Validation croisée et modèles statistiques appliqués

Matthieu Cornec

Université Paris X, Nanterre
Thèse dirigée par le Professeur Patrice Bertail

Le 4 Juin 2009

Plan

1. **Ch1 : ERM**
2. **Ch2 : Prédicteurs stables**
3. **Ch3 : Subagging**

Validation croisée : méthode très populaire depuis 40 ans au sein des practiciens pour estimer la performance d'une méthode. 200000 citations sous Google Scholar. Mais, questions ouvertes :

- Est-ce que ça marche dans un cadre général ?
- Faut-il découper en 2 ou en 10 ?
- Quand l'estimateur de validation croisée est-il meilleur que l'erreur de resubstitution ?

Remarque

Validation croisée utilisée dans la résolution d'un grand nombre de problèmes: la sélection de modèle, l'adaptivité et l'identification. Elle peut être vue comme une application particulière des méthodes dites de rééchantillonnage.

Principales méthodes de validation croisée

- leave-one-out cross-validation
- k-fold cross-validation
- Leave-v-out cross-validation
- split-sample method

Apprentissage statistique : grands principes

Définition (Prédicteur)

$\phi : \mathcal{X} \to \mathcal{Y}$, $\phi(X) = \widehat{Y}$

Définition (Learning set)

$\mathcal{D}_n = (X_i, Y_i)_{1 \leq i \leq n} \in \mathcal{X} \times \mathcal{Y}$ *i.i.d. variables aléatoires avec distribution inconnue* $\mathbb{P}$.

Définition (Learning algorithm)

$\Phi : \mathcal{X} \times \cup_n (\mathcal{X} \times \mathcal{Y})^n \to \mathcal{Y}$, $\Phi(X, \mathcal{D}_n) = \widehat{Y}$

Fonction de perte

Définition (Fonction de perte)

$L : \mathcal{Y}^2 \to \mathbb{R}^+$

Exemple

- *Pour la classification,* $L(y, \widehat{y}) = 1(y \neq \widehat{y})$.
- *Pour les problèmes de régression,* $L(y, \widehat{y}) = (y - \widehat{y})^2$.

Erreur de généralisation

Définition (Erreur de généralisation)

$$\begin{aligned} \widetilde{R}_n = \widetilde{R}_n(\phi) \quad &: \quad = \mathbb{E}_P[L(Y, \phi(X, \mathcal{D}_n)) \mid \mathcal{D}_n] \\ R(\phi) \quad &: \quad = \mathbb{E}_P[L(Y, \phi(X))] \end{aligned}$$

Erreur de resubstitution ou d'apprentissage

Définition (Erreur de resubstitution ou d'apprentissage)

$\widehat{R}_n := \frac{1}{n}\sum_{i=1}^{n} L(Y_i, \phi(X_i, \mathcal{D}_n))$

Exemple

L'erreur de resubstitution pour le 1*-plus proche voisin est toujours* 0 *quelque soit l'erreur de généralisation.*

Deux résultats importants

Théorème (Inégalité d'Hoeffding)

Soient $X_1,\ldots,X_n$ des variables aléatoires indépendantes dans $[a_i, b_i]$.
Alors pour tout $\epsilon > 0$,

$$\mathbb{P}(\sum X_i - \mathbb{E}(\sum X_i) \geq n\epsilon) \leq e^{-\frac{2\epsilon^2}{\sum_i (b_i - a_i)^2}}$$

Deux résultats importants

Théorème (Inégalité de Vapnik-Chervonenkis)

Soit $\mathcal{C}$ une classe de prédicteurs de dimension $V_{\mathcal{C}}$ et L une fonction de perte bornée par 1. Alors pour tout $\epsilon > 0$,

$$\mathbb{P}(\sup_{\phi \in \mathcal{C}}(\widehat{R}_n(\phi) - R(\phi)) \geq \epsilon) \leq c(n, V_{\mathcal{C}}) e^{-\frac{\epsilon^2}{2\sigma(n)^2}}$$

avec $c(n, V_{\mathcal{C}}) = 2\mathcal{S}(2n, \mathcal{C}) \leq 2(2n+1)^{V_{\mathcal{C}}}$ et si $n \geq V_{\mathcal{C}}$, $2\mathcal{S}(2n, \mathcal{C}) \leq 2(\frac{2ne}{V_{\mathcal{C}}})^{V_{\mathcal{C}}}$ et $\sigma(n)^2 = \frac{4}{n}$

Vecteurs de réechantillonnage

Définition (Vecteur binaire)

Soit $(v_{n,i})_{1\leq i\leq n}$ *un vecteur de réels de taille* n. v_n *est un vecteur binaire si* $\forall i, v_{n,i} \in \{0,1\}$ *et si* $\sum_i v_{n,i} \neq 0$.

