




# Sistemi Intelligenti Stima MAP

Alberto Borghese


Università degli Studi di Milano  
Laboratory of Applied Intelligent Systems (AIS-Lab)  
Dipartimento di Informatica  
[alberto.borghese@unimi.it](mailto:alberto.borghese@unimi.it)



A.A. 2022-2023 1/64 <http://vborghese.di.unimi.it/>





## Overview

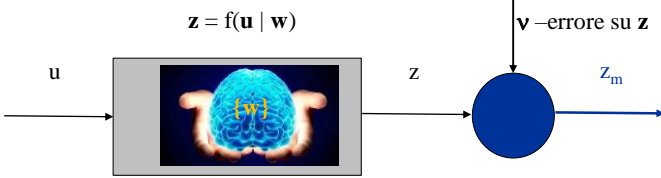


- Statistical filtering
- MAP estimate
- Different noise models
- Different regularizers

A.A. 2022-2023 2/64 <http://vborghese.di.unimi.it/>



## Statistical models

$$z = f(u | w)$$


$$v = z_m - z = z_m - f(u | w)$$

**What happens if u is extracted from a statistical distribution?**

A.A. 2022-2023
3/64
<http://borghese.di.unimi.it/>


## Teorema di Bayes

$$P(X,Y) = P(Y|X)P(X) = P(X|Y)P(Y)$$

$$P(X|Y) = \frac{P(Y|X)P(X)}{P(Y)}$$

X = causa      Y = effetto


$$P(\text{causa}|\text{effetto}) = \frac{P(\text{Effetto} | \text{Causa}) P(\text{Causa})}{P(\text{Effetto})}$$




We usually do not know the statistics of the cause, but we can measure the effect and , through frequency, build the statistics of the effect or we know it in advance.

A doctor knows  $P(\text{Symptoms}|\text{Causa})$  and wants to determine  $P(\text{Causa}|\text{Symptoms})$

A.A. 2022-2023
4/64
<http://borghese.di.unimi.it/>



## Variabili continue



Caso discreto: prescrizione della probabilità per ognuno dei finiti valori che la variabile  $X$  può assumere:  $p(x)$ .

Caso continuo: i valori che  $X$  può assumere sono infiniti. Devo trovare un modo per definirne la probabilità. Descrizione **analitica** mediante la funzione densità di probabilità.


Valgono le stesse relazioni del caso discreto, dove alla somma si sostituisce l'integrale.

$$p(x, y) = p(y|x) p(x) = p(x|y) p(y) \quad \text{Teorema di Bayes}$$


$$p(x|y) = \frac{p(y|x) p(x)}{p(y)} \quad \text{Problema Inverso}$$

$x = \text{causa}$   
 $y = \text{effetto}$

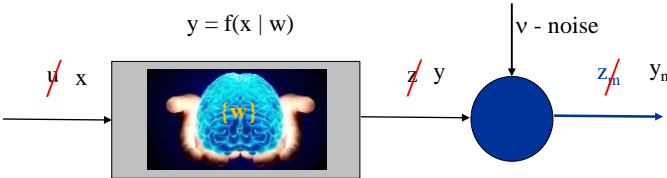
A.A. 2022-2023
5/64
<http://borghese.di.unimi.it/>



## Obiettivo




Determinare i dati (la causa,  $u$ ) più verosimile dato un insieme di misure  $z_m$ .

$y = f(x | w)$ 



$y = Ax$  se il modello è lineare

**Inverse problem: determine cause  $\{x\}$  from  $\{y_n\}, \{w\}$  – utilizzo backwards**

A.A. 2022-2023
6/64
<http://borghese.di.unimi.it/>



## Images are corrupted by noise...




i) When measurement of some physical parameter is performed, noise corruption cannot be avoided.

ii) Each pixel of a digital image measures a number of photons.

Therefore, from i) and ii)...


...Images are corrupted by noise!

How to go from noisy image to the true one? It is an inverse problem (true image is the cause, measured image is the measured effect).




A.A. 2022-2023

7/64



## Example: Filtering (denoising)




- $x = \{x_1, x_2, \dots, x_M\}$ ,  $x_k \in \mathbb{R}^M$       e.g. Pixel true luminance
- $y_n = \{y_{n1}, y_{n2}, \dots, y_{nM}\}$   $y_{nk} \in \mathbb{R}^N$       e.g. Pixel measured luminance (noisy)
- $y_n = \mathbf{I}x + n$       ->. Determining  $x$  is a **denoising problem** (the measuring device introduces only measurement error)


**Role of  $\mathbf{I}$ :**

- Identity matrix. Reproduces the input image,  $x$ , in the output  $y$ .

**Role of  $n$ :** measurement noise.

- $y_n = y + n = \mathbf{I}x + n$

$y_n$   



$y=x$   


Determining  $x$  is a denoising problem (image is a copy of the real one with the addition of noise)


A.A. 2022-2023

8/64

<http://borghese.di.unimi.it/>



## Esempio più generale (e.g. deblurring)



- $x = \{x_1, x_2, \dots, x_M\}$ ,  $x_k \in \mathbb{R}^M$                       e.g. Pixel true luminance
- $y_n = \{y_{n1}, y_{n2}, \dots, y_{nM}\}$   $y_{nk} \in \mathbb{R}^N$                       e.g. Pixel measured luminance (noisy)
- $y_n = y + n = A x + n + h$  -> determining  $x$  is a **deblurring problem** (the measuring device introduces easurement error and some blurring)
- **This is the very general equation that describes any sensor.**

**Role of  $A$ :**


- Matrix that produces the output  $y_i$  as a linear combination of other values of  $x$ .

**Role of  $h$ :** offset: background radiation (dark currents) has been compensated by calibration, regulation of the zero point.


**Role of  $n$ :** measurement noise.

- $y_n = y + n = A x + n$                       after calibration



A.A. 2022-2023
9/64
<http://borghese.di.unimi.it/>




## Gaussian noise and likelihood




- Images are composed by a set of pixels,  $x$
- Let us assume that the noise is Gaussian and that its mean and variance is equal for all pixels;
- Let us suppose that pixels are independent.
- Let  $y_{n,i}$  be the measured value for the  $i$ -th pixel ( $n$  = noise);
- Let  $x_i$  be the true (noiseless) value for the  $i$ -th pixel;
- How can we quantify the probability to measure the image  $x$ , given the probability density function for the measurement of each pixel  $y_n$ ?
- Which is the joint probability of measuring the set of pixels:  $y_{1n} \dots y_{Nn}$ ?

A.A. 2022-2023
10



## Gaussian noise and likelihood




- Images are composed by a set of pixels,  $\mathbf{x}$
- Let us assume that the noise is Gaussian and that its mean and variance is equal for all pixels;
- Let  $y_{n,i}$  be the measured value for the  $i$ -th pixel ( $n$  = noise);
- Let  $x_i$  be the true (noiseless) value for the  $i$ -th pixel;
- Let us suppose that pixels are independent.

