

MACHINE-LEARNING (2) SÉLECTION DE MODÈLES ET CAS AVANCÉS

Vincent Guigue
vincent.guigue@agroparistech.fr

EVALUATION(S)



Nombreuses métriques disponibles, pour différentes application

Soit les observations $\{\mathbf{x}_i, y_i\}_{i=1, \dots, n}$, $(\mathbf{x}_i, y_i) \in \mathcal{X} \times \mathcal{Y}$

Les métriques les plus classiques :

- Negative Log-Likelihood (NLL)

- MSE / RMSE : $\mathcal{L} = \frac{1}{n} \sum_i (f(\mathbf{x}_i) - y_i)^2$, $\mathcal{L} = \sqrt{\frac{1}{n} \sum_i (f(\mathbf{x}_i) - y_i)^2}$

- MAPE : $\mathcal{L} = \frac{1}{n} \sum_i \frac{|f(\mathbf{x}_i) - y_i|}{|y_i|}$

- Et les variantes dans le cas $y_i = 0$ (cf sMAPE)

- Accuracy : $\mathcal{L} = \frac{1}{n} \sum_i \mathbb{1}_{y_i = f(\mathbf{x}_i)}$

- Perceptron : $\mathcal{L} = \frac{1}{n} \sum_i (-y_i \cdot f(\mathbf{x}_i))_+$

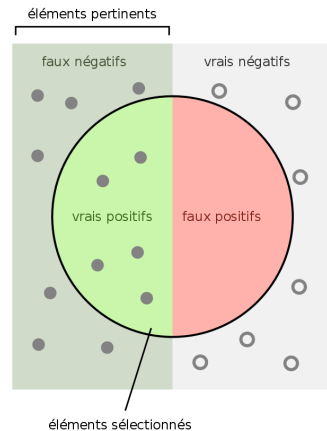
Quelle métrique pour quelle application ? Quelles contraintes sur $\mathcal{X} \times \mathcal{Y}$?

ATTENTION : ne pas confondre *métrique d'évaluation* & *coût à optimiser*

Détection d'évènement (vs bruit de fond)

Classification \Rightarrow **métrique dédiée à une classe particulière**

- Precision (sensibilité) $\frac{TP}{TP + FP} = \frac{\text{detections pertinentes}}{\text{detections}}$
- rappel (couverture) $\frac{TP}{TP + FN} = \frac{\text{detections pertinentes}}{\text{taille de la classe}}$
- $f1 = 2 \cdot \frac{\text{precision} \cdot \text{rappel}}{\text{precision} + \text{rappel}}$



Aggrégation sur toutes les classes (dans les cas pertinents)

- macro-moyenne : moyenne des scores des classes

- micro-moyenne : $\mu Prec = \frac{\sum_c TP_c + TP_c + TP_c}{\sum_c TP_c + FP_c}$

Combien de candidats sélectionnés sont pertinents ?

$$\text{Précision} = \frac{\text{vrais positifs}}{\text{vrais positifs} + \text{faux positifs}}$$

Combien d'éléments pertinents sont sélectionnés ?

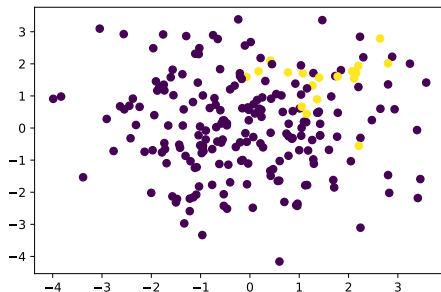
$$\text{Rappel} = \frac{\text{vrais positifs}}{\text{vrais positifs} + \text{faux négatifs}}$$



Classification déséquilibrée

Ex : Fraude à la carte bleue : 0.3 pour mille transactions...

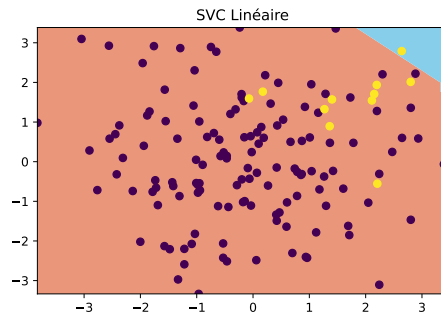
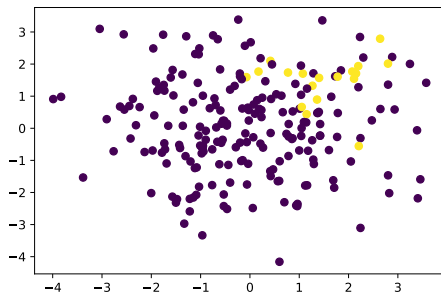
- Impact sur les modèles
- Impact sur métriques



Classification déséquilibrée

Ex : Fraude à la carte bleue : 0.3 pour mille transactions...

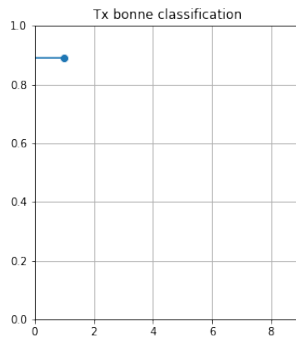
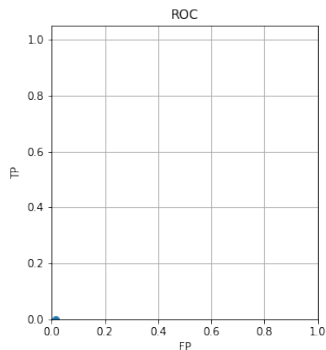
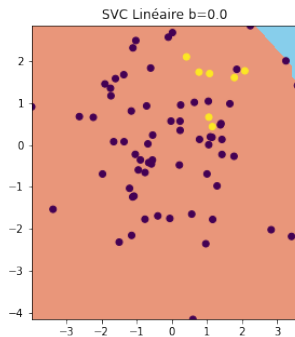
- Impact sur les modèles
 - Le modèle prédit que tout est OK
- Impact sur métriques
 - La métrique indique 99.07% de taux de bonne classification !



