Школа анализа данных

БММО

Домашнее задание №3

$$p(X, Z, v, \theta|\alpha, a, b) = \prod_{k=1}^{\infty} \Big[ Beta(v_k|1, \alpha) \prod_{d=1}^{D} Beta(\theta_k^d|a, b) \Big] \prod_{n=1}^{N} \prod_{k=1}^{\infty} \left( \prod_{d=1}^{D} \theta_k^{dx_n^d} (1-\theta_k^d)^{1-x_n^d} v_k \prod_{i=1}^{k-1} (1-v_i) \right)^{[z_n=k]}$$

# 1 Вариационный вывод

$$q(Z)q(\theta)q(v) \approx p(Z, \theta, v|X, \alpha, a, b) = \frac{p(X, Z, \theta, v|\alpha, a, b)}{p(X|\alpha, a, b)}$$

Воспользуемся формулами вариационного вывода

## 1.1 Расчет $q(z)$

$$\log q(z) \propto E_{q(\theta)q(v)} \sum_{n=1}^{N} \sum_{k=1}^{\infty} [z_n = k] \big( \sum_{d=1}^{D} [x_n^d \log \theta_k^d + (1 - x_n^d) \log(1 - \theta_k^d)] + \log v_k + \sum_{i=1}^{k-1} log(1 - v_i) \big) =$$

$$= \sum_{n=1}^{N} \sum_{k=1}^{\infty} [z_n = k] \big( E_{q(\theta)} \log p(x_n|\theta_k) + E_{q(v)} \log v_k + \sum_{i=1}^{k-1} E_{q(v)} log(1 - v_i) \big) \tag{1}$$

$$q(z) \sim \prod_{n=1}^{N} \prod_{k=1}^{\infty} \exp \big( [z_n = k] (E_{q(\theta)} \sum_{d=1}^{D} [x_n^d \log \theta_k^d + (1-x_n^d) \log(1-\theta_k^d)] + E_{q(v)} \log v_k + \sum_{i=1}^{k-1} E_{q(v)} log(1-v_i)) \big)$$

$$q(z_n = k) \sim \exp \big( E_{q(\theta)} \sum_{d=1}^{D} [x_n^d \log \theta_k^d + (1 - x_n^d) \log(1 - \theta_k^d)] + E_{q(v)} \log v_k + \sum_{i=1}^{k-1} E_{q(v)} log(1 - v_i) \big)$$

$$q(z_n = k) = \frac{\exp \big( \sum_{d=1}^{D} [x_n^d E_{q(\theta)} \log \theta_k^d + (1 - x_n^d) E_{q(\theta)} \log(1 - \theta_k^d)] + E_{q(v)} \log v_k + \sum_{i=1}^{k-1} E_{q(v)} log(1 - v_i) \big)}{\sum_{p=1}^{\infty} \exp \big( \sum_{d=1}^{D} [x_n^d E_{q(\theta)} \log \theta_k^d + (1 - x_n^d) E_{q(\theta)} \log(1 - \theta_p^d)] + E_{q(v)} \log v_p + \sum_{i=1}^{p-1} E_{q(v)} log(1 - v_i) \big)} = r_{nk}$$

$E_{q(z)}[z_n = k] = r_{nk}$

## 1.2 Расчет $q(\theta)$

$$\log q(\theta) \propto E_{q(v)q(z)} \sum_{k=1}^{\infty}\sum_{d=1}^{D}((a-1)log(\theta_k^d)+(b-1)log(1-\theta_k^d))+$$
$$\sum_{n=1}^{N}\sum_{k=1}^{\infty}[z_n=k]\sum_{d=1}^{D}[x_n^d\log\theta_k^d+(1-x_n^d)log(1-\theta_k^d)]= \quad (2)$$
$$=\sum_{d=1}^{D}\sum_{k=1}^{\infty}((a-1+\sum_{n=1}^{N}r_{nk}x_n^d)log(\theta_k^d)+(b-1+\sum_{n=1}^{N}r_{nk}(1-x_n^d))log(1-\theta_k^d))$$
$$\theta_k^d \sim Beta(a+\sum_{n=1}^{N}r_{nk}x_n^d, b+\sum_{n=1}^{N}r_{nk}(1-x_n^d)) = Beta(a_{\theta dk}, b_{\theta dk})$$

