# Machine Learning

## Lecture 6 & 7: Constrained Optimization, SVM and Kernels

Prof. Dr. Stephan Günnemann

Data Mining and Analytics
Technical University of Munich

26.11.2018 & 4.12.2018

# Roadmap

1. Constraints and the Lagrangian
2. Projected Gradient Descent
3. Support Vector Machines (SVM)
4. Soft Margin Support Vector Machines
5. Kernels

Section 1

# Constraints and the Lagrangian

# Non-negative least squares

- In many important use cases negative weights would not be sensible, e.g. when fitting physical properties like resistance or density
- Hence, we need an additional constraint:

$$\begin{aligned} \text{minimize} \quad & E_{\mathrm{LS}}(\boldsymbol{w}) = \frac{1}{2}\sum_{i=1}^{N}(\boldsymbol{w}^T\boldsymbol{x}_i - y_i)^2 \\ \text{subject to} \quad & w_i \geq 0, \quad i \in [1, d]\,. \end{aligned}$$

- How do we solve this?

$$\underbrace{\boxed{-w_i}}_{f(\theta)} \leq 0$$

# Optimization with inequality constraints

$\sum x_i = 1$

$\Leftrightarrow \sum x_i - 1 = 0$

$f_i(\theta) = 0$

$\hookrightarrow f_i(\theta) \geq 0$

$f_i(\theta) \leq 0$

$\Leftrightarrow -f_i(\theta) \geq 0$

## Constrained optimization problem

Given $f_0 : \mathbb{R}^D \to \mathbb{R}$ and $f_i : \mathbb{R}^D \to \mathbb{R}$,

$$\begin{aligned} \text{minimize} \quad & f_0(\boldsymbol{\theta}) \\ \text{subject to} \quad & f_i(\boldsymbol{\theta}) \leq 0, \quad i = 1, \ldots, M\,. \end{aligned}$$

## Feasibility

A point $\boldsymbol{\theta} \in \mathbb{R}^D$ is called feasible if and only if it satisfies all constraints $f_i(\boldsymbol{\theta}) \leq 0, i = 1, \ldots, M$ of the optimization problem.

## Minimum and minimizer

We call the optimal value the minimum $p^*$, and the point where the minimum is obtained the minimizer $\boldsymbol{\theta}^*$. Thus $p^* = f_0(\boldsymbol{\theta}^*)$.

# Standard problems with inequality constraints (I)

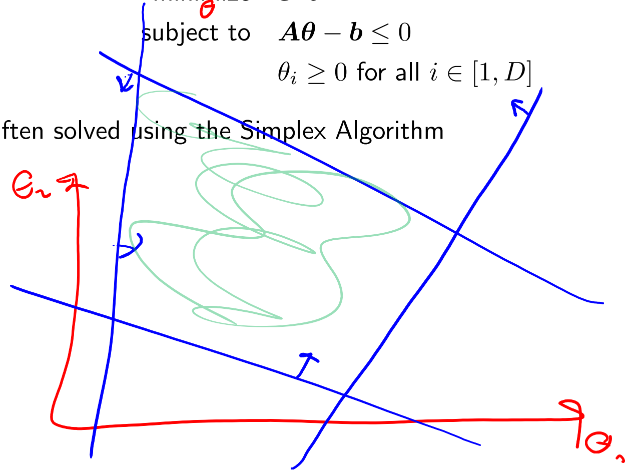
- Linear Programming (LP)

MAX $-c^T\theta$

$$\begin{aligned} \text{minimize} \quad & \boldsymbol{c}^T\boldsymbol{\theta} \\ \text{subject to} \quad & \boldsymbol{A}\boldsymbol{\theta} - \boldsymbol{b} \leq 0 \\ & \theta_i \geq 0 \text{ for all } i \in [1, D] \end{aligned}$$

$\theta$

Often solved using the Simplex Algorithm



# Standard problems with inequality constraints (II)

- Quadratic Programming (QP)

$$\begin{aligned} \underset{\theta}{\text{minimize}} \quad & \frac{1}{2}\boldsymbol{\theta}^T\boldsymbol{Q}\boldsymbol{\theta} + \boldsymbol{c}^T\boldsymbol{\theta} \\ \text{subject to} \quad & \boldsymbol{A}\boldsymbol{\theta} - \boldsymbol{b} \leq 0 \end{aligned}$$

  - If $\boldsymbol{Q}$ is positive semidefinite ⇒ convex
  - Non-negative least squares is an example of QP, with

$$\boldsymbol{\theta} = \boldsymbol{w}, \quad \boldsymbol{Q} = \boldsymbol{X}\boldsymbol{X}^T, \quad \boldsymbol{c} = \boldsymbol{X}\boldsymbol{y}, \quad \boldsymbol{A} = -\boldsymbol{I}_D, \quad \boldsymbol{b} = 0$$

- Semidefinite programming, conic optimization, ...

# Minimization with a single inequality constraint

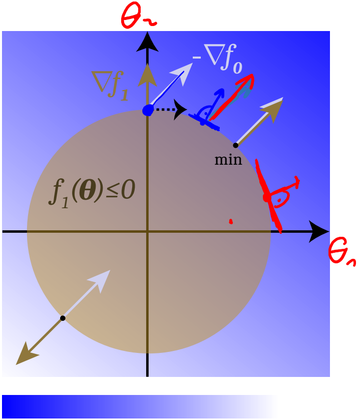
$$
\begin{array}{rl}
\text{minimize} & f_0(\boldsymbol{\theta}) = -(\theta_1 + \theta_2) \\
\text{subject to} & f_1(\boldsymbol{\theta}) = \theta_1^2 + \theta_2^2 - 1 \leq 0
\end{array}
$$

- If the minimum is in the interior of the circle ($f_1(\boldsymbol{\theta}^*) < 0$), we have reached it when $\boldsymbol{\nabla} f_0(\boldsymbol{\theta}^*) = 0$.
- If the minimum is on the circle ($f_1(\boldsymbol{\theta}^*) = 0$), we have reached it when the negative gradient $-\boldsymbol{\nabla} f_0(\boldsymbol{\theta}^*)$ has no component that we could follow without changing the value of $f_1$.

Thus at the minimizer $\boldsymbol{\theta}^*$ we have

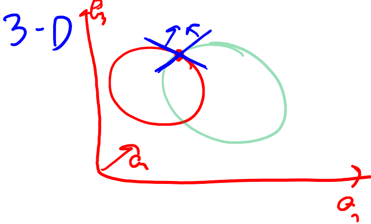
$$\boxed{-\boldsymbol{\nabla} f_0(\boldsymbol{\theta}^*) = \alpha\, \boldsymbol{\nabla} f_1(\boldsymbol{\theta}^*)}$$

with $\alpha \geq 0$ to ensure that $-\boldsymbol{\nabla} f_0$ and $\boldsymbol{\nabla} f_1$ do not point in opposite directions.



