоритмов полагается $\overline{p}_0 = (0.5; 0.5)$ и $\overline{\zeta}_0 = (0;0)$. Распределение Гиббса $\overline{G}_\beta$ определяется следующим образом: $\overline{G}_\beta(\overline{\zeta}) = S_\beta^{-1}(\overline{\zeta}) \cdot \left(e^{-\zeta^{(1)}/\beta}; e^{-\zeta^{(2)}/\beta}\right)$, где $S_\beta(\overline{\zeta}) = e^{-\zeta^{(1)}/\beta} + e^{-\zeta^{(2)}/\beta}$. Теперь Алгоритм 1 можно записать в следующей форме:

Алгоритм 1

Для всех *n* от 1 до *N* выполняем следующие действия:

1) Выбираем действие y_n, по вектору $\overline{p}_{n-1}$: $P(y_n = l) = p_{n-1}^{(l)}$, $l = 1,2$;
2) Рассчитываем доход ξ_n: $P(\xi_n = 1 \mid y_n = l) = \rho_l$, $P(\xi_n = 0 \mid y_n = l) = 1 - \rho_l$;
3) Вычисляем стохастический градиент $\overline{u}_n(\overline{p}_{n-1})$:

$$\overline{u}_n(\overline{p}_{n-1}) = \left\{\left(\frac{1-\xi_n}{p_{n-1}^{(1)}}; 0\right), \text{при } y_n = 1,\right\}, \left\{\left(0; \frac{1-\xi_n}{p_{n-1}^{(2)}}\right), \text{при } y_n = 2\right\};$$

4) Обновляем вектора $\overline{\zeta}_n$ и $\overline{p}_n$: $\overline{\zeta}_n = \overline{\zeta}_{n-1} + \overline{u}_n(\overline{p}_{n-1})$, $\overline{p}_n = \overline{G}_{\beta_n}(\overline{\zeta}_n)$, где $\beta_n = \beta_0 \sqrt{D(n+1)} = \beta_0 \cdot 0.5\sqrt{n+1}$

Для данного алгоритма при $\rho = 0.1$ и $\beta_0 = 2.25$ была получена численная оценка $r_1 = \inf\limits_{\beta_0 > 0} \max\limits_{1 \le d \le 10,\ 0.1 < p < 0.9} L'_N(\beta_0, \rho, d) \approx 2.0.$

Пакетный вариант (Алгоритм 2) [2] предполагает более глубокую модификацию. Для пакетного алгоритма $N = TM$, где *M* – размер пакета, а *T* – количество этапов. Оказывается, если

разделить все данные на пакеты (группы) определенного размера, то можно повысить скорость работы алгоритма, без падения его эффективности. Для пакетного алгоритма время обработки зависит не от общего объема данных, а от количества групп, на которые эти данные разбиты.

Алгоритм 2

Для всех t от 1 до T выполняем следующие действия:

1) Вычисляем, для какой части пакета следует использовать первое действие, а для какой второе: $M_t^{(l)} = p_{t-1}^{(l)} \times M,\ l = 1,2;$
2) Применяем l-е действие $[M_t^{(l)}]$ раз и получаем вектор $\overline{\eta}_t : \eta_t^{(l)} = \sum_{n=(t-1)M+1}^{tM} (1 - \xi_n \mid y_n = l);$
3) Вычисляем стохастический градиент $\overline{u}_t(\overline{p}_{t-1}):\ \overline{u}_t(\overline{p}_{t-1}) = \left(\eta_t^{(1)} / p_{t-1}^{(1)}; \eta_t^{(2)} / p_{t-1}^{(2)}\right);$
4) Обновляем вектора $\overline{\zeta}_t$ и $\overline{p}_t$: $\overline{\zeta}_t = \overline{\zeta}_{t-1} + \overline{u}_t(\overline{p}_{t-1}),\quad \overline{p}_t = \overline{G}_{\beta_t}(\overline{\zeta}_t),$ где $\beta_t = \beta_0 \sqrt{DM(t+0.5)} = \beta_0 \cdot 0.5 \sqrt{M(t+0.5)}$

Оценка r_2, полученная для пакетного алгоритма примерно в два раза лучше оценки r_1.

Однако пакетный алгоритм имеет определенный недостаток при работе с пакетами достаточного большого размера и небольшом количестве этапов $T < 100$. Причиной этого является так называемый «эффект начального пакета», который связан с начальным распределением $\overline{p}_0 = (0.5;0.5)$. Для минимизации этого эффекта разработан комбинированный алгоритм (Алгоритм 3) [2], который на начальном этапе получает оценки параметров $\overline{p}_{M_0}$ и $\overline{\zeta}_{M_0}$ с помощью базового алгоритма, а затем использует их для работы пакетного алгоритма.

Алгоритм 3

1) Определяем размер начального этапа M_0. Пусть $N = TM,\ 0 \le \alpha \le 1,$ тогда $M_0 = \alpha \cdot T;$
2) Применяем базовый алгоритм для $n = 1,\ldots,M_0$ и получаем значения $\overline{p}_{M_0}$ и $\overline{\zeta}_{M_0}$;
3) Применяем пакетный алгоритм для $n = M_0 + 1,\ldots,N$.

Комбинированный алгоритм сохраняет скорость обработки пакетного алгоритма и обеспечивает сходимость при достаточно больших размерах пакета и небольшом количестве этапов обработки.

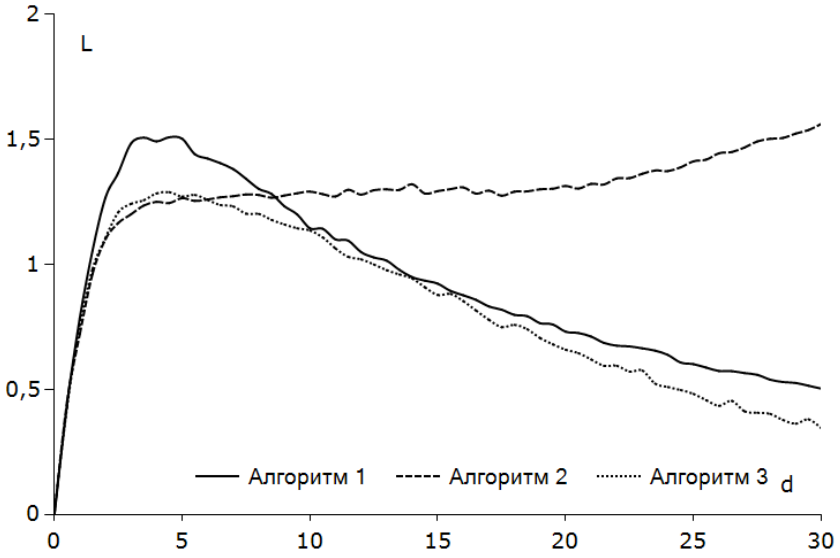

Рис. 1. Значения нормализованной функции потерь для различных алгоритмов при N = 10000 и M = 500

Литература

1. *Juditsky A., Nazin A.V., Tsybakov A.B., Vayatis N.* Gap-free Bounds for Stochastic Multi-Armed Bandit. // Proc. 17th World Congress IFAC (Seoul, Korea, July 6 - 11). 2008. P. 11560 - 11563
2. *Kolnogorov A., Nazin A., Shiyan D.* Two-Armed Bandit Problem, Data Processing, and Parallel Version of the Mirror Descent Algorithm. // arXiv:1705.09977v1 [math.ST] 28 May 2017