Решение задач анализа данных, основанное на линейной комбинации деформаций*

Дьяконов А. Г.
djakonov@mail.ru
Московский государственный университет имени М.В. Ломоносова;
Вычислительный центр им. А.А. Дородницына РАН

Дан обзор некоторых теоретических результатов представления функций и алгоритмов в специальном виде: линейной комбинации «деформации» линейных функций/алгоритмов. В теории интерполяции подобные результаты отталкиваются от работ А.Н. Колмогорова, а в теории классификации — от работ Ю.И. Журавлева. Показано, что идеи подобного представления можно успешно использовать на практике. Описаны решения нескольких прикладных задач в рамках крупных международных конкурсов.

Ключевые слова: *интерполяция, деформация, алгебраический подход, представление функций, регрессия, рекомендация, классификация, прикладные задачи.*

Data mining problems solving using linear combinations of deformations*

D'yakonov A.G.
Lomonosov Moscow State University; Dorodnicyn Computing Centre of RAS

A review of some theoretical results on function representation by linear combination of "deformations" of linear functions is presented. The results are started from Kolmogorov's papers on interpolation and Zhuravlev's papers on algebraic approach. It is shown that the idea of such representation can be useful in practice. Some solutions of real problems from international data mining competitions are described.

Keywords: *interpolation, calibrating, algebraic approach, function representation, regression, recommendation, classification, applied problems.*

Введение

Работа посвящена теоретическим обоснованиям и практическим приложениям поиска решения задачи регрессии в виде

$$c_1\varphi(B_1) + \cdots + c_r\varphi(B_r), \tag{1}$$

где $B_1, \ldots, B_r$ – решения отдельных регрессионных алгоритмов (либо очень простых, либо представимых в виде линейной комбинации «простых»), а φ – «функция деформации», которая выбирается заранее (исходя из специфики задачи) или специально строится. Поскольку практически любой алгоритм классификации сначала получает некоторые значения, которые естественно назвать «оценками принадлежности к классам», а затем на их основе классифицирует, речь в работе пойдёт не только о регрессии, а о более широком спектре приложений: задачи классификации, рекомендации и т.п. В алгебраическом подходе к распознаванию Ю.И. Журавлева вводятся операции над алгоритмами класси-

Работа выполнена при финансовой поддержке РФФИ, проект №12-07-00187-а.

фикации, которые, по сути, индуцируются операциями над ответами так называемых распознающих операторов. Поэтому с точки зрения этого подхода выражение (1) определяет вид алгоритма.

Сначала в работе дается обзор теоретических результатов, отдельно приводятся схемы доказательств некоторых теорем, затем описываются реальные прикладные задачи и методы их решения, основанные на представлении (1).

Обзор теоретических результатов

Теория интерполяции

В теории интерполяции хорошо известна следующая теорема А.Н. Колмогорова [1], которую постоянно приводят в учебниках по теории нейронных сетей (см., например, [2]) как фундаментальный результат, обосновывающий концепцию нейросетей.

Теорема 1. *Любую непрерывную функцию f, заданную на единичном кубе n-мерного пространства, можно представить в виде*

$$f(x_1,\ldots,x_n)=\sum_{q=1}^{2n+1} h_q\left(\sum_{p=1}^{n}\varphi_{p,q}(x_p)\right), \tag{2}$$

где h_q, $\varphi_{p,q}$ – непрерывные функции, кроме того, $\varphi_{p,q}$ не зависят от выбора функции f.

Естественно, выражение (2) в теореме лишь внешним видом напоминает функциональность нейронной сети, поскольку на практике используются достаточно простые функции активации (в (2) это функции h_q и $\varphi_{p,q}$). В конце XX века удалось получить несколько результатов, которые в большей степени оправдывают использование нейронных сетей на практике. В русскоязычной литературе чаще приводят следующую теорему 1998 г. [3].

Теорема 2. *Пусть X – компактное пространство, $E\subseteq C(X)$ – замкнутое линейное подпространство в $C(X)$, $1\in E$, функции из E разделяют точки в X и E замкнуто относительно нелинейной унарной операции $\varphi\in C(\mathbb{R})$, тогда $E=C(X)$.*

Здесь и далее $C(X)$ – множество непрерывных функций со значениями из $\mathbb{R}$, определенных на X. Под замкнутостью относительно φ понимается, что для любой функции $g\in E$ выполняется вложение $\varphi g\in E$ (где $\varphi g(x)=\varphi(g(x))$). Разделение точек означает, что

$$\forall x_1,x_2\in X\ \ \exists g\in E:\ g(x_1)\neq g(x_2).$$

Из теоремы следует, что любую непрерывную вещественную функцию $f(x_1,\ldots,x_n)$ можно приблизить нейросетью, у которой в качестве функций активации (у нейронов) используются фиксированная функция φ (любая удовлетворяющая условиям теоремы), и тождественная функция. Этот результат был получен значительно раньше (в 1991 г.) В. Крейновичем [4]. Нам потребуется следующий очень интересный результат, полученный в 1993 г. А. Пинкусом и соавторами [5], [6] (см. также обзор [7]):

Теорема 3. *В топологии равномерной сходимости на компактах*

$$C(\mathbb{R})=\overline{L}\left(\{\varphi(ax+b)\,|\,a,b\in\mathbb{R}\}\right),$$

где $\varphi\in C(\mathbb{R})$ – не полином.