Définition (Sous échantillon)

Soit $\mathcal{D}_n = (X_i, Y_i)_{1\leq i\leq n}$ *un learning set. Pour un vecteur binaire donné* $(v_{n,i})_{1\leq i\leq n}$ *de taille* n, *nous définissons le sous-échantillon par :*

$$\mathcal{D}_{v_n} := \{(X_i, Y_i) | v_{n,i} = 1, 1 \leq i \leq n\}$$

Définition (learning algorithm construit à partir d'un sous-échantillon)

Soit $\mathcal{D}_n = (X_i, Y_i)_i$ *un learning set. Pour un vecteur binaire* $(v_{n,i})_{1\leq i\leq n}$ *de taille* n, *nous définissons un learning algorithm construite à partir du sous-échantillon :*

$$\phi_{v_n}(.) := \phi(., \mathcal{D}_{v_n})$$

Erreur de validation croisée

Définition (Erreur empirique pondérée)

Soit $\mathcal{D}_n$ un learning set. Soit ϕ un prédicteur et L une fonction de perte. Pour un vecteur binaire donné $(v_{n,i})_{1\leq i\leq n}$ de taille n, nous définissons une erreur empirique pondérée notée $\widehat{R}_{v_n}(\phi)$ associée à ϕ :

$$\widehat{R}_{v_n}(\phi) := \frac{1}{\sum_i v_{n,i}} \sum_{i=1}^{n} v_{n,i} L(Y_i, \phi(X_i))$$

Définition (Vecteur d'apprentissage)

Soit $\mathcal{D}_n$ un learning set. Soit V_n^{tr} un vecteur binaire aléatoire de taille n indépendant de $\mathcal{D}_n$, appelé vecteur d'apprentissage. Nous notons $V_n^{tr} = 1_n - V_n^{tr}$ le vecteur de test.

Erreur de validation croisée

Définition (Erreur de validation croisée)

Soit $\mathcal{D}_n$ un learning set. Soit ϕ_n un learning algorithm et L une fonction de perte. Soit V_n^{tr} un vecteur d'apprentissage. Avec les notations précédentes, l'erreur de validation croisée notée $\widehat{R}_{CV}(\phi_n)$ est définie par :

$$\widehat{R}_{CV}(\phi_n) := \mathbb{E}_{V_n^{tr}} \widehat{R}_{V_n^{ts}}(\phi_{V_n^{tr}})$$

Exemples d'utilisation des notations

Exemple (leave-one-out cross-validation)

$$\Pr(V_n^{tr} = (0, 1, \ldots, 1)) = \frac{1}{n}$$
$$\Pr(V_n^{tr} = (1, 0, 1, \ldots, 1)) = \frac{1}{n}$$
$$\ldots$$
$$\Pr(V_n^{tr} = (1, \ldots, 1, 0)) = \frac{1}{n}.$$

Objectif de la première partie de la thèse est d'obtenir dans un cadre général des inégalités du type :

$$\mathbb{P}(|\widehat{R}_{CV} - \widetilde{R}_n| \geq \varepsilon) \leq B(n, p_n, \varepsilon) + V(n, p_n, \varepsilon)$$

