СПЕЦКУРС
Логический анализ данных в распознавании
(Logical data analysis in recognition)

лектор д.ф.-м.н. Елена Всеволодовна Дюкова

Спецкурс посвящён вопросам применения аппарата дискретной математики в задачах интеллектуального анализа данных. Излагаются общие принципы, лежащие в основе логического подхода к задачам машинного обучения. Описываются методы конструирования процедур классификации по прецедентам с использованием понятий теории булевых функций и теории покрытий булевых матриц. Рассматриваются основные модели логических процедур классификации, вопросы сложности их реализации и качества решения прикладных задач.

Спецкурс для бакалавров 2-4 курсов ВМК МГУ им. М.В. Ломоносова.

По спецкурсу издано учебное пособие:
http://www.ccas.ru/frc/papers/djukova03mp.pdf

Лекция 8
О сложности монотонной дуализации. Асимптотически оптимальные алгоритмы монотонной дуализации

• Среди труднорешаемых задач дискретной математики особой сложностью отличаются перечислительные задачи, в которых требуется найти (перечислить) все решения, при этом в типичной ситуации число решений растет экспоненциально с ростом размера задачи (размера входа). Главной перечислительной задачей считается монотонная дуализация (далее дуализация). Важность дуализации обусловлена большим числом приложений. Именно эта задача возникает при построении корректных элементарных классификаторов в логических процедурах классификации.

• Дуализация допускает несколько формулировок. В предыдущей лекции были приведены формулировки этой задачи с использованием понятия неприводимого покрытия булевой матрицы и с использованием понятий теории булевых функций.

• Итак, дана КНФ из $\boldsymbol{m}$ различных элементарных дизъюнкций, реализующая монотонную булеву функцию $\boldsymbol{F(x_1, \dots, x_n)}$. Требуется построить сокращенную ДНФ функции $\boldsymbol{F}$, т.е. найти все максимальные конъюнкции этой функции.

• Монотонная булева функция полностью определяется заданием её верхних нулей и нижних единиц. Каждая дизъюнкция в КНФ задаёт ноль функции $\boldsymbol{F}$. Фактически задаётся множество нулей функции $\boldsymbol{F}$, содержащее множество её верхних нулей. Требуется построить множество нижних единиц этой функции.

• Заметим, что если в искомой ДНФ заменить знаки **&** и ∨ соответственно на знаки ∨ и **&**, то получим КНФ для функции $\boldsymbol{dual\, F}$. По определению $\boldsymbol{dual\, F(\alpha_1, \dots, \alpha_n) = \overline{F}(\overline{\alpha}_1, \dots, \overline{\alpha}_n)}$ для любого двоичного набора $\boldsymbol{(\alpha_1, \dots, \alpha_n)}$. Отсюда название задачи.

• Приведём ещё одну, а именно, гиперграфовую формулировку дуализации. В этой постановке дуализация – это задача перечисления всех минимальных вершинных покрытий гиперграфа с $\boldsymbol{m}$ вершинами и $\boldsymbol{n}$ рёбрами. В графе ребро – это пара его вершин, в гиперграфе ребро – это подмножество заданного множества вершин.

• Связь с задачей построения неприводимых покрытий булевой матрицы $\boldsymbol{L}$ следующая. В роли матрицы $\boldsymbol{L}$ выступает матрица инцидентности гиперграфа размера $\boldsymbol{m} \times \boldsymbol{n}$, которая строится по аналогии с матрицей инцидентности графа. Таким образом, вершинное покрытие гиперграфа – это подмножество его вершин, содержащее хотя бы одну вершину из каждого ребра. Соответственно минимальное вершинное покрытие – это вершинное покрытие, любое собственное подмножество которого вершинным покрытием не является.

- Тривиальный алгоритм дуализации (лекция 5, алгоритм 1) основан на перемножении логических скобок согласно дистрибутивному закону с последующим удалением из построенных конъюнкций повторяющихся переменных и удалением из полученного множества конъюнкций повторяющихся конъюнкций и конъюнкций, поглощаемых другими конъюнкциями. Время работы алгоритма очень быстро растет с ростом n. Это время увеличивается примерно в 2 раза при увеличении n на 1.

