

Le modelage des travaux d'un Système de Grille

Xiangliang Zhang, Michèle Sebag, Cécile Germain

▶ **To cite this version:**

Xiangliang Zhang, Michèle Sebag, Cécile Germain. Le modelage des travaux d'un Système de Grille. 16th congrès francophone AFRIF-AFIA Reconnaissance des Formes et Intelligence Artificielle (RFIA), Jan 2008, Amiens, France. inria-00287384

HAL Id: inria-00287384
https://inria.hal.science/inria-00287384

Submitted on 11 Jun 2008

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Modélisation des tâches d'une grille de calcul

Modelling the Jobs of a Grid System

X. Zhang M. Sebag C. Germain-Renaud

TAO, INRIA-Futurs, CNRS UMR 8623, LRI

Bât. 490, LRI Université Paris-Sud 11, 91405 ORSAY, France
{xlzhang, sebag, cecile}@lri.fr

Résumé

Les grilles de calcul agrègent un très grand nombre de ressources hétérogènes et distribuées à grande échelle. La complexité de ces systèmes a motivé l'apparition d'un nouveau champ de recherche, l'Informatique Autonomique (ou Autonomic Computing*), visant le contrôle et l'administration automatiques de ces systèmes. Dans ce contexte, nous nous intéressons ici à l'analyse exploratoire et la modélisation des tâches soumises à la grille de calcul EGEE*[1]*, visant la découverte de catégories de tâches représentatives, et allant au-delà de la distinction élémentaire entre les tâches correctement exécutées et celles ayant échoué. La difficulté de cette modélisation tient entre autres à la taille (5Gb) et l'hétérogénéité des données disponibles. L'approche proposée se fonde i) sur la segmentation des données en sous-ensembles plus homogènes ; ii) sur la construction d'hypothèses discriminantes à partir de ces sous-ensembles ; iii) sur l'usage de ces hypothèses pour un changement de représentation des données, et l'exploitation de la métrique associée pour définir des clusters. La validation de l'approche repose sur la stabilité des clusters par rapport aux algorithmes discriminants utilisés dans l'étape ii), d'une part, et d'autre part, sur l'interprétation des clusters obtenus par les experts (*"la position des inputs n'est pas disponible' ; "problèmes sur l'évaluation de rang sur les ressources"*).*

Mots Clef

Fouille de Données, Clustering, Informatique Autonomique, Modélisation de Grille de Calcul.

Abstract

The rise of grid systems, made of a large number of heterogeneous resources, motivated the highly challenging field of Autonomic Computing, aimed at the self-management of such complex systems. A preliminary step, this paper is interested in modeling the jobs submitted to the grid and discovering meaningful job categories, beyond the coarse distinction between "successfully executed" and "failed" jobs. The difficulty lies in the huge size of the available observations (the Logs of the grid) and their heterogeneity, severely hindering Machine Learning algorithms at the state of the art. This difficulty is addressed through an original 3-step process : i) the data are firstly sliced into (more) homogeneous subsets, where a data slice involves jobs submitted by a single user, or during a single period of time ; ii) supervised ML algorithms are used to construct discriminant hypotheses on each data slice ; iii) these hypotheses are used to map the dataset onto a metric space, and thus enable clustering. The approach is validated from the cluster stability w.r.t. the supervised learning step in ii), and the "natural" interpretation of the clusters after the expert.

Keywords

Applications of Data Mining, Clustering, Grid Modelling, Autonomic Computing

1 Introduction

La complexité croissante des systèmes informatiques a provoqué une demande d'outils d'auto-configuration, d'auto-adaptation et d'auto-réparation, conduisant à l'apparition de l'Informatique Autonomique (Autonomic Computing[2]) depuis les années 2000. Concrètement, les systèmes autonomiques visent à "se comporter conformément aux objectifs de haut niveau fixés par des humains" [1] ; ils doivent pouvoir découvrir les problèmes, envoyer des rapports à l'administrateur du système, et dans certains cas se rétablir sans intervention humaine après des incidents de fonctionnement. Une façon de réaliser cet objectif consiste à munir chaque système d'un modèle de son comportement (normal), lui permettant ainsi de détecter les déviations éventuelles et d'anticiper l'apparition de problèmes ultérieurs. De tels modèles des systèmes complexes que sont les grilles peuvent difficilement être définis *a priori* ; l'alternative consiste à élaborer un modèle comportemental, en utilisant des approches de Fouille de Don-

[1] http ://www.eu-egee.org/

[2] http ://www.research.ibm.com/autonomic/

nées à partir des observations disponibles, e.g. des logs de la grille [2, 3].

Cet article concerne la modélisation du comportement de la grille EGEE[3], qui fédère environ 40.000 CPUS, 5 Petabytes de stockage, et exécute simultanément 20.000 tâches en moyenne. La modélisation d'un tel système soulève des difficultés diverses. En premier lieu, l'état de l'infrastructure (site, machines, réseau) ne peut être connu avec précision à tout instant ; en second lieu la charge dépend du comportement collectif des utilisateurs (paradigme de mutualisation des ressources), qui reflète la diversité des communautés se partageant la grille (e.g., physique des hautes énergies, sciences de la vie, chimie computationnelle, simulation financière), et celle de leur agenda scientifique (conférences, expériences à grande échelle).

A titre de première étape vers une modélisation du comportement d'EGEE, cet article se concentre sur les tâches contrôlées par l'intergiciel LCG/gLite (qui supervise la plus grande partie des tâches soumises à EGEE), et plus particulièrement sur la réussite ou l'échec de ces tâches.

Le cycle de vie d'une tâche est enregistré par le Logging and Bookkeeping (L&B), qui fait partie du groupe de services en charge de l'exécution des tâches (Workload Management System, WMS)[4] ; de façon sommaire, chaque tâche est étiquetée comme un succès (*successfully terminated*) ou un échec (tous les autres cas). Les données, ou traces, étudiées dans la suite correspondent aux tâches traitées par le WMS d'octobre 2004 à octobre 2005, donc lors du démarrage d'EGEE. Ceci explique le taux élevé des échecs (70%) dans les données, à comparer avec le taux actuel (10% de tâches en échec en 2007).

Le travail présenté s'inscrit dans le cadre d'une analyse exploratoire, et vise l'identification de sous-ensembles de tâches (job clusters) qui puissent être interprétés de façon naturelle et qui soient compatibles avec la distinction élémentaire entre les tâches ayant échoué et les autres. L'identification et la caractérisation de tels clusters sont directement opérationnelles pour améliorer le service rendu à l'utilisateur, e.g. permettant la prévention et l'explication des échecs les plus courants.

Les difficultés sont de trois ordres. Tout d'abord, les traces sont produites automatiquement pour le besoin des divers services du WMS ; elles sont structurées (spécifications en langage JDL, nombre variable d'évènements pour chaque tâche), redondantes, et aucune métrique naturelle n'est disponible. En second lieu, les tâches sont extrêmement hétérogènes, avec des variations d'un ou plusieurs ordres de grandeur selon la période de l'année et l'expertise des utilisateurs. La dernière difficulté, et non la moindre, est la taille des fichiers de Logging & Bookkeeping disponibles (5 Gb), interdisant de fait l'usage de la plupart des algorithmes d'apprentissage automatique pour des raisons de coût de calcul.

