

ОТ НАИВНОГО БАЙЕСА К LDA

Сергей Николенко

СПбГУ — Санкт-Петербург

8 апреля 2025 г.

Random facts:



- 8 апреля 1546 г. Тридентский собор признал Biblia Vulgata в качестве официальной версии Библии; работа над переводом на латынь шла ещё при папе Дамасии в IV-V веках, а в 1456 г. Иоганн Гутенберг воспроизвёл её текст в первой печатной книге
- 8 апреля 1513 г. Хуан Понсе де Леон объявил открытую им Флориду владением Испании (на самом деле он искал источник вечной молодости); в его честь назван Понсе, второй крупнейший город Пуэрто-Рико, и по некоторым подсчётам 30% современного населения Пуэрто-Рико являются дальними потомками Понсе де Леона
- 8 апреля 1820 г. Йоргос Кентротас, возделывая своё поле на Милосе, обнаружил разбитую на две части статую Венеры: торс без рук и задрапированную нижнюю часть
- 8 апреля 1912 г. прошёл премьерный показ кукольного мультфильма «Прекрасная Люканида, или Война усачей с рогачами»; Владислав Старевич создал любовную историю жука-рогача Люканиды и жука-усака Героса при помощи покадровой съёмки, но многие критики посчитали мультфильм результатом дрессировки жуков
- 8 апреля 1971 г. вблизи Лондона состоялся первый Всемирный конгресс цыган, который принял цыганские гимн и флаг; с тех пор 8 апреля — Международный день цыган

НАИВНЫЙ БАЙЕСОВСКИЙ КЛАССИФИКАТОР

- Классическая задача машинного обучения и information retrieval – категоризация текстов.
- Дан набор текстов, разделённый на категории. Нужно обучить модель и потом уметь категоризовать новые тексты.
- Атрибуты $a_1, a_2, \dots, a_n$ – это слова, v – тема текста (или атрибут вроде «спам / не спам»).
- Bag-of-words model: забываем про порядок слов, составляем словарь. Теперь документ – это вектор, показывающий, сколько раз каждое слово из словаря в нём встречается.

- Заметим, что даже это – сильно упрощённый взгляд: для слов ещё довольно-таки важен порядок, в котором они идут...
- Но и это ещё не всё: получается, что $p(a_1, a_2, \dots, a_n | x = v)$ – это вероятность *в точности такого набора слов* в сообщениях на разные темы. Очевидно, такой статистики взять неоткуда.
- Значит, надо дальше делать упрощающие предположения.
- Наивный байесовский классификатор – самая простая такая модель: давайте предположим, что все слова в словаре условно независимы при условии данной категории.

- Иначе говоря:

$$p(a_1, a_2, \dots, a_n | x = v) = p(a_1 | x = v) p(a_2 | x = v) \dots p(a_n | x = v).$$

- Итак, наивный байесовский классификатор выбирает v как

$$v_{NB}(a_1, a_2, \dots, a_n) = \arg \max_{v \in V} p(x = v) \prod_{i=1}^n p(a_i | x = v).$$

- В парадигме классификации текстов мы предполагаем, что разные слова в тексте на одну и ту же тему появляются независимо друг от друга. Однако, несмотря на такие бредовые предположения, naïve Bayes на практике работает очень даже неплохо (и этому есть разумные объяснения).

- Но в деталях реализации наивного байесовского классификатора есть тонкости.
- Сейчас мы рассмотрим два разных подхода к naïve Bayes, которые дают разные результаты: мультиномиальный (multinomial) и многомерный (multivariate).

- В многомерной модели документ – это вектор бинарных атрибутов, показывающих, встретилось ли в документе то или иное слово.
- Когда мы подсчитываем правдоподобие документа, мы перемножаем вероятности того, что встретилось каждое слово из документа и вероятности того, что не встретилось каждое (словарное) слово, которое не встретилось.
- Получается модель многомерных испытаний Бернулли. Наивное предположение в том, что события «встретилось ли слово» предполагаются независимыми.

