# 1 Метрики кластеризации

Если имеется некоторое множество объектов $U$, то множеством кластеров над $U$ мы будем называть всякую совокупность подмножеств $E \subset 2^U$, такую, что

$$\forall e_1, e_2 \in E \Rightarrow e_1 \cap e_2 \neq \emptyset \Leftrightarrow e_1 = e_2,$$

т.е. входящие в него подмножества не пересекаются. Мы не требуем от множества кластеров покрытия множества $U$: кластеризация может быть неполной.

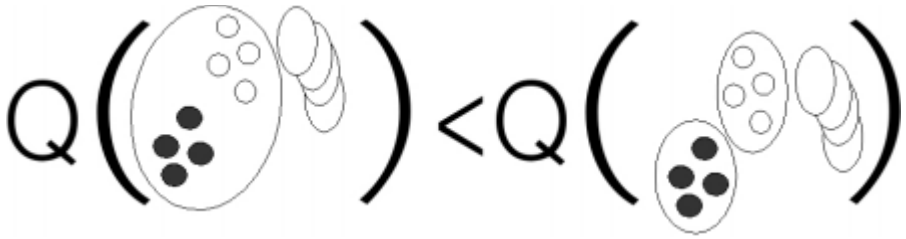
Будем теперь считать, что множеством объектов является множество документов $D = \{d_1, d_2, \dots, d_N\}$, и что заданы две кластеризации этого множества: образцовая кластеризация («разметка») $T = \{t_1, t_2, \dots, t_n\}$ и алгоритмическая кластеризация $C = \{c_1, c_2, \dots, c_m\}$. Будем обозначать через $t(d)$ тот кластер из образцовой кластеризации, которому принадлежит документ $d$, и через $c(d)$ тот кластер из алгоритмической разметки, которому принадлежит документ $d$.

Для простоты изложения будем считать, что алгоритмическая кластеризация содержит только такие кластера, которые имеют непустое пересечения хотя бы с одним кластером из разметки.

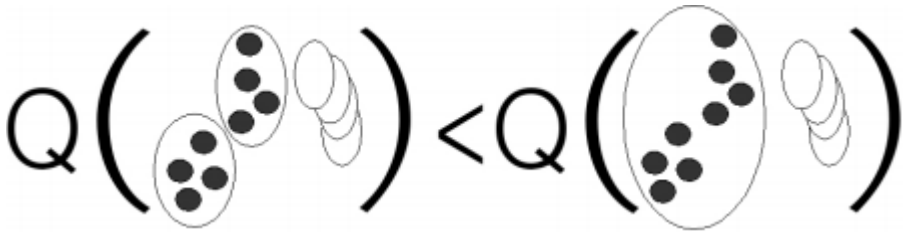
Подробнее о существующих метриках, критериях, которым они должны удовлетворять и т.д. можно почитать в статье http://nlp.uned.es/docs/amigo2007a.pdf (из этой же статьи взяты некоторые иллюстрации к настоящему тексту). О различных методах кластеризации можно почитать в статье http://www.datalaundering.com/download/mm2.pdf.

## 1.1 Формальные критерии для метрик кластеризации

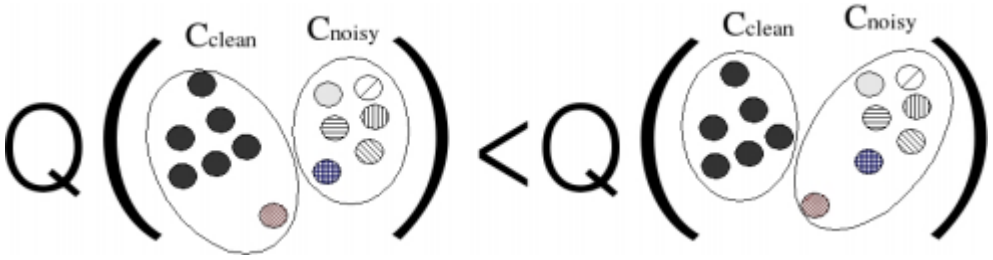
Формирование более однородных кластеров приводит к улучшению значения метрики (однородность кластеризации, «*cluster homogeneity*»):



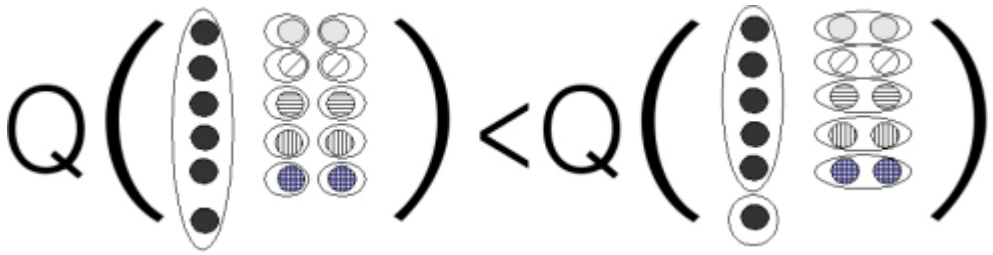

Формирование более полных кластеров приводит к улучшению метрики (полнота кластеризации, «*cluster completeness*»):



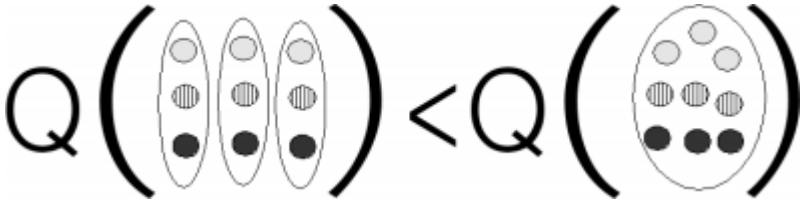

Кластеризация, в которой нерелевантный элемент попадает в кластер с низкой точностью лучше, чем та, в которой нерелевантный элемент попадает в кластер с высокой точностью («*rag bag*»):




Исправление незначительной ошибки в большом кластере хуже, чем исправление большого числа ошибок в маленьких кластерах («*size vs quantity*»):




Большее количество неточных кластеров – меньше качество:




### 1.2 Существующие метрики кластеризации

Мы рассмотрим только небольшое количество метрик некоторые семейства метрик, оставив вне рассмотрения метрики, основанные на энтропии, поскольку значения этих метрик весьма сложны для интерпретации.

