La fouille des usagers du web par application de l'algorithme Apriori sur les fichiers logs

SLIMANI Yacine, MOUSSAOUI Abdelouahab
Laboratoire de Recherche des Systèmes Intelligents, Faculté des Sciences de l'Ingénieur,
Université Ferhat Abbas, Cité Maabouda, Sétif -19000
slimani_y09@univ-setif.dz

Résumé : Le processus de la fouille des usagers du web est basé sur l'analyse des fichiers Logs. Dans cet article on présente les déférentes phases à savoir : Prétraitement, découverte et analyse du modèle. Le prétraitement consiste a nettoyé le fichier log et sa structuration en sessions. Dans la phase découverte de connaissance on a appliqué l'Algorithme Apriori pour l'extraction des règles d'association et enfin une analyse des résultats obtenus.

Mots clés : Entrepôt de données, Fouille des usagers du web, Règles d'associations, Apriori.

Introduction

Dans les dernières années il y a eu une croissance exponentielle du nombre de sites web et de leurs usagers. On recense à la fin du mois de mars 2009 environ de 360 Millions d'internautes (dont 3,5 Millions en Algérie) pour 231,5 Millions de Sites au monde.

Cette croissance phénoménale a produit une quantité de données énorme liées aux interactions d'utilisateurs avec les sites web, stockés par les serveurs web dans des fichiers Logs. Ces fichiers logs peuvent être utilisés par les administrateurs de sites web pour découvrir les intérêts de leurs visiteurs afin d'améliorer le service par l'adaptation du contenu et de la structure des sites à leurs préférences. L'analyse des fichiers logs permet à identifier des modèles du comportement des usagers, ce qui peut être exploité à la personnalisation du web.

Dans les phases de découverte et analyse de la connaissance, la fouille des usagers du web représente un champ de recherche pour découvrir les modèles comportementaux des usagers [6].

Dans un processus général de la fouille des usagers du web (WUM) on distingue trois phases principales : prétraitement de données, découverte du modèle et analyse du modèle [9].

Dans la phase du prétraitement de données, premièrement, les données sont nettoyées en enlevant l'information et le bruit non pertinents. Les données restantes sont arrangées d'une manière cohérente afin d'identifier d'une façon précise les sessions utilisateurs. Après l'identification des sessions utilisateurs, on applique l'algorithme Apriori affin d'extraire des règles d'associations qui définissent le comportement des usagers de notre site; et nous permettent de personnaliser ce derniers pour s'adapter aux besoins de nos utilisateurs.

1. Les fichiers Logs

Un fichier Log est un fichier texte qui contient les requêtes faites au serveur web enregistrées en ordre chronologique. Les formats les plus utilisés pour les fichiers log sont le CLF (Common Log Format) et le ECLF (Extended CLF). On a utilisé le standard ECLF, dont le format est comme suit :

41.200.89.109 - - [12/Oct/2008:20:18:23 +0100] "GET /citic2008/soumission.html HTTP/1.1" 200 23247 "http://www.univ-setif.dz/citic2008/index.html" "Mozilla/5.0 (Windows; U; Windows NT 5.1; fr; rv:1.9.0.3) Gecko/2008092417 Firefox/3.0.3"

où:1) le nom ou l'adresse IP de la machine appelante.

2) le nom et le login HTTP de l'utilisateur.

3) la date et l'heure de la requête.

4) la méthode utilisée par la requête (Get, Post, etc.) 5) l'URL de la requête.

6) Le protocole utilisé.

7) le statut de la requête.

8) la taille du fichier envoyé.

9) l'URL qui a référencé la requête.

10) l'Agent (navigateur et le système d'exploitation)

2. Architecture du processus de WUM

La structure fonctionnelle du processus de la fouille des usagers Web est structuré en six modules principaux comme représenter dans la Figure 1.