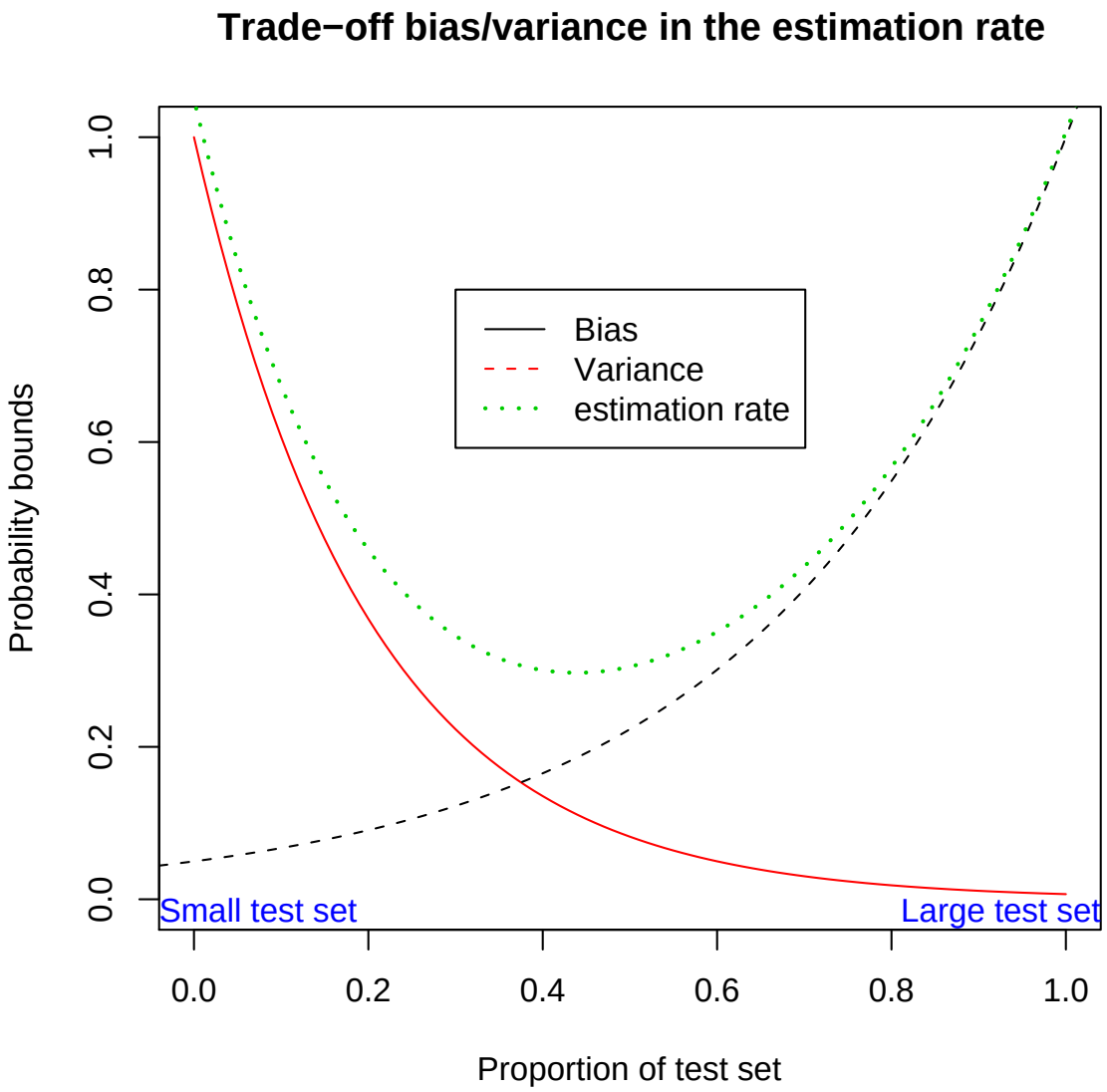
avec p_n pourcentage d'observations dans le test set et où $B(n, p_n, \varepsilon)$ et $V(n, p_n, \varepsilon)$ peuvent s'interpréter comme problème biais/variance.

Remarque

On aimerait en outre que

- *que les bornes ne se détériorent pas avec le rééchantillonnage*
- *que les bornes discriminent la $k-$fold validation croisée du split sample.*

Trade-off biais/variance



Résumé des résultats sur la validation croisée

	leave-one-out	hold-out	k-fold	ν-out	resubsitution
VC-algorithm	Kearn, Holden	Holden	Holden	$\times$	Vapnik-Cher
Hypothesis stability	Devroye and W	Devroye and W	$\times$	$\times$	$\times$
Error stability	Kearn	Kearn	$\times$	$\times$	$\times$
Uniform stability	Bousquet and E.	$\times$	$\times$	$\times$	$\times$
Partial stability	Kutin and N	$\times$	$\times$	$\times$	$\times$

Table: Bornes manquantes $\times$ à trouver

Résultats de la thèse

1. Chapitre 1 : Minimisateur du risque empirique d'une classe de VC-dimension finie.
2. Chapitre 2 : Algorithmes stables (k-plus proches voisins, adaboost, regularization networks).
3. Chapitre 3 : Amélioriation des bornes précédentes avec le Subagging.