ROC : 1 classifieur \Rightarrow ensemble des classifieurs biaisés

Courbe ROC : *receiver operating characteristic*

Mesure de la capacité à détecter les événements sans rajouter de bruit en faisant varier la sensibilité

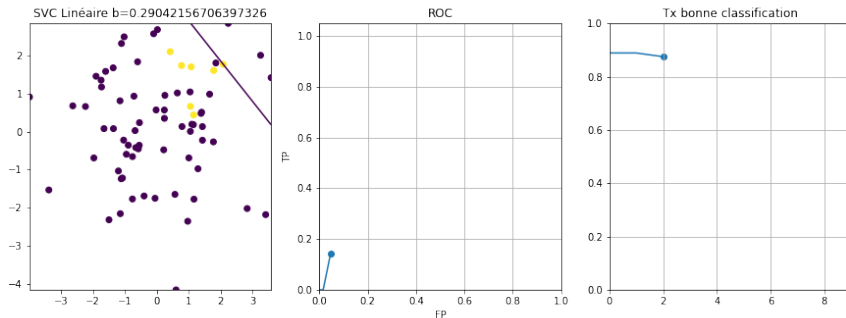




ROC : 1 classifieur \Rightarrow ensemble des classifieurs biaisés

Courbe ROC : *receiver operating characteristic*

Mesure de la capacité à détecter les événements sans rajouter de bruit en faisant varier la sensibilité

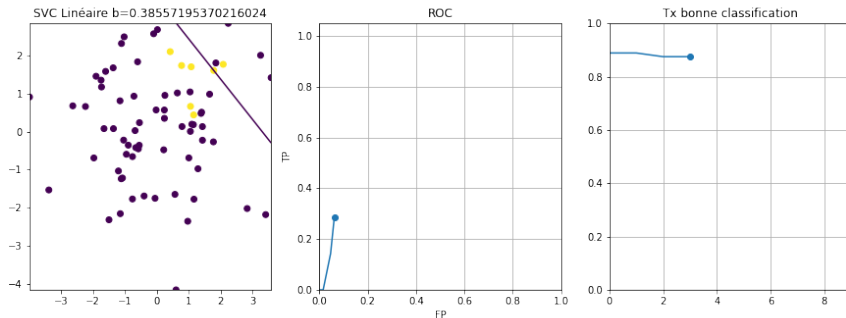




ROC : 1 classifieur \Rightarrow ensemble des classifieurs biaisés

Courbe ROC : *receiver operating characteristic*

Mesure de la capacité à détecter les événements sans rajouter de bruit en faisant varier la sensibilité

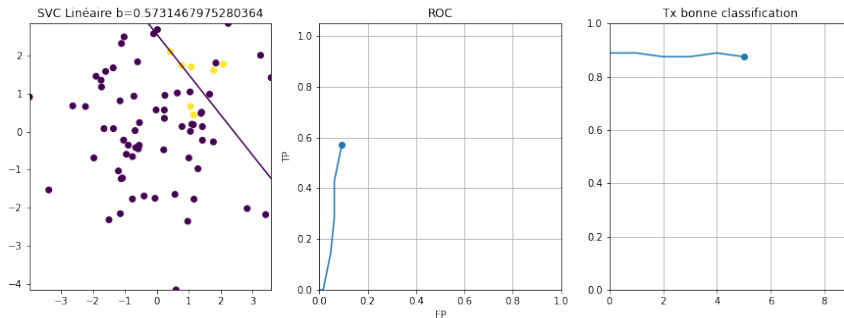




ROC : 1 classifieur \Rightarrow ensemble des classifieurs biaisés

Courbe ROC : *receiver operating characteristic*

Mesure de la capacité à détecter les événements sans rajouter de bruit en faisant varier la sensibilité

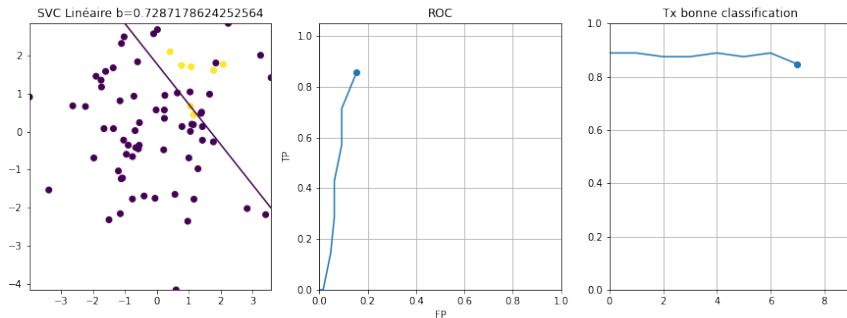




ROC : 1 classifieur \Rightarrow ensemble des classifieurs biaisés

Courbe ROC : *receiver operating characteristic*

Mesure de la capacité à détecter les événements sans rajouter de bruit en faisant varier la sensibilité

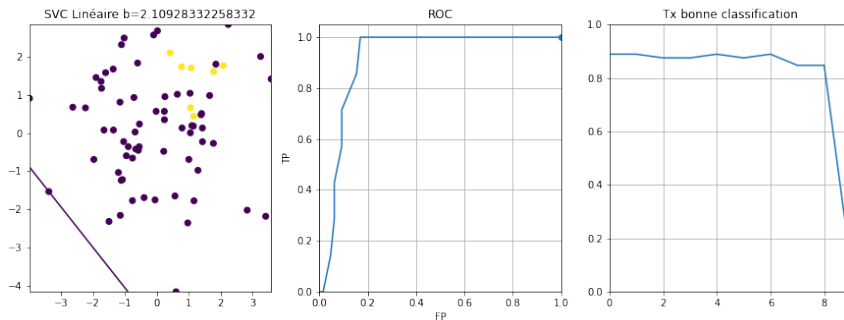




ROC : 1 classifieur \Rightarrow ensemble des classifieurs biaisés

Courbe ROC : *receiver operating characteristic*

Mesure de la capacité à détecter les événements sans rajouter de bruit en faisant varier la sensibilité





