

Pense-bête VIP : Apprentissage supervisé

Afshine AMIDI et Shervine AMIDI

6 octobre 2018

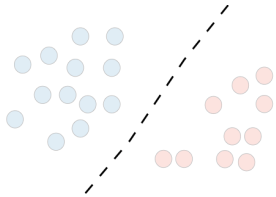
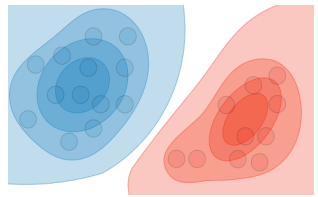
Introduction à l'apprentissage supervisé

Étant donné un ensemble de points $\{x^{(1)}, \dots, x^{(m)}\}$ associés à un ensemble d'issues $\{y^{(1)}, \dots, y^{(m)}\}$, on veut construire un classifieur qui apprend à prédire y depuis x .

□ **Type de prédiction** – Les différents types de modèle prédictifs sont résumés dans le tableau ci-dessous :

	Régression	Classifieur
Issue	Continu	Classe
Exemples	Régression linéaire	Régression logistique, SVM, Naive Bayes

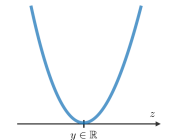
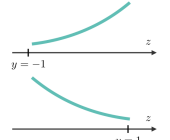
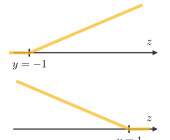
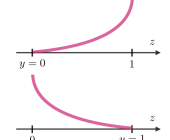
□ **Type de modèle** – Les différents modèles sont présentés dans le tableau ci-dessous :

	Modèle discriminatif	Modèle génératif
But	Estimer directement $P(y x)$	Estimer $P(x y)$ puis déduire $P(y x)$
Ce qui est appris	Frontière de décision	Distribution de proba des données
Illustration		
Exemples	Régressions, SVMs	GDA, Naive Bayes

Notations et concepts généraux

□ **Hypothèse** – Une hypothèse est notée h_θ et est le modèle que l'on choisit. Pour une entrée donnée $x^{(i)}$, la prédiction donnée par le modèle est $h_\theta(x^{(i)})$.

□ **Fonction de loss** – Une fonction de loss est une fonction $L : (z, y) \in \mathbb{R} \times Y \mapsto L(z, y) \in \mathbb{R}$ prenant comme entrée une valeur prédite z correspondant à une valeur réelle y , et nous renseigne sur la ressemblance de ces deux valeurs. Les fonctions de loss courantes sont récapitulées dans le tableau ci-dessous :

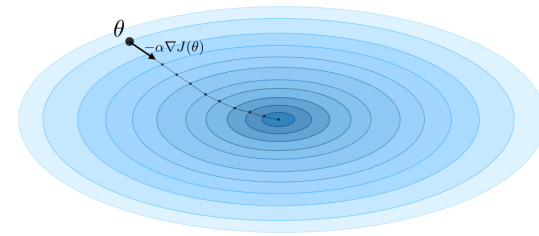
Moindres carrés	Logistique	Hinge loss	Cross-entropie
$\frac{1}{2}(y - z)^2$	$\log(1 + \exp(-yz))$	$\max(0, 1 - yz)$	$-\left[y \log(z) + (1 - y) \log(1 - z)\right]$
			
Régression linéaire	Régression logistique	SVM	Réseau de neurones

□ **Fonction de coût** – La fonction de coût J est communément utilisée pour évaluer la performance d'un modèle, et est définie avec la fonction de loss L par :

$$J(\theta) = \sum_{i=1}^m L(h_\theta(x^{(i)}), y^{(i)})$$

□ **Algorithme du gradient** – En notant $\alpha \in \mathbb{R}$ le taux d'apprentissage (en anglais *learning rate*), la règle de mise à jour de l'algorithme est exprimée en fonction du taux d'apprentissage et de la fonction de cost J de la manière suivante :

$$\theta \leftarrow \theta - \alpha \nabla J(\theta)$$



Remarque : L'algorithme du gradient stochastique (en anglais SGD - Stochastic Gradient Descent) met à jour le paramètre à partir de chaque élément du jeu d'entraînement, tandis que l'algorithme du gradient de batch le fait sur chaque lot d'exemples.

