



**THÈSE DE DOCTORAT DE
L'UNIVERSITÉ SORBONNE UNIVERSITÉ**

Spécialité

Informatique

École doctorale Informatique, Télécommunications et Électronique (Paris)

Présentée par

François MEUNIER

Pour obtenir le grade de

DOCTEUR de l'UNIVERSITÉ SORBONNE UNIVERSITÉ

Sujet de la thèse :

Prédiction de phénomènes géologiques pour l'aide à la décision lors de la prise de permis d'exploitation

soutenue le 13/12/2018

devant le jury composé de :

M. Pascal PONCELET	Université de Montpellier	Rapporteur
Mme Christel VRAIN	Université d'Orléans	Rapporteur
M. Bernd AMANN	Sorbonne Université	Examineur
M. Cyril DE RUNZ	Université de Reims Champagne-Ardenne	Examineur
M. Christophe MARSALA	Sorbonne Université	Directeur de thèse
M. Laurent CASTANIÉ	Total E&P	Encadrant
M. Bruno CONCHE	Total E&P	Invité

Cette thèse a été financée par Total E&P Recherche Développement dans le cadre de la convention de collaboration de recherche N° DGRIT-UPMC C140618-P140446, REF. TOTAL FR00007724, du 1er avril 2015 au 31 mars 2018. Elle a été réalisée en collaboration entre Total E&P Recherche Développement et le LIP6 pour Sorbonne Université (ex Université Pierre et Marie Curie – Paris 6).

— Superviseur de la thèse pour l'entreprise : Laurent Castanié

— Directeur de la thèse pour Sorbonne Université (ex UPMC) : Christophe Marsala

Remerciements

Mes premiers remerciements vont naturellement à mon directeur de thèse, Christophe Marsala, pour toute son aide. Je suis ravi d'avoir travaillé en sa compagnie car, outre son appui scientifique, il a toujours été là pour me soutenir et me conseiller au cours de l'élaboration de cette thèse. Il a par ailleurs pris le temps, tout au long de mon doctorat, de discuter avec moi et de me pousser à me poser les bonnes questions.

Je remercie également Laurent Castanié qui a su me donner les bons conseils pour aborder certaines problématiques, Bruno Conche pour le temps passé à fixer les principaux axes de recherche, et Sylvain Bard-Maier qui fut à l'origine de ce travail, et sans lequel rien n'aurait été possible.

Christel Vrain et Pascal Poncelet m'ont fait l'honneur d'être rapporteurs de ma thèse, ils ont pris le temps de m'apporter des remarques pertinentes qui m'ont permis d'envisager ce travail sous un autre angle. Pour tout cela je les remercie. Je tiens également à remercier Bernd Amann et Cyril De Runz pour avoir accepté de participer à mon jury de thèse en tant qu'examineurs et, par ailleurs, Cyril pour le temps qu'il a consacré à ma recherche et aux travaux que nous avons accomplis ensemble.

Présenter l'accomplissement de plusieurs années de travail devant ce jury est pour moi un honneur.

Je remercie toutes les personnes avec qui j'ai partagé ces années de thèse, qu'elles soient du laboratoire ou de l'entreprise : Bertrand pour son humour, Tanguy pour ces conseils techniques, Arthur pour les débats politico-scientifiques, et tant d'autres à qui je ferai l'affront de ne pas les citer.

En dehors, il me serait bien sûr reproché d'omettre mes amis proches, qui auront été, chacun à leur façon, de véritables rocs que leur terre-à-terre bien connu m'aura permis de prendre du recul sur beaucoup de choses : Aymeric pour les soirées et le ski, Clément pour nos discussions sur la culture turque et Jordy pour l'ensemble de son oeuvre.

Mes derniers remerciements vont à mes parents et à mon frère, qui ont tout fait pour m'aider, et m'ont soutenu dans ce que j'ai entrepris. Je ne pourrai jamais assez vous remercier.

Résumé

Dans le domaine de l'intelligence artificielle, l'apprentissage automatique (ou *Machine Learning*) voit son importance grandir de jour en jour. Des structures toujours plus complexes tendent à être étudiées par ce biais, haussant ainsi l'information disponible au rang de connaissance exploitable.

Ce travail de doctorat propose de valoriser un certain type de données que sont les objets (structures) 3D construits à partir de maillage, en justifiant empiriquement les apports indéniables d'une extraction de sous-parties issues de ces derniers. Cet objectif est atteint en résolvant un problème de prévision par une nouvelle approche de classification supervisée pour la recommandation d'information. Au delà du résultat attendu, une justification de ce dernier est également fournie sous forme de la visualisation de sous-parties extraites discriminantes, permettant ainsi l'interprétation par le spécialiste.

Au sein du service Exploration de Total, ce besoin de classification s'applique initialement aux grandes structures 3D telles que les géo-modèles de bassins géologiques, dont les éléments pertinents tiennent effectivement de sous-parties. Lors de l'étude d'un sous-sol, les géologues cherchent, à partir de données 3D reconstituées grâce à des ondes acoustiques, à comprendre le sous-sol. Cette compréhension peut être aidée en fournissant un moyen de détecter certains types de formes au sein de ces structures. Nous proposons logiquement, afin de répondre à ce problème, un système de classification de ces structures 3D.

Grâce à une adaptation des *Time series Shapelets* et des méthodes de sélection de *features*, on parvient à ne sélectionner que les parties les plus pertinentes pour la classification souhaitée. L'idée maîtresse consiste à extraire aléatoirement un certain nombre de sous-surfaces de chaque objet 3D du jeu d'apprentissage, puis à en étudier la pertinence pour la classification souhaitée, avant d'utiliser les plus pertinents pour un apprentissage plus classique basé sur le degré d'imbrication de l'extrait dans chaque objet.

En entreprise, l'absence de justification des résultats tend à assimiler l'apprentissage automatique à une boîte noire. La méthode proposée, quant à elle, corrige ce problème, et permet la compréhension du résultat de l'aide à la décision fournie par la classification mise en place. En effet, en plus de présenter des résultats de prévision légèrement meilleurs que ceux de l'état de l'art, elle offre une visualisation des sous-parties d'objets 3D les plus discriminantes et donc les zones qui auront le plus d'influence sur la classification des données. Par la suite, nous proposons une amélioration de cette méthode sur deux axes : le premier est l'apport d'une adaptation du transfert de connaissances (ou *transfer learning*) appliqué à l'algorithme précédemment proposé ; le second est la mise en œuvre d'une méthode novatrice de sélection d'attributs, basée sur des outils issus de la théorie des sous-ensembles flous. Cette dernière s'avère être potentiellement applicable à tout type de sélection d'attributs en classification supervisée.

Les multiples résultats obtenus confirment le potentiel général de la sélection aléatoire d'attributs candidats, en particulier dans un contexte de grandes quantités de données.

Abstract

Machine learning, which is considered as an integral part of artificial intelligence, and should ultimately make computers "smart", continues to grow with time, and opens unsuspected horizons. More and more complex structures tend to be studied by this way, raising the available information to the level of exploitable knowledge.

This doctoral work proposes to valorize a particular type of data that are the 3D objects (structures) constructed from mesh, by empirically justifying the undeniable contributions of an extraction of sub-parts coming from these last one. This objective is achieved by solving a forecast problem by a new supervised classification approach for information recommendation. Beyond the expected result, a justification is also provided in the form of the visualization of sub-parts extracted discriminant, thus allowing interpretation by the specialist.

In the Total Exploration service, this classification need is initially applied to large 3D structures such as geo-models of geological basins, whose relevant elements belong to sub-parts.

During the study of a subsoil, geologists try to understand the subsoil by using 3D data reconstructed through acoustic waves. This understanding can be helped by providing a way to detect some types of shapes within these structures. We propose, in order to answer this problem, a classification system of these 3D structures.

Thanks to an adaptation of *Time series Shapelets* and *features* selection methods, it is possible to only select the most relevant parts for the targeted classification.

To summarize, the main idea is to randomly extract a certain number of sub-surfaces from each 3D object of the learning set, then to study its relevance depending on the expected classification, before using the most relevant one for a more traditional learning based on the degree of belonging of the extract in each object.

In industrial companies, the lack of justification of results tends to assimilate machine learning techniques to a black box. The proposed method, however, corrects this problem, and allows the understanding of the result of the decision support provided by the classification built.

Indeed, in addition to presenting slightly better forecast results than those provided by the state of the art, it offers a visualization of the sub-parts of the most discriminating 3D objects within the framework of the implemented classification model, and therefore the areas that will have mostly allowed to classify the data.

Subsequently, we propose an improvement of this method by two main paths : the first one is the contribution of an adaptation of the transfer of knowledge (or *transfer learning* applied to the previously proposed algorithm ; the second one is an innovative method of attribute selection, based on tools derived from fuzzy subset theory, which proves to be potentially applicable to any type of attribute selection challenge in supervised classification.

These multiple results confirm the general potential of random selection of candidate at-

tributes, especially in the context of large amounts of data.

Table des matières

Introduction générale	15
I. 3DRESC : une méthode globale de classification d'objets 3D	21
1. État de l'art	25
1.1. Classification d'objets 3D	25
1.1.1. Généralités sur la classification supervisée	25
1.1.2. Classification de structures complexes	29
1.1.3. Calage de nuages de points pour la reconnaissance de formes	30
1.1.4. Descripteurs d'objets 3D	30
1.1.5. Choix du Heat Kernel Signature	34
1.2. Méthodes pour extraire, sélectionner et valoriser des caractéristiques pertinentes	38
1.2.1. Sélection d'attributs	38
1.2.2. Distance et similarité entre histogrammes de répartition de valeurs	41
1.2.3. Algorithme des shapelets	42
1.3. Conclusion	46
2. L'approche 3DRESC : une nouvelle méthode d'extraction de sous-parties pertinentes d'objets 3D	47
2.1. Protocole explicatif	48
2.1.1. Extraction aléatoire de sous-parties	49
2.1.2. Evaluation de la pertinence des sous-parties	50
2.1.3. Calcul des attributs des objets à partir des sous-parties les plus pertinentes	52
2.1.4. Récapitulatif	52
2.2. Protocole formel	53
2.3. Descripteur local d'objet 3D et distance liée	55
2.4. Sélection aléatoire des candidats	56
2.5. Sélection des sous-parties discriminantes parmi les candidats	57
2.5.1. Sélection supervisée de sous-parties pertinentes	57
2.5.2. Sélection non supervisée de sous-parties pertinentes : valorisation de données non étiquetées	57
2.6. L'algorithme 3DRESC	59
2.6.1. Description	59
2.6.2. Complexité algorithmique	61
2.6.3. Variante avec sélection non supervisée	61

2.7. Bilan	62
3. Expérimentations sur la méthode 3DRESC	63
3.1. Protocole expérimental	64
3.2. Détermination des paramètres de l'approche	65
3.2.1. Choix du descripteur et du calcul de similarité	66
3.2.2. Heat Kernel Signature	67
3.2.3. Choix du nombre d'extraits	75
3.3. Comparaison avec d'autres méthodes de classification supervisée d'objets 3D	77
3.4. Interprétation des sous-parties discriminantes	77
3.5. Sélection non supervisée de sous-parties	78
3.6. Discussion	80
3.6.1. Amélioration de la classification	81
3.6.2. Perspectives	81
4. Application aux données Total : adaptation de l'algorithme 3DRESC aux nuages de points	83
4.1. Modifications dues au type de données	84
4.2. Quantité de données : un frein à l'application directe de l'algorithme	84
4.3. Une optimisation en calcul et en mémoire	86
4.4. Des nuages de points multiples pour un apprentissage supervisé - application de la méthode 3DRESC modifiée aux données de Total	87
4.5. Bilan	87
5. Conclusion de la partie I	91
II. Les sous-parties, supports de connaissance	93
6. L'utilisation de données provenant de contextes différents	97
6.1. Etat de l'art de transfert de connaissances	97
6.1.1. Classification en fonction des relations entre la cible et la source	98
6.1.2. Classification en fonction des labels disponibles	99
6.1.3. Classification en fonction des approches	99
6.1.4. Bilan	100
6.2. Transfer learning pour l'extraction de sous-parties d'objets 3D	100
6.3. Expérimentations sur du transfer learning	102
6.3.1. Protocole expérimental	103
6.3.2. Expériences	104
6.4. Conclusion de l'adaptation du transfer learning	106
7. Une nouvelle approche de sélection d'attributs pour la classification basée sur du prototypage flou	107
7.1. Motivations	107
7.1.1. Rappels sur la sélection d'attributs	107

7.1.2. Discussions et objectifs	108
7.2. Sélection par typicalité proposée	109
7.3. Expérimentations sur la sélection de caractéristiques par des outils de logique floue	112
7.3.1. Sélection des ressemblance, dissimilarité et agrégation	112
7.3.2. Vérification de la pertinence du couple HKS / Matusita	112
7.3.3. Combinaison de méthodes différentes	116
7.3.4. Bilan	120
7.4. Conclusion de la sélection par typicalité	121
8. Conclusion de la partie II	123
Conclusion générale	125
Annexes	129
A. Formalisations des notions en lien avec les objets 3D et leurs descripteurs	131
A.1. Les objets 3D	131
A.2. La notion d'objet ouvert/fermé	131
A.3. Les caractéristiques des descripteurs	132
B. Formations et publications	133
B.1. Formations suivies et écoles d'été/hiver	133
Formations et écoles d'été/hiver	133
B.2. Publications réalisées	133
Publications réalisées	133
Bibliographie	139

Introduction générale

Ce premier chapitre vise à présenter le cadre général de cette thèse, son contexte, ses objectifs et ses réalisations.

Présentation

L'objectif initial de cette thèse consiste en la mise en place d'un système de recommandation d'information pour l'aide à la décision lors de la prise de permis d'exploitation au sein du service géosciences de Total. Ce service est intégré à la branche Exploration et Production. Partant de la notion de système de recommandation d'information au sens traditionnel dont [Béchet, 2011] réalise un état de l'art, il s'est avéré que les manques d'information disponible et d'adaptabilité des systèmes en place ont conduit à ne pas se restreindre aux frontières habituelles du domaine de la recommandation d'information.

Ainsi, tout en restant dans l'idée de venir proposer de l'information pertinente à l'utilisateur, nous nous sommes réorienté vers un autre défi.

Ce dernier a été choisi à la croisée :

1. des besoins de Total en termes de valorisation de l'information ;
2. des récentes recherches en matière de classification supervisée et de sélection et combinaison d'attributs.

Pour l'entreprise Total, lors de l'étude géologique d'un sous-sol potentiellement riche en hydrocarbures, il est nécessaire de bien comprendre sa structure. Pour cela, les géologues créent, à partir d'images sismiques reconstituées grâce à des ondes acoustiques envoyées dans le sol (Figure 0.1), un géo-modèle 3D censé représenter les principales couches et failles de la zone (Figure 0.2). Le géo-modèle, composé de surfaces 3D correspondant aux limites entre les couches sédimentaires, ne décrit alors que la structure géométrique du sous-sol. Ces surfaces sont donc non fermées et basées sur des maillages triangulaires potentiellement recoupés (cf Annexe A).

Ce dernier est ensuite, en fonction des phénomènes que l'on y trouve, classé par le géologue selon les ensembles pertinents auxquels il se rattache. Ces ensembles correspondent à la présence ou à l'absence d'un certain phénomène géologique, comme, par exemple, des failles (normales, inverses, ...) ou des phénomènes d'interaction entre différentes couches (subduction, ...). Il s'agit donc, pour le géologue, de déterminer l'absence ou la présence de

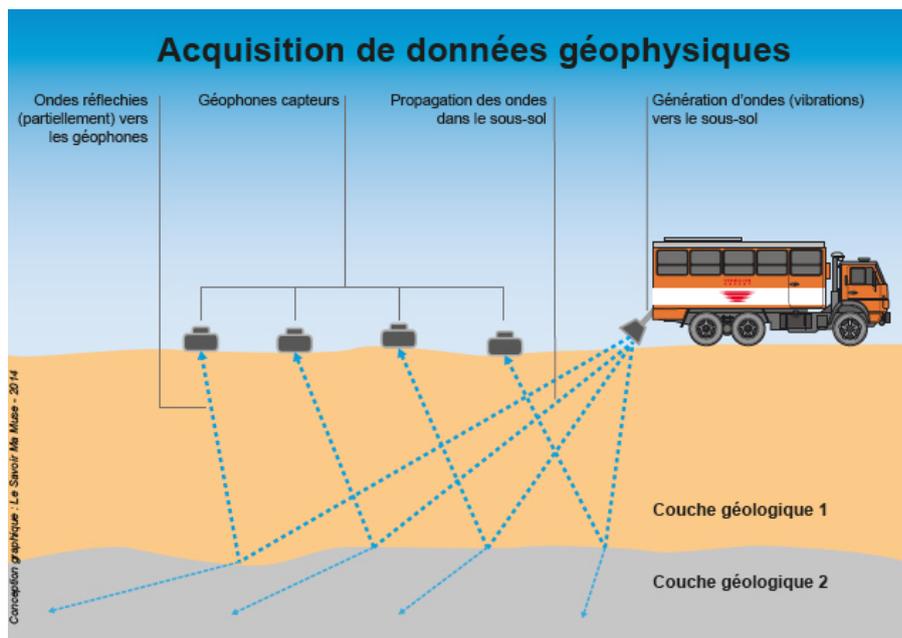


FIGURE 0.1. – Envoi d’ondes acoustiques dans le sous-sol pour l’acquisition de données géologiques - cours du CNAM source : <http://forumbachelor.free.fr/gma/GMA4/index.php?page=9>

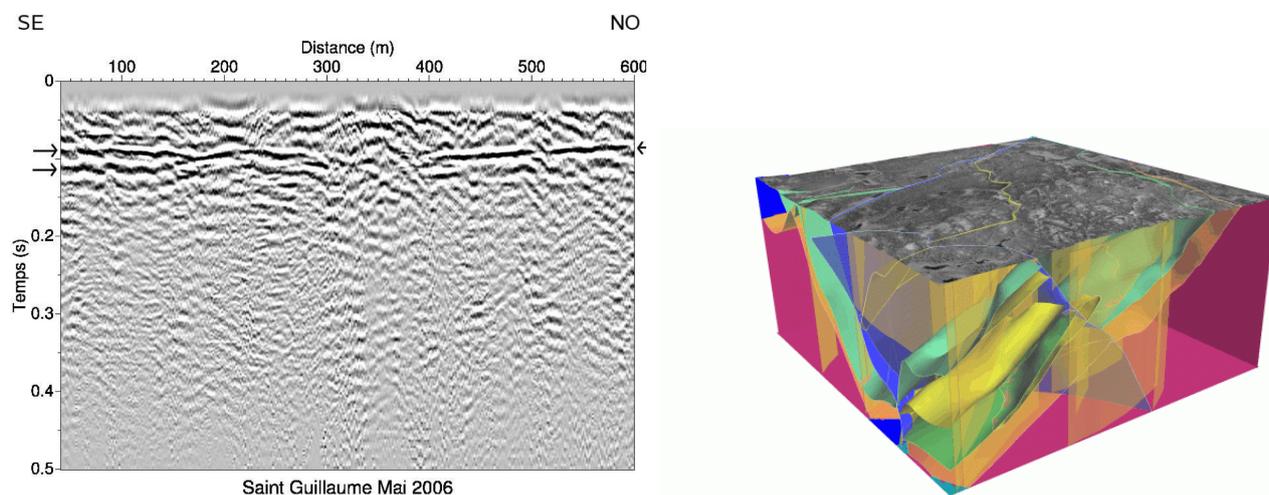


FIGURE 0.2. – Image sismique (à gauche, source : <https://omiv.osug.fr/SG/SR.html>) utilisée pour la création du géo-modèle 3D du sous-sol (à droite, source : <http://www.mirageosience.com/our-products/software-solutions/3d-geological-modelling>)

phénomènes géologiques locaux.

Aujourd'hui cette classification, comme le reste de l'étude, est réalisée manuellement par un géologue d'expérience qui connaît la zone. En effet, aucun système automatisé n'existe pour réaliser cette tâche d'étude des géo-modèles 3D. Cela représente une lourde charge de travail, qu'une automatisation pourrait grandement faciliter. De plus, certains géologues de moins d'expérience dans ce métier effectuent des classifications de moins bonne qualité (mauvaises identifications des phénomènes et mauvaises interprétations).

Toutes ces raisons justifient la nécessité de mettre en place un système qui, en plus de mieux partager la connaissance et l'expertise de l'ensemble des géologues, permet une capitalisation de la connaissance et sa pérennisation au sein de l'entreprise, en évitant les pertes d'expertise. Pouvoir utiliser des données aussi inhabituelles dans un système automatique n'est pas possible.

Ces données à étudier sont complexes par le nombre de points qui les composent, et intègrent logiquement de grandes quantités d'informations, exploitables mais peu exploitées.

Un point important est que, s'agissant d'un système d'aide à la décision dans un contexte industriel, il est nécessaire de justifier les résultats à l'utilisateur ou, au minimum, de mettre à sa disposition les outils permettant d'en comprendre en partie le résultat.

La première partie de ce travail de thèse traite de la mise en place d'un système de classification supervisée d'objets 3D, que sont les géo-modèles de couches géologiques, avec justification de la méthode et du résultat obtenu par l'exhibition des sous-parties les plus pertinentes ayant permis de les classifier. Le problème soulevé est celui du choix d'attributs pertinents et du pré-traitement que cela implique.

La seconde partie consiste à améliorer cet algorithme précédemment proposé, en utilisant d'une part des notions de transfert de connaissances, et d'autre part en proposant une nouvelle méthode de sélection d'attributs, applicable à tout type d'apprentissage supervisé.

La troisième partie propose une série d'expérimentations visant à prouver, de façon empirique, le bien fondé des propositions avancées dans cette thèse.

Objectifs et enjeux de cette thèse

Objectifs

L'objectif de cette thèse est d'aider l'utilisateur dans la détection de phénomènes géologiques locaux.

Pour ce faire, nous cherchons, dans un premier temps, à partir d'un géo-modèle fourni dont on connaît les phénomènes qui le composent, à extraire les sous-parties les plus pertinentes, puis à les utiliser pour classifier automatiquement les surfaces de géo-modèles inédits, ce en fonction de la présence ou non des phénomènes géologiques recherchés.

Dans la mesure où, à ce stade de l'étude, on ne dispose que d'informations géométriques, cela revient à classifier des structures composées de points 3D reliés par un ensemble de triangles, sans information sur une quelconque texture ou, dans le cas présent, sur la composition des couches. Il s'agit donc d'un problème d'apprentissage automatique sur des

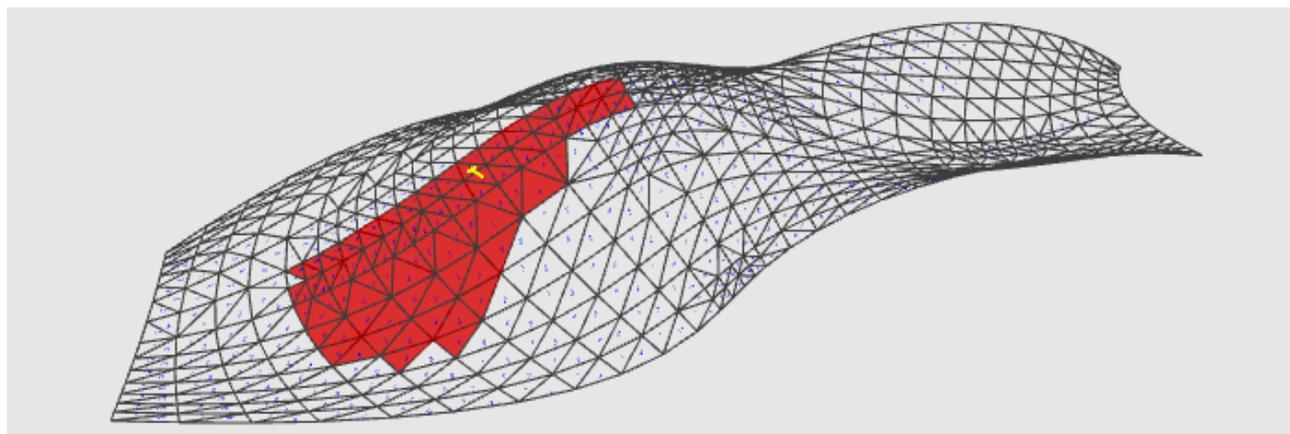


FIGURE 0.3. – Exemple de maillage triangulaire, avec une sous-partie extraite en rouge

données complexes.

Problématiques associées

L'objectif est de pouvoir extraire, de telles structures 3D maillées, certaines sous-parties, des ensembles contigus de triangles formant le maillage de la surface. Ces éléments offrent une aide à l'interprétation de la zone étudiée, en mettant en valeur une prévision basée sur des phénomènes géométriques.

Le chemin choisi est celui qui consiste à traiter les géo-modèles comme des entités indépendantes, et donc de chercher à extraire des sous-surfaces caractéristiques de leurs classes, qui servent également de constructeur d'attributs.

L'étude se décompose en 3 parties, en fonction des postulats que l'on décide d'imposer sur la classification. En effet, cette dernière, dans sa version originale, suppose une constance de la fonction de répartition de probabilité qui subdivise l'espace en différentes classes. En apprentissage supervisé, la répartition de probabilité correspond à la répartition théorique à la base de la distribution des éléments dans chaque classe, que la fonction de classification cherchera à retrouver par les données. C'est ce modèle qui a été choisi en premier lieu. Par ailleurs, un autre angle d'attaque est d'admettre la constance pour cette répartition lorsqu'il s'agit du même géo-modèle (propriétés conservées), mais avec modification en cas de passage d'un géo-modèle à un autre (dépendance spatiale). Enfin, et dernier postulat possible pour la répartition de probabilité, supposer que l'on a affaire à un *drift* [Gama et al., 2014] qui, à un certain moment, modifie la répartition initiale (dépendance temporelle). Cette dernière approche ne semble ici pas pertinente, dans la mesure où les échelles de temps géologiques sont bien plus grandes que les nôtres.

- On a donc, comme répartitions de probabilité à étudier :
- Une répartition constante pour tous les géo-modèles ;

— Une répartition dépendante du géo-modèle, ce qui nécessite des méthodes de *transfer learning* [Pan and Yang, 2010].

Par ailleurs, un challenge est de ne disposer que de très peu de données annotées, tout en restant dans le même domaine de classification (c'est-à-dire la fonction de classification d'éléments à découvrir). Pour résoudre ce problème, une sélection non supervisée des sous-surfaces pertinentes est aussi proposée.

Enfin, les méthodes traditionnelles de sélection d'attributs utilisant très souvent les mêmes outils, nous proposons une nouvelle méthode, basée sur des outils issus de la logique floue, pour sélectionner les sous-surfaces pertinentes.

L'apprentissage automatique supervisé offre des possibilités pour aider à classifier des données quel qu'en soit le type. Néanmoins, pour les objets 3D, les méthodes actuelles ne permettent pas, sans connaissance préalable, de comprendre quelles sous-parties d'un objet peuvent être discriminantes pour la présence du phénomène étudié, et d'expliquer pourquoi elles le sont.

Ce problème revient fréquemment en apprentissage, pour lequel la résolution est perçue du point de vue de l'utilisateur comme une boîte noire, dont la confiance ne peut lui être attribuée qu'au regard de ses performances passées. Il devient donc important d'expliquer le résultat de la classification à l'utilisateur, afin que celui-ci, faisant confiance au modèle, valide sa décision, et donc accepte l'apport de ce dernier.

Les conditions d'application de la méthode proposée sont basées sur les hypothèses suivantes :

- les éléments d'apprentissage sont des surfaces 3D de maillages triangulaires irréguliers ;
- la classification est basée sur la présence de phénomènes / ensembles de phénomènes locaux ;
- l'apprentissage supervisé s'applique sans changement de loi de répartition des classes dans l'ensemble étudié.

La première partie de notre travail s'articule autour de plusieurs domaines, sous-parties intégrantes du problème à résoudre. Un certain nombre de questions se posent, avec notamment, pour ne citer que les plus importantes :

- comment représenter les phénomènes locaux que l'on souhaite détecter ?
- comment classifier les objets 3D ?
- comment tenir compte des sous-parties comme attributs essentiels ?
- comment permettre à l'utilisateur de comprendre le choix du classifieur¹ construit / appris ?

La classification nécessite généralement un pré-traitement (du moins si l'on a affaire à des données complexes) effectué par le calcul d'attributs, ce afin de rendre les données utilisables par un classifieur.

Ainsi, la méthode choisie doit pouvoir gérer des particularités locales et donc réussir à

1. Un classifieur est un algorithme qui détermine automatiquement la ou les classes d'appartenance d'un élément

scinder le modèle 3D, de façon à isoler les éléments pertinents pour la classification souhaitée.

Ceci est réalisé en proposant une adaptation d'un modèle de classification supervisé de séries temporelles, appelé *time series shapelets*, initialement proposé par [Ye and Keogh, 2009].

Par la suite, il s'agit de vérifier que les extraits censés être les plus pertinents pour la classification sont bien significatifs d'un phénomène et interprétables par l'utilisateur.

Un système permettant de rassembler en un tout cohérent les fonctionnalités présentées ci-dessus n'existe pas. Le défi consiste donc dans un premier temps à chercher à adapter les traitements existants à d'autres types de données aux caractéristiques proches des structures 3D, puis de les compléter par nos propres apports.

Organisation du document

Ce mémoire de thèse est organisé comme suit :

1. la Partie 1 correspond à la méthode de base de classification d'objets 3D présentée, avec :
 - a) dans le chapitre 1, un état de l'art sur la classification supervisée d'objets 3D et les domaines connexes d'inspiration, avec notamment des techniques issues de la classification de séries temporelles,
 - b) dans le chapitre 2, la méthode 3DRESC proposée, méthode qui classe des objets décrits par des maillages ouverts en valorisant des sous-parties extraites ;
 - c) dans le chapitre 3, les expérimentations visant, d'une part, à fixer les paramètres de l'algorithme, et d'autre part à démontrer ses performances ;
 - d) dans le chapitre 4, des expérimentations sur des données réelles de Total sont présentées ;
 - e) dans le chapitre 5, une conclusion partielle ;
2. la Partie 2, quant à elle, correspond à la valorisation, par divers procédés, des sous-parties extraites, ce afin d'améliorer la souplesse de la méthode proposée :
 - a) dans le chapitre 5, une adaptation de la méthode de base à l'apprentissage par transfert, afin de transférer d'un domaine à un autre de la connaissance apprise ;
 - b) dans le chapitre 6, l'introduction d'une méthode novatrice de sélection de caractéristiques pertinentes rendue possible grâce à des outils issus de la théorie des sous-ensembles flous, et appliquée aux sous-parties d'objets 3D, qui permet de combiner les avantages de domaines divers ;

Première partie

**3DRESC : une méthode globale de
classification d'objets 3D**

Préambule

Dans cette partie, il est proposé une nouvelle méthode de classification supervisée d'objets 3D appelée *3DRESC* pour *3D Random Extraction of Sub-parts for Classification*.

Le modèle 3D de base, à partir duquel l'apprentissage est réalisé, est une surface d'un espace à 3 dimensions (cf. Annexe A).

Pour construire ce classifieur, il est nécessaire de convenir :

1. d'une représentation du maillage 3D ;
2. d'une sélection d'attributs adaptée ;
3. d'un algorithme d'apprentissage.

Le processus expérimental permettant l'évaluation de performances est ensuite explicité.

Le point fort de cette méthode est sa capacité, à partir d'une structure complexe dont l'extraction de *features* est loin d'être évidente, à pouvoir décrire cette dernière dans un espace plus adapté à la tâche de classification souhaitée.

Suite à un état de l'art de l'ensemble des domaines relatifs à notre étude, la méthode est présentée.

1. État de l'art

L'état de l'art se rapporte aux notions majeures qui sont les verrous principaux dont la résolution répond à la problématique posée.

Articulé en deux temps :

- tout d'abord, une présentation des techniques existantes pour classifier des objets 3D
- puis, l'introduction des domaines d'inspiration ayant mené à la méthode proposée, en particulier ceux traitant de la sélection de caractéristiques pertinentes.

1.1. Classification d'objets 3D

Les objets 3D, c'est-à-dire l'étude de ces derniers du point de vue de l'apprentissage automatique, ont déjà été abordés dans la littérature. Le point essentiel est la façon d'extraire, à partir d'un maillage d'origine, un vecteur de caractéristiques permettant de le décrire.

Nous proposons ici un tour d'horizon des principales approches existantes : suite à un rappel des généralités en classification supervisée, la classification de structures complexes (comprendre ici non définies par un vecteur de caractéristiques), en particulier celle des objets 3D, est présentée.

1.1.1. Généralités sur la classification supervisée

Dans la mesure où l'on dispose d'un ensemble d'éléments d'apprentissage, c'est-à-dire d'exemples dont la réponse (ou donnée cible) est connue à l'avance, et qu'il faut prévoir sur des éléments dont la réponse est inconnue, on se place tout naturellement dans le cadre de la classification supervisée.

Les approches d'apprentissage peuvent être paramétriques ou non paramétriques :

- Les approches paramétriques se basent sur des distributions statistiques supposées dans les données (autres que l'hypothèse nulle¹). Il est donc indispensable de voir certaines conditions vérifiées pour l'utiliser correctement. Par exemple, la régression linéaire, doit avoir une erreur (fonction d'écart entre la droite estimée et les vraies valeurs) dont les propriétés statistiques, comme la moyenne et la variance, sont constantes. Le nombre de paramètres d'une telle méthode est prédéterminé, et n'évolue donc pas au cours de l'apprentissage (que ce soit en fonction de la nature des données ou de la quantité croissante de données utilisées).
- Les approches non-paramétriques ne dépendent pas de distributions statistiques pré-requises. Il n'y a donc logiquement pas de condition à imposer aux données. L'inconvénient de ce modèle est d'être moins performant dans certains cas précis pour

1. Pour rappel, l'hypothèse nulle correspond à une situation où deux ensembles de données (deux populations) ont les mêmes paramètres statistiques.

lesquels on connaît la tendance statistique globale (si l'on sait par exemple que celle-ci est linéaire), c'est-à-dire dont on dispose de connaissances préalables particulières utilisables en l'état, ce qui n'est pas souvent le cas.

Aujourd'hui, pour ne pas avoir à présupposer un certain nombre d'hypothèses sur les données, et pouvoir utiliser les modèles proposés dans des situations diverses, le choix d'une approche non paramétrique est souvent préféré.

En apprentissage, il existe deux phases principales :

1. la première consiste à créer un modèle à partir des données ;
2. la seconde à utiliser ce modèle pour réaliser des prédictions.

Suite à cela, le modèle est évalué afin de vérifier la conformité au données pour lesquelles il est créé. Il existe de nombreux types d'apprentissage, mais dans ce qui suit l'on se restreint aux évocations des plus en lien avec nos objectifs, à savoir la classification supervisée.

Si l'objectif à atteindre est une valeur binaire ou catégorielle, on a affaire à un problème de classification. Il convient donc de chercher à rattacher un élément à une ou plusieurs classes à laquelle (auxquelles) il appartient. Si l'objectif est un numérique (potentiellement quelconque), alors il s'agit de régression. Dans ce cas, l'objectif est de prévoir une ou plusieurs valeurs réelles.

Au sein des problèmes de classification, la distinction se fait généralement entre classification supervisée, non supervisée et par renforcement.

La classification supervisée dispose dès le départ, d'un ensemble d'exemples contenant les réponses recherchées. En comprenant les règles implicites qui relient les données aux résultats attendus, elle recrée une fonction de passage entre ces données et les résultats qui lui sont attribués. Cette fonction est alors réutilisée lors de la prévision pour de nouvelles données, dont on ne connaît pas dans ce cas particulier la classe d'appartenance.

Pour la classification supervisée, on considère les notations suivantes :

- X un ensemble de m données tel que $\forall i \in \llbracket 1, m \rrbracket$ l'élément X_i est défini par n caractéristiques (ou *features*) tel que $X_i = (X_{i,1}, X_{i,2}, \dots, X_{i,n})$, et donc :

$$X = \begin{pmatrix} X_{1,1} & X_{1,2} & X_{1,3} & \dots & X_{1,n} \\ X_{2,1} & X_{2,2} & X_{2,3} & \dots & X_{2,n} \\ \dots & \dots & \dots & \dots & \dots \\ X_{m,1} & X_{m,2} & X_{m,3} & \dots & X_{m,n} \end{pmatrix} \quad (1.1)$$

- Y le vecteur de classes d'appartenance (un nombre C de classes fixé) des éléments de X , c'est-à-dire que à chaque élément X_i tel que défini plus haut est associé une variable catégorielle, ou classe :

$$Y = \begin{pmatrix} Y_1 \\ Y_2 \\ \dots \\ Y_n \end{pmatrix} \quad (1.2)$$

- F une fonction à calculer, telle que $F(X)$ soit la plus proche possible de Y .

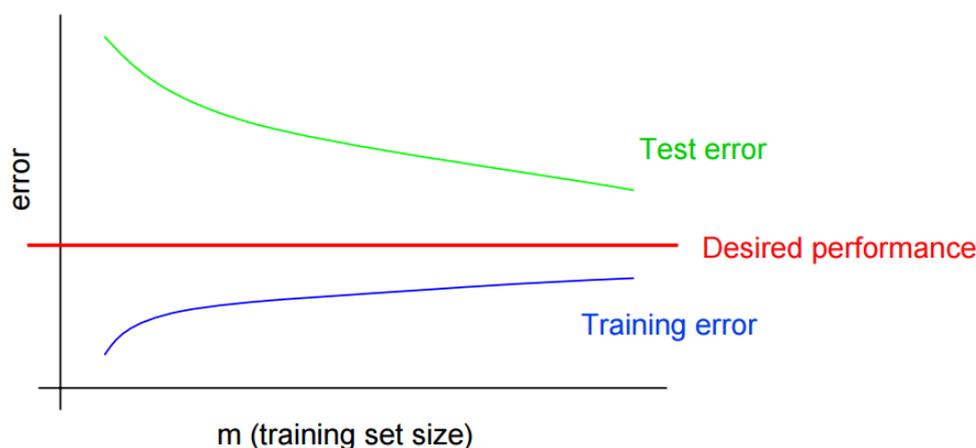


FIGURE 1.1. – Convergence des biais et variance

Le dernier point correspond à la minimisation d'une fonction $G(Y, F(X))$, qui nécessite d'être définie. On peut utiliser, entre les valeurs de Y et de $F(X)$, la distance Euclidienne, celle de Minkowski, ...

La classification supervisée est donc une technique d'apprentissage automatique (ou *Machine learning*) qui vise, à partir d'un ensemble d'exemples $\{X, Y\}$, à trouver la meilleure fonction F de X vers Y possible, fonction qui se devra d'être à la fois performante sur l'ensemble d'apprentissage (dont on connaît la solution) et sur l'ensemble de test (dont la solution est, volontairement ou non, indisponible).

Une algorithme adapté réussit à trouver un bon équilibre entre variance et biais.

La méthode généralement employée pour évaluer la performance d'un système d'apprentissage supervisé est la validation croisée [Ng, 1997]. Le principe est de subdiviser l'ensemble d'apprentissage en z parties, d'utiliser $z - 1$ parties de cet ensemble pour paramétrer le système d'apprentissage (calcul du biais), et la partie restante pour valider l'adaptabilité et la généralisation de cet apprentissage à un jeu de données inconnu (calcul de la variance). On réitère l'expérience z fois avec des combinaisons encore non testées, afin d'obtenir des couples de biais/variances différents.

Finalement, si l'on a :

- une variance moyenne faible ;
- un biais et une variance proches ;
- des écarts-types de biais et de variance faibles ;

il est alors possible d'affirmer que l'apprentissage réalisé est bon et généralise suffisamment sur des données inconnues, à condition que ces dernières respectent la même loi de répartition de probabilité.

Logiquement, on devrait, dans une écrasante majorité des cas d'apprentissage supervisé, obtenir un meilleur résultat de biais que de variance, les valeurs sur lesquelles le biais est calculé ayant été, comme dit précédemment, déjà rencontrées.

La Figure 1.1 résume bien l'objectif à atteindre, si l'on dispose de suffisamment de données.

