Оптимизация гиперпараметров

Илья Трофимов

08.09.2016

Машинное обучение и большие данные
ФИВТ, осень 2016

Подбор гиперпараметров

Многие методы машинного обучения имеют гиперпараметры, от которых существенно зависит качество предсказания:

- SVM - параметр C
- Коэффициенты L1/L2 регуляризации
- Gradient Boosted Decision Trees - размер композиции, высота дерева, shrinkage, мин. кол-во объектов в листе.
- Онлайн обучение Vowpal Wabbit: $\eta_t = \lambda d^k \left(\frac{t_0}{t_0 + w_t}\right)^p$, параметры λ, d, t_0, p
- Алгоритм обучения, как правило, сводится к алгоритму численной оптимизации, у которого тоже могут быть параметры.

Подбор гиперпараметров

Базовый алгоритм тестирования гиперпараметров - функция $f(x)$
Вход: вектор гиперпараметров $x \in \mathbb{R}^p$, обучающая выборка, тестовая выборка, алгоритм обучения

1. Обучить алгоритм на обучающей выборке, используя параметры x
2. Вернуть качество, измеренное по тестовой выборке

Подбор гиперпараметров

Базовый алгоритм тестирования гиперпараметров - функция $f(x)$
Вход: вектор гиперпараметров $x \in \mathbb{R}^p$, обучающая выборка, тестовая выборка, алгоритм обучения

1. Обучить алгоритм на обучающей выборке, используя параметры x
2. Вернуть качество, измеренное по тестовой выборке

Альтернатива - кросс-валидация

Подбор гиперпараметров

Базовый алгоритм тестирования гиперпараметров - функция $f(x)$
Вход: вектор гиперпараметров $x \in \mathbb{R}^p$, обучающая выборка, тестовая выборка, алгоритм обучения

1. Обучить алгоритм на обучающей выборке, используя параметры x
2. Вернуть качество, измеренное по тестовой выборке

Альтернатива - кросс-валидация

Задача сводится к оптимизации фукции $f(x)$, для которой мы умеем вычислять только значения, а не производные

Подбор гиперпараметров

Что делать?

- Перебор по сетке - нормальное решение, если комбинаций параметров не очень много

Подбор гиперпараметров

Что делать?

- Перебор по сетке - нормальное решение, если комбинаций параметров не очень много
- Для некоторых параметров имеет смысл использовать логарифмическую сетку (например, параметры L1/L2 регуляризации, learning rate)

Подбор гиперпараметров

Что делать?

- Перебор по сетке - нормальное решение, если комбинаций параметров не очень много
- Для некоторых параметров имеет смысл использовать логарифмическую сетку (например, параметры L1/L2 регуляризации, learning rate)
- Разные комбинации можно пробовать параллельно

Подбор гиперпараметров

Что делать?

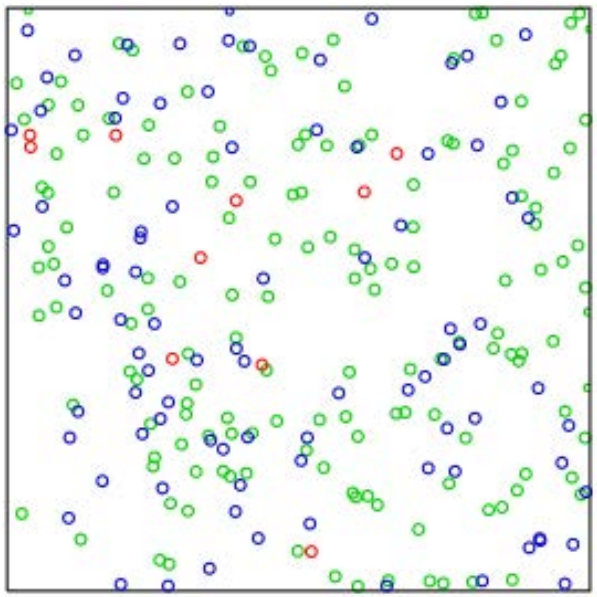
- Перебор по сетке - нормальное решение, если комбинаций параметров не очень много
- Для некоторых параметров имеет смысл использовать логарифмическую сетку (например, параметры L1/L2 регуляризации, learning rate)
- Разные комбинации можно пробовать параллельно
- Можно пробовать случайные точки по сетке

Подбор гиперпараметров

Что делать?

- Перебор по сетке - нормальное решение, если комбинаций параметров не очень много
- Для некоторых параметров имеет смысл использовать логарифмическую сетку (например, параметры L1/L2 регуляризации, learning rate)
- Разные комбинации можно пробовать параллельно
- Можно пробовать случайные точки по сетке
- При разных значениях параметров обучение будет занимать разное время. Например, чем больше нейронная сеть - тем больше вычислений нужно делать.

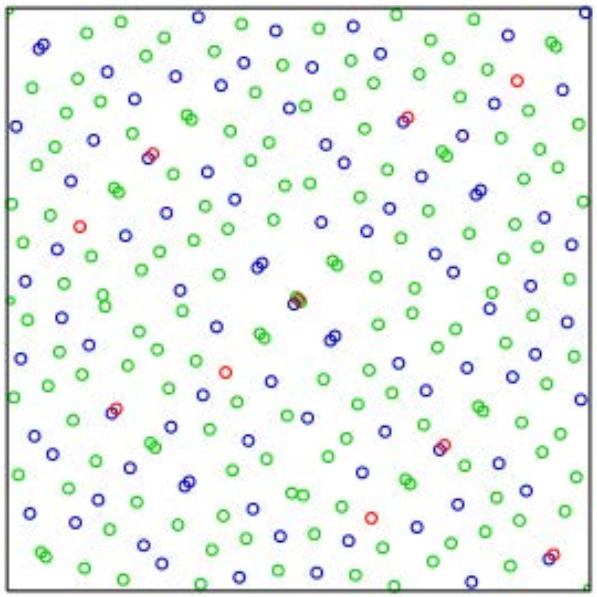
Случайный поиск



---

https://en.wikipedia.org/wiki/Sobol_sequence

Последовательность Соболя



https://en.wikipedia.org/wiki/Sobol_sequence

Метод Нелдера-Мида

aka метод деформируемого многогранника, симплекс-метод, "амеба"