## 1.3 Расчет $q(v)$

$$\log q(v) \propto E_{q(\theta)q(z)} \sum_{k=1}^{\infty}((1-1)log(v_k)+(\alpha-1)log(1-v_k))+$$
$$\sum_{n=1}^{N}\sum_{k=1}^{\infty}[z_n=k][\log v_k+\sum_{i=1}^{k-1}\log(1-v_i)]= \quad (3)$$
$$=\sum_{k=1}^{\infty}[(1-1+\sum_{n=1}^{N}r_{nk})\log v_k+(\alpha-1+\sum_{n=1}^{N}\sum_{i>k}r_{ni})]$$
$$v_k \sim Beta(1+\sum_{n=1}^{N}r_{nk}, \alpha+\sum_{n=1}^{N}\sum_{i>k}r_{ni}) = Beta(a_{vk}, b_{vk})$$

## 1.4 Достаточные статистики

Теперь нужно воспользоваться свойством для распределения из экспоненциального класса (в нашем случае бета-распределение)
Пусть $x \sim Beta(a,b)$:

$$p(x) = \frac{x^{a-1}(1-x)^(b-1)}{B(a,b)} = \frac{1}{B(a,b)}exp((a-1)\log x+(b-1)\log(1-x))$$

$$\theta_1 = a-1, \theta_2 = b-1$$
$$E\log x = \frac{d\log B(\theta_1+1,\theta_2+1)}{d\theta_1} = \frac{d\log\Gamma(a)-\log\Gamma(a+b)}{da} = \psi(a)-\psi(a+b)$$

аналогично

$$E\log(1-x) = \psi(b)-\psi(a+b)$$

Теперь мы можем рассчитать $r_{nk}$:

$$r_{nk} = \frac{\exp\big(\sum_{d=1}^{D}[\psi(b_{\theta dk})-\psi(a_{\theta dk}+b_{\theta dk})+x_n^d(\psi(a_{\theta dk})-\psi(b_{\theta dk}))]+\psi(a_{vk})-\psi(a_{vk}+b_{vk})+\sum_{i=1}^{k-1}(\psi(b_{vk})-\psi(a_{vk}+b_{vk})}{\sum_{p=1}^{\infty}(..)} \quad (4)$$

### 1.5 Вариационная нижняя оценка

$$\mathbb{L}(q) = E_{q(z)q(\theta)q(v)} \log P(X, Z, v, \theta|\alpha, a, b) - E_{q(z)q(\theta)q(v)} \log q(z, v\theta) =$$

$$= \sum_{k=1}^{\infty} \Bigg[ (1 - a_{vk})(\psi(a_{vk}) - \psi(a_{vk} + b_{vk})) + (\alpha - b_{vk})(\psi(a_{vk}) - \psi(a_{vk} + b_{vk})) + \log B(a_{vk}, b_{vk}) - \log B(1, \alpha) +$$

$$+ \sum_{d=1}^{D} \big( (a - a_{\theta dk})(\psi(a_{\theta dk}) - \psi(a_{\theta dk} + b_{\theta dk})) + (b - b_{\theta dk})(\psi(b_{\theta dk}) - \psi(a_{\theta dk} + b_{\theta dk})) + \log B(a_{\theta dk}, b_{\theta dk}) - \log B(a, b) \big) \Bigg] +$$

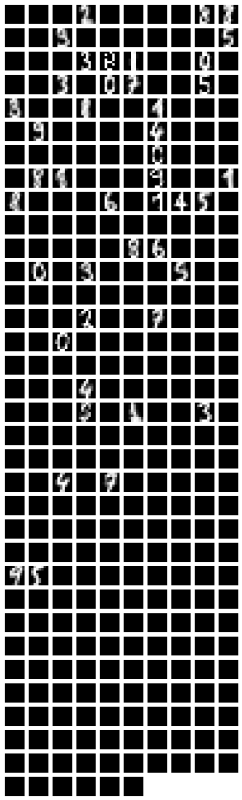
$$+ \sum_{n=1}^{N} \sum_{k=1}^{\infty} r_{nk} \Bigg[ \sum_{d=1}^{D} x_n^d (\psi(a_{\theta dk}) - \psi(a_{\theta dk} + b_{\theta dk})) + (1 - x_n^d)(\psi(b_{\theta dk}) - \psi(a_{\theta dk} + b_{\theta dk})) + \psi(a_{vk}) - \psi(a_{vk} + b_{vk}) +$$

$$+ \sum_{i=1}^{k-1} (\psi(b_{vk}) - \psi(a_{vk} + b_{vk})) - \log r_{nk} \Bigg] \tag{5}$$