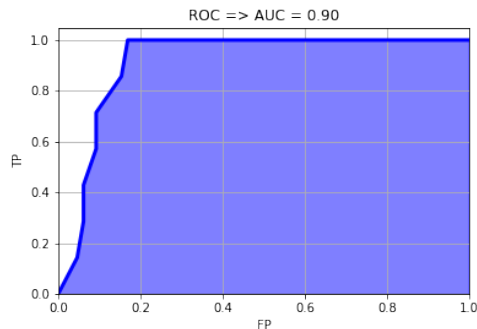
ROC : 1 classifieur \Rightarrow ensemble des classifieurs biaisés

Courbe ROC : *receiver operating characteristic*

Mesure de la capacité à détecter les événements sans rajouter de bruit en faisant varier la sensibilité

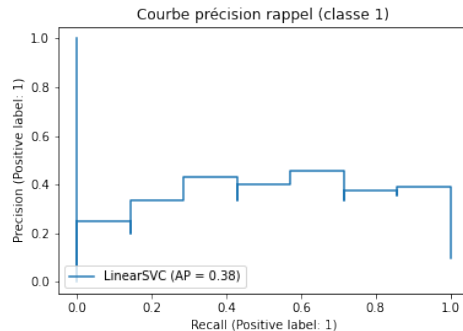
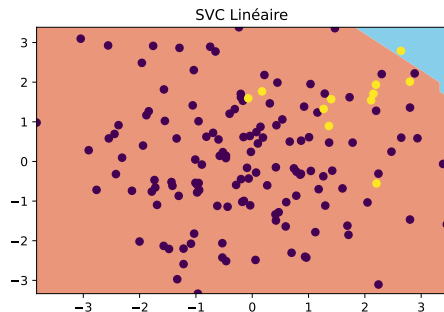
Indicateurs multiples = difficile à manipuler, expliquer...

- Précision + Rappel \Rightarrow f1
- ROC \Rightarrow AUC : Area Under the Curve



Précision/Rappel (avec tous les biais)

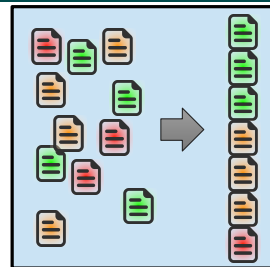
On peut faire la même chose en précision rappel :



La fonction d'affichage de scikit-learn est encore perfectible... mais on voit les informations importantes.

Autres problèmes, autres métriques

Comment évaluer un modèle d'ordonnement (= *Ranking*) ?
 Critique pour l'accès à l'information : **moteur de recherche, systèmes de recommandation**



- Distance d'édition (Levenshtein)
- Mean Reciprocal Rank (MRR) = en combien de coups j'attrape un document d'intérêt
- Favoriser le haut de la liste :
 - Mean Average Precision
 - nDCG
 - ATOP

⇒ A discuter si on fait un séminaire autour de ces thématiques

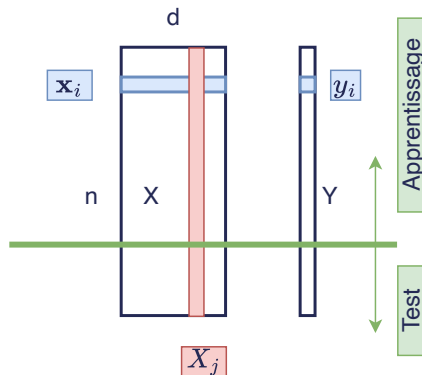
N	I	C	H	E		
		C	H	I	E	N

- effacement
- insertion
- substitution

Processus d'évaluation

!! L'évaluation est aussi importante que l'apprentissage !!

- Evaluer sur les données d'apprentissage (=qui ont servi à régler les paramètres)
⇒ **Tricherie, surestimation des performances**
- Evaluer sur des données vierges = OK



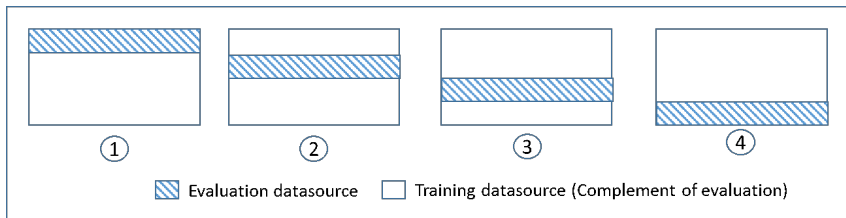
Problème de la répartition entre apprentissage et test

- La validation croisée

Processus d'évaluation

!! L'évaluation est aussi importante que l'apprentissage !!

- Evaluer sur les données d'apprentissage (=qui ont servi à régler les paramètres)
⇒ **Tricherie, surestimation des performances**
- Evaluer sur des données vierges = OK
- La validation croisée





Processus d'évaluation

!! L'évaluation est aussi importante que l'apprentissage !!

- Evaluer sur les données d'apprentissage (=qui ont servi à régler les paramètres)
⇒ **Tricherie, surestimation des performances**
- Evaluer sur des données vierges = OK
- La validation croisée

2 options pour l'implémentation :

- 1 Séparation train-test (itérative pour la validation croisée) + apprentissage + calcul de scores
- 2 Calcul direct des scores de validation croisée pour un modèle (la boucle, la division des données, l'apprentissage et scoring sont cachés dans la méthode)



Evaluation = clé de sélection des traitements

- 1 Phase critique de **dialogue avec le client**
 - Identifier les attentes
 - Montrer le niveau de performances
- 2 Outil critique pour la **sélection de modèles**

SÉLECTION DE MODÈLES

scikit-learn
algorithm cheat-sheet

Sélection de modèles : approche théorique

■ Bayes Information Criterion (BIC)

$$BIC = \log(n)k - 2\log(L), \quad k, n = \text{nb params/observations}, L = \text{vraisemblance}$$

■ Akaike Information Criterion (AIC)

$$AIC = 2k - \log(L), \quad k = \text{nb params}, L = \text{vraisemblance}$$

On cherche un modèle qui minimise un compromis entre complexité et vraisemblance
(=adéquation aux données)

⇒ Pour les modélisations probabilistes...

