Federated Multi-Task Learning

Артем Агарков

МФТИ

12 декабря 2023

Overview

Федеративное обучение

При федеративном обучении цель состоит в том, чтобы обучить модель на основе данных, которые находятся на m распределенных нодах и были сгенерированы ими. В качестве примера рассмотрим решение задачи классификации активности пользователей мобильных телефонов в сотовой сети на основе их индивидуальных данных датчиков, текстов или изображений.
Каждый узел (телефон), $t \in [m]$, может генерировать данные по своему распределению, поэтому естественно подбирать отдельные модели, $[w_1, w_2, \ldots, w_m]$, для распределенных данных - по одной для каждого локального набора данных.

Multi-task обучение

Multi-task обучение - это область машинного обучения, в которой одновременно решаются несколько задач обучения, при этом используются общие черты и различия между задачами. Это может привести к повышению эффективности обучения и точности предсказания для моделей, специфичных для конкретной задачи, по сравнению с обучением моделей по отдельности.

Multi-task обучение

Пример для задачи классификации: спам-фильтр, который можно рассматривать как отдельные, но связанные между собой задачи классификации для разных пользователей.
Предположим, что разные люди имеют различные распределения признаков, которые отличают спам от допустимых писем, например, англоговорящий человек может решить, что все письма на русском языке являются спамом, а русскоговорящий - нет. Тем не менее, в этой задаче классификации есть определенная общность для разных пользователей, например, одним из общих признаков может быть текст, связанный с переводом денег.

Федеративное multi-task обучение (FMTL)

Вернемся к примеру федеративного обучния для задачи изучения активности пользователей телефонов. Между моделями часто существует структура (например, люди могут вести себя одинаково при использовании своих телефонов). Тогда моделирование этой структуры с помощью multi-task обучения является естественной стратегией для **повышения производительности** и **увеличения размера выборки** для каждого узла.

Общая постановка multi-task обучния

Данные $X_t \in \mathbb{R}^{d \times n_t}$ для каждой из m нод в multi-task подходе обучаем параметры $w_t \in \mathbb{R}^d$ на данных для каждой ноды с произвольной выпуклой функцией потерь ℓ_t

$$\min_{\boldsymbol{W}, \boldsymbol{\Omega}} \left(\sum_{t=1}^{m} \sum_{i=1}^{n_t} \ell_t(\boldsymbol{W}_t^T \boldsymbol{x}_t^i, y_t^i) + \mathcal{R}(\boldsymbol{W}, \boldsymbol{\Omega}) \right) \tag{1}$$

$\boldsymbol{W} := [\boldsymbol{w}_1, \boldsymbol{w}_2, \ldots, \boldsymbol{w}_m] \in \mathbb{R}^{d \times m}$ (t-й столбец - параметры для t-й задачи. Матрица $\boldsymbol{\Omega} \in \mathbb{R}^{m \times m}$ отвечает за взаимосвязь между задачами. Может быть задана изначально или найдена в процессе обучения. Существуют различные подходы к определению $\mathcal{R}$ и, соответственно, задачи MTL. Наиболее распространенная бивыпуклая формулировка:

$$\mathcal{R}(\boldsymbol{W}, \boldsymbol{\Omega}) = \lambda_1 \mathrm{tr}(\boldsymbol{W} \boldsymbol{\Omega} \boldsymbol{W}^T) + \lambda_2 \|\boldsymbol{W}\|_F^2 \tag{2}$$

Некоторые наблюдения

Наблюдение 1

В общем случае, (1) не является выпуклой функцией по $\boldsymbol{W}, \boldsymbol{\Omega}$. И даже в слачае, когда (1) выпуклая, решение для $\boldsymbol{W}, \boldsymbol{\Omega}$ одновременно может быть сложным.

Наблюдение 2

При фиксировании $\boldsymbol{\Omega}$ обновление $\boldsymbol{W}$ зависит как от данных $\boldsymbol{X}$, которые распределены по узлам, так и от структуры $\boldsymbol{\Omega}$, которая известна централизованно.

