Linear classifiers

Gosia Migut

Admin stuff

- Answers (not solutions) to the labs 2 are on Brightspace.
- I like your tips at the end of each lecture. Bring them on!
- Next week exercises to practice for the exam.

Learning goals

- Explain logistic regression classifier, including cost function and it’s optimization
- Explain the following concept of support vector classifier: margin, support vectors, hinge loss
- Explain approaches to multi-class classification and their problems

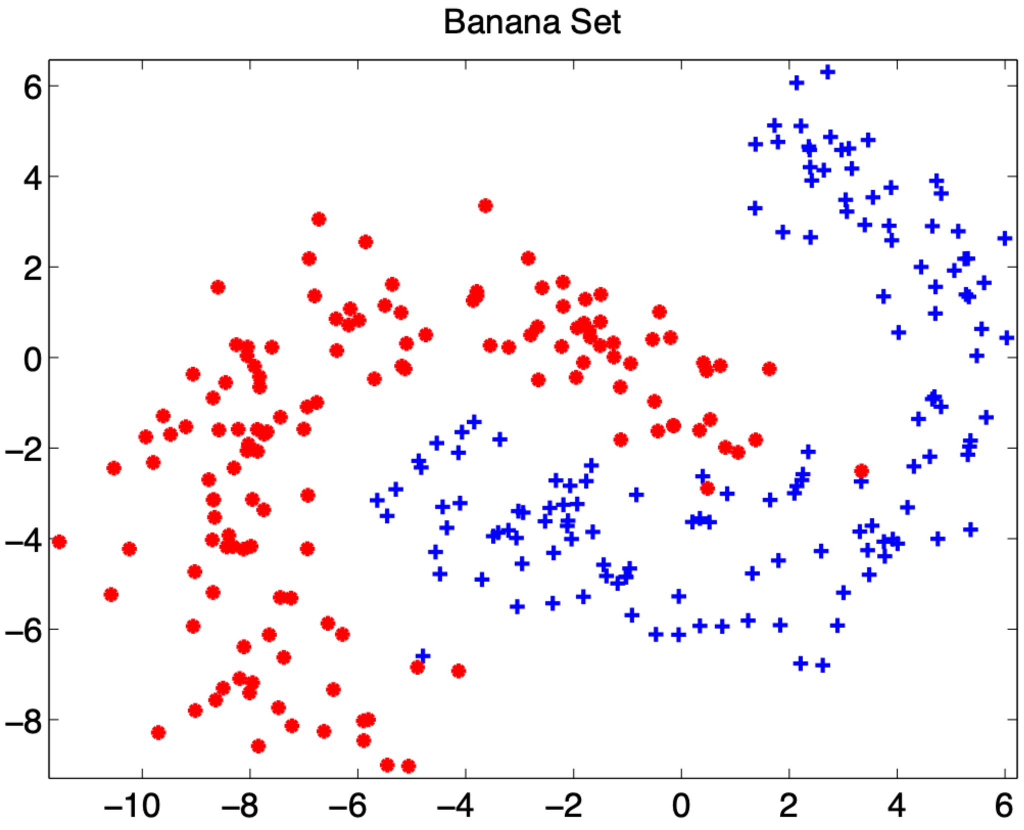
Reading

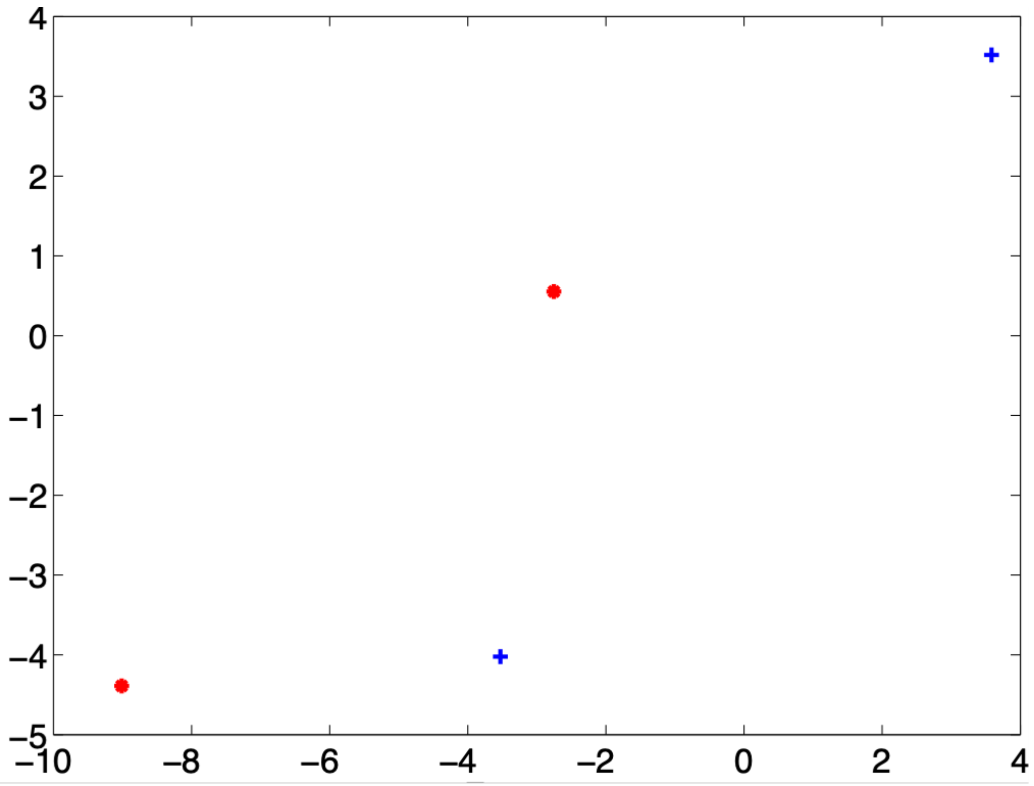
- Logistic regression: CS229 Lecture Notes by Andrew Ng
  http://cs229.stanford.edu/notes/cs229-notes1.pdf
- SVM: CS229 Lecture Notes by Andrew Ng
  http://cs229.stanford.edu/notes/cs229-notes3.pdf
- Multi-class classification: Bishop “Pattern recognition, section 4.1.2 (p.182-184)

