Вариационные приближения и LDA

Сергей Николенко

Академия MADE — Mail.Ru

12 июня 2020 г.

Random facts:

- 12 июня — День России; именно 12 июня 1990 г. на первом съезде народных депутатов РСФСР была принята Декларация о государственном суверенитете РСФСР, а 12 июня 1991 г. прошли выборы президента РСФСР, на которых победил Борис Ельцин
- 12 июня 1815 г. *Morning Chronicle* вследствие интриг Натана Ротшильда сообщила о победе Наполеона при Ватерлоо за несколько дней до сражения
- 12 июня 1817 г. Карл Дрез создал первый прототип велосипеда, по сути беговел, который сам назвал «машиной для ходьбы»; позднее эту машину стали называть «дрезиной» в честь изобретателя, хотя в наше время смысл слова изменился
- 12 июня 1849 г. Льюис Хаслетт запатентовал противогаз, а 12 июня 1897 г. Карл Элзенер запатентовал швейцарский армейский нож
- 12 июня 1824 г. в Париже вышла первая и единственная научная работа Никола Леонарда Сади Карно «Размышления о движущей силе огня и о машинах, способных развивать эту силу»
- 12 июня 1942 г. юная Анна получила подарок на тринадцатилетие — дневник

EM в общем виде

Приближения

- Часто нужно оценивать $p(\mathbf{Z} \mid \mathbf{X})$ для латентных переменных $\mathbf{Z}$ и данных $\mathbf{X}$.
- Но это слишком сложно! Один вариант — сэмплировать из $p(\mathbf{Z} \mid \mathbf{X})$.
- Другой вариант — лапласовские приближения, но тоже нечасто работают.
- Давайте решать в общем виде.

Обоснование алгоритма EM

- Вспомним сначала формальное обоснование алгоритма EM.
- Мы решаем задачу максимизации правдоподобия по данным $\mathsf{X} = \{x_1, \ldots, x_N\}$.

$$L(\theta \mid \mathsf{X}) = p(\mathsf{X} \mid \theta) = \prod p(x_i \mid \theta)$$

  или, что то же самое, максимизации $\ell(\theta \mid \mathsf{X}) = \log L(\theta \mid \mathsf{X})$.
- EM может помочь, если этот максимум трудно найти аналитически.

Обоснование алгоритма EM

- Давайте предположим, что в данных есть *скрытые компоненты*, такие, что если бы мы их знали, задача была бы проще.
- Замечание: совершенно не обязательно эти компоненты должны иметь какой-то физический смысл. :) Может быть, так просто удобнее.
- В любом случае, получается набор данных $\mathsf{Z} = (\mathsf{X}, \mathcal{Y})$ с совместной плотностью

$$p(z \mid \theta) = p(x, y \mid \theta) = p(y \mid x, \theta) p(x \mid \theta).$$

- Получается полное правдоподобие $L(\theta \mid \mathsf{Z}) = p(\mathsf{X}, \mathcal{Y} \mid \theta)$. Это случайная величина (т.к. $\mathcal{Y}$ неизвестно).

Обоснование алгоритма EM

- Заметим, что настоящее правдоподобие $L(\theta) = E_Y[p(\mathsf{X}, \mathcal{Y} \mid \theta) \mid \mathsf{X}, \theta]$.
- E-шаг алгоритма EM вычисляет условное ожидание (логарифма) полного правдоподобия при условии X и текущих оценок параметров θ_n:

$$Q(\theta, \theta_n) = E\left[\log p(\mathsf{X}, \mathcal{Y} \mid \theta) \mid \mathsf{X}, \theta_n\right].$$

- Здесь θ_n – текущие оценки, а θ – неизвестные значения (которые мы хотим получить в конечном счёте); т.е. $Q(\theta, \theta_n)$ – это функция от θ.

Обоснование алгоритма EM

- E-шаг алгоритма EM вычисляет условное ожидание (логарифма) полного правдоподобия при условии X и текущих оценок параметров θ:

$$Q(\theta, \theta_n) = E\left[\log p(\mathsf{X}, \mathcal{Y} \mid \theta) \mid \mathsf{X}, \theta_n\right].$$

- Условное ожидание – это

$$E\left[\log p(\mathsf{X}, \mathcal{Y} \mid \theta) \mid \mathsf{X}, \theta_n\right] = \int_y \log p(\mathsf{X}, y \mid \theta) p(y \mid \mathsf{X}, \theta_n) \mathrm{d}y,$$

  где $p(y \mid \mathsf{X}, \theta_n)$ – маргинальное распределение скрытых компонентов данных.
- EM лучше всего применять, когда это выражение легко подсчитать, может быть, даже аналитически.
- Вместо $p(y \mid \mathsf{X}, \theta_n)$ можно подставить $p(y, \mathsf{X} \mid \theta_n) = p(y \mid \mathsf{X}, \theta_n) p(\mathsf{X} \mid \theta_n)$, от этого ничего не изменится.

Обоснование алгоритма EM

- В итоге после E-шага алгоритма EM мы получаем функцию $Q(\theta, \theta_n)$.
- На M-шаге мы максимизируем

$$\theta_{n+1} = \arg\max_\theta Q(\theta, \theta_n).$$

- Затем повторяем процедуру до сходимости.
- В принципе, достаточно просто находить θ_{n+1}, для которого $Q(\theta_{n+1}, \theta_n) > Q(\theta_n, \theta_n)$ – Generalized EM.
- Осталось понять, что значит $Q(\theta, \theta_n)$ и почему всё это работает.

Обоснование алгоритма EM

- Мы хотели перейти от θ_n к θ, для которого $\ell(\theta) > \ell(\theta_n)$.

$$\ell(\theta) - \ell(\theta_n) =$$

$$= \log\left(\int_y p(\mathsf{X} \mid y, \theta) p(y \mid \theta) \mathrm{d}y\right) - \log p(\mathsf{X} \mid \theta_n) =$$

$$= \log\left(\int_y p(y \mid \mathsf{X}, \theta_n) \frac{p(\mathsf{X} \mid y, \theta) p(y \mid \theta)}{p(y \mid \mathsf{X}, \theta_n)} \mathrm{d}y\right) - \log p(\mathsf{X} \mid \theta_n) \geq$$

$$\geq \int_y p(y \mid \mathsf{X}, \theta_n) \log\left(\frac{p(\mathsf{X} \mid y, \theta) p(y \mid \theta)}{p(y \mid \mathsf{X}, \theta_n)}\right) \mathrm{d}y - \log p(\mathsf{X} \mid \theta_n) =$$

$$= \int_y p(y \mid \mathsf{X}, \theta_n) \log\left(\frac{p(\mathsf{X} \mid y, \theta) p(y \mid \theta)}{p(\mathsf{X} \mid \theta_n) p(y \mid \mathsf{X}, \theta_n)}\right) \mathrm{d}y.$$

Обоснование алгоритма EM

- Получили

$$\ell(\theta) \geq l(\theta, \theta_n) =$$
$$= \ell(\theta_n) + \int_y p(y \mid \mathsf{X}, \theta_n) \log \left(\frac{p(\mathsf{X} \mid y, \theta) p(y \mid \theta)}{p(\mathsf{X} \mid \theta_n) p(y \mid \mathcal{X}, \theta_n)} \right) \mathrm{d}y.$$

- Мы нашли нижнюю оценку на $\ell(\theta)$ везде, касание происходит в точке θ_n.

Вариационные приближения

Вариационный вывод

- Вариационный вывод: функционалы, производные по функциям... в общем, можно оптимизировать функционалы.
- Для нас это значит, что можно оптимизировать приближение q из какого-то класса к заданному p.
- Пусть есть $\mathbf{X} = \{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$ и $\mathbf{Z} = \{\mathbf{z}_1, \ldots, \mathbf{z}_N\}$.
- Мы знаем $p(\mathbf{X}, \mathbf{Z})$ из модели, хотим найти приближение для $p(\mathbf{Z} \mid \mathbf{X})$ и $p(\mathbf{X})$.

Вариационный вывод

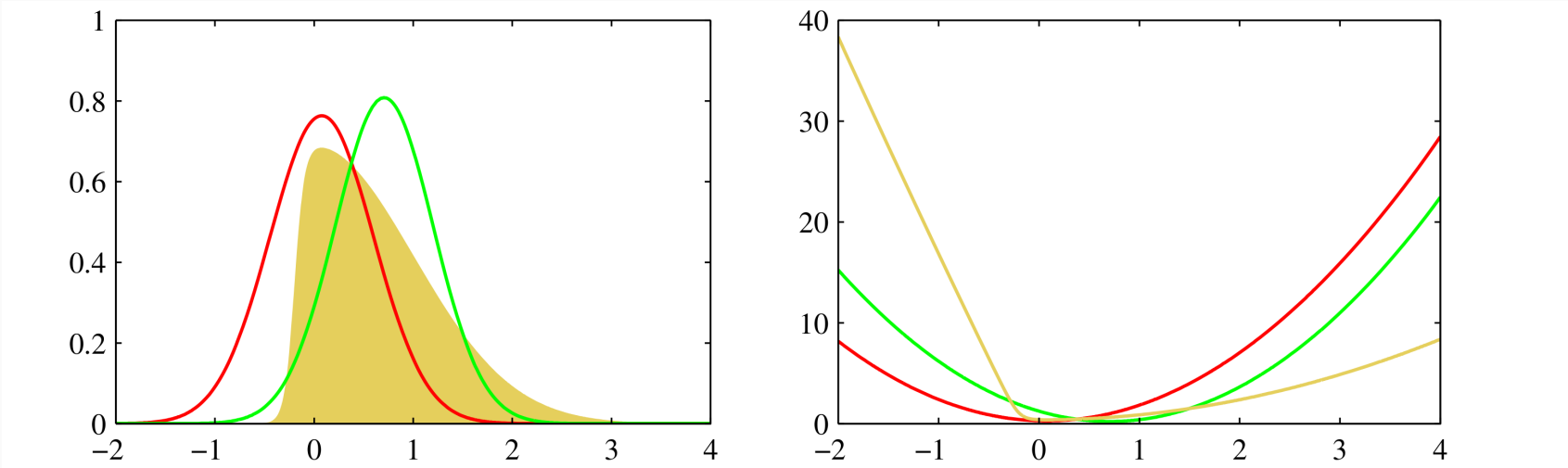
- Как и в EM:

$$\ln p(\mathsf{X}) = \mathcal{L}(q) + \mathrm{KL}(q\|p), \text{ где}$$
$$\mathcal{L}(q) = \int q(\mathsf{Z}) \ln \frac{p(\mathsf{X}, \mathsf{Z})}{q(\mathsf{Z})} \mathrm{d}\mathsf{Z},$$
$$\mathrm{KL}(q\|p) = -\int q(\mathsf{Z}) \ln \frac{p(\mathsf{Z} \mid \mathsf{X})}{q(\mathsf{Z})} \mathrm{d}\mathsf{Z}.$$

- $\mathcal{L}(q)$ — это вариационная нижняя оценка, её можно теперь максимизировать, и KL будет автоматически минимизироваться.

Вариационный вывод

- Пример – сравним с лапласовским:



- Если $q(\mathcal{Z})$ произвольное, то мы просто получим $q(\mathcal{Z}) = p(\mathsf{Z} \mid \mathsf{X})$; но это вряд ли получится.
- Будем ограничивать.

Факторизуемые распределения

- Главный частный случай — пусть $\mathbf{Z} = \mathbf{Z}_1 \sqcup \ldots \sqcup \mathbf{Z}_M$, и

$$q(\mathbf{Z}) = \prod_{i=1}^{M} q_i(\mathbf{Z}_i).$$

- Но больше никаких предположений! В этом прелесть — оптимизируем сразу функции!
- Это соответствует теории среднего поля в физике (mean field theory).

