

阿洲的程式教學

關於Qt、OpenCV、影像處理演算法

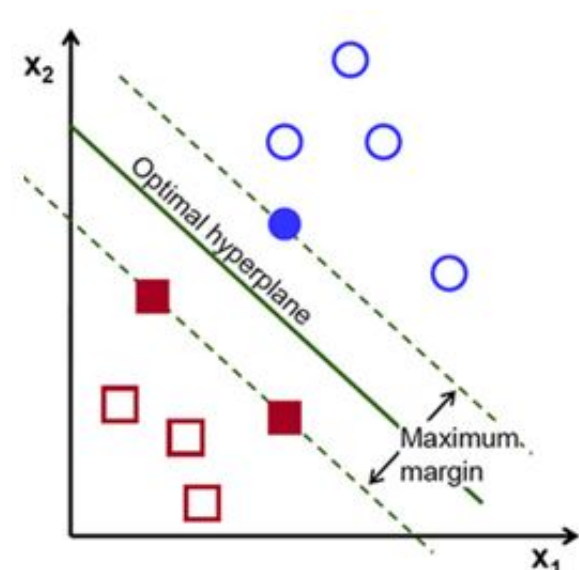
支撐向量機(SVM)

機器學習(Machine Learning)主要是設計算法，讓電腦能透過資料而有像人類的學習行為，算法通常是自動分析數據，獲得規律，並利用規律對未知數據進行預測，進而達到分類、回歸分析等目的，在影像處理上則可能是影像辨識。

依據輸入資料是否有標籤，我們分監督式學習和非監督式學習，資料有標籤的為監督式學習，沒有標籤的為非監督式學習，舉例來說，假如輸入臉的輪廓，輪廓本身沒有標籤，但加入每個輪廓年齡多少這個資料就是標籤。

這邊介紹支撐向量機SVM(Support Vector Machine)，這是一種監督式的機器學習算法，原先用於二元分類，比如說這封郵件是否為垃圾郵件，或是這個人是男是女，這種二個類別的問題，但現在已擴展且廣泛應用於統計分類和回歸分析。

SVM建構多維的超平面來分類資料點，這個超平面即為分類邊界，直觀來說，好的分類邊界要距離最近的訓練資料點越遠越好，因為這樣可以減低判斷錯誤的機率，而SVM的目標即為找出間隔最大的超平面來作為分類邊界，下面為SVM的示意圖，綠線為分類邊界，分類邊界與最近的訓練資料點之間的距離稱為間隔(margin)。



以下我們示範OpenCV SVM的使用方式，大概可分以下幾個步驟：

1. 在空間中選擇六個點作為輸入資料。
2. 給這些點相對的標籤，對輸入資料進行分類。
3. 設置CvSVMParams作為SVM的參數。
4. 將資料和參數輸入SVM::train()，進行訓練後即可求得分類邊界。
5. 之後可輸入新的資料，由SVM::predict()看此筆資料屬於哪一類。

以下為實際程式碼：

```
# include <cstdio>
# include <opencv2/opencv.hpp>
# include <vector>
using namespace cv;

int main(){
    int width = 300;
    int height = 300;
    Mat image = Mat::zeros(height, width, CV_8UC3);

    float trainingData[6][2] = {{250,250},{200,100},{260,180},{140,10},
    {30,70},{50,50}};
    Mat trainingDataMat(6, 2, CV_32FC1, trainingData);

    float labels[6] = {1.0, 1.0, 1.0, -1.0, -1.0, -1.0};
    Mat labelsMat(6, 1, CV_32FC1, labels);

    CvSVMParams params;
    params.svm_type = CvSVM::C_SVC;
    params.kernel_type = CvSVM::LINEAR;
    params.term_crit = cvTermCriteria(CV_TERMCRIT_ITER, 100, 1e-6);

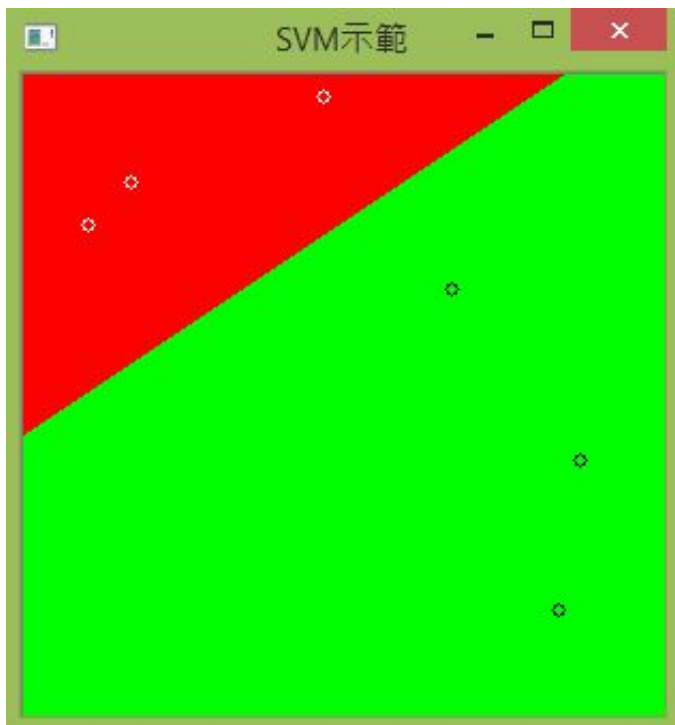
    CvSVM SVM;
    SVM.train(trainingDataMat, labelsMat, Mat(), Mat(), params);
    Vec3b green(0,255,0), red (0,0,255);
    for (int i=0; i<image.rows; ++i){
        for (int j=0; j<image.cols; ++j){
            Mat sampleMat = (Mat_<float>(1,2) << j,i);
            float response = SVM.predict(sampleMat);

            if(response == 1){
                image.at<Vec3b>(i,j)=green;
            }
            else if(response == -1){
                image.at<Vec3b>(i,j)=red;
            }
        }
    }
    circle(image, Point(250, 250), 3, Scalar(0, 0, 0));
```

```
circle(image, Point(200, 100), 3, Scalar(0, 0, 0));
circle(image, Point(260, 180), 3, Scalar(0, 0, 0));
circle(image, Point(140, 10), 3, Scalar(255, 255, 255));
circle(image, Point(30, 70), 3, Scalar(255, 255, 255));
circle(image, Point(50, 50), 3, Scalar(255, 255, 255));

imshow("SVM示範", image);
waitKey(0);

return 0;
}
```



