

# **IMAGE LOG**

## **Livrable Unique 4 - 11 mai 2014**

Polytech Nantes – Département Informatique

Binôme étudiant :

- Fabien RICHARD
- Valentin PROUST

Tuteur enseignant :

- Pascale KUNTZ
- Marc GELGON

Coordinateur :

- Jean Pierre GUEDON

Organisme commanditaire : ORANGE LABS

Tuteur industriel :

- Franck MEYER

## Clause de confidentialité

Les élèves-ingénieurs :

M. Fabien RICHARD , né le 07/06/1992 à Rennes

et

M. Valentin PROUST , né le 06/02/1992 à Luçon

s'engagent à ne pas publier ni divulguer de quelque façon que ce soit les informations scientifiques, techniques ou commerciales recueillies ou obtenues par eux au cours de la réalisation du projet décrit dans ce présent rapport, sans l'accord écrit préalable de l'organisme commanditaire.

Cet engagement vaut pour la durée du projet et les 12 mois qui suivent son expiration.

Les élèves-ingénieurs s'engagent à ne conserver, emporter ou prendre copie d'aucun document ou logiciel, de quelque nature que ce soit, appartenant à l'organisme commanditaire, sauf accord de ce dernier.

Cette confidentialité peut s'appliquer aux soutenances de projet des phases 1 et 3 qui dans ce cas, et sur demande écrite de l'organisme commanditaire, se dérouleront à huis clos.

A Nantes, le 27 octobre 2013

"Lu et approuvé"

Signature



"Lu et approuvé"

Signature



## Charte contre la fraude et le plagiat

Définitions :

**La fraude** : moyen quelconque pour ne pas être honnête lors d'un devoir surveillé, d'un rendu de projet ou de TP, seul ou en groupe. Pour chaque évaluation réalisée des élèves ingénieurs, la note personnelle ou de groupe doit refléter au mieux l'état des connaissances ou compétences acquises.

**Le plagiat** : c'est l'utilisation non mentionnée de contenu intellectuel déjà réalisé par une tierce personne ou groupe de personnes en vue de réutilisabilité illicite pour ne pas avoir soi même à développer ce contenu. Le plagiat n'est pas plus tolérable ni acceptable que la fraude : en plus de faire croire que l'on est l'auteur de ce que l'on n'a pas fait, on dépouille le véritable auteur de ses droits intellectuels ce qui devient un délit dans la société du savoir. La bonne attitude consiste à beaucoup se documenter mais toujours citer ses sources (textes, code, rendu de tp, etc.).

La fraude et le plagiat sont passibles de sanctions qui peuvent aller jusqu'à l'expulsion de l'Université.

**La bonne attitude** consiste à beaucoup se documenter mais à toujours citer ses sources.

- Tout travail d'un(e) étudiant(e) doit être personnel.
- Lorsque l'on utilise un passage d'un livre, d'une revue ou d'une page Web (traduit ou non), il doit être mis entre guillemets avec mention de la source et de la date.
- Lorsque l'on utilise des images, des graphiques, des données, etc. provenant de sources externes, celles-ci doivent être mentionnées.
- Lorsqu'un travail produit pour un cours est réutilisé pour un autre cours, il convient d'en demander l'autorisation.

## Introduction

Ceci est la quatrième version du rapport. Dans ce rapport nous décrirons l'avancée du travail, une présentation de l'entreprise, le contexte du projet, ses objectifs principaux ainsi que le planning prévisionnel. Il comporte également les problématiques environnementales et commerciales du projet, la révision de notre planning ainsi qu'une explication de notre modélisation. Enfin ce rapport présente le dossier de tests, le manuel d'utilisation et de réutilisation est une analyse critique du projet.

## Plan et annexes

### 1) Présentation de l'entreprise

### 2) Contexte du projet

### 3) Objectifs globaux du projet

### 4) Modèle du domaine

### 5) Estimation de l'effort

*annexe 1 : Justification de l'effort de chaque tache*

*annexe 2 : Gantt du projet*

### 6) Analyse des risques

### 7) Avancement du projet

*annexe 3 : Rapports hebdomadaire*

*annexe 4 : Journal d'actions*

*annexe 5 : Explication différence temps prévu vs effectif*

### 8) Révision du planning

*annexe 6 : Diagramme de Gantt révisé*

### 9) Architecture logicielle

*annexe 7.1 : Diagramme UML Modèle révisé*

### 10) Contexte commercial du projet

### 11) Contexte environnemental du projet

*annexe 13 : Rapport environnemental du projet*

### 12) Tests

*annexe 9.1 : Tests unitaires*

*annexe 9.2 : Tests recette*

*annexe 9.3 : Tests de temps de génération*

### 13) Manuel de déploiement et d'utilisation

*annexe 10 : Manuel de déploiement et d'utilisation*

### 14) Réutilisabilité du projet

*annexe 11 : Manuel de réutilisation*

### 15) Bilan et analyse personnelle sur le projet

*annexe 12 : Diagramme de Gantt du travail effectif*

## 1) Présentation de l'entreprise

Orange Labs est la division recherche et développement du groupe Orange. Orange est une entreprise française de télécommunications. Elle emploie près de 172 000 personnes, dont 105 000 en France, et sert près de 226 millions de clients dans le monde.

Fin 2012, Orange détient 7 493 brevets au niveau mondial, dont 291 déposés sur les 12 derniers mois.

## 2) Contexte du projet

Orange souhaite améliorer son système de recommandation automatique, en particulier pour son service de vidéo à la demande. Le principe des systèmes de recommandation vidéo est de suggérer aux utilisateurs les films susceptibles de les intéresser le plus, à partir de fichiers regroupant les achats des gens ou leurs notations sur les films visionnés. Pour cela, Orange utilise des méthodes de factorisation de matrices complexes et confidentielles. Ces algorithmes fournissent pour chaque film une coordonnée en deux dimensions.

Pour le moment quatre personnes travaillent sur la factorisation de matrice à Orange Labs. Ce projet a débuté au printemps 2013, suite aux travaux de thèse de Franck Meyer.

Le problème est de réussir à visualiser en deux dimensions sur une image le résultat de ces factorisations. En effet, le nombre de films étant très grand, il est impossible de générer une image de telles dimensions. La solution consiste donc à faire du “clipping”, c'est à dire découper la grande image en sous images plus petites.

Orange Labs nous demande de développer un utilitaire de génération et de visualisation de ces images qui ne sera dans un premier temps pas destiné au grand public mais aux chercheurs qui travaillent sur la factorisation de matrices. Elle permettra à l'équipe de chercheurs de mieux comprendre le fonctionnement des algorithmes de recommandation.

Le poids de ce projet est faible par rapport aux autres projets de Orange Labs car ils sont généralement de 6 hommes/mois. Ce projet nous a été confié car il est réalisable en un temps limité.

En résumé, notre projet s'inscrit dans le cadre de la recherche et du développement d'une grosse entreprise mais représente un poids faible en comparaison aux autres projets.

### 3) Objectifs globaux du projet

Notre travail consiste à développer une application en Java qui permet, à partir de fichiers d'achats ou de notations des utilisateurs, de générer une image et de naviguer dedans. L'image représente la proximité des films entre eux.

Par exemple, Toy Story et Pocahontas qui sont deux films de Disney seront peut être proches sur l'image. L'utilité de la visualisation des résultats des algorithmes du système de recommandation est, par exemple, de pouvoir optimiser les méthodes de factorisation pour améliorer les suggestions de films et pouvoir offrir un contenu personnalisé aux utilisateurs.

### 4) Modèle du domaine

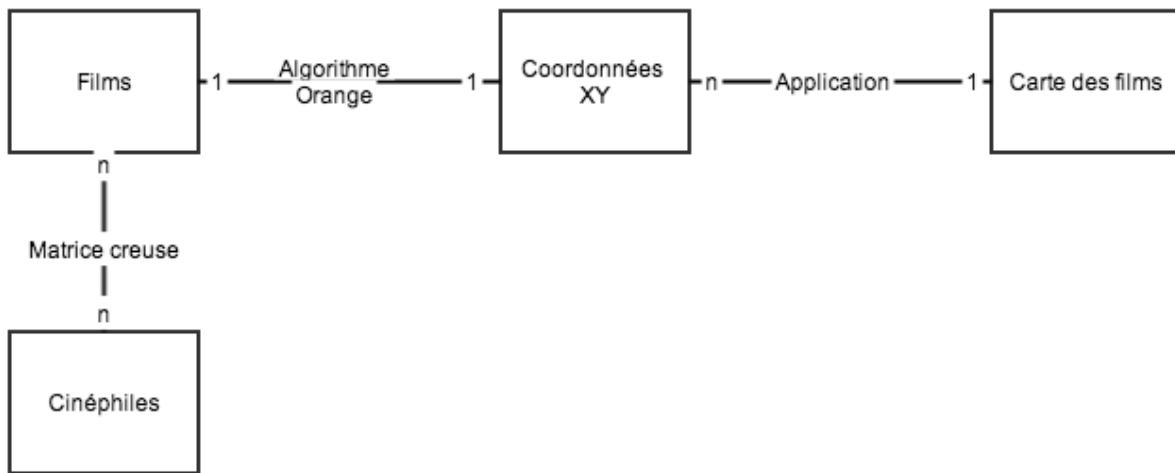


Fig 1 : Modèle du domaine

L'utilitaire demandé par le client ne spécifie pas un besoin d'interaction avec l'utilisateur pour le moment. Il est cependant possible que nous ayons à redéfinir des cas d'utilisation comme par exemple la possibilité pour l'utilisateur de choisir l'emplacement du fichier des logs et/ou l'emplacement de l'enregistrement de l'image en sortie.

### 5) Estimation de l'effort

Nous avons estimé le SLOC (nombre de lignes de code) de notre projet à 1000 (500 pour la génération de l'image et 500 pour la visualisation).

## Projet transversal en collaboration avec une entreprise 2013-14

Grâce au paramétrage de COCOMOII nous avons pu obtenir une estimation de l'effort (en hommes/mois) et des délais (en mois)

<http://csse.usc.edu/tools/COCOMOII.php>

Software Size	Sizing Method	Source Lines of Code				
		Source Lines of Code				
SLOC	% Design Modified	% Code Modified	% Integration Required	Assessment and Assimilation (0% - 8%)	Software Understanding (0% - 50%)	Unfamiliarity (0-1)
New 1000						
Reused	0	0				
Modified						
<b>Software Scale Drivers</b>						
Precededness	Very Low	Architecture / Risk Resolution	Low	Process Maturity	Nominal	
Development Flexibility	High	Team Cohesion	Extra High			
<b>Software Cost Drivers</b>						
<b>Product</b>		<b>Personnel</b>		<b>Platform</b>		
Required Software Reliability	Nominal	Analyst Capability	Nominal	Time Constraint	Nominal	
Data Base Size	Nominal	Programmer Capability	Nominal	Storage Constraint	Nominal	
Product Complexity	Nominal	Personnel Continuity	Nominal	Platform Volatility	Nominal	
Developed for Reusability	Nominal	Application Experience	Nominal	Project		
Documentation Match to Lifecycle Needs	Nominal	Platform Experience	Nominal	Use of Software Tools	Nominal	
		Language and Toolset Experience	Nominal	Multisite Development	Nominal	
				Required Development Schedule	Nominal	

Fig 2 : Paramétrage de COCOMOII

**Results**

**Software Development (Elaboration and Construction)**

Effort = 2.9 Person-months  
 Schedule = 5.2 Months  
 Cost = \$0

Total Equivalent Size = 1000 SLOC

Fig 3 : Résultats de COCOMOII

La complexité du projet détermine sa taille (en lignes de code).

La taille du projet (SLOC) détermine l'effort nécessaire (en hommes/mois).

L'effort sert à estimer les délais du projet (en mois).

Les délais sont utiles pour calculer le nombre de personnes nécessaires :

Nombre de personnes nécessaires = délais / effort

Dans notre cas : nombre de personnes =  $5.2 / 2.9 = 1.79$  personnes  $\rightarrow$  2 personnes.  
Ce qui correspond bien à notre projet.

Il a été difficile de faire une estimation COCOMOII pour les raisons suivantes :

- nous n'avons aucun projet précédent similaire permettant d'évaluer le SLOC
- nous avons une bibliographie à rédiger qui n'entre pas en compte dans le SLOC
- notre projet est un projet de recherche et développement

Nous avons donc choisi de diviser le projet en tâches et en sous tâches et d'estimer l'effort total en additionnant l'effort de chaque sous tâche. Nous avons tout d'abord posé une liste des différentes tâches à effectuer : bibliographie, modules à implémenter, documents à rédiger, réunions... Nous avons ensuite produit une estimation en temps pour chaque tâche. Cette estimation se base sur nos expériences passées et sur les conseils des tuteurs.

Nous avons choisi de mesurer l'effort en heures par semaine. La raison de ce choix est que cette mesure est proportionnelle et adaptée à notre projet. Par exemple, la mesure Hommes/An ne nous donnerait pas une bonne représentation de la réalité. A l'inverse, la mesure en nombre d'heures par jour ne nous donnerait pas assez de flexibilité dans la gestion de notre temps.

*voir annexe 1 : Justification de l'effort de chaque tache*

Nous tenons à jour un journal des actions où nous notons à chaque fois que l'on travaille sur le projet transversal, la tâche effectuée et le nombre d'heures passées sur la tâche. Nous pourrons donc à chaque fin de sprint savoir si l'estimation était juste, et réajuster le planning prévisionnel.

*voir annexe 2 : Gantt du projet*

## 6) Analyse des risques

### Risque sur les délais de livraison

La solution finale a peu de risque de ne peut être livrée à temps car nous espérons obtenir une solution de base assez rapidement qui pourra être améliorée en fonction du temps restant.

### Risque sur les besoins fonctionnels

La solution a des risques de ne pas correspondre aux attentes du tuteur entreprise car le cahier des charges n'est pas très précis.

### Risques liés à la technique

Il existe un risque élevé au niveau de l'affichage des noms de films car nous craignons que les noms de films se superposent en groupes et rendent la lecture impossible. Nous ne disposons pas pour l'instant des données permettant de tester la superposition.

## 7) Avancement du projet

Lors des premières semaines, nous avons concentré nos efforts sur une étude de la faisabilité pour limiter les risques liés à la technique : nous devions en effet nous assurer que la génération d'image avec 10 000 coordonnées de films était possible en temps et en lisibilité.

