МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ
ВЫСШЕГО ОБРАЗОВАНИЯ
"МОСКОВСКИЙ ФИЗИКО-ТЕХНИЧЕСКИЙ ИНСТИТУТ
(ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ)"

ФАКУЛЬТЕТ ИННОВАЦИЙ И ВЫСОКИХ ТЕХНОЛОГИЙ
КАФЕДРА АНАЛИЗА ДАННЫХ

Выпускная квалификационная работа по направлению 01.03.02
«Прикладные математика и информатика»
НА ТЕМУ:

ПОСТРОЕНИЕ РЕЙТИНГА ОБЪЕКТОВ НА ОСНОВЕ МЕДИАНЫ КЕМЕНИ И ПАРЕТО-ФРОНТОВ

Студент ______________________________ Ивановский Д.А.

Научный руководитель к.ф-м.н ______________________ Ивановский Д.А.

Зам. зав. кафедрой д.ф-м.н, профессор ________________ Ивановский Д.А.

МОСКВА, 2016

Содержание

Аннотация

Рассматривается задача построения рейтинга по заданной матрице экспертных оценок (объект-признак). В качестве методов решения предложенной задачи рассматриваются медиана Кемени и поиск Парето-оптимальных фронтов. Далее приводится новый алгоритм, являющийся комбинацией предыдущих двух, позволяющий при некоторых значениях параметров количество экспертов, количество признаков и согласованность экспертов, добиться качества близкого к медиане Кемени за меньшее время. Результат работы алгоритмов проиллюстрирован примерами на разных по согласованности экспертов матриц с данными, проведены замеры времени работы и качества.

Ключевые слова: *Парето-оптимальный фронт, рейтинг, медиана Кемени.*

Введение

Работа посвящена построению рейтинга объектов по матрице "объект-признак"$\mathbf{Q}$ оценок нескольких экспертов. Каждый из экспертов выставляет оценку объектам, где оценка для каждого объекта может принимать значения в какой-либо линейной шкале.

Область построения рейтингов объектов по оценочным суждениям экспертов является достаточно старой. Существует множество различных алгоритмов поиска результирующего рейтинга и не меньшее количество критериев, которым эти алгоритмы могут удовлетворять. Одним из самых лучших алгоритмов, удовлетворящий большому числу критериев и имеющий естественную интерпретацию, является алгоритм медианы Кемени [1].

Задачу вычисления медианы Кемени часто рассматривают в контексте голосования за фиксированный набор кандидатов. В процессе голосования определяется набор рейтингов по каждому критерию, сама медиана Кемени является результирующим рейтингом объектов. В нашем случае каждый критерий становится голосующим, и порядок в котором он проголосовал определяется значениями критерия на объектах.

В качестве стандартного алгоритма предлагается использовать алгоритм перебора. Задача поиска медианы Кемени [2] — это задача нахождения минимума функционала особого вида, позволяющего согласовывать экспертные оценки. Вычисление медианы Кемени является NP-полной задача [3]. Существует ряд алгоритмов, которые позволяют найти приближения к ответу, используя вероятностный подход [4].

Далее предлагается идея, позволяющая разбить множество объектов на несколько групп, между которыми уже определено отношение порядка и далее вычислять медиану Кемени только внутри групп. Разбиение на группы основано на расслоении по Парето-оптимальному фронту [5]. По матрице $\mathbf{Q}$ проводится поиск оптимальных Парето-фронтов по выбранным критериям.

Уже на этапе описания алгоритмов можно заметить, что алгоритм поиска Парето-оптимальных фронтов - это быстрый алгоритм, но при большом количестве признаков, все объекты попадут на один Парето фронт и алгоритм будет бесполезен.

Целью работы является привести пример комбинации алгоритмов поиска Парето-оптимальных фронтов и медианы Кемени и найти комбинации таких параметров, как количество экспертов, количество признаков, согласованность экспертов, при которых этот алгоритм работает по качеству близкому к медиане Кемени и при этом асимтотически быстрее.

Формальная постановка задачи

Задана матрица экспертных оценок

$$\mathbf{Q} = \{q_{ij}\} = \{\mathbf{X}_1, \ldots, \mathbf{X}_m\},$$

где q_{ij} — оценка i-го объекта j-го признака, $i \in \{1, \ldots, n\}$, $j \in \{1, \ldots, m\}$. $\mathbf{X}_k$ — вектор-столбец, значения объектов при фиксированном k-ом признаке. Элементы матрицы $\mathbf{Q}$ принимают значения из линейного пространства $\mathbb{L}$.

Каждый признак задает ранжирующую функцию $g_j : \{1, \ldots, n\} \to \mathbb{L}, g_j(i) = x_{ij}$. На g_j можно смотреть, как на конечную последовательность элементов из $\mathbb{L}$, $g_j = \{g_j(1), \ldots, g_j(n)\}$.

Каждой ранжирующей функции g_j соответствует функция рейтинга $r_j : \{1, \ldots, n\} \to [1, \ldots, n]$, $r_j(i)$ - рейтинг компании с номером i на признаке j.

Функция рейтинга r_j строится в 2 этапа. Сначала значения функции $g_j(i)$ сортируются по убыванию, значение $g_j(i)$ заменяется на номер компании i в отсортированном списке. Далее производится усреднение групп номеров с одинаковыми значениями функции, элементы внутри каждой группы заменяются на их среднее арифмитическое. Количество групп обозначим за $size_j$, а количество элементов в группе t_{jk}, где $k \in \{1, \ldots, size_j\}$

Рассмотрим пример нахождения рейтинга по ранжирующей функции:

$$m = 6, g_j = \{50, 80, 30, 50, 80, 80\}$$

g_j после сортировки значений преобразуется в $\{4, 1, 6, 5, 2, 3\}$ Выделяются 3 группы с одинаковыми значениями g_j: $\{1, 2, 3\}, \{4, 5\}, \{6\}$.

Заменяем номера внтури группы на среднее арифмитическое номеров группы.

$$r_j = \{4.5, 2, 6, 4.5, 2, 2\}, size_j = 3, t_{j1} = 3, t_{j2} = 2, t_{j3} = 1$$

На r_j можно смотреть, как на конечную последовательность элементов из $[1, \ldots, n]$, $r_j = \{r_j(1), \ldots, r_j(n)\}$, при этом, если $size_j = n$, то r это в точности перестановка множества $\{1, \ldots, n\}$.

