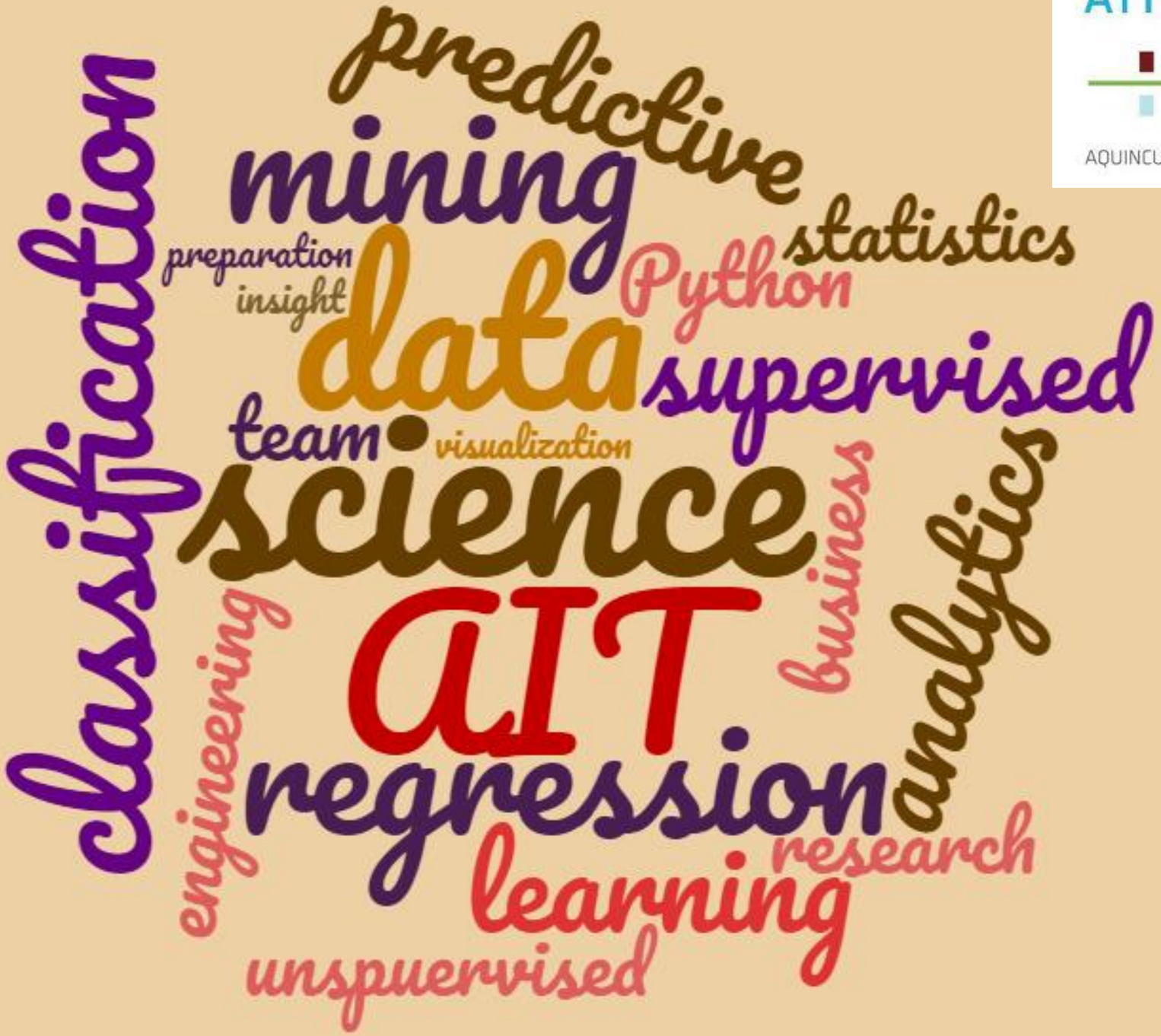
Data Science

April 14, 2020.
Logistic regression and SVM



AIT-BUDAPEST
AQUINCUM INSTITUTE OF TECHNOLOGY

Roland Molontay

New schedule of the semester

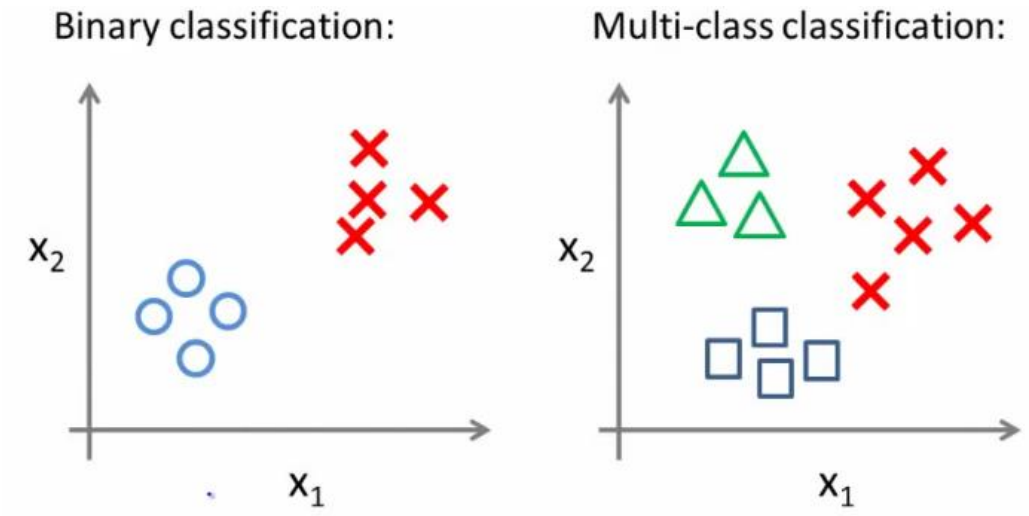
	Tuesday class	*Thursday*	*Friday class*	*Sunday*
W1 (02/03)				
W2 (02/10)			HW1 out	
W3 (02/17)				
W4 (02/24)		HW1 deadline	HW2 out	Forming teams
W5 (03/02)				
W6 (03/09)			CANCELLED	
BREAK	BREAK	HW2 deadline	BREAK	
W7 (13/23)	HW3 out		MIDTERM / Project plan	
W8 (03/30)				HW3 deadline
W9 (04/06)			GOOD FRIDAY	
W10 (04/13)	HW4 out	MILESTONE 1		
W11 (04/20)				HW4 deadline
W12 (04/27)			LABOR DAY	
W13 (05/04)		MILESTONE 2		
W14 (05/11)	FINAL		PROJECT presentations	

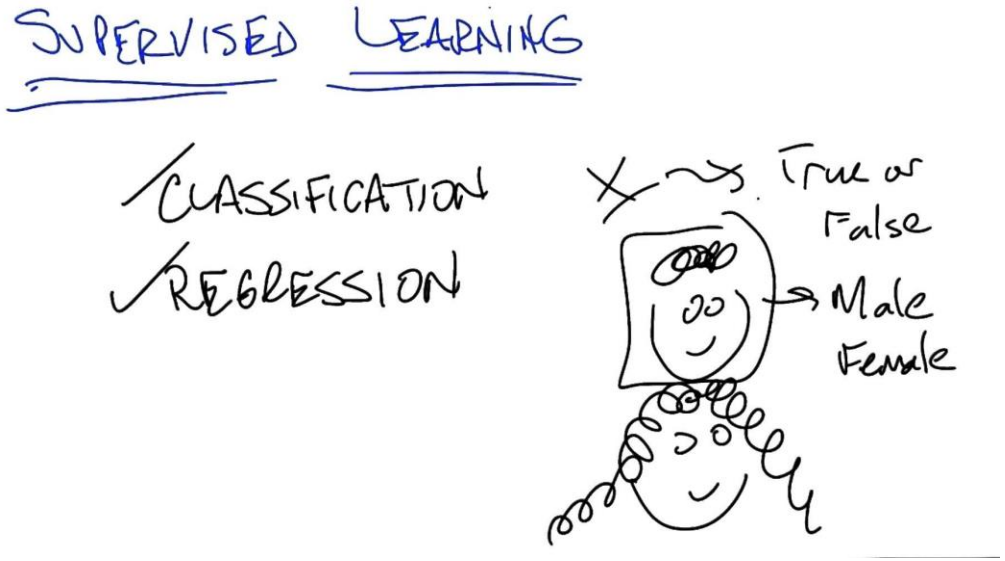
Project milestones

- W10: **Milestone 1**
  - Two-page-long report covering the followings:
    - Have you managed to gather the data? Do you have enough data of appropriate quality?
    - Did you collect the relevant related works? What useful information could you discover?
    - Initial data analysis steps
    - What next steps do you plan to take?
- W13: **Milestone 2**
  - Three-page-long report covering the followings:
    - Reviewing the related works
    - Data understanding and data preparation steps
    - More data analysis steps, implementing some models and evaluating them