*voir annexe 3 : rapports hebdomadaire*

*voir annexe 4 : journal d'actions*

Nous avons jusqu'à cette date respecté le diagramme de Gantt prévisionnel. Les échanges avec le tuteur entreprise ont été réguliers. Cela nous a permis de revoir au fur et à mesure des livrables de fin de sprint, les spécifications fonctionnelles de l'outil. Cependant nous avons choisi de réaliser les sprints sur la recherche des plus proches voisins plus tôt que prévu sur la demande du tuteur entreprise. Pour bien comprendre nos choix pour le diagramme de Gantt révisé, nous avons rédigé un document qui explique les différences entre les temps estimés et les temps effectifs pour la réalisation de chaque tâche.

*voir annexe 5 : explication différence temps prévu vs effectif*

## 8) Révision du planning

Il reste deux sprints avant la date de livraison. Nous avons donc révisé le diagramme de Gantt pour réaliser les spécificités fonctionnelles prioritaires du tuteur entreprise.

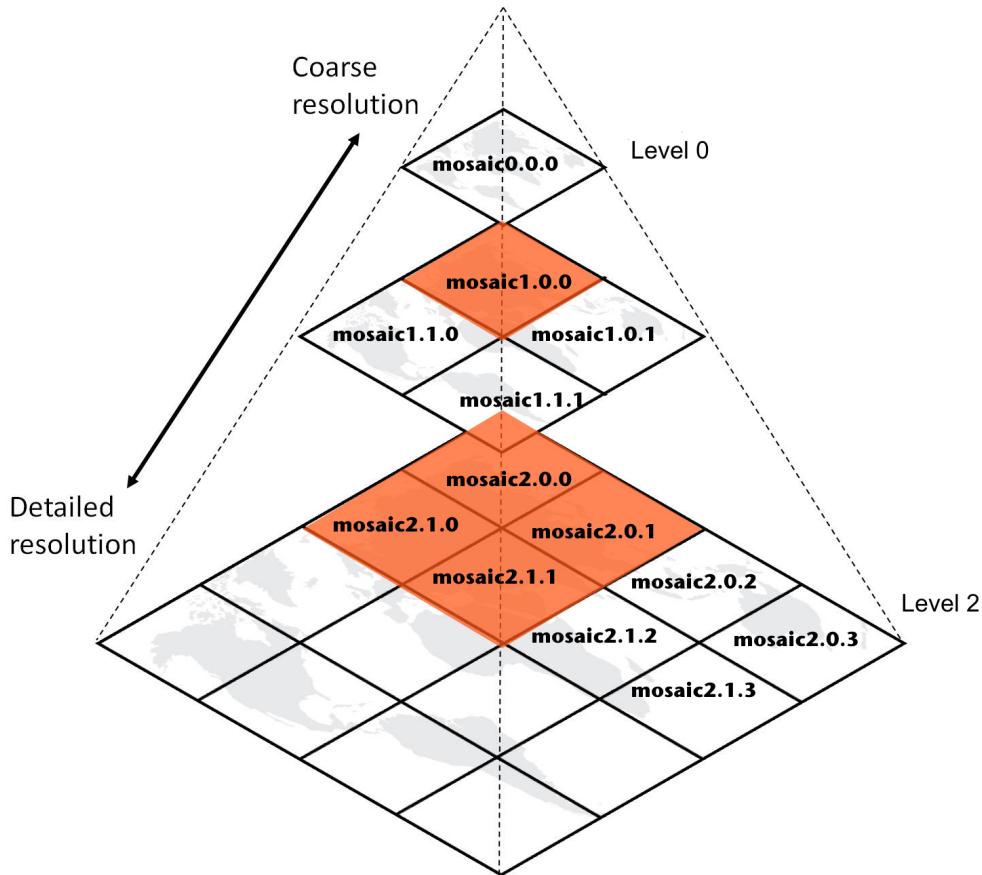
*voir annexe 6 : Diagramme de Gantt révisé*

## 9) Architecture logicielle

Nous avons choisi de décomposer le code de notre application en deux parties : package Modèle et package Contrôleur. Le but de cette séparation du code est de pouvoir rapidement savoir où se situe une méthode mais également de rendre le code plus lisible pour les éventuelles personnes qui travailleront dessus une fois notre livrable final rendu. Enfin cela permet une bonne évolutivité du code. Par exemple, si une partie visualisation des mosaïques

devait être intégrée au programme, il serait naturel de créer un package Vue qui viendrait s'ajouter aux packages Modèle et Contrôleur.

Pour générer les différents niveaux de mosaïques nous avions deux approches possibles : l'approche itérative et l'approche récursive.



*Fig. 1 Rappel du principe de génération des niveaux de mosaïques*

Le nom d'une mosaïque est défini de la manière suivante : mosaic[ZoomLevel].[Rang].[Colonne]

Nous avons choisi une génération récursive des mosaïques. Un avantage important de cette méthode est la rapidité. En effet, dès que les quatre mosaïques filles qui composent une mosaïque mère ont été générées, elles peuvent être assemblées et à leur tour composer une mosaïque mère. Nul besoin d'attendre que tout un niveau de mosaïque ait été généré pour passer au suivant. Le second avantage est que seuls les mosaïques qui contiennent des films sont générées. On évite ainsi de générer des mosaïques vides.

Déroulement de l'algorithme récursif pour l'exemple illustré en fig. 1 :

- On génère mosaic2.0.0, mosaic2.0.1, mosaic2.1.0 et mosaic2.1.1 (en rouge).

## Projet transversal en collaboration avec une entreprise 2013-14

- Puis elles sont assemblées pour former mosaic1.0.0 (en rouge).
- C'est ensuite au tour de mosaic2.0.2, mosaic2.0.3, mosaic2.1.2 et mosaic2.1.3 d'être générées
- Puis assemblées pour former mosaic1.0.1
- Et ainsi de suite

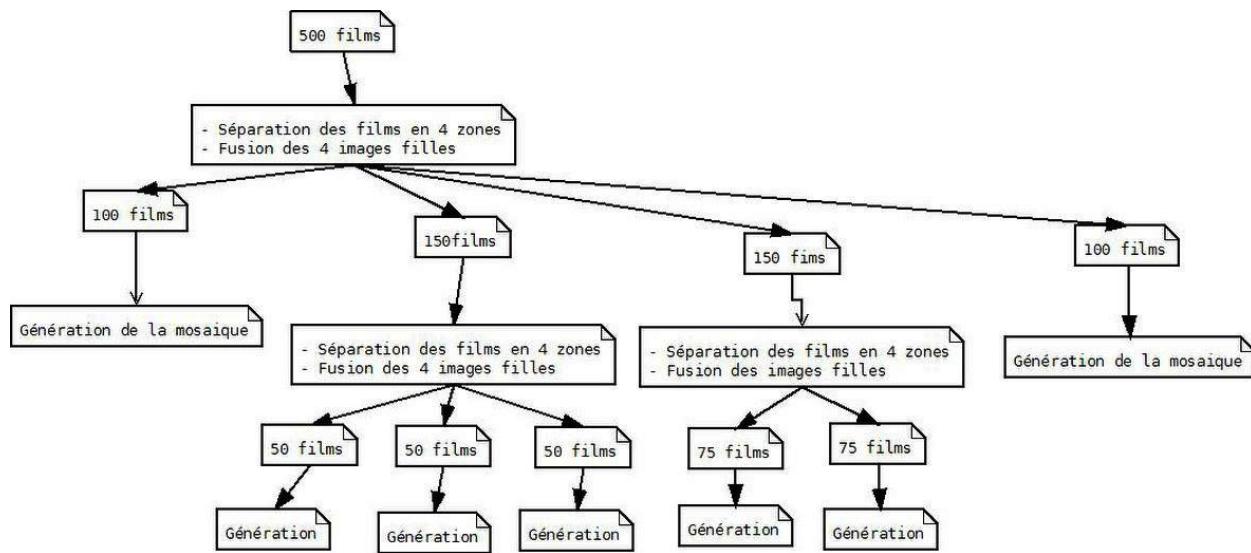


Fig 2. Illustration du déroulement de l'algorithme récursif

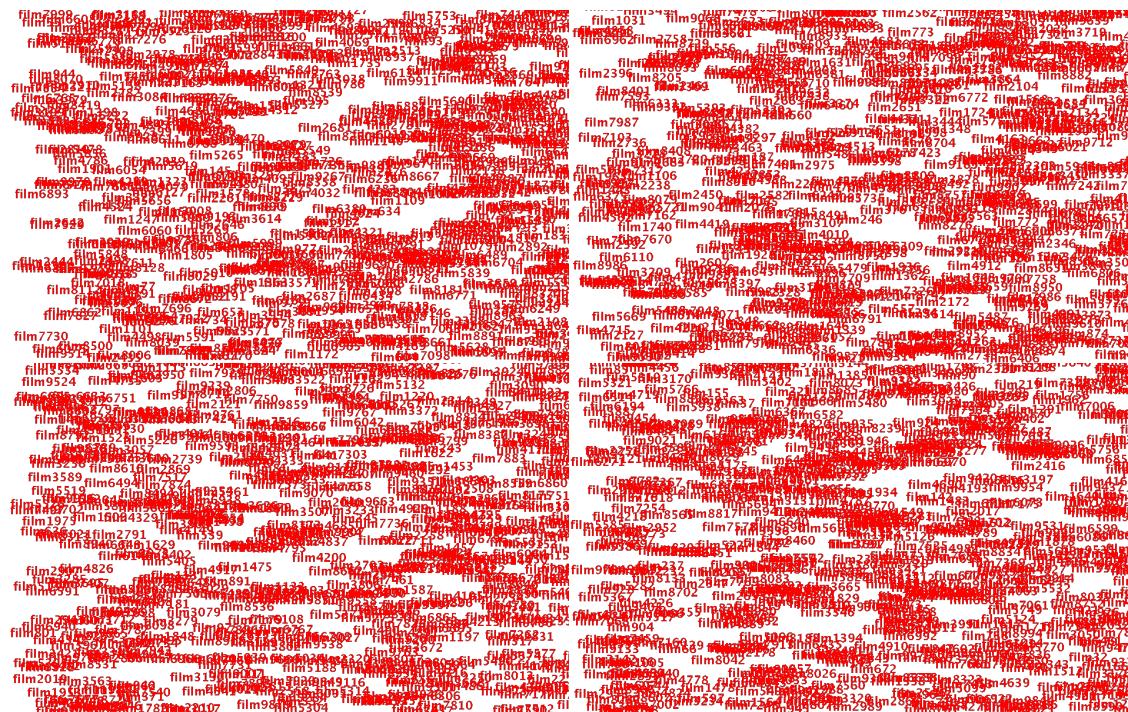


Fig 3. Exemple de mosaïque assemblée à partir de ses filles (ici mosaic1.0.0)

Il nous reste un travail sur les marges important pour pouvoir gérer le chevauchement des titres entre 2 mosaïques et ainsi éviter les démarcations visibles sur la figure 3.

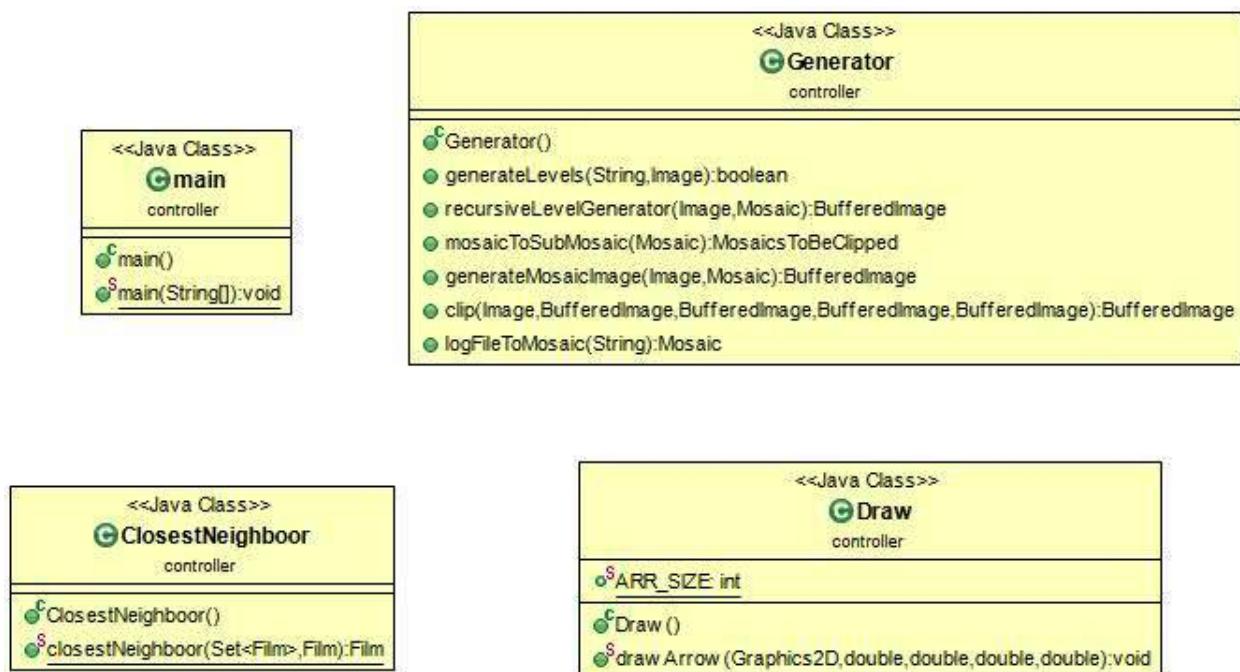
Description du fonctionnement du contrôleur :

- 1) Lecture du fichier de texte de logs de films
- 2) Génération récursive des mosaïques
  - Pour chaque mosaïque, tant que le nombre de films qu'elle contient est supérieur à 100, on la sépare en 4 sous mosaïques filles
  - Si le nombre de films qu'elle contient est inférieur à 100, on génère la mosaïque et on la retourne
  - En remontant, les 4 sous mosaïques sont fusionnées

Pour bien comprendre l'architecture logicielle de notre programme, nous avons réalisé deux diagrammes UML. Le premier est un diagramme UML du Modèle qui permet de bien voir les différents composants du modèle et les liens qu'ils ont entre eux.

