Chapitre 3: Régression

Abdelkrim El MOUATASIM
Professeur Habilité en Mathématique Appliquée
https://sites.google.com/a/uiz.ac.ma/elmouatasim/

FPO - SMI - S6

2018-2019

Plan

1. Corrélation et régression simple
   - Corrélation entre deux variables aléatoires.
   - Régression linéaire avec un modèle linéaire
2. Régression linéaire multiple
3. Applications
   - Régularisation de l'image
   - Linguistique
   - Énergie renouvelable

Organisation du cours

1. Corrélation et régression simple
   - Corrélation entre deux variables aléatoires.
   - Régression linéaire avec un modèle linéaire
2. Régression linéaire multiple
3. Applications
   - Régularisation de l'image
   - Linguistique
   - Énergie renouvelable

Si deux variables X et Y sont indépendantes, alors

$$E(XY) = E(X)E(Y)$$

Dans le cas général, cette relation n'est pas vraie, on définit la covariance entre X et Y par

$$C(X,Y) = E(XY) - E(X)E(Y)$$

Cette valeur n'est pas "normalisée" (si X et Y ont des unités physiques par exemple ce nombre a une dimension), on définit donc la corrélation par

$$c(X,Y) = \frac{C(X,Y)}{\sigma(X)\sigma(Y)}$$

ce nombre est sans unité.

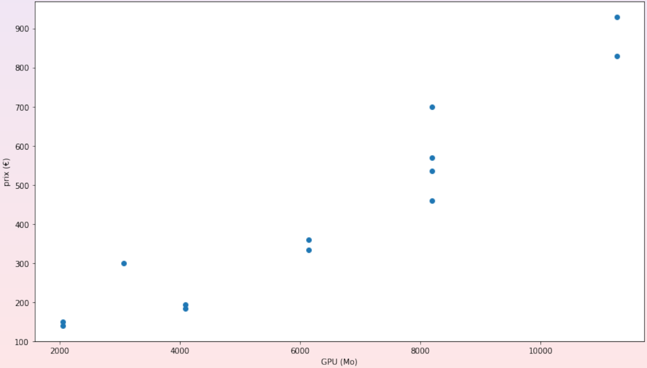
On montre qu'il appartient à l'intervalle $[-1, 1]$. Si $Y = X$ alors $c = 1$, si $Y = -X$ alors $c = -1$. Si les variables sont indépendantes, alors $c = 0$. Mais la réciproque est fausse. On utilise toutefois ce calcul pour estimer si deux variables sont corrélées ou indépendantes. Plus précisément, si deux variables paraissent corrélées, on peut essayer d'exprimer l'une en fonction de l'autre, c'est ce qu'on va faire dans la section suivante.

La régression linéaire

- Mais... À quoi ça sert en vrai ?
  - Prédiction d'une valeur en fonction de paramètres (prix de quelque chose)
  - Très utilisé en science (physiques des particules, sciences sociales) pour mettre en évidence des relations entre des variables ou ajuster un modèle
  - Dans le domaine médicales : les études épidémiologique
  - Dans la finance : prédictions des tendances, *Capital Asset Pricing Model*
  - . . .

Un exemple : le prix d'une carte graphique

- Ce prix va dépendre de la taille de la mémoire vive (GPU) de la carte (entre autre ...)
- On a un jeu de données, cad une liste de carte graphique dont on connait le couple $\{GPU; prix\}$:



Construire un modèle (regression linéaire)

- Appelons x_1 la taille de la mémoire vive de nos m carte graphiques, et y le prix correspondant.
- On cherche à trouver le modèle qui permet de prédire un prix $\hat{y}$ à partir x_1 :

$$\hat{y} = h_\theta(x_1)$$

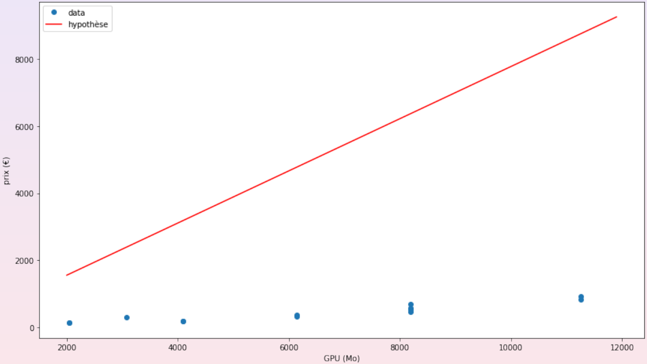
- On défini le paramètre θ_1 qui va *lier* x_1 à $\hat{y}$:

$$h_\theta(x) = \theta_1 x_1$$

- Rappel math : **fonction linéaire** $f(x) = kx$

Construire un modèle (regression linéaire)

