

УДК 004.852

doi: 10.15622/rcai.2025.077

МЕТОД УПРАВЛЕНИЯ ПОКРЫТИЕМ В ЗАДАЧЕ ОБУЧЕНИЯ С ДЕЛЕГИРОВАНИЕМ¹

А.В. Пономарев (*ponomarev@iias.spb.su*)

Санкт-Петербургский Федеральный исследовательский центр РАН,
Санкт-Петербург

Во многих случаях модель машинного обучения используется не автономно, а как часть более сложной системы, которая может включать экспертов-людей. Методы обучения с делегированием (learning to defer) позволяют создавать модели, учитывающие вероятность ошибок как модели машинного обучения, так и эксперта-человека, и соответствующим образом распределять образцы для максимизации общей точности системы. Однако многие такие методы не позволяют ограничивать долю образцов, назначаемых эксперту, что оказывается важным в практических приложениях, поскольку количество экспертов и их пропускная способность обычно ограничены. В данной статье предлагается простой, но эффективный эвристический метод, позволяющий накладывать ограничения на долю образцов, перенаправляемых эксперту, тем самым помогая сбалансировать метрики точности предсказаний и покрытия. Предложенный метод может использоваться совместно со многими существующими методами обучения с делегированием и обучения с отказом (rejection learning).

Ключевые слова: обучение с отказом, обучение с делегированием, человеко-машинная система, принятие решений, совместная работа человека и ИИ.

Введение

Во многих практических приложениях модели машинного обучения не используются автономно, они являются частью процесса принятия решений, включающего и эксперта-человека. При этом сама организация процесса принятия решений, объединяющая действия, совершаемые моделью и экспертом, может очень сильно разниться [Пономарев и др., 2025].

¹ Работа выполнена при финансовой поддержке РФФ (проект № 24-21-00337).

В ряде работ показано [Madras et al., 2018], [Wilder et al., 2020] что организация процесса обучения с учетом всех элементов системы, в рамках которой будет применяться модель, как правило, способствует улучшению результатов, получаемых этой системой. Таким образом, модели машинного обучения, учитывающие возможность взаимодействия с человеком (например, просто воздерживающиеся от классификации образцов, в отношении которых они не уверены), особенно востребованы в ответственных и критически важных приложениях. Наиболее широко используемый метод построения таких моделей заключается в обучении модели (например, для классификации) с последующей оценкой неопределенности предсказания на этапе вывода и перенаправлении к эксперту-человеку только тех образцов, для которых модель не уверена в результатах. Существует несколько подходов к оценке неопределенности [Cordelia et al., 1995], [Gal et al., 2016], [Lakshminarayanan et al., 2017], и потенциально может быть использован любой из них. Основным недостатком данного подхода заключается в том, что он игнорирует ограниченность знаний человека (и соответствующую вероятность ошибки), рассматривая его как оракула.

Данный недостаток был скорректирован в работах, посвященных обучению с делегированием (learning to defer) [Madras et al., 2018], в которых эксперт-человек рассматривается как часть оптимизируемой системы. Это означает, что такие алгоритмы оптимизируют стратегию делегирования, учитывая не только качество предсказания модели машинного обучения в различных областях пространства признаков, но и точность эксперта-человека, которая также может различаться в разных областях этого пространства.

Большинство подходов к обучению с перенаправлением опираются на специально разработанные функции потерь, балансирующие компоненты, отвечающие за автоматический классификатор и эксперта-человека [Mozannar et al., 2020], [Verma et al., 2022], [Wilder et al., 2020]. Однако существенным ограничением многих методов обучения с делегированием является то, что они игнорируют ограниченную пропускную способность эксперта [Leitão et al., 2022].