▪ Being the pixels independent, the total probability can be written in terms of product of independent conditional probabilities (likelihood function)  $L(\mathbf{y}_n | \mathbf{x}) = L(\mathbf{y}_n | \mathbf{x})$  :


$$L(\mathbf{y}_n | \mathbf{x}) = \prod_{i=1}^N p(y_{n,i} | x_i) = \prod_{i=1}^N \frac{1}{\sigma\sqrt{2\pi}} \exp\left[-\frac{1}{2}\left(\frac{y_{n,i} - x_i}{\sigma}\right)^2\right]$$

- $L(\mathbf{y}_n | \mathbf{x})$  describes the probability to measure the image  $\mathbf{y}_n$  (its  $N$  pixels), given the noise free value for each pixel,  $\{x\}$ .
- But we do not know these values....

A.A. 2022-2023
11/64
<http://borghese.di.unimi.it/>



## Do we get anywhere?



$L$  is the likelihood function of  $Y$ , given the object  $X$ .

$$L(y_n | x) = \prod_{i=1}^N p(y_{n,i} | x_i)$$

Determine  $\{x_i\}$  such that  $L(\cdot)$  is maximized. Negative log-likelihood is usually considered to deal with sums instead of products:

$$f(\cdot) = -\log(L(\cdot)) = -\sum_{i=1}^N \ln(p(y_{n,i} | x_i))$$

$$\min(f) = \min\left\{-\sum_i \left(\ln\left(\frac{1}{\sqrt{2\pi\sigma^2}}\right) + \frac{1}{\sigma^2}(y_{ni} - f(x_i))^2\right)\right\}$$

$$\min(f) = \min\left\{-\sum_i \left(\ln\left(\frac{1}{\sqrt{2\pi\sigma^2}}\right) + \frac{1}{\sigma^2}(y_{ni} - x_i)^2\right)\right\}$$


$y = f(x) \Rightarrow y = A x$   
if  $A = I$

$y = x \Rightarrow y_n = x + n$


If the pixels are independent, the system has a single solution, that is good. The solution is  $x_i = y_{n,i}$ , not a great result (the noisy image itself)....

Can we do any better?

A.A. 2022-2023
12/64
<http://borghese.di.unimi.it/>



## A better approach



$$L(y_n | x) = \prod_{i=1}^N p(y_{n,i} | x_i)$$

We have N pixels, for each pixel we get **one** measurement.

Let us analyze the probability for each pixel:  $p(y_{n,i} | x_i)$ . If we have more measurements for each pixel. **For each pixel**, we can write:

$$p(y_{n,i,1}; p_{n,i,2}; p_{n,i,3}; \dots; p_{n,i,M} | x_i) = \prod_{k=1}^M p(y_{n,k,i} | x_i)$$


If noise is independent, Gaussian, zero mean, the best estimate of  $x_i$  is the **samples average**, this converges to the distribution mean of the measurements in the position i.

$$x_i = \sum_k y_{n,k,i}$$


The accuracy of the estimate increases with  $\sqrt[2]{N}$  with N number of samples of the same datum.

But, **what happens if we do not have such multiple samples** or we have a few samples?

A.A. 2022-2023
13/64
<http://borghese.di.unimi.it/>




## Overview




- Statistical filtering
- MAP estimate**
- Different noise models
- Different regularizators

A.A. 2022-2023
14/64
<http://borghese.di.unimi.it/>



## The Bayesian framework



We assume that the object  $x$  is a realization of the “abstract” object  $X$  that can be characterized statistically as a density probability on  $X$ .  $x$  is considered extracted randomly from  $X$  (a bit Platonic).


The probability  $p(y_n | x)$  can be viewed as a conditional probability:  $J_0 = p(y_n | x = x^*)$

That is  $x$  will follow also a probability distribution. We will have  $p(x) = \dots$


Under this condition, the probability of observing  $y_n$  can be written as the joint probability of observing both  $y_n$  and  $x$ . This is equal to the product of the conditional probability  $p(y_n | x)$  by a-priori probability on  $x$ ,  $p(x)$ :

$$p(y_n, x) = p(y_n | x)p(x)$$

A.A. 2022-2023 15/64 <http://borghese.di.unimi.it/>



## The Bayesian framework



The probability of observing  $y_n$  can be written as the joint probability of observing both  $y_n$  and  $x$  is equal to the product of the conditional probability  $p(y_n | x)$  by an a-priori probability on  $x$ ,  $p(x)$ :

$$p(y_n, x) = p(y_n | x)p(x)$$

As we are interested in determining  $x$ , **inverse problem**, we have to write the conditional probability of  $x$ , having observed (measured)  $y_n$  :  $p(x | y_n)$ . We apply Bayes theorem:

$$p(x | y_n) = \frac{p(y_n | x)p(x)}{p(y_n)} = J_0(y_n | x) \frac{p(x)}{p(y_n)}$$

where  $p(y_n | x)$  is the conditional probability:  $J_0 = p(y_n | x = x^*)$

A.A. 2022-2023 16/64 <http://borghese.di.unimi.it/>





## A-priori types - $p(x)$



$p(x)$  describes the probability of having a certain type of data  $X$ . In this case it describes the probability of having one image or another.

- It can be the amplitude of the signal defined in terms of power.
- It can be the structure defined in terms of variations (gradients)
- It can be information gathered from the neighbour data (e.g. clique).
- Any statistical information on the distribution of  $x$ .
- It can be a morphable model
- .....




## MAP Estimate




Vogliamo trovare il valore di  $x$  più probabile congruente con le misure  $\{x, y\}$ :

$$\max_x \{p(x|y_n)\} = \max_x p(y_n|x) \frac{p(x)}{p(y_n)}$$

$$\max_x \{p(x|y_n)\} = \max_x \prod_i p(y_{n,i}|x_i) \frac{p(x_i)}{p(y_{n,i})}$$



## MAP Estimate



$$p(x | y_n) = \frac{p(y_n | x)p(x)}{p(y_n)} = L(y_n | x) \frac{p(x)}{p(y_n)}$$

$\ln(ab/c) = \ln(a) + \ln(b) - \ln(c)$

Logarithms help:


$$-\ln(p(x | y_n)) = -\ln\left\{\frac{p(y_n | x)p(x)}{p(y_n)}\right\} = -\{\ln(p(y_n | x)) + \ln(p(x)) - \ln(p(y_n))\}$$

We maximize the  $p(x | y_n)$ , by minimizing:


$$\arg \min_x \left\{ \ln\left(\frac{p(y_n | x)p(x)}{p(y_n)}\right) \right\} = \arg \min_x \left\{ \ln(p(y_n | x)) + \ln(p(x)) - \ln(p(y_n)) \right\}$$

We explicitly observe that the marginal distribution of  $y_n$ ,  $p(y_n)$ , is not dependent on  $x$ . It does not affect the minimization and it can be neglected. It represents the statistical distribution of the measurements alone, implicitly considering all the possible  $x$  values.

**Maximizing  $p(x | y_n)$  is called Maximum A-Posteriori Estimate – MAP (we collect the measurements  $y_n$  and then we estimate  $x$  taking into account also the information on  $x$ ).**



## MAP estimate components



We maximize the MAP of  $p(x | y_n)$ , by minimizing:


$$\arg \min_x \left\{ \ln(p(y_n | x)p(x)) \right\} = \arg \min_x \left\{ \ln(p(y_n | x)) + \ln(p(x)) \right\}$$

$J_0(y_{n,i} | x)$   
 Adherence to the data for  
 each  $x$  value (conditional  
 probability)


$J_R(x)$   
 A-priori  
 probability on  $x$

Depending on the shape of the noise (inside the joint probability) and the a-priori distribution of  $x(\cdot)$ ,  $J_R(x)$ , we get different solutions.

