УДК 004.855.5 DOI: 10.14529/ctcr150304

ОСОБЕННОСТИ ПРИМЕНЕНИЯ ДЕРЕВЬЕВ РЕШЕНИЙ В ЗАДАЧАХ КЛАССИФИКАЦИИ

И.Л. Кафтанников, А.В. Парасич
Южно-Уральский государственный университет, г. Челябинск

Рассматривается применение деревьев решений в задачах классификации. В последние годы деревья решений широко применяются в задачах компьютерного зрения, таких как распознавание объектов, классификация текстов, распознавание жестов, обнаружение спама, обучение ранжированию в информационном поиске, семантическая сегментация и кластеризация данных. Этому способствуют такие отличительные особенности деревьев решений как интерпретируемость, управляемость, возможность автоматического отбора информативных признаков. Однако имеется и ряд принципиальных недостатков, из-за которых задача обучения деревьев решений существенно усложняется. В статье приводится анализ преимуществ и недостатков деревьев решений, рассматриваются вопросы обучения и тестирования деревьев решений. Особое внимание уделяется проблемам сбалансированности обучающей выборки. Рассматриваются также леса решений и методы их обучения. Приводится краткий обзор методов снижения взаимозависимости ошибок деревьев решений при обучении лесов решений. Предлагаются методы преодоления недостатков деревьев решений, приводятся результаты работы данных методов.

Ключевые слова: деревья решений, леса решений, машинное обучение, классификация.

Введение

Деревья решений применяются в задачах классификации (принятие решения о принадлежности объекта к одному M из непересекающихся классов) и регрессии (предсказание значения из непрерывного диапазона). Классификация и регрессия на основе деревьев решений используются в задачах распознавания текста, информационного поиска, распознавания речи, анализе изображений, обнаружении спама, распознавания жестов и др. Для конструирования деревьев решений применяется *машинное обучение* – автоматическая настройка параметров алгоритма на основе *обучающей выборки* (множества объектов с известными правильными ответами). При этом от качества обучения зависит правильность решения задачи и практическая применимость результатов.

Под *алгоритмом* будем понимать функцию, принимающую на вход классифицируемый объект и возвращающую один из M классов – ответ алгоритма для данного объекта. Деревья решений состоят из вершин, в которых записаны проверяемые условия (будем называть эти условия *признаками*), и листьев, в которых записаны ответы дерева (один из M классов для задачи классификации). Под *обучающим примером* будем понимать объект обучающей выборки с известным правильным ответом (классом, к которому принадлежит данный объект). Обучение состоит в настройке условий в узлах дерева и ответов в его листьях с целью достижения максимального качества классификации.

Пусть заданы конечное множество объектов $X = \{x_1, \cdots, x_L\}$ и алгоритмов $A = \{a_1, \cdots, a_D\}$ и бинарная функция потерь $I: A \times X \to \{0, 1\}$, $I(a, x) = 1$ тогда и только тогда, когда алгоритм допускает ошибку на объекте x. Число ошибок алгоритма a на выборке X определяется как $n(a, X) = \sum_{x \in X} I(a, x)$. Частота ошибок алгоритма на выборке определяется как $v(a, X) = n(a, x)/|X|$. Под *качеством классификации* понимается частота ошибок алгоритма на контрольной выборке.

1. Преимущества и недостатки деревьев решений

Автоматический отбор признаков. Признаки в вершины дерева выбираются автоматически из набора признаков. Поэтому можно составить произвольный набор признаков, а в процессе обучения автоматически выберутся информативные и проигнорируются неинформативные при-

знаки. Нет необходимости в дополнительной процедуре отбора признаков, в отличие от других методов машинного обучения.

Интерпретируемость. Деревья решений позволяют строить решающие правила в форме, понятной эксперту. Это оказывается полезным в том случае, когда человеку требуется понимать, каким образом алгоритм будет принимать решения. Интерпретируемость также оказывается полезным свойством, если требуется понять, почему дерево решений работает неправильно.

Управляемость. Если некоторые примеры классифицируются неправильно, можно заново обучить только те вершины дерева, из-за которых это происходит, что очень удобно, когда объем обучающих данных большой и обучение занимает много времени. Кроме того, при тренировке разных поддеревьев могут оказаться более эффективными разные алгоритмы обучения. Обучение заново только части дерева позволяет изменить результат классификации одних объектов, не затрагивая классификацию других объектов.

