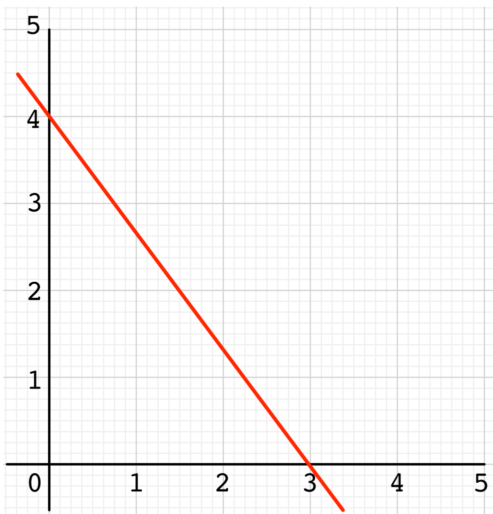
A simple linear classifier

Topics we'll cover

1. Linear decision boundary for binary classification
2. A loss function for classification
3. The Perceptron algorithm

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**.

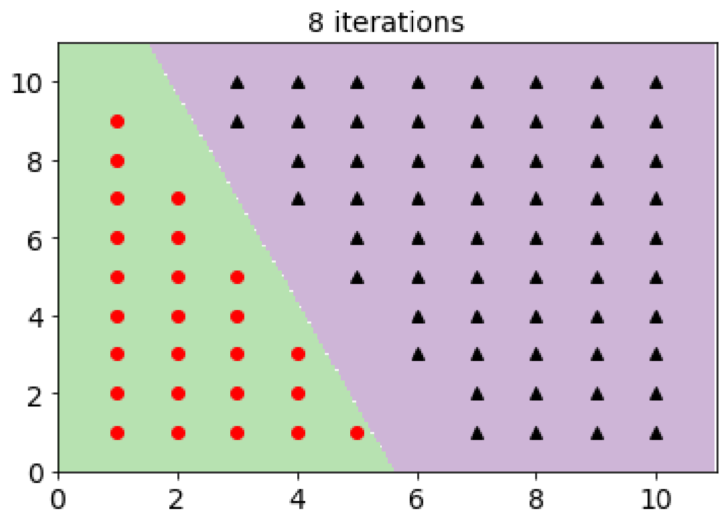
- Update w, b based on just one data point (x, y) at a time
- If $y(w \cdot x + b) > 0$: zero loss, no update
- If $y(w \cdot x + b) \leq 0$: loss is $-y(w \cdot x + b)$

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

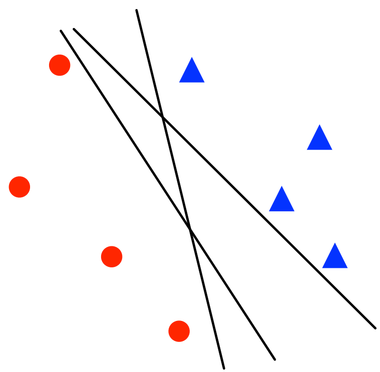
85 data points, linearly separable.



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?

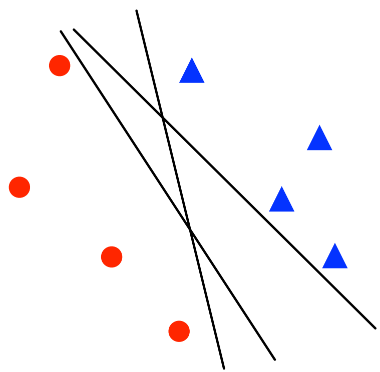
Support vector machines I:
Maximum-margin linear classifiers

Topics we'll cover

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

Improving upon the Perceptron

For a linearly separable data set, there are in general many possible separating hyperplanes, and Perceptron is guaranteed to find one of them.



Is there a better, more systematic choice of separator?
The one with the most buffer around it, for instance?

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.