- Initialisons aléatoirement la valeur de θ_1

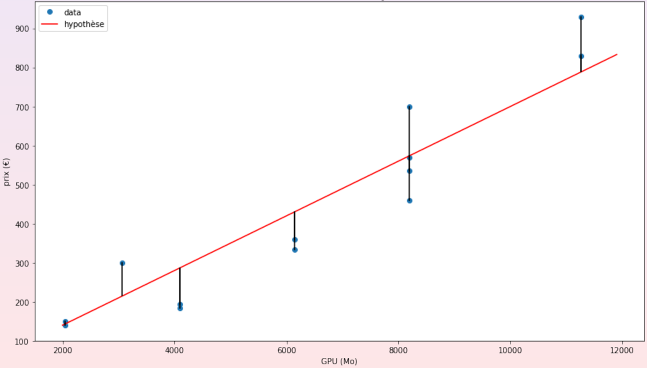


- C'est pas encore ça ...

La fonction de coût

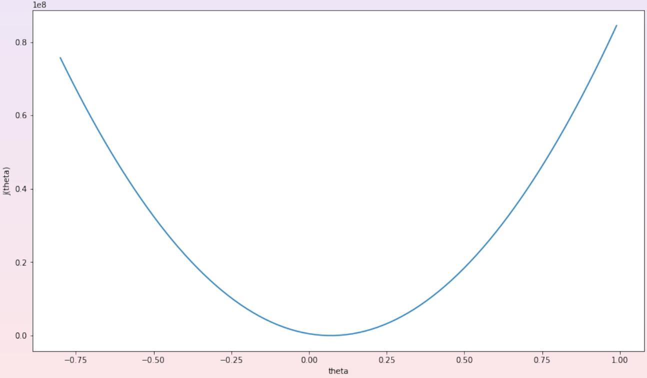
- Comment estimer la *véracité* de notre modèle ?
  - La **Fonction de coût** : $J(\theta)$
  - Une définition possible : somme quadratique des erreurs

$$J(\theta) = \frac{1}{2m} \sum_{i=0}^{m} (\hat{y}^{(i)} - y^{(i)})^2$$



La fonction de coût

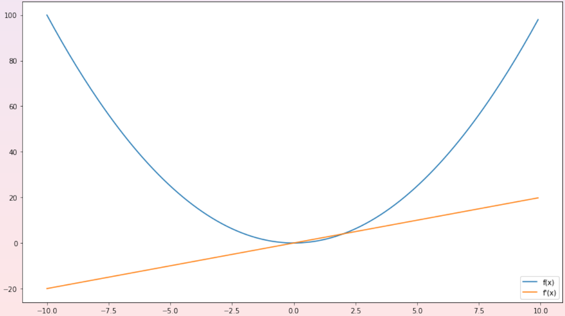
- On cherche à trouver la valeur de θ_1 qui **minimise** $J(\theta)$
- En Brute ...



- ... essayons d'optimiser

La descente de gradient

- C'est l'algorithme qui va nous permettre d'arriver "*rapidement*" au minimum de $J(\theta)$
- On va utiliser la *dérivation* : $\frac{d}{d\theta_1}J(\theta)$:
  - Si $J(\theta)$ est croissant : $\frac{d}{d\theta_1}J(\theta) > 0$
  - Si $J(\theta)$ est décroissant : $\frac{d}{d\theta_1}J(\theta) < 0$



La descente de gradient

- (Encore) un peu de math, la descente de gradient s'écrit :

Descente de gradient

Répéter jusqu'à convergence : $\{$

$$\theta_1 := \theta_1 - \alpha \frac{d}{d\theta_1} J(\theta)$$

$\}$

- α s'appelle le taux d'apprentissage (*learning rate*) et c'est le **seul** paramètre de l'algorithme.
- On va itérativement modifier la valeur de θ_1 en fonction de la dérivée de $J(\theta)$, jusqu'à minimiser $J(\theta)$ (*convergence*).