## 2 Протестировать полученный алгоритм на небольшой подвыборке Digits. Качественно охарактеризовать, как влияют параметры модели $\alpha$, $a$, $b$ на вид и количество образующихся кластеров. Выбрать конкретные значения $\alpha$, $a$, $b$ и использовать их во всех дальнейших экспериментах

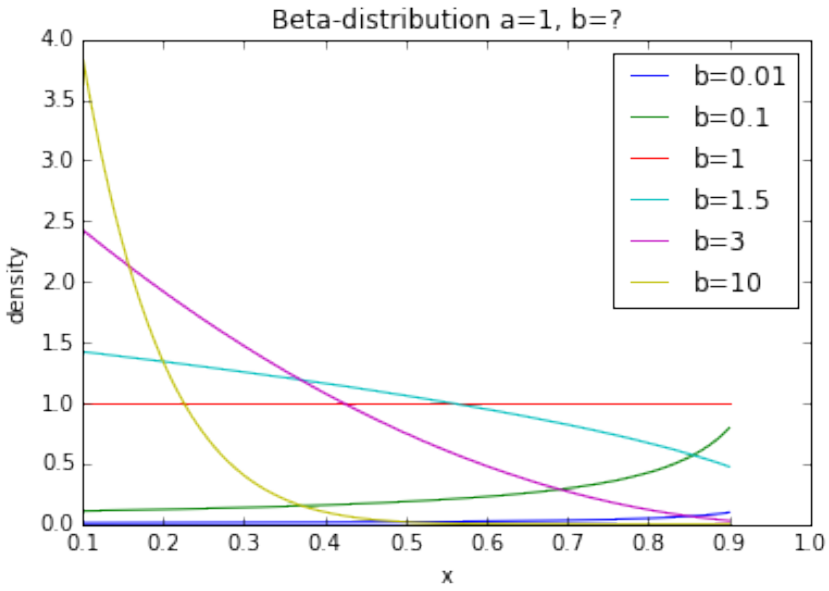
Рассмотрим подвыборку длины 3000

## 2.1 Выбор константы для вычисления числа кластеров



На изображении выше видно, что взятая константа, равная 100 слишком большая, т.к. быстро становится много пустых кластеров, и в конце они практичеки все нулевые. Будем испольвзовать далее предложенную константу, равную 10

# 3 Выбор $\alpha$




Из графика выше следует, чем выше значение параметра $\alpha$ ($b$) тем больше вероятность, что $v$ примет значение близкое к нулю, т.е. вероятность очередного кластера для любого объекта будет мала, чем меньше $\alpha$ тем вероятнее, что очереденой $x$ сгенерирован из данного кластера. Аналогично для малых $\alpha$ - только в этом случае, напротив вероятность первых кластеров будет очень высокая, а на остальные ничего не останется (следует из метода ломки палки) Таким образом, брать $\alpha > 1.5$ неразумно.

Посмотрим, какие кластеры будут получаться взависимости от $\alpha$

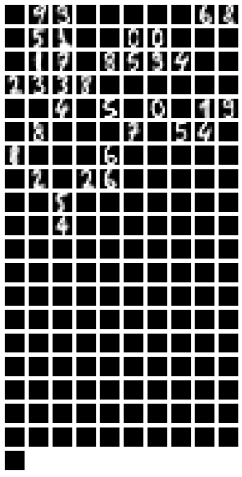


Рис. 3.0.1: $\alpha = 3$

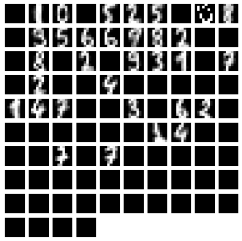


Рис. 3.0.2: $\alpha = 1.3$

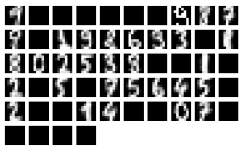


Рис. 3.0.3: $\alpha = 0.7$

Рис. 3.0.4: $\alpha = 0.15$

В нашей задаче мы знаем, что кластеров должно быть не очень много, поэтому будем использовать $\alpha = 0.15$ (тем более на большой выборке число кластеров увеличится)

