Support Vector Machines

Dr. Daniele Cattaneo, Prof. Dr. Josif Grabocka

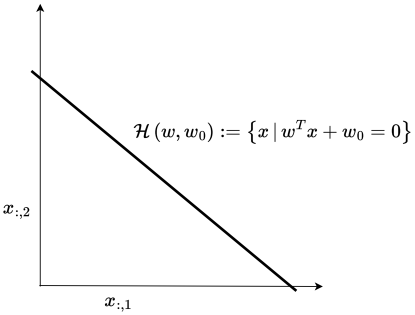
Machine Learning Course
Winter Semester 2023/2024

Albert-Ludwigs-Universität Freiburg

cattaneo@informatik.uni-freiburg.de, grabocka@informatik.uni-freiburg.de

November 06, 2023

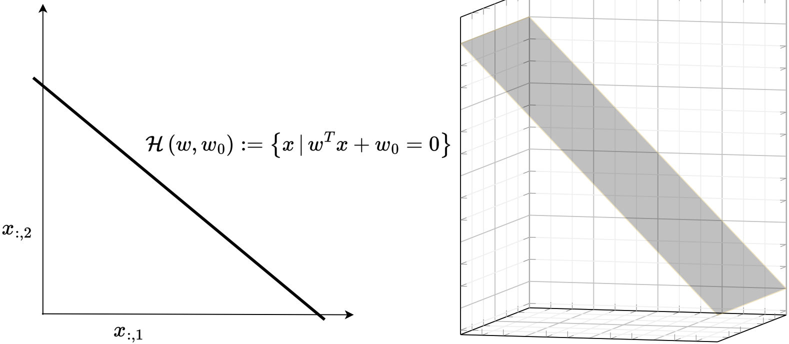
Linear Hyperplane



Example linear hyperplanes in 2D.

A linear hyperplane $\mathcal{H}(w, w_0)$ is a sub-space with dimension one less than the dimension of the space $x \in \mathbb{R}^M$.
Question: what would be the hyperplane of a 3-D space?.

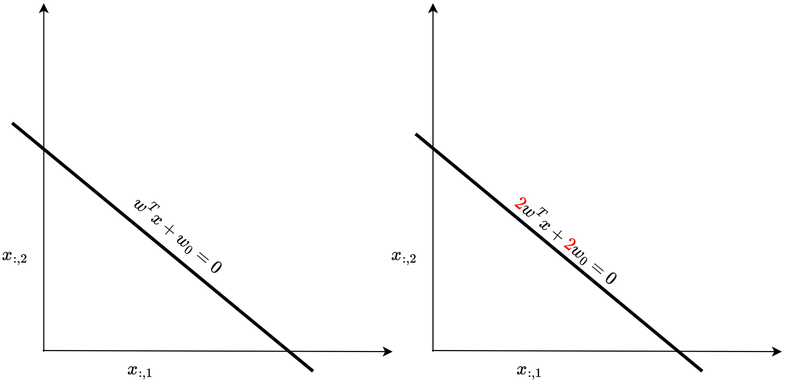
Linear Hyperplane



Example linear hyperplanes in 2D and 3D.

Answer: A 2D plane, as shown in the right figure.
Property: the hyperplane divides the space into two parts.

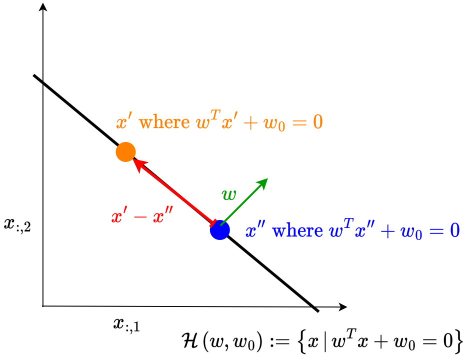
Linear Hyperplane — Scaling w, w_0



Infinitely-many scaled w, w_0 yield the same hyperplane.