La descente de gradient

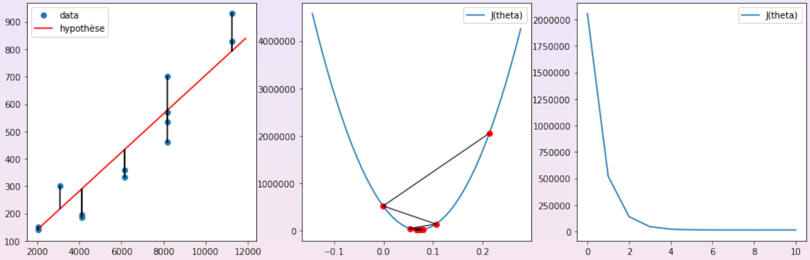
- Dérivons donc notre fonction de coût :

$$J(\theta) = \frac{1}{2m}\sum_{i=0}^{m}(\hat{y}^{(i)} - y^{(i)})^2 = \frac{1}{2m}\sum_{i=0}^{m}(\theta_1 x_1^{(i)} - y^{(i)})^2$$

$$\frac{d}{d\theta_1}J(\theta) = \frac{1}{m}\sum_{i=0}^{m}(\hat{y}^{(i)} - y^{(i)})x_1^{(i)}$$

- Un peu de *hand-tunning* :
  - Le learning rate (α) est fixé à 0.00000003
  - Défissons une précision $\epsilon = 0.001$ qui nous servira a arrêter la descente de gradient

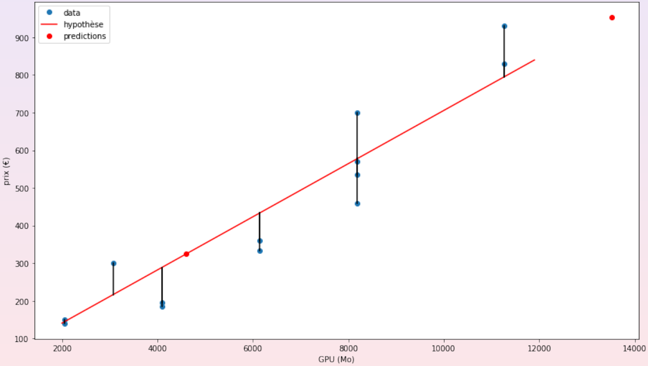
C'est parti !



- La descente de gradient c'est achevée au bout d'une dizaine d'itérations
- La valeur de notre paramètre θ_1 est 0.0706
- On peut voir que $J(\theta)$ a continuellement diminué à chaque itération

On peut maintenant faire une prédiction

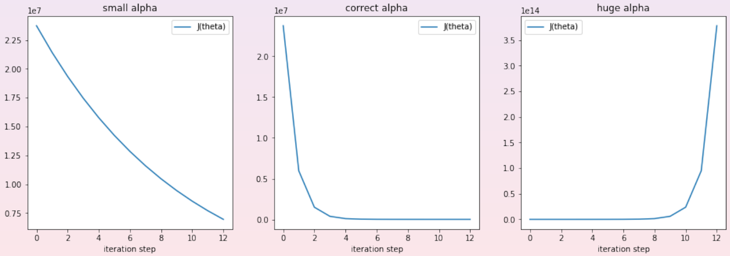
- Quel serait le prix d'une carte avec 4608 et 13516 Mo de GPU ? (ce qui n'a pas de sens, on est d'accord)



- On pourra les vendre autour de 325.10 et 953.62 euros !

Le choix du taux d'apprentissage

- **Learning Rate Très important :**
  - α **Trop grand :** la descente de gradient diverge
  - α **Trop petit :** la descente de gradient est très longue
- Pour choisir, on regarde l'évolution de la fonction de coût $J(\theta)$ en fonction du nombre d'itérations :



Régression linéaire avec un modèle linéaire

Soit $(X_i)_{1\leqslant i\leqslant n}$ et Y_i deux séries statistiques. On cherche à savoir s'il est raisonnable de prévoir Y_i en fonction de X_i, avec par exemple une relation linéaire $Y_i = a + bX_i$. Pour cela on minimise $f(a,b) = \sum(a + bX_i - Y_i)^2$ par rapport à a et b, ce qui donne les équations :

$$\left\{\begin{array}{lcl} \frac{\partial f}{\partial a} = 0 & = & 2\sum_{i=1}^{n}(a + bX_i - Y_i) \\ \frac{\partial f}{\partial b} = 0 & = & 2\sum_{i=1}^{n} X_i(a + bX_i - Y_i) \end{array}\right.$$