Recap last lecture

- Discriminative models
- Linear classifier
- Cost function
- Gradient descent

Generative vs discriminative models

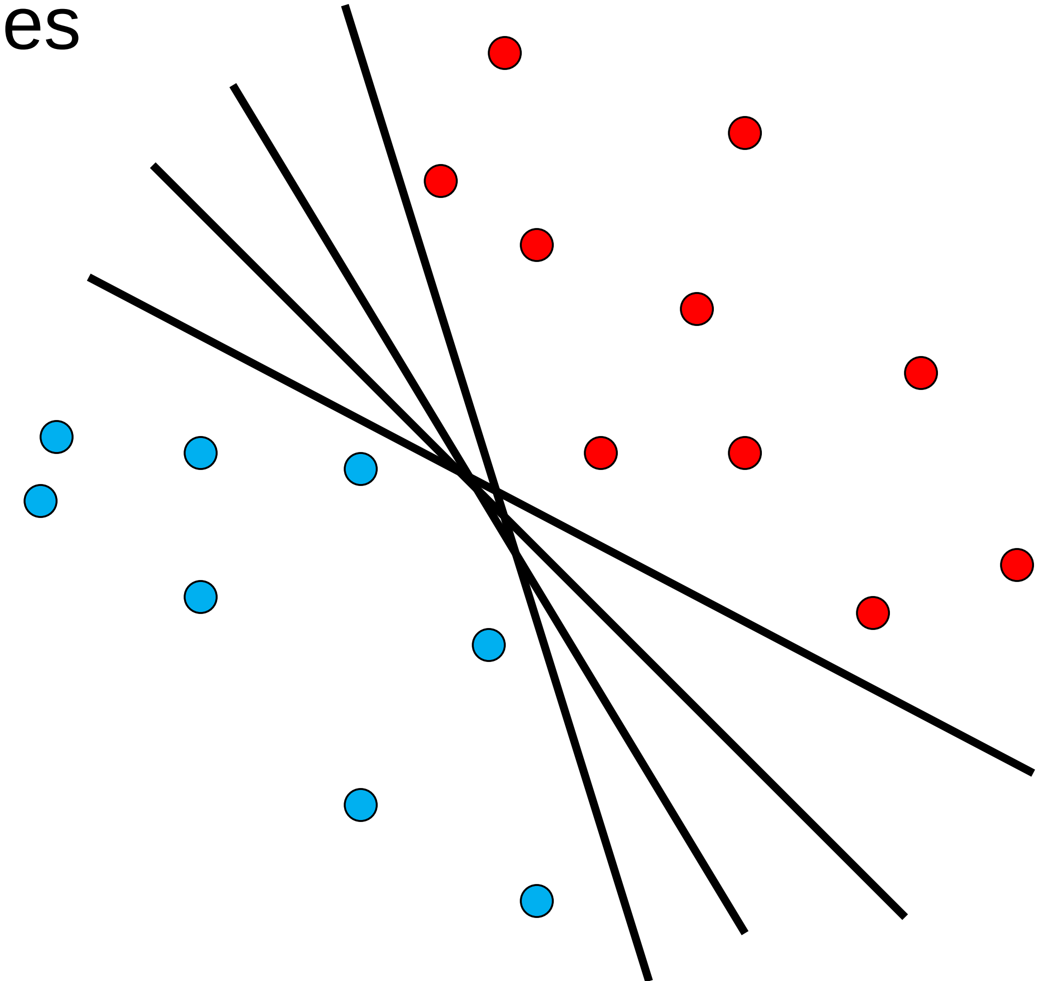




- Models the probability distribution of each class
- Models decision boundary between classes

Linear classifier

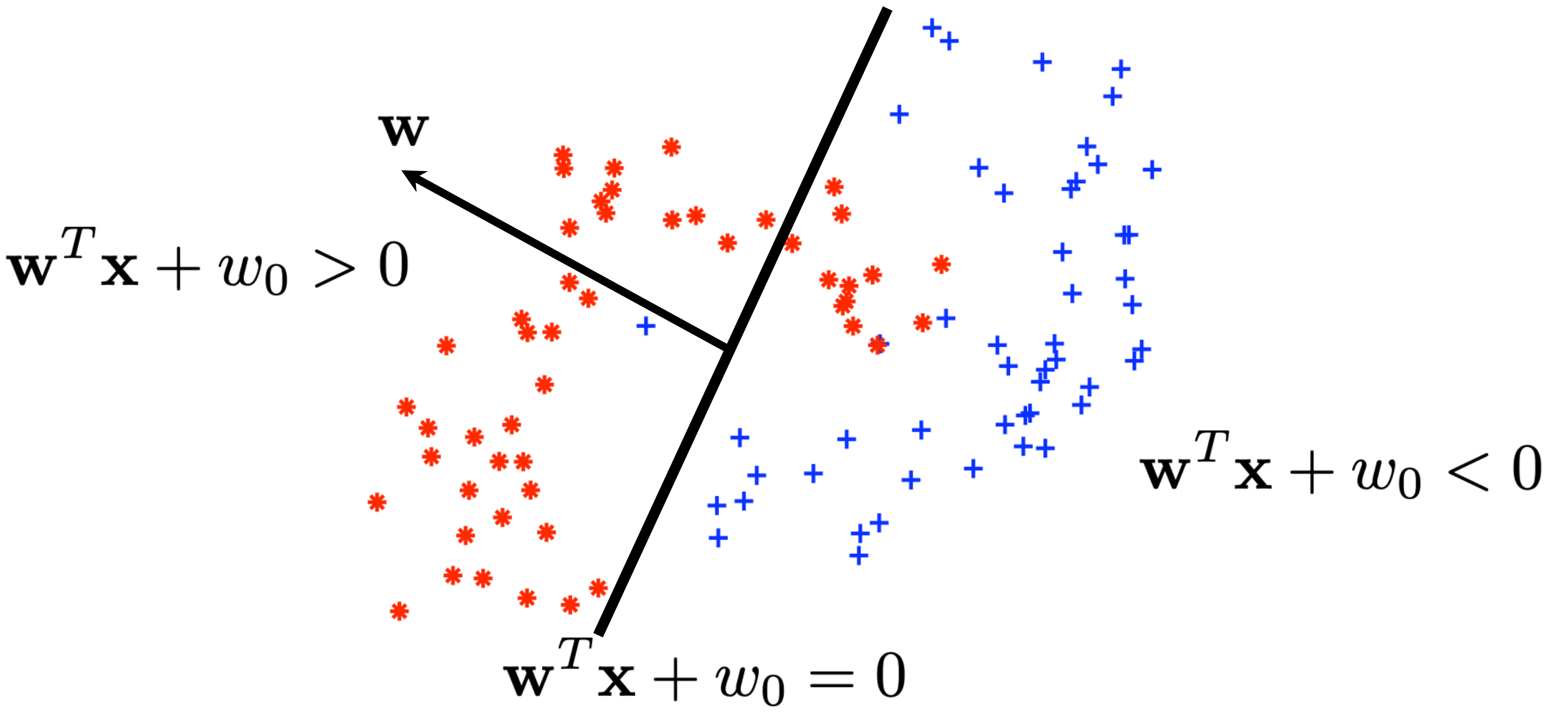
- Find linear function (*hyperplane*) to separate positive and negative examples



Which hyperplane is best?

Linear classifier

- $h(x) = w^T x + w_0$



- How to choose w ?

