Introduction to Statistical Learning Theory, Kernel Methods, and Support Vector Machines (SVM)

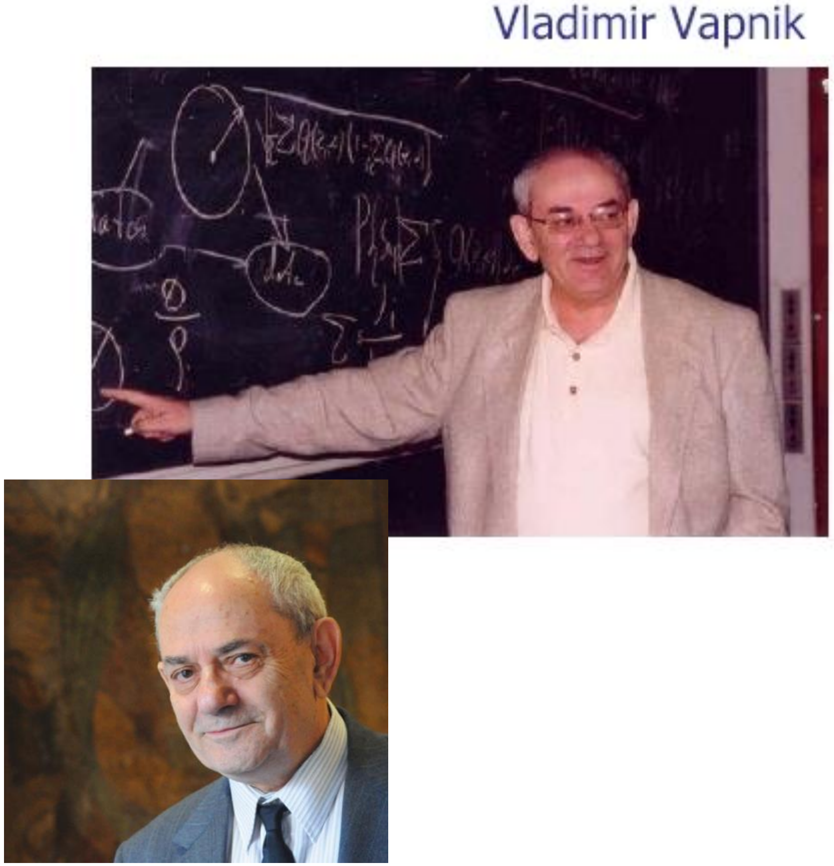
By: **Kaveh Kavousi**
Department of Bioinformatics
IBB (Institute of Biochemistry and Biophysics)
University of Tehran

Introduction

- SVMs provide a learning technique for
  - Pattern Recognition
  - Regression Estimation
- Solution provided SVM is
  - Theoretically elegant
  - Computationally Efficient
  - Very effective in many Large practical problems
- It has a simple geometrical interpretation in a high-dimensional feature space that is nonlinearly related to input space
- By using kernels all computations keep simple.
- It contains ANN, RBF and Polynomial classifiers as special cases.

History of SVM

- The Study on Statistical Learning Theory was started in the 1960s by Vapnik.
- Statistical Learning Theory is the theory about Machine Learning Principle from a small sample size.
- Support Vector Machine is a practical learning method based on Statistical Learning Theory
- A simple SVM could beat a sophisticated neural networks with elaborate features in a handwriting recognition task.



He was born in the Soviet Union. He received his master's degree in mathematics at the Uzbek State University, Samarkand, Uzbek SSR in 1958 and Ph.D in statistics at the Institute of Control Sciences, Moscow in 1964.

Learning through empirical risk minimization

Classification Engine

$$f : \mathbb{R}^N \rightarrow \{\pm 1\}$$

Training Dataset

$$(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_\ell, y_\ell) \in \mathbb{R}^N \times \{\pm 1\}$$

Testing Dataset

$$(\bar{\mathbf{x}}_1, \bar{y}_1), \ldots, (\bar{\mathbf{x}}_{\bar{\ell}}, \bar{y}_{\bar{\ell}}) \in \mathbb{R}^N \times \{\pm 1\}, \text{ satisfying } \{\bar{\mathbf{x}}_1, \ldots, \bar{\mathbf{x}}_{\bar{\ell}}\} \cap \{\mathbf{x}_1, \ldots, \mathbf{x}_\ell\} = \{\}$$

Learning through empirical risk minimization

- Estimate $f(\vec{x}_i, \alpha)$ from a finite set of observations by minimizing some kind of an error function, for example, the empirical risk:

$$R_{emp}(\alpha) = \frac{1}{2l} \sum_{i=1}^{l} \left| y_i - f(\vec{x}_i, \alpha) \right|$$

α : Parameters of Learning Engine

class labels: $$y_i = \begin{cases} +1 & if \quad \vec{x}_i \in \omega_1 \\ -1 & if \quad \vec{x}_i \in \omega_2 \end{cases}$$

Learning through empirical risk minimization

- As a special case, estimate Linear *f(x)* from a finite set of observations by minimizing the empirical risk:

$$R_{emp}(\omega, \omega_0) = \frac{1}{2l} \sum_{i=1}^{l} \left| y_i - f(\vec{x}_i, \omega, \omega_0) \right|$$

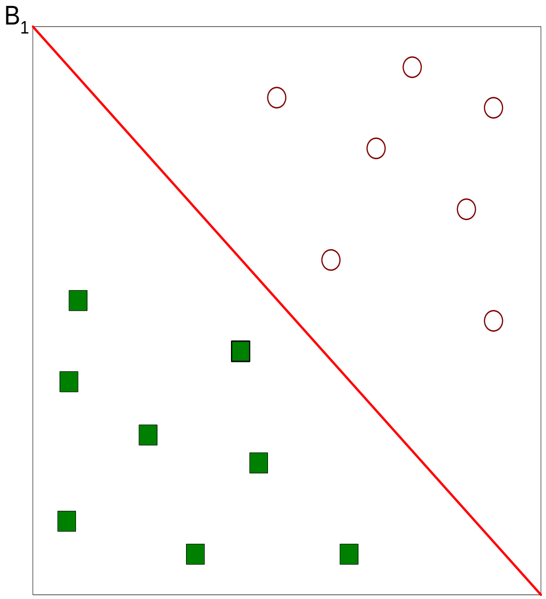
class labels: $$y_i = \begin{cases} +1 & if \quad \vec{x}_i \in \omega_1 \\ -1 & if \quad \vec{x}_i \in \omega_2 \end{cases}$$

Learning through empirical risk minimization

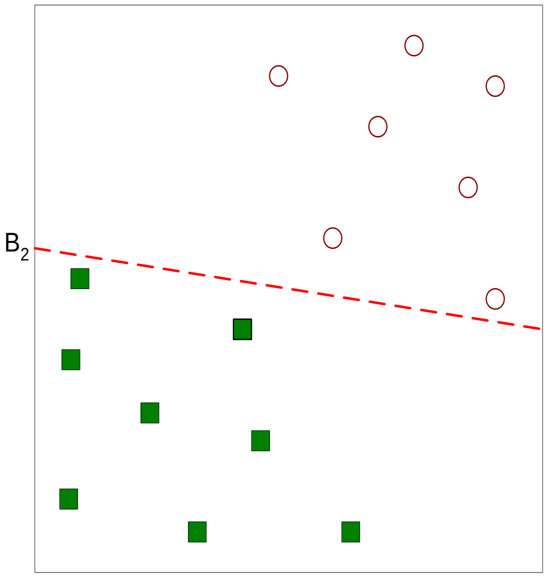
- Conventional empirical risk minimization over the training data **does not** imply good generalization to novel test data.
  - There could be a number of different functions which all approximate the training data set well.
  - Difficult to determine a function which **best** captures the true underlying structure of the data distribution (i.e., has good generalization capabilities)

Learning through empirical risk minimization

Solution 1

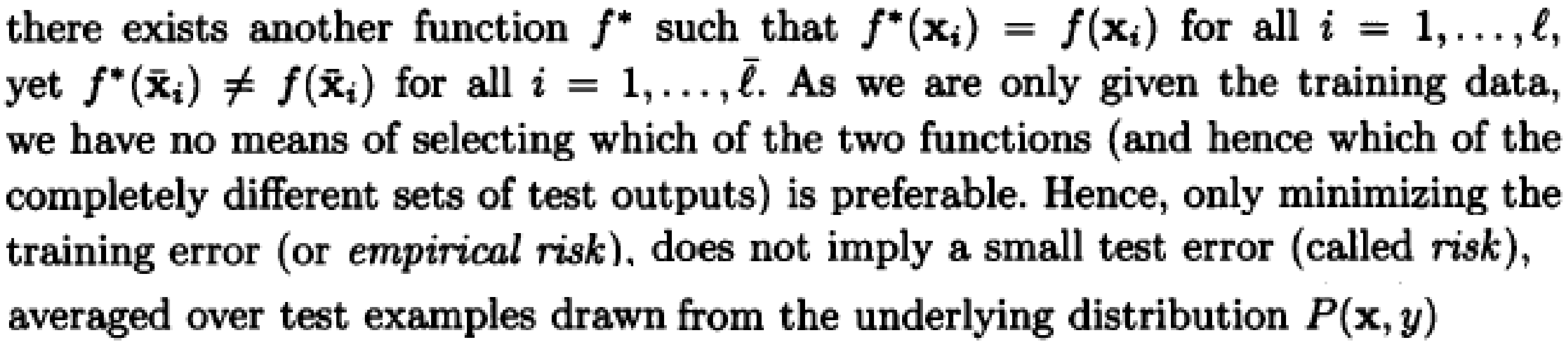


Solution 2



Which solution is better?