A.A. 2022-2023
20/64
<http://borghese.di.unimi.it/>



## Gaussian noise on samples



$$x = \arg \min_x - \{\ln(p(y_n | x)p(x))\} = \arg \min_x - \{\ln(p(y_n | x)) + \ln(p(x))\} =$$

$$\arg \min_x \{J_0(y_n | x) + J_R(x)\} =$$

- Gaussian noise on the data
- Zero mean
- Pixels are independent
- All measurements have the same variance,  $\sigma_0^2$
- $y = Ax$  – deblurring problem ( $A \neq I$ )


$$J_0(y_n | x) = \text{const} + \left(\frac{1}{\sigma^2}\right) \left(\sum_i \|y_{n,i} - Ax_i\|^2\right)$$

$$-\log(p(y_n | x)) = J_0(y_n | x) = \text{const} + \left(\frac{1}{\sigma^2}\right) \sum_i \|Ax_i - y_{n,i}\|^2$$


Mean squared error

What about  $J_R(x) = -\log(p(x))$ ?

A.A. 2022-2023
21/64
<http://borghese.di.unimi.it/>



## Gibb's priors for $p(x)$



We often define the a-priori term,  $J_R(x)$ , as Gibb's prior:

$$p_x = \frac{1}{Z} \left\{ e^{\left(-\frac{1}{\beta} U(x)\right)} \right\}$$

$$Z = \int_{-\infty}^{+\infty} e^{-\frac{1}{\beta} U(x)} dx$$

Integrale = 1


$U(x)$  è solitamente  $\geq 0$

E' una funzione esponenziale decrescente che è massima quando  $U(x)$  è minima  
(max  $e^{-U(x)}$  si ha quando  $U(x) = 0$ )


$U(x)$  sarà perciò minimo per le realizzazioni di  $x$  (dell'immagine) più probabili.

$U(x)$  è chiamato anche potenziale => potenziale minimo per realizzazioni più probabili.

A.A. 2022-2023
22/64
<http://borghese.di.unimi.it/>



## Gibb's priors for $p(x)$



We often define the a-priori term,  $J_R(x)$ , as Gibb's prior:

$$p_x = \frac{1}{Z} \left\{ e^{\left( -\frac{1}{\beta} U(x) \right)} \right\} \qquad Z = \int_{-\infty}^{+\infty} e^{\frac{-1}{\beta} U(x)} dx$$

Considerando il negativo del logaritmo di  $p(x)$ :

$$J_R(x) = -\ln(p_x) = +\ln(Z) + \frac{1}{\beta} U(x)$$


$\Rightarrow J_R(x)$  is a linear function of the potential  $U(x)$ . It is minimum when  $U(x)$  is minimum.

$Z$  does not depend on  $x \Rightarrow$  it is constant


$\beta$  is a constant that provides a scale to  $J_R(x)$ .

$\beta$  Explains how  $p(x)$  decreases with the decrease of the probability of  $x$ , described by  $U(x)$ .

A.A. 2022-2023
23/64
<http://borghese.di.unimi.it/>



## MAP estimate components



We maximize the MAP of  $p(x | y_n)$ , by minimizing:

$$\arg \min_x -\{\ln(p(y_n | x) p(x))\} = \arg \min_x -\{ \underbrace{\ln(p(y_n | x))}_{J_0(y_{n,i} | x)} + \underbrace{\ln(p(x))}_{J_R(x)} \}$$

$J_0(y_{n,i} | x)$


Adherence to the data for  
each  $x$  value (conditional  
probability)

$J_R(x)$


A-priori  
probability on  $x$

Depending on the shape of the noise (inside the joint probability) and the a-priori distribution of  $x(\cdot)$ ,  $J_R(x)$ , we get different solutions.

A.A. 2022-2023
24/64
<http://borghese.di.unimi.it/>



## P(x) in the Ridge regression



We choose as a-priori term the squared norm of the function x, weighted by P:  $U(x) = \|Px^2\|$


$$p(x) = \frac{1}{Z} \left\{ e^{\left( -\frac{1}{\beta} \|Px\|^2 \right)} \right\} \quad J_R(x) = -\log(p(x)) = \log(Z) + (1/\beta) \|Px^2\|$$

Nel caso del filtraggio:  $P = I$ , peso tutti i pixel dell'immagine allo stesso modo ( $P = I$ )


$$J_R(x) = \log(Z) + (1/\beta) \|x^2\|$$

Non voglio pixel che “sparino” – non voglio avere dati con valori troppo più elevati degli altri, questi sono improbabili (alto potenziale  $U(x)$ , basso valore di  $J_R(x)$ ).

A.A. 2022-2023
25/64
<http://borghese.di.unimi.it/>



## Map estimate with $U(x) = \|Px\|^2$




$$x = \underset{x}{\operatorname{argmin}} \left( \sum_i \|Ax_i - y_{n,i}\|^2 + \frac{1}{\beta} \sum_i \|p_{ii}x_i\|^2 \right) \quad \text{Funzione costo quadratica}$$


$J_0(y_{n,i} | x)$   
 Adherence to the data for  
 each x value (conditional  
 probability)

$J_R(x)$   
 A-priori  
 probability on x

A.A. 2022-2023
26/64
<http://borghese.di.unimi.it/>



## MAP estimate with $U(x) = \|Px\|^2$



$$x = \underset{x}{\operatorname{argmin}} \left( \sum_i \|Ax_i - y_{n,i}\|^2 + \frac{1}{\beta} \sum_i \|p_{ii}x_i\|^2 \right) \quad \text{Funzione costo quadratica}$$

Derivo rispetto a  $x$  per calcolare il minimo:

$$x : A^T y_n - A^T A x - \lambda P^T P x = 0 \quad \Rightarrow \quad A^T y_n = (A^T A + \lambda P^T P)x$$


$J_0(y_{n,i} | x)$

$J_R(x)$


Pongo  $\lambda = 1/\beta$

Without  $\lambda P^T P$  large values of  $x$  are obtained where  $A^T A$  is small. These are reduced by  $\lambda P^T P$

A.A. 2022-2023 27/64 <http://borghese.di.unimi.it/>



## Map estimate with $U(x) = \|Px\|^2$





$$x = \underset{x}{\operatorname{argmin}} \left( \sum_i \|Ax_i - y_{n,i}\|^2 + \frac{1}{\beta} \sum_i \|p_{ii}x_i\|^2 \right) \quad \text{Funzione costo quadratica}$$

$$x : A^T y_n - A^T A x - \lambda P^T P x = 0 \quad \Rightarrow \quad A^T y_n = (A^T A + \lambda P^T P)x$$

$$x = (A^T A + \lambda P^T P)^{-1} A^T y_n \quad \text{---}$$

(diventa risolubile anche quando  $A$  è singolare! – norma minima della soluzione)  
 (ottengo una soluzione che «scoraggia» i valori elevati di  $x$ ).  
 (per  $\lambda = 0$  ritorno alla soluzione con la pseudo-inversa, massima verosimiglianza;  
 non tengo conto del termine a-priori).

