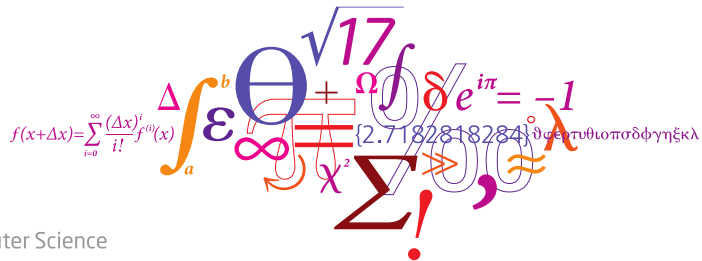
# 02471 Machine Learning for Signal Processing

## Sparsity-aware learning

Tommy Sonne Alstrøm

Cognitive Systems section




DTU Compute
Department of Applied Mathematics and Computer Science

## Outline

- Course admin
- Last week review
- Sparsity-aware learning
- Norms
- LASSO
- Next Week

Material: ML 8.2, 8.10.1–8.10.2, 9.1–9.5, 9.9.

# Feedback

- Feedback from last week:
  - List of symbols (todo, suggestion from week 4).
  - Latex files provided for problem set 2.
- Feedback from you is a critical component for improving both the course and my teaching.
- Type of feedback
  - Mention one thing that worked?
  - Mention one thing should be improved (both in current lecture and last weeks exercise)?
  - Mention one thing you would change if you gave the lecture.

# The story so far and what the future holds

- Problem set 1 submission corrections pending.
- Problem set 2 can be solved, except problem 2.6, which uses material for week 7.

What you have learned so far:

- Parameter estimation [L2 regularization, biased estimation, mean squared error minimization]. Todo: L1 regularization (this week), Bayesian parameter estimation (in 4 weeks)
- Filtering signals [Stochastic processes, correlation functions, Wiener filter, linear prediction, adaptive filtering using stochastic gradient decent (LMS, APA/NLMS), adaptive filtering using regularization (RLS)]

The topic the next three weeks will be sparse signal representations and dictionary learning

- Signal representations [Time frequency analysis, sparsity aware learning, factor models]
- Sparsity aware sensing (lasso, sparse priors), compressed sensing, dictionary learning [Independent component analysis, Non-negative matrix factorization, $k$-SVD]

# Learning objectives

**Learning objectives**

A student who has met the objectives of the course will be able to:

- Explain, apply and analyze properties of discrete time signal processing systems
- Apply the short time Fourier transform to compute the spectrogram of a signal and analyze the signal content
- Explain compressed sensing and determine the relevant parameters in specific applications
- Deduce and determine how to apply factor models such as non-negative matrix factorization (NMF), independent component analysis (ICA) and sparse coding
- Deduce and apply correlation functions for various signal classes, in particular for stochastic signals
- Analyze filtering problems and demonstrate the application of least squares filter components such as the Wiener filter
- Describe, apply and derive non-linear signal processing methods based such as kernel methods and reproducing kernel Hilbert space for applications such as denoising
- Derive maximum likelihood estimates and apply the EM algorithm to learn model parameters
- Describe, apply and derive state-space models such as Kalman filters and Hidden Markov models
- Solve and interpret the result of signal processing systems by use of a programming language
- Design simple signal processing systems based on an analysis of involved signal characteristics, the objective of the processing system, and utility of methods presented in the course
- Describe a number of signal processing applications and interpret the results

# Last week review

# The RLS algorithm

RLS minimized the exponentially weighted least-squares cost function

## Exponentially weighted least-squares

$$J(\boldsymbol{\theta}, \beta, \lambda) = \sum_{i=0}^{n} \beta^{n-i}(y_i - \boldsymbol{\theta}^T \boldsymbol{x}_i)^2 + \lambda \beta^{n+1} \|\boldsymbol{\theta}\|^2$$

Choose forget factor $\beta$ and regularization $\lambda$.

We have an iterative update formula: update $\boldsymbol{\theta}_n$ at each new observation $(\boldsymbol{x}_n, y_n, )$

## Update formula

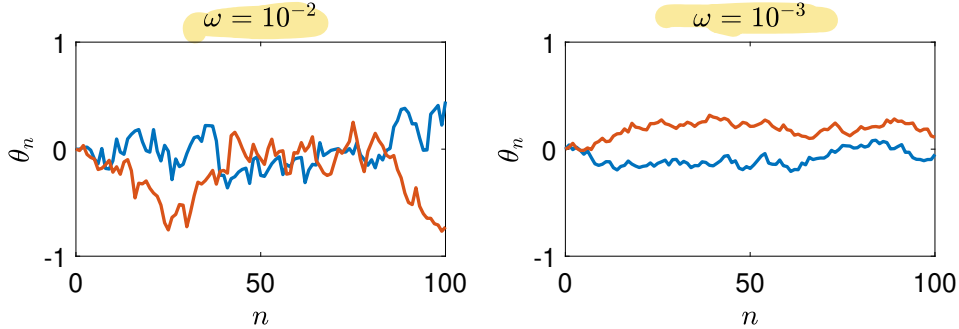
$$\begin{aligned}
e_n &= y_n - \boldsymbol{\theta}_{n-1}^T \boldsymbol{x}_n \\
\boldsymbol{z}_n &= P_{n-1} \boldsymbol{x}_n \\
\boldsymbol{k}_n &= \boldsymbol{z}_n / (\beta + \boldsymbol{x}_n^T \boldsymbol{z}_n) \\
\boldsymbol{\theta}_n &= \boldsymbol{\theta}_{n-1} + \boldsymbol{k}_n e_n \\
P_n &= \beta^{-1} P_{n-1} - \beta^{-1} \boldsymbol{k}_n \boldsymbol{z}_n^T
\end{aligned}$$