- **Пример** работы тривиального алгоритма.

$$(x_1 \vee x_2) \wedge (x_1 \vee x_3) \wedge (x_2 \vee x_4) =$$

$$= (x_1 \wedge x_2) \vee (x_1 \wedge x_4) \vee (x_1 \wedge x_3 \wedge x_2) \vee (x_1 \wedge x_3 \wedge x_4) \vee$$
$$(x_2 \wedge x_1) \vee (x_2 \wedge x_1 \wedge x_4) \vee (x_2 \wedge x_3) \vee (x_2 \wedge x_3 \wedge x_4) =$$

$$= (x_1 \wedge x_2) \vee (x_1 \wedge x_4) \vee (x_2 \wedge x_3) .$$

• Вопрос о существовании более эффективных алгоритмов был поставлен более 40 лет назад. Какие результаты удалось получить?

• Эффективность алгоритмов для перечислительных задач принято оценивать временем выполнения одного шага.

• Алгоритм с полиномиальной временной оценкой (***алгоритм с полиномиальной задержкой***) считается наиболее эффективным. Такой алгоритм на каждом шаге находит в точности одно решение и имеет временную оценку вида $\boldsymbol{O(N)}$, где $\boldsymbol{N}$ - полином от размера входа задачи (полином от $\boldsymbol{m}$ и $\boldsymbol{n}$). Причем оценка даётся для самой сложной индивидуальной задачи (для худшего случая). Требуемые алгоритмы удалось построить для немногих частных случаев дуализации, например, для случая, когда в исходной КНФ каждая элементарная дизъюнкция содержит не более двух переменных. В гиперграфовой постановке это случай графа, в матричной постановке случай, когда каждой строке булевой матрицы $\boldsymbol{L}$ не более двух единичных элементов.

• Наилучший теоретический результат получен в 1995 г. Л. Хачияном с соавторами. Построен алгоритм дуализации гиперграфа с квазиполиномиальной временной оценкой $\boldsymbol{O\left(N^{\log N}\right)}$, где $\boldsymbol{N}$ - полином от размера входа и выхода задачи (полином от $\boldsymbol{m}$, $\boldsymbol{n}$ и числа решений, найденных на предыдущих шагах). Алгоритм с временной оценкой, зависящей от входа и выхода задачи, называют ***инкрементальным.*** Инкрементальному алгоритму разрешено просматривать решения, найденные на предыдущих шагах, поэтому время поиска нового решения очень быстро растёт с ростом числа предыдущих шагов алгоритма. На практике подход применим только в некоторых специальных случаях. Например, когда матрица инцидентности гиперграфа сильно разряжена по числу единиц.

• Таким образом, статус дуализации в плане полиномиальной разрешимости до сих пор неизвестен.

• Далее речь пойдёт о сложности дуализации «в среднем» (для почти всех индивидуальных задач). Будем пользоваться матричной формулировкой задачи.

• Пусть $\boldsymbol{P(L)}$ – множество всех неприводимых покрытий булевой матрицы $\boldsymbol{L}$ размера $\boldsymbol{m \times n}$. Как уже говорилось, при построении $\boldsymbol{P(L)}$ обычно используется следующий критерий. Набор $\boldsymbol{H}$ из $\boldsymbol{r}$ различных столбцов матрицы $\boldsymbol{L}$ является неприводимым покрытием тогда и только тогда, когда выполнены два условия: 1) условие покрываемости; 2) условие совместимости.

• Если выполнено 1), то $\boldsymbol{H}$ – покрытие матрицы $\boldsymbol{L}$ ($\boldsymbol{H}$ покрывает строки матрицы $\boldsymbol{L}$). Если выполнено 2), то $\boldsymbol{H}$ – совместимый набор столбцов матрицы $\boldsymbol{L}$. Совместимый набор столбцов называется максимальным, если он не содержится ни в каком другом совместимом наборе столбцов. Пусть $\boldsymbol{H}$ – совместимый набор столбцов матрицы $\boldsymbol{L}$. Столбец $\boldsymbol{h}$ совместим с $\boldsymbol{H}$, если $\boldsymbol{H \cup h}$ – совместимый набор столбцов, иначе $\boldsymbol{h}$ несовместим с $\boldsymbol{H}$.

• В 1977 г. Е. В. Дюковой предложен подход к построению асимптотически оптимальных алгоритмов дуализации.