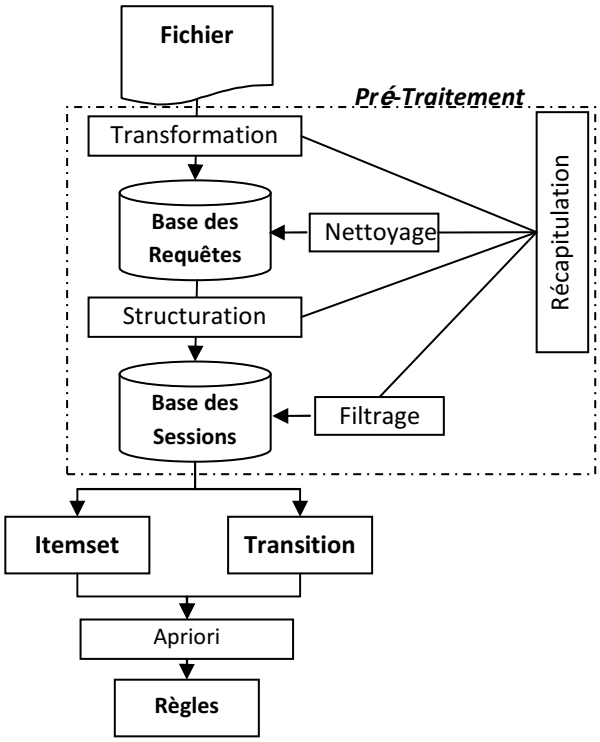

Figure1 : Architecture du processus de WUM

2.1. Module transformation des données

Le module transformation de données a comme entrée un fichier log sous sa forme textuelle brut, décrite dans la section 2. Une analyse lexicale permet d'extraire les différents champs de chaque ligne du fichier logs. En sortie, on obtient une table composée de plusieurs colonnes. Chaque colonne correspond à un champ spécifique du fichier logs (Figure 2).

Column Name	Datatype
Num	INTEGER
ip	VARCHAR(15)
ident	VARCHAR(20)
user	VARCHAR(20)
date	DATE
time	TIME
methode	VARCHAR(10)
url	VARCHAR(512)
protocole	VARCHAR(10)
statut	INTEGER
taille	INTEGER
refferer	VARCHAR(512)
agent	VARCHAR(512)

Figure 2 : Structure de la table Logs.

2.2. Module nettoyage des données

Le module nettoyage de données est prévu pour supprimer les enregistrements inutiles afin de maintenir seulement les données d'usagers qui peuvent être effectivement exploitées pour identifier le comportement de navigation des utilisateurs. Le choix des données à supprimer dépend de l'objectif ultime du système de personnalisation du Site. Dans notre cas, l'objectif est de développer un système WUM qui offre une personnalisation des liens dynamiques pour les visiteurs du site, d'où le système doit tenir compte seulement des enregistrements relatives à des requêtes explicites des usagers et qui représentent effectivement des actions des usagers. En conséquence, le module de nettoyage des données a été développé pour éliminer les requêtes suivantes :

2.2.1. Méthode différente de "GET" : En règle générale, les requêtes contenant une valeur différente de "GET" ne sont pas des requêtes explicites des utilisateurs, mais elles concernent souvent des accès avec CGI, des visites de robots, etc. Par conséquent, ces requêtes sont considérés comme non significatives et sont retirées du fichier journal d'accès.

2.2.2. Requêtes échouées et corrompues : Ces requêtes sont représentées par des enregistrements contenant un code d'erreur HTTP. Si la valeur du champ statut est 200, elle représente une requête réussie. Par contre les autres valeurs représentent une requête erronée (par exemple le statut 404 indique que le fichier demandé n'a pas été trouvé).

2.2.3. Requêtes d'objets multimédias : Dans le protocole HTTP, une requête d'accès est exécutée pour chaque fichier, image ou objet multimédia embarqué dans une page web demandé. En conséquence, une seule requête pour une page web produit souvent plusieurs entrées dans le journal, ce qui correspond à des fichiers téléchargés automatiquement sans une requête explicite de l'utilisateur.

