НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
УНИВЕРСИТЕТ

Ленивые методы ассоциативной классификации в машинном обучении

Семинар аспирантской школы по компьютерным наукам

Юрий Кашницкий

ykashnitsky@hse.ru
http://hse.ru/staff/ykashnitsky/

Национальный исследовательский университет
«Высшая школа экономики» (Москва)

28 января 2016

Методы классификации в машинном обучении

Одна из дихотомий методов классификации в машинном обучении:

1. Производящие прогнозную модель (”eager”)
   - Подавляющее большинство алгоритмов
   - Актуальны, когда прогноз необходимо делать быстро (показывать/не показывать баннер, какой товар рекомендовать зашедшему на сайт пользователю)
2. Производящие вычисления только для тестовых объектов (”lazy”)
   - Актуальны, когда прогноз можно делать за долгое время (болен пациент или нет, к какой категории отнести документ)
   - Обучающая выборка просто хранится, возможно, немного предобрабатывается
   - Самый известный пример - метод ближайших соседей

Плюсы
и минусы методов с построением модели

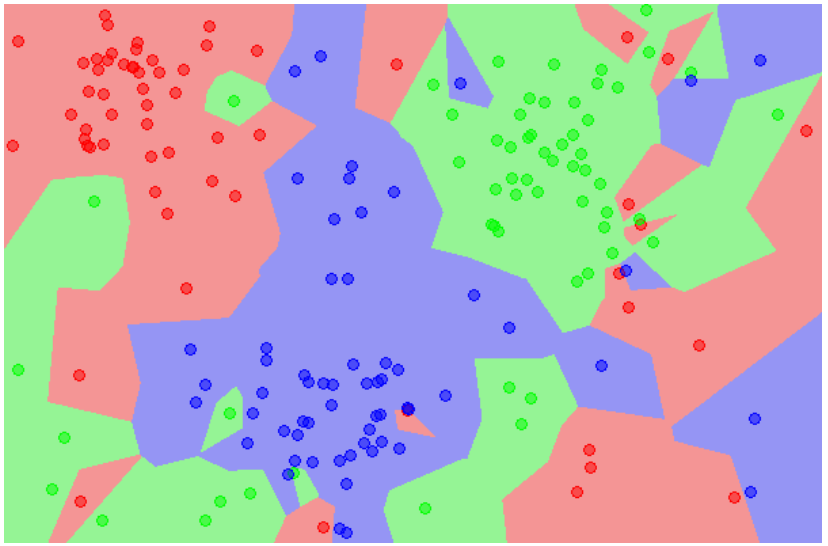
Плюсы:

- Скорость классификации новых объектов
- Незаменимость во многих приложениях (как следствие предыдущего пункта)

Минусы:

- Большинство методов имеют ограниченную “выразительность”, поскольку выбирают только одну гипотезу для всего признакового пространства
- Иногда приходится часто заново обучать модель при изменении данных (кредитный скоринг)

Плюсы методов без построения модели



Плюсы:

- Более богатое пространство гипотез, т.к. для каждого тестового объекта локальная аппроксимация целевой функции

- Для некоторых методов - улучшение качества классификации по сравнению с аналогом, производящим модель
- Для некоторых методов - хорошая интерпретируемость для каждого тестового объекта индивидуально
- Хорошее сочетание с экспертными знаниями в случае “рассуждения по прецедентам” (CBR)

Минусы методов без построения модели

Минусы:

- Низкая скорость классификации тестовых объектов
- Неэффективность применения во многих приложениях (как следствие)
- Тенденция к переобучению

Методы ленивой классификации и регрессии

Некоторые примеры:

- Метод ближайших соседей (k Nearest Neighbours, kNN)
- Наивный классификатор Бэйеса (Naive Bayes, NB)
- Локально взвешенная регрессия (Locally Weighted Regression, LWR)
- Ленивые деревья решений (Lazy Decision Trees, LDT)
- Ленивая классификация на основе бэйесовых правил (Lazy Bayesian Rules, LBR)
- Ленивая классификация на основе ассоциативных правил (Lazy Associative Classification, LAC)
- Рассуждение по прецедентам (Case-Based Reasoning CBR)

Ленивые деревья решений

Friedman, J.H, Kohavi, R., Yun, Y. "Lazy Decision Trees". AAAI, 1, 717-724, 1996

Algorithm 1 LazyDT

Вход: X – обучающая выборка, t – объект из тестовой выборки

Выход: y_t – предсказанная метка целевого класса для объекта t

1. Если все объекты в X имеют одну и ту же метку l, вернуть l
2. В противном случае выбрать признак A, пусть a — значение признака A у объекта t. Пусть X' – подмножество обучающих объектов со значением признака A, равным a. Применить алгоритм для X'

Для каждого тестового объекта строится свой "путь" дерева решений. На каждом шаге алгоритма выбирается разбиение среди признаков тестового объекта, приводящее к максимальному уменьшению энтропии целевого класса.
Преимущества по сравнению с "обычными" деревьями решений (ID3, C4.5, CART и т.д.):

- Правила намного короче и лучше трактуются
- Меньшая фрагментация данных (small disjuncts)
- Не так остра проблема пропусков в данных

Ленивая
классификация на основе бэйесовых правил

Zheng, Z., Webb, G.I. “Lazy Learning of Bayesian Rules”. Machine Learning, 41, 53–84, 2000

Идея похожа на алгоритм LazyDT. Для каждого тестового объекта применяется наивный бэйесов классификатор с признаками только этого тестового объекта.

