ВМК МГУ

Задание 2. Байесовская смесь распределений Бернулли

Курс: Байесовские методы в машинном обучении, 2014

Начало выполнения задания: 7 ноября.
Срок сдачи: **21 ноября, 23:59.**
Среда для выполнения задания: MATLAB или Python.

1 Описание модели

Пусть $\mathbf{x} = (x_1, \ldots, x_D)^{\mathrm{T}}$ — набор из D бинарных случайных величин x_i, каждая из которых имеет распределение Бернулли с параметром μ_i. Таким образом

$$p(\mathbf{x} \mid \boldsymbol{\mu}) = \prod_{i=1}^{D} \mu_i^{x_i}(1-\mu_i)^{(1-x_i)}, \tag{1}$$

где $\boldsymbol{\mu} = (\mu_1, \ldots, \mu_D)^{\mathrm{T}}$. Рассмотрим смесь таких распределений:

$$p(\mathbf{x} \mid \boldsymbol{\mu}, \boldsymbol{\pi}) = \sum_{k=1}^{K} \pi_k p(\mathbf{x} \mid \boldsymbol{\mu}_k), \tag{2}$$

где $\boldsymbol{\mu} = \{\boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_k\}$, $\boldsymbol{\pi} = \{\pi_1, \ldots, \pi_k\}$.

Пусть задана обучающая выборка $\mathbf{X} = \{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$. Для каждого $\mathbf{x}$ введем скрытую переменную $\mathbf{z} = (z_1, \ldots, z_K)^{\mathrm{T}}$ — бинарный вектор, у которого только одна компонента равна 1, а все остальные равны 0. Тогда можно записать условное распределение на $\mathbf{x}$ при известной $\mathbf{z}$:

$$p(\mathbf{x} \mid \mathbf{z}, \boldsymbol{\mu}) = \prod_{k=1}^{K} p(\mathbf{x} \mid \boldsymbol{\mu}_k)^{z_k}. \tag{3}$$

Введем распределение на $\mathbf{z}$:

$$p(\mathbf{z} \mid \boldsymbol{\pi}) = \prod_{k=1}^{K} \pi_k^{z_k}. \tag{4}$$

Также введем априорные распределения на параметры $\boldsymbol{\mu}$ и $\boldsymbol{\pi}$:

$$p(\boldsymbol{\pi} \mid \boldsymbol{\alpha}) = \mathrm{Dir}(\boldsymbol{\pi} \mid \boldsymbol{\alpha}), \tag{5}$$

$$p(\boldsymbol{\mu}_k \mid a, b) = \prod_{i=1}^{D} \mathrm{Beta}(\mu_{ki} \mid a, b), \quad k = 1, \ldots, K. \tag{6}$$

Будем рассматривать симметричное распределение Дирихле, т.е. $\boldsymbol{\alpha} = (\alpha, \ldots, \alpha)^{\mathrm{T}}$.

Совместное распределение модели:

$$p(\mathbf{X}, \mathbf{Z}, \boldsymbol{\mu}, \boldsymbol{\pi} \mid \alpha, a, b) = p(\mathbf{X} \mid \mathbf{Z}, \boldsymbol{\mu}) p(\mathbf{Z} \mid \boldsymbol{\pi}) p(\boldsymbol{\pi} \mid \alpha) \prod_{k=1}^{K} p(\boldsymbol{\mu}_k \mid a, b). \tag{7}$$

2 Формулировка задания

Требуется решить задачу

$$p(\mathbf{X}, \boldsymbol{\pi} \mid \alpha, a, b) \to \max_{\boldsymbol{\pi}}.$$

Для этого предлагается воспользоваться вариационным EM-алгоритмом, на E-шаге вычисляется вариационное приближение:

$$p(\mathbf{Z}, \boldsymbol{\mu} \mid \mathbf{X}, \boldsymbol{\pi}, \alpha, a, b) \approx q(\mathbf{Z}) q(\boldsymbol{\mu}),$$

на M-шаге вычисляется точечная оценка на $\boldsymbol{\pi}$:

$$\mathbb{E}_{q(\mathbf{Z})q(\boldsymbol{\mu})} \log p(\mathbf{X}, \mathbf{Z}, \boldsymbol{\mu}, \boldsymbol{\pi} \mid \alpha, a, b) \to \max_{\boldsymbol{\pi}}.$$

1. Выписать формулы для пересчета $q(\mathbf{Z})$ и $q(\boldsymbol{\mu})$.

