# ET4350: Applied Convex Optimization

Delft University of Technology

# Course Information

- Instructors:
  - Prof. Geert Leus
  - Prof. Borbala Hunyadi (Bori)
  - MSc. Alberto Natali
- Class schedule (watch out for any changes on TU Delft roosters):
  - Wednesdays between 10.45-12.30
  - Fridays between 10.45-12.30

# Course Information

- Book(s) are freely available online
  - Stephen Boyd and Lieven Vandenberghe, "Convex Optimization", Cambridge University Press, 2004.
  - Slides/lecture notes for subgradient methods.
- Assessment
  - Open-book written exam.
  - Compulsory lab assignment worth 1 EC (20%); report and short presentation. Enroll via Brightspace.
- Course information:
  - http://ens.ewi.tudelft.nl/Education/courses/ee4530/index.php

# Mathematical optimization

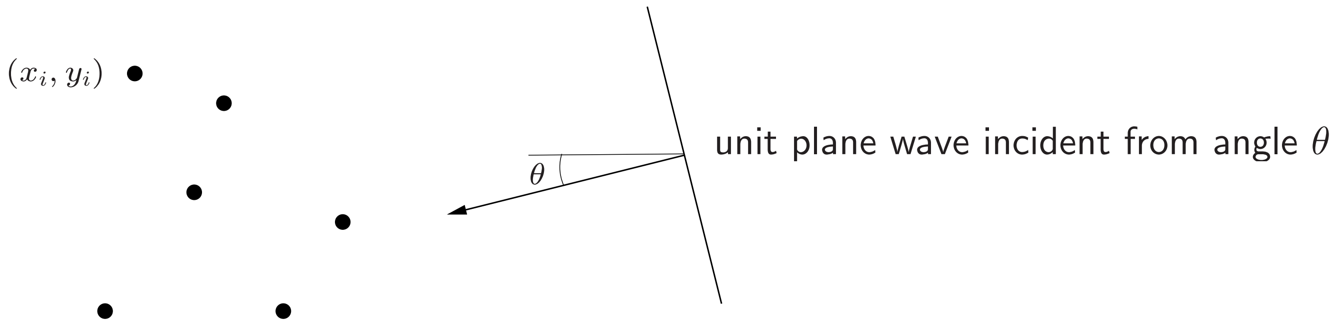
**(mathematical) optimization problem**

$$\begin{array}{ll} \text{minimize} & f_0(x) \\ \text{subject to} & f_i(x) \leq b_i, \quad i = 1, \ldots, m \end{array}$$

- $x = (x_1, \ldots, x_n)$: optimization variables
- $f_0 : \mathbf{R}^n \to \mathbf{R}$: objective function
- $f_i : \mathbf{R}^n \to \mathbf{R}$, $i = 1, \ldots, m$: constraint functions

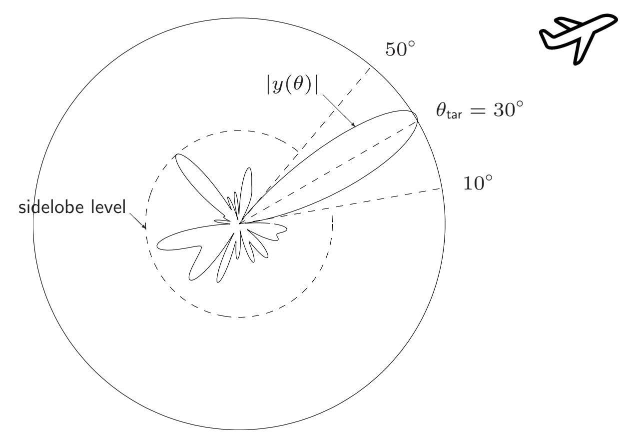
**optimal solution** $x^\star$ has smallest value of $f_0$ among all vectors that satisfy the constraints

# Array processing




- omnidirectional antenna elements at positions $(x_1, y_1)$, . . . , $(x_n, y_n)$
- linearly combine with complex weights $w_i$:

$$y(\theta) = \sum_{i=1}^{n} w_i e^{j(x_i \cos\theta + y_i \sin\theta)}$$



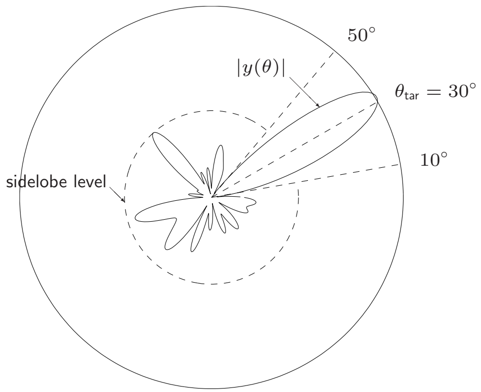

- $y(\theta)$ is (complex) *antenna array gain pattern*
- $|y(\theta)|$ gives sensitivity of array as function of incident angle $\theta$
- depends on design variables **Re** $w$, **Im** $w$
(called *antenna array weights* or *shading coefficients*)

