

**Московский Государственный Университет  
им. М.В. Ломоносова**

Факультет Вычислительной математики и кибернетики  
Кафедра Математических Методов Прогнозирования

**Самбурский Александр. 417 группа.  
Вариант 3.**

**Практическое задание 1.  
Байесовские рассуждения  
Отчёт о проделанной работе.**

Москва 2021

# 1 Введение

Данный отчёт представляет собой результаты, полученные при решении задачи исследования зависимостей распределений случайных величин, образующих единую вероятностную модель.

В работе изучался по большей части характер уточнения апостериорных распределений случайных величин при последовательном поступлении новой косвенной информации, в том числе общий вид распределений, их математические ожидания и дисперсии. Помимо этого рассматривался вопрос допустимости упрощённого приближения вероятностной модели более простыми распределениями, а также целесообразность такого перехода, что отразило исследование двух вероятностных моделей - "точной" и "аппроксимирующей".

В качестве общей цели выдвигалось изучение метода Байесовского вывода на примере этих вероятностных моделей, а также его возможности и свойства.

## Содержание

<b>1 Введение</b>	<b>1</b>
<b>2 Постановка задачи</b>	<b>2</b>
2.1 Вероятностные модели . . . . .	2
2.2 Условия корректности перехода между моделями . . . . .	3
<b>3 Вывод распределений</b>	<b>3</b>
3.1 Условные и априорные распределения . . . . .	3
3.2 Вывод апостериорного распределения $p(b d_1, \dots, d_N)$ . . . . .	5
3.3 Вывод апостериорного распределения $p(b a, d_1, \dots, d_n)$ . . . . .	6
<b>4 Подсчёт моментов априорных распределений</b>	<b>6</b>
4.1 Вспомогательные формулы . . . . .	6
4.2 Аналитический вывод моментов . . . . .	7
4.2.1 Математические ожидания . . . . .	7
4.2.2 Дисперсии . . . . .	8
<b>5 Характер уточнения апостериорного распределения</b>	<b>11</b>
<b>6 Сравнение программных реализаций моделей</b>	<b>15</b>
<b>7 Сравнение вероятностных моделей</b>	<b>16</b>
<b>8 Выводы</b>	<b>18</b>
<b>9 Список литературы:</b>	<b>18</b>

## 2 Постановка задачи

### 2.1 Вероятностные модели

При исследовании были рассмотрены две вероятностные модели, описывающие посещаемость студентами лекций. Первая из них является более точной, однако требующей более трудоёмкие вычисления для подсчёта распределений. Вторая вероятностная модель аппроксимирует первую более простым характером распределений входящих в неё случайных величин. Как будет показано далее, естественным образом реализация первой модели выигрывает в точности вычисляемых характеристик распределения, но зато программная реализация второй - более эффективна.

Пусть в течение курса было проведено  $N$  лекций. На каждую из лекций могли попасть студенты профильного факультета, а также студенты с остальных факультетов. Из этого числа на каждую лекцию приходила только часть студентов, при этом с некоторой вероятностью в журнал посещаемости студенты могли занести своих товарищей. На основе данных зависимых случайных величин требуется производить анализ их распределений.

Ниже используются следующие обозначения:

- $a$  - количество студентов, поступивших на профильный факультет
- $b$  – количество студентов других факультетов
- $c_n$  - число студентов, пришедших на  $n$ -ую лекцию
- $d_n$  - число студентов, попавших в журнал посещаемости

Первая вероятностная модель (в дальнейшем будем называть её третьей (3)) представляется следующим образом:

$$\begin{aligned} p(a, b, c_1, \dots, c_N, d_1, \dots, d_N) &= p(a)p(b) \prod_{n=1}^N p(d_n|c_n)p(c_n|a, b) \\ p(d_n|c_n) &\sim c_n + \text{Bin}(c_n, p_3) \\ p(c_n|a, b) &\sim \text{Bin}(a, p_1) + \text{Bin}(b, bp_2) \\ p(b) &\sim \text{Unif}[b_{min}, b_{max}] \\ p(a) &\sim \text{Unif}[a_{min}, a_{max}] \end{aligned}$$

Для второй (далее называется четвёртой (4)) вероятностной модели упрощается распределение  $p(c_n|a, b)$ :

$$\begin{aligned} p(a, b, c_1, \dots, c_N, d_1, \dots, d_N) &= p(a)p(b) \prod_{n=1}^N p(d_n|c_n)p(c_n|a, b) \\ p(d_n|c_n) &\sim c_n + \text{Bin}(c_n, p_3) \\ p(c_n|a, b) &\sim \text{Poiss}(ap_1 + bp_2) \\ p(b) &\sim \text{Unif}[b_{min}, b_{max}] \\ p(a) &\sim \text{Unif}[a_{min}, a_{max}] \end{aligned}$$

Целью работы является исследование данных вероятностных моделей - нахождение математических ожиданий и дисперсий входящих в них случайных величин, а также изучение характера уточнения апостериорных распределений при поступлении потока информации в модель. Также требуется произвести качественный анализ двух моделей, сравнение работы их программной реализации и исследование условий, при которых модели работают одинаково или, напротив, выдают различные результаты.

## 2.2 Условия корректности перехода между моделями

Здесь же вкратце и поясним, почему, и в каких случаях такая аппроксимация  $p(c_n|a, b)$  четвёртой моделью допустима. Как известно, биномиальное распределение  $Bin(n, p)$  переходит в Пуассоновское  $Poiss(\lambda)$ , когда  $n \rightarrow \infty$ ,  $p \rightarrow 0$  и  $np \rightarrow \lambda$ . Соответственно, можно приближать биномиальное распределение Пуассоновским, когда число  $n$  - велико, а  $p$  - мало. Запомним этот факт, он понадобится в дальнейшем, когда встанет вопрос о принципиальных различиях вероятностных моделей. А пока, действуя в предположениях, что допустимые значения величин  $a$  и  $b$  - достаточно велики, а вероятности  $p_1$  и  $p_2$  - достаточно малы, можно утверждать корректность приближения каждой из биномиальных компонент  $p(c_n|a, b)$ :  $Bin(a, p_1) \rightarrow Poiss(ap_1)$ ,  $Bin(b, p_2) \rightarrow Poiss(bp_2)$ . Следовательно, их сумма также представима, как сумма Пуассоновских распределённых случайных величин:  $Bin(a, p_1) + Bin(b, p_2) \rightarrow Poiss(ap_1) + Poiss(bp_2)$ .

Ещё один факт из теории вероятностей утверждает, что сумма Пуассоновских распределений тоже распределена по Пуассоновскому распределению с параметром, равным сумме параметров слагаемых случайных величин:

$$Poiss(\lambda_1) + Poiss(\lambda_2) = Poiss(\lambda_1 + \lambda_2) \quad (1)$$

Докажем его. Пусть  $\xi \sim Poiss(\lambda_1)$ ,  $\eta \sim Poiss(\lambda_2)$  - независимые случайные величины. Тогда:

$$\begin{aligned} P(\xi + \eta = n) &= \sum_{m=0}^n P(\xi = m, \eta = n - m) = \{\xi, \eta \text{ - независимы}\} = \\ &= \sum_{m=0}^n P(\xi = m)P(\eta = n - m) = \sum_{m=0}^n \frac{e^{-\lambda_1} \lambda_1^m}{m!} \times \frac{e^{-\lambda_2} \lambda_2^{n-m}}{(n-m)!} = \\ &= \frac{e^{-(\lambda_1+\lambda_2)}}{n!} \sum_{m=0}^n C_n^m \lambda_1^m \lambda_2^{n-m} = \frac{e^{-(\lambda_1+\lambda_2)}}{n!} (\lambda_1 + \lambda_2)^n \sim Poiss(\lambda_1 + \lambda_2) \end{aligned} \quad (2)$$

Тогда допустима аппроксимация  $p(c_n|a, b) \sim Bin(a, p_1) + Bin(b, p_2) \sim Poiss(ap_1 + bp_2)$  для больших значений  $a$  и  $b$  и малых вероятностях  $p_1$  и  $p_2$ .

Теперь изучим основные свойства данных распределений.

## 3 Вывод распределений

### 3.1 Условные и априорные распределения

По определению равномерного дискретного распределения:

$$p(a = k) = \begin{cases} \frac{1}{a_{max} - a_{min} + 1}, & \text{if } k \in [a_{min}, a_{max}] \\ 0, & \text{else} \end{cases} \quad (3)$$

$$p(b = k) = \begin{cases} \frac{1}{b_{max} - b_{min} + 1}, & \text{if } k \in [b_{min}, b_{max}] \\ 0, & \text{else} \end{cases} \quad (4)$$

Выведем  $p(c_n|a, b)$  для 3 модели, а потом для 4:

Напомним функцию биномиального распределения:

$$\xi \sim Bin(n, p) \implies p(\xi = k) = C_n^k p^k (1-p)^{n-k}$$

Введём обозначения:

$$\begin{aligned} c_n &= c_n^1 + c_n^2 \\ c_n^1 &\sim Bin(a, p_1) \\ c_n^2 &\sim Bin(b, p_2) \end{aligned}$$

Заметим, что распределение случайной величины  $c_n^1$  зависит только от  $a$ , а распределение  $c_n^2$  зависит только от  $b$ . Тогда:

$$\begin{aligned} p(c_n = k|a, b) &= p(c_n^1 + c_n^2 = k|a, b) = \sum_{m=\max\{0, k-b\}}^{\min\{a, k\}} p(c_n^1 = m, c_n^2 = k-m|a, b) = \\ &= \left\{ c_n^1 \text{ и } c_n^2 \text{ - независимы} \right\} = \sum_{m=\max\{0, k-b\}}^{\min\{a, k\}} p(c_n^1 = m|a) p(c_n^2 = k-m|b) = \\ &= \sum_{m=\max\{0, k-b\}}^{\min\{a, k\}} C_a^m p_1^m (1-p_1)^{a-m} C_b^{k-m} p_2^{k-m} (1-p_2)^{b-k+m} = \\ &= (1-p_1)^a (1-p_2)^{b-k} p_2^k a! b! \sum_{m=\max\{0, k-b\}}^{\min\{a, k\}} \frac{p_1^m (1-p_1)^{-m} p_2^{-m} (1-p_2)^m}{m! (k-m)! (a-m)! (b-k+m)!} \quad (5) \end{aligned}$$

Здесь границы суммирования по  $m$  позволяют удовлетворить следующие условия на множество значений  $c_n^1$  и  $c_n^2$ :  $c_n^1 = m \in [0, a]$  и  $c_n^2 = k-m \in [0, b]$ .

Выведем  $p(c_n|a, b)$  для 4 модели:

Напомним функцию пуассоновского распределения:

$$\xi \sim Poiss(\lambda) \implies p(\xi = k) = \frac{e^{-\lambda} \lambda^k}{k!}$$

Тогда:

$$p(c_n = k|a, b) = \frac{e^{-(ap_1+bp_2)} (ap_1 + bp_2)^k}{k!} \quad (6)$$

Для обеих моделей множество значений  $(c_n|a, b)$  находится в отрезке  $[0, a+b]$ . Крайние значения реализуются, когда ни одно испытание не стало успешным, и все испытания окончились успехом.

Для обеих моделей условное распределение  $p(d_n|c_n)$  выглядит следующим образом:

$$p(d_n = k|c_n) = C_{c_n}^{k-c_n} p_3^{k-c_n} (1-p_3)^{2c_n-k} = \frac{c_n!}{(k-c_n)!(2c_n-k)!} p_3^{k-c_n} (1-p_3)^{2c_n-k} \quad (7)$$

Вероятность  $p(d_n = k)$  в данной задаче равна вероятности того, что биномиальная величина  $Bin(c_n, p_3)$  равна  $k - c_n$ , то есть из  $c_n$  испытаний  $k - c_n$  окончились успехом. Отсюда и получается написанная выше формула. Множество допустимых значений  $(d_n|c_n)$  удовлетворяет условию:  $d_n \in [c_n, 2c_n]$ .

Для того, чтобы вычислить априорные распределения случайных величин, нужно воспользоваться правилом суммирования:

$$\begin{aligned} p(c_n = k) &= \sum_{a=a_{\min}}^{a_{\max}} \sum_{b=b_{\min}}^{b_{\max}} p(c_n = k|a, b) \\ p(d_n = k) &= \sum_{c_n=\lceil \frac{k}{2} \rceil}^{a_{\max}+b_{\max}} p(d_n = k|c_n) \end{aligned} \quad (8)$$

Во второй формуле границы суммирования позволяют удовлетворить условию:  $d_n \in [c_n, 2c_n]$ .

Приведём ниже два способа вычисления апостериорных распределений на величину  $b$ . Распределение  $p(b|d_1, \dots, d_N)$  будем моделировать в предположении, что вся информация о величинах  $d_1, \dots, d_N$  есть изначально - вычислять распределение будем за "один раз". Распределение  $p(b|a, d_1, \dots, d_N)$  будем вычислять, полагая, что информация о переменных  $d_1, \dots, d_N$  поступает последовательно - вычисление будет итеративным, на каждом шаге мы будем добавлять в модель очередную переменную  $d_n$ , используя уже "накопленное" апостериорное распределение из предыдущих переменных  $d_1, \dots, d_{n-1}$  в качестве априорного (подсчёт распределения "на лету"). В принципе нет причин, почему для этих распределений было выбрана именно такая расстановка способов вычисления, оба рассматриваемых метода применимы к обоим апостериорным распределениям, и в данной работе такая расстановка просто позволяет реализовать и исследовать оба метода.

### 3.2 Вывод апостериорного распределения $p(b|d_1, \dots, d_N)$

Пусть заданы все переменные  $d_1, \dots, d_N$ . Тогда:

$$\begin{aligned} p(b|d_1, \dots, d_N) &= \frac{p(b, d_1, \dots, d_N)}{p(d_1, \dots, d_N)} = \frac{\sum_a \sum_{c_1} \dots \sum_{c_N} p(a, b, c_1, \dots, c_N, d_1, \dots, d_N)}{\sum_b \sum_a \sum_{c_1} \dots \sum_{c_N} p(a, b, c_1, \dots, c_N, d_1, \dots, d_N)} = \\ &= \{ \text{По определению моделей} \} = \\ &= \frac{p(b) \sum_a p(a) \sum_{c_1} \dots \sum_{c_N} \prod_{n=1}^N p(d_n|c_n) p(c_n|a, b)}{\sum_b p(b) \sum_a p(a) \sum_{c_1} \dots \sum_{c_N} \prod_{n=1}^N p(d_n|c_n) p(c_n|a, b)}, \quad b \in [b_{min}, b_{max}] \end{aligned} \quad (9)$$

Упростим слагаемые числителя и знаменателя:

$$\begin{aligned} &\sum_{c_1} \sum_{c_2} \dots \sum_{c_N} \prod_{n=1}^N p(d_n|c_n) p(c_n|a, b) = \\ &= \sum_{c_1} \sum_{c_2} \dots \sum_{c_N} p(d_1|c_1) p(c_1|a, b) \times p(d_2|c_2) p(c_2|a, b) \times \dots \times p(d_N|c_N) p(c_N|a, b) = \\ &= \{ \text{Последовательно вносим знаки сумм в произведение} \} = \\ &= \sum_{c_1} p(d_1|c_1) p(c_1|a, b) \times \sum_{c_2} p(d_2|c_2) p(c_2|a, b) \times \dots \times \sum_{c_N} p(d_N|c_N) p(c_N|a, b) = \\ &= \prod_{n=1}^N \sum_{c_n} p(d_n|c_n) p(c_n|a, b) \end{aligned} \quad (10)$$

В итоге получаем следующую формулу для вычисления апостериорных распределений:

$$p(b|d_1, \dots, d_N) = \frac{p(b) \sum_a p(a) \prod_{n=1}^N \sum_{c_n} p(d_n|c_n) p(c_n|a, b)}{\sum_b p(b) \sum_a p(a) \prod_{n=1}^N \sum_{c_n} p(d_n|c_n) p(c_n|a, b)}, \quad b \in [b_{min}, b_{max}] \quad (11)$$

Для применения этого способа для подсчёта  $p(b|a, d_1, \dots, d_n)$ , нужно исключить из полученной дроби суммирования по  $a$  и априорное распределение  $p(a)$ :

$$p(b|a, d_1, \dots, d_N) = \frac{p(b) \prod_{n=1}^N \sum_{c_n} p(d_n|c_n) p(c_n|a, b)}{\sum_b p(b) \prod_{n=1}^N \sum_{c_n} p(d_n|c_n) p(c_n|a, b)}, \quad b \in [b_{min}, b_{max}] \quad (12)$$

### 3.3 Вывод апостериорного распределения $p(b|a, d_1, \dots, d_n)$

Теперь распишем способ последовательного вычисления  $p(b|a, d_1, \dots, d_n)$  при итеративном поступлении  $d_1, d_2, \dots, d_n$ . Это позволит вычислять апостериорное распределение для  $b$  "на лету".

$$p(b) = \begin{cases} \frac{1}{b_{max}-b_{min}+1}, & \text{if } b \in [b_{min}, b_{max}] \\ 0, & \text{else} \end{cases} \quad (13)$$

$$\begin{aligned} p(b|a, d_1, \dots, d_n) &= \frac{p(b, a, d_1, \dots, d_n)}{p(a, d_1, \dots, d_n)} = \frac{p(b, d_n|a, d_1, \dots, d_{n-1})p(a, d_1, \dots, d_{n-1})}{p(d_n|a, d_1, \dots, d_{n-1})p(a, d_1, \dots, d_{n-1})} = \\ &= \frac{p(b, d_n|a, d_1, \dots, d_{n-1})}{p(d_n|a, d_1, \dots, d_{n-1})} = \frac{\sum_{c_n} p(b, c_n, d_n|a, d_1, \dots, d_{n-1})}{\sum_{b, c_n} p(a, b, c_n, d_n|a, d_1, \dots, d_{n-1})} = \\ &= \frac{\sum_{c_n} p(d_n|c_n)p(c_n|a, b)p(b|a, d_1, \dots, d_{n-1})}{\sum_{b, c_n} p(d_1|c_1)p(c_1|a, b)p(b|a, d_1, \dots, d_{n-1})}, \quad b \in [b_{min}, b_{max}] \end{aligned} \quad (14)$$

Данные рекуррентные выражения позволяют последовательно обрабатывать поток информации. По сути каждое вычисленное апостериорное распределение  $p(b|a, d_1, \dots, d_{n-1})$  играет роль априорного для последующего вычисления  $p(b|a, d_1, \dots, d_n)$ . Плюсом такого подхода является возможность удаления информации о величинах  $(d_1, d_2, \dots, d_n)$  после подсчёта апостериорного распределения, так как вся важная информация содержится именно в нём.

