

Московский государственный университет имени М. В. Ломоносова



Факультет вычислительной математики и кибернетики

Кафедра математических методов прогнозирования

Медведев Алексей Владимирович

**Автоматический подбор
параметров алгоритмов машинного обучения
с помощью обучения с подкреплением
ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА**

Научный руководитель:

д.ф.-м.н., профессор

А. Г. Дьяконов

Москва, 2019

Содержание

1	Введение	3
2	Постановка задачи	3
2.1	Гауссовские процессы	4
2.2	Байесовская оптимизация	4
2.3	Многорукие бандиты	5
2.4	Трансферное обучение в байесовской оптимизации	5
3	Существующие подходы к трансферному обучению в байесовской оптимизации	6
4	Предложенный метод	8
5	Эксперименты	9
5.1	Случай одного близкого источника	11
5.2	Случай двух близких источников	13
5.3	Случай отсутствия близких источников	15
5.4	Обсуждение и выводы	16
6	Заключение	17
	Список литературы	18

Аннотация

Автоматический поиск оптимальных гиперпараметров для алгоритмов машинного обучения может предоставить множество преимуществ. Но такой поиск очень ресурсоемок, так как для оценки каждой конфигурации необходимо полностью обучать алгоритм заново.

В данной выпускной квалификационной работе предложен метод, основанный на трансферном обучении, позволяющий ускорить процедуру подбора оптимальной конфигурации для алгоритмов машинного обучения. Экспериментально показана эффективность предложенного подхода и устойчивость работы даже в условиях отсутствия подходящего источника для трансферного обучения.

1 Введение

Почти каждый алгоритм машинного обучения имеет набор гиперпараметров. Зачастую они довольно сильно влияют на качество работы этих алгоритмов. Автоматический подбор гиперпараметров может:

- уменьшить человеческие усилия, необходимые для применения алгоритмов машинного обучения;
- улучшить качество применяемых алгоритмов;
- внести вклад в увеличение уровня объективности научных работ. Автоматический подбор гиперпараметров предполагает честное сравнение различных алгоритмов, так как внимание к настройке алгоритмов будет сравнимо.

Однако существует ряд проблем, которые делают задачу автоматического подбора гиперпараметров довольно сложной:

- Оценка качества того или иного набора гиперпараметров может быть очень затратной, так как связана с обучением модели заново. Это особенно критично для задач связанных с большими объемами данных и сложными моделями.
- Зачастую не существует явного выражения зависимости качества работы алгоритма от его гиперпараметров. Поэтому доступа к градиентам функции потерь по гиперпараметрам у нас нет.

2 Постановка задачи

Популярным подходом к решению задачи автоматического подбора гиперпараметров является байесовская оптимизация с гауссовским процессом в качестве априорного распределения. Такой подход не требует поиска градиентов функции потерь по набору гиперпараметров.

Чтобы ускорить дорогостоящий поиск оптимальных для решения задачи параметров алгоритма, можно использовать информацию о том как алгоритм решал другие задачи, имея тот или иной набор гиперпараметров. Такой метод получил распространение под названием трансферное обучение.

2.1 Гауссовские процессы

Гауссовские процессы [1] популярный выбор для априорных распределений над функциями. На их основе можно строить регрессоры и приближать функции. Такие регрессоры полностью определены функцией среднего (μ) и функцией ковариации (k):

$$f(x) \sim GP(\mu(x), k(x, x'))$$

Предположим, что известны значения искомой функции f в t точках, обозначим $D_t = \{x_{1:t}, f_{1:t}\}$. Результатом применения такого регрессора будут матожидание, $\mu(x_{t+1}) = k^T K^{-1} f_{1:t}$ и дисперсия $\sigma^2(x_{t+1}) = k(x_{t+1}, x_{t+1}) - k^T K^{-1} k$, апостериорного распределения на значение функции в новой точке x_{t+1} , где $k = [k(x_{t+1}, x_1), k(x_{t+1}, x_2), \dots, k(x_{t+1}, x_t)]$ и $K_{i,j} = k(x_i, x_j)$.

Часто делают предположение о зашумленности наблюдаемых значений функции f , $y_t = f(x_t) + \epsilon$, где $\epsilon \sim \mathcal{N}(0, \sigma_n^2)$. Тогда $K(i, i) = k(x_i, x_i) + \sigma_n^2$.

2.2 Байесовская оптимизация

Байесовская оптимизация эффективный метод для поиска глобального оптимума черного ящика. Черным ящиком называют функцию, внутреннее устройство которой нам неизвестно, но её значение может быть оценено в произвольных точках. Данный подход к решению этой проблемы состоит из двух компонент.

Первая компонента – построение вероятностной модели черного ящика, введением априорного распределения на его значения. Часто таким априорным распределением выступает гауссовский процесс, но существует и множество других аналогов, нейронные сети [3], случайные леса [2].

Вторая компонента отвечает за выбор следующей точки, в которой должно быть оценено значение черного ящика. Выбор точки ведется путем оптимизации функции сборщика (acquisition function), которая комбинирует в себе текущее приближение черного ящика и неуверенность алгоритма в этом приближении. Баланс этих составляющих очень важен. Приходится решать что важнее: искать новые точки вблизи оптимумов текущего приближения или среди точек, о которых пока ничего не известно. Разные комбинации предсказанного среднего $\mu(x)$ и дисперсии $\sigma(x)^2$ созда-

ют разные функции сборщики, например верхняя граница уверенности [5] (gaussian process upper confidence bound), ожидаемое улучшение [4] (expected improvement).