Logistic regression

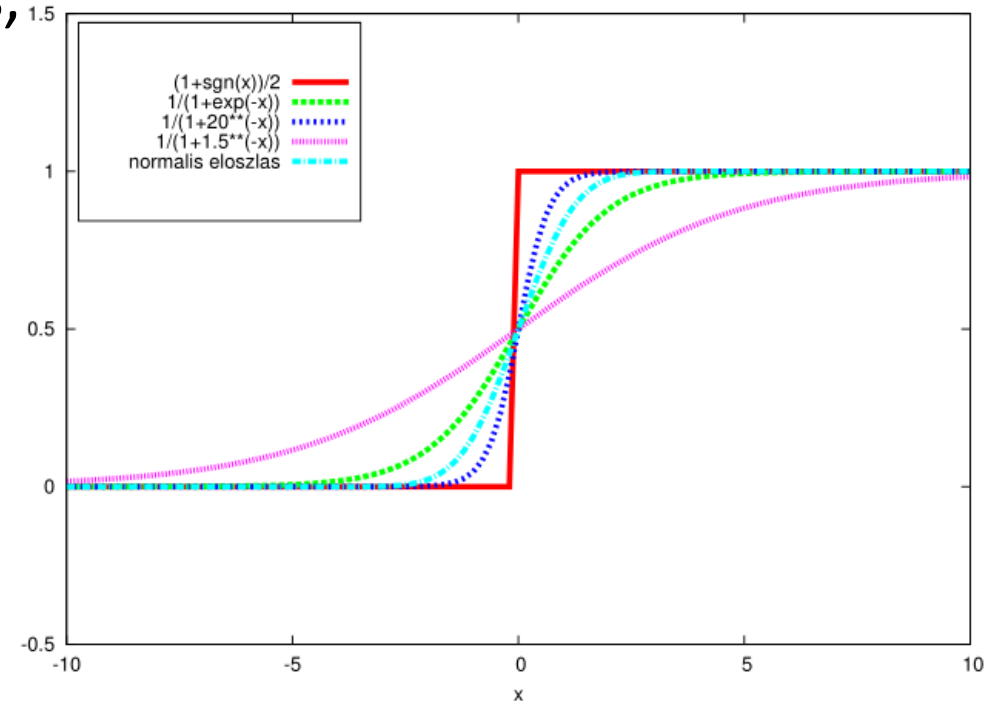
- It is a classification algorithm not a regression algorithm (despite its name)
- Classification via regression
- Binary classification algorithm





Classification via regression

- We fit a linear regression, if the predicted value is positive we assign the positive class, if the predicted value is negative we assign the negative class to the record
  - It means that we apply a step function on the result of the linear regression
  - We can also smooth the sign function:
    - It is desirable to have smoother function
    - It is a natural approach that the closer we are to the decision boundary, the more uncertain we are in our decision (the boundary is not sharp but rather smooth)
    - We can think of it as we predict how likely the positive label is, not the label itself

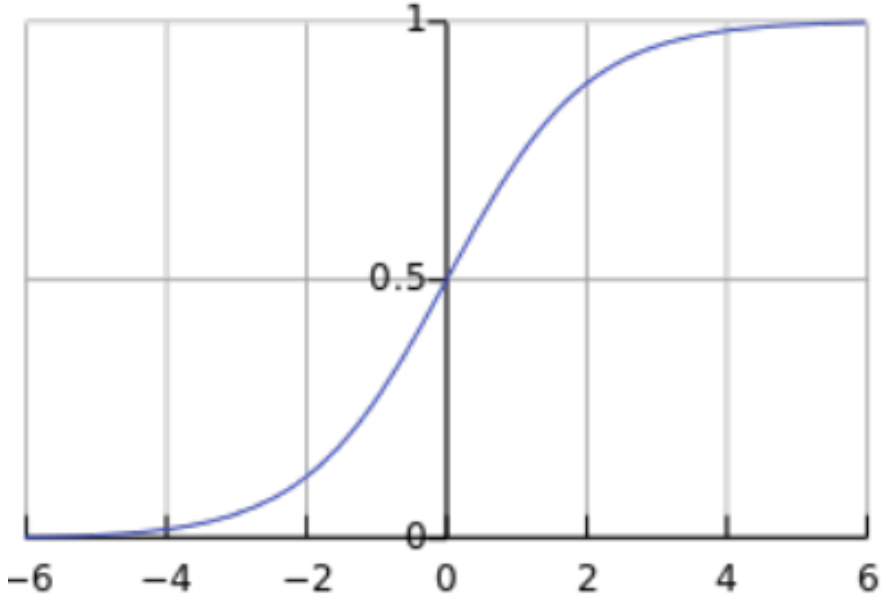


Logistic regression

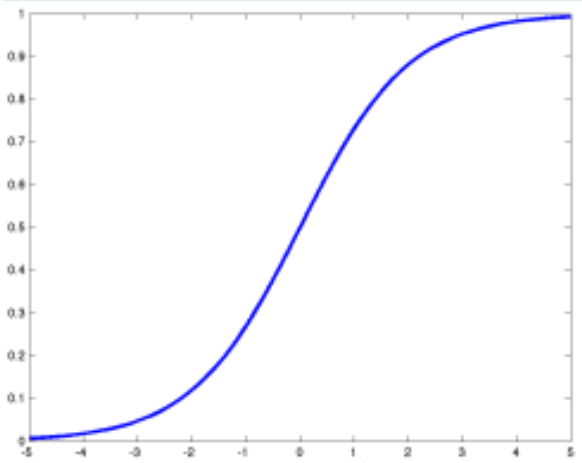
- Training data: $(\boldsymbol{x}_1, y_1), (\boldsymbol{x}_2, y_2), \ldots, (\boldsymbol{x}_n, y_n)$, where $\boldsymbol{x}_i = (x_{i1}, x_{i2}, \ldots, x_{ip})$
  - From now on we use a slightly modified notation as a matter of convenience Introduce the following notation: $x_{i0} = 1, \forall\, i$
  - Also: $\boldsymbol{x}_i = (x_{i0}, x_{i1}, x_{i2}, \ldots, x_{ip})$ and $\mathbf{w} = (w_0, w_1, w_2, \ldots, w_p)$
  - Using this new notation we can write $\boldsymbol{w}^T\, \boldsymbol{x_i}$ instead of $w_0 + \boldsymbol{w}^T\, \boldsymbol{x_i}$
- Hypothesis for linear regression:

$$\hat{y}_\mathrm{i} = h_{\boldsymbol{w}}(\boldsymbol{x}_i) = \boldsymbol{w}^T\, \boldsymbol{x_i}$$

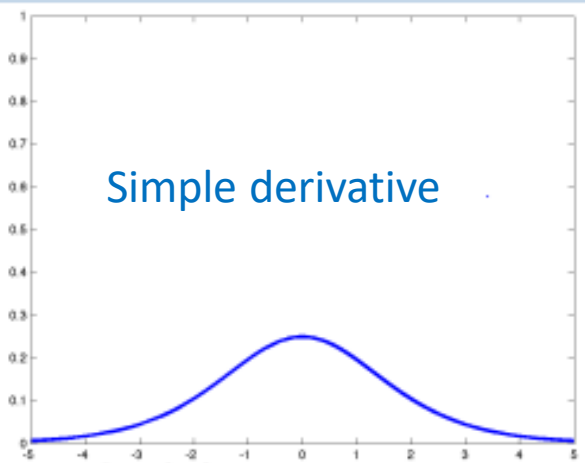
- Hypothesis for logistic regression: $h_{\boldsymbol{w}}(\boldsymbol{x}_i) = \sigma(\boldsymbol{w}^T\, \boldsymbol{x_i})$
  - Sigmoid function (logistic function) $\sigma(x) = \frac{1}{1+e^{-x}}$
  - Probabilistic interpretation



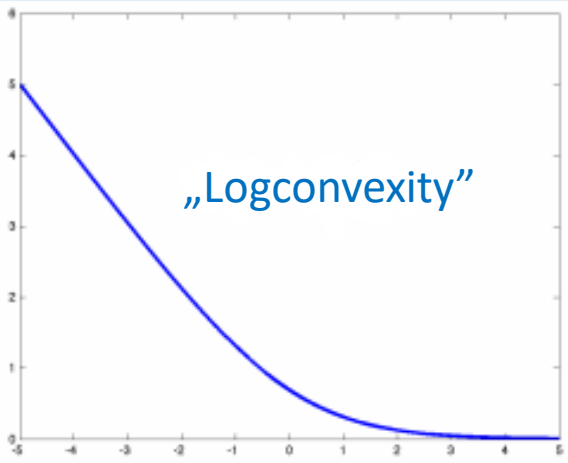
Characteristics of the sigmoid (logistic) function



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



$$\frac{\partial \sigma(z)}{\partial z} = \sigma(z)\sigma(-z)$$



$$-\log \sigma(z) = \log(1+e^{-z})$$

Simmetry:

$$\sigma(-z) = 1 - \sigma(z)$$

Log odds ratio:

$$\log \frac{\sigma(z)}{1-\sigma(z)} = z$$

What is the cost function?