$f_0(\boldsymbol{\theta})$ is color coded.

# Multiple inequality constraints



Given $f_0 : \mathbb{R}^D \to \mathbb{R}$ and $f_i : \mathbb{R}^D \to \mathbb{R}$,

$$\begin{aligned} \underset{\theta}{\text{minimize}} \quad & f_0(\boldsymbol{\theta}) \\ \text{subject to} \quad & f_i(\boldsymbol{\theta}) \leq 0, \quad i = 1, \ldots, M\,. \end{aligned}$$

For multiple constraints $f_1, \ldots, f_p$ we have reached the minimum when the negative gradient $-\boldsymbol{\nabla} f_0$ has no component that we could follow without changing the value of any constraint.

This is the case when $-\boldsymbol{\nabla} f_0$ is a linear combination of the $\boldsymbol{\nabla} f_i$'s,

$$\boxed{-\boldsymbol{\nabla} f_0(\boldsymbol{\theta}^*) = \sum_{i=1}^{M} \alpha_i \, \boldsymbol{\nabla} f_i(\boldsymbol{\theta}^*)\,, \quad \alpha_i \geq 0}\,.$$

# Lagrangian

$$\begin{aligned} \text{minimize} \quad & f_0(\boldsymbol{\theta}) \\ \text{subject to} \quad & f_i(\boldsymbol{\theta}) \leq 0, \quad i = 1, \ldots, M \end{aligned}$$

## Definition (Lagrangian)

We define the Lagrangian $L : \mathbb{R}^D \times \mathbb{R}^M \to \mathbb{R}$ associated with the above problem as

$$\boxed{L(\boldsymbol{\theta}, \boldsymbol{\alpha}) = f_0(\boldsymbol{\theta}) + \sum_{i=1}^{M} \alpha_i f_i(\boldsymbol{\theta})}$$

We refer to $\alpha_i \geq 0$ as the Lagrange multiplier associated with the inequality constraint $f_i(\boldsymbol{\theta}) \leq 0$.

Calculating the gradient of $L$ w.r.t. $\boldsymbol{\theta}$ shows that our optimality criterion at $\boldsymbol{\theta}^*$ is recovered,

$$\boldsymbol{\nabla}_{\boldsymbol{\theta}^*} L(\boldsymbol{\theta}^*, \boldsymbol{\alpha}) = \boldsymbol{\nabla} f_0(\boldsymbol{\theta}^*) + \sum_{i=1}^{M} \alpha_i \boldsymbol{\nabla} f_i(\boldsymbol{\theta}^*) = 0 \,.$$

# Interpretation of the Lagrangian

$$\begin{aligned} \text{minimize} \quad & f_0(\boldsymbol{\theta}) \\ \text{subject to} \quad & f_i(\boldsymbol{\theta}) \le 0, \quad i = 1, \dots, M \end{aligned}$$

$$L(\boldsymbol{\theta}, \boldsymbol{\alpha}) = f_0(\boldsymbol{\theta}) + \overbrace{\sum_{i=1}^{M} \alpha_i f_i(\boldsymbol{\theta})}^{\substack{\text{if } \theta \in \chi \\ \le 0}}$$

For every choice of $\boldsymbol{\alpha}$, the corresponding $\min_{\boldsymbol{\theta} \in \mathbb{R}^D} L(\boldsymbol{\theta}, \boldsymbol{\alpha})$ is a lower bound on the optimal value of the constrained problem.

Watch `lagrangian.avi` for a demonstration.

$\forall \alpha \geq 0:$

$$\min_{\theta \in \mathbb{R}^D} L(\theta, \alpha) \le \min_{\theta \in \chi} L(\theta, \alpha) \le \min_{\theta \in \chi} f_0(\theta)$$

# Lagrange dual function $\forall \alpha \geq 0: \; g(\alpha) \leq \min_{\theta \in \mathcal{X}} f_0(\theta)$

### Definition (Lagrange dual function)

The Lagrange dual function $g : \mathbb{R}^M \to \mathbb{R}$ is the minimum of the Lagrangian over $\boldsymbol{\theta}$ given $\boldsymbol{\alpha}$,

$$\boxed{g(\boldsymbol{\alpha}) = \min_{\boldsymbol{\theta} \in \mathbb{R}^D} L(\boldsymbol{\theta}, \boldsymbol{\alpha}) = \min_{\boldsymbol{\theta} \in \mathbb{R}^D} \left( f_0(\boldsymbol{\theta}) + \sum_{i=1}^{M} \alpha_i f_i(\boldsymbol{\theta}) \right)}.$$

It is concave in $\boldsymbol{\alpha}$ since it is the point-wise minimum of a family of affine functions of $\boldsymbol{\alpha}$.

$\theta = 1 \qquad f_0(1) + \sum \alpha_i \cdot f_i(1)$

$h''(\Delta) = \min\{h(\Delta), h'(\Delta)\} \qquad \theta = 2 \qquad f_0(2) + \sum \alpha_i \cdot f_i(2)$

# Lagrange dual problem

For each $\boldsymbol{\alpha} \geq \mathbf{0}$ the Lagrange dual function $g(\boldsymbol{\alpha})$ gives us a lower bound on the optimal value $p^*$ of the original optimization problem.

What is the best (highest) lower bound?

## Lagrange dual problem

$$
\begin{aligned}
\underset{\alpha}{\text{maximize}} \quad & g(\boldsymbol{\alpha}) = \min_{\boldsymbol{\theta}} L(\boldsymbol{\theta}, \boldsymbol{\alpha}) \\
\text{subject to} \quad & \alpha_i \geq 0, \quad i = 1, \ldots, m
\end{aligned}
$$

The maximum $d^*$ of the Lagrange dual problem is the best lower bound on $p^*$ that we can achieve by using the Lagrangian.

# Dualities

$\theta \in \mathbb{R}^D$

$i = 1 \ldots m$ $\quad f_i(\theta) \leq 0$ $\quad \theta^* \downarrow f_0(\theta)$

$\alpha^* \uparrow g(\alpha)$

$\alpha \in \mathbb{R}^m$

## Weak duality (always)

Since $g(\boldsymbol{\alpha})$ is a lower bound of $p^*$ we have weak duality,

$$\underline{d^*} \leq \underline{p^*} \,.$$

The difference between the solution of the original and the dual problem,

$$p^* - d^* \geq 0 \,,$$

is called the duality gap.

