

# Advanced Machine Learning



Bogdan Alexe,

[bogdan.alexe@fmi.unibuc.ro](mailto:bogdan.alexe@fmi.unibuc.ro)

University of Bucharest, 2<sup>nd</sup> semester, 2019-2020

# Grading scheme

A1 – points obtained for the first assignment (3.5points + bonus)

A2 – points obtained for the second assignment (3.5 points + bonus)

No final exam (3 points)

## 2 Formulas:

F1 = scaled A1 + scaled A2 (3 points equally distributed to A1 and A2)

A1 and A2 will worth 5 points instead of 3.5 points

$$F1 = 5/3.5 * A1 + 5/3.5 * A2$$

F2 = A1 + scaled A2 (3 points distributed only to A2)

A1 will worth 3.5 points and A2 will worth 6.5 points

$$F2 = A1 + 6.5/3.5 * A2$$

$$\text{Final grade} = \min(\max(F1, F2), 10)$$

# Recap - SVM

## SVM Definition

- A **Support Vector Machine (SVM)** is a *non-probabilistic binary linear classifier*.
  - **Classifier** – A supervised learning method which predicts a categorical class.
  - **Linear** – The decision boundary is an n-dimensional hyperplane.
  - **Binary** – It can learn to predict one of two classes (a “+” class and a “-” class).
  - **Non-probabilistic** – The output of its *decision function* is not bounded, so it cannot be interpreted as probability.

There are methods of adapting an SVM to be used for **regression** problems and for making it **non-linear**, **multiclass** and/or **probabilistic**.

- An SVM tries to find a *separating* hyperplane, which is as far away from all training points at the same time (a **maximum-margin hyperplane**)
  - A point is classified as “+” or “-”, depending on which part of the hyperplane it lies.

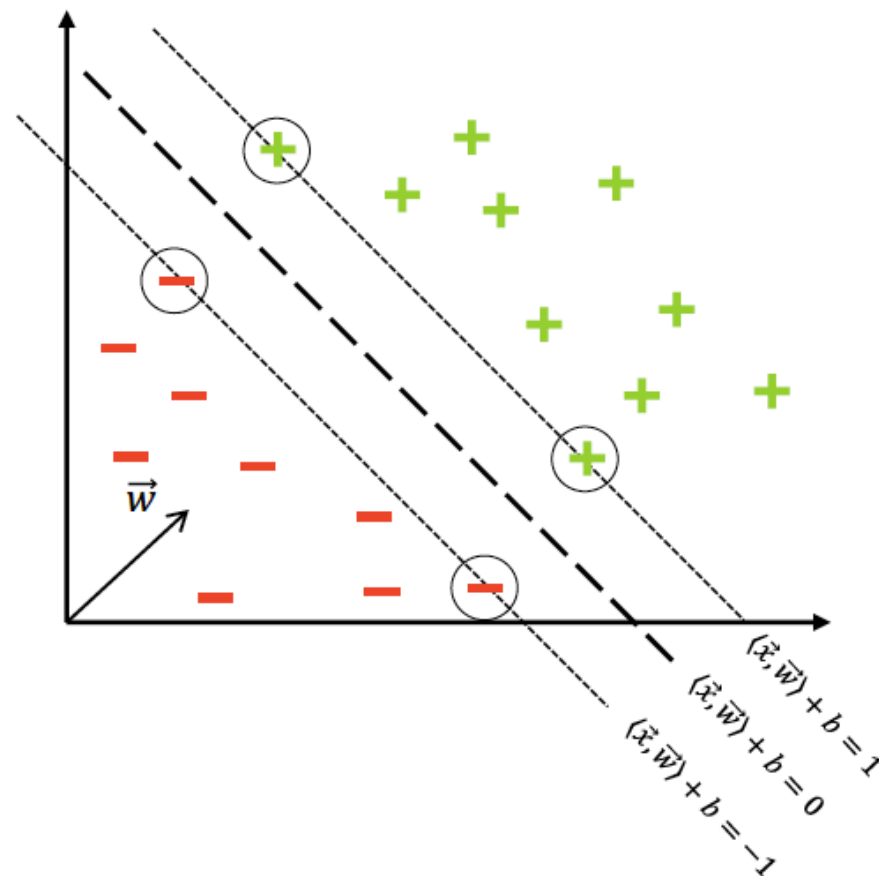
# Recap - SVM

## Learning a “good” separating hyperplane

- For training examples  $(\vec{x}^{(i)}, y^{(i)})$ , which lie exactly on the edges of the gap:

$$y^{(i)}(\langle \vec{x}^{(i)}, \vec{w} \rangle + b) - 1 = 0$$

- We call these examples “**Support Vectors**”



# Recap - SVM

## Making the margin “wide”

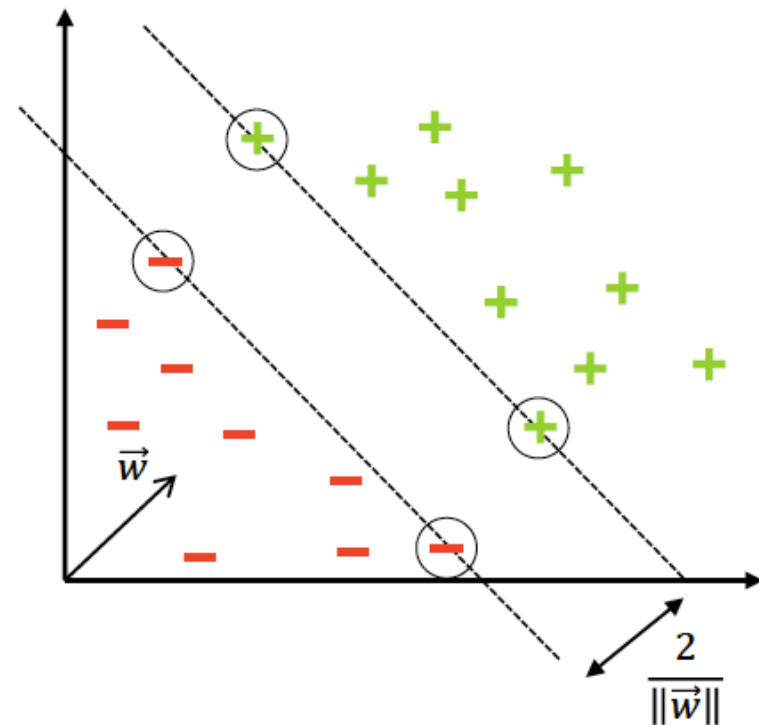
- How do we express the width of the gap?

