ТЕОРИЯ ОБОБЩАЮЩЕЙ СПОСОБНОСТИ АЛГОРИТМОВ МАШИННОГО ОБУЧЕНИЯ

ТОСАМО

Решетов Юрий Вячеславович

Вторая редакция

г. Ташкент

Оглавление

Прелюдия

Зачем нужна очередная теория машинного обучения, если в данной области уже имеют место другие альтернативные теории? Дело в том, что все предыдущие теории являлись теориями машинного обучения, а данная по праву уже может именоваться теорией машинного обобщения.

Суть в том, что теория обобщающей способности даёт ответы на такие вопросы, которые в предыдущих теориях ответов не имели:

1. Почему максимальная обучающая способность не не всегда соответствует максимальной обобщающей способности?
2. Как отличить информативные факторы от неинформативных?
3. Что делать, если решение неоднозначно?
4. Как получить неизменное качество классификации при наличии инвариантных искажений информации в выборке и каковы допустимые пределы инвариантности?
5. Где и в каких случаях искать причины низкой обобщающей способности: в выборке или в алгоритме?

Данная теория конструктивна в том смысле, что не только даёт ответы на теоретические вопросы, но ещё и предлагает соответствующие прикладные методы повышения обобщающей способности: как для предварительной обработки выборки, так и для последующего машинного обучения.

Исходная обучающая выборка

Пусть имеется обучающая выборка с примерами - фактами:

v_{11}, v_{12}, …, v_{1n}, dep_1
v_{21}, v_{22}, …, v_{2n}, dep_2
...
v_{m1}, v_{m2}, …, v_{mn}, dep_m

где:

Каждая строка выборки — некий эмпирический факт (пример).

v_{ij} – значение j- я объясняющая переменная для i-го примера
dep_i – зависимая переменная для i-го примера
n – количество факторов
m – количество примеров

В качестве наглядного примера мы возьмем простейшую задачу К. Нейлора для распознавания объектов: птицы, планера и самолёта по наличию или отсутствию их признаков: крыльев, хвоста, клюва, оперения, шасси, двигателя

В таком случае наша исходная выборка для задачи классификации объектов, где положительным исходом является принадлежность объекта к пернатым, а отрицательным — к непернатым, будет выглядеть так:

Объект\Признак	Крылья	Хвост	Клюв	Оперение	Шасси	Двигатель	Объясняющая переменная — принадлежность к пернатым
Птица	1	1	1	1	0	0	1
Планер	1	1	0	0	1	0	0
Самолёт	1	1	0	0	1	1	0

Примечание: Значение 1 — означает принадлежность, а 0 — отсутствие

Предварительная редукция неинформативных признаков

Редукция неинформативных признаков в ТОСАМО не является проблемой. В данный момент необходимо лишь показать, как нужно удалять некоторые неинформативные признаки из выборки ещё до того, как началась её предобработка. Если этого не сделать, то следующий этап нормировки будет невозможен.

Возьмём исходную выборку и внимательно посмотрим на значения признаков:

Объект\Признак	Крылья	Хвост	Клюв	Оперение	Шасси	Двигатель	Объясняющая переменная — принадлежность к пернатым
Птица	1	1	1	1	0	0	1
Планер	1	1	0	0	1	0	0
Самолёт	1	1	0	0	1	1	0

Мы видим, что значения нормированных признаков для факторов: Крылья и Хвост являются неизменными - константами. В таком случае их минимальное и максимальное значения равны. В тоже самое время значения зависимой переменной изменчивы. Это означает, что крылья и хвост не являются значимыми признаками и их необходимо подвергнуть редукции, т. е. удалить из выборки:

Объект\Признак	Клюв	Оперение	Шасси	Двигатель	Объясняющая переменная — принадлежность к пернатым
Птица	1	1	0	0	1
Планер	0	0	1	0	0
Самолёт	0	0	1	1	0

Нормировка обучающей выборки

Найдем максимумы и минимумы для всех зависимых переменных в выборке:

$max_j = Max(v_{1j}, v_{2j}, …;v_{mj})$

$min_j = Min(v_{1j}, v_{2j}, …;v_{mj})$

Нормируем все значения зависимых переменных из выборки к диапазону [-1, 1].

$x'_{ij} = 2 * (v_{ij} – min_j) / (max_j – min_j) - 1$

Также найдём максимум и минимум значения зависимой переменной из выборки.

$max_{dep} = Max(dep_1, dep_2, …;dep_m)$

$min_{dep} = Min(dep_1, dep_2, …;dep_m)$

Нормируем все значения зависимой переменной из выборки к диапазону [-1, 1].