## Strong duality (under certain conditions)

Under certain conditions we have

$$d^* = p^* \,,$$

i.e. the solution to the Lagrange dual problem is a solution of the original (primal) constrained optimization problem.

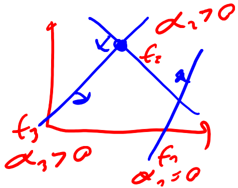
We then say that strong duality holds.

# Dual solution

$f_i(\theta) \le 0$

Let $\boldsymbol{\theta}^*$ be a minimizer of the primal problem and $\boldsymbol{\alpha}^*$ a maximizer of the dual problem. If strong duality holds, then

$$L(\boldsymbol{\theta}^*, \boldsymbol{\alpha}^*) = g(\boldsymbol{\alpha^*}) = \min_{\boldsymbol{\theta}} L(\boldsymbol{\theta}, \boldsymbol{\alpha}^*) \,.$$



## Proof.

Let $\tilde{\boldsymbol{\theta}}$ be a minimizer of $L(\boldsymbol{\theta}, \boldsymbol{\alpha}^*)$ over $\boldsymbol{\theta}$. We have

$$d^* = g(\boldsymbol{\alpha^*}) = \min_{\boldsymbol{\theta}} L(\boldsymbol{\theta}, \boldsymbol{\alpha}^*) = L(\tilde{\boldsymbol{\theta}}, \boldsymbol{\alpha^*}) \qquad \text{(minimizer)}$$

$$= f_0(\tilde{\boldsymbol{\theta}}) + \sum_{i=1}^{M} \alpha_i^* f_i(\tilde{\boldsymbol{\theta}}) \qquad \text{(Lagrangian)}$$

$$\le f_0(\boldsymbol{\theta}^*) \boxed{+ \sum_{i=1}^{M} \alpha_i^* f_i(\boldsymbol{\theta}^*)} \qquad (\tilde{\boldsymbol{\theta}} \text{ minimizer over } \boldsymbol{\theta})$$

$\le 0$

$= 0$

$$\le f_0(\boldsymbol{\theta}^*) = p^* \qquad (\boldsymbol{\alpha} \ge \mathbf{0};\ \ \boldsymbol{\theta}^* \text{ feasible: } f_i(\boldsymbol{\theta}^*) \le 0)$$

and since $d^* = p^*$, we have $g(\boldsymbol{\alpha}^*) = L(\tilde{\boldsymbol{\theta}}, \boldsymbol{\alpha}^*) = L(\boldsymbol{\theta}^*, \boldsymbol{\alpha}^*)$. □

# Dual solution

$$\alpha^* = \arg\max_{\alpha_i \geq 0} g(\alpha)$$

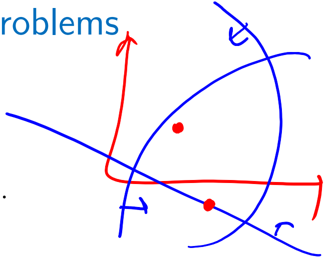
$\alpha_n$
$\alpha_1$

## Corollary

*Let $\boldsymbol{\theta}^*$ be a minimizer of the primal problem and $\boldsymbol{\alpha}^*$ a maximizer of the dual problem. If strong duality holds and $L(\boldsymbol{\theta}, \boldsymbol{\alpha})$ is convex in $\boldsymbol{\theta}$, then*

$$\boldsymbol{\theta}^* = \arg\min_{\boldsymbol{\theta} \in \mathbb{R}^D} L(\boldsymbol{\theta}, \boldsymbol{\alpha}^*)\,.$$

# Constraint qualifications for convex problems



Consider the constrained optimization problem,

$$\begin{aligned} \text{minimize} \quad & f_0(\boldsymbol{\theta}) \\ \text{subject to} \quad & f_i(\boldsymbol{\theta}) \leq 0\,. \end{aligned}$$

## Slater's constraint qualification

The duality gap is zero (i.e. strong duality holds) if $f_0, f_1, \ldots, f_M$ are convex and there exists a $\boldsymbol{\theta} \in \mathbb{R}^D$ such that for each constraint $f_i$, $i = 1, \ldots, M$ it holds that (a)

$$f_i(\boldsymbol{\theta}) \lneq 0$$

$f_i(\theta) < 0$

$\Rightarrow \exists$ PROPER INTERIOR POINT

or (b) the constraint is affine, that is

$$f_i(\boldsymbol{\theta}) = \boldsymbol{w}_{\boldsymbol{i}}^T \boldsymbol{\theta} + b_i \leq 0\,.$$

For a proof see Boyd p. 234.

# Recipe for solving constrained optimization problems

The constrained optimization problem,

$$\begin{aligned} \text{minimize} \quad & f_0(\boldsymbol{\theta}) \\ \text{subject to} \quad & f_i(\boldsymbol{\theta}) \leq 0 \quad i = 1, \dots, M\,, \end{aligned}$$

with $f_0$ convex and $f_1, \dots, f_M$ convex can be solved as follows:

1. Calculate the Lagrangian

$$L(\boldsymbol{\theta}, \boldsymbol{\alpha}) = f_0(\overset{\theta^*(\alpha)}{\boldsymbol{\theta}}) + \sum_{i=1}^{M} \alpha_i f_i(\overset{\theta^*(\alpha)}{\boldsymbol{\theta}})\,.$$

L is convex in $\theta$

2. Obtain the Lagrange dual function $g(\boldsymbol{\alpha})$ by solving

$$\boldsymbol{\theta}^*(\boldsymbol{\alpha}) = \arg \underbrace{\boxed{\min_{\boldsymbol{\theta}} L(\boldsymbol{\theta}, \boldsymbol{\alpha})}}_{g(\alpha)} \overset{\text{convex}}{\iff} \boldsymbol{\nabla}_{\boldsymbol{\theta}} L(\boldsymbol{\theta}, \boldsymbol{\alpha}) = \mathbf{0}\,.$$

3. Solve the dual problem

$$\begin{aligned} \text{maximize} \quad & g(\boldsymbol{\alpha}) = L(\boldsymbol{\theta}^*(\boldsymbol{\alpha}), \boldsymbol{\alpha}) \\ \text{subject to} \quad & \alpha_i \geq 0 \quad i = 1, \dots, M\,. \end{aligned}$$

The solution of this problem is also the solution of the original problem if Slater's condition is satisfied.