A.A. 2022-2023 28/64 <http://borghese.di.unimi.it/>

## Approccio algebrico

$$\mathbf{A} \mathbf{x} = \mathbf{b} + \mathbf{N} \quad \sum_k v_k^2 = \|\mathbf{A}x - \mathbf{b}\|^2$$

$$x = \underset{x}{\operatorname{argmin}} \left( \sum_i \|A_{*,i}x_i - y_{n,i}\|^2 \right) \implies x = (\mathbf{A}^T \mathbf{A})^{-1} \mathbf{A}^T y_n$$



Se la matrice di covarianza ha determinante vicino a zero (è mal condizionata) la soluzione può variare molto con il variare dei dati. **Otteniamo un modello la cui validità al di fuori dei dati già disponibili è molto limitata sui dati che si potranno collezionare in futuro.**

**Problema mal posto (Hadamard).**

- Esiste una soluzione
- La soluzione è unica
- **Varia con continuità con i dati.**

Come possiamo stabilizzare la soluzione?

A.A. 2022-2023 29/64 <http://borghese.di.unimi.it/>

## Approccio algebrico: regolarizzazione

$$\mathbf{A} \mathbf{x} = \mathbf{b} + \mathbf{N} \quad \sum_k v_k^2 = \|\mathbf{A}x - \mathbf{b}\|^2$$

$$x = \underset{x}{\operatorname{argmin}} \left( \sum_i \|A_{*,i}x_i - y_{n,i}\|^2 \right) \implies x = (\mathbf{A}^T \mathbf{A})^{-1} \mathbf{A}^T y_n$$

We add a penalty term to the solution that expresses the desired characteristics of the solution.

$$x = \underset{x}{\operatorname{argmin}} \left( \sum_i \|Ax_i - y_{n,i}\|^2 + \lambda \sum_i \|Px_i\|^2 \right)$$

This is the Tikhonov regularization (1963).

**It is the same cost function obtained when maximizing the MAP with Gibbs prior and quadratic potential function.**

A.A. 2022-2023 30/64 <http://borghese.di.unimi.it/>



## Which is the most adequate $p(x)$ for images?



We are very interested to borders, structure. This has to deal with **gradients**.  
=> we look at **differential properties**.

We look at the local gradient of the image:  $\nabla x$  (variazioni spaziali).

One possibility is to use the square of the gradient as a regularizer (o come funzione potenziale):  $\|\nabla x\|^2$

This is another form of Tikhonov regularization.



## Differential Gibbs prior



$$p_x = \frac{1}{Z} \left\{ e^{\left( -\frac{1}{\beta} U(x) \right)} \right\}$$

$$Z = \int_{-\infty}^{+\infty} e^{-\frac{1}{\beta} U(x)} dx$$


$$U(x) = \|\nabla x\|^2$$

$$\arg \min_x \left\{ \|(Ax - y_n)\|^2 + \lambda \|\nabla x\|^2 \right\}$$


$$x: \left\{ 2A^T(Ax - y_n) + 2\lambda \nabla x \right\} = 0$$

System of M linear differential equations. How does it become in the discrete case?





## Differential Gibbs prior



$$\arg \min_x \left\{ \|Ax - y_n\|^2 + \lambda \|\nabla x\|^2 \right\}$$

$$x: \left\{ 2A^T(Ax - y_n) + 2\lambda \nabla x \right\} = 0$$


If we approximate  $\nabla x$  with the finite differences, one possibility is the following:

$$\|\nabla x_{i,j}\|^2 = (x_{i+1,j} - x_{i-1,j})^2 + (x_{i,j+1} - x_{i,j-1})^2 \quad \text{Centered discrete gradient}$$


$$\arg \min_x \left\{ \sum_j \sum_i (A_{ji}x_i - y_j)^2 + \lambda \left( (x_{i,j+1} - x_{i,j-1})^2 + (x_{i+1,j} - x_{i-1,j})^2 \right) \right\}$$


Si può calcolare la derivate della somma, derivando per ciascun elemento  $x$  e ponendo la derivate uguale a zero. Diventa un sistema lineare.

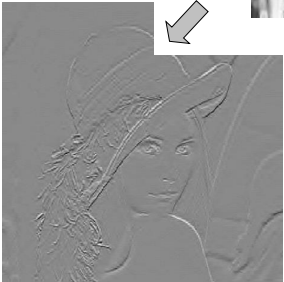
A.A. 2022-2023
33/64
<http://vborghese.di.unimi.it/>

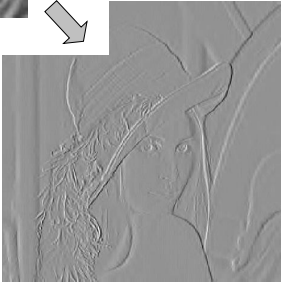


## A priori term - image gradients (no noise)







$p_x = p(i,j) - p(i-1,j)$ 



$p_y = p(i,j) - p(i,j-1)$ 


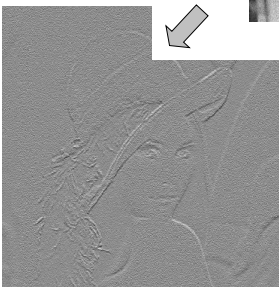
A.A. 2022-2023
34/64
<http://vborghese.di.unimi.it/>

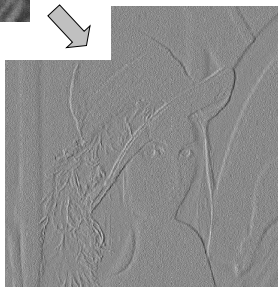


## A priori term - image gradients (with noise)







$$\Delta x_{row} = \frac{x_{i+1,j} - x_{i-1,j}}{2}$$


$$\Delta x_{col} = \frac{x_{i,j+1} - x_{i,j-1}}{2}$$



A.A. 2022-2023
35/64
<http://vborghese.di.unimi.it/>



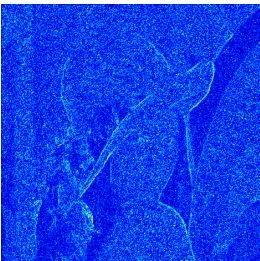
## A priori term - norm of image gradient



No noise




Noise




In the real image, most of the areas are characterized by an (almost) null gradient norm. When noise is added, local gradients appear everywhere in the image (real case).

