

Cycle des Ingénieurs diplômés de l'ENSG 3ème année

Projet Alternance

- Paul Guardiola -

Septembre 2023

Insitut National de l'Information géographique et forestière

Table des matières

1	Introduction	3
2	Etat de l’art	4
3	Algorithme d’appariement	5
3.1	Introduction	5
3.2	Appariement surfacique	5
3.2.1	Pre appariement surfacique	5
3.2.2	Recherche groupes optimaux	7
3.2.3	Filtres liens	7
3.3	Algorithme MultiCritère	8
3.3.1	Le choix des candidats	8
3.3.2	L’initialisation des masses de croyance	8
3.3.3	Fusion des critères	9
3.3.4	Fusion des candidats	11
3.3.5	Décision	13
4	Analyse de donnée	14
4.1	Introduction	14
4.2	Définition des données dites de sortie	14
4.3	Algorithme d’appariement surfacique	15
4.3.1	Analyse de sensibilité pour l’algorithme d’appariement surfacique . . .	15
4.3.2	Méthode OAT	16
4.3.3	Méthode de Monte-Carlo	17
4.3.4	Cas particulier de pourcentage_intersection_sur	18
4.4	Algorithme MultiCritère	19
4.4.1	Recherche des seuils optimaux pour les masses de croyance et générali- sation	19

1 Introduction

L'appariement s'intéresse au niveau sémantique de l'information géographique, c'est à dire aux attributs qui viennent compléter la représentation d'un objet. Elle est la filiation entre deux objets de différentes sources représentant le même objet dans la réalité. L'appariement permet alors d'enrichir une base de donnée grâce à plusieurs jeux de données décrivant la même réalité.

L'analyse de données est une discipline qui a pour objectif de tirer des informations de données. Souvent l'analyse s'exerce sur des données obtenues en 'sortie', c'est-à-dire les résultats d'une expérience, d'un modèle au vue de comprendre la variation de ces données.

2 Etat de l'art

3 Algorithme d'appariement

3.1 Introduction

Différents types d'algorithmes d'appariement ont été créés ces dernières années. Dans ce projet, nous avons étudié deux algorithmes en particulier : l'algorithme d'appariement surfacique et l'algorithme MultiCritère. L'algorithme d'appariement surfacique est le plus ancien des deux, il a été utilisé il y a quelques années pour appairer les bases de données de l'IGN et les bases de données open source comme OpenStreetMap. L'algorithme d'appariement MultiCritère a notamment été utilisé dans le projet choucas pour l'appariement en montagne. La rareté des objets présents dans les bases de données sur les objets en montagne en font un excellent domaine pour l'appariement. L'algorithme a notamment permis d'enrichir les bases de données incomplètes en croisant plusieurs sources de bases de données. Dans un cas général, pour exécuter un algorithme d'appariement, il est nécessaire de posséder au moins deux jeux de données, un de référence et un ou plusieurs autres à appairer au premier. Les deux jeux de données, au minimum, doivent être similaires dans la forme, traduire les mêmes objets mais pas nécessairement de la même réputation ou de qualité. L'algorithme va permettre de détecter les objets dans les bases de données qui représentent la même entité dans la réalité. Dans les deux cas donc, l'étape finale sera la création d'un nouveau fichier sous le format shapefile. Ce shapefile contiendra les identifiants des bâtiments de référence et de leurs objets appariés. Chaque ligne représente un lien dans le fichier et chaque lien contient une géométrie avec un linestring entre les deux centroides des surfaces de référence et apparié.

3.2 Appariement surfacique

L'algorithme d'appariement de surface est un algorithme spécifique à l'appariement des objets surfaciques. Il ne se base que sur la proximité surfacique de deux bâtiments entre eux.

L'étape AjoutPetitesSurfaces n'est plus opérationnelle. En effet, elle avait été ajoutée pour les anciennes BDtopo très parsemées et permettait de choisir la surface à appairer entre les petites surfaces précédemment appariées. *Param['ajoutpetitesurfaces']* est donc initialisé *False* pour l'ensemble du projet.

3.2.1 Pre appariement surfacique

L'algorithme de pre-appariement permet de prendre le plus large nombre de candidats à appairer pour une surface donnée. La première étape, si nous prenons une surface de référence, nous récoltons tous les objets qui s'intersectent à la surface de référence ou à sa forme déformée par Douglas-Peucker sur 10 itérations. Par la suite deux conditions doivent être réunies pour qu'un objet puisse être gardé et par la suite étudié plus profondément pour être apparié. Ces deux conditions sont qu'il faut que la surface de l'intersection doit être supérieure au paramètre *surface_min_intersection* et le pourcentage du recouvrement doit être inférieur au *recouvrement_min_intersection*. Si ces conditions sont respectées un lien provisoire sera créé entre les deux objets.