Метод Нелдера-Мида

aka метод деформируемого многогранника, симплекс-метод, "амеба"

$$\underset{x \in D,\ D \text{ - гиперкуб}}{\operatorname{argmin}} \quad f(x)$$

Метод Нелдера-Мида

aka метод деформируемого многогранника, симплекс-метод, "амеба"

$$\underset{x \in D,\ D \text{ - гиперкуб}}{\operatorname{argmin}} \quad f(x)$$

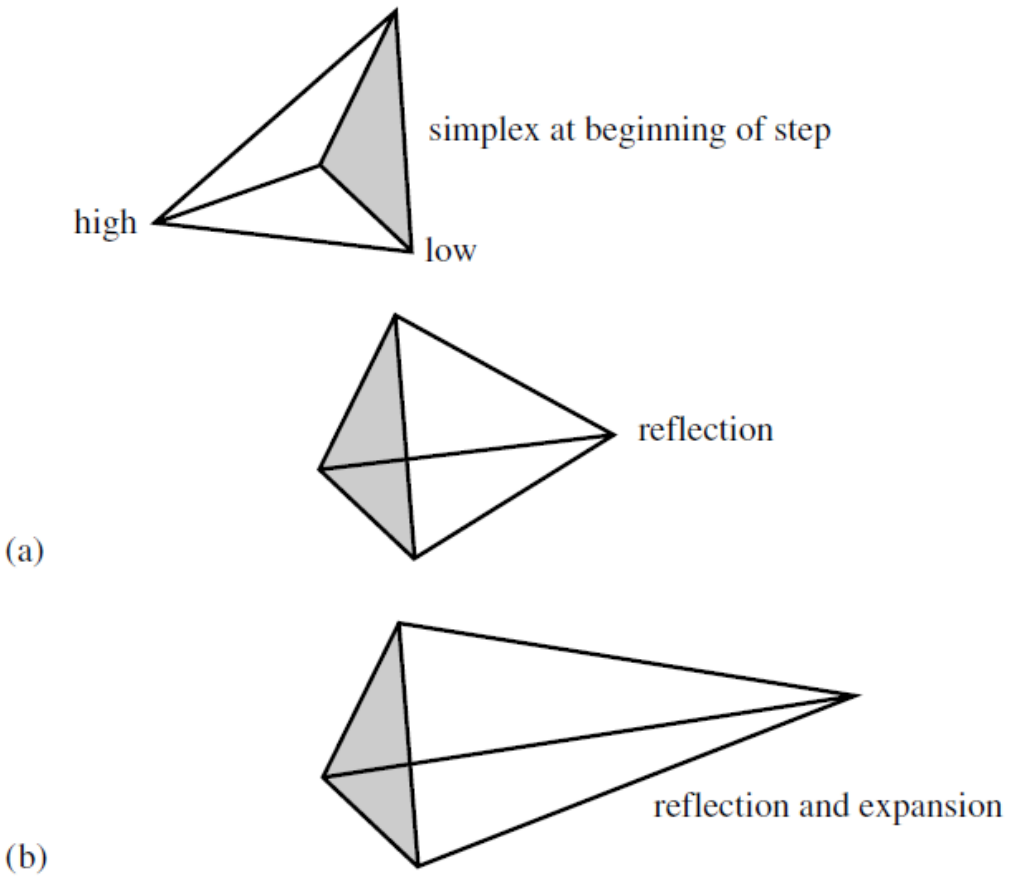
Метод Нелдера-Мида

Вход: Функция $f(\cdot)$, $p+1$ точек $x_1, ..., x_{p+1}$

Пока не выполнено условие останова:

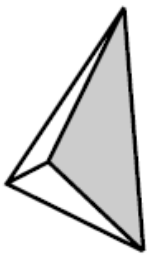
1. Отсортируем все точки по возрастанию целевой функции: $\{x_l, ..., x_g, x_h\}, \quad f(x_l) < ... < f(x_g) < f(x_h)$
2. Пробуем двигать точку x_h, выполняя отражение, растяжение или сжатие симплекса
3. Если не получилось уменьшить $f(x_h)$, то выполяем глобальное сжатие симплекса.

Метод Нелдера-Мида



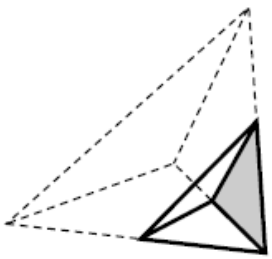
Метод Нелдера-Мида

(c)



contraction

(d)



multiple
contraction

Метод Нелдера-Мида

http://www.youtube.com/watch?v=HUqLxHfxWqU

Метод Нелдера-Мида

1. Простой в реализации

Метод Нелдера-Мида

1. Простой в реализации
2. Не работает с дискретными переменными
3. Сходимость зависит от начального приближения, может застрять в локальном минимуме
4. Желательно делать мультистарт
5. Плохо делает exploration
6. Обычно нужно много вычислений $f(\cdot)$

Ссылки

Метод Nelder-Mead

Функция fmin в scipy

http://docs.scipy.org/doc/scipy/reference/generated/scipy.optimize.fmin.html

Много методов для black-box оптимизации, в т.ч. метод Powell

http://docs.scipy.org/doc/scipy-0.14.0/reference/tutorial/optimize.html

HyperOpt (Random Search, TPE)

https://github.com/hyperopt/hyperopt

Байесовская оптимизация

Байесовская оптимизация строит распределение
$P(y|x, \textit{предыдущие измерения}, \textit{априорная информация})$

Байесовская оптимизация

Байесовская оптимизация строит распределение
$P(y|x, \textit{предыдущие измерения}, \textit{априорная информация})$

Гауссовский случайный процесс
Последовательность $\{y_t\}_{t \in T}$ называется гауссовским случайным процессом, если для любого конечного множества $\{t_1, t_2, ..., t_n\}$ случайные величины $\{y_{t_1}, y_{t_2}, ..., y_{t_n}\}$ имеют совместное многомерное нормальное распределение.

