EM-алгоритм для разделения смеси многомерных бинарных распределений

В рамках данного задания вам необходимо выделить из коллекции бинаризованных изображений размером 28×28 шаблоны цифр. Каждый шаблон задается вероятностями каждого пикселя изображения принимать значение 1. Данная задача может быть решена при помощи EM-алгоритма для разделения смеси многомерных бинарных распределений.

Пусть всего у нас K шаблонов, то есть K компонент смеси. Объекты выборки - это D-мерные вектора, состоящие только из нулей и единиц: $X = \{\mathbf{x}_i\}_{i=1}^N, \mathbf{x}_i \in \{0,1\}^D$. Каждое из K бинарных распределений задает вероятность признака d принимать значение 1, обозначаемое θ_{kd}. Соответственно вероятность признака d быть равным 0 есть $1 - \theta_{kd}$. Таким образом, правдоподобие объекта $\mathbf{x}_i$ при условии принадлежности распределению k есть:

$$p(\mathbf{x}_i|z_i = k, \theta) = \prod_{d=1}^{D} \theta_{kd}^{x_{id}}(1 - \theta_{kd})^{1-x_{id}},$$

где z_i - скрытая переменная, принимающая значения от 1 до K, кодирующие к какому распределению относится объект. Априорное распределение на z_i будем полагать равномерным: $p(z_i = k) = 1/K$.

EM-алгоритм итеративно оптимизирует логарифм неполного правдоподобия:

$$\log p(X|\theta) = \sum_{i=1}^{N} \log \sum_{k=1}^{K} p(\mathbf{x}_i|z_i = k, \theta) - N \log K$$

На E-шаге рассчитывается апостериорное распределение на скрытые переменные при старых значениях параметров:

$$p(\mathbf{z}|X, \theta^{old}) = \frac{p(\mathbf{z})p(X|\mathbf{z}, \theta^{old})}{p(X|\theta^{old})}$$

В данном случае апостериорные распределения для каждого объекта независимы, то есть, $p(\mathbf{z}|X, \theta) = \prod_{i=1}^{N} p(z_i|\mathbf{x}_i, \theta)$.

Затем, на M-шаге выполняется оптимизиация по параметрам распределений:

$$\theta^{new} = \arg\max_{\theta} \mathbb{E}_{p(\mathbf{z}|X,\theta^{old})} \log p(X, \mathbf{z}|\theta)$$

В данном случае максимум можно найти, приравняв производную по каждому из параметров к нулю.

Эти два шага чередуются необходимое число итераций.

1. Выпишите конечные формулы для апостериорных распределений $p(z_i|\mathbf{x}_i, \theta)$ и реализуйте функцию `posterior`.

2. Найдите оценки для параметров, получаемые на М-шаге, и реализуйте функцию `learn_clusters`.

3. Считайте обучающие и тестовые данные из файлов `mnist_train.csv` и `mnist_test.csv`. Первый столбец - это метка цифры, изображенной на данной картинке. Бинаризуйте все изображения по порогу 127.

4. Запустите EM-алгоритм на изображениях цифр 6 и 9 для $K = 2$, сделайте 30 итераций. Постройте график логарифма правдоподобия в зависимости от числа итераций, а также визуализируйте шаблоны, полученные после пересчета на каждой итерации. Удалось ли вам получить шаблоны этих цифр?

5. Выполните аналогичное исследование для всех изображений всех цифр. Используйте 50 итераций и разные значения $K = 10, 15, 20$, для каждого визуализируйте логарифм неполгого правдоподобия и получаемые шаблоны после каждой итерации EM-алгоритма. Если ваша реализация работает слишком медленно, отберите некоторое количество изображений каждой цифры из всей выборки и работайте только с ними. Эффективная реализация будет поощряться дополнительными баллами.

   (a) Для каких значений K вам удалось выделить шаблоны всех цифр?

   (b) Какие цифры оказались самыми сложными для распознавания и потребовали нескольких шаблонов?

   (c) Какое число шаблонов обеспечивает максимум неполного правдоподобия (после последней итерации EM-алгоритма)? Попробуйте увеличивать K с некоторым шагом до тех пор, пока значение неполного правдоподобия не начнет падать.

   (d) Измерьте для каждого значения K правдоподобие тестовой выборки. Коррелирует ли оно с правдоподобием обучающей выборки?

6. Для одной из обученных моделей вручную свяжите каждый шаблон с цифрой, которая на нем изображена. Затем, используя апостериорные распределения объектов тестовой выборки и привязку шаблонов к цифрам, определите цифру каждого тестового изображения и подсчитайте точность классификации. Какого качества классификации удалось достигнуть?

Для выполнения задания необходимо реализовать следующие функции:

`posterior(x, clusters)` $\rightarrow$ `z`

Функция возвращает апостериорное распределение каждой точки выборки (E-шаг).

Аргументы

- `x` - numpy-массив размера $N \times D$, в строках которого хранятся признаки объектов

- `clusters` - numpy-массив размера $K \times D$, каждая строка которого описывает одно из K распределений многомерных бинарных распределений: `clusters[k, d]` - вероятность того, что d-й признак объекта принимает значение 1.

Возвращаемые значения

- `z` - numpy-массив размера $K \times N$: `z[k, i]` - апостериорная вероятность объекта i принадлежать распределению k.

`learn_clusters(x, z)` $\rightarrow$ `clusters`

Функция восстанавливает параметры компонент смеси по данным и их апостериорным распределениям (M-шаг).

Аргументы

- `x` - numpy-массив размера $N \times D$, в строках которого хранятся признаки объектов
- `z` - numpy-массив размера $K \times N$: `z[k, i]` - апостериорная вероятность объекта i принадлежать распределению k.

Возвращаемые значения

- `clusters` - numpy-массив размера $K \times D$, каждая строка которого описывает одно из K распределений Бернулли: `clusters[k, d]` - вероятность того, что d-й признак объекта принимает значение 1.

`likelihood(x, clusters)` $\rightarrow$ `ll`

Функция возвращает логарифм неполного правдоподобия.

Аргументы

См. аргументы функции `posterior`.

Возвращаемые значения

- `ll` - логарифм неполного правдоподобия

`em_algorithm(x, K, maxiter)` $\rightarrow$ `clusters, ll`

Функция измеряет качество классификации объектов

Аргументы

- `x` - numpy-массив размера $N \times D$, содержащий объекты выборки
- `K` - число компонент смеси (кластеров)
- `maxiter` - число итераций, которое необходимо сделать алгоритму

Возвращаемые значения

- `ll` - numpy-массив размера `maxiter` со значениями неполного правдоподобия после каждой итерации алгоритма
- `clusters` - numpy-массив размера `maxiter` $\times K \times D$, в котором хранятся параметры всех распределений после каждой итерации.