Université de Strasbourg

Rapport du Travail d'Etude et de Recherche

Validation de l'approche clustering sous contraintes

Etudiant :
Pambou Moubogha Eddy
année scolaire 2019-2020

Encadrants :
P. Gançarski
A. Braud

1er Janvier 2020 — 12 Mai 2020

Table des matières

Chapitre 1

Introduction

Les glissements de terrain apparaissent lorsqu'une masse de terre descend sur un plan de glissement, provoqués par les activités anthropiques ou des phénomènes climatiques, géologiques ou géomorphologiques. Ces déplacements peuvent être lents (quelques millimètres par an) ou rapides (quelques centaines de mètres par jour).

Ces mouvements peuvent être à l'origine de catastrophes naturelles causant des pertes humaines et des dommages importants sur les infrastructures. Au plan mondial, les mouvements de terrain causent chaque année la mort de 800 à 1000 personnes. En France, ce risque concerne environ 7000 communes et présente, pour un tiers d'entre elles, un niveau de gravité fort.

Bien que la détection de ces phénomènes soit difficile, il est possible de surveiller les mouvements du sol dans les zones sensibles en utilisant les techniques de télédétection. L'interférométrie radar et l'imagerie optique sont deux techniques de télédétection qui permettent de mésurer les mouvements du sol. L'interférométrie radar et l'imagerie optique. L'interférométrie radar est une technique de télédetection qui permet de mésurer la déformation du sol dans la ligne de visée du satellite avec une précision millimétrique en calculant la différence de phase entre deux acquisitions. Cependant, cette technique n'est pas adaptée pour mésurer les déplacements rapides. De plus, la déformation mésurée est une projection du vecteur déformation sur la ligne de visée du satellite, ne facilitant pas ainsi son interprétation. Contrairement à l'imagerie radar, l'imagerie optique permet de mésurer les déplacements rapides et est sensible aux composantes Nord-Sud et Est-Ouest de la déformation.

Il existe plusieurs algorithmes de correlation d'images. Des chercheurs de l'EOST travaillant au sein de l'équipe Déformation Active ont récemment proposé une nouvelle version de MPIC-OPT, un workflow permettant de calculer des champs de déplacement en utilisant la corrélation d'images. Cette nouvelle approche utilise la décomposition en composantes indépendantes (ICA) pour débruiter les déplacements bruts. Cependant, cette méthode statistique nécessite des heuristiques pour identifier les sources contribuant aux signaux d'interêt. Actuellement, cette étape n'est pas automatique, et est réalisée en utilisant les connaissances sur la zone d'étude.

Dans un premier temps, est de propose d'automatiser la détection de glissements de terrain en utilisant les techniques de machine learning. Dans un second temps, il vise à dévélopper une approche pour fusionner les données de l'imagerie radar et de l'imagerie optique afin d'avoir une meilleure cartographie des glissements de terrain.

Chapitre 2

Présentation de l'organisme d'accueil

L'Ecole et Observatoire des Sciences de la Terre (EOST) est une école d'ingénieurs en géophysique créé en 1935 par Edmond Rothé. Comme son nom l'indique l'EOST est également un observatoire. L'objectif premier de l'UMS830 est de faciliter l'observation pérenne des phénomènes naturels et de rendre accessible les données recueillies à la communauté scientifique. Ses tâches d'observation entrent dans le cadre des Services Nationaux d'Observation (SNO) labellisés par l'Institut National des Sciences de l'Univers du CNRS. Au total, l'EOST est impliqué dans dix services d'observation dont il est pilote au plan national ou partenaire actif. Ils concernent le domaine "Terre solide" et le domaine "Surfaces et interfaces continentales".

Ce stage s'effectue au sein de l'équipe Déformation Active. Les travaux de l'équipe portent sur la déformation lithosphérique, le fonctionnement des failles sismiques et la déformation de subsurface. Elle est fédérée autour de plusieurs disciplines, telles que la géodésie, la tectonique active, la géomorphologie et la paléosismologie. Elle appuie fortement sa recherche sur les données acquises par les observatoires de l'EOST gérés par ses membres et sur des chantiers régionaux, principalement situés en Méditerranée et en Afrique.