Определим на множестве рейтингов метрику Кемени:

$$d_{Kem}(r_s, r_t) = \frac{1}{2} \sum_{i=1}^{m} \sum_{u=1}^{m} |z_{(s)iu} - z_{(t)iu}|,$$

где

$$z_{(s)iu} = \begin{cases} 1, \text{ если } r_s(i) > r_s(u), \\ 0, \text{ если } r_s(i) = r_s(u), \\ -1, \text{ иначе.} \end{cases}$$

Пусть $\mathbf{\Pi}$ - множество перестановок $\{1, \ldots, m\}$. Тогда медианой Кемени назовем рейтинг Kem

$$Kem(r_1, ..., r_m) = \arg\min_{r \in \mathbf{\Pi}} \sum_{j=1}^{n} d_{Kem}(r_j, r). \tag{1}$$

Функцию ошибки $Error$ для результирующего рейтинга r мы определим как отношение суммы расстояний рейтингов $r_1, ..., r_m$ от рейтинга r в метрике Кемени $d_{Kem}(r_s, r_t))$ к сумме расстойний рейтингов $r_1, ..., r_m$ от рейтинга Kem

$$Error(r, r_1, ..., r_m) = \frac{\sum_{i=1}^{m} d_{Kem}(r, r_i)}{Kem(r_1, ..., r_m)}$$

Следствия определения

$$\forall r, r1, ..., r_m Error(r, r_1, ..., r_m) >= 1 \forall r1, ..., r_m Err(Kem(r_1, ..., r_m), r_1, ..., r_m) = 1$$

Согласованность экспертных оценок $onsent$ мы определим как среднее арифметическое попарных расстояний рейтингов $r_1, ..., r_m$ в метрике Кемени $d_{Kem}(r_s, r_t))$

$$onsent(r_1, ..., r_m) = \frac{\sum_{s=1}^{m} \sum_{t=1, s!=t}^{m} d_{Kem}(r_s, r_t)}{n(n-1)},$$

Следствие определения

Если $r_1, ..., r_m$ рейтинги на n объектах, то

$$0 <= onsent(r1, ..., r_m) <= n(n-1)$$

$n(n-1)$ является диаметром на метрическом пространстве рейтингов, соответствует расстоянию между рейтингами, один из которых сортировка по возрастанию, а другой по убыванию по одному из признаков

Медиана Кемени в качестве результирующего рейтинга

Согласно теореме, доказанной в [1] медиана Кемени - это единственный рейтинг, который является нейтральным, последовательным и обладает свойством Condorset. Медиана Кемени учитывает всё множество признаков при минимизации функционала, и всегда находит рейтинг, в котором все объекты имеют разное значение рейтинга. Однако алгоритм вычисления медианы Кемени имеет экспоненциальную от количества объектов сложность по времени и лежит в классе NP-hard [3].

Алгоритм $MEDIANA\_KEMENI(r_1, ..., r_m, n)$:

1. Присвоить $min = MAX_V ALUE$
2. Выбрать перестановку r из множества всех перестановок $\mathbf{\Pi}$ на n элементах
3. $d_k em = d_{Kem}(r, r_1, ..., r_m)$
4. Если $d_k em < min$ присвоить $min = d_k em$ и $r_r es = r$
5. Повторять шаги 2-4 по всем перестановкам
6. Вернуть $r_r es$

Это простой переборный алгоритм, не использующий никаких предположений об аргументах и реализующий определение медианы Кемени. Принципиальным в этом алгоритме является то, что для поиска ответа необходимо проверить все перестановки размером n. Отсюда и экспоненциальная сложность. При этом в силу принадлежности классу NP-hard совершенно не ясно как в общем случае уменьшить множество перестановок, которые нужно перебрать или в каком порядке осуществить перебор, чтобы найти медиану за полиномиальное время.

Из известных алгоритмов существует оптимизация, основанная на идее целочисленного программирования [6], которая не дает асимтотического выйгрыша, но позволяет осуществить перебор гораздо быстрее. Также существует приближенная схема полиномиального времени [4].

Модификация базового алгоритма поиска медианы Кемени

Сложность поиска медианы Кемени заключается в количестве возможных рейтингов, которые необходимо просмотреть. Их количество равно мощности $\mathbf{\Pi}$, то есть $n!$.

Но если разбить множество объектов $\mathbf{X} = C_1 \bigsqcup \cdots \bigsqcup C_p$, на непересекающиеся подмножества, так что на этих множествах будет установлено отношения порядка

$$C_1 \prec \cdots \prec C_l$$

$$C_i \prec C_j \Leftrightarrow \forall x \in C_i, y \in C_j \ x \prec y$$

$x \prec y$ - означает объект x находится в рейтинге ниже, чем объект y

Алгоритм $FIND\_RATING(r_1, ..., r_m, n)$:
1. Разбить множество объектов на $C_1, ..., C_p$
2. Для каждого C_i из $C_1, ..., C_p$
3. $d_k em_i = MEDIANA\_KEMENI(r_1, ..., r_m, |C_i|)$
4. Повторять шаги 2-4
5. Вернуть объединенный рейтинг из $d_k em_1, ..., d_k em_p$

Количество рейтингов, которые необходимо проверить при поиске медианы Кемени уменьшается до $\sum\limits_{i=1}^{l} |C_i|!$, так как теперь нам нужно решить задачу поиска медианы Кемени независимо внутри каждого множества C_i.

Заметим, что алгоритм поиска медианы Кемени является частным случаем нового алгоритма, для случая $p = 1, |C_1| = n$. Также, если на строке 1, разбиение на множества работает за полиномиальное время, то алгоритм лежит в том же классе временной сложности, что и поиск медианы Кемени.

Воспользуемся фактом, что

$$\sum_{i=1}^{n} |C_i| = n$$

$$\sum_{i=1}^{l} |C_i|! <= n!$$

и введем функцию сложности как

$$Complexity(|C_1|, ..., |C_p|) = \frac{log_n \frac{\sum\limits_{i=1}^{l} |C_i|!}{\sum_{i=1}^{n} |C_i|!}}{n} = \frac{log_n \frac{\sum\limits_{i=1}^{l} |C_i|!}{n!}}{n}$$

Выбор этой формулы основан на том, что необходимо ввести оценку улучшения времени работы алгоритма по сравнению с поиском медианы Кемени, не зависящую от n

Пользуясь оценкой факториала из формулы Стирлинга

$$n! \sim \frac{(\frac{n}{e})^n}{\sqrt{2\pi n}}$$

можно доказать, что

$$0 <= Complexity(|C_1|, ..., |C_p|) <= 1$$

При этом близкие к 0 значения означают, что алгоритм работал в точности как Медиана Кемени, и разбиение на слои не дало существенного выйгрыша, подмножеств было немного и все они были порядка n. Значения близкие к 1 напротив означают разбиение на большое количество подмножеств порядка n, с константным размером каждого подмножества. Поведение этой формулы при небольших значениях n будет изучено в эксперименте.

Заметим, что результат нового алгоритма не всегда является медианой Кемени, то есть с точки зрения функции качества он не может быть лучше медианы Кемени, но при этом с точки зрения временной сложности, он работает не хуже.

Получается, что хорошее разбиение множества, может решить задачу поиска результирующего рейтинга существенно быстрее переборного алгоритма, но при этом хочется выбрать такое разбиение, которое не сильно ухудшит качество.

