

MAPSI — cours 10 :
Introduction au machine learning
Focus sur la Regression logistique

Vincent Guigue

`vincent.guigue@lip6.fr`

LIP6 – Université Paris 6, France

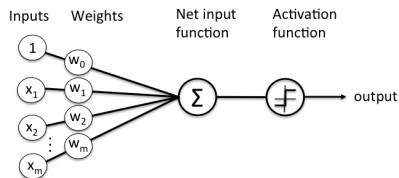
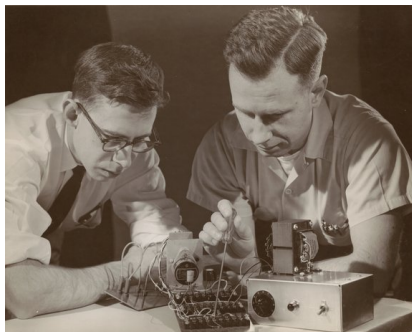
Perceptron

Vincent Guigue

`vincent.guigue@lip6.fr`

LIP6 – Université Paris 6, France

Historique



Schematic of Rosenblatt's perceptron.

$$f(\mathbf{x}_i) = \sum_j x_{ij} w_j,$$

$$C = \sum_{i=1}^N (-y_i \mathbf{x}_i \mathbf{w})_+$$

Liberté de la fonction de coût, comparaison des approches :

- Moindres carrés :
 - pas bien adapté à la classification
- Hinge (= charnière)
 - Limité au cas binaire, mais mieux adapté à la classification

Algorithme stochastique

Descente de gradient

- 1 Initialiser \mathbf{w}_0
- 2 Bouclage jusqu'à cvg
 - Calcul de
$$\nabla_{\mathbf{w}} C = \sum_{i|y_i f(\mathbf{x}_i) \leq 0} -y_i \mathbf{x}_i^T$$
 - MAJ : $\mathbf{w}^{t+1} = \mathbf{w}^t - \epsilon \nabla_{\mathbf{w}} C$

- 1 Initialiser \mathbf{w}_0
- 2 Bouclage jusqu'à cvg
 - Tirage aléatoire d'un échantillon i
 - Si $y_i \mathbf{x}_i \mathbf{w} \leq 0$
 - Calcul de
$$\nabla_{\mathbf{w}} C_i = -y_i \mathbf{x}_i^T$$
 - MAJ :
$$\mathbf{w}^{t+1} = \mathbf{w}^t - \epsilon \nabla_{\mathbf{w}} C_i$$

Renforcer la classification des points limites :

$$f(\mathbf{x}_i) = \sum_j x_{ij} w_j, \quad C = \sum_{i=1}^N (\textcolor{violet}{1} - y_i \mathbf{x}_i \mathbf{w})_+$$

Renforcer la classification des points limites :

$$f(\mathbf{x}_i) = \sum_j x_{ij} w_j, \quad C = \sum_{i=1}^N (\textcolor{violet}{1} - y_i \mathbf{x}_i \mathbf{w})_+$$

⇒ aucun impact... $\mathbf{x}_i \mathbf{w}$ n'est pas normalisé, la fonction de score est définie à un facteur près.

Renforcer la classification des points limites :

$$f(\mathbf{x}_i) = \sum_j x_{ij} w_j, \quad C = \sum_{i=1}^N (\textcolor{violet}{1} - y_i \mathbf{x}_i \mathbf{w})_+$$

⇒ aucun impact... $\mathbf{x}_i \mathbf{w}$ n'est pas normalisé, la fonction de score est définie à un facteur près.

⇒ imposer la marge + contraindre la norme de fonction de score :

$$f(\mathbf{x}_i) = \sum_j x_{ij} w_j, \quad C = \sum_{i=1}^N (\textcolor{violet}{1} - y_i \mathbf{x}_i \mathbf{w})_+ + \lambda \|\textcolor{blue}{\mathbf{w}}\|^2$$

Regression Logistique

Vincent Guigue

`vincent.guigue@lip6.fr`

LIP6 – Université Paris 6, France

Rappel sur les modèles génératifs

- 1 Choix d'une modélisation des données : $p(\mathbf{x}|\theta)$
- 2 Apprentissage = trouver θ
- 3 Application possible : décision bayésienne

$$r(\mathbf{x}) = \arg \max_k p(\theta_k|\mathbf{x}) = \frac{p(\mathbf{x}|\theta_k)p(\theta_k)}{p(\mathbf{x})}$$

- 4 Application bis : génération de $\tilde{\mathbf{x}} \sim \mathcal{D}(\theta_k)$

