PHYS/ASTR 8150
Linear Inverse Problems
Applications to Imaging : Denoising and Deblurring

Fabien Baron

Georgia State University

baron@chara.gsu.edu

Inverse problems and regularization

- **Inverse problem**: recovering a wanted set of parameters from noise-corrupted data (e.g. pixel fluxes in image reconstruction)
- **Ill-posed** inverse problem: when the effective number of parameters to recover is greater than the effective number of data points, the solution may not exist or may not be unique
- **Regularization**: introduction of priors (known as regularizers) which will try to compensate for our lack of data. As usual, the prior/regularizer expresses the *a priori* probability of an object when we have no data.

Linear inverse problems

- $y \in \mathbb{R}^m$ is our one-dimensional column vector of m observations (our data)
- $x \in \mathbb{R}^n$ is the one-dimensional column vector, the object of interest to recover. For convenience, if x is multi-dimensional, such as a $k \times k$ pixels image, it is sought as a column vector with $n = k^2$ pixels arranged in lexicographic order.
- $\boldsymbol{A} \in \mathbb{R}^{m \times n}$ is the observation matrix that transforms the object of interest $\boldsymbol{x}$ into data points $\boldsymbol{y}$. We have:

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

where n is a noise vector. In this course we will assume Gaussian white noise.

Linear inverse problems: examples

- In image deblurring & denoising, $\boldsymbol{y}$ = noisy and blurred observed $k \times k$-pixel image, $\boldsymbol{x}$ = denoised & deblurred $k \times k$-pixel object. $\boldsymbol{x}$ and $\boldsymbol{y}$ don't necessarily have the same size;
- In light curve inversion, $\boldsymbol{y}$ is the flux received as a function of time while $\boldsymbol{x}$ is a tessel map of the surface temperature.

Direct matrix inversion

- A is rectangular and in general not inversible since $A \in \mathbb{R}^{m \times n}$
- $A^{\mathsf{T}}A \in \mathbb{R}^{n \times n}$, $AA^{\mathsf{T}} \in \mathbb{R}^{m \times m}$ are invertible
- Direct linear inversion can be done in a non-Bayesian way:

$$\begin{aligned} A\boldsymbol{x} &= \boldsymbol{y} \quad (-\boldsymbol{n}) \\ A^{\mathsf{T}}Ax &= A^{\mathsf{T}}y \\ \boldsymbol{x} &= \left(A^{\mathsf{T}}A\right)^{-1} A^{\mathsf{T}}\boldsymbol{y} = A^{+}\boldsymbol{y} \end{aligned}$$

- $A^{+} = \left(A^{\mathsf{T}}A\right)^{-1} A^{\mathsf{T}}$ is the Moore-Penrose pseudoinverse/generalized inverse of A.
- Application to denoising of unblurred images: since $A = I_n$, $\boldsymbol{x} = \boldsymbol{y}$. Not very helpful.

Maximum Likelihood for inverse problems (1): the setup

- We know the distribution of the noise $\boldsymbol{n} = \boldsymbol{y} - \boldsymbol{A}\boldsymbol{x}$ and covariances $\{\sigma_{ij}\}_{i,j=1\ldots m}$ of our measurements (uncertainties).
- If the data points are independent then the inverse covariance matrix is diagonal: $\boldsymbol{\Sigma} = \begin{bmatrix} \frac{1}{\sigma_{11}^2} & & \\ & \ddots & \\ & & \frac{1}{\sigma_{mm}^2} \end{bmatrix}$, with $\sigma_{ij} = 0, \quad \forall i \neq j$.
- For a given $\boldsymbol{x}$, the realization of the noise $\boldsymbol{n} = \boldsymbol{y} - \boldsymbol{A}\boldsymbol{x}$ depends on $\boldsymbol{y}$ and is normally distributed; we want to maximize the likelihood, which is:

$$\Pr(\boldsymbol{y}|\boldsymbol{x}) = \mathcal{L}(\boldsymbol{x}|\boldsymbol{y}) \propto \prod_{i=1}^{m} \prod_{j=1}^{m} \exp\left[-\frac{(y_i - (\boldsymbol{A}\boldsymbol{x})_i)(y_j - (\boldsymbol{A}\boldsymbol{x})_j)}{2\sigma_{ij}^2}\right]$$

