Aalto University
School of Electrical
Engineering

CS-C3240 – Machine Learning D

Classification

Dariush Salami

Department of Communications and Networking
Aalto University, School of Electrical Engineering
dariush.salami@aalto.fi

Version 1.0, September 14, 2022

Learning goals

- Logistic Regression
  - Logistic Loss
- Support Vector Machines
  - Hinge loss
  - Maximum margin principle
- The perceptron algorithm
- Multi-class and multi-label problems

Outline

Recap: linear regression

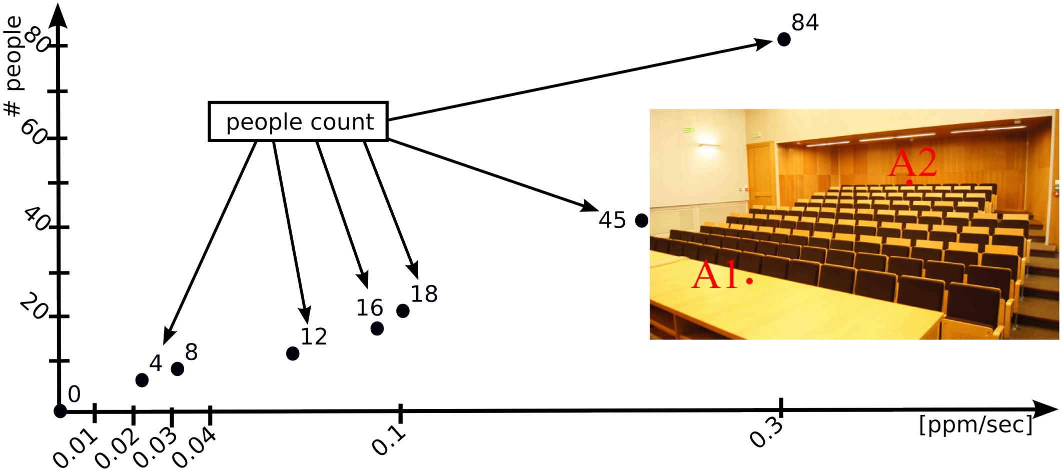
Logistic regression

The Perceptron algorithm

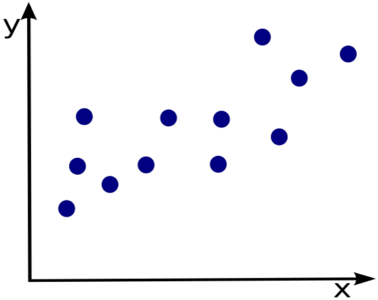
Support Vector Machines

Multiclass classification

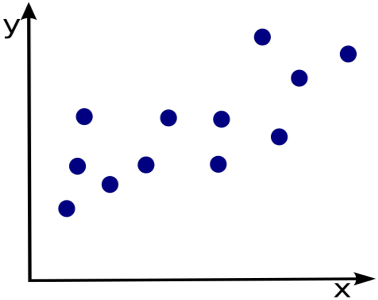
Recap: linear regression



Recap: linear regression

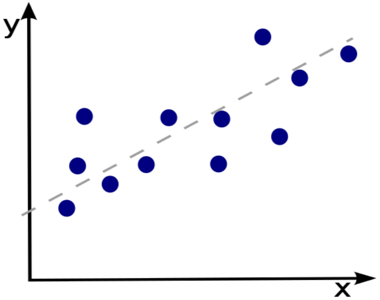


Recap: linear regression



What do we try to find with linear regression?

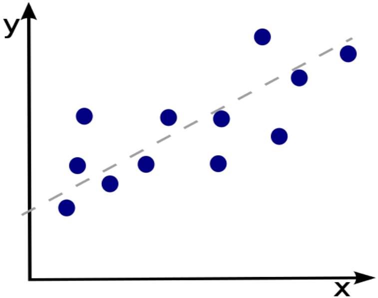
Recap: linear regression



Hypothesis: $h(x) = w_0 + w_1 x$

What do we try to find with linear regression?

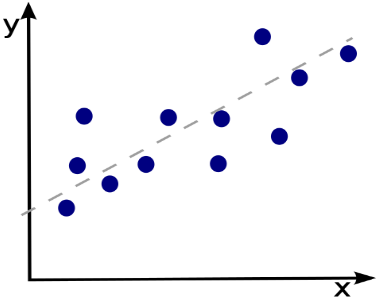
Recap: linear regression



Hypothesis: $h(x) = w_0 + w_1 x$

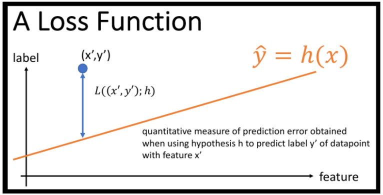
- What do we try to find with linear regression?
- How do we find proper parameters w_0 and w_1 ?

Recap: linear regression

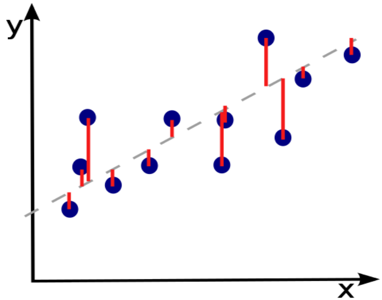


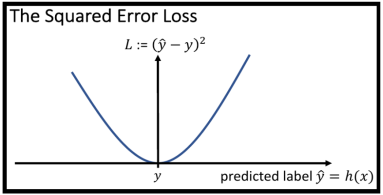
Hypothesis: $h(x) = w_0 + w_1 x$

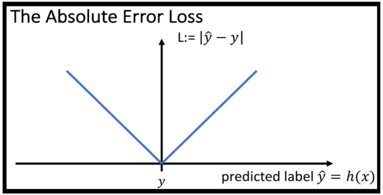
- What do we try to find with linear regression?
- How do we find proper parameters w_0 and w_1 ?



Recap: linear regression

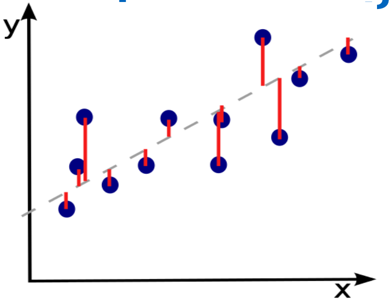






Hypothesis: $h(x) = w_0 + w_1 x$

Recap: linear regression



Loss function:
estimates quality of current solution;

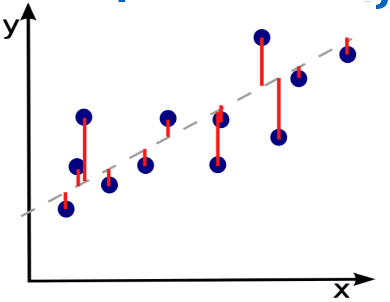
sometimes called *error function* or *cost function*.

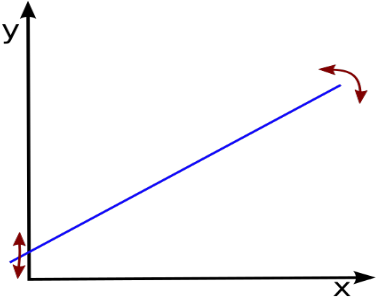
Hypothesis: $h(x) = w_0 + w_1 x$

$$\text{minimize } L[(\mathcal{X},\mathcal{Y}),h(\cdot)] = \frac{1}{2n}\sum_{i=1}^{n}(h(x_i)-y_i)^2$$

$$\text{weight update (step } t \rightarrow t+1\text{): } w_1^{t+1} = w_1^t - \delta \cdot \frac{\partial L[(\mathcal{X},\mathcal{Y}),h(\cdot)]}{\partial w_1}$$

Recap: linear regression





<u>Loss function</u>:
estimates quality of current solution;

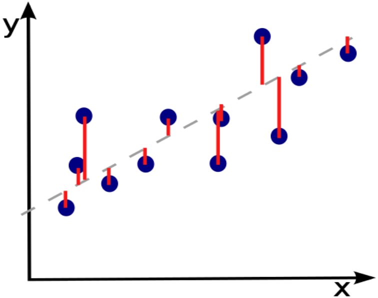
sometimes called *error function* or *cost function*.

