# Непараметрические байесовские методы: Китайский ресторан и индийский буфет

Сергей Николенко

Mail.Ru, 18 декабря 2013 г.

# Outline

# Суть лекции

- *Кластеризация* — типичная задача статистического анализа: задача классификации объектов одной природы в несколько групп так, чтобы объекты в одной группе обладали одним и тем же свойством.
- Под свойством обычно понимается близость друг к другу относительно выбранной метрики.

# Чуть более формально

- Есть набор тестовых примеров $X = \{x_1, \ldots, x_n\}$ и функция расстояния между примерами $\rho$.
- Требуется разбить $X$ на непересекающиеся подмножества (кластеры) так, чтобы каждое подмножество состояло из похожих объектов, а объекты разных подмножеств существенно различались.

# Методы

- Иерархическая кластеризация:
  - инициализируем каждую точку как отдельный кластер;
  - объединяем ближайшие точки, далее считаем их единым кластером;
  - повторяем; в результате получается дерево.
- Методами теории графов:
  - находим минимальное остовное дерево в графе из расстояний между точками (алгоритм Краскала, алгоритм Борувки);
  - выкидываем сколько нужно самых длинных рёбер.

# Алгоритм $k$-средних

- Вы, наверное, знаете алгоритм $k$-средних:
  - инициализировать центры кластеров $\mu_1, \ldots, \mu_K$;
  - пока принадлежность кластерам не перестанет изменяться:
    - определить принадлежность $x_i$ к кластерам:

$$\text{clust}_i := \arg\min_{c \in C} \rho(x_i, \mu_c);$$

    - определить новое положение центров кластеров:

$$\mu_c := \frac{\sum_{\text{clust}_i = c} f_j(x_i)}{\sum_{\text{clust}_i = c} 1};$$

- Тем самым мы (локально) минимизируем меру ошибки

$$E(X, C) = \sum_{i=1}^{n} \|x_i - \mu_i\|^2,$$

где $\mu_i$ — ближайший к $x_i$ центр кластера.

# О байесовском выводе вообще

- Дальше мы будем заниматься вероятностными моделями.
- Чтобы ими заниматься, нужно сначала некоторое введение о том, что вообще имеется в виду под байесовским выводом и зачем всё это надо.

# Основные определения

- Нам не понадобятся математические определения сигма-алгебры, вероятностной меры, борелевских множеств и т.п.
- Достаточно понимать, что бывают дискретные случайные величины (неотрицательные вероятности исходов в сумме дают единицу) и непрерывные случайные величины (интеграл неотрицательной функции плотности равен единице).

# Основные определения

- *Совместная вероятность* – вероятность одновременного наступления двух событий, $p(x, y)$; маргинализация:

$$p(x) = \sum_y p(x, y).$$

- *Условная вероятность* – вероятность наступления одного события, если известно, что произошло другое, $p(x \mid y)$:

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

- *Теорема Байеса* – из предыдущей формулы:

$$p(y|x) = \frac{p(x|y)p(y)}{p(x)} = \frac{p(x|y)p(y)}{\sum_{y'} p(x|y')p(y')}.$$

- *Независимость*: $x$ и $y$ независимы, если

$$p(x, y) = p(x)p(y).$$

# О болезнях и вероятностях

- Классический пример из классической области применения статистики — медицины.
- Пусть некий тест на какую-нибудь болезнь имеет вероятность успеха 95% (т.е. 5% — вероятность как позитивной, так и негативной ошибки).
- Всего болезнь имеется у 1% респондентов (отложим на время то, что они разного возраста и профессий).
- Пусть некий человек получил позитивный результат теста (тест говорит, что он болен). С какой вероятностью он действительно болен?

# О болезнях и вероятностях

- Классический пример из классической области применения статистики — медицины.
- Пусть некий тест на какую-нибудь болезнь имеет вероятность успеха 95% (т.е. 5% — вероятность как позитивной, так и негативной ошибки).
- Всего болезнь имеется у 1% респондентов (отложим на время то, что они разного возраста и профессий).
- Пусть некий человек получил позитивный результат теста (тест говорит, что он болен). С какой вероятностью он действительно болен?
- Ответ: 16%.

# Доказательство

- Обозначим через $t$ результат теста, через $d$ — наличие болезни.
- $p(t=1) = p(t=1|d=1)p(d=1) + p(t=1|d=0)p(d=0)$.
- Используем теорему Байеса:

$$p(d=1|t=1) = \\ = \frac{p(t=1|d=1)p(d=1)}{p(t=1|d=1)p(d=1) + p(t=1|d=0)p(d=0)} = \\ = \frac{0.95 \times 0.01}{0.95 \times 0.01 + 0.05 \times 0.99} = 0.16.$$

# Вывод

- Вот такие задачи составляют суть вероятностного вывода (probabilistic inference).
- Поскольку они обычно основаны на теореме Байеса, вывод часто называют байесовским (Bayesian inference).
- Но не только поэтому.

# Определения

- Запишем теорему Байеса:

$$p(\theta|D) = \frac{p(\theta)p(D|\theta)}{p(D)}.$$

- Здесь $p(\theta)$ — *априорная вероятность* (prior probability), $p(D|\theta)$ — *правдоподобие* (likelihood), $p(\theta|D)$ — *апостериорная вероятность* (posterior probability), $p(D) = \int p(D \mid \theta)p(\theta)\mathrm{d}\theta$ — вероятность данных (evidence).
- Вообще, *функция правдоподобия* имеет вид

$$a \mapsto p(y|x = a)$$

для некоторой случайной величины $y$.

# ML vs. MAP

- В статистике обычно ищут *гипотезу максимального правдоподобия* (maximum likelihood):

$$\theta_{ML} = \arg\max_\theta p(D \mid \theta).$$

- Это само по себе ведёт к оверфиттингу, поэтому обычно применяют *регуляризацию*.
- В байесовском подходе ищут *апостериорное распределение* (posterior)

$$p(\theta|D) \propto p(D|\theta)p(\theta)$$

и, возможно, *максимальную апостериорную гипотезу* (maximum a posteriori):

$$\theta_{MAP} = \arg\max_\theta p(\theta \mid D) = \arg\max_\theta p(D \mid \theta)p(\theta).$$

# Постановка задачи

- Простая задача вывода: дана нечестная монетка, она подброшена $N$ раз, имеется последовательность результатов падения монетки. Надо определить её «нечестность» и предсказать, чем она выпадет в следующий раз.
- Гипотеза максимального правдоподобия скажет, что вероятность решки равна числу выпавших решек, делённому на число экспериментов.

# Постановка задачи

- Простая задача вывода: дана нечестная монетка, она подброшена $N$ раз, имеется последовательность результатов падения монетки. Надо определить её «нечестность» и предсказать, чем она выпадет в следующий раз.
- Гипотеза максимального правдоподобия скажет, что вероятность решки равна числу выпавших решек, делённому на число экспериментов.
- То есть если вы взяли незнакомую монетку, подбросили её один раз и она выпала решкой, вы теперь ожидаете, что она всегда будет выпадать только решкой, правильно?

# Первые замечания

- Если у нас есть вероятность $p_h$ того, что монетка выпадет решкой (вероятность орла $p_t = 1 - p_h$), то вероятность того, что выпадет последовательность $s$, которая содержит $n_h$ решек и $n_t$ орлов, равна

$$p(s|p_h) = p_h^{n_h}(1 - p_h)^{n_t}.$$

- Сделаем предположение: будем считать, что монетка выпадает равномерно, т.е. у нас нет априорного знания $p_h$.
- Теперь нужно использовать теорему Байеса и вычислить скрытые параметры.

# Пример применения теоремы Байеса

- $p(p_h|s) = \frac{p(s|p_h)p(p_h)}{p(s)}$.
- Здесь $p(p_h)$ – непрерывная случайная величина, сосредоточенная на интервале $[0, 1]$.
- Мы предполагаем априорную вероятность $p(p_h) = 1$, $p_h \in [0, 1]$ (т.е. априори мы не знаем, насколько нечестна монетка, и предполагаем это равновероятным). А $p(s|p_h)$ мы уже знаем.
- Итого получается:

$$p(p_h|s) = \frac{p_h^{n_h}(1 - p_h)^{n_t}}{p(s)}.$$

# Пример применения теоремы Байеса

- Итого получается:

$$p(p_h|s) = \frac{p_h^{n_h}(1-p_h)^{n_t}}{p(s)}.$$

- $p(s)$ можно подсчитать как

$$p(s) = \int_0^1 p_h^{n_h}(1-p_h)^{n_t}\,dp_h = \\ = \frac{\Gamma(n_h+1)\Gamma(n_t+1)}{\Gamma(n_h+n_t+2)} = \frac{n_h!n_t!}{(n_h+n_t+1)!},$$

но найти $\arg\max_{p_h} p(p_h \mid s) = \frac{n_h}{n_h+n_t}$ можно и без этого.

# Пример применения теоремы Байеса

- Итого получается:

$$p(p_h|s) = \frac{p_h^{n_h}(1-p_h)^{n_t}}{p(s)}.$$

- Но это ещё не всё. Чтобы предсказать следующий исход, надо найти $p(\text{heads}|s)$:

$$p(\text{heads}|s) = \int_0^1 p(\text{heads}|p_h)p(p_h|s)dp_h = \int_0^1 \frac{p_h^{n_h+1}(1-p_h)^{n_t}}{p(s)}dp_h =$$
$$= \frac{(n_h+1)!n_t!}{(n_h+n_t+2)!} \cdot \frac{(n_h+n_t+1)!}{n_h!n_t!} = \frac{n_h+1}{n_h+n_t+2}.$$

- Получили правило Лапласа.

# Пример применения теоремы Байеса

- Итого получается:

$$p(p_h|s) = \frac{p_h^{n_h}(1-p_h)^{n_t}}{p(s)}.$$

- Это была иллюстрация двух задач байесовского вывода:
  1. найти апостериорное распределение на гипотезах/параметрах:

$$p(\theta \mid D) \propto p(D|\theta)p(\theta)$$

(и/или найти максимальную апостериорную гипотезу $\arg\max_\theta p(\theta \mid D)$);
  2. найти апостериорное распределение исходов дальнейших экспериментов:

$$p(x \mid D) \propto \int_{\theta\in\Theta} p(x \mid \theta)p(D|\theta)p(\theta)\mathrm{d}\theta.$$

# Сопряжённые априорные распределения

- Ещё надо научиться выбирать априорные распределения.
- Разумная цель: давайте будем выбирать распределения так, чтобы они оставались такими же и *a posteriori*.
- До начала вывода есть априорное распределение $p(\theta)$.
- После него есть какое-то новое апостериорное распределение $p(\theta \mid x)$.
- Я хочу, чтобы $p(\theta \mid x)$ тоже имело тот же вид, что и $p(\theta)$, просто с другими параметрами.

# Сопряжённые априорные распределения

- Не слишком формальное определение: семейство распределений $p(\theta \mid \alpha)$ называется семейством *сопряжённых априорных распределений* для семейства правдоподобий $p(x \mid \theta)$, если после умножения на правдоподобие апостериорное распределение $p(\theta \mid x, \alpha)$ остаётся в том же семействе: $p(\theta \mid x, \alpha) = p(\theta \mid \alpha')$.
- $\alpha$ называются *гиперпараметрами* (hyperparameters), это «параметры распределения параметров».
- Тривиальный пример: семейство всех распределений будет сопряжённым чему угодно, но это не очень интересно.

# Сопряжённые априорные распределения

- Разумеется, вид хорошего априорного распределения зависит от вида распределения собственно данных, $p(x \mid \theta)$.
- Сопряжённые априорные распределения подсчитаны для многих распределений, мы приведём два примера.