- We want to minimize the negative log-likelihood, which is:

$$-\log \Pr(\boldsymbol{y}|\boldsymbol{x}) = \text{cst} + \frac{1}{2} \sum_{i=1}^{m} \sum_{j=1}^{m} \frac{(y_i - (Ax)_i)(y_j - (Ax)_j)}{\sigma_{ij}^2} = \text{cst} + \frac{1}{2}\chi^2(\boldsymbol{x})$$

Maximum Likelihood for inverse problems (2): χ^2

- Let's express the χ^2 in matricial form:

$$\begin{aligned}\chi^2 &= \sum_{i=1}^{m}\sum_{j=1}^{m}\frac{(y_i-(Ax)_i)(y_j-(Ax)_j)}{\sigma_{ij}^2}\\ &= (\boldsymbol{y}-\boldsymbol{Ax})^{\mathsf{T}}\boldsymbol{\Sigma}(\boldsymbol{y}-\boldsymbol{Ax}) = \|\boldsymbol{y}-\boldsymbol{Ax}\|_{2,\boldsymbol{\Sigma}}^2\end{aligned}$$

- Note: the squared ℓ_2 norm of vector $\boldsymbol{\alpha}$ is:

$$\|\boldsymbol{\alpha}\|_2^2 = \boldsymbol{\alpha}^{\mathsf{T}}\boldsymbol{\alpha} = [\alpha_1 \ldots \alpha_m]\begin{bmatrix}\alpha_1\\ \vdots\\ \alpha_m\end{bmatrix} = \sum_{i=1}^{m}\alpha_i^2$$

- The weighted ℓ_2 norm of $\boldsymbol{\alpha}$ is $\|\boldsymbol{\alpha}\|_{2,\boldsymbol{M}}^2 = \boldsymbol{\alpha}^{\mathsf{T}}\boldsymbol{M}\boldsymbol{\alpha}$, where $\boldsymbol{M}$ is the weight matrix. For $M = \begin{bmatrix}M_{11} & & \\ & \ddots & \\ & & M_{mm}\end{bmatrix}$, $\|\boldsymbol{\alpha}\|_{2,\boldsymbol{M}}^2 = \sum_{i=1}^{m} M_{ii}\alpha_i^2$

Maximum Likelihood for inverse problems (3): minimization

- The most likely solution is:

$$\widetilde{\boldsymbol{x}} = \underset{\boldsymbol{x}\in\mathbb{R}^n}{\operatorname{argmin}}\ \chi^2(\boldsymbol{x}) = \underset{\boldsymbol{x}\in\mathbb{R}^n}{\operatorname{argmin}}\ (\boldsymbol{y}-\boldsymbol{A}\boldsymbol{x})^{\mathsf{T}}\boldsymbol{\Sigma}(\boldsymbol{y}-\boldsymbol{A}\boldsymbol{x})$$

- To minimize χ^2, we set its gradient with respect to $\boldsymbol{x}$ to 0:

$$\frac{\partial\chi^2}{\partial\boldsymbol{x}} = \begin{bmatrix} \frac{\partial\chi^2}{\partial x_1} \\ \vdots \\ \frac{\partial\chi^2}{\partial x_n} \end{bmatrix} = \underbrace{\frac{\partial\left[(\boldsymbol{y}-\boldsymbol{A}\boldsymbol{x})^{\mathsf{T}}\boldsymbol{\Sigma}(\boldsymbol{y}-\boldsymbol{A}\boldsymbol{x})\right]}{\partial\boldsymbol{x}}}_{-2\boldsymbol{A}^{\mathsf{T}}\boldsymbol{\Sigma}(\boldsymbol{y}-\boldsymbol{A}\boldsymbol{x})} = 0$$