有時因為資料的關係，無法取得完美的分類邊界，以下示範如何用**SVM**取得相對好的分類邊界，使用方式和上述例子差不多：

```
#include <stdio>
#include <opencv2/opencv.hpp>
#include <vector>
using namespace cv;

int main(){
    int width = 300;
    int height = 300;
    Mat I = Mat::zeros(height, width, CV_8UC3);
    Mat trainData(100, 2, CV_32FC1);
    Mat labels (100, 1, CV_32FC1);

    //設100個隨機點
```

```

RNG rng;
for(int i=0; i<50; i++){
    labels.at<float>(i,0) = 1.0;
    int tempY = rng.uniform(0,299);
    int tempX = rng.uniform(0,170);
    trainData.at<float>(i,0) = tempX;
    trainData.at<float>(i,1) = tempY;
}
for(int i=50; i<99; i++){
    labels.at<float>(i,0) = -1.0;
    int tempY = rng.uniform(0,299);
    int tempX = rng.uniform(130,299);
    trainData.at<float>(i,0) = tempX;
    trainData.at<float>(i,1) = tempY;
}

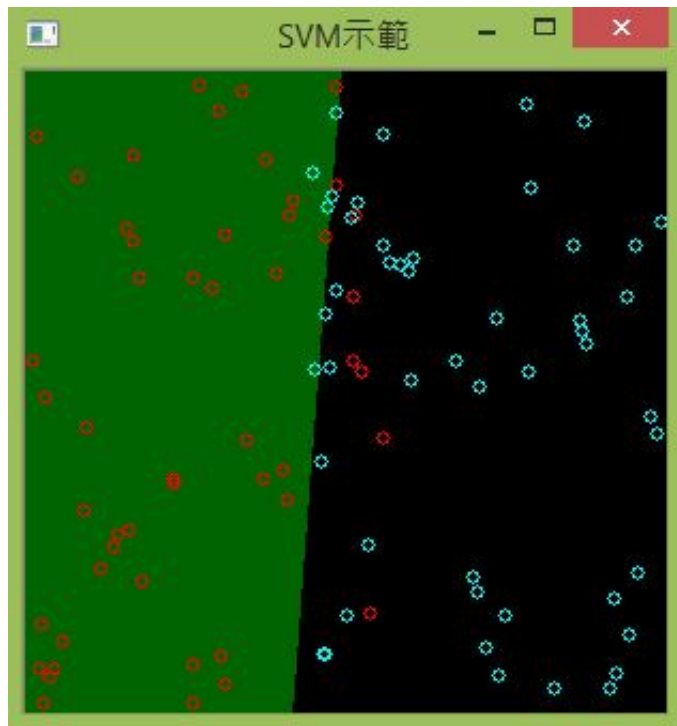
CvSVMParams params;
params.svm_type = SVM::C_SVC;
params.C = 0.1;
params.kernel_type = SVM::LINEAR;
params.term_crit = TermCriteria(CV_TERMCRIT_ITER, (int)1e7, 1e-6);
CvSVM svm;
svm.train(trainData, labels, Mat(), Mat(), params);

Vec3b green(0,100,0), blue (100,0,0);
for (int i = 0; i < I.rows; ++i){
    for (int j = 0; j < I.cols; ++j){
        Mat sampleMat = (Mat_<float>(1,2) << i, j);
        float response = svm.predict(sampleMat);
        if(response == 1){
            I.at<Vec3b>(j, i)=green;
        }
        else if (response == 2){
            I.at<Vec3b>(j, i)=blue;
        }
    }
}

float px, py;
for (int i=0; i<50; ++i){
    px = trainData.at<float>(i,0);
    py = trainData.at<float>(i,1);
    circle(I, Point((int)px, (int)py), 3, Scalar(0, 0, 255));
}
for (int i=50; i<100; ++i){
    px = trainData.at<float>(i,0);
    py = trainData.at<float>(i,1);
    circle(I, Point((int)px, (int)py), 3, Scalar(255, 255,0));
}

imshow("SVM示範", I);
waitKey(0);
return 0;
}

```



[回到首頁](#)

[回到OpenCV教學](#)

參考資料：

[OpenCV 教程](#)

📅 2016-01-03 👤 阿宅 📁 OpenCV, 特徵與機器學習 🔖 SVM, 支撐向量機

0 Comments

猴子遇到0與1! 程式學習筆記

 Login ▾ Recommend Share

Sort by Best ▾



Start the discussion...

Be the first to comment.

ALSO ON 猴子遇到0與1! 程式學習筆記

Qt主窗口(Top Level Window)

1 comment • 6 months ago

mike — 喔喔

文件對話框(QFileDialog)

1 comment • 6 months ago

楊政穎 — dialog.cpp 裡面的 QString s
=
QFileDialog::getOpenFileName(this, tr

 Subscribe Add Disqus to your site Add Disqus Add Privacy

自豪的採用 WordPress