Факторизуемые распределения

- Тогда:

$$\begin{aligned}\mathcal{L}(q) &= \int \prod_i q_i \left(\ln p(\mathsf{X}, \mathsf{Z}) - \sum_i \ln q_i \right) \mathrm{d}\mathsf{Z} \\ &= \int q_j \left[\int \ln p(\mathsf{X}, \mathsf{Z}) \prod_{i \neq j} q_i \mathrm{d}\mathsf{Z}_i \right] \mathrm{d}\mathsf{Z}_j - \int q_j \ln q_j \mathrm{d}\mathsf{Z}_j + \mathrm{const} \\ &= \int q_j \ln \tilde{p}(\mathsf{X}, \mathsf{Z}_j) \mathrm{d}\mathsf{Z}_j - \int q_j \ln q_j \mathrm{d}\mathsf{Z}_j + \mathrm{const},\end{aligned}$$

$$\text{где } \ln \tilde{p}(\mathsf{X}, \mathsf{Z}_j) = \mathsf{E}_{i \neq j} \left[\ln p(\mathsf{X}, \mathsf{Z}) \right] + \mathrm{const}.$$

- Как максимизировать теперь $\mathcal{L}(q)$ по q_j?

Факторизуемые распределения

- Надо заметить, что мы получили там KL-дивергенцию между $q_j(\mathsf{Z}_j)$ и $\tilde{p}(\mathsf{X}, \mathsf{Z}_j)$.
- Т.е. оптимальное решение получится при

$$\ln q_j^*(\mathsf{Z}_j) = \mathsf{E}\left[\ln p(\mathsf{X}, \mathsf{Z})\right] + \text{const}.$$

- Константа здесь просто для нормализации.
- Оказывается, достаточно взять ожидание от логарифма совместного распределения.
- Но явных формул не получается, потому что ожидание считается по остальным q_i^*, $i \neq j$.
- И всё-таки обычно что-то можно сделать; давайте рассмотрим примеры.

Двумерный гауссиан

- Первый пример — приблизим двумерный гауссиан одномерными:

$$p(\mathbf{z}) = \mathcal{N}(\mathbf{z} \mid \boldsymbol{\mu}, \boldsymbol{\Lambda}^{-1}),$$

$$\boldsymbol{\mu} = \begin{pmatrix} \mu_1 \\ \mu_2 \end{pmatrix}, \quad \boldsymbol{\Lambda} = \begin{pmatrix} \Lambda_{11} & \Lambda_{12} \\ \Lambda_{21} & \Lambda_{22} \end{pmatrix}.$$

- И мы хотим приблизить $q(\mathbf{z}) = q_1(z_1)q_2(z_2)$.
- Повычисляем...

Двумерный гауссиан

- ...получится, что

$$\ln q_1^*(z_1) = -\frac{1}{2} z_1^2 \Lambda_{11} + z_1 \mu_{11} \Lambda_{11} - z_1 \Lambda_{12} (\mathsf{E}[z_2] - \mu_2) + \text{const}.$$

- Чудесным образом получился гауссиан! Сам собой, без предположений.
- Найдём его среднее и дисперсию...

Двумерный гауссиан

- ...получится

$$q_1^*(z_1) = \mathcal{N}(z_1 \mid m_1, \Lambda_{11}^{-1}), \text{ где}$$

$$m_1 = \mu_1 - \Lambda_{11}^{-1}\Lambda_{12}(\mathbf{E}[z_2] - \mu_2).$$

- Аналогично,

$$q_2^*(z_2) = \mathcal{N}(z_2 \mid m_2, \Lambda_{11}^{-1}), \text{ где}$$

$$m_2 = \mu_2 - \Lambda_{22}^{-1}\Lambda_{21}(\mathbf{E}[z_1] - \mu_1).$$

- Какое решение у этой системы?

Двумерный гауссиан

- Да просто $\mathsf{E}[z_1] = m_1 = \mu_1$, $\mathsf{E}[z_2] = m_2 = \mu_2$.
- А если бы мы минимизировали $\mathrm{KL}(p\|q)$, получилось бы

$$\mathrm{KL}(p\|q) = -\int p(\mathbf{Z}) \left[\sum_i \ln q_i(\mathbf{Z}_i)\right] \mathrm{d}\mathbf{Z} + \mathrm{const},$$

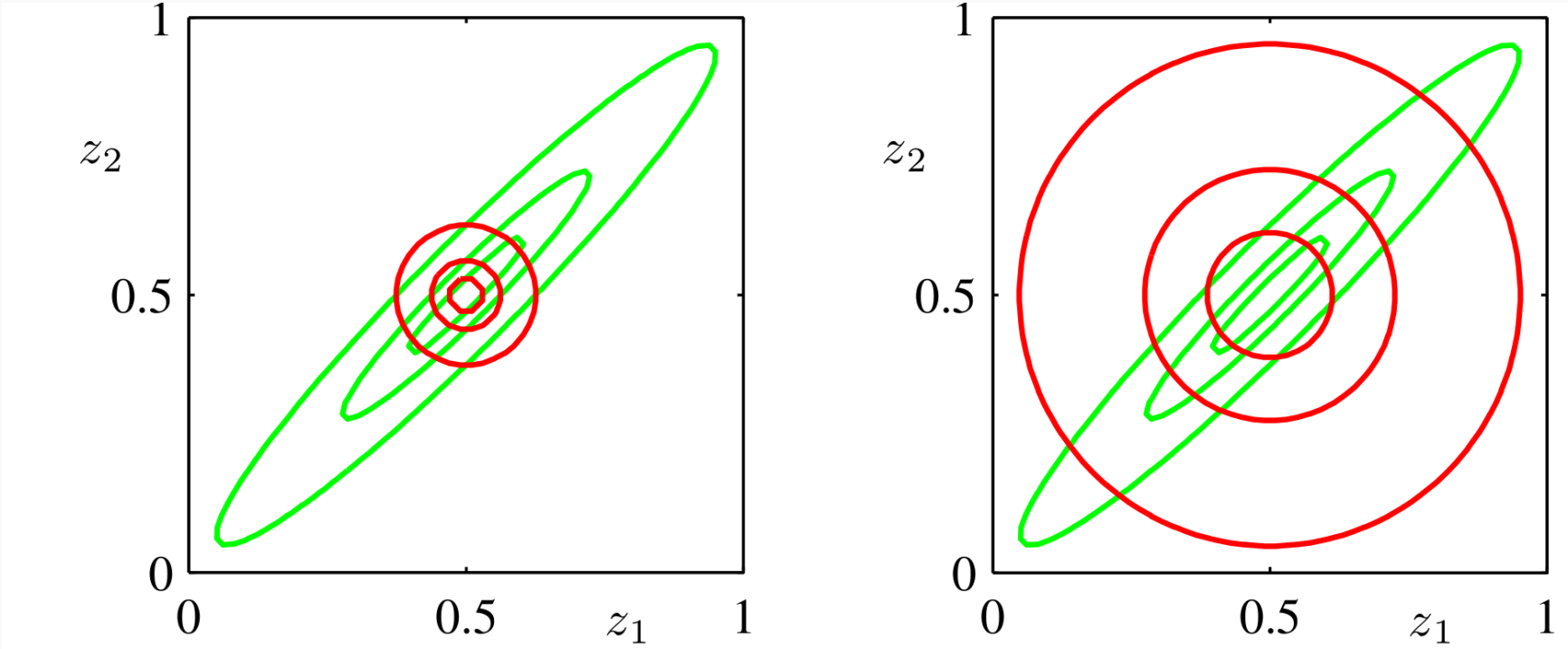
  и можно оптимизировать по отдельности:

$$q_j^*(\mathbf{Z}_j) = \int p(\mathbf{Z}) \prod_{i \neq j} \mathrm{d}\mathbf{Z}_i = p(\mathbf{Z}_j).$$

- Т.е. просто маргинализация.
- Почему бы так и не сделать? В чём разница?

Разные KL-дивергенции

- Разные дисперсии ответа:



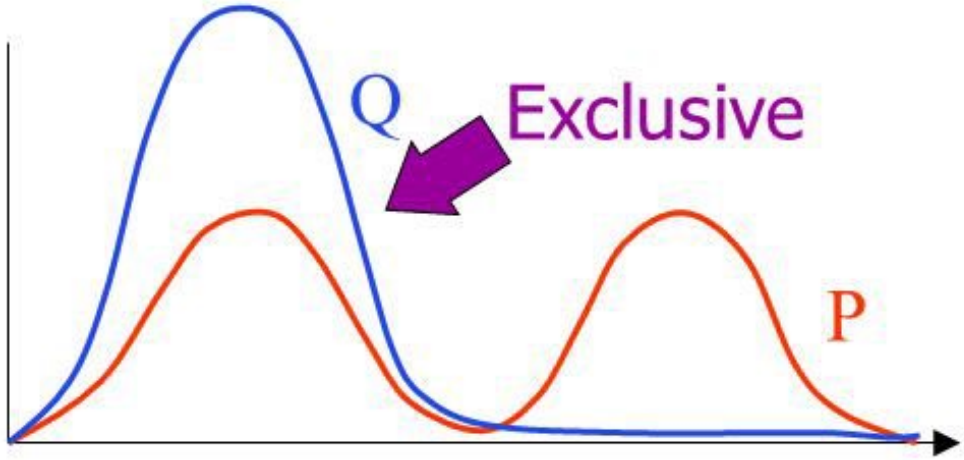
Разные KL-дивергенции

- Минимизация $\mathrm{KL}(p\|q)$ накрывает всю p, а $\mathrm{KL}(q\|p)$ строит всю q в регионе больших p:

Minimising

$\mathrm{KL}(Q||P)$

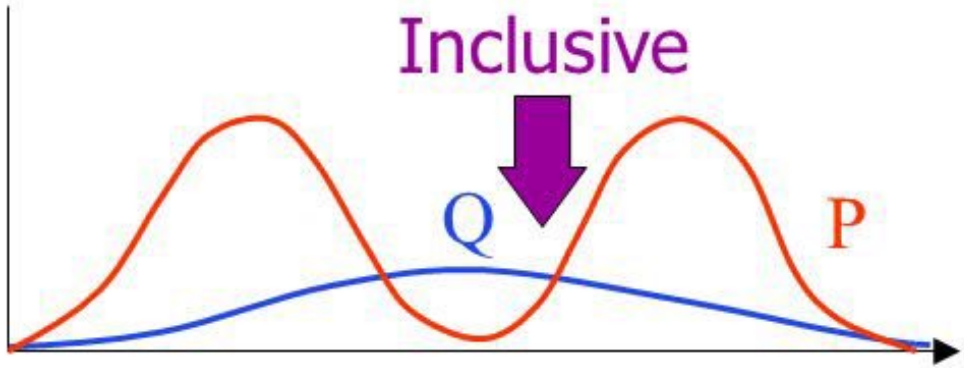
$$= \sum_H Q(H) \ln \frac{Q(H)}{P(H|V)}$$



