Instructions pour T.P. déconvolution d'image

Éric Thiébaut

juin 2015

1 Objectifs du T.P.

1.1 Environnements de travail

Pour les besoins du T.P. vous avez besoin d'un langage de traitement des données (ici nous supposerons Matlab/Octave/Scilab).

Dans le fichier `tp-matlab.zip` vous avez :

- `data` – Des données à traiter.
- `matlab` – Des scripts pour Matlab/Octave ;

1.2 Lire et afficher les données

Pour la déconvolution d'image, vous avez besoin d'une image de l'objet (*e.g.*, `saturn.fits`) et d'une image d'une source de référence (*e.g.*, `saturn_psf.fits`).

Utilisez l'environnement de travail pour lire ces images et les afficher.

Par exemple avec `Matlab` :

```
% Lecture de l'image
y = fitsread('../data/saturn.fits');
h = fitsread('../data/saturn_psf.fits');

% Affichage
figure(1);
imagesc(y);

% Affichage ameliore
clf();                  % on efface la figure
imagesc(max(0, y));     % affichage avec coupure a zero

% Affichage de la PSF dans une autre fenetre
figure(2);              % nouvelle fenetre
imagesc(h);
```

1.3 Filtre de Wiener / Maximum a posteriori

1.3.1 Cas général

Dans le cas linéaire, le modèle direct des mesures est :

$$\boldsymbol{y} = \mathbf{H} \cdot \boldsymbol{x} + \boldsymbol{e}\,, \tag{1}$$

avec $\boldsymbol{y}$ les données, $\mathbf{H}$ la réponse instrumentale, $\boldsymbol{x}$ les paramètres d'intérêt et $\boldsymbol{e}$ le bruit.

Le filtre de Wiener donne la solution suivante au problème de la reconstruction :

$$\begin{aligned} \boldsymbol{x}^+ &= \bar{\boldsymbol{x}} + \mathbf{C}_{\boldsymbol{x}} \cdot \mathbf{H}^{\mathrm{t}} \cdot \left(\mathbf{H} \cdot \mathbf{C}_{\boldsymbol{x}} \cdot \mathbf{H}^{\mathrm{t}} + \mathbf{C}_{\boldsymbol{e}}\right)^{-1} \cdot (\boldsymbol{y} - \mathbf{H} \cdot \bar{\boldsymbol{x}}) && (2)\\ &= \bar{\boldsymbol{x}} + \left(\mathbf{C}_{\boldsymbol{x}}^{-1} + \mathbf{H}^{\mathrm{t}} \cdot \mathbf{C}_{\boldsymbol{e}}^{-1} \cdot \mathbf{H}\right)^{-1} \cdot \mathbf{H}^{\mathrm{t}} \cdot \mathbf{C}_{\boldsymbol{e}}^{-1} \cdot (\boldsymbol{y} - \mathbf{H} \cdot \bar{\boldsymbol{x}})\,, && (3) \end{aligned}$$

en supposant le bruit centré ($\mathbb{E}\{\boldsymbol{e}\} = \mathbf{0}$) et avec :

$$\begin{aligned} \bar{\boldsymbol{x}} &= \mathbb{E}\{\boldsymbol{x}\} && (4a)\\ \mathbf{C}_{\boldsymbol{x}} &= \mathrm{Cov}\{\boldsymbol{x}\} && (4b)\\ \mathbf{C}_{\boldsymbol{e}} &= \mathrm{Cov}\{\boldsymbol{e}\}\,. && (4c) \end{aligned}$$

Nota bene : le filtre de Wiener est aussi la solution au sens du maximum *a posteriori* (MAP) dans le cas linéaire et gaussien.

1.3.2 Simplifications

Dans le cas de la déconvolution, nous allons simplifier le problème (dans un premier temps) et considérer que :

$$\bar{\boldsymbol{x}} = \mathbf{0} \tag{5a}$$
$$\mathbf{C}_{\boldsymbol{x}} = \sigma_{\boldsymbol{x}}^2\,\mathbf{I} \tag{5b}$$
$$\mathbf{C}_{\boldsymbol{e}} = \sigma_{\boldsymbol{e}}^2\,\mathbf{I}\,. \tag{5c}$$

Dans ce cas, montrez que la solution du problème de la déconvolution devient :

$$\boldsymbol{x}^+ = \mathbf{H}^{\mathrm{t}} \cdot \left(\mathbf{H} \cdot \mathbf{H}^{\mathrm{t}} + \mu\,\mathbf{I}\right)^{-1} \cdot \boldsymbol{y} \tag{6}$$
$$= \left(\mu\,\mathbf{I} + \mathbf{H}^{\mathrm{t}} \cdot \mathbf{H}\right)^{-1} \cdot \mathbf{H}^{\mathrm{t}} \cdot \boldsymbol{y}\,, \tag{7}$$

avec $\mu = \sigma_{\boldsymbol{e}}^2/\sigma_{\boldsymbol{x}}^2$.

Pour modéliser la convolution, nous allons approximer la réponse instrumentale par :

$$\mathbf{H} = \mathbf{F}^{-1} \cdot \mathrm{diag}\left(\hat{\boldsymbol{h}}\right) \cdot \mathbf{F} \tag{8}$$

où $\mathbf{F}$ est l'opérateur de transformée de Fourier discrète (DFT) et $\hat{\boldsymbol{h}} = \mathbf{F} \cdot \boldsymbol{h}$ est la DFT de la réponse impulsionnelle $\boldsymbol{h}$.

