Компьютерные задания по введению в матметоды распознавания образов и компьютерного зрения

1 Байесовская стратегия для бинарной штрафной функции

Задача

На вход программе посредством WebSocket подаются бинарные зашумлённые изображения известных эталонов. Шум — набор независимых одинаково распределённых случайных величин с распределением Бернулли и параметром $0 \leq p \leq 1$, который пользователь определяет самостоятельно. Шум накладывается попиксельно с помощью исключающего "ИЛИ". Минимизировать риск байесовской стратегии для штрафной функции $\omega_1(k, k') = \mathbb{1}(k \neq k')$.

Цель

Закрепить навыки максимизации апостериорной вероятности.

Задание

Задание First из `https://sprs.herokuapp.com`. Программа должна работать корректно при

1. Уровне шума 0 (без шума все ответы должны быть верны).
2. Уровне шума 1 (при полной инверсии все ответы должны быть верны).
3. Уровне шума 0.4 и меньше (а также 0.6 и больше) и масштабе 20 на 20 и больше: при размере картинки 100 на 60 крайне мала вероятность превращения одной цифры в другую посредством наложения такого уровня шума, поэтому ответ почти всегда должен быть верным.

2 Байесовская стратегия для интервальной штрафной функции

Задача

На вход программе посредством WebSocket подаются ненормированные значения $p(k \mid x)$. Минимизировать риск байесовской стратегии для штрафной функции $\omega_1(k, k') = \mathbb{1}(|k - k'| > d)$. Мощность множества $K = \{0, 1, \ldots, |K| - 1\}$ и число d определяет пользователь.

Цель

Закрепить понимание основ построения байесовских стратегий распознавания.

Задание

Задание Second из `https://sprs.herokuapp.com`.

3 Байесовская стратегия для штрафной функции L_1

Задача

На вход программе посредством WebSocket подаются ненормированные значения $p(k \mid x)$. Минимизировать риск байесовской стратегии для штрафной функции $\omega_1(k, k') = |k - k'|$. Мощность множества $K = \{0, 1, \ldots, |K| - 1\}$ определяет пользователь.

Цель

Усвоить на практике, что максимизация апостериорной вероятности не во всех ситуациях является оптимальной стратегией.

Задание

Задание Third из `https://sprs.herokuapp.com`.

4 Анализ небайесовской стратегии

Анализируемая задача

Минимизировать риск стратегии для штрафной функции $\omega_1(k, k') = (k - k')^2$, не давая риску для штрафной функции $\omega_2(k, k') = \mathbb{1}(k \neq k')$ превышать определённое значение ε.

Цель

Сравнить выведенную небайесовскую стратегию с оптимальными байесовскими стратегиями для штрафных функций ω_1 и ω_2 (считать "в лоб").

Возможный ход работы

Важная рекомендация: сначала ручка и бумакка, а потом код. Также после следующих пунктов приведены подсказки. Рекомендую дочитать до них перед выполнением работы.

1. Написать генератор распределений $p(k \mid x)$ на n значений (число n определяет пользователь) — просто генерируете n чисел и делите каждую ячейку на их исходную сумму

2. Написать генератор чисел из определённого пользователем набора K из n чисел ($|K| = n$) и соответствующего распределения $p(k \mid x)$

3. Реализовать оптимальные стратегии путём полного перебора:

   - для квадратичной функции потерь ω_2
   - для бинарной функции потерь ω_1
   - для анализируемой небайесовской задачи (не забудьте, что она требует параметр, который в аудитории назвали α)

4. Реализовать функции, которые считают приблизительный частный риск $R(q; x) = \sum\limits_{k \in K} \omega(k, q(x)) \cdot p(k \mid x)$ для двух функций потерь (бинарная ω_1 и квадратичная ω_2) с помощью метода Монте-Карло для произвольной стратегии q (которую можно передавать в качестве параметра)

5. Запустить функцию приблизительного подсчёта риска $R(q) = \sum\limits_{x \in X} p(x) \cdot \sum\limits_{k \in K} \omega(k, q(x)) \cdot p(k \mid x)$ на разных распределениях $p(k \mid x)$, а результат усреднить (считаем, что каждая гистограмма равновероятна, снова Монте-Карло)

6. Посмотреть, при каких значениях параметра α выведенная стратегия решения анализируемой небайесовской задачи работает как детерминированная для бинарной функции потерь ω_1, а при каких как детерминированная для квадратичной функции потерь ω_2

7. Понаблюдать, как ведут себя риски бинарной $R_1(q)$ и квадратичной $R_2(q)$ функции потерь при параметрах α, которые находятся между найденными в предыдущем пункте значениями. Объяснить результаты

8. Посмотреть, что будет, если α будет отрицательным. Объяснить результат

Для генерации чисел по гистограмме можно использовать следующий подход: генерируете число из $[0; 1)$, считаете cumulative sum для гистограммы, и выбираете первую из тех ячеек гистограммы, значение в которой больше сгенерированного числа. Или же ищете в поисковике, как можно генерировать числа по заданному распределению (например, в numpu).

