Introduction à la fouille de données

M. Ledmi
m_ledmi@esi.dz

Département d'Informatique Khenchela

2020/2021

Plan

Vous êtes ici

Quelques Notation

- On notera $\mathcal{X}$ un ensemble de données.
- Chaque donnée est décrite par un ensemble $\mathcal{A}$ d'attributs.
- Chaque attribut $a \in \mathcal{A}$ prend sa valeur dans un certain ensemble de valeurs $\mathcal{V}_a$
- Ainsi, on peut considérer l'ensemble des données x dont les coordonnées balayent toutes les valeurs possibles des attributs : c'est l'espace des données que nous noterons $\mathcal{D}$.
- Si l'on note $a_1, a_2 \ldots a_P$ les P attributs,$\mathcal{D} = V_{a1} \times V_{a2} \times \ldots \times V_{aP}$.
- Toute donnée appartient à cet ensemble et on a $\mathcal{X} \subset \mathcal{D}$.

Quelques Notation

- On notera $\mathcal{X}$ un ensemble de données.
- Chaque donnée est décrite par un ensemble $\mathcal{A}$ d'attributs.
- Chaque attribut $a \in \mathcal{A}$ prend sa valeur dans un certain ensemble de valeurs $\mathcal{V}_a$
- Ainsi, on peut considérer l'ensemble des données x dont les coordonnées balayent toutes les valeurs possibles des attributs : c'est l'espace des données que nous noterons $\mathcal{D}$.
- Si l'on note $a_1, a_2 \dots a_P$ les P attributs,$\mathcal{D} = V_{a1} \times V_{a2} \times \dots \times V_{aP}$.
- Toute donnée appartient à cet ensemble et on a $\mathcal{X} \subset \mathcal{D}$.

Quelques Notation

- On notera $\mathcal{X}$ un ensemble de données.
- Chaque donnée est décrite par un ensemble $\mathcal{A}$ d'attributs.
- Chaque attribut $a \in \mathcal{A}$ prend sa valeur dans un certain ensemble de valeurs $\mathcal{V}_a$
- Ainsi, on peut considérer l'ensemble des données x dont les coordonnées balayent toutes les valeurs possibles des attributs : c'est l'espace des données que nous noterons $\mathcal{D}$.
- Si l'on note $a_1, a_2 \dots a_P$ les P attributs,$\mathcal{D} = V_{a1} \times V_{a2} \times \dots \times V_{aP}$.
- Toute donnée appartient à cet ensemble et on a $\mathcal{X} \subset \mathcal{D}$.

Quelques Notation

- On notera $\mathcal{X}$ un ensemble de données.
- Chaque donnée est décrite par un ensemble $\mathcal{A}$ d'attributs.
- Chaque attribut $a \in \mathcal{A}$ prend sa valeur dans un certain ensemble de valeurs $\mathcal{V}_a$
- Ainsi, on peut considérer l'ensemble des données x dont les coordonnées balayent toutes les valeurs possibles des attributs : c'est l'espace des données que nous noterons $\mathcal{D}$.
- Si l'on note $a_1, a_2 \dots a_P$ les P attributs,$\mathcal{D} = V_{a1} \times V_{a2} \times \dots \times V_{aP}$.
- Toute donnée appartient à cet ensemble et on a $\mathcal{X} \subset \mathcal{D}$.

Quelques Notation

- On notera $\mathcal{X}$ un ensemble de données.
- Chaque donnée est décrite par un ensemble $\mathcal{A}$ d'attributs.
- Chaque attribut $a \in \mathcal{A}$ prend sa valeur dans un certain ensemble de valeurs $\mathcal{V}_a$
- Ainsi, on peut considérer l'ensemble des données x dont les coordonnées balayent toutes les valeurs possibles des attributs : c'est l'espace des données que nous noterons $\mathcal{D}$.
- Si l'on note $a_1, a_2 \dots a_P$ les P attributs,$\mathcal{D} = V_{a1} \times V_{a2} \times \dots \times V_{aP}$.
- Toute donnée appartient à cet ensemble et on a $\mathcal{X} \subset \mathcal{D}$.

Quelques Notation

- On notera $\mathcal{X}$ un ensemble de données.
- Chaque donnée est décrite par un ensemble $\mathcal{A}$ d'attributs.
- Chaque attribut $a \in \mathcal{A}$ prend sa valeur dans un certain ensemble de valeurs $\mathcal{V}_a$
- Ainsi, on peut considérer l'ensemble des données x dont les coordonnées balayent toutes les valeurs possibles des attributs : c'est l'espace des données que nous noterons $\mathcal{D}$.
- Si l'on note $a_1, a_2 \dots a_P$ les P attributs,$\mathcal{D} = V_{a1} \times V_{a2} \times \dots \times V_{aP}$.
- Toute donnée appartient à cet ensemble et on a $\mathcal{X} \subset \mathcal{D}$.

Natures d'attributs

Une donnée est un :

- **Enregistrement** au sens des bases de données,
- **Individu** (terminologie issue des statistiques).
- **Instance** (terminologie orientée objet en informatique) .
- **Tuple** (terminologie base de données)

Natures d'attributs

Une donnée est un :

- **Enregistrement** au sens des bases de données,
- **Individu** (terminologie issue des statistiques).
- **Instance** (terminologie orientée objet en informatique) .
- **Tuple** (terminologie base de données)

Natures d'attributs

Une donnée est un :

- **Enregistrement** au sens des bases de données,
- **Individu** (terminologie issue des statistiques).
- **Instance** (terminologie orientée objet en informatique) .
- **Tuple** (terminologie base de données)

Natures d'attributs

Une donnée est un :

- **Enregistrement** au sens des bases de données,
- **Individu** (terminologie issue des statistiques).
- **Instance** (terminologie orientée objet en informatique) .
- **Tuple** (terminologie base de données)

Natures d'attributs

Une donnée est un :

- **Enregistrement** au sens des bases de données,
- **Individu** (terminologie issue des statistiques).
- **Instance** (terminologie orientée objet en informatique) .
- **Tuple** (terminologie base de données)

> Une données est caractérisée par un ensemble de champs, de caractères , ou encore d'attributs.

Natures d'attributs

Un attribut peut être de nature *qualitative* ou *quantitative* en fonction de l'ensemble des valeurs qu'il peut prendre.

Attribut qualitatif

- Si on ne peut pas en faire une moyenne ;
- Sa valeur est d'un type défini en extension (une couleur, une marque de voiture, ...etc.).

Natures d'attributs

Un attribut peut être de nature *qualitative* ou *quantitative* en fonction de l'ensemble des valeurs qu'il peut prendre.

Attribut qualitatif

- Si on ne peut pas en faire une moyenne ;
- Sa valeur est d'un type défini en extension (une couleur, une marque de voiture, . . . etc.).

Natures d'attributs

Un attribut peut être de nature *qualitative* ou *quantitative* en fonction de l'ensemble des valeurs qu'il peut prendre.

Attribut qualitatif

- Si on ne peut pas en faire une moyenne ;
- Sa valeur est d'un type défini en extension (une couleur, une marque de voiture, ...etc.).

Attribut quantitatif

- L'attribut est de nature quantitative : un entier, un réel, ...etc.
- Il peut représenter un salaire, une surface, un nombre d'habitants, ...etc.

Natures d'attributs

Un attribut peut être de nature *qualitative* ou *quantitative* en fonction de l'ensemble des valeurs qu'il peut prendre.

Attribut qualitatif

- Si on ne peut pas en faire une moyenne ;
- Sa valeur est d'un type défini en extension (une couleur, une marque de voiture, ...etc.).

Attribut quantitatif

- L'attribut est de nature quantitative : un entier, un réel, ...etc.
- Il peut représenter un salaire, une surface, un nombre d'habitants, ...etc.

Natures d'attributs

Un attribut peut être de nature *qualitative* ou *quantitative* en fonction de l'ensemble des valeurs qu'il peut prendre.

Attribut qualitatif

- Si on ne peut pas en faire une moyenne ;
- Sa valeur est d'un type défini en extension (une couleur, une marque de voiture, ...etc.).

Attribut quantitatif

- L'attribut est de nature quantitative : un entier, un réel, ...etc.
- Il peut représenter un salaire, une surface, un nombre d'habitants, ...etc.

Natures d'attributs

La valeur d'un attribut est censée représenter une certaine mesure d'une quantitée

Attribut nominal

- Les valeurs sont des symboles (des noms)
- Aucune relation (ordre ou distance) entre les nominaux n'existe.
- Seuls des tests d'égalité peuvent être exécutés.

Natures d'attributs

La valeur d'un attribut est censée représenter une certaine mesure d'une quantitée

Attribut nominal

- Les valeurs sont des symboles (des noms)
- Aucune relation (ordre ou distance) entre les nominaux n'existe.
- Seuls des tests d'égalité peuvent être exécutés.

Natures d'attributs

La valeur d'un attribut est censée représenter une certaine mesure d'une quantitée

Attribut nominal

- Les valeurs sont des symboles (des noms)
- Aucune relation (ordre ou distance) entre les nominaux n'existe.
- Seuls des tests d'égalité peuvent être exécutés.

Exemples

- Les valeurs de Temps (Ensoleillé, Pluvieux, Neigeux, Gris)
- La couleur (Vert, Bleu, Rouge)

Natures d'attributs

La valeur d'un attribut est censée représenter une certaine mesure d'une quantitée

Attribut nominal

- Les valeurs sont des symboles (des noms)
- Aucune relation (ordre ou distance) entre les nominaux n'existe.
- Seuls des tests d'égalité peuvent être exécutés.

