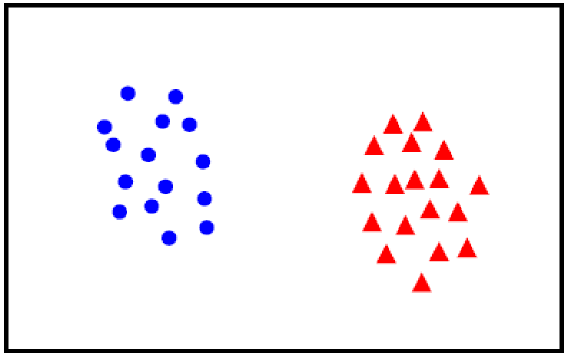
SVM classifiers

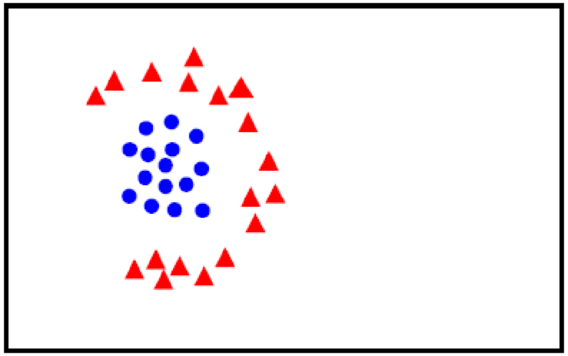
Binary classification

Given training data (x_i, y_i) for $i = 1 \ldots N$, with $x_i \in \mathbb{R}^d$ and $y_i \in \{-1,1\}$, learn a classifier $f(x)$ such that

$$f(x_i) = \begin{cases} \geq 0, & y_i = +1 \\ < 0, & y_i = -1 \end{cases}$$

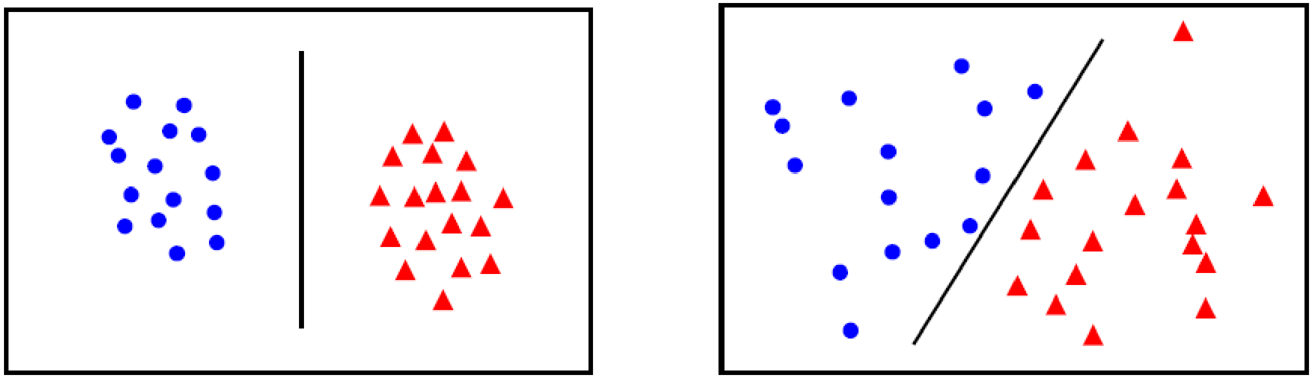
i.e. $y_i f(x_i) > 0$ for a correct classification.



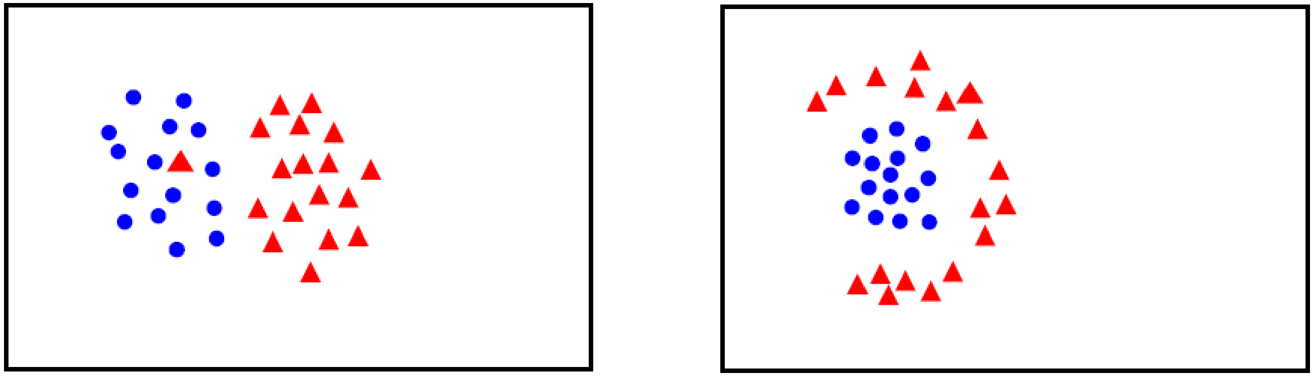


Linear separability

Linearly separable



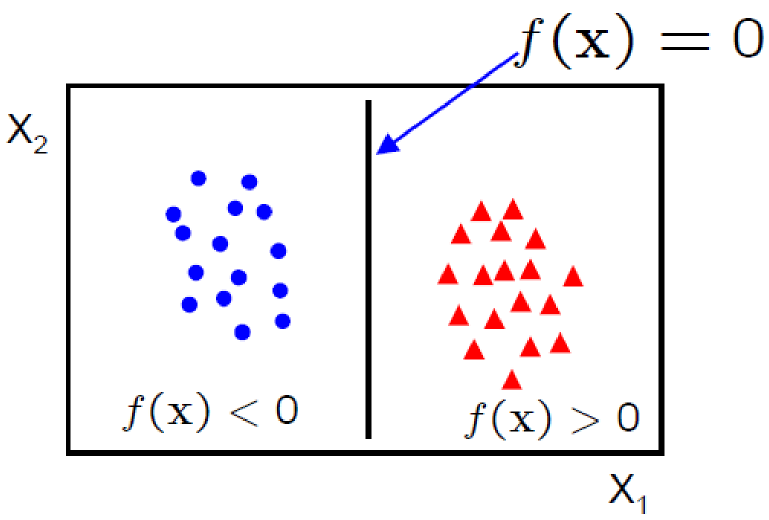
not Linearly separable



Linear classifiers

A linear classifier has the form

$$f(x) = w^T x + \mathrm{b}$$

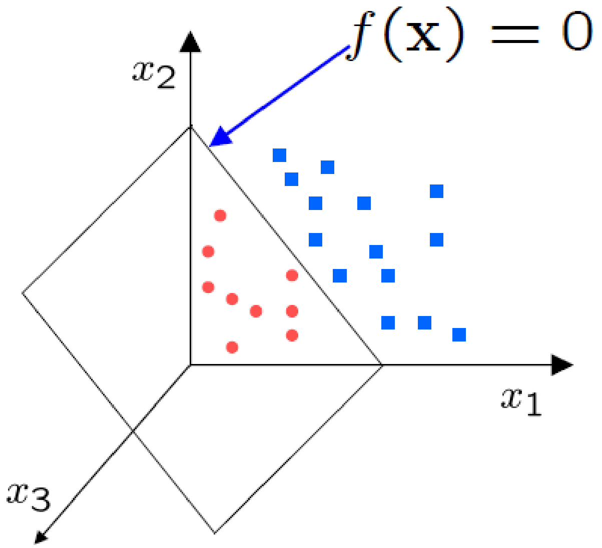


- In 2D the discriminant is a line
- w is the normal to the line, and b the bias
- w is known as the weight vector

Linear classifiers

A linear classifier has the form

$$f(x) = w^T x + \mathrm{b}$$



- In 3D the discriminant is a plane, and in nD it is a hyperplain

For a K-NN classifier it was necessary to ‘carry’ the training data

For a linear classifier, the training data is used to learn *w* and then discarded

Only *w* is needed for classifying new data

Reminder: The Perceptron Classifier

Given linearly separable data x_i labelled into two categories y_i = {-1,1}, find a weight vector w such that the discriminant function

$$f(x_i) = w^T x_i + \text{b}$$

Separates the categories for $i = 1, \dots, N$

- How can we find this separating hyperplane?

