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 2
Принципы конструирования классических логических процедур распознавания и основные модели

- При решении прикладных задач достоверная информация о структуре множества M, как правило, отсутствует, поэтому при построении алгоритма распознавания мы не можем гарантировать качество работы этого алгоритма на новых объектах, отличных от прецедентов. Однако, если обучающие примеры достаточно характерны для исследуемого множества объектов, то алгоритм, редко ошибающийся на обучении, будет давать неплохие результаты и на неизвестных (не входящих в обучающую выборку) объектах. В связи с этим большое внимание уделяется проблеме корректности распознающих алгоритмов. Алгоритм называется ***корректным***, если все объекты из обучающей выборки он распознает правильно.
- Будем предполагать, что описания объектов даны в виде значений целочисленных признаков и множество допустимых значений каждого признака ограничено. Рассмотрим простейший пример корректного распознающего алгоритма.

- Распознаваемый объект $\boldsymbol{S}$ будем сравнивать с каждым из объектов обучения $\boldsymbol{S_1, \dots, S_m}$. В случае, если описание объекта $\boldsymbol{S}$ совпадает с описанием обучающего объекта $\boldsymbol{S_i}$, то объект $\boldsymbol{S}$ будем относить к тому классу, которому принадлежит объект $\boldsymbol{S_i}$, в противном случае будем считать, что алгоритм отказывается от распознавания. Нетрудно видеть, что описанный алгоритм является корректным, однако он не сможет распознать ни один объект, описание которого не совпадает с описанием ни одного из обучающих объектов.
- Очевидно, что требование полного совпадения описаний распознаваемого объекта и одного из обучающих объектов является слишком осторожным. Анализ прикладных задач свидетельствует о том, что близость объектов и их принадлежность одному классу можно оценивать на основе сравнения отдельных фрагментов их описаний. Поэтому возникает вопрос, как выбирать такие фрагменты, по которым будут сравниваться объекты. Один из вариантов ответа на данный вопрос используется в модели ***алгоритмов вычисления оценок (АВО).***

- Введем следующие обозначения. Пусть $\boldsymbol{H}$ – некоторый набор из $\boldsymbol{r}, \boldsymbol{r} \leq \boldsymbol{n}$, различных признаков вида $\{\boldsymbol{x_{j_1}}, \dots, \boldsymbol{x_{j_r}}\}$. Близость объектов $\boldsymbol{S'} = (\boldsymbol{a'_1}, \dots, \boldsymbol{a'_n})$ и $\boldsymbol{S''} = (\boldsymbol{a''_1}, \dots, \boldsymbol{a''_n})$ из $\boldsymbol{M}$ по набору признаков $\boldsymbol{H}$ будем оценивать величиной $\boldsymbol{B}(\boldsymbol{S'}, \boldsymbol{S''}, \boldsymbol{H})$, принимающей значение **1**, если $\boldsymbol{a'_{j_t}} = \boldsymbol{a''_{j_t}}$ при $\boldsymbol{t} = \mathbf{1}, \mathbf{2}, \dots, \boldsymbol{r}$, и принимающей значение **0** иначе
- **Принципиальная схема построения алгоритмов АВО следующая.**
- В множестве признаков $\{\boldsymbol{x_1}, \dots, \boldsymbol{x_n}\}$ выделяется совокупность различных подмножеств вида $\{\boldsymbol{x_{j_1}}, \dots, \boldsymbol{x_{j_r}}\}$, $\boldsymbol{r} \leq \boldsymbol{n}$, не обязательно одинаковой мощности. В дальнейшем выделенные подмножества называются ***опорными множествами*** алгоритма, а вся их совокупность обозначается через $\boldsymbol{\Omega}$. Задаются параметры: $\boldsymbol{v_i}$ - параметр, характеризующий представительность (вес), обучающего объекта $\boldsymbol{S_i}$, $\boldsymbol{i} = \mathbf{1}, \mathbf{2}, \dots, \boldsymbol{m}$, и $\boldsymbol{P_H}$ – параметр, характеризующий представительность (вес) опорного множества $\boldsymbol{H}$, $\boldsymbol{H} \in \boldsymbol{\Omega}$. Далее проводится ***процедура голосования*** или ***вычисления оценок***. Распознаваемый объект $\boldsymbol{S}$ сравнивается с каждым обучающим объектом $\boldsymbol{S_i}$, по каждому опорному множеству $\boldsymbol{H}$.
-