- Математически: пусть $V = \{w_t\}_{t=1}^{|V|}$ – словарь. Тогда документ d_i – это вектор длины $|V|$, состоящий из битов B_{it} ; $B_{it} = 1$ iff слово w_t встречается в документе d_i .
- Правдоподобие принадлежности d_i классу c_j :

$$p(d_i | c_j) = \prod_{t=1}^{|V|} (B_{it}p(w_t | c_j) + (1 - B_{it})(1 - p(w_t | c_j))).$$

- Для обучения такого классификатора нужно обучить вероятности $p(w_t | c_j)$.

- Обучение – дело нехитрое: пусть дан набор документов $D = \{d_i\}_{i=1}^{|D|}$, которые уже распределены по классам c_j (возможно, даже вероятностно распределены), дан словарь $V = \{w_t\}_{t=1}^{|V|}$, и мы знаем биты B_{it} (знаем документы).
- Тогда можно подсчитать оптимальные оценки вероятностей того, что то или иное слово встречается в том или ином классе (при помощи лапласовой оценки):

$$p(w_t | c_j) = \frac{1 + \sum_{i=1}^{|D|} B_{it} p(c_j | d_i)}{2 + \sum_{i=1}^{|D|} p(c_j | d_i)}.$$

- Априорные вероятности классов можно подсчитать как $p(c_j) = \frac{1}{|D|} \sum_{i=1}^{|D|} p(c_j | d_i)$.
- Тогда классификация будет происходить как

$$\begin{aligned} c &= \arg \max_j p(c_j) p(d_i | c_j) = \\ &= \arg \max_j \left(\frac{1}{|D|} \sum_{i=1}^{|D|} p(c_j | d_i) \right) \prod_{t=1}^{|V|} (B_{it} p(w_t | c_j) + (1 - B_{it})(1 - p(w_t | c_j))) = \\ &= \arg \max_j \left(\log \left(\sum_{i=1}^{|D|} p(c_j | d_i) \right) + \sum_{t=1}^{|V|} \log (B_{it} p(w_t | c_j) + (1 - B_{it})(1 - p(w_t | c_j))) \right). \end{aligned}$$

- В мультиномиальной модели документ – это последовательность событий. Каждое событие – это случайный выбор одного слова из того самого «bag of words».
- Когда мы подсчитываем правдоподобие документа, мы перемножаем вероятности того, что мы достали из мешка те самые слова, которые встретились в документе. Наивное предположение в том, что мы достаём из мешка разные слова независимо друг от друга.
- Получается мультиномиальная генеративная модель, которая учитывает количество повторений каждого слова, но не учитывает, каких слов *нет* в документе.

- Математически: пусть $V = \{w_t\}_{t=1}^{|V|}$ – словарь. Тогда документ d_i – это вектор длины $|d_i|$, состоящий из слов, каждое из которых «вынуто» из словаря с вероятностью $p(w_t | c_j)$.
- Правдоподобие принадлежности d_i классу c_j :

$$p(d_i | c_j) = p(|d_i|) |d_i|! \prod_{t=1}^{|V|} \frac{1}{N_{it}!} p(w_t | c_j)^{N_{it}},$$

где N_{it} – количество вхождений w_t в d_i .

- Для обучения такого классификатора тоже нужно обучить вероятности $p(w_t | c_j)$.

- Обучение: пусть дан набор документов $D = \{d_i\}_{i=1}^{|D|}$, которые уже распределены по классам c_j (возможно, даже вероятностно распределены), дан словарь $V = \{w_t\}_{t=1}^{|V|}$, и мы знаем вхождения N_{it} .
- Тогда можно подсчитать апостериорные оценки вероятностей того, что то или иное слово встречается в том или ином классе (не забываем сглаживание – правило Лапласа):

$$p(w_t \mid c_j) = \frac{1 + \sum_{i=1}^{|D|} N_{it} p(c_j \mid d_i)}{|V| + \sum_{s=1}^{|V|} \sum_{i=1}^{|D|} N_{is} p(c_j \mid d_i)}.$$

