АЛГОРИТМ ОТБОРА ПОКАЗАТЕЛЕЙ ДЛЯ ПОСТРОЕНИЯ ДВУХЪЯРУСНОГО ДЕРЕВА РЕШЕНИЙ В ЗАДАЧАХ ОБЪЯСНЯЕМОГО ИСКУССТВЕННОГО ИНТЕЛЛЕКТА

Салтыков С.А.

Институт проблем управления им. В.А. Трапезникова РАН,
Россия, г. Москва, ул. Профсоюзная, д.65
ssaltykov@mail.ru

Аннотация: В работе показано, что традиционные подходы к отбору показателей могут привести к обучению двухъярусного дерева решений, существенно проигрывающего в точности по сравнению с одной из предлагаемых альтернатив — перебором всех непустых подмножеств показателей, содержащих не более, чем три показателя. Разработан алгоритм отбора показателей, позволяющий строить такие же по точности двухъярусные деревья решений, как и при полном переборе подмножеств показателей, но за меньшее время. Доказано утверждение, обосновывающее, почему исключение из перебора некоторых подмножеств показателей в предложенном алгоритме не приводит к снижению точности обучаемого двухъярусного дерева.

Ключевые слова: xAI, объясняемый искусственный интеллект, деревья принятия решений, отбор показателей.

Введение

На практике всё чаще возникают задачи, в которых необходимо получить небольшую, «прозрачную» для лица, принимающего решения, обученную модель (как, например, описано в работе [1]). Часто такими моделями оказываются деревья небольшого размера, иногда двухъярусные деревья решений. Последние мы и будет рассматривать в данной работе. Более внимательный взгляд на такие модели выявляет, что существенную роль для них играют так называемые комплементарные показатели [2]. Так получается и в данном исследовании.

В данной работе будет показано, что если использовать «стандартные», то есть наиболее распространенные методы отбора признаков, или не делать отбор признаков вообще, то точность дерева принятия решения для xAI может быть существенно ниже, чем если для отбора признаков использовать полный перебор сочетаний небольшого числа признаков.

В данной работе показано, что показатели могут быть комплементарны друг другу в том смысле, что они являются слабыми предсказателями целевой переменной по отдельности, но сильными, если используются вместе (в одной цепи дерева решений). Другими словами, есть синергетический эффект от их совместного использования при предсказании. Тогда возможен феномен «экранирования» этой пары показателей другим показателем средней силы. Он на каждом шаге построения дерева решений классическими методами (CART, например) будет помещаться в дерево решений вместо любого из показателей из комплементарной пары, не давая паре показателей оказаться в одной цепи дерева и таким образом проявить свой совместный синергетический эффект.

Работа построена следующим образом. Сначала раскрывается специфика задач объясняемого искусственного интеллекта. Затем анализируются имеющиеся, главным образом, самые популярные подходы к отбору показателей, области применения этих подходов, а также когда вообще стоит использовать хотя бы какие-то методы отбора показателей. После этого на реальном датасете приводятся иллюстративные примеры того, что использование наиболее часто используемых подходов к отбору показателей может приводить к обучению двухъярусных деревьев решений, неоптимальных по точности. Затем будет предложена естественная альтернатива часто используемым методам отбора показателей и показано, что хотя она имеет узкое применение, но именно для задач объясняемого искусственного интеллекта её целесообразно использовать. После этого предложена система определений, позволяющая сформулировать теорему, показывающую какие из наборов показателей можно не рассматривать при отборе показателей. На основании этой теоремы разработан и описан алгоритм отбора показателей для построения двухъярусного дерева решений. В заключении сформулированы главные результаты статьи, а также намечены направления дальнейшего развития данного исследования.

