RJS Aussois. 08/2013.

Forêts aléatoires et détection des irrégularités aux cotisations sociales

Saïp CISS
Modal'X. Université Paris Ouest Nanterre.
Laboratoire d'économétrie de l'Ecole Polytechnique.
Directeurs de thèse : Patrice BERTAIL, Pierre PICARD.

Sécurité sociale

- Recettes 2013 (LFSS) : 460 Mds €
- Dépenses : 470 Mds €

→ Cotisations sociales (entreprises) : 329 Mds €

→ Prestations sociales (maladie, vieillesse, accidents, retraites de base,…) : 340 Mds €

Cotisations sociales

- Recouvrement des cotisations par les URSSAF
- 1 200 000 entreprises (avec au moins un salarié) en France
- 80% entre 1 et 9 salariés
- 15% entre 10 et 49 salariés

URSSAF d'Île-de-France

- Recouvrement (Régime Général) : 80 Mds €
- 400 000 entreprises

→ Recouvrement sur une base déclarative

→ Législation complexe : { > 900 catégories déclaratives} x {assiette, taux, effectif, conditions}

✓ Contrôle des cotisations (2011) :
30 000 entreprises (1 entreprise sur 13).
14 Mds d'€ contrôlés. 170 millions d'€ redressés.

Problématiques

→ Taux de détection des irrégularités : < 55%

→ Rendement : 4 redressements sur 10 < 1 000 €

→ Asymétrie : 250 redressements = 100 millions d'€

→ Contrôle exhaustif impossible : coût prohibitif

→ Ensemble des irrégularités non estimable :

consensus > 6 à 8 Mds d'€/an.

❑ Pour un même nombre de contrôles, réduire le nombre de faux-positifs et augmenter les montants redressés.

Données

- n = 400 000. p > 1 000.
- Catégories déclaratives (cotisations): maladie, vieillesse, accidents, chômage, CSG, heures supplémentaires, aide au logement, versement transport, mesures de réduction,…
- Variables = { catégories déclaratives, durée de vie, effectif, masse salariale, secteur d'activité, régularisations, retards de paiement,…}
- Beaucoup de zéros (catégories non déclarées): > 89%
- Variables individuellement peu discriminantes.

Modèle

- Apprentissage statistique : « apprendre » la relation entre les déclarations de cotisation (X) et les irrégularités (Y), à partir des contrôles effectués. Généraliser à l'ensemble des déclarations de toutes les entreprises.
- « Apprendre » consiste à construire un (ou plusieurs) classifieur(s).

Arbre de décision (classifieur)

→ Structure algorithmique: partition de l'espace en 2 régions. « Récursion » jusqu'à l'atteinte d'une ou plusieurs conditions d'arrêt. Décision.

→ Plusieurs types: CART, ID3,…

→ Problématiques: définition d'une région, conditions d'arrêt, règle de décision. Instabilité.

Arbre de décision uniformément aléatoire

$X \in \Re^d, Y \in \{0, 1\}$

- $D_n = \{(X_i, Y_i), 1 \leq i \leq n\}$ est l'échantillon d'apprentissage
- $x = (x^{(1)}, ..., x^{(d)})$ un vecteur d'observations
- R est une région de la partition courante $\mathcal{P}$ si $R \cap T = \emptyset \; ou \; R \subseteq T, \forall \; T \in \mathcal{P}$.

Arbre de décision uniformément aléatoire

A chaque étape du partitionnement, tirer, avec remise, $\lceil \beta d \rceil$ variables, $\beta > 0$, et construire autant de régions.

$x \in R, si\ x \leq \alpha\ et\ x \in R^c,\ si\ x > \alpha,$ avec

$R \cap R^c = \emptyset\ et\ \alpha \in \mathcal{U}_{]min(X),max(X)[}$.

La région qui maximise un certain critère est retenue:

- Soit

$$\mathtt{H}(Y|D_n) = -\sum_{c=0}^{1} \left\{ \frac{1}{n} \sum_{i=1}^{n} \mathbf{I}_{\{Y_i=c\}} \log \left(\frac{1}{n} \sum_{i=1}^{n} \mathbf{I}_{\{Y_i=c\}} \right) \right\}$$

Arbre de décision uniformément aléatoire

$$\mathtt{IG}(j, D_n) = \mathtt{H}(Y|D_n) - [\, \mathtt{H}((Y|X^{(j)} \leq \alpha)|D_n) + \mathtt{H}((Y|X^{(j)} > \alpha)|D_n)\,]$$

- ✓ Région aléatoire optimale:

$$R = R(j^*, \alpha) \,/\, j^* = \underset{j \in [1,d]}{\mathtt{arg\ max}}\ \mathtt{IG}(j, D_n)$$

- ✓ Règle de décision (vote majoritaire):

$$g_{\mathcal{P}}(x) = \begin{cases} 1, \ si \ \sum_{i=1}^{n} \mathbf{I}_{\{X_i \in R, Y_i = 1\}} > \sum_{i=1}^{n} \mathbf{I}_{\{X_i \in R, Y_i = 0\}}, \ x \in R \\ \\ 0, \ sinon. \end{cases}$$

Forêt uniformément aléatoire

- Définition (courte): collection d'arbres dont la règle de décision est le vote majoritaire parmi l'ensemble.

