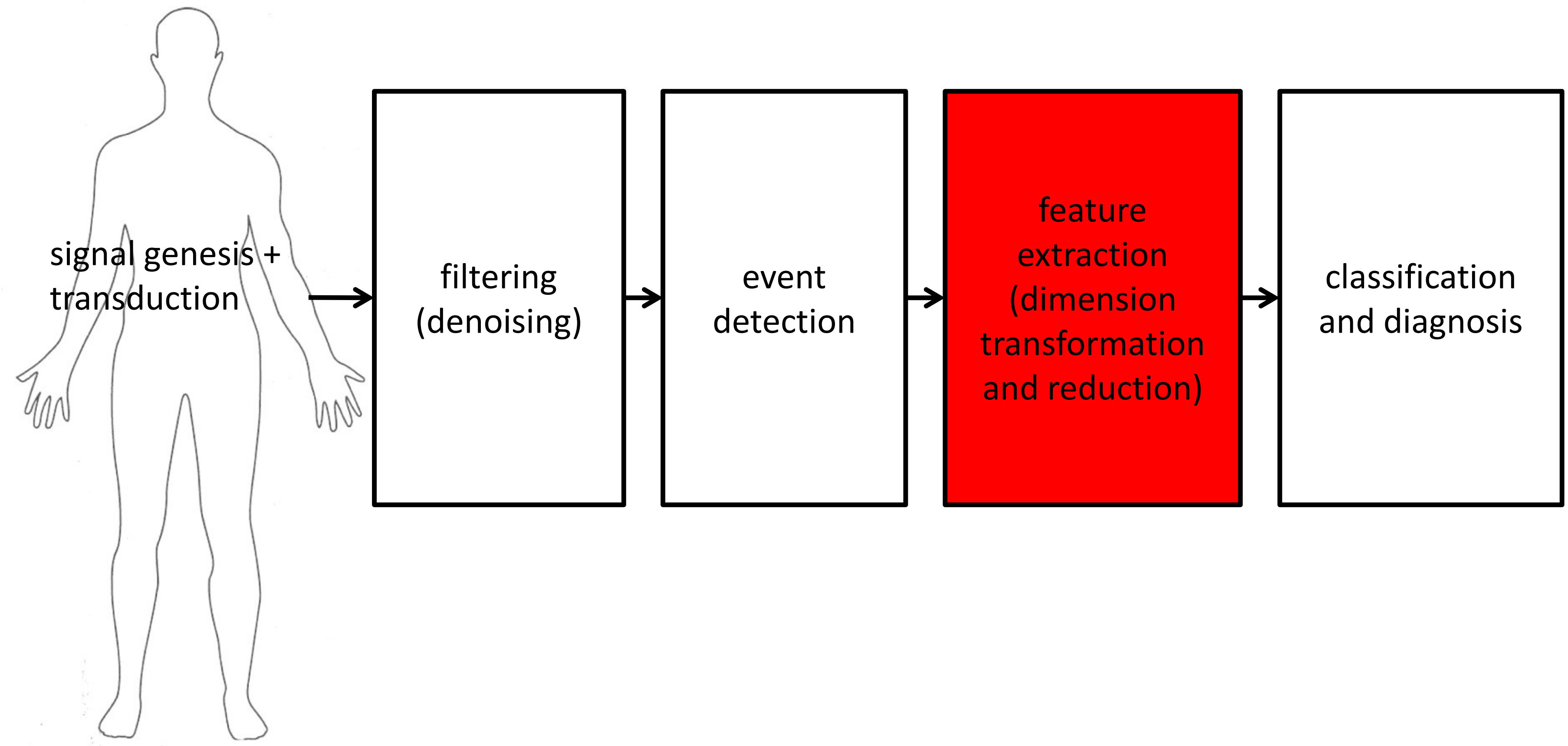
Biomedical Signal Processing
2018-2019 Autumn

Sparsity

Lecturer: Janka Hatvani
Responsible lecturer: dr. Miklós Gyöngy

The BSP Flow Chart



Wavelets and Sparsity Trilogy

Nov. 05.: Wavelets I: Time-Frequency Representation

Nov. 12.: Wavelets II: Decomposition

Nov. 19.: Sparsity

Today's goal

- **What is sparsity, and why do we seek it?**
- **Sparsity-inducing regulizer functions ($\ell_p, TV, Tikhonov$)**
- **How to solve the problem?**
  - Synthesis approach (eg. soft-thresholding)
  - Analysis approach (eg. Matching/Basis pursuit)
  - ADMM

As sparse as...