1 Методологические пояснения и допущения данного исследования

Дополнительно стоит отметить, что в общем случае уменьшение числа показателей в датасете до необходимого называется задачей снижения размерности. Методы снижения размерности делятся на два больших подкласса — отбор показателей и выделение показателей. И в общем случае если разрабатывается новый метод из какого-либо из двух этих подклассов, то он должен соревноваться со всеми методами снижения размерности. Однако, в нашем случае из-за специфики задач объясняемого

искусственного интеллекта мы можем не сравнивать разрабатываемый нами в данной работе алгоритм отбора показателей с методами выделения показателей и вот почему. «Прозрачность» обученной модели для лица, принимающего решения (ЛПР), подразумевает, среди прочего, не только то, что в модели задействованы небольшое число показателей, но и то, что каждый из таких показателей понятен ЛПРу, что его интерпретация не вызовет у него вопросов. А методы выделения показателей тем или иным образом формируют новые показатели, которые часто представляют собой некую линейную комбинацию уже имеющихся показателей. Содержательная интерпретация такого нового выделенного показателя будет проблематичной, неочевидной для лица, принимающего решения, поэтому полученная обученная модель не будет для него «прозрачной». Поэтому мы будем считать, что разработанный алгоритм отбора показателей не обязательно сопоставлять по точности построенных на этих показателях моделей с методами извлечения показателей. Однако, в дальнейших исследованиях, особенно, если окажется, что комплементарность показателей окажется полезной не только в сфере объясняемого искусственного интеллекта, сопоставить методы отбора показателей с методами выделения показателей всё же придется.

На первый взгляд кажется естественным, что добавление показателя в датасет не может снизить точность модели, которая строится на этом датасете. И действительно, если добавленный показатель является неинформативным, в обученную модель он просто не попадет. Поэтому кажется, что добавление показателя в датасет должно или увеличивать точность обученной модели, или никак на нее не влиять. Единственной возможностью для уменьшения точности при добавлении нового показателя в датасет связана с увеличением переобучения. И действительно, в общем случае увеличение число столбцов датасета увеличивает степень переобучения, а увеличение числа строк - уменьшает её. Однако, для задач xAI это представляется весьма не существенным, так как число различных значений показателей в датасете на много порядков обычно превосходит число свободных параметров модели, и поэтому переобучение почти не возможно. Действительно, переобучение возникает тогда, когда модель выучивает датасет, и закономерности растворяются в случайном шуме, отсутствующем в генеральной совокупности. А запомнить, выучить большой датасет тремя параметрами двухъярусного дерева точно не получится. Поэтому для таких задач переобучение играет крайне малую роль и, соответственно, добавление показателя если и может уменьшить точность из-за переобучения, то на значения, несущественные для практики. И этим можно пренебречь. Поэтому и столь контринтуитивно уменьшение точности при добавлении показателей, показанное в статье.

Отдельно стоит отметить, что в иллюстративных примерах на реальном датасете, приведенном в статье, мы не производим разбиение датасета на обучающую и тестовую выборку. И, соответственно, не повторяем процедуру такого разбиения несколько раз различным образом, иными словами, не проводим кроссвалиадацию. Это вызвано следующим. Как уже отмечалось, специфика объясняемого искусственного интеллекта как области исследования состоит в том, что обучаемые модели имеют мало свободных параметров по сравнению с числом различных значений показателей в датасете, поэтому для этого класса задач эффекта от переобучения если и есть, то крайне незначительные. Поэтому чтобы не снижать иллюстративную силу приводимым примеров, кроссвалидация не используется. Эффекты, описанные в данной работе, вероятнее всего, будет наличествовать и при использовании кроссвалидации. Тем не менее, это нужно еще строго экспериментально подтвердить, что и будет сделано в последующих работах.

2 Анализ существующих подходов к отбору показателей

Цели отбора показателей везде представляют следующим образом. Он помогает экономить вычислительные ресурсы — процессорное время и оперативную память — за счет удаления неинформативных и дублирующихся признаков. То есть вычислительные ресурсы можем сэкономить существенно, а точность, если и упадет из-за снижения числа признаков, то незначительно. А может даже и чуток возрасти из-за уменьшения эффекта переобучения. Из этого дата-аналитик может сделать вывод, что если он решает задачу в рамках xAI и, как следствие, параметров у модели мало, а обучающая выборка достаточно большая и чистая, то переобучения возникнуть не должно и, следовательно, отбор показателей может дать выгоды только по экономии вычислительеых ресурсов. И при том даже чуток снизить точность.

Но специфика xAI такова, что часто выборки большие, но не гигантские, поэтому потребности в экономии вычислительных ресурсов нету, а точность из-за потери информативности можно потерять, так зачем же вообще использовать хоть какой-либо отбор показателей?

То есть из описаний области применения методов отбора показателей не следует мысль, что используя какой-либо из них — приближенный или точный — можно не только не уменьшить точность, но и существенно — в разы и на порядки — увеличить точность. Поэтому большинство аналитиков для задач xAI скорее всего вообще не будут использовать хотя бы какие алгоримы отбора показателей.