Начнем с простого рассуждения. Пусть есть объект, который лучше всех других во всех рейтинга. Тогда он будет первым и в медиане Кемени. Логично, что его следует отнести в отдельное подмножество, которое будет доминировать остальные. Симметричная ситуация с худшим объектом. Что если есть 2 лучших объекта, которые не сравнимы друг с другом, но лучше остальных. Аналогично рейтинг на этих объектах следует посчитать отдельно, и объединить с рейтингом на оставшихся объектах. Продолжив эти рассуждения, мы приходим к отношению Парето на множестве объектов и понятию Парето-фронты на наборе признаков.

Парето-оптимальные фронты

Определим понятие Парето-фронта и доминирования [7].

Строки матрицы $\mathbf{X}$ соответствуют объектам. Строка с номером p соответствует объекту $\mathbf{x_p} = [x_{p1}, \ldots, x_{pn}]$

Отношение доминирования

Введем на объектах $\{\mathbf{x}_p : p \in \{1, \ldots, m\}$ отношение доминирования $\succ$. Объект $\mathbf{x}_p = [x_{p1}, \ldots, x_{pn}]^\top$ доминирует объект $\mathbf{x}_i = [x_{i1}, \ldots, x_{im}]^\top$, если значения всех его признаков не менее предпочтительны, чем значения признаков $\mathbf{x}_i$:

$$\mathbf{x}_p \succ \mathbf{x}_i, \quad \text{если} \quad x_{pj} \succeq x_{ij} \quad \text{для всех} \quad j = 1, \ldots, n.$$

Будем считать, что объект не доминирует сам себя ни в одном из смыслов:

$$\mathbf{x_p} \not\succ \mathbf{x_p}$$

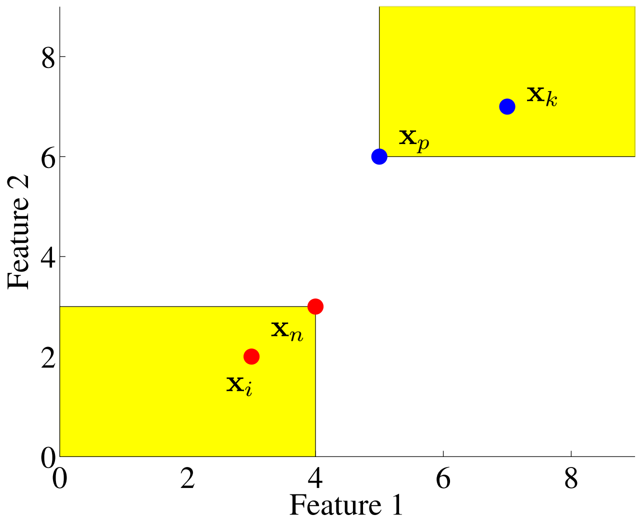


Рис. 1. Пример доминирования

На рис. 1 приведен пример доминирования для случая двух признаков. По осям отложены значения признаков из множества $\mathbb{L}$, желтым цветом показана область доминирования объекта $\mathbf{x}_n$. Объекты, попадающие в область, закрашенную желтым цветом, доминируются в соответствующем смысле рассмотренным объектом.

Построение Парето-оптимальных фронтов

Определим Парето-оптимальные фронты.

Определение 1. *Парето-оптимальный фронт POF — множество объектов $\mathbf{x}_p$, для каждого элемента которого $\mathbf{x}_p \in POF$ не существует ни одного объекта $\mathbf{x}$, такого, что $\mathbf{x} \succ \mathbf{x}_p$.*

На рис. 12 показаны примеры Парето-оптимальных фронтов, объекты которой описаны двумя признаками, принимающими значения из множества $\mathbb{L}$. Зелеными треугольниками обозначены объекты. Объекты, вошедшие во фронты, обозначены красными кружками. Граница класса, задаваемая фронтом, обозначена пунктирной линией. Объединения

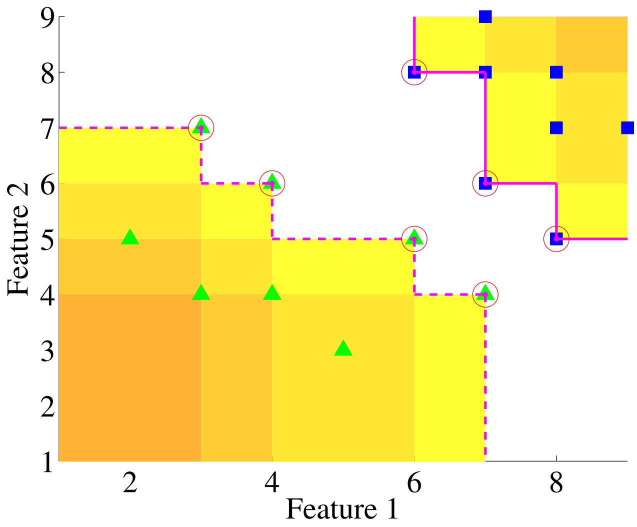


Рис. 2. Парето-оптимальные фронты

областей доминирования объектов из фронтов закрашены желтым цветом, чем темнее оттенок области, тем большее количество объектов ее доминирует. На рис. 12(a) изображены Парето-оптимальные фронты, соответствующие отношению доминирования.

Алгоритм поиска парето-оптимальных фронтов

Приведем псевдокод построения поиска парето-оптимальных фронтов для некоторого подмножества признаков $\chi \subseteq \{1, \dots, n\}$

Алгоритм FIND_POF:

1. Присвоить $i := 0$, $\mathbf{X} := \mathbf{x_1}, \dots, \mathbf{x_m}$
2. Повторять (3-7) пока $\mathbf{X}$ непусто
3. Найти POF на множестве X по приназнакам из χ.
4. $i := i + 1$
5. $\mathbf{X} = \mathbf{X} \setminus POF$
6. $C_i := POF$
7. $p := i$

Алгоритм находит все оценки объектов, которые не доминируются никакими другими оценками отношением Парето, включает их в первый фронт и присваивает всем этим объектам ретийнг с номером один. Далее удаляет эти объекты и повторяет процедуру.

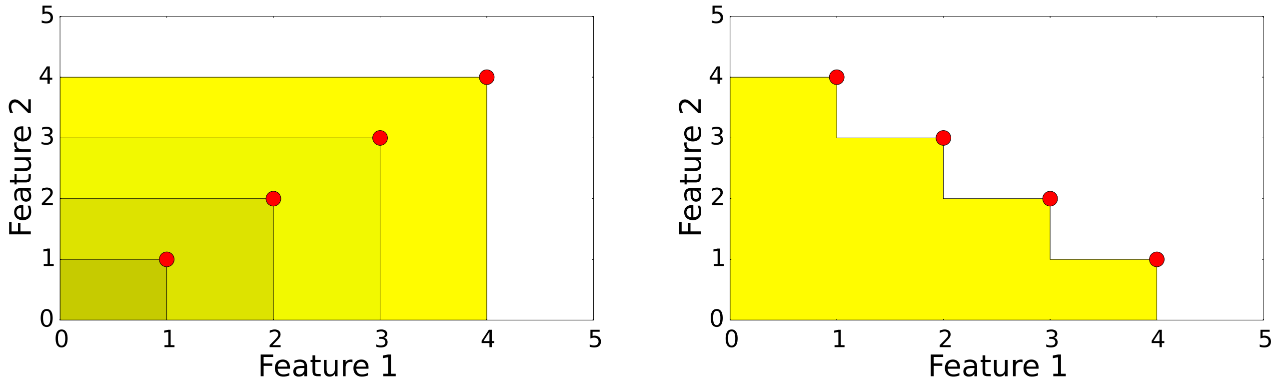
В результате алгоритм находит Парето фронты C_i и количество фронтов p.