$$\boldsymbol{A}^{\mathsf{T}}\boldsymbol{\Sigma}\boldsymbol{A}\boldsymbol{x} = \boldsymbol{A}^{\mathsf{T}}\boldsymbol{\Sigma}\boldsymbol{y}$$

$$\widetilde{\boldsymbol{x}} = (\boldsymbol{A}^{\mathsf{T}}\boldsymbol{\Sigma}\boldsymbol{A})^{-1}\boldsymbol{A}^{\mathsf{T}}\boldsymbol{\Sigma}\boldsymbol{y}$$

- $\widetilde{\boldsymbol{x}}$ is the linear least squares estimator
- As with direct inversion, we may be **overfitting**, i.e. minimizing χ^2 too much.

Maximum a Posteriori: regularization and constraints

- We want to find the most probable image $\tilde{\boldsymbol{x}}$ given data $\boldsymbol{y}$, i.e. we want:

$$\tilde{\boldsymbol{x}} = \underset{\boldsymbol{x}\in\mathbb{R}^n}{\operatorname{argmax}} \Pr(\boldsymbol{x}|\boldsymbol{y}) \propto \underset{\boldsymbol{x}\in\mathbb{R}^n}{\operatorname{argmax}} \Pr(\boldsymbol{y}|\boldsymbol{x}) \Pr(\boldsymbol{x})$$

- We add prior under the form of a **regularizer**: $R(\boldsymbol{x}) = -\log \Pr(\boldsymbol{x})$

$$x = \underset{\boldsymbol{x}\in\mathbb{U}}{\operatorname{argmin}} \, \boldsymbol{(y - Ax)^T \Sigma (y - Ax)} + \lambda R(\boldsymbol{x})$$

- Additional regularization is provided by restricting $\boldsymbol{x}$ to $\mathbb{U}$. This allows to include **support constraints** ($\mathbb{U} = \text{mask}$), positivity ($\mathbb{U} = \mathbb{R}^n_+$), or bound constraints ($\mathbb{U} = [a, b]^n$).
- The **objective function** $J(\boldsymbol{x}) = \boldsymbol{(y - Ax)^T \Sigma (y - Ax)} + \lambda R(\boldsymbol{x})$ is composed of the χ^2 term plus the **regularization function** R.
- $\lambda \in \mathbb{R}$ is a scalar **hyperparameter** governing the regularization weight
- R and λ should be chosen to prevent overfitting of the data.

Maximum a Posteriori: generalized Tikhonov regularization

- The Tikhonov regularization is a prior with a simple ℓ_2 norm on a linear transform of the $\boldsymbol{x}$, i.e. $R(\boldsymbol{x}) \propto \|\boldsymbol{\Gamma x}\|_2^2$
- The Tikhonov regularized maximum likelihood problem:

$$\widetilde{\boldsymbol{x}} = \underset{\boldsymbol{x} \in \mathbb{R}^n}{\operatorname{argmin}} \, (\boldsymbol{y} - \boldsymbol{Ax})^{\mathsf{T}} \boldsymbol{\Sigma} (\boldsymbol{y} - \boldsymbol{Ax}) + \lambda \|\boldsymbol{\Gamma x}\|_2^2$$

- The choice $\boldsymbol{\Gamma} = \boldsymbol{I}_n$ is called ridge regression regularization, i.e. $R(\boldsymbol{x}) = \lambda \|\boldsymbol{x}\|_2^2$. Ridge regression strongly penalizes pixels with large flux values, and weakly pixels with smaller flux values.
- The choice $\boldsymbol{\Gamma} = \boldsymbol{\nabla}$, where $\boldsymbol{\nabla}$ is the spatial gradient operator, is called Total Squared Variation. It penalizes strongly images with quickly varying fluxes (large spatial gradient), and weakly images with patches of uniform flux.

