Кластеризация документов

Лидия Михайловна Пивоварова

Системы понимания текста

Введение

- Кластеризация документов – это процесс обнаружения естественных групп в коллекции документов.
- Кластеризацию может служить как чисто исследовательской цели (выяснить структуру коллекции), так и лучшему поиску и представлению информации; классический пример: группировка по темам в системах автоматического сбора новостей.
- Кластеризация: мягкая/жесткая, иеррархическая/плоская

Содержание

1. Оценка качества кластеризации
2. Применение векторной модели в кластеризации
3. Иерархическая кластеризация
4. «Разделяющая» кластеризация
5. Генеративные алгоритмы
6. Спектральная кластеризация
7. Снижение размерности
8. Модели с учетом порядка слов

Оценка качества кластеризации

- Не существует единого (общепризнанного, применимого во всех случаях) метода оценки
- Оценка предполагает, что коллекция (или часть коллекции) размечена человеком
  - *Кластеры* – результат кластеризации, *классы* – результат ручной разметки
- Аналогичные методы могут использоваться для оценки классификации

Матрица несоответствий

	КЛАССЫ			
КЛАСТЕРЫ		A	B	C
	a	2	2	0
	b	2	2	0
	c	0	0	8

- способ примитивный, зато наглядный

Метрики заимствованные из информационного поиска

	Релевантные	Нерелевантные
Найденные	tp	fp
Ненайденные	fn	tn

Полнота (recall):
$R = tp / (tp+fn)$

Точность (presicion):
$P = tp / (tp+fn)$

F-мера:

$$F_\alpha = \frac{(1+\alpha)RP}{\alpha P + R}$$

Аккуратность (accuracy):
$A = (tp + tn) / (tp + tn + fp + fn)$

Применительно к кластеризации

$$F = \sum_i \frac{n_i}{n} \operatorname*{argmax}_j F(i,j)$$

i – классы, j – кластеры, n – общее число документов, n_i – число документов в классе i

Т.е. для каждого класса выбираем кластер, который ему больше соответствует (argmax), суммируем меры соответствия (*F*) для всех классов, при этом чем больше класс, тем больше его вес в общей сумме (n_i).

F-мера показывает общее качество кластеризации, но не показывает как устроены сами кластеры.

Чистота

$$Purity = \sum_j \frac{n_j}{n} \operatorname*{argmax}_i P(i,j)$$

i – классы, j – кластеры, n – общее число документов, n_j – число документов в кластере j, P(i,j) – доля документов из класса i в кластере j .

Т.е. берем долю доминирующего (argmax) класса в кластере (*P(i,j)*), и суммируем по всем кластерам, при этом чем больше кластер, тем больше его вес в сумме (n_j).

Чем выше значение чистоты, тем лучше. В идеальном случае P=1.

Энтропия

$$Entropy = -\frac{1}{\log k}\sum_j \frac{n_j}{n}\sum_i P(i,j)\log P(i,j)$$

i – классы, j – кластеры, n – общее число документов, n_j – число документов в кластере j, P(i,j) – доля документов из класса i в кластере j , k – число кластеров.

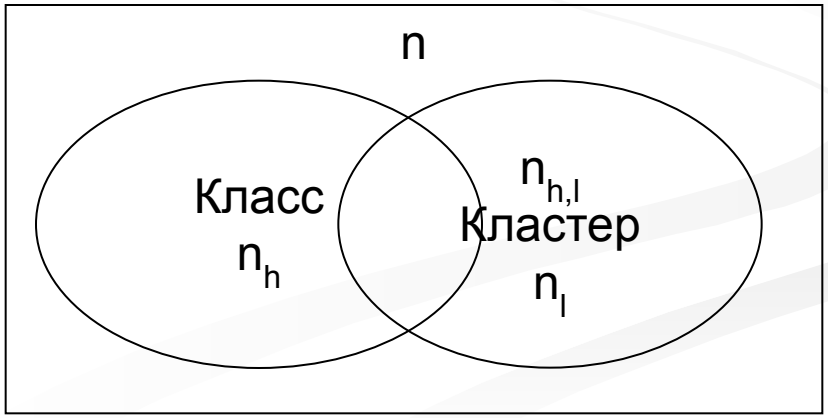
Энтропия – степень «размазанности» класса по кластерам. Чем меньше, тем лучше, в идеале E=0.