• ***Асимптотически оптимальный алгоритм*** $\boldsymbol{A}$ строит $\boldsymbol{P(L)}$ следующим образом. На каждом шаге строится совместимый набор столбцов $\boldsymbol{H}$ матрицы $\boldsymbol{L}$ и для $\boldsymbol{H}$ проверяется условие покрываемости. При этом на каждом шаге выполняется не более, чем $\boldsymbol{d}$ элементарных операций, где $\boldsymbol{d}$ ограничено сверху полиномом от $\boldsymbol{m}$ и $\boldsymbol{n}$. Под элементарной операцией понимается просмотр одного элемента матрицы $\boldsymbol{L}$. **Основное требование:** число шагов $\boldsymbol{N_A(L)}$ алгоритма $\boldsymbol{A}$ должно быть асимптотически равно мощности $\boldsymbol{P(L)}$ при $\boldsymbol{n \to \infty}$ для почти всех булевых матриц $\boldsymbol{L}$ размера $\boldsymbol{m \times n}$.

- Асимптотически оптимальный алгоритм отличается от алгоритма с полиномиальной задержкой тем, что имеет «лишние» полиномиальные шаги. Шаг считается лишним в двух следующих случаях.

- 1) Построенный совместимый набор столбцов уже строился на предыдущих шагах.
- 2) Построенный совместимый набор столбцов ранее не строился, но он не является покрытием.

- Лишний шаг определяется за полиномиальное время от размера входа. Для почти всех булевых матриц размера $\boldsymbol{m \times n}$ число лишних шагов должно иметь более низкий порядок роста по сравнению с числом неприводимых покрытий при росте размера задачи.

- Асимптотически оптимальные алгоритмы построены при условии, что число строк матрицы $\boldsymbol{L}$ существенно меньше числа её столбцов. Обоснование подхода опирается на технику получения асимптотических оценок числа неприводимых покрытий, которая первоначально была предложена в работах В.А. Слепян и В.Н. Носкова, а затем развита в работах Е.В. Дюковой и А.Е. Андреева. В частности, Е.В. Дюковой доказано следующее

- **Утверждение 1**. *Если* $\boldsymbol{\log m \le (1-\varepsilon)\log n}$, *то для почти всех булевых матриц* $\boldsymbol{L}$ *размера* $\boldsymbol{m \times n}$ *при* $\boldsymbol{n \to \infty}$ *мощность* $\boldsymbol{P(L)}$ *асимптотически равна числу единичных подматриц матрицы* $\boldsymbol{L}$.

- В 2011 – 2014 гг. появились публикации японских ученых К. Мураками и Т. Уно, в которых авторы утверждали, что им удалось построить новые алгоритмы дуализации, позволяющие решать задачи большого размера. Обращалось внимание на то, что в основе предлагаемых алгоритмов лежит некий новый принцип, названный авторами условием ***«crit»***. Оказалось, что это условие, сформулированное в указанных публикациях с помощью понятий теории гиперграфов, эквивалентно условию совместимости набора столбцов булевой матрицы. Таким образом, принципиальная схема работы алгоритмов, удовлетворяющих условию ***«crit»***, не отличается от схемы работы асимптотически оптимальных алгоритмов, построенных значительно ранее в отечественных работах.

• Имеет смысл рассматривать в качестве шага асимптотически оптимального алгоритма дуализации построение максимального совместимого набора столбцов. Для наглядности работу такого алгоритма можно представить в виде обхода дерева решений (ДР) в глубину: корнем ДР является пустой набор; вершины ─ совместимые наборы столбцов; висячие вершины ─ максимальные совместимые наборы столбцов, которые либо являются неприводимыми покрытиями, найденными впервые, либо соответствуют лишним шагам.

• Первоначально матрица $\boldsymbol{L}$ рассматривается в качестве текущей матрицы. Построение ДР начинается с вершины, которая порождается некоторым ненулевым столбцом матрицы $\boldsymbol{L}$. При построении каждой из остальных вершин в роли текущей матрицы выступает подматрица матрицы $\boldsymbol{L}$. Для построения дочерней вершины внутренней вершины $\boldsymbol{H}$ алгоритм выбирает в текущей матрице ненулевой столбец и строит новую вершину $\boldsymbol{H} \cup \boldsymbol{h}$, где $\boldsymbol{h}$ - соответствующий столбец матрицы $\boldsymbol{L}$.

• После построения вершины $H \cup h$ текущая матрица изменяется следующим образом. Из неё, как правило, удаляются все строки, покрытые столбцом h, и удаляются столбцы, образованные столбцами матрицы L, несовместимыми с набором столбцов $H \cup h$. Дополнительно могут удаляться и другие строки и столбцы. Если новая текущая матрица не содержит единичных элементов, то вершина $H \cup h$ становится висячей. В этом случае для набора $H \cup h$ проверяется условие покрываемости. Если это условие выполнено, то при необходимости дополнительно проверяется условие повторяемости: набор $H \cup h$ найден впервые. Если выполнены оба указанных условия, то множество неприводимых покрытий, построенных на предыдущих шагах, пополняется, в противном случае это множество не меняется и шаг объявляется лишним. В любом случае далее либо происходит переход к новому шагу путём возврата на более высокий уровень ДР, либо алгоритм заканчивает работу. Если же новая текущая матрица содержит единичные элементы, то $H \cup h$ - внутренняя вершина ДР и процесс построения ветви дерева продолжается.