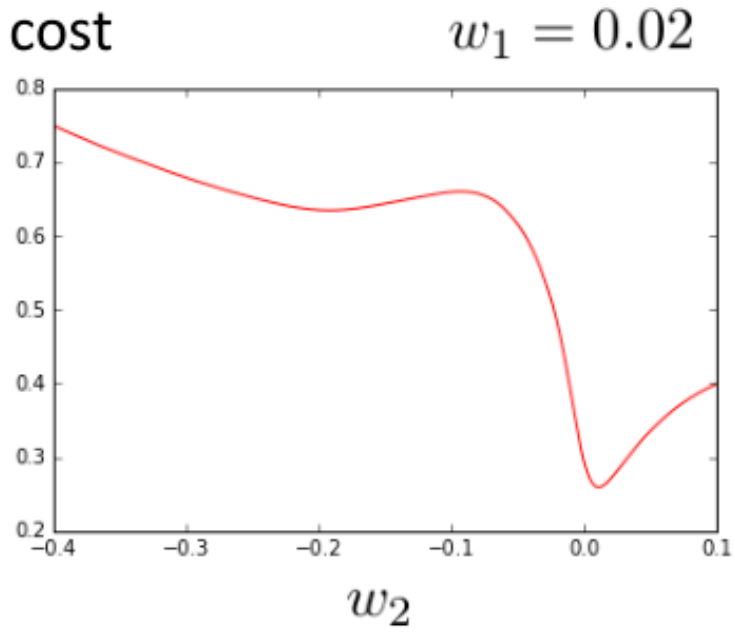
- For linear regression (with regularization)

$$\frac{1}{n}\sum_{i=1}^{n}(y_i - \boldsymbol{w}^T \boldsymbol{x_i})^2 + \lambda|\boldsymbol{w}|^2$$

- Can we use the „same” for logistic regression?

$$\frac{1}{n}\sum_{i=1}^{n}(y_i - \sigma(\boldsymbol{w}^T \boldsymbol{x_i}))^2 + \lambda|\boldsymbol{w}|^2$$

  - It is not practical because this function is not convex



- Sample data:

X=[[1,26], [1,33], [1,34], [25,33], [14,42], [32,56], [32,59], [1,120], [1,76], [1,80], [1,92], [25,135], [1,150], [1,26], [315,35], [218,39], [52,43]]
y=[0, 0, 0, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 0 , 0, 0, 1]

Cost function for logistic regression

- We form the cost function in the following way:
  - Cost for a single record (logarithmic cost)

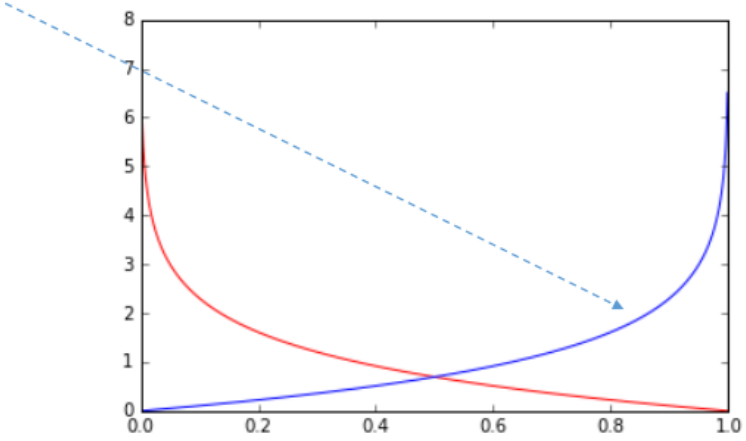
$$cost(h_w(x), y) = \begin{cases} -\log(h_w(x)) & \text{if } y = 1 \\ -\log(1 - h_w(x)) & \text{if } y = 0 \end{cases}$$

  - Total cost for the data set

$$\frac{1}{n}\sum_{i=1}^{n} cost(h_{\boldsymbol{w}}(\boldsymbol{x}_i), y_i)$$

    - A slight change of notation:

$$\frac{1}{n}\sum_{i=1}^{n} cost(h_w(x^{(i)}, y_i)$$



Cost function in closed form

- Reminder: $cost(h_w(x), y) = \begin{cases} -\log(h_w(x)) & \text{if } y = 1 \\ -\log(1 - h_w(x)) & \text{if } y = 0 \end{cases}$
- With a simple transformation, the cost for a single record:

$$cost(h_w(x), y) = -y\log(h_w(x)) - (1 - y)\log(1 - h_w(x))$$

- Total cost:

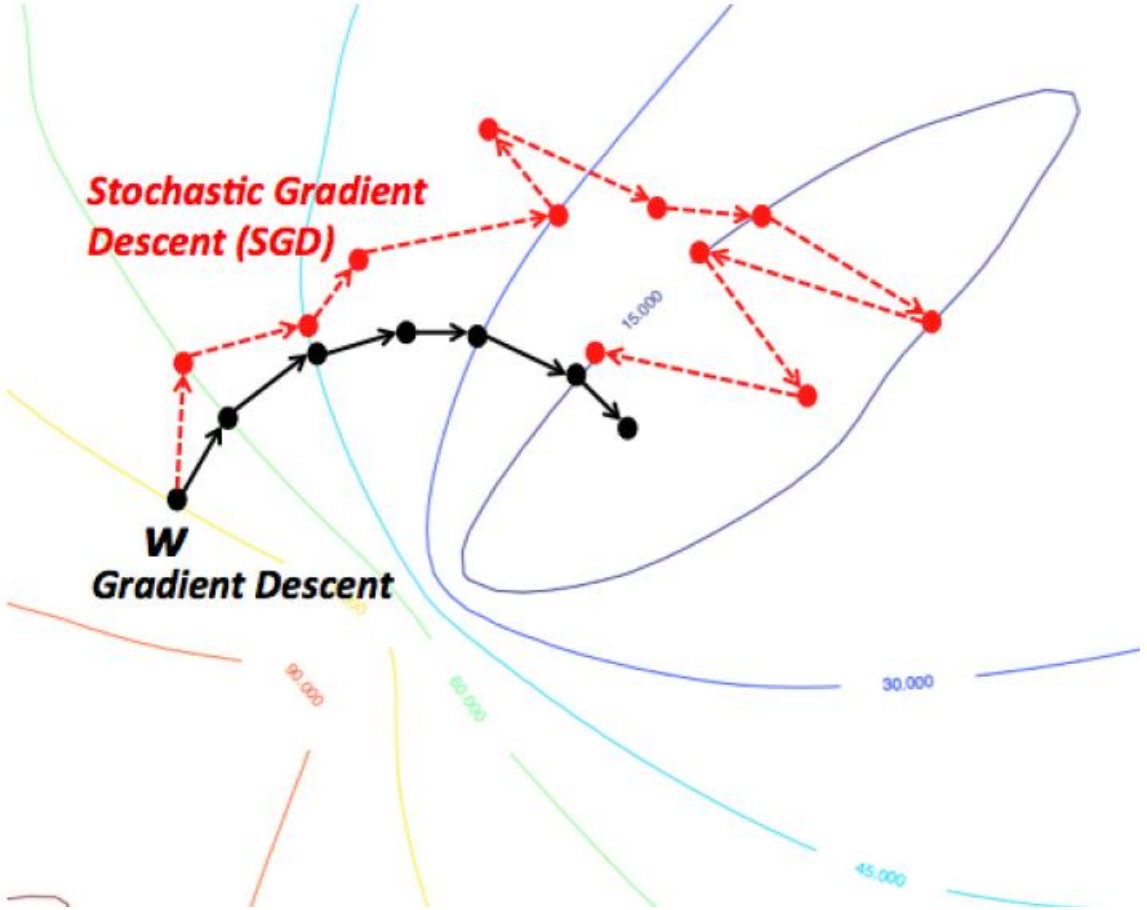
$$C(w) = -\frac{1}{n}\sum_{i=1}^{n}\left(y_i\log(h_w(x^{(i)})) + (1 - y_i)\log(1 - h_w(x^{(i)}))\right)$$