En classification supervisée, les n éléments de X sont décrits par un ensemble m de caractéristiques (ou attributs) afin de rendre possible leur utilisation dans une fonction (au sens large, non au sens mathématique, l'introduction de l'aléatoire bloquant parfois le déterminisme auquel on pourrait s'attendre). Cette recherche d'attributs est essentielle en apprentissage supervisé, le choix à ce stade de l'étude pouvant avoir des conséquences énormes sur la suite. Un attribut, au sens strict, est défini comme une propriété essentielle et permanente d'une substance : ici la substance n'est autre que l'élément X_i à classifier, que sa classe d'appartenance Y_i soit préalablement connue ou pas.

Chaque élément X_i dispose d'une classe d'appartenance Y_i (connue ou non - on remarquera que l'appartenance à aucune classe revient en soit à appartenir à la classe des non-classés) avec $\forall i \in [1, k], Y_i \in \{0, 1\}$ dans le cas d'une classification binaire.

En classification supervisée, la performance des algorithmes existants et des systèmes informatiques qui les utilisent permet généralement de bien optimiser les données disponibles. Le principal défi consiste donc à extraire, à partir de l'entité que l'on souhaite traiter, un ensemble de caractéristiques pertinentes et adaptée à la tâche de prédiction. En effet, en fonction du type de données que l'on souhaite traiter, les approches peuvent énormément différer. C'est le cas des images, mais aussi des données textuelles, pour lesquelles on ne dispose pas immédiatement de vecteurs d'attributs, et que de simples extractions statistiques ne permettent pas d'étudier en profondeur.

Un autre point que l'on se doit de mentionner ici est celui de l'évaluation des méthodes que l'on propose en termes de performances de prévision. En effet, la question est de savoir quelle métrique choisir, quelle erreur réduire, quelle fonction optimiser. Cette notion d'erreur, que l'on avait précédemment appelée ϵ , détermine l'objectif de la prévision souhaitée. Lorsque l'on cherche à réaliser une classification binaire (0 ou 1 pour l'exemple), quatre indicateurs de base permettent d'en établir la performance :

- Les vrais positifs ou vp - éléments classés comme positifs (+1) et qui le sont réellement) ;
- Les vrais négatifs ou vn - éléments classés comme négatifs et qui le sont réellement ;
- Les faux positifs ou fp - éléments classés comme positifs alors qu'ils sont négatifs ;
- Les faux négatifs ou fn - éléments classés comme négatifs alors qu'ils sont positifs.

Ils sont représentés sous la forme d'un tableau, appelé matrice de confusion (Figure 1.1), pour laquelle les lignes correspondent aux classes réelles et les colonnes aux classes prédites (ou plutôt prévues, si l'on s'en tient à la définition du terme).

		Prévisions	
		Positif	Négatif
Réalité	Positif	nombre de vrais positifs	nombre de faux négatifs
	Négatif	nombre de faux positifs	nombre de vrais négatifs

TABLE 1.1. – Matrice de confusion

En se basant sur ces indicateurs, les principales méthodes pour quantifier les performances, sous la forme de valeurs plus facilement comparables, sont les suivantes :

— La précision :

$$\frac{vp}{vp + fp}$$

— Le rappel :

$$\frac{vp}{vp + fn}$$

— La F-mesure [Rijsbergen, 1979] :

$$\frac{\text{précision} \cdot \text{rappel}}{\text{précision} + \text{rappel}}$$

— La mesure AUC (*Area Under Curve*) [Hilden, 1991], aire sous la courbe ROC (*receiver operating characteristic*). Cette courbe correspond à l'évolution du nombre de vrais positifs par rapport à celle des faux positifs, en faisant varier de 0 à 1 le seuil de discrimination qui détermine l'appartenance d'un élément à l'une des deux classes possibles. Le calcul de l'aire de cette dernière courbe permet d'obtenir une valeur, correspondant directement à la performance du modèle.

La courbe ROC est généralement un bon moyen d'obtenir un résultat suffisamment probant en classification binaire, même en cas de léger déséquilibre des classes, et relativement simple à comprendre (le cas 0.5 correspond à la situation où le classifieur prévoit toujours la même chose, et 1 celle où il ne se trompe jamais). Bien que chaque méthode dispose de ses avantages, cette dernière servira de référence pour le reste de ce travail de thèse.

Les méthodes d'apprentissage automatique permettent donc d'utiliser au mieux des attributs extraits, à condition qu'une relation de causalité existe entre ces derniers et l'objectif de prévision souhaité.

1.1.2. Classification de structures complexes

Afin de pouvoir classifier des données complexes (comme des séries temporelles, des images, des objets 3D, ...) à l'aide des outils habituels de classification supervisée, les méthodes classiques considèrent chaque élément à classifier comme un vecteur de valeurs correspondant aux paramètres de ce dernier. Ce vecteur, dans le cadre de l'apprentissage supervisé classique (dont les SVM [Cortes and Vapnik, 1995] et les forêts aléatoires [Breiman, 2001] sont parmi les méthodes les plus utilisées), possède souvent la même taille pour chaque élément du jeu d'apprentissage. Néanmoins, dans le cas de nombreux formats de données auxquels nous sommes amenés à faire face, on ne détient qu'un échantillon de l'espace discret décrivant lui-même l'entité étudiée, et cet échantillon n'est pas systématiquement de même taille. Par exemple, des séries temporelles peuvent être de tailles différentes mais caractériser le même phénomène.

Pour pouvoir comparer des données si différentes, deux possibilités se présentent :

— comparer directement les structures, que ce soit en échantillonnant par interpolation ou en associant par plus proches voisins pour appariement (ou *matching*, idée de calage de nuages de points) ; il y a donc perte d'information ;

— extraire des descripteurs, puis appliquer une distance adaptée.

L'échantillonnage, potentiellement efficace pour des données temporelles (devant dans ce cas n'être réalisé que dans une seule dimension), peut poser problème si l'on souhaite s'attaquer à des données plus complexes aux tailles trop différentes [Mitra et al., 2004], à l'instar des objets 3D auxquels nous sommes confrontés.

Il est donc proposé d'aborder en premier lieu le calage de nuages de points, et en second lieu les descripteurs d'objets 3D.

1.1.3. Calage de nuages de points pour la reconnaissance de formes

Utilisé principalement dans le cadre de la robotique afin de cerner l'environnement, l'objectif de ce calage est de chercher à faire correspondre le mieux possible deux nuages de points différents, mais qui s'avèrent décrire des objets proches (voire semblables). Il est réalisé sur des ensembles de points de \mathbb{R}^3 , afin de reconnaître des éléments similaires [Cotting et al., 2004].

Concrètement, le principe du calage peut être vu, selon [Mitra et al., 2004], comme la recherche des deux transformations R et T dans l'espace \mathbb{R}^3 (respectivement rotation et translation) entre les nuages de points P et Q minimisant la fonction suivante :

$$\epsilon(R, t) = \sum_{i=1}^{\|P\|} d^2(T((R(p_i)), \Phi_Q) \quad (1.3)$$

où $d^2(T((R(p_i)), \Phi_Q)$ n'est autre que la distance entre le point $p_i \in P$ et Φ_Q (correspondant à l'interpolation du nuage de points pour en faire une surface) du nuage Q . La méthode classique pour résoudre cette équation revient souvent à calculer la distance entre le point transformé $Rp_i + t$ et son plus proche voisin p_i du nuage Q . Pourtant, cette fonction n'est pas convexe, et les méthodes *Iterative Closest Point* de [Besl and McKay, 1992] et *Random Sample Consensus* (ou *RANSAC*) de [Fischler and Bolles, 1981] par descente de gradient, ne permettent de trouver qu'une solution locale (donc approchée).

De plus, la quantité de possibilités, exponentiellement croissante en fonction du nombre de points (notamment par rotations, translations, et homothétie), rend ces méthodes peu applicables à la détection de phénomènes locaux.

Enfin, cet ensemble de procédés considère les nuages de points de façon non supervisée, et ne peut donc pas, dans le cadre d'une classification supervisée particulière, évaluer lui-même l'importance et l'apport de chaque caractéristique.

1.1.4. Descripteurs d'objets 3D

Dans le cas de séries temporelles ou d'images, il est possible de comparer directement (avec la distance euclidienne, par exemple) les données des sous-parties extraites, celles-ci étant dans des espaces mesurables. Dans le cas d'objets 3D, comparer directement deux entités de ce type, comme nous l'avons vu dans le 2.1.2, n'est plus envisageable. En effet, en plus de devoir établir un alignement selon la translation, la rotation, l'échelle et la réflexion,

il est indispensable de prendre en compte le fait que l'échantillonnage n'est pas réalisé de la même façon entre deux extraits. Le calcul de la distance minimale entre chaque point du premier extrait avec la structure du second n'est possible, et difficilement, qu'en présence d'éléments de mêmes étendues.

L'étude d'objets 3D décrits par des maillages non réguliers ouverts nécessite à cet effet d'extraire d'abord des descripteurs qui ont pour but de convertir les données en une forme exploitable pour l'apprentissage et la comparaison. On met dès lors de côté tout ce qui se rapporte à des descripteurs de volumes ou de squelettes d'objets, les surfaces non fermées ne s'y prêtant pas [Alexandre, 2012]. En effet, on cherche par ces méthodes à trouver un ensemble de segments reliés qui correspondent au barycentre coulissant de l'objet, notion qui n'a plus de sens si l'objet n'est pas fermé, le squelette pouvant être totalement extérieur à la surface.

Il existe à l'heure actuelle de nombreuses méthodes visant à extraire des descripteurs d'objets 3D, notamment [Dang, 2014]. Outre celles de passage de la 3D à la 2D par coupe ou projection, qui ne s'adaptent pas à notre problème au regard de la trop grande perte de données qui en découle, la distinction se fait généralement entre les descripteurs locaux et globaux.

Dans notre cadre, les descripteurs doivent respecter les conditions suivantes :

- une bonne capacité à représenter un prototype de la classe concernée (donc du phénomène local détecté) ;
- une robustesse par rapport aux principales transformations géométriques (rotation, translation, passage à l'échelle, ...);
- une caractérisation d'une zone particulière au sein de l'objet 3D, qui permet une justification de la classification finale.

Les descripteurs globaux

Descripteurs d'objet 3D par excellence, ils représentent ces derniers dans leur globalité [Alexandre, 2012], c'est-à-dire que ce sont des descripteurs décrivant un objet 3D dans son ensemble [Tabia, 2011].

Ainsi, il s'agit de récupérer la tendance globale de l'objet. Les données sont lissées, ce qui entraîne la perte des particularités locales.

Les structures 3D que nous avons à traiter n'étant pas fermées, des techniques telles que les harmoniques sphériques ou les séries de Fourier 3D de [Vranic and Saupe, 2001] deviennent elles aussi inopérantes.

Les descripteurs globaux se regroupent en deux principaux ensembles :

1. les méthodes cherchant à caler l'objet sur un modèle de représentation global (exemple des séries de Fourier abordé ci-dessus) ;
2. les méthodes visant à calculer, en chaque point, un descripteur local dépendant de son entourage, et qui servira de base à un histogramme de répartition de valeurs. Le descripteur global est cet (ou cet ensemble d') histogramme(s) obtenu(s).

Bilan :

Le point 2 nous permet d'affirmer que les mêmes outils peuvent ainsi être utilisés dans des études globales et locales d'un objet 3D, en les rassemblant sous la forme d'un histogramme de répartition des valeurs obtenues. Ces descripteurs, potentiellement utiles pour décrire localement et globalement les objets, sont introduits dans la partie suivante.

Quant aux descripteurs exclusivement globaux du point 1, ces derniers ne sont pas adaptés à notre étude, car en utilisation normale (c'est-à-dire sur l'ensemble de la structure), ils perdent les caractéristiques purement locales, c'est-à-dire issues de sous-parties. En effet, on ne s'intéresse qu'au tout, et l'on omet logiquement certains détails présents à petite échelle dans de sous-parties de ces objets 3D. A force de chercher un prototype suffisamment généralisable, des détails qui peuvent s'avérer significatifs sont "oubliés". Nous nous servons néanmoins de certains d'entre eux afin de pouvoir comparer les taux de classification.

Les descripteurs locaux

L'idée est d'extraire des descripteurs à partir d'histogrammes de l'ensemble des points, ou de points saillants que l'on suppose discriminants. A titre d'exemple, il est possible d'approximer les courbures (spectre de forme), ou bien de chercher à définir la tendance des points du voisinage.

Pour décrire un objet avec des descripteurs locaux, deux possibilités sont envisageables [Shaiek, 2013] :

- les zones sont choisies aléatoirement ou en fonction d'une courbure ou cassure importante et/ou remarquable, ce qui peut poser problème pour la pertinence des descripteurs utilisés ultérieurement en classification, car indépendants de cette dernière ;
- toutes les zones sont choisies, ce qui fait perdre l'information de position de la sous-zone pertinente, et empêche ainsi la compréhension de la classification par l'utilisateur.

Bilan :

Malgré le choix déterminant à effectuer, les descripteurs d'objets locaux peuvent être de bons moyens de comparaison entre différents extraits récupérés. Néanmoins, le facteur d'échelle ne sera pas pris en compte, en particulier si l'on utilise un histogramme de répartition de valeurs.

Choix descripteurs globaux/locaux

On remarque que la frontière entre descripteurs locaux et globaux est extrêmement mince. En effet, il est tout à fait possible pour un descripteur global de décrire une forme locale, si ce dernier est appliqué à une sous-partie de l'élément uniquement. Cependant, à l'inverse des descripteurs globaux, les descripteurs locaux sont généralement indépendants de l'orientation, ce qui leur confère un avantage indéniable.

Par la suite, quelques descripteurs choisis sont présentés, ces derniers ayant comme avantages d'être à la fois robustes selon les principales transformations géométriques et suffisamment précis pour conserver les caractéristiques locales que l'on souhaite pouvoir utiliser.

Quelques descripteurs locaux

Le Unique Shape Context (USC)

Ce descripteur local, proposé par [Tombari et al., 2010] comme une amélioration du *3D Shape Context* de [Frome et al., 2004], ne considère, pour chaque point p de l'espace, que son voisinage de R (cercle de centre p et de rayon R) grâce à la formule :

$$M = \frac{1}{Z} \sum_{i:d_i < R} (R - d_i)(p_i - p)(p_i - p)^T \quad (1.4)$$

où p_i est le $i^{\text{ème}}$ point du voisinage de p , $d_i = \|p_i - p\|_2$ et $Z = \sum_{i:d_i < R} (R - d_i)$.

La décomposition en vecteurs propres de M permet ensuite d'en extraire un ensemble de repères (choisis aléatoirement), dont le plus petit vecteur (le 1^{er}) est normal à la surface. Ces repères, ordonnés selon les normes croissantes des vecteurs, respecteront l'orientation d'une base orthogonale directe.

L'idée de [Tombari et al., 2010] consiste ensuite à extraire aléatoirement un certain nombre de descripteurs (chaque descripteur correspondant au voisinage d'un point) parmi ceux calculés. Elle permet d'approximer avec une précision relativement bonne la structure de l'objet, donc de pouvoir effectuer de l'appariement, de la comparaison et de la classification. Pour définir l'objet, il s'agit là d'un moyen plus rapide que le calcul d'histogramme, mais qui perd en précision car tributaire de l'aléatoire et du nombre de points choisis.

Bilan :

On remarque néanmoins que, dans le cas présent, ce descripteur ne nécessite pas que les points soient reliés : il peut aussi s'appliquer aux nuages de points, mais perd une partie de l'information représentée par le maillage disponible dans l'étude.

Le Point Feature Histogram (PFH)

Le but du Point Feature Histogram, proposé par [Rusu et al., 2008], est de chercher à généraliser à la fois les normales aux surfaces et l'estimation de la courbure.

La relation entre les orientations est obtenue de la façon suivante : Soit p et q 2 points de l'objet, et (u, v, w) une référence construite comme suit :

- u est le vecteur normal à la surface en p
- $v = u \wedge \frac{p-q}{d}$
- $w = u \wedge v$

avec $d = \|p - q\|_2$.

Alors la différence entre les normales en p et q (nommées respectivement n_p et n_q) vaut :

- $\alpha = \arccos(v.n_q)$
- $\phi = \arccos(u.\frac{p-q}{d})$
- $\theta = \arctan(w.n_q/u.n_q)$

Une fois les valeurs des couples calculées, ces dernières servent à la construction de l'histogramme de répartition, descripteur des formes présentes sur l'objet.

Bilan :

Les caractéristiques de proximité entre les différents points sont utilisées grâce aux normales aux surfaces élémentaires, et donc les liaisons entre points sont cette fois-ci bien mises à profit.

Le Heat Kernel Signature

Le *Heat Kernel Signature* (HKS) est un descripteur local d'objet 3D utilisé pour la première fois par [Sun et al., 2009].

Ce dernier s'appuie sur la formule de diffusion de chaleur, elle-même calculée à l'aide de l'opérateur Laplace-Beltrami. Cet opérateur est une extension du Laplacien aux variétés Riemanniennes, qui sont des variétés différentielles disposant d'une métrique (dite Riemannienne) permettant de calculer la distance entre 2 points de la variété [Lee, 1991]. L'exemple le plus simple pour en illustrer l'utilité est celui de la distance entre 2 points sur une sphère, qui ne correspond pas à la distance euclidienne les séparant.

On assimile donc la surface étudiée à une variété Riemannienne M sur laquelle la diffusion de chaleur est calculée.

L'idée est que, lors du calcul de la répartition de chaleur, cette dernière dépend de la surface sur laquelle elle évolue, et permet de cette façon de caractériser cette même surface. On utilise donc la chaleur comme un moyen indirect d'évaluation de la surface.

Bilan :

Le HKS est par ailleurs adapté pour les réseaux de neurones profonds (deep learning) par [Fang et al., 2015]. Au delà de son efficacité pour ces cas concrets, il est indépendant de la distribution du maillage et robuste aux changements géométriques (échelle, rotation, etc...). Il s'avère être pour nos objectifs de prévision, le meilleur descripteur possible en termes de performances.

1.1.5. Choix du Heat Kernel Signature

Les expérimentations réalisées permettent de choisir le HKS comme descripteur d'objets 3D. Pour cette raison, nous en proposons une étude plus approfondie.

Notations :

Soit (cf la Figure 1.2) :

- un ensemble de points $P = \{p_1, p_2, p_3, \dots, p_k\}$ de \mathbb{R}^3 notés $\forall i \in \llbracket 1, k \rrbracket p_i = [x_i, y_i, z_i]$
- un ensemble d'arrêtes $ij \in V$ tels que $i, j \in \llbracket 1, k \rrbracket$ et ij relie les points p_i et p_j (pour $i \neq j$)
- un ensemble de triangles T composés de trois points $p_i, p_j, p_k \in \llbracket 1, n \rrbracket$ que l'on note (ijk)

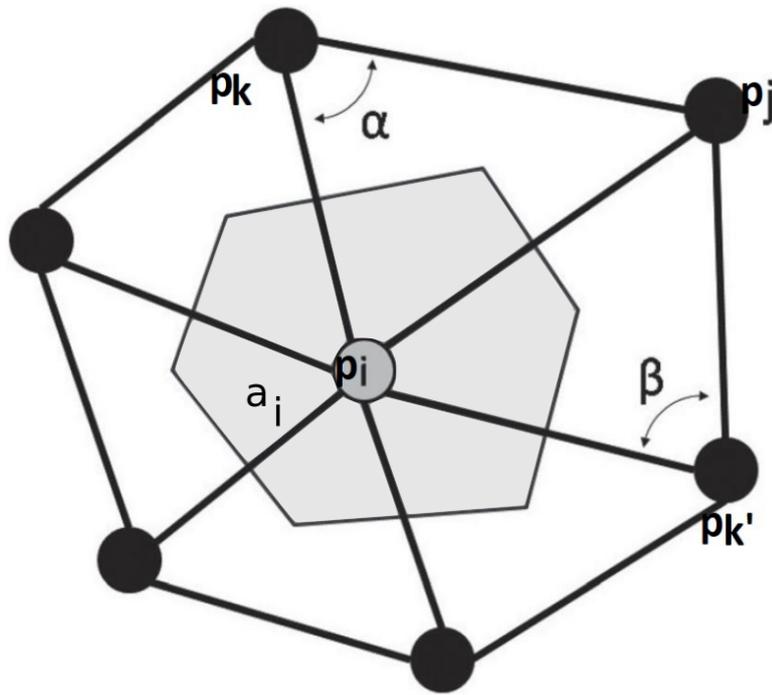


FIGURE 1.2. – Notations utiles pour le HKS

- A_{ijk} l'aire du triangle (ijk) .
- pour une arête ij , on définit la valeur

$$\delta_{ij} = \frac{\cot \alpha_{ij} + \cot \beta_{ij}}{2}$$

, avec α_{ij} et β_{ij} les angles $\widehat{ik'j}$ et \widehat{ikj} des 2 triangles (ijk') et (ijk) contenant l'arête $\langle i, j \rangle$

Théorie pour un maillage 3D rectangulaire :

On construit la matrice W de taille $k \times k$ telle que, $\forall (i, j) \in P^2$:

$$W_{ij} = \begin{cases} \delta_{ij} & \text{si } ij \in V \\ - \sum_{k \neq i} W_{ik} & \text{si } i = j \\ 0 & \text{sinon.} \end{cases} \quad (1.5)$$

État de l'art

Et la matrice diagonale :

$$A = \text{diag}(a_1, a_2, \dots, a_i, \dots) = \begin{pmatrix} a_1 & 0 & 0 & \dots & 0 \\ 0 & a_2 & 0 & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & \dots & a_k \end{pmatrix} \quad (1.6)$$

avec :

$$a_i = \frac{1}{3} \sum_{\{j,k \mid (i,j,k) \in T\}} A_{ijk}$$

On réalise le calcul terme à terme suivant :

$$L = A^{-1}W \quad (1.7)$$

A partir de cette matrice L , on calcule les valeurs et vecteurs propres de L . Soient Λ les valeurs propres ordonnées, et Φ les vecteurs propres classés selon l'ordonnement de leurs valeurs propres respectives.

On définit la valeur K_t défini par, $\forall t \in \mathbb{N}$ (t étant le moment permettant de calculer la répartition de chaleur, une plus grande valeur de t permet de mieux appréhender son évolution et donc de mieux définir la surface) :

$$K_t = \Phi' e^{-\Lambda t} \Phi^T$$

Le descripteur se définit finalement par, $\forall i \in P$:

$$HKS(i) = [K_{t_1}(i, i), K_{t_2}(i, i), K_{t_3}(i, i), \dots, K_{t_{max}}(i, i)]$$

avec $(t_1, t_2, \dots, t_{max}) \in \mathbb{R}^{max}$, les instants d'observation de la distribution de chaleur sur la surface étudiée.

Ce qui donne :

$$HKS = \begin{pmatrix} K_{t_1}(1, 1) & K_{t_1}(2, 2) & \dots & K_{t_1}(k, k) \\ K_{t_2}(1, 1) & K_{t_2}(2, 2) & \dots & K_{t_2}(k, k) \\ \dots & \dots & \dots & \dots \\ K_{t_{max}}(1, 1) & K_{t_{max}}(2, 2) & \dots & K_{t_{max}}(k, k) \end{pmatrix}$$

Néanmoins, utiliser l'ensemble des k valeurs et vecteurs propres de L n'est pas nécessaire, il convient donc de se restreindre à une valeur limite $g < k$:

$$HKS = \begin{pmatrix} K_{t_1}(1, 1) & K_{t_1}(2, 2) & \dots & K_{t_1}(g, g) \\ K_{t_2}(1, 1) & K_{t_2}(2, 2) & \dots & K_{t_2}(g, g) \\ \dots & \dots & \dots & \dots \\ K_{t_{max}}(1, 1) & K_{t_{max}}(2, 2) & \dots & K_{t_{max}}(g, g) \end{pmatrix}$$

Il y a donc 2 paramètres à fixer dans cette formule :

- t_{max} peut être interprété comme étant le niveau de précision que l'on souhaite obtenir de l'opérateur de Laplace-Beltrami, base de la représentation de la surface 3D. En effet, celui-ci est utilisé dans son approximation aux maillages 3D, approximation dont la valeur exacte ne saurait être trouvée qu'en calculant le descripteur jusqu'à $t = \text{inf}$.
- g est le nombre de valeurs propres et vecteurs propres calculés à partir de l'opérateur de Laplace-Beltrami, par lequel on souhaite décrire la surface 3D.

Dans un cas comme le nôtre, s'arrêter aux premières valeurs de t (voire la première valeur de t) est largement suffisant [Sun and Ponce, 2013], l'opérateur exponentiel dans la formule de K_t ayant tendance à tendre rapidement vers 0 si k croît ($K_t \gg K_{t+1}$).

Empiriquement, il n'est pas nécessaire d'excéder les 10 valeurs si l'on souhaite discriminer suffisamment la surface étudiée, cela dépend bien sûr de sa complexité (dans notre cas le maximum des valeurs de i vaut n).

On peut noter que si la précision utilisée pour calculer le Laplace-Beltrami est faible, il n'est pas utile de calculer beaucoup de vecteurs propres.

Exemple d'utilisation du HKS avec 3 points

Supposons que l'on dispose d'une surface 3D représentée par un maillage triangulaire dont les points sont $P = \{p_1, p_2, p_3\}$.

La matrice W dite de poids est définie par :

$$\begin{pmatrix} -\sum_{j=2,3} \delta_{1j} & \delta_{12} & \delta_{13} \\ \delta_{21} & -\sum_{j=1,3} \delta_{2j} & \delta_{23} \\ \delta_{31} & \delta_{32} & -\sum_{j=1,2} \delta_{3j} \end{pmatrix}$$

avec, pour rappel, $\delta_{ij} = \frac{\cot \alpha_{ij} + \cot \beta_{ij}}{2}$ et α_{ij} et β_{ij} les angles des triangles contenant l'arrête ij .

La matrice A dite d'aires, vaut elle :

$$\begin{pmatrix} \frac{1}{3} \sum_{\{j,k \mid (1,j,k) \in T\}} A_{1jk} & 0 & 0 \\ 0 & \frac{1}{3} \sum_{\{j,k \mid (2,j,k) \in T\}} A_{2jk} & 0 \\ 0 & 0 & \frac{1}{3} \sum_{\{j,k \mid (3,j,k) \in T\}} A_{3jk} \end{pmatrix}$$

Or la matrice L de l'opérateur de Laplace-Beltrami vaut :

$$L = A^{-1}W$$

A étant une matrice diagonale dont aucun des éléments n'est nul, on a donc :

$$L = \begin{pmatrix} \frac{1}{\frac{1}{3} \sum_{\{j,k \mid (1,j,k) \in T\}} A_{1jk}} & 0 & 0 \\ 0 & \frac{1}{\frac{1}{3} \sum_{\{j,k \mid (2,j,k) \in T\}} A_{2jk}} & 0 \\ 0 & 0 & \frac{1}{\frac{1}{3} \sum_{\{j,k \mid (3,j,k) \in T\}} A_{3jk}} \end{pmatrix} * \begin{pmatrix} -\sum_{j=2,3} \delta_{1j} & \delta_{12} & \delta_{13} \\ \delta_{21} & -\sum_{j=1,3} \delta_{2j} & \delta_{23} \\ \delta_{31} & \delta_{32} & -\sum_{j=1,2} \delta_{3j} \end{pmatrix}$$

Ce qui donne :

$$L = \begin{pmatrix} \frac{W_{11}}{A_{11}} & 0 & 0 \\ 0 & \frac{W_{22}}{A_{22}} & 0 \\ 0 & 0 & \frac{W_{33}}{A_{33}} \end{pmatrix}$$

$$= \begin{pmatrix} \frac{-\sum_{j=2,3} \delta_{1j}}{\frac{1}{3} \sum_{\{j,k \mid (1,j,k) \in T\}} A_{1jk}} & 0 & 0 \\ 0 & \frac{-\sum_{j=1,3} \delta_{2j}}{\frac{1}{3} \sum_{\{j,k \mid (2,j,k) \in T\}} A_{2jk}} & 0 \\ 0 & 0 & \frac{-\sum_{j=1,2} \delta_{3j}}{\frac{1}{3} \sum_{\{j,k \mid (3,j,k) \in T\}} A_{3jk}} \end{pmatrix}$$

L est donc une matrice de description d'un maillage triangulaire 3D, qui tient à la fois compte des angles et des surfaces touchant chaque sommet du maillage.

Suite à cela, on calcule les valeurs propres de L , que l'on ordonne et que l'on rassemble dans la matrice diagonale Λ , ainsi que les vecteurs propres correspondant aux valeurs propres (dans le même ordre, donc), au sein de la matrice Φ .

A partir de ces deux matrices de valeurs et vecteurs propres, pour chaque temps de diffusion t_i ($i \in \llbracket 1, max \rrbracket$), et comme abordé précédemment, on a :

$$K_t = \Phi' e^{-\Lambda t} \Phi^T$$

$$HKS(i) = [K_{t_1}(i, i), K_{t_2}(i, i), K_{t_3}(i, i), \dots, K_{t_{max}}(i, i)]$$

1.2. Méthodes pour extraire, sélectionner et valoriser des caractéristiques pertinentes

Ces autres domaines visent à cerner l'ensemble des sujets nécessaires à l'adaptation aux objets 3D d'une méthode de classification supervisée de séries temporelles : les time series shapelets.

Ainsi sont présentés :

1. dans un premier temps, la sélection d'attributs pertinents en classification supervisée,
2. puis certaines méthodes de classification de séries temporelles et d'images,
3. et enfin l'étude de l'aléatoire en apprentissage.

1.2.1. Sélection d'attributs

En présence d'un grand nombre d'attributs potentiels dont certains peuvent avoir un effet dommageable sur la classification en cours, il est nécessaire de ne sélectionner que les plus pertinents.

[Chandrashekar and Sahin, 2014] étudie le domaine de la sélection d'attributs (*feature selection*) basée sur le choix des sous-parties les plus discriminantes.

Les deux principales approches sur lesquelles nous nous focalisons ici sont :

1.2. Méthodes pour extraire, sélectionner et valoriser des caractéristiques pertinentes

1. les méthodes de filtrage qui, par maximisation d'entropie (gain d'information) de chaque attribut (évaluation individuellement), ne sélectionnent que les plus discriminants ;
2. les méthodes d'ensemble (wrapper methods) elles considèrent quant à elles des ensembles d'attributs candidats.

Les méthodes de filtrage

Dans le cas de sélection de sous-parties pertinentes, la méthode classique consiste à calculer la distance (minimale) entre chaque attribut candidat et chaque objet comme [Ye and Keogh, 2009]. Puis, pour chaque candidat, on ordonne les objets selon la distance les séparant de ce dernier. Enfin, on évalue la capacité de discrimination du candidat. Cette évaluation est réalisée par le gain d'information² (Mutual Information) qui vise à mesurer la dépendance entre deux variables.

Ce gain d'information est ensuite utilisé en testant chaque point de séparation possible entre 2 sous-parties, et le point maximisant le gain est retenu (Figure 1.3). Dans le cas idéal, tous les éléments d'une certaine classe sont d'un même côté de la séparation et sont les seuls à y être.

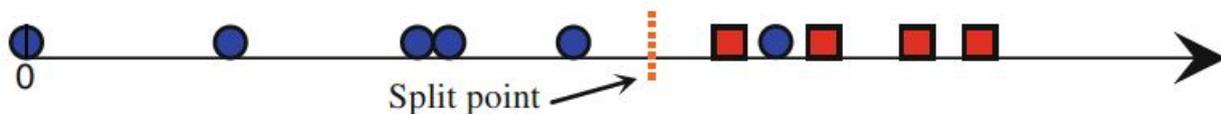


FIGURE 1.3. – Détermination des shapelets les plus déterminantes : capacité à scinder l'ensemble des objets en deux ensembles homogènes - [Ye and Keogh, 2009]

La méthode dite de F-statistic (ou f-stat, ou test de Fisher) est une autre méthode utilisée pour sélectionner des caractéristiques pertinentes. Issue des travaux originaux de [Kruskal, 1952], [Lines et al., 2012] en propose l'utilisation, et elle s'avère selon lui plus performante et plus précise pour évaluer le caractère discriminant d'une sous-partie que la gain d'information. Si l'on ajoute à cela un temps de calcul a priori plus faible, cette formule peut être plus

2. Soit Y une variable aléatoire discrète, et $p(y)$ sa probabilité (qui correspond à la proportion d'éléments de la classe considérée), l'entropie de Shannon [Shannon and Weaver, 1963] de Y est donnée par :

$$H(Y) = - \sum_{y \in Y} p(y) \log(p(y)) \quad (1.8)$$

L'entropie conditionnelle de Y sachant X est définie par :

$$H(Y|X) = - \sum_{x \in X} \sum_{y \in Y} p(x, y) \log(p(y|x)) \quad (1.9)$$

Le gain d'information [Quinlan, 1986] (Mutual Information ou MI) de la variable Y vaut finalement :

$$I(Y, X) = H(Y) - H(Y|X) \quad (1.10)$$

appropriée dans certains cas, bien que nos expériences n'aient pu le déduire :

$$f\text{-stat} = \frac{\frac{1}{C-1} \sum_{i=1}^C (\bar{D}_i - \bar{D})^2}{\frac{1}{n-C} \sum_{i=1}^C \sum_{d_j \in D_i} (d_j - \bar{D}_i)^2} \quad (1.11)$$

avec $C > 1$ le nombre de classes, n le nombre de sous-parties ($n > C$), \bar{D}_i la moyenne des degrés entre les surfaces de classe i et la sous-partie concernée, et \bar{D} la moyenne des degrés entre la sous-partie et les objets 3D.

Les méthodes d'ensemble

Le second type de méthodes implique de passer en revue toutes les possibilités d'ensembles d'attributs possibles.

Il s'agit d'évaluer l'information mutuelle entre n variables pour une classe C particulière, soit :

$$I(\{x_1, x_2, x_3, \dots, x_n\}; C) = \sum_{k=1}^m \sum_{\forall S \subseteq \{x_1, x_2, x_3, \dots, x_n\}; |S|=k} I([S \cup C]) \quad (1.12)$$

avec :

$$I([S \cup C]) = I(\{x_1, x_2, x_3, \dots, x_k\}; C)$$

$$I(x; y|z) = \sum_{i=1}^n p(z(i)) I(x; y|z = z(i))$$

$$I(x; y; z) = I(\{x, y\}; z) - I(x; z) - I(y; z) = I(y; z|x) - I(y; z)$$

On remarque que devoir calculer ces valeurs pour un grand nombre d'attributs candidats revient à calculer $\binom{n}{k}$ combinaisons avec k variant selon le nombre d'attributs souhaité à la fin, ce qui devient très coûteux en temps de calcul dès que l'on souhaite en tester de grandes quantités.

Bilan

C'est la raison pour laquelle des approches visant à réduire le temps de calcul des méthodes d'ensemble ont été imaginées : les méthodes intégrées (embedded methods). Leur objectif général est de chercher des attributs ayant un fort caractère discriminant pour la classification visée, tout en étant peu corrélés avec les attributs déjà choisis.

Bien que performantes, elles souffrent du même inconvénient que les méthodes d'ensemble dans notre cas d'application, à savoir un temps de calcul trop important. Ainsi, le choix des attributs les plus discriminants selon des critères individuels, c'est-à-dire selon

des méthodes par filtrage, est opportun.

1.2.2. Distance et similarité entre histogrammes de répartition de valeurs

L'utilisation d'un descripteur local requiert, dans le cadre d'une comparaison entre sous-surfaces extraites, une notion de distance permettant d'en évaluer la similarité.

Nous récupérons, lors du calcul du descripteur, un ensemble de données rassemblées dans un histogramme correspondant aux valeurs des descripteurs pour chaque point de la surface retenue. Plusieurs types de distances sont disponibles, afin de comparer ce qui se rapporte alors à des distributions de probabilités.

Quelques distances

Deux d'entre elles s'avèrent particulièrement adaptées à notre problème de comparaison de distributions de probabilités discrètes (donc à des comparaisons d'histogrammes), l'expérience ayant permis de les sélectionner, et sont par la suite testées dans le cadre de l'algorithme proposé :

- la mesure de Bhattacharyya et ses variantes ;
- la divergence de Kullback-Leibler.

Pour la suite, on pose $P = P_1, P_2, \dots, P_N$ et $P' = P'_1, P'_2, \dots, P'_N$ les répartitions de probabilités de dimension N que l'on compare (N étant donc le nombre d'intervalles de ces histogrammes).

La mesure de Bhattacharyya et ses variantes

En statistiques, la mesure de Bhattacharyya [Bhattacharyya, 1943] est une mesure de la similarité de deux distributions de probabilités discrètes. Elle est très utilisée en classification, et en particulier en vision par ordinateur (*Computer Vision*) [Leung and Malik, 2001] [Dubuisson, 2010] qui utilise entre autres les travaux de [Aherne et al., 1998].

Cette méthode se base sur la formule suivante :

$$Bhattacharyya(P, P') = \sum_{i=1}^N \sqrt{P_i P'_i} \quad (1.13)$$

[Comaniciu et al., 2003] propose pour le suivi d'objets une modification de cette mesure afin d'en faire une distance :

$$d(P, P') = \sqrt{1 - Bhattacharyya(P, P')} \quad (1.14)$$

La divergence de Kullback-Leibler

Davantage utilisée en cryptage, cette mesure proposée par [Kullback and Leibler, 1951] représente la différence moyenne du nombre de bits nécessaires au codage entre une vérité P et une répartition Q , basée elle sur un modèle théorique.

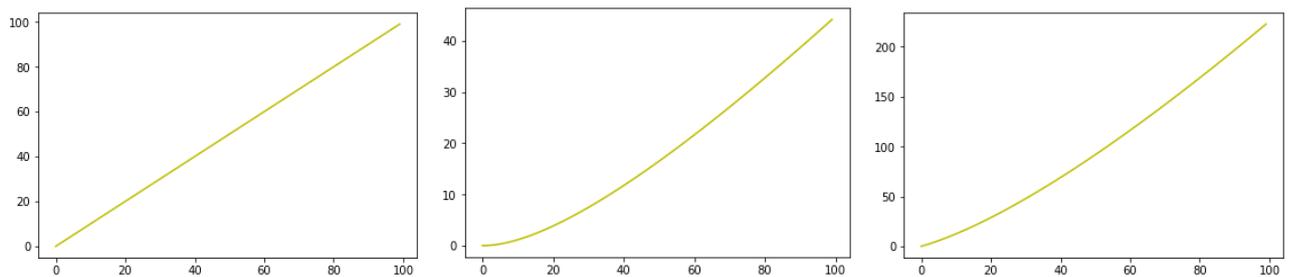


FIGURE 1.4. – Distances de Minkowski, Matusita et divergence de Kullback-Leibler en fonction de la distance Euclidienne

Dans le cas de probabilités discrètes, la divergence de Kullback-Leibler de Q par rapport à P donne :

$$D_{KL}(P|P') = \sum_i P_i \log \frac{P_i}{P'_i} \quad (1.15)$$

Cette mesure ne respecte pas l'inégalité triangulaire et n'est pas symétrique, donc ne correspond pas exactement à notre besoin.

La distance de Matusita

La distance Matusita de [Matusita, 1955], approche différente de la distance Euclidienne (et de ses variantes que sont les distances de Minkowski) est définie par :

$$Matusita(p, p') = \sum_{i=1}^N \left(\sqrt{p(i)} - \sqrt{p'(i)} \right)^2 \quad (1.16)$$

A priori, un lissage est effectué par la racine carrée, ce qui permet une plus grande tolérance aux légers changements des éléments P et P' .

Visualisation de l'évolution des distances

On propose une visualisation des distances considérées entre deux points de coordonnées, initialement quelconques, dont on augmente linéairement la distance Euclidienne les séparant.

La Figure 1.4 met en lumière le fait que la distance de Matusita semble plus tolérante aux faibles variations que les deux autres.

1.2.3. Algorithme des shapelets

Proposée par [Refregier, 2001], la notion de shapelets était initialement appliquée aux images. La méthode est alors basée sur une décomposition linéaire de chaque image en une série de fonctions élémentaires décrivant des sous-parties forcément locales ; ces fonctions sont appelées shapelets. Concrètement, cela permet, à partir de données complexes, d'obtenir une version simplifiée de ces données en les décomposant en entités plus simples à étudier.

