MA431 : Mathématiques appliquées à la sécurité

Bilan du cours : retour sur les tests d'intrusion

D. Barcelo

Grenoble INP ESISAR

2022/2023

Rappels sur l'ACP

Rappels sur l'ACP

Méthode

Méthode pour réaliser une ACP :

i) Vérifier que toutes les variables sont quantitatives.

ii) Calculer la matrice de variance-covariance, puis déterminer son spectre.

iii) Déterminer le nombre d'axes principaux que l'on conserve à l'aide de l'éboulis des valeurs propres.
Rappel : les axes principaux sont des combinaisons linéaires des axes initiaux (donc des variables initiales).

iv) Projeter les individus sur les composantes principales (technique de projection d'algèbre linéaire).

v) Représenter éventuellement le plan engendré par les deux plus grandes valeurs propres.

vi) Continuer à projeter si on souhaite avoir un pourcentage d'inertie restitué plus important

On appelle les coordonnées des projetés les **composantes principales**.

Pourcentage d'information restitué

Projeter implique accepter une déformation du nuage.
On sait que

- l'inertie du nuage est : $\mathcal{I} = tr(V)$,
- l'inertie du nuage projeté sur un sous-espace propre est : $\mathcal{I}_{\lambda_i} = \lambda_i$

On peut donc calculer la part d'inertie, absolue ou relative, restituée par la projection orthogonale :

- part d'inertie absolue : $\lambda_1 + \lambda_2$,
- part d'inertie relative : $\dfrac{\lambda_1 + \lambda_2}{\mathcal{I}}$.

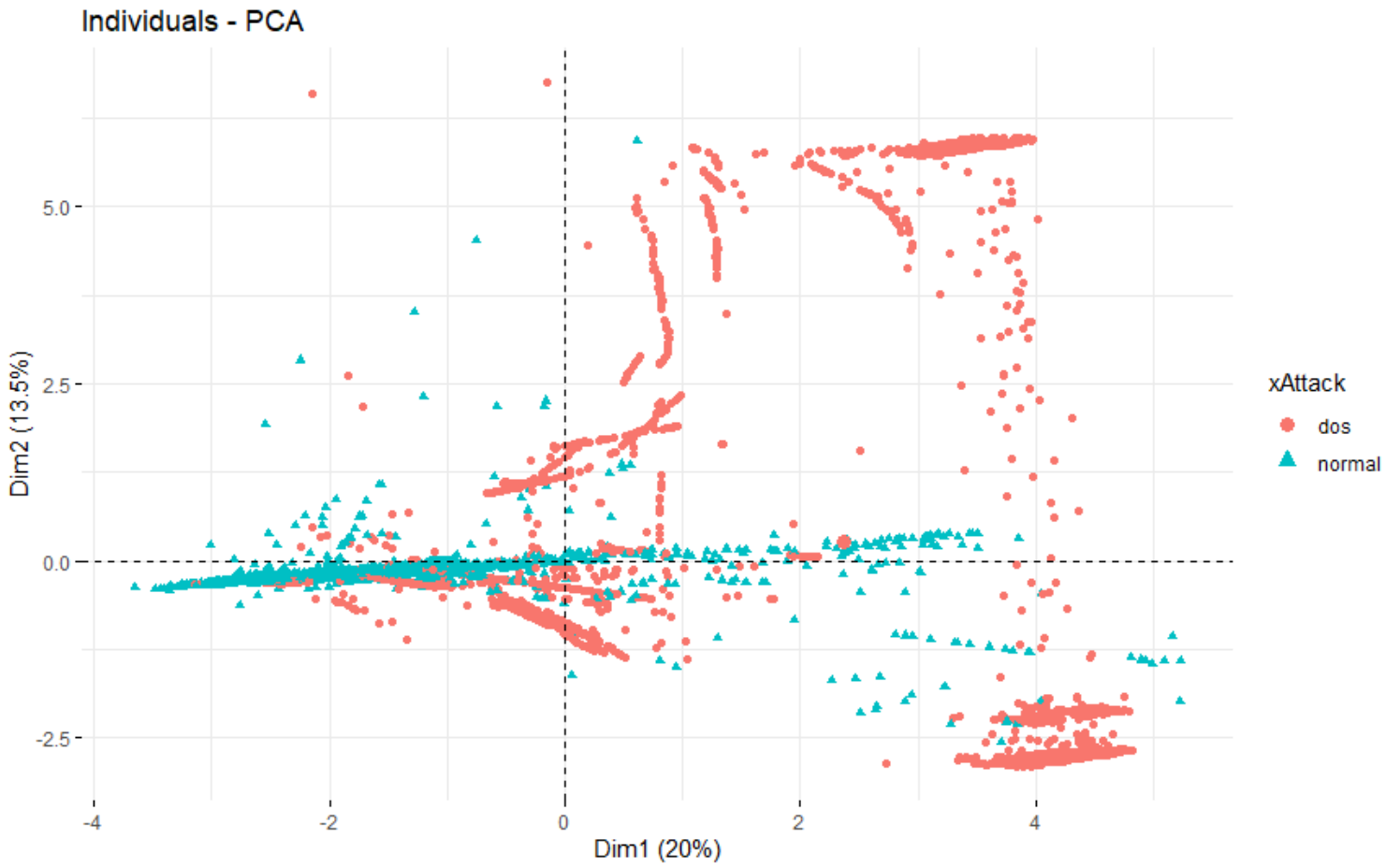
Qualité de la représentation graphique

- Lorsqu'on effectue une projection, deux individus éloignés dans l'espace peuvent avoir leurs projetés très proches.
- On mesure l'éloignement entre un point et son projeté sur un plan à l'aide de l'angle qu'ils forment dans l'espace ou à l'aide du cosinus de cet angle.
- Si on considère un individu $\vec{i_1}$ et $\vec{i_1'}$ son projeté orthogonal, on a alors :

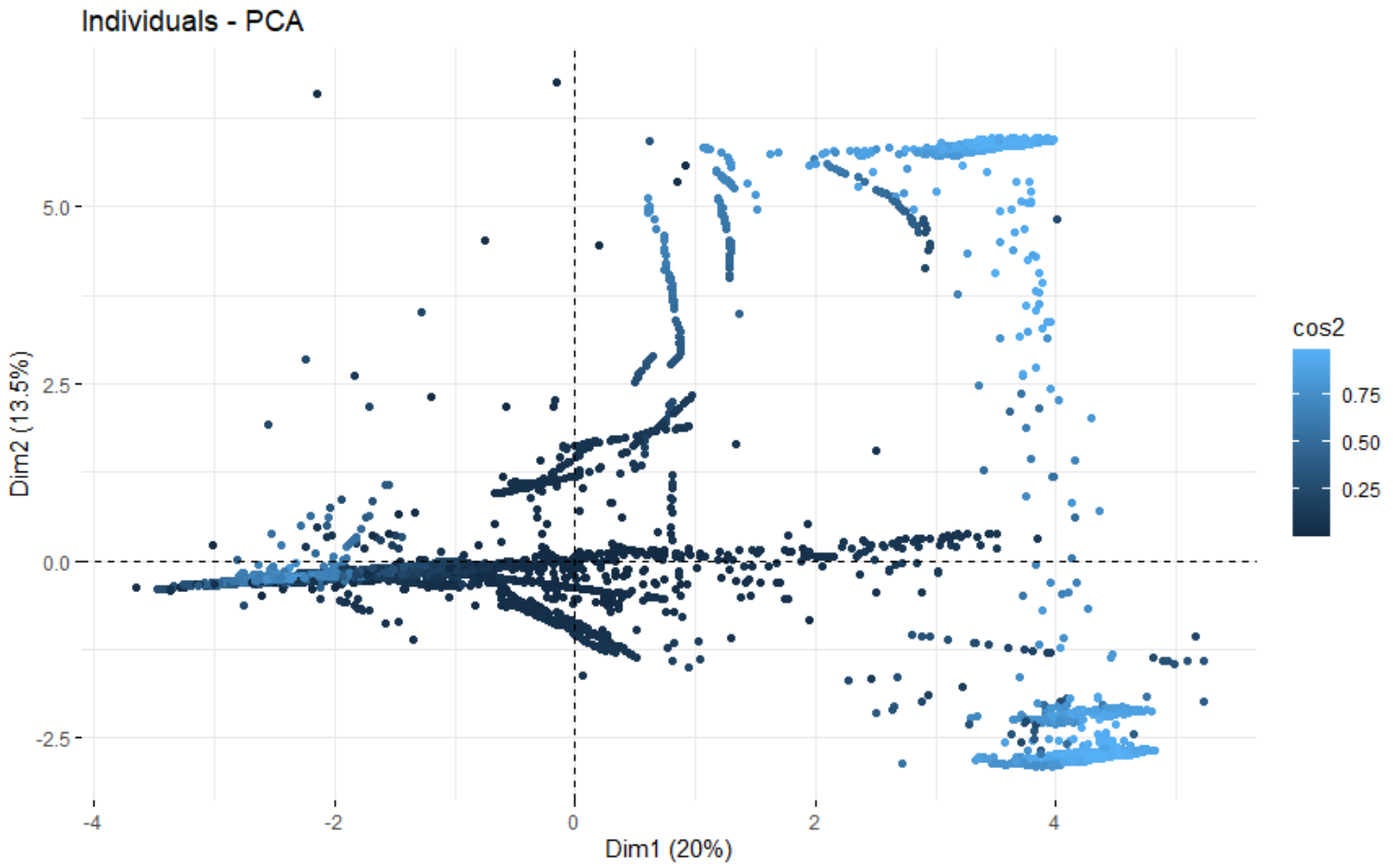
$$cos(\theta_1) = \frac{\vec{i_1}.\vec{i_1'}}{\left\|\vec{i_1}\right\| \times \left\|\vec{i_1'}\right\|} = \frac{\left\|\vec{i_1'}\right\|}{\left\|\vec{i_1}\right\|}$$

Généralement, on préfère calculer cos^2.

