Российская академия наук
Отделение математических наук РАН
Вычислительный центр им. А. А. Дородницына РАН

Национальная академия наук Украины
Крымский научный центр НАНУ
Крымская академия наук
Таврический национальный университет им. В. И. Вернадского

Национальная академия наук Беларуси

при поддержке

Российского фонда фундаментальных исследований
Компаний Форексис и ЦСПиР

Интеллектуализация обработки информации

ИОИ-9

Черногория, г. Будва, 2012

Доклады 9-й международной конференции

ТОРУС ПРЕСС МОСКВА 2012

УДК 004.85+004.89+004.93+519.2+519.25+519.7
ББК 22.1:32.973.26-018.2
И73

Интеллектуализация обработки информации: 9-я международная конференция. Черногория, г. Будва, 2012 г.: Сборник докладов. — М.: Торус Пресс, 2012. — 718 с.
И73
ISBN 978-5-94588-119-8

В сборнике представлены доклады 9-й Международной конференции «Интеллектуализация обработки информации–2012», проводимой Вычислительным центром им. А. А. Дородницына РАН, Таврическим национальным университетом им. В. И. Вернадского Национальной академии наук Украины и Национальной академией наук Беларуси при финансовой и организационной поддержке РФФИ и компаний Форексис и ЦСПиР.

Конференция проводится с 1989 года; начиная с 2000 года — регулярно один раз в два года, и является представительным научным форумом в области интеллектуального анализа данных, машинного обучения, распознавания образов и анализа изображений, обработки сигналов, дискретного анализа.

УДК 004.85+004.89+004.93+519.2+519.25+519.7
ББК 22.1:32.973.26-018.2

ISBN 978-5-94588-119-8

Некоторые технологии решения задач анализа данных*

Вороненко А. А., Дьяконов А. Г.
dm6@cs.msu.ru, djakonov@mail.ru
Москва, Московский государственный университет им. М. В. Ломоносова

В работе описаны общие методы решения современных прикладных задач анализа данных. Методы основаны на успешном авторском опыте участия в крупных Международных турнирах по анализу данных и часто позволяют получать не просто приемлемые, а лучшие результаты.

Some technologies for data mining problem solving*

Voronenko A. A., D'yakonov A. G.
Moscow State University, Moscow, Russia

The common methods for modern applied data mining problem solving are described. The methods are based on successful author's participation in large international data mining contests. They often allow to receive not only high but also the best results.

Анализ данных (data mining) развивается в последние годы стремительными темпами. Теории, которые недавно казались устоявшимися, не справляются с огромной массой новых проблем. Увеличение объемов данных, специфика входной информации и особые требования к решениям порождают новые классы прикладных задач. При этом более-менее успешно работают эвристические доработки «старых испытанных» методов: основанных на близости, случайных деревьях и бустинге.

Данная работа посвящена некоторым технологиям решения современных задач, но технологиям не в широком смысле (проблемам математической формулировки, хранения информации, средствам разработки алгоритмов, проектированию процесса решения, внедрению и т. д.), а в конкретном узком: что и как делать аналитику, чтобы построить алгоритм приемлемого качества. Описанные подходы быстро и просто реализуются в современных системах (например, в MatLab и R).

Главная особенность работы в том, что все описанные методы верифицированы на реальных, актуальных для бизнеса задачах, в рамках последних турниров платформ Kaggle [1] и TunedIT [2] (а не на таблицах из репозиториев). Таким образом, не нуждается в обосновании их эффективность и не требуется сравнение с другими технологиями.

Современные задачи data mining

Из основных проблем, с которыми сталкивается исследователь (большой объем начальной информации, особые ограничения на алгоритм, нетрадиционная постановка и т. д.), особо следует выделить нестандартные функционалы качества решений, поскольку они негладкие, не всегда допускают хорошие гладкие аппроксимации и под них «не заточены» классические методы.

Работа выполнена при финансовой поддержке гранта Президента МД-757.2011.9 и РФФИ, проект № 12-07-00187.

В задачах классификации вместо традиционного «процента верных ответов» все чаще используют функционал AUC-ROC (площадь под ROC-кривой) [3]. При этом в задаче с двумя классами $\{0, 1\}$ алгоритму разрешают выдавать значения из отрезка $[0, 1]$. Особенно этот функционал популярен в задачах скоринга (предсказание кредитоспособности потенциального клиента банка).