А если и будут использвать, то самые популярные, общеупотребимые, и в данной работе показано, что часто это приводит к потере точности.

Опишем пример того, как использование наиболее известных и частоприменяемых методов отбора показателей, а также неиспользование отбора показателей вообще приводит к потере точности. То есть приводит к построению двухъярусного дерева решений, которое менее точное, чем то, которое бы мы могли построить, если использовали бы только некоторое подмножество показателей.

Итак, возьмем известный открытый публичный датасет — так называемый «Бостонский датасет» [3]. Он содержит 506 экземпляров и 14 показателей, включая целевой показатель — медианную стоимость дома в одном из 506 районов Бостона, характеризующихся 13 показателями, по которым эту стоимость можно предсказать. Возьмем подмножество этого датасета. Во-первых, из всех показателей оставим лишь четыре, кроме целевого, — *RM, TAX, CRIM* и *AGE*. Во-вторых, оставим в датасете только экземпляры, соответствующие районам с домами со «средним» числом комнат: то есть когда $5 \leq RM < 6{,}7$. Этом условию удовлетворяют 379 экземпляров. Итак, мы получили датасет в котором 379 строк и 5 столбцов, один из которых целевая переменная. На таком датасете построим двухъярусное бинарное дерево решений четырьмя разными способами и сравним точности этих деревьев.

Сначала мы построим двухъярусное дерево решений без какого-то ни было отбора показателей. Получится дерево, изображенное на рис. 1. Дерево окажется построенным на двух последних показателях *CRIM* и *AGE* (на рисунке они обозначены f_3 и f_4, соответственно). Оно может объяснить 35,6% дисперсии целевой переменной.

Вторым способом построения дерева будет такой. Используя метод отбора показателей — обратный последовательный отбор (sequential backward selection — SBS [4]), выберем два наилучших показателя и затем на них построим дерево решений.

Работа этого метода состоит в следующем. На первом этапе из датасета последовательно по одному удаляются каждый из показателей. И обучается модель на таких датасетах без одного показателя. В итоге принимается решение удалить тот показатель, удаление которого менее всего снижает точность обученной модели. Если нужно отобрать меньшее число показателей, чем осталось в датасете, то такая операция проделывается снова. И так до тех пора, пока не останется необходимое число показателей.

Такими отобранными показателями окажутся те же два последних показателя, следовательно, и построенное таким способом дерево окажется таким же (рис. 1).

В рамках третьего способа построения дерева решений сначала используем метод отбора показателей, использующий важности показателей, для отбора двух наилучших показателей, а затем на них построим дерево решений. В этом методе на данном датасете обучается случайный лес [5]. И для всех деревьев леса считается насколько в среднем оказывается эффективными сплиты по некоторому показателю. в том смысле, что насколько сплит позволяет уменьшает разброс, неопределенность целевой переменной.

Вектор важностей четырых показателей у нас получился таким (0.19761737, 0.04306606, 0.54740128, 0.21191529). По нему видно, что два самых важных показателя опять же два последних *CRIM* (f_3) и *AGE* (f_4). При использовании разных гиперпараметров в методе случайного леса вектор важностей показателей может оказываться несколько различным, и, соответственно, могут оказаться самыми важными не всегда именно эти показатели. Однако здесь важно отметить два момента. Во-первых, и это самое главное, существуют такие гиперпараметры при которых самыми важными оказываются не первые два показателя. И во-вторых, перебор различных гиперпараметров показывает, что последние два показателя часто оказываются самыми весомыми, а первые два — практически никогда. Здесь требуются дополнительные исследования, чтобы определить, как часто это происходит, уже ясно, что существуют примеры, когда метод отбора показателей, основанный на построении случайного леса, «проскакивает» оптимальное дерево решений. Следовательно, при использовании этого третьего способа построения дерева решений вновь у нас получится дерево, изображенное на рис. 1.