# Испытания Бернулли

- Каким будет сопряжённое априорное распределение для бросания нечестной монетки (испытаний Бернулли)?
- Ответ: это будет бета-распределение; плотность распределения нечестности монетки $\theta$

$$p(\theta \mid \alpha, \beta) = \frac{\theta^{\alpha-1}(1-\theta)^{\beta-1}}{B(\alpha, \beta)}.$$

# Испытания Бернулли

- Плотность распределения нечестности монетки $\theta$

$$p(\theta \mid \alpha, \beta) = \frac{\theta^{\alpha-1}(1-\theta)^{\beta-1}}{B(\alpha, \beta)}.$$

- Тогда, если мы посэмплируем монетку, получив $s$ орлов и $f$ решек, получится

$$p(s, f \mid \theta) = \binom{s+f}{s} \theta^s (1-\theta)^f, \text{ и}$$

$$p(\theta|s, f) = \frac{\binom{s+f}{s} \theta^{s+\alpha-1}(1-\theta)^{f+\beta-1}/B(\alpha, \beta)}{\int_0^1 \binom{s+f}{s} x^{s+\alpha-1}(1-x)^{f+\beta-1}/B(\alpha, \beta)dx} =$$

$$= \frac{\theta^{s+\alpha-1}(1-\theta)^{f+\beta-1}}{B(s+\alpha, f+\beta)}.$$

# Испытания Бернулли

- Итого получается, что сопряжённое априорное распределение для параметра нечестной монетки $\theta$ – это

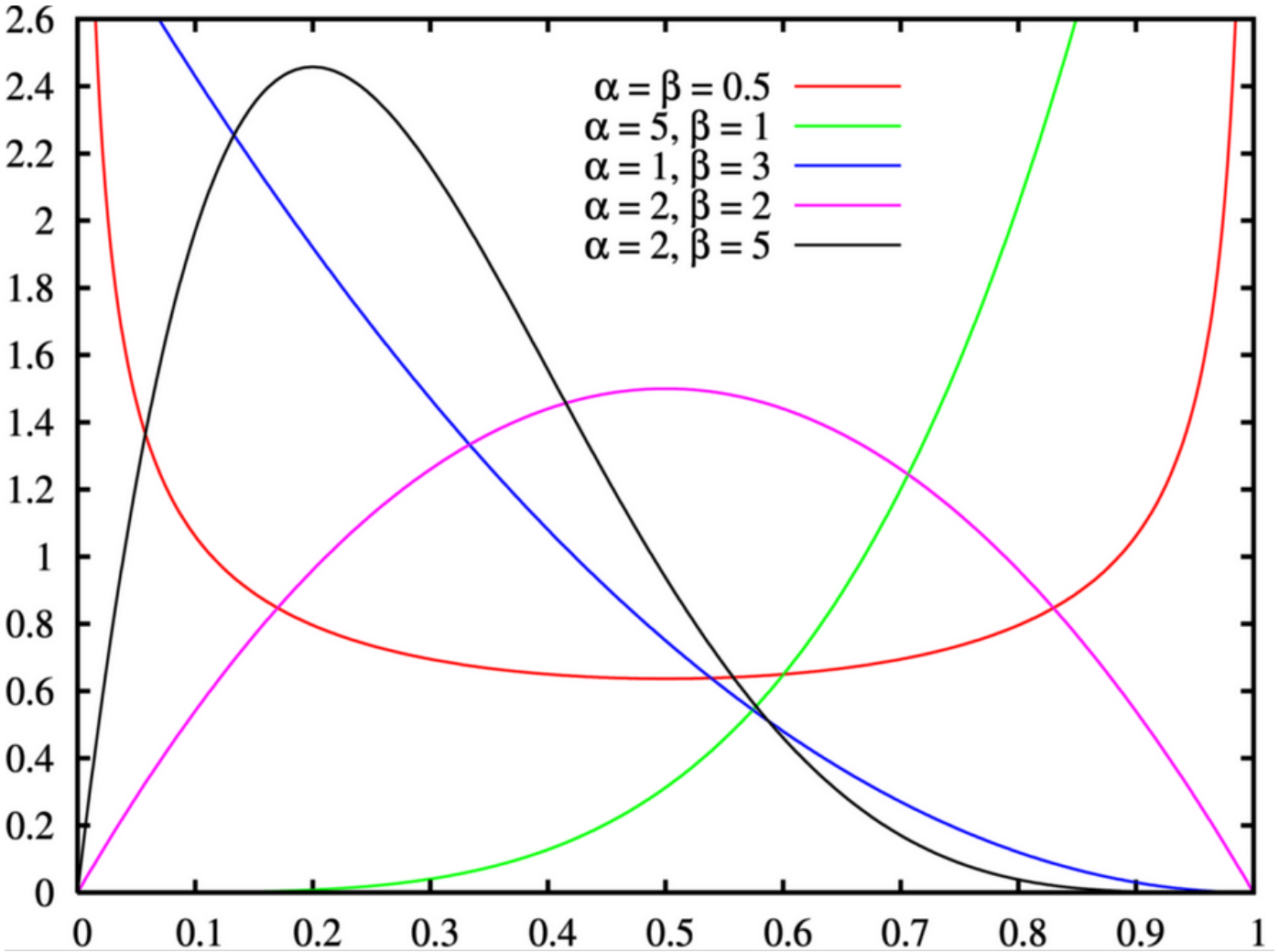
$$p(\theta \mid \alpha, \beta) \propto \theta^{\alpha-1}(1-\theta)^{\beta-1}.$$

- После получения новых данных с $s$ орлами и $f$ решками гиперпараметры меняются на

$$p(\theta \mid s+\alpha, f+\beta) \propto \theta^{s+\alpha-1}(1-\theta)^{f+\beta-1}.$$

- На этом этапе можно забыть про сложные формулы и выводы, получилось очень простое правило обучения (под обучением теперь понимается изменение гиперпараметров).

# Бета–распределение



# Мультиномиальное распределение

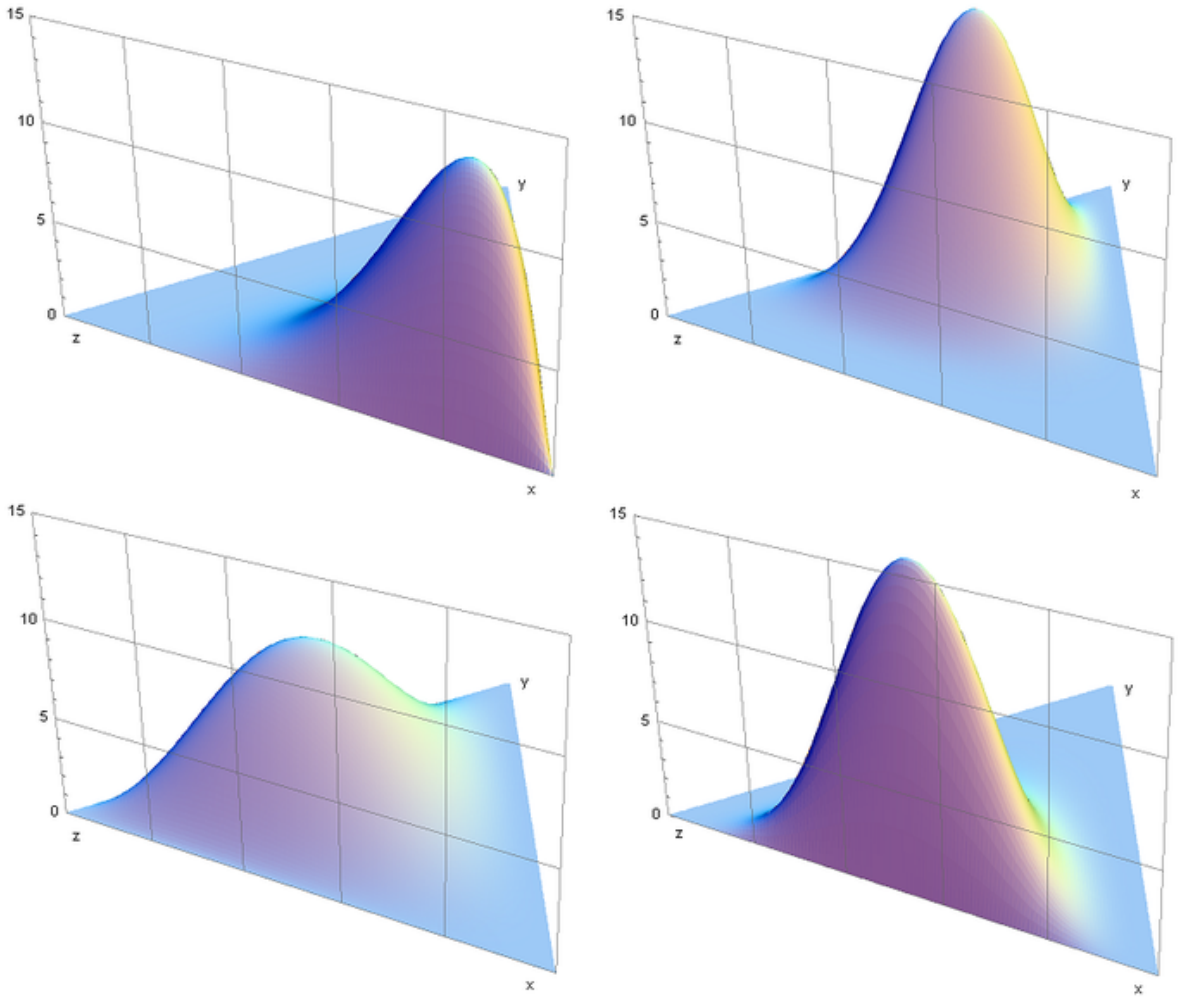
- Простое обобщение: рассмотрим мультиномиальное распределение с $n$ испытаниями, $k$ категориями и по $x_i$ экспериментов дали категорию $i$.
- Параметры $\theta_i$ показывают вероятность попасть в категорию $i$:

$$p(x \mid \theta) = \binom{n}{x_1, \ldots, x_n} \theta_1^{x_1} \theta_2^{x_2} \ldots \theta_k^{x_k}.$$

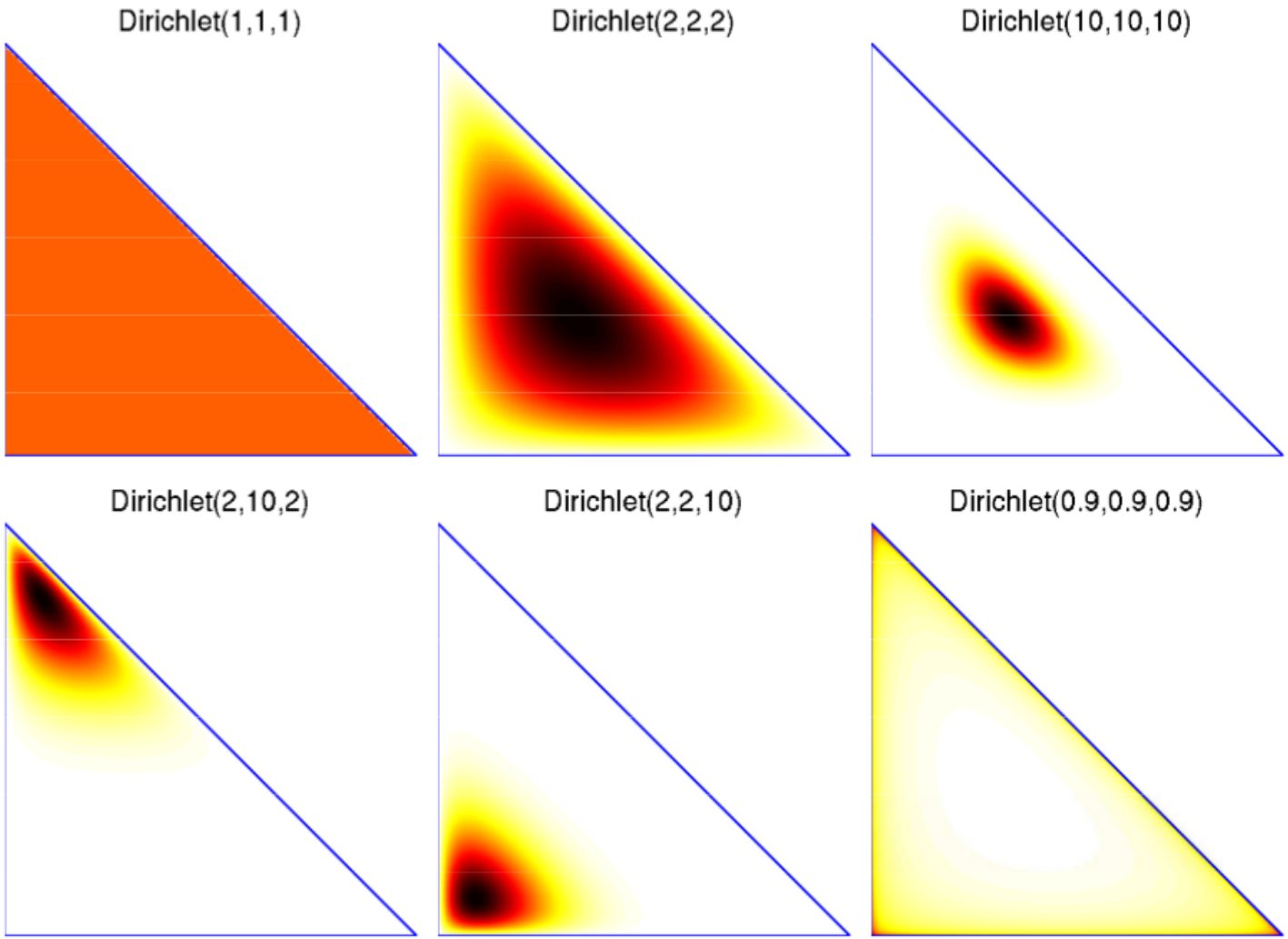
- Сопряжённым априорным распределением будет распределение Дирихле:

$$p(\theta \mid \alpha) \propto \theta_1^{\alpha_1 - 1} \theta_2^{\alpha_2 - 1} \ldots \theta_k^{\alpha_k - 1}.$$

# Распределение Дирихле



# Распределение Дирихле



# Вероятностная модель кластеризации

- Теперь давайте попробуем построить вероятностную модель для задачи кластеризации.
- Вероятностные модели задаются порождающими процессами (generative processes): надо понять, как порождается каждая точка, а потом записать каждый шаг этого порождения в виде распределения вероятностей.
- Если вы запишете порождающий процесс, обычно можно будет достаточно просто сделать вывод на этой модели (например, при помощи сэмплирования по Гиббсу).

