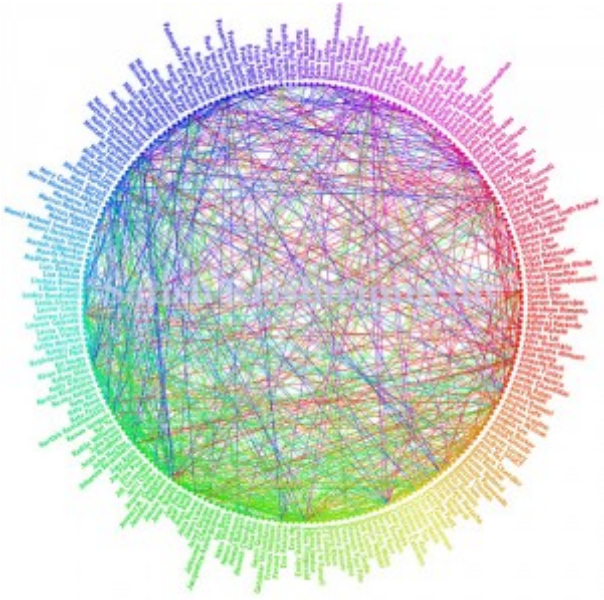
Edunov 20 декабря 2010 в 10:33

Латентно-семантический анализ

Алгоритмы

Из песочницы

Как находить тексты похожие по смыслу? Какие есть алгоритмы для поиска текстов одной тематики? – Вопросы регулярно возникающие на различных программистских форумах. Сегодня я расскажу об одном из подходов, которым активно пользуются поисковые гиганты и который звучит чем-то вроде мантры для SEO aka поисковых оптимизаторов. Этот подход называет латентно-семантический анализ (LSA), он же латентно-семантическое индексирование (LSI)



Предположим, перед вами стоит задача написать алгоритм, который сможет отличать новости о звездах эстрады от новостей по экономике. Первое, что приходит в голову, это выбрать слова которые встречаются исключительно в статьях каждого вида и использовать их для классификации. Очевидная проблема такого подхода: как перечислить все возможные слова и что делать в случае когда в статье есть слова из нескольких классов. Дополнительную сложность представляют омонимы. Т.е. слова имеющие множество значений. Например, слово «банки» в одном контексте может означать стеклянные сосуды а в другом контексте это могут быть финансовые институты.

Латентно-семантический анализ отображает документы и отдельные слова в так называемое «семантическое пространство», в котором и производятся все дальнейшие сравнения. При этом делаются следующие предположения:

1) Документы это просто набор слов. Порядок слов в документах игнорируется. Важно только то, сколько раз то или иное слово встречается в документе.
2) Семантическое значение документа определяется набором слов, которые как правило идут вместе. Например, в биржевых сводках, часто встречаются слова: «фонд», «акция», «доллар»
3) Каждое слово имеет единственное значение. Это, безусловно, сильное упрощение, но именно оно делает проблему разрешимой.

Пример

Для примера я выбрал несколько заголовков с различных новостей. Они выбраны не

совсем случайно, дело в том, что для случайной выборки потребовался бы очень большой объем данных, что сильно затруднило бы дальнейшее изложение. Итак, было выбрано несколько заголовков.

Первым делом из этих заголовков были исключены, так называемые, стоп-символы. Это слова которые встречаются в каждом тексте и не несут в себе смысловой нагрузки, это, прежде всего, все союзы, частицы, предлоги и множество других слов. Полный список использованных стоп-символов можно посмотреть в моей предыдущей статье о стоп-симолах

Далее была произведена операция стемминга. Она не является обязательной, некоторые источники утверждают, что хорошие результаты получаются и без нее. И действительно, если набор текстов достаточно большой, то этот шаг можно опустить. Если тексты на английском языке, то этот шаг тоже можно проигнорировать, в силу того, что количество вариаций той или иной словоформы в английском языке существенно меньше чем в русском. В нашем же случае, пропускать этот шаг не стоит т.к. это приведет к существенной деградации результатов. Для стемминга я пользовался алгоритмом Портера.