2.3 Многорукие бандиты

Задача о многоруких бандитах [6] – это задача о последовательном принятии решений, в которой агент на каждой итерации $t = 1, \dots, T$ должен взаимодействовать со средой одним из M способов (рук). Соревновательная постановка задачи о бандитах подразумевает, что среда и агент ведут антагонистическую игру. Таким образом исключаются предположения о случайности вознаграждений. В такой постановке случайность присутствует в выборе действий. На каждой итерации стратегия агента – это распределение над его действиями, обусловленное историей игры: последовательностью действий и вознаграждений.

Оценку действий агента дает сожаление (regret):

$$R = \mathbb{E} \left[\max_i \sum_{t=1}^T r_{t,i} - r_{t,A_t} \right]$$

Где $r_{t,i}$ – вознаграждение за действие номер i на итерации t , A_t – случайное, в соответствии с определенной на итерации t стратегией, действие агента. Такой функционал оценивает потерю в итоговом вознаграждении агента сравнительно с наилучшим действием за время игры. Существуют алгоритмы [7], [6], для которых сожаление растет с количеством итераций сублинейно, а именно как $O(\sqrt{T})$. Но такие алгоритмы существенно отличаются по дисперсии случайного сожаления. Так же некоторые алгоритмы позволяют определить какой вид будет иметь верхняя оценка на значение случайного сожаления с заданной вероятностью:

$$\hat{R} = \max_i \sum_{t=1}^T r_{t,i} - r_{t,A_t}$$

2.4 Трансферное обучение в байесовской оптимизации

Чтобы найти область, в которой черный ящик принимает оптимальные значения, может потребоваться достаточно большое количество итераций. Особенно если изначально у нас нет почти никакой информации о нем.

Трансферное обучение может стать решением проблемы «холодного старта». Идея таких методов состоит в использовании знаний о решении задачи оптимизации одной функции (источника), для решения аналогичной проблемы с уже интересующей нас целевой функцией. Проблемой может стать источник, который значительно отличается от целевой функции. Тогда процесс поиска оптимального значения наоборот только затянется. Поэтому ключевую роль играет связь функции источника и целевой функции.

3 Существующие подходы к трансферному обучению в байесовской оптимизации

В работе [8] строится общее для функций источников и целевой функции оптимизационное пространство. Для этого к вектору гиперпараметров добавляются различные мета-признаки, описывающие специфики задач источников. Признаками могут быть количество классов в задаче или размер датасета. Роль оптимизируемой функции играет отклонение функции метрики качества набора гиперпараметров для каждого из датасетов от среднего значения, масштабированные среднеквадратичным отклонением. Средние значения и среднеквадратичные отклонения считаются отдельно для каждой задачи и датасета.

Указанный выше подход выбирает релевантный источник, опираясь на мета-признаки, это может стать источником ошибок. В работе [9] польза источника для трансфера определяется во время процедуры байесовской оптимизации. В предложенном методе, известные оценки значений функции-источника считаются шумными оценками значений целевой функции. Поэтому авторы объединяют данные об оценках функции источника и данные об оценках целевой функции. Для этого объединения запускается процесс байесовской оптимизации. Ковариационная матрица в таком случае выглядит так:

$$\hat{K} = \begin{pmatrix} k(x_1^s, x_1^s) & \dots & k(x_1^s, x_{N_s}^s) & k(x_1^s, x_1) & \dots & k(x_1^s, x_{N_t}) \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ k(x_{N_s}^s, x_1^s) & \dots & k(x_{N_s}^s, x_{N_s}^s) & k(x_{N_s}^s, x_1) & \dots & k(x_{N_s}^s, x_{N_t}) \\ k(x_1, x_1^s) & \dots & k(x_1, x_{N_s}^s) & k(x_1, x_1) & \dots & k(x_1, x_{N_t}) \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ k(x_{N_t}, x_1^s) & \dots & k(x_{N_t}, x_{N_s}^s) & k(x_{N_t}, x_1) & \dots & k(x_{N_t}, x_{N_t}) \end{pmatrix}$$

$$K = \hat{K} + \begin{pmatrix} \sigma_s^2 \mathbb{I}_{N_s \times N_s} & 0 \\ 0 & \sigma_t^2 \mathbb{I}_{N_t \times N_t} \end{pmatrix}$$

Где x_i^s точки, в которых известна оценка функции источника, x_i точки, в которых известна оценка целевой функции. Параметр σ_t – предполагаемый уровень шума в наблюдаемых оценках целевой функции, а σ_s – настраиваемая компонента связанности целевой функции и функции источника трансферного обучения. Вероятностная модель для последнего параметра строится введением априорного обратного гамма распределения с параметрами a_0 , b_0 . Параметры апостериорного распределения обновляются согласно следующим формулам, полученным полным байесовским выводом:

$$b_n = b_0 + \frac{\sum_{i=1}^{N_t} (f(x_i) - f^s(x_i))^2}{2}$$

$$a_n = a_0 + \frac{n}{2}$$

Где n номер итерации байесовской оптимизации, $f(x_i)$ и $f^s(x_i)$ значения целевой функции и источника в точке x_i .