équivalentes à :

$$\left\{\begin{array}{llcl} a\sum_{i=1}^{n} + & b\sum_{i=1}^{n} X_i & = & \sum_{i=1}^{n} Y_i \\ a\sum_{i=1}^{n} X_i + & b\sum_{i=1}^{n} X_i^2 & = & \sum_{i=1}^{n} X_i Y_i \end{array}\right. \qquad (1)$$

Si on note U la matrice de première colonne remplie de 1, et de deuxième colonne remplie par X_i (en ligne i),

$$U = \begin{pmatrix} 1 & X_1 \\ 1 & X_2 \\ ... & ... \\ 1 & X_n \end{pmatrix}$$

et si on note U^* la transposée de U,

$$U^* = \begin{pmatrix} 1 & 1 & ... & 1 \\ X_1 & X_2 & ... & X_n \end{pmatrix}$$

on observe que

$$U^* U = \begin{pmatrix} \sum_{i=1}^n 1 & \sum_{i=1}^n X_i \\ \sum_{i=1}^n X_i & \sum_{i=1}^n X_i^2 \end{pmatrix}$$

le système (1) devient

$$U^* U \begin{pmatrix} a \\ b \end{pmatrix} = \begin{pmatrix} \sum_{i=1}^n Y_i \\ \sum_{i=1}^n X_i Y_i \end{pmatrix} = U^* Y, \quad \text{où } Y = \begin{pmatrix} Y_1 \\ Y_2 \\ ... \\ Y_n \end{pmatrix}$$

Donc la solution est donnée par

$$A = \begin{pmatrix} a \\ b \end{pmatrix} = \text{inv}(U^* U)(U^* Y)$$

On peut montrer que les coefficients de la droite sont donnés par :

$$b = \frac{E(XY) - E(X)E(Y)}{\sigma(X)^2}, \quad a = \overline{Y} - b\overline{X}$$

et

$$f(a, b) = n(1 - r^2)\sigma(Y)^2, \quad r = \frac{E(XY) - E(X)E(Y)}{\sigma(X)\sigma(Y)}$$

On montre que cette méthode se généralise (dans sa formulation matricielle), si on veut prévoir par exemple Y_i à partir de deux séries statistiques x_i et X_i avec une relation linéaire de la forme $Y_i = a + bx_i + cX_i$, alors (a, b, c) est obtenu en calculant $\mathrm{inv}(U^*U)U^*Y$, où cette fois la matrice U possède 3 colonnes, la colonne de 1, la colonne des x_i et la colonne des X_i. Les calculs pratiques se font de manière identique.

Autres régressions linéaires

On peut aussi supposer que Y dépend de manière non linéaire de X, par exemple logarithmique, exponentielle, polynomiale, etc. mais toujours de manière linéaire des paramètres à déterminer et calculer ces paramètres en utilisant la même méthode (minimiser l'erreur quadratique). Ceci peut servir par exemple à modéliser la consommation d'une denrée non renouvelable (par exemple les combustibles fossiles) par une fonction logistique ou par une gaussienne.

Organisation du cours

Un mot sur la regression linéaire multivariables

- Ici, nous avons vu le cas avec une seule variable x_1 (on aurait pu rajouter un biais θ_0, cad un terme constant : $\hat{y} = \theta_0 + \theta_1 x_1$)
- Le principe est le même, mais avec plusieurs variables x_i (donc plusieurs paramètres θ_i)
- Notre fonction hypothèse s'écrie alors :

$$h_\theta(x) = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \cdots + \theta_n x_n = \theta_0 + \sum_{i=1}^{n} \theta_i x_i$$

- **Astuce :** On définit $x_0 = 1$, et on re-écrit la fonction hypothèse :

$$h_\theta(x) = \theta_0 x_0 + \theta_1 x_1 \cdots + \theta_n x_n = \sum_{i=0}^{n} \theta_i x_i$$

- La fonction de coût reste inchangée

Un mot sur la regression linéaire multivariables

- Dans le cas multivariables, la descente de gradient devient :

Descente de gradient (cas multivariables)