Pour cette raison, l'approche présentée dans cet article est *any-time*, combinant induction constructive (changement de représentation), apprentissage supervisé et non supervisé. L'étape d'induction constructive utilise un ensemble d'apprentissage formé de 90% des données disponibles ; la catégorisation, ou clustering, opère sur un ensemble de validation, qui comprend les 10% de données restantes.

Dans la première étape, l'ensemble d'apprentissage est divisé en sous-ensembles dans le but de réduire l'hétérogénéité des données d'une part et le coût de calcul d'auytre part. Concrètement deux partitions sont définies. La première se fonde sur les utilisateurs ; chaque sous-ensemble est formé par les tâches soumises par un même utilisateur. Cette partition réduit donc l'hétérogénéité due à l'expertise des utilisateurs. La seconde partition est fondée sur le temps ; chaque sous-ensemble est formé par les tâches soumises au cours d'une même semaine. Cette seconde partition réduit ainsi l'hétérogénéité due au fait que la charge de la grille (et donc le traitement des tâches) varie au cours du temps.

Dans la deuxième étape, chaque sous-ensemble est soumis à un apprentissage discriminant ; les hypothèses discriminant les succès et les échecs dans le sous-ensemble sont extraites. Deux algorithmes sont considérés. La méthode de référence est une machine à vecteurs supports (SVM) linéaire, implémentée par SVM^{Light} [4]. Une seconde méthode se fonde sur l'optimisation du critère de discrimination donné par l'aire sous la courbe ROC (critère Wilcoxon), par optimisation stochastique ; l'implémentation utilisée est ROGER (*ROc-based GEnetic learneR*) [5]. Dans les deux cas, une hypothèse est une fonction de l'espace de description initial des tâches sur la droite réelle. L'ensemble des hypothèses extraites à partir des sous-ensembles issus de la première partition est appelé U-representation (pour utilisateur) ; l'ensemble des hypothèses extraites à partir des sous-ensembles issus de la seconde partition est appelé W-representation (pour "week").

Dans la troisième étape, les hypothèses issues de la première ou la seconde partition sont utilisées pour transformer chaque tâche en un vecteur de réels. Pour limiter la redondance éventuelle des hypothèses et des tâches, un double clustering est effectué en s'inspirant de Slonim et Tishby [6]. Les clusters ainsi obtenus sont tout d'abord évalués en fonction de leur stabilité au sens des critères définis par Meila [7] ; on examine la stabilité en fonction i) des différentes partitions ; ii) des différents algorithmes d'apprentissage supervisé utilisés, SVM^{Light} ou ROGER. Enfin, les clusters sont soumis aux experts et plusieurs interprétations significatives sont obtenues ("Input position non available, "Retry count hit").

L'article est organisé de la façon suivante. La section 2 présente brièvement l'état de l'art dans les domaines de la réduction de dimensionnalité et du clustering, et discute les algorithmes existants en fonction de leur com-

[3] *Enabling Grid for E-Science in Europe*, `http ://www.eu-egee.org/`

[4] Formellement, le L&B est un service compagnon du service *broker*. Ce dernier effectue le placement des tâches pendant que le L&B enregistre l'ensemble des transactions qui interviennent au cours de la vie de la tâche.

plexité. La section 3 décrit la préparation et la segmentation des données. La section 4 présente la méthode d'induction constructive. La section 5 décrit le double clustering des hypothèses et les tâches. La section 6 présente le protocole expérimental et la section 7 décrit les résultats obtenus. En particulier, la qualité du clustering est discutée par rapport à la pureté des clusters obtenus (vis à vis du classement des tâches en succes/échec), et par rapport aux critères de stabilité non-supervisée [7]. L'article conclut sur les forces et les faiblesses de l'approche présentée, et discute les perspectives de recherche ouvertes par ce travail.

2 L'état de l'art

L'apprentissage automatique dépend de manière cruciale de la représentation des données disponibles ; aussi l'induction constructive, ou changement de représentation des données, a-t-elle été considérée comme l'une des étapes clé de l'apprentissage. Sans prétendre à l'exhaustivité, nous nous limiterons à considérer les approches de la réduction de dimensionnalité et l'apprentissage non supervisé ou clustering.

2.1 Réduction de dimensionnalité

On distingue classiquement les approches linéaires et les approches non linéaires en réduction de dimensionnalité. Les approches linéaires classiques sont l'analyse en composantes principales (Principal Component Analysis, PCA) et la décomposition en valeurs singulières (Singular Value Decomposition, SVD). La limitation principale de ces méthodes dans notre contexte est l'hétérogénéité des données au sens de la description initiale des tâches. Typiquement, quand les valeurs d'un attribut numérique diffèrent par plusieurs ordres de grandeur, la valeur moyenne fournit peu d'information.

Depuis 2000, plusieurs approches non supervisées de réduction de dimensionnalité non linéaire ont été présentées, comme Isomap [8] et le plongement localement linéaire (Locally Linear Embedding, LLE) [9]. Ces approches de référence sont difficilement exploitables dans notre contexte en raison de leur complexité (au moins quadratique dans le nombre d'exemples) ; les versions améliorées (e.g., [10]) exigent des connaissances a priori préalable (par exemple, sur la sélection de bornes).

Quelques extensions de la réduction de dimensionnalité non linéaire au cas supervisé ont été proposées. Par exemple, Sugiyama [11] a proposé une méthode supervisée (Local Fisher discriminant analysis, LFDA), qui combine l'analyse discriminant de Fisher et le plongement localement linéaire, et qui est efficace sur les problèmes multimodaux. Une autre approche, partialement supervisée, proposée par Yang et al. [12], utilise la connaissance a priori pour améliorer la stabilité de la solution.

Une problématique étroitement liée à la réduction de dimensionnalité est l'apprentissage de fonctions de similarité ou de distance. Par exemple, Weinberger et al. [13] proposent de considérer l'apprentissage d'une distance comme un problème d'optimisation, en maximisant (resp. minimisant) la distance de chaque point à ses K voisins appartenant à des classes différentes (resp. identiques) ; une distance de Mahalanobis est déterminée, en résolvant ce problème d'optimisation par programmation semi-définie. Cette approche souffre du même problème de complexité quadratique en fonction du nombre d'exemples.

2.2 Clustering

Le très classique algorithme de clustering des K-moyennes ou K-means repose sur une fonction de similarité ou une distance définie sur l'espace de représentation. Une question clé concerne la stabilité des clusters obtenus, dans la mesure où les K-means font en général intervenir une initialisation aléatoire qui influence le resultat final.

La relation entre PCA et K-means a été étudiée par Ding et He [14], prouvant que les centres des clusters sont liés aux vecteurs propres du PCA. À partir de ce résultat, sous l'hypothèse de *bonne-separation* (well-separateness), Meila a prouvé qu'un bon clustering peut être approché (à une permutation près des clusters) par les composantes principales des données [7] ; en conséquence, une borne sur la qualité et la stabilité d'un clustering peut être dérivée de sa distance aux vecteurs propres du PCA.