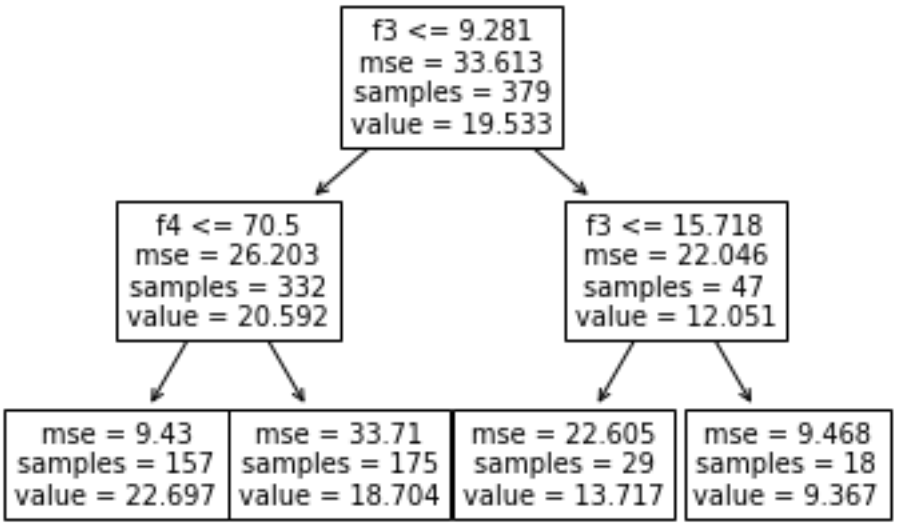

Рис. 1. Дерево решений, построенное «стандартными» методами

Таким образом, в данном примере использование самых общеупотребимых методов отбора показателей приводит к построению дерева на основе показателей *CRIM* и *AGE*, позволяющему объяснить 35,6% дисперсии целевой переменной. Но оптимально ли отработали эти методы отбора показателей? Существует ли такое двухэлементное подмножество имеющегося четырехэлементного множества показателей, на которых построенное двухъярусное дерево решений давало бы большую точность? Ответить на этом вопрос легко, перебрав все сочетания по два показателя из четвырех имеющихся, и окажется, что если строить дерево решений на первых двух показателях — *RM* и *TAX* (на рисунке f_1 и f_2, соответственно), — то получится дерево, представленное на рис. 2 и оно уже позволяет объяснить 38,8%, что на 9% больше, чем дерево на рис. 1.

Это позволяет сделать вывод о том, что имеющиеся наиболее часто используемые методы отбора показателей иногда отбирают неоптимальный набор показателей. Как часто это происходит и насколько этот набор оказывается неоптимальным, это предмет отдельного исследования. В рамках данной работы важно отметить, что такое принципиально может происходить и понять, почему такое происходит.

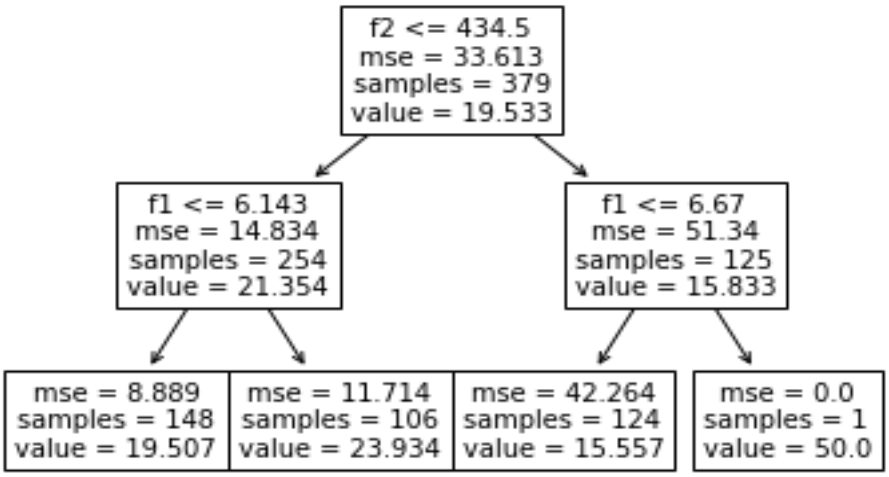

Рис. 2. Дерево решений, построенное предагаемым методом

Более внимательный взгляд на методы отбора показателей показывает, что такие методы являются приближенными, и поэтому иногда они достигают не оптимального, а субоптимального (в случае использования «жадного» алгоритма) набора показателей или лишь близкого к оптимальному. По-

видимому, это происходит из-за того, что точный метод — это полный перебор всех подмножеств некоторого множества показателей, а он экспоненциально сложен. И действительно число подмножеств показателей множество показателей из N элементов равно 2^N. Поэтому уход от полного перебора важен, поэтому и были разработаны приближенные, «жадные» алгоритмы. Но это имеет смысл не для всех классов задач. Например, в объясняемом искусственном интеллекте — xAI — число параметров модели часто очень небольшое, а перебрать все подмножества, содержащие небольшое число элементов, может быть уже вполне посильная задача.