Возможное применение [урезанного] метода Монте-Карло для этой задачи:

- на вход подаётся набор чисел, гистограмма и стратегия
- внутри генерируется несколько сотен тысяч или миллионов чисел из набора согласно гистограмме
- считаются штрафы, умножаются на соответствующие им вероятности, и складываются
- на выход даётся полученное значение

5 Быстрый подсчёт суммы подмассива

Анализируемая задача

Имеется n-мерный целочисленный массив A (n принимает значения 1, 2 или 3). На вход даётся два n-мерных набора целочисленных индексов ℓ и r. Нужно посчитать значение выражения

$$\sum_{\substack{\ell_i \le x_i \le r_i \\ i=\overline{1,n}}} A_x,$$

Вывести эффективный алгоритм подсчёта необходимой суммы посредством предварительных расчётов и формулы включения-исключения.

Цель

Сравнить алгоритм подсчёта "в лоб" с алгоритмом, который использует предварительные расчёты теоретически. Проверить на практике полученные результаты.

Возможный ход работы

1. Реализовать функцию подсчёта суммы подмассива "в лоб"
2. Реализовать функцию расчёта вспомогательных значений
3. Реализовать функцию подсчёта суммы подмассива с помощью вспомогательных значений
4. Проверить корректность реализованной функции путём сравнения её результатов с результатами метода подсчёта "в лоб"
5. Проверить на практике теоретическую сложность выведенного алгоритма
6. Проверить на практике теоретическую сложность метода "в лоб" по относительно выбранного параметра (например, зависимость от длины подмассива или площади подматрицы)

Проверка сложности на практике — это эксперимент. Поставить его можно следующим образом:

1. Выяснить, как сложность зависит от размеров подмассива и исходного массива теоретически (включая сложность этапа подсчёта вспомогательных значений)
2. Для сотни (а лучше тысячи) разных значений этих размеров подмассива замерить время выполнения алгоритма; для качественных замеров нужно несколько раз повторить следующую процедуру (необходимое количество раз можно понять по среднеквадратическому отклонению)
   - Сделать предвычисления (не нужно для алгоритма "в лоб")
   - Посчитать суммы подмассивов фиксированной длины (количество выбирайте такое, чтобы время их выполнения было заметным и не подавлялось временем на использование таймера полностью)
3. При желании постройте интерполяцию, и сделайте вывод, похожи ли теоретические ожидания на практические результаты

6 Определение делимости

Задача

На вход подаются бинарные изображения чисел состоящих из t цифр $X = \{0,1\}^{h \times w \cdot t}$, где h — количество строк (высота изображения), w — количество столбцов в изображении цифры (ширина изображения одной цифры), $w \cdot t$ — количество столбцов в картинке (ширина изображения числа), количество цифр $t = 21$. Каждая цифра k_i числа k генерируется независимо по распределению $p(k_i)$. Распределение цифр не зависит от их позиции в числе. Априорное распределение цифр может быть неравномерным. Эталонные изображения каждой цифры известны (нарисуйте их сами, возьмите из задания для первого раздела или найдите где-либо). Каждый пиксель картинки независимо друг от друга зашумляется с помощью шума Бернулли с известным параметром $0 \le p \le 1$. Рекомендуется начать с небольших изображений (площадью до 20 пикселей), но итоговая программа должна уметь разбираться с картинками в несколько [десятков] килопикселей (чтобы даже `float128` не помог). Множество решений $D = \{\text{делится нацело}, \text{делится с остатком}\}$, где речь идёт о делимости числа k на число m, которое зависит от варианта. Штрафная функция

$$\omega(k,d) = \mathbb{1}(k \bmod m > 0) \cdot \mathbb{1}(d = \text{делится нацело}) + \mathbb{1}(k \bmod m = 0) \cdot \mathbb{1}(d = \text{делится с остатком}).$$

То есть, неправильный ответ даёт штраф 1.

Есть два варианта: $m = \langle 3, 11 \rangle$. Если в бригаде присутствует два человека из разных вариантов, они выбирают любой вариант и выполняют его. Список людей и их чисел для выполнения задания:

Цель

Освоить на практике основы работы со скрытыми марковскими моделями.

Возможный ход работы

1. Написать генератор последовательности случайных цифр по предъявленным априорным вероятностям (или использовать готовый со второго раздела).

2. Написать генератор идеальных изображений по последовательности цифр.

3. Написать зашумление идеальных изображений чисел.

4. Реализовать подсчёт вероятностей всех возможных [знакочередующихся] сумм цифр с помощью динамического программирования.

