Отчёт по практической домашней работе №2 по предмету «Байесовские методы машинного обучения»

Артём Чубов

Ноябрь 2024

Теория:

- **E - шаг:**

  Найдем апостериорное распределение на координаты лица для конкретной фотографии:

$$p(d_k \mid X_k, \theta, \mathcal{A}) = \frac{p(d_k, X_k, \theta, \mathcal{A})}{p(X_k, \theta, \mathcal{A})} = \frac{p(X_k \mid d_k, \theta) p(d_k \mid \mathcal{A})}{\sum\limits_{d_k} p(X_k \mid d_k, \theta) p(d_k \mid \mathcal{A})} =$$

$$= \frac{\prod\limits_{i,j} \mathcal{N}(X_k(i,j) \mid F(i-d_k^h, j-d_k^w), s^2)^{[(i,j)\in faceArea(d_k)]} \mathcal{N}(X_k(i,j) \mid B(i,j), s^2)^{[(i,j)\notin faceArea(d_k)]} \mathcal{A}(d_k^h, d_k^w)}{\sum\limits_{d_k^h, d_k^w} \prod\limits_{i,j} \mathcal{N}(X_k(i,j) \mid F(i-d_k^h, j-d_k^w), s^2)^{[(i,j)\in faceArea(d_k)]} \mathcal{N}(X_k(i,j) \mid B(i,j), s^2)^{[(i,j)\notin faceArea(d_k)]} \mathcal{A}(d_k^h, d_k^w)}$$

  Тогда распределение на координаты лица на всех изображениях задаётся произведением отдельных апостериорных распределений:

$$q(d) = \prod_k p(d_k \mid X_k, \theta, \mathcal{A})$$

- **M-шаг в стандартном EM - алгоритме:**

$$\mathrm{Q}(\boldsymbol{\theta}, \boldsymbol{\mathcal{A}}) = \mathbb{E}_{q(d)} \left[\log p(\boldsymbol{X}, \boldsymbol{d} \mid \boldsymbol{\theta}, \boldsymbol{\mathcal{A}})\right] = \mathbb{E}_{q(d)} \left[\log \left(\prod_k p(X_k \mid d_k, \theta) p(d_k \mid \mathcal{A})\right)\right] = \sum_k \mathbb{E}_{q(d)} \left[\log\left(p(X_k \mid d_k, \theta)\right]+\right.$$

$$+ \sum_k \mathbb{E}_{q(d)} \left[\log\left(p(d_k \mid \mathcal{A})\right] =\right.$$

$$= \sum_k \mathbb{E}_{q(d)} \left[\sum_{i,j} \log\left(\mathcal{N}(X_k(i,j) \mid F(i-d_k^h, j-d_k^w), s^2)^{[(i,j)\in faceArea(d_k)]} \mathcal{N}(X_k(i,j) \mid B(i,j), s^2)^{[(i,j)\notin faceArea(d_k)]}\right)\right]+$$

$$+\sum_k \mathbb{E}_{q(d)}\left[\mathcal{A}(d_k^h, d_k^w)\right] = \sum_k \sum_{i,j} \mathbb{E}_{q(d)}\left[(i,j) \in faceArea(d_k)\right] \log\left(\mathcal{N}(X_k(i,j) \mid F(i-d_k^h, j-d_k^w), s^2)\right] +$$

$$+ \quad \sum_k \sum_{i,j} \mathbb{E}_{q(d)}\left[[(i,j) \notin faceArea(d_k)] \log\left(\mathcal{N}(X_k(i,j) \mid B(i,j), s^2)\right)\right] \; + \; \sum_k \mathbb{E}_{q(d)}\left[\mathcal{A}(d_k^h, d_k^w)\right]$$

Рассмотрим логарифмы гауссиан, фигурирующих в выражении:

$$\log\left[\mathcal{N}(X_k(i,j) \mid F(i-d_k^h, j-d_k^w), s^2)\right] = -\frac{1}{2s^2}\left(X_k(i,j) - F(i-d_k^h, j-d_k^w)\right)^2 - \log s - \frac{1}{2}\log(2\pi)$$

$$\log\left[\mathcal{N}(X_k(i,j) \mid B(i,j), s^2)\right] = -\frac{1}{2s^2}\left(X_k(i,j) - B(i,j)\right)^2 - \log s - \frac{1}{2}\log(2\pi)$$

В плотностях фигурируют константы, которые, тем не менее, не повлияют на задачу максимизации, поэтому ими можно будет пренебречь.

Перепишем множество faceArea(d_k) относительно d_k:

$$\begin{cases} d_k^h \in [i-h+1, i], \\ d_k^w \in [j-w+1, j] \end{cases}$$

Далее, после взятия математического ожидания для фиксированных i, j, будем обозначать это множество fa(i,j)

Подставим эти представления в функционал:

$$Q(\boldsymbol{\theta}, \boldsymbol{\mathcal{A}}) = \sum_k \sum_{i,j} \mathbb{E}_q\left[[(i,j) \in \mathrm{fa}(d_k)] \cdot \left(-\frac{1}{2s^2}\left(X_k(i,j) - F(i-d_k^h, j-d_k^w)\right)^2 - \frac{1}{2}\log s^2\right) +\right.$$

$$\left. +[(i,j) \notin \mathrm{fa}(d_k)] \cdot \left(-\frac{1}{2s^2}\left(X_k(i,j) - B(i,j)\right)^2 - \frac{1}{2}\log s^2\right)\right] + \sum_k \mathbb{E}_q\left[\ln A(d_k^h, d_k^w)\right] + const =$$

$$= -\frac{1}{2s^2}\sum_k \sum_{i,j} \mathbb{E}_q\left[[(i,j) \in \mathrm{fa}(d_k)]\left(X_k(i,j) - F(i-d_k^h, j-d_k^w)\right)^2 + [(i,j) \notin \mathrm{fa}(d_k)]\left(X_k(i,j) - B(i,j)\right)^2\right] -$$