- Априорные вероятности классов можно подсчитать как $p(c_j) = \frac{1}{|D|} \sum_{i=1}^{|D|} p(c_j | d_i)$.
- Тогда классификация будет происходить как

$$\begin{aligned} c &= \arg \max_j p(c_j) p(d_i | c_j) = \\ &= \arg \max_j \left(\frac{1}{|D|} \sum_{i=1}^{|D|} p(c_j | d_i) \right) p(|d_i|) |d_i|! \prod_{t=1}^{|V|} \frac{1}{N_{it}!} p(w_t | c_j)^{N_{it}} = \\ &= \arg \max_j \left(\log \left(\sum_{i=1}^{|D|} p(c_j | d_i) \right) + \sum_{t=1}^{|V|} N_{it} \log p(w_t | c_j) \right). \end{aligned}$$

- В наивном байесе есть два сильно упрощающих дело предположения:
 - мы знаем метки тем всех документов;
 - у каждого документа только одна тема.
- Мы сейчас уберём оба эти ограничения.
- Во-первых, что можно сделать, если мы не знаем метки тем, т.е. если датасет неразмеченный?

- Тогда это превращается в задачу кластеризации.
- Её можно решать ЕМ-алгоритмом (Expectation-Maximization, используется в ситуациях, когда есть много скрытых переменных, причём если бы мы их знали, модель стала бы простой):
 - на Е-шаге считаем ожидания того, какой документ какой теме принадлежит;
 - на М-шаге пересчитываем наивным байесом вероятности $p(w | t)$ при фиксированных метках.
- Это простое обобщение.

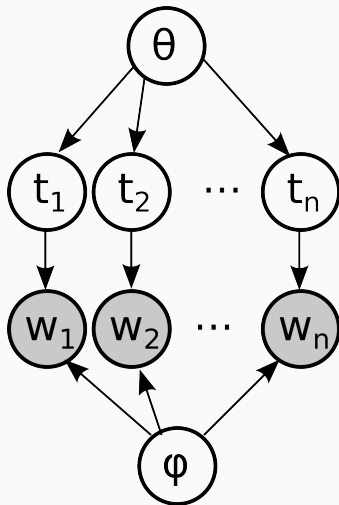
ТЕМАТИЧЕСКОЕ МОДЕЛИРОВАНИЕ

- В прошлый раз мы много говорили о наивном байесе и начали его обобщать.
- Пока обобщили на обучение без учителя (кластеризацию).
- А ещё в наивном байесе у документа только одна тема.
- Но это же не так! На самом деле документ говорит о многих темах (но не слишком многих).
- Давайте попробуем это учесть.

- Рассмотрим такую модель:
 - каждое слово в документе d порождается некоторой темой $t \in T$;
 - документ порождается некоторым распределением на темах $p(t \mid d)$;
 - слово порождается именно темой, а не документом: $p(w \mid d, t) = p(w \mid t)$;
 - итог получается такая функция правдоподобия:

$$p(w \mid d) = \sum_{t \in T} p(w \mid t) p(t \mid d).$$

- Эта модель называется probabilistic latent semantic analysis, pLSA (Hoffmann, 1999).



- Получается как-то так:

Алгоритм 2. Рациональный ЕМ-алгоритм для тематической модели (2).

Вход: коллекция D , число тем $|T|$, начальные приближения матриц Φ и Θ ;

Выход: параметры модели Φ и Θ ;

1 **повторять**

2 обнулить n_{wt} , n_{td} , n_t для всех $d \in D$, $w \in W$, $t \in T$;

3 **для всех** $d \in D$, $w \in d$

4 $n_{tdw} := n_{dw} \varphi_{wt} \theta_{td} / \sum_{\tau} \varphi_{w\tau} \theta_{\tau d}$ для всех $t \in T$;

5 увеличить n_{wt} , n_{td} , n_t на n_{tdw} для всех $t \in T$;

6 $\varphi_{wt} := n_{wt} / n_t$ для всех $w \in W$, $t \in T$;

7 $\theta_{td} := n_{td} / n_d$ для всех $d \in D$, $t \in T$;

8 **пока** Φ и Θ не сойдутся;

- Как её обучать? Мы можем оценить $p(w \mid d) = \frac{n_{wd}}{n_d}$, а нужно найти:
 - $\phi_{wt} = p(w \mid t)$;
 - $\theta_{td} = p(t \mid d)$.
- Максимизируем правдоподобие

$$p(D) = \prod_{d \in D} \prod_{w \in d} p(d, w)^{n_{dw}} = \prod_{d \in D} \prod_{w \in d} \left[\sum_{t \in T} p(w \mid t) p(t \mid d) \right]^{n_{dw}}.$$

- Как максимизировать такие правдоподобия?

