Predict Pollution in an Asian Megacity

FRAGNEAU Christopher, DU Qiming

Master 2, Spécialité Statistiques, *Université Pierre et Marie Curie*

5MS02. Modèles Dérivés. (Agathe Guilloux)

Introduction

Au sein d'une mégapole asiatique, on veut prédire la concentration des polluants NO_2, $PM_{2.5}$, O_3 ,PM_{10} pour les prochaines 24 heures sur un site particulier. Dans un premier temps, nous analysons la structure du jeu de données et proposons dans le respect de celle-ci une classification des variables. Dans la suite, nous présenterons notre étude avec NO_2, le traitement des autres variables de concentration étant analogue. Cette étude, se déroule avec une sélection préliminaire grâce à des boxplots, l'ANOVA et des tests de Student, suivie d'une expérimentation de trois méthodes que sont: la régression LASSO, la régression ElasticNet et la regression MLP pour lesquelles une comparaison en termes de limites et de scores MSE est donnée.

Préparation des données

Puisque le nombre de variables est plus grand que 3000, on aura besoin de bien construire la base de données pour simplifier la procédure d'échantillonage. Grâce au package *Pandas* , on construit des fonctions *DataSample* et *DataOutput* (cf. PLUME_fonction.py dans l'annexe) d'usage crucial pour simplifier la procédure d'extraction de variables selon des critères prédéfinis. Celles-ci servent également de passerelle entre *Python* et *R*. Pour l'ensemble de test, on le construit en respectant la structure de $Xtrain$ et $Xtest$. Pour chaque bloc de date continu, on le coupe en deux parties et on conserve la première pour construire l'ensemble d'apprentissage et la seconde pour l'ensemble de test.

I. Analyse Préliminaire

Après examen du jeu de données, nous disposons de données qualitatives, et d'autres quantitatives issues de données météo et de mesures de concentration de ces polluants dont l'ensemble est tout ou partie disséminé sur 17 sites, avec des mesures opérées sur 24 ou 48 heures. Ce fait, se retranscrit sur l'architecture des noms de nos variables et justifie l'emploi des fonctions *DataSample* et *DataOutput*. Plus précisément, nous regrouperons avec numéros de sites et heures confondues nos variables comme suit:

Type Date	**Météo**	**Type Vent**	**Concentration**
day	cloudcover	windspeed	NO_2
hour	dewpoint	windbearingcos	O_3
is Public Holiday	humidity	windbearingsin	$PM_{2.5}$
is Saturday	precipIntensity		PM_{10}
is Sunday or PublicHoliday	precipProbability		
	pressure		
	sunpower		
	temperature		

Table 1: Classification de Variables

Dans la suite, nous présenterons notre étude principalement avec NO_2, le traitement des autres variables de concentration étant analogue.

II. Sélection Préliminaires des Variables

Nous nous focalisons sur la variable NO_2 du site à prédire. Nous cherchons à sélectionner des variables de type date, grâce à des boxplot et l'ANOVA, et celles dites météo via un test de Student. L'allure de l'histogramme de NO_2 pour ce site, nous invite à opérer une transformation $\log$ sur les données de NO_2, afin de les rendre linéaires. A ce stade pour nos différentes catégories de variables, nous procédons à des tests d'ANOVA.
D'abord, on dresse les boxplots de $\log(NO_2)$ contre les variables de type date. Puis on applique un test d'ANOVA dont les p-values sont recensées dans le tableau ci-contre:

Variable	day	hour	is Public Holiday	is Saturday	is S or PH	month
p-value	$\leq 10^{-39}$	$\leq 10^{-108}$	$\leq 10^{-9}$	$\leq 10^{-18}$	0.007	$\leq 10^{-218}$

Table 2: p-values d'ANOVA pour les variables de type date

Comme les p-values sont très petites, nous conserverons toutes les variables de type date.
Puis, pour les variables de type météo, notre méconnaissance du domaine nous a conduit à des tests Fisher à l'issue desquels nous avons retenu XPrecipIntensity_04143 et XPrecipProbability_04143.

