Modèles de validation syntaxique et sémantique pour un stockage cohérent de données géo-spatiales imprécises dans les systèmes NoSQL document.

Besma KHALFI[*,***] Cyril De RUNZ[*,**] Sami FAIZ[***], Herman AKDAG[*]

[*]LIASD, université de Paris 8, Paris, France
khalfi@ai.univ-paris8.fr, akdag@ai.univ-paris8.fr
[**]CReSTIC, université de Reims Champagne-Ardenne, Reims, France
cyril.de-runz@univ-reims.fr
[***]LTSIRS, université de Tunis el Manar, Tunis, Tunisie
sami.faiz@insat.rnu.tn

Résumé. Avec l'explosion quantitative de données numériques et l'apparition du concept Big Data, le stockage de grandes masses de données géographiques trouvent les solutions NoSQL beaucoup plus efficaces en termes de performance et de temps de traitement. Cependant ces dernières n'offrent pas une cohérence des données stricte. En outre, si les données sont géo-spatiales et de nature imprécise, des complexités supplémentaires sont imposées à cause des caractéristiques structurelles et sémantiques plus complexes des données spatiales imprécises. Dans cet article, nous proposons une nouvelle méthodologie basée sur une validation syntaxique et sémantique des entités géo-spatiales floues avant d'être stockées dans des bases de données NoSQL orientées document.

1 Introduction

La croissance importante de données géo-spatiales soulève de gros problèmes de stockage et d'analyse de données et pose des défis supplémentaires dans la vérification de leur qualité (Li et al., 2016). Les données géo-spatiales étant volumineuses et hétérogènes, les imperfections les affectent encore plus quantitativement et qualitativement. Afin d'assurer une meilleure qualité de données géographiques, il est important d'étudier leur caractère imparfait et, par conséquent, de l'intégrer dans le processus d'analyse (Li et al., 2016).

Parmi les sources d'imperfection de données géo-spatiales, nous nous intéressons aux données imprécises pouvant être issues d'approximations, ou de sources subjectives (Desjardin et al., 2015). Basées sur la théorie des ensembles flous Zadeh (1965), de nombreuses approches ont été suggérées afin de modéliser l'imprécision dans les bases de données (Schneider, 2008).

Les systèmes NoSQL sont des systèmes non relationnels ; leurs capacités en termes de stockage, de passage à l'échelle et d'hétérogénéité des données dépassent celles des systèmes relationnels (Borkar et al., 2012). Cependant, les bases de données NoSQL ne fournissent pas une cohérence stricte des données et ne sont connues que pour les propriétés BASE [1]. Même si

1. Les propriétés BASE (Basically Availabe, Soft State, Eventuallty Consistent) des systèmes NoSQL.

l'absence du schéma est un atout, tout système doit faire quelques suppositions sur la structure de données et ajouter plus de code pour leur exploitation, ce qui implique un fort impact sur la qualité du traitement de données.

L'objectif de notre recherche est de fournir des modèles capables d'assurer la cohérence syntaxique et sémantique de données géo-spatiales imprécises en vue de pouvoir exploiter des données de qualité dans un environnement de Big Data.

Dans le reste de l'article, nous présenterons, dans un premier temps, les caractéristiques des données géo-spatiales floues en abordant les problèmes de cohérence dans les systèmes NoSQL. Ensuite, nous décrirons les solutions mises en place permettant de valider leurs cohérences structurelles et sémantiques. La troisième partie sera consacrée aux résultats d'expérimentations. Enfin, nous tirerons les conclusions et présenterons nos perspectives.

2 Contexte

2.1 Les données géo-spatiales imprécises

Les observations de l'espace révèlent la complexité et la variation de la réalité géographique. Contrairement aux espaces géographiques bien définis ayant des limites précises, nettes et linéaires, beaucoup d'autres situations dégagent des objets géographiques avec des limites plus ou moins nettes et continues. Chacun de ces objets présente un cœur où tous les éléments en son sein appartiennent pleinement au dît objet et des bordures qui lui appartiennent plus ou moins.

Par conséquent, la question « Comment caractériser un territoire imprécis ? » est une question liée aux outils de représentation en termes de modèles mathématiques et de modèles informatiques.

La théorie des ensembles flous est une des solutions mathématiques définies pour représenter l'imprécision. Grâce à elle, il est possible d'exprimer une appartenance partielle d'une valeur à un ensemble. Alors, si E est un ensemble flou et e est un élément de E, la proposition « e est un membre de E » n'est pas nécessairement soit vraie soit fausse. Elle peut être vraie que dans une certaine mesure. E est caractérisée par une fonction d'appartenance μ_E prenant ses valeurs dans [0, 1].