Классификация
на основе ассоциативных правил

Существует множество алгоритмов классификации, основанных на поиске классифицирующих ассоциативных правилах. В одном из них ищутся все такие правила модификацией алгоритма Apriori.

Algorithm 2 Eager Associative Classifier

Вход: X_{train} – обучающая выборка, X_{test} – тестовая выборка

Выход: y_t – вектор предсказанных меток целевого класса для объекта тестовой выборки

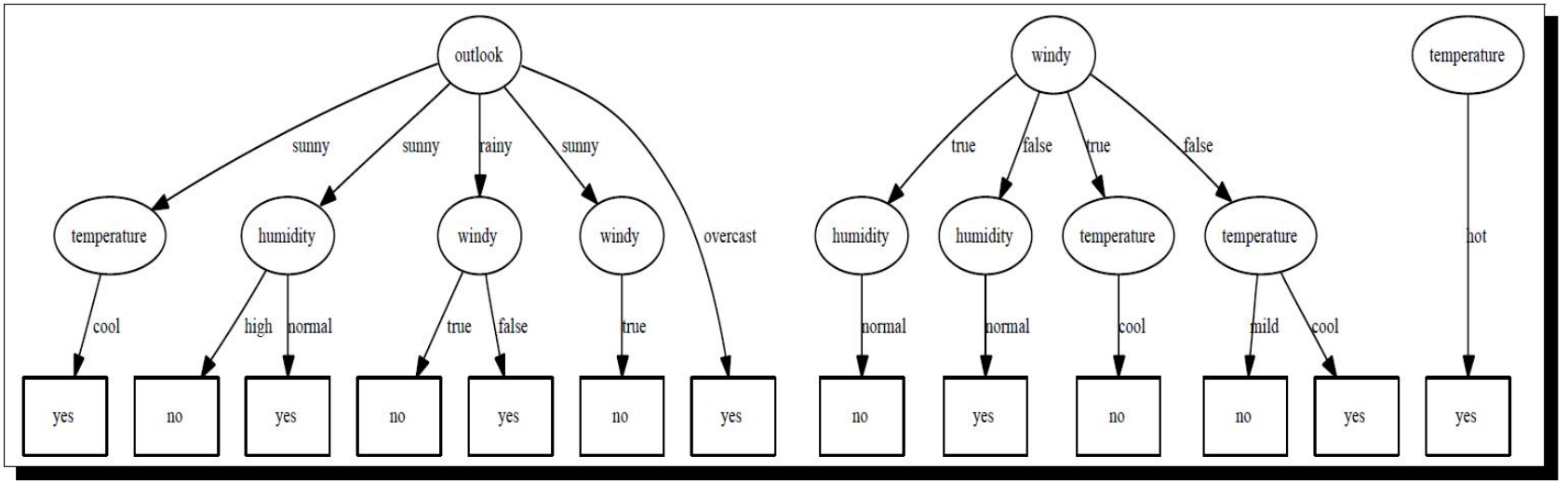
1. Найти в X_{train} множество C ассоциативных правил вида $\{\chi \to y_i\}$, где χ – подмножество признаков из объектов X_{train}, y_i – метка целевого класса
2. Отсортировать правила C по приросту информации
3. Для каждого $t_i \in X_{test}$
4. Определить метку y_{t_i} как посылку "первого подходящего" правила $\{\chi_i \to y_i\} \in C$, где χ_i – подмножество признаков t_i
5. Добавить метку y_{t_i} в вектор y_t

Набор данных для иллюстрации

Набор данных для иллюстрации

Play	Outlook	Temperature	Humidity	Windy
yes	rainy	cool	normal	false
no	rainy	cool	normal	true
yes	overcast	hot	high	false
no	sunny	mild	high	false
yes	rainy	cool	normal	false
yes	sunny	cool	normal	false
yes	rainy	cool	normal	false
yes	sunny	hot	normal	false
yes	overcast	mild	high	true
no	sunny	mild	high	true
?(yes)	sunny	cool	high	false

Пример классификации на основе ассоциативных правил



Для каждого тестового объекта приходится искать правила среди всех порожденных.

Ленивая классификация на основе ассоциативных правил

Veloso, A., Meira, W. Jr, Zaki, M., J. “Lazy Associative Classification”. ICDM, 645-654, 2006

Algorithm 3 Lazy Associative Classifier

Вход: X_{train} – обучающая выборка, X_{test} – тестовая выборка

Выход: y_t – вектор предсказанных меток целевого класса для объекта тестовой выборки

Для каждого $t_i \in X_{test}$

1. Пусть X^i_{train} – проекция обучающей выборки X_{train} на признаки объекта t_i
2. Найти в X^i_{train} множество C_{t_i} ассоциативных правил вида $\{\chi \to y_i\}$, где χ – подмножество признаков объекта t_i, y_i – метка целевого класса
3. Отсортировать правила C_{t_i} по приросту информации
4. Определить метку y_{t_i} как посылку правила из C_{t_i} с максимальным приростом информации
5. Добавить метку y_{t_i} в вектор y_t

Пример ленивой классификации на основе ассоциативных правил

Проекция обучающей выборки на признаки тестового объекта:

Play	Outlook	Temperature	Humidity	Windy
no	–	–	–	true
yes	overcast	hot	–	–
yes	–	hot	–	–
yes	overcast	–	–	true
no	–	–	–	true
?(yes)	overcast	hot	low	true