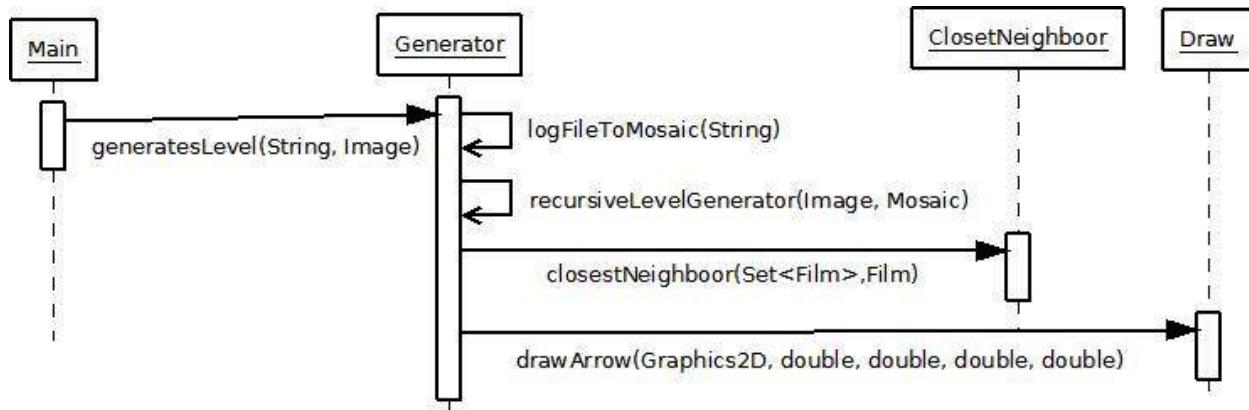
*voir annexe 7.1 : diagramme UML Modèle révisé*

De plus, nous avons réalisé un diagramme UML du Contrôleur pour pouvoir illustrer les différentes classes qui utilisent le modèle pour générer les mosaïques :



*Annexe 7.2 : diagramme UML Contrôleur révisé*

Nous avons également réalisé un digramme séquence qui permet de voir les différents échanges entre les classes dans un ordre chronologique :



*Fig 4. Diagramme séquence illustrant la génération des mosaïques*

La classe Generator est le coeur de notre application. C'est cette classe qui appelle les méthodes contenues dans d'autres classes comme la classe Draw qui est chargée de faire tous les dessins sur les mosaïques (points, arcs des plus proches voisins, titres de film etc) ou la classe ClosestNeighbor qui est chargé de retourner le plus proche voisin d'un Film dans un Set de Films. Chacune de ces classes utilise le Modèle pour créer un objet Film ou un objet Mosaic par exemple.

## 10) Contexte commercial du projet

En 2009, Amazon a annoncé que son outil d'aide à la navigation a généré 30% de ses bénéfices. Nous souhaitons savoir dans quelle mesure un système de recommandation pour un service de vidéo à la demande peut avoir des répercussions sur les ventes. De plus, notre travail portant sur l'outil de recommandation, nous souhaitons savoir si les consommateurs sont plus sensibles à des imagettes, une liste, ou une carte pour visualiser la recommandation. Enfin, nous souhaitons savoir s'il est possible de proposer des recommandations personnalisées tout en respectant la vie privée des consommateurs.

## 11) Contexte environnemental du projet

L'efficacité des systèmes de recommandation est en constante augmentation et permet aux entreprises d'augmenter leur bénéfices, mais leur coût énergétique est immense. En effet la factorisation de matrices pour générer les recommandations les plus pertinentes nécessitent une puissance de calcul immense. On peut alors se demander quelles stratégies pourraient être mises en place pour informer l'utilisateur sur la consommation énergétique d'un site internet. On

peut par exemple penser à une échelle allant de A à G comme pour les appareils électroménagers qui serait affichée sur les sites internet, pouvant motiver les utilisateurs à n'utiliser que les moins gourmands.

Nous avons choisi de nous intéresser au contexte environnemental du projet. C'est cette problématique que nous développons dans le rapport HES.

*voir annexe 13 : Rapport environnemental du projet*

## 12) Tests

Nous avons réalisé deux types de tests : des tests unitaires au fur et à mesure de l'implémentation des classes et des tests recettes sur l'outil fonctionnel. Pour les deux types de tests, nous avons fait en sorte que la rédaction des tests ne soit pas été réalisée par la personne qui effectue ces tests. Ainsi on évite que certaines erreurs soient oubliées par négligence.

### Tests unitaires

Nous n'avons pas utilisé de module de test. Pour chaque classe créé, nous avons écrit une classe de tests qui implémente tous les cas de tests.

*voir annexe 9.1 : Tests unitaires*

### Tests recette

Nous avons conçu les tests recette à partir du cahier des charges.

*voir annexe 9.2 : Tests recette*

### Tests de temps de génération

Notre projet a une problématique de complexité en temps assez marquée due au nombre important de films qu'il peut être nécessaire de traiter. C'est pourquoi nous avons réalisé des test de temps de génération de mosaïques. Ces tests sont fonction de la taille en pixels des mosaïques ainsi que du nombre de films contenus dans le fichier de log. Ils nous permettent d'indiquer au client quels sont les temps de génération auxquels il doit s'attendre selon le nombre de films qu'il souhaite traiter.

*voir annexe 9.3 : Tests de temps de génération*

### **Analyse critique de la phase de test**

Nous avons été rigoureux sur les phases de tests unitaires, ce qui nous a permis de limiter le temps de débugage de l'application.

En revanche les difficultés que nous avons rencontrées pendant les phases de test ont été les suivantes :

- Difficulté à tester la fonction récursive du fait de sa nature récursive
- Quelques oublis dans la définition des cas de test unitaires comme le test d'existence du fichier de log par exemple

## **13) Manuel de déploiement et d'utilisation**

Nous avons choisi de réaliser un manuel de déploiement et d'utilisation au format HTML et au format texte. En effet, le format HTML permet une navigation dynamique et rapide et offre une compatibilité intéressante (il suffit d'avoir un navigateur web pour pouvoir le consulter). Le format texte nous a permis de pouvoir intégrer le manuel de déploiement et d'utilisation à ce Livrable Unique.

*voir annexe 10 : Manuel de déploiement et d'utilisation*

## **14) Réutilisabilité du projet**

Nous avons également choisi le format HTML pour la documentation du projet. Nous pensons que l'avantage du document HTML est qu'il est facile de naviguer entre les différentes pages et ainsi d'accéder aux ressources intéressantes facilement comme à la manière d'une page wikipedia.

Nous avons créé un document synthétique sur les choix techniques et de conception du projet avec différents liens vers les documents spécifiques.

*voir annexe 11 : Manuel de réutilisation*

## **15) Bilan et analyse personnelle sur le projet**

### **Points positifs et apprentissages**

Nous sommes contents du projet dans la mesure où nous avons réussi à finir l'outil dans les délais donnés par l'industriel et à répondre au cahier des charges.

C'était notre première expérience avec la méthodologie agile. Nous avons aimé travailler avec cette méthode car le fait de créer une partie de l'application à chaque sprint permet de garder une certaine motivation. L'utilisation de cette méthodologie a été facilité par le fait que nous ayons bien structuré notre application dès le début. Il a été ensuite très facile d'ajouter des modules complémentaires qui n'étaient pas prévus dans le cahier des charges au départ, comme la recherche d'un film par mot clé par exemple

Nous avons réussi à respecter l'envoi de fiches hebdomadaires ainsi qu'à conserver du temps sur le dernier sprint pour la conception d'un manuel de déploiement et d'utilisation. Nous avons réussi à communiquer régulièrement avec le tuteur entreprise.

Nous avons appris à respecter les normes qui facilitent le travail en équipe : utilisation de diagrammes de Gantt, communication régulière par email et sous forme de réunions ou d'entretiens téléphoniques pour les prises de décisions majeurs, respect des normes d'Orange pour la rédaction du code et conformité du format de rendu de projet. Nous avons également appris à répartir le travail et à estimer l'effort nécessaire à une tâche donnée. Enfin, nous avons découvert les enjeux d'une bonne communication avec le client : satisfaction au rendu de projet, augmentation de la flexibilité et de la rapidité d'évolution du cahier des charges.

*voir annexe 12 : Diagramme de Gantt du travail effectif*

## **Difficultés**

Un point difficile de ce projet transversal a sans doute été de réfléchir à l'architecture du logiciel. En effet, notre manque d'expérience pour cette tâche importante faisait que nous avions parfois l'envie de coder nos idées immédiatement, sans attendre de les définir proprement en amont. Un autre point difficile a été de nous forcer à passer du temps à faire la bibliographie du projet. Nous avons pourtant trouvé ce temps investi très utile par exemple pour le choix de la méthode de recherche des plus proches voisins.

## Conclusion

En conclusion, nous avons réussi à définir notre projet autant de par son contexte que par ses objectifs. Les tests de faisabilité dès le début du projet nous ont permis de limiter les risques liés à la technique. Les phases de réalisation et de bibliographie effectuées en parallèles grâce à la méthode agile, nous ont permis de garder une grande motivation tout au long du projet et de construire chaque module du logiciel sans précipitation. Durant les deux derniers sprints, nous avons pu consacré nos efforts sur la réalisation d'un outil de visualisation des mosaïques et sur la rédaction d'un manuel d'utilisation et de déploiement.

## Annexes

- Annexe 1 : Justification de l'effort de chaque tâche
- Annexe 2 : Diagramme de Gantt
- Annexe 3 : Rapports hebdomadaires semaines 40-41-42
- Annexe 4 : Journal d'actions
- Annexe 5 - Explication différence temps prévu vs effectif
- Annexe 6 - Diagramme de Gantt révisé
- Annexe 7.1 - Diagramme UML Modele révisé
- Annexe 7.2 - Diagramme UML Controleur révisé
- Annexe 8 - Rapports hebdomadaires phase 2
- Annexe 9.1 - Tests unitaires
- Annexe 9.2 - Tests recette
- Annexe 9.3 - Tests de temps de génération
- Annexe 10 - Manuel de déploiement et d'utilisation
- Annexe 11 - Manuel de réutilisation
- Annexe 12 - Diagramme de Gantt du travail effectif
- Annexe 13 - Rapport environnemental du projet

Annexe 1 - Justification de l'effort de chaque tache

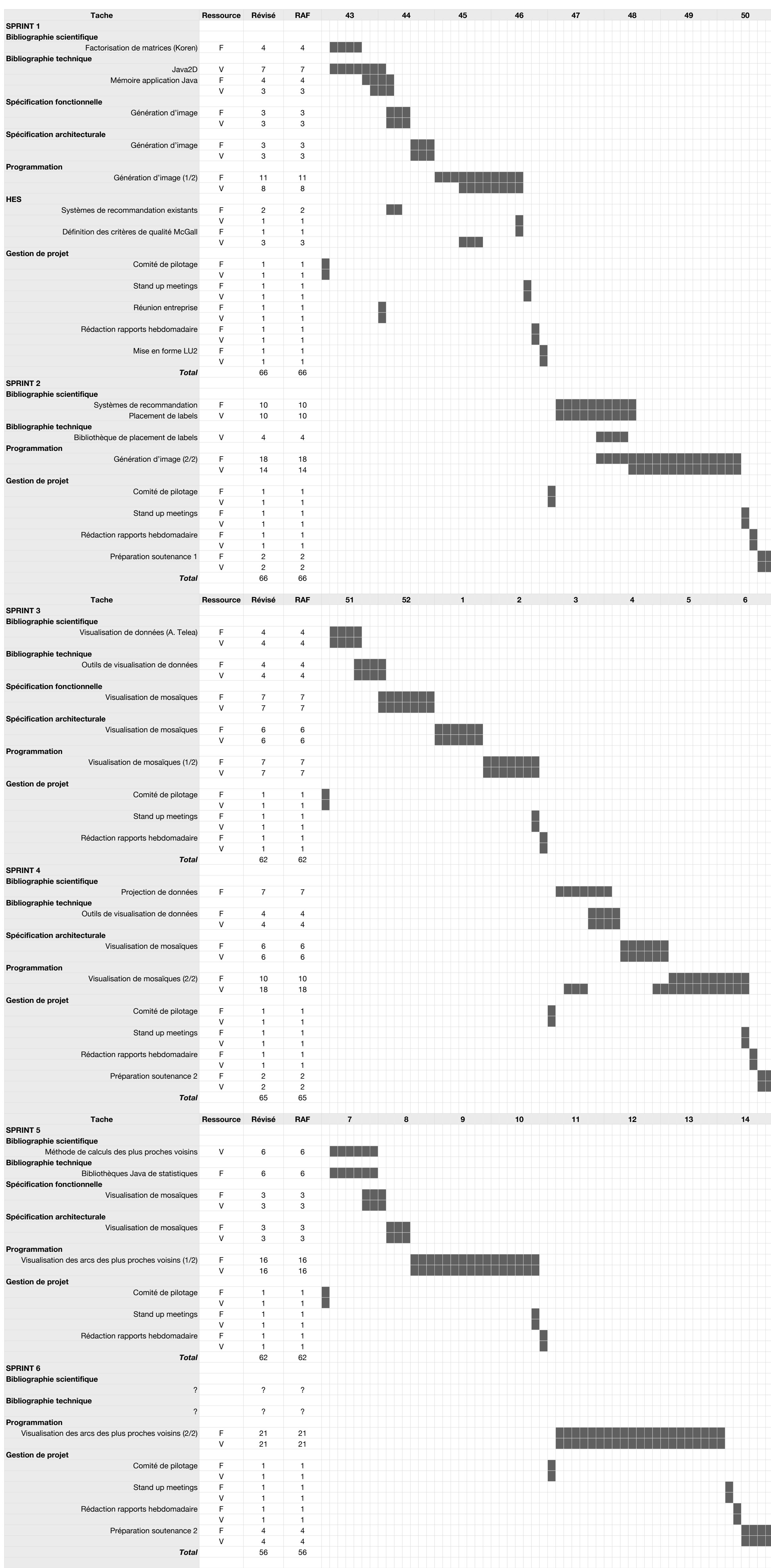
Tache	Estimation du temps pour Fabien	Estimation du temps pour Valentin	Decomposition Lecture/écriture	Atouts	Faiblesses
<b>SPRINT 1</b>					
<b>Bibliographie scientifique</b>					
Factorisation de matrices (Koren)	4	0	1+3	A l'aise en Anglais, document d'introduction à la factorisation de matrices et aux systèmes de recommandation, connaissance du site Netflix dont parle l'article	Aucune connaissance dans le domaine de factorisation de matrices et des systèmes de recommandation
<b>Bibliographie technique</b>					
Java2D	0	7	3+4	Connaissance de Java, expérience avec Java2D dans le cadre du projet	Première bibliographie scientifique
Mémoire application Java	4	3	3+4	Nombreux articles accessibles	Peu de connaissances en matière de mémoire d'application et autre
<b>Spécification fonctionnelle</b>					
Génération d'images	3	3		On possède déjà des informations	C'est la rédaction du document qui nous prendra le plus de temps
<b>Spécification architecturale</b>					
Génération d'image	3	3		Déjà commencé à travailler sur le sujet	C'est la rédaction du document qui nous prendra le plus de temps
<b>Programmation</b>					
Génération d'image (1/2)	11	8		Les paramètres sont simples et fixés, pas d'interaction avec un utilisateur. Premiers résultats obtenus dans la phase de faisabilité	Estimation du temps de codage du module sous le conseil de Marc Gelgon, sera reprécisé après rédaction des spécifications
<b>HES</b>					
Systèmes de recommandation existants	2	1		Connaissance de nombreux sites d'achat, vidéo, musique etc	
Définition des critères de qualité McGall	1	3		Cours détaillé	Jamais utilisé
<b>Gestion de projet</b>					
Comité de pilotage	1	1			
Stand up meetings	1	1			
Réunion entreprise	1	1			
Rédaction rapports hebdomadaire	1	1			
Mise en forme LU2	1	1			
<b>total temps sprint 1</b>	<b>33</b>	<b>33</b>			
<b>SPRINT 2</b>					
<b>Bibliographie scientifique</b>					
Systèmes de recommandation	10	0	4+6	Nombreux articles, sujet intéressant	Il existe de nombreux systèmes de recommandation qui fonctionnent tous différemment et utilisent des techniques complexes
Placement de labels	0	10	4+6		Aucunes connaissances sur le sujet, documents scientifiques longs
<b>Bibliographie technique</b>					
Bibliothèque de placement de labels	0	4	1+3		
<b>Programmation</b>					
Génération d'image (2/2)	18	14		Resultat du premier sprint	Inclure les remarques des tuteurs pour améliorer la release
<b>Gestion de projet</b>					
Comité de pilotage	1	1			
Stand up meetings	1	1			
Rédaction rapports hebdomadaire	1	1			