$$g = \frac{2}{\|\vec{w}\|}$$

- We want to *maximize the gap*:

$$\text{maximize } \frac{2}{\|\vec{w}\|} \Rightarrow \text{minimize } \|\vec{w}\| \Rightarrow$$

$$\text{minimize } \frac{\|\vec{w}\|^2}{2}$$



# Recap - SVM

## What we have so far

### SVM Primal Form

- The decision rule is:

$\vec{x}$  is a “+” sample if  $\langle \vec{x}, \vec{w} \rangle + b \geq 0$

- In order to obtain  $\vec{w}$  and  $b$  we need to:

$$\begin{aligned} & \text{minimize} \quad \frac{\|\vec{w}\|^2}{2} \\ & \text{subject to} \quad y^{(i)} (\langle \vec{x}^{(i)}, \vec{w} \rangle + b) - 1 \geq 0 \end{aligned}$$

# Hard SVM

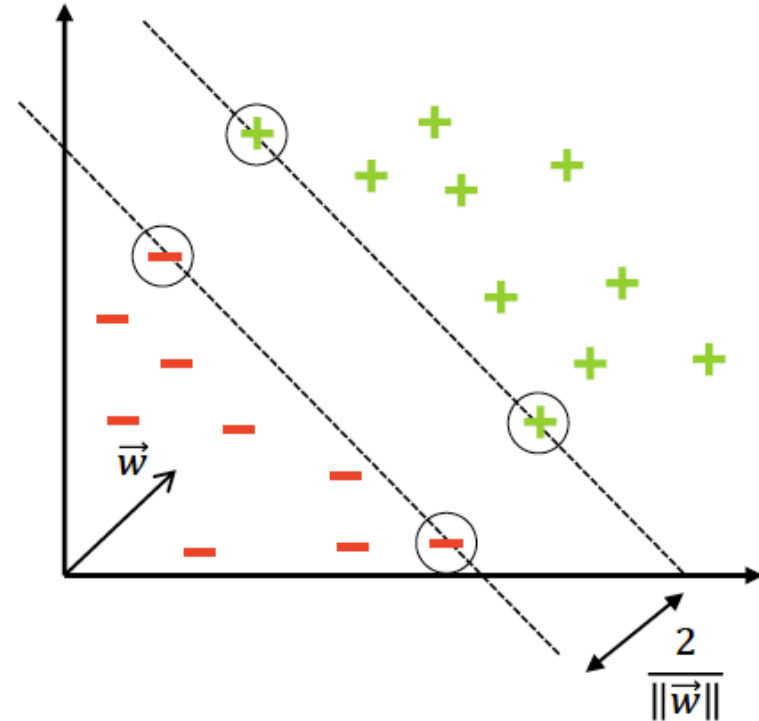
# Hard SVM

**Hard-SVM** is the learning rule in which we return an ERM hyperplane that separates the training set  $S = \{(\mathbf{x}_1, y_1), (\mathbf{x}_1, y_1), \dots, (\mathbf{x}_m, y_m)\}$  with the largest possible margin.

$$\begin{aligned} &\text{minimize } \frac{\|\vec{w}\|^2}{2} \\ &\text{subject to } y^{(i)}(\langle \vec{x}^{(i)}, \vec{w} \rangle + b) - 1 \geq 0 \end{aligned}$$

$\Updownarrow \gamma = \frac{1}{\|\vec{w}\|}$

$$\begin{aligned} &\text{minimize } \frac{1}{2\gamma^2} \\ &\text{subject to } y^{(i)} \left( \left\langle x^{(i)}, \frac{w}{\|w\|} \right\rangle + \frac{b}{\|w\|} \right) - \frac{1}{\|w\|} \geq 0 \end{aligned}$$






# Hard SVM

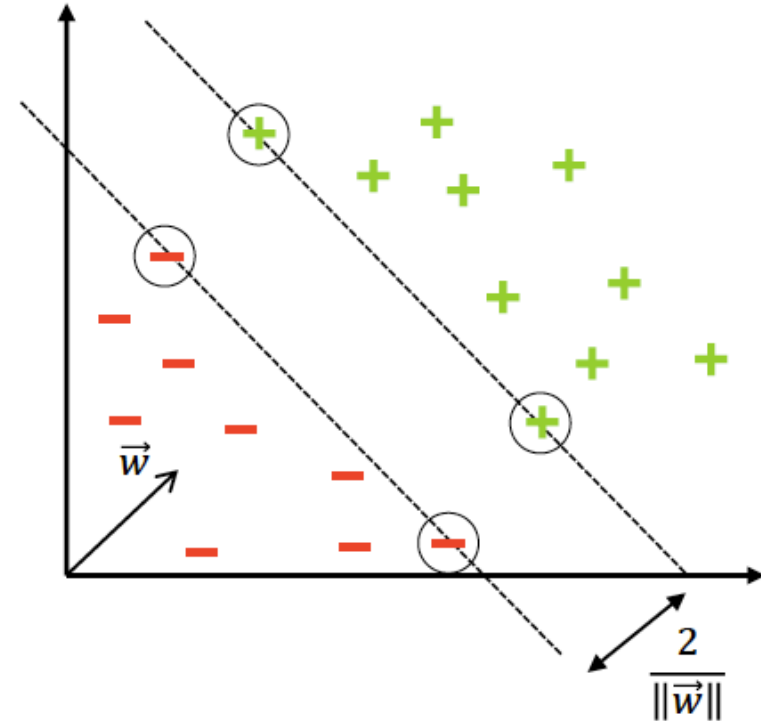
**Hard-SVM** is the learning rule in which we return an ERM hyperplane that separates the training set  $S = \{(x_1, y_1), (x_1, y_1), \dots, (x_m, y_m)\}$  with the largest possible margin.

$$\begin{aligned} & \text{minimize } \frac{\|\vec{w}\|^2}{2} \\ & \text{subject to } y^{(i)}(\langle \vec{x}^{(i)}, \vec{w} \rangle + b) - 1 \geq 0 \end{aligned}$$


 $\gamma = \frac{1}{\|\vec{w}\|}$

**maximize**  $\gamma^2$


**subject to**  $y^{(i)} \left( \left\langle x^{(i)}, \frac{w}{\|w\|} \right\rangle + \frac{b}{\|w\|} \right) \geq \gamma$