The Perceptron Algorithm

Write classifier as $f(x_i) = \tilde{w}^T \tilde{x}_i + \omega_0 = w^T x_i$

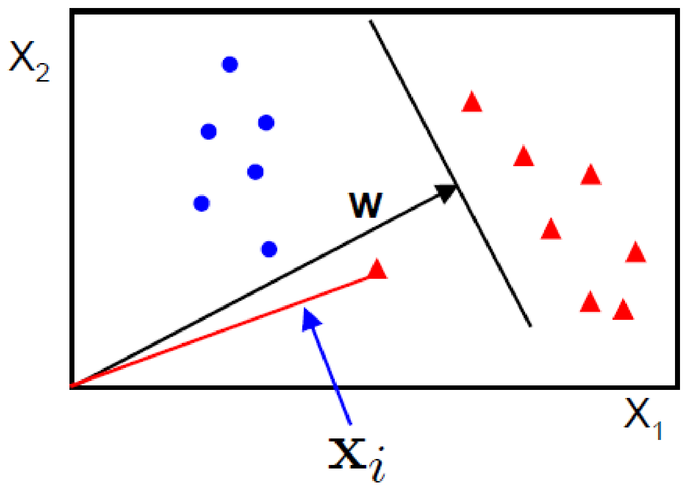
where $w = (\tilde{w}, \omega_0)$, $x_i = (\tilde{x}_i, 1)$

- Initialize $w = 0$
- Cycle though the data points $\{x_i, y_i\}$
  - If x_i is misclassified then $w \leftarrow w + \alpha \text{sign}(f(x_i)) x_i$
- Until all the data is correctly classified

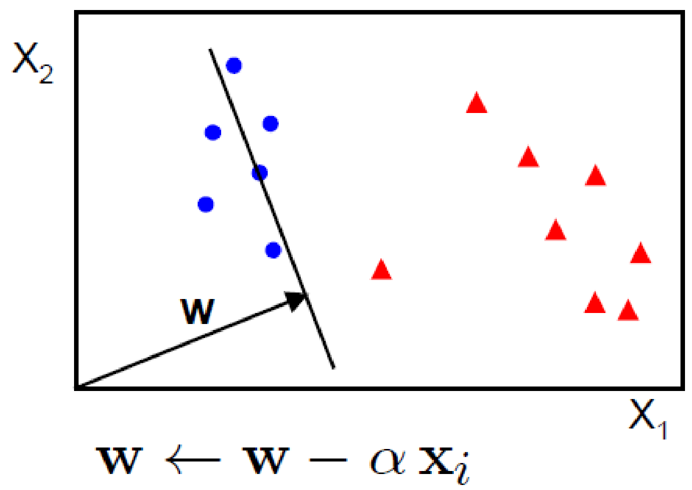
For example in 2D

- Initialize $w = 0$
- Cycle though the data points $\{x_i, y_i\}$
  - If x_i is misclassified then $w \leftarrow w + \alpha \mathrm{sign}(f(x_i))x_i$
- Until all the data is correctly classified