# KKT conditions

If we have the following optimization problem

$$\begin{aligned} \text{minimize} \quad & f_0(\boldsymbol{\theta}) \\ \text{subject to} \quad & f_i(\boldsymbol{\theta}) \leq 0\,, \end{aligned}$$

with $f_0$ convex and $f_1, \dots, f_M$ convex and strong duality holds, the following statement is true:

## KKT conditions

$\boldsymbol{\theta}^*$ and $\boldsymbol{\alpha}^*$ are the optimal solutions of the constrained optimization problem and the corresponding Lagrange dual problem if and only if they satisfy the Karush-Kuhn-Tucker (KKT) conditions:

$$\begin{aligned} f_i(\boldsymbol{\theta}^*) &\leq 0 && \text{primal feasibility,} \quad \checkmark \\ \alpha_i^* &\geq 0 && \text{dual feasibility,} \quad \checkmark \\ \alpha_i^* f_i(\boldsymbol{\theta}^*) &= 0 && \text{complementary slackness,} \quad \checkmark \\ \boldsymbol{\nabla}_{\boldsymbol{\theta}} L(\boldsymbol{\theta}^*, \boldsymbol{\alpha}^*) &= 0 && \boldsymbol{\theta}^* \text{ minimizes Lagrangian,} \quad \checkmark \end{aligned}$$

for $i = 1, \dots, M$.

# Summary

- Lagrange formalism reformulates constrained optimization problem into a dual problem.
- For convex objective and constraints, solving the dual problem allows to solve the primal problem under rather weak conditions (duality gap is zero).
- KKT conditions characterize the solution and sometimes allow to obtain it without minimization.
- Convexity is necessary to guarantee optimality.

# Section 2

## Projected Gradient Descent

# Motivation

- Most of the time we use Gradient Descent to solve optimization problems
- Can we apply Gradient Descent to solve constrained problems?

# Constrained Optimization with Gradient Descent

$\theta_{t+1} \leftarrow \theta_t$
$- \tau \cdot \nabla$

Why can't we just apply Gradient Descent for constrained optimization?

- Assume parameters are restricted to convex set $\mathcal{X}$ (the domain) $\Rightarrow$ Solution after gradient step might be outside of region $\mathcal{X}$
- Even stronger: If current solution $\boldsymbol{\theta}^t$ is at the "border" of $\mathcal{X}$, each move along the gradient might lead to an infeasible solution:

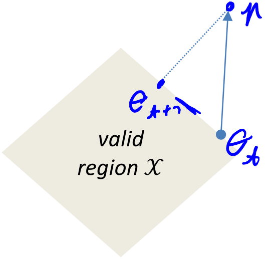
$$\boldsymbol{\theta}^t \in \mathcal{X}, \text{ but } \boldsymbol{\theta}^{t+1} = \boldsymbol{\theta}^t - \tau \boldsymbol{\nabla} f(\boldsymbol{\theta}^t) \notin \mathcal{X} \text{ for all } \tau > 0$$

**Idea**: Project new point back on the convex set $\mathcal{X}$

$$\boldsymbol{\theta}^{t+1} \leftarrow \pi_{\mathcal{X}}(\boldsymbol{\theta}^t - \tau \boldsymbol{\nabla} f(\boldsymbol{\theta}^t))$$

Here:

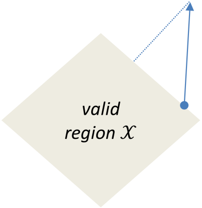
$$\pi_{\mathcal{X}}(\boldsymbol{p}) = \underset{\boldsymbol{\theta} \in \mathcal{X}}{\arg\min} \|\boldsymbol{\theta} - \boldsymbol{p}\|^2$$



# Projected Gradient Descent

$\min f_0(\theta)$
$\theta \in \mathcal{X}$

- Projection itself is a convex optimization problem



$$\pi_{\mathcal{X}}(\boldsymbol{p}) = \arg\min_{\boldsymbol{\theta} \in \mathcal{X}} \|\boldsymbol{\theta} - \boldsymbol{p}\|^2$$

- If all constraints are linear ⇒ quadratic program
  - Possibility to use standard solvers
  - Still quite costly; projection has to be done after each gradient step

# Efficient Projections (I)

Goal:

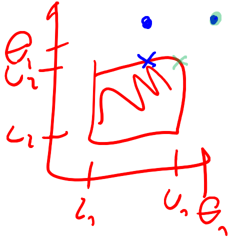
$$\pi_{\mathcal{X}}(\boldsymbol{p}) = \arg\min_{\boldsymbol{\theta}\in\mathcal{X}} \|\boldsymbol{\theta} - \boldsymbol{p}\|^2$$

$l_i$ $u_i$ $:$

Some (frequently observed) cases can be solved efficiently:

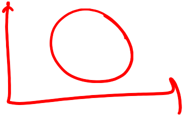
- Projection onto box-constraints $\mathcal{X} = \left\{\boldsymbol{\theta} \in \mathbb{R}^d \,\middle|\, \forall i \in [1,d] : l_i \leq \theta_i \leq u_i\right\}$

$$\pi_{\mathcal{X}}(\boldsymbol{p}) = \min(\max(l_i, p_i), u_i)$$



- Projection onto $L_2$-ball $\mathcal{X} = \left\{\boldsymbol{\theta} \in \mathbb{R}^d \,\middle|\, \|\boldsymbol{\theta}\|_2 \leq c\right\}$

$$\pi_{\mathcal{X}}(\boldsymbol{p}) = \begin{cases} \boldsymbol{p} & \text{if } \|\boldsymbol{p}\| \leq c, \\ \frac{c}{\|\boldsymbol{p}\|_2}\boldsymbol{p} & \text{otherwise} \end{cases}$$



# Efficient Projections (II)

Goal:

$$\pi_{\mathcal{X}}(\boldsymbol{p}) = \underset{\boldsymbol{\theta} \in \mathcal{X}}{\arg\min} \|\boldsymbol{\theta} - \boldsymbol{p}\|^2$$

Some (frequently observed) cases can be solved efficiently:

- Projection onto $L_1$-ball $\mathcal{X} = \left\{\boldsymbol{\theta} \in \mathbb{R}^d \,\middle|\, \|\boldsymbol{\theta}\|_1 \leq c\right\}$

  → Linear time algorithms ("Projection onto an $L_1$-norm Ball with Application to Identification of Sparse Autoregressive Models" by Jitkomut Songsiri)

- Projection onto $L_1$-ball and box-constraints $\mathcal{X} = \left\{\boldsymbol{\theta} \in \mathbb{R}^d \,\middle|\, \|\boldsymbol{\theta}\|_1 \leq c \text{ and } \forall i \in [1, d] : l_i \leq \theta_i \leq u_i\right\}$

  → Linear time algorithms ("$L_1$ Projections with Box Constraints" by Mithun Das Gupta et al.)