Основной алгоритм

Рассмотрим в качестве разбиения для алгоритма FIND_RATING алгоритм FIND_POF

Цель работы определить при каких исходных рейтингах алгоритм FIND_RATING будет работать лучше, а именно быстрее и не сильно хуже. Используя рассуждения выше, о том что скорость алгоритма зависит от размера слоев, но от чего же зависит размеры слоев?

Рассмотрим простой пример. 4 Объекта, 2 эксперта. Значения шкалы от 0 до 5. В первом случае результаты экспертов совпали на всех объектах и колчиество слоев равно n. Отличный результат, алгоритм будет работать полиномиальное время с качеством медианы Кемени, так как не нужно будет вычислять медиану Кемени, из-за единичных размеров слоёв. Во втором случае результаты экспертов противоположны, объекты у второго эксперта упорядочены в другом порядке. Все объекты попали на один слой. Алгоритм работает аналогично медиане Кемени по качеству и времени.



(a) Пример 1, хорошего разделения

(b) Пример 2, плохого разделения

Рис. 3. Парето-оптимальные фронты

	Consent	*Complexity*
Пример 1	0	0.32
Пример 2	12	0

В обоих случаях алгоритм работает с одинаковой нулевой ошибкой, используя для этого дополнительные полиномиальные по времени вычисления. При этом мы рассмотели граничные значения функции согласованности. Поэтому интересным представляется поведение функции ошибки и сложности при промежуточных значениях согласованности. Конечно основной гипотезой работы является то, что когда эксперты согласованны алгоритм работает хорошо, а когда не солгасованны не сильно плохо. В данном контексте качество определяется только функцией ошибки, так как мы уже доказывали, что новый алгоритм не медленнее старого.

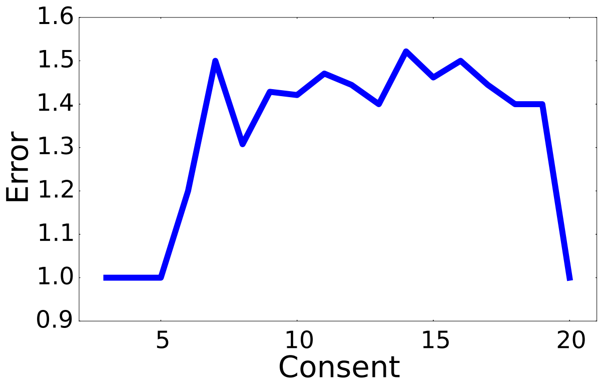
Вычислительный эксперимент

Эксперимент на случайных данных

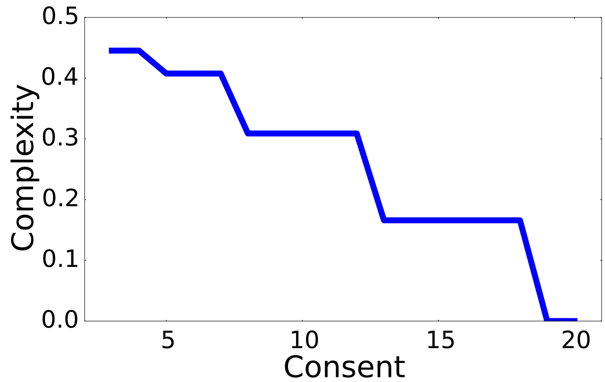
Определим 3 параметра. Количество объектов n, количество признаков m, и размер шкалы k. Шкала будет представлять собой все целые числа от 0 до k.

Будем генерировать таблицу размером nm, каждое значение которой будет из дискретного равномерного распределения от 0 до k. Далее будем вычислять результирующий рейтинг алгоритмами FIND_RATING и MEDIANA_KEMENI. И сохранять тройки (согласованность, сложность, ошибка)

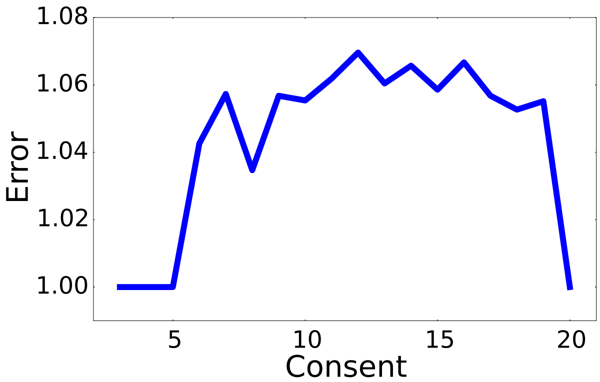
Рассмотрим первый тест, $n = 6, m = 3, k = 10$, количество запусков 10000 Прежде всего рассмотрим худший случай. Максимальные значения функции ошибки и минимальныв значения функции сложности в зависимости от согласованности



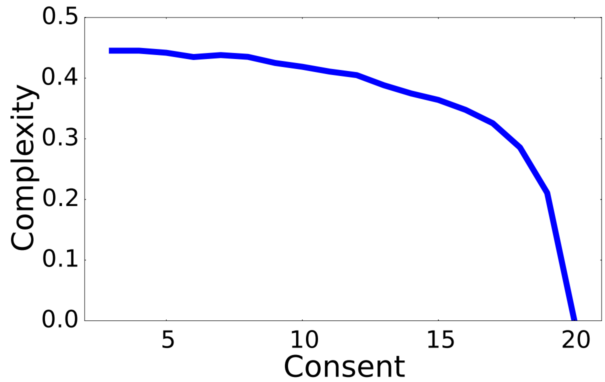
(a) Согласованность/Максимальная ошибка



(b) Согласованность/Худшая Сложность



(c) Согласованность/Средняя ошибка



(d) Согласованность/Средняя сложность

Рис. 4. Худшие и средние значения, n = 6, m = 3

Эти графики позволяют определить худшие случаи. На первом графике видно, что максимальная ошибка лежит около значения 1.5, значит ингогда мы на 50 процентов хуже чем медиана Кемени. Худшее значение сложности равно 0, что соответствует попаданию всех объектов на один фронт. Далее рассмотрим усредненные по абциссе значения предыдущих графиков, чтобы сделать вывод о качестве алгоритма. Значения графиков получаются усреднением всех результатов попавших в интервал длиной в одно целое. Это уже позволяет судить о качестве алгоритма в среднем.