⇒ Ne marche pas en pratique (ou alors pour les modèles où k a peu d'impact, e.g. modèle AR)

Sélection de modèles : approche empirique

[en complément de l'approche expert]

- Liste des modèles de références : Naïve Bayes, Régression logistique, SVM, Decision Tree
- Liste des modèles avancés : Random Forest, Gradient Boosting, réseaux de neurones

Essai + Performance \Rightarrow sélection

Tout en gardant en tête :

- 1 que les références et l'état de l'art dépendent des applications
- 2 la puissance des approches ensemblistes



Sélection de modèles : les outils pratiques

Création d'un ensemble de paramètres à tester S

- Boucle for + critère pour scorer les S_i
 - apprentissage de modèle + perf. en test
 - validation croisée
 - critère de séparabilité des données (Fisher, ...)
- Grid Search
 - Même procédure généralisée à plusieurs paramètres

En pratique, on tatonne souvent à la main avant de tester finement sur une grille.

- Trouver les paramètres les plus influents / les plus sensibles
- Déterminer la plage de paramètres à explorer

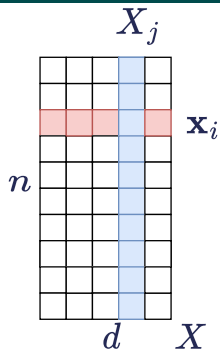
SÉLECTION DE CARACTÉRISTIQUES

14/26

14/26



Sélection de caractéristiques/variables/features



- Individu : $\mathbf{x}_i \in \mathbb{R}^d$
- Caractéristique (feature) X_j : variable de description des individus

Contre exemple :

Résolution d'un problème de régression au sens des moindres carrés avec $d > n$:

$$X^T X \cdot \mathbf{w} = X^T y$$

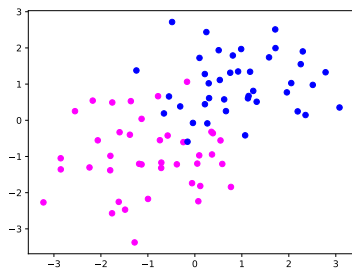
... Quid des dimensions ?

$X^T X \in \mathbb{R}^{d \times d}$... Mais une matrice de rang n : on a en fait n équations indépendantes pour trouver d inconnues \Rightarrow Problème mal posé (probablement) insoluble

Fléau de la dimensionalité - Curse of dimensionality

A classical toy example to illustrate the curse of dimensionality :

Original dataset :



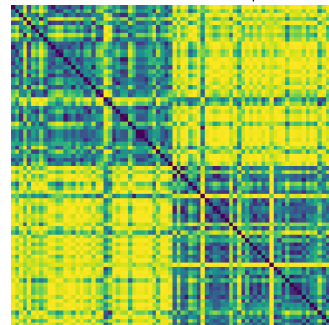
Matrix view

Matrix of raw points



Distance matrix

Matrix of distance between points

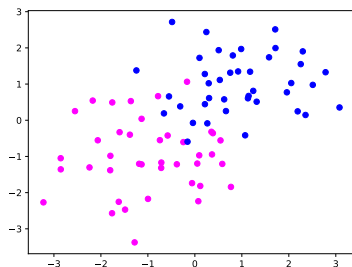


Easy problem / classes are clearly separated

Fléau de la dimensionalité - Curse of dimensionality

A classical toy example to illustrate the curse of dimensionality :

Original dataset :



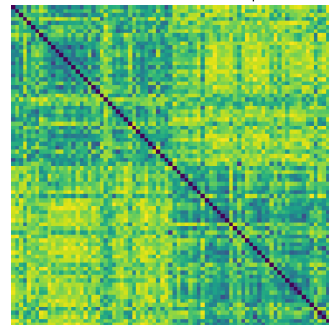
Matrix view

Matrix of raw points



Distance matrix

Matrix of distance between points

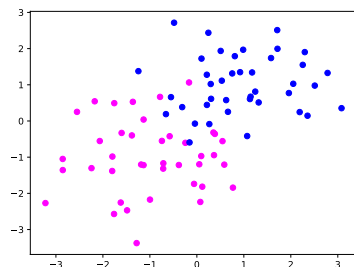


Adding some noisy dimensions in the dataset

Fléau de la dimensionalité - Curse of dimensionality

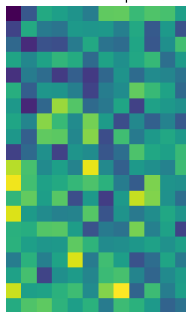
A classical toy example to illustrate the curse of dimensionality :

Original dataset :



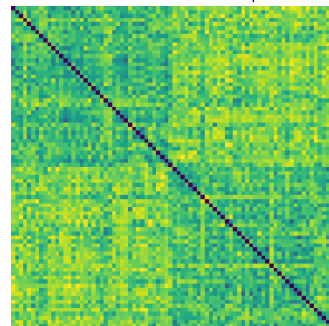
Matrix view

Matrix of raw points



Distance matrix

Matrix of distance between points



Adding **more** noisy dimensions in the dataset

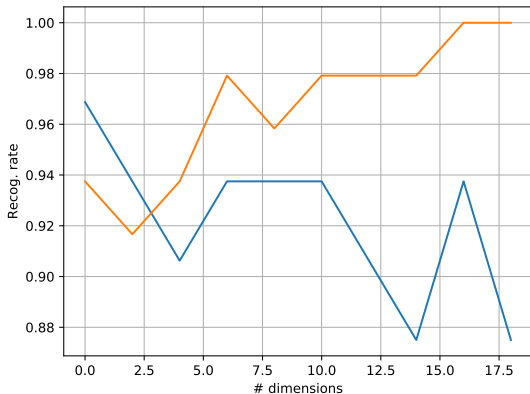
⇒ Euclidian distance is very sensitive to the dimensionality issue