Для применения этого способа для подсчёта  $p(b|d_1, \dots, d_n)$ , нужно добавить в числитель и знаменатель полученной дроби суммирования по  $a$  и априорное распределение  $p(a)$  в качестве дополнительного множителя внутри этих сумм:

$$p(b|a, d_1, \dots, d_n) = \frac{\sum_{a, c_n} p(d_n|c_n)p(c_n|a, b)p(b|a, d_1, \dots, d_{n-1})p(a)}{\sum_{a, b, c_n} p(d_1|c_1)p(c_1|a, b)p(b|a, d_1, \dots, d_{n-1})p(a)}, \quad b \in [b_{min}, b_{max}] \quad (15)$$

Итак, формулы (3)-(8) задают условные и априорные распределения на величины, входящие в 3 и 4 модели. Формулы (11), (13)-(14) позволяют реализовать подсчёт апостериорных распределений (двумя разными способами).

## 4 Подсчёт моментов априорных распределений

В качестве первичного анализа вероятностных моделей можно привести математические ожидания и дисперсии априорных распределений входящих в них случайных величин. Это позволит проводить рассуждения о среднем и разбросе модели при отсутствии какой-либо информации о реализации случайных величин. В данном разделе будут приведены формулы для вычисления данных моментов, а также приведены результаты их подсчёта программным образом по их определению.

### 4.1 Вспомогательные формулы

Для начала покажем, как искать нужные моменты случайных величин, зависящих от других случайных величин.

Пусть  $x, y$  - случайные величины.  $x$  - зависит от  $y$ . Пусть нас интересует  $\mathbb{E}(x)$ .

$$\begin{aligned}\mathbb{E}(x) &= \sum_x xP(x) = \sum_x x \left[ \sum_y P(x|y)P(y) \right] = \sum_y P(y) \left[ \sum_x xP(x|y) \right] = \\ &= \sum_y P(y)\mathbb{E}(x|y) = \left\{ \mathbb{E}(x|y) \text{ можно воспринимать как функцию от } y \right\} = \mathbb{E}_y[\mathbb{E}_x(x|y)]\end{aligned}$$

Покажем, как искать дисперсию случайной величины, распределение которой зависит от другой случайной величины:

$$\begin{aligned}\mathbb{D}(x) &= \mathbb{E}(x^2|y) - \mathbb{E}^2(x|y) \\ \mathbb{E}(x^2|y) &= \mathbb{D}(x|y) + \mathbb{E}^2(x|y) \quad | \quad \mathbb{E}(\ast) \\ \mathbb{E}[\mathbb{E}(x^2|y)] &= \mathbb{E}[\mathbb{D}(x|y)] + \mathbb{E}[\mathbb{E}^2(x|y)] \\ \mathbb{E}[\mathbb{E}(x^2|y)] &= \mathbb{E}x^2 \quad \text{по доказанному выше}\end{aligned}$$

Вычтем  $\mathbb{E}^2 x$ :

$$\begin{aligned}\mathbb{E}x^2 - \mathbb{E}^2 x &= \mathbb{D}x = \mathbb{E}[\mathbb{D}(x|y)] + \mathbb{E}[\mathbb{E}^2(x|y)] - \mathbb{E}^2 x \\ \mathbb{E}[\mathbb{E}^2(x|y)] - \mathbb{E}^2 x &= \mathbb{E}[\mathbb{E}^2(x|y)] - 2\mathbb{E}x\mathbb{E}x + \mathbb{E}^2 x = \mathbb{E}[\mathbb{E}^2(x|y) - 2\mathbb{E}(x|y)\mathbb{E}x + \mathbb{E}^2 x] = \\ &= \mathbb{E}[\mathbb{E}(x|y) - \mathbb{E}x]^2 = \sum_k [\mathbb{E}(x|y) - \mathbb{E}x]^2 P(y=k) = \\ &= \sum_k [\mathbb{E}(x|y) - \mathbb{E}[\mathbb{E}(x|y)]]^2 P(y=k) = \mathbb{D}[\mathbb{E}(x|y)]\end{aligned}$$

Тогда:

$$\begin{aligned}\mathbb{D}x &= \mathbb{E}[\mathbb{D}(x|y)] + \mathbb{D}[\mathbb{E}(x|y)] \\ \mathbb{D}x &= \mathbb{E}_y[\mathbb{D}_x(x|y)] + \mathbb{D}_x[\mathbb{E}_y(x|y)]\end{aligned}$$

Итак, было показано, что:

$$\begin{aligned}\mathbb{E}x &= \mathbb{E}_y[\mathbb{E}_x(x|y)] \\ \mathbb{D}x &= \mathbb{E}_y[\mathbb{D}_x(x|y)] + \mathbb{D}_x[\mathbb{E}_y(x|y)]\end{aligned} \tag{16}$$

Основываясь на этих правилах, можно посчитать моменты априорных распределений величин  $a, b, c_n, d_n$  аналитически.

## 4.2 Аналитический вывод моментов

Замечание: во избежание путаницы в данном разделе будут наименованы числами только конечные формулы моментов. Остальные, участвующие в выводе, не помечены.

### 4.2.1 Математические ожидания

Введём обозначения:

$$\begin{aligned}N_a &:= a_{max} - a_{min} + 1 \\ N_b &:= b_{max} - b_{min} + 1\end{aligned}$$

$$\begin{aligned}
\mathbb{E}(a) &= \sum_{k=a_{min}}^{a_{max}} k P(a=k) = \left\{ P(a=k) = \frac{1}{N_a} \right\} = \frac{1}{N_a} \sum_{k=a_{min}}^{a_{max}} k = \\
&= \frac{1}{N_a} \left( \sum_{k=1}^{a_{max}} k - \sum_{k=1}^{a_{min}-1} k \right) = \frac{1}{N_a} \left( \frac{a_{max}(a_{max}+1)}{2} - \frac{a_{min}(a_{min}-1)}{2} \right) = \\
&= \frac{1}{2N_a} \left( a_{max}(a_{max}-a_{min}+1) + a_{max}a_{min} + a_{min}(a_{max}-a_{min}+1) - a_{max}a_{min} \right) = \\
&= \left\{ N_a = a_{max} - a_{min} + 1 \right\} = \frac{1}{2} (a_{min} + a_{max})
\end{aligned} \tag{17}$$

$$\text{Аналогично, } \mathbb{E}(b) = \frac{1}{2} (a_{min} + a_{max}) \tag{18}$$

Выведем математическое ожидание для  $c_n$ :  
Для (3) модели:

$$\begin{aligned}
\mathbb{E}(c_n) &= \mathbb{E}_{a,b} [\mathbb{E}(c_n|a,b)] = \\
&= \left\{ \text{Пусть } c_n = c_n^1 + c_n^2, \quad c_n^1 \sim Bin(a, p_1), \quad c_n^2 \sim Bin(b, p_2), \quad c_n^1, c_n^2 \text{-независимы} \right\} = \\
&= \left\{ \xi \sim Bin(n, p) \implies \mathbb{E}(\xi) = np \right\} = \\
&= \mathbb{E}_{a,b}(ap_1 + bp_2) = p_1 \mathbb{E}(a) + p_2 \mathbb{E}(b) = \frac{p_1}{2} (a_{min} + a_{max}) + \frac{p_2}{2} (b_{min} + b_{max})
\end{aligned} \tag{19}$$

Для (4) модели:

$$\begin{aligned}
\mathbb{E}(c_n) &= \mathbb{E}_{a,b} [\mathbb{E}(c_n|a,b)] = \left\{ \xi \sim Poiss(\lambda) \implies \mathbb{E}(x) = \lambda \right\} = \\
&= \mathbb{E}_{a,b}(ap_1 + bp_2) = p_1 \mathbb{E}(a) + p_2 \mathbb{E}(b) = \frac{p_1}{2} (a_{min} + a_{max}) + \frac{p_2}{2} (b_{min} + b_{max})
\end{aligned} \tag{20}$$

Математическое ожидание для  $d_n$ :

$$\begin{aligned}
\mathbb{E}(d_n) &= \left\{ \xi \sim Bin(n, p) \implies \mathbb{E}(\xi) = np \right\} = \mathbb{E}_{c_n}(c_n + c_n p_3) = \\
&= (1 + p_3) \mathbb{E}c_n = (1 + p_3) \left( \frac{p_1}{2} (a_{min} + a_{max}) + \frac{p_2}{2} (b_{min} + b_{max}) \right)
\end{aligned} \tag{21}$$

#### 4.2.2 Дисперсии

$$\begin{aligned}
\mathbb{E}(a^2) &= \sum_{k=a_{min}}^{a_{max}} k^2 P(a=k) = \left\{ P(a=k) = \frac{1}{N_a} \right\} = \frac{1}{N_a} \sum_{k=a_{min}}^{a_{max}} k^2 = \\
&= \frac{1}{N_a} \left( \sum_{k=1}^{a_{max}} k^2 - \sum_{k=1}^{a_{min}-1} k^2 \right) = \frac{1}{N_a} \left( \frac{a_{max}(a_{max}+1)(2a_{max}+1)}{6} - \frac{(a_{min}-1)a_{min}(2a_{min}-1)}{6} \right) = \\
&= \frac{1}{6} \left( a_{max}(2a_{max}+1) + \frac{a_{max}a_{min}(2a_{max}+1)}{N_a} + a_{min}(2a_{min}-1) - \frac{a_{max}a_{min}(2a_{min}-1)}{N_a} \right) = \\
&= \left\{ N_a = a_{max} - a_{min} + 1 \right\} = \frac{1}{6} (2a_{max}^2 + 2a_{min}^2 + 2a_{max}a_{min} + a_{max} - a_{min})
\end{aligned}$$

Аналогично,  $\mathbb{E}(b^2) = \frac{1}{6} (2b_{max}^2 + 2b_{min}^2 + 2b_{max}b_{min} + b_{max} - b_{min})$

$$\begin{aligned}
\mathbb{D}(a) &= \mathbb{E}(a^2) - (\mathbb{E}(a))^2 = \frac{1}{6}(2a_{max}^2 + 2a_{min}^2 + 2a_{max}a_{min} + a_{max} - a_{min}) - \frac{1}{4}(a_{min} + a_{max})^2 = \\
&= \frac{1}{12}(a_{max}^2 + a_{min}^2) - \frac{1}{6}a_{max}a_{min} + \frac{1}{6}(a_{max} - a_{min}) = \\
&= \frac{1}{12}(a_{max}^2 + a_{min}^2 - 2a_{max}a_{min} + 2a_{max} - 2a_{min}) = \frac{1}{12}((a_{max} - a_{min} + 1)^2 - 1) = \frac{1}{12}(N_a^2 - 1)
\end{aligned} \tag{22}$$

$$\text{Аналогично, } \mathbb{D}(b) = \frac{1}{12}(N_b^2 - 1) \tag{23}$$

Выведем  $\mathbb{D}c_n$ , используя правила (21):

Для (3) модели:

$$\begin{aligned}
\mathbb{D}(c_n|a, b) &= \{\text{Пусть } c_n = c_n^1 + c_n^2, \quad c_n^1 \sim Bin(a, p_1), \quad c_n^2 \sim Bin(b, p_2), \quad c_n^1, c_n^2\text{-независимы}\} = \\
&= \{\xi \sim Bin(n, p) \implies \mathbb{D}(\xi) = np(1-p)\} = ap_1(1-p_1) + bp_2(1-p_2)
\end{aligned}$$

$$\mathbb{E}(c_n|a, b) = \{\text{Аналогичной заменой}\} = ap_1 + bp_2$$

$\mathbb{E}[\mathbb{D}(c_n|a, b)] = \{\text{Один раз подпишем индексы у моментов, чтобы было ясно,}$   
по каким случайным величинам они берутся,

в остальных случаях это соответствие подразумевается неявно

$$\begin{aligned}
&= \mathbb{E}_{a,b}[\mathbb{D}_{c_n}(c_n|a, b)] = \mathbb{E}[ap_1(1-p_1) + bp_2(1-p_2)] = p_1(1-p_1)\mathbb{E}a + p_2(1-p_2)\mathbb{E}b \\
&\mathbb{D}[\mathbb{E}(c_n|a, b)] = \mathbb{D}_{a,b}[\mathbb{E}_{c_n}(c_n|a, b)] = \mathbb{D}[p_1a + p_2b] = p_1^2\mathbb{D}a + p_2^2\mathbb{D}b
\end{aligned}$$

Наконец,

$$\begin{aligned}
\mathbb{D}c_n &= \mathbb{E}[\mathbb{D}(c_n|a, b)] + \mathbb{D}[\mathbb{E}(c_n|a, b)] = \\
&= p_1(1-p_1)\mathbb{E}a + p_2(1-p_2)\mathbb{E}b + p_1^2\mathbb{D}a + p_2^2\mathbb{D}b
\end{aligned} \tag{24}$$

Для (4) модели:

$$\mathbb{D}(c_n|a, b) = \{\xi \sim Poiss(\lambda) \implies \mathbb{D}(\xi) = \lambda\} = ap_1 + bp_2$$

$$\mathbb{E}(c_n|a, b) = ap_1 + bp_2$$

$$\mathbb{E}[\mathbb{D}(c_n|a, b)] = \mathbb{E}_{a,b}[\mathbb{D}_{c_n}(c_n|a, b)] = \mathbb{E}[ap_1 + bp_2] = p_1\mathbb{E}a + p_2\mathbb{E}b$$

$$\mathbb{D}[\mathbb{E}(c_n|a, b)] = \mathbb{D}_{a,b}[\mathbb{E}_{c_n}(c_n|a, b)] = \mathbb{D}[p_1a + p_2b] = p_1^2\mathbb{D}a + p_2^2\mathbb{D}b$$

Наконец,

$$\begin{aligned}
\mathbb{D}c_n &= \mathbb{E}[\mathbb{D}(c_n|a, b)] + \mathbb{D}[\mathbb{E}(c_n|a, b)] = \\
&= p_1\mathbb{E}a + p_2\mathbb{E}b + p_1^2\mathbb{D}a + p_2^2\mathbb{D}b
\end{aligned} \tag{25}$$

Выведем  $\mathbb{D}d_n$ :

$$\mathbb{D}(d_n|c_n) == \{d_n|c_n \sim Bin(c_n, p_3) + c_n,$$

$$\begin{aligned}
\text{при фиксированном } c_n \text{ дисперсия второго слагаемого равна 0, как и } Cov(Bin(c_n, p_3), c_n) \} = \\
= c_n p_3(1-p_3) + 0 + 0 = c_n p_3(1-p_3)
\end{aligned}$$

$$\begin{aligned}\mathbb{E}(d_n|c_n) &= c_n p_3 + c_n = (1 + p_3)c_n \\ \mathbb{E}[\mathbb{D}(d_n|c_n)] &= \mathbb{E}_{c_n}[\mathbb{D}_{d_n}(d_n|c_n)] = \mathbb{E}[c_n p_3(1 - p_3)] = p_3(1 - p_3)\mathbb{E}c_n \\ \mathbb{D}[\mathbb{E}(c_n|a, b)] &= \mathbb{D}_{c_n}[\mathbb{E}_{d_n}(d_n|c_n)] = \mathbb{D}[(1 + p_3)c_n] = (1 + p_3)^2\mathbb{D}c_n\end{aligned}$$

Наконец,

$$\begin{aligned}\mathbb{D}c_n &= \mathbb{E}[\mathbb{D}(d_n|c_n)] + \mathbb{D}[\mathbb{E}(d_n|c_n)] = \\ &= p_3(1 - p_3)\mathbb{E}c_n + (1 + p_3)^2\mathbb{D}c_n\end{aligned}\tag{26}$$

Итак, формулы (17)-(21) задают аналитическую запись математических ожиданий априорных распределений случайных величин  $a, b, c_n, d_n$  для 3 и 4 моделей, а формулы (22)-(26) - дисперсии этих распределений.

При подсчёте моментов в качестве значений параметров модели использовались следующие величины:

$$a_{min} = 75, \quad a_{max} = 90, \quad b_{min} = 500, \quad b_{max} = 600, \tag{27}$$

$$p_1 = 0.1, \quad p_2 = 0.01, \quad p_3 = 0.3 \tag{28}$$

В ходе исследования были получены следующие значения данных моментов:

Величина	$\mathbb{E}$ 3 модели	$\mathbb{E}$ 4 модели	$\mathbb{D}$ 3 модели	$\mathbb{D}$ 4 модели
$a$	82.5	82.5	21.25	21.25
$b$	550	550	850	850
$c_n$	13.75	13.75	13.1675	14.0475
$d_n$	17.875	17.875	25.140575	26.627775

Табл. 1: Моменты априорных распределений

Результаты подсчёта моментов по их распределению (через распределения) и по аналитическим формулам выше отличаются не более, чем на  $1.2 \times 10^{-10}$ .

Уже сейчас можно заметить, что распределения 3 и 4 моделей выдают примерно одинаковые результаты. Визуально графики их распределений не отличаются. Это объясняется выполнением условий перехода от биномиальных распределений к распределениям Пуассона: большое число испытаний и малая вероятность успеха. И всё же видно, что в 4 модели априорные распределения на  $c_n, d_n$  имеют больший разброс, что в плане точности выдвигает на выигрышную позицию 3 модель, хоть и немного - она выдаёт более "уверенное априорное распределение" на величины в модели.

Рассмотрим общий вид априорных распределений  $c_n$  и  $d_n$ :

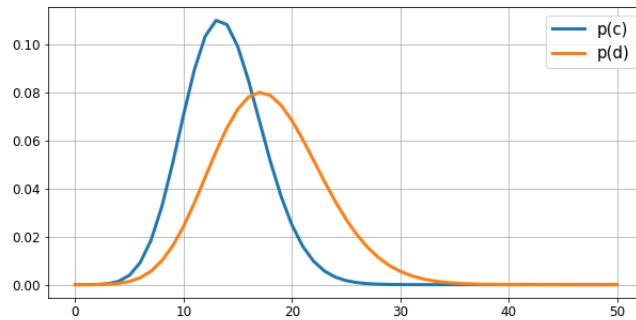


Рис. 1: Априорные распределения

Как видно, мода распределения  $d_n$  больше, чем у  $c_n$ . Разброс тоже больше, что естественным образом вытекает из модельного задания их распределений ( $c_n$  дополнительно вносит свою случайность в  $d_n$ ). Среднее  $c_n$  находится около 14 человек, а  $d_n$  - около 18, что также согласуется с параметрами модели.

## 5 Характер уточнения апостериорного распределения

Для анализа поведения апостериорных распределений  $p(b|d_1, \dots, d_n)$  и  $p(b|a, d_1, \dots, d_n)$  был реализован генератор, позволяющий семплировать последовательно параметры  $c_n, d_n$  при фиксированных параметрах  $a, b$  согласно их распределению внутри каждой из моделей (3) и (4). В его программной реализации используются генераторы из библиотеки NumPy, что позволяет быстро и эффективно вносить в вероятностную модель косвенную информацию.

В ходе экспериментов были получены следующие последовательности распределений. Для улучшения читаемости эти результаты сгруппированы парами и разделены друг от друга в работе чертой.