# Вероятностная модель кластеризации

- Порождающий процесс для кластеризации – для очередной точки $y_n$:
  - выбрать кластер $c_n$, из которого мы будем порождать $y_n$;
  - выбрать точку $y_n$ из этого кластера.
- Нам нужно придумать распределение для точки в одном кластере и распределение на кластерах.

# Вероятностная модель кластеризации

- Распределение для точки внутри кластера $p(y \mid \theta)$ давайте считать гауссовским (нормальным; хотя это не важно для нас сейчас):
  - параметры кластера $\theta_k$, $k = 1..K$, – это среднее и матрица ковариаций;
  - точку берём как $y \sim F(y \mid \theta_k) = \mathcal{N}(y \mid \theta_k)$.
- А распределение на кластерах – это просто дискретное распределение, бросание кубика: задаём $K$ чисел $p_1, \ldots, p_K$, для которых $\sum_i p_i = 1$, и выбираем кластер для очередной точки.

# Вероятностная модель кластеризации

- Итого получается, что порождающее распределение для одной точки выглядит как

$$p(y_n, c_n) = p_{c_n} F(y_n \mid \theta_{c_n}),$$

а для всего датасета $y_1, \ldots, y_N$ – как

$$p(\mathbf{y}, \mathbf{c}) = \prod_{n=1}^{N} p_{c_n} F(y_n \mid \theta_{c_n}).$$

- Зная параметры $p_k$ и $\theta_k$, мы можем понять про точки, к каким кластерам с какими вероятностями они принадлежат:

$$p(\mathbf{c} \mid \mathbf{y}) = \frac{p(\mathbf{y}, \mathbf{c})}{\sum_{\mathbf{c}} p(\mathbf{y}, \mathbf{c})}.$$

# Вероятностная модель кластеризации

- Но это распределение для известных параметров $p_k$ и $\theta_k$; в реальности они неизвестны, есть только датасет.
- Классический "статистический" подход – найти параметры максимального правдоподобия, т.е. $p_k$ и $\theta_k$, для которых максимизируется правдоподобие

$$p(\mathbf{y} \mid p_k, \theta_k) = \prod_{n=1}^{N} \sum_{k=1}^{K} p_k F(y_n \mid \theta_k).$$

# EM-алгоритм

- На самом деле у нас здесь получается смесь вероятностных распределений:

$$p(\mathbf{y} \mid p_k, \theta_k) = \prod_{n=1}^{N} \sum_{k=1}^{K} p_k F(y_n \mid \theta_k).$$

- И кластеризация – это задача разделения смеси вероятностных распределений. Её решают так называемым *EM-алгоритмом* (Expectation Maximization).
- Введём скрытые переменные — вероятности $g_{nk}$ того, что точка $y_n$ принадлежит кластеру $k$.

# EM-алгоритм

- Идея EM-алгоритма:
  - на E-шаге подсчитаем ожидания скрытых переменных;
  - на M-шаге зафиксируем скрытые переменные и найдём параметры максимального правдоподобия;
  - повторим до сходимости.
- Математический смысл здесь в том, что на каждой такой итерации общее правдоподобие увеличивается.

# EM-алгоритм

- Для задачи кластеризации:
  - $E$–шаг: по формуле Байеса вычисляются скрытые переменные $g_{nk}$:

$$g_{nk} = \frac{p_k F(y_n \mid \theta_k)}{\sum_{s=1}^{K} p_s F(y_n \mid \theta_s)}.$$

  - $M$–шаг: с использованием $g_{nk}$ уточняем параметры кластеров; например, для нормального распределения:

$$p_k = \frac{1}{N} \sum_{n=1}^{N} g_{nk},$$

$$\mu_k = \frac{1}{N} \sum_{n=1}^{N} g_{nk} y_n, \quad \Sigma_{ck} = \frac{1}{N p_k} \sum_{i=1}^{n} g_{nk} (y_n - \mu_k)(y_n - \mu_k)^\top.$$

- Кстати, метод $k$-средних – это фактически упрощённый EM для случая, когда $g_{nk}$ могут быть только 0 или 1.

# Байесовский подход к кластеризации

- Однако максимизировать правдоподобие не всегда хорошо, потому что точка максимума – это ещё не всё правдоподобие, да и максимум мы можем искать только локальный.
- В реальности это приводит к оверфиттингу (overfitting): параметры максимального правдоподобия обычно слишком хорошо подходят к данным и из-за этого только хуже обобщаются.
- Поэтому используют байесовские методы – назначают *априорное распределение* на параметры кластеров, а затем интегрируют по *всем возможным* значениям параметров кластеров, используя это априорное распределение.

# Байесовский подход к кластеризации

- Формально говоря, давайте введём априорное распределение $G_0$, из которого будем брать $\theta \sim G_0$; в генеративный процесс добавится начальный шаг:
  - выбрать параметры каждого кластера $\theta_k \sim G_0$,

  и формально мы получаем

$$p(\mathbf{y}, \mathbf{c}, \boldsymbol{\theta}) = \prod_{k=1}^{K} G_0(\theta_k) \prod_{n=1}^{N} p_{c_n} F(y_n \mid \theta_{c_n}).$$

# Байесовский подход к кластеризации

- ...формально мы получаем

$$p(\mathbf{y}, \mathbf{c}, \theta) = \prod_{k=1}^{K} G_0(\theta_k) \prod_{n=1}^{N} p_{c_n} F(y_n \mid \theta_{c_n}).$$

- Теперь можно найти $p(\mathbf{y} \mid \mathbf{c})$, проинтегрировав по всем $\theta$:

$$p(\mathbf{y} \mid \mathbf{c}) = \int_{\theta} \left( \prod_{k=1}^{K} G_0(\theta_k) \prod_{n=1}^{N} F(y_n \mid \theta_{c_n}) \right) \mathrm{d}\theta.$$

- И кластеризацию можно найти как

$$p(\mathbf{c} \mid \mathbf{y}) = \frac{p(\mathbf{y} \mid \mathbf{c}) p(\mathbf{c})}{\sum_{\mathbf{c}} p(\mathbf{y} \mid \mathbf{c}) p(\mathbf{c})}.$$

# Model selection

- ...кластеризацию можно найти как

$$p(\mathbf{c} \mid \mathbf{y}) = \frac{p(\mathbf{y} \mid \mathbf{c})p(\mathbf{c})}{\sum_{\mathbf{c}} p(\mathbf{y} \mid \mathbf{c})p(\mathbf{c})}.$$

- Остаётся тонкий вопрос: а что такое $p(\mathbf{c})$? Это должно быть априорное распределение на всех возможных назначениях индексов кластеров к точкам (groupings).
- Перечислить их мы не можем, в EM-алгоритме мы просто фиксировали число кластеров и выражали $p(\mathbf{c})$ как коэффициенты смеси $p_k$.

# Model selection

- Но при таком подходе остаётся самый интересный вопрос: а откуда взять число кластеров?
- Обычно заранее число кластеров неизвестно.
- Более того, если мы будем считать, например, внутрикластерное расстояние или общее правдоподобие точек в моделях с разным числом кластеров, оно будет просто монотонно убывать.
- Оптимальная с точки зрения правдоподобия кластеризация – когда каждая точка сама себе кластер; это возможно, потому что с таким числом кластеров в модели получается очень много свободных параметров.

# Model selection

- Это называется задача выбора модели (model selection): как выбрать модель правильной сложности, т.е. с оптимальным числом параметров.
- Чем больше параметров, тем “лучше” кластеризация.
- Есть ряд критериев, которые можно применять для выбора модели и которые учитывают как правдоподобие, так и сложность модели.

# Model selection

- *Байесовский информационный критерий* (Bayesian information criterion, BIC), он же *критерий Шварца* (Schwarz criterion):

$$\mathrm{BIC}(M) = \ln p(D \mid \theta_{\mathrm{MAP}}) - \frac{1}{2} M \ln N,$$

  где $M$ – число параметров, $N$ – число точек в данных.
- Получается как грубое приближение апостериорного распределения, “усреднённого” по разным моделям.
- Т.е. алгоритм такой: кластеризовать много раз для разного числа кластеров, посчитать BIC, сравнить; так действительно часто делают на практике.
- Сегодня я рассказываю о том, как можно делать по-другому.

# Outline

# Идея непараметрических методов

- Как мы уже видели, в вероятностных моделях есть важная проблема: выбор сложности модели.
- Как найти, например, число кластеров?
- Можно пользоваться методами model selection, обучив несколько моделей.
- А можно сделать по-другому – ввести априорное распределение *сразу на все модели разной сложности*; это и есть основная идея байесовских непараметрических методов.

# Идея непараметрических методов

- Непараметрическая байесовская статистика (Bayesian nonparametric) появилась ещё в 1970-х: [Ferguson, 1973; Antoniak, 1974] – изучали *процессы Дирихле* (Dirichlet processes), в основном из чисто математического интереса.
- [Rasmussen, 2000]: "The Infinite Gaussian Mixture Model" – первое применение процессов Дирихле к машинному обучению.
- Затем в машинное обучение пришли процессы Дирихле (с представлением в виде процесса китайского ресторана или процесса разлома палки, stick breaking process), индийского буфета, процесс Питмана–Йора (Pitman–Yor) и т.п.
- Много расширений и применений – поговорим позже.

# Идея непараметрических методов

- Смысл: давайте попробуем ввести единое распределение на всех моделях разной сложности.
- При этом мы введём его так, чтобы менее сложные модели имели более
- Затем можно будет применить обычный байесовский подход и понять, какая сложность модели наиболее правдоподобна из данных.
- “Непараметрический” не значит “нет параметров”, это значит, что число параметров зависит от данных и может расти, когда данных становится больше.
- Бывают невероятностные непараметрические методы, например метод ближайших соседей.

# Процесс китайского ресторана

- Теперь возвращаемся к кластеризации.
- В байесовском выводе для кластеризации участвовало распределение $p(\mathbf{c})$; переменные, взятые по этому распределению, должны были показывать, как точки распределены по кластерам.
- Именно этим и занимается *процесс китайского ресторана* (Chinese restaurant process).

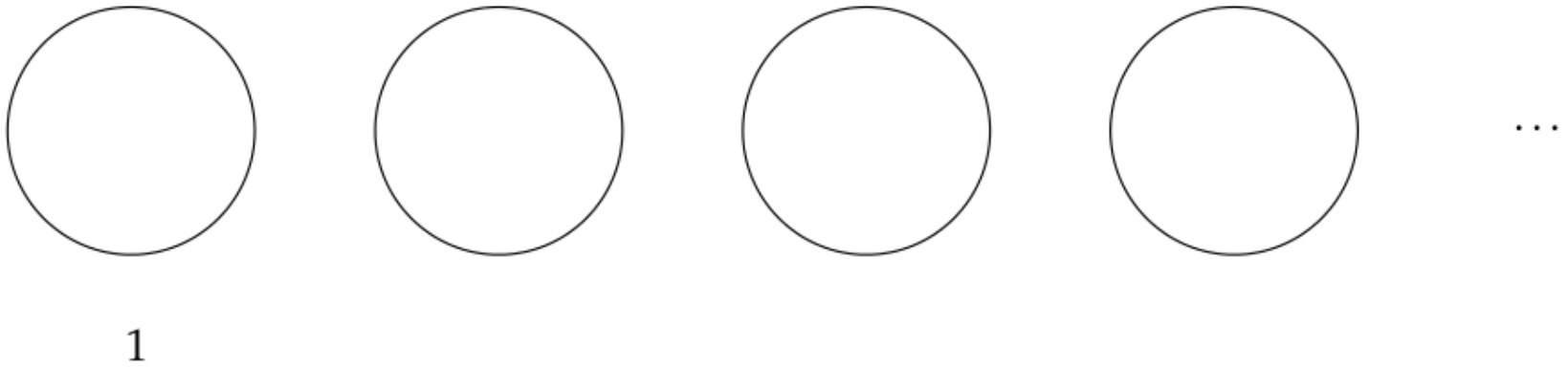
# Процесс китайского ресторана

- Есть ресторан с бесконечным числом столов. В него один за другим заходят посетители:
  - первый посетитель заходит и садится за первый столик;
  - второй заходит и садится за первый столик с вероятностью $\frac{1}{1+\alpha}$, за второй (пустой) – с вероятностью $\frac{\alpha}{1+\alpha}$;
  - ...
  - $n$-й посетитель садится за каждый из уже занятых столиков с вероятностью, пропорциональной тому, сколько там сидят человек, либо садится за пустой столик с вероятностью, пропорциональной $\alpha$:

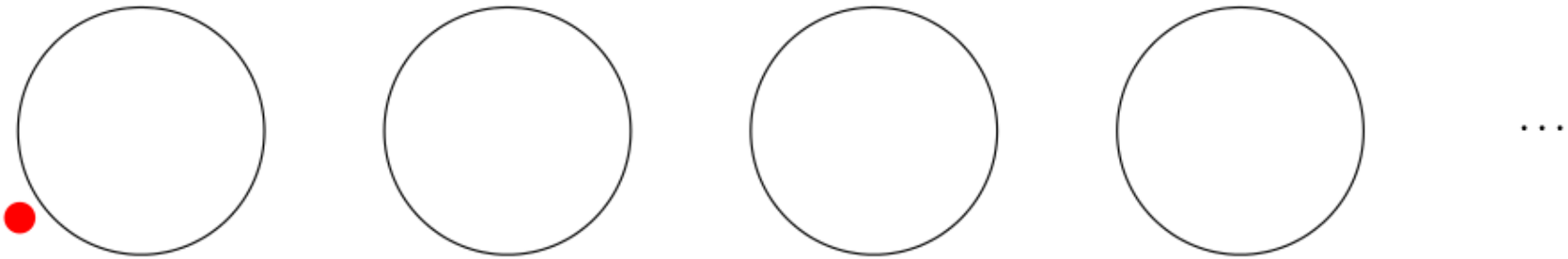
$$p(c_n = k \mid c_1, \ldots, c_{n-1}) = \begin{cases} \frac{m_k}{n-1+\alpha}, & \text{если } k \leq K_{n-1}, \\ \frac{\alpha}{n-1+\alpha}, & \text{если } k = K_{n-1}. \end{cases}$$

- Столики – это кластеры, посетители – это точки, $\alpha$ – *концентрационный параметр* (concentration parameter).