$$-\frac{1}{2}\sum_k \sum_{i,j} \log s^2 + \sum_k \mathbb{E}_q\left[\ln A(d_k^h, d_k^w)\right] + const =$$

$$= -\frac{1}{2s^2}\left(\sum_k \sum_{i,j} \sum_{a=i-h+1}^{i} \sum_{b=j-w+1}^{j} P\{d_k^h = a, d_k^w = b\}\left(X_k(i,j) - F(i-a, j-b)\right)^2 +\right.$$

$$\left. + \sum_k \sum_{i,j} P\{d_k \notin \mathrm{fa}(i,j)\}\left(X_k(i,j) - B(i,j)\right)^2\right) - \frac{1}{2}\sum_k \sum_{i,j} \log s^2 + \sum_k \sum_{a=0}^{H-h} \sum_{b=0}^{W-w} P\{d_k^h = a, d_k^w = b\} \log A(a,b) + const =$$

Для упрощения индексации и последующего дифференцирования, осуществим замену индексов в первом слагаемом:

$$u = i - a, \;\; v = j - b$$

$$u \in [i - H + h, i], \;\; v \in [j - W + w, j]$$

$$= -\frac{1}{2s^2}\Bigg(\sum_k \sum_{i,j} \sum_{u=i-H+h}^{i} \sum_{v=j-W+w}^{j} P\{d_k^h = i-u, d_k^w = j-v\}\big(X_k(i,j) - F(u,v)\big)^2 +$$

$$+\sum_k \sum_{i,j} P\{d_k \notin \mathrm{fa}(i,j)\}\big(X_k(i,j) - B(i,j)\big)^2\Bigg) - \frac{1}{2}\sum_k \sum_{i,j} \log s^2 + \sum_k \sum_{a=0}^{H-h} \sum_{b=0}^{W-w} P\{d_k^h = a, d_k^w = b\} \log A(a,b) + const$$

Помимо этого при оптимизации стоит учесть, что $\sum_{i,j} \mathcal{A}(i,j) = 1$,поэтому воспользуемся методом множителей Лагранжа и запишем финальный вид функционала:

$$Q\left(\boldsymbol{\theta}, \boldsymbol{\mathcal{A}}, \boldsymbol{\lambda}\right) =$$

$$= -\frac{1}{2s^2}\Bigg(\sum_k \sum_{i,j} \sum_{u=i-H+h}^{i} \sum_{v=j-W+w}^{j} P\{d_k^h = i-u, d_k^w = j-v\} P\{d_k^h = i-u, d_k^w = j-v\}\big(X_k(i,j) - F(u,v)\big)^2 +$$

$$+\sum_k \sum_{i,j} P\{d_k \notin \mathrm{fa}(i,j)\}\big(X_k(i,j) - B(i,j)\big)^2\Bigg) - \frac{1}{2}\sum_k \sum_{i,j} \log s^2 + \sum_k \sum_{a=0}^{H-h} \sum_{b=0}^{W-w} P\{d_k^h = a, d_k^w = b\} \log A(a,b) +$$

$$+ \lambda\left(\sum_{i,j} \mathcal{A}(i,j) - 1\right) + const.$$

Теперь займёмся оптимизацией функционала по параметрам, выписав условия первого порядка:

- **Точечная оценка для A**:

$$\frac{\partial Q}{\partial \mathcal{A}(i,j)} = \sum_k P\{d_k^h = i, d_k^w = j\}\frac{1}{\mathcal{A}(i,j)} + \lambda = 0 \Rightarrow \mathcal{A}(i,j) = \frac{\sum_k P\{d_k^h = i, d_k^w = j\}}{-\lambda}$$

$$\frac{\partial Q}{\partial \lambda} = \sum_{i,j} \mathcal{A}(i,j) - 1 = 0 \Rightarrow \frac{\sum_k \sum_{i,j} P\{d_k^h = i, d_k^w = j\}}{-\lambda} = \frac{K}{-\lambda} = 1 \Rightarrow \lambda = -K$$

Тогда:

$$\boldsymbol{A^*(i,j)} = \frac{\sum_k P\{d_k^h = i, d_k^w = j\}}{K}$$

- **Точечная оценка для F**:

$$\frac{\partial Q}{\partial F(u,v)} = \sum_k \sum_{i,j} \frac{1}{s^2} P\{d_k^h = i-u, d_k^w = j-v\}\big(X_k(i,j) - F(u,v)\big) = 0 \quad / \cdot s^2$$

$$\sum_k \sum_{i,j} P\{d_k^h = i-u, d_k^w = j-v\} X_k(i,j) - \sum_k \sum_{i,j} P\{d_k^h = i-u, d_k^w = j-v\} F(u,v) = 0$$

$$\boldsymbol{F^*(u,v)} = \frac{\sum_k \sum_{i,j} P\{d_k^h = i-u, d_k^w = j-v\} X_k(i,j)}{\sum_k \sum_{i,j} P\{d_k^h = i-u, d_k^w = j-v\}} = \frac{\sum_k \sum_{i,j} P\{d_k^h = i-u, d_k^w = j-v\} X_k(i,j)}{K}$$

– **Точечная оценка для B**:

$$\frac{\partial Q}{\partial B(i,j)} = \sum_k P\{d_k \notin \text{fa}(i,j)\}\big(X_k(i,j) - B(i,j)\big) = 0$$

$$\boldsymbol{B^*(i,j)} = \frac{\sum_k P\{d_k \notin \text{fa}(i,j)\} X_k(i,j)}{\sum_k P\{d_k \notin \text{fa}(i,j)\}}$$

