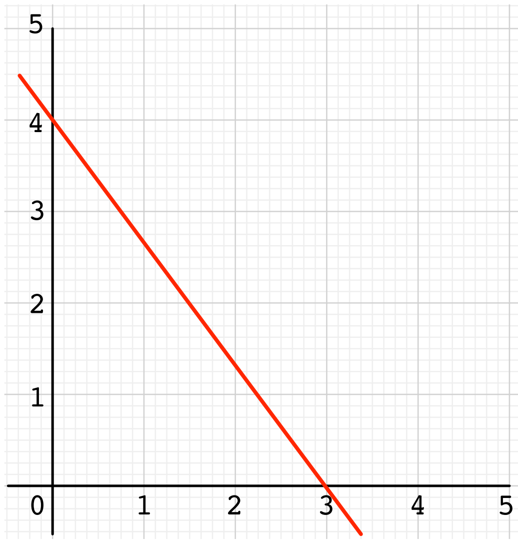
Linear classification

Topics we'll cover

1. Linear decision boundary for binary classification
2. The Perceptron algorithm
3. Maximizing the margin
4. The soft-margin SVM

Linear decision boundary for classification: example



- What is the formula for this boundary?
- What label would we predict for a new point x?

Linear classifiers

Binary classification problem: data $x \in \mathbb{R}^d$ and labels $y \in \{-1, +1\}$

- Linear classifier:
  - Parameters: $w \in \mathbb{R}^d$ and $b \in \mathbb{R}$
  - Decision boundary $w \cdot x + b = 0$
  - On point x, predict label $\text{sign}(w \cdot x + b)$
- If the true label on point x is y:
  - Classifier correct if $y(w \cdot x + b) > 0$

A loss function for classification

What is the **loss** of our linear classifier (given by w, b) on a point (x, y)?

One idea for a loss function:

- If $y(w \cdot x + b) > 0$: correct, no loss
- If $y(w \cdot x + b) < 0$: loss $= -y(w \cdot x + b)$

A simple learning algorithm

Fit a linear classifier w, b to the training set using **stochastic gradient descent**.

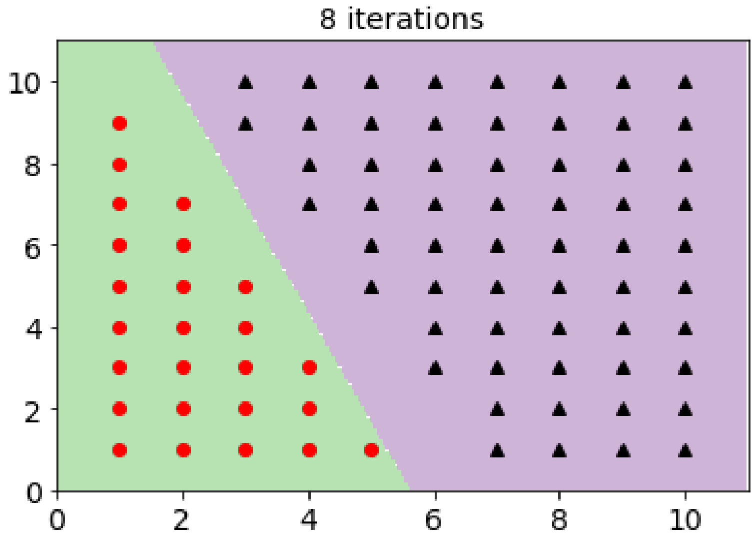
- Update w, b based on just one data point (x, y) at a time
- If $y(w \cdot x + b) > 0$: zero loss, no update
- If $y(w \cdot x + b) \leq 0$: loss is $-y(w \cdot x + b)$

The Perceptron algorithm

- Initialize $w = 0$ and $b = 0$
- Keep cycling through the training data (x, y):
    - If $y(w \cdot x + b) \leq 0$ (i.e. point misclassified):
        - $w = w + yx$
        - $b = b + y$

The Perceptron in action

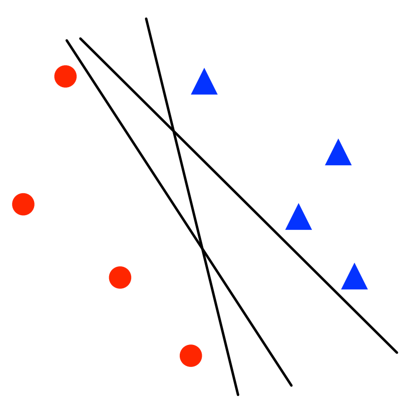
85 data points, linearly separable.