Более того, для задачи построения двухъярусного дерева всё еще более упрощается, так как у нем только три сплита, поэтому нужно перебрать все подмножества, содержащие не более, чем три показателя. А это уже полиномиальная задача невысокой размерности. Таким образом, получается, что специфика xAI позволяет использовать для некоторых задач точные, полнопереборные методы отбора показателей и тем самым повышать точность деревьев решений. Но можно ли для построения двухъярусных деревьев решений сконструировать такой алгоритм отбора показателей, который находил бы оптимальный набор из не более, чем трех показателей, но при этом в общем случае перебирал бы меньше вариантов, чем полнопереборный метод? В следующем разделе сконструируем такой алгоритм.

3 Алгоритм отбора показателей для построения двухъярусного дерева решений

Два показателя на данной выборке будем назвать комплементарными друг другу по отношению к третьему показателю, если дерево решений, построенное на комплементарных показателях оказывается более точным, чем построенное на трех показателях.

В этом случае третий показатель будем называть экранирующим.

Цель 1. Посчитать, насколько большим может быть эффект на реальных датасетах от выкидывания (учета) экранирующих показателей. На синтетическом датасете уже показано, что такой эффект может быть велик, но как дела обстоят на реальных данных?

Цель 2. Проверить, имеет ли смысл накапливать информацию (transfer learning) или можно просто выкидывать показатели наугад, и это позволит приемлемо избавиться от эффекта экранирования.

Из общих соображений ясно, что когда мы обучаемся на шумных и/или малых датасетах, то на обучающей выборке может казаться, что отбросить данный показатель это хорошо, но на тестовой выборке может оказаться, что это не так. Сохранение, аккумулирование более тщательно полученных данных может добавить робастности, а как следствие, точности. Но как велика будет эта прибавка в точности? Стоит ли принимать во внимание эту накопленную информацию или на нее вообще можно не ориентироваться?

4 Основные дефиниции. Комплементарная двойка

Будем говорить, что сплит $S(f, t)$ представляет собой нелистовую вершину дерева решений, в которой указано условие, что значение показателя f больше некоторого граничного значения t. Если это условие выполняется для некоторого вектора значений показателей, то при поиске решений согласно этому дереву идем из данной вершины по одному ребру к следующей вершине, если невыполняется, то к другому. В этом случае для краткости будем говорить, что имеем сплит S по показателю f.

Двухъярусное дерево решений будем представлять как кортеж $T = (S_1(f_1, t_1), (S_2(f_2, t_2), S_3(f_3, t_3)))$.

В этом случае будем говорить, что дерево T содержит сплиты по показателям f_1, f_2, f_3. Или еще короче, что дерево T содержит показатели f_1, f_2, f_3, построено на этих показателях. Также будем говорить, что набор данных есть подмножество некоторого набора данных на некоторых показателях, если это подмножество содержит только эти показатели.

Определение 1. Точностью некоторого непустого множества показателей на данном наборе данных назовем точность (определяемую некоторой метрикой, например, долей объясненной дисперсии) двухъярусного дерева принятия решений, построенного на подмножестве данного набора данных, содержащего только данные показатели.

Определение 2. Пару показателей будем называть комплементарной, если точность этой пары показателей будет строго больше точности каждого из показателей по отдельности.

Определение 3. Ядро показателей для двухярусного дерева это такое минимальное подмножество показателей, точность на которых максимальна.

Теорема 1. Если ядро содержит два показателя, то они комлементарны между собой.

Доказательство. Предположим противное: дерево решений построено на двух различных между собой показателях, и эти два показателя не является комплементарными. Тогда по определению

комплементарности это дерево не будет превышать по точности деревья, построенные на каждом из показателей по отдельности, тогда ядро — минимальный набор показателей, обеспечивающий максимальную точность, — будет состоять из одного показателя. А это противоречит тому, что ядро состоит из двух показателей. Пришли к противоречию, значит исходное предположение неверно, и два показателя, входящих в ядро, комплементарны. Что и требовалось доказать.