# Discussion: Projected GD

- Often used for solving (large-scale) constrained optimization problems
- Highly efficient if projection can be evaluated efficiently
- Each step leads to a feasible solution

  (i.e. at each step the solution remains inside the valid domain – there also exist methods that step outside the feasible region)
- Like GD, choice of learning rate etc. required
- *Note:* Projected gradient descent is a special case of so called proximal methods

Section 3

# Support Vector Machines (SVM)

# Reading material

## Reading material

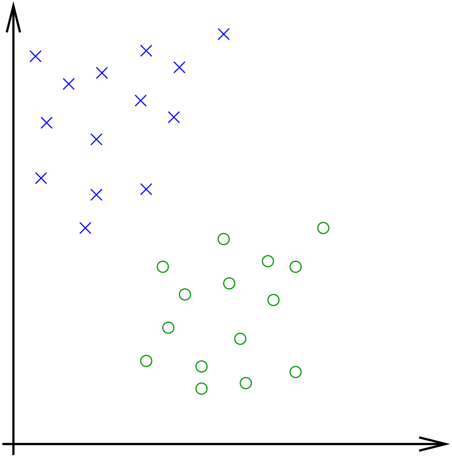
- Bishop: chapters 7.1.0, 7.1.1, 7.1.2

## Acknowledgements

- Slides are based on an older version by S. Urban

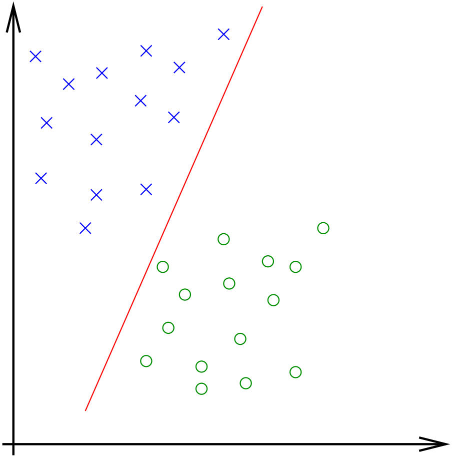
# Recall: Linear classification with a hyperplane

- Binary classification problem:
  × are in class 1
  ○ are in class -1



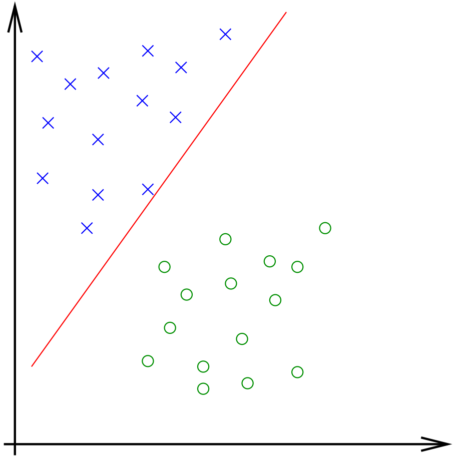
# Recall: Linear classification with a hyperplane

- Binary classification problem:
  × are in class 1
  ○ are in class -1
- Linear classifier:
  Find the optimal decision hyperplane.



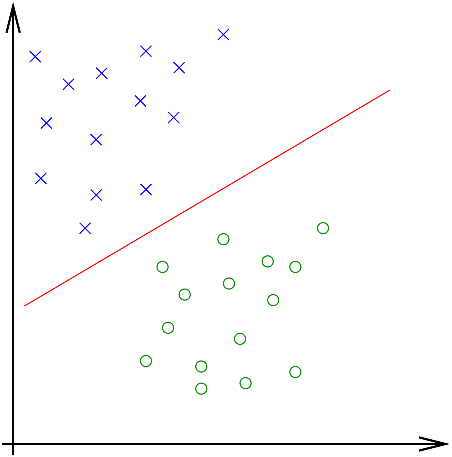
# Recall: Linear classification with a hyperplane

- Binary classification problem:
  × are in class 1
  ○ are in class -1
- Linear classifier:
  Find the optimal decision hyperplane.



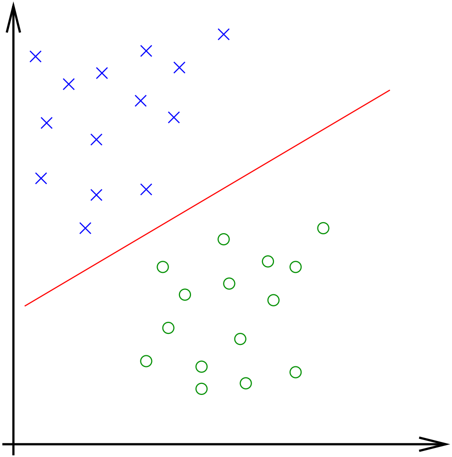
# Recall: Linear classification with a hyperplane

- Binary classification problem:
  × are in class 1
  ○ are in class -1
- Linear classifier:
  Find the optimal decision hyperplane.
- There are infinitely many solutions. Problem is not well-defined.



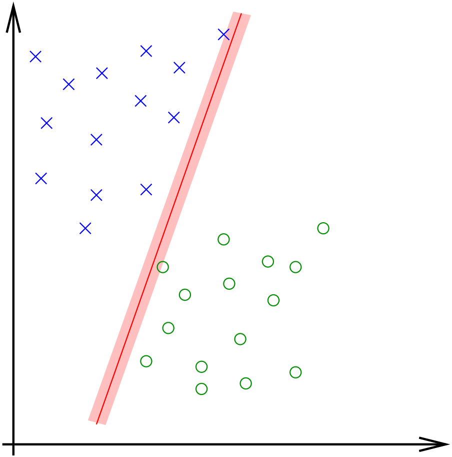
# Recall: Linear classification with a hyperplane

- Binary classification problem:
  × are in class 1
  ○ are in class -1
- Linear classifier:
  Find the optimal decision hyperplane.
- There are infinitely many solutions. Problem is not well-defined.
- A better objective:
  Find a separating hyperplane, such that subsequent points are classified correctly.