Взаимная информация

- Чистота и энтропия хороши тогда, когда число классов и кластеров совпадает. В других случаях лучше MI (или NMI – нормализованная взаимная информация).

$$NMI = \frac{\sum_{h,l} n_{h,l} \log \frac{n n_{h,l}}{n_h n_l}}{\sqrt{\left(\sum_h n_h \log \frac{n_h}{n}\right)\left(\sum_l n_l \log \frac{n_l}{n}\right)}}$$

n – общее число документов, n_h – число документов в классе h, n_l – число документов в кластере l, $n_{h,l}$ – число документов в пересечении.



Стабильность

С помощью взаимной информации можно считать стабильность, т.е. степень пересечения кластеризации при разных прогонах одного и того же алгоритма.

$$\varphi(\mathbf{\Lambda}, \hat{\lambda}) = \frac{1}{r} \sum_i NMI(\hat{\lambda}, \lambda_i)$$

Λ – множество различных кластеризаций, λ – конкретная кластеризация, r – число кластеров.

Содержание

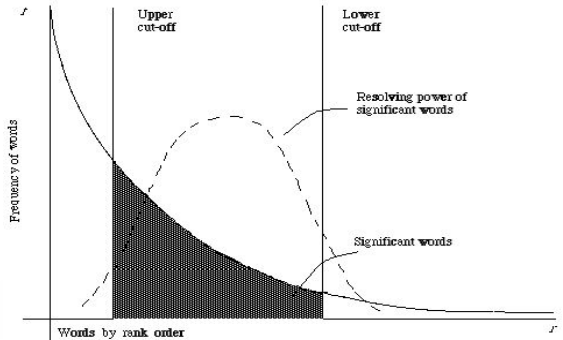
1. Оценка качества кластеризации
2. Применение векторной модели в кластеризации
3. Иерархическая кластеризация
4. «Разделяющая» кластеризация
5. Генеративные алгоритмы
6. Спектральная кластеризация
7. Снижение размерности
8. Модели с учетом порядка слов

Векторная модель

- Коллекция из n документов и m различных терминов представляется в виде матрицы mxn, где каждый документ – вектор в m-мерном пространстве.
- Веса терминов можно считать по разному: частота, бинарная частота (входит – не входит), tf*idf…
- Порядок слов не учитывается (bag of words)
- Матрица очень большая (большое число различных терминов в гетерогенной коллекции).
- В матрице много нулей

Предобработка

- Фильтрация (удаление спецсимволов и пунктуации)
- Токенизация (разбиваем текст на термины – слова или словосочетания)
- Стемминг (приведение слова к основе)
- Удаление стоп-слов
- Сокращение (удаление низкочастотных слов)



Содержание

1. Оценка качества кластеризации
2. Применение векторной модели в кластеризации
3. Иерархическая кластеризация
4. «Разделяющая» кластеризация
5. Генеративные алгоритмы
6. Спектральная кластеризация
7. Снижение размерности
8. Модели с учетом порядка слов

Иерархическая кластеризация

- На начальной стадии каждый документ – сам себе кластер.
- На каждом шаге документы объединяются до построения полного дерева.
- Число кластеров заранее не оговаривается.
- Не подходит для больших объемов данных (подсчет расстояния на каждой стадии).

Содержание

1. Оценка качества кластеризации
2. Применение векторной модели в кластеризации
3. Иерархическая кластеризация
4. «Разделяющая» кластеризация
5. Генеративные алгоритмы
6. Спектральная кластеризация
7. Снижение размерности
8. Модели с учетом порядка слов

«Разделяющая» кластеризация

- Классический пример - kmeans:
- Выбирается k случайных документов, которые считаются центроидами кластеров, все остальные документы распределяются по кластерам по степени близости к центроидам
- На следующих итерациях центроиды пересчитываются и документы перераспределяются
- Косинусная метрика лучше, чем Евклидово расстояние

$$\cos(\theta) = \frac{A \cdot B}{\|A\|\|B\|} = \frac{\sum_{i=1}^{n} A_i \times B_i}{\sqrt{\sum_{i=1}^{n} (A_i)^2} \times \sqrt{\sum_{i=1}^{n} (B_i)^2}}$$

Недостатки kmeans

- Результаты могут быть различными в зависимости от инициализации.
- Может останавливаться на субоптимальном локальном минимуме
- Чувствителен к шуму и случайным выбросам

- **Вычислительная сложность**: $\mathcal{O}(nkl)$

где n – число документов, k – число кластеров, l – число итераций.