$y_i = 2 * (dep_i – min_{dep}) / (max_{dep} – min_{dep}) – 1$

Перенормируем выборку для задачи классификации пернатых:

Объект\Признак	Клюв	Оперение	Шасси	Двигатель	Объясняющая переменная — принадлежность к пернатым
Птица	1	1	-1	-1	1
Планер	-1	-1	1	-1	-1
Самолёт	-1	-1	1	1	-1

Примечание: Нормирование не обязательно должно быть линейным, но обязательно должно приводить к распределению значений в диапазоне [-1, 1].

Т. е. в некоторых случаях возможно нормирование и нелинейные функций, в том числе и от нескольких аргументов. Применение нелинейных функций именуется ядерными преобразованиями (kernel tricks), например:

$x'_{ij} = 2 * (f(v_{i1}, v_{i2}, …, v_{in}) – min_j) / (max_j – min_j) – 1$

В данном случае max_j и min_j – это максимальное и минимальное значения фукнции f().

Система линейных неравенств

Составим из нормированных переменных обучающей выборки систему линейных неравенств:

$x'_{11} * w_1 + x'_{12} * w_2 + \dots x'_{1n} * w_n \; Z_i \; y_1$
$x'_{21} * w_1 + x'_{22} * w_2 + \dots x'_{2n} * w_n \; Z_i \; y_2$
...
$x'_{m1} * w_1 + x'_{m2} * w_2 + \dots x'_{mn} * w_n \; Z_i \; y_m$

Где:

w_i – неизвестный весовой коэффициент искусственного нейрона
Z_i – знак неравенства i-го примера, принимающий направление ≥, если значение нормированной зависимой переменной более нулевого, либо ≤, если значение зависимой переменной менее нулевого.

Для примера классификации пернатых система линейных неравенств будет выглядеть так:

$1 * w_1 + 1 * w_2 - 1 * w_3 - 1 * w_4 \geq 1$
$-1 * w_1 - 1 * w_2 + 1 * w_3 - 1 * w_4 \leq -1$
$-1 * w_1 - 1 * w_2 + 1 * w_3 + 1 * w_4 \leq -1$

Где:

w_1 – неизвестный весовой коэффициент для признака Клюв
w_2 – неизвестный весовой коэффициент для признака Оперение
w_3 – неизвестный весовой коэффициент для признака Шасси
w_4 – неизвестный весовой коэффициент для признака Двигатель

Инвариант линейных неравенств

Приведём систему линейных неравенств к инвариантной:

$(x'_{11} * w_1 + x'_{12} * w_2 + \ldots x'_{1n} * w_n) / y_1 \geq 1$
$(x'_{21} * w_1 + x'_{22} * w_2 + \ldots x'_{2n} * w_n) / y_2 \geq 1$
...
$(x'_{m1} * w_1 + x'm2 * w2 + \ldots x'_{mn} * w_n) / y_m \geq 1$

Обозначим для всех i и j:

$x_{ij} = x'_{ij} / y_i$

В таком случае первоначальная нормированная система линейных неравенств станет эквивалентной для нижеуказанной нормированно инвариантной:

$x_{11} * w_1 + x_{12} * w_2 + \ldots x_{1n} * w_n \geq 1$
$x_{21} * w_1 + x_{22} * w_2 + \ldots x_{2n} * w_n \geq 1$
...
$x_{m1} * w_1 + x_{m2} * w_2 + \ldots x_{mn} * w_n \geq 1$

Подвергнем инварианту пример классификации пернатых:

$1 * w_1 + 1 * w_2 - 1 * w_3 - 1 * w_4 \geq 1$
$1 * w_1 + 1 * w_2 - 1 * w_3 + 1 * w_4 \geq 1$
$1 * w_1 + 1 * w_2 - 1 * w_3 - 1 * w_4 \geq 1$

Примечания:

Предназначением нормировки к диапазону [-1, 1] и последующей инвариантности является перевод изначальной задачи линейной регрессии:

$v_{11} * w_1 + v_{12} * w_2 + \ldots v_{1n} * w_n \approx dep_1$
$v_{21} * w_1 + v_{22} * w_2 + \ldots v_{2n} * w_n \approx dep_2$
...
$v_{m1} * w_1 + v_{m2} * w_2 + \ldots v_{mn} * w_n \approx dep_m$

к задаче классификации. При этом наличие диапазона [-1, 1] для зависимых переменных y_i позволяет определиться со знаками для неравенств для задачи классификации в зависимости от знака значения зависимой переменной.

Если разделить все значения объясняющих переменных и значение зависимой переменной в каждом неравенстве нормированной системы на абсолютное значение зависимой переменной, то получим стандартную задачу бинарной классификации.

В линейном программирование вышеуказанные трюки с изменением знаков неравенств именуются приведением к стандартной форме.