Minimising

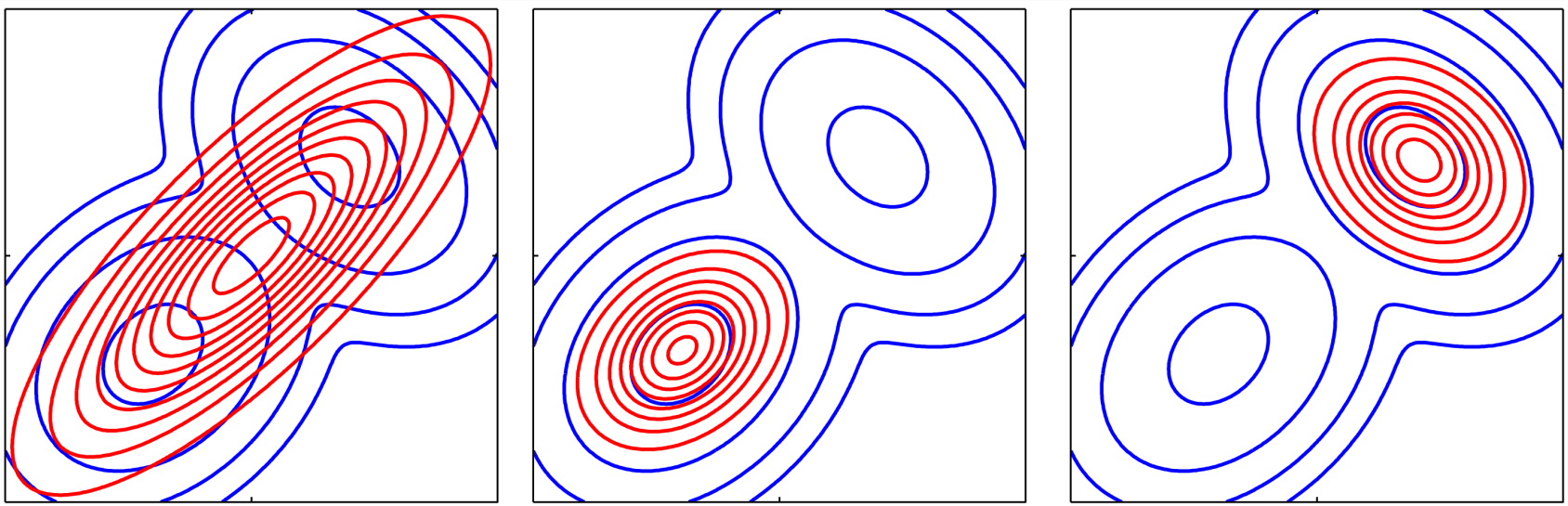
$\mathrm{KL}(P||Q)$

$$= \sum_H P(H|V) \ln \frac{P(H|V)}{Q(H)}$$



Разные KL-дивергенции

- Например, для двумерного гауссиана:



- В машинном обучении гораздо интереснее, конечно, пик найти.

Вариационное приближение для гауссиана

Одномерный гауссиан

- И ещё пример: давайте найдём параметры одномерного гауссиана по точкам $\mathbf{X} = \{x_1, \ldots, x_N\}$. Правдоподобие:

$$p(\mathbf{X} \mid \mu, \tau) = \left(\frac{\tau}{2\pi}\right)^{N/2} e^{-\frac{\tau}{2}\sum_{n=1}^{N}(x_n-\mu)^2}.$$

- Вводим сопряжённые априорные распределения:

$$p(\mu \mid \tau) = \mathcal{N}(\mu \mid \mu_0, (\lambda_0\tau)^{-1}),$$
$$p(\tau) = \mathrm{Gamma}(\tau \mid a_0, b_0).$$

- Мы это только что подсчитали точно, но давайте приблизим теперь апостериорное распределение как

$$q(\mu, \tau) = q_\mu(\mu)q_\tau(\tau).$$

- На самом деле так не раскладывается!
- Это то, что мы делали для $q(\mathbf{Z}) = \prod_{i=1}^{M} q_i(\mathbf{Z}_i)$. Посчитаем...

Одномерный гауссиан

- ...$q_\mu(\mu)$ – гауссиан с параметрами

$$\mu_N = \frac{\lambda_0\mu_0 + N\bar{x}}{\lambda_0 + N}, \quad \lambda_N = (\lambda_0 + N)\mathsf{E}[\tau].$$

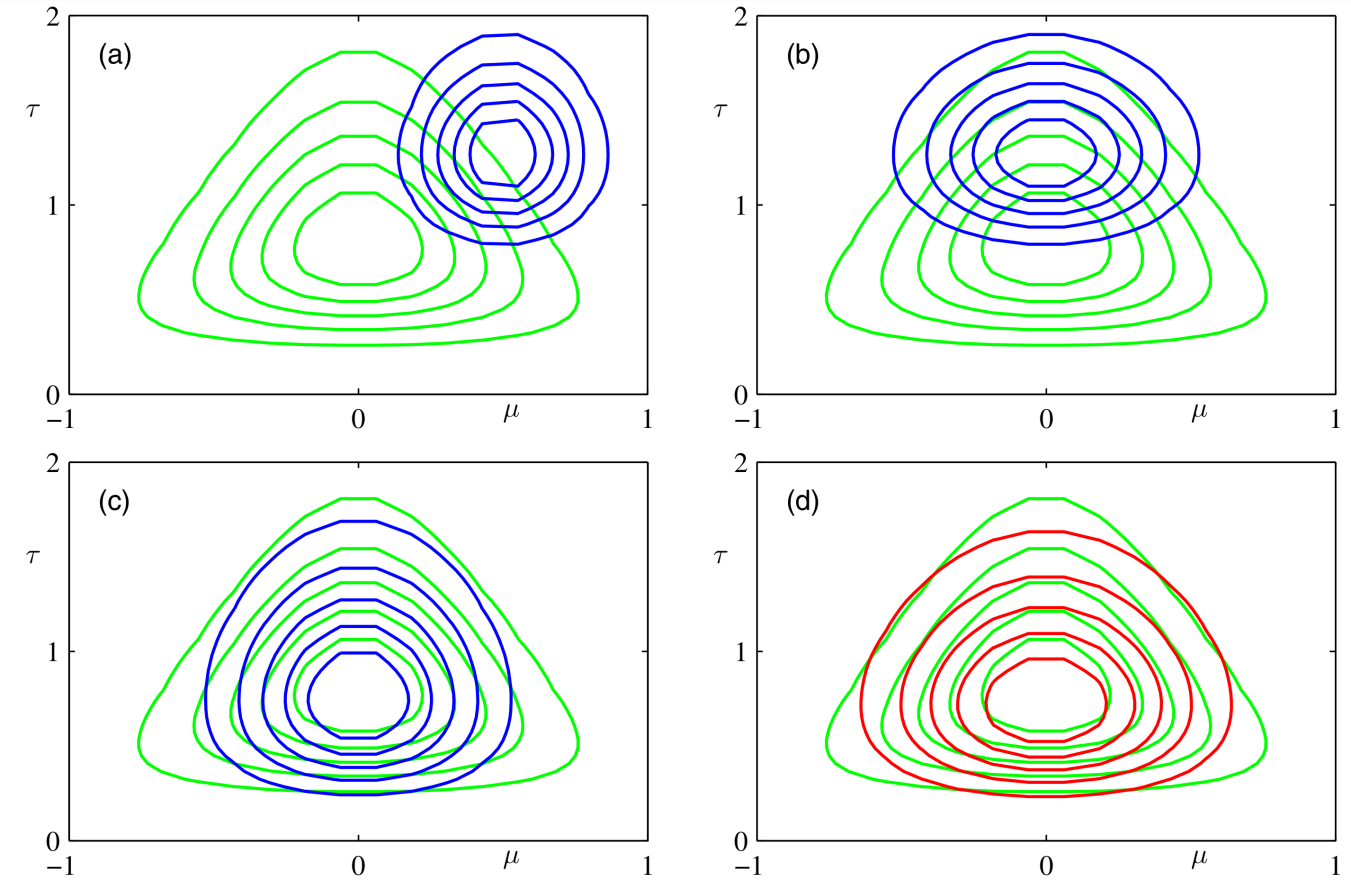
- А $q_\tau(\tau)$ – гамма-распределение с параметрами

$$a_N = a_0 + \frac{N+1}{2}, \quad b_N = b_0 + \frac{1}{2}\mathsf{E}_\mu\left[\sum_n (x_n - \mu)^2 + \lambda_0(\mu - \mu_0)^2\right].$$

- Всё получилось как надо, но без предположений о форме q_τ и q_μ.

Одномерный гауссиан

- Вот такой вывод в пространстве (μ, τ):



- А для $\mu_0 = a_0 = b_0 = \lambda_0 = 0$ (non-informative priors) можно и точно посчитать...

Одномерный гауссиан

- Получатся моменты для μ

$$\mathsf{E}[\mu] = \bar{x}, \quad \mathsf{E}[\mu^2] = \bar{x}^2 + \frac{1}{N\mathsf{E}[\tau]}.$$

- Это можно подставить и найти $\mathsf{E}[\tau]$:

$$\frac{1}{\mathsf{E}[\tau]} = \frac{1}{N}\sum_{n=1}^{N}(x_n - \bar{x})^2.$$

Наивный байесовский классификатор

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

- Но в деталях реализации наивного байесовского классификатора есть тонкости.
- 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

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

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

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

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

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

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

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

Как можно обобщить наивный байес

- 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?

Кластеризация

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

Тематическое моделирование

Как ещё можно обобщить наивный байес

- 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.

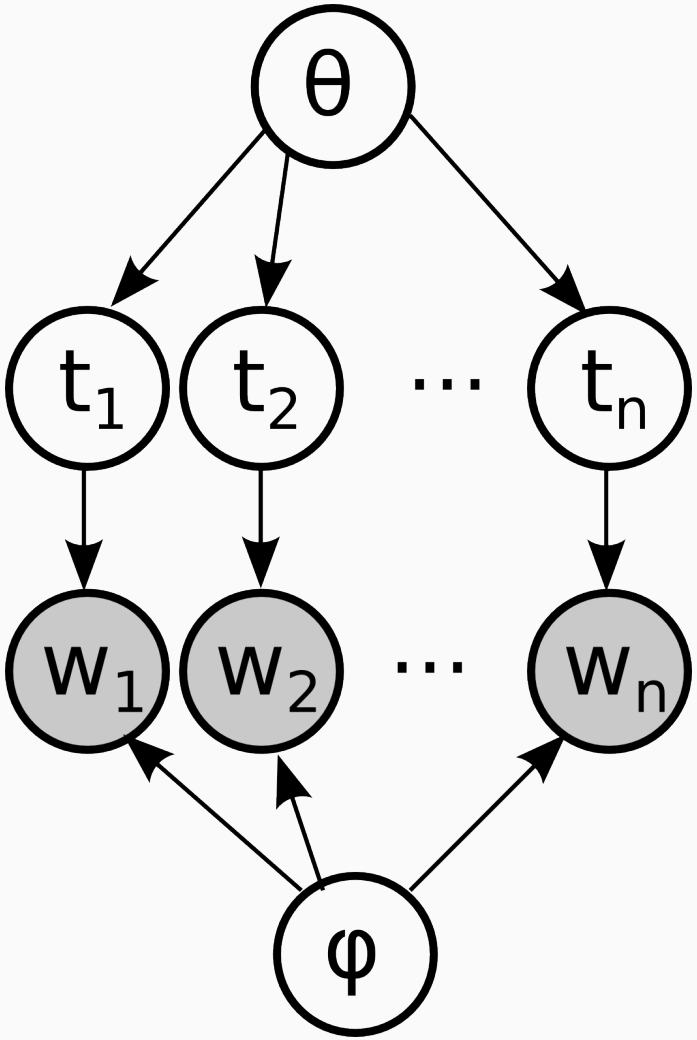
pLSA

- 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: $p(w \mid d, t) = p(w \mid d)$;
  - итого 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).

pLSA: графическая модель документа



Пример

- Получается как-то так:

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

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

повторять
  обнулить n_{wt}, n_{td}, n_t для всех $d \in D,\ w \in W,\ t \in T$;
  для всех $d \in D,\ w \in d$
    $n_{tdw} := n_{dw}\varphi_{wt}\theta_{td} \,/\, \sum_\tau \varphi_{w\tau}\theta_{\tau d}$ для всех $t \in T$;
    увеличить n_{wt}, n_{td}, n_t на n_{tdw} для всех $t \in T$;
  $\varphi_{wt} := n_{wt}/n_t$ для всех $w \in W,\ t \in T$;
  $\theta_{td} := n_{td}/n_d$ для всех $d \in D,\ t \in T$;
пока Φ и Θ не сойдутся;

pLSA

- 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?

pLSA

- EM-алгоритмом. На E-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 M-шаге пересчитываем параметры модели:

$$n_{wt} = \sum_d n_{dwt}, \qquad n_t = \sum_w n_{wt}, \quad \phi_{wt} = \frac{n_{wt}}{n_t},$$
$$n_{td} = \sum_{w \in d} n_{dwt}, \quad \theta_{td} = \frac{n_{td}}{n_d}.$$

- Вот и весь вывод в pLSA.

