Fouille de Données Massives
TD - Généralisaton

M2 Informatique - SISE (2022-2023)

Guillaume Metzler

Institut de Communication (ICOM)
Université de Lyon, Université Lumière Lyon 2
Laboratoire ERIC UR 3083, Lyon, France

guillaume.metzler@univ-lyon2.fr

Abstract

Ce TD a pour but de vous faire découvrir comment il est possible de construire des bornes en généralisation pour des algorithmes de Machine Learning. Il existe de multiples méthodes pour construire ce type de borne mais nous nous concentrons sur une méthode basée sur la notion de *Stabilité uniforme* [Bousquet and Elisseeff, 2002]. On commence par rappeler quelques définitions et résultats nécessaires pour établir ce type de bornes. Ce TD vous proposera ensuite d'établir, pas à pas, une borne généralisation dans un contexte général et il vous sera ensuite demander d'appliquer les résultats obtenus à un algorithme en particulier : les *Séparateurs à Vaste Marge (SVM)*. En outre, nous resterons dans le cas de classification binaire dans le cadre de ce TD.

1 Introduction

On rappelle que l'objectif d'un algorithme $\mathcal{A}$ est de déterminer une hypothèse h à partir d'un ensemble de données S issu d'une distribution inconnue $\mathcal{D}$, qui (i) minimise le risque empirique, *i.e.* l'erreur sur le jeu d'entraînement à l'aide d'une fonction de loss ℓ et (ii) est capable de généralise sur de nouvelles données issues de $\mathcal{D}$.

Notre objectif consiste à borner l'écart entre le *risque réel* $\mathcal{R}^\ell$ et le risque empirique, *i.e.* l'erreur évaluée sur les données d'entraînement $\mathcal{R}^\ell_S$ avec une certaine probabilité $1-\delta$:

$$|\mathcal{R}^\ell - \mathcal{R}^\ell_S| \leq \varepsilon\,(\delta, m)\,,$$

où ε est une fonction qui sera décroissante en le nombre d'exemples m et qui dépend aussi du niveau de confiance que l'on aura en notre borne. Il s'agit également d'*une fonction de* δ. On verra dans la suite que cette fonction dépendra aussi des spécificités de notre algorithme.

Avant de se lancer dans le vif du sujet, nous commençons par introduire quelques notations et par faire quelques rappels élémentaires qui pourront nous servir tout au long du TD.

1.1 Notations

Dans la suite, nous considérerons un jeu de données $S = \{(\mathbf{x}_i, y_i)\}_{i=1}^m$ issu d'une distribution $\mathcal{D} = \mathcal{X} \times \mathcal{Y}$ où $\mathcal{X}$ représente la distribution de l'espace des features et $\mathcal{Y}$ représente la distribution de l'espace des étiquettes. Dans la suite, on supposera que l'espace des features est un sous espace de $\mathbb{R}^d$ et que l'espace des étiquettes sera $\{-1, +1\}$.

On notera h une hypothèse, le plus souvent un classifieur, retournée par un algorithme d'apprentissage $\mathcal{A}$. On désignera par ℓ une fonction de loss qui pénalise les erreurs effectuée par l'hypothèse h, *i.e.*

$$\begin{aligned} \ell : \quad & \mathcal{X} \times \mathcal{Y} \to \mathbb{R}, \\ & (\mathbf{x}, y) \mapsto \ell(h(\mathbf{x}), y). \end{aligned}$$

Etant donné un ensemble de données

$$S = \{(\mathbf{x}_1, y_1), ..., (\mathbf{x}_{i-1}, y_{i-1}), (\mathbf{x}_i, y_i), (\mathbf{x}_{i+1}, y_{i+1}), ..., (\mathbf{x}_m, y_m)\},$$

on désignera par S^i le même ensemble d'entraînement pour lequel on aura remplacer le i-ème exemple de S par un autre exemple issu de $\mathcal{D}$, *i.e.*

$$S^i = \{(\mathbf{x}_1, y_1), ..., (\mathbf{x}_{i-1}, y_{i-1}), (\mathbf{x}'_i, y'_i), (\mathbf{x}_{i+1}, y_{i+1}), ..., (\mathbf{x}_m, y_m)\},$$

et par par S^{-i} le même ensemble d'entraînement pour lequel on aura retirer le i-ème exemple de S, *i.e.*

$$S^{-i} = \{(\mathbf{x}_1, y_1), ..., (\mathbf{x}_{i-1}, y_{i-1}), (\mathbf{x}_{i+1}, y_{i+1}), ..., (\mathbf{x}_m, y_m)\},$$