В последнее время также часто используют

$$-\frac{1}{q}\sum_{i=1}^{q}\big(y_i\log(a_i)+(1-y_i)\log(1-a_i)\big)\,; \qquad (1)$$

(capped binomial deviance), где $y_i \in \{0, 1\}$ — верная классификация i-го контрольного объекта; $a_i \in [0, 1]$ — ответ алгоритма; q — число контрольных объектов. Если алгоритм полностью ошибается ($a_i = 1-y_i$) хотя бы на одном объекте, то ошибка алгоритма считается равной $+\infty$.

В задачах с большим числом l пересекающихся классов, где ответ выглядит как бинарный вектор $y_i = (y_i^1, \ldots, y_i^l)$, часто используют F-меру:

$$\frac{2}{q}\sum_{i=1}^{q}\left(\frac{\sum_{j=1}^{l} y_i^j a_i^j}{\sum_{j=1}^{l}(y_i^j + a_i^j)}\right); \qquad (2)$$

которая раньше была традиционна в задачах информационного поиска [4].

В задачах с более сложным форматом ответа возникают еще «более недифференцируемые» функции. Например, при синтезе рекомендательных систем, когда алгоритм должен «угадать» список рекомендаций $(s_1, \ldots, s_S)$, его ответ $(r_1, \ldots, r_R)$ оценивается по формуле

$$\sum_{z\in Z}\frac{\{s_1,\ldots,s_{\min(S,R,z)}\}\cap\{r_1,\ldots,r_{\min(S,R,z)}\}}{|Z|\min(S,R,z)}\,; \qquad (3)$$

Z — некоторое множество натуральных чисел. Таким образом учитывается порядок в списке.

Ниже перечислим некоторые примеры современных задач, на которых были верифицированы описанные в работе технологии. Отметим, что только одна из них признаковая (задана таблица «объект-признак»), но и в ней признаки не самой простой природы (тексты, названия файлов и т.п.). В перечне указываем название Международного турнира, на котором была решена задача, ссылку на платформу, функционал качества и результат.

Предсказание правильности ответов студента на вопросы теста. «What Do You Know» [1], (1), 2-е место из 252 (0.247 CBD). Дана статистика ответов студентов на вопросы тестов: кто, когда и на какой вопрос отвечал, как ответил, сколько времени потратил, в каком режиме (экзамен, промежуточный зачет или тренировка), категория вопроса и т. д. Необходимо разработать алгоритм, предсказывающий правильность ответов конкретного студента на вопросы теста. Подобный алгоритм может быть полезен в качестве рекомендательной системы: он сообщает студенту, какие вопросы могут у него вызвать трудности, и советует повторить соответствующие темы.

Оценка фотографий по метаданным. «Photo Quality Prediction» [1], (1), 3/212 (0.185 CBD). Под метаданными понимается информация не касающаяся непосредственно изображения. Необходимо по обучающей выборке (эксперты оценили некоторые фотографии) разработать алгоритм, который определяет «привлекательность фотографии» по названию, описанию, количеству фотографий в альбоме, размеру сторон, GPS-координатам съемки и т. д. Подобный алгоритм может применяться на крупном файлообменном ресурсе для предварительной классификации загруженных изображений с целью дальнейшего выделения кандидатов на иллюстрации в журналах, выставление на заглавную страницу и т. п.

Разработка рекомендательной системы. «`VideoLectures.Net` Recommender System Challenge» [2], (3), 1/62 (0.359). Известна статистика просмотров пользователями лекций из некоторого репозитория, а также описание каждой лекции: название, аннотация, слайды, предмет, авторы, дата съемки, дата загрузки на сайт и т. д. Необходимо разработать систему, которая новому пользователю (который просмотрел только одну лекцию) рекомендует что-нибудь из новинок (для которых еще нет статистики просмотра). Актуальность разработок подобных систем не вызывает сомнений.