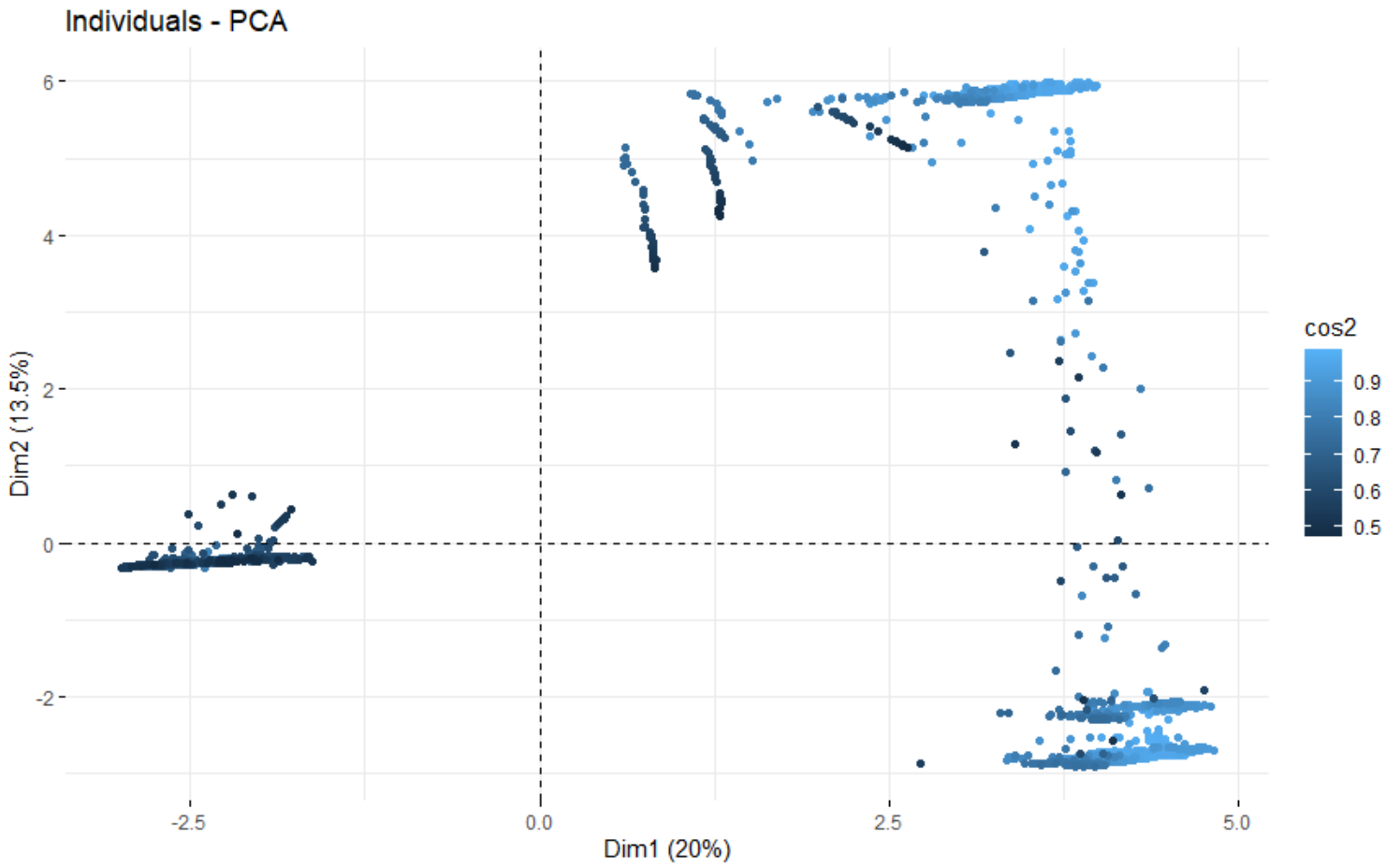
Exemple d'ACP : Traffic avec attaque DOS



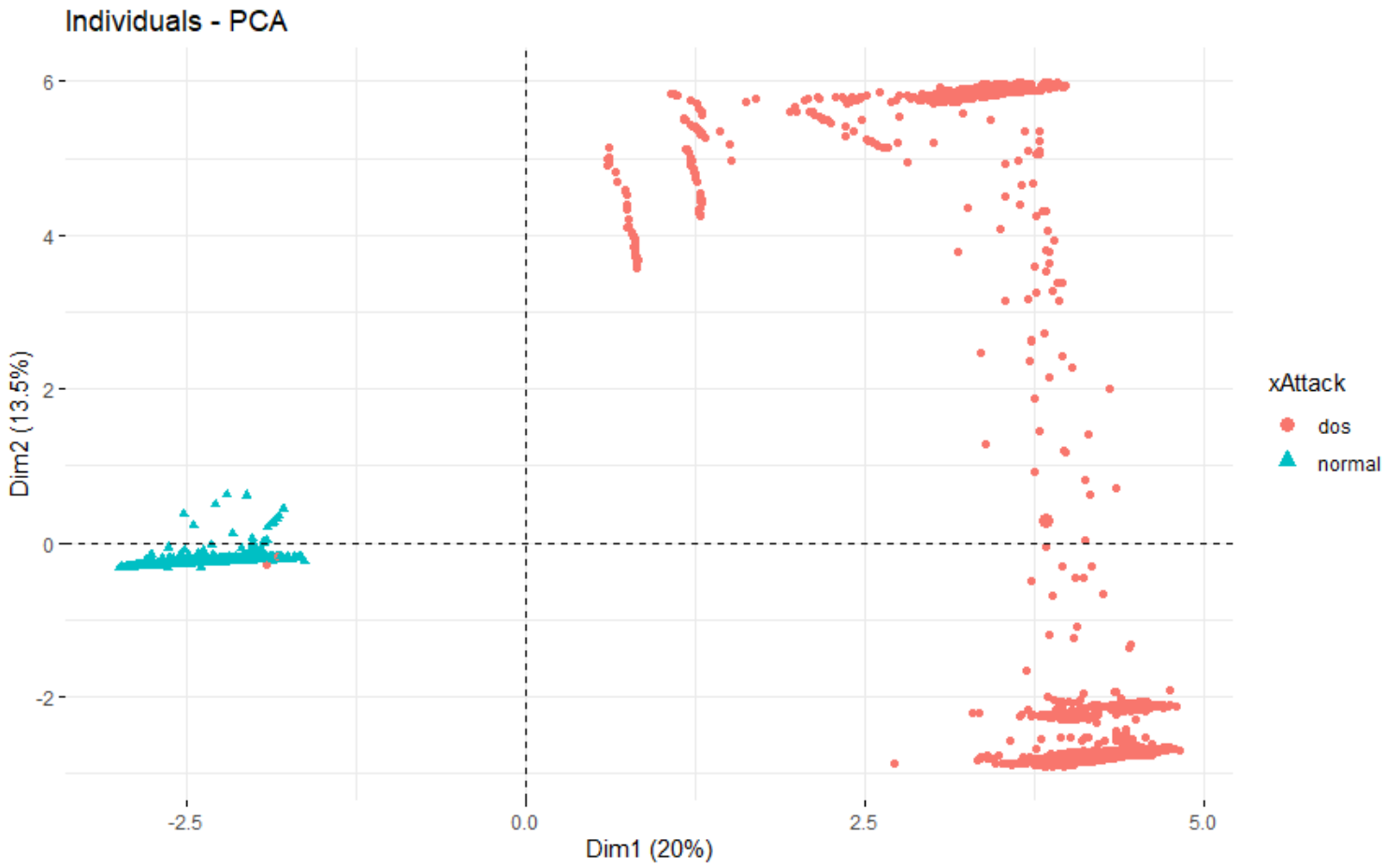
Exemple d'ACP : Traffic avec attaque DOS



Exemple d'ACP : Traffic avec attaque DOS



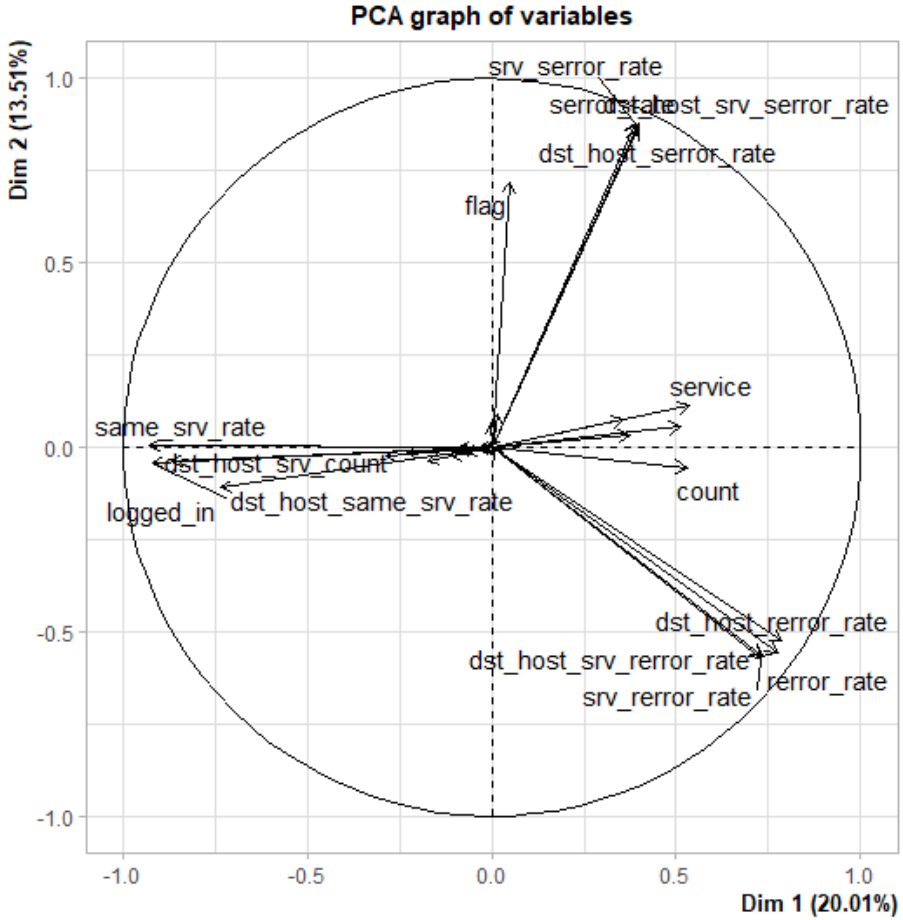
Exemple d'ACP : Traffic avec attaque DOS



Représentation graphique des variables

- On observe p variables quantitatives sur n individus.
- On peut donc considérer que les p variables représentent p vecteurs d'un espace des individus de dimension n. On note F cet espace.
- On peut appliquer la même méthode et projeter les vecteurs variables sur les sous-espaces propres de ${}^tX_0X_0$.
- Les deux matrices ${}^tX_0X_0$ et $X_0{}^tX_0$ ayant les mêmes valeurs propres, l'inertie expliquée par chaque axe principal sera la même que pour le nuage des individus.

Exemple d'ACP : Traffic avec attaque DOS



Compléments sur l'ACP

Compléments sur l'ACP

Individus supplémentaires

Individus supplémentaires

Objectif : enrichir une analyse, comparer les résultats obtenus à une base test, comparer le profil des nouveaux individus à des profils répertoriés etc. .
Méthode : ajouter des groupes d'individus étudiés ultérieurement.
Ajout d'individus : Il faut centrer les nouvelles données puis on peut les projeter sur le plan principal.

Variables supplémentaires

Variables supplémentaires

Objectif : Croiser les données avec de nouvelles variables, faire apparaitre une classification, etc..

Ajout de variables : On peut ajouter des variables quantitatives ou qualitatives transformées :

- **Variables quantitatives :** On centre puis on projette.
- **Variables qualitatives :** On ne peut pas fonctionner de la même manière. On ramène la variable à m groupes d'individus et on projette les centres de gravité de ces groupes dans le nuage des individus.

poids des individus

On peut également affecter un poids différent à chaque individu.

Dans une ACP classique, chaque individu a un poids de $\frac{1}{n}$.

Soit X_0 la matrice des données centrées. On a $V = \frac{1}{n}\,{}^t X_0 \times X_0$.