Solving the unconstrained Tikhonov equation

- Solving for $\widetilde{\boldsymbol{x}}$, the most likely image/object of interest given the data:

$$\begin{aligned}\widetilde{\boldsymbol{x}} &= \operatorname*{argmin}_{\boldsymbol{x}\in\mathbb{R}^n} (\boldsymbol{y}-\boldsymbol{A}\boldsymbol{x})^{\mathsf{T}}\boldsymbol{\Sigma}(\boldsymbol{y}-\boldsymbol{A}\boldsymbol{x}) + \lambda\left\|\boldsymbol{\Gamma}\boldsymbol{x}\right\|_2^2\\ \implies & -2\boldsymbol{A}^{\mathsf{T}}\boldsymbol{\Sigma}(\boldsymbol{y}-\boldsymbol{A}\widetilde{\boldsymbol{x}}) + 2\lambda\boldsymbol{\Gamma}^{\mathsf{T}}\boldsymbol{\Gamma}\widetilde{\boldsymbol{x}} = 0\\ &(\boldsymbol{A}^{\mathsf{T}}\boldsymbol{\Sigma}\boldsymbol{A}+\lambda\boldsymbol{\Gamma}^{\mathsf{T}}\boldsymbol{\Gamma})\widetilde{\boldsymbol{x}} = \boldsymbol{A}^{\mathsf{T}}\boldsymbol{\Sigma}\boldsymbol{y}\end{aligned}$$

$$\boxed{\widetilde{\boldsymbol{x}} = (\boldsymbol{A}^{\mathsf{T}}\boldsymbol{\Sigma}\boldsymbol{A}+\lambda\boldsymbol{\Gamma}^{\mathsf{T}}\boldsymbol{\Gamma})^{-1}\boldsymbol{A}^{\mathsf{T}}\boldsymbol{\Sigma}\boldsymbol{y}}$$

- The choice $\boldsymbol{\Gamma} = \boldsymbol{I}_n \implies \boldsymbol{\Gamma}^{\mathsf{T}}\boldsymbol{\Gamma} = \boldsymbol{I}_{n\times n}$.
- The choice $\boldsymbol{\Gamma} = \boldsymbol{\nabla} \implies \boldsymbol{\Gamma}^{\mathsf{T}}\boldsymbol{\Gamma} = \boldsymbol{\nabla}^{\mathsf{T}}\boldsymbol{\nabla}$.

Choosing λ, under and overfitting, L-Curve

- Play with the Tikhonov code on our github repository

Separable regularizers and ℓ_p norm

- For a set of independent parameters with identical priors (e.g. independent pixel fluxes with Poisson priors), the global regularizer is a **separable function**:

$$\Pr(\boldsymbol{x}) = \Pr(x_1) \times \Pr(x_2) \ldots \times \Pr(x_N) \implies R(\boldsymbol{x}) = \sum_i R(x_i)$$

- There are many classic regularizers. Amongst them, the ℓ_p, $p \geq 1$ **norm** of vector $\boldsymbol{x}$ penalizes higher fluxes using a power law. The squared ℓ_2 norm is used in Tikhonov regularization:

$$\ell_p(x) = \left[\sum_{i=1}^{N} |x_i|^p\right]^{\frac{1}{p}}$$

- The **pseudo-norm** ℓ_0 is even more strongly regularizing, since it counts all non-zero elements of $\boldsymbol{x}$ equally:

$$\ell_0(\boldsymbol{x}) = \sum_{i=1}^{N} \mathbb{1}(x_i > 0) \quad \text{where } \mathbb{1}() \text{ is the indicator function}$$