Hypothèses Chapitre 1

1. le training set et le test set disjoints
2. nous supposons que les prédicteurs ϕ_n des minimiseurs du risque empirique de VC-dimension $V_{\mathcal{C}}$ et L une fonction de perte bornée par 1.
3. le nombre d'observations dans le training set et dans le test set sont respectivement $n(1-p_n)$ and np_n

Notons ces hypothèses $\mathcal{H}$.

Validation croisée avec de grands test sets

Corollaire (Déviation absolue pour les grands échantillons)

Supposons $\mathcal{H}$. Alors, nous avons, pour tout $\varepsilon > 0$,

$$\Pr(|\widetilde{R}_n - \widehat{R}_{CV}| \geq \varepsilon) \leq B(n, p_n, \varepsilon) + V(n, p_n, \varepsilon),$$

avec

- $B(n, p_n, \varepsilon) = 5(2n(1-p_n)+1)^{\frac{4V_{\mathcal{C}}}{1-p_n}} \exp(-\frac{n\varepsilon^2}{25})$, *une borne de type Vapnik-Chervonenkis*
- $V(n, p_n, \varepsilon) = \exp(-\frac{2np_n\varepsilon^2}{25})$, *une borne de type Hoeffding.*

Validation croisée avec de petits ensembles de test

Corollaire (Déviation absolue pour les petits échantillons)

Supposons $\mathcal{H}$. Alors, nous avons pour tout $\varepsilon > 0$,

$$\Pr(|\widetilde{R}_n - \widehat{R}_{CV}| \geq \varepsilon) \leq B(n, p_n, \varepsilon) + V(n, p_n, \varepsilon),$$

- $B(n, p_n, \varepsilon) = 5(2n(1-p_n)+1)^{\frac{4V_{\mathcal{C}}}{1-p_n}} \exp(-\frac{n\varepsilon^2}{64})$
- $V(n, p_n, \varepsilon) = \frac{16}{\varepsilon}\sqrt{\frac{V_{\mathcal{C}}(\ln(2n(1-p_n)+1)+4)}{n(1-p_n)}}.$

k-fold cross-validation

Proposition (k-fold)

Supposons $\mathcal{H}$. Nous avons la borne suivante pour la k-fold cross-validation:

$$\mathbb{P}(|\widetilde{R}_n - \widehat{R}_{CV}| \geq \varepsilon) \leq 4(2n(1-p_n)+1)^{\frac{4V_{\mathcal{C}}}{1-p_n}} \exp(-\frac{n\epsilon^2}{9}) + 22^{\frac{1}{p_n}} \exp\left(-\frac{n\epsilon^2}{16(\sqrt{V_{\mathcal{C}}\log(2(2np_n+1))}+2)}\right)$$

Remarque

Fait important, il n'est pas nécessaire que la taille du test set croisse vers l'infini pour que la validation croisée soit consistante.

k-fold cross-validation

Corollaire (L_1 error for large test sample)

Supposons $\mathcal{H}$. Alors, nous avons:

$$\mathbb{E}|\widehat{R}_{CV} - \widetilde{R}_n| \leq 10\sqrt{\frac{V_{\mathcal{C}}(\ln(2n(1-p_n)+1)+4)}{n(1-p_n)}} + 5\sqrt{\frac{2}{np_n}}.$$

Remark

Nous pouvons montrer qu'une règle optimale est alors $p_n^\star \propto V_{\mathcal{C}}^{-1/3}$

Transition

- k-plus proches voisins, ou boosting sont de VC-dimension infinie
- chapitre 2 : introduction de la notion d'algorithmes stables.