Сильная зависимость от сбалансированности числа обучающих примеров разных классов. При обучении деревьев решений в вершину выбирается признак с максимальной информативностью порождаемого этим признаком разбиения тренировочных примеров на две части. Информативность разбиения вычисляется по формуле

$$I = \frac{|L|}{|L|+|R|} \cdot H(L) + \frac{|R|}{|L|+|R|} \cdot H(R),$$

где L и R – множества обучающих примеров, попадающих в левый и правый дочерние узлы в зависимости от функции разбиения, $H(L)$ и $H(R)$ – оценка информативности подмножества обучающих данных, в качестве которой может использоваться энтропия Шеннона или индекс Гини.

Рассмотрим особенности обучения деревьев решений на примере индекса Гини. Формула для расчета индекса Гини представлена ниже:

$$H(S) = 1 - \sum_{k=1}^{K} p^2(k|S), \quad (1)$$

где K – число классов, а $p(k|S)$ – доля примеров класса k в множестве S.

Допустим, что в процессе обучения одной из вершин дерева существуют признаки $f0$, $f1$ и $f2$, которые порождают одинаковое разбиение множества данных S этой вершины на подмножества L и R (при этом для некоторых классов a и b $p(a|S) \gg p(b|S)$, кроме того, $p(a|L) \gg p(a|R)$ и $p(b|L) = p(b|R)$, то есть данный признак является неинформативным по отношению к классу b), за исключением того, что для признака $f1$ число примеров класса a $n_{f1}(a|L) = n_{f0}(a|L) - 1$, $n_{f1}(a|R) = n_{f0}(a|R) + 1$, а для признака $f2$ $n_{f2}(b|L) = n_{f0}(b|L) - 1$, $n_{f2}(b|R) = n_{f0}(b|R) + 1$. Из формулы (1) видно, что увеличение информативности по сравнению с информативностью разбиения по признаку $f0$ в случае с признаком $f1$ будет гораздо больше, чем случае с признаком $f2$. Допустим, мы можем выбрать признак $f3$ таким образом, что $n_{f3}(b|L) = 0$, $n_{f3}(b|R) = n(b|S)$, а $n_{f3}(a|L) = n_{f1}(a|L) = n_{f0}(a|L) - 1$, $n_{f3}(a|R) = n_{f1}(a|R) = n_{f0}(a|R) + 1$, то есть признак $f3$ позволяет правильно распознать все примеры класса b, но один пример класса a попадет в меньшее по размеру подмножество по сравнению с признаком $f0$. Однако информативность признака $f0$ будет выше информативности признака $f3$, если $(n_{f0}(a|L))^2 - (n_{f0}(a|L) - 1)^2 > (n(b|S))^2 - 2(n(b|S) / 2)^2$, или $2n_{f0}(a|L) - 1 > n^2(b|S) / 2$, что будет выполняться при условии $p(a|S) \gg p(b|S)$.

Таким образом, при обучении дерево уделяет повышенное внимание классам с большим числом обучающих примеров, и может полностью проигнорировать классы с малым числом обучающих примеров. Поэтому сбалансированность обучающего множества при обучении деревьев очень важна. Это может являться как преимуществом, так и недостатком. С одной стороны, при неправильных пропорциях классов в обучающей выборке дерево обучится некорректно, с другой стороны, с помощью балансировки тренировочной выборки можно управлять процессом обучения дерева произвольным образом. Если некоторые важные случаи распознаются недостаточно хорошо, можно добавить данные примеры в выборку или поднять их вес, и качество их распознавания улучшится. Однако чтобы повысить качество распознавания в целом таким способом, требуется большой объем ручной настройки. В общем случае рекомендуется поддерживать одинаковое число примеров каждого класса в обучающей выборке.

Требуются специальные методы предотвращения переобучения. Рассмотрим некоторое разбиение множества объектов $X = \{x_1, \cdots, x_L\}$ на две выборки – обучающую X_t длины l и контрольную X_c длины $k = L - l$. Переобученностью алгоритма a называется величина отклонения частоты ошибок на контроле и обучении $q(a, X) = \nu(a, X_c) - \nu(a, X_t)$.