Learning through empirical risk minimization

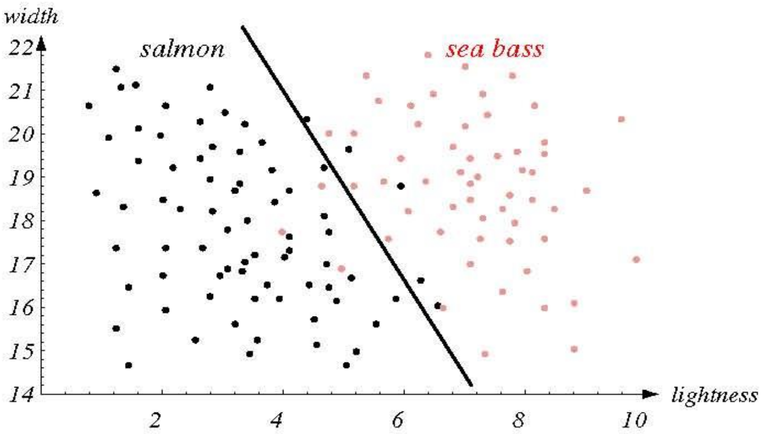


there exists another function f^* such that $f^*(\mathbf{x}_i) = f(\mathbf{x}_i)$ for all $i = 1, \ldots, \ell$, yet $f^*(\bar{\mathbf{x}}_i) \neq f(\bar{\mathbf{x}}_i)$ for all $i = 1, \ldots, \bar{\ell}$. As we are only given the training data, we have no means of selecting which of the two functions (and hence which of the completely different sets of test outputs) is preferable. Hence, only minimizing the training error (or *empirical risk*), does not imply a small test error (called *risk*), averaged over test examples drawn from the underlying distribution $P(\mathbf{x}, y)$

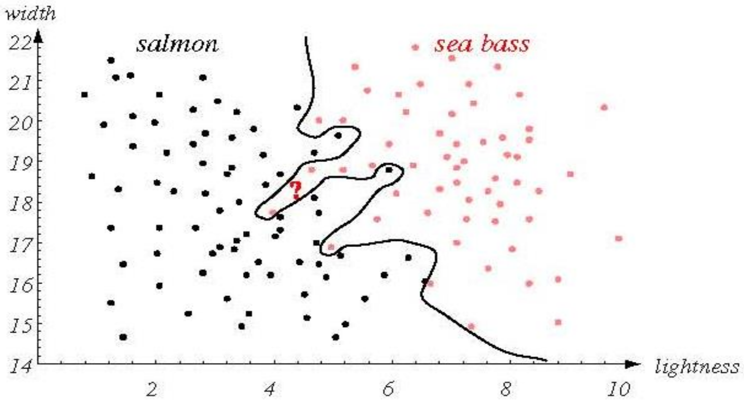
Learning through empirical risk minimization Capacity and VC dimension

- To guarantee good generalization performance, the **capacity** of the learned functions must be controlled.
- Functions with high **capacity** are more complicated (i.e., have many degrees of freedom**.**

low capacity



high capacity



Learning through empirical risk minimization
Capacity and VC dimension

- In statistical learning, the **Vapnik-Chervonenkis (VC) dimension** is one of the most popular measures of **capacity**.
- The **VC dimension** can predict a probabilistic upper bound on the test error (generalization error) of a classification model.

- A function that

  (1) minimizes the empirical risk **and**

  (2) has low VC dimension

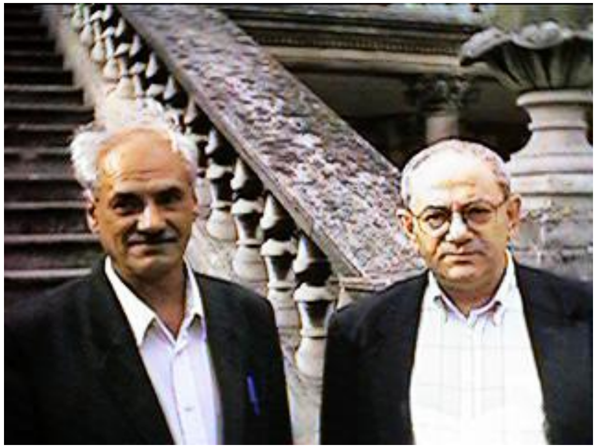
  will generalize well regardless of the dimensionality of the input space

VC-Dimension Definition (1)

Defenition 1:

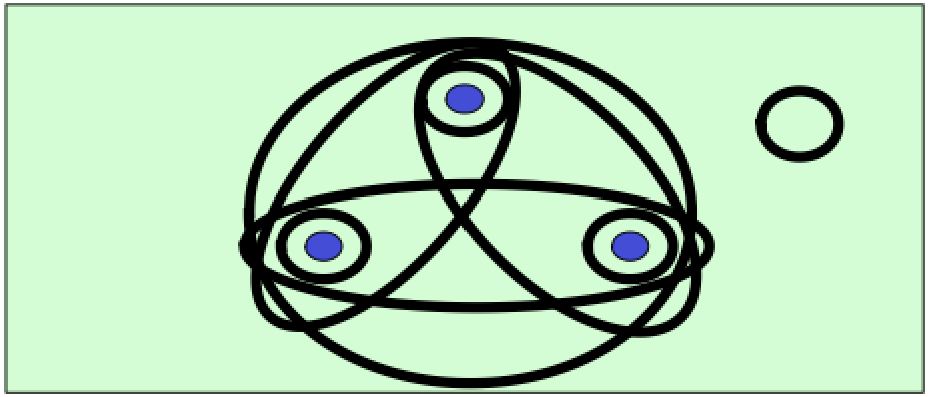
(***set shattering***): a subset S of instances of a set X is shattered by a collection of functions F *if* $\forall\ S' \subseteq S$ there is a function $f \in F$ *such data:*

$$f(x)=\begin{cases}1 & x\in S' \\ 0 & x\in S-S'\end{cases}$$



Alexey Chervonenkis
1938-2014

Vladimir Vapnik
1936-Present



VC-Dimension Definition (2)

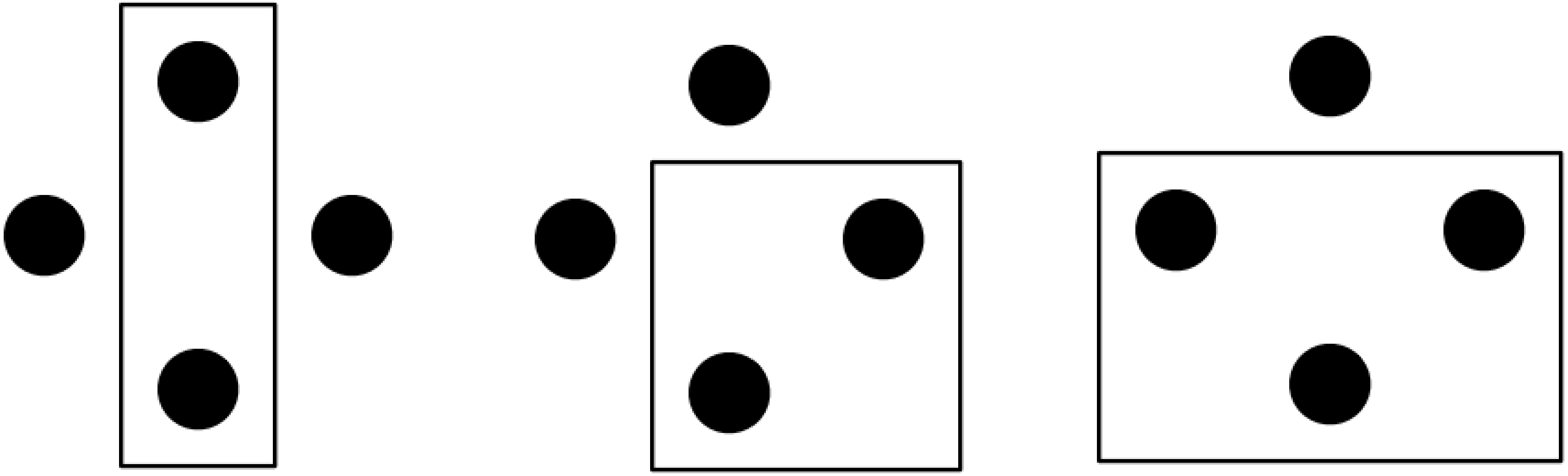
Defenition 2:
The VC-dimension of a function set *F* (VC-dim(*F*))
is the cardinality of the largest dataset that can be shattered by F.

VC-Dimension Examples - Rectangles

Rectangles with horizontal and vertical edges.

The VC dimension is 4. Why?

How we can capture 2 points and 3 points?



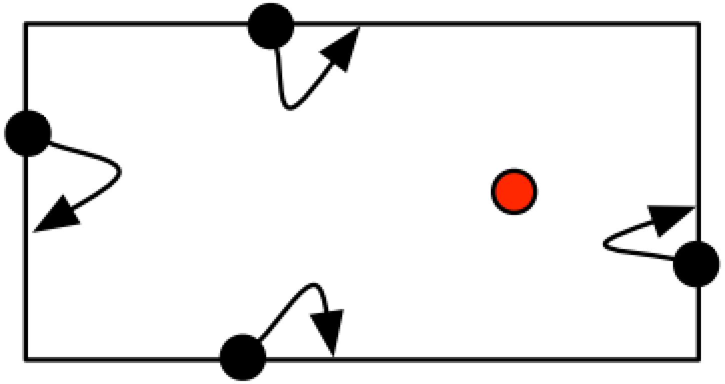
No set of 5 points can be shattered

VC-Dimension Examples - Rectangles

Rectangles with horizontal and vertical edges.

The VC dimension is 4. Why?

No set of 5 points can be shattered



Show that for squares with horizontal and vertical edges, the VC-dimension is three!

Show that for rotatable rectangles, the VC-dimension is seven!

What is the VC-dimension for rectangles in general?

What about Isosceles and equilateral triangles?

Ref: Bhaskar, A. and Sukhar, I. "VC-Dimension." 2008. http://www.cs.cornell.edu/courses/cs683/2008sp/lecture%20notes/683notes_0428.pdf.

VC-Dimension – Machine Learning Perspective

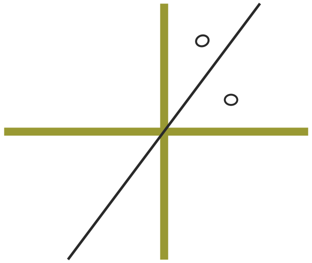
- Machine f can ***shatter*** a set of points $x_1, x_2 .. x_l$ if and only if…

  for every possible training set of the form (x_1,y_1) , (x_2,y_2) ,… (x_l ,y_l)

  …There exists some value of α that for which the $f\ (\vec{x}_i\ , \alpha)$

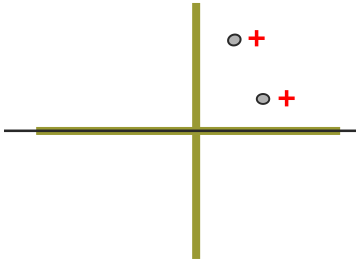
  gets zero training error.

- Question: Can the following f shatter the following points?