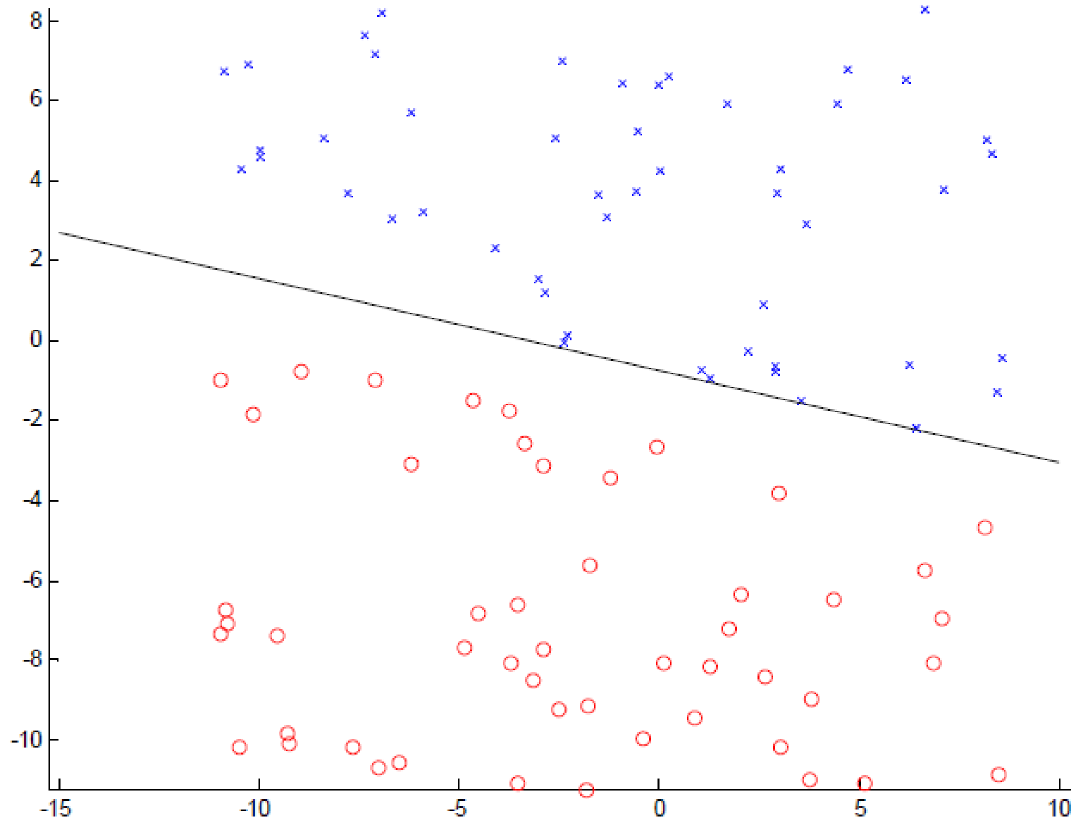
before update



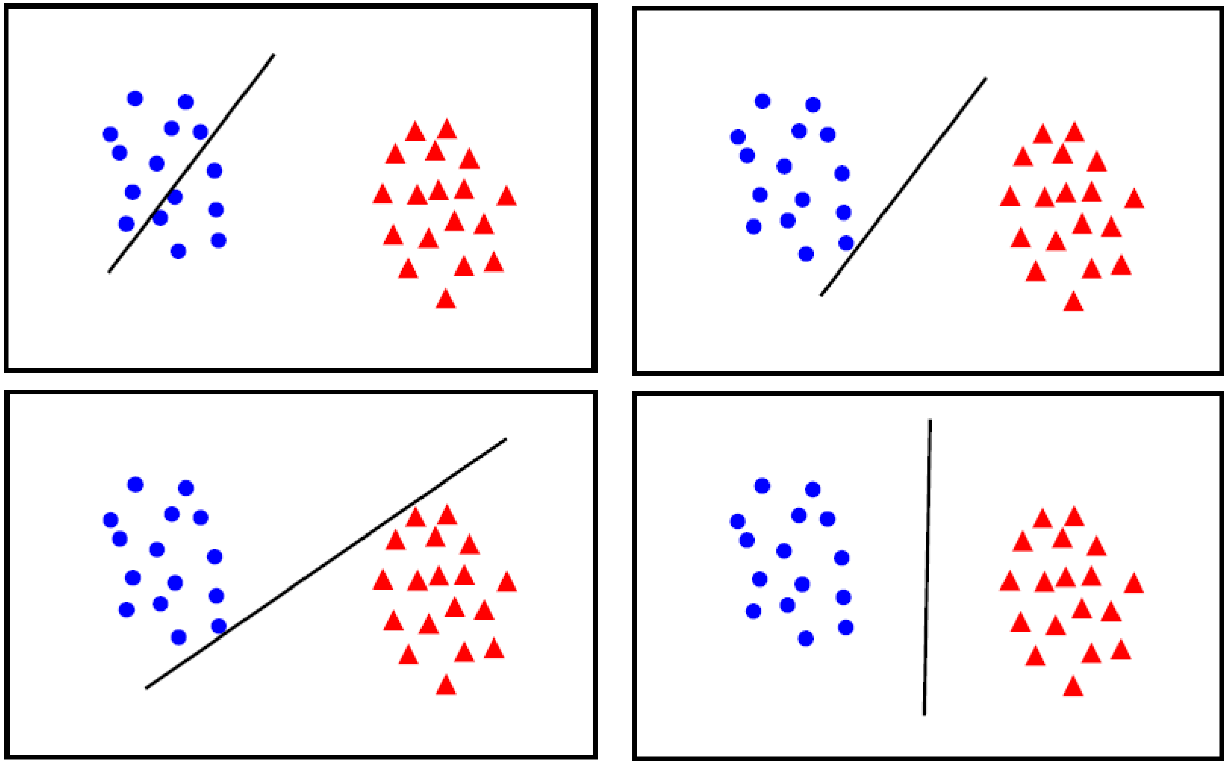
after update



Perceptron example

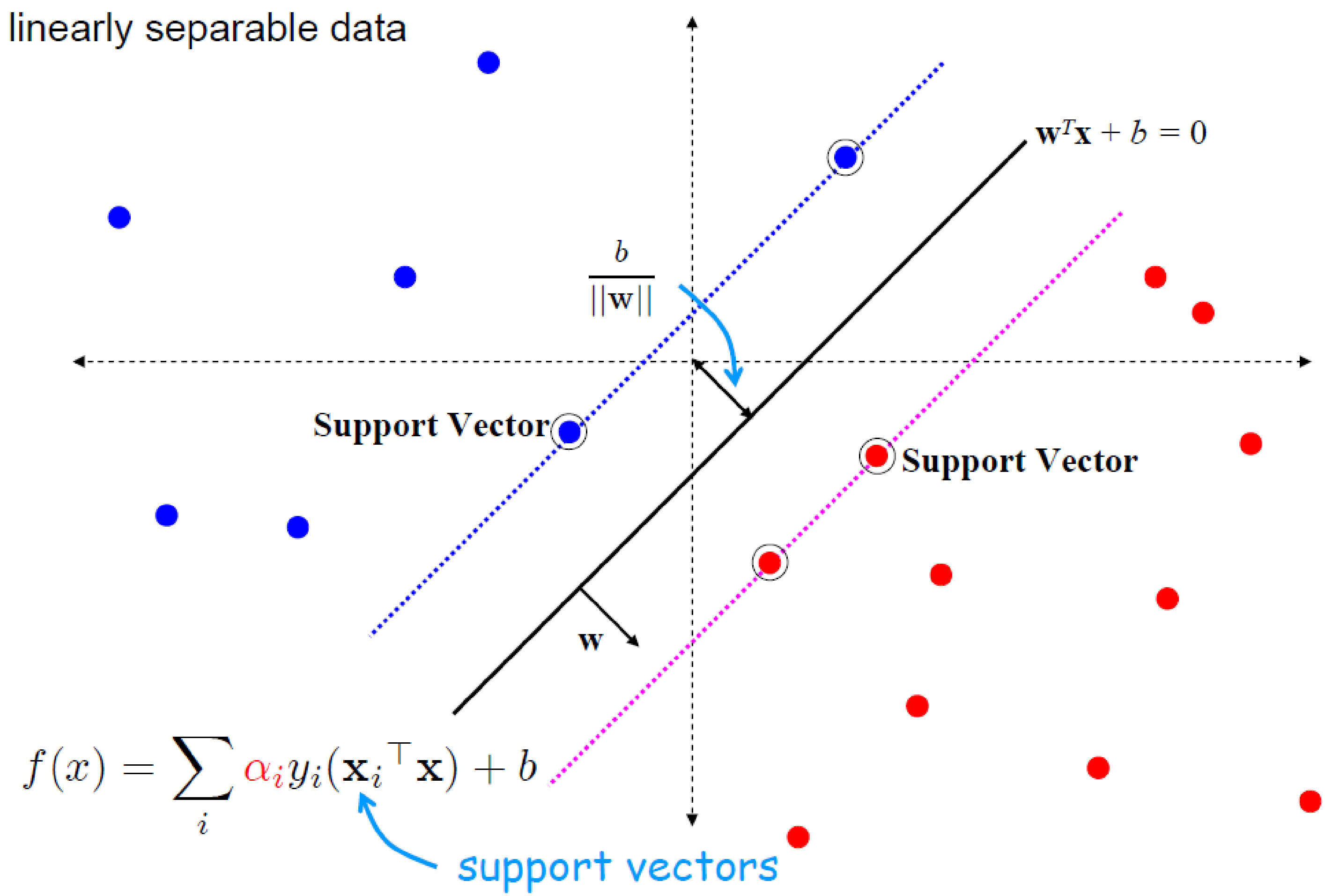


- If the data is linearly separable, then the algorithm will converge
- Convergence can be slow …
- Separating line close to training data
- We would prefer a larger margin for generalization



- Maximum margin solution: most stable under perturbations of the inputs

Support Vector Machine



SVM – sketch derivation

- Since $w^T x$+ b = 0 and c($w^T x$+ b) = 0 define the same plane, we have the freedom to choose the normalization of w