Pour des raisons pratiques, nous supposerons que toutes nos données se trouvent dans un espace borné, *i.e.* $\forall \mathbf{x} \in \mathbb{R}^d,\ \|\mathbf{x}\|_2 \leq B$ où $B > 0$.

1.2 Quelques définitions

Dans cette section, on rappelle brièvement ce que l'on appelle le risque empirique et le risque réel, quelques définitions utiles sur les fonctions ainsi que quelques résultats en probabilité. N'hésitez pas à vous reporter à ces résultats à tout moment pour les besoins du TD.

Définition 1.1: Risque empirique]

Soit $S = \{(\mathbf{x}_i, y_i)\}$ un ensemble de m exemples d'entraînement tirés identiquement et indépendamment selon une distribution inconnue $\mathcal{D}$ et soit $\ell : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$. Le risque empirique $\mathcal{R}_S^\ell$ d'une hypothèse h est défini par

$$\mathcal{R}_S^\ell(h) = \mathop{\mathbb{E}}_{(\mathbf{x},y)\in S}[\ell(h(\mathbf{x}), y))] = \frac{1}{m}\sum_{i=1}^{m} \ell(h(\mathbf{x}_i), y_i)).$$

Définition 1.2: Risque réel

Soit $\ell : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$ notre loss et $\mathcal{D}$ la distribution inconnue de nos données. Le risque réel $\mathcal{R}^\ell$ d'une hypothèse h est défini par :

$$\mathcal{R}^\ell = \mathop{\mathbb{E}}_{(\mathbf{x},y)\sim\mathcal{D}}[\ell(h(\mathbf{x}), y)].$$

Définition 1.3: Convexité

Une fonction $f : \mathbb{R}^d \to \mathbb{R}$ est dite convexe si pour tout $\mathbf{x}, \mathbf{x}' \in \mathbb{R}^d$ et pour tout $t \in [0, 1]$, nous avons

$$f(t\mathbf{x} + (1-t)\mathbf{x}') \leq tf(\mathbf{x}) + (1-t)f(\mathbf{x}').$$

Définition 1.4: Lipschitzienne

Une fonction $f : \mathbb{R}^d \to \mathbb{R}$ est dite lipschitzienne s'il existe une constante $C > 0$ telle que pour tout $\mathbf{x}, \mathbf{x}' \in \mathbb{R}^d$ nous avons

$$\|f(\mathbf{x}) - f(\mathbf{x}')\| \leq C\|\mathbf{x} - \mathbf{x}'\|.$$

Théorème 1.1: Inégalité de McDiarmid

Soient deux ensembles S et S^i et $F : (\mathcal{X} \times \mathcal{Y})^m \to \mathbb{R}$ une fonction mesurable (on utilisera le fait que c'est une fonction "classique") pour laquelle il existe des constantes c_i telles que :

$$|F(S) - F(S^i)| \leq c_i,$$

alors

$$\mathbb{P}[(F(S) - \mathbb{E}_S[F(S)]) \geq \varepsilon] \leq \exp\left(\frac{-2\varepsilon^2}{\sum_{i=1}^m c_i^2}\right).$$

Nous pouvons maintenant entrer dans le vif du sujet et parler de bornes en généralisations.