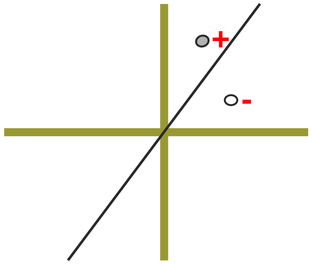


f(x,w) = sign(x.w)

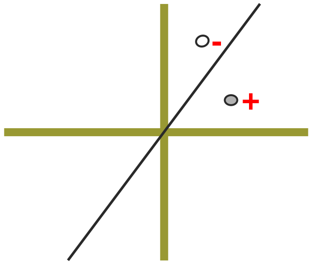
- Answer: No problem. There are four training sets to consider



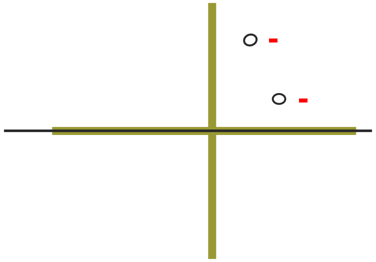
w=(0,1)



w=(-2,3)



w=(2,-3)



w=(0,-1)

Ref: Andrew W. Moore’s tutorials (2001): http://www.cs.cmu.edu/~awm/tutorials

VC-Dimension – Machine Learning Perspective

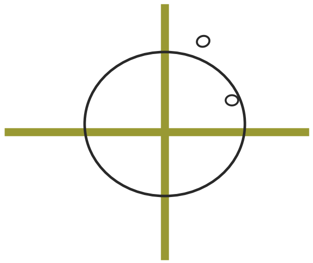
- Machine f can ***shatter*** a set of points $x_1, x_2 .. x_l$ if and only if…

  for every possible training set of the form (x_1,y_1) , (x_2,y_2) ,… $(x_l\,,y_l)$

  …There exists some value of α that for which the $f\ (\vec{x}_i\,, \alpha)$

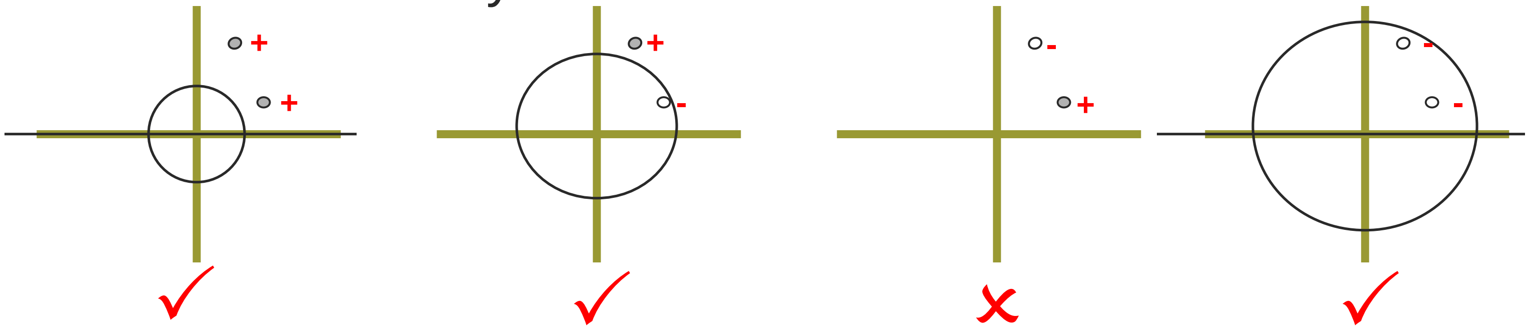
  gets zero training error.

- Question: Can the following f shatter the following points?



f(x,b) = sign(x.x-b)=sign($x_1^2 + x_2^2 - b$)

- Answer: No way!



Ref: Andrew W. Moore’s tutorials (2001): http://www.cs.cmu.edu/~awm/tutorials

VC-Dimension for Linear Classifiers

If input space is m-dimensional and if **f** is *sign*(***w***.***x***-*b*), what is the VC-dimension (*h*)?

Proof that *h* >= *m*: Show that *m* points can be shattered

Define m input points thus:

$\mathbf{x}_1 = (1,0,0,\ldots,0)$
$\mathbf{x}_2 = (0,1,0,\ldots,0)$
:
$\mathbf{x}_m = (0,0,0,\ldots,1)$ So $x_k[j] = 1$ if $k=j$ and 0 otherwise

Let $y_1, y_2, \ldots y_m$, be any one of the 2^m combinations of class labels.

Guess how we can define $w_1, w_2, \ldots w_m$ and b to ensure sign($\mathbf{w} . \mathbf{x}_k$ + b) = y_k for all *k* ? Note:

Answer: b=0 and $w_k = y_k$ for all *k.*

$$\operatorname{sign}(\mathbf{w}.\mathbf{x}_k + b) = \operatorname{sign}\left(b + \sum_{j=1}^{m} w_j \,.\, x_k[j]\right)$$

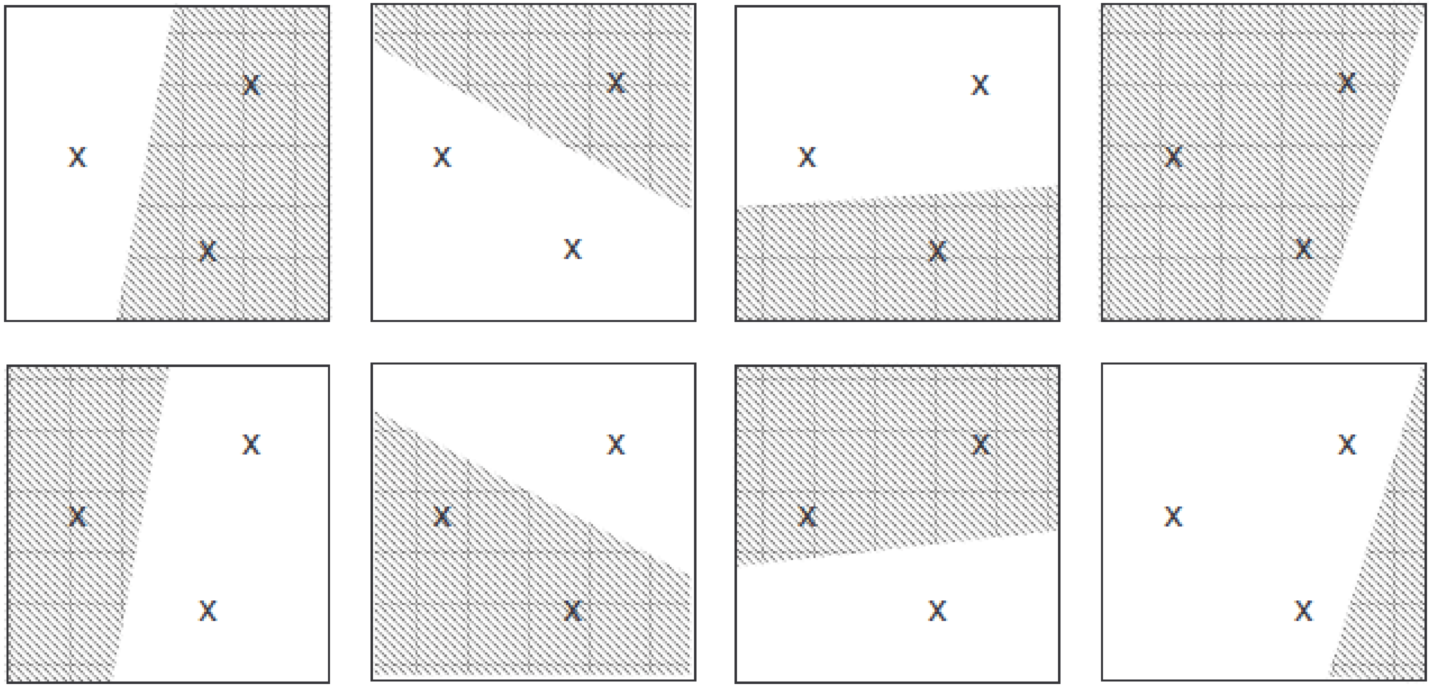
Ref: Andrew W. Moore's tutorials (2001): http://www.cs.cmu.edu/~awm/tutorials

VC dim of linear classifiers in m-dimensions

If input space is m-dimensional and if **f** is *sign*(***w***.***x***-*b*), what is the VC-dimension?

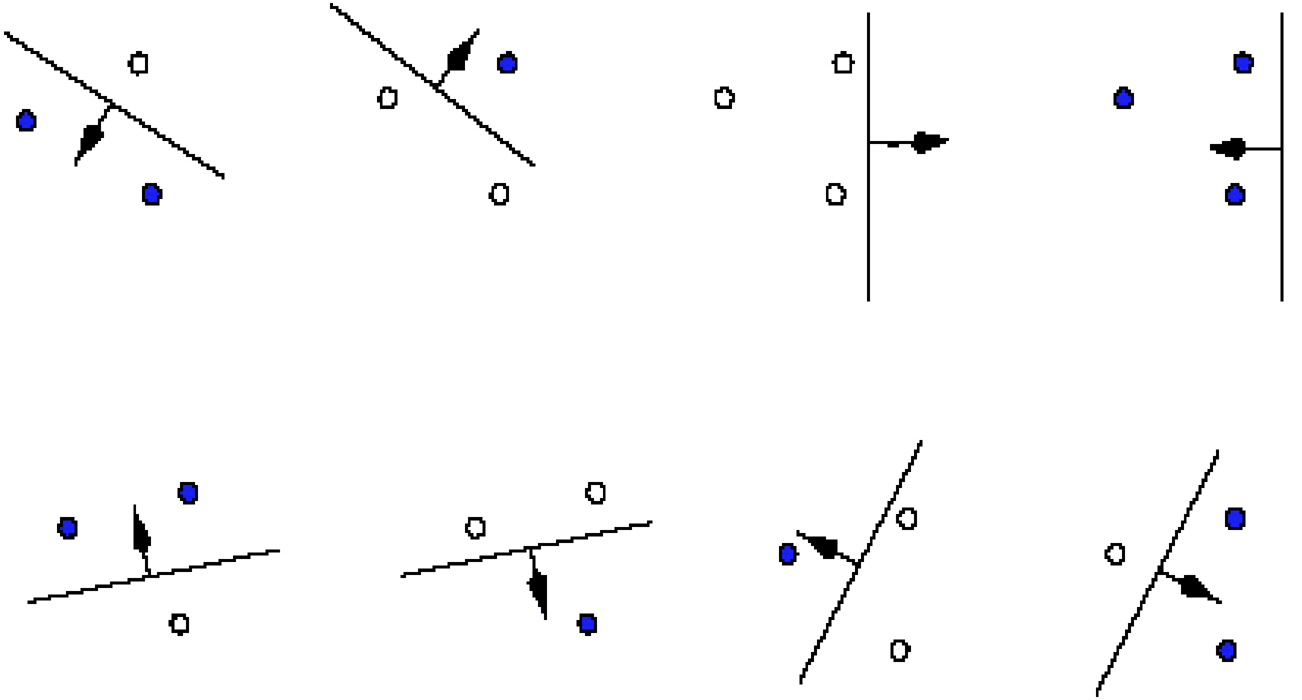
- Now we know that h >= m
- In fact, h=m+1
- Proof that h >= m+1 is easy
- Proof that h < m+2 is moderate

VC Dimension – An example



A simple VC dimension example. There are $2^3 = 8$ ways of assigning 3 points to two classes. For the displayed points in $\mathbb{R}^2$, all 8 possibilities can be realized using separating hyperplanes, in other words, the function class can shatter 3 points. This would not work if we were given 4 points, no matter how we placed them. Therefore, the VC dimension of the class of separating hyperplanes in $\mathbb{R}^2$ is 3.