#### 1.2.1 Метрики, производные от метрики purity («чистота кластеров»)

$$Purity(D,T,C) = \frac{1}{\sum_{t\in T}|t|}\sum_{t\in T}|t| \max_{c\in C} Precision(c,t),$$

где

$$Precision(c,t) = \frac{|c \cap t|}{|c|}.$$

Эта метрика принимает большие значения, если для каждой темы существует достаточно точный кластер, содержащий ее элементы.

$$InvPurity(D,T,C) = \frac{1}{\sum_{c\in C}|c|}\sum_{c\in C}|c| \max_{t\in T} Precision(c,t)$$

Инвертированная метрика по отношению к «чистоте»: меняются местами «разметка» и «алгоритм». Таким образом, измеряется уже не точность, а полнота.

Большие значения обеих этих метрик не означают, что кластеризация хороша. Это легко видеть на следующем примере.

Пусть имеются кластера $c_1, c_2$ и тема $t$, причем их размеры соотносятся следующим образом:

$$|c_1| \ll |t| \ll |c_2|,$$

и при этом

$$Precision(c_1, t) = 1, Recall(c_2, t) \approx 1,$$

Этого можно добиться, например, так: кластер $c_1$ содержит только небольшое количество документов темы $t$ и только их, кластер $c_2$ – все остальные документы темы $t$ и большое количество других документов.

Тогда для темы $t$ возможно предъявить и кластер, для которых значение точности близко к единице, и кластер, для которого значение полноты близко к единице. В то же время, эти кластера будут различны, так что для темы $t$ невозможно будет предъявить один-единственный кластер, адекватно ее представляющий (с большой и точностью, и полнотой).

Некоторым образом с этой проблемой борется $F$-мера, основанная на «чистоте»:

$$F(D, T, C) = \frac{1}{\sum_{t \in T} |t|} \sum_{t \in T} |t| \max_{c \in C} F(c, t),$$

где $F(c, t)$ – $F$-мера точности и полноты:

$$Recall(c, t) = \frac{|c \cap t|}{|t|},$$

$$F(c, t) = \frac{2 \cdot Recall(c, t) \cdot Precision(c, t)}{Recall(c, t) + Precision(c, t)}.$$

Впрочем, эта метрика, равно как и метрики чистоты и инвертированной чистоты, не проходит критериев полноты и *rag bag*. Это связано с тем, то при вычислении всех указанных метрик используются лишь кластера, на которых достигаются максимальные значения метрик. Поэтому изменения в кластерах, на которых метрика не достигает максимального значения ни для одной из тем, не окажут влияния на интегральную метрику.

### 1.2.2 Классификационные метрики

Эта группа метрик рассматривает кластеризацию как бинарную классификацию на множестве пар элементов. Например, «классификационные» точность и полнота:

$$BinaryPrecision = \frac{\sum_{1 \le i < j \le N} [c(d_i) = c(d_j) \wedge t(d_i) = t(d_j)]}{\sum_{1 \le i < j \le N} [c(d_i) = c(d_j)]},$$

$$BinaryRecall = \frac{\sum_{1 \le i < j \le N} [c(d_i) = c(d_j) \wedge t(d_i) = t(d_j)]}{\sum_{1 \le i < j \le N} [t(d_i) = t(d_j)]}.$$

Ясно, что при использовании классификационного подхода возможно использовать и другие классификационные метрики.

Метрики этого класса не удовлетворяют критерию *rag bag* (т.к. количество нерелевантных пар, порождаемых нерелевантным документом, зависит лишь от размера кластера, но не от качества этого кластера). Кроме того, они не удовлетворяют критерию *size vs quantity*, поскольку количество пар, порождаемых некоторым кластером, пропорционально квадрату его размера; добавление одного релевантного документа в кластер размера $2k$ порождает $2k$ положительных пар – столько же, сколько порождается при верном объединении $k$ двухэлементных кластеров.

### 2.2.3 Бикубические (BCubed) метрики

Эти метрики основаны на подсчете поэлементных характеристик точности и полноты. Для документа $d$ определена точность и полнота кластера, в который его отнес алгоритм:

$$BCP(d) = \frac{|c(d) \cap t(d)|}{|c(d)|},$$

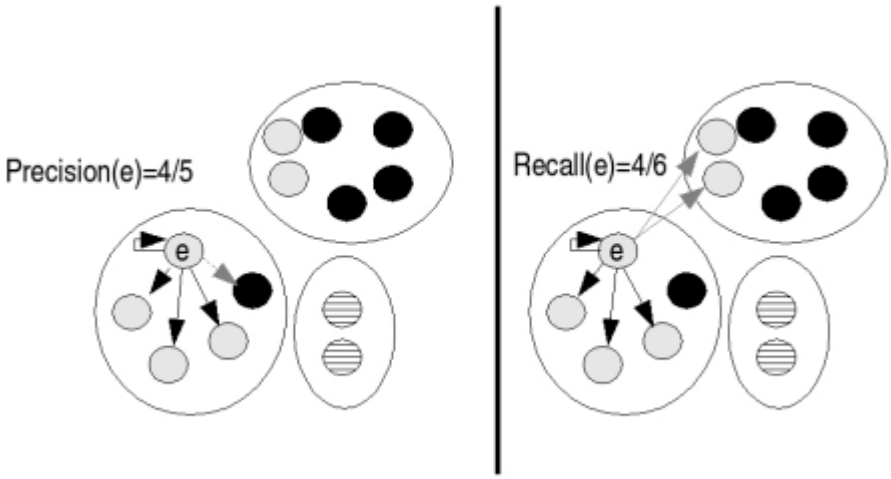
$$BCR(d) = \frac{|c(d) \cap t(d)|}{|t(d)|}.$$

