Московский государственный университет имени М. В. Ломоносова

Факультет Вычислительной Математики и Кибернетики
Кафедра Математических Методов Прогнозирования
Байесовские методы машинного обучения

Отчет по практическому заданию №2

EM алгоритм для детектива

Выполнил:
студент 417 группы
Драпак Степан Николаевич

Москва, 2017

1 Постановка задачи

Пусть дана выборка $\mathbf{X} = \{X_k\}_{k=1}^{K}$, где X_k – картинка размером HxW. При этом известно, что на этой картинке есть лицо, размером hxw. Лица на разных картинках располагаются в разных частях изображения. Задача состоит в том чтобы отделить фон от лица. Введем обозначения:

$X_k(i,j)$ – Пиксель k-го изображения
$\mathbf{B} \in \mathbb{R}^{HxW}$ – маска фона без лица.
$\mathbf{F} \in \mathbb{R}^{hxw}$ – маска лица
$\mathbf{d}_k = (d_k^h, d_k^w)$ — координаты верхнего левого угла маски лица на k-ом изображении (d_k^h — по вертикали, d_k^w — по горизонтали), $\mathbf{d} = (\mathbf{d}_1, ..., \mathbf{d}_K)$ — набор координат для всех изображений выборки.

Также будем считать шум на изображении независимым для каждого пикселя и принадлежащим нормальному распределению $N(0, s_2)$, где s — стандартное отклонение. Таким образом для одного изображения имеем:

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

$$p(d_k|A) = A(d_k^h, d_k^w), \sum_{ij} A(i,j) = 1$$

В итоге вероятностная модель выглядит так:

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

2 Вывод формул для EM-Алгоритма

2.1 E-шаг

В нашем алгоритме в качестве скрытых переменных выступает вектор $q(\mathbf{d})$. На E-шаге нас будет интересовать формула для пересчета распределений на них.

$$q(\mathbf{d}) = \prod_k p(d_k|X_k,\theta,A) \propto \prod_k p(d_k,X_k|\theta,A) = p(X_k|d_k,\theta,A)p(d_k|A)$$

$$\log p(X_k|d_k,\theta) = \sum_{ij} \Big[-\frac{1}{2}\log(2\pi s) -$$

$$\frac{1}{2s} \underbrace{\Big([(i,j) \in faceArea(d_k)](X_k(i,j) - F(i+d_k^h, j+d_k^w))^2 + [(i,j) \notin faceArea(d_k)](X_k(i,j) - B(i,j))^2 \Big)\Big]}_{\gamma_{ij}}$$

2.2 M-шаг

$$\mathcal{L}(\theta,A) = \mathbb{E}_{q(d)} \log p(X,d|\theta,A) = \mathbb{E}_{q_k(d_k)}\big(\log p(X_k|d_k,\theta) + \log p(d_k|A)\big) =$$

$$\sum_k \sum_{d_k} \Big[q(d_k) \sum_{ij} \Big(-\frac{1}{2}\log(2\pi s) - \frac{1}{2s}\gamma_{ij} \Big) + q(d_k)\log A(d_k) \Big]$$

2.3 Оптимизация A

Отбрасывая все, что не зависит от A получаем:

$$\mathcal{L}^A = \sum_k \sum_{d_k} q(d_k) \log A(d_k) = \sum_{ij} \log A(i,j) \sum_k q_k(i,j) \to \max_A$$

При этом учитываем, что $\sum_{ij} A(i,j) = 1$

Выпишем лагранжиан:

$$\mathbf{L} = \sum_{ij} \log A(d_k) \sum_k q_k(i,j) - \lambda \Big(\sum_{ij} A(i,j) - 1 \Big)$$

Дифференцируем и получаем условия на экстремум:

$$\begin{cases} A(i,j) = \frac{\sum_k q_k(i,j)}{\lambda} \\ \sum_{ij} A(i,j) = 1 \end{cases}$$

В итоге $A(i,j) \propto \sum_k q_k(i,j)$, а окончательный результат получим из второго условия, нормировки.