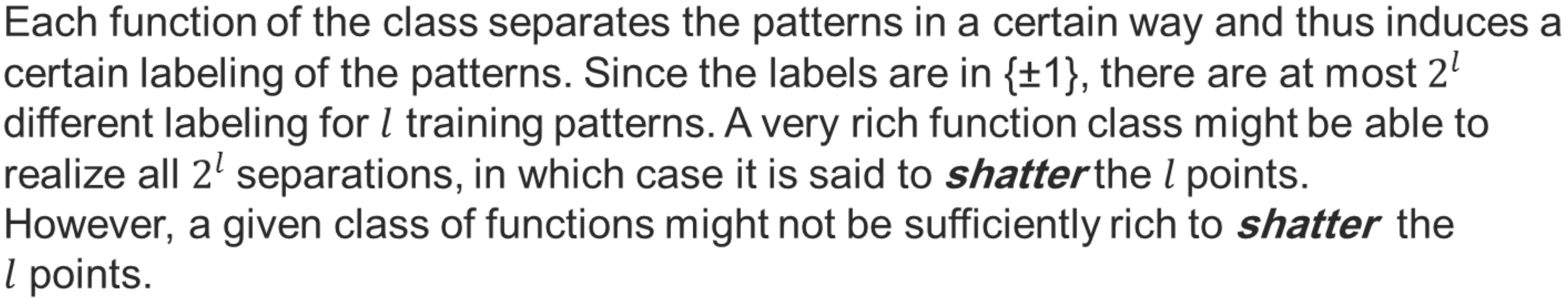
VC Dimension – An example



The VC-Dimension of the set of oriented hyperplanes in $\mathbf{R}^m$ is (m+1)

VC Dimension

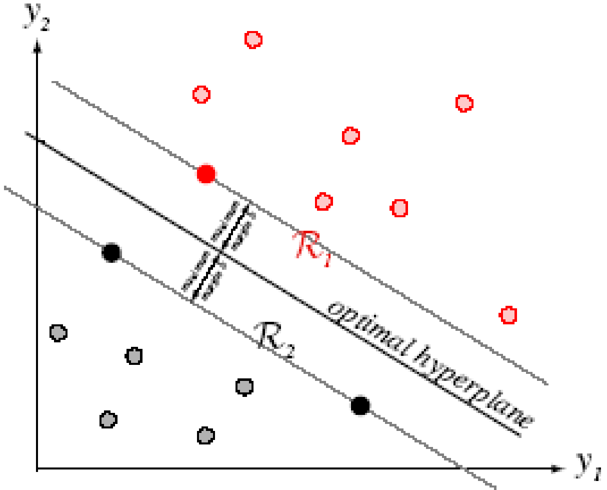
- The VC dimension is a property of a set of functions $\{f(\alpha)\}$, and can be defined for various classes of function $\boldsymbol{f}$.
- The VC dimension for the set of functions $\{f(\alpha)\}$ is defined as the maximum number of training points that can be shattered by $\{f(\alpha)\}$.



Each function of the class separates the patterns in a certain way and thus induces a certain labeling of the patterns. Since the labels are in {±1}, there are at most 2^l different labeling for l training patterns. A very rich function class might be able to realize all 2^l separations, in which case it is said to ***shatter*** the l points. However, a given class of functions might not be sufficiently rich to ***shatter*** the l points.

VC dimension and margin of separation

- Vapnik has shown that **maximizing** the margin of separation between classes is equivalent to **minimizing** the VC dimension (Hyper-planes have such properties).
- The optimal hyper-plane is the one giving the **largest margin** of separation between the classes.



Learning Machine

- A bound on the Generalization Performance of Learning Machine
  - Expected Risk: $R(f(\vec{x},\alpha)) = \int \frac{1}{2}|y - f(\vec{x},\alpha)| dP(\vec{x},y)$
  - Empirical Risk: $R_{emp}(f(\vec{x},\alpha)) = \frac{1}{2l}\sum_{i=1}^{l}|y_i - f(\vec{x}_i,\alpha)|$

$$\underbrace{R(f(\vec{x},\alpha))}_{\text{True Error}} \le \underbrace{R_{emp}(f(\vec{x},\alpha))}_{\text{Train Error}} + \sqrt{\underbrace{\left(\frac{h(\log(2l/h)+1)-\log(\eta/4)}{l}\right)}_{\text{Confidence Term}}}$$

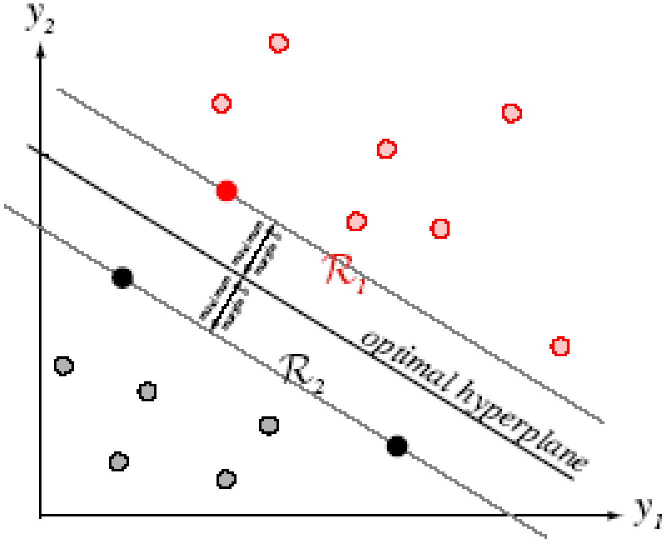
with probability *(1-η)*;   (*l*: training set size)

- h is the VC dimension, a measure of the notion of capacity of a classifier.

(Vapnik, 1995, "*Structural Risk Minimization Principle*")

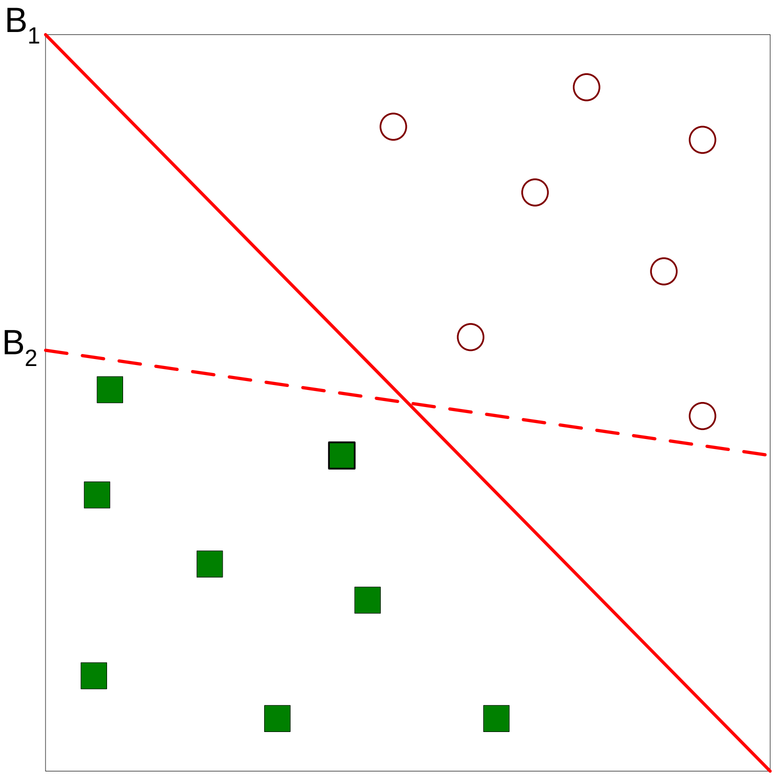
Margin of separation and support vectors

- The margin (i.e., empty area around the decision boundary) is defined by the distance to the nearest training patterns which we refer to as support vectors.
  - Intuitively speaking, these are the most difficult patterns to classify.

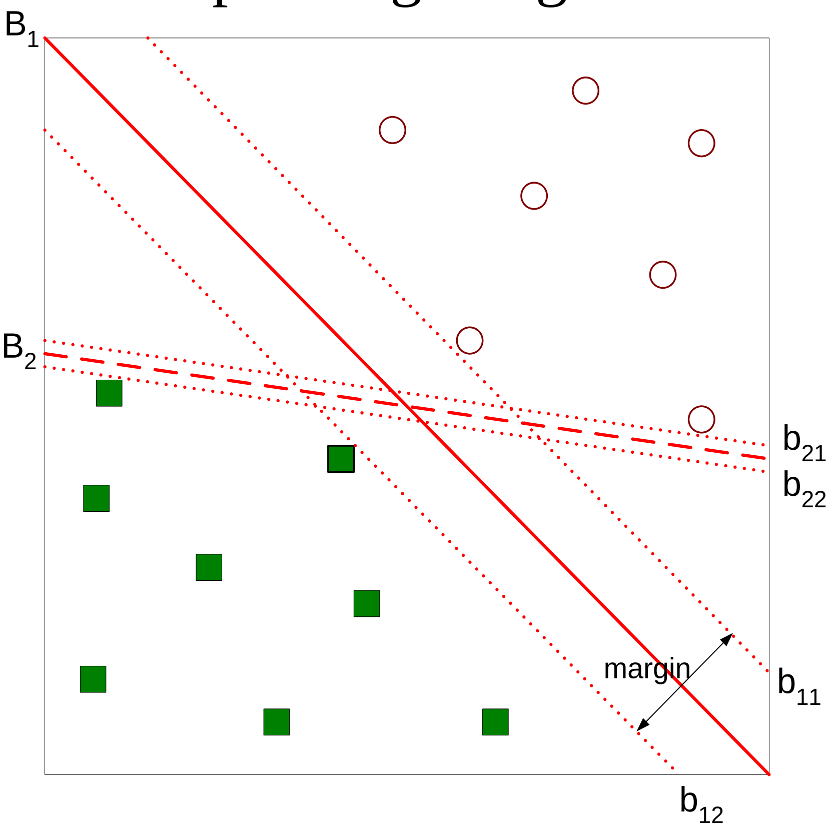


Margin of separation and support vectors (cont’d)

different solutions



corresponding margins



SVM Overview

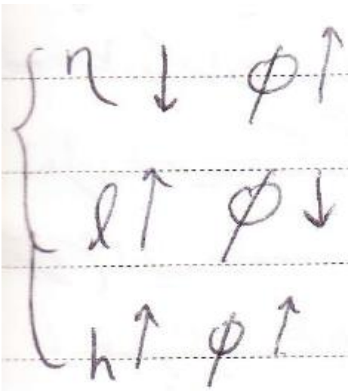
- SVMs perform structural risk minimization to achieve good generalization performance.

- The optimization criterion is the **margin** of separation between classes.

- Training is equivalent to solving a **quadratic programming** problem with **linear constraints**.

- Primarily **two-class** classifiers but can be extended to **multiple** classes.