2 Stabilité Uniforme

La notion de stabilité uniforme [Bousquet and Elisseeff, 2002], contrairement à d'autres résultats fournissant des garanties théoriques en généralisation, ne repose pas sur une mesure de complexité de l'espace d'hypothèse [Koltchinskii and Panchenko, 2000], [Vapnik and Chervonenkis, 1971]. Ces bornes reposent sur le concept de *stabilité*. Pour énoncer les choses simplement, on dira qu'un algorithme est stable si sa résultante est peu sensible à une variation dans l'ensemble d'entraînement. Ici, nous nous intéresserons plus précisément à la notion de *stabilité uniforme* : on va regarder quelle sera la plus grande "modification" en terme de valeurs de la loss que l'on peut observer si l'on perturbe très faiblement (*i.e.* si on modifie un exemple) notre ensemble d'entraînement. Plus formellement :

Définition 2.1: Stabilité Uniforme

Un algorithme d'apprentissage $\mathcal{A}$ est uniformément stable avec une constante de stabilité uniforme égale à $\beta > 0$ vis à vis d'une loss ℓ et d'un paramètre θ si :

$$\forall S,\ \forall i,\ 1 \leq i \leq m, \quad \sup_{\mathbf{x}} |\ell(\theta_S, \mathbf{x}) - \ell(\theta_{S^i})| \leq \beta,$$

où S est notre ensemble d'apprentissage de taille m, θ_S sont les paramètres du modèle appris avec S et θ_{S^i} sont les paramètres du modèle appris avec S^i.

Le paramètre θ serait par exemple le paramètre définissant l'hyper-plan séparateur de notre SVM et nous verrons par la suite que β est une constante qui va dépendre de la valeur de m.

La constante β jouera un rôle important dans notre cas, c'est en effet cette constante qui va contenir toutes les propriétés de notre loss ℓ mais aussi du terme de régularisation de notre problème d'optimisation. En utilisant la *convexité* de la fonction de loss ℓ et l'inégalité de McDiarmid présentée plus tôt, on a le résultat suivant :

Théorème 2.1: Bornes via Stabilité Uniforme

Soit $\delta > 0$ and $m > 1$. Pour tout algorithme avec une constante de stabilité uniforme égale à β utilisant une loss ℓ bornée par K, nous avons, avec probabilité au moins $1 - \delta$:

$$\mathcal{R}^{\ell}(\theta_S) \leq \mathcal{R}^{\ell}_S(\theta_S) + 2\beta + (4m\beta + K)\sqrt{\frac{\ln(1/\delta)}{2m}},$$

où $\mathcal{R}^{\ell}(\cdot)$ représente le risque réel et $\mathcal{R}^{\ell}_S(\cdot)$ sa version empirique.

Notre objectif sera de démontrer ce résultat et de déterminer les valeurs précises des différentes constantes pour un algorithme de type SVM.

3 Bornes en généralisation pour un SVM en classification

La preuve se déroule en plusieurs étapes :

- On commence par réécrire notre loss sous une forme adéquate et on montre certaines propriétés relatives à cette loss comme la *convexité*, *σ-admissibilité*.
- La deuxième étape plus technique consistera à montrer que notre fonction est bien uniformément stable.

- Il nous restera ensuite à conclure à l'aide du Théorème 1.1 pour obtenir notre borne.

Commençons d'abord par montrer quelques propriétés sur notre loss.

3.1 Préliminaires

On rappelle que le problème d'optimisation d'un SVM s'écrit :

$$\begin{array}{ll} \min\limits_{\xi\in\mathbb{R}^m,\ (\mathbf{w},b)\in\mathbb{R}^{d+1}} & \frac{1}{2}\|\mathbf{w}\|^2 + \frac{C}{m}\sum_{i=1}^m \xi_i \\ s.t. & y_i(\langle \mathbf{w}, \mathbf{x}_i\rangle + b) \geq 1-\xi_i, \quad \forall\, i = 1, ..., m, \\ & \xi_i \geq 0, \qquad\qquad\qquad\quad \forall\, i = 1, ..., m. \end{array}$$

ou encore, si l'on considère la version *hinge-loss* :

$$\min_{(\mathbf{w},b)\in\mathbb{R}^{d+1}} \frac{1}{2}\|\mathbf{w}\|^2 + \frac{C}{m}\sum_{i=1}^m [1 - y_i(\langle \mathbf{w}, \mathbf{x}_i\rangle + b)]_+.$$

C : paramètre de régularisation qui permet de contrôler l'importance que l'on accorde aux poids du modèle $\mathbf{w}$(*i.e.* sa complexité) par rapport aux **erreurs** effectués par le modèle . En outre, $[f(\mathbf{x})]_+ = \max(0, f(\mathbf{x}))$
C'est cette deuxième version de formulation du problème qui va nous intéresser. En outre pour simplifier les notations dans la suite, on supposera que l'on a $\mathbf{w} = (\mathbf{w}, b)$.

