



Main Page

Related Pages

Namespaces ▾

Classes ▾

Files ▾

Examples

Java documentation

Q ▾ Search

1

OpenCV Tutorials

Other tutorials (ml, objdetect, photo, stitching, video)

Support Vector Machines for Non-Linearly Separable Data

[Prev Tutorial: Introduction to Support Vector Machines](#)

[Next Tutorial: Introduction to Principal Component Analysis \(PCA\)](#)

Original author	Fernando Iglesias García
Compatibility	OpenCV >= 3.0

Goal

In this tutorial you will learn how to:

- Define the optimization problem for SVMs when it is not possible to separate linearly the training data.
- How to configure the parameters to adapt your SVM for this class of problems.

Motivation

Why is it interesting to extend the SVM optimization problem in order to handle non-linearly separable training data? Most of the applications in which SVMs are used in computer vision require a more powerful tool than a simple linear classifier. This stems from the fact that in these tasks **the training data can be rarely separated using an hyperplane**.

Consider one of these tasks, for example, face detection. The training data in this case is composed by a set of images that are faces and another set of images that are non-faces (*every other thing in the world except from faces*). This training data is too complex so as to find a representation of each sample (*feature vector*) that could make the whole set of faces linearly separable from the whole set of non-faces.

Extension of the Optimization Problem

Remember that using SVMs we obtain a separating hyperplane. Therefore, since the training data is now non-linearly separable, we must admit that the hyperplane found will misclassify some of the samples. This *misclassification* is a new variable in the optimization that must be taken into account. The new model has to include both the old requirement of finding the hyperplane that gives the biggest margin and the new one of generalizing the training data correctly by not allowing too many classification errors.

We start here from the formulation of the optimization problem of finding the hyperplane which maximizes the **margin** (this is explained in the previous tutorial "Introduction to Machine Learning").

Table of Contents

- ↓ Goal
- ↓ Motivation
- ↓ Extension of the Optimization Problem
- ↓ Source Code
- ↓ Explanation
- ↓ Results