1.2. Méthodes pour extraire, sélectionner et valoriser des caractéristiques pertinentes

TABLE 1.2. – Notations

Symbole	Signification
T	Série temporelle
S	Sous-partie ou sous-séquence d'une série temporelle
T_{taille}	Taille de la série temporelle
S_{taille}	Taille de la sous-partie
D	Ensemble des séries temporelles du jeu d'apprentissage
P	Ensemble des sous-séries extraites du jeu d'apprentissage

De la même façon que pour les images, la dépendance entre les éléments de l'espace discret dans lequel est définie une série temporelle est une particularité qu'un simple vecteur ne peut que difficilement prendre en compte. Il convient donc de créer des techniques pour pouvoir traiter ce cas particulier de données.

Si l'on considère que l'échantillonnage est régulier, il est possible d'effectuer un parallèle avec les images, qui sont alors comme des séries numériques en non plus une mais deux dimensions. De nombreuses méthodes permettant de les traiter existent, mais l'idée générale la plus intéressante dans notre cas est celle vue plus haut développée par [Ye and Keogh, 2009] (et [Lozano Vega, 2015] pour les images) qui cherche à déterminer les sous-parties discriminantes pour le type de données étudié, ces sous-parties n'étant plus forcément des versions élémentaires du type de données traité, mais des extraits pouvant être de tendances quelconques.

Cette méthode de classification supervisée a par la suite été adaptée aux séries temporelles pour la classification supervisée par [Ye and Keogh, 2009], et dont le taux de bonne classification et la vitesse de calcul ont été améliorés, entre autres avec les travaux de [Grabocka et al., 2014] et [Hills et al., 2014].

Time series shapelets

Pour la suite, et lorsque l'on parle de séries temporelles, on admet les notations de la Table 1.2. Une série temporelle T de longueur T_{taille} est une suite ordonnée finie de valeurs réelles, soit :

$$T = t_1, t_2, \dots, t_{T_{taille}}$$

Ces données sont généralement indexées selon un critère temporel.

Une sous-série S de la série temporelle T de taille T_{taille} est un extrait de taille $S_{taille} \leq T_{taille}$ d'éléments contigus de T , telle que :

$$S = t_p, \dots, t_{p+S_{taille}-1}$$

avec $1 \leq p \leq T_{taille} - S_{taille} + 1$.

L'algorithme extrait toutes les sous-séries possibles de chaque série temporelle du jeu d'apprentissage dans l'ensemble P . Suite à cela, ces sous-séries rassemblées dans P sont évaluées individuellement afin de n'en conserver que les plus discriminantes pour la classification. La

méthode employée pour une telle tâche est le calcul du gain d'information (*mutual information* en anglais) qui est une quantité mesurant la dépendance statistique de deux variables. Chaque sous-série se voit finalement attribuée d'un score de pertinence, qui dépend à la fois des distances entre chaque objet et elle-même, et des classes d'appartenance de ces objets. Les Q meilleures sous-séries sont sélectionnées et deviennent chacun une partie de l'arbre de classification construit. Par la suite, la réutilisation de cet arbre de décision, avec les multiples optimisations qui sont proposées, permettent le calcul de la classe d'appartenance de chaque série.

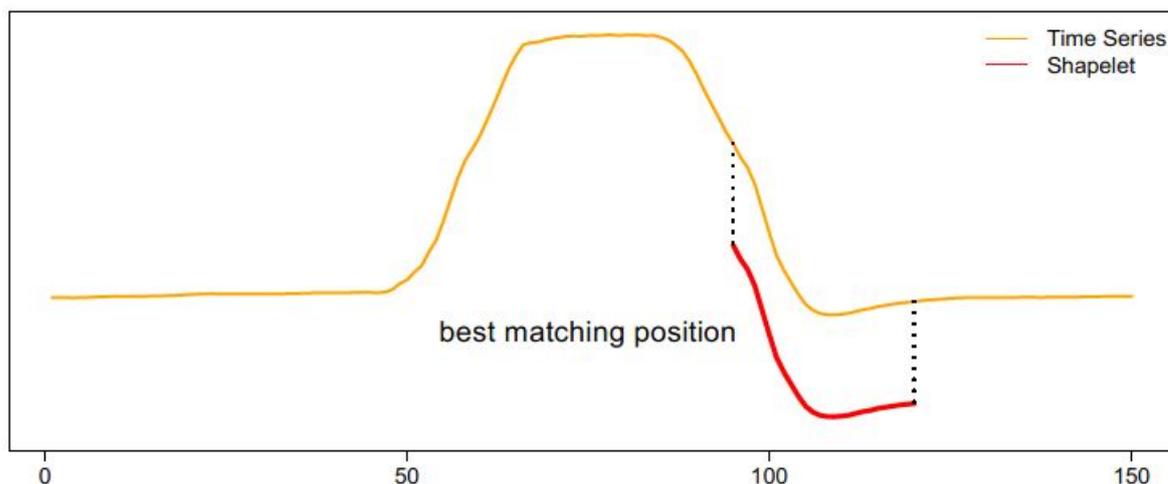


FIGURE 1.5. – Shapelets de séries temporelles - [Ye and Keogh, 2009]

Pour simplifier, dans le cas des séries temporelles, on extrait l'ensemble des sous-parties (sous-séries) possibles de chaque élément (série) de l'ensemble d'apprentissage, et on détermine lesquelles scindent le mieux l'ensemble des éléments, dans le but de maximiser un certain indicateur permettant de juger de leurs capacités de discrimination (Figure 6.2).

La méthode classique de maximisation d'entropie (vue dans le 2.4), qui est aussi celle retenue par [Ye and Keogh, 2009], consiste à calculer la distance (minimale) entre chaque attribut candidat et chaque objet. Puis, pour chaque candidat, on ordonne les objets selon la distance les séparant de ce dernier. Enfin, on évalue la capacité de discrimination du candidat.

Cet indicateur, issu des méthodes de *Feature Selection* dont [Chandrashekar and Sahin, 2014] étudie le domaine, permet de déterminer les plus pertinentes des sous-parties. Cette évaluation est réalisée par le gain d'information (Mutual Information) qui vise à mesurer la dépendance entre 2 variables.

On utilise ensuite les shapelets dans l'ordre des mieux "notés", pour créer un arbre de décision permettant de classifier de nouvelles séries temporelles.

Dans des améliorations ultérieures, [Lines et al., 2012] et [Hills et al., 2014] utilisent les

1.2. Méthodes pour extraire, sélectionner et valoriser des caractéristiques pertinentes

shapelets (les plus discriminantes) en calculant la distance minimale qui permet de "matcher" au mieux avec les nouveaux éléments à classer, c'est-à-dire les séries temporelles encore non annotées. Par la suite, ces distances deviennent les attributs utilisés pour l'apprentissage et non plus l'arbre de décision directement. Il devient ainsi possible de ramener le problème à de l'apprentissage supervisé classique, et donc utiliser d'autres méthodes plus efficaces que les arbres de décision. Les performances sont améliorées, tout comme le temps de calcul.

La sélection des sous-séries se décompose finalement en :

1. Extraction de toutes les sous-séries P de l'ensemble D
2. Pour chaque série S calcul de la distance minimale entre la série S et l'ensemble des sous-séries
3. Calcul de la pertinence des sous-séries en fonction de la classe d'appartenance de leurs séries
4. Sélection des sous-séries les plus pertinentes pour la classification

L'avantage évident de la méthode des shapelets appliquée aux séries temporelles est de pouvoir comparer des éléments de tailles très variées et dont on ne possède pas de connaissance a priori (comme le seraient les fonctions élémentaires de la méthode d'origine). Ce ne sont en effet plus les éléments dans leur globalité qui peuvent être utiles, mais la juxtaposition de certaines des sous-parties de ces derniers, aux caractéristiques diverses, qui mises ensemble permettent de déduire les classes d'appartenance.

L'idée brute soulevée ici est proche de celle des réseaux de neurones convolutifs initiés par [LeCun and Bengio, 1998], dont le principe consiste à détecter des zones d'images qui permettent la construction des caractéristiques (aussi appelés *features*, et qui deviennent des attributs pour la classification) les plus pertinentes et donc retenues. L'inconvénient qu'il convient néanmoins de relever est la complexité en temps de calcul de la méthode.

Introduction de l'aléatoire

Les descripteurs locaux d'un élément, quels qu'ils soient, nécessitent en premier lieu la sélection de sous-parties (points d'un même voisinage / d'une même zone) à partir desquelles ils vont être construits.

Lorsqu'elle est possible, la sélection exhaustive est logiquement à privilégier. Néanmoins, dans de nombreux cas concrets d'application, l'immense quantité de possibilités rend cette extraction irréalisable. Ainsi, il est parfois préférable de n'en garder que certaines. Appliquée à un contexte connu, la sélection (au moins partielle) de sous-parties répondant à des critères précis peut être bénéfique [Huber, 2004] [Shalom et al., 2008]. Néanmoins, dans un cas plus général, et en l'absence de connaissance dans le domaine spécifié, il est plus judicieux de considérer aussi des parties ne répondant a priori à aucune exigence.

L'idée de sélection aléatoire est notamment avancée par [Fischler and Bolles, 1981] pour la méthode *RANSAC*, qui permet un gain de temps non négligeable.

Parmi les nombreuses optimisations en terme de temps de calcul réalisées dans ce domaine, [Renard et al., 2015] utilise une extraction aléatoire de ces sous-parties dans le cadre de la classification de séries temporelles. En raison de la redondance des sous-parties discriminantes dans les données ciblées, les performances en termes de classification restent bonnes.

1.3. Conclusion

Dans ce chapitre, nous avons abordé les principales méthodes de traitement des objets 3D pour la classification, ainsi que les techniques utilisées en sélection de caractéristiques pertinentes, puis les *time series shapelets*, adaptation des shapelets d'origine aux séries temporelles.

Les méthodes actuelles traitant les objets 3D cherchent à les décrire en les projetant dans un autre espace de représentation, dont les valeurs obtenues sont appelées descripteurs. Ces descripteurs cherchent à simplifier les objets initiaux en les calant sur un modèle de représentation qui entraîne une perte importante de la qualité de la donnée. C'est d'autant plus le cas que nous avons affaire à des éléments dont les parties discriminantes en sont des extraits. Le résultat prévisible est que l'on est incapable de ne tenir compte que de certaines parties pour la classification de l'objet dans son ensemble. L'autre point bloquant est qu'il est impossible pour l'utilisateur, dans cette situation, de séparer le bon grain de l'ivraie lors de la compréhension de la méthode.

Ainsi, au vu des méthodes existantes en classification supervisée d'objets 3D, il n'est aujourd'hui pas possible de catégoriser des objets 3D sur la présence de phénomènes locaux, tout en justifiant le processus et les prévisions.

Néanmoins, certains travaux en classification de séries temporelles ont résolu ce problème en cherchant à extraire, pour chaque classification, les sous-parties les plus discriminantes. Des méthodes issues de domaines aussi variés que le traitement d'images et de séries temporelles présentent des possibilités encore inexploitées dans l'extraction de sous-parties discriminantes d'objets 3D.

C'est en nous basant sur ces concepts développés au cours des années passées, où la sélection aléatoire de potentielles caractéristiques en est une des extensions, que nous proposons une nouvelle méthode répondant aux contraintes citées plus avant. Cette nouvelle approche est décrite dans le chapitre suivant.

2. L'approche 3DRESC : une nouvelle méthode d'extraction de sous-parties pertinentes d'objets 3D

Une méthode permettant de classifier des objets 3D définis par un maillage et se basant sur une extraction adaptée de sous-parties pertinentes n'existe pas. C'est la raison pour laquelle il est ici proposé une nouvelle approche respectant ces contraintes.

La méthode proposée est à la croisée des différents domaines présentés dans le chapitre précédent :

1. les shapelets, appliqués à la fois à la classification supervisée de séries temporelles par [Ye and Keogh, 2009] et d'images par [Lozano Vega, 2015] ;
2. les calculs de similarité de surfaces 3D par l'extraction de descripteurs d'objets 3D.

Ces techniques apparaissent comme étant potentiellement complémentaires. En effet, le balayage par fenêtrage d'une image (sous forme d'un rectangle ou d'un carré extrait de cette image) par [Lozano Vega, 2015] permet de déterminer quelle partie est la plus intéressante à relever dans le cadre d'une classification particulière.

L'idée initiale de [Refregier, 2001] permet elle aussi de tester puis d'extraire les sous-parties les plus discriminantes d'un ensemble cohérent de données, dans le cadre de la classification souhaitée. Néanmoins, ce qui est envisageable avec un maillage dont on pouvait établir la liste exhaustive des candidats possibles, car de forme rectangulaire, et dans un ensemble prédéfini de tailles possibles, pose problème pour des maillages 3D irréguliers. Nous sommes en effet confrontés à une explosion du temps de calcul nécessaire à l'extraction et l'évaluation des très nombreux candidats.

Lorsque [Renard et al., 2015] propose une recherche aléatoire de sous-parties potentiellement discriminantes pour la classification de séries temporelles, ses résultats ne sont quasiment pas dégradés.

En effet, en raison de grandes quantités de données, la redondance des extraits discriminants permet de ne pas diminuer la performance de la prévision, si on la compare au modèle exhaustif.

Cette idée a été adaptée à notre problème, même si dans notre cas une bien plus faible part des candidats au titre de sous-partie discriminante (l'ordre de grandeur est d'environ $10^{-6}\%$ de l'ensemble, contre $10^{-1}\%$ pour les séries temporelles de [Renard et al., 2015]), a été conservée.

Ainsi, nous proposons d'adapter aux objets 3D ce qui a été fait avec les séries temporelles en classification supervisée, la sélection des sous-parties discriminantes utilisées pour

calculer les attributs d'apprentissage étant réalisée aléatoirement en ne sélectionnant qu'un extrait de l'ensemble des possibilités (Figure 2.1).

Afin de garder la plus grande souplesse possible, il est donc proposé de distinguer, d'une part, la sélection de sous-surfaces pertinentes, coeur de l'algorithme, et d'autre part la méthode de classification utilisée, qui elle pourrait être aisément adaptée à un autre contexte d'utilisation.

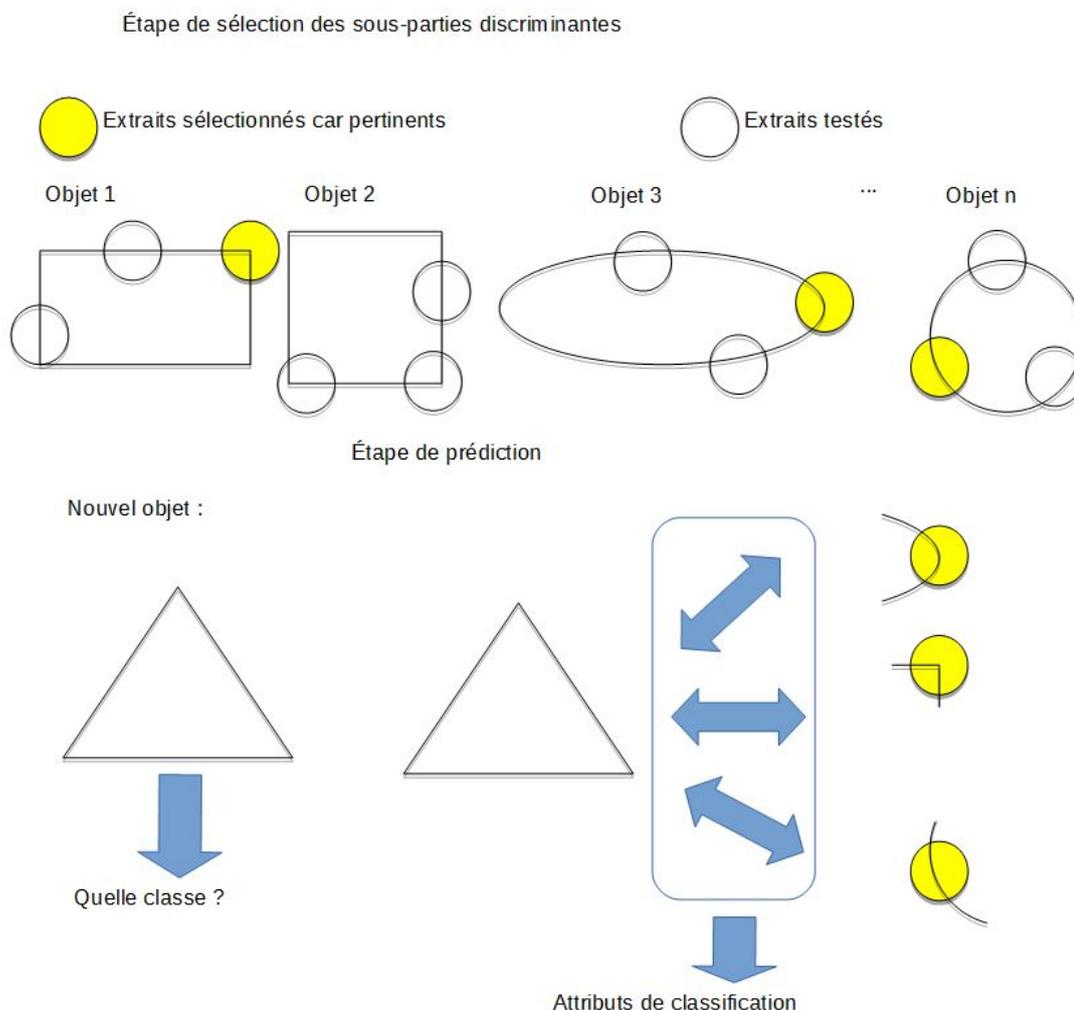


FIGURE 2.1. – Méthode proposée

2.1. Protocole explicatif

En cherchant à appliquer directement les *Time Series Shapelets* à la classification d'objets 3D, il est possible d'en assimiler directement les grandes étapes, que la Figure 2.2 schématise sur des objets 2D :

1. Extraction aléatoire de sous-parties

2. Évaluation de la pertinence des sous-parties extraites et récupération des mieux notées
3. Calcul des attributs des objets du jeu d'apprentissage à partir des sous-parties les plus pertinentes

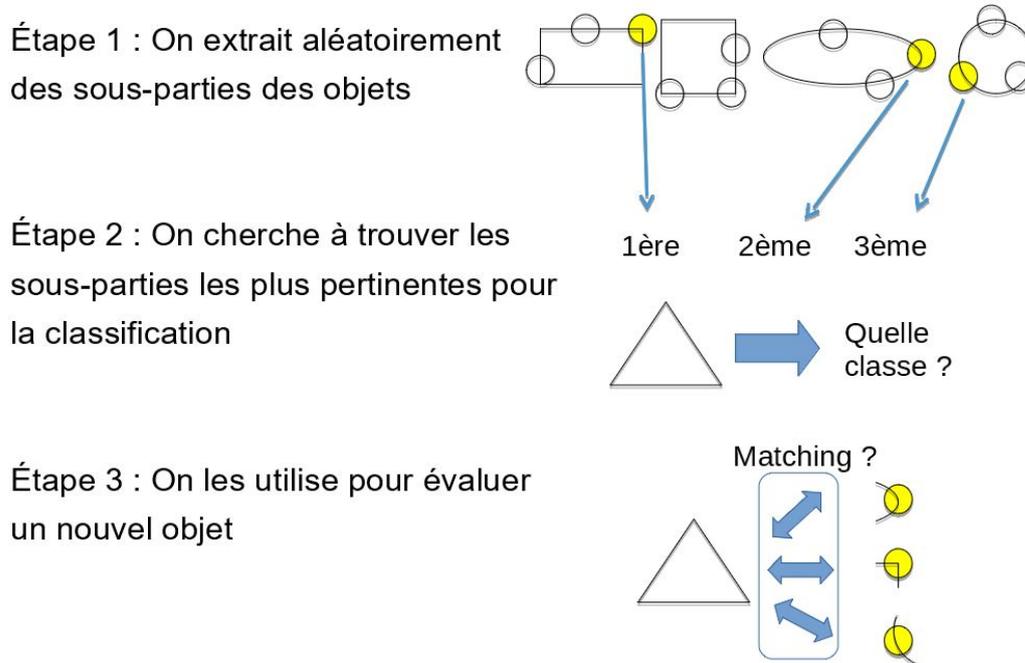


FIGURE 2.2. – Principales étapes de la méthode

Soient O_1, \dots, O_m l'ensemble des objets annotés qui constituent notre base d'apprentissage.

2.1.1. Extraction aléatoire de sous-parties

Dans un premier temps, il convient d'extraire ce que [Ye and Keogh, 2009] appelait des sous-séries. Pour rappel, en supposant $T = \{T_1, T_2, \dots, T_{T_{taille}}\}$ une série temporelle de taille T_{taille} , une sous-série S issue de T de taille $S_{taille} < T_{taille}$ et définie par : $S = \{T_a, T_{a+1}, \dots, T_b\}$ avec $1 \leq a < b \leq T_{taille}$ et $b - a = S_{taille}$.

Ce qui est défini, dans le cadre des séries temporelles, comme une sous-série, est appelé sous-partie pour les objets 3D. Une sous-partie de taille t est définie, de façon arbitraire, comme étant les t points de maillage les plus proches d'un point sélectionné aléatoirement sur l'objet. Ce point est donc la base de la sous-partie extraite de l'objet.

La méthode initiale des shapelets préconise l'extraction de l'intégralité des sous-séries possibles, donc des sous-parties dans notre cas. [Renard et al., 2015] propose de n'en sélectionner qu'une fraction d'entre elles, entre 10^{-3} et 10^{-6} , la répétition de motifs pertinents permettant de combler les zones potentiellement évitées par l'aléatoire.

Ici, cela revient à ne sélectionner qu'une possibilité sur 10^6 .

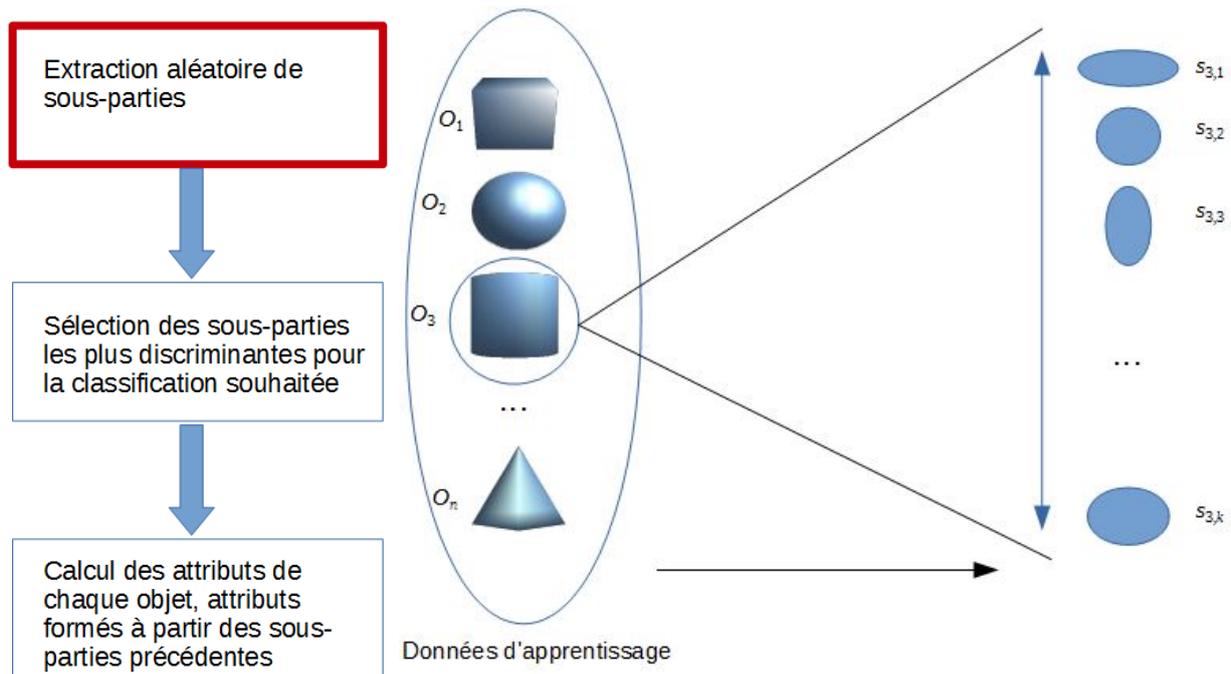


FIGURE 2.3. – Extraction aléatoire de sous-parties d'objets 3D

Une sous-série est définie par plusieurs paramètres :

1. la position de départ dans la série temporelle
2. la taille, nombre de valeurs contenues dans la sous-série

Combinés, ils conditionnent la position de la fin de la sous-série au sein de la série dont elle est issue.

Nous proposons de définir une sous-partie d'objet par :

1. le point du maillage, qui sert de base (centre) à la sous-partie
2. la taille, nombre de points de la sous-partie

Un tel parallèle confirme la possible transposition de cette méthode des *time series shapelets* à des objets 3D définis par des maillages de points.

La première étape consiste donc à extraire aléatoirement une partie des sous-parties possibles de tous les objets 3D de l'ensemble d'apprentissage, comme illustré dans la Figure 2.3 : $\forall O_{i=1,\dots,m}$, on extrait aléatoirement g sous-surfaces que sont $S_{i,1}, \dots, S_{i,g}$.

2.1.2. Evaluation de la pertinence des sous-parties

Une fois cet ensemble de sous-parties récupéré, la méthode des shapelets consiste en un classement de ces dernières selon leur pertinence par rapport à la classification souhaitée, ce qui ici correspond à leur capacité à scinder l'ensemble des objets selon leur classe.

Dans un premier temps, et ce pour chaque sous-partie $S_{i,j}$ évaluée, une mesure de distance entre celle-ci et tous les objets est calculée. Initialement, les *time series shapelets* résolvent ce problème en décrétant que, intuitivement, cette distance n'est autre que la capacité d'appariement d'une sous-série à une série entière, c'est-à-dire la plus petite distance entre la sous-série et toutes les sous-séries composant l'objet (elles aussi extraites aléatoirement), soit :

$\forall i' \in (1, \dots, m),$

$$Distance(S_{i,j}, O_{i'}) = \min_{l=1\dots m}(Distance(S_{i,j}, S_{i',l}))$$

La distance choisie entre 2 séries est la distance Euclidienne qui offre des résultats similaires à des méthodes plus poussées telles que le *DTW* (Dynamic Time Warping) si l'on dispose de beaucoup de sous-séries à comparer, mais offre un gain de temps important.

Grâce à la grande quantité de séries temporelles disponibles (plusieurs milliers par classification) et du fait de l'échantillonnage régulier de ces dernières, il est possible de ne comparer que les séries de même taille, ce qui permet l'application directe de la distance Euclidienne. En se basant sur ces ensembles de distances, on parvient à évaluer la capacité de discrimination d'une sous-série grâce à la méthode du gain d'information.

Les objets 3D, par la nature de leur maillage irrégulier et de leur faible nombre d'éléments d'apprentissage (au mieux une centaine), nécessitent de pouvoir en comparer des sous-parties de tailles différentes (en termes de nombre de points ou d'étendue de la surface) : la distance Euclidienne n'est donc plus adaptée.

Pour contourner ce problème, les descripteurs d'objets 3D (vus dans l'état de l'art) s'avèrent être le moyen le plus adéquat (Figure 2.4), afin de décrire les sous-parties dans un espace de représentation plus homogène, puis d'utiliser une mesure de répartition de probabilité afin de les comparer. Ce choix est expliqué par le fait que les descripteurs d'objets 3D permettent d'obtenir un indicateur de la géométrie aux alentours d'un point de la surface, mais dépendent toujours du nombre de points pris en compte. Pour résoudre ce problème, il est proposé de récupérer, non pas le descripteur en lui-même, mais des histogrammes de répartitions des valeurs des descripteurs de tous les points de la sous-partie. Cet ensemble d'histogrammes, dont on fixe les paramètres, correspondent à des répartitions de probabilités (une fois normalisés), et donc permettent une comparaison facile entre les sous-parties dont ils sont issus.

On conserve donc la formule initiale de la distance dite d'appariement entre une sous-partie (respectivement sous-série) et un objet (respectivement série) en ne changeant que la notion de distance, ce qui donne finalement **le degré d'appariement** :

$$Degre(S_{i,j}, O_{i'}) = \min_{l=1\dots m}(Distance_{histogrammes}(Descripteur(S_{i,j}), Descripteur(S_{i',l})))$$

Ce degré d'appariement d'une sous-surface à un objet permet d'établir une forme de proximité (ou distance) entre la sous-surface étudiée et chaque objet.

Les choix du descripteur et de la distance seront déterminés dans de prochaines expérimentations.

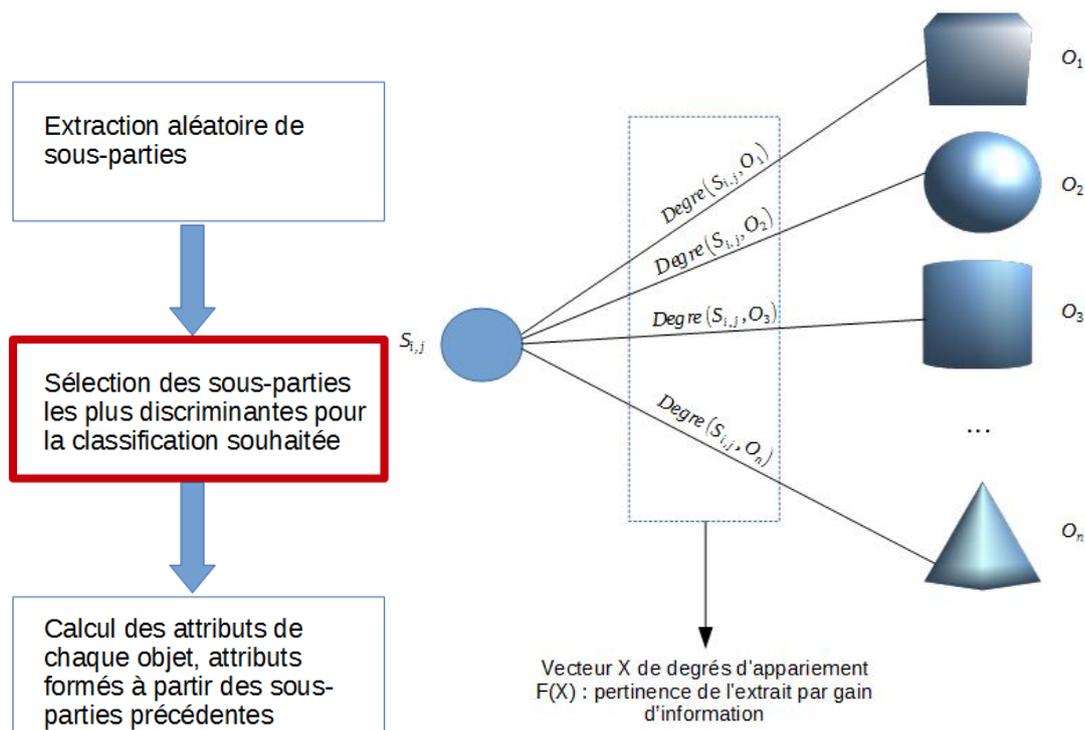


FIGURE 2.4. – Evaluation des sous-parties extraites en vue d'en sélectionner les plus pertinentes

A partir de ces valeurs de degré d'appariement, on peut calculer par gain d'information, comme pour les shapelets, les N sous-parties plus pertinentes. Cette valeur N verra plusieurs valeurs potentielles testées par la suite.

2.1.3. Calcul des attributs des objets à partir des sous-parties les plus pertinentes

Les sous-parties les plus pertinentes étant choisies, il ne reste plus, afin d'obtenir les valeurs des attributs propres à chaque objet, qu'à calculer ce même degré d'appariement (Figure 2.5) entre l'objet en question et chacune des sous-parties pertinentes sélectionnées précédemment.

2.1.4. Récapitulatif

Ces calculs permettent finalement d'extraire des objets 3D un certain nombre de valeurs les décrivant en se basant sur les sous-parties pertinentes. Cela revient à projeter les objets dans un nouvel espace dont les dimensions ne sont autres que chacune des sous-parties, pour former les attributs de la classification, comme illustré en Figure 2.6.

Suite à cela, un modèle d'apprentissage supervisé classique est utilisé pour l'apprentissage automatique. Le cœur de la méthode réside donc dans le calcul des attributs permettant de décrire au mieux les objets (respectivement séries temporelles) à partir de sous-objets (sous-séries) sélectionné(e)s selon leur pertinence vis-à-vis de la classification en cours.

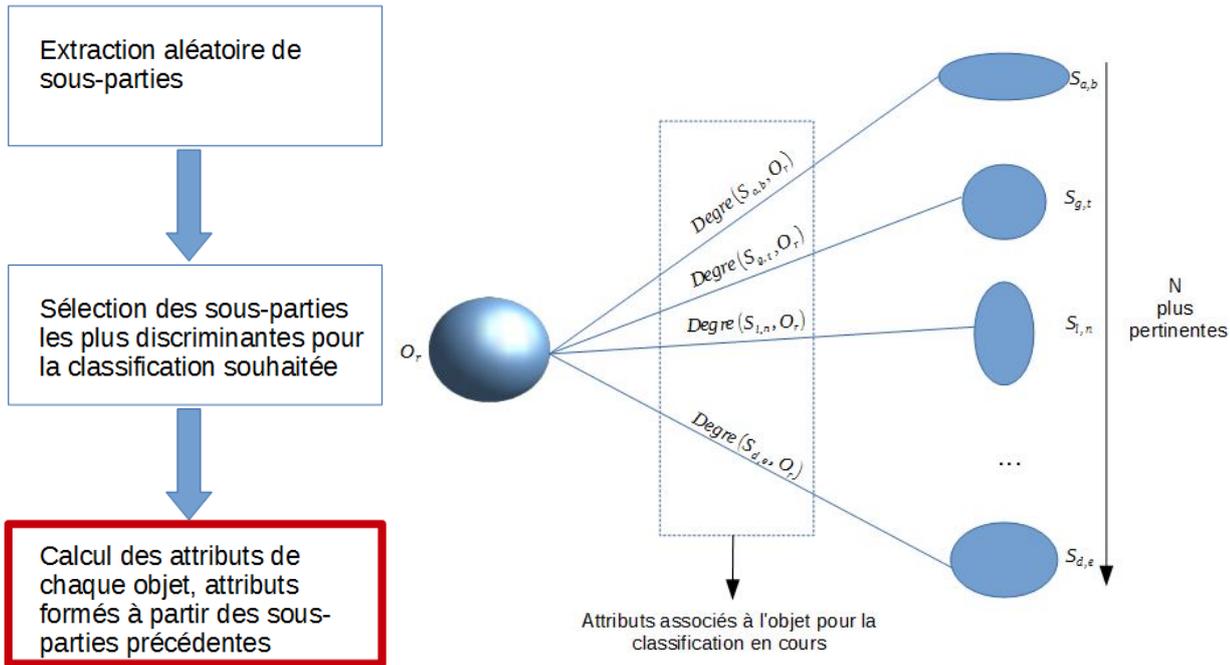


FIGURE 2.5. – Calcul des attributs à partir des sous-surfaces pertinentes

Considérons les versions améliorées des time series shapelets, c'est-à-dire celles n'utilisant pas un arbre de décision comme [Ye and Keogh, 2009] le propose mais celles qui, à partir de sous-séries et de leurs proximités à une série entière, calculent directement les attributs de la série [Lines et al., 2012] [Hills et al., 2014]. La Table 2.1 permet une comparaison aisée des similitudes et différences entre la méthode des time series shapelets dans ses dernières versions avec extraction aléatoire, et la méthode proposée dans cette thèse.

2.2. Protocole formel

Soient O_1, \dots, O_m l'ensemble des objets annotés qui constituent la base d'apprentissage.

- pour chaque objet O_i , on extrait aléatoirement g sous-surfaces que sont $S_{i,1}, \dots, S_{i,g}$.
- On considère un descripteur d'objet 3D particulier $Desc$, et une distance $Dist$. $\forall i \in (1, \dots, m)$ et $\forall j \in (1, \dots, g)$, on calcule le descripteur $Desc(S_{i,j})$ de la sous-surface $S_{i,j}$.

Puis, pour chaque sous-surface $S_{i,j}$ à évaluer, on calcule son degré d'appariement avec chaque objet :

$$\forall i' \in (1, \dots, m),$$

$$(S_{i,j}, O_{i'}) = \min_{l=1 \dots g} (Distance_{histogrammes}(Descripteur(S_{i,j}), Descripteur(S_{i',l}))) \quad (2.1)$$

Suite à cela, on sélectionne les N sous-parties ayant eu le meilleur $Degré$:

$$SousPartiesSelectionnees = \{S_{i_1, j_1}, S_{i_2, j_2}, S_{i_3, j_3}, \dots, S_{i_N, j_N}\}$$

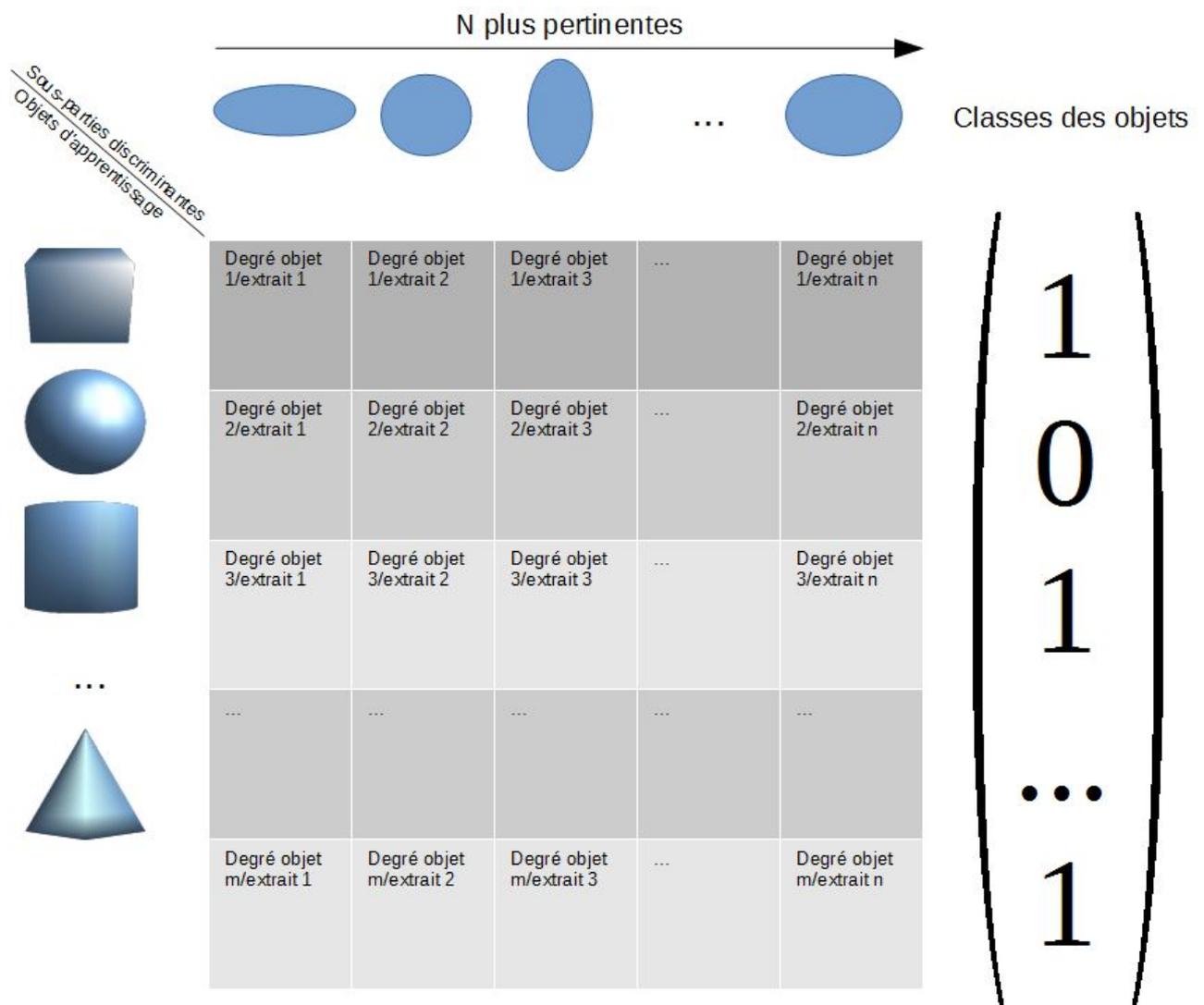


FIGURE 2.6. – Représentation des objets 3D permettant de récupérer des attributs à partir de sous-parties

Etapes	Times Series Shapelets	3DRESC
Type d'éléments classifiés	séries temporelles	objets 3D
Extrait récupéré	sous-série	sous-partie
Nombre d'éléments d'apprentissage	plusieurs milliers	au mieux une centaine
Sélection des sous-parties	exhaustive ([Ye and Keogh, 2009]) puis aléatoire ([Renard et al., 2015], [Wistuba et al., 2015])	aléatoire
Paramètres des sous-parties	début et taille	centre et taille
Fraction des sous-parties testées	entre 10^{-3} et 10^{-6}	moins de 10^{-6}
Distance entre deux extraits	distance Euclidienne directe entre séries	distances entre descripteurs d'objets 3D
Tailles de deux extraits comparés	mêmes tailles	tailles potentiellement différentes

TABLE 2.1. – Comparaison des 2 méthodes de classifications

Enfin, pour pouvoir extraire des attributs depuis les objets à classifier (qu'ils soient de la base d'apprentissage ou non), on réutilise le calcul du degré a_{ij} de chaque sous-partie j avec l'objet i en question :

$\forall O_{i=1,\dots,L}, \forall SousPartiesSelectionnees_{j=1,\dots,g}$:

$$a_{ij} = \text{Degré}(SousPartiesSelectionnees_j, O_i)$$

avec L le nombre d'objets à classifier, N le nombre de sous-parties sélectionnées car pertinentes, a_{ij} le $j^{\text{ème}}$ attribut de l'objet O_i calculé à partir de la $j^{\text{ème}}$ sous-partie. a_i est donc le vecteur d'attributs de l'objet O_i .