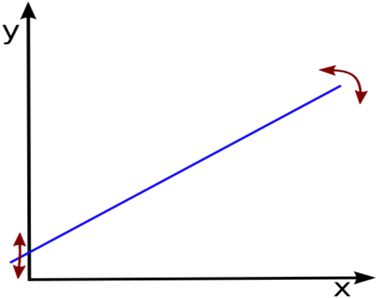
Hypothesis: $h(x) = w_0 + w_1 x$

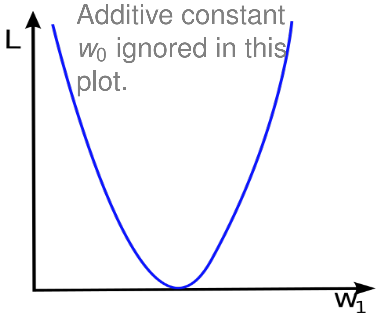
$$\text{minimize } L[(\mathcal{X}, \mathcal{Y}), h(\cdot)] = \frac{1}{2n} \sum_{i=1}^{n} (h(x_i) - y_i)^2$$

$$\text{weight update (step } t \to t+1\text{): } w_1^{t+1} = w_1^t - \delta \cdot \frac{\partial L[(\mathcal{X}, \mathcal{Y}), h(\cdot)]}{\partial w_1}$$

Recap: linear regression





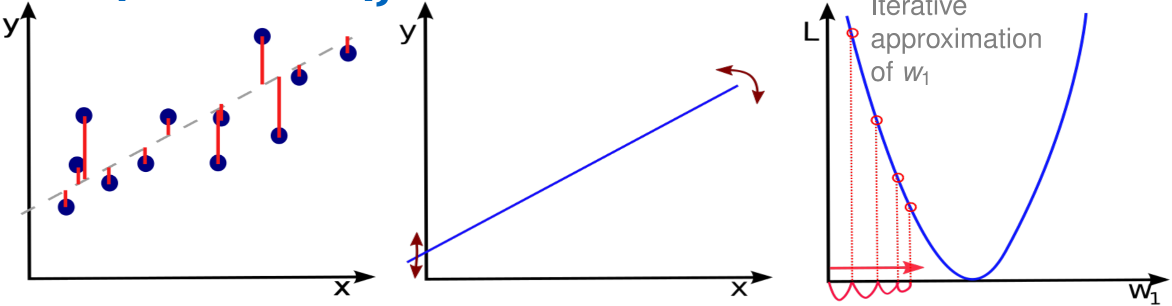


Hypothesis: $h(x) = w_0 + w_1 x$

$$\text{minimize } L[(\mathcal{X}, \mathcal{Y}), h(\cdot)] = \frac{1}{2n} \sum_{i=1}^{n} (h(x_i) - y_i)^2$$

$$\text{weight update (step } t \rightarrow t+1\text{): } w_1^{t+1} = w_1^t - \delta \cdot \frac{\partial L[(\mathcal{X}, \mathcal{Y}), h(\cdot)]}{\partial w_1}$$

Recap: linear regression



Hypothesis: $h(x) = w_0 + w_1 x$

$$\text{minimize } L[(\mathcal{X}, \mathcal{Y}), h(\cdot)] = \frac{1}{2n} \sum_{i=1}^{n} (h(x_i) - y_i)^2$$

$$\text{weight update (step } t \rightarrow t+1\text{): } w_1^{t+1} = w_1^t - \delta \cdot \frac{\partial L[(\mathcal{X}, \mathcal{Y}), h(\cdot)]}{\partial w_1}$$

Outline

Recap: linear regression

Logistic regression

The Perceptron algorithm

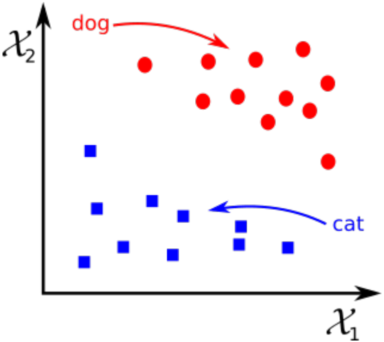
Support Vector Machines

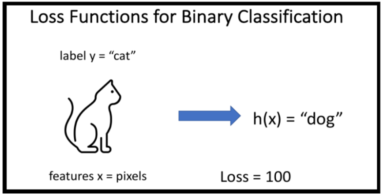
Multiclass classification

Logistic regression

Nominal classes

Classes might be nominal in real-world problems

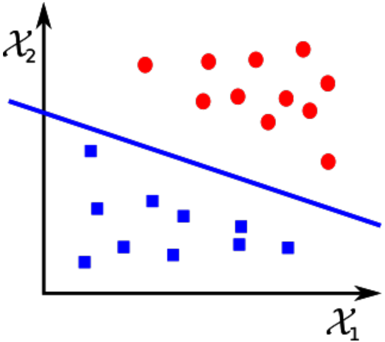


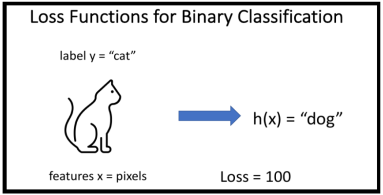


Logistic regression

Nominal classes

Classes might be nominal in real-world problems



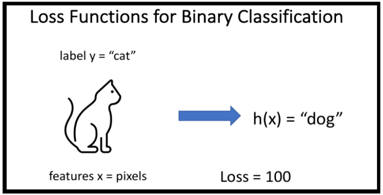


Logistic regression

Nominal classes

Classes might be nominal in real-world problems

Weather	Sunny, rainy
Medical	positive diagnosis, negative diagnosis
Localisation	indoor, outdoor



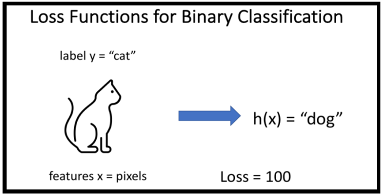
Logistic regression

Nominal classes

Classes might be nominal in real-world problems

- **Weather** Sunny, rainy
- **Medical** positive diagnosis, negative diagnosis
- **Localisation** indoor, outdoor

In such case, classification is binary: $y \in \{0, 1\}$



Logistic regression

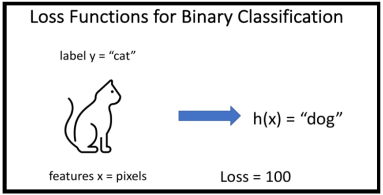
Nominal classes

Classes might be nominal in real-world problems

- **Weather** Sunny, rainy
- **Medical** positive diagnosis, negative diagnosis
- **Localisation** indoor, outdoor

In such case, classification is binary: $y \in \{0, 1\}$

Linear regression: $h(x)$ can be smaller than 0 or greater than 1



Logistic regression

Nominal classes

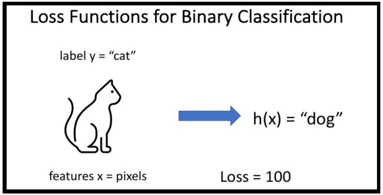
Classes might be nominal in real-world problems

Weather	Sunny, rainy
Medical	positive diagnosis, negative diagnosis
Localisation	indoor, outdoor

In such case, classification is binary: $y \in \{0, 1\}$

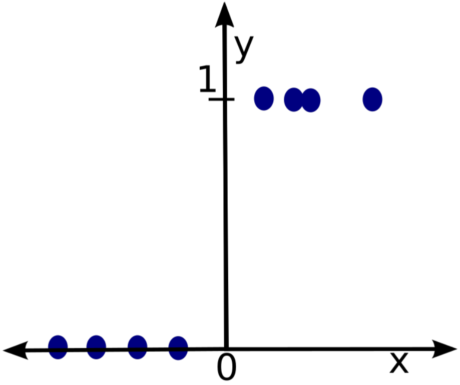
Linear regression: $h(x)$ can be smaller than 0 or greater than 1