Байесовская оптимизация

Байесовская оптимизация строит распределение
$P(y|x, \textit{предыдущие измерения}, \textit{априорная информация})$

Гауссовский случайный процесс
Последовательность $\{y_t\}_{t \in T}$ называется гауссовским случайным процессом, если для любого конечного множества $\{t_1, t_2, ..., t_n\}$ случайные величины $\{y_{t_1}, y_{t_2}, ..., y_{t_n}\}$ имеют совместное многомерное нормальное распределение.

$$y_i = f(x_i)$$

$$[y_{t_1}, y_{t_2}, ..., y_{t_n}] \sim N(m, \Sigma)$$

Ковариация

Нужно иметь модель для матрицы ковариации:

$$\Sigma_{ij} = cov(y_i, y_j) = K(x_i, x_j)$$

Ковариация

Нужно иметь модель для матрицы ковариации:

$$\Sigma_{ij} = cov(y_i, y_j) = K(x_i, x_j)$$

$$r^2(x, x') = \sum_{d=1}^{p} \frac{(x_d - x'_d)^2}{\theta_d^2}$$

Ковариация

Нужно иметь модель для матрицы ковариации:

$$\Sigma_{ij} = cov(y_i, y_j) = K(x_i, x_j)$$

$$r^2(x, x') = \sum_{d=1}^{p} \frac{(x_d - x'_d)^2}{\theta_d^2}$$

Ядро squared exponential

$$K_{SE}(x, x') = \theta_0 \exp(-\frac{1}{2} r^2(x, x'))$$

Ковариация

Нужно иметь модель для матрицы ковариации:

$$\Sigma_{ij} = cov(y_i, y_j) = K(x_i, x_j)$$

$$r^2(x, x') = \sum_{d=1}^{p} \frac{(x_d - x'_d)^2}{\theta_d^2}$$

Ядро squared exponential

$$K_{SE}(x, x') = \theta_0 \exp(-\frac{1}{2} r^2(x, x'))$$

Ядро ARD Matern 5/2

$$K_{M52}(x, x') = \theta_0 \left(1 + \sqrt{5r^2(x, x')} + \frac{5}{3} r^2(x, x')\right) \exp(-\sqrt{5r^2(x, x')})$$

Оценка функции в новой точке

$(x_1, y_1), ..., (x_n, y_n)$ - уже проверенные точки, $\mathbf{y} \in \mathbb{R}^n$

x_{n+1} - новая точка

Оценка функции в новой точке

$(x_1, y_1), ..., (x_n, y_n)$ - уже проверенные точки, $\mathbf{y} \in \mathbb{R}^n$

x_{n+1} - новая точка

$$\begin{bmatrix} \mathbf{y} \\ y_{n+1} \end{bmatrix} \sim N(m, \Sigma)$$

Оценка функции в новой точке

$(x_1, y_1), ..., (x_n, y_n)$ - уже проверенные точки, $\mathbf{y} \in \mathbb{R}^n$
x_{n+1} - новая точка

$$\begin{bmatrix} \mathbf{y} \\ y_{n+1} \end{bmatrix} \sim N(m, \Sigma)$$

$$\Sigma = \begin{bmatrix} \mathbf{K} & \mathbf{k} \\ \mathbf{k}^T & k(x_{n+1}, x_{n+1}) \end{bmatrix}$$

$$K \in \mathbb{R}^{n \times n}, \quad K_{ij} = k(x_i, x_j)$$

$$\mathbf{k} = [k(x_{n+1}, x_1), k(x_{n+1}, x_2), ..., k(x_{n+1}, x_n)]$$

Оценка функции в новой точке

$(x_1, y_1), ..., (x_n, y_n)$ - уже проверенные точки, $\mathbf{y} \in \mathbb{R}^n$
x_{n+1} - новая точка

$$\begin{bmatrix} \mathbf{y} \\ y_{n+1} \end{bmatrix} \sim N(m, \Sigma)$$

$$\Sigma = \begin{bmatrix} \mathbf{K} & \mathbf{k} \\ \mathbf{k}^T & k(x_{n+1}, x_{n+1}) \end{bmatrix}$$

$$K \in \mathbb{R}^{n \times n}, \quad K_{ij} = k(x_i, x_j)$$

$$\mathbf{k} = [k(x_{n+1}, x_1), k(x_{n+1}, x_2), ..., k(x_{n+1}, x_n)]$$

$$P\left(\begin{bmatrix} y_{n+1} \\ \mathbf{y} \end{bmatrix}\right) = \frac{1}{\sqrt{(2\pi)^k |\Sigma|}} \exp\left(-\frac{\left(\begin{bmatrix} y_{n+1} \\ \mathbf{y} \end{bmatrix} - m\right)^T \Sigma^{-1} \left(\begin{bmatrix} y_{n+1} \\ \mathbf{y} \end{bmatrix} - m\right)}{2}\right)$$