Perceptron: convergence

If the training data is linearly separable:

- The Perceptron algorithm will find a linear classifier with zero training error
- It will converge within a finite number of steps.



But is there a better, more systematic choice of separator?

The hard-margin support vector machine

1. The margin of a linear classifier
2. Maximizing the margin
3. A convex optimization problem
4. Support vectors

The learning problem

Given: training data $(x^{(1)}, y^{(1)}), \ldots, (x^{(n)}, y^{(n)}) \in \mathbb{R}^d \times \{-1, +1\}$.

Find: $w \in \mathbb{R}^d$ and $b \in \mathbb{R}$ such that $y^{(i)}(w \cdot x^{(i)} + b) > 0$ for all i.

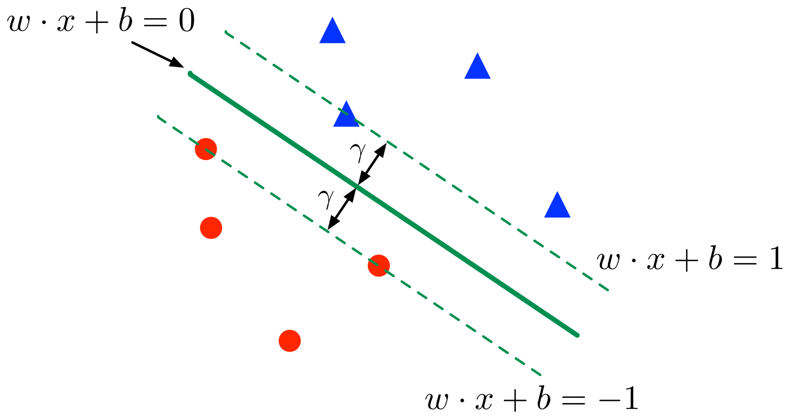
By scaling w, b, can equivalently ask for

$$y^{(i)}(w \cdot x^{(i)} + b) \geq 1 \quad \text{for all } i$$

Maximizing the margin

Given: training data $(x^{(1)}, y^{(1)}), \ldots, (x^{(n)}, y^{(n)}) \in \mathbb{R}^d \times \{-1, +1\}$.
Find: $w \in \mathbb{R}^d$ and $b \in \mathbb{R}$ such that

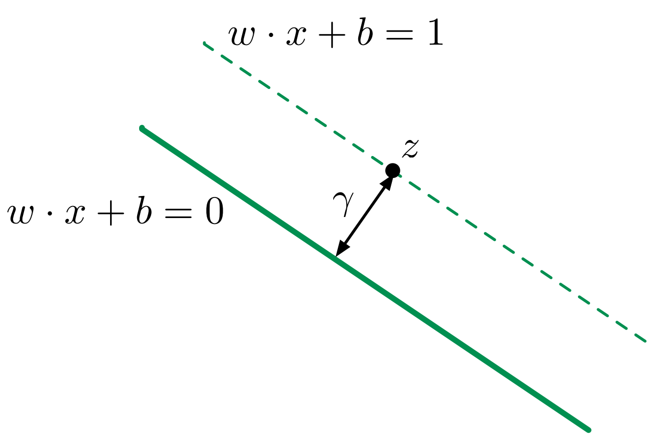
$$y^{(i)}(w \cdot x^{(i)} + b) \geq 1 \quad \text{for all } i.$$



Maximize the **margin** γ.

A formula for the margin

Close-up of a point z on the positive boundary.



A quick calculation shows that $\gamma = 1/\|w\|$.

In short: to maximize the margin, minimize $\|w\|$.