– **Точечная оценка для s**^{2}:

$$\text{Пусть } C(X,F,B) = \sum_k \sum_{i,j} \Bigg[\sum_{u=i-H+h}^{i} \sum_{v=j-W+w}^{j} P\{d_k^h = i-u, d_k^w = j-v\}\big(X_k(i,j) - F(u,v)\big)^2 + \\ + P\{d_k \notin \text{fa}(i,j)\}\big(X_k(i,j) - B(i,j)\big)^2 \Bigg]$$

$$\frac{\partial Q}{\partial s^2} = \frac{1}{2s^4} \cdot C(X, \boldsymbol{F^*}, \boldsymbol{B^*}) - \frac{1}{2}\sum_k \sum_{i,j} \frac{1}{s^2} = 0$$

$$C(X, \boldsymbol{F^*}, \boldsymbol{B^*}) - \sum_k \sum_{i,j} s^2 = 0 \quad \Rightarrow \quad s^2 = \frac{C(X, \boldsymbol{F^*}, \boldsymbol{B^*})}{\sum_k \sum_{i,j} 1} = \frac{C(X, \boldsymbol{F^*}, \boldsymbol{B^*})}{H \cdot W \cdot K}$$

M-шаг для hard EM:

По сути, в случае hard EM мы будем иметь дело с вырожденным распределением q, где только для одной точки вероятность ненулевая(и равная 1), поэтому в имеющихся формулах достаточно заменить вероятности на индикаторные функции. E-шаг будет отличаться лишь взятием аргмаксимума от распределения q, формулы же для M-шага представлены ниже:

$$\text{Пусть } (d_k^{h*}, d_k^{w*}) = \underset{d_k}{argmax}(q(d_k))$$

Тогда:

$$Q\left(\boldsymbol{\theta}, \boldsymbol{\mathcal{A}}, \boldsymbol{\lambda}\right) = -\frac{1}{2s^2}\left(\sum_k \sum_{i,j} [d_k^* \in \text{fa}(i,j)]\big(X_k(i,j) - F(u,v)\big)^2 + [d_k^* \notin \text{fa}(i,j)]\big(X_k(i,j) - B(i,j)\big)^2 \right) - \\ - \frac{1}{2}\sum_k \sum_{i,j} \log s^2 + \sum_k \sum_{a=0}^{H-h} \sum_{b=0}^{W-w} [a = d_k^{h*}, b = d_k^{w*}] \log A(a,b) + \lambda \left(\sum_{i,j} \mathcal{A}(i,j) - 1 \right).$$

– **Точечная оценка для A**:

$$\boldsymbol{A^*(i,j)} = \frac{\sum_k [d_k^{h*} = i, d_k^{w*} = j]}{K}$$

– **Точечная оценка для F**:

$$\boldsymbol{F^*(u,v)} = \frac{\sum_k \sum_{i,j} \left[d_k^{h*} = i-u, d_k^{w*} = j-v\right] X_k(i,j)}{\sum_k \sum_{i,j} \left[d_k^{h*} = i-u, d_k^{w*} = j-v\right]} = \frac{\sum_k \sum_{i,j} \left[d_k^{h*} = i-u, d_k^{w*} = j-v\right] X_k(i,j)}{K}$$

– **Точечная оценка для B**:

$$\boldsymbol{B^*(i,j)} = \frac{\sum_k [d_k^* \notin \mathrm{fa}(i,j)] X_k(i,j)}{\sum_k [d_k^* \notin \mathrm{fa}(i,j)]}$$

– **Точечная оценка для s**^{2}:

$$s^2 = \frac{\sum_k \sum_{i,j} \left([d_k^* \in \mathrm{fa}(i,j)] \left(X_k(i,j) - F(u,v)\right)^2 + [d_k^* \notin \mathrm{fa}(i,j)] \left(X_k(i,j) - B(i,j)\right)^2 \right)}{H \cdot W \cdot K}$$

- **Вывод** $\mathcal{L}(q, \theta, \mathcal{A})$:

$$\begin{aligned}
&\mathcal{L}\left(q, \theta, \mathcal{A}\right) = \mathbb{E}_{q(d)}\left[\log p(\boldsymbol{X}, \boldsymbol{d} \mid \boldsymbol{\theta}, \boldsymbol{\mathcal{A}})\right] - \mathbb{E}_{q(d)}\left[\log(q)\right] = \\
&= -\frac{1}{2s^2}\Bigg(\sum_k \sum_{i,j} \sum_{u=0}^{h-1} \sum_{v=0}^{w-1} P\{d_k^h = i-u, d_k^w = j-v\}\left(X_k(i,j) - F(u,v)\right)^2 + \\
&+ \sum_k \sum_{i,j} P\{d_k \notin \mathrm{fa}(i,j)\}\left(X_k(i,j) - B(i,j)\right)^2 \Bigg) - \frac{1}{2} \sum_k \sum_{i,j} \log s^2 + \\
&+ \sum_k \sum_{a=0}^{H-h} \sum_{b=0}^{W-w} P\{d_k^h = a, d_k^w = b\} \log(A(a,b)) - \sum_k \sum_{a=0}^{H-h} \sum_{b=0}^{W-w} P\{d_k^h = a, d_k^w = b\} \log(P\{d_k^h = a, d_k^w = b\}) - \\
&- \sum_k \sum_{i,j} \log(2\pi) = \boldsymbol{-\frac{1}{2s^2}\Bigg(\sum_k \sum_{i,j} \sum_{u=0}^{h-1} \sum_{v=0}^{w-1} P\{d_k^h = i-u, d_k^w = j-v\}\left(X_k(i,j) - F(u,v)\right)^2} + \\
&+ \boldsymbol{\sum_k \sum_{i,j} P\{d_k \notin \mathrm{fa}(i,j)\}\left(X_k(i,j) - B(i,j)\right)^2 \Bigg) - H \cdot W \cdot K \log s} + \\
&+ \boldsymbol{\sum_k \sum_{a=0}^{H-h} \sum_{b=0}^{W-w} P\{d_k^h = a, d_k^w = b\} \log(A(a,b))} - \\
&- \boldsymbol{\sum_k \sum_{a=0}^{H-h} \sum_{b=0}^{W-w} P\{d_k^h = a, d_k^w = b\} \log(P\{d_k^h = a, d_k^w = b\}) - \frac{1}{2} \cdot H \cdot W \cdot K \log(2\pi)}
\end{aligned}$$