- Время работы асимптотически оптимального алгоритма в большой степени зависит от сложности ДР (числа вершин в ДР), которое строит этот алгоритм.

- Существуют два вида асимптотически оптимальных алгоритмов: алгоритмы с повторяющимися шагами и алгоритмы без повторяющихся шагов. Исторически первыми появились алгоритмы с повторяющимися шагами, среди которых следует выделить алгоритмы АО1 (Е.В. Дюкова, 1977 г.) и АО2 (Е.В. Дюкова, 2004 г.).

- **Алгоритм АО1** фактически основан на переборе с полиномиальной задержкой $O(qmn)$, где $q = \min(m, n)$, максимальных единичных подматриц матрицы L (единичная подматрица матрицы L называется максимальной, если она не является подматрицей никакой другой единичной подматрицы матрицы L).
- Каждая найденная максимальная единичная подматрица очевидным образом порождает максимальный совместимый набор. Каждый максимальный совместимый набор H строится столько раз, сколько максимальных единичных подматриц он содержит. Причём из этих подматриц в первую очередь строится единичная подматрица, в которой каждая строка имеет наименьший возможный номер. Поэтому проверка на повторяемость осуществляется путём просмотра строк подматрицы, образованной столбцами из H.

- Подробное описание алгоритма АО1 можно найти в учебном пособии:
- **http://www.ccas.ru/frc/papers/djukova03mp.pdf**

- **Алгоритм АО2** является модификацией АО1. Алгоритм основан на построении с полиномиальной задержкой $\boldsymbol{O(qm^2n)}$, где $\boldsymbol{q = \min(m, n)}$, только таких максимальных единичных подматриц, которые порождают неприводимые покрытия. В этом алгоритме лишний шаг – это повторно построенное неприводимое покрытие. Алгоритм АО2 строит менее сложное ДР по сравнению с алгоритмом АО1 и работает быстрее.
- Среди отечественных алгоритмов без повторений следует выделить алгоритмы ОПТ (Е.В. Дюкова, А.С. Инякин, 2008 г.) и RUNC-M (Е.В. Дюкова, П.А. Прокофьев, 2015 г.).
- **Алгоритм ОПТ** перечисляет с полиномиальной задержкой некоторое подмножество максимальных совместимых наборов столбцов, содержащее множество неприводимых покрытий, причем каждый элемент этого подмножества алгоритм выдаёт только один раз. Этот алгоритм строит менее сложное дерево по сравнению с алгоритмом АО2, имея ту же сложность шага, и по скорости счета превосходит алгоритм АО2.

- Наилучшие результаты показывает **алгоритм RUNC-M**, построенный на базе алгоритма ОПТ. В RUNC-M применяется «жадная» стратегия выбора столбцов для построения дочерних вершин в ДР. Всякий раз после построения очередной внутренней вершины $\boldsymbol{H}$ в текущей матрице выбирается строка $\boldsymbol{i}$, имеющая наименьшее число единиц. Дочерняя для $\boldsymbol{H}$ вершина строится путём добавления к набору $\boldsymbol{H}$ столбца матрицы $\boldsymbol{L}$, имеющего наименьший номер среди столбцов, образующих текущую матрицу и покрывающих строку $\boldsymbol{i}$. Сложность шага алгоритма $\boldsymbol{O(qmn)}$, $\boldsymbol{q = \min(m, n)}$. Таким образом, по сравнению с алгоритмом ОПТ алгоритм RUNC-M строит менее сложное ДР и имеет менее сложный шаг.

- Следует отметить, что по скорости счёта асимптотически оптимальным алгоритмам значительно уступают инкрементальные алгоритмы, которые показывают высокую скорость только на первых шагах работы, пока построенное множество решений имеет незначительную мощность.

УПРАЖНЕНИЯ

1. Утверждение 1 сформулировать, используя понятия неприводимой и максимальной конъюнкции монотонной булевой функции.

2. Является ли асимптотически оптимальным алгоритм 2, описанный в лекции 5?

3. Задачу из примера на слайде 5 решить, применяя алгоритм АО1.