Maximum-margin linear classifier

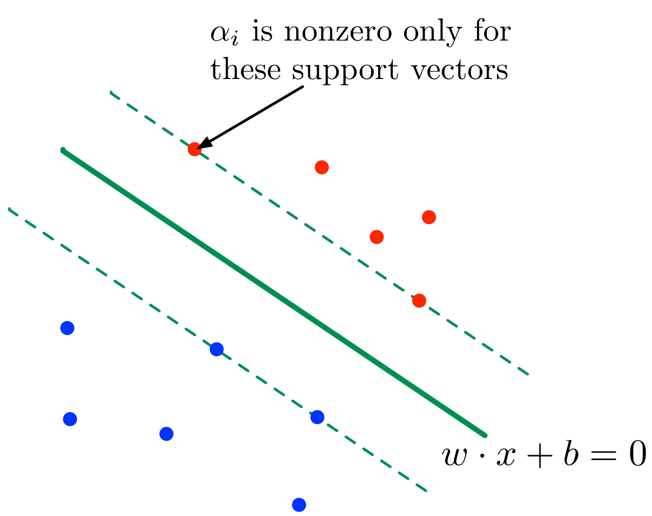
- Given $(x^{(1)}, y^{(1)}), \ldots, (x^{(n)}, y^{(n)}) \in \mathbb{R}^d \times \{-1, +1\}$

$$\boxed{\begin{array}{l} \displaystyle\min_{w \in \mathbb{R}^d, b \in \mathbb{R}} \quad \|w\|^2 \\ \text{s.t.:} \quad y^{(i)}(w \cdot x^{(i)} + b) \geq 1 \quad \text{for all } i = 1, 2, \ldots, n \end{array}}$$

- This is a **convex optimization problem**:
  - Convex objective function
  - Linear constraints
- This means that:
  - the optimal solution can be found efficiently
  - **duality** gives us information about the solution

Support vectors

Support vectors: training points that lie exactly on the margin, i.e. $y^{(i)}(w \cdot x^{(i)} + b) = 1$.



$w = \sum_{i=1}^{n} \alpha_i y^{(i)} x^{(i)}$ is a function of just the support vectors.

Small example: Iris data set

Fisher's **iris** data

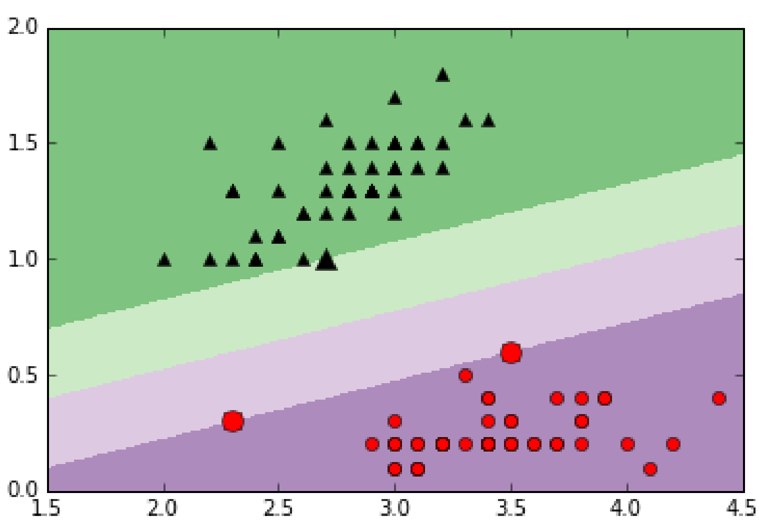
150 data points from three classes:

- iris setosa
- iris versicolor
- iris virginica

Four measurements: petal width/length, sepal width/length

Small example: Iris data set

Two features: sepal width, petal width.
Two classes: setosa (red circles), versicolor (black triangles)



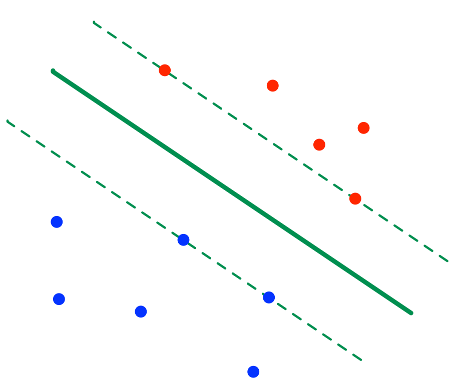
The soft-margin support vector machine

1. Data that isn't linearly separable
2. Adding slack variables for each point
3. Revised convex optimization problem
4. Setting the slack parameter

