

Statistique asymptotique

TD/TP 2 : Régression de Poisson, pénalisation et sur-dispersion

Avant de commencer, récupérer sur ma page le dossier `twitter.zip`.

Les données contiennent des informations sur des tweets. J'ai créé un sous-jeu de données de 50000 observations. Pour chaque tweet sont enregistrés les variables suivantes pour les 6 jours précédents. Pour la variable `var`, le numéro du jour est indiqué avec le tiret `_`

- `NCD`: nombre de discussions créés (colonnes 0 à 5)
- `AI` : author increase, le nombre de nouveaux auteurs (colonnes 6 à 10)
- `ASNA` : attention level, niveau d'attention
- `BL` : burstiness level ; niveau de rafale
- `NAC` : number of atomic containers, nombre de conteneurs atomiques
- `ASNAC` : attention level, niveau d'attention
- `CS` : contribution sparseness, faibleses des contributions
- `AT` : author interaction, interactions avec l'auteur
- `NA` : number of authors, nombre d'auteurs
- `ADL` : average discussions length, longueur moyenne des discussions
- `NDA` : number of active discussion

On veut expliquer et prédire la variable `label` qui correspond à `NDA_6`.

Vous trouverez une description plus détaillée des données sur <https://archive.ics.uci.edu/ml/datasets/Buzz+in+social+media+>

Exercice 1

1. Charger les données et vérifier le nombre d'observations et de variables ainsi que leur type. Faire les changements de type nécessaires.
2. Faire un histogramme pour les variables explicatives au temps 0 et pour le label. Quelle modélisation proposez vous ?
3. Appliquer à une transformation log (appliqué à la variable +1) pour toutes les features dont le maximum dépasse 5.
4. Créer un jeu de données d'apprentissage 80% contenant des observations et un jeu de données de test contenant les 20% restants.
5. Pour chaque feature, calculer la moyenne et l'écart-type sur le train et standardisé les features du train et du test avec ces valeurs.

Exercice 2

1. Faire un premier modèle poissonnien `poisson_all`, en prenant en compte toutes les vfeatures.
2. Calculer les prédictions obtenues avec le modèle sur les individus du train et du test.
3. Créer une fonction qui calcule la MAPE et le log-MSE

$$\text{MAPE} = \frac{1}{n} \sum_i \frac{\hat{Y}_i^P - Y_i}{Y_i + 1}$$
$$\text{log-MSE} = \frac{1}{n} \sum_i (\log(\hat{Y}_i^P + 1) - \log(Y_i + 1))^2.$$

4. Calculer la MAPE et le log-MSE du modèle. A l'avenir, conserver dans un même tableau les valeurs de ces mesures (sur le train et sur le test) dans un même tableau `errors`.

Exercice 3

1. Appliquer une pénalisation elastic-net au modèle précédent (en prenant $\alpha = 0.8$) et en considérant 3 folds pour la cross-validation, on l'appellera `poisson_lasso`
2. Refaire un modèle non-pénalisé en ne considérant que les colonnes dont les coefficients dans le modèle précédent sont non-nuls, on l'appellera `poisson_lasso_refit`
3. Calculer la MAPE et le log-MSE de ces deux modèles. Afficher le tableau `errors`. Les nouveaux modèles ont-ils de meilleures performances que le précédent ?
4. Faire sur ce modèle une sélection de modèles par AIC `poisson_lasso_selected_AIC` ? Est ce que les performances sont améliorées ?

Exercice 4

- A partir du modèle précédent, changer la famille pour une quasi-poisson. Quel est l'estimation du paramètre de sur-dispersion ?
- Quelle conséquence cela a sur les intervalles de confiance et donc sur la sélection de variable ?
- Calculer les résidus de déviance comme définis dans le cours (c'est-à-dire en divisant par le paramètre estimé de dispersion).
- Eliminer les individus pour lesquels le résidus dépasse 4 en valeur absolu et refaire une estimation dans le modèle précédent. Ce nouveau modèle a-t-il de meilleures performances que les précédents ? Expliquer pourquoi.