Покажем те же самые графики при разных параметрах n и m в Приложении.

Также покажем сравнение среднего случая с наивным алгоритмом результирующего рейтинга, сортирующем объекты по величене среднего значения оценки экспертов на объекте. Сравнение построено на параметрах $n = 6, m = 3, k = 10$

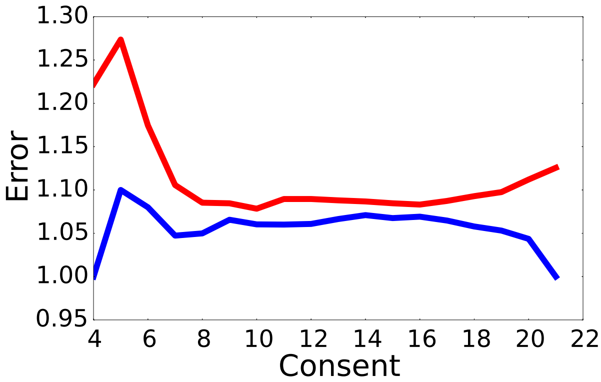


Рис. 5. Красный график - наивный алгоритм, синий график - алгоритм FIND_RATING

Эксперимент на данных платежных компаний

Сложность алгоритма зависит от количества и размеров Парето-фронтов. В предыдущем эксперименте, в качестве рассматриваемых признаков выбирались все признаки. Однако такой подход может привести к тому, что даже при хорошей согласованности экспертов, все объекты попадут на один слой, и алгоритм не даст никакого выйгрыша.

Проведем эксперимент на данных платежных компаний, чтобы выяснить какое количество признаков необходимо учитывать при построении Парето-оптимальных фронтов, чтобы добиться разделения выборки на наибольшее количество слоев.

Данные представляют собой 3-хмерную матрицу(объект-признак-эксперт). Всего в матрице 11 объектов, 17 признаков и 9 экспертов. Рассмотрим проекции этой матрицы при фиксированном признаке, то есть рассмотрим матрицы(объект-эксперт). Сопоставим каждой проекции вектор интегральных индикаторов, то есть каждому объекту при фиксированном признаке сопоставим среднее значение признаков, оценки экспертов, не оценивающих признак в среднее не входят. Тем самым получим матрицу(объект-признак) с усредненными по экспертам значениями признаков.

Рассмотрим случай, когда матрица не содержит пропусков. Если они есть, выкинем из рассмотрения объекты имеющие неопределнность в значении признака. В итоге остается 8 объектов.

Большое количество признаков не позволяет добиться хорошего разделения выборки на фронты, использование алгоритма парето-расслоения на всей матрице признаков нецелесообразно, то есть все объекты принадлежат одному слою. На рисунке 6 приведена зависимость максимального количества возможных слоев Парето-расслоения от количества выбранных признаков для 8-ми объектов данных. Заметим, что максимум достигается на 2 признаках и далее только уменьшается, значит для проведения Парето-расслоения нужно использовать от 2-х до 4-х признаков иначе выйгрыша по времени не будет.

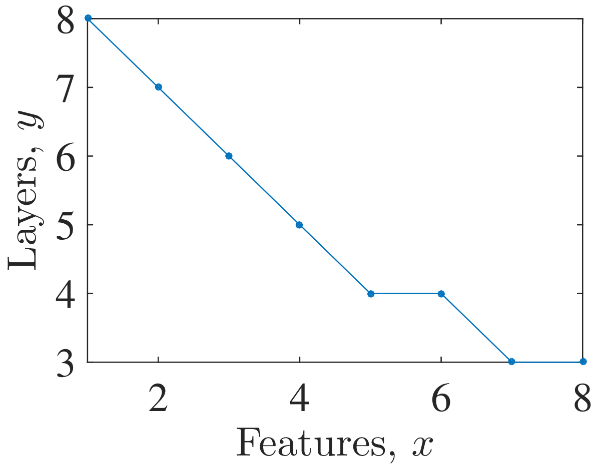


Рис. 6. Максимальное количество слоев Парето-расслоения в зависимости от количества признаков

Выводы о применимости алгоритма

На основе эксперементов можно сказать, что прежде всего гипотеза о том, что алгоритм хорошо работает на согласованных экспертах подтверждена. Это особенно видно на графике Согласованность/Средняя ошибка. При минимальных значениях функции согласованности качество алгоритма максимально, отклонение не больше нескольких процентов. При этом Средняя сложность также показывает лучший вариант. Далее идут средние значения согласованности, на них наблюдаются худшие результаты по качеству и средние результаты по сложности. И наконец максимальные значения функции согласованности, качество восстанавливается до максимального, но и сложность так же деградирует до стандартного алгоритма поиска медианы на всех объектах.

Из ограничений алгоритма стоит отметить, что при большом количестве экспертов, необходимо проводить отбор экспертов, которые будут давать максимальное разделение на Парето-фронты, иначе не будет прироста в производительности.

Также наблюдается закономерность, что при увеличении количества экспертов при фиксированном числе объектов, граница после которой сложность начинает сильно ухудшаться смещается влево.

Галвное свойство алгоритма в том, что он не ухуджает асимптотику вычисления медианы Кемени. Поэтому может работать с любой реализацией непосредственно поиска медианы Кемени, в том числе и её апроксимирующими вариантами. При этом, как было показано, для согласованных экспертов мы получаем заметный прирост в производительности, не теряя в качестве.

Дальнейшее исследование можно проводить как раз в алгоритмах поиска группы экспертов, которые определяют исход рейтинга, так что отбрасывание других экспертов не ухудшает качество. Здесь могут быть рассмотрены методы машинного обучения такие как, кластеризация и отбор признаков.

Литература

[1] Levenglick A.A. Young, H.P. Consistent extension of condorset's election principle. *SIAM J. Appl. Math.*, pages v.35, n. 2, 1978.

[2] John G. Kemeny. Mathematics without Numbers. *Daedalus*, 88(4), 1959.

[3] John J Bartholdi III, Craig A Tovey, and Michael A Trick. The computational difficulty of manipulating an election. *Social Choice and Welfare*, 6(3):227–241, 1989.

[4] Schudy Kenyon Mathieu. How to rank with few errors: A ptas for weighted feedback arc set on tournaments. *Symposium on Theory of Computing*, 2007.

[5] V. D. Nogin. The Edgeworth-Pareto principle and the relative importance of criteria in the case of a fuzzy preference relation. *Zh. Vychisl. Mat. Mat. Fiz.*, 43(11):1666–1676, 2003.

[6] Andrew Davenport Vincent Conitzer and Jayant Kalagnanam. Improved bounds for computing kemeny rankings. 2006.

[7] Mariya M. Medvednikova, Vadim V. Strijov, and Mikhail P. Kuznetsov. Algorithm of multiclass monotonous pareto-classification. *Notices of Tula State University*, 3:132–141, 2012.