Recall: maximum-margin linear classifier

Given: $(x^{(1)}, y^{(1)}), \ldots, (x^{(n)}, y^{(n)}) \in \mathbb{R}^d \times \{-1, +1\}$.
Find: the linear separator w that perfectly classifies the data and has maximum margin.

$$\boxed{\begin{array}{l} \displaystyle\min_{w \in \mathbb{R}^d, b \in \mathbb{R}} \quad \|w\|^2 \\ \text{s.t.: } \; y^{(i)}(w \cdot x^{(i)} + b) \geq 1 \quad \text{for all } i = 1, 2, \ldots, n \end{array}}$$



Solution $w = \sum_{i=1}^n \alpha_i y^{(i)} x^{(i)}$ is a function of just the support vectors.

What if data is not separable?

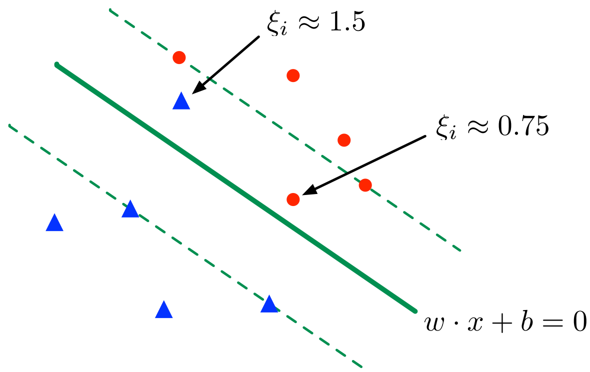
The non-separable case

Idea: allow each data point $x^{(i)}$ some **slack** ξ_i.

$$\min_{w \in \mathbb{R}^d, b \in \mathbb{R}, \xi \in \mathbb{R}^n} \quad \|w\|^2 + C \sum_{i=1}^{n} \xi_i$$

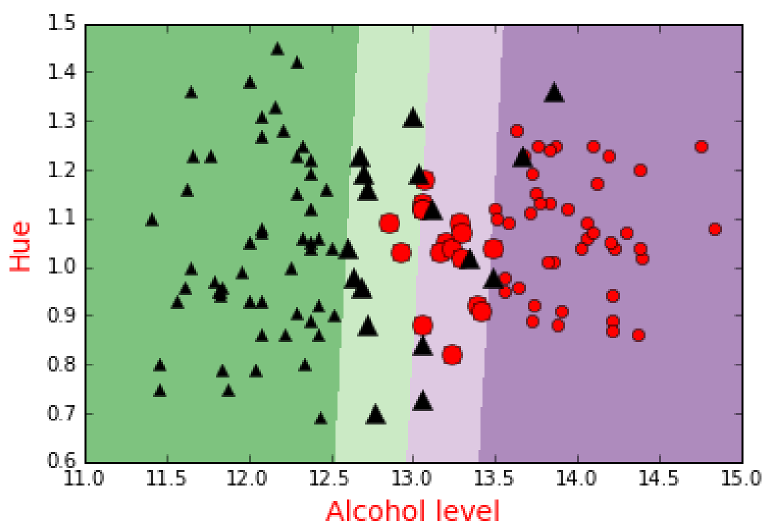
$$\text{s.t.:} \quad y^{(i)}(w \cdot x^{(i)} + b) \geq 1 - \xi_i \quad \text{for all } i = 1, 2, \ldots, n$$