Chapitre 3

Méthodes

Chapitre 4

Résultats

4.1. Contraintes

Lorsque les données sont labellisées, les contraintes Must-Link et Cannot-Link se déduisent aisement . Dans la pratique , les données labellisées sont difficiles à obtenir (trop couteux) . En l'absence de telles données , les contraintes peuvent etre neanmoins rajoutées en se basant uniquement sur des connaissances . Les contraintes sont donc plus générales que les étiquettes .

On peut citer d'autres contraintes parmi lesquelles :
— le nombre de cluster ;
— le diamètre maximum : les clusters doivent avoir un diamètre maximum de γ ;
— le ϵ - voisinage : un objet doit avoir au moins un voisin dans un rayon de ϵ ;
— la séparation : les clusters doivent au moins être distants de δ .

FIGURE 4.1 – Example de contraintes ML , CL , γ , δ et ϵ

Chapitre 5

Les méthodes de partitionnement

5.1. K-means

5.2. SAMARAH

La phase (2) se compose de 4 sous-phases :

1. détection des conflits en évaluant les dissimilarités entre les pairs de résultats ;
2. le choix des conflits à résoudre ;
3. la résolution locale des conflits ;
4. l'integration des modifications locales dans le résultat global .

Modifications et convergence des résultats

Pour faire converger les résultats des clustering , il faut une méthode qui permet de mésurer la similarité entre deux clusters . Elle ne repose pas sur une distance puisque le calcul d'une distance entre deux objets n'est pas toujours possible . Pour une classe C_k^i appartenant au résultat R_i, sa classe correspondante dans le resultat R^j est la classe $C_{k_m}^j$ avec laquelle il a le plus d'objets en commun .

Pour calculer la similarité entre la classe $C_{k_m}^i$ et sa classe correspondante, il faut connaitre :

— la distribution de la classe C_k^i dans le résultat R^j :

$$\rho_k^{i,j} = \sum_{l=1}^{n_j} \left(\alpha_{k,l}^{i,j}\right)^2 avec \alpha_{k,l}^{i,j} = \frac{|C_k^i \cap C_l^i|}{|C_k^i|}$$

— la proportion d'objets de la classe C_k^i dans le résultat R^j :

$$\alpha_{k_m,k}^{j,i} = \frac{|C_{k_m}^j \cap C_k^i|}{|C_{k_m}^j|}$$

Combinaison des résultats

Quand il n'y a plus de conflits à résoudre , les résultats de chaque classifieur sont très similaires. On distingue deux cas :

— les résultats ont tous le même nombre de clusters : on peut alors associer à chaque cluster d'un résultat R^i , le cluster qui lui correspond dans le résultat R^j de manière unique et appliquer un alorgorithme de vote ;

— la condition précédente n'est pas vérifiée : une nouvelle méthode de vote est définie .

Lorsque les résultats n'ont pas tous le même nombre de classes , on s'interesse aux groupes consensuels et aux objets non consensuels . Un groupe d'objets consensuels est un ensemble d'objets appartenant au même cluster et qui sont classés de la même manière dans la majorité des résultats (un tel groupe est peut-être une forme forte) .

Un objet non consensuel est un objet qui n'appartient pas à un groupe consensuel . Ces objets correspondent souvent à ceux les plus éloignés du centre d'un cluster .
En resumé , la solution unifiée est composée des groupes consensuels auxquels on réaffecte les objets non consensuels (à l'aide des k-means par exemple) .

Algorithm 1: SAMARAH

```
1  procedure SAMARAH(dataset 𝒪 , ensemble d'agents 𝒜 , contraintes must-link ML ⊆ 𝒪 × 𝒪
     , contraintes cannot-link CL ⊆ 𝒪 × 𝒪 )
2  pour chaque agent a_i de 𝒜 faire
3      partitionner 𝒪 en utilisant a_i
4  Créer l'ensemble des conflits 𝒞 en evaluant les dissimilarités entre les pairs de resultats
5  Soit ℰ l'évaluation des résultats initiaux en fonction du critère de collaboration
6  tant que 𝒞 est non vide faire
7      choisir un conflit à résoudre de 𝒞
8      effectuer une résolution locale avec les agents impliqués
9      Soit ℰ′ l'évaluation des nouveaux resultats en fonction du critère de collaboration
10     si ℰ′ > ℰ alors
11         ℰ′ = ℰ
12         appliquer les modifications aux agents
13         calculer le nouvel ensemble de conflits
14         supprimer les conflits non résolus
15 calculer la solution unifiée en utilisant un algorithme de vote
```