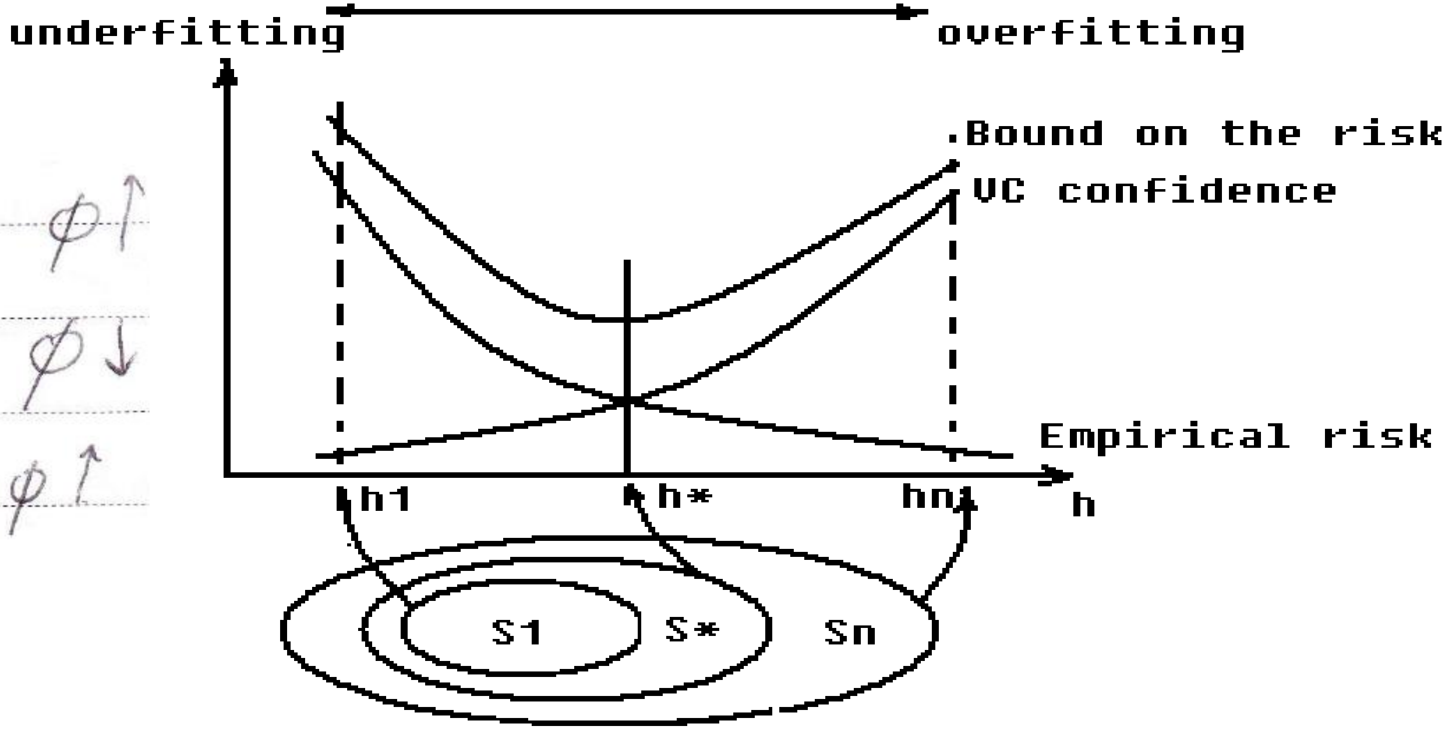
Structural Risk Minimization



$\eta \downarrow \quad \phi \uparrow$

$l \uparrow \quad \phi \downarrow$

$h \uparrow \quad \phi \uparrow$



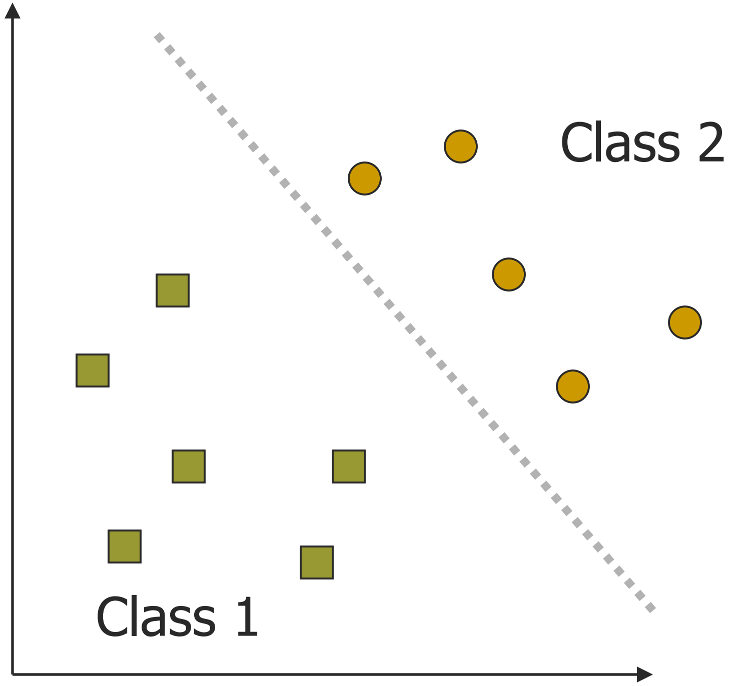
$$R(\alpha) \le R_{emp}(\alpha) + \sqrt{\left(\frac{h(\log(2l/h)+1) - \log(\eta/4)}{l}\right)}$$

Two Approaches

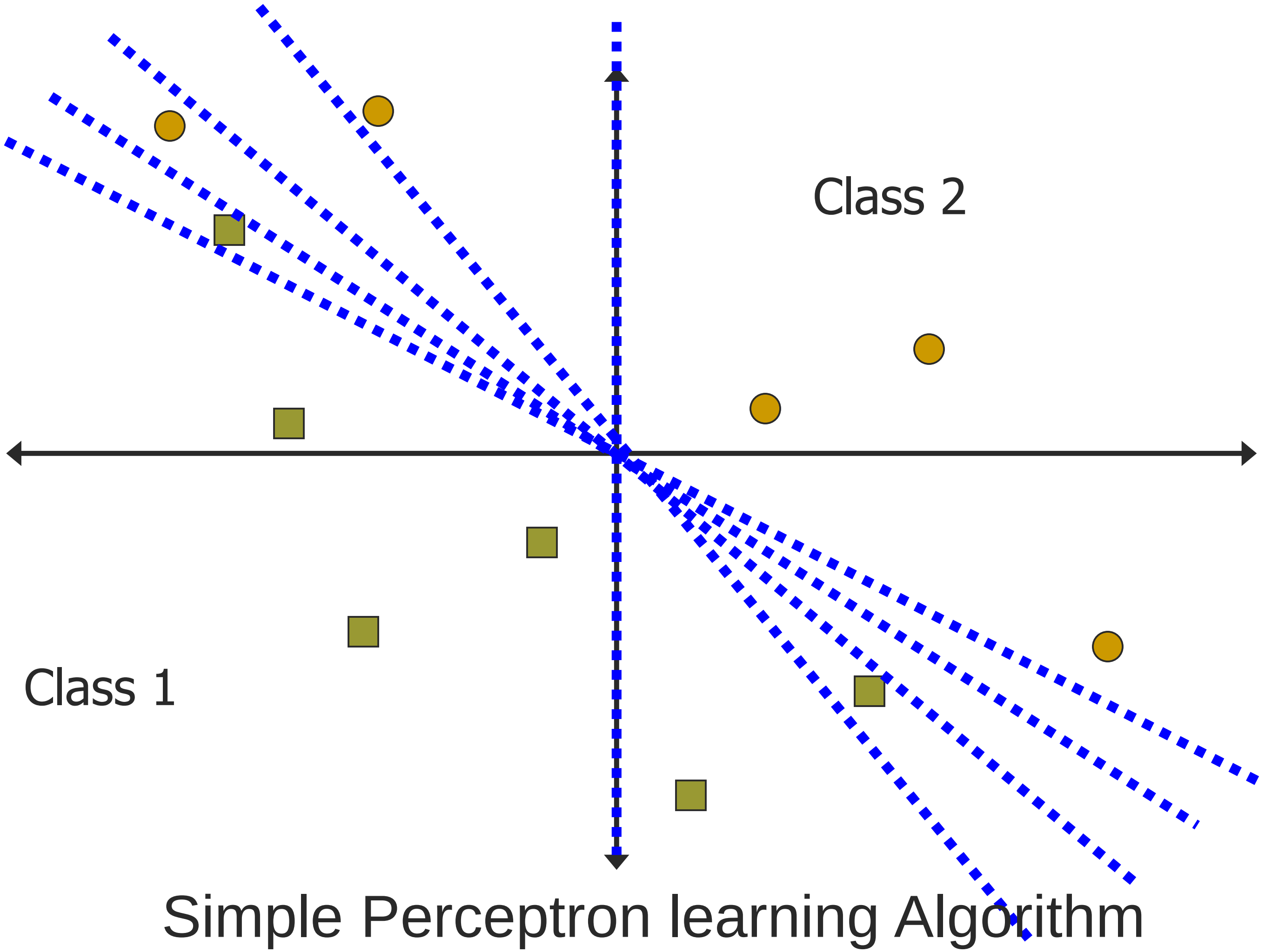
- **Goal:** To find a trained machine in the series whose sum of empirical risk and VC confidence is minimal.
- **Neural Network**
  - Fix the VC confidence and minimize the empirical risk. May be find High capacity mappings (very nonlinear → Overfitting).
- **Support Vector Machine**
  - Fix the empirical risk and minimize the VC confidence → Equivalent to maximizing the margin

The Two Class Problem

- Several decision boundaries can separate these two classes.
- Perceptron Algorithm learns any separating hyperplane.
- SVM learns the best separating hyperplane.