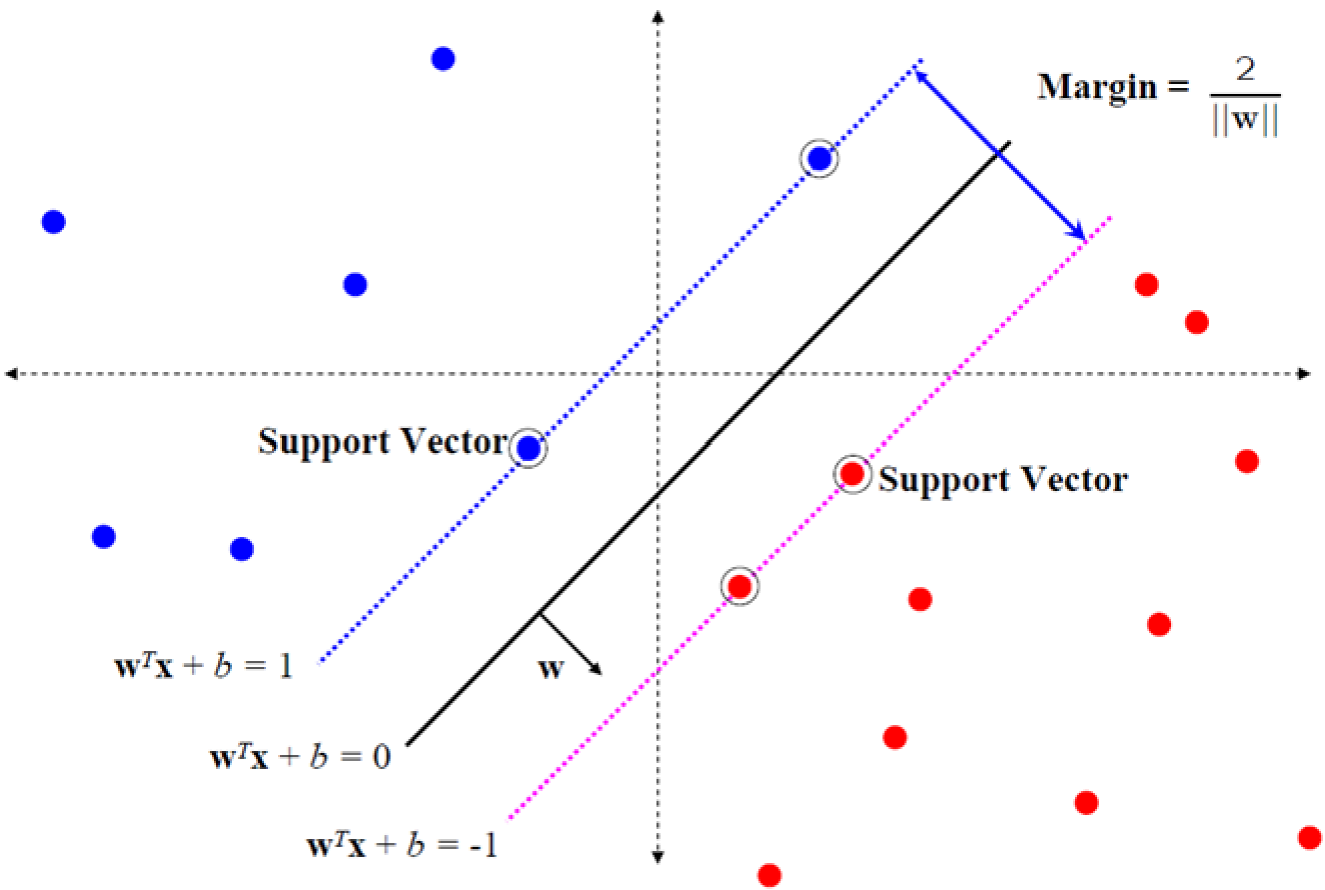
- Choose normalization such that $w^T x_+$+ b = +1 and $w^T x_-$+ b = -1 for the positive and negative support vectors respectively

- Then the margin is given by

$$\frac{w}{||w||} \cdot (x_+ - x_-) = \frac{w^T (x_+ - x_-)}{||w||} = \frac{2}{||w||}$$

Support Vector Machine

Linearly separable data



SVM – Optimization

- Learning the SVM can be formulated as an optimization:

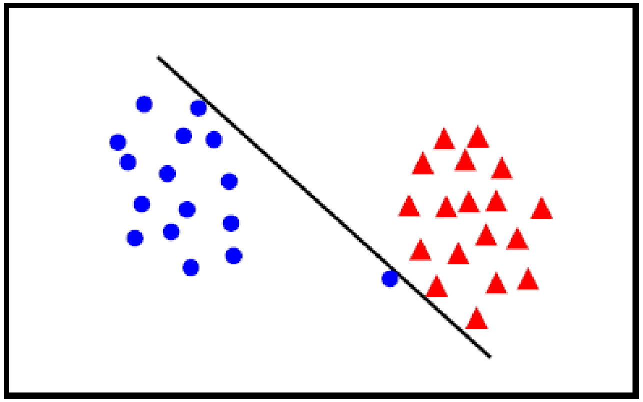
$$\max_{w} \frac{2}{\|w\|} \text{ subject to } w^T x_i + b \begin{cases} \geq \ \ 1 & if \ y_i = +1 \\ \leq -1 & if \ y_i = -1 \end{cases} \text{ for } i = 1 \ldots N$$

- Or equivalently

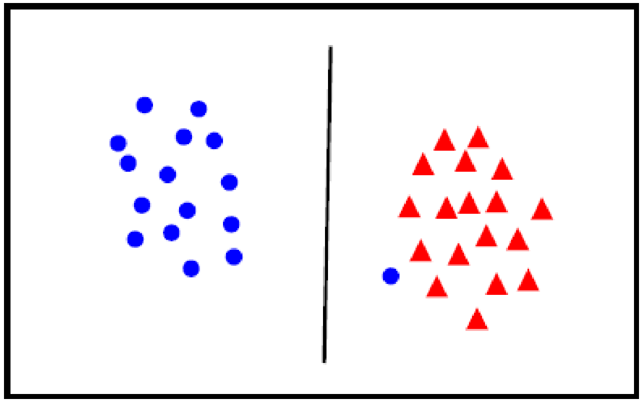
$$\min_{w} \|w\|^2 \text{ subject to } y_i(w^T x_i + b) \geq 1 \text{ for } i = 1 \ldots N$$

- This is a quadratic optimization problem subject to linear constraints and there is a unique minimum

Linear separability again: What is the best w?



- The points can be linearly separated but there is a very narrow margin



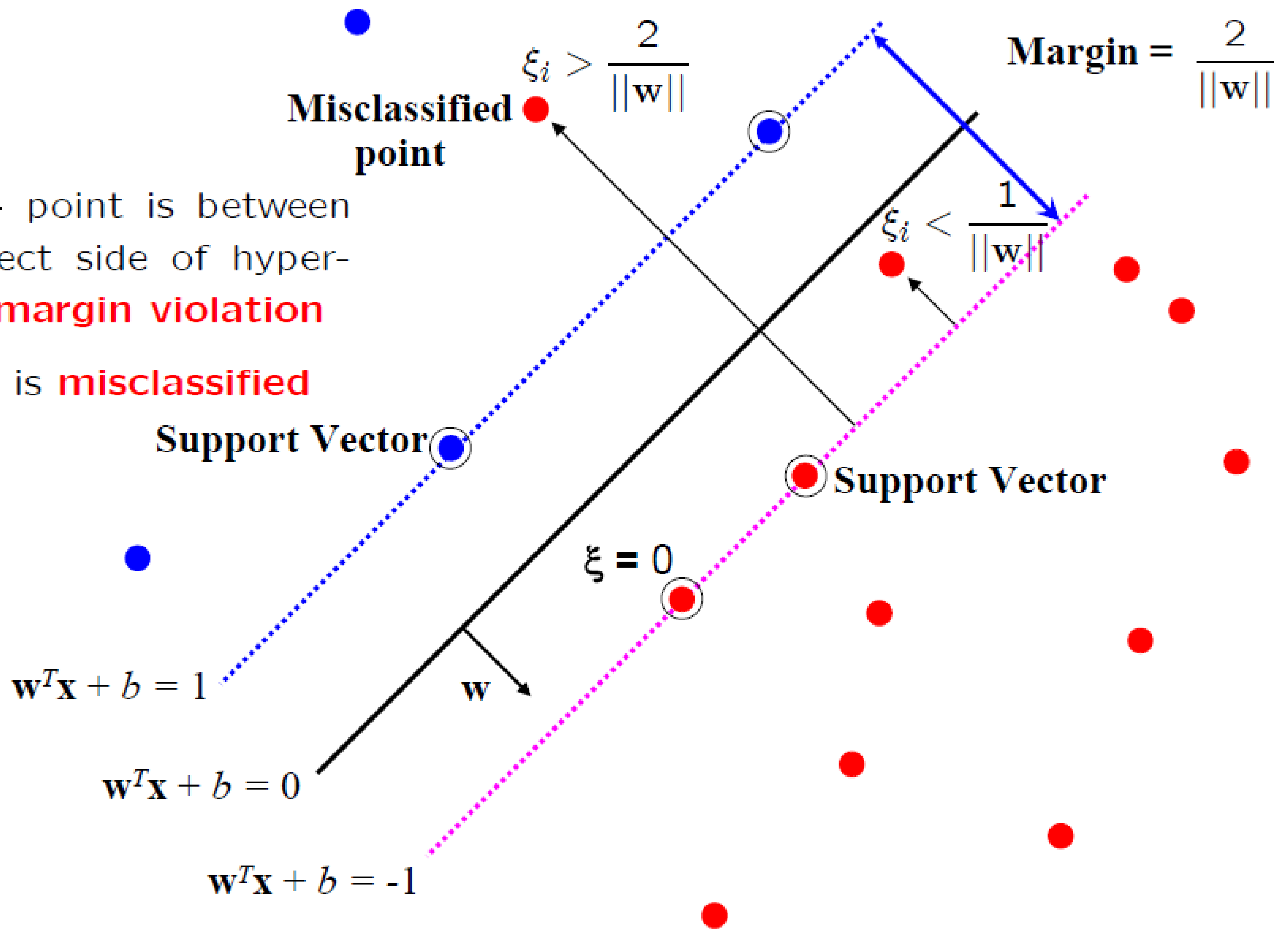
- But possibly the large margin solution is better, even though one constraint is violated