Logistic regression: $0 \leq h(x) \leq 1$



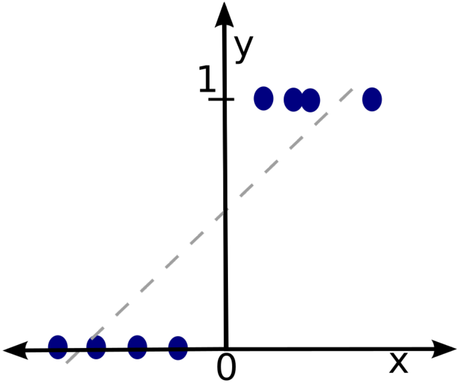
Logistic regression

Nominal classes



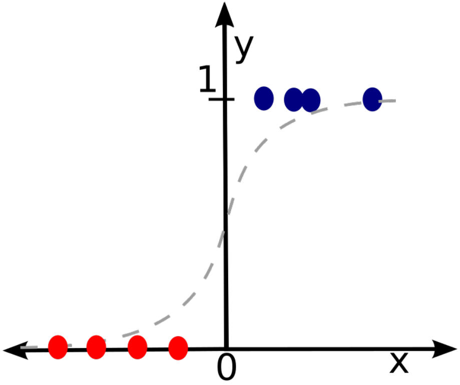
Logistic regression

Nominal classes



Logistic regression

Loss function



Logistic regression

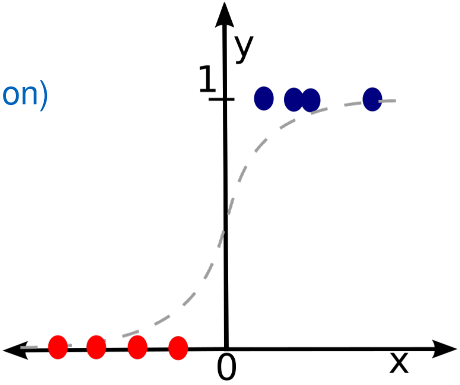
Loss function

Linear regression

$$h(x) = \vec{w}^T x$$

Logistic regression (Sigmoid function)

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



Logistic regression

Loss function

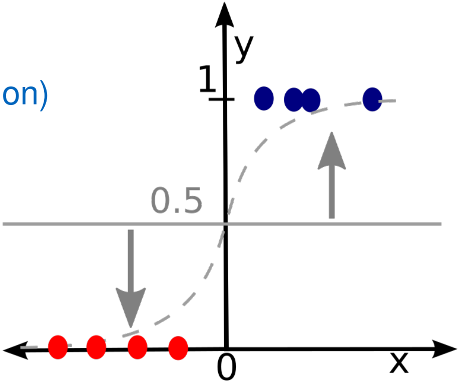
Linear regression

$$h(x) = \vec{w}^T x$$

Logistic regression (Sigmoid function)

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

$$y = \begin{cases} 1 & \text{if } h(x) \geq 0.5 \\ 0 & \text{else} \end{cases}$$



Logistic regression

Linear vs logistic regression

Linear regression: the model computes the weighted some of the input features (plus a bias term).

$$\hat{y} = w_0 x_0 + w_1 x_1 + ... + w_n x_n = W^T X$$

Logistic regression

Linear vs logistic regression

Linear regression: the model computes the weighted some of the input features (plus a bias term).

$$\hat{y} = w_0x_0 + w_1x_1 + ... + w_nx_n = W^TX$$

Logistic regression the model computes the logistic of the weighted some of the input features (plus a bias term).

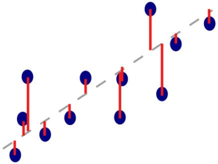
$$z = w_0x_0 + w_1x_1 + ... + w_nx_n = W^TX$$

$$\hat{y} = \sigma(z) = \frac{1}{1 + e^{-z}} = \frac{1}{1 + e^{-W^TX}}$$

How to learn parameters of the model W?

Logistic regression

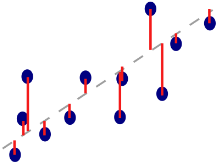
Loss function



- A reasonable model should predict $\hat{y}$ close to y, at least in the training data

Logistic regression

Loss function



- A reasonable model should predict $\hat{y}$ close to y, at least in the training data
- Cost function $L(W)$: the Mean Squared Error (MSE)

$$L(W) = \frac{1}{m}\sum_{i}^{m}(\hat{y}^{(i)} - y^{(i)})^2$$

Logistic regression

Loss function

- Naive choice: minimizing the Mean Squared Error(MSE)

$$L(W) = \frac{1}{m}\sum_{i}^{m}(\hat{y}^{(i)} - y^{(i)})^2$$

$$L(W) = \frac{1}{m}\sum_{i}^{m}(\frac{1}{1 + e^{-W^T x^{(i)}}} - y^{(i)})^2$$

Logistic regression

Loss function

- Naive choice: minimizing the Mean Squared Error(MSE)

$$L(W) = \frac{1}{m}\sum_{i}^{m}(\hat{y}^{(i)} - y^{(i)})^2$$

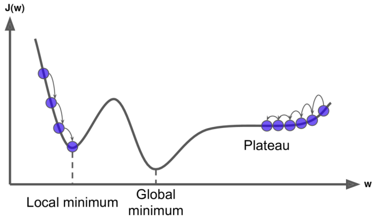
$$L(W) = \frac{1}{m}\sum_{i}^{m}(\frac{1}{1 + e^{-W^T x^{(i)}}} - y^{(i)})^2$$

- This cost function is a non-convex function for the optimizer.

Logistic regression

Loss function

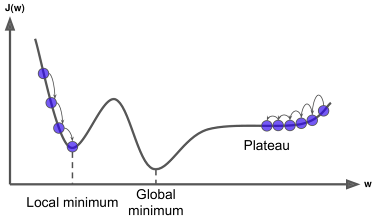
- What does non-convex mean?



Logistic regression

Loss function

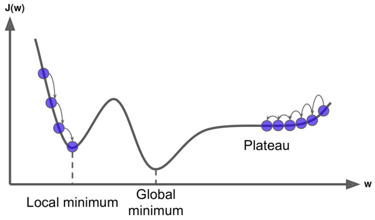
- What does non-convex mean?
- Given any two points on the curve there will be at least one intersection



Logistic regression

Loss function

- What does non-convex mean?
- Given any two points on the curve there will be at least one intersection
- Problem: The optimizer might end up in a local minima.

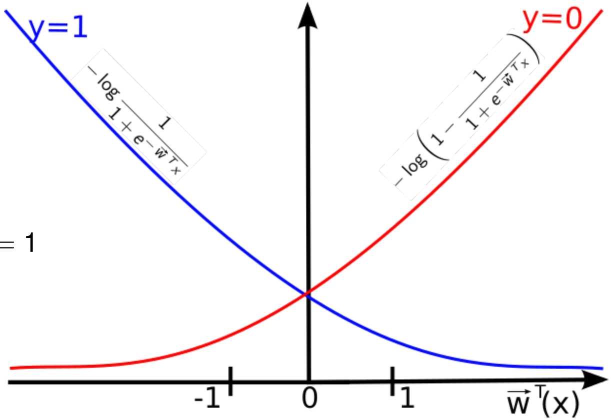


Logistic regression

Loss function

$$y = \begin{cases} 1 & \text{if } h(x) \geq 0.5 \\ 0 & \text{else} \end{cases}$$

$$L[(\mathcal{X}, \mathcal{Y}), h(\cdot)] = \begin{cases} -\log(h(x)) & \text{if } y = 1 \\ -\log(1 - h(x)) & \text{else} \end{cases}$$



Logistic regression

Gradient descent

Goal: find w that minimizes

$$E(W) = -\frac{1}{n}\sum_{i}^{n}(y^{(i)}log(\hat{y}^{(i)}) + (1 - y^{(i)})log(1 - \hat{y}^{(i)}))$$

Logistic regression

Gradient descent

Goal: find w that minimizes

$$E(W) = -\frac{1}{n}\sum_{i}^{n}(y^{(i)}log(\hat{y}^{(i)}) + (1 - y^{(i)})log(1 - \hat{y}^{(i)}))$$

- Randomly initialize the parameters, and repeat the following steps, until the stopping criterion:
  - Find a descent direction $\frac{\partial E(W)}{\partial W}$
  - Choose a step size δ
  - Update the parameters: $w^{(next)} = w - \delta\frac{\partial E(W)}{\partial W}$ (simultaneously for all parameters)

Logistic regression

Gradient descent

$$\hat{y} = \sigma\left(w^\top x\right) = \frac{1}{1 + e^{-w^\top x}}$$

$$E(w) = \frac{1}{n}\sum_{i}^{n} \text{cost}\left(\hat{y}^{(i)}, y^{(i)}\right) = -\frac{1}{n}\sum_{i}^{n}\left(y^{(i)} \log\left(\hat{y}^{(i)}\right) + \left(1 - y^{(i)}\right) \log\left(1 - \hat{y}^{(i)}\right)\right)$$

$$\frac{\partial E(w)}{\partial w_j} = \frac{1}{n}\sum_{i}^{n} -\left(y^{(i)}\frac{1}{\hat{y}^{(i)}} - \left(1 - y^{(i)}\right)\frac{1}{1 - \hat{y}^{(i)}}\right)\frac{\partial \hat{y}^{(i)}}{\partial w_j}$$

$$= \frac{1}{n}\sum_{i}^{n} -\left(y^{(i)}\frac{1}{\hat{y}^{(i)}} - \left(1 - y^{(i)}\right)\frac{1}{1 - \hat{y}^{(i)}}\right)\hat{y}^{(i)}\left(1 - \hat{y}^{(i)}\right)\frac{\partial w^\top x}{\partial w_j}$$

$$= \frac{1}{n}\sum_{i}^{n} -\left(y^{(i)}\left(1 - \hat{y}^{(i)}\right) - \left(1 - y^{(i)}\right)\hat{y}^{(i)}\right)x_j$$

$$= \frac{1}{n}\sum_{i}^{n}\left(\hat{y}^{(i)} - y^{(i)}\right)x_j$$

Same update rule as linear regression! Coincidence?