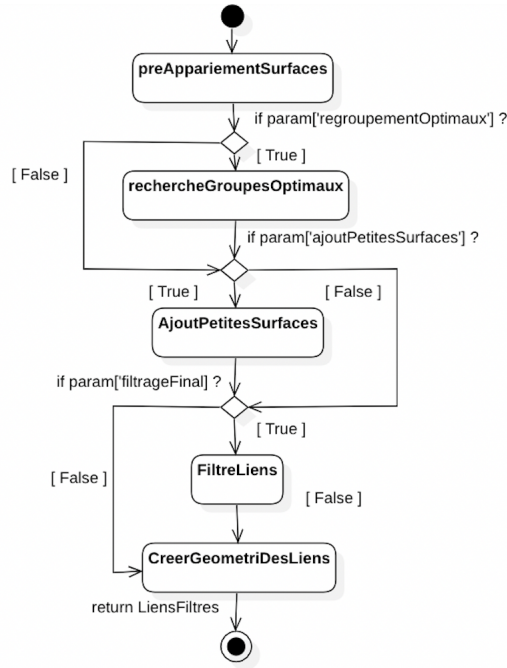


FIGURE 1 – Diagramme d'activité de l'algorithme d'appariement de surface

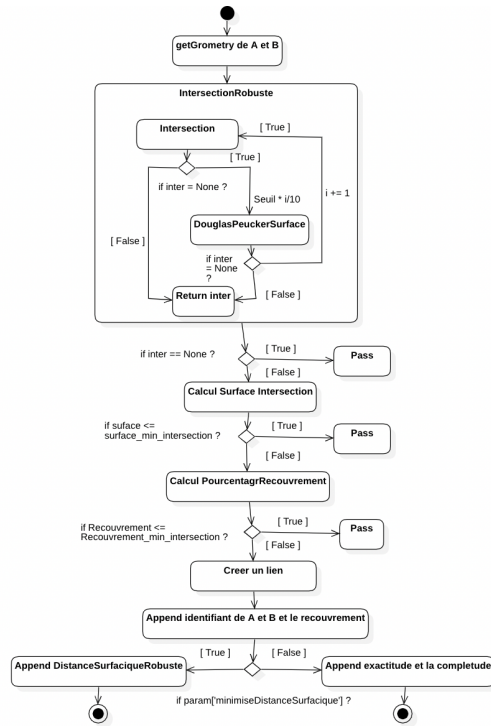


FIGURE 2 – Diagramme d'activité de l'algorithme de pre-appariement de surface

3.2.2 Recherche groupes optimaux

Une fois la première étape exécutée, nous allons passer à l'étape *Recherche groupes optimaux* pour ne garder que les liens les plus significatifs.

3.2.3 Filtres liens

3.3 Algorithme MultiCritère

L'algorithme multiCritère se base sur la théorie de la décision et notamment sur la théorie de Dempster-Shafer. Il permet en 5 temps d'apparier une base de donnée à une autre. Le premier temps est le choix des candidats, le second c'est l'initialisation des masses de croyance, la fusion des critères, la fusion des candidats et la décision.

3.3.1 Le choix des candidats

Le choix des candidats se fait simplement par une requête d'intersection autour du bâtiment de référence. En effet, l'appariement de base de donnée géographique se fait fatalement entre deux objets proche dans l'espace. Dans l'idéale trois candidats pour un objet est un bon niveau d'intersection.

Exemple

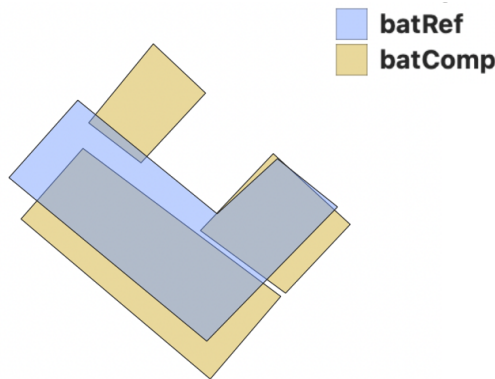


FIGURE 3 – Bâtiments de référence et à apparier retenue

Dans l'exemple ci-dessus, les batiments Comp sont retenue pour être apparier au bâtiment de référence.

3.3.2 L'initialisation des masses de croyance

Ici, nous donnons à chaque critère l'importance qu'il aura pas rapport aux autres. Nous comptons à ce jour plusieurs critères. Il y a les critères géographique (distance euclidienne, distance d'Hausdorff, distance radiale, orientation) et les critères attributaires (critère onthologique, critère textuel).

Critère radiale

Le critère radiale mesure la proximité de la forme entre deux entités. Celui-ci se base sur la différence de la signature radiale des deux formes. La distance se calcule ensuite avec cette différence.

Critère textuel

Pour évaluer la proximité textuelle entre deux objets nous pouvons utiliser une distance sémantique comme la distance de Samal. Pour deux chaînes de caractère, les espaces, les ponctuations, majuscules, tirets et underscore sont supprimés. Nous dressons une matrice avec en colonne le nombre de mots d'une des deux chaînes et les lignes pour le nombre des mots de l'autre.