Définition

$$\mathcal{Z} := \mathcal{X} \times \mathcal{Y}$$

$$(Z_i)_{1 \leq i \leq n} := ((X_i, Y_i))_{1 \leq i \leq n}$$

$$\mathcal{F} := \{\psi \in \mathbb{R}_+^{\mathcal{Z}} | \psi(Z) = L(Y, \phi(X)), \phi \in \mathcal{G}\}$$

$$\Psi(Z, \mathcal{D}_n) := L(Y, \Phi(X, \mathcal{D}_n))$$

Notations

Définition

$$\widetilde{R}_n := \mathbb{E}_Z[\Psi(Z, \mathcal{D}_n) \mid \mathcal{D}_n]$$

$$\psi_{V_n}(.) := \Psi(., \mathcal{D}_{V_n}).$$

Distance entre deux prédicteurs

Définition

Soient $\psi_1, \psi_2 \in \mathcal{F}$. La distance uniforme est définie par :

$$d_\infty(\psi_1, \psi_2) = \sup_{Z \in \mathcal{Z}} |\psi_1(Z) - \psi_2(Z)|$$

la distance L_1 par

$$d_1(\psi_1, \psi_2) = \mathbb{P}|\psi_1 - \psi_2|$$

la distance erreur par

$$d_e(\psi_1, \psi_2) = |\mathbb{P}(\psi_1 - \psi_2)|.$$

Definition stabilité

Définition (Stabilité faible)

Soit $\mathcal{D}_n$ un échantillon initial. Soit $\alpha > 0$. Soient $\lambda, (\delta_{n,p_n})_{n,p_n}$ des réels positifs. Un algorithme d'apprentissage Ψ est dit faiblement $(\lambda, (\delta_{n,p_n})_{n,p_n}, d)$ stable si pour tout vecteur d'apprentissage u_n dont la somme des composantes est égale à $n(1-p_n)$:

$$\Pr(d(\psi_{u_n}, \psi_n) \geq \lambda ||\mathbb{P}_{n,u_n} - \mathbb{P}_n||^\alpha) \leq \delta_{n,p_n}.$$

Pour les algorithmes classiques, nous avons à l'esprit $\delta_{n,p_n} = O_n(\exp(-n(1-p_n)a))$.

Definition stabilité et validation croisée

Définition (Validation croisée et stabilité faible)

Soit $\mathcal{D}_n = (Z_i)_{1 \leq i \leq n}$ *un échantillon. Soit* V_n^{tr} *un vecteur d'apprentissage de distribution* $\mathbb{Q}$*. Soient* $\lambda, (\delta_{n,p_n})_{n,p_n}$ *des réels positifs. Un algorithme d'apprentissage* Ψ *est dit être faiblement* $(\lambda, (\delta_{n,p_n})_{n,p_n}, d, \mathbb{Q})$ *stable s'il est faiblement* $(\lambda, (\delta_{n,p_n})_{n,p_n}, d)$ *stable et si :*

$$\Pr\Big(\sup_{U_n \in support(\mathbb{Q})} \frac{d(\psi_{U_n}, \psi_n)}{||\mathbb{P}_{n,U_n} - \mathbb{P}_n||^{\alpha}} \geq \lambda \Big) \leq \delta_{n,p_n}.$$

Definition stabilité et validation croisée

Définition (Stabilité forte)

Soit $z \in \mathcal{Z}$. Soit $\mathcal{D}_n = \mathcal{D}_{n-1} \cup \{z\}$ un échantillon initial. Soient $\lambda, (\delta_{n,p_n})_{n,p_n}$ des réels positifs. Un algorithme d'apprentissage Ψ est dit fortement $(\lambda, \delta_{n,p_n}, d)$ stable si pour tout vecteur d'apprentissage U_n dont la somme des composantes est égale à $n(1-p_n)$:

$$\Pr(d(\psi_{U_n}, \psi_n) \geq \lambda ||\mathbb{P}_{n,U_n} - \mathbb{P}_n||^{\alpha}) \leq \delta_{n,p_n}.$$

Définition stabilité et validation croisée

Définition (Validation croisée et stabilité forte)

Soit $z \in \mathcal{Z}$. Soit $\mathcal{D}_n = \mathcal{D}_{n-1} \cup \{z\}$ un échantillon. Soit V_n^{tr} un vecteur d'apprentissage de distribution $\mathbb{Q}$. Un algorithme d'apprentissage Ψ est dit être fortement $(\lambda, (\delta_{n,p_n})_{n,p_n}, d, \mathbb{Q})$ stable s'il est fortement $(\lambda, (\delta_{n,p_n})_{n,p_n}, d)$ stable et si :

$$\Pr\Big(\sup_{U_n \in support(\mathbb{Q})} \frac{d(\psi_{U_n}, \psi_n)}{||\mathbb{P}_{n,U_n} - \mathbb{P}_n||^{\alpha}} \geq \lambda \Big) \leq \delta_{n,p_n}.$$