Автоматическая классификация тестов. «JRS 2012 Data Mining Competition: Topical Classification of Biomedical Research Papers» [2], (2), 3/126 (0.532 F-мера). Необходимо создать автоматический рубрикатор медицинских научных статей (заданных числами вхождений специальных терминов) по 83 пересекающимся классам-рубрикам.

Также технологии были верифицированы на стандартной задаче скоринга «Give Me Some Credit» [1], 4-е место из 970, и задаче ранжирования документов «Интернет-математика 2011» [5], 3-е место из 126 (участвовали студенты авторов).

Технология «ЛЕНКОР»

Помимо теоремы А. Н. Колмогорова о представлении функции многих переменных с помощью суперпозиций и сумм функций одного переменного, которая является теоретическим базисом в теории нейронных сетей [3], есть почему-то малоизвестный, но очень изящный результат [6]. Из него следует, что любая непрерывная функция может быть сколь угодно точно приближена с помощью двухслойной нейронной сети, причем в последнем слое функция активации тождественная, а в первом — фиксированная — произвольная непрерывная, отличная от полинома.

Представим аналог этого результата в алгебраическом подходе Ю. И. Журавлева [7]. За основную модель в теореме взята модель алгоритмов вычисления оценок (АВО), хотя можно использовать и многие другие модели (терминологию см. в [7], [8]).

Теорема 1. *Пусть $F\colon \mathbb{R} \to \mathbb{R}$ — непрерывная функция, отличная от полинома, тогда в любой регулярной задаче корректный алгоритм можно построить в виде*

$$\big(c_1 F(B_1) + \ldots + c_r F(B_r)\big) \circ C\,, \qquad (4)$$

где $B_1, \ldots, B_r$ — операторы из линейного замыкания АВО; C — пороговое решающее правило (или любое корректное [7] *решающее правило).*

Здесь, как принято в алгебраическом подходе, $F(B)$ обозначает оператор, который получает матрицу оценок поэлементным применением функции F к матрице оценок оператора B. В случае, когда F является полиномом, алгоритмы (4) исследованы в [8]. При $F = x^k$ вид (4) описывает все алгоритмы из алгебраического замыкания k-й степени. Аналогичная теория справедлива для операторов, матрицы оценок которых имеют «противоположный метрический смысл»: ij-й элемент матрицы описывает не оценку близости i-го объекта к j-му классу, а расстояние (см. [8]).

Подобные результаты, возникающие в различных областях теории классификации, регрессии и интерполяции, наводят на мысль о виде «универсального» алгоритма (4). На практике его удалось реализовать, «скрестив» с классическим методом k ближайших соседей (kNN).

Итак, будем решать задачу взвешенным алгоритмом kNN [3], в котором метрику ищем в виде

$$c_1\rho_1 + \ldots + c_r\rho_r\,. \qquad (5)$$

Первый шаг — задание метрик $\rho_1, \ldots, \rho_r$. Они должны быть достаточно простыми и описывать сходства объектов по частям имеющейся информации. Результаты [8] показывают, что можно ограничиться l_1-метриками, хотя в задачах с текстовыми описаниями лучше работают классические косинусные меры сходства [4]. Подобные метрики легко придумать даже в задаче без признакового описания (они описывают сходство множеств, текстов и т. д., см. [9]).

Второй шаг — настройка коэффициентов в (5). Это делается обычным покоординатным спуском, причем сразу на нужный функционал качества. Как правило, при этом многие веса обращаются в ноль. Соответствующие метрики можно удалять или заменять новыми. Строго не доказано, но практика подтверждает, что достаточно быстро определяется, стоит ли вообще использовать какую-то информацию в задаче. Если простейшая метрика ρ_j, обычно Хэмминга, в линейной комбинации (5) не улучшает, а наоборот ухудшает качество при увеличении c_j, то данный вид информации можно не использовать. Если качество не меняется, то можно попробовать усложнить метрику.

Одновременно на втором шаге происходит оптимизация по числу соседей k и выбору весовой схемы. Как правило, достаточно перебрать весовые схемы следующего типа: вес t-го соседа.

$$w_t = \frac{(k-t+1)^p}{\sum_{i=1}^{k} i^p}, \quad t \in \{1, \ldots, k\}\,. \qquad (6)$$

Чем выше степень в весах, тем требуется большее число соседей для достижения хорошего качества. Кстати, есть совсем «провальные» весовые схемы, например веса $w_t = 1/t,\ t \in \{1, \ldots, k\}$ (авторам не удалось пока найти ни одной реальной задачи, в которой бы они работали лучше, чем (6)).