- Similarly to linear regression we can add the regularization term:

$$C(\boldsymbol{w}) + \lambda|\boldsymbol{w}|^2$$

Minimizing the cost function

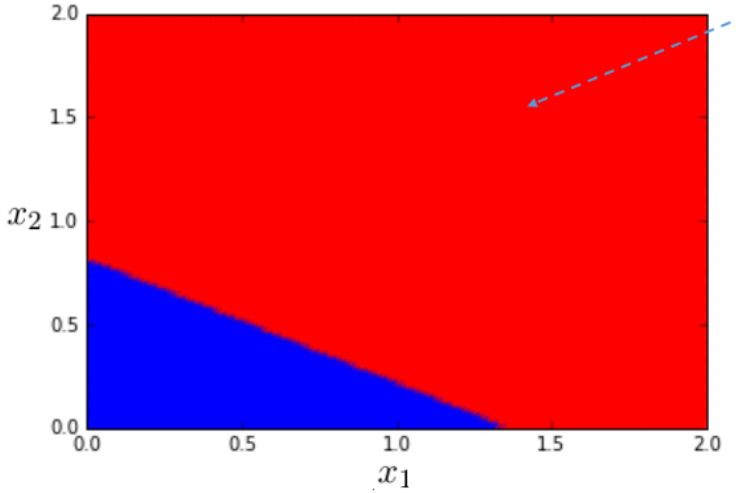
- It is possible to minimize the cost function
  - Analytically (very slow)
  - With gradient descent
  - With stochastic gradient descent
  - With other optimizing methods

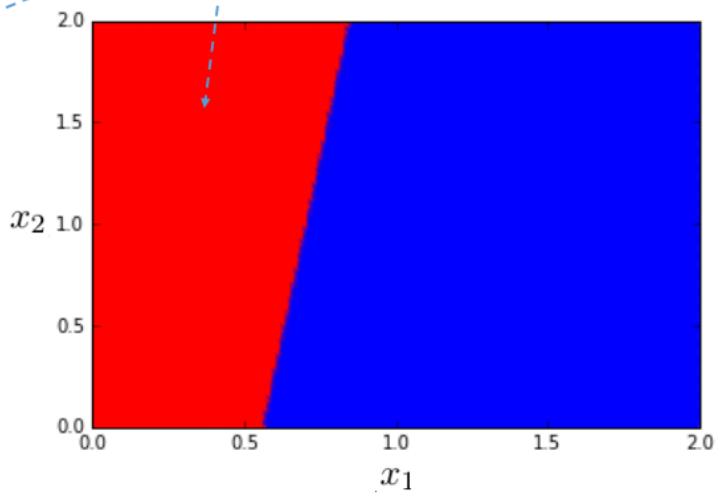


Linear decision boundary

- Having a linear hypothesis ($\boldsymbol{w}^T\ \boldsymbol{x_i}$) the decision boundary of the logistic regression is also linear

$h(x) > 0.5$





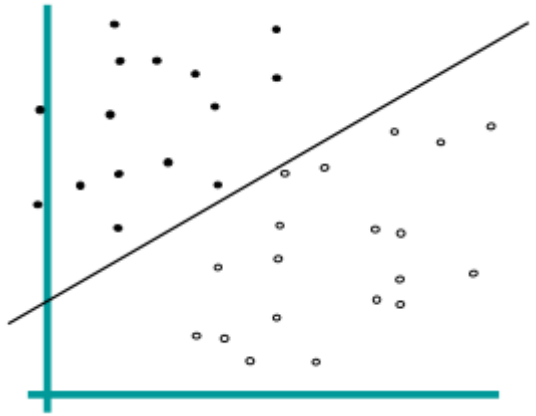
$h(x) = \sigma(-1 + 0.75x_1 + 1.25x_2)$

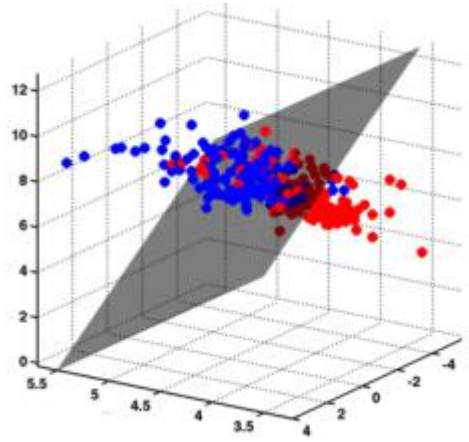
$h(x) = \sigma(1 - 1.75x_1 + 0.25x_2)$

$x = (x_1, x_2)$

Linear separability

- The data is linearly separable if
  - In 2D: there exists a line in the plane with all the positive („blue”) points on one side and all the negative „red” points on the other side
  - In 3D: there exists such a plane
  - In higher dimension: such a hyperplane





- The algorithm is looking for the equation of the separating hyperplane

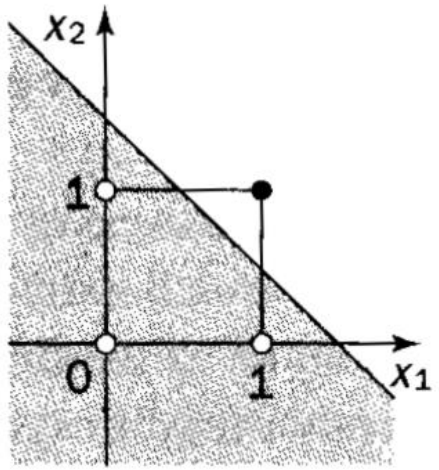
$$\boldsymbol{w}^T\ \boldsymbol{x} = 0$$

$$w_0 + w_1 x_1 + w_2 x_2 + \cdots + w_p x_p = 0$$

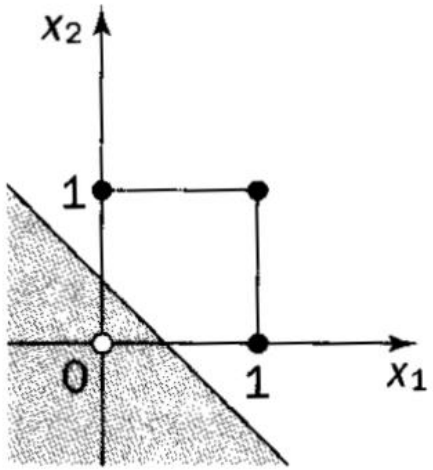
  - One side of the hyperplane (positive records): $\boldsymbol{w}^T\ \boldsymbol{x} > 0$
  - Other side of the hyperplane (negative records): $\boldsymbol{w}^T\ \boldsymbol{x} < 0$
  - Vector $\boldsymbol{w}$ is orthogonal to the hyperplane it gives the direction of the separation

Linear separability of Boolean functions

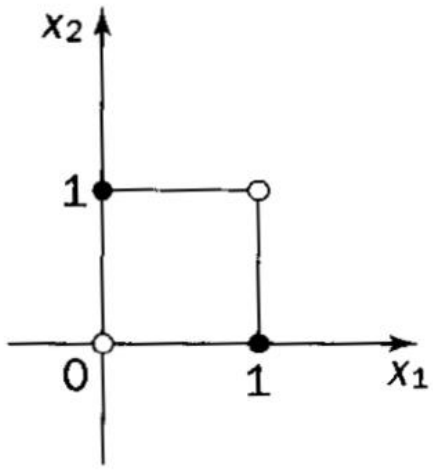
- Some Boolean functions are linearly separable but not all



(a) AND ($x_1 \cap x_2$)



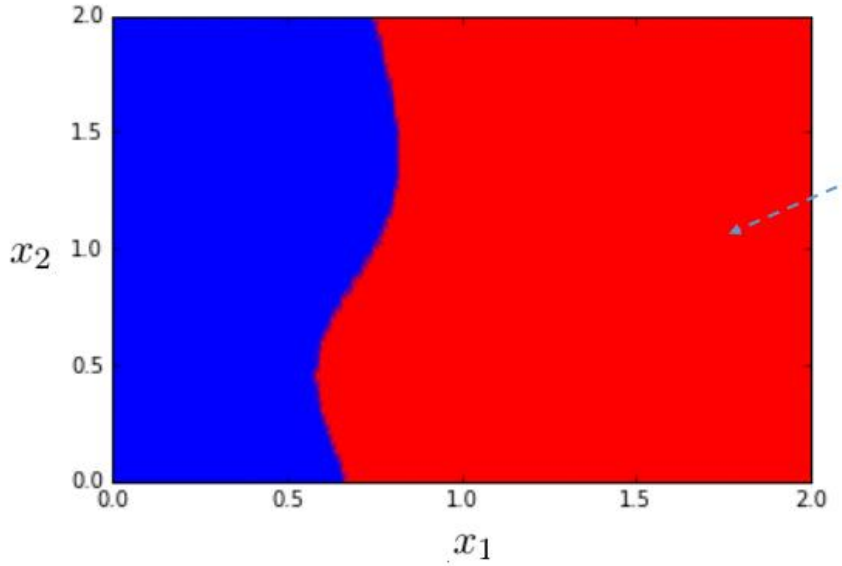
(b) OR ($x_1 \cup x_2$)