Apprentissage d'un modèle génératif \Leftrightarrow Estimation de densité

- Estimer θ_k = estimer une densité de probabilité d'une classe k
- Hypothèse (forte) : les θ_k sont supposés indépendants
- Techniques d'estimation des θ_k

Maximum de vraisemblance

- $D_k = \{\mathbf{x}_1, \dots, \mathbf{x}_N\}$ exemples supposés générés par $p(\mathbf{x}|\theta_k)$
Seulement pour la classe k
- *Faire coller* le modèle au données

$$\mathcal{L}(D_k, \theta_k) = p(D_k|\theta_k) = \prod_{i=1}^N p(\mathbf{x}_i|\theta_k)$$

- Optimisation : $\theta_k^* = \arg \max_{\theta_k} (\log \mathcal{L}(D_k, \theta_k))$
- Résolution :
 - Analytique : $\frac{\partial \mathcal{L}(D, \theta)}{\partial \theta} = 0$ ou Approchée : EM, gradient...
- Inférence sur une nouvelle donnée \mathbf{x} :

$$decision = k^* = \arg \max_k p(\mathbf{x}|\theta_k)$$

Limite du modèle génératif pour la classification

- Approche **générative** : travail classe par classe

Quel modèle colle le mieux à mon observation ?

- Approche **discriminante** : travail classe i VS classe j

Qu'est ce qui distingue la classe i de la classe j ?

Idée : travailler sur les $p(Y|X)$

Modèle (le plus connu) : **Régression logistique**

Formulation du problème

- Echantillons $\{\mathbf{x}_i\}_{i=1,\dots,n}, \mathbf{x} \sim X$
- Deux classes : $Y = 0$ ou $Y = 1$. Réalisation des $y_i \sim Y$
- En fait : $D = (\mathbf{x}_1, y_1), \dots, (\mathbf{x}_N, y_N)$
- Comment modéliser $p(Y|X)$?

Formulation du problème

- Echantillons $\{\mathbf{x}_i\}_{i=1,\dots,n}, \mathbf{x} \sim X$
- Deux classes : $Y = 0$ ou $Y = 1$. Réalisation des $y_i \sim Y$
- En fait : $D = (\mathbf{x}_1, y_1), \dots, (\mathbf{x}_N, y_N)$
- Comment modéliser $p(Y|X)$?

- 1 On repère une variable de Bernoulli
 $p(Y = 1|X = \mathbf{x}) = 1 - p(Y = 0|X = \mathbf{x})$
- 2 On choisit de modéliser *arbitrairement* :

$$p(Y = 1|X = \mathbf{x}) =$$

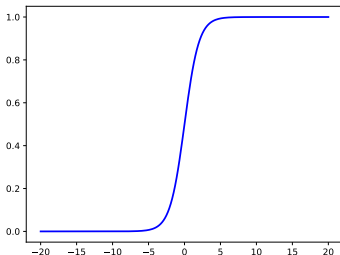
$$f(\mathbf{x}) = \frac{1}{1 + \exp(-(\mathbf{x}\mathbf{w} + b))}$$

Formulation du problème

- Echantillons $\{\mathbf{x}_i\}_{i=1,\dots,n}, \mathbf{x} \sim X$
- Deux classes : $Y = 0$ ou $Y = 1$. Réalisation des $y_i \sim Y$
- En fait : $D = (\mathbf{x}_1, y_1), \dots, (\mathbf{x}_N, y_N)$
- Comment modéliser $p(Y|X)$?

- 1 On repère une variable de Bernoulli
 $p(Y = 1|X = \mathbf{x}) = 1 - p(Y = 0|X = \mathbf{x})$
- 2 On choisit de modéliser *arbitrairement* :

$$p(Y = 1|X = \mathbf{x}) =$$
$$f(\mathbf{x}) = \frac{1}{1 + \exp(-(\mathbf{x}\mathbf{w} + b))}$$



- Données : X, Y

- Modèle :

$$p(Y = 1|X = \mathbf{x}) = f(\mathbf{x}) = \frac{1}{1 + \exp(-(\mathbf{xw} + b))}$$

- Bornes du modèle :

- $\lim_{\mathbf{xw} + b \rightarrow -\infty} f(\mathbf{x}) = 0$
- $\lim_{\mathbf{xw} + b \rightarrow \infty} f(\mathbf{x}) = 1$

Dimension des éléments en présence

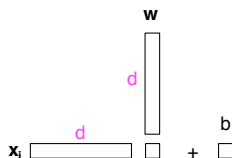
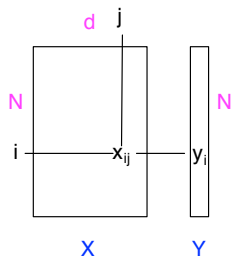
- Données : X, Y

