

Machine Learning (CS 181):

10. Max-Margin Methods

David C. Parkes and Sasha Rush

Spring 2017

1 / 54

Contents

- 1 Binary Classification
- 2 Max-margin methods
- 3 Hard Margin Formulation
- 4 Soft Margin Formulation
- 5 Application: Pedestrian Detection
- 6 Loss functions Revisited

2 / 54

Credit

Credit: A. Zisserman (Oxford)for slides throughout this deck.

3 / 54

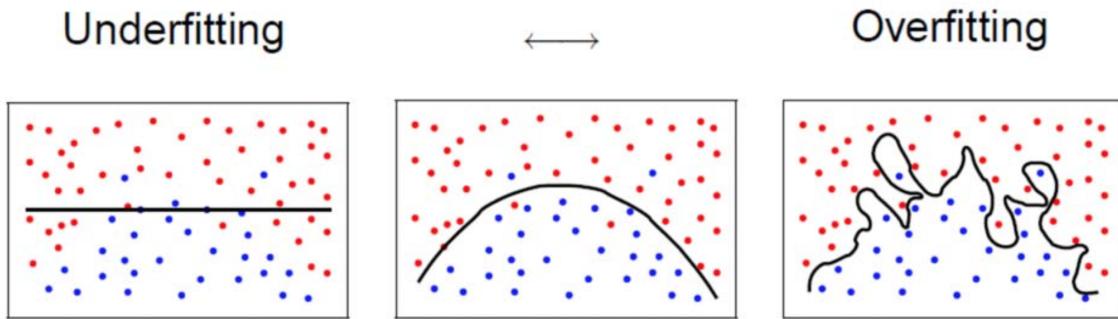
Contents

- [1] Binary Classification
- [2] Max-margin methods
- [3] Hard Margin Formulation
- [4] Soft Margin Formulation
- [5] Application: Pedestrian Detection
- [6] Loss functions Revisited

4 / 54

Goal: Good Generalization Performance

Minimize expected 0/1 classification error on **unseen data**.



Work with **discriminant-based classifiers**:

$$\hat{y} = \begin{cases} 1 & \text{if } h(\mathbf{x}; \mathbf{w}, w_0) > 0 \\ -1 & \text{o.w.} \end{cases}$$

5 / 54

Neural Networks: Review

Learn non-linear function $h(\mathbf{x}; \mathbf{w}, w_0, \mathbf{W}^1, \mathbf{w}_0^1)$. If $h > 0$, predict +1 else predict -1.

- Adaptive basis functions through optimization of weights, use of non-linear activations.
- Outstanding performance on many problems.
- Some **drawbacks**:
 - Non-convex (dragons!)... can use stochastic gradient descent, but not well understood and long training time
 - Adding domain knowledge is an art
 - Hard to interpret, complex representation

6 / 54

Max-margin methods

Learn function $h(\mathbf{x}; \mathbf{w}, w_0)$ (non-linear when used together with basis functions). If $h(\mathbf{x}; \mathbf{w}, w_0) > 0$, predict $+1$ else predict -1 .

- **Convex.** Can train via gradient descent (or other standard methods), will find global minimum.
- Coherent theory (“max margin”)
- Can readily engineer new basis functions (“kernel engineering”)
- Can obtain a succinct representation (thus, relatively interpretable.)
- The best performance

7 / 54

Max-margin methods

Learn function $h(\mathbf{x}; \mathbf{w}, w_0)$ (non-linear when used together with basis functions). If $h(\mathbf{x}; \mathbf{w}, w_0) > 0$, predict $+1$ else predict -1 .

- **Convex.** Can train via gradient descent (or other standard methods), will find global minimum.
- Coherent theory (“max margin”)
- Can readily engineer new basis functions (“kernel engineering”)
- Can obtain a succinct representation (thus, relatively interpretable.)
- ~~The best performance~~ Very good performance (!)

8 / 54

Applications of SVMs

- Face and object recognition ← ← ← ← ← ← ← see this later
- Predicting a cancer type from cell samples
- Fake news characterization
- Predict the function of proteins

Especially suited to problems with non-linear interactions between features because handle basis functions very nicely (next lecture!).

9 / 54

Review: Binary Classification

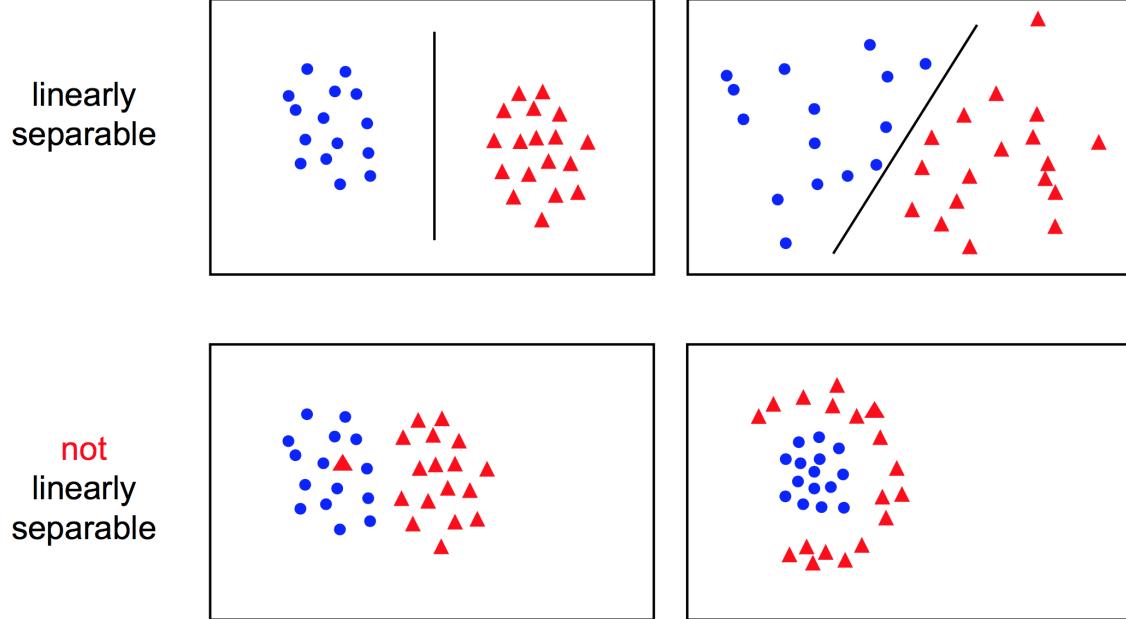
Training data (\mathbf{x}_i, y_i) for $i = 1, \dots, n$, $\mathbf{x}_i \in \mathbb{R}^m$, and $y_i \in \{-1, 1\}$, learn a **discriminant function** $h(\mathbf{x}; \mathbf{w}, w_0)$ such that

$$\hat{y} = \begin{cases} 1 & \text{if } h(\mathbf{x}; \mathbf{w}, w_0) > 0 \\ -1 & \text{o.w.} \end{cases}$$

Decision boundary:

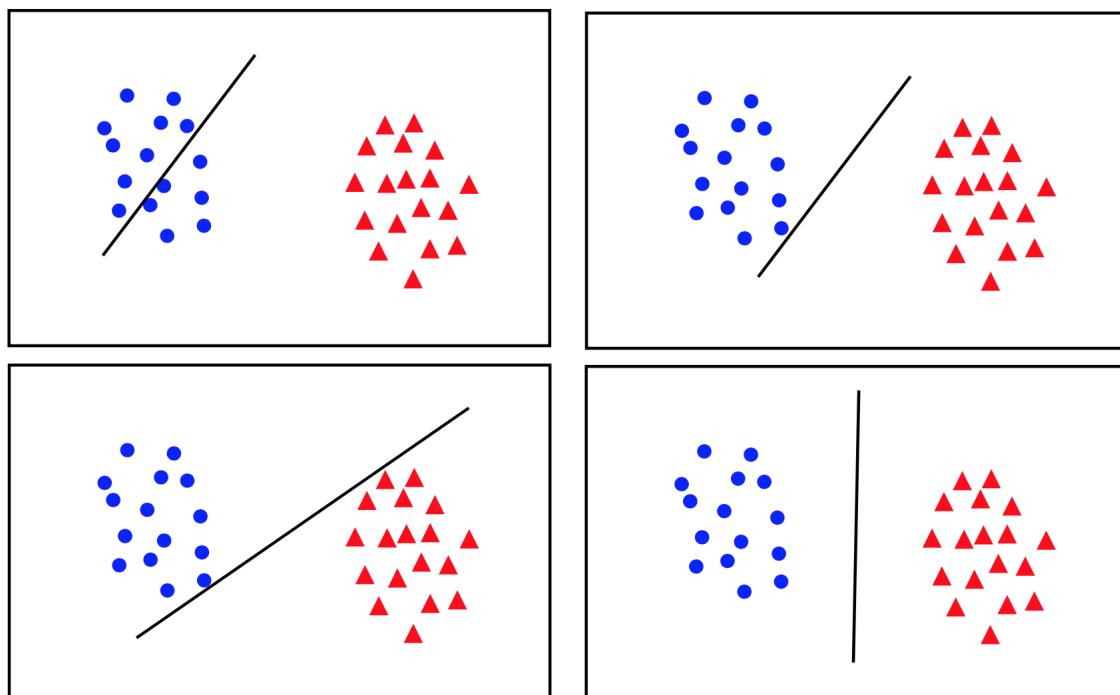
$$\mathcal{S} = \{\mathbf{x} \in \mathbb{R}^m : \mathbf{w}^\top \mathbf{x} + w_0 = 0\}$$