Прямая и двойственная задачи

У всякой системы неравенств имеется двойственная система нормированно инвариантных неравенств:

$x_{11} * w'_1 + x_{21} * w'_2 + \ldots x_{m1} * w'_m \leq 1$
$x_{12} * w'_1 + x_{22} * w'_2 + \ldots x_{m2} * w'_m \leq 1$
...
$x_{1n} * w'_1 + x_{2n} * w'_2 + \ldots x_{mn} * w'_m \leq 1$

Вектор двойственной задачи: w'_1, w'_2, …, w'_m - это частоты (встречаемость) примеров. Если нам известны частоты примеров, предъявляемых классификатору вне выборки, то поиск решения задачи уже тривиален, т. к. зная решение двойственной задачи, получить решение прямой уже не составляет труда.

Другое дело, что частоты нам не всегда известны. Хотя грубо для выборки большой длины m, можно принять, что все частоты для примеров равны значению 1 / m.

Проблема машинного переобучения

Предположим, что мы решили экстремально оптимизировать задачу, т. е. получить максимальную результативность классификации на обучающей выборке. В таком случае мы получим некое решение прямой задачи в виде вектора весовых коэффициентов искусственного нейрона и решение для двойственной задачи, которое соответствует прямому решению. Но если допустить, что решение двойственной задачи это частоты примеров, предъявляемых для классификации примерно равные 1 / m при длине выборки m, то в таком случае полученное решение прямой задачи из двойственной с малой вероятностью совпадёт с экстремально оптимальным. Более того, вне выборки частоты примеров могут значительно отличаться и от значения 1 / m, поскольку сбор фактов для составления выборки может быть ограниченным по тем или иным причинам.

Проще говоря, экстремальная оптимизация прямой задачи за очень редким исключением может соответствовать вектору двойственной задачи. Такое несоответствие приводит к снижению обобщающей способности моделей, полученных по результатам машинного обучения с целью экстремальной максимизации обучающей способности, именуемой машинным переобучением.

Т. е. для того, чтобы обобщающая способность максимально совпадала с обучающей, необходимо и достаточно, чтобы:

1. Либо примеры предъявляемые для классификации вне выборки подстраивались под переоптимизированную модель,
2. Либо найти такой вектор решения прямой задачи, при котором оценка обучающей способности являлась независимой от вектора решения двойственной задачи и равной константе.

Выполнение условий для первого пункта в силу того, что на практике его требования не соответствуют действительности, т. е. модель во время машинного обучения подстраивается под некую наиболее удачную частоту примеров в обучающей выборке, с целью максимально результативно классифицировать самый оптимистичный вариант, реализующийся только при полном соответствии вектора двойственной задачи вектору прямой. Вне выборки, частота классифицируемых примеров уже может отличаться от наиболее удачной и не соответствовать вектору прямой задачи. Поскольку в таком случае происходит путаница местами причины и следствия, то неудивительно, что машинное обучение при попытке поиска наиболее удачной модели на обучающей выборке, приводит к переобучению вне выборки.

Выполнение условий второго пунта возможно в некоторых тривиальных случаях, когда максимальная и минимаксная оценки качества обучения совпадают. Например, если обучающая выборка содержит всего два примера с двумя предикторами. При увеличении длины выборки и количества предикторов в ней, вероятность того, что существует такой вектор для прямой задачи при котором независимо от значений вектора двойственной задачи, оценка качества обучения останется неизменной, умаляется.

Постановка задачи обобщения системы линейных неравенств, полученной из обучающей выборки

Обобщающая задача заключается в поиске значений весовых коэффициентов $w_1, w_2, \ldots, w_n$ для прямой задачи системы линейных неравенств вида:

$f(x_{11}) * w_1 + f(x_{12}) * w_2 + \ldots f(x_{1n}) * w_n \geq f(c)$
$f(x_{21}) * w_1 + f(x_{22}) * w_2 + \ldots f(x_{2n}) * w_n \geq f(c)$
...
$f(x_{m1}) * w_1 + f(x_{m2}) * w_2 + \ldots f(xmn) * w_n \geq f(c)$

при котором достигается максимально возможное выполнение условий системы линейных неравенств, при условии независимости от любого решения двойственной задачи в виде произвольных значений вектора: $w'_1, w'_2, \ldots, w'_m$, заданного в виде системы линейных неравенств:

$f(x_{11}) * w'1 + f(x_{21}) * w'_2 + \ldots f(x_{m1}) * w'_m \leq f(c)$
$f(x_{12}) * w'1 + f(x_{22}) * w'_2 + \ldots f(x_{m2}) * w'_m \leq f(c)$
...
$f(x_{1n}) * w'1 + f(x_{2n}) * w'_2 + \ldots f(x_{mn}) * w'_m \leq f(c)$

где:

$f(z) = a * z + b$ — линейный инвариант

b — может принимать действительные значения не меньшие a

a, c, $w'_1, w'_2, \ldots, w'_n$ – произвольные действительные числа

Теорема о линейной инвариантности системы линейных неравенств

Общий случай:

Пусть имеются два вектора одинаковой длины n с действительными значениями: $\bar{a} = \{a_1, a_2, \ldots, a_n\}$ и $\bar{b} = \{b_1, b_2, \ldots, b_n\}$.