- ЕМ-алгоритмом. На Е-шаге ищем, сколько слов w в документе d из темы t :

$$n_{dwt} = n_{dw}p(t \mid d, w) = n_{dw} \frac{\phi_{wt}\theta_{td}}{\sum_{s \in T} \phi_{ws}\theta_{sd}}.$$

- А на М-шаге пересчитываем параметры модели:

$$\begin{aligned} n_{wt} &= \sum_d n_{dwt}, & n_t &= \sum_w n_{wt}, & \phi_{wt} &= \frac{n_{wt}}{n_t}, \\ n_{td} &= \sum_{w \in d} n_{dwt}, & \theta_{td} &= \frac{n_{td}}{n_d}. \end{aligned}$$

- Вот и весь вывод в pLSA.

- Можно даже не хранить всю матрицу n_{dwt} , а двигаться по документам, каждый раз добавляя n_{dwt} сразу к счётчикам n_{wt} , n_{td} .

Алгоритм 2. Рациональный ЕМ-алгоритм для тематической модели (2).

Вход: коллекция D , число тем $|T|$, начальные приближения матриц Φ и Θ ;

Выход: параметры модели Φ и Θ ;

1 **повторять**

2 обнулить n_{wt} , n_{td} , n_t для всех $d \in D$, $w \in W$, $t \in T$;

3 **для всех** $d \in D$, $w \in d$

4 $n_{tdw} := n_{dw} \varphi_{wt} \theta_{td} / \sum_{\tau} \varphi_{w\tau} \theta_{\tau d}$ для всех $t \in T$;

5 увеличить n_{wt} , n_{td} , n_t на n_{tdw} для всех $t \in T$;

6 $\varphi_{wt} := n_{wt} / n_t$ для всех $w \in W$, $t \in T$;

7 $\theta_{td} := n_{td} / n_d$ для всех $d \in D$, $t \in T$;

8 **пока** Φ и Θ не сойдутся;

- Чего тут не хватает?
 - Во-первых, разложение такое, конечно, будет сильно не единственным.
 - Во-вторых, параметров очень много, явно будет оверфиттинг, если корпус не на порядки больше числа тем.
 - А совсем хорошо было бы получать не просто устойчивое решение, а обладающее какими-нибудь заданными хорошими свойствами.
- Всё это мы можем решить как?

- Правильно, регуляризацией. Есть целая наука о разных регуляризаторах для pLSA (К.В. Воронцов).
- В общем виде так: добавим регуляризаторы R_i в логарифм правдоподобия:

$$\sum_{d \in D} \sum_{w \in d} n_{dw} \ln \sum_{t \in T} \phi_{wt} \theta_{td} + \sum_i \tau_i R_i(\Phi, \Theta).$$

- Тогда в EM-алгоритме на M-шаге появятся частные производные R :

$$n_{wt} = \left[\sum_{d \in D} n_{dwt} + \phi_{wt} \frac{\partial R}{\partial \phi_{wt}} \right]_+,$$
$$n_{td} = \left[\sum_{w \in d} n_{dwt} + \theta_{td} \frac{\partial R}{\partial \theta_{td}} \right]_+$$

- Чтобы доказать, EM надо рассмотреть как решение задачи оптимизации через условия Каруша-Куна-Такера.

- И теперь мы можем кучу разных регуляризаторов вставить в эту модель:
 - регуляризатор сглаживания (позже, это примерно как LDA);
 - регуляризатор разреживания: максимизируем KL-расстояние между распределениями ϕ_{wt} и θ_{td} и равномерным распределением;
 - регуляризатор контрастирования: минимизируем ковариации между векторами ϕ_t , чтобы в каждой теме выделилось своё лексическое ядро (характерные слова);
 - регуляризатор когерентности: будем награждать за слова, которые в документах стоят ближе друг к другу;
 - и так далее, много всего можно придумать.