$$\xi \geq 0$$



Wine data set

Here $C = 1.0$



The tradeoff between margin and slack

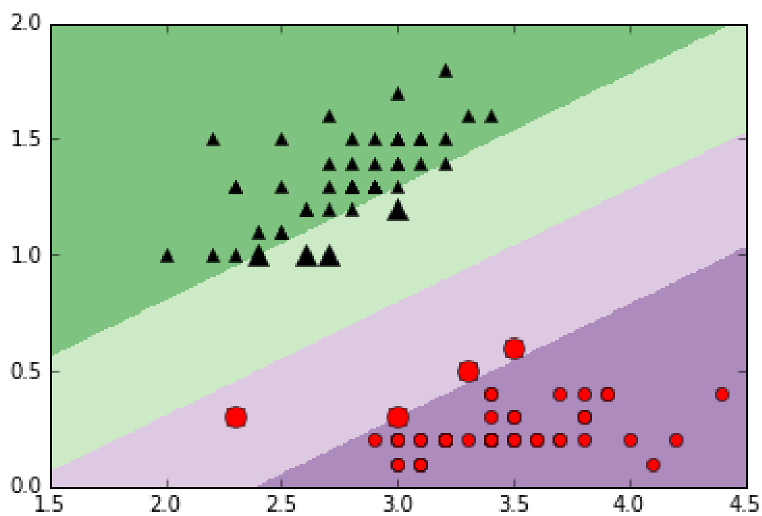
$$\min_{w \in \mathbb{R}^d, b \in \mathbb{R}, \xi \in \mathbb{R}^n} \quad \|w\|^2 + C \sum_{i=1}^{n} \xi_i$$

$$\text{s.t.:} \quad y^{(i)}(w \cdot x^{(i)} + b) \geq 1 - \xi_i \quad \text{for all } i = 1, 2, \ldots, n$$

$$\xi \geq 0$$

Back to Iris

$C = 1$



Sentiment data

Sentences from reviews on Amazon, Yelp, IMDB, each labeled as positive or negative.

- `Needless to say, I wasted my money.`
- `He was very impressed when going from the original battery to the extended battery.`
- `I have to jiggle the plug to get it to line up right to get decent volume.`
- `Will order from them again!`

Data details:

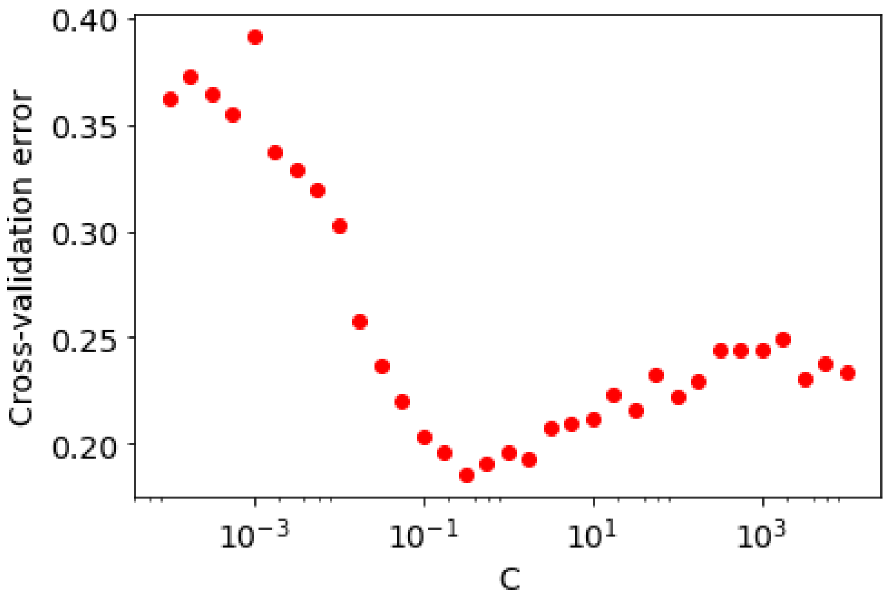
- Bag-of-words representation using a vocabulary of size 4500
- 2500 training sentences, 500 test sentences

What C to use?

C	training error (%)	test error (%)	# support vectors
0.01	23.72	28.4	2294
0.1	7.88	18.4	1766
1	1.12	16.8	1306
10	0.16	19.4	1105
100	0.08	19.4	1035
1000	0.08	19.4	950

Cross-validation

Results of 5-fold cross-validation:



Chose $C = 0.32$. Test error: 15.6%