(c) Exclusive-OR
($x_1 \oplus x_2$)

Non-linear decision boundary

- If higher powers of the variables are also allowed in the hypothesis than logistic regression can also create non-linear decision boundary



$$h(x) = \sigma(-2 + 0.3x_1 + 0.25x_1^2 + 6x_1^3 + 2x_2 + 0.5x_2^2 - 4x_2^3 + 2x_1^3x_2^5)$$

$$x = (x_1, x_2)$$

Logit model

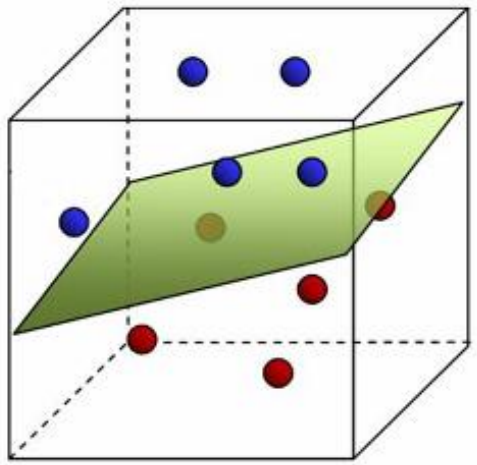
- The inverse function of sigmoid function is the logit function, another way to look at it
- Based on the probabilistic interpretation let *p* be:
- $p = h_{\boldsymbol{w}}(\boldsymbol{x}) = \sigma(\boldsymbol{w}^T\ \boldsymbol{x}) = \frac{1}{1+e^{-w^T x}}$
- After inverting : $\ln\left(\frac{p}{1-p}\right) =$

$$= \boldsymbol{w}^T\boldsymbol{x} = \mathbf{w}_0 + \mathbf{w}_1\mathbf{x}_1 + \cdots + \boldsymbol{w}_p\boldsymbol{x}_p$$

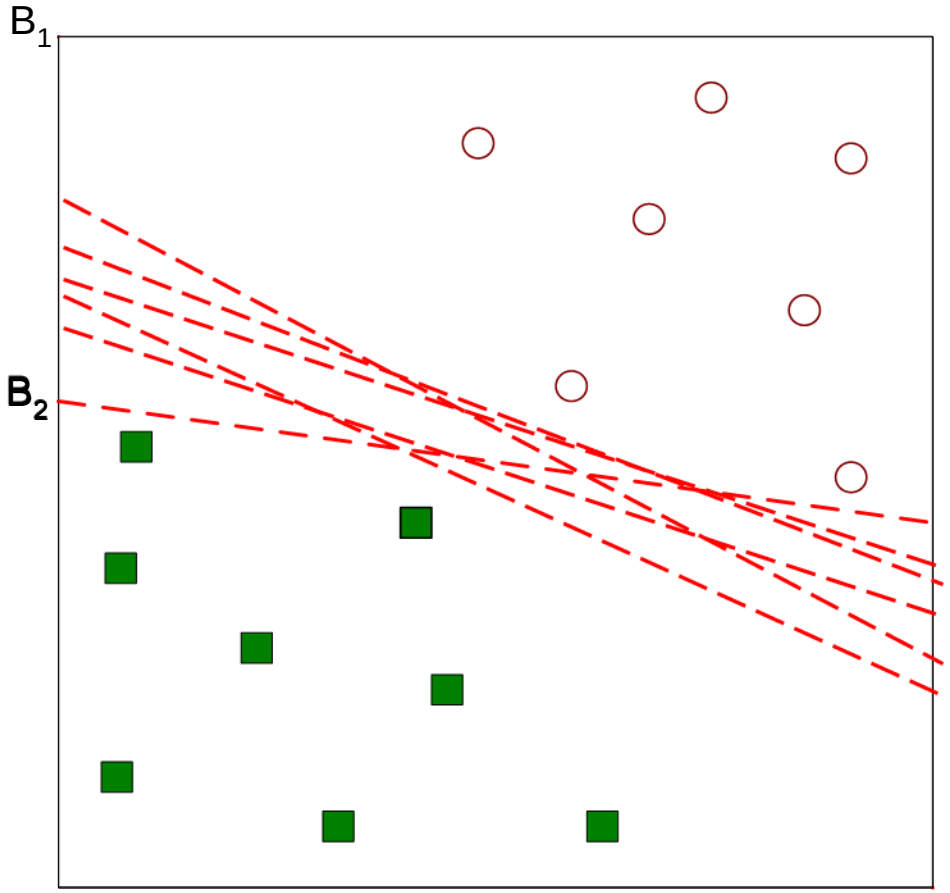
  - This function is the logit function: $\text{logit}(p) = \ln\left(\frac{p}{1-p}\right)$
  - Based on that logistic regression is also called as logit model

Support Vector Machine (SVM)

- Normally SVM is used to classify (linearly separate) binary data
  - It can be extended for multi-class classification, for non-linear separation and for non-linear regression
  - Goal: to find a separating hyperplane that separate the records of the two classes well
    - What do we mean by „well”?



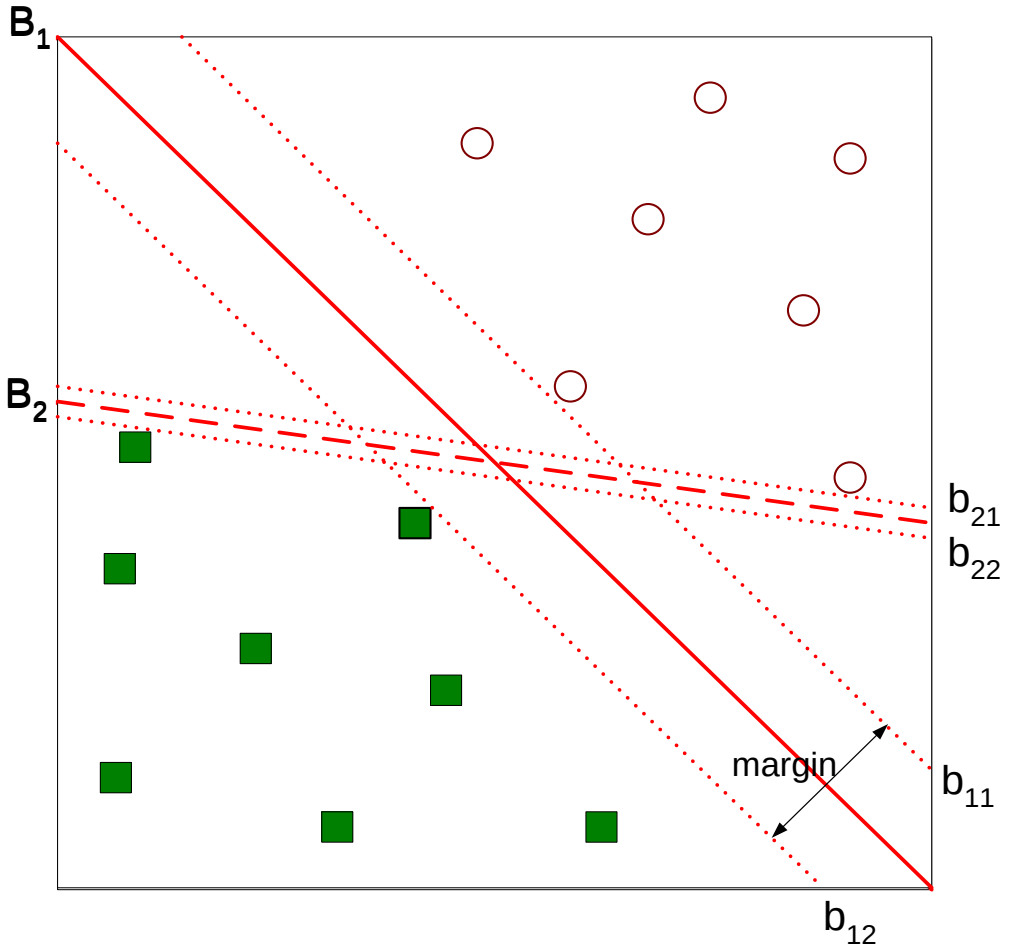
Linear separation, but how?



- If the data is linearly separable there are infinitely many separable line (hyperplane)
- Which to choose?

Which is the better linear separator?