□ **Vraisemblance** – La vraisemblance d'un modèle $L(\theta)$ de paramètre θ est utilisée pour trouver le paramètre optimal θ par le biais du maximum de vraisemblance. En pratique, on utilise la log vraisemblance $\ell(\theta) = \log(L(\theta))$ qui est plus facile à optimiser. On a :

$$\theta^{\text{opt}} = \arg \max_{\theta} L(\theta)$$

□ **Algorithme de Newton** – L'algorithme de Newton est une méthode numérique qui trouve θ tel que $\ell'(\theta) = 0$. La règle de mise à jour est :

$$\theta \leftarrow \theta - \frac{\ell'(\theta)}{\ell''(\theta)}$$

Remarque : la généralisation multidimensionnelle, aussi connue sous le nom de la méthode de Newton-Raphson, a la règle de mise à jour suivante :

$$\theta \leftarrow \theta - \left(\nabla_{\theta}^2 \ell(\theta) \right)^{-1} \nabla_{\theta} \ell(\theta)$$

Régression linéaire

On suppose ici que $y|x; \theta \sim \mathcal{N}(\mu, \sigma^2)$

□ **Équations normales** – En notant X la matrice de design, la valeur de θ qui minimize la fonction de cost a une solution de forme fermée tel que :

$$\theta = (X^T X)^{-1} X^T y$$

□ **Algorithme LMS** – En notant α le taux d'apprentissage, la règle de mise à jour d'algorithme des moindres carrés (LMS) pour un jeu de données d'entraînement de m points, aussi connu sous le nom de règle de Widrow-Hoff, est donné par :

$$\forall j, \quad \theta_j \leftarrow \theta_j + \alpha \sum_{i=1}^m [y^{(i)} - h_{\theta}(x^{(i)})] x_j^{(i)}$$

Remarque : la règle de mise à jour est un cas particulier de l'algorithme du gradient.

□ **LWR** – Locally Weighted Regression, souvent noté LWR, est une variante de la régression linéaire appliquant un coefficient à chaque exemple dans sa fonction de coût via $w^{(i)}(x)$, qui est défini avec un paramètre $\tau \in \mathbb{R}$ de la manière suivante :

$$w^{(i)}(x) = \exp \left(- \frac{(x^{(i)} - x)^2}{2\tau^2} \right)$$

Classification et régression logistique

□ **Sigmoïde** – La sigmoïde g , aussi connue sous le nom de fonction logistique, est définie par :

$$\forall z \in \mathbb{R}, \quad g(z) = \frac{1}{1 + e^{-z}} \in]0, 1[$$

□ **Régression logistique** – On suppose ici que $y|x; \theta \sim \text{Bernoulli}(\phi)$. On a la forme suivante :

$$\phi = p(y = 1|x; \theta) = \frac{1}{1 + \exp(-\theta^T x)} = g(\theta^T x)$$

Remarque : il n'y a pas de solution fermée dans le cas de la régression logistique.

□ **Régression softmax** – Une régression softmax, aussi appelée un régression logistique multi-classe, est utilisée pour généraliser la régression logistique lorsqu'il y a plus de 2 classes à prédire. Par convention, on fixe $\theta_K = 0$, ce qui oblige le paramètre de Bernoulli ϕ_i de chaque classe i à être égal à :

$$\phi_i = \frac{\exp(\theta_i^T x)}{\sum_{j=1}^K \exp(\theta_j^T x)}$$

Modèles linéaires généralisés

□ **Famille exponentielle** – Une classe de distributions est issue de la famille exponentielle lorsqu'elle peut être écrite en termes d'un paramètre naturel, aussi appelé paramètre canonique ou fonction de lien η , d'une statistique suffisante $T(y)$ et d'une fonction de log-partition $a(\eta)$ de la manière suivante :

$$p(y; \eta) = b(y) \exp(\eta T(y) - a(\eta))$$

Remarque : on aura souvent $T(y) = y$. Aussi, $\exp(-a(\eta))$ peut être vu comme un paramètre de normalisation s'assurant que les probabilités somment à un.

