

# Paramétrage de classifieurs

Damien Nouvel



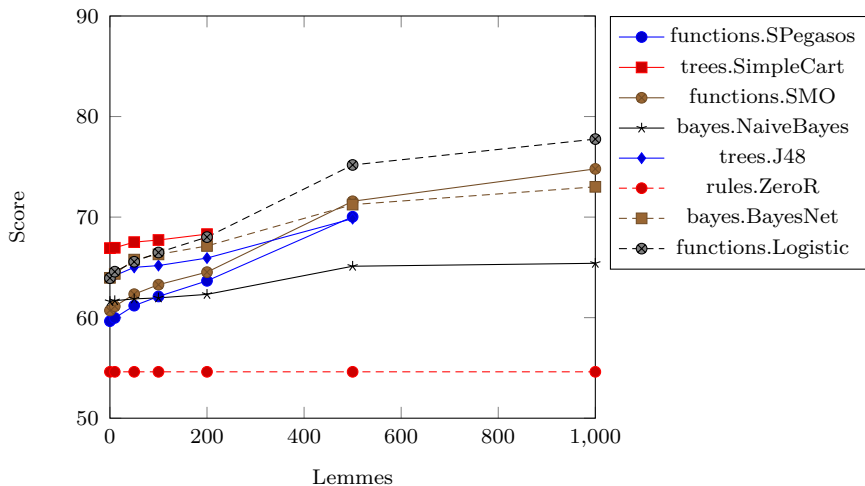
# Plan

1. Visualisation des résultats
2. Sélection de traits
3. Régularisation

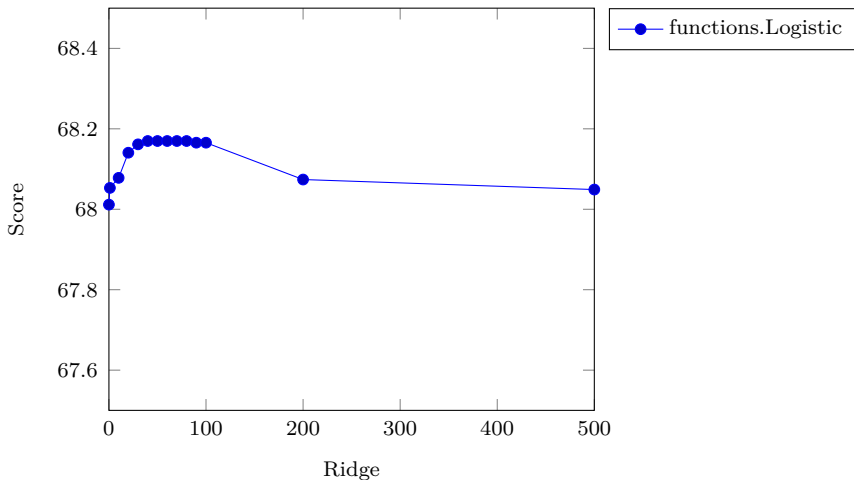
# Modes de visualisation

- ▶ Selon le logiciel
  - Bureautique (office / libreoffice / ...)
    - Script de conversion (csv, tsv)
    - Copier / coller
    - Insertion de graphique
  - Latex
    - Script de conversion format LaTeX
    - Utilisation de tikz / PGFPlots

# Comparaison d'algorithmes



# Valeurs ridge pour la régression logistique



# Plan

1. Visualisation des résultats
2. Sélection de traits
3. Régularisation

# Pourquoi sélectionner

- ▶ Ensemble d'apprentissage volumineux
  - Beaucoup de paramètres à évaluer
  - Pas toujours contrôlés par l'humain
- ⇒ Sélectionner automatiquement les traits **utiles**
- ▶ **Sous-ensemble** de traits :  $2^n$
- ▶ Méthodes de sélection
  - Méta-apprentissage
  - ⇒ Très long
  - Statistiques sur corpus
  - ⇒ Sélection binaire
  - Régularisation
  - ⇒ Pénalisation

# Statistiques pour la sélection

► Traits utiles pour prédire  $Y$  en connaissant  $X$

- Coefficient de Pearson :

$$\frac{\sigma_{xy}}{\sigma_x * \sigma_y}$$

⇒ Favorise les traits corrélés

- Information mutuelle

$$\sum_{x \in X, y \in Y} P(X = x, Y = y) * \log \left( \frac{P(X = x, Y = y)}{P(X = x) * P(Y = y)} \right)$$

⇒ Favorise les traits qui apportent de l'information

- Critère d'Information Bayésien (BIC)

$$-2 \ln(P(X|w^*)) + k * \ln(n)$$

⇒ Sélectionne un nombre de paramètres  $k$  (pour  $n$  données)



# Plan

1. Visualisation des résultats
2. Sélection de traits
3. Régularisation

# Limites des données d'apprentissage

- ▶ Les données d'apprentissage peuvent être trompeuses
  - Présence de **bruit**
  - Données **éparses**
- ⇒ Sélectionner des données utiles
- ▶ Rapport inverse
  - **Fréquence** des traits
  - **Pertinence** pour l'apprentissage
- ⇒ Les traits les plus fréquents sont les moins utiles
- ⇒ Le Graal : très fréquent et très pertinent ...

# Surapprentissage

- ▶ Algorithmes privilégient
  - Beaucoup de traits
  - Peu fréquents
  - Incompatibles
- ⇒ Chaque trait correspond à un exemple
- ⇒ Apprentissage *par cœur*
- ⇒ Aucune chance de les retrouver pour le test !
  - ▶ Limiter ce biais pour
    - Meilleure adaptation au test
    - Modèle plus compréhensible
    - Éviter les erreurs liées au bruit

# Pénaliser les poids

- ▶ Surapprentissage donne des poids  $w_i$  trop importants
- ⇒ **Pénaliser** les poids importants (SVM, V. Vapnik)
- ▶ Méthodes (moindres carrés)
  - Régularisation  $L_1$ 

$$w^* = \operatorname{argmin}_w \sum_i (y_i - w * x_i)^2 + \lambda * |w|$$
  - Régularisation  $L_2$ 

$$w^* = \operatorname{argmin}_w \sum_i (y_i - w * x_i)^2 + \lambda * |w|^2$$
  - Elastic net (combinaison)
 
$$w^* = \operatorname{argmin}_w \sum_i (y_i - w * x_i)^2 + \lambda_1 * |w| + \lambda_2 * |w|^2$$
- ⇒ **Régler** les paramètres de la régularisation !