**design problem:** choose $w$ to achieve desired gain pattern

## Sidelobe level minimization

make $|y(\theta)|$ small for $|\theta - \theta_{\text{tar}}| > \alpha$

($\theta_{\text{tar}}$: target direction; $2\alpha$: beamwidth)




**via least-squares** (discretize angles)

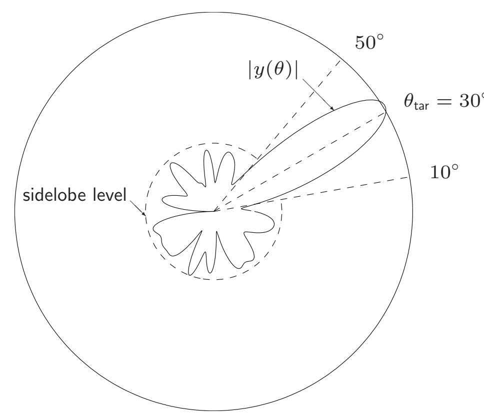
$$\begin{array}{ll} \text{minimize} & \sum_i |y(\theta_i)|^2 \\ \text{subject to} & y(\theta_{\text{tar}}) = 1 \end{array}$$

(sum is over angles outside beam)

**minimize sidelobe level** (discretize angles)

$$\begin{array}{ll} \text{minimize} & \max_i |y(\theta_i)| \\ \text{subject to} & y(\theta_{\text{tar}}) = 1 \end{array}$$

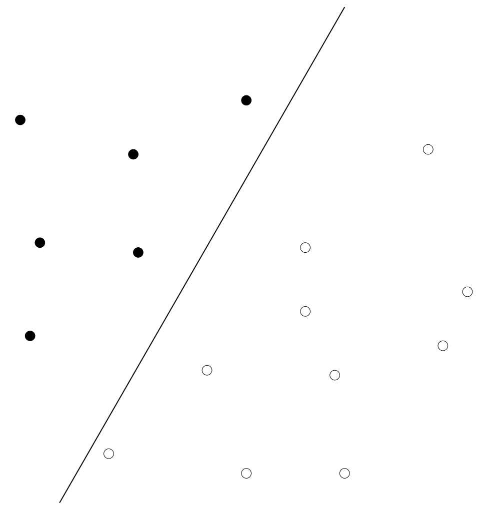
(max over angles outside beam)



# Machine learning

separate two sets of points $\{x_1, \ldots, x_N\}$, $\{y_1, \ldots, y_M\}$ by a hyperplane:

$$a^T x_i + b > 0, \quad i = 1, \ldots, N, \qquad a^T y_i + b < 0, \quad i = 1, \ldots, M$$

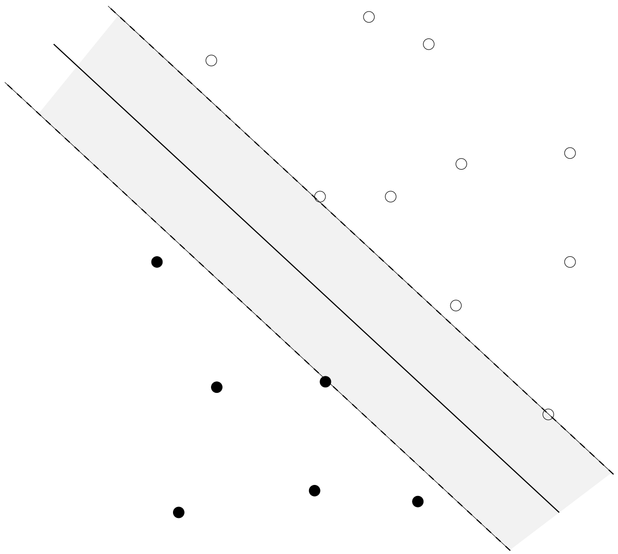


homogeneous in $a$, $b$, hence equivalent to

$$a^T x_i + b \geq 1, \quad i = 1, \ldots, N, \qquad a^T y_i + b \leq -1, \quad i = 1, \ldots, M$$

a set of linear inequalities in $a$, $b$

# Machine learning



(Euclidean) distance between hyperplanes

$$\begin{aligned}\mathcal{H}_1 &= \{z \mid a^T z + b = 1\} \\ \mathcal{H}_2 &= \{z \mid a^T z + b = -1\}\end{aligned}$$

is $\mathbf{dist}(\mathcal{H}_1, \mathcal{H}_2) = 2/\|a\|_2$

to separate two sets of points by maximum margin,

$$\begin{array}{ll} \text{minimize} & (1/2)\|a\|_2 \\ \text{subject to} & a^T x_i + b \geq 1, \quad i = 1, \ldots, N \\ & a^T y_i + b \leq -1, \quad i = 1, \ldots, M \end{array}$$

