CSci 4270 and 6270
Computational Vision, Spring 2021
Lecture 14: Object Detection and SVMS
March 15, 2021

Overview

- Problem statement: for each image find all locations of a particular class of object.
- Most common examples are faces, cars and pedestrians.
- Note that the class of objects is known in advance and specialized training is applied to build the detection algorithm.
- Here are examples of pedestrians from the Dalal-Triggs paper we will discuss in class.

Figure 2. Some sample images from our new human detection database. The subjects are always upright, but with some partial occlusions and a wide range of variations in pose, appearance, clothing, illumination and background.

Important Ideas to Watch For

1. Example of using hand-crafted features together with a machine learning algorithm.
2. Introduction to non-SIFT descriptors
3. Introduction to SVM
4. Training using skewed distribution and selecting "hard negatives"

Materials Distributed

1. These lecture notes.
2. Dalal & Triggs paper from CVPR 2005
3. Introduction to SVMs from Professor Zaki's book: Data Mining and Machine Learning: Fundamental Concepts and Algorithms, first edition

Sliding Window Algorithms

- Detection occurs by testing a subregion of an image of fixed / known size, such as 64x128.
- Starting with the upper left corner of the image, the subregion or "window" is placed at locations

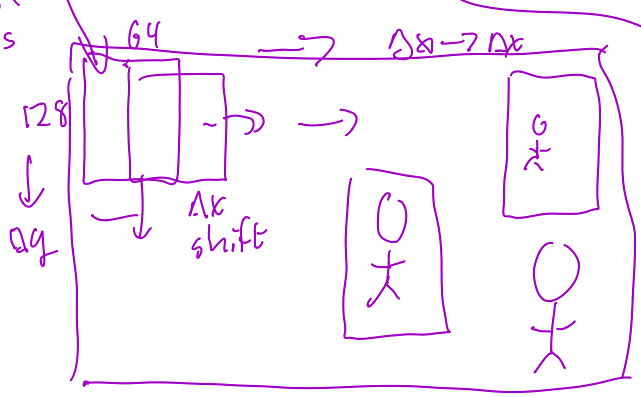
$$(i\Delta x, j\Delta y), \qquad \text{for } i = 0, 1, \ldots \text{ and } j = 0, 1, \ldots$$

- In each location, a feature vector is extracted from the image subregion and tested for the presence of the object.
- Note that Δx and Δy are both much smaller that the 64 and 128 pixels that define the horizontal and vertical dimensions of the subregion.
- Different sizes are handled by rescaling the image in non-integral increments.
- Classes of methods:
  - Extract sophisticated descriptor vector and classify it.
  - Keypoints, descriptors and vocabularies.
  - Collection of simple classifiers: AdaBoost or random forests.

There are deep neural network analogs to these methods.

Sophisticated descriptor

5K–10K values



classifier → Measure of "pedestrianess"

Our Focus: Histogram of Oriented Gradients

- Dalal and Triggs, *IEEE CVPR* 2005; included with these notes.

- Here is the plot of the work flow of the algorithm.

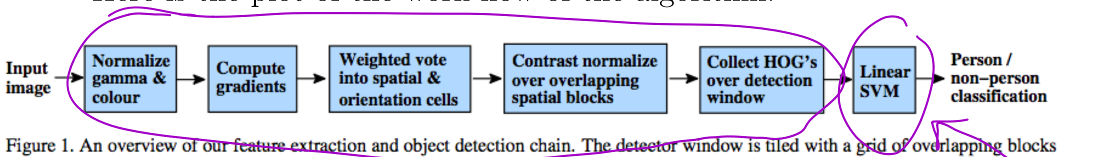


Figure 1. An overview of our feature extraction and object detection chain. The detector window is tiled with a grid of overlapping blocks in which Histogram of Oriented Gradient feature vectors are extracted. The combined vectors are fed to a linear SVM for object/non-object classification. The detection window is scanned across the image at all positions and scales, and conventional non-maximum suppression is run on the output pyramid to detect object instances, but this paper concentrates on the feature extraction process.

- We'll start with an extended discussion of linear support vector machines, based on Professor Zaki's book chapter, distributed with these notes.