Chapitre 6

Validation

Dans ce chapitre , nous exposerons premièrement les approches utilisées pour valider les résultats d'un algorithme de classification . Ensuite , nous présenterons quelques indices qui nous permettent d'implémenter la validation .

6.1. Méthodes de validation

La notion de cluster est difficile à définir . Elle dépend de l'objectif à atteindre et nécessite une bonne connaissance des données . Il existe deux approches permettant de valider les résultats d'un algorithme d'apprentissage : la validation interne et externe . A ces deux approches , nous rajouterons une étude de la stabilité de l'algorithme .

6.1.1 Validation interne

On parle de validation interne lorsque les résultats d'un clustering sont évalués par rapport à eux-mêmes . La validation interne est réalisée à l'aide de critères de qualité dit *internes* . Ces critères permettent de mesurer la compacité et la séparabilité des clusters .

Dans cette étude , plusieurs critères semblent peu pertinents . Il s'agit notamment de l'inertie et des indices de Dunn et Davies-Bouldin . La difficulté à utiliser ces indices provient du fait que Samarah ne renvoie pas toujours un résultat avec le même nombre de clusters pour une configuration de départ donnée . Comme le nombre de clusters en sortie varie , moyenner les résultats n'est pas approprié . Pour évaluer la qualité interne d'un clustering , nous utiliserons plutôt une approche *objet bien classé* et *objet mal classé* .

Pour quantifier l'homogénéité et la séparation d'un clustering , on peut mesurer ce que l'on appelle le coefficient de silhouette . Pour un point x donné, le coefficient de silhouette $s(x)$ permet d'évaluer si ce point est proche des points du cluster auquel il appartient (homogénéité) et loin des autres autres points(séparation) .

Pour quantifier l'homogénéité , on calcule la distance moyenne de x à tous les autres points du cluster C_k auquel il appartient :

$$a(x) = \frac{1}{|C_k| - 1} \sum_{u \in C_k, u \notin x} d(u, x)$$

Pour quantifier la séparation , on calcule la plus petite valeur que pourrait prendre a(x), si x était

assigné à un autre cluster :

$$b(x) = min_{l \neq l} \frac{1}{|C_l|} \sum_{u \in C_l} d(u, x)$$

Si x a été correctement assigné, alors $a(x) < b(x)$. Le coefficient de silhouette est donné par :

$$s(x) = \frac{a(x) - b(x)}{max(a(x), b(x))}$$

Il est compris entre -1 et 1, et d'autant plus proche de 1 que l'assignation de x à son cluster est satisfaisante . Pour évaluer la qualité interne globale , on calcule coefficient de silhouette moyen.

6.1.2 Validation externe

On parle de validation externe lorsque les résultats d'un clustering sont évalués à l'aide de connaissances extérieures . Elle est réalisée à l'aide de critères de qualité dit *externes* . De manière générale , ils permettent de comparer deux partitions : l'une est le résultat d'un algorithme de classification et l'autre est une vérité-terrain (les deux partitions peuvent avoir un nombre de clusters différent) . Il existe plusieurs indices externes , nous présentons ici ceux choisis pour notre étude .

Soit deux partitions π_1 et π_2 , et soient les comptages suivantes :

- sd : le nombre de couple d'objets dans le même cluster dans la partition π_1 ;
- dd : le nombre de couples d'objets differents dans les deux partitions ;
- ds : le nombre de couple d'objets dans le même cluster dans la partition π_2 ;
- ss : le nombre de couples d'objets dans le même cluster dan les deux classes ;
- mm : le nombre total de couple d'objets .