Оценка функции в новой точке

$$P(y_{n+1}|x_{n+1}, \{x_i, y_i\}_{i=1}^{n}) = N(\mu, \sigma)$$

$$\mu = \mathbf{k}^T \mathbf{K}^{-1} \mathbf{y}$$

$$\sigma = k(x_{n+1}, x_{n+1}) - \mathbf{k}^T \mathbf{K}^{-1} \mathbf{k}$$

Оценка функции в новой точке

$$P(y_{n+1}|x_{n+1}, \{x_i, y_i\}_{i=1}^n) = N(\mu, \sigma)$$

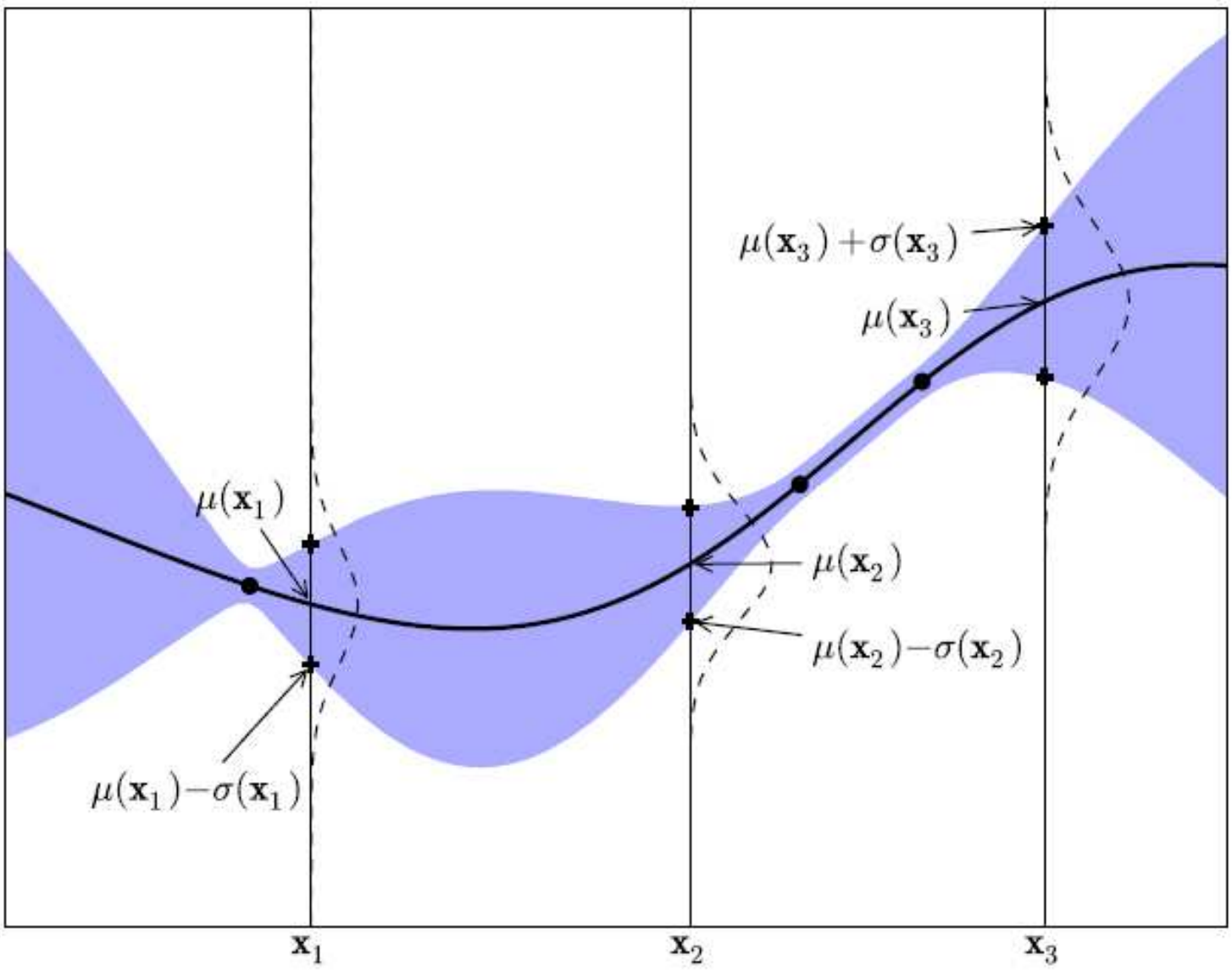
$$\mu = \mathbf{k}^T \mathbf{K}^{-1} \mathbf{y}$$

$$\sigma = k(x_{n+1}, x_{n+1}) - \mathbf{k}^T \mathbf{K}^{-1} \mathbf{k}$$

Будем использовать обозначения

$$\mu(x; \{x_i, y_i\}_{i=1}^n, \theta) \qquad \sigma^2(x; \{x_i, y_i\}_{i=1}^n, \theta)$$

Оценка функции в новой точке



Функция выгоды

Функция выгоды $a : X \to \mathbb{R}$

$$x_{next} = \underset{x}{\operatorname{argmax}}\, a(x)$$

Будем обозначать эту функцию $a(x; \{x_i, y_i\}_{i=1}^n, \theta)$ где θ - параметры алгоритма гауссовских процессов

Алгоритм Байесовской оптимизации

Алгоритм Байесовской оптимизации

Вход: функция $f(\cdot)$, функция выгоды $a(\cdot)$, макс. кол-во вычислений функции T, начальное множество значений $D = \{x_i, y_i\}_{i=1}^{k}$

1. for t = 1 ... T
2. $x_t \leftarrow \operatorname{argmax}_x a(x; D)$
3. Вычислить $y_t = f(x_t)$
4. $D \leftarrow D \cup \{x_t, y_t\}$

Функция выгоды

$$x_{best} = \underset{x \in \{x_1, \ldots, x_n\}}{\operatorname{argmin}} f(x)$$

$$\gamma(x) = \frac{f(x_{best}) - \mu(x; \{x_n, y_n\}, \theta)}{\sigma(x; \{x_n, y_n\}, \theta)}$$

$\Phi(\cdot)$ - кумулятивная функция нормального распределения

Функция выгоды

Варианты для функции выгоды $a(x; \{x_n, y_n\}_{i=1}^n, \theta)$:

Функция выгоды

Варианты для функции выгоды $a(x; \{x_n, y_n\}_{i=1}^n, \theta)$:

1. Вероятность улучшения:

$$a_{PI}(x; \{x_n, y_n\}_{i=1}^n, \theta) = P(y < f(x_{best})) = \Phi(\gamma(x))$$

Функция выгоды

Варианты для функции выгоды $a(x; \{x_n, y_n\}_{i=1}^n, \theta)$:

1. Вероятность улучшения:

$$a_{PI}(x; \{x_n, y_n\}_{i=1}^n, \theta) = P(y < f(x_{best})) = \Phi(\gamma(x))$$

2. Ожидаемое улучшение:

$$a_{EI}(x; \{x_n, y_n\}_{i=1}^n, \theta) = \mathbb{E}[(f(x_{best}) - y)_+]$$

Функция выгоды

Варианты для функции выгоды $a(x; \{x_n, y_n\}_{i=1}^n, \theta)$:

1. Вероятность улучшения:

$$a_{PI}(x; \{x_n, y_n\}_{i=1}^n, \theta) = P(y < f(x_{best})) = \Phi(\gamma(x))$$