Найденные правила:

{windy=false and humidity=normal→play=yes}

{windy=false and temperature=cool→play=yes}

Преимущества
LAC по сравнению с EAC и деревьями решений

Veloso, A., Meira, W. Jr, Zaki, M., J. “Lazy Associative Classification”. ICDM, 645-654, 2006

Преимущества ленивой классификации на основе ассоциативных правил (LAC) по сравнению с классификацией с помощью всех классифицирующих ассоциативных правил (EAC) и деревьями решений:

- Порождается меньше правил
- Порождаются правила, “необходимые для классификации”, то есть классифицирующие конткретный тестовый объект
- Классифицирующие правила короче
- Не так остра проблема пропусков в данных
- Лучше качество классификации на многих наборах данных UCI

Частые замкнутые множества признаков

В интеллектуальном анализе данных активно развивается поиск частых замкнутых множеств признаков (frequent closed itemset mining).
Рассмотрим их смысл (на уровне интуиции) на игрушечном примере анализа потребительской корзины.

TID	Items
1	Bread, Milk
2	Bread, Diaper, Beer, Eggs
3	Milk, Diaper, Beer, Coke
4	Bread, Milk, Diaper, Beer
5	Bread, Milk, Diaper, Coke

	Beer	Bread	Milk	Diaper	Eggs	Coke
T_1	0	1	1	0	0	0
T_2	1	1	0	1	1	0
T_3	1	0	1	1	0	1
T_4	1	1	1	1	0	0
T_5	0	1	1	1	0	1

Заметим, что множество $\{Beer,\ Bread,\ Diaper\}$ – частое при ограничении на поддержку в 0.4. т.к. такое сочетание товаров наблюдается в 2 транзакциях из 5.

Частые замкнутые множества признаков II

TID	Items
1	Bread, Milk
2	Bread, Diaper, Beer, Eggs
3	Milk, Diaper, Beer, Coke
4	Bread, Milk, Diaper, Beer
5	Bread, Milk, Diaper, Coke

	Beer	Bread	Milk	Diaper	Eggs	Coke
T_1	0	1	1	0	0	0
T_2	1	1	0	1	1	0
T_3	1	0	1	1	0	1
T_4	1	1	1	1	0	0
T_5	0	1	1	1	0	1

Множество признаков называется *замкнутым*, если никакое из его непосредственных надмножеств не имеет ту же поддержку.
Множество $\{Beer,\ Bread,\ Diaper\}$ замкнуто – нельзя его расширить (добавить товары), не уменьшив при этом поддержку.
Частые замкнутые множества признаков полезно искать потому, что любое подмножество частого замкнутого множества тоже будет частым.

Анализ Формальных Понятий

Анализ Формальных Понятий - прикладная ветвь алгебраической теории решеток. Приведем только основные определения.

	G \ M	a	b	c	d
1	△	×			×
2	◺	×		×	
3	▭		×	×	
4	□		×	×	×

a - ровно 3 вершины
b - ровно 4 вершины
c - имеет прямой угол
d - все стороны равны

Формальный контекст $\mathbb{K}$ есть тройка (G, M, I), где G – множество, называемое множеством объектов, M – множество, называемое множеством признаков, $I \subseteq G \times M$ – отношение инцидентности, где отношение I интерпретируется следующим образом: для $g \in G, m \in M$ имеет место gIm, если объект g обладает признаком m.

Анализ Формальных Понятий II

Для формального контекста $\mathbb{K} = (G, M, I)$ и произвольных $A \subseteq G$ и $B \subseteq M$ определена пара отображений:

$A' \stackrel{\text{def}}{=} \{m \in M \mid gIm \text{ for all } g \in A\}$,

$B' \stackrel{\text{def}}{=} \{g \in G \mid gIm \text{ for all } m \in B\}$.

Множество A называется *замкнутым* если $A'' = A$.

Формальное понятие формального контекста $\mathbb{K} = (G, M, I)$ есть пара (A, B), где $A \subseteq G, B \subseteq M, A' = B$ и $B' = A$.

Множество A называется объёмом, а B – содержанием понятия (A, B).