Miuku • a year ago
women in IT helpdesks
1 ^ | v • Reply • Share ›

Elaine Binney • 4 years ago
the amount of crisps left in the cupboard. I was hungry.
1 ^ | v • Reply • Share ›

Megan Thomas • 5 months ago
I wish nuts were in chocolate brownies.
^ | v • Reply • Share ›

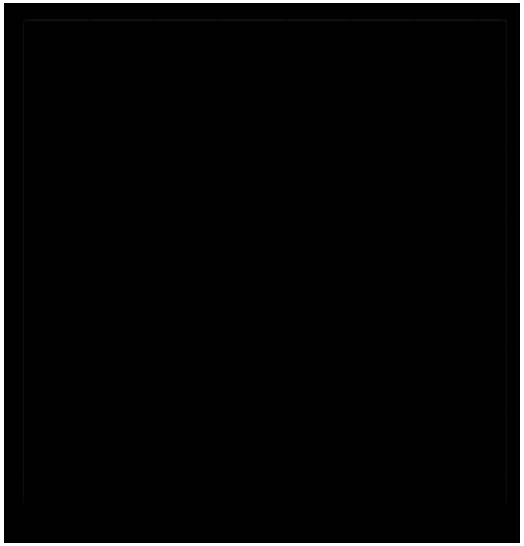
Michael Suleiman Kiarie • 8 months ago
women presidents
^ | v • Reply • Share ›

Jansie • 3 years ago
a newly waxed eyebrow
^ | v • Reply • Share ›

Erika E • a year ago
the hairs on his balding head
^ | v • Reply • Share ›

What is sparsity?

„sparse matrix or sparse array is a matrix in which most of the elements are zero"



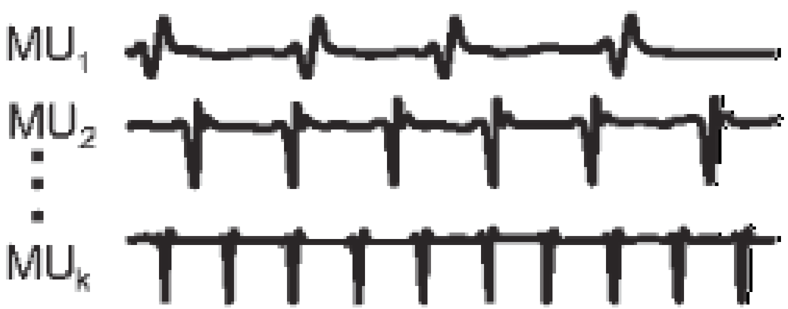
$$\begin{bmatrix} 0 & 0 & 0 & 0 & 9 & 0 \\ 0 & 8 & 0 & 0 & 0 & 0 \\ 4 & 0 & 0 & 2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 5 \\ 0 & 0 & 2 & 0 & 0 & 0 \end{bmatrix}$$

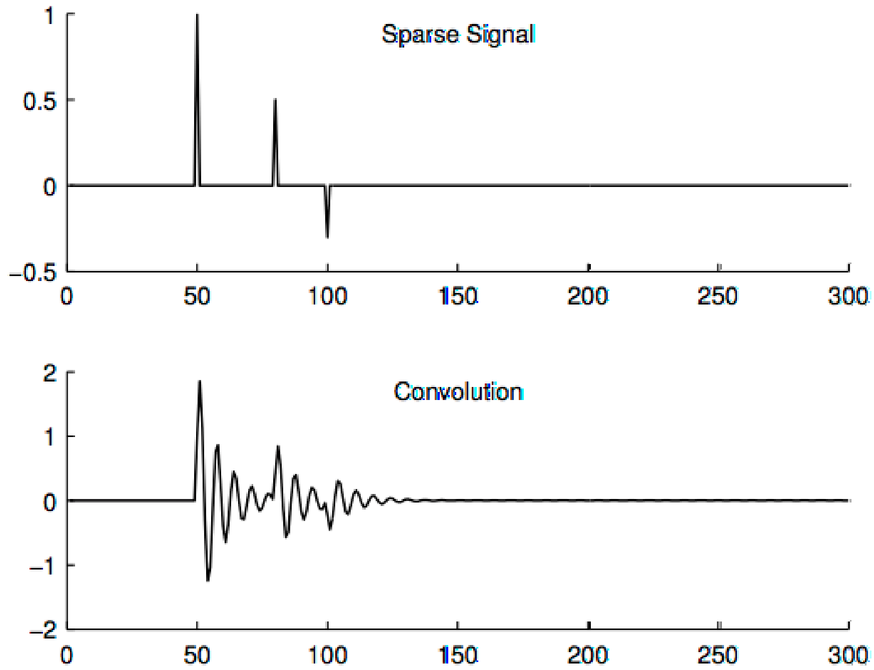
Rows	Columns	Values
5	6	6
0	4	9
1	1	8
2	0	4
2	2	2
3	5	5
4	2	2

What is sparsity?