# Examples

**portfolio optimization**

- variables: amounts invested in different assets
- constraints: budget, max./min. investment per asset, minimum return
- objective: overall risk or return variance

**device sizing in electronic circuits**

- variables: device widths and lengths
- constraints: manufacturing limits, timing requirements, maximum area
- objective: power consumption

**data fitting**

- variables: model parameters
- constraints: prior information, parameter limits
- objective: measure of misfit or prediction error

# Solving optimization problems

**general optimization problem**

- very difficult to solve
- methods involve some compromise, $e.g.$, very long computation time, or not always finding the solution

**exceptions:** certain problem classes can be solved efficiently and reliably

- least-squares problems
- linear programming problems
- convex optimization problems

# Least-squares

$$\text{minimize} \quad \|Ax - b\|_2^2$$

**solving least-squares problems**

- analytical solution: $x^\star = (A^T A)^{-1} A^T b$
- reliable and efficient algorithms and software
- computation time proportional to $n^2 k$ ($A \in \mathbf{R}^{k \times n}$); less if structured
- a mature technology

**using least-squares**

- least-squares problems are easy to recognize
- a few standard techniques increase flexibility ($e.g.$, including weights, adding regularization terms)

# Linear programming

$$\begin{array}{ll} \text{minimize} & c^T x \\ \text{subject to} & a_i^T x \leq b_i, \quad i = 1, \ldots, m \end{array}$$

**solving linear programs**

- no analytical formula for solution
- reliable and efficient algorithms and software
- computation time proportional to $n^2 m$ if $m \geq n$; less with structure
- a mature technology

**using linear programming**

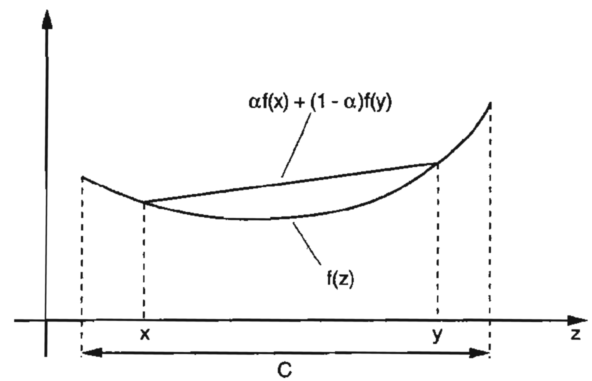
- not as easy to recognize as least-squares problems
- a few standard tricks used to convert problems into linear programs ($e.g.$, problems involving $\ell_1$- or $\ell_\infty$-norms, piecewise-linear functions)

# Convex optimization problem

$$\begin{array}{ll} \text{minimize} & f_0(x) \\ \text{subject to} & f_i(x) \le b_i, \quad i = 1, \ldots, m \end{array}$$

- objective and constraint functions are convex:

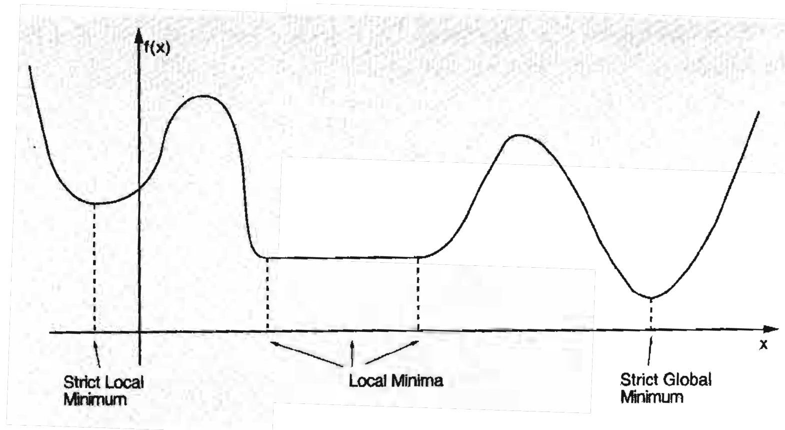
$$f_i(\alpha x + \beta y) \le \alpha f_i(x) + \beta f_i(y)$$




  if $\alpha + \beta = 1$, $\alpha \ge 0$, $\beta \ge 0$

- includes least-squares problems and linear programs as special cases

# The case of a convex cost function




**Local minima**: $x^\star$ is an unconstrained local minimum of $f_0 : \mathbf{R}^n \mapsto \mathbf{R}$ if is no worse than its neighbors.