Les distributions exponentielles les plus communément rencontrées sont récapitulées dans le tableau ci-dessous :

Distribution	η	$T(y)$	$a(\eta)$	$b(y)$
Bernoulli	$\log \left(\frac{\phi}{1-\phi} \right)$	y	$\log(1 + \exp(\eta))$	1
Gaussian	μ	y	$\frac{\eta^2}{2}$	$\frac{1}{\sqrt{2\pi}} \exp \left(-\frac{y^2}{2} \right)$
Poisson	$\log(\lambda)$	y	e^{η}	$\frac{1}{y!}$
Geometric	$\log(1 - \phi)$	y	$\log \left(\frac{e^{\eta}}{1 - e^{\eta}} \right)$	1

□ **Hypothèses pour les GLMs** – Les modèles linéaires généralisés (GLM) ont pour but de prédire une variable aléatoire y comme une fonction de $x \in \mathbb{R}^{n+1}$ et reposent sur les 3 hypothèses suivantes :

$$(1) \quad y|x; \theta \sim \text{ExpFamily}(\eta) \quad (2) \quad h_{\theta}(x) = E[y|x; \theta] \quad (3) \quad \eta = \theta^T x$$

Remarque : la méthode des moindres carrés ordinaires et la régression logistique sont des cas spéciaux des modèles linéaires généralisés.

Support Vector Machines

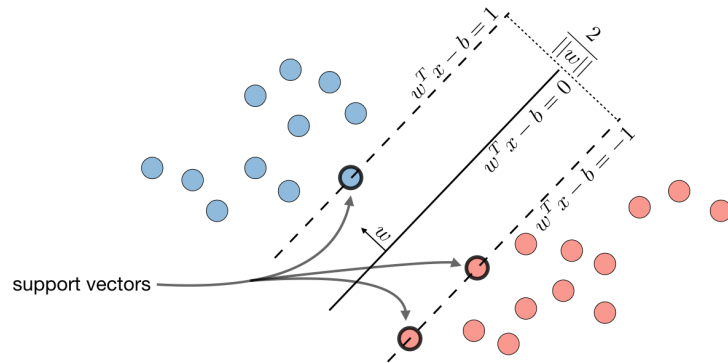
Le but des support vector machines est de trouver la ligne qui maximise la distance minimum à la ligne.

□ **Classifieur à marges optimales** – Le classifieur à marges optimales h est tel que :

$$h(x) = \text{sign}(w^T x - b)$$

où $(w, b) \in \mathbb{R}^n \times \mathbb{R}$ est une solution du problème d'optimisation suivant :

$$\min \frac{1}{2} \|w\|^2 \quad \text{tel que} \quad y^{(i)} (w^T x^{(i)} - b) \geq 1$$



Remarque : la ligne est définie par $w^T x - b = 0$.

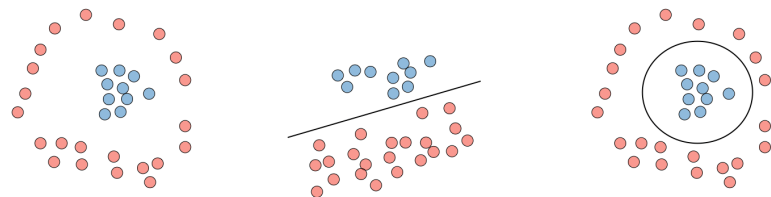
□ **Hinge loss** – Le hinge loss est utilisé dans le cadre des SVMs et est défini de la manière suivante :

$$L(z, y) = [1 - yz]_+ = \max(0, 1 - yz)$$

□ **Noyau** – Étant donné un feature mapping ϕ , on définit le noyau K par :

$$K(x, z) = \phi(x)^T \phi(z)$$

En pratique, le noyau K défini par $K(x, z) = \exp\left(-\frac{\|x - z\|^2}{2\sigma^2}\right)$ est nommé noyau gaussien et est communément utilisé.



Séparation non linéaire → Mapping de noyaux ϕ → Ligne de décision dans l'espace initial

Remarque : on dit que l'on utilise "l'astuce du noyau" (en anglais *kernel trick*) pour calculer la fonction de coût en utilisant le noyau parce qu'il se trouve que l'on n'a pas besoin de trouver le mapping explicite, qui est souvent compliqué. Il suffit de connaître les valeurs de $K(x, z)$.

□ **Lagrangien** – On définit le lagrangien $L(w, b)$ par :

$$\mathcal{L}(w, b) = f(w) + \sum_{i=1}^l \beta_i h_i(w)$$

Remarque : les coefficients β_i sont appelés les multiplicateurs de Lagrange.