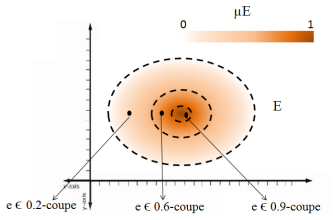


FIG. 1 – *Un ensemble flou E avec trois α-coupes.*

Pour stocker un ensemble flou, il est nécessaire de le discrétiser de sorte qu'un nombre fini de coupes doit être pris en considération afin de représenter l'ensemble flou (voir FIG. 1). Chaque coupe, que l'on appelle α-coupe, est l'ensemble des valeurs du domaine (l'ensemble

des e) ayant un degré d'appartenance supérieur ou égal à α ($\mu_E(e) \geq \alpha$). Une α-coupe de E est notée E_α. E_1 est appelé noyau ou cœur de E. Par convention,

— E_0, appelé support de E, est l'ensemble des éléments ayant un degré d'appartenance strictement supérieur à 0.
— un objet flou est un objet imprécis représenté par la théorie des ensembles flous.

2.2 La cohérence de données dans les systèmes NoSQL

Le maintien de la cohérence de données est une caractéristique essentielle des systèmes relationnels où le SGBD doit assurer une cohérence par rapport aux schémas à travers un contrôle de types et de structure et aussi une cohérence de données entre elles à travers un contrôle sémantique. Pour son maintien, le concept de règles d'intégrité (structurelles et non structurelles) est fondamental. Actuellement, les bases de données NoSQL sont le plus utilisés pour stocker de grandes masses de données grâce à l'ensemble d'outils et de langages de programmation fournis pour développer des applications sur ces nouveaux systèmes. Le problème d'incohérence de données constitue un défi pour ces systèmes car la conception sans avoir un modèle de données explicite et sans contraintes d'intégrité rend la cohérence de données très douteuse et le contrôle d'accès très complexe.

Compte tenu de la diversité significative de données dans les systèmes NoSQL, certains travaux ont mis en place des solutions de migration de systèmes relationnels vers les systèmes NoSQL mais qui ne semblent pas être une option réalisable à cause du coût d'implémentation élevé (Vitolo et al., 2015). Certaines autres méthodologies (Lotfy et al., 2016; Baker et al., 2011; Kanwar et al., 2013), ont proposé des améliorations pour conserver la cohérence et les propriétés ACID [2] des systèmes relationnels dans un environnement NoSQL. Ces solutions visent à fournir les propriétés ACID, mais ils n'abordent pas le problème de la cohérence de la structure de données lors du stockage.

En fait, les bases de données NoSQL sont utilisées aujourd'hui pour traiter les grandes masses de données géo-spatiales. Malheureusement, comme elles ne prennent pas en charge les transactions, l'intégrité de données et la durabilité, elles ne garantissent pas la qualité de données. La plupart d'entre elles n'exigent pas de contraintes structurelles sur les données et n'ont pas de schéma. Sans un schéma explicite, il est très difficile de s'assurer de la cohérence de données floues. De plus, l'accès et l'analyse de ces données sans avoir une certaine idée de leur schéma ou de leurs contraintes structurelles ne donnnent pas lieu à une analyse significative et intéressante.

Comprendre cette problématique implique de relever des défis pour le stockage et l'analyse des données. Dans la section suivante, nous présenterons notre solution pour gérer la cohérence des données géo-spatiales floues dans des systèmes non relationnels de type document.

3 Méthodologie de validation de données géo-spatiales floues

Nous travaillons avec le système NoSQL orienté document (Laxmaiah et Govardhan, 2013). Le modèle document offre un meilleur moyen que les modèles clé-valeur ou colonne ou graphe pour représenter des données complexes telles que les données géo-spatiales floues. Basés sur

2. Les propriétes ACID (Atomicity, Consistency, Isolation, Durability) des systèmes relationnels.

une approche d'agrégation, les systèmes document permettent d'améliorer les performances d'accès aux données et réduire le temps de latence.

Une base de données document est une collection de documents au format JSON. Ce format contient des informations structurelles implicites par lesquelles chaque document présente une structure d'un seul objet ou d'une collection d'objets. Les propriétés des objets sont des paires nom/valeur qui incluent des types de données de base, des valeurs multiples ou encore des types plus complexes (Crockford, 2006). GeoJSON est une extension de JSON pour l'encodage des données spatiales. Par rapport à d'autres vocabulaires connus tels que GeoSPARQL et NeoGeo, GeoJson est un vocabulaire compact disposant d'un modèle de données de base et prend en charge plusieurs types de géométrie. C'est un encodage léger, et moins verbeux que les encodages GML ou WKT utilisé par exemple par GeoSPARQL. En plus qu'il peut couvrir un grand nombre de cas d'utilisation.