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 EM-алгоритм для тематической модели (2).

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

повторять
  обнулить n_{wt}, n_{td}, n_t для всех $d \in D,\ w \in W,\ t \in T$;
  для всех $d \in D,\ w \in d$
    $n_{tdw} := n_{dw}\varphi_{wt}\theta_{td} \,/\, \sum_\tau \varphi_{w\tau}\theta_{\tau d}$ для всех $t \in T$;
    увеличить n_{wt}, n_{td}, n_t на n_{tdw} для всех $t \in T$;
  $\varphi_{wt} := n_{wt}/n_t$ для всех $w \in W,\ t \in T$;
  $\theta_{td} := n_{td}/n_d$ для всех $d \in D,\ t \in T$;
пока Φ и Θ не сойдутся;

pLSA

- 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?

pLSA

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

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

pLSA

- 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.

pLSA

- 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

LDA

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

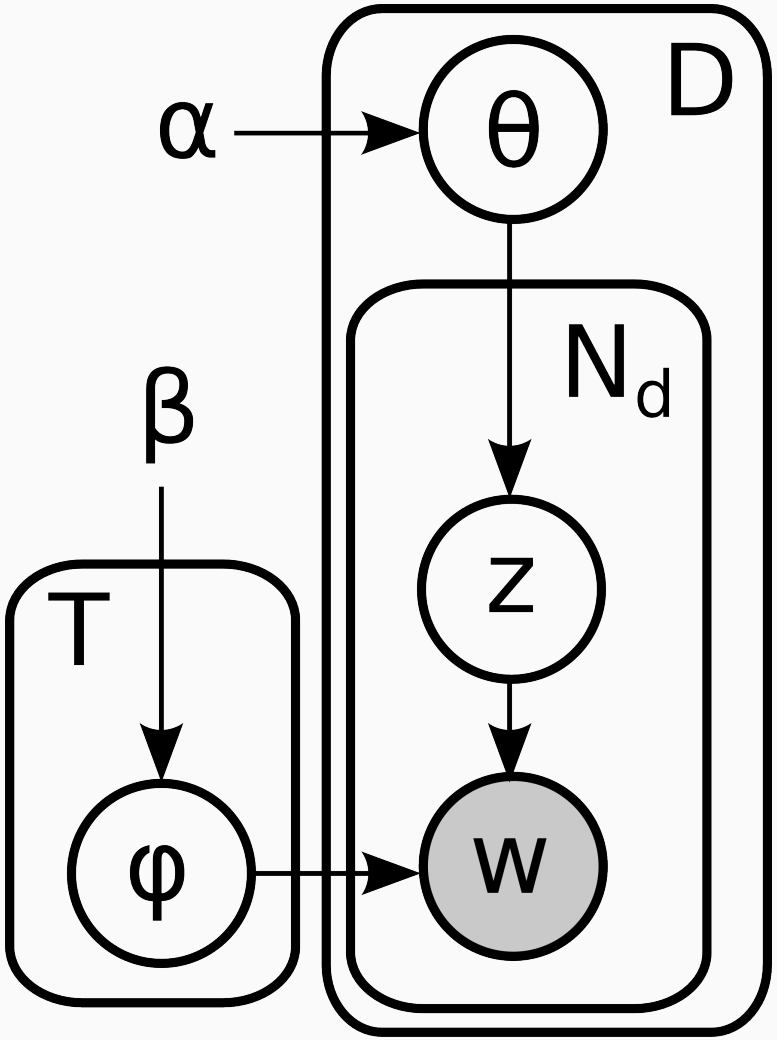
LDA

- 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.

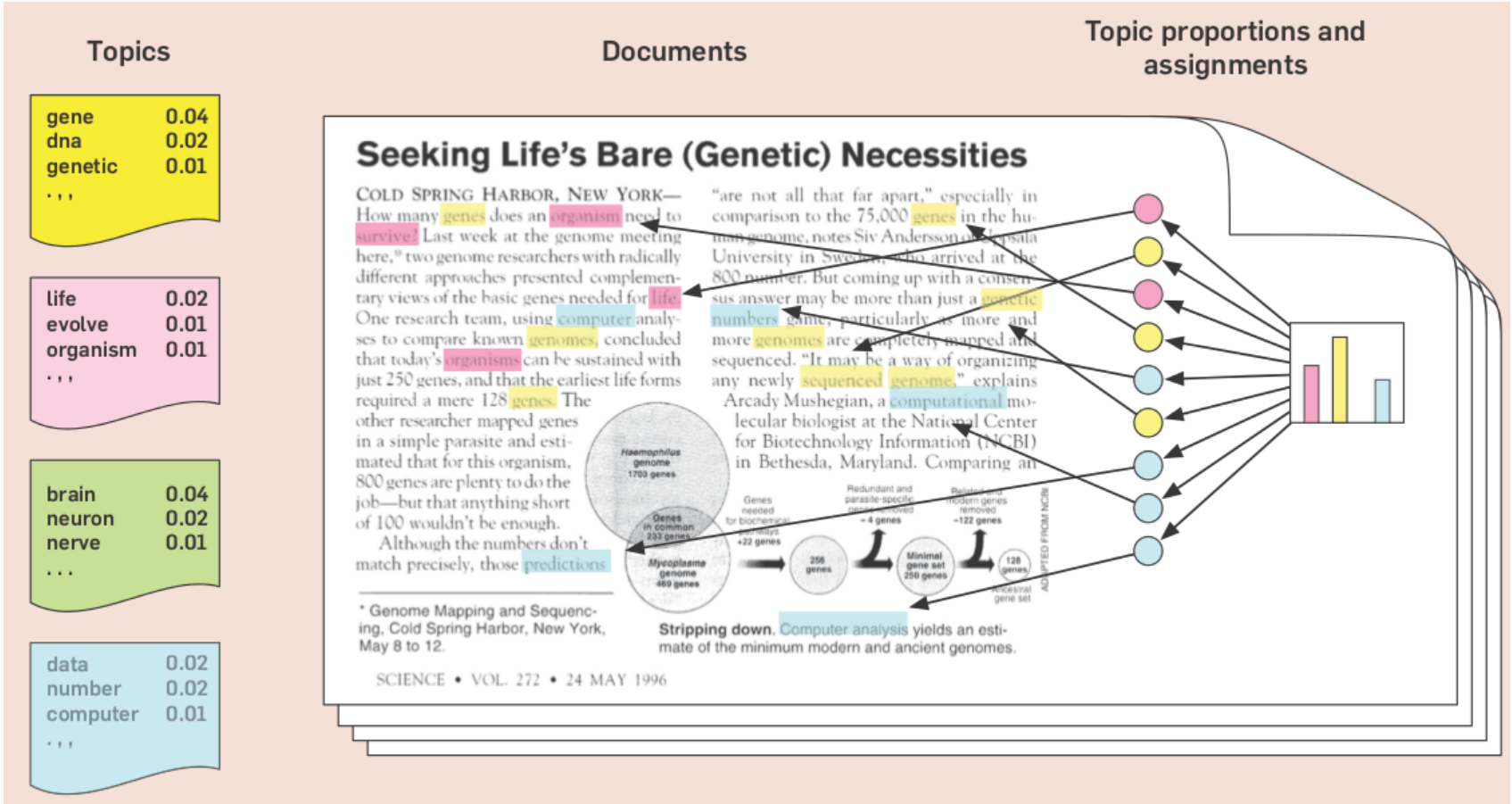
LDA

- 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 $\mathcal{D} = \{\mathbf{w}_d \mid d = 1..M\}$.
- Порождающая модель LDA выглядит так:
  - выбрать $\theta \sim \mathrm{Di}(\alpha)$;
  - для каждого из N слов w_n:
    - выбрать тему $z_n \sim \mathrm{Mult}(\theta)$;
    - выбрать слово $w_n \sim p(w_n \mid z_n, \beta)$ по мультиномиальному распределению.

LDA: графическая модель



LDA: что получается [Blei, 2012]



LDA: вывод

- 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.

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.

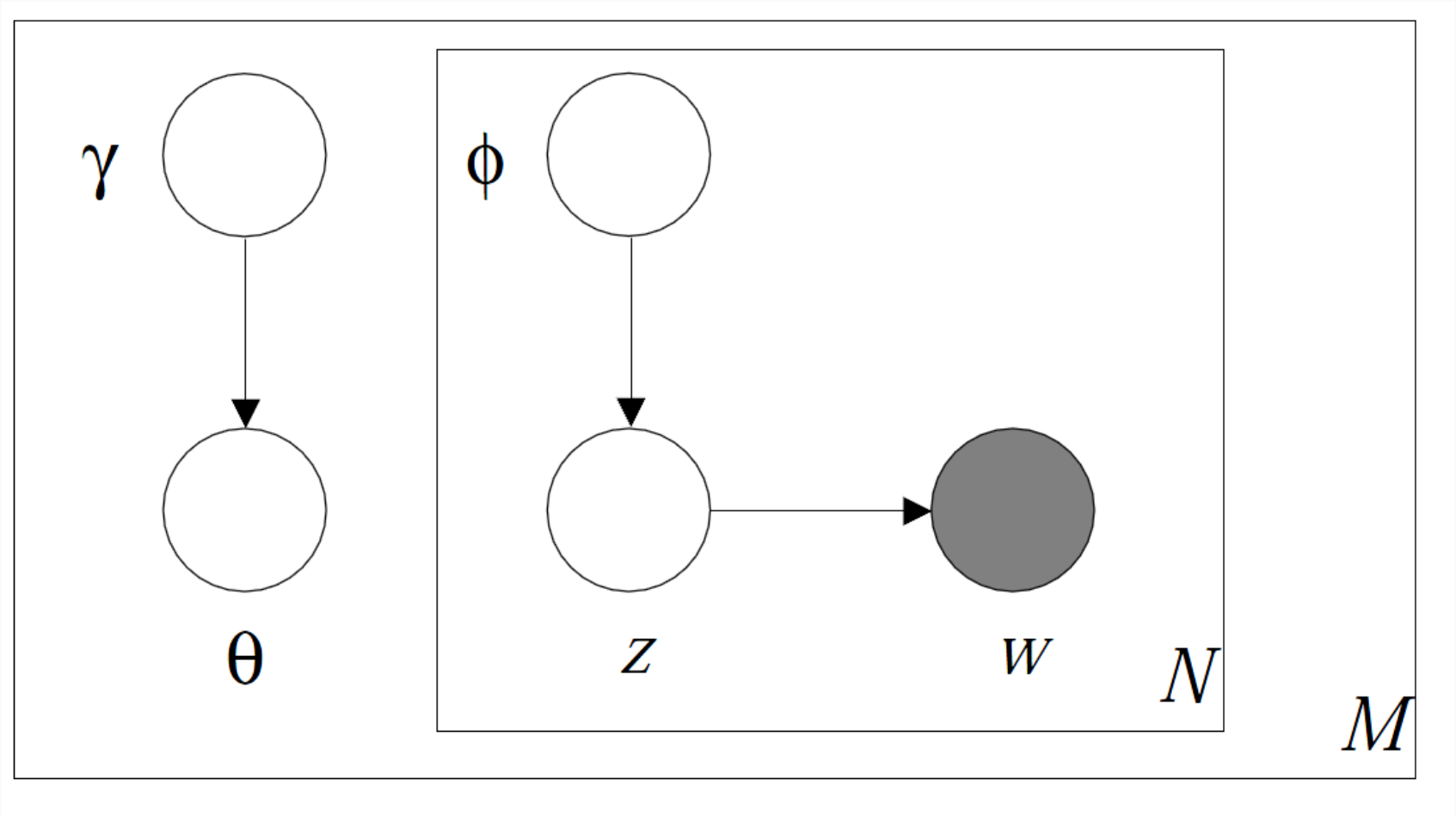
Вывод в LDA

- 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



LDA: вариационный вывод

- Теперь можно искать минимум KL-расстояния:

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

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

$$\log p(\mathsf{w} \mid \alpha, \beta) = \log \int_\theta \sum_{\mathsf{z}} p(\theta, \mathsf{z}, \mathsf{w} \mid \alpha, \beta) d\theta =$$

$$= \log \int_\theta \sum_{\mathsf{z}} \frac{p(\theta, \mathsf{z}, \mathsf{w} \mid \alpha, \beta) q(\theta, \mathsf{z})}{q(\theta, \mathsf{z})} d\theta \geq$$

$$\geq E_q \left[\log p(\theta, \mathsf{z}, \mathsf{w} \mid \alpha, \beta)\right] - E_q \left[\log q(\theta, \mathsf{z})\right] =: \mathcal{L}(\gamma, \phi; \alpha, \beta).$$

LDA: вариационный вывод

- Распишем произведения:

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

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

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

  где $\Psi(x) = \frac{\Gamma'(x)}{\Gamma(x)}$ – дигамма-функция.
- Теперь можно выписать каждый из пяти членов.

LDA: вариационный вывод

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

LDA: вариационный вывод

- 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:

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

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

LDA: вариационный вывод

- 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.

LDA: оценка параметров

- Теперь давайте попробуем оценить параметры α и β по корпусу документов $\mathcal{D}$.
- 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 $\mathcal{L}(\gamma, \phi; \alpha, \beta)$, т.к.

$$p(\mathbf{w}_d \mid \alpha, \beta) = \\ = \mathcal{L}(\gamma, \phi; \alpha, \beta) + \mathrm{KL}(q(\theta, z \mid \mathbf{w}_d, \gamma\phi) \| p(\theta, \mathbf{z} \mid \mathbf{w}_d, \alpha, \beta)).$$

LDA: оценка параметров

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

LDA: оценка параметров

- 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.

LDA: сэмплирование по Гиббсу

- В базовой модели LDA сэмплирование по Гиббсу после несложных преобразований сводится к так называемому *сжатому сэмплированию по Гиббсу* (collapsed Gibbs sampling), где переменные z_w итеративно сэмплируются по следующему распределению:

$$p(z_w = t \mid \mathbf{z}_{-w}, \mathbf{w}, \alpha, \beta) \propto q(z_w, t, \mathbf{z}_{-w}, \mathbf{w}, \alpha, \beta) =$$

$$\frac{n^{(d)}_{-w,t} + \alpha}{\sum_{t' \in T} \left(n^{(d)}_{-w,t'} + \alpha \right)} \frac{n^{(w)}_{-w,t} + \beta}{\sum_{w' \in W} \left(n^{(w')}_{-w,t} + \beta \right)},$$

где $n^{(d)}_{-w,t}$ – число слов в документе d, выбранных по теме t, а $n^{(w)}_{-w,t}$ – число раз, которое слово w было порождено из темы t, не считая текущего значения z_w; заметим, что оба этих счётчика зависят от других переменных $\mathbf{z}_{-w}$.

LDA: сэмплирование по Гиббсу

- Из сэмплов затем можно оценить переменные модели

$$\theta_{d,t} = \frac{n^{(d)}_{-w,t} + \alpha}{\sum_{t' \in T} \left(n^{(d)}_{-w,t'} + \alpha \right)},$$

$$\phi_{w,t} = \frac{n^{(w)}_{-w,t} + \beta}{\sum_{w' \in W} \left(n^{(w')}_{-w,t} + \beta \right)},$$

где $\phi_{w,t}$ – вероятность получить слово w в теме t, а $\theta_{d,t}$ – вероятность получить тему t в документе d.

Варианты и расширения модели LDA

- В последние десять лет эта модель стала основой для множества различных расширений.
- Каждое из этих расширений содержит либо вариационный алгоритм вывода, либо алгоритм сэмплирования по Гиббсу для модели, которая, основываясь на LDA, включает в себя ещё и какую-либо дополнительную информацию или дополнительные предполагаемые зависимости.
- Обычно – или дополнительная структура на темах, или дополнительная информация.

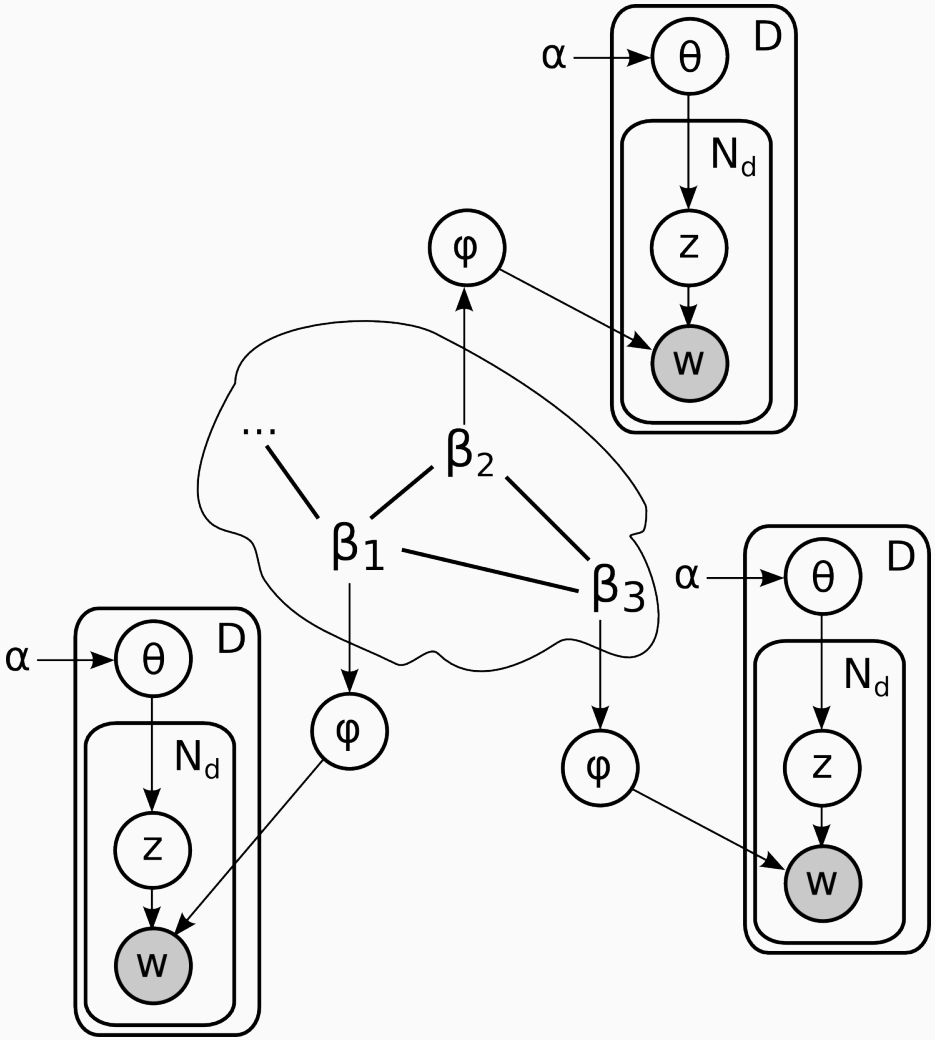
Коррелированные тематические модели

- В базовой модели LDA распределения слов по темам независимы и никак не скоррелированы; однако на самом деле, конечно, некоторые темы ближе друг к другу, многие темы делят между собой слова.
- Коррелированные тематические модели (correlated topic models, CTM); отличие от базового LDA здесь в том, что используется логистическое нормальное распределение вместо распределения Дирихле; логистическое нормальное распределение более выразительно, оно может моделировать корреляции между темами.
- Предлагается алгоритм вывода, основанный на вариационном приближении.

Марковские тематические модели

- Марковские тематические модели (Markov topic models, MTM): марковские случайные поля для моделирования взаимоотношений между темами в разных частях датасета (разных корпусах текстов).
- MTM состоит из нескольких копий гиперпараметров β_i в LDA, описывающих параметры разных корпусов с одними и теми же темами. Гиперпараметры β_i связаны между собой в марковском случайном поле (Markov random field, MRF).
- В результате тексты из i-го корпуса порождаются как в обычном LDA, используя соответствующее β_i.
- В свою очередь, β_i подчиняются априорным ограничениям, которые позволяют «делить» темы между корпусами, задавать «фоновые» темы, присутствующие во всех корпусах, накладывать ограничения на взаимоотношения между темами и т.д.

Марковские тематические модели



Реляционная тематическая модель

- Реляционная тематическая модель (relational topic model, RTM) – иерархическая модель, в которой отражён граф структуры сети документов.
- Генеративный процесс в RTM работает так:
  - сгенерировать документы из обычной модели LDA;
  - для каждой пары документов d_1, d_2 выбрать бинарную переменную y_{12}, отражающую наличие связи между d_1 и d_2:

$$y_{12} \mid \mathbf{z}_{d_1}, \mathbf{z}_{d_2} \sim \psi(\cdot \mid \mathbf{z}_{d_1}, \mathbf{z}_{d_2}, \boldsymbol{\eta}).$$

- В качестве ψ берутся разные сигмоидальные функции; разработан алгоритм вывода, основанный на вариационном приближении.

Модели, учитывающие время

- Ряд важных расширений LDA касается учёта трендов, т.е. изменений в распределениях тем, происходящих со временем.
- Цель – учёт времени, анализ «горячих» тем, анализ того, какие темы быстро становятся «горячими» и столь же быстро затухают, а какие проходят «красной нитью» через весь исследуемый временной интервал.