**We can for instance suppose that the noise is a random variable with Gaussian distribution, zero mean and variance equal to  $\beta^2$  (sampling noise).**

A.A. 2022-2023
36/64
<http://vborghese.di.unimi.it/>



## MAP estimate components



We maximize the MAP of  $p(x | y_n)$ , by minimizing:


$$\arg \min_x - \{\ln(p(y_n | x)p(x))\} = \arg \min_x - \{\ln(p(y_n | x)) + \ln(p(x))\}$$

$J_0(y_{n,i} | x)$   
 Adherence to the data for  
 each x value (conditional  
 probability)


$J_R(x)$   
 A-priori  
 probability on x

$$\arg \min_x \left\{ \sum_j \sum_i (A_{ji}x_i - y_j)^2 + \lambda \left( (x_{i,j+1} - x_{i,j-1})^2 + (x_{i+1,j} - x_{i-1,j})^2 \right) \right\}$$

A.A. 2022-2023
37/64
<http://borghese.di.unimi.it/>



## Tikhonov regularization




$$x = \operatorname{argmin}_x \left( \sum_i \|Ax_i - y_{n,i}\|^2 + \lambda \sum_i \|Px_i\|^2 \right)$$

$$x = \operatorname{argmin}_x \left( \sum_i \|Ax_i - y_{n,i}\|^2 + \lambda \sum_i \|\nabla x_i\|^2 \right)$$


It is a quadratic cost function. We find  $x$  minimizing with respect to  $x$  the cost function.

This approach is derived in the domain of mathematics. It leads to the same cost function of the MAP approach.

A.A. 2022-2023
38/64
<http://borghese.di.unimi.it/>



# Overview



Statistical filtering

MAP estimate


Different noise models

Different regularizers


A.A. 2022-2023

39/64

<http://borghese.di.unimi.it/>



## Departing from Tikhonov regularization: different noise models



$$\arg \min_x -\{\ln(p(y_n | x)p(x))\} = \arg \min_x -\{\ln(p(y_n | x)) + \ln(p(x))\}$$

$J_0(y_{n,i} | x)$

$J_R(x)$

Adherence to the data for each x value (conditional probability)


Two actors:

- $J_0(y_{n,i} | x)$  Conditional probability of having the measurements given a certain input.
  - **We can have different noise models.**
- $J_R(x)$  Probability of having a certain solution.
  - **We can have different regularizers**


A.A. 2022-2023

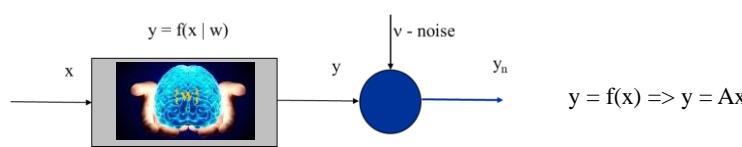
40/64

<http://borghese.di.unimi.it/>



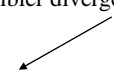
## Different noise models






	<b>Gaussian noise:</b>	Square regularization
Tikhonov	$J_0(y_{n,i}   x) = \ Ax - b\ ^2$	$J_R(x) = (1/\beta) \ Px\ ^2$
Ridge regression	$J_0(y_{n,i}   x) = \ Ax - b\ ^2$	$J_R(x) = (1/\beta) \ x\ ^2$


  

	<b>Poisson noise:</b>	Kullback-Leibler divergence
	$J_0(y_{n,i}   x) = \sum_i y_{n,i} \ln \left( \frac{y_{n,i}}{Ax} + Ax_i - y_{n,i} \right)$	

A.A. 2022-2023
41/64
<http://borghese.di.unimi.it/>



## KL and the Poisson noise



$v_i = \|Ax - y_{ni}\|$

We know the statistical distribution of the noise inside the conditional probability of  $y_{ni}$  given  $x$ .

For one pixel:  $p(y_{ni}, x_i) = \left\{ \frac{e^{-Ax_i} (Ax_i)^{y_{ni}}}{y_{ni}!} \right\}$

$$-\ln(L(y_n; x)) = -\ln \left( \prod_{i=1}^N p(y_{n,i}; x_i) \right) = -\sum_{i=1}^N (-Ax_i + y_{n,i} \ln(Ax_i) - \ln(y_{n,i}!))$$

To eliminate the factorial term, we normalize the likelihood by  $L(y_n, y_n)$ :

$$-\ln \left( \frac{L(y_n, x)}{L(y_n, y_n)} \right) = -\sum_{i=1}^N (y_n \ln(Ax) - \ln(y_n) + y_n - Ax) = \text{KL divergence}$$

$$= \sum_i y_n \ln \left( \frac{y_n}{Ax} + Ax - y_n \right)$$

It is not a distance!

It is not linear

A.A. 2022-2023
42/64
<http://borghese.di.unimi.it/>

## Tikhonov regularization - simulations

Edge smoothing effect with Tikhonov-like regularizator ( $P(x) = \|\nabla x\|^2$ )  
 Poisson noise on the image –  $\lambda = 0.5$ . **KL is applied in the first term.**  
 P is the gradient operator

A.A. 2022-2023 43/64 <http://vborgnese.di.unimi.it/>

## Tikhonov regularization - panoramic images

Edge smoothing effect with Tikhonov-like regularization  
 Poisson noise model -  $\lambda = 0.5$ . **KL is applied in the first term.**  
 P is the gradient operator

A.A. 2022-2023 44/64 <http://vborgnese.di.unimi.it/>

## Tikhonov regularization - endo-oral images


Edge smoothing effect with Tikhonov-like regularization  
 Poisson noise model -  $\lambda = 0.1$  - **KL is applied in the first term.**  
 P is the gradient operator

A.A. 2022-2023 45/64 <http://vborghese.di.unimi.it/>


## Overview

- Statistical filtering
- MAP estimate
- Different noise models
- Different regularizators**
- A-priori and Markov Random Fields
- Cost function minimization

A.A. 2022-2023 46/64 <http://vborghese.di.unimi.it/>



## Departing from Tikhonov regularization: different regularizers



$$\arg \min_x -\{\ln(p(y_n | x)p(x))\} = \arg \min_x -\{\ln(p(y_n | x)) + \ln(p(x))\}$$

$J_0(y_{n,i} | x)$


$J_R(x)$

Adherence to the data for each x value (conditional probability)


Two actors:

- $J_0(y_{n,i} | x)$  Conditional probability of having the measurements given a certain input.
  - **We can have different noise models.**
- $J_R(x)$  - Probability of having a certain solution.
  - **We can have different regularizers**

A.A. 2022-2023
47/64
<http://borghese.di.unimi.it/>



## Non-quadratic a-priori: norm $l_2$



$$\arg \min_x -\{\ln(p(y_n | x)p(x))\} = \arg \min_x -\{\ln(p(y_n | x)) + \ln(p(x))\}$$

$J_0(y_{n,i} | x)$

$J_R(x)$


Adherence to the data for each x value (conditional probability)

$$J_R(x) = \sum_i^2 \sqrt{x_1^2 + x_2^2 + \dots + x_N^2}$$


Norma  $l_2$  di x  
Il modulo di x è minimo.

A.A. 2022-2023
48/64
<http://borghese.di.unimi.it/>





## Non-quadratic a-priori: total variation




$$\arg \min_x -\{\ln(p(y_n | x)p(x))\} = \arg \min_x -\{J_0(y_{n,i} | x) + J_R(x)\}$$

Adherence to the data for each x value (conditional probability)


$$J_R(x) = \sqrt[2]{\Delta x_1^2 + \Delta x_2^2 + \dots \Delta x_N^2}$$

Norma  $l_2$  delle variazioni di x o variazione totale di x (**total variation**)  
 Il modulo della somma degli incrementi (in valore assoluto) di x è minima.