Наблюдение 3

При фиксированном $\boldsymbol{W}$ оптимизация для $\boldsymbol{\Omega}$ зависит только от $\boldsymbol{W}$, но не от данных $\boldsymbol{X}$.

Некоторые наблюдения

Исходя из этих наблюдений, естественно предложить подход попеременной оптимизации для решения задачи (1), при котором на каждой итерации мы фиксируем либо $\boldsymbol{W}$, либо $\boldsymbol{\Omega}$ и оптимизируем по другому параметру, чередуя их до достижения сходимости. Отметим, что решение для $\boldsymbol{\Omega}$ не зависит от данных и поэтому может быть вычислено централизованно; поэтому далее рассматриваем только задачу оптимизации для $\boldsymbol{W}$.

Вывод двойственной задачи

Итак, мы рассматриваем итерацию с оптимизацией $\boldsymbol{W}$:

$$\min_{\boldsymbol{W}} \left(\sum_{t=1}^{m} \sum_{i=1}^{n_t} \ell_t(\boldsymbol{W}_t^T \boldsymbol{x}_t^i, y_t^i) + \mathcal{R}(\boldsymbol{W}, \Omega) \right)$$

Более абстрактно, можем записать исходную задачу в виде:

$$\min_{\boldsymbol{w}, \boldsymbol{v}} (f(\boldsymbol{v}) + g(\boldsymbol{w})), \text{s.t.} \boldsymbol{v} = \boldsymbol{X}^T \boldsymbol{w},$$

где $\boldsymbol{w} \in \mathbb{R}^{dm}$ - "вытянутый"вектор параметров,
$X := \text{Diag}(\boldsymbol{X}_1, \boldsymbol{X}_2, \ldots, \boldsymbol{X}_m) \in \mathbb{R}^{md \times n}$. Будем искать двойственную задачу для этой абстрактной формулировки.

Вывод двойственной задачи

Лагранжиан: $L(\boldsymbol{w}, \boldsymbol{v}; \boldsymbol{\alpha}) = f(\boldsymbol{v}) + g(\boldsymbol{w}) + \boldsymbol{\alpha}^T(\boldsymbol{X}^T\boldsymbol{w} - \boldsymbol{v})$.
Тогда инфинум Лагранжиана может быть преобразован так:

$$\begin{aligned}\inf_{\boldsymbol{w},\boldsymbol{v}} L(\boldsymbol{\alpha}, \boldsymbol{w}, \boldsymbol{v}) &= \inf_{\boldsymbol{v}} \left(f(\boldsymbol{v}) - \boldsymbol{\alpha}^T\boldsymbol{v}\right) + \inf_{\boldsymbol{w}} \left(g(\boldsymbol{w}) + \boldsymbol{\alpha}^T\boldsymbol{X}^T\boldsymbol{w}\right) \\ &= -\sup_{\boldsymbol{v}} \left(\boldsymbol{\alpha}^T\boldsymbol{v} - f(\boldsymbol{v})\right) - \sup_{\boldsymbol{w}} \left(-(\boldsymbol{\alpha}^T\boldsymbol{X}^T)\boldsymbol{w} - g(\boldsymbol{w})\right) \\ &= -f^*(\boldsymbol{\alpha}) - g^*(-\boldsymbol{X}\boldsymbol{\alpha})\end{aligned}$$

Далее просто меняем знаки и получаем двойственную задачу оптимизации:

$$\max_{\boldsymbol{\alpha}} \left(-f^*(\boldsymbol{\alpha}) - g^*(-\boldsymbol{X}\boldsymbol{\alpha})\right) = \min_{\boldsymbol{\alpha}} \left(f^*(\boldsymbol{\alpha}) + g^*(-\boldsymbol{X}\boldsymbol{\alpha})\right)$$

И еще раз поменяем знак для эстетики:

$$\min_{\boldsymbol{\alpha}} \left(f^*(-\boldsymbol{\alpha}) + g^*(\boldsymbol{X}\boldsymbol{\alpha})\right)$$