Явление переобучения возникает из-за излишней сложности модели, когда обучающих данных недостаточно для того, чтобы восстановить по ним информативную закономерность. При нехватке тренировочных данных высока вероятность выбрать закономерность, которая выполняется только на этих данных, но не будет верна для других объектов. Для деревьев решений сложность модели – это глубина дерева. Но в разные вершины в процессе обучения попадает разное число тренировочных примеров, из-за чего в разных ветвях оптимальной будет разная глубина дерева. Поэтому для деревьев решений требуются специальные методы контроля переобучения. Обычно применяется *pruning* – удаление тех вершин дерева, которые ухудшают качество классификации данных, не входящих в обучающее множество.

Экспоненциальное уменьшение обучающей выборки. После обучения каждой вершины дерева происходит разделение ее тренировочного множества на два подмножества. Таким образом, на каждом следующем уровне дерева обучающее множество вершины содержит все меньше и меньше примеров. В идеальном случае, если в каждой вершине ее обучающая выборка делится пополам, на уровне дерева h размер выборки в вершине будет в 2^h раз меньше исходного размера выборки. А чем меньше размер обучающего множества, тем выше вероятность переобучения. Поэтому для обучения деревьев решений требуются выборки большого размера. Несколько методов решения данной проблемы будут рассмотрены далее.

Явление разбалансировки. Допустим, имеется признак $f1$, и для некоторой разновидности объектов $z \in X$

$$f1(x) = \begin{cases} 0 \text{ если } y(x) = y_1 \\ 1 \text{ иначе} \end{cases},$$

иными словами, признак $f1$ позволяет определить принадлежность объекта x к классу y для объектов данного типа, однако данная закономерность не выполняется для объектов x, не относящихся к разновидности z. При этом более надёжным признаком для определения принадлежности объекта x к классу y является признак $f2$. Однако по каким-то причинам большое число объектов в обучающей выборке для всего дерева или для отдельной вершины дерева относятся к разновидности z. В таком случае информативность признака $f1$ окажется выше информативности признака $f2$, и признак $f1$ будет использоваться деревом вместо признака $f2$ для принятия решения о принадлежности объекта x к классу y. После этого на качество обучения будут оказывать негативное влияние явления уменьшения размера обучающей выборки, переобучения и сильной зависимости от соотношения числа примеров разных классов в обучающей выборке, рассмотренные выше. На практике высока вероятность того, что разбалансировка обучающего множества существует в некоторой узкой области пространства признаков (например, в задаче классификации изображений в обучающей выборке зелёный цвет фона на всех изображениях машин виден только на ярко освещённых изображениях), или возникает при обучении некоторых вершин дерева. В таком случае обнаружить и устранить проблему будет очень сложно.

2. Леса решений

Для решения проблем переобучения деревьев решений применяются леса решений – несколько деревьев, результат классификации определяется путем голосования (ответом выбирается тот класс, который предсказало наибольшее число деревьев).

Пусть имеется множество деревьев решений, каждое из которых относит объект $x \in X$ к одному из классов $c \in Y$. Будем считать, что если $f_c^t(x) = 1$, то дерево t относит объект $x \in X$ к классу c. При использовании алгоритма простого голосования для каждого класса $c \in Y$ подсчитывается число деревьев, относящих объект к данному классу:

$$G_c(x) = \frac{1}{T_c}\sum_{t=1}^{T_c} f_c^t(x), c \in Y.$$

Ответом леса является тот класс, за который подано наибольшее число голосов:

$$a(x) = \operatorname{argmax}_{c \in Y} G_c(x).$$

Для того чтобы лес решений повышал качество, необходима независимость ошибок деревьев. Если все деревья будут ошибаться на одних и тех же примерах, не будет выигрыша от использования леса. Верхняя граница ошибки обобщения для леса определяется как

$$GE = p\frac{1 - s^2}{s^2},$$

где *GE* – ошибка обобщения, *s* – качество классификации дерева, *p* – средняя попарная корреляция между ошибками деревьев [1].

Если обучать деревья на одном и том же множестве тренировочных примеров одним и тем же методом, получатся одинаковые или очень похожие деревья. Поэтому для достижения независимости ошибок деревьев, составляющих лес решений, требуется применять специальные методы. Рассмотрим некоторые из них.