## Annexe 1 - Justification de l'effort de chaque tache

Préparation soutenance 1	2	2			
<b>total temps sprint 2</b>	<b>33</b>	<b>33</b>			
<b>SPRINT 3</b>					
<b>Bibliographie scientifique</b>					
Visualisation de données (A. Telea)	4	4	1+3		
<b>Bibliographie technique</b>					
Outils de visualisation de données	4	4	1+3		
<b>Spécification fonctionnelle</b>					
Visualisation de mosaïques	7	7		Liberté relativement grande dans le choix de la solution	Pour l'instant rien n'a été défini avec le tuteur entreprise pour la visualisation d'image
<b>Spécification architecturale</b>					
Visualisation de mosaïques	6	6			La visualisation d'image constitue une partie importante du produit final
<b>Programmation</b>					
Visualisation de mosaïques (1/2)	7	7			
<b>Gestion de projet</b>					
Comité de pilotage	1	1			
Stand up meetings	1	1			
Rédaction rapports hebdomadaire	1	1			
<b>total temps sprint 3</b>	<b>31</b>	<b>31</b>			
<b>SPRINT 4</b>					
<b>Bibliographie scientifique</b>					
Projection de données	7	0			
<b>Bibliographie technique</b>					
Outils de visualisation de données	4	4			
<b>Spécification architecturale</b>					
Visualisation de mosaïques	6	6			
<b>Programmation</b>					
Visualisation de mosaïques (2/2)	10	18			
<b>Gestion de projet</b>					
Comité de pilotage	1	1			
Stand up meetings	1	1			
Rédaction rapports hebdomadaire	1	1			
Préparation soutenance 2	2	2			
<b>total temps sprint 4</b>	<b>32</b>	<b>33</b>			
<b>SPRINT 5</b>					
<b>Bibliographie scientifique</b>					
Méthode de calculs des plus proches voisins	0	6			
<b>Bibliographie technique</b>					
Bibliothèques Java de statistiques	6				
<b>Spécification fonctionnelle</b>					
Visualisation des arcs des plus proches voisins	3	3			
<b>Spécification architecturale</b>					
Visualisation des arcs des plus proches voisins	3	3			
<b>Programmation</b>					
Visualisation des arcs des plus proches voisins (1/2)	16	16			
<b>Gestion de projet</b>					
Comité de pilotage	1	1			
Stand up meetings	1	1			

Annexe 1 - Justification de l'effort de chaque tache

Rédaction rapports hebdomadaire	1	1			
total temps sprint 5	31	31			
<b>SPRINT 6</b>					
<b>Bibliographie scientifique</b>	?	?			
<b>Bibliographie technique</b>	?	?			
<b>Programmation</b>					
Visualisation des arcs des plus proches voisins (2/2)	21	21		Le plus gros du code est déjà écrit	Il reste à s'assurer que tout est cohérent et bien commenté
<b>Gestion de projet</b>					
Comité de pilotage	1	1			
Stand up meetings	1	1			
Rédaction rapports hebdomadaire	1	1			
Préparation soutenance finale	4	4			
total temps sprint 6	28	28			



## **Compte rendu de la réunion de présentation du projet : 2 octobre 2013**

Orange Labs travail actuellement sur des algorithmes de classification. Les résultats sont des matrices binaires (Produits \* caractères). Ces matrices creuses sont représentées en index (logs).

Il est difficile de visualiser ces résultats. Orange Labs a donc mis au point des algorithmes pour représenter ces résultats en 2D. A partir de ces représentations en deux coordonnées on peut créer des "cartes".

La première étape du projet est d'étudier la faisabilité de ces "cartes" notamment sur les critères de lisibilité et de rapidité de génération.

Pour l'environnement de développement, Orange Labs nous impose Java. Concernant le format de l'image générée, les tuteurs entreprise et enseignant nous ont conseillés PNG.

### **Travail effectué semaine 40 :**

Notre travail cette semaine s'est partagé entre organisation du projet (Mise en place des outils de travail en commun), bibliographie et premiers tests en Java (Lecture des fichiers log, création d'une image avec Graphics2D et écriture de cette image sur le disque).

Le détail de notre travail est rédigé dans le "journal d'actions".

### **Travail pour la suite :**

Répondre aux questions suivantes :

- Afficher 10000 noms de films sur une seule image, est ce que c'est possible au niveau de la mémoire virtuelle?
- Comment analyser l'évolution du temps d'exécution en fonction du nombre de films?
- Comment faire si deux titres se superposent?
- Quelle police est la plus lisible en taille réduite?
- Quelle taille de police est confortable à la lecture?



Valentin Proust et Fabien Richard - Lundi 7 octobre 2013

## Journal d'actions

Valentin Proust		Fabien Richard	
Date	Tache	temps (h)	documents produits
02/10/2013	Chercher à savoir comment créer et enregistrer une image au format PNG	1	Liens vers la doc web parlant de la classe Image I/O
02/10/2013 ?	Lire le guide de la classe Image I/O		
04/10/2013	Emprunter des bouquins sur java et plus spécialement sur les images	1	emprunt du livre "the art of image processing with Java" de Kenny A. Hunt. Description de la librairie Image I/O et des transformations usuelles.
05/10/2013	Lire le guide Image I/O	0,3	On peut faire une barre de progression de la lecture d'une image
05/10/2013	Lire le tutoriel officiel graphic2d rédiger les questions de ce qui nous embarrasse	1	J'ai réussi à prendre une sous partie d'une image grâce à la méthode "getSubimage"
06/10/2013	prendre connaissance des noms de programmation et du jeu de données		
06/10/2013	Trouver le nombre minimal de pixels pour afficher un film	1	Pas beaucoup d'infos sur ce sujet, il semble que la police de l'exemple soit un bon compromis
	Rechercher quel est le meilleur conteneur pour stocker les films et leur position et l'implémenter		
06/10/2013	Implémenter la fonction "créer carte"	0,5	1 Classe Film et sa classe de test
07/10/2013	Mail hebdo : Faire un résumé de la réunion + Résumé de l'étape1 + résumé des trucs fait cette semaine	0,7	
07/10/2013	Réunion pour mettre en place le github en commun	1	

## **Travail effectué semaine 41**

**Les problématiques proposées par Frank Meyer cette semaine ont été les suivantes :**

- a) il sera intéressant de voir combien de titres de films on peut afficher de manière lisible sur une image de taille relativement standard (1900 x 1200 par exemple, c'est juste un ordre de grandeur).
- b) si la génération d'une sous image est très rapide, on peut imaginer que cela suffise et qu'on ne génère qu'une image à la volée à la fois (en fonction de la fenêtre de clipping) en supposant qu'on a un outil de visualisation à côté et qu'on peut très rapidement demander à la brique de générer, à partir de la dernière image courante, un zoom in/out, un scrolling horizontal ou vertical,....
- c) il serait également intéressant de rapidement regarder quelles sont les limites des outils de visualisation standards très communs sur PC comme les gestionnaires d'images par défaut et les navigateurs internet.

**Nous pensons pouvoir répondre à quelques questions :**

- Tout d'abord il nous semble plus logique de travailler sur des images carres. En effet le jeu de données fournies prend ses valeurs dans un carré.
- Nous avons effectué des tests avec une police de 20 car cette police reste lisible jusqu'à un niveau de zoom suffisamment important.
- Nous avons testé la taille maximale des images que l'on peut générer avec Java sur nos ordinateurs personnels :

Sur un mac de 2010 avec 2Go de RAM, le maximum est de 5000\*5000px avant "heap space" en 12s environ.

Sur PC de 2013 avec 4Go de RAM, le maximum est 10000\*10000px. Le temps d'exécution est de 12s.

- Il semble que le nombre maximal de noms de films affichables sur une image de 3000\*3000px soient d'environ 500 avec un temps d'exécution d'environ 3s. Ces chiffres sont à nuancer car dans le jeu de données fourni, les mots n'ont que peu de lettres et leur caractère aléatoire limite les superpositions. De plus le temps d'exécution ne prend pas en compte le tri des films pour savoir lesquels appartiennent à la sous image. L'image en pièce jointe donne un ordre d'idée.

**Travail en cours de réalisation :**

- On se pose la question de savoir si les temps de génération des images s'additionnent. Par exemple : est ce que ça prend autant de temps de générer 5 grosses images ou 10 petites images?
- Idem pour le temps de visualisation des images
- On travaille sur l'écriture de la fonction de décision de l'appartenance d'un film à une sous image à partir de ses coordonnées dans le fichier de log.

## **Travail effectué semaine 42**

L'effort cette semaine a été porté sur la dimension organisationnelle du projet. En effet l'équipe pédagogique de l'école nous demande de fournir un planning prévisionnel. L'exercice est difficile car le sujet sur lequel nous travaillons est un sujet recherche.

Nous avons décidé de travailler en sprints de 4 semaines. Ces sprints comprendront un travail de bibliographie technique et scientifique, une réflexion sur les problématiques HES (homme entreprise et société, service transversal de l'école) et une partie solution logicielle (testée de façon unitaire). Nous continuerons d'envoyer un rapport hebdomadaire toutes les semaines. A chaque fin de sprint nous enverrons le travail réalisé au tuteur entreprise.

Le travail de bibliographie demandé est de développer 3 concepts scientifiques en une dizaine de pages et 3 sujets technologiques un peu plus brièvement.

Sur le conseils des tuteurs entreprise et école, nous avons choisi de nous concentrer sur les concepts scientifiques suivants :

- Factorisation de matrice
- Systèmes de recommandations
- Visualisation de données