Здесь $L(X)$ – множество конечных линейных комбинаций элементов из X, верхняя черта – замыкание. Изначально [5], теорема была доказана при более сильных предположениях, однако в дальнейшем – для класса непрерывных функций, обобщена на многомерный случай, кроме того, сформулирована в виде критерия (см. предположение 6.4 работы [7]):

Теорема 4. *Для любой функции $\varphi \in C(\mathbb{R})$ справедливо равенство*

$$C(\mathbb{R}^n) = \overline{L}\left(\left\{\varphi\left(\sum_{i=1}^{n} a_i x_i + b\right) \,\middle|\, (a_1,\ldots,a_n) \in \mathbb{R}^n,\, b \in \mathbb{R}\right\}\right). \tag{3}$$

тогда и только тогда, когда φ не является полиномом.

Таким образом, любая непрерывная функция может быть приближена двухслойной нейронной сетью, причем в первом слое – фиксированная функция активации φ (непрерывная, неполиномиальная), а во втором – тождественная (открытым остается вопрос о числе нейронов для достижения заданного приближения). Результат А.Н. Колмогорова часто интерпретируют как «отсутствие функций большого числа переменных», т.е. существуют только сумма и функции одного переменного, а все остальные функции – их суперпозиции (речь идет, естественно, о непрерывных функциях). У результата А. Пинкуса и соавторов более сильная интерпретация. С точностью до приближения существуют только линейные комбинации и ровно одна функция одного переменного! Причем на роль такой единственной функции φ подходит любая непрерывная, кроме полинома. Кроме того, при приближении любой непрерывной функции не надо использовать «вложенных суперпозиций», т.е. вида

$$\varphi(\ldots\varphi(\ldots)\ldots). \tag{4}$$

В данной работе мы покажем, что эту смелую интерпретацию часто удается эксплуатировать на практике при решении реальных прикладных задач анализа данных.

Деформации в алгебраическом подходе к решению задач классификации

Как уже было сказано, в алгебраическом подходе вводятся операции над алгоритмами. Подробнее о подходе можно узнать в работах [8],[9]. Пусть алгоритм A должен классифицировать объекты $S_1,\ldots,S_q$ (контрольную выборку) на l, вообще говоря, пересекающихся класса, т.е. получить бинарную $q \times l$-матрицу, ij-й элемент которой является ответом алгоритма на вопрос «принадлежит ли j-му классу объект S_i». В алгебраическом подходе постулируется, что любой алгоритм классификации A можно представить в виде суперпорзиции распознающего оператора B и решающего правила C (в [8] есть даже теорема, доказывающая это утверждение, но она немного искусственная). Распознающий оператор получает вещественную $q \times l$-матрицу оценок $\Gamma[B] = ||\gamma_{ij}||_{q\times l}$: её ij-й элемент интерпретируется как оценка принадлежности j-му классу объекта S_i. Решающее правило переводит матрицу оценок в бинарную матрицу, например, так:

$$C(||\gamma_{ij}||) = ||\alpha_{ij}||,\ \alpha_{ij} = \begin{cases} 1, & \gamma_{ij} \geqslant c, \\ 0, & \gamma_{ij} < c \end{cases} \tag{5}$$

(пороговое решающее правило). В алгебраическом подходе решающее правило фиксируется, а операции над алгоритмами индуцируются операциями над распознающими операторами, которые вводятся следующим образом:

$$\Gamma[B_1 + B_2] = \Gamma[B_1] + \Gamma[B_2],$$

$$\Gamma[c \cdot B_1] = c \cdot \Gamma[B_1],$$

$$\Gamma[B_1 \cdot B_2] = \Gamma[B_1] \cdot \Gamma[B_2],$$

здесь B_1, B_2 – распознающие операторы, умножение «$\cdot$» – поэлементное.

С помощью введенных операций можно строить полиномы над алгоритмами (распознающими операторами).

Весь смысл введения операций объясняется в основной теореме алгебраического подхода [8] (см. также [9]): при необременительных ограничениях на постановку задачи можно с помощью полиномов над некорректными алгоритмами (которые допускают ошибки на контрольной выборке) строить корректные алгоритмы.

