ABClass : Une approche d'apprentissage multi-instances pour les séquences

Manel Zoghlami[1,2] Sabeur Aridhi[3]
Mondher Maddouri[4] Engelbert Mephu Nguifo [1]

[1] Université Clermont Auvergne, CNRS, LIMOS, BP 10125, 63173 Clermont Ferrand, France
[2] Université Tunis El Manar, Faculté des sciences de Tunis, LIPAH, 1060 Tunis, Tunisie
[3] Université de Lorraine, CNRS, Inria, LORIA, F-54000 Nancy, France
[4] Université de Jeddah, Faculté d'administration des affaires, BP 80327, 21589 Jeddah, KSA

manel.zoghlami@etu.uca.fr

Résumé

Dans le cas du problème de l'apprentissage multi-instances (MI) pour les séquences, les données d'apprentissage consistent en un ensemble de sacs où chaque sac contient un ensemble d'instances/séquences. Dans certaines applications du monde réel, comme la bioinformatique, comparer un couple aléatoire de séquences n'a aucun sens. En fait, chaque instance de chaque sac peut avoir une relation structurelle et/ou fonctionnelle avec d'autres instances dans d'autres sacs. Ainsi, la tâche de classification doit prendre en compte la relation entre les instances sémantiquement liées à travers les sacs. Dans cet article, nous présentons ABClass, une nouvelle approche de classification MI des séquences. Chaque séquence est représentée par un vecteur d'attributs extraits à partir de l'ensemble des instances qui lui sont liées. Pour chaque séquence du sac à prédire, un classifieur discriminant est appliqué afin de calculer un résultat de classification partiel. Ensuite, une méthode d'agrégation est appliquée afin de générer le résultat final. Nous avons appliqué ABClass pour résoudre le problème de la prédiction de la résistance aux rayonnements ionisants (RRI) chez les bactéries. Les résultats expérimentaux sont satisfaisants.

Mots clés

apprentissage multi-instances, séquences protéiques, prédiction de la résistance aux rayonnements ionisants chez les bactéries

Abstract

In Multiple Instance Learning (MIL) problem for sequence data, the learning data consist of a set of bags where each bag contains a set of instances/sequences. In some real world applications such as bioinformatics comparing a random couple of sequences makes no sense. In fact, each instance of each bag may have structural and/or functional relationship with other instances in other bags. Thus, the classification task should take into account the relation between semantically related instances across bags. In this paper, we present ABClass, a novel MIL approach for sequence data classification. Each sequence is represented by one vector of attributes extracted from the set of related instances. For each sequence of the unknown bag, a discriminative classifier is applied in order to compute a partial classification result. Then, an aggregation method is applied in order to generate the final result. We applied ABClass to solve the problem of bacterial Ionizing Radiation Resistance (IRR) prediction. The experimental results were satisfactory.

Keywords

multiple instance learning, protein sequences, prediction of bacterial ionizing radiation resistance

1 Introduction

L'apprentissage multi-instances (MI) est une variante des méthodes d'apprentissage classiques qui peut être utilisée pour résoudre des problèmes dans lesquels les étiquettes sont affectées à des sacs, c'est-à-dire un ensemble d'instances, plutôt que des instances individuelles. Une hypothèse majeure de la plupart des méthodes d'apprentissage MI existantes est que chaque sac contient un ensemble d'instances qui sont distribuées indépendamment. De nombreuses applications du monde réel telles que la bioinformatique, la fouille de sites Web et la fouille de textes doivent traiter des données séquentielles (données sous forme de séquences). Lorsque le problème abordé peut être formulé comme un problème MI, chaque instance de chaque sac peut avoir une relation structurelle et/ou fonctionnelle avec d'autres instances dans d'autres sacs. Le problème que nous voulons résoudre dans ce travail est le problème d'apprentissage MI pour les séquences qui présentent des relations / dépendances à travers les sacs.

Ce travail a été initialement proposé pour résoudre le problème de la prédiction de la résistance aux rayonnements ionisants (RRI) chez les bactéries [21] [4]. L'objectif est d'apprendre un classifieur qui classe une bactérie soit comme bactérie résistante aux radiations ionisantes

(BRRI) soit comme bactérie sensible aux radiations ionisantes (BSRI). Les BRRI sont importantes en biotechnologie. Elles pourraient être utilisées pour la bioremédiation de déchets radioactifs et dans l'industrie thérapeutique [5] [10]. Cependant, un nombre limité de travaux d'apprentissage automatique a été proposé pour résoucre le problème de la prédiction de la RRI chez les bactéries. Ce problème pourrait être formalisé en tant que problème d'apprentissage MI : chaque bactérie est représentée par un ensemble de séquences protéiques. Les bactéries représentent les sacs et les séquences protéiques représentent les instances. En particulier, le contenu de chaque séquence protéique peut être différent d'une bactérie à une autre, par exemple, chaque sac contient la protéine appelée *endonucléase III*, mais elle est exprimée différemment d'un sac à un autre : il s'agit de protéines orthologues [7]. Afin d'apprendre l'étiquette d'une bactérie inconnue, comparer un couple aléatoire de séquences n'a pas de sens, il est plutôt préférable de comparer les séquences protéiques qui ont une relation/dépendance fonctionnelle : les protéines orthologues. Ce travail traite donc le problème d'apprentissage MI ayant les trois critères suivants : (1) les instances à l'intérieur des sacs sont des séquences, nous devons donc traiter le format de la représentation des données, (2) les instances peuvent avoir des dépendances entre les sacs et (3) toutes les instances à l'intérieur d'un sac contribuent à définir l'étiquette du sac.

L'hypothèse standard de l'apprentissage MI indique qu'un sac est positif si au moins une de ses instances est positive alors que dans chaque sac négatif toutes les instances sont négatives [6]. Ceci n'est pas garanti dans certains domaines donc des hypothèses alternatives ont été proposées [9]. Particulièrement, l'hypothèse standard n'est pas adaptée au problème relatif à la RRI car une instance positive n'est pas suffisante pour classer un sac comme positif. Nous optons plutôt à l'hypothèse collective [2] [9] : toutes les instances contribuent à la définition de l'étiquette du sac.