Separable regularizers and ℓ_p norm: examples

- In practice for e.g. $\boldsymbol{x} = [0, 3, 0, 4]$
- ℓ_0 is the number of non-zero x_i, $\ell_0(\boldsymbol{X}) = 2$.
- $\ell_1 = \sum_i |x_i|$ is the sum of moduli, $\ell_1(\boldsymbol{x}) = 7$.
- $\ell_2 = (\sum_i |x_i|^2)^{\frac{1}{2}}$ is the square root of the sum of square moduli, $\ell_2(\theta) = \sqrt{3^2 + 4^2} = 5$.
- Different values of p penalize more or less higher vs lower fluxes.
- In MAP image reconstruction, when using ℓ_p regularizers in conjunction with χ^2, the higher p, the smoother and less noisy the resulting image will be. E.g. ℓ_2 give soft images, ℓ_1 sharp images, ℓ_0 very spiky images.
- $\ell_p(\theta - \gamma)$, where γ is a default expected level, expressing defaults values for parameter set θ, i.e. values that we would expect to be "normal" in absence of data. Example: the default expected flux level in a star field is zero in the absence of data, so the default image γ is an image with zero everywhere. But the default image for an image of the Sun is a mostly uniform disc, so γ would be this expected disc.

Sparsity & dictionaries

- An object (image, array, vector) is said **sparse** in a basis called a **dictionary** if it can be represented in this basis by a small number of non-zero coefficients
- The **impulsion/image dictionary** is the conventional grid of square pixels. A stellar field image, with only pointlike stars and zeros elsewhere is sparse in the impulsion basis.
- JPEG stores he coefficients of images expressed in the DCT (discrete cosine transform) that work well with most natural pictures. Since fewer coefficients can be used than in image dictionary, this results in compression of the image data.
- JPEG2000 stores the coefficients of images expressed in **wavelet dictionaries** (CDF 9/7 or 5/3)

Compressed sensing, analysis form

- **Compressed sensing theory** is a recent mathematical paradigm (from the 2000s) that asserts that the optimal regularization can be obtained by imposing sparsity in the basis where the solution is the most sparse. The MAP problem takes the **analysis form**:

$$\widetilde{\boldsymbol{x}} = \underset{\boldsymbol{x}\in\mathbb{R}^n}{\operatorname{argmin}} \{-\log \mathcal{L}(x) + \lambda\, \ell_0(\boldsymbol{\Gamma x})\}$$

- Compressed sensing literature recommends using ℓ_0 on the sparsity basis as regularizers, in order to enforce sparsity. Since minimizing ℓ_0 is a NP-hard problem, ℓ_1 is often used instead.
- **Total variation** is $\ell_1(\nabla \boldsymbol{x})$, i.e. the ℓ_1 nor of the **spatial gradient** of $\boldsymbol{x}$. The dictionary in this case is the vectorial space of spatial gradients. In image reconstruction this regularizer favors patches of uniform flux with sharp transitions.

Compressed sensing, synthesis form

- The **synthesis** approach is to look directly in the dictionary space, i.e. to look for a sparse vector $\boldsymbol{\theta} = \Gamma \boldsymbol{x}$ that represents the object of interest in the dictionary:

$$\widetilde{\boldsymbol{\theta}} = \underset{\boldsymbol{\theta} \in \mathbb{R}^k}{\operatorname{argmin}} - \log \mathcal{L}(\Gamma^{-1}\boldsymbol{\theta}) + \lambda \; \ell_0(\boldsymbol{\theta})$$

  then do $\boldsymbol{x} = \Gamma^{-1}\boldsymbol{\theta}$ to find the wanted solution.

- Doing image reconstruction by directly recovering CDF wavelets coefficients instead of pixels (impulsion dictionary).
- Analysis and synthesis are not equivalent. The dimensions of $\boldsymbol{x}$ and $\boldsymbol{\theta}$ may be very different. At this point in this course we cannot solve ℓ_0 or ℓ_1 regularization problems.