Ceci devrait s'interpréter par : $V = {}^t X_0 \times \frac{1}{n} I_n \times X_0$.

Pour changer le poids des individus, il suffit donc de ne pas conserver une matrice de produit scalaire qui soit l'identité mais une matrice diagonale que l'on multipliera par l'inverse de sa trace.

poids des variables

On peut aussi choisir de comparer des variables sans unités.
Il faut alors raisonner sur des données centrées et réduites.
Cela revient à donner le même poids à chaque variable.

Matrice des données centrées réduites

On note X_0 la matrice des données centrées.
On appelle matrice des données centrées réduites la matrice X_0* obtenue en multipliant à droite la matrice X_0 par la matrice $diag(\frac{1}{\sigma(x_1)}, \ldots, \frac{1}{\sigma(x_p)})$.

$$X_0* = \begin{pmatrix} \frac{x_{1,1} - \bar{x_1}}{\sigma_{x_1}} & \ldots & \frac{x_{1,p} - \bar{x_p}}{\sigma_{x_p}} \\ \frac{x_{2,1} - \bar{x_1}}{\sigma_{x_1}} & \ldots & \frac{x_{2,p} - \bar{x_p}}{\sigma_{x_p}} \\ \ldots & \ldots & \ldots \\ \frac{x_{n,1} - \bar{x_1}}{\sigma_{x_1}} & \ldots & \frac{x_{n,p} - \bar{x_p}}{\sigma_{x_p}} \end{pmatrix}.$$

Matrice des corrélations

Matrice des corrélations

Soit X_0* la matrice des données centrées. On pose $R = \frac{1}{n}{}^t X_0 * \times X_0*$. On a alors :

$$R = \begin{pmatrix} 1 & r_{x_1x_2} & \dots & r_{x_1x_p} \\ r_{x_2x_1} & 1 & \dots & r_{x_2x_p} \\ \dots & \dots & \dots & \dots \\ r_{x_px_1} & r_{x_px_2} & \dots & 1 \end{pmatrix}.$$

ACP normée

ACP normée

On applique la même méthode d'ACP mais sur la matrice des corrélations R et non sur la matrice de variance-covariance.

- détermination du spectre de R,
- projection sur les sous-espaces propres engendrés par les plus grandes valeurs propres.
- inertie du nuage : $\mathcal{I} = tr(R)$ donc $\mathcal{I} = p$ nombre de variables.

ACP normée

Cercle des corrélations :

On va projeter les vecteurs variables normés de $\mathbb{R}^n$ sur un plan de $\mathbb{R}^n$ défini par les deux composantes principales calculées lors de la projection du nuage des individus C_1 et C_2.
Soit $i \in [\![1; p]\!]$ et $j \in [\![1; 2]\!]$

$$p_{\text{Axe}j}(\vec{x_i}) = \frac{\vec{x_i}.\vec{C_j}}{\|\vec{x_i}\| \left\|\vec{C_j}\right\|}$$

Ceci revient donc à projeter dans un cercle et à calculer les corrélations entre chaque variable et chaque composante principale.

Remarque sur le coefficient de corrélation

Calcul du coefficient de corrélation

$$r_{x_1,x_2} = \frac{Cov(x_1; x_2)}{\sigma(x_1)\sigma(x_2)}$$

$$r_{x_1,x_2} = \frac{\vec{x_1}.\vec{x_2}}{\sqrt{\|\vec{x_1}\|^2}\sqrt{\|\vec{x_2}\|^2}}$$

$$r_{x_1,x_2} = cos\theta_{\vec{x_1},\vec{x_2}}$$

On peut donc en déduire que deux variables sont proches dans $\mathbb{R}^n$ si l'angle entre les deux variables est proche de 0, donc leur cosinus proche de 1, soit un coefficient de corrélation proche de 1.

Le cercle des corrélations

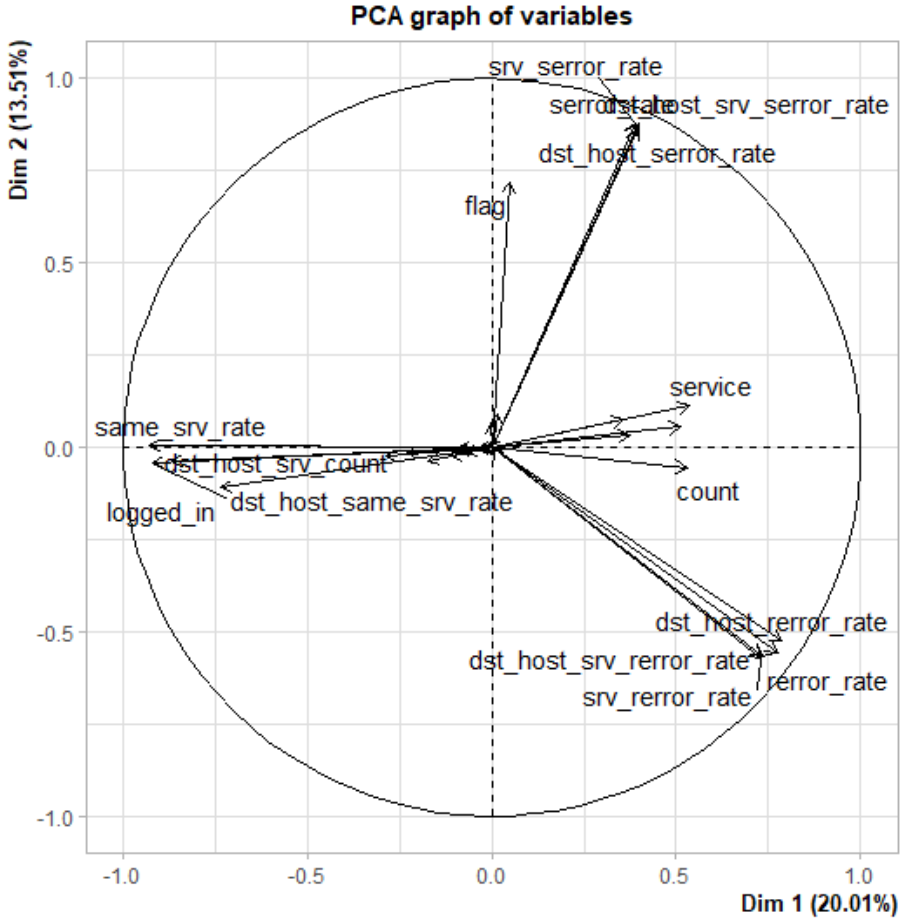
Cercle des corrélations

La représentation graphique des projections est appelée cercle des corrélations. Il permet une interprétation rapide des composantes principales et aide à repérer les variables liées ou opposées.

Interprétation

- On n'interprète que les variables qui sont corrélées avec les composantes principales, donc proches du cercle.
- Les groupes de variables permettent d'interpréter les axes principaux.
- Si deux variables ne sont pas corrélées, les vecteurs projetés sont orthogonaux dans le cercle.
- Deux variables corrélées sont proches si la corrélation est positive et opposées si la corrélation est négative.

Exemple d'ACP : Traffic avec attaque DOS

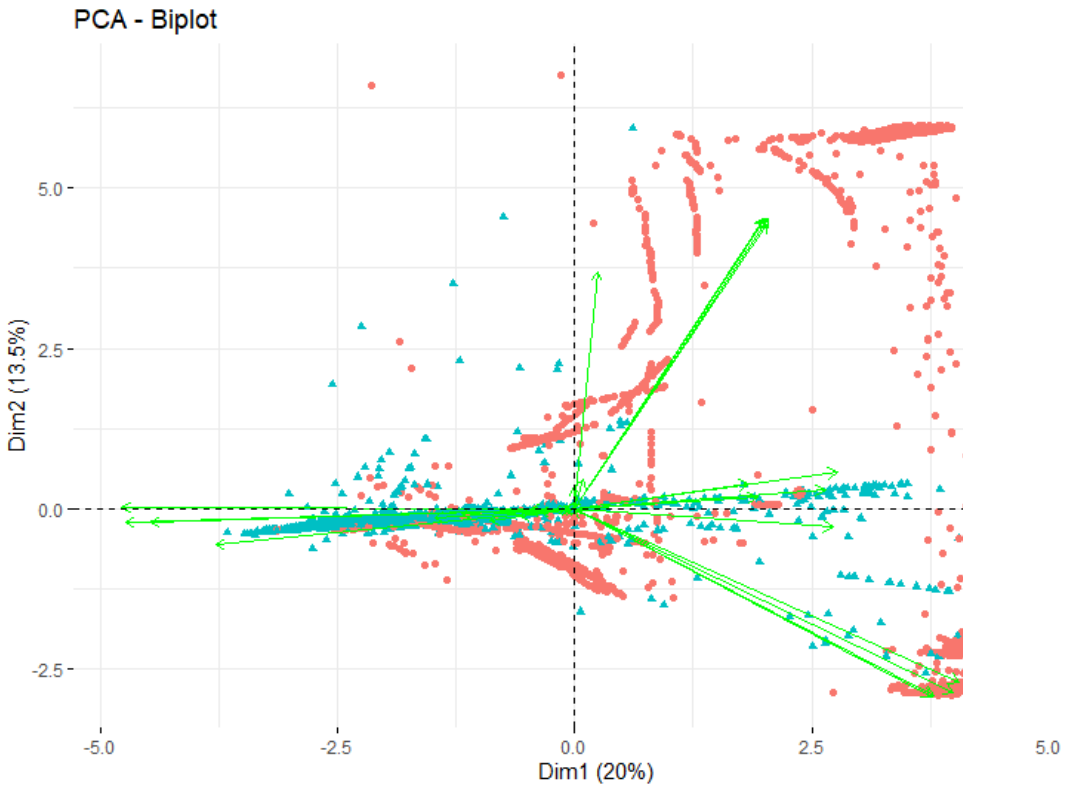


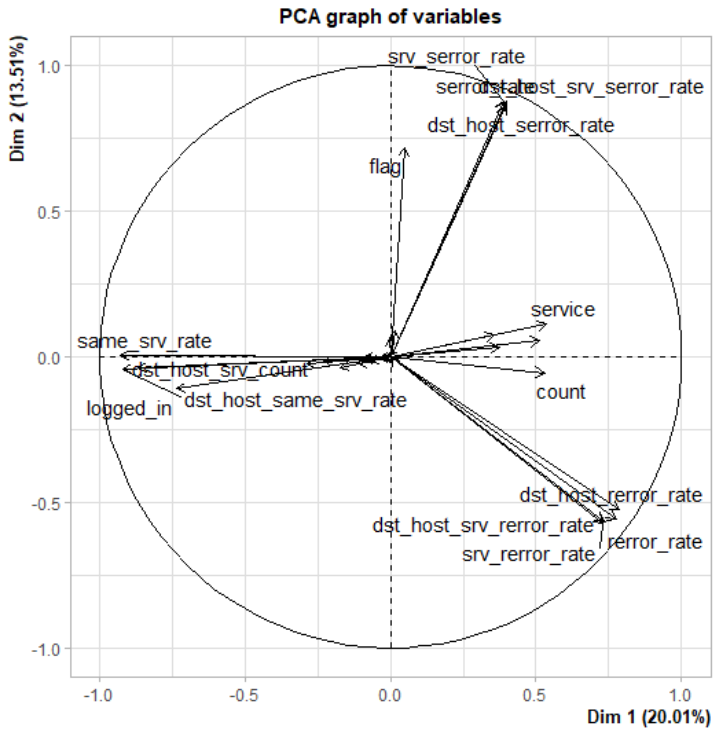
Représentation simultanée

Les analyses des nuages des variables et des individus se complètent. Les interprétations des axes dans chaque espace permettent de mieux comprendre la représentation globale.

