Задание 2. EM алгоритм для детектива

Курс: Байесовские методы в машинном обучении, осень 2021

1 Введение

Леди Маргалотта обратилась к известному детективу Нику Картеру с просьбой расследовать похищение ее любимого песика. Даже огромная страховка не покроет горя хозяйки... Во время ее поездки в Лондон незнакомец выкрал любимца леди Маргалотты. Камеры видеонаблюдения, зафиксировавшие вора оказались плохими помощниками, т.к. подверглись воздействию направленного электро-магнитного постановщика помех. Тем не менее, это единственная зацепка. Под подозрением группа Байесовских методов из России.

Нику Картеру необходимо обработать снимки и восстановить лицо вора. Известно, что на разных фотографиях оно расположено в случайных координатах на неподвижном фоне. К счастью, видеолекции по курсу БММО уже выложены в сеть и пройдя его Ник Картер смог решить задачу и, с помощью EM-алгоритма, установить личность вора. Его ждал сюрприз...

Помогите детективу изобличить преступника. Зашумленные фотографии поступают из лаборатории порциями, которые будут выкладываться каждые 2-3 дня. Пример зашумленной фотографии показан на рисунке 1. Помните, что чем быстрее обнаружен преступник, тем проще его поймать! Определившему просьба оперативно сообщить об этом Нику (о том, как можно с ним связаться, читайте в разделе «Оформление задания»). Первые трое сообщивших правильный ответ получат благодарность от Ника и небольшие призы.

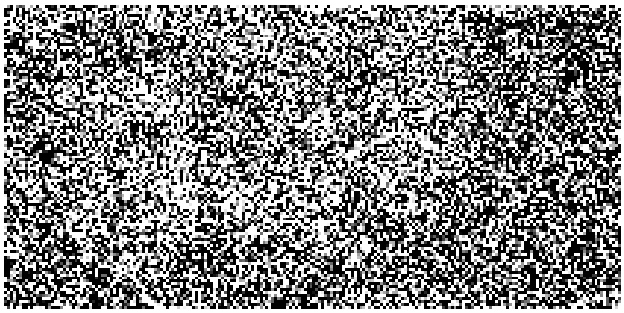

Рис. 1: Пример зашумленного изображения с камер видеонаблюдения.

2 Описание модели

Дана выборка $\boldsymbol{X} = \{\boldsymbol{X}_k\}_{k=1}^K$ сильно зашумленных черно-белых изображений размера $H \times W$ пикселей. Каждое из этих изображений содержит один и тот же неподвижный фон и лицо преступника в неизвестных координатах, при этом лицо попадает в любое изображение целиком. Будем считать, что изоражение лица имеет прямоугольную форму размера $h \times w$ пикселей. Значения h, w в выданных данных указаны в описании задания в anytask. Макет изображения показан на рисунке 2 .

Введем следующие обозначения:

- $\boldsymbol{X}_k(i,j)$ — пиксель k-ого изображения;
- $\boldsymbol{B} \in \mathbb{R}^{H \times W}$ — изображение чистого фона без лица преступника, $\boldsymbol{B}(i,j)$ — пиксель этого изображения;
- $\boldsymbol{F} \in \mathbb{R}^{h \times w}$ — изображение лица преступника, $\boldsymbol{F}(i,j)$ — пиксель этого изображения;
- $\boldsymbol{d}_k = (d_k^h, d_k^w)$ — координаты верхнего левого угла изображения лица на k-ом изображении (d_k^h — по вертикали, d_k^w — по горизонтали), $\boldsymbol{d} = (\boldsymbol{d}_1, \ldots, \boldsymbol{d}_K)$ — набор координат для всех изображений выборки.