---

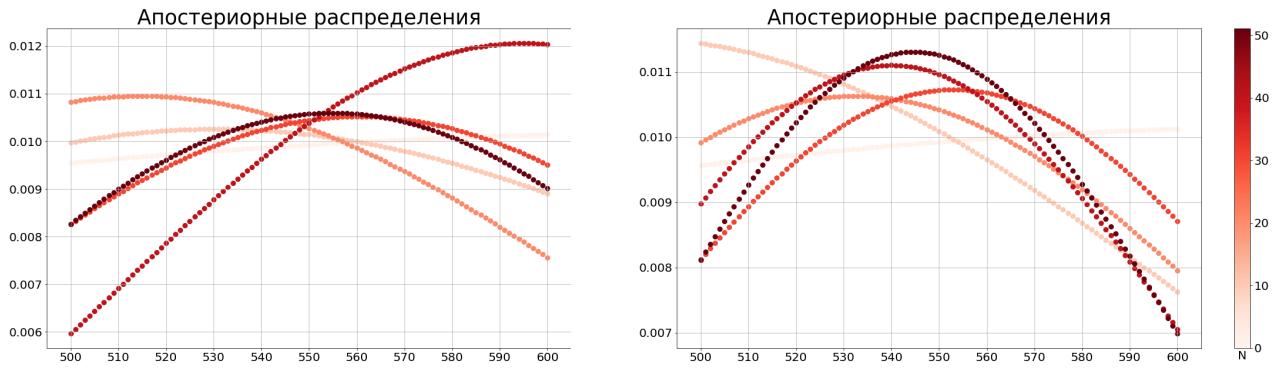


Рис. 2: Динамика  $p(b|d_1, \dots, d_N)$  и  $p(b|a, d_1, \dots, d_N)$  при генерируемых  $d_n(a, b)$ ,  $a = [\mathbb{E}a]$ ,  $b = [\mathbb{E}b]$

$N$	$\mathbb{E}(b d_1, \dots, d_N)$	$\mathbb{D}(b d_1, \dots, d_N)$
1	550.51	847.78
10	549.05	832.65
20	547.05	815.94
30	551.13	811.97
40	555.53	786.78
50	550.67	803.88

$N$	$\mathbb{E}(b a, d_1, \dots, d_N)$	$\mathbb{D}(b a, d_1, \dots, d_N)$
1	550.48	847.9
10	546.62	825.52
20	548.18	811.8
30	550.52	795.17
40	548.08	776.19
50	548.8	760.07

На рисунке 2 представлены результаты изменения распределений  $p(b|d_1, \dots, d_N)$  для 3 модели (слева) и  $p(b|a, d_1, \dots, d_N)$  для 4 модели (справа) при генерируемых значениях  $d_n$ . При указанных в (27), (28) параметрах моделей допустим переход между биномиальными и пуассоновскими случайными величинами в записи  $c_n$ . Поэтому модели в данном случае буду работать практически одинаково. Учитывая это, ниже все сравнения будут производиться внутри третьей модели. Что касается полученных распределений в данном примере, то можно легко видеть, что апостериорные распределения начинаются около равномерных (бледная практически горизонтальная линия), но с приходом новой информации более явно образуют "купол" (тёмная линия). Таким образом, можно заключить, что, во-первых, обе модели способны корректно обрабатывать поток косвенной информации, уточняя апостериорные распределения, а, во-вторых, что сами распределения  $p(b|d_1, \dots, d_N)$  и  $p(b|a, d_1, \dots, d_N)$  - при увеличении наблюдений могут переходить к более чётким результатам. Это проявляется в постепенном убывании дисперсий этих распределений.

Далее, сравнивая в данном случае апостериорные распределения между собой, видно, что  $p(b|a, d_1, \dots, d_N)$  может, как по форме, так и по значению её разброса демонстрировать более "уверенные" распределения (более узкое распределение), что логично, так как в системе имеется дополнительная информация о величине  $a$ .

---

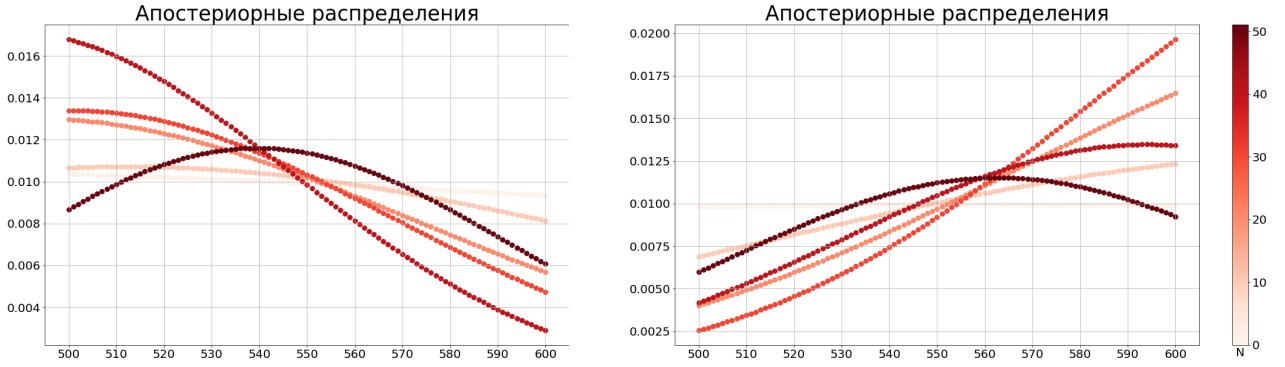


Рис. 3: Динамика  $p(b|a, d_1, \dots, d_N)$  при генерируемых  $d_n(a, b)$ ,  $a = a_{min} = 75$  (left) и  $a = a_{max} = 90$  (right),  $b = [\mathbb{E}b]$

$N$	$\mathbb{E}(b a, d_1, \dots, d_N)$	$D(b a, d_1, \dots, d_N)$	$N$	$\mathbb{E}(b a, d_1, \dots, d_N)$	$D(b a, d_1, \dots, d_N)$
1	549.12	847.69	1	550.07	848.11
10	547.74	826.85	10	554.78	816.77
20	543.35	785.2	20	561.07	740.65
30	541.87	754.02	30	565.19	664.5
40	536.93	680.27	40	558.81	732.74
50	547.25	746.85	50	553.23	753.39

В следующих экспериментах исследовалось поведение  $p(b|a, d_1, \dots, d_N)$  при генерируемых значениях  $d_n$  с крайними возможными фиксированными значениями  $a$ . В данном случае апостериорные распределения аналогичным образом уточняются. Видно, что в данном случае дисперсии не убывают монотонно, это связано с тем, что новая пришедшая в систему информация о  $d_n$  могла быть не согласована с предыдущими наблюдениями и потребовала внесения корректировки в распределение, что однако не помешало прийти к распределению, имеющему ещё меньший разброс по сравнению с предыдущим экспериментом. От "удачливости" генерации  $d_n$  (то есть от согласованности получаемых данных между собой) зависит то, насколько итоговое распределение будет казаться точным. На самом деле, апостериорные распределения содержат всю важную информацию о наблюдениях, и, как бы неточно они не выглядели, они будут одним из лучших аккумуляторов данных, что можно получить. При росте наблюдаемой выборки распределения будут иметь более точную форму.

Видно, что при малых значениях  $a$  распределение сначала склоняется к меньшим значениям  $b$ , и наоборот - при больших  $a$  сначала склоняется к большим значениям  $b$ , но потом в обоих случаях выравнивается (это можно наблюдать по математическим ожиданиям). Такой эффект связан с тем, что генерируемые  $d_n$  при малых  $a$  сами имеют небольшие значения. Из-за этого первое впечатление таково, что и  $b$  будет мало. Так и происходит. Однако впоследствии в системе начинает играть роль известность  $a$  (мы рассматриваем апостериорное распределение  $p(b|a, d_1, \dots, d_N)$ ). Поэтому модель способна учесть это и понять, что малость  $d_n$  связана именно с малостью  $a$ , а не  $b$ , и начинает выравнивать распределение по  $b$  к её середине. Со второй цепочкой распределений на рисунке 3 всё аналогично, только в этом случае  $a$  - завышено.

Отметим, что если бы информация об  $a$  не имелась в системе, то выравнивание бы не происходило. Причём для допустимых фиксируемых  $a$  такая модель бы была настроена на уменьшение значения  $b$ .

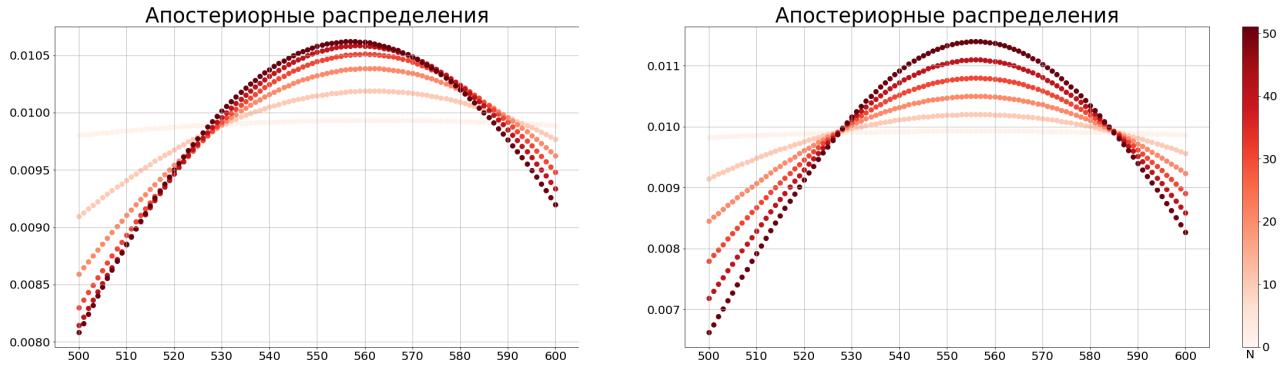


Рис. 4: Динамика  $p(b|d_1, \dots, d_N)$  (left) и при  $p(b|a, d_1, \dots, d_N)$  (right),  $d_n = [\mathbb{E}d_n] = 18$ ,  $a = [\mathbb{E}a]$

$N$	$\mathbb{E}(b d_1, \dots, d_N)$	$\mathbb{D}(b d_1, \dots, d_N)$	$N$	$\mathbb{E}(b a, d_1, \dots, d_N)$	$\mathbb{D}(b a, d_1, \dots, d_N)$
1	550.07	848.04	1	550.04	848.03
10	550.59	832.84	10	550.36	830.4
20	550.92	820.6	20	550.69	811.0
30	551.06	812.24	30	551.02	791.86
40	551.07	806.77	40	551.32	773.0
50	550.99	803.38	50	551.61	754.45

В данном эксперименте (рис.4) значения  $d_n$  фиксировались равными своим математическим ожиданиям и производилось сравнений апостериорных распределений при неизвестном и известном значении  $a$  (которое фиксировалось равным своему априорному математическому ожиданию).

Как видно, в данном случае дисперсии монотонно убывают, так как каждое входящее в систему  $d_n$  лишь подтверждает уже сформированное распределение (так как все  $d_n$  - одинаковы). Соответственно модели не нужно перестраиваться под новые данные с учётом старых значений - она всего лишь делает то же предсказание более уверенно. Можно заметить, что математические ожидания апостериорных распределений практически не меняются, а горбы распределений становятся более крутыми, что подтверждает написанное выше.

Здесь же прослеживается аналогичное прошлым пунктам эксперимента правило, по которому при известном системе значении  $a$  предсказанные ею распределения обладают меньшим разбросом, что довольно естественно (если сравнить представленные 2 распределения по дисперсиям, то при известной величине  $a$  дисперсия апостериорного распределения меньше).

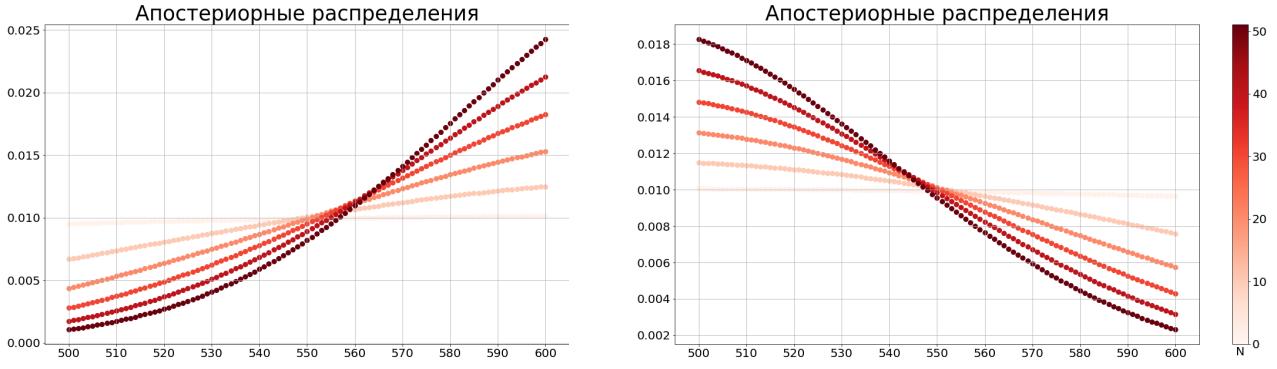


Рис. 5: Динамика  $p(b|a, d_1, \dots, d_N)$  при  $d_n = [\mathbb{E}d_n] = 18$ ,  $a = a_{min} = 75$  (left) и  $a = a_{max} = 90$  (right)

$N$	$\mathbb{E}(b a, d_1, \dots, d_N)$	$D(b a, d_1, \dots, d_N)$
1	550.53	847.62
10	555.13	812.66
20	559.83	751.84
30	564.0	678.43
40	567.62	601.86
50	570.72	528.67

$N$	$\mathbb{E}(b a, d_1, \dots, d_N)$	$D(b a, d_1, \dots, d_N)$
1	549.65	848.14
10	546.56	825.28
20	543.31	788.98
30	540.29	744.56
40	537.53	695.4
50	535.01	644.42

В эксперименте, результаты которого приведены на рисунке 5, фиксировались значения  $d_n$  и варьировались значения  $a$  для анализа апостериорных распределений  $p(b|a, d_1, \dots, d_N)$ . Были получены следующие результаты:

Аналогично прошлому пункту дисперсии распределений монотонно убывают (все  $d_n$  - одинаковы). Кроме того, математические ожидания, как и сами распределения, монотонно отклоняются от "середины" распределения. Например, если зафиксировать наименьшее  $a = a_{min}$ , то модель, зная, что  $a$  мало, а  $d_n$  - средне ожидаемо, будет завышать значения  $b$ . График распределения всё больше склоняется к наибольшему значению  $b$ , с математическим ожиданием - то же самое. Аналогично и для завышенного фиксированного значения  $a$  - модель пытается сильнее занизить значения  $b$ . Поэтому полученные графики выглядят практически симметрично - в них представлены противоположные случаи.

Подытожим, выделив основные выводы этого раздела:

- При поступлении новой информации модели способны адаптировать апостериорные распределения под них, как правило результат становится более и более точным (рис. 2);
- Распределение  $p(b|a, d_1, \dots, d_N)$  выходит более точным (имеет меньший разброс), чем  $p(b|d_1, \dots, d_N)$ , так как обладает более объёмной информацией (рис. 2 и 4);
- При поступлении неоднородной информации моделям требуется подстроится под новые данные, используя уже накопленную информацию, что может повысить её разброс (рис. 3). Вместе с этим, это ни в коем случае не значит, что модели работают некорректно, так как они как раз выражают наиболее компромиссный вариант учёта всех доступных данных, что является очень сильной стороной метода Байесовского вывода;
- При поступлении согласованной информации моделям достаточно сужать распределение, делая его более точным (рис. 4), что они успешно реализуют;

- Модели способны самостоятельно настраивать распределения на любые допустимые входящие данные, и это будет согласовано с реальными условиями задачи (рис. 5);
- 3 и 4 модели в данном случае показывают практически одинаковые результаты, так как при заданных параметрах моделей допустим переход между ними с сохранением точности;

Таким образом, Байесовский подход позволяет подстраиваться под поток входящих данных и выражать самые точные распределения, которые можно получать из этой косвенной информации. В этом заключается одно из его полезных мощных свойств.

## 6 Сравнение программных реализаций моделей

В данном пункте рассмотрим вопрос эффективности реализации каждой из моделей через замеры времени функций подсчёта распределений. В ходе экспериментов были запущены функции вычисления априорных распределений  $p(c_n)$  и  $p(d_n)$ , а также апостериорных распределений  $p(b|d_1, \dots, d_N)$  и  $p(b|a, d_1, \dots, d_N)$  отдельно для 3 и 4 моделей. Для апостериорных распределений отдельно варьировались размерности её параметров:  $N$  - число величин  $d_n$ ,  $k_a$  - вторая размерность  $d_n$  - значение величины  $a$  для генерации наборов  $d_n$ . Ниже приведены результаты череды запусков этих функций.

Распределение	3 модель	4 модель
$p(c_n)$	$141 \text{ ms} \pm 1.89 \text{ ms}$	$65.4 \text{ ms} \pm 1.38 \text{ ms}$
$p(d_n)$	$203 \text{ ms} \pm 1.83 \text{ ms}$	$130 \text{ ms} \pm 1.43 \text{ ms}$

Табл. 2: Время вычисления априорных распределений

Распределение	3 модель	4 модель
$p(b d_1, \dots, d_N)$	$298 \text{ ms} \pm 4.1 \text{ ms}$	$216 \text{ ms} \pm 10.9 \text{ ms}$
$p(b a, d_1, \dots, d_N)$	$276 \text{ ms} \pm 6.45 \text{ ms}$	$186 \text{ ms} \pm 1.78 \text{ ms}$

Табл. 3: Время вычисления апостериорных распределений,  $N = 50, k_a = 1$

Распределение	3 модель	4 модель
$p(b d_1, \dots, d_N)$	$641 \text{ ms} \pm 21.2 \text{ ms}$	$541 \text{ ms} \pm 4.61 \text{ ms}$
$p(b a, d_1, \dots, d_N)$	$834 \text{ ms} \pm 13 \text{ ms}$	$743 \text{ ms} \pm 8.12 \text{ ms}$

Табл. 4: Время вычисления апостериорных распределений,  $N = 50, k_a = 100$

Распределение	3 модель	4 модель
$p(b d_1, \dots, d_N)$	$1.88 \text{ s} \pm 14.5 \text{ ms}$	$1.82 \text{ s} \pm 47.8 \text{ ms}$
$p(b a, d_1, \dots, d_N)$	$2.79 \text{ s} \pm 13.9 \text{ ms}$	$2.78 \text{ s} \pm 73.6 \text{ ms}$

Табл. 5: Время вычисления апостериорных распределений,  $N = 250, k_a = 100$

Из данных результатов можно сделать следующие выводы:

- Четвёртая модель работает быстрее третьей. Собственно, в этом и заключался смысл аппроксимации суммы биномиальных распределений одним пуассоновским для величины  $c_n$  - его как аналитически, так и программно считать намного проще.