Outline

The perceptron algorithm

Binary classification ($\overrightarrow{y} \in \{-1, 1\}^n$) with $\overrightarrow{x_i} \in \mathcal{X}$, $i \in \{1, \ldots, n\}$.

The perceptron algorithm

Binary classification ($\overrightarrow{y} \in \{-1, 1\}^n$) with $\overrightarrow{x_i} \in \mathcal{X}$, $i \in \{1, \ldots, n\}$.

We define a nonlinear hypothesis function as:

$$h(\overrightarrow{w}^T \overrightarrow{x}) = \begin{cases} +1, & \overrightarrow{w}^T \overrightarrow{x} \geq 0 \\ -1, & \overrightarrow{w}^T \overrightarrow{x} < 0. \end{cases}$$

The perceptron algorithm

Let $\mathcal{D}^t \subseteq \mathcal{X}$ describe the set of all misclassified x_i at step t and the loss function

$$L[\overrightarrow{x_i}, \overrightarrow{y}, \overrightarrow{w}, h(\cdot)] = \begin{cases} -\overrightarrow{w}^T \overrightarrow{x_i} y_i & ; \overrightarrow{x_i} \in \mathcal{D} \\ 0 & ; \text{else} \end{cases}$$

The perceptron algorithm

Let $\mathcal{D}^t \subseteq \mathcal{X}$ describe the set of all misclassified x_i at step t and the loss function

$$L[\overrightarrow{x_i}, \overrightarrow{y}, \overrightarrow{w}, h(\cdot)] = \begin{cases} -\overrightarrow{w}^T \overrightarrow{x_i} y_i & ; \overrightarrow{x_i} \in \mathcal{D} \\ 0 & ; \text{else} \end{cases}$$

$L[\overrightarrow{x_i}, \overrightarrow{y}, \overrightarrow{w}, h(\cdot)]$ is piecewise linear:

linear in regions of the feature space where x_i are misclassfied

0 in regions where it is classified correctly

The perceptron algorithm

Let $\mathcal{D}^t \subseteq \mathcal{X}$ describe the set of all misclassified x_i at step t and the loss function

$$L[\overrightarrow{x_i}, \overrightarrow{y}, \overrightarrow{w}, h(\cdot)] = \begin{cases} -\overrightarrow{w}^T \overrightarrow{x_i} y_i & ; \overrightarrow{x_i} \in \mathcal{D} \\ 0 & ; \text{else} \end{cases}$$

$L[\overrightarrow{x_i}, \overrightarrow{y}, \overrightarrow{w}, h(\cdot)]$ is piecewise linear:

linear in regions of the feature space where x_i are misclassfied

0 in regions where it is classified correctly

Apply stochastic gradient descent to this loss function:

$$\overrightarrow{w}^{t+1} = \overrightarrow{w}^t - \begin{cases} \delta \frac{\partial L[\overrightarrow{x_i}, \overrightarrow{y}, \overrightarrow{w}, h(\cdot)]}{\partial \overrightarrow{w}} & ; x_i \in \mathcal{D} \\ 0 & ; \text{else} \end{cases}$$

$$= \overrightarrow{w}^t + \begin{cases} \delta \overrightarrow{x_i} y_i & ; x_i \in \mathcal{D} \\ 0 & ; \text{else} \end{cases}$$

The perceptron algorithm

Interpretation of the learning function

$$\overrightarrow{w}^{t+1} = \overrightarrow{w}^{t} + \begin{cases} \delta \overrightarrow{x_i} y_i & x_i \in \mathcal{D} \\ 0 & \text{else} \end{cases}$$

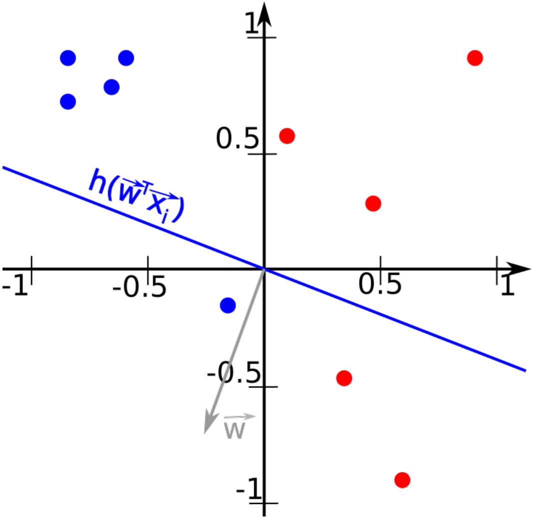
for each x_i:

correct classification: weight vector remains unchanged

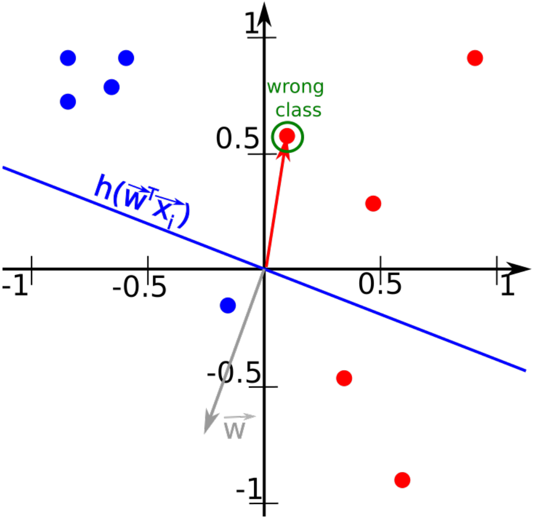
incorrect classification:

$$\begin{aligned} y_i = 1 &: \quad \text{add vector } \overrightarrow{x_i} \\ y_i = -1 &: \quad \text{subtract vector } \overrightarrow{x_i} \end{aligned}$$