Linear separability (assume for now)



11 / 54

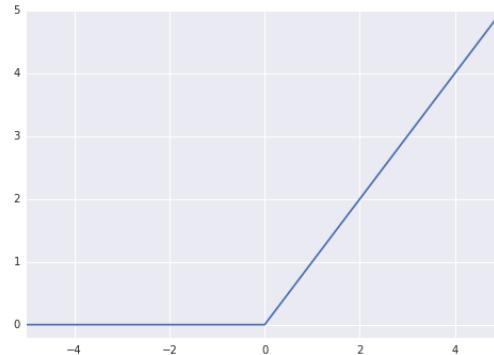
What's a good decision boundary?



12 / 54

Review: Perceptron Algorithm

Hinge loss function (-ve example, x-axis is $h(\mathbf{x}; \mathbf{w}, w_0) = \mathbf{w}^\top \mathbf{x} + w_0$):



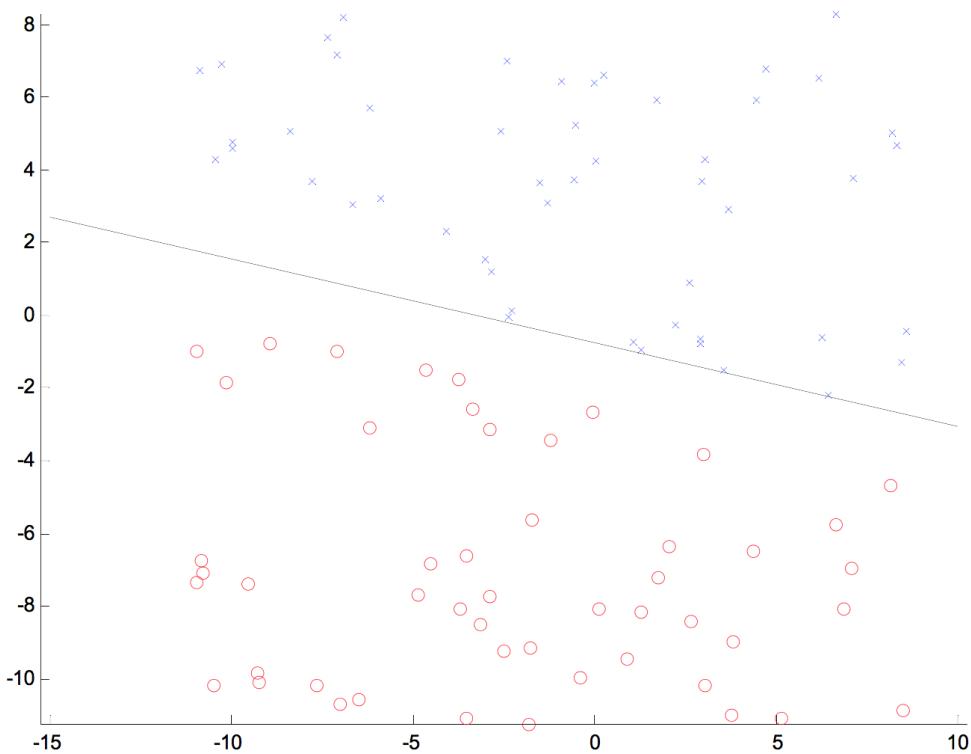
When summed up over all training data:

$$\mathcal{L}(\mathbf{w}) = \sum_{i=1}^n f_{relu}(-h(\mathbf{x}_i; \mathbf{w}, w_0)y_i) = \sum_{i=1: y_i \neq \hat{y}_i}^n -y_i(\mathbf{w}^\top \mathbf{x}_i + w_0)$$

Convex, but can overshoot. Only guaranteed to converge if data is linearly separable, and convergence can be slow.

13 / 54

Percptron also finds “bad” decision boundaries



14 / 54

Contents

[1] Binary Classification

[2] Max-margin methods

[3] Hard Margin Formulation

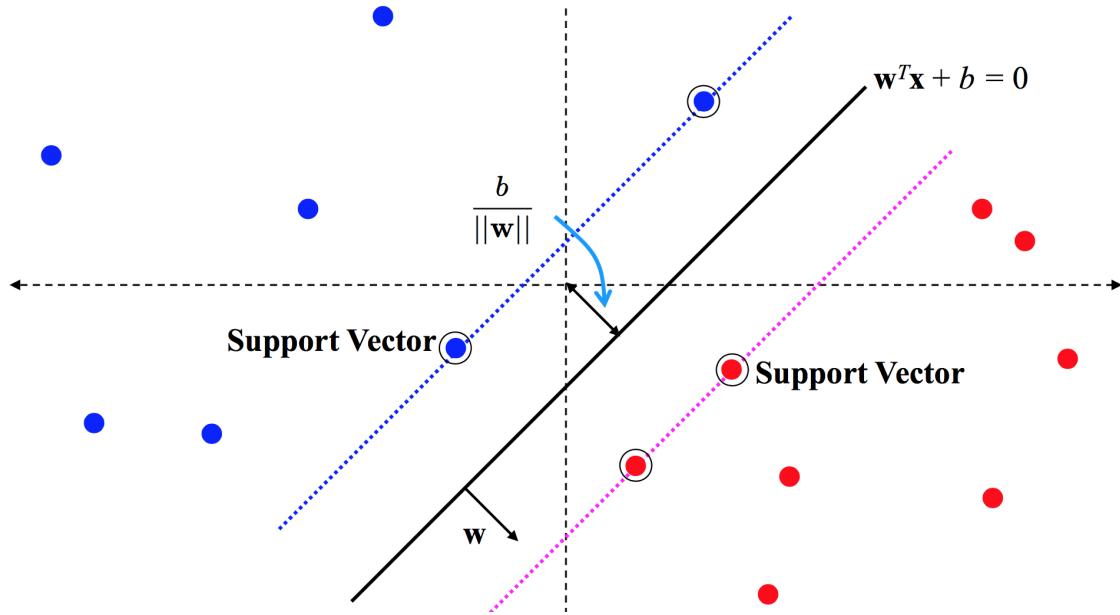
[4] Soft Margin Formulation

[5] Application: Pedestrian Detection

[6] Loss functions Revisited

15 / 54

Max-margin methods



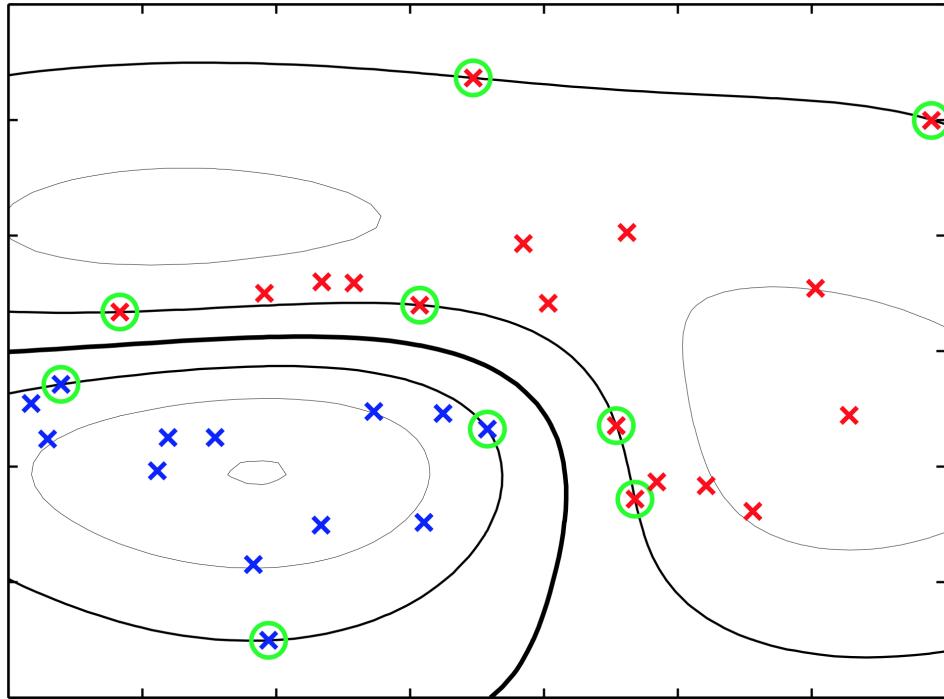
[read b as w_0 .][also ignore ‘support vectors’ for now!]