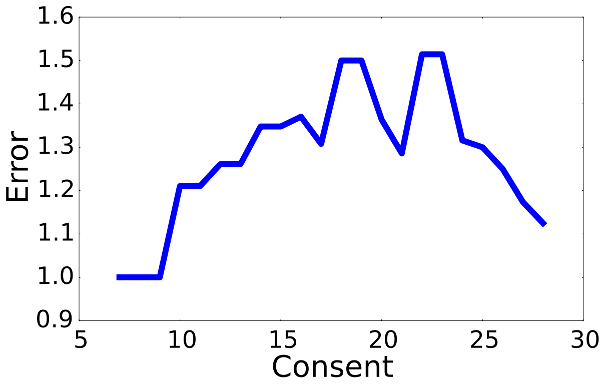
[8] Nir Ailon, Moses Charikar, and Alantha Newman. Aggregating inconsistent information: Ranking and clustering. *J. ACM*, 55(5):23:1–23:27, November 2008.

[9] Willem J Heiser and Antonio D'Ambrosio. Clustering and prediction of rankings within a kemeny distance framework. In *Algorithms from and for Nature and Life*, pages 19–31. Springer, 2013.

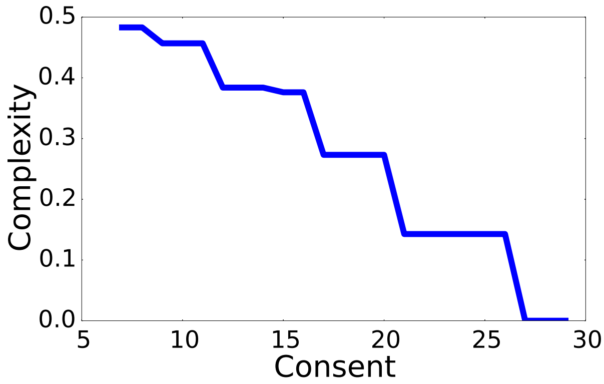
[10] M. Karpinski and W. Schudy. Faster algorithms for feedback arc set tournament, kemeny rank aggregation and betweenness tournament. *Cheong, O., Chwa, K.-Y., and Park, K. (Eds.): ISAAC 2010, Part I, LNCS 6506, pp. 3-14*.

[11] C. A. Tovey J. Bartholdi III and M. A. Trick. Voting schemes for which it can be difficult to tell who won the election. *Social Choice and Welfare, Vol. 6, No. 2 (1989), pp. 157–165.*

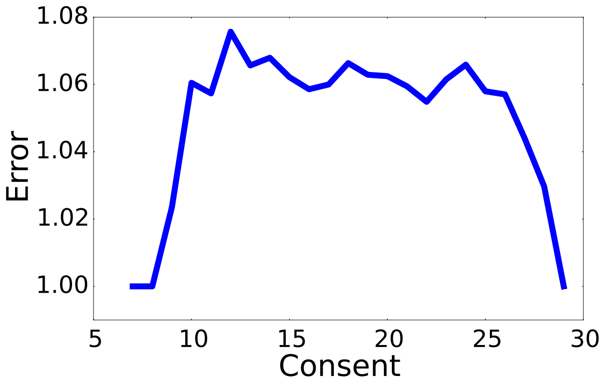
Приложение



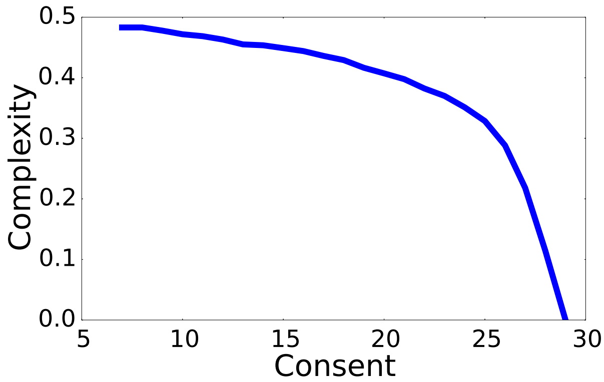
(a) Согласованность/Ошибка



(b) Согласованность/Сложность

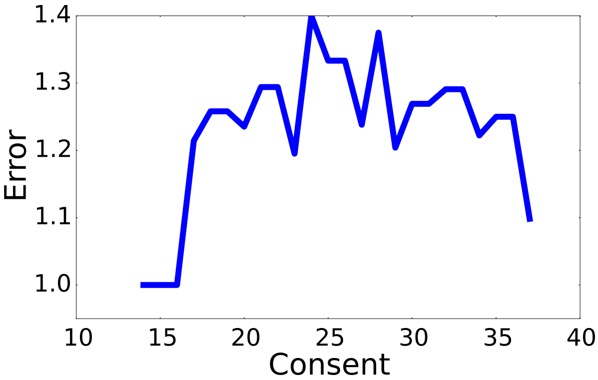


(c) Согласованность/Средняя ошибка

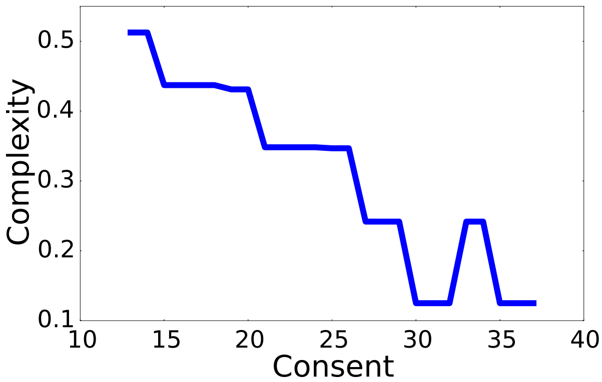


(d) Согласованность/Средняя сложность

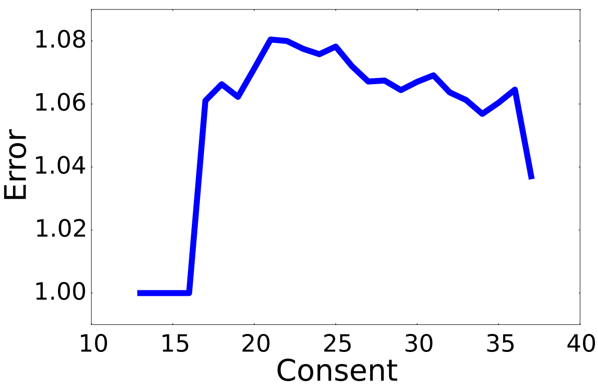
Рис. 7. Худшие и средние значения, n = 7, m = 3