В работе [10] авторы предлагают способ активно выбирать источник для запуска вышеописанного алгоритма [9]. Метод предполагает решение задачи о многоруких бандитах с использованием алгоритма EXP3 [6]. В такой постановке каждая функция источник представляется действием, которое может выбрать агент. В качестве функции вознаграждения предлагается использовать $-(f(x_t) - \mu^s(x_t))^2$.

4 Предложенный метод

Использование алгоритма EXP3 гарантирует, что математическое ожидание сожаления будет расти сублинейно. Но гарантий того, в какие границы будет укладываться сожаление при случайном запуске, алгоритм не дает. Можно показать [11], что дисперсия случайного сожаления при определенных условиях может быть $\mathbb{V}[\hat{R}] = \Omega(T^2)$. Где $\Omega(g(n)) = \liminf_{n \rightarrow \infty} \frac{f(n)}{g(n)} > 0$, T число итераций. Одно из условий такого асимптотического ограничения снизу значения дисперсии – существование последовательности действий агента, таких что $P\left(\hat{R} > \frac{T}{4}\right) > \frac{1}{65}$.

То есть при достаточно большой доле нерелевантных функций источников, дисперсия случайного сожаления ограничена снизу квадратичной функцией от числа итераций алгоритма. Это делает сценарий применения такого метода малопривлекательным, если мы знаем наверняка, что доля релевантных источников для трансферного обучения в нашей задаче невысока. Метод попросту будет работать не лучше чем обычная байесовская оптимизация без трансферного обучения, но вычислительно гораздо затратнее.

Возможным решением может стать замена алгоритма EXP3 на EXP3-IX [7] (implicit exploration). Использование нового алгоритма также гарантирует сублинейный рост сожаления с ростом числа итераций. Но в отличие от EXP3 для EXP3-IX можно гарантировать, что случайное сожаление с произвольной вероятностью $1 - \delta$ будет ограничено сверху:

$$\hat{R} \leq 2\sqrt{2MT \log M} + \left(\sqrt{\frac{2MT}{\log M}} + 1 \right) \log \frac{2}{\delta}$$

Можно показать [11], что в таком случае $\mathbb{V}[\hat{R}] = O(T)$.

После выбора источника для трансферного обучения, запускается алгоритм **EnvGP** [9]. Стоит отметить, что для каждого источника параметры распределения над σ_s пересчитываются отдельно.

Как уже говорилось выше алгоритм байесовской оптимизации состоит из двух компонент: вероятностной модели черного ящика и функции сборщика. Довольно популярным выбором является верхняя граница уверенности [5] (gaussian process upper confidence bound):

$$x_t = \arg \max_{x \in D} \mu_{t-1}(x) + \beta_t^{\frac{1}{2}} \sigma_{t-1}(x)$$

В работе [5] показано, что для конечного D при выборе $\beta_t = 2 \log(|D| t^2 \pi^2 / 6\delta)$ гарантирует ограниченность сожаления $\hat{R} = \sum_{t=1}^T f(x^*) - f(x_t)$. Для любой δ с вероятностью не меньше $1 - \delta$:

$$\hat{R} \leq 2\sqrt{2\beta_T \gamma_T T / \log(1 + \delta^{-2})}$$

Где γ_T – максимальное, по выбору точек для оценки целевой функции, приращение количества информации о черном ящике. Однако неограниченный рост β кажется не лучшим решением, особенно когда значения становятся близкими к оптимальному. Поэтому предлагается новый метод выбора коэффициента β_t с помощью обучения с подкреплением.

Предположение заключается в том, что можно представить эту задачу как задачу о многоруком бандите. Где каждое действие агента – это выбор между исследованием целевой функции в областях, о которых алгоритму еще ничего неизвестно, и поиском оптимальных значений вблизи текущих оптимумов. Вместо традиционных алгоритмов предлагается использовать полносвязную нейронную сеть с малым количеством весов. К выходам нейронной сети применяется *Softmax*, так что модель предсказывает распределение над β_t . Изначально модель моделирует равномерное распределение. Но на каждой итерации происходит корректировка весов модели, алгоритмом REINFORCE [12]. Агент выбирает из конечного числа β сэмплированием из полученного категориального распределения. Затем вычисляется награда $r_t = -(f(x^*) - f(x_t))^2$, и происходит обновление модели:

$$\theta_{t+1} = \theta_t + lr \nabla_{\theta_t} \log(\pi_{\theta_t})(r_t - b)$$

Где π_{θ_t} – распределение, полученное нашей моделью, b – бейзлайн, он необходим для уменьшения дисперсии оценки градиента.

5 Эксперименты

Для исследования предлагается несколько сценариев: случай одного, двух близких источников из четырех доступных и отсутствия таковых. В каждом из сценариев роли целевой функции и функций источников играют двумерные нормированные на $[0, 1]$ суммы двух Гауссиан с ковариационными матрицами I и $0.25I$. В частности

целевая функция остается неизменной для любого сценария и вершины её гауссиан расположены в точках: $[0.7, 0.7]$ и $[2.7, 2.7]$.