# Hard SVM

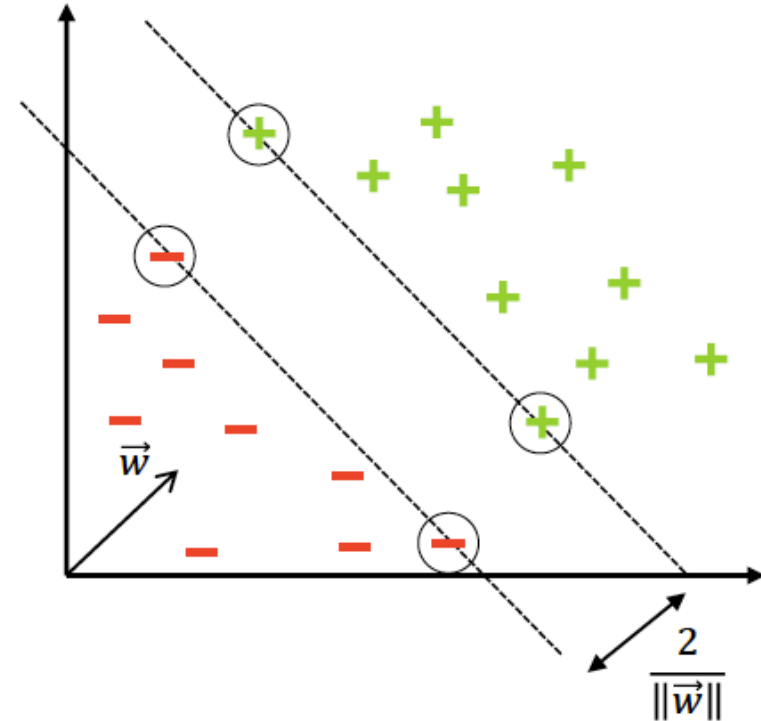
**Hard-SVM** is the learning rule in which we return an ERM hyperplane that separates the training set  $S = \{(x_1, y_1), (x_1, y_1), \dots, (x_m, y_m)\}$  with the largest possible margin.

$$\begin{aligned} &\text{minimize } \frac{\|\vec{w}\|^2}{2} \\ &\text{subject to } y^{(i)}(\langle \vec{x}^{(i)}, \vec{w} \rangle + b) - 1 \geq 0 \end{aligned}$$


 $\gamma = \frac{1}{\|\vec{w}\|}$

**maximize**  $\gamma$

**subject to**  $y^{(i)} \left( \left\langle x^{(i)}, \frac{w}{\|w\|} \right\rangle + \frac{b}{\|w\|} \right) \geq \gamma$



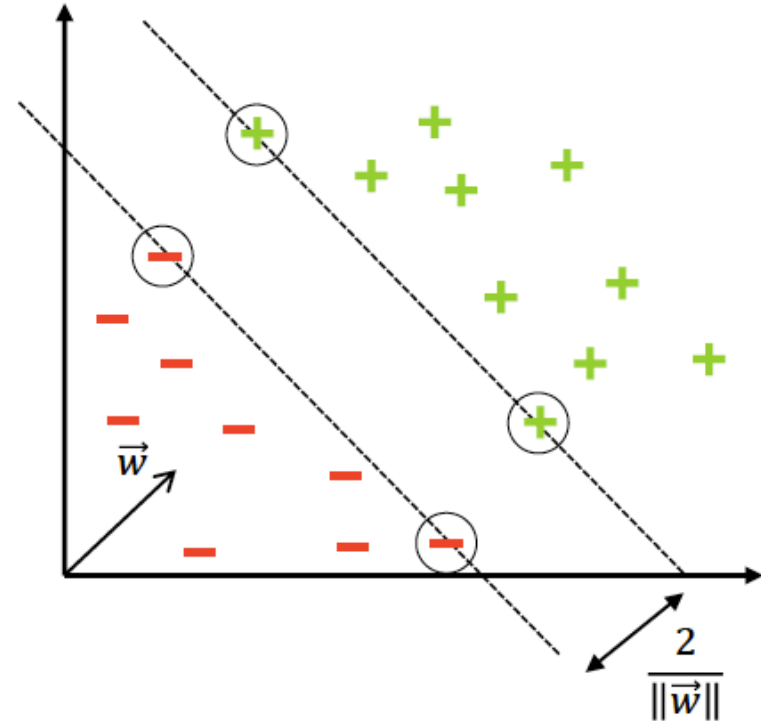
# Hard SVM

**Hard-SVM** is the learning rule in which we return an ERM hyperplane that separates the training set  $S = \{(x_1, y_1), (x_1, y_1), \dots, (x_m, y_m)\}$  with the largest possible margin.

$$\begin{aligned} & \text{minimize } \frac{\|\vec{w}\|^2}{2} \\ & \text{subject to } y^{(i)}(\langle \vec{x}^{(i)}, \vec{w} \rangle + b) - 1 \geq 0 \end{aligned}$$

$\Updownarrow \gamma = \frac{1}{\|\vec{w}\|}$

$$\operatorname{argmax}_{w,b} \min_{i=1,m} y^{(i)} \left( \left\langle x^{(i)}, \frac{w}{\|w\|} \right\rangle + \frac{b}{\|w\|} \right)$$



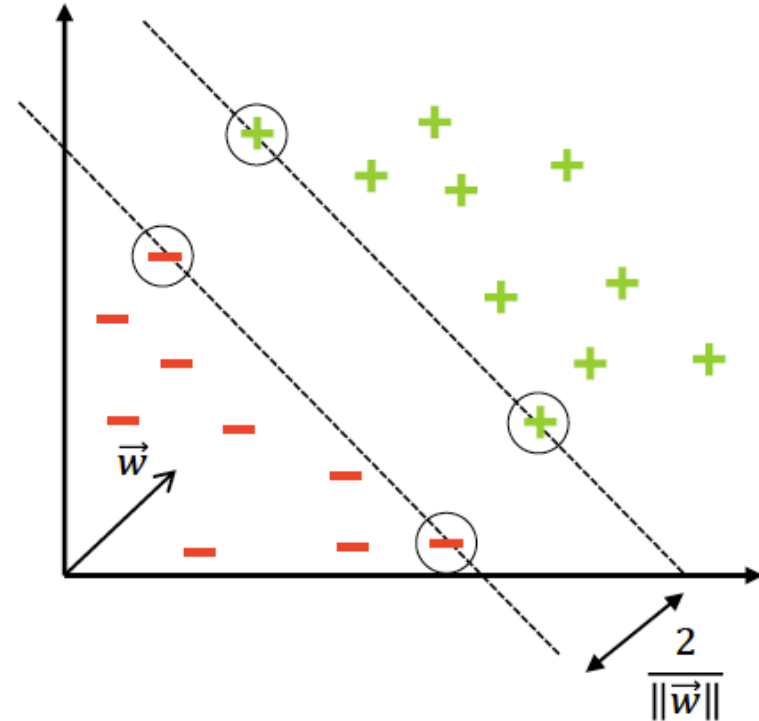
# Hard SVM

**Hard-SVM** is the learning rule in which we return an ERM hyperplane that separates the training set  $S = \{(x_1, y_1), (x_1, y_1), \dots, (x_m, y_m)\}$  with the largest possible margin.