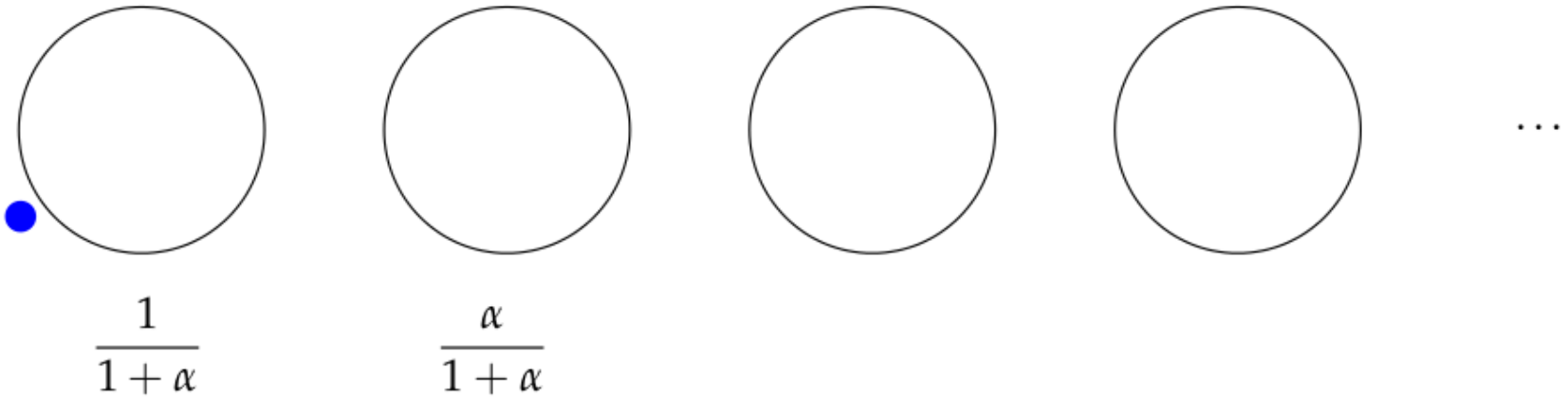
# Процесс; картинки из [Johnson]



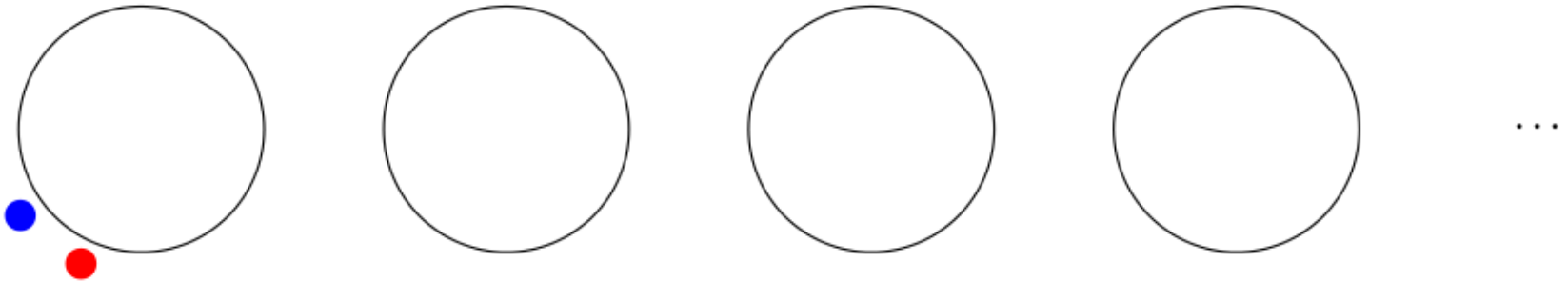
# Процесс; картинки из [Johnson]



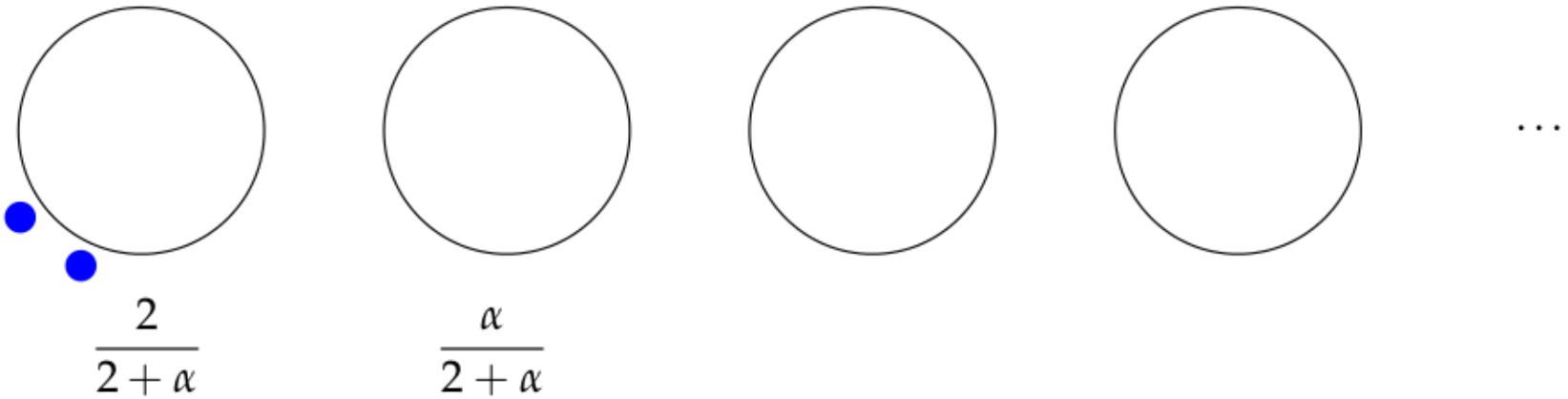
# Процесс; картинки из [Johnson]



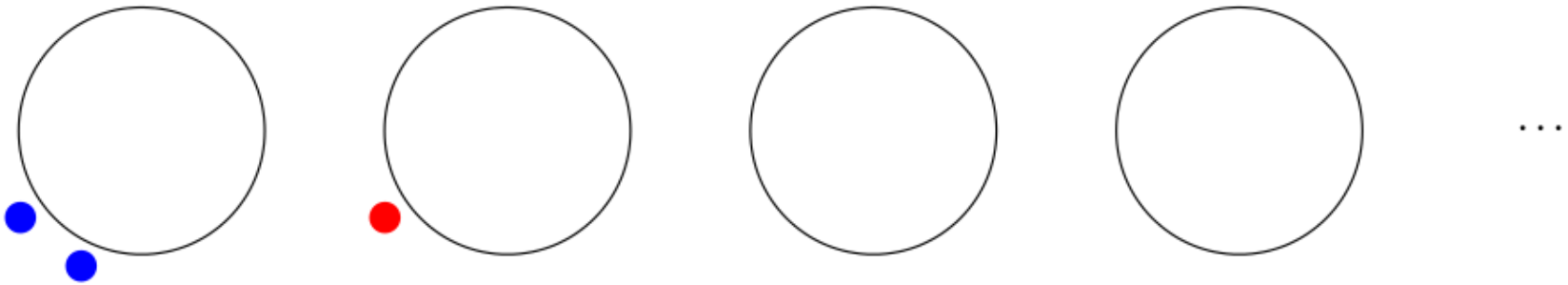
# Процесс; картинки из [Johnson]



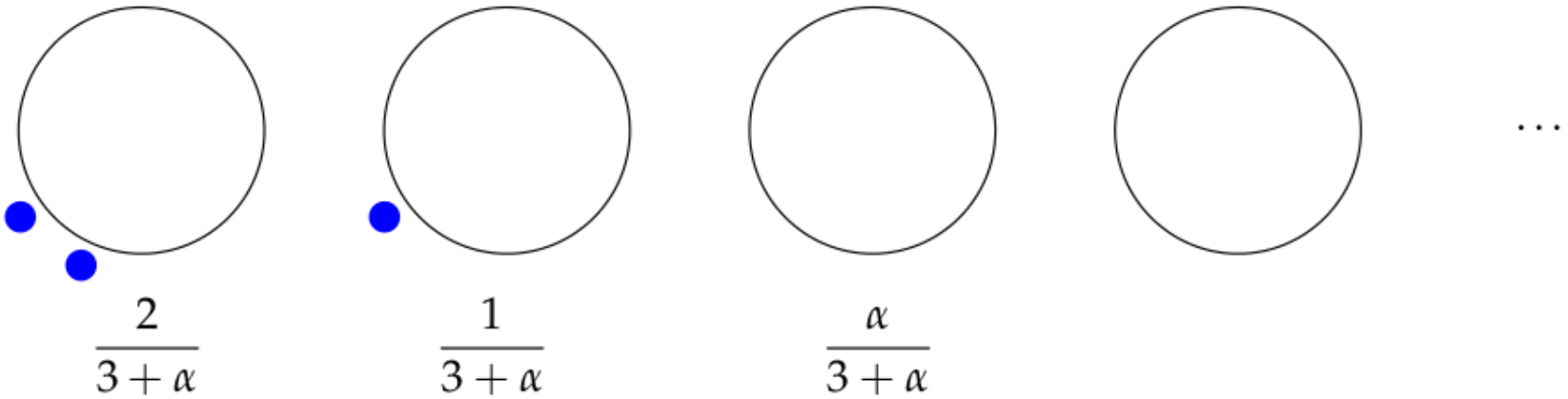
# Процесс; картинки из [Johnson]



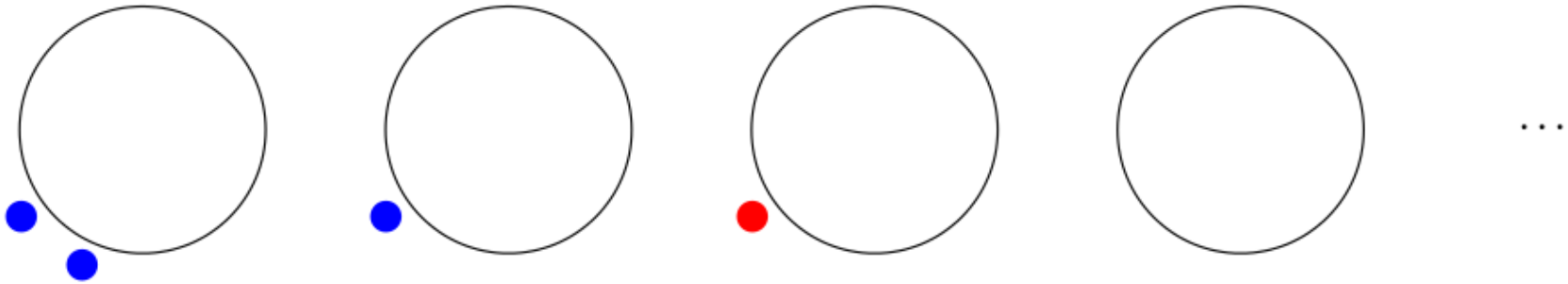
# Процесс; картинки из [Johnson]