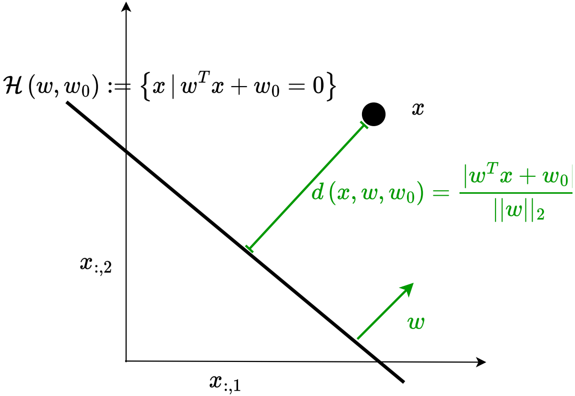
$$w^T x + w_0 = \beta \left(w^T x + w_0\right) = (\beta w)^T x + \beta w_0 = 0, \ \forall \beta \in \mathbb{R}, \beta \neq 0$$

w is orthogonal to the hyperplane



Subtracting the hyperplane equations:
$w^T x' + w_0 - (w^T x'' + w_0) = w^T (x' - x'') = 0$.
Using the dot product definition:
$w^T (x' - x'') = ||w||_2 ||x' - x''||_2 \cos(w, x' - x'') = 0$.
Cosine zero means w is orthogonal to vectors on the plane.

Distance between a hyperplane and a Point

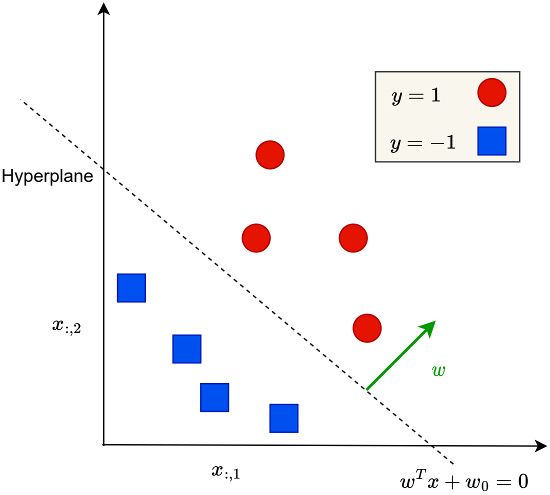


Scaling w, w_0 by any $\beta \in \mathbb{R}, \beta \neq 0$ yields the same distance.

$$d(x, \beta w, \beta w_0) = \frac{|(\beta w)^T x + \beta w_0|}{\sqrt{(\beta w)^T (\beta w)^T}} = \frac{|\beta||w^T x + w_0|}{\sqrt{\beta^2}\sqrt{w^T w}} = d(x, w, w_0)$$

A linear model for a linearly-separable binary classification

- Features $x \in \mathbb{R}^{N \times M}$, Target $y_i \in \{-1, 1\}^N$



Perceptron: Linear Classification Model

Linear classification problem:

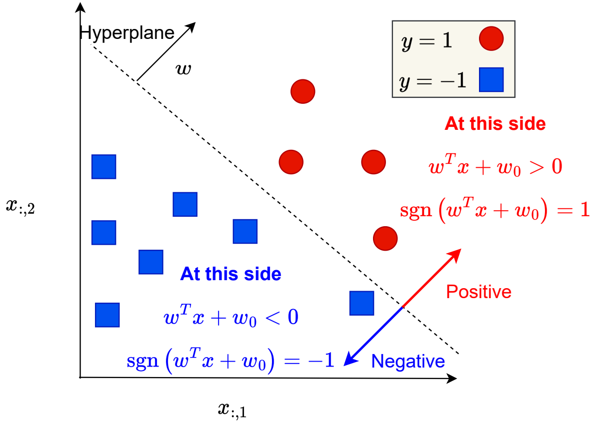
- Model $f(\cdot; w) : \mathbb{R}^M \to \{-1, 1\}$ with params $w \in \mathbb{R}^{M+1}$

$$w^{\text{opt}} := \operatorname*{argmin}_w \sum_{i=1}^{N} \mathcal{L}\left(y_i, f(x_i; w)\right)$$

Linear model with a sign function:

$$f(x; w) := \operatorname{sgn}(w^T x + w_0), \quad \text{with} \quad \operatorname{sgn}(x) = \begin{cases} 1 & x > 0 \\ 0 & x = 0 \\ -1 & x < 0 \end{cases}$$

Geometric Interpretation of the Linear Classifier



Classification errors can be formalized as:

$$\forall i: \quad y_i \left(w^T x_i + w_0\right) < 0, \text{ or } \forall i: \quad y_i \, \mathrm{sgn}\left(w^T x_i + w_0\right) = -1$$

Optimizing the perceptron

Loss over miss-classified instances $y_i \neq \operatorname{sgn}\left(w^T x_i + w_0\right)$ as:

$$w^{\text{opt}} := \operatorname*{argmin}_{w} \sum_{i=1:y_i f(x_i;w)=-1}^{N} -y_i\left(w^T x_i + w_0\right)$$