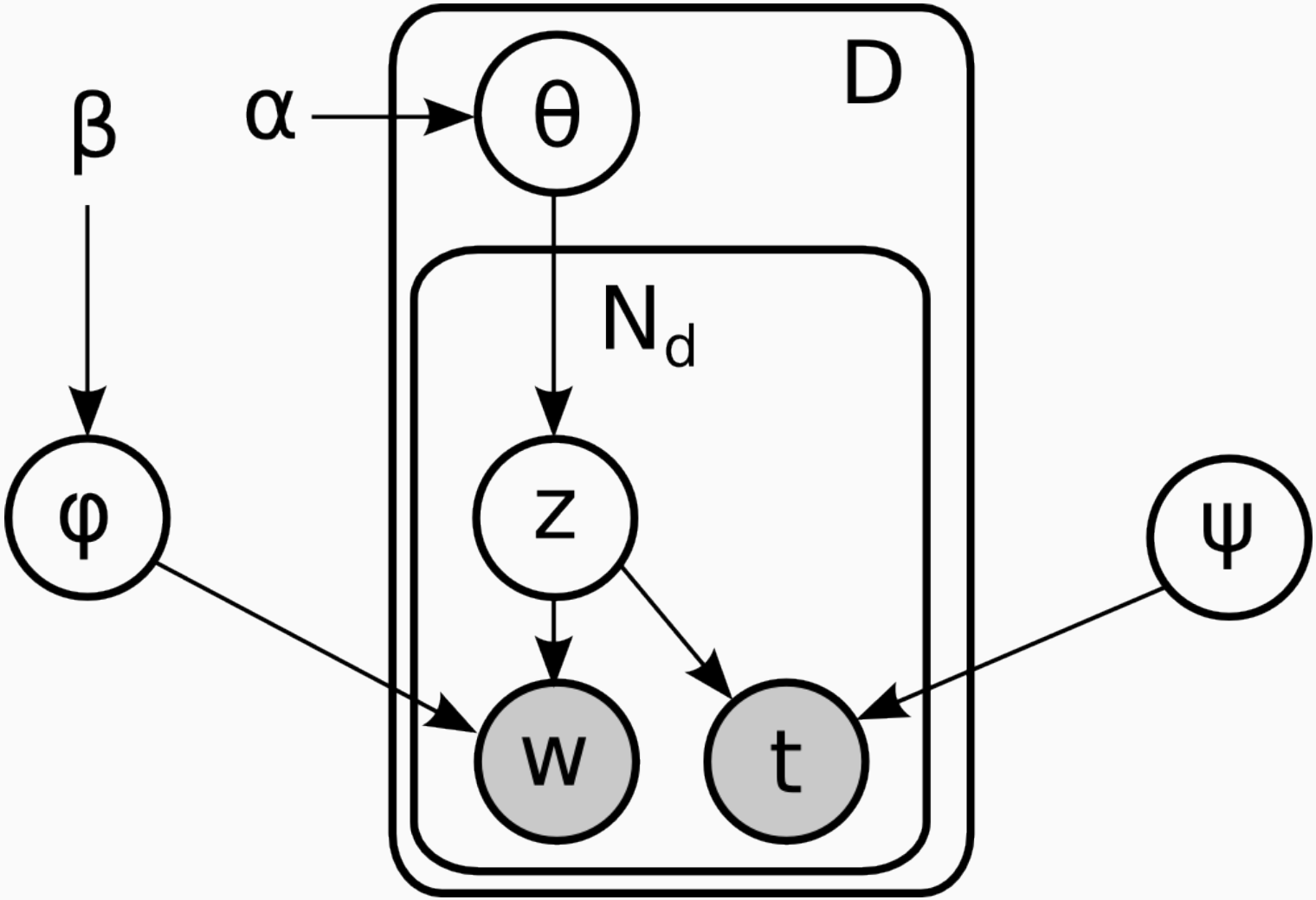
Topics over Time

- В модели TOT (Topics over Time) время предполагается непрерывным, и модель дополняется бета-распределениями, порождающими временные метки (timestamps) для каждого слова.
- Генеративная модель модели Topics over Time такова:
  - для каждой темы $z = 1..T$ выбрать мультиномиальное распределение ϕ_z из априорного распределения Дирихле β;
  - для каждого документа d выбрать мультиномиальное распределение θ_d из априорного распределения Дирихле α, затем для каждого слова $w_{di} \in d$:
    - выбрать тему z_{di} из θ_d;
    - выбрать слово w_{di} из распределения $\phi_{z_{di}}$;
    - выбрать время t_{di} из бета-распределения $\psi_{z_{di}}$.

Topics over Time

- Основная идея заключается в том, что каждой теме соответствует её бета-распределение ψ_z, т.е. каждая тема локализована во времени (сильнее или слабее, в зависимости от параметров ψ_z).
- Таким образом можно как обучить глобальные темы, которые всегда присутствуют, так и подхватить тему, которая вызвала сильный краткий всплеск, а затем пропала из виду; разница будет в том, что дисперсия ψ_z будет в первом случае меньше, чем во втором.

Topics over Time



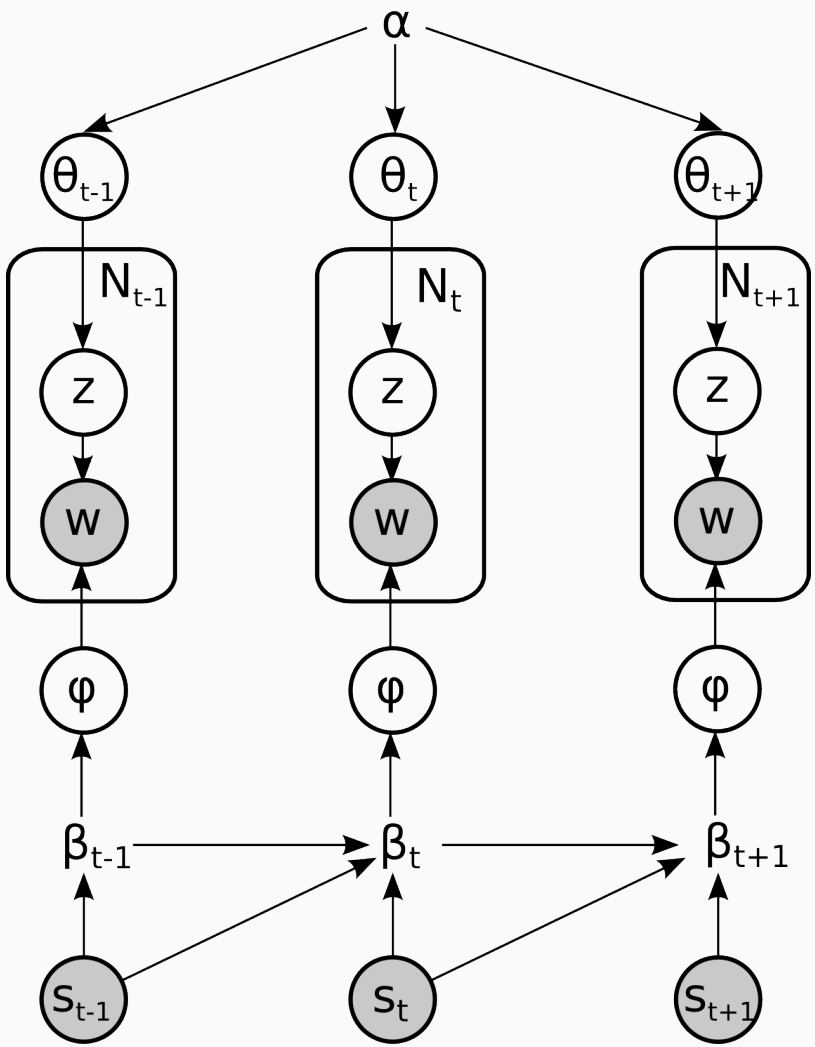
Динамические тематические модели

- *Динамические тематические модели* представляют временную эволюцию тем через эволюцию их гиперпараметров α и/или β.
- Бывают дискретные ([d]DTM), в которых время дискретно, и непрерывные, где эволюция гиперпараметра β (α здесь предполагается постоянным) моделируется посредством броуновского движения: для двух документов i и j (j позже i) верно, что

$$\beta_{j,k,w} \mid \beta_{i,k,w}, s_i, s_j \sim \mathcal{N}(\beta_{i,k,w}, v\Delta_{s_i,s_j}),$$

  где s_i и s_j – это отметки времени (timestamps) документов i и j, $\Delta(s_i, s_j)$ – интервал времени, прошедший между ними, v – параметр модели.
- В остальном генеративный процесс остаётся неизменным.

Непрерывная динамическая тематическая модель (cDTM)



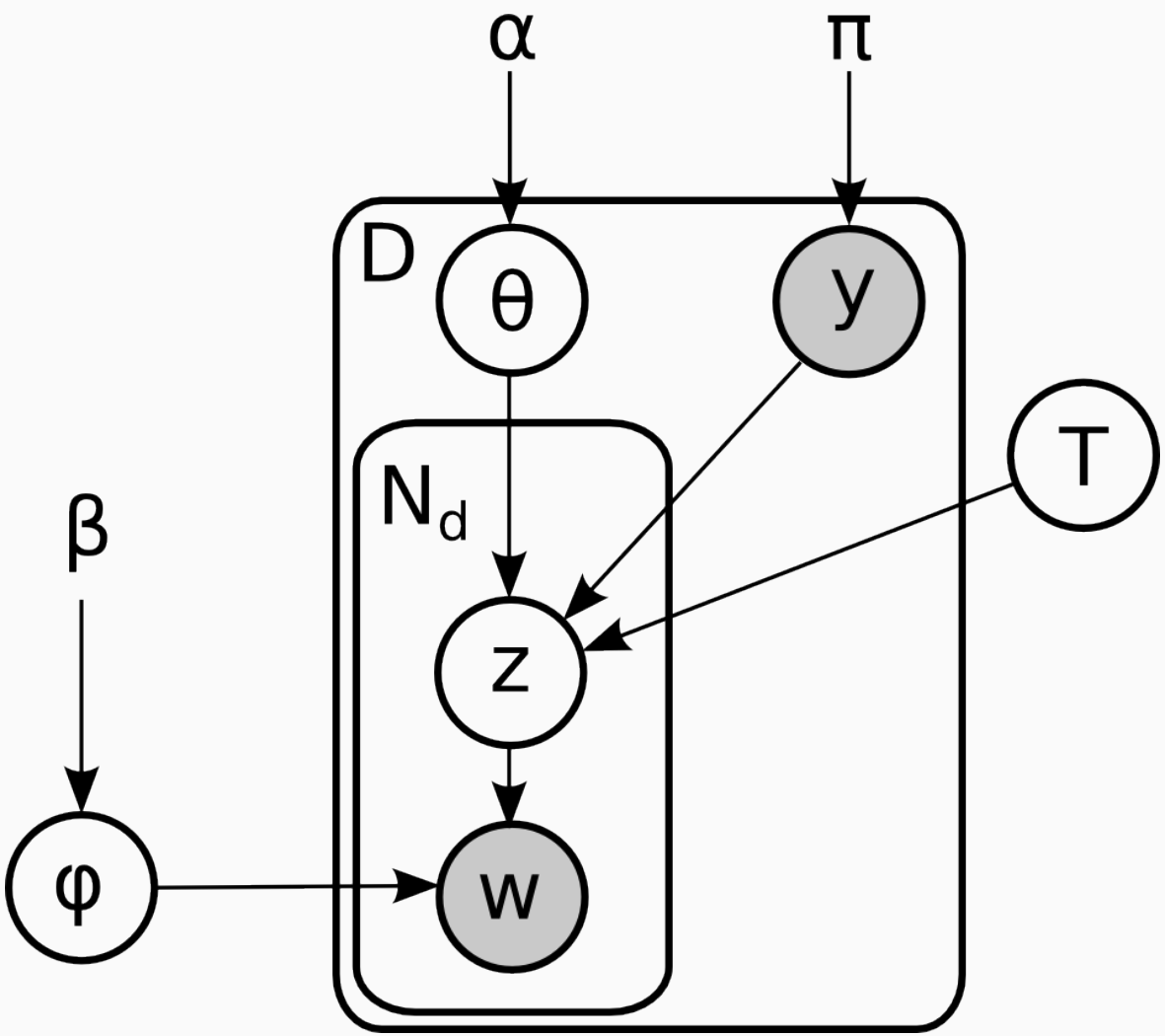
Supervised LDA

- Supervised LDA: документы снабжены дополнительной информацией, дополнительной переменной отклика (обычно известной).
- Распределение отклика моделируется обобщённой линейной моделью (распределением из экспоненциального семейства), параметры которой связаны с полученным в документе распределением тем.
- Т.е. в генеративную модель добавляется ещё один шаг: после того как темы всех слов известны,
  - сгенерировать переменную-отклик $y \sim \mathrm{glm}(\mathsf{z}, \eta, \delta)$, где z – распределение тем в документе, а η и δ – другие параметры glm.
- К примеру, в контексте рекомендательных систем дополнительный отклик может быть реакцией пользователя.