## 3.1 Выбор $a, b$

### 3.1.1 $a = b$

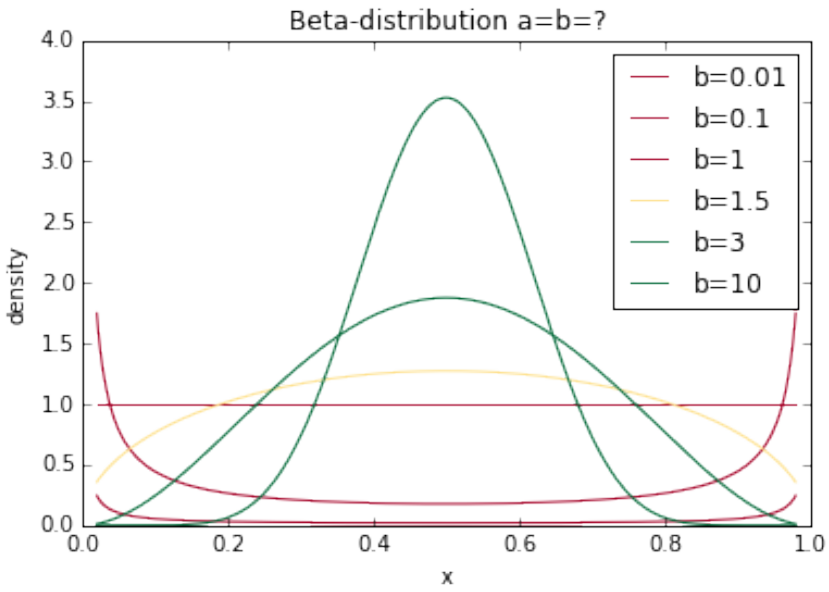



Рис. 3.1.1: $a = b$

Рис. 3.1.2: $a = b = 1e - 5$

Рис. 3.1.3: $a = b = 10$

При больших значениях $a, b$ получаем, что значение каджого пиксела стремится быть сконцентрировано вблизи значения 0.5 (серый фон), а при малых $a = b$ значения пикселов сконцентрированы вблизи 0 и 1, это то, что нам нужно. Поэтому выбираем $a, b < 1$

### 3.1.2 $a \neq b$

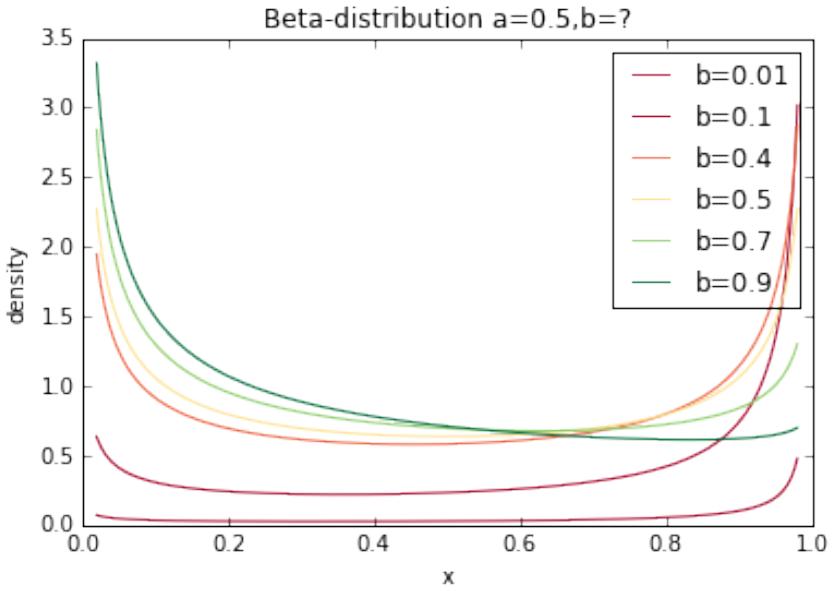



Рис. 3.1.4: $a = b$

Если $b > a$ получаем больше черных пикселов, если $b < a$, больше белых пикселов.