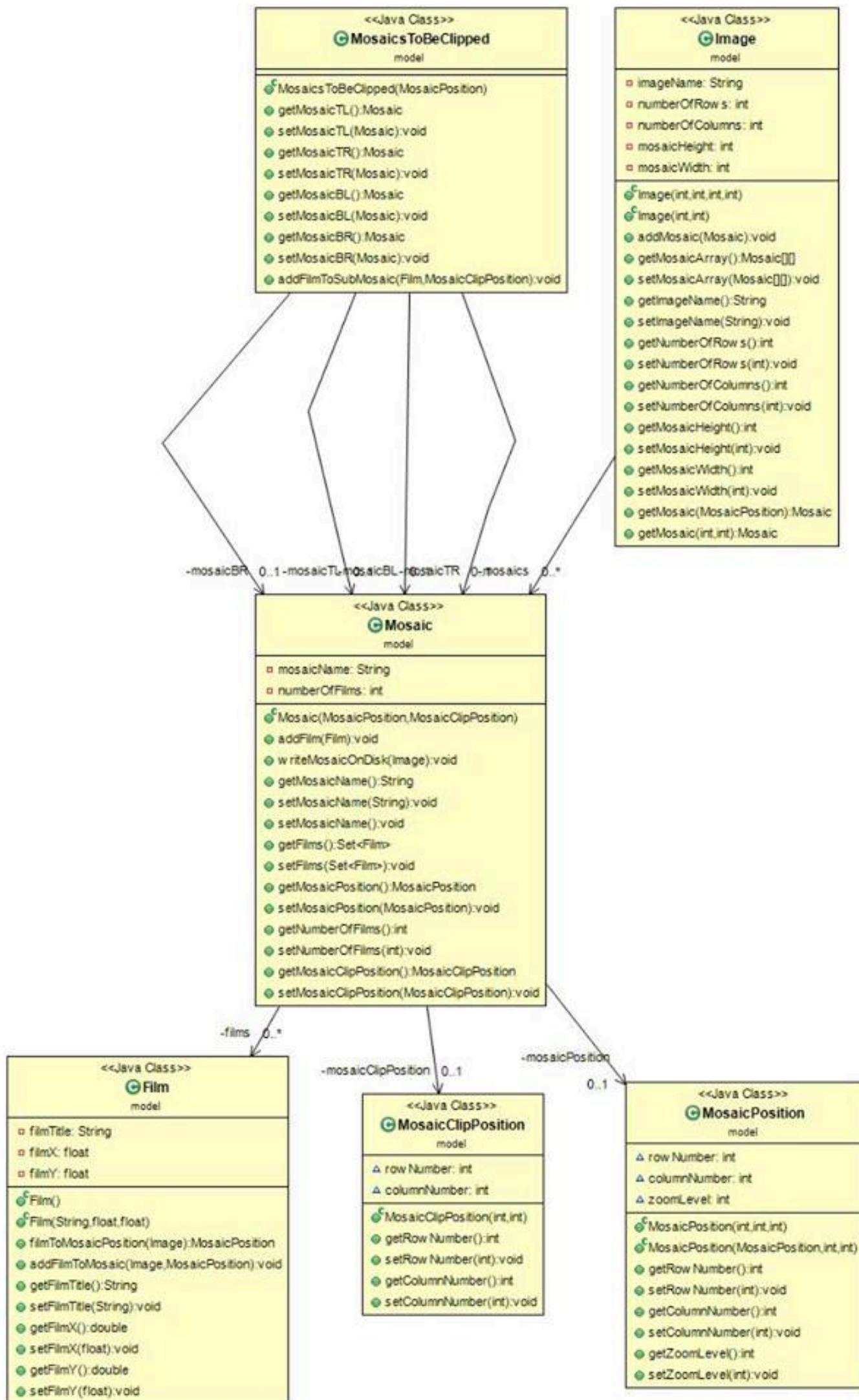
#### Annexe 4 - Journal d'actions

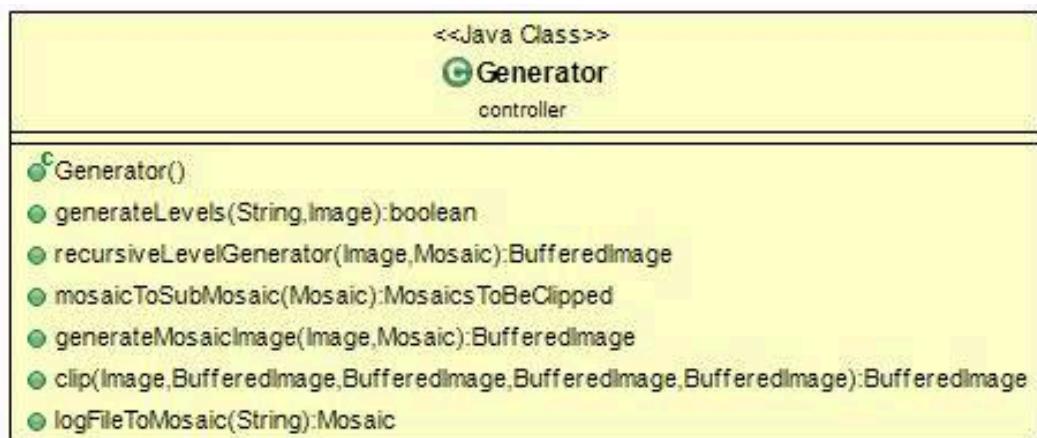
Valentin Proust				Fabien Richard			
Date	Tache	temps (h)	documents produits	Date	Tache	temps (h)	documents produits
02/10/2013	Chercher à savoir comment créer et enregistrer une image au format PNG	1	Liens vers la doc web parlant de la classe Image I/O	06/10/2013	Comprendre l'utilisation de Git	1,5	Document commandes utiles + http://rogerdudler.github.io/git-guide/ + http://try.github.io/levels/1/challenges/1
02/10/2013 ?	Lire le guide de la classe Image I/O			06/10/2013	Mise en place sur GitHub	0,5	https://github.com/richfab/Orange.git
04/10/2013	Emprunter des bouquins sur java et plus spécialement sur les images	1	emprunt du livre "the art of image processing with java" de Kenny A. Hunt. Description de la librairie image I/O et des transformations usuelles.	06/10/2013	Création d'un projet java et commit	0,5	
05/10/2013	Lire le guide Image I/O	0,3	On peut faire une barre de progression de la lecture d'une image J'ai réussi à prendre une sous partie d'une image grâce à la méthode "getSubimage"	06/10/2013	Prise de connaissance des règles de programmation d'Orange	0,2	
05/10/2013	Lire le tutoriel officiel graphic2d réddiger les questions de ce qui nous émbarasse	1		06/10/2013	Creation et partage d'un calendrier	0,2	https://www.google.com/calendar/feeds/n7m0v68su0e6me...
06/10/2013	prendre connaissance des normes de programmation et du jeu de données			06/10/2013	Lecture d'un fichier log	0,5	On arrive à parser un fichier de log de 10000 film et à les afficher
06/10/2013	Trouver le nombre minimal de pixels pour afficher un film	1	Pas beaucoup d'infos sur ce sujet, il semble que la police de l'exemple soit un bon compromis	10/06/2013	Codage		On instancie des objets films et on les place dans un Set
06/10/2013	Rechercher quel est le meilleur conteneur pour stocker les films et leur position et l'implémenter	1	Classe Film et sa classe de test	12/10/2013	Travail sur les tests de faisabilité Réunion de débriefing de la réponse du tuteur entreprise et reflexion sur l'organisation générale du projet	0,5	document excel
07/10/2013	Implémenter la fonction "créer carte"	0,5		16/10/2013		1	
07/10/2013	Mail hebdo : Faire un résumé de la réunion + Résumé de l'étape1 + résumé des trucs fait cette semaine	0,7		17/10/2013	entretien téléphonique avec Frank Meyer et Mar Gelgon	1	Proposition de planning et de bibliographie
07/10/2013	Réunion pour mettre en place le github en commun	1		18/10/2013	comité de pilotage	2	Rédaction d'un planning et de questions à l'entreprise
09/10/2013	Recherches sur le "placement de label"	1	bibliographie sur le placement de labels	18/10/2013	Lecture d'un article sur la factorisation de matrice appliquée aux systèmes de recommandation (Koren)	1,5	
10/10/2013	Dessin des cartes à partir du fichier de positions aléatoires	1	Images Png	21/10/2013	Rédaction du rapport hebdo	0,5	
11/10/2013	Rangement des fichiers de dessin de carte	1		22/10/2013	LU1	1	
11/10/2013	Organisation du projet	0,5		23/10/2013	Planning prévisionnel	2	
11/10/2013	Propositions des tests de calcul des sous image	1		24/10/2013	Mise en forme du LU1	1	
14/10/2013	créer des images carrées	1		24/10/2013	Comité de pilotage	1	
17/10/2013	entretien téléphonique avec Frank Meyer et Mar Gelgon	1	Proposition de planning et de bibliographie	25/10/2013	Finitions du LU1	1	
18/10/2013	comité de pilotage	2	Rédaction d'un planning et de questions à l'entreprise	27/10/2013	Gantt	2	Diagramme de Gantt
21/10/2013	Rédaction du rapport hebdo	0,5		27/10/2013	Relecture du LU1 et dernières retouches	1	LU1
22/10/2013	LU1	1					
23/10/2013	Planning prévisionnel	2					
27/10/2013	Relecture du LU1 et dernières retouches	1	LU1				



Visualisation de mosaïques	7	3	7		Nous avons commencé cette tâche mais la génération de mosaïque n'était pas assez aboutie pour la réaliser entièrement. Nous 3 l'avons donc déplacé en sprint 6	Liberté relativement grande dans le choix de la solution	Pour l'instant rien n'a été défini avec le tuteur entreprise pour la visualisation d'image
<b>Spécification architecturale</b>							
Visualisation de mosaïques	6	0	6		Déplacé en sprint 6 pour les mêmes 0 raisons que ci dessus		La visualisation d'image constitue une partie importante du produit final
<b>Programmation</b>							
Visualisation de mosaïques (1/2)	7	7	7		Déplacé en sprint 6 pour les mêmes 0 raisons que ci dessus		
<b>Gestion de projet</b>							
Comité de pilotage	1		1				
Stand up meetings	1		1				
Rédaction rapports hebdomadaire	1		1		Il est difficile d'estimer le temps passé à la gestion de projet		
<b>total temps sprint 3</b>	31		31				
<b>SPRINT 4</b>							
<b>Bibliographie scientifique</b>							
Projection de données	7	0	0		Nous avons décidé de déplacer cette tâche au sprint 6 suite à un changement 0 des priorités		
<b>Bibliographie technique</b>							
Outils de visualisation de données	4	5	4		L'effort pour cette tâche a été légèrement 5 sous-estimé		
<b>Spécification architecturale</b>							
Visualisation de mosaïques	6	3	6		Nous avons échangé avec notre tuteur entreprise à ce sujet mais cela nous a pris 3 moins de temps que prévu		
<b>Programmation</b>							
Visualisation de mosaïques (2/2)	10	18	18		Le temps alloué à Fabien pour la bibliographie projection de donnée a été consacré à la programmation pour la visualisation de mosaïques 18		
<b>Gestion de projet</b>							
Comité de pilotage	1		1				
Stand up meetings	1		1				
Rédaction rapports hebdomadaire	1		1		Il est difficile d'estimer le temps passé à la gestion de projet car c'est difficile de comptabiliser les heures passées entre deux cours à discuter, envoyer des emails, rencontrer nos enseignants		
Préparation soutenance 2	2		2				
<b>total temps sprint 4</b>	32		33				
<b>SPRINT 5</b>							
<b>Bibliographie scientifique</b>							
Méthode de calculs des plus proches voisins	6	6	6		Nous avions correctement estimé le temps 6 nécessaire à cette tâche		
<b>Bibliographie technique</b>							
Bibliothèques Java de statistiques	6	0			Nous n'avons pas eu besoin de faire cette bibliographie technique suite à un 0 changement dans les spécificités		
<b>Spécification fonctionnelle</b>							
Visualisation des arcs des plus proches voisins	3	3	3		Nous avons échangé à de nombreuses reprises avec notre tuteur entreprise sur la 3 visualisation des plus proches voisins		
<b>Spécification architecturale</b>							
Visualisation des arcs des plus proches voisins	3	2	3		Notre tuteur entreprise nous a rapidement aiguillés sur une piste pour la recherche 2 optimisée des plus proches voisins		
<b>Programmation</b>							
Visualisation des arcs des plus proches voisins (1/2)	16	15	16		Notre modèle étant bien conçu il a été facile d'écrire et implémenter l'algorithme 15 de recherche des plus proches voisins		
<b>Gestion de projet</b>							
Comité de pilotage	1		1				
Stand up meetings	1		1				
Rédaction rapports hebdomadaire	1		1		Il est difficile d'estimer le temps passé à la gestion de projet		
<b>total temps sprint 5</b>	37		31				







## Travail effectué semaine 49

Nous avons en semaine 49 pris du temps pour planifier nos sprints à venir. Nous avons décidé de l'organisation suivante :

### Sprint 3 :

Biblio scient : visualisation de données (Alex Téléa)
Biblio Tech : outils de visualisation de données
Programmation : outils de visualisation de mosaïques (1/2)

### Sprint 4 :

Biblio scient : projection de données
Biblio Tech : outils de visualisation de données
Programmation : outils de visualisation de mosaïques (2/2)

### Sprint 5 :

Biblio scient : méthodes de calculs de voisins les plus proches
Biblio Tech : bibliothèques java de statistique (R sous Java)
Programmation : rajout de la fonctionnalité de visualisation des arcs des voisins les plus proches (1/2)

### Sprint 6 :

Biblio scient : projection de données
Biblio Tech : Outils de visualisation
Programmation : Création de marges sur chaque mosaïque

### Sprint 7 :

Préparation du livrable final
Programmation : Amélioration de l'outil de visualisation

## Travail pour la suite

Nous allons discuter avec notre tuteur entreprise des choix à faire concernant l'outil de visualisation des mosaïques.

	Outil de génération				
Test n°	Classe testée	Fonction testée	paramètres	résultat attendu	résultat obtenu
1	Film	addFilmToMosaic(Image image, MosaicPosition mosaicPosition)	Nouvelle image, position(0.5,0.5)	"Film ajouté à la position X,Y"	OK
2	Film	addFilmToMosaic(Image image, MosaicPosition mosaicPosition)	Nouvelle image, positions (-0.5,3)	Exception : "paramètres incorrects"	OK
3	Film	filmToMosaicPosition(Image image)	Nouvelle image, film avec position (0.5,0.5)	"Film sur la mosaïque X,Y"	OK
4	Film	filmToMosaicPosition(Image image)	Nouvelle image, film avec position (-0.5,3)	Exception : "paramètres incorrects"	OK
5	Film	filmToMosaicPosition(Image image)	Nouvelle image, film avec position (2,2)	"Film ajouté à la mosaïque 0,0"	OK
6	Film	filmToMosaicPosition(Image image)	Nouvelle image, film avec position (0,0)	"Film ajouté à la position 1,1"	OK
7	Image	addMosaic(Mosaic mosaic)	Nouvelle mosaic avec un seul film valide	"mosaic ajouté à la position X,Y"	OK
8	Mosaic	writeMosaicOnDisk(Image image)	Nouvelle image préparée	Mosaïque écrite sur le disque avec les bons noms de films	OK
9	ClosestNeighbo	Film closestNeighbo(Set<Film> films,Film film)	Set de films :[(2,2),(3,3)], Film (1,1)	"Le film le plus proche est à la position (2,2)"	OK
10	ClosestNeighbo	Film closestNeighbo(Set<Film> films,Film film)	Set de films :[(2,2),(3,3)], Film (4,4)	"Le film le plus proche est à la position (3,3)"	OK
11	Draw	drawArrow(Graphics2D g, double x1, double y1, double x2, double y2)	x1=0,y1=2,x2=3;y2=3	La flèche affichée va de la gauche vers la droite	OK
12	Draw	drawArrow(Graphics2D g, double x1, double y1, double x2, double y2)	x1=2,y1=2,x2=1;y2=1	La flèche affichée va de la droite vers la gauche	OK
13	Draw	drawArrow(Graphics2D g, double x1, double y1, double x2, double y2)	x1=1,y1=2,x2=1;y2=1	Aucune flèche n'apparaît	OK
14	Generator	deleteDir(File dir)	Un fichier qui existe sur le disque	"Fichier supprimé"	OK
15	Generator	deleteDir(File dir)	UN fichier qui n'existe sur le disque	Exception : "Le fichier n'existe pas"	OK
16	Generator	deleteDir(File dir)	UN fichier dont les droits d'accès sont protégé	Exception : "Le fichier est protégé"	OK
17	Generator	BufferedImage clip(Image image,BufferedImage biMosaicTL, BufferedImage biMosaicTR, BufferedImage biMosaicBL,	Utiliser 4 images	Le fichier clippé est enregistré sur le disque et est correct	OK
18	LogFile	logFileToMosaic(String filePath)	Utiliser un jeu de 4 films	vérifier que tous les noms de films sont dans le set	OK
19	Main	main(args)	arguments -o C:\Users\Valentin\ptrans -i C:\Users\Valentin\ptrans\input -k film47	arguments valides	OK
20	Main	main(args)	arguments -o C:\Users\Valentin\ptrans -i -k film48	arguments invalides	OK
<b>Outil de visualisation</b>					
21	viz	function zoom_in(element)	Clique sur une imagette	Le zoom fonctionne correctement	OK
22	viz	function on_image_loaded()	Chargement de la page web	La page se charge avec une image	OK

## Travail effectué semaine 50

Durant cette semaine 50 nous avons testé et adapté différents outils de visualisation de mosaïques. Nous avons notamment testé et adapté :

- Leaflet : <http://fabienrichard.fr/projects/imagelog/>
- Mapbox
- maps.stamen

Le problème de toutes ces API est qu'elles ne peuvent pas être utilisées comme telles avec nos mosaïques (ou tuiles). En effet, notre programme de génération de mosaïque ne génère pas les tuiles de la même façon que les outils de cartographie : si un film est seul sur sa mosaïque au niveau 2 par exemple, on estime qu'il est inutile de générer la mosaïque pour ce film en niveau 3. Alors que certaines zones plus denses de l'image (nombreux films très proches les uns des autres) nécessiteront peut-être de descendre jusqu'à un niveau 4 ou 5 pour que les titres deviennent lisibles. On se retrouve ainsi avec des niveaux de zoom maximum différents pour chaque partie de l'image alors que pour un système de cartographie, toutes les mosaïques sont générées pour chaque niveau de zoom.