Ensuite, pour les variables de type vent, on note l'angle de windBearing θ, et l'angle de la direction de station que l'on veut prédire α. on a donc :

$$windSpeed \times \cos(\alpha - \theta) = windSpeed \times \cos(\theta)\cos(\alpha) + windSpeed \times \sin(\theta)\sin(\alpha)$$

Ici, par manque d'informations géographique relatives aux stations, soit sur α, nous conduit à conserver toutes les variables de **type vent**
Enfin, par la connaissance a priori en chimie, on conserve les mesures de NO_2 et O_3 tous sites

et heures confondus pour prédire celle du site voulu.

Finalement nous retenons les variables suivantes:

Type Date	Météo	Type Vent	Concentration
day	precipIntensity	windSpeed	NO_2
hour	precipProbability	windbearingcos	O_3
is Public Holiday		windbearingsin	$PM_{2.5}$
is Saturday			PM_{10}
is Sunday or PublicHoliday			

Table 3: Variables retenues pour les modèles

III Régression de Type Différente

Dans la suite, nous testons plusieurs régressions pour lesquelles, nous obtenons un estimateur $\mathrm{Y}_{i,prédit}$ de $\mathrm{Y}_{i,vrai}$ qui désigne notre vecteur des concentrations des polluants à prédire en terme de MSE.

Le MSE, risque quadratique est définit par :

$$\frac{1}{N}\sum_{i=1}^{N}(\mathrm{Y}_{i,prédit} - \mathrm{Y}_{i,vrai})^2$$

avec N le nombre d'observations.

III.1 Régression Ridge

La régression Ridge conserve toutes les variables mais, contraignant la norme des paramètres β_j , elle les empêche de prendre de trop grandes valeurs et limite ainsi la variance. La méthode Ridge correspond à la minimisation d'un critère des moindres carrés avec une pénalité de type l_2, on a $\|\beta\|_2 = \sum_{j=1}^{p} |\beta_j|^2$ dans un modèle $Y = X\beta + \epsilon$

L'estimateur $\hat{\beta}_{ridge}$ vérifie $\hat{\beta}_{ridge} = argmin_{\beta,\|\beta\|_2 \leq t}\|Y - X\beta\|^2$ pour un t convenablement choisi, et le paramètre λ est un paramètre de régularisation :

- Si $\lambda = 0$, on retrouve l'estimateur des moindres carrés.
- Si λ tend vers l'infini, on peut annuler aucun des $\hat{\beta}_j$ où $j = 1, \ldots, p$.

Par conséquent, la méthode de Ridge ne permet pas de sélectionner des variables. De plus, comme il est un cas particulier de la régression ElasticNet, on ne la teste pas pour le projet.

III.2 Régression LASSO

La méthode Lasso correspond à la minimisation d'un critère des moindres carrés avec une pénalité de type l_1, on a $\|\beta\|_1 = \sum_{j=1}^{p} |\beta_j|$ dans un modèle $Y = X\beta + \epsilon$

L'estimateur $\hat{\beta}_{lasso}$ vérifie $\hat{\beta}_{lasso} = argmin_{\beta, \|\beta\|_1 \leq t} \|Y - X\beta\|^2$ pour un t convenablement choisi, et le paramètre est un paramètre de régularisation :

- Si $\lambda = 0$, on retrouve l'estimateur des moindres carrés.
- Si λ tend vers l'infini, on annule tous les $\hat{\beta}_j$ où $j = 1, \ldots, p$.

La solution obtenue est dite parcimonieuse, elle comporte beaucoup de coefficients nuls. Ce qui permet de voir cette méthode du point de vue de sélection de variables.

La pénalisation λ est optimisée par validation croisée.