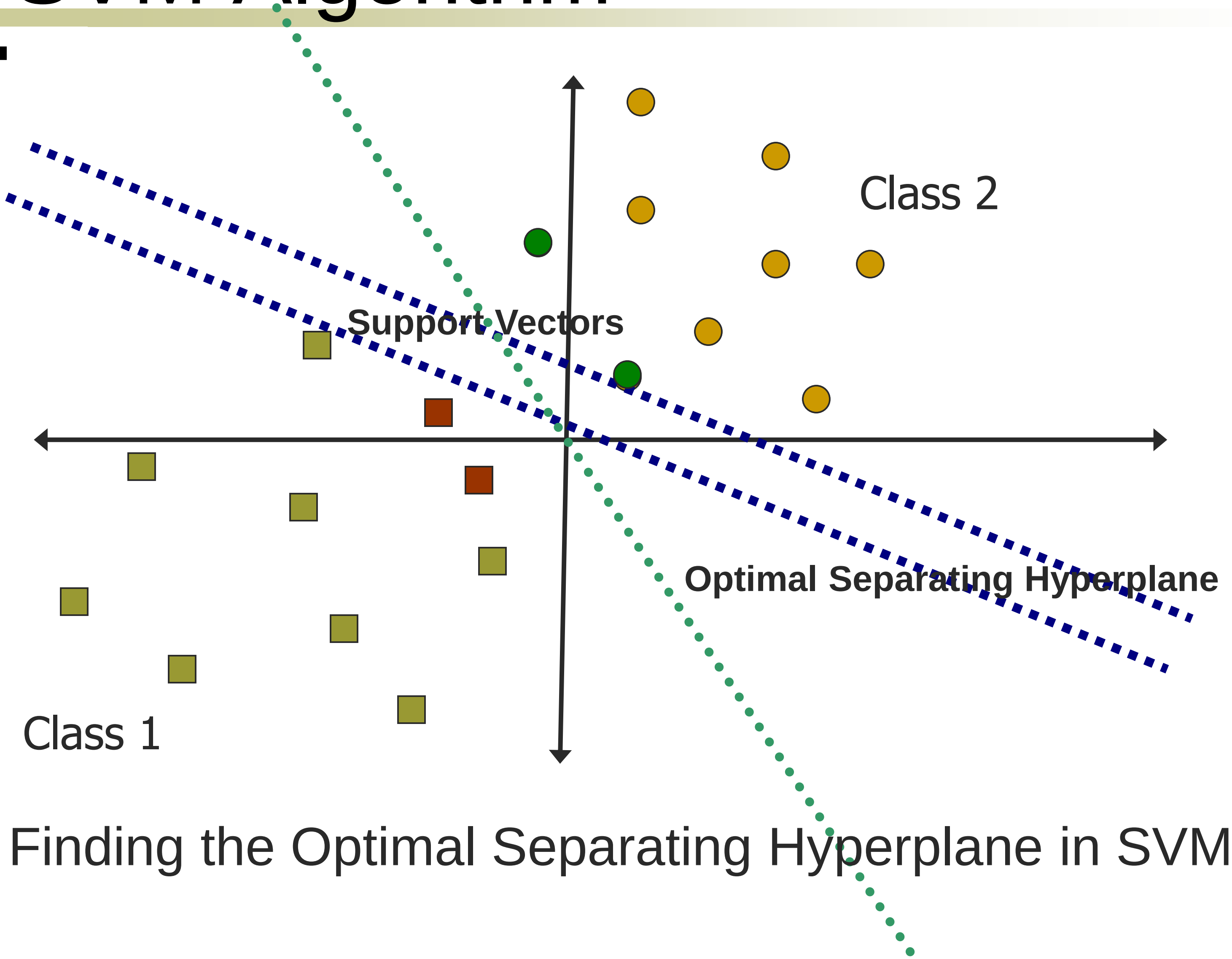


Perceptron Algorithm



Simple Perceptron learning Algorithm

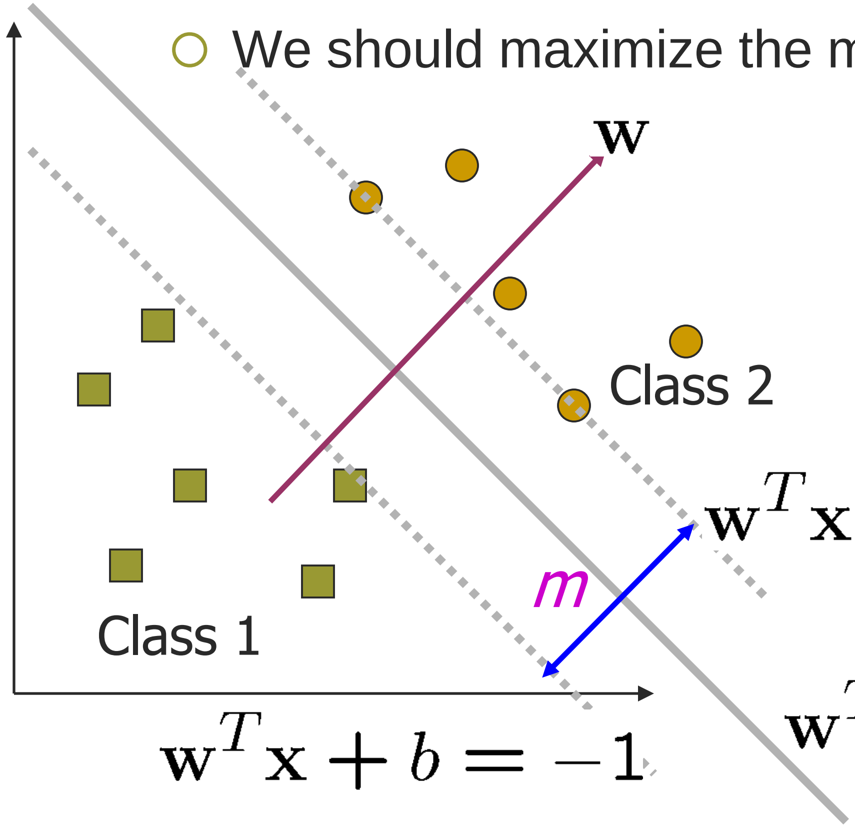
SVM Algorithm



Finding the Optimal Separating Hyperplane in SVM

Decision Boundary

- The decision boundary/hyperplane should be as far away from the data of both classes as possible.
  - ○ We should maximize the margin, *m*



$$m = \frac{2}{||\mathbf{w}||}$$

$\mathbf{w}^T\mathbf{x} + b = 1$

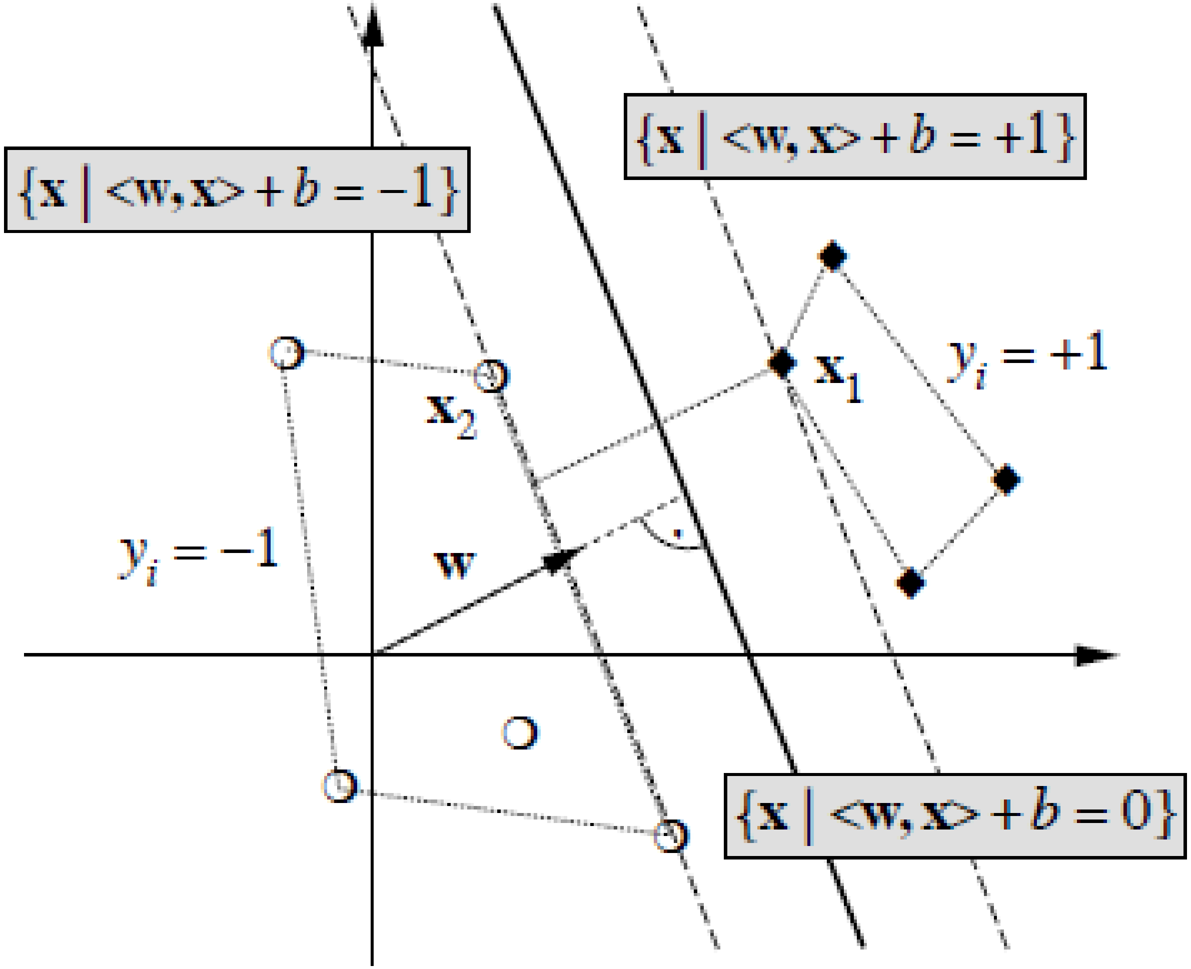
W and *b* are unknown.

$\mathbf{w}^T\mathbf{x} + b = 0$

$\mathbf{w}^T\mathbf{x} + b = -1$

$(w_1x_1 + w_1x_1 + \ldots + w_Nx_N + b = 0)$

Decision Boundary


$\{\mathbf{x} \mid \langle \mathbf{w}, \mathbf{x} \rangle + b = +1\}$
$\{\mathbf{x} \mid \langle \mathbf{w}, \mathbf{x} \rangle + b = -1\}$
$\mathbf{x}_1$
$y_i = +1$
$\mathbf{x}_2$
$y_i = -1$
$\mathbf{w}$
$\{\mathbf{x} \mid \langle \mathbf{w}, \mathbf{x} \rangle + b = 0\}$

Note:

$$\langle \mathbf{w}, \mathbf{x}_1 \rangle + b = +1$$
$$\langle \mathbf{w}, \mathbf{x}_2 \rangle + b = -1$$

$$\Rightarrow \quad \langle \mathbf{w}, (\mathbf{x}_1 - \mathbf{x}_2) \rangle = 2$$

$$\Rightarrow \left\langle \frac{\mathbf{w}}{\|\mathbf{w}\|}, (\mathbf{x}_1 - \mathbf{x}_2) \right\rangle = \frac{2}{\|\mathbf{w}\|}$$

The Optimization Problem of SVM

we have the following optimization problem:

$$\min_{w,b} \quad \frac{1}{2}||w||^2$$

$$\text{s.t.} \quad y^{(i)}(w^T x^{(i)} + b) \geq 1, \quad i = 1, \ldots, l \quad (l : \text{number of training samples})$$

- The above is an optimization problem with a convex quadratic objective and only linear constraints. Its solution gives us the optimal margin classifier. This optimization problem can be solved using commercial quadratic programming (QP) code.
- **Lagrange duality** will lead us to this optimization problem’s dual form, which will play a key role in allowing us to use kernels to get optimal margin classifiers to work efficiently in very high dimensional spaces.
- The **dual form** will also allow us to derive an efficient algorithm for solving the above optimization problem that will typically do much better than generic QP software.