Critère sémantique

Le critère sémantique se base sur la similarité sémantique entre deux objets. Celle-ci se définit par la proximité ou non sur la nature des objets grâce à une ontologie qui hiérarchise la nature des objets. La distance caractérisant la similarité sémantique se nomme la distance de Wu-Palmer défini ci-dessous :

Orientation

Création du critère d'orientation qui va permettre de répondre au problème des bâtiments reconstruit qui s'appartient aux bâtiments de référence. En effet, nous pouvons espérer que si un bâtiment a été reconstruit l'orientation soit différente entre les deux.

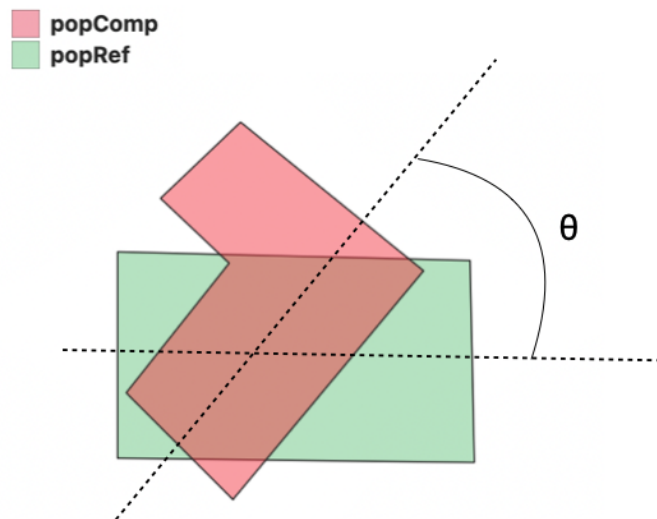


FIGURE 4 – Schéma de l'orientation entre deux bâtiments

3.3.3 Fusion des critères

Pour chaque candidat, les critères qui expriment la proximité des candidats à une même référence vont être fusionnés en trois masses pour définir cette proximité. Les trois masses traduisent trois hypothèses : l'appariement, le non-appariement et le 'on-ne-sait-pas' ($C_i, \neg C_i, \theta$). Nous noterons $m_1(C_i), m_1(\neg C_i), m_1(\theta)$

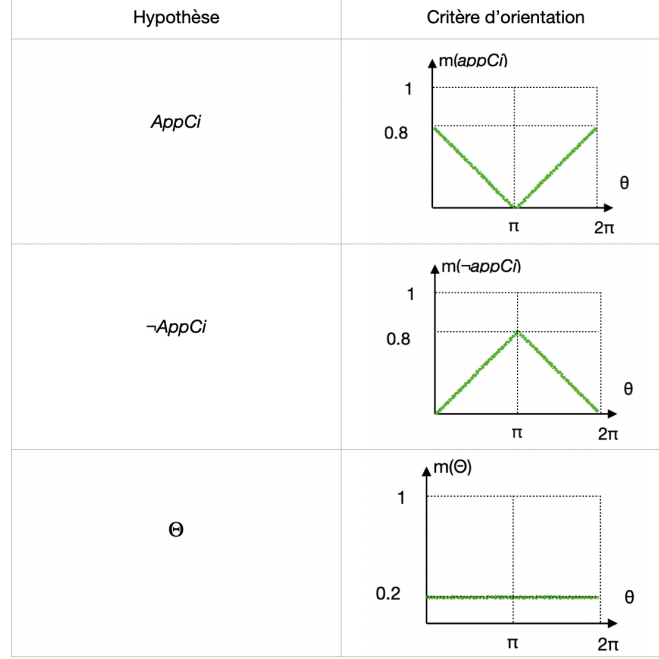


FIGURE 5 – Représentation des connaissances pour le critère d'orientation

les masses du critère 1 du candidat i pour les différentes hypothèses. Nous utilisons par la suite un opérateur conjonctif pour fusionner les masses pour n'obtenir que trois masses par candidat. L'associativité de la règle de combinaison conjonctive permet d'ajouter autant de critères que possible et permet *in fine* une meilleure décision.

	app_2	$\neg app_2$	θ_2
app_1	app	ϕ	app
$\neg app_1$	ϕ	$\neg app$	$\neg app$
θ_1	app	$\neg app$	θ

$$\begin{aligned}
m_{12}(app) &= m(app_1) * m(app_2) + m(app_1) * m(\theta_2) + m(app_2) * m(\theta_1) \\
m_{12}(\neg app) &= m(\neg app_1) * m(\neg app_2) + m(\neg app_1) * m(\theta_2) + m(\neg app_2) * m(\theta_1) \\
m_{12}(\theta) &= m(\theta_1) + m(\theta_2) \\
m_{12}(\phi) &= m(app_1) * m(\neg app_2) + m(app_2) * m(\neg app_1)
\end{aligned}$$

	app_3	$\neg app_3$	θ_3
$m_{12}(app)$	app	ϕ	app
$m_{12}(\neg app)$	ϕ	$\neg app$	$\neg app$
$m_{12}(\theta)$	app	$\neg app$	θ

$$\begin{aligned}
m_{123}(app) &= m_{12}(app) * m(app_3) + m_{12}(app) * m(\theta_3) + m(app_3) * m_{12}(\theta) \\
m_{123}(\neg app) &= m_{12}(\neg app) * m(\neg app_3) + m_{12}(\neg app) * m(\theta_3) + m(\neg app_3) * m_{12}(\theta) \\
m_{123}(\theta) &= m_{12}(\theta) + m(\theta_3) \\
m_{123}(\phi) &= m_{123}(app) * m(\neg app_3) + m(app_3) * m_{123}(\neg app)
\end{aligned}$$