Сравниваются следующие алгоритмы:

- обычная байесовская оптимизация, без трансферного обучения (simple bo);
- метод, описанный в [8] (smbo transfer);
- метод , описанный в [10] (exp3-auer);
- предложенный в этой работе метод, но без модификации функции сборщика (exp3-IX);
- предложенный в этой работе метод (exp3-IX active theta);

В экспериментах все алгоритмы используют гауссовские процессы с экспоненциальной ковариационной функцией. В качестве бейзлайна было выбрано скользящее экспоненциальное среднее по наградам предыдущих итераций с коэффициентом сглаживания 0.8. В качестве оптимизатора модели подбора β_t выбран Adam с длиной градиентного шага 0.005. Модель выбирает β_t из: 0.01, 0.1, 1.0, 10. Максимизация функции сборщика идет по сетке из 30 значений для каждой координаты из отрезка $[-4, 5]$, то есть всего 900 вариантов. Дополнительно запускается метод градиентного подъема из пяти произвольных точек. Для каждого источника известны значения в 100 произвольных точках из $[-4, 5] \times [-4, 5]$.

Алгоритмы, основанные на использовании бандитов, на каждой итерации выбирают один из пяти источников. Четыре из которых это действительно какие-то функции, а пятый это пустое действие, при выборе которого алгоритм проводит итерацию обычной байесовской оптимизации без трансферного обучения.

5.1 Случай одного близкого источника

Для моделирования такого сценария мы генерируем 4 Гауссианы с первыми вершинами в $[-5, -5]$, $[0.8, 0.8]$, $[5, 5]$, $[6, 6]$ и вторыми вершинами в $[-3, -3]$, $[2.8, 2.8]$, $[3, 3]$, $[4, 4]$. На следующем рисунке изображены взаимные расположения источников и целевой функции.

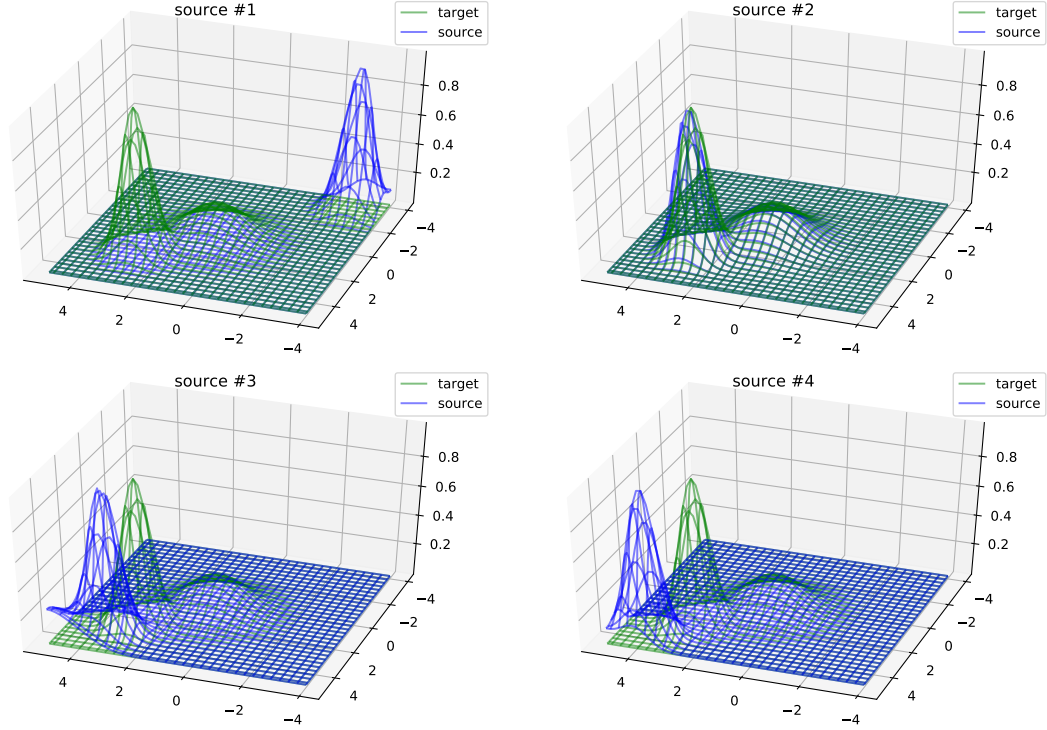


Рис. 1: Визуализация полученных функций. Можно заметить, что второй источник является единственным близким к целевой функции.

Для большей объективности каждый алгоритм запускается 10 раз, а затем результат усредняется. Каждый из сравниваемых алгоритмов работает на протяжении 30 итераций. На следующем графике отображены результаты алгоритмов.

Из графика видно, что лучше всего справился новый метод с подбором β . В то же время алгоритмы использующие бандитов EXP3 и EXP3-IX показали почти одинаковый результат. Так же можно заметить, что трансферное обучение заметно влияет на ход байесовской оптимизации.