- Modèle :

$$p(Y = 1|X = \mathbf{x}) = f(\mathbf{x}) = \frac{1}{1 + \exp(-(\mathbf{x}\mathbf{w} + b))}$$

- Bornes du modèle :

- $\lim_{\mathbf{x}\mathbf{w} + b \rightarrow -\infty} f(\mathbf{x}) = 0$
- $\lim_{\mathbf{x}\mathbf{w} + b \rightarrow \infty} f(\mathbf{x}) = 1$
- Si : $\mathbf{x}\mathbf{w} + b = 0 \Rightarrow f(\mathbf{x}) = 0.5$



Comment trouver les \mathbf{w}^* ?

Comment trouver les \mathbf{w}^* ?

⇒ **Par maximum de vraisemblance (conditionnelle) !**

Vraisemblance (conditionnelle) -indépendance entre échantillons-

$$L = \prod_{i=1}^N p(Y = y_i | X = \mathbf{x}_i)$$

- Truc de Bernoulli :

$$p(Y = y_i | X = \mathbf{x}_i) = p(Y = 1 | X = \mathbf{x}_i)^{y_i} (1 - p(Y = 1 | X = \mathbf{x}_i))^{1-y_i}$$

- Passage au log
- Remplacement des $p(Y = 1 | X = \mathbf{x})$ par des $f(\mathbf{x})$
- Nouvelle formulation :

$$\mathbf{w}^*, b^* = \arg \max_{\mathbf{w}, b} \sum_{i=1}^N [y_i \log(f(\mathbf{x}_i)) + (1 - y_i) \log(1 - f(\mathbf{x}_i))]$$

Données : $D = (\mathbf{x}_1, y_1), \dots, (\mathbf{x}_N, y_N)$

$$\mathbf{w}^*, b^* = \arg \max_{\mathbf{w}, b} \sum_{i=1}^N [y_i \log(f(\mathbf{x}_i)) + (1 - y_i) \log(1 - f(\mathbf{x}_i))]$$

On remplace $f(x)$ par sa valeur et on développe le coût...

... [quelques lignes de calcul] ...

Données : $D = (\mathbf{x}_1, y_1), \dots, (\mathbf{x}_N, y_N)$

$$\mathbf{w}^*, b^* = \arg \max_{\mathbf{w}, b} \sum_{i=1}^N [y_i \log(f(\mathbf{x}_i)) + (1 - y_i) \log(1 - f(\mathbf{x}_i))]$$

On remplace $f(x)$ par sa valeur et on développe le coût...

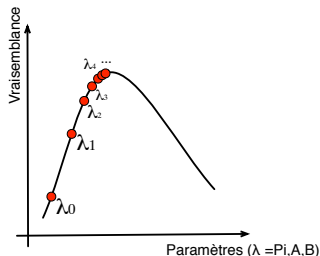
... [quelques lignes de calcul] ...

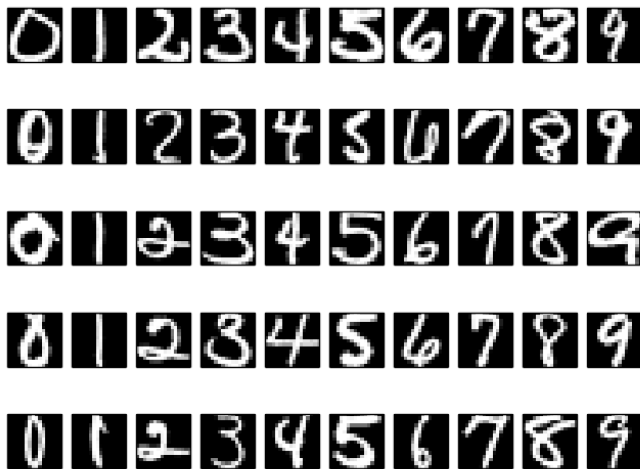
$$\frac{\partial}{\partial \mathbf{w}_j} L_{\log} = \sum_{i=1}^N x_{ij} \left(y_i - \frac{1}{1 + \exp(-(\mathbf{w} \cdot \mathbf{x}_i + b))} \right)$$

$$\frac{\partial}{\partial b} L_{\log} = \sum_{i=1}^N \left(y_i - \frac{1}{1 + \exp(-(\mathbf{w} \cdot \mathbf{x}_i + b))} \right)$$