Двойственная задача

Для задачи (1) рассмотрим выведенную двойственную задачу, которая позволит перейти к распределенным подзадачам.
Пусть $n := \sum_{t=1}^{m} n_t$ и $X := \mathrm{Diag}(\boldsymbol{X}_1, \boldsymbol{X}_2, \ldots, \boldsymbol{X}_m) \in \mathbb{R}^{md \times n}$. При фиксированной $\boldsymbol{\Omega}$:

$$\min_{\boldsymbol{\alpha}} \left(\mathcal{D}(\boldsymbol{\alpha} := \sum_{t=1}^{m} \sum_{i=1}^{n_t} \ell_t^*(-\boldsymbol{\alpha}_t^i) + \mathcal{R}^*(\boldsymbol{X}\boldsymbol{\alpha}) \right), \tag{3}$$

где ℓ_t^* и $\mathcal{R}^*$ сопряженные двойственный функции ℓ_t и $\mathcal{R}$ соответственно, и $\boldsymbol{\alpha}_t^i$ - двойственная переменная для данных $(\boldsymbol{x}_t^i, y_t^i)$

Двойственная задача

Заметим, что $\mathcal{R}^*$ зависит от $\mathbf{\Omega}$, но для простоты обозначений не будем это указывать. Чтобы вывести распределенные подзадачи из этой глобальной двойственной задачи, сделаем предположение, описанное ниже, на регуляризатор $\mathcal{R}$.

Предположение 1

Для $\mathbf{\Omega}$, мы предполагаем, что существует симметричная положительно определенная матрица $\boldsymbol{M} \in \mathbb{R}^{md \times md}$, зависящая от $\mathbf{\Omega}$, для которой функция $\mathcal{R}$ является сильно выпуклой относительно M^{-1}. Заметим, что это соответствует предположению, что $\mathcal{R}^*$ будет гладкой относительно матрицы $\boldsymbol{M}$.

Локальные квадратичные подзадачи

Для решения (1) на узлах определяем следующие локальные подзадачи, которые являются квадратичной аппроксимацией двойственной задачи (3). Эти подзадачи находят обновления $\Delta\boldsymbol{\alpha}_t \in \mathbb{R}^{n_t}$ для двойственных переменных в $\boldsymbol{\alpha}$, и требуют только локальных данных X_t. t-я подзадача задается:

$$\min_{\Delta\boldsymbol{\alpha}_t} \mathcal{G}_t^{\sigma'}(\Delta\boldsymbol{\alpha}_t; \boldsymbol{v}_t, \boldsymbol{\alpha}_t)$$

$$\mathcal{G}_t^{\sigma'}(\Delta\boldsymbol{\alpha}_t; \boldsymbol{v}_t, \boldsymbol{\alpha}_t) := \sum_{i=1}^{n_t} \ell_t^*(-\boldsymbol{\alpha}_t^* - \Delta\boldsymbol{\alpha}_t^i) + \langle \boldsymbol{w}_t(\boldsymbol{\alpha}), \boldsymbol{X}_t\Delta\boldsymbol{\alpha}_t\rangle + \\ + \frac{\sigma'}{2}\|\boldsymbol{X}_t\Delta\boldsymbol{\alpha}_t\|_{M_t}^2 + c(\boldsymbol{\alpha}),$$

где $c(\boldsymbol{\alpha}) = \frac{1}{m}\mathcal{R}^*(\boldsymbol{X}\boldsymbol{\alpha})$, $M_t \in \mathbb{R}^{d\times d}$ - t-й диагональный блок симметричной положительно определенной матрицы M, $\boldsymbol{v} = \boldsymbol{X}\boldsymbol{\alpha}$.