5. Сделать вывод о делимости на m.

7 Бинокулярное стереоскопическое зрение

Требования к приложению

Должна быть возможность передавать путь к левому изображению, путь к правому изображению, максимальный сдвиг и параметр сглаживания посредством аргументов командной строки. Обработка изображений размером $1'000 \times 1'000$ с максимальным сдвигом 50 должна идти не дольше пяти минут требовать не более 2ГБ оперативной памяти.

Задача

Построить карту сдвигов для пары изображений полученных идеальной проективной фотокамерой (без радиальных искажений) посредством её идеального перемещения вправо: пикселям в строке r левого изображения соответствуют пиксели лишь в строке r правого изображения.

Рассматривается простой случай, когда разные строки изображения не зависят друг от друга, и карты (строки) сдвигов для них можно искать по отдельности. Для каждой строки $y \in \{1, \dots, height\}$ изображения

$$L : \{1, \dots, height\} \times \{1, \dots, width\} \to C$$

нужно найти такую функцию

$$d : \{1, \dots, width\} \to \{0, \dots, \max D\},$$

которая минимизирует штрафную функцию

$$E(d, y) = \sum_{x=1}^{width} h\left(L(y, x), R(y, x - d(x))\right) + \alpha \cdot \sum_{x=1}^{width-1} g\left(d(x), d(x+1)\right),$$

где $R : \{1, \dots, height\} \times \{1, \dots, width\} \to C$ — правое изображение, а $\alpha \geq 0$ — параметр сглаживания, который выбирается пользователем. У каждого варианта свои штрафные функции h и g, которые составляются из различных сочетаний норм L_1 и L_2. Список людей и их штрафных функций для выполнения задания:

Цель

Изучить основы определения формы трёхмерных объектов по их изображениям.

Возможный ход работы

Один из сайтов с изображениями для экспериментов `http://vision.middlebury.edu/stereo/data/`.

1. Изучить наборы изображений на сайте колледжа Мидлбери: какие они есть, чем отличаются, какие форматы изображений, какие максимальные значения сдвигов между изображениями;

2. Загрузить несколько наборов изображений с сайта колледжа Мидлбери;

3. Реализовать алгоритм поиска кратчайшего пути на графе для задачи стереоскопического зрения;

4. Построить граф для решения задачи стереоскопического зрения и найти кратчайший путь на нём (список сдвигов) для каждой строчки левого изображения;

5. Сохранить карту сдвигов в виде чёрно-белого изображения: чем больше сдвиг, тем светлее пиксель;

6. Выяснить, при каких параметрах сглаживания карта сдвигов наиболее корректная для определённых изображений.

Конкретные значения глубин искать необязательно — достаточно найти карту сдвигов.

8 Кластеризация изображений

Задача

Исходя из предположения о том, что бинарные изображения моделируются набором независимых случайных величин распределённых по закону Бернулли с разными параметрами p для разных пикселей, построить и реализовать EM-алгоритм кластеризации двух классов изображений из MNIST или EMNIST на выбор.

`http://blog.manfredas.com/expectation-maximization-tutorial/`

Цель

Закрепить на практике EM-алгоритм самообучения.

Возможный ход работы

1. Выбрать два визуально разных класса изображений (например, цифры 0 и 1 или буквы V и S).

2. Реализовать EM-алгоритм для решения поставленной задачи.

3. Запустить построенный алгоритм на тренировочной выборке для выбранных классов.

4. Проверить результат кластеризации на тестовой выборке.

9 Поиск разделяющей окружности

Задача

На плоскости находится окружность неизвестного конечного радиуса с неизвестным центром. Генерируется набор точек X_1, которые находятся внутри окружности, а также набор точек X_2, которые находятся снаружи её. Ни одна точка не лежит на границе окружности. Модифицировать персептрон Розенблатта таким образом, чтобы он нашёл такую окружность, которая отделяет точки множества X_1 от точек множества X_2.

Цель

Закрепить на практике алгоритм обучения персептрона и освоить навыки искривления пространства для поиска разделяющих кривых с помощью линейного классификатора.

Возможный ход работы

1. Реализовать функцию, которая генерирует центр и радиус окружности.

2. Реализовать функцию, которая генерирует определённое пользователем количество точек на плоскости.

3. Реализовать функцию, которая принимает на вход параметры окружности и набор точек, а на выход даёт два набора точек: внешние точки и внутренние; при этом функция должна выбрасывать те точки, которые находятся к границе окружности слишком близко (порог ε, который определяет пользователь).

4. Реализовать алгоритм на основе персептрона, который принимает на вход два набора точек на плоскости, а на выход даёт параметры окружности, которая разделяет два множества точек; у пользователя должна быть возможность ввести максимальное количество коррекций, чтобы алгоритм гарантированно оканчивал свою работу за конечное время.