On dispose finalement, pour chaque objet $O_{i=1,\dots,L}$, d'une liste d'attributs $a_{i=1,\dots,N}$, permettant l'application directe de méthodes telles que les arbres de décision, les forêts aléatoires, les SVM, etc...

2.3. Descripteur local d'objet 3D et distance liée

Comme abordé précédemment, le calcul de la distance point à point étant difficilement réalisable pour des objets 3D aux maillages irréguliers, il devient indispensable de chercher un autre terrain de comparaison. Ceci peut être réalisé grâce à un descripteur invariant selon les transformations géométriques classiques. L'état de l'art a permis de faire ressortir les descripteurs suivants comme de bons candidats potentiels, car correspondant à nos critères :

- Le Heat Kernel Signature (*HKS*)
- Le Unique Shape Context (*USC*)
- Le Point Feature Histogram (*PFH*)

Une fois le descripteur local choisi, il est calculé pour chaque point de la surface et l'ensemble de ces valeurs est rassemblé dans un histogramme de répartition. De cette façon, le calcul de ce descripteur est en effet indépendant du nombre de points par surface extraite.

Il en aurait été de même pour un objet complet (exemple de [Sun and Ponce, 2013] pour le *HKS*). Le choix du descripteur le plus efficace parmi ces trois sélectionnés se base sur les performances.

Suite à cela, la distance de Bhattacharyya et la divergence de Kullback-Leibler sont testées pour chaque descripteur, afin de récupérer les meilleures combinaisons de descripteurs et de distances.

2.4. Sélection aléatoire des candidats

Une fois les descripteurs calculés et les distances entre sous-parties recueillies, il convient de chercher à évaluer ces entités afin d'en extraire les plus pertinentes pour la classification souhaitée. L'évaluation de l'ensemble des candidats étant impossible, une pré-sélection aléatoire de ces derniers s'est imposée.

L'idée initiale, inspirée de [Wistuba et al., 2015] et [Renard et al., 2015], suppose :

- un rapport minimal (de l'ordre de 10^{-3} pour [Renard et al., 2015]) entre le nombre de parties sélectionnées et celles qui ne le sont pas ;
- une répétition des extraits pertinents afin qu'une couverture, même faible, puisse en détecter une partie.

Concrètement, cela signifie, pour le premier point, un rapport d'environ 10^{-3} entre les parties sélectionnées et celles qui ne le sont pas. Néanmoins, et on a pu le voir plus haut, nous sommes confrontés à un rapport beaucoup plus faible dû à un nombre de combinaisons bien plus grand.

Afin de couvrir au mieux les possibilités, nous proposons de chercher, dans un premier temps, les surfaces correspondant grossièrement à des disques de rayons inclus dans un certain intervalle, rayon dépendant lui-même de la taille totale de l'objet : de $\frac{1}{15}$ à $\frac{1}{25}$ de l'objet total. De plus, étaler de façon régulière la source d'une partie des extraits candidats, permettrait d'éviter les redondances entre extraits extrêmement proches (et donc quasi-semblables du point de vue du descripteur d'objet). La proportion optimale d'éléments à échantillonner de façon totalement aléatoire (par rapport à ceux qui dépendraient d'un aléatoire "dirigé") étant alors inconnue, il paraît plus opportun de chercher d'abord à évaluer l'apport de cet aléatoire dirigé. Par cette méthode, on réduit le champ des possibilités, et donc le facteur de sélection des extraits testés.

Autre point, il est également pertinent de pouvoir sélectionner les zones dont on sait à l'avance qu'elles détiennent probablement des éléments pertinents : par exemple, une forte dérivée de la forme locale pourrait être un indice à considérer.

Pour récapituler, trois types de sélection de sous-surfaces sont possibles, à savoir :

- l'aléatoire total ;
- l'aléatoire partiel, conditionné afin que deux sous-surfaces ne se recoupent pas au delà d'un certain pourcentage de recouvrement (étalement des extraits choisis), que l'on peut assimiler ici à la distance entre leurs centres ;

— pas d'aléatoire, le choix de zones répond à des caractéristiques précises (importante dérivée ou forte densité de points par exemple), ce qui biaise l'analyse car suppose une connaissance qui n'est pas forcément disponible ; on exclut dès maintenant cette possibilité, la connaissance nécessaire n'étant pour l'heure pas encore disponible.

Cette phase de la méthode, qui conditionne le recouvrement par les shapelets de la surface étudiée, est primordiale et dépend pour partie des données traitées.

2.5. Sélection des sous-parties discriminantes parmi les candidats

Pour sélectionner les sous-parties les plus pertinentes, il est possible d'utiliser des critères supervisés ou non.

Les N premiers candidats selon le critère choisi sont utilisés pour le calcul des attributs, N est défini expérimentalement à environ 0.1% du nombre total d'extraits.

Il est logique de ne pas mettre plus d'attributs que la moitié du nombre de sous-parties extraites par objet 3D. Cela peut s'expliquer assez aisément par le fait que si l'on met autant de sous-parties attributs que d'extraits d'un objet 3D, alors les attributs manqueront de choix dans la sélection de l'extrait le plus proche lors du calcul de distance entre objet et shapelet candidat.

Une fois les sous-surfaces les plus pertinentes extraites, ces dernières sont utilisées pour, d'une part, réaliser l'apprentissage sur un modèle de classification supervisé traditionnel (les degrés entre l'élément d'apprentissage, et chacune des sous-surfaces donnent le vecteur d'attributs), et d'autre part réaliser le même procédé pour la prévision de nouveaux objets.

2.5.1. Sélection supervisée de sous-parties pertinentes

Pour l'évaluation de la pertinence de la sous-surface à partir des degrés, et bien que [Ye and Keogh, 2009] utilise la méthode du gain d'information pour évaluer les shapelets candidats, il s'avère que [Lines et al., 2012] propose l'utilisation de la formule *f-stat*, selon lui plus performante et plus précise pour évaluer le caractère discriminant d'une sous-partie que la méthode classique. Néanmoins, cela ne ressortait pas dans nos expérimentations, d'où l'abandon précoce de cette méthode.

L'avantage de la sélection supervisée de sous-parties est qu'elle se rattache à la sélection de caractéristiques (ou features), domaine très utilisé en apprentissage.

2.5.2. Sélection non supervisée de sous-parties pertinentes : valorisation de données non étiquetées

Les expériences précédentes montrent une petite amélioration par rapport aux prévisions de l'état de l'art. Cependant, il est souhaitable que les experts obtiennent de meilleures prévisions afin d'accorder plus facilement leur confiance au modèle.

Le problème de la sélection des extraits les plus pertinents pour caractériser une classe est primordial. En effet, les sous-parties (caractéristiques extraites des objets 3D) les plus pertinentes pour la classification pourraient potentiellement ne pas appartenir à un élément (objet 3D) du jeu d'apprentissage. C'est la raison pour laquelle nous proposons de sélectionner pour partie les extraits au sein de données dont on ne dispose pas de la classe d'appartenance.

Pour revenir aux séries temporelles utilisées dans la version originales des shapelets, certaines façons de créer des sous-parties pertinentes étaient de reconstruire les shapelets [Hou et al., 2016] (afin de créer certaines d'entre elles qui n'existent pas dans les ensembles de données) et de les utiliser comme des éléments discriminants. Néanmoins, si l'on cherche à appliquer ces principes, deux principaux problèmes apparaissent :

- ces formules n'appartiennent pas à un objet réel, donc ne sont pas facilement interprétables ;
- il est nécessaire de disposer de descripteurs pour notre méthode 3DRESC, la reconstruction d'un objet correspondant à ce descripteur devient encore plus difficile alors que la complexité augmente considérablement.

Principes de l'approche envisagée

Pour résoudre ce problème, nous proposons de sélectionner d'autres sous-parties à partir d'autres objets 3D, eux-mêmes issus d'un jeu de données différent : les données non étiquetées, celles dont on ignore la classe d'appartenance.

Pour évaluer la pertinence de ces sous-parties, il faut utiliser un autre critère que la méthode du gain d'information (2.2.1), celle-ci nécessitant de connaître les classes.

L'idée d'une évaluation non supervisée est de sélectionner, à partir des données qui ne sont toujours pas étiquetées (donc autres que les ensembles de données sur lesquels on apprend lors d'une classification supervisée), d'autres sous-parties qui répondent à un critère de pertinence différent.

Pour ce faire, il est nécessaire d'utiliser trois ensembles de données : l'ensemble d'apprentissage classique, le jeu de tests et l'ensemble de référence, qui est celui utilisé pour la sélection non supervisée de sous-parties pertinentes selon le critère particulier fixé. Cet ensemble de référence est composé d'objets 3D eux-mêmes logiquement non étiquetés. Ce dernier est effectivement très présent dans les bases de données, le problème de la grande quantité de données non annotées étant récurrent.

De cette façon, nous espérons apporter de la variété à la sélection de sous-parties déjà recueillie par la méthode classique. Cette sélection non supervisée est proposée pour valoriser l'ensemble des données inutilisées. Ceci dans le but de valoriser et d'utiliser au mieux de l'information absente que les données habituellement utilisées, à savoir les données annotées, ne contiennent pas.

Méthode proposée

Dans l'optique de pouvoir adapter à un contexte différent des méthodes issues des séries temporelles, le critère utilisé pour évaluer la pertinence d'une sous-série sans connaître les

classes d'appartenance des séries, et proposé par [Zakaria et al., 2012], est le suivant :

$$Score_{nonSupervise} = moyenne(degre(S, D_A)) + ecartType(degre(S, D_A)) \quad (2.2)$$

où *degre* reste la distance minimale entre la sous-surface étudiée et toutes les sous-surfaces d'un objet particulier, *S* la sous-partie évaluée et *D_A* l'ensemble des objets les plus proches qu'une distance particulière, appelée point de partage ou *split point* (voir [Ye and Keogh, 2009] pour plus de détails sur le principe du point de partage).

L'idée est de permettre de sélectionner, à partir des formules extraites du domaine source, celles qui obtiennent le meilleur score dans les éléments non marqués du domaine cible. Plus concrètement, cela revient à utiliser un travail sur du clustering de séries temporelles basé sur des shapelets [Zakaria et al., 2012] (ou extraction de sous-séries), pour déterminer la capacité des sous-surfaces à diviser un ensemble non annoté. L'algorithme 3 concrétise l'idée explicitée.

Nous proposons donc de compléter l'ensemble de sous-surfaces récupérées par la méthode classique à l'aide d'une sélection adaptée à l'absence d'étiquetage.

La méthode de [Zakaria et al., 2012] peut être résumée simplement comme la recherche de "formes" (dans son cas, extraits continus de séries temporelles) qui correspondent aux caractéristiques prototypiques des groupes ou classes.

Intuitivement, cela revient à chercher des sous-parties qui "répartissent" le plus possible les objets sur l'axe de distance à ces dernières, celles qui plus aptes à discriminer les objets selon leurs classes d'appartenance.

En outre, nous proposons non seulement d'utiliser l'ensemble de données de test de façon non supervisée, mais aussi la partie de d'apprentissage supervisé classique (cf Algorithme 2). De cette façon, il est possible d'évaluer les sous-parties par un autre moyen, et donc d'introduire de la diversité dans notre sélection.

2.6. L'algorithme 3DRESC

Nous proposons l'algorithme 3DRESC pour 3D Random Extraction of Sub-parts for Classification (cf Algorithme 1) qui récapitule les principales étapes de l'approche proposée dans les sections précédentes. Les parties que l'on peut retrouver dans les références citées n'ont pas été à nouveau explicitées, dans le but de ne pas alourdir ce dernier.

2.6.1. Description

Dans l'algorithme 1, on a :

- *Extraction(objet, tailles, nombres)* qui extrait d'un objet 3D un certain nombre de sous-parties de tailles prédéfinies ;
- *CalculDescripteur(sous – surface)* qui calcule le descripteur pour chaque point de la sous-surface ;
- *CalculDistances(Desc1, Desc2)* qui calcule, à partir de deux descripteurs de 2 objets, la similitude de leurs histogrammes de répartition de valeurs.

Algorithme 1 : 3DRESC

```

1  Entrée : Objets : objets 3D ; Classes : classes correspondantes ; N : nombre d'attributs
   souhaité ; tailles : tailles des extraits ; nombres : nombre d'extraits pour chaque taille;
2  Sortie : SousParties : Liste de sous-parties discriminantes d'objets 3D;
3  Début
4   $i \leftarrow 0$ ;
5  pour Objet dans Objets faire
6  |    $i \leftarrow i + 1$ ;
7  |    $j \leftarrow 0$ ;
8  |    $candidats \leftarrow \text{Extraction}(\text{Objet}, \text{tailles}, \text{nombres})$ ;
9  |   pour candidat dans candidats faire
10 | |    $j \leftarrow j + 1$ ;
11 | |    $\text{DescObjet3D}[i][j] \leftarrow \text{CalculDescripteur}(\text{candidat})$ ;
12 |   fin
13 fin
14  $i \leftarrow 0$ ;
15 pour Objet1 dans Objets faire
16 |    $i \leftarrow i + 1$ ;
17 |    $j \leftarrow 0$ ;
18 |   pour Objet2 dans Objets faire
19 | |    $j \leftarrow j + 1$ ;
20 | |   si Objet1 différent de Objet2 alors
21 | | |    $\text{Distances} \leftarrow \min(\text{CalculDistances}(\text{DescObjet3D}[i], \text{DescObjet3D}[j]))$ ;
22 | | |    $\text{Gains} \leftarrow \text{CalculGains}(\text{Distances}, \text{Classes})$ ;
23 | | |    $\text{ListeGain}[i] \leftarrow \max(\text{Gains})$ ;
24 |   fin
25 fin
26  $\text{ListeGain} \leftarrow \text{Ordonner}(\text{ListeGain})$ ;
27  $\text{SousParties} \leftarrow \text{ListeGain}[1, \dots, N]$ ;
28 Fin

```

- $CalculGains(distances, classes)$ qui calcule le gain d'information de l'attribut correspondant pour la classification souhaitée.

2.6.2. Complexité algorithmique

Posons :

- NO le nombre total d'objets (jusqu'à 100 dans nos expériences)
- NT la nombre moyen de triangles (mailles) par objet (environ 3000)
- NE le nombre de sous-parties extraites par objet
- TE la taille moyenne des sous-parties extraites ($\frac{NT}{20}$ qui est d'environ 150 dans notre expérience)

L'algorithme 3DRESC [Meunier et al., 2017b] a une complexité d'environ $O(NO * TE^2 * NE)$, ce qui correspond à $2.25 * 10^6 * NE$. La méthode globale, appelée ici *HKS* globale, (qui revient à calculer le *HKS* sur chaque point, et ensuite réaliser un histogramme sur l'ensemble des points pour créer les attributs), nécessite approximativement $O(NO * NT^2)$, ce qui correspond à $9 * 10^8$. Finalement, ces deux méthodes sont relativement équivalentes en terme de temps de calcul si le nombre de sous-parties extraites est d'environ

$$\frac{9 * 10^8}{2.25 * 10^6} = 4 * 10^2$$

soit 400 sous-parties extraites, équivalent à l'ordre de grandeur de nos expériences.

Pour ce qui est des descripteurs d'objets 3D, si l'on considère un extrait de taille TE , un rapide aperçu des formules données nous renseigne sur leur complexité algorithmique, qui est systématiquement de $O(TE^2)$. Les différences en termes de temps de calcul ne devraient être significatives à ce niveau, l'ordre de grandeur restant le même.

Il était exclu de tester en augmentant sensiblement le temps de calcul (avec plus de sous-parties par exemple), car l'utilisateur pourrait facilement être amené à devoir utiliser la technique introduite ici sur des données plus nombreuses, et l'on risquerait de voir une explosion du temps de calcul, chose qui la rendrait inutilisable dans un temps raisonnable (quelques jours, selon nos critères).

2.6.3. Variante avec sélection non supervisée

La combinaison devrait être telle que sa performance s'améliore, si on la compare à la méthode classique, ce en utilisant le même nombre d'attributs.

Algorithme 2 : Utilisation du non supervisé pour l'amélioration de la sélection des sous-parties discriminantes

```

1 Entrée ensemble : ensemble de valeur de rapports entre les 2 mesures utilisées ;
   GainInformation : mesure du Gain d'Information pour pertinence supervisée des extraits ;
   UnsupervisedTest : mesure en non supervisée pour pertinence des extraits ;
   Algorithme3DRESC-TF(DS,DT) ; ; Performance(Fonction) : évaluation de la performance,
   par validation croisée, de la fonction classifieur en entrée - utilisation de la méthode mise au
   point dans la Partie 1;
2 Sortie algorithmeFinal : algorithme finalement obtenu en décelant le bon rapport entre extraits
   récupérés par études supervisée et non supervisée;
3 Début
4 pour  $\alpha_{test} \in \alpha_{ensemble}$  faire
5   | fonctionPertinence  $\leftarrow$ 
6   |   GainInformation(Shapelet, Objets, Classes) +  $\alpha$ .UnsupervisedTest(Shapelets, Objets);
7   | ResultatAlgorithme3DRESC-TF  $\leftarrow$  Algorithme3DRESC-TF(DS,DT)(fonctionPertinence);
7 fin
8 algorithmeFinal  $\leftarrow$   $\max(\text{ResultatAlgorithme}_{3DRESC-TF})$ ;
9 Fin

```

2.7. Bilan

Ce chapitre décrit la méthode proposée, 3DRESC, une nouvelle méthode de classification supervisée d'objets 3D non fermés décrits par un maillage irrégulier.

Inspirée de la méthode des shapelets, qui s'est avérée être un succès, respectivement sur les images et les séries temporelles, l'intérêt de cette méthode n'est autre que de chercher à adapter la notion de "shapelets" aux objets 3D. Les parties les plus discriminantes sont sélectionnées par le processus d'apprentissage, et donc interprétables telles quelles par l'utilisateur, permettant à ce dernier d'en comprendre le cheminement. Elle est basée sur un postulat, qui est que la classification dépend de certaines sous-parties des objets.

Par ailleurs, une variante dans la sélection des sous-parties sélectionnées est proposée.

3. Expérimentations sur la méthode 3DRESC

L'algorithme 3DRESC étant adapté à un contexte de classification de grandes structures 3D dont les éléments permettant cette dernière ne sont que des sous-parties, il est assez compliqué de trouver un jeu de données correspondant à nos besoins. En effet, la classification supervisée d'objets 3D répond généralement au besoin de classer des objets fermés dont l'objet dans son ensemble appartient à une classe, alors que notre objectif consistait à détecter des phénomènes locaux, sans que l'on en connaisse leur forme ou leur position au sein des structures 3D.

Ensuite, le choix du calcul de similarité entre sous-parties (afin de calculer la distance entre un candidat et un objet 3D) est assez gourmand en temps de calcul. Encore à l'état expérimental, il n'a pour l'heure pas été optimisé mais cela fera l'objet d'un développement futur.

Enfin, notre méthode vise aussi à pouvoir aider l'utilisateur à comprendre la classification. Sont importantes :

- non seulement les performances permettant de valider la classification en elle-même (et donc le taux de prévision),
- mais aussi la capacité à fournir une explication à la classification proposée, ce qui constitue un point primordial.

La classification est donc complétée par une justification qui découle de la construction de la méthode.

Nous avons testé cet algorithme (ainsi que les améliorations ultérieures) avec des données libres du *Princeton Shape Benchmark*¹ (PSB), et dont l'avantage principal est de provenir de crawlers (programme informatique permettant de visiter automatiquement une ou plusieurs pages internet) ayant récupéré des objets de multiples jeux de données déjà existants. Cette base de données propose différentes granularités.

Par ailleurs, elle a l'avantage de posséder déjà une grande variété de formes différentes ce qui permet de couvrir différents types de problèmes susceptibles d'être rencontrés. Pour chaque batterie de tests, des extraits différents de cette base sont récupérés ce qui représente virtuellement des bases de données diverses et donc permet un cadre applicatif plus large. Par ailleurs, les bases de données en libre accès contenant des objets 3D sont généralement bien plus petites que celle-ci, mais, surtout, elles sont généralement plutôt adaptées à des domaines d'utilisation bien particuliers.

Ces données sont ceux qui, en termes de formats et de types, se rapprochent le plus des géo-modèles 3D utilisés dans le rendu applicatif chez Total. Les classes les plus fines sont

1. consulté le 29/03/2018

retenues (3 niveaux étant possibles), afin de rajouter encore davantage de contraintes et de difficultés à la prévision lors des tests. Ainsi, un même nombre d'objets issus de deux classes différentes sont sélectionnés aléatoirement. Une fois les sous-parties extraites, l'algorithme des forêts aléatoires de la librairie Python *scikit-learn* est utilisé pour construire le classifieur. Une validation croisée (*4-folds cross-validation*) est par la suite effectuée, afin de vérifier la stabilité des résultats. La mesure utilisée pour évaluer la performance reste l'*AUC* (*Area Under Curve ROC*), avec laquelle on précise l'écart-type de la validation croisée. Cette mesure de performance est calculée en utilisant la même librairie *scikit-learn* mentionnée précédemment.

Les tests sont effectués sur un processeur *Intel Core i7 vPro* avec 16 Go de RAM. L'échantillon comprend 40, 60 ou 100 objets selon l'expérience, extraits aléatoirement, et répartis en 2 classes de même taille. Les objets en eux-même ont des tailles très variables, allant de 250 points à 5000 points environ. Aucune normalisation n'ayant été faite, le faible nombre d'objets sélectionnés et le fort écart-type de taille d'objet (1946) fait logiquement baisser les résultats de la prévision.

3.1. Protocole expérimental

1. Dans un premier temps, le jeu de données est séparé en 5 parties de même taille, dont 4 servent à l'apprentissage et 1 au test (Figure 2.3), et est issu de 2 classes différentes de même taille. On fera en sorte que cet équilibre de classe soit maintenu dans chacune de ces parties.
2. Pour chaque sous-partie extraite, le degré d'appariement (2.1) de cette dernière avec chacun des objets de l'ensemble d'apprentissage est calculé (Figure 2.4). Ces degrés sont la base du calcul de pertinence de la sous-partie étudiée (gain d'information, par exemple). Les n sous-parties les plus pertinentes sont conservées comme constructeurs d'attributs pour la classification en cours.
3. Puis, pour chaque objet annoté (donc de l'ensemble de données disponible), les degrés d'appariement des n sous-parties précédemment sélectionnées par rapport à l'objet sont calculés. Ces n valeurs deviennent les attributs associés à l'objet pour la classification en cours (Figure 2.5).
4. On obtient finalement la matrice représentée en Figure 2.6, avec en lignes les objets annotés et en colonnes les attributs correspondant aux n sous-parties discriminantes.
5. Pour la phase de test (ou prévision), on dispose déjà des n sous-parties discriminantes, il ne nous reste alors plus qu'à appliquer les points 2 et 4, permettant de récupérer les attributs utilisés de la même façon qu'un système de classification supervisé traditionnel.

Il existe plusieurs méthodes permettant le calcul d'appariement d'une surface dans un objet, mais elles se basent systématiquement (pour des raisons de temps de calculs) sur l'utilisation d'extraits de l'objet cible pour le représenter, et non pas de l'objet dans son intégralité en tant qu'entité individuelle.

Ainsi, on récupère les distances entre sous-parties (la surface qui doit s'appareiller dans l'ob-

jet et les sous-parties extraites de celui-ci), comme dans la Figure 3.1, pour calculer le degré (2.1).

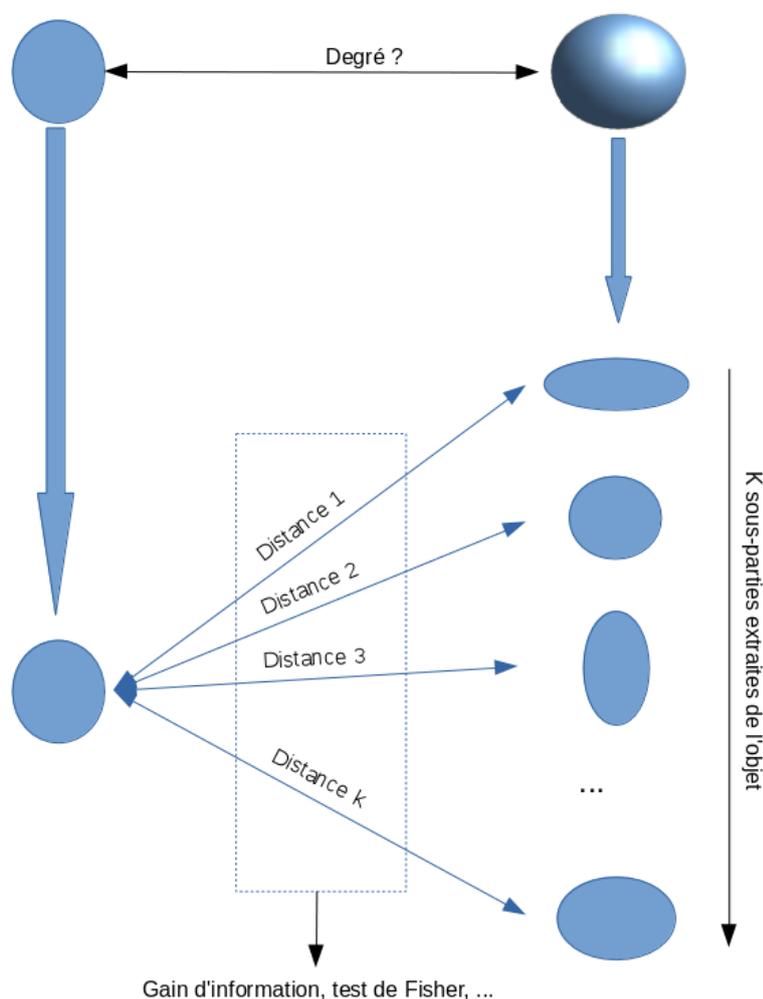


FIGURE 3.1. – La récupération des éléments nécessaires au calcul du degré d'appariement

3.2. Détermination des paramètres de l'approche

Les expériences réalisées avec des descripteurs de la littérature sont effectuées dans les mêmes conditions que celles utilisées pour la méthode proposée, à savoir les mêmes bibliothèques et la même extraction de descripteurs (selon que l'on en extrait de sous-parties ou de l'objet dans son ensemble).

Nous proposons les expérimentations suivantes :

1. Des tests visant à choisir un descripteur et un calcul de similarité ;

2. Des tests sur le descripteur retenu, le HKS, afin d'en vérifier la stabilité, et par là même la fiabilité ;
3. Des tests visant à fixer le nombre d'extraits par objet.

La taille des sous-parties extraites a quant à elle été échelonnée entre $\frac{1}{15}$ à $\frac{1}{25}$ de la taille de l'objet total. Cet intervalle s'est avéré correspondre au meilleur choix possible, étant suffisamment précis prendre en compte des détails de l'objet, sans pour autant rendre les sous-parties impossibles à interpréter.

3.2.1. Choix du descripteur et du calcul de similarité

On obtient, en extrayant à chaque fois 200 sous-surfaces de tailles approximatives de $\frac{1}{20}$ de chaque objet, par validation croisée (en apprenant sur 75% et en testant sur 25%) sur 60 objets (subdivisés en 2 classes de mêmes tailles), les résultats comparatifs entre les descripteurs d'objets 3D de la Table 3.1, que sont le *Heat Kernel Signature*, le *Point Feature Histogram*, et le *Unique Shape Context*.

Descripteur d'objets 3D	divergence de Kullback-Leibler	mesure de Bhattacharyya	distance de Hellinger	distance de Matusita	Temps de calcul moyen (secondes)
HKS	0.89 ± 0.13	0.85 ± 0.13	0.86 ± 0.07	0.90 ± 0.12	1208
PFH	0.79 ± 0.16	0.75 ± 0.14	0.60 ± 0.08	0.71 ± 0.12	1098
USC	0.86 ± 0.11	0.85 ± 0.15	0.84 ± 0.13	0.87 ± 0.11	1152

TABLE 3.1. – Performances comparées entre descripteurs d'objets 3D

En colonnes de la Table 3.1, on retrouve la méthode utilisée pour le calcul de la distance entre les sous-parties, ainsi que le temps de calcul, et en lignes les descripteurs à partir desquelles ces distances sont calculées.

Au taux de bonne classification, l'écart-type de la validation croisée y est aussi mentionné. Les légères différences en temps de calcul ne dépendent que de l'implémentation des algorithmes de descripteurs, et l'ordre de grandeur reste le même.

Les meilleurs résultats trouvés sont ceux du HKS, bien que le USC le talonne. Avec le jeu de données qui est le nôtre, il est apparu que la distance de Matusita, combinée à ce même HKS, semble former la meilleure combinaison.

On choisit donc, pour nos expériences futures, comme descripteur d'objet 3D le HKS et comme distance entre ces descripteurs la distance de Matusita.

On remarque néanmoins que n'importe quel descripteur d'objet 3D peut être utilisé pour décrire les sous-surfaces extraites, et donc que la vraie comparaison doit avoir lieu entre un descripteur global et notre méthode utilisant ce même descripteur pour les extraits.

Triangle 3.2	Distance de Manhattan	Distance de Matusita	Distance de Kullback-Leibler	Distance Euclidienne
A $par = 0.1$	0.0	0.0	0.0	0.0
A $par = 0.2$	0.0618	0.00257	0.00370	0.0336
A $par = 0.3$	0.142	0.0125	0.00905	0.0755
A $par = 0.4$	0.235	0.0331	0.0237	0.125
A $par = 0.5$	0.339	0.0667	0.0552	0.183
A $par = 0.6$	0.451	0.114	0.108	0.252

TABLE 3.2. – Comparaison de chaque triangle avec le premier à $t = 0.1$

3.2.2. Heat Kernel Signature

Le HKS étant le descripteur retenu, il est ici proposé de l'évaluer expérimentalement afin d'en confirmer la pertinence et l'adéquation avec la distance de Matusité.

Exemple sur des données simples

Afin de tester le HKS et l'impact de la forme, il est proposé de se baser sur une forme très simple (un tétraèdre), et de faire faire évoluer cette forme pour en observer l'impact sur le descripteur.

Dans le cadre d'un cas aussi simple, nous ne nous intéresserons qu'à l'instant $t = 0.1$, la suite convergeant très vite.

En posant comme cas d'étude un tétraèdre composé des points suivants : $p_1 = [0, 0, 0]$, $p_2 = [1, 0, 0]$, $p_3 = [0.5, 0.8, 0]$, $p_4 = [par, par, 0.8]$, par étant la valeur du paramètre que nous modifions. Et les arêtes suivantes (on indique les points reliés, pas d'orientation ici) : $\{p_1, p_2\}$, $\{p_2, p_3\}$, $\{p_3, p_4\}$, $\{p_1, p_4\}$, $\{p_1, p_3\}$, $\{p_2, p_4\}$

On propose de faire évoluer le paramètre par ce qui revient en déplaçant le point p_4 dans le plan parallèle à $\langle p_1, p_2, p_3 \rangle$ et passant par p_4 en l'éloignant de l'origine $[0, 0, 0]$ (cf la Figure 3.2).

Si l'on compare chaque tétraèdre avec le premier, soit $distance(HKS(tetraedre_1), HKS(tetraedre_k))$, et en appliquant différentes distances, on obtient le tableau 3.2.

Dans la Figure 3.3, les résultats obtenus sont représentés :

- en rouge, par la distance de Manhattan ;
- en violet la distance Euclidienne ;
- en bleu celle de Matusita ;
- en jaune celle de Kullback-Leibler ;
- en vert celle de Wasserstein.

On remarque que, sur cet exemple, deux distances se détachent : la distance de Matusita et celle de Kullback-Leibler.

En effet, ces deux distances offrent une certaine robustesse aux changements légers des formes, et ne commencent à être augmentées de façon notable qu'à partir d'une modification importante de l'objet 3D. Lorsque l'objet est peu modifié, la distance reste proche de zéro et n'augmente que petit à petit.

Néanmoins, et de façon purement intuitive, la distance de Kullback-Leibler, mieux adaptée aux comparaisons de répartitions de probabilités, offre l'avantage de paraître encore un peu plus robuste aux déformations légères que celle de Matusita (au début, sa courbe est plus

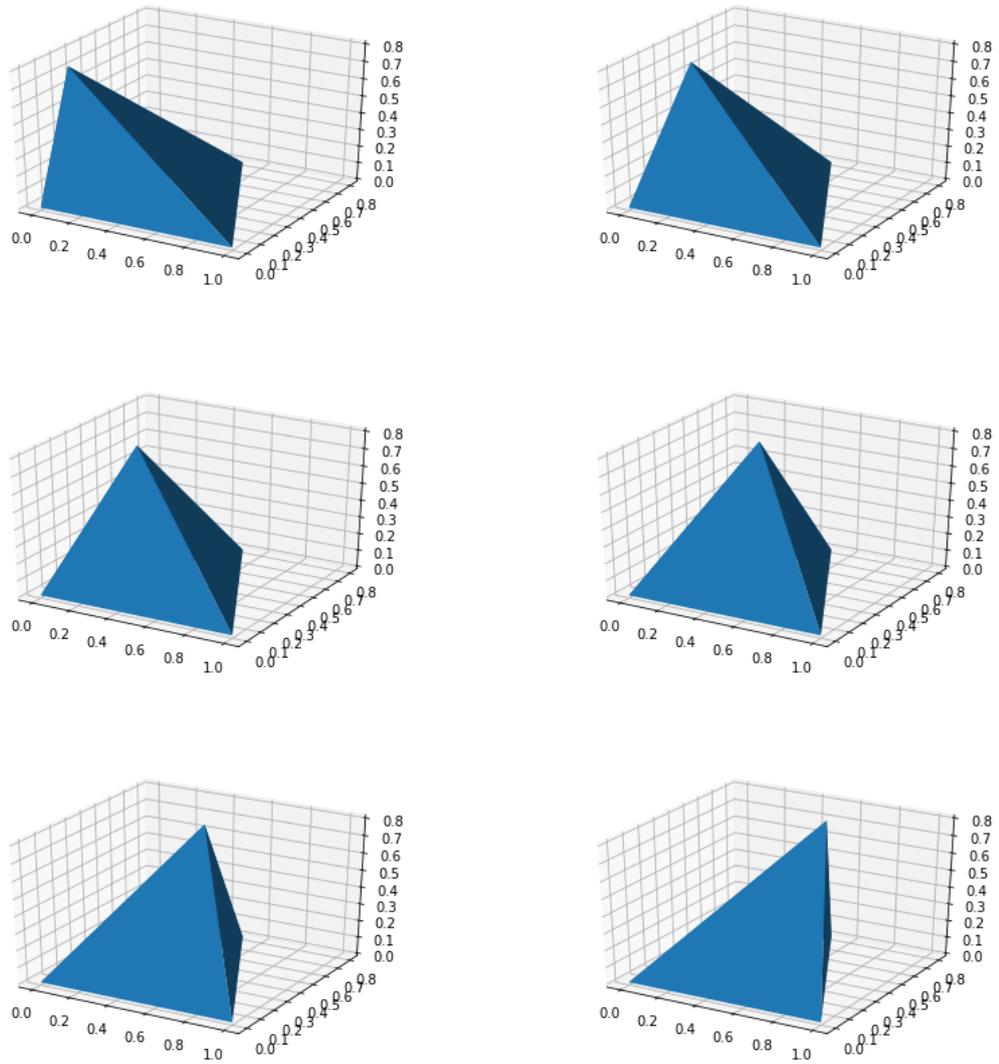


FIGURE 3.2. – Evolution du tétraèdre pour t allant de 0.1 à 0.6

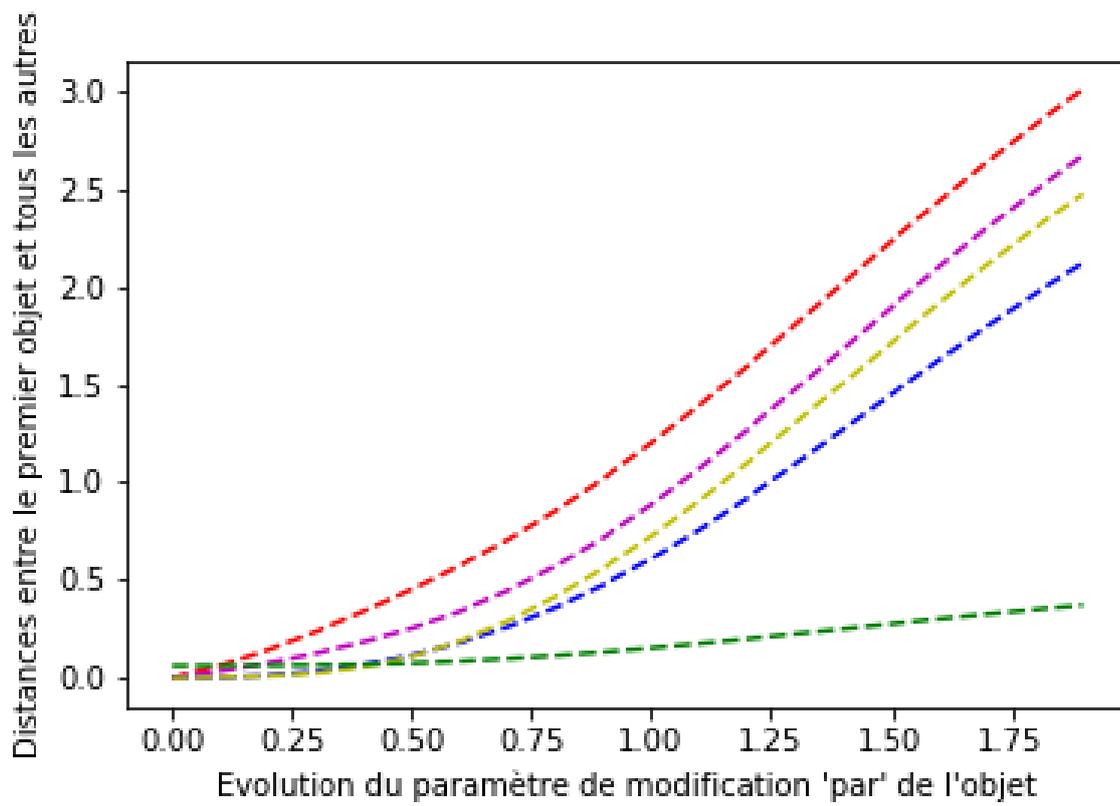


FIGURE 3.3. – Résultats comparés avec des distances différentes

Distance utilisée	Distance moyenne entre objets de même classe	Distance moyenne entre objets de classes différentes	Rapport entre ces 2 valeurs
Kullback-Leibler	460.17	1665.3	0.27632
Euclidienne	297.92	1111.7	0.26797
Matusita	328.94	1489.6	0.22082

TABLE 3.3. – Test de la pertinence des distances par du non supervisé

”basse”).