Exemples

- Les valeurs de Temps (Ensoleillé, Pluvieux, Neigeux, Gris)
- La couleur (Vert, Bleu, Rouge)
- Si $Temps = Pluvieux$ alors $Jouer = Non$

Natures d'attributs

La valeur d'un attribut est censée représenter une certaine mesure d'une quantitée

Attribut nominal

- Les valeurs sont des symboles (des noms)
- Aucune relation (ordre ou distance) entre les nominaux n'existe.
- Seuls des tests d'égalité peuvent être exécutés.

Exemples

- Les valeurs de Temps (Ensoleillé, Pluvieux, Neigeux, Gris)
- La couleur (Vert, Bleu, Rouge)
- Si $Temps = Pluvieux$ alors $Jouer = Non$

Natures d'attributs

La valeur d'un attribut est censée représenter une certaine mesure d'une quantitée

Attribut nominal

- Les valeurs sont des symboles (des noms)
- Aucune relation (ordre ou distance) entre les nominaux n'existe.
- Seuls des tests d'égalité peuvent être exécutés.

Exemples

- Les valeurs de Temps (Ensoleillé, Pluvieux, Neigeux, Gris)
- La couleur (Vert, Bleu, Rouge)
- Si $Temps = Pluvieux$ alors $Jouer = Non$

Natures d'attributs

La valeur d'un attribut est censée représenter une certaine mesure d'une quantitée

Attribut ordinal

- Une notion d'ordre s'impose sur les ordinaux.
- Mais il n'est pas possible de calculer directement des distances entre des valeurs ordinales.
- Les opérations d'addition et de soustraction ne sont pas possibles.

Natures d'attributs

La valeur d'un attribut est censée représenter une certaine mesure d'une quantitée

Attribut ordinal

- Une notion d'ordre s'impose sur les ordinaux.
- Mais il n'est pas possible de calculer directement des distances entre des valeurs ordinales.
- Les opérations d'addition et de soustraction ne sont pas possibles.

Natures d'attributs

La valeur d'un attribut est censée représenter une certaine mesure d'une quantitée

Attribut ordinal

- Une notion d'ordre s'impose sur les ordinaux.
- Mais il n'est pas possible de calculer directement des distances entre des valeurs ordinales.
- Les opérations d'addition et de soustraction ne sont pas possibles.

Exemples

- La température est décrite par les adjectifs chaud, froid, moyen, et chaud > moyen > froid

Natures d'attributs

La valeur d'un attribut est censée représenter une certaine mesure d'une quantitée

Attribut ordinal

- Une notion d'ordre s'impose sur les ordinaux.
- Mais il n'est pas possible de calculer directement des distances entre des valeurs ordinales.
- Les opérations d'addition et de soustraction ne sont pas possibles.

Exemples

- La température est décrite par les adjectifs chaud, froid, moyen, et chaud > moyen > froid

Natures d'attributs

La valeur d'un attribut est censée représenter une certaine mesure d'une quantitée

Attribut ordinal

- Une notion d'ordre s'impose sur les ordinaux.
- Mais il n'est pas possible de calculer directement des distances entre des valeurs ordinales.
- Les opérations d'addition et de soustraction ne sont pas possibles.

Exemples

- La température est décrite par les adjectifs chaud, froid, moyen, et chaud $>$ moyen $>$ froid

Natures d'attributs

La valeur d'un attribut est censée représenter une certaine mesure d'une quantitée

Attribut de type intervalle

- Les intervalles impliquent une notion d'ordre, et les valeurs sont mesurées dans des unités spécifiques et fixées
- La somme, la différence et le produit de deux intervalles ne sont pas possibles.

Natures d'attributs

La valeur d'un attribut est censée représenter une certaine mesure d'une quantitée

Attribut de type intervalle

- Les intervalles impliquent une notion d'ordre, et les valeurs sont mesurées dans des unités spécifiques et fixées
- La somme, la différence et le produit de deux intervalles ne sont pas possibles.

Exemples

- La température exprimée en degrés *Celsius* ou *Fahrenheit*
- L'attribut année : (2000,2010,2014).

Natures d'attributs

La valeur d'un attribut est censée représenter une certaine mesure d'une quantitée

Attribut de type intervalle

- Les intervalles impliquent une notion d'ordre, et les valeurs sont mesurées dans des unités spécifiques et fixées
- La somme, la différence et le produit de deux intervalles ne sont pas possibles.

Exemples

- La température exprimée en degrés *Celsius* ou *Fahrenheit*
- L'attribut année : (2000,2010,2014).

Natures d'attributs

La valeur d'un attribut est censée représenter une certaine mesure d'une quantitée

Attribut de type intervalle

- Les intervalles impliquent une notion d'ordre, et les valeurs sont mesurées dans des unités spécifiques et fixées
- La somme, la différence et le produit de deux intervalles ne sont pas possibles.

Exemples

- La température exprimée en degrés *Celsius* ou *Fahrenheit*
- L'attribut année : (2000,2010,2014).

Natures d'attributs

La valeur d'un attribut est censée représenter une certaine mesure d'une quantitée

Attribut de type rapport (ratio)

- Toutes les opérations mathématiques sont autorisées sur les attributs de ce type.

Natures d'attributs

La valeur d'un attribut est censée représenter une certaine mesure d'une quantitée

Attribut de type rapport (ratio)

- Toutes les opérations mathématiques sont autorisées sur les attributs de ce type.

Exemples

L'attribut distance :

- On peut comparer deux distances.
- On peut additionner deux distances.
- La distance entre un objet et lui-même est zéro.

Natures d'attributs

La valeur d'un attribut est censée représenter une certaine mesure d'une quantitée

Attribut de type rapport (ratio)

- Toutes les opérations mathématiques sont autorisées sur les attributs de ce type.

Exemples

L'attribut distance :

- On peut comparer deux distances.
- On peut additionner deux distances.
- La distance entre un objet et lui-même est zéro.

Natures d'attributs

La valeur d'un attribut est censée représenter une certaine mesure d'une quantitée

Attribut de type rapport (ratio)

- Toutes les opérations mathématiques sont autorisées sur les attributs de ce type.

Exemples

L'attribut distance :

- On peut comparer deux distances.
- On peut additionner deux distances.
- La distance entre un objet et lui-même est zéro.

Classification supervisée

On dispose d'un ensemble $\mathcal{X}$ de N exemples, i.e. des couples (donnée, étiquette). Chaque donnée $x_i \in \mathcal{D}$ est caractérisée par P attributs et par sa classe $y_i \in \mathcal{Y}$.

Définition

Dans un problème de classification supervisée, la classe prend sa valeur parmi un ensemble $\mathcal{Y}$ fini. Le problème consiste alors, en s'appuyant sur l'ensemble d'exemples $X = (x_i, y_i) \quad i \in \{1 \dots N\}$, à prédire la classe de toute nouvelle donnée $x \in \mathcal{D}$.

Classification supervisée

Remarque :

- On parle de classification **binaire** quand le nombre de classes y_i est deux .
- Il peut naturellement être quelconque. Dans tous les cas, il s'agît d'un attribut qualitatif pouvant prendre un nombre fini de valeurs.
- Dans l'absolu, une donnée peut appartenir à plusieurs classes : c'est alors un problème **multi-classes**.

Classification supervisée

Remarque :

- On parle de classification **binaire** quand le nombre de classes y_i est deux .
- Il peut naturellement être quelconque. Dans tous les cas, il s'agît d'un attribut qualitatif pouvant prendre un nombre fini de valeurs.
- Dans l'absolu, une donnée peut appartenir à plusieurs classes : c'est alors un problème **multi-classes**.
- Ici, on considère que chaque donnée appartient à une et une seule classe.

Classification supervisée

Remarque :

- On parle de classification **binaire** quand le nombre de classes y_i est deux .
- Il peut naturellement être quelconque. Dans tous les cas, il s'agît d'un attribut qualitatif pouvant prendre un nombre fini de valeurs.
- Dans l'absolu, une donnée peut appartenir à plusieurs classes : c'est alors un problème **multi-classes**.
- Ici, on considère que chaque donnée appartient à une et une seule classe.
- On utilise le mot *étiquette* comme synonyme de *classe*.

Classification supervisée

Remarque :

- On parle de classification **binaire** quand le nombre de classes y_i est deux .
- Il peut naturellement être quelconque. Dans tous les cas, il s'agît d'un attribut qualitatif pouvant prendre un nombre fini de valeurs.
- Dans l'absolu, une donnée peut appartenir à plusieurs classes : c'est alors un problème **multi-classes**.
- Ici, on considère que chaque donnée appartient à une et une seule classe.
- On utilise le mot *étiquette* comme synonyme de *classe*.

Classification supervisée

Remarque :

- On parle de classification **binaire** quand le nombre de classes y_i est deux .
- Il peut naturellement être quelconque. Dans tous les cas, il s'agît d'un attribut qualitatif pouvant prendre un nombre fini de valeurs.
- Dans l'absolu, une donnée peut appartenir à plusieurs classes : c'est alors un problème **multi-classes**.
- Ici, on considère que chaque donnée appartient à une et une seule classe.
- On utilise le mot *étiquette* comme synonyme de *classe*.

Classification supervisée

Définition

Un **classifieur** est une procédure (un algorithme) qui, à partir d'un ensemble d'exemples, produit une prédiction de la classe de toute donnée.

Remarque :

- D'une manière générale, un classifieur procède par *induction* : à partir d'exemples (donc de cas particuliers), on construit une connaissance plus générale.
- Parmi les modèles, on distingue ceux qui peuvent être interprétés par un humain (arbre de décision) et ceux qui ne le peuvent pas (réseau de neurones).

Classification supervisée

Définition

Un **classifieur** est une procédure (un algorithme) qui, à partir d'un ensemble d'exemples, produit une prédiction de la classe de toute donnée.

Remarque :

- D'une manière générale, un classifieur procède par *induction* : à partir d'exemples (donc de cas particuliers), on construit une connaissance plus générale.
- Parmi les modèles, on distingue ceux qui peuvent être interprétés par un humain (arbre de décision) et ceux qui ne le peuvent pas (réseau de neurones).