- Attention : les deux nuages ne sont pas dans le même espace ! (E et F de dimensions respectives p et n)
- En théorie : impossibilité d'une représentation simultanée.
- On peut en envisager une si on ne considère plus les vecteurs comme des variables mais comme les directions de ces variables.
- En fait, on "écrase" les axes du repère orthonormé de l'espace des variables dans le plan principal.
- On peut alors comparer le positionnement d'un groupe d'individus vis-à-vis des variables et réciproquement

Exemple d'ACP : Traffic avec attaque DOS



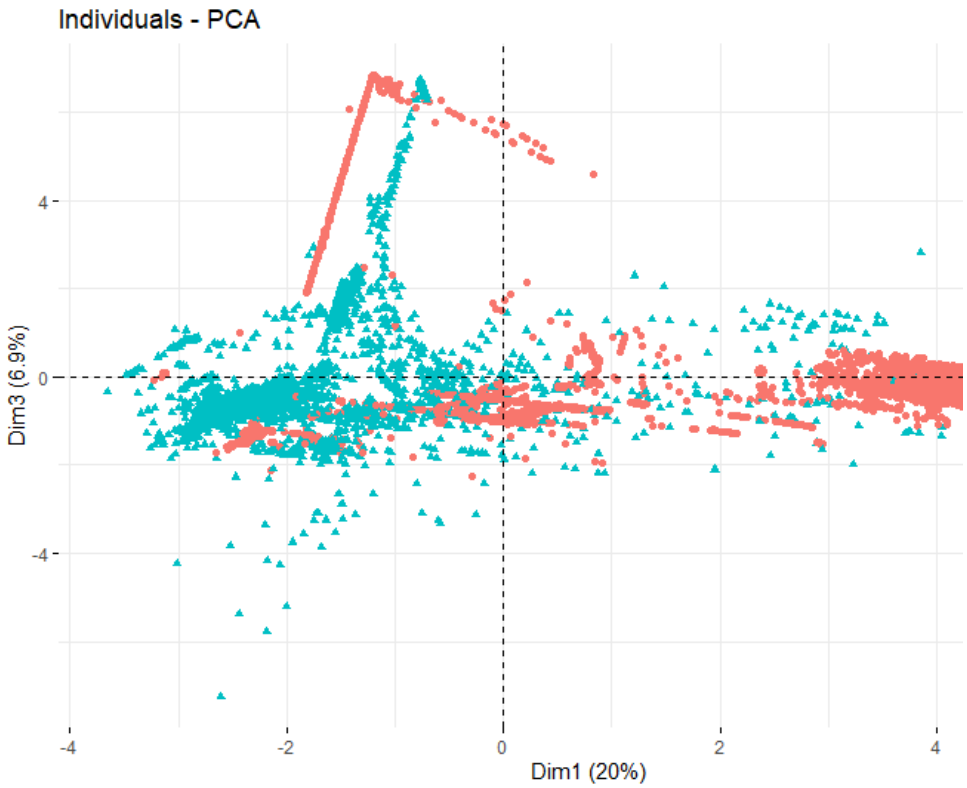


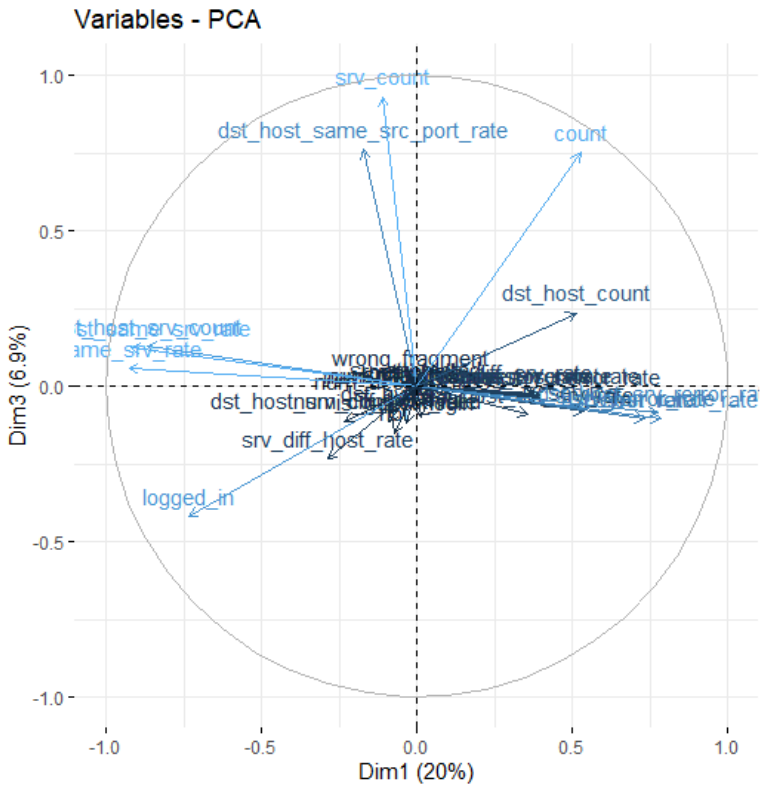
Autres plans principaux

On peut envisager de représenter les projections sur d'autres plans principaux pour mieux interpréter les résultats.

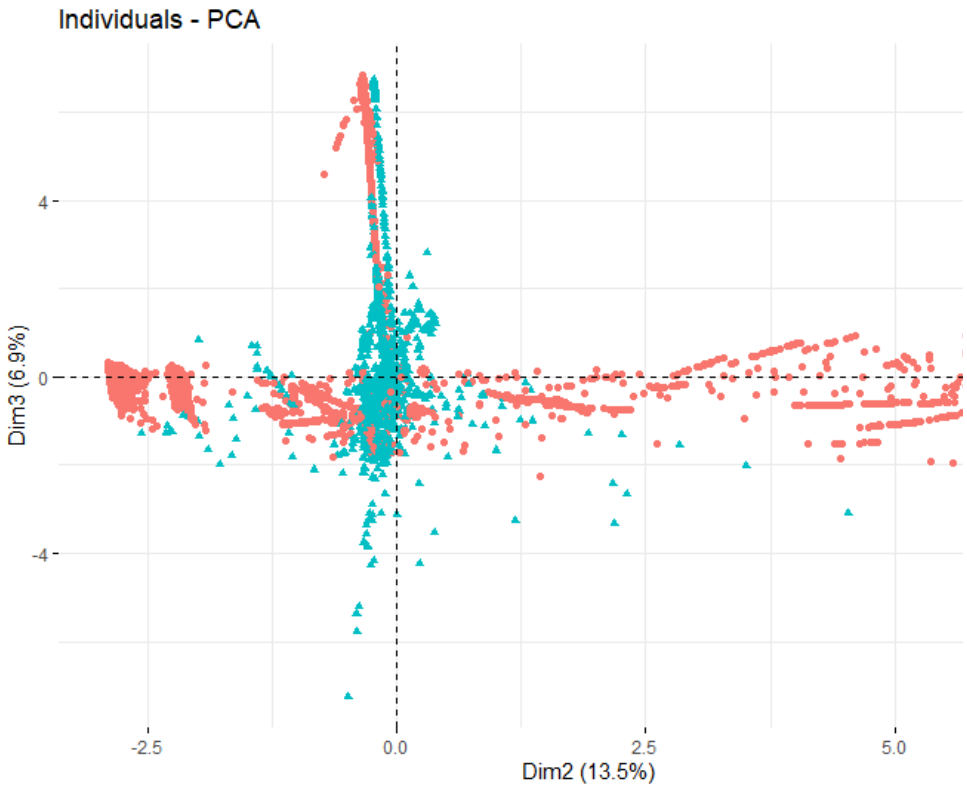
- La technique reste la même.
- On projette sur les axes engendrés par la plus grande valeur propre et par la troisième plus grande valeur propre de la matrice de variance-covariance.
- L'intérêt de cette projection supplémentaire est liée à l'ébouli des valeurs propres.
- Cela peut permettre de mieux interpréter des positionnements d'individus.