Nous proposons dans ce travail une formalisation du problème de l'apprentissage MI pour les séquences. Nous présentons aussi une approche naïve et une nouvelle approche appelée ABClass. ABClass effectue d'abord une étape de prétraitement des séquences en entrée qui consiste à extraire des motifs à partir de chaque ensemble de séquences liées. Ces motifs seront utilisés comme attributs pour construire une matrice binaire pour chaque ensemble où chaque ligne correspond à une séquence. Ensuite, un classifieur discriminant est appliqué aux séquences d'un sac inconnu afin de prédire son étiquette. Nous décrivons l'algorithme de notre approche et nous présentons une étude expérimentale en l'appliquant au problème de la prédiction de la RRI chez les bactéries .

Le reste de cet article est organisé comme suit. La section 2 définit le problème de l'apprentissage MI pour les séquences. Dans la section 3, nous présentons un aperçu de quelques travaux relatifs aux problèmes MI. Dans la section 4, nous décrivons l'approche que nous proposons pour la classification MI des séquences ayant des dépendances à travers les sacs. Dans la section 5, nous décrivons notre environnement expérimental et nous discutons les résultats obtenus.

2 Contexte

Dans cette section, nous présentons les notions de base relatives à l'apprentissage MI pour les séquences. Nous décrivons d'abord la terminologie et la formulation de notre problème. Ensuite, nous présentons un cas d'utilisation simple qui sert d'exemple illustratif tout au long de ce document.

2.1 Formulation du problème

Une séquence est une liste ordonnée d'évènements. Un évènement peut être représenté comme une valeur symbolique, une valeur numérique, un vecteur de valeurs ou un type de données complexe [19]. Il existe de nombreux types de séquences comme les séquences symboliques, les séries temporelles simples et les séries temporelles multivariées [19]. Dans notre travail, nous nous intéressons aux séquences symboliques puisque les séquences protéiques sont décrites à l'aide de symboles (acides aminés). On note Σ l'*alphabet* défini par un ensemble fini de caractères ou de symboles. Une séquence symbolique simple est donc définie comme une liste ordonnée de symboles de Σ.

Soit BD une base de données d'apprentissage qui contient un ensemble de n sacs étiquetés : $BD = \{(B_i, Y_i), i = 1, 2 \ldots, n\}$ où $Y_i = \{-1, 1\}$ est l'étiquette du sac B_i. Les instances de B_i sont des séquences et sont notées B_{ij}. Formellement $B_i = \{B_{ij}, j = 1, 2 \ldots, m_{Bi}\}$, où m_{Bi} est le nombre total d'instances dans le sac B_i. Nous notons que les sacs ne contiennent pas nécessairement le même nombre d'instances. Le problème étudié dans ce travail consiste à apprendre un classifieur MI à partir de BD. Étant donné un sac inconnu $Q = \{Q_k, k = 1, 2 \ldots, q\}$, où q est le nombre total d'instances dans Q, le classifieur doit utiliser les séquences dans ce sac et celles dans chaque sac de BD afin de prédire l'étiquette de Q.

Nous notons qu'il existe une relation notée $\Re$ qui relie les instances à travers les différents sacs. Elle est définie en fonction du domaine d'application. Pour représenter cette relation, nous optons pour une représentation à base d'indice. Nous notons que cette notation ne signifie pas que les instances sont ordonnées. En fait, une étape de prétraitement attribue un indice aux instances de chaque sac selon la façon suivante : chaque instance B_{ij} d'un sac B_i est reliée par $\Re$ à l'instance B_{hj} d'un autre sac B_h dans BD. Une instance peut ne pas avoir d'instances correspondantes dans certains sacs, c'est-à-dire qu'une séquence est liée à zéro ou une séquence par sac. La relation $\Re$ pourrait être généralisée pour traiter les problèmes où chaque instance a plus d'une instance cible par sac. La notation d'indice telle que décrite précédemment ne sera pas appropriée dans ce cas.

2.2 Exemple illustratif

Afin d'illustrer notre approche, nous nous appuyons sur l'exemple suivant. Soit $\Sigma = \{A, B, \ldots, Z\}$ un alphabet. Soit $BD = \{(B_1, +1), (B_2, +1), (B_3, -1), (B_4, -1), (B_5, -1)\}$ une base d'apprentissage contenant 5 sacs (B_1 et B_2 sont des sacs positifs, B_3, B_4 et B_5 sont des sacs négatifs). Initialement, les sacs contiennent les séquences suivantes :
$B_1 =$ {ABMSCD, EFNOGH, RUVR}
$B_2 =$ {CCGHDDEF, EABZQCD}
$B_3 =$ {GHWMY, ACDXYZ}
$B_4 =$ {ABIJYZ, KLSSO, EFYRTAB}
$B_5 =$ {EFFVGH, KLSNAB}
Nous utilisons d'abord la relation inter-sacs $\Re$ pour représenter les instances reliées en utilisant la notation d'indice décrite précédemment.

$$B_1 = \begin{cases} B_{11} = \mathbf{AB}MS\mathbf{CD} \\ B_{12} = \mathbf{EF}NO\mathbf{GH} \\ B_{13} = RUVR \end{cases} \quad B_2 = \begin{cases} B_{21} = E\mathbf{AB}ZQ\mathbf{CD} \\ B_{22} = CC\mathbf{GH}DD\mathbf{EF} \end{cases}$$

$$B_3 = \begin{cases} B_{31} = A\mathbf{CD}X\mathbf{YZ} \\ B_{32} = \mathbf{GH}WMY \end{cases} \quad B_4 = \begin{cases} B_{41} = \mathbf{AB}IJ\mathbf{YZ} \\ B_{42} = \mathbf{EF}YRT\mathbf{AB} \\ B_{43} = \mathbf{KL}SSO \end{cases}$$

$$B_5 = \begin{cases} B_{52} = \mathbf{EF}FV\mathbf{GH} \\ B_{53} = \mathbf{KL}SN\mathbf{AB} \end{cases}$$