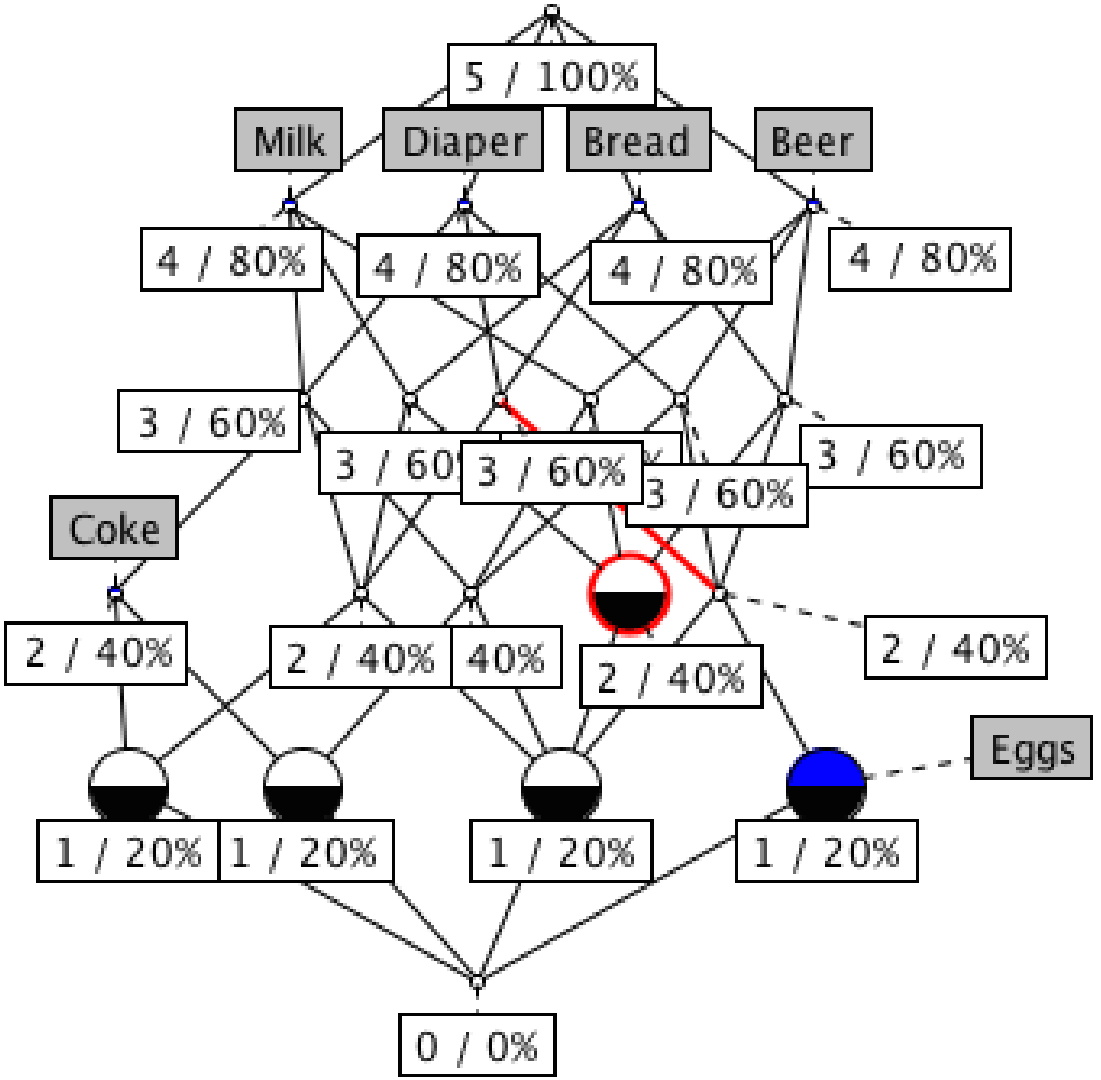
АФП - пример замкнутого множества

TID	Items
1	Bread, Milk
2	Bread, Diaper, Beer, Eggs
3	Milk, Diaper, Beer, Coke
4	Bread, Milk, Diaper, Beer
5	Bread, Milk, Diaper, Coke

	Beer	Bread	Milk	Diaper	Eggs	Coke
T_1	0	1	1	0	0	0
T_2	1	1	0	1	1	0
T_3	1	0	1	1	0	1
T_4	1	1	1	1	0	0
T_5	0	1	1	1	0	1

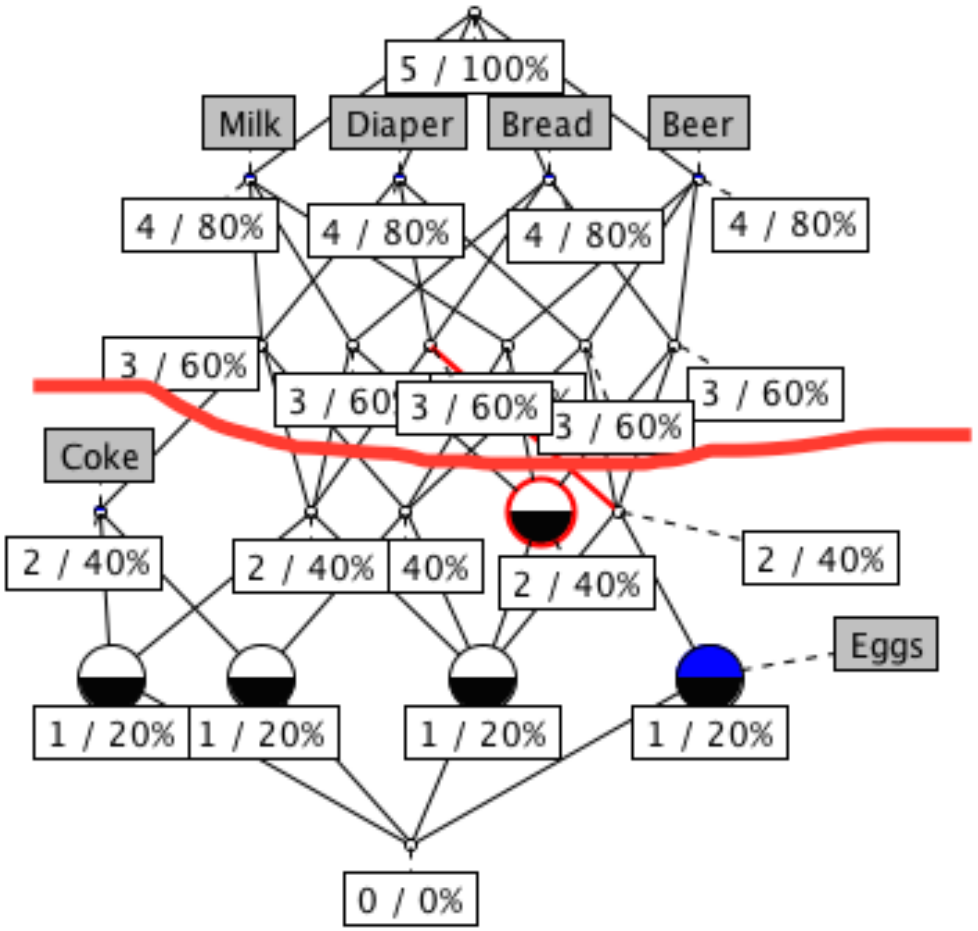
В данном примере
$G = \{T_1, T_2, T_3, T_4, T_5\}$, $M = \{Beer, Bread, Milk, Eggs, Coke\}$.
Множество $\{Beer, Bread, Diaper\}$ действительно замкнуто:
$\{Beer, Bread, Diaper\}' = \{T_2, T_4\}$
$\{Beer, Bread, Diaper\}'' = \{T_2, T_4\}' = \{Beer, Bread, Diaper\}$