$$f_0(x^\star) \leq f_0(x), \quad \forall x \in \mathbf{R}^n \text{ with } \|x - x^\star\| < \epsilon$$

for $\epsilon > 0$.

**Global minima**: $x^\star$ is an unconstrained local minimum of $f_0 : \mathbf{R}^n \mapsto \mathbf{R}$ if it is no worse than all other vectors.

$$f_0(x^\star) \leq f_0(x), \quad \forall x \in \mathbf{R}^n.$$

When the function is convex every local minimum is also global.

**solving convex optimization problems**

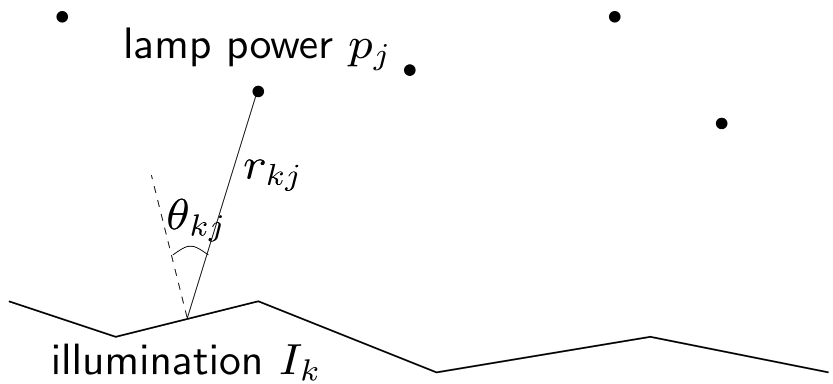
- no analytical solution
- reliable and efficient algorithms
- computation time (roughly) proportional to $\max\{n^3, n^2m, F\}$, where $F$ is cost of evaluating $f_i$'s and their first and second derivatives
- almost a technology

**using convex optimization**

- often difficult to recognize
- many tricks for transforming problems into convex form
- surprisingly many problems can be solved via convex optimization

# Example

$m$ lamps illuminating $n$ (small, flat) patches




intensity $I_k$ at patch $k$ depends linearly on lamp powers $p_j$:

$$I_k = \sum_{j=1}^{m} a_{kj} p_j, \qquad a_{kj} = r_{kj}^{-2} \max\{\cos\theta_{kj}, 0\}$$

**problem**: achieve desired illumination $I_{\text{des}}$ with bounded lamp powers

$$\begin{array}{ll} \text{minimize} & \max_{k=1,\ldots,n} |\log I_k - \log I_{\text{des}}| \\ \text{subject to} & 0 \le p_j \le p_{\max}, \quad j = 1, \ldots, m \end{array}$$

**how to solve?**

1. use uniform power: $p_j = p$, vary $p$
2. use least-squares:

$$\text{minimize} \quad \sum_{k=1}^{n} (I_k - I_{\text{des}})^2$$

   round $p_j$ if $p_j > p_{\text{max}}$ or $p_j < 0$

3. use weighted least-squares:

$$\text{minimize} \quad \sum_{k=1}^{n} (I_k - I_{\text{des}})^2 + \sum_{j=1}^{m} w_j (p_j - p_{\text{max}}/2)^2$$

   iteratively adjust weights $w_j$ until $0 \leq p_j \leq p_{\text{max}}$

4. use linear programming:

$$\begin{array}{ll} \text{minimize} & \max_{k=1,\ldots,n} |I_k - I_{\text{des}}| \\ \text{subject to} & 0 \leq p_j \leq p_{\text{max}}, \quad j = 1, \ldots, m \end{array}$$