Оказывается (мы сформулируем соответствующую теорему в следующем параграфе), что корректные полиномы можно строить в виде (1), где φ – полином специального вида. Но «корректность» начинается с определённой степени полинома. Если же взять в качестве φ неполиномиальную операцию, при этом считаем, что

$$\Gamma[\varphi(B)] = \varphi(\Gamma[B]),$$

где функция φ действует поэлементно, то также корректный алгоритм представим в виде (1). Первоначально автор доказал это для деформации $\varphi(x) = \frac{1}{x}$ [10], но после того как ему стали известны результаты А. Пинкуса (о справедливости которых он уже сам догадался), общий случай стал очевидным (см. доказательство ниже).

Определение 1. *Линейным замыканием модели (множества матриц оценок) называется множество всевозможных линейных комбинаций операторов модели (матриц оценок).*

Отметим, что если линейное замыкание модели алгоритмов пополнить специальными операциями нормировки [10], которые часто применяются на практике, то корректный алгоритм можно не получить, но все, что получается, можно представить в виде (1). Таким образом, опять при пополнении линейного замыкания вложения вида (4) можно не использовать.

Доказательства некоторых результатов алгебраического подхода

Этот параграф не является историческим обзором, и его можно пропустить читателю без потери нити рассуждения. Здесь мы сформулируем и докажем результаты, которые описали раньше. В формулировках будет фигурировать модель алгоритмов B^* (точнее, это фиксированное множество операторов). На нее накладываются следующие условия:

1. В линейном замыкании пространства матриц оценок (операторов модели) есть константная матрица (все элементы которой равны ненулевой константе).
2. В линейном замыкании пространства матриц оценок есть базис, состоящий из бинарных матриц.
3. В линейном замыкании пространства матриц оценок есть матрица с попарно различными элементами.

Этим требованиям удовлетворяет, например, модель операторов вычисления оценок (см. [9]).

При отказе от «экзотического» требования (2) формулировки теорем немного поменяются; на самом деле, многие модели этому требованию удовлетворяют. Требование (3) является скорее свойством задачи, а не модели (ясно, что если $S_1 = \ldots = S_q$, то вряд ли оно выполняется в «разумных моделях»). Оно следует из «непротиворечивости» исходной информации.

Теорема 5. *Любую матрицу оценок, которую можно получить полиномом степени не выше k над операторами модели B^* можно получить полиномом вида* (1), *где φ – полином k-й степени специального вида (например, подойдет $\varphi(x) = x^k$), $B_1, \ldots, B_r$ – линейные комбинации операторов модели (причём они не зависят от получаемой матрицы и степени k).*

Доказательство. Приведем лишь схему доказательства, которое основано на технике, похожей на обоснование применения полиномиального ядра в SVM. По этой схеме и работе [9] без труда восстанавливается полное доказательство.

Распознающий оператор можно отождествить с его $q \times l$-матрицей оценок, а её – с ql-мерным вектором, достаточно «вытянуть» матрицу в вектор:

$$||\gamma_{ij}||_{q\times l} \;\to\; (\gamma_{11},\ldots,\gamma_{1l},\gamma_{21},\ldots,\gamma_{ql}).$$

Из ограничения на модель следует, что в линейном замыкании M таких ql-мерных векторов есть базис $\tilde{x}_1,\ldots,\tilde{x}_s$ из бинарных векторов и вектор $\tilde{1}=(1,\ldots,1)^{\mathrm{T}}$.

Запишем базисные векторы $\tilde{x}_i$ и их инверсии $\tilde{1}-\tilde{x}_i$ по столбцам в матрицу X. Нетрудно показать, что линейное замыкание $L(X)$ столбцов матрицы X (а это и есть линейное замыкание M) совпадает с линейным замыканием $L(XX^{\mathrm{T}})$ столбцов матрицы Грама XX^{T}.

Пусть теперь матрица X' состоит из всевозможных мономов степени не выше k (мономы x_ix_j и x_jx_i считаем разными) над столбцами матрицы X. Оказывается, что матрица $X'X'^{\mathrm{T}}=\varphi(XX^{\mathrm{T}})$, где функция $\varphi(x)=x^k$ применяется поэлементно. При этом $L(X')=$ $=L(XX')$. Таким образом, в пространстве полиномов степени не выше k над векторами исходного пространства есть база векторов $\varphi(y_j)$, где y_j пробегает столбцы матрицы XX^{T}, т.е. является линейной комбинацией столбцов матрицы X.

Переходя теперь от ql-мерных векторов к соответствующим операторам получаем справедливость утверждения теоремы.