# How do we simulate non-stationary environment?

## Non-stationary environment

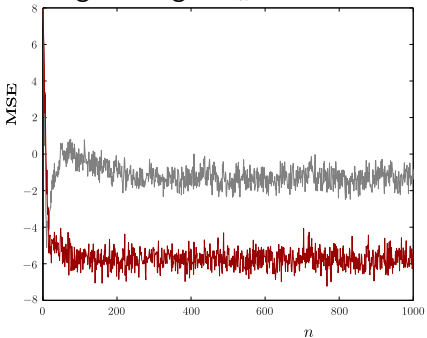
$$y_n = \boldsymbol{\theta}_o^T \boldsymbol{x}_n + \eta_n \quad \text{obs}$$
$$\boldsymbol{\theta}_{o,n} = \alpha \boldsymbol{\theta}_{o,n-1} + \boldsymbol{\omega}_n \quad \text{state}$$

- $\alpha < 1$ is the autoregressive coefficient (memory).
- $\boldsymbol{\omega}_n$ is zero mean Gaussian i.i.d with diagonal covariance (white noise).

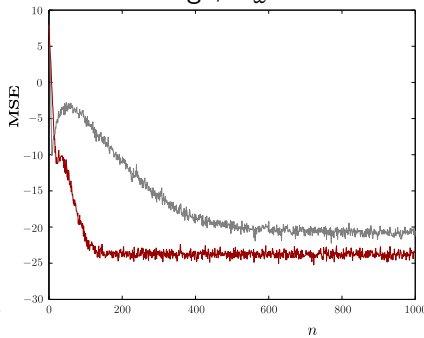


# Convergence rates of RLS, time–tracking

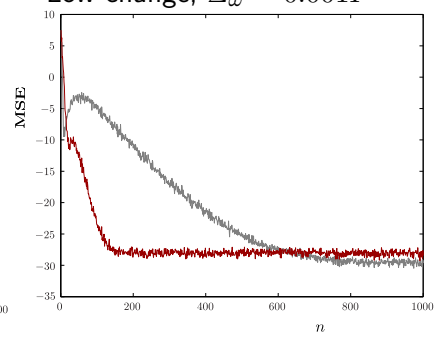
High change, $\Sigma_\omega = 0.1I$




Medium change, $\Sigma_\omega = 0.01I$




Low change, $\Sigma_\omega = 0.001I$




Gray: RLS, Red: NLMS. $y$–axis is in dB.

Convergence results explained by $J_{\text{exc}} = J_{\text{excess}} + J_{\text{lag}}$.

For NLMS: $J_{\text{exc}} = \frac{1}{2}\mu\sigma_\eta^2 \operatorname{trace}\{\Sigma_x\} \frac{1}{\sigma_x^2(l-2)} + \frac{1}{2}\mu^{-1} \operatorname{trace}\{\Sigma_x\} \operatorname{trace}\{\Sigma_\omega\}$

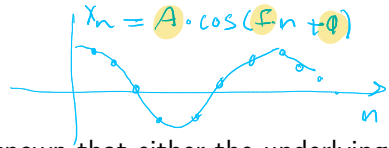
For RLS: $J_{\text{exc}} = \frac{1}{2}(1-\beta)\sigma_\eta^2 l + \frac{1}{2}(1-\beta)^{-1} \operatorname{trace}\{\Sigma_\omega \Sigma_x\}$.

# Summary

- RLS can under certain circumstances converge faster than NLMS
  - There exist more theory on convergence analysis of RLS vs LMS, on when to choose RLS over LMS, it depends on the signal properties
- We need to consider both time-tracking performance and stationary performance.
- We have models to simulate both scenarios.
- Even though the algorithms are "old", they are still very relevant.

# Sparsity-aware learning

## What is sparsity-aware learning




In a number of practical problems, it is known that either the underlying model is sparse or it is sparse in a "transform" domain, e.g., in the Fourier transform domain.

By sparse models is meant that most of the unknown coordinates in the model are zero.

Recall from Chapter 3, the use of a regularizer can shrink the norm of the obtained solution. In this vein, we will see that by adopting appropriate regularizers, one can help the optimization process to identify the coordinates of the zeros.

Leads to compressed sensing, where the goal is to directly acquire as few samples as possible that encode the minimum information, which is needed to obtain a compressed signal representation.

Algorithms covered so far will struggle with either non-white noise, or presence of outliers.

# Starting point

A linear system with multiple solutions (under-determined system):

$$y = \boldsymbol{a} \cdot \boldsymbol{x} \Rightarrow 2 = \begin{bmatrix} 1 & 1 \end{bmatrix} \cdot \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

$\hat{x}_1 = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$ $l_2$

Which solution would you pick?