2.4 Оптимизация по F

$$\mathcal{L}^F(i,j) \propto \sum_k \sum_{d_k} q(d_k) \big(X_k(i + d_k^h, j + d_k^w) - F(i,j) \big)^2$$

Найдем экстремум:

$$(\mathcal{L}^{F(i,j)})' \propto \sum_k \sum_{d_k} q(d_k) \big(X_k(i + d_k^h, j + d_k^w) - F(i,j) \big) = 0$$

$$F(i,j) = \frac{\sum_k \sum_{d_k} q(d_k) \big(X_k(i + d_k^h, j + d_k^w) - F(i,j) \big)}{K}$$

2.5 Оптимизация по B

$$\mathcal{L}^B(i,j) \propto \sum_k \sum_{(i,j) \notin faceArea(d_k)} q(d_k)(X_k(i,j) - B(i,j))^2$$

$$(\mathcal{L}^B(i,j))' \propto \sum_k \sum_{(i,j) \notin faceArea(d_k)} q(d_k)(X_k(i,j) - B(i,j)) = 0$$

$$\sum_k \sum_{(i,j) \notin faceArea(d_k)} q(d_k) X_k(i,j) = B(i,j) \sum_k \sum_{(i,j) \notin faceArea(d_k)} q(d_k)$$

$$B(i,j) = \frac{\sum_k \sum_{(i,j) \notin faceArea(d_k)} q(d_k) X_k(i,j)}{\sum_k \sum_{(i,j) \notin faceArea(d_k)} q(d_k)}$$

2.6 Оптимизация по s

$$\mathcal{L}^s \propto \sum_k \sum_{d_k} q(d_k) \sum_{ij} \Big[\frac{1}{2} \log(2\pi s) + \frac{1}{2s} \gamma_{ij} \Big]$$

$$(\mathcal{L}^s)' = \sum_k \sum_{d_k} q(d_k) \sum_{ij} \Big[\frac{1}{2s} - \frac{1}{2s^2} \gamma_{ij} \Big] = 0$$

$$\frac{HW}{s} \sum_k \sum_{d_k} q(d_k) = \frac{1}{s^2} \sum_k \sum_{d_k} q(d_k) \sum_{ij} \gamma_{ij}$$

$$s = \frac{\sum_k \sum_{d_k} q(d_k) \sum_{ij} \gamma_{ij}}{HW \sum_k \sum_{d_k} q(d_k)} = \frac{\sum_k \sum_{d_k} q(d_k) \sum_{ij} \gamma_{ij}}{HWK}$$

2.7 Модификация для Hard-EM

Для E-шага получаем:

$$q(d_k) = arg\max_{i,j} p(d_k|X_k, \theta, A)$$

С точки зрения математики для M-шага ничего не меняется. С точки зрения программирования получаются некоторые упрощения.

2.8 Нижняя оценка на неполное правдоподобие

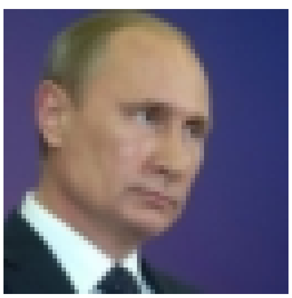
$$\mathcal{L} = \mathbb{E}\log p(X, d|\theta, A) - \mathbb{E}\log q(d)$$

$$\mathcal{L} = \sum_k \mathbb{E}\log p(X_k|d_k, \theta) + \sum_k \mathbb{E}\log p(d_k|A) - \mathbb{E}\log q(d)$$

$$\mathcal{L} = \sum_k \mathbb{E}q(d_k)\underbrace{\log p(X_k|d_k, \theta)}_{\text{E-step}} + q(d_k)\log A(d_k) + q(d_k)\log q(d_k)$$

3 Эксперименты

Для базовых экспериментов рассмотрим следующие картинки:



Портрет, 64x64

Фон, 96x96

Пример исходных данных для алгоритма

Результат при исходной дисперссии 50, K = 200

Рис. 1: Пример данных

Теперь сравним результаты при на одинаковой выборке, но с разными начальными приближениями. Возьмем дисперсию 50 и 2 выборки, размером 50 (2) и 200 (3).