■

Эта теорема описывает полиномиальные замыкания модели, т.е. пространства полиномов степени не выше k (впервые были полностью описаны в [9]). С прикладной точки зрения удручает тот факт, что, вообще говоря, в теореме 5 $r=ql$ (понижение r возможно лишь в частных случаях).

Рассмотрим теперь неполиномиальные операции:

Теорема 6. *Любую $q\times l$-матрицу можно получить оператором вида* (1) *где $B_1,\ldots,B_r$ – операторы из линейного замыкания модели B^*, $\varphi\in C(\mathbb{R})$ – не полином.*

Доказательство. Как и в предыдущем доказательстве отождествим оператор с ql-мерным вектором. В линейном замыкании множества векторов, которое соответствует модели, есть вектор $\tilde{1}=(1,\ldots,1)^{\mathrm{T}}$ и вектор $\tilde{x}=(x_1,\ldots,x_{ql})^{\mathrm{T}}$ с попарно различными элементами. Рассмотрим выражение вида

$$\sum_i c_i\varphi(a_i\tilde{1}+b_i\tilde{x})=\sum_i c_i\begin{bmatrix}\varphi(a_i+b_ix_1)\\ \vdots\\ \varphi(a_i+b_ix_{ql})\end{bmatrix} \tag{6}$$

(сумма конечная). Из теоремы 3 следует, что любой ql-мерный вектор можно представить с любой наперед заданной точностью вектором вида (6) и из векторов такого вида можно составить базис ql-мерного пространства. Действительно, это следует из того, что непрерывную функцию одного аргумента, которая равна 1 в точке x_j, а в остальных x_t, $t\neq j$, равна нулю, можно сколь угодно точно приблизить выражением

$$\sum_i c_i\varphi(a_i+b_ix).$$

Линейная комбинация полученных базисных векторов ql-мерного пространства также будет иметь вид (6). Переходя теперь от векторов к соответствующим операторам получаем справедливость утверждения теоремы. ■

Решение прикладных задач

Приведем примеры реальных прикладных задач, которые удалось успешно решить, используя описанные выше результаты. Причём не просто успешно, а существенно лучше многих классических методов, поскольку решения стали победителями в рамках крупных международных конкурсов. Все они получены алгоритмами, которые имеют вид (1).

Технология LENKOR

Технология разрабатывалась для решения задач с разнородной информацией и изначально применялась для построения рекомендательных систем и прогнозирования деятельности научных коллективов. В [11] описано решение задачи Международного конкурса «ECML/ PKDD Discovery Challenge 2011 (VideoLectures.Net Recommender System Challenge)» [12] по созданию рекомендательной системы ресурса VideoLectures.net. Ниже мы опишем принципы решения этой задачи, чтобы продемонстрировать связь с поиском решения в виде (1).

Для некоторого множества лекций заданы их описания:

— идентификационный номер лекции,
— язык лекции (английский, французский, русский и т.п.),
— категория лекции (например «Computer Science» или «Biology»),
— число просмотров лекции,
— дата публикации на сайте,
— автор лекции (а также подробная информация о нём: e-mail, сайт в Интернете и т.д.),
— название лекции,
— описание лекции (небольшой текст).

Аналогичные данные имеются по событиям, к которым относятся лекции (конференции, на которых они прочитаны, школы-семинары, циклы лекций и т.д., см. подробнее [12]).

Множество описанных лекций разбито на два непересекающихся подмножества: 6983 «старые лекции», которые были опубликованы на сайте до 1 июля 2009 г. (по ним известна вся статистика) и 1122 «новые лекции», которые выложены на сайте после этой даты (по ним не известна информация о просмотрах). Есть также контрольная выборка – это подмножество множества старых лекций. Для каждой лекции из контрольной выборки необходимо предложить список рекомендуемых 30 новых лекций, таким образом, если новый пользователь (интересы которого нам не известны) заходит на сайт и начинает смотреть лекцию – мы можем рекомендовать ему новинки (упорядоченный список из 30 недавно закачанных на сайт лекций).

Качество рекомендации лекций с номерами $a_1, \ldots, a_{30}$ по лекции из контрольной выборки оценивалась по формуле

$$\frac{1}{6} \sum_{z \in \{5,10,15,20,25,30\}} \frac{|\{y_1, \ldots, y_{\min(z,s)}\} \cap \{a_1, \ldots, a_{\min(z,s)}\}|}{\min(z, s)}, \tag{7}$$

где $y_1, \ldots, y_s$ – номера лекций, которые действительно просматривались после рассматриваемой «старой» лекции (идут по уменьшению популярности).

