Detection of Pedestrians in Images using SVM Project Report

Gopal Menon
Machine Learning
Fall 2016, University of Utah

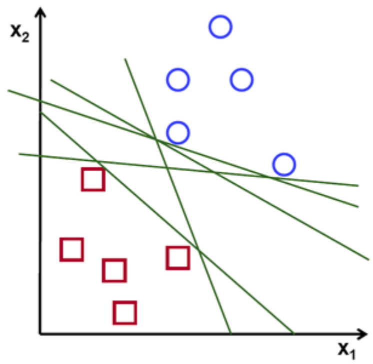

Fig. 1. Many possible separating boundaries[1]

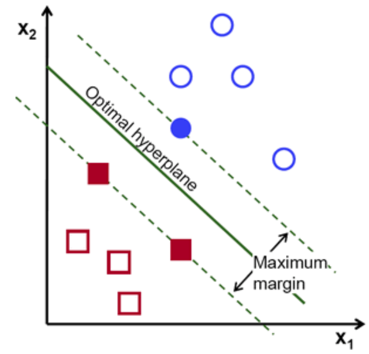

Fig. 2. Separation with maximum margin[1]

I. Project Scope

The scope was to construct a classifier using a Support Vector Machine (SVM) that can classify a street image as one containing or not containing a pedestrian. The main reason why it is interesting to me is that it is one of those problems that can be easily explained to a human, while being extremely difficult to program using traditional methods.

An SVM is a linear classifier that finds the separating boundary between instances of two classes or labels. Although there may be many separating boundaries as shown in figure 1, the linear separation is achieved by selecting a boundary that maximizes the separation between the classes. The vectors that lie on the separating band are called support vectors. These are shown as the solid squares and circle in figure 2. By finding a such a band that maximizes the separation between classes, the classifier has better generalization for examples that have not been seen yet.

A. Initial Assumptions

When I started with the project, I went forward with the assumption that through use of a suitable kernel function, images can be linearly separated into ones that contain and do not contain a human. My understanding was that even if the images were not linearly separable in the input space, they would be linearly separable in some higher dimension space through the use of Kernels. Since then, I learnt during office hours after class that this may not be possible However I am not sure why it is not possible. One suggestion I got during office hours was to use the input to the final layer of a Visual Geometry Group (VGG) Very Deep Convolutional Net (VGG NET) classifier [5] as features for an SVM to do linear separation.

II. Progress

A. Literature and Resources used

1) Since I needed to start work on the project before SVMs were covered in class, I used the class recordings from last year to get familiarized with the subject.
2) In addition to the class recordings, I also used SVM class recordings from a Caltech course on machine learning available at http://amlbook.com and based on the book *Learning from Data: A Short Course* [2].
3) Once I got comfortable with the material, I referred to the SVM book by Christiani et. al. 2000 [3] for an understanding of the dual form of linear learning machines and kernel functions.

B. Current Status

1) I have coded an SVM in Java that finds the separator with maximum margin when it is given a training set where each training set entry is an array of floating point numbers. There is no maximum or minimum limitation for the dimensionality of the training set, but all the training and test set entries must have the same dimensionality. The SVM needs to be provided an object that is an instance of a class that implements a method that returns the dot product of its inputs in a different space. I have successfully tested the SVM with a kernel

that just returns the dot product of its inputs by giving it one-dimensional and two-dimensional inputs.

2) I have coded the following kernels in addition to the trivial implementation mentioned above. These are based on the kernels described in the book *Learning from Data: A Short Course* [2].
   a) Polynomial Kernel
   b) Degree Q Polynomial Kernel
   c) Gaussian Radial Bias Function Kernel
3) I have obtained test data consisting of images containing humans and those not containing humans. The images were downloaded from the *INRIA Person Dataset* [4].

III. Strategy

A. Further Work

1) Understand the working of SVMs with Kernels
2) Understand the limitations of SVMs particularly why they cannot classify images
3) Understand VGG Net classifier
4) Obtain an implementation of VGG Net classifier
5) Perform any image pre-processing that is required by a VGG Net classifier.
6) Build and test a pedestrian image classifier using the input to the last layer of a VGG Net classifier as input to an SVM.

IV. Classifier Evaluation

1) The VGG Net classifier can be used as a baseline for evaluating the SVM classifier. Since I plan to use the input to the last layer of the VGG Net classifier as the input to the SVM, I expect both the baseline and the SVM classifier to have similar results. The reason I say this is because the last layer of a VGG Net classifier will function as a linear threshold detector, or a perceptron, which will be possible to recreate using an SVM.
2) In addition to comparing the accuracy of the implemented SVM classifier with that of a VGG Net classifier, I also plan to do manual checks in order to make sure that the results are accurate.

References

[1] "Introduction to Support Vector Machines." *Introduction to Support Vector Machines* - OpenCV 2.4.13.1 Documentation. N.p., n.d. Web. 27 Sept. 2016.

[2] Abu-Mostafa, Yaser S., Malik Magdon-Ismail, and Hsuan-Tien Lin. *Learning from Data: A Short Course*. United States: AMLBook.com, 2012. Print.

[3] Cristianini, Nello, and John Shawe-Taylor. *An Introduction to Support Vector Machines: And Other Kernel-based Learning Methods*. Cambridge: Cambridge UP, 2000. Print.

[4] "INRIA Person Dataset." *INRIA Person Dataset*. N.p., n.d. Web. 08 Nov. 2016.

[5] "Visual Geometry Group Home Page." *Visual Geometry Group Home Page*. N.p., n.d. Web. 08 Nov. 2016.