Les aires, angles et valeurs de HKS des triangles contenant le point P_i évoluent comme indiqué dans la Figure 3.4.

Si l’on travaille sur des objets de même taille et ne possédant pas trop de points, il est possible de tenir compte de l’ensemble des valeurs et vecteurs propres pour calculer les descripteurs. Néanmoins, et comme mentionné précédemment, on peut avoir affaire à de plus grands objets, ou plus simplement pouvoir comparer deux objets de tailles différentes (en nombre de points / segments / triangles) sans pour autant faire appel à un histogramme de répartition de ces valeurs. A cette fin, il convient de fixer un nombre limite de valeurs propres (et de vecteurs propres) à partir duquel le descripteur est calculé.

De cette façon, on parvient à proposer un descripteur uniforme, plus rapide à calculer, et tout de même suffisamment représentatif de l’objet ou de la sous-partie de l’objet pour permettre de discriminer selon les classes.

Dans un cas plus complexe

Il est donc proposé, dans cette sous-section, de chercher à retrouver par apprentissage non supervisé les classes des objets, ou du moins de pouvoir observer l’évolution de la distance entre descripteurs, en fonction de l’appartenance d’objets à une même classe ou non.

On dispose d’une centaine d’objets 3D, appartenant chacun à une catégorie. Certains groupes d’objets peuvent appartenir à la même classe.

Si l’on réalise une étude des distances, on obtient les moyennes rassemblées dans la Table 3.3.

A première vue, bien que les résultats soient proches, on note un léger avantage pour la distance de Matusita, ce qui tendrait à confirmer l’intuition qui fut la notre dans la sous-section 1.4.

Impact de la granularité sur le descripteur :

La notion de granularité définit la taille du plus petit élément, de la plus grande finesse d’un système.

Ici, il s’agit de la taille des mailles permettant de définir un objet. Un moyen d’augmenter la granularité (et donc de diminuer la précision de la représentation) est de retirer aléatoirement des points. Ce faisant, la description de l’objet en devient plus grossière.

Il est possible que deux objets, en plus d’être échantillonnés différemment, soient décrits

3.2. Détermination des paramètres de l'approche

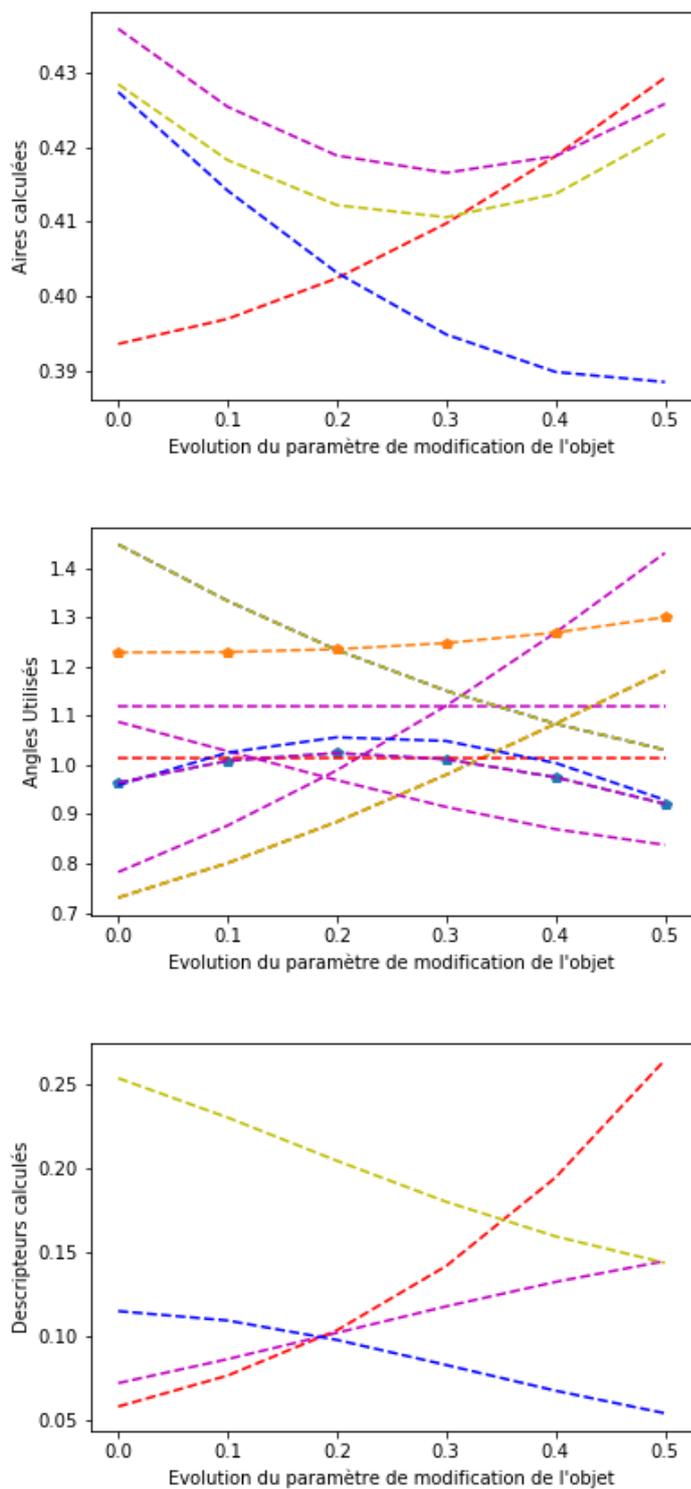


FIGURE 3.4. – Résultats comparés de l'évolution des aires, angles et valeurs de HKS attribuées au tétraèdre

Distance utilisée	0%	5%	10%	15%	20%	25%
Euclidienne	0	32.682	77.354	201.58	227.10	239.59
Matusita	0	18.636	63.407	165.09	206.61	221.63
Kullback-Leibler	0	30.368	96.905	216.53	288.72	309.57

TABLE 3.4. – Evolution du rapport entre la distance de l’objet de base avec chacune de ses versions aux granularités différentes

Distance utilisée	5 à 10%	10 à 15%	15 à 20%	20 à 25%
Euclidienne	2.3668	2.6059	1.1265	1.0549
Matusita	3.4023	2.6036	1.2514	1.0726
Kullback-Leibler	3.1910	2.2344	1.3334	1.0722

TABLE 3.5. – Evolution du rapport entre 2 distances successives de granularités

par des granularités différentes. Cependant, il peut très bien s’agir du même objet représenté. La question qui apparait est la suivante : dans quelle mesure la granularité, en fonction des paramètres que l’on a choisis, peut-elle influencer sur l’évolution du descripteur, évolution évaluée par sa distance vis-à-vis de l’objet d’origine ?

Pour cela, on propose de comparer une série d’objets à des versions d’eux-mêmes échantillonnées différemment. L’obtention des versions aux granularités différentes est possible en supprimant, de 5 à 25% du nombre total de points (et implicitement d’arêtes) d’un objet, puis de comparer les distances obtenues entre l’objet d’origine et chacun de ces nouveaux objets créés.

Comme le jeu de données sélectionné pour cette étude contenait une soixantaine d’objets, il était nécessaire d’en calculer l’évolution moyenne de distance, en prenant toujours comme base la première version de l’objet, la plus complète ; celle dont la granularité est donc la plus faible.

Les résultats de l’expérience proposée sont rassemblés en Table 3.5 et Figure 3.3.

Tout d’abord, il est nécessaire de rappeler que :

1. Les points supprimés le sont aléatoirement, des points essentiels comme des points qui ne le sont pas peuvent être retirés.
2. L’aléatoire ne garantit pas une répartition uniforme sur la surface.
3. Les distances retenues dans la comparaison initialement proposée en Figure 3.3 ont une allure exponentielle.

La Figure 3.3 présente l’éloignement d’un objet en fonction de la granularité qui lui est appliquée, par rapport à sa version d’origine. Réalisée sur une soixantaine d’objets choisis aléatoirement, la visualisation de la moyenne permet de mieux en appréhender les modifications.

Pour observer l’évolution des valeurs de distance en fonction de la granularité, on propose l’observation de $\Delta_{granularite}$, correspondant à l’augmentation de la distance entre 2 granularités successives en Table 3.5 et Figure 3.5.

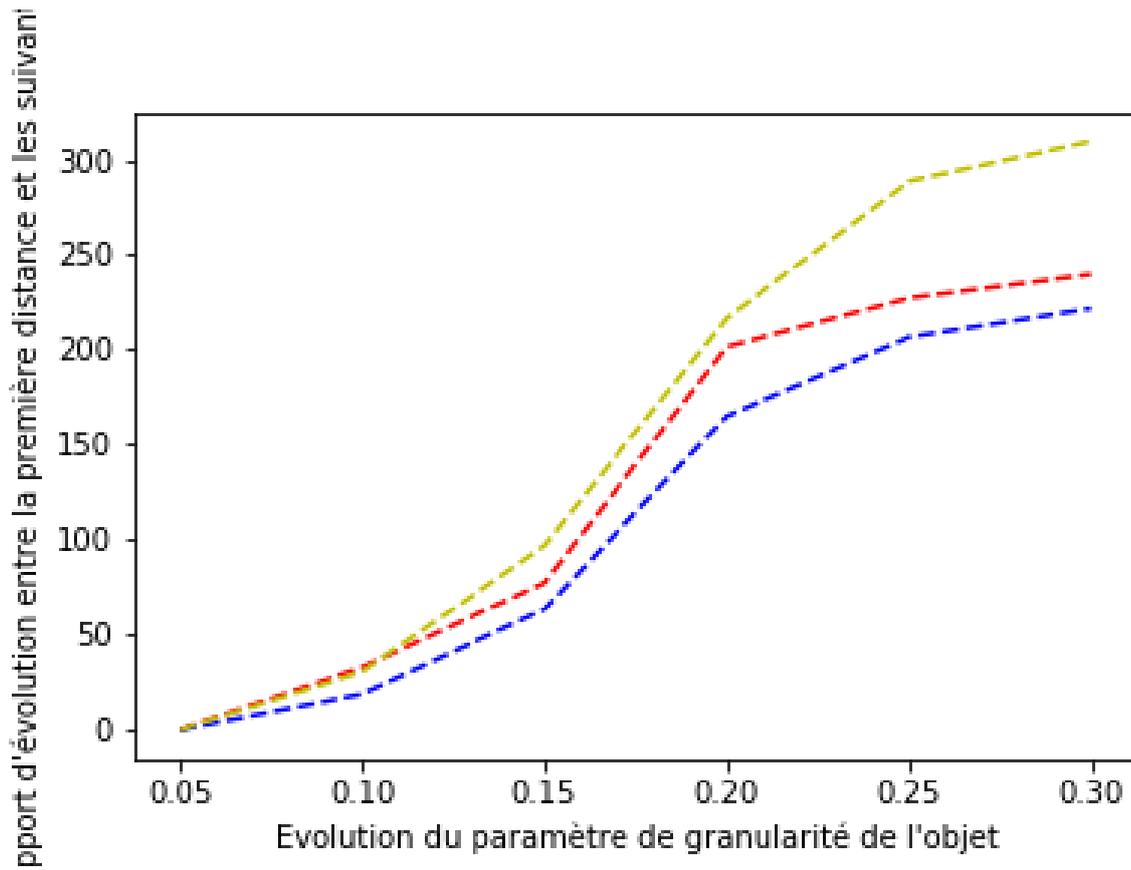


FIGURE 3.5. – Résultats de l'évolution de la distance à l'objet d'origine en fonction de la granularité

Baisse de granularité	Moyenne distances intra-classe	Moyenne distances inter-classe	Rapport de ces deux valeurs
0%	460.17	1665.3	0.27632
5%	342.65	1004.5	0.34109
10%	379.88	1437.8	0.26419
15%	14522.5	5996.4	2.4218
20%	71137140	22301639	3.1897

TABLE 3.6. – Evolution des distances intra et inter classes en fonction de la granularité avec la distance de Kullback-Leibler

Baisse de granularité	Moyenne distances intra-classe	Moyenne distances inter-classe	Rapport de ces deux valeurs
0%	297.92	1111.7	0.26797
5%	220.50	627.16	0.35158
10%	292.47	973.69	0.30037
15%	13288	5111.9	2.5994
20%	71135763	22300531	3.1898

TABLE 3.7. – Evolution des distances intra et inter classe en fonction de la granularité avec la distance Euclidienne

Le descripteur semble relativement robuste. Néanmoins, la distance entre l’objet de base et l’objet nouvellement créé tend à rejoindre celle de deux objets de la même classe pour une augmentation de 30% de la granularité, si l’on en croit les résultats précédemment établis en Table 3.3.

Pour confirmer ou récuser cette affirmation, il est proposé de calculer l’évolution du ratio entre les distances intra-classes et inter-classes en fonction de la granularité, et ce pour chaque distance retenue.

Les Tables 3.6, 3.7 et 3.8, utilisent respectivement les distances de Kullback-Leibler, Euclidienne et de Matusita.

Elles permettent de mettre en lumière un point : passés les 10% de points retirés, il s’avère que la distance moyenne inter-classe devient plus petite que la distance moyenne intra-classe. Résultat étonnant, qui ne pourrait être expliqué que par le fait que des erreurs sont introduites lors du retrait de points, distances ayant toutes tendance à exploser.

Bilan :

Baisse de granularité	Moyenne distances intra-classe	Moyenne distances inter-classe	Rapport de ces deux valeurs
0%	328.94	1489.6	0.22082
5%	311.14	912.23	0.34107
10%	398.44	1319.3	0.30200
15%	14340	5806.3	2.4698
20%	71102149	22293150	3.1894

TABLE 3.8. – Evolution des distances intra et inter classe en fonction de la granularité avec la distance de Matusita

Suite aux différentes expérimentations réalisées, le descripteur s'avère être suffisamment bon pour décrire des objets par maillage, tout en conservant la souplesse nécessaire à leur comparaison.

3.2.3. Choix du nombre d'extraits

En réutilisant le descripteur *HKS* choisi précédemment, on obtient, par validation croisée en quatre parties un jeu d'apprentissage de 40 et 100 objets, de deux classes de mêmes tailles mais tirées aléatoirement du *PSB*, les résultats donnés dans la Table 3.9.

Nombre d'objets 3D	Nombre de sous-parties extraites (par objet)	Taux de bonne classification	Temps de calcul (secondes)
40	25	0.65 ± 0.15	122
	50	0.75 ± 0.16	237
	100	0.81 ± 0.14	468
	200	0.87 ± 0.10	831
	300	0.89 ± 0.11	1469
	400	0.92 ± 0.08	2160
	<i>HKS</i> global	0.85 ± 0.08	1020
100	25	0.59 ± 0.28	230
	50	0.71 ± 0.29	396
	100	0.78 ± 0.18	902
	200	0.87 ± 0.16	1728
	300	0.90 ± 0.14	2645
	400	0.91 ± 0.13	3587
	<i>HKS</i> global	0.82 ± 0.11	2285

TABLE 3.9. – Performances comparées : précision pour 40 et 100 objets

La première colonne de la Table 3.9 indique le nombre de sous-surfaces extraites et les tailles de ces différents extraits. Par exemple, pour la première ligne, 40 sous-surfaces ont été extraites, avec des tailles de 20, 40, 60 et 80 points reliés.

Comme pour la comparaison de descripteurs, les taux de bonne classification de la 2ème colonne sont complétés par l'écart-type de la validation croisée.

La dernière ligne présente les résultats d'une méthode de classification globale, en utilisant les histogrammes des *HKS* de l'ensemble des points comme attributs.

A première vue, et cela paraît logique, on remarque que plus les tailles d'extraits sont variées, et plus le nombre de ces extraits augmente, meilleure est la classification. La confiance dans les prévisions est assez limitée, cela est dû à la sélection aléatoire des sous-parties qui rend la méthode légèrement moins stable qu'une déterministe. Cette perte de stabilité des performances est moins importante que celle causée par la validation croisée. il convient

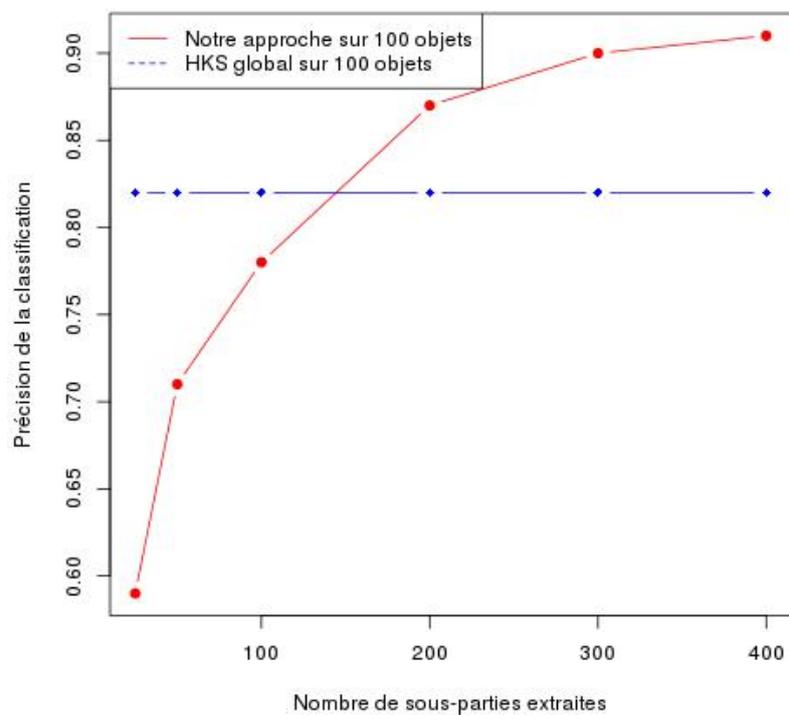
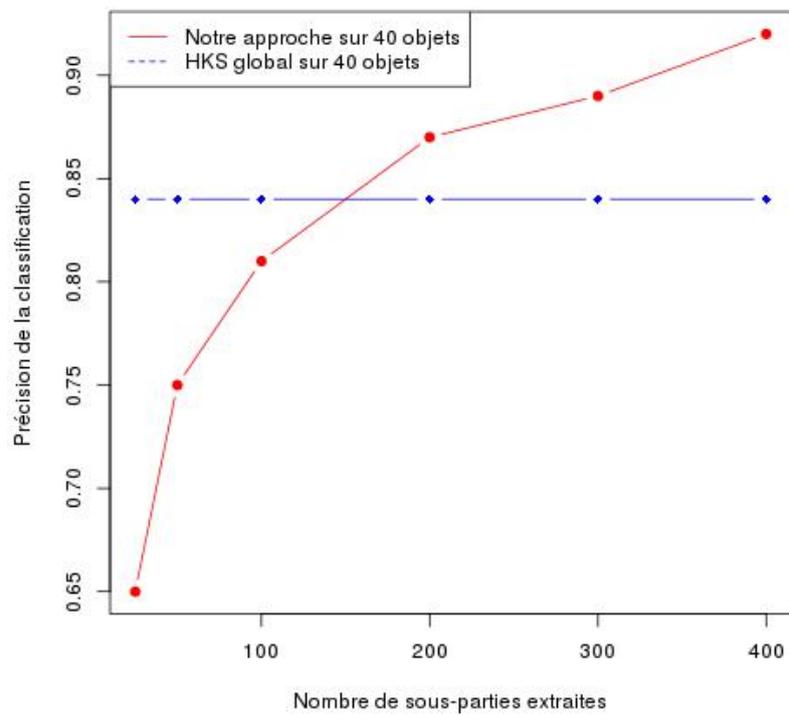


FIGURE 3.6. – Comparaison de taux de bonne classification en fonction du nombre d'extraits

3.3. Comparaison avec d'autres méthodes de classification supervisée d'objets 3D

néanmoins de préciser que cette observation est valable pour nos données et pour la quantité de données que l'on a souhaité sélectionner. Utiliser un plus grand jeu d'apprentissage réduirait sans aucun doute la portée de cet inconvénient, chose que des études ultérieures pourraient traiter.

Pourtant, la méthode proposée est plus rapide que la globale, cela venant du fait qu'elle ne considère que certaines parties de l'objet.

Qui plus est, la comparaison peut être reproduite avec n'importe quel descripteur, dans la mesure où une technique visant à définir un objet 3D peut être adaptée à notre contexte. En effet, celle-ci est utilisable sur n'importe quel descripteur d'objet 3D.

3.3. Comparaison avec d'autres méthodes de classification supervisée d'objets 3D

A titre indicatif, voici un comparatif des méthodes les plus utilisées avec la nôtre, bien que les techniques actuelles ne permettent pas l'explication du résultat à l'utilisateur (Table 3.10).

Cette fois-ci, les tests sont réalisés avec 60 objets de deux classes de mêmes tailles du *PSB*.

On a choisi d'utiliser 200 extraits par objet, ce qui était un bon compromis entre le temps d'exécution et la précision de la classification, si l'on se réfère à la Figure 3.6.

Méthode de classification d'objets 3D	Précision de la classification	Temps de calcul moyen (secondes)
Harmoniques sphériques	0.71 ± 0.09	1205
Shape Histogram [Ankerst et al., 1999]	0.74 ± 0.07	952
Extended Gaussian image [Horn, 1984]	0.83 ± 0.10	1452
Gaussian Euclidean Distance Transform [Kazhdan et al., 2003]	0.88 ± 0.08	1356
Hough 3D et SURF [Knopp et al., 2010]	0.87 ± 0.06	1325
USC global	0.83 ± 0.08	1244
PFH global	0.78 ± 0.10	1546
HKS global	0.84 ± 0.07	1253
Notre approche	0.89 ± 0.12	1152

TABLE 3.10. – Performances comparées entre méthodes de classification supervisée d'objets 3D

On remarque que notre méthode obtient des résultats globalement équivalents à ceux de la littérature, pour un temps de calcul du même ordre de grandeur, bien qu'il ne s'agisse que d'une classification simple (2 classes) sur un nombre restreint d'objets 3D (60).

3.4. Interprétation des sous-parties discriminantes

Le principal apport de notre méthode, dans la mesure où ses résultats sont globalement équivalents à ceux de la littérature, est de pouvoir permettre la compréhension de la classification proposée par l'exhibition des sous-parties qui ont été choisies comme étant les plus pertinentes. Dans notre exemple, il s'agissait de classer des objets 3D entre 2 catégories, qui

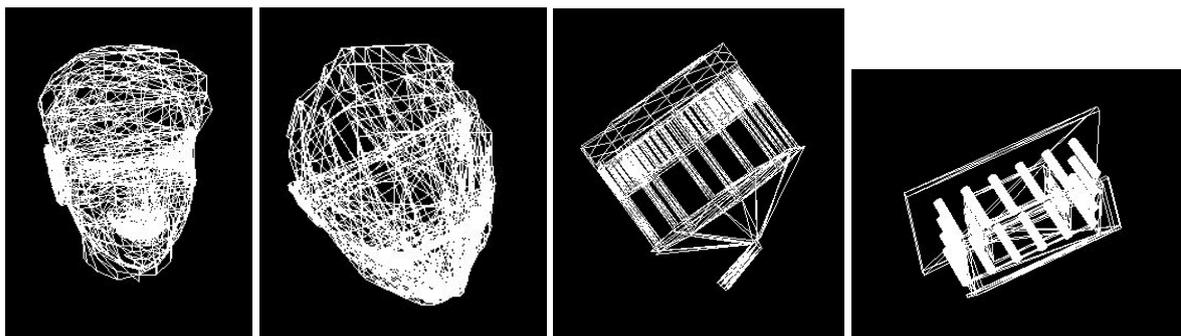


FIGURE 3.7. – Objets 3D à classifier : les 2 premiers sont de la 1^{ère} classe (partie du corps humain), et les 2 derniers de l’autre classe (bâtiments)

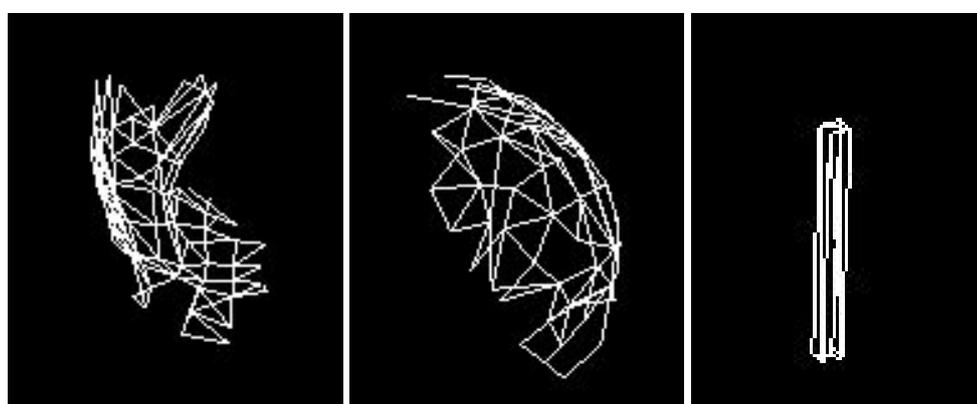


FIGURE 3.8. – Extraits discriminants d’objets 3D

étaient d’un côté des objets inanimés et de l’autre des parties (tête ou buste) du corps humain (Figure 3.7).

On récupère des extraits tels que ceux de la Figure 3.8, qui correspondent pour les deux premiers à de l’anatomie humaine (probablement la partie courbée de l’arrière d’une tête) et le dernier à une partie spécifique aux bâtiments présentés (certainement une colonne). Ces éléments permettent d’exhiber certaines des sous-parties ayant permis de créer les attributs de l’objet, et donc les extraits qui sont les plus représentatifs de la classification.

Néanmoins, il s’agit d’une face de la méthode qui va surtout dépendre de l’application souhaitée. Au final, l’interprétabilité mise en avant est liée à la proposition par l’algorithme des sous-parties récupérées qui peuvent être utilisées pour expliquer la classification.

3.5. Sélection non supervisée de sous-parties

Les tests de calcul multiples sont présentés dans le tableau 3.11. Nous présentons maintenant des résultats comparatifs avec la méthode originale, sur un ensemble de données plus difficiles à classer (la résolution / précision de chaque classe était plus précise dans ce cas).

Nous conservons également les 200 extraits par objet et les autres paramètres trouvés dans les expériences précédentes. La granularité de classe est plus petite cette fois, d'où des valeurs de résultats tout à fait différentes de celles des expériences précédentes. Bien que cela ressemble à une classification semi-supervisée, nous n'utilisons que le concept de non supervisé pour la sélection des fonctionnalités, et non pour la partie apprentissage, qui reste une classification supervisée traditionnelle.

Utilisation de la sélection non supervisée	Nombre d'objets dans le jeu de données		
	20	40	60
Non	0.62 ± 0.12	0.66 ± 0.11	0.71 ± 0.09
Oui	0.62 ± 0.14	0.64 ± 0.13	0.68 ± 0.11

TABLE 3.11. – Performances comparées : utilisation du non supervisé pour la sélection d'attributs

Les résultats ne permettent pas d'affirmer que notre façon non supervisée de sélectionner ces sous-parties est suffisamment bénéfique pour être justifiée. Cependant, il existe un moyen de rendre cette idée de sélection non supervisée utilisable (bien que, dans ce cas, elle ne dépende pas directement de la classification en elle-même). La comparaison, comme précédemment, a été faite avec le même nombre d'extraits.

Jusqu'à présent, afin de calculer la notion de degré d'une sous-partie à un objet, nous avons utilisé la formule expliquée dans l'équation 1. Cette notion était l'idée originale [Ye and Keogh, 2009] pour rechercher l'extrait le plus proche de l'objet. Ici, nous proposons une alternative à cette formule en choisissant, non pas le minimum, mais une fonction pondérée particulière qui considère tout l'extrait de l'objet. Cette formule, utilisée en logique floue, s'appelle *Ordered Weighted Averaging* [Yager, 1988] (OWA), et remplace le minimum d'origine des distances entre les sous-parties étudiées et chaque sous-partie de l'objet actuel. Appliqué à notre problème, cette fonction est définie comme suit :

$$OWA(a_1, \dots, a_n) = \frac{\sum_{j=1}^n w_j b_j}{\sum_{j=1}^n w_j} \quad (3.1)$$

avec $(a_1, \dots, a_n) \in \mathbb{R}^n$, b_j le j^{eme} des a_i ordonnés et $(w_1, \dots, w_n) \in [0, 1]^n$ les poids donnés.

Le résultat est normalisé, afin d'être utilisé pour la comparaison, même si le nombre d'éléments n est différent. Les poids doivent également être définis. Nous proposons de classer toutes les distances trouvées à partir du calcul original du *Degree*, et de leur donner un poids correspondant à leur classement. Ce poids doit diminuer afin de mieux considérer les sous-parties les plus proches au détriment des plus éloignées. Certains de ces poids sont proposés pour le calcul : avec $j = 1, \dots, n$, les poids testés sont $\frac{1}{\log(j)}$, $\frac{1}{j}$, $\frac{1}{j^2}$, $\frac{1}{j^3}$ et $\frac{1}{e^j}$ (appelés OWA-1, ..., OWA-5).

Dans le tableau 3.12, nous présentons les résultats en utilisant les différents poids possibles déjà mentionnés pour OWA.

TABLE 3.12. – Performances comparées : non supervisé avec OWA

Degré utilisé	Nombre d'objets dans le jeu de données		
	20	40	60
Fonction "min" classique	0.62 ± 0.14	0.64 ± 0.13	0.68 ± 0.11
OWA-1	0.64 ± 0.12	0.67 ± 0.15	0.71 ± 0.11
OWA-2	0.62 ± 0.11	0.64 ± 0.15	0.68 ± 0.17
OWA-3	0.68 ± 0.13	0.72 ± 0.11	0.75 ± 0.11
OWA-4	0.65 ± 0.12	0.66 ± 0.11	0.70 ± 0.10
OWA-5	0.63 ± 0.11	0.65 ± 0.10	0.68 ± 0.17

La meilleure combinaison trouvée semble être l'utilisation d'une sélection non supervisée d'attributs avec un OWA degré de $\frac{1}{j^2}$ (OWA-3 dans le Tableau 3.12). Le point intéressant à mettre en évidence est le fait que le non supervisé est utile avec le degré OWA, mais pas avec le classique *min* utilisé initialement.

Cette combinaison finale de différentes modifications de la méthode d'origine donne de meilleurs résultats et confirme notre approche initiale. Ceci en montrant la contribution que représente la sélection non supervisée de certaines des sous-parties pertinentes pour la construction de caractéristiques (attributs).

3.6. Discussion

Les résultats obtenus lors de la comparaison à d'autres techniques ne correspondent qu'à une partie du véritable objectif de cet algorithme (exhiber à l'utilisateur des sous-parties discriminantes pour l'aider à la compréhension de la classification). Pourtant, la méthode que nous proposons s'avère au moins aussi bonne (voir légèrement meilleure) que les méthodes plus traditionnelles, bien que l'aléatoire semble en diminuer légèrement la stabilité. Les premières expériences de visualisation sont prometteuses, mais nécessitent pour être confirmées une mise en pré-production au sein du métier ciblé, afin de se confronter aux besoins des géologues.

Par ailleurs, la sélection aléatoire a été choisie par analogie avec les random shapelets car elle permet de réduire le temps de calcul (et donc la complexité de la recherche) de façon très intéressante tout en fournissant des résultats analogues. Qui plus est, la méthode s'est montrée stable en termes de performances, l'incertitude qu'implique l'aléatoire étant plus faible que celle observée par la validation croisée.

Il convient néanmoins de préciser que certains choix, comme celui de séparer la sélection des sous-parties pertinentes de l'apprentissage pur effectué ensuite, sont inhérents à la méthode proposée, qui consistait à adapter un algorithme issu de la classification supervisée de séries temporelles. Il serait tout à fait possible, dans le cadre de recherches ultérieures, de chercher

à appliquer d'autres concepts tels que le bagging.

3.6.1. Amélioration de la classification

Poursuivre sur cet axe de recherche implique de remettre en cause l'ensemble des valeurs données aux paramètres. Le choix de certains paramètres de l'algorithme, c'est-à-dire :

1. le nombre de sous-parties par objet ;
2. la taille de ces sous-parties ;
3. le nombre de sous-parties retenues pour l'apprentissage ;

n'ont été choisis que par tests empiriques successifs sur différents ensembles de données, et en accordant notre confiance à des publications antérieures.

A terme, il s'agira bien entendu de chercher à automatiser ces choix de paramètres. Il peut s'agir ici de la connaissance initiale nécessaire à la classification, ou bien d'un pré-traitement automatique d'estimation de ces paramètres.

3.6.2. Perspectives

D'autres concepts tels que les nuages de points récupérés lors du scan d'une pièce pourraient se voir traités par ce principe, le maillage n'étant utile que dans le cas du descripteur choisi (*HKS*) pour nos expériences.

Qui plus est, et afin de pouvoir adapter ce modèle à des situations plus courantes de classification de structures 3D, il est opportun de chercher à l'adapter à des cas d'objets partiellement occultés. Cela permettrait ainsi de l'utiliser également en vision par ordinateur.

Enfin, des utilisations en temps réel des idées proposées dans cette première partie de thèse pourraient être envisagées.

4. Application aux données Total : adaptation de l'algorithme 3DRESC aux nuages de points

Dans ce chapitre, les tests sont réalisés sur des données propres à Total, afin d'en vérifier l'adaptabilité et la performance. A cette fin, l'algorithme 3DRESC est légèrement modifié, afin de tenir compte du type de données différent auquel nous sommes confrontés.

Ainsi, on expérimente les propositions des chapitres antérieurs de classification supervisée, en particulier l'algorithme 3DRESC, sur les données de l'entreprise TOTAL. Pour des raisons de confidentialité, il n'a pas été possible d'accéder aux données des géo-modèles réels et opérationnels de TOTAL, représentant les failles et changements de couches dans les sous-sols explorés. Par contre, nous avons pu accéder à d'autres données mais sous une autre forme.

Lors du survol de certaines installations par des méthodes de recueil de données comme le LIDAR¹, des nuages de points sont récupérés et utilisables sans contrainte de confidentialité. Ils correspondent à une représentation 3D d'une zone qui permet de détecter automatiquement un phénomène particulier. Cette masse de données est utilisée par exemple dans la prévention ultérieure d'accidents.

Elle est composée de millions de points mais ne dispose pas d'arêtes, ce qui ne permet pas de maillages triangulaires, contrairement au pré-requis de l'algorithme 3DRESC. En effet, le principe du LIDAR étant de calculer la position d'une multitude de points par rapport à l'émetteur, il n'existe pas de relation directe entre ces différents points.

La structure de données P est donc un nuage de points classique défini par :

$$P = \{p_i | p_i = \{x_i, y_i, z_i\}, i \in [1, n]\} \quad (4.1)$$

Ces données sont présentées sous le format *PTX*, format d'enregistrement adapté au stockage de nuages de points, et composé pour la majorité d'une succession de coordonnées de points en 3 dimensions.

En cherchant à appliquer la méthode 3DRESC, on fait face à deux difficultés qui vont nous obliger à reconsidérer deux points essentiels de l'algorithme d'origine :

1. le type de données n'est pas le même : on avait affaire à des maillages triangulaires 3D dans le précédent cas, des nuages de points dans celui-ci ;
2. les données très nombreuses entraînent un temps de calcul bien trop important pour être traitées sur un ordinateur individuel, condition nécessaire à la création d'un pro-

1. Le LIDAR est technique de mesure de distance basée sur un faisceau de lumière renvoyé vers son émetteur.

totype ; par ailleurs, le besoin en mémoire excède les capacités de mémoire d'une machine individuelle classique.

On précise dès maintenant que la création des classes, pour notre exemple, est réalisée de façon arbitraire par la présence ou absence de la formation circulaire censée correspondre à l'entrée du puits sur la plate-forme.

L'objectif reste une classification supervisée, dont les performances sont évaluées par validation croisée et AUC.

4.1. Modifications dues au type de données

Si l'on souhaite pouvoir utiliser 3DRESC sur des données non définies par un maillage, des étapes sont à modifier, en particulier :

1. l'extraction de sous-parties à partir d'un point sélectionné aléatoirement ;
2. le calcul de distances entre sous-parties.

La première étape soulevée ici impose le choix d'une autre technique permettant de sélectionner le voisinage d'un point (toujours choisi aléatoirement) dans le but de créer une sous-partie. La notion de proximité entre deux points, symbolisée par une arête les reliant, peut être remplacée par un système de simple distance Euclidienne. La zone de sélection autour du point considéré devient une sphère, celle-ci limitée en fonction du nombre de points souhaité.

La seconde étape avait jusqu'alors été traitée en calculant, dans un premier temps, un descripteur de sous-parties, puis dans un second temps une distance entre répartitions de probabilités pour calculer une "proximité" (l'inégalité triangulaire n'étant alors pas forcément vérifiée) entre ces sous-parties.

Face à des nuages de points, il devient possible et nécessaire de considérer un autre paramétrage pour les deux points suivants :

- le premier est le choix de la mesure de similarité entre deux nuages de points en fonction de ses performances, d'où une remise en cause nécessaire ;
- le second est celui du descripteur lui-même, si l'on suppose la structure initiale du couple descripteur/distance conservée : certains descripteurs, comme le HKS, ne traitent que des structures de points reliés (graphes ou maillages), contrairement à d'autres. Le Unique Shape Context (USC) [Tombari et al., 2010], par exemple, est moins pertinent pour les maillages (et moins performant en utilisant 3DRESC) car il ne prend en compte que des points isolés (non reliés) ; il possède néanmoins l'avantage de pouvoir être utilisé dans notre cas d'application.

Ainsi, seules les données et expérimentations sur celles-ci vont permettre de trancher entre les différentes possibilités envisageables.

4.2. Quantité de données : un frein à l'application directe de l'algorithme

Afin de mieux appréhender le problème que constitue l'important volume de données, il est proposé d'évaluer le temps que mettrait, sur une machine individuelle classique, l'exé-

4.2. Quantité de données : un frein à l'application directe de l'algorithme

cution de l'algorithme 3DRESC sur une dizaine de nuages composés chacun de 50 millions de points.

Pour rappel, les différentes étapes de l'algorithme 3DRESC sont :

1. pour chaque nuage de points (dont on fixe à N le nombre) : l'extraction aléatoire de S sous-parties de taille équivalente à $\frac{1}{t}$ de la taille totale du nuage de points (noté T) :

$$\begin{aligned} O(\text{nombreExtraitParObjet} * \text{tailleChaqueExtrait} * N) \\ = O(S * \frac{T}{t} * N) \end{aligned}$$

2. pour chaque extrait ou sous-partie, on calcule un descripteur (dont le calcul se fait en $O(n)$) permettant de diminuer la taille de l'espace de description de ces dernières, donc la complexité :

$$O(\text{nombreExtraitParObjet} * N * \text{tailleChaqueExtrait}) = O(S * N * \frac{T}{t})$$

3. les sous-parties sont comparées 2 à 2 pour le calcul ultérieur des degrés d'appariement entre éléments et sous-parties, soit :

$$O((S * N)^2)$$

4. chaque sous-partie est évaluée en se basant sur la matrice de distances créée :

$$O(S * N)$$

5. les k meilleures sous-parties selon le critère précédemment utilisé sont récupérées et utilisées pour l'apprentissage, ce qui donne :

$$O(k * S * N)$$

6. pour la prévision sur des nuages de points inconnus, il était nécessaire de recalculer les distances entre chaque sous-partie sélectionnée (les k précédentes) et les S extraites du nouveau nuage de points, soit :

$$O(k * S * \frac{T}{t} * N)$$

Il est proposé en Table 4.1 une comparaison entre l'application de 3DRESC aux différents types de données, à savoir les objets 3D qui ont servi aux expérimentations de la méthode originale, et les nuages de points fournis par Total.

On remarque aisément qu'il existe, en termes de temps de calcul, un facteur de près de $2 * 10^2$ entre l'étude de ces deux types de données. Cela est dû aux tailles très différentes des entités en présence (toutes choses étant supposées égales par ailleurs).