Интегральными метриками точности и полноты являются усреднения по всем размеченным документам:

$$BCP(D,T,C) = \frac{1}{\sum_{t\in T}|t|} \sum_{d\in \bigcup_{t\in T} t} \frac{|c(d) \cap t(d)|}{|c(d)|},$$

$$BCR(D,T,C) = \frac{1}{\sum_{t\in T}|t|} \sum_{d\in \bigcup_{t\in T} t} \frac{|c(d) \cap t(d)|}{|t(d)|}.$$

Процесс вычисления метрик для конкретного документа может быть проиллюстрирован следующим образом:




В качестве единой интегральной метрики можно использовать -меру интегральных точности и полноты, либо усредненную подокументную $F$-меру. Бикубическая -мера удовлетворяет всем перечисленным критериям.

### 1.3 Новостные бикубические метрики

Все рассмотренные выше метрики обладают одним очень важным недостатком: различные темы вносят в значение интегральных метрик различный вклад в зависимости от своих размеров. Вклад темы в метрики чистоты или бикубические метрики прямо пропорционален размеру этой темы, а в классификационные метрики – пропорционален квадрату размера этой темы.

В целом, нельзя отрицать наличия зависимости «важности» темы от ее размера. В то же время, даже для новостей эта зависимость совсем не очевидна: так, существует большое количество рубрик и выпусков, в которых часто встречаются кластера малого размера. Поэтому утверждение о том, что для новостей важно хорошо кластеризовать лишь большие темы, которые попадают в топ-5, неверно.

В любом случае, хотелось бы иметь возможность при необходимости указывать важность той или иной темы явно так, чтобы вклад этой темы в интегральную метрику был пропорционален весу этой темы. Другими словами, хочется иметь метрику качества кластеризации для каждой конкретной темы, на основании которой затем вычислять интегральную метрику, учитывающую веса тем.

В простейшем случае можно считать, что все темы имеют одинаковый вес.

Добиться указанного свойства для бикубических метрик несложно. Определим бикубические метрики качества тем следующим образом:

$$BCP(t) = \frac{1}{|t|}\sum_{d \in t}\frac{|c(d) \cap t|}{|c(d)|},$$

$$BCR(t) = \frac{1}{|t|}\sum_{d \in t}\frac{|c(d) \cap t|}{|t|}.$$

Интегральными метриками в таком случае будем считать результат усреднения метрик по темам:

$$BCP(D,T,C) = \frac{1}{|T|}\sum_{d \in D} BCP(t),$$

$$BCR(D,T,C) = \frac{1}{|T|}\sum_{d \in D} BCR(t).$$

При необходимости можно очевидным образом ввести различные веса для различных тем.

Поскольку документы, входящие в один и тот же кластер, дают одинаковый вклад в значение метрик бикубических точности и полноты для конкретной темы, мы можем переписать формулы следующим образом. Пусть $c_{i_1}, c_{i_2}, \dots, c_{i_k}$ – все кластера, имеющие непустое пересечение с темой $t$. Тогда

$$BCP(t) = \frac{1}{|t|}\sum_{j=1}^{k}\frac{\left|c_{i_j} \cap t\right|^2}{\left|c_{i_j}\right|},$$

$$BCR(t) = \frac{1}{|t|^2}\sum_{j=1}^{k}\left|c_{i_j} \cap t\right|^2.$$

То есть, метрики квадратично зависят от размера пересечения. Это их свойство нужно всегда надо иметь в виду, т.к. это может иметь неожиданные последствия.

Рассмотрим, например, две темы одинакового размера, $t_1$ и $t_2$, которые имеют непустые пересечения с кластерами, соответственно, $c_1$ и $c_2$, и оба этих кластера имеют точность, равную единице. Предположим, имеются два варианта улучшения кластеризации: один подклеивает дополнительный документ из $t_1$ в $c_1$, а другой – из $t_2$ в $c_2$. Тогда согласно метрике $BCR$ лучшим будет тот вариант кластеризации, который подклеивает документ в больший из кластеров. Это может показаться неожиданным, т.к. с точки зрения обыкновенной, «линейной», полноты улучшения идентичны, т.к. увеличивают полноту одной из тем на одну и ту же величину.

При оптимизации этих метрик можно столкнуться с тем, что мы стремимся увеличивать метрики для тех тем, для которых они и так уже достаточно велики.

Впрочем, квадратичная зависимость объяснима с точки зрения физического смысла (вытекающего из вероятностной постановки бикубических метрик). К тому же, именно квадратичность оказывается тем свойством, которое обеспечивает метрике разумность. Действительно, представим себе линейную метрику полноты:

$$BCR(t) = \frac{1}{|t|}\sum_{j=1}^{k}\left|c_{i_j} \cap t\right|.$$

Тогда окажется, что эта метрика в любом случае равняется единице.

### 1.4 Метрика AlexRecall

Новостная специфика заключается в том, что для отражения каждой темы возможно предъявить только один кластер, причем каждый кластер возможно предъявить только для одной темы.

Если тема уже представлена каким-либо кластером, представлять второй для отражения той же темы нет никакого смысла: новой информации в нем не будет. При этом он будет занимать место (в топах рубрик, в поисковой выдаче или в списке связанных сюжетов), которое можно было бы потратить на отображение какой-либо другой темы.