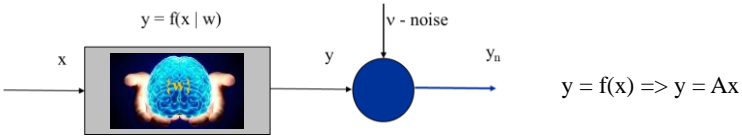
A.A. 2022-2023
49/64
<http://borghese.di.unimi.it/>



## Different a-priori



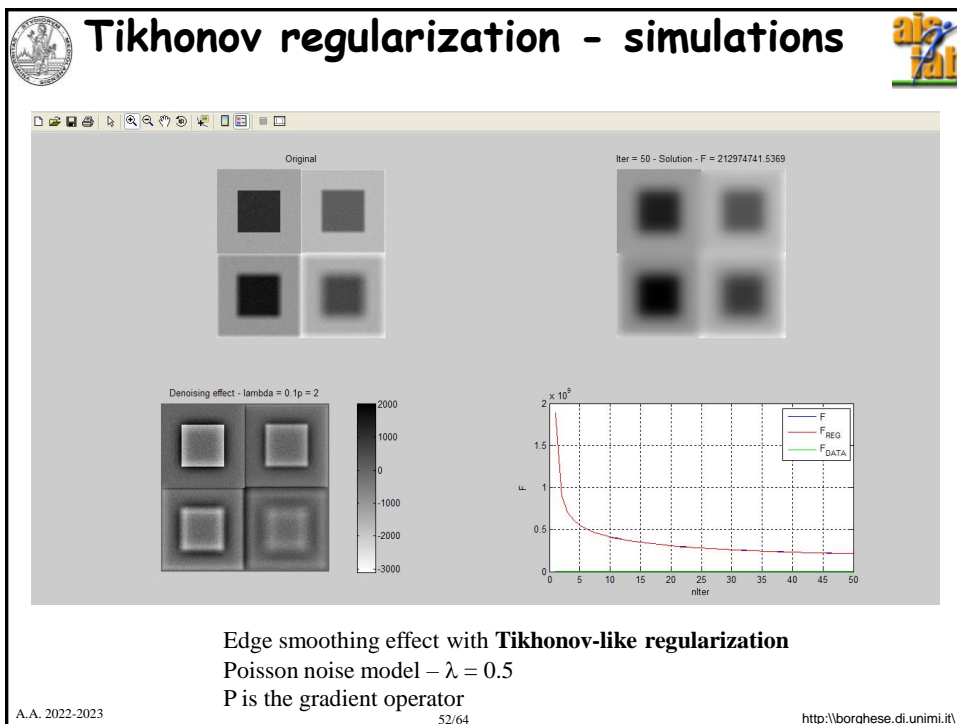
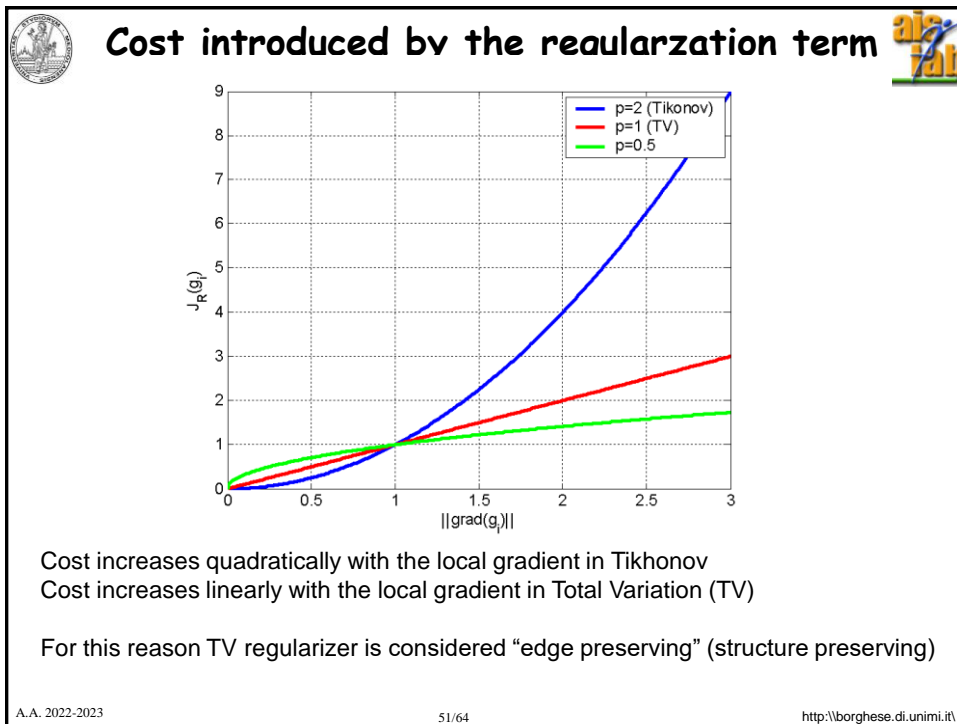
$y = f(x | w)$



$y = f(x) \Rightarrow y = Ax$

	Gaussian noise:	Square regularization
Tikhonov	$J_0(y_{n,i}   x) = \ Ax - b\ ^2$	$J_R(x) = (1/\beta) \ Px\ ^2$
Ridge regression	$J_0(y_{n,i}   x) = \ Ax - b\ ^2$	$J_R(x) = (1/\beta) \ x\ ^2$
$l_2$ (total variation) regularization	$J_0(y_{n,i}   x) = \ Ax - b\ ^2$	$J_R(x) = (1/\beta) \sqrt{\Delta x_1^2 + \Delta x_2^2 + \dots \Delta x_N^2}$
Lasso regression	$J_0(y_{n,i}   x) = \ Ax - b\ ^2$	$J_R(x) = (1/\beta) ( \Delta x_1  +  \Delta x_2  + \dots +  \Delta x_N )$

A.A. 2022-2023
50/64
<http://borghese.di.unimi.it/>



## Total variation regularization - simulations

Original

Iter = 50 - Solution - F = 1767624.3724

Denoising effect - lambda = 0.1p = 1

No appreciable edge smoothing with **total variation** regularizer  
Poisson noise model -  $\lambda = 0.5$   
P is the gradient operator

A.A. 2022-2023 53/64 <http://vborghese.di.unimi.it/>

## Tikhonov regularization - panoramic images

Original

Iter = 20 - Solution - F = 120825433.9031

Denoising effect - lambda = 0.5p = 2

Edge smoothing effect with **Tikhonov-like** regularization  
Poisson noise model -  $\lambda = 0.5$   
P is the gradient operator

A.A. 2022-2023 54/64 <http://vborghese.di.unimi.it/>

## Total variation regularization - panoramic images


No appreciable edge smoothing with **total variation** regularizer  
 Poisson noise model -  $\lambda = 0.5$   
 $P$  is the gradient operator