Третий шаг — «деформация линейной комбинации». Вместо выражения (5) используем выражения типа

$$\sum_i d_i F_i\Big(\sum_j c_j\rho_j\Big), \quad d_i, c_i \in \mathbb{R}\,, \qquad (7)$$

где нелинейные монотонные функции F_i находятся перебором (например, с помощью генетических алгоритмов). После удаления на втором этапе «ненужных» метрик, в выражении (7) не слишком много слагаемых, поэтому перебор возможен за приемлемое время.

Решение *задачи о рекомендательной системе* с помощью описанной технологии подробно изложено в [9]. В *задаче классификации статей* нужно было выдавать в качестве ответов бинарные векторы строгой принадлежности. Поэтому весовой k-NN алгоритм реализовывать не было необходимости: он бывает очень полезен именно для настройки ответов из отрезка $[0, 1]$. Вместо этого использовалось «усложненное решающее правило»:

$$\big(c_1F(H_1C_1) + \ldots + c_rF(H_rC_r)\big) \neq c\,,$$

где H_i — матрицы оценок; C_i — матрицы, которые определялись как решение регрессионной задачи $H_iC_i = Y$; Y — верный ответ; $\neq$ — сравнение с порогом. Более подробно см. в [10].

Случайные леса (RF) и бустинг

Среди прикладников бытует мнение, что «бустинг над решающими деревьями и случайные леса (RF)» — самые лучшие методы. Поэтому решение многих задач сводится к генерации признакового пространства и настройке этих методов.

Итак, *шаг первый* — генерация признаков. На них можно не экономить, поскольку алгоритмы сами уберут шумовые.

Известно, что качество можно повысить (правда, ненамного), добавив линейные комбинации признаков. Разные авторы объясняют этот эффект по-разному (см. [11]). Видимо, он связан с тем, что решающие деревья пытаются аппроксимировать границы классов кусочно-линейно: гиперплоскостями, ортогональными осям координат. После добавления линейных комбинаций получаются более сложные кусочно-линейные поверхности. Основная проблема — как и что добавлять в признаковое пространство.

Мы предлагаем следующий способ: как и в методе «LENKOR» будем использовать покоординатный спуск для настройки линейной комбинации, теперь уже признаков, при этом будем оптимизировать заданный в задаче функционал качества. Естественно, при этом комбинация должна «иметь смысл ответа», например в двухклассовой задаче принимать значения из отрезка $[0, 1]$ (этого можно добиться, настраивая сигмоиду от линейной комбинации). Чтобы добавить несколько разных признаков, настраивая каждый следующий, меняем веса только у тех коэффициентов, которые не были максимальными по модулю при предыдущих настройках. Таким образом, мы подаем на вход случайному лесу признаковые описания и результаты работы простых нейронных сетей.

Подобные схемы известны способностью переобучаться. Этого не происходит, если правильно генерировать признаковое пространство: сразу включать в него «логичные признаки». При применении этой схемы в классической задаче, с матрицей «объект–признак», естественно, происходит переобучение.

Шаг второй — настройка алгоритмов. Как правило, сразу удается понять, следует ли использовать бустинг. Настройка заключается в переборе параметров алгоритмов и определении качества на скользящем контроле. Для экономии времени чаще используется отложенный контроль. Важно понимать, что выбор контроля также определяется функционалом качества и алгоритмом. Например, с помощью скользящего контроля нельзя отлаживать центроидные алгоритмы [10].

Интересно, что RF годится практически для всех нестандартных функционалов качества. Алгоритм универсален в том смысле, что «настраивается сразу под все». Ответы бустинговых алгоритмов иногда приходится «деформировать», чтобы улучшить качество, например, по формуле:

$$y_i \to y_i + \alpha y_i(1 - y_i)\,.$$

Если под переобучением понимать эффект, когда невозможно спрогнозировать ошибку на отложенной выборке по ошибке на обучении, то случайный лес практически не подвержен переобучению при наращивании числа деревьев в лесе. Бустинговые схемы, напротив, сильно переобучаются, и увеличение числа деревьев приводит к ухудшению качества.