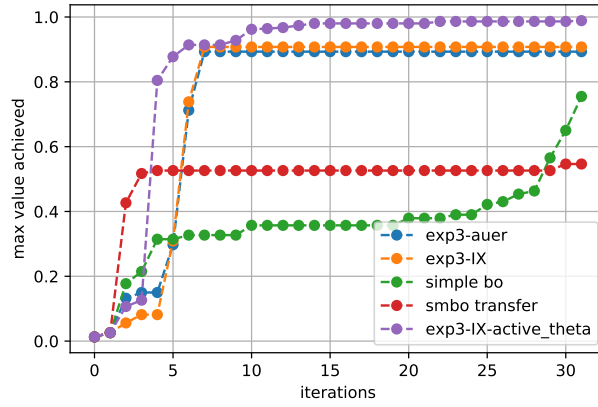


Рис. 2: Максимальное значение найденное к наступлению определенной итерации.

На следующем графике показано распределение выбора агентами действий для двух алгоритмов EXP3 и EXP3-IX, суммарно по всем 10 запускам.

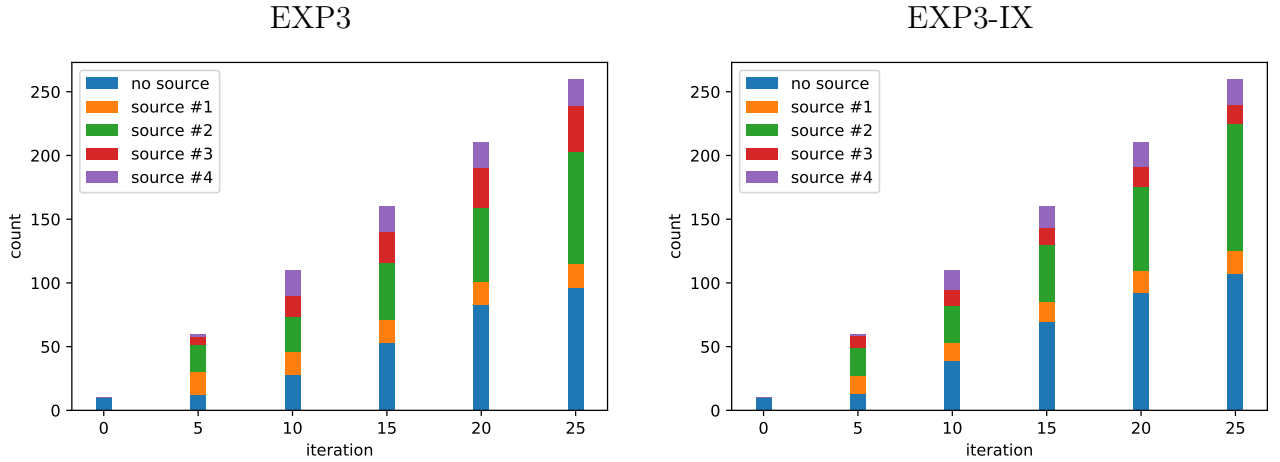


Таблица 1: Распределение выбора агентами действий, суммарно по всем 10 запускам.

Можно заметить, что распределения похожи, но алгоритм с бандитом EXP3-IX меньше внимания уделяет источникам 3,4,5. Оба алгоритма чаще всего выбирают правильное действие, а ближе к 10-й итерации все чаще начинают выбирать отсутствие трансфера. Это можно объяснить тем, что про целевую функцию накоплено уже достаточно информации, чтобы обычная байесовская оптимизация оказалась эффективнее даже близкого источника.

5.2 Случай двух близких источников

Теперь вершины располагаются в точках $[-5, -5]$, $[0.8, 0.8]$, $[0.4, 0.4]$, $[6, 6]$ и $[-3, -3]$, $[2.8, 2.8]$, $[2.4, 2.4]$, $[4, 4]$ соответственно. В таком случае взаимное расположение источников и целевой функции будет выглядеть следующим образом.

