МИНИСТЕРСТВО ОБЩЕГО И ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ

Московский физико-технический институт
(государственный университет)

Алгоритм построения логических функций для кластеризации документов

Выпускная квалификационная работа на степень бакалавра

Студент 312 группы ФРТК
Костин Михаил Юрьевич

Научный руководитель
к.ф.-м.н, ассистент Ворожцов Артём Игоревич

МОСКВА, 2007 г.

Содержание

Аннотация

Данная работа посвящена разработке и применению нового подхода к задаче автоматической кластеризации документов. Разработан алгоритм, который для набора кластеров (групп) документов составляет логические выражения, которые позволяют отличать кластеры друг от друга. На входе алгоритма набор документов (либо откластеризованный, либо нет), а на выходе - откластеризованный набор документов и логическое выражение для каждого набора, которое отличает его от остальных.

1 Введение

В задаче *контролируемого обучения*, нам даётся тренировочный набор классифицированных векторов *признаков* фиксированной длины, по которому мы строим модель классификации (в нашем случае, у нас просто есть набор отклассифицированных документов). Эта модель, в свою очередь, используется для определения класса элементов из нового набора документов. Таким образом, в построении модели классификации, крайне важна информация о классе характерном для документов с данной особенностью. Хотя, с теоретической точки зрения, большее количество документов должно способствовать более точному определению классов, но на самом деле есть много причин почему это может не иметь места.

Большинство методов обучения подвержено *проклятию размерности (curse of dimensionality)*. То есть с увеличением количества документов в тренировочном наборе, время, требуемое для выполнения алгоритма значительно растёт, иногда даже эскпоненциально. Поэтому, когда начальный набор документов достаточно велик, большинство из алгоритмов обучения просто неприемлемы. Более того, некоторые *признаки* в тренировочном наборе могут быть излишни или несвойственны другим признакам. То есть, подобные признаки не несут никакой полезной информации, а просто увеличивают время обработки.

Большинство алгоритмов обучения могут рассматриваться как алгоритмы оценивающие вероятность того, что данный набор признаков характерен для данного класса. В областях с большИм количеством признаков получится очень сложное многомерное распределение. К сожалению, в реальном мире, мы часто сталкиваемся с проблемой огра-

ниченности данных, по которым строится модель классификации. В связи с этим, очень трудно получить оценки некоторых вероятностных параметров. Чтобы избежать *переполнения (overfitting)* модели, используется принцип *бритвы Оккама (Occam's Razor)*:

> При прочих равных, следует выбирать теорию, которая пользуется наименьшим количеством предположений.

То есть строится как можно более простая модель, которая, тем не менее, даёт приемлимые результаты на тренировочном наборе. Следуя этому же принципу, часто вместо полного набора признаков предпочитают небольшой набор, который будучи взятым в соответствующей сложной комбинации будет достаточно точно предсказывать класс документа. Несвойственные или излишниие признаки также создают трудности, т.к. они могут « запутать» алгоритм обучения, « затуманив» набор действительно важных признаков.

В свете данных рассуждений, большое количество исследователей в последнее время относят задачу выделения подмножества признаков к *машинному обучению*. Как полагают John, Kohavi и Pfleger, эту работу часто можно разделить на две основополагающие: *предварительной фильтрации* и *последующей обработки*.

В модели предварительной фильтрации выделение признаков предшествует шагу построения модели кластеризации. Таким образом, алгоритм обучения не пересекается с алгоритмом выделения признаков. Два наиболее известных метода фильтрования для выделения признаков: RELIEF (Kira & Rendell, 1992) и FOCUS (Almuallim & Dietterich, 1991). В RELIEF, подмножество признаков не определяется непосредственно, а каждому признаку даётся коэффициент релевантности, ко-

торый показывает насколько характерен данный признак для данного класса. Важно отметить, что этот метод неэффективен для удаления излишних признаков, так как два коррелирующих признака скорее всего получат высокие весовые коэффициенты релевантносты. Алгоритм FOCUS руководствуется полным поиском всех подмножеств признаков, чтобы определить минимальный набор признаков, который может обеспечить соответствующее разбиение на классы документы тренировочного набора. Из-за этого критерия *соответствия* FOCUS очень чувствителен к шумам и несоответсвиям в обучающем наборе документов. Более того, этот алгоритм неприемлем для областей с количеством признаков более 25−30 из-за экспоненциального роста мощности множества подмножеств признаков.

