Outils mathématiques - Optimisation
DEA IVR
Examen du 17 décembre 2003
Tous les documents sont autorisés

1 Calcul de la longueur de pas

Considérons la fonction de coût suivante qui est à minimiser :

$$f(x_1, x_2) = ax_1^3 + bx_1^2 x_2 + cx_1 x_2^2 + dx_2^3$$

Ici, nous voulons minimiser cette fonction en utilisant une méthode d'optimisation locale, comme vu lors du cours (cette fonction pourrait aussi être minimisée directement...). C'est-à-dire que nous supposons disposer d'une estimation initiale pour x_1 et x_2 : $x_{1,0}$ et $x_{2,0}$.

1.1 Question

Considérons une direction de recherche (d_1, d_2) (par exemple, calculée à partir du gradient de la fonction f). Décrivez comment déterminer la meilleure longueur de pas s, avec :

$$\min_s f(x_{1,0} + sd_1, x_{2,0} + sd_2)$$

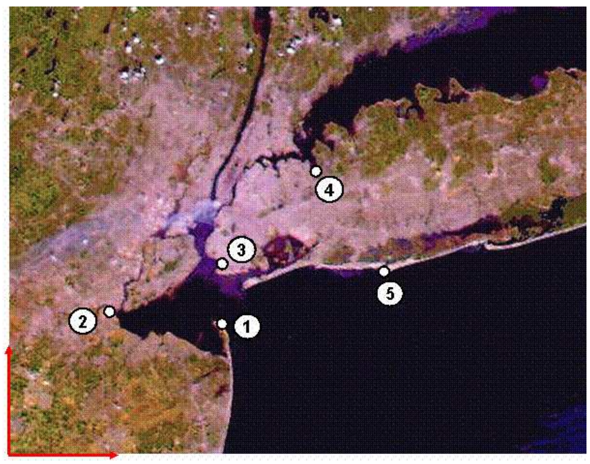
2 Alignement carte/photo

Considérons la figure ci-dessous : elle montre une carte et une photo aérienne. Une application potentielle consiste en l'alignement des deux, ou bien, en la détermination de la position de la photo par rapport à la carte. Pour ce faire, un opérateur a extrait quelques points homologues dans la carte et la photo. Soient n le nombre de points et $(x_{ci}, y_{ci}), i = 1..n$ les coordonnées des points sur la carte et $(x_{pi}, y_{pi}), i = 1..n$ celles des points homologues sur la photo. Ces coordonnées sont données par rapport à des repères propres à la carte et la photo, et l'alignement consiste en l'estimation du changement de repère approprié. Nous supposons ici que généralement, ce changement de repère comporte un changement d'échelle, une rotation ainsi qu'une translation, et qu'il est donné par :

$$\begin{pmatrix} x'_c \\ y'_c \end{pmatrix} = s \begin{pmatrix} \cos\Theta & -\sin\Theta \\ \sin\Theta & \cos\Theta \end{pmatrix} \begin{pmatrix} x_p \\ y_p \end{pmatrix} + \begin{pmatrix} t_x \\ t_y \end{pmatrix}$$

où s est un facteur d'échelle, Θ un angle de rotation (t_x, t_y) un vecteur de translation, (x_p, y_p) un point sur la photo et (x'_c, y'_c) le même point, après le changement de repère (donc, un point sur la carte).

Remarque. La dérivée de $\cos\Theta$ est $-\sin\Theta$ et la dérivée de $\sin\Theta$ est $\cos\Theta$.



2.1 Questions

Définissez une fonction de coût pour l'estimation des inconnues s, Θ, t_x et t_y, basée sur les n points homologues (x_{ci}, y_{ci}) et (x_{pi}, y_{pi}).

Qualifiez la nature de cette fonction (moindres carrés linéaires, moindres carrés non-linéaires, fonction non-linéaire générale, ...).

Avec quelle méthode (utilisez une méthode locale – une solution directe serait sinon aussi possible) résoudriez-vous ce problème d'optimisation, et pourquoi ?

Donnez les ingrédients nécessaires à la méthode choisie (gradients, matrice Jacobienne, etc., selon le cas).