Применение технологии «LENKOR» для решения подобных задач состоит в следующих этапах:

1. Выделение различных видов информации, описание способов вычислений близости по каждому виду.
2. Формирование линейной комбинации функций близости, настройка коэффициентов (методом покоординатного спуска).
3. «Деформирование» комбинации (попытка построить нелинейную формулу решения путем перебора различных алгебраических выражений), настройка коэффициентов (методом покоординатного спуска).

В этой задачи на первом этапе были разработаны способы сравнения похожести заголовков лекции, их тематики, авторов и т.д. Например, при сравнении заголовков применяется стандартный подход к обработке текстов: стемминг, TF·IDF-преобразование и косинусная мера сходства [13]. «Вид информации» здесь понимается в широком смысле, например, бралась вся текстовая информация (заголовок лекции, описание, текст из слайдов), объединялась, и сравнивались такие общие тексты.

На втором этапе применение очень простого оптимизационного метода – покоординатного спуска – оправдано тем, что приходится максимизировать нестандартный функционал качества (7) (он максимизировался напрямую). Замечено, что автоматически определяются «ненужные» виды информации: те, которые не нужно учитывать в решении – им соответствуют нулевые веса. Важность этого наблюдения состоит в том, что деформация не сделает их «нужными», т.е. если на втором этапе некоторый коэффициент нулевой, то он будет нулевым и при настройке коэффициентов на третьем этапе (поэтому сразу лишние слагаемые можно удалить). Естественно, это наблюдение теоретически не обосновано.

На третьем этапе происходит перебор различных φ в выражении вида (1) (из некоторого списка функций). В качестве B_i здесь брались подкомбинации линейной комбинации построенной на первых двух этапах. Конкретно в этой задаче наилучшей оказалась деформация вида $\varphi(x) = \sqrt{x}$.

В результате строится функция близости двух лекций вида (1). Все параметры в ней настроены так, чтобы близкими считались лекции, которые смотрят вместе. Для «старой» лекции находим 30 новых с наибольшими значениями близостей к ней (в соответствующем порядке их и выдаем).

Видно, что идея описанной технологии соответствует теоретическим результатам, описанным в обзоре. Разработанный алгоритм занял первое место среди 62 участников с результатом 35.857% и достаточно солидным отрывом от второго места (30.743%). Более подробное описание алгоритма можно найти в [11].

Задача классификации биомедицинских статей

На Международном соревновании «Topical Classification of Biomedical Research Papers» [14] была представлена задача классификации биомедицинских научных статей на $l = 83$, вообще говоря, пересекающихся класса. Каждый класс описывал принадлежность к определенной теме («хирургия», «стоматология» и т.п.). Каждая статья описывалась значениями $n = 25640$ признаков, природа признаков участникам соревнования не разглашалась, но скорее всего это были числа вхождений определенных терминов («трансплантация», «зуб» и т.п.). В обучающей выборке было $m = 10000$ статей, которые вручную раскласифицированы экспертами, в контрольной, по которой оценивалось качество алгоритмов, также было $q = 10000$ статей (их классификация участникам не была известна).

Функционалом качества решения была F-мера: качество ответа алгоритма $(y_1, \ldots, y_l)$ (бинарный вектор, i-й элемент которого – ответ на вопрос, принадлежит ли рассматрива-

емая статья i-му классу) при правильном ответе $(\alpha_1, \ldots, \alpha_l)$ вычислялась как

$$\frac{\sum_{i=1}^{l} \alpha_i y_i}{\sum_{i=1}^{l} \alpha_i + \sum_{i=1}^{l} y_i}.$$

Для определения победителя бралось среднее арифметическое значений F-меры для всех объектов контрольной выборки.

Достаточно качественные решения поставленной задачи (могли бы войти в 10 сильнейших на соревновании) можно получить следующим образом. Поскольку на вход алгоритму дается матрица $X_{m\times n}$, а на выходе – матрица $Y_{m\times l}$, то логично искать матрицу $W_{n\times l}$ такую, что

$$XW = Y. \tag{8}$$

Ясно, что при этом определяется слишком большое число nl неизвестных (параметров модели), что вызывает эффект переобучения (качество на тесте существенно меньше качества на обучении). Поэтому уравнение (8) «упрощается»: вместо матрицы X используют матрицу X', которая получается в результате неполного сингулярного разложения (SVD):

$$X \approx X'_{m\times k} \Lambda_{k\times k} U_{k\times n}.$$

Используя парадигму алгебраического подхода находят C:

$$C(X'W) = Y,$$

где C – пороговое решающее правило (5) (мы в любом случае вынуждены бинаризовать ответ). Наилучшее качество достигалось при использовании $k = 700$ максимальных собственных значений в сингулярном разложении, т.е. матрица X' имела размеры 10000×700.