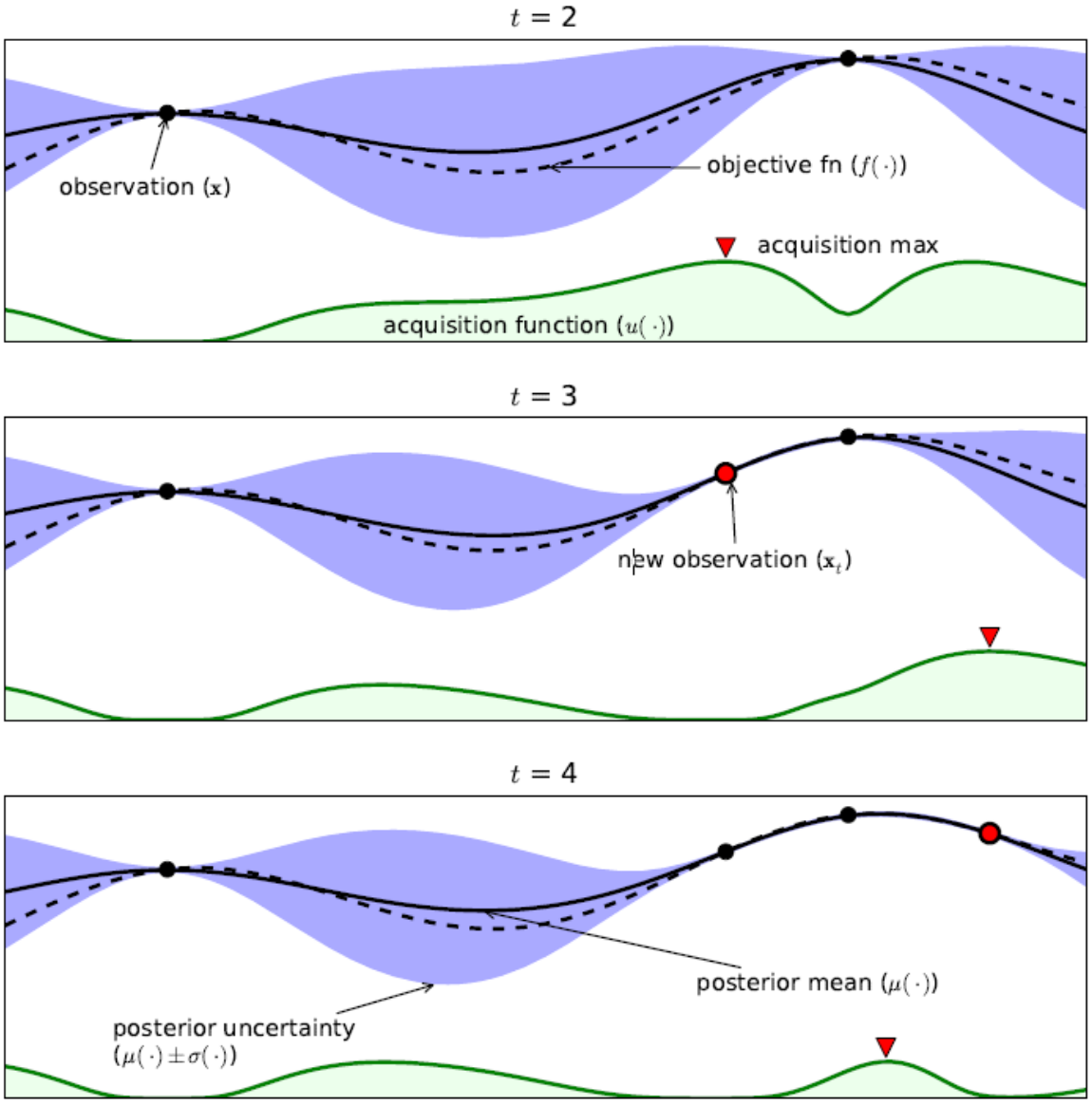
2. Ожидаемое улучшение:

$$a_{EI}(x; \{x_n, y_n\}_{i=1}^n, \theta) = \mathbb{E}[(f(x_{best}) - y)_+]$$

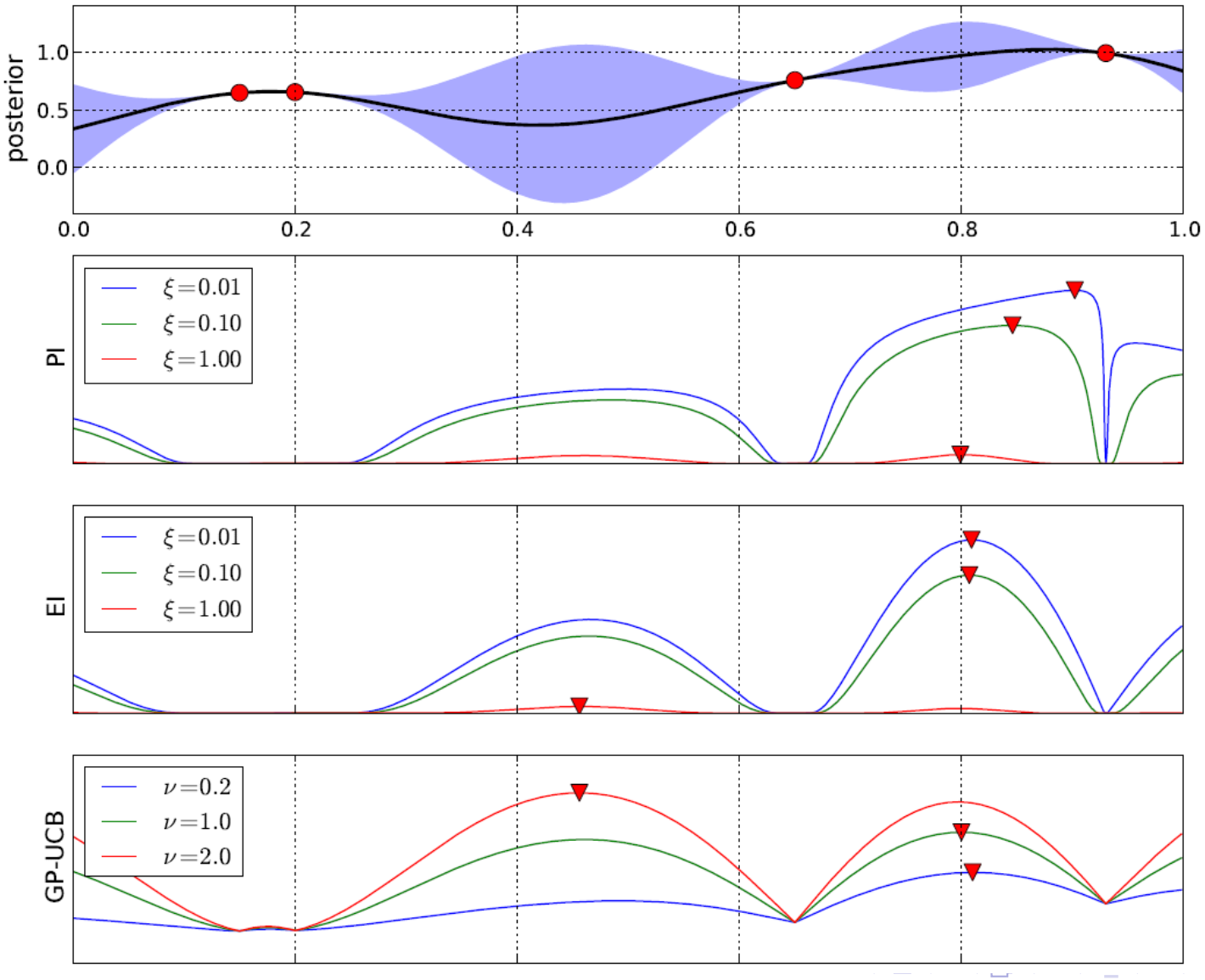
3. Нижний доверительный интервал:

$$a_{LCB} = f(x_{best}) - (\mu(x; \{x_n, y_n\}_{i=1}^n, \theta) - \kappa\sigma(x; \{x_n, y_n\}_{i=1}^n, \theta))$$

Эксперименты



Эксперименты



Шум

Если значения функции измеряются не точно

$$y_i = f(x_i) + \xi_i$$

$$\xi_i \sim N(0, \nu)$$

Шум

Если значения функции измеряются не точно

$$y_i = f(x_i) + \xi_i$$

$$\xi_i \sim N(0, \nu)$$

то ковариационная матрица модифицируется

$$cov(y_i, y_i) = cov(f(x_i), f(x_j)) + \nu[i = j] = K_{ij} + \nu[i = j]$$

$$\Sigma \leftarrow \Sigma + \nu I$$

Шум

Если значения функции измеряются не точно

$$y_i = f(x_i) + \xi_i$$

$$\xi_i \sim N(0, \nu)$$

то ковариационная матрица модифицируется

$$cov(y_i, y_i) = cov(f(x_i), f(x_j)) + \nu[i = j] = K_{ij} + \nu[i = j]$$

$$\Sigma \leftarrow \Sigma + \nu I$$

$$\mu = \mathbf{k}^T (\mathbf{K} + \nu I)^{-1} \mathbf{y}, \quad \sigma = k(x_{n+1}, x_{n+1}) - \mathbf{k}^T (\mathbf{K} + \nu I)^{-1} \mathbf{k}$$

Шум

Если значения функции измеряются не точно

$$y_i = f(x_i) + \xi_i$$

$$\xi_i \sim N(0, \nu)$$

то ковариационная матрица модифицируется

$$cov(y_i, y_i) = cov(f(x_i), f(x_j)) + \nu[i = j] = K_{ij} + \nu[i = j]$$

$$\Sigma \leftarrow \Sigma + \nu I$$

$$\mu = \mathbf{k}^T (\mathbf{K} + \nu I)^{-1} \mathbf{y}, \quad \sigma = k(x_{n+1}, x_{n+1}) - \mathbf{k}^T (\mathbf{K} + \nu I)^{-1} \mathbf{k}$$

В качестве лучшего найденого значения функции $f(\cdot)$ рекомендуется использовать

$$f(x_{best}) \leftarrow \underset{x \in \{x_1, \ldots, x_n\}}{\operatorname{argmin}} \mu(x; \{x_i, y_i\}_{i=1}^n, \theta)$$

Модель с шумом помогает в случае не непрерывных $f(\cdot)$

Проблемы с Байесовской оптимизацией

У алгоритма поиска гиперпараметров тоже есть параметры :(

$$\theta_{1\ldots D},\ \theta_0,\ \nu,\ m$$

Закмнутый круг?

Проблемы с Байесовской оптимизацией

У алгоритма поиска гиперпараметров тоже есть параметры :(

$$\theta_{1...D},\ \theta_0,\ \nu,\ m$$

Закмнутый круг?

Варианты решения проблемы:

Проблемы с Байесовской оптимизацией

У алгоритма поиска гиперпараметров тоже есть параметры :(

$$\theta_{1...D},\ \theta_0,\ \nu,\ m$$

Закмнутый круг?

Варианты решения проблемы:

1. Найти параметры максимизацией правдободобия

$$p(y_1, ..., y_N | x_1, ..., x_N, \theta, \nu, m) = N(y | m, \Sigma_\theta + \nu I)$$

Проблемы с Байесовской оптимизацией

У алгоритма поиска гиперпараметров тоже есть параметры :(

$$\theta_{1...D},\ \theta_0,\ \nu,\ m$$

Закмнутый круг?

Варианты решения проблемы:

1. Найти параметры максимизацией правдободобия

$$p(y_1, ..., y_N | x_1, ..., x_N, \theta, \nu, m) = N(y | m, \Sigma_\theta + \nu I)$$

2. Наложить на θ априорное распределение и проинтегрировать по апостериорному распределению θ

$$\hat{a}(x; \{x_n, y_n\}_{i=1}^n) = \int a(x; \{x_n, y_n\}_{i=1}^n, \theta) p(\theta | \{x_n, y_n\}_{i=1}^n) d\theta$$

Вычисляется методом Markov Chain Monte Carlo (MCMC).

Проблемы с Байесовской оптимизацией

Поиск

$$x_{next} = \operatorname*{argmax}_x a(x)$$

само по себе является проблемой black-box оптимизации. Байесовскую оптимизацию имеет смысл использовать, если нахождение x_{next} занимает гораздо меньше времени, чем вычисление $f(x_{next})$

Параллельные вычисления

Пусть у нас есть возможность тестировать наборы гиперпараметров параллельно.
Ситуация: в точках $\{x_1, ..., x_n\}$ вычисления произведены, в точках $\{x_{n+1}, ..., x_{n+J}\}$ вычисления не закончены. Какую точку выбрать следующей?