Le critère de similarité de Jaccard est donnée par :

$$J(\pi_1, \pi_2) = \frac{|\pi_1 \cap \pi_2|}{|\pi_1 \cup \pi_2|} = \frac{ss}{ss + sd + ds} \in [0, 1]$$

Le critère de qualité de Folkes-Mallows est donnée par :

$$FM(\pi_1, \pi_2) = \sqrt{\left(\frac{ss}{ss + sd}\right)\left(\frac{ss}{ss + ds}\right)}$$

L'indice de Hubert permet de mésurer la corrélation entre deux matrices [8] . Lorsque les matrices sont symétriques , il se definit par :

$$\bar{\Gamma}(P, Q) = \frac{\sum_{i=0, i<j}^{n-1} (P_{ij} - \mu_P)(Q_{ij} - \mu_Q)}{\sigma_P \sigma_Q} \in [-1, 1]$$

Où P est la matrice de proximité du jeu de données , Q est une matrice $n \times n$ où (i, j) represente la distance entre les centres des clusters auxquels les objets O_i et O_j appartiennent , μ_P, μ_Q, σ_P, σ_Q sont respectivement les moyennes et les variances des matrices P et Q .

En apprentissage semi-supervisée , la validation externe est par définition impossible à réaliser puisqu'on ne dispose pas de données etiquetées . Les indices choisis nous permettront d'étudier la stabilité de l'algorithme .

6.2. Stabilité des clusters

Le résultat d'un algorithme de classification dépend de ses paramètres et des données sur lequel il est appliqué . On dit qu'il est stable si de legères variations dans les données ou des initialisations

différentes n'induisent pas de grandes variations en sortie . L'idée est que s'il existe un nombre de clusters n correspondant à la structure naturelle des données (en excluant les cas triviaux), plusieurs exécutions de l'algorithme avec n clusters et des paramètres differents doivent conduire à des partitions très similaires . De ce fait , ce principe peut être utilisé pour trouver le nombre optimal de clusters .

Chapitre 7

Application

7.1. Jeux de données

Les jeux de données proviennent de la base de données du projet ANR FRESQAU . Chaque jeu de données est composé d'un ensemble de stations identifiés par des numéros . Chaque station mésure la concentration de divers éléments chimiques ou biochimiques renseignant sur la qualité de l'eau .

FIGURE 7.1 – Exemple de données collectées par la station 402123 (premières lignes) .

7.2. Méthodologie

Deux jeux de données de la base de donnees issue du projet FRESQAU ont été choisis pour réaliser nos experiences . Elles comportent moins de 50 % de valeurs manquantes . Les données de base sont transformées et stockées suivant le mode séquentiel . En mode séquentiel , les valeurs manquantes des données de base sont ignorées dans le format final des données .

Dans la configuration de Samarah , il est possible de spécifier le nombre de clusters mininum et le nombre de clusters maximum (il s'agit d'une contrainte sur le nombre de clusters) . Cela signifie qu'on souhaite obtenir un résultat dont le nombre de clusters est situé entre ces deux bornes . Cette contrainte est interessante si elle est suggerée par un expert . Dans notre cas , il apparait très peu pertinent de vouloir imposer une contrainte sur le nombre de cluster alors que c'est l'inconnue que l'on cherche à déterminer . Le nombre de cluster minimum a donc été fixé à 2 et le nombre de cluster maximum à 20 pour tous les tests .

Le nombre de clusters varie entre 2 et 20 . Nous avons choisi 2 agents k-means avec le même nombre de clusters . Pour étudier la stabilité de l'algorithme sur les données , nous avons exécuté SAMARAH 10 fois pour un nombre de cluster fixé .

Les critères de qualité de Hubbert, Jaccard , Folkes-Mallows , Wemmert obtenus pendant les tests sont moyennés (Ces critères sont déjà implémentés dans Multicube) .

Pour évaluer la qualité interne d'une partition , nous avons choisi le coefficient de silhouette . Il est borné et donc plus facile à intepréter . Toutefois pour s'assurer de la fiabilité des résultats obtenus , nous avons défini un critère de qualité Δ . Δ est une moyenne des écarts entre le nombre de clusters en entrée et le nombre de nombre de clusters en sortie . S'il est proche de 0 alors le nombre de clusters en entrée a été égale au nombre de clusters en sortie . Plus généralement , une faible valeur indique une faible variabilité des résultats en terme de nombre de clusters .

