

Генеративные вероятностные модели: на примере текстов: NB, LSI, PLSI, LDA

И. Куралёнок, Н. Поваров

Яндекс

СПб, 2013

Немного о статистической обработке текстов

Пара фактов:

- Нужно исходно для поиска
- Область называется Information Retrieval (SIGIR/CIKM/ECIR/ПОМИП/RCDL/Диалог)
- Читать “Modern Information Retrieval”, Ricardo Baeza-Yates, Berthier Ribeiro-Neto

Из всего этого добра нам сегодня нужно только “bag of words”.

Как можно представить текст

В IR:

Документ: последовательность абзацев

Абзац: последовательность предложений

Предложение: бог знает что такое, но со словами
внутри

К черту подробности! Документ — банка со словами,
которую еще и потрясли.

Банка бывает: бинарная, частотная, нормированная,
BM25, TFIDF, etc.

Наивный байесов классификатор

Даже в такой простой модели можно делать например так:

$$\begin{aligned} p(c|d) &= \frac{p(c) \prod_{w_i} p(c|w_i, w_{i-1}, \dots, w_1)}{\prod_{w_i} p(w_i)} \\ &= \frac{1}{Z} p(c) \prod_{w_i} p(c|w_i) \end{aligned}$$

Такая штука называется “наивный байес” и долгое время считалась стандартом де-факто текстового классификатора.

Вспоминая прошлую лекцию (LSI)

Сложим всю коллекцию документов $X = (d_1, \dots, d_m)^T$. Каждый документ: $d_i = (w_1, \dots, w_n)$, где w_i — частота слова в документе. Матрица разрежена, можно попробовать проделать фокус из прошлой лекции:

$$X = U_r^T \Sigma_r V_r$$

Тогда вектора в U_r и V_r , можно рассматривать как образы слов и документов в общем пространстве гипотез размерности r . Будем предсказывать по словам запроса (клиентам) документы (товары), которые соответствуют предыдущим покупкам (документам, где слово засветилось).

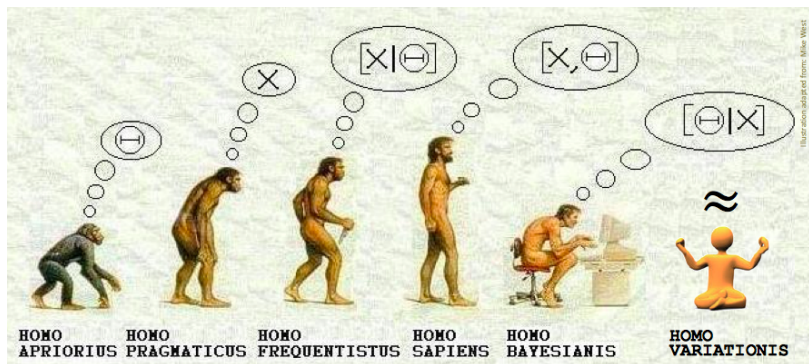
LSI в терминах вероятностей

Мы знаем (уже), что SVD разложение работает “не очень”, может быть можно все переложить в плоскость вероятностного моделирования по следующей схеме:

- сформулируем способ получить документ с помощью некоего случайного процесса;
- подберем параметры процесса так, чтобы наилучшим способом объяснить появление коллекции;
- для нового документа найдем вероятность быть “сгенерированным” полученной моделью.

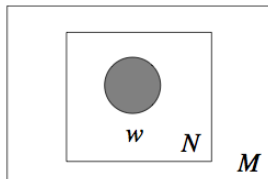
Параметров распределения может быть сильно меньше, чем данных в коллекции и в этом смысле мы делаем “понижение ранга” модели. Чтобы описывать это безобразия нам помогут графические модели.

Вероятностные модели



картинка из презентации Kay P. Brodersen

Униграммная модель



$$p(d) = \prod_{w_i} p(w_i)$$

- 1 Кинем количество слов n по Пуассону
- 2 По полученному количеству будем независимо и с повторениями выбирать слова $p(w_i)$

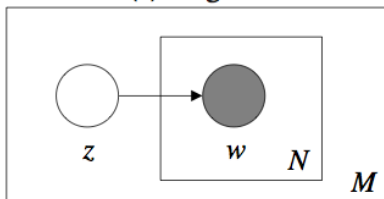
Униграммный классификатор

- 1 поделим коллекцию на классы;
- 2 для каждого класса подберем параметры униграммной модели;
- 3 по новому документу сравним вероятности быть сгенерированным по соответствующей классу модели.

$$p(d|c) = \prod_{w_i} p(w_i|c)$$

Чем отличается от Naive Bayes?

Смесь униграмм



$$p(d) = \sum_z p(z) \prod_{w_i} p(w_i|z)$$

- 1 Кинем топик z по весам $p(z)$
- 2 Кинем количество слов n по Пуассону
- 3 По полученному количеству будем независимо и с повторениями выбирать слова с вероятностями $p(w_i|z)$

Документ может относиться только к одной скрытой теме z .