DiscLDA

- Дискриминативное LDA (DiscLDA), другое расширение модели LDA для документов, снабжённых категориальной переменной y, которая в дальнейшем станет предметом для классификации.
- Для каждой метки класса y в модели DiscLDA вводится линейное преобразование $T^y : \mathbb{R}^K \to \mathbb{R}^L_+$, которое преобразует K-мерное распределение Дирихле θ в смесь L-мерных распределений Дирихле $T^y\theta$.
- В генеративной модели меняется только шаг порождения темы документа z: вместо того чтобы выбирать z по распределению θ, сгенерированному для данного документа,
    - сгенерировать тему z по распределению $T^y\theta$, где T^y – преобразование, соответствующее метке данного документа y.

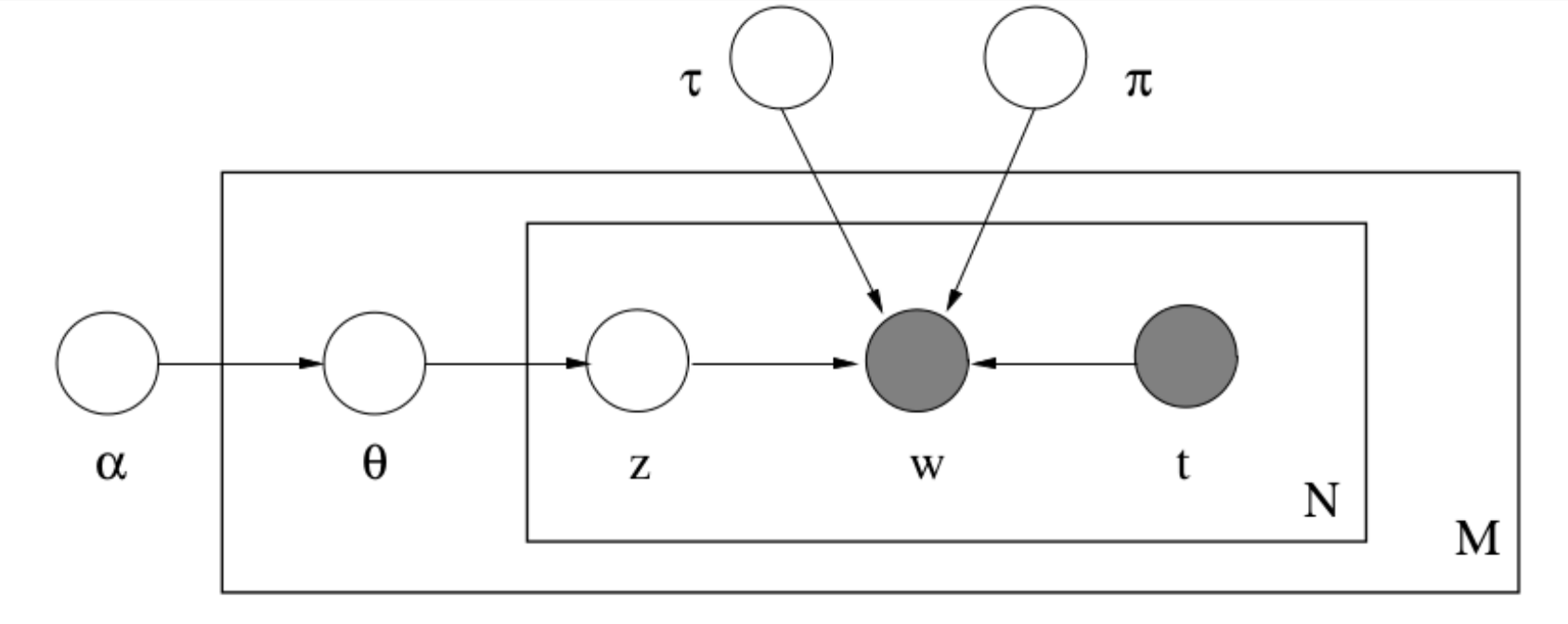
DiscLDA



TagLDA

- TagLDA: слова имеют теги, т.е. документ не является единым мешком слов, а состоит из нескольких мешков, и в разных мешках слова отличаются друг от друга.
- Например, у страницы может быть название – слова из названия важнее для определения темы, чем просто из текста. Или, например, теги к странице, поставленные человеком – опять же, это слова гораздо более важные, чем слова из текста.
- Математически разница в том, что теперь распределения слов в темах – это не просто мультиномиальные дискретные распределения, они факторизованы на распределение слово-тема и распределение слово-тег.

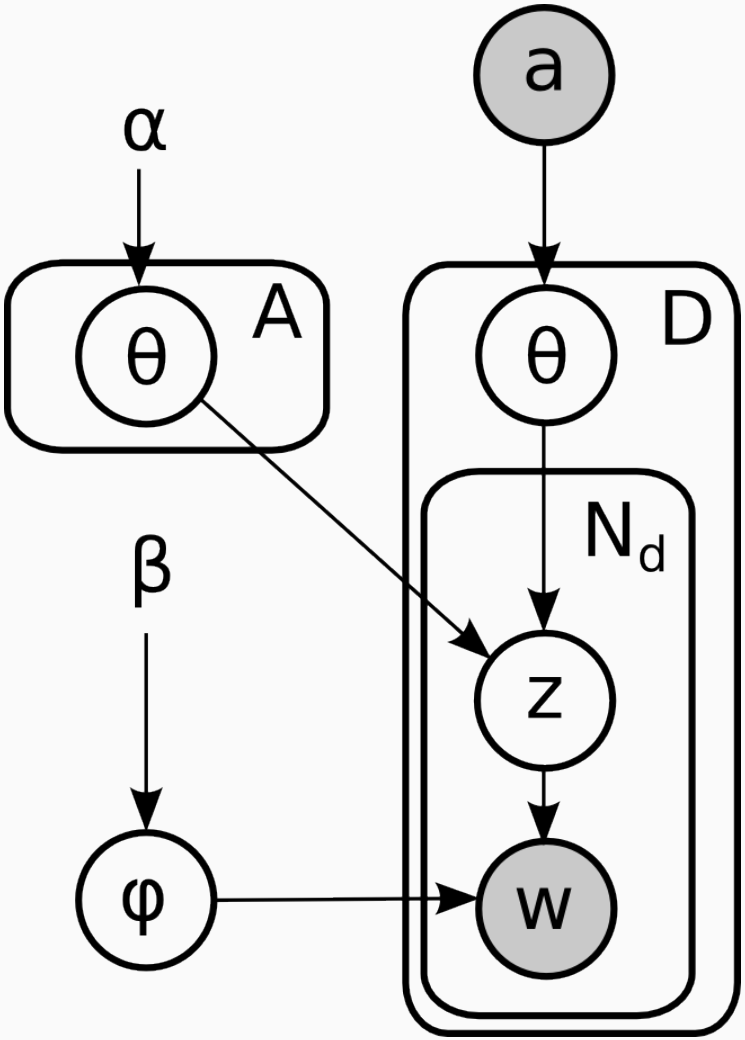
TagLDA



Author-Topic model

- Author-Topic modeling: кроме собственно текстов, присутствуют их авторы; или автор тоже представляется как распределение на темах, на которые он пишет, или тексты одного автора даже на разные темы будут похожи.
- Базовая генеративная модель Author-Topic model (остальное как в базовом LDA):
  - для каждого слова w:
    - выбираем автора x для этого слова из множества авторов документа $\boldsymbol{a}_d$;
    - выбираем тему из распределения на темах, соответствующего автору x;
    - выбираем слово из распределения слов, соответствующего этой теме.

Author-Topic model



Author-Topic model

- Алгоритм сэмплирования, соответствующий такой модели, является вариантом сжатого сэмплирования по Гиббсу:

$$p(z_w = t, x_w = a \mid \mathbf{z}_{-w}, \mathbf{x}_{-w}, \mathbf{w}, \alpha, \beta) \propto$$

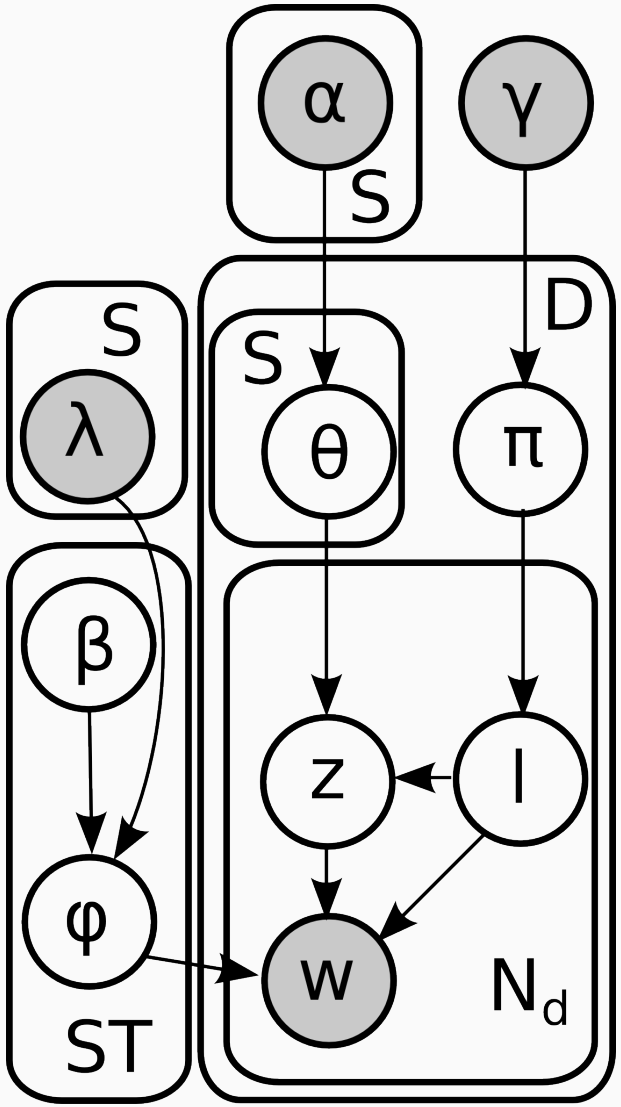
$$\propto \frac{n^{(a)}_{-a,t} + \alpha}{\sum_{t' \in T} \left(n^{(a)}_{-w,t'} + \alpha \right)} \frac{n^{(w)}_{-w,t} + \beta}{\sum_{w' \in W} \left(n^{(w')}_{-w,t} + \beta \right)},$$

  где $n^{(a)}_{-a,t}$ – то, сколько раз автору a соответствовала тема t, не считая текущего значения x_w, а $n^{(w)}_{-w,t}$ – число раз, которое слово w было порождено из темы t, не считая текущего значения z_w; заметим, что оба этих счётчика зависят от других переменных $\mathbf{z}_{-w}$, $\mathbf{x}_{-w}$.
- Давайте теперь чуть подробнее разберём тематические модели с сентиментом...

Joint Sentiment-Topic

- JST: темы зависят от тональностей из распределения π_d документа, слова зависят от пар тональность-тема.
- Порождающий процесс – для каждой позиции слова j:

  (1) выберем метку тональности $l_j \sim \mathrm{Mult}(\pi_d)$;
  (2) выберем тему $z_j \sim \mathrm{Mult}(\theta_{d,l_j})$;
  (3) выберем слово $w \sim \mathrm{Mult}(\phi_{l_j,z_j})$.