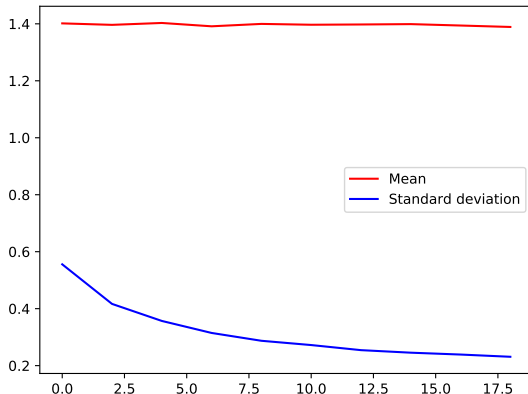
Fléau de la dimensionalité - Curse of dimensionality

A classical toy example to illustrate the curse of dimensionality :

Basic classifier on those datasets



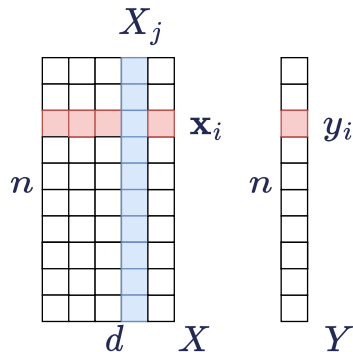
Distances between points in the dataset



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

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

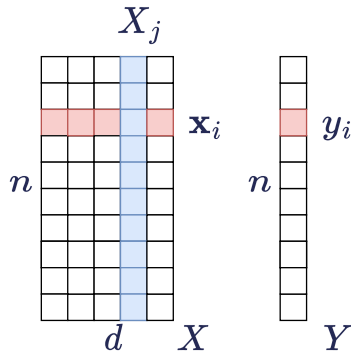
Bonne ou mauvaise variable ?



Comment déterminer si X_j est une variable d'intérêt ?

- Calculer le score de corrélation $X_j^T Y$

Bonne ou mauvaise variable ?



Comment déterminer si X_j est une variable d'intérêt ?

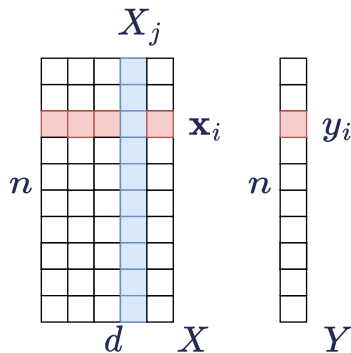
- Calculer le score de corrélation $X_j^T Y$

Ne prend pas en compte les approches non linéaires :

Avec les SVM, on peut facilement construire une variable X_j

- Idéale pour estimer Y
- De corrélation faible ou nulle avec Y

Bonne ou mauvaise variable ?



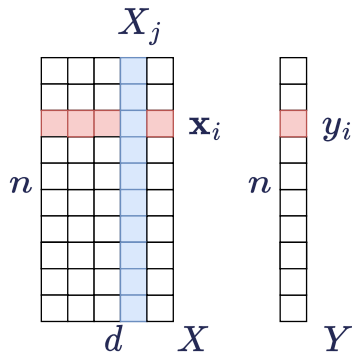
Comment déterminer si X_j est une variable d'intérêt ?

- Calculer le score de corrélation $X_j^T Y$

Quid des variables intéressantes mais corrélées entre elles :

$$X_j^T Y \nearrow \nearrow, \quad X_k^T Y \nearrow \nearrow \text{ Mais avec : } X_j \approx X_k$$

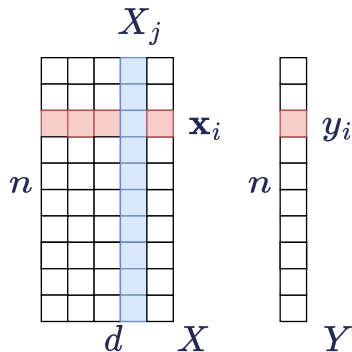
Bonne ou mauvaise variable ?



Comment déterminer si X_j est une variable d'intérêt ?

- Calculer le score de corrélation $X_j^T Y$
- Combien de variables conserver ?

Bonne ou mauvaise variable ?



Comment déterminer si X_j est une variable d'intérêt ?

- Calculer le score de corrélation $X_j^T Y$
- Combien de variables conserver ?
- Certaines variables ont-elles un intérêt *combinées* à d'autres ? (retour au cas non linéaire.)

Sélection de caractéristiques

⇒ Un besoin de obtenir de bonnes performances
... mais comment faire ?

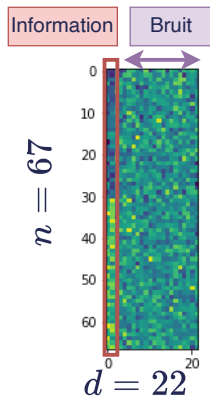
Plusieurs grandes familles de solution :

- 1 Choisir les caractéristiques pertinentes et éliminer les autres
 - à l'aide d'un expert ;
 - en analysant les paramètres des classifieurs (introspection)
 - en testant des combinaisons + scoring (extrospection)
- 2 Utiliser l'ACP (Analyse en Composantes Principales = PCA en anglais) pour trouver les meilleures combinaisons de variables.
 - Lien avec l'apprentissage non supervisé
- 3 Essayer d'apprendre les variables à éliminer au cours de l'apprentissage.