Было также предложено несколько методов для учёта ограниченной пропускной способности эксперта, обычно через формулировку задачи обучения с перенаправлением как задачи смешанного программирования (MILP, Mixed-Integer Linear Programming), например [Alves et al., 2024], [De et al., 2020], [De et al., 2021], [Mozannar et al., 2023]. В работах [De et al., 2020], [De et al., 2021] MILP решается для обучающего набора, после чего его решение аппроксимируется дополнительной моделью (предложенная MILP-формулировка ограничена определёнными типами моделей). В [Alves et al., 2024], [Mozannar et al., 2023] MILP строится на этапе вывода, поэтому эти подходы могут лишь распределять фиксиро-

ванный набор образцов, но плохо подходят для ситуаций, когда образцы должны распределяться между моделью и экспертом по мере их поступления. Кроме того, решение MILP может быть вычислительно затратным, особенно для больших наборов данных.

В статье рассматривается классическая постановка задачи обучения с делегированием, когда имеется набор данных, содержащий для образов помимо эталонных меток (истинных классов) также экспертные метки, что в неявной форме задает модель ошибок эксперта и характеризует области его компетенций. Необходимо найти функцию, для заданного образца возвращающую либо класс, к которому относится образец, либо осуществляющую перенаправление образца эксперту. При этом в статье предлагается эффективный эвристический метод, позволяющий накладывать ограничения на долю образцов, направляемых эксперту, тем самым способствуя балансированию метрик точности и покрытия. Данный метод может использоваться совместно со многими существующими методами обучения с делегированием и обучения с отказом (rejection learning). Проведена оценка эффективности метода с использованием трёх распространённых методов обучения с делегированием и двух наборов данных – синтетического и реального, собранного с помощью краудсорсинга.

Постановка задачи

Имеется набор данных \mathcal{D} , где x_i – признаки, описывающие объекты, y_i – истинные метки объектов, а \hat{y}_i – метки, присвоенные объектам экспертом (не обязательно соответствующие истинным меткам). Необходимо найти две функции – классификатор f и функцию делегирования g . Предсказание для определенного образца получается с помощью этой пары функций следующим образом:

$$\hat{y} = \begin{cases} \hat{y}_i & \text{если } g(x_i) > 0 \\ f(x_i) & \text{иначе} \end{cases}$$

Поскольку доступ к экспертным оценкам может быть ограничен во время вывода, эти функции должны максимизировать качество классификации при соблюдении ограничения на количество обращений к эксперту (или долю образцов из общего множества, для которых такое обращение производится). Формально:

На практике математические ожидания в уравнении выше обычно оцениваются с помощью эмпирических метрик, вычисляемых на основе тестового набора данных, взятого из того же распределения. Математическое ожидание числа правильных ответов соответствует точности, а математическое ожидание выборов, назначенных модели, соответствует покрытию (доля образцов, классифицированных моделью без обращения к эксперту).

Необходимость учёта ошибок как модели машинного обучения, так и эксперта приводит к естественной функции потерь, используемой для обучения и [Madras et al., 2018]:

где L_m – функция потерь модели, а L_e – функция потерь эксперта. Например, если это задача бинарной классификации, то L_e может быть бинарной кросс-энтропией (для L_m ситуация несколько сложнее, см. ниже).

Однако прямое использование естественной функции потерь имеет два основных недостатка:

- В обучающих данных m представляет класс, предоставленный конечным пользователем, поэтому, если эксперт ошибается, бинарная кросс-энтропия не имеет конечного значения. На практике можно либо использовать какое-то достаточно большое значение, либо использовать другую функцию потерь (например, L_1). В любом случае, конкретное значение, соответствующее ошибке пользователя, должно быть каким-то образом масштабировано до диапазона первого члена (L_m).

- Функция потерь оптимизирует модели только с точки зрения точности, не обращая внимания на то, что доступ к пулу экспертов может быть ограничен.

В литературе предложены некоторые суррогатные функции потерь (например, [Mozannar et al., 2020]), смягчающие первую проблему, но они по-прежнему оптимизируют только с точки зрения точности.

Предлагаемый метод

Предлагаемый метод основан на возможности получения оценки, отражающей относительную уверенность классификации образца моделью по отношению к классификации того же образца экспертом. Обозначим эту оценку как o . Абсолютные значения этой оценки не имеют значения; вместо этого, оценка устанавливает порядок: если

$o_1 > o_2$, то назначение модели (а не эксперту) приведёт к меньшей вероятности ошибки, чем назначение модели. То есть, данная функция устанавливает предпочтительность назначения объекта модели. Конкретные примеры построения o для нескольких существующих методов обучения с делегированием приведены в этом разделе.