Remarque : Dans le cadre de l'étude d'un SVM, l'hypothèse h étant définie uniquement par le paramètre de l'hyperplan séparateur $\mathbf{w}$, on fera souvent l'abus de notation $h \simeq \mathbf{w}$.

Question 1. Dans le problème d'optimisation précédent, identifiez la loss ℓ ainsi que le terme de régularisation N et écrivez le problème d'optimisation sous la forme

$$\frac{1}{m}\sum_{i=1}^m \ell(h(\mathbf{x}), y)) + \lambda N(h),$$

ou on aura posé $\lambda = 1/2C$.

Question 2. Montrez que le problème d'optimisation ainsi défini est bien un problème d'optimisation convexe.

Dans la suite nous adopterons la notation suivante

$$F_S(\mathbf{w}) = \mathcal{R}_S(\mathbf{w}) + \lambda N(\mathbf{w}).$$

Question 3. Montrez que si $\mathbf{w}^\star$ est solution du problème d'optimisation, en particulier nous avons :

$$\|\mathbf{w}^\star\| \leq \sqrt{\frac{1}{\lambda}},$$

on pourra par exemple estimer la valeur de notre problème d'optimisation lorsque $\mathbf{w} = 0$.

Question 4. En déduire qu'il existe une constante K telle que pour tout $\mathbf{x} \in \mathcal{X}$ nous avons :

$$\ell(h(\mathbf{x}), y)) \leq K.$$

Nous sommes maintenant prêt à montrer que notre fonction est σ-admissible.

Définition 3.1: σ-admissibilité

Une loss ℓ définie est dite σ-admissible si la fonction $\ell(\mathbf{w}, \mathbf{z})$, où $\mathbf{z} = (\mathbf{x}, y)$, est convexe par rapport à son premier argument et si la condition suivante est vérifiée :

$$\forall \mathbf{z} \in \mathcal{D}, \quad |\ell(\mathbf{w}_1, \mathbf{z}') - \ell(\mathbf{w}_2, \mathbf{z}')| \leq \sigma \|\mathbf{w}_1 - \mathbf{w}_2\|,$$

où $\mathcal{D} = \{\mathbf{z} : \exists h \in \mathcal{H},\ \exists \mathbf{x} \in \mathcal{X} : sign(h(\mathbf{x})) = y\}$. On notera en particulier que l'espace $\mathcal{H}$ est l'espace des valeurs de $\mathbf{w} \in \mathbb{R}^d$.
Autrement dit, ℓ est σ-admissible si elle est convexe et lispchitzienne par rapport à son premier argument.

Question 5. En reprenant ce qui a été fait à la question 2, montrez que la loss ℓ est σ-admissible et déterminez la valeur de σ.

3.2 Stabilité uniforme

La section précédente a permis de fournir quelques résultats préliminaires qui vont nous être utile dans cette section pour démontrer que notre algorithme est uniformément stable. Pour cela nous devons commencer par montrer le lemme technique suivant, nous l'appliquerons ensuite dans un cas particulier pour en déduire la valeur de la constante de stabilité uniforme β.