$$\begin{aligned} & \text{minimize } \frac{\|\vec{w}\|^2}{2} \\ & \text{subject to } y^{(i)} (\langle \vec{x}^{(i)}, \vec{w} \rangle + b) - 1 \geq 0 \end{aligned}$$

$\Updownarrow \gamma = \frac{1}{\|\vec{w}\|}$

$$\operatorname{argmax}_{w_0, b_0} \min_{i=1, m} y^{(i)} \left( \left\langle x^{(i)}, \frac{w_0}{\|w_0\|} \right\rangle + \frac{b_0}{\|w_0\|} \right)$$



# Hard SVM

**Hard-SVM** is the learning rule in which we return an ERM hyperplane that separates the training set  $S = \{(x_1, y_1), (x_1, y_1), \dots, (x_m, y_m)\}$  with the largest possible margin.

$$\begin{array}{ll} \text{minimize} & \frac{\|\vec{w}\|^2}{2} \\ \text{subject to} & y^{(i)}(\langle \vec{x}^{(i)}, \vec{w} \rangle + b) - 1 \geq 0 \end{array}$$

$\updownarrow \gamma = \frac{1}{\|\vec{w}\|}$



**Hard-SVM**

**input:**  $(x_1, y_1), \dots, (x_m, y_m)$

**solve:**

$$(w_0, b_0) = \underset{(w, b)}{\operatorname{argmin}} \|w\|^2 \text{ s.t. } \forall i, y_i(\langle w, x_i \rangle + b) \geq 1$$

**output:**  $\hat{w} = \frac{w_0}{\|w_0\|}, \hat{b} = \frac{b_0}{\|w_0\|}$

$$\operatorname{argmax}_{w_0, b_0} \min_{i=1, m} y^{(i)} \left( \left\langle x^{(i)}, \frac{w_0}{\|w_0\|} \right\rangle + \frac{b_0}{\|w_0\|} \right)$$

# Hard SVM

**Hard-SVM** is the learning rule in which we return an ERM hyperplane that separates the training set  $S = \{(x_1, y_1), (x_1, y_1), \dots, (x_m, y_m)\}$  with the largest possible margin.

$$\begin{array}{ll} \text{minimize} & \frac{\|\vec{w}\|^2}{2} \\ \text{subject to} & y^{(i)}(\langle \vec{x}^{(i)}, \vec{w} \rangle + b) - 1 \geq 0 \end{array}$$

$\gamma = \frac{1}{\|w\|}$

**Hard-SVM**

**input:**  $(x_1, y_1), \dots, (x_m, y_m)$

**solve:**

$$(w_0, b_0) = \underset{(w, b)}{\operatorname{argmin}} \|w\|^2 \text{ s.t. } \forall i, y_i(\langle w, x_i \rangle + b) \geq 1$$

**output:**  $\hat{w} = \frac{w_0}{\|w_0\|}, \hat{b} = \frac{b_0}{\|w_0\|}$

$$\operatorname{argmax}_{\|w\|=1, b} \min_{i=1, m} y^{(i)}(\langle x^{(i)}, w \rangle + b)$$

The sample complexity of Hard SVM

# The sample complexity of Hard SVM

Consider  $\mathcal{H} = \mathcal{HS}^d$  be the set of halfspaces (linear classifiers) in  $\mathbf{R}^d$

$$\mathcal{H} = \mathcal{HS}^d = \{h_{w,b}: \mathbf{R}^d \rightarrow \{-1, 1\}, h_{w,b}(x) = \text{sign}\left(\sum_{i=1}^d w_i x_i + b\right) \mid w \in \mathbf{R}^d, b \in \mathbf{R}\}$$

“Homogenous” linear classifiers:  $b = 0$ .

$$\mathcal{HS}_0^d = \{h_{w,0}: \mathbf{R}^d \rightarrow \{-1, 1\}, h_{w,0}(x) = \text{sign}\left(\sum_{i=1}^d w_i x_i\right) \mid w \in \mathbf{R}^d\}$$

$$\text{VCdim}(\mathcal{HS}_0^d) = d \text{ (proof in Lecture 6)}$$

$$\text{VCdim}(\mathcal{HS}^d) = d + 1 \text{ (proof in the book, easy to extend from the one given in the lecture)}$$



# The sample complexity of Hard SVM

$\mathcal{H} = \mathcal{H}S^d$ ,  $\text{VCdim}(\mathcal{H}) = d < \infty$ . From the fundamental theorem of statistical learning we have that there are absolute constants  $C_1, C_2$  such that:

$\mathcal{H}$  is PAC learnable with sample complexity:

$$C_1 \frac{d + \log(1/\delta)}{\epsilon} \leq m_{\mathcal{H}}(\epsilon, \delta) \leq C_2 \frac{d \log(1/\epsilon) + \log(1/\delta)}{\epsilon}$$

The sample complexity of learning halfspaces  $m_{\mathcal{H}}(\epsilon, \delta)$  grows with the dimensionality  $d$  of the problem. Furthermore, the fundamental theorem of learning tells us that if the number of examples is significantly smaller than  $d/\epsilon$  then no algorithm can learn an accurate halfspace. This is problematic when  $d$  is very large.

# Text classification with bag-of-words

- classify a short text document according to its topic, say, whether the document is about sports or not.
- represent documents as vectors.
  - effective way is to use a bag-of-words representation.
  - define a dictionary of words and set the dimension  $d$  to be the number of words in the dictionary.
  - we represent a document as a vector  $\mathbf{x} \in \{0,1\}^d$ , where  $x_i = 1$  if the  $i$ -th word in the dictionary appears in the document and  $x_i = 0$  otherwise.
- a halfspace for the problem of text classification assigns weights to words
- common to have  $d >$  number of training examples. In practice the problem is solvable, we can categorize text based on BOW.