3 Estimation d'un polygone

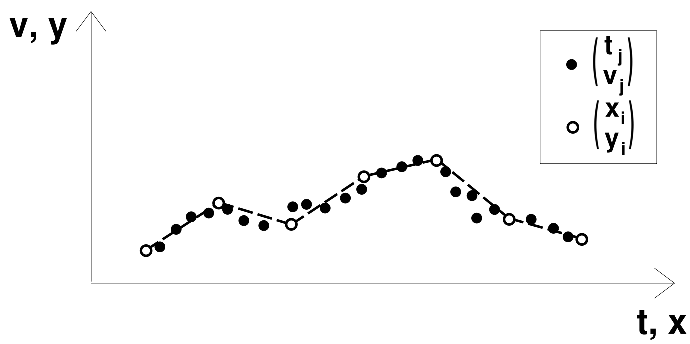
Considérons le problème de modélisation suivant. On dispose de mesures qui décrivent l'évolution d'un phénomène au cours du temps (par exemple, le cours d'une action à la Bourse). Le but de cet excercice est de trouver un modèle pour cette évolution.

Nous supposons que m mesures sont données, et qu'elles sont représentées par des instants de temps t_j et les valeurs v_j associées, pour $j = 1..m$ (voir la figure ci-dessous). Nous décidons ici de modéliser le phénomène sous-jacent par une courbe polygonale. Plus précisément, nous définissons n instants équidistants $x_i, i = 1..n$, avec n beaucoup plus petit que m, c'est-à-dire que chaque intervalle $[x_i, x_{i+1}]$ contient plusieurs mesures (plusieurs instants t_j). La courbe polygonale est définie par des valeurs y_i associées aux instants x_i : elle consiste alors en les $n-1$ segments de droite donnés par des couples de points successifs :

$$\begin{pmatrix} x_i \\ y_i \end{pmatrix} \text{ et } \begin{pmatrix} x_{i+1} \\ y_{i+1} \end{pmatrix} \quad i = 1..n-1$$

Remarque. Considérons un segment de droite donné par deux points (x_i, y_i) et (x_{i+1}, y_{i+1}). Un point quelconque (x, y) sur ce segment, satisfait la relation suivante entre x et y :

$$y = y_i + \frac{x - x_i}{x_{i+1} - x_i}(y_{i+1} - y_i)$$



3.1 Question (a)

Le but ici est d'estimer les meilleurs valeurs pour les y_i (les x_i sont supposés fixes ici), telles que la courbe polygonale approche au mieux les mesures originales (t_j, v_j).

Définissez une fonction de coût de type moindres carrés pour ce problème d'optimisation. Décrivez des pré-traitements qui pourraient être nécessaires pour établir la fonction de coût.

Donnez la matrice Jacobienne J de la fonction de coût. Décrivez avec une ou deux phrases la forme particulière de la matrice résultant du produit $\mathsf{J}^\mathsf{T}\mathsf{J}$ (du style : matrice diagonale, triangulaire, ...). Comment pourrait-on profiter de cette forme particulière ?

3.2 Question (b)

Pour la question (a), nous avons laissé fixes les abscisses x_i de la courbe polygonale. Afin que la courbe capte mieux des évolutions « spontanées » du phénomène à modéliser, nous nous décidons ici d'optimiser à la fois les y_i et les x_i.

Quelles en sont les conséquences sur par exemple la nature de la fonction de coût, la méthode d'optimisation à utiliser, etc. ?

A Solution Question 1 – Calcul de la longueur de pas

Il s'agit d'un problème d'optimisation non-linéaire (mais pas d'un problème de moindres carrés) qui peut donc être résolu avec n'importe quelle méthode générale (par exemple, Newton). A noter que le gradient et le Hessien sont des scalaires ici, donc si l'on veut, des matrices de dimensions 1×1.

Au lieu d'appliquer une méthode d'optimisation générale, on pourrait ici aussi exploiter le fait que la fonction de coût est un polynome cubique en la variable s. Nous cherchons le minimum de f en s, donc une racine de la dérivée de f, ou bien:

$$\begin{aligned} f'(s) = \quad & 3\left[ad_1^3 + bd_1^2 d_2 + cd_1 d_2^2 + dd_2^3\right] s^2 \\ + \quad & \left[6ad_1^2 x_{1,0} + 2bd_1(d_1 x_{2,0} + 2d_2 x_{1,0}) + 2cd_2(d_2 x_{1,0} + 2d_1 x_{2,0}) + 6dd_2^2 x_{2,0}\right] s \\ + \quad & \left[3ad_1 x_{1,0}^2 + bx_{1,0}(d_2 x_{1,0} + 2d_1 x_{2,0}) + cx_{2,0}(d_1 x_{2,0} + 2d_2 x_{1,0}) + 3dd_2 x_{2,0}^2\right] \\ = \quad & 0 \end{aligned}$$

Il suffit de résoudre cette équation quadratique et, des deux solutions réelles possibles, choisir celle qui correspond au minimum (valeur négative de la deuxième dérivée de f).