Étapes	Ordre de grandeur	Avec objets 3D du PSB	Avec nuages de points
1	$O(S * \frac{T}{t} * N)$	$4 * 10^6$	$4 * 10^8$
2	$O(S * N * \frac{T}{t})$	$5 * 10^9$	$1 * 10^{12}$
3	$O((S * N)^2)$	$4 * 10^8$	$4 * 10^4$
4	$O(S * N)$	$2 * 10^4$	$2 * 10^0$
5	$O(k * S * N)$	$8 * 10^5$	$8 * 10^1$
6	$O(k * S * \frac{T}{t} * N)$	$1.6 * 10^8$	$1.6 * 10^{11}$
Ordre de grandeur de la somme globale	$\sum_{1 \leq i \leq 6} \text{Étapes}_i$	$\approx 5 * 10^9$	$\approx 10^{12}$

TABLE 4.1. – Temps de calcul en nombre approximatif d'opérations

Lors de précédentes expériences visant à confirmer la pertinence de l'algorithme 3DRESC, le temps d'apprentissage avait été mesuré aux alentours de 1200 secondes. Cela restait alors tout à fait convenable. Si de tels calculs, en raison du type de données, voyaient leur ordre de grandeur augmenté d'un facteur $2 * 10^2$, cela deviendrait littéralement impossible en un temps raisonnable : $1200 * 2 * 10^2 = 2.4 * 10^5$ secondes, soit plusieurs jours.

Appliquer cette même méthode n'est plus envisageable, et une adaptation visant à contourner le problème s'impose naturellement.

A cette fin, il est proposé d'optimiser le temps de calcul et le besoin en mémoire.

4.3. Une optimisation en calcul et en mémoire

L'étape que l'on cherche à réduire est celle de l'extraction des 200 sous-parties pour chaque entité (objet puis nuage de points), dont la taille rend le traitement très long.

Cet axe d'optimisation se base sur le fait que le nuage dispose d'un très grand nombre de points, suffisamment proches pour qu'une grande partie soient supposés redondants. Il est donc proposé de tenter de ne tenir compte que d'une partie des points. Pour les besoins de nos tests sur machine locale, nous proposons de n'utiliser, dans un premier temps, qu'un point sur 100, ces derniers étant classés par proximité dans le fichier source. A cette fin, et en raison du calcul de descripteur, un gain de 10^2 en termes de temps d'exécution est immédiatement obtenu.

Pour néanmoins justifier de la pertinence d'une telle mesure, potentiellement cause d'une perte d'information et donc de performances, il est également proposé, pour l'un des nuages de points, de comparer les descripteurs des sous-parties avec des échantillonnages différents.

Par ce biais, la mémoire n'est alors plus un problème, une réduction par un facteur 100 de cette dernière ayant été appliquée.

4.4. Des nuages de points multiples pour un apprentissage supervisé - application de la méthode 3DRESC modifiée aux données de Total

Une fois la première étape d'exploration, de présélection et de traitement de sous-parties terminée, il convient tout naturellement d'appliquer la méthode principale proposée au cours de cette thèse, à savoir 3DRESC, aux nuages de points introduits dans cette partie.

On dispose d'une dizaine de nuages de points de tailles sensiblement équivalentes. Chacun est annoté pour la présence ou l'absence de certains phénomènes.

Les principales étapes de cette méthode, sur une machine individuelle et pour la partie d'apprentissage, sont :

1. Pour chaque nuage de points séparément :
 - extraction de sous-parties (200, chacun de taille $\frac{1}{30}$ ^{ème} du nuage complet) en ne considérant qu'un point sur 100 du nuage de points initial ;
 - pour chaque sous-partie, calcul du descripteur adapté ; le Unique Shape Context est ici préféré, ce dernier pouvant traiter des points seuls, sans arête les reliant, et est le deuxième dans l'ordre des descripteurs offrant les meilleures performances ;
2. Calcul des distances entre sous-parties ;
3. Calcul des degrés de chaque sous-partie par la méthode du gain d'information ($f - stat$ également possible) ;
4. Les n premiers (entre 40 et 80 semble un bon compromis - cf **Chapitre 2**) selon le critère précédemment utilisé sont les sous-parties sélectionnées pour servir de dimensions de projection aux nuages de points, créant ainsi un espace propice à la réalisation de l'objectif souhaité.

Le point 1 est schématisé en Figure 4.1.

Les points 3 et 4 sont décrits succinctement par les Figures 4.2 et 4.3 (le point 2 n'étant pas schématisé car évident).

En agissant de la même façon que pour les objets 3D constitués d'un maillage, on obtient les résultats de la Table 4.2.

Les résultats obtenus permettent de mettre en lumière deux points essentiels :

- un échantillonnage de $\frac{1}{100}$ est un bon compromis entre le temps de calcul et la performance de prévision ;
- la méthode reste efficace sur des données pour lesquelles elle n'avait pas forcément été conçue ; une telle capacité d'adaptation du principe initialement proposé confirme la pertinence de l'approche de base.

4.5. Bilan

Nous avons finalement appliqué l'algorithme 3DRESC à un ensemble de données réelles fourni par Total. A cette fin, il a été nécessaire de réaliser certaines adaptations des données et de l'algorithme afin de pouvoir l'appliquer à des nuages de points de grande taille.

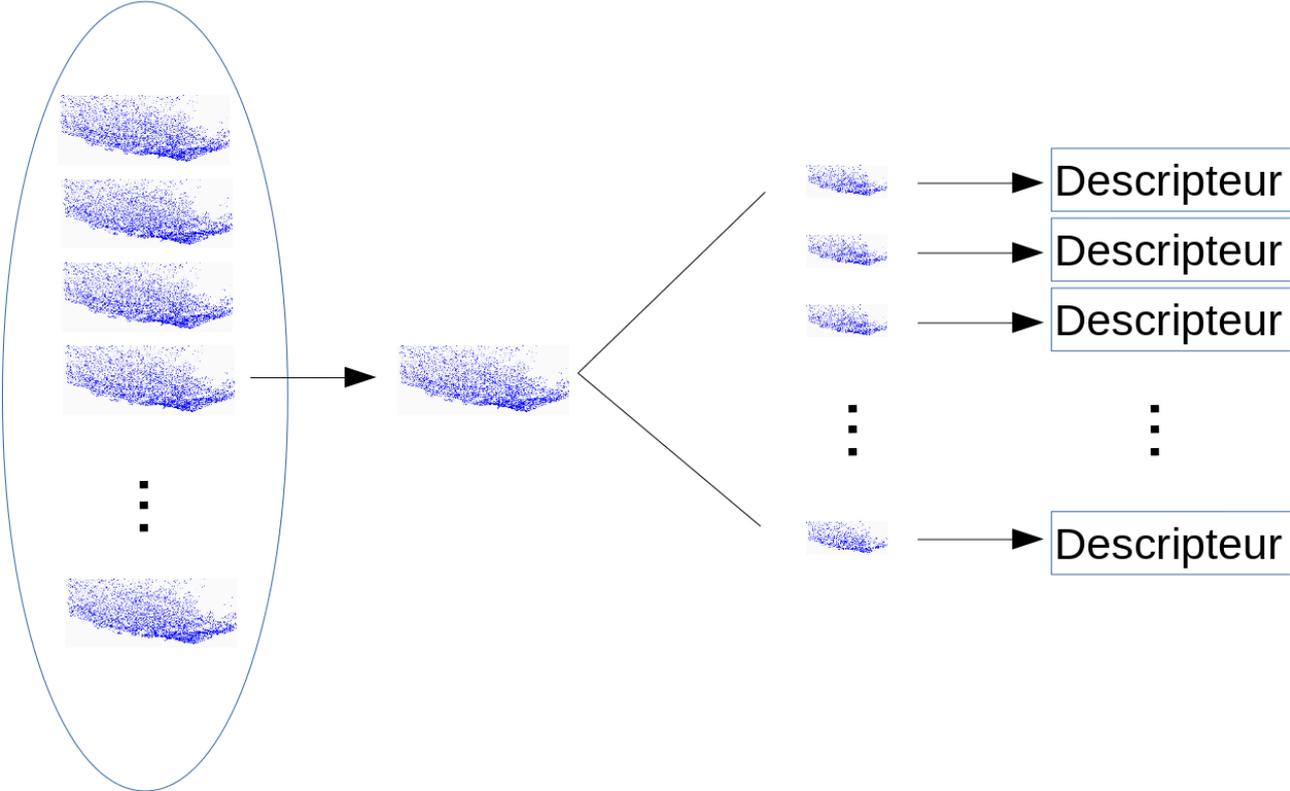


FIGURE 4.1. – Extraction de sous-parties et calcul des descripteurs

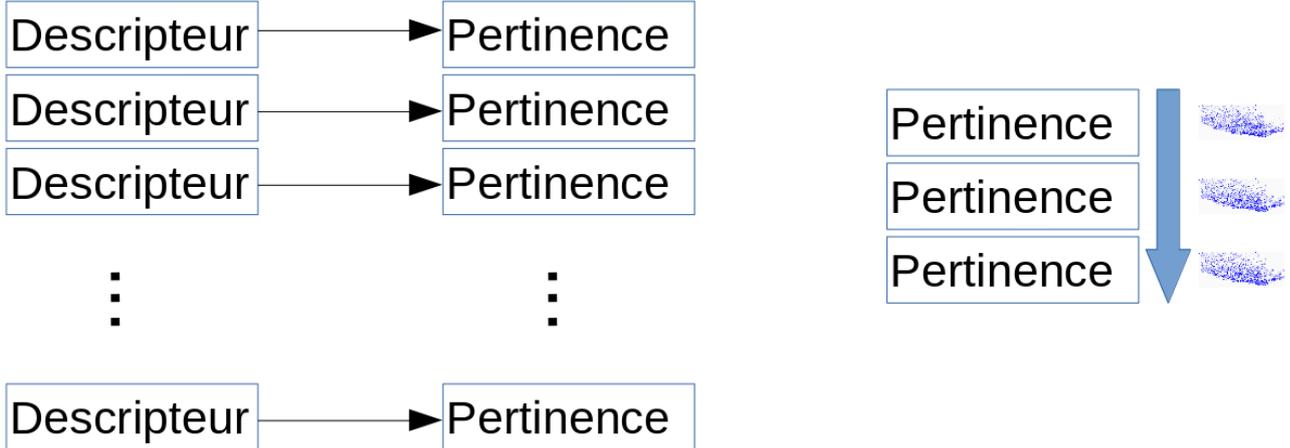


FIGURE 4.2. – Calcul de la pertinence de chaque sous-partie, et sélection des meilleures

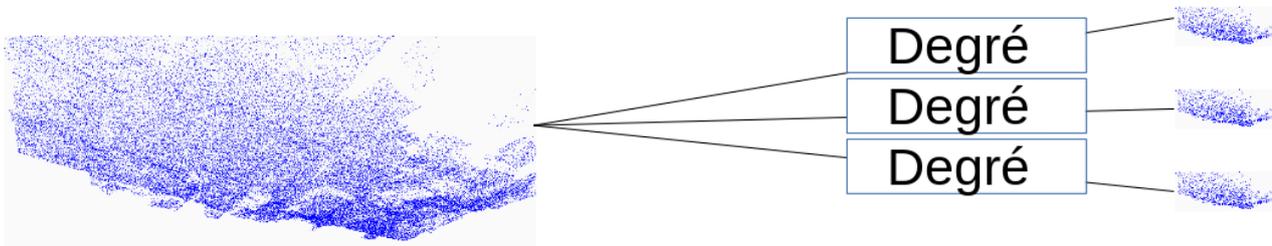


FIGURE 4.3. – Calcul de caractéristiques d’un nouveau nuage de points à partir des sous-parties précédemment extraites et sélectionnées

Échantillonnage	AUC (Area Under Curve ROC)	Temps de calcul (secondes)
$\frac{1}{10}$	0.88 ± 0.09	10^{11}
$\frac{1}{50}$	0.86 ± 0.09	2.10^{10}
$\frac{1}{100}$	0.83 ± 0.11	10^{10}
$\frac{1}{200}$	0.76 ± 0.25	5.10^9

TABLE 4.2. – Résultats obtenus par validation croisée sur 10 nuages de points. L’échantillonnage correspond à la proportion de points conservés, points sélectionnés aléatoirement.

Application aux données Total : adaptation de l'algorithme 3DRESC aux nuages de points

L'intérêt de cette étude permet de mettre en avant la capacité de l'algorithme 3DRESC de s'adapter à un tel type de données tout en permettant de conserver des performances appréciables.

5. Conclusion de la partie I

Nous avons, dans ce chapitre, cerné et résolu pour un cas particulier d'application, le problème que l'on rencontre généralement en apprentissage automatique : la compréhension par l'utilisateur du résultat en ayant à disposition les moyens utilisés pour atteindre ce dernier. En effet, les "boîtes noires" que sont généralement les techniques d'apprentissage, sont un véritable frein à l'utilisation de ces méthodes par les professionnels. Cela est d'autant plus un problème d'actualité, que les méthodes actuelles de prévision ont atteint un niveau de maturité qui leur permet, dans un cadre productif, de remplacer les experts pour certaines tâches particulières.

Notre problème, initialement rattaché à de la recommandation d'information pertinente pour le géologue, a finalement pu être ramené à une classification plus générale de structures 3D (des géo-modèles du sous-sol utilisés en prospection pétrolière).

De plus, il nous a fallu contourner le problème généralement rencontré en classification d'objets 3D, qui est celui de la sélection de caractéristiques (*features*) pertinents, ces derniers sont supposés être, dans notre cas d'application, des sous-parties de ces objets.

Pour cela, nous nous sommes inspirés des techniques de classification employées pour des types de données différents, que sont les images et les séries temporelles.

Enfin, des expérimentations sur des données réelles de Total ont été réalisées, avec adaptation de l'algorithme aux nuages de points, ce qui tend à prouver son adaptabilité.

Une nouvelle méthode de classification d'objets 3D, avec justification de cette dernière par mise en lumière des éléments les plus discriminants, a donc été proposée. La notion de "shapets", que [Ye and Keogh, 2009] applique aux séries temporelles, a été prouvée comme étant adaptable aux domaines des objets 3D.

A la croisée des techniques employées pour les séries temporelles et le traitement d'images, notre méthode vise à aider le géologue à mieux comprendre les zones clés pour pouvoir valider la classification proposée sur un ensemble géologique.

Les résultats obtenus lors des expérimentations permettent de mettre en avant l'intérêt de la création d'attributs basée sur la sélection aléatoire pour la classification d'objets 3D, et encouragent logiquement à approfondir cette idée. En effet, de ces multiples tests apparaît une "fenêtre d'ouverture", basée sur le nombre de sous-parties extraites par objet, au sein de laquelle la méthode proposée est à la fois plus rapide et plus précise (en terme de bonne classification) que les méthodes de l'état de l'art.

Pourtant, certains paramètres devront être adaptés aux données rencontrées, paramètres responsables des bonnes performances de l'algorithme *3DRESC*.

A terme, l'objectif est de mettre en place un modèle apprenant "en permanence", ce en se basant sur les classifications antérieures, tout en conservant un des points forts précédem-

Conclusion de la partie I

ment explicités qui était la compréhension par l'utilisateur de la méthode. Cette compréhension, aujourd'hui principal obstacle à l'adaptation à grande échelle des techniques élaborées en intelligence artificielle, est pour ce domaine un des plus gros défis des années à venir [Bouillot et al., 2011]. Par ailleurs, remettre en cause le gain d'information, ou de façon plus large les techniques habituelles de sélection d'attributs pertinents pour l'apprentissage automatique, paraît opportun.

Deuxième partie

Les sous-parties, supports de connaissance

Préambule

Dans cette partie, nous présentons différentes améliorations pour l'algorithme 3DRESC, dans le but de valoriser au mieux les sous-parties d'objets 3D extraites dans le cadre de la classification supervisée. Deux améliorations majeures sont proposées :

1. L'introduction du transfer learning [Pan and Yang, 2010] [Weiss et al., 2016] au sein de la classification de structures 3D afin de rendre le modèle plus adaptable, notamment en sachant réutiliser l'expérience acquise au sein de contextes d'apprentissage différents. Cela peut s'appliquer aisément en géologie, lorsque l'on apprend sur un certain type de bassin géologique, mais qu'un manque de données nous contraint d'en réutiliser certaines pour classifier d'autres types de bassins géologiques.
2. Une nouvelle méthode de sélection de caractéristiques (features) en classification supervisée, basée sur des techniques issues de la logique floue. En effet, le choix de la méthode pour sélectionner les sous-parties extraites est resté celui initialement proposé par [Ye and Keogh, 2009]. Une remise en cause de ce choix, conservé dans la majorité des études sur les time series shapelets qui ont suivi, est une suite toute naturelle.

6. L'utilisation de données provenant de contextes différents

Dans la partie précédente, on a admis le fait que l'on avait affaire à des données appartenant à un même ensemble.

Cela implique de traiter les surfaces des géo-modèles comme des entités indépendantes, sans tenir compte de leur appartenance à un géo-modèle qui tendrait à modifier la prédiction, ou même des différentes interactions des surfaces entre elles.

On a alors supposé une constance de la fonction de répartition de probabilité qui subdivise l'espace en différentes classes, ce qui nous permet d'appliquer les principes habituels de la classification supervisée.

Néanmoins, il paraît logique d'avancer le fait que le passage d'un géo-modèle ou d'un type de géomodèle à un autre, peut modifier la fonction de répartition de probabilité de la classification, et donc que de telles modifications doivent être prises en compte lors de l'apprentissage.

A cette fin, il a été décidé d'adapter des techniques de transfert de connaissance ou *transfer learning* à la méthode proposée dans la *Partie 1*. Le *transfer learning* vise à transférer des connaissances d'une ou plusieurs tâches sources vers une tâche cible. Cette tâche est généralement de l'apprentissage automatique. Cet ensemble de méthodes vise implicitement à rendre l'apprentissage toujours plus autonome, en permettant une meilleure adaptabilité entre des apprentissages issus de données, et donc de contextes, différents [Pan and Yang, 2010] [Weiss et al., 2016] [Wei et al., 2018] [Zhang et al., 2018].

6.1. Etat de l'art de transfert de connaissances

Un des postulats de base en apprentissage traditionnel est que les données d'apprentissage et de test sont issues du même domaine, de telle sorte que l'espace d'attributs et les distributions de données soient identiques, ou du moins très proches. Cependant, dans le cas d'applications concrètes, cette assertion ne tient plus. De plus, il arrive parfois que les données d'apprentissage soient compliquées à rassembler. Il devient alors essentiel de mettre au point des algorithmes d'apprentissage efficaces utilisant des données provenant de domaines différents. De là est né le *transfer learning*.

Dans cette partie, nous considérons les articles de [Pan and Yang, 2010] et [Weiss et al., 2016], qui tous deux présentent des états de l'art complets du domaine, et dont les notations utilisées sont admises comme références.

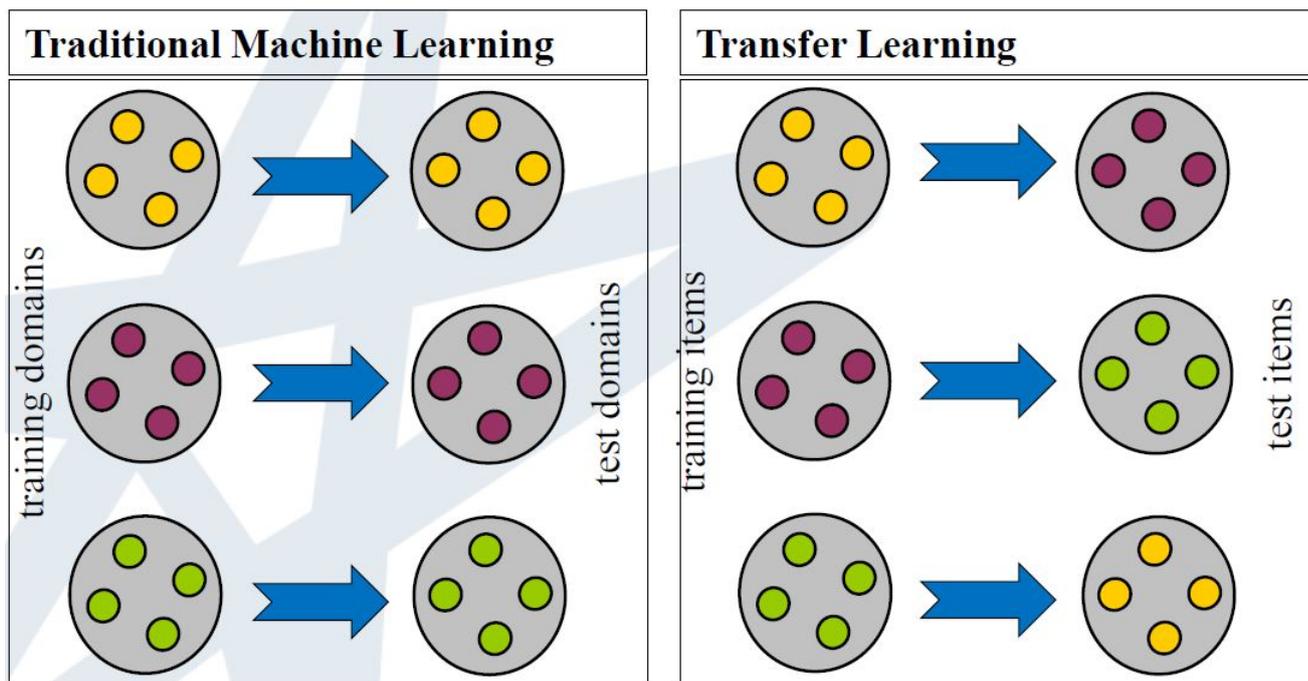


FIGURE 6.1. – Principe du transfer learning [Pan and Yang, 2010], les couleurs représentant des domaines et les flèches le passage entre l'apprentissage à la prédiction

Un domaine D se compose de deux parties, que sont l'espace d'attributs χ et une distribution marginale $P(X)$ (avec $X = \{x_1, x_2, \dots, x_n\} \in \chi$).

En complément, on définit une tâche T par deux composantes, que sont un espace de label Υ ($\Upsilon = \{0, 1\}$ ou $\Upsilon = \{-1, 1\}$ dans le cadre d'une classification binaire) et une fonction de prévision $f(\cdot)$, qui elle-même est apprise à partir des couples (x_i, y_i) dont $x_i \in X$ et $y_i \in \Upsilon$ issus de la base d'apprentissage.

On a donc finalement $D = \{\chi, P(X)\}$ et $T = \{\Upsilon, f(\cdot)\}$.

Dans le cadre du *transfer learning* classique, nous disposons d'un domaine source D_S et d'un domaine cible D_T , ainsi que d'une tâche source $T_S = \{\Upsilon, f_S(\cdot)\}$ et une cible $T_T = \{\Upsilon, f_T(\cdot)\}$. Le but est de chercher à améliorer la fonction de prévision cible f_T en utilisant les informations de D_S et T_S , dans le cas où $D_S \neq D_T$ ou $T_S \neq T_T$. Pour le cas d'un apprentissage supervisé traditionnel, on a $D_S = D_T$ et $T_S = T_T$.

Il existe différentes façons de classer les méthodes de *transfer learning*, bien que celles-ci se recoupent.

6.1.1. Classification en fonction des relations entre la cible et la source

Tout d'abord, il est parfois proposé la division entre le *transfer learning* homogène, pour lequel $\chi_S = \chi_D$, et hétérogène, où $\chi_S \neq \chi_D$ (les espaces d'attributs sont différents).

Transfer learning homogène

Dans cette version du transfer learning, on suppose que $T_S \neq T_T$, sans condition particulière sur les domaines. Il est nécessaire de disposer, au sein du domaine cible, de certains éléments annotés afin d'orienter la fonction prédictive $f_T(\cdot)$. Le domaine source peut être quant à lui annoté ou non. Dans le premier cas, on se réfère à de l'apprentissage multi-tâches (entre source et cible), alors que dans le second, cela correspond à du *self-taught learning* [Raina et al., 2007].

Transfer learning hétérogène

Ici, on dispose de domaines cible et source différents ($D_S \neq D_T$), alors que les tâches sont quant à elles les mêmes ($T_S = T_T$). Deux situations peuvent exister : soit $\chi_S \neq \chi_D$, soit $P(X_S) \neq P(X_T)$.

6.1.2. Classification en fonction des labels disponibles

Le *transfer learning*, en fonction des données en présence, peut aussi se diviser en différents ensembles de méthodes dépendant davantage des données en présence, que sont :

- le *transfer learning* par induction : le domaine cible est annoté, le source pouvant l'être ou non ;
- le *transfer learning* par transduction : le domaine cible n'est pas annoté mais le domaine source l'est ;
- le *transfer learning* non supervisé : le domaine cible et le domaine source ne sont pas annotés.

6.1.3. Classification en fonction des approches

La seconde catégorisation des méthodes de transfer learning correspond aux approches en elles-mêmes (et non plus aux données en présence). Ces 4 principales approches sont les suivantes :

Approche basée sur les instances

On suppose ici qu'il y a suffisamment d'attributs en commun. L'objectif est de chercher à réutiliser directement, après pondération, des données annotées du domaine source au sein du domaine cible.

Intuitivement, l'idée est que, bien que ne pouvant pas réutiliser telles quelles les données du domaine source, certaines peuvent être bénéfiques à la prévision du domaine cible.

En y appliquant une pondération adaptée entre les fonctions d'apprentissage de chaque domaine, et entre les éléments au sein même des ces domaines, un tel objectif devient atteignable.

Les travaux de [Chattopadhyay et al., 2012] en est un exemple, bien qu'adapté à de l'apprentissage par transfert sur des données plus classiques.

Approche basée sur les caractéristiques

Ici, il est nécessaire de ne disposer que de certains attributs en commun entre les domaines. On cherche alors à trouver une représentation qui réduit les différences entre les domaines, en utilisant des caractéristiques (*features*) "pivots" comme bases [Daume, 2007].

Approche basée sur les paramètres

Il s'agit, à partir d'un modèle entraîné sur le domaine source, de pouvoir le réutiliser partiellement au sein du domaine cible [Duan et al., 2012]. En effet, si deux domaines sont liés, alors il est très probable que les modèles de prévision construits partagent certains vecteurs de paramètres.

Approche basée sur les relations

Si deux modèles relationnels sont dépendants, alors ils partagent certaines relations de similarités entre objets. Ce sont ces relations qui peuvent s'avérer utiles pour le transfert de connaissances entre domaines [Li et al., 2012].

6.1.4. Bilan

Notre cas d'application correspond au transfert learning homogène, c'est-à-dire que les espaces d'attributs entre les domaines cible et source sont les mêmes (les modèles de surfaces 3D vu dans la première partie de cette thèse).

Si l'on se place dans la seconde méthode de caractérisation, nous nous plaçons dans le premier cas, l'idée étant de venir compléter la classification au sein d'un géo-modèle par des apports potentiellement utiles issus d'un autre (ou de plusieurs autres). Il est donc nécessaire que le domaine cible possède lui aussi certains de ces éléments déjà annotés.

6.2. Transfer learning pour l'extraction de sous-parties d'objets 3D

Dans le cadre de la classification supervisée basée sur de l'extraction de sous-parties discriminantes, nous cherchons à conserver ce côté aisément interprétable de la méthode proposée initialement.

Dans cette étude, le domaine source est annoté, et le domaine cible l'est pour partie (un minimum étant nécessaire).

Pour rappel, notre méthode *3DRESC* vise à sélectionner des sous-parties pertinentes lors de la phase d'apprentissage. Ces sous-parties permettent de créer de nouvelles dimensions de projection pour les éléments à classifier. De cette manière, il est possible de créer un espace de projection adapté à la tâche en cours. Ces dimensions contiennent en elles-mêmes une partie de l'apprentissage, en orientant la description des éléments dans une direction particulière. Dans le cadre d'une utilisation en transfert learning, réutiliser ces dimensions sur un autre apprentissage que celui dont elles sont issues peut potentiellement apporter une valeur ajoutée non négligeable.

Le plus naturel est l'utilisation des sous-parties trouvées au cours de l'apprentissage du domaine source, au sein du domaine cible.

Ces sous-parties, qui sont les bases de la création des attributs utilisés pour la classification, peuvent être aisément transférées d'une classification à une autre, ceci dans la mesure où l'utilisation d'une méthode classique de sélection de caractéristiques (*feature selection*) permet de mettre de côté les éléments qui n'auraient été que porteurs d'apprentissage négatif (*negative learning*), c'est-à-dire de transfert diminuant la qualité de la prévision.

L'idée initiale consiste donc à récupérer les sous-parties extraites du domaine source et de les utiliser conjointement (en ne conservant que les plus utiles) à celles du domaine cible. On réalise donc un *feature engineering* qui permet de récupérer l'essence même de la classification source au sein de la cible, tout en évitant le *negative learning* mentionné précédemment. On se place donc dans le cadre du transfert de modèle de classification, donc des constructeurs d'attributs, que sont les sous-surfaces pertinentes extraites. Il s'avère que l'ensemble des méthodes dont nous nous rapprochons le plus est le transfer learning basé sur les paramètres du modèle construit. Néanmoins, notre cas étant singulier, il n'est pas possible d'y appliquer directement des méthodes existantes.

Soit F_S la fonction de classification apprise pour le domaine source et F_T celle du domaine cible. Le but est de chercher à améliorer cette dernière en utilisant la précédente.

On a donc E_S l'ensemble des extraits pertinents récupérés de D_S pour alimenter F_S et E_T celui de F_T . Soit F_N la nouvelle fonction de classification visant à être meilleure que F_T au sein des mêmes données cible.

Proposition

A l'instar des travaux de [Renard et al., 2016] sur les séries temporelles, nous proposons d'évaluer la capacité d'apport des extraits, c'est-à-dire individuellement par le gain d'information envisagé dans la première partie de thèse, mais aussi en cherchant de façon conjointe si les extraits de D_S sont ou non des apports substantiels aux extraits déjà présents de E_T .

En présence d'un nombre relativement réduit d'extraits à tester en tant qu'ajout à F_T , il est possible de chercher la combinaison la plus adaptée sans pour autant appliquer une méthode aléatoire telle que Monte Carlo. Concrètement, cela revient à dire qu'à chaque ajout d'extrait au modèle, on teste le nouveau modèle créé, et en cas d'amélioration de la performance, il est conservé pour le nouveau classifieur.

Néanmoins, une telle approche requiert de recalculer un classifieur qui est alors testé pour chaque extrait rajouté, ce qui peut s'avérer long en temps de calcul.

Travailler uniquement, dans un premier temps, sur l'évaluation indépendante des extraits, comme nous l'avons proposé dans la première partie, paraît beaucoup plus opportun. Ainsi, le test sur la performance pour chaque ajout d'extrait source à l'ensemble d'extraits cibles sera retiré, la vérification seule de la pertinence indépendante des extraits ayant déjà fait ses preuves.

Nous proposons donc de simplifier grandement la notion de *transfer learning* appliquée

Algorithme 3 : Transfer learning basé sur un transfert de sous-parties discriminantes - évaluation de l'apport individuel de chaque extrait au classifieur

```

1 Entrée  $D_S$  : domaine source ;  $D_T$  : domaine cible ou target ;  $F_S$  : classifieur de  $D_S$  ;  $F_T$  :
  classifieur de  $D_T$ , complété et amélioré par l'algorithme ;  $E_S$  : extraits utilisés pour construire
   $F_S$  ;  $E_T$  : extraits utilisés pour construire  $F_T$  ; ConstruireClassifieur(Extraits) :
  Construction d'un classifieur en se basant sur les sous-parties en entrée ;
  Performance(Fonction) : évaluation de la performance, par validation croisée, de la
  fonction classifieur en entrée - utilisation de la méthode mise au point dans la Partie 1;
2 Sortie  $F_{T_{complet}}$  : classifieur de  $D_T$  complété par  $F_S$  ;  $E_{T_{complet}}$  : nouvel ensemble d'extraits
  utilisés pour bâtir  $F_T$ ;
3 Début
4  $E_{T_{complet}} \leftarrow E_T$ ;
5 pour Extrait dans  $E_S$  faire
6   | si GainInfo(Extrait) > min GainInfo( $E_T$ ) alors
7   |   |  $E_{T_{completBis}} \leftarrow E_{T_{complet}} \cup \textit{Extraits}$ ;
8   |   |  $F_{T_{completBis}} \leftarrow \textit{ConstruireClassifieur}(E_{T_{completBis}})$ ;
9   |   | si Performance( $F_{T_{completBis}}$  > Performance( $F_{T_{complet}}$ )) alors
10  |   |   |  $E_{T_{complet}} \leftarrow E_{T_{completBis}}$ ;
11  |   |   |  $F_{T_{complet}} \leftarrow F_{T_{completBis}}$ ;
12 fin
13 Fin

```

jusqu'alors, dans la mesure où la complexification des features déjà présents entraînerait logiquement la perte d'interprétabilité, contrainte pourtant essentielle que nous avons posée pour notre étude.

De plus, on se rend aisément compte que les problèmes classiques rencontrés en transfer learning, à savoir :

- comment transférer avec des espaces de *features* hétérogènes,
- utiliser conjointement de l'*active learning*,
- utiliser des sources différentes,

ne seront alors plus un problème car ce nouveau type de transfer learning ne transfère que des constructeurs d'attributs, écartant ainsi le besoin d'adaptation d'espaces de features ainsi que leurs différences de répartitions.

En effet, les simples transferts et tests des sous-surfaces venant d'autres domaines permettent dès lors de ne sélectionner que ce qui est bénéfique au domaine cible. De cette manière, on récupère le maximum de ce qui provient d'autres domaines et qui constitue les éléments les plus déterminants pour chaque classifieur source.

6.3. Expérimentations sur du transfer learning

Dans le but d'améliorer les performances de l'approche 3DRESC en présence d'un manque de données annotées, l'adaptation au transfer learning présentée est ici évaluée expérimentalement.

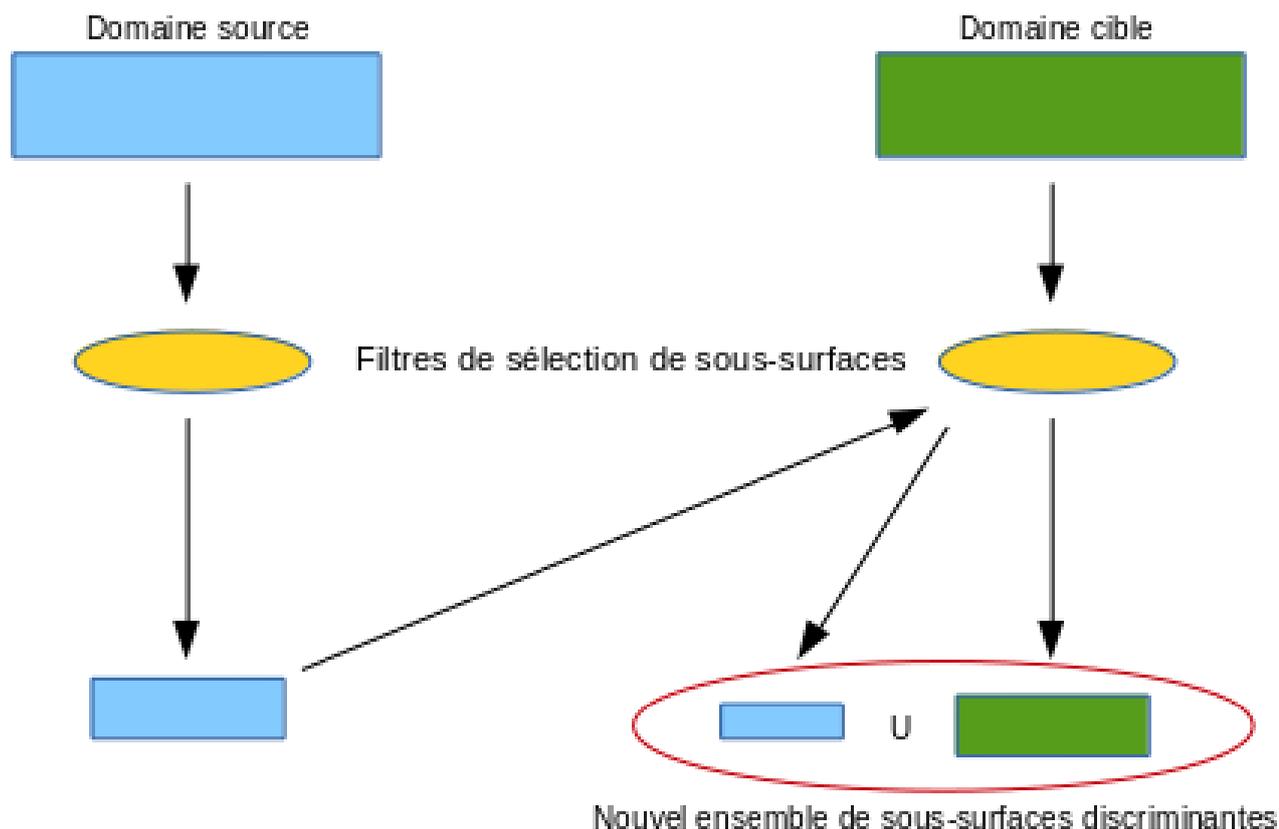


FIGURE 6.2. – Réutilisation de sous-parties pertinentes issues d’une autre classification

6.3.1. Protocole expérimental

A cette fin, le processus de test proposé consiste à récupérer un ensemble d’objets 3D annotés qui sert de domaine source, ainsi que d’un autre ensemble annoté, le domaine cible. Nous conservons les mêmes conditions que dans la partie expérimentations de la méthode 3DRESC, avec le PSB.

Le domaine cible est subdivisé en deux parties, dont l’une sert de test pour la vérification de la pertinence de la méthode.

On a donc D_S comme domaine source dont les éléments sont tous annotés, et D_T dont seule une partie l’est.

Il s’agit donc, dans un premier temps, d’apprendre indépendamment sur les deux domaines, puis de chercher à évaluer les extraits les plus discriminants de D_S dans D_T .

Les tests sont réalisés comme suit :

1. nous disposons d’un ensemble d’objets annotés du domaine source, et d’objets partiellement annotés du domaine cible (25%);

2. nous apprenons sur les objets du domaine source et cible séparément, et en récupérons les sous-parties extraites pertinentes ;
3. les extraits du domaine source récupérés sont testés selon les mêmes critères de pertinence que ceux qui ont été sélectionnés au sein du domaine cible ;
4. les extraits du domaine source précédemment filtrés au sein du domaine cible, ainsi que ceux du domaine cible sont mis en commun pour servir de base à l'algorithme 3DRESC dans le domaine cible ;
5. on en compare les résultats avec un apprentissage composé exclusivement d'extraits issus du domaine cible.

6.3.2. Expériences

L'exemple proposé est celui de traiter des représentations 3D de modèles d'avions. Les similarités sont suffisamment importantes pour que la méthode devienne alors essentielle. La classification source vise à distinguer un biplan d'un avion commercial, la cible un avion

TABLE 6.1. – Classification en fonction du nombre d'objets des domaines source et cible

Nombre d'objets dans le domaine source	Nombre d'objets dans le domaine cible	Sans apprentissage par transfert	Avec apprentissage par transfert	Amélioration de la prévision
60	40	0.616 ± 0.037	0.642 ± 0.034	4.22%
	30	0.581 ± 0.049	0.617 ± 0.047	6.20%
	20	0.545 ± 0.055	0.583 ± 0.065	7.01%
80	40	/	0.656 ± 0.036	6.49%
	30	/	0.625 ± 0.041	7.57%
	20	/	0.601 ± 0.060	10.3%
100	40	/	0.663 ± 0.034	7.63%
	30	/	0.639 ± 0.048	9.9%
	20	/	0.624 ± 0.069	14.4%

de chasse d'un planeur.

On considère, pour se rapprocher d'un problème réaliste, que 25% des objets dans le domaine cible sont annotés et 75% ne le sont pas. On récupère donc les sous-surfaces les plus discriminantes du domaine source que l'on teste dans le domaine cible, et l'on calcule celles du domaine cible annoté. Enfin, l'apprentissage est réalisé en utilisant comme constructeurs d'attributs l'union des extraits récupérés sur tous les domaines.

Il convient néanmoins de préciser qu'en présence de domaines éloignés, aucune amélioration n'aurait pu être envisagée, et que ces tests fonctionnent donc essentiellement sur des prévisions issues de domaines cible et source relativement proches. Sur des domaines comprenant à chaque fois entre 40 et 100 objets, une validation croisée est réalisée.

Les résultats s'améliorent (les domaines étant liés), et permettent de valider en Table 6.1 cette première approche.