Если $\bar{b}' = \bar{b} / c$, где $c = \Sigma_{i=1}^{n} b_i = Const \neq 0$, т. е. $\Sigma_{i=1}^{n} b_i' = 1$

То результат скалярного произведения ненормированного и нормированного векторов:

$y = \langle \bar{a}, \bar{b}' \rangle$

становится линейно инвариантным, т. к. линейное преобразование ненормированного вектора приводит к аналогичному линейному преобразованию скалярного произведения:

$d * y + e = \langle d * \bar{a} + e, \bar{b}' \rangle$

При $e \geq d$

В случае искусственного нейрона:

Если общая сумма всех весовых коэффициентов искусственного нейрона равна единице:

$w_1 + w_2 + \ldots w_n = 1$

то система линейных неравенств:

$x_{11} * w_1 + x_{12} * w_2 + \ldots x_{1n} * w_n \geq c$
$x_{21} * w_1 + x_{22} * w_2 + \ldots x_{2n} * w_n \geq c$
...
$x_{m1} * w_1 + x_{m2} * w_2 + \ldots x_{mn} * w_n \geq c$

Эквивалентна (линейно инвариантна) системам линейных неравенств:

$f(x_{11}) * w_1 + f(x_{12}) * w_2 + \ldots f(x_{1n}) * w_n = f(x_{11} * w_1 + x_{12} * w_2 + \ldots x_{1n} * w_n) \geq f(c)$
$f(x_{21}) * w_1 + f(x_{22}) * w_2 + \ldots f(x_{2n}) * w_n = f(x_{21} * w_1 + x_{22} * w_2 + \ldots x_{2n} * w_n) \geq f(c)$
...
$f(x_{m1}) * w_1 + f(x_{m2}) * w_2 + \ldots f(x_{mn}) * w_n = f(x_{m1} * w_1 + x_{m2} * w_2 + \ldots x_{mn} * w_n) \geq f(c)$

где:

$f(z) = a * z + b$

a и b- произвольные действительные числа

В таком случае: $b \geq -a * z$, но поскольку z после нормировки может принимать значения только в диапазоне [-1, 1], то:

$$b \geq a$$

Доказательство:

$(a * x_{i1} + b) * w_1 + (a * x_{i2} + b) * w_2 + \ldots (a * xin + b) * w_n \geq a * c + b$
$a * (x_{i1} * w_1 + x_{12} * w_2 + \ldots x_{1n} * w_n) + b \geq a * c + b$
$x_{i1} * w_1 + x_{12} * w_2 + \ldots x_{1n} * w_n \geq c$

Теорема доказана.

Выводы: Если искусственный нейрон решает задачу бинарной классификации, то линейные искажения для всех входных переменных вида $y = ax_i + b$, где x_i – значение на i-м входе нейрона, а y – выходное значение нейрона, при условии, что $b \geq a$ и общей сумме значений весовых коэффициентов нейрона не равной нулю, не повлияет на результаты бинарой классификации. Для этого необходимо и достаточно перенормировать весовые коэффициенты искусственного нейрона таким образом, чтобы сумма их значений была равна единице.

Обучение с оппонентом

Чтобы получить модель в виде решения прямой задачи, максимально независимой от решения двойственной задачи, можно воспользоваться минимаксом результативности классификации примеров из обучающей выборки.

Суть заключается в использовании теоремы о минимаксе фон Неймана-Моргенштерна[1], согласно которой если в антагонистической игре двух лиц (прямая и двойственная задача) с нулевой суммой (соответствие решения для прямой и двойственной задач) найдена минимаксная стратегия для одного из игроков (вектор решений для прямой задачи) и этот игрок будет перманентно придерживаться данной стратегии, то в таком случае минимальный выигрыш (оценка качества классификации) будет не хуже минимаксной оценки независимо от стратегии (вектора решения для двойственной задачи) второго игрока.

В этом случае мы получим нижнюю оценку качества обучающей способности, независимую от любого решения двойственной задачи.

Такой метод, т. е. поиск минимаксной результативности модели на обучающей выборке именуется обучением с оппонентом. На каждой итерации алгоритма обучения выполняются последовательно две задачи:

1. Попытка улучшить качество модели для всех примеров из обучающей выборки, которые были предъявлены ранее оппонентом — прямая задача обучающегося;
2. Попытка предъявить для улучшенной в п. 1 модели пример из обучающей выборки, на котором её результативность будет минимальной — двойственная задача оппонента.

Главным недостатком метода обучения с оппонентом является ограниченность обучающей выборки из-за чего в ней могут отсутствовать некоторые примеры, пригодные для адекватной коррекции модели, которые могут встретиться вне выборки. Впрочем, этот недостаток неустраним, т. к. в таком случае алгоритм должен был бы обладать сверхъестественными способностями. Этот же недостаток также является причиной машинного переобучения независимо алгоритма.