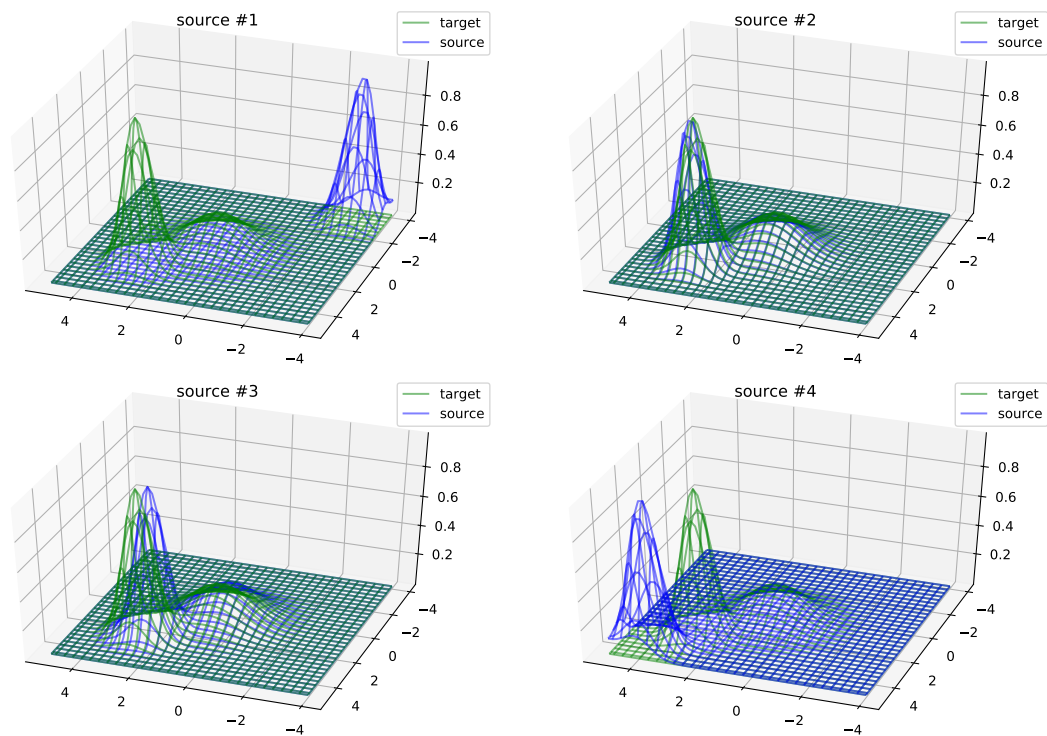


Рис. 3: Визуализация полученных функций. Можно заметить, что второй и третий источники являются самыми близкими к целевой функции.

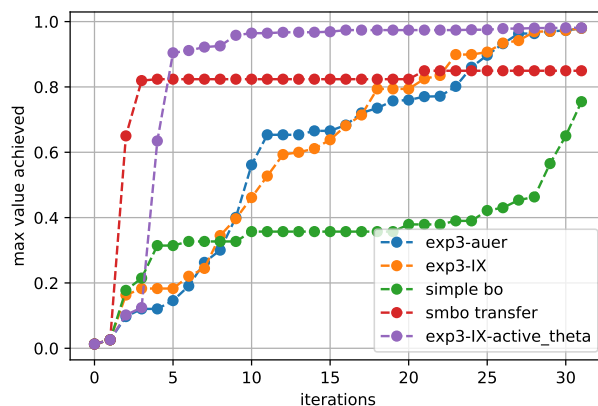


Рис. 4: Максимальное значение найденное к наступлению определенной итерации.

Результат схож с полученным ранее. Отличие между алгоритмами EXP3 и EXP3-IX почти нет. В этот раз уже три алгоритма находят близкие к оптимальным точки.

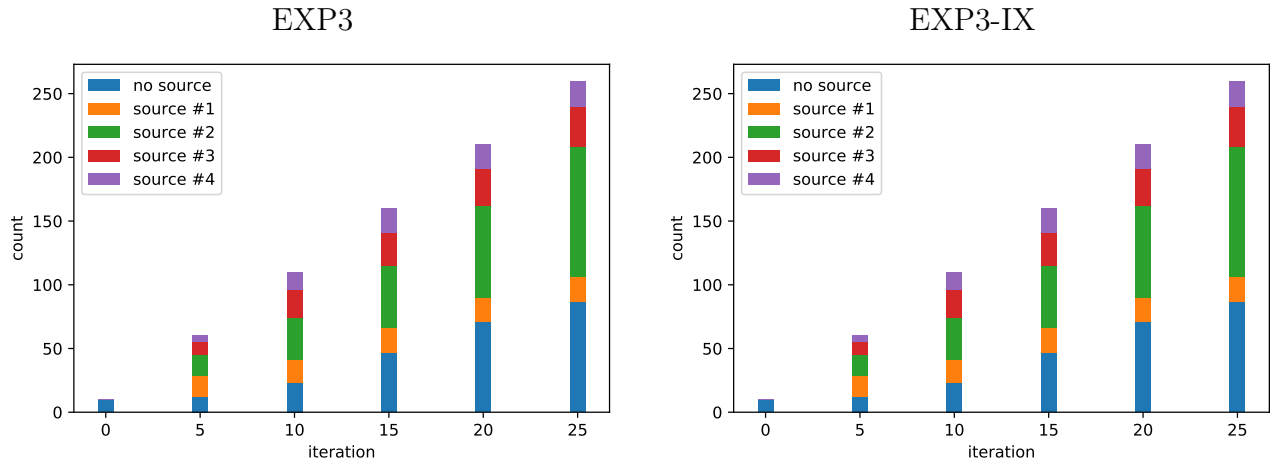


Таблица 2: Распределение выбора агентами действий, суммарно по всем 10 запускам.

Распределения получились почти идентичными. Третий по популярности вариант выбора 3-й источник. Вероятно такой результат получается из-за того, что третий источник все же отстоит дальше от целевой функции, чем второй.

5.3 Случай отсутствия близких источников

Для этого случая выберем координаты следующим образом $[-5, -5]$, $[4.5, 4.5]$, $[5, 5]$, $[6, 6]$ и $[-3, -3]$, $[6.5, 6.5]$, $[4, 4]$, $[4, 4]$ соответственно.