In general there is a trade off between the margin and the number of Mistakes on the training data

Introduce "slack" variables

$$\xi_i \geq 0$$

- for $0 < \xi \leq \frac{1}{||\mathbf{w}||}$ point is between margin and correct side of hyper-plane. This is a **margin violation**
- for $\xi > \frac{1}{||\mathbf{w}||}$ point is **misclassified**



“Soft” margin solution

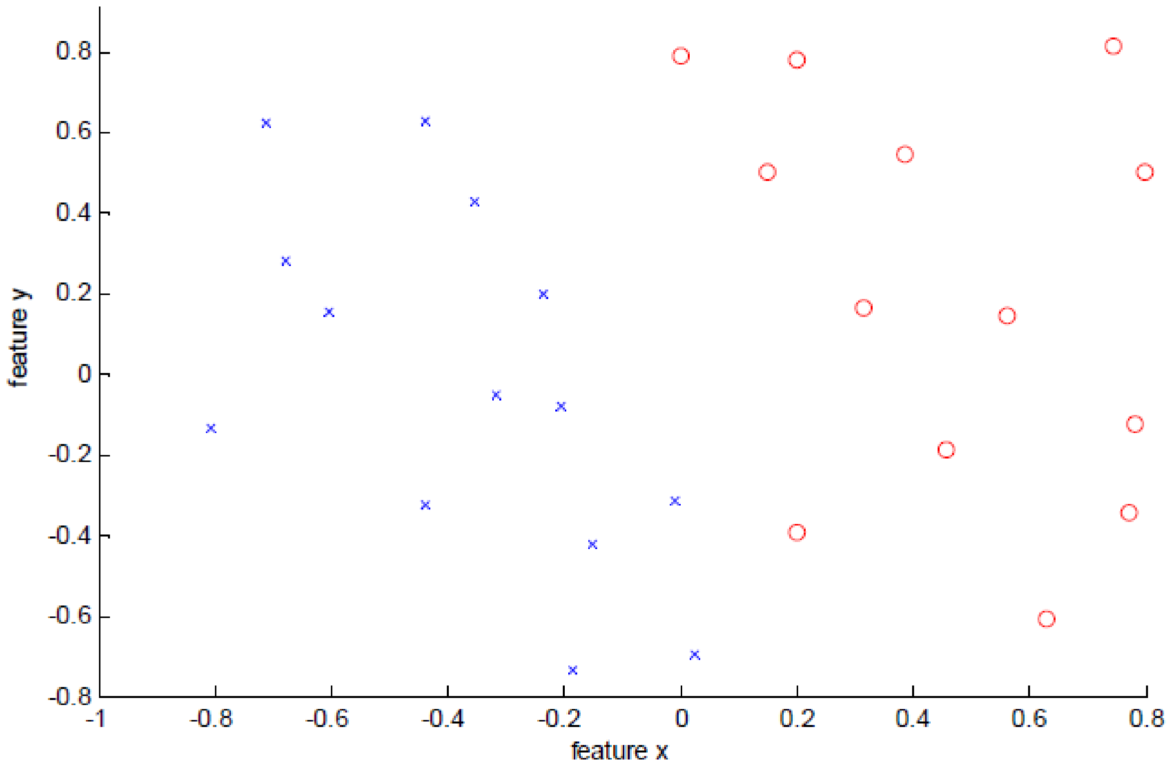
The optimization problem becomes

$$\min_{w \in Rd, \xi_i \in R^+} \|w\|^2 + c \sum_{i}^{N} \xi$$

subject to

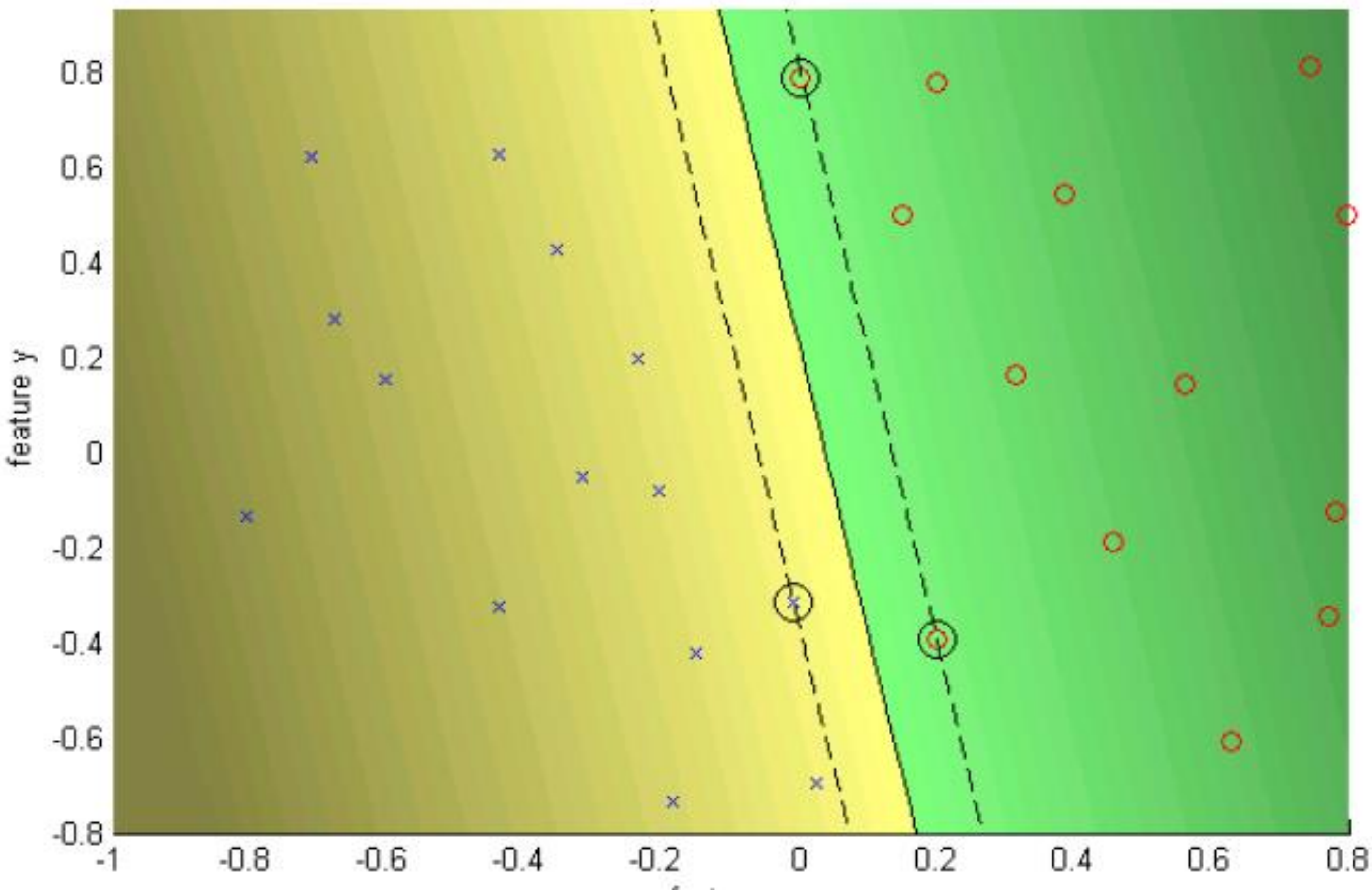
$$y_i(w^T x_i + b) \geq 1 - \xi_i \text{ for } i = 1 \ldots N$$

