Avancement Stage TDF

14 mars 2014

1 Mercredi 22/01

Première séance, il y a eu une prise de contact entre Emile, Baptiste et Hoël afin de jeter les bases du stage, présentations, présentation du projet, nous avons également légèrement survolé les procédés que nous allons mettre en oeuvre durant cette période.

Puis nous avons travaillé sur l'outil Matlab afin de déterminer certaines caractéristiques des donnees cf stage1.m, dans le but de voir ce qui était le plus pertinent, et de réfléchir si nous aurions à demander d'autres informations à Tdf, et le cas échéant lesquelles.

2 Vendredi 24/01

Prise de contact de Laure Quivy avec les étudiants, les premières estimation de dates ont été jetées, un code grossier devrait être fourni pour le mois de Mai (TDF a une réunion concernant les tarifs, ils aimeraient présenter ça à ce moment là), puis si on peut l'améliorer, continuer pour ne fournir une version plus performante en Juin ou Juillet.

Mise en place du Git hub en commun, vision ou révision de Matlab, méthodes d'interpolation, et de régression pour les deux étudiants.

3 Mercredi 29/01

3.1 Réunion à TDF

Aujourd'hui nous avons eu la réunion avec Olivier Marzouk de TDF, pour lui présenter les premières analyses de données.

Apparemment la donnée qui sensiblement pourrait nous manquer porte sur la date de construction des différents sites, il faut donc voir par la suite si elle explique les points étranges. Nous n'aurons pas accès au code du logiciel de l'ARCEP, mais une partie du calcul est disponible en ligne (à voir sur le lien suivant que nous a fourni Olivier : https ://docs.google.com/open ?id=0B4dfKcelvACpNFJIdDN2SEFXN00).

Le premier nombre de panneau serait à priori inutile, c'est plutôt le nombre maximum de panneaux (donc utiliser X(:,2) plutôt que X(:,1)).

3.2 Hors réunion

Il reste à vérifier si l'apparence de droites des graphes scatter(X(:,3),Y(:,1)) exhibées par Baptiste (cf décompo_3_droites_2015.m) reste valable en faisant varier les valeurs de X(:,2), X(:,4) et X(:,5). Si c'est le cas, il ne reste plus qu'à trouver ce qui fait varier le coefficient directeur de la droite, et on obtient une relation affine entre le prix et les critères... ce qui parait trop simple.

Les valeurs étranges ont été envoyées à Olivier afin qu'il vérifie qu'elles ne sont pas fausses (l'erreur est humaine).

Hoël a exhibé les configs qui apparaissent étranges sur les graphes (du moins celles qui sautent au yeux), c'est-à-dire ceux qui ont des caractéristiques proches d'autres sites mais qui présentent avec eux un écart de prix d'au moins $1/4$. Il y en a une dizaine.

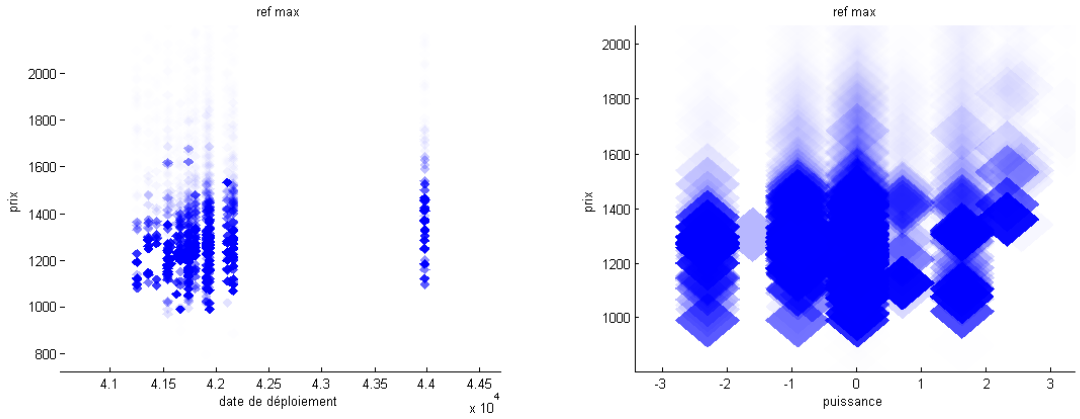
Baptiste a écrit un script permettant d'exhiber toutes les configs qui ont les mêmes caractéristiques mais pas le même prix (cf egal.m) , il y a 391 groupes comprenant chacun au moins 2 configs qui correspondent à cette description. Celà confirme qu'il nous faut d'autres critères.