Pour coder correctement la structure de données géo-spatiales floues, nous proposons premièrement le schéma *Fuzzy Geo-JSON*. Il est basé sur le format *GeoJSON* pour les données géo-spatiales (Butler et al., 2008). Fuzzy Geo-JSON étend GeoJSON afin de donner la définition de structures requises pour les différents types d'objets géo-spatiaux flous. Les caractéristiques sémantiques d'objets géo-spatiaux flous ne peuvent pas être contrôlées par le schéma Geo-JSON flou. Par conséquent, nous définissons dans un second lieu des modèles pour valider sémantiquement ces caractéristiques.

3.1 Le schéma Fuzzy GeoJSON

Le schéma $Fuzzy\ GeoJSON$ est basé sur les spécifications *json-schema-core* et *json-schema-validation*. Il spécifie les termes utilisés pour identifier les objets, les types de données et la structure de données définis dans l'IETF RFC 4627 (Crockford, 2006). FIG. 2-(a) montre la structure générale du schéma, certains détails ont été remplacés par le bloc $\{\cdots\}$. Fuzzy Geo-JSON garde la partie ***géométries*** comme elle est définie avec Geo-JSON. Cette partie donne les définitions de structure JSON des géométries de base à savoir le point, la ligne et le polygone. En se basant sur ces structures, la partie ***définitions*** donne le schéma des structures des objets géo-spatiaux flous.

FIG. 2 – *Le schéma Fuzzy GeoJSON pour les données géo-spatiales floues.*

Deux classes génériques sont fournies : ***fuzzyFeature*** et ***fuzzyFeatureCollection***. Le *fuzzyFeatureCollection* (FIG. 2-(b)) présente une collection de *fuzzyfeature*. Le *fuzzyFeature* (FIG.

2-(c)) définit tout type d'objet géo-spatial flou. Les objets géo-spatiaux flous peuvent être de structure géometrique simple :

— point flou (*fuzzyPoint*),
— polygone flou (*fuzzyPolygon*)
— ou ligne floue (*fuzzyLineString*).

Les objets géo-spatiaux flous peuvent être aussi de géométrie composite. Le schéma définit deux sous-types :

— les géometries composites homogènes : collection de points flous (*multiFuzzyPoint*), collection de polygones flous (*multiFuzzyPolygon*) et collection de lignes floues (*multiFuzzyLineString*)
— les géométries composites hétérogènes telles que les géométries alternatives, facultatives et multiples (*FuzzyAlternativeGeometry*, *FuzzyFacultativeGeometry* et *FuzzyMultipleGeometry*).

3.2 Les contraintes sémantiques

Étant donné un ensemble d'objets géo-spatiaux flous $O^E = \{O_1, \ldots, O_n\}$ où chaque objet flou O possède une géometrie $G_O \epsilon \{Point, Line, Polygon\}$ et est composé d'un ensemble fini de coupes $C^O = \{C_1, \ldots, C_m\}$. Chaque coupe C est définie par le couple (G_C, α_C) où $G_C \epsilon \{Point, Line, Polygon\}$ est la géometrie de la coupe et $\alpha_C \epsilon [0, 1]$ est le degré d'appartenance de la coupe C à l'objet flou O.

Une première contrainte sémantique est liée à la cohérence de chaque coupe C en fonction de son degré d'appartenance et de la géométrie de l'objet flou. Une coupe cohérente $C \epsilon C^O$, est définie par (O, $consistentCut^C$) où O est l'objet flou auquel appartient la coupe et $consistentCut^C$ la fonction qui vérifie que (1) pour un degré d'appartenance $\alpha_C = 1$, la géometrie de la coupe (G_C) doit être la même que celle de l'objet O ou (2) pour un degré d'appartenance $\alpha_C \neq 1$, la géometrie de la coupe G_C doit être de type $Polygon$ (voir algorithme 1).