Remarque

Dans des cas particulier, la somme des masses de croyances est inférieure à 1. Ce phénomène arrive lorsque les masses de croyances ne mettent pas tous en avant la même hypothèse.

critère1 = (0.4, 0, 0.6); critère2 = (0.3, 0, 0.7); critère3 = (0.5, 0.2, 0.3);
critère4 = (0.5, 0.1, 0.4)

$$m_{123}(app) = 0.6696$$

$$m_{123}(\neg app) = 0.0546$$

$$m_{123}(\theta) = 0.0504$$

$$m_{123}(\phi) = 0.1094$$

On obtient $\sum_{P \in \{app, \neg app, \theta, \phi\}} m_{123}(P) = 0.884 \neq 1$

3.3.4 Fusion des candidats

À l'issue de l'étape précédente, nous obtenons pour chaque candidat trois masses pour les hypothèses exprimées par rapport à l'objet de référence. Pour rappel il faut que l'objet choisi correspond à l'objet de référence car dans la réalité il sont le même objet. Pour cela, nous n'allons pas simplement prendre le candidat qui a une masse la plus forte mais utilisé la Théorie de la Décision et en particulier la Théorie de Dempster-Shafer. La théorie de Dempster-Shafer se base sur l'utilisation des fonctions de croyances. Une fois la combinaison de Dempster exécuté dans la partie sur la fusion des critères, nous calculons la probabilité pignistique qui à chaque hypothèse associe une probabilité. Les probabilités les plus intéressantes sont les probabilités d'appariement et d'ignorance.

La loi à les propriété suivante :

$$\begin{aligned}
C_X * C_Y &= \phi \\
C_X * \neg C_Y &= C_X \\
C_X * \theta &= C_X \\
C_X * \emptyset &= \emptyset \\
\neg C_X * \neg C_Y &= C_Z \cap NA \\
C_Z * \neg C_Z &= C_X \cap C_Y \cap NA
\end{aligned}$$

Nous pouvons retrouver des propriétés des lois traditionnelles comme :

Élément neutre θ : $C * \theta = C$

Élément nulle \emptyset : $C * \emptyset = \emptyset$

Idempotence : $C * C = C$

Nous calculons la probabilité pignistique :

$$m_{12}(A) = \sum_{B * C = A} m(B)m(C)$$

La fusion des candidats peut donc se faire de façon itérative. En effet, nous faisons en premier la probabilité entre le candidat 1 et 2 puis la fusion avec le candidats 3 etc.

	C_2	$\neg C_2$	θ_2	ϕ_2
C_1	ϕ	C_1	C_1	ϕ
$\neg C_1$	C_2	NA	$\neg C_1$	ϕ
θ_1	C_2	$\neg C_2$	θ	ϕ
ϕ_1	ϕ	ϕ	ϕ	ϕ

$$\begin{aligned}
m_{12}(C_1) &= m(C_1) * (m(\neg C_2) + m(\theta_2)) \\
m_{12}(\neg C_1) &= m(\neg C_1) + m(\theta_2) \\
m_{12}(C_2) &= m(C_2) * (m(\neg C_1) + m(\theta_2)) \\
m_{12}(\neg C_2) &= m(\neg C_2) + m(\theta_1) \\
m_{12}(\theta) &= m(\theta_1) + m(\theta_2) \\
m_{12}(\phi) &= m(\phi_1) * (m(C_2) + m(\neg C_2) + m(\theta_2)) + m(\phi_2) * (m(C_1) + m(\neg C_1) \\
&\quad + m(\theta_1))
\end{aligned}$$

	C_3	$\neg C_3$	θ_3	ϕ_3
$m_{12}(C_1)$	ϕ	C_1	C_1	ϕ
$m_{12}(\neg C_1)$	C_3	NA	$\neg C_1$	ϕ
$m_{12}(C_2)$	ϕ	C_2	C_2	ϕ
$m_{12}(\neg C_2)$	C_3	NA	$\neg C_2$	ϕ
$m_{12}(\theta)$	C_3	$\neg C_3$	θ	ϕ
$m_{12}(\phi)$	ϕ	ϕ	ϕ	ϕ