Do you remember point processes?

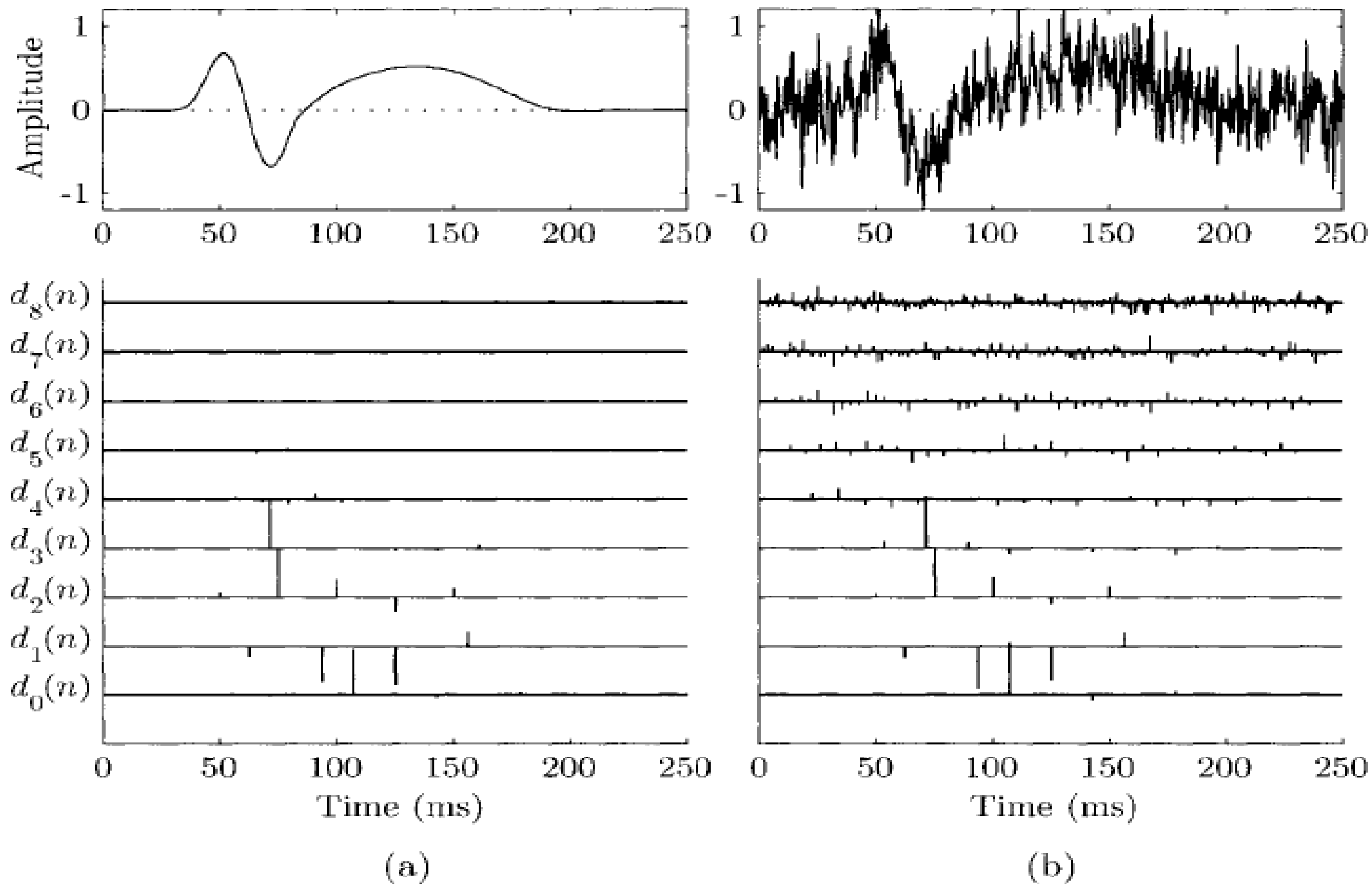
MUAP signals





What is sparsity?

And the DWT?



What is sparsity?

From last lecture:

- **„The DWT provides a sparse representation for many natural signals"**

?

„A natural signal is said to be sparse, if it can be compactly expressed as a linear combination of a few number of basis vectors."

Why sparsity?