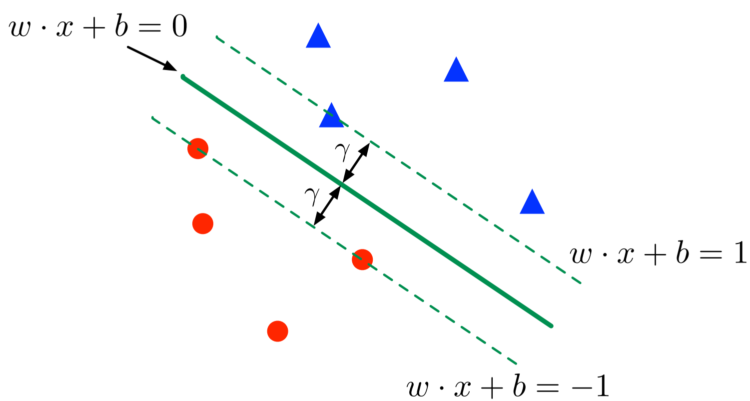
By scaling w, b, can equivalently ask for

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

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

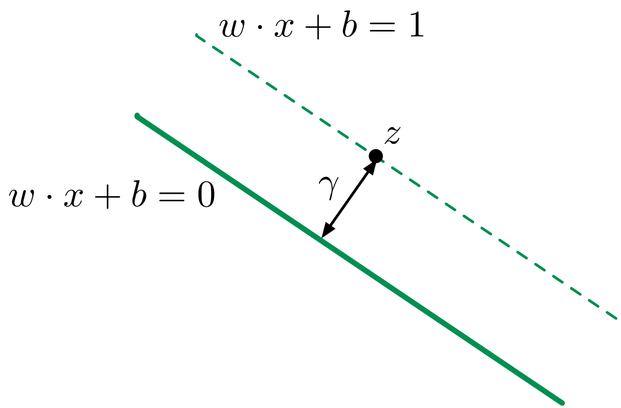
$$y^{(i)}(w \cdot x^{(i)} + b) \geq 1 \quad \text{for all } i.$$



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

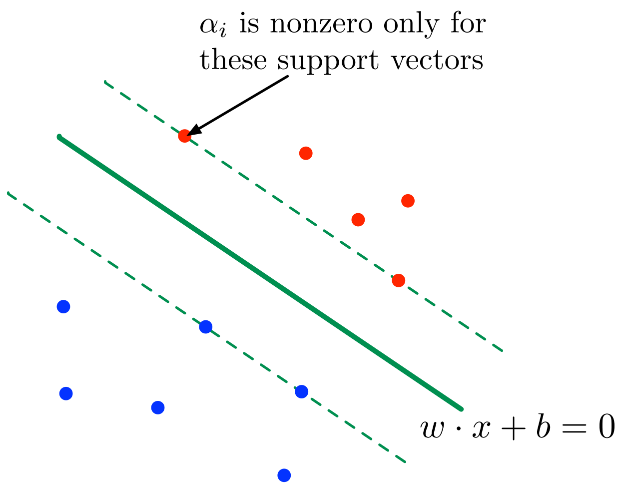
- Given $(x^{(1)}, y^{(1)}), \ldots, (x^{(n)}, y^{(n)}) \in \mathbb{R}^d \times \{-1, +1\}$

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

- 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 right 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



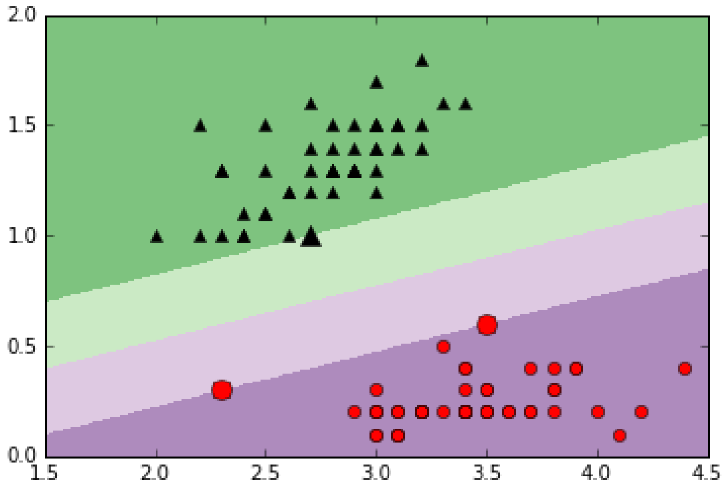
150 data points from three classes:

- 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)



Support vector machines II:
Soft-margin SVM

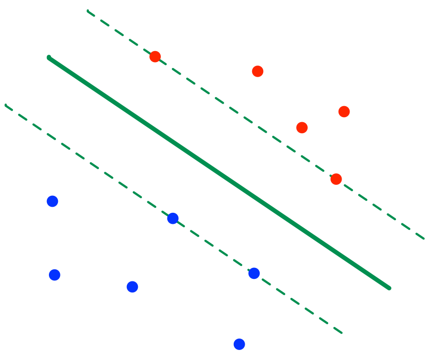
Topics we'll cover

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.