Шаг третий — построение смеси алгоритмов. Хотя сами по себе случайные леса являются сложными композициями алгоритмов, простая линейная комбинация с SVM или взвешенным ближайшим соседом часто повышает качество.

Мы предлагаем строить линейную комбинацию, как и в первом подходе, в виде (4). Причем включать в нее несколько случайных лесов, построенных на разных признаковых множествах. Часто, когда по смыслу удается разбить признаки на группы: $I = \{1, \ldots, n\} = I_1 \cup \ldots \cup I_s$, хорошим набором алгоритмов для композиции являются леса, обученные на признаках из $I \backslash I_j$, $j \in \{0, \ldots, s\}$, $I_0 = \varnothing$.

Отметим, что для алгоритмов RF можно не использовать «нелинейности» (функции F в (4)) — они редко дают повышения качества. Видимо, это объясняется тем, что нелинейность уже заложена в самих лесах, кроме того, они и без этого правильно настраиваются на выборку.

Выводы

Мы описали два достаточно общих подхода к решению прикладных задач, выверенных в рамках Международных соревнований ученых-прикладников по анализу данных. Мы постарались описать, насколько позволил объем статьи, все тонкости применения методов.

К сожалению, грамотной литературы на русском языке по прикладному анализу данных почти нет. Чтобы убедиться в этом, достаточно в поисковике набрать некоторые ключевые фразы данной работы. Например, «генерация признаков для случайных лесов» (а ведь это один из мощнейших методов анализа данных) или «взвешенный ближайший сосед» (который при правильной весовой схеме является лучшим из простейших). Кстати, с помощью kNN алгоритма была построена рекомендательная система для ресурса `VideoLectures.Net` (эффективность около 0,359, тогда как у занявшей второе место в соревновании — 0,307 [9]).

К недостаткам подходов можно отнести слишком примитивные способы настройки параметров: покоординатный спуск. Авторы не уверены, можно ли использовать здесь SVM-подобную технику для определения коэффициентов. Примитивность метода продиктована желанием настраиваться именно на нестандартный функционал качества, а не на его сглаженный аналог.

Также в качестве недостатка отметим «эвристичность» некоторых этапов: процедуры выбора метрик в первом подходе и признаков во втором.

Литература

[1] `www.kaggle.com` — Платформа для проведения соревнований по анализу данных. — 2012.

[2] `www.tunedit.org` — Платформа для проведения соревнований по анализу данных. — 2012.

[3] *Воронцов К. В.* Лекции по машинному обучению. — `www.machinelearning.ru`.

[4] *Маннинг К. Д., Рагхаван П., Шютце Х.* Введение в информационный поиск. — М.: ООО «И. Д. Вильямс», 2011. — 528 с.

[5] *Figurnov M., Kirillov A.* Linear combination of random forests for the relevance prediction challenge // Proceedings of the Workshop on Web Search Click Data, WSDM, 2012.

[6] *Pinkus A., Leshno M., Lin V. Ya., Schocken S.* Multilayer Feedforward Networks with a Non-Polynomial Activation Function can Approximate any Function // Neural Networks, 1993. — 6. — Pp. 861–867.

[7] *Журавлев Ю. И.* Корректные алгоритмы над множествами некорректных (эвристических) алгоритмов. II // Кибернетика. — 1977. — № 6. — С. 21–27.

[8] *Дьяконов А. Г.* Теория систем эквивалентностей для описания алгебраических замыканий обобщенной модели вычисления оценок. II // ЖВМиМФ, 2011. — Т. 51, № 3. — С. 529–544.

[9] *Дьяконов А. Г.* Алгоритмы для рекомендательной системы: технология LENKOR // Бизнес-Информатика, 2012. — Т. 19, № 1. — С. 32–39.

[10] `www.alexanderdyakonov.narod.ru/DyakonovReportBIOMED.pdf` — Отчет об используемом в соревновании методе — 2012.

[11] `http://download.yandex.ru/company/SnD.pdf` — *Гуда С., Рябов Д.* Отчет об используемом в соревновании методе. — 2012.