Répéter jusqu'à convergence : $\{$

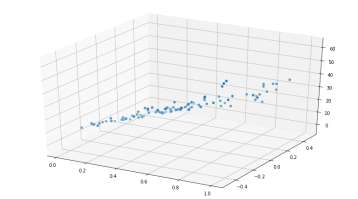
$$\theta_0 := \theta_0 - \alpha \frac{d}{d\theta_0} J(\theta)$$
$$\theta_1 := \theta_1 - \alpha \frac{d}{d\theta_1} J(\theta)$$
$$\theta_2 := \theta_2 - \alpha \frac{d}{d\theta_2} J(\theta)$$
$$\ldots$$

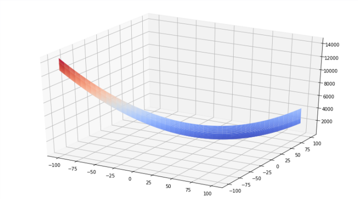
$\}$

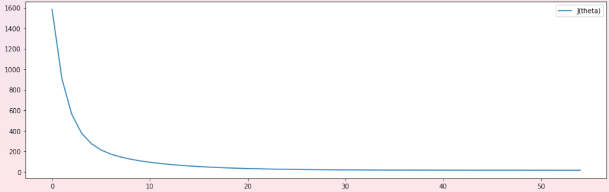
- Il est très important de simultanément changer les valeurs des paramètres.

Un mot sur la regression linéaire multivariables

- Pour illustrer : régression linéaire à deux dimensions







Généralités

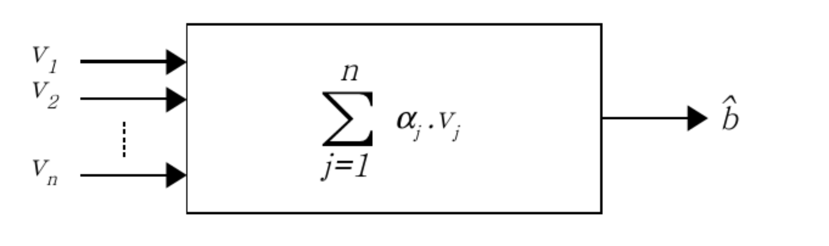
L'étude d'un phénomène peut, le plus souvent, être schématisé de la manière suivante : on s'intéresse à une grandeur b, que nous appellerons par la suite **réponse** ou **variable expliquée**, qui dépend d'un certain nombre de variables $v_1, v_2, \ldots, v_n$ que nous appellerons **facteurs** ou variables explicatives.

Notion de modèle

On cherche à mettre en évidence la liaison (relation fonctionnelle) pouvant exister entre la variable expliquée b et les variables explicatives $v_1, v_2, \ldots, v_n$. On s'intéresse aux modèles dits linéaires, i.e. aux modèles du type :

$$b = \alpha_1 v_1 + \alpha_2 v_2 + \cdots + \alpha_n v_n = \sum_{j=1}^{n} \alpha_j v_i j$$

les α_i sont des réels appelés **coefficients du modèle**.



Pourquoi $\hat{b}$ et pas b ?
On cherche à expliquer la variable b par n autres variables $v_1, v_2, \ldots, v_n$ mais on n'est pas certain que b ne dépend que de ces variables. Dans l'idéal $\hat{b} = b$ mais le plus souvent $\hat{b} \approx b$ avec $\hat{b} \neq b$.

Critère des moindres carrés- formulation

Critère

On cherche donc un modèle qui nous permet d'obtenir un $\hat{b}$ le plus « proche » possible de b. Pour cela, on effectue m mesures $(m > n)$ des variables $v_1, v_2, \ldots, v_n$ et de b. On cherche alors $\alpha_1, \alpha_2, \ldots, \alpha_n$ tel que, pour $i = 1 \ldots m$:

$$\hat{b}_i = \alpha_1 v_{i,1} + \alpha_2 v_{i,2} + \cdots + \alpha_n v_{i,n} = \sum_{j=1}^{n} \alpha_j v_{i,j}$$

soit le plus « proche » possible de b_i.