# Text Categorization with Support Vector Machines: Learning with Many Relevant Features

Thorsten Joachims

Universität Dortmund  
Informatik LS8, Baroper Str. 301  
44221 Dortmund, Germany

**Abstract.** This paper explores the use of Support Vector Machines (SVMs) for learning text classifiers from examples. It analyzes the particular properties of learning with text data and identifies why SVMs are appropriate for this task. Empirical results support the theoretical findings. SVMs achieve substantial improvements over the currently best performing methods and behave robustly over a variety of different learning tasks. Furthermore, they are fully automatic, eliminating the need for manual parameter tuning.

[https://www.cs.cornell.edu/people/tj/publications/joachims\\_98a.pdf](https://www.cs.cornell.edu/people/tj/publications/joachims_98a.pdf)

# Sample complexity for separability case

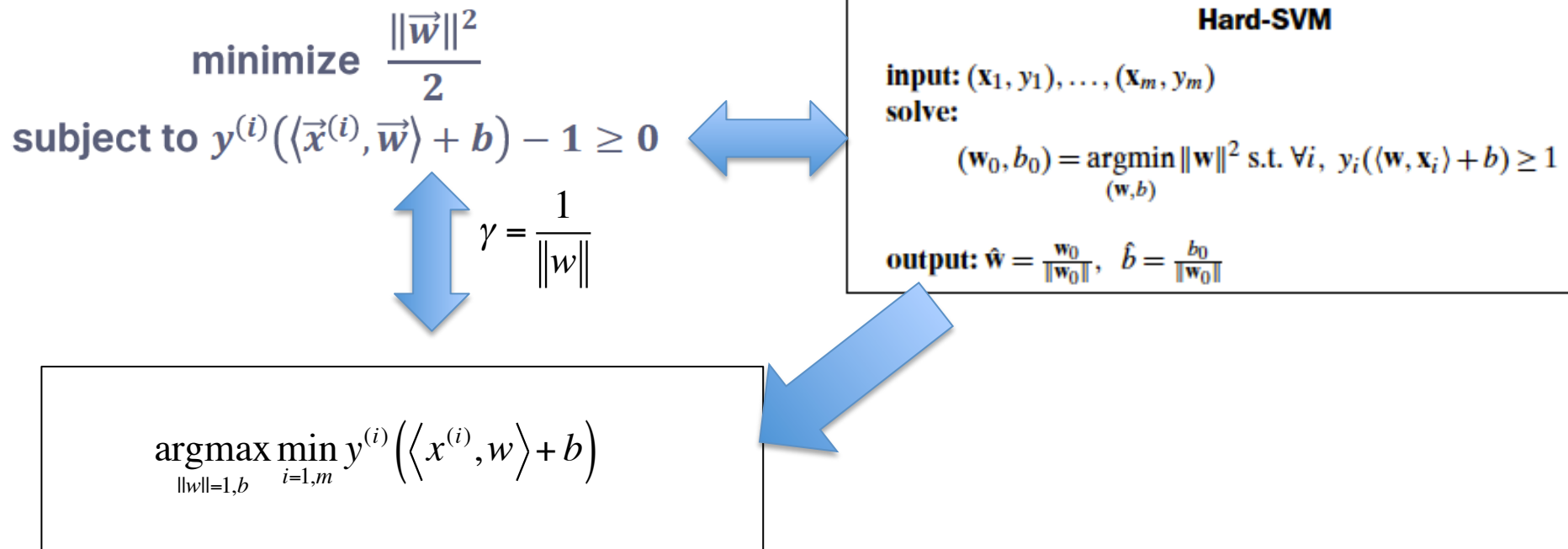
- if the number of examples is significantly smaller than  $d/\varepsilon$  then no algorithm can learn an accurate halfspace.
  - this is problematic when  $d$  is very large.
- make an additional assumption on the underlying data distribution
  - define a “separability with margin  $\gamma$ ” assumption
  - if the data follows this assumption (is separable with margin  $\gamma$ ) then the sample complexity is bounded from above by a function of  $1/\gamma^2$ .
  - even if the dimensionality  $d$  is very large (or even infinite), as long as the data adheres to the separability with margin assumption we can still have a small sample complexity.
  - *the sample complexity will not depend on  $d$ .*
  - there is no contradiction to the lower bound given in the fundamental theorem of learning because we are now making an additional assumption on the underlying data distribution.

# Sample complexity for separability case

If  $S = \{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \dots, (\mathbf{x}_m, y_m)\}$  separable with margin  $\gamma$  then we have that

$S' = \{(\alpha \mathbf{x}_1, y_1), (\alpha \mathbf{x}_2, y_2), \dots, (\alpha \mathbf{x}_m, y_m)\}$  is separable with margin  $\alpha \gamma$  for any  $\alpha > 0$ .

**Definition 15.3.** Let  $\mathcal{D}$  be a distribution over  $\mathbb{R}^d \times \{\pm 1\}$ . We say that  $\mathcal{D}$  is separable with a  $(\gamma, \rho)$ -margin if there exists  $(\mathbf{w}^*, b^*)$  such that  $\|\mathbf{w}^*\| = 1$  and such that with probability 1 over the choice of  $(\mathbf{x}, y) \sim \mathcal{D}$  we have that  $y(\langle \mathbf{w}^*, \mathbf{x} \rangle + b^*) \geq \gamma$  and  $\|\mathbf{x}\| \leq \rho$ . Similarly, we say that  $\mathcal{D}$  is separable with a  $(\gamma, \rho)$ -margin using a homogenous halfspace if the preceding holds with a halfspace of the form  $(\mathbf{w}^*, 0)$ .



# Sample complexity for separability case

If  $S = \{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \dots, (\mathbf{x}_m, y_m)\}$  separable with margin  $\gamma$  then we have that

$S' = \{(\alpha \mathbf{x}_1, y_1), (\alpha \mathbf{x}_2, y_2), \dots, (\alpha \mathbf{x}_m, y_m)\}$  is separable with margin  $\alpha \gamma$  for any  $\alpha > 0$ .

