Machine Learning with OpenCV

转载 2015年07月31日 17:18:00 699

Machine Learning with OpenCV

By Philipp Wagner (http://bytefish.de/author/philipp-wagner.html) | May 25, 2010

Machine Learning is a branch of Artificial Intelligence and concerned with the question how to make machines able to learn from data. The core idea is to enable a machine to make intelligent decisions and predictions based on experiences from the past. Algorithms of Machine Learning require interdisciplinary knowledge and often intersect with topics of statistics, mathematics, physics, pattern recognition and more.

OpenCV (http://opencv.org/) (Open Source Computer Vision) is a library for computer vision and comes with a machine learning library for:

- Decision Trees
- Boosting
- Support Vector Machines
- Expectation Maximization
- Neural Networks
- ...

Finding simple examples to get started is difficult, so I wrote a document and a program for the C++ Machine Learning API of OpenCV. You can download it from my github account at:

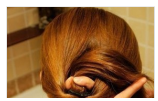
- https://github.com/bytefish/opencv (https://github.com/bytefish/opencv)

[cpp]

```
#include <iostream>
#include <math.h>
#include <string>
#include "cv.h"
#include "ml.h"
#include "highgui.h"

using namespace cv;
using namespace std;

```

```
bool plotSupportVectors = true;
int numTrainingPoints = 200;
int numTestPoints = 2000;
int size = 200;
int eq = 0;

// accuracy
float evaluate(cv::Mat& predicted, cv::Mat& actual) {
    assert(predicted.rows == actual.rows);
    int t = 0;
    int f = 0;
    for(int i = 0; i < actual.rows; i++) {
        float p = predicted.at<float>(i,0);
        float a = actual.at<float>(i,0);
        if((p >= 0.0 && a >= 0.0) || (p <= 0.0 &&  a <= 0.0)) {
            t++;
        } else {
            f++;
        }
    }
    return (t * 1.0) / (t + f);
}

// plot data and class
void plot_binary(cv::Mat& data, cv::Mat& classes, string name) {
    cv::Mat plot(size, size, CV_8UC3);
    plot.setTo(cv::Scalar(255.0,255.0,255.0));
    for(int i = 0; i < data.rows; i++) {

        float x = data.at<float>(i,0) * size;
        float y = data.at<float>(i,1) * size;

        if(classes.at<float>(i, 0) > 0) {
            cv::circle(plot, Point(x,y), 2, CV_RGB(255,0,0),1);
        } else {
            cv::circle(plot, Point(x,y), 2, CV_RGB(0,255,0),1);
        }
    }
    cv::imshow(name, plot);
}

// function to learn
int f(float x, float y, int equation) {
    switch(equation) {
    case 0:
        return y > sin(x*10) ? -1 : 1;
        break;
    case 1:
        return y > cos(x * 10) ? -1 : 1;
        break;
    case 2:
        return y > 2*x ? -1 : 1;
        break;
    case 3:
        return y > tan(x*10) ? -1 : 1;
        break;
    default:
        return y > cos(x*10) ? -1 : 1;
    }
}

// label data with equation
cv::Mat labelData(cv::Mat points, int equation) {
    cv::Mat labels(points.rows, 1, CV_32FC1);
    for(int i = 0; i < points.rows; i++) {
            float x = points.at<float>(i,0);
            float y = points.at<float>(i,1);
            labels.at<float>(i, 0) = f(x, y, equation);
```


三维姿态：关于solvePnP与cvPOSIT (http://blog.csdn.net/abc20002929/article/details/8520063)

透视变换矩阵(单应矩阵)计算：findHomography 与 getPerspectiveTransform (http://blog.csdn.net/abc20002929/article/details/8709902)

opencv背景建模mog2源码剖析 (http://blog.csdn.net/abc20002929/article/details/43247425)

opencv关于对比度和亮度的误解 (http://blog.csdn.net/abc20002929/article/details/40474807)