Метод Bagging [2] – каждое дерево обучается на собственном подмножестве размера l обучающей выборки $D = \{x_i, y_i\}_{i=1}^{l}$, выбранном случайно. Недостаток данного метода – при росте размера обучающей выборки эффект пропадает, так как подвыборки становятся все более похожими (поскольку взяты из одного вероятностного распределения, а влияние случайных отклонений ослабевает).

Метод Boosting [3] – тренировочным примерам назначаются веса $(w_1^1, \dots, w_m^1)$ в зависимости от их сложности. Поскольку веса имеют вероятностную природу, для них выполняется условие $\sum_{j=1}^{m} w_j^1 = 1, w_j^1 \in [0,1]$. Начальное распределение весов выбирается равномерным: $w_j^1 = \frac{1}{m}$, $j \in 1, \dots, m$. Обучается первое дерево, с его помощью производится классификация тренировочных примеров. Веса примеров, классифицированных правильно – снижаются, классифицированных неправильно – повышаются. Следующее дерево леса строится с учетом обновленных весов, и так далее до достижения заданного количества деревьев или требуемой ошибки классификации. Для этого при расчете информативности признака вместо доли объектов $p(k|S)$ класса k в подмножестве *S* используется отношение суммы весов объектов, принадлежащих данному классу, к сумме весов всех объектов подмножества *S:*

$$p(k|S) = \frac{\sum_{i=1}^{|S|} I[y_i=k] \cdot w_i^d}{\sum_{i=1}^{|S|} w_i^d}.$$

ComBoost [4] – после обучения очередного классификатора для каждого тренировочного примера рассчитывается отступ – степень уверенности композиции в классификации данного примера [5]. Объекты со слишком большим отступом считаются выбросами и удаляются из выборки. Далее перебирается нижняя граница отступа. Примеры с отступом ниже минимального считаются простыми для классификации, и не участвуют в обучении нового классификатора. После нескольких итераций выбирается классификатор, добавление которого в лес больше всего снижает ошибку всей композиции на обучающей выборке.

3. Методика тестирования в машинном обучении

Рассмотрим методику тестирования алгоритмов машинного обучения. Качество классификатора можно определить по методу скользящего контроля. Исходное множество примеров $D = \{x_i, y_i\}_{i=1}^{l}$ с известными правильными ответами разбивается на две подвыборки: тренировочную и контрольную. После обучения алгоритма на тренировочной выборке вычисляется число ошибок классификации на контрольной выборке, которое используется как мера качества алгоритма. Для задачи классификации с конечным числом классов $Y = \{y_i\}$ вводится функция ошибки $I(a,x) = [a(x) \neq y(x)]$, где $y(x)$ – класс объекта x, $a(x)$ – ответ алгоритма для объекта x.

Следует отметить, что такого вида оценка качества классификаторов обладает рядом принципиальных недостатков. Подобная методика тестирования не позволяет обнаружить ошибки в формировании обучающей выборки, которые часто оказываются более критичными, чем недостатки алгоритма обучения. Обучающая и контрольная выборка сгенерированы из одного и того же вероятностного распределения, которое не всегда точно отражает истинное распределение. Например, мы хотим обучить алгоритм распознавания цифр, а в выборке все цифры «1» написаны красным цветом, а все цифры «2» написаны синим цветом. Распознавание только на основе цвета даст стопроцентный результат и на обучающей, и на контрольной выборке, однако такой алгоритм не будет работать на других данных. Деление выборки на обучающую и контрольную не позволит обнаружить подобную проблему. Подобного рода ложные зависимости могут присутствовать в обучающих данных в менее явной форме. Поэтому лучше оценивать качество классификации по выборке из независимого источника, или как среднее по нескольким выборкам из разных независимых источников, так как каждая выборка имеет свои особенности формирования. Кроме того, оценка зависит от баланса примеров разного типа в выборке. Например, если

объектов класса A в тестовой выборке в несколько раз больше, чем объектов класса B, то в спорном случае нам выгоднее возвращать в качестве ответа класс A. Однако на реальных данных подобное соотношение частот появления классов может не выполняться. Поэтому, пытаясь улучшить результат классификации на тесте, можно слишком сильно подстроить алгоритм под особенности выборки.

