Grouper les Capteurs Similaires Grace à leurs Données dans le Contexte de Massive IoT

Gwen Maudet, Mireille Batton-Hubert, Patrick Maillé, Laurent Toutain

Grouper les Capteurs Similaires Grace à leurs Données dans le Contexte de Massive IoT

Gwen Maudet[1] et Mireille Batton-Hubert[2] et Patrick Maillé[3] et Laurent Toutain[3]

[1]*SnT, University of Luxembourg, Esch-sur-Alzette, Luxembourg*
[2]*Mines Saint-Etienne, Univ Clermont Auvergne, UMR CNRS 6158, F-42023, Saint-Etienne, France*
[3]*IMT Atlantique, IRISA, UMR CNRS 6074, F-35700, Rennes, France*

L'expansion de l'Internet des objets, conjointement à la réduction du coût des appareils connectés, a permis le déploiement massif de capteurs. Puisque les capteurs sont présents en grande quantité, ils fournissent souvent des données similaires en raison de leur proximité. Dans cet article, nous cherchons à identifier de telles similitudes entre les capteurs en fonction de leurs données renvoyées, en constituant des groupes de capteurs similaires. Nous considérons un scénario générique où les capteurs sont déployés à différents moments et existent dans l'environnement pour une durée limitée, transmettant des données bruitées et irrégulières au fil du temps, sans synchronisation entre eux. Pour résoudre ce problème, nous introduisons une métrique de distance basée sur des interpolations et une solution de regroupement hiérarchique. À travers des simulations, nous démontrons la supériorité de notre méthode par rapport aux propositions de la littérature.

Mots-clefs : Massive IoT, Environnement Incertain, Distance entre Time Series, Clustering

1 Introduction

Les avancées dans le domaine de l'électronique, associées à l'émergence de réseaux à haute contrainte, ont conduit au développement de capteurs intégrés capables d'effectuer des tâches spécialisées simples telles que la mesure de la température, de l'humidité ou du CO2. Alimentés par des batteries et disponibles à faible coût, ces capteurs peuvent être déployés à grande échelle avec facilité. Par exemple, on peut envisager l'intégration de capteurs de température dans des objets du quotidien [MLN+20]. En raison du nombre important de capteurs déployés, il est fréquent que certains d'entre eux soient à proximité les uns des autres, conduisant à la présence de capteurs "similaires" qui transmettent des données étroitement liées. Cet article vise à développer une méthode permettant d'identifier des groupes de capteurs similaires en se basant sur leurs données. Contrairement à toutes les propositions précédentes [LWP07, LXWL13], nous considérons un cas général, où les messages des capteurs sont envoyés de manière irrégulière et renvoient des observations bruitées. De plus, étant donné que ces capteurs peuvent être intégrés à des objets du quotidien, ils peuvent entrer et sortir de l'environnement au fil du temps.

La solution que nous proposons se divise en deux composants principaux. Tout d'abord, nous construisons une interpolation pour chaque capteur en fonction des données qu'ils renvoient, et la distance entre deux capteurs est définie comme la différence moyenne de magnitude entre leurs interpolations respectives sur l'intervalle de définition commun. Ensuite, nous exposons une méthode de clustering agglomératif, pour laquelle nous proposons une méthode de *linkage* qui accorde plus d'importance aux distances calculées sur des intervalles communs plus longs.

Nous démontrons la supériorité de notre solution par rapport à la distance Dynamic Time Warping. De plus, nous adaptons les principes de [LWP07, LXWL13] à notre contexte, démontrant la supériorité de notre méthode de clustering.

Les sections suivantes de l'article sont présentées comme suit : tout d'abord, nous présentons le contexte en section 2. Notre métrique de similarité et la méthode de groupement sont présentées en section 3. Les simulations sont ensuite présentées en section 4, suivies de nos conclusions en section 5. Une extension de cet article peut être consultée dans [MBHMT24].

