**OpenCV3中EM聚类算法的使用**

EM算法OpenCV3机器学习模块中新增加算法，和其他算法一样，使用十分简单。

1、  创建EM算法类对象，类似于以下代码：

Ptr<ml::EM> em\_model = ml::EM::create();

2、  设置算法相关参数

em\_model->setClustersNumber(5);

em\_model->setCovarianceMatrixType(ml::EM::COV\_MAT\_SPHERICAL);

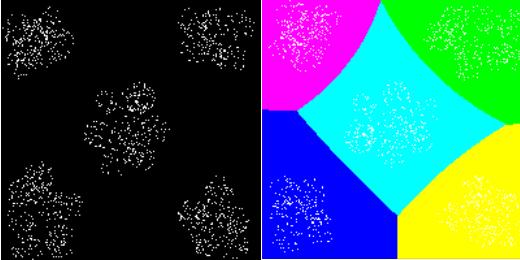
em\_model->setTermCriteria(TermCriteria(TermCriteria::COUNT + TermCriteria::EPS, 300, 0.1));

3、构造训练数据矩阵，必须为CV\_32FC1类型，每行为一个样本。

4、调用trainEM训练

5、使用predict测试

以下程序说明了该过程，下图中第一幅图像用于生成训练数据，第二幅图像为分类结果。



例程：

#include "stdafx.h"

#include <opencv2\opencv.hpp>

using namespace std;

using namespace cv;

Mat cluster;

Mat makeSamples()

{

    Mat binary;

    vector<float> data;

    threshold(cluster, binary, 100, 255, CV\_THRESH\_BINARY);

    for (int i = 0; i < cluster.rows; i++)

    {

        uchar \*pData = cluster.ptr<uchar>(i);

        for (int j = 0; j < cluster.rows; j++)

        {

            if (\*(pData + j))

            {

                data.push\_back(float(i));

                data.push\_back(float(j));

            }

        }

    }

    Mat trainMat(data,true);

    trainMat = trainMat.reshape(1, data.size() / 2);

    return trainMat;

}

Scalar colorTable[6] = {

    Scalar(255,0,0),

    Scalar(255,255,0),

    Scalar(255,0,255),

    Scalar(0,255,255),

    Scalar(0,255,0),

    Scalar(0,0,255)

};

void setPoint(Mat img, Point pt, Scalar color)

{

    img.at<Vec3b>(pt) = Vec3b(color[0], color[1], color[2]);

}

int \_tmain(int argc, \_TCHAR\* argv[])

{

    cluster = imread("cluster.bmp", 0);

    Mat trainMat = makeSamples();

    Ptr<ml::EM> em\_model = ml::EM::create();

    em\_model->setClustersNumber(5);

    em\_model->setCovarianceMatrixType(ml::EM::COV\_MAT\_SPHERICAL);

    em\_model->setTermCriteria(TermCriteria(TermCriteria::COUNT + TermCriteria::EPS, 300, 0.1));

    //聚类训练

    em\_model->trainEM(trainMat);

    //预测

    Mat sample(1, 2, CV\_32FC1);

    Mat result(cluster.size(), CV\_8UC3, Scalar(255, 255, 255));

    for (int i = 0; i < cluster.rows; i++)

    {

        for (int j = 0; j < cluster.rows; j++)

        {

            sample.at<float>(0, 0) = (float)i;

            sample.at<float>(0, 1) = (float)j;

            int res = cvRound(em\_model->predict(sample));

            setPoint(result, Point(j, i), colorTable[res]);

        }

    }

    //绘制训练数据

    for (int i = 0; i < trainMat.rows; i++)

    {

        int x = trainMat.at<float>(i, 0);

        int y = trainMat.at<float>(i, 1);

        setPoint(result, Point(x, y), Scalar(255,255,255));

    }

    imshow("RESUTL", result);

    waitKey(0);

    return 0;

}