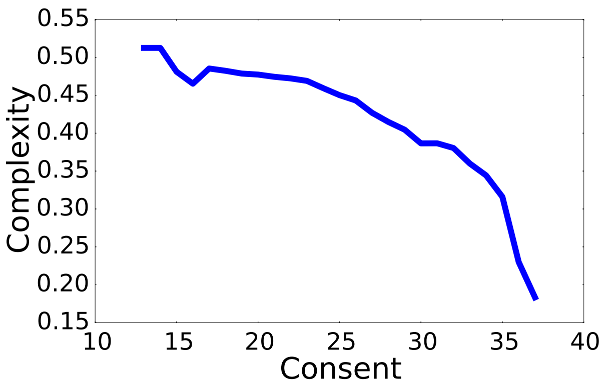
(a) Согласованность/Ошибка



(b) Согласованность/Сложность

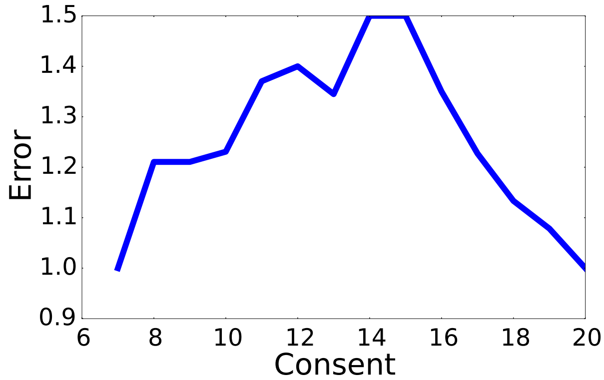


(c) Согласованность/Средняя ошибка

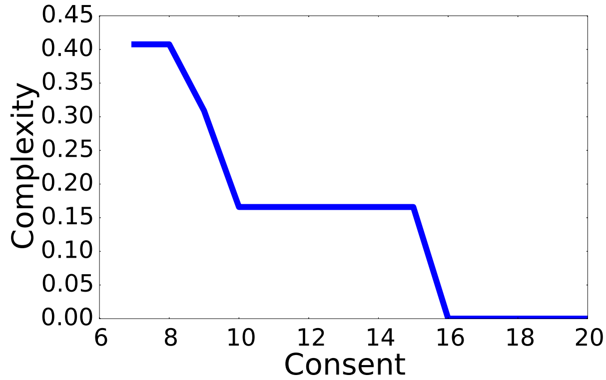


(d) Согласованность/Средняя сложность

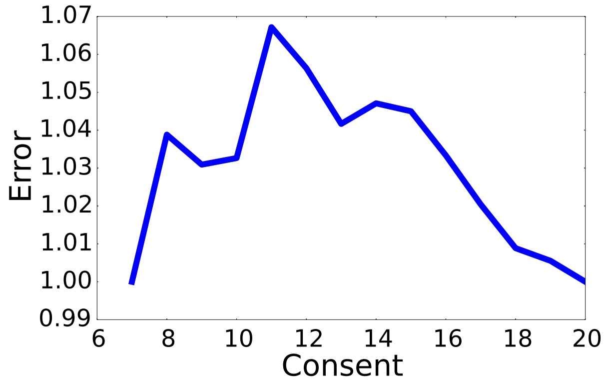
Рис. 8. Худшие и средние значения, n = 8, m = 3



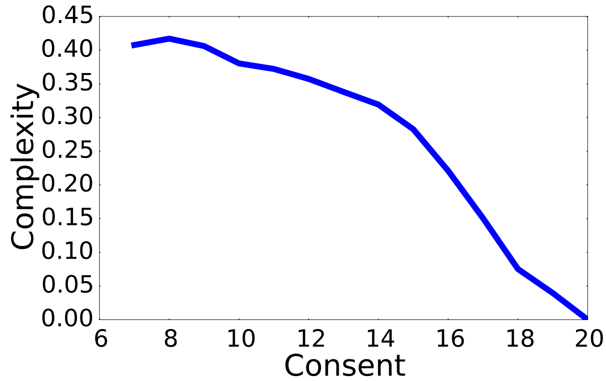
(a) Согласованность/Ошибка



(b) Согласованность/Сложность

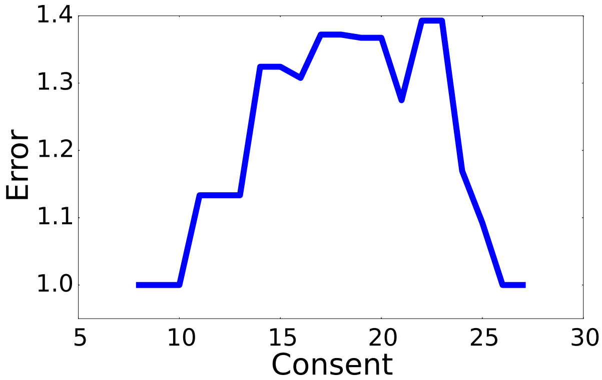


(c) Согласованность/Средняя ошибка

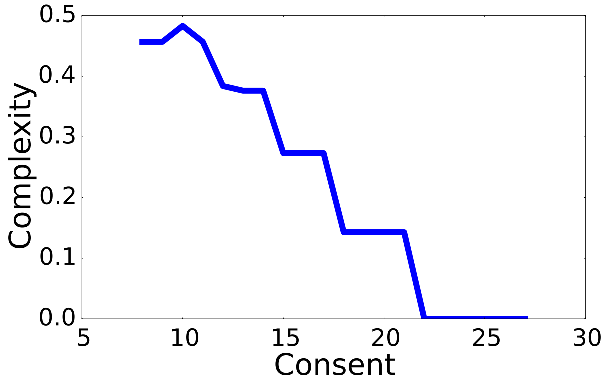


(d) Согласованность/Средняя сложность

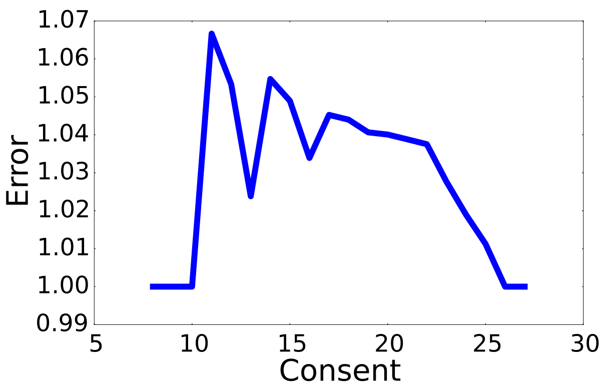
Рис. 9. Худшие и средние значения, n = 6, m = 4