Анализ результатов:

Для проведения экспериментов использовались картинки малого разрешения (изображение 16×16, фон 20×30), чтобы было проще отлаживать код:

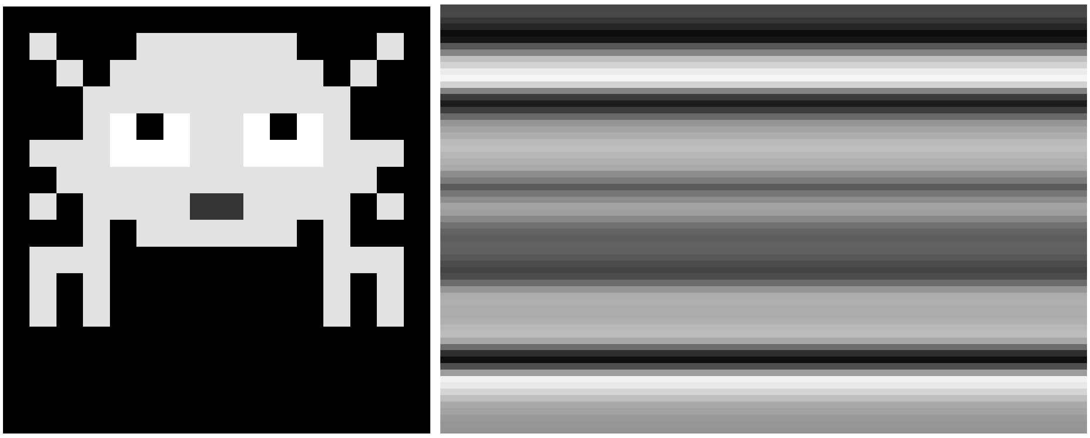

Рис. 1: Используемое изображение и его фон

- **Сравнение инициализаций:**

  В качестве инициализаций использовались следующие подходы:

  – Нулевые приближения, представленные чёрным фоном для всех матриц и малым значением для s.

  – Равномерное распределение для A, приближения пикселями максимальной яркости 255 и аналогичные начальные значения для других параметров.

  – Случайная инициализация для всех параметров, где матрицы заполнялись равномерно распределёнными числами в диапазоне от 0 до 255, s $\sim$ $\mathbf{exp(10)}$.

  – Инициализация, в которой матрицы F и B были получены в результате обработки гауссовским фильтром случайных матриц из выборки, где фильтр имел среднее 0 и стандартное отклонение, усредненное по всем имеющимся картинкам с поправкой на размеры F и B.

  Ниже приведены результаты работы алгоритма на каждой из инициализаций, а также время работы и финальное значение нижней оценки правдоподобия:

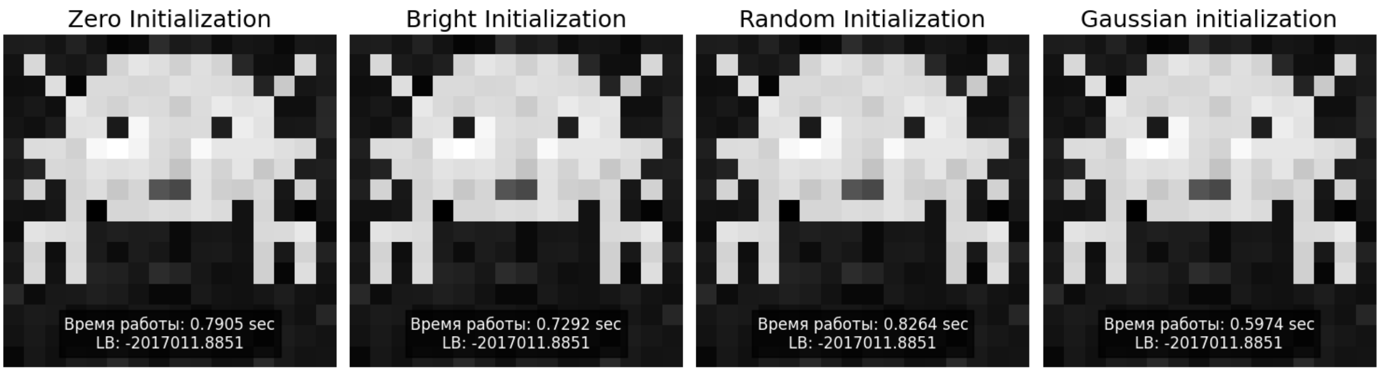

Рис. 2: Результаты работы алгоритма на разных инициализациях

Значение нижней вариационной оценки оказывается одинаковым у всех подходов с точностью до 4-го знака после запятой, что говорит об успешной сходимости каждого из них. Интересно, что инициализации самыми яркими и самыми темными пикселями приводят к одному и тому же времени работы, случайная инициализация ведет к чуть дольшей сходимости и самый быстрый результат алгоритм демонстрирует на инициализации с гауссовским сглаживанием, что логично: в постановке задачи пиксели зашумленной картинки имеют нормальное распределение, и некоторое "сглаживание" их значений приводит к лучшей сходимости алгоритма. Рассмотрим динамику сходимости алгоритма с разными инициализациями:

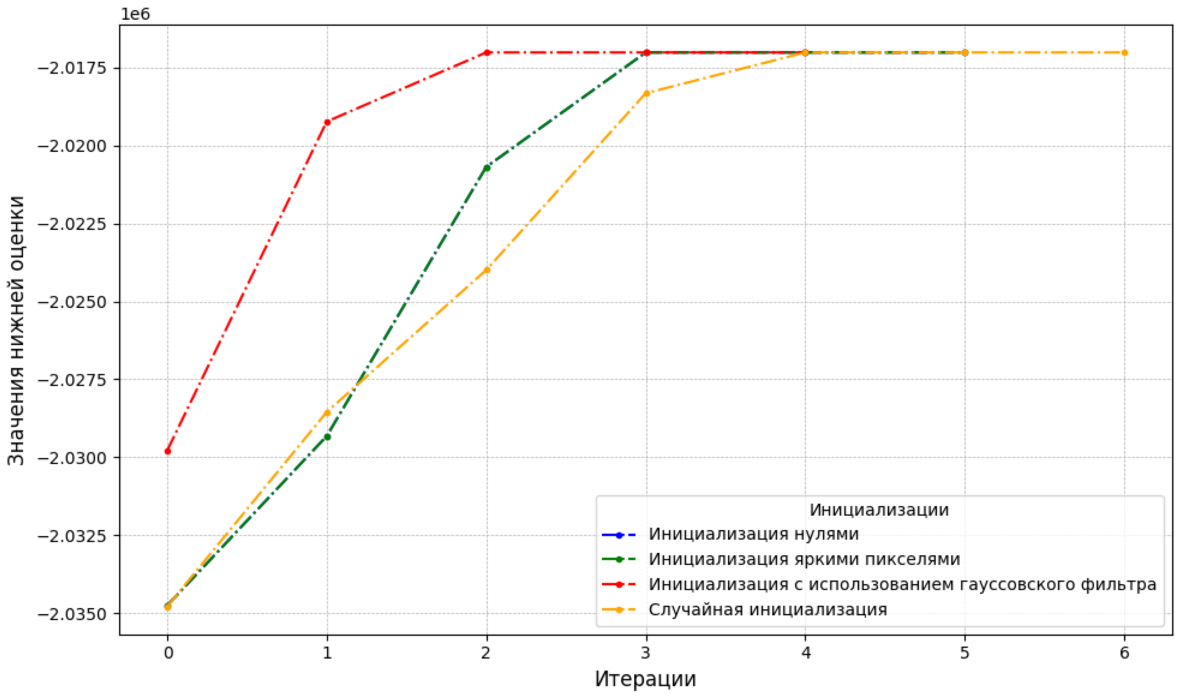

Рис. 3: Сравнение сходимостей подходов

По **рисунку 3** видно, что быстрее всех сходится алгоритм с инициализацией гауссовской свёрткой, что хорошо бьётся с общей интуицией данного подхода. Инициализации очень яркими и очень тёмными пикселями одинаково влияют на сходимость алгоритма - с ними он ведёт себя идентично. Впрочем, это вполне может зависеть и от характера изображения и тех паттернов, что наблюдаются в нём. Алгоритму в условиях случайной инициализации требуется больше всего итераций для выхода на плато - это тоже можно объяснить тем, что с такой инициализацией мы, по сути, не закладываем никакого априорного знания о данных.

- **Сравнение результатов при разных размерах выборок и стандартных отклонениях шума:**

  Для проведения данного эксперимента были сгенерированы выборки размеров 100, 500 и 1500 наблюдений; значения стандартного отклонения - 200, 400, 1000.

  На **рисунке 4** видно, что при меньших значениях дисперсии шума алгоритму легче справиться с задачей расшумления в том смысле, что для получения визуально различимого результата достаточно даже 100 наблюдений, что весьма немного. При стандартном отклонении в размере 1000 алгоритму не хватает и 1500 наблюдений для качественного расшумления. Это можно объяснить и разбросом значений шума и тем, что для тестов используются изображения малого разрешения сопоставимого масштаба, а значит, количество положений чудика на фоне не так велико, что снижает разнообразие имеющейся выборки.