Define the **gradient**: (here $\mathcal{L}_i = \mathcal{L}\left(y_i, f(x_i; w, w_0)\right)$):

$$\frac{\partial \sum_i \mathcal{L}_i}{\partial w} = \sum_{i=1:\ y_i f(x_i;w)=-1}^{N} -y_i x_i; \qquad \frac{\partial \sum_i \mathcal{L}_i}{\partial w_0} = \sum_{i=1:\ y_i f(x_i;w)=-1}^{N} -y_i$$

Update by step $\eta \in \mathbb{R}_+$ with $\forall (x_i, y_i) : y_i f(x_i; w) = -1$:

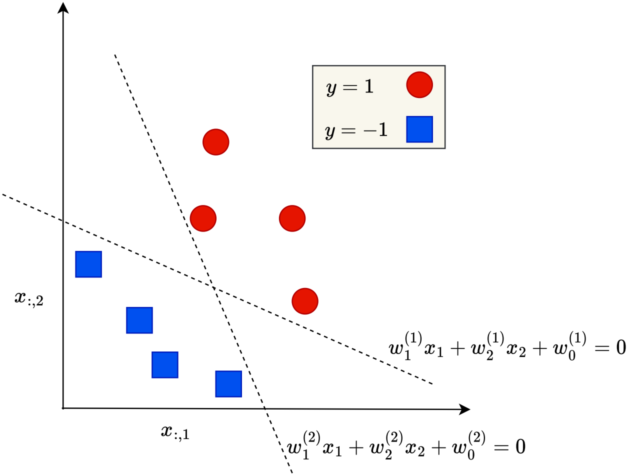
$$w^{(t)} \leftarrow w^{(t-1)} + \eta y_i x_i, \qquad w_0^{(t)} \leftarrow w_0^{(t-1)} + \eta y_i$$

Algorithm 1 Learning the Perceptron Model

Require: Data $x \in \mathbb{R}^{N \times M}, y_i \in \{-1, 1\}^N$, Learning rate $\eta \in \mathbb{R}^+$
Ensure: $w \in \mathbb{R}^M, w_0 \in \mathbb{R}$

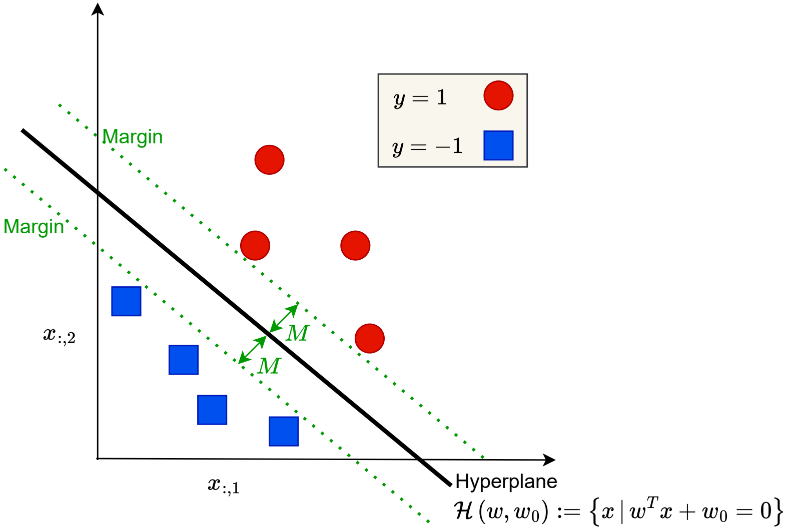
$w \sim \mathcal{N}(0, \sigma^2)_M, w_0 \sim \mathcal{N}(0, \sigma^2)$ ▷ Random initial hyperplane
errors $\leftarrow 1$
while errors > 0 **do**
    errors $\leftarrow 0$
    for $i = 1, \ldots, N$ **do**
        if $y_i \neq \text{sgn}\left(w^T x_i + w_0\right)$ **then**
            errors $\leftarrow$ errors $+ 1$
            $w \leftarrow w + \eta y_i x_i$
            $w_0 \leftarrow w_0 + \eta y_i$
return w, w_0

Sub-optimality of the Linear Classifier

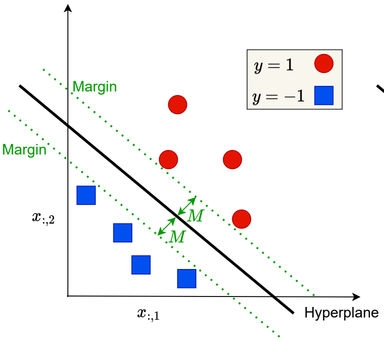


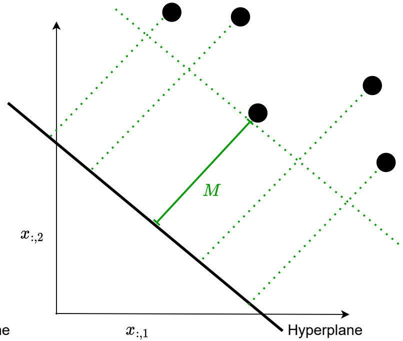
Multiple solution hyperplanes exist. Which one is the optimal?