Les requêtes de ce type de fichiers peuvent être facilement identifiées car elles contiennent l'extension du nom du fichier, comme .gif, .jpeg, .jpg, etc. La préservation ou la suppression de ces objets multimédia dépend du genre du site web à personnaliser et de leurs natures. En général, ces requêtes ne représentent pas une activité de l'usager dans le site, d'où elles sont censées être enlevées. Dans d'autres cas, l'élimination des requêtes d'objets multimédias cause une perte d'informations utiles.

2.2.4. Requête à l'origine des robots

Les fichiers journaux contiennent un certain nombre d'enregistrements correspondant aux requêtes provenant des robots. Les robots d'exploration du web sont des programmes qui parcourent automatiquement les sites web en suivant tous les hyperliens sur chaque page du site afin de mettre à jour l'index du moteur de recherche. Ces requêtes ne sont pas considérées comme des actions par les usagers et, par conséquent, doivent être supprimées. Pour identifier les requêtes des robots, le module de nettoyage des données met en œuvre deux heuristiques [8].

Tout d'abord, tous les enregistrements contenant le nom "robots.txt" dans les URL sont identifiés et supprimés. La seconde heuristique est basée sur le fait que les robots navigues dans les pages d'une façon automatique et exhaustive, de sorte qu'ils se caractérisent par une très haute vitesse de navigation (calculée par nombre total de pages visitées / total temps passé à visiter ces pages). Par conséquent, pour chaque adresse IP, on calcule la vitesse de navigation et toutes les requêtes ayant une valeur supérieure à un seuil (pages / seconde) sont considérées comme faites par des robots et sont donc éliminées. La valeur du seuil est établie par l'analyseur après l'examen des fichiers journaux.

Après nettoyage des données, seules les requêtes pour des ressources pertinentes sont conservées dans la base de données.

A la fin de cette étape, on définit $R = \{r_1, r_2, ..., r_{n_r}\}$ comme l'ensemble des ressources distinctes demandées à partir du site web en cours d'analyse.

2.3. Module structuration des données

Le module de structuration de données regroupe les requêtes du journal dans des sessions d'usagers. Une session est définie comme un ensemble limité de ressources accessibles par le même utilisateur pendant une visite. L'identification des sessions d'utilisateur depuis les fichiers logs est une tâche difficile parce que beaucoup d'utilisateurs peuvent utiliser le même ordinateur et le même utilisateur peut utiliser différents ordinateurs. Par conséquent, le principal problème est de savoir comment

identifier l'utilisateur. Pour des sites web exigeant l'enregistrement de l'utilisateur, le fichier de log contient le nom d'utilisateur qui peut être utilisé pour l'identification de 37l'utilisateur. Lorsque le nom de l'utilisateur n'est pas disponible, on identifie l'utilisateur par son adresse IP. On considère chaque adresse IP en tant qu'utilisateur différent (en prenant compte qu'une adresse IP pourrait être utilisée par plusieurs utilisateurs) [7].

On définit $U = \{u_1, u_2, ..., u_{n_u}\}$ comme l'ensemble de tous les utilisateurs (Ip) qui ont accédé au site web.

On exploite une méthode basée sur le temps pour identifier les sessions [2], on considère une session d'un utilisateur comme l'ensemble des accès provenant du même utilisateur dans un délai prédéfini. Cette période est définie en tenant compte d'un temps maximum $\Delta t_{\max}$ entre deux accès. D'ailleurs, afin de mieux gérer des situations particulières qui pourraient se produire (comme les utilisateurs accédant à plusieurs reprises pour la même page, dues à la lenteur des connexions réseau ou du trafic intense), un temps minimum $\Delta t_{\min}$ écoulé entre deux accès consécutifs est également fixé [4].

On définit une session d'utilisateur comme le triplé : $s^{(i)} = (u^{(i)}, t^{(i)}, r^{(i)})$ avec :

$u^{(i)} \in U$ représente l'identificateur de l'usager.

$t^{(i)}$ est le temps d'accès de l'ensemble de la session.