Связь primal и dual переменных

После того, как мы нашли обновление $\Delta\boldsymbol{\alpha}$, необходимо обновить параметры $\boldsymbol{w}_t$ на каждой ноде. Связь primal и dual переменных найдем из условий оптимальности первого порядка.
Заметим, что если $\mathcal{R}$ выпуклая, замкнутая, собственная функция, мы можем воспользоваться теоремой Фенхеля-Моро: $\mathcal{R}^{**} = \mathcal{R}$.
Перепишем по определению:

$$\mathcal{R}(\boldsymbol{w}) = \sup_{\boldsymbol{\alpha}} \left(\boldsymbol{w}^T\boldsymbol{\alpha} - \mathcal{R}^*(\boldsymbol{X}\boldsymbol{\alpha}) \right).$$

Тогда если на $\boldsymbol{\alpha}^*$ достигается sup, выполнены условия оптимальности первого порядка:

$$\nabla_{\boldsymbol{\alpha}} \left(\boldsymbol{w}^T\boldsymbol{\alpha}^* - \mathcal{R}^*(\boldsymbol{X}\boldsymbol{\alpha}^*) \right) = 0 \Rightarrow \boldsymbol{w}(\boldsymbol{\alpha}^*) = \nabla\mathcal{R}^*(\boldsymbol{X}\boldsymbol{\alpha}^*)$$

Обновление Ω

Мы выяснили, как делать распределенную оптимизацию по $\boldsymbol{W}$.
Однако в нашем алгоритме еще есть шаг с централизованнным обновлением $\boldsymbol{\Omega}$.
Как мы говорили в начале, существуют разные подходы к задаче MTL (определению $\mathcal{R}$). Будем рассматривать задачу MTL со структурой кластеров.

Структура с кластерами

При такой постановке векторы весов для каждой задачи $\boldsymbol{w}_t$ предполагаются "близкими"по некоторой метрике к остальным весам в кластере. То есть все веса в калстере располагаются "близко"к среднему.
Такую структуру описывает матрица $\boldsymbol{\Omega} = (\boldsymbol{I}_{m\times m} - \frac{1}{m}11^T)^2$.
В случае кластеров регуляризатор имеет вид:

$$\mathcal{R}(\boldsymbol{W}, \boldsymbol{\Omega}) = \lambda_1 \text{tr}(\boldsymbol{W}\boldsymbol{\Omega}\boldsymbol{W}^T) + \lambda_2 \|\boldsymbol{W}\|_F^2,$$

где $\lambda_1, \lambda_2 > 0$ параметры. Здесь считаем матрицу $\boldsymbol{\Omega}$ известной заранее, количество кластеров $= 1$.
Однако в случае большего числа кластеров $\mathcal{R}$ может быть невыпуклой, поэтому производится выпуклая релаксация:

$$\mathcal{R}(\boldsymbol{W}, \boldsymbol{\Omega}) = \lambda \text{tr}(\boldsymbol{W}(\eta\boldsymbol{I}+\boldsymbol{\Omega})^{-1}\boldsymbol{W}^T), \boldsymbol{\Omega} \in \mathcal{Q} = \{\boldsymbol{Q} | \boldsymbol{Q} \succeq 0, \text{tr}(\boldsymbol{Q} = k, \boldsymbol{Q} \preceq \boldsymbol{I}\},$$

где λ и ν параметры регуляризации, k - количество кластеров.