Apprentissage génératif

Un modèle génératif essaie d'abord d'apprendre comment les données sont générées en estimant $P(x|y)$, nous permettant ensuite d'estimer $P(y|x)$ par le biais du théorème de Bayes.

Gaussian Discriminant Analysis

□ **Cadre** – Le Gaussian Discriminant Analysis suppose que y et $x|y = 0$ et $x|y = 1$ sont tels que :

$$y \sim \text{Bernoulli}(\phi)$$

$$x|y = 0 \sim \mathcal{N}(\mu_0, \Sigma) \quad \text{et} \quad x|y = 1 \sim \mathcal{N}(\mu_1, \Sigma)$$

□ **Estimation** – Le tableau suivant récapitule les estimations que l'on a trouvées lors de la maximisation de la vraisemblance :

$\hat{\phi}$	$\hat{\mu}_j \quad (j = 0, 1)$	$\hat{\Sigma}$
$\frac{1}{m} \sum_{i=1}^m 1_{\{y^{(i)}=1\}}$	$\frac{\sum_{i=1}^m 1_{\{y^{(i)}=j\}} x^{(i)}}{\sum_{i=1}^m 1_{\{y^{(i)}=j\}}}$	$\frac{1}{m} \sum_{i=1}^m (x^{(i)} - \mu_{y^{(i)}})(x^{(i)} - \mu_{y^{(i)}})^T$

Naive Bayes

□ **Hypothèse** – Le modèle de Naive Bayes suppose que les caractéristiques de chaque point sont toutes indépendantes :

$$P(x|y) = P(x_1, x_2, \dots | y) = P(x_1|y)P(x_2|y)\dots = \prod_{i=1}^n P(x_i|y)$$

□ **Solutions** – Maximiser la log vraisemblance donne les solutions suivantes, où $k \in \{0, 1\}, l \in \llbracket 1, L \rrbracket$

$$P(y = k) = \frac{1}{m} \times \#\{j|y^{(j)} = k\} \quad \text{et} \quad P(x_i = l|y = k) = \frac{\#\{j|y^{(j)} = k \text{ et } x_i^{(j)} = l\}}{\#\{j|y^{(j)} = k\}}$$

Remarque : Naive Bayes est couramment utilisé pour la classification de texte et pour la détection de spams.

Méthode à base d'arbres et d'ensembles

Ces méthodes peuvent être utilisées pour des problèmes de régression et de classification.

□ **CART** – Les arbres de classification et de régression (en anglais *CART - Classification And Regression Trees*), aussi connus sous le nom d'arbres de décision, peuvent être représentés sous la forme d'arbres binaires. Ils ont l'avantage d'être très interprétables.

□ **Random forest** – C'est une technique à base d'arbres qui utilise un très grand nombre d'arbres de décisions construits à partir d'ensembles de caractéristiques aléatoirement sélectionnés. Contrairement à un simple arbre de décision, il n'est pas interprétable du tout mais le fait qu'il ait une bonne performance en fait un algorithme populaire.

Remarque : les random forests sont un type de méthode ensembliste.

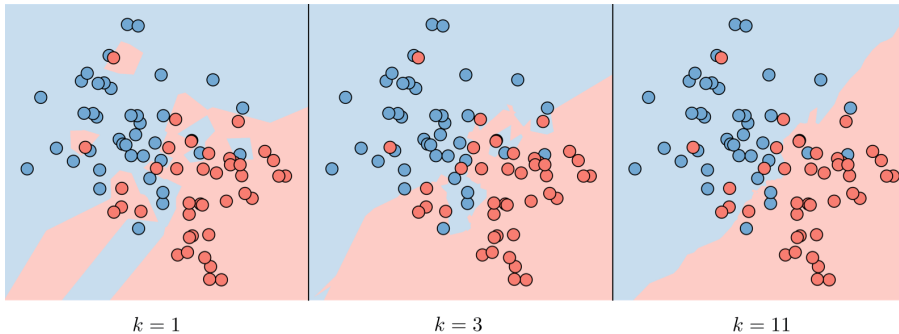
□ **Boosting** – L'idée des méthodes de boosting est de combiner plusieurs modèles faibles pour former un modèle meilleur. Les principales méthodes de boosting sont récapitulées dans le tableau ci-dessous :

Boosting adaptatif	Boosting par gradient
- De grands coefficients sont mis sur les erreurs pour s'améliorer à la prochaine étape de boosting - Connus sous le nom d'Adaboost	- Les modèles faibles sont entraînés sur les erreurs résiduelles

Autres approches non-paramétriques

□ **k-nearest neighbors** – L'algorithme des k plus proches voisins (en anglais *k-nearest neighbors*), aussi connu sous le nom de k -NN, est une approche non-paramétrique où la réponse d'un point est déterminée par la nature de ses k voisins du jeu de données d'entraînement. Il peut être utilisé dans des cadres de classification et de régression.