4 Vendredi 14/02

Aujourd'hui Hoël a fait de la biblio sur "The Elements of statistical learning" sur le régression et la méthode des moindres carrés. Baptiste de son côté a étudié la régression par noyau (Kernel regression).

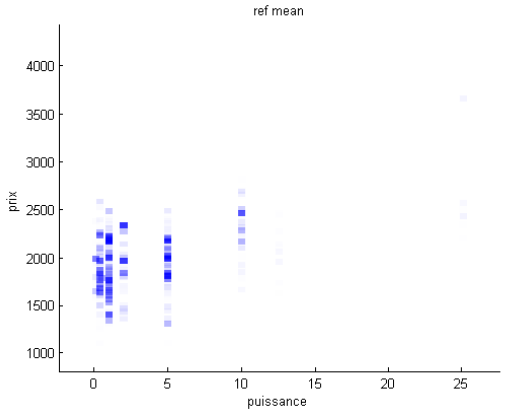
De plus des plots avec transparence (basée sur la distance à un point de référence après normation) ont été tracés en fonction de chacune des coordonnées, avec deux points de référence différents : la ref mean, où la référence est le point nul (sauf pour la coordonnées qui varie et pour celles qui paraissent moins utiles ie colonnes 1 5 et 6), et la ref max, où les coordonnées sont celles les plus représentées indépendamment.

Le prix ne semble pas influencé par les données puissance et date de déploiement (4 et 7). Le nombre de panneaux et la hauteur semblent par contre intéressants..



5 Mercredi 19/02

Aujourd'hui un script avec arguments optionnels a été écrit afin de faire des plots avec transparence bien plus facilement (cf plottrans.m). On a ainsi exhibé une certaine corrélation entre la puissance et le prix (simple croissance globale), et que le réseau RC ne comprenait pas de config dont le prix est très élevé.



D'autre part Hoël a fait de la biblio sur les splines. Il faut s'intéresser aux smoothing splines pour obtenir une résistance au bruit sur les données. Il y a alors une fonction à minimiser, comportant des moindres carrés et un poids sur l'intégrale de la dérivée seconde. Le problème étant qu'il n'y a pas d'algorithme fonctionnant sur tous les datas sets donc ce sera du cas par cas. Les splines sont déjà codées sous matlab (cf fit pchip etc...) donc il faudra regarder ce que ça donne. (pour une base cf smoothing splines sur google)

Le code excel de l'ARCEP présente énormément de données que TDF ne nous a pas fournies, dont une bonne partie en libre accès sur le document en ligne. Il faut donc regarder si ces données peuvent avoir une influence (rappelons nous qu'il nous manque un élément déterminant, et que celà étant donné on se rapprocherait énormément d'un calcul multilinéaire...).

Baptiste a écrit plusieurs scripts permettant de faire des regressions kernels sur les datasets. (cf kernel_ridge_regression_*.m)

6 Mercredi 26/02

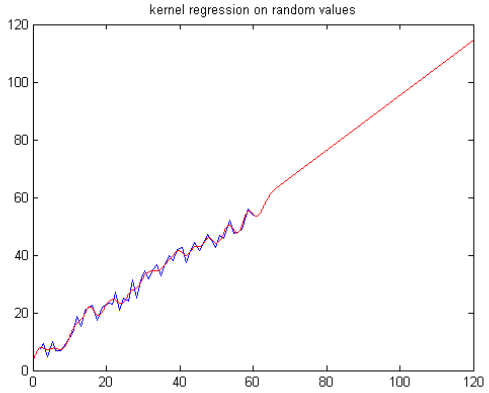
Aujourd'hui, une hypothèse sur le temps de calcul phénoménal du code de l'Arcep a été émise, après s'être rendu compte que le fichier mettait plusieurs minutes rien que pour être ouvert.. On suppose qu'il a beaucoup trop d'appels, et qu'il est beaucoup trop compliqué, et qu'on pourrait obtenir une précision suffisante en enlevant beaucoup de critères qu'il considère.