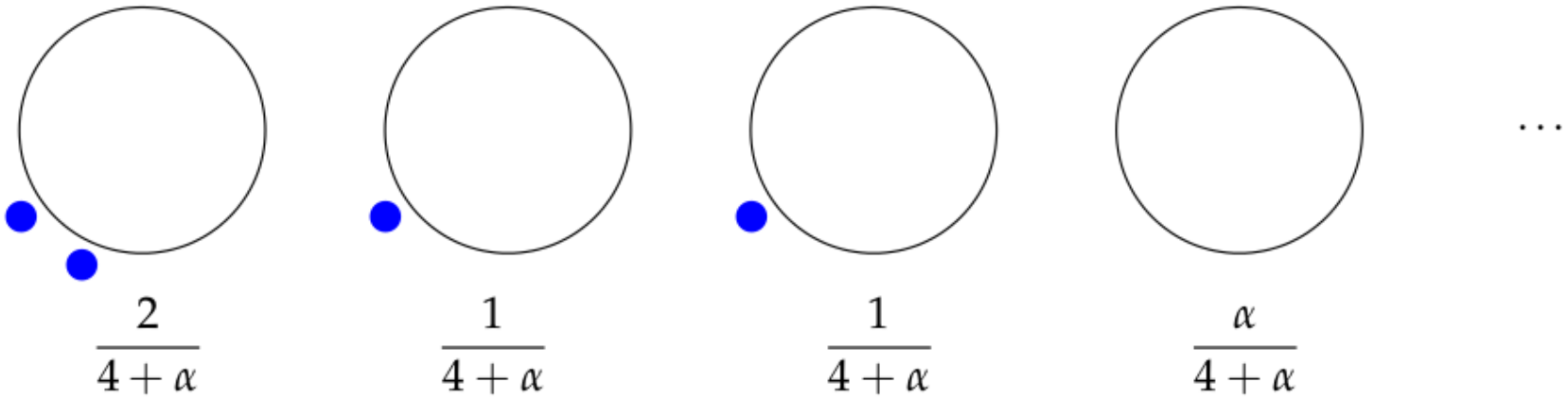
# Процесс; картинки из [Johnson]



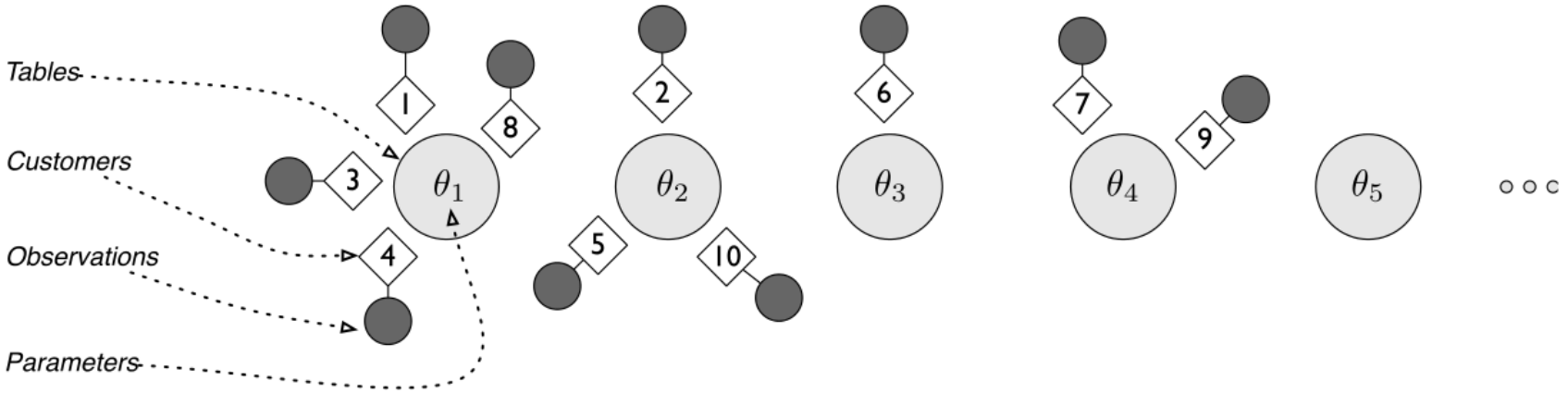
# Процесс; картинки из [Johnson]



# Процесс; картинки из [Johnson]



# Картинка из [Gershman, Blei, 2012]



# Процесс китайского ресторана

- Оказывается, что для такого процесса верно свойство перестановочности (exchangeability): группировка посетителей по столикам не зависит от того, в каком порядке они приходили:
  - выделим из совместного распределения

$$p(c_1, \ldots, c_N) = p(c_1)p(c_2 \mid c_1) \ldots p(c_N \mid c_1, \ldots, c_{N-1})$$

посетителей, севших за столик $k$:

$$\frac{\alpha \cdot 1 \cdot 2 \cdot \ldots \cdot (N_k - 1)}{(I_{k,1} - 1 + \alpha)(I_{k,2} - 1 + \alpha) \ldots (I_{k,N_k} - 1 + \alpha)}.$$

  - числитель тут $\alpha(N_k - 1)!$.

# Процесс китайского ресторана

- Оказывается, что для такого процесса верно свойство перестановочности (exchangeability): группировка посетителей по столикам не зависит от того, в каком порядке они приходили:
  - числитель тут $\alpha(N_k - 1)!$, а знаменатели объединим обратно:

$$p(\mathbf{c}) = \prod_{k=1}^{K} \frac{\alpha(N_k - 1)!}{(I_{k,1} - 1 + \alpha)(I_{k,2} - 1 + \alpha)\dots(I_{k,N_k} - 1 + \alpha)} = \\ = \frac{\alpha^K \prod_{k=1}^{K}(N_k - 1)!}{\prod_{i=1}^{N}(i - 1 + \alpha)},$$

и это зависит только от $N_k$, а от порядка не зависит.

# Процесс китайского ресторана

- Другая метафора для того же самого процесса – *урны Пойя*:
  - начинаем с урны, в которой лежат $\alpha G_0(x)$ шаров цвета $x$, где $G_0$ – некоторое начальное распределение;
  - на каждом шаге вытягиваем шар, записываем его цвет, потом возвращаем шар и ещё один шар того же цвета.

# Процесс китайского ресторана

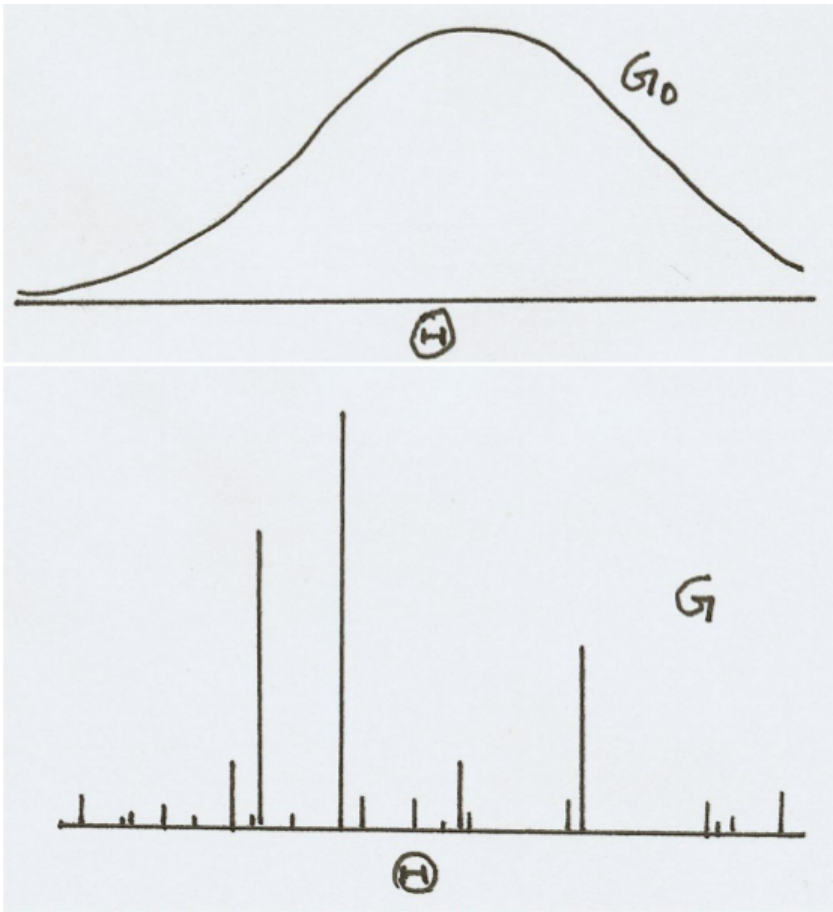
- А для генерации только весов можно сразу использовать процесс разлома палки (stick breaking process):
  - начинаем с палки длиной 1;
  - ломаем её в точке $\beta_1 \sim \text{Beta}(1, \alpha)$, где Beta – бета-распределение; вес $p_1$ – это длина палки слева, т.е. $\beta_1$;
  - дальше ломаем оставшуюся палку в пропорции $\beta_2 \sim \text{Beta}(1, \alpha)$, т.е. вес $p_2$ – это $(1 - \beta_1)\beta_2$;
  - и так далее...
- Всё это ещё можно рассматривать как распределение над распределениями; тогда это называется *процесс Дирихле* (Dirichlet process), и мы вытягиваем $G \sim \text{DP}(G_0, \alpha)$.

# Процесс Дирихле

- $G \sim \mathrm{DP}(G_0, \alpha)$
- Сэмплы из процесса Дирихле с вероятностью 1 дискретные:

$$G(\theta) = \sum_{k=1}^{\infty} p_k \delta_{\theta_k}(\theta).$$

- Но в ожидании всё равно $G_0$ получается.



# Процесс китайского ресторана

- Таким образом, у нас получается следующий генеративный процесс для кластеризации:
  - сгенерировать $g_1, \ldots, g_N \sim \mathrm{CRP}(\alpha)$ китайским рестораном; получатся $K$ непустых столиков;
  - сгенерировать параметры каждого столика $\theta_1, \ldots, \theta_K \sim G_0$;
  - сгенерировать каждую точку $y_n \sim F(y_n \mid \theta_{g_n})$ из соответствующего ей столика.

# Вывод в модели китайского ресторана

- Как теперь вести вывод в этой модели?
- Основная идея – *сэмплирование по Гиббсу* (Gibbs sampling); это очень общий алгоритм вывода на вероятностных моделях:
  - дано сложное распределение $p(x_1, x_2, \ldots, x_m)$; хотим породить сэмплы из этого распределения;
  - инициализируем переменные как-нибудь;
  - фиксируем все переменные, кроме $x_1$, порождаем $x_1 \sim p(x_1 \mid x_2, \ldots, x_m)$;
  - фиксируем все переменные, кроме $x_2$, порождаем $x_2 \sim p(x_2 \mid x_1, x_3 \ldots, x_m)$;
  - ...
  - фиксируем все переменные, кроме $x_m$, порождаем $x_m \sim p(x_m \mid x_1, \ldots, x_{m-1})$;
  - повторяем; если сделать достаточно много итераций, получится сэмпл из $p(x_1, x_2, \ldots, x_m)$.

# Вывод в модели китайского ресторана

- В случае кластеризации идея получается такая:
  - берём точки, случайно назначаем точки по кластерам;
  - выбираем точку, фиксируем кластеры всех остальных точек, затем выбираем кластер для текущей точки по CRP;
  - повторяем.
- Рано или поздно этот процесс должен будет сойтись, так что просто ждём, пока назначения по кластерам перестанут значительно меняться.
- По сути пользуемся перестановочностью, т.к. для нас каждая точка – как будто последняя.

# Вывод в модели китайского ресторана

- Если формально – сам китайский ресторан сэмплируется так:
  - случайно назначаем точки по кластерам;
  - в цикле по точкам выбираем:

$$p(c_i = c \mid \mathbf{c}_{-i}) = \frac{n_{-i,c}}{n - 1 + \alpha}, \quad n_{-i,c} > 0,$$
$$p(c_i \neq c \mid \mathbf{c}_{-i}) = \frac{\alpha}{n - 1 + \alpha};$$

  - и так пока не сойдётся.

# Вывод в модели китайского ресторана

- Но в этом пока никак не участвуют сами точки; чтобы их добавить, надо перейти к сэмплированию из полноценного процесса Дирихле, где $G_0$ – априорное распределение для распределения кластеров; получается:
  - случайно назначаем точки по кластерам;
  - в цикле по точкам выбираем:

$$p(c_i = c \mid \mathbf{c}_{-i}, y_i, \theta) \propto F(y_i \mid \theta_c)\frac{n_{-i,c}}{n-1+\alpha}, \quad n_{-i,c} > 0,$$

$$p(c_i \neq c \mid \mathbf{c}_{-i}, y_i, \theta) \propto \frac{\alpha}{n-1+\alpha}\int F(y_i \mid \theta)\mathrm{d}G_0(\theta).$$

  - и так пока не сойдётся.
- Здесь, когда нужен новый кластер, надо выбрать $\theta_c \sim H_i$, по апостериорному распределению с априорным $G_0$ и одним наблюдением $y_i$.