Но суть в том, что поскольку всякая произвольная модель — это всего лишь потенциальная гипотеза, то при обучении с оппонентом модель на каждом шаге итерации модифицируется таким образом, чтобы улучшить качество классификации при наихудшем для нее векторе двойственной задачи, который можно получить из примеров обучающей выборки. В результате чего модель вынужденно максимально обобщается, вместо того, чтобы банально вызубрить примеры из выборки в качестве готовых «решений». Ведь самые лёгкие для зубрёжки примеры либо вообще не будут предъявлены оппонентом, либо частота их предъявления сведётся к минимуму и результат запоминания примеров почти никак не скажется на общей результативности классификации моделью. Зато «лёгкие» примеры могут быть предъявлены уже вне выборки с более высокой частотой, что потенциально повышает обобщающую способность.

Решение проблемы неотрицательности значений вектора для прямой задачи

Известные до сих пор методы поиска минимаксных решений, позволяют находить решения только при условии неотрицательности значений векторов для прямой и двойственной задачи.

Однако это не является проблемой.

Представим значения номрированно-инвариантных объясняющих переменных в виде значений платёжной матрицы:

$x_{11}, x_{12}, \ldots, x_{1n}$
$x_{21}, x_{22}, \ldots, x_{2n}$
…
$x_{m1}, x_{m2}, \ldots, x_{mn}$

В таком случае, если удвоить матрицу вправо за счет копирования прежней матрицы в правую часть, но с нулевыми значениями, мы получим удвоенную матрицу:

$x_{11}, x_{12}, \ldots, x_{1n}, -x_{11}, -x_{12}, \ldots, -x_{1n}$
$x_{21}, x_{22}, \ldots, x_{2n}, -x_{21}, -x_{22}, \ldots, -x_{2n}$
…
$x_{m1}, x_{m2}, \ldots, x_{mn}, -x_{m1}, -x_{m2}, \ldots, -x_{mn}$

Решим платёжную матрицу с помощью метода поиска минимакса с целью максимизации решения для игрока по столбцам и минимизации для игрока по строкам.

В этом случае мы получим решение для прямой задачи в виде удвоенного вектора:

$w'_1, w'_2, \ldots, w'_n, w''_1, w''_2, \ldots, w''_n$

Чтобы получить вектор значений минимимаксного решения, в виде весовых коэффициентов искусственного нейрона независимых от знака, преобразуем значения удвоенного вектора для значений i от 1 до n:

$w_i = w'_i - w''_i$

Создадим удвоенную матрицу весовых коффициентов для примера классификации пернатых:

Объект\Признак	Клюв	Оперение	Шасси	Двигатель	Не Клюв	Не оперение	Не шасси	Не двигатель
Птица	1	1	-1	-1	-1	-1	1	1
Не Планер	1	1	-1	1	-1	-1	1	-1
Не Самолёт	1	1	-1	-1	-1	-1	1	1

Вектор Шепли

Если прямая задача имеет множество решений (неоднозначноть), то в таком случае необходимо придерживаться аксиом так называемого вектора Шепли[2] — принципа оптимальности распределения выигрыша, который чаще всего применяется в кооперативных играх.

Однако несмотря на то, что речь идет о применении антагонистической игры для двух лиц, но в силу того, что прямая задача не имеет однозначного решения, появляется необходимость соблюдения оптимального принципа распределения значений весовых коэффициентов в решении, где каждое значение вектора — это отдельные игроки, способные объединяться в коалиции против игроков коалиции двойственной задачи — оппонентов. Выигрыш, который им предстоит разделить является минимаксом общей результативности классификации. Наградой являются абсолютные значение весовых коэффициентов, общая сумма которых равна единице, т. е. ограничена.

Несмотря на то, что принципы оптимальности, соответствующие вектору Шепли применяются к решениям, полученным на обучающей выборке, тем не менее их соблюдение обеспечивает минимизацию риска снижения обобщающей, а не обучающей способности. Что касается обучающей способности, то некоторые аксиомы вектора Шепли могут её заметно умалить.

Редукция незначимых (неинформативных) предикторов и неопорных векторов (примеров) в обучающей выборке

Редукция, т. е. обнуление весовых коэффициентов для неинформативных предикторов прямой задачи, соответствует аксиоме болвана для вектора Шепли.

Выявление неинформативных признаков и неопорных векторов примеров обучающей выборки является тривиально разрешимой задачей в теории обобщающей способности алгоритмов машинного обучения, а вовсе не проблемой, как это представлялось в предыдшествующих теориях машинного обучения.

Зная платёжную матрицу, т. е. значения объясняющих переменных в виде таблицы после их нормализации и инварианта, можно тривиально выявить незначимые объясняющие переменные и неопорные векторы примеров, удалив их из матрицы ещё до того, как будет получено решение.

В теории игр это называется удалением доминируемых строк и столбцов.

