Категоризация текстов и модель LDA

Сергей Николенко

Центр Речевых Технологий, 2012

Центр Речевых Технологий

Outline

Категоризация текстов

- 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 $a_1, a_2, \ldots, a_n$ – 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.

Naive Bayes

- 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 $p(a_1, a_2, \ldots, a_n | x = v)$ – 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.

Naive Bayes

- Иначе говоря:

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

- Итак, наивный байесовский классификатор выбирает v как

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

- 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, naive Bayes на практике работает очень даже неплохо (и этому есть разумные объяснения).

Многомерная модель

- В деталях реализации наивного байесовского классификатора прячется небольшой дьяволёнок.
- Сейчас мы рассмотрим два разных подхода к naive Bayes, которые дают разные результаты: мультиномиальный (multinomial) и многомерный (multivariate).

Многомерная модель

- 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.

Многомерная модель

- 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:

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

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

Многомерная модель

- 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):

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

Многомерная модель

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

$$c = \arg\max_j p(c_j)p(d_i \mid c_j) =$$

$$= \arg\max_j \left(\frac{1}{|D|}\sum_{i=1}^{|D|} p(c_j \mid d_i)\right) \prod_{t=1}^{|V|} (B_{it}p(w_t \mid c_j) + (1 - B_{it})(1 - p(w_t \mid c_j))) =$$

$$= \arg\max_j \left(\log(\sum_{i=1}^{|D|} p(c_j \mid d_i)) + \sum_{t=1}^{|V|} \log\left(B_{it}p(w_t \mid c_j) + (1 - B_{it})(1 - p(w_t \mid c_j\right)\right.$$

Мультиномиальная модель

- 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.

Мультиномиальная модель

- 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 $p(w_t \mid c_j)$.
- Правдоподобие принадлежности d_i классу c_j:

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

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

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

Мультиномиальная модель

- 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):

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

Мультиномиальная модель

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

$$c = \arg\max_j p(c_j)p(d_i \mid c_j) =$$

$$= \arg\max_j \left(\frac{1}{|D|}\sum_{i=1}^{|D|} p(c_j \mid d_i)\right) p(|d_i|)|d_i|! \prod_{t=1}^{|V|} \frac{1}{N_{it}!} p(w_t \mid c_j)^{N_{it}} =$$

$$= \arg\max_j \left(\log\left(\sum_{i=1}^{|D|} p(c_j \mid d_i)\right) + \sum_{t=1}^{|V|} N_{it} \log p(w_t \mid c_j)\right).$$

LDA

- Более сложная модель – LDA (Latent Dirichlet Allocation).
- Задача: смоделировать большую коллекцию текстов (например, для information retrieval или классификации).
- Мы знаем наивный подход: скрытая переменная – тема, слова получаются из темы независимо по дискретному распределению.
- Аналогично работают и подходы, основанные на кластеризации.
- Давайте чуть усложним.

LDA

- Очевидно, что у одного документа может быть несколько тем; подходы, которые кластеризуют документы по темам, никак этого не учитывают.
- 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.

LDA

- Если формально: слова берутся из словаря $\{1, \ldots, V\}$; 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 $\mathcal{D} = \{\mathbf{w}_d \mid d = 1..M\}$.
- Генеративная модель LDA выглядит так.
  1. Выбрать $N \sim p(N \mid \xi)$.
  2. Выбрать $\theta \sim \mathrm{Di}(\alpha)$.
  3. Для каждого из N слов w_n:
     1. выбрать тему $z_n \sim \mathrm{Mult}(\theta)$;
     2. выбрать слово $w_n \sim p(w_n \mid z_n, \beta)$ по мультиномиальному распределению.