Решетка формальных понятий



Частые замкнутые множества признаков в решетке формальных понятий

minsupp = 60%



Предлагаемый алгоритм

Для каждого тестового объекта:

- Оставить в выборке только признаки этого объекта
- Построить решетку формальных понятий для полученного контекста
- Оставить только “верх” решетки, ограничив мощность замкнутых множеств признаков параметром k
- При построении решетки отслеживать для каждого узла соотношение классов целевого признака и информационный критерий, такой как Gini, GiniRatio или InformationGain.

Предлагаемый алгоритм II

- Отслеживать топ-N правил вида $\{A_i \to c_i\}, i = 1 \ldots N$ с наибольшими значениями информационного критерия. Здесь A_i - замкнутые множества признаков (= содержания формальных понятий), c_i - преобладающая метка целевого признака среди объектов A_i'.
- Вернуть предсказанную метку для данного тестового объекта простым голосованием среди правил $\{A_i \to c_i\}$

Бинаризованные признаки

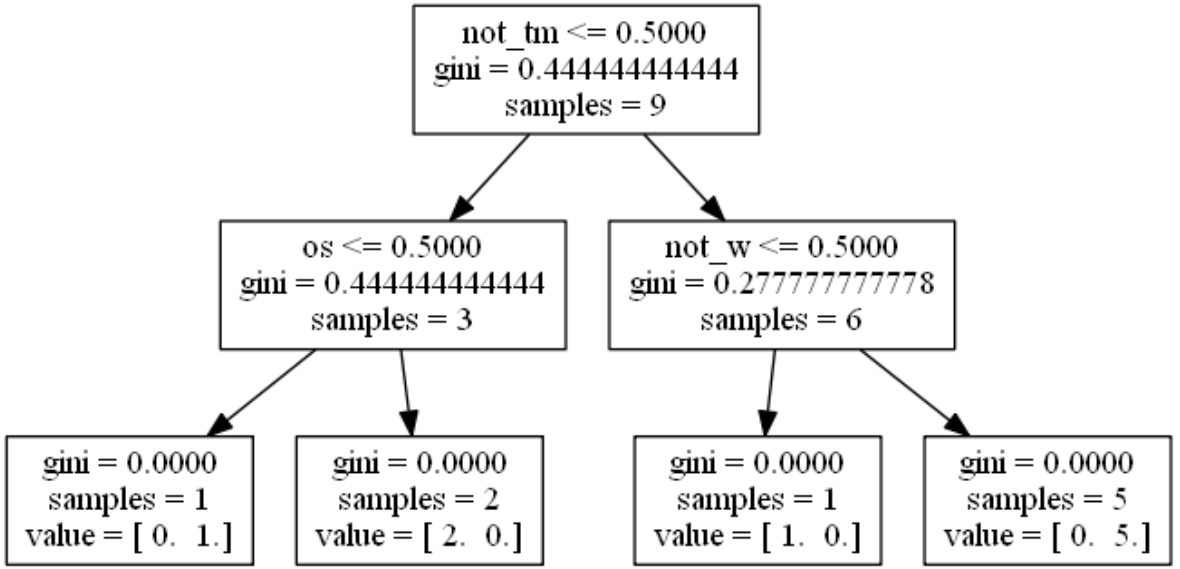
Предыдущий пример с бинарными признаками:

	or	oo	os	tc	tm	th	hn	w	play
1	1	0	0	1	0	0	1	0	1
2	1	0	0	1	0	0	1	1	0
3	0	1	0	0	0	1	0	0	1
4	0	0	1	0	1	0	0	0	0
5	1	0	0	1	0	0	1	0	1
6	0	0	1	1	0	0	1	0	1
7	0	0	1	0	0	1	1	0	1
8	0	1	0	0	1	0	0	1	1
9	0	0	1	0	1	0	0	1	0

or = [Outlook=rainy]
oo = [Outlook=overcast]
os = [Outlook=sunny]
tc = [Temperature=cool]
tm = [Temperature=mild]
th = [Temperature=high]
hn = [Humidity=normal]
w = [Windy]

Дерево решений

Дерево решений для данного примера



sklearn.tree.DecisionTreeClassifier(max_depth=2, random_state=42)

Тестовый объект с признаками $\{Outlook = sunny, Temperature = cool, Humidity = high, Windy = false\}$ классифицируется таким деревом как ситуация, когда играть можно.