简单opencv人脸检测代码：LBP/Haar特征 (http://blog.csdn.net/abc20002929/article/details/23624287)

```
            labels.at<float>(i, 0) = f(x, y, equation);
        }
    return labels;
}

void svm(cv::Mat& trainingData, cv::Mat& trainingClasses, cv::Mat& testData, cv::Mat& testClass
    CvSVMParams param = CvSVMParams();

    param.svm_type = CvSVM::C_SVC;
    param.kernel_type = CvSVM::RBF; //CvSVM::RBF, CvSVM::LINEAR ...
    param.degree = 0; // for poly
    param.gamma = 20; // for poly/rbf/sigmoid
    param.coef0 = 0; // for poly/sigmoid

    param.C = 7; // for CV_SVM_C_SVC, CV_SVM_EPS_SVR and CV_SVM_NU_SVR
    param.nu = 0.0; // for CV_SVM_NU_SVC, CV_SVM_ONE_CLASS, and CV_SVM_NU_SVR
    param.p = 0.0; // for CV_SVM_EPS_SVR

    param.class_weights = NULL; // for CV_SVM_C_SVC
    param.term_crit.type = CV_TERMCRIT_ITER +CV_TERMCRIT_EPS;
    param.term_crit.max_iter = 1000;
    param.term_crit.epsilon = 1e-6;

    // SVM training (use train auto for OpenCV>=2.0)
    CvSVM svm(trainingData, trainingClasses, cv::Mat(), cv::Mat(), param);

    cv::Mat predicted(testClasses.rows, 1, CV_32F);

    for(int i = 0; i < testData.rows; i++) {
        cv::Mat sample = testData.row(i);

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

        predicted.at<float>(i, 0) = svm.predict(sample);
    }

    cout << "Accuracy_{SVM} = " << evaluate(predicted, testClasses) << endl;
    plot_binary(testData, predicted, "Predictions SVM");

    // plot support vectors
    if(plotSupportVectors) {
        cv::Mat plot_sv(size, size, CV_8UC3);
        plot_sv.setTo(cv::Scalar(255.0,255.0,255.0));

        int svec_count = svm.get_support_vector_count();
        for(int vecNum = 0; vecNum < svec_count; vecNum++) {
            const float* vec = svm.get_support_vector(vecNum);
            cv::circle(plot_sv, Point(vec[0]*size, vec[1]*size), 3 , CV_RGB(0, 0, 0));
        }
    cv::imshow("Support Vectors", plot_sv);
    }
}

void mlp(cv::Mat& trainingData, cv::Mat& trainingClasses, cv::Mat& testData, cv::Mat& testClass

    cv::Mat layers = cv::Mat(4, 1, CV_32SC1);

    layers.row(0) = cv::Scalar(2);
    layers.row(1) = cv::Scalar(10);
    layers.row(2) = cv::Scalar(15);
    layers.row(3) = cv::Scalar(1);

    CvANN_MLP mlp;
    CvANN_MLP_TrainParams params;
    CvTermCriteria criteria;
    criteria.max_iter = 100;
    criteria.epsilon = 0.00001f;
```

```
    criteria.type = CV_TERMCRIT_ITER | CV_TERMCRIT_EPS;
    params.train_method = CvANN_MLP_TrainParams::BACKPROP;
    params.bp_dw_scale = 0.05f;
    params.bp_moment_scale = 0.05f;
    params.term_crit = criteria;

    mlp.create(layers);

    // train
    mlp.train(trainingData, trainingClasses, cv::Mat(), cv::Mat(), params);

    cv::Mat response(1, 1, CV_32FC1);
    cv::Mat predicted(testClasses.rows, 1, CV_32F);
    for(int i = 0; i < testData.rows; i++) {
        cv::Mat response(1, 1, CV_32FC1);
        cv::Mat sample = testData.row(i);

        mlp.predict(sample, response);
        predicted.at<float>(i,0) = response.at<float>(0,0);

    }

    cout << "Accuracy_{MLP} = " << evaluate(predicted, testClasses) << endl;
    plot_binary(testData, predicted, "Predictions Backpropagation");
}

void knn(cv::Mat& trainingData, cv::Mat& trainingClasses, cv::Mat& testData, cv::Mat& testClass

    CvKNearest knn(trainingData, trainingClasses, cv::Mat(), false, K);
    cv::Mat predicted(testClasses.rows, 1, CV_32F);
    for(int i = 0; i < testData.rows; i++) {
            const cv::Mat sample = testData.row(i);
            predicted.at<float>(i,0) = knn.find_nearest(sample, K);
    }

    cout << "Accuracy_{KNN} = " << evaluate(predicted, testClasses) << endl;
    plot_binary(testData, predicted, "Predictions KNN");
}

void bayes(cv::Mat& trainingData, cv::Mat& trainingClasses, cv::Mat& testData, cv::Mat& testCla

    CvNormalBayesClassifier bayes(trainingData, trainingClasses);
    cv::Mat predicted(testClasses.rows, 1, CV_32F);
    for (int i = 0; i < testData.rows; i++) {
        const cv::Mat sample = testData.row(i);
        predicted.at<float> (i, 0) = bayes.predict(sample);
    }

    cout << "Accuracy_{BAYES} = " << evaluate(predicted, testClasses) << endl;
    plot_binary(testData, predicted, "Predictions Bayes");
}

void decisiontree(cv::Mat& trainingData, cv::Mat& trainingClasses, cv::Mat& testData, cv::Mat&

    CvDTree dtree;
    cv::Mat var_type(3, 1, CV_8U);

    // define attributes as numerical
    var_type.at<unsigned int>(0,0) = CV_VAR_NUMERICAL;
    var_type.at<unsigned int>(0,1) = CV_VAR_NUMERICAL;
    // define output node as numerical
    var_type.at<unsigned int>(0,2) = CV_VAR_NUMERICAL;

    dtree.train(trainingData,CV_ROW_SAMPLE, trainingClasses, cv::Mat(), cv::Mat(), var_type, cv
    cv::Mat predicted(testClasses.rows, 1, CV_32F);
    for (int i = 0; i < testData.rows; i++) {
        const cv::Mat sample = testData.row(i);
        CvDTreeNode* prediction = dtree.predict(sample);
```

```
        predicted.at<float> (i, 0) = prediction->value;
    }

    cout << "Accuracy_{TREE} = " << evaluate(predicted, testClasses) << endl;
    plot_binary(testData, predicted, "Predictions tree");
}


int main() {

    cv::Mat trainingData(numTrainingPoints, 2, CV_32FC1);
    cv::Mat testData(numTestPoints, 2, CV_32FC1);

    cv::randu(trainingData,0,1);
    cv::randu(testData,0,1);

    cv::Mat trainingClasses = labelData(trainingData, eq);
    cv::Mat testClasses = labelData(testData, eq);

    plot_binary(trainingData, trainingClasses, "Training Data");
    plot_binary(testData, testClasses, "Test Data");

    svm(trainingData, trainingClasses, testData, testClasses);
    mlp(trainingData, trainingClasses, testData, testClasses);
    knn(trainingData, trainingClasses, testData, testClasses, 3);
    bayes(trainingData, trainingClasses, testData, testClasses);
    decisiontree(trainingData, trainingClasses, testData, testClasses);

    cv::waitKey();

    return 0;
}
```