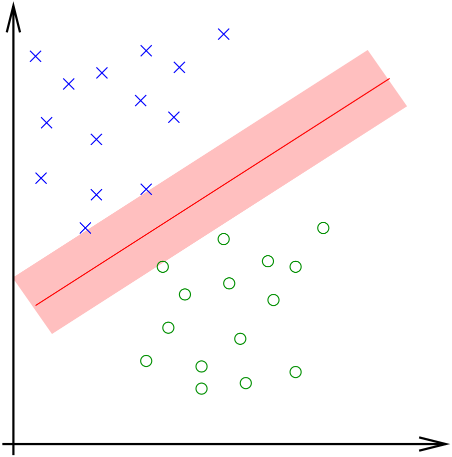
# Maximum margin classifier

- Intuitively, a wide margin around the dividing line makes it more likely that new samples will fall on the right side of the boundary.



# Maximum margin classifier

- Intuitively, a wide margin around the dividing line makes it more likely that new samples will fall on the right side of the boundary.
- Actual rigorous motivation comes from Statistical Learning Theory [1]
- Objective:
  Find a hyperplane that separates both classes with the maximum margin.



[1] V. Vapnik - "Statistical Learning Theory", 1995

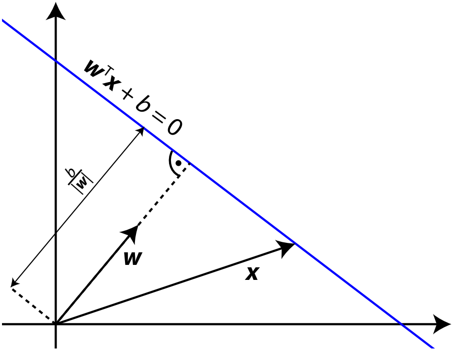
# Hyperplanes in Hessian normal form

A hyperplane is defined by a set of all points $\boldsymbol{x} \in \mathbb{R}^D$ that satisfy

$$\boldsymbol{w}^T \boldsymbol{x} + b = 0 \,.$$

Explanation:

- $\boldsymbol{w}$ is a normal vector of the hyperplane and $b$ defines the offset.
- $\boldsymbol{w}^T \boldsymbol{x}$ is the length of the projection of $\boldsymbol{x}$ on $\boldsymbol{w}$ in units (multiples) of $||\boldsymbol{w}||$.


$\boldsymbol{w}^T\boldsymbol{x} + b = 0$
$\frac{b}{||\boldsymbol{w}||}$
$\boldsymbol{w}$
$\boldsymbol{x}$

# Hyperplanes in Hessian normal form

For a point $\boldsymbol{x_1}$ that lies on the far side of the hyperplane we have

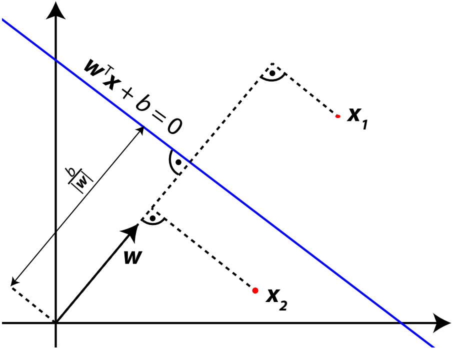
$$\boldsymbol{w}^T \boldsymbol{x_1} + b > 0$$

because the projection of $\boldsymbol{x_1}$ on $\boldsymbol{w}$ is *larger* than for points $\boldsymbol{x}$ that are on the hyperplane.

Conversely for a point $\boldsymbol{x_2}$ that lies on the near side of the hyperplane we have

$$\boldsymbol{w}^T \boldsymbol{x_2} + b < 0$$

because the projection of $\boldsymbol{x_2}$ on $\boldsymbol{w}$ is *smaller* than for points $\boldsymbol{x}$ that are on the hyperplane.



$\boldsymbol{w}^T\boldsymbol{x}+b=0$
$\boldsymbol{x}_1$
$\frac{b}{\|\boldsymbol{w}\|}$
$\boldsymbol{w}$
$\boldsymbol{x}_2$

# Linear classifier

We can use this to build a linear classifier by assigning all $\boldsymbol{x}$ with

$$\boldsymbol{w}^T\boldsymbol{x} + b > 0$$

to class blue and all $\boldsymbol{x}$ with

$$\boldsymbol{w}^T\boldsymbol{x} + b < 0$$

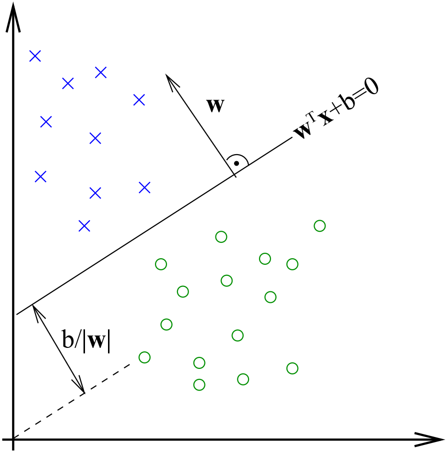
to class green.

Thus the class of $\boldsymbol{x}$ is given by

$$h(\boldsymbol{x}) = \operatorname{sgn}(\boldsymbol{w}^T\boldsymbol{x} + b)$$

with

$$\operatorname{sgn}(z) = \begin{cases} -1 & \text{if } z < 0 \\ 0 & \text{if } z = 0 \\ +1 & \text{if } z > 0 \end{cases} .$$



# Linear classifier with margin

We add two more hyperplanes that are parallel to the original hyperplane and require that no training points must lie between those hyperplanes.

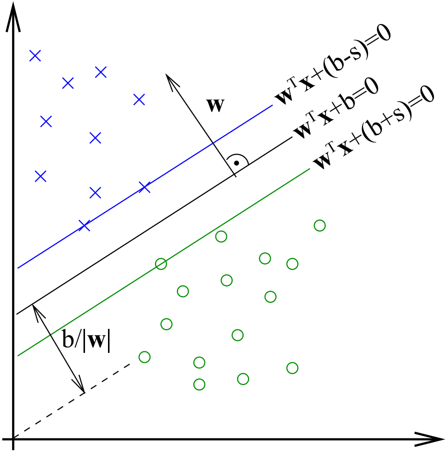
Thus we now require

$$\boldsymbol{w}^T \boldsymbol{x} + (b - s) > 0$$

for all $\boldsymbol{x}$ from class blue and

$$\boldsymbol{w}^T \boldsymbol{x} + (b + s) < 0$$

for all $\boldsymbol{x}$ from class green.