Outline of SVM Discusion, Part 1

Start with the ideal case: maximizing the margin between two linearly separable sets of points (feature vectors).

- The combined set $\{(\mathbf{x}_i, y_i)\}$, where $\mathbf{x}_i$ is a point vector, $y_i = 1$ if the point is in one set, and $y_i = -1$ if the point is in the other.
  - The positive set consists of points with label $y_i = 1$ and they have

    $$\mathbf{w}^\top \mathbf{x}_i + b \geq 0$$

  - The negative set consists of points with label $y_i = -1$ and they have

    $$\mathbf{w}^\top \mathbf{x}_i + b \leq 0$$

    $y_i =$

    $y_i = 1$

    flip ≥ 0 to ≤ 0

  - We combine these by writing

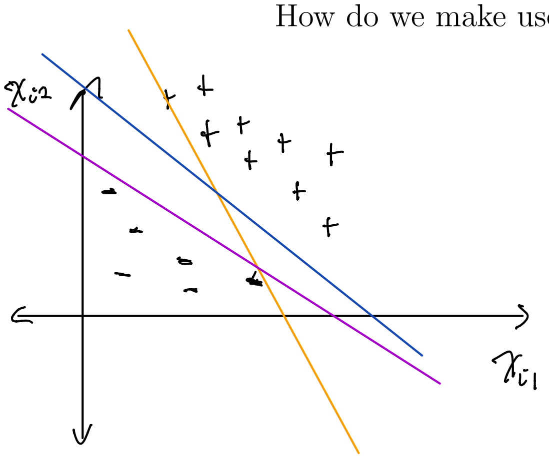
    $$y_i(\mathbf{w}^\top \mathbf{x}_i + b) \geq 0$$

- Remember, our goal is to estimate the parameters of $\mathbf{w}$ and b.
- We'll get started by building on what we already know: we've written (hyper)planes as

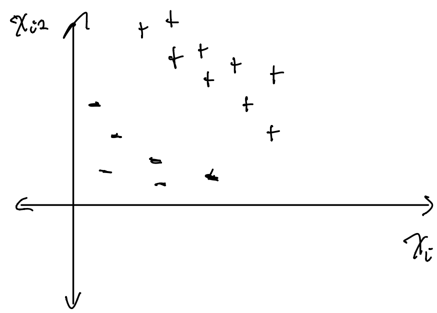
$$\mathbf{w}^\top \mathbf{x} + b = 0, \quad \text{where } \|\mathbf{w}\| = 1.$$

How do we make use of this?

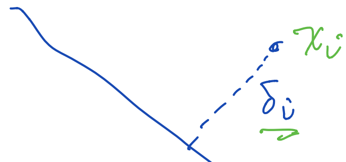
Feature space



Maximizes margin / separation
Intuition that you have best chance of correctly classifying future point $\vec{x}$



x_{i2}
x_{i1}



- We'll define δ_i as the distance from the hyperplane, and then allow $\mathbf{w}$ to vary in magnitude. In this case, the distance of a point from the hyperplane becomes

$$\delta_i = \frac{y_i(\mathbf{w}^\top \mathbf{x}_i + b)}{\|\mathbf{w}\|}.$$

$\|w\| \delta_i = y_i(w^T x_i + b)$

distance

- Then, we'll add constraints to form

$$y_i(\mathbf{w}^\top \mathbf{x}_i + b) \geq 1$$

which can be imposed as long as the points are linearly separable.