L'hypothèse de travail de bonne-séparation exige que les données ne soient pas situées sur une variété de dimension inférieure à $K-1$. Au cas contraire, la qualité des clusters construits par K-moyennes n'est pas en cause ; toutefois la borne définie par [7] n'est plus opérationnelle.

Notons que d'autres approches de clustering, tel le clustering spectral [15], n'ont pu être considérées pour des raisons de complexité. La construction de la matrice des distances entre les exemples, quadratique en fonction du nombre d'exemples, n'est en effet pas envisageable pour des applications de grande taille.

3 Préparation et segmentation des données

Cette section présente le pré-traitement des fichiers de logs et la prise en compte l'hétérogénéité des tâches traitées par la grille de calcul.

3.1 Préparation des données

La base de données du L&B contient les traces comportementales des tâches ; le cycle de vie de chaque tâche est représenté comme une séquence d'événements de longueur variable ; le nombre d'événements intervenant dans une tâche est compris entre 1 et 174. Environ 300.000 tâches et 3.300.000 événements sont représentés dans les données disponibles. Les événements, décrivant les différentes étapes de traitement de la tâche par les différents services de la grille, sont sauvegardés dans trois tables différentes. Le schéma de la base de donnée est présentée fig. 1.

Il est bien connu que la préparation des données est l'activité la plus consommatrice de temps dans une application de fouille de données, représentant jusqu'à 80% du

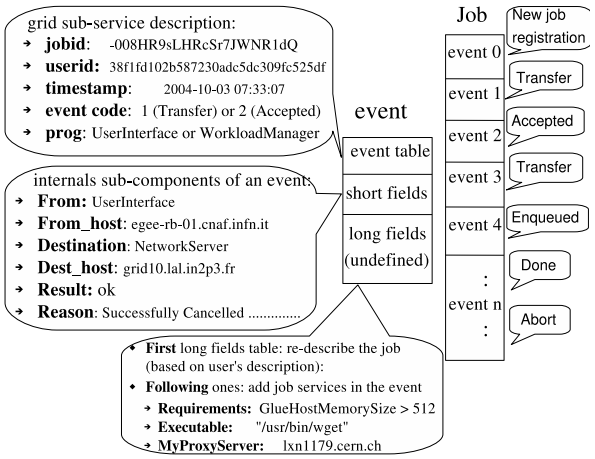


FIG. 1 – Schéma de la base de données du L&B

temps total. Une suite logicielle a été conçue pour identifier des dépendances fonctionnelles, segmenter les attributs composés, segmenter les chaînes de caractères, et plus généralement éliminer l'information redondante, et formater chaque tâche dans un vecteur d'attributs propositionnels.

Les attributs numériques sont normalisés (centrés avec écart-type 1). Les connaissances a priori sont exploitées pour segmenter les valeurs des attributs catégoriques (les noms de machine, les noms de fichier, les logins d'utilisateur). Chaque attribut prenant m valeurs nominales est transformé en m attributs booléens ; plus précisément, les valeurs suffisamment fréquentes deviennent des attributs booléens ; les autres valeurs sont fusionnées sous l'attribut booléen *other-value*.

A la fin de ce processus, chaque tâche est décrite par 408 attributs numériques et booléens. Il faut souligner que la distribution des attributs numériques est loin d'une distribution normale ; les premières tentatives de modélisation, de type identification de mixture de gaussiennes, n'ont pas donné de résultats.

3.2 Segmentation des données

Comme il a été mentionné dans l'introduction, les tâches sont divisées en deux classes, celle des succès (*successfully terminated jobs*, qui représentent environ 30% de l'échantillon) et celle des échecs (tous les autres jobs). Les deux classes sont très hétérogènes, pour plusieurs raisons. Le premier type d'hétérogénéité est dû aux utilisateurs qui ont lancé les jobs ; bien entendu, les niveaux d'expertise des utilisateurs diffèrent. Mais aussi, la grille est utilisée par les communautés scientifiques différentes, allant de la physique des hautes énergies au multimédia. Les usages sont également très différents d'une communauté à l'autre ; par exemple les jobs soumis dans le domaine de la physique des hautes énergies sont en moyenne plus longs et plus consommateurs de ressources que pour d'autres communautés.

Le second type d'hétérogénéité est dû au fait que la charge de la grille, et donc le cycle de vie des jobs, varie énormément au cours du temps. Ceci s'explique par le paradigme de mutualisation des ressources ; certaines semaines sont considérablement plus chargées que d'autres.

Une heuristique de segmentation agressive inspirée de Kearns et Li [16] est utilisée pour prendre en compte l'hétérogénéité des données. Plus précisément, deux partitions sont définies sur l'ensemble des jobs ; la première dépend des utilisateurs (à tout utilisateur est associé le sous-ensemble des jobs soumis par cet utilisateur), et la seconde dépend du temps (à toute semaine de la période considérée est associé le sous-ensemble des jobs soumis durant cette semaine). Les sous-ensembles ainsi obtenus sont ensuite filtrés : les sous-ensembles de trop petite taille, ainsi que ceux dont la distribution est trop déséquilibrée (e.g. ne contenant que des jobs en échec) sont éliminés.

De cette façon, chaque sous-ensemble contient des jobs relativement homogènes ; la variabilité initiale est réduite en fixant soit l'expertise de l'utilisateur et la communauté scientifique auquelle il appartient, soit la charge de la grille au moment où les jobs ont été traités.

4 Induction constructive

La deuxième étape de l'approche présentée concerne la recherche d'une bonne représentation, ou induction constructive. Cette étape cherche à remédier à la redondance de la description initiale des jobs, tout en préservant l'information discriminante entre les succès et les échecs. Elle exploite les sous-ensembles construits dans la partie précédente.

4.1 Principe

Formellement, chaque sous-ensemble est traité par apprentissage supervisé pour apprendre une hypothèse discriminante h, définie sur l'ensemble de description initiale $\mathcal{X}$ des jobs et à valeurs dans $\mathbf{R}$. Cette hypothèse définit donc un attribut numérique calculable ; par construction, cette hypothèse est partiellement indépendante de l'hétérogénéité des données et plus à même de capturer des facteurs simples de discrimination entre jobs satisfaisants et en échec.

Enfin, l'ensemble des hypothèses issues de différents sous-ensembles construits à l'étape antérieure est utilisé pour redécrire chaque job comme un vecteur de réels. Cette redescription, inspirée des approches de type cascade, permet de plonger l'espace initial $\mathcal{X}$ dans l'espace métrique $\mathbb{R}^d$, si d est le nombre d'hypothèses considérées. Ce plongement permettra d'appliquer des techniques de clustering (section 5).

4.2 Influence de l'apprentissage

L'intérêt de cette approche a été étudié en comparant les résultats obtenus avec deux algorithmes d'apprentissage discriminant. L'algorithme de référence est une machine à vecteur de support (SVM) [17] ; nous avons utilisé l'implémentation SVM^{Light} avec noyau linéaire et paramètres par défaut [4].