Также будем считать шум на изображении независимым для каждого пикселя и принадлежащим нормальному распределению $\mathcal{N}(0, s^2)$, где s — стандартное отклонение. Таким образом для одного изображения имеем:

$$p(\boldsymbol{X}_k \mid \boldsymbol{d}_k, \boldsymbol{\theta}) = \prod_{ij} \begin{cases} \mathcal{N}(\boldsymbol{X}_k(i,j) \mid \boldsymbol{F}(i - d_k^h, j - d_k^w), s^2), & \text{если } (i,j) \in faceArea(\boldsymbol{d}_k) \\ \mathcal{N}(\boldsymbol{X}_k(i,j) \mid \boldsymbol{B}(i,j), s^2), & \text{иначе} \end{cases},$$

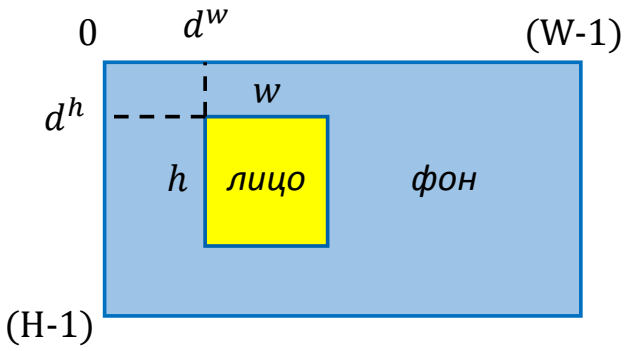

Рис. 2: Макет изображения с камеры наблюдения без шума.

где $\boldsymbol{\theta} = \{\boldsymbol{B}, \boldsymbol{F}, s^2\}$, $faceArea(\boldsymbol{d}_k) = \{(i,j) \mid d_k^h \le i \le d_k^h + h - 1, d_k^w \le j \le d_k^w + w - 1\}$.

Распределение на неизвестные координаты лица на изображении зададим общим для всех изображений с помощью матрицы параметров $\boldsymbol{A} \in \mathbb{R}^{H-h+1, W-w+1}$ следующим образом:

$$p(\boldsymbol{d}_k \mid \boldsymbol{A}) = \boldsymbol{A}(d_k^h, d_k^w), \qquad \sum_{ij} A(i,j) = 1,$$

где $\boldsymbol{A}(i,j)$ — элемент матрицы $\boldsymbol{A}$.

В итоге имеем следующую совместную вероятностную модель:

$$p(\boldsymbol{X}, \boldsymbol{d} \mid \boldsymbol{\theta}, \boldsymbol{A}) = \prod_k p(\boldsymbol{X}_k \mid \boldsymbol{d}_k, \boldsymbol{\theta}) p(\boldsymbol{d}_k \mid \boldsymbol{A}).$$

3 Формулировка задания

Требуется решить задачу

$$p(\boldsymbol{X} \mid \boldsymbol{\theta}, \boldsymbol{A}) \to \max_{\boldsymbol{\theta}, \boldsymbol{A}}.$$

Для этого предлагается воспользоваться EM-алгоритмом, то есть перейти к следующей задаче оптимизации нижней оценки на логарифм неполного правдоподобия:

$$\mathcal{L}(q, \boldsymbol{\theta}, \boldsymbol{A}) = \mathbb{E}_{q(\boldsymbol{d})} \log p(\boldsymbol{X}, \boldsymbol{d} \mid \boldsymbol{\theta}, \boldsymbol{A}) - \mathbb{E}_{q(\boldsymbol{d})} \log q(\boldsymbol{d}) \to \max_{q, \boldsymbol{\theta}, \boldsymbol{A}}$$

На E-шаге вычисляется оценка на апостериорное распределение на координаты лица на изображениях:

$$q(\boldsymbol{d}) = p(\boldsymbol{d} \mid \boldsymbol{X}, \boldsymbol{\theta}, \boldsymbol{A}) = \prod_k p(\boldsymbol{d}_k \mid \boldsymbol{X}_k, \boldsymbol{\theta}, \boldsymbol{A}),$$

а на M-шаге вычисляется точечная оценка на параметры $\boldsymbol{\theta}, \boldsymbol{A}$:

$$\mathbb{E}_{q(\boldsymbol{d})} \log p(\boldsymbol{X}, \boldsymbol{d} \mid \boldsymbol{\theta}, \boldsymbol{A}) \to \max_{\boldsymbol{\theta}, \boldsymbol{A}}.$$

Также далее будет рассматриваться упрощенный вариант EM-алгоритма, который называется hard EM. В нем после E шага берется не все апостериорное распределение на координаты лица на изображениях, а только MAP оценка на эти координаты (то есть после E шага $q(\boldsymbol{d})$ преобразовывают так, что для каждого изображения $\boldsymbol{X}_k$ оценка $q(\boldsymbol{d}_k)$ принимает значение 1 только в одной точке — точке аргмаксимума апостериорного распределения $p(\boldsymbol{d}_k \mid \boldsymbol{X}_k, \boldsymbol{\theta}, \boldsymbol{A})$).