Содержание

1. Оценка качества кластеризации
2. Применение векторной модели в кластеризации
3. Иерархическая кластеризация
4. «Разделяющая» кластеризация
5. Генеративные алгоритмы
6. Спектральная кластеризация
7. Снижение размерности
8. Модели с учетом порядка слов

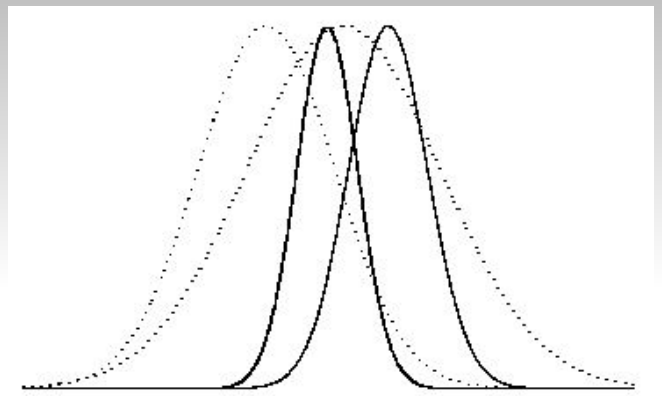
Генеративные алгоритмы

- Дискриминативные алгоритмы, которые основаны на попарной близости документов, имеют сложность $O(n^2)$ **по определению**.

- Генеративные алгоритмы не требуют такого сравнения, используя итеративные процедуры.

Гауссова модель

- Предполагается, что распределение документов в векторном пространстве – это набор Гауссовых распределений; каждый кластер ассоциирован со средним распределения и матрицей ковариации.



- Ковариация: $$\operatorname{cov}(X, Y) = \sum_{i=1}^{N} \frac{(x_i - \bar{x})(y_i - \bar{y})}{N}.$$
- Если между x и y нет корреляции, то ковариация равна нулю.
- Матрица ковариации: матрица, элементы которой – это попарные ковариации двух векторов.
- Если речь идет об одном и том же наборе векторов (наш случай: одни и те же документы в столбцах и строках), то матрица ковариации – это обобщение дисперсии для многомерной случайной величины.

Гауссова модель

- Вероятность того, что документ d принадлежит кластеру θ из набора Θ:

$$\mathcal{P}(\mathbf{d}|\theta) = \frac{1}{(2\pi)^{\frac{m}{2}}|\Sigma|^{\frac{1}{2}}} \exp\left(-\frac{(\mathbf{d}-\mu)^T\Sigma^{-1}(\mathbf{d}-\mu)}{2}\right)$$

P(d| θ) - вероятность того, что документ d принадлежит кластеру θ, m – размерность пространства, μ – центроид, Σ – матрица ковариации.

Общая вероятность (правдоподобие того, что данный документ описывается моделью):

$$\mathcal{P}(\mathbf{d}|\Theta) = \sum_{\theta\in\Theta} \mathcal{P}(\theta)\mathcal{P}(\mathbf{d}|\theta)$$

Задача кластеризации: м исло, максимизировав каждое из слагаемых (т.е. найдя наилучшее среднее и матрицу ковариации для каждого кластера).

Expectation maximization (EM-алгоритм)

- Итеративная процедура для нахождения максимального правдоподобия параметров модели.
- Две стадии:
  - E(xpectation) – вывод скрытых данных из наблюдаемых данных (документы) и текущей модели (кластеры)

$$\mathcal{P}(\theta|\mathbf{d}) = \frac{\mathcal{P}(\theta)\mathcal{P}(\mathbf{d}|\theta)}{\sum_{\theta\in\Theta}\mathcal{P}(\theta)\mathcal{P}(\mathbf{d}|\theta)}$$