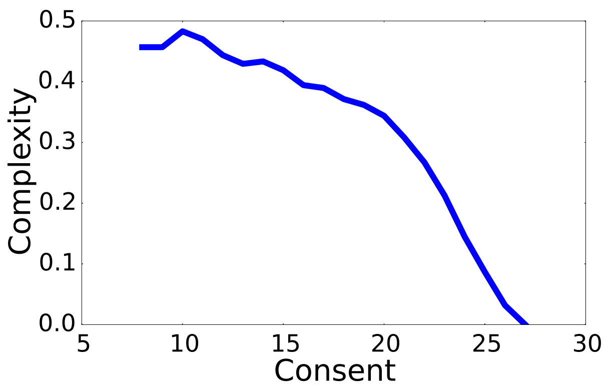
(a) Согласованность/Ошибка



(b) Согласованность/Сложность

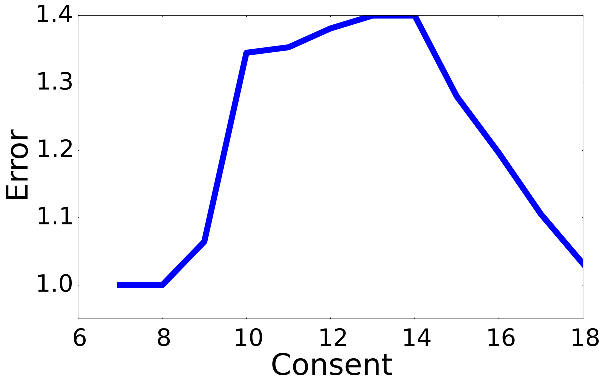


(c) Согласованность/Средняя ошибка

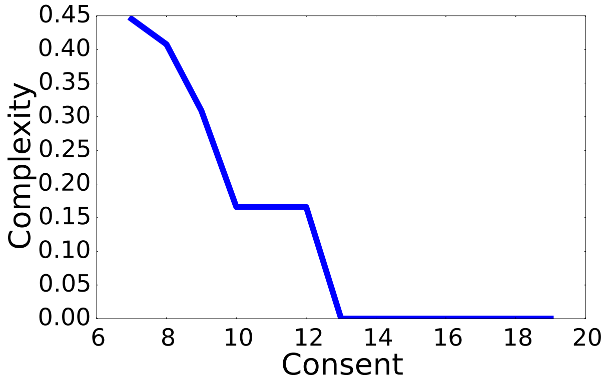


(d) Согласованность/Средняя сложность

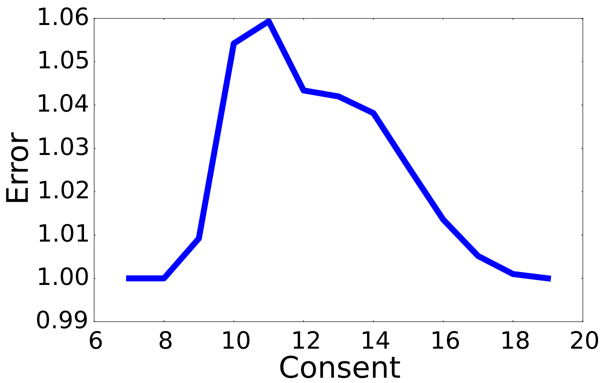
Рис. 10. Худшие и средние значения, n = 7, m = 4



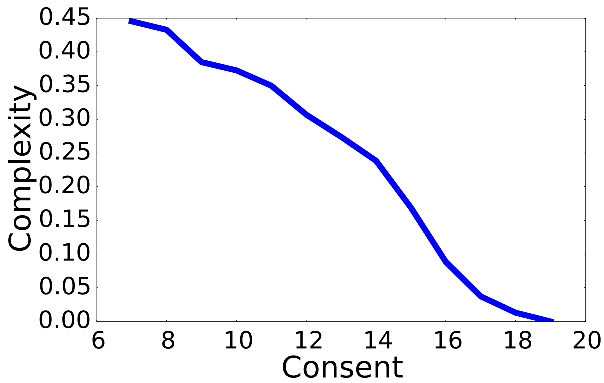
(a) Согласованность/Ошибка



(b) Согласованность/Сложность

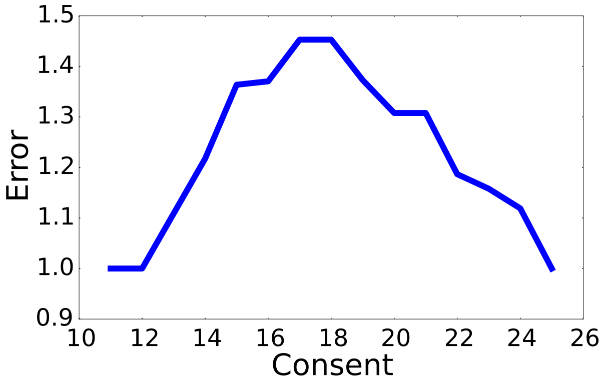


(c) Согласованность/Средняя ошибка

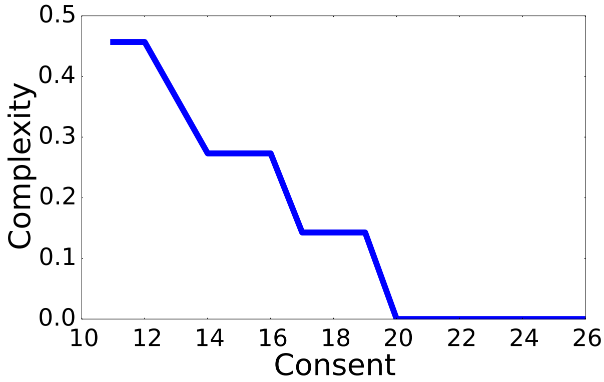


(d) Согласованность/Средняя сложность

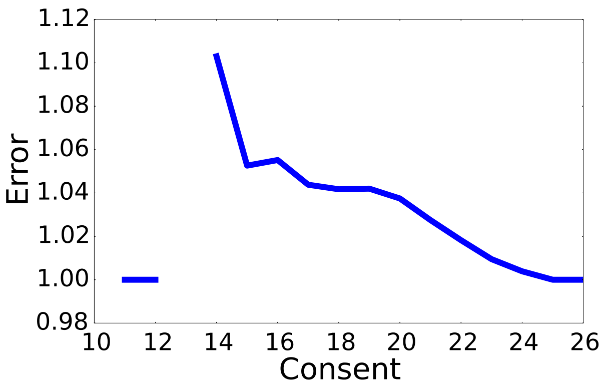
Рис. 11. Худшие и средние значения, n = 6, m = 5



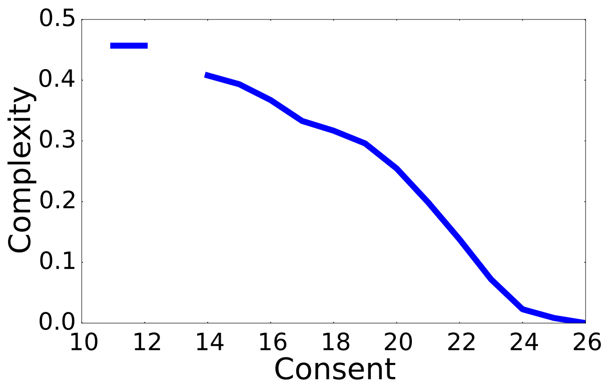
(a) Согласованность/Ошибка



(b) Согласованность/Сложность



(c) Согласованность/Средняя ошибка



(d) Согласованность/Средняя сложность

Рис. 12. Худшие и средние значения, n = 7, m = 5