Classification supervisée

Plusieurs problèmes sont à résoudre :

- Méthode d'induction du classeur ?
- Comment utiliser le classifieur obtenu ?
- Comment évaluer la qualité du classeur obtenu : taux d'erreur (ou de succés) ?
- Comment traiter les attributs manquants dans le jeu d'apprentissage ?
- Comment traiter les attributs manquants dans une donnée à classer ?

Classification supervisée

Plusieurs problèmes sont à résoudre :

- Méthode d'induction du classeur ?
- Comment utiliser le classifieur obtenu ?
- Comment évaluer la qualité du classeur obtenu : taux d'erreur (ou de succés) ?
- Comment traiter les attributs manquants dans le jeu d'apprentissage ?
- Comment traiter les attributs manquants dans une donnée à classer ?

Classification supervisée

Plusieurs problèmes sont à résoudre :

- Méthode d'induction du classeur ?
- Comment utiliser le classifieur obtenu ?
- Comment évaluer la qualité du classeur obtenu : taux d'erreur (ou de succés) ?
- Comment traiter les attributs manquants dans le jeu d'apprentissage ?
- Comment traiter les attributs manquants dans une donnée à classer ?

Classification supervisée

Plusieurs problèmes sont à résoudre :

- Méthode d'induction du classeur ?
- Comment utiliser le classifieur obtenu ?
- Comment évaluer la qualité du classeur obtenu : taux d'erreur (ou de succés) ?
- Comment traiter les attributs manquants dans le jeu d'apprentissage ?
- Comment traiter les attributs manquants dans une donnée à classer ?

Classification supervisée

Plusieurs problèmes sont à résoudre :

- Méthode d'induction du classeur ?
- Comment utiliser le classifieur obtenu ?
- Comment évaluer la qualité du classeur obtenu : taux d'erreur (ou de succés) ?
- Comment traiter les attributs manquants dans le jeu d'apprentissage ?
- Comment traiter les attributs manquants dans une donnée à classer ?

Arbres de décision

On se donne :

- un ensemble $\mathcal{X}$ de N exemples notés x_i dont les P attributs sont quantitatifs ou qualitatifs.
- Chaque exemple x est étiqueté, c'est-à-dire qu'il lui est associée une *classe* ou un *attribut cible* que l'on note $y \in \mathcal{Y}$.

Arbres de décision

On se donne :

- un ensemble $\mathcal{X}$ de N exemples notés x_i dont les P attributs sont quantitatifs ou qualitatifs.
- Chaque exemple x est étiqueté, c'est-à-dire qu'il lui est associée une *classe* ou un *attribut cible* que l'on note $y \in \mathcal{Y}$.

Arbres de décision

Définition

A partir de ces exemples, on construit un **arbre** dit **de décision** tel que :

- Chaque noeud correspond à un test sur la valeur d'un ou plusieurs attributs ;
- Chaque branche partant d'un noeud correspond à une ou plusieurs valeurs de ce test ;
- A chaque feuille est associée une valeur de l'attribut cible.

Arbres de décision

Définition

A partir de ces exemples, on construit un **arbre** dit **de décision** tel que :

- Chaque noeud correspond à un test sur la valeur d'un ou plusieurs attributs ;
- Chaque branche partant d'un noeud correspond à une ou plusieurs valeurs de ce test ;
- A chaque feuille est associée une valeur de l'attribut cible.

Arbres de décision

Définition

A partir de ces exemples, on construit un **arbre** dit **de décision** tel que :

- Chaque noeud correspond à un test sur la valeur d'un ou plusieurs attributs ;
- Chaque branche partant d'un noeud correspond à une ou plusieurs valeurs de ce test ;
- A chaque feuille est associée une valeur de l'attribut cible.

Arbres de décision

Construction de l'arbre

Plusieurs algorithmes ont été proposés, notamment :

- **CART** proposé par Breiman dans les années 1980 .
- **ID3** de Quinlan proposé en 1986 qui a été raffiné par la suite (**C4.5** puis **C5**) .

- **ID3** ne prend en compte que des attributs nominaux.
- Son successeur, C4.5, prend également en charge des attributs quantitatifs.

Arbres de décision

Construction de l'arbre

Plusieurs algorithmes ont été proposés, notamment :

- **CART** proposé par Breiman dans les années 1980 .
- **ID3** de Quinlan proposé en 1986 qui a été raffiné par la suite (**C4.5** puis **C5**) .

- **ID3** ne prend en compte que des attributs nominaux.
- Son successeur, C4.5, prend également en charge des attributs quantitatifs.

Arbres de décision

Construction de l'arbre

Plusieurs algorithmes ont été proposés, notamment :

- **CART** proposé par Breiman dans les années 1980 .
- **ID3** de Quinlan proposé en 1986 qui a été raffiné par la suite (**C4.5** puis **C5**) .

- **ID3** ne prend en compte que des attributs nominaux.
- Son successeur, C4.5, prend également en charge des attributs quantitatifs.

Arbres de décision

Construction de l'arbre

Plusieurs algorithmes ont été proposés, notamment :

- **CART** proposé par Breiman dans les années 1980 .
- **ID3** de Quinlan proposé en 1986 qui a été raffiné par la suite (**C4.5** puis **C5**) .

- **ID3** ne prend en compte que des attributs nominaux.
- Son successeur, C4.5, prend également en charge des attributs quantitatifs.

Algorithme **ID3**

- Les tests placés dans un noeud par l'algorithme **ID3** concernent exclusivement le test de la valeur d'un et seul attribut.
- **ID3** fonctionne récursivement :
  - Il détermine un attribut à placer en racine de l'arbre.

Algorithme **ID3**

- Les tests placés dans un noeud par l'algorithme **ID3** concernent exclusivement le test de la valeur d'un et seul attribut.
- **ID3** fonctionne récursivement :
  - Il détermine un attribut à placer en racine de l'arbre.
  - Cette racine possède autant de branches que cet attribut prend de valeurs.
  - A chaque branche est associé un ensemble d'exemples dont l'attribut prend la valeur qui étiquette cette branche ;
  - On accroche alors au bout de cette branche l'arbre de décision construit sur ce sous-ensemble des exemples et en considérant tous les attributs excepté celui qui vient d'être mis à la racine.
  - Par cette procédure, l'ensemble des exemples ainsi que l'ensemble des attributs diminuent petit à petit au long de la descente dans l'arbre.

Algorithme ID3

- Les tests placés dans un noeud par l'algorithme **ID3** concernent exclusivement le test de la valeur d'un et seul attribut.
- **ID3** fonctionne récursivement :
  - Il détermine un attribut à placer en racine de l'arbre.
  - Cette racine possède autant de branches que cet attribut prend de valeurs.
  - A chaque branche est associé un ensemble d'exemples dont l'attribut prend la valeur qui étiquette cette branche ;
  - On accroche alors au bout de cette branche l'arbre de décision construit sur ce sous-ensemble des exemples et en considérant tous les attributs excepté celui qui vient d'être mis à la racine.
  - Par cette procédure, l'ensemble des exemples ainsi que l'ensemble des attributs diminuent petit à petit au long de la descente dans l'arbre.

Algorithme **ID3**

- Les tests placés dans un noeud par l'algorithme **ID3** concernent exclusivement le test de la valeur d'un et seul attribut.
- **ID3** fonctionne récursivement :
  - Il détermine un attribut à placer en racine de l'arbre.
  - Cette racine possède autant de branches que cet attribut prend de valeurs.
  - A chaque branche est associé un ensemble d'exemples dont l'attribut prend la valeur qui étiquette cette branche ;
  - On accroche alors au bout de cette branche l'arbre de décision construit sur ce sous-ensemble des exemples et en considérant tous les attributs excepté celui qui vient d'être mis à la racine.
  - Par cette procédure, l'ensemble des exemples ainsi que l'ensemble des attributs diminuent petit à petit au long de la descente dans l'arbre.

Algorithme **ID3**

- Les tests placés dans un noeud par l'algorithme **ID3** concernent exclusivement le test de la valeur d'un et seul attribut.
- **ID3** fonctionne récursivement :
  - Il détermine un attribut à placer en racine de l'arbre.
  - Cette racine possède autant de branches que cet attribut prend de valeurs.
  - A chaque branche est associé un ensemble d'exemples dont l'attribut prend la valeur qui étiquette cette branche ;
  - On accroche alors au bout de cette branche l'arbre de décision construit sur ce sous-ensemble des exemples et en considérant tous les attributs excepté celui qui vient d'être mis à la racine.
  - Par cette procédure, l'ensemble des exemples ainsi que l'ensemble des attributs diminuent petit à petit au long de la descente dans l'arbre.

Algorithme **ID3**

- Les tests placés dans un noeud par l'algorithme **ID3** concernent exclusivement le test de la valeur d'un et seul attribut.
- **ID3** fonctionne récursivement :
  - Il détermine un attribut à placer en racine de l'arbre.
  - Cette racine possède autant de branches que cet attribut prend de valeurs.
  - A chaque branche est associé un ensemble d'exemples dont l'attribut prend la valeur qui étiquette cette branche ;
  - On accroche alors au bout de cette branche l'arbre de décision construit sur ce sous-ensemble des exemples et en considérant tous les attributs excepté celui qui vient d'être mis à la racine.
  - Par cette procédure, l'ensemble des exemples ainsi que l'ensemble des attributs diminuent petit à petit au long de la descente dans l'arbre.