Переобучение на тестовую выборку. Если тестовая выборка фиксирована и не меняется во времени, мы можем поочередно удалять из обучающего множества те примеры, удаление которых не приводит к увеличению числа ошибок на тестовой выборке. При каждом таком удалении качество на тестовой выборке будет или не изменяться, или увеличиваться. Следовательно, таким образом можно улучшить качество на тестовой выборке, но при этом алгоритм перестанет работать на данных, непохожих на тестовые. Если размер тестовой выборки много меньше размера обучающей (что обычно выполняется на практике), то после применения данного алгоритма качество работы на реальных данных может сильно пострадать.

4. Предлагаемые методы

Рассмотрим некоторые возможные методы решения проблемы уменьшения размера обучающей выборки в процессе обучения деревьев решений.

Случайное перемешивание. В процессе обучения дерева после выбора признака в вершине производится разделение множества тренировочных примеров, попавших в данную вершину, на два подмножества. Функция разбиения $S(x, \Theta)$ обычно зависит от двух величин: выбранный признак $\Theta_1 \in \{1, \ldots, M\}$ и порог $\Theta_1 \in R$. В этом случае функция разбиения выглядит так:

$$s(x, \theta) = \begin{cases} 0 \text{ если } x(\Theta_1) < \Theta_2 \\ 1 \text{ иначе} \end{cases}.$$

Примеры x, для которых значение выбранного признака $s(x, \Theta) = 0$, попадают в левое подмножество, а примеры, для которых $s(x, \Theta) = 1$, попадают в правое подмножество. Суть предлагаемого метода состоит в случайном изменении подмножества, в которое попадёт данный пример, с некоторой вероятностью p_m . При этом если подмножество L получилось больше подмножества R по количеству обучающих примеров без учета случайного перемешивания, то математическое ожидание числа примеров, перемещенных из L в R, будет больше математического ожидания числа примеров, перемещенных из R в L. В результате распределение примеров по вершинам получается более равномерным, становится меньше вершин, в которых не хватает обучающих примеров для выбора адекватного признака.

В таблице приведено сравнение результатов, полученных с использованием данного метода при разных значениях вероятности p_m изменения подмножества, и результатов, полученных без использования данного метода. Эксперимент проводился на задаче сегментации органов человека по рентгеновским снимкам. Обучающая и тестовая выборка содержали по 1000 изображений, на каждом примерно по 10 000 классифицируемых пикселей (суммарный объём обучающей выборки составил 9 159 775 обучающих примеров). В качестве признаков использовалось сравнение яркостей пикселей на заданных смещениях относительно классифицируемого пикселя. Видно, что использование предложенного метода позволяет повысить качество классификации, максимальный прирост качества составил 0,73 %. Полученные результаты демонстрируют возможность повышения качества классификации с помощью деревьев решений путем решения проблемы уменьшения размера обучающей выборки. Достоинства предлагаемого метода – простота реализации, а также отсутствие дополнительных расходов времени и памяти на этапе обучения.

Качество классификации на тестовой выборке при разных значениях вероятности p_m изменения подмножества примера

Вероятность изменения подмножества	$p_m = 0$	$p_m = 0{,}1$	$p_m = 0{,}04$	$p_m = 0{,}02$	$p_m = 0{,}01$	$p_m = 0{,}005$
Качество классификации на тестовой выборке, %	67,66	68,24	68,39	68,22	67,96	67,82

Добавление похожих примеров. В процессе обучения дерева в некоторых вершинах нижних уровней может оказаться слишком мало тренировочных примеров, в результате в данных верши-

нах могут быть выбраны неадекватные признаки, может возникнуть переобучение. Логичным выглядит добавлять в обучающее множество таких вершин тренировочные примеры, похожие на примеры, уже находящиеся в данном множестве. Выберем вершины, мощность $|N|$ обучающего множества которых меньше k. Возьмем случайное подмножество v множества признаков $F = \{f_1, \ldots, f_m\}$. Будем добавлять в обучающее множество с некоторой вероятностью p_a те примеры, для которых значение всех признаков из подмножества v совпадает со значением этих признаков для большинства примеров из исходного обучающего множества данной вершины. Недостаток данного метода по сравнению с предыдущим – увеличение объема используемой памяти, что может быть важным при обучении на наборе данных большого размера, а также дополнительные затраты времени на выбор добавляемых в обучающее множество примеров.