2. Вывести формулу для подсчета функционала $\mathcal{L}(q)$ (вариационной нижней границы).

3. Реализовать вариационный EM-алгоритм.

4. Реализовать в EM-алгоритме поиск из нескольких случайных начальных приближений, с выбором лучшего по значению $\mathcal{L}(q)$.

5. Протестировать EM-алгоритм на модельных данных (сгенерировав выборку из смеси с известными параметрами).

6. Протестировать полученный алгоритм на базе MNIST. Сколько получается кластеров? Как меняется число кластеров при изменении параметров K, a, b, α? Визуализируйте полученные центры кластеров $\mathbb{E}_{q(\boldsymbol{\mu})}\boldsymbol{\mu}$, картинки вставьте в отчет. Как вы можете проинтерпретировать полученные результаты? Почему получились именно такие центры кластеров?

7. Исследовать зависимость логарифма правдоподобия на обучающей и контрольной выборках от кластеризации. Правдоподобие вычислять по формуле

$$p(\mathbf{X}) = \prod_{n=1}^{N} p(\mathbf{x}_n \mid \mathbb{E}_{q(\boldsymbol{\mu})}\boldsymbol{\mu}, \boldsymbol{\pi}_{ML}),$$

   где $p(\mathbf{x} \mid \boldsymbol{\mu}, \boldsymbol{\pi})$ задано формулой (2), $\boldsymbol{\pi}_{ML}$ — точечная оценка на параметры $\boldsymbol{\pi}$, полученная в результате работы EM-алгоритма.

8. Рассмотреть величины $q(z_{nk} = 1)$ в качестве признаков n-го объекта. Обучить любой классификатор на базе MNIST. Исследовать как ведет себя матрица точности на контрольной выборке в зависимости от кластеризации.

9. Написать отчет в формате pdf с описанием всех проведенных исследований. В отчете также должен содержаться вывод всех формул (набранных в системе LaTeX).

   При оценке выполнения задания будет учитываться эффективность программного кода. Так, на полной базе MNIST одна итерация алгоритма должна укладываться в 10 секунд.

Формат данных MNIST

Подготовленные бинаризованные данные MNIST можно скачать по ссылке http://yadi.sk/d/4Ox_Q0q_JnBsU. Это csv файл с 60000 строк, в каждой строке записано бинарное изображение цифры от 0 до 9 и метка класса. Каждое изображение имеет размер 28 на 28 пикселей, соответственно в строке записано 785 чисел. (i, j) пиксель изображения записан в позиции $28i + j$, нумерация с нуля. Последнее число в строке — метка класса.

3 Рекомендации по выполнению задания

1. Функционал $\mathcal{L}(q)$ *должен* возрастать с течением итераций. Если это не так, в реализации или в выводе формул ошибка. Это хороший способ отладки вашей программы.

2. Для ускорения можно вычислять функционал $\mathcal{L}(q)$, например, раз в 10 итераций.

3. Для того, чтобы избежать проблем с точностью вычислений, следует везде, где это возможно, переходить от произведений к суммированию логарифмов.

4. Для эффективной реализации следует избегать использования циклов.

5. В исследовательской части задания предлагается рассматривать такие значения параметров:

   - $a = b \in (0, \infty), \alpha \in (0, 1)$,
   - $a \in (0, 1), b = 1, \alpha \in (0, 1)$.

6. При реализации на языке Python стоит использовать библиотеку numpy. Для подсчета дигамма, гамма, и логарифма гамма функции можно использовать библиотеку scipy (модуль `scipy.special`, функции `psi`, `gamma`, `gammaln` соответственно). В MATLAB эти функции также есть и имеют такие же названия.