- Annulation directe du gradient **impossible**
- \Rightarrow Algorithme itératif :
 - Init des paramètres \mathbf{w}_0, b_0
 - Tant que convergence non atteinte
 - Calcul des : $\frac{\partial L_{\log}}{\partial b}, \frac{\partial L_{\log}}{\partial w_j}$
 - Mise à jour (montée de gradient) : $\theta^t = \theta^{t-1} + \epsilon \frac{\partial L_{\log}}{\partial \theta}$

Cas convexe :

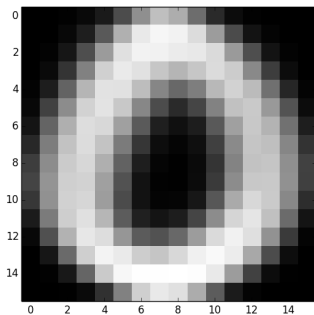




USPS : génératif VS discriminant

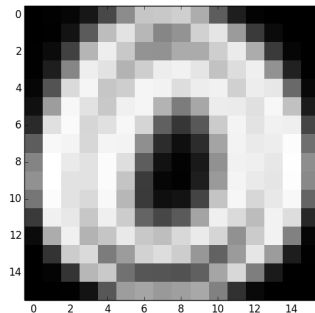
Modèle génératif gaussien : classe 0

Visualisation de la moyenne de la classe :



+ reshape

Visualisation de la variance de la classe :

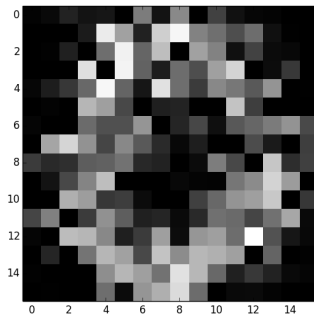


+ reshape

USPS : génératif VS discriminant

Possibilité de générer un échantillon :

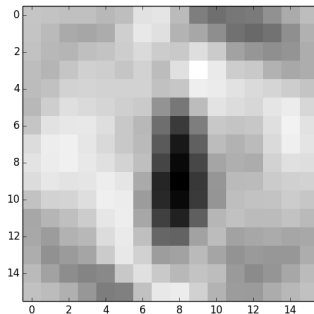
Tirage d'une valeur gaussienne pour chaque pixel...



Mais hypothèse d'indépendance des pixels \Rightarrow qualité BOF

USPS : génératif VS discriminant

En régression logistique : classe 0 VS toutes les autres



Mise en évidence des zones qui ne sont utilisées **que** par les 0

- Apprentissage : 7291 images
- Test : 2007 images

Naive Bayes (modèle de pixel gaussien)

Taux bonne classif. en apprentissage : 0.785

Taux bonne classif. en test : 0.739

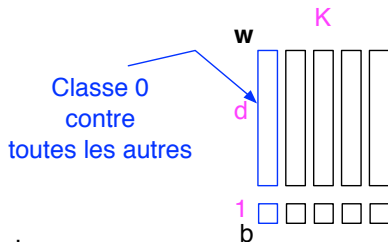
Regression logistique

Taux bonne classif. en apprentissage : 0.943

Taux bonne classif. en test : 0.903

Comment passer au multi-classes ?

un contre tous (*one against all*) : K classes $\Rightarrow K$
classifieurs **appris séparément**
sur **toutes les données**



- $f(\mathbf{x}) \Rightarrow f_k(\mathbf{x})$ et critère de décision :

$$k^* = \arg \max_k f_k(\mathbf{x})$$

Quelle classe veut **le plus** l'échantillon \mathbf{x} ?

- Critères de rejet :
 - pas de $f_k(\mathbf{x}) > 0.5$
 - plusieurs $f_k(\mathbf{x}) > 0.5$

Système de recommandation

Vincent Guigue

`vincent.guigue@lip6.fr`

LIP6 – Université Paris 6, France

On considère un examen, pour lequel un étudiant s peut répondre à la question q correctement, ce qui est noté $x_{qs} = 1$ ou incorrectement, $x_{qs} = 0$. On suppose que la chance de succès dépend de la capacité de l'étudiant α_s et de la difficulté de la question δ_q . On postule le modèle de réponse suivant :

$$p(x_{qs} = 1 | \alpha_s, \delta_q) = \sigma(\alpha_s - \delta_q), \quad \text{avec : } \sigma(x) = \frac{1}{1 + \exp(-x)}$$

$$p(x_{qs} = 1 | \alpha_s, \delta_q) = \sigma(\alpha_s - \delta_q), \quad \text{avec : } \sigma(x) = \frac{1}{1 + \exp(-x)}$$

- 1 Etudier rapidement la fonction σ et conclure qu'elle permet effectivement de modéliser une probabilité.
- 2 A quelle condition un étudiant s a-t-il autant de chance répondre correctement que de se tromper à une question q ?
- 3 Quelle loi permet de modéliser la variable x_{qs} ? Montrer que :

$$p(x_{qs} | \alpha_s, \delta_q) = \sigma(\alpha_s - \delta_q)^{x_{qs}} (1 - \sigma(\alpha_s - \delta_q))^{(1-x_{qs})}$$

où x_{qs} peut prendre les valeurs 0 ou 1.