Le second algorithme d'apprentissage discriminant considéré se fonde sur l'optimisation de l'aire sous la courbe ROC (AUC), qui optimise la statistique classique de Wilcoxon-Mann-Whitney (WMW) [18]. Soit $\mathcal{E} = \{(\mathbf{x}_i, y_i), \mathbf{x}_i \in \mathcal{X},\ y_i \in \{1, -1\}, i = 1 \ldots n\}$ l'ensemble d'apprentissage disponible, où $\mathbf{x}_i \in \mathcal{X}$ est la description du i-ème exemple et y_i son étiquette. Soit h une hypothèse définie de $\mathcal{X}$ sur $\mathbb{R}$; la statistique de Wilcoxon de h mesure la fraction de paires d'exemples (i, j) qui sont correctement ordonnées par h :

$$WMW(h) = Pr(h(\mathbf{x_i}) > h(\mathbf{x}_j) | y_i > y_j)$$

Comme indiqué par [18], ce critère est plus stable que l'erreur empirique ; il est quadratique et non linéaire en fonction du nombre d'exemples. Par ailleurs les hypothèses basées sur l'optimisation de ce critère sont directement interprétables en termes de probabilité de classification.

Cependant, à la différence du problème d'optimisation quadratique défini par une SVM, l'optimisation de l'aire sous la courbe ROC n'est pas un problème bien posé. L'optimisation est ici effectuée par une approche stochastique (stratégie d'évolution $(\lambda + \mu)$). Nous avons utilisé l'implémentation de l'algorithme ROGER (*ROC-based Genetic Learner*) [5, 19] pour extraire des hypothèses linéaires ; la complexité est $\mathcal{O}(n \ln n)$ où n est le nombre d'exemples.

Une retombée intéressante de l'optimisation stochastique est de fournir "gratuitement" un ensemble d'hypothèses ; chaque run lancé à partir d'une même base d'apprentissage fournit une hypothèse différente. Cette propriété a été utilisée pour construire un ensemble diversifié d'hypothèses et donc un ensemble diversifié d'attributs, dans l'esprit de l'apprentissage d'ensemble. En effet la diversité des hypothèses de l'ensemble est un facteur essentiel pour la qualité de prédiction de l'ensemble ; de même, la diversité des attributs d'une représentation favorise l'apprentissage ... jusqu'à un certain point : l'augmentation du nombre d'attributs mène à la malédiction de la dimensionalité, nous y reviendrons en section 5.

Formellement, à partir de chaque sous-ensemble $\mathcal{E}$ sont extraites : i) une hypothèse linéaire construite par SVM^{Light} ; ii) ℓ hypothèses linéaires construites par ROGER. Chacune de ces hypothèses, fonction de $\mathcal{X}$ dans $\mathbf{R}$, définit un nouvel attribut, et l'ensemble de ces nouveaux attributs définit ainsi une nouvelle représentation du domaine.

4.3 Influence de la segmentation

L'influence de la segmentation est étudiée en comparant les représentations issues i) de la partition des jobs fondée sur les utilisateurs ; ii) de la partition fondée sur les semaines.

Formellement, soit h_u (respectivement h_w) l'hypothèse extraite par SVM^{Light} du sous-ensemble de tâches associé à l'utilisateur u (resp. à la semaine w). La U-représentation (resp. W-représentation) associe à chaque tâche $\mathbf{x}$ un vecteur de réels $U(\mathbf{x})$ (resp. $W(\mathbf{x})$) défini par $(h_u(\mathbf{x}))$ (resp. $(h_w(\mathbf{x}))$) pour u (resp. w) variant dans la série des utilisateurs (resp. des semaines) ; soit N_u (N_w) la dimension des vecteurs $U(\mathbf{x})$ (resp. $W(\mathbf{x})$).

De la même manière, $h'_{u,i}$ (respectivement $h'_{w,i}$) indique la i-ème hypothèse extraite par ROGER du sous-ensemble de tâches associé à l'utilisateur u (resp. à la semaine w), pour $i = 1 \ldots \ell$. La U'-représentation (W'-représentation) associe chaque tâche $\mathbf{x}$ à un vecteur de réels $U'(\mathbf{x})$ ($W'(\mathbf{x})$) défini par $(h'_{u,i}(\mathbf{x}))$ $((h'_{w,i}(\mathbf{x}))$) pour $i = 1 \ldots \ell$ et u (w) variant dans l'ensemble des utilisateurs (resp. des semaines) ; soit $N'_u = N_u \times \ell$ ($N'_w = N_w \times \ell$) la dimension des vecteurs $U'(\mathbf{x})$ ($W'(\mathbf{x})$).

5 Clustering et double clustering

Dans une troisième étape, les représentations construites ci-dessus sont utilisées pour l'analyse exploratoire et le clustering des données. La qualité des clusters obtenus est traditionnellement évaluée par la pureté des clusters obtenus (par rapport aux deux classes de jobs satisfaisants ou en échec). Un autre critère d'évaluation issu des travaux de Meila [7], est donné par la stabilité des clusters.

Formellement, l'approche est évaluée par la stabilité des clusters i) en fonction de l'algorithme d'apprentissage utilisé, SVM^{Light}ou par ROGER ; ii) en fonction de la partition des données utilisée pour apprendre, respectivement fondée sur les utilisateurs, et sur les semaines (section 3.2). Nous rappellerons tout d'abord les critères de stabilité du clustering [7], en définissant une borne inférieure de stabilité. Dans un second temps, une approche de double clustering inspirée de Slonim et Tishby [6] est introduite pour limiter la redondance éventuelle des représentations apprises. En effet, si les nouveaux attributs sont par construction plus pertinents que les attributs initiaux (par rapport à l'objectif de discrimination entre jobs satisfaisants et jobs en échec), ils peuvent néanmoins être aussi redondants. Le problème est encore plus sévère pour les représentations extraites par ROGER, où ℓ hypothèses sont extraites d'un même sous-ensemble.

5.1 Stabilité d'un Clustering

Suivant Meila [7], un clustering $C = \{C_1, \ldots, C_K\}$ est représenté comme une matrice $J \times K$ $\widetilde{C}$ où $\widetilde{C}_{i,k}$ est 1 ssi le i-ème exemple appartient à C_k et 0 sinon. On définit $\widehat{C}$ en normalisant $\widetilde{C}$; en notant n_k la taille du cluster C_k,

$$\widehat{C}_{i,k} = \begin{cases} 1/\sqrt{n_k} & \text{si le } i\text{-ème exemple appartient à } C_k \\ 0 & \text{sinon} \end{cases} \quad (1)$$

La similarité entre deux clusterings $\widehat{C}$ et $\widehat{C}'$ définis sur le même ensemble de données est calculée à partir du produit scalaire de $\widehat{C}$ et $\widehat{C}'$:

$$S(\widehat{C}, \widehat{C}') = \|\widehat{C}^T \widehat{C}'\|^2_{Frobenius} = \sum_{i,j=1}^{K} n_{i,j}^2 \frac{1}{n_i n'_j} \quad (2)$$

où $n_{i,j}$ est le nombre de jobs dans $C_i \bigcap C'_j$, et n_i et n'_j sont respectivement la taille de C_i et C'_j.