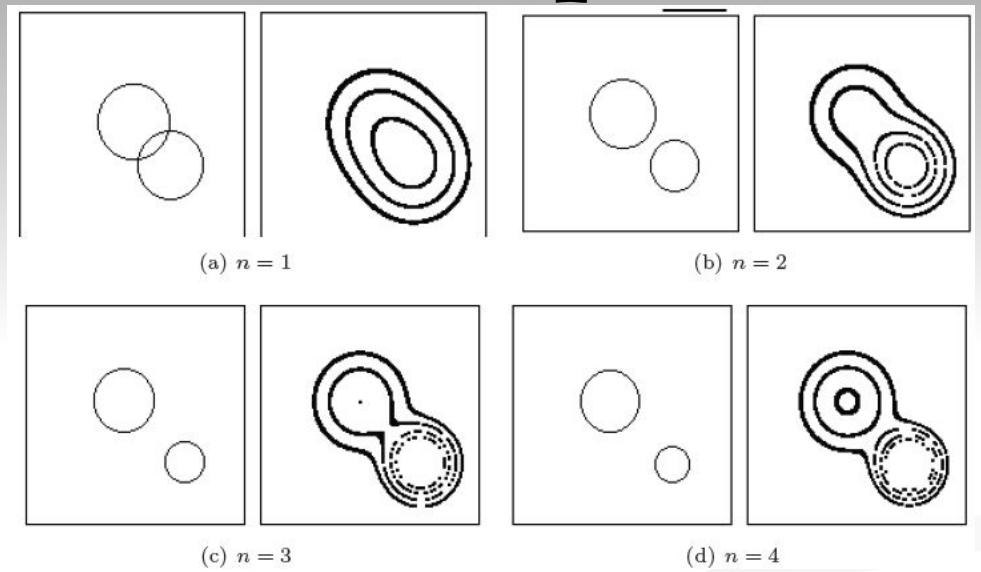
$$\mathcal{P}(\theta)^* = \sum_{d\in\mathcal{D}}\mathcal{P}(\theta|\mathbf{d}).$$

  - M(aximization) – максимизация правдоподобия в предположении, что скрытые данные известны

$$\mu = \frac{\sum_{\mathbf{d}\in\mathcal{D}}\mathcal{P}(\theta|\mathbf{d})\mathbf{d}}{\sum_{\mathbf{d}\in\mathcal{D}}\mathcal{P}(\theta|\mathbf{d})}$$

$$\Sigma = \frac{\sum_{\mathbf{d}\in\mathcal{D}}\mathcal{P}(\theta|\mathbf{d})(\mathbf{d}-\mu)(\mathbf{d}-\mu)^T}{\sum_{\mathbf{d}\in\mathcal{D}}\mathcal{P}(\theta|\mathbf{d})}$$

EM-алгоритм



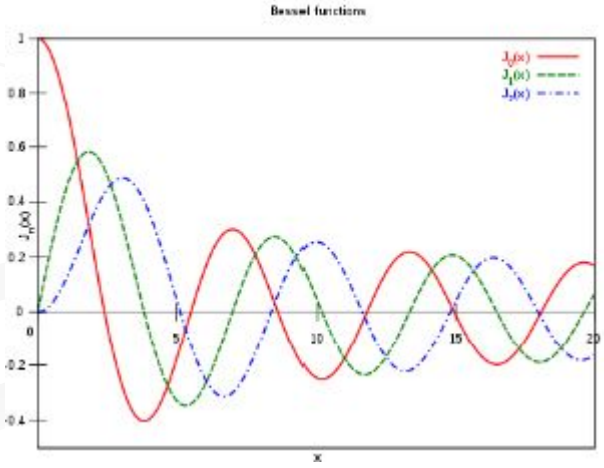
- Большое число свободных параметров может приводить к переобучению.
- Сокращение размерности: выбор дискриминирующих свойств для каждого кластера.
- Сложность: $O(k^2n)$
- Нестабильность, зависимость от инициализации.

Модель фон Мисес-Фишера

- На самом деле, распределение текстов по кластерам гауссианами описывается плохо. Было доказано, что лучше всего подходит vMF-распределение:

$$\mathcal{P}(\mathbf{d}|\theta) = \frac{1}{Z(\Sigma)} \exp\left(\Sigma \frac{\mathbf{d}^T \mu}{|\mu|}\right)$$

- Z – функция Бесселя (фактор нормализации). Затем используют алгоритм, похожий на em. Качество получается лучше, чем spherical k-means.