Définition (Stabilité sûre)

Remarquons que si $\delta_{n,p_n} = 0$, les deux notions coincident et nous appelons alors la ***stabilité sûre****.*

Définition stabilité et validation croisée

distance	d_∞	d_1	d_e
Faible			Lasso
Fort	Adaboost	ERM k-nearest rule	Bayesian algorithm
Sure	Regularization networks		

Définition stabilité et validation forte

Théorème (Validation croisée et stabilité forte)

Supposons $\mathcal{H}$. Soit Ψ une machine learning qui est fortement $(\lambda, (\delta_{n,p_n})_{n,p_n}, d, \mathbb{Q})$ stable. Alors, pour tout $\varepsilon \geq 0$,

$$\Pr\left(|\widehat{R}_{CV} - \widetilde{R}_n| \geq \varepsilon + \lambda(2p_n)^\alpha\right) \leq 2\exp(-2np_n\varepsilon^2) + \delta_{n,p_n}.$$

De plus, si d est la distance uniforme d_∞, alors nous avons pour tout $\varepsilon \geq 0$:

$$\begin{aligned}\Pr\left(|\widehat{R}_{CV} - \widetilde{R}_n| \geq \varepsilon + \delta_{n,p_n} + \lambda(2p_n)^\alpha\right) \leq & 4(\exp(-\frac{\varepsilon^2}{8(10\lambda)^2 n(2p_n)^{2\alpha}}) \\ & + \frac{n}{5\lambda(2p_n)^\alpha}\delta'_{n,p_n}).\end{aligned}$$

avec $\delta'_{n,p_n} = \delta_{n,p_n} + (n+1)\delta_{n+1,1/(n+1)}$.

Définition stabilité et validation croisée

Théorème (L_1 norme)

Supposons $\mathcal{H}$ et $\alpha = 1$. Soit Ψ une machine learning qui est faiblement $(\lambda, (\delta_{n,p_n})_{n,p_n}, d, mathbbQ)$ stable. Alors, nous avons :

$$\mathbb{E}_{\mathcal{D}_n} |\widehat{R}_{CV} - \widetilde{R}_n| \leq 2\lambda p_n + \sqrt{\frac{2}{np_n}} + \delta_{n,p_n}.$$

De plus, si Ψ est une règle d'apprentissage qui est fortement $(\lambda, (\delta_n)_n, d_\infty, \mathbb{Q})$ stable, nous avons :

$$\mathbb{E}_{\mathcal{D}_n} |\widehat{R}_{CV} - \widetilde{R}_n| \leq \delta_{n,p_n} + 2\lambda p_n + 51\lambda\sqrt{n}p_n + \frac{n}{9\lambda p_n}\delta'_{n,p_n},$$

avec $\delta'_{n,p_n} = \delta_{n,p_n} + (n+1)\delta_{n+1,1/n+1}$

Definition stabilité et validation croisée

Corollaire

Supposons $\mathcal{H}$. Soit Ψ une machine learning faiblement $(\lambda, (\delta_{n,p_n})_{n,p_n}, d)$ stable, nous définissons une règle de séparation optimale $p_n^\star = p_n^\star \propto (\lambda)^{2/3}(1/n)^{1/3}$. Alors, nous obtenons :

$$\mathbb{E}_{\mathcal{D}_n}|\widehat{R}_{CV} - \widetilde{R}_n| \leq O_n((\lambda/n)^{1/3}).$$

De plus, si Ψ est une machine learning qui est fortement $(\lambda, (\delta_n)_n, d_\infty, \mathbb{Q})$ stable, nous utilisons la leave-one-out cross-validation pour n assez grand. Nous avons alors

$$\mathbb{E}_{\mathcal{D}_n}|\widehat{R}_{CV} - \widetilde{R}_n| = O_n(\lambda/\sqrt{n}).$$

Transition

Les limites sont :