2 Problématique et Modèle

Nous envisageons un environnement représenté par plusieurs phénomènes distincts, chacun manifestant des variations propres de la quantité physique étudiée au fil du temps. Par exemple, dans un bâtiment, les variations de température différent d'une pièce à l'autre. Des capteurs sont déployés dans cet environnement, chacun suivant l'un des phénomènes présents ; notre objectif est de regrouper les capteurs qui observent le même phénomène. Un objectif plus large, non explicitement abordé dans cet article, concerne le développement de mécanismes efficaces de collecte de données capitalisant sur le concept de similarité pour répartir la charge de transmission entre les capteurs [MBHMT22, MBHMT23].

Dans le contexte d'un déploiement massif d'IoT a ressources limitées, les capteurs sont intégrés dans des objets du quotidien. De nouveaux capteurs peuvent ainsi être intégrés au cours du temps, d'autre peuvent sortir. De plus, certains capteurs peuvent devenir inactifs en raison de problèmes matériels ou d'une batterie épuisée. En conséquence de cette dynamicité du champ de capteurs, deux capteurs ne peuvent être comparés que lorsqu'ils coexistent dans l'environnement. Notamment, cet intervalle de définition commun est variable, voire inexistant. De plus, un capteur fournit des observations d'un phénomène particulier, qu'il transmet directement au terminal. Ici, les capteurs transmettent des observations bruitées en raison de dispositifs de mesure imprécis. De plus, nous considérons un cas général où les capteurs n'envoient pas des messages de manière régulière au terminal, et les transmissions ne sont pas synchronisées entre capteurs.

3 Métrique de Similarité et Méthode de Groupement

3.1 Similarité entre Capteurs : Différence moyenne entre les Interpolations

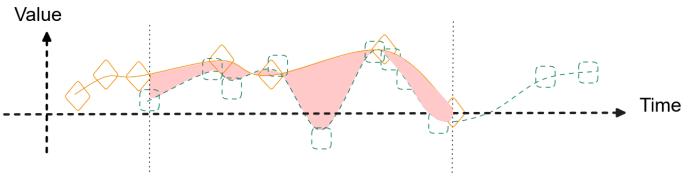

Fig. 1: Représentation de deux ensembles de messages : les losanges orange et les carrés verts en pointillés représentent deux ensembles de messages, avec le temps sur l'axe des x et les valeurs sur l'axe des y. Les interpolations des messages sont représentées par des lignes orange continues et des lignes vertes en pointillés. Les lignes verticales en pointillés indiquent l'intervalle de définition commun. La zone entre les deux interpolations est représentée par un remplissage rouge.

À partir des hypothèses juste présentés, nous présentons un exemple de deux ensembles de messages en fig. 1.

Les observations sont irrégulièrement espacées et bruitées, rendant les comparaisons directes difficiles. Par conséquent, comme première étape, nous proposons d'utiliser une méthode d'interpolation sur chaque historique d'observations. Cette approche transforme un historique d'observations en une fonction continue, facilitant les comparaisons. Plus précisément, nous utilisons le *krigeage*, qui est une méthode d'interpolation basée sur des processus gaussiens régis par des covariances, permettant l'estimation et la prise en compte du bruit de mesure [Kle09].

En fonction des courbes d'interpolation construites pour chaque capteur, nous définissons la distance entre deux capteurs comme la différence moyenne des magnitudes entre les interpolations sur leur intervalle de définition commun.

3.2 Clustering Hiérarchique Avec Linkage comme Moyenne Pondérée

Dans un problème de clustering classique, on considère des objets avec n variables et on cherche à regrouper des objets qui sont proches lorsqu'ils sont représentés dans un espace où chaque variable constitue une dimension. Dans notre contexte, un objet représente un ensemble de messages, où nous avons défini une *distance* sur leur *durée d'intervalle de définition commun* pour chaque pair de capteurs, ce qui rend différent le développement de l'algorithme de clustering.