Дальше были исключены слова встречающиеся в единственном экземпляре. Это тоже необязательный шаг, он не влияет на конечный результат, но сильно упрощает математические вычисления. В итоге у нас остались, так называемые, индексируемые слова, они выделены жирным шрифтом:

1. Британская **полици**я знает о местонахождении **основател**я **WikiLeaks**
2. В **суд**е **США** начинается процесс **прот**ив россиянина, рассылавшего спам
3. **Церемони**ю **вручени**я **Нобелевск**ой **премии** мира бойкотируют 19 **стран**
4. В **Великобритани**и **арестова**н **основател**ь сайта **Wikileaks** Джулиан Ассандж

5. Украина игнорирует **церемон**ию **вручен**ия **Нобелевск**ой **прем**ии
6. Шведский **суд** отказался рассматривать апелляцию **основател**я **Wikileaks**
7. НАТО и **США** разработали планы обороны **стран** Балтии **прот**ив России
8. **Полиц**ия **Великобритан**ии нашла **основател**я **WikiLeaks**, но, не **арестова**ла
9.В Стокгольме и Осло сегодня состоится **вручен**ие **Нобелевск**их **прем**ий

Латентно семантический анализ

На первом шаге требуется составить частотную матрицу индексируемых слов. В этой матрице строки соответствуют индексированным словам, а столбцы — документам. В каждой ячейке матрицы указано какое количество раз слово встречается в соответствующем документе.

	T1	T2	T3	T4	T5	T6	T7	T8	T9
wikileaks	1	0	0	1	0	1	0	1	0
арестова	0	0	0	1	0	0	0	1	0
великобритан	0	0	0	1	0	0	0	1	0
вручен	0	0	1	0	1	0	0	0	1
нобелевск	0	0	1	0	1	0	0	0	1
основател	1	0	0	1	0	1	0	1	0
полиц	1	0	0	0	0	0	0	1	0
прем	0	0	1	0	1	0	0	0	1
прот	0	1	0	0	0	0	1	0	0
стран	0	0	1	0	0	0	1	0	0
суд	0	1	0	0	0	1	0	0	0
сша	0	1	0	0	0	0	1	0	0
церемон	0	0	1	0	1	0	0	0	0

Следующим шагом мы проводим сингулярное разложение полученной матрицы. Сингулярное разложение это математическая операция раскладывающая матрицу на три составляющих. Т.е. исходную матрицу M мы представляем в виде:

$M = U*W*V^t$

где U и V^t – ортогональные матрицы, а W – диагональная матрица. Причем диагональные элементы матрицы W упорядочены в порядке убывания. Диагональные элементы матрицы W называются сингулярными числами.

wikileaks	0.57	-0.01	0.01	-0.2	0.13	0.16	-0.16	-0.25	-0.64
арестова	0.34	-0	0.07	0.41	-0.42	-0.02	0.1	0.17	0.01
великобритан	0.34	-0	0.07	0.41	-0.42	-0.02	0.1	0.17	-0.01
вручен	0	0.52	0.07	-0.06	-0.08	-0.15	-0.17	0.02	-0.07
нобелевск	0	0.52	0.07	-0.06	-0.08	-0.15	-0.17	0.02	0.32
основател	0.57	-0.01	0.01	-0.2	0.13	0.16	-0.16	-0.25	0.64
полиц	0.31	-0	0.05	0.07	0.57	-0.6	0.29	0.37	-0
прем	0	0.52	0.07	-0.06	-0.08	-0.15	-0.17	0.02	-0.25
прот	0.02	0.03	-0.61	0.13	-0.05	-0.22	0	-0.25	0
стран	0.01	0.22	-0.31	0.39	0.41	0.56	-0.22	0.4	-0
суд	0.12	0.01	-0.38	-0.62	-0.3	0.12	0.21	0.55	-0
сша	0.02	0.03	-0.61	0.13	-0.05	-0.22	0	-0.25	0
церемон	0	0.38	0.03	0.02	0.08	0.31	0.82	-0.29	0

·

3.41	0	0	0	0	0	0	0	0
0	3.3	0	0	0	0	0	0	0
0	0	2.27	0	0	0	0	0	0
0	0	0	1.49	0	0	0	0	0
0	0	0	0	1.19	0	0	0	0
0	0	0	0	0	0.98	0	0	0
0	0	0	0	0	0	0.71	0	0
0	0	0	0	0	0	0	0.43	0
0	0	0	0	0	0	0	0	0

·

T1	T2	T3	T4	T5	T6	T7	T8	T9
0.43	0.05	0.01	0.54	0	0.37	0.01	0.63	0
-0	0.02	0.65	-0.01	0.59	-0	0.09	-0.01	0.47
0.03	-0.7	-0.04	0.06	0.1	-0.16	-0.67	0.09	0.09
-0.22	-0.24	0.15	0.28	-0.11	-0.68	0.44	0.33	-0.13
0.69	-0.32	0.22	-0.49	-0.12	-0.03	0.27	-0.02	-0.19
-0.27	-0.34	0.44	0.29	-0.13	0.45	0.12	-0.31	-0.45
-0.03	0.3	0.14	-0.17	0.44	-0.15	-0.3	0.24	-0.71
-0.3	0.12	0.4	-0.39	-0.53	0.12	-0.23	0.46	0.13
0.35	0.35	0.35	0.35	-0.35	-0.35	-0.35	-0.35	0