En utilisant les notations matricielles, le système :

$$\left\{ \begin{array}{lcl} \hat{b}_1 & = & \alpha_1 v_{1,1} + \alpha_2 v_{1,2} + \cdots + \alpha_n v_{1,n} \\ \hat{b}_2 & = & \alpha_1 v_{2,1} + \alpha_2 v_{2,2} + \cdots + \alpha_n v_{2,n} \\ \vdots & & \\ \hat{b}_m & = & \alpha_1 v_{m,1} + \alpha_2 v_{m,2} + \cdots + \alpha_n v_{m,n} \end{array} \right.$$

s'écrit :

$$\hat{b} = \underbrace{\begin{pmatrix} v_{1,1} & v_{1,2} & \cdots & v_{1,n} \\ v_{2,1} & v_{2,2} & \cdots & v_{2,n} \\ \vdots & \vdots & & \vdots \\ v_{m,1} & v_{m,2} & \cdots & v_{m,n} \end{pmatrix}}_{A} \times \underbrace{\begin{pmatrix} \alpha_1 \\ \alpha_2 \\ \vdots \\ \alpha_m \end{pmatrix}}_{x}$$

Ainsi, on cherche $x = (\alpha_1, \alpha_2, \ldots, \alpha_n)^T$ tel que Ax soit le plus « proche » possible de b.

On comprend alors que la notion de distance apparaît. On rappelle que la distance euclidienne usuelle est définie comme suite :

$$\forall x, y \in \mathbb{R}^m, d(x, y) = \sqrt{\|x - y\|^2}$$

où $\|.\|$ est la norme euclidienne :

$$\forall x \in \mathbb{R}^m, \|x\|^2 = x^T x = \sum_{j=1} m x_i^2$$

On souhaite que $d(\hat{b} = Ax, b)$ soit minimale, ce qui s'écrit :

$$\min_{x \in \mathbb{R}^n} \|Ax - b\|^2.$$

Organisation du cours

1. Corrélation et régression simple
   - Corrélation entre deux variables aléatoires.
   - Régression linéaire avec un modèle linéaire
2. Régression linéaire multiple
3. Applications
   - Régularisation de l'image
   - Linguistique
   - Énergie renouvelable

2.3 La régularisation

- Contraindre les paramètres θ_j sans réduire le nombre de variables
- On ré-écrit la fonction de coût avec le terme de régularisation :

$$J(\theta) = \frac{1}{2m}\left(\sum_{i=1}^{m}(h_\theta(x^{(i)}) - y^{(i)})^2 + \lambda \sum_{j=1}^{n} \theta_j^2\right)$$

- $\boldsymbol{\lambda}$: Paramètre de régularisation
- Si $\boldsymbol{\lambda}$ est trop grand : risque d'*underfitting*
- Si $\boldsymbol{\lambda} = \mathbf{0}$: pas de régularisation.
- **Remarque :** On ne régularise pas le terme constant $\boldsymbol{\theta_0}$

Problème de restauration d'image :

Soit $u \in \mathbb{R}^n$ l'image original à n pixels, et $v \in \mathbb{R}^n$ les observations effectuées, $v = Au + \eta$ où η est le bruit et A la matrice d'opérateur (par exemple dégradation par flou optique, sous-échantillonnage etc.)
L'image restaurée $\hat{u}$ est définie comme une solution de problème d'optimisation pour la fonction objectif f et les contraintes S.
(Dans d'autres applications on cherche seulement le minimum $=$ la valeur $f(\hat{u})$.)

Fonction-objectif régularisée :

$$f(u) = \Psi(u,v) + \beta\Phi(u) \quad (2)$$

$$\Phi(u) = \sum_i \varphi(\| D_i u \|) \quad (3)$$

$D_i u$ est l'approximation du gradient de l'image au pixel i ou la différences (pondérées) entre les pixels voisins.
$\beta > 0$ est un paramètre de régularisation, φ est une fonction croissante sur $\mathbb{R}_+$.
Ψ est le terme de fidélité aux données $v \in \mathbb{R}^m$, usuellement

$$\Psi(u,v) = \| Au - v \|^2$$

où A est une matrice $m \times n$.
Les contraintes possibles sont :

- $U = \{u \in \mathbb{R}^n : -u[i] \leqslant 0, \quad 1 \leqslant i \leqslant n\}$;
- $U = \{u \in \mathbb{R}^n : 0 \leqslant u[i] \leqslant 1, \quad 1 \leqslant i \leqslant n\}$.

Régularisation ℓ_2

Soit D est la matrice de différences premières et

$$f(u) = \| Au - v \|^2 + \beta \| Du \|^2 \tag{4}$$

- Moindres carrés : $\beta = 0$ dans (4).
- Minimisation sous contraintes :

$$f(u) = \sum_i \varphi(\| D_i u \| \quad \textbf{et} \quad \hat{u} \in U$$

Les contraintes possibles sont :

- $U = \{u \in \mathbb{R}^n : Au = v\}$ si rank$A < n$;
- $U = \{u \in \mathbb{R}^n : \| Au - v \| - \sigma \leqslant 0\}$.