Experimental Settings

Parameter	Value
Trainingdata size	200
Testdata size	2000

Please note: I didn't optimize any parameters in these experiments. A Support Vector Machine may perform much better, if you choose an appropriate Kernel and optimize the parameters subsequently (for example with a Grid Search). A Neural Network may perform much better, when choosing the appropriate number of layers and training iterations. And so on... So don't interpret too much into these experimental results, I really wanted to show some features of OpenCV only.

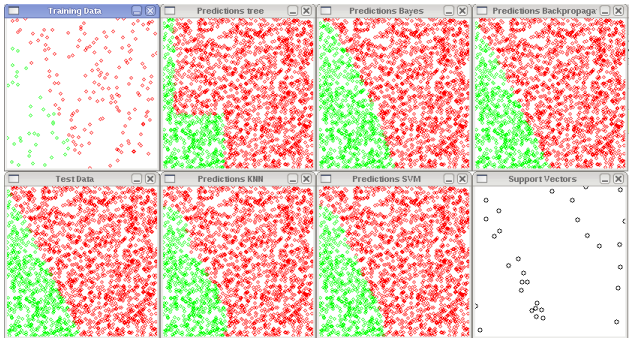
Results

y = 2x

Predictor	Accuracy
Support Vector Machine	0.99
Multi Layer Perceptron (2, 10, 15, 1)	0.994

Multi Layer Perceptron (2, 10, 15, 1)	0.994
k-Nearest-Neighbor (k = 3)	0.9825
Normal Bayes	0.9425
Decision Tree	0.923

Plot

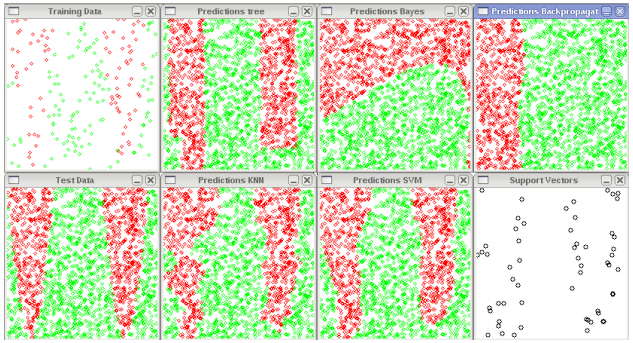


y = sin(10x)

Predictor	Accuracy
Support Vector Machine	0.913
Multi Layer Perceptron (2, 10, 15, 1)	0.6855
k-Nearest-Neighbor (k = 3)	0.9
Normal Bayes	0.632