$GiniGain([Temperature\ != mild, Windy = false]) = 0.28$

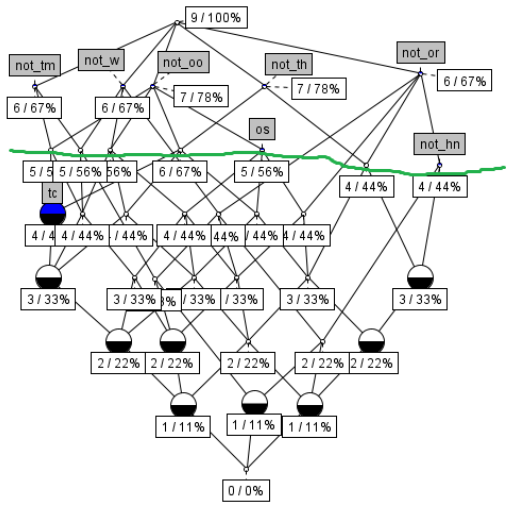
Обучающая выборка с признаками тестового объекта

Обучающая выборка в “проекции” на признаки тестового объекта (добавлены отрицания исходных признаков)

	os	tc	not_or	not_oo	not_tm	not_th	not_hn	not_w	play
1	0	1	0	1	1	1	0	1	1
2	0	1	0	1	1	1	0	0	0
3	0	0	1	0	1	0	1	1	1
4	1	0	1	1	0	1	1	1	0
5	0	1	0	1	1	1	0	1	1
6	1	1	1	1	1	1	0	1	1
7	1	0	1	1	1	0	0	1	1
8	0	0	1	0	0	1	1	0	1
9	1	0	1	1	0	1	1	0	0

Решетка и рейтинг правил

Решетка формальных понятий для формального контекста предудущей выборки и “рейтинг” полученных правил



	GiniGain
['not_tm', 'not_w']	0.277744
['not_oo', 'not_th']	0.111144
['os', 'not_oo']	0.044444
['not_oo', 'not_tm']	0.044444
['not_hn', 'not_or']	0.044444
['not_oo', 'not_w']	0.044444
['not_or', 'not_th']	0.044444

Сверху от границы - формальные понятия с содержаниями из двух и менее признаков.

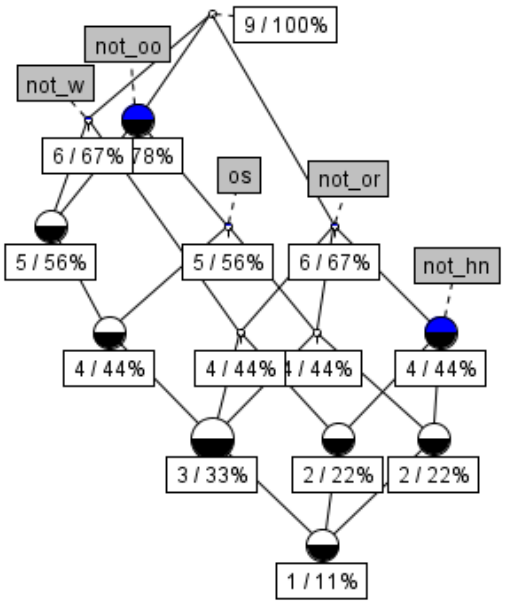
Случай пропуска в данных

Предположим, признак *Temperature* для тестового объекта был неизвестен. Тогда берем проекцию обучающей выборки на те признаки, значения которых известны.

	os	not_or	not_oo	not_w	not_hn
1	0	0	1	1	0
2	0	0	1	0	0
3	0	1	0	1	1
4	1	1	1	1	1
5	0	0	1	1	0
6	1	1	1	1	0
7	1	1	1	1	0
8	0	1	0	0	1
9	1	1	1	0	1

Решетка и рейтинг правил

Решетка формальных понятий для формального контекста предыдущей выборки и “рейтинг” полученных правил



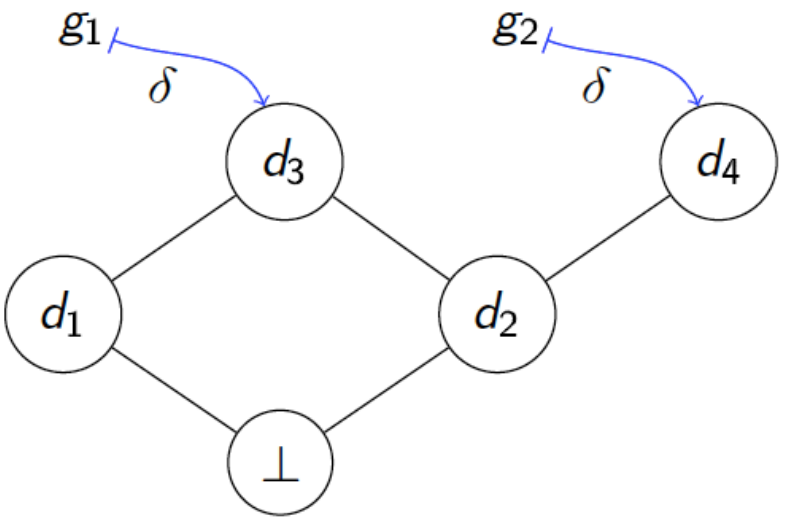
	GiniGain
['os', 'not_oo']	0.044444
['not_oo', 'not_w']	0.044444
['not_or', 'not_hn']	0.044444
['not_or', 'not_w']	0.011144