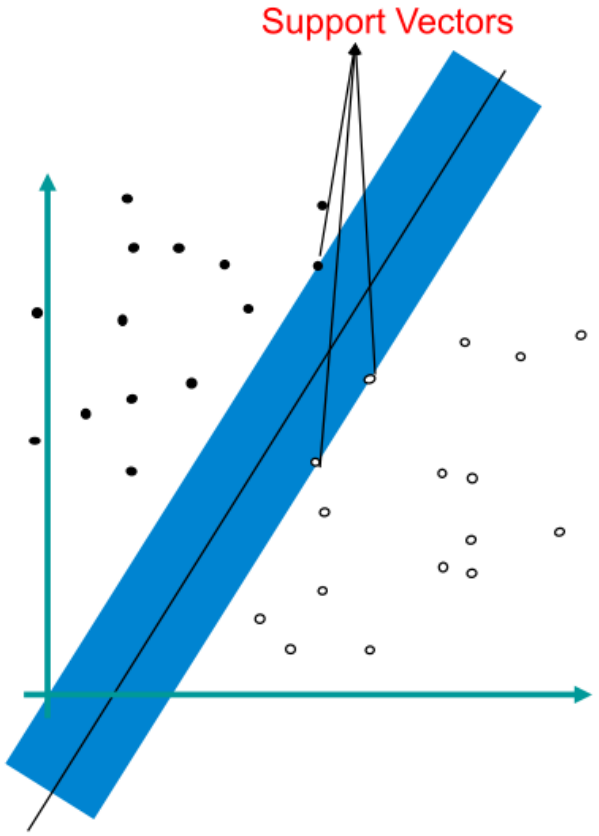
- B_1 or B_2 is the better?
- What do we mean by goodness?
  - The width of margin
  - „Maximizing the margin”
  - So B_1 is better



- Margin: The maximal width of the slice parallel to the separating hyperplane that has no interior data points

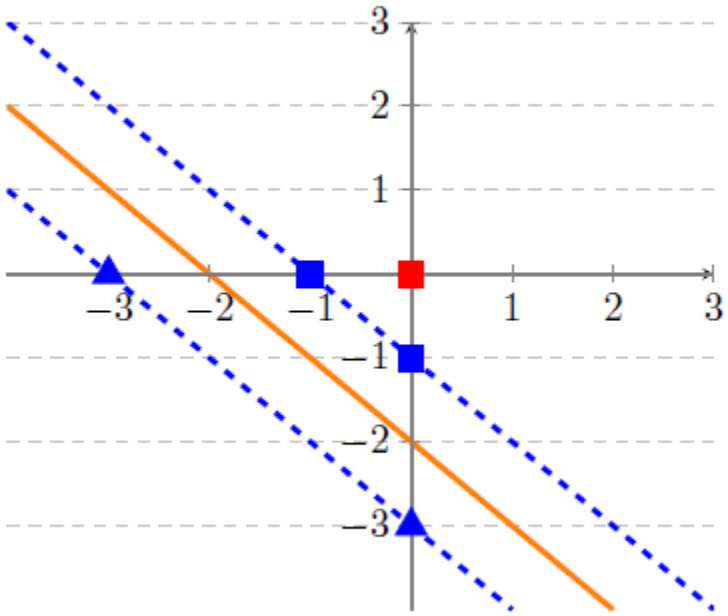
Support vectors

- Support vectors are the data points closest to the separating hyperplane from both classes
- Support vectors are the records of the training set that would change the position of the dividing (separating) hyperplane if removed
  - The „critical” elements



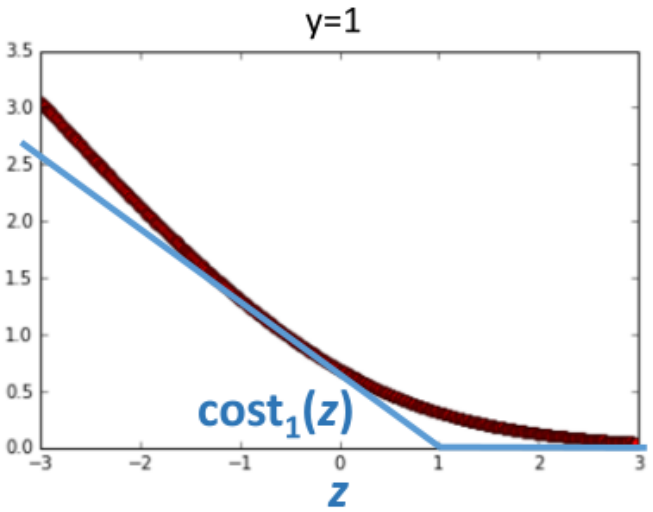
Problem

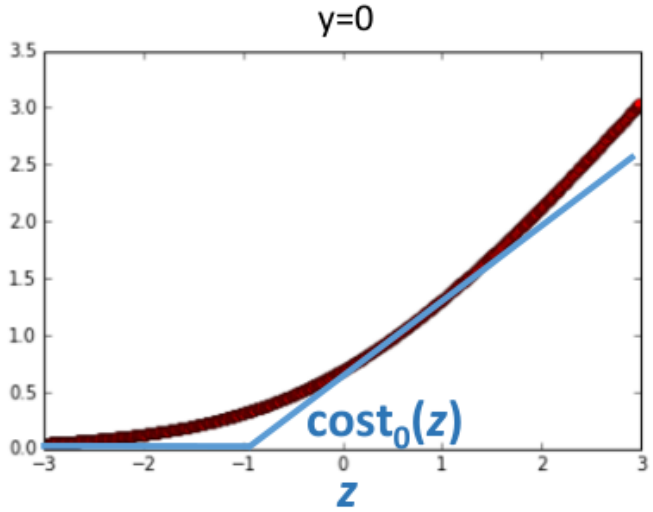
How does the $2 + x + y = 0$ line separates the 2-dimensional plane? Draw the line in a coordinate system. Which of the following records are support vectors of the given line: (–3, 0); (0, –3); (–1, 0); (0, –1); (0, 0)?



Cost function for SVM

- Cost for one single record
  - For logistic regression : $cost(h_w(x), y) = -y\log(h_w(x)) - (1-y)\log(1-h_w(x))$
    where $h_{\boldsymbol{w}}(\boldsymbol{x})$ = σ($\boldsymbol{w}^T\boldsymbol{x}$)
  - For SVM it is modified a bit (see figure)
    - Black curve: cost function for logistic regression
    - Blue curve: cost function for SVM
    - $z = \boldsymbol{w}^T\boldsymbol{x}$





Cost function for SVM II.

- Total cost function for logistic regression:

$$-\frac{1}{n}\sum_{i=1}^{m} y_i \log h_{\underline{w}}\left(\underline{x}^{(i)}\right) + (1-y_i)\log\left(1-h_{\underline{w}}\left(\underline{x}^{(i)}\right)\right) + \lambda\sum_{j=1}^{n} w_j^2$$

- Total cost function for SVM:
  - Here the weights are denoted by **θ** instead of **w**

$$C\sum_{i=1}^{m} y_i \text{cost}_1\left(\underline{\theta}^{\mathsf{T}}\underline{x}^{(i)}\right) + (1-y_i)\text{cost}_0\left(\underline{\theta}^{\mathsf{T}}\underline{x}^{(i)}\right) + \frac{1}{2}\sum_{j=1}^{n}\theta_j^2$$

    - Where C is a parameter to set

Minimizing the cost function

- If in the total cost function $C\sum_{i=1}^{m} y_i \text{cost}_1\left(\underline{\theta}^T \underline{x}^{(i)}\right) + (1-y_i)\text{cost}_0\left(\underline{\theta}^T \underline{x}^{(i)}\right) + \frac{1}{2}\sum_{j=1}^{n}\theta_j^2$

  C is chosen to be big enough then the minimization task is equivalent with the following constrained optimization problem:

  $\frac{1}{2}\sum_{j=1}^{n}\theta_j^2$ --> min

  subject to $\underline{\theta}^T \underline{x}^{(i)} \geq 1, \quad \text{if} \quad y_i = 1,$

  $\underline{\theta}^T \underline{x}^{(i)} \leq -1, \quad \text{if} \quad y_i = 0.$

  - Constrained problems can be solved e.g. using Lagrange multipliers

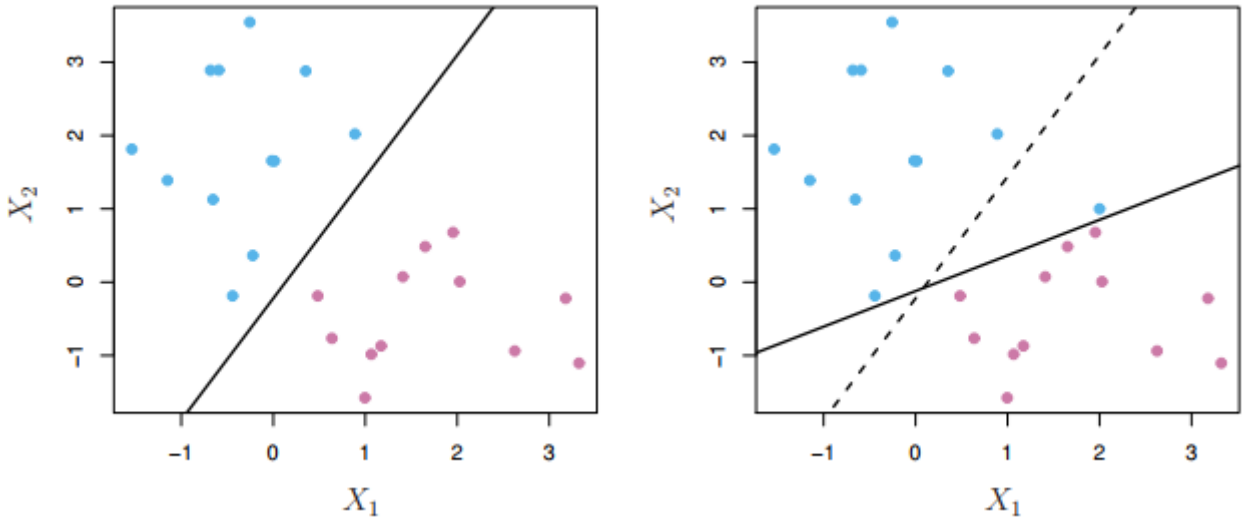
Connection of the minimization problem to the maximal margin