Другой способ выделения признаков, которому в последнее время уделяют гораздо больше внимания, метод последующей обработки (John et al. 1994) (Caruana & Freitag, 1994) (Langley & Sage, 1994). Эта модель применяет поиск по множеству подмножеств признаков, используя оценочную точность алгоритма обучения в качестве коэффициента полезности данного подмножества. Таким образом, при выделении признаков используется алгоритм обучения, то есть действия при поиске и в алгоритме обучения тесно взаимосвязаны. Хотя этот способ успешен в задачах обучения, он черезмерно дорог для выполнения и может не сработать в случаях когда дано большое количество признаков. К тому же, этот метод оставляет желать лучшего с точки зрения теоретического обоснования. Хотя важным аспектом выделения признаков является то, насколько хорошо данный метод помогает алгоритму обучения, не менее важно понимать как в целом влияет выделение признаков на алгоритм обучения.

2 Теоретические основы

Пример входных данных представляет собой *набор связей* между набором значений $\mathbf{f} = (f_1, \ldots, f_n)$ и набором признаков $\mathbf{F} = (F_1, \ldots, F_n)$. Иными словами, у нас есть двудольный граф из документов с одной стороны и их признаков (слов, комбинаций слов) с другой. Как обычно, мы полагаем, что все входные данные (в том числе и обучающий набор) получены из некоторого вероятностного распределения в пространстве векторов признаков. Формально, для каждого набора связей значений $\mathbf{f}$ и $\mathbf{F}$, у нас есть вероятность $Pr(\mathbf{F} = \mathbf{f})$.

Прежде всего, стоит обратить внимание, на различие таких понятий как кластеризация и классификация. Классификация - метод, который получает на вход единицу входных данных и классифицирует её как принадлежающую к одному из возможных классов $c_1, \ldots, c_l$. Классификатор должен принимать решение на основе связей между $\mathbf{f}$ и единицей входных данных. В лучшем случае, вектор признаков полностью определяет подходящую классификацию. Однако, такое редко случается: как правило, у нас нет доступа к достаточному количеству признаков, чтобы принять однозначное решение. Следовательно, мы используем вероятностное распределение в модели классифицирующей функции. Более детально, для каждого набора связей между $\mathbf{f}$ и $\mathbf{F}$, у нас есть распределение $Pr(C|\mathbf{F} = \mathbf{f})$ по различным возмножным классам, C. При кластеризации мы разбиваем множество документов на классы (подмножества), которые заранее не известны.

Алгоритм обучения неявно использует приближённую версию условного распределения $Pr(C|\mathbf{F})$ (эмпирические частоты наблюдаемые на обучающем наборе), чтобы построить классификатор для данной зада-

чи. Хорошо известно, что количество признаков непосредственно влияет на производительность алгоритма обучения. С одной стороны, существование несвойственных или излишних признаков может привести к тому, что обучающий алгоритм будет использовать менее оптимальный набор признаков, чем ухудшит точность обучающего алгоритма. С другой стороны, расчётная сложность многих алгоритмов обучения сильно зависит от количества признаков. Поэтому, часто бывает полезно уменьшить количество признаков, прежде чем начинать строить классификатор.

Давайте рассмотрим как влияет уменьшение количества признаков на распределение, которое характеризует данную задачу. Пусть $\mathbf{G}$ это некоторое подмножество $\mathbf{F}$. Например, $\mathbf{F}$ может состоять из пары признаков (A, B), а $\mathbf{G}$ может состоять из одного признака A. Если у нас есть вектор признаков $\mathbf{f}$, то через $\mathbf{f_G}$ мы будем обозначать проекцию $\mathbf{f}$ на значения $\mathbf{G}$. Например, для вектора признаков $\mathbf{f} = (a, b)$, $\mathbf{f_{(A)}} = (\mathbf{a})$. Теперь, рассмотрим конкретные данных задаваемые $\mathbf{f}$. В первоначальном распределении, эти данные задавали распределение $Pr(C|\mathbf{F} = \mathbf{f})$. В уменьшенном пространстве признаков, те же самые данные задают (возможно отличающееся) распределение $Pr(C|\mathbf{G} = \mathbf{f_G})$. Нашей целью будет выбрать $\mathbf{G}$ таким образом, чтобы эти два распределения были как можно более похожи. В качестве метрики, мы будем использовать информационно-теоретичское измерение взаимной энтропии (также известной как *KL-расстояние* (Kullback & Leibler, 1951)). Таким образом, мы можем рассматривать данную задачу, как задачу выбора подмножества $\mathbf{G}$, которое приведёт к наименьшей потери информации о начальном распределении.