Nous avons donc commencé à implémenter notre propre outil de visualisation :

<http://fabienrichard.fr/projects/imagelog/viz/>

## Travail pour la suite

Nous allons discuter avec notre tuteur entrepris des choix à faire concernant l'outil de visualisation des mosaïques.

## Travail effectué semaine 1

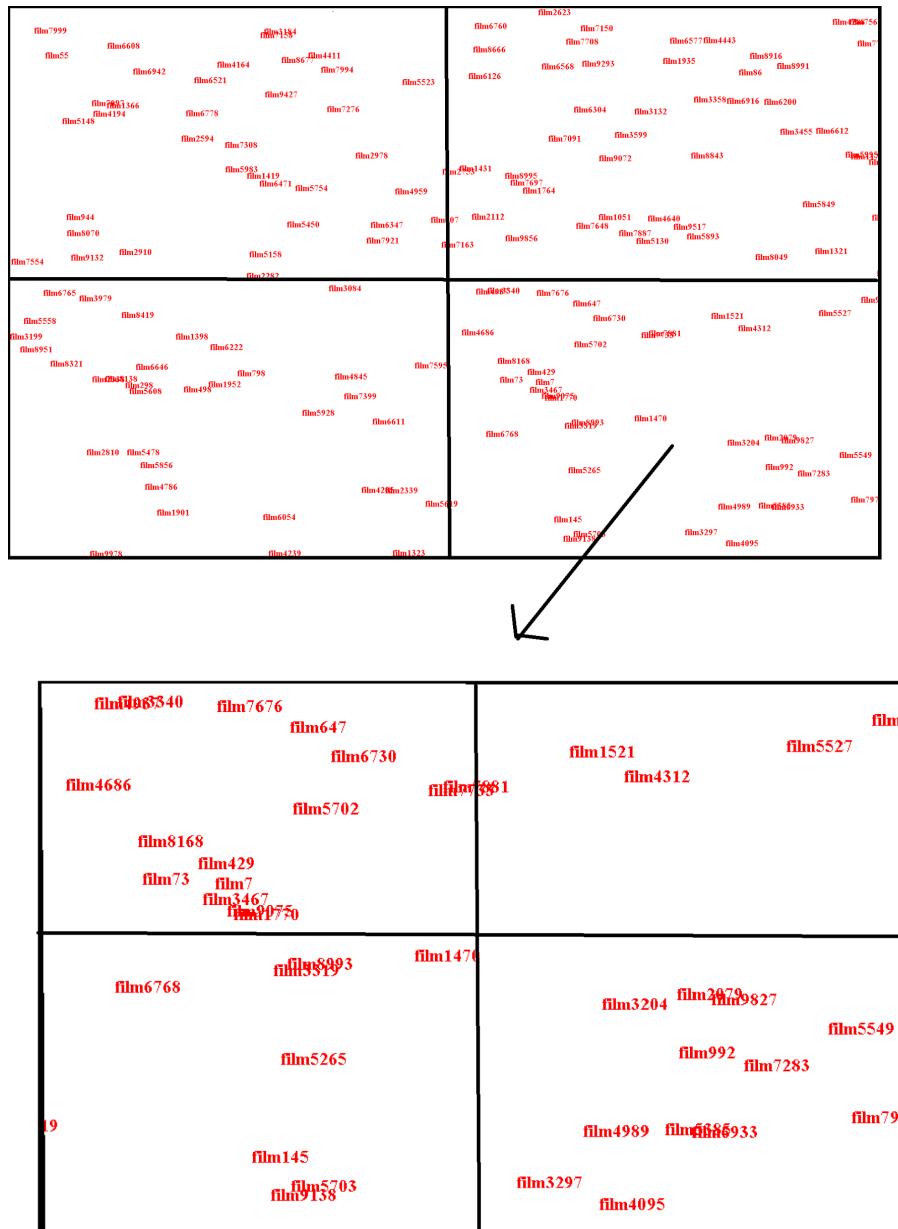
Nous avons commencé pendant les vacances à rédiger une bibliographie sur les méthodes de recherche des plus proches voisins d'un point.

Concernant la partie logicielle, nous avons émis des hypothèses sur les différents niveaux de zoom que devait comporter la carte. De plus nous avons commencé à réaliser la méthode de génération des mosaïques correspondant à ces niveaux de zoom.

## Travail pour la suite

Nous avons trois objectifs pour le reste du sprint :

- Dessiner les liens des plus proches voisins
- Créer des marges de recouvrement sur les mosaïques pour éviter d'avoir des noms de film à cheval entre deux mosaïques
- Décider d'une méthode efficace de génération des différents niveaux de zoom (en effet ayant déjà la projection des points sur le zoom  $m$  maximum, ne pourrait-on pas "simplement" assembler les mosaïques 4 par 4 pour générer la mosaïque de zoom  $m-1$  ?)



*Fig. 1 Illustration grossière de la navigation entre niveaux de zoom*

La fenêtre de notre outil de visualisation comprendra 4 zones cliquables qui permettront d'accéder au niveau de zoom supérieur. Le dernier niveau de zoom laissera apparaître les liens entre les plus proches voisins.

Par la suite, nous pourrons imaginer pouvoir naviguer latéralement dans l'image, d'augmenter ou diminuer le niveau de zoom à l'aide de la molette de défilement de la souris, etc.

## Travail effectué semaine 2

Cette semaine nous avons décidé d'une méthode efficace de génération des différents niveaux de zoom et nous l'avons implémentée. La solution la plus rapide et la plus naturelle est en utilisant la récursivité. Le principe est le suivant : on commence par générer les mosaïques de niveau de zoom le plus élevé (en détectant le nombre de films maximal qui doivent composer la mosaïque). Dès que 4 mosaïques du niveau de zoom  $m$  sont générées, on les assemble pour générer la mosaïque de niveau  $m-1$ .

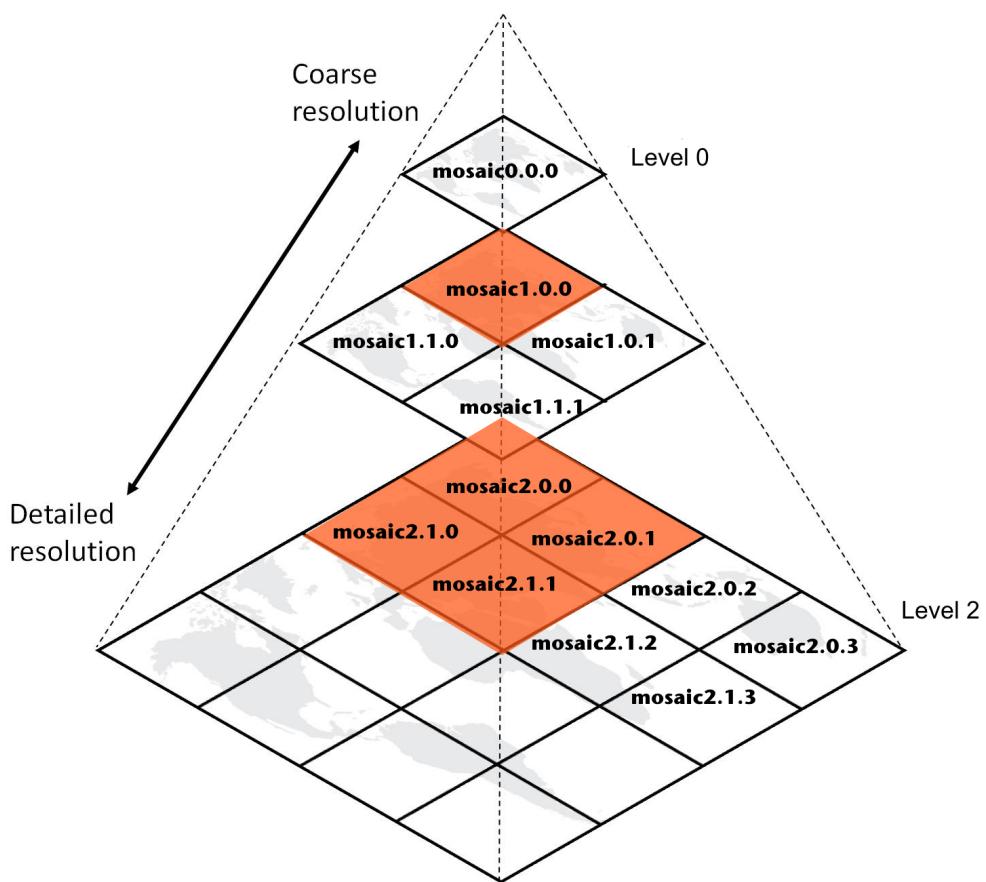


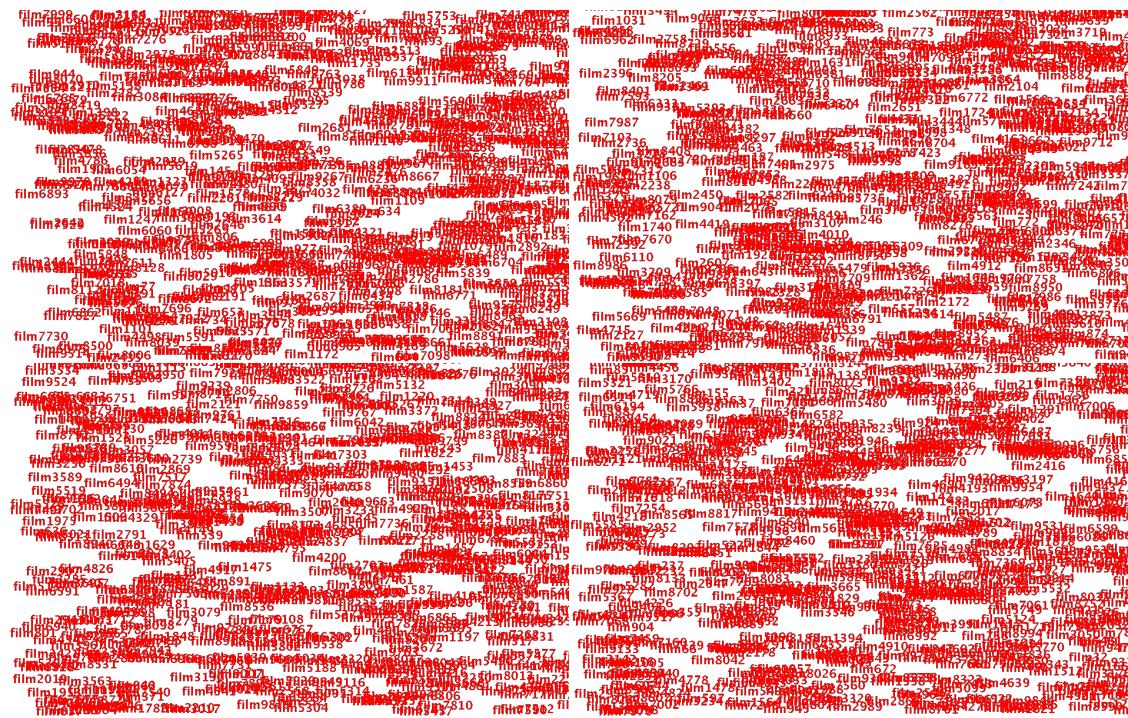
Fig. 1 Illustration de la génération des niveaux de mosaïques

Le nom d'une mosaïque est défini de la manière suivante :  
**mosaicZoomLevel.Rang.Colonne**

Déroulement de l'algorithme récursif : on génère mosaic2.0.0, mosaic2.0.1, mosaic2.1.0 et mosaic2.1.1 (en rouge). Puis elles sont assemblées pour former

mosaic1.0.0 (en rouge). C'est ensuite au tour de mosaic2.0.2, mosaic2.0.3, mosaic2.1.2 et mosaic2.1.3 d'être générées puis assemblées pour former mosaic1.0.1.

Un avantage important de cette méthode est la rapidité. Le second avantage est que seuls les mosaïques qui contiennent des films sont générées. On évite ainsi de générer des mosaïques vides.



*Fig 2. Exemple de mosaïque assemblée à partir de ses filles (ici mosaic1.0.0)*

## Travail pour la suite

Nous avons deux objectifs pour le reste du sprint :

- Dessiner les liens des plus proches voisins
- Créer des marges de recouvrement sur les mosaïques pour éviter d'avoir des noms de film à cheval entre deux mosaïques

## Travail effectué semaine 3

Cette semaine nous avons réussi à générer les mosaïques selon plusieurs niveaux de zooms, comme expliqué dans le dernier rapport.

Nous avons aussi adapté et testé notre algorithme de génération avec la base de donnée MovieLens2factor. Il reste très rapide : 35" pour la génération de toutes les mosaïques nécessaires aux 3700 films de MovieLens.

## Travail pour la suite

Nous avons deux objectifs pour le reste du sprint :

- Dessiner les liens des plus proches voisins
- Décider de la position du titre du film (en haut à gauche ou en bas à droite) pour éviter que les titres ne soient coupés

Nous allons ensuite travailler sur un système de navigation dans les mosaïques, soit avec :

- Un squelette html
- Outil de visualisation de mosaïques (utilisé notamment en cartographie)

## Travail effectué semaine 4

Cette semaine nous avons amélioré l'outil de visualisation de mosaïques :  
<http://fabienrichard.fr/projects/imagelog/viz/>

## Travail pour la suite

Nous avons un objectif pour le reste du sprint :

- Dessiner les liens des plus proches voisins

## Travail effectué semaine 5

Nous avons cette semaine travaillé sur la génération de flèches indiquant pour chaque film le plus proche voisin. L'algorithme de recherche du plus proche voisin est un simple parcours itératif de la liste des films contenus dans la mosaïque. En effet, chaque mosaïque ne contient que quelques films, il n'était donc pas nécessaire d'implémenter un algorithme efficace comme vu dans la bibliographie. La limite de cette recherche des PPV par mosaïque est que le PPV d'un film peut se trouver sur la mosaïque voisine.