Algorithme 1 Vérifier la cohérence des coupes

Requis : Initialiser avec O
  $G_O \leftarrow$ la géometrie de chaque objet O
  pour chaque C_i dans C^O **faire**
    $G_{C_i} \leftarrow$ la géometrie de C_i
    $\alpha_{C_i} \leftarrow$ le degré d'appartenance de C_i
    si $consistentCut(G_O, G_{C_i}, \alpha_{C_i})$ est FAUX **alors**
      retourner message d'erreur et SORTIR
    fin si
  fin pour
  retourner VRAI

La deuxieme contrainte sémantique est liée à la cohérence de l'ensemble de coupes définissant l'objet flou. Un objet flou cohérent doit avoir un ensemble de coupes connectées et normalisées. Un objet flou connecté et normalisé $O \epsilon O^E$, est défini par (C^O, $Norm^O$, $Inclus^O$) où C^O = $\{C_1, \ldots, C_m\}$ présente l'ensemble de coupes cohérentes de l'objet flou O. La fonction

$Norm^O$ vérifie que $\forall\ C\ \epsilon\ C^O$ avec le degré d'appartenance α_c, $Max(\alpha_c) = 1$. L'ensemble des coupes est donc normalisé. Pour chaque couple $\{C_i, C_k\}$ avec respectivement leurs types de géométrie et leurs degrés d'appartenance $\{(G_{Ci}, G_{Ck}), (\alpha_{Ci}, \alpha_{Ck})\}$, la fonction $Inclus^O$ vérifie la relation d'inclusion entre les coupes et vérifie que si $(\alpha_{Ci} > \alpha_{Ck})$ alors $(G_{Ci} \subset G_{Ck})$. Si l'ensemble des coupes n'est pas normalisé ou si la contrainte d'inclusion est violée, un message d'erreur est signalé à l'utilisateur (voir algorithme 2).

Algorithme 2 vérifier la cohérence de chaque objet

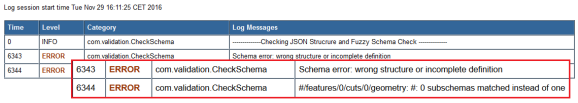
Requis : Initialiser avec O
- **si** $norm(O) \neq 1$ **alors**
  - retourner message d'erreur et SORTIR
- **sinon**
  - **pour** chaque C_i dans C^O **faire**
    - $G_{C_i} \leftarrow$ la géometrie de la coupe
    - $\alpha_{C_i} \leftarrow$ le degré d'appartenance de la coupe
    - $RC \leftarrow$ le reste des coupes de l'objet C^O
    - **pour** chaque C_k dans RC **faire**
      - $G_{C_k} \leftarrow$ la géometrie de la coupe
      - $\alpha_{C_k} \leftarrow$ le degré d'appartenance de la coupe
      - **si** $Inclus((G_{C_i}, \alpha_{C_i}), (G_{C_k}, \alpha_{C_k}))$ est FAUX **alors**
        - retourner message d'erreur et SORTIR
      - **fin si**
    - **fin pour**
  - **fin pour**
- **fin si**
- retourner VRAI

4 Expérimentations

Les principes de la méthodologie de validation sont implémentés à travers un prototype développé en Java où l'utilisateur vérifie l'ensemble de données floues syntaxiquement contre le schéma Fuzzy GeoJSON et sémantiquement contre un ensemble de fonctions développées vérifiant les contraintes sémantiques. Dans la suite, quelques résultats d'évaluation sont donnés.

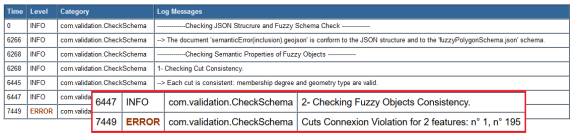
Un premier exemple de validation syntaxique génère un message d'erreur dans le fichier de journalisation est illustré par FIG. 3. Il affiche une erreur liée à une incohérence de structure d'une coupe. On suppose qu'un polygone flou ne doit pas avoir que des coupes avec des géométries de type Polygone. Dans l'ensemble de coupes, une est détectée avec un type de géométrie point.

Un deuxième exemple de validation sémantqiue est présenté par FIG. 4. Un message d'erreur lié à des coupes non connexes est affiché. Deux objets flous ne forment pas des ensembles flous connexes. Les coupes pour chaque objet doivent être incluses entre-elles selon la valeur du degré d'appartenance.