Узорные структуры

Ganter, B., Kuznetsov, S.O., “Pattern Structures and Their Projections”. ICCS 2001, LNAI, 129-142, 2001

Узорная структура: $(G, (D, \sqcap), \delta)$ [Ganter and Kuznetsov, 2001]

- $G = \{g_1, g_2, \ldots\}$ – множество объектов
- $(D, \sqcap)$ – полурешётка описаний объектов, т.е. D – некоторое множество, а $\sqcap : D \times D \to D$ – коммутативная, ассоциативная и идемпотентная операция на описаниях из D
- Функция описания объектов δ: $G \to D$

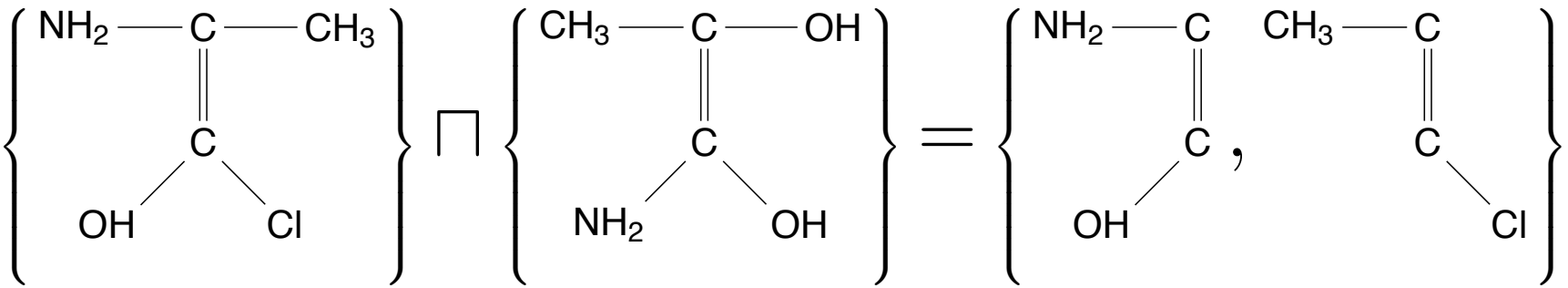


Порядок на помеченных графах

Помеченный граф $\Gamma_1 := ((V_1, lv_1), (E_1, le_1))$ **доминирует** над графом $\Gamma_2 := ((V_2, lv_2), (E_2, le_2))$ или $\Gamma_2 \leq \Gamma_1$
если существует взаимнооднозначное отображение $\varphi\colon V_2 \to V_1$, которое

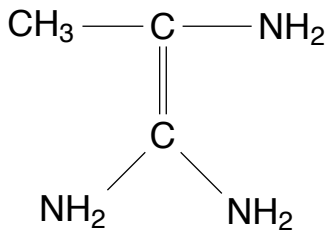
- учитывает ребра: $(v, w) \in E_2 \Rightarrow (\varphi(v), \varphi(w)) \in E_1$,
- учитывает порядок на метках: $lv_2(v) \leq lv_1(\varphi(v))$, $le_2(v, w) \leq le_1(\varphi(v), \varphi(w))$.

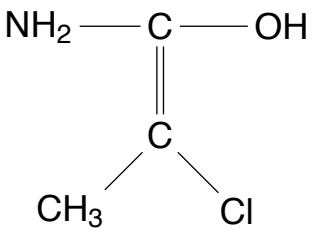
Пример:

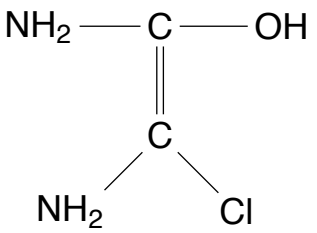


Пример для графов

Замкнутые множества графов могут быть представлены узорной структурой. $\{1, 2, 3\}$ - объекты, $\{G_1, G_2, G_3\}$ – их молекулярные графы:

G_1 : 

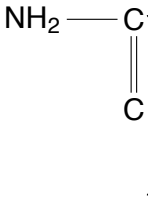
G_2 : 

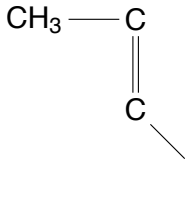
G_3 : 

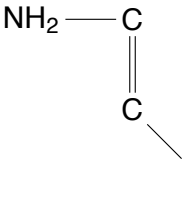
Объекты $\{1, 2, 3\}$, графы $D = \{G_1, G_2, G_3\}$ ($\delta(i) = G_i, i = 1, \ldots, 3$) и оператор пересечения $\sqcap$ образуют узорную структуру $(\{1, 2, 3\}, (D, \sqcap), \delta)$.