Théorème
Avec les notations ci-dessus, la similarité $S(\widehat{C}, \widehat{C}')$ admet une borne inférieure et une borne supérieure :

$$K \geq S(\widehat{C}, \widehat{C}') \geq \frac{J}{(J-K+1)} \frac{1}{K} \tag{3}$$

Preuve : La borne supérieure $K \geq S(\widehat{C}, \widehat{C}')$ dérive immédiatement de Meila [7].
La borne inférieure est obtenue en remarquant d'abord que $n_{i,j} < min(n_i, n'_j)$ et $n_i, n'_j \leq (\frac{J}{K} + \delta)$, où $\delta \leq (J - K + 1 - \frac{J}{K})$. Donc $\frac{1}{n_i n'_j} \geq \frac{1}{(\frac{J}{K}+\delta)^2}$. D'autre part, la somme de $n_{i,j}^2$ atteint son maximum pour $n_{i,j} = \frac{J}{K^2}$ (quand tout $n_{i,j}$ est égal, sachant que $\sum_{i,j} n_{i,j} = J$). Il vient

$$\begin{aligned} S(\widehat{C}, \widehat{C}') &\geq \frac{1}{(\frac{J}{K}+\delta)^2} \sum_{i,j=1}^{K} n_{i,j}^2 \\ &\geq \frac{1}{(\frac{J}{K}+\delta)^2} \frac{J^2}{K^2} = \frac{J}{J+\delta K} \\ &\geq \frac{J}{J+(J-K+1-\frac{J}{K})K} = \frac{J}{(J-K+1)K} \end{aligned}$$

ce qui conclut la preuve □

La borne supérieure permet d'évaluer la stabilité du clustering et donc la qualité de l'approche [7] : plus $S(\widehat{C}, \widehat{C}')$ est proche de K, meilleurs sont $\widehat{C}$ et $\widehat{C'}$. Cependant, cette borne repose sur l'hypothèse de "bonne-séparation" ; si cette hypothèse n'est pas vérifiée, rien ne dit que la borne supérieure peut être atteinte. La borne supérieure permet même dans ce cas de qualifier la stabilité des clusters.

5.2 Double clustering

Dans le cas particulier de la représentation basée sur les hypothèses de ROGER, nous devons réduire la dimensionnalité. Comme la taille des données interdit les méthodes classiques de réduction de dimensionnalité, nous proposons une approche inspirée du double clustering de Slonim et Tishby [6]. La méthode du double clustering est la suivante (en considérant indépendamment la U'-représentation et le W'-représentation) :

1. Chaque attribut est considéré comme un vecteur de $\mathbb{R}^J$, où J indique le nombre total de jobs ;
2. Les attributs de la U'-representation (respectivement W'-representation) sont regroupés en clusters en appliquant l'algorithme K-means avec la distance Euclidienne sur $\mathbb{R}^J$;
3. Pour chaque cluster d'attributs F_i, on construit l'attribut moyen f_i : si $|F_i|$ est la taille de cluster F_i,

$$f_i : X \mapsto \mathbb{R} \qquad f_i(\mathbf{x}) = \frac{1}{|F_i|} \sum_{h \in F_i} h(\mathbf{x})$$

4. La représentation U'-clustered (respectivement W'-clustered) associe à chaque job $\mathbf{x}$ le vecteur $f_i(\mathbf{x})$, où i varie parmi les clusters d'attributs construits sur les U'-representation (resp. W'-representation) ;
5. On applique finalement l'algorithme K-means avec la distance euclidienne sur les représentations U'-clustered (resp. W'-clustered).

Cette procédure définit au total deux clusterings des jobs, basés respectivement sur la représentation U'-clustered et W'-clustered. Ces clusterings sont évalués par leur stabilité, comme détaillé dans la section précédente.

6 Protocole expérimental

Cette section décrit les données et le protocole expérimental suivi pour la validation de l'approche présentée.

6.1 Méthodologie

Les données disponibles décrivent 248.967 jobs ; 86% sont sélectionnés par échantillonnage uniforme pour former l'ensemble d'apprentissage qui servira à l'induction constructive et la définition des différentes représentations. Les jobs restants (32.836) forment l'ensemble de test sur lequel seront appliqués les changements de représentations et le clustering.
La première étape (section 3.2) est de segmenter les données d'apprentissage. La segmentation fondée sur les utilisateurs produit $N_u = 36$ sous-ensembles mono-utilisateur (contenant les jobs soumis par un seul utilisateur) ; de même, la segmentation fondée sur les semaines produit $N_w = 47$ sous-ensembles mono-semaine.
La seconde étape crée les nouvelles représentations à partir des sous-ensembles ainsi définis. Sur chaque sous-ensemble, SVM^{Light} est lancé avec les paramètres par défaut et construit une hypothèse linéaire. L'ensemble des N_u (resp. N_w) hypothèses apprises des sous-ensembles mono-utilisateur (resp. mono-semaine) définit la U-représentation (resp. W-représentation). De même, ROGER est lancé $\ell = 50$ fois sur chaque sous-ensemble, avec les paramètres décrits en Table 1 ; il construit des hypothèses linéaires optimisant le critère de l'aire sous la courbe ROC (section 4). L'ensemble des $N'_u = N_u \times \ell$ (resp. $N'_w = N_w \times \ell$) hypothèses apprises des sous-ensembles mono-utilisateur (resp. mono-semaine) définit la U'-représentation (resp. W'-représentation).

TAB. 1 – Paramètres de ROGER

Taille de population	N. de parents : 10
	N. de enfants : 70
Max N. d'évaluations	1000
Croisement	uniforme, taux 60%
Mutation	self-adaptive, taux 100%

La troisième étape consiste à changer la représentation des données de test, utilisant les U, W, U' and W' représentations définies ci-dessus. Les U' et W' représentations des données de test sont soumises au double clustering (section 5.2) en faisant varier le nombre T de clusters des attributs de 6 à 36 par pas de 2, produisant les représentations $U' - c$ et $W' - c$.

Finalement, les quatre représentations $U, W, U' - c$ et $W' - c$ sont utilisées pour le clustering des données de test, effectuées par K-means en faisant varier K de 4 à 32 par pas de 1. Dans chaque cas, les expériences sont répétées avec différentes initialisations indépendantes de K-means.

6.2 Critères d'évaluation

Deux critères d'évaluation sont considérés. Le premier concerne uniquement la stabilité des clusters appris avec les quatre différentes représentations (section 5.1), qui permet d'évaluer la cohérence des données (en fonction de l'initialisation de K-means) et la qualité de l'induction constructive.

Le second critère concerne l'interprétation des clusters obtenus par rapport aux connaissances du domaine. L'expert peut en effet fournir des catégories plus fines des jobs en échec ; il est donc intéressant de voir si l'approche proposée permet de les retrouver, voire d'identifier d'autres catégories interprétables.

Les jobs en échec (70% des données considérées) comprennent les cas où l'exécution n'est pas déclarée terminée, ou bien l'utilisateur ne peut pas obtenir le résultat [20]. De très nombreuses causes d'échec existent, incluant les problèmes de matériel, les erreurs de configuration, les bogues de l'intergiciel et les erreurs des utilisateurs. Trois classes d'échec sont identifiées par l'expert (mais non disponibles dans les données) :

NAR No Adequate Resource. Cet échec se produit à l'étape de placement ; il peut résulter i) d'une erreur de l'utilisateur, imposant des conditions irréalistes ou incompatibles sur la configuration d'une machine ; ii) d'une vraie absence de ressource ; iii) d'une instabilité de l'intergiciel, qui ne découvre pas à temps les ressources disponibles.

