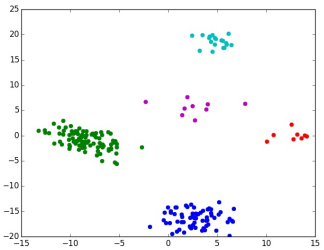
ВМК МГУ

Задание 4. Методы восстановления плотности распределений в задаче вычитания фона

Практикум 317 группы, осень 2016

Начало выполнения задания: 7 декабря 2016 года.
Срок сдачи: **21 декабря 2016 года, 23:59.**
Максимальный балл: 5.0 (плюс бонусные баллы).



1 Задание

Вычитание фона — важная прикладная задача. Часто оно является первой стадией в системах анализа видео с камер наблюдения. Необходимо классифицировать пиксели видеопоследовательности на принадлежащие фону и принадлежащие объектам переднего плана. Обычно предполагается, что камера статична. По обучающей выборке необходимо оценить модель фона, чтобы затем для каждого пикселя каждого кадра тестовой видеопоследовательности уметь предсказать вероятность того, что он является фоном. Предполагая равномерное распределение на принадлежность пикселей объектам переднего плана (обычно нет априорных сведений, чтобы использовать другое распределение), принятие решения для данного пикселя эквивалентно сравнению предсказанной вероятности фона с некоторым порогом.

Часто (хотя и не всегда) в качестве обучающей выборки удаётся взять один или несколько кадров, на которых отсутствуют объекты переднего плана. Но даже в этом случае задача оценки модели фона не является тривиальной из-за того, что фон никогда не бывает полностью статичен: всегда есть шум камеры, также может меняться освещение, тени, камера может дрожать, фон может быть динамическим (листва или вода в присутствии ветра), к фону могут добавляться дополнительные объекты. Вам предлагается протестировать несколько моделей разной сложности и сделать выводы о применимости моделей в разных ситуациях.

2 Данные

В задании предлагается использовать стандартные тестовые последовательности с веб-сайта http://wordpress-jodoin.dmi.usherb.ca/dataset2012/. Два набора данных являются обязательными:

- Baseline/pedestrians (train — $1, \ldots, 299$; test — $300, \ldots, 1099$),
- Camera jitter/traffic (train — $[581, \ldots, 723, 752, \ldots, 951]$; test — $1000, \ldots, 1570$).

Можно использовать и другие последовательности для лучшего изучения свойств методов.

3 Оценка алгоритмов

Для каждого кадра тестовой части последовательности можно оценить качество алгоритма, зная его верную разметку (директория groundtruth). Обратите внимание, что только часть каждого из кадров является областью интереса. Например, на последовательности traffic область интереса ограничивается дорогой. Зная, какие части изображения на самом деле относятся к фону (класс static) и к переднему плану (класс motion), можно оценить количество верных положительных (TP) и отрицательных (TN) обнаружений пикселей, а также ложноположительных (FP) и ложноотрицательных (FN). Пиксели с промежуточными метками (50, 85, 170) не входят в регион интереса и не участвуют в подсчёте точности. Другими словами, значения TP, FP, TN, FN не зависят от того, что вернул алгоритм на этих пикселях.

Для оценки качества вычитания фона следует использовать следующие три инструмента:

- Для каждого тестового кадра вычисляется количество ошибок 1 рода (FP) и 2 рода (FN). Далее можно построить график зависимости ошибок от номера кадра.
- Визуальная оценка качества. Для этого на каждом из кадров выделяются области, отнесённые к переднему плану.
- Оценка величин TPR и FPR (доли верноположительных и ложноположительных обнаружений, соответственно) для разных значений порога по всей тестовой последовательности и построение графика ROC-кривой. Площадь под графиком ROC кривой (AUC ROC) является одним из способов сравнить качество разных моделей. Кроме того, можно оценивать величину AUC ROC для каждого кадра по отдельности.