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



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?

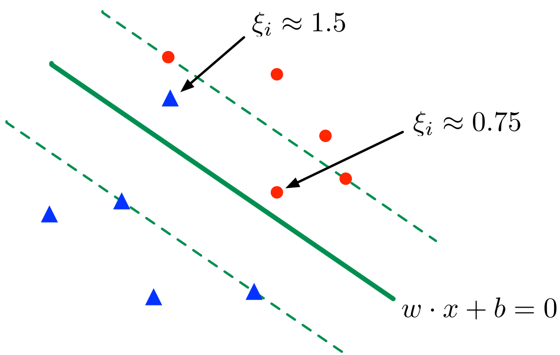
The non-separable case

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

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

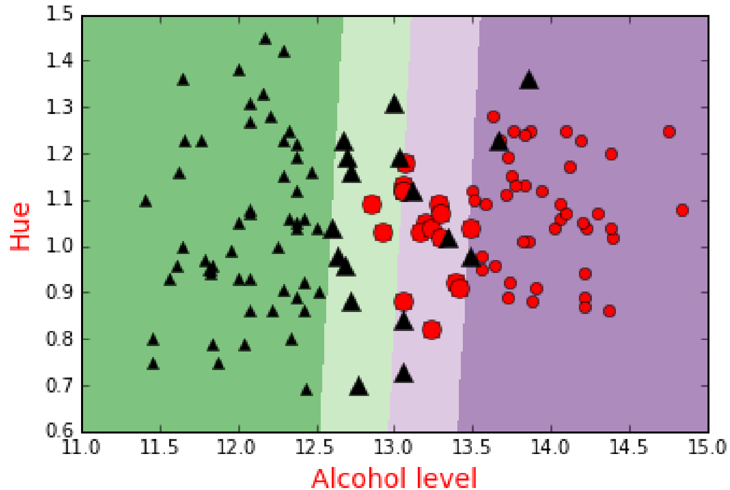
$$\text{s.t.:} \quad y^{(i)}(w \cdot x^{(i)} + b) \geq 1 - \xi_i \quad \text{for all } i = 1, 2, \ldots, n$$

$$\xi \geq 0$$



Wine data set

Here $C = 1.0$

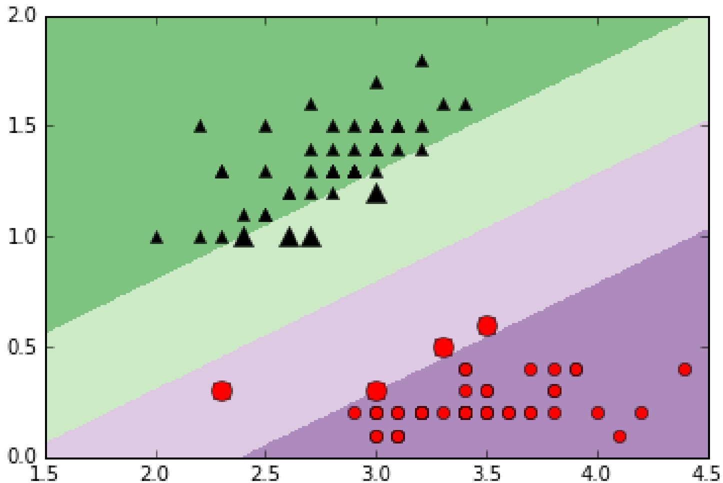


The tradeoff between margin and slack

$$\boxed{\begin{aligned} &\min_{w\in\mathbb{R}^d, b\in\mathbb{R}, \xi\in\mathbb{R}^n} \quad \|w\|^2 + C\sum_{i=1}^{n}\xi_i \\ &\text{s.t.:} \quad y^{(i)}(w\cdot x^{(i)}+b) \geq 1-\xi_i \quad \text{for all } i=1,2,\ldots,n \\ &\qquad \xi \geq 0 \end{aligned}}$$

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:

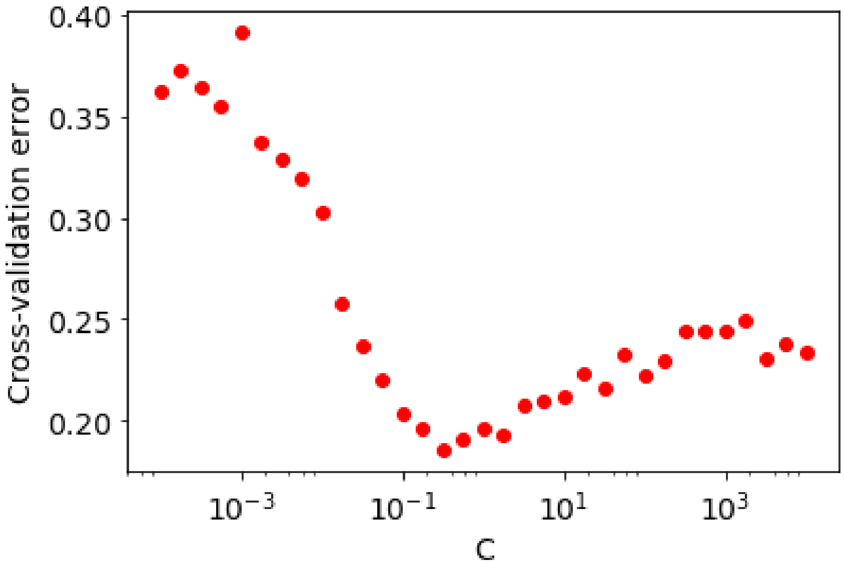
- Bag-of-words representation using a vocabulary of size 4500
- 2500 training sentences, 500 test sentences

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%

Duality

Topics we'll cover

1. Dual form of the Perceptron
2. Dual form of the support vector machine

Dual form of the Perceptron solution

Given a training set of points $\{(x^{(i)}, y^{(i)}) : i = 1 \ldots n\}$:

Perceptron algorithm

- Initialize $w = 0$ and $b = 0$
- While some training point (x, y) is misclassified:
  - $w = w + yx$
  - $b = b + y$

The final answer is of the form:

$$w = \sum_i \alpha_i y^{(i)} x^{(i)},$$

where $\alpha_i = \#$ of times an update occurred on point i.

Can equivalently represent w by $\alpha = (\alpha_1, \ldots, \alpha_n)$.

Dual form of the Perceptron algorithm

Perceptron algorithm: primal form

- Initialize $w = 0$ and $b = 0$
- While some training point $(x^{(i)}, y^{(i)})$ is misclassified:
  - $w = w + y^{(i)} x^{(i)}$
  - $b = b + y^{(i)}$

Perceptron algorithm: dual form

- Initialize $\alpha = 0$ and $b = 0$
- While some training point $(x^{(i)}, y^{(i)})$ is misclassified:
  - $\alpha_i = \alpha_i + 1$
  - $b = b + y^{(i)}$

Answer: $w = \sum_i \alpha_i y^{(i)} x^{(i)}$

Hard-margin SVM

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

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

- This is a **convex optimization problem**:
  - Convex objective function
  - Linear constraints
- As such, it has a **dual maximization problem**.
- The **primal** and **dual** problems have the same optimum value.

The dual program

$$\text{(PRIMAL)} \quad \min_{w \in \mathbb{R}^d, b \in \mathbb{R}} \quad \|w\|^2$$
$$\text{s.t.:} \quad y^{(i)}(w \cdot x^{(i)} + b) \geq 1 \quad \text{for all } i = 1, 2, \dots, n$$

$$\text{(DUAL)} \quad \max_{\alpha \in \mathbb{R}^n} \quad \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{n} \alpha_i \alpha_j y^{(i)} y^{(j)} (x^{(i)} \cdot x^{(j)})$$
$$\text{s.t.:} \quad \sum_{i=1}^{n} \alpha_i y^{(i)} = 0$$
$$\alpha \geq 0$$

Complementary slackness: At optimality, $w = \sum_{i=1}^{n} \alpha_i y^{(i)} x^{(i)}$ and

$$\alpha_i > 0 \quad \Rightarrow \quad y^{(i)}(w \cdot x^{(i)} + b) = 1$$

Points $x^{(i)}$ with $\alpha_i > 0$ are **support vectors**.