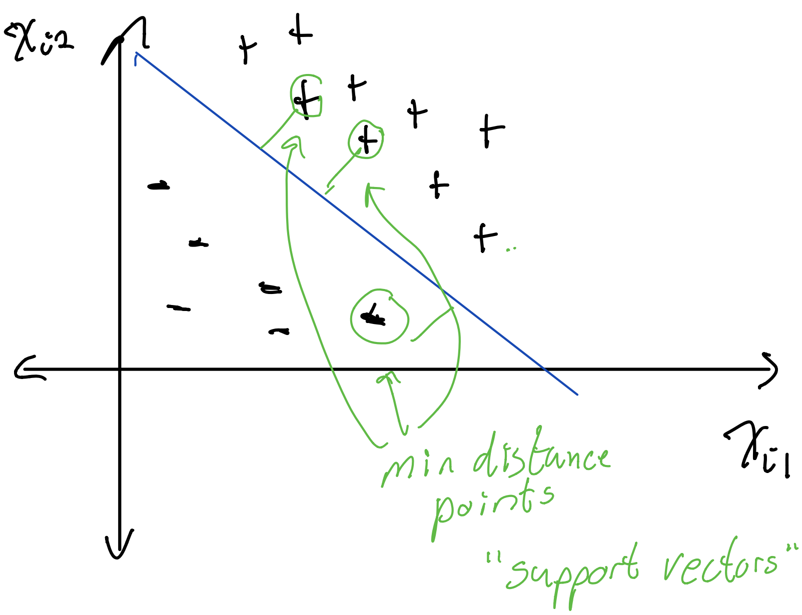
Scale w to make $\|w\| \delta_i \geq 1$

- Moreover, there will always be minimum-distance points where

$$y_i(\mathbf{w}^\top \mathbf{x}_i + b) = 1$$

These are the *support vectors*, denoted δ_i^*

δ_{i*} are points in data set closest to separating hyperplane.



- We'd like to maximize the distance of the support vectors from the separating hyperplane.
  - This distance is called the *margin*.
- This will create the goal of minimizing the magnitude of $\mathbf{w}$.
- Combining all of this will produce the constrained optimization:

$$\min \frac{\|\mathbf{w}\|^2}{2} \qquad \text{subject to} \qquad \forall i : y_i(\mathbf{w}^\top \mathbf{x}_i + b) \geq 1.$$

- This is the final objective function for the linearly separable case, which can be solved using standard quadratic programming methods.

$$\frac{\cancel{y_i(w^T x_i + b)}\ 1}{\|w\|} = \delta_i^*$$

distance

for support vectors

$$\frac{1}{\|w\|} = \delta_i^*$$

distance we want to maximize

minimize mag of $\vec{w}$

magnitude

$$\min \frac{\|w\|^2}{2} \qquad \forall i \quad y_i(w^T x + b) \geq 1$$

after we get w finding b is a linear search

Outline of SVM Discusion, Part 2

balance cost of mistake versus size of margin

- When the sets are not separable we introduce what are known as "slack variables", $\epsilon_i \geq 0$.
- Interpretation of the values of ϵ_i:
  - $\epsilon_i = 0$ means the point is classified correctly and outside the margin
  - $0 \leq \epsilon_i \leq 1$ means the point is classified correctly, but inside the margin
  - $\epsilon_i > 1$ means the point is classified incorrectly.
- There objective function for the non-separable case using the "hinge loss" is

$$\min \left[\frac{\|\mathbf{w}\|^2}{2} + C \sum_{i=1}^{N} \epsilon_i \right]$$

cost for each slack

subject to

$$\forall i : y_i(\mathbf{w}^\top \mathbf{x}_i + b) \geq (1 - \epsilon_i) \text{ and } \epsilon_i \geq 0$$