**Definition 15.3.** Let  $\mathcal{D}$  be a distribution over  $\mathbb{R}^d \times \{\pm 1\}$ . We say that  $\mathcal{D}$  is separable with a  $(\gamma, \rho)$ -margin if there exists  $(\mathbf{w}^*, b^*)$  such that  $\|\mathbf{w}^*\| = 1$  and such that with probability 1 over the choice of  $(\mathbf{x}, y) \sim \mathcal{D}$  we have that  $y(\langle \mathbf{w}^*, \mathbf{x} \rangle + b^*) \geq \gamma$  and  $\|\mathbf{x}\| \leq \rho$ . Similarly, we say that  $\mathcal{D}$  is separable with a  $(\gamma, \rho)$ -margin using a homogenous halfspace if the preceding holds with a halfspace of the form  $(\mathbf{w}^*, 0)$ .

It can be shown that the sample complexity of Hard-SVM depends on  $(\rho/\gamma)^2$  and is independent of the dimension  $d$ .

**Theorem 15.4.** Let  $\mathcal{D}$  be a distribution over  $\mathbb{R}^d \times \{\pm 1\}$  that satisfies the  $(\gamma, \rho)$ -separability with margin assumption using a homogenous halfspace. Then, with probability of at least  $1 - \delta$  over the choice of a training set of size  $m$ , the 0-1 error of the output of Hard-SVM is at most

$$\sqrt{\frac{4(\rho/\gamma)^2}{m}} + \sqrt{\frac{2\log(2/\delta)}{m}}.$$

# Soft SVM

## Hard-SVM

**input:**  $(\mathbf{x}_1, y_1), \dots, (\mathbf{x}_m, y_m)$

**solve:**

$$(\mathbf{w}_0, b_0) = \underset{(\mathbf{w}, b)}{\operatorname{argmin}} \|\mathbf{w}\|^2 \text{ s.t. } \forall i, y_i(\langle \mathbf{w}, \mathbf{x}_i \rangle + b) \geq 1 \quad (15.2)$$

**output:**  $\hat{\mathbf{w}} = \frac{\mathbf{w}_0}{\|\mathbf{w}_0\|}, \quad \hat{b} = \frac{b_0}{\|\mathbf{w}_0\|}$

## Soft-SVM

**input:**  $(\mathbf{x}_1, y_1), \dots, (\mathbf{x}_m, y_m)$

**parameter:**  $\lambda > 0$

**solve:**

$$\begin{aligned} \min_{\mathbf{w}, b, \xi} \quad & \left( \lambda \|\mathbf{w}\|^2 + \frac{1}{m} \sum_{i=1}^m \xi_i \right) \\ \text{s.t. } \quad & \forall i, y_i(\langle \mathbf{w}, \mathbf{x}_i \rangle + b) \geq 1 - \xi_i \text{ and } \xi_i \geq 0 \end{aligned} \quad (15.4)$$

**output:**  $\mathbf{w}, b$

The optimization problem in Equation (15.2) enforces the hard constraints  $y_i(\langle \mathbf{w}, \mathbf{x}_i \rangle + b) \geq 1$  for all  $i$ . A natural relaxation is to allow the constraint to be violated for some of the examples in the training set. This can be modeled by introducing nonnegative slack variables,  $\xi_1, \dots, \xi_m$ , and replacing each constraint  $y_i(\langle \mathbf{w}, \mathbf{x}_i \rangle + b) \geq 1$  by the constraint  $y_i(\langle \mathbf{w}, \mathbf{x}_i \rangle + b) \geq 1 - \xi_i$ . That is,  $\xi_i$  measures by how much the constraint  $y_i(\langle \mathbf{w}, \mathbf{x}_i \rangle + b) \geq 1$  is being violated. Soft-SVM jointly minimizes the norm of  $\mathbf{w}$  (corresponding to the margin) and the average of  $\xi_i$  (corresponding to the violations of the constraints). The tradeoff between the two terms is controlled by a parameter  $\lambda$ .

# Soft SVM

## Hard-SVM

**input:**  $(\mathbf{x}_1, y_1), \dots, (\mathbf{x}_m, y_m)$

**solve:**

$$(\mathbf{w}_0, b_0) = \underset{(\mathbf{w}, b)}{\operatorname{argmin}} \|\mathbf{w}\|^2 \text{ s.t. } \forall i, y_i(\langle \mathbf{w}, \mathbf{x}_i \rangle + b) \geq 1 \quad (15.2)$$

**output:**  $\hat{\mathbf{w}} = \frac{\mathbf{w}_0}{\|\mathbf{w}_0\|}, \hat{b} = \frac{b_0}{\|\mathbf{w}_0\|}$

## Soft-SVM

**input:**  $(\mathbf{x}_1, y_1), \dots, (\mathbf{x}_m, y_m)$

**parameter:**  $\lambda > 0$

**solve:**

$$\begin{aligned} \min_{\mathbf{w}, b, \xi} \quad & \left( \lambda \|\mathbf{w}\|^2 + \frac{1}{m} \sum_{i=1}^m \xi_i \right) \\ \text{s.t. } \quad & \forall i, y_i(\langle \mathbf{w}, \mathbf{x}_i \rangle + b) \geq 1 - \xi_i \text{ and } \xi_i \geq 0 \end{aligned} \quad (15.4)$$

**output:**  $\mathbf{w}, b$

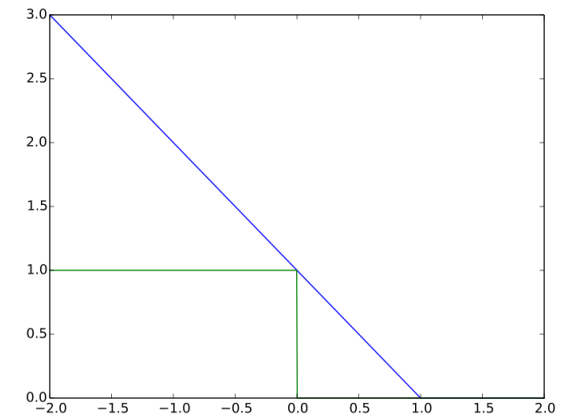
We can rewrite Equation (15.4) as a regularized loss minimization problem. Recall the definition of the hinge loss:

$$\ell^{\text{hinge}}((\mathbf{w}, b), (\mathbf{x}, y)) = \max\{0, 1 - y(\langle \mathbf{w}, \mathbf{x} \rangle + b)\}.$$

Given  $(\mathbf{w}, b)$  and a training set  $S$ , the averaged hinge loss on  $S$  is denoted by  $L_S^{\text{hinge}}((\mathbf{w}, b))$ . Now, consider the regularized loss minimization problem:

$$\min_{\mathbf{w}, b} \left( \lambda \|\mathbf{w}\|^2 + L_S^{\text{hinge}}((\mathbf{w}, b)) \right). \quad (15.5)$$