- Solving the previous minimization problem the SVM finds a separating hyperplane that maximizes the margin
- If the data is not linearly separable, then it finds a hyperplane that has a small misclassification error and big margin
  - Which is more important that depends on the constant C
- The equation of the hyperplane: $\boldsymbol{\theta}^T \boldsymbol{x} = 0$

The role of C in the cost function

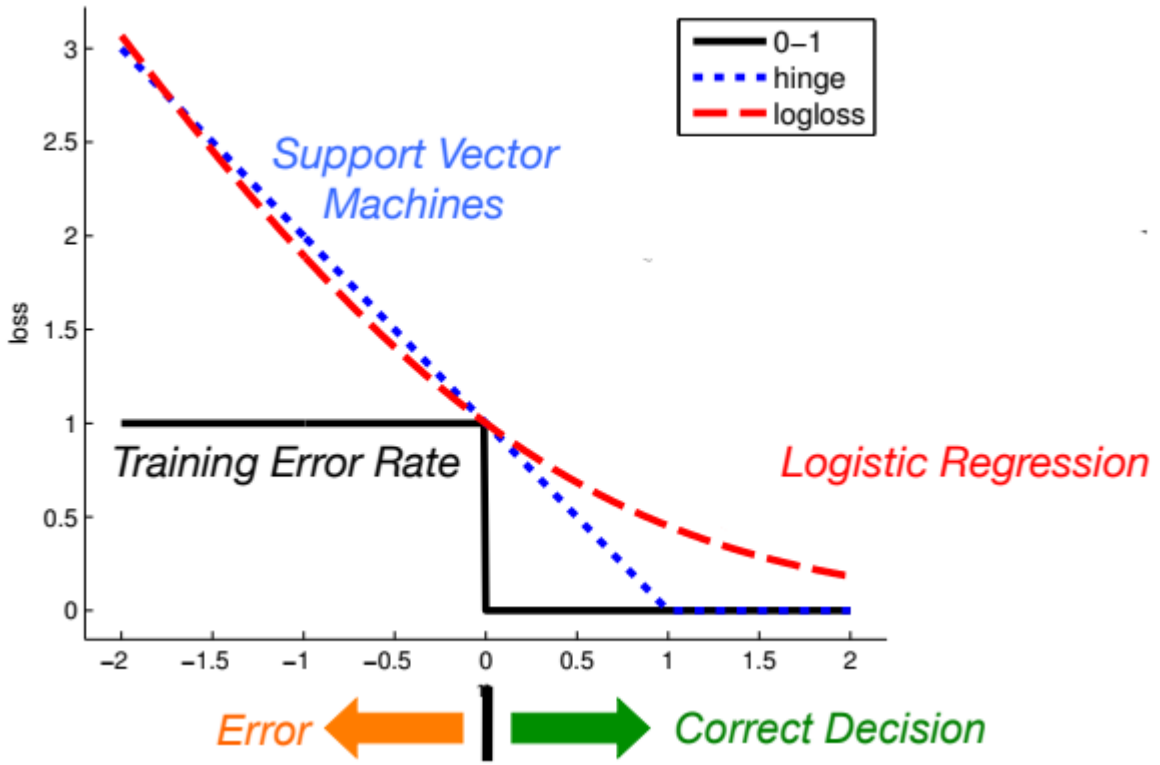
$$C\sum_{i=1}^{m} y_i \text{cost}_1\left(\underline{\theta}^T \underline{x}^{(i)}\right) + (1-y_i)\text{cost}_0\left(\underline{\theta}^T \underline{x}^{(i)}\right) + \frac{1}{2}\sum_{j=1}^{n}\theta_j^2.$$

- If *C* is large than it is more expensive to make misclassification errors so it gives solution with less misclassification error but a smaller margin
  - Risk of overfitting (high variance)
  - Even the addition of one data point can cause big changes in the result
- If *C* is small, it focuses more on finding a hyperplane with big margin



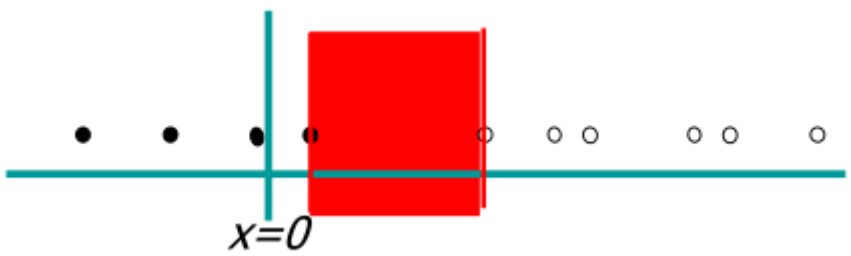
Cost functions for binary classification

- 0-1 loss
  - For some applications that is the important
  - It is difficult to optimize for (not differentiable)
- Logarithmic loss (for logistic regression)
  - Probabilistic interpretation of the output
  - Easy to optimize for (convex)
  - Scalable for big data
- Hinge loss (for SVM)
  - A loss function that maximizes the margin
  - Convex function (usual convex optimizers work)
  - Scalable for big data



Linear separability in 1D

- Are the following data linearly separable?
  - What can be done for the second one? Is it possible to make it linearly separable?