Le but ici est de prédire l'étiquette d'un sac inconnu $Q = \{Q_1, Q_2, Q_3\}$ où :

$$Q = \begin{cases} Q_1 = \mathbf{AB}WX\mathbf{CD} \\ Q_2 = \mathbf{EF}XY\mathbf{GH}N \\ Q_3 = \mathbf{KL}OF \end{cases}$$

3 Travaux existants

Plusieurs algorithmes d'apprentissage MI ont été proposés incluant Diverse Density (DD) [15], MI-SVM [3], Citation-kNN [18] et MILKDE [8]. Un état de l'art des approches d'apprentissage MI avec une étude comparative pourrait être trouvé dans [1] et [2].
L'idée principale de l'approche DD [15] est de trouver les points qui sont proches d'au moins une instance de chaque sac positif et qui sont loin des instances des sacs négatifs. On cherche ensuite le point optimal selon une mesure définie par les auteurs et qui porte le même nom que l'algorithme. Cette mesure prend en considération le nombre de sacs positifs ayant des instances proches du point en question et la distance entre ce point et les instances négatives. L'approche MI-SVM [3] est une adaptation des machines à vecteurs de support au problème d'apprentissage MI. Elle reformule la maximisation de la marge des sacs en prenant en considération les contraintes du MI. Pour un sac positif, seule l'instance ayant la plus grande marge a un impact sur l'apprentissage. Les autres instances sont ignorées. MISMO utilise l'algorithme SMO [16] basé sur l'apprentissage par machines à vecteurs de support en conjonction avec un noyau MI [11]. L'algorithme Citation-kNN [18] est basé sur la règle des plus proches voisins et sur le concept de citation et de référence. Un sac est étiqueté non seulement selon ses voisins (les références), mais aussi selon les sacs qui le reconnaissent comme leur voisin (les citeurs). Dans [8], les auteurs présentent l'algorithme MILKDE qui se base sur l'identification des instances les plus représentatives dans chaque sac positif en utilisant un calcul de vraisemblance. Dans [20], les auteurs traitent l'apprentissage MI sur des données structurées. Ils décrivent trois scénarios des relations existantes entre les données : I-MILSD où les relations sont disponibles au niveau des instances, B-MILSD où les relations sont au niveau des sacs et BI-MILSD où les relations sont disponibles au niveau des instances et des sacs.

L'application des algorithmes MI présentés ci-dessus sur des sacs de séquences entraîne deux problèmes. Le premier problème réside dans la représentation des données à traiter. Elles sont représentées dans un format attribut-valeur. Dans le cas des séquences, la technique la plus utilisée pour transformer les données en un format attribut-valeur consiste à extraire des motifs qui servent comme attributs. Nous notons que trouver une description uniforme de toutes les instances en utilisant un ensemble de motifs n'est pas toujours une tâche facile. En apprentissage MI, cela pourrait conduire à une matrice énorme et creuse. Par exemple, dans [3], l'évaluation empirique est effectuée sur la base de données TREC9 pour la catégorisation des documents. Les termes sont utilisés pour présenter le texte. On obtient alors une matrice creuse et de grande dimension . Dans [17] et [20], un ensemble de séquences protéiques a été utilisé dans l'évaluation empirique. L'objectif est d'identifier les protéines dites *Trx-fold*. Chaque séquence est considérée comme un sac et certaines de ses sous-séquences sont considérées comme des instances. Ces sous-séquences sont alignées et représentées en utilsant 8 attributs numériques [13] [17]. Le deuxième problème avec les algorithmes présentés est qu'ils ne traitent pas les relations inter-sacs qui peuvent exister entre les instances, à l'exception des algorithmes dans [20]. Dans [20], le score d'alignement est utilisé pour identifier les relations entre les protéines : si le score entre une paire de protéines dépasse 25, alors les auteurs considèrent qu'il existe un lien entre elles. Seul l'algorithme B-MILSD qui traite l'information au niveau du sac a été utilisé dans l'étude expérimentale. Dans un travail antérieur, nous avons proposé l'algorithme MIL-ALIGN [4] qui traite le problème de la prédiction de la RRI chez les bactéries. Il utilise une technique d'alignement pour discriminer les séquences puis applique une méthode d'agrégation pour générer le résultat de prédiction final. Dans ABClass, nous représentons les sé-

quences en utilisant un format attribut-valeur qui prend en compte les dépendances entre les instances liées à travers les sacs.

4 Approches d'apprentissage MI pour les séquences

Dans cette section, nous présentons d'abord l'approche naïve pour traiter le problème de l'apprentissage MI pour les séquences ayant des relations à travers les sacs. Ensuite, nous présentons notre approche nommée ABClass.

4.1 Approche MI naïve pour les séquences

La méthode la plus simple qu'on peut utiliser pour résoudre le problème de l'apprentissage MI pour les séquences est d'utiliser des classifieurs MI standards. Cependant, les algorithmes MI couramment utilisés nécessitent une description en format attribut-valeur uniforme pour toutes les instances des différents sacs. L'approche naïve contient deux étapes. La première est une étape de prétraitement qui transforme l'ensemble des séquences en une matrice attribut-valeur où chaque ligne correspond à une séquence et chaque colonne correspond à un attribut. La deuxième étape consiste à appliquer un classifieur MI existant. La Figure 1 illustre l'approche naïve pour l'apprentissage MI pour les séquences. La technique la plus utilisée pour transformer les séquences en un format attribut-valeur consiste à extraire des motifs qui seront utilisés comme attributs.