- На небольших значениях  $N$  различие в скорости работы программ этих моделей проявляется более явно, с увеличением этого параметра они по эффективности выходят на один уровень. Это связано с тем, что рост  $N$  обеспечивает более высокую точность аппроксимации, и соответственно используемые модели становятся более схожи друг с другом, как в плане результатов вычислений распределений, так и во времени работы их реализаций.
- Как видно, реализация вычисления апостериорного распределения  $p(b|a, d_1, \dots, d_N)$  "на лету" (формулы (13), (14)) при небольших значениях  $N$  и  $k_a$  выигрывает в эффективности у реализации  $p(b|d_1, \dots, d_N)$  "подсчёт за один раз" (формула (12)). Однако с увеличением значений этих параметров отмеченная эффективность падает, и подсчёт апостериорного распределения "за один раз", как в случае  $p(b|d_1, \dots, d_N)$ , более выгоден. Вместе с тем стоит сделать важное замечание, что подсчёт распределений "на лету" в конце своей работы способен выдать последовательность распределений - каждое получено при добавлении очередного  $d_n$ , так как такой метод подразумевает итеративное преобразование вероятностного вектора для величины  $b$ . То есть результат данной функции на самом деле более объёмный, чем у "подсчёта за один раз". Сразу здесь и отметим, что оба способа допускают обобщение на случай обработки последовательного набора информации, когда в вероятностную модель входят значения величин  $d_n$  группами. Данное свойство позволяет дополнительно повысить эффективность реализации подсчёта апостериорных распределений, как по требуемой памяти, так и по времени вычисления (если требуется выводить динамику распределения с приходом новой косвенной информации).

Таким образом, 4 модель работает, как правило, быстрее третьей. При увеличении размерностей входящих в вычисления параметров эта разница сокращается. Также обработка информации "на лету" более эффективна в случае, когда необходимо выводить динамику изменения распределений при потоке косвенной информации.

## 7 Сравнение вероятностных моделей

Теперь проведём принципиальное сравнение моделей 3 и 4. Оно содержит в себе два основных фактора: вычислительная сложность реализаций которая проявляется в разнице времени работы программ, а также различие самих распределений, производимых данными моделями при одинаковых параметрах.

Напомним, что концептуально 4 модель нам требуется, как облегчённая "аппроксимация" третьей модели. Это значит, что она более эффективна по времени работы, однако точность её приближения полагается на соблюдение условий перехода биномиального распределения в пуассоновское - большое число испытаний, и малая вероятность в биномиальном распределении. Соответственно, чтобы пронаблюдать, как модели работают по-разному, достаточно нарушить эти условия приближения.

Ниже представлены результаты работы функций вычисления распределений для 3 и 4 моделей при определённых параметрах (2 эксперимента разделены по колонкам соответственно):

$$a_{min} = 75$$

$$a_{max} = 90$$

$$b_{min} = 500$$

$$b_{max} = 600$$

$$p_1 = 0.9$$

$$p_2 = 0.9$$

$$p_3 = 0.3$$

$$a_{min} = 15$$

$$a_{max} = 30$$

$$b_{min} = 100$$

$$b_{max} = 200$$

$$p_1 = 0.5$$

$$p_2 = 0.1$$

$$p_3 = 0.3$$

$$N = 10$$

$$d_i = 30$$

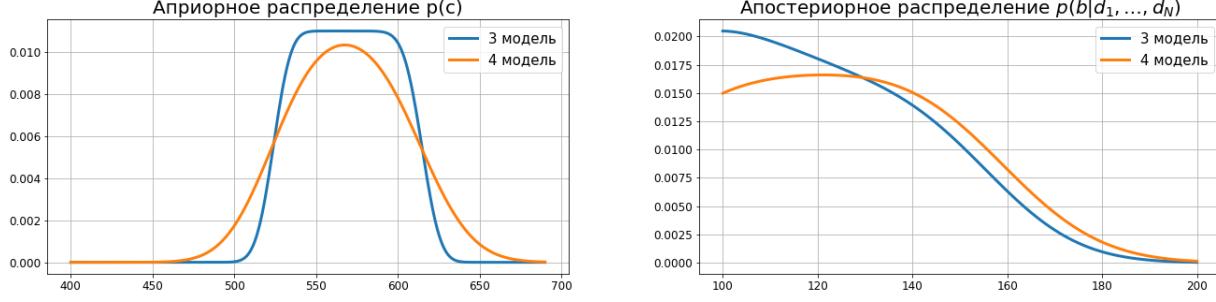


Рис. 6: Пример отличия работы моделей

	3 модель	4 модель		3 модель	4 модель
$\mathbb{E}c_n$	569.250	569.168	$\mathbb{E}(b d_1, \dots, d_N)$	128.260	132.232
$\mathbb{D}c_n$	762.638	1311.114	$\mathbb{D}(b d_1, \dots, d_N)$	377.882	424.928

Табл. 6: Моменты полученных распределений

Во-первых, сразу видно, что вид распределений от разных моделей явно отличается. Поэтому в данных случаях приближать третью модель упрощённой четвёртой не следует. Во-вторых, моменты этих распределений также отличаются. Единственное, что практически совпадает - это  $\mathbb{E}c_n$ , что и должно быть всегда в теории. Однако 4 модель в данных примерах выдаёт превосходящую по значению дисперсию, нежели 3 модель, что не играет ей в плюс. Данные различия проявляются из-за изменений, которые были произведены над параметрами, входящими в запись  $p(c|a, b)$  - сумме биномиальных величин: в этих величинах число испытаний было снижено, а вероятность успеха - завышена. Из-за этого нарушаются условия перехода биномиального распределения в пуассоновское, из-за чего и получаются демонстрируемые различия.

Что касается разницы по времени выполнения функций вычисления распределений, то результаты уже были продемонстрированы в таблицах (2)-(4), и дублировать их тут не требуется. Повторим, что 4 модель является более простой для вычислений, а потому и её программная реализация работает быстрее третьей.

Резюмируя результаты данного раздела, отметим, что четвёртая модель является упрощённой версией третьей, и поэтому позволяет производить подсчёт распределений и их моментов быстрее. Однако для её корректного использования необходимо выполнение условий допустимости перехода биномиальных распределений, используемых в величинах  $c_n$ , в пуассоновские, что достигается при большом числе испытаний и малой вероятности успеха. В ином случае, как было показано, генерируются сильно различающиеся распределения, что недопустимо при решении данной задачи.

## **8 Выводы**

В данной работе был изучен метод применения Байесовских рассуждений на примере задачи анализа числа посещающих лекции студентов.

Было рассмотрено две вероятностные модели: точная - 3, и приближённая - 4, более эффективная в плане вычислительных затрат. Для них производился вывод формул распределений, а также их статистических моментов, использующихся далее. Были изучены условия, при которых переход между моделями с определённой точностью можно считать корректным. Производилось сравнение этих моделей, которое показало, что программная реализация 4 модели производит вычисления быстрее, продемонстрировались случаи, когда замена 3 модели на 4 недопустима из-за потери точности приближения.

Большое значение уделялось исследованию способности моделей обрабатывать поступающую информацию на примере вычисления апостериорных распределений  $p(b|d_1, \dots, d_N)$  и  $p(b|a, d_1, \dots, d_N)$ . Было представлено два способа их подсчёта - "за один раз" и "на лету". Они также были сравнены на качественном уровне. По итогу этого блока исследований было продемонстрировано, что данные модели могут вычленять из поступающих косвенных данных всю полезную информацию, находя наиболее релевантное, компромиссное состояние, агрегируя всю доступную информацию.

Также хочется отметить, что несмотря на простоту реализации Байесовского вывода в данной задаче - по сути использование основных правил математической статистики по работе с вероятностями - получившийся по итогу инструмент представляет мощную вероятностную модель, предоставляющую доступ к оперированию больших массивов сложных по статистической структуре данных, что довольно интересно.

## **9 Список литературы:**

1. Лекции и семинары по курсу "Байесовские методы машинного обучения"

**Московский Государственный Университет  
им. М.В. Ломоносова**

Факультет Вычислительной математики и кибернетики  
Кафедра Математических Методов Прогнозирования

Самбурский Александр. 417 группа.

**Практическое задание 2  
ЕМ алгоритм для детектива  
Отчёт о проделанной работе**

Москва 2021

# 1 Введение

В направлении байесовских методов значительную роль занимают модели, основанные на использовании латентных переменных. Одним из наиболее знаменитых методов среди таких моделей является EM-алгоритм. Задав вероятностную модель для метода прогнозирования, и корректным образом определив ненаблюданную информацию в виде латентных переменных, можно свести сложные задачи машинного обучения к более простым, эффективным и довольно точным численным методам оптимизации. В некоторых случаях EM-алгоритм предоставляет возможность вычленить из доступных данных много дополнительной полезной информации, что является одним из важных преимуществ такого подхода.

В данном отчёте представлены результаты изучения работы EM-алгоритма на примере задачи распознавания группы зашумлённых изображений. Для этого была применена предлагаемая вероятностная модель, для неё предварительно был произведён теоретический вывод необходимых формул, которые впоследствии были реализованы программно. Проведённые эксперименты продемонстрировали различные полезные свойства EM-алгоритма, как в плане его эффективности, так и по качеству и объёму предоставляемой им информации об исходных данных.

## Содержание

<b>1 Введение</b>	<b>1</b>
<b>2 Постановка задачи</b>	<b>2</b>
2.1 Используемые объекты . . . . .	2
2.2 Вероятностная модель . . . . .	3
2.3 Применение EM-алгоритма. Общие формулы . . . . .	3
<b>3 Вывод EM-алгоритма</b>	<b>4</b>
3.1 $p(X_k d_k, F, B, s)$ и его логарифм . . . . .	4
3.2 E-шаг, $p(d_k X_k, F, B, s, A)$ . . . . .	5
3.3 M-шаг . . . . .	5
3.4 ELBO . . . . .	7
<b>4 Вывод hard EM-алгоритма</b>	<b>8</b>
4.1 $p(X_k d_k, F, B, s)$ и его логарифм . . . . .	8
4.2 E-шаг, $p(d_k X_k, F, B, s, A)$ . . . . .	8
4.3 M-шаг . . . . .	8
4.4 ELBO . . . . .	10
<b>5 Эксперименты</b>	<b>10</b>
5.1 Сгенерированные данные . . . . .	10
5.2 Влияние начального приближения на прогноз . . . . .	11
5.3 Влияние качества и объёма выборки на прогноз . . . . .	13
5.4 Сравнение EM- и hard EM-алгоритмов . . . . .	16
5.5 Запуск EM-алгоритма на исходных данных . . . . .	17
<b>6 Идеи по модификации алгоритма</b>	<b>19</b>
<b>7 Выводы</b>	<b>20</b>
<b>8 Список литературы:</b>	<b>20</b>

## 2 Постановка задачи

Рассматриваемой задачей является анализ группы зашумлённых чёрно-белых изображений. Все эти изображения имеют один и тот же фон, и в различных нефиксированных местах их перекрывает картинка меньшего размера, на которой изображено лицо человека. Данного человека предстоит распознать, используя ЕМ-алгоритм на доступных данных.

### 2.1 Используемые объекты

Перейдём к формализации задачи. Каждое изображение рассматривается как множество чёрно-белых пикселей, яркость которых задаётся тензорами соответствующего размера. В дальнейшем не будем учитывать разницу между изображением и его тензором. Выборка представляет из себя набор  $K$  тензоров (соответствующих своим изображениям)  $\{X_k\}_{k=1}^K = X$ , каждое размера  $H \times W$ . Эти изображения заполняются одинаковым фоном, задающимся тензором  $B$  такого же размера. Мaska с лицом имеет размер  $h \times w$ , и накладывается на все изображения в различных положениях, ей соответствует тензор  $F$ . Каждое изображение  $X_k$  получается наслаждением маски  $F$  в некоторую позицию на фоне  $B$ , после чего результат зашумляется. Для положения левого верхнего угла маски определим отдельную переменную для каждого изображения - получим набор пар координат  $\{(d_k^h, d_k^w)\}_{k=1}^K$ . Считаем, что шум распределён согласно нормальному распределению независимо для каждого пикселя. Итак, далее будем пользоваться следующими обозначениями:

- $X_k(i, j)$  - пиксель  $k$ -ого изображения;
- $B \in \mathbb{R}^{H \times W}$  — изображение чистого фона без лица преступника,  $B(i, j)$  — пиксель этого изображения;
- $F \in \mathbb{R}^{h \times w}$  — изображение лица преступника,  $F(i, j)$  — пиксель этого изображения;
- $d_k = (d_k^h, d_k^w)$  — координаты верхнего левого угла изображения лица на  $k$ -ом изображении ( $d_k^h$  — по вертикали,  $d_k^w$  — по горизонтали),  $d = (d_1, \dots, d_K)$  — набор координат для всех изображений выборки;
- $s$  - дисперсия для стандартного шума.

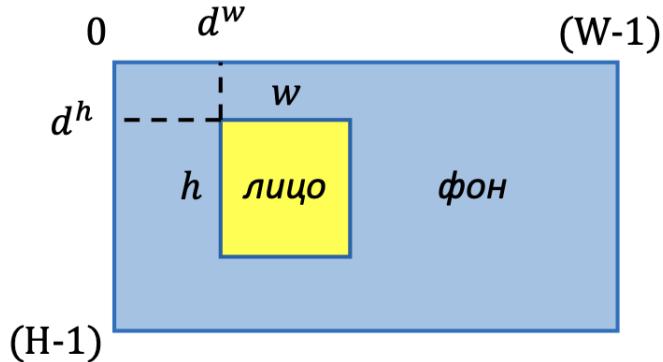


Рис. 1: Визуализация семантики переменных

По условиям данной задачи набор истинных значений переменных  $\{d_k\}$  - неизвестен - так как нам остаётся неизвестным конкретное расположение маски на входных изображениях. Однако при использовании ЕМ-алгоритма мы как раз их и будем считать латентными (ненаблюдаемыми) переменными, которые принимают свои значения согласно некоторому вероятностному распределению. По ходу работы алгоритма это распределение будет уточняться и приближаться к истинному - вырожденному. Эту латентность целесообразно использовать в данной задаче, так как благодаря ей можно ввести интуитивно понятную вероятностную модель и проводить оптимизацию по относительно простым формулам.

На самом деле, ненаблюдаемыми переменными здесь являются фактически почти все используемые переменные - помимо  $d_k$  ещё  $F$ ,  $B$  и  $s$ . Из наблюдаемых - только  $X_k$ .  $F$ ,  $B$ , и  $s$  будут рассматриваться, как параметры модели, и подлежат изменению на М-шаге. Это множество параметров обозначим за  $\theta$ :  $\theta = \{F, B, s^2\}$ . Такая расстановка латентных переменных и параметров оказывается наиболее удачной, так как позволяет моделировать достаточно простые распределения для ЕМ-алгоритма. Вместе с этим, в конце его работы у нас появятся оценки и на  $d_k$ , и на  $F$ ,  $B$  и  $s$ . Тут и проявится сильное свойство ЕМ-алгоритма вычленять много дополнительной информации из исходных данных.

## 2.2 Вероятностная модель

Сначала кратко опишем, по какой интуиции будет построена вероятностная модель. Если бы мы знали расположение маски для  $k$ -го изображения (известен тензор фона с наложенной поверх него маской), и тем самым незашумлённое изображение было бы определено однозначно, то тогда тензор  $X_k$  получался бы из него добавлением "попиксельного" шума с нулевым средним и некоторой всюду одинаковой дисперсией  $s$ . Таким образом, при зафиксированном  $d_k$  тензор  $X_k$  состоит из нормально распределённых элементов с дисперсией  $s$  и средним, соответствующим нужному пикслю маски или фона - в зависимости от расположения этого пикселя на  $X_k$ :

$$p(X_k | d_k, \theta) = \prod_{ij} \begin{cases} \mathcal{N}(X_k(i, j) | F(i - d_k^h, j - d_k^w), s^2), & \text{если } (i, j) \in faceArea(d_k), \\ \mathcal{N}(X_k(i, j) | B(i, j), s^2), & \text{иначе;} \end{cases} \quad (1)$$

Здесь  $\theta = \{B, F, s^2\}$  по введённому обозначению,

$faceArea(d_k) = \{(i, j) | d_k^h \leq i \leq d_k^h + h - 1, d_k^w \leq j \leq d_k^w + w - 1\}$  - координаты пикселей, принадлежащих части изображения с маской при известном  $d_k$ .

Теперь "разбиваем" такую модель нашим незнанием о расположении маски  $d_k$  для каждого изображения  $X_k$ . Распределение на неизвестные координаты маски на изображении зададим общим для всех изображений с помощью матрицы параметров  $A \in \mathbb{R}^{(H-h+1) \times (W-w+1)}$  следующим образом:

$$\begin{aligned} p(d_k | A) &= A(d_k^h, d_k^w), \\ \sum_{ij} A(i, j) &= 1, \end{aligned} \quad (2)$$

где  $A(i, j)$  — элемент матрицы  $A$ .

И тогда совместное распределение наблюдаемых и латентных переменных - оно же и полное правдоподобие - формулируется в следующем виде:

$$p(X, d | \theta, A) = \prod_k p(X_k | d_k, \theta) p(d_k | A). \quad (3)$$

## 2.3 Применение ЕМ-алгоритма. Общие формулы

Опишем общую стратегию применения ЕМ-алгоритма к введённой вероятностной модели, опираясь на описанную выше расстановку переменных -  $\{X_k\}_{k=1}^K$  - наблюдаемая выборка,  $\{d_k\}_{k=1}^K$  - латентные переменные,  $\{\theta, A\} = \{F, B, s^2, A\}$  - параметры модели.