Пусть μ и σ два распределения в некотором вероятностном про-

странстве Ω. Взаимная энтропия μ и σ определяется как:

$$D(\mu, \sigma) = \sum_{x \in \Omega} \mu(x) log \frac{\mu(x)}{\sigma(x)}$$

Заметьте, что μ и σ входят в это определение не симметричным образом. Условно говоря, смысл в том, что μ - « правильное» распределение, а σ его приближение. Тогда $D(\mu, \sigma)$ будет как раз мерой расстояния или « ошибки», которую мы сделали, когда заменили распределение μ на σ. Таким образом, взаимная энтропия вполне применима в данной ситуации, с $Pr(C|\mathbf{f})$ в роли « правильного» распределения μ, а $Pr(C|\mathbf{f_G})$ в роли σ. В этом случае вероятностное пространство Ω - набор возможных кластеров $\{c_1, \ldots, c_n\}$. Таким образом, мы определили:

$$\delta_{\mathbf{G}}(f) = D(Pr(C|\mathbf{f}), Pr(C|\mathbf{f_G}))$$

Конечно, чтобы сравнивать одно подмножество признаков $\mathbf{G}$ с другим, мы должны связать значения $\delta_{\mathbf{G}}(f)$ для разных векторов признаков $\mathbf{f}$ в одну характеристику. Проще всего, мы могли бы просто сложить взаимные энтропии для разных векторов признаков, или брать наибольшую взаимную энтропию из всех. Ни одна из этих идей не берёт во внимание тот факт, что некторые вектора признаков имеют место гораздо чаще, чем другие, и такой подход может привести к большим ошибкам в определённой доле случаев. Поэтому, мы хотим найти подмножество признаков $\mathbf{G}$, для которого $\Delta_{\mathbf{G}} = \sum_{\mathbf{f}} Pr(\mathbf{f}) \delta_{\mathbf{G}}(\mathbf{f})$ достаточно мала.

Поясним, множество признаков, которое минимизирует эту величину, это просто $\mathbf{F}$, так как оно задаёт именно первоначальное распределение. На каждом шаге нашего алгоритма мы исключаем признак F_i таким образом, чтобы полученное распределение оставалось как можно

ближе к первоначальному. Мы просто используем *жадный* алгоритм: исключаем тот признак F_i, который приведёт к наименьшему приросту Δ. Иными словами, у нас есть текущий набор признаков $\mathbf{G}$, который в самом начале равен $\mathbf{F}$. На каждом шаге мы хотим исключить признак F_i так, чтобы $\Delta_{(\mathbf{G}-\{F_i\})}$ была как можно ближе к $\Delta_{\mathbf{G}}$.

К сожалению, невозможно просто применить эту идею так, как она была описана, так как вычисление $\Delta_{\mathbf{G}}$ занимает экспоненциальное время по количеству признаков. Более того, мы, на самом деле, не можем сравнить наше приближённое распределение с истинным условным распределением $Pr(C|\mathbf{F})$, так как у нас просто нет точного распределения. Вместо этого, у нас есть обучающий набор, который является лишь его грубым приближением. В тех случаях, когда мы располагаем большим количеством признаков, количество данных в обучающем наборе, соответствующих некоторому конкретному $\mathbf{f}$, будет очень мало. Поэтому, с ростом количества признаков, наша способность использовать обучающий набор для приближения условного распределения уменьшается экспоненциально.

Как мы сейчас покажем, мы можем использовать вероятностные рассуждения, чтобы обойти эту проблему, в некоторой степени. Признаки, которые приводят к небольшому увеличению Δ, те, которые дают нам наименьший прирост к информации, которую мы получили бы от всех остальных признаков в $\mathbf{G}$. Мы можем обратить это в формальную идею *условной независимости*.

Определение 1 *Две переменные называются* условно независимыми*, если для данного набора значений* $\mathbf{X}$*, для любых* a*,* b *и* $\mathbf{x}$ *из наборов* A*,* B *и* $\mathbf{X}$*, соответственно,* $Pr(A = a|\mathbf{X} = \mathbf{x}, B = b) = Pr(A = a|\mathbf{X} = \mathbf{x})$*. То есть,* B *не даёт никакой информации об* A*, кроме той, что*

уже содержится в $\mathbf{X}$.