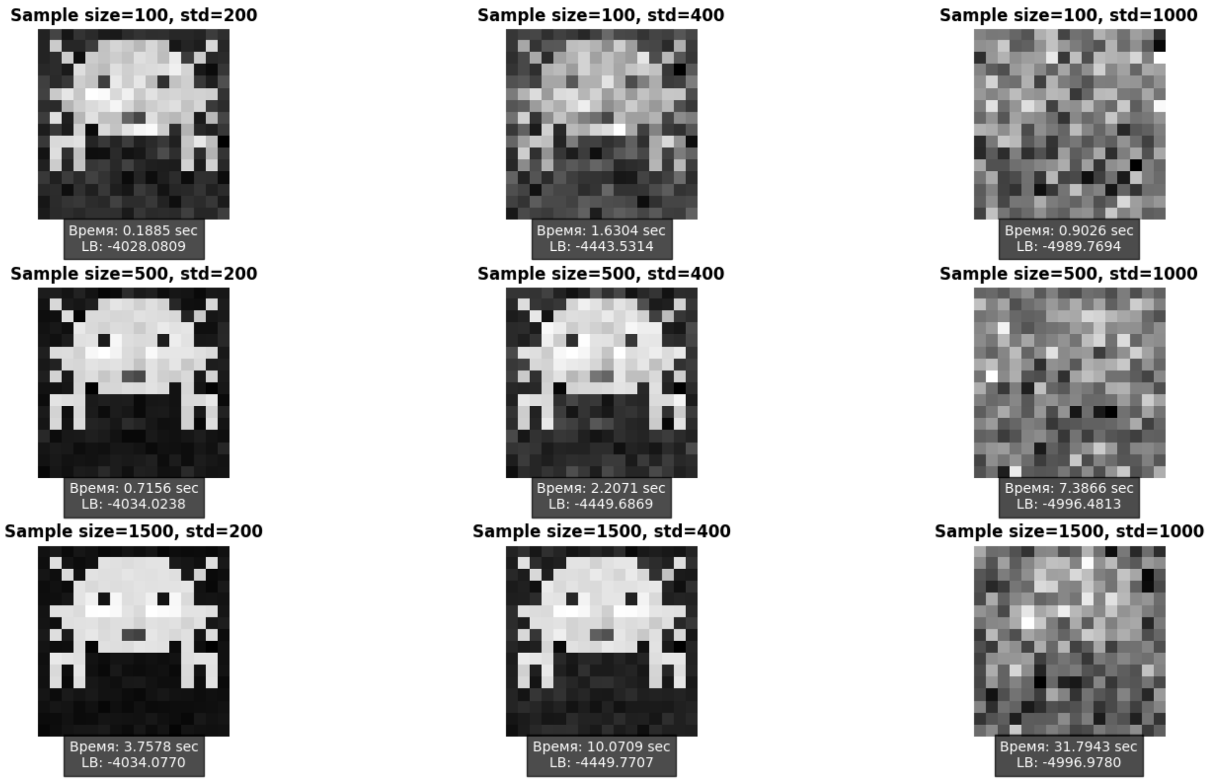

Рис. 4: Сравнение результатов алгоритма в зависимости от размера выборки и стандартного отклонения шума

Теперь посмотрим на сходимость алгоритма в разных условиях:

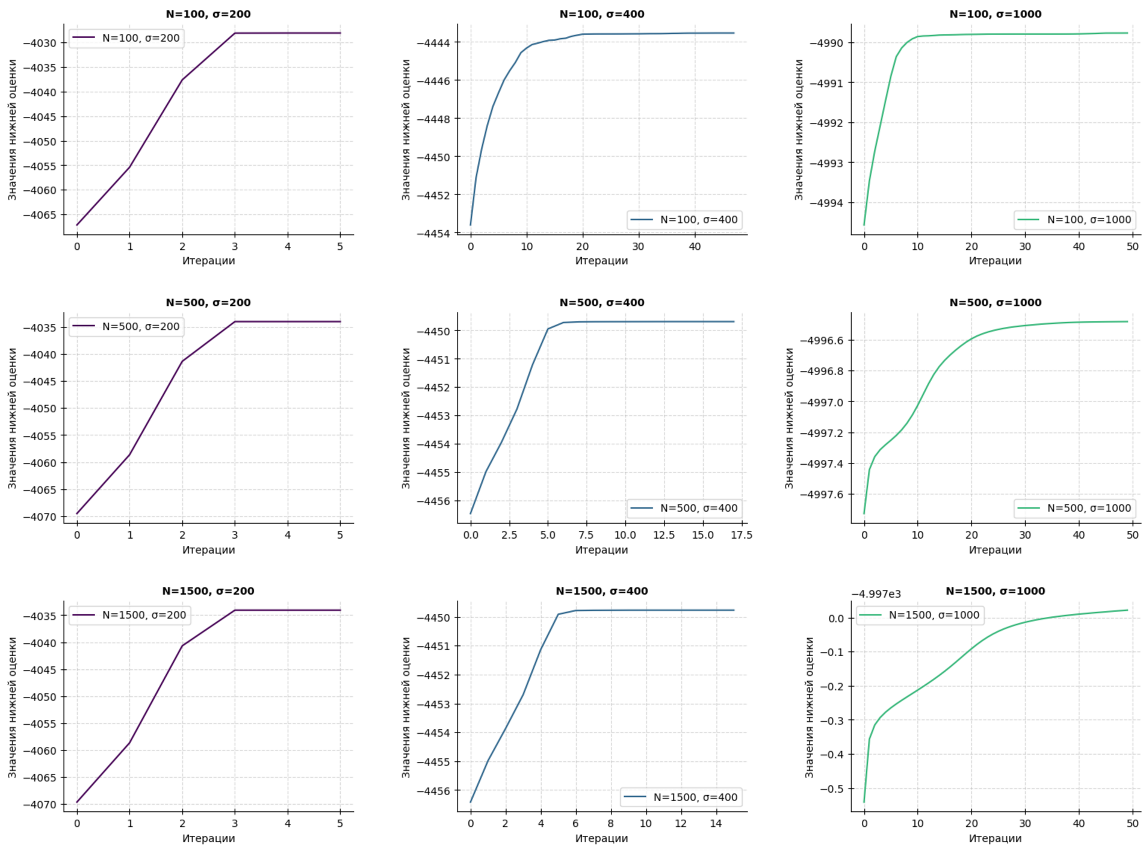

Рис. 5: Сходимость алгоритма в зависимости от размера выборки и дисперсии шума

Из графиков видно, что при малой дисперсии шума сходимость алгоритма на большой и малой выборках практически неотличимы, но изображение при 100 наблюдениях не такое четкое, вероятно, из-за меньшей вариативности сгенерированных данных. При росте дисперсии, алгоритму на 100 картинках требуется больше итераций, чтобы выйти на плато в терминах значения нижней оценки на правдоподобие, чем на 500 или 1500. Вместе с тем, что логично, время работы алгоритма также растёт. В случае с самым высоким значением стандартного отклонения, результаты на всех выборках примерно одинаковы: алгоритм отрабатывает практически все итерации и плохо справляется с расшумлением изображения. Любопытным тут кажется тот факт, что при большем разрешении изображения и фона алгоритм более устойчив к высокой дисперсии шума, что может быть связано с большей гетерогенностью данных в таких условиях.