При выполнении итераций EM алгоритма нужно следить за значением оптимизируемого функционала $\mathcal{L}(q, \boldsymbol{\theta}, \boldsymbol{A})$.

Теория.

Вывести формулы для подсчета следующих величин:

1. апостериорного распределения на координаты лица на изображениях $p(\boldsymbol{d}_k \mid \boldsymbol{X}_k, \boldsymbol{\theta}, \boldsymbol{A})$ на E-шаге;
2. точечных оценок на параметры $\boldsymbol{A}, \boldsymbol{\theta} = \{\boldsymbol{F}, \boldsymbol{B}, s^2\}$ на M-шаге для EM и MAP-EM алгоритмов (учтите, что точечные оценки здесь нужно получать именно в таком порядке: $\boldsymbol{A}, \boldsymbol{F}, \boldsymbol{B}, s^2$);
3. нижней оценке на логарифм неполного правдоподобия $\mathcal{L}(q, \boldsymbol{\theta}, \boldsymbol{A})$.

Программирование.

Прототипы всех основных функция выданы вместе с заданием. При оценке выполнения задания будет учитываться эффективность программного кода. Необходимо реализовать:

1. EM-алгоритм со вспомогательными функциями. В качестве критерия останова использовать следующее условие на нижнюю оценку на логарифм неполного правдоподобия:

$$\mathcal{L}(q, \boldsymbol{\theta}^{(t+1)}, \boldsymbol{A}^{(t+1)}) - \mathcal{L}(q, \boldsymbol{\theta}^{(t)}, \boldsymbol{A}^{(t)}) < tol,$$

   где (t) и $(t+1)$ означают номера итераций, а tol — небольшая константа. Для возможности проведения автоматической проверки решений вводится следующая унификация процедуры EM-алгоритма: на каждой итерации должен сначала запускаться E-шаг, потом M-шаг; вне итераций E и M шаги исполняться не должны (то есть если алгоритм работает 1 итерацию, E и M шаги должны быть исполнены ровно по 1 разу каждый).

2. Дополнить функции для выполнения M шага и EM алгоритма на случай hard EM алгоритма.

3. Функцию, запускающую EM-алгоритм несколько раз из разных начальных приближений.

4. Для проверки работы алгоритма сгенерировать выборку из небольших зашумленных нормальным шумом черно-белых изображений с одинаковым фоном и каким-то объектом в случайной позиции. Пример таких изображений до и после зашумления показан на рисунке 3. Эти изображения должны быть достаточно маленькими для быстрой отладки кода.

Рис. 3: Пример изображений для тестирования работы алгоритма до и после зашумления. Здесь $h = w = 15, H = 20, W = 30$.

Рекомендации по реализации:

- Значение $\mathcal{L}(q, \boldsymbol{\theta}, \boldsymbol{A})$ *должно* возрастать с течением итераций. Если это не так, в реализации или в выводе формул ошибка. Это хороший способ отладки вашей программы.

- Для ускорения можно вычислять $\mathcal{L}(q, \boldsymbol{\theta}, \boldsymbol{A})$ не на каждой итерации алгоритма. Используйте этот совет только в своих экспериментах, а на проверку код отправляйте с подсчетом этого функционала на каждой итерации, как и написано в прототипе.

- Для того, чтобы избежать проблем с точностью вычислений, следует везде, где это возможно, переходить от произведений к суммированию логарифмов. Этот момент частично учтен в прототипах функций: функция для подсчета $p(\boldsymbol{X}_k|\boldsymbol{d}_k, \boldsymbol{\theta})$ должна выдавать прологарифмированные значения вероятностей.