III.3 Régression ElasticNet

La méthode Elastic Net permet de combiner la régression ridge et la régression Lasso, en introduisant les deux types de pénalités simultanément. Le critère à minimiser est

$$\sum_{i=1}^{n} (Y_i - \beta_0 - \beta_1 X_i^{(1)} - \beta_2 X_i^{(2)} - \ldots - \beta_p X_i^{(p)})^2 + \lambda(\alpha \sum_{j=1}^{p} |\beta_j| + (1-\alpha) \sum_{j=1}^{p} \beta_j^2)$$

Pour $\alpha = 1$ nous retrouvons la méthode LASSO, tandis que pour $\alpha = 0$ nous retrouvons la régression RIDGE. Ici, on laisse deux paramètres à optimiser par validation croisée.

III.4 Régression avec Multi-Layer Perceptron

Du fait des limites des méthodes de regression linéaire, on expérimente une approximation non linéaire : MLP.

III.4.1 Présentation

La régression avec MLP est un sysyème de régression linéaire utilisant l'algorithme de rétropropagation pour l'appentissage. Voici une interprétation graphique :

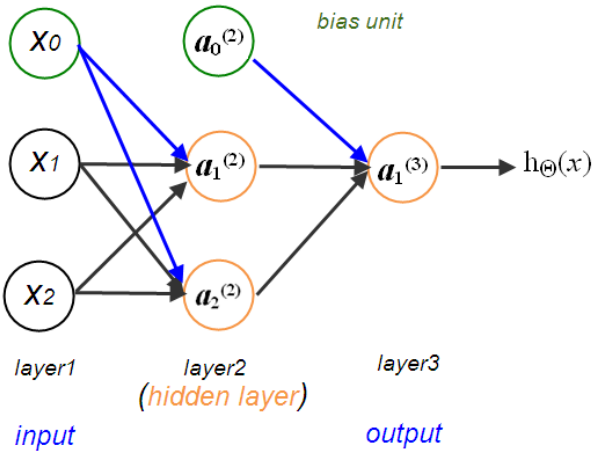

Figure 1: MLP à une couche

Pour le graphe, on utilise les définitions et notations suivantes :

$a_i^{(j)}$: "activation" de noeud i dans la couche j $\Theta^{(j)}$: matrice des poids de la fonction liant les layer j aux layer $j+1$. A titre d'exemple, on peut résumer les relations du schéma ci-contre comme suit:

$$a_0^{(2)} = g(\Theta_{00}^{(1)}x_0 + \Theta_{01}^{(1)}x_1 + \Theta_{02}^{(1)}x_2) = g(\Theta_0^T x) = g(z_0^{(2)})$$

$$a_1^{(2)} = g(\Theta_{10}^{(1)}x_0 + \Theta_{11}^{(1)}x_1 + \Theta_{12}^{(1)}x_2) = g(\Theta_1^T x) = g(z_1^{(2)})$$

$$a_2^{(2)} = g(\Theta_{20}^{(1)}x_0 + \Theta_{21}^{(1)}x_1 + \Theta_{22}^{(1)}x_2) = g(\Theta_2^T x) = g(z_2^{(2)})$$

$$h_\Theta(x) = a_1^{(3)} = g(\Theta_{10}^{(2)}a_0^{(2)} + \Theta_{11}^{(2)}a_1^{(2)} + \Theta_{12}^{(2)}a_2^{(2)})$$

Puisque l'on s'intéresse à l'appliquer pour la régression, on choisit une fonction linéaire comme fonction d'activiation

Dans notre étude, pour NO_2 à prédire selon une heure fixée, nous implémentons un MLP avec une seule couche cachée. Pour simplifier le procédure du choix de nombre de noeuds, on le teste brutalement à la main, puis on voit les performances pour les structures différentes. Dans la partie suivante, nous exposerons quelques résultats.