Algorithme **ID3**

- Les tests placés dans un noeud par l'algorithme **ID3** concernent exclusivement le test de la valeur d'un et seul attribut.
- **ID3** fonctionne récursivement :
  - Il détermine un attribut à placer en racine de l'arbre.
  - Cette racine possède autant de branches que cet attribut prend de valeurs.
  - A chaque branche est associé un ensemble d'exemples dont l'attribut prend la valeur qui étiquette cette branche ;
  - On accroche alors au bout de cette branche l'arbre de décision construit sur ce sous-ensemble des exemples et en considérant tous les attributs excepté celui qui vient d'être mis à la racine.
  - Par cette procédure, l'ensemble des exemples ainsi que l'ensemble des attributs diminuent petit à petit au long de la descente dans l'arbre.

Algorithme **ID3**

- Ayant l'idée de l'algorithme, il reste à résoudre une question centrale : quel attribut placer en racine ?
- Une fois cette question résolue, on itérera le raisonnement pour les sous-arbres.

Algorithme **ID3**

- Ayant l'idée de l'algorithme, il reste à résoudre une question centrale : quel attribut placer en racine ?
- Une fois cette question résolue, on itérera le raisonnement pour les sous-arbres.

Algorithme **ID3**

- Ayant l'idée de l'algorithme, il reste à résoudre une question centrale : quel attribut placer en racine ?
- Une fois cette question résolue, on itérera le raisonnement pour les sous-arbres.

On tente de réduire l'hétérogénéïté à chaque noeud :

- Les données qui atteignent un certain noeud de l'arbre de décision doivent être plus homogènes que les données atteignant un noeud ancêtre.
- Pour cela, on a besoin de pouvoir mesurer l'homogénéïté d'un ensemble de données.
- En physique, on mesure l'homogénéïté par l'**entropie**.

Algorithme **ID3**

- Ayant l'idée de l'algorithme, il reste à résoudre une question centrale : quel attribut placer en racine ?
- Une fois cette question résolue, on itérera le raisonnement pour les sous-arbres.

On tente de réduire l'hétérogénéïté à chaque noeud :

- Les données qui atteignent un certain noeud de l'arbre de décision doivent être plus homogènes que les données atteignant un noeud ancêtre.
- Pour cela, on a besoin de pouvoir mesurer l'homogénéïté d'un ensemble de données.
- En physique, on mesure l'homogénéïté par l'**entropie**.

Algorithme **ID3**

- Ayant l'idée de l'algorithme, il reste à résoudre une question centrale : quel attribut placer en racine ?
- Une fois cette question résolue, on itérera le raisonnement pour les sous-arbres.

On tente de réduire l'hétérogénéïté à chaque noeud :

- Les données qui atteignent un certain noeud de l'arbre de décision doivent être plus homogènes que les données atteignant un noeud ancêtre.
- Pour cela, on a besoin de pouvoir mesurer l'homogénéïté d'un ensemble de données.
- En physique, on mesure l'homogénéïté par l'**entropie**.

Algorithme **ID3**

Entropie

Soit un ensemble $\mathcal{X}$ d'exemples dont une proportion p_+ sont positifs et une proportion p_- sont négatifs. (Bien entendu, $p_+ + p_- = 1$.) L'entropie de $\mathcal{X}$ est :

$$H(\mathcal{X}) = -p_+ log_2(p_+) - p_- log_2(p_-)$$

Algorithme **ID3**

Entropie

Soit un ensemble $\mathcal{X}$ d'exemples dont une proportion p_+ sont positifs et une proportion p_- sont négatifs. (Bien entendu, $p_+ + p_- = 1$.) L'entropie de $\mathcal{X}$ est :

$$H(\mathcal{X}) = -p_+ log_2(p_+) - p_- log_2(p_-)$$

Remarque :

1. $0 \leq H(\mathcal{X}) \leq 1$;
2. Si $p_+ = 0$ ou $p_- = 0$, alors $H(\mathcal{X}) = 0$: ainsi, si tous exemples sont soit tous positifs, soit tous négatifs, l'entropie est nulle ;
3. Si $p_+ = p_- = 0.5$, alors $H(\mathcal{X}) = 1$: ainsi, s'il y a autant de positifs

Algorithme **ID3**

Entropie

Soit un ensemble $\mathcal{X}$ d'exemples dont une proportion p_+ sont positifs et une proportion p_- sont négatifs. (Bien entendu, $p_+ + p_- = 1$.) L'entropie de $\mathcal{X}$ est :

$$H(\mathcal{X}) = -p_+ log_2(p_+) - p_- log_2(p_-)$$

Remarque :

1. $0 \leq H(\mathcal{X}) \leq 1$;
2. Si $p_+ = 0$ ou $p_- = 0$, alors $H(\mathcal{X}) = 0$: ainsi, si tous exemples sont soit tous positifs, soit tous négatifs, l'entropie est nulle ;
3. Si $p_+ = p_- = 0.5$, alors $H(\mathcal{X}) = 1$: ainsi, s'il y a autant de positifs

Algorithme **ID3**

Entropie

Soit un ensemble $\mathcal{X}$ d'exemples dont une proportion p_+ sont positifs et une proportion p_- sont négatifs. (Bien entendu, $p_+ + p_- = 1$.) L'entropie de $\mathcal{X}$ est :

$$H(\mathcal{X}) = -p_+ log_2(p_+) - p_- log_2(p_-)$$

Remarque :

1. $0 \leq H(\mathcal{X}) \leq 1$;
2. Si $p_+ = 0$ ou $p_- = 0$, alors $H(\mathcal{X}) = 0$: ainsi, si tous exemples sont soit tous positifs, soit tous négatifs, l'entropie est nulle ;
3. Si $p_+ = p_- = 0.5$, alors $H(\mathcal{X}) = 1$: ainsi, s'il y a autant de positifs

Algorithme **ID3**

La définition précédente de l'entropie se généralise aisément à un attribut pouvant prendre plus de deux valeurs distinctes :

Définition

Pour une classe prenant n valeurs distinctes (numérotées de 1 à n) , notons p_i la proportion d'exemples dont la valeur de cet attribut est i dans l'ensemble d'exemples considéré $\mathcal{X}$. L'entropie de l'ensemble d'exemples $\mathcal{X}$ est :

$$H(\mathcal{X}) = -\sum_{i=1}^{n} p_i log_2(p_i)$$

Algorithme **ID3**

Remarque :

- Avoir des sous-sensembles dont l'entropie est minimale est intéressant : cela signifie que l'attribut placer à la racine discrimine les exemples en fonction de leur classe.
- Il est naturel de sommer ces entropies en les pondérant en fonction de la proportion d'exemples dans chacun des sous-ensembles.
- Voulant sélectionner l'attribut discriminant au mieux, on peut utiliser la différence entre l'entropie de l'ensemble d'exemples initial (utilisé pour déterminer la racine) et cette somme pondérée pour trouver l'attribut le plus intéressant à placer dans la racine.

Algorithme **ID3**

Remarque :

- Avoir des sous-sensembles dont l'entropie est minimale est intéressant : cela signifie que l'attribut placer à la racine discrimine les exemples en fonction de leur classe.
- Il est naturel de sommer ces entropies en les pondérant en fonction de la proportion d'exemples dans chacun des sous-ensembles.
- Voulant sélectionner l'attribut discriminant au mieux, on peut utiliser la différence entre l'entropie de l'ensemble d'exemples initial (utilisé pour déterminer la racine) et cette somme pondérée pour trouver l'attribut le plus intéressant à placer dans la racine.

Algorithme **ID3**

Remarque :

- Avoir des sous-sensembles dont l'entropie est minimale est intéressant : cela signifie que l'attribut placer à la racine discrimine les exemples en fonction de leur classe.
- Il est naturel de sommer ces entropies en les pondérant en fonction de la proportion d'exemples dans chacun des sous-ensembles.
- Voulant sélectionner l'attribut discriminant au mieux, on peut utiliser la différence entre l'entropie de l'ensemble d'exemples initial (utilisé pour déterminer la racine) et cette somme pondérée pour trouver l'attribut le plus intéressant à placer dans la racine.

Algorithme **ID3**

Remarque :

- L'attribut qui maximise cette différence est l'attribut qui discrimine le mieux les exemples en fonction de leur classe.
- Cette différence porte le nom de **gain d'information**.

Algorithme **ID3**

Gain d'information

Soit une population d'exemples $\mathcal{X}$. Le gain d'information de $\mathcal{X}$ par rapport à un attribut a_j donné est la variation d'entropie causée par la partition de $\mathcal{X}$ selon a_j :

$$Gain(\mathcal{X}, a_j) = H(\mathcal{X}) - \sum_{v \in valeurs(a_j)} \frac{\left|\mathcal{X}_{a_j=v}\right|}{|\mathcal{X}|} H(\mathcal{X}_{a_j=v})$$

Algorithme **ID3**

Gain d'information

Soit une population d'exemples $\mathcal{X}$. Le gain d'information de $\mathcal{X}$ par rapport à un attribut a_j donné est la variation d'entropie causée par la partition de $\mathcal{X}$ selon a_j :

$$Gain(\mathcal{X}, a_j) = H(\mathcal{X}) - \sum_{v \in valeurs(a_j)} \frac{|\mathcal{X}_{a_j=v}|}{|\mathcal{X}|} H(\mathcal{X}_{a_j=v})$$

où

- $\mathcal{X}_{a_j=v} \subset \mathcal{X}$ est l'ensemble des exemples dont l'attribut considéré a_j prend la valeur v,
- La notation $|\mathcal{X}|$ indique le cardinal de l'ensemble $\mathcal{X}$;

Algorithme **ID3**

Gain d'information

Soit une population d'exemples $\mathcal{X}$. Le gain d'information de $\mathcal{X}$ par rapport à un attribut a_j donné est la variation d'entropie causée par la partition de $\mathcal{X}$ selon a_j :

$$Gain(\mathcal{X}, a_j) = H(\mathcal{X}) - \sum_{v \in valeurs(a_j)} \frac{\left|\mathcal{X}_{a_j=v}\right|}{|\mathcal{X}|} H(\mathcal{X}_{a_j=v})$$

où

- $\mathcal{X}_{a_j=v} \subset \mathcal{X}$ est l'ensemble des exemples dont l'attribut considéré a_j prend la valeur v,
- La notation $|\mathcal{X}|$ indique le cardinal de l'ensemble $\mathcal{X}$.