Intuition: Maximum Margin Hyperplane

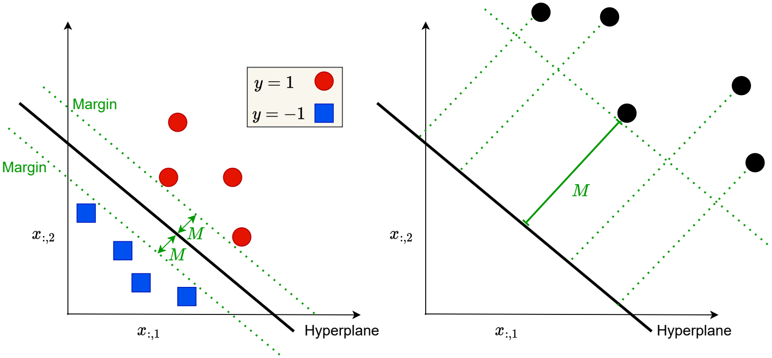


Margin of a hyperplane to a set of points





Margin of a hyperplane to a set of points



Given points $\{x_1, \ldots, x_N\}$ and plane w, w_0:

$$M(w, w_0) = \min_{x \in \{x_1, \ldots, x_N\}} d(x, w, w_0) = \min_{x \in \{x_1, \ldots, x_N\}} \frac{|w^T x + w_0|}{||w||_2}$$

Maximum margin hyperplane

- Can I optimize the hyperplane directly to yield the maximum margin as follows?

$$\underset{w,w_0}{\operatorname{argmax}}\ M(w, w_0) = \underset{w,w_0}{\operatorname{argmax}} \min_{x \in \{x_1,\ldots,x_N\}} \frac{|w^T x + w_0|}{||w||_2}$$

Maximum margin hyperplane

- Can I optimize the hyperplane directly to yield the maximum margin as follows?

$$\underset{w,w_0}{\operatorname{argmax}}\; M(w, w_0) = \underset{w,w_0}{\operatorname{argmax}} \min_{x \in \{x_1,\ldots,x_N\}} \frac{|w^T x + w_0|}{||w||_2}$$

  - No, the margin will be increased to infinity.
  - We need to keep the hyperplane between the two classes.

Definition of the Max Margin Hyperplane

Ensure all data points are correctly classified as constraints:

$$\underset{w,w_0}{\operatorname{argmax}} \min_{x \in \{x_1,\ldots,x_N\}} \frac{|w^T x + w_0|}{||w||_2}$$

$$\text{s.t. } \forall i: \quad y_i \left(w^T x_i + w_0\right) \geq 0$$

Get $||w||_2$ out of the inner minimization:

$$\underset{w,w_0}{\operatorname{argmax}} \frac{1}{||w||_2} \min_{x \in \{x_1,\ldots,x_N\}} |w^T x + w_0|$$
$$\text{s.t. } \forall i: \quad y_i \left(w^T x_i + w_0\right) \geq 0$$

Simplifying the optimization

Notice there are infinitely many w, w_0 for the same plane:

$$\mathcal{H}(w, w_0) = \left\{ x \mid w^T x + w_0 = 0 \right\}$$

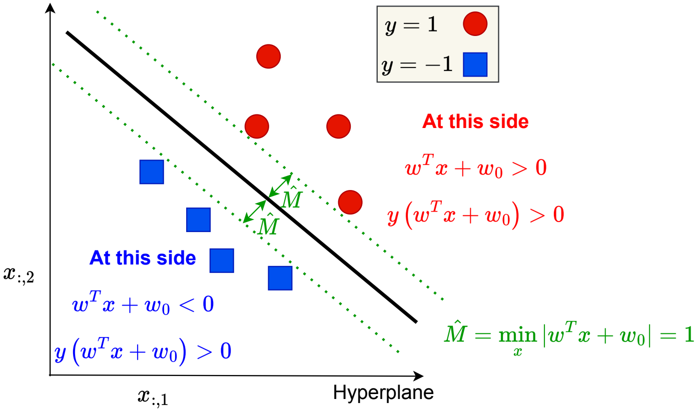
We restrict the infinite space of parameters w, w_0 to a subset:

$$1 = \min_{x \in \{x_1, \ldots, x_N\}} |w^T x + w_0|$$

... in order to simplify our optimization:

$$\begin{aligned} &\operatorname*{argmax}_{w, w_0} \frac{1}{||w||_2} \\ &\quad \text{s.t. } \forall i : \quad y_i \left(w^T x_i + w_0 \right) \geq 0 \\ &\quad \text{s.t. } \min_{x \in \{x_1, \ldots, x_N\}} |w^T x + w_0| = 1 \end{aligned}$$