**CLAIM 15.5** Equation (15.4) and Equation (15.5) are equivalent.



**Hinge loss vs 0-1 loss**



# Soft SVM

We can rewrite Equation (15.4) as a regularized loss minimization problem. Recall the definition of the hinge loss:

$$\ell^{\text{hinge}}((\mathbf{w}, b), (\mathbf{x}, y)) = \max\{0, 1 - y(\langle \mathbf{w}, \mathbf{x} \rangle + b)\}.$$

Given  $(\mathbf{w}, b)$  and a training set  $S$ , the averaged hinge loss on  $S$  is denoted by  $L_S^{\text{hinge}}((\mathbf{w}, b))$ . Now, consider the regularized loss minimization problem:

$$\min_{\mathbf{w}, b} \left( \lambda \|\mathbf{w}\|^2 + L_S^{\text{hinge}}((\mathbf{w}, b)) \right). \quad (15.5)$$

CLAIM 15.5 *Equation (15.4) and Equation (15.5) are equivalent.*

It is often more convenient to consider Soft-SVM for learning a homogenous halfspace, where the bias term  $b$  is set to be zero, which yields the following optimization problem:

$$\min_{\mathbf{w}} \left( \lambda \|\mathbf{w}\|^2 + L_S^{\text{hinge}}(\mathbf{w}) \right), \quad (15.6)$$

where

$$L_S^{\text{hinge}}(\mathbf{w}) = \frac{1}{m} \sum_{i=1}^m \max\{0, 1 - y\langle \mathbf{w}, \mathbf{x}_i \rangle\}.$$

# The sample complexity for Soft-SVM

**COROLLARY 15.7** *Let  $\mathcal{D}$  be a distribution over  $\mathcal{X} \times \{0, 1\}$ , where  $\mathcal{X} = \{\mathbf{x} : \|\mathbf{x}\| \leq \rho\}$ . Consider running Soft-SVM (Equation (15.6)) on a training set  $S \sim \mathcal{D}^m$  and let  $A(S)$  be the solution of Soft-SVM. Then, for every  $\mathbf{u}$ ,*

$$\mathbb{E}_{S \sim \mathcal{D}^m} [L_{\mathcal{D}}^{\text{hinge}}(A(S))] \leq L_{\mathcal{D}}^{\text{hinge}}(\mathbf{u}) + \lambda \|\mathbf{u}\|^2 + \frac{2\rho^2}{\lambda m}.$$

*Furthermore, since the hinge loss upper bounds the 0–1 loss we also have*

$$\mathbb{E}_{S \sim \mathcal{D}^m} [L_{\mathcal{D}}^{0-1}(A(S))] \leq L_{\mathcal{D}}^{\text{hinge}}(\mathbf{u}) + \lambda \|\mathbf{u}\|^2 + \frac{2\rho^2}{\lambda m}.$$

*Last, for every  $B > 0$ , if we set  $\lambda = \sqrt{\frac{2\rho^2}{B^2 m}}$  then*

$$\mathbb{E}_{S \sim \mathcal{D}^m} [L_{\mathcal{D}}^{0-1}(A(S))] \leq \mathbb{E}_{S \sim \mathcal{D}^m} [L_{\mathcal{D}}^{\text{hinge}}(A(S))] \leq \min_{\mathbf{w}: \|\mathbf{w}\| \leq B} L_{\mathcal{D}}^{\text{hinge}}(\mathbf{w}) + \sqrt{\frac{8\rho^2 B^2}{m}}.$$

We therefore see that we can control the sample complexity of learning a half-space as a function of the norm of that halfspace, independently of the Euclidean dimension of the space over which the halfspace is defined. This becomes highly significant when we learn via embeddings into high dimensional feature spaces, as we will consider in the next chapter.

# Margin and norm based bounds vs dimension

Remember from lecture 7 (use of the Sauer lemma): for every  $\mathcal{D}$  and every  $\delta \in (0,1)$ , with probability of at least  $1 - \delta$  over the choice of  $S \sim \mathcal{D}^m$  we have:

$$|L_D(h) - L_S(h)| \leq \frac{4 + \sqrt{\log(\tau_H(2m))}}{\delta\sqrt{2m}} \leq \frac{4 + \sqrt{d \log(2em/d)}}{\delta\sqrt{2m}} \leq \frac{2\sqrt{d \log(2em/d)}}{\delta\sqrt{2m}} \quad \begin{array}{l} d = \text{VCdim}(\mathcal{H}), \\ \mathcal{H} \text{ is the class of} \\ \text{halfspaces} \end{array}$$

*probability of at least  $1 - \delta$  over the choice of a training set of size  $m$ , the 0-1 error of the output of Hard-SVM is at most*

Loss for Hard-SVM

$$\sqrt{\frac{4(\rho/\gamma)^2}{m}} + \sqrt{\frac{2\log(2/\delta)}{m}}.$$

$$\mathbb{E}_{S \sim \mathcal{D}^m} [L_D^{0-1}(A(S))] \leq \mathbb{E}_{S \sim \mathcal{D}^m} [L_D^{\text{hinge}}(A(S))] \leq \min_{\mathbf{w}: \|\mathbf{w}\| \leq B} L_D^{\text{hinge}}(\mathbf{w}) + \sqrt{\frac{8\rho^2 B^2}{m}}.$$

Loss for Soft-SVM

The bounds we have derived for Hard-SVM and Soft-SVM do not depend on the dimension of the instance space. Instead, the bounds depend on the norm of the examples,  $\rho$ , the norm of the halfspace  $B$  (or equivalently the margin parameter  $\gamma$ ) and, in the nonseparable case, the bounds also depend on the minimum hinge loss of all halfspaces of norm  $\leq B$ . In contrast, the VC-dimension of the class of homogenous halfspaces is  $d$ , which implies that the error of an ERM hypothesis decreases as  $\sqrt{d/m}$  does. We now give an example in which  $\rho^2 B^2 \ll d$ ; hence the bound given in Corollary 15.7 is much better than the VC bound.