Decision Tree	0.886

Plot



y = tan(10x)

Predictor	Accuracy
Support Vector Machine	0.7815
Multi Layer Perceptron (2, 10, 15, 1)	0.5115
k-Nearest-Neighbor (k = 3)	0.8195
Normal Bayes	0.542
Decision Tree	0.9155

Plot

n16kPWKWrHnvnHRvnvNBuyD4PHqdIAdxTvqdThP-5HDknWFWmhkEusKzujYk0AFV5H00TZcqn0KdpyfqnHRLPjnvnfKEpyfqnHnsnj0YnsKWpyfqP1cvrHnz0AqLUWYs0ZK45HcsP6KWThnqPj0YPHb)

【投稿】Machine Learning With Spark Note 2:构建简单的推荐系统 (http://blog.csdn.net/u0…

本文为数盟特约作者投稿，欢迎转载，请注明出处“数盟社区”和作者 博主简介：段石石，1号店精准化推荐算法工程师，主要负责1号店用户画像构建，喜欢钻研点Machine Learning的黑科技，对D...

u013886628 (http://blog.csdn.net/u013886628) 2016年07月05日 07:58 1490

Machine learning and Data Mining - Association Analysis with Python (http://blog.csdn.n…

Recently I've been working with recommender systems and association analysis. This last one, specia...

derekrose (http://blog.csdn.net/derekrose) 2014年01月07日 14:22 1302

Machine Learning With Spark学习笔记 (http://blog.csdn.net/LXYTSOS/article/details/4670…

此笔记为本人在阅读Machine Learning With Spark的时候所做的，笔记有翻译不准确或错误的地方欢迎大家指正。Spark集群 Spark集群由两种进程组成：一个驱动程序和多个执行程序。...

LXYTSOS (http://blog.csdn.net/LXYTSOS) 2015年07月01日 14:23 3693

Machine Learning：Linear Regression With One Variable (http://blog.csdn.net/u0127360…

Machine Learning:Linear Regression With One Variable 机器学习可以应用于计算机视觉，自然语言处理，数据挖掘等领域，可以分为监督学习（S...

u012736084 (http://blog.csdn.net/u012736084) 2014年11月04日 23:20 808

【Stanford Machine Learning】Lecture 2--Linear Regression with Multiple Variables (htt…

本系列(Stanford Machine Learning)

mingyong_blog (http://blog.csdn.net/mingyong_blog) 2016年08月04日 00:05 639

The Steps of Machine Learning with Python (http://blog.csdn.net/Xw_Classmate/article/d…

开始。这是最容易令人丧失斗志的两个字。迈出第一步通常最艰难。当可以选择的方向太多时，就更让人两腿发软了。 从哪里开始？ 本文旨在通过七个步骤，使用全部免费的线上资料，帮助新人获取最基本的 ...

Xw_Classmate (http://blog.csdn.net/Xw_Classmate) 2016年04月28日 19:02 387

Coursera Machine Learning 第一周 quiz Linear Regression with One Variable 习题答案 (h…

1.Consider the problem of predicting how well a student does in her second year of college/universit...

mupengfei6688 (http://blog.csdn.net/mupengfei6688) 2016年11月08日 23:54 5046

Machine Learning week 2 quiz: Linear Regression with Multiple Variables (http://blog.cs…

Linear Regression with Multiple Variables 5 试题 1. Suppose m=4 students have taken some class, and ...

GarfieldEr007 (http://blog.csdn.net/GarfieldEr007) 2015年11月15日 11:30 11904

An introduction to machine learning with scikit-learn (http://blog.csdn.net/zqxnum1/artic…

scikit-learn 是一个基于SciPy和Numpy的开源机器学习模块，包括分类、回归、聚类的一系列算法，而且有详细的文档，是边学边练的绝佳教材，本文将通过一个简单的例子向大家展示如何使用sci...

zqxnum1 (http://blog.csdn.net/zqxnum1) 2015年01月23日 13:36 723

building machine learning system with Python 学习笔记--从零开始机器学习（0）(http://bl…

作为机器学习的小白，想打好基础，先学Google TensorFlow、百度PaddlePaddle这些成熟的框架，只能知其然，而不知其所以然。想深入理解机器学习，还是要从学习底层的实现细节开始。 b...

qq_25203493 (http://blog.csdn.net/qq_25203493) 2017年05月07日 15:30 604

Machine Learning - IV. Linear Regression with Multiple Variables多变量线性规划 (Week 2)…

机器学习Machine Learning - Andrew NG courses学习笔记 linear regression works with multiple variables or wi...

pipisorry (http://blog.csdn.net/pipisorry) 2015年02月05日 16:44 2539

Machine Learning week 1 quiz: Linear Regression with One Variable (http://blog.csdn.ne…

Linear Regression with One Variable 5 试题 1. Consider the problem of predicting how wel...