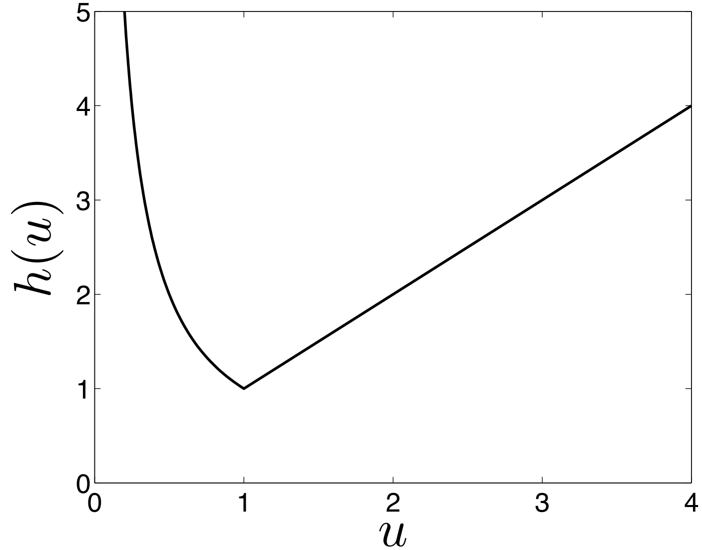
   which can be solved via linear programming

of course these are approximate (suboptimal) 'solutions'

5. use convex optimization: problem is equivalent to

$$\begin{array}{ll} \text{minimize} & f_0(p) = \max_{k=1,\ldots,n} h(I_k/I_{\mathsf{des}}) \\ \text{subject to} & 0 \leq p_j \leq p_{\mathsf{max}}, \quad j = 1, \ldots, m \end{array}$$

with $h(u) = \max\{u, 1/u\}$




$f_0$ is convex because maximum of convex functions is convex

**exact** solution obtained with effort $\approx$ modest factor $\times$ least-squares effort

**additional constraints:** does adding 1 or 2 below complicate the problem?

1. no more than half of total power is in any 10 lamps
2. no more than half of the lamps are on ($p_j > 0$)

- answer: with (1), still easy to solve; with (2), extremely difficult
- moral: (untrained) intuition doesn't always work; without the proper background very easy problems can appear quite similar to very difficult problems