- Пусть $\boldsymbol{S_i}$ – обучающий объект, $\boldsymbol{S_i} \in \boldsymbol{K}$, $\boldsymbol{K} \in \{\boldsymbol{K_1}, \dots, \boldsymbol{K_l}\}$, $\boldsymbol{H} \in \boldsymbol{\Omega}$. Считается, что объект $\boldsymbol{S}$ получает голос за принадлежность классу $\boldsymbol{K}$, если описания объектов $\boldsymbol{S}$ и $\boldsymbol{S_i}$ совпадают по опорному множеству $\boldsymbol{H}$ (в этом случае $\boldsymbol{B(S, S_i, H) = 1}$). Для каждого класса $\boldsymbol{K}$, $\boldsymbol{K} \in \{\boldsymbol{K_1}, \dots, \boldsymbol{K_l}\}$, вычисляется оценка принадлежности $\boldsymbol{\Gamma(S, K)}$ объекта $\boldsymbol{S}$ классу $\boldsymbol{K}$, имеющая вид
- $\boldsymbol{\Gamma(S, K) = \frac{1}{N_K} \sum_{S_i \in K} \sum_{H \in \Omega} v_i P_H B(S, S_i, H)}$,
- где $\boldsymbol{N_K}$ - число обучающих объектов из класса $\boldsymbol{K}$.
-
- Объект $\boldsymbol{S}$ относится к тому классу, который имеет наибольшую оценку. Если классов с наибольшей оценкой несколько, то происходит отказ от классификации. Очевидно, что построенный алгоритм не всегда является корректным. Для корректности этого алгоритма требуется выполнение системы линейных неравенств указанного ниже вида.

- Для простоты пусть $\boldsymbol{l = 2}$, $\boldsymbol{S_i \in K_1}$ при $\boldsymbol{1 \le i \le m_1}$, $\boldsymbol{S_i \in K_2}$ при $\boldsymbol{m_1 + 1 \le i \le m}$, $\boldsymbol{1 \le m_1 \le m - 1}$. Тогда система неравенств имеет вид

- $\boldsymbol{\Gamma(S_1, K_1) > \Gamma(S_1, K_2)}$,
- ……
- $\boldsymbol{\Gamma(S_{m_1}, K_1) > \Gamma(S_{m_1}, K_2)}$,
-
- $\boldsymbol{\Gamma(S_{m_1+1}, K_2) > \Gamma(S_{m_1+1}, K_1)}$,
- ……
- $\boldsymbol{\Gamma(S_m, K_2) > \Gamma(S_m, K_1)}$.
-

- Решение приведённой выше системы неравенств заключается в выборе параметров $\boldsymbol{v_i}$, $\boldsymbol{i = 1,2, ..., m}$, и $\boldsymbol{P_H}$, $\boldsymbol{H \in \Omega}$. В случае, если система несовместна, находится ее максимальная совместная подсистема и из решения этой подсистемы определяются значения параметров $\boldsymbol{v_i}$ и $\boldsymbol{P_H}$.

• Другой способ добиться корректности алгоритма – выбрать «хорошую» систему опорных множеств. В частности, выбрать ее так, чтобы для любого обучающего объекта $\boldsymbol{S' \in K}$ было выполнено $\boldsymbol{\Gamma(S', K) > 0}$ и для любого обучающего объекта $\boldsymbol{S'' \notin K}$ было выполнено $\boldsymbol{\Gamma(S'', K) = 0}$. Это можно сделать следующим образом.

• Пусть $\boldsymbol{H}$ – некоторое опорное множество. Набор признаков $\boldsymbol{H}$ называется ***тестом***, если для любых двух обучающих объектов $\boldsymbol{S'}$ и $\boldsymbol{S''}$, принадлежащих разным классам, выполнено $\boldsymbol{B(S', S'', H) = 0}$. Другими словами, тест – это набор признаков, по которому различаются любые два объекта из разных классов.