- Data compression
- Denoising
- Superresolution, inpainting, deblurring
- Source separation (next lecture)
- Efficient data acquisition tenchniques

The inverse problem

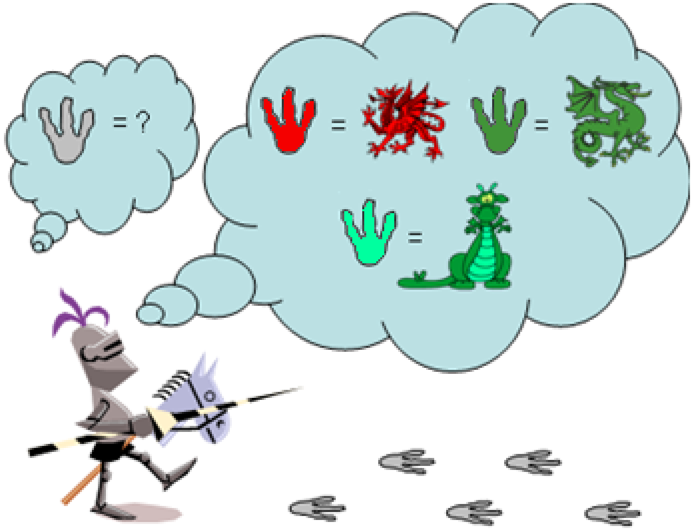
$$\boldsymbol{y} = \boldsymbol{A} \cdot \boldsymbol{x} + \boldsymbol{n}$$

This problem is underdetermined (because of the noise, and/or basis overcompleteness)

- It can have infinite number of solutions
- To choose one solution from this set, we add some regularization functions ($\phi(x)$)

The inverse problem

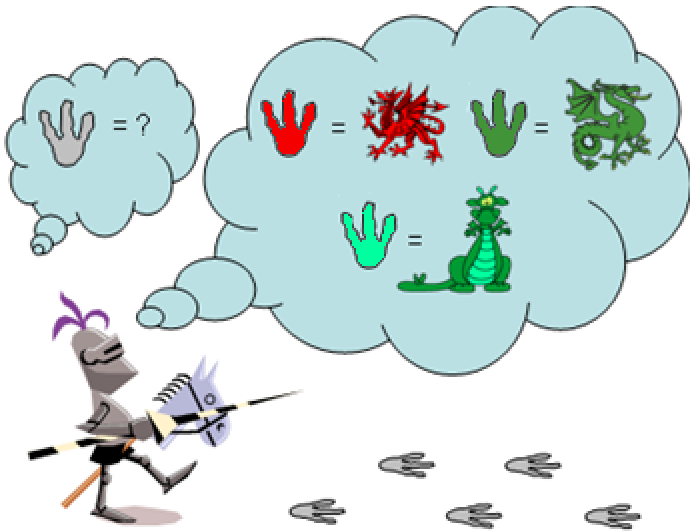
Underdetermined problem

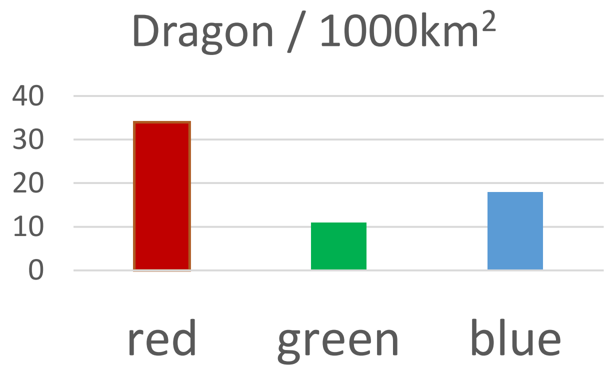


A whole set of possible solutions…

The inverse problem

Underdetermined problem





Penalize green and blue dragons, because they are less likely

$$\phi(x)$$

Is there a solution for the inverse problem? And how stable is it?

- Consider it without noise: $y = Ax$
- Using the **singular value decomposition** $A = U\Sigma V^*$
  $x = (U\Sigma V^*)^{-1}y = V^{*-1}\Sigma^{-1}U^{-1}y = V\Sigma^{-1}U^*y$
  where $\Sigma = \mathrm{diag}(\sigma_1, \sigma_2, \dots, \sigma_n)$
  so that $\Sigma^{-1} = \mathrm{diag}(1/\sigma_1, 1/\sigma_2, \dots, 1/\sigma_n)$
- There **exists** a solution, if $\boldsymbol{\sigma_i} \neq \mathbf{0}, \forall\, i \in \{1,2, \dots, N\}$
- **Stability**: if σ_i is near zero, its inverse will ‚explode'.
  Measure of stability is the **condition number**: $k = \frac{max_i(\sigma_i)}{min_i(\sigma_i)}$

How to induce sparsity?

$$\boldsymbol{y} = \boldsymbol{A} \cdot \boldsymbol{x} + \boldsymbol{n}$$

How to obtain a sparse x?