A.A. 2022-2023 55/64 <http://vborgnese.di.unimi.it/>


## Tikhonov regularization - endo-oral images

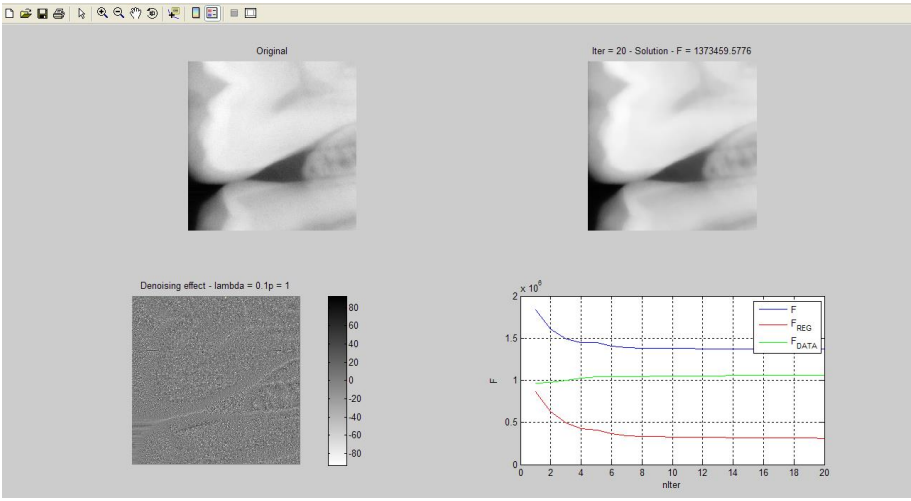
Edge smoothing effect with **Tikhonov-like** regularization  
 Poisson noise model -  $\lambda = 0.1$   
 $P$  is the gradient operator

A.A. 2022-2023 56/64 <http://vborgnese.di.unimi.it/>




## Total variation - endo-oral images






No appreciable edge smoothing with **total variation** regularizer  
Poisson noise model -  $\lambda = 0.1$   
P is the gradient operator

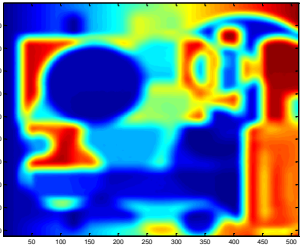
A.A. 2022-2023
57/64
<http://vborghese.di.unimi.it/>



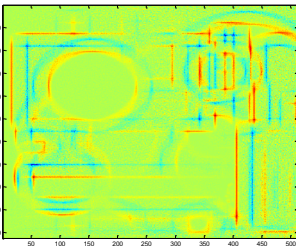
## Tikhonov vs. TV (preview)



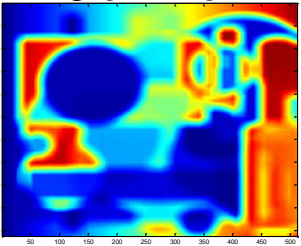
Filtered image



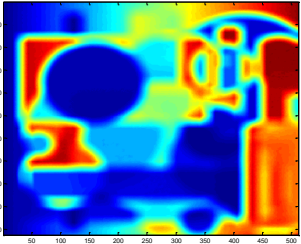
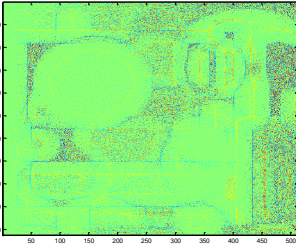
Difference



Tikhonov =>



TV =>

A.A. 2022-2023
58/64
<http://vborghese.di.unimi.it/>



## Open problems in TV



- Better images with TV regularizer, but:

Non linear cost functions (non quadratic) also with Gaussian noise model

$$x = \arg \min_x \left( \sum_i \|y_n - Ax\|^2 + \lambda \sqrt{\sum_p x_{p,i}^2} \right)$$

Minimization does not lead to a linear function (because of the square root) →  
It requires non-linear iterative minimization.

The derivative of a square root provides a function of the type  $k/\sqrt{\cdot}$

Singularity in  $x = 0 \rightarrow x \neq 0$

We can use algorithms for constrained minimization (solution should stay inside the first quadrant, e.g. split gradient).

A.A. 2022-2023

59/64

<http://borghese.di.unimi.it/>

## How to set the regularization parameter ( $\lambda = 1/\beta$ )



$$J(f) = J_o(f) + \lambda J_R(f)$$

$$\arg \min_x -\{\ln(p(y_n | x)p(x))\} = \arg \min_x -\{\ln(p(y_n | x)) + \ln(p(x))\}$$

Gaussian noise:

Square regularization

Tikhonov  $J_0(y_{n,i} | x) = \|Ax - b\|^2$   $J_R(x) = (1/\beta) \|Px\|^2$

Ridge regression  $J_0(y_{n,i} | x) = \|Ax - b\|^2$   $J_R(x) = (1/\beta) \|x\|^2$


$l_2$  (total variation) regularization  $J_0(y_{n,i} | x) = \|Ax - b\|^2$   $J_R(x) = (1/\beta) \sqrt{\Delta x_1^2 + \Delta x_2^2 + \dots + \Delta x_N^2}$

Lasso regression  $J_0(y_{n,i} | x) = \|Ax - b\|^2$   $J_R(x) = (1/\beta) (|\Delta x_1| + |\Delta x_2| + \dots + |\Delta x_N|)$


A.A. 2022-2023

60/64

<http://borghese.di.unimi.it/>



## Role of $\lambda$



$$K(\sigma) \sum_i \|g_{n,i} - Af_i\|^2 \quad - \ln \left\{ \frac{1}{Z} e^{\left\{ -\frac{1}{\beta} U(\mathbf{f}) \right\}} \right\}$$


$$J(\mathbf{x}) = J_0(\mathbf{x}) + \lambda J_R(\mathbf{x})$$

$\lambda$  incorporates different elements here:


- the standard deviation of the noise in the likelihood
- the “temperature”, that is the decrease in the energy of the configurations with their cost ( $\beta$ )
- the normalized constant Z.

$\lambda$  has been investigated in the classical regularization theory (Engl et al., 1996), but not as deep in the Bayesian framework  $\rightarrow$   $\lambda$  is set experimentally through cross-validation.

A.A. 2022-2023
61/64
<http://borghese.di.unimi.it/>



## How to set the regularization parameter - Gaussian case



Analysis of the residual after the estimate  $\mathbf{n} = \mathbf{y} - \mathbf{Ax}$

- The residual should be distributed as the noise distribution

**Gaussian case:**  
Start with  $\lambda = 0 \rightarrow x$  minimizerà la likelihood  $J_0(\mathbf{x}) = 0$  ( $\mathbf{n} = 0$ ).

Is this a good solution? No!!

$$J(\mathbf{x}) = \|\mathbf{Ax} - \mathbf{b}\|^2 + \lambda \sqrt{\Delta x_1^2 + \Delta x_2^2 + \dots + \Delta x_N^2}$$

$$J(\mathbf{x}) = J_0(\mathbf{x}) + \lambda J_R(\mathbf{x})$$


We are reconstructing the data **and** the error. The latter is usually rapidly varying (e.g. grain images)

We get a better result if we throw away from  $\mathbf{x}$  the error. This happens when  $\mathbf{n} \neq 0$ . Increasing  $\lambda$ , we penalize rapid variations  $\rightarrow J_0(\mathbf{x})$  increases,  $\mathbf{n}$  increases  $\rightarrow$  it approaches the shape of the measurement error.