$\|\hat{x}_1\|_0 = 2$ $\quad \|\hat{x}_1\|_1 = \|\hat{x}_2\|_1$

$\|\hat{x}_2\|_0 = 1$ $\quad \|\hat{x}_1\|_2 < \|\hat{x}_2\|$

$\hat{x}_2 = \begin{bmatrix} 2 \\ 0 \end{bmatrix}$

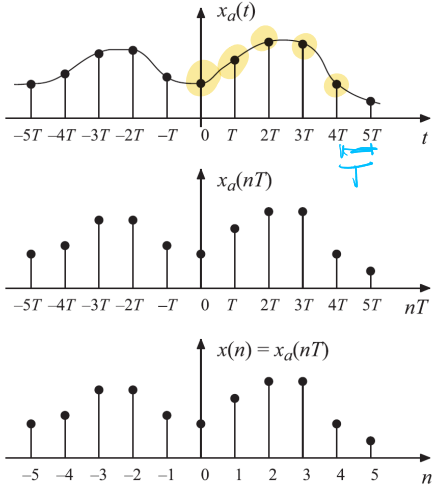
$\hat{x}_3 = \begin{bmatrix} 0 \\ 2 \end{bmatrix}$

# Traditional sampling rate

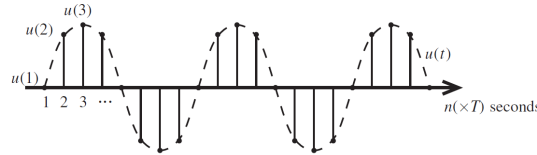
$F_{max} \leq 2 \cdot F_s$

$10\,Hz$ $20\,Hz$

Traditional sampling

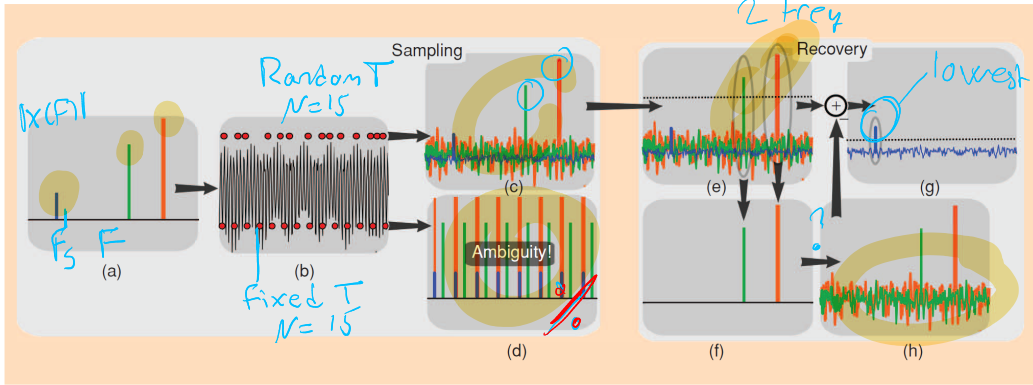



Suppose you were given this signal:




Can you sample more clever?

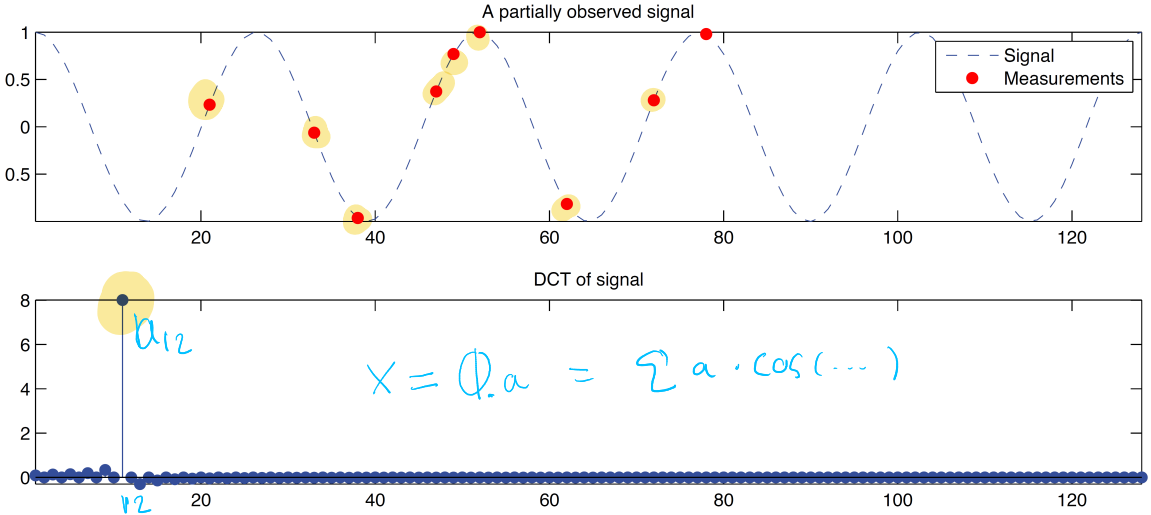
# Traditional under-sampling




**[FIG5]** Heuristic procedure for reconstruction from undersampled data. A sparse signal (a) is 8-fold undersampled in its 1-D $k$-space domain (b). Equispaced undersampling results in signal aliasing (d) preventing recovery. Pseudo-random undersampling results in incoherent interference (c). Some strong signal components stick above the interference level, are detected and recovered by thresholding (e) and (f). The interference of these components is computed (g) and subtracted (h), thus lowering the total interference level and enabling recovery of weaker components.

M. Lustig, D. L. Donoho, J. M. Santos and J. M. Pauly, "Compressed Sensing MRI," in IEEE Signal Processing Magazine, vol. 25, no. 2, pp. 72-82, March 2008.