Joint Sentiment-Topic

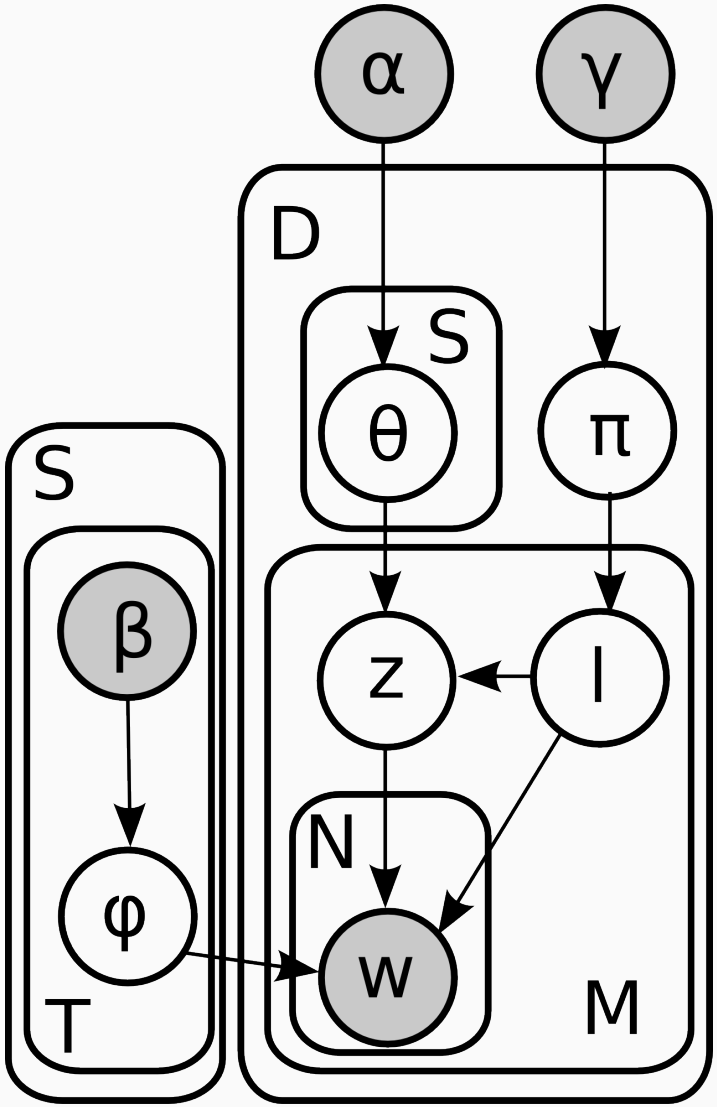
- В сэмплировании по Гиббсу можно выинтегрировать π_d:

$$p(z_j = t, l_j = k \mid \mathbf{z}_{-j}, \mathbf{w}, \alpha, \beta, \gamma, \lambda) \propto$$
$$\frac{n^{\neg j}_{*,k,t,d} + \alpha_{tk}}{n^{\neg j}_{*,k,*,d} + \sum_t \alpha_{tk}} \cdot \frac{n^{\neg j}_{w,k,t,*} + \beta_{kw}}{n^{\neg j}_{*,k,t,*} + \sum_w \beta_{kw}} \cdot \frac{n^{\neg j}_{*,k,*,d} + \gamma}{n^{\neg j}_{*,*,*,d} + S\gamma},$$

где $n_{w,k,t,d}$ — число слов w, порождённых темой t и меткой тональности k в документе d, α_{tk} — априорное распределение Дирихле для темы t с меткой тональности k.

Aspect and Sentiment Unification Model

- ASUM: aspects + sentiment для обзоров пользователей; разбиваем обзор на предложения, предполагая, что в каждом предложении один аспект.
- Базовая модель – Sentence LDA (SLDA): для каждого отзыва d с распределением θ_d, для каждого предложения в d,
  (1) выбираем метку тональности $l_s \sim \mathrm{Mult}(\pi_d)$,
  (2) выбираем тему $t_s \sim \mathrm{Mult}(\theta_{dl_s})$ при условии тональности l_s,
  (3) порождаем слова $w \sim \mathrm{Mult}(\phi_{l_s t_s})$.



Сэмплирование по Гиббсу в ASUM

- Обозначим через $s_{k,t,d}$ число предложений (а не слов), которым присвоена тема t и метка t в документе d:

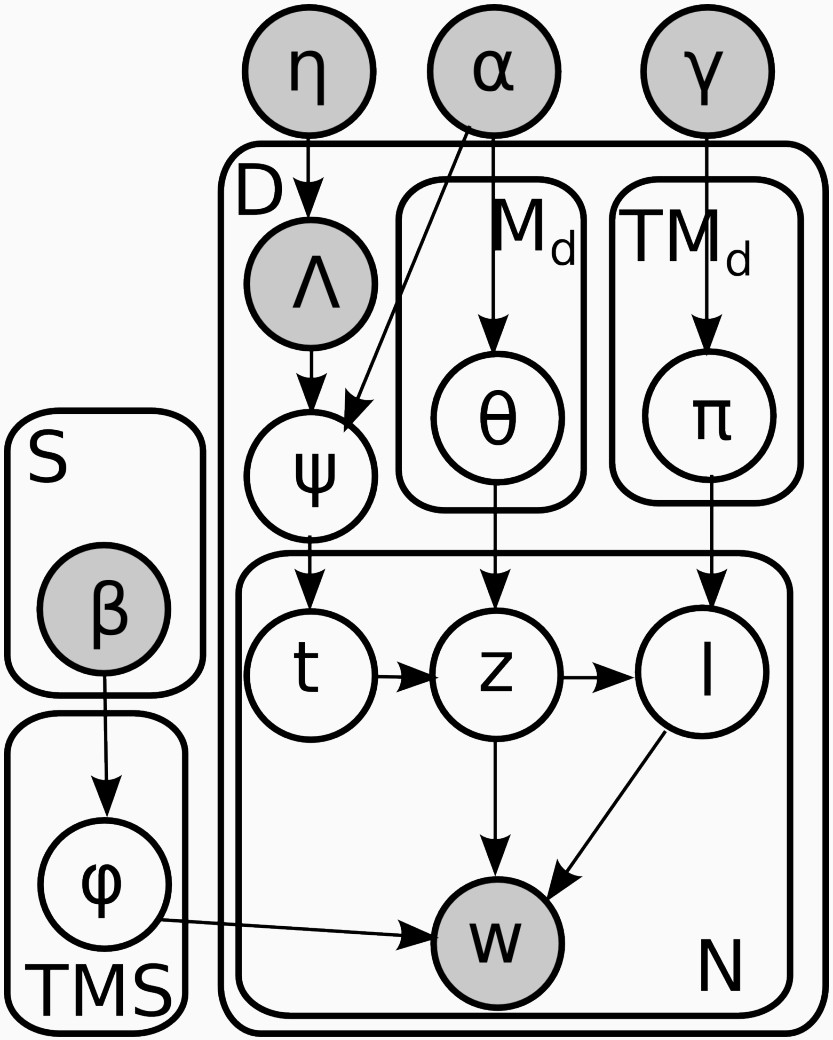
$$p(z_j = t, l_j = k \mid \mathbf{l}_{-j}, \mathbf{z}_{-j}, \mathbf{w}, \boldsymbol{\gamma}, \boldsymbol{\alpha}, \boldsymbol{\beta}) \propto$$
$$\frac{s^{\neg j}_{k,t,d} + \alpha_t}{s^{\neg j}_{k,*,d} + \sum_t \alpha_t} \cdot \frac{s^{\neg j}_{k,*,d} + \gamma_k}{s^{\neg j}_{*,*,d} + \sum_{k'} \gamma_{k'}} \times$$
$$\times \frac{\Gamma\left(n^{\neg j}_{*,k,t,*} + \sum_w \beta_{kw}\right)}{\Gamma\left(n^{\neg j}_{*,k,t,*} + \sum_w \beta_{kw} + W_{*,j}\right)} \prod_w \frac{\Gamma\left(n^{\neg j}_{w,k,t,*} + \beta_{kw} + W_{w,j}\right)}{\Gamma\left(n^{\neg j}_{w,k,t,*} + \beta_{kw}\right)},$$

где $W_{w,j}$ – число слов w в предложении j.

User-Aware Sentiment Topic Models

- USTM: добавим ещё метаданные/теги для пользователя (место, пол, возраст и т.п.) к темам и тональностям.
- Каждый документ снабжён комбинацией тегов, темы порождаются при условии тегов, тональности при условии троек (документ, тег, тема), слова при условии тем, тональностей и тегов.
- Формально, распределение тегов ψ_d порождается для каждого документа (с априорным распределением Дирихле с параметром η), для каждой позиции j порождаем тег $a_j \sim \mathrm{Mult}(\psi_d)$ из ψ_d, а распределения тем, тональностей и слов будут условными по тегу a_j.

Графическая модель USTM



Сэмплирование по Гиббсу для USTM

- Обозначим через $n_{w,k,t,m,d}$ число слов w, порождённых темой t, меткой тональности k и тегом метаданных m в документе d; тогда

$$p(z_j = t, l_j = k, a_j = m \mid \mathbf{l}_{-j}, \mathbf{z}_{-j}, \boldsymbol{a}_{-j}, \mathbf{w}, \boldsymbol{\gamma}, \boldsymbol{\alpha}, \boldsymbol{\beta}) \propto$$
$$\frac{n^{\neg j}_{*,*,t,m,d} + \alpha}{n^{\neg j}_{*,*,*,m,d} + TM_d\alpha} \cdot \frac{n^{\neg j}_{w,*,t,m,*} + \beta}{n^{\neg j}_{*,*,t,m,*} + W\beta} \cdot$$
$$\frac{n^{\neg j}_{w,k,t,m,*} + \beta_{wk}}{n^{\neg j}_{*,k,t,m,*} + \sum_w \beta_{wk}} \cdot \frac{n^{\neg j}_{*,k,t,m,d} + \gamma}{n^{\neg j}_{*,*,t,m,d} + S\gamma},$$

где M_d — число тегов в документе d.

Примеры тем

#	sent.	sentiment words
1	neu	соус, салат, кусочек, сыр, тарелка, овощ, масло, лук, перец
	pos	приятный, атмосфера, уютный, вечер, музыка, ужин, романтический
	neg	ресторан, официант, внимание, сервис, обращать, обслуживать, уровень
2	neu	столик, заказывать, вечер, стол, приходить, место, заранее, встречать
	pos	место, хороший, вкус, самый, приятный, вполне, отличный, интересный
	neg	еда, вообще, никакой, заказывать, оказываться, вкус, ужасный, ничто
3	neu	девушка, спрашивать, вопрос, подходить, официантка, официант, говорить
	pos	большой, место, выбор, хороший, блюдо, цена, порция, небольшой, плюс
	neg	цена, обслуживание, качество, уровень, кухня, средний, ценник, высоко

Примеры окрашенных слов для разных аспектов

aspect	sentiment words
баранина	вкусный, сытный, аппетитный, душистый, деликатесный, сладкий, ароматный, черствый, ароматичный, пресный
караоке	музыкальный, попсовый, классно, развлекательный, улетный
пирог	вкусный, аппетитный, обсыпной, сытный, черствый, ароматный, сладкий
ресторан	шикарный, фешенебельный, уютный, люкс, роскошный, недорогой, шикарно, престижный, модный, развлекательный,
вывеска	обветшалый, выцветший, аляповатый, фешенебельный, фанерный, респектабельный, помпезный, ржавый
администратор	люкс, неисполнительный, ответственный, компетентный, толстяк, высококвалифицированный, высококлассный, толстяк
интерьер	уют, уютны, стильный, просторный, помпезный, роскошный, шикарный, шикарный, мрачноватый, комфортабельный
вежливый	вежливый, учтивы, обходительный, доброжелательный, тактичный

Спасибо!

Спасибо за внимание!