Следуя универсальной формуле (1), необходимо было подобрать преобразование φ. Из логики решения и с помощью перебора было установлено, что лучше всего подходят преобразования типа нормировки. Наилучшее качество показала нормировка

$$\varphi(||\gamma_{ij}||) = ||\theta_{ij}||, \ \theta_{ij} = \frac{\gamma_{ij}}{\sqrt{\max\limits_{t}(\gamma_{it})}}.$$

Коэффициенты в (1) настраивались методом покоординатного спуска, напрямую оптимизируя F-меру решения. В качестве алгоритмов B_i в (1) были взяты

— Описанный выше линейный алгоритм над SVD-разложением.
— Два LIBSVM-алгоритма [15] (отличались предварительной нормировкой матрицы X, в первом все элементы делились на максимальный элемент в строке, во втором — в столбце).
— Алгоритм k ближайших соседей с косинусной мерой сходства (может быть исключён без сильной потери качества).

При решении задачи оказалось, что порог в пороговом решающем правиле (5) также можно выбирать, повышая качество классификации. Оптимальное значение порога – для i-й строки матрицы оценок $||\gamma_{ij}||$

$$c(i) = \min(\max_{j}(\gamma_{ij}), 0.345)$$

(необходимо было гарантировать отсутствие нулевых строк в матрице-ответе).

Таким образом, в этой задаче также удалось получить решение в виде (1). Здесь B_i брались из разных семейств алгоритмов, в остальном применение формулы (1) стандартное (каждый оператор B_i получал вещественную $q \times l$-матрицу, матрицы «деформировались» нормировками и складывались с определенными коэффициентами). Итоговое решение показало результат 0.53242 (по F-мере) и заняло 3 место на соревновании среди 126 участников (0.53579 — результат команды победителей). Подробнее о решении написано в [16].

Задача предсказания поведения клиентов сети супермаркетов

На Международном соревновании «Dunnhumby's Shopper Challenge» [17] была представлена задача прогнозирования даты следующего визита и суммы покупок каждого клиента сети супермаркетов. Техника решения самой задачи немного выходит за рамки данной статьи, поскольку ответ алгоритма для данного клиента считался верным, если точно угадывался день следующего визита и с точностью до 10$ угадывалась сумма. Таким образом, задача разбивалась на две разные подзадачи, которые требовалось решить согласованно (например, траты клиента могут зависеть от «типичности дня визита»).

Для нас интересно решение первой подзадачи — прогнозирования даты визита. Пусть у нас есть матрица визитов конкретного клиента

$$||v_{ij}||_{d\times 7},$$

в которой $v_{ij} = 1$, если был визит в j-й день i недель назад (если сейчас конец воскресения, то v_{12} соответствует вторнику этой недели, а v_{25} — пятнице прошлой), d — число недель, за которое есть статистика.

Есть несколько различных способов оценки вероятности следующего прихода в каждый конкретный день. Первый способ — оценить вероятности визитов

$$p_j = \frac{1}{d}\sum_{i=1}^{d} v_{ij}, \tag{9}$$

а затем пересчитать их в искомые вероятности

$$\tilde{p}_j^1 = p_j \prod_{k=1}^{j-1}(1 - p_k). \tag{10}$$

Второй способ — оценивать вероятности непосредственно:

$$\tilde{p}_j^2 = \frac{|\{i \in \{1, 2, \ldots, d\} \mid v_{ij} = 1,\ v_{i1} = \cdots = v_{i,j-1} = 0\}|}{d}.$$

Если от матрицы визитов $||v_{ij}||_{d\times 7}$ перейти к матрице первых визитов $||v'_{ij}||_{d\times 7}$, оставив только первые ненулевые элементы в каждой строке, то формула для второго метода запишется по аналогии с (9):

$$\tilde{p}_j^2 = \frac{1}{d}\sum_{i=1}^{d} v'_{ij}.$$

Это наблюдение будет в дальнейшем использовано следующим образом. Разумно комбинацией двух методов назвать метод, который вычисляет вероятности так:

$$\tilde{p}_j = \alpha\tilde{p}_j^1 + (1 - \alpha)\tilde{p}_j^2,\ \alpha \in [0, 1].$$

Однако, поскольку в решении будет использована деформация (которая должна «одинаково действовать на алгоритмы»), учитывая схожесть вычислений p_j и $\tilde{p}_j^2$, мы будет оценивать вероятности визитов по формуле

$$p_j^{\text{new}} = \alpha p_j + (1-\alpha)\tilde{p}_j^2,$$

а затем пересчитывать их в вероятности первых визитов (10). Это не совсем правильно с точки зрения теории, но дало выигрыш на практике. В качестве ответа выдается день, в который следующий визит максимально вероятен:

$$\arg\max_j \tilde{p}_j.$$

Далее логика решения простая. Надо определиться с «деформацией». Поскольку статистика дана за длительный период, надо учесть «устаревание» данных (элементы из первых строк матриц $||v_{ij}||$, $||v'_{ij}||$ более полезны для решения, чем элементы последних).