Посмотрим на динамику изменения матриц F и B в зависимости от значений стандартного отклонения:

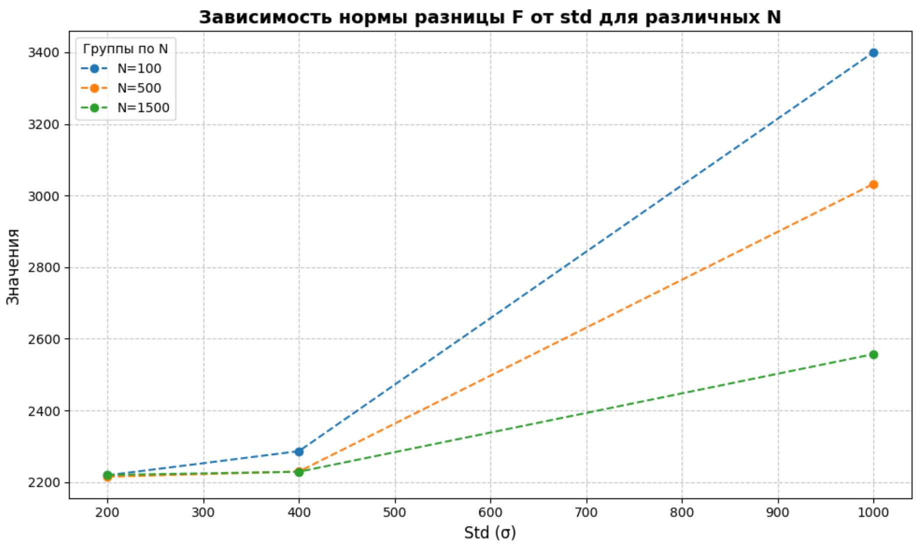

Рис. 6: Динамика нормы разности F

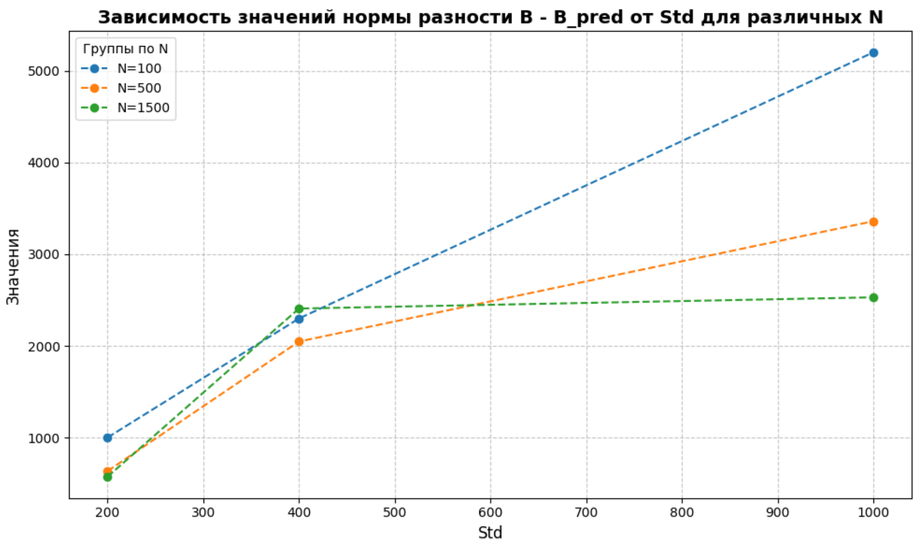

Рис. 7: Динамика нормы разности B

Результаты довольно разумные: с ростом дисперсии алгоритм сильнее ошибается, при этом для матрицы F кол-во наблюдений позитивно влияет на точность оценки. В случае с матрицей B также видно, что точность прогноза отрицательно зависит от значения стандартного отклонения, однако, рост количества наблюдений не всегда положительно сказывается на точности. Возможно, это связано с тем, что при более точной оценке матрицы F, матрица B становится более зашумлённой.

- **Сравнением обычного EM и hard EM:**

  Для сравнения подходов, посмотрим на результаты модификации на различных инициализациях и сравним с базовым подходом:

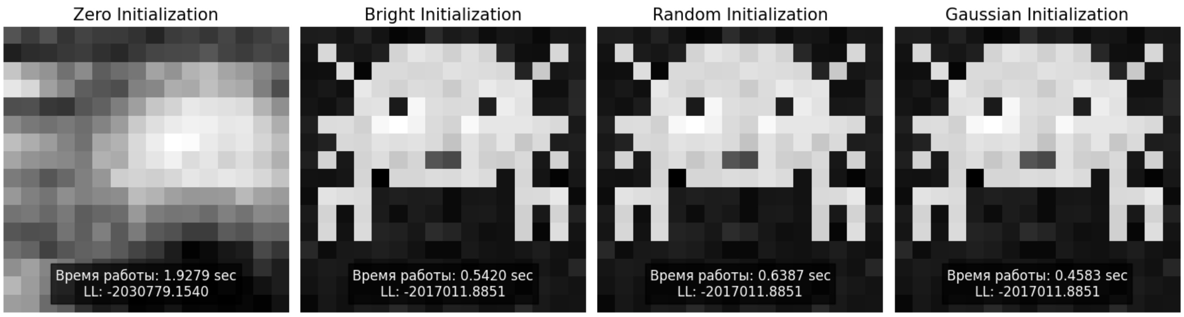

Рис. 8: Результаты работы EM с MAP

Визуальный анализ наводит на мысли, что hard EM не дружит с нулевыми инициализациями, но хорошо себя показывает в иных условиях, при этом демонстрируя улучшение во времени работы в 1.5 - 2 раза. На изображениях большего размера разница более заметная - там MAP - оценки позволяют получить многократное ускорение времени работы. Такой прирост в скорости можно объяснить тем, что модификация намного проще с точки зрения вычислений и, по сути, убирает необходимость взятия математического ожидания на этапе M-шага.