# Course goals and topics

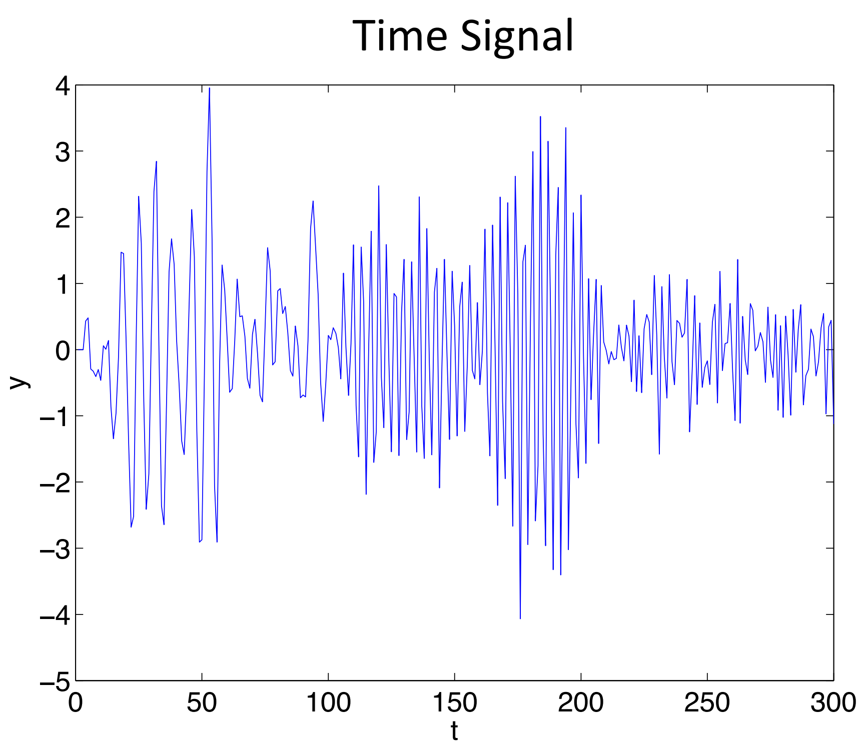
**Goals**

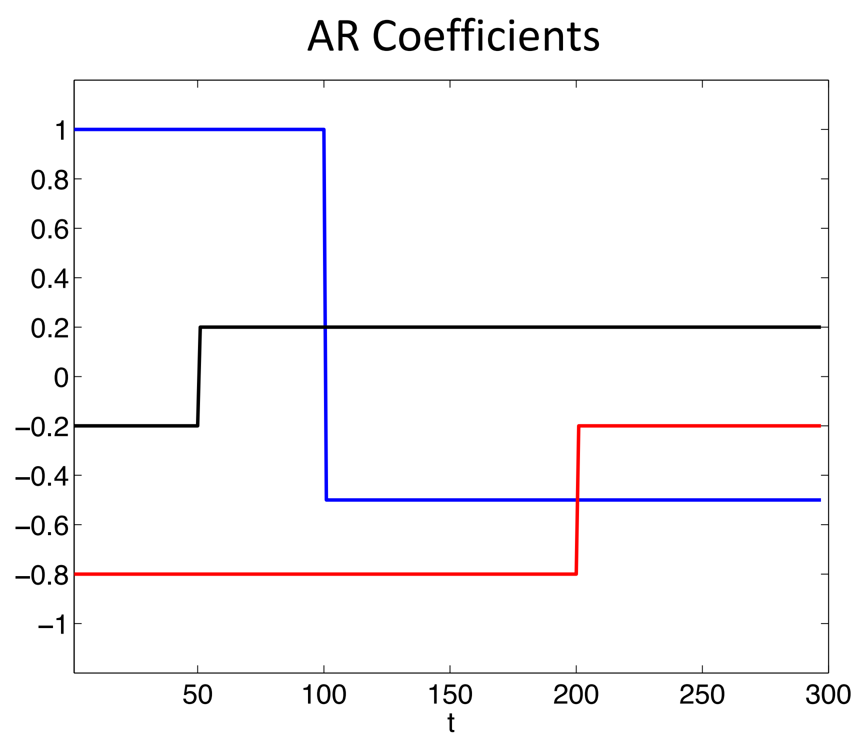
1. recognize and formulate problems (such as the illumination problem, classification, etc.) as convex optimization problems
2. Use optimization tools (CVX, YALMIP, etc.) as a part the lab assignment.
3. characterize optimal solution (optimal power distribution), give limits of performance, etc.

**Topics**

1. Background and optimization basics;
2. Convex sets and functions;
3. Canonical convex optimization problems (LP, QP, SDP);
4. Second-order methods (unconstrained and constrained optimization);
5. First-order methods (gradient, subgradient);