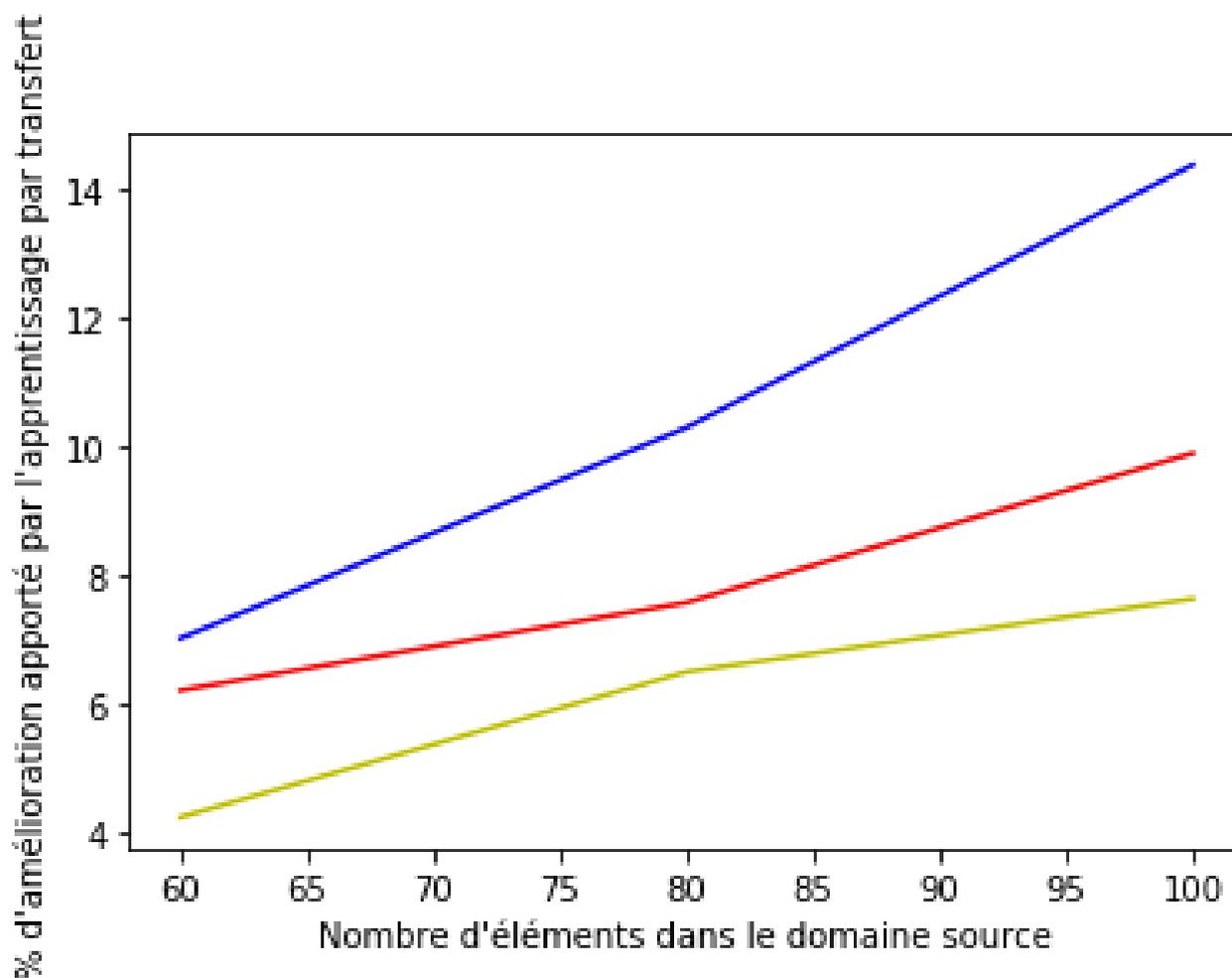


FIGURE 6.3. – Amélioration de la performance par le transfert de sous-parties, avec en bleu, rouge et jaune, respectivement 20, 30 et 40 éléments à classifier (domaine cible)

Une comparaison, pour des domaines source de tailles différentes (1ère colonne), entre une version classique de la méthode 3DRESA et un ajout de l'apprentissage par transfert introduit dans cette partie (2ème et 3ème colonnes), permet de faire ressortir cette amélioration

notable, symbolisée en % d'augmentation dans la 4ème colonne.

On remarque aisément que plus le domaine source est important, plus sa contribution peut potentiellement augmenter (Figure 6.3).

6.4. Conclusion de l'adaptation du transfer learning

Nous avons proposé une extension de la méthode 3DRESC au transfer learning [Meunier et al., 2017a], en permettant le transfert de la connaissance présente au sein es sous-parties extraites initialement, par ailleurs base de l'algorithme d'origine.

Cette adaptation de la notion de transfer learning à la méthode initiale voit son concept confirmé par les expérimentations. Il est ainsi démontré empiriquement le bien-fondé de cette idée. En effet, la réutilisation de certaines des sous-parties pertinentes du domaine source au sein d'un domaine cible différent, s'est avéré être un bon moyen d'améliorer les résultats de ce dernier. Qui plus est, la réutilisation directe, sans calcul supplémentaire, des sous-parties présélectionnées, permet un gain de performances sans pour autant alourdir le système.

Néanmoins, cette application, comme toute technique de transfert de connaissances, se limite essentiellement à la réutilisation, au sein d'un domaine cible, de sous-parties extraites d'une classification d'un ou de plusieurs domaines sources qui se doivent d'être relativement proches.

7. Une nouvelle approche de sélection d'attributs pour la classification basée sur du prototypage flou

Jusqu'à maintenant, l'algorithme 3DRESC était basé sur un certain nombre de choix, eux-mêmes issus de la méthode originale qu'étaient les Time Series Shapelets. L'un de ces verrous était la sélection, une fois le degré d'appariement calculé, des sous-parties les plus pertinentes. Le gain d'information, technique classique d'évaluation individuel de features, avait été conservée.

Pourtant, ce choix de raison, non seulement n'est pas le seul envisageable au sein de la famille des méthodes basées sur des critères statistiques, mais pourrait être remplacé par une nouvelle méthode visant à sélectionner les sous-parties les plus représentatives d'une classe.

Dans ce chapitre, il est proposé une nouvelle méthode de sélection d'attributs basée sur la théorie des sous-ensembles flous et la notion de prototype flou qui en découle. Adaptée, à notre problème, elle permet de mieux sélectionner les sous-surfaces les plus pertinentes lors de l'utilisation de l'algorithme 3DRESC.

7.1. Motivations

7.1.1. Rappels sur la sélection d'attributs

La sélection d'attributs, réalisée en prétraitement d'un processus d'apprentissage automatique, permet généralement une amélioration notable de ses performances. En effet, un nombre d'attributs trop important peut handicaper la tâche d'apprentissage, l'augmentation de la dimensionnalité du problème entraînant une instabilité du modèle appris [Friedman, 1997]. Dans la mesure où certains attributs peuvent avoir un effet négatif sur la classification, ne conserver que les quelques plus pertinents (discriminants) est primordial.

Pour rappel, en sélection d'attributs, deux principales méthodes peuvent être distinguées :

- *les méthodes par filtrage individuel* (ou filter methods) qui, en ne tenant compte que de la corrélation (ou causalité, ce qui dans ce cas revient au même) possible entre l'attribut étudié et la variable cible, évalue la première ;
- *les méthodes d'ensemble* (ou ensemble methods) qui considère un ensemble d'attributs candidat, et sélectionne un ensemble de combinaisons possibles en tenant cette fois compte des possibles interactions entre elles.

7.1.2. Discussions et objectifs

Il nous paraît essentiel de mentionner que le problème de la sélection des sous-parties pertinentes n'est pas exactement le même que celui de la sélection des attributs, pour autant que cette dernière repose sur des valeurs numériques et non sur des distances entre objets. La méthode introduite ici vise à sélectionner des caractéristiques (ou *features*) pertinentes d'éléments (des sous-parties de ces derniers) en basant sur une mesure de distance entre elles.

Ce cas peut être résumé comme un problème de sélection entre de grandes quantités de candidats et peu de places disponibles. Il est donc indispensable de pouvoir évaluer rapidement la pertinence d'une caractéristique, en utilisant seulement ses distances par rapport aux éléments et les étiquettes (ou classes d'appartenance) de ces mêmes éléments. La seule famille de méthodes répondant à cette exigence est la méthode de filtrage. De plus, si la possibilité de proposer une visualisation à l'utilisateur souhaite être conservée, l'on doit naturellement présenter des caractéristiques intrinsèquement, et non pas des bribes d'explication.

Les méthodes de filtrage sont généralement plus intéressantes que les méthodes globales quand un grand nombre de fonctionnalités doit être traité [Chandrashekar and Sahin, 2014]. Le gain d'information, mentionné précédemment, est une méthode de filtrage, accessoirement utilisée pour l'algorithme 3DRESC. Elle a par la suite été conservée au sein des algorithmes proposés dans des articles plus récents de *time series shapelets*.

Jusqu'à présent, les méthodes en sélection des caractéristiques par filtrage utilisent des outils statistiques, qui tentent généralement d'optimiser un critère précis basé sur des probabilités. On propose d'utiliser des études d'un autre domaine, à savoir la théorie des ensembles flous, pour évaluer la capacité d'une sous-partie à être représentative d'une classe, c'est-à-dire d'en être une forme de prototype. Cette introduction d'un nouvel outil dans le choix des caractéristiques permet d'espérer une sélection plus diversifiée, et une meilleure récupération des plus pertinentes, pour finalement améliorer la prévision proposée par 3DRESC.

Si l'on reprend la chaîne de traitement proposée par l'algorithme 3DRESC, on remarque la présence de trois verrous (Figure 7.1) :

1. Le descripteur d'objet 3D ;
2. La mesure de distance/similarité entre descripteurs ;
3. La sélection de sous-parties pertinentes.

Ce dernier point n'a pas été remis en question jusqu'à maintenant.

Dans cette partie, une nouvelle approche pour déterminer des mesures de typicalité des sous-parties est introduite. Elle permet la mise en évidence de sous-parties prototypiques de chaque classe [Lesot et al., 2008, Rifqi, 1996].

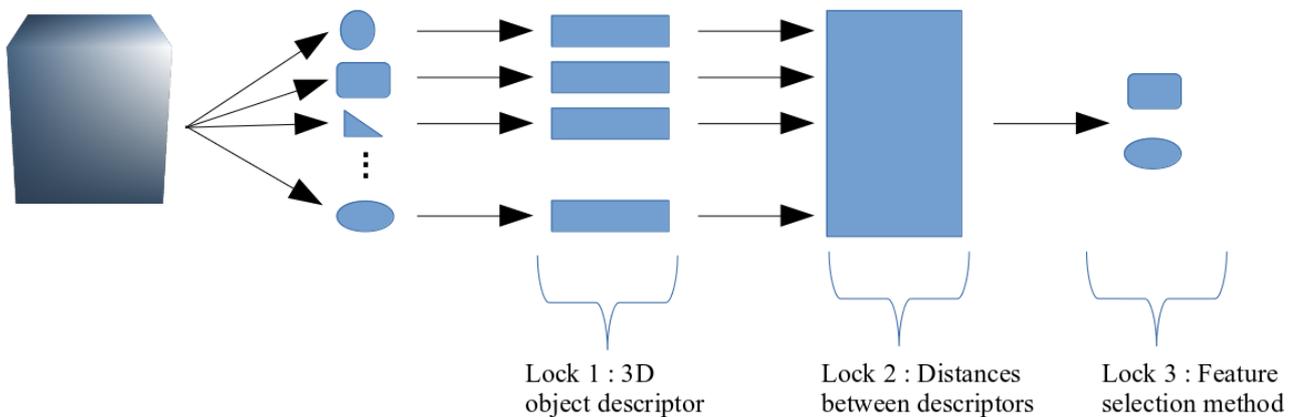


FIGURE 7.1. – Verrous à résoudre au sein de l’algorithme 3DRESC

7.2. Sélection par typicalité proposée

L’algorithme 3DRESC utilise une sélection d’attributs basée sur le gain d’information. Dans l’approche par shapelets [Ye and Keogh, 2009], d’autres techniques de sélection d’attributs du même type ont été proposées comme, par exemple, une sélection aléatoire [Renard et al., 2016] [Wistuba et al., 2015]. Dans ce chapitre, nous proposons une nouvelle approche basée sur l’utilisation d’un degré de typicalité [Lesot et al., 2008, Rifqi, 1996, Rifqi, 2010] pour réaliser la sélection de sous-parties discriminantes. Dans ce cadre, le prototype d’une classe constitue l’élément le plus représentatif de cette classe. À partir d’un ensemble d’éléments, la construction d’un prototype s’effectue de la manière suivante :

1. calculer la ressemblance d’un élément avec les autres éléments de sa classe, et sa dissimilarité avec les éléments des autres classes puis agréger ces 2 valeurs pour constituer le *degré de typicalité* de l’élément vis à vis de sa classe ;
2. ordonner les éléments d’une même classe en fonction de leur degré de typicalité ;
3. agréger les objets qui sont “les plus typiques” de leur classe afin de créer un prototype flou.

Dans ce travail, le troisième point n’est pas considéré car nous nous concentrons uniquement sur la modification de l’étape de sélection de sous-parties de l’algorithme 3DRESC, donc la sélection des éléments les plus caractéristiques / prototypiques.

Le calcul du degré de typicalité d’un élément se base sur le choix d’une mesure de ressemblance, d’une mesure de dissimilarité et d’un opérateur d’agrégation. Les mesures proposées par [Lesot et al., 2008] sont ici utilisées.

Soit $X = \{x_1, x_2, \dots, x_n\}$ un ensemble d’éléments et Y un ensemble de classes associées. Pour tout $x_i \in X$, $i = 1, \dots, n$, on note y_i la classe associée à x_i . Étant donné $x_i \in X$, on note C_i l’ensemble des exemples de X qui possèdent la même classe que x_i : $C_i = \{x_j \in X \mid x_j \neq x_i \text{ et } y_j = y_i\}$. Dans la suite, on considère $\bar{C}_i = \{x_j \in X \mid x_j \notin C_i\}$.

Soit δ , une mesure de distance définie sur X . Les ressemblances et dissimilarités usuelles sont appelées dans [Lesot, 2005].

La dissimilarité $d(x_i, x_j)$ entre deux éléments x_i et x_j de X donnée, dans sa version linéaire, est définie par :

$$d_1(x_i, x_j) = \frac{\delta(x_i, x_j) - d_m}{d_M - d_m} \quad (7.1)$$

avec d_m et d_M des paramètres de normalisation fixés, par exemple, à 0 pour d_m et au maximum des distances entre élément de X pour d_M . Le problème rappelé dans [Lesot, 2005] est celui de la distance maximale qui, en cas de données très grandes, peut interférer négativement dans le processus de normalisation.

Un moyen de contourner ce problème est l'utilisation de paramètres $Z = (m, M)$ telle que $d < M$ et $d > m$:

$$d_2(x_i, x_j) = \max \left(\min \left(\frac{\delta(x_i, x_j) - m}{M - m}, 1 \right), 0 \right) \quad (7.2)$$

De cette manière, les valeurs extrêmes sont définies par l'utilisateur et non pas dépendantes des données.

La ressemblance entre deux éléments x_i et x_j de X peut se déterminer, soit à partir de la mesure de dissimilarité, soit à partir du produit scalaire.

Si l'on choisit de se baser sur la dissimilarité, alors les formules possibles sont :

— par complémentaire à 1 :

$$r_1(x_i, x_j) = 1 - d(x_i, x_j) \quad (7.3)$$

— par la formule de Laplace :

$$r_2(x_i, x_j) = \frac{1}{1 + \left(\frac{d(x_i, x_j)}{\sigma} \right)^\gamma} \quad (7.4)$$

— par la Gaussienne généralisée :

$$r_3(x_i, x_j) = \exp \left(- \left(\frac{d(x_i, x_j)}{\sigma} \right)^\gamma \right) \quad (7.5)$$

— par la fonction sigmoïde :

$$r_4(x_i, x_j) = \frac{F(d(x, y)) - F(2\sigma)}{F(0) - F(2\sigma)} \quad (7.6)$$

avec :

$$F(z) = \frac{1}{1 + \exp \left(\frac{z - \sigma}{\gamma} \right)} \quad (7.7)$$

et γ un paramètre fixé par l'utilisateur. Par commodité, le choix s'est porté sur $\gamma = 2$ lors des expérimentations.

Si l'on choisit, au contraire, de se baser sur le produit scalaire, alors on a la fonction de base suivante :

$$r_5(x_i, x_j) = \langle x, y \rangle \quad (7.8)$$

[Lesot, 2005] propose l'utilisation de la fonction Gaussienne :

$$r_6(x_i, x_j) = \exp\left(-\left(\frac{\delta(x-y)}{\sigma}\right)^2\right) \quad (7.9)$$

Ici, et contrairement à la fonction Gaussienne mentionnée plus haut, on n'utilise non pas la dissimilarité mais la distance.

Le degré de typicalité d'un élément $x_i \in X$, comme par exemple celui proposé par [Lesot et al., 2008], se calcule à partir de la ressemblance globale R de x_i à tous les éléments de sa classe et de sa dissimilarité globale D à tous les éléments qui ne sont pas de sa classe.

$$R(x_i, C_i) = \frac{1}{|C_i|} \sum_{x_j \in C_i} r(x_i, x_j)$$

et

$$D(x_i, C_i) = \frac{1}{|\bar{C}_i|} \sum_{x_j \in \bar{C}_i} d(x_i, x_j)$$

Le degré de typicalité T pour un élément $x_i \in X$ vis à vis de sa classe se détermine en agrégeant ces deux mesures globales. Par exemple, si l'on considère la mesure d'agrégation proposée dans [Lesot et al., 2008] :

$$T(x_i, C_i) = \frac{R(x_i, C_i) + D(x_i, C_i)}{-R(x_i, C_i) * D(x_i, C_i)} \quad (7.10)$$

Une autre possibilité serait de proposer d'adapter une version normalisée de la t-norme paramétrique de Yager [Yager, 1980] [?], définie par :

$$T(x_i, C_i) = \frac{\max(0, 1 - [(1 - R_n(x_i, C_i))^\gamma + (1 - D_n(x_i, C_i))^\gamma]^{\frac{1}{\gamma}})}{1} \quad (7.11)$$

avec R_n et D_n les versions normalisées de R et D , et $\gamma > 0$ un paramètre de l'équation fixé à 2 par commodité.

Au final, dans notre cadre, pour l'application à l'algorithme 3DRESC, l'ensemble des sous-parties issues des objets est ordonné à l'aide du degré de typicalité de celles-ci vis-à-vis de leurs classes.

7.3. Expérimentations sur la sélection de caractéristiques par des outils de logique floue

Nous conservons les mêmes conditions que dans la partie expérimentations de la méthode 3DRESC, avec le PSB.

Pour le reste du chapitre, les notations suivantes sont admises :

- r_n pour ressemblance n
- d_m pour la dissimilarité m
- a_k pour l'agrégation k

Les expérimentations sont faites avec les mesures de dissimilarité des équations (7.1) et (7.2), et les mesures de ressemblance des équations (7.3), (7.4), (7.5), (7.6), (7.8) et (7.9), et comme fonction d'agrégation, celle de l'équation (7.10).

7.3.1. Sélection des ressemblance, dissimilarité et agrégation

Dans la table 7.1, les résultats de la comparaison de différentes approches de sélections sont présentés :

- une méthode de sélection aléatoire des extraits utilisés pour la construction d'attributs (comme *baseline*) ;
- la méthode de sélection utilisant le gain d'information (méthode classique conservée pour les shapelets de séries temporelles et 3DRESC) ;
- la méthode de sélection utilisant le gain d'information et les différentes méthodes de sélection issues des diverses dissimilarité, ressemblance et typicalité (exclusivement l'équation (7.10)) explicitées plus haut.

L'association de la ressemblance r_n avec la dissimilarité d_m en utilisant l'agrégation a_k est appelée $a_k - r_n - d_m$.

Le choix pour le calcul par typicalité se concentre sur le triplet $a_1 - r_2 - d_2$, des équations (7.10)–(7.4)–(7.2), car il combine à la fois simplicité et performance.

7.3.2. Vérification de la pertinence du couple HKS / Matusita

Comparons maintenant cette nouvelle méthode, créée par le triplet précédemment trouvé, à des techniques plus courantes.

L'utilisation d'une sélection aléatoire peut être maintenant exclue en raison des mauvaises prévisions observées dans la Table 7.1. Il s'avère que la méthode proposée ici est égale ou supérieure à sa performance du gain d'information (Figure 7.2).

Cependant, il est toujours possible que le choix du couple descripteur / distance trouvé, adapté à la méthode originale 3DRESC (HKS et Matusita), devienne moins efficace avec la nouvelle approche qu'est $a_1 - r_2 - d_2$.

Afin de résoudre ce problème, on propose de comparer certains des meilleurs couples de descripteurs / distance (selon AUC) déterminés lors des études précédentes sur 3DRESC sur chacune des approches de sélection de caractéristiques proposées précédemment.

7.3. Expérimentations sur la sélection de caractéristiques par des outils de logique floue

TABLE 7.1. – AUC de la classification avec utilisation de la typicalité comme critère de sélection d'attribut

Méthode de sélection d'attribut			Nombre total d'objets			
			40 objets	60 objets	80 objets	100 objets
Typycalité						
a_1	r_1	d_1	0.622 ± 0.081	0.635 ± 0.084	0.637 ± 0.085	0.651 ± 0.082
		d_2	0.631 ± 0.079	0.629 ± 0.099	0.639 ± 0.103	0.650 ± 0.079
	r_2	d_1	0.741 ± 0.099	0.761 ± 0.093	0.786 ± 0.091	0.803 ± 0.094
		d_2	0.747 ± 0.103	0.763 ± 0.105	0.785 ± 0.091	0.801 ± 0.097
	r_3	d_1	0.723 ± 0.088	0.739 ± 0.092	0.756 ± 0.091	0.782 ± 0.085
		d_2	0.717 ± 0.089	0.740 ± 0.086	0.759 ± 0.097	0.778 ± 0.091
	r_4	d_1	0.727 ± 0.099	0.742 ± 0.093	0.754 ± 0.091	0.775 ± 0.094
		d_2	0.716 ± 0.087	0.738 ± 0.087	0.755 ± 0.092	0.773 ± 0.097
	r_5	d_1	0.711 ± 0.054	0.729 ± 0.087	0.744 ± 0.098	0.759 ± 0.081
		d_2	0.714 ± 0.085	0.731 ± 0.081	0.739 ± 0.094	0.757 ± 0.089
	r_6	d_1	0.743 ± 0.075	0.761 ± 0.099	0.782 ± 0.083	0.802 ± 0.101
		d_2	0.745 ± 0.097	0.759 ± 0.099	0.779 ± 0.091	0.798 ± 0.095
a_2	r_1	d_1	0.612 ± 0.078	0.631 ± 0.087	0.640 ± 0.098	0.652 ± 0.085
		d_2	0.630 ± 0.075	0.627 ± 0.091	0.640 ± 0.097	0.655 ± 0.079
	r_2	d_1	0.738 ± 0.095	0.754 ± 0.097	0.783 ± 0.084	0.898 ± 0.096
		d_2	0.751 ± 0.101	0.758 ± 0.105	0.791 ± 0.087	0.797 ± 0.094
	r_3	d_1	0.708 ± 0.091	0.727 ± 0.090	0.738 ± 0.094	0.789 ± 0.081
		d_2	0.723 ± 0.085	0.731 ± 0.088	0.754 ± 0.101	0.775 ± 0.094
	r_4	d_1	0.722 ± 0.076	0.737 ± 0.091	0.749 ± 0.086	0.778 ± 0.093
		d_2	0.721 ± 0.071	0.739 ± 0.082	0.757 ± 0.090	0.777 ± 0.099
	r_5	d_1	0.717 ± 0.051	0.728 ± 0.089	0.741 ± 0.102	0.752 ± 0.084
		d_2	0.715 ± 0.084	0.727 ± 0.085	0.740 ± 0.091	0.752 ± 0.092
	r_6	d_1	0.737 ± 0.077	0.759 ± 0.097	0.782 ± 0.081	0.795 ± 0.97
		d_2	0.746 ± 0.098	0.757 ± 0.101	0.775 ± 0.093	0.791 ± 0.094
Gain d'information			0.737 ± 0.097	0.762 ± 0.097	0.783 ± 0.097	0.801 ± 0.097
Sélection aléatoire			0.594 ± 0.156	0.609 ± 0.146	0.611 ± 0.147	0.604 ± 0.143

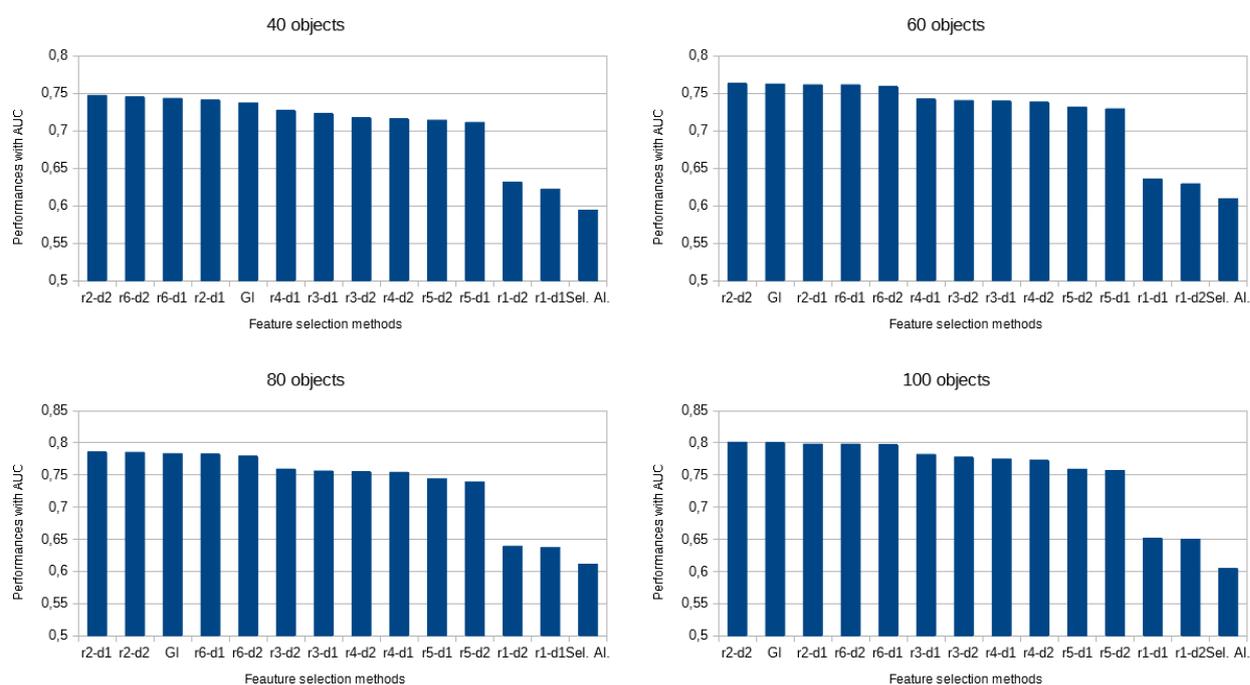


FIGURE 7.2. – Performances des méthodes de sélection d'attributs (gain d'information, sélection aléatoire et différentes typicalités) pour plusieurs tailles de jeu de données

7.3. Expérimentations sur la sélection de caractéristiques par des outils de logique floue

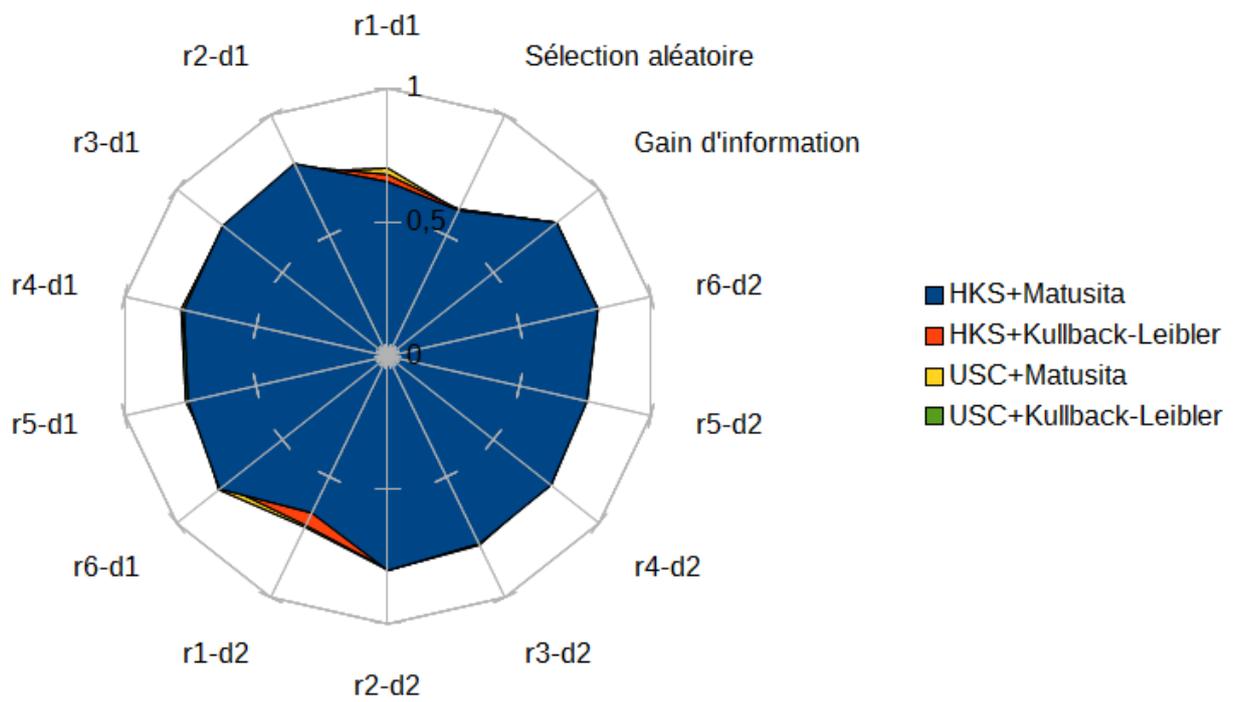


FIGURE 7.3. – Performances des couples de distance / descripteur en fonction des méthodes de sélection de caractéristiques employées

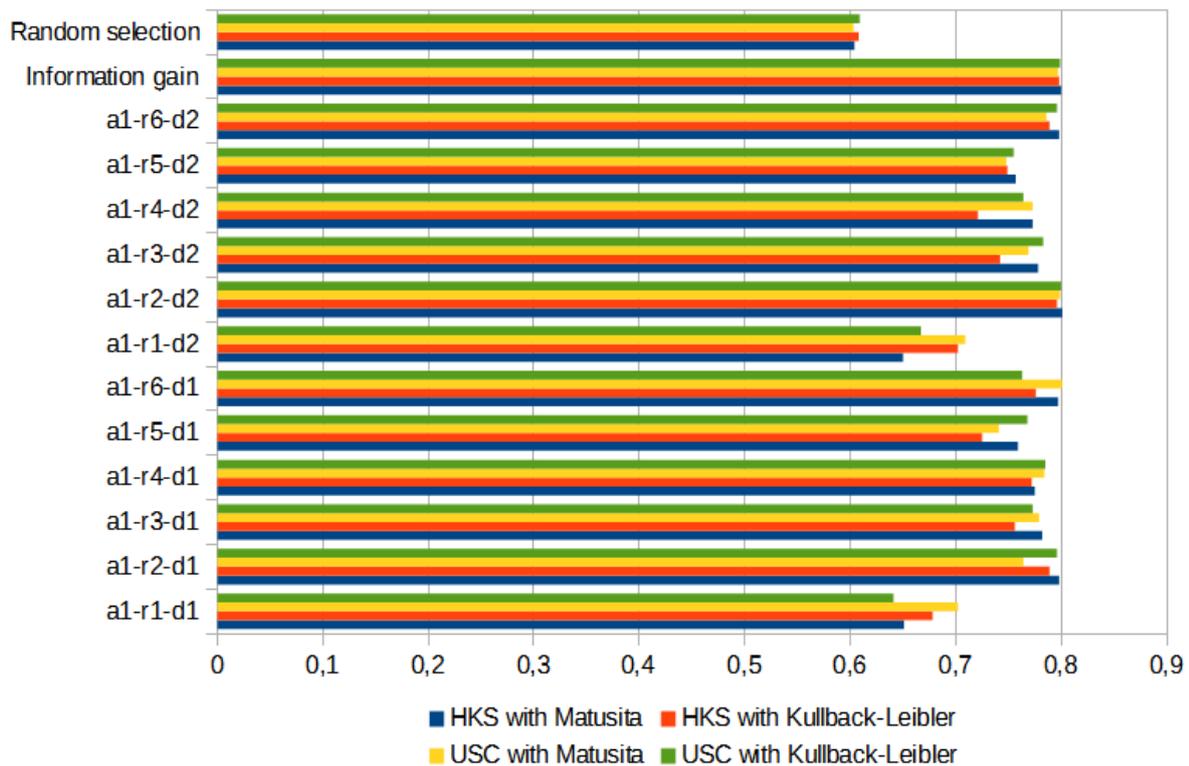


FIGURE 7.4. – Performances des couples de distance / descripteur en fonction des méthodes de sélection de caractéristiques employées (bis)

Figure 7.3 présente une visualisation de ces résultats, permettant de mettre en évidence le fait que, hormis quelques méthodes qui semblent globalement moins efficaces, le couple original, qu’était HKS / Matusita, reste le meilleur en termes de performances de prévision dans la mesure où ce dernier recouvre graphiquement presque tous les autres couples possibles.

Un autre graphique proposé, la Figure 7.4, représente différemment ces résultats, de façon plus traditionnelle, et confirme la précédente affirmation.

7.3.3. Combinaison de méthodes différentes

Néanmoins, il ne suffit pas d’égaliser les performances actuelles des méthodes de sélection d’attributs pour justifier pleinement la nécessité d’une nouvelle approche. Aussi, nous proposons désormais d’améliorer la portée de cette proposition en combinant les avantages que ces deux approches peuvent apporter (c’est-à-dire la typicalité $a_1 - r_2 - d_2$ avancée et le gain d’information d’origine). L’intérêt de l’utilisation d’une méthode aussi différente réside surtout dans la diversification qu’elle apporte lors de la sélection des sous-parties.

7.3. Expérimentations sur la sélection de caractéristiques par des outils de logique floue

On a déjà mentionné le point clé de cette nouvelle technique de sélection des caractéristiques : les entités extraites sont sélectionnées en fonction d'autres critères que ceux utilisés par les méthodes classiques. Par ailleurs, en utilisant deux méthodes, on obtient deux ensembles de caractéristiques assez différents (affirmation confirmée par les résultats de Figure 7.3).

Il en découle naturellement la dernière proposition de ce chapitre qui est de comparer les résultats de 3DRESC, avec une variation de la distribution (ou répartition) de l'utilisation des méthodes de sélection des caractéristiques :

- typicalité seule ;
- gain d'information seul ;
- combinaison des deux précédentes méthodes, définie en % du nombre total d'attributs sélectionnés.

Concrètement, au lieu de choisir n attributs en utilisant une seule méthode, on propose d'utiliser conjointement $k\%$ de n avec de la typicalité et $(100 - k\%)$ de n avec du gain d'information. Afin d'obtenir une vue d'ensemble des effets de cette combinaison, plusieurs proportions de sous-parties sélectionnées pour chaque méthode de sélection sont testées dans Table 7.2.

Ce dernier ensemble d'expériences met en évidence la combinaison du nouveau type de sélection par degré de typicalité en le combinant avec l'approche du gain d'information. Dans ce cas, la contribution est faite par l'utilisation conjointe des sous-parties trouvées par chacune de ces deux approches. Ainsi, les sous-parties sélectionnées proviennent soit de l'approche par gain d'information (*I.G.*), soit de l'approche par typicalité.

L'intérêt de l'utilisation de méthodes aussi différentes réside dans la diversification qu'elle apporte pour la sélection des sous-parties, et non plus de la performance de la seule méthode proposée.

L'idée revient à utiliser l'union des meilleurs extraits de chaque méthode de sélection de caractéristiques pour couvrir la plupart des possibilités d'éléments pertinents selon chaque critère.

Quel que soit le nombre d'objets, on peut constater que les résultats sont améliorés et permettent de confirmer l'intérêt de cette sélection par typicalité. Le gain de performance peut s'expliquer par l'intérêt de constituer un ensemble de sous-parties sélectionnées en introduisant une diversité apportée par le choix de différents filtres. En effet, ils résultent directement de l'utilisation conjointe de différents critères de pertinence. Il s'avère que, d'après les résultats donnés dans la Figure 7.5, la distribution de 50% – 50% entre typicalité et gain d'information permet la meilleure amélioration de la prévision, bien que le nombre total de sous-parties reste le même.

Afin de mieux comprendre les raisons de cette amélioration, la Table 7.3 montre le rapport du nombre de doublons dans les extraits retenus sur le nombre total d'extraits dans le cas d'une distribution équilibrée (50% – 50%).