Теорема 2. Если ядро содержит три показателя, то в ядре есть, по крайней мере, одна пара комплементарных показателей.

Доказательство. Предположим противное: ядро из трех показателей не содержит ни одной пары комплементарных показателей. Тогда в любой из цепей дерева можно два показателя заменить на любой один показатель, и при этом новое, полученное дерево, будет обладать не большей точностью. Следовательно, так мы получим дерево из двух показателей, обладающее не меньшей точностью, значит, по определению ядро содержит менее, чем три показателя, что противоречит исходному предположению. Пришли к противоречию, значит исходное предположение неверно, и в ядре из трех показателей по крайней мере два комплементарны. Что и требовалось доказать.

Определение 4. Экранирующий показатель это тот, добавление которого в датасет приводит к уменьшению точности.

Введенные определения и доказанные утверждения позволяют сконструировать Алгоритм 1, в котором в общем случае перебираются не все подмножества показателей, не превышающие три показателя, но при этом алгоритм гарантировано отбирает оптимальный набор показателей, то есть ядро. Опишем его. Сначала определяется точность каждого из показателей (в смысле определения, данного выше). Затем определяем точность каждой из пар показателей (сочетаний по два показателя). На основании этой информации мы можем определить, какие из пар показателей являются комплементарными. После этого мы берем лишь комплементарные двойки и к какой из таких двоек добавляем по очереди какой из показателей и определяем точность полученной тройки. Каждый раз, когда мы считали точность для какого набора показателей, мы мы сохраняли упорядоченную пару (кортеж), состоящий из этого набора показателей и построенного по этих показателях двухъярусного дерева. Так как по определению, функция есть набор упорядоченных пар, то такую сохраненную информацию можно трактовать как функцию *FtoT* из множества всех наборов показателей во множество двухъярусных деревьев. Затем мы ищем такие наборы показателей, на которых строятся самые точные деревья. Мы предполагаем, что в общем случае нескольким разным наборам показателей соответствуют деревья одинаковой максимальной точности. После этого мы для каждого такого набора сохраняем размер этого набора в виде кортежа. И получаем новую функцию *FtoN* из множества всех наборов показателей во множество натуральных чисел.

Алгоритм 1. Отбор показателй для обучения двухъярусного дерева решений.

FtoT $\leftarrow \varnothing$;

foreach $f \in F$ do

  $t \leftarrow \mathrm{CART}_D(\{f\})$;

  $FtoT \leftarrow FtoT \cup \{(\{f\}, t)\}$;

end

$C_{F,2} \leftarrow$ getCombinations(F, 2);

foreach $Comb_f \in C_{F,2}$ **do**

  $t \leftarrow \mathrm{CART}_D(Comb_f)$;

  $FtoT \leftarrow FtoT \cup \{(Comb_f, t)\}$;

end

$F_{Comp,2} \leftarrow$ getComplementaryPairs($FtoT$, 2);

foreach $Comp_f \in F_{Comp,2}$ **do**

  foreach $f \in F$ do

    $t \leftarrow \mathrm{CART}_D(Comp_f \cup \{f\})$;

    $FtoT \leftarrow FtoT \cup \{(Comp_f \cup \{f\}, t)\}$;

  end

end

$FM \leftarrow \arg\max_H \mathrm{Score}(FtoT(H))$;

FtoN ← ∅;

foreach $F^* \in FM$ do

FtoN ← FtoN ∪ {F^*, $|F^*|$};

end

$F_{comp} \leftarrow \arg\min_{FM} FtoN(FM)$;

В конце алгоритм ищет набор показателей (среди тех, которые обеспечивают наибольшую точность) наименьшего размера. В итоге получаем ядро показателей — минимальный набор показателей, обеспечивающий максимальную точность.

Проанализируем приведенный алгоритм. Алгоритм достигает ядра: при переборе наборов показателей, он не «пропустит» ядро, так как согласно теореме 1, ядро, содержащее три показателя, обязательно содержит хотя бы одну комплементарную пару показателей. Алгоритм в общем случае быстрее достигает цели по сравнению с полным перебором всех подмножеств показателей, содержащих как минимум три показаталя, так как в общем случае существуют некомплементарны пары показателей, и они в переборе не участвуют, тем самым экономится время.