Для удобства к заданию прилагается python код, который содержит функцию подсветки маски переднего плана, а также функцию, которая позволяет проигрывать видео. Данный код может быть полезным для визуальной оценки качества работы методов. Для отображения видео в IPython Notebook рекомендуется установить виджет `JSAnimation` из репозитория https://github.com/jakevdp/JSAnimation и импортировать следующий модуль – `from JSAnimation import IPython_display`. Подробнее информацию про установку можно посмотреть здесь: https://gist.github.com/gforsyth/188c32b6efe834337d8a Для создания отдельного видео-файла в функции `make_video()` в последней строчке нужно убрать lambda-функцию.

4 Смесь многомерных нормальных распределений

Рассмотрим вероятностную модель смеси K нормальных распределений:

$$p(\boldsymbol{x}|\boldsymbol{w}, \{\boldsymbol{\mu}_k\}, \{\Sigma_k\}) = \sum_{k=1}^{K} \omega_k \mathcal{N}(\boldsymbol{x}|\boldsymbol{\mu}_k, \Sigma_k), \ \sum_{k=1}^{K} \omega_k = 1, \ \omega_k \geq 0 \tag{1}$$

Можно ввести скрытую переменную t, которая будет обозначать номер компоненты смеси:

$$p(t = k) = \omega_k, \ p(\boldsymbol{x}|t = k) = \mathcal{N}(\boldsymbol{x}|\boldsymbol{\mu}_k, \Sigma_k)$$

Для аппроксимации выборки $X = (x_1, \ldots, x_n)$ с помощью модели смеси из K гауссиан можно воспользоваться методом максимального правдоподобия:

$$p(X|\boldsymbol{w}, \{\boldsymbol{\mu}_k\}, \{\Sigma_k\}) = \prod_{n=1}^{N} p(\boldsymbol{x}_n|\boldsymbol{w}, \{\boldsymbol{\mu}_k\}, \{\Sigma_k\}) \to \max_{\boldsymbol{w}, \{\boldsymbol{\mu}_k\}, \{\Sigma_k\}}$$

$$\sum_{k=1}^{K} \omega_k = 1, \ \omega_k \geq 0$$

$$\Sigma_k = \Sigma_k^T, \ \Sigma_k \succ 0$$

Данная задача может быть эффективно решена с помощью EM-алгоритма, подробное описание которого можно найти в книге [1], глава 9.

5 Модели фона

5.1 Одномерная гауссиана

В самом простом варианте можно моделировать распределение цвета в каждом пикселе одномерным нормальным распределением. Для этого нужно перевести видеопоследовательность в полутоновую. Это можно сделать, например, с помощью формулы из стандарта `NTSC`: $gs = 0.2126r + 0.7152g + 0.0722b$. Затем для каждого пикселя обучающей части последовательности оцениваются параметры нормального распределения (например, для **pedestrians** будет 299 точек для оценки плотности в каждом пикселе). Стоит обратить внимание, что при такой оценке некоторые гауссианы могут получиться вырожденными. В этом случае значение дисперсии разумно ограничить снизу. Далее, при классификации пикселя тестовой выборки, его нужно относить к фону тогда и только тогда, когда его яркость отклоняется от μ меньше, чем на $k\sigma$. Если установить $\sigma = 3$, то ожидается, что к переднему плану будут отнесены только 0.27% пикселей фона ("правило 3σ"). Изменяя этот порог, можно регулировать соотношение между количеством ложноположительных и ложноотрицательных обнаружений.

5.2 Оценка гауссианы на лету

Если в последовательности присутствует дрейф фона, то желательно оценивать модель фона адаптивно с учётом информации от нескольких последних кадров. Для этого можно использовать приведённый ниже итеративный алгоритм оценки параметров гауссовской модели фона на лету (оценка также производится по полутоновым изображениям независимо для разных пикселей):

Параметр ρ отвечает за величину памяти метода (размер окна): чем меньше ρ, тем большее влияние имеют старые кадры. Здесь также стоит вводить регуляризацию и устанавливать минимальное значение дисперсии.