- **Synthesis approach:** using the wavelet basis A calculate an initial x. With small modifications induce sparsity (regularization, like thresholding the coefficients)
- **Analysis approach:** estimate x directly while ensuring sparsity (like Matching/Basis Pursuit)

How to add the regularizer $\phi(x)$?

- **Forward model:**

$$\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x} + \boldsymbol{n}$$

- **Synthesis approach:**

$$\min_{\boldsymbol{x}} \phi(x) \quad \text{subject to: } ||\boldsymbol{x} - \boldsymbol{x}_0||_2^2 \leq \boldsymbol{\epsilon} \quad \rightarrow \quad \min_{x} ||\boldsymbol{x} - \boldsymbol{x}_0||_2^2 + \tau\, \phi(x)$$

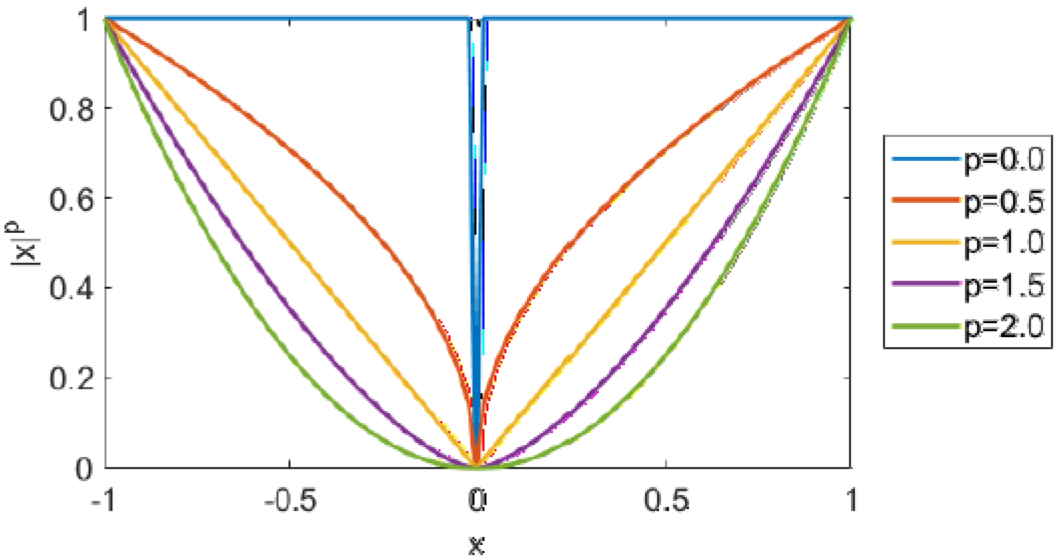
- **Analysis approach:**

$$\min_{\boldsymbol{x}} \phi(x) \quad \text{subject to: } ||\boldsymbol{y} - \boldsymbol{A}\boldsymbol{x}||_2^2 \leq \boldsymbol{\epsilon} \quad \rightarrow \quad \min_{x} ||\mathbf{y} - \mathbf{A}\mathbf{x}||_2^2 + \tau\, \phi(x)$$

- **What are** $\tau, \phi(\mathrm{x})$?

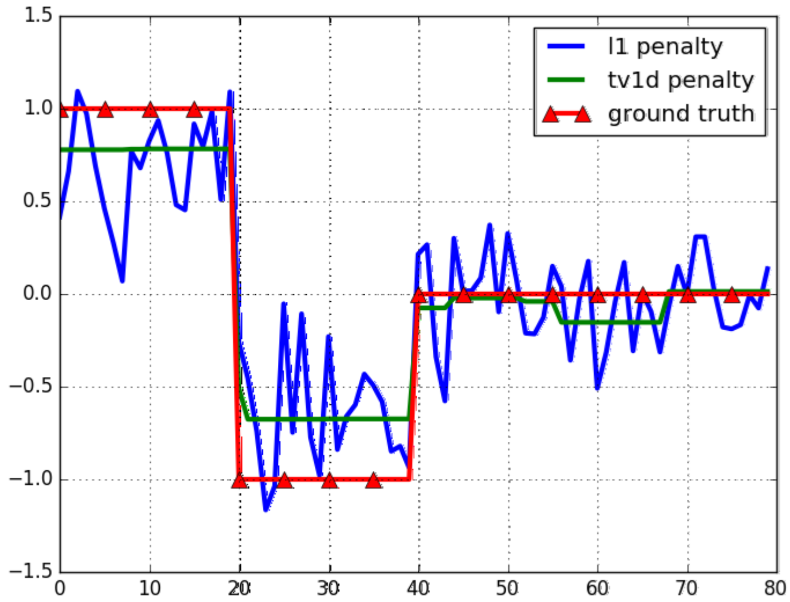
What can be a regularizer $\phi(x)$?