# Вывод в модели китайского ресторана

- Если же $G_0$ – сопряжённое распределение для распределения кластеров, то $\theta_c$ можно вообще исключить из процесса:
  - случайно назначаем точки по кластерам;
  - в цикле по точкам выбираем:

$$p(c_i = c \mid \mathbf{c}_{-i}, y_i) \propto \frac{n_{-i,c}}{n-1+\alpha} \int F(y_i \mid \theta) \mathrm{d}H_{-i,c}(\theta), \quad n_{-i,c} > 0,$$

$$p(c_i \neq c \mid \mathbf{c}_{-i}, y_i) \propto \frac{\alpha}{n-1+\alpha} \int F(y_i \mid \theta) \mathrm{d}G_0(\theta).$$

- Здесь $H_{-i,c}$ – апостериорное распределение на $\theta$ с априорным $G_0$ и наблюдениями $y_j$, для которых $c_j = c$, $j \neq i$.

# Вывод в модели китайского ресторана

- Сэмплирование по Гиббсу – очень мощный инструмент; его легко расширить на новые модели.
- Если вы придумали (или используете) вероятностную модель, заданную порождающим процессом, весьма вероятно, что алгоритм сэмплирования по Гиббсу можно сделать просто посмотрев на этот процесс, без какой-то сложной математики; но сходиться может медленно.
- Альтернативный способ – вариационные приближения; совсем другой подход, сложнее, но часто эффективнее.
- Для обычного CRP для задачи кластеризации вариационное приближение, конечно, известно.

# Апостериорное распределение

- Последнее замечание – давайте посмотрим на апостериорное распределение модели CRP, т.е. что модель думает про следующую точку:

$$p(y_{N+1} \mid \mathbf{y}) = \sum_{\mathbf{c}} \int_{\theta} p(y_{N+1} \mid c_{N+1}, \theta) p(c_{N+1} \mid \mathbf{c}) p(\mathbf{c}, \theta \mid \mathbf{y}) \mathrm{d}\theta.$$

- В нём участвует априорное распределение CRP, $p(c_{N+1} \mid \mathbf{c})$.
- Фактически это означает, что если добавлять новые данные в обученную модель CRP, то новые данные могут привести к тому, что породится новый кластер.

# Outline

# Скрытые факторы

- Другая классическая задача машинного обучения – выделение скрытых факторов (latent factors).
- Факторный анализ (FA), анализ главных компонент (PCA), анализ независимых компонент (ICA)...
- Здесь тоже возникает проблема model selection: сколько брать факторов?

# Скрытые факторы

- Рассмотрим самую простую модель факторного анализа: пусть мы наблюдаем $N$ $M$-мерных векторов $\mathbf{Y} = \{\mathbf{y}_1, \ldots, \mathbf{y}_N\}$; т.е. в матрице $\mathbf{Y}$ по строкам идут наблюдения, по столбцам – размерности наблюдений.
- Предположим, что данные порождены зашумленной линейной комбинацией скрытых факторов:

$$\mathbf{y}_n = \mathbf{G}\mathbf{x}_n + \boldsymbol{\epsilon},$$

где $\mathbf{x}_n$ – $K$-мерный вектор выраженности факторов в $\mathbf{y}_n$, а $\mathbf{G}$ – $M \times K$ матрица, которая показывает, как каждый фактор влияет на наблюдения.

# Скрытые факторы

- $\mathbf{y}_n = \mathbf{G}\mathbf{x}_n + \boldsymbol{\epsilon}$.
- Сделаем модель разреженной – пусть

$$G_{mk} = z_{mk} w_{mk},$$

где $w_{mk}$ – вес, а $z_{mk}$ – бинарная переменная, которая показывает, "включён" ли фактор $k$ в размерности $m$.

- Это называется "spike and slab" модель, потому что получается смесь из "spike" в нуле размера $p(z_{mk} = 0)$ и "slab" $p(w_{mk})$ (обычно гауссовского) по остальным переменным.

# Скрытые факторы

- $\mathbf{y}_n = \mathbf{G}\mathbf{x}_n + \boldsymbol{\epsilon}$, $G_{mk} = z_{mk} w_{mk}$.
- Теперь будем делать байесовский вывод:

$$p(\mathbf{X}, \mathbf{W}, \mathbf{Z} \mid \mathbf{Y}) \propto p(\mathbf{Y} \mid \mathbf{X}, \mathbf{W}, \mathbf{Z}) p(\mathbf{X}) p(\mathbf{W}) p(\mathbf{Z}).$$

- Для этого надо выбрать априорное распределение $p(\mathbf{Z})$; это легко, если известно $K$.
- А если $K$ (число факторов) неизвестно? Тогда как раз можно опять непараметрически.

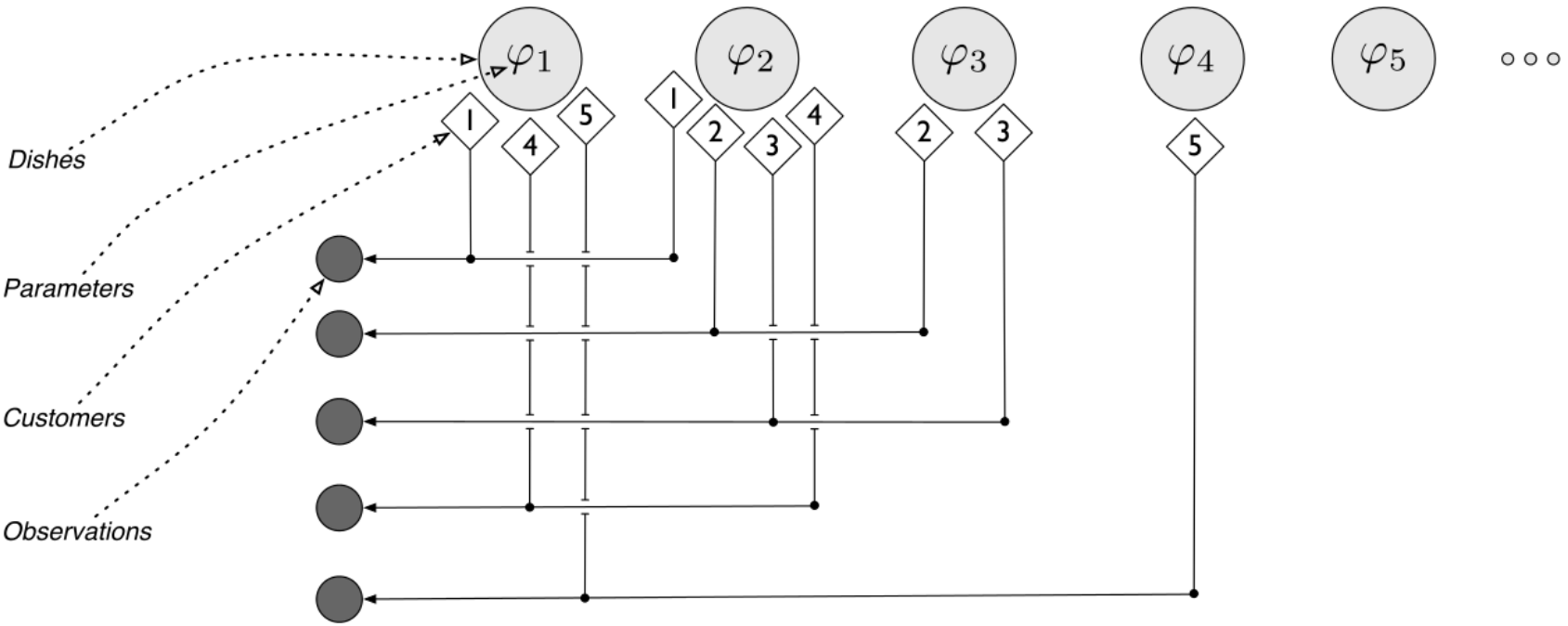
# Скрытые факторы

- Рассмотрим $\mathbf{Z}$ как бесконечную матрицу с $M$ строками (по размерности наблюдений) и бесконечным числом столбцов (потенциальные факторы).
- Нам нужно расставить там 0 и 1 так, чтобы ненулевых столбцов получалось конечное число, и априорно меньшее число столбцов было бы более вероятно.
- Китайский ресторан не подойдёт, т.к. тогда получится, что в каждой строке только одна единица (один фактор выражен); нам же нужно, чтобы в одной точке были выражены несколько факторов.

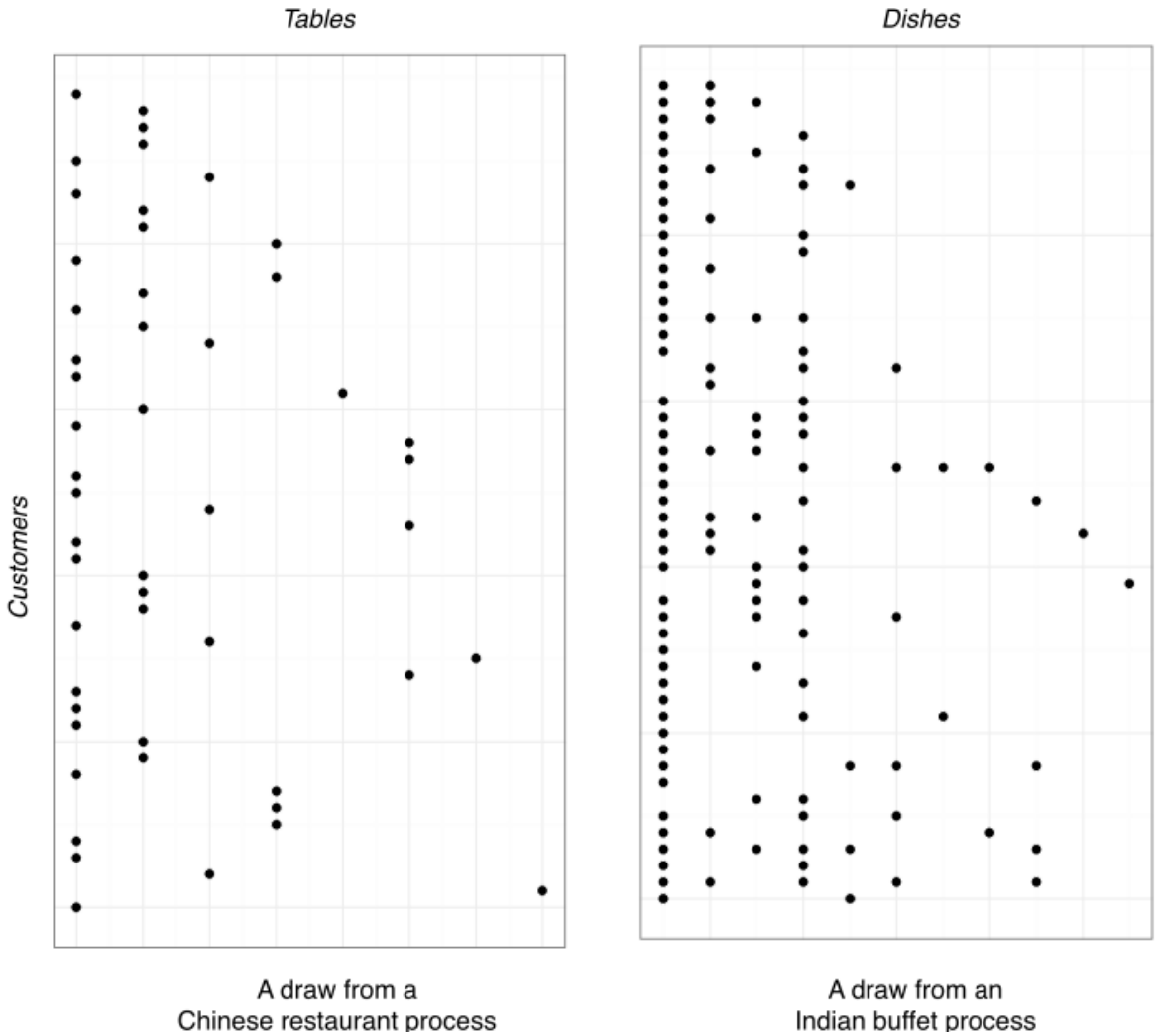
# Скрытые факторы

- *Процесс индийского буфета* (Indian buffet process):
  - $n$-й посетитель (размерность) подходит к шведскому столу, где стоит бесконечная последовательность разных блюд;
  - пробует каждое из блюд, которые уже пробовали другие посетители, с вероятностью, пропорциональной тому, сколько посетителей их пробовали;
  - после этого пробует ещё $\mathrm{Poisson}(\alpha/n)$ новых блюд, которые другие раньше не пробовали.
- Для индийского буфета тоже можно разработать алгоритм вывода на сэмплировании по Гиббсу.