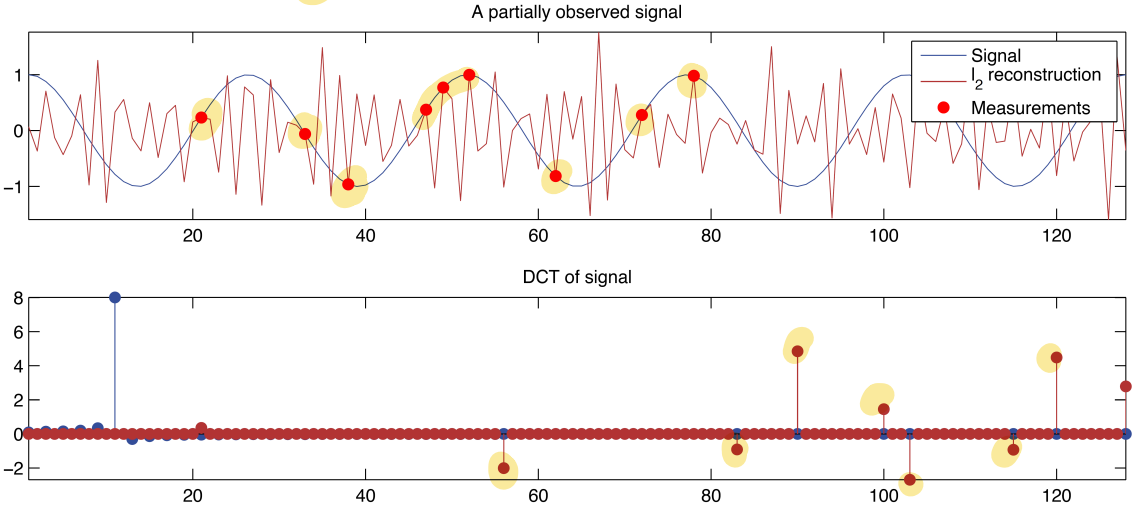
# Random sampling




Example courtesy Prof. Paris Smaragdis

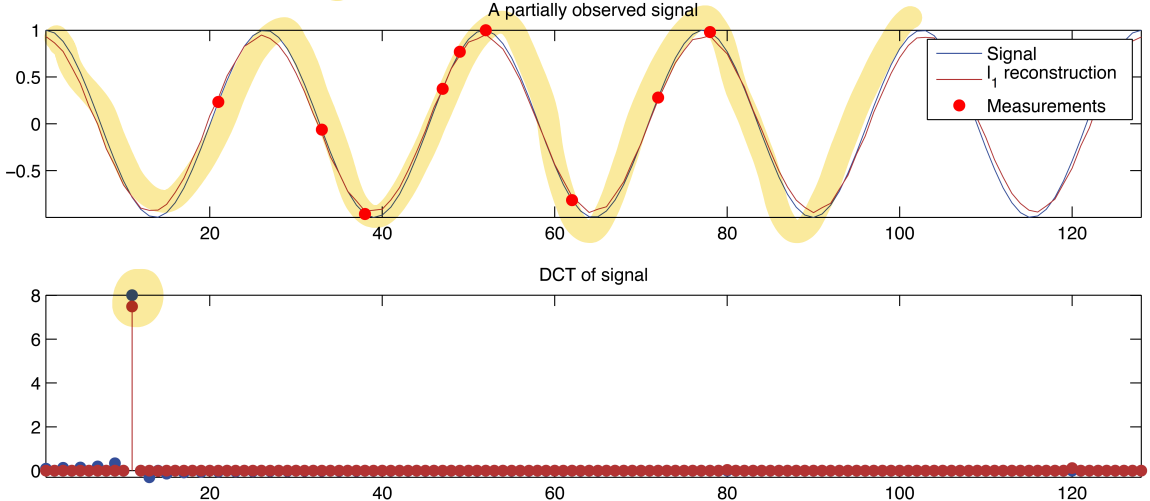
# Reconstruction using $\ell_2$

Ridge reg



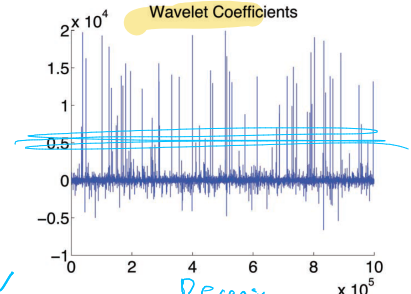

Example courtesy Prof. Paris Smaragdis

# Reconstruction using $\ell_1$




Example courtesy Prof. Paris Smaragdis

# Wavelet compression




Wavelet matrix

$x = \Phi a$

RAW

Recons.

Jpeg !

# Summary

Sparsity-aware learning have many cool applications, in statistics, signal processing, and machine learning.

- Signal recovery in under-determined system.
- Leads to dictionary learning. Week 8
- ... and to compressed sensing, where we sample and compress at the same time.

# Norms

# Cost functions

## Ridge regression

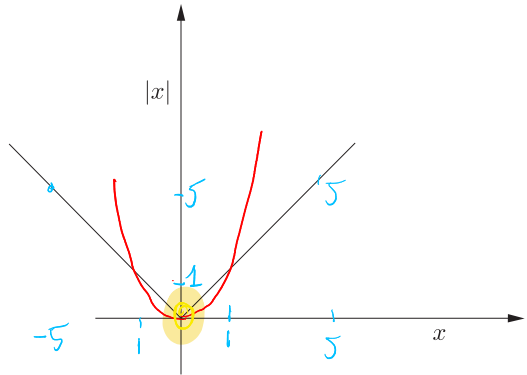
$$J(\boldsymbol{\theta}, \lambda) = \sum_{i=1}^{N} (y_i - \boldsymbol{\theta}^T \boldsymbol{x}_i)^2 + \lambda \|\boldsymbol{\theta}\|_2^2 \qquad eq.(3.39)$$

Can we think of another way to construct our regularization to promote sparsity

## How about the absolute value?

$\frac{\partial J(\theta)}{\partial \theta} = 0$

$f(x) = |x|$ l1 in 1d



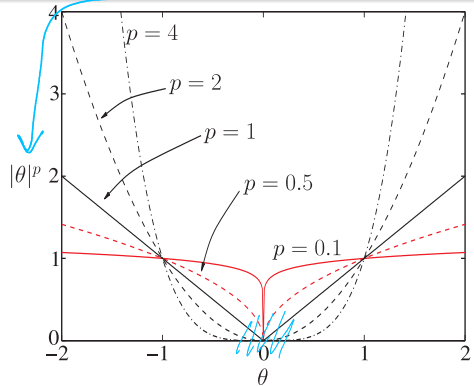

Is there any problems in using this function?