$$\begin{aligned}
m_{123}(C_1) &= m_{12}(C_1) * (m(\neg C_3) + m(\theta_3)) \\
m_{123}(\neg C_1) &= m_{12}(\neg C_1) + m(\theta_3) \\
m_{123}(C_2) &= m_{12}(C_2) * (m(\neg C_3) + m(\theta_3)) \\
m_{123}(\neg C_2) &= m_{12}(\neg C_2) + m(\theta_3) \\
m_{123}(C_3) &= m(C_3) * (m_{12}(\neg C_1) + m_{12}(\neg C_2) + m_{12}(\theta)) \\
m_{123}(\neg C_3) &= m(\neg C_3) + m_{12}(\theta) \\
m_{123}(\theta) &= m_{12}(\theta) + m(\theta_3) \\
m_{123}(\phi) &= m_{12}(\phi) * (m(C_3) + m(\neg C_3) + m(\theta_3) + m(\phi_3)) \\
&\quad + m(\phi_3) * (m(C_3) + m(\neg C_3) + m(\theta_3)) \\
m_{123}(NA) &= m(\neg C_3) * (m_{12}(\neg C_1) + m_{12}(\neg C_2))
\end{aligned}$$

Nous pouvons continuer ces itérations pour plusieurs candidats pour trouver la probailité pignistique de chaque hypothèse.

3.3.5 Décision

Une fois toutes les probabilités pignistique obtenues, les probabilités sont normalisé par $k = 1/(1 - m_{1..n}(\phi))$. Ensuite la probabilité pignistique normalisée la plus élevée nous donne l'hypothèse retenue. La plus grande probabilité traduira le décision à prendre, si il faut apparier, ou pas et si oui, avec quel candidat.

4 Analyse de donnée

4.1 Introduction

L'analyse de sensibilité est définie par Salteli et al, 2008 comme décrivant « l'importance relative de chaque entrée dans la détermination de la variabilité des sorties » <https://openmole.org/Sensitivity.html> qualitativement ou quantitativement. Pour explorer ces systèmes, beaucoup trop long dans la vie réelle, nous utilisons des modèles mathématiques et informatique pour différentes natures (sociales, médicales ou urbaines).

Dans un modèle mathématiques, le système est composé d'équations, de facteurs d'entrée (input) et de paramètres. Les inputs sont souvent caractérisé par des erreurs ou des approximations qui doivent être investigué. Originellement, SA a été inventé pour traiter de ces incertitudes. SA est utilisé pour augemnter la confiance dans les models et leur prédictions. Donc, SA est liée au analyse d'incertitude.

4.2 Définition des données dites de sortie

L'analyse de sensibilité confronte deux types de données : les données dites d'entrée et de sortie. Les données d'entrées sont naturellement les paramètres utilisées par les algorithmes mais les données de sorties sont moins explicite. Toutefois les résultats permettent de faire émerger des indices sur la performances de l'appariement. C'est le cas des indices qui peuvent petre issue des matrices de confusion. Nous pourrons utiliser les indices de précision, du rappelle et du F score. Nous devons donc définir ce qui est un lien " faux positif", "positif positif" etc. Nous allons considérer que un "positif-positif" est lorsque soit un bâtiment de référence s'est bien apparellié avec un bâtiment test ou lorsque un bâtiment test n'a pas eu de référent et donc il n'y a pas eu de lien. Un "négatif-négatif" est un bâtiment qui aurait du s'apparier mais ne l'a pas été. Enfin un "Négatif-positif" est un lien qui aurait du se faire avec un bâtiment de référence dont le bâtiment de référence a déjà un lien. ce qui équivaut à une sous-détection. Enfin le "Positif-négatif" correspond à une surdétection, c'est à dire qu'il existe un lien entre deux bâtiment qui ne représente pas le même bâtiment dans la réalité.

En calculant le nombre des cas suivant, nous pouvons calculer les indices de recall, précision et F score. Ceci seront nos paramètres de sortie qui pourront être étudié en fonction des paramètres d'entrée pour faire une analyse de sensibilité.

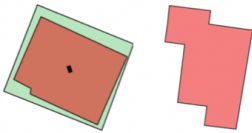
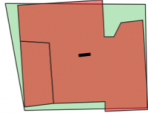
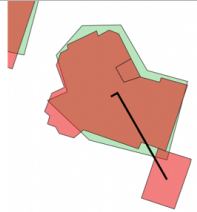

	Positif (P)	Negative (N)
Positive (P)		
Negative (N)		

FIGURE 6 – Matrice de confusion

En dehors des indices issues de la matrice de confusion, nous pourrions regarder le temps d'exécution.

4.3 Algorithme d'appariement surfacique

4.3.1 Analyse de sensibilité pour l'algorithme d'appariement surfacique

La première étape est de lister les données de sortie avec leurs intervalles. Dans notre algorithme, il y a 10 paramètres d'entrée.