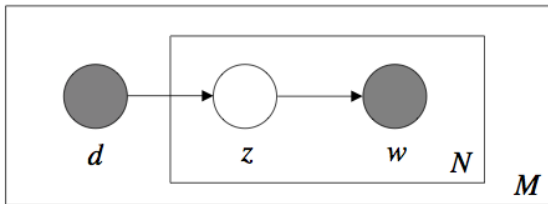
Как подбирать смесь униграмм?

Пусть класс документа является скрытой переменной.

Expectation исходя из текущих представлений о $p(w_i|z)$ найдем к какому классу принадлежит документ

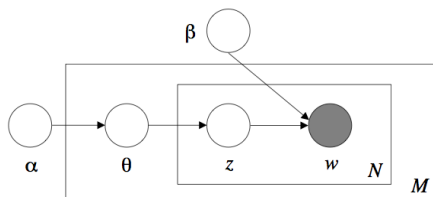
Maximization с учетом того, как распределились документы по классам, максимизируем правдоподобие коллекции

Probabilistic latent semantic allocation (pLSA)



- 1 Выберем для документа топик z по весам $p(z|d)$
- 2 Кинем количество слов n по Пуассону
- 3 По полученному количеству будем независимо и с повторениями выбирать слова с вероятностями $p(w_i|z)$

Latent Dirichlet Allocation (LDA)



- 1 Сгенерируем распределения весов топиков $\theta \sim Dir(\alpha)$
- 2 Кинем количество слов n по Пуассону
- 3 Для каждого слова:
 - 1 Выберем $z_i \sim Multinomial(\theta)$
 - 2 Получим слово с вероятностью $p(w|z_i, \beta)$

Вывод LDA I

Теперь все стало сильно сложнее:

$$p(d|\alpha, \beta) = \int p(\theta|\alpha) \left(\prod_1^n \sum_z p(z_i|\theta) p(w_i|z_i, \beta) \right) d\theta$$

Или, с учетом известных распределений:

$$p(d|\alpha, \beta) = \frac{\Gamma(\sum_i \alpha_i)}{\prod_i \Gamma(\alpha_i)} \int \left(\prod_{t=1}^k \theta_i^{\alpha_t - 1} \right) \left(\prod_{i=1}^n \sum_{t=1}^k \prod_{j=1}^m (\theta_t \beta_{tj})^{w_i^j} \right) d\theta$$

А для коллекции еще и так:

$$\arg \max_{\alpha, \beta} \sum_d \log p(d|\alpha, \beta)$$

Variational bayes

$$\begin{aligned}\log(p(d)) &= \log\left(\frac{p(y,\theta)}{p(\theta|y)}\right) \\&= \int q(\theta) \log \frac{p(y,\theta)}{p(\theta|y)} d\theta \\&= \int q(\theta) \log \frac{p(y,\theta)}{p(\theta|y)} \frac{q(\theta)}{q(\theta)} d\theta \\&= \int q(\theta) \left(\log \frac{q(\theta)}{p(\theta|y)} + \log \frac{p(y,\theta)}{q(\theta)} \right) d\theta \\&= \left(\int q(\theta) \log \frac{q(\theta)}{p(\theta|y)} d\theta \right) + \left(\int q(\theta) \log \frac{p(y,\theta)}{q(\theta)} d\theta \right)\end{aligned}$$

Красная часть называется Kullback–Leibler divergence ($KL(p\|q) = D_{KL}(p\|q)$). Можно искать не точное распределение p , а его приближение q .

Вывод LDA II

В нашем случае q выберем так:

$$q(\theta, z|\gamma, \phi) = q(\theta|\gamma) \prod_{i=1}^n q(z_i|\phi_i)$$

где $\gamma \sim Dir$, и решим проблему:

$$(\hat{\gamma}, \hat{\phi}) = \arg \min_{(\gamma, \phi)} D_{KL}(q(\theta, z|\gamma, \phi) \| p(\theta, z|d, \alpha, \beta))$$

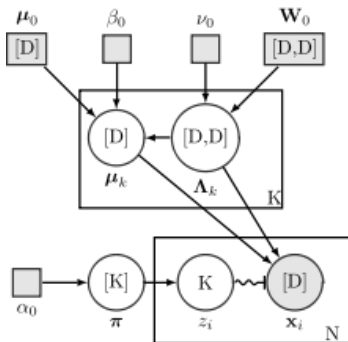
Вывод LDA III

Expectation: для каждого документа найдем параметры $\{\hat{\gamma}, \hat{\phi}\}$

Maximization: используя $q(\theta, z | \hat{\gamma}_d, \hat{\phi}_d)$ вместо $p(\theta, z | d, \alpha, \beta)$ максимизируем исходный likelihood

Где еще применяется VB

- Для вывода сложных графических моделей типа:



- В моделировании сложных распределений независимыми простыми (ICA)

Задание на дом

- Задание будет в svn.
- Домашнее задание за пятую неделю ещё проверяется, но пока - "ужас-ужас".