XAI - TME 1

Explication par génération d'exemples contre-factuels

L'objectif du TME est d'implémenter la méthode de génération d'explications par exemples contre-factuels appelée Growing Spheres et de faire des expérimentations en l'appliquant à divers jeux de données et classifieurs. Pour commencer, le TME utilise des données synthétiques en deux dimensions permettant de visualiser les exemples contre-factuels générés.

1. **Données**

   Implémenter une fonction qui génère une base de données synthétique de type `halfMoons`, telle qu'illustrée sur la figure ci-dessous pour quatre niveaux de bruit différents (sans bruit à gauche).

   La classe du haut contient les points de coordonnées $(\cos(t), \sin(t))$ pour t compris entre 0 et π, auxquelles on ajoute un bruit gaussien. La classe du bas contient les points de coordonnées $(1 - \cos(t), 0.5 - \sin(t))$ pour t compris entre 0 et π, auxquelles on ajoute un bruit gaussien.

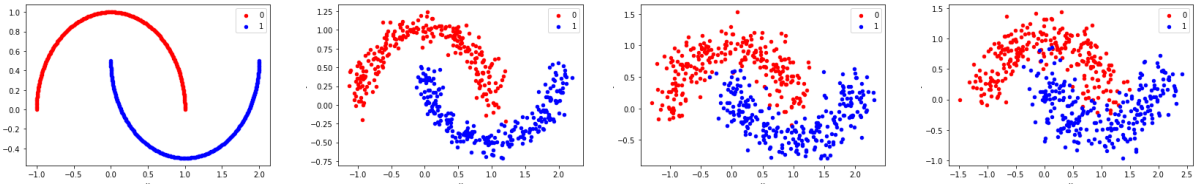

Figure 1 – Données `HalfMoons`

2. **Classifieurs**

   En utilisant par exemple `sklearn`, implémenter quelques classifieurs (par exemple k-ppv, forêt aléatoire, SVM) et les entraîner le cas échéant sur une base d'apprentissage constituée d'un sous-ensemble des données générées.

   On doit alors disposer d'une fonction `predict` permettant de prédire la classe pour toute donnée.

3. **Génération d'exemples contrefactuels**

   Implémenter l'algorithme GS dont le pseudo-code est fourni en annexe.

4. **Visualisation**

   Implémenter une fonction permettant de visualiser les données, la frontière de décision, une donnée à expliquer ainsi l'exemple contrefactuel qui lui est associé.

5. **Expérimentations sur données artificielles**

   Définir un plan d'expérience permettant d'examiner la pertinence des exemples contre-factuels générés.

   Il pourra par exemple examiner la stabilité de l'algorithme par rapport à sa composante aléatoire (qui peut varier selon le classifieur utilisé et la complexité de sa frontière de décision), faire varier la donnée à expliquer, le classifieur, les paramètres de l'algorithme (en particulier le cas extrême sans prise en compte de la parcimonie).

6. **Expérimentations sur données classiques**

   Appliquer à d'autres jeux de données de dimensions supérieures disponibles dans `sklearn` comme `wine` ou `Breast Cancer Wisconsin` et évaluer subjectivement la qualité des explications générées.

   Remarque : l'exemple contrefactuel n'est pas en lui-même une explication, celle-ci est constituée des attributs dont la valeur est modifiée.

Pseudo-code de l'algorithme Growing Spheres

(Laugel et al., 18)

On note $SL(x, a_0, a_1)$ la couche sphérique (*Spherical Layer*) de centre x et de rayon interne a_0 et de rayon externe a_1 : $SL(x, a_0, a_1) = \{z \in \mathcal{X} | a_0 \leq \|x - z\|_2 \leq a_1\}$.

Algorithm 1 Algorithme Growing Spheres Generation

Require: $f : \mathcal{X} \to \{-1, 1\}$ a binary classifier
Require: $x \in \mathcal{X}$ an observation to be interpreted
Require: Hyperparameters : η, n
Ensure: enemy e
  Generate $(z_i)_{i \leq n}$ in $SL(x, 0, \eta)$ following a uniform distribution
  while $\exists e \in (z_i)_{i \leq n} | f(e) \neq f(x)$ **do**
    $\eta = \eta / 2$
    Generate $(z_i)_{i \leq n}$ in $SL(x, 0, \eta)$ following a uniform distribution
  end while
  Set $a_0 = \eta$, $a_1 = 2\eta$
  while $\nexists e \in (z_i)_{i \leq n} | f(e) \neq f(x)$ **do**
    Generate $(z_i)_{i \leq n}$ in $SL(x, a_0, a_1)$ following a uniform distribution
    $a_0 = a_1$
    $a_1 = a_1 + \eta$
  end while
  return e, the l_2-closest generated enemy from x

Algorithm 2 Algorithme Growing Spheres Feature Selection

Require: $f : \mathcal{X} \to \{-1, 1\}$ a binary classifier
Require: $x \in \mathcal{X}$ an observation to be interpreted
Require: $e \in \mathcal{X} | f(e) \neq f(x)$ the solution of Algorithm **??**
Ensure: enemy e^*
  Set $e' = e$
  while $f(e') \neq f(x)$ **do**
    $e^* = e'$
    $i = \arg\min_{j \in \{1, \ldots, d\}, e'_j \neq x_j} |e'_j - x_j|$
    Update $e'_i = x_i$
  end while
  return e^*