# The $\ell_p$ norm

## The $\ell_p$ norm

$$\|\boldsymbol{\theta}\|_p := \left(\sum_{i=1}^{l} |\theta_i|^p\right)^{1/p}$$

$l$ is the size of vector $\boldsymbol{\theta}$.



# Norms

## The $\ell_p$ norm

$$\|\boldsymbol{\theta}\|_p := \left( \sum_{i=1}^{l} |\theta_i|^p \right)^{1/p}, \quad p \geq 1$$

ex 6.1

$l$ is the size of vector $\boldsymbol{\theta}$.

## Definition of norm

Let $V$ be a vector space. A norm on $V$ is a function $\|\cdot\| : V \to \mathbb{R}$ that satisfies the following three conditions, $\forall \boldsymbol{\theta} \in V$:

1. $\|\boldsymbol{\theta}\|_p \geq 0, \quad \|\boldsymbol{\theta}\|_p = 0 \Leftrightarrow \boldsymbol{\theta} = \mathbf{0}$ (positive)
2. $\|\alpha\boldsymbol{\theta}\|_p = |\alpha| \|\boldsymbol{\theta}\|_p, \forall \alpha \in \mathbb{R}$
3. $\|\boldsymbol{\theta}_1 + \boldsymbol{\theta}_2\|_p \leq \|\boldsymbol{\theta}_1\|_p + \|\boldsymbol{\theta}_2\|_p$ (triangle inequality)

LHS RHS

$\theta_1 = 1 \quad \theta_2 = 1 \quad p = 1/2$

$\left(1^{1/2} + 1^{1/2}\right)^{1/(1/2)} = 2^2 = 4 \quad \not\leq \quad 2$

$\left(1^{1/2}\right)^{1/(1/2)} = 1$

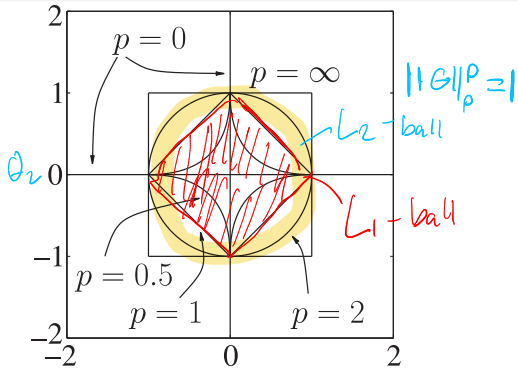
Why is $\ell_p$ not a norm for $p < 1$ ? can you create an example that breaks the triangle inequality?

# How does the $\ell_p$ norm behave

## Special cases

$$\|\boldsymbol{\theta}\|_\infty := \arg\max_i |\theta_i| \quad \text{(max element)}$$

$$\|\boldsymbol{\theta}\|_0 := |\{i \,|\, \theta_i \neq 0, i = 1, \cdots, l\}| \quad \text{(number of nonzeros)}$$

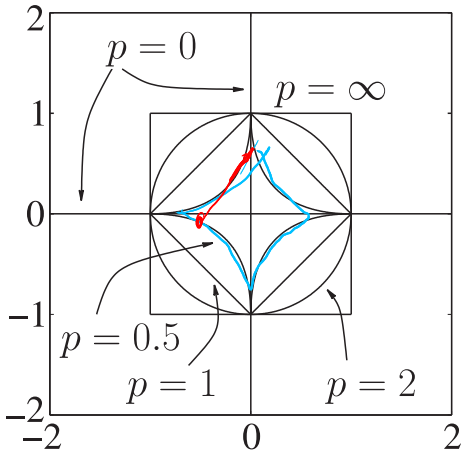


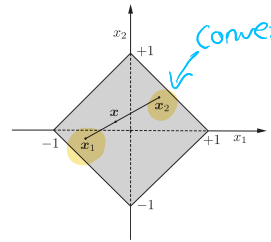

Notation remarks:
$|x|$ is the numerical value when $x \in \mathbb{R}$.
$|\mathcal{X}|$ is the cardinality (size) of the set $\mathcal{X}$.

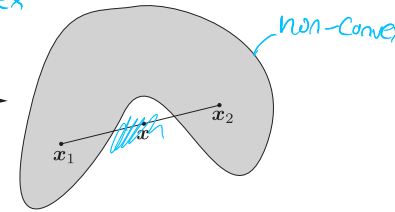
How can these different norms promote sparsity?

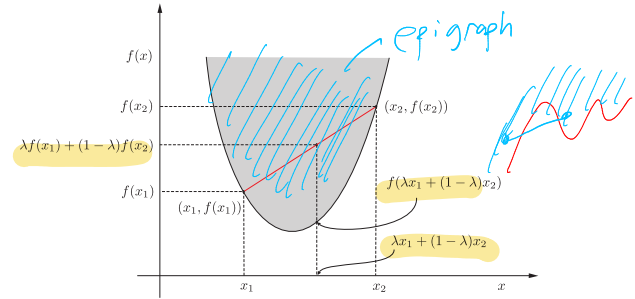
## The $\ell_p$ norm is convex ($p \geq 1$)

IML 8.2 3 pages




Can be proven by showing triangle inequality implies convexity.