$\mathbf{w}^T\mathbf{x}+(b\text{-}s)=0$
$\mathbf{w}$
$\mathbf{w}^T\mathbf{x}+b=0$
$\mathbf{w}^T\mathbf{x}+(b+s)=0$
$b/|\mathbf{w}|$

# Size of the margin

Signed distance from the origin to the hyperplane is given by

$$d = -\frac{b}{||\boldsymbol{w}||} \,.$$

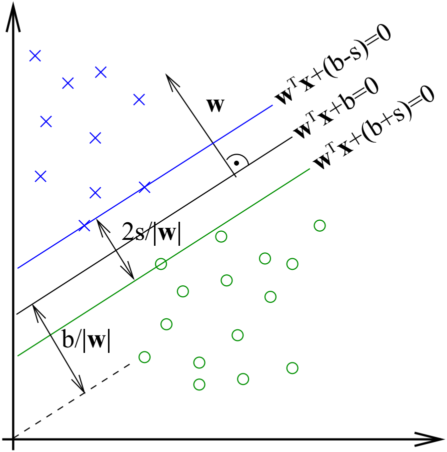
Thus we have

$$d_{blue} = -\frac{b-s}{||\boldsymbol{w}||}$$

$$d_{green} = -\frac{b+s}{||\boldsymbol{w}||}$$

and the margin is

$$m = d_{blue} - d_{green} = \frac{2s}{||\boldsymbol{w}||} \,.$$



$\mathbf{w}^T\mathbf{x}$+(b-s)=0
$\mathbf{w}^T\mathbf{x}$+b=0
$\mathbf{w}^T\mathbf{x}$+(b+s)=0
$\mathbf{w}$
2s/|$\mathbf{w}$|
b/|$\mathbf{w}$|

# Redundancy of parameter $s$

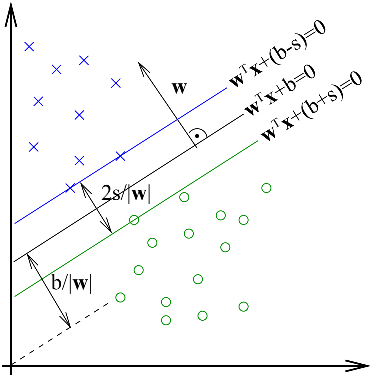
1) $||w|| = 1$

✓ 2) $s = 1$

The size of the margin,

$$m = \frac{2s}{||\boldsymbol{w}||}$$

only depends on the ratio, so w.l.o.g. we can set $s = 1$ and get



$\mathbf{w}^T\mathbf{x}+(b-s)=0$
$\mathbf{w}^T\mathbf{x}+b=0$
$\mathbf{w}^T\mathbf{x}+(b+s)=0$
$\mathbf{w}$
$2s/|\mathbf{w}|$
$b/|\mathbf{w}|$

# Redundancy of parameter $s$

The size of the margin,

$$m = \frac{2s}{||\boldsymbol{w}||}$$

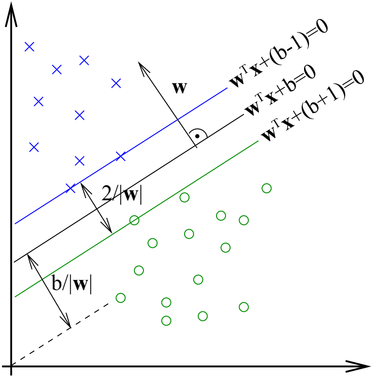
only depends on the ratio, so w.l.o.g. we can set $s = 1$ and get

$$\boxed{m = \frac{2}{||\boldsymbol{w}||}}$$

Although the distance from the origin to the black plane,

$$d = -\frac{b}{||\boldsymbol{w}||} ,$$

also depends on two parameters we *cannot* set $b = 1$ as this would link the distance $d$ to the size of the margin $m$.


$\mathbf{w}^T\mathbf{x}+(b-1)=0$
$\mathbf{w}^T\mathbf{x}+b=0$
$\mathbf{w}^T\mathbf{x}+(b+1)=0$
$\mathbf{w}$
$2/|\mathbf{w}|$
$b/|\mathbf{w}|$

# Set of constraints

Let $\boldsymbol{x}_i$ be the $i$th sample, and
$y_i \in \{-1, 1\}$ the class assigned to $\boldsymbol{x}_i$.

The constraints

s=1

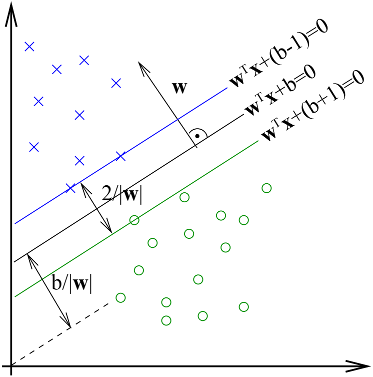
$$\boldsymbol{w}^T \boldsymbol{x}_i + b \geq +1 \quad \text{for } y_i = +1\,,$$
$$\boldsymbol{w}^T \boldsymbol{x}_i + b \leq -1 \quad \text{for } y_i = -1$$

can be condensed into

$$y_i(\boldsymbol{w}^T \boldsymbol{x}_i + b) \geq 1 \quad \text{for all } i\,.$$

If these constraints are fulfilled the margin is

$$m = \frac{2}{||\boldsymbol{w}||} = \frac{2}{\sqrt{\boldsymbol{w}^T \boldsymbol{w}}}\,.$$



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

$$\text{s.t. } y_i \cdot (w^T x_i + b) \geq 1 \quad \forall i$$

# Optimization problem

Let $\boldsymbol{x}_i$ be the $i$th data point, $i = 1, \dots, N$, and $y_i \in \{-1, 1\}$ the class assigned to $\boldsymbol{x}_i$.

To find the separating hyperplane with the maximum margin we need to find $\{\boldsymbol{w}, b\}$ that

$$\begin{aligned} \text{minimize} \quad & f_0(\boldsymbol{w}, b) = \frac{1}{2}\boldsymbol{w}^T\boldsymbol{w} \\ \text{subject to} \quad & f_i(\boldsymbol{w}, b) = y_i(\boldsymbol{w}^T\boldsymbol{x}_i + b) - 1 \geq 0 \quad \text{for } i = 1, \dots, N\,. \end{aligned}$$

This is a constrained convex optimization problem (more specifically, quadratic programming problem).

We go from $||\boldsymbol{w}||$ to $||\boldsymbol{w}||^2 = \boldsymbol{w}^T\boldsymbol{w}$ as square root is a monotonic function that doesn't change the location of the optimum.

# Primal problem

$$\frac{1}{2}\cdot\left[\sum \alpha_i y_i x_i\right]^T \cdot \left[\sum_j \alpha_j \cdot y_j \cdot x_j\right]$$

We apply our recipe for solving the constrained optimization problem.