Общей задачей является максимизация неполного правдоподобия  $p(X|\theta, A)$ :

$$p(X|\theta, A) \rightarrow \max_{\theta, A} \quad (4)$$

Обозначим за  $q(d)$  некоторое распределение на латентные переменные  $d$  (в будущем оно будет играть роль апостериорного распределения на Е-шаге очередной итерации алгоритма). Перейдём стандартным выводом ЕМ-алгоритма к задаче максимизации нижней вариационной оценки (НВО) логарифма неполного правдоподобия (интегральный символ  $d$  будем обозначать как  $\delta$  во избежание путаницы):

$$\begin{aligned} \log p(X|\theta, A) &= \int q(d) \log p(X|\theta, A) \delta d = \int q(d) \log \frac{p(X, d|\theta, A)}{p(d|X, \theta, A)} \delta d = \\ &= \int q(d) \log \frac{p(X, d|\theta, A)q(d)}{p(d|X, \theta, A)q(d)} \delta d = \int q(d) \log \frac{p(X, d|\theta, A)}{q(d)} \delta d + \int q(d) \log \frac{q(d)}{p(d|X, \theta, A)} \delta d = \\ &= \int q(d) \log \frac{p(X, d|\theta, A)}{q(d)} \delta d + KL(q(d)||p(d|X, \theta, A)) \geq \\ &\geq \int q(d) \log \frac{p(X, d|\theta, A)}{q(d)} \delta d = \mathcal{L}(q, \theta, A) \text{ - НВО логарифма неполного правдоподобия.} \end{aligned} \quad (5)$$

В ЕМ-алгоритме предлагается максимизировать полученную НВО численным методом (оно также кратко называется ELBO, это название будем использовать ниже):

$$\begin{aligned} \mathcal{L}(q, \theta, A) &= \int q(d) \log \frac{p(X, d|\theta, A)}{q(d)} \delta d = \\ &= \int q(d) \log p(X, d|\theta, A) \delta d - \int q(d) \log q(d) \delta d = \\ &= \mathbb{E}_{q(d)} \log p(X, d|\theta, A) - \mathbb{E}_{q(d)} \log q(d) \rightarrow \max_{q, \theta, A} \end{aligned} \quad (6)$$

Итерации численного метода состоят из двух этапов: Е- и М-шагов. На Е-шаге вычисляется оценка на апостериорное распределение на координаты маски на изображениях. На данном этапе происходит минимизация (обнуление) дивергенции Кульбака-Лейблера  $KL(q(d)||p(d|X, \theta, A))$ , полученной ранее в выводе ЕМ-алгоритма, что эквивалентно максимизации  $\mathcal{L}(q, \theta, A)$  по  $q$ :

$$q(d) = p(d|X, \theta, A) = \prod_k p(d_k|X_k, \theta, A) = \prod_k q_k(d_k) \quad (7)$$

На М-шаге вычисляется точечная оценка на параметры  $\theta, A$ :

$$\mathbb{E}_{q(d)} \log p(X, d|\theta, A) \rightarrow \max_{\theta, A} \quad (8)$$

Кроме рассмотренного классического применения ЕМ-алгоритма будет изучен упрощенный вариант - hard EM. В нем после Е-шага берется не все апостериорное распределение на координаты маски на изображениях, а только оценка максимума апостериорного распределения на эти координаты.

Итак, формулы (7), (8) задают формулы для выполнения одной итерации ЕМ-алгоритма.

### 3 Вывод ЕМ-алгоритма

Для того, чтобы программно реализовать метод и контролировать корректность его работы, необходимо аналитически вывести формулы для трёх программных блоков: Е- и М-шаги, а также формула для вычисления НВО для логарифма неполного правдоподобия (ELBO  $\mathcal{L}(q, \theta, A)$ ) для мониторинга процесса обучения.

Примем следующие обозначения:

$$\begin{aligned} I &= I(d_k) = \{i : i \in [d_k^h, d_k^h + h - 1]\}, \\ J &= J(d_k) = \{j : j \in [d_k^w, d_k^w + w - 1]\}, \\ \bar{I} &= \bar{I}(d_k) = [0, H - 1] \setminus I, \\ \bar{J} &= \bar{J}(d_k) = [0, W - 1] \setminus J; \end{aligned} \quad (9)$$

Забегая вперёд, отметим, что для реализации алгоритма нам потребуется сперва вычислить распределение  $p(X_k|d_k, F, B, s)$  и его логарифм. Выделим для этого отдельный подраздел.

#### 3.1 $p(X_k|d_k, F, B, s)$ и его логарифм

Используем введённую вероятностную модель, пройдёмся по всем пикселям изображения (используя описанную модель (1)):

$$\begin{aligned} &p(X_k|d_k, F, B, s) = \\ &= \prod_{i \in I} \prod_{j \in J} \mathcal{N}\left(X_k(i, j) \mid F(i - d_k^h, j - d_k^w), s^2\right) \times \prod_{i \in \bar{I}} \prod_{j \in \bar{J}} \mathcal{N}\left(X_k(i, j) \mid B(i, j), s^2\right) = \\ &= \prod_{i \in I} \prod_{j \in J} \frac{1}{\sqrt{2\pi}s} \exp\left\{-\frac{1}{2s^2} (X_k(i, j) - F(i - d_k^h, j - d_k^w))^2\right\} \times \prod_{i \in \bar{I}} \prod_{j \in \bar{J}} \frac{1}{\sqrt{2\pi}s} \exp\left\{-\frac{1}{2s^2} (X_k(i, j) - B(i, j))^2\right\} = \\ &= \frac{1}{(2\pi s^2)^{\frac{WH}{2}}} \exp\left\{-\frac{1}{2s^2} \left(\sum_{i \in I} \sum_{j \in J} (X_k(i, j) - F(i - d_k^h, j - d_k^w))^2 + \sum_{i \in \bar{I}} \sum_{j \in \bar{J}} (X_k(i, j) - B(i, j))^2\right)\right\} \end{aligned} \quad (10)$$

Замена:

$$Q(X_k, F, B, d_k) = \sum_{i \in I} \sum_{j \in J} (X_k(i, j) - F(i - d_k^h, j - d_k^w))^2 + \sum_{i \in \bar{I}} \sum_{j \in \bar{J}} (X_k(i, j) - B(i, j))^2 \quad (11)$$

Итак,

$$\begin{aligned} p(X_k|d_k, F, B, s) &= (2\pi s^2)^{-\frac{WH}{2}} \exp\left\{-\frac{1}{2s^2} Q(X_k, F, B, d_k)\right\} \\ \log p(X_k|d_k, F, B, s) &= -\frac{WH}{2} \log(2\pi s^2) - \frac{1}{2s^2} Q(X_k, F, B, d_k) \end{aligned} \quad (12)$$

И полное правдоподобие для одного объекта имеет такой вид:

$$\log p(X_k, d_k | F, B, s, A) = \log \left( p(X_k | d_k, F, B, s) p(d_k | A) \right) = -\frac{WH}{2} \log(2\pi s^2) - \frac{1}{2s^2} Q(X_k, F, B, d_k) + \log A(d_k^h, d_k^w) \quad (13)$$

### 3.2 Е-шаг, $p(d_k | X_k, F, B, s, A)$

Напомним, что на Е-шаге для максимизации ELBO по распределению  $q$  необходимо приравнять это распределение к апостериорному на латентные переменные. По формуле Байеса:

$$q_k(d_k) = p(d_k | X_k, F, B, s, A) = \frac{p(X_k | d_k, F, B, s) p(d_k | A)}{\sum_{d_n} p(X_k, d_n, F, B, s) p(d_n | A)} \quad (14)$$

Воспользуемся предыдущим разделом (например, возведём (13) в экспоненту):

$$\begin{aligned} p(d_k | X_k, F, B, s, A) &= \frac{(2\pi s^2)^{-\frac{W_H}{2}} \exp\left\{-\frac{1}{2s^2} Q(X_k, F, B, d_k)\right\} A(d_k^h, d_k^w)}{\sum_{d_n^h=0}^{H-h} \sum_{d_n^w=0}^{W-w} (2\pi s^2)^{-\frac{W_H}{2}} \exp\left\{-\frac{1}{2s^2} Q(X_k, F, B, d_n)\right\} A(d_n^h, d_n^w)} = \\ &= \frac{\exp\left\{-\frac{1}{2s^2} Q(X_k, F, B, d_k)\right\} A(d_k^h, d_k^w)}{\sum_{d_n^h=0}^{H-h} \sum_{d_n^w=0}^{W-w} \exp\left\{-\frac{1}{2s^2} Q(X_k, F, B, d_n)\right\} A(d_n^h, d_n^w)} \end{aligned} \quad (15)$$

Воспользуемся приёмом для увеличения стабильности нормировки:

$$\begin{aligned} q_k(d_k) &= p(d_k | X_k, F, B, s, A) = \frac{\exp\left\{-\frac{1}{2s^2} Q(X_k, F, B, d_k) + \log A(d_k^h, d_k^w) - M(X_k, F, B, s, A)\right\}}{\sum_{d_n^h=0}^{H-h} \sum_{d_n^w=0}^{W-w} \exp\left\{-\frac{1}{2s^2} Q(X_k, F, B, d_n) + \log A(d_n^h, d_n^w) - M(X_k, F, B, s, A)\right\}}, \\ \text{где } M(X_k, F, B, s, A) &= \max_{d_n} \left\{ -\frac{1}{2s^2} Q(X_k, F, B, d_n) + \log A(d_n^h, d_n^w) \right\} \end{aligned} \quad (16)$$

### 3.3 М-шаг

Напомним, какую задачу необходимо решить на М-шаге:

$$L = L(F, B, s, A) = \mathbb{E}_{q(d)} \log p(X, d | F, B, s, A) \rightarrow \max_{F, B, s, A} \quad (17)$$

Пары  $(X_k, d_k)$  - независимы. Можно использовать это упрощение в М-шаге EM-алгоритма.

$$\begin{aligned} L(F, B, s, A) &= \mathbb{E}_{q(d)} \log p(X, d | F, B, s, A) = \mathbb{E}_{q(d)} \sum_{k=1}^K \log p(X_k, d_k | F, B, s, A) = \sum_{k=1}^K \mathbb{E}_{q_k(d_k)} \log p(X_k, d_k | F, B, s, A) \\ L &= \sum_{k=1}^K \sum_{d_k^h=0}^{H-h} \sum_{d_k^w=0}^{W-w} q_k(d_k) \log p(X_k, d_k | F, B, s, A) = \sum_{k=1}^K \sum_{d_k^h=0}^{H-h} \sum_{d_k^w=0}^{W-w} q_k(d_k) \left( \log p(X_k | d_k, F, B, s) + \log p(d_k | A) \right) \\ L &= \sum_{k=1}^K \sum_{d_k^h=0}^{H-h} \sum_{d_k^w=0}^{W-w} q_k(d_k) \left( -\frac{WH}{2} \log(2\pi s^2) - \frac{1}{2s^2} Q(X_k, F, B, d_k) + \log A(d_k^h, d_k^w) \right) \end{aligned} \quad (18)$$

Также есть ограничения:

$$\sum_{ij} A_{ij} = 1 \quad (19)$$

При дифференцировании по параметрам учитываем, что часть выражения с апостериорным распределением на латентные переменные  $q_k(d_k)$  состоит из фиксированных параметров с прошлой итерации, и по ним дифференцировать не нужно. Они в этой части не варьируются.

Окончательно имеем при переходе к лагранжиану для условной задачи минимизации (переход от исходной задачи максимизации  $L(F, B, s, A) \rightarrow \max_{F, B, s, A}$  сопровождается сменой знака в большой скобке в (18)):

$$\begin{aligned} &\text{Lagrangian} = \\ &= \sum_{k=1}^K \sum_{d_k^h=0}^{H-h} \sum_{d_k^w=0}^{W-w} q_k(d_k) \left( \frac{WH}{2} \log(2\pi s^2) + \frac{1}{2s^2} Q(X_k, F, B, d_k) - \log A(d_k^h, d_k^w) \right) - \lambda \left( 1 - \sum_{d_n^h=0}^{H-h} \sum_{d_n^w=0}^{W-w} A(d_n^h, d_n^w) \right) \end{aligned} \quad (20)$$

Теперь последовательно выразим каждый параметр, оптимизирующий ELBO на М-шаге, дифференцируя полученный лагранжиан.

$$A(d_k^h, d_k^w)$$

$$\frac{\partial \text{Lagrangian}}{\partial A(d_k^h, d_k^w)} = - \sum_{k=1}^K \left[ \frac{q_k(d_k)}{A(d_k^h, d_k^w)} \right] + \lambda = 0$$

Отметим, что сами элементы матрицы  $A$  не привязаны к конкретным изображениям  $X_k$ . Поэтому их можно вынести за знак суммы:

$$\begin{aligned} \frac{\partial \text{Lagrangian}}{\partial A(d_k^h, d_k^w)} &= \frac{1}{A(d_k^h, d_k^w)} \sum_{k=1}^K \left[ q_k(d_k) \right] - \lambda = 0 \\ A(d_k^h, d_k^w) &= \frac{\sum_{k=1}^K [q_k(d_k)]}{\lambda} \\ \sum_{d_n^h=0}^{H-h} \sum_{d_n^w=0}^{W-w} A(d_k^h, d_k^w) &= \frac{\sum_{k=1}^K \sum_{d_n^h=0}^{H-h} \sum_{d_n^w=0}^{W-w} [q_k(d_k)]}{\lambda} \\ 1 = \frac{K}{\lambda} &\implies \lambda = K \\ A(d_k^h, d_k^w) &= \frac{1}{K} \sum_{n=1}^K q_n(d_k) \end{aligned} \tag{21}$$

$$F(i, j)$$

$$\frac{\partial \text{Lagrangian}}{\partial F(i, j)} = 2 \sum_{k=1}^K \sum_{d_k^h=0}^{H-h} \sum_{d_k^w=0}^{W-w} \left[ q_k(d_k) \frac{1}{2s^2} (F(i, j) - X_k(i + d_k^h, j + d_k^w)) \right] = 0$$

$$\frac{1}{s^2} F(i, j) \sum_{k=1}^K \sum_{d_k^h=0}^{H-h} \sum_{d_k^w=0}^{W-w} q_k(d_k) = \frac{1}{s^2} \sum_{k=1}^K \sum_{d_k^h=0}^{H-h} \sum_{d_k^w=0}^{W-w} q_k(d_k) X_k(i + d_k^h, j + d_k^w))$$

$$KF(i, j) = \sum_{k=1}^K \sum_{d_k^h=0}^{H-h} \sum_{d_k^w=0}^{W-w} q_k(d_k) X_k(i + d_k^h, j + d_k^w))$$

$$F(i, j) = \frac{1}{K} \sum_{k=1}^K \sum_{d_k^h=0}^{H-h} \sum_{d_k^w=0}^{W-w} q_k(d_k) X_k(i + d_k^h, j + d_k^w)) \tag{22}$$

$$B(i, j)$$

$$\frac{\partial \text{Lagrangian}}{\partial B(i, j)} = 2 \sum_{k=1}^K \sum_{d_k^h=0}^{H-h} \sum_{d_k^w=0}^{W-w} \left[ q_k(d_k) \frac{1}{2s^2} ((B(i, j) - X_k(i, j)) - (B(i, j) - X_k(i, j)) \times \mathbb{I}[d_k^h \leq i \leq d_k^h + h - 1, d_k^w \leq j \leq d_k^w + w - 1]) \right] = 0$$

$$B(i, j) \sum_{k=1}^K \sum_{d_k^h=0}^{H-h} \sum_{d_k^w=0}^{W-w} q_k(d_k) - \sum_{k=1}^K X_k(i, j) \sum_{d_k^h=0}^{H-h} \sum_{d_k^w=0}^{W-w} q_k(d_k) =$$

$$= B(i, j) \sum_{k=1}^K \sum_{d_k^h=\max(0, i-h+1)}^{\min(H-h, i)} \sum_{d_k^w=\max(0, j-w+1)}^{\min(W-w, j)} q_k(d_k) - \sum_{k=1}^K X_k(i, j) \sum_{d_k^h=\max(0, i-h+1)}^{\min(H-h, i)} \sum_{d_k^w=\max(0, j-w+1)}^{\min(W-w, j)} q_k(d_k)$$

$$KB(i, j) - B(i, j) \sum_{k=1}^K \sum_{d_k^h=\max(0, i-h+1)}^{\min(H-h, i)} \sum_{d_k^w=\max(0, j-w+1)}^{\min(W-w, j)} q_k(d_k) = \sum_{k=1}^K X_k(i, j) - \sum_{k=1}^K X_k(i, j) \sum_{d_k^h=\max(0, i-h+1)}^{\min(H-h, i)} \sum_{d_k^w=\max(0, j-w+1)}^{\min(W-w, j)} q_k(d_k)$$

$$B(i, j) \left( K - \sum_{k=1}^K \sum_{d_k^h=\max(0, i-h+1)}^{\min(H-h, i)} \sum_{d_k^w=\max(0, j-w+1)}^{\min(W-w, j)} q_k(d_k) \right) = \sum_{k=1}^K X_k(i, j) \left( 1 - \sum_{d_k^h=\max(0, i-h+1)}^{\min(H-h, i)} \sum_{d_k^w=\max(0, j-w+1)}^{\min(W-w, j)} q_k(d_k) \right)$$

$$B(i, j) = \frac{\sum_{k=1}^K X_k(i, j) \left( 1 - \sum_{d_k^h=\max(0, i-h+1)}^{\min(H-h, i)} \sum_{d_k^w=\max(0, j-w+1)}^{\min(W-w, j)} q_k(d_k) \right)}{K - \sum_{k=1}^K \sum_{d_k^h=\max(0, i-h+1)}^{\min(H-h, i)} \sum_{d_k^w=\max(0, j-w+1)}^{\min(W-w, j)} q_k(d_k)} \tag{23}$$

$$\begin{aligned}
\frac{\partial \text{Lagrangian}}{\partial s} &= \frac{KWH}{s} - \frac{1}{s^3} \sum_{k=1}^K \sum_{d_k^h=0}^{H-h} \sum_{d_k^w=0}^{W-w} q_k(d_k) Q(X_k, F, B, d_k) = 0 \\
s^2 &= \frac{1}{KWH} \sum_{k=1}^K \sum_{d_k^h=0}^{H-h} \sum_{d_k^w=0}^{W-w} q_k(d_k) Q(X_k, F, B, d_k)
\end{aligned} \tag{24}$$

Фомулы (21) - (24) задают правила обновления параметров  $A, F, B, s^2$  соответственно на M-шаге.