Nous notons que trouver une description uniforme de toutes les instances utilisant un ensemble de motifs n'est pas toujours une tâche facile. Puisque notre approche naïve prend en compte les relations entre les instances à travers les sacs, l'étape de prétraitement extrait les motifs à partir de chaque ensemble d'instances reliées. L'union de ces motifs est ensuite utilisée comme attributs pour construire une matrice attribut-valeur où chaque ligne correspond à une séquence. La présence ou l'absence d'un attribut dans une séquence est notée respectivement par 1 ou 0. Il est utile de mentionner que seul un sous-ensemble des attributs utilisés est représentatif pour chaque séquence. Par conséquent, nous pouvons avoir une grande matrice creuse.

Nous appliquons l'approche naïve sur notre exemple illustratif. Nous supposons que les attributs sont des sous-séquences (longueur minimale = 2) qui se trouvent au moins dans deux instances. Soit $listeMotifs_1 = \{AB, CD, YZ\}$ la liste des motifs extraits à partir des instances $\{B_{i1}, i = 1, \ldots, 4\}$. $listeMotifs_2 = \{EF, GH\}$ est la liste des motifs extraits à partir de $\{B_{i2}, i = 1, \ldots, 5\}$ et $listeMotifs_3 = \{KL\}$ est la liste des motifs extraits à partir de $\{B_{i3}, i \in \{1, 4, 5\}\}$. L'union de $listeMotifs_1$, $listeMotifs_2$ et $listeMotifs_3$ produit la liste $listeMotifs = \{AB, CD, YZ, EF, GH, KL\}$. Afin de coder les séquences de la base d'apprentissage, nous générons la matrice attribut-valeur suivante notée M :

$$M = \begin{array}{c} \begin{array}{cccccc|cccccc|cccccc} \multicolumn{6}{c}{Instance1} & \multicolumn{6}{c}{Instance2} & \multicolumn{6}{c}{Instance3} \end{array} \\ \left(\begin{array}{cccccc|cccccc|cccccc} 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & - & - & - & - & - & - \\ 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & - & - & - & - & - & - \\ 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ - & - & - & - & - & - & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 1 \end{array}\right) \begin{array}{c} B_1 \\ B_2 \\ B_3 \\ B_4 \\ B_5 \end{array} \end{array}$$

Le taux d'éparsité de M est 77.2%.

4.2 Approche proposée : *ABClass*

Afin d'éviter l'utilisation d'un grand vecteur d'attributs pour décrire les séquences, nous présentons ABClass (pour Across Bag Sequences Classification), une nouvelle approche qui prend en compte les relations entre les instances à travers les sacs. La Figure 2 illustre le principe de ABClass. Durant la phase d'apprentissage, chaque ensemble d'instances reliées sera représenté par son propre vecteur de motifs. Cela réduit le nombre d'attributs qui ne sont pas représentatifs de la séquence traitée. En appliquant un classifieur classique (mono-instance) sur chaque vecteur d'attributs, un modèle de classification est construit. Durant la phase de prédiction, des résultats de prédiction partiels sont produits pour chaque instance du sac inconnu. Ces résultats sont ensuite agrégés pour avoir le résultat final. Lors de l'exécution de l'algorithme, nous allons utiliser les variables suivantes :

- Une matrice M pour stocker les données codées de la base d'apprentissage.
- Un vecteur QV pour stocker les données codées du sac à prédire.
- Un vecteur PV pour stocker les résultats de prédiction partiels.

ABClass est décrit dans l'Algorithme 1. La fonction *SéqLiéesEntreSacs* regroupe les instances liées à travers les sacs dans des listes. Informellement, les principales étapes de l'algorithme *ABClass* sont les suivantes :

1. Pour chaque séquence Q_k du sac inconnu Q, les instances reliées à travers les sacs sont regroupées dans une liste (lignes 1 et 2).
2. L'algorithme extrait des motifs de la liste des instances regroupées. Ces motifs sont utilisés pour coder les instances afin de créer un modèle discriminant (lignes 3 à 5).

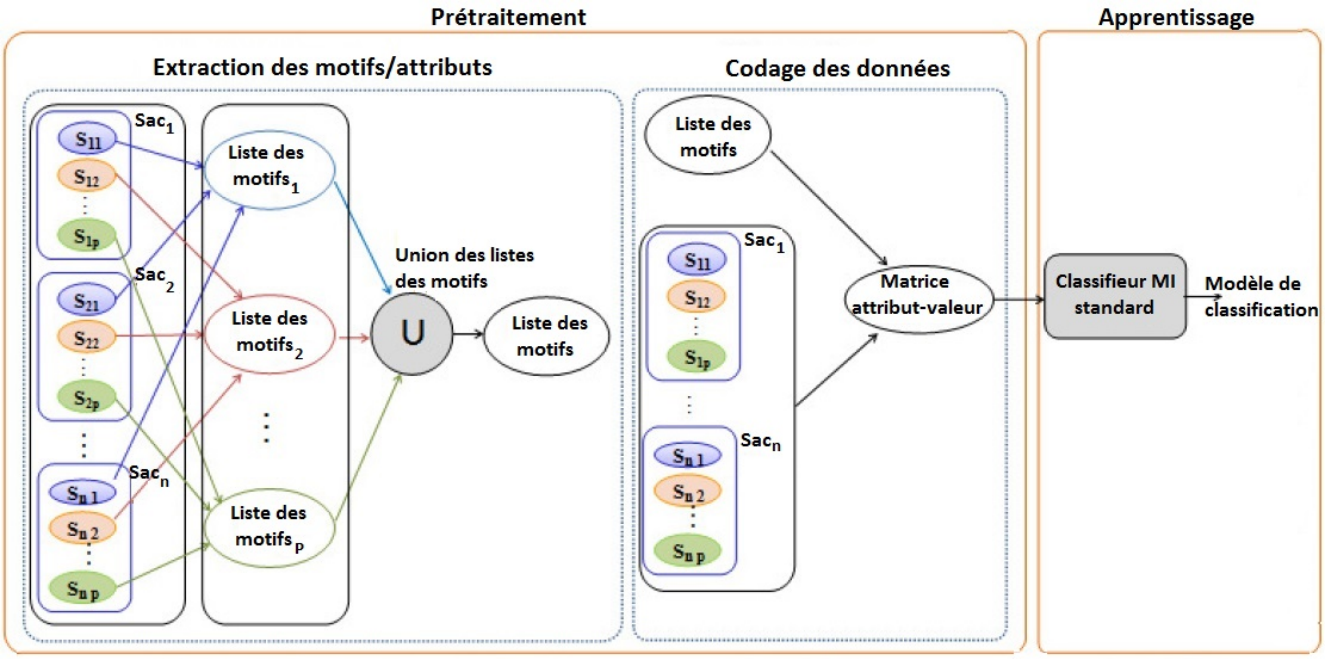


FIGURE 1 – Vue d'ensemble de l'approche MI naïve.

