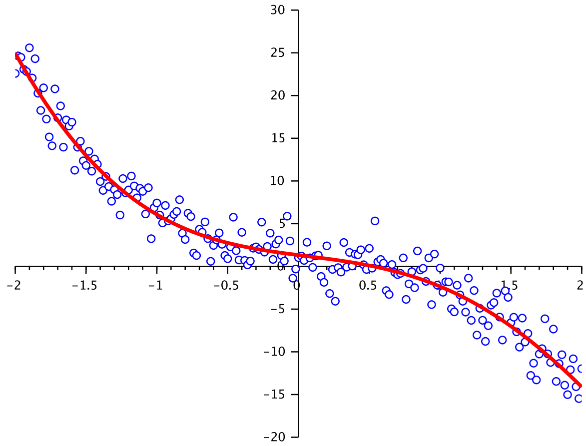
Exercice 1 - régression polynômiale avec validation



Le premier exercice de ce TP a pour but de vous initier à la méthode des moindres carrés ainsi qu'au principe de la validation.

On considère n couples $(t_i, y_i)_{i=1...n}$ (les vecteurs `t` et `y` sont dans le fichier `data1.sci` qui vous a été envoyé avec le sujet). Pour charger ce fichier dans Scilab et représenter les couples vous pouvez procéder ainsi :

```
--> load data1.dat;
--> plot(t,y,'o');
```

1 Régression

On cherche un polynôme de degré p défini par

$$P(t) = \theta_1 + \theta_2 t + \cdots + \theta_{p+1} t^p$$

tel que la quantité

$$S(\theta) = \sum_{i=1}^{n} (P(t_i) - y_i)^2,$$

est minimale, où on a noté $\theta = (\theta_1, \ldots, \theta_{p+1})^\top$.

1. Montrez que
$$S(\theta) = \| A\theta - y\|^2,$$
où vous préciserez la matrice A.
2. A quelle condition sur la matrice A le problème de moindres carrés admet-il une solution unique? A quelle condition sur les $(t_i)_{i=1...n}$ cette condition est-elle vérifiée (on pourra faire des calculs pour $p = 1$ et $p = 2$ puis conjecturer...).
3. Ecrire un programme Scilab permettant de résoudre le problème de moindres carrés. Le programme doit représenter à l'écran les points $(t_i, y_i)_{i=1...n}$ et le polynôme $P(t)$ obtenu pour des valeurs croissantes de son degré.

2 Validation

1. Tracer l'erreur de régression $S(\hat{\theta})$ en fonction du degré du polynôme. Que constatez-vous ? Cela permet-il de choisir le degré le plus adapté ?
2. Lisez la section "Model selection" du cours sur les moindres carrés et proposez une répartition des points $(t_i, y_i)_{i=1\dots n}$ en un ensemble d'apprentissage $\mathcal{T}$ et un ensemble de validation $\mathcal{V}$.
3. On définit l'erreur d'apprentissage par

$$S_{\mathcal{T}}(\theta) = \sum_{i\in\mathcal{T}} (P(t_i) - y_i)^2,$$

   et $\hat{\theta}^{(p)}$ les coefficients du polynôme de degré p rendant cette erreur minimale. Calculer et tracer $S_{\mathcal{T}}(\hat{\theta}^{(p)})$ en fonction de p (valeurs de p entre 1 et 8).
4. On définit l'erreur de validation par

$$S_{\mathcal{V}}(\theta) = \sum_{i\in\mathcal{V}} (P(t_i) - y_i)^2.$$

   Superposer au tracé précédent $S_{\mathcal{V}}(\hat{\theta}^{(p)})$ en fonction de p et conclure.

Exercice 2 - restauration d'une image

Il s'agit d'améliorer la qualité visuelle d'une image, représentée ici sur la figure 1. Cette image est du type de celles rencontrées en astronomie, dans le domaine de l'imagerie médicale, etc. Dans ces divers domaines, et pour des raisons variées, les images brutes sont souvent de mauvaise qualité : on dit qu'elles sont "bruitées". Avant de pouvoir les exploiter il faut leur faire subir un traitement permettant d'améliorer leur qualité visuelle. Les techniques utilisées sont variées. Nous vous proposons, dans le cadre de ce TP, d'utiliser l'une d'elles (dans une version simplifiée).