Cost/Loss function

- General idea:

$$J(w) = \sum_{i=1}^{n} cost(h(x_i), y_i)$$

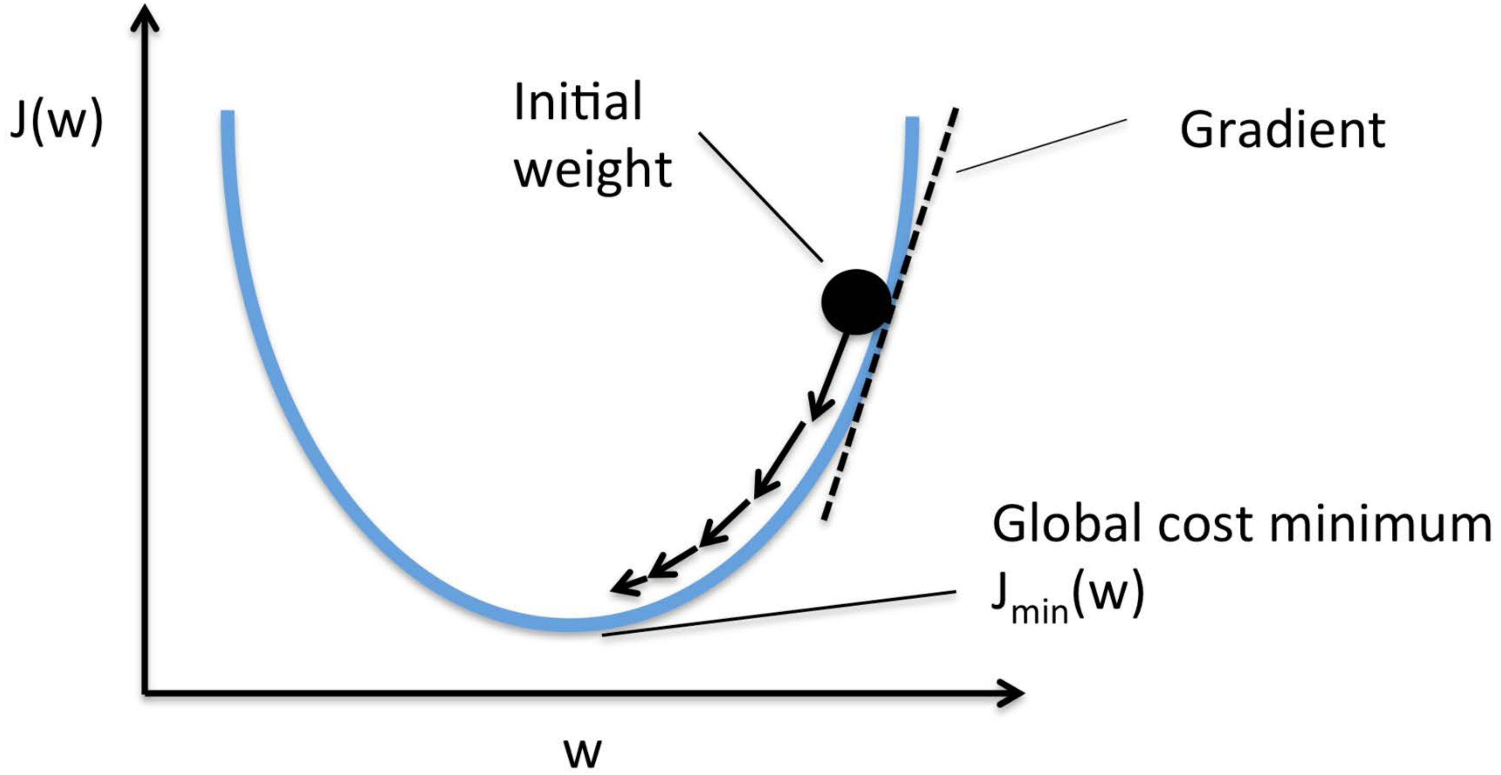
- Examples: least squares, logistic loss, hinge loss, perceptron loss etc.

- Goal: optimize cost function
  - Analytical solution $\frac{\partial J(w)}{\partial w} = 0$, if possible
  - Gradient descent

TUDelft

Gradient descent

- $w_j := w_j - \alpha \dfrac{\partial J(w)}{\partial w_j}$



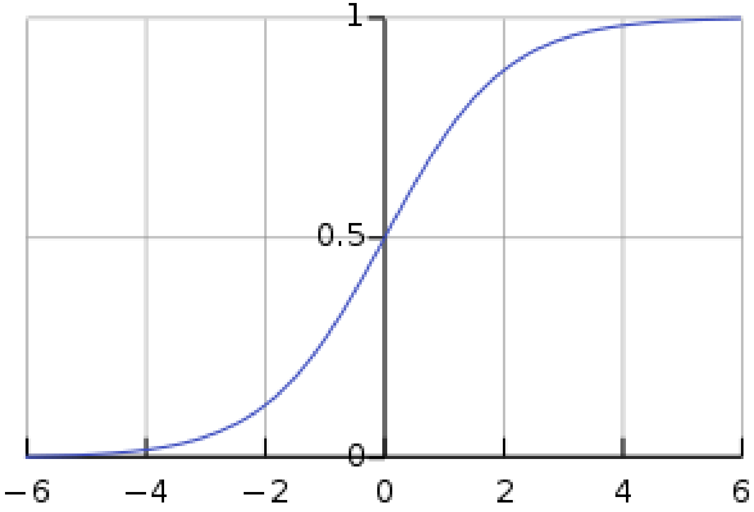
Logistic regression

Logistic regression

- Let's change the form of linear hypotheses

  $h(x) = w^T x$ to satisfy $0 \leq h(x) \leq 1$

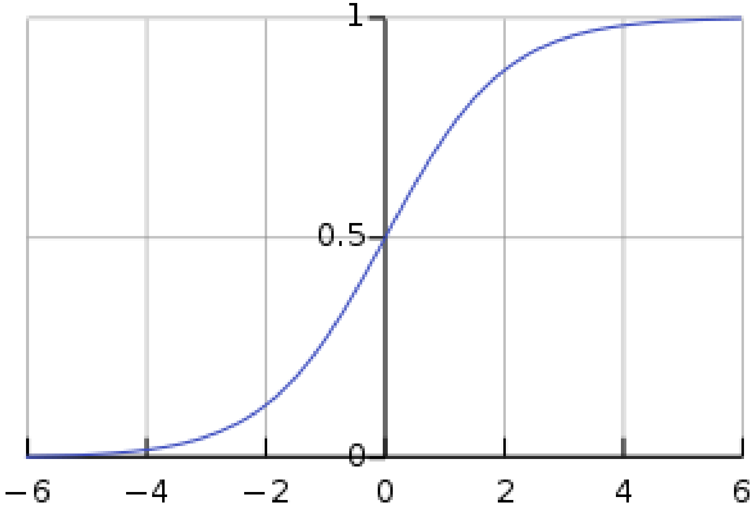
  $$g(z) = \frac{1}{1+e^{-z}}$$



- Let's plug $w^T x$ into the logistic function
- $z = w^T x$
- $h(x) = g(w^T x)$

TUDelft

Logistic function



- $\mathrm{h}(x) = \frac{1}{1+e^{(-w^T x)}}$

- $0 \leq h(x) \leq 1$

- $\mathrm{h}(x)$ gives us the probability that our output is 1

How to choose parameters w ?

- Training set: $\left\{\left(x^{(1)}, y^{(1)}\right), \left(x^{(2)}, y^{(2)}\right) \dots \left(x^{(n)}, y^{(n)}\right)\right\}$
- D features: $\begin{bmatrix} x_0 \\ x_1 \\ \dots \\ x_d \end{bmatrix}$, $x_0 = 1$
- $y \in \{0, 1\}$
- $h(x) = \frac{1}{1+e^{-w^T x}}$

- Define cost function and optimize!