Выводы

В статье рассмотрены основные преимущества и недостатки использования деревьев решений в задачах классификации, предложен ряд методов устранения этих недостатков. Приведены результаты, достигнутые с помощью метода случайного перемешивания. Показано, что применение данного метода позволяет повысить качество классификации.

Литература/References

1. Breiman L. Random Forests. *Machine Learning*, 2001, vol. 45(1), pp. 5–32. DOI: 10.1023/A:1010933404324

2. Breiman L. Bagging Predictors. *Machine Learning*, 1996, vol. 24, no. 2, pp. 123–140. DOI: 10.1007/BF00058655

3. Freund Y, Schapire R.E. Experiments with a New Boosting Algorithm. *International Conference on Machine Learning*, 1996, pp. 148–156.

4. Маценов А.А. Комитетный бустинг: минимизация числа базовых алгоритмов при простом голосовании. Всероссийская конференция ММРО-13. 2007. С. 180–183. [Matsenov A.A. *Komitetnyy busting: minimizatsiya chisla bazovykh algoritmov pri prostom golosovanii* (Committee Boosting: Number of Base Algorithms Minimization for Simple Voting). *Vserossiyskaya konferentsiya MMRO-13* (All-Russian Conference MMRO-13). St. Peterburg, 2007, pp. 180–183.]

5. Mason L., Bartlett P., Baxter J. Direct Optimization of Margins Improves Generalization in Combined Classifiers. *Proc. of the 1998 conf. on Advances in Neural Information Processing Systems II*, MIT Press, 1999, pp. 288–294.

Кафтанников Игорь Леопольдович, канд. техн. наук, доцент кафедры электронных вычислительных машин, Южно-Уральский государственный университет, г. Челябинск; kil@is74.ru.

Парасич Андрей Викторович, аспирант кафедры электронных вычислительных машин, Южно-Уральский государственный университет, г. Челябинск; parasich_av@yandex.ru.

Поступила в редакцию 20 мая 2015 г.

DOI: 10.14529/ctcr150304

DECISION TREE'S FEATURES OF APPLICATION IN CLASSIFICATION PROBLEMS

***I.L. Kaftannikov**, South Ural State University, Chelyabinsk, Russian Federation, kil@is74.ru,*
***A.V. Parasich**, South Ural State University, Chelyabinsk, Russian Federation, parasich_av@yandex.ru*

The article describes the application of decision trees in classification problems. In recent years, decision trees are widely used for computer vision tasks, including object recognition, text classifica-

tion, gesture recognition, spam detection, training in ranking for information search, semantic segmentation and data clustering. This is facilitated by such distinctive features as interpretability, controllability and an automatic feature selection. However, there are number of fundamental shortcomings, due to which the problem of decision trees learning becomes much more complicated. The article provides the analysis of advantages and disadvantages of decision trees, the issues of decision trees learning and testing are considered. Particular attention is given to balance of training dataset. We also consider the decision forests and methods of its learning. A brief overview of methods for reducing errors interdependence of decision trees in decision forests learning is given. Methods for overcoming of drawbacks of decision trees are offered, results of these methods are proposed.

Keywords: decision trees, decision forests, machine learning, classification.

Received 20 May 2015

ОБРАЗЕЦ ЦИТИРОВАНИЯ

Кафтанников, И.Л. Особенности применения деревьев решений в задачах классификации / И.Л. Кафтанников, А.В. Парасич // Вестник ЮУрГУ. Серия «Компьютерные технологии, управление, радиоэлектроника». – 2015. – Т. 15, № 3. – С. 26–32. DOI: 10.14529/ctcr150304

FOR CITATION

Kaftannikov I.L., Parasich A.V. Decision Tree's Features of Application in Classification Problems. *Bulletin of the South Ural State University. Ser. Computer Technologies, Automatic Control, Radio Electronics*, 2015, vol. 15, no. 3, pp. 26–32. (in Russ.) DOI: 10.14529/ctcr150304