- Every constraint can be satisfied if ξ_i is sufficiently large
- C is regularization parameter:
  - small C allows constraints to be easily ignored → large margin
  - large C makes constraints hard to ignored → narrow margin
  - $C = \infty$ enforces all constraints: hard margin
- This is still a quadratic optimization problem and there is a unique minimum. Note, there is only one parameter, C.

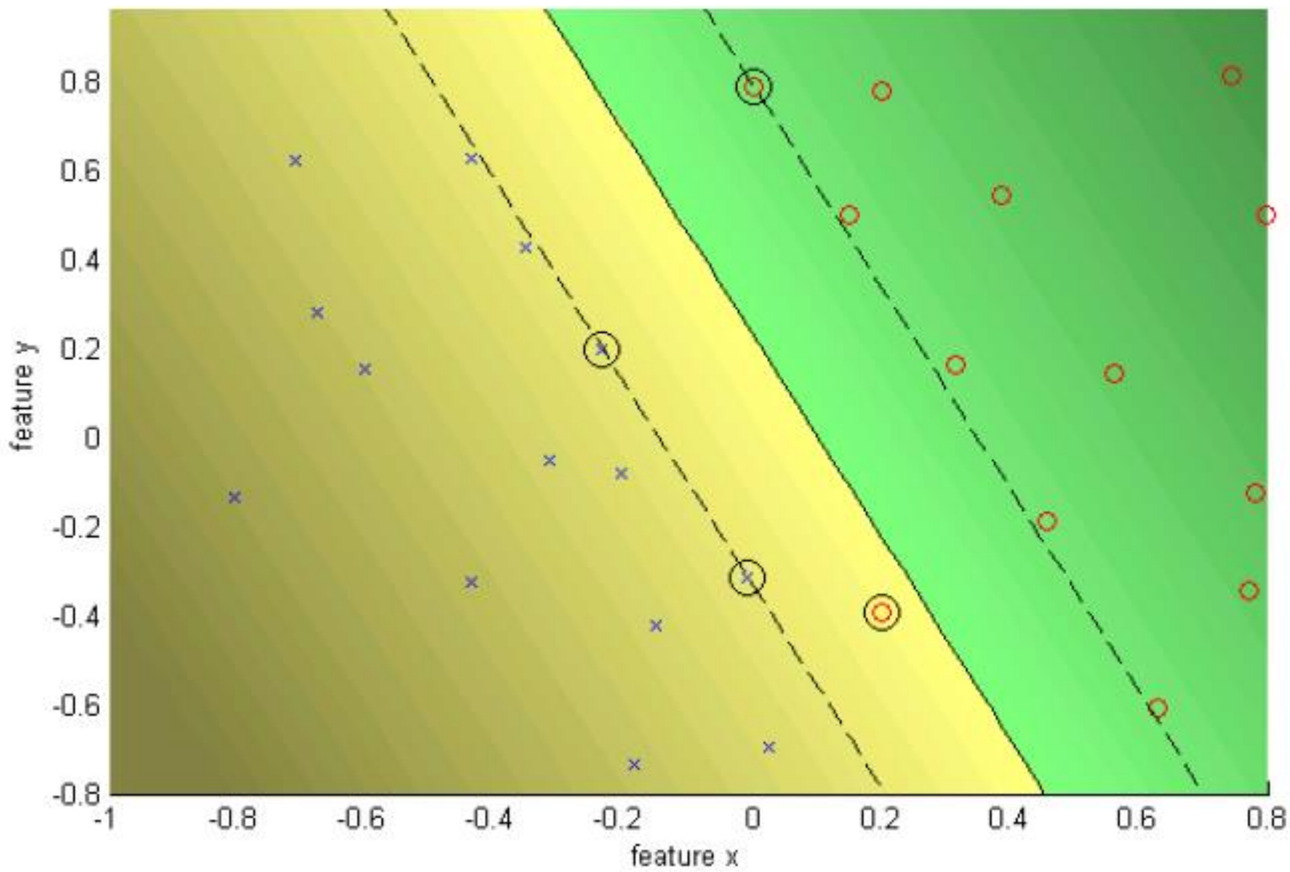


- Data is linearly separable
- But only with a narrow margin

$C = \infty$: hard margin



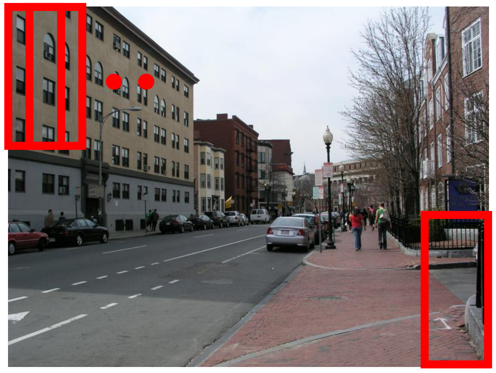
C = 10 soft margin



Application: Pedestrian detection in Computer Vision

- Objective: detect (localize) standing humans in an image (c.f. face detection with a sliding window classifier)•
  - reduces object detection to binary classification
  - does an image window contain a person or not?

Detection problem → (binary) classification problem

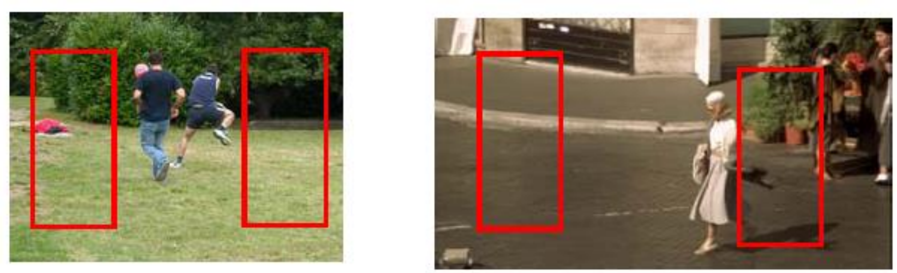


Each window is separately classified

Training data

- 64x128 images of humans cropped from a varied set of personal photos
- Positive data – 1239 positive window examples (reflections->2478)

- Negative data – 1218 person-free training photos (12180 patches)