Logistic regression cost function

- We defined that: $p(y_1|x) = h_w(x)$
- For a 2 class problem: $p(y_0|x) = 1 - h_w(x)$
- We can rewrite:
- $p(y|x) = \begin{cases} h_w(x) & : y = 1 \\ 1 - h_w(x) & : y = 0 \end{cases}$

- This is discrete probability distribution Bernoulli which takes the value 1 with probability p and the value 0 with probability 1-p

Logistic regression cost function

- $p(y|x) = \begin{cases} h_w(x) & : y = 1 \\ 1 - h_w(x) & : y = 0 \end{cases}$

- We can interpret it as:
  - Given x, class y=1 occurs with probability $h_w(x)^y$
  - Given x, class y=0 occurs with probability $1 - h_w(x)^{1-y}$

- Therefore: $p(y|x) = h_w(x)^y (1 - h_w(x))^{1-y}$

$$p(y|x) = h_w(x)^y (1 - h_w(x))^{1-y}$$

Logistic regression cost function

- For the entire dataset (assuming samples were drawn independently):

$$p(y|x) = \prod_{i=1}^{n} p(y^{(i)}|x^{(i)}) = \prod_{i=1}^{n} h_w\left(x^{(i)}\right)^{y^{(i)}} (1 - h_w\left(x^{(i)}\right))^{1-y^{(i)}}$$

- We can interpret this as the likelihood of the data given the parameter $w \rightarrow l(w)$

- Maximum likelihood estimator: $\widehat{w} = \mathrm{argmax}_w \log(l(w))$
- Or: $\widehat{w} = \underset{w}{\mathrm{argmin}}(-\log\left(l(w)\right))$

TUDelft

Logistic regression cost function

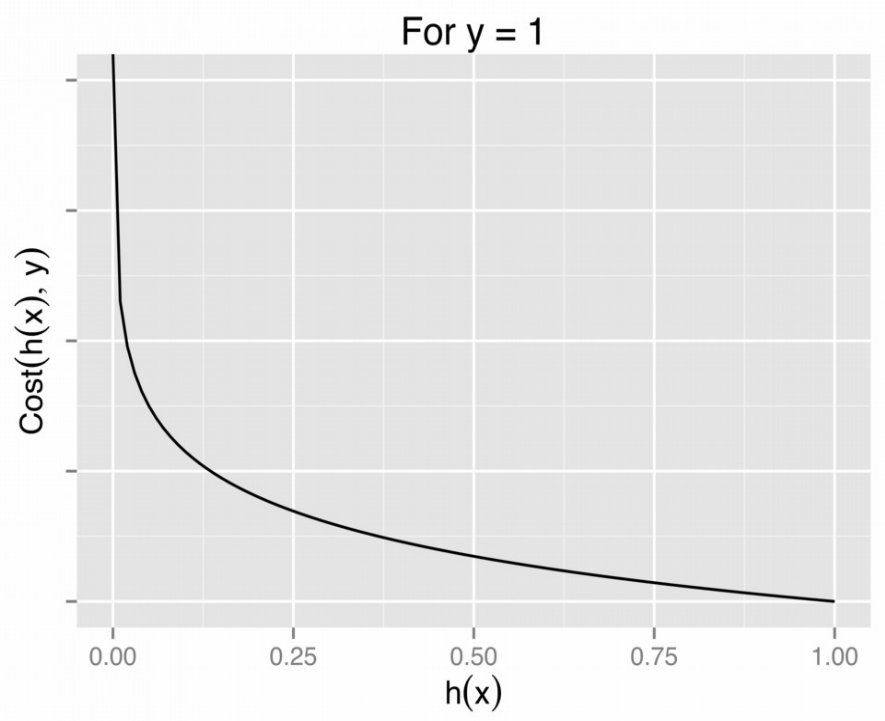
- $J(w) = -\log\left(l(w)\right)$
- $l(w) = \prod_{i=1}^{n} h_w\left(x^{(i)}\right)^{y^{(i)}} (1 - h_w\left(x^{(i)}\right))^{1-y^{(i)}}$
- $J(w) = -\log\left(\prod_{i=1}^{n} h_w\left(x^{(i)}\right)^{y^{(i)}} (1 - h_w\left(x^{(i)}\right))^{1-y^{(i)}}\right) =$
- $\sum_{i=1}^{n} -log\left(h_w\left(x^{(i)}\right)^{y^{(i)}}\right) - \log\left((1 - h_w\left(x^{(i)}\right))^{1-y^{(i)}}\right) =$
- $\sum_{i=1}^{n} -y^{(i)} log\left(h_w\left(x^{(i)}\right)\right) - (1 - y^{(i)})\log\left(1 - h_w\left(x^{(i)}\right)\right)$

Cost function

$$J(w) = \sum_{i=1}^{n} -y^{(i)} log\left(h_w\left(x^{(i)}\right)\right) - \left(1 - y^{(i)}\right)\log\left(1 - h_w\left(x^{(i)}\right)\right)$$

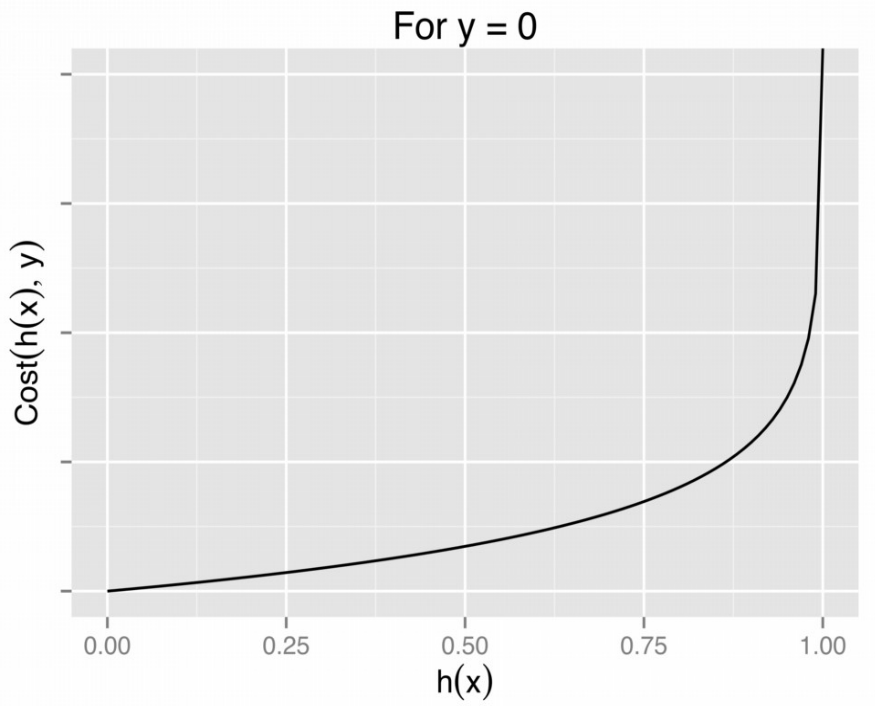
- $Cost(h(x), y) = \begin{cases} -\log\left(h_w\left(x^{(i)}\right)\right) & if\ y = 1 \\ -\log\left(1 - h_w\left(x^{(i)}\right)\right) & if\ y = 0 \end{cases}$
- If y = 1 and h(x) = 1, Cost = 0
- If $h_w(x) \rightarrow 0,\ Cost\ \rightarrow\ \infty$
- Captures intuition:

  if prediction is h(x) = 0, but y = 1, learning algorithm will be penalized by large cost