Инициализировать μ_1, σ_1^2.
for $t = 1, \ldots,$ *Number_ of_ frames* **do**
  Принять решение о метке пикселя на кадре t: передний план тогда и только тогда, когда $\frac{|(I_t - \mu_t)|}{\sigma_t} > k$, где I_t — яркость пикселя на кадре t.
  Если пиксель отнесён к фону, обновить $\mu_{t+1} = \rho I_t + (1 - \rho)\mu_t$, иначе $\mu_{t+1} = \mu_t$.
  Если пиксель отнесён к фону, обновить $\sigma_{t+1}^2 = (I_t - \mu_{t+1})^2\rho + (1 - \rho)\sigma_t^2$, иначе $\sigma_{t+1}^2 = \sigma_t^2$.

5.3 Многомерная гауссиана

Рассмотрим более гибкую модель фона: трёхмерное нормальное распределение в цветовом пространстве RGB. При этом можно рассматривать два варианта модели: с полной и с диагональной матрицей ковариаций. При оценке модели по данным стоит использовать регуляризацию матрицы ковариации (например, путём добавления положительного числа к диагонали). Решение о метке пикселя тестового кадра принимается путём сравнения значения плотности нормального распределения в данной точке с порогом.

5.4 Смесь гауссиан

Смесь распределений может быть полезна для моделирования фона, если он нестатичен, например, содержит воду или листву, т.е. когда распределение может иметь несколько мод.

6 Формулировка задания

1. Реализовать EM-алгоритм для восстановления смеси многомерных нормальных распределений, согласно спецификации (реализация возможности нескольких запусков из случайных начальных приближений с выбором наилучшего по значению найденного правдоподобия, возможности восстановления смеси нормальных распределений с диагональными матрицами ковариации). Требования по эффективности реализации: среднее время одной EM-итерации для $N = 10000$ объектов, $D = 100$ признаков и $K = 10$ компонент смеси не должно превышать одной секунды. Для реализации из математических пакетов разрешается использовать только `numpy` и модули `scipy.linalg` и `scipy.misc`.

2. Провести тестирование реализованного EM-алгоритма на двумерных модельных данных. Для этого сгенерировать данные из смеси распределений с заданными параметрами, а затем восстановить по этим данным параметры смеси с помощью EM-алгоритма. Отобразить результат восстановления, где объекты выборки, соответствующие одинаковым компонентам смеси, показаны одинаковыми цветами. Убедиться в том, что значение правдоподобия в EM-итерациях монотонно не убывает. Привести пример ситуации, когда результат работы EM-алгоритма зависит от начального приближения.

3. Реализовать оценку модели фона с помощью одномерной гауссианы (секция 5.1) и протестировать на последовательности **pedestrians**. Проанализировать качество вычитания фона с помощью трёх инструментов, описанных в секции 3.

4. Реализовать оценку модели фона на основе адаптивной одномерной гауссианы (секция 5.2). Протестировать метод на последовательности **pedestrians** и сравнить результаты с предыдущим пунктом.

5. Реализовать оценку модели фона с помощью многомерной гауссианы в цветовом пространстве RGB и протестировать результат на последовательности **pedestrians**. Проанализировать ошибки метода и сравнить его с остальными. Повторить эксперимент, используя цветовое пространство HSV, в котором каналы коррелируют меньше. Для перевода изображения можно воспользоваться встроенной функцией `matplotlib.colors.rgb_to_hsv`. Сделать выводы об уместности использования диагональной матрицы ковариаций для разных задач.

6. Запустить алгоритм разделения смеси 3 трёхмерных гауссиан для вычитания фона в последовательности **traffic**, выбрав модель с учётом результатов предыдущего пункта (важны точность и скорость работы). Проанализировать ошибки метода и сравнить результат с использованием одной гауссианы.