Рис. 3.1.5: $a = 0.5b = 0.001$

Рис. 3.1.6: $a = 0.5b = 0.99$

На рисунках разница не заметна, видимо для $a, b < 1$ метод хорошо обучается и стартовое значение нивелируется. В таком случае далее будем использовать $a = 0.5b = 0.5$

## 4 Запустить алгоритм на полной выборке Digits. Сколько и каких получилось кластеров? Рассмотреть величины q(zn = k) в качестве признаков n-го объекта выборки, обучить любой классификатор на образованных данных, построить матрицу точности на контрольной выборке. Проинтерпретировать результат

Рис. 4.0.1: Центры кластеров На всех данных (14 кластеров)

2 раза - 5
2 раза - 9
1 раз - 4
1 раз - 0
2 раза - 2
2 раза - 1
1 раз - 8
1 раз - 7
1 раз - 3
1 раз - 6

Присутствуют все 10 цифр, это хорошо.
Обучим классификатор (SVC) на признаках z: получили accuracy=0.84814 ($test_size = 0.3$)
Рассмотрим матрицу Precision (строки-истиррые значения, столбцы-предсказанные

| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|---|---|---|---|---|---|---|---|---|---|
| 0.9773 | 0.0000 | 0.0000 | 0.0000 | 0.0000 | 0.0286 | 0.0000 | 0.0000 | 0.0000 | 0.0462 |
| 0.0000 | 0.9804 | 0.2037 | 0.0000 | 0.0000 | 0.0143 | 0.0000 | 0.0000 | 0.0526 | 0.0000 |
| 0.0227 | 0.0000 | 0.7407 | 0.0000 | 0.0000 | 0.0000 | 0.0000 | 0.0000 | 0.1053 | 0.0923 |
| 0.0000 | 0.0000 | 0.0185 | 1.0000 | 0.0000 | 0.0143 | 0.0000 | 0.0290 | 0.0702 | 0.0923 |
| 0.0000 | 0.0000 | 0.0000 | 0.0000 | 0.9574 | 0.0143 | 0.0000 | 0.0580 | 0.0000 | 0.0000 |
| 0.0000 | 0.0000 | 0.0000 | 0.0000 | 0.0213 | 0.8429 | 0.0000 | 0.0000 | 0.0000 | 0.0769 |
| 0.0000 | 0.0196 | 0.0000 | 0.0000 | 0.0213 | 0.0000 | 1.0000 | 0.0000 | 0.0351 | 0.0000 |
| 0.0000 | 0.0000 | 0.0000 | 0.0000 | 0.0000 | 0.0000 | 0.0000 | 0.7971 | 0.0000 | 0.0000 |
| 0.0000 | 0.0000 | 0.0370 | 0.0000 | 0.0000 | 0.0429 | 0.0000 | 0.0290 | 0.7018 | 0.0308 |
| 0.0000 | 0.0000 | 0.0000 | 0.0000 | 0.0000 | 0.0429 | 0.0000 | 0.0870 | 0.0351 | 0.6615 |

Наиболее интересно, что часто предсказываем:
для цифры 1 значение 2 (см. класер 6)

Для цифры 9 значение 7 (см. кластер 10)
для цифры 3 значение 8 (см. кластер 9)
для цифры 3,4,6 значение 9 (см. кластер 9)

## 5 Протестировать алгоритм в условиях, когда обучающие данные приходят порциями. Как меняются кла- стеры по мере добавления данных с новыми видами цифр?



Рис. 5.0.1: Каждая строка это дообучение 150 образцами

Когда приходят новые виды цифр кластеры немного расплываются (облако становится менее четким). В этом случае следовало бы увеличить число кластеров, но в случае дообучения число кластеров постоянно. С другой стороны мы получаем дополнительную информацию - цифры становятся более "общими"