- $\|x\|_p^p$ for $0 \le p < 2$ is the ℓ_p-**(pseudo)norm**.
  Non-sparse solutions are penalized



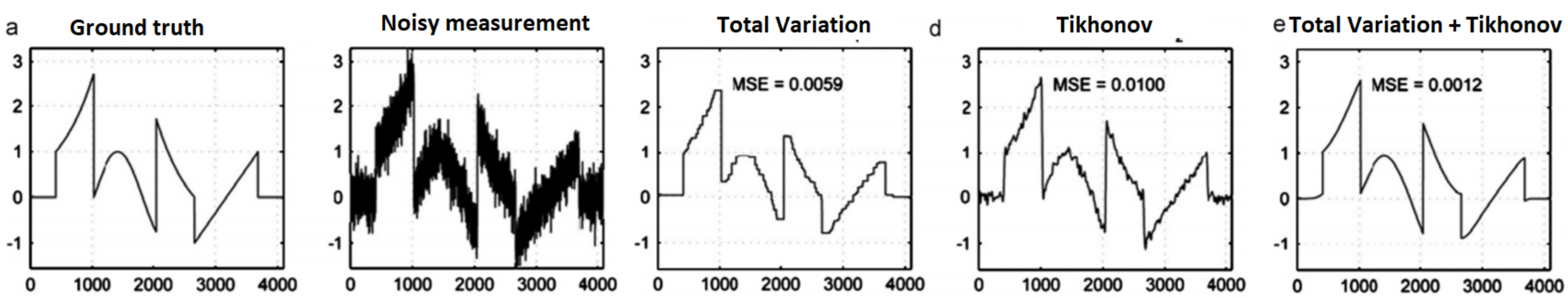
What can be a regularizer $\phi(x)$?

- $\|x\|_{TV} = \int x'(u)du$ is the total variation regulizer. It will promote piecewise constant solutions.



What can be a regularizer $\phi(x)$?

- $||\Gamma x||_2^2$, where Γ is the Tikhonov matrix, hence the name: **tikhonov-regularization**.



Solution of the *synthesis* approach

$$\min_{x} ||\boldsymbol{x} - \boldsymbol{x_0}||_2^2 + \tau\, \phi(x)$$

Solving this problem is equivalent to the so-called **Moreau Proximal mapping**.

$$\mathbf{prox}_{\tau,\phi}(\mathrm{x}_0) = \arg\min_{x}(\phi(\mathrm{x}) + \frac{1}{2\tau}\|x - x_0\|_2^2)$$

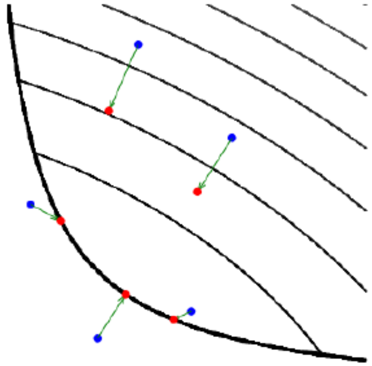


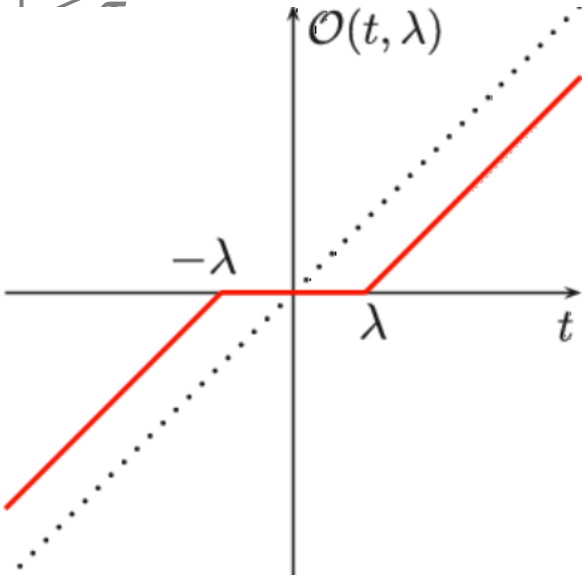
Figure: Evaluation of proximal operator at blue points moves them towards red points. Points always move to or stay inside domain of evaluated function. They also move closer to function minimum, with τ determining the extent to which the points will move towards the minimum. From Pahrik and Boyd (2012).