$r^{(i)}$ est l'ensemble de toutes les ressources (avec temps d'accès correspondant) demandé au cours de la i^{ème} session, à savoir :

$$r^{(i)} = ((t_1^i, r_1^i), (t_2^i, r_2^i), ..., (t_{n_i}^i, r_{n_i}^i)) \text{ avec } r_j^i \in R,$$

où les temps d'accès t_k^i a une seule ressource vérifie les contraintes suivants:

$$t_{k+1}^i \geq t_k^i \text{ et } \Delta t_{\min} < t_{k+1}^i - t_k^i < \Delta t_{\max}$$

Récapitulant, après la phase de structuration de données, un ensemble de ns sessions $s^{(i)}$ est identifié depuis le fichier log. On note l'ensemble des sessions identifié par:

$$S = (s^{(1)}, s^{(2)}, ..., s^{(n_s)}).$$

Une fois toutes les sessions ont été identifiés, le module de structuration des données présente un panneau qui répertorie tous les sessions extraites et permet à l'analyste de visualiser et éventuellement enregistrer les détails (adresse IP, les ressources demandées dans la session, la date et l'heure des requêtes) de chaque session d'utilisateur.

2.4. Module filtrage des données

Après l'identification des sessions utilisateurs, on effectue un filtrage des données pour supprimer les ressources les moins demandées. Pour chaque ressource r_i, on considère NS_i le nombre de sessions qui effectuent un appel à la ressource r_i et on calcule la valeur $NS = \max_{i..n_R} NS_i$. Ensuite, on définit le seuil ε, et tous les requêtes ayant NS_i < ε sont supprimées.

De cette façon, le module filtrage de données peut considérablement réduire le nombre des requêtes, ce qui facilite les traitements des prochaines phases de la fouille des usagers.

En plus le module de filtrage élimine toutes les sessions qui comportent seulement des ressources les moins demandées (la session de vient vide après la suppression des ressources les moins demandées).

2.5. Module Apriori

Après structuration du fichier Log sous forme de sessions, on va adapter les données obtenues pour l'application de l'algorithme des règles d'association.

2.5.1. Notations :

- **L'ensemble des items :** On définit l'ensemble des items comme étant l'ensemble des ressources, et on obtient I = $R = \{r_1, r_2, ..., r_{n_r}\}$ définit dans la section 3.2

- **La base de données :** On définit l'ensemble des transactions comme étant l'ensemble des sessions extraites dans le paragraphe 3.3, et on obtient $\beta = S = (s^{(1)}, s^{(2)}, ..., s^{(n_s)})$; Où chaque transaction est un ensemble de ressources (Requêtes) demandées pendant une session donnée.

- **k-Itemset :** est un Itemset contenant k items.

- **Une règle d'association :** est de la forme X⇒Y, où X⊆I, Y⊆I, et X∩Y=∅, où X et Y sont des itemsets.

- Le support de l'itemset X : est le nombre de transactions de la base β contenant X divisé par le nombre total de transactions.

$$\sup(X) = \frac{\left|\{t \in \beta / X \subseteq \beta\}\right|}{|\beta|}$$

- Le support d'une règle d'association X⇒Y : est le rapport entre le nombre de transactions de β contenant X⇒Y, et le nombre total de transactions.

$$\sup(X \Rightarrow Y) = \frac{\left|\{t \in \beta / X \cup Y \subseteq \beta\}\right|}{|\beta|}$$

- La confiance d'une règle : est le rapport entre le nombre de transactions de B contenant X⇒Y, et le nombre de transactions de B contenant X.

$$\text{conf}(X \Rightarrow Y) = \frac{\left|\{t \in \beta / X \cup Y \subseteq \beta\}\right|}{\{t \in \beta / X \subseteq \beta\}} = \frac{\sup(X \cup Y)}{\sup(X)}$$

- L'itemset X est un itemset fréquent si :

$$\sup(X) \geq \text{minsup}.$$

2.5.2. L'algorithme :

La recherche de règles d'associations dans un ensemble de transactions s'opère en deux temps :

1. On cherche les ensembles d'items fréquents, c'est-à-dire ceux qui apparaissent un nombre minimum de fois dans l'ensemble des transactions.