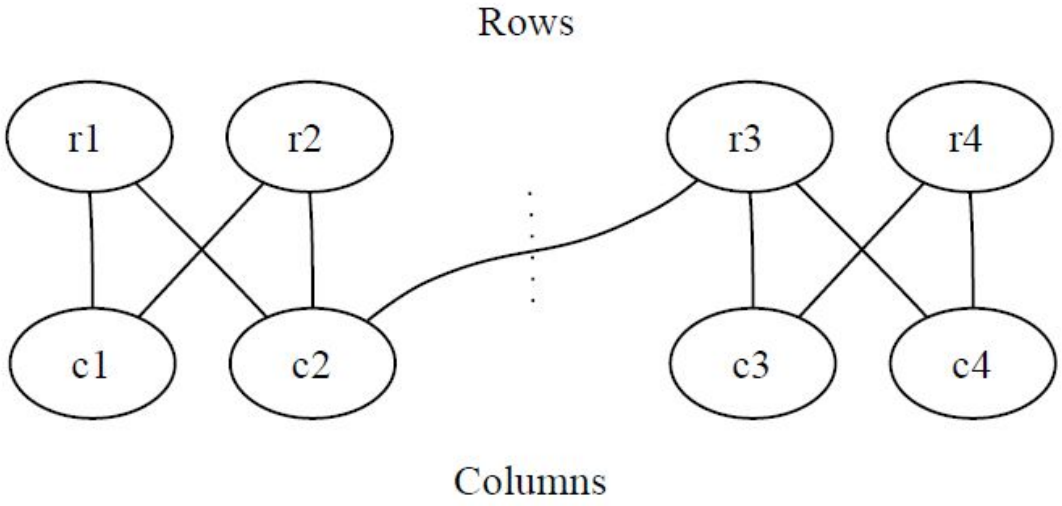


Содержание

1. Оценка качества кластеризации
2. Применение векторной модели в кластеризации
3. Иерархическая кластеризация
4. «Разделяющая» кластеризация
5. Генеративные алгоритмы
6. Спектральная кластеризация
7. Снижение размерности
8. Модели с учетом порядка слов

Спектральная кластеризация

- Основная гипотеза: термины, которые часто встречаются вместе, описывают близкие понятия. Поэтому важна группировка не только кластеров, но и терминов. Т.е. речь идет о **совместной кластеризации** терминов и документов.
- Матрица термин-документ преобразуется в двудольный граф:



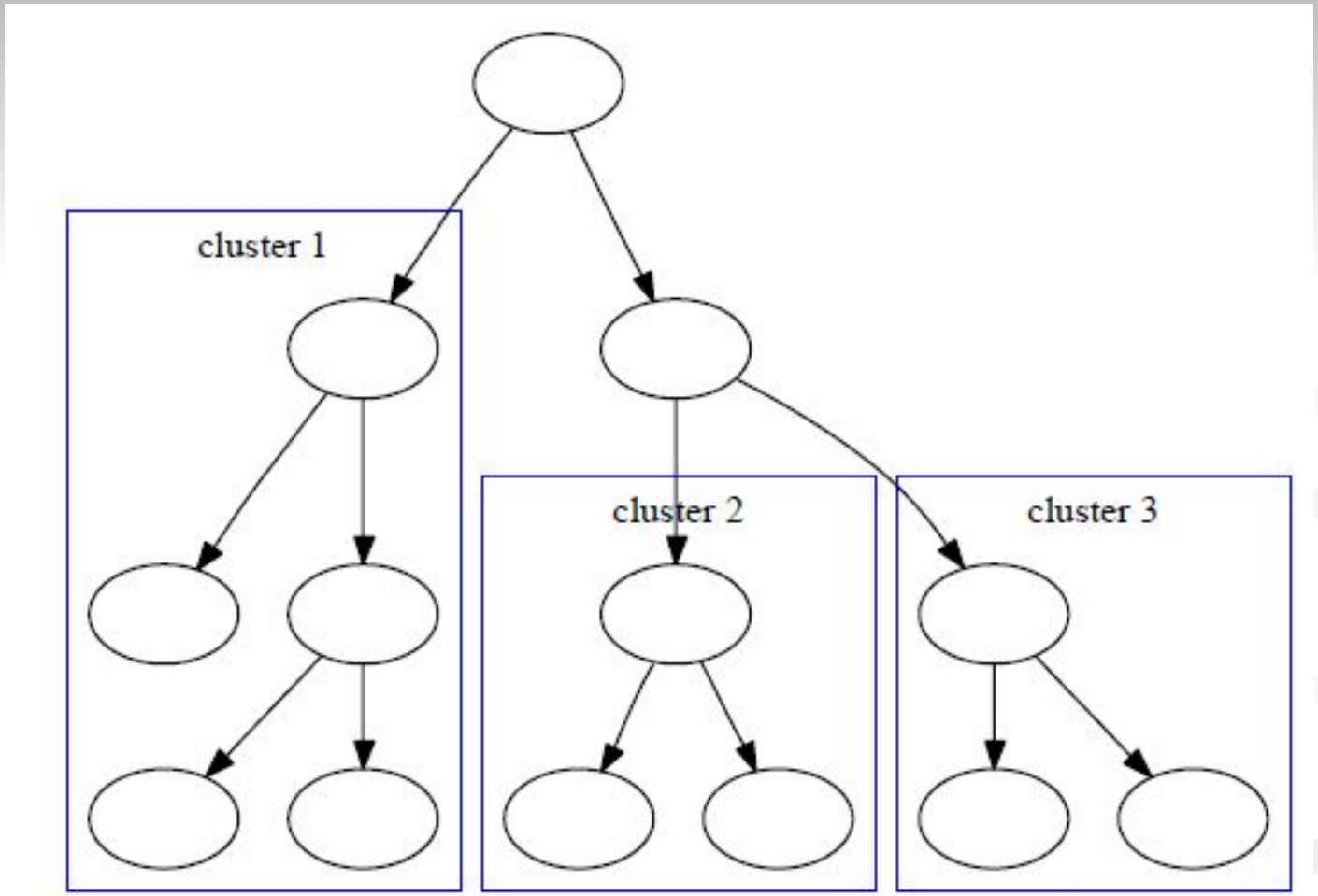
  - Тогда задача кластеризации – разбить этот граф на сильно связанные компоненты.
  - Почему спектральная: используется сразу несколько функций-критериев разбиения.