Pour ce problème de clustering, nous choisissons de nous concentrer sur des solutions basées sur la méthode de Clustering Ascendant Hiérarchique (CAH). Le principe est de fusionner de manière itérative les clusters dont la distance est la plus faible jusqu'à un seuil d'arrêt [LW67]. Une méthode de *linkage* définie la distance entre clusters, généralement comme combinaisons linéaires de distances entre les capteurs des clusters comparés. Ici, nous choisissons d'adapter le *linkage* moyen pour mieux correspondre à notre problème. Nous souhaitons donner plus d'importance aux distances calculées sur des périodes plus longues. Ainsi, nous définissons la distance entre deux clusters comme la somme des distances entre les paires d'objets de clusters différents, pondérées par leur durée d'intervalle de définition commun. Soient $d(i,j)$ la distance entre les capteurs i et j, et $\delta(i,j)$ la durée de leur intervalle de définition commun. En considérant $i \in I$ les capteurs du cluster I, et $j \in J$ pour J, nous définissons la distance entre I et J par $D(I,J) = \frac{\sum_{i\in I}\sum_{j\in J}\delta(i,j)d(i,j)}{\sum_{i\in I}\sum_{j\in J}\delta(i,j)}$.

4 Simulations

Dans cette section, nous réalisons des simulations en générant deux phénomènes continus distincts. Un phénomène est construit à partir de la fonction générique $f(t) = \sum_{i=1}^{30}(\alpha_i \cos\omega_i t + \beta_i \sin\phi_i t)$, $\alpha_i, \beta_i \sim \mathscr{U}(-100, 100)$, $\omega_i, \phi_i \sim \mathscr{U}(0, \frac{2\pi}{30})$ (garantissant une période d'oscillation minimale de 30, limitant la variabilité). Ensuite, nous mettons à l'échelle la fonction dans l'intervalle $[-1,1]$ (compressant les valeurs du phénomène dans un petit segment de valeurs).

Chaque capteur suit l'un des deux phénomènes, toujours le même, et envoie une observation bruitée du phénomène, avec un bruit gaussien d'écart-type $\sigma = 0.2$. De nouveaux capteurs entrent dans l'environnement au fil du temps, suivant une distribution de Poisson avec un paramètre $\lambda = 0.1$, et chacun suit l'un des deux phénomènes avec une probabilité égale. La durée d'un capteur dans l'environnement suit une distribution exponentielle avec un paramètre $\mu = 0.01$. Lorsqu'il est dans l'environnement, un capteur transmet des observations suivant une distribution de Poisson avec un paramètre $\gamma = 1$. Nous terminons la simulation à $t = 1000$. Pour éviter les capteurs qui n'ont d'intervalles de définition communs qu'avec d'autres capteurs, nous ne considérons que les capteurs toujours actifs après $t = 200$.

Nous comparons notre méthode à une autre méthode de similarité et une méthode de clustering. Nous choisissons la méthode Dynamic Time Warping comme algorithme mesurant la distance entre deux séries temporelles, développée pour gérer les décalages temporels et les différences d'échantillonnage entre les séries temporelles comparées. De plus, nous nous appuyons sur la méthode de clustering provenant de [LWP07, LXWL13], qui a également pour objectif de créer des groupes de capteurs similaires. Ils utilisent une formulation en graphe pour poser un problème de Clique Partitioning. Nous adaptons cette méthode à l'algorithme CAH pour avoir des méthodes comparables, en utilisant le *linkage* complet, qui définit la distance entre clusters comme la distance maximale entre capteurs de clusters différents. De cette façon, lors de la fusion des deux clusters ayant la distance la plus faible, nous limitons la distance globale maximale entre deux capteurs appartenant au même cluster. Nous fixons le seuil du nombre de clusters maximum à 3.

Nous analysons les performances de clustering par rapport à l'appartenance réelle des capteurs au phénomène grâce aux critères d'Homogeneity et de Completness, donnant la V-measure [RH07]. Nous présentons les performances des combinaisons possibles entre notre approche et l'approche comparative en Table 1.