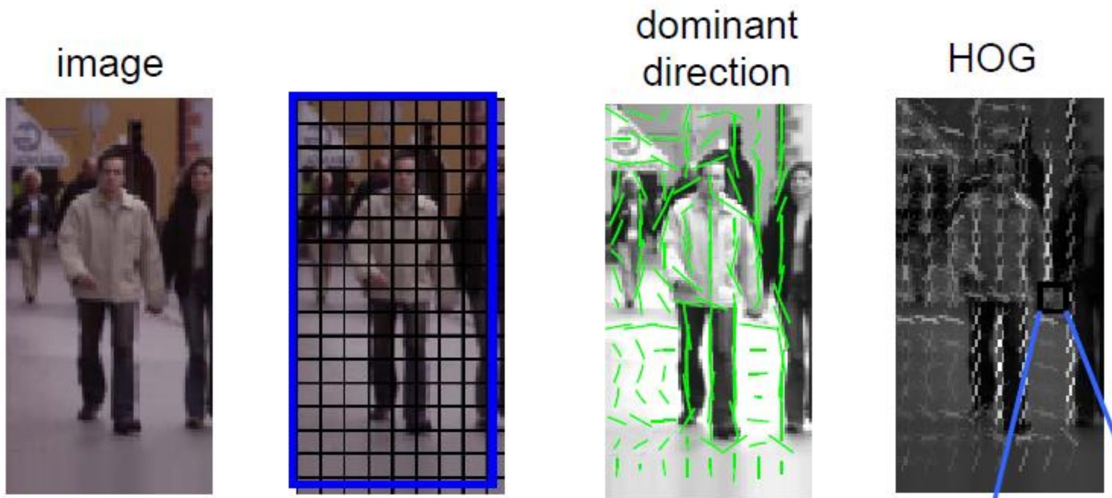


Training

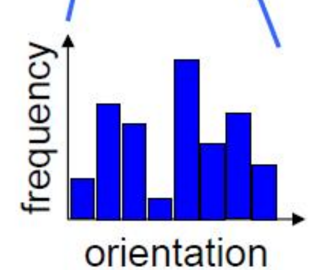
- A preliminary detector
  - Trained with (2478) vs (12180) samples

- Retraining
  - With augmented data set
    - initial 12180 + hard examples
  - Hard examples
    - 1218 negative training photos are searched exhaustively for false positive

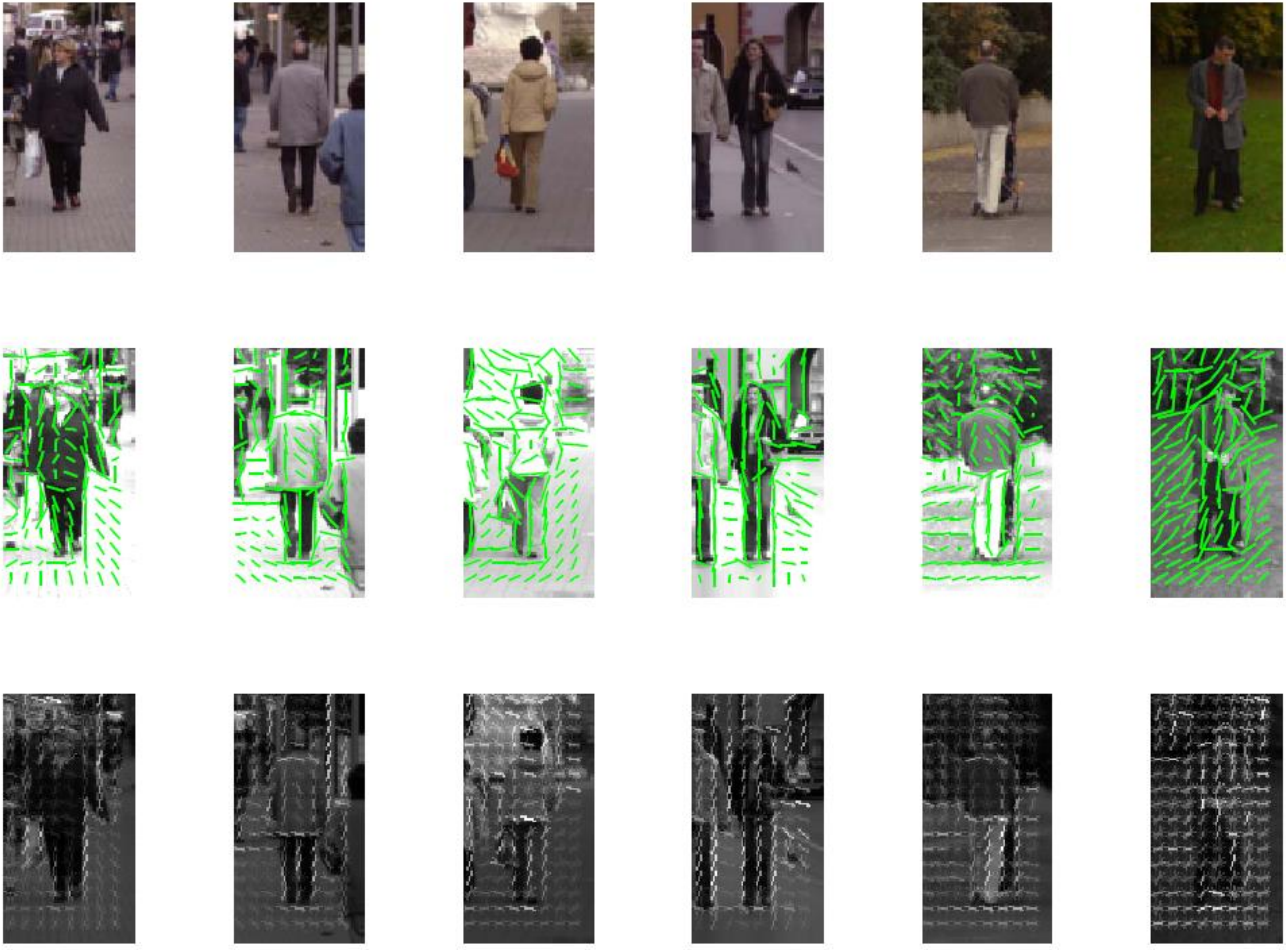
Feature: histogram of oriented gradients (HOG)