GNG Generic and Non Generic errors. Cet échec comprend les erreurs non spécifiques d'un job telles que "le proxy de l'utilisateur a expiré" ou "la taille du job dépasse les limites fixées". Cette catégorie d'erreur comprend aussi les problèmes d'entrées/sorties "Cannot download/upload/retrieve".

ABU Aborted By User. Cet échec comprend les cas où l'utilisateur demande l'annulation du job, soit qu'il se soit ravisé, soit que le job ne retourne pas de résultats dans un temps raisonnable.

La Table 2 décrit la distribution des catégories de jobs dans les données d'apprentissage et de test ; seule l'information succès ou échec est fournie aux algorithmes d'apprentissage.

7 Validation expérimentale

Cette section décrit et discute les résultats de l'approche présentée, en considérant d'abord la pureté et l'interprétation des clusters, puis leur stabilité.

TAB. 2 – Les données utilisées

		Total	Apprentissage	Test
Succes		88,131	78,131	10,000
Echec	NAR	117,369	100,000	17,369
	GNG	40,906	36,000	4,906
	ABU	2,561	2,000	561
Total		248,967	216,131	32,836

7.1 Critère de pureté

La première évaluation des clusters obtenus porte sur leur pureté relativement aux quatre catégories définies dans la Table 2. L'étiquette d'un cluster est l'étiquette majoritaire des jobs du cluster.

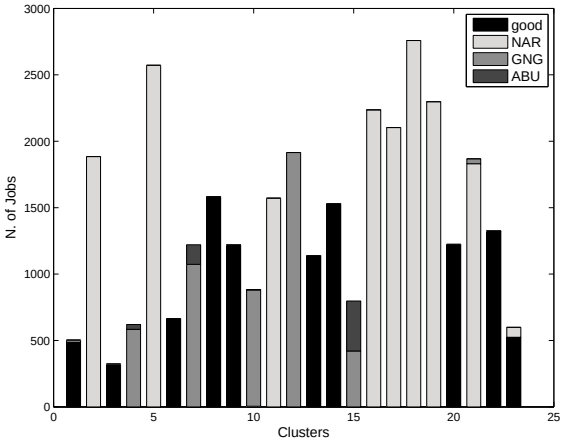


FIG. 2 – W-représentation (SVM^{Light}, mono-semaine) : clusters pour $K = 23$

La Fig. 2 montre les clusters obtenus avec la W-représentation pour $K = 23$. La plupart des clusters sont purs relativement aux quatre catégories (la catégorie *good* correspond aux succès et les trois catégories *bad* correspond aux échecs). Quelques clusters présentent un mélange de jobs GNG et ABU.

De la même manière, la Fig. 3 présente les clusters obtenus avec la $W' - c$-representation pour $T = 30$ clusters d'attributs et $K = 29$ clusters de jobs. Certains clusters (1, 8, 9, 13, 14) sont des clusters purs, ne contenant que des jobs ayant réussi. Le 6ème cluster contient des jobs qui ont rencontré quelques erreurs, par exemple "cannot download/upload" pendant leur exécution, mais qui ont finalement réussi après resoumission. Ceci explique l'impureté du cluster 6 ; les impuretés des autres clusters des jobs ayant réussi sont expliquées par de façon similaire.

Le 24ème cluster inclut seulement les jobs GNG, où les erreurs arrivent avant qu'ils n'aient été placés, par exemple : "input position not available", "Problems during rank evaluation on resources". Les autres jobs de GNG dans quelques clusters ont des échecs du type "Retry Count Hit" après re-soumissions multiples, à cause d'erreurs diverses dès le premier essai.

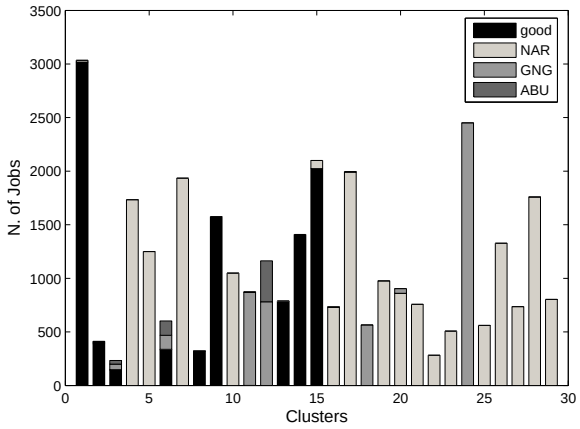


FIG. 3 – $W' - c$-representation (ROGER, mono-semaine) : clusters pour $T = 30$ et $K = 29$

Le taux d'erreur associé à chaque catégorie (succès, NAR, GNG et ABU) est la fraction des jobs de cette catégorie appartenant à un cluster d'étiquette différente ; la moyenne des taux d'erreur sur les quatre catégories est reportée Fig. 4. On reporte également la moyenne obtenue en excluant la catégorie ABU (2% des données), qui est la plus difficile à identifier ; en effet elle présente des erreurs similaires à celles qui surviennent dans les autres cas d'échec, et qui ont motivé la décision de l'utilisateur d'interrompre le job.

Les Fig. 4.(a) et 4.(b) illustrent respectivement l'influence du nombre K de clusters d'exemples et du nombre T de clusters d'attributs sur le taux d'erreur, pour la représentation $U' - c$. Comme on peut s'y attendre, l'erreur diminue avec la finesse du clustering des exemples (lorsque K augmente). La valeur de T (la finesse du clustering des attributs) ne fait pas de différence dans l'intervalle de valeurs considéré ; cependant, les résultats obtenus pour la U' représentation, i.e. sans clustering des attributs, sont très mauvais (omis faute de place) et justifient ainsi l'intérêt du double clustering.

Les Fig. 4.(c) et 4.(d) illustrent de même l'influence du nombre de clusters d'exemples et d'attributs en considérant la représentation $W' - c$.

Il est visible que les différences des taux d'erreur entre le minimum et la moyenne, ainsi que l'influence apparente du nombre T de clusters d'attributs proviennent surtout de la catégorie ABU. En rejetant cette catégorie (2% des données), la stabilité de l'erreur est excellente, que ce soit par rapport au nombre K de clusters de jobs ou par rapport au nombre T de clusters d'attributs.

En résumé, l'approche présentée donne des résultats satisfaisants ; l'erreur relative des clusters obtenus à partir des représentations $U, W, U' - c$ et $W' - c$ est inférieure à 10% si on exclut la catégorie multi-forme et ultra-minoritaire ABU, alors même que les catégories NAR et GNG n'étaient pas connues lors de l'induction constructive.