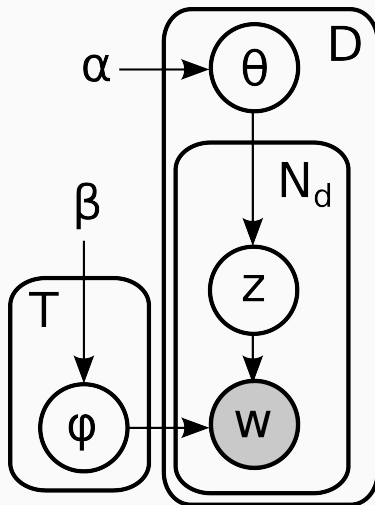
LDA



- Развитие идей pLSA – LDA (Latent Dirichlet Allocation).
- Это фактически байесовский вариант pLSA, сейчас нарисуем картинку, добавим априорные распределения и посмотрим, как сработают наши методы приближённого вывода.
- Задача та же: смоделировать большую коллекцию текстов (например, для information retrieval или классификации).

- У одного документа может быть несколько тем. Давайте построим иерархическую байесовскую модель:
 - на первом уровне – смесь, компоненты которой соответствуют «темам»;
 - на втором уровне – мультиномиальная переменная с априорным распределением Дирихле, которое задаёт «распределение тем» в документе.

- Если формально: слова берутся из словаря $\{1, \dots, V\}$; слово – это вектор w , $w_i \in \{0, 1\}$, где ровно одна компонента равна 1.
- Документ – последовательность из N слов $\mathbf{w}$. Нам дан корпус из M документов $D = \{\mathbf{w}_d \mid d = 1..M\}$.
- Генеративная модель LDA выглядит так:
 - выбрать $\theta \sim \text{Di}(\alpha)$;
 - для каждого из N слов w_n :
 - выбрать тему $z_n \sim \text{Mult}(\theta)$;
 - выбрать слово $w_n \sim p(w_n \mid z_n, \beta)$ по мультиномиальному распределению.



LDA: ЧТО ПОЛУЧАЕТСЯ [BLEI, 2012]

Topics

gene 0.04
dna 0.02
genetic 0.01
...

life 0.02
evolve 0.01
organism 0.01
...

brain 0.04
neuron 0.02
nerve 0.01
...

data 0.02
number 0.02
computer 0.01
...

Documents

Seeking Life's Bare (Genetic) Necessities

COLD SPRING HARBOR, NEW YORK—How many **genes** does an **organism** need to survive? Last week at the genome meeting here,* two genome researchers with radically different approaches presented complementary views of the basic genes needed for **life**. One research team, using **computer** analyses to compare known **genomes**, concluded that today's **organisms** can be sustained with just 250 genes, and that the earliest life forms required a mere 128 **genes**. The other researcher mapped genes in a simple parasite and estimated that for this organism, 800 genes are plenty to do the job—but that anything short of 100 wouldn't be enough.

Although the numbers don't match precisely, those **predictions**

"are not all that far apart," especially in comparison to the 75,000 **genes** in the human genome, notes Siv Andersson, a biologist at Uppsala University in Sweden who arrived at the 800 number. But coming up with a consensus answer may be more than just a **genetic** numbers game, particularly as more and more **genomes** are completely mapped and sequenced. "It may be a way of organizing any newly **sequenced genome**," explains Arcady Mushegian, a **computational** molecular biologist at the National Center for Biotechnology Information (NCBI) in Bethesda, Maryland. Comparing an

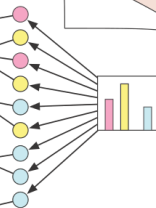


* Genome Mapping and Sequencing, Cold Spring Harbor, New York, May 8 to 12.

Stripping down. Computer analysis yields an estimate of the minimum modern and ancient genomes.

SCIENCE • VOL. 272 • 24 MAY 1996

Topic proportions and assignments



- Два основных подхода к выводу в сложных вероятностных моделях, в том числе LDA:
 - *вариационные приближения*: рассмотрим более простое семейство распределений с новыми параметрами и найдём в нём наилучшее приближение к неизвестному распределению;
 - *сэмплирование*: будем набрасывать точки из сложного распределения, не считая его явно, а запуская марковскую цепь под графиком распределения (частный случай – сэмплирование по Гиббсу).
- Сэмплирование по Гиббсу обычно проще расширить на новые модификации LDA, но вариационный подход быстрее и часто стабильнее.