Enforcing a margin of 1 unit



Unifying the constraints

The constraints here:

$$
\begin{aligned}
&\underset{w,w_0}{\operatorname{argmax}} \frac{1}{||w||_2} \\
&\quad \text{s.t. } \forall i: \quad y_i \left(w^T x_i + w_0 \right) \geq 0 \\
&\quad \text{s.t. } \min_{x \in \{x_1, \ldots, x_N\}} |w^T x + w_0| = 1
\end{aligned}
$$

are equivalent to:

$$
\begin{aligned}
&\underset{w,w_0}{\operatorname{argmax}} \frac{1}{||w||_2} \\
&\quad \text{s.t. } \forall i: \quad y_i \left(w^T x_i + w_0 \right) \geq 1
\end{aligned}
$$

Converting the objective to a minimization

Convert the maximization of $||w||_2$:

$$\underset{w,w_0}{\operatorname{argmax}} \frac{1}{||w||_2}$$

$$\text{s.t. } \forall i : \quad y_i \left(w^T x_i + w_0 \right) \geq 1$$

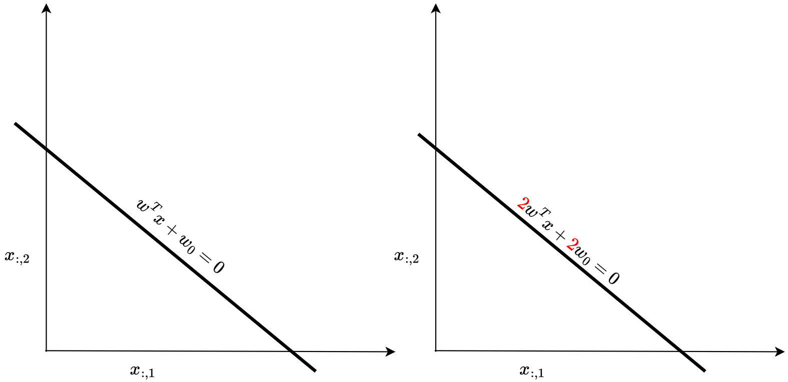
To a minimization of $w^T w$:

$$\underset{w,w_0}{\operatorname{argmin}} \; w^T w$$

$$\text{s.t. } \forall i : \quad y_i \left(w^T x_i + w_0 \right) \geq 1$$

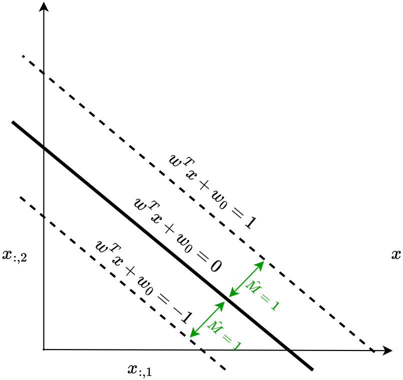
Yielding the objective function for a Linear SVM on a linearly separable task.

Why minimizing $w^T w$? - Geometric Interpretation



Reminder: by scaling w, w_0, the hyperplane remains the same

Why minimizing $w^T w$? - Geometric Interpretation



... but the margin decreases inversely proportional to the scaling factor. **Note:** by definition, $\hat{M}$ (also called functional margin) remains 1, but the actual distance M decreases.