Cost function

- $Cost(h(x), y) = \begin{cases} -\log\left(h_w\left(x^{(i)}\right)\right) & if\ y = 1 \\ -\log\left(1 - h_w\left(x^{(i)}\right)\right) & if\ y = 0 \end{cases}$
- If y = 0 and h(x) = 0, Cost = 0
- If $h_w(x) \rightarrow 1$ $Cost \rightarrow \infty$
- Captures intuition:
  if prediction is h(x) = 1, but y = 0,
  learning algorithm will be
  penalized by large cost



TU Delft

How to minimize the -$\log(l(w))$?

- No analytical solution for logistic regression.
- Do gradient descent:
- Repeat {

$$w_j := w_j - \alpha \frac{\partial J(w)}{\partial w_j}$$

}

- $\frac{\partial J(w)}{\partial w} = \sum_{i=1}^{n} (y^{(i)} - h(x^{(i)}))x^{(i)}$
- Where $\mathrm{h}(x) = \frac{1}{1+e^{(-w^T x)}}$

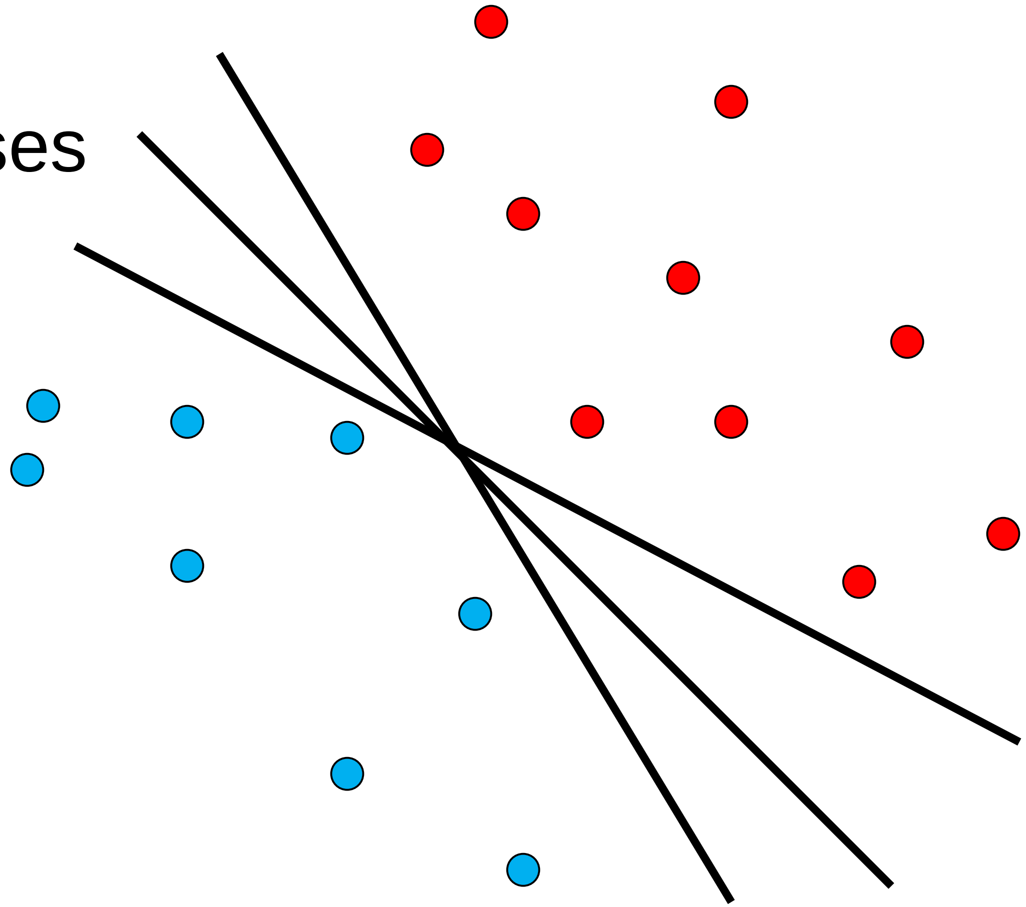
TU Delft

Logistic regression summary

- Linear classifier
- Models decision boundary

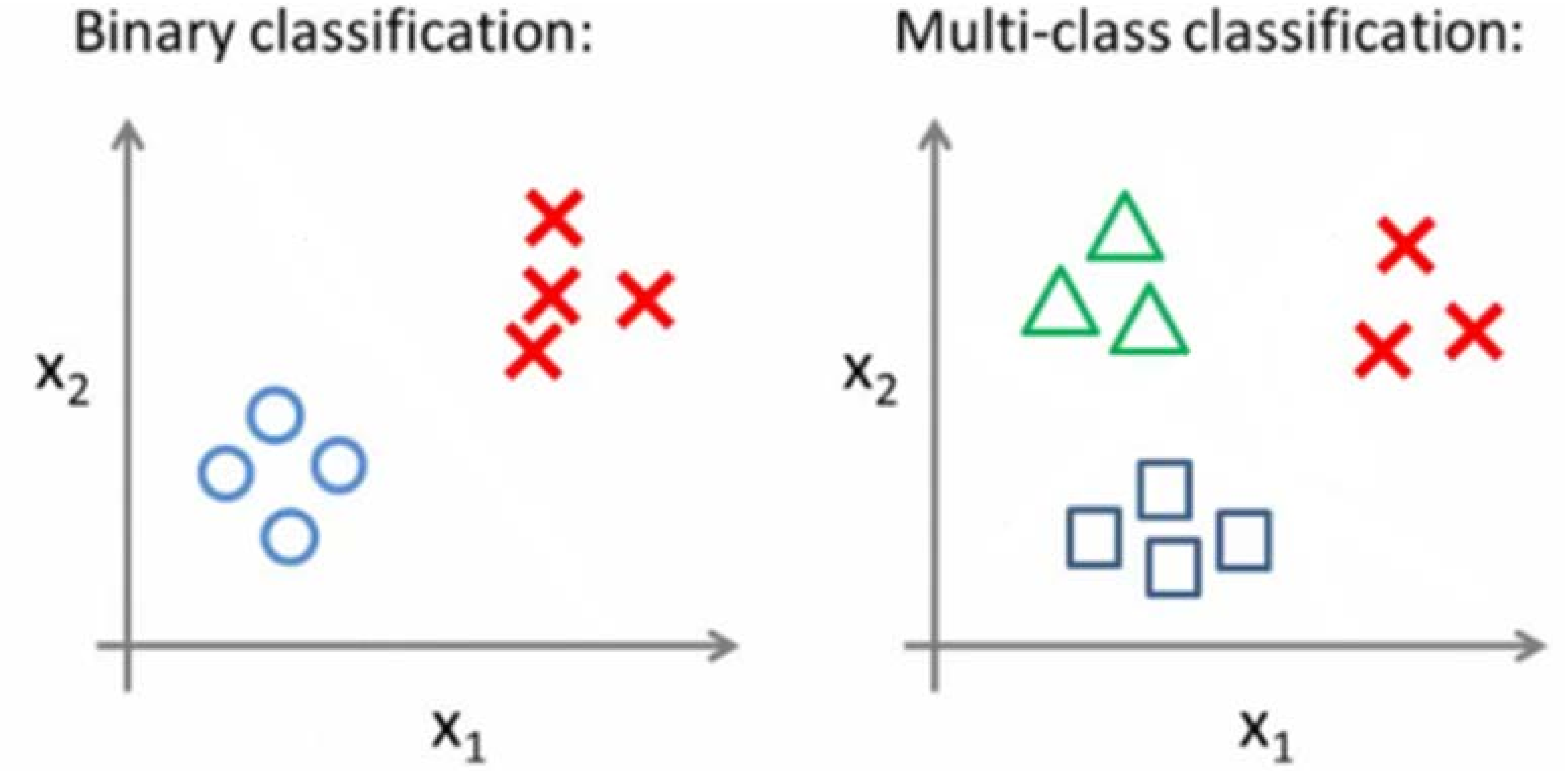
  by modelling probability of the classes