Algorithme **ID3**

Exemple :

Supposons que $\mathcal{X}$ soit constitué de 14 exemples, 9 positifs et 5 négatifs. Parmi ces exemples, 6 positifs et 2 négatifs prennent la valeur *oui* pour l'attribut a, tandis que les autres exemples prennent la valeur *non* pour cet attribut.

Algorithme **ID3**

Exemple :

Supposons que $\mathcal{X}$ soit constitué de 14 exemples, 9 positifs et 5 négatifs. Parmi ces exemples, 6 positifs et 2 négatifs prennent la valeur *oui* pour l'attribut a, tandis que les autres exemples prennent la valeur *non* pour cet attribut.

$$Gain(\mathcal{X}, a) = H(\mathcal{X}) - \sum_{v \in \{oui, non\}} \frac{|\mathcal{X}_{a=v}|}{|\mathcal{X}|} H(\mathcal{X}_{a=v})$$

Algorithme **ID3**

Exemple :

Supposons que $\mathcal{X}$ soit constitué de 14 exemples, 9 positifs et 5 négatifs. Parmi ces exemples, 6 positifs et 2 négatifs prennent la valeur *oui* pour l'attribut a, tandis que les autres exemples prennent la valeur *non* pour cet attribut.

$$Gain(\mathcal{X}, a) = H(\mathcal{X}) - \frac{8}{14} H(\mathcal{X}_{a=oui}) - \frac{6}{14} H(\mathcal{X}_{a=non})$$

Algorithme **ID3**

Exemple :

Supposons que $\mathcal{X}$ soit constitué de 14 exemples, 9 positifs et 5 négatifs. Parmi ces exemples, 6 positifs et 2 négatifs prennent la valeur *oui* pour l'attribut a, tandis que les autres exemples prennent la valeur *non* pour cet attribut.

$$Gain(\mathcal{X}, a) = H(\mathcal{X}) - \frac{8}{14} H(\mathcal{X}_{a=oui}) - \frac{6}{14} H(\mathcal{X}_{a=non})$$

$$H(\mathcal{X}_{a=oui}) = -\left(\frac{6}{8}\log_2\frac{6}{8} + \frac{2}{8}\log_2\frac{2}{8}\right) \approx 0.811$$

Algorithme **ID3**

Exemple :

Supposons que $\mathcal{X}$ soit constitué de 14 exemples, 9 positifs et 5 négatifs. Parmi ces exemples, 6 positifs et 2 négatifs prennent la valeur *oui* pour l'attribut a, tandis que les autres exemples prennent la valeur *non* pour cet attribut.

$$Gain(\mathcal{X}, a) = H(\mathcal{X}) - \frac{8}{14} H(\mathcal{X}_{a=oui}) - \frac{6}{14} H(\mathcal{X}_{a=non})$$

$$H(\mathcal{X}_{a=oui}) = -\left(\frac{6}{8}\log_2\frac{6}{8} + \frac{2}{8}\log_2\frac{2}{8}\right) \approx 0.811$$

$$H(\mathcal{X}_{a=non}) = -\left(\frac{3}{6}\log_2\frac{3}{6} + \frac{3}{6}\log_2\frac{3}{6}\right) = 1.0$$

Algorithme **ID3**

Exemple :

Supposons que $\mathcal{X}$ soit constitué de 14 exemples, 9 positifs et 5 négatifs. Parmi ces exemples, 6 positifs et 2 négatifs prennent la valeur *oui* pour l'attribut a, tandis que les autres exemples prennent la valeur *non* pour cet attribut.

$$Gain(\mathcal{X}, a) = H(\mathcal{X}) - \frac{8}{14} H(\mathcal{X}_{a=oui}) - \frac{6}{14} H(\mathcal{X}_{a=non})$$

$$Gain(\mathcal{X}, a) \approx 0.940 - \frac{8}{14} 0.811 - \frac{6}{14} \approx 0.048$$

Algorithme **ID3**

Entrées : ensemble d'exemples $\mathcal{X}$, ensemble d'attributs $\mathcal{A}$
Sorties : racine d'un noeud de l'arbre de décision

Créer un noeud racine;
si *tous les élements de* $\mathcal{X}$ *sont positifs* **alors**
    racine. étiquette $\leftarrow \oplus$;
    retourner racine;
si *tous les élements de* $\mathcal{X}$ *sont négatifs* **alors**
    racine. étiquette $\leftarrow \ominus$;
    retourner racine;
si $\mathcal{A} = \emptyset$ **alors**
    racine. étiquette $\leftarrow$ valeur la plus présente de la classe parmi les $\mathcal{X}$;
    retourner racine;
$\vdots$

Algorithme **ID3**

Entrées : ensemble d'exemples $\mathcal{X}$, ensemble d'attributs $\mathcal{A}$
Sorties : racine d'un noeud de l'arbre de décision

Créer un noeud racine;
si *tous les élements de* $\mathcal{X}$ *sont positifs* **alors**
  racine. étiquette $\leftarrow \oplus$;
  retourner racine;
si *tous les élements de* $\mathcal{X}$ *sont négatifs* **alors**
  racine. étiquette $\leftarrow \ominus$;
  retourner racine;
si $\mathcal{A} = \emptyset$ **alors**
  racine. étiquette $\leftarrow$ valeur la plus présente de la classe parmi les $\mathcal{X}$;
  retourner racine;
$\vdots$

Algorithme **ID3**

Entrées : ensemble d'exemples $\mathcal{X}$, ensemble d'attributs $\mathcal{A}$
Sorties : racine d'un noeud de l'arbre de décision

```
Créer un noeud racine;
si tous les élements de X sont positifs alors
    racine. étiquette ← ⊕;
    retourner racine;
si tous les élements de X sont négatifs alors
    racine. étiquette ← ⊖;
    retourner racine;
si A = ∅ alors
    racine. étiquette ← valeur la plus présente de la classe parmi les X;
    retourner racine;
⋮
```

Algorithme **ID3**

Entrées : ensemble d'exemples $\mathcal{X}$, ensemble d'attributs $\mathcal{A}$
Sorties : racine d'un noeud de l'arbre de décision

```
Créer un noeud racine;
si tous les élements de X sont positifs alors
    racine. étiquette ← ⊕;
    retourner racine;
si tous les élements de X sont négatifs alors
    racine. étiquette ← ⊖;
    retourner racine;
si A = ∅ alors
    racine. étiquette ← valeur la plus présente de la classe parmi les X;
    retourner racine;
⋮
```

Algorithme **ID3**

Entrées : ensemble d'exemples $\mathcal{X}$, ensemble d'attributs $\mathcal{A}$
Sorties : racine d'un noeud de l'arbre de décision

```
Créer un noeud racine;
si tous les élements de 𝒳 sont positifs alors
    racine. étiquette ← ⊕;
    retourner racine;
si tous les élements de 𝒳 sont négatifs alors
    racine. étiquette ← ⊖;
    retourner racine;
si 𝒜 = ∅ alors
    racine. étiquette ← valeur la plus présente de la classe parmi les 𝒳;
    retourner racine;
⋮
```

Algorithme **ID3**

Entrées : ensemble d'exemples $\mathcal{X}$, ensemble d'attributs $\mathcal{A}$
Sorties : racine d'un noeud de l'arbre de décision

```
Créer un noeud racine;
si tous les élements de 𝒳 sont positifs alors
    racine. étiquette ← ⊕;
    retourner racine;
si tous les élements de 𝒳 sont négatifs alors
    racine. étiquette ← ⊖;
    retourner racine;
si 𝒜 = ∅ alors
    racine. étiquette ← valeur la plus présente de la classe parmi les 𝒳;
    retourner racine;
⋮
```

Algorithme **ID3**

Entrées : ensemble d'exemples $\mathcal{X}$, ensemble d'attributs $\mathcal{A}$
Sorties : racine d'un noeud de l'arbre de décision

```
⋮
a* ← arg max_{a∈A} Gain(X, a) ;
racine. étiquette ← a*;
pour toutes les valeurs v_i de a* faire
    ajouter une branche à racine correspondant à la valeur v_i;
    former X_{a*=v_i} ⊆ X dont l'attribut a* vaut v_i;
    si X_{a*=v_i} = ∅ alors
        a l'extrémité de cette branche, mettre une feuille étiquetée avec la
        valeur la plus présente de la classe parmi les X;
    sinon
        a l'extrémité de cette branche, mettre ID3(X_{a*=v_i}, A − a*);
retourner racine;
```

In math notation: line 2 is $a^* \leftarrow \arg\max_{a \in \mathcal{A}} Gain(\mathcal{X}, a)$; line 6 forms $\mathcal{X}_{a^*=v_i} \subseteq \mathcal{X}$; line 7 tests $\mathcal{X}_{a^*=v_i} = \emptyset$; line 10 calls $ID3(\mathcal{X}_{a^*=v_i}, \mathcal{A} - a^*)$.