Lagrange Duality

- In optimization theory, **duality** or the **duality principle** is the principle that optimization problems may be viewed from either of two perspectives, the **primal problem** or the **dual problem**.

- The solution to the dual problem provides a lower bound to the solution of the primal (minimization) problem.

- However in general the optimal values of the primal and dual problems need not be equal.

- Their difference is called the duality gap.

- For convex optimization problems, the duality gap is zero under a constraint qualification condition (such as **Karush-Kuhn-Tucker (KKT)** conditions).

- Usually the term "dual problem" refers to the ***Lagrangian dual problem,*** but other dual problems are used

Lagrange Duality

- The Lagrangian dual problem is obtained by forming the **Lagrangian** of a minimization problem by using nonnegative **Lagrange multipliers** to add the constraints to the objective function, and then solving for the primal variable values that minimize the original objective function.

- This solution gives the primal variables as functions of the **Lagrange multipliers**, which are called **dual variables**, so that the new problem is to maximize the objective function with respect to the dual variables under the derived constraints on the dual variables (including at least the nonnegativity constraints).

Lagrange Duality

The **primal** (against **dual**) optimization problem :

(ω is **primal** variable)

$$\begin{aligned} \min_w \quad & f(w) \\ \text{s.t.} \quad & g_i(w) \leq 0, \quad i = 1, \ldots, k \\ & h_i(w) = 0, \quad i = 1, \ldots, t. \end{aligned}$$

The **Lagrangian dual problem** is defined as:

$$\max \;\; \mathcal{L}(w, \alpha, \beta) = f(w) + \sum_{i=1}^{k} \alpha_i g_i(w) + \sum_{i=1}^{t} \beta_i h_i(w).$$

$$s.t. \quad \alpha_i \geq 0, \qquad i = 1, \ldots, k$$

Consider the quantity $\theta_{\mathcal{P}}(w) = \max_{\alpha, \beta \,:\, \alpha_i \geq 0} \mathcal{L}(w, \alpha, \beta)$ the $\mathcal{P}$ stands for “primal.”

Lagrange Duality

Consider the following, which we'll call the **primal** (against **dual**) optimization problem :

(ω is **primal** variable)

$$\begin{aligned} \min_w \quad & f(w) \\ \text{s.t.} \quad & g_i(w) \le 0, \quad i = 1, \ldots, k \\ & h_i(w) = 0, \quad i = 1, \ldots, t. \end{aligned}$$

To solve it, the **generalized Lagrangian** can be used:

$$\mathcal{L}(w, \alpha, \beta) = f(w) + \sum_{i=1}^{k} \alpha_i g_i(w) + \sum_{i=1}^{t} \beta_i h_i(w).$$

the α_i's and β_i's are the Lagrange multipliers or **dual** variables

Consider the quantity $\theta_{\mathcal{P}}(w) = \max_{\alpha, \beta \,:\, \alpha_i \ge 0} \mathcal{L}(w, \alpha, \beta)$ the $\mathcal{P}$ stands for "primal."

Lagrange Duality

$$\begin{aligned} \min_w \quad & f(w) \\ \text{s.t.} \quad & g_i(w) \le 0, \quad i = 1, \dots, k \\ & h_i(w) = 0, \quad i = 1, \dots, t. \end{aligned}$$

For some given ω :

- If ω violates any of the **primal** constraints (i.e., if either $g_i(w) > 0$ or $h_i(w) \neq 0$ for some i) then we have:

$$\theta_{\mathcal{P}}(w) \quad = \quad \max_{\alpha,\beta:\alpha_i \ge 0} f(w) + \sum_{i=1}^{k} \alpha_i g_i(w) + \sum_{i=1}^{l} \beta_i h_i(w) = \infty$$

- Otherwise: $\theta_{\mathcal{P}}(w) = f(w)$

Hence:

$$\theta_{\mathcal{P}}(w) = \begin{cases} f(w) & \text{if } w \text{ satisfies primal constraints} \\ \infty & \text{otherwise.} \end{cases}$$

Lagrange Duality

From previous we have:

$$\theta_{\mathcal{P}}(w) = \max_{\alpha,\beta\,:\,\alpha_i \geq 0} \mathcal{L}(w, \alpha, \beta)$$

and

$$\theta_{\mathcal{P}}(w) = \begin{cases} f(w) & \text{if } w \text{ satisfies primal constraints} \\ \infty & \text{otherwise.} \end{cases}$$

Now, consider the following minimization problem:

$$\min_w \theta_{\mathcal{P}}(w) = \min_w \max_{\alpha,\beta\,:\,\alpha_i \geq 0} \mathcal{L}(w, \alpha, \beta),$$

is the same problem and has the same solutions as our original, primal problem.

Also, let define the optimal value of the objective to be $p^* = \min_w \theta_{\mathcal{P}}(w)$

we call this the **value of the primal problem**

Lagrange Duality

Now, we define: $\theta_{\mathcal{D}}(\alpha, \beta) = \min_w \mathcal{L}(w, \alpha, \beta)$ the $\mathcal{D}$ stands for "dual."

Note that in the definition of $\theta_{\mathcal{P}}$ we were optimizing (maximizing) with respect to α and β , But, here we are minimizing with respect to ω.

Now, the dual optimization problem can be posed as follows:

$$\max_{\alpha,\beta\,:\,\alpha_i\geq 0} \theta_{\mathcal{D}}(\alpha, \beta) = \max_{\alpha,\beta\,:\,\alpha_i\geq 0} \min_w \mathcal{L}(w, \alpha, \beta)$$

This is exactly the same as our primal problem shown above, except that the order of the "max" and the "min" are now exchanged.

let define the optimal value of the objective to be $d^* = \max_{\alpha,\beta\,:\,\alpha_i\geq 0} \theta_{\mathcal{D}}(\alpha, \beta)$

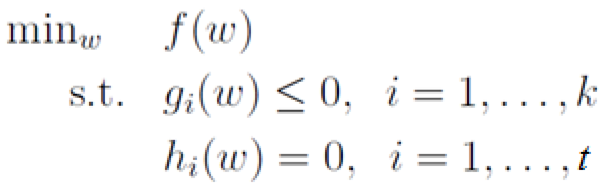
It can be shown that:

$$d^* = \max_{\alpha,\beta\,:\,\alpha_i\geq 0} \min_w \mathcal{L}(w, \alpha, \beta) \leq \min_w \max_{\alpha,\beta\,:\,\alpha_i\geq 0} \mathcal{L}(w, \alpha, \beta) = p^*$$

Because the "max min" of a function always being less than or equal to the "min max"

under certain conditions, we will have: $d^* = p^*$

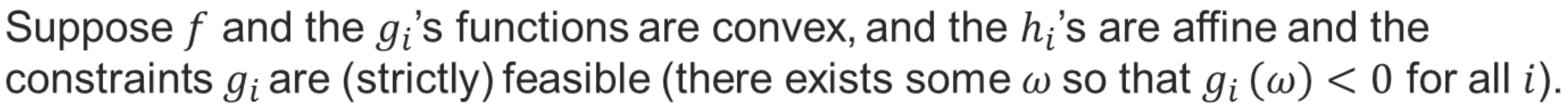
Lagrange Duality, Equality Conditions



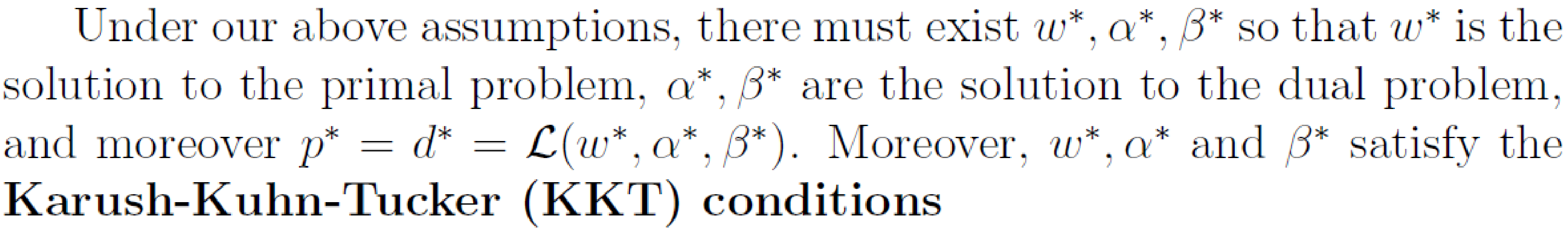
$$\begin{aligned} \min_w \quad & f(w) \\ \text{s.t.} \quad & g_i(w) \leq 0, \quad i = 1, \ldots, k \\ & h_i(w) = 0, \quad i = 1, \ldots, t \end{aligned}$$

we can solve the dual problem in lieu of the primal problem in $d^* = p^*$ conditions.

Conditions:



Suppose f and the g_i's functions are convex, and the h_i's are affine and the constraints g_i are (strictly) feasible (there exists some ω so that $g_i(\omega) < 0$ for all i).



Under our above assumptions, there must exist w^*, α^*, β^* so that w^* is the solution to the primal problem, α^*, β^* are the solution to the dual problem, and moreover $p^* = d^* = \mathcal{L}(w^*, \alpha^*, \beta^*)$. Moreover, w^*, α^* and β^* satisfy the **Karush-Kuhn-Tucker (KKT) conditions**

Lagrange Duality, Equality Conditions

Karush-Kuhn-Tucker (KKT) conditions

Theorem: Suppose the problem

$$\begin{aligned} \min_w \quad & f(w) \\ \text{s.t.} \quad & g_i(w) \leq 0, \quad i = 1, \ldots, k \\ & h_i(w) = 0, \quad i = 1, \ldots, t \end{aligned}$$

has a local (maximum) minimum at $\omega = \omega^*$, and that a constraint qualification (to be specified) is satisfied at ω^*. Then there are $\alpha_1^*, \alpha_2^*, \cdots, \alpha_k^*$ such that:

$$\begin{aligned} \alpha_i^* g_i(w^*) &= 0, \quad i = 1, \ldots, k \\ g_i(w^*) &\leq 0, \quad i = 1, \ldots, k \\ \alpha^* &\geq 0, \quad i = 1, \ldots, k \end{aligned}$$

KKT Complementarity Condition

According to complementarity condition, if $\alpha_i^* > 0$, then it implies that $g_i(w^*) = 0$
This is the key for showing that the SVM has only a small number of “support vectors”.

The KKT conditions are necessary conditions for a local maximum (maximum). They don't guarantee that a point satisfying them is actually a local maximum (maximum).