Pour évaluer le comportement de l'algorithme en fonction de la consistence des contraintes, deux jeux de contraintes sont crées à partir des resultats de Samarah : un jeu de contraintes *consistentes*(au sens de non aléatoire) et un jeu de contraintes aléatoires . Pour créer une contrainte consistente , on choisi

un objet aléatoirement . On calcule son coefficient de silhouette . Si la valeur obtenue est inférieure à une valeur seuil γ alors l'objet est considéré mal classé . Dans ce cas , on crée une contrainte Cannot-Link entre cet objet et un objet quelconque appartenant au même cluster . Si la valeur obtenue est supérieure à γ , l'objet est considéré bien classé . On crée alors une contrainte Must-Link entre cet objet et un objet quelconque appartenant au cluster dont il est le plus proche . Comme le coefficient de silhouette varie entre -1 et 1 , la valeur la plus naturelle de γ semble être 0 .Cependant , quelques tests ont montré qu'il y a très peu d'objets dont le coefficient de silhouette est inferieure à 0 . De ce fait , fixer la valeur de γ à 0 va plus favoriser la création de contraintes Must-Link . Pour éviter ce déséquilibre , nous avons choisi de fixer la valeur de γ à 0.25 . Pour créer une contrainte aléatoire , on choisit deux points au hasard ; s'ils appartiennent au même cluster , on crée une contrainte Cannot-Link sinon on crée une contrainte Must-Link . La création d'une contrainte aléatoire ne depend pas de γ . Il n' y a aucune méthode pour mésurer la consistence d'un jeu de contraintes avec la distance élastique . Pour avoir un aperçu de l'utilité des contraintes , on calcule le ratio $\frac{DistML}{DistCL}$. Le numérateur représente la distance moyenne entre les pairs de contraintes Must-Link ; le dénominateur représente la distance moyenne entre les pairs de contraintes Cannot-Link . Si ce ratio est inférieure à 1 , cela signfie que les paires de contraintes Must-Link sont en moyenne plus proches que les paires de contraintes Cannot-Link . Ce ratio a été aussi utilisé pour bien s'assurer que les contraintes créées aléatoirement ne sont pas nécessairement *consistentes* .

Pour chaque résultat de Samarah , nous avons relancé une itération avec le jeu de contraintes consistentes et le jeu de contraintes aléatoires . Les deux nouvelles partitions obtenues sont comparées en utilisant les indices de qualités retenues . Le pourcentage d'objets auxquels on assigne des contraintes a été fixé à 15% .

7.3. Résultats

Les résultats du jeu de données PHOSV2 obtenus sans contraintes montrent que tous les indices de similarité décroîssent quand le nombre de clusters augmente (Figure 7.2) . On remarque que Δ varie très peu pour un nombre de cluster donné . On note de très faibles valeurs entre 2 et 4 clusters pour le jeu de données PHOSV2 . Dans l'ensemble , il croît quand le nombre de clusters augmente . Il vaut en moyenne 0,98 pour 2 clusters . L'ajout des contraintes aléatoires et non aléatoires n'affecte pas le comportement des critères de similarité (Figure 7.3). Sur le jeu NITRV1 , on note une décroissance rapide du coefficient de silhouette . La meilleure partition affiche une valeure inférieure à 0.4 (Figure 7.4).

FIGURE 7.2 – Evolution des critères de similarité en fonction de nombre de classes pour le jeu de données PHOSV2 : en bleu les résultats obtenus avec les données standardisées , en rouge ceux obtenus avec les données normalisées .