Пример для графов II

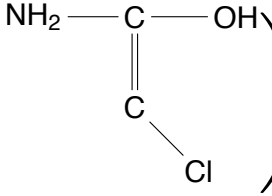
Можно найти все *узорные понятия* для такой узорной структуры:

$\{$ $\left(\{1,2,3\}\ ,\ \right.$ 

$\left.\right)$, $\left(\{1,2\}\ ,\ \right.$ 

$\left.\right)$, $\left(\{1,3\}\ ,\ \right.$ 

$\left.\right)$,

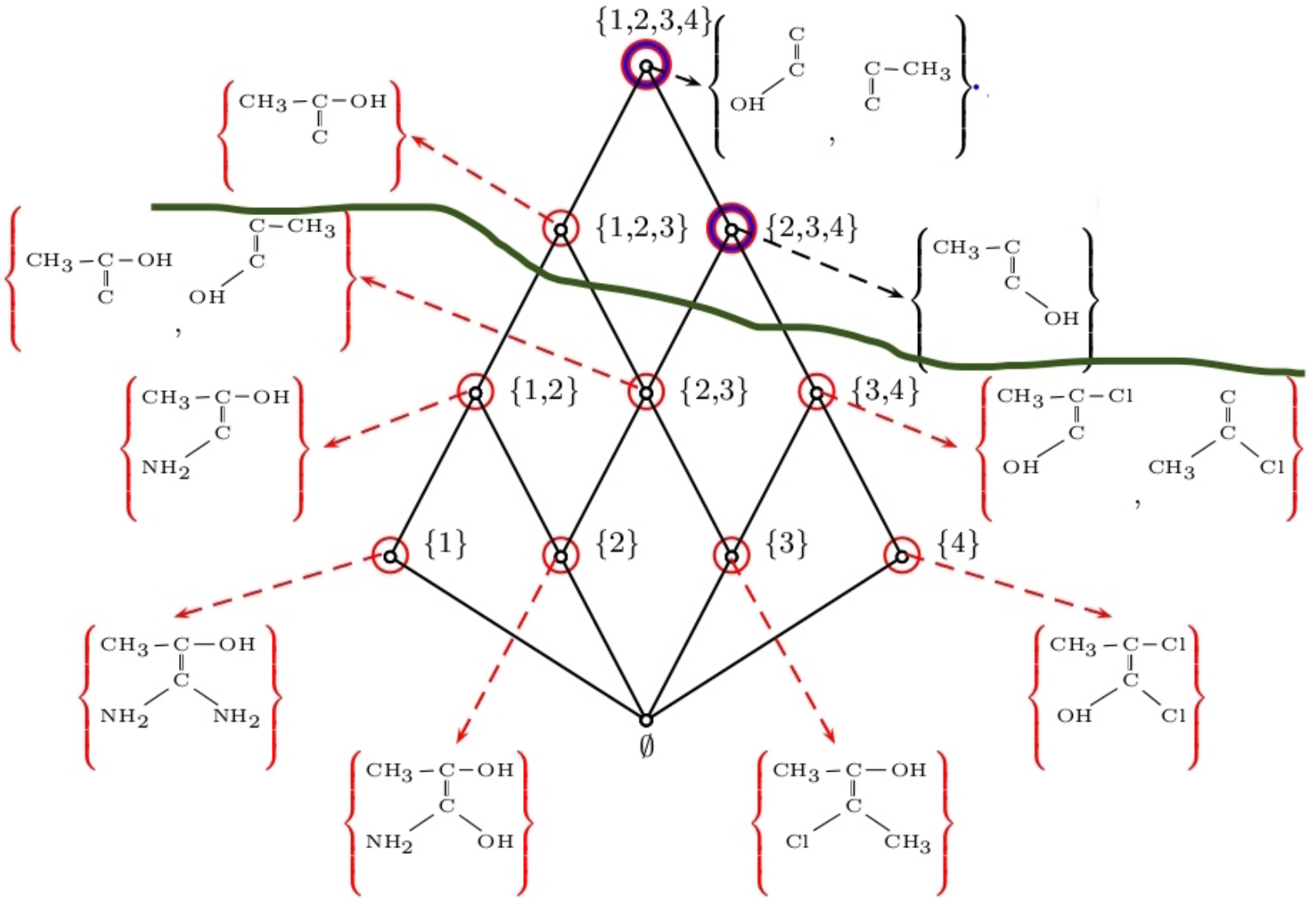
$\left(\{2,3\}\ ,\ \right.$ 

$\left.\right)$, $(1,\{G_1\})$, $(2,\{G_2\})$,

$(3,\{G_3\})$, $(\emptyset,\{G_1,G_2,G_3\})$ $\}$.

Их содержания - это замкнутые множества графов.

Частые замкнутые множества графов



Предлагаемый алгоритм классификации данных, представленных графами (в разработке)

Для каждого тестового объекта:

- Ограничить множество графов из обучающей выборки (TODO)
- Построить решетку узорных понятий для полученного контекста
- Ограничить число понятий по мощности объема (числа объектов)
- При построении решетки отслеживать для каждого узла соотношение классов целевого признака и информационный критерий, такой как Gini, GiniRatio или InformationGain.

Предлагаемый алгоритм II

- Отслеживать топ-N правил вида $\{A_i \to c_i\}, i = 1 \ldots N$ с наибольшими значениями информационного критерия. Здесь A_i - замкнутые множества графов (= содержания узорных понятий), c_i - преобладающая метка целевого признака среди объектов A'_i.
- Вернуть предсказанную метку для данного тестового объекта простым голосованием среди правил $\{A_i \to c_i\}$

Спасибо за внимание!

Ленивые методы ассоциативной классификации в машинном обучении

Юрий Кашницкий

ykashnitsky@hse.ru

http://hse.ru/staff/ykashnitsky/

Национальный исследовательский университет
«Высшая школа экономики» (Москва)

28 января 2016