Если в платёжной матрице, решение которой заключается максимизации стратегии для игрока по столбцам (прямая задача), имеет место такой столбец, все значения которого для всех индексов (номеров ячееек) являются не большими по сравнению со значениями в любых других столбцах при равных индексах ячеек сравниваемых столбцов, но при некоторых индексах присутствуют значения меньшие по сравнению с равноиндексной ячейкой в каким либо ином столбце, то такой столбец является доминируемым и его можно удалить - редуцировать. Поскольку каждый столбец матрицы — это значения какой либо объясняющей переменной (предиктора, фактора), то в таком случае происходит редукция доминируемых (незначимых, неинформативных) факторов. Оставшиеся после редукции всех доминируемых столбцов, доминирующие столбцы являются значимыми (информативными) факторами.

Значимые факторы являются системой линейных уравнений, решение которой является решением минимаксной двойственной задачи исходной системы линейных неравенств.

Если в платёжной матрице, решение которой заключается минимизации стратегии для игрока по строкам (двойственная задача), имеет место такая строка, все значения которой для всех индексов (номеров ячееек) являются не меньшими по сравнению со значениями в любых других строках при равных индексах ячеек сравниваемых строк, но при некоторых индексах присутствуют значения большие по сравнению с равноиндексной ячейкой в какой либо иной строке, то такая строка является доминируемой и её можно удалить - редуцировать.. Поскольку каждая строка матрицы — это пример из обучающей выборки, то удаление доминируемых строк соответствует неопорности векторов обучающих примеров. Оставшиеся после редукции всех доминируемых строк, доминирующие строки являются опорными векторами.

Опорные (доминирующие) векторы являются системой линейных уравнений, решение которой является минимаксным решением прямой задачи исходной системы линейных неравенств.

Что произойдёт, если не соблюдать аксиому болвана, т. е. не обнулить весовые коэффициенты неинформативных признаков? Во многих случаях, это приведёт к тому, что обучающая способность повысится. Но что касается обобщающей способности, то результаты могут умалиться. Ведь неинформативные признаки «значимы» только в пределах обучающей выборки и незначимы за её пределами. За пределами выборки интерпретация незначимых факторов будет вносить искажения в качество классификации. Даже если математическое ожидание от таких искажений будет стремиться к нулевому значению, то тем не менее, поскольку максимальной инвариантности можно добиться лишь при сумме значений всех весовых коэффициентов равной единице, то это означает, что увеличение весового коэффициента для любого незначимого признака возможно только за счёт умаления весовых коэффициентов значимых признаков в силу ограничения общей суммы весовых коэффициентов. А следовательно учёт и интерпретация незначимых признаков моделью всегда будет отрицательно влиять на обобщающую способность.

Попробуем удалить незначимые признаки для примера классификации пернатых.

Поскольку минимальным является значение -1 для всех строк без исключения, то всякий столбец, состоящий только из значений -1 является доминируемым:

Объект\Признак	Клюв	Оперение	Шасси	Двигатель	Не Клюв	Не оперение	Не шасси	Не двигатель
Птица	1	1	-1	-1	-1	-1	1	1
Не Планер	1	1	-1	1	-1	-1	1	-1
Не Самолёт	1	1	-1	-1	-1	-1	1	1

Удалим доминируемые столбцы.

Объект\Признак	Клюв	Оперение	Двигатель	Не шасси	Не двигатель
Птица	1	1	-1	1	1
Не Планер	1	1	1	1	-1
Не Самолёт	1	1	-1	1	1

Смотрим далее. Столбец для признака двигатель не лучше столбца для признака клюв, а по некоторым значениям хуже. В таком случае столбец для признака Клюв является доминирующим, а для признака Двигатель доминируемым, т. е. подлежащим редукции. Точно также доминируемым признаком по отношению к доминирующему признаку Клюв является Не двигатель.

Удалим доминируемые столбцы:

Объект\Признак	Клюв	Оперение	Не шасси
Птица	1	1	1
Не Планер	1	1	1
Не Самолёт	1	1	1

Поскольку в результате редукции лишь обнуляются все значения в доминируемом столбце, а впоследствии происходит вычитание значений из столбцов без отрицания значений столбцов с отрицанием, то в коненом результате получим:

Объект\Признак	Клюв	Оперение	Шасси
Птица	1	1	-1
Не Планер	1	1	-1
Не Самолёт	1	1	-1

Проблема неоднозначности решений прямой задачи

В некоторых случаях система линейных уравнений, составленная из опорных векторов, после редукции неинформативных объсняющих переменных, может оказаться неоднозначной в случае, если значимые объясняющие переменные линейно зависимы.

В таком случае, вектор решения можно представить в виде коалиции множества лиц (по числу элементов вектора) и соблюдением критерия Шепли. Поскольку «болваны», т. е. доминируемые элементы вектора уже редуцированы, то остаётся лишь равноправно распределить значения весовых коэффициентов между равнозначащими (информативность которых одинакова) объясняющими переменными.