III.4.2 Algorithme de rétropropagation du gradient

Afin de minimiser les erreurs, la méthode MLP est toujours couplée avec un algorithme d'apprentissage dit de rétropropagation divisé en deux phases:

- **Phase 1:** Traverse directe et Calcul de la prédiction

  Propagation directe de l'entrée d'un modèle de formation à travers le réseau de neurones afin de générer des activations de sortie de la propagation. Propagation arrière des activations de sortie de la propagation à travers le réseau de neurones en utilisant le modèle cible de formation afin de générer les deltas (les gradients du poids) de toutes les sorties et les neurones cachés.

- **Phase 2:** Traverse inverse et Mise à jour de poids

  Multiplier son delta de sortie et d'entrée d'activation pour obtenir le gradient du poids. Soustraire un rapport (en pourcentage) de la pente à partir du poids, nommé le taux d'apprentissage qui influe sur la rapidité et qualité d'apprentissage. Plus le rapport est élevé, plus vite les apprentissages de neurones se font; plus le ratio est élevé, plus précise la formation est.

Cependant, la structure adaptée aux problème différents varie et il n'y a pas une façon efficace de déterminer le nombre de couches et de noeuds dans chaque couche. De plus,

l'algorithme de rétro-propagation partage les idées de l'algorithme du gradient, qui ne donne qu'une solution locale. C'est-à-dire, même si il marche très bien pour une certaine situation, la robustesse n'est pas du tout garantie.

Afin d'optimiser la méthode, nous devons faire un compromis entre la capacité de nos machines, le temps de calcul et et la précision de la formation.

IV. Comparaisons et Analyses

Nous menons des régressions de types différents sur l'ensemble de test qu'on a construit, et les scores se trouvent dans le tableau suivant:

Régression	LASSO	ELASTIC NET	MLP
MSE total	21137	21178	20163
MSE (NO_2)	5771.84	6196.31	6101.32
MSE (O_3)	10038.84	10256.83	9284.61
MSE ($PM_{2.5}$)	2892.50	2876.57	2641.83
MSE (PM_{10})	2434.42	2448.60	2135.58

Table 4: MSE des différentes régressions

En premier lieu, ce tableau nous fournit différentes combinaisons de modèles hybrides pour prédire nos 4 polluants. Nous entendons par modèle hybride un modèle utilisant un LASSO pour NO_2, un MLP pour O_3, un ELASTIC NET pour $PM_{2.5}$ et un LASSO pour PM_{10},par exemple.

Sous la contrainte d'un MSE minimal, nous retenons d'abord la régression MLP au détriment des autres. Compte tenu du temps de calcul, de la robustesse et de la difficulté pour le choix de paramètres, on va choisir LASSO comme modèle plus adapté. On remarque que si on met toutes les variables sans sélection dans LASSO, les variables sélectionnées coïncident beaucoup avec les variables sélectionnées par la sélection préliminaire. Pour diminuer le temps de calculs, on conserve le choix établi au sein de la partie II.

On trouve aussi un phénomène intéressant. Si on ne met pas les variables de type date ou si on les laisse sous la forme numérique, le MLP aura un meilleur score mais qui n'est pas stable (moins de 16000 quelque fois). En même temps, les qualités de LASSO et ElasticNet sont tellement mauvais(plus que 25000). Donc, on peut conclure que les variables de type date est un factor très important pour la méthode linéaire. Et pour MLP avec une structure adaptée, il semble qu'il dispose d'une capacité d'apprentissage automatique des effets du temps. Néanmoins, la structure de $Xtest$ ne comporte pas les sauts de date observés sur $Xtrain$, ce qui limite l'usage de MLP.

De façon générale, après avoir les valeurs prédites par les 96 régressions, on fait une hypothèse raisonnable pour obtenir les résultats finaux. On suppose que l'estimation d'une heure après est la plus précise pour tous les 4 valeurs qu'on veut prédire. Cette hypothèse est validée par le phénomène qu'on a vu pendant l'exploration de ce jeux de données. Par exemple, pour la prédiction de NO_2, on voit que les MSE se varient comme suivante :