Paramètre	valeur par défaut	intervalle de variation
surface_min_intersection	1	$[0 ; +\infty]$
pourcentage_min_intersection	0	$[0 ; 1]$
pourcentage_intersection_sur	0.8	$[0 ; 1]$
minimiseDistanceSurfacique	True	True or False
distSurfMaxFinal	0.6	$[0 ; 1]$
completudeExactitudeMinFinal	0.3	$[0 ; +\infty]$
regroupementOptimal	True	True or False
filtrageFinal	True	True or False
resolutionMin	1	$] 0 ; +\infty [$
resolutionMax	11	$] resolutionMin ; +\infty [$

Par la suite ces paramètres devront être comparé à des paramètres de sortie.

4.3.2 Méthode OAT

Dans un premier temps on va faire l'étude de 5 paramètres : `surface_min_intersection`, `pourcentage_min_intersection`, `pourcentage_intersection_sur`, `distSurfMaxFinal` et `completudeExactitudeMinFinal`.

La methode AOT permet ensuite d'évaluer l'influence d'une variable d'entrée sur la sortie. Pour cela, on fait varier un paramètre dans son intervalle en prenant toutes les autres valeurs par défaut.

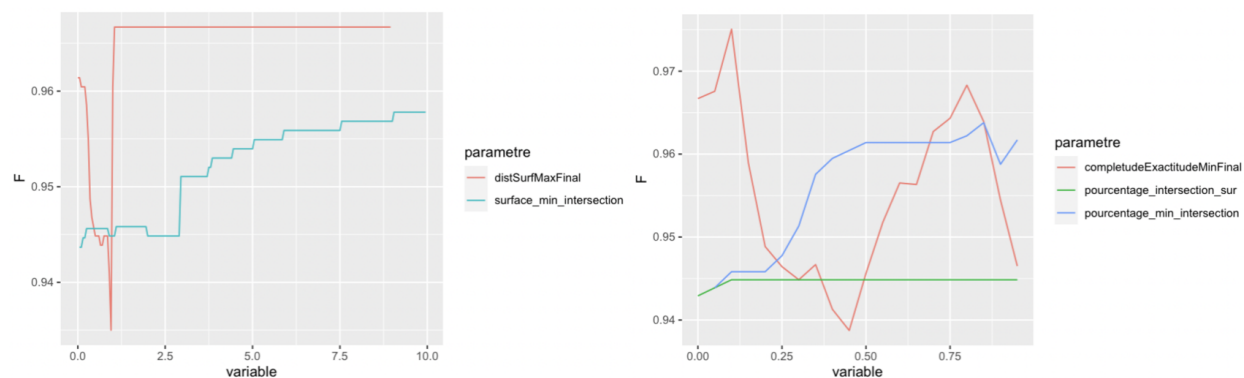


FIGURE 7 – Courbe du F score en fonction de la `distSurfMax`, `SurfMinIntersection`, la `completudeExactitude`, `PourIntersectionSurf` et `pourMinIntersection` sur une zone pavillonnaire