3. *ABClass* utilise les motifs extraits pour représenter l'instance Q_k du sac inconnu dans un vecteur QV_k, puis le compare avec le modèle correspondant. Le résultat de la comparaison est stocké dans le k ^{ime} élément du vecteur PV (lignes 6 et 7).
4. Une méthode d'agrégation est appliquée à PV pour calculer le résultat final P (ligne 9) qui consiste en une étiquette positive ou négative.

Algorithm 1 L'algorithme *ABClass*

Entrée: Base d'apprentissage $BD = \{(B_i, Y_i)|i = 1, 2, \ldots, n\}$, Sac inconnu $Q = \{Q_k|k = 1, 2, \ldots, q\}$
Sortie: Résultat de prédiction P

1: **Pour chaque** $Q_k \in Q$ **Faire**
2: $listeSéqLiées_k \leftarrow SéqLiéesEntreSacs(k, DB)$
3: $listeMotifs_k \leftarrow extraireMotifs(listeSéqLiées_k)$
4: $M_k \leftarrow coderDonnées(listeMotifs_k, listeSéqLiées_k)$
5: $Modèle_k \leftarrow générerModèle(M_k)$
6: $QV_k \leftarrow coderDonnées(listeMotifs_k, Q_k)$
7: $PV_k \leftarrow appliquerModèle(QV_k, Modèle_k)$
8: **Fin**
9: $P \leftarrow Agrégation(PV)$
10: **Retourner** P

Nous appliquons l'approche *ABClass* sur notre exemple illustratif. Puisque le sac à prédire contient 3 instances Q_1, Q_2 et Q_3, nous avons besoin de 3 itérations suivies d'une étape d'agrégation.

Itération 1 : L'algorithme regroupe l'ensemble des instances reliées et extrait les motifs correspondants.

$$listeSéqLiées_1 = \{B_{11}, B_{21}, B_{31}, B_{41}\}$$
$$listeMotifs_1 = \{AB, CD, YZ\}$$

Ensuite, il génère la matrice attribut-valeur M_1 décrivant les séquences liées à Q_1 .

$$M_1 = \begin{array}{c} \begin{matrix} AB & CD & YZ \end{matrix} \\ \begin{pmatrix} 1 & 1 & 0 \\ 1 & 1 & 0 \\ 0 & 1 & 1 \\ 1 & 0 & 1 \end{pmatrix} \end{array} \begin{matrix} B_{11} \\ B_{21} \\ B_{31} \\ B_{41} \end{matrix}$$

Le pourcentage d'éparsité de la matrice M_1 est réduit à 33% car il n'est pas nécessaire d'utiliser les motifs extraits des instances $\{B_{i2}, i = 1, .., 5\}$ et $\{B_{i3}, i \in \{1, 4, 5\}\}$ pour décrire les instances $\{B_{i1}, i = 1, .., 4\}$. Un modèle est ensuite créé en utilisant les données codées et un vecteur QV_1 est généré pour décrire Q_1.

$$QV_1 = \begin{pmatrix} 1 \\ 1 \\ 0 \end{pmatrix}$$

En appliquant le modèle au vecteur QV_1, on obtient le premier résultat de prédiction partiel et on le stocke dans le vecteur PV.

$$PV_1 \leftarrow appliquerModèle(QV_1, Modèle_1)$$

Itération 2 : La deuxième itération concerne la deuxième instance Q_2 du sac à prédire. Nous appliquons les mêmes instructions que celles décrites dans la première itération.

$$listeSéqLiées_2 = \{B_{21}, B_{22}, B_{32}, B_{42}, B_{52}\}$$
$$listeMotifs_2 = \{EF, GH\}$$