Параллельные вычисления

Пусть у нас есть возможность тестировать наборы гиперпараметров параллельно.
Ситуация: в точках $\{x_1, ..., x_n\}$ вычисления произведены, в точках $\{x_{n+1}, ..., x_{n+J}\}$ вычисления не закончены. Какую точку выбрать следующей?
Оценим распределение $y_{n+1}, ..., y_{n+J}$ в точках $\{x_{n+1}, ..., x_{n+J}\}$ с помощью гауссовских процессов

$$p(y_{n+1}, ..., y_{n+J}) = p(y_{n+1}, ..., y_{n+J} | x_{n+1}, ..., x_{n+J}, \{x_n, y_n\}_{i=1}^n)$$

Параллельные вычисления

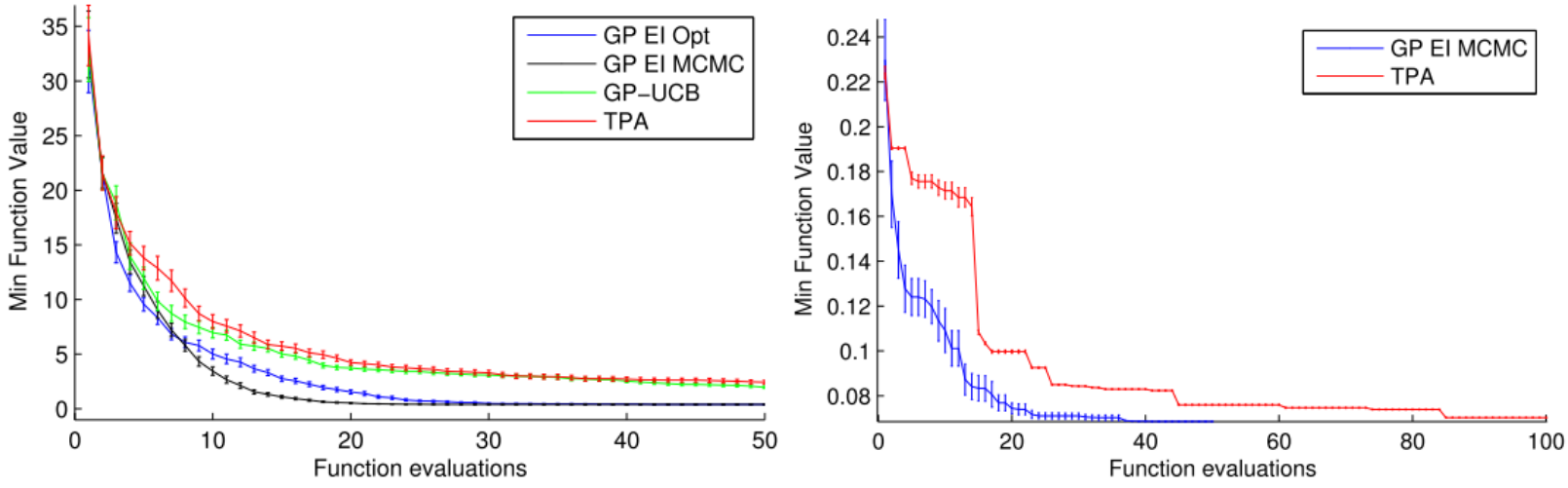
Пусть у нас есть возможность тестировать наборы гиперпараметров параллельно.
Ситуация: в точках $\{x_1, ..., x_n\}$ вычисления произведены, в точках $\{x_{n+1}, ..., x_{n+J}\}$ вычисления не закончены. Какую точку выбрать следующей?
Оценим распределение $y_{n+1}, ..., y_{n+J}$ в точках $\{x_{n+1}, ..., x_{n+J}\}$ с помощью гауссовских процессов

$$p(y_{n+1}, ..., y_{n+J}) = p(y_{n+1}, ..., y_{n+J} | x_{n+1}, ..., x_{n+J}, \{x_n, y_n\}_{i=1}^n)$$

$$\hat{a}(x; \{x_n, y_n\}_{i=1}^{n+J}; \theta, \{x_j\}) =$$

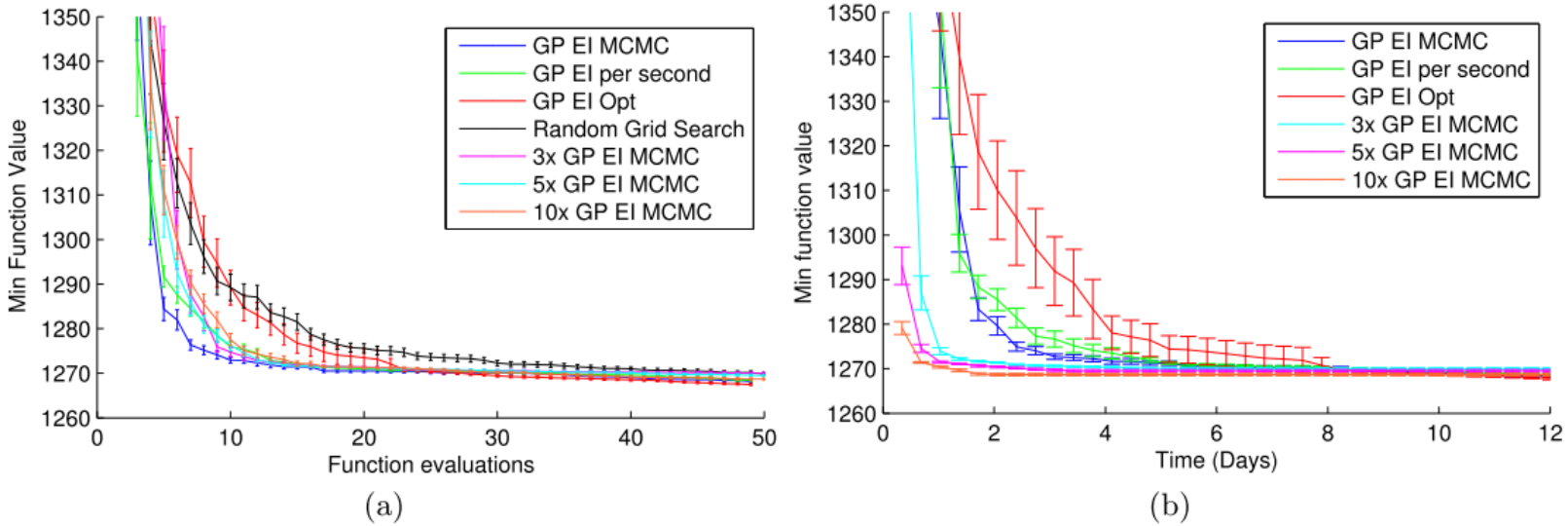
$$= \int_{\mathbb{R}^J} a(x; \{x_n, y_n\}_{i=1}^{n+J}, \theta) p(y_{n+1}, ..., y_{n+J}) dy_{n+1} .. dy_{n+J}$$