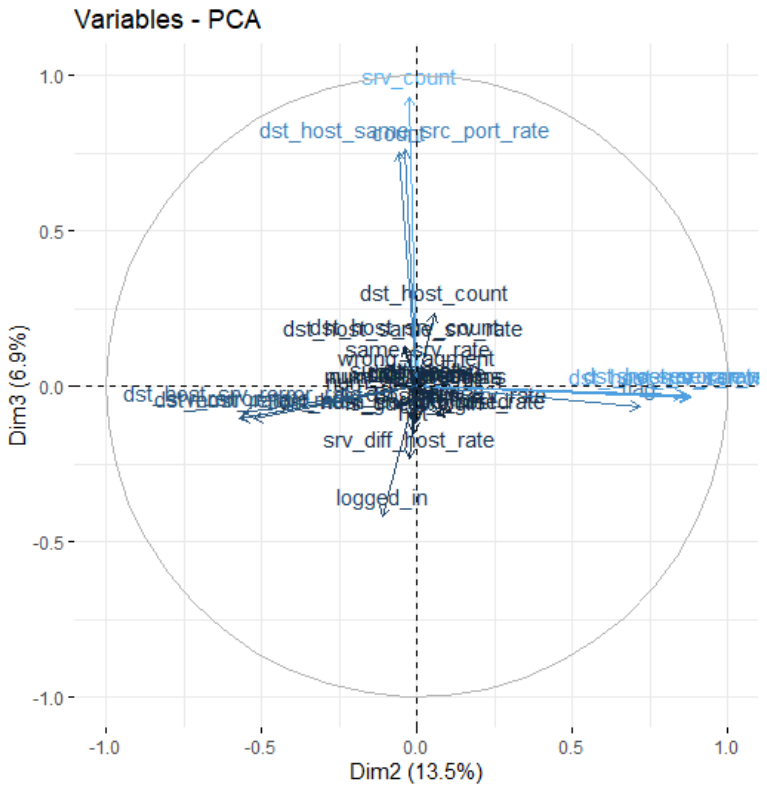
Autres plans principaux





Autres plans principaux





Applications

Applications

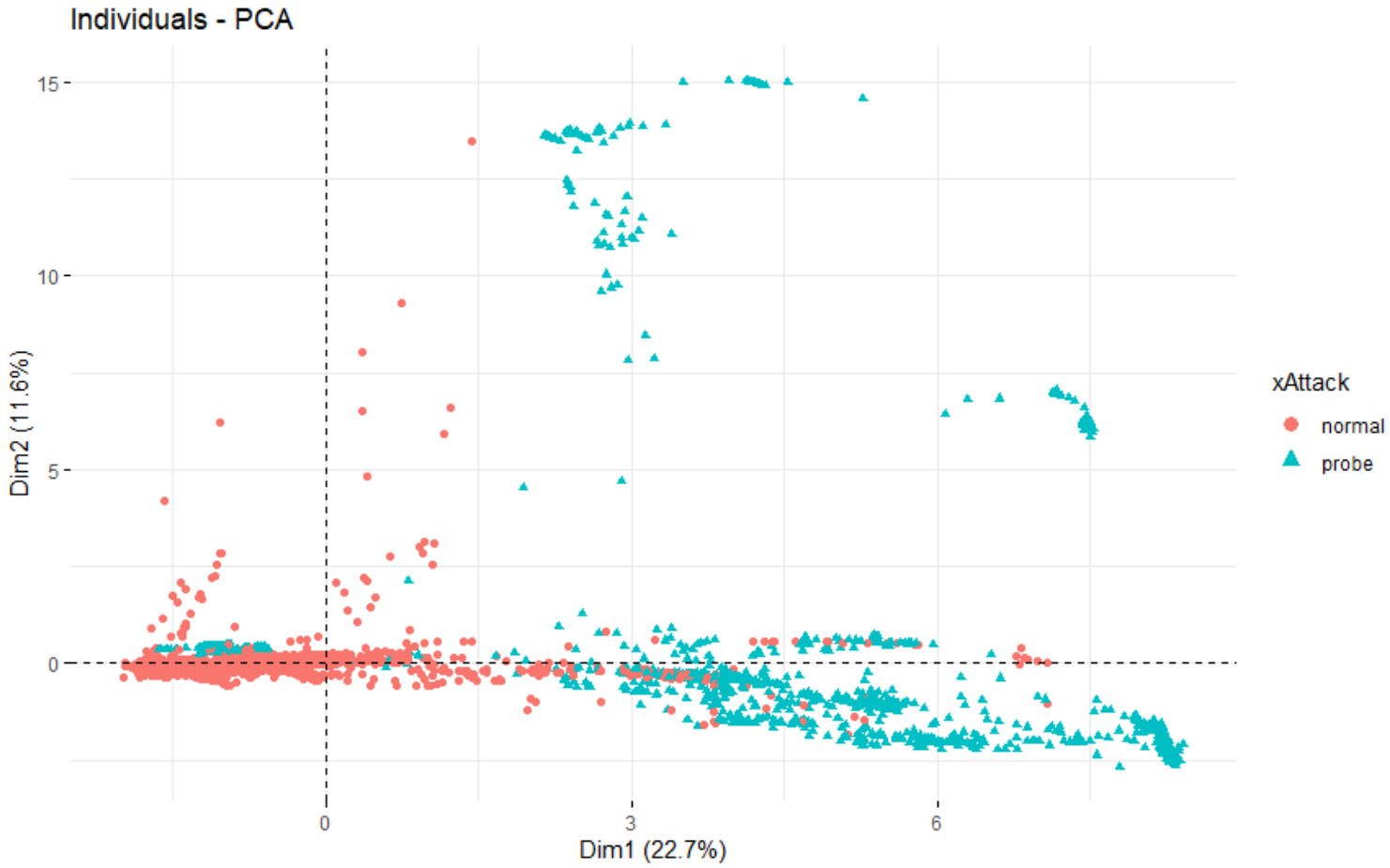
Amélioration des algorithmes

- algorithme de classement lourds en calculs
- utilisation de l'ACP pour réduire la dimension
- application de l'algorithme de classement avec moins de données

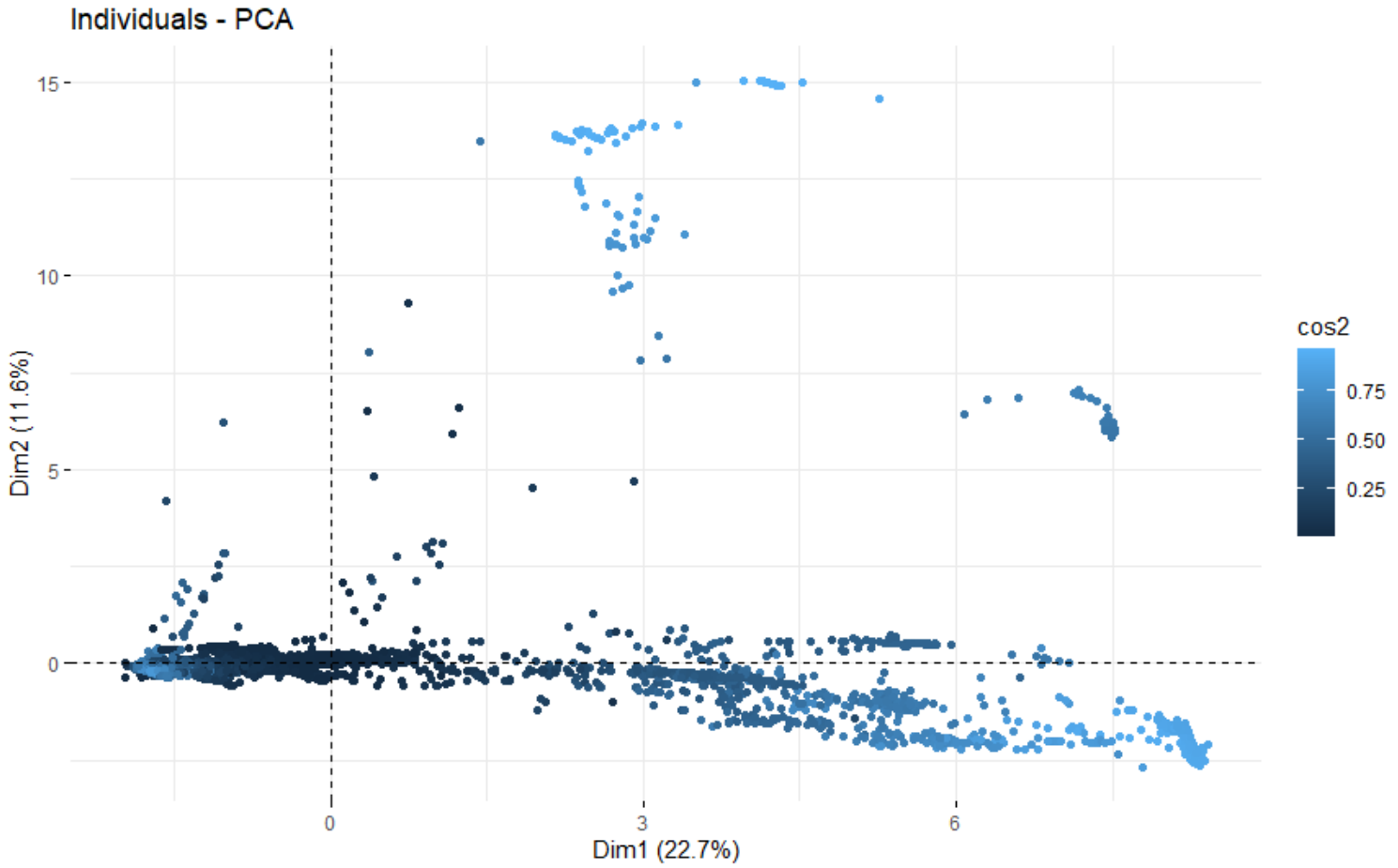
En clustering

- représentation graphique permet une vérification visuelle du résultat d'un clustering
- aide au choix du nombre de clusters
- isole individus aberrants
- permet de repérer les individus représentatifs d'un cluster et ceux proches d'un cluster voisin.