(1) Utiliser les poids d'un classifieur linéaire

Un classifieur ne sait pas bien gérer les dimensions superflues...
mais la hierarchie des poids peut avoir un sens.

$$f(\mathbf{x}_i) = \sum_{j=1}^d w_j x_{ij}, \quad \Rightarrow w_j \approx \text{importance de } X_j \text{ dans la décision}$$



Comment est ce que les w_j vont pondérer les X_j ?

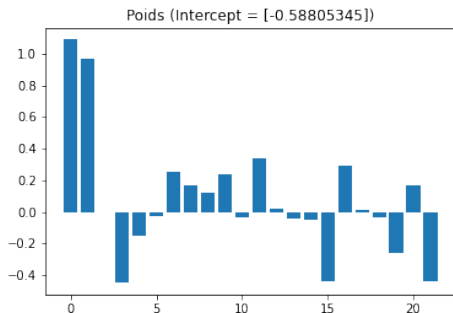
Classifieurs linéaires populaires :

- SVM linéaire
- Régression logistique
- Ridge classification
- ...

(1) Utiliser les poids d'un classifieur linéaire

Un classifieur ne sait pas bien gérer les dimensions superflues...
mais la hierarchie des poids peut avoir un sens.

$$f(\mathbf{x}_i) = \sum_{j=1}^d w_j x_{ij}, \quad \Rightarrow w_j \approx \text{importance de } X_j \text{ dans la décision}$$



- **Bonne nouvelle** : les poids des caractéristiques importantes sont plus hauts
- **Mauvaise nouvelle** : les autres poids sont assez importants



(1) Sélection Itérative de caractéristiques (par élimination)

Des approches gloutonnes (*greedy*)... Mais très chères !

Recursive Backward Elimination

- 1 Init. de toutes les caractéristiques : $S = \{X_1, \dots, X_d\}$
- 2 Faire d fois :
 - Pour tous les sous-ensembles : $S_i = S/X_i$
 - $p_i = \text{perf}(S_i)$ (taux en test, validation croisée, ...)
 - Conserver le meilleur sous ensemble : $S = S_i^*$

Quel coût ?

(1) Sélection Itérative de caractéristiques (par élimination)

Des approches gloutonnes (*greedy*)... Mais très chères !

Recursive Backward Elimination

- 1 Init. de toutes les caractéristiques : $S = \{X_1, \dots, X_d\}$
- 2 Faire d fois :
 - Pour tous les sous-ensembles : $S_i = S/X_i$
 - $p_i = \text{perf}(S_i)$ (taux en test, validation croisée, ...)
 - Conserver le meilleur sous ensemble : $S = S_i^*$

Quel coût ?

- d iterations
- × une itération = $d - 1$ tests (avec d décroissant)
- × un test = apprentissage + évaluation OU n_{CV} apprentissages + évaluations

Quel coût pour la méthode précédente ?



(1) Sélection Itérative de caractéristiques (par agrégation)

Idem... Mais par agrégation : on part d'un ensemble vide et on remplit

Forward selection

- 1 Initialisation vide : $S = \emptyset$
- 2 Faire d fois :
 - Pour tous les sous-ensembles : $S_i = S + X_i$
 - $p_i = \text{perf}(S_i)$ (taux en test, validation croisée, ...)
 - Conserver le meilleur sous ensemble : $S = S_i^*$



(1) Sélection Itérative de caractéristiques (par agrégation)

Idem... Mais par agrégation : on part d'un ensemble vide et on remplit

Forward selection

- 1 Initialisation vide : $S = \emptyset$
- 2 Faire d fois :
 - Pour tous les sous-ensembles : $S_i = S + X_i$
 - $p_i = \text{perf}(S_i)$ (taux en test, validation croisée, ...)
 - Conserver le meilleur sous ensemble : $S = S_i^*$

Le prix est (un peu) moindre... Mais pourquoi ?

Backward

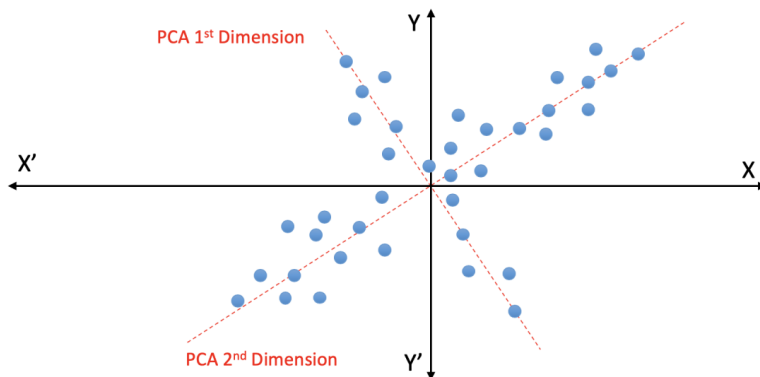
- d iterations
- × une itération = $d - 1$ tests
(avec d **décroissant**)
- × un test = apprentissage + évaluation
OU n_{CV} apprentissages + évaluations

Forward

- d iterations
- × une itération = $d - 1$ tests
(avec d **croissant**)
- × un test = apprentissage + évaluation
OU n_{CV} apprentissages + évaluations

(2) ACP/PCA : Analyse en composantes principales

Idée : trouver la combinaison d'axes expliquant au mieux la variance des données...
la valeur propre donne la *force* de l'explication



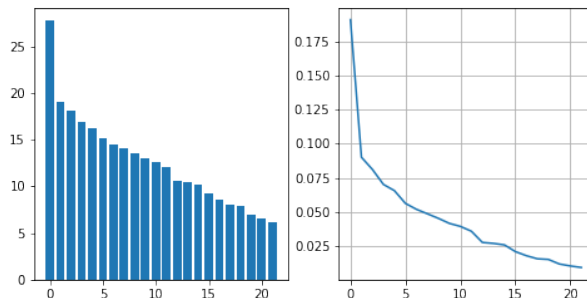
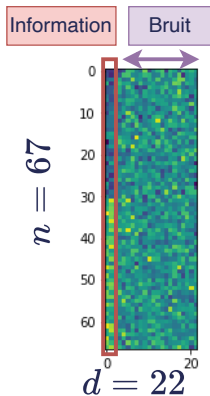
Est ce de la **sélection** ou de la **construction** de caractéristiques ??