Solution of the *synthesis* approach

- **For most regularizer function the proximal operator has no analytic solution.**
- **One counter-example: for $\phi(x) = \|x\|_1$**

$$prox_{\tau,\phi}(x_0) = \begin{cases} x_0 - \tau & x_0 \geq \tau \\ 0 & |x \ldots \\ x_0 + \tau & \text{otl}\ldots \end{cases}$$

This is the soft-thresholding function



- **For other functions numerical solutions:**
  ISTA, FISTA, TwIST algorithms

Solution of the *analysis* approach

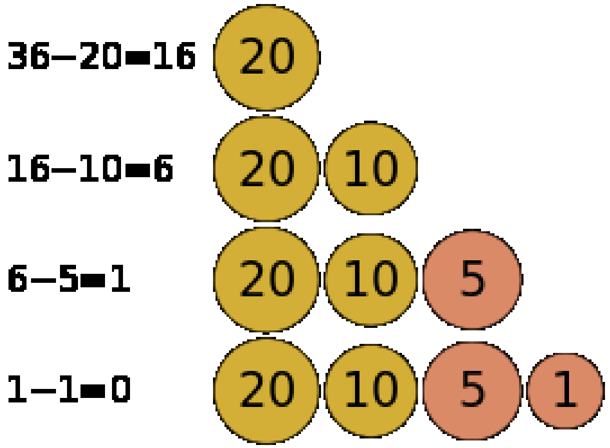
$$\min_{x} ||\mathbf{y} - \mathbf{A}\mathbf{x}||_2^2 + \tau\, \phi(x)$$

Ideas:

- ℓ_0 **(matching pursuit): find elements one by one in a greedy manner**
- ℓ_0**: relax discontinuous problem to continuous** $\ell_p; p \in [0, 1]$ **or smooth x**
- ℓ_1 **(basis pursuit): employ linear programming techniques**
- ℓ_p, ℓ_{TV}**: look for general optimization techniques to solve the problem**

Greedy optimization

- **Locally optimal** action for each iteration
- **Advantage:** conceptually **simple** and computationally **inexpensive** algorithm
- **Disadvantage:** often **suboptimal**
- Example: solving the coin-problem. You pay 0.36 €, by placing the largest coin that fits the remaining sum (residual) at each iteration.



Greedy optimization
Matching Pursuit

- **Problem:**

$$\min_{x} ||\mathbf{y} - \mathbf{Ax}||_2^2 + \tau \, ||x||_0^0$$

- **The algorithm:**
  1. Initialize the residual with your signal: $r \leftarrow y$
  2. Find the atom index j having the maximal absolute value m of inner product with your residual: $j, m \leftarrow \arg\max Ar$
  3. Substract from the residual the projection of the residual to the found atom $r = r - A_j m$
  4. Repeat 2-3. until r reaches a threshold

Greedy optimization
Orthogonal Matching Pursuit

- Apply **Gram-Schmidt orthogonalization** on signal basis of N atoms to get orthogonal basis
- Apply **MP** on orthogonal basis
- Can **convert back** to original signal basis
- **Advantage**: guaranteed to **converge** in N steps
- **Disadvantage**: sparsity will be in orthogonal basis and not in original basis – less information about the physical problem