# Summary

- The $\ell_p$ norm can in different configurations regularize our learning problem.
- $\ell_p$ is only a true norm for $p \geq 1$.
- $\ell_0$ is the most sparse "norm", and counts the number of nonzero elements.
- The $\ell_1$ norm is the most sparse true norm, and is convex, hence we can optimize it.
- The $\ell_1$ norm is not differentiable though. need fixing

For a rigorous treatment of norms: 01325 – Function spaces and mathematical analysis.

For a rigorous treatment of optimization: 02612 – Constrained Optimization.

# LASSO

# RLS-weighted Lasso

## Exponentially weighted least-squares

$$J(\boldsymbol{\theta}, \beta, \lambda) = \sum_{i=0}^{n} \beta^{n-i}(y_i - \boldsymbol{\theta}^T \boldsymbol{x}_i)^2 + \lambda \beta^{n+1} \|\boldsymbol{\theta}\|_2^2$$

RLS

## Recursive LASSO cost function

$$J(\boldsymbol{\theta}, \beta, \lambda) = \sum_{i=0}^{n} \beta^{n-i}(y_i - \boldsymbol{\theta}^T \boldsymbol{x}_i)^2 + \lambda \beta^{n+1} \|\boldsymbol{\theta}\|_1$$

Ref: D. Angelosante and G. B. Giannakis, "RLS-weighted Lasso for adaptive estimation of sparse signals," 2009 IEEE International Conference on Acoustics, Speech and Signal Processing, Taipei, 2009, pp. 3245-3248.

# RLS-weighted Lasso convergence

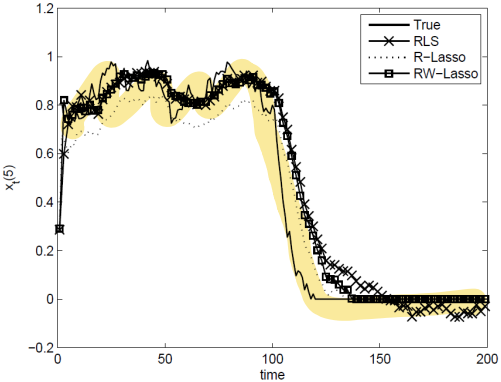



**Fig. 3**. Time evolution of the $x_5$ signal entry.

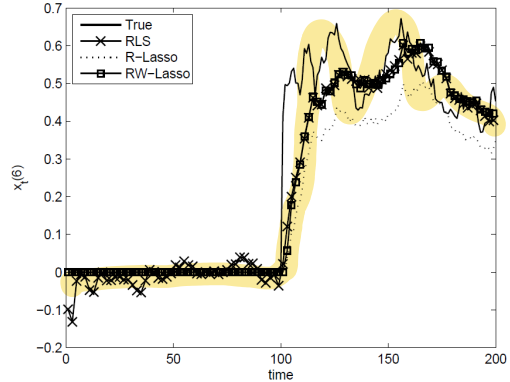



**Fig. 4**. Time evolution of the $x_6$ signal entry.

Ref: D. Angelosante and G. B. Giannakis, "RLS-weighted Lasso for adaptive estimation of sparse signals," 2009 IEEE International Conference on Acoustics, Speech and Signal Processing, Taipei, 2009, pp. 3245-3248.

# The Least Absolute Shrinkage and Selection Operator (LASSO)

We have the familiar regression task (on matrix form)

$$\boldsymbol{y} = X\boldsymbol{\theta} + \boldsymbol{\eta}, \quad \boldsymbol{y} \in \mathbb{R}^N,\ X \in \mathbb{R}^{N \times l},\ \boldsymbol{\theta} \in \mathbb{R}^l,\ \boldsymbol{\eta} \in \mathbb{R}^N,$$

**LASSO cost function**

$$J(\boldsymbol{\theta}, \lambda) = \sum_{i=1}^{N} (y_i - \boldsymbol{\theta}^T \boldsymbol{x}_i)^2 + \lambda \|\boldsymbol{\theta}\|_1$$

Note: $\sum_{i=1}^{N} a_i^2 = \boldsymbol{a} \cdot \boldsymbol{a} = \boldsymbol{a}^T \boldsymbol{a} = \|\boldsymbol{a}\|_2^2$

The LASSO cost function can equivalently be written as

$$J(\boldsymbol{\theta}, \lambda) = (\boldsymbol{y} - X\boldsymbol{\theta})^T (\boldsymbol{y} - X\boldsymbol{\theta}) + \lambda \|\boldsymbol{\theta}\|_1$$

# LASSO minimization problem

**LASSO minimization, $p = 1$**

$$\hat{\boldsymbol{\theta}}_1 = \arg\min_{\boldsymbol{\theta}} J(\boldsymbol{\theta}, \lambda) = \sum_{i=1}^{N} (y_i - \boldsymbol{\theta}^T \boldsymbol{x}_i)^2 + \lambda \|\boldsymbol{\theta}\|_p^p$$

Nomenclature

- $\hat{\boldsymbol{\theta}}_{LS}$ denotes the least squares solution $\lambda = 0$
- $\hat{\boldsymbol{\theta}}_R$ denotes the least squares solution with $\ell_2$ regularization (Ridge regression) $p = 2$
- $\hat{\boldsymbol{\theta}}_1$ denotes the least squares solution with $\ell_1$ regularization (LASSO) $p = 1$

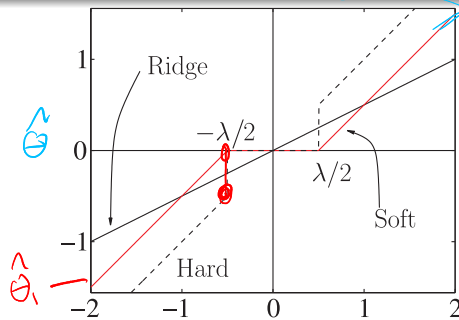
# Input–output of the weights when $X^T X = I$