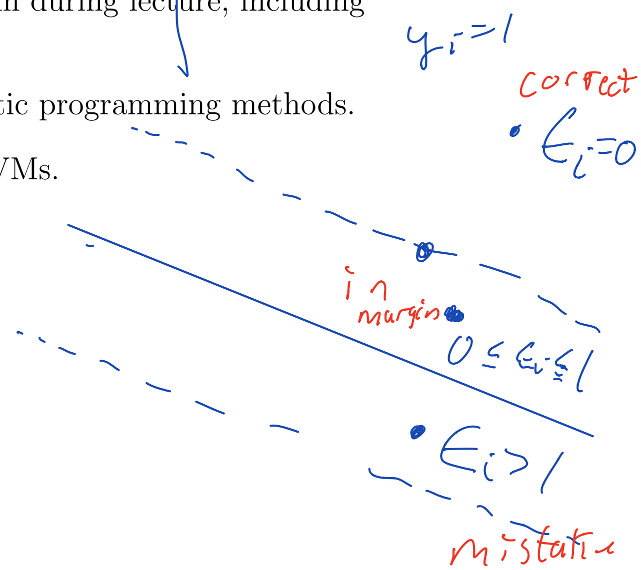
absorb error

positive

C is tuning constant

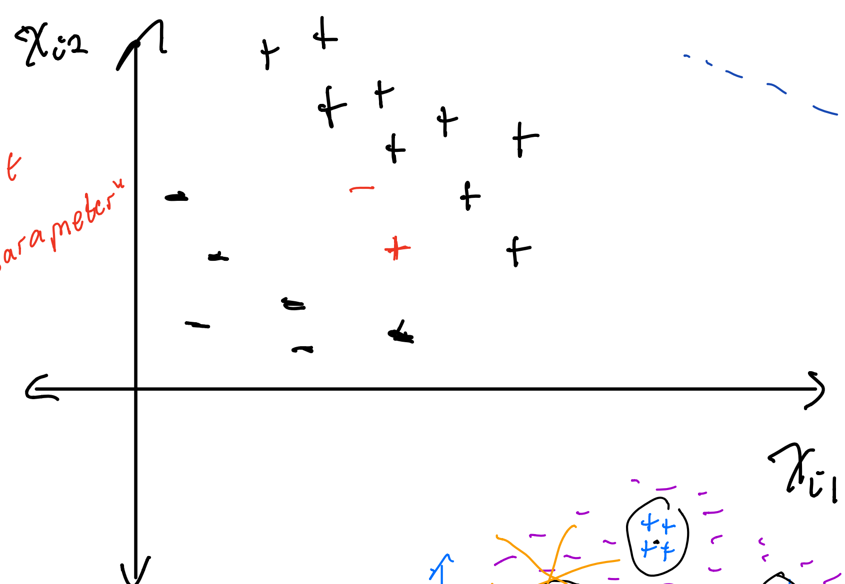
Lagrange multiple

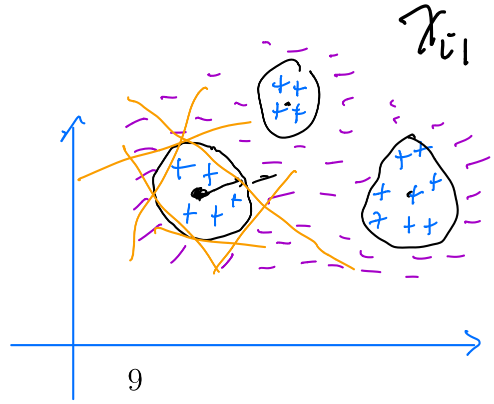
- We'll look closely at the meaning of each term during lecture, including the constant C.
- The result is solved once again using quadratic programming methods.
- Finally, we will briefly consider non-linear SVMs.



C set during a validation step --- try multiple values on small subset

C is a "hyperparameter"





Histogram of Oriented Gradients Summary

- Sliding window as described above:
  - Extract "Histogram of Oriented Gradients" (HOG) descriptor vector in each window.
  - Classify each vector as a detection or not using trained SVM.
  - Non-maximum suppression in overlapping descriptor regions.

- Evaluation: detection rate as a function of the "false positives per window"
- Training in two cycles using "hard negatives":
  - **Cycle 1:** All negatives randomly selected.
  - **Cycle 2:** About half of the negatives chosen from near the 1st margin's boundary.
- Formation of descriptor vector:
  - Gamma correction and color space
  - Gradient computation
  - Orientation histogram and interpolation
  - Blocks and block normalization
  - Formation of the final descriptor vector
    - Note that it is not much of a reduction in size from the original image, but the information is reorganized and normalized.
- The experimental tuning of various parameters anticipates what would soon be done automatically and implicitly in the training of the neural network.
- The weight vector of the learned SVM indicates what the algorithm gives importance to.

Summary

- Pedestrian detection problem using sliding window
- Descriptor — much more sophisticated than SIFT
- Training linear SVM using multistage process and hard negatives.
- Overall: important step toward modern use of deep learning networks.