1. Calculate the Lagrangian

$f_0(w)$

$$L(\boldsymbol{w}, b, \boldsymbol{\alpha}) = \boxed{\frac{1}{2}\boldsymbol{w}^T\boldsymbol{w}} - \underbrace{\sum_{i=1}^{N} \alpha_i [y_i(\boldsymbol{w}^T\boldsymbol{x}_i + b) - 1]}_{\sum_i \left(\alpha_i y_i \cdot \left[\sum_j \alpha_j \cdot y_j \cdot x_j\right] \cdot x_i\right) + \sum \alpha_i \cdot y_i \cdot b^* - \sum \alpha_i} .$$

2. Minimize $L(\boldsymbol{w}, b, \boldsymbol{\alpha})$ w.r.t. $\boldsymbol{w}$ and $b$.

$w^*(\alpha)$

$b^*(\alpha)$

$$\nabla_{\boldsymbol{w}} L(\boldsymbol{w}, b, \boldsymbol{\alpha}) = \boldsymbol{w} - \sum_{i=1}^{N} \alpha_i y_i \boldsymbol{x_i} \overset{!}{=} 0$$

$$\frac{\partial L}{\partial b} = \sum_{i=1}^{N} \alpha_i y_i \overset{!}{=} 0$$

Thus the weights are a linear combination of the training samples,

$$w^*(\alpha) = \boldsymbol{w} = \sum_{i=1}^{N} \alpha_i y_i \boldsymbol{x_i} .$$

$b^*(\alpha) = ?$

# Dual problem

Substituting both relations back into $L(\boldsymbol{w}, b, \boldsymbol{\alpha})$ gives the Lagrange dual function $g(\boldsymbol{\alpha})$.

Thus we have reformulated our original problem as

$$
\begin{aligned}
\underset{\alpha}{\text{maximize}} \quad & g(\boldsymbol{\alpha}) = \sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} y_i y_j \alpha_i \alpha_j \boldsymbol{x_i}^T \boldsymbol{x_j} \\
\text{subject to} \quad & \sum_{i=1}^{N} \alpha_i y_i = 0 \\
& \alpha_i \geq 0, \qquad \text{for} \quad i = 1, \ldots, N \,.
\end{aligned}
$$

3. Solve this problem.

# Solving the dual problem

We can rewrite the dual function $g(\boldsymbol{\alpha})$ in vector form

$$g(\boldsymbol{\alpha}) = \frac{1}{2}\boldsymbol{\alpha}^T \boldsymbol{Q} \boldsymbol{\alpha} + \boldsymbol{\alpha}^T \mathbf{1}_N$$

where $\boldsymbol{Q}$ is a symmetric negative (semi-)definite matrix, and the constraints on $\boldsymbol{\alpha}$ are linear.

This is an instance of a quadratic programming problem.
There exist efficient algorithms for its solution, such as Sequential minimal optimization (SMO) [2].

A number of implementations, such as `LIBSVM` [3] are available and are widely used in practice.

[2] `http://cs229.stanford.edu/materials/smo.pdf`

[3] C.-C. Chang and C.-J. Lin. *LIBSVM : a library for support vector machines*, 2011

# Recovering $\boldsymbol{w}$ and $b$ from the dual solution $\boldsymbol{\alpha}^*$

Having obtained the optimal $\boldsymbol{\alpha}^*$ using our favorite QP solver, we can compute the parameters defining the separating hyperplane.

Recall, that from the optimality condition, the weights $\boldsymbol{w}$ are a linear combination of the training samples,

$$w^*(\alpha) = \boldsymbol{w} = \sum_{i=1}^{N} \alpha_i^* y_i \boldsymbol{x_i} = w^*(\alpha^*)$$

From the complementary slackness condition $\alpha_i^* f_i(\boldsymbol{x}) = 0$ we can easily recover the bias.

When we take any vector $\boldsymbol{x}_i$ for which $\alpha_i \neq 0$. The corresponding constraint $f_i(\boldsymbol{w}, b)$ must be zero and thus we have

$$\boldsymbol{w}^T \boldsymbol{x_i} + b = y_i \, .$$

Solving this for $b$ yields the bias

$$b = y_i - \boldsymbol{w}^T \boldsymbol{x_i}$$

We can also average the $b$ over all support vectors to get a more stable solution.

# Support vectors

Let's look closer at the complimentary slackness condition

$$\alpha_i f_i(\boldsymbol{x}^*) = 0\,.$$

In our case this means

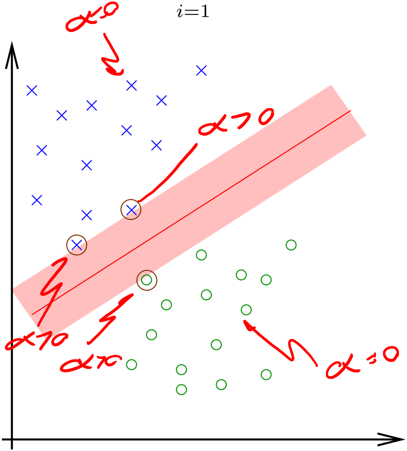
$$\alpha_i[y_i(\boldsymbol{w}^T\boldsymbol{x}_i + b) - 1] = 0 \qquad \text{for all } i\,.$$

Hence a training sample $\boldsymbol{x_i}$ can only contribute to the weight vector ($\alpha_i \neq 0$) if it lies on the margin, that is

$$y_i(\boldsymbol{w}^T\boldsymbol{x}_i + b) = 1\,.$$

A training sample $\boldsymbol{x_i}$ with $\alpha_i \neq 0$ is called a support vector.

$$\boldsymbol{w} = \sum_{i=1}^{N} \alpha_i y_i \boldsymbol{x_i}$$



# Classifying

The class of $\boldsymbol{x}$ is given by

$$h(\boldsymbol{x}) = \operatorname{sgn}(\boldsymbol{w}^T\boldsymbol{x} + b)\,.$$

Substituting

$$\boldsymbol{w} = \sum_{i=1}^{N} \alpha_i y_i \boldsymbol{x_i}$$

gives

MOST ARE 0

$$h(\boldsymbol{x}) = \operatorname{sgn}\left(\sum_{i=1}^{N} \alpha_i y_i \boldsymbol{x_i}^T \boldsymbol{x} + b\right)\,.$$

Since the solution is sparse (most $\alpha_i$s are zero) we only need to remember the few training samples $\boldsymbol{x_i}$ with $\alpha_i \neq 0$.