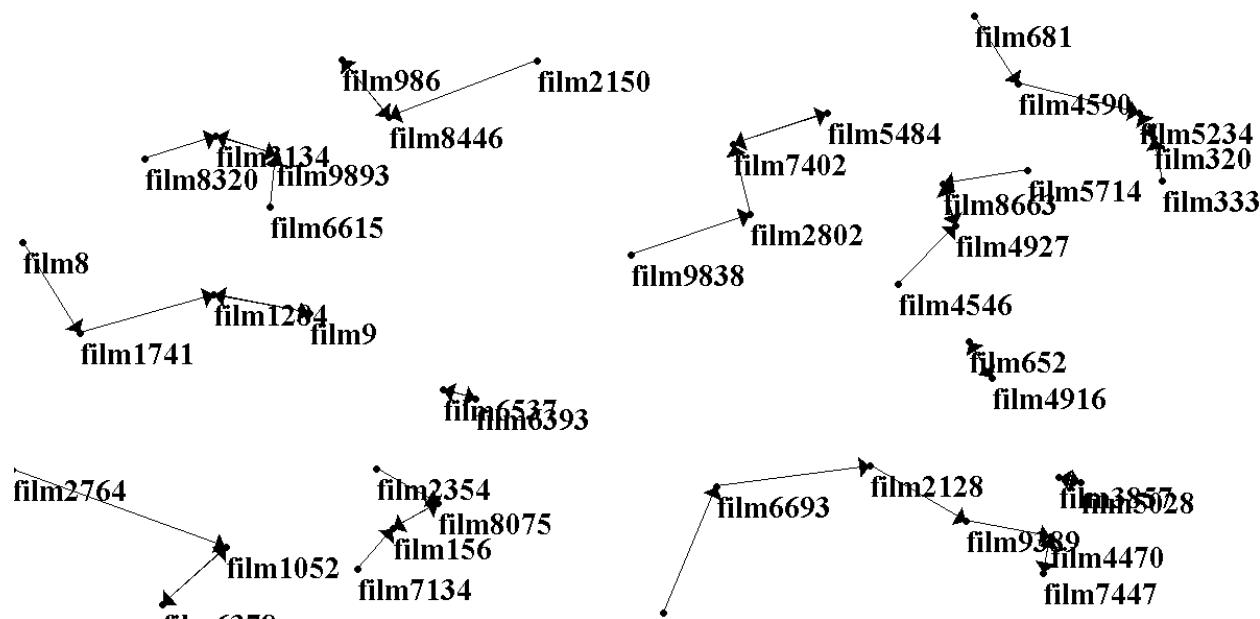


Fig. 1 Exemple de mosaïque générée avec les PPV

## Travail pour la suite

Notre objectif pour la suite est d'améliorer l'interface de navigation. Nous allons garder le squelette HTML.

Voies d'amélioration (à définir avec Franck Meyer)

- Créer des marges sur chaque mosaïque pour :
  - Éviter d'avoir des noms de films coupés et donc illisibles
  - Découvrir des plus proches voisins hors de la mosaïque
  
- Pour ne pas pouvoir zoomer au delà du dernier niveau, nous avons deux pistes :
  - Accompagner le squelette HTML de la structure des mosaïques (ex : json)
  - Nommer différemment les fichiers de feuilles ou de noeuds

L'avantage de la première solution est de permettre par exemple une recherche d'un film par son titre, une navigation horizontale/verticale.

La seconde solution est plus simple à implémenter mais moins évolutive.

## Travail effectué semaine 6

Nous avons cette semaine, choisi avec notre tuteur, la méthode pour la visualisation des mosaïques. Nous nommerons une feuille `leaf.[zoomLevel].[Rang].[Colonne]` et un parent `clip.[zoomLevel].[Rang].[Colonne]`. Ainsi, si l'image affichée commence par "leaf" on sait qu'on ne doit pas proposer à l'utilisateur de zoomer au delà (masquage du bouton `zoom+`). De même, si l'image affichée commence par "clip" on sait que l'utilisateur doit pouvoir zoomer dans l'image (affichage du bouton `zoom+`). Pour masquer le bouton `zoom-` il suffit de tester le `[zoomLevel]` du nom de fichier de l'image, s'il est égal à 0 alors c'est l'image de niveau maximum et on masque le bouton `zoom-`.

## Travail pour la suite

Notre objectif pour la suite est d'améliorer l'interface de navigation. Notre tuteur entreprise nous a conseillé de concentrer nos efforts sur les éléments de base (la navigation par exemple).

Nous avons pour objectif pour la suite de :

- Créer des marges sur chaque mosaïque pour :
  - Éviter d'avoir des noms de films coupés et donc illisibles
  - Découvrir des plus proches voisins hors de la mosaïque

## Travail effectué semaine 7

Nous avons cette semaine, nous avons commencé l'implémentation de l'interface web de navigation. C'est une simple page web qui permet en cliquant sur une zone de l'image d'accéder au niveau de zoom supérieur.

## Travail pour la suite

Il nous faut encore gérer le masquage des boutons *zoom+* et *zoom-* lorsqu'ils ne doivent pas pouvoir être cliqués (niveaux de zoom maximum).

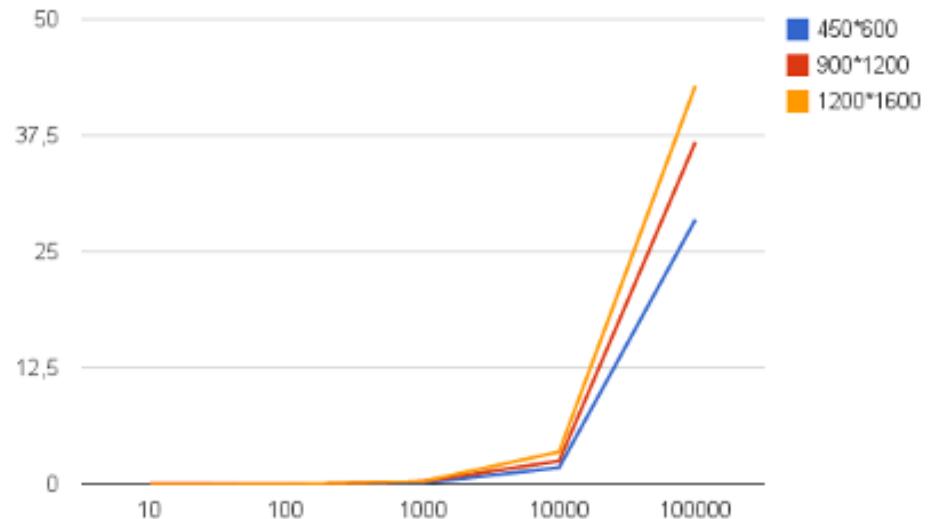
Nous avons pour objectif pour la suite de :

- Créer des marges sur chaque mosaïque pour :
  - Éviter d'avoir des noms de films coupés et donc illisibles
  - Découvrir des plus proches voisins hors de la mosaïque

	Outil de génération				
Test n°	Classe testée	Fonction testée	paramètres	résultat attendu	résultat obtenu
1	Film	addFilmToMosaic(Image image, MosaicPosition mosaicPosition)	Nouvelle image, position(0.5,0.5)	"Film ajouté à la position X,Y"	OK
2	Film	addFilmToMosaic(Image image, MosaicPosition mosaicPosition)	Nouvelle image, positions (-0.5,3)	Exception : "paramètres incorrects"	OK
3	Film	filmToMosaicPosition(Image image)	Nouvelle image, film avec position (0.5,0.5)	"Film sur la mosaïque X,Y"	OK
4	Film	filmToMosaicPosition(Image image)	Nouvelle image, film avec position (-0.5,3)	Exception : "paramètres incorrects"	OK
5	Film	filmToMosaicPosition(Image image)	Nouvelle image, film avec position (2,2)	"Film ajouté à la mosaïque 0,0"	OK
6	Film	filmToMosaicPosition(Image image)	Nouvelle image, film avec position (0,0)	"Film ajouté à la position 1,1"	OK
7	Image	addMosaic(Mosaic mosaic)	Nouvelle mosaic avec un seul film valide	"mosaic ajouté à la position X,Y"	OK
8	Mosaic	writeMosaicOnDisk(Image image)	Nouvelle image préparée	Mosaïque écrite sur le disque avec les bons noms de films	OK
9	ClosestNeighbo	Film closestNeighbo(Set<Film> films,Film film)	Set de films :[(2,2),(3,3)], Film (1,1)	"Le film le plus proche est à la position (2,2)"	OK
10	ClosestNeighbo	Film closestNeighbo(Set<Film> films,Film film)	Set de films :[(2,2),(3,3)], Film (4,4)	"Le film le plus proche est à la position (3,3)"	OK
11	Draw	drawArrow(Graphics2D g, double x1, double y1, double x2, double y2)	x1=0,y1=2,x2=3;y2=3	La flèche affichée va de la gauche vers la droite	OK
12	Draw	drawArrow(Graphics2D g, double x1, double y1, double x2, double y2)	x1=2,y1=2,x2=1;y2=1	La flèche affichée va de la droite vers la gauche	OK
13	Draw	drawArrow(Graphics2D g, double x1, double y1, double x2, double y2)	x1=1,y1=2,x2=1;y2=1	Aucune flèche n'apparaît	OK
14	Generator	deleteDir(File dir)	Un fichier qui existe sur le disque	"Fichier supprimé"	OK
15	Generator	deleteDir(File dir)	UN fichier qui n'existe sur le disque	Exception : "Le fichier n'existe pas"	OK
16	Generator	deleteDir(File dir)	UN fichier dont les droits d'accès sont protégé	Exception : "Le fichier est protégé"	OK
17	Generator	BufferedImage clip(Image image,BufferedImage biMosaicTL, BufferedImage biMosaicTR, BufferedImage biMosaicBL,	Utiliser 4 images	Le fichier clippé est enregistré sur le disque et est correct	OK
18	LogFile	logFileToMosaic(String filePath)	Utiliser un jeu de 4 films	vérifier que tous les noms de films sont dans le set	OK
19	Main	main(args)	arguments -o C:\Users\Valentin\ptrans -i C:\Users\Valentin\ptrans\input -k film47	arguments valides	OK
20	Main	main(args)	arguments -o C:\Users\Valentin\ptrans -i -k film48	arguments invalides	OK
<b>Outil de visualisation</b>					
21	viz	function zoom_in(element)	Clique sur une imagette	Le zoom fonctionne correctement	OK
22	viz	function on_image_loaded()	Chargement de la page web	La page se charge avec une image	OK

	<b>Cas d'utilisation</b>	<b>Scénario de test</b>	<b>Résultat attendu</b>	<b>Résultat obtenu</b>
GENERATION	On souhaite générer une carte avec les paramètres par défaut	1.Se placer dans le dossier qui contient le jar executable (imagelog.jar) 2.Ouvrir un terminal et saisir la ligne de commande suivante : \$ java -jar imagelog.jar -o ./output/mosaic/ -l ./input/fichierdelog.txt	un dossier output est créé et les mosaïques qui composent la carte sont générées (taille 900*1200px)	
	On souhaite générer une carte avec des dimensions personnalisées	1.Se placer dans le dossier qui contient le jar executable (imagelog.jar) 2.Ouvrir un terminal et saisir la ligne de commande suivante : \$ java -jar imagelog.jar -o ./output/mosaic/ -l ./input/fichierdelog.txt -w 1200 -h 600	un dossier output est créé et les mosaïques qui composent la carte sont générées (taille 600*1200px)	
	On souhaite générer une carte avec les paramètres par défaut et rechercher un mot clé qui existe dans le log	1.Se placer dans le dossier qui contient le jar executable (imagelog.jar) 2.Ouvrir un terminal et saisir la ligne de commande suivante : \$ java -jar imagelog.jar -o ./output/mosaic/ -l ./input/fichierdelog.txt -k film85	un dossier output est créé et les mosaïques qui composent la carte sont générées (taille 900*1200px). Celles qui comportent le mot clé film85 sont coloriées en rouge et le titre du film est écrit en blanc	
	On souhaite générer une carte avec les paramètres par défaut et rechercher un mot clé qui n'existe pas dans le log	1.Se placer dans le dossier qui contient le jar executable (imagelog.jar) 2.Ouvrir un terminal et saisir la ligne de commande suivante : \$ java -jar imagelog.jar -o ./output/mosaic/ -l ./input/fichierdelog.txt -k motcleabsent	un dossier output est créé et les mosaïques qui composent la carte sont générées (taille 900*1200px). Aucune mosaïque n'est coloriée en rouge	
	On souhaite générer une carte avec un fichier de log vide	1.Se placer dans le dossier qui contient le jar executable (imagelog.jar) 2.Ouvrir un terminal et saisir la ligne de commande suivante : \$ java -jar imagelog.jar -o ./output/mosaic/ -l ./input/fichierdelog_vide.txt	un dossier output est créé et une image vide est générée	
	On souhaite générer une carte avec un fichier de log qui comporte un erreur à la ligne x	1.Se placer dans le dossier qui contient le jar executable (imagelog.jar) 2.Ouvrir un terminal et saisir la ligne de commande suivante : \$ java -jar imagelog.jar -o ./output/mosaic/ -l ./input/fichierdelog_erreur.txt	un dossier output est créé et les mosaïques qui composent la carte sont générées (taille 900*1200px). Une erreur est affichée dans la console : Error trying to read line x in ./input/fichierdelog_erreur.txt	
	On souhaite générer une carte avec un fichier de log inexistant	1.Se placer dans le dossier qui contient le jar executable (imagelog.jar) 2.Ouvrir un terminal et saisir la ligne de commande suivante : \$ java -jar imagelog.jar -o ./output/mosaic/ -l ./input/fichierdelog_inexistant.txt	Le programme se stoppe et une erreur est affichée dans la console : Error trying to read './input/fichierdelog_erreuu.txt'. Check logfile path.	
	On souhaite générer une carte avec des mauvais paramètres	1.Se placer dans le dossier qui contient le jar executable (imagelog.jar) 2.Ouvrir un terminal et saisir la ligne de commande suivante : \$ java -jar imagelog.jar -m mauvais_param	Le programme se stoppe et une information est affichée dans la console : Usage: java -o <outputPath> -l <logFilePath> [-w <width>] [-h <height>] [-k keywords1;keywords2;...;keywordsN ]	
VISUALISATION	On souhaite visualiser une carte	1. Ouvrir l'outil de visualisation avec un navigateur récent (ex : chrome) : ~/path/Orange/viz/index.html 2. Naviguer	La carte s'affiche dans le navigateur et on peut naviguer dedans	
	On souhaite visualiser une carte mais les mosaïques ne sont pas disponibles	1. Supprimer le dossier mosaic du répertoire ~/path/Orange/viz/ s'il existe 2. Ouvrir l'outil de visualisation avec un navigateur récent (ex : chrome) : ~/path/Orange/viz/index.html	Un logo d'image cassée s'affiche en plein milieu de la page	
	On souhaite zoomer au delà d'une mosaïque la plus profonde	1. Ouvrir l'outil de visualisation avec un navigateur récent (ex : chrome) : ~/path/Orange/viz/index.html 2. Zoomer en cliquant sur une zone de l'image jusqu'à arriver au zoom le plus profond 3. Effectuer un clique supplémentaire pour tenter de zoomer au delà du maximum	Le zoom s'effectue mais revient ensuite au zoom maximum	

Nombre de films	450*600	900*1200	1200*1600
10	0	0.1	0.01
100	0.02	0.03	0.03
1000	0.15	0.31	0.33
10000	1.81	2.52	3.5
100000	28.46	36.79	42.85



L'objectif de ce document est d'expliquer comment utiliser l'outil de génération et de visualisation d'une carte

# Génération

## Lancement de l'outil

Le programme prend la forme d'un fichier .jar exécutable directement en ligne de commande avec des options ou avec un fichier de configuration.