Dû à la complexité dudit fichier, il parait complexe de travailler dessus (ne serait ce que d'extraire des pages à isoler) car libre office ne supporte pas de fichiers aussi lourds, il faudrait donc essayer avec excel.

Le fichier permettant de plotter avec transparence a été amélioré, en refaisant les légendes, une option pour choisir sa figure, et la taille des éléments plottés a été refaite. Il parait donc plus ou moins complet pour l'instant.

Une page semblant intéressante du fichier Excel a été extraite (d'ailleurs il faut des licences Excel car on arrive aux limites de libre office), mais n'a pas été transférée sous matlab, parce qu'il y a plus de 200 critères à prendre en compte, donc c'est du travail lourd à faire à la main.

La priorité actuelle est de travailler sur de la regression Kernel, dans cette optique le script de Baptiste sur le premier script kernel linexp a été corrigé. (Kernel_ridge_regression_linexp_parameters.m)



Maintenant il faut écrire différents scripts permettant de lancer un certain nombre (à entrer en tant que paramètre) de kernels en prenant différentes parties des datasets pour train et test. Puis calculant l'erreur sur le data test, faisant la moyenne de ces erreurs sur des valeurs lambda, sigma, et mu données. Puis grâce à des boucles déterminer les valeurs de lambda, sigma, et mu optimales. Pour l'instant on fait des boucles avec un nombre restreint de valeurs pour les paramètres, pour les faire tourner sur nos ordinateurs, puis on fera de plus grosses boucles pour les faire tourner sur le serveur pour améliorer les résultats.

Le script séparant de manière aléatoire des valeurs de X et Y a été écrit. (cf split.m)

7 Vendredi 28/02

Baptiste est arrivé avec une version du script linexp_parameters qui prend en argument les input de la fonction kernel, et qui sort l'erreur de la regression. Ce script sera utile pour le calcul des coefficients optimaux.

Il a été déterminé que les boucles de ce script prennent beaucoup trop de temps de calcul (de l'ordre du mois ?). De plus il faut passer le script de regression pour prendre en compte plusieurs dimensions en entrée.

Etant donné qu'il faut faire une estimation par majoration il faut revoir la fonction d'erreur plutôt que les moindres carrés : les moindres carrés donnent le même poids au majoration et minoration, il faut éviter ça en mettant plus de poids aux erreurs par minoration. On obtient alors une approximation qui n'est pas du tout par majoration :

8 Mercredi 05/03

Baptiste a expliqué l'essentiel de la théorie kernel à Hoël.

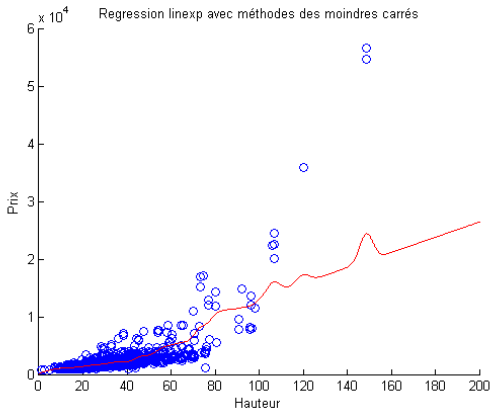


FIGURE 1 – On voit que la regression ne majore pas du tout notre data set

Baptiste a retapé tous les scripts afin de diminuer le temps de calcul et que ce soit plus lisible (commentaires, tout sous forme de fonctions, tout avec des arguments définis dans le script, etc...). Hoël a écrit un script pour déterminer (une fois les autres scripts finalisés) quelles seront les valeurs optimales de lambda sigma et mu. Ce script ne fonctionne peut être pas, car le temps de calcul est beaucoup trop long donc on ne peut pas le lancer pour vérifier (le linexp parameter tourne en 10 secondes, il est appelé 20*20*20*3=24000 fois par le script de calcul de critères optimaux...donc approximativement 3 jours de calcul), il faudra donc le lancer sur le serveur.