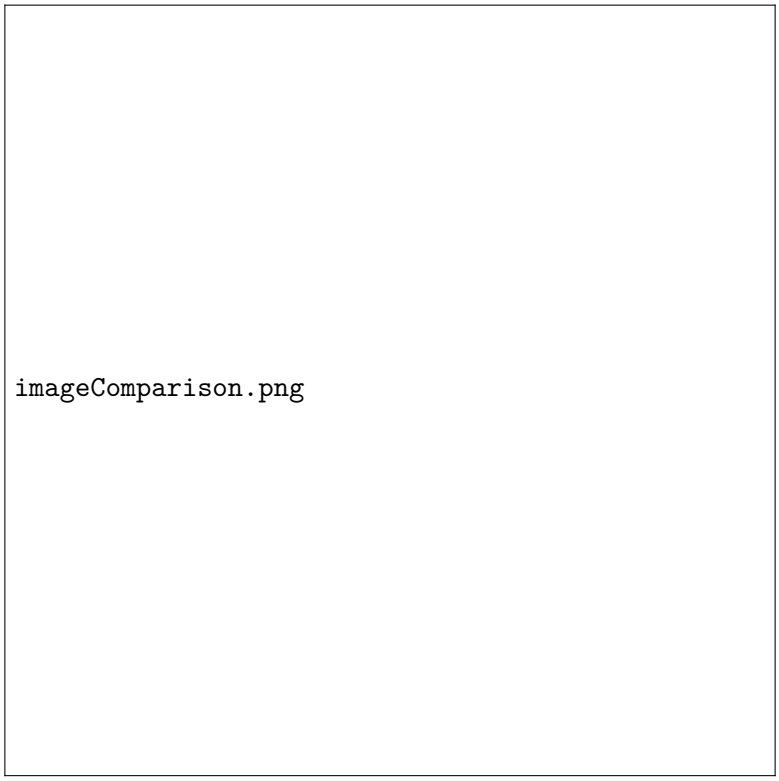

FIGURE 7.3 – Evolution des critères de similarité après ajout de contraintes aléatoires et consistentes pour le jeu de données PHOSV2 . Les courbes sont confondues car aucune contrainte n'a été violée .

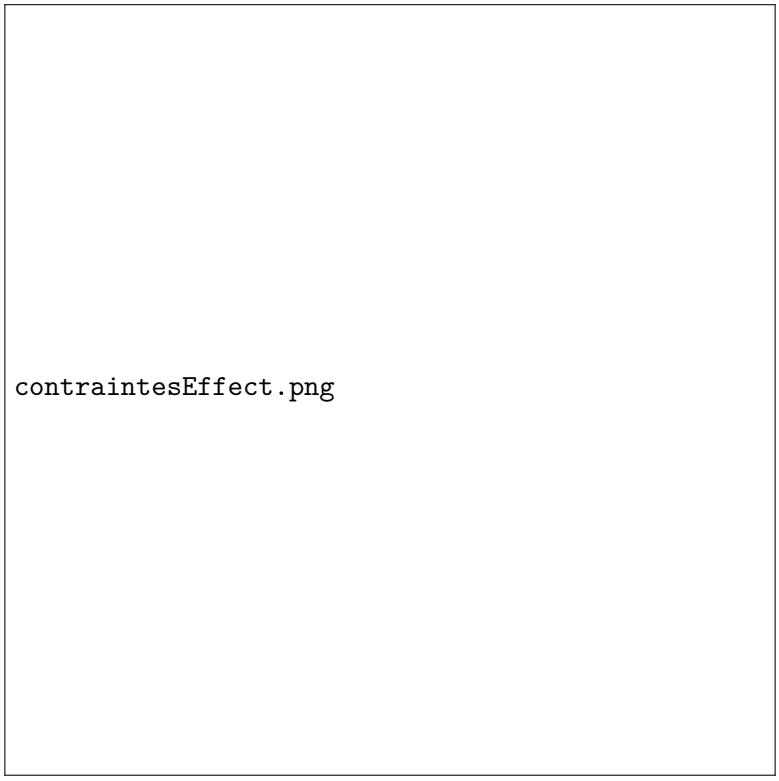

FIGURE 7.4 – Effet des contraintes aléatoires obtenu avec le jeu de données NITRV2 standardisées . Le pourcentage de contraintes violées est calculé après avoir relancé Samarah avec un jeu contraintes.

7.4. Discussion

Dans cette partie , nous analyserons les résultats de nos tests en les mettant en relation avec le fonctionnemment de Samarah et les connaissances générales sur le clustering .