### 3.4 ELBO

Аналогично выводу M-шага:

$$\begin{aligned}
\mathcal{L}(F, B, s, A) &= \mathbb{E}_{q(d)} \log p(X, d|F, B, s, A) - \mathbb{E}_{q(d)} \log q(d) = \mathbb{E}_{q(d)} \sum_{k=1}^K \log p(X_k, d|F, B, s, A) - \mathbb{E}_{q(d)} \log q(d) = \\
&= \sum_{k=1}^K \mathbb{E}_{q_k(d_k)} \log p(X_k|d_k, F, B, s) + \sum_{k=1}^K \mathbb{E}_{q_k(d_k)} \log p(d_k|A) - \sum_{k=1}^K \mathbb{E}_{q_k(d_k)} \log q_k(d_k) \\
\mathcal{L}(F, B, s, A) &= \sum_{k=1}^K \sum_{d_k^h=0}^{H-h} \sum_{d_k^w=0}^{W-w} q_k(d_k) \left( \log p(X_k|d_k, F, B, s) + \log p(d_k|A) - \log q_k(d_k) \right) \\
\mathcal{L}(F, B, s, A) &= \sum_{k=1}^K \sum_{d_k^h=0}^{H-h} \sum_{d_k^w=0}^{W-w} q_k(d_k) \left( -\frac{WH}{2} \log(2\pi s^2) - \frac{1}{2s^2} Q(X_k, F, B, d_k) + \log A(d_k^h, d_k^w) \right)
\end{aligned} \tag{25}$$

Примем обозначение:

$$r(k) := \exp \left\{ -\frac{1}{2s^2} Q(X_k, F, B, d_k) \right\} A(d_k^h, d_k^w) \tag{26}$$

Тогда:

$$\begin{aligned}
\mathcal{L}(q, \theta, A) &= \sum_{k=1}^K \frac{\sum_{d_k^h=0}^{H-h} \sum_{d_k^w=0}^{W-w} r(k) \left( -\frac{WH}{2} \log(2\pi s^2) - \frac{1}{2s^2} Q(X_k, F, B, d_k) + \log A(d_k^h, d_k^w) \right)}{\sum_{d_n^h=0}^{H-h} \sum_{d_n^w=0}^{W-w} r(n)} - \\
&- \sum_{k=1}^K \frac{\sum_{d_k^h=0}^{H-h} \sum_{d_k^w=0}^{W-w} r(k) \left( -\frac{1}{2s^2} Q(X_k, F, B, d_k) + \log A(d_k^h, d_k^w) - \log \left[ \sum_{d_n^h=0}^{H-h} \sum_{d_n^w=0}^{W-w} r(n) \right] \right)}{\sum_{d_n^h=0}^{H-h} \sum_{d_n^w=0}^{W-w} r(n)} \\
\mathcal{L}(q, \theta, A) &= \sum_{k=1}^K \frac{\left( -\frac{WH}{2} \log(2\pi s^2) - \log \left[ \sum_{d_n^h=0}^{H-h} \sum_{d_n^w=0}^{W-w} r(n) \right] \right) \sum_{d_k^h=0}^{H-h} \sum_{d_k^w=0}^{W-w} r(k)}{\sum_{d_n^h=0}^{H-h} \sum_{d_n^w=0}^{W-w} r(n)} \\
\mathcal{L}(q, \theta, A) &= \sum_{k=1}^K \left( -\frac{WH}{2} \log(2\pi s^2) - \log \left[ \sum_{d_n^h=0}^{H-h} \sum_{d_n^w=0}^{W-w} r(n) \right] \right) \\
\mathcal{L}(q, \theta, A) &= -\frac{KWH}{2} \log(2\pi s^2) - \sum_{k=1}^K \log \left[ \sum_{d_n^h=0}^{H-h} \sum_{d_n^w=0}^{W-w} \exp \left\{ -\frac{1}{2s^2} Q(X_k, F, B, d_n) \right\} A(d_n^h, d_n^w) \right]
\end{aligned} \tag{27}$$

## 4 Вывод hard EM-алгоритма

Для hard EM-алгоритма первые этапы каждой итерации дублируют обычный EM-алгоритм, формулы которого рассмотрены выше.

### 4.1 $p(X_k|d_k, F, B, s)$ и его логарифм

$$p(X_k|d_k, F, B, s) = (2\pi s^2)^{-\frac{WH}{2}} \exp\left\{-\frac{1}{2s^2}Q(X_k, F, B, d_k)\right\} \quad (28)$$

$$\log p(X_k|d_k, F, B, s) = -\frac{WH}{2} \log(2\pi s^2) - \frac{1}{2s^2}Q(X_k, F, B, d_k) \quad (29)$$

Полное правдоподобие для одного объекта:

$$\log p(X_k, d_k|F, B, s, A) = \log(p(X_k|d_k, F, B, s)p(d_k|A)) = -\frac{WH}{2} \log(2\pi s^2) - \frac{1}{2s^2}Q(X_k, F, B, d_k) + \log A(d_k^h, d_k^w) \quad (30)$$

### 4.2 E-шаг, $p(d_k|X_k, F, B, s, A)$

$$q_k(d_k) = p(d_k|X_k, F, B, s, A) = \frac{\exp\left\{-\frac{1}{2s^2}Q(X_k, F, B, d_k) + \log A(d_k^h, d_k^w) - M(X_k, F, B, s, A)\right\}}{\sum_{d_n^h=0}^{H-h} \sum_{d_n^w=0}^{W-w} \exp\left\{-\frac{1}{2s^2}Q(X_k, F, B, d_n) + \log A(d_n^h, d_n^w) - M(X_k, F, B, s, A)\right\}},$$

где  $M(X_k, F, B, s, A) = \max_{d_n} \left\{-\frac{1}{2s^2}Q(X_k, F, B, d_n) + \log A(d_n^h, d_n^w)\right\}$

(31)

После вычисления апостериорного распределения для каждому изображению  $X_k$  сопоставляется аргмаксимум апостериорного распределения  $\arg \max_{d_n} q_k(d_n)$ . В hard EM-алгоритме для простоты мы будем обозначать его как  $d_k$ . Уточним ещё раз: в обычном EM за  $d_k$  была обозначена случайная величина, которая играла роль аргументов для распределений изображения  $X_k$ . В hard EM  $d_k$  - это аргмаксимум апостериорного распределения, и для каждого изображения  $X_k$  он зафиксирован. В hard EM-алгоритме не будет участвовать апостериорное распределение на  $d_k$ , только его аргмаксимум. За  $\tilde{q}_k(d_k) = \tilde{p}(d_k|X_k, F, B, s, A)$  обозначим вырожденное распределение с единственным значением, имеющим ненулевую вероятность, достигающимся на максимуме обычного апостериорного распределения, полученного на E-шаге.

### 4.3 M-шаг

$$L(F, B, s, A) = \mathbb{E}_{\tilde{q}(d)} \log p(X, d|F, B, s, A) \rightarrow \max_{F, B, s, A} \quad (32)$$

Аналогично прошлому разделу, пары  $(X_k, d_k)$  - независимы. Можно использовать это упрощение в M-шаге EM-алгоритма. Ниже за максимум апостериорного распределения на  $d_k$  обозначаются пары  $(d_k^h, d_k^w) = d_k$ .

$$L(F, B, s, A) = \mathbb{E}_{\tilde{q}(d)} \log p(X, d|F, B, s, A) = \mathbb{E}_{\tilde{q}(d)} \sum_{k=1}^K \log p(X_k, d|F, B, s, A) = \sum_{k=1}^K \mathbb{E}_{\tilde{q}_k(d_k)} \log p(X_k, d_k|F, B, s, A)$$

$$L = L(F, B, s, A) = \sum_{k=1}^K \tilde{p}(d_k|X_k, F, B, s, A) \log p(X_k, d_k|F, B, s, A) = \sum_{k=1}^K \log p(X_k, d_k|F, B, s, A)$$

$$L = \sum_{k=1}^K \left( -\frac{WH}{2} \log(2\pi s^2) - \frac{1}{2s^2}Q(X_k, F, B, d_k) + \log A(d_k^h, d_k^w) \right)$$
(33)

Так же есть ограничения:

$$\sum_{ij} A_{ij} = 1 \quad (34)$$

Аналогично прошлому пункту перейдём к рассмотрению лагранжиана условной задачи минимизации:

$$\text{Lagrangian} = \frac{KWH}{2} \log(2\pi s^2) + \sum_{k=1}^K \left( \frac{1}{2s^2}Q(X_k, F, B, d_k) - \log A(d_k^h, d_k^w) \right) - \lambda \left( 1 - \sum_{d_k^h=0}^{H-h} \sum_{d_k^w=0}^{W-w} A(d_k^h, d_k^w) \right)$$
(35)

$A(d_k^h, d_k^w)$

Обозначим за  $K_{d_k}$  число реализованных аргмаксимумов апостериорного распределения положения лица в позиции  $d_k$ . В данном пункте  $d_k$  взаимно однозначно сопоставляется  $X_k$ .

$$\begin{aligned} \frac{\partial \text{Lagrangian}}{\partial A(d_k^h, d_k^w)} &= -\frac{K_{d_k}}{A(d_k^h, d_k^w)} + \lambda = 0 \implies A(d_k^h, d_k^w) = \frac{K_{d_k}}{\lambda} \\ \sum_{d_n^h=0}^{H-h} \sum_{d_n^w=0}^{W-w} A(d_k^h, d_k^w) &= \frac{\sum_{d_n^h=0}^{H-h} \sum_{d_n^w=0}^{W-w} K_{d_k}}{\lambda} \\ 1 &= \frac{K}{\lambda} \implies \lambda = K \\ A(d_k^h, d_k^w) &= \frac{K_{d_k}}{K} \end{aligned} \quad (36)$$

$F(i, j)$

$$\begin{aligned} \frac{\partial \text{Lagrangian}}{\partial F(i, j)} &= 2 \sum_{k=1}^K \left[ \frac{1}{2s^2} (F(i, j) - X_k(i + d_k^h, j + d_k^w)) \right] = 0 \\ F(i, j) \sum_{k=1}^K 1 &= \sum_{k=1}^K X_k(i + d_k^h, j + d_k^w) \\ F(i, j) &= \frac{1}{K} \sum_{k=1}^K X_k(i + d_k^h, j + d_k^w) \end{aligned} \quad (37)$$

$B(i, j)$

$$\begin{aligned} \frac{\partial \text{Lagrangian}}{\partial B(i, j)} &= 2 \sum_{k=1}^K \left[ \frac{1}{2s^2} ((B(i, j) - X_k(i, j)) - (B(i, j) - X_k(i, j)) \mathbb{I}[d_k^h \leq i \leq d_k^h + h - 1, d_k^w \leq j \leq d_k^w + w - 1]) \right] = 0 \\ B(i, j) \sum_{k=1}^K 1 - \sum_{k=1}^K X_k(i, j) &= B(i, j) \sum_{k=1}^K \mathbb{I}[d_k^h \leq i \leq d_k^h + h - 1, d_k^w \leq j \leq d_k^w + w - 1] - \\ &\quad - \sum_{k=1}^K X_k(i, j) \mathbb{I}[d_k^h \leq i \leq d_k^h + h - 1, d_k^w \leq j \leq d_k^w + w - 1] \\ KB(i, j) - B(i, j) \sum_{d_k^h=\max(0, i-h+1)}^{\min(H-h, i)} \sum_{d_k^w=\max(0, j-w+1)}^{\min(W-w, j)} K_{d_k} &= \sum_{k=1}^K X_k(i, j) - \sum_{k=1}^K X_k(i, j) \mathbb{I}[d_k^h \leq i \leq d_k^h + h - 1, d_k^w \leq j \leq d_k^w + w - 1] \\ B(i, j) \left( K - \sum_{d_k^h=\max(0, i-h+1)}^{\min(H-h, i)} \sum_{d_k^w=\max(0, j-w+1)}^{\min(W-w, j)} K_{d_k} \right) &= \sum_{k=1}^K X_k(i, j) \left( 1 - \mathbb{I}[d_k^h \leq i \leq d_k^h + h - 1, d_k^w \leq j \leq d_k^w + w - 1] \right) \\ B(i, j) &= \frac{\sum_{k=1}^K X_k(i, j) \left( 1 - \mathbb{I}[d_k^h \leq i \leq d_k^h + h - 1, d_k^w \leq j \leq d_k^w + w - 1] \right)}{K - \sum_{d_k^h=\max(0, i-h+1)}^{\min(H-h, i)} \sum_{d_k^w=\max(0, j-w+1)}^{\min(W-w, j)} K_{d_k}} \\ \text{Или: } B(i, j) &= \frac{\sum_{k=1}^K X_k(i, j) \left( 1 - \mathbb{I}[i-h+1 \leq d_k^h \leq i, j-w+1 \leq d_k^w \leq j] \right)}{K - \sum_{k=1}^K \mathbb{I}[i-h+1 \leq d_k^h \leq i, j-w+1 \leq d_k^w \leq j]} \end{aligned} \quad (38)$$

Пояснение: усредняем значения в пикселе для тех  $X_k$ , аргмаксимум по  $d_k$  которых не попадает в ограничительный прямоугольник маски.

$s^2$

$$\begin{aligned} \frac{\partial \text{Lagrangian}}{\partial s} &= \frac{KWH}{s} - \frac{1}{s^3} \sum_{k=1}^K Q(X_k, F, B, d_k) = 0 \\ s^2 &= \frac{1}{KWH} \sum_{k=1}^K Q(X_k, F, B, d_k) \end{aligned} \quad (39)$$

Фомулы (36) - (39) задают правила обновления параметров  $A, F, B, s^2$  соответственно на M-шаге hard EM-алгоритма.

## 4.4 ELBO

Вывод происходит почти так же, как и для обычного EM алгоритма.

$$\begin{aligned} \mathcal{L}(q, F, B, s, A) &= \mathbb{E}_{q(d)} \log p(X, d|F, B, s, A) - \mathbb{E}_{q(d)} \log q(d) \rightarrow \max_{q, F, B, s, A} \\ \mathcal{L}(F, B, s, A) &= \mathbb{E}_{q(d)} \log p(X, d|F, B, s, A) - \mathbb{E}_{q(d)} \log q(d) = \mathbb{E}_{q(d)} \sum_{k=1}^K \log p(X_k, d|F, B, s, A) - \mathbb{E}_{q(d)} \log q(d) = \\ &= \sum_{k=1}^K \mathbb{E}_{q_k(d_k)} \log p(X_k|d_k, F, B, s) + \sum_{k=1}^K \mathbb{E}_{q_k(d_k)} \log p(d_k|A) - \sum_{k=1}^K \mathbb{E}_{q_k(d_k)} \log q_k(d_k) \end{aligned} \quad (40)$$

В данном случае, при использовании hard EM-алгоритма, выражение значительно упрощается: последняя сумма вырождается в ноль, как энтропия вырожденного распределения (чем является  $q$  после E-шага). В оставшихся суммах для каждого объекта выборки будет участвовать только одно положение маски  $d_k$  - апостериорный аргмаксимум - вероятности остальных значений равны нулю.

$$\mathcal{L}(q, F, B, s, A) = \sum_{k=1}^K \left( -\frac{WH}{2} \log(2\pi s^2) - \frac{1}{2s^2} Q(X_k, F, B, d_k) + \log A(d_k^h, d_k^w) \right) \quad (41)$$

На этом все необходимые формулы для реализации EM-алгоритма можно считать выведенными. Перейдём к экспериментальной части работы.

## 5 Эксперименты

В данном разделе опишем основные свойства работы EM-алгоритма на сгенерированных выборках, а также влияние уровня зашумлённости и количество наблюдаемых объектов на качество результата его работы. Также проведём сравнение EM- и hard EM- алгоритмов по качеству и скорости их действия. В конце представим итог работы алгоритма над исходными данными.

### 5.1 Сгенерированные данные

Опишем, какие изображения будем использовать для быстрого исследования свойств программной реализации EM-алгоритма.

В ходе подбора выяснилось, что для удобной визуализации результатов могут подойти следующие два изображения - фон и маска:



Рис. 2: Фон и маска для экспериментов

Маска сжата вдвое относительно фона, и отзеркалена. Такой выбор данных для экспериментов обусловлен возможностью детального анализа получаемых результатов: на исходных изображениях есть много мелких деталей, по которым можно будет оценивать визуальную точность итогового прогноза.

Размеры фона:  $60 \times 34$ , размеры маски:  $30 \times 17$  пикселей (сначала указан размер по горизонтали, затем по вертикали).

Как было описано в первом разделе, генерация наблюдаемой выборки происходит в два этапа: сначала маска накладывается в произвольное место на фоне, после чего это изображение зашумляется. В качестве примера изобразим одно из получаемых таким образом изображение (рис. 3):



Рис. 3: Пример данных для ЕМ-алгоритма

Обучающую (наблюдаемую выборку) для ЕМ-алгоритма составляет совокупность таких запущенных изображений. Целью алгоритма является прогнозирование вида исходной маски и фона, а также положения самой маски для всех изображений.

Выбранный размер изображений и их собственный вид позволяют быстро прогонять ЕМ-алгоритм для изучения его свойств, а также оценивать качество его работы с визуальной точки зрения.

**Важное замечание:** при визуализации работы алгоритма во время экспериментов будем изображать две спрогнозированные картинки: фон сверху и маску под ним.

## 5.2 Влияние начального приближения на прогноз

Для многих численных методов точность результата определяется по большей части качеством оперируемых данных. Однако зачастую на неё влияет также изначально заданные значения параметров метода. Это может быть справедливо и для ЕМ-алгоритма.

Начнём со следующего замечания: при корректной работе алгоритма спрогнозируемые изображения фона и маски приближаются к истинным. При этом разброс в данных должен снижаться, аналогично "расшумлению" картинок дисперсия шума монотонно убывает от итерации к итерации. Это хорошо видно на следующем графике - результате серии запусков ЕМ-алгоритма:

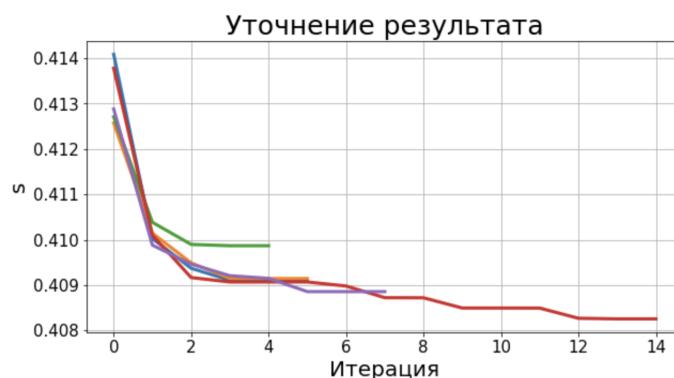


Рис. 4: Снижение разброса в течение работы

Вопрос: что будет, если искусственно занизить исходное значение параметра  $s$ ? Можно считать, что мы даём модели первичное представление того, что данные зашумлены слабо. Модель будет уточнять свой прогноз, но из-за первоначальной малости разброса она не сможет значительно поменять исходные данные и приблизить их к истинным.

На рисунке ниже представлены результаты работы алгоритма с заниженной (слева) и завышенной (справа) дисперсией, полученной в качестве начального приближения:

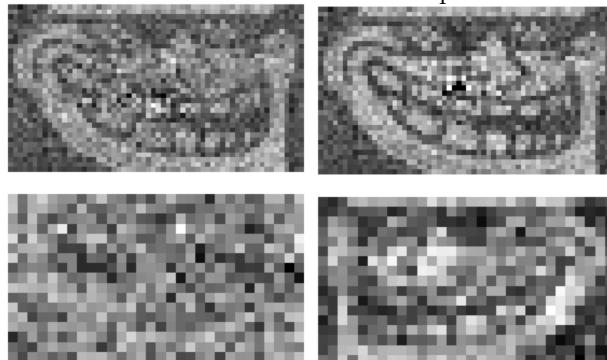


Рис. 5: Варьирование начального значения  $s$

Следующая таблица отражает небольшую статистику по параметрам запуска и точности результата работы метода:

Результат на рисунке	Слева	Справа
Начальное $s$	1e-5	100
ELBO	-32562.65	-31162.33

Рис. 6: Варьирование начального значения  $s$

Видна визуальная разница в чёткости прогноза. Разница в ELBO также присутствует - качество получается выше при более высоком значении инициализации дисперсии шума. То есть, при неудачном выборе  $s$  качество результата может испортиться. В случае параметра  $s$  это не так критично: нам достаточно принять  $s$  за некоторое большое число - чтобы у модели было больше пространства для преобразований.