Другими словами, тройки попарно некомплементарных показателей не образуют ядро показателей, поэтому такие тройки можно не перебирать в процедуре отбора показателей без потери точности. Исключение таких троек из перебора повышает быстродействие алгоритма при прочих равных.

Заключение

Таким образом, в данной работе разработан алгоритм отбора показателей, подходящий для целей объясняемого искусственного интеллекта, а конкретно для построения двухъярусного дерева решений. Во-первых, этот алгоритм в общем случае позволяет отобрать показатели, на которых строится более точное дерево решений. А во-вторых, этот отбор осуществляется алгоритмом быстрее, чем сплошной перебор всех подмножеств показателей, состоящих из не более, чем трех элементов.

Каким же видится естественное продолжение проведенного исследования? Во-первых, нужно определить, как часто различные наиболее часто используемые методы отбора показателей отбирают неоптимальный набор показателей, и когда это происходит, насколько велика потеря в точности? Каково распределение этой потери в точности? И как часто встречаются ситуации, когда потеря в точности будет, скажем, от 10% и выше? А может ли на реальных датасетах доля объясненной дисперсии уменьшится в разы при отборе неоптимального набора показателей и если да, то как часто такое может случаться?

Во-вторых, стоит иметь в виду, что минимальный набор показателей, обеспечивающих максимальную точность (то есть, ядро показателей в терминах этой статьи), определяется для конкретного датасета, содержащего экземпляры, определенными значениями показателей. Например, в данной работе приводился пример районов Бостона со «среднекомнатными» домами. В общем случае у разных подмножеств этого датасета могут быть различные ядра показателей. Как искать такие подмножества некоторого датасета, обладающие разными ядрами показателей? Насколько такая информация поможет увеличить точность обученных моделей?

Наконец, в-третьих, по-видимому, собранная информация о парах комплементарных показателей может быть полезной не только для построения двухъярусных деревьев, но и для построения деревьев любой глубины и даже для ансамблей деревьев. И действительно, когда два яруса дерева построены, мы оказываемся перед такой же задачей: каждой терминальной вершине соответствует некоторое подмножество исходного датасета, и для этого подмножества нужно подобрать подходящие, приводящие в оптимальному результату, сплиты. И для этого подмножества, соответствующего терминальной вершине, тоже может существовать эффект экранирования комплементарности показателей. Причем комплементарные показатели для этого подмножества датасета могут быть совсем не те же, что для всего датасета в целом. Но если эти комплементарные показатели для подмножества датасета известны и учтены, мы можем получить еще больший прирост точности по сравнению с учетом лишь комплементарных показателей для всего датасета. И этот прирост точности будет получен на дереве решений произвольной глубины.

Но всё это уже предмет дальнейших исследований.

Литература

1. *Салтыков, С. А.* Корреляция наукометрических показателей из РИНЦ с цитированием по базе web of science / С. А. Салтыков // Управление развитием крупномасштабных систем mlsd'2020 : ТРУДЫ ТРИНАДЦАТОЙ МЕЖДУНАРОДНОЙ КОНФЕРЕНЦИИ, Москва, 28–30 сентября 2020 года / Под общей редакцией С.Н. Васильева, А.Д. Цвиркуна. – Москва: Институт проблем управления им. В.А. Трапезникова РАН, 2020. – С. 1677-1684. – DOI 10.25728/mlsd.2020.1677.
2. *Saltykov, S.* Algorithm of Building Regression Decision Tree Using Complementary Features / S. Saltykov // Proceedings of 2020 13th International Conference Management of Large-Scale System Development, MLSD 2020 : 13, Moscow, 28–30 сентября 2020 года. – Moscow, 2020. – P. 9247785. – DOI 10.1109/MLSD49919.2020.9247785.
3. «Бостонский» датасет. URL: https://www.cs.toronto.edu/~delve/data/boston/bostonDetail.html (дата обращения: 30.05.2020).
4. *Рашка С.* Python и машинное обучение / пер. с англ. А. В. Логунова. - М.: ДМК Пресс, 2017.-418 с.: ил., ISBN 978-5-97060-409-0
5. *Ho T. K.* Random decision forests // Proceedings of 3rd international conference on document analysis and recognition. – IEEE, 1995. – Т. 1. – С. 278-282.