Метод состоит из трёх этапов:

- Обучение моделей m и e с использованием существующего алгоритма обучения с делегированием. Полученная пара может выполнять делегирование, обычно достигая хорошей (или оптимальной в некотором смысле) точности, но не удовлетворяя ограничениям по покрытию.

- С использованием отдельного набора данных D и Алгоритма 1 (рис. 1) найти пороговое значение оценки τ , соответствующее требуемому покрытию γ .

- Применить Алгоритм 2 (рис. 2) для классификации любых поступающих экземпляров, взятых из выборки .

Алгоритм обучения (Алгоритм 1) оценивает значения оценок для всех образцов , а затем рассматривает каждое значение оценки как возможное пороговое значение для назначения всех образцов с большими (или равными) значениями оценки модели, а остальных – эксперту (что выполняется с помощью функции , определённой как , если , и в противном случае). Затем он оценивает точность и покрытие каждого такого разбиения и выбирает значение оценки такое, что: по меньшей мере (требуемое покрытие) образцов имеют большие значения оценок, а точность максимальна (среди всех значений, удовлетворяющих ограничению по требуемому покрытию).

Алгоритм 1 – Обучение

Входные данные:

Выходные данные:

for do

end for

for all do

if and then

end for

end for

Рис. 1. Алгоритм обучения

Алгоритм 2 – Вывод

Входные данные:

Выходные данные: значение (класс) или запрос эксперту

if then

return

else

return

end if

Рис. 2. Алгоритм вывода

Алгоритм вывода (Алгоритм 2) оценивает значение функции заданного образца, сравнивает его с пороговым значением, найденным в процессе обучения, и перенаправляет его соответствующим образом. Следует отметить, что алгоритм использует только значение параметра, оцененное с помощью Алгоритма 1, и образец x , следовательно, он может применяться к выборкам по мере их поступления, без необходимости обработки больших объемов данных (в отличие, например, от [Alves et al., 2024]).

Рассмотрим функции оценки для нескольких алгоритмов обучения с делегированием. Простейший алгоритм основан на пороговом значении уверенности (будем обозначать его как Threshold). Он не имеет отдельной модели отклонения, но перенаправляет экземпляры, для которых максимальный выходной сигнал многопеременной логистической функции (softmax) [Cordelia et al., 1995] ниже определенного порога (порог обычно устанавливается для максимизации точности), к эксперту-человеку. Данный алгоритм нечувствителен к различиям компетентности эксперта в пространстве входных признаков, поэтому функция может быть определена как любая мера уверенности классификатора, например, максимальное значение многопеременной логистической функции (softmax).

Другой подход заключается в прямом использовании «естественной» функции потерь для одновременного обучения двух моделей – классификатора и функции делегирования (NatLoss). В этом случае имеется отдельная функция делегирования, выход которой (сигмоидальная функция до бинаризации) задает значение, соответствующее

Наконец, рассмотрим потерю, основанную на параметризации многопеременной логистической функции, предложенную в [Mozannar et al., 2020] (SE). В этом случае и моделируются с помощью одной нейронной сети с выходом (где – число классов). Выходы с 1 по соответствуют вероятностям классов, а -й выход соответствует перенаправлению к эксперту-человеку. Пусть – значение i -го выхода. В этой модели определяется как, а равно 1, если, и 0 в противном случае. Функцию оценки предлагается определить как. Интуитивно, это отражает разницу между уверенностью классификатора и оценкой надежности эксперта-человека для рассматриваемого примера.

Экспериментальное исследование

Экспериментальное исследование проведено с использованием двух наборов данных – синтетического и набора данных с реальными метками, полученного с применением краудсорсинга (CIFAR-10H).