“margin” == min dist. to decision boundary. Maximize this!

16 / 54

Max-margin + Basis functions

Linear separator in higher dimensional space is non-linear in \mathbb{R}^2 :



17 / 54

Training problem (informal)

- Binary classification, with $\mathbf{x}_i \in \mathbb{R}^m$ and $h(\mathbf{x}; \mathbf{w}, w_0) = \mathbf{w}^\top \mathbf{x} + w_0$.
Predict $y_i \in \{-1, +1\}$
- $$\hat{y} = \begin{cases} 1 & \text{if } h(\mathbf{x}; \mathbf{w}, w_0) \geq 0 \\ -1 & \text{o.w.} \end{cases}$$
- For now: assume data is linearly separable.
- Goal: Find \mathbf{w} to correctly classify examples, and maximize margin on data D .
- Margin == min dist to decision boundary across all examples in D .

18 / 54

Example: Distance to decision boundary

Suppose $\mathbf{x} \in \mathbb{R}^2$

$$h(\mathbf{x}; \mathbf{w}, w_0) = 2x_1 - x_2 - 1.$$

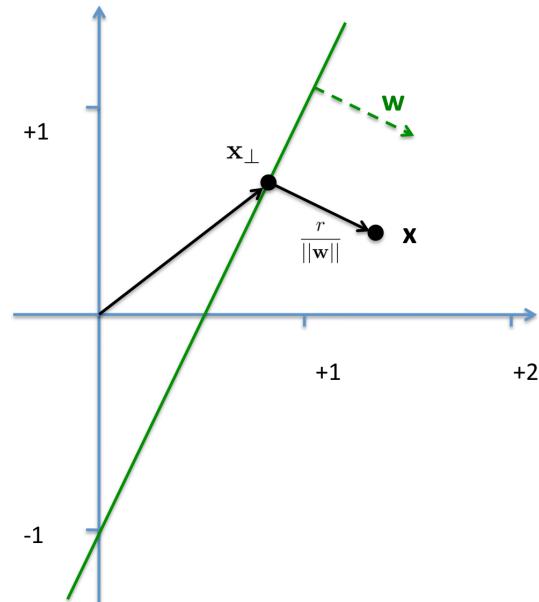
$$\|\mathbf{w}\| = \sqrt{\mathbf{w}^\top \mathbf{w}} = \sqrt{5}.$$

Write example \mathbf{x} as

$$\mathbf{x} = \mathbf{x}_\perp + r \frac{\mathbf{w}}{\|\mathbf{w}\|}, \quad (1)$$

where \mathbf{x}_\perp is projection of \mathbf{x} onto the boundary, and r is the distance.

19 / 54



Solving for distance to decision boundary

We want [signed, normalized, orthogonal distance](#) from an example \mathbf{x} to decision boundary associated with (\mathbf{w}, w_0) .

Calculate as:

$$\begin{aligned} \mathbf{x} &= \mathbf{x}_\perp + r \frac{\mathbf{w}}{\|\mathbf{w}\|} \\ \Leftrightarrow \mathbf{w}^\top \mathbf{x} &= \mathbf{w}^\top \mathbf{x}_\perp + r \frac{\mathbf{w}^\top \mathbf{w}}{\|\mathbf{w}\|} \\ \Leftrightarrow h(\mathbf{x}; \mathbf{w}, w_0) - w_0 &= -w_0 + r \|\mathbf{w}\| \\ \Leftrightarrow r &= \frac{h(\mathbf{x}; \mathbf{w}, w_0)}{\|\mathbf{w}\|} \end{aligned}$$

Lovely! A simple expression. Also, if $r > 0$, predict 1; predict -1 otherwise.

Defining the Margin of a Classifier

The “margin” for an example is the absolute value of distance r . Can write as:

Definition (Margin on an example)

The margin of a classifier on a correctly classified example (\mathbf{x}_i, y_i) is

$$\text{margin}(\mathbf{x}_i) = y_i \times r = \frac{y_i(\mathbf{w}^\top \mathbf{x}_i + w_0)}{\|\mathbf{w}\|} \quad (\geq 0)$$

Definition (Margin on data)

The margin of a classifier on data D is the minimum margin over correctly classified examples $(\mathbf{x}_i, y_i) \in D$.

21 / 54

Contents

- [1] Binary Classification
- [2] Max-margin methods
- [3] Hard Margin Formulation
- [4] Soft Margin Formulation
- [5] Application: Pedestrian Detection
- [6] Loss functions Revisited

22 / 54

The Max Margin Training Problem

$$\max_{\mathbf{w}, w_0} \left[\min_i \frac{1}{\|\mathbf{w}\|} y_i (\mathbf{w}^\top \mathbf{x}_i + w_0) \right] \quad [\text{P1}]$$

Yikes! Non-linear in \mathbf{w} , and ugly “max min” structure.

23 / 54

Simplifying (1 of 3)

$$\max_{\mathbf{w}, w_0} \left[\min_{i=1 \dots n} \frac{1}{\|\mathbf{w}\|} y_i (\mathbf{w}^\top \mathbf{x}_i + w_0) \right] \quad [\text{P1}]$$

We have:

- Separable, and thus in solution, margin $\frac{y_i(\mathbf{w}^\top \mathbf{x}_i + w_0)}{\|\mathbf{w}\|} > 0$ for all i
- Margin $\frac{y_i(\mathbf{w}^\top \mathbf{x}_i + w_0)}{\|\mathbf{w}\|}$ is invariant to multiplying (\mathbf{w}, w_0) by any $\beta > 0$.

Therefore, it does not preclude any solutions to rewrite as:

$$\begin{aligned} \max_{\mathbf{w}, w_0} & \left[\min_{i=1 \dots n} \frac{1}{\|\mathbf{w}\|} y_i (\mathbf{w}^\top \mathbf{x}_i + w_0) \right] \\ \text{s.t. } & y_i (\mathbf{w}^\top \mathbf{x}_i + w_0) \geq 1, \quad \text{for all } i \end{aligned} \quad [\text{P2}]$$

24 / 54

Simplifying (2 of 3)

$$\begin{aligned} & \max_{\mathbf{w}, w_0} \frac{1}{\|\mathbf{w}\|} \min_{i=1 \dots n} y_i (\mathbf{w}^\top \mathbf{x}_i + w_0) \\ \text{s.t. } & y_i (\mathbf{w}^\top \mathbf{x}_i + w_0) \geq 1, \quad \text{for all } i \end{aligned} \tag{P2}$$

is equivalent to

$$\begin{aligned} & \max_{\mathbf{w}, w_0} \frac{1}{\|\mathbf{w}\|} \\ \text{s.t. } & y_i (\mathbf{w}^\top \mathbf{x}_i + w_0) \geq 1, \quad \text{for all } i \end{aligned} \tag{P3}$$

- (1) By the β scaling argument, it would not preclude any solutions to restrict [P2] to insist at least one constraint is binding.
- (2) Optimal solution to [P3] will achieve this, and thus also solve [P2] since objective is the same amongst solutions where at least one constraint is binding.