Используя простейшую весовую схему $w_1 \geqslant w_2 \geqslant \ldots \geqslant w_d \geqslant 0$, $\sum_{i=1}^d w_i = 1$, получаем оценку вероятности визита

$$p_j^{\text{new}} = \alpha \sum_{i=1}^d w_i v_{ij} + (1-\alpha) \sum_{i=1}^d w_i v'_{ij} = \sum_{i=1}^d w_i (\alpha v_{ij} + (1-\alpha) v'_{ij}).$$

В этой задаче в качестве весовой схемы была выбрана «степенная»:

$$w_i = \frac{(d-i+1)^r}{\sum_{k=1}^d k^r}$$

(хотя возможен и выбор других схем). Параметры r, α были выбраны так, чтобы максимизировать число верных прогнозов на последней неделе: статистика «обрезалась» 7 дней назад и предсказывались первые визиты каждого клиента (если они были) на последней неделе.

В результате был построен алгоритм, который занял первое место на соревновании среди 279 команд [17]. Алгоритм способен предсказывать день следующего визита с точностью 43%, но финальная версия показывала лишь 41.79%, поскольку параметры настраивались с учётом второй подзадачи (предсказывался не самый вероятный визит, поскольку учитывалась дисперсия цен покупок в эти дни недели).

Деформация ответов

«Деформации ответов» алгоритмов, т.е. применение к ответам некоторой функции φ часто используются прикладниками. При этом они называются по-разному («деформация» — авторский термин), например, «calibrating» [18]. Часто они бывают полезными при подобных функциях ошибки:

$$-\frac{1}{q}\sum_{i=1}^q \begin{cases} \log(1-a_i), & y_i = 0, \\ \log(a_i), & y_i = 1 \end{cases} \tag{11}$$

(Logarithmic Loss — логарифм функции правдоподобия распределения Бернулли), где y_i — верный ответ в задаче классификации с двумя классами $\{0,1\}$ на i-м объекте контрольной выборки из q объектов, а $a_i \in [0,1]$ — ответ алгоритма (он может лежать в отрезке).

Особенность функции (11) в том, что «она не прощает ошибок»: если $a_i = 1 - y_i$, то штраф равен бесконечности.

Если про объект известно, что с вероятностью p он принадлежит классу 1, тогда матожидание ошибки на нем

$$(1-p)\log(1-a) + p\log(a).$$

Приравняв к нулю производную, получаем, что $a = p$ — оптимальный ответ с точки зрения рассматриваемого функционала ошибки.

Таким образом, в случае функционала (11) надо выдавать вероятность. Поскольку алгоритмы не всегда вычисляют именно вероятность, они могут проигрывать простым наивным байесовским техникам по функционалу (11). Выход — перевод ответа в вероятность. Часто срабатывает такой прием: для $a \in [0,1]$ берутся все объекты из отложенного контроля (на них не обучался наш алгоритм), такие, что ответы нашего алгоритма на них попали в отрезок $[a-\varepsilon, a+\varepsilon]$. Функция φ в точке a полагается равной среднему арифметическому меток классов этих объектов (или взвешенной линейной комбинации, с весами, зависящими от расстояния). Подробнее об улучшении качества таким способом в реальных задачах анализа данных можно узнать из дипломной работы «Задачи анализа данных с нестандартным функционалом качества» дипломницы автора Ермушевой А.А. [19].

Таким образом, в случае некоторых функций ошибок, деформацию удается строить в явном виде (задавая значения в каждой точке), а потом использовать стандартную схему, описанную в этой работе. Интересно, что часто деформация похожа на сигмоиду (т.е. в нейронных сетях используется «правильная» функция активации — такое искривление ответов, как правило, и нужно для получения приемлемого качества).

Заключение

Мы описали универсальную форму представления функций и алгоритмов, а также способы её реализации на практике. Естественно, эти способы не претендуют на создание универсального алгоритма решения задач классификации, регрессии или построения рекомендаций. Тем не менее несколько побед на крупных соревнованиях по анализу данных автору удалось одержать благодаря такому «правильному построению алгоритма». При этом решались совершенно разные задачи: рекомендации, классификации и прогнозирования.

Доклад по схожей тематике впервые делался автором в 2012 г. на научной школе КРОМШ, статья с аналогичным теоретическим обзором (меньшим по объему и без доказательств) была подана в журнал «Spectral and Evolution Problems» (также содержала лишь один пример прикладной задачи).