Log session start time Tue Nov 29 16:11:25 CET 2016

Time	Level	Category	Log Messages
0	INFO	com.validation.CheckSchema	----------Checking JSON Structure and Fuzzy Schema Check ----------
6343	ERROR	com.validation.CheckSchema	Schema error: wrong structure or incomplete definition
6344	ERROR	com.validation.CheckSchema	#/features/0/cuts/0/geometry: #: 0 subschemas matched instead of one

FIG. 3 – *Detection d'une erreur liée à la géometrie d'une coupe.*



Time	Level	Category	Log Messages
0	INFO	com.validation.CheckSchema	----------Checking JSON Structure and Fuzzy Schema Check ----------
6266	INFO	com.validation.CheckSchema	--> The document 'semanticError(inclusion).geojson' is conform to the JSON structure and to the 'fuzzyPolygonSchema.json' schema.
6268	INFO	com.validation.CheckSchema	----------Checking Semantic Properties of Fuzzy Objects ----------
6268	INFO	com.validation.CheckSchema	1- Checking Cut Consistency.
6445	INFO	com.validation.CheckSchema	--> Each cut is consistent: membership degree and geometry type are valid.
6447	INFO	com.valida	6447 INFO com.validation.CheckSchema 2- Checking Fuzzy Objects Consistency.
7449	ERROR	com.valida	7449 ERROR com.validation.CheckSchema Cuts Connexion Violation for 2 features: n° 1, n° 195

FIG. 4 – *Detection d'erreur de connection.*

5 Conclusion

La simplification de modèles géo-spatiaux flous ou leur absence entraîne la propagation des erreurs, ou même leur augmentation. Nous avons proposé une méthodologie basée sur un schéma spécifique appelé *Fuzzy GeoJSON* augmenté d'un ensemble de contraintes sémantiques que les données doivent valider pour être stockées correctement dans les systèmes NoSQL document.

Cette méthodologie de validation est une première étape dans l'objectif de fournir un cadre complet de gestion du Big Geo Data flou en offrant d'autres possibilités d'interrogation, d'exploration et de visualisation. Notre prochaine direction de recherche consiste à interroger les grandes masses de données geo-spatiales floues sur les relations topologiques.

Références

Baker, J., C. Bond, J. C. Corbett, J. Furman, A. Khorlin, J. Larson, J.-M. Leon, Y. Li, A. Lloyd, et V. Yushprakh (2011). Megastore : providing scalable, highly available storage for interactive services. In *Conference on Innovative Database Research (CIDR)*, Volume 11, pp. 223–234.

Borkar, V. R., M. J. Carey, et C. Li (2012). Big data platforms : what's next ? *ACM 19*(1), 44–49.

Butler, H., M. Daly, A. Doyle, S. Gillies, T. Schaub, et C. Schmidt (2008). The GeoJSON format specification. Technical report.

Crockford, D. (2006). The application/json media type for JavaScript Object Notation (JSON). Technical report.

Desjardin, E., B. Lefebvre, et C. D. Runz (2015). Intégration de l'imperfection de l'information dans les dynamiques spatiales. définitions, outils et exemples. *Revue Internationale de Géomatique 25*(3), 437–463.

Kanwar, R., P. Trivedi, et K. Singh (2013). NoSQL, a solution for distributed database management system. *International Journal of Computer Applications (IJCA) 67*(2).

Laxmaiah, M. et A. Govardhan (2013). A conceptual metadata framework for spatial data warehouse. *International Journal of Data Mining & Knowledge Management Process 3*(3), 63–73.

Li, S., S. Dragicevic, F. A. Castro, M. Sester, S. Winter, A. Coltekin, C. Pettit, B. Jiang, J. Haworth, A. Stein, et T. Cheng (2016). Geospatial big data handling theory and methods : A review and research challenges. *Journal of Photogrammetry and Remote Sensing (JPRS) 115*, 119 – 133.

Lotfy, A. E., A. I. Saleh, H. A. El-Ghareeb, et H. A. Ali (2016). A middle layer solution to support ACID properties for NoSQL databases. *Journal of King Saud University-Computer and Information Sciences 28*(1), 133–145.

Schneider, M. (2008). Fuzzy spatial data types for spatial uncertainty management in databases. In *Handbook of Research on Fuzzy Information Processing in Databases*, pp. 490–515. Information Science Reference.

Vitolo, C., Y. Elkhatib, D. Reusser, C. J. Macleod, et W. Buytaert (2015). Web technologies for environmental big data. *Environmental Modelling & Software 63*, 185–198.

Zadeh, L. A. (1965). Information and control. *Fuzzy sets 8*(3), 338–353.

Summary

With the quantitative explosion of digital data and the appearance of the Big Data concept, as relational database are not efficient, NoSQL databases are a good solution in the data storing and retrieval.

However, NoSQL databases do not offer strict data consistency. In addition, if data is geo-spatial and imprecise, additional complexities appear because of the complex syntax and semantic features of such data.

The paper presents a new methodology based on a syntax and semantic validation of the fuzzy geo-spatial entities before being stored in the document-based NoSQL databases.