Алгоритм divide & merge

- Нахождение оптимального разбиения в графе – NP-полная задача (на практике означает, что алгоритм экспоненциальный). Однако существует аппроксимация.

- Две стадии:
  - Иерархическая кластеризация (существует метод с использованием собственных векторов матрицы, который позволяет избежать неэффективного попарного сравнения)
  - Кластеризация результатов предыдущей стадии с использованием стандартных алгоритмов – kmeans, либо другие алгоритмы, с **неизвестным заранее числом кластеров**

Алгоритм divide & merge



Нечеткая совместная корреляция

- Кластеризуются сразу и термины, и документы
- Границы между кластерами нечеткие - термин или документ может входить сразу в несколько кластеров (с различными весами)
- Пример: Fuzzy Codok алгоритм

$$u_{ci} = \frac{1}{C} + \frac{1}{2T_u}\left(\sum_{j=1}^{m} v_{cj}d_{ij} - \frac{1}{C}\sum_{j=1}^{m} v_{cj}d_{ij}\right)$$

$$v_{cj} = \frac{1}{K} + \frac{1}{2T_v}\left(\sum_{i=1}^{n} u_{ci}d_{ij} - \frac{1}{K}\sum_{i=1}^{n} u_{ci}d_{ij}\right)$$

u_{ci} – степень вхождения документа i в кластер c, v_{cj} – степень вхождения термина j в кластер c, d_{ij} – уровень корреляции между документом и термином; m – число терминов, n – число документов, C – число кластеров документов, K – число кластеров терминов.

T_u, T_v – параметры, их надо подбирать – слабое место алгоритма; оптимальные значения параметров зависят от коллекции.

Содержание

1. Оценка качества кластеризации
2. Применение векторной модели в кластеризации
3. Иерархическая кластеризация
4. «Разделяющая» кластеризация
5. Генеративные алгоритмы
6. Спектральная кластеризация
7. Снижение размерности
8. Модели с учетом порядка слов

Снижение размерности

- Матрица термин документ A аппроксимируется матрицей меньшего ранга k A_k.
- Принятая мера качества такой аппроксимации – норма Фробениуса (чем меньше, тем лучше):

$$|\mathbf{A} - \mathbf{A}_k| = \sqrt{\sum_{a \in \mathbf{A}} \sum_{a_k \in \mathbf{A_k}} (a - a_k)^2}$$

Метод главных компонентов (PCA)

- Главные компоненты – ортогональные (независимые) проекции, которые вместе описывают максимальное разнообразие в данных
- Задача эквивалентна поиску оптимального разбиения в двудольном графе

- Главные компоненты получаются из сингулярного разложения матрицы:

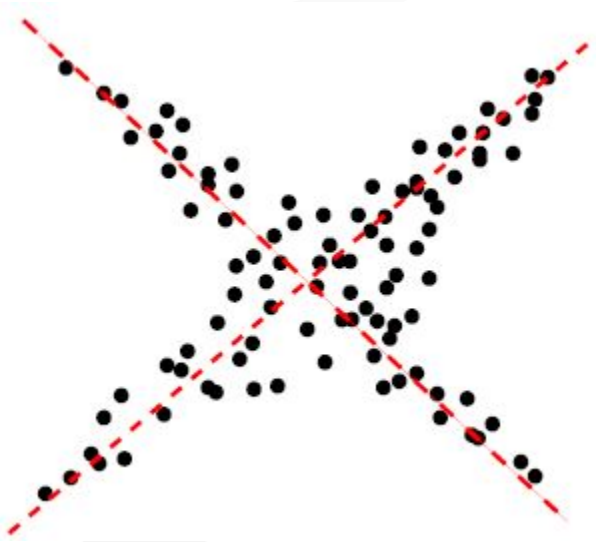
$$A = U\Sigma V^T,$$

Σ – диагональная

- Σ_k – диагональная матрица меньшего ранга, в нее входят k наибольших чисел из Σ
- Искомая проекция:

$$A = U\Sigma_k V^T$$

- Чем больше k, тем лучше аппроксимация



Метод главных компонентов

+ В результате получается оптимальная аппроксимация

+ Различие в расстояниях внутри кластеров и между кластерами становится более резким

- В новом пространстве остаются недискриминирующие свойства – т.е. результаты метода нельзя рассматривать как готовую кластеризацию
- Компоненты должны быть ортогональными – не совсем подходит для текстов, которые могут покрывать несколько тем
- Вычислительно сложный алгоритм, не может использоваться итеративно

Неотрицательная факторизация (NMF)

- Цель: получить аппроксимацию, которая содержит только дискриминирующие факторы
- Исходная матрица аппроксимируется произведением:

$$A \approx UV^T$$

U – базовые вектора mxk, V – матрица коэффициентов nxk

- U может интерпретироваться как набор семантических переменных, V - распределение документов по этим темам
- Начальные значения U и V инициализируются случайно, затем итеративно улучшаются (em-алгоритм)
- Мера качества – обычно Евклидово расстояние (чем меньше, тем лучше):

$$\Theta = \sum_i \sum_j \left(A_{ij} - \sum_l U_{il} V_{jl} \right)$$

- Вместо случайно инициализации можно использовать результаты более простого метода кластеризации (skmns)
- Быстрее, чем метод главных компонентов

Мягкая спектральная кластеризация

- Из редуцированного пространства трудно породить нечеткую кластеризацию, потому что усечение матрицы приводит к искажениям

- Выход: независимая кластеризация терминов и документов; на основе кластеризации терминов порождается нечеткая кластеризация документов и vice versa

Мягкая спектральная кластеризация

- Пространство редуцируется методом главных компонентов
- Проводится кластеризация методом kmeans (или другим)
- Для этих кластеров порождается матрица
- P_1 описывает распределение терминов по кластерам, P_2 – документов

$$P = \begin{bmatrix} P_1 \\ P_2 \end{bmatrix}$$

- Веса терминов высчитываются с помощью трансформации $A P_2$ – **проекция ценроидов** в исходное пространство
- Аналогичная матрица S порождается из кластеризации исходного пространства
- $A^T S_1$ используется как функция вхождения для документов, $A P_2$ – для терминов (используются только дискриминирующие термины)

$$S = \begin{bmatrix} S_1 \\ S_2 \end{bmatrix}$$

- Хорошее качество для пересекающихся тем, но высокая вычислительная сложность

Lingo

- “description comes first”
- Сокращается размерность пространства
- Базисные вектора полученной редуцированной матрицы воспринимаются как метки кластеров
- Эти метки используются для «поиска» документов (как в информационном поиске)

Содержание

1. Оценка качества кластеризации
2. Применение векторной модели в кластеризации
3. Иерархическая кластеризация
4. «Разделяющая» кластеризация
5. Генеративные алгоритмы
6. Спектральная кластеризация
7. Снижение размерности
8. Модели с учетом порядка слов

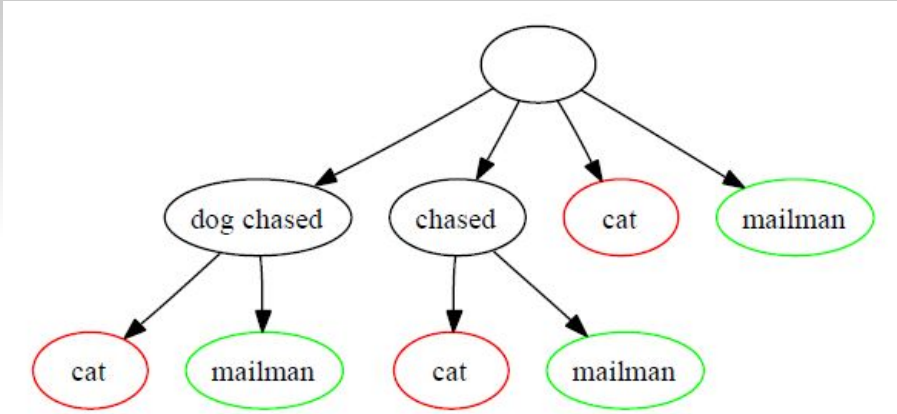
Модели с учетом порядка слов

- *Маша любит Васю, Вася любит Машу* – векторная модель не учитывает различие, но оно есть