В данной работе было представлено несколько реальных прикладных задач, впервые опубликовано решение задачи о визитах клиентов, предложены схемы доказательств теорем о представлении алгоритмов в виде (1) (для строгих доказательств потребовалось бы погружение в формалистику алгебраического подхода).

Когда работа была уже подписана в печать, с автором связался профессор А. Н. Горбань и сообщил, что результат, из которого следует, что для приближения непрерывной функции можно использовать нейросети с произвольной нелинейной функцией активацией (и тождественной) был получен еще в статье [20] (почти на 30 лет раньше В. Крейновича [4]). Автор выражает благодарность А. Н. Горбаню за внимание к этой работе.

Литература

[1] *Колмогоров А. Н.* О представлении непрерывных функций нескольких переменных в виде суперпозиции непрерывных функций одного переменного // *Докл. АН СССР.* 1957. Т. 114, №. 5. С. 953–956.

[2] *Хайкин С.* Нейронные сети. Полный курс. — М.: Вильямс, 2006.

[3] *Горбань А. Н.* Обобщенная аппроксимационная теорема и вычислительные возможности нейронных сетей // *Сибирский журнал вычислительной математики.* 1998. Т. 1, №. 1. С. 12–24.

[4] *Kreinovich V. Y.* Arbitrary nonlinearity is sufficient to represent all functions by neural networks: A theorem // *Neural Networks.* 1991. Vol. 4, no. 3. C. 381–383.

[5] *Leshno M., Lin V. Ya., Pinkus A., Schocken S.* Multilayer feedforward networks with a nonpolynomial activation function can approximate any function // *Neural Networks.* 1993. No. 6. P. 861–867.

[6] *Pinkus A.* TDI-subspaces of $C(\mathbb{R}^d)$ and some density problems from neural networks // *Journal of Approximation Theory.* 1996. Vol. 85. P. 269–287.

[7] *Pinkus A.* Density in approximation theory // *Surveys in Approximation Theory.* 2005. No. 1. P. 1–45.

[8] *Журавлёв Ю. И.* Корректные алгоритмы над множествами некорректных (эвристических) алгоритмов. I–II // *Кибернетика.* 1977. №. 4. С. 5–7; 2011. №. 6. С. 21–27.

[9] *Дьяконов А. Г.* Теория систем эквивалентностей для описания алгебраических замыканий обобщенной модели вычисления оценок // *Ж. вычисл. матем. и матем. физ.* 2010. Т. 50, №. 2. С. 388–400; 2011. Т. 51, №. 3. С. 529–544.

[10] *Дьяконов А. Г.* Алгебра над алгоритмами вычисления оценок: нормировка и деление // *Ж. вычисл. матем. и матем. физ.* 2007. Т. 47, №. 6. С. 1099–1109.

[11] *Дьяконов А. Г.* Алгоритмы для рекомендательной системы: технология LENKOR // *Бизнес-Информатика.* 2012. №. 1(19). С. 32–39.

[12] *Antulov-Fantulin N., Bošnjak M., Šmuc T., Jermol M., Žnidaršič M., Grčar M., Keše P., Lavrač N.* ECML/PKDD 2011 — Discovery challenge: "VideoLectures.Net Recommender System Challenge." 2012. `http://tunedit.org/challenge/VLNetChallenge`.

[13] *Маннинг К. Д., Рагхаван П., Шютце Х.* Введение в информационный поиск. М.: Вильямс, 2011.

[14] Международное соревнование "JRS 2012 Data Mining Competition: Topical Classification of Biomedical Research Papers." 2012. `http://tunedit.org/challenge/JRS12Contest`.

[15] Библиотека для решения задач классификации и регрессии на базе метода SVM. 2012. `http://www.csie.ntu.edu.tw/~cjlin/libsvm/`.

[16] *D'yakonov A.* A blending of simple algorithms for topical classification // *Rough Sets and Current Trends in Computing, Lecture Notes in Computer Science.* 2012. Vol. 7413. P. 432–438.

[17] Международное соревнование "dunnhumby's Shopper Challenge." 2012. `http://www.kaggle.com/c/dunnhumbychallenge`.

[18] *Bostrom H.* Calibrating random forests // *Proceedings of the 2008 7th International Conference on Machine Learning and Applications.* 2008. P. 121–126.

[19] Страница спецсеминара Дьяконова А.Г. на факультете ВМК МГУ. 2012. `http://www.machinelearning.ru/wiki/index.php?title=Алгебра_над_алгоритмами_и_эвристический_поиск_закономерностей`.

[20] *Leeuw K., Katznelson Y.* Functions that operate on non-self-adjoint algebras // *J. d'Anal. Math.* 1963. Vol. 11. P. 207–219.