# Картинка из [Gershman, Blei, 2012]



# Картинка из [Gershman, Blei, 2012]



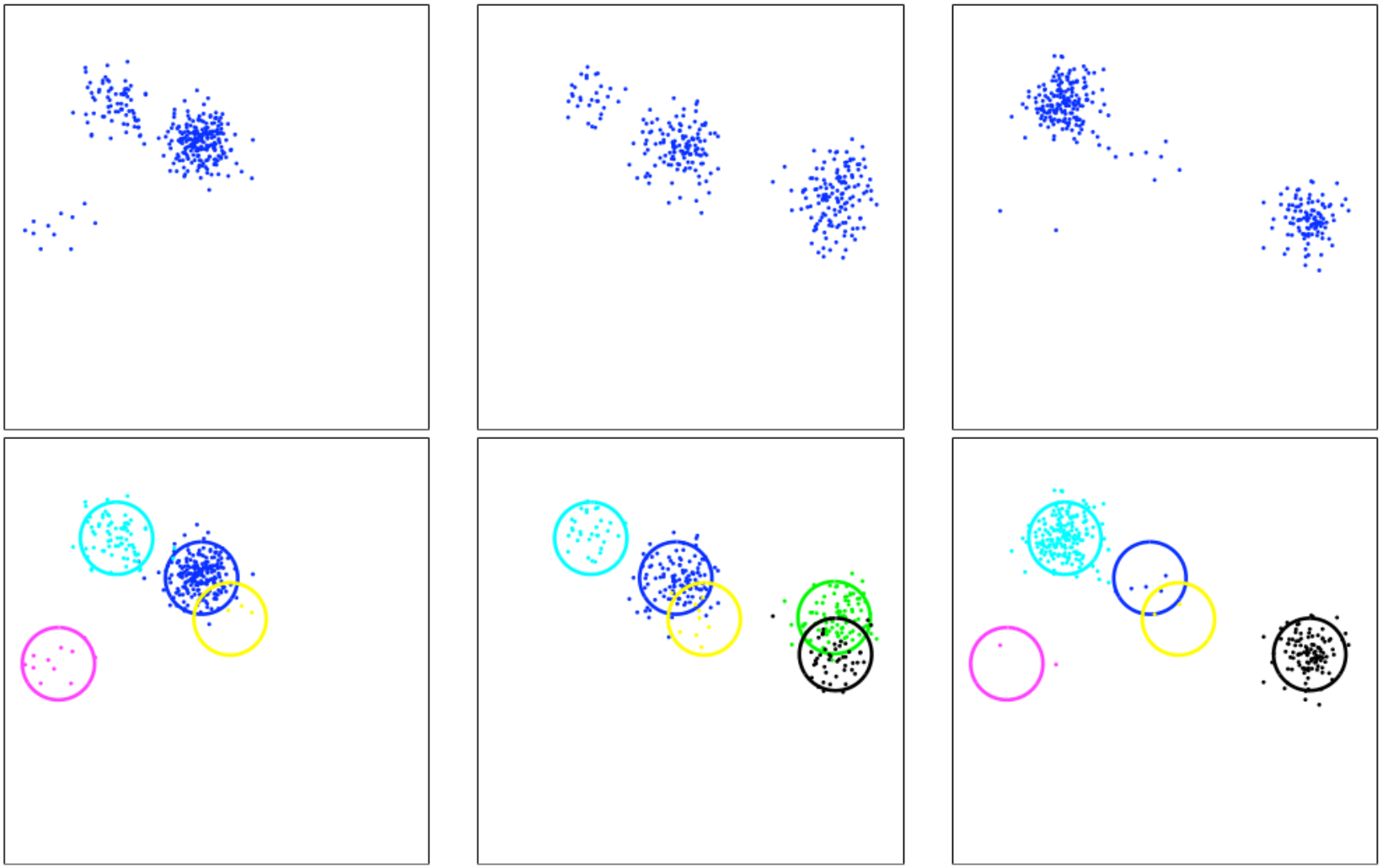
# Применения

- У процесса индийского буфета тоже много применений:
  - [Griffiths, Ghahramani, 2005]: модели со скрытыми факторами;
  - [Meeds et al., 2007]: разложение матриц для коллаборативной фильтрации;
  - [Knowles, Ghahramani, 2007]: анализ независимых компонент;
  - [Wood et al., 2006]: медицинская диагностика (обнаружение причин);
  - [Chu et al., 2006]: обнаружение комплексов белков (protein complex discovery);
  - ...

# Иерархические процессы Дирихле

- Иногда возникает ситуация, когда у нас несколько датасетов, в каждом свои кластеры, но иногда встречаются общие.
- То есть получаются как бы “группы кластеров”, в которых некоторые кластеры общие, а некоторые свои.

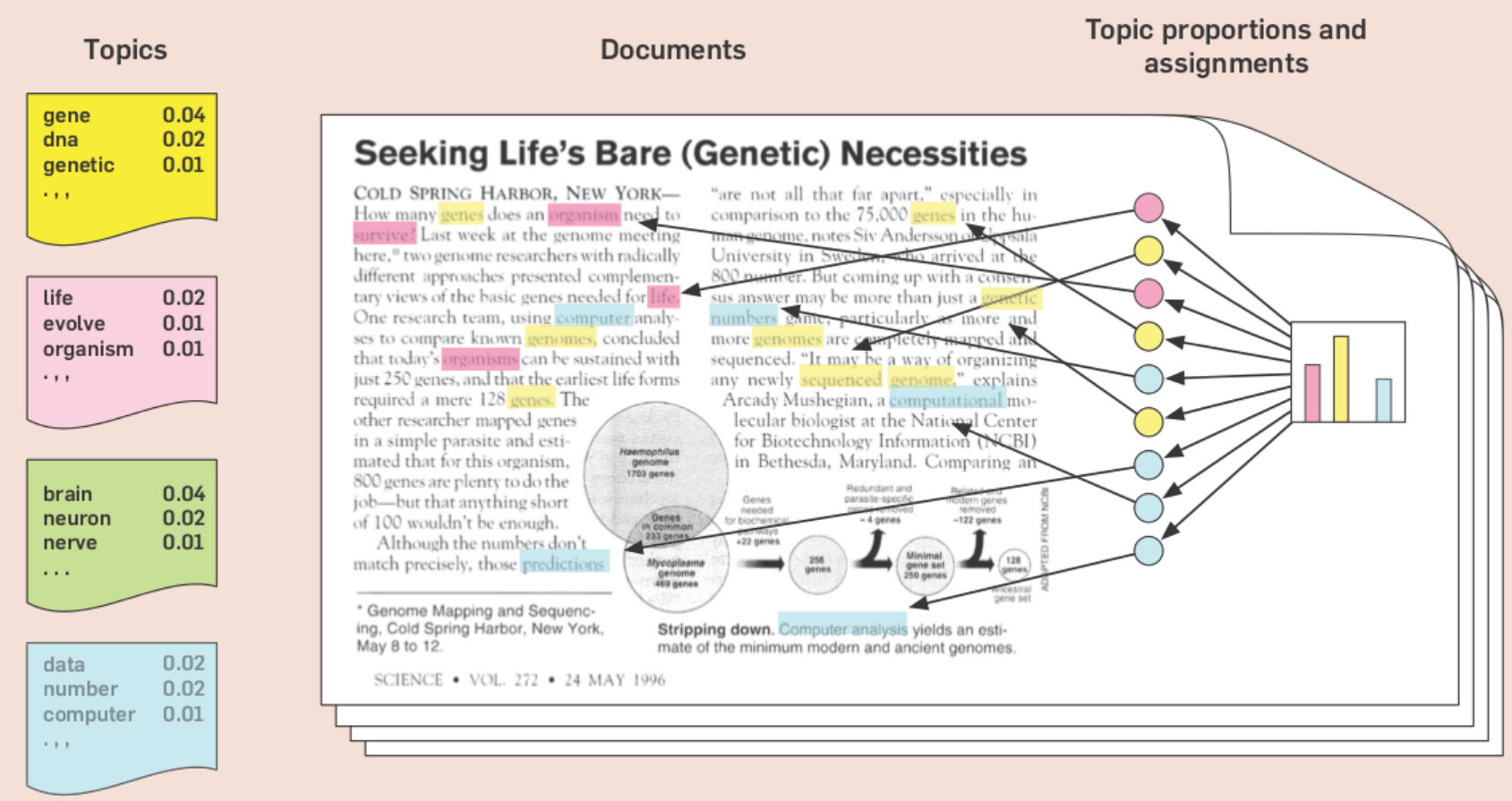
# Картинка из [Teh, 2009]



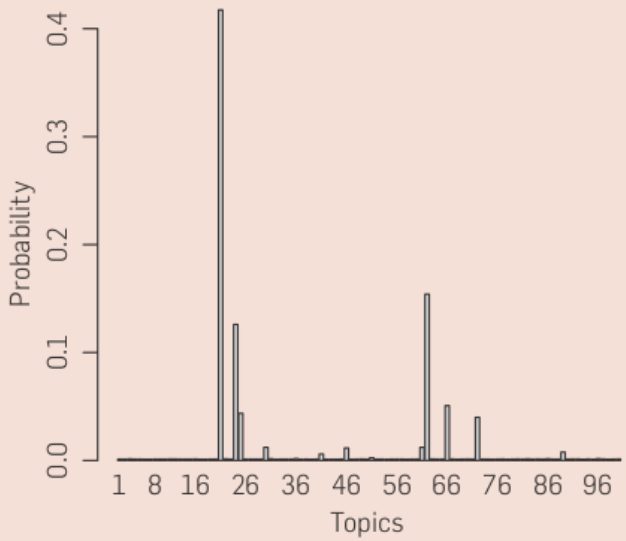
# LDA

- Пример: тематическое моделирование (topic modeling).
- В наивных подходах к категоризации (кластеризации) текстов, один документ принадлежит одной теме.
- Латентное размещение Дирихле (latent Dirichlet Allocation, LDA): разумно предполагаем, что один документ касается нескольких тем:
  - тема – это (мультиномиальное) распределение на словах (в модели bag-of-words);
  - документ – это (мультиномиальное) распределение на темах.

# Картинка из [Blei, 2012]



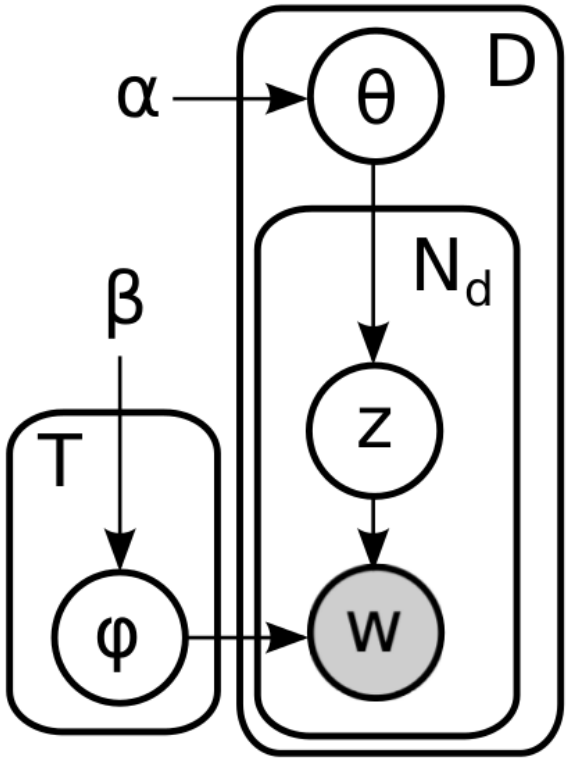
# Картинка из [Blei, 2012]




| "Genetics" | "Evolution" | "Disease" | "Computers" |
|---|---|---|---|
| human | evolution | disease | computer |
| genome | evolutionary | host | models |
| dna | species | bacteria | information |
| genetic | organisms | diseases | data |
| genes | life | resistance | computers |
| sequence | origin | bacterial | system |
| gene | biology | new | network |
| molecular | groups | strains | systems |
| sequencing | phylogenetic | control | model |
| map | living | infectious | parallel |
| information | diversity | malaria | methods |
| genetics | group | parasite | networks |
| mapping | new | parasites | software |
| project | two | united | new |
| sequences | common | tuberculosis | simulations |

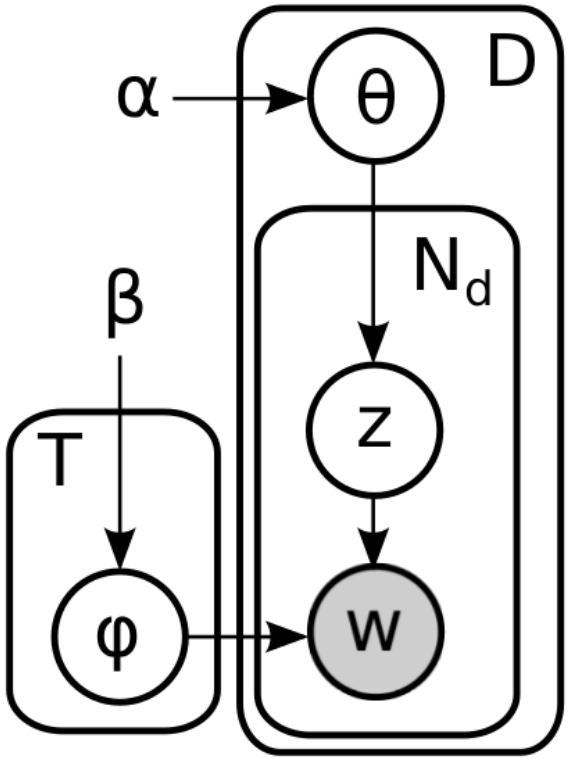
# LDA

- LDA – это иерархическая вероятностная модель:
  - на первом уровне: смесь тем $\varphi$ с весами $z$;
  - на втором уровне: мультиномиальная переменная $\theta$; её реализации $z$ – это темы каждого слова в документе.
- Называется Dirichlet allocation, потому что сопряжённые априорные для мультиномиальных – это распределения Дирихле, и мы назначаем априорные распределения Дирихле $\alpha$ и $\beta$ для параметров модели.