Dual of soft-margin SVM

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

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

$$\xi \geq 0$$

$$\text{(DUAL)} \qquad \max_{\alpha\in\mathbb{R}^n} \quad \sum_{i=1}^{n} \alpha_i - \frac{1}{2}\sum_{i,j=1}^{n} \alpha_i\alpha_j y^{(i)} y^{(j)} (x^{(i)} \cdot x^{(j)})$$

$$\text{s.t.:} \quad \sum_{i=1}^{n} \alpha_i y^{(i)} = 0$$

$$0 \leq \alpha_i \leq C$$

At optimality, $w = \sum_i \alpha_i y^{(i)} x^{(i)}$, with

$$0 < \alpha_i < C \quad \Rightarrow \quad y^{(i)}(w \cdot x^{(i)} + b) = 1$$

$$\alpha_i = C \quad \Rightarrow \quad y^{(i)}(w \cdot x^{(i)} + b) = 1 - \xi_i$$

Multiclass linear prediction

Topics we'll cover

1. Multiclass logistic regression
2. Multiclass Perceptron
3. Multiclass support vector machines

Multiclass classification

Of the classification methods we have studied so far, which seem inherently binary?

- Nearest neighbor?
- Generative models?
- Linear classifiers?

The main idea

Remember Gaussian generative models...

From binary to multiclass logistic regression

Binary logistic regression: for $\mathcal{X} = \mathbb{R}^d$, classifier given by $w \in \mathbb{R}^d$ and $b \in \mathbb{R}$:

$$\Pr(y = 1|x) = \frac{e^{w \cdot x + b}}{1 + e^{w \cdot x + b}}$$

Labels $\mathcal{Y} = \{1, 2, \ldots, k\}$: specify a classifier by $w_1, \ldots, w_k \in \mathbb{R}^d$ and $b_1, \ldots, b_k \in \mathbb{R}$:

$$\Pr(y = j|x) \;\propto\; e^{w_j \cdot x + b_j}$$

- What is the fully normalized form of the probability?

- Given a point x, which label to predict?

Multiclass logistic regression

- **Label space**: $\mathcal{Y} = \{1, 2, \ldots, k\}$
- **Parametrized classifier**: $w_1, \ldots, w_k \in \mathbb{R}^d$, $b_1, \ldots, b_k \in \mathbb{R}$:

$$\Pr(y = j | x) = \frac{e^{w_j \cdot x + b_j}}{e^{w_1 \cdot x + b_1} + \cdots + e^{w_k \cdot x + b_k}}$$

- **Prediction**: given a point x, predict label $\arg\max_j \, (w_j \cdot x + b_j)$.
- **Learning**: Given: $(x^{(1)}, y^{(1)}), \ldots, (x^{(n)}, y^{(n)})$.
  Find: $w_1, \ldots, w_k \in \mathbb{R}^d$ and $b_1, \ldots, b_k$ that maximize the likelihood

$$\prod_{i=1}^{n} \Pr(y^{(i)} | x^{(i)})$$

  Taking negative log gives a convex minimization problem.

Multiclass Perceptron

Setting: $\mathcal{X} = \mathbb{R}^d$ and $\mathcal{Y} = \{1, 2, \ldots, k\}$

Model: $w_1, \ldots, w_k \in \mathbb{R}^d$ and $b_1, \ldots, b_k \in \mathbb{R}$

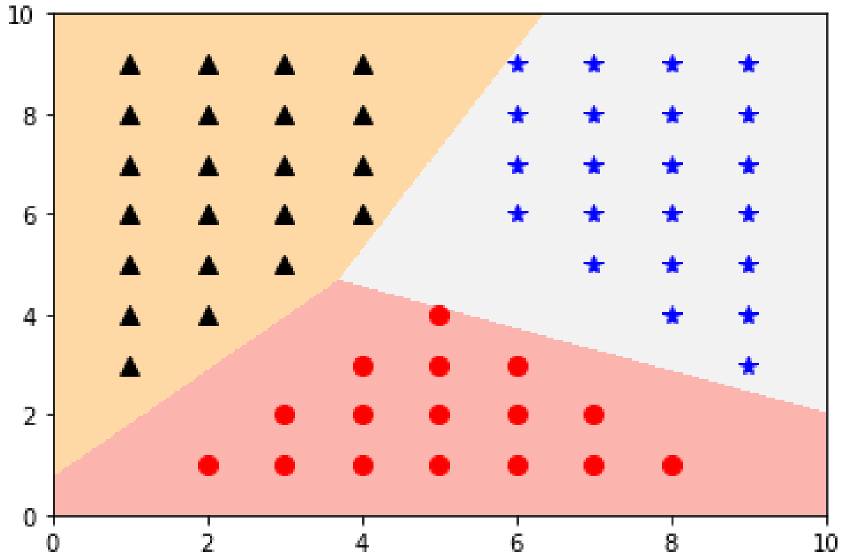
Prediction: On instance x, predict label $\arg\max_j (w_j \cdot x + b_j)$