- Рассмотрим задачу байесовского вывода, т.е. оценки апостериорного распределения θ и z после нового документа:

$$p(\theta, \mathbf{z} \mid \mathbf{w}, \alpha, \beta) = \frac{p(\theta, \mathbf{z}, \mathbf{w} \mid \alpha, \beta)}{p(\mathbf{w} \mid \alpha, \beta)}.$$

- Правдоподобие набора слов $\mathbf{w}$ оценивается как

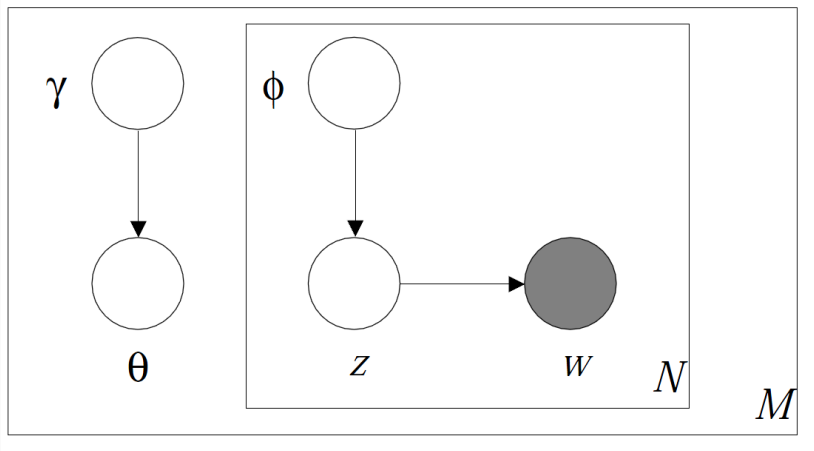
$$p(\mathbf{w} \mid \alpha, \beta) = \frac{\Gamma(\sum_i \alpha_i)}{\prod_i \Gamma(\alpha_i)} \int \left[\prod_{i=1}^k \theta_i^{\alpha_i - 1} \right] \left[\prod_{n=1}^N \sum_{i=1}^k \prod_{j=1}^V (\theta_i \beta_{ij})^{w_n^j} \right] d\theta,$$

и это трудно посчитать, потому что θ и β путаются друг с другом.

- Вариационное приближение – рассмотрим семейство распределений

$$q(\theta, z \mid \mathbf{w}, \gamma, \phi) = p(\theta \mid \mathbf{w}, \gamma) \prod_{n=1}^N p(z_n \mid \mathbf{w}, \phi_n).$$

- Тут всё расщепляется, и мы добавили вариационные параметры γ (Дирихле) и ϕ (мультиномиальный).
- Заметим, что параметры для каждого документа могут быть свои – всё условно по $\mathbf{w}$.



- Теперь можно искать минимум KL-расстояния:

$$(\gamma^*, \phi^*) = \arg \min_{(\gamma, \phi)} \text{KL}(q(\theta, z \mid \mathbf{w}, \gamma\phi) \parallel p(\theta, \mathbf{z} \mid \mathbf{w}, \alpha, \beta)).$$

- Для этого сначала воспользуемся уже известной оценкой из неравенства Йенсена:

$$\begin{aligned} \log p(\mathbf{w} \mid \alpha, \beta) &= \log \int_{\theta} \sum_{\mathbf{z}} p(\theta, \mathbf{z}, \mathbf{w} \mid \alpha, \beta) d\theta = \\ &= \log \int_{\theta} \sum_{\mathbf{z}} \frac{p(\theta, \mathbf{z}, \mathbf{w} \mid \alpha, \beta) q(\theta, \mathbf{z})}{q(\theta, \mathbf{z})} d\theta \geq \\ &\geq E_q [\log p(\theta, \mathbf{z}, \mathbf{w} \mid \alpha, \beta)] - E_q [\log q(\theta, \mathbf{z})] =: L(\gamma, \phi; \alpha, \beta). \end{aligned}$$

- Распишем произведения:

$$L(\gamma, \phi; \alpha, \beta) = E_q [p(\theta | \alpha)] + E_q [p(\mathbf{z} | \theta)] + E_q [p(\mathbf{w} | \mathbf{z}, \beta)] - \\ - E_q [\log q(\theta)] - E_q [\log q(\mathbf{z})] .$$

- Свойство распределения Дирихле: если $X \sim \text{Di}(\alpha)$, то

$$E[\log(X_i)] = \Psi(\alpha_i) - \Psi\left(\sum_i \alpha_i\right),$$

где $\Psi(x) = \frac{\Gamma'(x)}{\Gamma(x)}$ – дигамма-функция.