GarfieldEr007 (http://blog.csdn.net/GarfieldEr007) 2015年11月07日 21:33 2993

Coursera Machine Learning 第二周 quiz Linear Regression with Multiple Variables 习题答…

1.Suppose m=4 students have taken some class, and the class had a midterm exam and a final exam. You...

mupengfei6688 (http://blog.csdn.net/mupengfei6688) 2016年11月09日 14:17 6126

building machine learning system with Python 学习笔记--从零开始机器学习（3）第一个应…

这个小应用是根据已有的网站访问量来预测什么时候到达现有设施的极限，估计是每小时100000个请求。 这种问题在初高中求函数极值时经常遇到，只是现在函数形式是未知的，只有一定量的离散数据。机器学习就派上...

qq_25203493 (http://blog.csdn.net/qq_25203493) 2017年05月07日 20:10 124

Machine Learning With Spark学习笔记（在10万电影数据上训练、使用推荐模型）(http://blo…

我们现在开始训练模型，还输入参数如下： rank：ALS中因子的个数，通常来说越大越好，但是对内存占用率有直接影响，通常rank在10到200之间。 iterations：迭代次数，每次迭代都会减...

LXYTSOS (http://blog.csdn.net/LXYTSOS) 2015年08月14日 16:56 3926

《machine learning with spark》学习笔记--推荐模型 (http://blog.csdn.net/pangjiuzala/arti…

Prepare dataDataSource DownloadUpload data to HDFSIt's easy for the programmers who are familiar to ...

pangjiuzala (http://blog.csdn.net/pangjiuzala) 2016年02月02日 21:21 1270

building machine learning system with Python 学习笔记--从零开始机器学习（2）第一章 (h...

Python机器学习入门 ps：想了解机器学习发展历史、使命、面临的问题这些的可以看百度BOSS李彦宏新书《智能革命》，挺通俗易懂的。 机器学习的目标就是通过若干示例让机器学会完成人物，例如电子邮...

qq_25203493 (http://blog.csdn.net/qq_25203493) 2017年05月07日 17:34 234

Machine Learning：Linear Regression With Multiple Variables (http://blog.csdn.net/u012...

Machine Learning:Linear Regression With Multiple Variables 接着上次预测房子售价的例子，引出多变量的线性回归。 接着上次预测房子售...

u012736084 (http://blog.csdn.net/u012736084) 2014年11月07日 00:02 1058

【投稿】Machine Learning With Spark Note 1:数据基本处理 (http://blog.csdn.net/u013886...

本文为数盟特约作者投稿，欢迎转载，请注明出处"数盟社区"和作者 博主简介：段石石，1号店精准化推荐算法工程师，主要负责1号店用户画像构建，喜欢钻研点Machine Learning的黑科技，对D...

u013886628 (http://blog.csdn.net/u013886628) 2016年07月05日 07:57 462

TensorFlow Machine Learning with Financial Data on Google Cloud Platform (http://blog...

TensorFlow Machine Learning with Financial Data on Google Cloud Platform最近开始用TensorFlow跑一些机器学习算法，至于为...

ly1046906153 (http://blog.csdn.net/ly1046906153) 2016年05月19日 21:58 3473

Machine Learning - Linear Regression with One Variable (http://blog.csdn.net/iracer/artic...

This article contains learning model representation, cost function and Gradient Descent algorithm to...

iracer (http://blog.csdn.net/iracer) 2016年02月13日 17:22 1031

Detecting Falls with Wearable Sensors Using Machine Learning Techniques (http://blog....

基于机器学习和可穿戴传感器的跌倒探测系统 摘要：对于易跌倒人群来说，跌倒是一个严重的公众的健康问题甚至可能威胁到生命安全。为此，我和我的团队开发了一套自动跌倒检测系统，通过固定在人体六个不同部位的传感...

guoyang1305584135 (http://blog.csdn.net/guoyang1305584135) 2016年11月01日 19:47 356

Teaching Mario to play with himself: AI, machine learning, and Super Mario Bros. (http://...

原文地址：http://www.extremetech.com/extreme/197886-teaching-mario-to-play-with-himself-ai-machine-learni...

wolf96 (http://blog.csdn.net/wolf96) 2015年08月08日 17:48 601