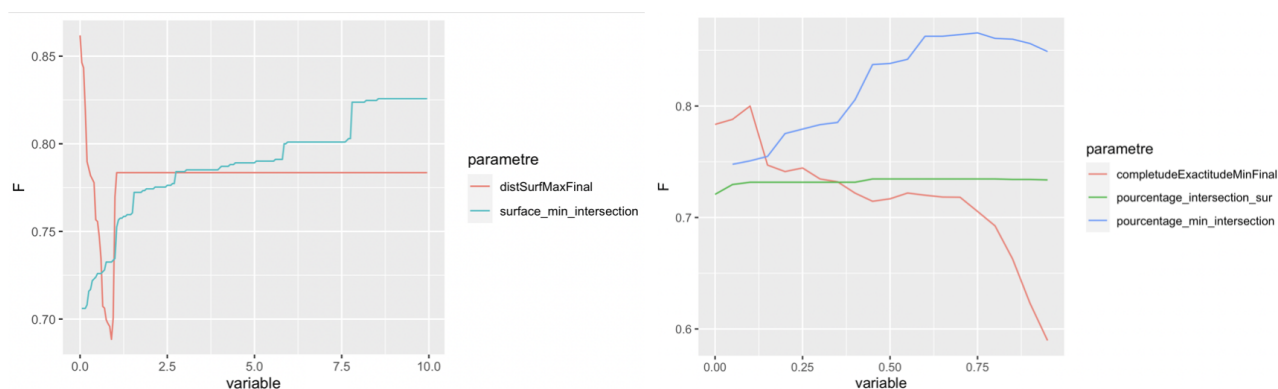


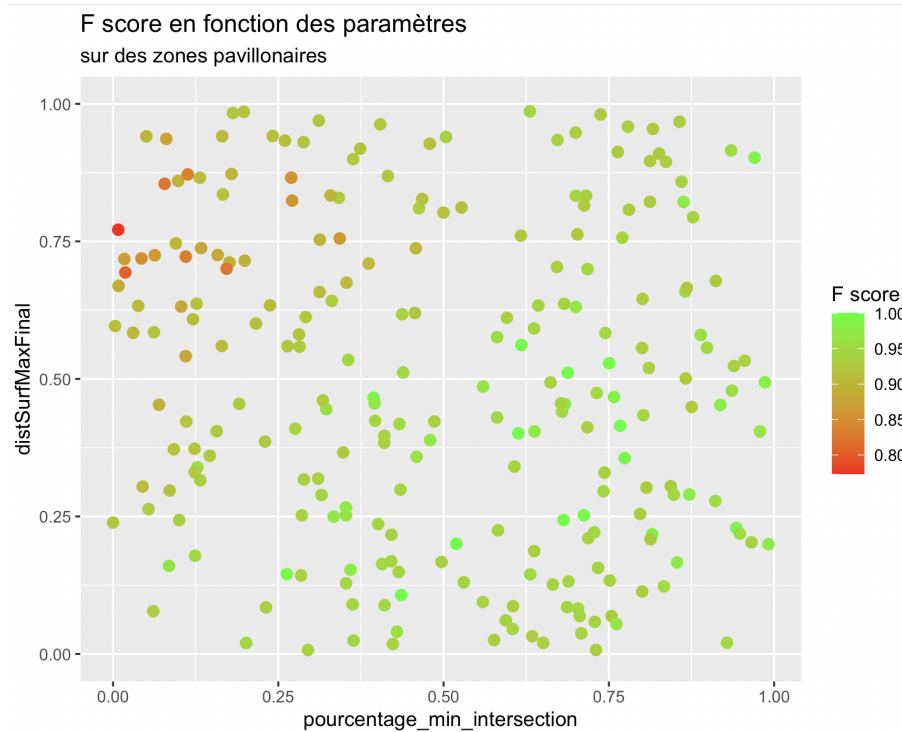
FIGURE 8 – Courbe du F score en fonction de la `completudeExactitude`, `PourIntersectionSurf` et `pourMinIntersection` sur une zone urbaine

Ces courbes sont moyenné sur 5 paysages types de ville ou de zone pavillonnaire avec plus d'une centaine de bâtiments pour chaque zone.

Nous observons sur cette figure que chaque paramètre a une influence plus ou moins importante sur la valeur du F score en sortie et nous pouvons souligner quelque dynamique sur les variations. Par exemple, `distSurfMaxFinal` semble faire décroître le F score. De même on peut retrouver le même dynamisme pour différent jeu de donnée. Par forcément centré sur les mêmes valeurs de F score mais présentant les mêmes courbes.

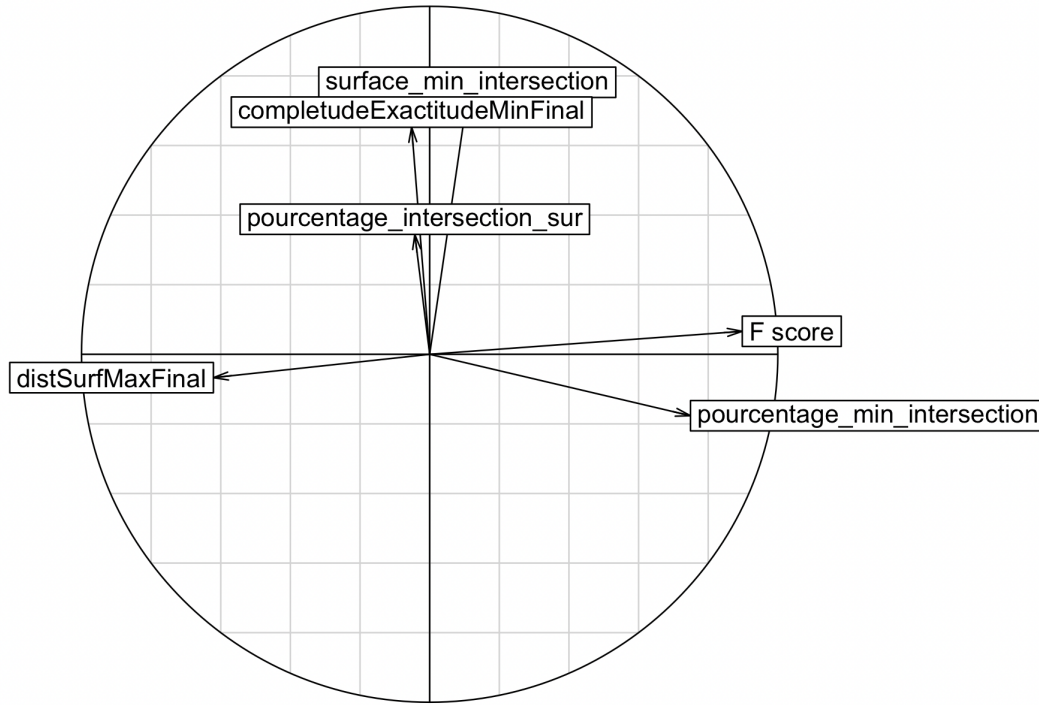
4.3.3 Méthode de Monte-Carlo

La méthode de Monte-Carlo consiste à faire varier l'ensemble des paramètres de façon aléatoire. Ceci, nous permet de souligner le dynamisme entre les variables et voir si deux variables ne seraient pas couplé.



