

МАШИННЕ НАВЧАННЯ ДЛЯ РОЗПІЗНАВАННЯ ОБ'ЄКТІВ НА БАЗІ БІБЛІОТЕКИ OPENCV

*Могильний С. Б., к.т.н., доцент; Задорожний Г. С., студент
Національний технічний університет України
«Київський політехнічний Інститут імені Ігоря Сікорського»,
м. Київ, Україна*

OpenCV — бібліотека, створена для використання у реальному часі, розвитком якої займається корпорація Intel. Бібліотека є кросплатформенною та безкоштовною для використання у межах ліцензії BSD для відкритих ресурсів. Метою дослідження є використання методів даної бібліотеки разом з машинним навчанням для розпізнавання об'єктів.

У роботі використана версія OpenCV 3.4.0, яка є останньою на даний час. Розглянуті 3 методи розпізнавання, а саме:

- k-Nearest Neighbour (kNN)
- Support Vector Machines (SVM)
- k-Means Clustering

kNN – один із найпростіших алгоритмів класифікації доступний для контрольованого навчання, ціль якого класифікувати об'єкт відносно к-того найближчого заданого елемента.

Наприклад, є два типи будівель (рис. 1): сині квадрати та червоні трикутники. Кожен тип будівлі будемо називати **Клас (Class)**. Карта, на якій зображене розташування будівель, це простір характеристик. (У нашому випадку розглядаємо двовимірні координати X та Y, але цей простір може бути N-вимірним, залежно від характеристик об'єкту). Додаємо новий клас — зелене коло. Процес визначення класу нового елемента називається **Класифікація (Classification)**. Залежно від k новий елемент можна віднести до того чи іншого класу, тому кожному елементу надаємо вагу, залежно від відстані до нового класу. Чим менша відстань — тим менша вага. Знаходимо вагу для кожного класу, додаючи вагу всіх елементів. Класифікуємо елемент відносно більшої ваги.

SVM має методи: Linearly Separable Data та Non-Linearly Separable Data.

У Linearly Separable Data знаходять пряму $f(x) = ax_1 + bx_2 + c$, яка ділить дані на два регіони (рис.2). Ця лінія називається **Границя Рішення (Decision Boundary)**. Дані, які можна розділити прямою лінією називаються

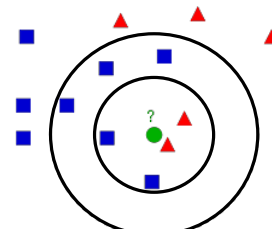


Рисунок 1. Алгоритм kNN

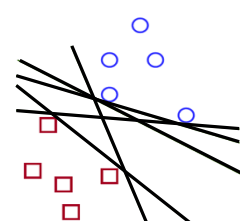


Рисунок 2. Алгоритм SVM

Лінійно Подільні (Linear Separable). Новий елемент стає простіше класифікувати оскільки маємо усього два варіанти (для більшої кількості параметрів використовуються гіперплощини (hyperplanes)), які розділені лінією.

Наприклад, сині значення представлені як $w^T x + b_0 > 1$, а червоні $w^T x + b_0 < 1$. Де w (**weight vector**) — це вектор ваги $w = [w_1, w_2, \dots, w_n]$, x (**feature vector**) — вектор характеристики $x = [x_1, x_2, \dots, x_n]$ та b_0 (**bias**) — зміщення. Вектор ваги визначає орієнтацію границі рішення, у той час як точка зміщення визначає розташування. Границю рішення можна визначити як середину між гіперплощинами (рис. 3), отже вона записується як $w^T x + b_0 = 1$. Мінімальна відстань від допоміжного вектору до границі рішення

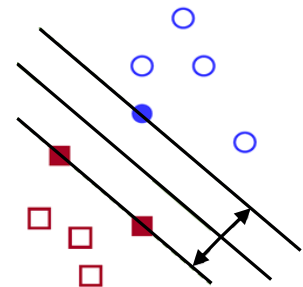


Рисунок 3. Визначення границі рішення

$distance_{support\ \dots\ vectors} = \frac{1}{\|w\|}$. Треба вибрати найбільшу

відстань до класів, тобто мінімізувати функцію $L(w, b_0)$:

$$\min_{w, b_0} L(w, b_0) = \frac{1}{2} \|w^T\| \text{ subject to } t_i(w^T x + b_0) \geq 1 \forall i$$

де t_i — позначка кожного класу $t_i \in [-1, 1]$.

k-Means Clustering. Метод має досить складне математичне пояснення, тому буде представлений поверхневий опис.

Наприклад, компанія виготовляє футболки. Будується зібрана статистика про розміри (рис. 4), та, відповідно ділиться на області, щоб визначити обсяги виробництва.