Синтетический набор разработан для случая бинарной классификации. Подмножество из этого набора данных представлено на рис. 3. График слева иллюстрирует два класса, а график справа использует цветовое кодирование для выделения образцов, в которых эксперт выдал правильный (зелёный) и неправильный (красный) результат. Общая идея здесь в том, что существует область (низкие значения x_1), где классы относительно хорошо разделены, и область (высокие значения x_1), где их разделение затруднено. При этом компетентность эксперта максимальна при высоких значениях x_1 и минимальна при низких. В результате можно ожидать, что при низких значениях x_1 классификация будет выполняться моделью, а при высоких — делегирована эксперту. Набор данных генерируется с помощью нормального распределения и параметр этого распределения управляет перекрытием между классами и ограничивает точность модели классификации (даже при низких значениях x_1). В ходе экспериментального исследования использовался синтетический набор размером 5000 образцов.

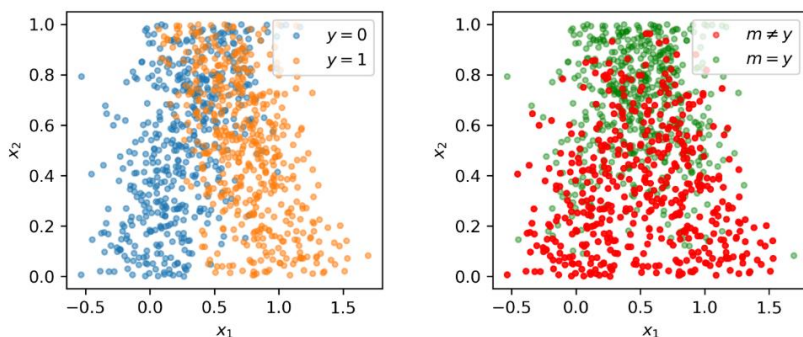


Рис. 3. Подмножество синтетического набора данных

В качестве реального набора данных используется CIFAR-10H [Peterson et al., 2019], подмножество распространенного в компьютерном зрении набора данных CIFAR-10 [Krizhevsky, 2009], содержащее 10000 изображений, для которых метки были получены с помощью краудсорсинга. Для каждого изображения в этом наборе данных имеется несколько меток, полученных от разных людей; для формирования метки одна из них была выбрана случайным образом.

Разбиение на обучающее, калибровочное и тестовое множества производилось в соотношении 80/10/10. Таким образом, размеры множеств для синтетического набора данных были 4000/500/500, а для CIFAR-10H — 8000/1000/1000.

Для синтетического набора данных использовалась архитектура многослойного перцептрона (MLP) с двумя скрытыми слоями (по 40 нейронов в каждом) и функцией активации ReLU. Количество таких моделей, конфигурация выходного слоя и функция потерь варьировались в зависимости от исследуемых подходов обучения делегированию:

- для Threshold использовалась одна модель MLP с двумя выходными нейронами (соответствующими классам), обученная с использованием кросс-энтропии только на метках истинных значений;
- для NatLoss использовались две модели MLP (одна для \mathcal{D} и одна для \mathcal{D}^*), каждая с одним выходным нейроном, обученные одновременно с использованием $\mathcal{L}_{\text{NatLoss}}$;
- для SP использовалась одна модель MLP с тремя выходными нейронами (два из которых соответствуют классам, а один – откладыванию решения эксперту), обученная с использованием функции потерь с softmax-параметризацией из [Mozannar et al., 2020].

Во всех случаях использовался оптимизатор Adam, и обучение проводилось в пакетном режиме до сходимости (изменение значения функции потерь на обучающем множестве менее 10^{-6}).

Для набора данных CIFAR-10H была обучена модель ResNet-18 на наборе данных CIFAR-10 (исключая изображения, также входящие в CIFAR-10H), достигнув точности классификации около 86%. Затем изображения CIFAR-10H были преобразованы в их сжатые представления шириной 512, используя выход слоя, предшествующего классификационному блоку. Все модели для обучения «человек-ИИ» (классификация и стратегии делегирования) представляют собой MLP с двумя скрытыми слоями по 80 и 40 нейронов соответственно и функцией активации ReLU. Конфигурация выходного слоя и функции потерь были такими же, как описано выше (но с 10 классами).