B Solution Question 2 – Alignement carte/photo

$$f_1(s,\Theta,t_x,t_y) = \frac{1}{2}\sum_{i=1}^{n}\left[(x_{ci} - sx_{pi}\cos\Theta + sy_{pi}\sin\Theta - t_x)^2 + (y_{ci} - sx_{pi}\sin\Theta - sy_{pi}\cos\Theta - t_y)^2\right]$$

C'est une fonction de type moindres carrés (il s'agit d'une somme de termes qui sont des carrés) *non-linéaires* (les termes mis au carré ne sont pas linéaires en les inconnues : $s\cos\Theta$ et $s\sin\Theta$).

Les $2n$ résidus du problème sont :

$$\begin{aligned} r_i &= x_{ci} - sx_{pi}\cos\Theta + sy_{pi}\sin\Theta - t_x \\ r_{i+n} &= y_{ci} - sx_{pi}\sin\Theta - sy_{pi}\cos\Theta - t_y \end{aligned}$$

La matrice Jacobienne et le gradient de la fonction de coût sont :

$$\mathsf{J}(s,\Theta,t_x,t_y) = \begin{pmatrix} y_{p1}\sin\Theta - x_{p1}\cos\Theta & s(x_{p1}\sin\Theta + y_{p1}\cos\Theta) & -1 & 0 \\ \vdots & \vdots & \vdots & \vdots \\ y_{pn}\sin\Theta - x_{pn}\cos\Theta & s(x_{pn}\sin\Theta + y_{pn}\cos\Theta) & -1 & 0 \\ -x_{p1}\sin\Theta - y_{p1}\cos\Theta & s(y_{p1}\sin\Theta - x_{p1}\cos\Theta) & 0 & -1 \\ \vdots & \vdots & \vdots & \vdots \\ -x_{pn}\sin\Theta - y_{pn}\cos\Theta & s(y_{pn}\sin\Theta - x_{pn}\cos\Theta) & 0 & -1 \end{pmatrix}$$

$$\mathbf{g}(s,\Theta,t_x,t_y) = \sum_{i=1}^{n}\begin{pmatrix} s(x_{pi}^2 + y_{pi}^2) + (x_{ci} - t_x)(y_{pi}\sin\Theta - x_{pi}\cos\Theta) - (y_{ci} - t_y)(x_{pi}\sin\Theta + y_{pi}\cos\Theta) \\ s\left[(x_{ci} - t_x)(x_{pi}\sin\Theta + y_{pi}\cos\Theta) + (y_{ci} - t_y)(y_{pi}\sin\Theta - x_{pi}\cos\Theta)\right] \\ s(x_{pi}\cos\Theta - y_{pi}\sin\Theta) - x_{ci} + t_x \\ s(x_{pi}\sin\Theta + y_{pi}\cos\Theta) - y_{ci} + t_y \end{pmatrix}$$

Le problème d'optimisation peut être résolu avec n'importe quelle méthode de moindres carrés non-linéaires (Gauß-Newton ou Levenberg-Marquardt par exemple). Pourtant, comme les dérivées secondes de la fonction de coût ne sont pas difficiles à calculer, on pourrait aussi utiliser la méthode de Newton.