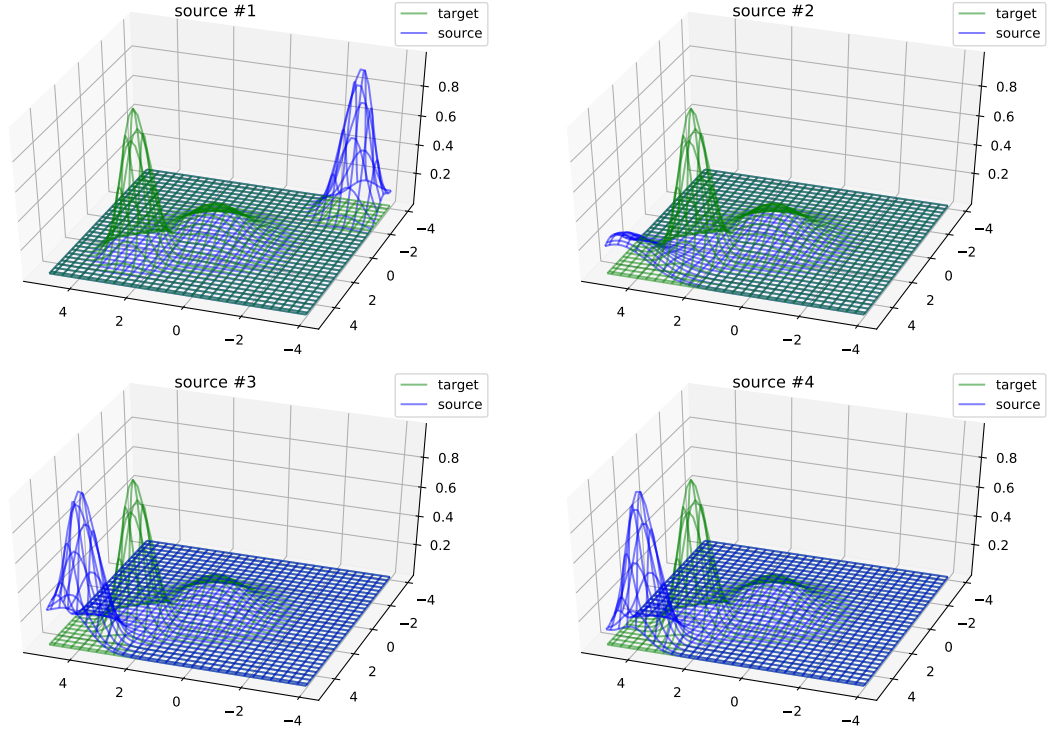


Рис. 5: Визуализация полученных функций.

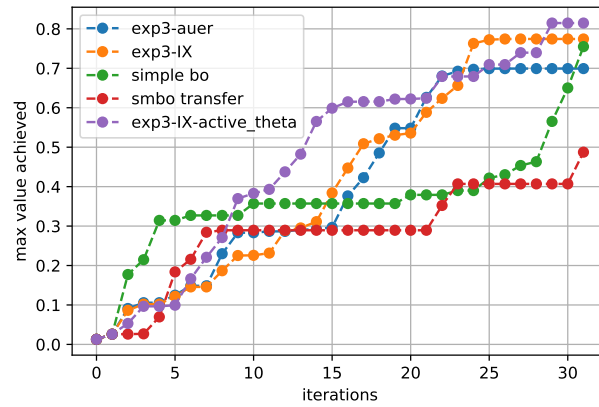


Рис. 6: Максимальное значение найденное к наступлению определенной итерации.

Результаты всех трёх алгоритмов (exp3, exp3-IX, exp3-IX active theta) оказались не хуже обычной байесовской оптимизации. Это значит, что модели не испытывают

негативных эффектов отсутствия хорошего трансфера. Так же можно заметить, что из всех трех сценариев именно в этом видна наибольшая разница между методами EXP3 и EXP3-IX. Это следствие того, что в этом сценарии доля полезных источников наименьшая, а в таких случаях алгоритм EXP3 уступает в устойчивости EXP3-IX.

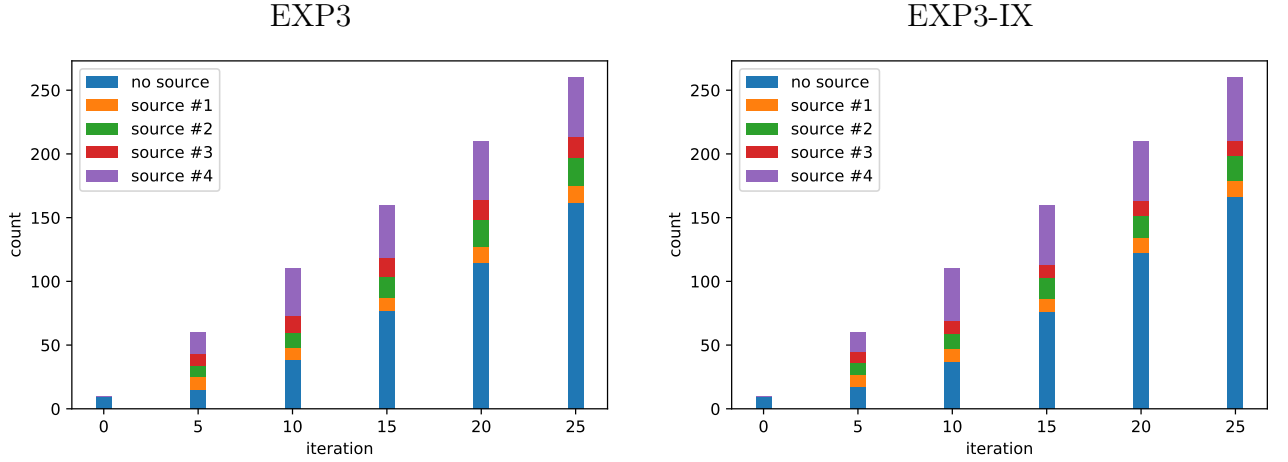


Таблица 3: Распределение выбора агентами действий, суммарно по всем 10 запускам.