На примере классификации пернатых мы видим, что все значения столбцов абсолютно равнозначны по абсолютному значению, хотя знаки и не совпадают:

Объект\Признак	Клюв	Оперение	Шасси
Птица	1	1	-1
Не Планер	1	1	-1
Не Самолёт	1	1	-1

В таком случае конечным результатом может быть решение в виде векторов:

$w_1 = 1$
$w_2 = 1$
$w_3 = -1$

Но при этом имеет место линейная зависимость признаков, т. е.: $w_1 + w_2 – w_3 = Const$

Теоретически мы могли бы обойтись одним единственным любым из всего линейно зависимого набора признаков для того, чтобы правильно классифицировать пернатых. Но в практических целях этого делать ни в коем случае не следует. По той простой причине, что наша задача заключена не в том, насколько результативен классификатор в плане обучающей способности, а в том насколько он результативен в плане обобщающей способсности. Об результативности классификатора в обобщающей способности можно только догадываться, т. к. вне выборки могут попасться такие примеры для классификации, которые в обучающую выборку по тем или иным причинам не попали. В таком случае, попытка сокращения пространства признаков за счёт редукции некоторых линейно зависимых факторов может привести к максимально нежелательным результатам, по сравнению с соблюдением аксиоматики вектора Шепли.

Предположим, что в обучающую выборку по классификации пернатых не попал такой объект, как утконос. В таком случае, несмотря на то, что классификатор был недообучен, тем не менее результат классификации с вышеуказанным решением окажется верным. По той простой причине, что все равнозначимые признаки получили одинаковые абсолютные и адекватные по знаку значения. И из трёх признаков (Клюв, Оперение и Шасси), два оказались верными (Оперение и Шасси) и один дал погрешность (Клюв). Любая редукция равнозначимых признаков дала бы худший результат вне выборки по сравнению с принципом оптимальности вектора Шепли.

Вывод: соблюдение критерия Шепли минимизирует риск снижения обобщающей способности.

Резюмируя вышесказанное

Если быть очень кратким, то вышеуказанный метод — это метод поиска максимальной линейной зависимости между предикторами искусственного нейрона для опорных векторов, ведь после инварианта вместо зависимой переменной мы получаем некую константу, а не зависимую переменную, относительно которой и ищется такая зависимость.

Формула этой зависимости и есть математическая модель, т. е. та самая закономерность, которую мы ищем в обучающей выборке.

Линейных зависимостей пытаются избежать в линейной алгебре, поскольку в этом случае решение системы линейных уравнений не единственно и определитель матрицы нулевой.

В машинном обучении дело обстоит с точностью до наоборот, т. е. целью является поиск обобщающей линейной зависимости, а вовсе не частного случая, уже не подлежащего никакому обобщению.

Однако, в результате обобщения мы получаем неединственное решение, но нам необходимо найти весовые коэффициенты, т. е. вектор решения прямой задачи в виде значений — констант, при котором линейная зависимость будет наиболее оптимальна. Таким решением является оптимальный вектор Шепли.

По сути задача состоит из двух подзадач:

1. Поиск наиболее общего решения, с целью устранения всего излишнего, т. е. не имеющего отношения к обобщению.
2. Поиск частного оптимального решения в обобщённом решении.

Несмотря на то, что обе подзадачи представляются конфликтующими (имеющими разные цели), тем не менее их можно решать параллельно с помощью метода Брауна-Робинсон-Решетова, стабильно приближающего к конечной цели на каждом шаге итерации.

Метод Брауна-Робинсон-Решетова

Метод фиктивной игры Брауна-Робинсон[3][4] является наиболее старым (1951 г.) алгоритмом итеративного решения минимаксных задач, представленных в виде платёжных матриц. При этом он является методом поиска решения с оппонентом и способностью автоследукции доминируемых строк и столбцов.

Однако, ему присущи ряд недостатков:

1. Несоответствие решения аксиоматике вектора Шепли, т. к. полученное решение зависит от порядка расположения строк и столбцов в платёжной матрице.
2. Низкая сходимость к решениям

Чтобы устранить вышеуказанные недостатки, пришлось усовершенствовать алгоритм, внеся в него изменения.

Возьмём прежний алгоритм Брауна-Робинсон:

Для строк платёжной матрицы в нём предусмотрен вектор аккумулятора, равный длине векторов строк и счётчик используемых алгоритмом столбцов. Оба вектора в начале алгоритма содержат нулевые значения.

Точно также для столбцов платежной матрицы в нём предусмотрен и вектора аккумулятора, равный длине векторов столбцов и счётчик используемых алгоритмом строк. Оба вектора в начале алгоритма содержат нулевые значения.

Для начала выбирается произвольный столбец платёжной матрицы и задаётся значение счётчика итераций.