(2) ACP/PCA : Analyse en composantes principales

Idée : trouver la combinaison d'axes expliquant au mieux la variance des données...
la valeur propre donne la *force* de l'explication

22 dimensions \Rightarrow 22 valeurs propres + 22 vecteurs propres

... mais peu de valeurs significatives :

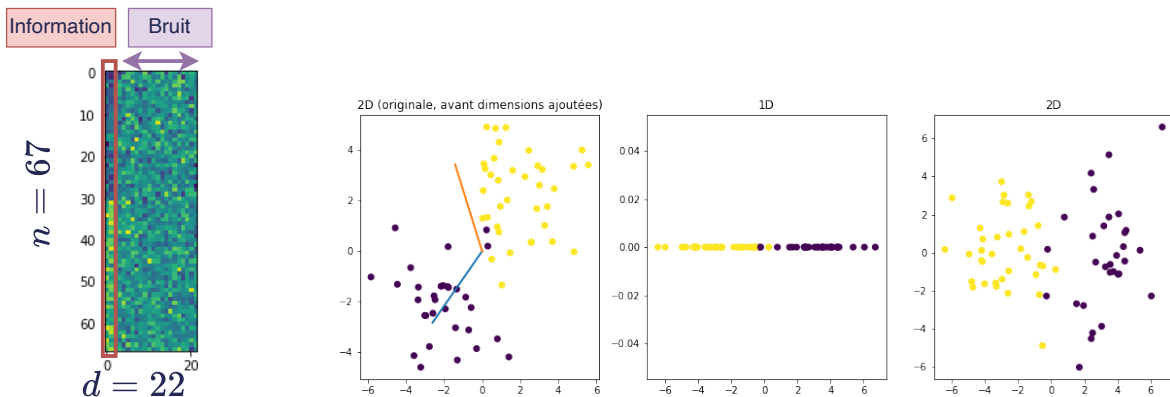


[Valeur propre / % de la variance expliquée par chaque val. p.]

(2) ACP/PCA : Analyse en composantes principales

Idée : trouver la combinaison d'axes expliquant au mieux la variance des données...
la valeur propre donne la *force* de l'explication

Tentons des projections :



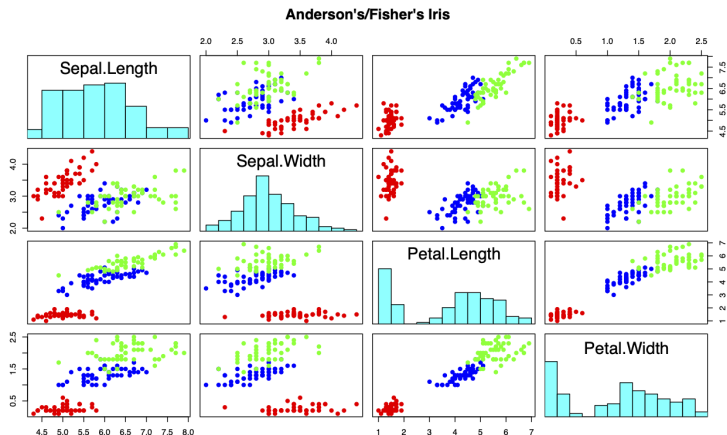
Les axes de la PCA sont orthogonaux... En 22 dimensions !



(2) ACP/PCA : un algo à tout faire

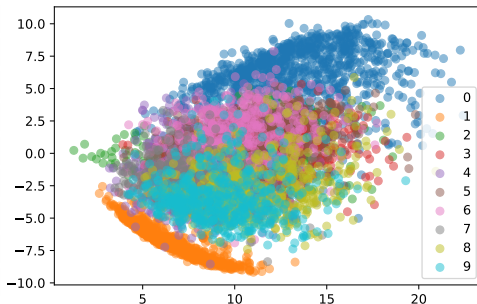
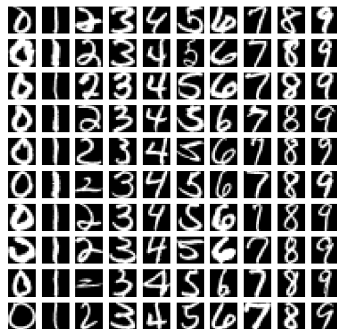
- Variantes pour les réseaux de capteurs
 - Combinaison linéaire de sources
- Visualisation de données

Iris : données 4D
Comment visualiser ?

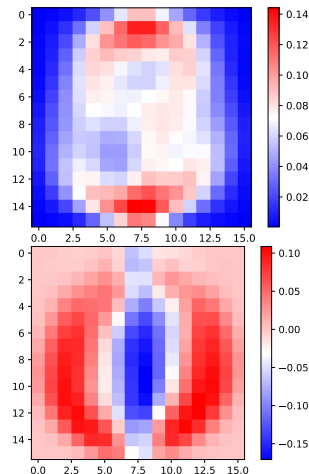


(2) ACP/PCA : un algo à tout faire

- Variantes pour les réseaux de capteurs
 - Combinaison linéaire de sources
- Visualisation de données



256 dimensions...