Выбор темы, которую представляет некоторый кластер, происходит на этапе аннотирования: тогда выбирается заголовок, который, конечно, может представлять только одну из тем, представленных в кластере.

Мы хотели создать метрику, учитывающую эти новостные особенности. Очень хотелось иметь метрику, которая сопоставляет каждому кластеру некоторую тему и оценивает качество этого сопоставления. Впрочем, указать единственное сопоставление часто невозможно, а, если и возможно, то это чревато большой неустойчивостью метрики (небольшие изменения в кластерах могут приводить к существенным изменениям в сопоставлении).

Поэтому была предложена метрика, рассматривающая вероятностно все возможные сопоставления.

Скажем, что вероятность сопоставления кластеру $c \in C$ темы $t \in T$ равна точности:

$$P(f(c) = t) = \frac{|c \cap t|}{|c|},$$

а все сопоставления происходят независимо. Тогда вероятность построить функцию $f: C \to T$ равна

$$P(f) = \prod_{c \in C} \frac{|c \cap f(c)|}{|c|}.$$

Скажем теперь, что величина «покрытия» темы $t$ при отображении $f$ определяется величиной

$$Cover(t, C, f) = \max_{c \in f^{-1}(t)} \frac{|c \cap t|}{|t|}.$$

Это разумно, поскольку разумно было бы выбирать в качестве представителя темы тот кластер, который отражает ее с наибольшей полнотой.

Метрика покрытия темы $t$ кластеризацией $C$ – это математическое ожидание покрытия темы:

$$AlexRecall(t, C) = \sum_{f: C \to T} P(f) \cdot Cover(t, C, f).$$

А интегральная метрика – усредненное значение покрытия всех тем:

$$AlexRecall(T, C) = \sum_{t \in T} AlexRecall(t, C).$$

Вычислять величину $AlexRecall(t, C)$ проще следующим образом. Пусть $\{c_{i_1}, c_{i_2}, \dots, c_{i_k}\}$ – множество всех кластеров, имеющих непустое пересечение с $t$, занумерованное по невозрастанию размера пересечения:

$$|t \cap c_{i_1}| \geq |t \cap c_{i_2}| \geq \cdots \geq |t \cap c_{i_k}|.$$

Кластер $c_{i_1}$ отобразится на $t$ с вероятностью

$$P(f(c_{i_1}) = t) = \frac{|t \cap c_{i_1}|}{|c_{i_1}|},$$

и в этом случае величина покрытия будет равна

$$\frac{|t \cap c_{i_1}|}{|t|}$$

вне зависимости от того, каким образом отобразились другие кластера.

Если же кластер $c_{i_1}$ не отобразится на $t$, то величина покрытия будет вычисляться по остальным кластерам.

Таким образом, можно написать итеративный метод вычисления метрики для темы:

$$Cover\left(t,\left\{c_{i_1},c_{i_2},\dots,c_{i_k}\right\}\right)=\frac{\left|t\cap c_{i_1}\right|}{\left|c_{i_1}\right|}\cdot\frac{\left|t\cap c_{i_1}\right|}{|t|}+\left(1-\frac{\left|t\cap c_{i_1}\right|}{\left|c_{i_1}\right|}\right)\cdot Cover\left(t,\left\{c_{i_2},\dots,c_{i_k}\right\}\right)$$

Метрика AlexRecall не во всех ситуациях удовлетворяет указанным в начале условиям. Если кластер, на котором достигается максимум полноты для некоторой темы, имеет при этом точность, равную единице, то изменения в других кластерах, пересекающихся с этой темой, не окажут влияния на метрику этой темы – то есть, могут оказаться не выполненными свойства точности и полноты (так же, как и для метрик purity). Однако причины, по которым это происходит, обусловлены именно продуктовыми требованиями к качеству новостей. В тех случаях, когда кластера не являются стопроцентно точными, все свойства оказываются выполненными.

Заметим также, что, как и в бикубических метриках, в метрике *AlexRecall* присутствует квадратичная зависимость от размера пересечения.

## 2 Алгоритм кластеризации

Алгоритм кластеризации состоит из трех этапов:

1. Фильтрация, в процессе которой определяются пары документов, для которых имеет смысл определять близость.
2. Осуществление предсказаний близости для выбранных пар.
3. Собственно кластеризация по полученному графу близостей.

Рассмотрим сначала наш алгоритм кластеризации (п. 3), который является разновидностью графовой кластеризации и не имеет никаких особенностей, связанных с обработкой новостей. Т.е., алгоритм универсален: его можно применять для любых задач, в которых присутствует граф близостей некоторого множества объектов.

### 2.1 Агломеративный алгоритм кластеризации

Предполагаем, что на множестве пар документов определена функция близости:

$$w: D^2 \to [0,1].$$

Будем считать, что документ всегда близок к самому себе:

$$w(d,d)=1.$$

Пусть имеется некоторое разбиение множества документов:

$$D = D_1 \sqcup D_2 \sqcup \dots \sqcup D_k$$

Для каждого документа можно определить точность и полноту: если $d \in D_i$, то

$$P(d)=\frac{1}{|D_i|}\sum_{d'\in D_i} w(d,d_i)$$

$$R(d) = \frac{\sum_{d' \in D_i} w(d, d_i)}{\sum_{d' \in D} w(d, d_i)}$$

Аналогичные метрики для любого подмножества $D_i \subseteq D$ можно вычислить простым усреднением:

$$P(D_i) = \frac{1}{|D_i|} \sum_{d \in D_i} P(d), R(D_i) = \frac{1}{|D_i|} \sum_{d \in D_i} R(d).$$

Агрегируем метрики:

$$M_\alpha\big(P(D_i), R(D_i)\big) = \big(R(D_i)\big)^\alpha \cdot P(D_i).$$

Такое агрегирование проводится по аналогии с метрикой AlexRecall, в которой точность и полнота умножаются. Коэффициент $\alpha$ управляет гранулярностью кластеризации: чем больше этот параметр, тем более важной оказывается полнота.