2. On génère les règles d'associations pertinentes, c'est-à-dire celles qui vérifient simultanément la contrainte minimale sur le support et la confiance.

La recherche de tous les sous-ensembles fréquents consiste à déterminer parmi l'ensemble de toutes les parties de $X = (X_1;X_2; \ldots ;X_p)$ les sous-ensembles fréquents, c'est-à-dire présents dans un nombre assez conséquent de transactions.

L'algorithme Apriori consiste à chercher des ensembles fréquents de cardinal k+1 à partir des ensembles fréquents de cardinal k. Ainsi pour trouver les ensembles fréquents ayant deux items, on utilisera exclusivement les ensembles fréquents ayant un item.

Le nombre d'ensembles fréquents diminue avec le nombre d'items : il y a moins d'ensemble fréquents à 2 items qu'à un item. Cette propriété permet ainsi de restreindre la taille de l'espace à explorer pour trouver tous les ensembles fréquents nécessaires à la deuxième étape de l'algorithme qui comporte deux points :

1. Pour chaque ensemble fréquent X_a on génère tous les sous-ensembles non vides.

2. Pour chaque sous-ensemble non vide $X_b \subset X_a$, on génère la règle ($X_b \Rightarrow (X_a - X_b)$) si $\frac{\text{support}(X_a)}{\text{support}(X_b)} > c_0$.

2.6. Module récapitulation des données

Le module récapitulation, génère des rapports à la fin de chaque traitement. Un premier rapport est généré après l'analyse du fichier log. Il synthétise le nombre total de requêtes du fichier log, le nombre de requêtes satisfaites et échouées, la taille d'octets transférées, etc.

Un autre rapport, est généré après le nettoyage des données, contient des informations comme le nombre de requêtes ayant une méthode différent que GET, le nombre de requêtes à des objets multimédias, le nombre de visites effectuées par des robots, etc.

Lorsque l'étape de structuration des données est terminée, un autre rapport est créé qui contient le nombre de sessions utilisateur extraites et le détail de chaque session.

Enfin, après que le filtrage des données est effectué, un rapport qui contient le nombre des requêtes Figurant dans le fichier journal avant et après le filtrage, le nombre des requêtes supprimer, le pourcentage de requêtes supprimé, etc.

3. Les résultats expérimentaux

L'analyseur des fichiers d'accès développé, a été testé sur des fichiers Logs stockés sur le serveur du site de l'Université Ferhat Abas Sétif (www.univ-setif.dz). Le fichier traité couvre l'activité du site pendant la période du 23/02/2008 au 12/03/2008).

3.1. Transformation des données

Les résultats de l'analyse préliminaire de ce fichier Logs sont indiqués dans le tableau 1, qui synthétise les résultats fournies par le module récapitulation des données dès que fichier Logs (sous sa forme textuel brut) est chargé et traité.

Taille du Fichier	177 494 706 Octets
Date début	12/10/2008 03:32:35
Date Fin	12/11/2008 09:53:17
Nombre de Lignes	675 490

Tableau 1 : Récapitulatif après transformation.

Le module nettoyage des données analyse le fichier log afin de déterminer les requêtes jugées non pertinentes parce qu'elles ne représentent pas des requêtes explicites des utilisateurs.

3.2.1. Requêtes d'objets multimédias

Le nombre de requêtes aux objets multimédias est très important. Comme il est représenté dans la Figure 3, elles représentent 82% des requêtes. Après nettoyage, seul 18% des requêtes sont maintenues dans la base.

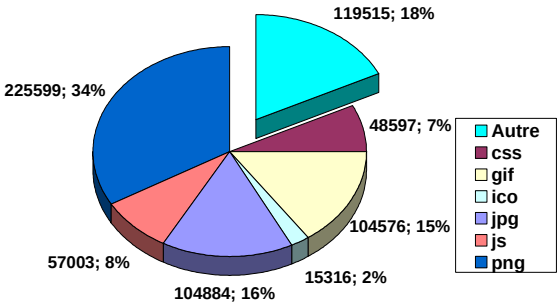

Figure 3 : Les objets multimédias.