25 / 54

Simplifying (3 of 3)

$$\begin{aligned} & \max_{\mathbf{w}, w_0} \frac{1}{\|\mathbf{w}\|} \\ \text{s.t. } & y_i (\mathbf{w}^\top \mathbf{x}_i + w_0) \geq 1, \quad \text{for all } i \end{aligned} \tag{P3}$$

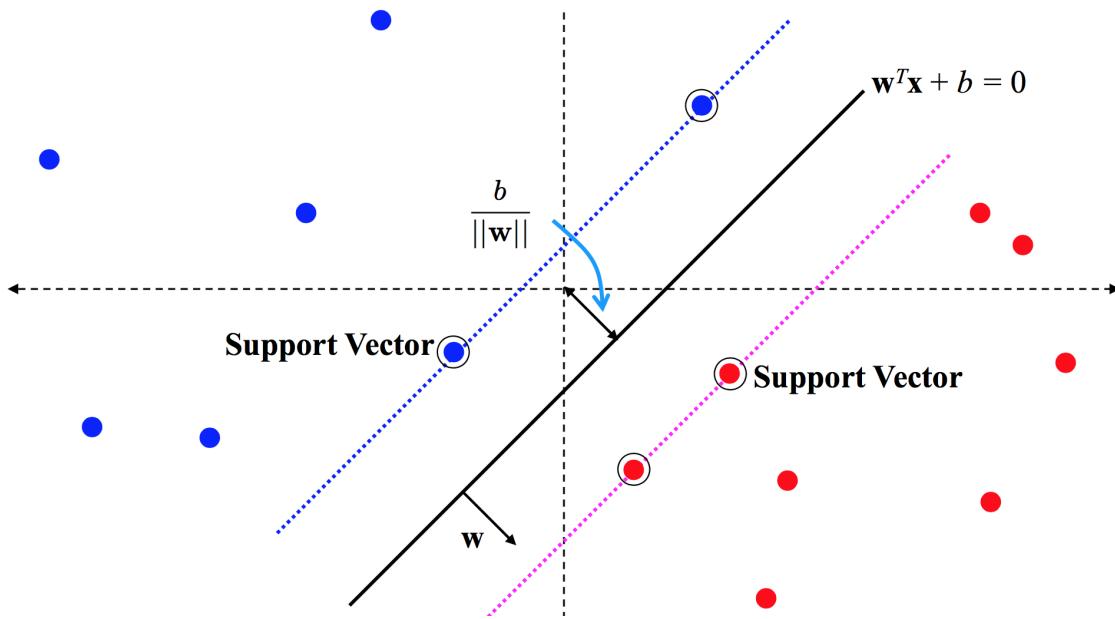
Can rewrite as:

$$\begin{aligned} & \min_{\mathbf{w}, w_0} \frac{1}{2} \|\mathbf{w}\|^2 \\ \text{s.t. } & y_i (\mathbf{w}^\top \mathbf{x}_i + w_0) \geq 1, \quad \text{for all } i \end{aligned} \tag{P4}$$

Nice! This is convex and differentiable, with linear constraints (a “quadratic program”). Can solve via gradient descent. Obtain global minimum.

26 / 54

Interpretation: Max-Margin Solution



[read b as w_0 .][also ignore ‘support vectors’ for now!]

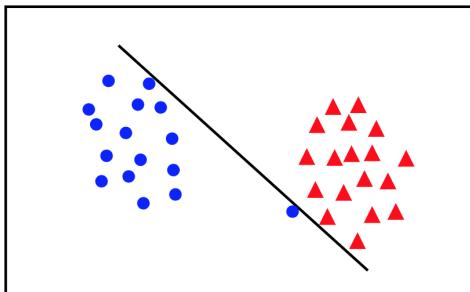
Normalization: bottom margin boundary has $w^T x + w_0 = +1$, top has $w^T x = -1$. “margin region” == between margin boundaries.

27 / 54

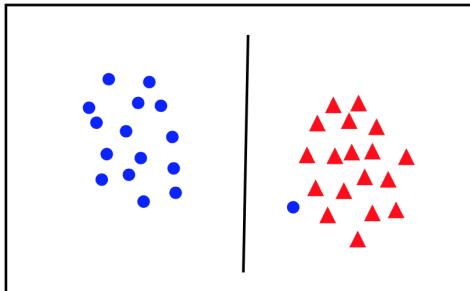
Contents

- [1] Binary Classification
- [2] Max-margin methods
- [3] Hard Margin Formulation
- [4] Soft Margin Formulation
- [5] Application: Pedestrian Detection
- [6] Loss functions Revisited

Linear separability revisited.



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

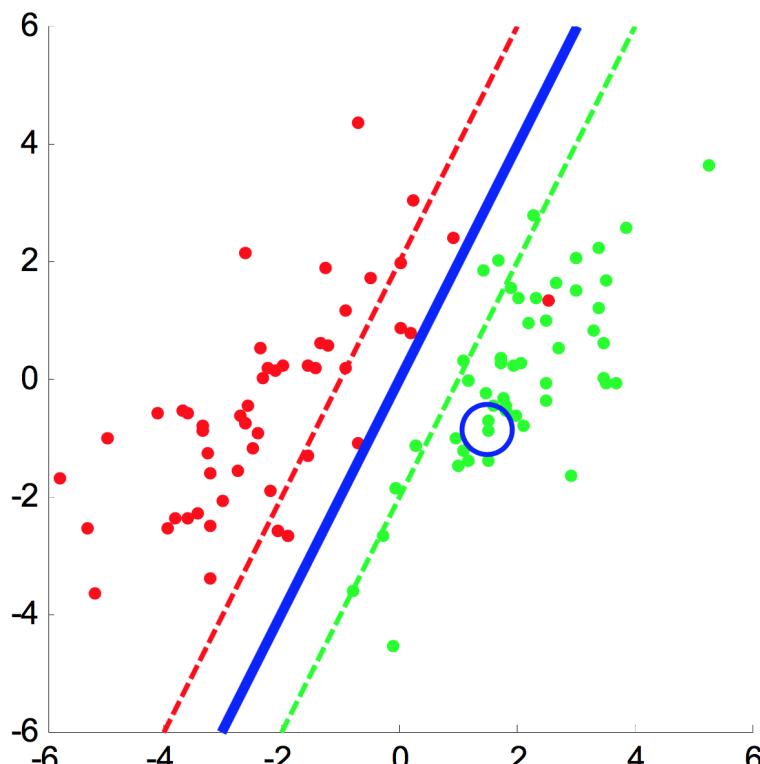


- but possibly the large margin solution is better, even though one constraint is violated

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

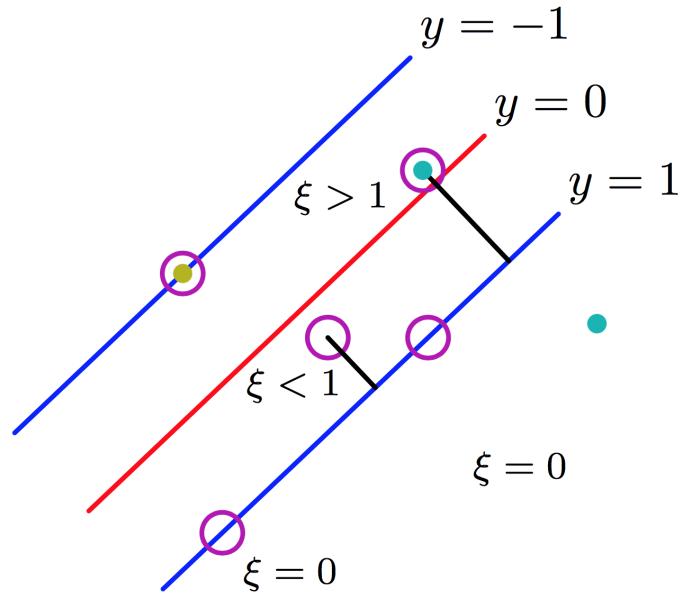
29 / 54

Moreover, training data may not be linearly separable!



30 / 54

A relaxed formulation



[read y as h]

Introduce variable $\xi_i \geq 0$, for each i , to represent how much constraint is violated on example i .

31 / 54

The Soft Margin Formulation

$$\begin{array}{ll} \min_{\mathbf{w}, w_0} \frac{1}{2} \|\mathbf{w}\|^2 & [\text{P4}] \\ \text{s.t. } y_i(\mathbf{w}^\top \mathbf{x}_i + w_0) \geq 1 & \end{array} \quad \begin{array}{ll} \min_{\mathbf{w}, w_0} \frac{1}{2} \|\mathbf{w}\|^2 + C \sum_{i=1}^n \xi_i & [\text{P5}] \\ \text{s.t. } y_i(\mathbf{w}^\top \mathbf{x}_i + w_0) \geq 1 - \xi_i \\ \xi_i \geq 0 & \end{array}$$

Constant $C > 0$ is a regularization parameter:

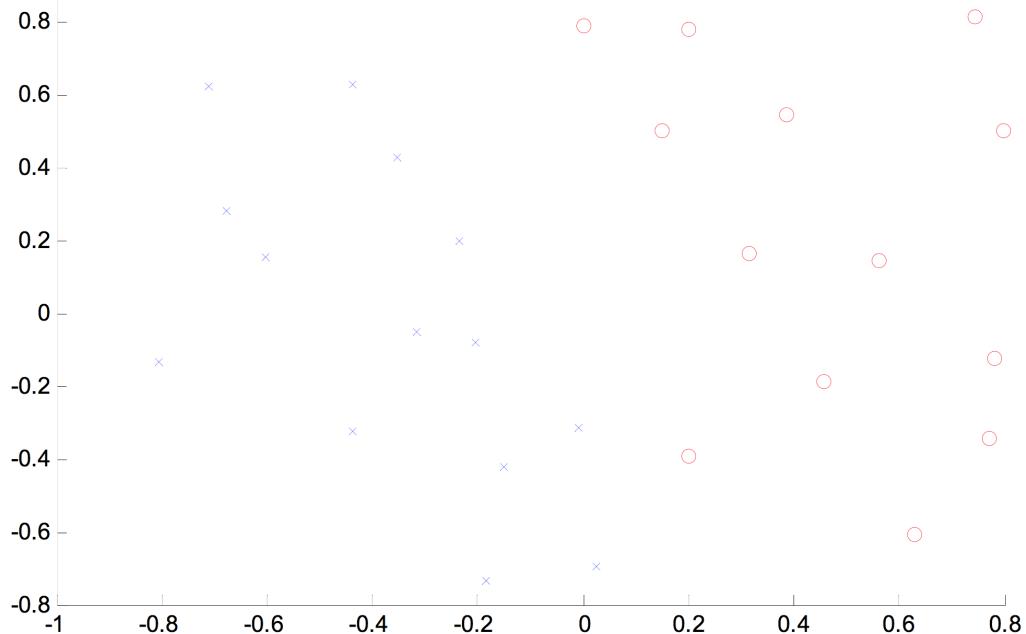
- small C , can ignore constraints, get larger margin.
- large C , constraints hard to ignore, get smaller margin.
- $C = \infty$ enforces all constraints.

Set C via cross-validation (careful, larger is less regularization).

32 / 54

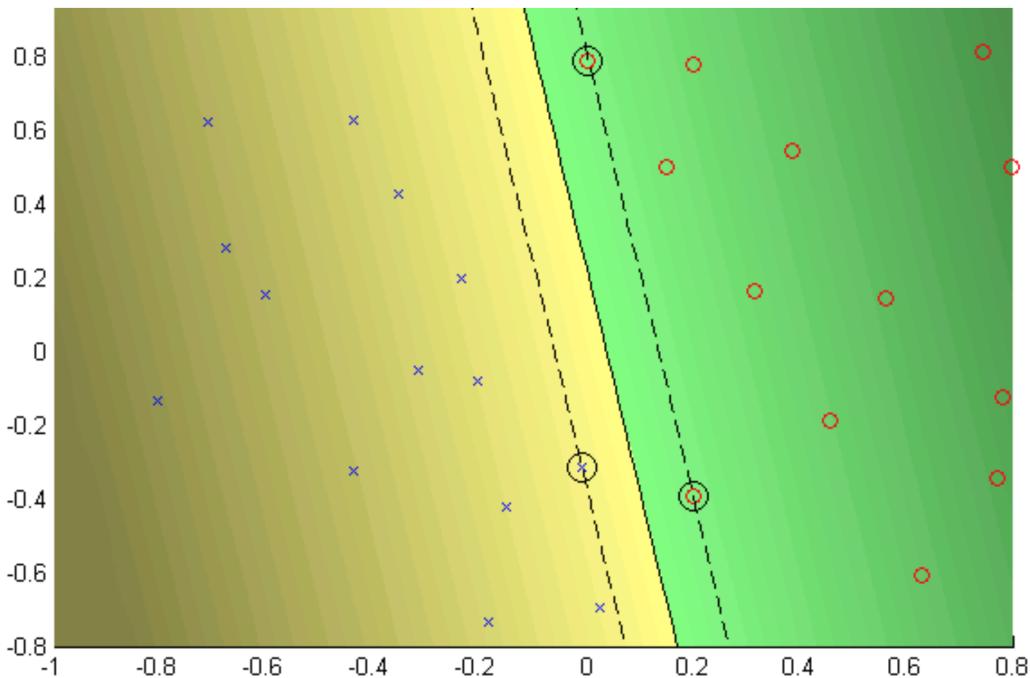
Example: Soft Margin Classifier

Data linearly separable, but only with a small margin.

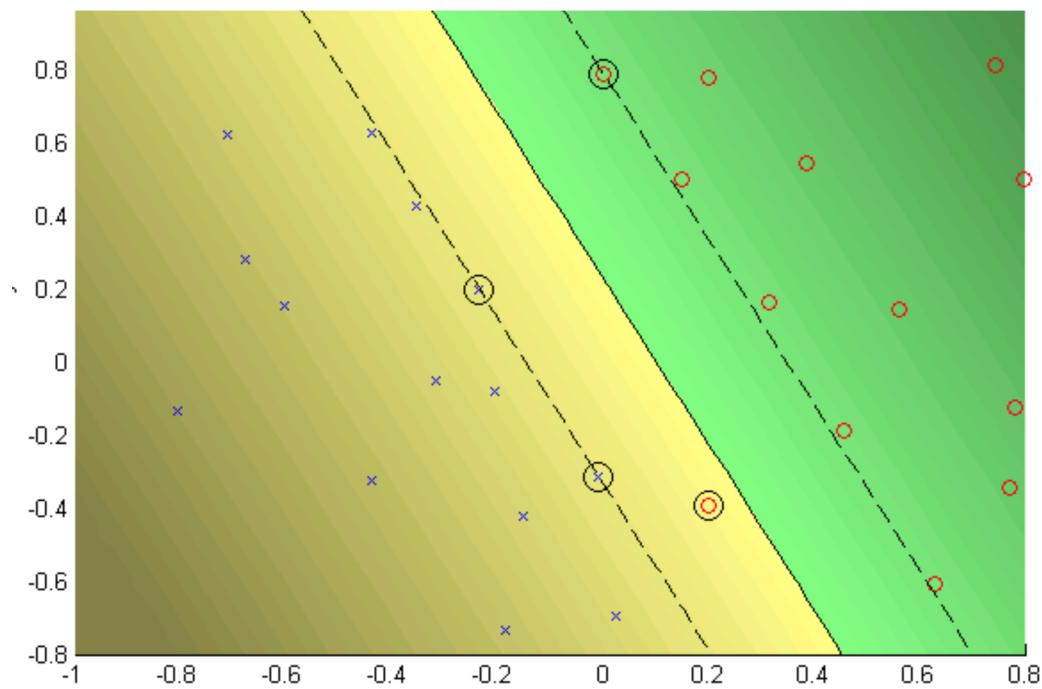


33 / 54

$C = \text{Infinity}$ (hard margin)



34 / 54



35 / 54

Contents

- [1] Binary Classification
- [2] Max-margin methods
- [3] Hard Margin Formulation
- [4] Soft Margin Formulation
- [5] Application: Pedestrian Detection
- [6] Loss functions Revisited

36 / 54

Application: Pedestrian detection in computer vision

Objective: detect standing humans in an image window.



[Dalal and Triggs'05]

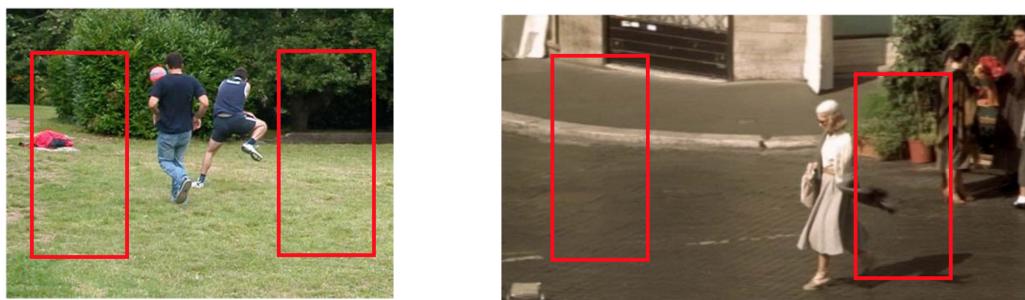
37 / 54

Training data and features

■ Positive window examples

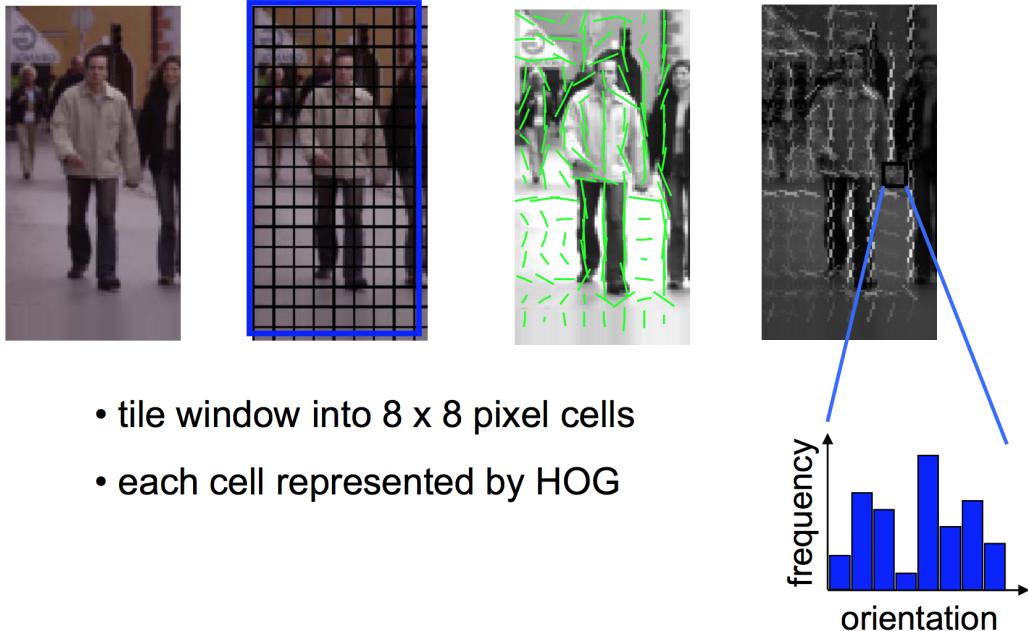


■ Negative window examples



38 / 54

Feature encoding: Histogram of oriented gradients (HOG)