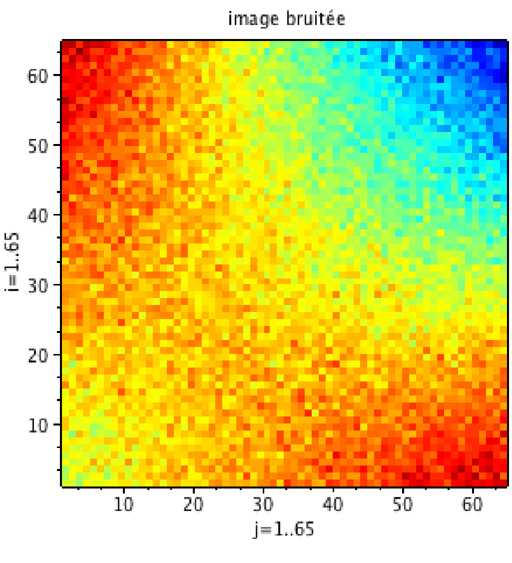


FIGURE 1 – Image avant restauration

Considérons que l'image est une matrice Z à n lignes et n colonnes, dont chaque élément z_{ij}, représentant un pixel de l'image, a une valeur donnée, proportionelle à l'intensité lumineuse. Pour débruiter cette image, on approcher chaque terme z_{ij} par une fonction polynômiale des deux variables i et j, dont le degré par rapport à l'une de ces deux variables est au plus 1, c'est à dire de la forme

$$f_c(i,j) = c_1 + c_2\, i + c_3\, j + c_4\, i\, j,$$

où $c_1,\dots,c_4$ sont les quatre coefficients qu'il va falloir déterminer.

Pour déterminer les coefficients $c_1,\dots,c_4$ tels que les valeurs de la fonction f_c aux points (i,j) approchent au mieux l'image, on va chercher à minimiser par rapport à c la fonction coût quadratique

$$J(c) = \sum_{i=1}^{n}\sum_{j=1}^{n}(z_{ij} - f_c(i,j))^2.$$

Une fois déterminés les coefficients $c_1,\dots,c_4$ minimisant la fonction coût $J(c)$, on peut calculer une nouvelle image dite image restaurée représentée par une matrice $\tilde{Z}$, dont les éléments sont donnés par

$$\tilde{z}_{ij} = f_c(i,j).$$

1 Etude du problème

On définit le vecteur

$$b = \left(\begin{array}{c} z_{1,1} \\ z_{2,1} \\ \vdots \\ z_{n,1} \\ \hline z_{1,2} \\ z_{2,2} \\ \vdots \\ z_{n,2} \\ \hline \vdots \\ \hline z_{1,n} \\ z_{2,n} \\ \vdots \\ z_{n,n} \end{array}\right).$$

Précisez alors la structure de la matrice A pour que l'on ait

$$J(c) = \sum_{i=1}^{n}\sum_{j=1}^{n}(z_{ij} - f_c(i,j))^2 = \|Ac - b\|^2,$$

et écrivez les instructions Scilab permettant de fabriquer A et b pour n quelconque.

2 Programation Scilab

L'image est dans le fichier `image.dat` que vous avez reçu par mail. Sauvez le fichier dans votre répertoire personnel, et une fois dans Scilab, tapez :

```
--> load image.dat
```

L'image se trouve dans la matrice Z et sa visualisation à l'écran se fait en utilisant la macro `grayplot`

```
--> set(gcf(),'color_map',jetcolormap(128));
--> grayplot(1:65,1:65,Z)
```

Réalisez un script qui réalise dans l'ordre les opérations suivantes :

- Chargement de l'image.
- Construction de la matrice A et du vecteur b.
- Résolution du problème de moindres carrés permettant d'obtenir le vecteur c.
- Calcul de l'image restaurée $\tilde{Z}$.
- Affichage dans une même fenêtre de l'image originale et de l'image restaurée.