Предположим, мы решили объединить два кластера: $D_i$ и $D_j$. До объединения метрики по входящим в них документам были равны

$$P_1 = P(D_i) + P\big(D_j\big),$$

$$R_1 = R(D_i) + R\big(D_j\big),$$

а после объединения они окажутся равны

$$P_2 = P\big(D_i \sqcup D_j\big),$$

$$R_2 = R\big(D_i \sqcup D_j\big).$$

Тогда метрика качества такого объединения будет равна

$$M_\alpha(P_2, R_2) - M_\alpha(P_1, R_1).$$

Изначально алгоритм на каждом шаге объединял два кластера, для которых эта метрика достигает максимального значения, и останавливался, если не было такой пары кластеров, для которых метрика объединения положительна. Оказалось, впрочем, что такая стратегия вполне способна порождать совсем нелогичные кластера, и лучшее качество достигается, если производить объединение таких кластеров, точность объединения которых максимальна, а критерий останова оставить тем же:

$$M_\alpha(P_2, R_2) > M_\alpha(P_1, R_1).$$

Скорость выполнения алгоритма определяется тем, насколько быстро оказывается возможным вычислять метрики качества объединения для пар кластеров.

Будем хранить для каждой имеющейся пары кластеров следующие величины:

$$pCross\big(D_i, D_j\big) = \sum_{d' \in D_i} \sum_{d'' \in D_j} w(d', d''),$$

$$rCross(D_i, D_j) = \frac{1}{2} \cdot \sum_{d' \in D_i} \sum_{d'' \in D_j} \left[ w(d', d'') \cdot \left( \frac{1}{r(d')} + \frac{1}{r(d'')} \right) \right],$$

где

$$r(d) = \sum_{d' \in D} w(d, d').$$

Тогда оказывается, что:

$$P(D_i) = \frac{1}{|D_i|^2} pCross(D_i, D_i),$$

$$R(D_i) = \frac{1}{|D_i|} rCross(D_i, D_i),$$

$$P(D_i \sqcup D_j) = \frac{pCross(D_i, D_i) + pCross(D_j, D_j) + 2 \cdot pCross(D_i, D_j)}{|D_i \sqcup D_j|^2},$$

$$R(D_i \sqcup D_j) = \frac{rCross(D_i, D_i) + rCross(D_j, D_j) + 2 \cdot rCross(D_i, D_j)}{|D_i \sqcup D_j|}.$$

Если кластеры $D_i$ и $D_j$ были объединены, то для каждого соседнего одному из них кластера $D_k$ необходимо вычислить величины

$$pCross(D_i \sqcup D_j, D_k) = pCross(D_i, D_k) + pCross(D_j, D_k),$$

$$rCross(D_i \sqcup D_j, D_k) = rCross(D_i, D_k) + rCross(D_j, D_k)$$

и, используя их, вычислить метрики объединения.

Благодаря тому, что объединения производятся исходя из максимальной точности, оказывается возможным управлять процессом объединения кластеров. Так, например, был добавлен механизм объединения документов с похожими заголовками: если два заголовка достаточно похожи, документам проставляется близость, равная единице. В таком случае кластеризация на первых порах объединяет документы с похожими заголовками, а уж затем – все остальные, что приводит к улучшению пользовательского свойства «одинаковые заголовки должны попадать в один сюжет». Аналогично можно осуществлять и другие «бустинги» для кластеризации: например, проставлять близости, равные единице, сообщениям, посвященным одним и тем же спортивным событиям, и так далее.

Приведенный алгоритм никогда достаточно аккуратно не сравнивался с аналогами. Тут необходимо сказать, что такие сравнения сильно затруднены с учетом сложности нашей задачи (кластеризуются сотни тысяч документов), и большинство алгоритмов на таких задачах будут работать либо очень плохо, либо очень медленно. Тем не менее, на имеющихся предсказаниях (в их состоянии на февраль 2013 года) было проведено сравнение агломеративного алгоритма с алгоритмами *MCL* и *Oslom*. Результаты – не в пользу последних:

| | Agglomerative | MCL | Oslom |
|---|---|---|---|
| **AlexRecall** | 0.80088 | 0.64857 | 0.39953 |
| **BCP** | 0.86399 | 0.68525 | 0.44486 |
| **BCR** | 0.83504 | 0.82931 | 0.73295 |
| **BCF1** | 0.82706 | 0.70527 | 0.47306 |

К тому же, указанные алгоритмы работают на порядок дольше агломеративного.

## 2.2 Фильтрация

### 2.2.1 Представление документов

Документы представляются следующим образом. Пусть $\langle w_1, w_2, \dots, w_l \rangle$ – последовательность слов документа $d$. Обычно новости пишутся таким образом, что основная информация содержится в начале документа, поэтому мы вводим функцию $\gamma: \mathbb{N} \to \mathbb{R}_+$, монотонно убывающую, и с ее помощью определяем функцию частоты слова в документе:

$$\gamma(d, w) = \sum_{i=1}^{l} \gamma(i) \cdot [w_i = w].$$

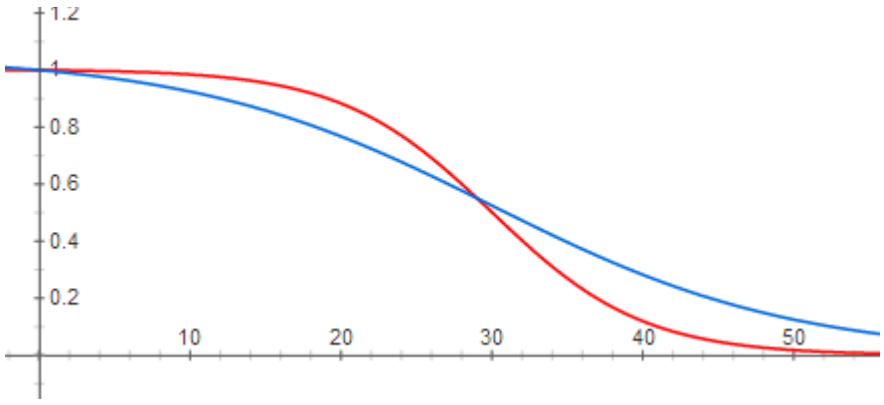
Конкретно мы используем функцию $\gamma$ следующего вида:

$$\gamma(k) = \frac{1 + \exp(-\alpha \cdot \beta)}{1 + \exp\big(-\alpha \cdot (\beta + 1 - k)\big)},$$