3.2.1. Méthodes différentes de "GET" : Dans cette section on supprime les requêtes qui ont des méthodes d'accès différentes de "GET". La Figureure 4 présente le nombre de chaque type de méthode dans le fichier log. On remarque que le nombre de requêtes à supprimer est très petit par rapport au nombre total (0,75%).

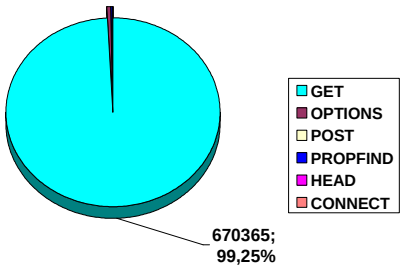

Figure 4 : Les requêtes par méthode

3.2.2. Requêtes échouées et corrompues : Les requêtes qui ont un statut différent de 200 sont considérées comme des requêtes erronées. On a trouvé 2% des requêtes avec code d'erreur 404 qui signifie des fichiers demandés mais non trouvés et 26% des requêtes avec code d'erreur 304 qui signifie des fichiers avec le problème de rafraîchissement dans le browser. A la fin du nettoyage 70% des requêtes sont retenues.

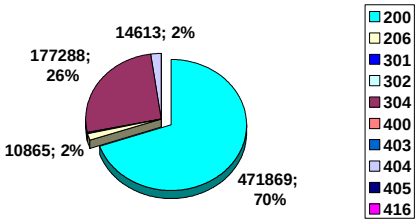

Figure 5 : Les requêtes par statut**.**

Le tableau 2, est un récapitulatif des différentes requêtes supprimées de la base de données.

Multimédia		**Méthode**		**Statut**	
.gif	104576	Options	2 838	206	10 865
.png	225 599	Post	1 010	301	549
.jpg	104 884	Propfind	542	304	177 288
.ico	15 316	Head	400	400	21
.rm	2	Connect	335	403	244
.tif	2			404	14 613
.css	48 597			405	15
.js	57 003			416	26
Som:	**555 979**	**Som:**	**5 125**	**Som:**	**203 621**
Autre	**119 511**	Get	**670 365**	200	**471 869**

Tableau 2 : Récapitulatif après nettoyage.

On signale que des chevauchements peuvent se produire entre deux catégories supprimées. Par exemple, une requête avec la méthode "HEAD" peut également être une requête pour un objet multimédia. Dans ces cas, les données du module récapitulatif compte deux fois la suppression de la requête, même si un seul enregistrement est supprimé du fichier.

Catégorie des requêtes nettoyées			**Nb**
Multimédia	Méthode ≠ Get	Statut ≠ 200	
x			**363 333**
	X		**4 540**
		X	**10 432**

x	**X**		**21**
x		**X**	**192 625**
	X	**X**	**564**
x	**X**	**X**	**0**
555 979	**5 125**	**203 621**	**571 515**

Tableau 3 : Chevauchement entre les Catégories des requêtes nettoyées.

3.3. Structuration des données

Après le nettoyage de la base, la première étape du module structuration de données est de définir les deux ensembles R et U :

- R l'ensemble des ressources distinctes demandées à partir du site web en cours d'analyse.

- U l'ensemble des usagers du site web.

Ensuite on applique l'algorithme de structuration pour déterminer les sessions en prenant compte des deux valeurs de $\Delta t_{\max}$ et $\Delta t_{\min}$ [3]. $\Delta t_{\min}$ pour la détection des robots et les aspirateurs, et $\Delta t_{\max}$ pour la détection des nouveaux sessions. Le tableau 4 récapitule les résultats obtenus.

Données en entée

Nb de Requêtes	103975
Nb d'IP (U)	8 676
Nb d'URL (R)	7 707

Paramètres

$\Delta t_{\max}$	30min
$\Delta t_{\min}$	05sec

Nb Sessions: 17 379

Tableau 4 : Récapitulatif après structuration.