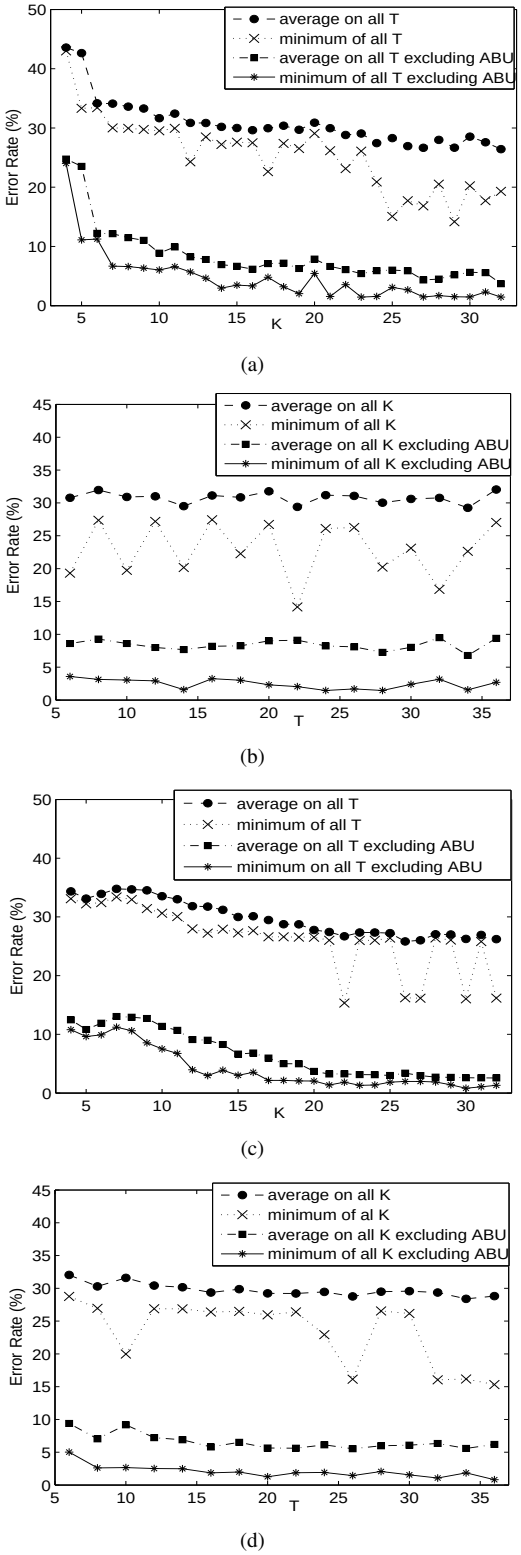


FIG. 4 – Le taux d'erreur : l'influence du nombre K de clusters d'exemple et du nombre T de clusters d'attributs pour les représentations $U' - c$ (a),(b), et $W' - c$ (c), (d)

7.2 Critère de stabilité

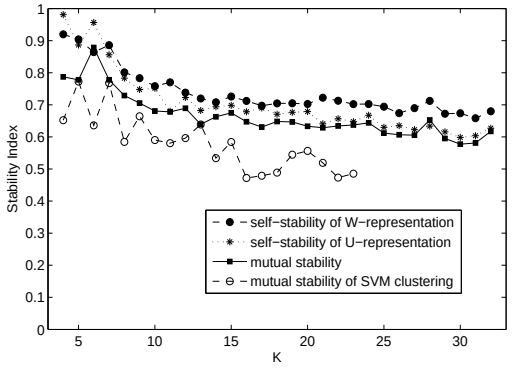
Le deuxième critère d'évaluation concerne la stabilité des clusters obtenus par l'approche proposée. La stabilité est classiquement mesurée par $D(C, C') = \frac{1}{K} S(C, C')$ (section 5.1).

Notons $C_u^{K,T}$ le clustering fondé sur la $U' - c$ représentation, obtenu pour un nombre K (resp. T) de clusters d'exemples (resp., d'attributs). L'auto-stabilité de la représentation $U' - c$ mesure l'influence du nombre de clusters d'attributs pour un nombre K de clusters de jobs donné : elle est définie comme la moyenne des stabilités $D(C_u^{K,T}, C_u^{K,T'})$, pour $T \neq T'$ variant dans l'intervalle expérimental $[6, 36]$ (section 6). Définissant de même $C_w^{K,T}$ le clustering fondé sur la $W' - c$ représentation, obtenu pour un nombre K (resp. T) de clusters d'exemples (resp., d'attributs), l'auto-stabilité de $W' - c$ est donnée par la moyenne des stabilités $D(C_w^{K,T}, C_w^{K,T'})$.

Dans le même esprit, la stabilité mutuelle des représentations $U' - c$ et $W' - c$ est définie pour un K donné par la moyenne des stabilités $D(C_u^{K,T}, C_w^{K,T})$ pour T variant dans $[6, 36]$; symétriquement, la stabilité mutuelle des représentations $U' - c$ et $W' - c$ est définie pour un T donné par la moyenne des stabilités $D(C_u^{K,T}, C_w^{K,T})$ pour K variant dans [4,32].

Enfin, la stabilité mutuelle des représentations U et W est définie pour un K donné.

La Fig. 5 montre la stabilité en fonction de K des différents clusterings considérés ; la stabilité ne dépend pas du nombre T de clusters d'attributs définis dans le double clustering (résultats omis faute de place).



(a)

FIG. 5 – Auto-stabilité et stabilité mutuelle des clusters. Auto-stabilité des représentations $U' - c$ et $W' - c$, Stabilité mutuelle de $U' - c$ et $W' - c$, et stabilité mutuelle de U et W, en fonction de K.

7.3 Discussion

Les résultats obtenus permettent de valider l'approche proposée selon plusieurs critères.

Le premier aspect concerne tout d'abord le passage à l'échelle de l'approche proposée. La première étape de segmentation rend possible l'usage de l'apprentissage discriminant (de complexité quadratique dans le cas de SVM^{Light}, en $nlogn$ dans le cas de ROGER, voir ci-dessous). Les autres étapes sont linéaires en fonction de la taille des données et des paramètres K et N considérés. Plus précisément, le temps de l'apprentissage discriminant est de l'ordre de 10 minutes (PC Pentium IV 1,7 GHz, 2Gb mémoire) ; le temps de clustering varie de 6 à 120 minutes en fonction des paramètres T et K.

Le second aspect concerne la qualité des clusters obtenus en terme de pureté, i.e. la capacité de l'approche proposée à retrouver des catégories qui n'étaient pas connues. La pureté des clusters obtenus a été considérée satisfaisante (Fig. 2 et Fig. 3) par rapport aux catégories NAR et GNG, particulièrement quand le nombre K de clusters augmente ; de surcroît, les clusters suggèrent des raffinements de ces catégories[5] qui sont considérés pertinents par l'expert.

La pureté relative à la catégorie ABU est nettement moins satisfaisante, ce qui s'explique par le polymorphisme de cette catégorie et la multiplicité des raisons conduisant un utilisateur à vouloir annuler un job.

L'algorithme d'apprentissage supervisé (SVM^{Light} ou ROGER) fait peu de différences à ce stade. Dans le cas de ROGER, ℓ hypothèses sont extraites de chaque sous-ensemble et une étape de compression des hypothèses obtenues est nécessaire ; cependant la valeur du paramètre T intervenant dans le double clustering n'est pas sensible dans le cadre expérimental. Dans le cas de SVM^{Light}, une seule hypothèse est extraite de chaque sous-ensemble et aucune compression n'est donc nécessaire. De manière intéressante, la différence intervient au niveau de la controllabilité et de la variance du temps de calcul. La controllabilité de ROGER s'effectue par le nombre d'itérations (Table **??**) ; la controllabilité de SVM^{Light} s'effectue par le paramètre de précision [4], dont la maitrise demande plus d'expertise. Indépendamment, la variance du temps de calcul de SVM^{Light} est d'un ou plusieurs ordres de grandeur supérieure à celle de ROGER.