Теперь легко показать, что:

Утверждение 1 *Пусть* $\mathbf{G}$ - *подмножество признаков и* $\mathbf{F_i}$ - *признак из* $\mathbf{G}$. *Тогда* $\mathbf{F_i}$ *условно независимо от* $\mathbf{C} \in \mathbf{G'} = \mathbf{G} - \{F_i\}$ *тогда и только тогда, когда* $\Delta_{\mathbf{G'}} = \Delta_{\mathbf{G}}$.

Таким образом, мы можем убирать условно независимый признак F_i из $\mathbf{G}$, не увеличивая при этом расстояние от желаемого распределения. Также очевидно, что убирая признак, который « почти» условно независим, мы не сильно отдалимся.

Хотя проверять каждый признак из $\mathbf{G}$ на условную независимость со всеми остальными не представляется возможным, данная переформулировка приводит к одному из способов решения задачи. Понятно, что если вся информация от признака F_i содержится в $\mathbf{G'}$, то она скорее всего содержится в каком-то подмножестве $\mathbf{G'}$. В конце концов, очень маловероятно, что все эти признаки (обычно их очень много) будут необходимы.

Определение 2 *Пусть* $\mathbf{M}$ - *некоторое множество признаков, не содержащее* F_i. *Будем говорить, что* $\mathbf{M}$ *есть* марковское ограничение *для* F_i, *если* F_i *условно независимо от* $\mathbf{F} - \mathbf{M} - \{F_i\}$.

Легко видеть, что если $\mathbf{M}$ является марковским ограничением F_i, то также имеет место тот факт, что класс C условно независим от признака F_i. Таким образом:

Следствие 1 *Пусть* $\mathbf{G}$ - *подмножество признаков и* $F_i \in \mathbf{G}$. *Предположим, что некоторое подмножество* $\mathbf{M} \subset \mathbf{G}$ *есть марковское ограничение* F_i, *тогда* $\Delta_{\mathbf{G'}} = \Delta_{\mathbf{G}}$.

Утверждение с марковским ограничением сильнее, чем с условной независимостью. Оно требует, чтобы $\mathbf{M}$ содержало не только информацию, которая есть в F_i о , но также информацию о всех остальных признаках. Хотя найти такое подмножество $\mathbf{M}$ - очень трудная задача, метод использования марковских ограничений как основы для устранения признаков обладает многими полезными свойствами.

Проще говоря, мы хотим убрать те признаки, для которых мы нашли марковское ограничение как подмножество всех остальных признаков. Покажем, что признаки, которые мы сочли ненужными, основываясь на этом критерии, остаются ненужными в течение всего последующего процесса. Предположим, к примеру, что мы убрали признак F_i, основываясь на марковском ограничении $\mathbf{M}$. На каком-то следующем этапе, мы могли бы убрать какой-нибудь из признаков $F_j \in \mathbf{M}$. В общем случае, устранение признака F_j может привести к тому, что признак F_i уже не будет являться ненужным. Т.е. если бы мы вернули признак F_i, мы не могли бы убрать его. Как мы сейчас покажем, такой случай не имеет места.

Теорема 1 *Пусть* $\mathbf{G}$ *наш текущий набор признаков. Предположим, что у некоторого (ранее удалённого) признака* $F_i \notin \mathbf{G}$ *есть марковское ограничение* $\mathbf{M} \subset \mathbf{G}$*. Пусть* $F_j \in \mathbf{G}$ *- некоторый признак, который мы собираемся удалить, основываясь на критерии марковских ограничений. Тогда у* F_i *есть марковское ограничение в оставшемся множестве* $\mathbf{G} - \{F_j\}$*.*

Доказательство: Будем обозначать условную независимость двух переменных или наборов переменных X и Y при заданном наборе Z как $I(X, Y|Z)$. Пусть $\mathbf{M_i} \subseteq \mathbf{G}$ - марковское ограничение F_i (причём $\mathbf{M_i}$ не