Learning. Given training set $(x^{(1)}, y^{(1)}), \ldots, (x^{(n)}, y^{(n)})$:

- Initialize $w_1 = \cdots = w_k = 0$ and $b_1 = \cdots = b_k = 0$
- Repeat while some training point (x, y) is misclassified:

$$
\begin{aligned}
\text{for correct label } y: \quad & w_y = w_y + x \\
& b_y = b_y + 1 \\
\text{for predicted label } \widehat{y}: \quad & w_{\widehat{y}} = w_{\widehat{y}} - x \\
& b_{\widehat{y}} = b_{\widehat{y}} - 1
\end{aligned}
$$

Multiclass Perceptron: example



Multiclass SVM

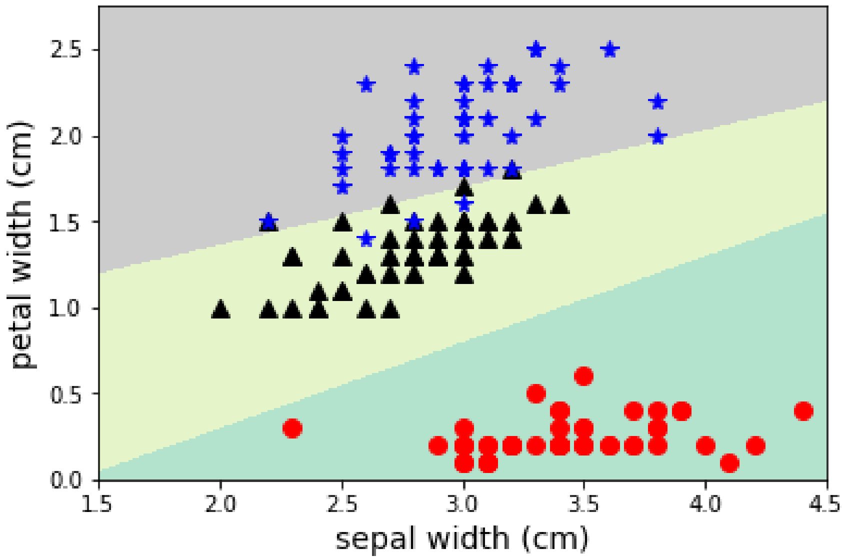
Model: $w_1, \ldots, w_k \in \mathbb{R}^d$ and $b_1, \ldots, b_k \in \mathbb{R}$

Prediction: On instance x, predict label $\arg\max_j (w_j \cdot x + b_j)$

Learning. Given training set $(x^{(1)}, y^{(1)}), \ldots, (x^{(n)}, y^{(n)})$:

$$\boxed{\begin{array}{c} \displaystyle\min_{w_1,\ldots,w_k \in \mathbb{R}^d, b_1,\ldots,b_k \in \mathbb{R}, \xi \in \mathbb{R}^n} \quad \sum_{j=1}^{k} \|w_j\|^2 + C \sum_{i=1}^{n} \xi_i \\ w_{y^{(i)}} \cdot x^{(i)} + b_{y^{(i)}} - w_y \cdot x^{(i)} - b_y \geq 1 - \xi_i \quad \text{for all } i \text{ and all } y \neq y^{(i)} \\ \xi \geq 0 \end{array}}$$

Multiclass SVM example: iris



Multiclass SVM

Given training set $(x^{(1)}, y^{(1)}), \ldots, (x^{(n)}, y^{(n)})$:

$$\boxed{\begin{array}{c} \min_{w_1,\ldots,w_k \in \mathbb{R}^d, b_1,\ldots,b_k \in \mathbb{R}, \xi \in \mathbb{R}^n} \quad \sum_{j=1}^{k} \|w_j\|^2 + C \sum_{i=1}^{n} \xi_i \\ w_{y^{(i)}} \cdot x^{(i)} + b_{y^{(i)}} - w_y \cdot x^{(i)} - b_y \geq 1 - \xi_i \quad \text{for all } i \text{ and all } y \neq y^{(i)} \\ \xi \geq 0 \end{array}}$$

Once again, a convex optimization problem.

Question: how many variables and constraints do we have?