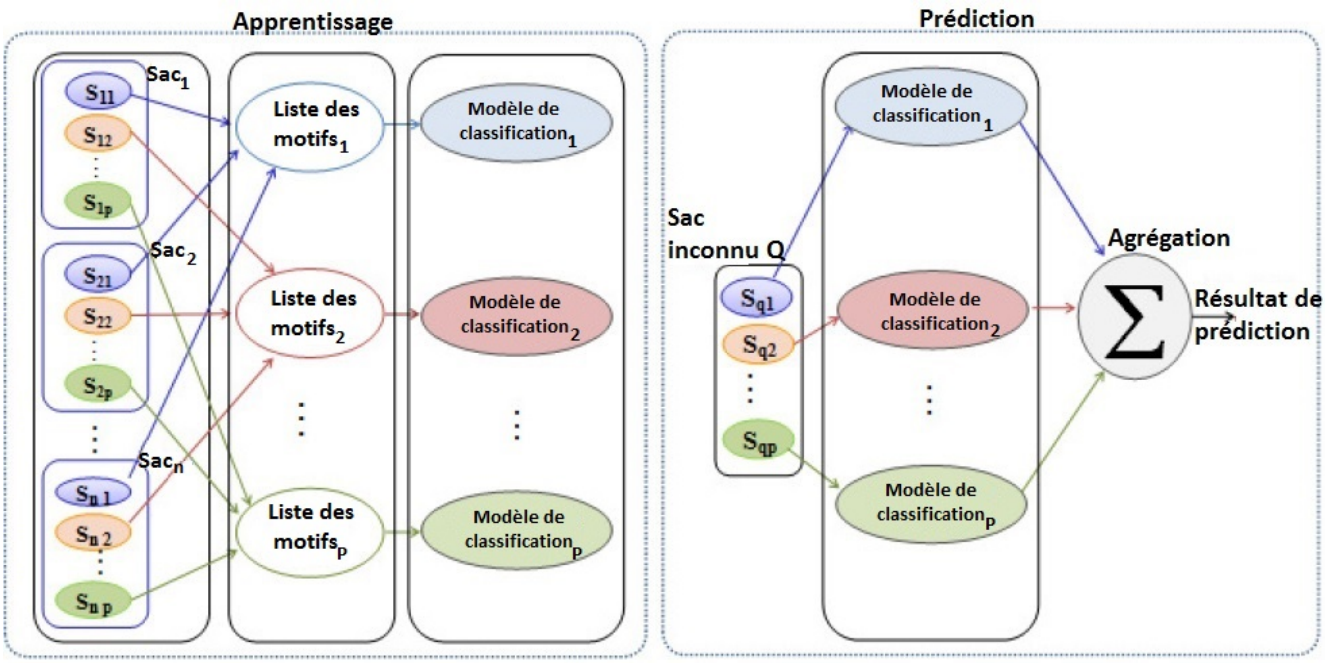


FIGURE 2 – Vue d'ensemble de l'approche *ABClass*.

$$M_2 = \begin{array}{c} \begin{array}{cc} EF & GH \end{array} \\ \begin{pmatrix} 1 & 1 \\ 1 & 1 \\ 0 & 1 \\ 1 & 0 \\ 1 & 1 \end{pmatrix} \end{array} \begin{array}{c} B_{12} \\ B_{22} \\ B_{32} \\ B_{42} \\ B_{52} \end{array}$$

$$QV_2 = \begin{pmatrix} 1 \\ 1 \end{pmatrix}$$

$$PV_2 \leftarrow appliquerModèle(QV_2, Modèle_2)$$

Itération 3 : Seuls les sacs B_1, B_4 et B_5 ont des instances reliées à Q_3.

$$listeSéqLiées_3 = \{B_{13}, B_{43}, B_{53}\}$$
$$listeMotifs_3 = \{KL\}$$

$$M_3 = \begin{array}{c} \begin{array}{c} KL \end{array} \\ \begin{pmatrix} 0 \\ 1 \\ 1 \end{pmatrix} \end{array} \begin{array}{c} B_{13} \\ B_{43} \\ B_{53} \end{array}$$

$$QV_3 = (1)$$

$$PV_3 \leftarrow appliquerModèle(QV_3, Modèl_3)$$

L'étape d'agrégation est finalement utilisée pour générer la décision finale de prédiction en utilisant les résultats partiels. Nous optons pour le vote majoritaire.

5 Etude expérimentale

Nous appliquons ABClass et l'approche naïve au problème de la prédiction de la RRI chez les bactéries qui peut être formulé comme un problème MI pour séquences. Les bactéries représentent les sacs et la structure primaire des protéines de réparation de l'ADN représentent les séquences. Une bactérie inconnue est classée comme BRRI ou BSRI. Pour nos tests, nous avons utilisé la base de données décrite dans [4]. Cet ensemble de données comprend 28 sacs (14 BRRI et 14 BSRI). Chaque bactérie/sac contient 25 à 31 instances qui correspondent aux protéines. Nous avons utilisé des classifieurs implémentés dans l'outil de fouille de données WEKA [12] afin de tester les approches proposées.

5.1 Protocole expérimental

Nous utilisons la technique d'évaluation Leave-One-Out (LOO) dans nos expérimentations. Afin d'évaluer l'approche naïve et l'approche ABClass, nous codons d'abord les séquences protéiques de chaque sac en utilisant un ensemble de motifs générés par une méthode d'extraction de motifs existante. Nous utilisons la méthode DMS [14] pour l'extraction des motifs. DMS permet de construire des motifs pouvant discriminer une famille de protéines d'une autre. Elle identifie d'abord les motifs dans les séquences protéiques. Ensuite, les motifs extraits sont filtrés afin de ne conserver que les motifs discriminants et minimaux. Une sous-chaîne est considérée comme discriminante entre la famille F et les autres familles si elle apparaît dans F significativement plus que dans les autres familles. DMS extrait des motifs selon deux seuils α et β où α est le taux minimum d'occurrences des motifs dans les séquences d'une

TABLE 1 – Éparsité de la matrice attribut-valeur utilisée dans l'approche naïve.

Paramètre d'extraction des motifs	Nombre total des motifs	Éparsité (%)
S1	671	73.9
S2	1490	73.8
S3	4562	85.7
S4	8077	91.1

famille F et β est le taux maximum d'occurrences des motifs dans toutes les séquences sauf celles de la famille F. Dans ce qui suit, nous présentons les paramètres d'extraction des motifs utilisés en fonction des valeurs de α et β :

- **Le paramètre S1 :** ($\alpha = 1$ et $\beta = 0.5$) : pour extraire des motifs fréquents avec une discrimination moyenne.
- **Le paramètre S2 :** ($\alpha = 1$ et $\beta = 1$) : pour extraire des motifs fréquents et non discriminants.
- **Le paramètre S3 :** ($\alpha = 0.5$ et $\beta = 1$) : pour extraire des motifs non discriminants avec des fréquences moyennes.
- **Le paramètre S4 :** ($\alpha = 0$ et $\beta = 1$) : pour extraire des motifs non fréquents et non discriminants.
- **Le paramètre S5 :** ($\alpha = 1$ et $\beta = 0$) : pour extraire des motifs fréquents et strictement discriminants.