  by minimizing the logistic loss
- $\mathrm{h}(x) = \frac{1}{1+e^{(-w^T x)}}$



Multi-class classification

Multi-class

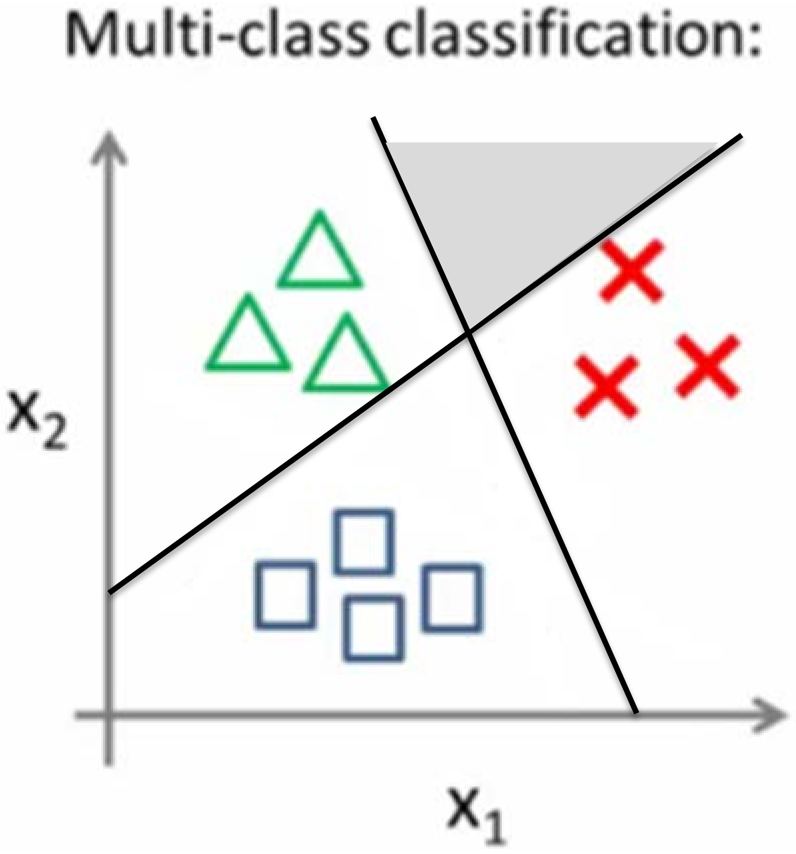

Binary classification:
Multi-class classification:
x_2
x_1
x_2
x_1

One-versus-the-rest (one-versus-all)

- Use K-1 binary classifiers
- Separate one class from the rest

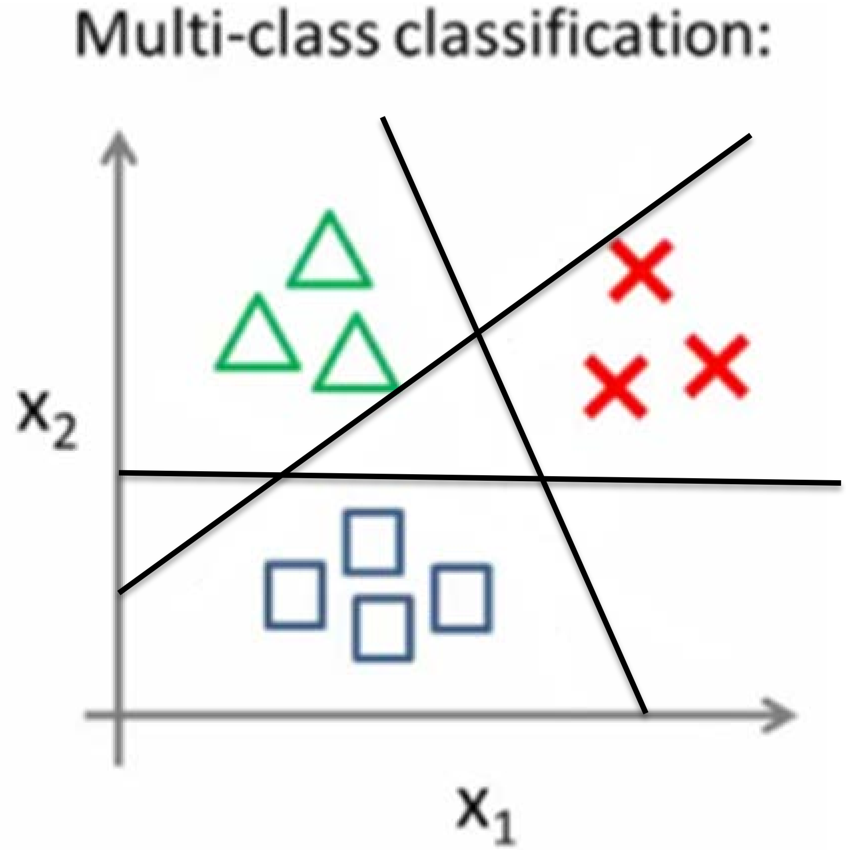
- Problem?