## The weight estimates when $X^T X = I$

$$\hat{\boldsymbol{\theta}}_{LS} = X^T \boldsymbol{y}$$

$$\hat{\boldsymbol{\theta}}_{R} = \frac{1}{1+\lambda}\hat{\boldsymbol{\theta}}_{LS}$$

$$\hat{\theta}_{1,i} = \text{sgn}\left(\hat{\theta}_{LS,i}\right)\left(|\hat{\theta}_{LS,i}| - \frac{\lambda}{2}\right) \qquad i = 1, 2, \cdots, l$$



## An example

Assume $\lambda = 1$ and $X^T X = I$, then given a solution for $\hat{\boldsymbol{\theta}}_{LS}$, compute the weights for the Ridge, Soft, and Hard thresholds. We have

$$\hat{\boldsymbol{\theta}}_{LS} = [0.4, 0.6]^T$$

$$\hat{\boldsymbol{\theta}}_R = \frac{1}{1+\lambda}\hat{\boldsymbol{\theta}}_{LS} = \frac{1}{2}[0.4 \;\; 0.6]^T = [0.2 \;\; 0.3]^T$$

$$\hat{\theta}_{1,i} = \text{sgn}\left(\hat{\theta}_{LS,i}\right)\left(|\hat{\theta}_{LS,i}| - \frac{\lambda}{2}\right)_+ \qquad i = 1, 2, \cdots, l$$

We get ...

Compute on your own

$$\theta_1 = \begin{bmatrix} 0.4 - 0.5 \\ 0.6 - 0.5 \end{bmatrix}_+ = \begin{bmatrix} 0 \\ 0.1 \end{bmatrix}$$

# LASSO minimization problem

## LASSO minimization

$$\hat{\boldsymbol{\theta}}_1 = \arg\min_{\boldsymbol{\theta}} J(\boldsymbol{\theta}, \lambda) = (\boldsymbol{y} - X\boldsymbol{\theta})^T(\boldsymbol{y} - X\boldsymbol{\theta}) + \lambda\|\boldsymbol{\theta}\|_1$$

The minimization problem can equivalently be written as

$$\hat{\boldsymbol{\theta}}_1 = \arg\min_{\boldsymbol{\theta}} (\boldsymbol{y} - X\boldsymbol{\theta})^T(\boldsymbol{y} - X\boldsymbol{\theta})$$
$$\text{Subject to} : \|\boldsymbol{\theta}\|_1 \leq \rho$$

For a specific choice of $\lambda$ and $\rho$.
To see this, use Lagrangian multipliers (appendix C):

$$\begin{aligned} \text{minimize} : \quad & J(\boldsymbol{\theta}) \\ \text{subject to} : \quad & f(\boldsymbol{\theta}) \geq 0 \\ \text{Lagrangian} : \quad & L(\boldsymbol{\theta}, \lambda) = J(\boldsymbol{\theta}) - \lambda f(\boldsymbol{\theta}) \end{aligned}$$

## The LASSO solution

Take the derivative of the cost function, and put to zero, solve for $\boldsymbol{\theta}$: $\frac{d}{d\boldsymbol{\theta}} J(\boldsymbol{\theta}, \lambda) = \mathbf{0}$

Problem: the $\ell_1$ norm is not differentiable.

Solution: use the subgradient method (sec 8.10).

**Subgradient Algorithm**

$$\boldsymbol{\theta}^{(i)} = \boldsymbol{\theta}^{(i-1)} - \mu_i J'\left(\boldsymbol{\theta}^{(i-1)}\right)$$

$J'(\cdot)$ denotes any subgradient of $J(\cdot)$.

Converges if the stepsize $\mu_i$ diminishes over time. Additionally

**Condition for a minimizer**

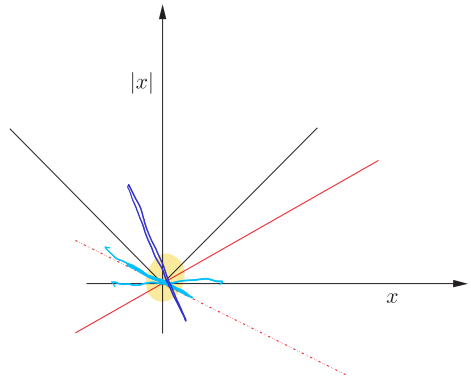
Given a convex function $f$, the zero vector must belong to the subgradient at $\boldsymbol{x}_*$, then $\boldsymbol{x}_*$ is a minimizer, i.e

$$\mathbf{0} \in \partial f(\boldsymbol{x}_*)$$

$\partial f(\boldsymbol{x}_*)$ is the subdifferential of $f(\boldsymbol{x})$ at $\boldsymbol{x}_*$ (the set of all subgradients).

## Subgradient of $\ell_1$

$$\partial f(x) = \begin{cases} \operatorname{sgn}(x), & \text{if } x \neq 0 \\ [-1, 1], & \text{if } x = 0 \end{cases}$$




Can formally be derived using definition of subdifferentials (example 8.4).