Algorithme **ID3**

Entrées : ensemble d'exemples $\mathcal{X}$, ensemble d'attributs $\mathcal{A}$
Sorties : racine d'un noeud de l'arbre de décision

$\vdots$
$a^* \leftarrow \arg\max_{a \in \mathcal{A}} Gain(\mathcal{X}, a)$;
racine. étiquette $\leftarrow a^*$;
pour *toutes les valeurs* v_i *de* a^* **faire**
    ajouter une branche à racine correspondant à la valeur v_i;
    former $\mathcal{X}_{a^*=v_i} \subseteq \mathcal{X}$ dont l'attribut a^* vaut v_i;
    si $\mathcal{X}_{a^*=v_i} = \emptyset$ **alors**
        a l'extrémité de cette branche, mettre une feuille étiquetée avec la valeur la plus présente de la classe parmi les $\mathcal{X}$;
    sinon
        a l'extrémité de cette branche, mettre ID3($\mathcal{X}_{a^*=v_i}$, $\mathcal{A} - a^*$);
retourner racine;

Algorithme **ID3**

Entrées : ensemble d'exemples $\mathcal{X}$, ensemble d'attributs $\mathcal{A}$
Sorties : racine d'un noeud de l'arbre de décision

$\vdots$
$a^* \leftarrow \arg\max_{a \in \mathcal{A}} Gain(\mathcal{X}, a)$;
racine. étiquette $\leftarrow a^*$;
pour *toutes les valeurs v_i de a^** **faire**
  ajouter une branche à racine correspondant à la valeur v_i;
  former $\mathcal{X}_{a^*=v_i} \subseteq \mathcal{X}$ dont l'attribut a^* vaut v_i;
  si $\mathcal{X}_{a^*=v_i} = \emptyset$ **alors**
    a l'extrémité de cette branche, mettre une feuille étiquetée avec la valeur la plus présente de la classe parmi les $\mathcal{X}$;
  sinon
    a l'extrémité de cette branche, mettre ID3($\mathcal{X}_{a^*=v_i}$, $\mathcal{A} - a^*$);
retourner racine;

Algorithme **ID3**

Entrées : ensemble d'exemples $\mathcal{X}$, ensemble d'attributs $\mathcal{A}$
Sorties : racine d'un noeud de l'arbre de décision

⋮
$a^* \leftarrow \arg\max_{a \in \mathcal{A}} Gain(\mathcal{X}, a)$;
racine. étiquette $\leftarrow a^*$;
pour *toutes les valeurs* v_i *de* a^* **faire**
  ajouter une branche à racine correspondant à la valeur v_i;
  former $\mathcal{X}_{a^*=v_i} \subseteq \mathcal{X}$ dont l'attribut a^* vaut v_i;
  si $\mathcal{X}_{a^*=v_i} = \emptyset$ **alors**
    a l'extrémité de cette branche, mettre une feuille étiquetée avec la valeur la plus présente de la classe parmi les $\mathcal{X}$;
  sinon
    a l'extrémité de cette branche, mettre ID3($\mathcal{X}_{a^*=v_i}$, $\mathcal{A} - a^*$);
retourner racine;

Algorithme **ID3**

Entrées : ensemble d'exemples $\mathcal{X}$, ensemble d'attributs $\mathcal{A}$
Sorties : racine d'un noeud de l'arbre de décision

$\vdots$
$a^* \leftarrow \arg\max_{a \in \mathcal{A}} Gain(\mathcal{X}, a)$;
racine. étiquette $\leftarrow a^*$;
pour *toutes les valeurs* v_i *de* a^* **faire**
    ajouter une branche à racine correspondant à la valeur v_i;
    former $\mathcal{X}_{a^*=v_i} \subseteq \mathcal{X}$ dont l'attribut a^* vaut v_i;
    si $\mathcal{X}_{a^*=v_i} = \emptyset$ **alors**
        a l'extrémité de cette branche, mettre une feuille étiquetée avec la valeur la plus présente de la classe parmi les $\mathcal{X}$;
    sinon
        a l'extrémité de cette branche, mettre ID3($\mathcal{X}_{a^*=v_i}$, $\mathcal{A} - a^*$);
retourner racine;

Algorithme **ID3**

Entrées : ensemble d'exemples $\mathcal{X}$, ensemble d'attributs $\mathcal{A}$
Sorties : racine d'un noeud de l'arbre de décision

$\vdots$

$a^* \leftarrow \arg\max_{a \in \mathcal{A}} Gain(\mathcal{X}, a)$;

racine. étiquette $\leftarrow a^*$;

pour *toutes les valeurs* v_i *de* a^* **faire**

- ajouter une branche à racine correspondant à la valeur v_i;
- former $\mathcal{X}_{a^*=v_i} \subseteq \mathcal{X}$ dont l'attribut a^* vaut v_i;
- **si** $\mathcal{X}_{a^*=v_i} = \emptyset$ **alors**
  - a l'extrémité de cette branche, mettre une feuille étiquetée avec la valeur la plus présente de la classe parmi les $\mathcal{X}$;
- **sinon**
  - a l'extrémité de cette branche, mettre ID3($\mathcal{X}_{a^*=v_i}$, $\mathcal{A} - a^*$);

retourner racine;

Algorithme **ID3**

Entrées : ensemble d'exemples $\mathcal{X}$, ensemble d'attributs $\mathcal{A}$
Sorties : racine d'un noeud de l'arbre de décision

$\vdots$
$a^* \leftarrow \arg\max_{a \in \mathcal{A}} Gain(\mathcal{X}, a)$;
racine. étiquette $\leftarrow a^*$;
pour *toutes les valeurs* v_i *de* a^* **faire**
  ajouter une branche à racine correspondant à la valeur v_i;
  former $\mathcal{X}_{a^*=v_i} \subseteq \mathcal{X}$ dont l'attribut a^* vaut v_i;
  si $\mathcal{X}_{a^*=v_i} = \emptyset$ **alors**
    a l'extrémité de cette branche, mettre une feuille étiquetée avec la valeur la plus présente de la classe parmi les $\mathcal{X}$;
  sinon
    a l'extrémité de cette branche, mettre ID3($\mathcal{X}_{a^*=v_i}$, $\mathcal{A} - a^*$);
retourner racine;

Algorithme **ID3** : Déroulement

On considère l'ensemble exemplees suivant. L'attribut cible est donc "*Jouer au tennis ?*"

Jour	Ciel	Température	Humidité	Vent	Jouer au tennis ?
1	Ensoleillé	Chaude	Elevée	Faible	Non
2	Ensoleillé	Chaude	Elevée	Fort	Non
3	Couvert	Chaude	Elevée	Faible	Oui
4	Pluie	Tiède	Elevée	Faible	Oui
5	Pluie	Fraîche	Normale	Faible	Oui
6	Pluie	Fraîche	Normale	Fort	Non
7	Couvert	Fraîche	Normale	Fort	Oui
8	Ensoleillé	Tiède	Elevée	Faible	Non
9	Ensoleillé	Fraîche	Normale	Faible	Oui
10	Pluie	Tiède	Normale	Faible	Oui
11	Ensoleillé	Tiède	Normale	Fort	Oui
12	Couvert	Tiède	Elevée	Fort	Oui
13	Couvert	Chaude	Normale	Faible	Oui
14	Pluie	Tiède	Elevée	Fort	Non

TABLE – Jeu de données *jouer au tennis ?*

Algorithme **ID3** : Déroulement

Attribut	Gain
Ciel	0.246

Algorithme **ID3** : Déroulement

Attribut	Gain
Ciel	0.246
Humidité	0.151

Algorithme **ID3** : Déroulement

Attribut	Gain
Ciel	0.246
Humidité	0.151
Vent	0.048

Algorithme **ID3** : Déroulement

Attribut	Gain
Ciel	0.246
Humidité	0.151
Vent	0.048
Tempétarure	0.029

Algorithme **ID3** : Déroulement

Attribut	Gain
Ciel	0.246
Humidité	0.151
Vent	0.048
Tempétarure	0.029

Ciel ?

Algorithme **ID3** : Déroulement

Ciel ?

Attribut	Gain
Humidité	0.970

Algorithme **ID3** : Déroulement

Attribut	Gain
Humidité	0.970
Tempétarure	0.570

Ciel ?

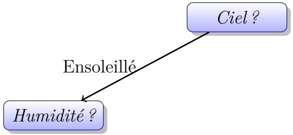
Algorithme **ID3** : Déroulement

Ciel ?

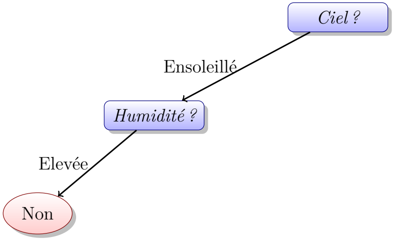
Attribut	Gain
Humidité	0.970
Tempétarure	0.570
Vent	0.019

Algorithme **ID3** : Déroulement

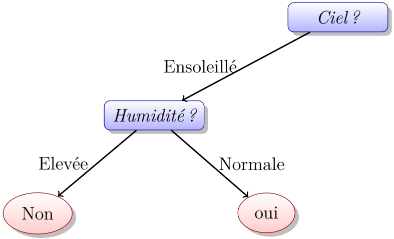
Attribut	Gain
Humidité	0.970
Tempétarure	0.570
Vent	0.019