- Ambigiously classified regions



One-versus-one

- Use K(K-1)/2 binary classifiers
- One for each pair of classes
- Take majority vote among classifiers

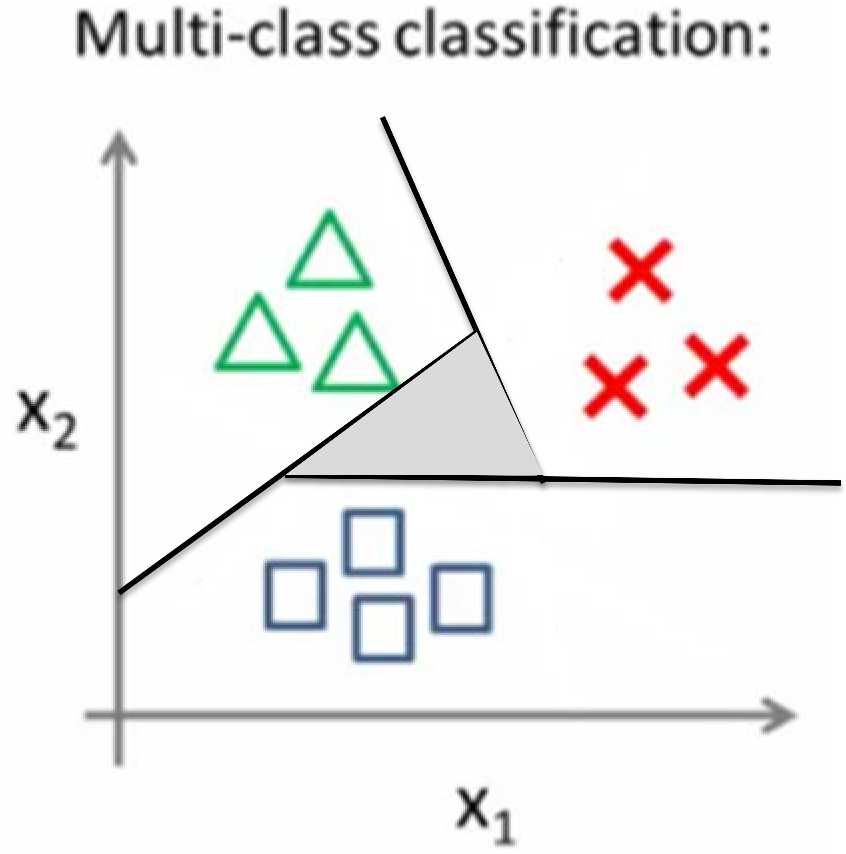


One-versus-one

- Use K(K-1)/2 binary classifiers
- One for each pair of classes
- Take majority vote among classifiers

- Problem?

- Ambigiously classified regions

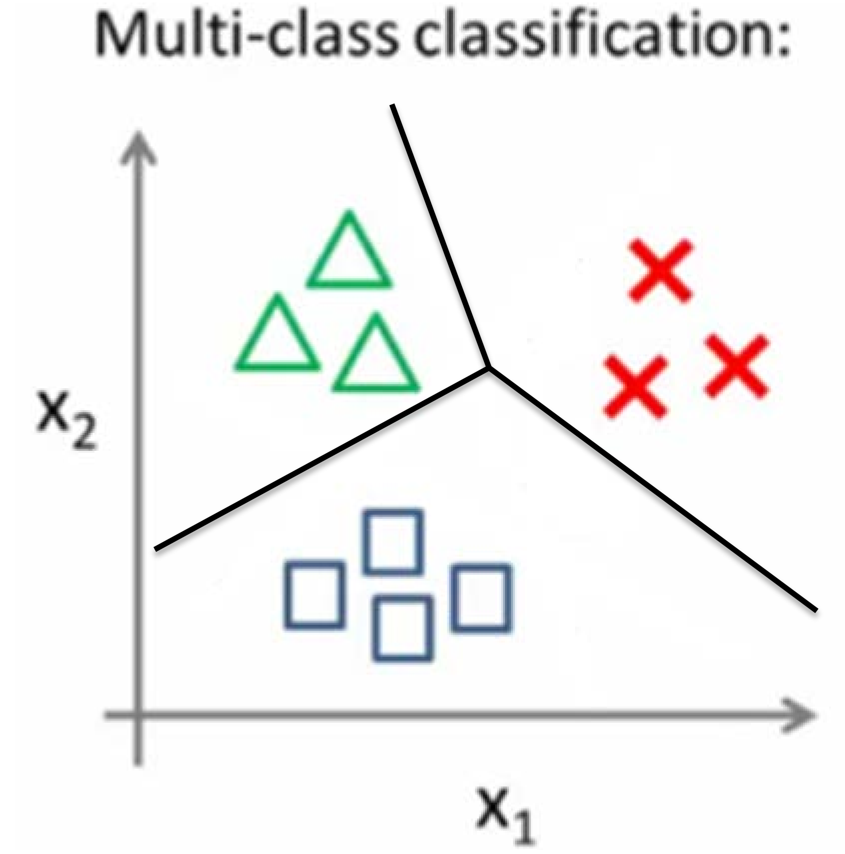


Single k-class discriminant

- Comprises of K functions
  - $h_k(x) = w_k^T x + w_{k0}$

- Assign point x to class C_k
  if $h_k(x) > h_j(x)$

- The decision boundary between class C_j and C_k is given by $y_j(x) = y_k(x)$ and defined as:
  $(w_k - w_j)^T x + (w_{k0} - w_{j0}) = 0$