- Гипотеза: учет порядка слов может улучшить качество кластеризации
- Кроме того, он позволит создавать более разумные описания кластеров (не набор слов, а короткие фразы)

Кластеризация на основе суффиксных деревьев

- Суффикс – несколько слов с конца предложения (вплоть до предложения целиком)
- Суффиксное дерево: описывает все общие суффиксы документов



dog chased cat,
dog chased mailman

- Общие суффиксы используются для выделения базовых кластеров, которые затем объединяются методом связанных компонентов
- Общая сложность алгоритма: O(n log(n))

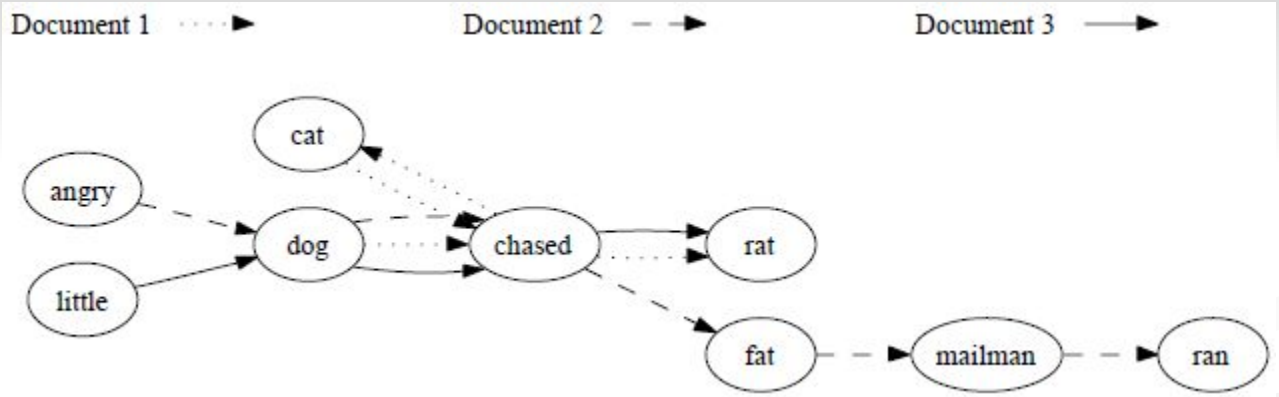
Кластеризация на основе суффиксных деревьев

- Кластеры включают не все документы (документ может иметь суффикс, которые не пересекается ни с одним другим)
- Не учитывается распределение слов по коллекции (не все слова одинаково полезны)
- Учитываются только совпадающие суффиксы, не совпадающие суффиксы не учитываются
- Проверялось на сниппетах, на длинных текстах и больших коллекциях работает плохо

- Можно совмещать учет порядка слов с обычной косинусной мерой

Граф документа

- Doc1: “*cat chased rat*”, “*dog chased rat*”
- Doc2: “*angry dog chased fat mailman*” “*mailman ran*”
- Doc3: “*little dog chased ran*”



- Слова хранятся в вершинах, с учетом частоты
- Нет избыточной информации (как в суффиксных деревьях)
- Это не алгоритм кластеризации, а модель документа; мера близости основана на перекрывающихся подграфах
- Лучше работает совместно с косинусной метрикой, но это – двойная стоимость вычислений

Заключение

- Качество кластеризации определяется по стандартным мерам, при этом итоговая кластеризация не всегда выглядит «естественно»
- Проблема инициализации (итеративные алгоритмы используют случайную инициализацию)
- Проблема описания кластеров (меток)
- Проблема числа кластеров
- Существуют другие методы кластеризации, возможно, они окажутся хороши для текстовых данных
- Возможно другие меры близости, помимо косинусной, окажутся применимы

Источники

- Nicholas O. Andrews and Edward A. Fox, Recent Developments in Document Clustering, October 16, 2007 - http://eprints.cs.vt.edu/archive/00001000/01/docclust.pdf