## Simplified analysis of weights

Assume $X^T X = I$, then

$$\hat{\boldsymbol{\theta}}_{LS} = (X^T X)^{-1} X^T \boldsymbol{y} = X^T \boldsymbol{y}$$

$$\hat{\boldsymbol{\theta}}_{R} = (X^T X + \lambda I)^{-1} X^T \boldsymbol{y} = \frac{1}{1+\lambda} \hat{\boldsymbol{\theta}}_{LS}$$

For $\ell_1$, we get

$$J(\boldsymbol{\theta}, \lambda) = (\boldsymbol{y} - X\boldsymbol{\theta})^T (\boldsymbol{y} - X\boldsymbol{\theta}) + \lambda \|\boldsymbol{\theta}\|_1$$

$$\partial_{\boldsymbol{\theta}} J(\boldsymbol{\theta}, \lambda) = -2X^T \boldsymbol{y} + 2X^T X \boldsymbol{\theta} + \lambda \partial \|\boldsymbol{\theta}\|_1$$

Which lead to

$$0 \in -\theta_{LS,i} + \hat{\theta}_{i,1} + \frac{\lambda}{2} \partial |\hat{\theta}_{i,1}|$$

Which again leads to (which you will derive in the exercise)

$$\hat{\theta}_{1,i} = \operatorname{sgn}\left(\hat{\theta}_{LS,i}\right) \left( |\hat{\theta}_{LS,i}| - \frac{\lambda}{2} \right)_+ \qquad i = 1, 2, \cdots, l$$

# When is $\ell_1$ unique? Example of recovery I

- Consider an unknown system.
- The system only has two parameters, and you would like to recover those parameters.
- You know the system is on the form $f(x_1, x_2) = x_1\theta_1 + x_2\theta_2$.
- You can only measure once, that is, you can only select one pair of $(x_1, x_2)$.
- To simulate this environment, we create the $\theta$'s for the unknown system, and select those as $(\theta_1, \theta_2) = (0, 1)$ (arbitrary choice).
- We will now see what happens if we select 3 possible measurements, $\boldsymbol{x}_a = (1/2, 1)$, $\boldsymbol{x}_b = (1, 1)$, $\boldsymbol{x}_c = (2, 1)$.
- These three measurements will all give a response of $f(x_1, x_2) = 1$ for the unknown system.

# When is $\ell_1$ unique? Example of recovery II

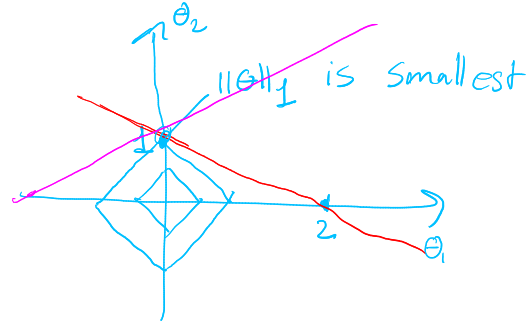
True $\boldsymbol{\theta} = (0, 1)$. find $\theta$

Test measurement: $\boldsymbol{x}_a = (0.5, 1)$ $\Rightarrow f(x) = 1$

System: $f(\boldsymbol{x}) = x_1\theta_1 + x_2\theta_2$ "box"

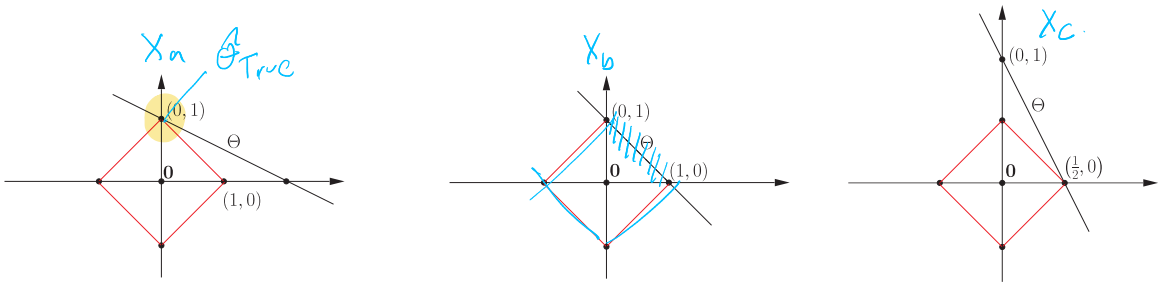
$1 = 0.5 \cdot 0 + 1 \cdot 1$

$1 = 0.5\,\theta_1 + 1 \cdot \theta_2$



## When is $\ell_1$ unique? Example of recovery II

True $\boldsymbol{\theta} = (0, 1)$.
Test measurements: $\boldsymbol{x}_a = (1/2, 1)$, $\boldsymbol{x}_b = (1, 1)$, $\boldsymbol{x}_c = (2, 1)$ (all result in $f(\boldsymbol{x}) = 1$).
Let us draw possible solutions to the linear system, $f(\boldsymbol{x}) = x_1\theta_1 + x_2\theta_2 = 1$, and select the solution with the lowest $\ell_1$ norm.




Compressed sensing explores the construction of $X$, and it turns out that a good $X$ is a random $X$ (sec 9.6–9.8).

## Lecture summary

- In the pursuit of sparse solution, we arrive at using the $\ell_1$ norm as regularizer (LASSO) as the computationally most efficient norm.
  - The norm is convex.
  - No closed form solution, but solved with subgradients.
  - Has a soft thresholding operation, in the special case $X^T X = I$, sets weights to zero once they are numerically smaller than $\lambda/2$.
  - Almost always have a unique solution.
- The $\ell_0$ "norm" leads to the sparsest solution, but is not convex.
- Under special circumstances, the $\ell_1$ regularization will find the sparsest solution.
- We only have ONE solution (so far) for the LASSO when $X^T X = I$. Algorithms will be looked at next week.

# Next week

Material: ML 10.1, 10.2–10.2.1 (until p.476), 10.2.2 (until p.482), 10.5–10.6.

- Estimating the LASSO solution:
  - LARS and Matching Pursuit algorithms
- Towards dictionary learning
  - Sparse analysis models
  - Time–frequency analysis

2022. pl.3