Теперь искусственно испортим параметр  $A$  - матрицу, задающую распределение на координаты положения маски. Допустим, сделаем это распределение вырожденным - матрица будет нулевой за исключением одного элемента, равного единице. Таким образом, модель на первых итерациях будет уверена, где находится маска - своеобразный аналог переобучения. Эта уверенность будет ей мешать, алгоритм не успеет перестроиться для обобщения наблюдаемых данных, и поэтому выдаст менее точные результаты.

Например, были получены результаты при такой инициализации  $A$ , при которой модель будет думать, что маска находится в правом верхнем углу всех изображений. Эти результаты представлены на рис. 7 (слева):

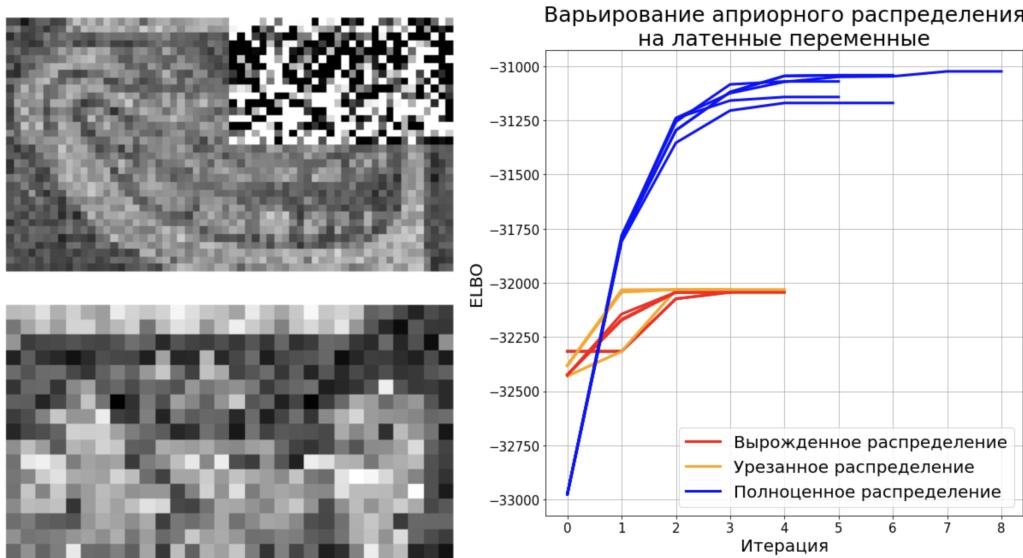


Рис. 7: Результат работы при вырожденном параметре  $A$  (слева).  
Варьирование начального значения  $A$  (справа)

Здесь виден интересный результат. Модель верно спрогнозировала три четверти фона, но оставшуюся его часть (которая верно распознавалась по фоновым пикселям) отдала пикселям маски. То есть алгоритм не стал производить вычисления для "ненужной" правой верхней части фона, так как был уверен в том, что там всегда находится маска. Однако он смог правильно распознать эту четверть фона, но представил её в маске, так как она, по его мнению, всегда находится там. Естественно, данный результат в конечном итоге нам не подходит, однако поведение алгоритма вызывает интерес.

На рис. 7 (справа) график динамики обучения для разных вариантов инициализации матрицы  $A$ : вырожденный случай, когда  $A$  содержит кроме нулей ровно одну единицу; урезанный случай, при котором четверть элементов матрицы  $A$  остаётся ненулевой; и случай полноценной не разреженной матрицы. Для каждого из трёх вариантов было проведено несколько запусков.

Отсюда видно, что при неудачной инициализации параметра  $A$  модель может выдать неточные результаты, и этот параметр так же важен, как и  $s$ . Однако в большинстве случаев случайная генерация матрицы  $A$  предотвратит указанные проблемы, исключив риск подобной вырожденности распределения.

Начальная инициализация маски F и фона B влияет на точность работы метода не столь существенно. Алгоритм позволяет выравнить их достаточно быстро, и неудачный выбор этих параметров вначале не сыграет критической роли во время обучения.

Теперь отойдём от конкретных случаев неудачных параметров, и не будем специально портить их инициализацию. Посмотрим, насколько сильно влияет начальное приближение на точность при обычном запуске. В результате исследования этого вопроса была получена следующая статистика (несколько запусков обозначены разными цветами):

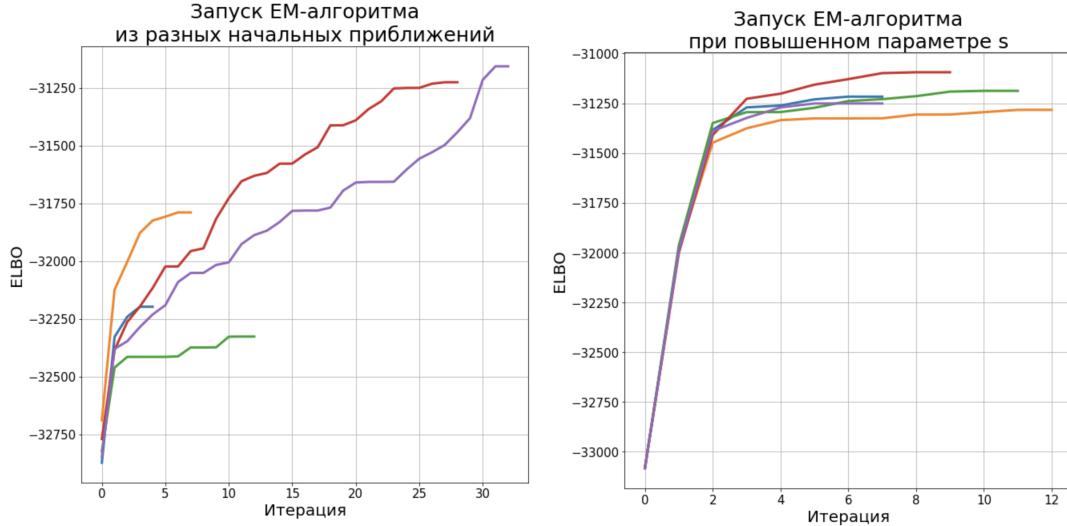


Рис. 8: Запуск из разных начальных приближений

Видно, что в общем случае алгоритм может вести себя по-разному в зависимости от начальной инициализации. Однако можно собственноручно улучшить его качество и снизить разброс в результатах, повышая исходный параметр  $s$ .

### 5.3 Влияние качества и объёма выборки на прогноз

Рассмотрим естественный вопрос, насколько сильно влияют характеристики исходных данных на качество результата работы алгоритма. В данном эксперименте составлялось несколько выборок разного размера и разного уровня зашумлённости. На них производился запуск EM-алгоритма.

При варьировании уровня шума на выборке в 30 изображений были получены следующие результаты:

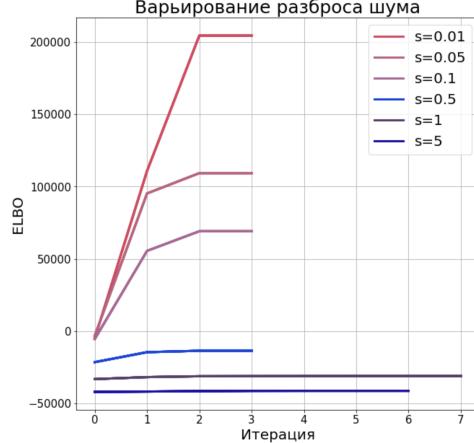


Рис. 9: Влияние зашумлённости на результат

Логично, что чем чётче исходные изображения, тем проще методу сопоставлять их между собой. Поэтому с уменьшением разброса шума метод работает быстрее и точнее. Ещё одно замечание заключается в том, что с понижением чёткости изображений алгоритм сильнее ограничен в своих возможностях, что проявляется в общем снижении темпов роста ELBO при увеличении шума. Когда шум будет настолько сильным, что "перекроет" любую исходную полезную информацию, EM-алгоритм не сможет ничего сделать, и ELBO в таком случае останется практически неизменным.

Ниже приведены результаты экспериментов по изменению запущленности исходных данных:

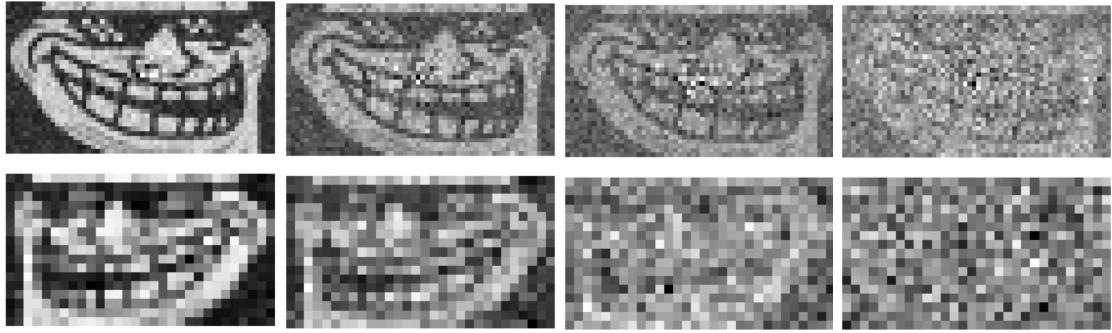


Рис. 10: Результаты работы алгоритма в зависимости от запущленности выборки  $K = 50$

Случай	1	2	3	4
Разброс шума исходных данных $s$	0.5	1	1.5	2.5
Отмасштабированное ELBO	-456.5	-1049.5	-1215.5	-1324.6

Видно, что для незначительных значений шума  $s$  EM-алгоритму удаётся практически безошибочно восстановить исходные фон и маску. При увеличении параметра  $s$  результаты теряют чёткость, и при значении  $s = 2.5$  на этих данных маска не сходится к истинной, и остаётся случайным набором пикселей. При этом видно, что во всех случаях фон распознаётся лучше маски. Это связано с тем, что даже неточный алгоритм обладает большим числом наблюдений для его восстановления. Если продолжить повышать разброс в исходных данных, то EM-алгоритм уже ничего не сможет сделать, так как полезной информации в данных не останется. ELBO, естественно, уменьшается при росте  $s$ , так как алгоритм теряет качество прогноза.

Далее приведём результаты эксперимента по варьированию объёма выборки с фиксированным значением шума  $s = 1$  и с нормированием на объём выборки значений ELBO.

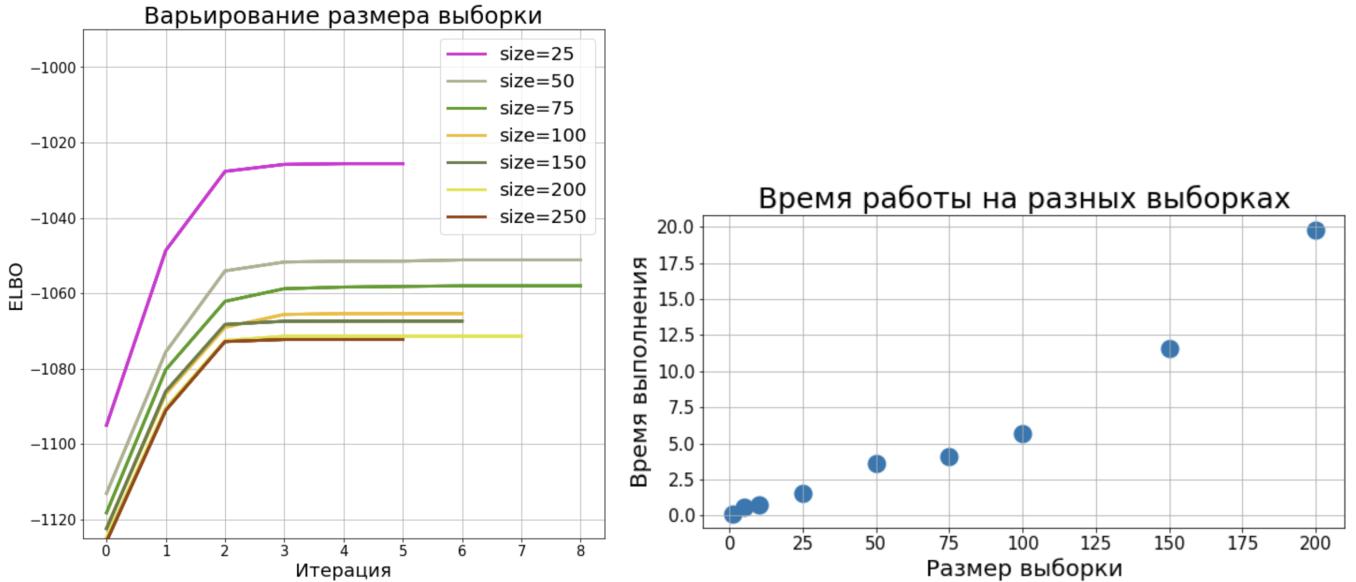


Рис. 11: Результаты варьирования объёма выборки

Здесь можно сделать два наблюдения. Отмасштабированное значение ELBO на больших выборках меньше, чем на маленьких. Это свидетельствует о том, что алгоритму сложнее настроиться на больший объём данных, поэтому он себя и "критикует" сильнее. Однако важно понимать, что фактическое качество работы метода с ростом выборки будет расти. Алгоритму легче проанализировать маленькую выборку и приблизить параметры модели к оптимумам по ней, но точность результата будет зависеть от числа прецедентов, на которые опирается алгоритм, и, естественно, чем больше наблюдений, тем сложнее будет программе, но и тем точнее будет фактический итог.

При увеличении размера выборки время работы алгоритма также сильно вырастает. Соответственно для особо больших данных необходимы эвристики, позволяющие отбирать "более полезные" изображения с наименьшими потерями полезной информации.

На следующем рисунке изображены фактические результаты метода при варьировании выборки (были проведены новые запуски алгоритма):

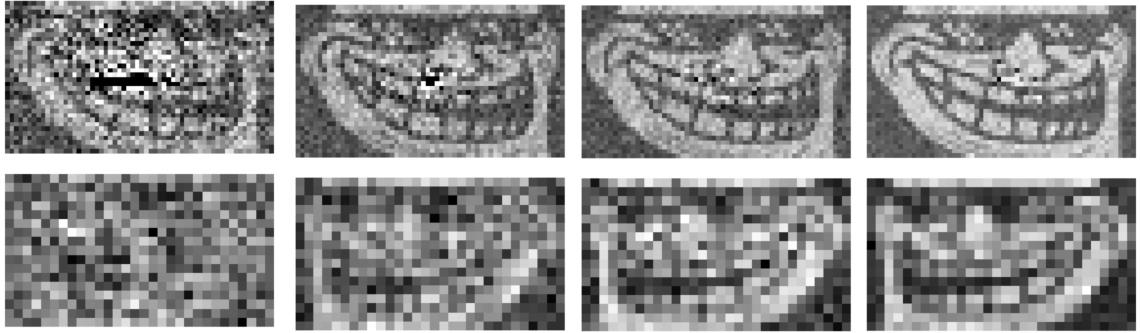


Рис. 12: Результаты варьирования размера выборки при  $s = 1$

Случай	1	2	3	4
Размер выборки $K$	10	30	50	100
Отмасштабированное ELBO	-953.4	-1035.4	-1048.3	-1065.9

Если анализировать эти результаты с визуальной точки зрения, то можно сделать следующие выводы. Во-первых, рост выборки, как было сказано ранее, способствует росту качества полученных изображений. Вместе с этим отмасштабированное под размер выборки значение ELBO будет становиться меньше из-за общего усложнения задачи. Это совершенно не будет означать падение качества финального результата.

На маленьких выборках проявляются "дырки" на прогнозе фона - чёрные необработанные области. Это связано с тем, что у метода не было достаточно наблюдений, у которых были открыты нужные пиксели для фона, и поэтому алгоритм не смог корректно рассчитать яркость нужных пикселей. При увеличении выборки такие дефекты перестают проявляться.

Приведём также результаты одновременного варьирования шума и размера выборки для получения результатов, визуально одинаковых:



Рис. 13: Взаимозависимость шума и размера выборки

Случай	1	2	3	4
Размер выборки $K$	10	30	50	100
Величина разброса шума $s$	0.5	0.7	1	1.5
Отмасштабированное ELBO	-348.4	-783.8	-1054.2	-1229.5

Как видно, с ростом зашумления исходных данных требуется увеличивать объём выборки экспоненциально, чтобы сохранить определённый уровень чёткости результата. Однако, как было замечено ранее, если шум настолько сильный, что перекрывает всю полезную информацию о выборке, то даже увеличения её объёма не поможет вытянуть нужную информацию.

Подытожив, отметим, что на качество работы алгоритма напрямую влияет чёткость и объём наблюдаемых данных. На выборках, включающих "достаточное" число наблюдений алгоритм способен выдавать качественные результаты, однако может затратить на это много времени, и поэтому на больших выборках следует производить предварительный отбор наиболее репрезентативной подвыборки. В то же время, напомним, что понятие "достаточности" прецедентов опирается как на размер изображений, так и на уровень зашумлённости. При усложнении выборки в обоих случаях требуется значительное повышение объёма выборки для сохранения качества работы.

## 5.4 Сравнение EM- и hard EM-алгоритмов

hard EM модификация алгоритма является хорошим примером метода с небольшой потерей точности взамен экономии времени. Его упрощение на M-шаге позволяет эффективно пропускать громоздкие вычисления и на каждой итерации пользоваться вырожденной частью от полученной информации для каждого из наблюдаемых изображений. Приведём ниже результаты его сравнения с обычным EM-алгоритмом. Запуски для этих двух методов производились на одной выборке, включающей 50 изображений с использованием стандартного шума ( $s = 1$ ).