MAPSI et le machine learning

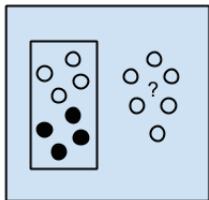
Vincent Guigue

`vincent.guigue@lip6.fr`

LIP6 – Université Paris 6, France

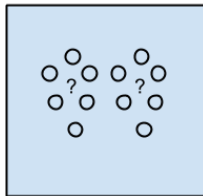
Différents cadres de machine learning

Supervisé



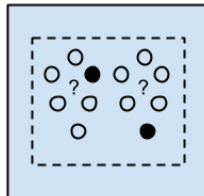
Supervised Learning Algorithms

Non-supervisé



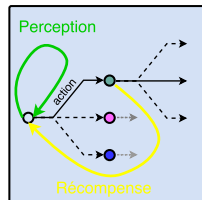
Unsupervised Learning Algorithms

Semi-supervisé



Semi-supervised Learning Algorithms

Renforcement



Jason Brownlee

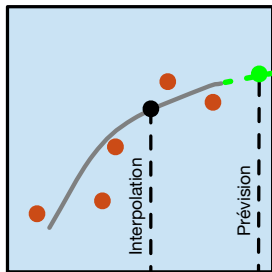
● Différents algorithmes... ... et différentes évaluations

● Différentes **données**, différents **coûts**...

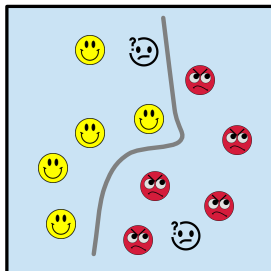
Et une nouvelle donne avec *Amazon Mechanical Turk*

Grande familles de problématiques

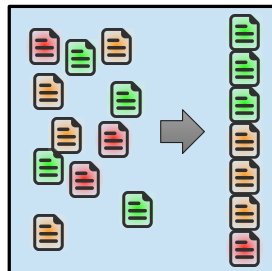
Régression



Classification

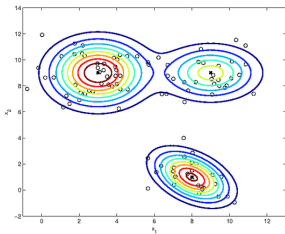


Ordonnancement



Modèles génératifs apprentissage Bayésien

- 1 Choix d'un modèle paramétrique
e.g. mixture de 3 Gaussiennes
- 2 Apprentissage des paramètres
maximum de vraisemblance
- 3 Exploitation = inférence

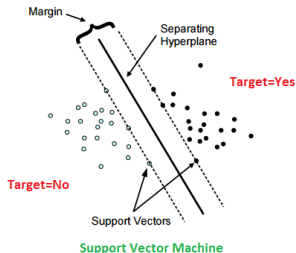


e.g. Mixtures, Naive Bayes, MM, HMM...

Usages : extraction thématique,
classification de spam

Modélisation discriminante (classification)

- 1 Apprentissage d'une frontière
i.e. chercher les différences dans les
caractéristiques
- 2 Classification de nouveaux
points

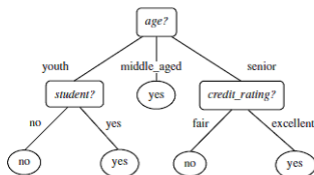


e.g. Perceptron, SVM, Régression
logistique,...