Régularisation ℓ_1

La fonction de régularisation ℓ_1 dans le problème (2) est considéré comme suivante :

$$\Phi(u) = \| u \|_1 .$$

Control subgradient algorithm for image $$ \ell _1$$ ℓ 1 regularization

Abdelkrim El Mouatasim & Mohammed Wakrim

Signal, Image and Video Processing
ISSN 1863-1703
SIViP
DOI 10.1007/s11760-015-0815-z

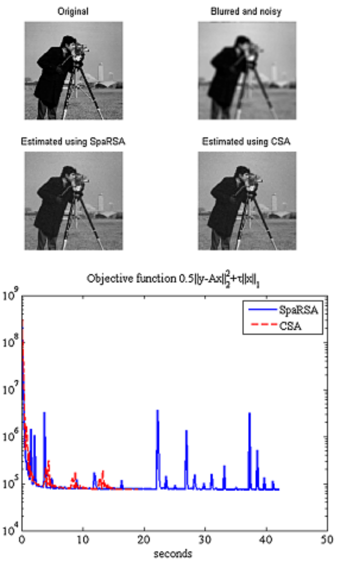


Fig. 1 Cameraman: origin, blurred and noise, SpaRSA estimated and CSA estimated images

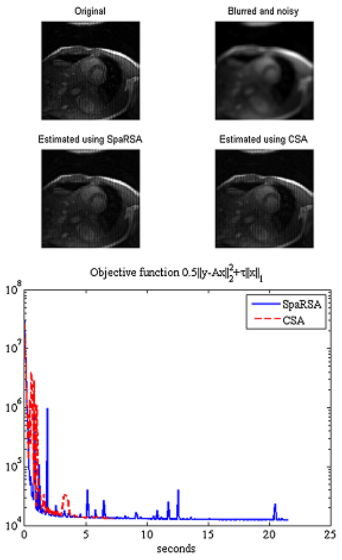


Fig. 2 Heart: origin, blurred and noise, SpaRSA estimated and CSA estimated images

Scientific Research Publishing

American Journal of Computational Mathematics, 2018, 8, 68-77
http://www.scirp.org/journal/ajcm
ISSN Online: 2161-1211
ISSN Print: 2161-1203

Simple and Multi Linear Regression Model of Verbs in Quran

Abdelkrim El Mouatasim

Faculty of Polydisciplinary Ouarzazate (FPO), Ibn Zohr University, Ouarzazate, Morocco
Email: a.elmouatasim@uiz.ac.ma

How to cite this paper: El Mouatasim, A. (2018) Simple and Multi Linear Regression Model of Verbs in Quran. *American Journal of Computational Mathematics*, **8**, 68-77.
https://doi.org/10.4236/ajcm.2018.81006

Received: February 17, 2018
Accepted: March 13, 2018
Published: March 16, 2018

Abstract

This paper mainly presented a good simple and multi-linear regression model of verbs in the Quran book. This model, gives an analysis for the influence to frequency of words with the form (—un, ـــون) made by the frequency of plural present verbs (t—un, تـــون) or (y—un, يـــون), and models, and the relationship between independent variables and dependent variable by fitting a linear equation to the observed data with simple linear regression model. The matlab function is used for finding the parameters of the linear regression model and plotting the fits. The results show that the parameters of the model are one vector (1, 1) and mean of dataset is (6, 7). Its corresponding to the verb with input is frequency of the verb they enter and the frequency of enter (yadkolun يدخلون ، دخلون dakilun), also other 17 points exist in the line and in the dataset of 387 verbs and their derivate verbs in Quran. The name of Allah (الله) showed when we use tree variables and plot it in 3D with option "Show Text" for a multi regression model.

Keywords

Linear Regression, Text Mining, Quran Statistics, Matlab, Arabic Grammar, Optimization, Computation Linguistics

Regression analysis of a photovoltaic (PV) system in FPO

A. El Mouatasim, and Y. Darmane

Citation: AIP Conference Proceedings **2056**, 020008 (2018); doi: 10.1063/1.5084981
View online: https://doi.org/10.1063/1.5084981
View Table of Contents: http://aip.scitation.org/toc/apc/2056/1
Published by the American Institute of Physics