Главный вопрос, требующий ответа в ходе эксперимента, заключается в том, позволяет ли предложенный эвристический метод накладывать ограничение на значение покрытия во время вывода. Поведение конкретной модели относительно этого требования может быть визуализировано с помощью графиков «требуемое покрытие – тестовое покрытие» (или СС-диаграммы). По оси X откладывается требуемое покрытие (задаваемое во время обучения модели), по оси Y – тестовое покрытие, оцениваемое с использованием тестового набора. Следует отметить, что в соответствии с формальным определением задачи, тестовое покрытие должно быть больше или равно необходимому покрытию, следовательно, график должен располагаться выше диагональной линии.

Рис. 4 демонстрирует примеры СС-диаграмм для синтетического набора данных (слева) и для набора данных CIFAR-10H (справа). Видно, что тестовое покрытие фактически следует за необходимым покрытием в определенном диапазоне необходимого покрытия, однако при низком необходимом покрытии тестовое покрытие оказывается значительно больше необ-

ходимого. Это происходит потому, что предлагаемый алгоритм определяет значение параметра, максимизирующее точность и удовлетворяющее ограничению на покрытие. Однако при определённом покрытии достигается максимальная точность, поэтому для всех значений необходимого покрытия, меньших этого, возвращается одно и то же тестовое покрытие (соответствующее общей наилучшей точности системы «человек-ИИ»).

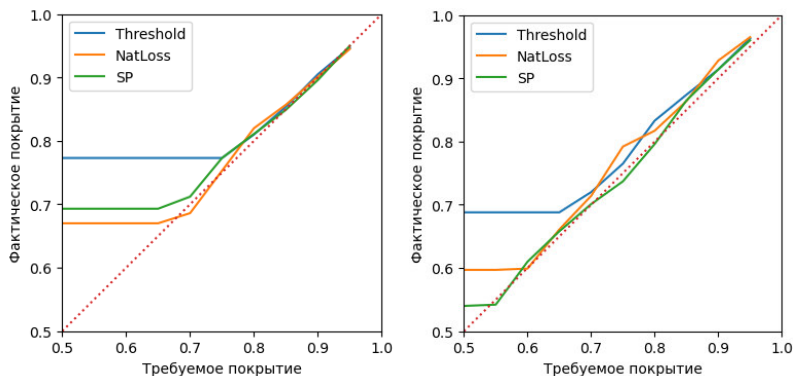


Рис. 4. СС-диаграммы для синтетического набора (слева) и CIFAR-10H (справа)

СС-диаграммы позволяют получить представление о поведении алгоритма при использовании определённой обученной модели и заданных меток, предоставленных человеком. Однако такие диаграммы обладают достаточно низкой обобщающей способностью. В связи с этим, нами также были построены диаграммы, демонстрирующие распределение максимального нарушения требования покрытия для модели. Рис. 5 иллюстрирует пример такой диаграммы, построенной для 100 сгенерированных синтетических наборов данных (и соответствующих моделей). Можно видеть, что ограничение, накладываемое на покрытие может быть нарушено, однако в подавляющем большинстве случаев это нарушение составляет менее 1%, а во всех случаях – менее 5%.

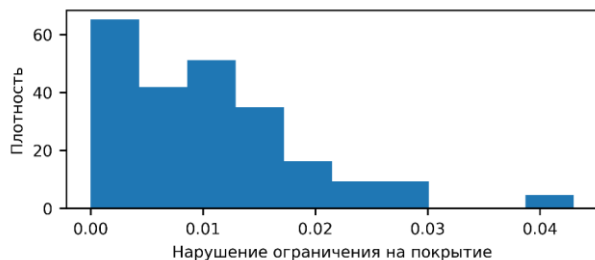


Рис. 5. Гистограмма величины нарушения ограничения на покрытие

Заключение

В статье описан эвристический метод, позволяющий накладывать ограничения на покрытие в существующих моделях обучения с делегированием. Проведена оценка предложенного метода в сочетании с тремя популярными методами обучения с делегированием: основанным на уверенности модели, на естественной функции потерь и на суррогатной функции потерь.