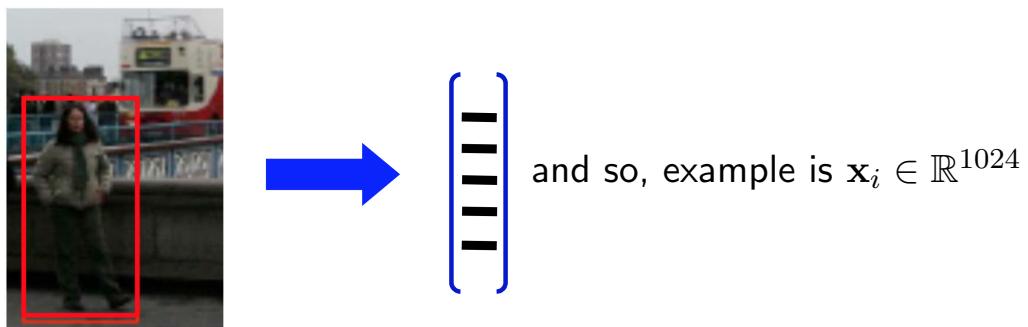


Number features is $d = 16 \times 8$ (tiling) $\times 8$ (orientations) = 1024.

39 / 54

Max-margin algorithm

Represent each example window by a HOG feature vector



Train a Support Vector Machine. Predict a standing human based on

$$\hat{y} = \begin{cases} 1 & \text{if } h(\mathbf{x}; \mathbf{w}, w_0) > 0 \\ -1 & \text{o.w.} \end{cases}$$

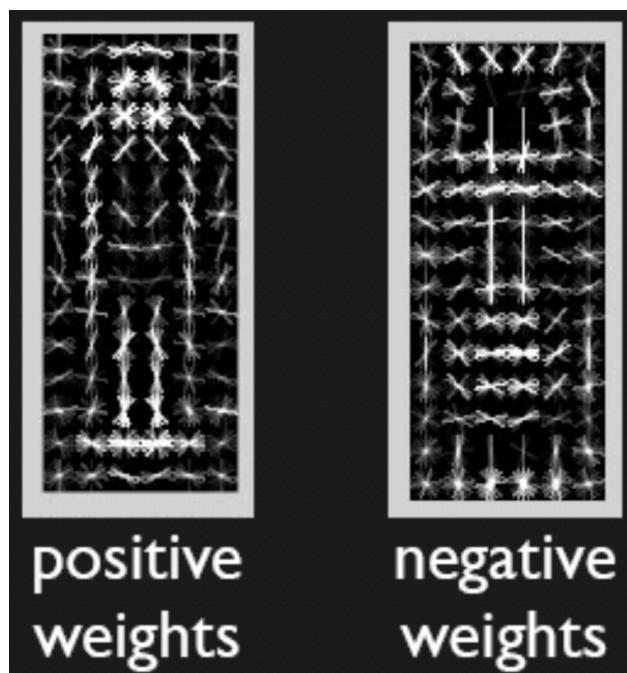
40 / 54

Prediction on a new example



41 / 54

Example Weights in Trained Model

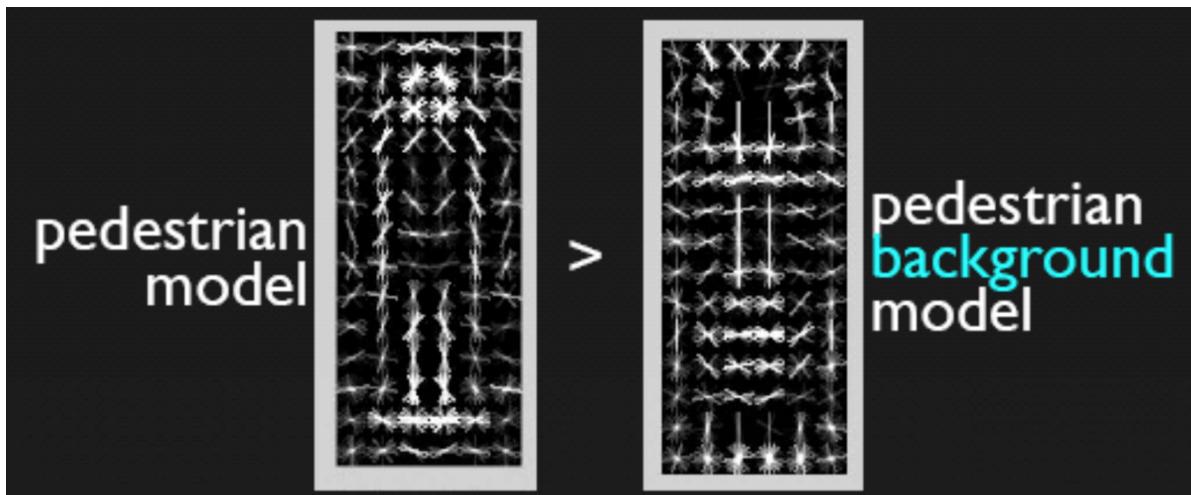


[graphic: D. Ramanan]

42 / 54

Interpretation of the Classifier

Will predict a standing human if:



[graphic: D. Ramanan]

43 / 54

Contents

- [1] Binary Classification
- [2] Max-margin methods
- [3] Hard Margin Formulation
- [4] Soft Margin Formulation
- [5] Application: Pedestrian Detection
- [6] Loss functions Revisited

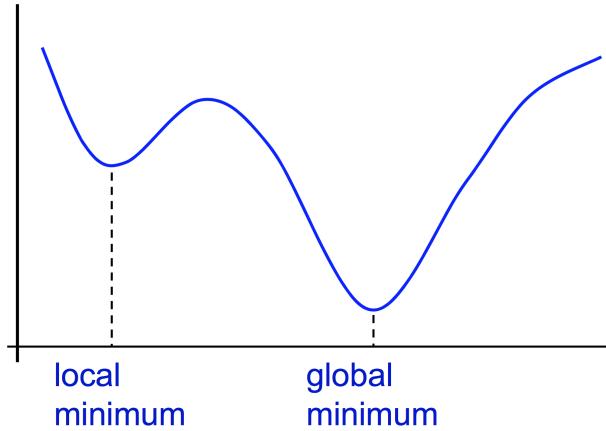
44 / 54

The Soft-Margin Optimization Problem

We ended up here:

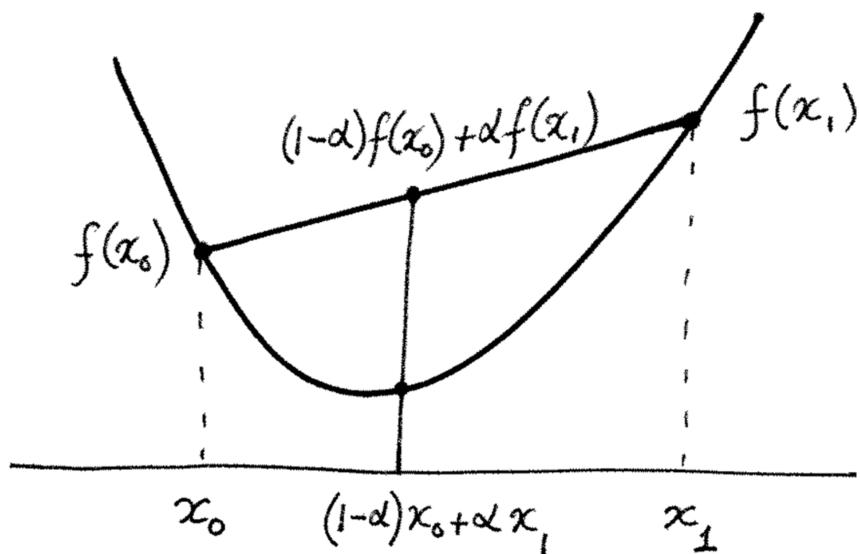
$$\min_{\mathbf{w}, w_0} C \sum_{i=1}^n \max(0, 1 - y_i(\mathbf{w}^\top \mathbf{x}_i + w_0)) + \frac{1}{2} \|\mathbf{w}\|^2 \quad [\text{P5}]$$

Is this loss function convex (i.e., will a locally optimal point be globally optimal?)