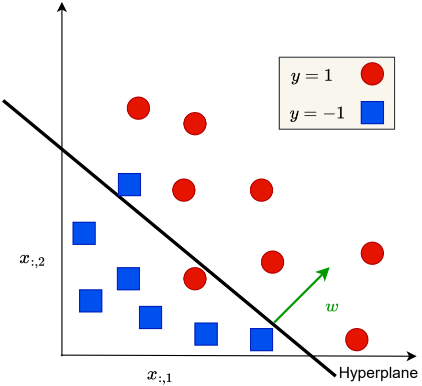
Violations to the Linear Separability Assumption

Can we solve w, w_0:

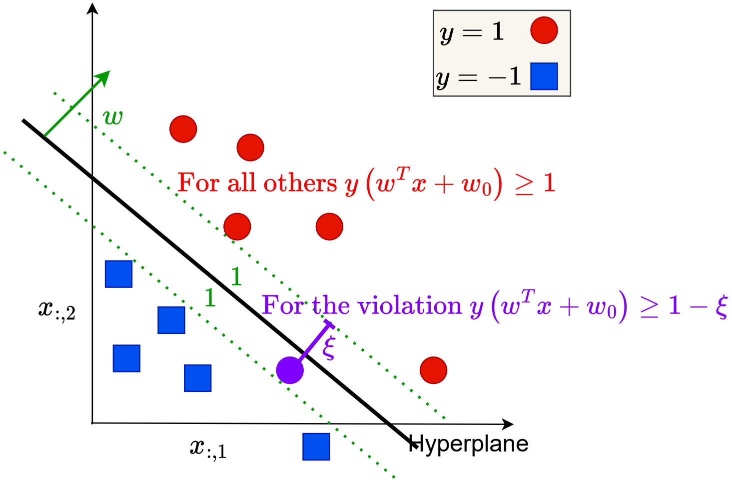
$$\underset{w,w_0}{\text{argmin}}\ w^T w$$

$$\text{s.t. } \forall i: \quad y_i \left(w^T x_i + w_0 \right) \geq 1$$

for $\forall i : x_i, y_i$ from the dataset on the right?



Slack margin



Tolerate mistakes

... by an amount of violation ξ_i in correctly classifying each y_i, x_i:

$$\begin{aligned} &\underset{w,w_0}{\operatorname{argmin}}\ w^T w \\ &\quad \text{s.t. } \forall i: \quad y_i \left(w^T x_i + w_0 \right) \geq 1 - \xi_i \\ &\quad \text{s.t. } \forall i: \quad \xi_i \geq 0 \end{aligned}$$

... but the total amount of violations should be minimized:

$$\begin{aligned} &\underset{w,w_0}{\operatorname{argmin}}\ w^T w + C \sum_{i=1}^{N} \xi_i \\ &\quad \text{s.t. } \forall i: \quad y_i \left(w^T x_i + w_0 \right) \geq 1 - \xi_i \\ &\quad \text{s.t. } \forall i: \quad \xi_i \geq 0 \end{aligned}$$

with $C \in \mathbb{R}_+$ controlling the tolerance to violations.

Solving for ξ

From:

$$\underset{w,w_0}{\operatorname{argmin}}\; w^T w + C\sum_{i=1}^{N}\xi_i$$

$$\text{s.t. } \forall i: \quad y_i\left(w^T x_i + w_0\right) \geq 1 - \xi_i$$

$$\text{s.t. } \forall i: \quad \xi_i \geq 0$$

... we can deduce:

$$\xi_i = \begin{cases} 0 & y_i\left(w^T x_i + w_0\right) \geq 1 \\ 1 - y_i\left(w^T x_i + w_0\right) & y_i\left(w^T x_i + w_0\right) < 1 \end{cases}$$

or rewritten equivalently:

$$\xi_i = \max\left(0, 1 - y_i\left(w^T x_i + w_0\right)\right)$$

Regularized Hinge Loss Optimization

Replacing ξ we get:

$$\underset{w,w_0}{\operatorname{argmin}}\; w^T w + C \sum_{i=1}^{N} \max\left(0, 1 - y_i \left(w^T x_i + w_0\right)\right)$$

Multiplying by $1/C$ and defining $\lambda = 1/C$:

$$\underset{w,w_0}{\operatorname{argmin}}\; \sum_{i=1}^{N} \max\left(0, 1 - y_i \left(w^T x_i + w_0\right)\right) + \lambda w^T w$$

Linear SVM as a Regularized Loss

- Model: $\hat{y}_i(w, w_0) = w^T x_i + w_0, f(x_i; w, w_0) = \text{sgn}\left(\hat{y}_i(w, w_0)\right)$

- Loss: $\mathcal{L}\left(y, \hat{y}(w, w_0)\right) = \max\left(0, 1 - y_i\left(w^T x_i + w_0\right)\right)$
- Regularization $\Omega\left(w\right) = \lambda w^T w = \lambda \sum_{m=1}^{M} w_m^2$

$$\underset{w, w_0}{\text{argmin}} \sum_{i=1}^{N} \mathcal{L}\left(y_i, w^T x_i + w_0\right) + \lambda \sum_{m=1}^{M} w_m^2$$