где $\alpha$ и $\beta$ – параметры, имеющие следующий смысл:

- $\beta$ – параметр, такой, что $\gamma(\beta + 1) \approx 0.5$.
- $\alpha$ отвечает за резкость перехода функции из значений, близких единице, к значениям, близким к нулю.

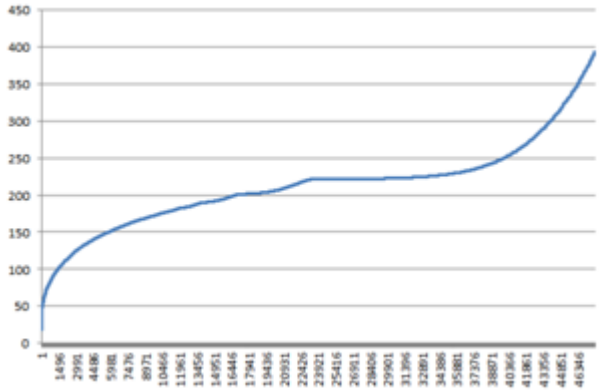
На рисунке ниже представлены графики функции $\gamma$ для параметров $\alpha = 0.1, \beta = 30$ (красная) и $\alpha = 0.2, \beta = 30$ (синяя).




### 2.2.2 *Tf-Idf*

Для использования весов типа *tf-idf* необходимо еще определиться с тем, что такое *idf*. Можно использовать аналог корпусного *idf* (например, по первому вхождению слова в документ), можно использовать инверсные корпусные частоты слов, можно использовать частоты из *pure* и т.д. По

факту, все частотные *idf*'ы весьма плохо работают с редкими словами. График зависимости *idf* от позиции слова в ранжированном по уменьшению частоты списке выглядит следующим образом:




Видно, что вес сильно растет на хвосте, и слова оттуда получают непропорционально большой вес даже при небольших значениях *tf*. При этом первая половина графика очень хорошо аппроксимируется логарифмом позиции (работает закон Ципфа). Поэтому в итоге в качестве *idf* мы используем логарифм позиции слова в отсортированном по частоте списке слов: в полезной зоне он работает так же, как и *idf* по частотам, но при этом не допускает сильного роста веса в зоне низкочастотных слов.

### 2.2.3 Определение ключевых слов и фильтрация

Для каждого документа определим список его «ключевых слов» и список «слов-кандидатов». Эти слова будут иметь следующий смысл: будем считать, что два документа прошли фильтрацию, если хотя бы $k$ ключевых слов одного из документов встречаются среди слов-кандидатов другого (отношение симметрично).

Изначально, ключевые слова – это просто первые $c_1$ слов документа, отранжированных по *tf-idf*, а слова-кандидаты – это первые $(c_1 + c_2)$ слов в этом списке. Оказывается, что при таком определении уже получается очень хорошая фильтрация.

Впрочем, дело портят документы, содержащие большое количество редких слов. У таких документов топы слов по *tf-idf* забивают редкие слова, и для них находится мало соседей. Ситуацию можно исправить, увеличив числа $c_1$ и $c_2$, но это увеличит количество пар, проходящих фильтрацию, и замедлит метод.

Поэтому осуществляется следующая процедура: вычисляется средний вес слова на $c_1$-й позиции и средний вес слова на позиции $(c_1 + c_2)$ среди всех документов. Затем для каждого документа ключевыми считаются слова, для которых выполнено хотя бы одно из двух условий:

- слово находится среди первых $c_1$ при ранжировании по *tf-idf*;
- вес по *tf-idf* превышает средний вес слова на позиции $c_1$, вычисленный среди всех документов.