4 Спецификация

Python

Вы должны предоставить файл, в котором будет реализована функция
`EM(X, a=1, b=1, alpha=0.001, K=50, max_iter=500, tol=1e-3, n_start=1)`

- `X` — переменная типа `numpy.array`, матрица размера $N \times D$, наблюдаемые бинарные переменные $\mathbf{X}$,
- `a`, `b` — параметры априорного Бета распределения,
- `alpha` — параметр априорного распределения Дирихле,
- `K` — число компонент K,
- `max_iter` — максимальное число итераций,
- `tol` — точность оптимизации по $\mathcal{L}(q)$,
- `n_start` — число запусков из различных случайных начальных приближений.

Функция должна возвращать словарь с ключами `'pi'`, `'mu'`, `'L'`:

- `'pi'` — оценка параметров $\boldsymbol{\pi}$, `numpy.array`, матрица размера $K \times 1$,
- `'mu'` — величина $\mathbb{E}_{q(\boldsymbol{\mu})}\boldsymbol{\mu}$, `numpy.array`, матрица размера $K \times D$,
- `'L'` — значения функционала $\mathcal{L}(q)$ по итерациям для лучшего начального приближения, список или `numpy.array`.

Следует выбрать оценки параметров для лучшего начального приближения.
В файле также должна быть реализована функция для вычисления $\log p(\mathbf{X})$:
`log_likelihood(X, mu, pi)`

- `X` — переменная типа `numpy.array`, матрица размера $N \times D$, наблюдаемые бинарные переменные $\mathbf{X}$,
- `mu` — параметры распределений Бернулли $\boldsymbol{\mu}$, `numpy.array`, матрица размера $K \times D$,
- `pi` — параметры смеси $\boldsymbol{\pi}$, `numpy.array`, матрица размера $K \times 1$.

Функция должна возвращать число.

MATLAB

Вы должны предоставить файл `EM.m`, в котором реализована функция
`[pi, mu, L] = EM(X, options)`

- `X` — матрица размера $N \times D$, наблюдаемые данные $\mathbf{X}$,
- `options` — структура с полями `'a'`, `'b'`, `'alpha'`, `'K'`, `'max_iter'`, `'tol'`, `'n_start'`, описание полей см. в предыдущем пункте. Если какое-то из полей не задано, нужно считать, что параметр имеет значение по умолчанию, как в предыдущем пункте.

Выход алгоритма:

- `pi` — оценка параметров $\boldsymbol{\pi}$, матрица размера $K \times 1$,
- `mu` — величина $\mathbb{E}_{q(\boldsymbol{\mu})}\boldsymbol{\mu}$, матрица размера $K \times D$,
- `L` — значения функционала $\mathcal{L}(q)$ по итерациям, для лучшего начального приближения.

Следует выбрать оценки параметров для лучшего начального приближения.
Также вы должны предоставить файл `log_likelihood.m`, в котором реализована функция для вычисления $\log p(\mathbf{X})$:
`LL = log_likelihood(X, mu, pi)`

- `X` — матрица размера $N \times D$, наблюдаемые бинарные переменные $\mathbf{X}$,
- `mu` — параметры распределений Бернулли $\boldsymbol{\mu}$, матрица размера $K \times D$,
- `pi` — параметры смеси $\boldsymbol{\pi}$, матрица размера $K \times 1$.

5 Оформление задания

Выполненное задание следует отправить письмом по адресу bayesml@gmail.com с заголовком письма

«[БММО14] Задание 2, Фамилия Имя».

Убедительная просьба присылать выполненное задание только один раз с окончательным вариантом. Также убедительная просьба строго придерживаться заданных прототипов реализуемых функций (для проверки задания используются, в том числе, автоматические процедуры, которые являются чувствительными к неверным прототипам).

Присланный вариант задания должен содержать в себе:

- Текстовый файл в формате PDF с указанием ФИО, содержащий описание всех проведённых исследований.
- Все исходные коды с необходимыми комментариями.