Est considérée comme doublon une sous-partie sélectionnée qui a été mise en avant à la fois par le gain d'information et les méthodes de typicalité. La présence de doublons étant très

TABLE 7.2. – AUC pour la classification avec combinaison d'attributs

Répartition de sous-parties choisie (en % de l'ensemble)		Nombre total d'objets			
Gain d'information	Typicalité	40 objets	60 objets	80 objets	100 objets
100%	0%	0.737 ± 0.097	0.762 ± 0.099	0.783 ± 0.106	0.801 ± 0.105
80%	20%	0,763 ± 0.094	0.784 ± 0.096	0.811 ± 0.098	0.826 ± 0.102
60%	40%	0.787 ± 0.111	0.806 ± 0.102	0.821 ± 0.088	0.836 ± 0.089
50%	50%	0.790 ± 0.105	0.798 ± 0.102	0.824 ± 0.101	0.842 ± 0.094
40%	60%	0.786 ± 0.108	0.790 ± 0.089	0.819 ± 0.101	0.837 ± 0.085
20%	80%	0.779 ± 0.099	0.782 ± 0.093	0.807 ± 0.091	0.826 ± 0.094
0%	100%	0.743 ± 0.099	0.774 ± 0.093	0.782 ± 0.091	0.811 ± 0.094

TABLE 7.3. – Présence de doublons

Nombre d'objets	Pourcentage de doublons
40	0%
60	0%
80	0.5%
100	0%

7.3. Expérimentations sur la sélection de caractéristiques par des outils de logique floue

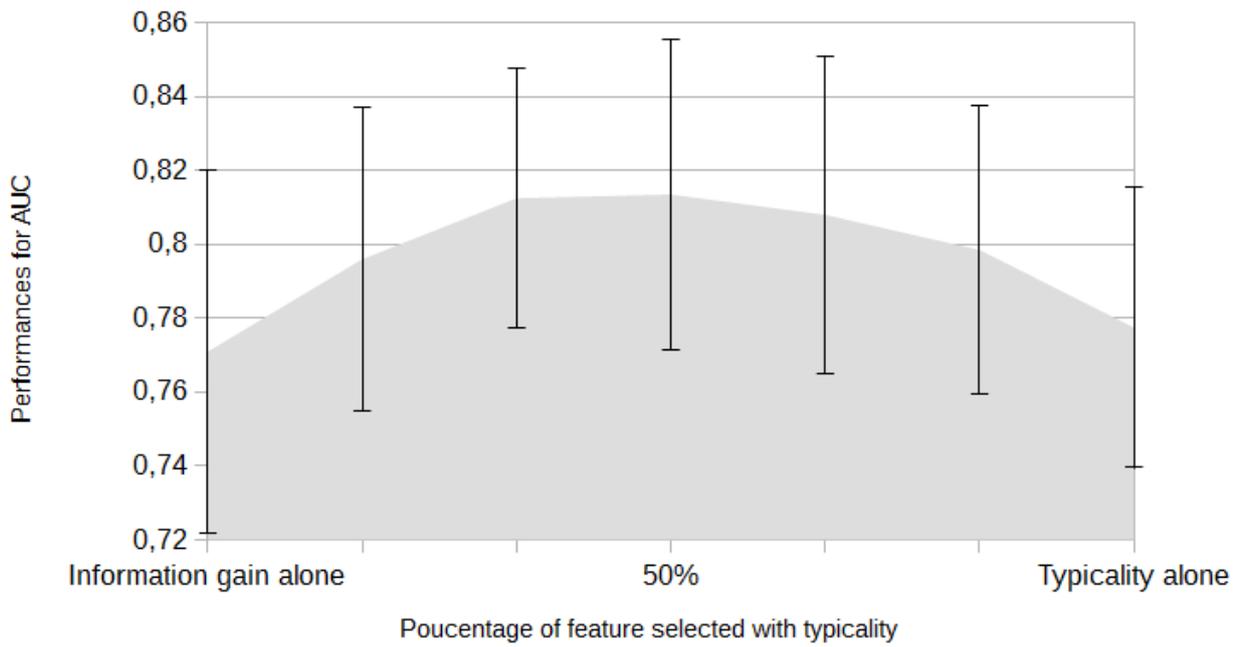


FIGURE 7.5. – Gain d’information (à gauche) ou typicalité (à droite) pour la répartition des sous-parties selon leur méthode de sélection

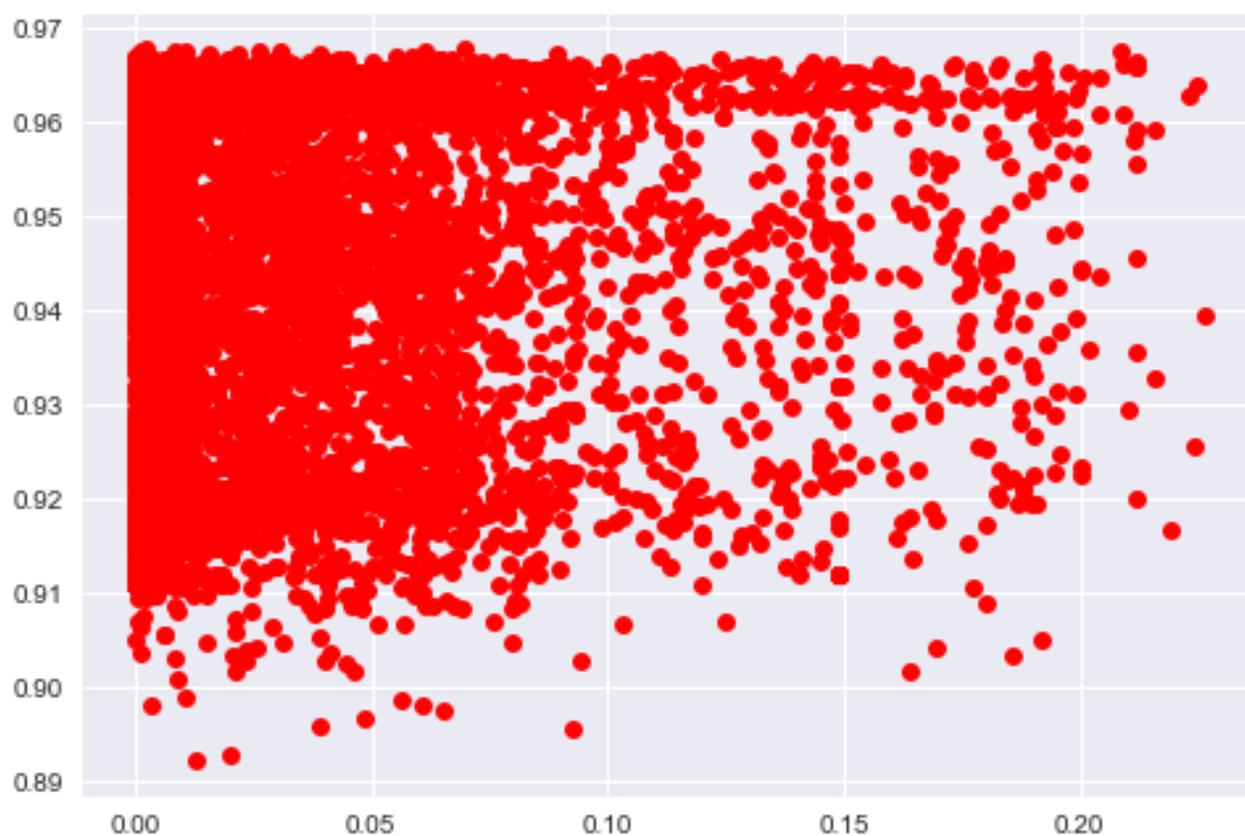


FIGURE 7.6. – Gain d'information (en abscisse) et typicalité (en ordonnée)

faible, on peut en déduire que :

- utiliser conjointement ces deux méthodes n'est pas absurde,
- le gain de performance est dû à la variété introduite par l'utilisation de différentes approches de sélection.

De plus, la non-corrélation des valeurs de gain d'information et de typicalité est également illustrée par une projection sur la Figure 7.6. Chaque point de cette route correspond à une sous-partie pour laquelle la valeur en abscisse est le gain d'information obtenu, et la valeur en ordonnée est son degré de typicalité.

7.3.4. Bilan

Après expérimentations, il s'est avéré que l'idée initialement proposée qui consistait à utiliser des outils issus de la logique floue ont permis de mettre en place une nouvelle méthode de sélection de caractéristiques pertinentes. Appliquée à la méthode 3DRESC proposée en chapitre 2, une combinaison de la méthode avec le gain d'information classique améliore la performance de façon notable. Par ailleurs, les paramétrages précédemment choisis restent adaptés, ne remettant ainsi en cause que la méthode de sélection de sous-surfaces.

7.4. Conclusion de la sélection par typicalité

Dans cette partie, nous avons proposé une nouvelle méthode de sélection d'attributs pertinents pour la classification supervisée d'objets 3D [Meunier et al., 2017c]. Cette approche combine l'utilisation du gain d'information et d'un degré de typicalité afin de retenir les caractéristiques les plus pertinentes pour la reconnaissance de la classe. Ainsi, l'introduction d'une plus grande diversité dans le choix des caractéristiques ou sous-parties (dans notre cas) sélectionnées devient un critère pour l'amélioration des performances, permettant à l'algorithme 3DRESC de voir ses résultats encore améliorés.

Une étude plus approfondie, en particulier des différences d'ordonnancement provoquées par chacune de ces approches reste à mener. D'autres mesures de ressemblance, de dissimilarité ou d'agrégation pourraient également être étudiées afin de mettre en évidence d'autres apports éventuels d'amélioration des performances.

Par ailleurs, cette méthode de sélection d'attributs, ici utilisée dans le cadre de l'amélioration des performances de 3DRESC, pourrait fort bien être appliquée à une classification supervisée plus traditionnelle.

8. Conclusion de la partie II

Dans ce chapitre, nous avons proposé de valoriser différemment les sous-parties d'objets 3D, et ce de deux manières différentes.

La première consiste à utiliser conjointement des sous-parties issues de domaines différents, afin de permettre, par une adaptation du transfer learning à l'algorithme 3DRESC, de valoriser des classifications passées, proches de celle que l'on veut améliorer, mais sans être similaires à cette dernière.

Le seconde cherche à proposer une méthode novatrice de sélection de sous-parties, basée sur des outils issues de la théorie des sous-ensembles flous.

Ces apports confirment la pertinence du choix de sous-surfaces discriminantes comme supports de connaissances dans un cadre de classification supervisée d'objets 3D.

Conclusion générale

Dans ce chapitre, nous rappelons les objectifs initiaux de cette thèse, avant d'en présenter les principales propositions. Finalement, plusieurs perspectives à ce travail sont abordées.

Bilan

A la croisée des besoins de Total en termes de valorisation de l'information et des récentes recherches en matière de classification supervisée, de sélection et de combinaison d'attributs, cette thèse propose une nouvelle approche de traitement des structures 3D complexes.

Il s'agit dans un premier temps de traiter la mise en place d'un système de classification supervisée d'objets 3D, que sont les géo-modèles de structures géologiques, avec justification de la méthode et du modèle obtenus. En effet, l'exhibition des sous-parties les plus pertinentes pour la classification permet l'interprétation par l'utilisateur ainsi que la compréhension du système proposé. Le principal problème soulevé est celui de la création d'attributs pertinents, donc du pré-traitement de ces derniers et de la sélection qui en découle. Il est mis en lumière, de façon empirique, l'intérêt de valoriser l'extraction aléatoire de sous-parties d'une entité. Une telle démarche se révèle possible grâce aux progrès dans les domaines des séries temporelles, des objets 3D, des nuages de points, et aux capacités de calcul toujours croissantes.

Il importe ensuite d'améliorer l'algorithme précédemment proposé, en utilisant d'une part des notions de transfert de connaissances et de sélection non supervisée de caractéristiques (ou sous-parties), et d'autre part en suggérant une nouvelle méthode de sélection d'attributs, applicable à tout type d'apprentissage supervisé.

L'apport fondamental de ce travail est la valorisation de données par une classification supervisée. En utilisant une extraction aléatoire de sous-parties issues de ces données, de nouvelles possibilités de prévision se dévoilent. Cela confirme le rôle fondamental du traitement des attributs, préalable à tout processus d'apprentissage, afin d'en optimiser grandement les retombées. Rôle d'autant plus fondamental que l'on fait face à des types de données complexes.

Ainsi, dans cette thèse, nous introduisons :

1. 3DRESC, un algorithme de classification d'objets 3D par extraction aléatoire de sous-parties discriminantes.

Il s'agit d'une nouvelle approche qui justifie la méthode et ses résultats en présentant les sous-parties les plus discriminantes. Ces sous-parties sont les éléments les plus ca-

ractéristiques pour la classification et permettent ainsi à l'utilisateur de faire confiance au modèle et le valider. Au carrefour des techniques utilisées pour les séries temporelles et le traitement des objets 3D, cette méthode aide un expert à mieux appréhender les sous-parties les plus utiles d'un objet 3D en les portant à sa connaissance.

Ainsi, l'utilisateur comprend la classification car le point fort n'est pas exclusivement la classification (et donc le taux de précision le seul objectif), mais la capacité à fournir une explication au modèle de classification donné.

Une variante proposée est celle de la sélection non supervisée de sous-parties. Selon des critères de distribution, et afin d'apporter de la variété dans les attributs récupérés, on complète l'ensemble des sous-parties sélectionnées par 3DRESC. Ainsi, les performances s'en trouvent améliorées de façon non négligeable, sans que la complexité de l'algorithme ne s'en trouve modifiée.

2. 3DRESC-TF, une adaptation du précédent algorithme grâce à une approche de transfert de connaissances.

Nous avons proposé d'adapter l'algorithme 3DRESC à l'aide du transfer learning. Ce concept permet de mettre en avant la connaissance présente, non pas dans le modèle d'apprentissage en lui-même, mais au sein de l'extraction et de la sélection de sous-parties discriminantes. Les expériences confirment la pertinence du transfert de sous-surfaces, bases de la connaissance créée par 3DRESC lors de la classification.

3. Une sélection non supervisée des sous-parties pertinentes.

Ne se reposant plus sur les classes d'appartenance des objets auxquels appartiennent les sous-parties, l'approche avancée ici utilise exclusivement la forme des sous-parties pour en sélectionner les plus pertinentes. Cela permet de valoriser les ensembles de données non annotées, potentiellement bien plus nombreuses que celles qui le sont déjà.

4. Une nouvelle méthode de sélection de sous-parties basée sur la théorie des sous-ensembles flous.

Au lieu d'utiliser les méthodes traditionnelles de sélection individuelle d'attributs pertinents (par filtrage), une méthode utilisant des outils issus de la théorie des ensembles flous a été proposée. Cette méthode se base sur des indicateurs tels que la ressemblance, la dissimilarité et l'agrégation, pour évaluer la capacité d'une sous-partie à représenter une classe.

Ainsi, en basant la sélection d'attributs sur du prototypage flou, on propose des résultats encore plus satisfaisants. On recherche alors des sous-parties prototypiques, caractérisant l'ensemble auquel l'objet initial appartient.

5. Une adaptation aux données réelles de Total.

Finalement, des tests sur données réelles ont été mis en oeuvre, bien que celles-ci soient des nuages de points de grandes tailles et non plus des maillages ouverts irréguliers. Néanmoins, une légère adaptation des paramètres a pu aisément adapter la méthode 3DRESC aux nouvelles contraintes rencontrées.

Les bonnes performances obtenues ainsi que la robustesse face à un sous-échantillonnage permettent d'entrevoir les multiples possibilités qu'un tel concept recèle.

Perspectives

On peut dès lors distinguer deux principaux axes d'ouverture permis par ce travail de thèse : améliorer les méthodes proposées, et considérer de nouveaux champs de recherche.

Continuer sur cet axe de recherche implique de remettre en question certaines des plages de valeurs données aux paramètres. En effet, les données réelles à rentrer dans l'algorithme tendront à reconsidérer des décisions qui sont parfois empiriques. Elles devront néanmoins être adaptées si les données diffèrent, comme par exemple le choix de la taille des sous-parties extraites et leur nombre. Ce genre de paramétrage serait alors atteint par l'utilisation d'une connaissance préalable à la classification, sous la forme d'une connaissance des sous-parties pertinentes et de leur fréquence/redondance dans les objets.

L'automatisation du choix de ces paramètres deviendrait logiquement plausible. La méthode s'adapterait alors naturellement aux types de données et à leurs particularités statistiques. Qui plus est, la méthode proposée en sélection de caractéristiques pertinentes, basée sur des outils issus de la logique floue, devrait être testée sur des tâches de classifications plus traditionnelles, afin d'en évaluer l'adaptabilité à des domaines pour lesquelles une telle méthode n'était a priori pas conçue.

En plus de cette application directe, la méthode pourrait être réutilisée en traitement automatisé de structures issues de l'étude d'images traditionnelles (et non pas venant de la reconstitution d'ondes acoustiques).

Par ailleurs, on peut aisément entrevoir la possibilité de réutiliser les méthodes proposées pour le traitement automatisé de nuages de points, afin d'aider, dans un autre cadre d'application, un robot à reconnaître des objets.

Aussi, la méthode pourrait aussi être adaptée aux cas d'objets partiellement obscurcis (dont l'information de description n'est pas connue en totalité, donc qu'une partie de l'objet n'est pas connue), ouvrant la voie à la vision par ordinateur.

Mais les possibilités ne s'arrêtent pas là : dans le domaine de l'apprentissage des séries temporelles, une de nos sources d'idées dans cette thèse, pourraient se voir adaptées les propositions concernant, non seulement le transfer learning, mais aussi la sélection non supervisée de sous-parties, ce dans le but de faire face à un manque, aujourd'hui très fréquent, de données annotées.

Une perspective à plus long terme serait de considérer conjointement les différentes sous-parties, et pouvoir en créer automatiquement de nouvelles nous semble particulièrement pertinent. Il s'agirait d'une création de sous-parties, spécialement conçues pour la tâche que l'on souhaite leur attribuer, la classification dans notre cas. Ces sous-parties alors créées seraient plus pertinentes car mieux adaptées à la visualisation et à la prédiction des classes d'appartenance, offrant ainsi des performances encore supérieures. De plus, il deviendrait envisageable de reconstituer un objet prototype [Li et al., 2017] en entier, permettant de

Conclusion de la partie II

définir plus clairement les caractéristiques d'une classe, afin de faire toujours mieux accepter le principe de l'intelligence artificielle, et par la même l'apprentissage automatique, au sein des métiers.

Par ailleurs, il a été conservé, lors de la mise au point de l'algorithme 3DRESC, la structure initiale des Times Series Shapelets. Cette structure initiale impose, entre autres, de distinguer la partie qui crée les features pertinents de celle, plus classique, d'apprentissage par random forests. cette deuxième partie n'était pas le coeur de la méthode. Une vision globale, par exemple en utilisant des techniques de bagging, pourrait aisément être envisagée.

Enfin, nous nous sommes jusqu'alors intéressés principalement à la classification supervisée : permettre le passage de nos travaux sur les shapelets aux domaines de la régression et de l'ordonnancement, et ainsi réussir à valoriser, dans ces domaines, les sous-parties d'entités qui jusqu'à maintenant n'étaient utilisées que pour classifier.

Annexes

A. Formalisations des notions en lien avec les objets 3D et leurs descripteurs

Cette annexe vise à mieux définir des notions utilisées au cours de ce travail de thèse, en lien avec les contraintes pesant sur les objets 3D et leurs descripteurs. A cette fin, il est proposé, dans un premier temps, de rappeler la définition que l'on entend par objet 3D, puis les conditions admises dans cette thèse afin de correspondre aux besoins de Total, et enfin les caractéristiques qu'un descripteur doit respecter pour être un bon candidat.

A.1. Les objets 3D

On appelle objets 3D une surface évoluant dans un espace à 3 dimensions qu'est l'espace usuel.

On note $P \subseteq \mathbb{R}^3$ l'ensemble des points de cette surface. Cette dernière est étudiée comme un maillage triangulaire (cf. Figure A.1) modélisé sous la forme d'un graphe $G = \{S, V, T\}$, avec $S \subseteq P$ les sommets, $V = \{\langle s_1, s_2 \rangle \mid s_1, s_2 \in S\}$ les arêtes reliant les sommets et $T = \{\langle s_1, s_2, s_3 \rangle \mid s_1, s_2, s_3 \in S\}$ les triangles du maillage.

Nous considérons donc le modèle 3D comme une surface telle que $M = \{P, G\}$. Cette surface est non fermée et le maillage qui la décrit est irrégulier. Un maillage irrégulier signifie qu'il n'y a pas de règle a priori sur les longueurs des segments composant les mailles. Quant à la caractéristique d'objet 3D non fermée, elle est explicitée dans la section suivante.

A.2. La notion d'objet ouvert/fermé

Ne trouvant pas de définition dans la littérature, il est possible de définir formellement la notion d'objet 3D non fermé en utilisant des notions qui se réfèrent à la topologie.

On considère qu'un objet 3D (défini plus haut) est ouvert (ou non fermé) s'il est possible, pour tout couple de points de l'espace, tous deux n'appartenant pas à une maille de l'objet, de les relier par une succession de segments, sans pour autant traverser une maille de l'objet. La maille est comprise ici comme la surface triangulaire formée par les trois points qui la définissent.

En utilisant les notations précédemment admises :

Définition 1. *Un objet 3D est dit ouvert si, pour tout couple de points $(p_1, p_2) \in S^2$ tels que p_1 et p_2 n'appartiennent pas à une des mailles de T , alors il existe un ensemble de segments $R = \{\langle p_a, p_b \rangle \mid p_a, p_b \in S\}$ tels que :*

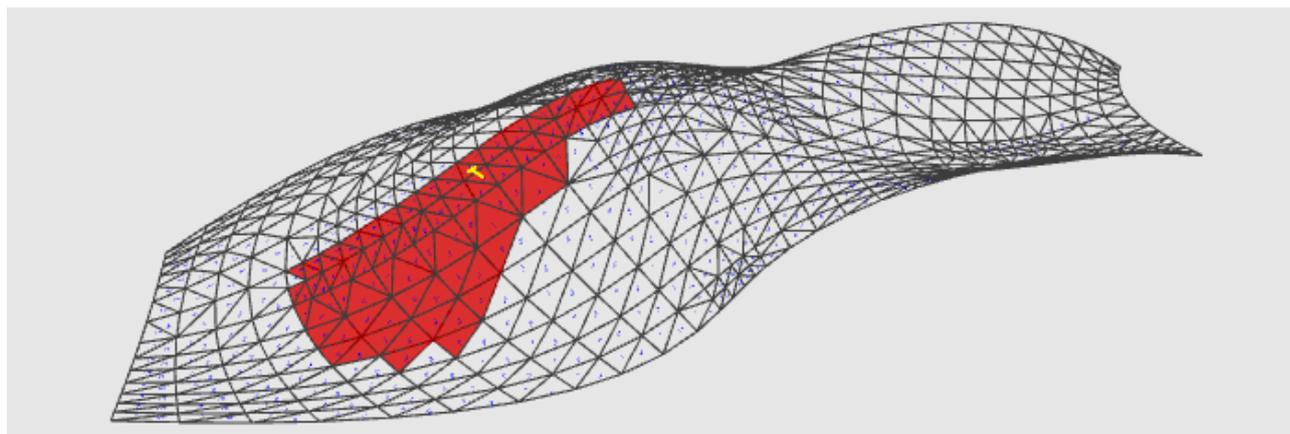


FIGURE A.1. – Exemple de maillage triangulaire, avec une sous-partie extraite en rouge

- pour tout point p_i tel que p_i est un point appartenant à un des segments de R , alors $\forall t \in T, p_i \notin t$ (cela revient à dire qu'un segment appartenant à l'ensemble des segments permettant de relier les points p_1 et p_2 ne peut pas traverser une maille de l'objet) ;
- pour tout segment de R , alors chacune des deux extrémités de R est soit p_1 , soit p_2 , soit présente en tant qu'extrémité chez un autre segment de R (cela signifie que les segments sont soit les extrémités de la chaîne, soit reliés à un autre segment).

On comprend aisément qu'un objet fermé est une surface 3D et pour laquelle il existe au moins un couple de points qui ne respecte pas la condition explicitée précédemment. Il serait facile de prouver que, l'espace étant continu, s'il existe un couple de points ne respectant pas la règle présentée plus haut, alors il en existe une infinité.

A.3. Les caractéristiques des descripteurs

Le choix du descripteurs est un point essentiel de l'étude proposée. En effet, ces derniers se doivent de respecter certaines contraintes afin d'être les plus efficaces possibles. Les contraintes d'invariance admises sont les suivantes :

- robustesse aux changements d'échelle (ou homothétie) : cela signifie que deux objets de même forme et de mêmes proportions, mais ayant des tailles différentes donneront le mêmes descripteur ; autrement dit, quelle que soit la norme du repère, les descripteurs donneront les mêmes résultats ;
- robustesse aux rotations : quelle que soit l'orientation du repère, les descripteurs donneront les mêmes résultats ;
- robustesse aux translations : quelle que soit l'origine du repère, les descripteurs donneront les mêmes résultats.

Cela revient à exiger l'indépendance du descripteur choisi vis-à-vis du repère, il ne peut donc dépendre que de la forme de l'objet et de ses proportions.

B. Formations et publications

B.1. Formations suivies et écoles d'été/hiver

- 18 et 19 Janvier 2016 : Ecole d'Hiver é-EGC, sur le thème "Pattern Mining"
- 23 et 24 Janvier 2017 : Ecole d'Hiver é-EGC, sur le thème "Web Sémantique"
- 22-27 aout 2016 : Vision and Sports Summer School (Prague) - "Computer Vision"

B.2. Publications réalisées

- Meunier, F., Marsala, C., and Castanié, L. (2015). *Development of a recommendation system for decision support in geosciences*. In MATHIAS – TOTAL Symposium on Mathematics, Paris, France.
- Meunier, F., Marsala, C. et Castanié, L. (2016). *Heat Kernel Signature pour la sélection et la classification d'objets - application aux géo-modèles 3D*. La 5ème édition des journées "Big Data Mining and Visualization", Metz, France, 23-24 juin 2016
- Meunier, F., Marsala, C., Castanié, L., et Conche, B. (2017). *Classification d'objets 3D par extraction aléatoire de sous-parties discriminantes pour l'étude du sous-sol en prospection pétrolière*. *Revue des Nouvelles Technologies de l'Information, Extraction et Gestion des Connaissances, RNTI-E-33* : p 225–236, sélectionné pour les post-actes internationaux chez Springer en version étendue (à venir).
- Meunier, F., Marsala, C., et Castanié, L. (2017), *3DRESC-TF : Apprentissage par transfert pour la réutilisation de connaissances en classification d'objets 3D*. pages 11-18. PFIA / APIA : Conférence Nationale sur les Applications Pratiques de l'Intelligence Artificielle, 3-4 juillet 2017.
- Meunier F., Marsala, C., De Runz, C., Castanié, L. (2017). *Sélection d'attributs pour la classification d'objets 3D*. pages 83-90, LFA 2017 - 26èmes Rencontres Francophones sur la Logique Floue et ses Applications, Amiens, France (2017)
- Meunier, F., Marsala, C., and Castanié, L. (2018), *Mining 3D-Structures : Subparts Extraction and Transfer Learning*. Proc. of the IEEE International Conference on Systems, Man, and Cybernetics (SMC'2018), Oct 2018, Miyazaki, Japan.

Bibliographie

- [Aherne et al., 1998] Aherne, F. J., Thacker, N. A., and Rockett, P. I. (1998). The bhattacharyya metric as an absolute similarity measure for frequency coded data. *Kybernetika*, 34(4) :363–368.
- [Alexandre, 2012] Alexandre, L. A. (2012). 3D descriptors for object and category recognition : a comparative evaluation. In *Workshop on Color-Depth Camera Fusion in Robotics at the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, pages 1–6.
- [Ankerst et al., 1999] Ankerst, M., Kastenmüller, G., Kriegel, H.-P., and Seidl, T. (1999). 3d shape histograms for similarity search and classification in spatial databases. In *Proceedings of the 6th International Symposium on Advances in Spatial Databases, SSD '99*, pages 207–226, London, UK. Springer-Verlag.
- [Béchet, 2011] Béchet, N. (2011). Etat de l’art sur les Systemes de Recommandation.
- [Besl and McKay, 1992] Besl, P. J. and McKay, N. D. (1992). A method for registration of 3-d shapes. *IEEE Trans. Pattern Anal. Mach. Intell.*, 14(2) :239–256.
- [Bhattacharyya, 1943] Bhattacharyya, A. (1943). On a measure of divergence between two statistical populations defined by their probability distributions. *Bulletin of Cal. Math. Soc.*, 35(1) :99–109.
- [Bouillot et al., 2014] Bouillot, F., Poncelet, P., and Roche, M. (2014). Classification of small datasets : Why using class-based weighting measures? In Andreassen, T., Christiansen, H., Cubero, J.-C., and Raś, Z. W., editors, *Foundations of Intelligent Systems*, pages 345–354, Cham. Springer International Publishing.
- [Breiman, 2001] Breiman, L. (2001). Random forests. *Mach. Learn.*, 45(1) :5–32.
- [Chandrashekar and Sahin, 2014] Chandrashekar, G. and Sahin, F. (2014). A survey on feature selection methods. *Computers and Electrical Engineering*, pages 16–28.
- [Chattopadhyay et al., 2012] Chattopadhyay, R., Sun, Q., Fan, W., Davidson, I., Panchanathan, S., and Ye, J. (2012). Multisource domain adaptation and its application to early detection of fatigue. *ACM Transactions on Knowledge Discovery from Data*, 6(4).
- [Comaniciu et al., 2003] Comaniciu, D., Ramesh, V., and Meer, P. (2003). Kernel-based object tracking. *IEEE Trans. Pattern Anal. Mach. Intell.*, 25(5) :564–575.
- [Cortes and Vapnik, 1995] Cortes, C. and Vapnik, V. (1995). Support-vector networks. *Mach. Learn.*, 20(3) :273–297.
- [Cotting et al., 2004] Cotting, D., Weyrich, T., Pauly, M., and Gross, M. H. (2004). Robust watermarking of point-sampled geometry. In *SMI*, pages 233–242. IEEE Computer Society.
- [Dang, 2014] Dang, Q. V. (2014). *Similarités dans des modèles BRep paramétriques : détection et applications*. PhD thesis, Université de Toulouse.

Bibliographie

- [Daume, 2007] Daume (2007). Frustratingly easy domain adaptation. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 256–263, Prague, Czech Republic. Association for Computational Linguistics.
- [Duan et al., 2012] Duan, L., Xu, D., and Chang, S.-F. (2012). Exploiting web images for event recognition in consumer videos : A multiple source domain adaptation approach. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 1338–1345. IEEE.
- [Dubuisson, 2010] Dubuisson, S. (2010). The computation of the Bhattacharyya distance between histograms without histograms. In *2nd International Conference on Image Processing Theory Tools and Applications (IPTA'10)*, pages 373–378. IEEE.
- [Fang et al., 2015] Fang, Y., Xie, J., Dai, G., Wang, M., Zhu, F., Xu, T., and Wong, E. (2015). 3D Deep Shape Descriptor. pages 2319–2328.
- [Fischler and Bolles, 1981] Fischler, M. A. and Bolles, R. C. (1981). Random sample consensus : A paradigm for model fitting with applications to image analysis and automated cartography. *Commun. ACM*, 24(6) :381–395.
- [Friedman, 1997] Friedman, J. H. (1997). On bias, variance, 0/1-loss, and the curse-of-dimensionality. *Data mining and knowledge discovery*, 1(1) :55–77.
- [Frome et al., 2004] Frome, A., Huber, D., Kolluri, R., Bulow, T., and Malik, J. (2004). Recognizing objects in range data using regional point descriptors. In *Proceedings of the European Conference on Computer Vision (ECCV)*.
- [Gama et al., 2014] Gama, J. a., Žliobaitė, I., Bifet, A., Pechenizkiy, M., and Bouchachia, A. (2014). A survey on concept drift adaptation. *ACM Comput. Surv.*, 46(4) :44 :1–44 :37.
- [Grabocka et al., 2014] Grabocka, J., Schilling, N., Wistuba, M., and Schmidt-Thieme, L. (2014). Learning time-series shapelets. In *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '14*, pages 392–401, New York, NY, USA. ACM.
- [Hilden, 1991] Hilden, J. (1991). The area under the roc curve and its competitors. *Medical Decision Making*, 11(2) :95–101. PMID : 1865785.
- [Hills et al., 2014] Hills, J., Lines, J., Baranauskas, E., Mapp, J., and Bagnall, A. (2014). Classification of time series by shapelet transformation. *Data Min. Knowl. Discov.*, 28(4) :851–881.
- [Horn, 1984] Horn, B. K. P. (1984). Extended gaussian images. *Proceedings of the IEEE*, 72(2) :1671–1686.
- [Hou et al., 2016] Hou, L., Kwok, J. T., and Zurada, J. M. (2016). Efficient learning of time-series shapelets. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence, AAAI'16*, pages 1209–1215.
- [Huber, 2004] Huber, D. (2004). Parts-based 3D object classification.
- [Kazhdan et al., 2003] Kazhdan, M., Funkhouser, T., and Rusinkiewicz, S. (2003). Rotation invariant spherical harmonic representation of 3d shape descriptors. In *Proceedings of the 2003 Eurographics/ACM SIGGRAPH Symposium on Geometry Processing, SGP '03*, pages 156–164, Aire-la-Ville, Switzerland, Switzerland. Eurographics Association.

- [Knopp et al., 2010] Knopp, J., Prasad, M., Willems, G., Timofte, R., and Gool, L. J. V. (2010). Hough transform and 3d surf for robust three dimensional classification. In Daniilidis, K., Maragos, P., and Paragios, N., editors, *ECCV (6)*, volume 6316 of *Lecture Notes in Computer Science*, pages 589–602. Springer.
- [Kruskal, 1952] Kruskal, W. H. (1952). A nonparametric test for the several sample problem. *Ann. Math. Statist.*, 23(4) :525–540.
- [Kullback and Leibler, 1951] Kullback, S. and Leibler, R. A. (1951). On information and sufficiency. *Annals of Mathematical Statistics*, 22(1) :79–86.
- [LeCun and Bengio, 1998] LeCun, Y. and Bengio, Y. (1998). The handbook of brain theory and neural networks. chapter Convolutional Networks for Images, Speech, and Time Series, pages 255–258. MIT Press, Cambridge, MA, USA.
- [Lee, 1991] Lee, J. M. (1991). *Riemannian Manifolds : an introduction to curvature*. Springer.
- [Lesot, 2005] Lesot, M.-J. (2005). Similarity, typicality and fuzzy prototypes for numerical data. *Res-Systemica*, (5).
- [Lesot et al., 2008] Lesot, M.-J., Rifqi, M., and Bouchon-Meunier, B. (2008). Fuzzy prototypes : From a cognitive view to a machine learning principle. In Bustince, H., Herrera, F., and Montero, J., editors, *Fuzzy Sets and Their Extensions : Representation, Aggregation and Models*, pages 431–452. Springer.
- [Leung and Malik, 2001] Leung, T. and Malik, J. (2001). Representing and recognizing the visual appearance of materials using three-dimensional textons. *International Journal of Computer Vision*, 43(1) :29–44.
- [Li et al., 2012] Li, F., Pan, S. J., Jin, O., Yang, Q., and Zhu, X. (2012). Cross-domain co-extraction of sentiment and topic lexicons. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics : Long Papers - Volume 1*, ACL '12, pages 410–419, Stroudsburg, PA, USA. Association for Computational Linguistics.
- [Li et al., 2017] Li, O., Liu, H., Chen, C., and Rudin, C. (2017). Deep learning for case-based reasoning through prototypes : A neural network that explains its predictions. *CoRR*, abs/1710.04806.
- [Lines et al., 2012] Lines, J., Davis, L. M., Hills, J., and Bagnall, A. (2012). *A Shapelet Transform for Time Series Classification*. KDD '12. ACM, New York, NY, USA.
- [Lozano Vega, 2015] Lozano Vega, G. (2015). *Image-based detection and classification of allergenic pollen*. Theses, Université de Bourgogne.
- [Matusita, 1955] Matusita, K. (1955). Decision rules, based on the distance, for problems of fit, two samples, and estimation. *Annals of Mathematical Statistics*, 26(4) :631–640.
- [Meunier et al., 2017a] Meunier, F., Marsala, C., and Castanié, L. (2017a). 3DRESC-TF : Apprentissage par transfert pour la réutilisation de connaissances en classification d'objets 3D. In *Conférence Nationale sur les Applications Pratiques de l'Intelligence Artificielle*, Caen, France.
- [Meunier et al., 2017b] Meunier, F., Marsala, C., Castanié, L., and Conche, B. (2017b). Classification d'objets 3d par extraction aléatoire de sous-parties discriminantes pour l'étude du sous-sol en prospection pétrolière. In *Conf. Int. Francophone sur l'Extraction et la Gestion des Connaissances (EGC)*, pages 225–236.

Bibliographie

- [Meunier et al., 2017c] Meunier, F., Marsala, C., De Runz, C., and Castanié, L. (2017c). Feature selection for 3D-object classification. In *26èmes Rencontres Francophones sur la Logique Floue et ses Applications (LFA'2017)*, pages 83–90, Amiens, France. Cépaduès Editions.
- [Mitra et al., 2004] Mitra, N. J., Gelfand, N., Pottmann, H., and Guibas, L. (2004). Registration of point cloud data from a geometric optimization perspective. In *Symposium on Geometry Processing*, pages 23–31.
- [Ng, 1997] Ng, A. Y. (1997). Preventing "overfitting" of cross-validation data. In *Proceedings of the Fourteenth International Conference on Machine Learning, ICML '97*, pages 245–253, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.
- [Pan and Yang, 2010] Pan, S. J. and Yang, Q. (2010). A survey on transfer learning. *IEEE Trans. on Knowl. and Data Eng.*, 22(10) :1345–1359.
- [Quinlan, 1986] Quinlan, J. R. (1986). Induction of decision trees. *Machine Learning*, 1(1) :81–106.
- [Raina et al., 2007] Raina, R., Battle, A., Lee, H., Packer, B., and Ng, A. Y. (2007). Self-taught learning : Transfer learning from unlabeled data. In *Proceedings of the 24th International Conference on Machine Learning, ICML '07*, pages 759–766, New York, NY, USA. ACM.
- [Refregier, 2001] Refregier, A. (2001). Shapelets : I. a method for image analysis. *Monthly Notices of the Royal Astronomical Society*, 338 :35.
- [Renard et al., 2015] Renard, X., Rifqi, M., and Detyniecki, M. (2015). Random-shapelet : an algorithm for fast shapelet discovery. In *IEEE Int. Conf. on Data Science and Advanced Analytics (DSAA'2015)*, pages 1–10.
- [Renard et al., 2016] Renard, X., Rifqi, M., Fricout, G., and Detyniecki, M. (2016). EAST representation : fast discovery of discriminant temporal patterns from time series. In *ECML/PKDD Workshop on Advanced Analytics and Learning on Temporal Data*.
- [Rifqi, 1996] Rifqi, M. (1996). *Mesures de comparaison, typicalite et classification d'objets flous : théorie et pratique*. PhD thesis, Université Pierre et Marie Curie, Paris 6.
- [Rifqi, 2010] Rifqi, M. (2010). *Mesures de similarité, raisonnement et modélisation de l'utilisateur*. Technical report, Université Pierre et Marie Curie, Paris 6. Habilitation à diriger des recherches.
- [Rijsbergen, 1979] Rijsbergen, C. J. V. (1979). *Information Retrieval*. Butterworth-Heinemann, Newton, MA, USA, 2nd edition.
- [Rusu et al., 2008] Rusu, R. B., Blodow, N., Marton, Z. C., and Beetz, M. (2008). Aligning Point Cloud Views using Persistent Feature Histograms. In *Proceedings of the 21st IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Nice, France, September 22–26*.
- [Shaiek, 2013] Shaiek, A. (2013). *3D object recognition with points of interest*. These, Ecole Nationale Supérieure des Mines de Paris.
- [Shalom et al., 2008] Shalom, S., Shapira, L., Shamir, A., and Cohen-Or, D. (2008). Part Analogies in Sets of Objects. In Perantonis, S., Sapidis, N., Spagnuolo, M., and Thalmann, D., editors, *Eurographics 2008 Workshop on 3D Object Retrieval*. The Eurographics Association.

- [Shannon and Weaver, 1963] Shannon, C. E. and Weaver, W. (1963). *A Mathematical Theory of Communication*. University of Illinois Press, Champaign, IL, USA.
- [Sun et al., 2009] Sun, J., Ovsjanikov, M., and Guibas, L. (2009). A concise and provably informative multi-scale signature based on heat diffusion. In *Symposium on Geometry Processing, SGP'09*, pages 1383–1392. Eurographics Association.
- [Sun and Ponce, 2013] Sun, J. and Ponce, J. (2013). *Learning Discriminative Part Detectors for Image Classification and Cosegmentation*. PhD thesis.
- [Tabia, 2011] Tabia, H. (2011). *Contributions to 3D-shape matching retrieval and classification*. PhD thesis, Université Lille 1.
- [Tombari et al., 2010] Tombari, F., Salti, S., and Di Stefano, L. (2010). Unique shape context for 3d data description. In *ACM Workshop on 3D Object Retrieval, 3DOR '10*, pages 57–62, New York, NY, USA. ACM.
- [Vranic and Saupe, 2001] Vranic, D. V. and Saupe, D. (2001). 3D Shape Descriptor Based on 3D Fourier Transform. In *Proceedings of the EURASIP Conference on Digital Signal Processing for Multimedia Communications and Services (ECMCS 2001)* (editor K. Fazekas), pages 271–274, Budapest, Hungary.
- [Wei et al., 2018] Wei, Y., Zhang, Y., Huang, J., and Yang, Q. (2018). Transfer learning via learning to transfer. In Dy, J. and Krause, A., editors, *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 5085–5094, Stockholmsmässan, Stockholm Sweden. PMLR.
- [Weiss et al., 2016] Weiss, K., Khoshgoftaar, T. M., and Wang, D. (2016). A survey of transfer learning. *Journal of Big Data*, 3(1) :9.
- [Wistuba et al., 2015] Wistuba, M., Grabocka, J., and Schmidt-Thieme, L. (2015). Ultra-fast shapelets for time series classification. *CoRR*, 1503.05018.
- [Yager, 1980] Yager, R. (1980). A general class of fuzzy connectives. 4 :235–242.
- [Yager, 1988] Yager, R. R. (1988). On ordered weighted averaging aggregation operators in multicriteria decisionmaking. *IEEE Trans. Syst. Man Cybern.*, 18(1) :183–190.
- [Ye and Keogh, 2009] Ye, L. and Keogh, E. (2009). Time Series Shapelets : A New Primitive for Data Mining. In *15th ACM SIGKDD Int. Conf. on Knowl. Discovery and Data Mining, KDD '09*, pages 947–956.
- [Zakaria et al., 2012] Zakaria, J., Mueen, A., and Keogh, E. (2012). Clustering time series using unsupervised-shapelets. In *Proceedings of the 2012 IEEE 12th International Conference on Data Mining, ICDM '12*, pages 785–794, Washington, DC, USA. IEEE Computer Society.
- [Zhang et al., 2018] Zhang, H., Chen, P.-H., and Ramadge, P. (2018). Transfer learning on fmri datasets. In Storkey, A. and Perez-Cruz, F., editors, *Proceedings of the Twenty-First International Conference on Artificial Intelligence and Statistics*, volume 84 of *Proceedings of Machine Learning Research*, pages 595–603, Playa Blanca, Lanzarote, Canary Islands. PMLR.