5.2 Résultats

La Table 1 représente pour chaque paramètre d'extraction le nombre de motifs (longueur minimale = 3) extraits à partir de chaque ensemble de séquences de protéines orthologues. Pour le paramètre S5 ($\alpha = 1$ et $\beta = 0$), aucun motif fréquent et strictement discriminant n'a été trouvé pour la plupart des protéines. C'est pourquoi nous n'allons pas utiliser ces valeurs de α et β pour le reste des expérimentations. Nous notons que le nombre de motifs extraits augmente pour les valeurs élevées de β et les valeurs faibles de α. Comme présenté dans la Table 1, le nombre de motifs non fréquents et non discriminants est très élevé. Afin de coder les données dans l'approche naïve, on utilise l'union des motifs comme attributs. Par conséquent, la matrice attribut-valeur est grande et creuse. Nous montrons dans la Table 1 le taux d'éparsité de la matrice qui mesure le pourcentage des éléments nuls par rapport au nombre total des éléments. L'éparsité est généralement proportionnelle au nombre de motifs utilisés. Par exemple, Elle va de 73.9% avec 671 motifs à 91.1% avec 8077 motifs.

La Figure 3 montre les taux de bonne classification obtenus en appliquant l'approche naïve et l'approche ABClass. Elle montre l'impact de l'ensemble des motifs utilisés dans l'étape de prétraitement sur les résultats de prédiction. Par exemple, en utilisant le classifieur MISVM, la précision varie de 53.5% à 78.5%. Bien que les motifs extraits en utilisant le paramètre S1 soient discriminants, l'approche naïve ne fournit pas globalement de bons résultats pour ce paramètre. La raison pourrait être que le nombre de motifs discriminants pour certaines protéines est très faible (limité à 10 motifs au maximum). En utilisant l'approche naïve, le meilleur résultat est fourni par le classifieur MISMO.Les résultats des autres classifieurs MI dépendent des motifs utilisés. La plupart d'entre eux fournissent un bon résultat en utilisant le paramètre S3 (motifs non discriminants avec une fréquence moyenne). Le nombre de motifs extraits par protéine en utilisant ce paramètre est compris entre 228 et 1505. Ce nombre de motifs est acceptable pour coder une séquence protéique.

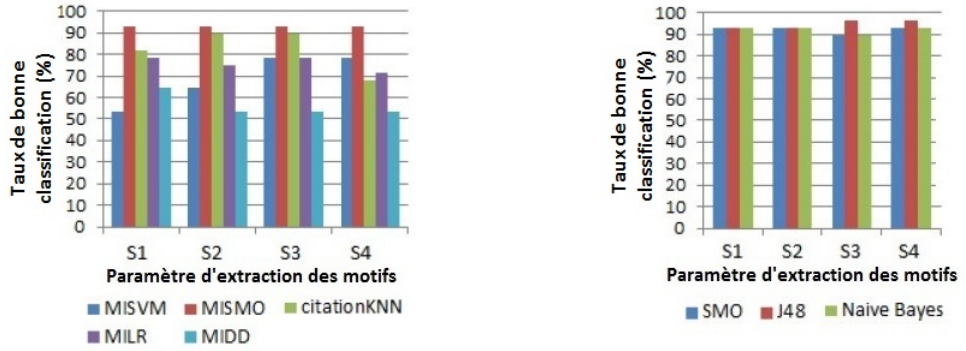
L'approche ABClass fournit globalement de bons résultats puisque le taux de bonne classification le plus faible est 89.2%. Cela montre que notre approche est efficace. Le meilleur résultat est atteint en utilisant le classifieur J48 et les paramètres d'extraction de motifs S3 et S4. En utilisant ces deux paramètres, un grand nombre de motifs non discriminants est extrait. La Table 2 représente le taux des modèles de classification qui contribuent à prédire la bonne classe de chaque bactérie en utilisant l'approche ABClass. Nous présentons ce taux pour les deux paramètres d'extraction de motifs qui fournissent les meilleurs résultats, à savoir S3 et S4. Le taux des modèles réussis pour B1, B11 et B15 est marqué en gras parce que ces trois bactéries génèrent toujours des taux faibles comparés au taux des autres bactéries. Nous notons que les résultats sont similaires à ceux trouvés dans [4]. Ces résultats peuvent aider à comprendre certaines caractéristiques des bactéries étudiées. En particulier *M. radiotolerans* (B11) et *B. abortus* (B15) présentent les taux les plus bas. Une explication biologique possible est fournie dans [4] et [21].

6 Conclusion

Dans cet article, nous avons abordé le problème d'apprentissage MI dans le cas où les instances sont des séquences. Nous nous sommes concentrés sur les données qui présentent des dépendances entre les instances des différents sacs. Nous avons décrit notre nouvelle approche nommée ABClass et nous l'avons appliquée au problème de la prédiction de la RRI chez les bactéries. Dans l'étude expérimentale, nous avons montré que l'approche proposée est efficace. Dans le futur travail, nous étudierons comment utiliser la connaissance du domaine afin d'améliorer l'efficacité de notre algorithme. Nous voulons spécifiquement définir des poids pour les séquences dans la phase d'apprentissage en utilisant la connaissance du domaine.

Remerciements

Ce travail a été partiellement soutenu par le projet franco-tunisien : Direction Générale de la Recherche Scientifique en Tunisie (DGRST) / Centre National de la Recherche Scientifique en France (CNRS) [IRRB11 / R-14-09], par la Région française d'Auvergne et par la Fédération de Recherche en Environnement [UBP / CNRSFR-3467].



(a) L'approche naïve

(b) L'approche ABClass

FIGURE 3 – Résultats de la classification en utilisant l'approche naïve et l'approche ABClass.