Прелесть сингулярного разложения состоит в том, что оно выделяет ключевые составляющие матрицы, позволяя игнорировать шумы. Согласно простым правилам произведения матриц, видно, что столбцы и строки соответствующие меньшим сингулярным значениям дают наименьший вклад в итоговое произведение. Например, мы можем отбросить последние столбцы матрицы U и последние строки матрицы V^t, оставив только первые 2. Важно, что при этом гарантируется, оптимальность полученного произведения. Разложение такого вида называют двумерным сингулярным разложением:

wikileaks	0.57	-0.01
арестова	0.34	-0
великобритан	0.34	-0
вручен	0	0.52
нобелевск	0	0.52
основател	0.57	-0.01
полиц	0.31	-0
прем	0	0.52
прот	0.02	0.03
стран	0.01	0.22
суд	0.12	0.01
сша	0.02	0.03
церемон	0	0.38

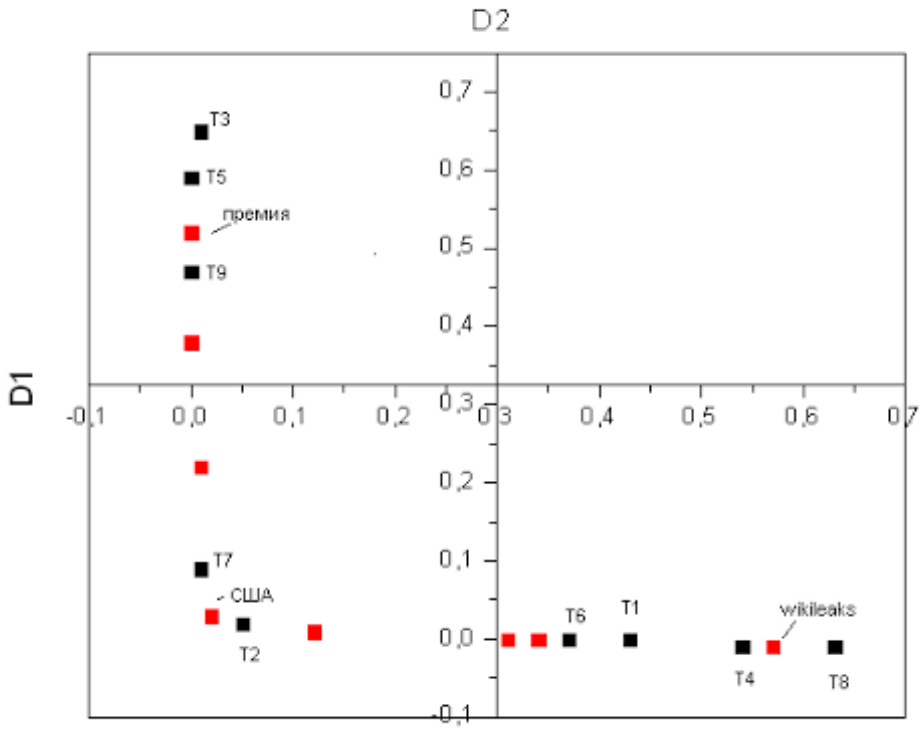
·

3.41	0
0	3.3

·

T1	T2	T3	T4	T5	T6	T7	T8	T9
0.43	0.05	0.01	0.54	0	0.37	0.01	0.63	0
-0	0.02	0.65	-0.01	0.59	-0	0.09	-0.01	0.47

Давайте теперь отметим на графике точки соответствующие отдельным текстам и словам, получится такая занятная картинка:



Из данного графика видно, что статьи образуют три независимые группы, первая группа статей располагается рядом со словом «wikileaks», и действительно, если мы посмотрим названия этих статей становится понятно, что они имеют отношение к wikileaks. Другая группа статей образуется вокруг слова «премия», и действительно в них идет обсуждение нобелевской премии.

На практике, конечно, количество групп будет намного больше, пространство будет не двумерным а многомерным, но сама идея остается той же. Мы можем определять местоположения слов и статей в нашем пространстве и использовать эту информацию

для, например, определения тематики статьи.

Улучшения алгоритма

Легко заметить что подавляющее число ячеек частотной матрицы индексируемых слов, созданной на первом шаге, содержат нули. Матрица сильно разрежена и это свойство может быть использовано для улучшения производительности и потребления памяти при создании более сложной реализации.