Видно, что результаты совершенно разные. Даже на выборке в 200 объектов результаты хоть и не так сильно расходятся как на маленькой выборке, но все равно, значительно отличаются, как визуально, так и по правдоподобию. Хотя, есть гипотеза, что на очень больших выборках локальные минимумы будут не так далеко друг от друга и скатиться в них сложнее, но возможности проверить это пока нет. Исходя из этого, очевидно, что есть смысл запускать алгоритм несколько раз из

mean LL = -55567

mean LL = -50061

mean LL = -49989

Рис. 2: σ^2=50, K=50

mean LL = -49281

mean LL = -49102

mean LL = -50368

Рис. 3: σ^2=50, K=200

разных приближений. Так же очевидно, что дополнительные объекты неплохо увеличили среднее правдоподобие и качество картинки.

Теперь посмотрим, как дисперсия влияет на результаты восстановления 4

$\sigma = 50$, mean LL = -50061

$\sigma = 100$, mean LL = -55490

$\sigma = 150$, mean LL = -59392

Рис. 4: Разные дисперсиии, K = 50

Далее сравним Hard-EM и обычный алгоритм на средней по качеству выборке. 5

Получилось, что при K = 100 Hard-EM даже немного выиграл, хотя, мне кажется, тут все дело в удачном начальном приближении, хотя результаты были взяты лучшие среди 3х попыток. При совсем малом объем классический алгоритм показал себя чуть лучше. При этом, Hard-EM работал чуть медленнее, хотя возможно дело в удачном хеширование. Так или иначе, я считаю, что в моей реализации основное узкое место – функция get lpx d all, а она одинакова для этих алгоритмов.

Протестируем алгоритмы на разных подвыборках выборки с фотографиями Ïреступника ̈: 6 7 8.

Вообще говоря, Hard-EM не сильно уступает. Все больше зависит от начальных приближений, хотя и скорость работы у него не сильно выше.

Базовый алгоритм, mean LL = -52925, K = 100

Hard, mean LL = -52762, K = 100

Базовый алгоритм, mean LL = -42198, K = 50

Hard, mean LL = -42698, K = 50

Рис. 5: Сравнение Hard-EM и обычного подхода, $\sigma = 75$

4 Возможные модификации

Самым простым было бы, пожалуй, поработать с инициализацией. Например, взять и инициализировать фон как среднее по всем изображениям. Но практика показывает, что фон и так сходится почти после первой же итерации, поэтому большого смысла в этом нет. При этом, мы никак не учитываем, что у объекта, фотографии, есть достаточно четкие края в исходном изображении. Хотелось бы это каким-то образом учесть. Самое простое что приходит в голову, давайте возьмем и пропусти исходное изображения через фильтр и добавим их в EM-алгоритм. Тогда мы получим не 1 канал(яркость), а два канала. Предположим, что результат применения фильтра к нашим картинкам независимы с самими картинками(что разумеется не так). Тогда имеем:

$$p(X|d,B,F) = P(X_{srcImage}|d,B,F)p(X_{edge}|d,B,F)$$

Здесь $P(X_{srcImage}|d,B,F)$ – остается старым, а для второго множителя предположим:

$$p(X_{edge}|d,B,F) = N(X_{edge}|\hat{X}, s1^2)$$

Здесь $\hat{X}$ – картинка, получающаяся нахождением границ на картинке, с фоном F и наложенным B в позиции d.

Базовый алгоритм,K = 1000

Hard, K = 1000

Рис. 6: Полная выборка

5 Выводы

В целом, какие-то результаты получаются, но очень медленно. Да и задача сильно вырождена. Hard-EM показал себя... никак. Он просто не отличается от обычного метода в данных экспериментах. На таких выборках все сводится к удачному начальному приближению и хешу.

Базовый алгоритм, K = 500

Hard, K = 500

Рис. 7: 500 объектов

Базовый алгоритм, K = 100

Hard, K = 100

Рис. 8: 100 объектов