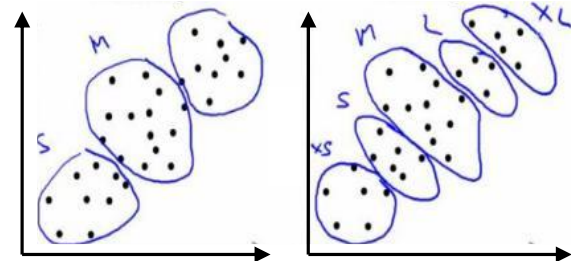


Рисунок 4. Алгоритм k-Means Clustering

Потрібно розділити два типи даних (рис. 5).

Крок 1: випадковим чином вибираємо два центри

Крок 2: розрахунок відстані від кожної точки до центрів. Дані, ближчі до C_1 позначаються «0», до C_2 — «1». (більше центрів — більше позначень). Всі 0 позначимо червоним, 1 — синім (рис. 6)

Крок 3: Знаходимо середнє значення усіх синіх точок окремо, щоб знайти нове

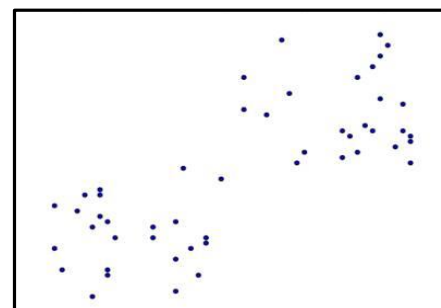


Рисунок 5. Приклад розділення двох типів даних

положення центру. Потім знову крок 2, щоб отримати нові центри з позначенням даних 0 та 1. (рис. 7)

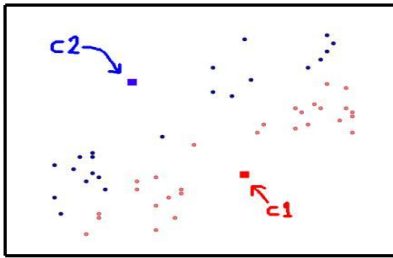


Рисунок 6. Розрахунок відстані від точок до центру

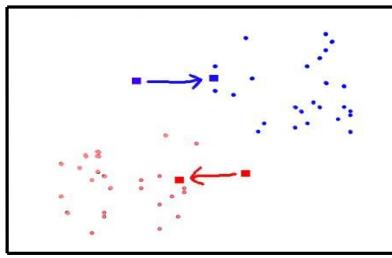


Рисунок 7. Нові центри для даних 0 і 1

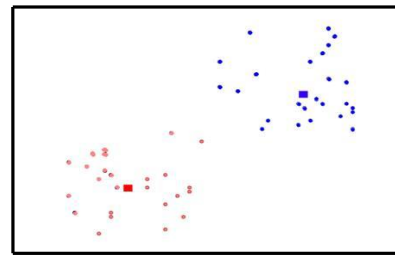


Рисунок 8. Фіксація положень центрів

$$\min\left[\sum_{All\ Red_Points} distance(C1, Red_Point) + \sum_{All\ Blue_Points} distance(C2, Blue_Point)\right]$$

Повторюємо крок 2 та крок 3 поки обидва центри (рис. 8) не зафіксують свої положення (це можна зупинити, поставивши певну кількість ітерацій, або до певної точності). Ці точки – це мінімум суми відстаней між даними та відповідними місцями центрів.

Цей метод має багато модифікації, як наприклад, спеціальний вибір центрів, або збільшення швидкості ітерацій кроків 2 та 3.

Перелік посилань

1. OpenCV 3.4.1-dev. Open Source Computer Vision. – Режим доступу: – <https://docs.opencv.org/master/> Назва з екрану.

Анотація

Запропоновані методи класифікації об'єктів на базі бібліотеки OpenCV (Open Source Computer Vision). Тестування методів виконане на задачі розпізнавання фруктів. Зроблено порівняння точності кожного метода при однакових розмірах масивів даних, використаних для навчання.

Ключові слова: OpenCV, розпізнавання.

Аннотация

Предложены методы классификации объектов на базе библиотеки OpenCV (Open Source Computer Vision). Тестирование методов выполнено на задаче распознавания фруктов. Сделано сравнение точности каждого метода при одинаковых размерах массивов данных, использованных для обучения.

Ключевые слова: OpenCV, распознавание.

Abstract

Methods for classifying objects on the basis of the OpenCV library (Open Source Computer Vision) are proposed. Testing methods is performed on the problem of fruit recognition. A comparison is made between the accuracy of each method with the same size of the data sets used for training.

Keywords: OpenCV, recognition.