Lagrange Duality, Equality Conditions

Karush-Kuhn-Tucker (KKT) conditions

$$\begin{aligned}
\frac{\partial}{\partial w_i}\mathcal{L}(w^*,\alpha^*,\beta^*) &= 0, \quad i=1,\ldots,n \\
\frac{\partial}{\partial \beta_i}\mathcal{L}(w^*,\alpha^*,\beta^*) &= 0, \quad i=1,\ldots,t \\
\alpha_i^* g_i(w^*) &= 0, \quad i=1,\ldots,k \\
g_i(w^*) &\leq 0, \quad i=1,\ldots,k \\
\alpha^* &\geq 0, \quad i=1,\ldots,k
\end{aligned}$$

(n: feature space dimensionality)

KKT Complementarity Condition

if some w^*, α^*, β^* satisfy the KKT conditions, then it is also a solution to the primal and dual problems.

According to complementarity condition, if $\alpha_i^* > 0$, then it implies that $g_i(w^*) = 0$ This is the key for showing that the SVM has only a small number of “support vectors”.

The KKT conditions are necessary conditions for a local maximum (maximum). They don't guarantee that a point satisfying them is actually a local maximum (maximum).

Lagrange Duality, Maximizing SVM Margin

Again, the (primal) optimization problem for finding the optimal margin classifier:

$$\begin{aligned} \min_{w,b} \quad & \frac{1}{2}||w||^2 \\ \text{s.t.} \quad & y^{(i)}(w^T x^{(i)} + b) \geq 1, \quad i = 1, \ldots, l \end{aligned}$$

We can write the constraints as:

$$g_i(w) = -y^{(i)}(w^T x^{(i)} + b) + 1 \leq 0$$

from the KKT dual complementarity condition, we will have $\alpha_i > 0$ only for the training examples that have functional margin exactly equal to one

Lagrangian for SVM

$$\mathcal{L}(w,b,\alpha) = \frac{1}{2}||w||^2 - \sum_{i=1}^{l} \alpha_i \left[y^{(i)}(w^T x^{(i)} + b) - 1 \right]$$

Note that there're only "α_i" but no "β_i" Lagrange multipliers, since the problem has only inequality constraints.

Let's find the dual form of the problem. To get $\theta_{\mathcal{D}}$ we need to first maximize $\mathcal{L}(w,b,\alpha)$ by setting the derivatives of $\mathcal{L}$ with respect to ω and b (for fixed α) to zero.

$$\nabla_w \mathcal{L}(w,b,\alpha) = w - \sum_{i=1}^{l} \alpha_i y^{(i)} x^{(i)} = 0 \quad \Longrightarrow \quad w = \sum_{i=1}^{l} \alpha_i y^{(i)} x^{(i)}$$

$$\frac{\partial}{\partial b} \mathcal{L}(w,b,\alpha) = \sum_{i=1}^{l} \alpha_i y^{(i)} = 0$$

$$\mathcal{L}(w,b,\alpha) = \sum_{i=1}^{l} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{l} y^{(i)} y^{(j)} \alpha_i \alpha_j (x^{(i)})^T x^{(j)} - b \sum_{i=1}^{l} \alpha_i y^{(i)}$$

Inner product of $x^{(i)}$ and $x^{(j)}$

Lagrangian for SVM

$$\mathcal{L}(w,b,\alpha)=\sum_{i=1}^{l}\alpha_i-\frac{1}{2}\sum_{i,j=1}^{l}y^{(i)}y^{(j)}\alpha_i\alpha_j\underset{\textbf{Inner product of } x^{(i)} \textbf{ and } x^{(j)}}{(x^{(i)})^T x^{(j)}}-\underbrace{b\sum_{i=1}^{l}\alpha_i y^{(i)}}_{\text{zero}}$$

we got to the equation above by minimizing $\mathcal{L}$ with respect to ω and b.

Also,

$$\alpha_i \geq 0 \quad \text{and} \quad \frac{\partial}{\partial b}\mathcal{L}(w,b,\alpha)=\sum_{i=1}^{m}\alpha_i y^{(i)}=0$$

Therefore, we obtain the following dual optimization problem:

$$\begin{aligned}\max_{\alpha}\quad & W(\alpha)=\sum_{i=1}^{l}\alpha_i-\frac{1}{2}\sum_{i,j=1}^{l}y^{(i)}y^{(j)}\alpha_i\alpha_j\langle x^{(i)},x^{(j)}\rangle\\ \text{s.t.}\quad & \alpha_i\geq 0,\quad i=1,\ldots,l\\ & \sum_{i=1}^{l}\alpha_i y^{(i)}=0,\end{aligned}$$

$$f(x)=sgn\left(\sum_{i=1}^{l}y^{(i)}\alpha_i\langle x,x^{(i)}\rangle+b\right)$$

Decision hyper-plane

Lagrange duality, Equality Conditions

$$\begin{aligned} \max_\alpha \quad & W(\alpha) = \sum_{i=1}^{l} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{l} y^{(i)} y^{(j)} \alpha_i \alpha_j \langle x^{(i)}, x^{(j)} \rangle \\ \text{s.t.} \quad & \alpha_i \geq 0, \quad i = 1, \ldots, l \\ & \sum_{i=1}^{l} \alpha_i y^{(i)} = 0, \end{aligned}$$

The conditions required for $d^* = p^*$ and the KKT conditions to hold are indeed satisfied in our optimization problem. Hence, we can solve the dual in lieu of solving the primal problem. Specifically, in the dual problem above, we have a maximization problem in which the parameters are the α_i's.

Lagrange duality, Equality Conditions

$$\begin{aligned} \max_\alpha \quad & W(\alpha) = \sum_{i=1}^{l} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{l} y^{(i)} y^{(j)} \alpha_i \alpha_j \langle x^{(i)}, x^{(j)} \rangle \\ \text{s.t.} \quad & \alpha_i \geq 0, \quad i = 1, \ldots, l \\ & \sum_{i=1}^{l} \alpha_i y^{(i)} = 0, \end{aligned}$$

If we to solve it (i.e., find the α’s that maximize W(α), subject to the constraints), then we can use

$$w = \sum_{i=1}^{l} \alpha_i y^{(i)} x^{(i)}$$

To find the optimal w’s as a function of the α’s. Having found w*, by considering the primal problem, it is also straightforward to find the optimal value for term b as:

$$b^* = -\frac{\max_{i:y^{(i)}=-1} w^{*T} x^{(i)} + \min_{i:y^{(i)}=1} w^{*T} x^{(i)}}{2}$$

Decision hyper-plane

$$f(x) = sgn\left(\sum_{i=1}^{l} y^{(i)} \alpha_i \langle x, x^{(i)} \rangle + b^* \right)$$

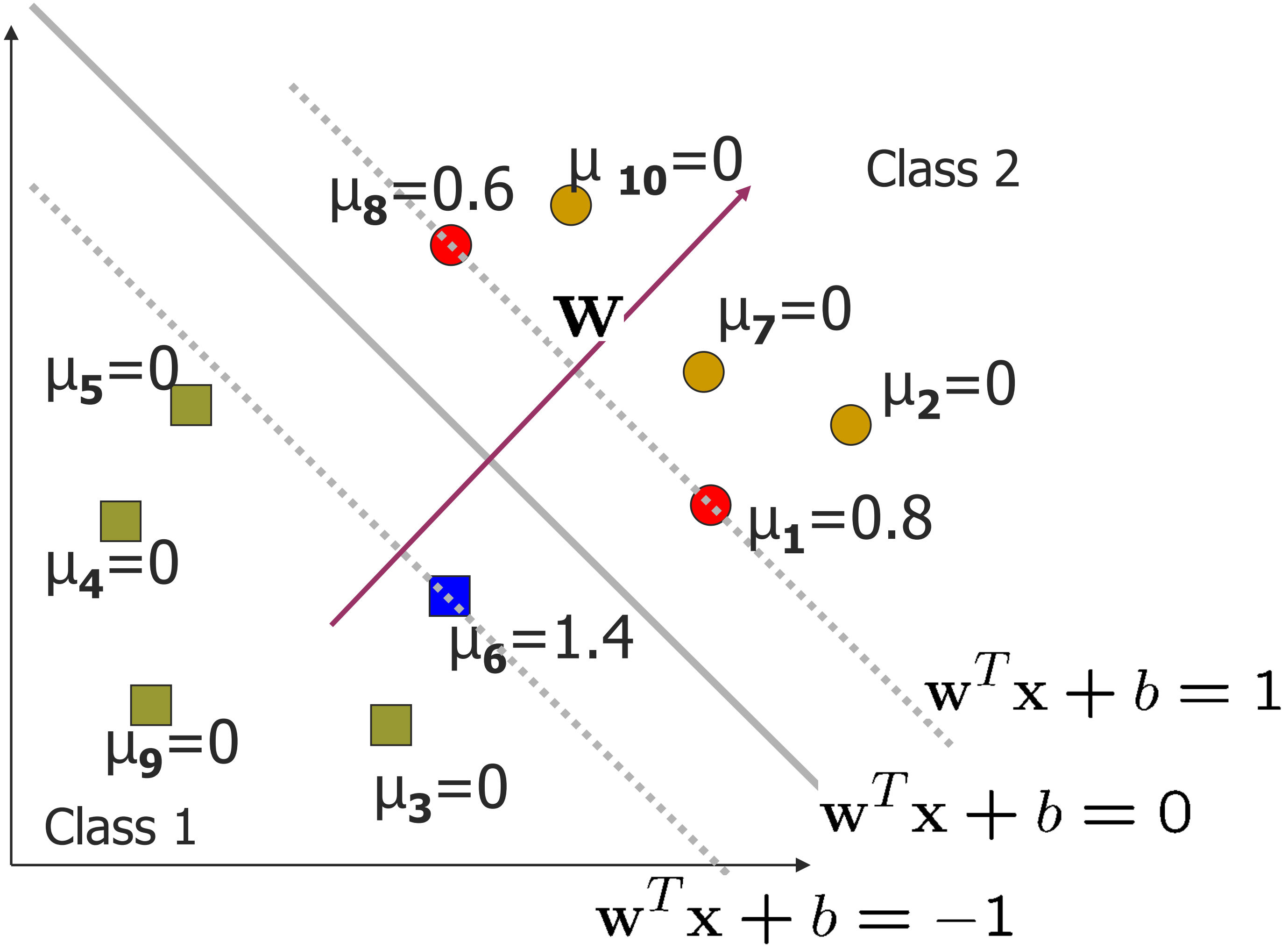
Lagrange Duality

- By Karush-Kuhn-Tucker theorem, the solution we find in the dual problem will be the same as the solution to the original problem.

- But why are we doing this???? (why not just solve the original problem????)

- Because this will let us solve the problem by computing the just the inner products of x_i , x_j (which will be very important later on when we want to solve non-linearly separable classification problems)

A Geometrical Interpretation



Some Notes

- There are theoretical upper bounds on the error on unseen data for SVM
  - The larger the margin, the smaller the bound
  - The smaller the number of SV, the smaller the bound
- Note that in both training and testing, the data are referenced only as inner product, $\mathbf{x}^T\mathbf{y}$
  - This is important for generalizing to the non-linear case