Basis Pursuit

- **Problem in the unconstrained form:**

$$\min_x \|x\|_1^1 \text{ subject to } y = Ax$$

- **This is a support minimization problem, a solution is the Simplex method:**

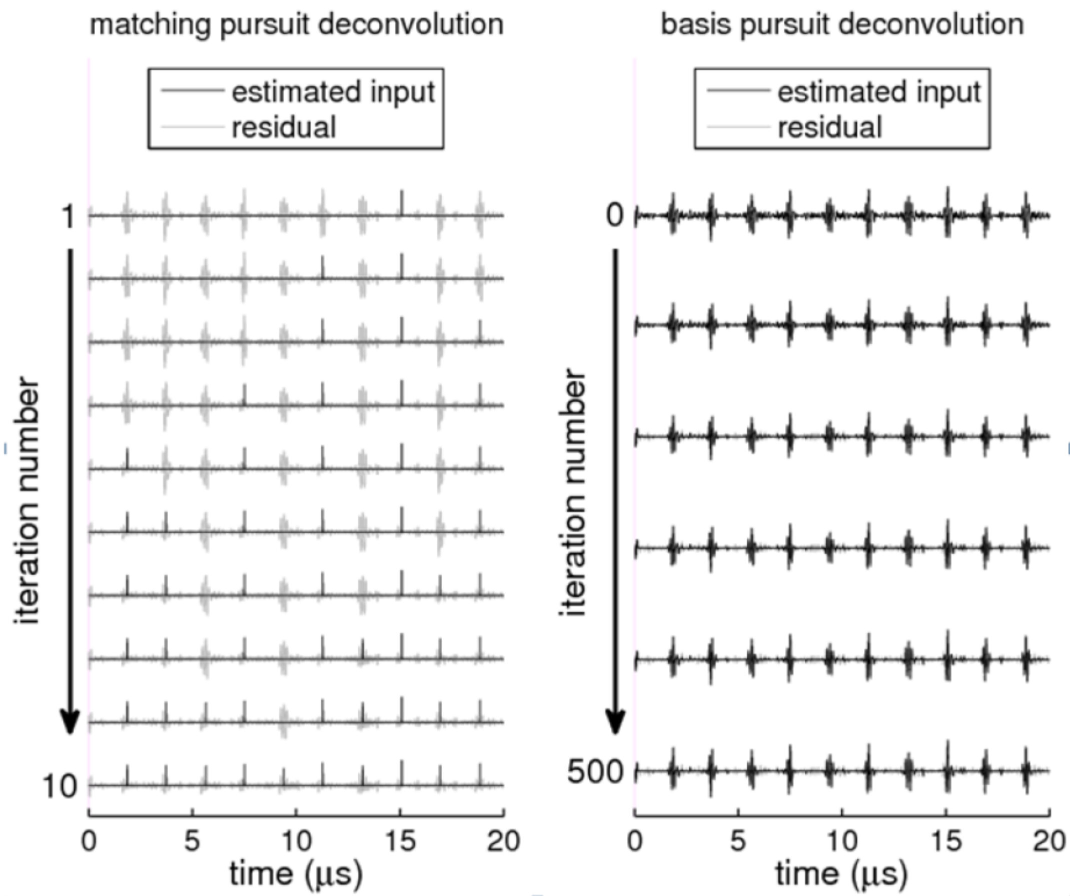
$$\|x\|_1 = \sum_i |x_i| = \sum_i (x_i^+ + x_i^-)$$

where

$$\begin{cases} x_i^+ = x_i, x_i^- = 0 & if\ x_i \geq 0 \\ x_i^+ = 0, x_i^- = x_i & if\ x_i < 0 \end{cases}$$

- **Substituting this to the original problem allows to perform linear minimization**

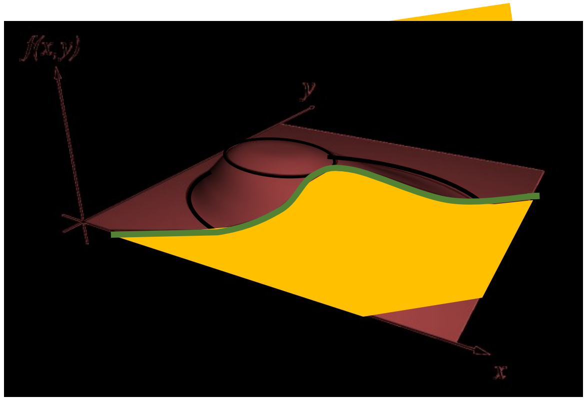
Matching Pursuit vs. Basis Pursuit



Alternating Direction Method of Multipliers

$$\min_{x} \; \|A \cdot x - y\|_2^2 + \tau\,\phi(x)$$

Too complex …



$$\min_{x,g} \; \|A \cdot x - y\|_2^2 + \tau\phi(g)$$
$$s.t. \quad g = x$$

Make it unconstrained using the augmented Lagrangian form

$$\min_{x,g} \; \|A \cdot x - y\|_2^2 + \tau\phi(g) + \lambda(x - g) + \frac{\mu}{2}\|x - g\|_2^2$$

Alternating Direction Method of Multipliers

- **The algorithm:**
  1. Minimize for x:
  $$\min_{\boldsymbol{x}} \|\boldsymbol{A} \cdot \boldsymbol{x} - \boldsymbol{y}\|_2^2 + \boldsymbol{\lambda}(\boldsymbol{x} - \boldsymbol{g}) + \frac{\boldsymbol{\mu}}{\mathbf{2}} \|\boldsymbol{x} - \boldsymbol{g}\|_2^2$$
  2. Minimize for y:
  $$\min_{\boldsymbol{x}} \boldsymbol{\tau\phi}(\boldsymbol{g}) + \boldsymbol{\lambda}(\boldsymbol{x} - \boldsymbol{g}) + \frac{\boldsymbol{\mu}}{\mathbf{2}} \|\boldsymbol{x} - \boldsymbol{g}\|_2^2$$
  3. Update the Lagrangian, $\boldsymbol{\lambda}$
  $$\lambda^{k+1} = \lambda^k + (x^k - g^k)$$
- These subproblems are easier to solve, than the original problem. Either by an **analytic solution** (like the sub-thresholding for the ℓ_1-proximal operator), or by **numerical solution** (like the gradient descent, Newton-method)

Convergence