• Пусть совокупность опорных множеств алгоритма есть некоторое множество тестов $\boldsymbol{\Omega_T}$. Очевидно, такой алгоритм является корректным при любых положительных значениях параметров $\boldsymbol{v_i}$, $\boldsymbol{i = 1, 2, \ldots, m}$, и $\boldsymbol{P_H}$, $\boldsymbol{H \in \Omega_T}$.

- Если набор признаков $\boldsymbol{H_1}$ – тест, то любой набор признаков $\boldsymbol{H_2}$ такой, что $\boldsymbol{H_1 \subset H_2}$, также является тестом. При этом если объекты близки по $\boldsymbol{H_2}$ то они будут близки и по $\boldsymbol{H_1}$, если же два объекта близки по набору признаков $\boldsymbol{H_1}$, то они не всегда будут близки по $\boldsymbol{H_2}$. В этом смысле более короткие тесты обладают большей информативностью, и разумно ограничивать длины тестов или строить тупиковые тесты.

- Набор признаков *H* называется ***тупиковым тестом***, если выполнены следующие два условия 1) $\boldsymbol{H}$ – является тестом; 2) любое собственное подмножество набора $\boldsymbol{H}$ не является тестом. Другими словами, тупиковым тестом является не укорачиваемый набор признаков, по которому любые два обучающих объекта из разных классов отличаются друг от друга.
-
- Оценка $\boldsymbol{\Gamma(S, K)}$ принадлежности распознаваемого объекта $\boldsymbol{S}$ классу $\boldsymbol{K}$ в алгоритме ***голосования по тестам*** (далее тестовом алгоритме) вычисляется также как и в алгоритме АВО.

- Пусть $\boldsymbol{H} = \{\boldsymbol{x_{j_1}}, \dots, \boldsymbol{x_{j_r}}\}$ – некоторый набор признаков, $\boldsymbol{S} = (\boldsymbol{a_1}, \dots, \boldsymbol{a_n})$ – объект из $\boldsymbol{M}$. Фрагмент $(\boldsymbol{a_{j_1}}, \dots, \boldsymbol{a_{j_r}})$ описания объекта $\boldsymbol{S}$ обозначим $(\boldsymbol{S}, \boldsymbol{H})$.
- Каждый тест $\boldsymbol{H}$ порождает множество фрагментов описаний объектов вида $(\boldsymbol{S_i}, \boldsymbol{H})$, $\boldsymbol{i} = \boldsymbol{1, 2, \dots, m}$, где $\boldsymbol{S_i}$ - обучающий объект, причем каждый из этих фрагментов встречается в некотором классе и не встречается в остальных. При распознавании объектов производится голосование по множеству всех таких фрагментов. Рассмотрим простой пример **.**
- $\{\boldsymbol{S_1} = (\boldsymbol{0, 1, 1, 0}),\ \boldsymbol{S_2} = (\boldsymbol{1, 2, 0, 1}),\ \boldsymbol{S_3} = (\boldsymbol{0, 1, 0, 2})\}$ – класс $\boldsymbol{K_1}$
- $\{\boldsymbol{S_4} = (\boldsymbol{1, 2, 1, 0}),\ \boldsymbol{S_5} = (\boldsymbol{1, 1, 0, 1}),\ \boldsymbol{S_6} = (\boldsymbol{1, 1, 1, 2})\}$ – класс $\boldsymbol{K_2}$
- В данном примере наборы признаков $\{\boldsymbol{x_1, x_2, x_3}\}$ и $\{\boldsymbol{x_1, x_2, x_4}\}$ являются тупиковыми тестами. Других тупиковых тестов нет. Если использовать тестовый алгоритм, то объект $\boldsymbol{S} = (\boldsymbol{0, 1, 2, 1})$ не будет отнесен ни к одному из классов $\boldsymbol{K_1}$ и $\boldsymbol{K_2}$. Однако $(\boldsymbol{S_1}, \{\boldsymbol{x_1, x_2}\}) = (\boldsymbol{S_3}, \{\boldsymbol{x_1, x_2}\}) = (\boldsymbol{S}, \{\boldsymbol{x_1, x_2}\})$ и нет ни одного объекта из класса $\boldsymbol{K_2}$, содержащего такой же фрагмент, что даёт нам основание полагать, что распознаваемый объект $\boldsymbol{S}$ более близок к классу $\boldsymbol{K_1}$.