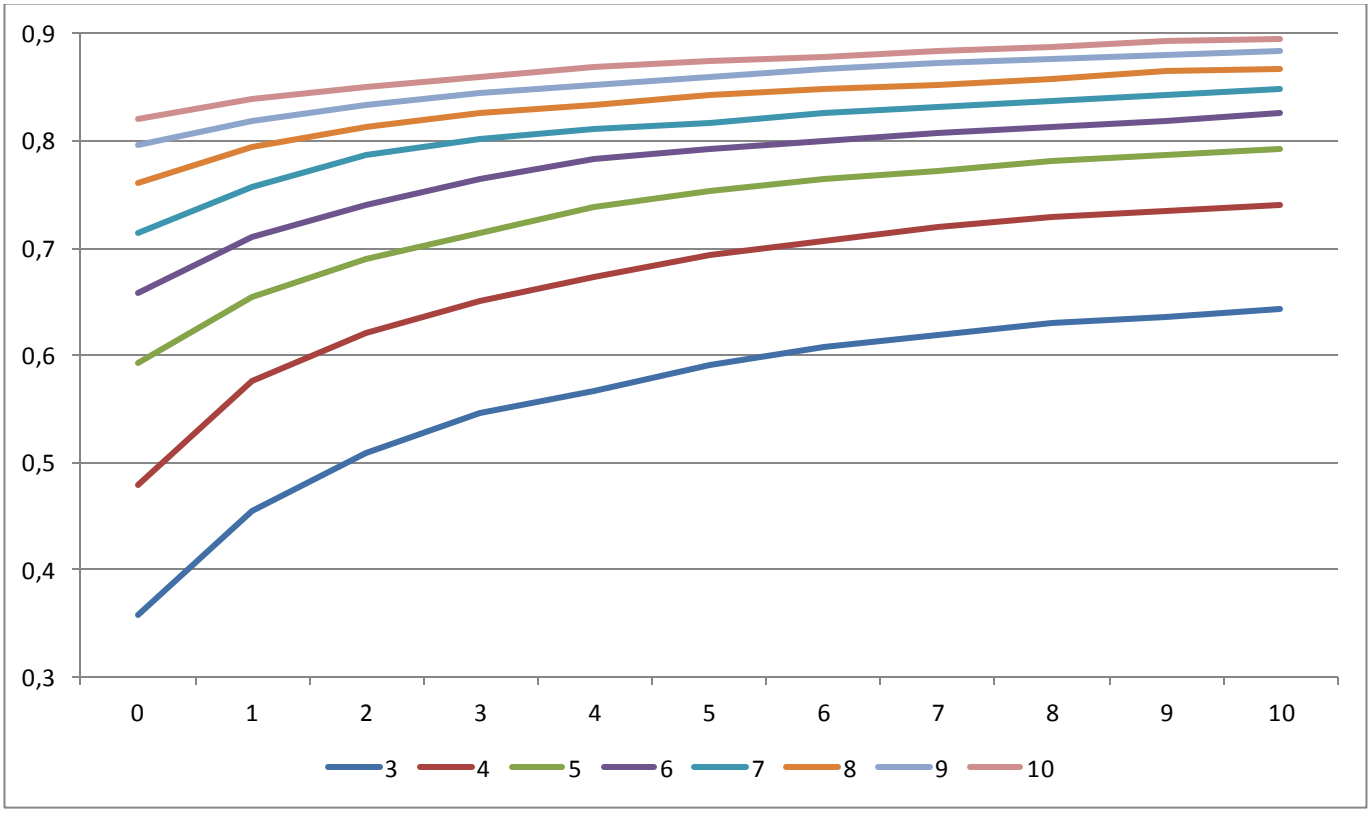
Аналогично определяются слова-кандидаты, но при их определении используется число $(c_1 + c_2)$.

### 2.2.4 Исследование качества фильтрации

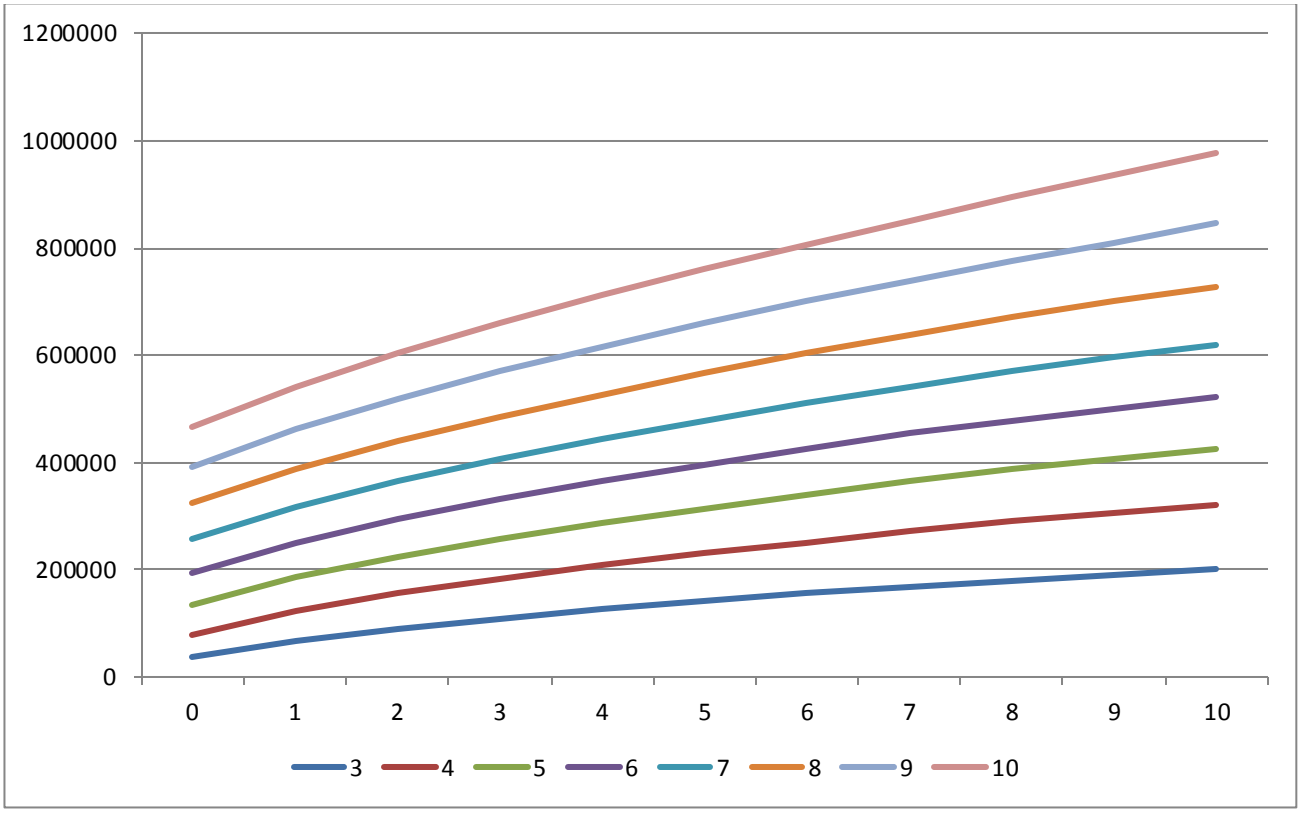
Исследовать качество фильтрации достаточно просто. Для этого необходимо для всех полученных в ходе фильтрации пар проставить верный ответ в качестве близости, запустить на полученных парах алгоритм кластеризации, и метрики этого алгоритма окажутся верхней границей на качество кластеризации (при условии идеальной функции попарной близости).