Usages : classification de signaux, de
textes

Arbre de décision

- 1 Sélectionner un caractéristique
- 2 Couper
- 3 Décider

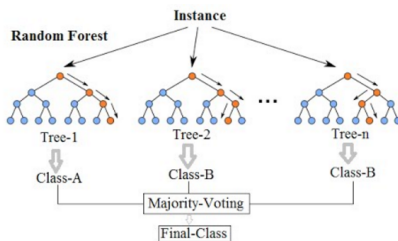


e.g. C4.5

Usages : besoin d'une décision expliquée

Approches ensemblistes

- 1 Multiplier les classifieurs
- 2 Fusionner les décisions

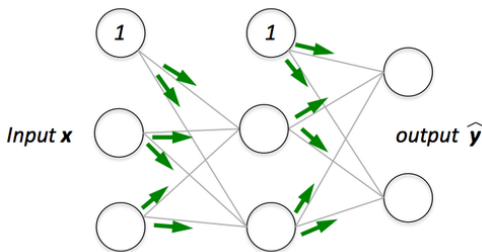


e.g. Random Forest, Boosting, Bagging

Usages : classification robuste,
parallélisable

Les réseaux de neurones

- 1 Au départ, un **opérateur paramétrique complexe...**
Et un algorithme d'apprentissage efficace.
- 2 une chaîne de traitements capable
d'**extraire des caractéristiques pertinentes automatiquement**

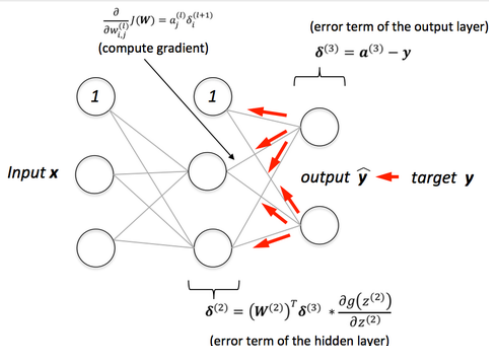


Sebastian Raschka

Les familles d'algorithmes (3/3)

Les réseaux de neurones

- 1 Au départ, un **opérateur paramétrique complexe...**
Et un algorithme d'apprentissage efficace.
- 2 une chaîne de traitements capable
d'**extraire des caractéristiques pertinentes automatiquement**

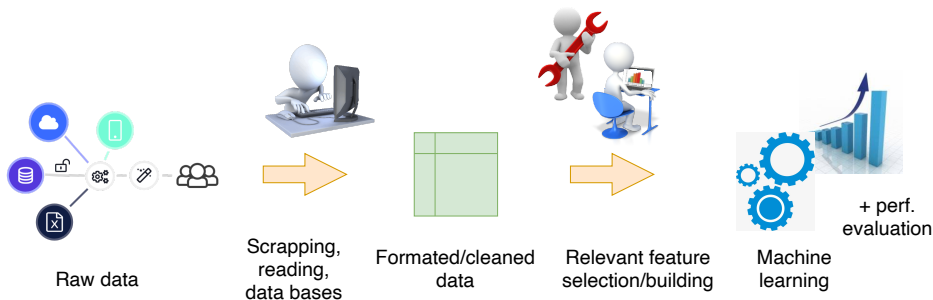


Sebastian Raschka

Toutes les étapes comptent...

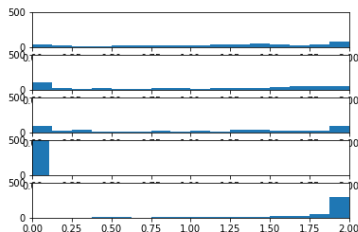
Surtout les premières pour les performances !

Choisir & optimiser un modèle n'est qu'une partie du travail...



⇒ Valoriser vos compétences d'informaticien !

- Histogramme de répartition des classes
 - Critique pour la définition des *a priori*
- Histogramme de d'analyse des valeurs d'une variable
 - Comprendre les données
 - Choisir un modèle pour coller à cette variable en fonction de l'Histogramme
 - Faire marcher les arbres de décision –plus tard :)–



Distribution de différents pixels
dans une base USPS

- Supervisé/non-supervisé
- Complétion de données manquante
- Classification / régression
- Compléter des données manquantes
(est-ce une classe de problème ?)

Exemple :

- Prédire les ventes de parfum lors des prochaines soldes
- Reconnaître qu'une image contient un chat

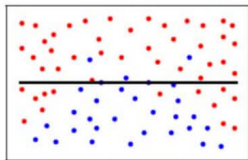
- Choisir une ou plusieurs loi de proba. pour les données
- Formuler la vraisemblance
- Optimiser la vraisemblance
 - Trouver les paramètres optimaux des lois usuelles sur wikipedia
 - Sinon, annuler la dérivée de la vraisemblance
 - Sinon, maximiser itérativement la vraisemblance
- Traiter de nouvelles données en inférence (les faire passer dans le modèle optimisé)

