

TP4 : Régression linéaire

Dans ce TP, vous mettrez en place un algorithme de régression linéaire basée sur une descente de gradient le long d'un critère des moindres carrés.

1 Hein ??

Vous avez précédemment vu lors de l'interpolation que l'on pouvait, étant donné n points dans le plan, trouver le polynôme passant par ces n points : il s'agit du polynôme d'interpolation de Lagrange. Ce que l'on souhaite faire maintenant, c'est trouver la droite (polynôme de degré 1) qui approxime le mieux un ensemble de n points plus ou moins alignés.

La méthode est la suivante : On possède n points $(x_1, y_1), \dots, (x_n, y_n)$ plus ou moins alignés. On souhaite trouver la droite d'équation $y = ax + b$ qui minimise l'erreur au carré avec les points. L'erreur au carré de notre approximation par la droite $y = ax + b$ avec un point (x_i, y_i) s'écrit

$$(y_i - ax_i - b)^2 \tag{1}$$

On souhaite donc obtenir les paramètres a et b qui minimisent la quantité

$$Err(a, b) = \sum_{i=1}^n (y_i - ax_i - b)^2 \tag{2}$$

L'algorithme de descente de gradient se propose de faire ce travail : en partant de valeurs de a et de b quelconque (par exemple, 1), il cherche un minimum de la fonction $Err(a, b)$ en suivant la direction de plus forte pente descendante (imaginez $Err(a, b)$ comme une surface dans l'espace, l'algorithme de descente de gradient essaye d'y trouver un creux).

L'algorithme fonctionne ainsi :

```
1 while on n'est pas satisfait du résultat do
2   |   a = a -  $\frac{\alpha}{n}$  *  $\frac{\partial Err}{\partial a}$  ;
3   |   b = b -  $\frac{\alpha}{n}$  *  $\frac{\partial Err}{\partial b}$  ;
4 end
```

Le paramètre α est à fixer par l'utilisateur (vous)... Il vaut en général 10^{-3} .

1. Commencez par calculer la valeur de $\frac{\partial Err}{\partial a}$ et $\frac{\partial Err}{\partial b}$ (avec papier et crayon).
2. A votre avis, quelle sera la condition d'arrêt de votre descente de gradient ?
3. Récupérez des fichiers de points à l'adresse <http://goo.gl/4bW8kQ>. Etudiez un fichier de points avec un programme comme gnuplot, Matlab ou excel.
4. Codez l'algorithme de descente de gradient afin de trouver la meilleure droite approximant un ensemble de points donnés. Vérifiez vos résultats avec un traceur graphique (gnuplot, Matlab, excel).

2 Si les points sont plus ou moins alignés sur un polynôme

1. Si les points sont plus ou moins alignés sur un polynôme, comment modifier l'algorithme de descente de gradient pour qu'il fonctionne toujours ?
2. Modifiez votre programme afin qu'il fonctionne avec tout type de polynôme et testez-le.
3. Si votre programme ne fonctionne pas, modifiez la valeur de α (plus petite).
4. Votre programme est plus lent après avoir baissé α . Afin de conserver un programme rapide, renseignez-vous sur le principe de normalisation des données et modifiez votre programme en conséquent (rétablissez aussi α à sa valeur initiale).