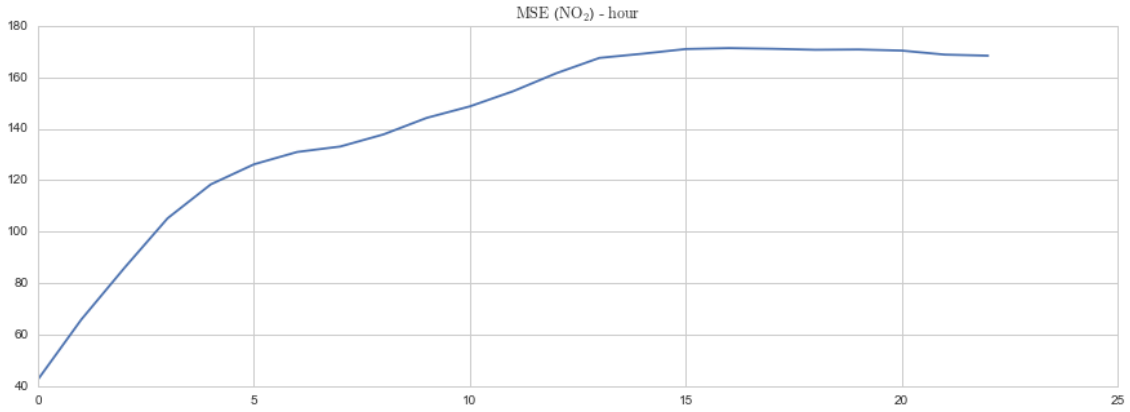

Figure 2: MSE(NO2)-hour

V. Modèle Final et Résultats du Test

Après examen de modèles divers et variés en regard des contraintes précédentes, nous retenons un modèle hybride pour le test final :

- **Pour NO_2, $PM_{2.5}$ et PM_{10}**

  On les prédit par LASSO, méthode relativement stable par rapport à MLP. Comme le modèle se construit par une façon grossière et la linéarité n'est donc pas vérifiée, LASSO est donc un bon candidat pour la prédiction.

- **Pour O_3**

  Compte tenu du potentiel de la régression avec MLP, et de sa bonne prédiction de O_3 sur l'ensemble de test, nous employons la méthode MLP pour prédire la valeur O_3. Cependant, limités par la capacité de nos machines, notre optimisation de la structure de réseaux demeure partielle.

Ensuite, on suppose que l'estimation d'une heure après est la plus précise pour tous les 4 valeurs qu'on veut prédire.

Donc, pour les résultats finaux $Ytest$, on fait une transformation des données avec que la première colonne et la dernière ligne, en respectant la structure de $Ytrain$. Autrement dit, pour objet à prédire donné, on utilise la prédiction de la première heure pour engendrer la valeur qu'on va prédire de n heures après, et on conserve les valeurs qui ne figurent pas dans la prédiction d'une heure après. A ce propos, pour de plus amples détails, nous nous référerons Les détails à l'annexe.

Finalement, notre modèle est validé avec un MSE de 14406.1.

Conclusion

Premièrement, on a fait une analyse préliminaire sur la classification de variables, suivie d'une sélectionn par la méthode classique de statistiques. Ensuite, pour la partie de régression, on a essayé principalement LASSO, ElasticNet et MLP. Après une comparaison entre les différentes méthodes, on introduit un modèle hybride pour la partie régression. A l'issue d'une analyse sur les caractéristiques du jeu de données, on émet une hypothèse sur la précision de la prédiction ce qui opère une transformation sur les résultats et améliore le score. Compte tenu de nos capacité de machines limitées, nous n'avons pas pu ajuster le réseau de neurones de façon optimale.

Il en résulte que le choix de la structure adaptée reste un sujet à explorer. Nous pourrions aussi améliorer nos résultats en considérant la transformation de SVM pour O_3 et l'emploi de machines de plus grande capacité. Une autre chose qu'il reste à faire est de transformer les variables et valeurs à prédire avant la régression. Puisqu'on n'a pas assez de la connaissance a priori sur ce domaine, on n'a pas faits des hypothèse sur la partie de transformation.