Sensibilité de l'algorithme

Les résultats montrent que l'algorithme est sensible à l'initialisation . Cette remarque s'explique par la faible variation de Δ et à la croissance observée . Si par exemple , la structure des données contient 4 clusters , il est peu probable que l'algorithme renvoie souvent 4 clusters si les agents ont un nombre de clusters très éloigné de ce dernier . Par conséquent , il est important d'imposer une contrainte sur le nombre de clusters minimum et maximum en se basant sur l'avis d'un expert .

Qualité interne et nombre de clusters

La décroissance des critères de similarité quand le nombre de clusters augmente est un comportement évident . Le fait le plus marquant est la croissance de Δ . Les faibles variations de Δ pour un cluster donné peuvent s'expliquer par le fait que lorsque les agents ont au départ le même nombre de clusters , il y a plus de chance d'avoir des objets distribué de la même façon et donc très peu de conflits à résoudre . On se retrouve très vite dans la configuration que Samarah essaie d'atteindre . En conséquence le nombre de clusters final aura tendance à osciller légèrement autour du nombre de cluster en entrée . Si cette hypothèse était vraie alors l'écart absolu mésuré serait relativement constant pour tous les tests . Or on remarque que cet écart croît . Cette croissance provient donc du fait que l'algorithme devient de plus en instable lorsque qu'on s'éloigne du nombre de clusters optimal . Cette instabilité se traduit par une distribution aléatoire des objets d'où la décroissance du coefficient de silhouette qui montre que les objets sont de plus en plus mal classés (Figure 7.2) .

A cause de la variabilité des résultats , on peut seulement affirmer que le nombre de clusters se situe entre 2 et 4 pour le jeu de données PHOSV2 et 2 pour NITRV2 (Figure 7.4) .

Influence de la normalisation

Influence de la cohérence des contraintes

Les tests effectués pour étudier le comportement de l'algorithme en fonction des contraintes ne sont pas concluants . Sur le jeu de données PHOSV2 , on ne note aucune différence après l'ajout des contraintes (Figure 7.3) .
Sur le jeu de données NITRV2 réalisé avec un autre scénario de test (on a juste lancé samarah une fois au lieu de deux) , le ratio $\frac{DistML}{DistCL}$ est très élévé , ce qui implique que les paires de contraintes Cannot-Link sont plus proche en moyenne que les paires de contraintes Must-Link (c'est effectivement le cas par construction des contraintes aléatoires) . Aucune relation claire entre la qualité des contraintes et le pourcentage de contraintes violées ne se dégage (Figure 7.4) .

Chapitre 8

Conclusion

Bibliographie

[1] Ulrike von Luxburg , *Clustering stability : an overview* .

[2] Ismail Bin Mohamad and Dauda Usman , *Standardization and Its Effects on K-Means Clustering Algorithm* .

[3] Nicolas Labroche , *Méthodes d'apprentissage automatique pour l'analyse des interactions utilisateurs* .

[4] T. Lampert, T. Dao, B. Lafabregue, N. Serrette, G. Forestier, B. Crémilleux, C. Vrain, P. Gancarski , *Constrained distance based clustering for time-series : a comparative and experimental study* .

[5] Kiri Lou Wagstaff , *Intelligent clustering with instance level constraints* .

[6] F. Petitjean , A. Ketterlin, P. Gancarski , *A global averaging method for dynamic time warping,with applications to clustering* .

[7] A. Braud, S. Bringay, F. Cernesson, X. Dolques, M. Fabrègue, C. Grac, N. Lalande, F. Le Ber, M. Teisseire , *Une expérience de constitution d'un système d'information multi-sources pour l'étude de la qualité de l'eau* .

[8] M. Charrad , N. Ghazzali , V. Boiteau , A. Niknafs , *NbClust : An R Package for Determining the Relevant Number of Clusters in a Data Set* .

Algorithm 2: CONTRAINTES ALEATOIRES

```
1 procedure CréerContraintesAléatoires(Résultat 𝒫 , Nombre d'objets N)
2 n ← 0
3 tant que n < N faire
4 |   choisir aléatoirement deux objets x et y différents dans 𝒫
5 |   si cluster(x) = cluster(y) alors
6 |   |   créer une contrainte Cannot-Link entre x et y
7 |   sinon
8 |   |   créer une contrainte Must-Link entre x et y
9 |   n ← n + 1
```