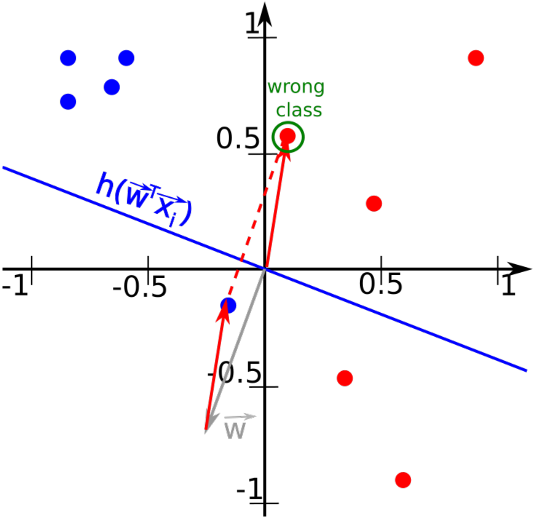
The perceptron algorithm



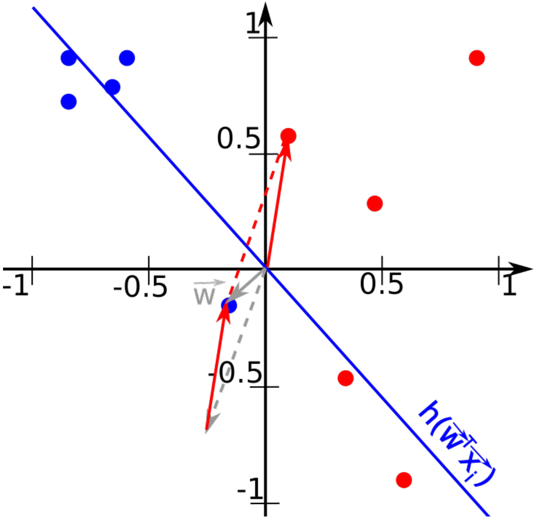
The perceptron algorithm



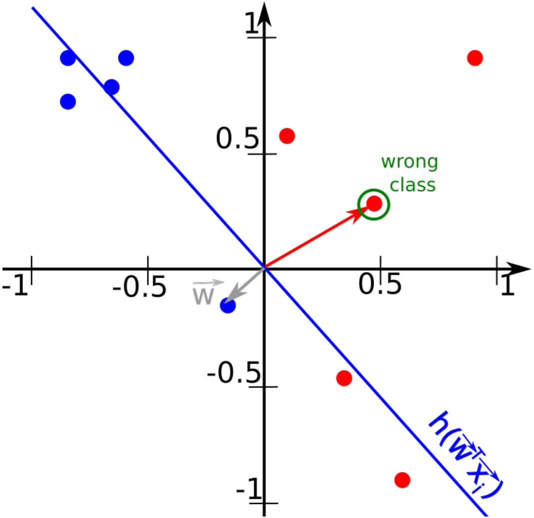
The perceptron algorithm



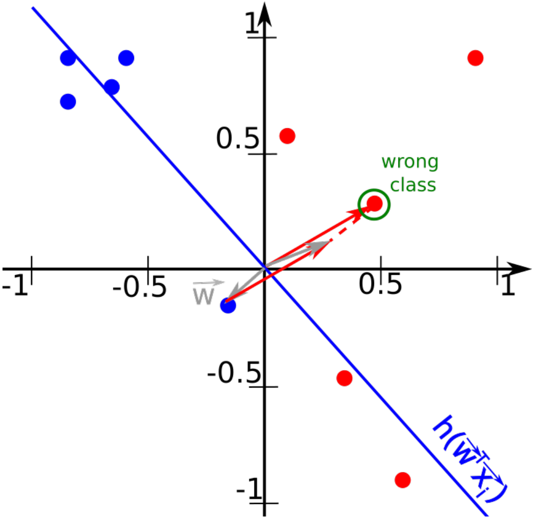
The perceptron algorithm



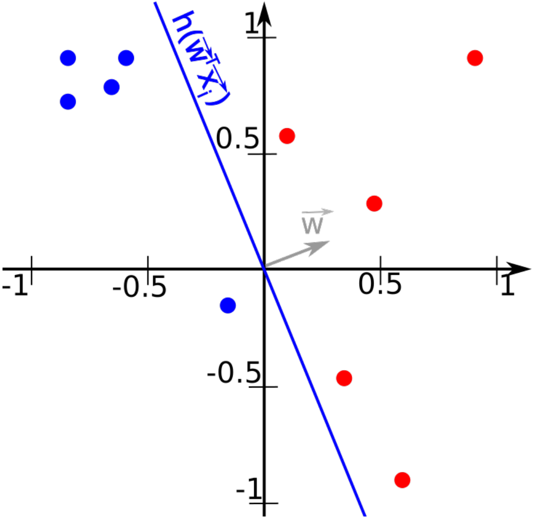
The perceptron algorithm



The perceptron algorithm



The perceptron algorithm



The perceptron algorithm

Perceptron convergence theorem

IFF the training data is linearly separable, then the perceptron learning algorithm will always find an exact solution in finite number of steps.

→ Number of steps required might be large

→ Until convergence, not possible to distinguish separable problem from non-separable

→ For non-separable data sets the algorithm will never converge

Try yourself:
https://sergedesmedt.github.io/MathOfNeuralNetworks/PerceptronLearningMath.html

Outline

Recap: linear regression

Logistic regression

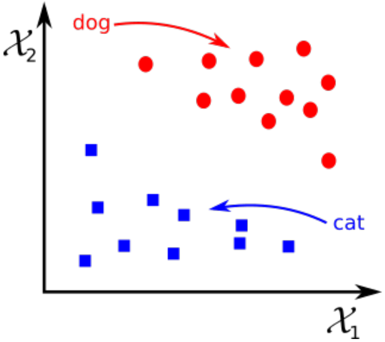
The Perceptron algorithm

Support Vector Machines

Multiclass classification

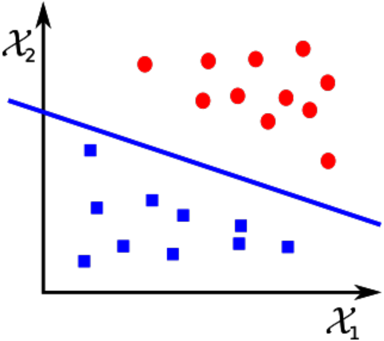
Support vector machines (SVM)

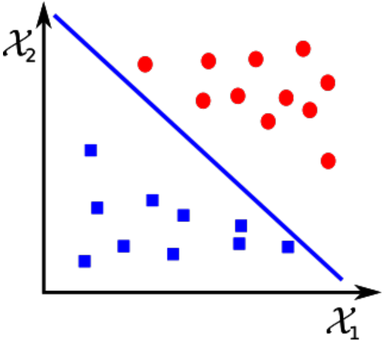
Large margin classifier



Support vector machines (SVM)

Large margin classifier



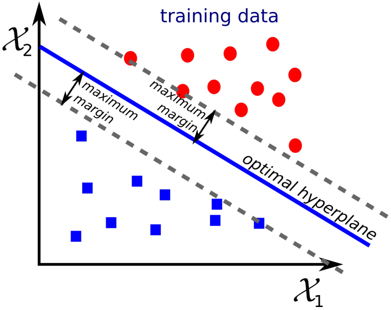


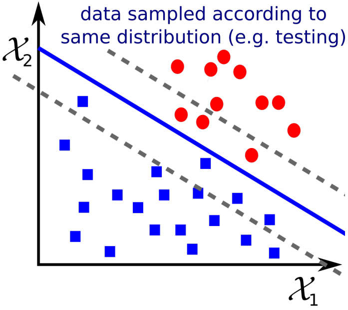
Support vector machines (SVM)

Large margin classifier

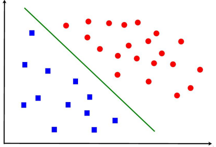
The goal for support vector machines is to find a linear and separating hyperplane with the largest margin to the outer points in all sets

If needed, map all points into a higher dimensional space until such a plane exists





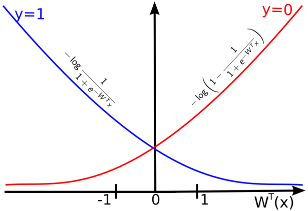
Support vector machines (SVM)



Contribution of a single sample to the overall loss:

Logistic regression

$$L[(\mathcal{X},\mathcal{Y}),h(\cdot)] = -y\cdot\log\left(1-\frac{1}{1+e^{-\overrightarrow{w}^{T}x}}\right)-(1-y)\cdot\log\frac{1}{1+e^{-\overrightarrow{w}^{T}x}}$$

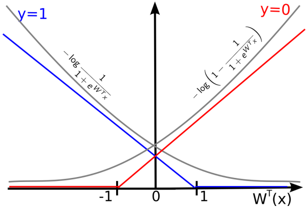


Support vector machines (SVM)

Contribution of a single sample to the overall loss:

SVM

$$L[(\mathcal{X},\mathcal{Y}),h(\cdot)] = -y\cdot \text{cost}_{y=1}(\overrightarrow{w}^T x) + -(1-y)\cdot \text{cost}_{y=0}(\overrightarrow{w}^T x)$$

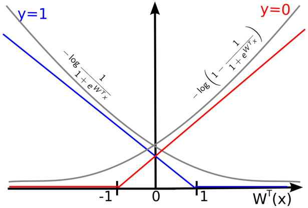


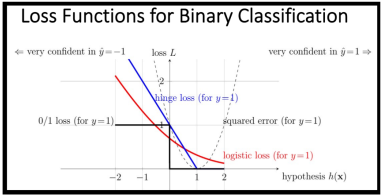
Support vector machines (SVM)

Contribution of a single sample to the overall loss:

SVM

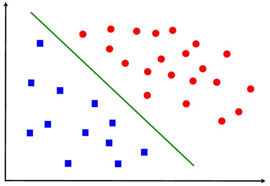
$$L[(\mathcal{X},\mathcal{Y}),h(\cdot)] = -y\cdot \text{cost}_{y=1}(\vec{w}^{T}x) + -(1-y)\cdot \text{cost}_{y=0}(\vec{w}^{T}x)$$





Support vector machines (SVM)

Cost function



Logistic regression

$$\min_{W} \quad \frac{1}{m}\left[\sum_{i=1}^{m} y_i \left(-\log\left(1-\frac{1}{1+e^{-\vec{w}^T x_i}}\right)\right)+(1-y_i)\left(-\log\frac{1}{1+e^{-\vec{w}^T x_i}}\right)\right]+\frac{\lambda}{2m}\sum_{j=1}^{n} w_j^2$$

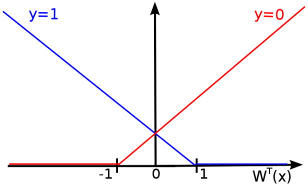
SVM

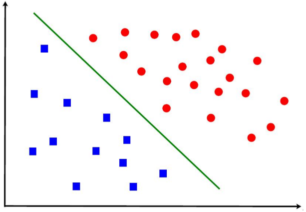
$$\min_{W} \quad \frac{1}{\lambda'}\sum_{i=1}^{m}\left[y_i \text{cost}_{y=1}(\vec{w}^T x_i)+(1-y_i)\text{cost}_{y=0}(\vec{w}^T x_i)\right]+\frac{1}{2}\sum_{j=1}^{n} w_j^2$$

λ has a similar effect on the overall term as $\frac{1}{\lambda'}$

Support vector machines (SVM)

SVM hypothesis

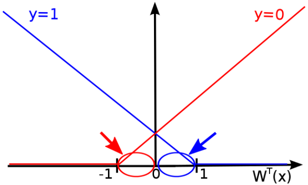


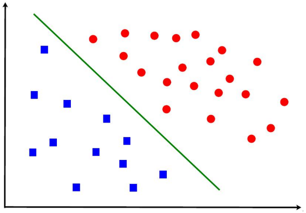


$$\min_{W} \frac{1}{\lambda'} \sum_{i=1}^{m} \left[y_i \text{cost}_{y=1}(\vec{w}^T x_i) + (1 - y_i) \text{cost}_{y=0}(\vec{w}^T x_i) \right] + \frac{1}{2} \sum_{j=1}^{n} w_j^2$$

Support vector machines (SVM)

SVM hypothesis



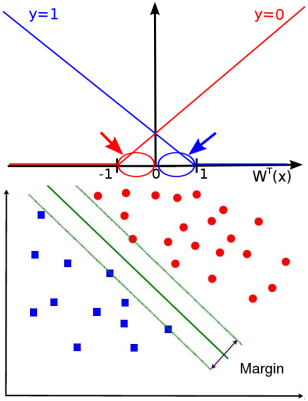


$$\vec{w}^T x \begin{cases} \geq 0 \\ < 0 \end{cases} \text{ sufficient}$$

$$\min_{W} \frac{1}{\lambda'} \sum_{i=1}^{m} \left[y_i \text{cost}_{y=1}(\vec{w}^T x_i) + (1 - y_i) \text{cost}_{y=0}(\vec{w}^T x_i) \right] + \frac{1}{2} \sum_{j=1}^{n} w_j^2$$

Support vector machines (SVM)

SVM hypothesis



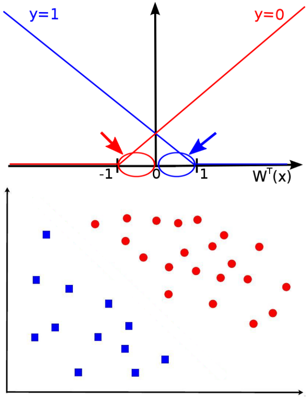
$$\vec{w}^T x \begin{cases} \geq 0 \\ < 0 \end{cases} \text{sufficient}$$

$$\vec{w}^T x \begin{cases} \geq 1 \\ \leq -1 \end{cases} \Rightarrow \text{confidence}$$

$$\min_{W} \frac{1}{\lambda'} \sum_{i=1}^{m} \left[y_i \mathrm{cost}_{y=1}(\vec{w}^T x_i) + (1 - y_i)\mathrm{cost}_{y=0}(\vec{w}^T x_i) \right] + \frac{1}{2} \sum_{j=1}^{n} w_j^2$$

Support vector machines (SVM)

SVM hypothesis



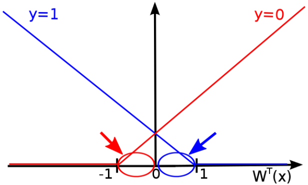
$$\vec{w}^T x \begin{cases} \geq & 1 \\ \leq & -1 \end{cases} \Rightarrow \text{confidence}$$

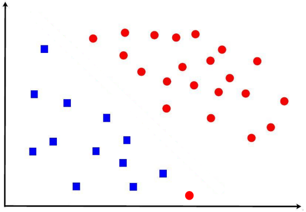
Outliers: Elastic decision boundary

$$\min_W \frac{1}{\lambda'} \sum_{i=1}^{m} \left[y_i \text{cost}_{y=1}(\vec{w}^T x_i) + (1 - y_i)\text{cost}_{y=0}(\vec{w}^T x_i) \right] + \frac{1}{2} \sum_{j=1}^{n} w_j^2$$

Support vector machines (SVM)

SVM hypothesis





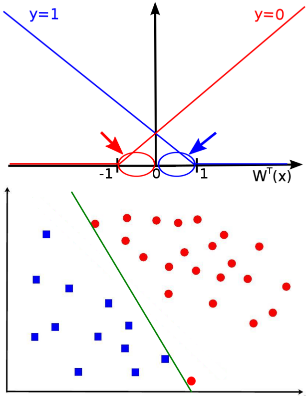
$$\vec{w}^T x \begin{cases} \geq 1 \\ \leq -1 \end{cases} \Rightarrow \text{confidence}$$

Outliers: Elastic decision boundary

$$\min_{W} \frac{1}{\lambda'} \sum_{i=1}^{m} \left[y_i \text{cost}_{y=1}(\vec{w}^T x_i) + (1 - y_i)\text{cost}_{y=0}(\vec{w}^T x_i) \right] + \frac{1}{2} \sum_{j=1}^{n} w_j^2$$

Support vector machines (SVM)

SVM hypothesis



$$\vec{w}^T x \begin{cases} \geq & 1 \\ \leq & -1 \end{cases} \Rightarrow \text{confidence}$$

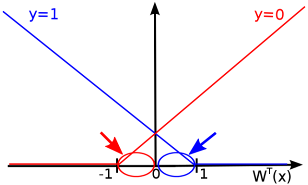
Outliers: Elastic decision boundary

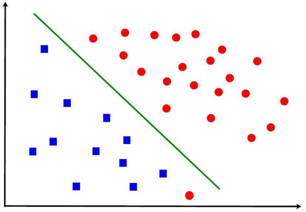
small λ' stricter boundary at the cost of smaller margin

$$\min_W \frac{1}{\lambda'} \sum_{i=1}^{m} \left[y_i \text{cost}_{y=1}(\vec{w}^T x_i) + (1 - y_i)\text{cost}_{y=0}(\vec{w}^T x_i) \right] + \frac{1}{2} \sum_{j=1}^{n} w_j^2$$

Support vector machines (SVM)

SVM hypothesis





$$\vec{w}^T x \begin{cases} \geq & 1 \\ \leq & -1 \end{cases} \Rightarrow \text{confidence}$$

Outliers: Elastic decision boundary

large λ' tolerates outliers

$$\min_W \frac{1}{\lambda'} \sum_{i=1}^{m} \left[y_i \text{cost}_{y=1}(\vec{w}^T x_i) + (1 - y_i)\text{cost}_{y=0}(\vec{w}^T x_i) \right] + \frac{1}{2} \sum_{j=1}^{n} w_j^2$$

Support vector machines (SVM)

Large margin classifier

$$\min_{W} \frac{1}{\lambda'} \sum_{i=1}^{m} \left[y_i \text{cost}_{y=1}(\overrightarrow{w}^T x_i) + (1 - y_i)\text{cost}_{y=0}(\overrightarrow{w}^T x_i) \right] + \frac{1}{2} \sum_{j=1}^{n} w_j^2$$