Lemme 3.1: Un résultat intermédiaire

Soit ℓ une fonction σ-admissible vis-à-vis de l'ensemble $\mathcal{H}$ et notons N le terme de régularisation de notre problème d'optimisation définie sur $\mathcal{H}$. On supposera que, pour tout ensemble d'apprentissage S et S^i, notre problème d'optimisation admet bien un minimum en $\mathbf{w}$ et $\mathbf{w}^i$ respectivement égaux $F_S(\mathbf{w})$ et $F_{S^i}(\mathbf{w}^i)$. Alors pour tout $t \in [0,1]$ nous avons :

$$N(\mathbf{w}) - N(\mathbf{w} + t\Delta\mathbf{w}) + N(\mathbf{w}^i) - N(\mathbf{w}^i - t\Delta\mathbf{w}) \leq \frac{2t\sigma}{\lambda m}\|\Delta\mathbf{w}\|,$$

où $\Delta\mathbf{w} = \mathbf{w}^i - \mathbf{w}$.

Question 6. Notre premier objectif est démontrer ce lemme, pour cela on procèdera en plusieurs étapes en gardant à l'esprit la convexité de notre problème d'optimisation :

(i) En utilisant la convexité du risques empirique $\mathcal{R}_{S^i}$, montrez que :

$$\mathcal{R}_{S^i}(\mathbf{w} + t\Delta\mathbf{w}) - \mathcal{R}_{S^i}(\mathbf{w}) + \mathcal{R}_{S^i}(\mathbf{w}^i - t\Delta\mathbf{w}) - \mathcal{R}_{S^i}(\mathbf{w}^i) \leq 0$$

(ii) Expliquez pour les inégalités suivantes sont vraies :

$$F_S(\mathbf{w}) - F_S(\mathbf{w} + t\Delta\mathbf{w}) \leq 0,$$

$$F_{S^i}(\mathbf{w}^i) - F_S(\mathbf{w}^i - t\Delta\mathbf{w}) \leq 0.$$

(iii) En utilisant les deux questions précédentes , montrez que l'on a

$$\begin{aligned}\mathcal{R}_S(\mathbf{w}) - \mathcal{R}_S(\mathbf{w} + t\Delta\mathbf{w}) - \mathcal{R}_{S^i}(\mathbf{w}) + \mathcal{R}_{S^i}(\mathbf{w} + t\Delta\mathbf{w}) \\ + \lambda[N(\mathbf{w}) + N(\mathbf{w}^i) - N(\mathbf{w} + t\Delta\mathbf{w}) - N(\mathbf{w}^i - t\Delta\mathbf{w})] \leq 0\end{aligned}$$

(iv) Notons

$$R = \mathcal{R}_S(\mathbf{w}) - \mathcal{R}_{S^i}(\mathbf{w}) - \mathcal{R}_S(\mathbf{w} + t\Delta\mathbf{w}) + \mathcal{R}_{S^i}(\mathbf{w} + t\Delta\mathbf{w}).$$

Simplifiez cette expression et montrez, en utilisant le fait que ℓ est σ-admissible, que l'on a le résultat suivant

$$|R| \leq \frac{2t\sigma}{m}\|\Delta\mathbf{w}\|$$

(v) Conclure la démonstration du lemme.

Question 7. On va maintenant déterminer la constante de stabilité uniforme β de notre algorithme telle que définie à la Section 2.

(i) En posant $t = \frac{1}{2}$, montrez que l'on a :

$$N(\mathbf{w}) + N(\mathbf{w}^i) - N\left(\mathbf{w} + \frac{1}{2}\Delta\mathbf{w}\right) - N\left(\mathbf{w}^i - \frac{1}{2}\Delta\mathbf{w}\right) = \frac{1}{2}\|\Delta\mathbf{w}\|^2,$$

(ii) En déduire la valeur de la constante de stabilité uniforme β, i.e. montrez que l'on a

$$\beta = \frac{2\sigma^2}{\lambda m}$$

Question 8. Appliquez le Théorème 2.1 dans le contexte présent d'un SVM et donnez l'expression d'une borne en généralisation d'un SVM.

Question 9. Interpréter la borne obtenue. Que peut-on dire de la valeur de la borne en fonction de n ? En fonction de l'hyper-paramètre λ ? Est-ce que cela vous semble cohérent ?

On se propose, dans la section suivante de démontrer ce théorème afin de compléter notre preuve.

3.3 Démonstration du Théorème 2.1

L'objectif de cette section est démontrer le Théorème 2.1, à nouveau, nous allons décomposer cette démonstration en plusieurs étapes, donc en plusieurs questions.
Dans la suite, on supposera que les hypothèses du Théorème 2.1 sont vérifiées.