3.4. Filtrage des données

La dernière phase du processus du prétraitement est le filtrage des ressources, on supprime les URLs les moins demandées avec un seuil ε et on a obtenu les résultats illustrés par le tableau 5.

	Avant	**ε = 5**	**ε = 50**	**ε = 100**
Nb Req.	103 975	66 792	48 571	29 481
Nb URL	7 707	1 607	193	105
Nb IP	8 676	2 843	2 829	1 480
Nb Sess.	17 379	4 665	4 571	2 199

Tableau 5 : Récapitulatif après filtrage.

3.5. Apriori

La dernière phase de notre travail, consiste à appliquer l'algorithme des règles d'associations sur la base des sessions trouvées dans la section 4.4.

L'algorithme Apriori, a comme données en entrée un ensemble d'items (l'ensemble des ressources) et une base de données des transitions.

Avec les paramètres :

- seuil de filtrage des requêtes **ε = 100;**
- Support min des items **minsup** = **5**;
- Confiance min des règles **confmin** = **60**.

L'Algorithme Apriori opère en deux phases:

- La cherche des ensembles d'items fréquents, on trouve 105 1-items, 60 2-itmes et 10 3-items.
- La recherche des règles d'associations, a permet d'extraire deux catégories de règles :

1ère Catégorie de règles :

R_3 , R_{26} → R_1 : support = 2 , Confiance = 89.

R_3 , R_{34} → R_1 : support = 2 , Confiance = 92.

R_2 , R_6 → R_1 : support = 4 , Confiance = 73.

R_3 , R_{26} → R_1 : support = 2 , Confiance = 89.

R_3 , R_{26} → R_1 : support = 2 , Confiance = 89.

2ème Catégorie de règles :

R_5 , R_{17} → R_1 : support = 3 , Confiance = 90.

R_5 , R_{31} → R_1 : support = 2 , Confiance = 90.

Sachant que ces ressources ont la sémantique suivante:

R_1 : Page d'accueil

R_2 : Page de Formation Poste Graduation

R_3 : Page du Concours d'accès à la Post Graduation (MAGISTER) 2008-2009

R_{26} : Document Word sur le Concours de la faculté des sciences économiques et sciences de gestion.

R_{34} : Document Word sur le Concours de la faculté des sciences.

R_5 : Galerie d'images.

R_{17} : Page 1 de la Galerie d'images.

R_{31} : Page 2 de la Galerie d'images.

Donc on distingue clairement qu'il y a deux types de navigations :

- Les usagers qui ont été intéressés par les concours de poste graduation et Magistère. Et on doit bien souligner que la période d'étude (du 12/10/2008 au 12/11/2008) est la période des concours dans les déférentes facultés.

- La deuxième catégorie de navigation qui a été décelé, et la navigation entre les déférentes galeries d'images.

3.5. Activité du site

L'analyse des accès des utilisateurs dans l'axe du temps permet de trouver les heures où le serveur est le plus chargé. Comme illustre la Figure 6, l'accès au serveur

accroît considérablement vers 10 h du matin et commence a baissé a partir de 07h de l'après midi. L'administrateur peut décider depuis ce graphe si le matériel utilisé et la bonde passante du serveur est suffisante pour satisfaire les besoins de leurs usagers.

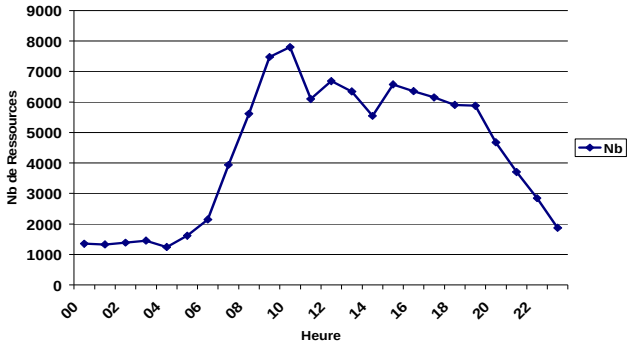

Figure 6 : Fréquence d'accès par heure du jour.