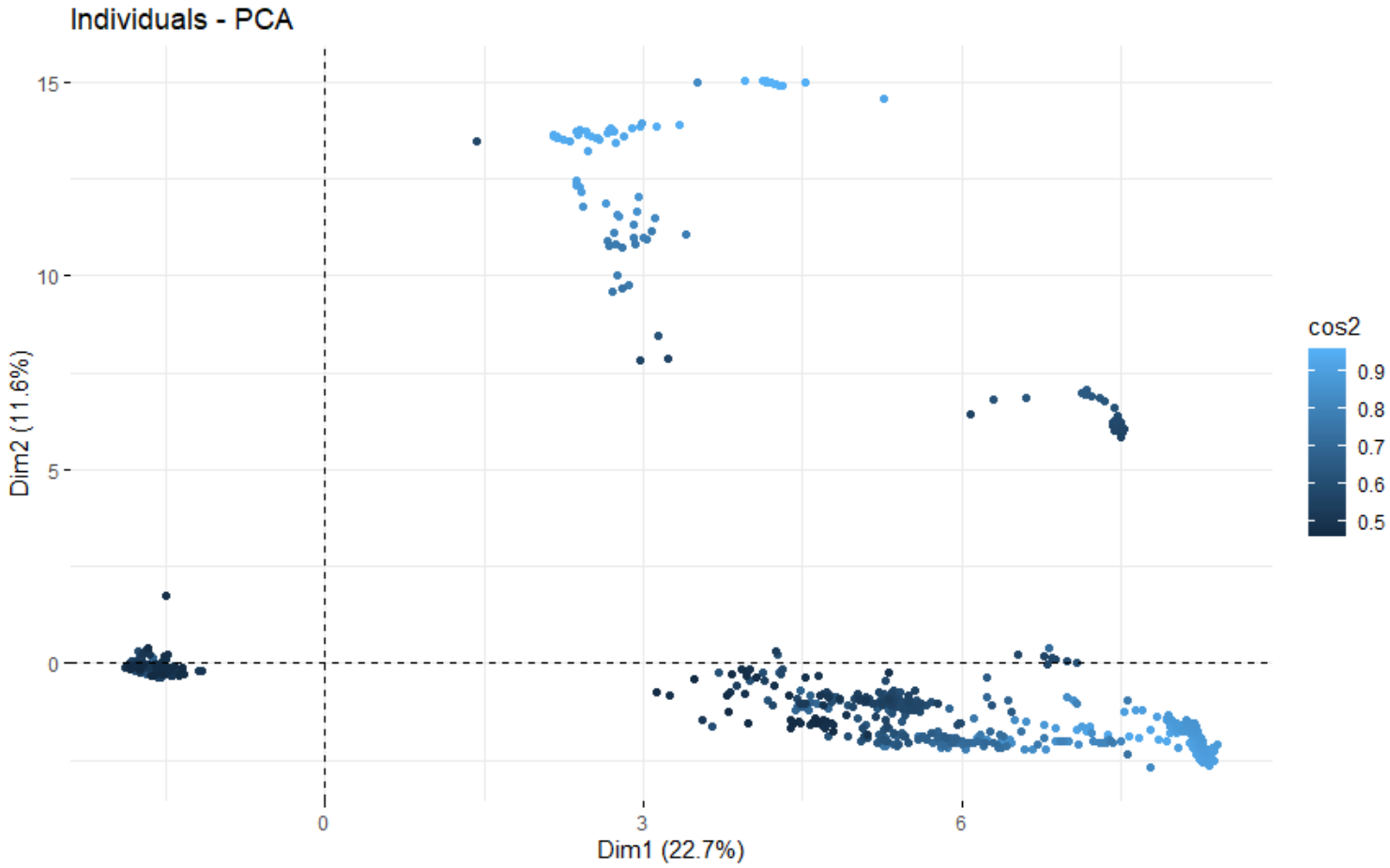
Test d'intrusion Attaque Probe



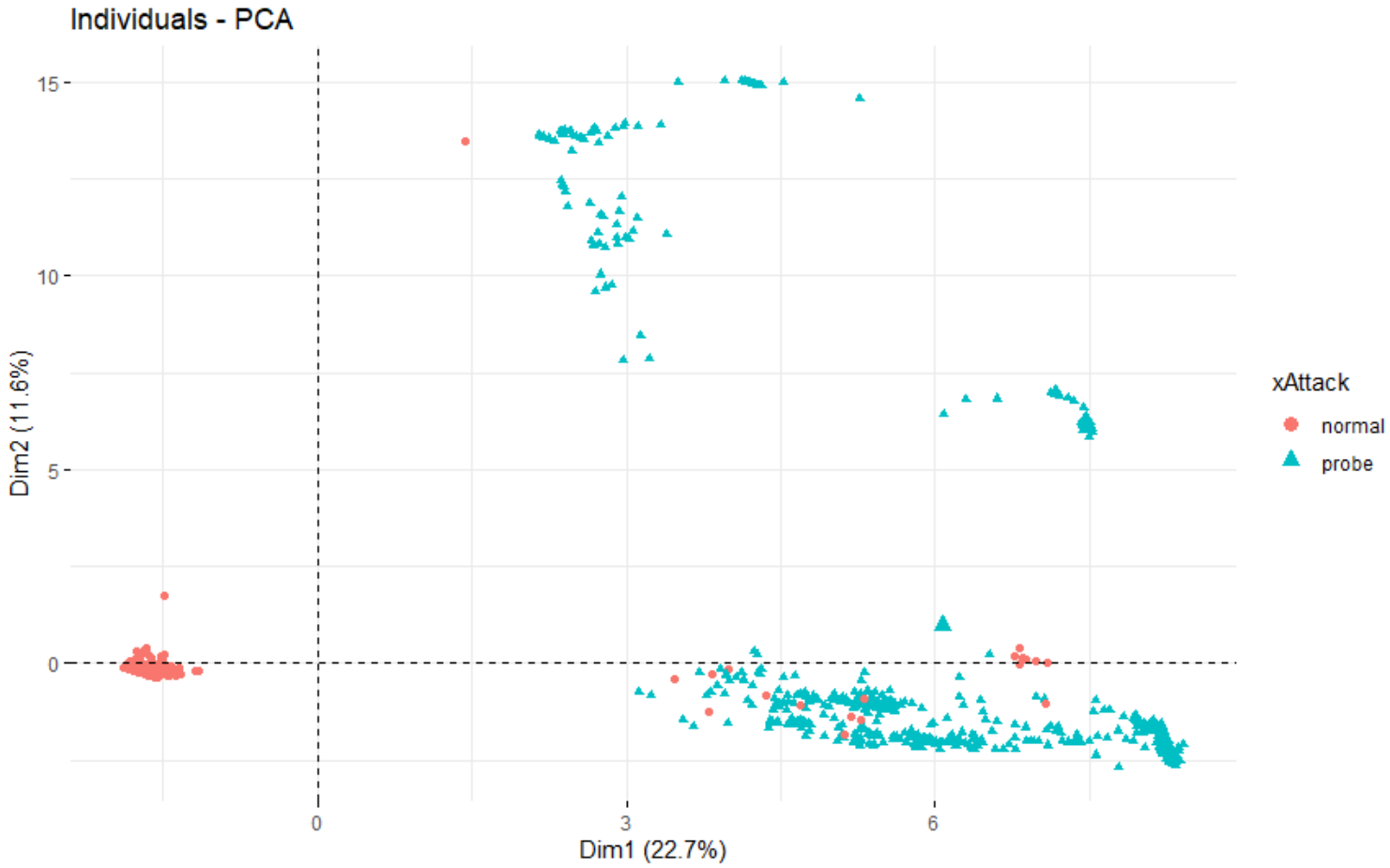
Test d'intrusion Attaque Probe



Test d'intrusion Attaque Probe



Test d'intrusion Attaque Probe



Eigenfaces

L'ACP peut être utilisée pour des problèmes de reconnaissances faciales. La méthode étant basée sur les éléments propres d'une matrice, on lui donne le nom de Eigenfaces.

- On transforme une base test de n images faciales (visages de plusieurs personnes ou d'une même personne) en n vecteurs de dimension k^2 (en fonction du nombre de pixels).
- On applique une ACP au nuage des vecteurs faces. *On a donc soustrait un visage moyen à chaque visage pour ne garder que ce qui le distingue des autres ! !*
- On choisit la dimension m de l'espace sur lequel on projette.
- On détermine les axes principaux et les vecteurs propres associés. *On parle donc de visages qui sont des vecteurs propres*

Eigenfaces

- Les n vecteurs faces de la base test sont projetés dans l'espace propres.
- Les coordonnées des projetés dans l'espace principal permettent d'obtenir ainsi un vecteur "poids" de la face projetée.
- Fin de la phase d'entrainement
- Pour reconnaitre un nouveau visage, on le projette sur l'espace principal et on calcule sa distance à chaque projeté.
- Si la nouvelle face projetée est assez proche d'une des autres faces on considère qu'elle appartient à la base test. On l'a reconnue !
- La partie entrainement sur une base test est assez lourde et assez longue mais la partie reconnaissance est simple et rapide.

Eigenfaces

EigenSpams et EigenViruses

Sur le même principe que les Eigenfaces, on peut reconnaitre des images SPAMS et des malwares.

- Pour les EigenSpams, il suffit de disposer d'une base test assez conséquente d'images Spams et d'images non spams.
- Pour les Eigenviruses, la base test est composée de malwares et d'autres fichiers dont on extrait une séquence du code du fichier exécutable pour former un vecteur.

EigenSpams


Merry X-mas!
CLICK HERE TO SEE OUR WATCHES!
RPBAZAAR
Christmas Special!
Most Popular
Click Here For More Meds

Retour sur la classification

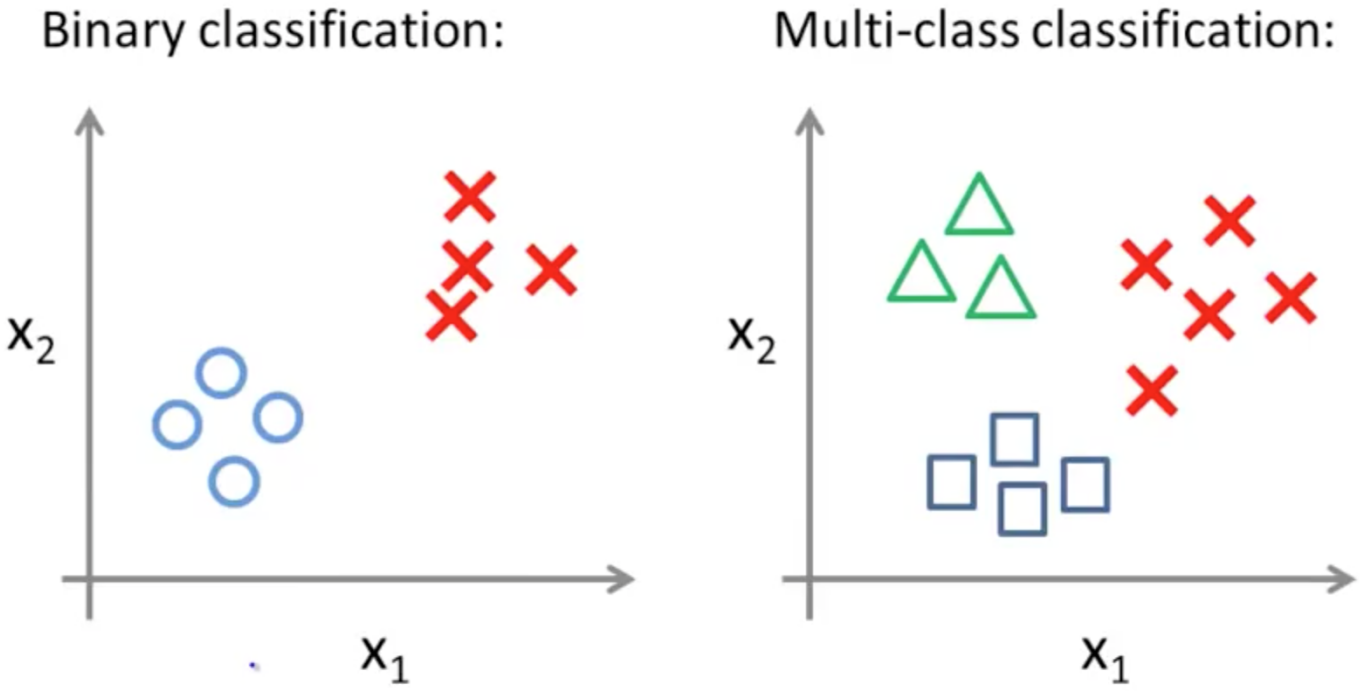
La classification multi-classes

Classification multi-classes

En général, les algorithmes de classification sont de deux types :

- les algorithmes de classification binaire (uniquement deux classes possibles, ex : Traffic normal / Attaque par DOS).
- les algorithmes de classification multi-classes (au moins deux classes possibles, ex : traffic norma / Attaque par DOS / Attaque par Probe).