45 / 54

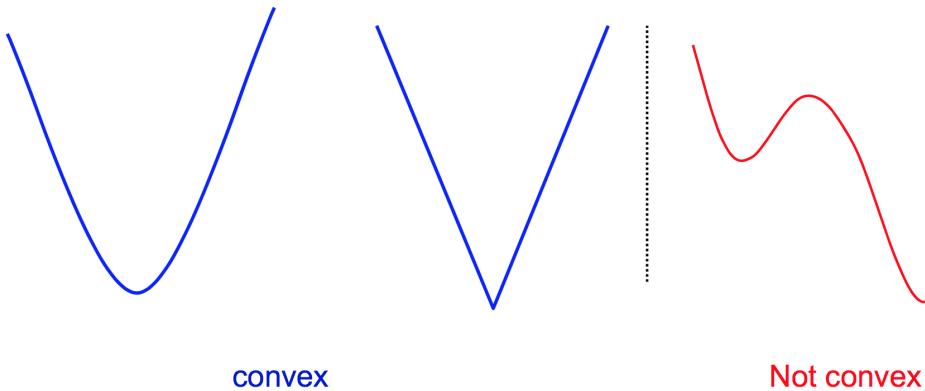
Review: Convex functions



Line joining $(x_0, f(x_0))$ and $(x_1, f(x_1))$ lies above the function graph.

46 / 54

Example functions

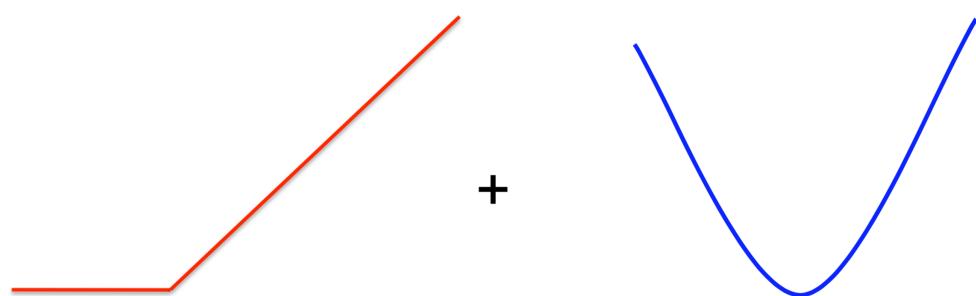


Fact: A non-negative sum of convex functions is convex.

47 / 54

The Soft-Margin Loss function

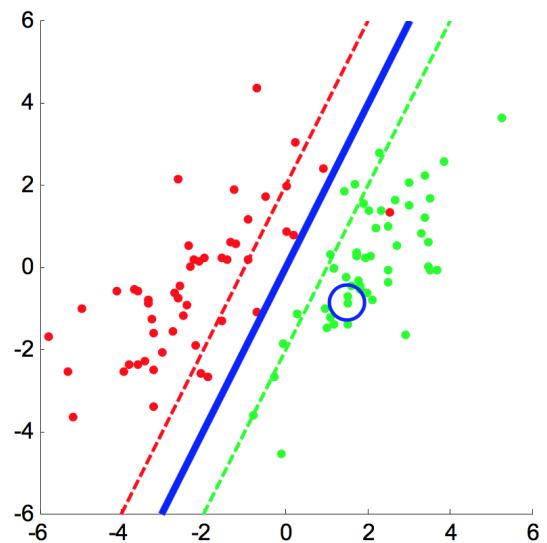
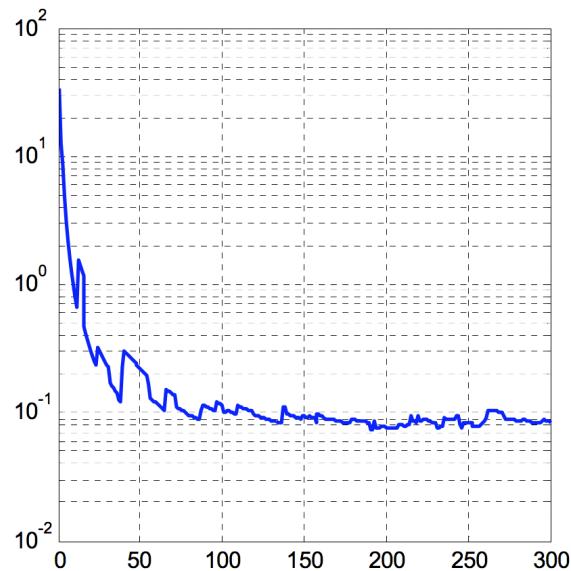
$$\min_{\mathbf{w}, w_0} C \sum_{i=1}^n \max(0, 1 - y_i(\mathbf{w}^\top \mathbf{x}_i + w_0)) + \frac{1}{2} \|\mathbf{w}\|^2$$



Convex! Can derive SGD updates to optimize.

48 / 54

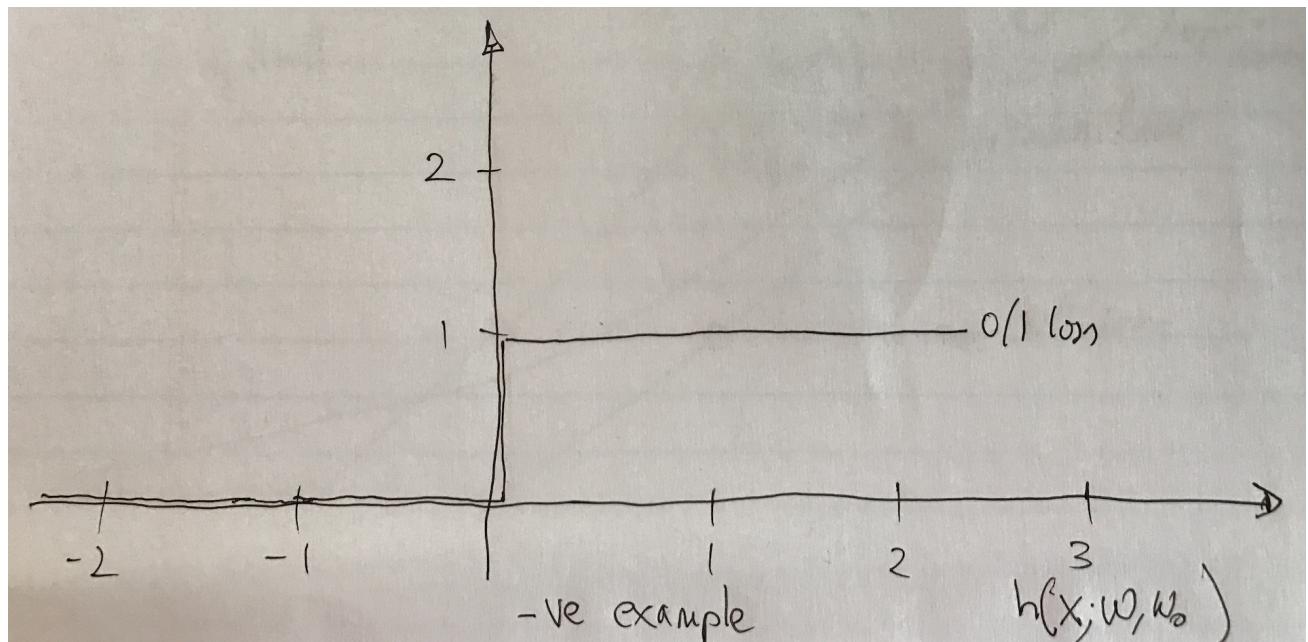
SGD based, Soft-margin training



Pegasos algorithm, Shalev-Shwartz et al. 2011.