• Таким образом, если при построении алгоритмов классификации перейти от рассмотрения опорных множеств к анализу фрагментов описаний объектов, то можно строить менее осторожные и при этом корректные процедуры. Примерами таких процедур являются **алгоритмы голосования по представительным наборам или алгоритмы типа "Кора" .**

• Фрагмент описания объекта $\boldsymbol{S'}$ из класса $\boldsymbol{K}$ вида $(\boldsymbol{S'}, \boldsymbol{H})$ называется ***представительным набором для*** $\boldsymbol{K}$, если для любого обучающего объекта $\boldsymbol{S''}$, не принадлежащего классу $\boldsymbol{K}$, имеет место $\boldsymbol{B(S', S'', H) = 0}$.

• Фрагмент описания объекта $\boldsymbol{S'}$ из класса $\boldsymbol{K}$ вида $(\boldsymbol{S'}, \boldsymbol{H})$ называется ***тупиковым представительным набором для*** $\boldsymbol{K}$, если выполнены два условия: 1) для любого обучающего объекта $\boldsymbol{S'' \notin K}$ имеет место $\boldsymbol{B(S', S'', H) = 0}$; 2) для любого набора $\boldsymbol{H'}$, $\boldsymbol{H' \subset H}$, найдется обучающий объект $\boldsymbol{S'' \notin K}$, для которого $\boldsymbol{B(S', S'', H') = 1}$.

• В классической модели алгоритма голосования по (тупиковым) представительным наборам для каждого класса $\boldsymbol{K}$ строится множество (тупиковых) представительных наборов, обозначаемое далее через $\boldsymbol{\mathcal{T}(K)}$. Распознавание объекта $\boldsymbol{S}$ осуществляется на основе процедуры голосования. В простейшей модификации для оценки принадлежности объекта $\boldsymbol{S}$ классу $\boldsymbol{K}$ используется величина

• $\boldsymbol{\Gamma(S,K)} = \frac{\mathbf{1}}{|\boldsymbol{\mathcal{T}(K)}|} = \sum_{(\boldsymbol{S'},\boldsymbol{H})\in\boldsymbol{\mathcal{T}(K)}} \boldsymbol{B(S,S',H)}$.

• Очевидно, что все фрагменты описаний обучающих объектов, порожденные некоторым тестом, являются представительными наборами. Очевидно также, что не все представительные наборы, порожденные тупиковым тестом, являются тупиковыми представительными наборами, т.е. в алгоритме голосования по тупиковым представительным наборам строится больше коротких фрагментов описаний объектов, следовательно, он менее осторожный и реже отказывается от распознавания.

- На практике в моделях с представительными наборами отбираются наиболее “весомые” представительные наборы, например те, которые по данному набору признаков встречаются в классе у достаточно большого числа объектов. Рассматриваются также модели с «почти представительными» наборами. В этих моделях информативными для класса $\boldsymbol{K}$ считаются такие фрагменты описаний обучающих объектов, которые по данному набору признаков встречаются достаточно часто в описаниях объектов из класса $\boldsymbol{K}$ и достаточно редко в описаниях объектов из других классов.
- В тестовых моделях для увеличения быстродействия применяются стохастические алгоритмы, в которых построение множества всех тупиковых тестов заменено построением достаточно представительной случайной выборки из него. В данном случае оценки $\boldsymbol{\Gamma\left(S, K_j\right),\ j = 1, 2, \ldots, l}$, вычисляются приближённо и оценивается возможная при этом ошибка.

• Рассматриваемые модели могут быть модифицированы на случай вещественнозначной информации путём задания дополнительных параметров, позволяющих устанавливать близость отдельных значений признаков. Для каждого признака $\boldsymbol{x_j}$, $\boldsymbol{j = 1, 2, \dots, n}$, задаётся некоторый вещественный параметр $\boldsymbol{E_j}$ – точность измерения признака $\boldsymbol{x_j}$. Опять же в простейшей модификации величина $\boldsymbol{B(S', S'', H)}$ определяется следующим образом. Пусть $\boldsymbol{S' = (a'_1, \dots, a'_n)}$ и $\boldsymbol{S'' = (a''_1, \dots, a''_n)}$, $\boldsymbol{H = \{x_{j_1}, \dots, x_{j_r}\}}$. Тогда $\boldsymbol{B(S', S'', H) = 1}$, если $\boldsymbol{\left|a'_{j_t} - a''_{j_t}\right| \le E_{j_t}}$ при $\boldsymbol{t = 1, 2, \dots, r}$, и $\boldsymbol{B(S', S'', H) = 0}$ иначе.