Впрочем, при этом способе контроля качества фильтрации оказывается весьма опасно изменять параметры алгоритма: поскольку используется идеальная функция близости, оказывается выгодным объединять два любых кластера, между которыми есть хотя бы какая-то положительная связь. Такая стратегия оказывается, разумеется, очень плохой в случае использования реальных функций близости.

Ниже показано, как изменяется метрика *AlexRecall* при изменении величин $c_1$ и $c_2$, а также каким образом меняется количество получаемых после фильтрации пар.




Разными цветами выделены различные значения параметра $c_1$ (изменяется от 3 до 10), по горизонтали отложены значения параметра $c_2$ (изменяется от 0 до 10). По вертикали отложены значения метрики *AlexRecall*. Другой важный показатель метода фильтрации – количество генерируемых пар. Зависимость количества пар от параметров фильтрации отражена на графике ниже.




Обычно, чем больше метод генерирует пар, тем большего качества можно достигнуть, но тем медленнее будет работать алгоритм.

## 3 Обучение и контроль качества

Для каждой пары документов, полученной в результате фильтрации, известно, находятся ли формирующие ее документы в одной теме.

Считая пары документов, принадлежащих одной теме, положительными примерами, а пары документов, принадлежащих различным темам, отрицательными примерами, можно сформировать бинарную классификационную задачу.

Большие темы (как правило) будут порождать большее количество пар, поэтому выборка получится смещенной с точки зрения метрик, для которых все темы равнозначны. Для борьбы с этим мы используем взвешивание: пара документов получает вес, обратно пропорциональный размеру тем, в которые входят эти документы. Точнее, если $d_i \in t', d_j \in t''$, то пара $\langle d_i, d_j \rangle$ получает вес, равный $1/|t'| + 1/|t''|$. Такое взвешивание обладает понятным физическим смыслом. Рассмотрим ситуацию, в которой документ $d_i$ входит в кластер, содержащий все документы темы $t'$ и имеющий стопроцентную точность. Тогда добавление в этот кластер какого-либо одного нерелевантного документа уменьшает точность кластера *примерно* на $1/|t'|$, а удаление из этого кластера любого релевантного документа уменьшает полноту кластера на $1/|t'|$. Возможно, использование более точной оценки веса для отрицательных примеров имеет некоторый смысл (выяснить это на практике весьма затруднительно).

Если с порожденными парами неразмеченных документов все ясно (их не надо включать в задачу), то с парами, в которых размечен ровно один документ, не все очевидно. Ведь на самом деле в этом случае значение целевой функции неизвестно: неразмеченный документ может оказаться как релевантным, так и нерелевантным. Впрочем, на практике оказывается, что, как

правило, неразмеченные документы нерелевантны: разметка составляется достаточно качественно, и имеющиеся в ней темы покрыты разметкой достаточно полно. Поэтому такие пары мы при обучении относим в отрицательный класс.

Использовать в такой задаче для контроля процедуру обыкновенного скользящего контроля, очевидно, некорректно. Для обеспечения большей корректности используется следующая процедура.

Множество тем разбивается на подмножества: $T = T_1 \sqcup T_2 \sqcup \ldots \sqcup T_k$. Затем для $i \in \{1, \ldots, k\}$ множество пар документов для обучения формируется с использованием в качестве разметки тем из объединения

$$L_i = \sum_{\substack{1 \le j \le k \\ j \ne i}} T_i,$$

а метрики вычисляются с использованием в качестве разметки тем из $T_i$. Вопрос, соответственно, стоит теперь лишь в том, каким образом осуществлять разбиение $T$ на подмножества.

Оказывается, что для задачи кластеризации хорошим способом является разбиение со стратификацией по размеру темы. Связано это с тем, что тем большого размера устроены весьма специфическим образом, и при этом их достаточно небольшое количество. Поэтому при случайных разбиениях легко получить ситуацию, в которой почти все большие темы принадлежат только обучению или только контролю, что приводит к большому разбросу метрик.

Конкретнее, стратификация осуществляется следующим образом. Упорядочим темы по размеру:

$$\left|t_{i_1}\right| \le \left|t_{i_2}\right| \le \cdots \le \left|t_{i_n}\right|,$$

причем, если несколько тем имеют одинаковый размер, перетасуем их случайным образом в этом списке. После этого нужное количество раз повторим следующие действия:

- случайным образом перетасуем первые $k$ тем из этого списка;
- тему с номером $i$ $(1 \le i \le k)$ в получившемся списке тем отнесем к множеству $T_i$;
- удалим первые $k$ из списка всех тем.

При использовании такой схемы можно ожидать, что оценка скользящего контроля окажется адекватной оценкой для реального качества кластеризации в предположении, что в разметке распределение размеров тем близко к реальному распределению.

Впрочем, возможно, это не наилучший способ стратификации. Другой возможный вариант – стратификация тем по качеству фильтрации. Оказывается, что качество фильтрации достаточно сильно варьируется от подмножества к подмножеству, а качество кластеризации сильно коррелирует с качеством фильтрации и на обучающем, и на тестовом множествах.

Необходимо также отметить, что в силу того, что пары документов, содержащие ровно один размеченный документ, считаются отрицательными, возможно получение абсолютно идентичных строк в обучающем и тестовом пулах. Влияние этого эффекта на качество оценки при помощи скользящего контроля пока что остается под вопросом.