обязательно то самое марковское ограничение, которое мы использовали, чтобы удалить F_i в первый раз); аналогично определим $\mathbf{M_j} \subseteq \mathbf{G}$. Очевидно, что если $\mathbf{M_i}$ не содержит F_j, то оно так и останется марковским ограничением для F_i даже после удаления F_j из $\mathbf{G}$. Получаем, что $F_j \in \mathbf{M_i}$, определим $\mathbf{M_i} = \mathbf{M'_i} \cup \{F_j\}$. Покажем, что $\mathbf{M'_i} \cup \mathbf{M_j}$ - марковское ограничение F_i. Обозначим $\mathbf{G} - \{F_j\} - (\mathbf{M'_i} \cup \mathbf{M_j})$ за X. Необходимо показать $I(F_i, X|(\mathbf{M_i} \cup \mathbf{M_j}))$. Из предположения о марковском ограничении F_j и свойства разложения, получаем $I(F_j, (X \cup \mathbf{M'_i})|\mathbf{M_j})$. Из этого следует $I(F_j, X|\mathbf{M'_i} \cup \mathbf{M_j})$. Аналогично, можем вывести $I(F_i, (X \cup (\mathbf{M_j} - \mathbf{M'_i}))|\mathbf{M'_i} \cup \{F_j\})$, и, таким образом, $I(F_i, X|\mathbf{M'_i} \cup \mathbf{M_j} \cup \{F_j\})$. Из последних двух следует искомый результат.

Таким образом, мы получили, что критерий марковских ограничений удаляет только действительно ненужные признаки. Есть два типа признаков, которые обычно являются ненужными: признаки, которые совсем не свойственны, признаки, которые излишни, - критерий марковских ограничений « борется» с обоими. Признаки, которые не свойственны, будут просто независимы от всех остальных. Более того, если у нас есть набор признаков, которые коррелируют только друг с другом, то мы их все уберём по очереди. С другой стороны, если у нас есть признак, значение которого однозначно (или просто с очень высокой вероятностью) определяется каким-либо набором $\mathbf{M}$ из оставшегося количества признаков, то мы можем убрать этот признак, используя $\mathbf{M}$ в качестве марковского ограничения. Очень немногие методы выбора признаков могут разбираться с обоими случаями.

Интересен также другой подход к задаче выбора признаков. Вместо того, чтобы убирать ненужные признаки, мы начнём с пустого набора признаков и будем добавлять их по одному. Для этого будет необхо-

димо понятние *прироста информации (information gain)*: к уже имеющемуся набору $\mathbf{G}$ мы добавляем признак F_j, который максимизирует взаимную энтропию между $Pr(C|\mathbf{G})$ и $Pr(C|\mathbf{G} \cup \{F_j\})$. Легко показать, что метод марковских ограничений применим и в этом случае. Можно подумать, что это просто два разных варианта одного и того же метода, но это не так. Смысл в следующем. Наша цель, как мы говорили, слегка изменить начальное распределение, убрав несколько признаков, которые не сильно влияют на него. В другом случае, мы начинаем с пустого набора признаков и нулевого распределения, а затем делаем « большие» шаги от него. Но вполне возможен такой случай: признак F_j даёт больший « прыжок» от текущего распределения, чем F_i, но при этом распределение $Pr(C|F_i)$ может быть ближе к исходному, чем $Pr(C|F_j)$. Этот случай имеет место быть для некоторых входных данных.

3 Типы кластеризации

Алгоритмы кластеризации можно разделить по нескольким признакам.

1) иерархические и неирерархические

2) масштабируемые и немасштабируемые

Основным отличием иерархических методов является то, что они разбивают документы постепенно, строя при этом некую иерархию групп. Неирархические алгоритмы при кластеризации руководствуются некоторой целевой функцией, значение которой пытаются минимизировать. Наиболее популярные неиерархические алгоритмы: k-кластеризация и EM-алгоритм (Excpectation-Maximization).

3.1 k-кластеризация

В данном случае кластеры представляют из себя « сферу» документов некоторого радиуса. На вход алгоритма поступает k и множество всех документов D, а на выходе получаем документы разбитые на кластеры-« сферы» $D_1, D_2, \ldots, D_k$. Данный алгоритм предполагает, что элементы D принадлежат множеству $\mathbb{R}^d$, и для каждого кластера определяется оценочная функция $f : \{X : X \subset D\} \to R^+$. Далее мы просто пытаемся минимизировать сумму значений этой функции, то есть $\sum_{i=1}^{k} f(D_i)$. Приведём несколько способов выбора функции f:

- *Метод наименьших квадратов.* Пусть x_r^i - r-тый элемент D_i. Тогда:

  $$f(D_i) = \sum_{1 \le r,s \le |D_i|} d(x_r^i, x_s^i)$$

  где $d(x, y)$ - расстояние между точками в пространстве $\mathbb{R}^d$, а $|D_i|$ - количество элементов в D_i.