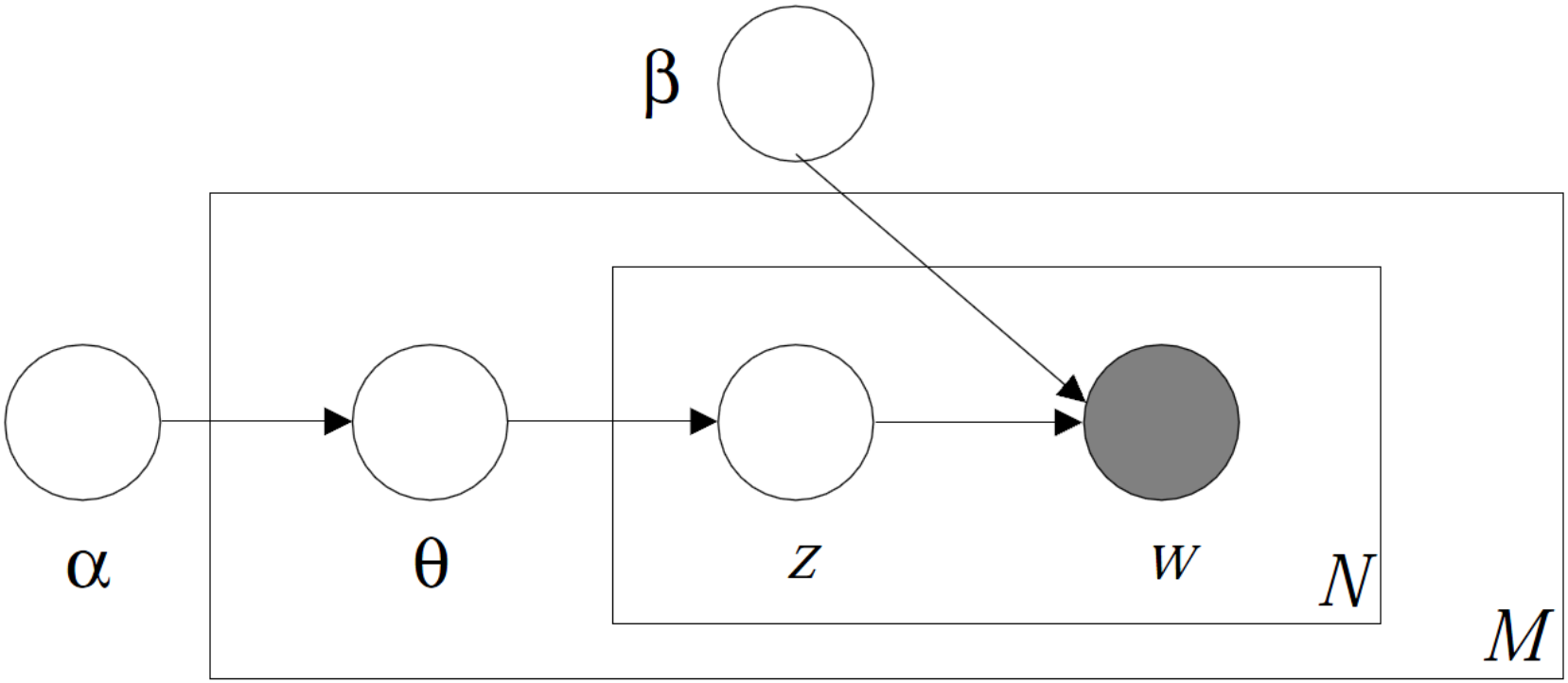
LDA

- Мы пока для простоты фиксируем число тем k, считаем, что β – это просто набор параметров $\beta_{ij} = p(w^j = 1 \mid z^i = 1)$, которые нужно оценить, и не беспокоимся о распределении на N.
- Совместное распределение тогда выглядит так:

$$p(\theta, \mathbf{z}, \mathbf{w}, N \mid \alpha, \beta) = p(N \mid \xi) p(\theta \mid \alpha) \prod_{n=1}^{N} p(z_n \mid \theta) p(w_n \mid z_n, \beta).$$

- В отличие от обычной кластеризации с априорным распределением Дирихле, мы тут не выбираем кластер один раз, а затем накидываем слова из этого кластера, а для каждого слова выбираем по распределению θ, по какой теме оно будет набросано.

LDA: графическая модель



Вывод в LDA

- 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.