Multi-classes



Classification multiclasse

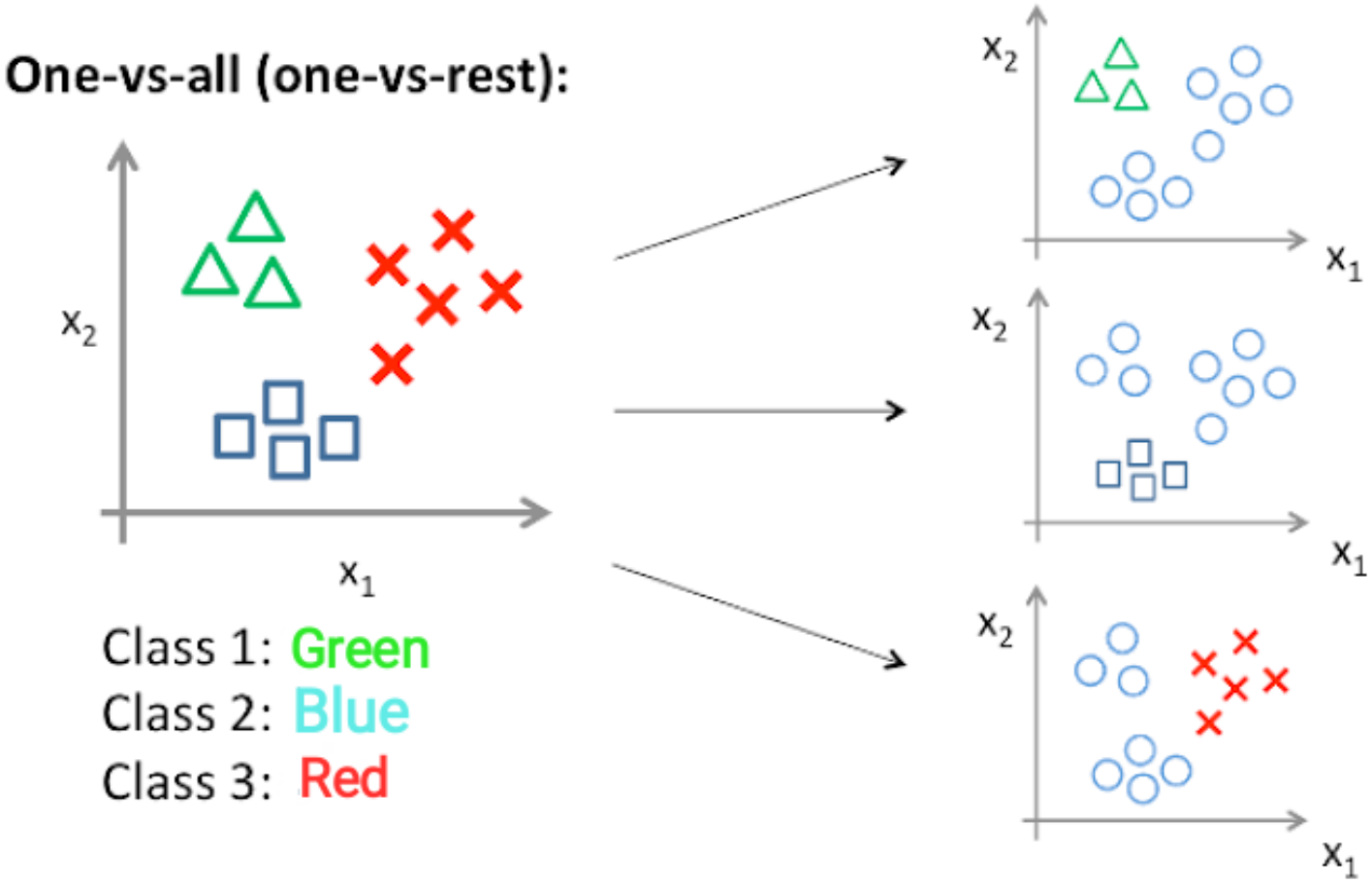
Les algorithmes de classification que nous avons étudiés proposent-ils tous une classification binaire ?

One vs Rest

One versus Rest

On considère une classification multi-classes en k classes.
La méthode **One versus Rest (OVR)**, aussi appelée **One versus All (OVA)**, consiste à résoudre le problème de classification multi-classes en décomposant le problème en k classifications binaires.
Chacune de ces classifications permet de reconnaitre une classe sans distinguer les autres entre elles On obtient alors k classifieurs f_i avec $i \in [\![1, k]\!]$.
On construit ensuite la fonction de décision : $f(x) = argmax_k f_k(x)$.

OvR


One-vs-all (one-vs-rest):
x2
x1
Class 1: Green
Class 2: Blue
Class 3: Red
x2
x1
x2
x1
x2
x1

One vs Rest

Exemple :
On construit une classification multi-classes à l'aide d'une régression logistique :

- On observe du traffic normal, des attaques par DOS, des attaques par rdl et des attaques par Probe.
- On a donc quatre classifieurs à créer :
  f_1 pour le traffic normal,
  f_2 pour les attaques par DOS,
  f_3 pour les attaques par rdl,
  f_4 pour les attaques par Probe.par Probe.
- On utilise les trois classifieurs sur une nouvelle observation x.
  On obtient $f_1(x) = 0,3$, $f_2(x) = 0,5$, $f_3(x) = 0,1$ et $f_4(x) = 0,4$.
- Que décide-t-on ? Pourquoi ?

One vs Rest

Inconvénients

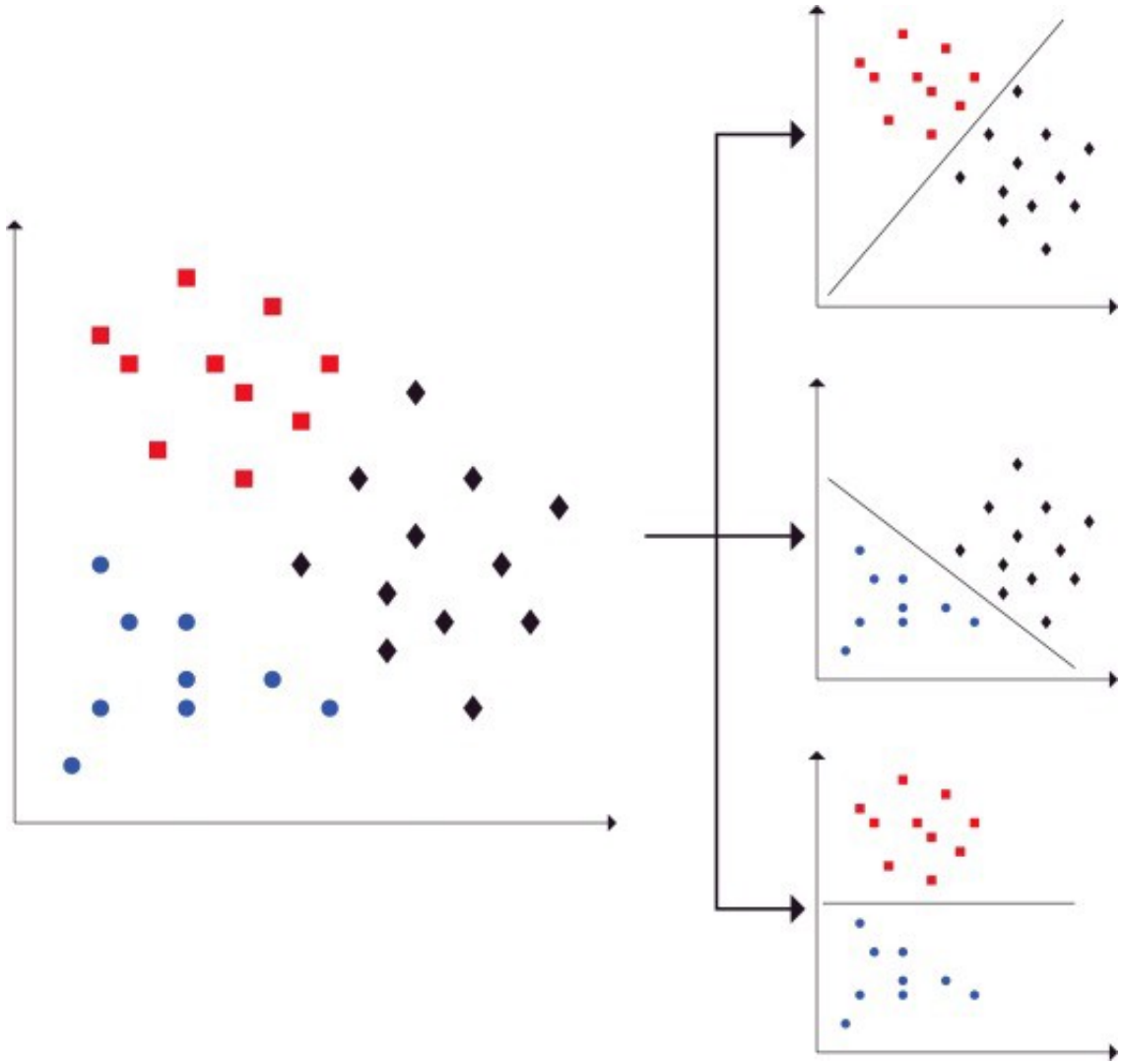
Il faut créer autant de classifieurs que de classes !
Problème de ralentissement potentiel du processus, sauf pour un petit nombre de classes.
Complexité algorithmique : $\mathcal{O}(kC)$ avec C complexité du classifieur binaire.

One vs One

One versus One

On considère une classification multi-classes en k classes.
La méthode **One versus One (OVO)** consiste à résoudre le problème de classification multi-classes en décomposant le problème en $\frac{k(k-1)}{2}$ classifications binaires.
Chaque classification On obtient alors $\frac{k(k-1)}{2}$ classifieurs f_i avec $i \in [\![1, k]\!]$.
On construit ensuite la fonction de décision par un vote à la majorité. La classe affectée sera celle prédite par le plus grand nombre de classifieurs.

One vs One



One vs One

Exemple :
On construit une classification multi-classes à l'aide d'une régression logistique :

- On observe du traffic normal, des attaques par DOS et des attaques par Probe.
- On a donc six classifieurs à créer :
  f_1 pour normal vs DOS, f_2 pour normal vs rlt, f_3 pour normal vs Probe, f_4 pour DOS vs rlt, f_5 pour DOS vs Probe et f_6 pour rlt vs Probe.
- On utilise les six classifieurs sur une nouvelle observation x.
  On obtient $f_1(x) =$DOS, $f_2(x) =$normal, $f_3(x) =$Probe, $f_4(x) =$DOS, $f_5(x) =$DOS, et $f_6(x) =$Probe.
- Que décide-t-on ? Pourquoi ?

One vs One

Inconvénients

Il faut créer autant plus de classifieurs que de classes !
En revanche, on entraine les classifieurs sur des données moins importantes.

Les tests d'intrusion

Les tests d'intrusion

Tests d'intrusion

- Détection des intrusions par analyse du traffic,
- Utilisation de modèles issus du machine learning,
- Détection des menaces connues,
- Détecter des nouvelles menaces (zero-day attacks, inconnues jusqu'à l'attaque).

Signatures et anomalies

Approche par signature

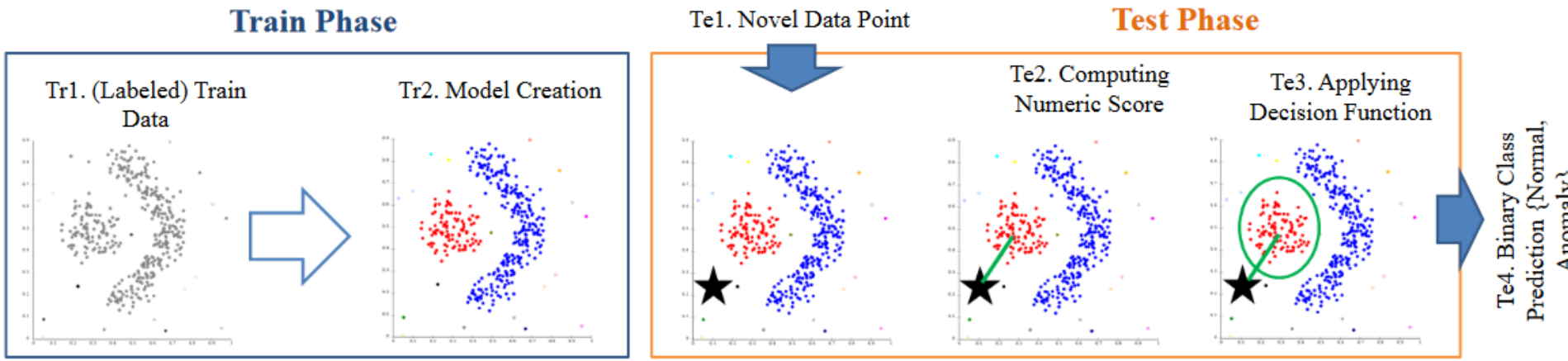
- Principe des antivirus,
- les attaques connues sont analysées et étiquetées,
- Apprentissage supervisé pour apprendre à les reconnaitre,
- Problème pour reconnaitre les variations des attaques connues ou des nouvelles attaques.