Эксперименты



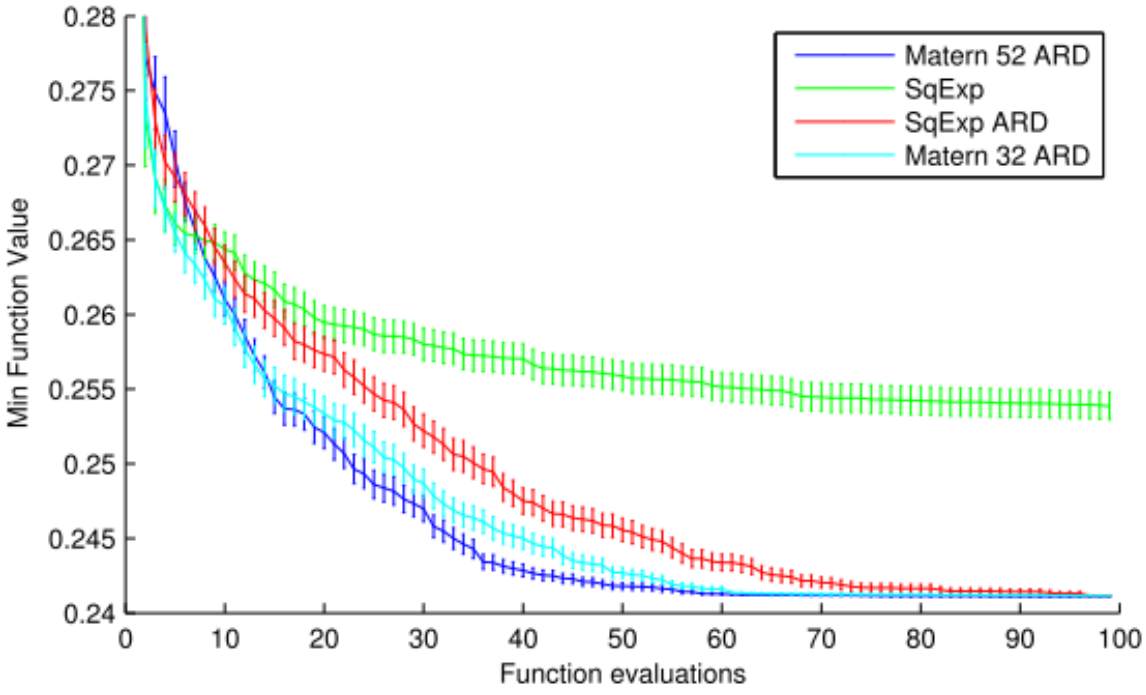
Слева: оптимизация функции Branin-Hoo.
Справа: обучение логистической регресии на MNIST, изменялись: шаг стох. градиента, размер minibatch, кол-во эпох, параметр L2-регуляризациии.

Эксперименты



Online LDA, изменялись параметры τ_0, κ в шаге стохастического градиента $\rho_t = (\tau_0 + t)^{-\kappa}$ а также размер minibatch. Количество тем было зафиксировано и равно 100.

Эксперименты



Сравнение различных ядер для задачи motif finding task с помощью Latent Structured Support Vector Machines.

Вычислительные эксперименты

Задача: нужно настроить параметры сервиса (поиск, реклама, музыка ...) для оптимизации целевой метрики.

Вычислительные эксперименты

Задача: нужно настроить параметры сервиса (поиск, реклама, музыка ...) для оптимизации целевой метрики.
Пример: критерий отбора рекламных объявлений $P(click)^{\alpha} Bid > A$.
Параметры α, A подбираются исходя из максимизации прибыли $R(\alpha, A)$.

Вычислительные эксперименты

Задача: нужно настроить параметры сервиса (поиск, реклама, музыка ...) для оптимизации целевой метрики.
Пример: критерий отбора рекламных объявлений $P(click)^{\alpha} Bid > A$.
Параметры α, A подбираются исходя из максимизации прибыли $R(\alpha, A)$.
Вычисление $R(\alpha, A)$ - это эксперимент на доле пользователей в течение нескольких дней. Результат будет шумным, часто статистически не значимым.

Вычислительные эксперименты

Задача: нужно настроить параметры сервиса (поиск, реклама, музыка ...) для оптимизации целевой метрики.
Пример: критерий отбора рекламных объявлений $P(click)^{\alpha} Bid > A$.
Параметры α, A подбираются исходя из максимизации прибыли $R(\alpha, A)$.
Вычисление $R(\alpha, A)$ - это эксперимент на доле пользователей в течение нескольких дней. Результат будет шумным, часто статистически не значимым. Можно попробовать Баейсовскую оптимизацию.

Байесовская оптимизация

1. Более сложная в реализации
2. Работает с дискретными переменными
3. Сходимость не сильно зависит от начального приближения
4. Хорошо делает exploration
5. Обычно нужно немного вычислений $f(\cdot)$

Ссылки

Spearmint ($\star$)
`https://github.com/JasperSnoek/spearmint`

Metric Optimization Engine
`https://github.com/Yelp/MOE`

Bayesian Optimization ($\star$)
`https://github.com/fmfn/BayesianOptimization`

($\star$) - «хорошие»

Метод Кригинга в геологии можно рассматривать как специфичный случай Байесовской оптимизации

Список литературы

tutorial E. Brochu, V. M. Cora and N. de Freitas. A Tutorial on Bayesian Optimization of Expensive Cost Functions, with Application to Active User Modeling and Hierarchical Reinforcement Learning, 2010. http://arxiv.org/abs/1012.2599

practical By Jasper Snoek, Hugo Larochelle and Ryan P. Adams University. Practical Bayesian Optimization of Machine Learning Algorithms, 2012. http://arxiv.org/abs/1206.2944

simplex Press, WH; Teukolsky, SA; Vetterling, WT; Flannery, BP (2007). Numerical Recipes: The Art of Scientific Computing (3rd ed.). New York: Cambridge University Press.

NN G. Hinton. Bayesian Optimization of Hyperparameters. http://www.youtube.com/watch?v=cWQDeB9WqvU