7. Написать отчёт в системе LaTeXв формате PDF или в IPython notebook с описанием результатов проведённых исследований. Наилучшее из полученных решений для алгоритма вычитания фона должно быть добавлено в отчёт в виде анимации. Формы анимации могут быть различными: виджет в IPython notebook `JSAnimation`, отдельный файл с видео, анимированные изображения кадров, как в примере ниже, и проч.

7 Спецификация к заданию

Необходимо написать модуль `em.py`, в котором должны присутствовать следующие функции:

1. Функция `run_em(X, n_components, tol, max_iter, diag, mu_start, sigma_start)` — реализация EM-алгоритма для восстановления смеси многомерных нормальных распределений.

   Описание параметров:

   - `X` — `numpy ndarray` размера $N \times D$
   - `n_components` —`int`, число компонент смеси (соответствует K в модели 1)
   - `tol` — `float`, точность метода (критерий останова итерационного алгоритма)
   - `max_iter` — `int`, максимальное количество итераций алгоритма
   - `diag` — `bool`, если значение переменной `True`, то необходимо восстанавливать смесь с диагональными матрицами ковариаций
   - `mu_start` — `numpy ndarray` размера $K \times D$ или `None`, начальное приближение для параметра $\{\mu_k\}$ в модели 1 (при `None` использовать случайное приближение)
   - `sigma_start` — `numpy ndarray` размера $K \times D \times D$ в случае `diag=False`, размера $K \times D$ в случае `diag=True` или `None`, начальное приближение для параметра $\{\Sigma_k\}$ в модели 1 (при `None` использовать случайное приближение)

   Функция возвращает точечные оценки на параметры смеси — `tuple` из следующих элементов:

   - `w` — `numpy ndarray` размера $1 \times K$
   - `mu` — `numpy ndarray` размера $K \times D$
   - `sigma` — `numpy ndarray` размера $K \times D \times D$ в случае `diag=False` и размера $K \times D$ в случае `diag=True`
   - `log_likelyhood_curve` — список, размер которого равен количеству прошедших итераций алгоритма, содержащий значения логафрима правдоподобия выборки после каждой итерации алгоритма

2. Функция `run_em_with_restarts(X, n_restars, n_components, tol, max_iter, diag)` — реализация EM-алгоритма для восстановления смеси многомерных нормальных распределений с возможностью нескольких запусков из разных начальных приближений.

   Функция возвращает наилучшие точечные оценки на параметры смеси по всем начальным приближениям. Параметры и список возвращаемых значений функции такие же, как и у функции `run_em`, за исключением параметра `n_restars` — `int`, число запусков из разных начальных приближений.

8 Рекомендации по реализации

1. При нормировке на E-шаге могут возникать численные неустойчивости. Для более усточивой нормировки рекомендуется воспользоваться следующим трюком:

$$\alpha_i = \log p_i(\dots), \quad \frac{e^{\alpha_i}}{\sum_s e^{\alpha_s}} = \frac{e^{\alpha_i - \max_j \alpha_j}}{\sum_s e^{\alpha_s - \max_j \alpha_j}}$$

2. Значение правдоподобия должно монотонно не убывать с течением итераций. Это хороший способ отладки программы.

9 Бонусные баллы

- до +0.3 балла. Вывести формулы для подсчёта апостериорного распределения на E-шаге и точечных оценок на параметры $\boldsymbol{w}, \{\boldsymbol{\mu}_k\}, \{\Sigma_k\}$ на M-шаге. Вывод формул должен присутствовать в отчёте.
- до +1.0 балла. Реализация и оценка работы модификаций предложенных методов. Размер бонуса зависит от качества работы и оригинальности модификации, а также является субъективным и безапелляционным.

Список литературы

[1] C. Bishop. Pattern recognition and machine learning. Springer, 2006