В нашем случае тексты были примерно одной и той же длины, в реальных ситуациях частотную матрицу следует нормализовать. Стандартный способ нормализации матрицы TF-IDF

Мы использовали двухмерную декомпозицию SVD-2, в реальных примерах, размерность может составлять несколько сотен и больше. Выбор размерности определяется конкретной задачей, но общее правило таково: чем меньше размерность тем меньше семантических групп вы сможете обнаружить, чем больше размерность, тем большее влияние шумов.

Замечания

Для написания статьи использовалась Java-библиотека для работы с матрицами Jama. Кроме того, функция SVD реализована в известных математических пакетах вроде Mathcad, существуют библиотеки для Python и C++.

Теги: lsa, text mining

Хабы: Алгоритмы

+98 410 82,6k 27 Поделиться

34,0 Карма 0,0 Рейтинг

Сергей @Edunov

Пользователь

ПОХОЖИЕ ПУБЛИКАЦИИ

27 февраля 2013 в 14:46

Рекомендательная система: text mining как средство борьбы с холодным стартом

+26 16,5k 134 8

11 ноября 2012 в 00:50

Text Mining Framework (Java)

+32 29,2k 136 39

2 июня 2010 в 01:00

Data Mining: что внутри

+56 38,3k 115 47

КУРСЫ

Курс по аналитике данных
5 октября 2020 • 6 месяцев • 63 000 ₽ • SkillFactory

Профессия Java-разработчик
6 октября 2020 • 18 месяцев • 99 000 ₽ • SkillFactory

PostgreSQL
6 октября 2020 • 4 месяца • 80 000 ₽ • OTUS

Профессия Android-разработчик
6 октября 2020 • 18 месяцев • 85 200 ₽ • SkillFactory

Факультет интернент-маркетинга
8 октября 2020 • 14 месяцев • 210 000 ₽ • GeekBrains

Больше курсов на Хабр Карьере

Комментарии 27

ЧТО ОБСУЖДАЮТ

aavezel 20 декабря 2010 в 11:12 0

Осталось только повесить этот алгоритм на какой нибудь агрегатор новостей, и сделать индивидуальное ранжирование групп… Web 3.0 в действии…

pixx 21 декабря 2010 в 00:13 0

news.yandex.ru/

aavezel 21 декабря 2010 в 09:33 0

Нифига. Там нет кнопки Like/UnLike, да и групп там явно ограниченное количество…

Antelle 20 декабря 2010 в 11:13 +2

Для выделения нф слов (то, где у вас использовался алгоритм Портера) ещё есть хорошая опен-сорсная библиотека: aot.ru/ — может, кому пригодится

kzn 20 декабря 2010 в 11:32 0

Не совсем так. При использовании AOT потребуется еще один шаг — разрешение возможной омонимии.
В этом смысле Портер лучше :)

Antelle 20 декабря 2010 в 11:34 +2

Сейчас | Вчера | Неделя

Как защищать авторские права, чтобы не чувствовать себя беспомощным идиотом

507 9

WhatsApp, Telegram и Signal выдают телефонные номера всех пользователей

31,7k 93

Личные пристрастия: Sennheiser HD 560S или о том, какие наушники можно купить за $ 200

3,3k 26

Как Tesla выжимает дальность пробега из своих автомобилей

32,1k 139

Фриланс с соцпакетом – дело недалёкого будущего. Или нет?

Мегапост

При использовании Портера разрешение омонимии фактически тоже производится, но единственным простым способом: что получится, если откинуть окончание. При использовании aot вы можете разрешить её как больше нравится, например, по частотному словарю (что там и делается по умолчанию)

 **kzn** 20 декабря 2010 в 11:34 0

Еще плюс — Портер от словаря не зависит

 **Antelle** 20 декабря 2010 в 11:37 +1

Да, безусловно, плюсы есть. Я просто намекнул, что есть такая библиотека, может читателей этой статьи она заинтересует. Я в своё время не знал о ней и изобретал велосипеды.

 **JeanLouis** 20 декабря 2010 в 11:24 0

Поясните, пожалуйста, это:
«Дальше были исключены слова встречающиеся в единственном экземпляре.»
Где встречаются? В отдельных статьях/текстах? Или во всех статьях вместе/тестах?

 **Edunov** 20 декабря 2010 в 11:30 0

В данной конкретной выборке. Например слово «Британская», возможно, в другой выборке оно было бы очень важно. Здесь же оно встречается только один раз и поэтому

включать его в частотную матрицу нет смысла. Это просто оптимизация в целях экономии вычислительных ресурсов.