La Figure 7 présente les accès des utilisateurs pendant le mois d'étude. La première remarque est que l'activité du site est nulle pendant les 3 jours de la fête du 1er Novembre, le serveur était arrêtée pour maintenance. Le deuxième point, on constate que la charge du site augmente considérablement pendant le milieu de la semaine le lundi et le mardi. On satisfait en moyenne 4500 requêtes valides pour ces journées.

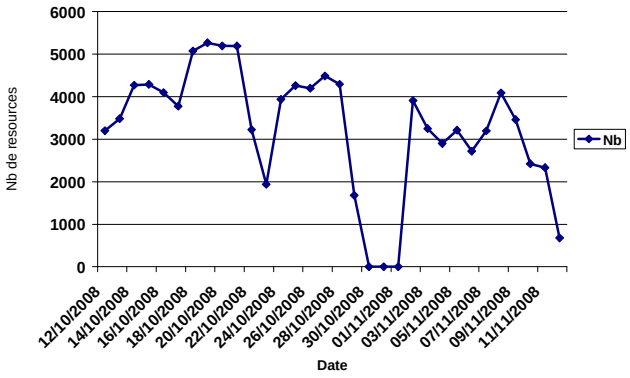

Figure 7 : Fréquence d'accès par jour

Conclusion et perspective

Les résultats expérimentaux obtenus avec le prétraitement des fichiers Logs, a permet non seulement de réduire considérablement la taille des fichiers Logs, mais

également à regrouper les requêtes web dans un certain nombre de sessions utilisateur ce qui peut aider à déterminer le comportement de l'utilisateur d'une manière significative. En effet, une fois les sessions des utilisateurs ont été identifiées,
on les utilise pour extraire le degré d'intérêt des utilisateurs pour chaque ressource web. Sachant que le degré d'intérêt à une ressource est strictement liés à la fréquence d'accès à cette ressource. Après l'achèvement de l'étape du prétraitement et la structuration des requêtes en sessions, l'algorithme Apriori a été utilisé pour extraire des règles d'associations entre les sessions des utilisateurs. Un intérêt particulier pour le concours de poste graduation et magistère a été identifié, ainsi que la consultation des déférentes galeries d'images. Les résultats obtenus sont très significatifs, mais on a comme perspective d'analyser le comportement des usagers par d'autres algorithmes comme les arbres de décisions.

Références

[1] Ardissono,L. and Torasso, P.. Dynamic User Modeling in a Web Store Shell, In: Proceedings of the 14th Conference ECAI, Berlin, Germany, p. 621-625 (2000).

[2] Cooley R.. Web Usage Mining: Discovery and Application of Interesting Patterns from Web Data. PhD thesis, University of Minnesota (2000).

[3] Nasraoui O.. World Wide Web Personalization. In J. Wang (ed), Encyclopedia of Data Mining and Data Warehousing, Idea Group (2005).

[4] Paliouras G. et al. Large-scale mining of usage data on Web sites. AAAI Spring Symposium on Adaptive User Interface, Stanford, California, p.92-97 (2000).

[5] [5] Pei, J. et al. Mining access patterns efficiently from web logs. in: Pacific-Asia Conference on Knowledge Discovery and Data Mining, p. 396–407 (2000).

[6] Pierrakos D. et al. Web usage mining as a tool for personalization: a survey. User Modeling and User-Adapted Interaction, 13(4), p. 311-372 (2003).

[7] Suryavanshi B.S. et al. A Fuzzy Hybrid Collaborative Filtering Technique for Web Personalization. In Proc. of 3rd Workshop on Intelligent Techniques for Web Personalisation (ITWP'05).

[8] Tan,P. N. and Kumar, V.. Discovery of Web Robot Sessions Based on their Navigational Patterns, Data Mining and Knowledge Discovery, 6(1), p. 9 -35 (2002).

[9] Tanasa D. and Trousse B.. Advanced Data Preprocessing for Intersites Web Usage Mining. In IEEE Intelligent Systems, 19(2), p. 59-65 (2004).