Il faut,pour l'erreur, une fonction croissante sur R+ et décroissante sur R-, c'est logique car il faut quand même minimiser les grandes distances. On cherche aussi à ce que les erreurs sur une petite plage de R+ aient un faible poids (penser à x^n avec n grand). Puis que la pente sur R- soit importante de manière à vouloir minimiser ces écarts là (penser à une fonction du type $exp(x^2) - 1$) puis mettre un poids intéressant sur de grandes distances de R+ (le x^n marcherait bien à cet endroit).

Sauf que pour tout ça il faut un test si x est positif ou négatif, ce qui empêche de faire l'inversion de matrice qui permet de faire la regression. Il faudrait donc plutôt faire une somme de deux fonctions, tout en gardant les critères précédents (minimum en zéro, faible poids au début de R+ etc...).

D'autre part le kernel cub_exp serait plus intéressant que le lin_exp car il colle mieux à l'évolution générale des prix par rapport à la hauteur.

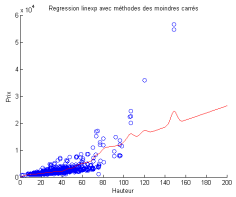


FIGURE 2 – Ici on a du lin_exp qui s'éloigne fortement de notre data set pour de fortes hauteurs

9 Vendredi 07/03

Aujourd'hui Hoël a essayé de réécrire le script de calcul optimal des paramètres de la regression kernel en enlevant les boucles. Pour celà il faut réécrire le script de la regression pour qu'elle prenne des vecteurs en argument pour les paramètres, mais aussi créer une matrice à partir de vecteurs contenant les combinaisons de valeurs possibles parmi ces vecteurs.

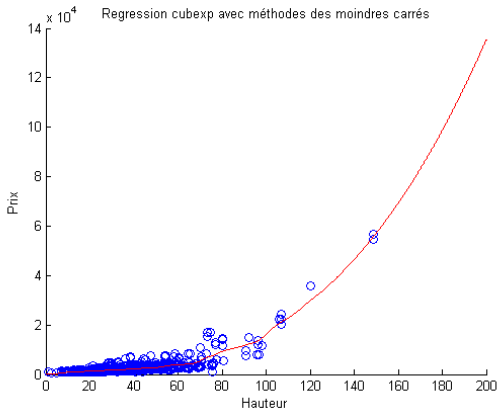


FIGURE 3 – Et là du cub_exp qui colle bien le prix pour de fortes hauteurs

10 Mercredi 12/03

Emile a suggéré qu'au lieu de créer une matrice comprenant toutes les combinaisons, on choisisse de manière aléatoire 1000 combinaisons possibles, puis qu'on lance la regression sur ces valeurs. On obtiendra alors un maillage uniforme de la distribution des paramètres et on diminuera le nombre de calculs (ici par 8, cf optimisation_parametre_rand.m). L'inconvénient est que si on trouve plusieurs minima locaux, le bruit fait qu'on ne prendra pas forcément le bon pour "zoomer" sur l'intervalle et relancer les calculs.

On peut même aller plus loin en faisant de l'optimisation du choix des valeurs, c'est à dire qu'au début le script prend des valeurs aléatoires puis qu'il déduise des valeurs qu'il possède où calculer les prochaines. Le problème est que l'optimisation sur un espace de dimension $2600 * 2600$ n'est pas envisageable.

Le script fonctionne avec un maillage uniforme, mais il est toujours trop long pour tourner sur un ordinateur, il prendrait environ 8 jours pour tourner avec les paramètres par défaut.

Ce script a été lancé sur le serveur, pour un temps de calcul de 2h.

11 Vendredi 14/03

On a reçu les résultats des calculs sur le serveur. Le lambda apparait avoir une valeur optimale légèrement supérieure à 1, mu ne doit pas avoir de valeur "moyenne", c'est-à-dire qu'il doit être soit petit soit grand (aux alentours de 1 ou 8) mais pas entre les deux. Et le sigma doit être très grand car il est du même ordre de grandeur que les données, on relance donc les calculs avec un zscore sur les données X afin de pouvoir calculer le sigma optimal. On ne trouve pas de minimum qui sorte du lot, il faut donc affiner le calcul en multidimensionnel.

Pour ça il faut réécrire le script de Kernel_cubexp pour qu'il prenne en compte des matrices et non plus des vecteurs. Pour l'instant les meilleurs valeurs trouvées sont :

$\lambda = 0.0005$

$\sigma = 6.4303$

$\mu = 8.1383$.