• Для оценки качества работы распознающего алгоритма $\boldsymbol{A}$ часто используется ***процедура скользящего контроля*** (leave one out cross validation), которая заключается в следующем. Для каждого $\boldsymbol{i}$ из $\boldsymbol{\{1, \dots, m\}}$ по обучающей выборке $\boldsymbol{\{S_1, \dots, S_m\} \setminus \{S_i\}}$ вычисляются оценки $\boldsymbol{\Gamma(S_i, K_j)}$, $\boldsymbol{j = 1, 2, \dots, l}$. Пусть $\boldsymbol{q}$ – число правильно распознанных объектов $\boldsymbol{S_i}$. Тогда качество работы алгоритма $\boldsymbol{A}$ оценивается функционалом $\boldsymbol{\varphi_{\text{ск}}(A) = q/m}$.

- Для больших задач описанная процедура скользящего контроля требует существенных вычислительных затрат. В случае, когда $\boldsymbol{A}$ - алгоритм голосования по представительным наборам, для процедуры скользящего контроля существует "быстрый" метод вычисления оценок $\boldsymbol{\Gamma(S_i, K_j\ ),\ i = 1, 2, \ldots, m,\ j = 1, 2, \ldots, l}$. Данный метод позволяет сократить время счета примерно в $\boldsymbol{m}$ раз.
- Для оценки качества работы алгоритма $\boldsymbol{A}$ может быть использован и независимый контроль – набор из $\boldsymbol{t}$ объектов, не вошедших в обучающую выборку, про которые также известно, каким классам они принадлежат. Такой набор объектов называется контрольной выборкой. Пусть $\boldsymbol{q}$ - число правильно распознанных объектов из контрольной выборки. В данном случае качество работы алгоритма $\boldsymbol{A}$ оценивается функционалом $\boldsymbol{\varphi_{\text{контр}}(A) = q/t}$.

- **Замечание.** Среди рассмотренных трёх моделей исторически первыми появились тестовые модели (точнее алгоритмы голосования по множеству тупиковых тестов). Тестовые алгоритмы оказались очень трудоемкими в смысле вычислительных затрат. Теоретические исследования статистических свойств тупиковых тестов показывают, что почти всегда тупиковые тесты имеют примерно одну и ту же длину. Эта длина, вообще говоря, зависит от соотношения между параметрами $\boldsymbol{m}$ и $\boldsymbol{n}$. Указанное обстоятельство послужило одним из обоснований для построения алгоритмов вычисления оценок, в которых в качестве опорных множеств берутся всевозможные наборы признаков, имеющие фиксированную мощность $\boldsymbol{r}$. Значение параметра $\boldsymbol{r}$ может задаваться путем некоторого предварительного анализа обучающей выборки. Для этой модели существует формула эффективного вычисления оценки принадлежности $\boldsymbol{\Gamma(S,K)}$ объекта $\boldsymbol{S}$ классу $\boldsymbol{K}$.

- УПРАЖНЕНИЯ

- 1. Пусть (S', H) – представительный набор класса K. В каком случае при голосовании по (S', H) на скользящем контроле распознаваемый объект S будет отнесен к классу K и в каком случае этот объект будет отнесен к другому классу?
-
- 2. Будем говорить, что признак x_j является фиктивным, если для любого обучающего объекта S', $S' = (a'_1, \dots, a'_n)$, найдется обучающий объект $S'' = (a''_1, \dots, a''_n)$, принадлежащий тому же классу, что и объект S' и такой, что $a'_j \neq a''_j$, $a'_t = a''_t$ при $t \neq j$, $t \in \{1, 2, \dots, n\}$. Доказать, что фиктивный признак не входит ни в один тупиковый тест.