We stop when

- $(r_i, r_j) = \Sigma^2$  ( $\|r\|^2 = \sigma^2$ )
- Sample covariance is equal to distribution covariance
- Average value of the residual is zero,

mi.it



## How to set the regularization parameter - Poisson case



Analysis of the residual after the estimate  $\mathbf{n} = \mathbf{y} - \mathbf{Ax}$

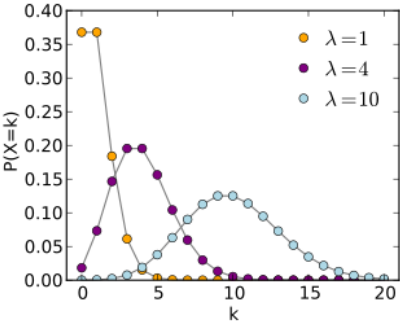
- The residual should be distributed as the noise distribution

**Poisson case:**

- $r_i$  tends to be larger, the larger is  $x_i$ .
- $\lambda$  is increased until  $\|r\|^2 / \mu \rightarrow 1$  (the mean is equal to variance)

1 parametro (media = varianza):


$$\mu = \sigma^2$$




A.A. 2022-2023

63/64

<http://borghese.di.unimi.it/>



## Overview




- Statistical filtering
- MAP estimate
- Different noise models
- Different regularizators

A.A. 2022-2023


64/64

<http://borghese.di.unimi.it/>






# A-priori on cliques




A.A. 2022-2023

65/64

<http://borghese.di.unimi.it/>

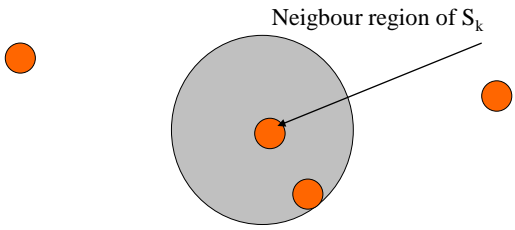


# A-priori



We can insert in the a-priori term all the desirable characteristic of the image: local smoothness, edges, piece-wise constancy,....

The idea of defining a **neighboring system** is a natural one:



Neighbour region of  $S_k$

Images have a natural neighbouring system: the pixels structure. We want to consider the local properties of the image considering neighboring pixels (in particular differential properties - our vision system is particularly tuning to gradients both spatial and temporal). Ideas have been borrowed from physics.

66 of 57



## Neighboring System



Let  $P$  be the set of pixels of the image:  $P = \{p_1, p_2, \dots, p_p\}$

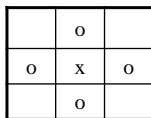
The neighboring system defined over  $P, S$ , is defined as  $H = \{\mathcal{N}_p | p, \forall p \in P\}$ , that has the following properties:

An element is not a neighbour of itself:  $p_k \notin \mathcal{N}_{p_k}$

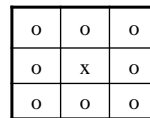
Mutuality of the neighboring relationship:  $p_k \in \mathcal{N}_{p_j} \iff p_j \in \mathcal{N}_{p_k}$

$(S, P)$  constitute a graph where  $P$  contains the nodes of the graph and  $S$  the the links.

Depending on the distance from  $p$ , different neighboring systems can be defined:



First order neighboring System  
4-neighboring System



Second order neighboring System  
8-neighboring System

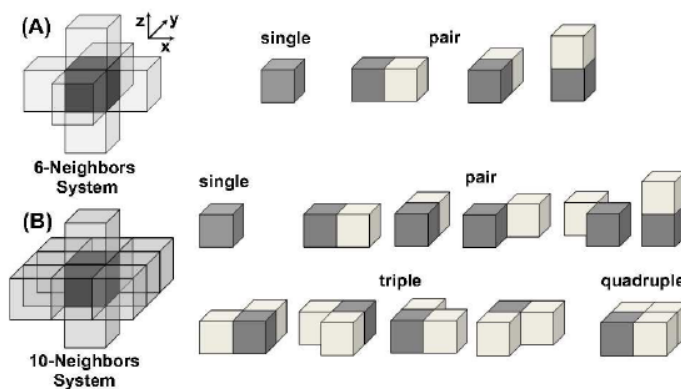
67 of 57



## Clique



Borrowed from physics.



A clique  $C$ , for  $(S, P)$ , is defined as a subset of sites in  $S$ .

It can be considered ordered sets of voxels, that are connected to  $p$  through  $S$ .

Types of cliques: single-site, pairs of neighboring sites, triples of neighboring sites, ... up to the cardinality of  $\mathcal{N}_p$

68 of 57



## Markov Random Field



Given  $(S, P)$  we can define a set of random values,  $\{f_k(p)\}$  for each element defined by  $S$ , that is in  $\mathcal{N}_p$ . Therefore we define a **random field**,  $\mathcal{F}$ , over  $S$ :

$$\mathcal{F}(\mathcal{N}_p) = \{f_k(m) \mid m \in \mathcal{N}_p\} \forall p$$

Under the Markovian hypotheses:

$$P(f(p)) \geq 0 \quad \forall p$$

Positivity

$$P(f(p) \mid g(P - \{p\})) = P(f(p) \mid g(\mathcal{N}_p))$$

Markovianity

2 expresses the fact that the probability of  $p$  assuming a certain value,  $f$  (e.g. a certain gradient), is the same considering all the pixel of  $P$  but  $p$  or only the neighbor pixels is the same, that is the value of  $f$  depends only on the gray value of the pixels in  $\mathcal{N}_p$ .

the random field  $\mathcal{F}$  is named **Markov Random Field**.

69 of 57



## Energy in a Markov Random Field



A “potential” function,  $\phi_c(f)$ , can be defined for a MRF. This is a scalar value that is a function of the random value associated to the pixels for all the possible elements of a clique:

$$\phi_c(f) = \sum_{j \in c} f(p_j)$$

If we consider all the possible cliques defined for each element  $p$ , we can define a potential energy function associated to the MRF:

$$U(f) = \sum_{c \in C} \phi_c(f)$$

The higher is the potential energy, the lower is the probability that the set of random values of the elements of the cliques is realized, that is the higher is the penalization for the associated configuration.

70 of 57



## Gibbs prior



If we consider all the possible cliques defined for each element  $p$ , we can define a potential energy function associated to the MRF:

$$U(\mathbf{f}) = \sum_{c \in C} \phi_c(\mathbf{f})$$

The higher is the potential energy, the lower is the probability that the set of random values of the elements of the cliques is realized, that is the higher is the penalization for the associated configuration.

This is well captured by the Gibbs distribution, that describes the probability of a certain configuration to occur. It is a function exponentially decreasing of  $U$ :

$$P(\mathbf{f}) = \frac{1}{Z} e^{\left\{ -\frac{1}{\beta} U(\mathbf{f}) \right\}}$$

$P(\mathbf{f})$  is a Gibbs random field, Hammersley-Clifford theorem (1971).  $\beta$  regulates the decrease in probability and it is associated with temperature in physics.  $Z$  is a normalization constant. NB to define Gibbs random fields,  $P(\mathbf{f}) > 0$ ,  $P(\mathbf{f}) \rightarrow 0$   $U(\mathbf{f}) \rightarrow \infty$ : there are not configurations with 0 probability.

71 of 57



## A-priori on cliques