TABLE 2 – Taux des modèles de classification réussis en utilisant l'approche ABClass et la méthode d'évaluation LOO

ID de la bactérie	paramètre d'extraction des motifs S3			paramètre d'extraction des motifs S4		
	SMO	J48	Naive Bayes	SMO	J48	Naive Bayes
B1	**44**	**60**	**36**	**60**	**64**	**68**
B2	100	100	100	100	100	100
B3	100	90.3	100	100	90.3	100
B4	100	96.6	100	100	93.3	100
B5	100	90	100	100	90	100
B6	100	83.3	100	100	83.3	100
B7	100	93.5	100	100	93.5	100
B8	96.5	96.5	96.5	100	93.1	100
B9	100	84	100	100	84	100
B10	100	82.1	92.8	100	82.1	100
B11	**17.8**	**50**	**17.8**	**21.4**	**50**	**35.7**
B12	100	92.8	96.4	100	92.8	100
B13	88.8	66.6	70.3	88.8	66.6	77.7
B14	90	73.3	100	93.3	70	96.6
B15	**3.5**	**32.1**	**35.7**	**0**	**32.1**	**14.2**
B16	100	96.6	96.6	100	96.6	100
B17	96.2	96.2	96.2	96.2	96.2	96.2
B18	100	100	100	100	100	100
B19	100	100	100	100	100	100
B20	89.6	62	96.5	82.7	62	86.2
B21	96.5	82.7	96.5	93.1	82.7	93.1
B22	100	100	96.6	100	96.6	100
B23	100	96.7	93.5	100	96.7	100
B24	100	100	93.5	100	100	100
B25	100	96.7	93.5	100	96.7	100
B26	100	100	93.5	100	100	100
B27	100	100	100	100	100	100
B28	96.6	100	96.6	96.6	93.3	96.6

Références

[1] E. Alpaydin, V. Cheplygina, M. Loog, and D. M. Tax, Single-vs. multiple-instance classification, *Pattern Recognition*, Vol. 48, pp. 2831–2838, 2015.

[2] J. Amores, Multiple instance classification : Review, taxonomy and comparative study, *Artificial Intelligence*, Vol. 201, pp. 81–105, 2013.

[3] S. Andrews, I. Tsochantaridis, and T. Hofmann, Support vector machines for multiple-instance learning. *Advances in Neural Information Processing Systems*, pp. 561–568, MIT Press, Cambridge, MA, 2003.

[4] S. Aridhi, H. Sghaier, M. Zoghlami, M. Maddouri, and E. Mephu Nguifo, Prediction of ionizing radiation resistance in bacteria using a multiple instance learning model, *Journal of Computational Biology*, Vol. 23, pp. 10–20, 2016.

[5] H. Brim, A. Venkateswaran, H. M. Kostandarithes, J. K. Fredrickson, and M. J. Daly, Engineering Deinococcus Geothermalis for bioremediation of hightemperature radioactive waste environments, *Applied and environmental microbiology*, Vol. 69, pp. 4575–4582, 2003.

[6] T. G. Dietterich, R. H. Lathrop, and T. Lozano-Pérez, Solving the multiple instance problem with axis-parallel rectangles, *Artificial Intelligence*, Vol. 89, pp. 31–71, 1997.

[7] G. Fang, N. Bhardwaj, R. Robilotto, and M. B. Gerstein, Getting started in gene orthology and functional analysis, *PLoS computational biology*, Vol. 6, e1000703, 2010.

[8] A. W. Faria, F. G. F. Coelho, A. Silva, H. Rocha, G. Almeida, A. P. Lemos, and A. P. Braga, MILKDE : A new approach for multiple instance learning based on positive instance selection and kernel density estimation, *Engineering Applications of Artificial Intelligence*, Vol. 59, pp. 196–204, 2017.

[9] J. Foulds and E. Frank, A review of multi-instance learning assumptions, *The Knowledge Engineering Review*, Vol. 25, pp. 1–25, 2010.

[10] P. Gabani and O. V. Singh, Radiation-resistant extremophiles and their potential in biotechnology and therapeutics, *Applied microbiology and biotechnology*, Vol. 97, pp. 993–1004, 2013.

[11] T. Gärtner, P. A. Flach, A. Kowalczyk, and A. J. Smola, Multi-instance kernels, *In Proceedings of the 19th International Conference on Machine Learning*, 2002.

[12] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten, The weka data mining software : an update, *ACM SIGKDD explorations newsletter*, Vol. 11, pp. 10–18, 2009.

[13] J. Kim, E. N. Moriyama, C. G. Warr, P. J. Clyne, and J. R. Carlson, Identification of novel multitransmembrane proteins from genomic databases using quasi-periodic structural properties, *Bioinformatics*, Vol. 16, pp. 767–775, 2000.

[14] M. Maddouri and M. Elloumi, Encoding of primary structures of biological macromolecules within a data mining perspective, *Journal of Computer Science and Technology*, Vol. 19, pp. 78–88, 2004.

[15] O. Maron and T. L. Pérez, A framework for multiple-instance learning, *Advances in Neural Information Processing Systems*, Vol. 10, pp. 570–576, 1998.

[16] J. C. Platt, Fast training of support vector machines using sequential minimal optimization, *Advances in kernel methods*, pp. 185–208, 1999.

[17] Q. Tao, S. Scott, N. Vinodchandran, and T. T. Osugi, SVM-based generalized multiple-instance learning via approximate box counting, *In Proceedings of the 21st international conference on Machine learning*, Vol. 10, pp. 799–806, 2004.

[18] J. Wang and J. D. Zucker, Solving multipleinstance problem : A lazy learning approach, *In Proceedings of the 17th International Conference on Machine Learning*, pp. 1119–1125, 2000.

[19] Z. Xing, J. Pei, and E. Keogh, A brief survey on sequence classification, *ACM SIGKDD Explorations Newsletter*, Vol. 12, pp. 40–48, 2010.

[20] D. Zhang, Y. Liu, L. Si, J. Zhang, and R. D. Lawrence, Multiple instance learning on structured data, *Advances in Neural Information Processing Systems*, pp. 145–153, 2011.

[21] M. Zoghlami, S. Aridhi, M. Maddouri, and E. Mephu Nguifo, An overview of in silico methods for the prediction of ionizing radiation resistance in bacteria, *Ionizing Radiation : Advances in Research and Applications, Physics Research and Technology Series*, pp. 241-256, Nova Science Publishers Inc., 2018.