- *"sanity check bounds"* dans l'esprit de [KEA95]

$$V(n, p_n, \varepsilon) + B(n, p_n, \varepsilon) >> 1$$

  pour n petit.
- dans le cadre de la minimisation empirique du risque,

$$VC(n, \varepsilon) \leq V(n, p_n, \varepsilon) + B(n, p_n, \varepsilon)$$

Le but du chapitre 3 est d'utiliser le réechantillonnage par le subagging pour améliorer ces résultats :

$$\Pr(|\hat{R}_{CV} - \tilde{R}_n| \geq \varepsilon) \leq \min(B(n, p_n, \varepsilon), V(n, p_n, \varepsilon)) < 1$$

Definition subagging

Définition (Regresseur subaggé)

Le prédicteur subaggé construit à partir de ϕ_n et noté ϕ_n^B est défini par:

$$\phi_n^B(.) := \mathbb{E}_{V_n^{tr}} \phi_{V_n^{tr}}(.)$$

Dans le cas des classificateurs, la règle bagging correspond au vote à la majorité. Nous supposons dans ce cas que $\mathcal{Y} = \{1, \ldots, M\}$.

Définition (Classificateur subaggé.)

Le classificateur subaggé dénoté ϕ_n^B peut être défini par:

$$\phi_n^B(X) := \arg \min_{k \in \{1,\ldots,M\}} \mathbb{E}_{V_n^{tr}} L(k, \Phi(X, \mathcal{D}_{V_n^{tr}}))$$

Definition validation croisée

L'idée est d'adapter la validation croisée au prédicteur subaggé :

Définition (Estimateur de validation croisée pour le subagging)

L'estimateur de validation croisée adapté au prédicteur subaggé ϕ_n^B est défini par :

$$\widehat{R}_{CV}^{Out}(\Phi_n^B) := \mathbb{E}_{V_n^{tr}} \mathbb{P}_{n,V_n^{ts}}(\psi_{V_n^{tr}})$$

Hypothèses

- la validation croisée symétrique.
- ϕ_n appartient à une famille de prédicteurs de VC-dimension finie.
- L est supposée bornée de la façon suivante : $L(Y, \phi(X)) \leq C(h(Y, \phi(X))$ avec C une fonction convexe -bornée elle-même par 1 sur le support de $h(Y, \phi_{V_n^{tr}}(X)$-,
- h telle que pour n'importe quel λ, $0 < \lambda < 1$, nous avons

$$h(y, \lambda\phi(x_1)+(1-\lambda)\phi(x_2)) = \lambda h(y, \phi(x_1))+(1-\lambda)h(y, \phi(x_2)).$$

Nous notons ces hypothèses $\mathcal{H}$.

Etat de l'art

- [BRE96] introduit le bagging.
- [BUH00] considère des prédicteurs non différentiables et discontinus
- [GRA04] apporte de nouveaux arguments pour expliquer l'effet du bagging : les améliorations/détériorations du bagging sont expliquées par l'influence positive ou négative d'exemples très influençants.
- [ELI05] insiste sur le caractère stabilisant du bagging sur le prédicteur d'origine et prouve des bornes non asymptotiques sur la déviation entre l'erreur de généralisation et l'erreur leave one out.
- [BUJ02] propose des statistiques quadratiques générales pour lesquelles le prédicteur baggé augmente à la fois le biais et la variance.

Resultats

Théorème (Déviation absolue pour la validation croisée symétrique)

Supposons $\mathcal{H}$. Alors, nous avons pour tout $\varepsilon > 0$,

$$\Pr(\widetilde{R}_n(\Phi_n^B) - \hat{R}_{CV}^{Out} \geq \varepsilon) \leq \min(B_{sym}(n, p_n, \varepsilon), V_{sym}(n, p_n, \varepsilon)) < 1$$

avec

- $B_{sym}(n, p_n, \varepsilon) = (2np_n + 1)^{4V_{\mathcal{C}}/p_n} e^{-n\varepsilon^2}$
- $V_{sym}(n, p_n, \varepsilon) = \exp(-2np_n\varepsilon^2)$.