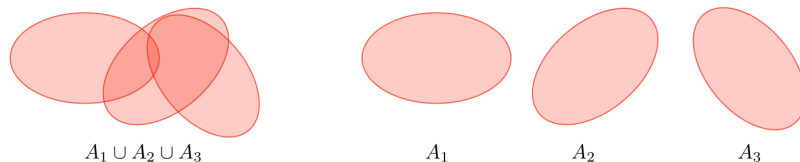
Remarque : Plus le paramètre k est élevé, plus le biais est élevé, et plus le paramètre k est faible, plus la variance est élevée.



Théorie d'apprentissage

□ **Inégalité de Boole** – Soit A_1, \dots, A_k k événements. On a :

$$P(A_1 \cup \dots \cup A_k) \leq P(A_1) + \dots + P(A_k)$$



□ **Inégalité d'Hoeffding** – Soit Z_1, \dots, Z_m m variables iid tirées d'une distribution de Bernoulli de paramètre ϕ . Soit $\hat{\phi}$ leur moyenne empirique et $\gamma > 0$ fixé. On a :

$$P(|\phi - \hat{\phi}| > \gamma) \leq 2 \exp(-2\gamma^2 m)$$

Remarque : cette inégalité est aussi connue sous le nom de borne de Chernoff.

□ **Erreur de training** – Pour un classifieur donné h , on définit l'erreur d'entraînement $\hat{\epsilon}(h)$, aussi connu sous le nom de risque empirique ou d'erreur empirique, par :

$$\hat{\epsilon}(h) = \frac{1}{m} \sum_{i=1}^m 1_{\{h(x^{(i)}) \neq y^{(i)}\}}$$

□ **Probablement Approximativement Correct (PAC)** – PAC est un cadre dans lequel de nombreux résultats d'apprentissages ont été prouvés, et contient l'ensemble d'hypothèses suivant :

- les jeux d'entraînement et de test suivent la même distribution
- les exemples du jeu d'entraînement sont tirés indépendamment

□ **Éclatement** – Étant donné un ensemble $S = \{x^{(1)}, \dots, x^{(d)}\}$, et un ensemble de classifieurs \mathcal{H} , on dit que \mathcal{H} brise S si pour tout ensemble de labels $\{y^{(1)}, \dots, y^{(d)}\}$, on a :

$$\exists h \in \mathcal{H}, \quad \forall i \in [1, d], \quad h(x^{(i)}) = y^{(i)}$$

□ **Théorème de la borne supérieure** – Soit \mathcal{H} une hypothèse finie de classe telle que $|\mathcal{H}| = k$, soit δ , et soit m la taille fixée d'un échantillon. Alors, avec une probabilité d'au moins $1 - \delta$, on a :

$$\epsilon(\hat{h}) \leq \left(\min_{h \in \mathcal{H}} \epsilon(h) \right) + 2 \sqrt{\frac{1}{2m} \log \left(\frac{2k}{\delta} \right)}$$

□ **Dimension VC** – La dimension de Vapnik-Chervonenkis (VC) d'une classe d'hypothèses de classes infinies donnée \mathcal{H} , que l'on note $VC(\mathcal{H})$, est la taille de l'ensemble le plus grand qui est brisé par \mathcal{H} .

Remarque : la dimension VC de $\mathcal{H} = \{\text{set of linear classifiers in 2 dimensions}\}$ est égale à 3.



□ **Théorème (Vapnik)** – Soit \mathcal{H} donné, avec $VC(\mathcal{H}) = d$ avec m le nombre d'exemples d'entraînement. Avec une probabilité d'au moins $1 - \delta$, on a :

$$\epsilon(\hat{h}) \leq \left(\min_{h \in \mathcal{H}} \epsilon(h) \right) + O \left(\sqrt{\frac{d}{m} \log \left(\frac{m}{d} \right)} + \frac{1}{m} \log \left(\frac{1}{\delta} \right) \right)$$