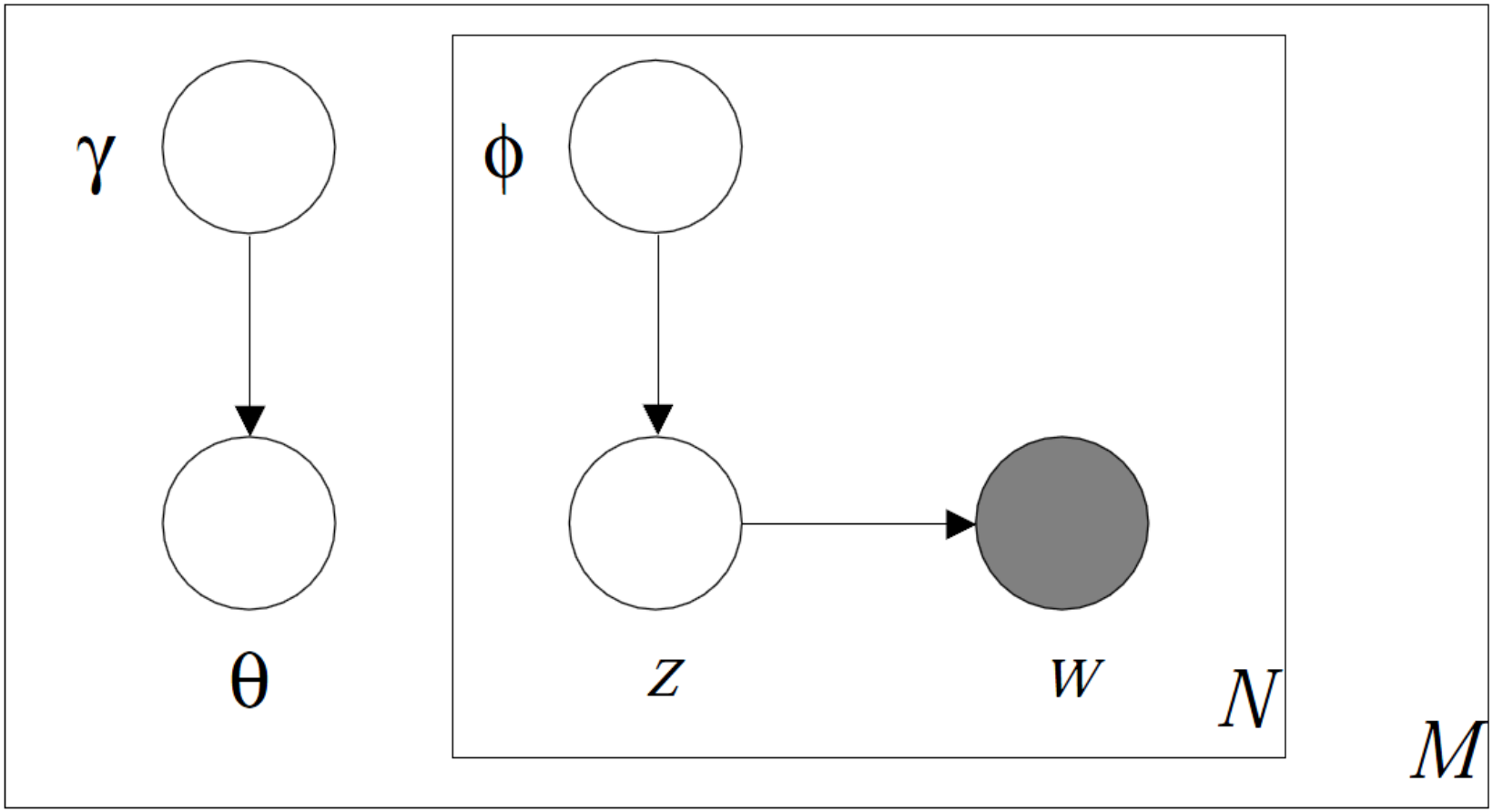
Вывод в LDA

- 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 $\mathbf{w}$.

LDA: вариационное приближение



Thank you!

Спасибо за внимание!