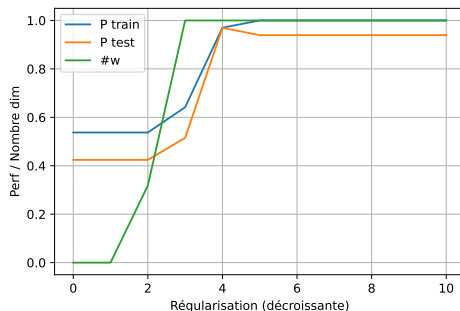
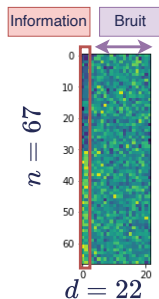
(3) Régularisation L2

Idee : apprendre les variables utiles automatiquement (pour une tâche donnée)

Formulation Ridge :

$$\mathcal{L} = \sum_{i=1}^n \left(\sum_{j=1}^d w_j x_{ij} - y_i \right)^2 + C \|w\|^2$$

- Cas limites : $C = 0$ moindres carrés, $C \rightarrow \infty \Rightarrow w = 0$
- Quid des cas intermédiaires ? [notion de chemin de régularisation]



⇒ Peu satisfaisant :
on utilise directement les 22
dimensions

(3) Régularisation L2

Idée : apprendre les variables utiles automatiquement (pour une tâche donnée)

Formulation Ridge :

$$\mathcal{L} = \sum_{i=1}^n \left(\sum_{j=1}^d w_j x_{ij} - y_i \right)^2 + C \|w\|^2$$

- Cas limites : $C = 0$ moindres carrés, $C \rightarrow \infty \Rightarrow \mathbf{w} = 0$
- Quid des cas intermédiaires ? [notion de chemin de régularisation]

Explications en revenant sur le gradient :

$$\nabla_1 = 2X^T(X\mathbf{w} - Y), \quad \text{MAJ : } \mathbf{w} \leftarrow \mathbf{w} - \varepsilon \nabla_1$$

$$\nabla_2 = 2\mathbf{w}, \quad \text{MAJ : } \mathbf{w} \leftarrow \mathbf{w} - \nabla_2 = \mathbf{w} \cdot (1 - 2\varepsilon) \quad \text{Affaiblissement des poids}$$

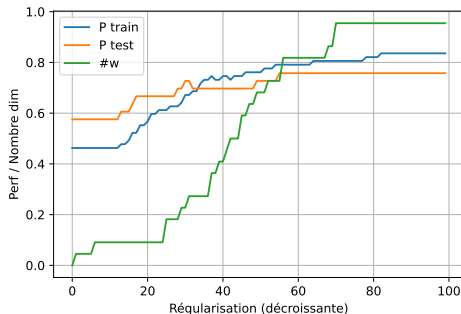
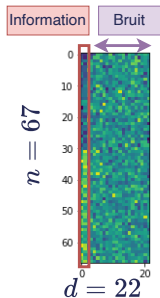
\neq annulation

(3) Régularisation L1

Idee : apprendre les variables utiles automatiquement (pour une tâche donnée)

LASSO :
$$\mathcal{L} = \sum_{i=1}^n \left(\sum_{j=1}^d w_j x_{ij} - y_i \right)^2 + C \|w\|_{\mathcal{L}_1} = \sum_{i=1}^n \left(\sum_{j=1}^d w_j x_{ij} - y_i \right)^2 + C \sum_{j=1}^d |w_j|$$

■ (Mêmes) cas limites : $C = 0$ moindres carrés, $C \rightarrow \infty \Rightarrow \mathbf{w} = 0$



\Rightarrow Bien sur les variables...
Perf. moins bonnes (difficile à régler...)

(3) Régularisation L1

Idée : apprendre les variables utiles automatiquement (pour une tâche donnée)

$$\text{LASSO : } \mathcal{L} = \sum_{i=1}^n \left(\sum_{j=1}^d w_j x_{ij} - y_i \right)^2 + C \|w\|_{\mathcal{L}_1} = \sum_{i=1}^n \left(\sum_{j=1}^d w_j x_{ij} - y_i \right)^2 + C \sum_{j=1}^d |w_j|$$

■ (Mêmes) cas limites : $C = 0$ moindres carrés, $C \rightarrow \infty \Rightarrow \mathbf{w} = 0$

Explications en revenant sur le gradient :

$$\nabla_1 = 2X^T(X\mathbf{w} - Y), \quad \text{MAJ : } \mathbf{w} \leftarrow \mathbf{w} - \varepsilon \nabla_1$$

$$\nabla_2 = \text{sign}(\mathbf{w}), \quad \text{MAJ : } \mathbf{w} \leftarrow \mathbf{w} - \varepsilon \cdot \text{sign}(\mathbf{w})$$

\Rightarrow On cherche bien l'annulation des poids

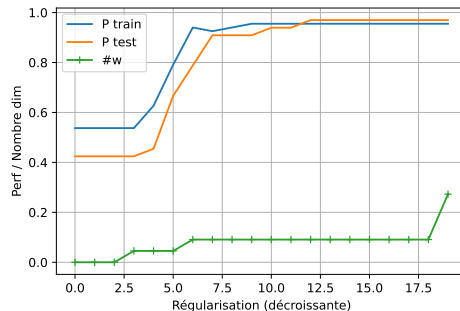
(3) Régularisation Elastic Net

Elastic net :

$$\mathcal{L} = \sum_{i=1}^n \left(\sum_{j=1}^d w_j x_{ij} - y_i \right)^2 + C_1 \sum_{j=1}^d |w_j| + C_2 \|\mathbf{w}\|^2$$

⇒ Combiner la régularisation L1 et L2 pour avoir le confort et les performances du L2 avec la sélection du L1

- Beaucoup plus dur d'explorer deux paramètres ...
- ... Mais en réalité, C_1 est epsilonesque et on ne joue presque que sur C_2





Apprentissage-Test \Rightarrow App/val/test

La multiplication des hyper-paramètres pose problème

Protocole dégradé :

- Pour tout les hyper-paramètres
(C , σ , early-stopping...)
 - Apprentissage (X_{app}, y_{app})
 - Evaluation (X_{test}, y_{test})

\Rightarrow Sur-apprentissage du jeu de test avec les hyper-paramètres...

Nouveau protocole :

- Pour tout les hyper-paramètres
(C , σ , early-stopping...)
 - Apprentissage (X_{app}, y_{app})
 - Evaluation (X_{val}, y_{val}) \Rightarrow Sélection de modèles
- Evaluation finale : (X_{test}, y_{test})