- Теперь можно выписать каждый из пяти членов.

$$\begin{aligned}
 L(\gamma, \phi; \alpha, \beta) = & \log \Gamma\left(\sum_{i=1}^k \alpha_i\right) - \sum_{i=1}^k \log \Gamma(\alpha_i) + \sum_{i=1}^k (\alpha_i - 1) \left[\Psi(\gamma_i) - \Psi\left(\sum_{j=1}^k \gamma_j\right) \right] + \\
 & + \sum_{n=1}^N \sum_{i=1}^k \phi_{ni} \left[\Psi(\gamma_i) - \Psi\left(\sum_{j=1}^k \gamma_j\right) \right] + \\
 & + \sum_{n=1}^N \sum_{i=1}^k \sum_{j=1}^V w_n^j \phi_{ni} \log \beta_{ij} - \\
 & - \log \Gamma\left(\sum_{i=1}^k \gamma_i\right) + \sum_{i=1}^k \log \Gamma(\gamma_i) - \sum_{i=1}^k (\gamma_i - 1) \left[\Psi(\gamma_i) - \Psi\left(\sum_{j=1}^k \gamma_j\right) \right] - \\
 & - \sum_{n=1}^N \sum_{i=1}^k \phi_{ni} \log \phi_{ni}.
 \end{aligned}$$

- Теперь осталось только брать частные производные этого выражения.
- Сначала максимизируем его по ϕ_{ni} (вероятность того, что n -е слово было порождено темой i); надо добавить λ -множители Лагранжа, т.к. $\sum_{j=1}^k \phi_{nj} = 1$.
- В итоге получится:

$$\phi_{ni} \propto \beta_{iv} e^{\Psi(\gamma_i) - \Psi(\sum_{j=1}^k \gamma_j)},$$

где v – номер того самого слова, т.е. единственная компонента $w_n^v = 1$.

- Потом максимизируем по γ_i , i -й компоненте апостериорного Дирихле-параметра.
- Получится

$$\gamma_i = \alpha_i + \sum_{n=1}^N \phi_{ni}.$$

- Соответственно, для вывода нужно просто пересчитывать ϕ_{ni} и γ_i друг через друга, пока оценка не сойдётся.

- Теперь давайте попробуем оценить параметры α и β по корпусу документов D .
- Мы хотим найти α и β , которые максимизируют

$$\ell(\alpha, \beta) = \sum_{d=1}^M \log p(\mathbf{w}_d \mid \alpha, \beta).$$

- Подсчитать $p(\mathbf{w}_d \mid \alpha, \beta)$ мы не можем, но у нас есть нижняя оценка $L(\gamma, \phi; \alpha, \beta)$, т.к.

$$\begin{aligned} p(\mathbf{w}_d \mid \alpha, \beta) &= \\ &= L(\gamma, \phi; \alpha, \beta) + \text{KL}(q(\theta, z \mid \mathbf{w}_d, \gamma\phi) \parallel p(\theta, \mathbf{z} \mid \mathbf{w}_d, \alpha, \beta)). \end{aligned}$$

- EM-алгоритм:
 1. найти параметры $\{\gamma_d, \phi_d \mid d \in D\}$, которые оптимизируют оценку (как выше);
 2. зафиксировать их и оптимизировать оценку по α и β .

- Для β это тоже делается нехитро:

$$\beta_{ij} \propto \sum_{d=1}^M \sum_{n=1}^{N_d} \phi_{dni} w_n^j.$$

- Для α_i получается система уравнений, которую можно решить методом Ньютона.

СПАСИБО!

Спасибо за внимание!