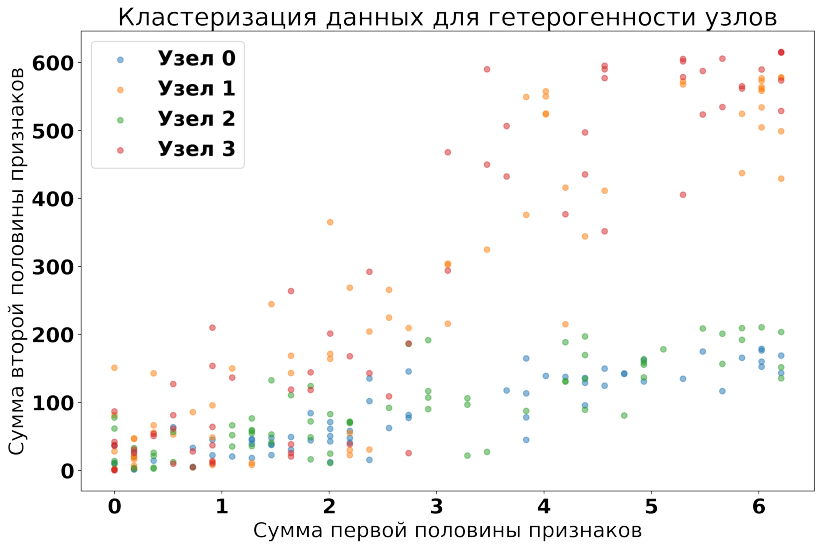
Алгоритм

Algorithm 1 MOCHA: Federated Multi-Task Learning Framework

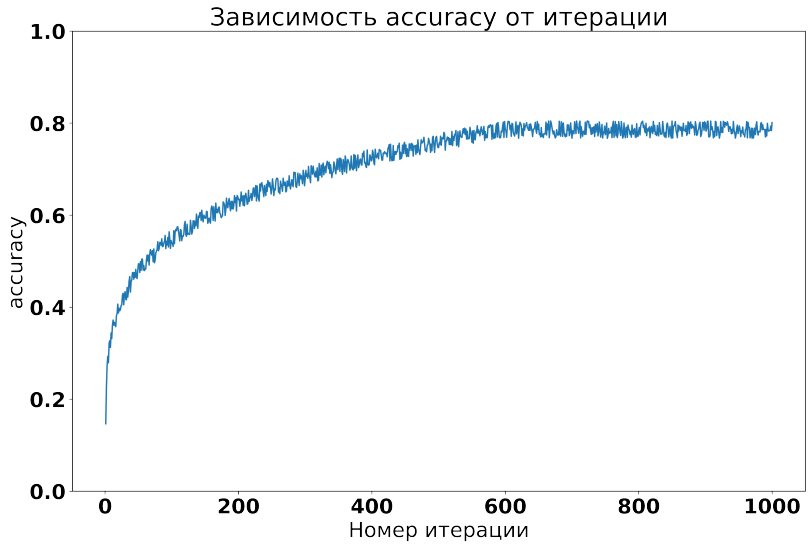
Input: Data $\mathbf{X}_t$ from $t = 1, \ldots, m$ tasks, stored on one of m nodes, and initial matrix $\mathbf{\Omega}_0$
Starting point $\boldsymbol{\alpha}^{(0)} := \mathbf{0} \in \mathbb{R}^n$, $\mathbf{v}^{(0)} := \mathbf{0} \in \mathbb{R}^b$
for iterations $i = 0, 1, \ldots$ **do**
    Set subproblem parameter σ' and number of federated iterations, H_i
    for iterations $h = 0, 1, \cdots, H_i$ **do**
        for tasks $t \in \{1, 2, \ldots, m\}$ **in parallel over** m **nodes do**
            call local solver, returning θ_t^h-approximate solution $\Delta\boldsymbol{\alpha}_t$ of the local subproblem
            update local variables $\boldsymbol{\alpha}_t \leftarrow \boldsymbol{\alpha}_t + \Delta\boldsymbol{\alpha}_t$
            return updates $\Delta\mathbf{v}_t := \mathbf{X}_t\Delta\boldsymbol{\alpha}_t$
        reduce: $\mathbf{v}_t \leftarrow \mathbf{v}_t + \Delta\mathbf{v}_t$
    Update $\mathbf{\Omega}$ centrally based on $\mathbf{w}(\boldsymbol{\alpha})$ for latest $\boldsymbol{\alpha}$
Central node computes $\mathbf{w} = \mathbf{w}(\boldsymbol{\alpha})$ based on the lastest $\boldsymbol{\alpha}$
return: $\mathbf{W} := [\mathbf{w}_1, \ldots, \mathbf{w}_m]$

Эксперименты

С помощью рассмотренного метода попробуем решить задачу бинарной классификации.
Датасет: phishing из LibSVM
Чтобы обеспечить гетерогенность в данных, выделим кластеры с помощью алгоритма k-means. Полученные кластеры будут нодами.



(a) Гетерогенность



(b) Accuracy

The End