Распределение рук получилось довольно похожим. Алгоритмы дают предпочтение в основном варианту с отсутствием трансфера, но так же можно заметить, что довольно популярны 3 и 4 источника. Но их доля растет только между 5-й и 10-й итерациями. На этих итерациях алгоритмы достигают точек с значением 0.2, что соответствует значениям точек, расположенных вблизи оснований гауссиан целевой функции и соответственно источников 3 и 4. Например в точке $[3.7, 3.7]$ третий и четвертый источники принимают довольно высокие значения 0.68 и 0.70 соответственно, в той же точке целевая функция равна 0.18.

5.4 Обсуждение и выводы

В результате экспериментов установлено:

- Подходы, основанные на активном выборе источников для трансфера с помощью обучения с подкреплением, показали более высокую скорость нахождения оптимальных точек черного ящика.

- Подходы, основанные на алгоритме EXP3-IX, оказались более устойчивы при низких долях близких источников.
- Полученные распределения действий агентов показали, что предложенные алгоритмы обладают способностью отбора близких к целевой функции источников.
- При росте количества точек, в которых оценена целевая функция, доля действия сопряженного с отсутствием источника трансфера увеличивается.
- Активный выбор параметра β_t функции сборщика с помощью обучения с подкреплением помогает увеличить скорость нахождения оптимальных значений целевой функции.

6 Заключение

В рамках проведенного исследования предложен новый подход к решению задачи автоматического поиска гиперпараметров на основе байесовской оптимизации с трансферным обучением.

Была экспериментально показана эффективность предложенного метода и его устойчивость даже при малом количестве близких источников.

Методы, активно выбирающие источники для трансфера с помощью обучения с подкреплением, не только оказались способны быстрее находить области оптимальных значений черного ящика, но и смогли выделить близкие к целевой функции источники. В дальнейших исследованиях можно использовать этот факт для проверки близости функций/задач машинного обучения.

В данной работе рассмотрены методы байесовской оптимизации, основанные на гауссовских процессах. Такие модели имеют существенный недостаток. Они не масштабируемы на большое количество данных об источниках трансфера, так как сложность гауссовской регрессии $O(N^3)$, где N количество оценок функции источника. Интерес представляют исследования масштабируемых модификаций предложенного подхода.

Список литературы

- [1] *Rasmussen, C. E. and Williams, C. K. I.*, Gaussian Processes for Machine Learning. — MIT Press, 2006.
- [2] *Hutter, F., Hoos, H., Leyton-Brown, K.*, Sequential model-based optimization for general algorithm configuration. — Springer, 2011.
- [3] *Snoek, J., Rippel, O., Swersky, K., Kiros, R., Satish, N., Sundaram, N., Patwary, M., Prabhat, Adams, R.*, Scalable Bayesian optimization using deep neural networks. — Proceedings of the 32nd International Conference on Machine Learning (ICML15), vol. 37. pp. 2171–2180. Omnipress , 2015.
- [4] *Jones, D., Schonlau, M., Welch, W.*, Efficient global optimization of expensive black box functions. — Journal of Global Optimization 13, 455-492, 1998.
- [5] *Srinivas N., Krause A., Kakade S. M. , Seeger M.*, Gaussian Process Optimization in the Bandit Setting: No Regret and Experimental Design. — *arXiv preprint arXiv:0912.3995*, 2010.
- [6] *P. Auer, N. Cesa-Bianchi, Y. Freund, and R. E. Schapire*, The nonstochastic multiarmed bandit problem. — SIAM J. Comput., 32(1):48-77, 2002
- [7] *G. Neu*, Explore no more: Improved high-probability regret bounds for non-stochastic bandits. — *arXiv preprint arXiv:1506.03271*, 2015.
- [8] *Dani Yogatama and Gideon Mann*, Efficient Transfer Learning Method for Automatic Hyperparameter Tuning. — Proceedings of the Seventeenth International Conference on Artificial Intelligence and Statistics, vol. 33. pp. 1077–1085. PMLR , 2014.
- [9] *Tinu Theckel Joy, Santu Rana, Sunil Gupta, Svetha Venkatesh*, A flexible transfer learning framework for Bayesian optimization with convergence guarantee. — Expert Systems with Applications, vol. 115. pp. 656–672., 2018.
- [10] *Ramachandran, A., Gupta, S., Rana, S., Venkatesh, S.*, Selecting optimal source for transfer learning in Bayesian optimisation. — Proceedings of PRICAI. pp. 42–56, 2018.

- [11] The variance of Exp3. — <https://banditalgs.com/2019/02/16/the-variance-of-exp3>, 2019.
- [12] *Ronald J. Williams.*, Simple statistical gradient-following algorithms for connectionist reinforcement learning. — In Machine Learning, 1992.
- [13] Code and experiments. — https://github.com/doorinthewall/diary_of_science_work, 2019.