# Project 1: Change Detection in Time Series Model

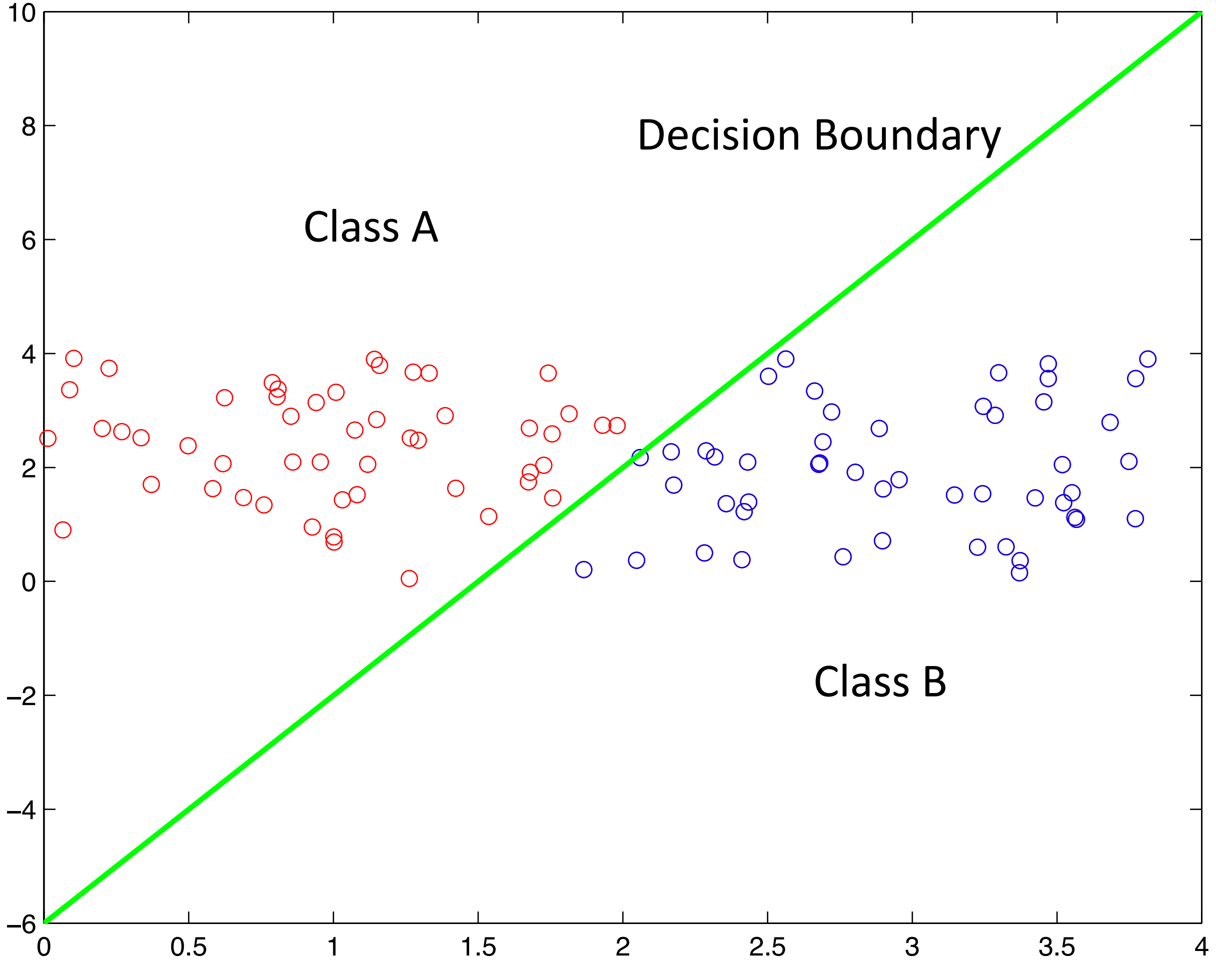


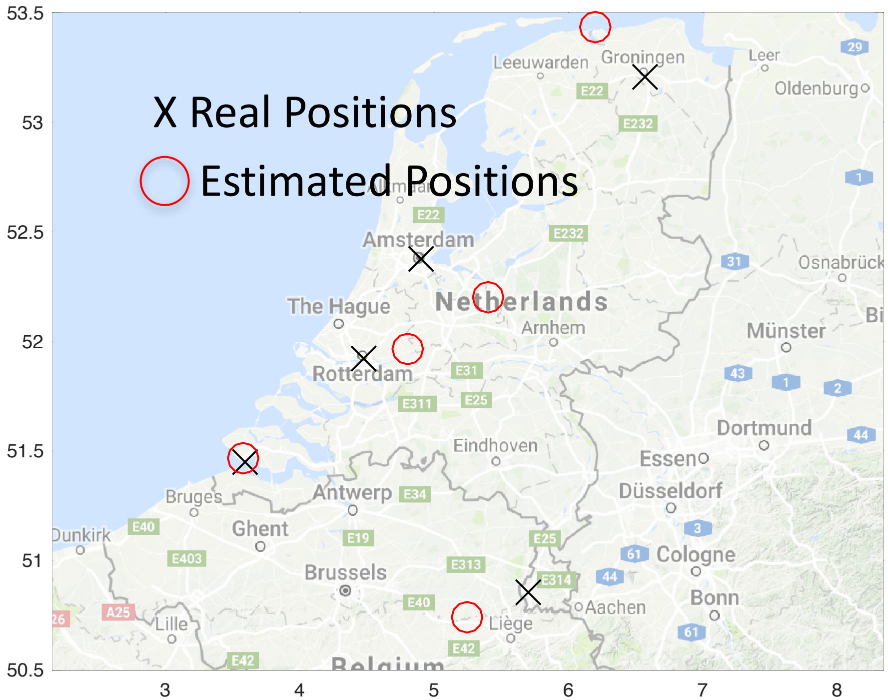

$$y(t+3) = a(t)y(t+2) + b(t)y(t+1) + c(t)y(t) + v(t); \quad v(t) \sim \mathcal{N}(0, 0.5^2).$$

assumption: $a(t)$, $b(t)$, and $c(t)$ are piecewise constant, change infrequently

# Project 2: Linear Support Vector Machines



# Project 3: Multidimensional Scaling for Localization.

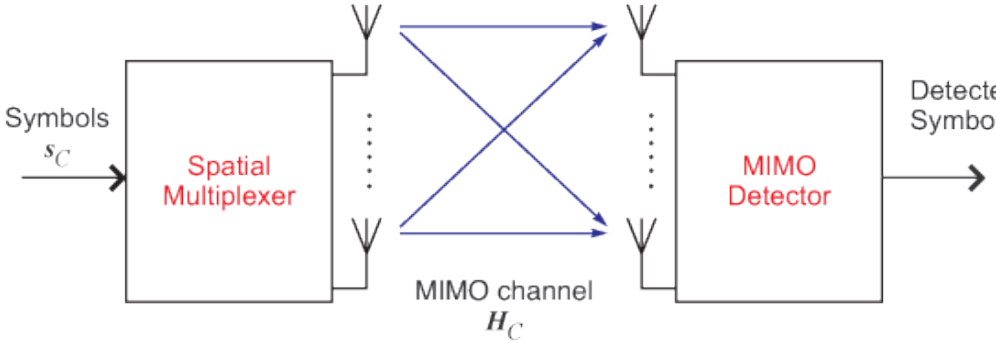



| | A | R | M | V | G |
|---|---|---|---|---|---|
| A | 0 | 71 | 146 | 177 | 127 |
| R | 71 | 0 | 136 | 104 | 159 |
| M | 146 | 136 | 0 | 208 | 258 |
| V | 177 | 104 | 208 | 0 | 279 |
| G | 127 | 159 | 258 | 279 | 0 |