Support vector machines (SVM)

Large margin classifier

$$\min_{W} \frac{1}{\lambda'} \sum_{i=1}^{m} \left[y_i \text{cost}_{y=1}(\overrightarrow{w}^T x_i) + (1 - y_i)\text{cost}_{y=0}(\overrightarrow{w}^T x_i) \right] + \frac{1}{2} \sum_{j=1}^{n} w_j^2$$

Rewrite the SVM optimisation problem as

$$\begin{aligned} \min_{W} \quad & \tfrac{1}{2} \textstyle\sum_{j=1}^{n} w_j^2 \\ s.t. \quad & \overrightarrow{w}^T x_i \geq 1 \quad \text{if } y_i = 1 \\ & \overrightarrow{w}^T x_i \leq -1 \quad \text{if } y_i = 0 \end{aligned}$$

Support vector machines (SVM)

Large margin classifier

$$\min_{w} \quad \frac{1}{2}\sum_{j=1}^{n} w_j^2$$

$$s.t. \qquad \overrightarrow{w}^T x_i \geq 1 \;\text{ if } y_i = 1$$
$$\overrightarrow{w}^T x_i \leq -1 \text{ if } y_i = 0$$

Support vector machines (SVM)

Large margin classifier

$$
\begin{aligned}
\min_{w} \quad & \tfrac{1}{2}\textstyle\sum_{j=1}^{n} w_j^2 = \tfrac{1}{2}\left(\sqrt{w_1^2 + \cdots + w_n^2}\right)^2 \\
s.t. \quad & \vec{w}^T x_i \geq 1 \ \text{ if } y_i = 1 \\
& \vec{w}^T x_i \leq -1 \text{ if } y_i = 0
\end{aligned}
$$

Support vector machines (SVM)

Large margin classifier

$$\min_{w} \quad \frac{1}{2}\sum_{j=1}^{n} w_j^2 = \frac{1}{2}\left(\sqrt{w_1^2 + \cdots + w_n^2}\right)^2 \quad = \frac{1}{2}||\vec{w}||^2$$

$$s.t. \qquad \vec{w}^T x_i \geq 1 \ \text{ if } y_i = 1$$

$$\vec{w}^T x_i \leq -1 \text{ if } y_i = 0$$

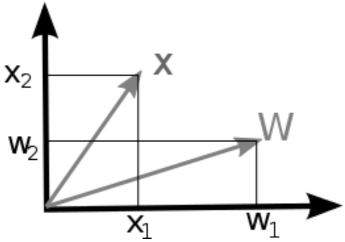
Support vector machines (SVM)

Large margin classifier

$$\min_{w} \quad \frac{1}{2}\sum_{j=1}^{n} w_j^2 = \frac{1}{2}\left(\sqrt{w_1^2+\cdots+w_n^2}\right)^2 \quad = \frac{1}{2}||\vec{w}||^2$$

$$s.t. \qquad \vec{w}^T x_i \geq 1 \text{ if } y_i = 1$$

$$\vec{w}^T x_i \leq -1 \text{ if } y_i = 0$$



$$\vec{w}^T x = w_1 x_1 + w_2 x_2$$

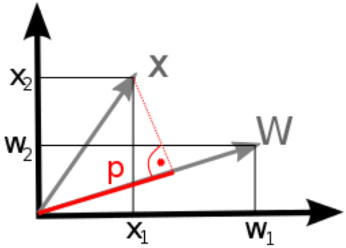
Support vector machines (SVM)

Large margin classifier

$$\min_{w} \quad \frac{1}{2}\sum_{j=1}^{n} w_j^2 = \frac{1}{2}\left(\sqrt{w_1^2 + \cdots + w_n^2}\right)^2 \quad = \frac{1}{2}||\overrightarrow{w}||^2$$

$$s.t. \qquad \overrightarrow{w}^T x_i \geq 1 \text{ if } y_i = 1$$

$$\overrightarrow{w}^T x_i \leq -1 \text{ if } y_i = 0$$



$$\overrightarrow{w}^T x = w_1 x_1 + w_2 x_2 = ||\overrightarrow{w}|| \cdot p$$

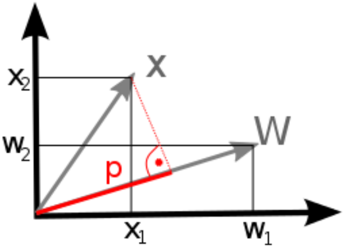
Support vector machines (SVM)

Large margin classifier

$$\min_{w} \quad \frac{1}{2}\sum_{j=1}^{n} w_j^2 = \frac{1}{2}\left(\sqrt{w_1^2+\cdots+w_n^2}\right)^2 \quad = \frac{1}{2}||\vec{w}||^2$$

$$s.t. \qquad \vec{w}^{\,T} x_i \geq 1 \;\text{ if } y_i = 1 \qquad \rightarrow ||\vec{w}|| \cdot p_i \geq 1$$

$$\vec{w}^{\,T} x_i \leq -1 \;\text{ if } y_i = 0 \qquad \rightarrow ||\vec{w}|| \cdot p_i \leq -1$$



$$\vec{w}^{\,T} x = w_1 x_1 + w_2 x_2 = ||\vec{w}|| \cdot p$$

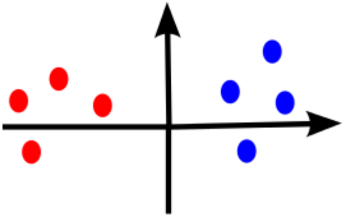
Support vector machines (SVM)

Large margin classifier

$$\min_{W} \quad \frac{1}{2}\sum_{j=1}^{n} w_j^2 = \frac{1}{2}\left(\sqrt{w_1^2+\cdots+w_n^2}\right)^2 \quad = \frac{1}{2}||\vec{w}||^2$$

$$s.t. \quad \vec{w}^T x_i \geq 1 \text{ if } y_i = 1 \quad \rightarrow ||\vec{w}|| \cdot p_i \geq 1$$

$$\vec{w}^T x_i \leq -1 \text{ if } y_i = 0 \quad \rightarrow ||\vec{w}|| \cdot p_i \leq -1$$



Which decision boundaray is found?

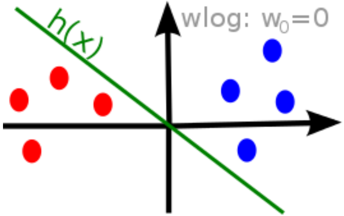
Support vector machines (SVM)

Large margin classifier

$$\min_{W} \quad \frac{1}{2}\sum_{j=1}^{n} w_j^2 = \frac{1}{2}\left(\sqrt{w_1^2 + \cdots + w_n^2}\right)^2 \quad = \frac{1}{2}||\vec{w}||^2$$

$$s.t. \quad \vec{w}^T x_i \geq 1 \text{ if } y_i = 1 \quad \rightarrow ||\vec{w}|| \cdot p_i \geq 1$$

$$\vec{w}^T x_i \leq -1 \text{ if } y_i = 0 \quad \rightarrow ||\vec{w}|| \cdot p_i \leq -1$$



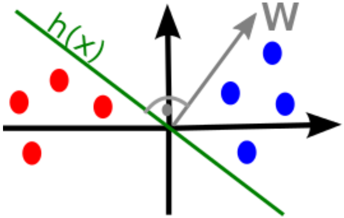
Which decision boundaray is found?

Support vector machines (SVM)

Large margin classifier

$$\min_{W} \quad \tfrac{1}{2}\textstyle\sum_{j=1}^{n} w_j^2 = \tfrac{1}{2}\left(\sqrt{w_1^2 + \cdots + w_n^2}\right)^2 \quad = \frac{1}{2}||\vec{w}||^2$$

$$\begin{aligned} s.t. \quad & \vec{w}^{\,T} x_i \geq 1 \ \text{ if } y_i = 1 && \rightarrow ||\vec{w}|| \cdot p_i \geq 1 \\ & \vec{w}^{\,T} x_i \leq -1 \text{ if } y_i = 0 && \rightarrow ||\vec{w}|| \cdot p_i \leq -1 \end{aligned}$$



Which decision boundaray is found?