Signatures et anomalies

Approche par anomalies

- Détection d'activités suspectes,
- algorithme basé sur l'étude du trafic passé,
- classe tout nouveau point soit en trafic normal soit en anomalie (donc attaque potentielle).

Approche par anomalies



Apprentissage supervisé ou non supervisé

	Attaque connue	Attaque inconnue
Algorithme Supervisé	Très bon	Pas très bon
Algorithme non supervisé	Plutôt Bon	Plutôt Bon

Apprentissage supervisé ou non supervisé

- Apprentissage supervisé très efficace pour détecter les menaces connues,
- mais beaucoup moins d'efficacité pour détecter les nouvelles menaces (zero-day attacks, inconnues lors de la phase d'apprentissage),
- Apprentissage non supervisé et clustering parvient plus facilement à détecter les zero-day attacks.
- Idée : combiner les deux méthodes pour obtenir une meilleure protection.

Modèle mixte

- Idée : utiliser des algorithmes d'apprentissage supervisé et non supervisé pour obtenir une meilleure protection,
- Création un Meta-classifieur,
- Différentes techniques pour optimiser le classifieur : Bagging, Boosting, Voting, etc.
- Techniques encore étudiées pour trouver le meilleur Meta-Classifieur.

Bilan

Bilan

Méthodes étudiées

A. Méthodes descriptives basées sur des modèles géométriques :
   a) Méthode factorielle : projection et visualisation dans un espace de dimension inférieure
      ACP
   b) Analyse typologique : regroupement en classes homogènes
      k-means, DBSCAN, CAH

B. Méthodes prédictives :
   a) Modèles paramétriques ou semi-paramétriques : regression linéaire, modèle linéaire, regression logistique, SVM
   b) Prédiction sans modèle par analyse probaliliste : k-NN

Evaluation des méthodes

a) Indicateurs spécifiques aux méthodes
(R^2, déviance, tests)

b) Courbe ROC
(sensibilité en fonction de la spécificité)

c) Matrice de confusion
(taux d'erreurs, taux de faux positifs, taux de faux négatifs)

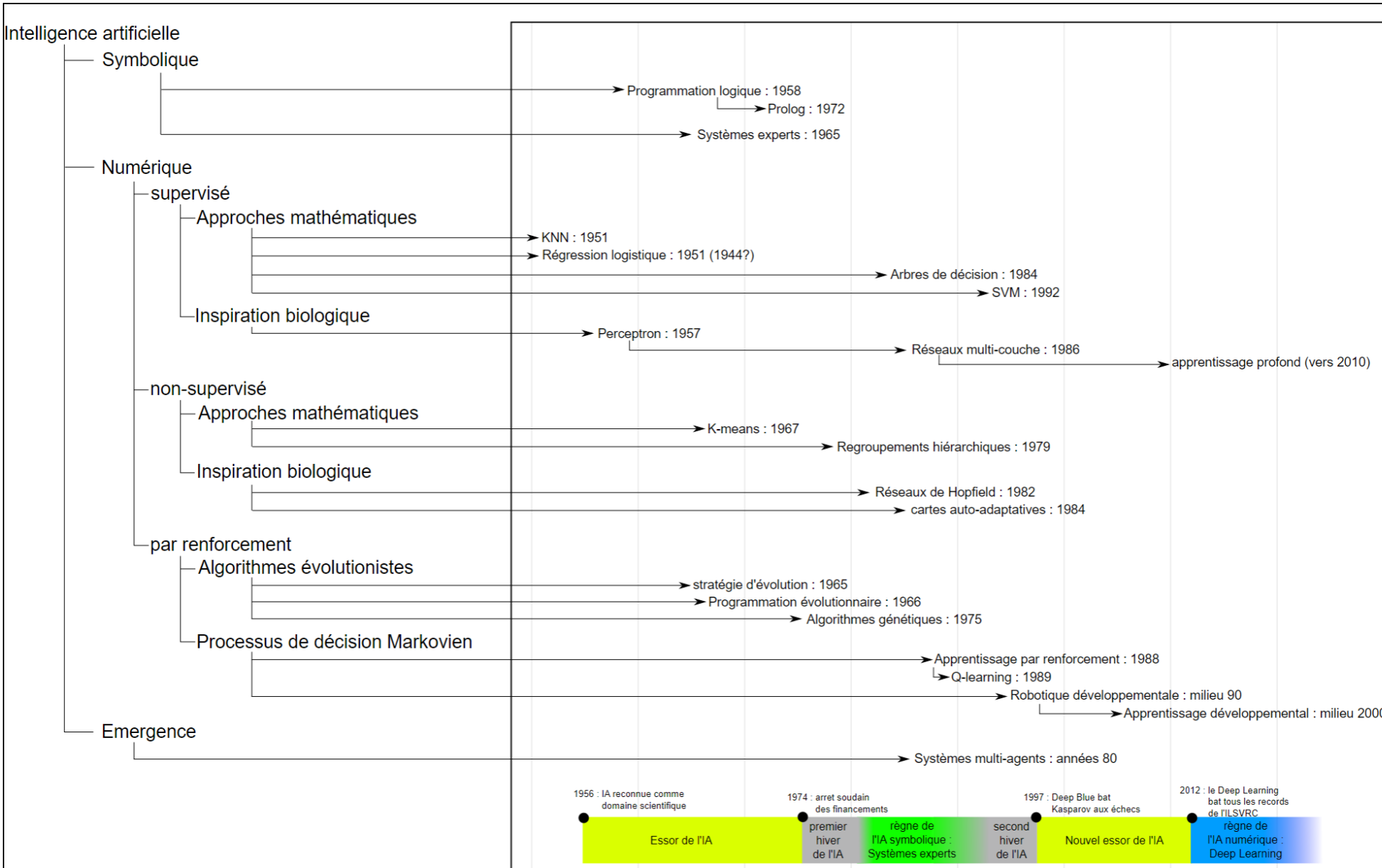
Vers l'I.A.

A. symbolique
   - Systèmes experts
   - langages de programmation logique (eg PROLOG)

B. numérique
   a) Supervisés : la machine apprend avec des données labellisées
      - SVM
      - K-NN
      - Arbres de décision
      - Réseaux de neurones supervisés (Perceptrons Multi-couches, Apprentissage Profond)
   b) Non-supervisés : apprentissage sur des données sans label
      - K-moyenne
      - Regroupement hiérarchiques
      - Réseaux de neurones non-supervisés (cartes auto-organisatrices ou de Konohen)
   c) Par renforcement :
      associations données-résultat par l'expérience d'un agent, IA incarné

C. émergence (SMA : comportements complexes émergent d'agents très simples, pas d'apprentissage)

Vers l'I.A.


Intelligence artificielle
Symbolique
Programmation logique : 1958
Prolog : 1972
Systèmes experts : 1965
Numérique
supervisé
Approches mathématiques
KNN : 1951
Régression logistique : 1951 (1944?)
Arbres de décision : 1984
SVM : 1992
Inspiration biologique
Perceptron : 1957
Réseaux multi-couche : 1986
apprentissage profond (vers 2010)
non-supervisé
Approches mathématiques
K-means : 1967
Regroupements hiérarchiques : 1979
Inspiration biologique
Réseaux de Hopfield : 1982
cartes auto-adaptatives : 1984
par renforcement
Algorithmes évolutionistes
stratégie d'évolution : 1965
Programmation évolutionnaire : 1966
Algorithmes génétiques : 1975
Processus de décision Markovien
Apprentissage par renforcement : 1988
Q-learning : 1989
Robotique développementale : milieu 90
Apprentissage développemental : milieu 200
Emergence
Systèmes multi-agents : années 80
1956 : IA reconnue comme domaine scientifique
1974 : arret soudain des financements
1997 : Deep Blue bat Kasparov aux échecs
2012 : le Deep Learning bat tous les records de l'ILSVRC
Essor de l'IA
premier hiver de l'IA
règne de l'IA symbolique : Systèmes experts
second hiver de l'IA
Nouvel essor de l'IA
règne de l'IA numérique : Deep Learning

Bibliographie

Bibliographie et webographie

Bibliographie

- **Statistique exploratoire multidimensionnelle**, Lebart-Morineau-Piron, Dunod
- **Analyses factorielles simples et multiples**, Escoffier-Pagès, Dunod
- **Data mining, découverte de connaissances dans les données**, Larose-Larose, Vuibert
- **Data Mining et statistique décisionnelle**, S Tuffery, Éditions Technip
- "**Big Data, Machine Learning et Apprentissage profond**", S Tuffery, Éditions Technip
- **Introduction to Machine Learning with Applications in Information Security** by Mark Stamp, Chapman and Hall/CRC

Bibliographie

- **Data Mining and machine learning in Cybersecurity**, Sumeet Dua and Xian Du, CRC Press
- Article Profils Propres pour la Détection d'Intrusion, Yacine Bouzida and Sylvain Gombault
- Vers une détection à la source des activités malveillantes dans les clouds publics : application aux attaques de déni de service, Badis Hammi, Thèse de doctorat à l'UTT
- **Data Science In Cybersecurity And Cyberthreat Intelligence**, Leslie F. Sikos, Kim-Kwang Raymond Choo, Springer
- **Data science fondamentaux et études de cas Machine learning avec Python et R** Eric Biernat, Michel Lutz, Eyrolles

Webographie

- Base de données disponibles sur le site de l'UCI :
  https ://archive.ics.uci.edu/ml/datasets.php
- Les cours de Ricco Rakotomalala (une mine d'or !) :
  http ://eric.univ-lyon2.fr/%7Ericco/cours/supports_data_mining.html
- un article sur les liens entre data mining et sécurité :
  https ://www.kdnuggets.com/2019/09/applying-data-science-cybersecurity-network-attacks-events.html
- blog sur le data-mining (entre autres) :
  https ://freakonometrics.hypotheses.org/category/courses/ia-data-science
- un blog sur les statistiques (très complet) :
  https ://lemakistatheux.wordpress.com/
- Les cours de Stéphane Mallat au collège de France :
  https ://www.college-de-france.fr/site/stephane-mallat/_audiovideos.htm

Webographie

- un youtuber qui parle de l'ACP et du fléau de la dimension :
  https ://www.youtube.com/watch ?v=Z2kqh−pItQ
- Articles sur le machine learning et ses liens avec la sécurité :
  https ://www.sentryo.net/fr/machine-learning-cybersecurite-trafics-chiffres/
  et
  https ://www.securityinsider-wavestone.com/2016/10/machine-learning-cybersecurite.html