- *Алгоритм МакКвина.* Вместо суммирования всех попарных расстояний будем суммировать только расстояния от всех элементов до « центра масс» x^i кластера:

  $$f(D_i) = \sum_{1 \le r \le |D_i|} d(x^i, x_r^i)$$

- Ещё один метод основан на том, что вместо того, чтобы минимизировать $\sum_{i=1}^{k} f_1(D_i)$ можно минимизировать $max_{1 \le i \le k} f_2(D_i)$, где $f_2(D_i) = max_{1 \le r,s \le |D_i|} d(x_r^i, x_s^i)$.

3.2 Expectation-Maximization алгоритм

В данном алгоритме вместо « центров» кластеров предполагается наличие некоторой функции плотности вероятности для каждого кластера с определённым значением математического ожидания и дисперсии. Разбиение на кластеры идёт путём поиска этих значений по принципу максимального правдоподобия. Проблема возникает тогда, когда кластеризуемые элементы имеют аттрибуты из разных категорий (от этой проблемы не уйти и в k-кластеризации). Например, когда часть аттрибутов числовая, а часть - категорийная. В данном случае задачу кластеризации могут помочь решить иерархические алгоритмы кластеризации.

3.3 Иерархическая кластеризация

Есть три основных метода иерархической кластеризации данных: *Single Link, Complete Link, Group Average*. Принцип их работы примерно следующий (изначально каждый документ помещён в отдельный кластер):

0) построение матрицы близости

1) два наиболее близких элемента объединяются в один

2) *обновление матрицы близости* и переход на шаг №1, пока не будет достигнуто условие выхода

В качестве условия выхода обычно принимают максимальное значение элементов в кластере. Различие этих трёх методов состоит в пункте « обновление матрицы близости». По точности определения кластеров

более продуктивными являются Complete Link и Group Average. Скорось работы алгоритмов Single Link и Group Average - $O(n^2)$, Complete Link - $O(n^3)$.

Алгоритмы неиерархической кластеризации, основанные на оптимизации значения целевой функции, имеют итеративный характер и требуют пересчёта матрицы близости на каждом шаге. При достаточно большом количестве элементов это бывает неэффективно. Имеется также недостаток в самой идее поиска « сферических» кластеров. Этот подход даёт хорошие результаты, когда элементы в пространстве образуют некоторые « кучки» на достаточном расстоянии друг от друга. Но, например, случай, когда одна группа элементов находится внутри другой, такой метод даст заведомо неверный результат.

3.4 Алгоритм PLSA

Алгоритм, который стал следующим шагом после LSA (Latent Semantic Analysis). Однако, в отличие от LSA, в методе PLSA используется статистическая модель, которая даёт намного более лучшие результаты.

3.5 Алгоритм Bootstrapping

Алгоритм Bootstrapping или алгоритм «вытягивания шнурков», назван так потому, что в процессе его работы, мы кластеризуем документы на основе небольшого набора откластеризованных документов, который у нас уже есть, а после этого откластеризованные документы добавляем к нашему набору. Более подробно. У нас есть множество откластеризованных документов MasterSet. На основе этого набора мы можем построить несколько кластеров. Каждый приходящий новый до-

кумент имеет некоторый вектор распределения по этим кластерам, и если в этом векторе есть ровно одна достаточно большая величина, то мы помечаем этот документ как «хороший» и добавляем его к набору MasterSet. Эту процедуру проделываем, пока не откластеризуем все документы.