Resultats

Théorème (Déviation absolue pour la validation croisée symétrique)

Supposons $\mathcal{H}$. Supposons aussi que ϕ_n minimise le risque empirique. Mais au lieu de minimiser $\widehat{R}_n(\phi)$, nous supposons que ϕ_n minimise $\frac{1}{n}\sum_{i=1}^n C(h(Y_i, \phi(X_i))$. Par simplicité, nous supposons que l'infimum est atteint i.e. $\phi_n = \arg\min_{\phi\in\mathcal{C}} \frac{1}{n}\sum_{i=1}^n C(h(Y_i, \phi(X_i))$. Alors, nous avons pour tout $\varepsilon > 0$,

$$\Pr(\widetilde{R}_n(\Phi_n^B) - \hat{R}_{CV}^{Out} \geq \varepsilon) \leq \min(B_{ERM}(n, p_n, \varepsilon), V_{ERM}(n, p_n, \varepsilon)) < 1$$

avec

- $B_{ERM}(n, p_n, \varepsilon) =$
  $\min((2np_n + 1)^{4V_{\mathcal{C}}/p_n} \exp(-n\varepsilon^2), (2n(1-p_n)+1)^{\frac{4V_{\mathcal{C}}}{1-p_n}} \exp(-n\varepsilon^2/9))$
- $V_{ERM}(n, p_n, \varepsilon) = \exp(-2np_n\varepsilon^2)$.

Demeure la question : en pratique, comment choisir p_n?
Supposons que l'utilisateur final souhaite un degré de confiance η.
Dans ce contexte, définissons :

- si $\delta \geq \delta_n$

$$f(n, p_n, \delta) := \sqrt{\frac{\ln(1/\delta)}{2np_n}}$$

- et si $\delta < \delta_n$,

$$f(n, p_n, \delta) := 3\sqrt{\frac{4V_{\mathcal{C}} \ln(2n(1-p_n)+1)/(1-p_n) + \ln(1/\delta)}{n}}$$

avec $\delta_n := (2n(1-p_n)+1)^{-\frac{4p_n V_{\mathcal{C}}}{(1-p_n)(1/9-2p_n)}}$.

Définissons $\delta_{n,k}$ par $f(n, k/n, \delta_{n,k}) = \eta$
Notons

$$k_n^\star := \arg \min_{k \in \{1 \dots n-1\}} \hat{R}_{CV}^{Out}(k/n) + f(n, k/n, \delta_{n,k})$$

et

$$p_n^\star := k_n^\star / n.$$

Bibliographie

S. B. Holden
Cross-validation and the PAC learning model.
Research Note RN/96/64, Dept. of CS, Univ. College, London, 1996.

S. B. Holden
PAC-like upper bounds for the sample complexity of leave-one-out cross validation.
In Proceedings of the Ninth Annual ACM Workshop on Computational Learning Theory, pages 41 50, 1996.

M. Kearns and D. Ron.
Algorithmic stability and sanity-check bounds for leave-one-out cross-validation.
Neural Computation, 11:1427 1453, 1999.

M. Kearns and D. Ron.
A bound on the error of cross validation, with consequences for the training-test split.
In Advances in Neural Information Processing Systems 8. The MIT Press.

Bibliographie

M. J. Kearns, Y. Mansour, A. Ng, , and D. Ron.
An experimental and theoretical comparison of model selection methods.
In Proceedings of the Eighth Annual ACM Workshop on Computational Learning Theory, pages 21 30, 1995. To Appear in Machine Learning, COLT95 Special Issue.

S. Kutin.
Extensions to McDiarmid's inequality when differences are bounded with high probability.
Technical report, Department of Computer Science, The University of Chicago, 2002. In preparation.

S. Kutin and P. Niyogi.
Almost-everywhere algorithmic stability and generalization error.
Technical report TR-2002-03, University of Chicago.

Bibliographie

- L. Breiman. Bagging predictors. *Machine Learning*, 24:2, 123–140,1996.
- P. Buhlmann, and B. Yu. Explaining Bagging. , Vol. 30, No. 4 (Aug., 2002), pp. 927-961 *The Annals of Statistics*,2002.
- A. Buja and W. Stuetzle, “Observations on Bagging”, University of Pennsylvania and University of Washington, Seattle,2002.
- Y. Grandvalet. Bagging Equalizes Influence. *Machine Learning*. 55,3: 251 – 270.,2004.
- A. Elisseeff, T. Evgeniou, & M. Pontil. Stability of randomized learning algorithms. *Journal of Machine Learning Research*, 6: 55-79,2005.