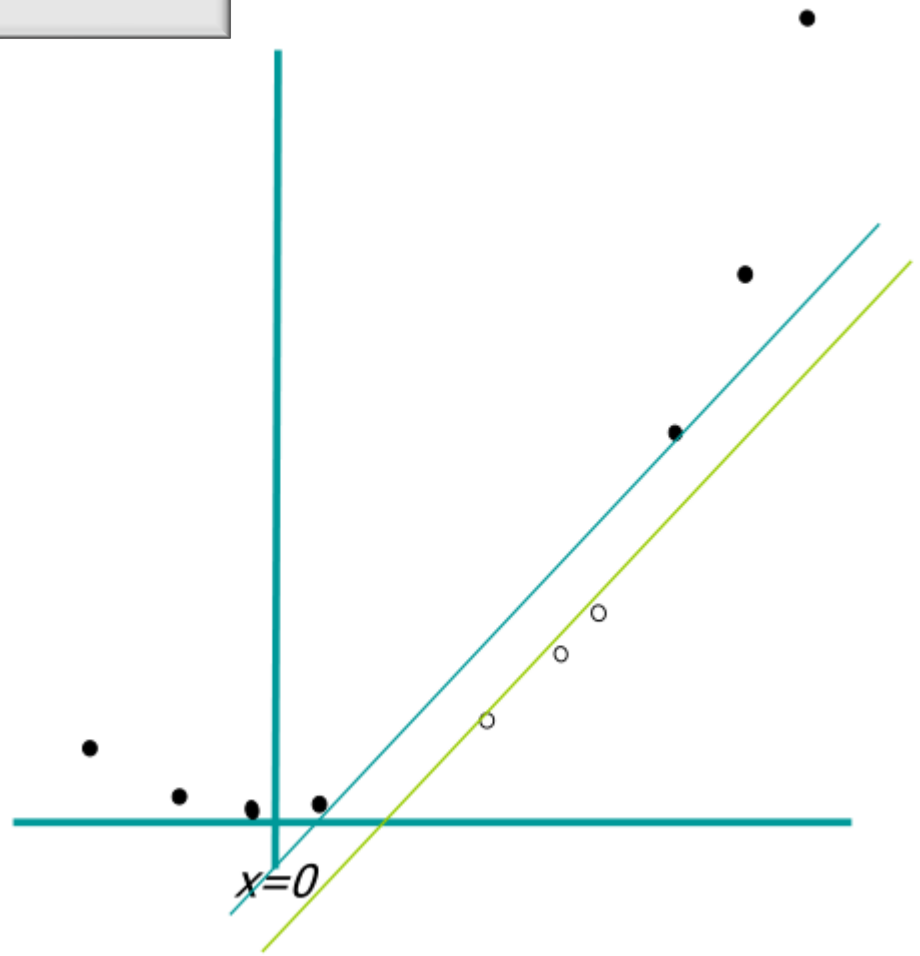


Transformation to higher dimension

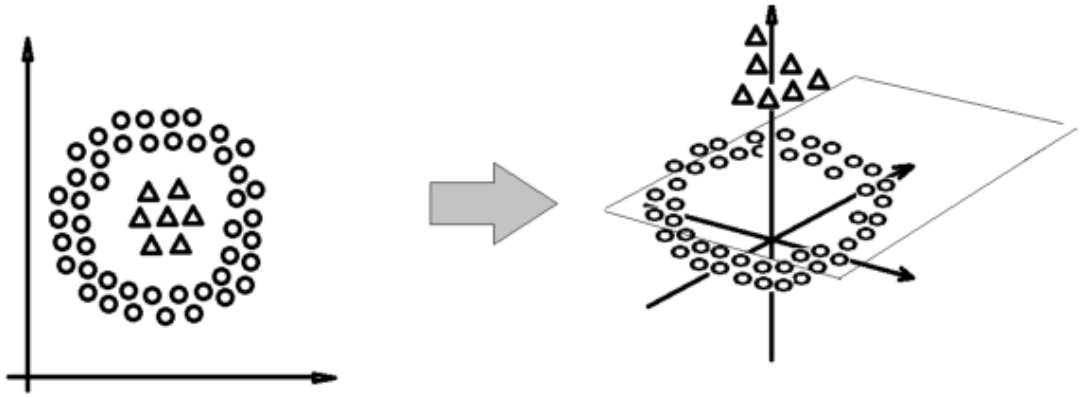
- Apply the following transformation:

$$z_i = (x_i, x_i^2)$$

  - Now it is linearly separable!

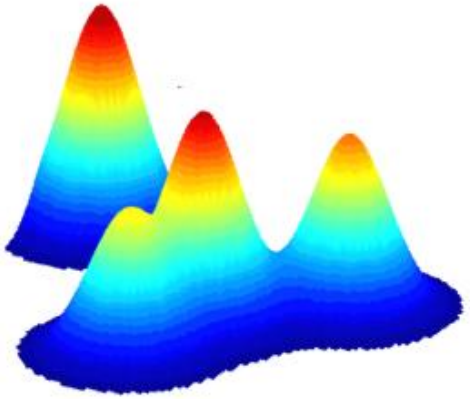


Examples for transformations

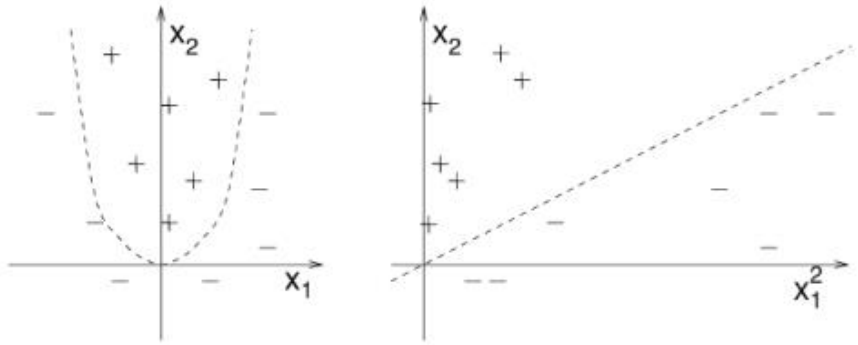


Radial

Radial



Quadratic

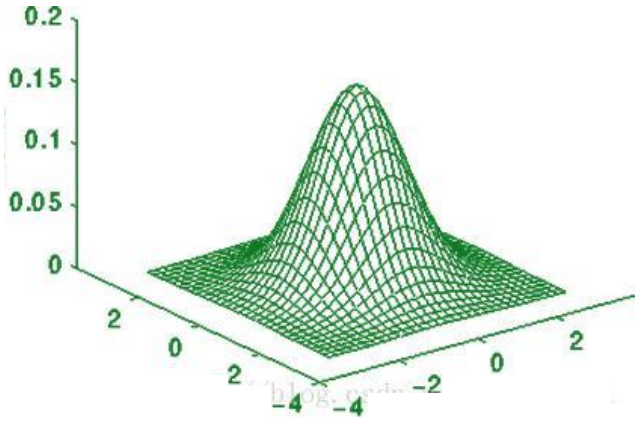


Examples for transformations II.

- Polynomial
  z_i = (some polynomial expressions of x_i)
- Radial
  z_i = (radial functions of x_i)
  - Radial function is a function whose value at each point depends only on the distance between that point and a certain point (e.g the origin or **c** in the example)

$$\mathbf{z}_i = \exp\left(-\frac{|\mathbf{x}_i - \boldsymbol{c}|^2}{\sigma^2}\right)$$

- Sigmoid
  z_i = (sigmoid functions of x_i)



Problem

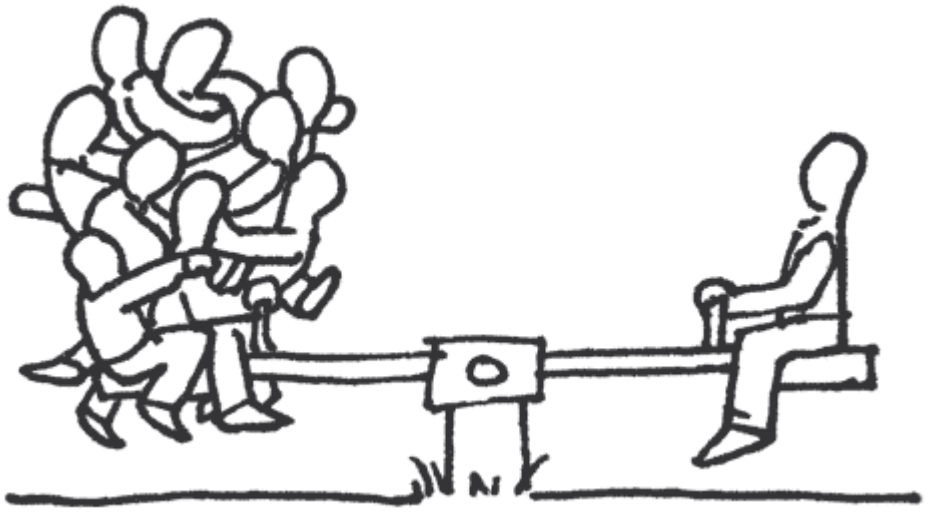
Consider the following data, where the first and the second coordinates are binary attributes, and the third is the class label: (1, 1, –); (1, –1, +); (–1, 1, +); (–1, –1, –). Which Boolean function do you recognize in the data? Is it linearly separable? If so, give the equation of the separating line with maximal margin. If the function is not linearly separable, then transform the data into the following three-dimensional feature space: $(x_1, x_2, x_1 \cdot x_2)$. Furthermore, find the equation of the separating plane with the maximum margin in the transformed space!

Multi-class classification with binary classifiers

- Logistic regression and SVM are binary classifiers
- How can we solve multi-class classification problems with them?
- One-versus-one (OvO) strategy: The problem is reduced to $\binom{c}{2}$ binary classification problems
  - Each receives the samples of a pair of classes from the original training set and must distinguish these two classes
  - At prediction a voting scheme is applied: the class that got the highest number of „votes” (+1 predictions) get predicted by the combined classifier
    - Ambiguity: it is not necessarily unique
  - The result of the combined classifier can be right even if some of the pairwise classifiers were not right
  - Computationally expensive

One-versus-rest (OvR) strategy

- Training a single classifier per class with the samples of that class as positive samples and the samples of all other classes as negatives (creating a virtual class)
- The base classifier produces a real-valued confidence score for its decision
- Making decisions means applying all classifiers to an unseen sample and predicting the label for which the corresponding classifier reports the highest confidence score



Acknowledgement

- András Benczúr, Róbert Pálovics, SZTAKI-AIT, DM1-2
- Krisztián Buza, MTA-BME, VISZJV68
- Bálint Daróczy, SZTAKI-BME, VISZAMA01
- Judit Csima, BME, VISZM185
- Gábor Horváth, Péter Antal, BME, VIMMD294, VIMIA313
- Lukács András, ELTE, MM1C1AB6E
- Tim Kraska, Brown University, CS195
- Dan Potter, Carsten Binnig, Eli Upfal, Brown University, CS1951A
- Erik Sudderth, Brown University, CS142
- Joe Blitzstein, Hanspeter Pfister, Verena Kaynig-Fittkau, Harvard University, CS109
- Rajan Patel, Stanford University, STAT202
- Andrew Ng, John Duchi, Stanford University, CS229