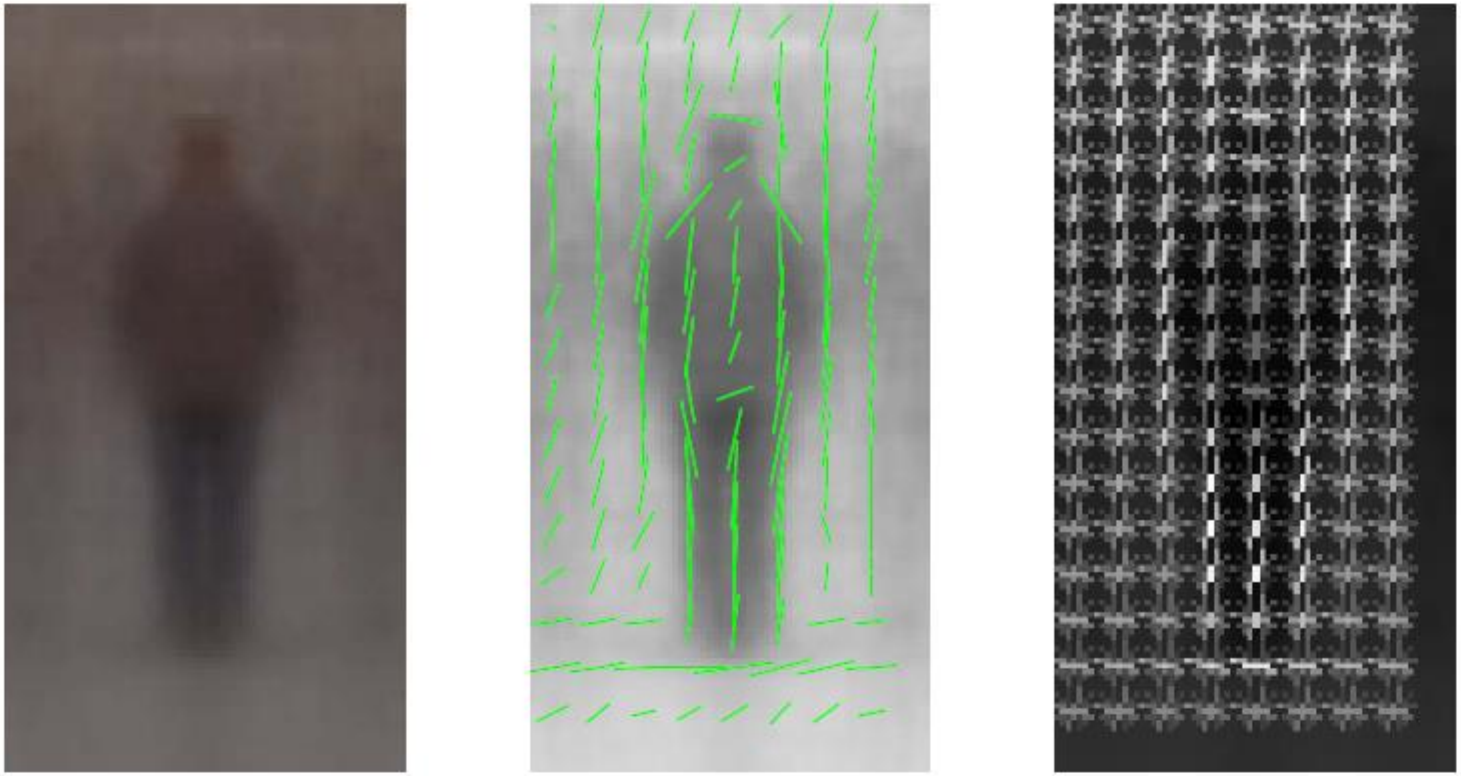
- tile window into 8 x 8 pixel cells
- each cell represented by HOG



Feature vector dimension = 16 x 8 (for tiling) x 8 (orientations) = 1024

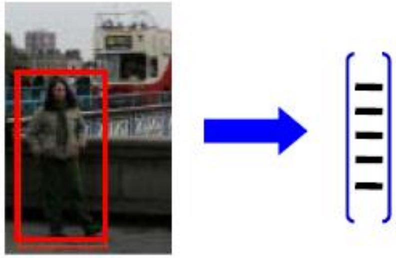


Averaged examples



Algorithm

- Training(Learning)
  - Represent each example window by a HOG feature vector



$$x_i \in \mathbb{R}^d, with\ d = 1024$$

  - Train a SVM classifier

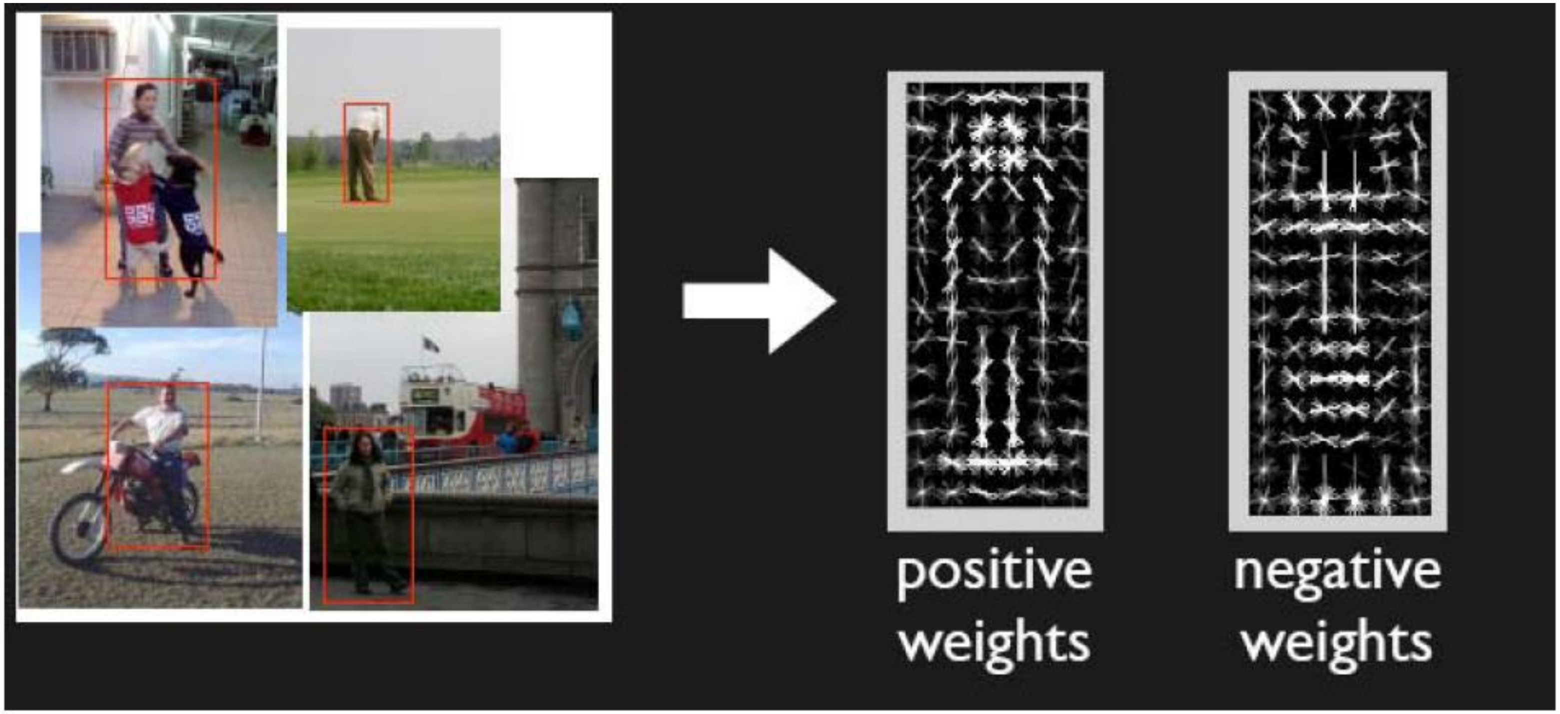
- Testing(Detection)
  - Sliding window classifier

$$f(x) = wTx + \mathrm{b}$$


0.62
1.21

Learned model

$$f(x) = wTx + \mathrm{b}$$



Slide from Deva Ramanan

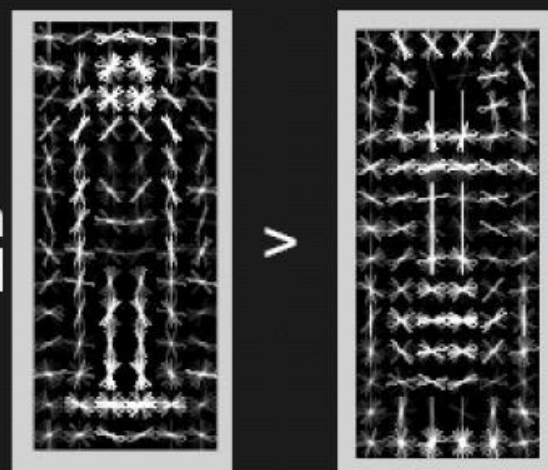
What do negative weights mean?

$$wx > 0$$

$$(w_+ - w_-)x > 0$$

$$w_+ > w_- x$$

pedestrian model



>

pedestrian background model

Complete system should compete pedestrian/pillar/doorway models

Discriminative models come equipped with own bg

(avoid firing on doorways by penalizing vertical edges)

Slide from Deva Ramanan