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:

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

- $A^+ = \left(A^\mathsf{T} A\right)^{-1} A^\mathsf{T}$ 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:

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

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

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

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

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

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

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

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

- 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:

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

- 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:

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

- 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:

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

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

- 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