Вычислительные эксперименты на двух наборах данных – синтетическом и наборе CIFAR-10H, собранном методом краудсорсинга, – показали, что заданное ограничение на покрытие в основном соблюдается. Хотя возможны отдельные случаи нарушения ограничений, они встречаются нечасто и не являются значительными. Тем не менее, целесообразно провести теоретический анализ метода с целью установления теоретических границ на величину и частоту возможного нарушения задаваемого ограничения по покрытию.

Список литературы

- [Пономарев и др., 2025] Пономарев А.В., Агафонов А.А. Аналитический обзор методов распределения задач при совместной работе человека и модели ИИ // Информатика и автоматизация. – 2025. – № 1(24). – С. 229-274.
- [Alves et al., 2024] Alves J.V. et al. Cost-Sensitive Learning to Defer to Multiple Experts with Workload Constraints. *Transactions on Machine Learning Research*. – 2024.
- [Cordelia et al., 1995] Cordelia L.P. et al. A Method for Improving Classification Reliability of Multilayer Perceptrons // *IEEE Transactions on Neural Networks*. – 1995. – No. 5(6). – P. 1140-1147. – doi: 10.1109/72.410358.
- [De et al., 2020] De A., Koley P., Ganguly N., Gomez-Rodriguez M. Regression under Human Assistance // In: *Proc. of the AAAI Conference on Artificial Intelligence*. – 2020. – 34(03). – P. 2611-2620. – doi: 10.1609/aaai.v34i03.5645.
- [De et al., 2021] De A., Okati N., Zarezade A., Gomez-Rodriguez M. Classification Under Human Assistance // In: *Proc. of the AAAI Conference on Artificial Intelligence*. – 2021. – 35. – P. 5905-5913. – doi: 10.1609/aaai.v35i7.16738.
- [Gal et al., 2016] Gal Y., Ghahramani Z. Dropout as a Bayesian approximation: Representing model uncertainty in deep learning // In: *33rd International Conference on Machine Learning (ICML 2016)*. – 2016. – P. 1050-1059.
- [Krizhevsky, 2009] Krizhevsky A. Learning Multiple Layers of Features from Tiny Images. Science Department, University of Toronto, Tech. Report. – 2009.
- [Lakshminarayanan et al., 2017] Lakshminarayanan B., Pritzel A., Blundell C. Simple and scalable predictive uncertainty estimation using deep ensembles // In: *Advances in Neural Information Processing Systems*. – 2017. – P. 6405-6416.
- [Leitão et al., 2022] Leitão D., Saleiro P., Figueiredo M., Bizarro P. Human-AI Collaboration in Decision-Making: Beyond Learning to Defer. In: *ICML 2022, Workshop on Human-Machine Collaboration and Teaming, 2022*. ArXiv: arXiv:2206.13202.

- [Madras et al., 2018]** Madras D., Pitassi T., Zemel R. Predict Responsibly: Improving Fairness and Accuracy by Learning to Defer // In: 32nd Conference on Neural Information Processing Systems. – 2018. – P. 6150-6160.
- [Mozannar et al., 2023]** Mozannar H. et al. Who Should Predict? Exact Algorithms For Learning to Defer to Humans // In: Proceedings of Machine Learning Research. – 2023. – 206. – P. 10520-10545.
- [Mozannar et al., 2020]** Mozannar H., Sontag D. Consistent estimators for learning to defer to an expert // In: Proc. of the 37th International Conference on Machine Learning. – 2020. – P. 7076-7087.
- [Peterson et al., 2019]** Peterson J. et al. Human uncertainty makes classification more robust // In: Proc. of the IEEE International Conference on Computer Visio. – 2019. – P. 9616-9625.
- [Verma et al., 2022]** Verma R., Nalisnick E. Calibrated Learning to Defer with One-vs-All Classifiers // In: Proc. of the 39th International Conference on Machine Learning. – 2022. – PMLR 162:22184-22202.
- [Wilder et al., 2020]** Wilder B., Horvitz E., Kamar E. Learning to Complement Humans // In: Proc. of the 29th International Joint Conference on Artificial Intelligence (IJCAI-20). – 2020. – P. 1526-1533.