En effet, une formulation de type moindres carrés *linéaires* est aussi possible : on peut constater que les inconnues s et Θ n'apparaissent que de manière conjointe ; si l'on substitue $s \cos \Theta$ par a et $s \sin \Theta$ par b, nous obtenons une fonction de coût de type moindres carrés linéaires :

$$f_2(a, b, t_x, t_y) = \frac{1}{2} \sum_{i=1}^{n} \left[(x_{ci} - x_{pi}a + y_{pi}b - t_x)^2 + (y_{ci} - x_{pi}b - y_{pi}a - t_y)^2 \right]$$

Cette fonction peut être minimisée aisément en utilisant la méthode des moindres carrés. La matrice Jacobienne J et le vecteur $\mathbf{b}$ (cf. §9 du poly) sont donnés par :

$$\mathsf{J} = \begin{pmatrix} -x_{p1} & y_{p1} & -1 & 0 \\ \vdots & \vdots & \vdots & \vdots \\ -x_{pn} & y_{pn} & -1 & 0 \\ -y_{p1} & -x_{p1} & 0 & -1 \\ \vdots & \vdots & \vdots & \vdots \\ -y_{pn} & -x_{pn} & 0 & -1 \end{pmatrix}$$

$$\mathbf{b} = \begin{pmatrix} x_{c1} \\ \vdots \\ x_{cn} \\ y_{c1} \\ \vdots \\ y_{cn} \end{pmatrix}$$

Le système des équations normales est :

$$\begin{pmatrix} \sum(x_{pi} + y_{pi}) & 0 & \sum x_{pi} & \sum y_{pi} \\ 0 & \sum(x_{pi} + y_{pi}) & -\sum y_{pi} & \sum x_{pi} \\ \sum x_{pi} & -\sum y_{pi} & n & 0 \\ \sum y_{pi} & \sum x_{pi} & 0 & n \end{pmatrix} \begin{pmatrix} a \\ b \\ t_x \\ t_y \end{pmatrix} = - \begin{pmatrix} \sum(x_{pi}x_{ci} + y_{pi}y_{ci}) \\ \sum(y_{pi}x_{ci} - x_{pi}y_{ci}) \\ -\sum x_{ci} \\ -\sum y_{ci} \end{pmatrix}$$

Une fois a et b estimés, on pourra déterminer s et Θ :

$$\begin{aligned} s &= \sqrt{a^2 + b^2} \\ \Theta &= \arccos\,(a/s) = \arcsin\,(b/s) \end{aligned}$$

C Solution Question 3 – Estimation d'un polygone

C.1 Question (a)

En pré-traitement, il faut bien sûr déterminer, pour toute valeur t_j, à quelle intervalle $[x_i, x_{i+1}]$ elle appartient. Dans la suite, nous supposons que x_1 est plus petite que la plus petite des t_j

et que x_n et plus grande que la plus grande des t_j. Soit $m_i, i = 1..n-1$ le nombre de t_j dans l'intervalle $[x_i, x_{i+1}]$ et $(t_{i,k}, v_{i,k}), k = 1..m_i$ les données associées à cette intervalle. Une fonction de coût pour notre problème est alors :

$$\min_{\mathbf{y}} \frac{1}{2} \sum_{i=1}^{n-1} \sum_{k=1}^{m_i} \left(v_{i,k} - y_i - \frac{t_{i,k} - x_i}{x_{i+1} - x_i} (y_{i+1} - y_i) \right)^2$$

Sa matrice Jacobienne est :

$$\mathsf{J} = \begin{pmatrix}
\frac{t_{1,1}-x_1}{x_2-x_1} - 1 & -\frac{t_{1,1}-x_1}{x_2-x_1} & 0 & \cdots & 0 & 0 \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
\frac{t_{1,m_1}-x_1}{x_2-x_1} - 1 & -\frac{t_{1,m_1}-x_1}{x_2-x_1} & 0 & \cdots & 0 & 0 \\
0 & \frac{t_{2,1}-x_2}{x_3-x_2} - 1 & -\frac{t_{2,1}-x_2}{x_3-x_2} & \cdots & 0 & 0 \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
0 & \frac{t_{2,m_2}-x_2}{x_3-x_2} - 1 & -\frac{t_{2,m_2}-x_2}{x_3-x_2} & \cdots & 0 & 0 \\
\vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\
0 & 0 & 0 & \cdots & \frac{t_{n-1,1}-x_{n-1}}{x_n-x_{n-1}} - 1 & -\frac{t_{n-1,1}-x_{n-1}}{x_n-x_{n-1}} \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
0 & 0 & 0 & \cdots & \frac{t_{n-1,m_{n-1}}-x_{n-1}}{x_n-x_{n-1}} - 1 & -\frac{t_{n-1,m_{n-1}}-x_{n-1}}{x_n-x_{n-1}}
\end{pmatrix}$$