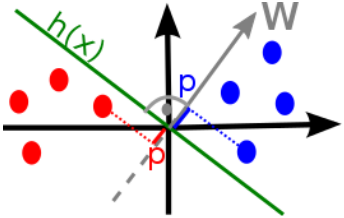
$h(x) = w_1 x_1 + w_2 x_2$

$\rightarrow$ W orthogonal to all x with $h(x) = 0$

Support vector machines (SVM)

Large margin classifier

$$
\begin{aligned}
\min_{W} \quad & \tfrac{1}{2}\textstyle\sum_{j=1}^{n} w_j^2 = \tfrac{1}{2}\left(\sqrt{w_1^2+\cdots+w_n^2}\right)^2 && = \frac{1}{2}||\vec{w}||^2 \\
s.t. \quad & \vec{w}^{\,T}x_i \geq 1 \;\text{ if } y_i = 1 && \rightarrow ||\vec{w}|| \cdot p_i \geq 1 \\
& \vec{w}^{\,T}x_i \leq -1 \text{ if } y_i = 0 && \rightarrow ||\vec{w}|| \cdot p_i \leq -1
\end{aligned}
$$



Which decision boundaray is found?

$$h(x) = w_1x_1 + w_2x_2$$

$\rightarrow$ W orthogonal to all x with $h(x) = 0$

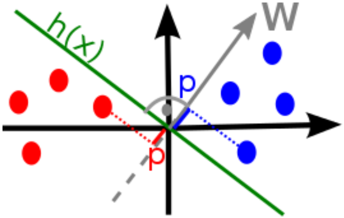
Support vector machines (SVM)

Large margin classifier

$$\min_{W} \quad \tfrac{1}{2}\sum_{j=1}^{n} w_j^2 = \tfrac{1}{2}\left(\sqrt{w_1^2+\cdots+w_n^2}\right)^2 = \frac{1}{2}||\vec{w}||^2$$

$$s.t. \qquad \vec{w}^T x_i \geq 1 \text{ if } y_i = 1 \quad \rightarrow ||\vec{w}|| \cdot p_i \geq 1$$

$$\vec{w}^T x_i \leq -1 \text{ if } y_i = 0 \quad \rightarrow ||\vec{w}|| \cdot p_i \leq -1$$



Which decision boundaray is found?

$$h(x) = w_1 x_1 + w_2 x_2$$

$\rightarrow$ W orthogonal to all x with $h(x) = 0$

$\Rightarrow$ $\min \frac{1}{2}||\vec{w}||^2$ and $||\vec{w}|| \cdot p_i \geq 1$ necessitate larger p_i

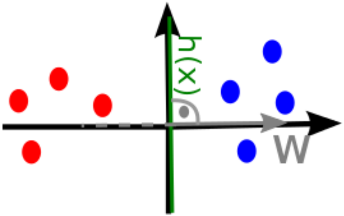
Support vector machines (SVM)

Large margin classifier

$$\min_{W} \quad \frac{1}{2}\sum_{j=1}^{n} w_j^2 = \frac{1}{2}\left(\sqrt{w_1^2 + \cdots + w_n^2}\right)^2 \quad = \frac{1}{2}||\overrightarrow{w}||^2$$

$$s.t. \quad \overrightarrow{w}^T x_i \geq 1 \;\text{ if } y_i = 1 \quad \rightarrow ||\overrightarrow{w}|| \cdot p_i \geq 1$$

$$\overrightarrow{w}^T x_i \leq -1 \text{ if } y_i = 0 \quad \rightarrow ||\overrightarrow{w}|| \cdot p_i \leq -1$$



Which decision boundaray is found?

$$h(x) = w_1 x_1 + w_2 x_2$$

$\rightarrow$ W orthogonal to all x with $h(x) = 0$

$\Rightarrow$ $\min \frac{1}{2}||\overrightarrow{w}||^2$ and $||\overrightarrow{w}|| \cdot p_i \geq 1$
necessitate larger p_i

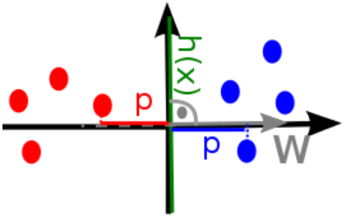
Support vector machines (SVM)

Large margin classifier

$$\min_{W} \quad \frac{1}{2}\sum_{j=1}^{n} w_j^2 = \frac{1}{2}\left(\sqrt{w_1^2+\cdots+w_n^2}\right)^2 \quad = \frac{1}{2}||\overrightarrow{w}||^2$$

$$s.t. \quad \overrightarrow{w}^T x_i \geq 1 \text{ if } y_i = 1 \quad \rightarrow ||\overrightarrow{w}|| \cdot p_i \geq 1$$

$$\overrightarrow{w}^T x_i \leq -1 \text{ if } y_i = 0 \quad \rightarrow ||\overrightarrow{w}|| \cdot p_i \leq -1$$



Which decision boundaray is found?

$$h(x) = w_1 x_1 + w_2 x_2$$

$\rightarrow$ W orthogonal to all x with $h(x) = 0$

$\Rightarrow$ $\min \frac{1}{2}||\overrightarrow{w}||^2$ and $||\overrightarrow{w}|| \cdot p_i \geq 1$ necessitate larger p_i

Outline

Recap: linear regression

Logistic regression

The Perceptron algorithm

Support Vector Machines

Multiclass classification

Multiclass classification

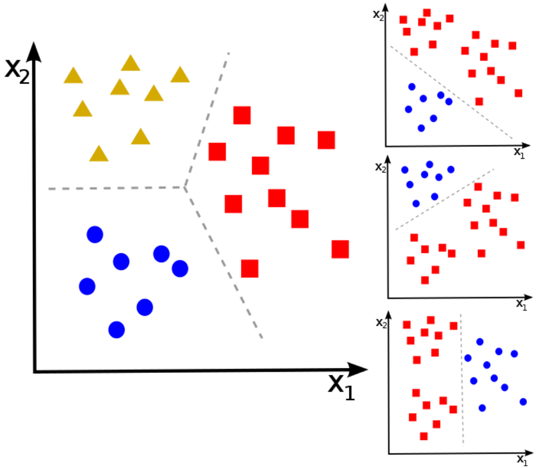
Multi-class: One-versus all:

Train classifiers for each class to obtain probability that x belongs to class i:

$$h_i(x) = P(y = i | \vec{x}, \vec{W})$$

then, choose

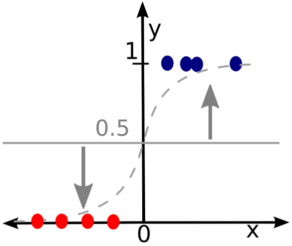
$$max_i\,(h_i(x))$$

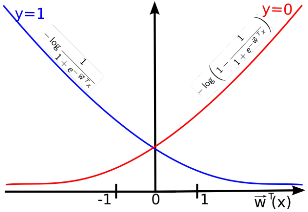


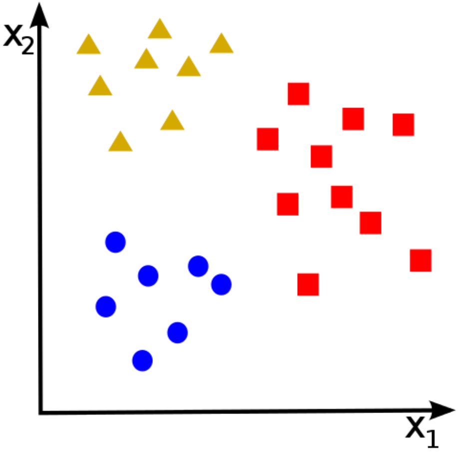
Multiclass classification

Multiple classes

Can we use logistic regression for problems with more than two classes?





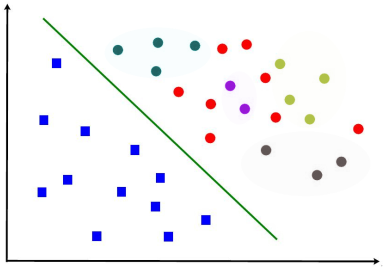


Application to several classes iteratively: One-versus-all

belongs to class 1 or not?

belongs to class 2 or not?

...



Questions?

Dariush Salami
dariush.salami@aalto.fi

Literature

- C.M. Bishop: Pattern recognition and machine learning, Springer, 2007.
- R.O. Duda, P.E. Hart, D.G. Stork: Pattern Classification, Wiley, 2001.