Теперь посмотрим на сходимость hard EM:

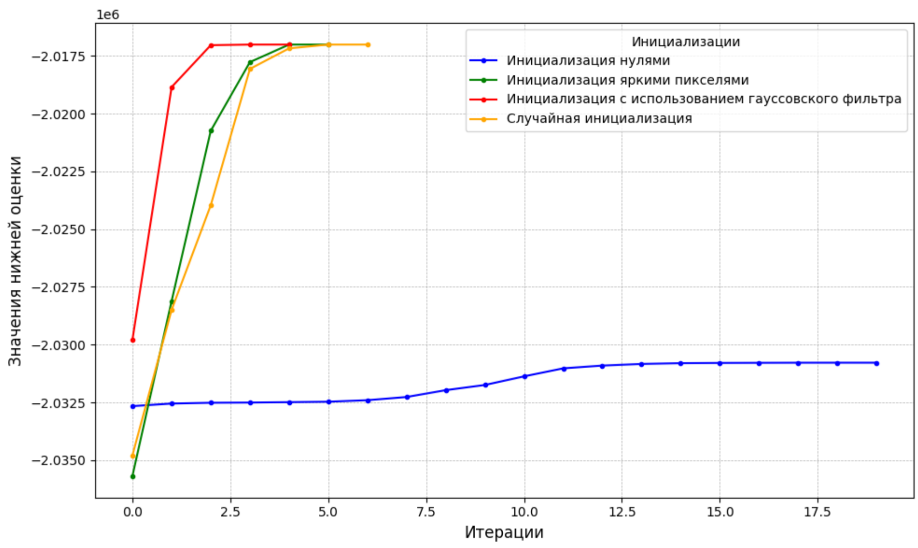

Рис. 9: Динамика изменения нижней оценки правдоподобия hard EM

По графику видно, что всё также лучше всех себя показывает гауссовская инициализация, затем идут инициализация яркими пикселями и случайная. При нулевой инициализации, нижняя оценка практически не возрастает с ростом итераций и, как следствие, не обеспечивает сходимость алгоритма. Видимо, вырожденное распределение, получаемое на Е-шаге, не позволяет алгоритму "прийти" из стартового приближения в оптимум из-за большого числа нулевых значений параметров.

- **Применение алгоритма к зашумлённым снимкам преступника:**

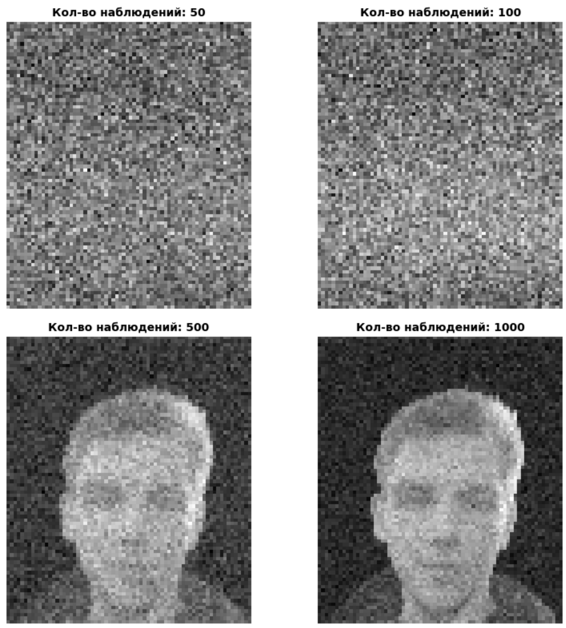

Рис. 10: Результаты работы алгоритма на зашумлённых фото преступника

Примерно с 300 наблюдений при большом желании можно разглядеть лицо Дениса Ракитина, победа. На 1000 наблюдений сомнений не остаётся и вовсе.

Посмотрим на фон:

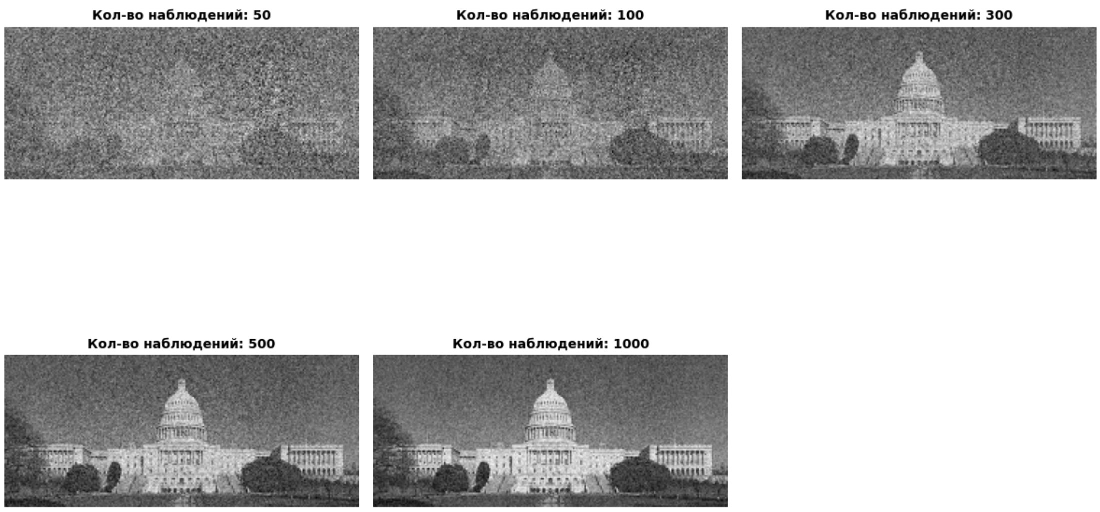

Рис. 11: Фон зашумлённого фото преступника

Ну, кажется, преступник сделал селфи в Лондоне.

- Предложение для модификации алгоритма: Довольно логичной кажется идея не брать математических ожиданий на М - шаге, а сэмплировать некоторое количество латентных переменных(координат углов изображения), а затем, подставив их в логарифм полного правдоподобия, надеяться на попадание в оптимум после оптимизации:

  После Е-шага:

  $$z^i \sim p(z \mid x, \theta)$$

  М-шаг:

  $$\log p(x, z \mid \theta) \to \max_{\theta}$$

  Для сэмплированных значений $z_1^i, \ldots, z_n^i$ на M-шаге:

  $$\frac{1}{n} \sum_{j=1}^{n} \log p(x_i, z_j^{(i)} \mid \theta) \to \max_{\theta}$$