[MAPSI] Optimiser un modèle basé sur une fonction de coût

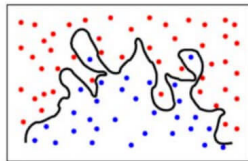
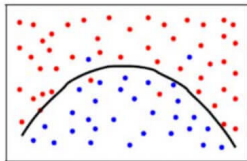
- Choisir une représentation des données, un modèle
- Choisir une fonction de coût
- Optimiser le coût
 - Annuler la dérivée du coût
 - Sinon, minimiser itérativement le coût
- Traiter de nouvelles données en inférence (les faire passer dans le modèle optimisé)

- **Apprendre** un modèle est aussi important que de **l'évaluer**
- **Apprendre** et **évaluer** sur les mêmes données est aberrant

Underfitting



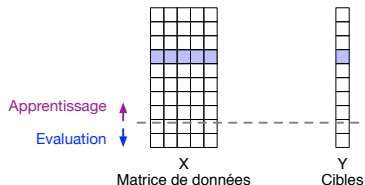
Overfitting



Tom Robertshaw

- **Apprendre** un modèle est aussi important que de **l'évaluer**
- **Apprendre** et **évaluer** sur les mêmes données est aberrant

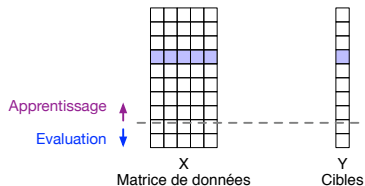
Dilemme de répartition
des données :



- **Apprendre** un modèle est aussi important que de **l'évaluer**
- **Apprendre** et **évaluer** sur les mêmes données est aberrant

Dilemme de répartition
des données :

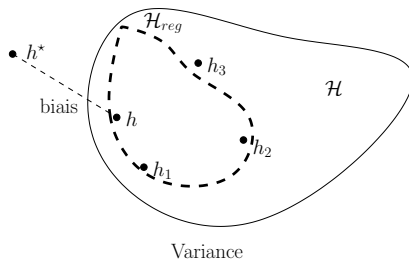
Solution = validation croisée



Golden Helix

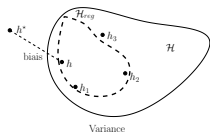
Mais pourquoi aller plus loin ???

Une proposition d'analyse avec le dilemme biais-variance



- Variance = taille de l'espace de recherche du modèle
 - Représentation des données + complexité du modèle (paramètres & hyper-paramètres)
- Biais = distance entre le meilleur modèle et le modèle retenu

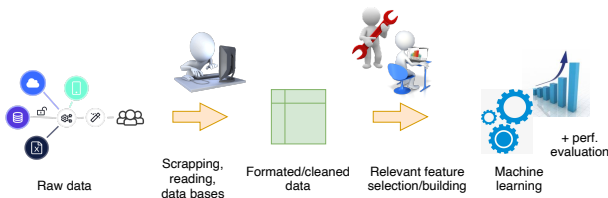
Différents paradigmes au fil des époques (1/4)



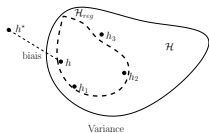
Exploiter seulement des informations pertinentes proposées/construites par un expert

- Efficace très vite
- Peu de bruit = les modèles simples marchent bien

⇒ Approches très rentables... Que vous êtes déjà en mesure d'implémenter !



Différents paradigmes au fil des époques (2/4)



Générer [automatiquement] plein de caractéristique... Puis sélectionner celles qui sont utiles

A priori Critères de sélection/transformation de variables

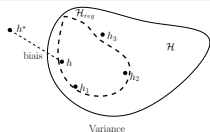
- ACP
- Sélection de variables sur critère de séparabilité des données

On Line Apprendre ce qui est utile ou pas = **régularisation** Soit des données $X \in \mathbb{R}^{N \times d}$ avec d très grand et une fonction de décision/régression linéaire $f(\mathbf{x}) = \mathbf{x} \cdot \mathbf{w}$

$$\arg \min_{\mathbf{w}} \mathcal{C} + \lambda \Omega(\mathbf{w}), \Omega(\mathbf{w}) = \begin{cases} \sum_j w_j^2 & \text{régul. } L_2 \\ \sum_j |w_j| & \text{régul. } L_1 \end{cases}$$

$\Rightarrow \lambda$ devient un hyper-paramètre critique !

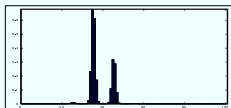
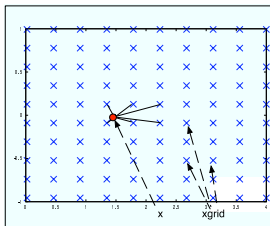
Différents paradigmes au fil des époques (3/4)



Les méthodes à noyaux et les espaces de représentation universel

- Moins travailler sur les descripteurs = moins besoin d'expertise externe.
- Avoir des fonctions à la fois simple à apprendre [formulation convexe, régularisation] mais capable de traiter des cas complexe.