4 Feature Extraction

Теперь рассмотрим ещё один подход к нашей задаче кластеризации. Допустим у нас уже есть несколько откластеризованных документов. После того как построены некоторые кластеры, можно попытаться выявить уникальные особенности каждого кластера, которые отличают его от других. Во-первых, для каждого набора слов мы можем определить ключевые документы для этих слов. Делается это примерно так. Пусть у нас есть некоторый набор слов, для которых мы хотим определить множество ключевых документов для них. Для каждого слова введём величину « информативности» этого слова, которая будет определяться либо методом Information Gain, либо Information Gain Ratio. Упорядочим все слова по убыванию информативности, сгладим полученную зависимость и отбросим слова идущие после точки перегиба. Для полученного набора слов возьмём документы, которые содержат хотя бы одно из наших слов, и тоже упорядочим их по значимости, значимость каждого документа определяется как сумма значимостей слов из нашего набора, входящих в него. Аналогично « отрежем» малозначимые документы. Это и будет искомый набор ключевых документов для данного набора слов. Аналогично для любого набора документов мы можем построить набор ключевых слов.

Теперь перейдём непосредственно к задаче построения выражений. Делается это достаточно просто. Для каждого кластера мы определяем основные значимые слова в его документах, далее для этих слов мы рассматриваем две величины. Первая - то, насколько сильно они хотят « объединиться» союзом "и", а вторая - то, насколько сильно они хотят « объединиться» союзом "или". Первая величина определяется количеством документов, в которые входят оба эти слова, а вторая величина определяется несколько сложнее. Если какое-то одно из слов содержалось в наборе документов, а второе нет, но после пинг-понга, стало содержаться, то это может означать, что данные слова, например, являются синонимами и их стоит объединить союзом или.

После того как мы построили две матрицы, которые характеризуют, степень « желания объедениться». Выбираем из них наиболее « ярых» представителей и объединяем их союзом "и" или "или". После этого мы получили уже новый набор признаков, которые есть у документов (раньше это были просто слова). Теперь можно повторить заново эту процедуру, но уже со « сложными словами». В результате мы получим несколько сложных логических выражений, которые можно будет использовать как определяющие выражения для кластеров, то есть как их уникальные признаки, которые отличают кластеры друг от друга.

Основным отличием представленного в данной работе алгоритма от всех остальных является то, что для его работы не требуется квадратичное время. Если алгоритм использует матрицу близости, то это сразу означает, что время работы данного алгоритма - как минимум $O(n^2)$. В текущем алгоритме матрицы близости не строится, что существенно сокращает время его работы. Вторым отличием данного алгоритма является двойственность работы с объектами, то есть для

нас нет разницы «слова-документы», и все результаты полученные для слов могут также применяться и для документов.

Точность алгоритма оценивается следующим образом. Пусть у нас есть два класса документов A и $\bar{A}$. Запускаем алгоритм и даём ему на вход эти документы, какие-то документы он определит правильно, какие-то нет. По доле правильно определённых документов определяется точность данного алгоритма. Почти все имеющиеся на сегодняшний день алгоритмы (в том числе и представленный в данной работе) имеют точность $\sim 90\%$. Однако у нашего алгоритма есть несколько преимуществ (как уже было описано выше), это, во-первых, время работы алгоритма, во-вторых, когда у нас построены логические выражения для каждого кластера, то мы можем кластеризовать все приходящие документы «на лету».

5 Заключение

Задача кластеризации на сегодняшний день является одной из наиболее важных, в связи с большим ростом потребности к организации, классификации и индексированию информации. Была реализована программа, действующая по представленному алгоритму. Программа позволяет кластеризовать документы, находить ключевые слова для набора документов, находить ключевые документы для набора слов. Последние две задачи также имеют значительную область применения и без задачи кластеризации (например, поиск релевантной информации по нескольким заданным терминам). Результаты работы данной программы свидетельствуют о целесообразности развития данного алгоритма. Данную работу можно развивать как в направлении повышения удобности использования программы, так и в направлении улучшения математических моделей, которые лежат в основе алгоритма.

6 Список литературы

1. Daphne Koller, Mehran Sahami “Toward Optimal Feature Selection”, International Conference on Machine Learning, 1-6 (1996)

2. Li Li, Wu Chou “Improving Latent Semantic Indexing Based Classifier With Information Gain”, Proc. of the 7th International Conference on Spoken Language Processing (2002)

3. John G. Cleary, W. J. Teahan “Unbounded Length Contexts for PPM”, The Computer Journal (1997)

4. W. J. Teahan “Text classification and segmentation using minimum cross-entropy”, Proceeding of RIAO-00, 6th International Conference “Recherche d’Information Assistee par Ordinateur”, (2000)

5. Thomas Hofmann “Probabilistic Latent Semantic Analysis”, Proc. of Uncertainty in Artificial Intelligence, (1999)