On peut ensuite voir des relations particulière entre les données comme le lien entre la `distSurfMaxFinal` et le `pourcentage_min_intersection`. On remarque que lorsque nous avons des hautes valeurs de poucentage pour des basse valeur de `distSurfMaxFinal` on obtient les meilleurs F score. D'une façon générale n peut faire une étude ACP pour voir l'influence des paramètres sur le F score.

Avec ce cercle des corrélations nous pouvons noter plusieurs informations. Premièrement, les paramètres qui captent le plus d'informations sont le `pourcentage_min_intersection`, la `surface_min_intersection` et la `distSurfMaxFinal`.



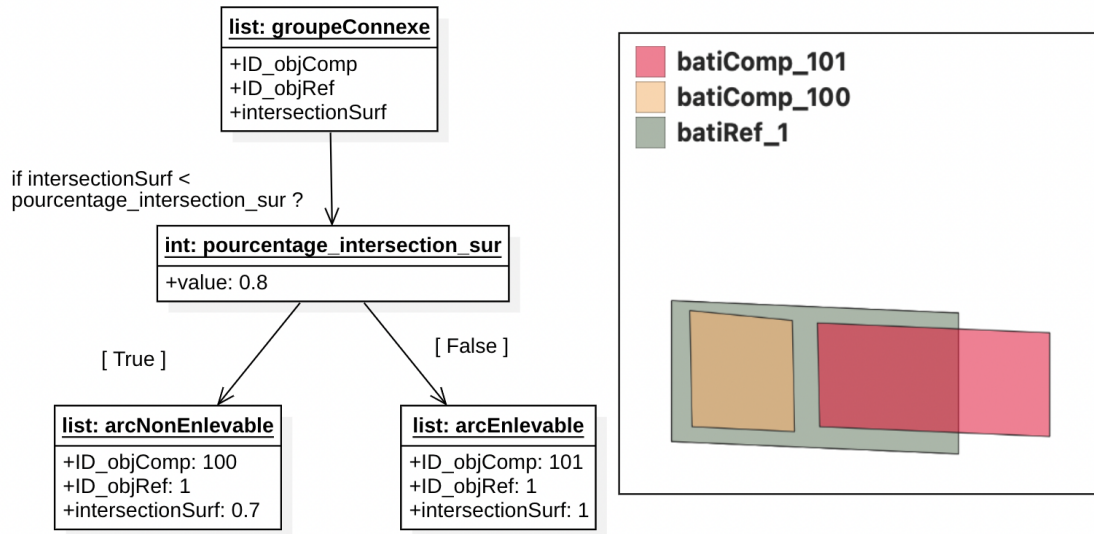
De plus nous voyons que `pourcentage_intersection_sur`, `CompExactitudeMinFinal` et `surf_min_intersection` ne sont pas corréllé au F score. La `distSurfMaxFinal` est anti-corréllé au F score et le `pourcentage_min_intersection` et lui quasi corréllé au F score.

4.3.4 Cas particulier de `pourcentage_intersection_sur`

Le paramètre `pourcentage_intersection_sur` intervient dans l'algorithme `RechercheGroupeOptimaux` qui cherche à réduire le nombre d'appariement entre un bâtiments de la base de données de référence et plusieurs bâtiments de la base de données à apparier.

Ensuite, on se concentre sur la liste `arcEnlevable`. Dans la suite on va considérer que les objets de cette liste sont enlevé si la distance surfacique de l'objet est inférieur à 2 pour le premier puis à chaque distance surfacique minimale. C'est valeurs sont fixé dans l'algorithme est donc ne sont peut être pas adapté à tous les jeux de données. On remarque que pour notre jeu de données les distance surfacique des objets présents dans la liste `arcEnlevable` vont de $[0.3; 1]$, sans jamais atteindre la valeur 1.

La distance surfacique représente la distance avec l'union de l'ensemble des petites surfaces avec le bâtiment des surfaces et l'algorithme renvoie cette valeur comprise entre $[0;1]$ donc fatalement inférieur à 2.



En plus de cela, la différence entre les deux il y a une différence de 16 objets c'est à dire que sur 950 appariements seulement 16 ont été enlevé ce qui n'est pas du tout énorme et explique pourquoi le F score est quasi constant. On peut donc conclure que le fait que ce soit constant est la faible valeur de `pourcentage_intersection_sur`.

4.4 Algorithme MultiCritère

4.4.1 Recherche des seuils optimaux pour les masses de croyance et généralisation

Premier test sur la vérité terrain.

Critère	Euclidien	Hausdorff	radiale	orientation	nb erreurs
Test 1	X				2
Test 2		X			2
Test 3			X		38
Test 4				X	59
Test 5	X	X			2
Test 6	X		X		5
Test 7	X			X	6
Test 8	X	X	X		4
Test 9	X		X	X	7
Test 10	X	X	X	X	5

Nous remarquons que les masses n'ont pas toutes la même valeur. De même si on s'intéresse à la répartition