# LDA

- Порождающая модель LDA:
  - выбрать размер документа $N \sim p(N \mid \xi)$;
  - выбрать распределение тем $\theta \sim \mathrm{Di}(\alpha)$;
  - для каждого из $N$ слов $w_n$:
    - выбрать тему для этого слова $z_n \sim \mathrm{Mult}(\theta)$;
    - выбрать слово $w_n \sim p(w_n \mid \varphi_{z_n})$ по соответствующему мультиномиальному распределению.

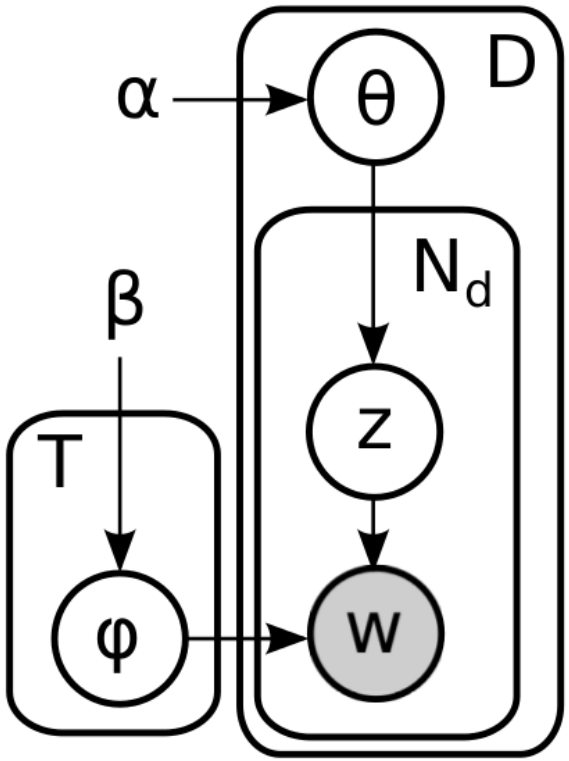


# LDA

- И совместное распределение модели получается как

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

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




- Как бы нам теперь непараметрически сделать так, чтобы число тем не нужно было выбирать?
- Обычный процесс Дирихле не пойдёт, потому что темы у документов общие.

# Иерархические процессы Дирихле

- Хорошо было бы использовать процессы Дирихле в каждой группе с общим априорным распределением:
  - из общего априорного распределения выбрать параметры кластеров для каждой группы;
  - дальше используем их как обычно в процессе Дирихле в каждой группе независимо.
- Но есть проблема: априорное распределение непрерывное, и кластеры не получатся общими.

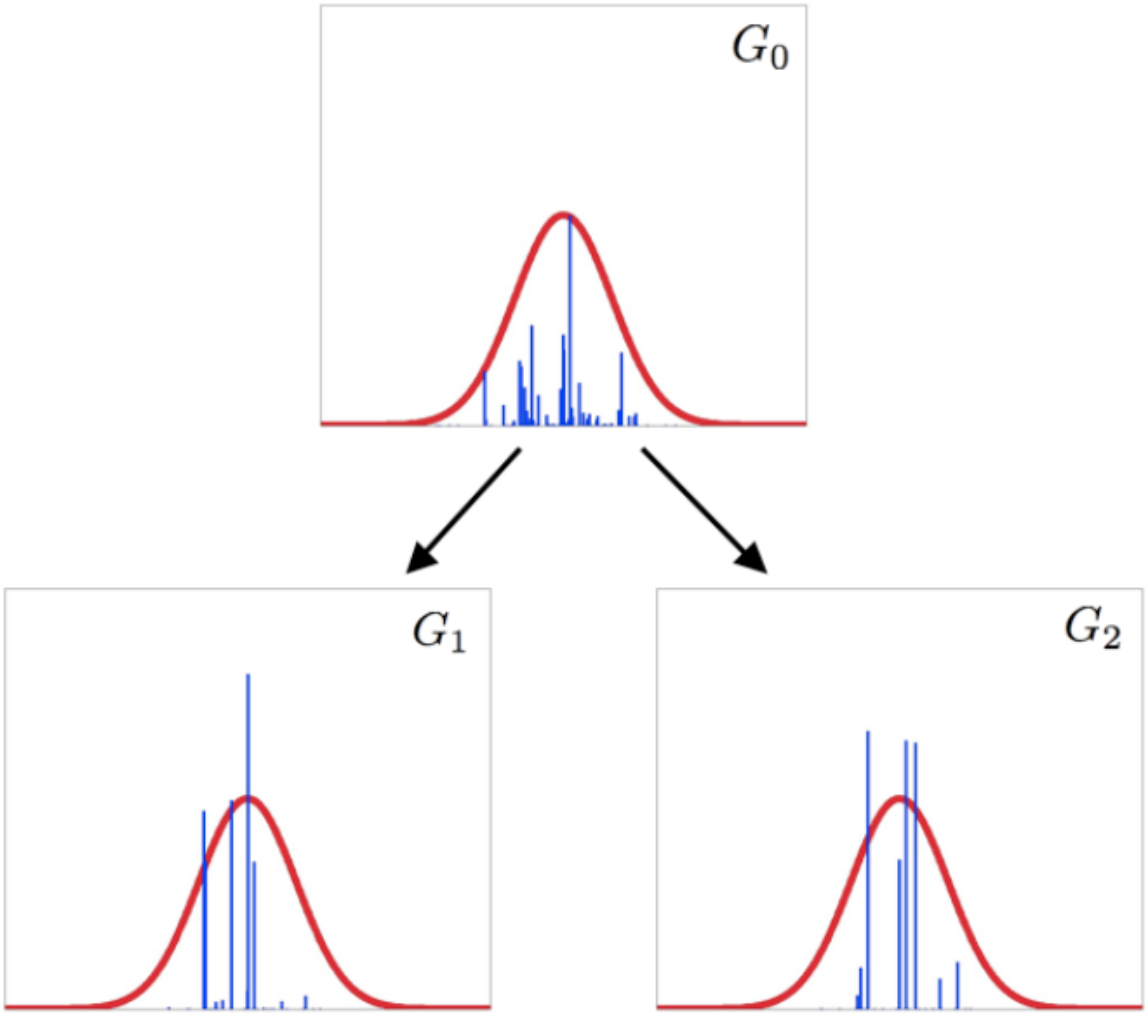
# Иерархические процессы Дирихле

- Решение: пусть априорное распределение будет дискретным, и его мы породим из процесса Дирихле, а потом из него уже наберём кластеров для каждой группы:

$$G_0 \sim \mathrm{DP}(\alpha_0, H),$$
$$G_1, G_2 \sim \mathrm{DP}(\alpha, G_0).$$

- Точно так же можно любые сложные иерархии строить.

# Картинка из [Teh, 2009]



# Франшиза китайских ресторанов

- Аналогия продолжается – теперь *франшиза китайских ресторанов*:
  - есть множество ресторанов из одной франшизы, у которых общее меню;
  - в каждом ресторане процесс рассаживания свой, независимый;
  - но когда надо выбирать новое блюдо для нового столика, его выбирают из общего меню.

# Применения

- Применения HDP:
  - тематическое моделирование (topic modeling): документы моделируются DP-смесями тем, а темы общие по всему корпусу;
  - бесконечные скрытые марковские модели (HMM): HMM с бесконечным множеством состояний и переходами из любого состояния в любое;
  - обучение дискретных структур, выделение общих частей и т.п.

# Процесс Питмана–Йора

- В китайском ресторане вероятность сгенерировать новый столик убывает пропорционально числу посетителей, и в результате с ростом числа посетителей число столиков растёт логарифмически, как $O(\alpha \log n)$.
- Но как известно, на самом деле в жизни часто встречается степенной закон.
- Например, если моделировать встречаемость слов процессом Дирихле, получится не очень, потому что частота слов затухает по закону Ципфа, т.е. степенному, а не экспоненциальному.

# Процесс Питмана–Йора

- Чтобы это смоделировать, надо, чтобы новый столик становился более вероятен.
- Процесс Питмана–Йора (Pitman–Yor process):
  - $n$-й посетитель садится за каждый из уже занятых столиков с вероятностью, пропорциональной тому, сколько там сидят человек, либо садится за пустой столик с вероятностью, пропорциональной $\alpha + dn$:

$$p(c_n = k \mid c_1, \ldots, c_{n-1}) = \begin{cases} \frac{m_k}{n-1+\alpha+dn}, & \text{если } k \leq K_{n-1}, \\ \frac{\alpha+dn}{n-1+\alpha+dn}, & \text{если } k = K_{n-1}. \end{cases}$$

- Здесь $0 \leq d \leq 1$ – новый параметр; число столиков будет расти как $O(\alpha n^d)$.

# Применения

- Применения процесса Питмана–Йора:
  - применения в NLP: моделирование слов, биграмм, триграмм;
  - иерархические процессы Питмана–Йора: моделирование текстов $k$-граммами с нефиксированным $k$;
  - сегментация изображений и другие применения к естественно возникающим датасетам со степенным законом появления новых объектов.

# Резюме

- Байесовский вывод с априорными распределениями – нужен для того, чтобы не получалось оверфиттинга; это концептуальная замена регуляризации.
- Непараметрические байесовские модели – концептуальная замена model selection; вводим априорное распределение на всевозможных наборах параметров разного размера (бесконечно большом числе), так, чтобы наборы с меньшим числом параметров имели преимущество.
- Варианты: китайский ресторан (один посетитель – одно блюдо), индийский буфет (один посетитель – набор блюд), иерархические процессы Дирихле (разные рестораны с общими столиками), процесс Питмана–Йора (больше разных столиков, степенной закон).

# Что почитать

- Сборник, в котором много полезных обзоров:
  - Hjort, N., Holmes, C., Müller, P., and Walker, S., editors, Bayesian Nonparametrics. Cambridge University Press.
- Обзоры, подробные статьи, диссертации:
  - Samuel J. Gershman, David M. Blei. A tutorial on Bayesian nonparametric models. Journal of Mathematical Psychology 56 (2012) 1–12.
  - Radford M. Neal. Markov chain sampling methods for Dirichlet process mixture models. Technical Report No. 9815, Dept. of Statistics, University of Toronto, 1998.
  - Hjort, Nils Lid (2003). Topics in nonparametric Bayesian statistics, in *Highly Structured Stochastic Systems*, Oxford University Press, Chap. 15, pp. 455–487.

# Что почитать

- Обзоры, подробные статьи, диссертации:
  - Cowans, P. Information retrieval using hierarchical Dirichlet processes. Proc. Annual International Conference on Research and Development in Information Retrieval, vol. 27, 2004, pp. 564–565.
  - Ghahramani, Z., Griffiths, T. L., and Sollich, P. (2007). Bayesian nonparametric latent feature models (with discussion and rejoinder). In Bayesian Statistics, vol. 8.
  - Goldwater, S. (2006). Nonparametric Bayesian Models of Lexical Acquisition. PhD thesis, Brown University.
  - Görür, D. (2007). Nonparametric Bayesian Discrete Latent Variable Models for Unsupervised Learning. PhD thesis, Technischen Universität Berlin.
  - Rasmussen, C. E. and Williams, C. K. I. (2006). Gaussian Processes for Machine Learning. MIT Press.

# Что почитать

- Презентации:
  - Michael I. Jordan. Dirichlet Processes, Chinese Restaurant Processes, And All That.
    `http://videolectures.net/icml05_jordan_dpcrp/`
  - Yee Whye Teh. Modern Bayesian Nonparametrics.
    `http://www.youtube.com/watch?v=F0_ih7THV94`
  - Yee Whye Teh. Bayesian Nonparametrics.
    `http://videolectures.net/mlss2011_teh_nonparametrics/`
  - Pierre Chainais. Bayesian nonparametric approaches: an introduction.
  - Mark Johnson. Chinese Restaurant Processes.
  - Khalid El-Arini. Dirichlet Processes: a Gentle Tutorial.

# Thank you!

**Спасибо за внимание!**