# Margin and norm based bounds vs dimension

Consider the problem of learning to classify a short text document according to its topic, say, whether the document is about sports or not. We first need to represent documents as vectors. One simple yet effective way is to use a *bag-of-words* representation. That is, we define a dictionary of words and set the dimension  $d$  to be the number of words in the dictionary. Given a document, we represent it as a vector  $\mathbf{x} \in \{0,1\}^d$ , where  $x_i = 1$  if the  $i$ 'th word in the dictionary appears in the document and  $x_i = 0$  otherwise. Therefore, for this problem, the value of  $\rho^2$  will be the maximal number of distinct words in a given document.

A halfspace for this problem assigns weights to words. It is natural to assume that by assigning positive and negative weights to a few dozen words we will be able to determine whether a given document is about sports or not with reasonable accuracy. Therefore, for this problem, the value of  $B^2$  can be set to be less than 100. Overall, it is reasonable to say that the value of  $B^2\rho^2$  is smaller than 10,000.

On the other hand, a typical size of a dictionary is much larger than 10,000. For example, there are more than 100,000 distinct words in English. We have therefore shown a problem in which there can be an order of magnitude difference between learning a halfspace with the SVM rule and learning a halfspace using the vanilla ERM rule.

# Assignment 2

# Assignment 2 – good to know

- 3 problems = 3.5 points
- 1 bonus problem = 1 point
- deadline: in 3 weeks time, Sunday, 21<sup>st</sup> of June 2020, 23:59
  - late submission policy: maximum 3 days allowed, -10% (= 0.35 points) for each day
  - upload a pdf written in a scientific editor (Word, Latex, LyX) containing your solution here: <https://tinyurl.com/AML-2020-ASSIGNMENT2>
  - *is mandatory that you write your solution with a scientific editor, otherwise your solution would not be taken into account*
  - you can insert drawings for your proofs
- for every problem write clear explanations, proofs to justify your answer (if you write just some indications you will not get too many points)
- do not share/copy the solution with/from your colleagues: you + your colleague/s will get 0 points

# Problem 1

## Assignment 2

*Deadline: Sunday, 21<sup>st</sup> of June, 23:59.*

*Upload your solutions as a zip archive at: <https://tinyurl.com/AML-2020-ASSIGNMENT2>*

1. **(1 point)** Consider  $\mathcal{H} = \{h_{\theta_1}: \mathbb{R} \rightarrow \{0,1\}, h_{\theta_1}(x) = \mathbf{1}_{[x \geq \theta_1]}(x) = \mathbf{1}_{[\theta_1, +\infty)}(x), \theta_1 \in \mathbb{R}\} \cup \{h_{\theta_2}(x) = \mathbf{1}_{[x < \theta_2]}(x) = \mathbf{1}_{(-\infty, \theta_2)}(x), \theta_2 \in \mathbb{R}\}$ .
  - a. Compute the shattering coefficient  $\tau_H(m)$  of the growth function for  $m \geq 0$ . **(0.5 points)**
  - b. Compare your result with the general upper bound for the growth functions. **(0.25 points)**
  - c. Does there exist a hypothesis class  $\mathcal{H}$  for which  $\tau_H(m)$  is equal to the general upper bound (over  $\mathbb{R}$  or another domain  $\mathcal{X}$ )? If your answer is yes please provide an example, if your answer is no please provide a justification. **(0.25 points)**



# Problem 2

2. **(1.25 points)** Let  $\Sigma$  be a finite alphabet and let  $\mathcal{X} = \Sigma^m$  be a sample space of all strings of length  $m$  over  $\Sigma$ . Let  $\mathcal{H}$  be a hypothesis space over  $\mathcal{X}$ , where  $\mathcal{H} = \{h_w: \Sigma^m \rightarrow \{0,1\}, w \in \Sigma^*, 0 < |w| \leq m, \text{ s.t. } h_w(x) = 1 \text{ if } w \text{ is a substring of } x\}$ .
- a. Give an upper bound (any upper bound that you can come up) of the VC-dimension of  $\mathcal{H}$  in terms of  $|\Sigma|$  and  $m$ . **(0.25 points)**
  - b. Give an efficient algorithm for finding a hypothesis  $h_w$  consistent with a training set in the realizable case. What is the complexity of your algorithm? **(1 point)**

*Example:* let  $\Sigma = \{a, b, c\}$ ,  $m = 4$  and the training set  $S = \{(aabc, 1), (baca, 0), (bcac, 0), (abba, 1)\}$ .

The output of the algorithm should be  $h_{ab}$ .



# Problem 3

3. **(1.25 points)** Consider the boosting algorithm described (page 4) in the article “[Rapid object detection using a boosted cascade of simple features](#)”, P. Viola and M. Jones, CVPR 2001. Consider that the number of positives is equal with the number of negative examples ( $l = m$ ).
- a. Prove that the distribution  $w_{t+1}$  obtained at round  $t + 1$  based on the algorithm described in the article is the same with the distribution  $D^{(t+1)}$  obtained based on the procedure described in lecture 11 (slides 10-12). **(0.25 points)**
  - b. Prove that the two final classifiers (the one described in the article and the one described in the lecture) are equivalent. **(0.50 points)**
  - c. Assume that at each iteration  $t$  of AdaBoost, the weak learner returns a hypothesis  $h_t$  for which the error  $\varepsilon_t$  satisfies  $\varepsilon_t \leq 1/2 - \gamma$ ,  $\gamma > 0$ . What is the probability that the classifier  $h_t$  (selected as the best weak learner at iteration  $t$ ) will be selected again at iteration  $t+1$ ? Justify your answer. **(0.50 points)**

# Bonus Problem

## Bonus Problem (1 point)

Consider  $H_{2DNF}^d$  the class of 2-term disjunctive normal form formulae consisting of hypothesis of the form  $h: \{0,1\}^d \rightarrow \{0,1\}$ ,

$$h(\mathbf{x}) = A_1(\mathbf{x}) \vee A_2(\mathbf{x}),$$

where  $A_i(\mathbf{x})$  is a Boolean conjunction of literals (in  $H_{conj}^d$ ).

It is known that the class  $H_{2DNF}^d$  is not efficiently properly learnable but can be learned improperly considering the class  $H_{2CNF}^d$ .

Give a  $\gamma$ -weak-learner algorithm for learning the class  $H_{2DNF}^d$  which is not a stronger PAC learning algorithm for  $H_{2DNF}^d$  (like the one considering  $H_{2CNF}^d$ ). Prove that this algorithm is a  $\gamma$ -weak-learner algorithm for  $H_{2DNF}^d$ .

*Hint: Find an algorithm that returns  $h(x)=0$  or the disjunction of 2 literals.*