TUDelft

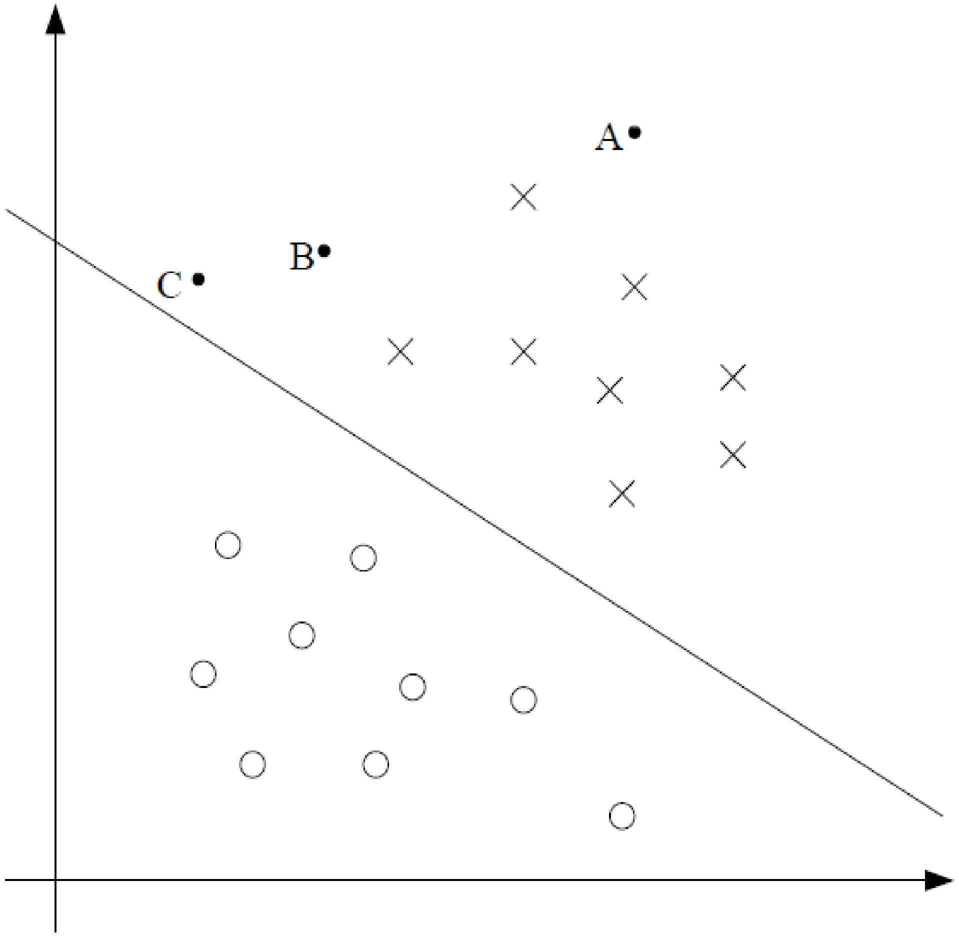
Support vector machine

SVM classifier

- SVM is much more then I will tell you today

- Intuition about
  - the cost function
  - the margin
  - the support vectors

Support vector machine intuition

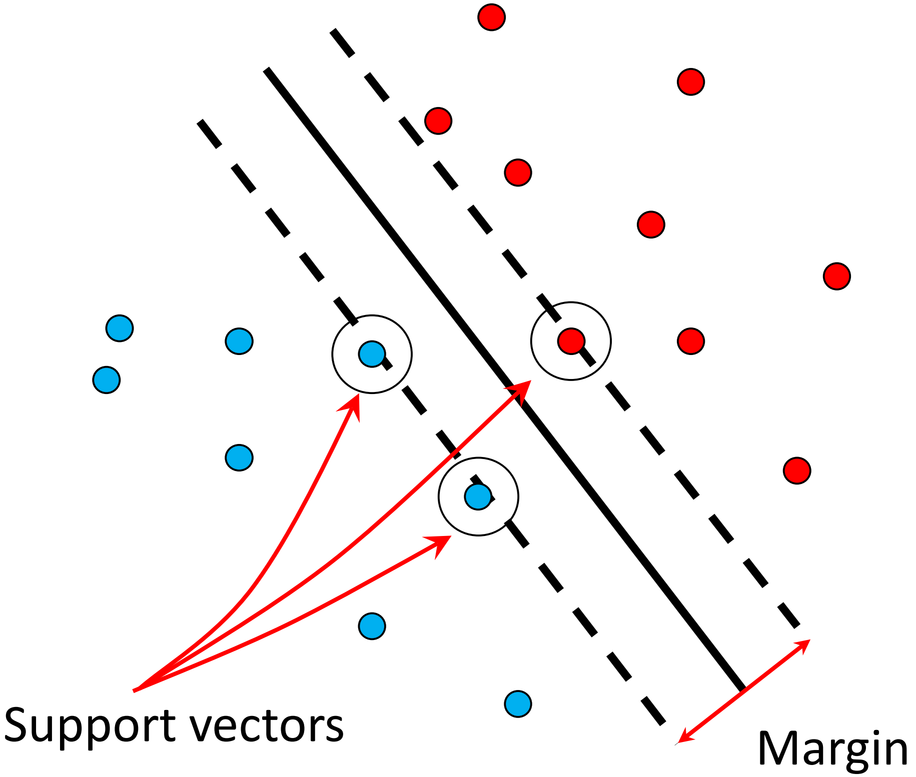


Support vector machines

- Find hyperplane that maximizes the *margin* between the positive and negative examples

Support vector machines

- Find hyperplane that maximizes the *margin* between the positive and negative examples



x_i positive ($y_i = 1$): $\quad w^T x_i + b \geq 1$
x_i negative ($y_i = -1$): $\quad w^T x_i + b \leq -1$

For support vectors, $w^T x_i + b = \pm 1$

Distance between point and hyperplane: $\frac{w^T x_i + b}{\|w\|}$

The margin is $\frac{2}{\|w\|}$

Find the maximum margin hyperplane

- Correctly classify all training data:

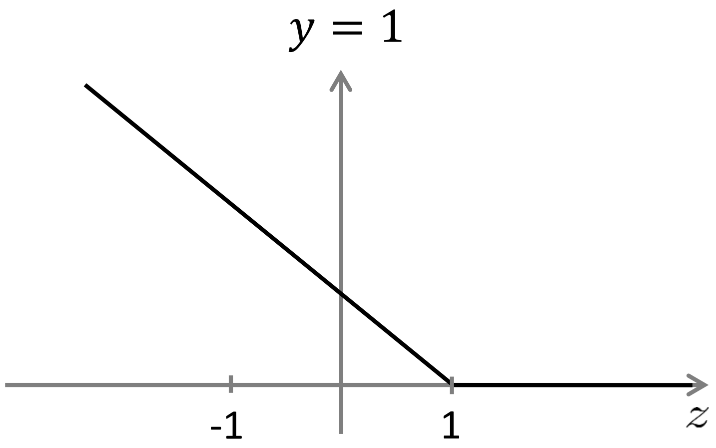
x_i positive ($y_i = 1$): $\quad w^T x_i + b \geq 1$
x_i negative ($y_i = -1$): $\quad w^T x_i + b \leq -1$

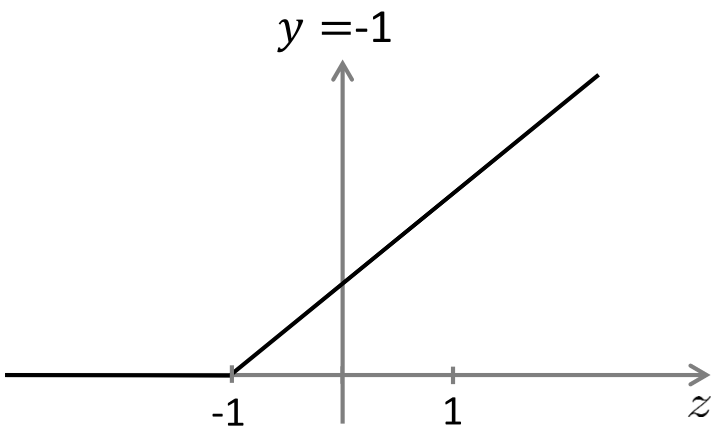
- Maximize margin $\frac{2}{\|w\|}$

- $\mathrm{J}(W) = \frac{1}{2}\|w\|^2$
- $\min \mathrm{J}(W)$

Find the maximum margin hyperplane: Hinge loss

- If y =1, we want $w^T x \geq 1$ (not just $w^T x \geq 0$)
- If $y = 0$, we want $w^T x \leq -1$ (not just $w^T x < 0$)





- $J(w) = \left[\frac{1}{n}\sum_{i=1}^{n} max(0, 1 - y_i(w^T x_i - b))\right]$

Svm summary

- Find hyperplane that maximizes the *margin* between the positive and negative examples
- Maximize the margin and correctly classify all examples
- Use hinge loss to penalize for errors

Summary

- Discriminative linear classifiers
  - Linear decision boundary
  - Models decision boundary
  - Through minimizing the loss/cost function, eg.
    - Logistic loss
    - Hinge loss