Enfin, le troisième critère de validation concerne la stabilité du clustering obtenu par les différentes méthodes. La stabilité est évaluée par rapport à sa borne sup (1) et sa borne inf ($1/K^2$ si K est le nombre de clusters). On voit ainsi que la stabilité est généralement excellente pour de petites valeurs de K ($K < 8$). L'auto-stabilité des représentations $U' - c$ et $W' - c$ est élevée, ce qui confirme la faible sensibilité du paramètre T.

La stabilité mutuelle des représentations $U' - c$ et $W' - c$ est significativement meilleure que celle des représentations U et W, ce qui est attribué à la diminution de la variance due au double clustering. La stabilité décroit lentement avec K ; la stabilité observée pour $K \simeq 30$ est encore très élevée ($\sim 0,6$, à comparer avec la borne inf. de .001, Fig. 5.(a)).

[5] Problems during rank evaluation ; Job proxy is expired.

8 Conclusion et perspectives

Cet article a présenté essentiellement deux contributions. La première est générale et concerne l'analyse exploratoire d'une grande masse de données, sur laquelle on dispose de catégories pauvres (ici, les catégories de jobs en échec ou ayant réussi). La seconde est relative au domaine de l'application, l'informatique autonomique et plus précisément la modélisation comportementale d'une grille de calcul.

Du point de vue général, le point saillant concerne la mise au point d'une méthodologie *any-time* lorsque les algorithmes de la littérature sont inapplicables pour des raisons de taille et de complexité. Cette méthodologie repose sur la segmentation des données en fonction de critères *a priori*, visant i) à réduire les sources d'hétérogénéité ; ii) à rendre faisable l'usage d'algorithmes d'apprentissage supervisés (de complexité non linéaire). L'application de ces algorithmes sur les sous-ensembles des données ainsi définis, exploitant les catégories connues, permet d'amorcer le processus d'induction constructive et de recherche d'une représentation métrique pertinente. La pertinence de la représentation et de la métrique permettent ensuite, via l'usage d'algorithmes linéaires, le raffinement et la découverte de nouvelles catégories.

D'un point de vue applicatif, l'approche présentée a permis le traitement de grandes masses de données. Les résultats obtenus ont été validés tout d'abord à partir d'une analyse de stabilité, fournissant des garanties sur la qualité des clusters trouvés selon deux protocoles d'induction constructive indépendants (SVM^{Light}+ clustering, et ROGER+ double clustering). L'approche a également été validée par sa capacité i) à retrouver des catégories connues de l'expert mais qui n'étaient pas connues des algorithmes d'apprentissage ; ii) à proposer des catégories nouvelles et plus fines, qui ont été considérées comme pertinentes par l'expert. Par exemple un des clusters contient les jobs de test de fonctionnement lancés automatiquement, restreints à certaines périodes spécifiques de l'année, et dont la cause d'échec est *rank evaluation*. L'étape suivante est de comprendre ce qui est arrivé pendant ces périodes.

Les perspectives ouvertes par le travail présenté concernent premièrement l'analyse théorique du processus combiné de redescription/double clustering. L'objectif est de caractériser ce processus par rapport à des approches telles que le clustering spectral, sachant que cette dernière n'est pas applicable sur des données de grande taille pour des raisons de complexité.

D'un point de vue applicatif, une autre perspective concerne l'utilisation des clusters de jobs pour le profilage des utilisateurs, considérant un utilisateur comme un ensemble de jobs. Les clusters ainsi définis sont immédiatement utilisables pour un soutien personnalisé aux utilisateurs. De manière symétrique, la même approche peut être utilisée pour profiler les semaines d'utilisation de la grille. L'objectif est ici de détecter des schémas de charge dans l'usage régulier de la grille, et à terme comprendre le comportement collectif des utilisateurs et son évolution.

Références

[1] J. O. Kephart, and D. M. Chess. *The vision of autonomic computing computer*. 36(1) :41-50, 2003.

[2] I. Rish, M. Brodie, S. Ma, et al. Adaptive diagnosis in distributed dystems. *IEEE Trans. on Neural Networks (special issue on Adaptive Learning Systems in Communication Networks)*. 16 :1088-1109, 2005.

[3] N. Palatin, A. Leizarowitz, A. Schuster, and R. Wolff. Mining for misconfigured machines in grid systems. In *KDD '06*, 687–692. ACM Press, 2006.

[4] T. Joachims. Making large-Scale SVM Learning Practical. *Advances in Kernel Methods - Support Vector Learning*. B. Schölkopf and C. Burges and A. Smola (ed.), MIT-Press, 41-56, 1999.

[5] M. Sebag, N. Lucas, J. Azé. Impact studies and sensitivity analysis in medical data mining with ROC-based genetic learning. *ICDM*. 637-640, 2003.

[6] N. Slonim, and N. Tishby. Document clustering using word clusters via the information bottleneck method. *Research and Development in Information Retrieval*. 208-215, 2000.

[7] M. Meila. The uniqueness of a good optimum for K-means. *ICML*. 625-632,2006.

[8] J. B. Tenenbaum, V. D. Silva, and J. C. Langford. A global geometric framework for nonlinear dimensionality reduction. *Science*. 290 :2319-2323, 2000.

[9] S. Roweis, L. Saul. Nonlinear dimensionality reduction by locally linear embedding. *Science*. 290 :2323-2326, 2000.

[10] V. de Silva and J. Tenenbaum. Global versus local methods in nonlinear dimensionality reduction.In *NIPS*. 705-712, 2002.

[11] M. Sugiyama. Local fisher discriminantanalysis for supervised dimensionality reduction. *ICML*. 905-912, 2006.

[12] X. Yang, H. Fu, H. Zha, et al. Semi-supervised nonlinear dimensionality reduction. *ICML*. 1065-1072, 2006.

[13] K. Q. Weinberger, J. Blitzer, and L. K. Saul. Distance metric learning for large margin nearest neighbor classification. In *NIPS*. 1473-1480, 2005.

[14] C. Ding, and X. He. K-means clustering via principal component analysis. *ICML*. 225-232, 2004.

[15] A. Ng, M. Jordan, and Y. Weiss. On spectral clustering : Analysis and an algorithm. In *NIPS*. 849-856, 2001.

[16] M. Kearns, and M. Li. Learning in the Presence of Malicious Errors, in *SIAM J. Comput*. 22 :807-837, 1993.

[17] V. N. Vapnik. *The Nature of Statistical Learning*. Springer Verlag, 1995.

[18] S. Rosset. Model selection via the AUC. *ICML*. 89-96, 2004.

[19] K. Jong, J. Mary, A. Cornuejols, et al. Ensemble feature ranking. In *Proc. ECML/PKDD*. 267-278, 2004.

[20] K. Neocleous, M. Dikaiakos, P. Fragopoulou, et al. Failure management in grids : the case of the EGEE infrastructure. Technical report, Institute on System Architecture, CoreGRID - Network of Excellence. 2006.