- Soit $\{g_{\mathcal{P}}^{(b)}, 1 \leq b \leq B\}$, une suite de classifieurs, alors la règle de décision de la forêt uniformément aléatoire est notée :

$$\bar{g}_{\mathcal{P}}^{(B)}(x) = \begin{cases} 1, \ si \ \sum_{b=1}^{B} \mathbf{I}_{\{g_{\mathcal{P}}^{(b)}(x)=1\}} > \sum_{b=1}^{B} \mathbf{I}_{\{g_{\mathcal{P}}^{(b)}(x)=0\}} \\ \\ 0, \ sinon. \end{cases}$$

Algorithme : randomUniformForest

Forêt uniformément aléatoire incrémentale

→ Apprendre au fur et à mesure que les nouvelles données sont disponibles:

$$\bar{g}^{(B)}_{\mathcal{P},big} = \begin{cases} 1, \ si \ \sum_{s=1}^{S}\sum_{b=1}^{B} \mathbf{I}_{\left\{g^{(b)}_{\mathcal{P}_s}(x)=1\right\}} > \sum_{s=1}^{S}\sum_{b=1}^{B} \mathbf{I}_{\left\{g^{(b)}_{\mathcal{P}_s}(x)=0\right\}} \\ \\ 0, \ sinon \end{cases}$$

- ✓ Meilleures performances (généralement) en classification.
- ✓ Pas de modification des arbres déjà construits. « Mémoire ».
- ✓ Corollaire: « preprocessing » des données. Optimisation globale des arbres.

Algorithme : growingRandomUniformForest (growingRUF)

Données synthétiques

$$Z \sim \mathcal{U}_{[-10,10]}.\ \epsilon_1 \sim \mathcal{U}_{[-1,1]}.\ \epsilon_2 \sim \mathcal{U}_{[-1,1]}.$$

$$X^{(j)} \sim \mathcal{N}(z_j, z_j^2).\ X = (X^{(1)}, X^{(2)}, ..., X^{(10)}).$$

$$R(X, \epsilon) = 2(X^{(1)}X^{(2)} + X^{(3)}X^{(4)}) + \epsilon_1 X^{(5)} + \epsilon_2 X^{(6)}.$$

$$Y = \mathbf{I}_{\{R(X,\epsilon) > \bar{R}(X,\epsilon)\}}.$$

n = 1 000. Echantillon d'apprentissage: 50%.

Données synthétiques

- Résultats (paramètres par défaut B = 500, β = 4/3)

OOB estimate of error rate: 8.8%
OOB confusion matrix:

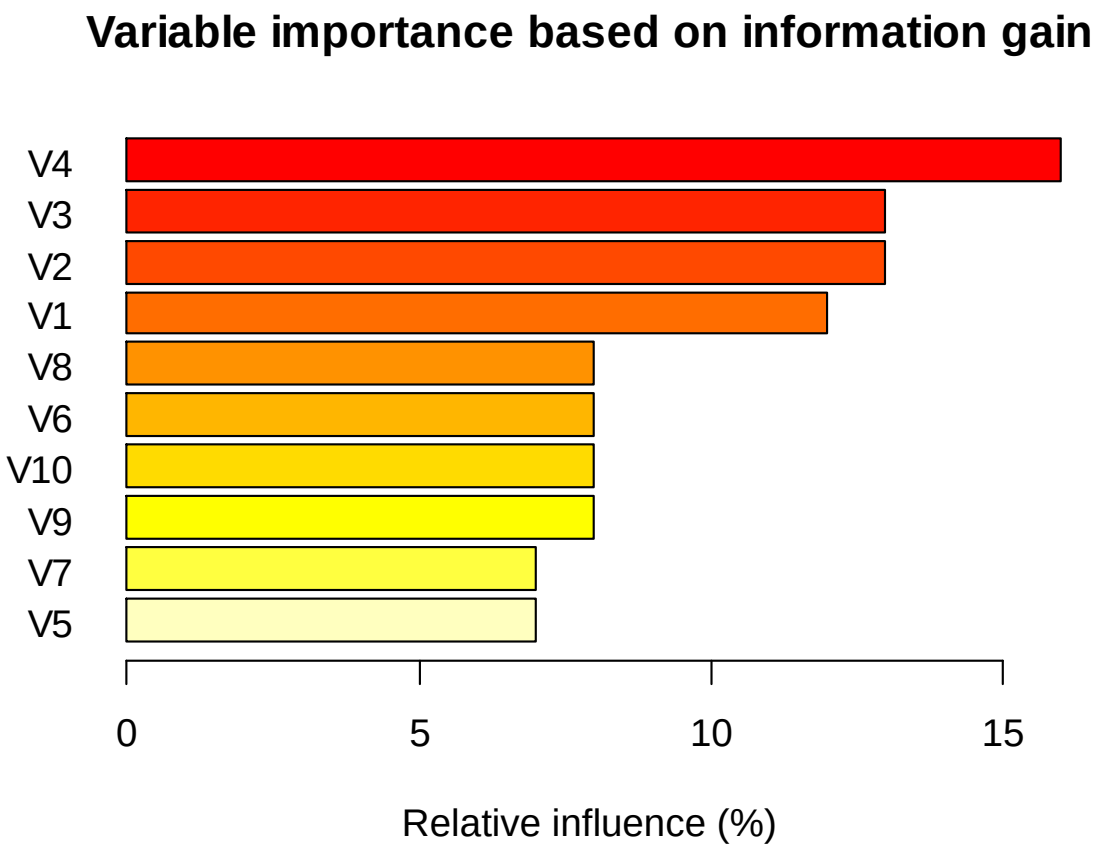
	0	1	class.error
0	221	27	0.1089
1	17	235	0.0675

Theorical (Breiman) bounds:
Prediction error (expected to be lower than): 9.28%
Trees average correlation: 0.0588
Strength (margin): 0.6227
Standard deviation of strength: 0.2852

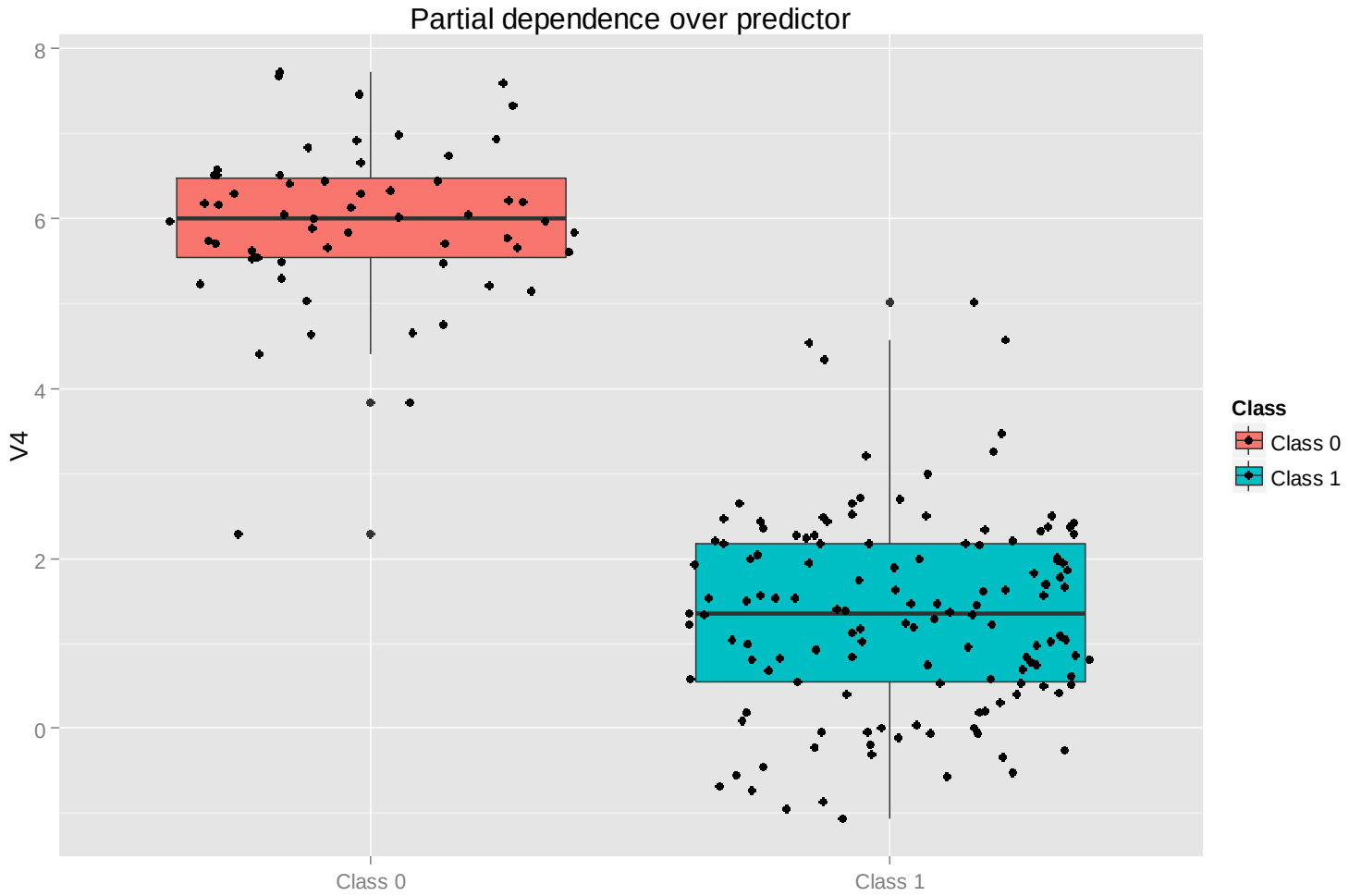
Test set:
Error rate: 6.6%
Confusion matrix:

	0	1	class.error
0	214	16	0.0696
1	17	253	0.0630

Données synthétiques: exemples de visualisation



Données synthétiques: exemples de visualisation



Détecter les irrégularités aux cotisations sociales

- **Laboratoire :**

→ 4069 contrôles de 2011 (après filtrage)

→ 1065 variables

→ 1698 irrégularités (cas positifs)

→ tirage aléatoire : 10% entraînement, 90% test (conformité à la réalité opérationnelle) répété 100 fois.

→ **R** : packages randomForest (Breiman), gbm (« stochastic gradient boosting », Friedman)

→Paramètres par défaut (sauf gbm et growingRUF)

→growingRUF: données d'entraînement facultatives.

Détecter les irrégularités aux cotisations sociales

	Erreur de test	Précision (écart-type)	AUC
randomForest	0.2729	72.55% (0.0374)	0.7319
Sto.GradientBoosting	0.2473	70.08% (0.0191)	0.7467
randomUniformForest	0.2419	70.44% (0.0235)	0.7534
growingRUF	0.2324	79.03% (-)	0.7737

→ « Précision » = rapport entre le nombre de cas positifs correctement prédits et le nombre de cas positifs prédits.

✓ Augmenter la « précision » est équivalent à réduire les faux positifs: le nombre de contrôles physiquement réalisables est limité.

→ Plusieurs options : « subsampling », « over/under sampling », « class reweighting », « class cut-offs »,…

Détecter les irrégularités aux cotisations sociales

- **Expérimentation réelle (2012, URSSAF d'IdF) :**

→ 167 contrôles réalisés sur la base du modèle

→ Pas de biais de sélection (preprocessing)

→ Irrégularités détectées: 69%

→ rendement moyen: 5 300€/contrôle

→ Montant total net redressé: 885 000 €.

Détecter les irrégularités aux cotisations sociales

- **Phase industrielle : En 2013, en Île-de-France…**

→ Modèle: growingRandomUniformForest

→ > 50 000 irrégularités estimées

→ Faux positifs: < 30%

→ Rendement moyen estimé: > 5 000 €/contrôle

→ Montant total net estimé des redressements:
> 250 millions d'€, < 1 500 millions d'€.

❑ Phase industrielle abandonnée par l'URSSAF après le départ de l'ex-équipe dirigeante.

Sources et Références

- Biau, Devroye, Lugosi, 2008. « Consistency of random forests and others averaging classifiers »
- Breiman, Friedman, Olshen, Stone, 1984. « Classification And Regression Trees » (CART)
- Breiman, 1996. « Bagging Predictors »
- Breiman, 2001. « Random Forests »
- Breiman web site : http://www.stat.berkeley.edu/~breiman/RandomForests/cc_home.htm
- **Devroye, Györfi, Lugosi, 1996. « A Probabilistic Theory of Pattern Recognition »**
- Friedman, 1999. « Stochastic Gradient Boosting »
- Friedman, 2001. « Greedy function approximation : A gradient boosting machine »
- **Hastie, Tibshirani, Friedman, 2009. « The Elements of Statistical Learning ». 2nd Edition**
- **Vapnik, 1995. « The Nature of Statistical Learning Theory ».**

Merci.