SeriousDron 20 декабря 2010 в 11:35 0

А если добавили новую статью все надо пересчитывать сначала?
Размерность и содержимое частотной матрицы изменится, некоторые слова могут появится поскольку станут встречаться не в одном экземпляре и т.п.
Или это как обучение, теперь мы знаем какие слова в какой сектор и новые просто смотрим уже по этому.

Edunov 20 декабря 2010 в 14:50 0

Если добавляется новая статья то можно не пересчитывать, но тогда вы не сможете выявить новые измерения (кластеры, группы).
Поэтому, на практике, имеет смысл регулярно пересчитывать, но не обязательно с каждой новой статьей.

lightcaster 20 декабря 2010 в 11:42 0

> В нашем случае тексты были примерно одной и той же длины, в реальных ситуациях частотную матрицу > следует нормализовать. Стандартный способ нормализации матрицы TF-IDF

TF-IDF не столько способ нормализации, сколько способ выделить наиболее значимые в рамках документа слова. Он максимален, если термин часто встречается в документе, и редко — во всем наборе документов.

В остальном хорошая статья на правильную тему.

ps кстати, не в курсе как работать с большим набором терминов? Обычно это проблема для LSA.

lightcaster 20 декабря 2010 в 11:46 0

… с большим набором терминов… — имел ввиду измерения.

Edunov 20 декабря 2010 в 14:55 0

Например, существуют алгоритмы случайной проекции. Описание на английском: www.rni.org/kanerva/cogsci2k-poster.txt

НЛО прилетело и опубликовало эту надпись здесь

trurl123 8 июня 2011 в 11:52 0

ссылка не работает

edeldm 20 декабря 2010 в 12:09 +2

сравнивал автор алгоритм с другими? например с методом главных компонент?

Edunov 20 декабря 2010 в 14:57 0

PCA (метод главных компонент) и LSA чисто технически очень похожи, здесь не ставилась задача обзора всех возможных техник, но за идею спасибо, сравню и напишу результаты.

 **mikhailian** 20 декабря 2010 в 14:59 –1

Схожих алгоритмов пруд пруди. Вот ещё один, там несколько «русских» подвязаются:

 **mikhailian** 20 декабря 2010 в 15:12 –1

Мля… Хотел дать ссылку на страничку википедии про Formal Concept Analysis, но карма не та.

Кстати, у автора есть одно неочевидное упущение: первая матрица у него бинарная, а базар про целочисленную. Хоть бы двоечку где вписал, чтобы было понятнее, о чём речь.

 **Edunov** 20 декабря 2010 в 15:17 +1

Там ясно написано «В каждой ячейке матрицы указано **какое количество раз** слово встречается в соответствующем документе» Двоечку из заголовков новостей не выудишь, их авторы избегают повторения слов в названии.

 **ocnk** 20 декабря 2010 в 12:45 0

Стоит наверное еще использовать стемпер со словарем. с тем же hunspell, что бы не собирать статистику по не существующим словам

Quiz 20 декабря 2010 в 14:36 +4

Солнышко красивое :)

Romachev 20 декабря 2010 в 15:01 –2

Спасибо, очень интересный материал.

corbenov 20 декабря 2010 в 15:41 +2

автору срочно нужно попасть на Хабр

Kallikanzarid 22 января 2014 в 14:52 0

А в чем преимущество сингулярного разложения над наивным байесовским классификатором?

Только полноправные пользователи могут оставлять комментарии. Войдите, пожалуйста.

САМОЕ ЧИТАЕМОЕ

Сутки | Неделя | Месяц

Гугл-программисты. Как идиот набрал на работу идиотов

+108 52,5k 108 306

Стивен Вольфрам: кажется, мы близки к пониманию фундаментальной теории физики, и она прекрасна

+259 118k 502 368

Программисты, ходите на собеседования

+40 17k 78 178

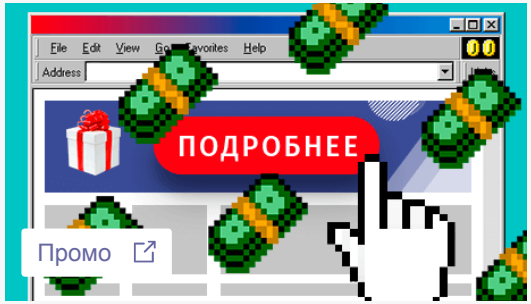
WhatsApp, Telegram и Signal выдают телефонные номера всех пользователей

+41 31,7k 84 93

Розыгрыш Logitech MX Master 3 в комментариях

Мегапост

МИНУТОЧКУ ВНИМАНИЯ



Кешбэк на контекст: партнерская программа

Эволюция доменных имён: от простого ASCII к Unicode

Разместить