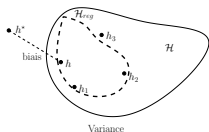
$$f(\mathbf{x}) = \sum_i w_i k(\mathbf{x}, \mathbf{x}_i), \text{ où } k \text{ est une fonction de similarité}$$



Représentation en histogramme des similarités gaussiennes entre un point (rond rouge) et tous les points de la grille

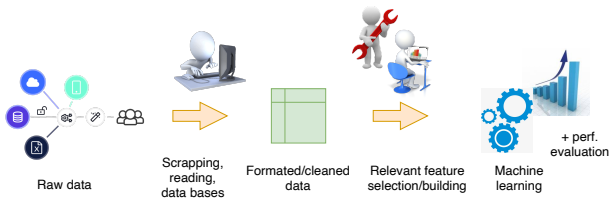
$$\phi(\mathbf{x})_j = \exp\left(-\frac{\|\mathbf{x} - \mathbf{x}_{\text{grid}_j}\|^2}{2\sigma^2}\right)$$

Différents paradigmes au fil des époques (4/4)



Créer des opérateurs super-complexes... Mais efficaces

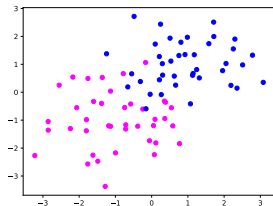
- Réseaux de neurones
 - La promesse d'une extraction de caractéristiques automatiques
 - L'apprentissage de représentation et la sémantique
- XGBoost



Curse of dimensionality

A classical toy example to illustrate the curse of dimensionality :

Original dataset :



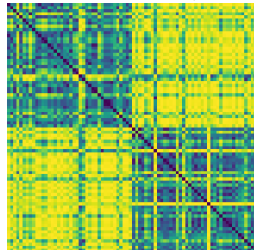
Matrix view

Matrix of raw points



Distance matrix

Matrix of distance between points

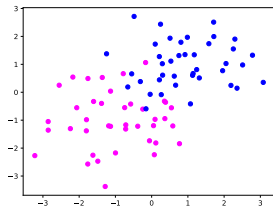


Easy problem / classes are clearly separated

Curse of dimensionality

A classical toy example to illustrate the curse of dimensionality :

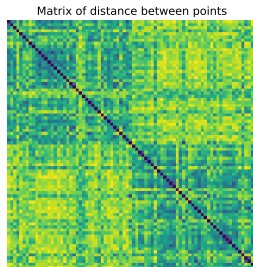
Original dataset :



Matrix view
Matrix of raw points



Distance matrix

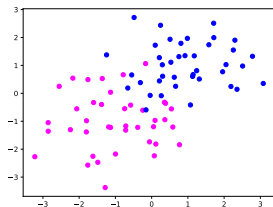


Adding some noisy dimensions in the dataset

Curse of dimensionality

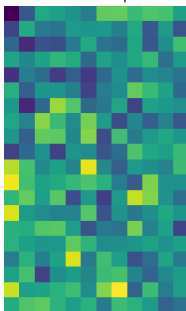
A classical toy example to illustrate the curse of dimensionality :

Original dataset :



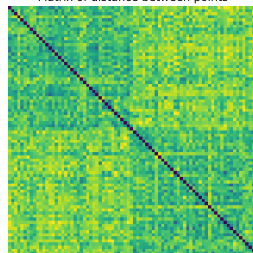
Matrix view

Matrix of raw points



Distance matrix

Matrix of distance between points



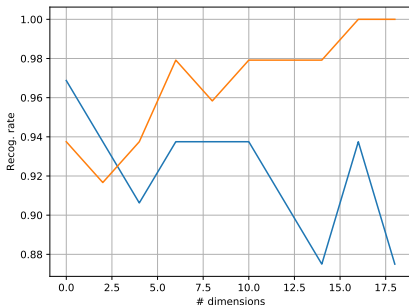
Adding **more** noisy dimensions in the dataset

⇒ Euclidian distance is very sensitive to the dimensionality issue

Curse of dimensionality

A classical toy example to illustrate the curse of dimensionality :

Basic classifier on those datasets



⇒ Learn accuracy ↗, test accuracy ↘ = **overfitting**

⇒ All points tend to lay on an hypersphere (they become equidistant)

Distances between points in the dataset