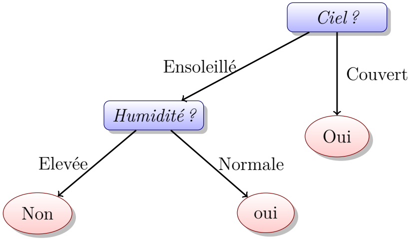
Algorithme **ID3** : Déroulement



Algorithme **ID3** : Déroulement

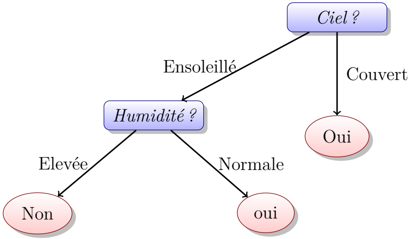


Algorithme **ID3** : Déroulement



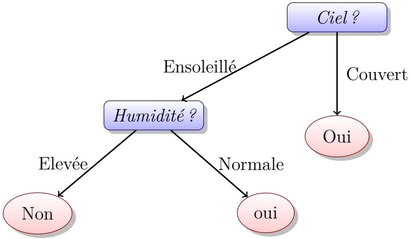
Algorithme **ID3** : Déroulement

Attribut	Gain
Vent	0.970



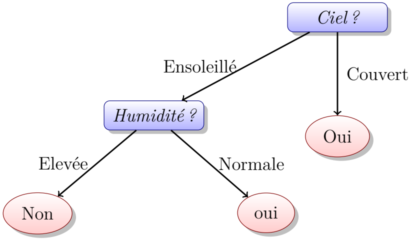
Algorithme **ID3** : Déroulement

Attribut	Gain
Vent	0.970
Humidité	0.0.019



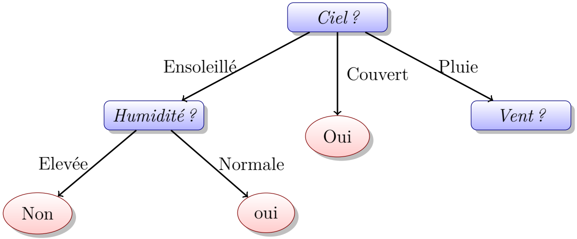
Algorithme **ID3** : Déroulement

Attribut	Gain
Vent	0.970
Humidité	0.0.019
Tempétarure | 0.019 |

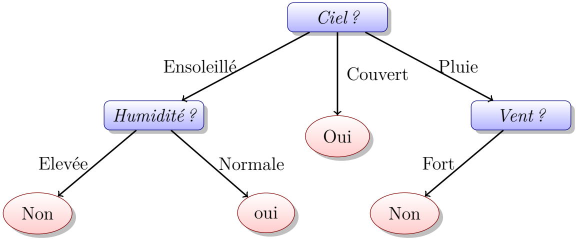


Algorithme **ID3** : Déroulement

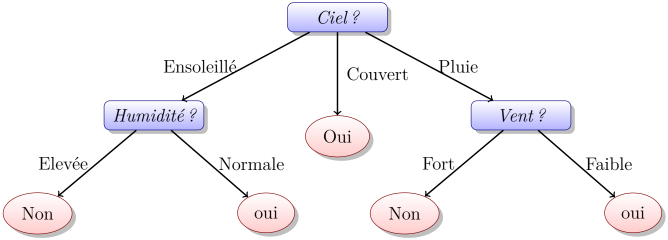
Attribut	Gain
Vent	0.970
Humidité	0.0.019
Tempétarure	0.019



Algorithme **ID3** : Déroulement



Algorithme **ID3** : Déroulement



Ensemble d'Apprentissage, de Test et de Validation

- La tâche de classification repose sur la disponibilité d'un corpus initial
$$\Omega = d_1, \ldots, d_{|\Omega|} \subset D$$
de exemples pré-classifiés sous
$$C = \{c_1, \ldots, c_{|C|}\}$$
- Les valeurs de la fonction $\Phi : D \times C \to \{T, F\}$ sont connues pour chaque paire $\langle d_j, c_i \rangle$.
- Un exemples d_j est un exemple positif de c_i si
$$\Phi(d_j, c_i) = T$$
,
- Un exemple négatif de c_i si
$$\Phi(d_j, c_i) = F$$

Ensemble d'Apprentissage, de Test et de Validation

- La tâche de classification repose sur la disponibilité d'un corpus initial

$$\Omega = d_1, \ldots, d_{|\Omega|} \subset D$$

de exemples pré-classifiés sous

$$C = \{c_1, \ldots, c_{|C|}\}$$

- Les valeurs de la fonction $\Phi : D \times C \rightarrow \{T, F\}$ sont connues pour chaque paire $\langle d_j, c_i \rangle$.
- Un exemples d_j est un exemple positif de c_i si

$$\Phi(d_j, c_i) = T$$

,

- Un exemple négatif de c_i si

$$\Phi(d_j, c_i) = F$$

Ensemble d'Apprentissage, de Test et de Validation

- La tâche de classification repose sur la disponibilité d'un corpus initial

$$\Omega = d_1, \ldots, d_{|\Omega|} \subset D$$

de exemples pré-classifiés sous

$$C = \{c_1, \ldots, c_{|C|}\}$$

- Les valeurs de la fonction $\Phi : D \times C \rightarrow \{T, F\}$ sont connues pour chaque paire $\langle d_j, c_i \rangle$.
- Un exemples d_j est un exemple positif de c_i si

$$\Phi\left(d_j, c_i\right) = T$$

,

- Un exemple négatif de c_i si

$$\Phi\left(d_j, c_i\right) = F$$

Ensemble d'Apprentissage, de Test et de Validation

- La tâche de classification repose sur la disponibilité d'un corpus initial

$$\Omega = d_1, \ldots, d_{|\Omega|} \subset D$$

  de exemples pré-classifiés sous

$$C = \{c_1, \ldots, c_{|C|}\}$$

- Les valeurs de la fonction $\Phi : D \times C \rightarrow \{T, F\}$ sont connues pour chaque paire $\langle d_j, c_i \rangle$.
- Un exemples d_j est un exemple positif de c_i si

$$\Phi\left(d_j, c_i\right) = T$$

  ,
- Un exemple négatif de c_i si

$$\Phi\left(d_j, c_i\right) = F$$

Ensemble d'Apprentissage, de Test

- Une fois qu'un classifieur Φ a été construit il est souhaitable d'évaluer son efficacité.
- Avant la construction du classifieur, le corpus initial est divisé en deux séries, pas nécessairement de taille égale : un ensemble d'apprentissage et un ensemble de test.

Ensemble d'Apprentissage, de Test

- Une fois qu'un classifieur Φ a été construit il est souhaitable d'évaluer son efficacité.
- Avant la construction du classifieur, le corpus initial est divisé en deux séries, pas nécessairement de taille égale : un ensemble d'apprentissage et un ensemble de test.

Ensemble d'Apprentissage, de Test

- Une fois qu'un classifieur Φ a été construit il est souhaitable d'évaluer son efficacité.
- Avant la construction du classifieur, le corpus initial est divisé en deux séries, pas nécessairement de taille égale : un ensemble d'apprentissage et un ensemble de test.

Ensemble d'apprentissage

$$EA = \{d_1, \ldots, d_{|EA|}\}$$

Le classifieur Φ pour les catégories $C = \{c_1, \ldots, c_{|C|}\}$ est construit en observant les caractéristiques de ces exemples.

Ensemble d'Apprentissage, de Test

- Une fois qu'un classifieur Φ a été construit il est souhaitable d'évaluer son efficacité.
- Avant la construction du classifieur, le corpus initial est divisé en deux séries, pas nécessairement de taille égale : un ensemble d'apprentissage et un ensemble de test.

Ensemble de Test

$$ET = \{d_{|EA+1|}, \ldots, d_{|\Omega|}\}$$

Il est utilisé pour tester l'efficacité des classifieurs. Chaque $d_j \in ET$ est donné au classifieur, et les décisions du classifieur $\Phi\left(d_j, c_i\right)$ sont comparées avec les décisions d'expert.

Ensemble d'Apprentissage, de Test

Remarque :

- Les documents de ET ne peuvent pas participer d'une façon quelconque à la construction d'induction du classement.
- Si cette condition n'était satisfaite, les résultats expérimentaux obtenus seraient probablement trop bons, et l'évaluation n'aurait donc pas de caractère scientifique.
- La validation est une phase indispensable à tout processus d'apprentissage.

Ensemble d'Apprentissage, de Test

Remarque :

- Les documents de ET ne peuvent pas participer d'une façon quelconque à la construction d'induction du classement.
- Si cette condition n'était satisfaite, les résultats expérimentaux obtenus seraient probablement trop bons, et l'évaluation n'aurait donc pas de caractère scientifique.
- La validation est une phase indispensable à tout processus d'apprentissage.
- Elle consiste à vérifier que le modèle construit sur l'ensemble d'apprentissage permet de classer tout individu avec le minimum d'erreurs possible.

Ensemble d'Apprentissage, de Test

Remarque :

- Les documents de ET ne peuvent pas participer d'une façon quelconque à la construction d'induction du classement.
- Si cette condition n'était satisfaite, les résultats expérimentaux obtenus seraient probablement trop bons, et l'évaluation n'aurait donc pas de caractère scientifique.
- La validation est une phase indispensable à tout processus d'apprentissage.
- Elle consiste à vérifier que le modèle construit sur l'ensemble d'apprentissage permet de classer tout individu avec le minimum d'erreurs possible.

Ensemble d'Apprentissage, de Test

Remarque :

- Les documents de ET ne peuvent pas participer d'une façon quelconque à la construction d'induction du classement.
- Si cette condition n'était satisfaite, les résultats expérimentaux obtenus seraient probablement trop bons, et l'évaluation n'aurait donc pas de caractère scientifique.
- La validation est une phase indispensable à tout processus d'apprentissage.
- Elle consiste à vérifier que le modèle construit sur l'ensemble d'apprentissage permet de classer tout individu avec le minimum d'erreurs possible.

Ensemble de validation

Validation par test

- les résultats de l'évaluation seraient une estimation pessimiste de la performance réelle,
- L'ensemble d'apprentissage permet de générer le modèle, l'ensemble de test permet d'évaluer l'erreur réelle du modèle sur un ensemble indépendant évitant ainsi un biais d'apprentissage.
- S'il s'agit de tester plusieurs modèles et de les comparer, on peut sélectionner le meilleur modèle selon ses performances sur l'ensemble de validation et ensuite évaluer l'erreur réelle sur l'ensemble de test.