Montrez que dans ce cas, la transformée de Fourier discrète de la solution $\hat{\boldsymbol{x}}^+ = \mathbf{F} \cdot \boldsymbol{x}^+$ s'écrit très simplement comme :

$$\hat{x}_u^+ = \frac{\hat{h}_u^\star\,\hat{y}_u}{\left|\hat{h}_u\right|^2 + \mu}\,. \tag{9}$$

1.4 Mise en œuvre du filtre de Wiener approché

1. Lire l'image de Saturne par le HST (`data/saturn.fits` ou `data/saturn.mda`) et la réponse impulsionnelle (`data/saturn_psf.fits` ou `data/saturn_psf.mda`).

2. Afficher ces deux images.

3. Dans un premier temps, nous allons utiliser une version simplifiée du filtre de Wiener :

$$\widetilde{x}_u^+ = \frac{\widetilde{h}_u^\star\,\widetilde{y}_u}{\left|\widetilde{h}_u\right|^2 + \mu} \tag{10}$$

   Expliquer ce que représentent les différentes variables de la formule ci-dessus. Quel est le rôle du terme μ ? Donner le pseudo-code correspondant à ce filtrage.

4. Discuter des modalités d'application sur l'exemple de l'image de Saturne. Coder une fonction effectuant ce filtrage et régler visuellement la valeur de μ de façon optimale.

1.5 Approche inverse par méthode itérative

Nous allons résoudre le problème de la déconvolution en minimisant la fonction de coût :

$$\phi(\boldsymbol{x}) = \phi_{\mathsf{data}}(\boldsymbol{x}) + \mu\,\phi_{\mathsf{prior}}(\boldsymbol{x}) \tag{11}$$

avec

$$\phi_{\mathsf{data}}(\boldsymbol{x}) = (\mathbf{H} \cdot \boldsymbol{x} - \boldsymbol{y})^{\mathrm{t}} \cdot \mathbf{W} \cdot (\mathbf{H} \cdot \boldsymbol{x} - \boldsymbol{y}) \tag{12}$$
$$= \sum_i w_i \left[(\boldsymbol{h} \star \boldsymbol{x})_i - y_i\right]^2 \tag{13}$$

où $\mathbf{H}$ est l'opérateur de convolution par $\boldsymbol{h}$ et

$$\phi_{\mathsf{prior}}(\boldsymbol{x}) = \|\mathbf{D} \cdot \boldsymbol{x}\|^2 \tag{14}$$
$$= \sum_{i,j} \left[x_{i+1,j} - x_{i,j}\right]^2 + \sum_{i,j} \left[x_{i,j+1} - x_{i,j}\right]^2 \tag{15}$$

avec $\mathbf{D}$ un opérateur linéaire (de type différences finies).

1. En utilisant les expressions algébriques, montrer que le minimum de la fonction de coût est la solution d'un système linéaire d'équations de la forme :

$$\mathbf{A} \cdot \boldsymbol{x} = \boldsymbol{b} \tag{16}$$

   Donner les expressions de $\mathbf{A}$ et de $\boldsymbol{b}$.

2. Écrire un code Matlab qui calcule $\mathbf{A} \cdot \boldsymbol{x}$ étant donné $\boldsymbol{x}$. Il faut utiliser la FFT pour la convolution. La fonction `deconv_dtd` (dans `deconv_dtd.m`) permet d'appliquer l'opérateur $\mathbf{D}^{\mathrm{t}} \cdot \mathbf{D}$.

3. La méthode des gradients conjugués linéaires est un algorithme simple et efficace (rapide et peu gourmand en mémoire) pour résoudre de façon itérative un système linéaire d'équations de la forme :

$$\mathbf{A} \cdot \boldsymbol{x} = \boldsymbol{b} \tag{17}$$

   où le « vecteur » $\boldsymbol{x}$ contient les inconnues et la matrice $\mathbf{A}$ est positive définie positive. L'algorithme des gradients conjugués linéaires est donné par la Figure 1. Les ingrédients de la méthode sont :

Gradients conjugués

initialisation:
  compute $\boldsymbol{r}_0 = \boldsymbol{b} - \mathbf{A} \cdot \boldsymbol{x}_0$ for some initial guess $\boldsymbol{x}_0$
  let $k = 0$
until convergence do
  $\rho_k = \boldsymbol{r}_k^{\mathrm{t}} \cdot \boldsymbol{r}_k$
  if $k = 0$, then
    $\boldsymbol{p}_k = \boldsymbol{r}_k$
  else
    $\beta_k = \rho_k / \rho_{k-1}$
    $\boldsymbol{p}_k = \boldsymbol{r}_k + \beta_k \boldsymbol{p}_{k-1}$
  endif
  $\boldsymbol{q}_k = \mathbf{A} \cdot \boldsymbol{p}_k$
  $\alpha_k = \rho_k / (\boldsymbol{p}_k^{\mathrm{t}} \cdot \boldsymbol{q}_k)$ (optimal step size)
  $\boldsymbol{x}_{k+1} = \boldsymbol{x}_k + \alpha_k \boldsymbol{p}_k$
  $\boldsymbol{r}_{k+1} = \boldsymbol{r}_k - \alpha_k \boldsymbol{q}_k$
  $k \leftarrow k + 1$
done

Figure 1: Algorithme des gradients conjugués linéaires.

- un opérateur réalisant l'opération $\boldsymbol{x} \mapsto \mathbf{A} \cdot \boldsymbol{x}$;
- une estimation initiale de $\boldsymbol{x}$ (une image remplie de zéros ou le résultat d'une recosntruction précédente feront l'affaire) ;
- le « vecteur » $\boldsymbol{b}$;

La fonction `conjgrad` en Matlab (dans `conjgrad.m`) permet d'appliquer les gradients conjugués.

4. En variant les poids $\mathbf{W} = \mathrm{diag}(\boldsymbol{w})$, vous simulerez des cas avec données manquantes.