1. Значения аккумулятора столбцов суммируются со значением выбранного столбца платёжной матрицы.
2. Среди значений аккумулятора столбцов ищется максимальное.
3. Если максимальных значений в аккумуляторе столбцов несколько, то произвольно выбирается одно из них.
4. Индекс выбранного максимального значения в аккумуляторе столбцов является индексом следующей выбранной строки в платежной матрице.
5. В счётчике строк значение с индексом выбранной строки из п. 4 увеличивается на единицу.
6. Значения аккумулятора строк суммируются со значениями выбранной строки платёжной матрицы.
7. Среди значений аккумулятора строк ищется минимальное.
8. Если минимальных значений в аккумуляторе строк несколько, то произвольно выбирается одно из них.
9. Индекс выбранного минимального значения в аккумуляторе строк, является индексом следующего выбранного столбца из платежной матрицы.
10. Значение счётчика итераций уменьшается на единицу.

11. Если значение счётчика итераций не меньше нулевого, то переход к п. 1

Значения вектора счетчика столбцов является вектором решения по столбцам для прямой задачи. Значения вектора счётчика строк является вектором решения по строкам для двойственной задачи.

Поскольку алгоритмы поиска устроены так, что в цикле они сравнивают значения для двух соседних индексов, то в п. 3 и п. 8, то чаще выбираются строки и столбцы, которые стоят ближе к началу цикла, а соответственно такие строки и столбцы получают максимальные решения в векторах счётчиков. Т. е. в этом случае аксиоматика вектора Шепли для векторов решений не соблюдается, т. к. в случае неоднозначности решений, значения в векторах зависят от порядка строк и столбцов, расположенных в платёжной матрице.

Чтобы избавиться от этого недостатка, а также максимально соблюсти ещё одну аксиому вектора Шепли, согласно которой награда (значение вектора) должна быть равна для тех членов коалиции вклад которых равнозначен (под вкладом в данном случае имеется в виду значение доминирования), пришлось внести изменение в алгоритм в п. 3 и п. 8:

...

3. Если максимальных значений в аккумуляторе столбцов несколько, то выбирается то значение с индексом, значение счетчика с таким же индексом для которого имеет наименьшее значение.

...

8. Если минимальных значений в аккумуляторе строк несколько, то выбирается то значение с индексом, значение счётчика с таким же индексом для которого имеет наименьшее значение.

…

Проще говоря, распределение значений в счётчиках после модификации стало таким, чтобы максимально соответствовать аксиомам Шепли.

Второй недостаток, а именно низкая сходимость исходного алгоритма Брауна-Робинсон к точным решениям при наличии однозначности решений, заключалась в том, что максимум погрешностей алгоритм вносил в начальных итерациях, где мог выбрать доминирущие столбцы или строки. В конечном итоге он сходился к тому, что на всех итерациях зацикливался только на доминируемых строках и столбцах.

Чтобы избавиться от этого недостатка, был в алгоритм введён дополнительный счётчик начальных итераций, по достижении которого счётчики используемых строк и столбцов сбрасывались и далее алгоритм уже стабильно зациклившись только на доминируемых строках и столбцах давал высокое схождение к точным решениям.

В результате внесённых мною модификаций в метод Брауна-Робинсон, получился улучшенный метод Брауна-Робинсон-Решетова.

Реализация

Поскольку теоретические рассуждения зачастую могут содержать ошибки или неточности и в результате допущенных ошибок на каком либо этапе рассуждений, все последующие этапы также могут оказаться ошибочными, то для того чтобы удостовериться в правильности теории, желательно испытать её на практике.

Для проведения таких испытаний и была создана реализация векторной машины с минимаксом обучающей способности в виде библиотеки libVMR на Java, доступная в виде исходных кодов под лицензией свободного исходного кода GPL v. 3, а также в виде уже скомпилированного и платформеннонезависимого приложения с оконным графическим интерфейсом в формате JAR.

Загрузить исходные коды и приложение можно по адресу: http://libvmr.sourceforge.net

Свежие редакции данного документа

Наиболее свежую редакцию данного документа можно загрузить по адресу:
https://sites.google.com/site/libvmr/home/theory

Список литературы

1. Нейман Дж., Моргенштерн О. Теория игр и экономическое поведение. М. Наука 1970. (Перевод с английского).
2. Lloyd S. Shapley. "A Value for n-person Games". In Contributions to the Theory of Games, volume II, by H.W. Kuhn and A.W. Tucker, editors. Annals of Mathematical Studies v. 28, pp. 307–317. Princeton University Press, 1953.
3. Brown, G.W. "Iterative Solutions of Games by Fictitious Play" In Activity Analysis of Production and Allocation, T.C. Koopmans (Ed.), New York: Wiley, 1951
4. Robinson, J. "An Iterative Method of Solving a Game", Annals of Mathematics 54, 296-301, 1951