Ensemble de validation

Validation par test

- les résultats de l'évaluation seraient une estimation pessimiste de la performance réelle,
- L'ensemble d'apprentissage permet de générer le modèle, l'ensemble de test permet d'évaluer l'erreur réelle du modèle sur un ensemble indépendant évitant ainsi un biais d'apprentissage.
- S'il s'agit de tester plusieurs modèles et de les comparer, on peut sélectionner le meilleur modèle selon ses performances sur l'ensemble de validation et ensuite évaluer l'erreur réelle sur l'ensemble de test.

Ensemble de validation

Validation par test

- les résultats de l'évaluation seraient une estimation pessimiste de la performance réelle,
- L'ensemble d'apprentissage permet de générer le modèle, l'ensemble de test permet d'évaluer l'erreur réelle du modèle sur un ensemble indépendant évitant ainsi un biais d'apprentissage.
- S'il s'agit de tester plusieurs modèles et de les comparer, on peut sélectionner le meilleur modèle selon ses performances sur l'ensemble de validation et ensuite évaluer l'erreur réelle sur l'ensemble de test.

Ensemble de validation

Validation Croisée

- Les k différents classifieurs $\Phi_1, \cdots, \Phi_k$ sont construits par le partitionnement initial du corpus en k ensembles disjoints $ET_1, \cdots, ET_k$.
- La validation par test est ensuite appliquée de façon itérative sur les paires $\langle EA_i = \Omega - ET_i, ET_i \rangle$.
- L'efficacité finale est obtenue par le calcul individuel de l'efficacité de $\Phi_1, \cdots, \Phi_k$.
- La validation croisée ne construit pas de modèle utilisable, elle estime juste l'erreur réelle.
- En général le nombre k de parties est fixé à 10.

Ensemble de validation

Validation Croisée

- Les k différents classifieurs $\Phi_1, \cdots, \Phi_k$ sont construits par le partitionnement initial du corpus en k ensembles disjoints $ET_1, \cdots, ET_k$.
- La validation par test est ensuite appliquée de façon itérative sur les paires $\langle EA_i = \Omega - ET_i, ET_i \rangle$.
- L'efficacité finale est obtenue par le calcul individuel de l'efficacité de $\Phi_1, \cdots, \Phi_k$.
- La validation croisée ne construit pas de modèle utilisable, elle estime juste l'erreur réelle.
- En général le nombre k de parties est fixé à 10.

Mesures de d'éfficacité

L'efficacité de classification se mesure généralement par les paramètres classiques du domaine de la recherche d'information :

- La précision,
- Le rappel.

Précision

- La précision c'est un ratio entre le nombre de documents pertinents trouvés et le nombre total de documents trouvés.
- Elle mesure le bruit, et plus elle est proche de 100%, moins il y a de bruit, et donc meilleure est la réponse.

Mesures de d'éfficacité

L'efficacité de classification se mesure généralement par les paramètres classiques du domaine de la recherche d'information :

- La précision,
- Le rappel.

Précision

- La précision c'est un ratio entre le nombre de documents pertinents trouvés et le nombre total de documents trouvés.
- Elle mesure le bruit, et plus elle est proche de 100%, moins il y a de bruit, et donc meilleure est la réponse.

Mesures de d'éfficacité

L'efficacité de classification se mesure généralement par les paramètres classiques du domaine de la recherche d'information :

- La précision,
- Le rappel.

Rappel

- Le rappel est un ratio entre le nombre de documents pertinents trouvés et le nombre de documents pertinents présents dans la base.
- Plus il est proche de 100%, moins il y a de silence, et meilleure est la réponse.

Mesures de d'éfficacité

L'efficacité de classification se mesure généralement par les paramètres classiques du domaine de la recherche d'information :

- La précision,
- Le rappel.

Rappel

- Le rappel est un ratio entre le nombre de documents pertinents trouvés et le nombre de documents pertinents présents dans la base.
- Plus il est proche de 100%, moins il y a de silence, et meilleure est la réponse.

Mesures de d'éfficacité

Ces mésures peuvent être estimées par un tableau qui montre toutes les possibilités d'un classifieur

Catégorie c_i		Classement de l'Expert	
		Vrai	**Faux**
Jugement du Classifieur	**Positif**	TP_i	FP_i
	Négatif	FN_i	TN_i

- FP_i : le nombre de documents de test mal classés dans la catégorie c_i ,
- TN_i : le nombre de documents bien classés qui n'appartiennent pas à la catégorie c_i,
- TP_i : le nombre de documents bien classés qui appartiennent à la catégorie c_i,et
- FN_i : le nombre de documents de catégorie c_i non classés par le classifieur.

Mesures de d'éfficacité

Ces mésures peuvent être estimées par un tableau qui montre toutes les possibilités d'un classifieur

Catégorie c_i		Classement de l'Expert	
		Vrai	**Faux**
Jugement du Classifieur	**Positif**	TP_i	FP_i
	Négatif	FN_i	TN_i

- FP_i : le nombre de documents de test mal classés dans la catégorie c_i ,
- TN_i : le nombre de documents bien classés qui n'appartiennent pas à la catégorie c_i,
- TP_i : le nombre de documents bien classés qui appartiennent à la catégorie c_i,et
- FN_i : le nombre de documents de catégorie c_i non classés par le classifieur.

Mesures de d'éfficacité

Ces mésures peuvent être estimées par un tableau qui montre toutes les possibilités d'un classifieur

Catégorie c_i		Classement de l'Expert	
		Vrai	**Faux**
Jugement du Classifieur	**Positif**	TP_i	FP_i
	Négatif	FN_i	TN_i

- FP_i : le nombre de documents de test mal classés dans la catégorie c_i ,
- TN_i : le nombre de documents bien classés qui n'appartiennent pas à la catégorie c_i,
- TP_i : le nombre de documents bien classés qui appartiennent à la catégorie c_i,et
- FN_i : le nombre de documents de catégorie c_i non classés par le classifieur.

Mesures de d'éfficacité

Ces mésures peuvent être estimées par un tableau qui montre toutes les possibilités d'un classifieur

Catégorie c_i		Classement de l'Expert	
		Vrai	**Faux**
Jugement du Classifieur	**Positif**	TP_i	FP_i
	Négatif	FN_i	TN_i

- FP_i : le nombre de documents de test mal classés dans la catégorie c_i ,
- TN_i : le nombre de documents bien classés qui n'appartiennent pas à la catégorie c_i,
- TP_i : le nombre de documents bien classés qui appartiennent à la catégorie c_i,et
- FN_i : le nombre de documents de catégorie c_i non classés par le classifieur.

Mesures de d'éfficacité

La précision et le rappel peuvent être exprimés localement de la façon suivante :

$$\pi_i = \frac{TP_i}{TP_i + FP_i}$$

$$\rho_i = \frac{TP_i}{TP_i + FN_i}$$

Deux approches sont adoptées pour exprimer globalement la précision et le rappel :

- Macro-moyenne : donner un poids égal à toutes les classes.
- Micro-moyenne : Donner un poids proportionnel à la fréquence de chaque classe.

Mesures de d'éfficacité

La précision et le rappel peuvent être exprimés localement de la façon suivante :

$$\pi_i = \frac{TP_i}{TP_i + FP_i}$$

$$\rho_i = \frac{TP_i}{TP_i + FN_i}$$

Deux approches sont adoptées pour exprimer globalement la précision et le rappel :

- Macro-moyenne : donner un poids égal à toutes les classes.
- Micro-moyenne : Donner un poids proportionnel à la fréquence de chaque classe.

Mesures de d'éfficacité

Catégories $C = \{c_i, \cdots, c_{\vert C\vert}\}$		Classement de l'Expert	
		Vrai	**Faux**
Jugement du Classifieur	**Positif**	$TP = \sum_{i=1}^{\vert C\vert} TP_i$	$FP = \sum_{i=1}^{\vert C\vert} FP_i$
	Négatif	$FN = \sum_{i=1}^{\vert C\vert} FN_i$	$TN = \sum_{i=1}^{\vert C\vert} TN_i$

TABLE – Table de contingence globale

Mesures de d'éfficacité

La micro-moyenne *(ang : micro-averaging)* calcule les mesures rappel et précision de façon globale : si l'on considère les tables de contingences associées à chaque catégorie, cela revient à sommer les cases TP et FP de chaque catégorie pour obtenir la table de contingence globale. Les différentes mesures comme le π et ρ sont ensuite calculées à partir des valeurs cumulées.

$$\pi^{\mu} = \frac{TP}{TP + FP} = \frac{\sum_{i=1}^{|C|} TP_i}{\sum_{i=1}^{|C|}(TP_i + FP_i)}$$

$$\rho^{\mu} = \frac{TP}{TP + FN} = \frac{\sum_{i=1}^{|C|} TP_i}{\sum_{i=1}^{|C|}(TP_i + FN_i)}$$

Mesures de d'éfficacité

La macro-moyenne *(ang : macro-averaging)* évalue d'abord indépendamment chaque catégorie. Ensuite, la performance globale du classifieur est calculée en faisant la moyenne des mesures individuelles. Les différentes catégorie sont alors la même importance. La précision et le rappel macro-moyenne sont calculés comme suit :

$$\pi^M = \frac{\sum_{i=1}^{|C|} \pi_i}{|C|}$$

$$\rho^M = \frac{\sum_{i=1}^{|C|} \rho_i}{|C|}$$

Mesures de d'éfficacité

F_1 peut être considéré comme le degré relatif d'importance attribué à la précision et au rappel :

$$F_1 = \frac{2\pi\rho}{\pi + \rho}$$

Le taux de succès *(ang : accuracy rate)* désigne le pourcentage d'exemples bien classés par le classifieur :

$$A = \frac{TP + TN}{TP + TN + FP + FN}$$

Taux d'erreur *(ang : error rate)* désigne le pourcentage d'exemples mal classés.

$$E = \frac{FP + FN}{TP + TN + FP + FN} = 1 - A$$