Can be solved with Stochastic Gradient Descent exactly like the Logistic Regression. However, using the sub-gradient of the loss:

$$\frac{d\mathcal{L}(y, \hat{y})}{d\hat{y}} = \frac{d \max(0, 1 - y\hat{y})}{d\hat{y}} = \begin{cases} 0 & y\hat{y} >= 1 \\ -y & y\hat{y} < 1 \end{cases}$$

Dual Optimization

- **Primal form:**
  Constrained optimization of $f(x)$ subject to K constraints $g_1(x) \leq 0, \ldots, g_K(x) \leq 0$:

$$\begin{aligned} \underset{x}{\operatorname{argmin}} \;\; & f(x) \\ \text{s.t. } \forall k : \;\; & g_k(x) \leq 0 \end{aligned}$$

- **Dual Form:**
  An equivalent and simpler form:

$$\begin{aligned} \underset{x}{\operatorname{argmin}} \, \underset{\alpha}{\operatorname{argmax}} \;\; & f(x) + \sum_{k=1}^{K} \alpha_k g_k(x) \\ \text{s.t. } \forall : \;\; & \alpha_k \geq 0 \end{aligned}$$

Primal and Dual SVM formulation

- **Primal SVM form**, notice a constant of $\frac{1}{2}$ is added:

$$\begin{aligned} \underset{w,w_0}{\operatorname{argmin}}\ & \frac{1}{2} w^T w \\ \text{s.t. } \forall i: \quad & y_i \left(w^T x_i + w_0 \right) \geq 1 \\ \text{or equivalently,} \forall i: \quad & -y_i \left(w^T x_i + w_0 \right) + 1 \leq 0 \end{aligned}$$

- **Dual SVM form**:

$$\begin{aligned} \underset{w,w_0}{\operatorname{argmin}}\ \underset{\alpha}{\operatorname{argmax}}\ & \frac{1}{2} w^T w - \sum_{i=1}^{N} \alpha_i \left(y_i \left(w^T x_i + w_0 \right) - 1 \right) \\ \text{s.t. } \forall i: \quad & \alpha_i \geq 0 \end{aligned}$$

Simplify the Dual form by solving for w

$$\underset{w,w_0}{\operatorname{argmin}}\ \underset{\alpha}{\operatorname{argmax}}\ \frac{1}{2}w^T w - \sum_{i=1}^{N} \alpha_i \left(y_i \left(w^T x_i + w_0\right) - 1\right)$$

$$\text{s.t. } \forall i: \quad \alpha_i \geq 0$$

Let $\mathcal{L}(w, w_0, \alpha) = \frac{1}{2}w^T w - \sum_{i=1}^{N} \alpha_i \left(\left(w^T x_i + w_0\right) y_i - 1\right)$. Then, solve for w, w_0 in terms of α (recall that derivatives at min $= 0$):

$$0 = \frac{\partial \mathcal{L}(w, w_0, \alpha)}{\partial w} = w - \sum_{i=1}^{N} \alpha_i x_i y_i \qquad w = \sum_{i=1}^{N} \alpha_i x_i y_i$$

$$0 = \frac{\partial \mathcal{L}(w, w_0, \alpha)}{\partial w_0} = \sum_{i=1}^{N} \alpha_i y_i \qquad 0 = \sum_{i=1}^{N} \alpha_i y_i$$

Dual SVM Objective

Plugging $w = \sum_{i=1}^{N} \alpha_i x_i y_i$ and setting $0 = \sum_{i=1}^{N} \alpha_i y_i$ to:

$$\underset{w,w_0}{\text{argmin}} \ \underset{\alpha}{\text{argmax}} \ \frac{1}{2} w^T w - \sum_{i=1}^{N} \alpha_i \left(y_i \left(w^T x_i + w_0 \right) - 1 \right)$$

$$\text{s.t. } \forall i : \quad \alpha_i \geq 0$$

yields:

$$\underset{\alpha}{\text{argmax}} \sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_i \alpha_j y_i y_j x_i^T x_j$$

$$\text{s.t. } \sum_{i=1}^{N} \alpha_i y_i = 0, \ \ \forall i : \ \alpha_i \geq 0$$

Dual Prediction Model

Remember the linear prediction model:

$$f(x, w, w_0) = \operatorname{sgn}\left(w^T x + w_0\right)$$

Plugging in $w = \sum_{i=1}^{N} \alpha_i x_i y_i$ leads to:

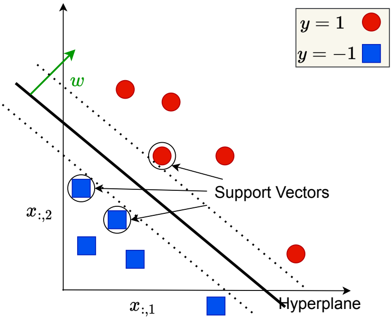
$$f(x, \alpha, w_0) = \operatorname{sgn}\left(\sum_{i=1}^{N} \alpha_i y_i x_i^T x + w_0\right)$$

Where w_0 is computed as:

$$\forall i : y_i \left(w^T x_i + w_0\right) = 1 \text{ leads to } \quad \forall i : w_0 = y_i - w^T x_i$$

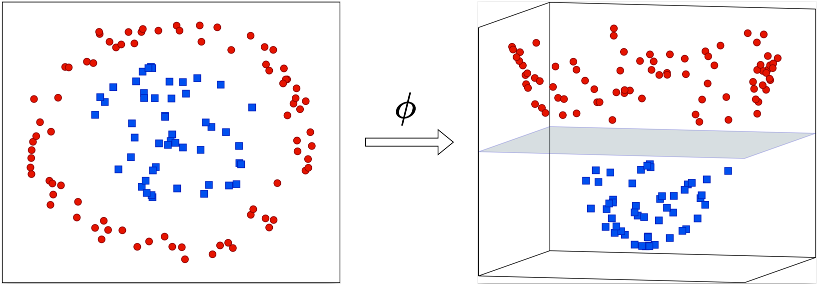
Only the instances with $\alpha_i > 0$ matter in the prediction. They are the "support" vectors/points.

Dual Prediction Model — Inference



In the dual formulation, to predict the label of a new instance, we only need to compute its similarity (dot product) with the support vectors.

Nonlinear Mapping



By applying a nonlinear mapping $\phi(x)$ to the data, we can make the data linearly separable in a higher dimensional space.

Nonlinear mapping in the optimization

Dual objective:

$$\underset{\alpha}{\text{argmax}} \sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_i \alpha_j y_i y_j (\phi(x_i)^T \phi(x_j))$$

Problem: the dot product $\phi(x_i)^T \phi(x_j)$ is expensive to compute for high dimensional features. **Solution:** kernel functions. A kernel function $K(x_i, x_j)$ is a function that computes the dot product in a higher dimensional space, without explicitly computing the mapping ϕ:

$$K(x_i, x_j) = \phi(x_i)^T \phi(x_j)$$

Replacing the dot product with a kernel function

Dual objective:

$$\underset{\alpha}{\operatorname{argmax}} \sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_i \alpha_j y_i y_j K(x_i, x_j)$$

$$\text{s.t.} \ \sum_{i=1}^{N} \alpha_i y_i = 0, \ \ \forall i : \ \alpha_i \geq 0$$

Dual prediction model:

$$f(x, w, w_0) = \operatorname{sgn}\left(\sum_{i=1}^{N} \alpha_i y_i K(x_i, x) + w_0\right)$$

The kernel creates a nonlinear classifier.

Kernels yield nonlinear models

$$f(x, w, w_0) = \text{sgn}\left(\sum_{i=1}^{N} \alpha_i y_i K(x_i, x) + w_0\right)$$

RBF $K(p,q) = e^{-\gamma(p-q)^2}$, polynomial $K(p,q) = \left(p^T q + c\right)^d$, etc.

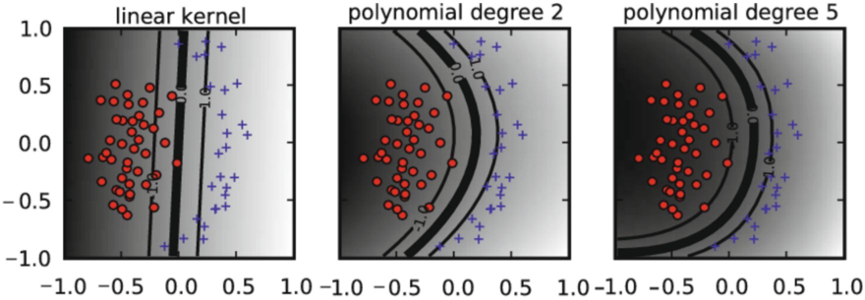


Image source: Asa Ben-Hur et al., 2010

Optimizing the Dual Form

How to find α?

Unfortunately, the optimization of the Dual SVM objective (with slack margins) is not covered in this course.

However, there exist many algorithms for solving the dual formulation. The classic approach is:

- Platt, Sequential Minimal Optimization: A Fast Algorithm for Training Support Vector Machines. [Link]