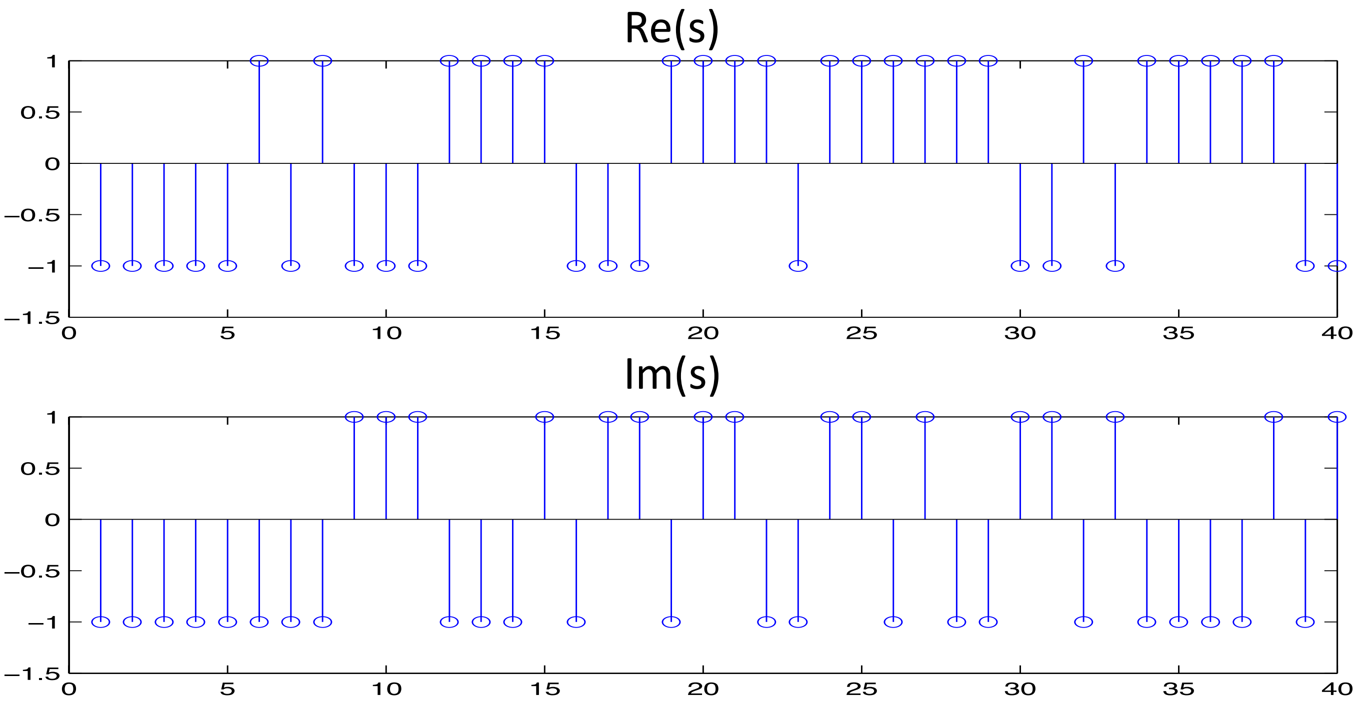
$$t_{ij}^2 \propto \|\mathbf{x}_i - \mathbf{x}_j\|^2$$

$$\mathbf{T} = \mathbf{1}\mathrm{diag}(\mathbf{X}^T\mathbf{X}) - 2\mathbf{X}^T\mathbf{X} + \mathrm{diag}(\mathbf{X}^T\mathbf{X})\mathbf{1}^T$$

# Project 4: MIMO Detection




$$\mathbf{y}_c = \mathbf{H}_c \mathbf{s}_c + \mathbf{v}_c.$$




entries of $\mathbf{s}_c$ belong to the finite-alphabet set $\{\pm 1 \pm j\}$

# Project 5: Compressed Sensing