ou bien, après une légère simplification :

$$\mathsf{J} = \begin{pmatrix}
\frac{t_{1,1}-x_2}{x_2-x_1} & -\frac{t_{1,1}-x_1}{x_2-x_1} & 0 & \cdots & 0 & 0 \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
\frac{t_{1,m_1}-x_2}{x_2-x_1} & -\frac{t_{1,m_1}-x_1}{x_2-x_1} & 0 & \cdots & 0 & 0 \\
0 & \frac{t_{2,1}-x_3}{x_3-x_2} & -\frac{t_{2,1}-x_2}{x_3-x_2} & \cdots & 0 & 0 \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
0 & \frac{t_{2,m_2}-x_3}{x_3-x_2} & -\frac{t_{2,m_2}-x_2}{x_3-x_2} & \cdots & 0 & 0 \\
\vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\
0 & 0 & 0 & \cdots & \frac{t_{n-1,1}-x_n}{x_n-x_{n-1}} & -\frac{t_{n-1,1}-x_{n-1}}{x_n-x_{n-1}} \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
0 & 0 & 0 & \cdots & \frac{t_{n-1,m_{n-1}}-x_n}{x_n-x_{n-1}} & -\frac{t_{n-1,m_{n-1}}-x_{n-1}}{x_n-x_{n-1}}
\end{pmatrix}$$

Le produit $\mathsf{J}^\mathsf{T}\mathsf{J}$ a la forme suivante :

$$\mathsf{J}^\mathsf{T}\mathsf{J} = \begin{pmatrix} a_1 & b_1 & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ b_1 & a_2 & b_2 & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & b_2 & a_3 & b_3 & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & b_3 & a_3 & b_4 & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \ddots & \ddots & \ddots & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & b_{n-3} & a_{n-3} & b_{n-2} & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & b_{n-2} & a_{n-2} & b_{n-1} \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & b_{n-1} & a_{n-1} \end{pmatrix}$$

où les $\cdot$ représentent des zéros et :

$$\begin{aligned} a_i &= \sum_{k=1}^{m_i} \left(\frac{t_{i,k} - x_{i+1}}{x_{i+1} - x_i} \right)^2 \\ b_i &= \sum_{k=1}^{m_i} \frac{(t_{i,k} - x_{i+1})(x_i - t_{i,k})}{(x_{i+1} - x_i)^2} \end{aligned}$$

La matrice $\mathsf{J}^\mathsf{T}\mathsf{J}$ a donc une structure que l'on qualifie de « bande » (*band-diagonal* en anglais). Grâce à cette structure l'inverse de $\mathsf{J}^\mathsf{T}\mathsf{J}$ peut être calculé efficacement et aussi, J et $\mathsf{J}^\mathsf{T}\mathsf{J}$ peuvent être stockées efficacement.

C.2 Question (b)

Le nouveau problème d'optimisation peut être formulé ainsi :

$$\min_{\delta_{ij}, x_i, y_i} \frac{1}{2} \sum_{i=1}^{n-1} \sum_{j=1}^{m} \delta_{ij} \left(v_j - y_i - \frac{t_j - x_i}{x_{i+1} - x_i} (y_{i+1} - y_i) \right)^2$$

avec $\delta_{ij} = 1$ si $t_j \in [x_i, x_{i+1}]$ et $\delta_{ij} = 0$ sinon.

Premièrement, il ne s'agit plus d'un problème de moindres carrés linéaires comme pour la question (a). Mais le problème majeur est que la fonction de coût contient une composante discrète (les valeurs 0 ou 1 des δ_{ij}) et n'est plus dérivable : en effet, un changement de l'une des valeurs x_i peut entraîner un changement dans les δ_{ij}, c'est-à-dire dans l'affectation des t_j aux intervalles $[x_i, x_{i+1}]$, ce qui crée des discontinuités dans le gradient de la fonction de coût.

Une méthode populaire pour résoudre ce type de problème procède ainsi :

- pour des δ_{ij} donnés, optimiser les x_i et les y_i (problème de moindres carrés non-linéaires avec la formulation ci-dessus) ;
- recalculer les δ_{ij} ;
- itérer ces deux étapes jusqu'à ce qu'il n'y ait plus de changement dans les δ_{ij}.