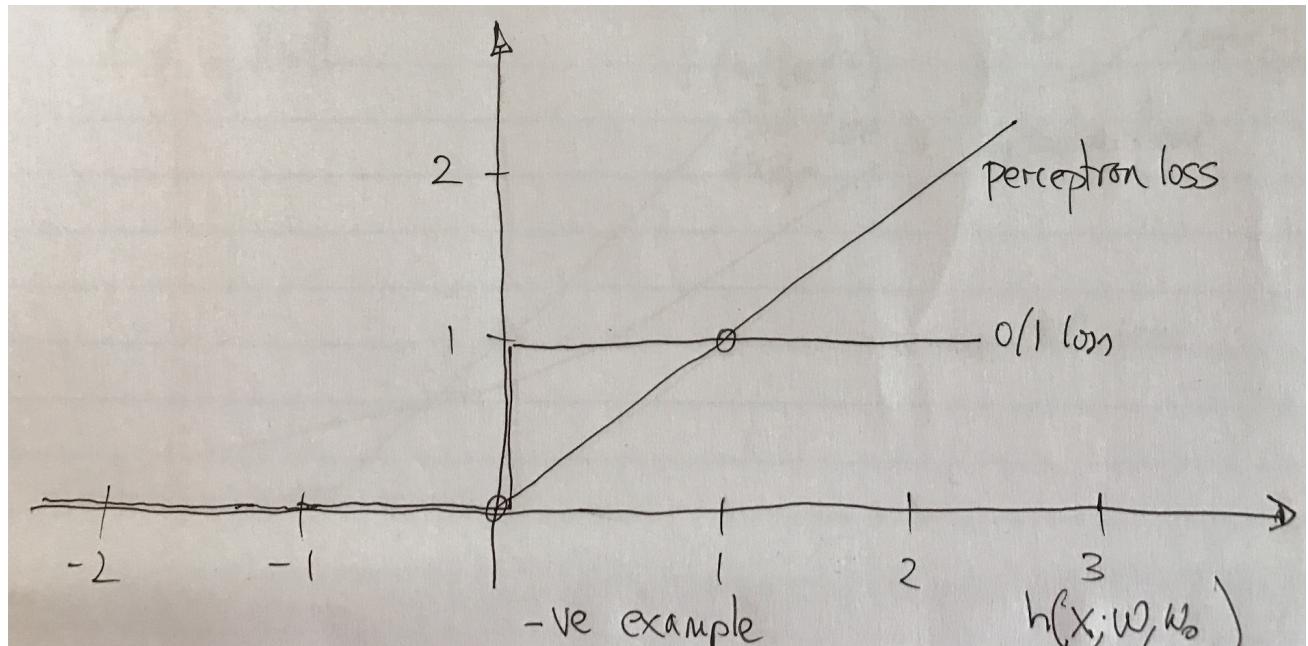
49 / 54

Comparing Loss Functions



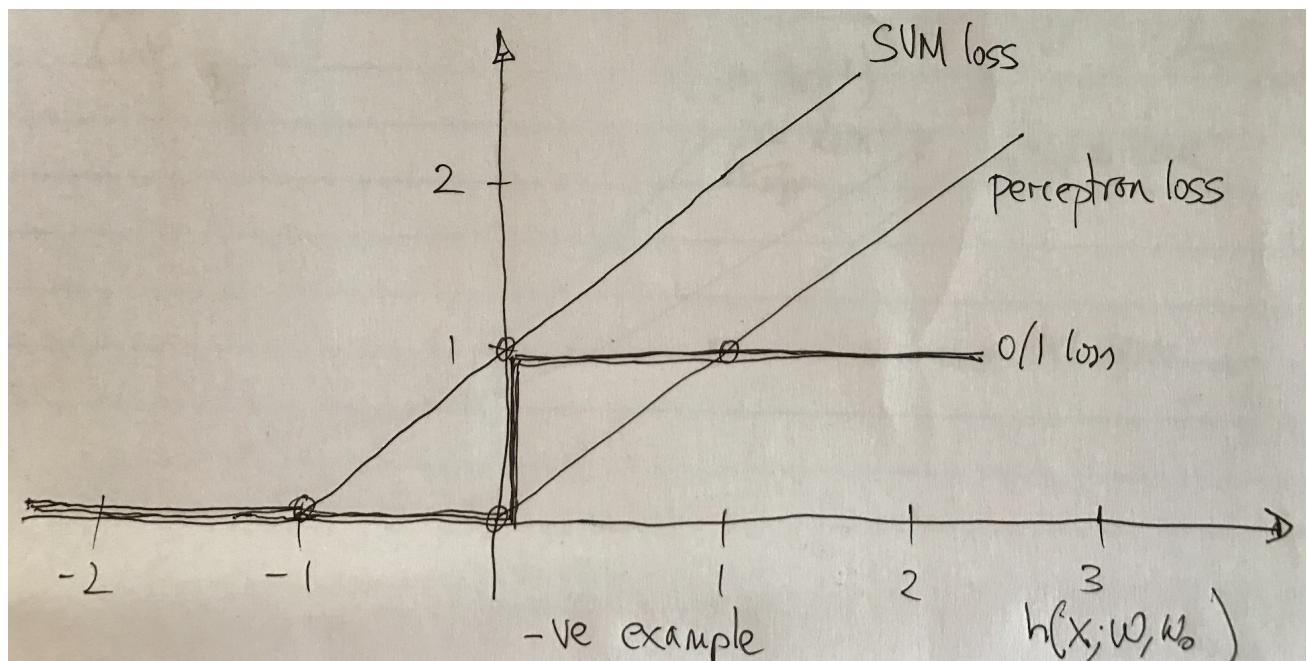
50 / 54

Comparing Loss Functions



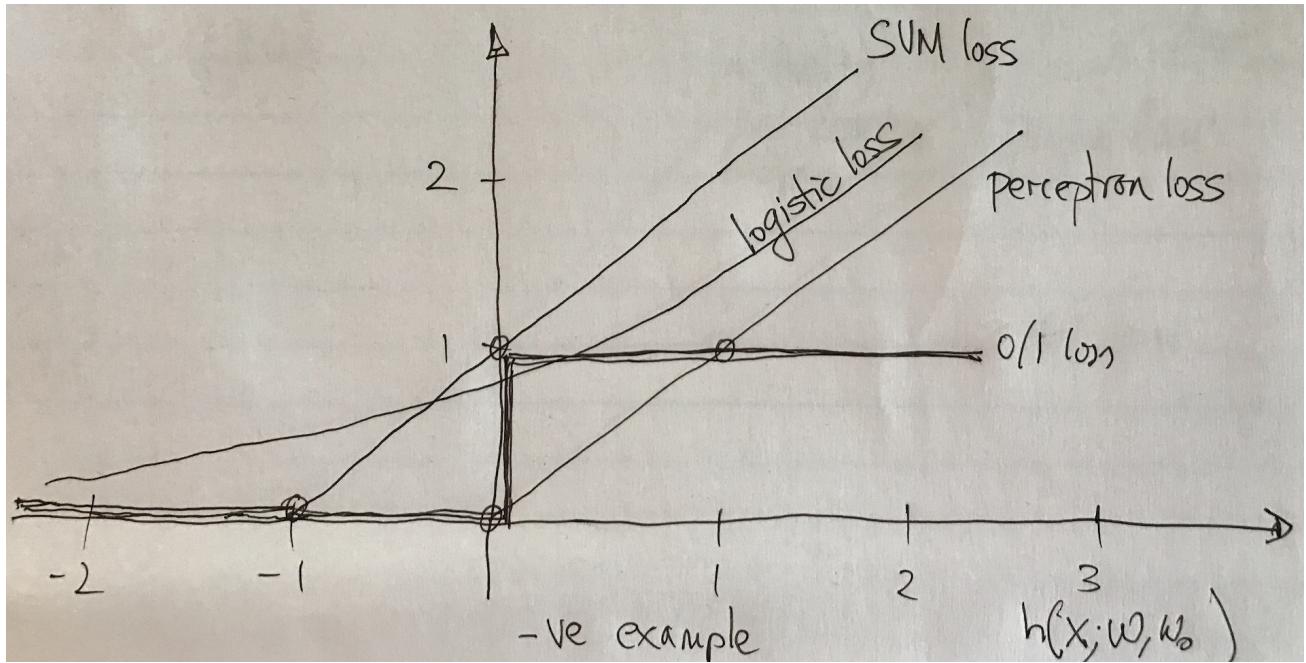
51 / 54

Comparing Loss Functions



52 / 54

Comparing Loss Functions



53 / 54

Next class

- Limitation of current formulation is that $w \in \mathbb{R}^m$ and m may be very large (especially if $m \rightarrow d$ following basis transform).
- Can solve [P5] via a dual formulation.
- Allows for the “kernel trick.” A beautiful way to work with basis functions.
- Leads to the “dual form of the predictor”, which is a succinct representation via “support vectors.”
- Understanding kernel functions.

54 / 54