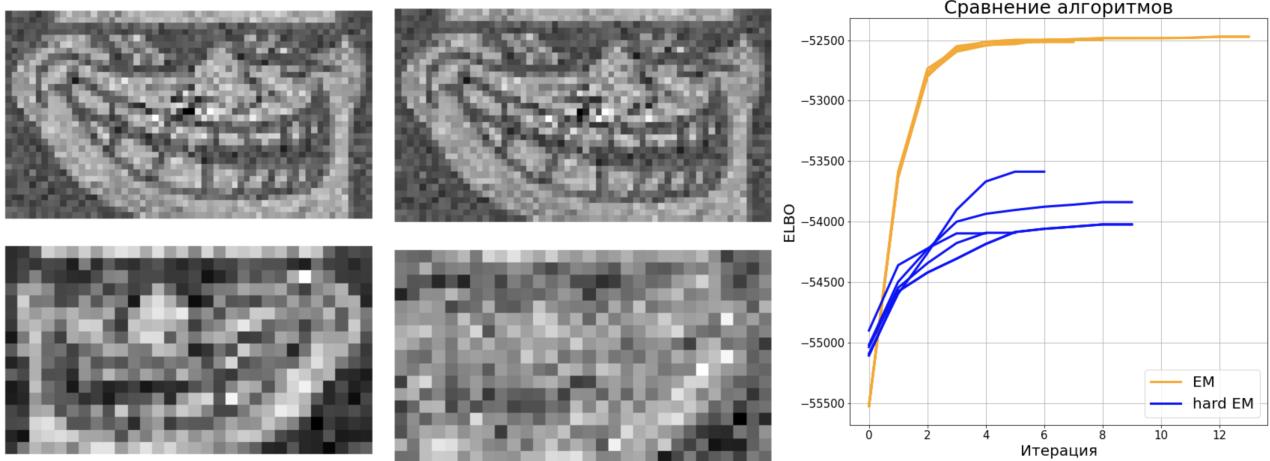


Рис. 14: Результаты запуска EM (слева) и hard EM (в середине), а также их сравнение (справа)

Алгоритм	EM	hard EM
Финальное ELBO	-52471.6	-53603.5
Среднее время выполнения (в сек)	3.41	1.66

Из этой статистики видно, что обычный EM-алгоритм работает стабильнее. Каждый из запусков приводил метод к похожим значениям ELBO, в то время как у hard EM-алгоритма сильный разброс в "удачливости" запуска, на что также влияет и выбор начального приближения. Однако, справедливости ради, отметим, что при дополнительном повышении объёма выборки hard EM работает почти одинаково с EM, но затрачивает меньше времени.

Также отметим, что обе версии метода практически одинаково распознают фон. Для hard EM в данных условиях маска уже даётся сложнее - видны общие паттерны истинного изображения, но без деталей. Однако это весьма неплохой результат, учитывая, что hard EM на каждом M-шаге пользуется только небольшой частью информации, которая становится доступной на E-шаге - наиболее вероятное положение маски. То есть этой информации может быть достаточно для того, чтобы с некоторой невысокой точностью получить необходимые изображения за более короткий промежуток времени - в данном случае выигрыш по времени почти двукратный, на больших изображениях эта экономия проявляется ещё сильнее.

Обозначим дополнительные сложности при использовании hard модификации. Как было сказано ранее, на малых выборках обычный EM-алгоритм может оставлять необработанные "дыры" на изображениях фона, маску на эту недостающую часть фона. Данный эффект может усиливаться для hard EM:

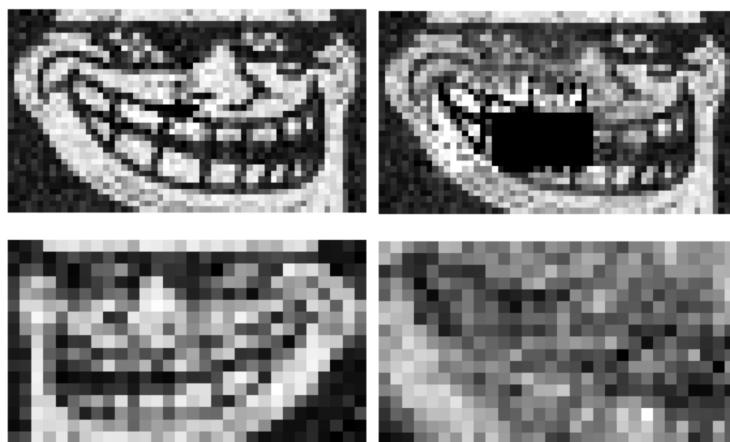


Рис. 15: Проблемный запуск hard EM (справа) и сравнение с работой обычного EM (слева)

Аналогично экспериментам, описанным выше для EM-алгоритма, hard EM смог корректно распознать фон, однако из-за недостаточности прецедентов для обучения посчитал, что маска является его частью, а в самой этой части нет светлых пикселей. Таким образом, метод оставил необработанной часть фона, и перенёс её в маску. Для hard EM-алгоритма, как и для обычного EM, данная проблема устраняется при увеличении объёма выборки, однако для hard модификации на успешность запуска существенное будет влиять начальное приближение. Опять же, для больших выборок, и эта проблема становится несущественной и hard EM работает так же хорошо, как и обычный EM.

Напоследок приведём статистику запусков этих двух версий алгоритма на выборках разного размера:

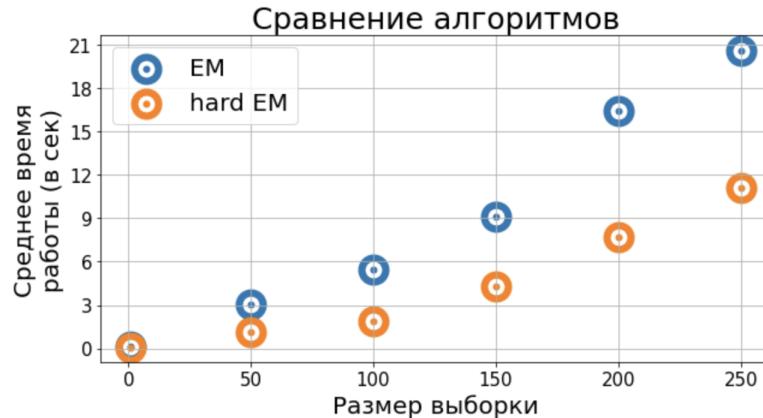


Рис. 16: Сравнение алгоритмов по времени работы

Видно, что hard EM обгоняет обычную версию по времени работы. К тому же на невырожденных по объёму выборках он может демонстрировать визуально похожие результаты прогнозирования, что и является его преимуществом.

## 5.5 Запуск EM-алгоритма на исходных данных

После исследования основных свойств программной реализации на небольших данных был про- ведён запуск EM-алгоритма на большой выборке крупных изображений. Полная выборка состоит из 1000 запутанных изображений, каждое размера  $200 \times 105$ , такой же размер, конечно, имеет и фон. Маска с лицом имеет размеры  $66 \times 100$ .

На рис. 17 приведён пример одного из изображений обучающей выборки:

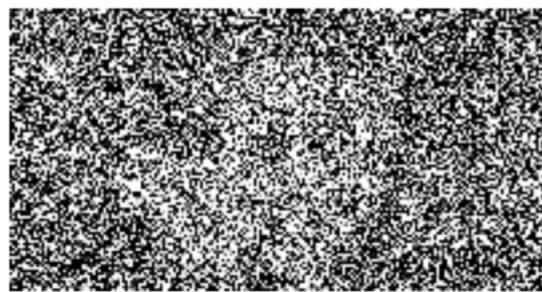


Рис. 17: Пример наблюдаемых данных

В данном эксперименте был запущен EM-алгоритм на трёх выборках, отличающихся по объёму: малая - 50 объектов, средняя - 200 объектов и полная - 1000 объектов.

Результаты запуска приведены на следующей странице.

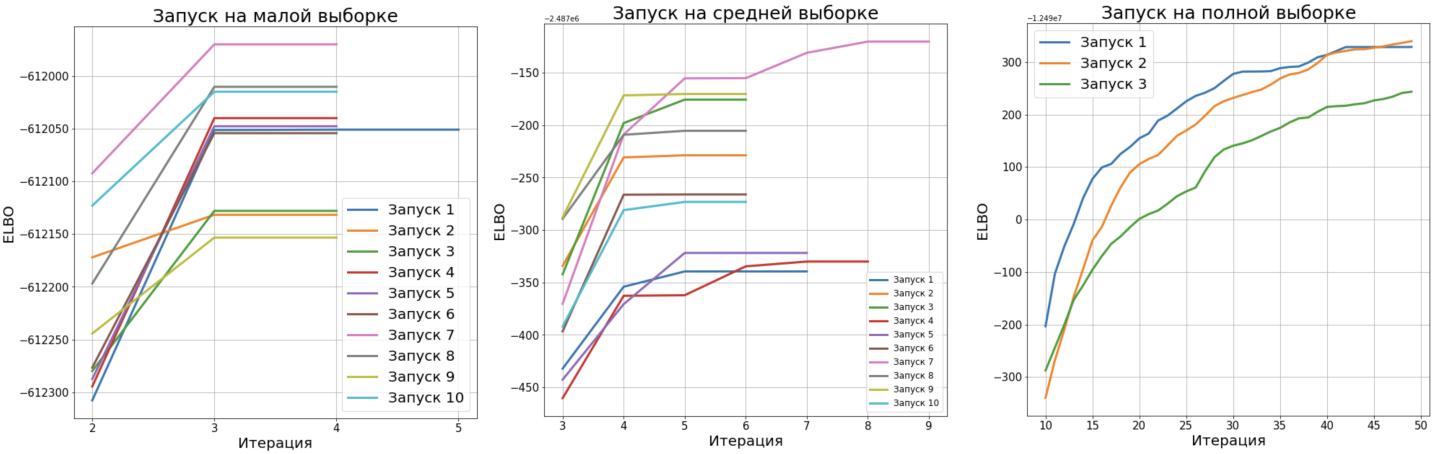


Рис. 18: Процесс работы алгоритма на разных выборках

Выборка	Малая	Средняя	Полная
Размер выборки $K$	50	200	1000
Финальное ELBO	-611 970.14	-2 487 120.40	-12 489 660.24
Финальное нормированное ELBO	-12 239.4	-12 435.6	-12 489.7
Среднее время одного запуска (в сек)	72.2	314.4	12801.0

Рис. 18 отражает основную статистику по работе EM-алгоритма, а рис. 19 - итоговые результаты прогноза. Проанализируем их.

На малой выборке EM-алгоритму, как видно, было достаточно всего несколько итераций, чтобы сойтись к какому-то решению. Процесс сходился на таких запусках быстро, однако смог выдать общие черты истинных изображений. Уже с такой маленькой частью данных можно сказать, где на фоне расположены здания, и в каком положении лицо на маске. Что важно, эти приближённые данные удалось получить за короткое время.

Как вариант, данную процедуру можно было бы попробовать проделать с помощью hard EM-алгоритма и сэкономить ещё больше времени, но нужно учитывать, что он менее стабилен обычного EM в случаях малых выборок. В данном случае для стабильной работы hard EM выборку требуется увеличивать до 250-300 изображений, чтобы избежать проблем, описанных в разделе сравнения EM- и hard EM-алгоритмов.



Рис. 19: Результаты работы алгоритма на разных выборках

Увеличив выборку до 200 изображений, мы можем наблюдать, что алгоритму требуется больше итераций для сходимости, на всех запусках параметры модели сошлись к некоторым значениям, но за более длительное время, что естественно.

Вместе с этим чёткость изображений заметно улучшилась: уже можно уверенно заявить, что в роли фона выступает Красная площадь. Чертые лица на маске выражение становятся разборчивыми, и изображённого улыбающегося человека уже можно узнать, но с некоторым сомнением, конечно.

Запустив алгоритм на полной выборке в 1000 изображений, получается совершенно разборчивые изображения, качество картинок значительно улучшилось, однако и время работы алгоритма стало большим. На такой выборке уже, конечно, можно смело запускать и hard EM-алгоритм, который выдаст такие же результаты, но за меньшее время, так как число прецедентов для обучения в данных условиях безусловно достаточно.

Стали видны мелкие детали на фоне, картинка стала более контрастной и живой. Это же относится и к фото Максима Находного.

Прокомментируем некоторые моменты процесса обучения на полной выборке. В отличие от прошлых выборок, алгоритм завершил свою работу немного раньше, чем успел сойтись, по достижении предела в 50 итераций. Это свидетельствует о том, что методу требуется много времени и вычислительных ресурсов, чтобы обработать такие большие данные, и полностью закончить свою работу он не успевает в силу большого объёма выборки и самих данных.

Значения ELBO (табл. на рис. 18), отмасштабированные под объёмы выборок, очень похожи. Тут, как было описано ранее, проявляется следующий эффект: рост выборки ведёт к небольшому снижению ELBO ввиду того, что алгоритму сложнее настроиться на все доступные данные, когда их становится больше. Вместе с этим, даже здесь видна закономерность, которую мы ранее не описали: нормированное ELBO не просто снижается с ростом выборки, но, похоже, что даже сходится к некоторому предельному значению. Это было видно и на рис. 11, это заметно и для данного эксперимента. Это можно объяснить так: из-за роста выборки алгоритму сложнее настроиться на данные, из-за чего ELBO убывает. Однако новые данные всё же превносят полезную информацию, которая помогает алгоритму сопоставлять разные наблюдения и замедляет падение ELBO. Модели становятся относительно проще обобщить имеющиеся данные. Таким образом, убывание ELBO с ростом выборки на самом деле положительный фактор, символизирующий повышение свойств обобщаемости модели и снижения её переобучаемости.

Таким образом, данный эксперимент ещё раз показал состоятельность EM-алгоритма: данный метод позволил обработать запущённые изображения, и вынести из исходных данных много полезной информации. Его модификация - hard EM-алгоритм - как было показано ранее, способен значительно ускорить время работы метода, а на такой большой выборке он может работать практически без потери качества. В данном разделе не приводились результаты этих запусков для избежания нагромождений - итоговые результаты работы этих методов одинаковы на выборках с размерами 200 и 1000, выигрыш по времени у hard EM примерно двукратный, как и было описано в предыдущих разделах.

## 6 Идеи по модификации алгоритма

В данном разделе опишем методы, которые должны обеспечить рост скорости и точности используемого алгоритма.

Как было замечено ранее, с ростом выборки заметно увеличивается вычислительные и временные затраты метода. Соответственно не рационально запускать честный алгоритм на очень больших выборках. Требуется придумать, как оптимизировать процесс в таком случае, используя все доступные изображения. Дополнительно хотелось бы сохранить качество работы алгоритма.

### Использование генетических алгоритмов

Учитывая, что на небольших выборках алгоритм работает быстро, но не точно, можно попробовать произвести отбор самых релевантных изображений из наблюдаемой выборки, чтобы на них обучать EM в дальнейшем. На роль метода отбора можно предложить генетические алгоритмы: первоначально выборка разбивается на небольшие подвыборки, на каждой из них производится несколько запусков EM-алгоритма, которые можно сравнить по нормированным значениям ELBO. Данный процесс можно распараллелить, так как все расчёты производятся независимо между подвыборками. После этого топ лучших подвыборок по показателю ELBO скрещиваются между собой, получаются новые комбинации непересекающихся подвыборок. С некоторой случайной вероятностью к ним могут вклиниваться единичные не вошедшие эти топы изображения. Процесс запускается снова. После запуска EM опять отбирается топ лучших подвыборок, остальные отбрасываются, а эти скрещиваются со случайнм добавлением новых изображений из исходной выборки, т.д.

Удерживая небольшим объём подвыборок можно относительно быстро определить наиболее подходящие изображения для включения в используемую выборку. Это позволит приблизительно оценить важность каждого из наблюдаемых объектов и составить финальную выборку из наиболее представительных.

## Эволюция инициализаций

Как было показано выше, в разделе экспериментов, даже небольшая случайная подвыборка может выдать неплохие результаты за короткое время (рис. 19 (слева)). Идея предлагаемого метода очень проста: добавить мониторинг по текущим наилучшим результатам запусков и на очередном перезапуске алгоритма брать в качестве инициализации параметров  $F, B, A$  полученные результаты на наилучшем запуске (предполагается, что запуски будут проводиться также на небольших подвыборках). При невырожденных размерах выборок такая инициализация не будет склонна к переобучению и позволит облегчать работу метода. Благодаря этому за несколько "недорогих" запусков EM-алгоритма можно провести такой процесс дообучения и повысить качество результатов и облегчить сходимость метода. Данная идея справедлива и для случая подготовки "хорошей" инициализации для запуска на полноценной выборке - в любом случае это облегчит работу программе.

Идею можно развить так: необходимо предварительно разбить выборку на независимые подвыборки, на каждой из них запустить EM-алгоритм. Далее усреднить пиксели полученных прогнозов  $F, B, A$  и подать в качестве инициализации генеральному запуску на всей выборке. Но у нас также есть сформированные апостериорные распределения на положение маски для каждого изображения. Поэтому генеральный запуск можно начать с M-шага, чтобы практически бесплатно использовать сформированные апостериоры. Дальнейшие итерации будут обобщать "опыт" предшествующих подзапусков. Это должно улучшить качество модели, так как мы сами помогаем EM-алгоритму быстрее и точнее сходиться.

## Простое ансамблирование

Ещё более простая мысль: разбить данные на независимые подвыборки, запустить на каждой из них EM-алгоритм, а затем взять от полученных прогнозов взвешенную сумму: каждый вес будет пропорционален отмасштабированной под размер подвыборки величине ELBO, полученной на соответствующем запуске. Это также позволит значительно сэкономить время, так как такую программу можно, опять же, распараллеливать, а после всех запусков просто собрать итоги и проаггрегировать, что будет очень быстрым способом.

## Промежуточный вариант между EM и hard EM

Как вариант повышения точности относительно hard EM-алгоритма и скорости работы относительно обычного EM, можно предложить после E-шага брать не одно положение маски  $d_k$ , а выбирать некоторый топ по убыванию апостериорного распределения, перенормировывать их и давать в M-шаг полученные новые апостериоры. Это позволит повысить гибкость hard метода, но снижает его скорость, конечно. Однако все программные аспекты можно будет регулировать размером отобранного топа.

Отметим, что указанные идеи могут задействовать в себе и hard EM алгоритм, для дополнительного повышения эффективности программы. В данном случае важно соблюдение условия, что используемые подвыборки не слишком малы для работы метода - чтобы избежать проблем с маленькими выборками, описанными в разделе экспериментов. Также для некоторых методов допустима их комбинация для усиления их положительных эффектов.

## 7 Выводы

В данной работе были изучены основные свойства EM-алгоритма и его модификации - hard EM- в задаче обработки зашумлённых изображений.

В ходе экспериментов было получено, что эти методы на адекватных наборах данных способны выдавать довольно точные результаты, предоставляя много полезной информации из исходных данных. В сравнении с hard EM-алгоритмом обычный EM более стабильный и лучше работает на небольших выборках, но зато hard модификация способна значительно превзойти обычный метод по времени работы, показывая хорошие результаты на средних и больших выборках данных.

Также были предложены методы для развития EM-алгоритма для данной задачи, благодаря чему возможно улучшить его показатели по качеству работы и скорости обработки.

Эти методы демонстрируют свои преимущества как в точности, так и в скорости работы. Опираясь на серьёзный математический аппарат, EM-алгоритм, подкреплённый, как с теоретической точки зрения, так и в плане удобства и эффективности своей реализации, является очень хорошим инструментом в машинном обучении.

## 8 Список литературы:

1. Лекции и семинары по курсу "Байесовские методы машинного обучения"