Dans l'ensemble, nous constatons une amélioration de 23% des performances en termes de V-measure par rapport à la méthode choisie pour la comparaison. Il est à noter que notre métrique de similarité présente un impact positif plus important que la méthode de *linkage*.

Métrique de similarité	Méthode de Linkage	Homogeneity		Completness		V-measure	
		Mean	Std	Mean	Std	Mean	Std
Différence moyenne	**Moyenne pondérée**	0.72	0.23	0.60	0.20	0.65	0.22
Différence moyenne	Complet	0.70	0.21	0.52	0.16	0.59	0.18
Dynamic Time Warping	**Moyenne pondérée**	0.53	0.34	0.50	0.27	0.50	0.31
Dynamic Time Warping	Complet	0.60	0.22	0.43	0.17	0.50	0.19

Tab. 1: Comparaison des résultats de clustering en utilisant la métrique de similarité correspondante et la méthode de *linkage* pour le CAH avec un nombre final de clusters fixé à 3. Mise en évidence de nos propositions en **gras**.

5 Conclusion and Perspectives

Dans cet article, nous avons présenté une méthode pour regrouper des capteurs en fonction de leurs observations. Notre approche a été conçue pour traiter un scénario pessimiste où les capteurs entrent et sortent au fil du temps, envoyant des observations irrégulières et bruitées de phénomènes potentiellement différents.

Ce travail ouvre la voie à d'autres analyses. Premièrement, il serait plus réaliste d'étudier ce problème de manière dynamique, où des décisions seraient prises au cours de la surveillance. De plus, il serait pertinent d'examiner le cas où le capteur continue d'être dans l'environnement mais ne suit plus le phénomène : suit un phénomène différent ou retourne des données aberrantes.

References

[Kle09] Jack P.C. Kleijnen. Kriging metamodeling in simulation: A review. *European Journal of Operational Research*, 192(3):707–716, 2009.

[LW67] G. N. Lance and W. T. Williams. A general theory of classificatory sorting strategies: II. clustering systems. *The Computer Journal*, 10(3):271–277, 1967.

[LWP07] Chong Liu, Kui Wu, and Jian Pei. An energy-efficient data collection framework for wireless sensor networks by exploiting spatiotemporal correlation. *IEEE Transactions on Parallel and Distributed Systems*, 18(7):1010–1023, 2007.

[LXWL13] Zhidan Liu, Wei Xing, Yongchao Wang, and Dongming Lu. Hierarchical spatial clustering in multihop wireless sensor networks. *International Journal of Distributed Sensor Networks*, 9(11):528980, 2013.

[MBHMT22] Gwen Maudet, Mireille Batton-Hubert, Patrick Maille, and Laurent Toutain. Emission scheduling strategies for massive-IoT: Implementation and performance optimization. In *IEEE NOMS*, pages 1–4, 2022.

[MBHMT23] Gwen Maudet, Mireille Batton-Hubert, Patrick Maillé, and Laurent Toutain. Energy efficient message scheduling with redundancy control for massive iot monitoring. In *IEEE WCNC*, pages 1–6, 2023.

[MBHMT24] Gwen Maudet, Mireille Batton-Hubert, Patrick Maillé, and Laurent Toutain. Grouping Similar Sensors Based on their Sent Data in a Massive IoT Scenario. preprint, January 2024.

[MLN+20] Naser Hossein Motlagh, Eemil Lagerspetz, Petteri Nurmi, Xin Li, Samu Varjonen, Julien Mineraud, Matti Siekkinen, Andrew Rebeiro-Hargrave, Tareq Hussein, Tuukka Petaja, Markku Kulmala, and Sasu Tarkoma. Toward massive scale air quality monitoring. *IEEE Communications Magazine*, 58(2):54–59, 2020.

[RH07] Andrew Rosenberg and Julia Hirschberg. V-measure: A conditional entropy-based external cluster evaluation measure. In Jason Eisner, editor, *EMNLP-CoNLL*, pages 410–420, Prague, Czech Republic, June 2007. Association for Computational Linguistics.