- При подсчете $q(\boldsymbol{d})$ для более устойчивой нормировки также стоит перейти к логарифмам вероятностей и воспользоваться следующим трюком:

$$\alpha_i = \log p_i(\dots) \quad \rightarrow \quad \frac{e^{\alpha_i}}{\sum_k e^{\alpha_k}} = \frac{e^{(\alpha_i - \max_j \alpha_j)}}{\sum_k e^{(\alpha_k - \max_j \alpha_j)}}.$$

- Для эффективной реализации вам могут пригодиться стандартные функции, которые вычисляют свертки. Например, fftconvolve.

Анализ.

Обязательно приведите в отчете примеры исходных данных и полученных результатов (F, B).

1. Протестируйте полученный EM алгоритм на сгенерированных данных. Сильно ли влияет начальное приближение на параметры на результаты работы? Стоит ли для данной задачи запускать EM алгоритм из разных начальных приближений?
2. Запустите EM алгоритм на сгенерированных выборках разных размеров и с разным уровнем зашумления. Как изменения в обучающей выборке влияют на результаты работы (получаемые F, B и $\mathcal{L}(q, \boldsymbol{\theta}, \boldsymbol{A})$)? При каком уровне шума EM-алгоритм перестает выдавать вменяемые результаты? В данном пункте учтите, что для сравнения значения $\mathcal{L}(q, \boldsymbol{\theta}, \boldsymbol{A})$ для выборок разного размера стоит нормировать его на объем выборки.
3. Сравните качество и время работы EM и hard EM на сгенерированных данных. Как Вы думаете, почему разница в результатах работы так заметна?
4. Примените EM алгоритм к данным с зашумленными снимками преступника. Приведите результаты работы алгоритма на выборках разного размера.
5. Предложите какую-нибудь модификацию полученного EM алгоритма, которая бы работала на данной задаче качественнее и/или быстрее.
6. * Реализуйте предложенную модификацию EM алгоритма и сравните ее с исходной по качеству результатов и времени работы.

Пункт со * является необязательным. За его выполнение можно получить дополнительные баллы. Также дополнительные баллы будут выставлены за наиболее интересные ответы на пункт 5.

Формат данных

Данные с зашумленными фотографиями преступника будут выкладываться каждые 3 дня начиная с дня выдачи задания (всего 5 раз). Каждый раз будет выкладываться файл, который будет содержать все ранее выданные данные и новую порцию данных. Файл будет иметь формат `.npy`, то есть это будет сохраненный `numpy array` размера $H \times W \times K$, где $H \times W$ — размер каждого изображения, а K — их число. Значения h, w в выданных данных указаны в описании задания в anytask.

4 Оформление задания

Срочное сообщение Нику. Сообщение Нику является необязательным и не влияет на оценку за задание. Сообщение следует отправить письмом по адресу nickolas.j.carter@gmail.com с темой письма «Место обучения - Ваше Имя - Имя преступника». Возможные места обучения: ВМК, ШАД, ФКН.

В письме нужно указать кого Вы подозреваете в преступлении. Также свои подозрения нужно подтвердить, например, фотографией преступника, которую Вам удалось получить. Присылать сообщение Нику можно **только один раз**. Сдача полного задания не считается сообщением Нику. Первые трое приславших сообщение с правильным ответом получат благодарность от Ника и небольшие призы.

Полная сдача задания. На проверку в ejudge нужно отправить Python модуль со всеми требуемыми функциями в соответствии с прототипами, приведенными в отдельном файле. Модуль должен называться name_surname.py, например, petr_ivanov.py. Модуль не должен содержать никакого main! То есть при импорте модуля никакие вычисления производиться не должны.

Перед отправкой кода в ejudge его нужно проверить с помощью выдаваемых открытых тестов. Если какой-то из них выдает предупреждение (кроме тестов по времени), то ваш код не соответствует прототипам и не может быть проверен. Предупреждения по времени говорят о том, что ваш код не достаточно эффективен, что может привести к понижению оценки.

На проверку в anytask нужно отправить:

- Тот же Python модуль, который был отправлен в ejudge. Если вы реализовывали модификацию как бонус (пункт 6 анализа), то также приложите код модификации.
- Отчет в формате PDF с указанием ФИО, содержащий описание всех проведённых исследований (вывод необходимых формул, графики, анализ и выводы). Отчет не должен содержать листинга кода и подобных вещей! Желательно для составления отчета использовать latex. Файл должен называться name_surname.pdf.

Будьте внимательны к формату названий файлов!