Question 10. Expliquez pourquoi l'inégalité suivante est vraie :

$$|\mathcal{R}(\theta_S) - \mathcal{R}(\theta_S^{-1})| \leq \mathbb{E}_z[|\ell(\theta_S, \mathbf{x}) - \ell(\theta_{S^{-i}}, \mathbf{z})| \leq \beta$$

Question 11. Montrez que l'on a

$$|\mathcal{R}_S(\theta_S) - \mathcal{R}_{S^{-i}}(\theta_{S^{-i}})| \leq \beta + \frac{K}{m}.$$

Question 12. En déduire une borne sur $|\mathcal{R}(\theta_S) - \mathcal{R}(\theta_{S^i})|$ et sur $|\mathcal{R}_S(\theta_S) - \mathcal{R}_{S^i}(\theta_{S^i})|$.

Question 13. Si ce n'est pas le résultat obtenu à la question précédente, montrez que l'on peut avoir une borne plus fine sur $|\mathcal{R}_S(\theta_S) - \mathcal{R}_{S^i}(\theta_{S^i})|$, *i.e.*

$$|\mathcal{R}_S(\theta_S) - \mathcal{R}_{S^i}(\theta_{S^i})| \leq 2\beta + \frac{K}{m}.$$

Question 14. En déduire que la variable aléatoire $\mathcal{R}(\theta_S) - \mathcal{R}_S(\theta_S)$ satisfait les conditions du Théorème 1 et en déduire la valeur de la constante c_i pour tout i.

Question 15. Appliquez l'inégalité de Mc Diarmid en utilisant le fait que $\mathbb{E}_S[\mathcal{R}(\theta_S) - \mathcal{R}_S(\theta_S)] \leq 2\beta$ et en déduire la borne du Théorème 2.1.

4 Illustration du résultat

L'objectif de cette section est d'illustrer le résultat théorique obtenu précédemment en réalisant des petites simulations sur votre ordinateur.

Pour cela vous allez procéder à des tests en utilisant un classifieur de type SVM afin de mettre en évidence le résultat obtenu. Vous choisirez des jeux de données disponibles en ligne (sur Kaggle, Keel ou UCI) ou jeu de données simulées en 2D.

Une fois que ce protocole est écrit, vous représenterez plusieurs courbes sur un même graphe et en fonction du nombre d'exemples :

- la différence entre l'erreur en entraînement et l'erreur en test, on prendra soin de fixer le jeu de données test
- la valeur de votre borne, *i.e.* le membre de droite de l'inégalité présentée dans le Théorème 2.1

5 Pour aller plus loin

Il nous restera un dernier résultat à démontrer, il s'agit du Théorème 1.1. Votre défi, essayer de démontrer ce Théorème. Vous pourrez commencer par montrer le résultat suivant :

Lemme 5.1: Une transformée de Laplace

Soit X une variable aléatoire réelle, centrée et presque sûrement bornée par 1. On note L_X sa transformée de Laplace, *i.e.* $L_X(t) = \mathbb{E}[\exp(tX)]$, pour tout $t \in \mathbb{R}$. Alors, pour tout $t \in \mathbb{R}$, nous avons :

$$L_X(t) \leq \exp\left(\frac{t^2}{2}\right).$$

References

[Bousquet and Elisseeff, 2002] Bousquet, O. and Elisseeff, A. (2002). Stability and generalization. *Journal of Machine Learning Research*, 2:499–526.

[Koltchinskii and Panchenko, 2000] Koltchinskii, V. and Panchenko, D. (2000). Rademacher processes and bounding the risk of function learning. In Giné, E., Mason, D. M., and Wellner, J. A., editors, *High Dimensional Probability II*, pages 443–457. Birkhäuser Boston.

[Vapnik and Chervonenkis, 1971] Vapnik, V. and Chervonenkis, A. (1971). On the uniform convergence of relative frequencies of events to their probabilities. *Theory of Probability & Its Applications*, 16(2):264–280.