### Les options sont :

- Le chemin de sortie des images sur le disque
- Le chemin d'entrée du fichier de log
- La largeur d'une mosaïque (optionnel)
- La hauteur d'une mosaïque (optionnel)
- Des mots clés cibles (1 à 3) qui seront mis en surbrillance dans l'image (optionnel)
- Une mosaïque à approfondir si besoin pour la lecture (optionnel)

```
java imageLog.jar -o < outputPath > -l < logFilePath > [-w < width >] [-h < height >] [-k  
keywords1;keywords2;...;keywordsN ]
```

# Visualisation

Pour lancer l'outil de visualisation, [./viz/index.html](#) à ouvrir avec un navigateur web.

L'objectif de ce document est d'expliquer le cheminement qui a abouti à cet outil de génération de mosaïques.

## Contexte

### Orange Labs

Orange Labs est la division recherche et développement du groupe Orange. Orange est une entreprise française de télécommunications. Elle emploie près de 172 000 personnes, dont 105 000 en France, et sert près de 226 millions de clients dans le monde.

Fin 2012, Orange détient 7 493 brevets au niveau mondial, dont 291 déposés sur les 12 derniers mois.

### Image Log

Orange souhaite améliorer son système de recommandation automatique, en particulier pour son service de vidéo à la demande. Le principe des systèmes de recommandation vidéo est de suggérer aux utilisateurs les films susceptibles de les intéresser le plus, à partir de fichiers regroupant les achats des gens ou leurs notations sur les films visionnés. Pour cela, Orange utilise des méthodes de factorisation de matrices complexes et confidentielles. Ces algorithmes fournissent pour chaque film une coordonnée en deux dimensions.

Pour le moment quatre personnes travaillent sur la factorisation de matrice à Orange Labs. Ce projet a débuté au printemps 2013, suite aux travaux de thèse de Franck Meyer.

Le problème est de réussir à visualiser en deux dimensions sur une image le résultat de ces factorisations. En effet, le nombre de films étant très grand, il est impossible de générer une image de telles dimensions. La solution consiste donc à faire du “clipping”, c'est à dire découper la grande image en sous images plus petites.

Image Log est une application qui permet, à partir de fichier de log de films projetés en 2 dimensions, de générer une image et de naviguer dedans à l'aide d'un navigateur.

# Génération

## Fichier d'entrée

Le fichier pris en entrée par l'outil de génération de la carte est le suivant :

```
film1 -30.70167667 -21.12435552
film2 -42.83304432 44.17294808
film3 -97.99147845 66.23100067
film4 10.06499901 -86.50034343
film5 -19.67704963 -37.67695622
```

## Architecture de l'outil :

- Nous avons décomposé le code de notre outil en Modèle-Controlleur
  - Nous avons choisi une méthode de génération des mosaiques récursive
  - Les mosaiques sont notées selon la règle suivante  
mosaic[ZoomLevel].[Rang].[Colonne]
1. On génère mosaic2.0.0, mosaic2.0.1, mosaic2.1.0 et mosaic2.1.1 (en rouge).
  2. Puis elles sont assemblées pour former mosaic1.0.0 (en rouge).
  3. C'est ensuite au tour de mosaic2.0.2, mosaic2.0.3, mosaic2.1.2 et mosaic2.1.3 d'être générées
  4. Puis assemblées pour former mosaic1.0.1
  5. Et ainsi de suite

**Pour plus d'information concernant la modélisation architecturale de l'outil, veuillez consulter le livrable unique du projet**

# Visualisation

**Nous avons utilisé un squelette html pour la navigation dans les mosaiques**



# Rapport HES

## Rappel du projet

Orange souhaite améliorer son système de recommandation automatique, en particulier pour son service de vidéo à la demande. Le principe des systèmes de recommandation vidéo est de suggérer aux utilisateurs les films susceptibles de les intéresser le plus, à partir de fichiers regroupant les achats des gens ou leurs notations sur les films visionnés. Pour cela, Orange utilise des méthodes de factorisation de matrices complexes et confidentielles. Ces algorithmes fournissent pour chaque film une coordonnée en deux dimensions. Orange Labs nous demande de développer un utilitaire de génération et de visualisation de ces images qui ne sera dans un premier temps pas destiné au grand public mais aux chercheurs qui travaillent sur la factorisation de matrices.

L'efficacité des systèmes de recommandation est en constante augmentation et permet aux entreprises d'augmenter leurs bénéfices, mais leur coût énergétique est immense. En effet la factorisation de matrices pour générer les recommandations les plus pertinentes nécessitent une puissance de calcul et de stockage immense. On peut alors se demander quelles stratégies pourraient être mises en place pour informer l'utilisateur sur la consommation énergétique d'un tel service. L'organisme Nantais *Green Code Lab* a par exemple pensé à une échelle allant de A à G comme pour les appareils électroménagers qui serait affichée sur les sites internet, pouvant motiver les utilisateurs à n'utiliser que les moins gourmands.

Nous allons dans ce document rappeler les enjeux environnementaux en informatique, et les solutions déjà proposées pour réduire son empreinte énergétique. La dernière partie du rapport comprend un interview de Olivier Philippot que nous avons réalisé le mercredi 7 mai ainsi qu'une analyse de notre projet vis à vis de cette problématique.

## 1) Constat environnemental

Il est très difficile d'obtenir des statistiques précises sur la consommation d'énergie par les centres de traitement des données. Les chiffres publiés par les groupes de recherche sur la consommation de l'IT sont donc des estimations. Les impacts environnementaux des centres de traitement données ont lieu lors de :

- la fabrication de ces centres : des bâtiments, des équipements liés aux bâtiments (groupes froid, groupes électrogènes, onduleurs, etc.) et des équipements informatiques et télécoms qu'ils contiennent

- l'utilisation du centre de données (énergie consommée notamment pour le refroidissement des serveurs)

D'après Qarnot Computing, en France, en 2013, plus de 200 centres de traitement de données consomment plus de 7 % de l'électricité du pays. D'après le livre blanc d'Orange sur les solutions pour réduire la consommation énergétique dans l'IT, les enjeux sont de 3 sortes :

- Le prix de l'énergie augmente de façon constante. Les coûts d'électricité en Europe pourraient subir une hausse de 30% d'ici à fin 2016. L'augmentation de ces coûts peut donc impacter de façon importante les marges des entreprises
- La part énergétique du coût des équipements a été multipliée par 10 en 20 ans
- L'augmentation en besoin de calcul et de stockage entraîne l'augmentation de la consommation d'énergie de l'IT

# Classement des entreprises

Entreprise	Indice d'énergie propre (Clean Energy Index)	Charbon	Nucléaire	Transparence en matière d'approvisionnement énergétique	Lieux d'implantation des infrastructures	Efficacité énergétique et atténuation des GES	Plaidoyer en faveur des renouvelables
 <b>Akamai</b>	NA	NA	NA	A	C	B	D
 <b>amazon.com</b>	13.5%	33.9%	29.9%	F	F	D	F
 <b>apple</b>	15.3%	55.1%	27.8%	D	F	D	D
 <b>DELL</b>	56.3%	20.1%	6.4%	C	C	C	D
 <b>facebook</b>	36.4%	39.4%	13.2%	D	B	B	C
 <b>Google</b>	39.4%	28.7%	15.3%	B	C	B	A
 <b>hp</b>	19.4%	49.7%	14.1%	C	D	B	C
 <b>IBM</b>	12.1%	49.5%	11.5%	C	D	C	D
 <b>Microsoft</b>	13.9%	39.3%	26%	C	D	C	C
 <b>ORACLE</b>	7.1%	48.7%	17.2%	D	D	C	D
 <b>rackspace HOSTING</b>	27%	31.6%	22.3%	C	C	C	C
 <b>Salesforce</b>	4%	33.9%	31%	B	C	C	C
 <b>twitter</b>	21.3%	35.6%	12.8%	F	D	F	D
 <b>YAHOO!</b>	56.4%	20.3%	14.6%	C	B	B	B

Fig 1. Classement des entreprises - GreenPeace : "Votre Cloud est il vert ?

## 2) Solutions qui existent pour limiter la consommation des centres de traitement de données

Des solutions sont déjà en place pour réduire l'impact environnemental des centres de traitement de données. En voici trois exemples.

### Facebook installe ses serveurs en Suède

Pour refroidir ses serveurs à moindre coût, Facebook a fait le choix en juin 2013 d'implanter ses serveurs dans une région où il fait naturellement froid.

Source : [lesaffaires.com](http://lesaffaires.com)

“En juin 2013, le géant d'internet a inauguré là-haut, à plus de 700 km au nord de Stockholm, son premier centre de données en dehors des États-Unis. [...] Il y a le climat subarctique, grâce auquel la température moyenne reste sous les 10° neuf mois par an. La plupart du temps, c'est le vent qui rafraîchit Facebook gratuitement, ce qui modère la facture que génèrent ces serveurs, monstres de consommation électrique.”

### **Google Maps passe au vectoriel**

En Décembre 2013, Google Maps a pris la décision de ne plus utiliser des images pour afficher les cartes mais d'utiliser des vecteurs. Les images étaient très lourdes à charger donc leur affichage consommaient beaucoup d'énergie. Grâce aux images vectorielles, Google Maps a permis de réduire énormément la facture d'énergie puisque plusieurs centaines de millions d'utilisateurs utilisent ce service chaque jour.

Source : [appleinsider.com](http://appleinsider.com)

### **La virtualisation des serveurs**

Une solution intéressante pour économiser de l'énergie consommée par les serveurs est de virtualiser ses serveurs : plutôt que d'acheter des serveurs qui ne seront que partiellement utilisés (la nuit par exemple), on loue à une entreprise la quantité de serveurs dont on a besoin.

“La virtualisation des serveurs consiste à partager un serveur physique en plusieurs serveurs virtuels. On considère que la plupart du temps, les serveurs ne sont utilisés qu'à seulement 20 % de leur capacité. La virtualisation des serveurs permet donc d'utiliser au maximum un serveur physique, en faisant l'économie de serveurs supplémentaire. Cette technique permet d'économiser sur les coûts d'achats et de maintenance des serveurs, mais aussi sur les coûts énergétiques induits par la multiplication de ces serveurs.”

Source : [commentcamarche.net](http://commentcamarche.net)

### **3) Pistes de réflexions personnelles**

Pour mieux évaluer l'impact environnemental de notre projet transversal, nous avons interviewé Olivier Philippot, qui travaille sur ces problématiques.

Interview de Olivier Philippot, Consultant en green IT, rencontré le 7 mai 2014



### **Qu'est ce que KaliTerre ?**

KaliTerre est une jeune entreprise innovante. Basée à Nantes depuis sa création en 2010, elle intervient sur un périmètre en lien avec le Développement Durable sur des projets locaux, régionaux et nationaux sur 3 offres majeurs :

- la RSE ou Responsabilité Sociétale des Entreprises,
- le Green IT ou Numérique Eco-responsable
- l'éco-conception des logiciels et des sites / applications Web.

**Dans le cadre de notre projet avec l'entreprise Orange nous nous intéressons à la consommation des centres de traitement de données, quel est l'enjeu environnemental de ces installations ?**

Il devient urgent, tant d'un point de vue économique qu'écologique, de réduire la consommation et de prolonger la durée de vie des équipements informatiques et en particulier des centres de traitement de données. Deux solutions existent pour limiter cette consommation :

- Solutions informatique : mieux Gérer le cycle de vie des données, mieux hiérarchiser le stockage des données...
- Solutions matérielles : organisation spatiale des équipements dans le data center, méthodes de refroidissements...

C'est dans ce cadre que nous proposons une formation permettant aux entreprises d'obtenir une certification européenne : *The European Code of Conduct for Energy Efficiency in Data Centre*.

Ce code est une liste de bonnes pratiques, une sorte d'état de l'art de ce qui est possible de mettre en place en entreprise pour limiter la consommation énergétique des centre de traitement de données. Cette certification permet aux entreprises d'obtenir une note de 1 à 5 sur la

consommation en énergie. La prochaine formation se déroule à Paris début juillet 2014.

**Nous avons fait des recherches documentaires pour pouvoir donner l'impact environnemental de notre projet et donner des chiffres exacts sur l'impact environnemental du numérique, mais il est très difficile de trouver une information précise. Comment faites vous ?**

En effet, il est difficile pour les entreprises d'évaluer la consommation énergétique de leurs services. Certaines commencent à le faire, comme ebay par exemple. Mais ces évaluations nécessitent des instruments de mesure et une méthodologie particulière que la plupart des entreprises n'ont pas. Et c'est là que nous intervenons ! Nous proposons un Audit Green IT. Grâce à une grille d'audit standardisée, nous évaluons le niveau de maturité et préconisons un plan d'action chiffré en terme de gain et de coût.

**Nous avons beaucoup parlé de l'enjeu matériel des projets informatiques, mais il existe aussi une "éco-conception logicielle". Où estimez vous l'enjeu le plus important pour les 10 ans à venir : sur l'impact environnemental du matériel ou celui du code lui même ?**

Il y a eu beaucoup de progrès menés du côté matériel. Je pense que la clé pour le green IT c'est la façon dont les développeurs conçoivent leurs applications.

**Pensez vous que nous pourrions évaluer l'impact environnemental de l'outil que nous avons créé dans le cadre de notre projet ?**

Nous travaillons actuellement au développement d'une application d'audit de code qui s'appelle green spector. Un peu de patience !

**Avez vous eu l'occasion de travailler avec l'entreprise Orange ?**

Nous travaillons justement en collaboration avec Orange sur l'analyse du cycle de vie d'un service. Ce projet a débuté il y a 2 mois et doit durer 2 ans pour une charge de 200 jours répartis entre Orange et Kaliterre. Nous tentons de répondre à plusieurs problématiques : Où se situe l'impact environnemental ? Est ce à la fabrication, à l'usage, au moment de s'en débarrasser ? Comment mesurer cet impact ? C'est ces questions que vous devriez vous poser pour évaluer l'impact environnemental de votre projet.

**Merci.**

## 4) Évaluation de l'impact environnemental de notre projet

Afin de mesurer l'impact environnemental de notre projet, nous avons suivi les conseils d'Olivier Philippot et nous avons mesuré son importance à la fabrication, à l'usage et au recyclage du logiciel que nous avons conçu.

### a. La fabrication

Ce projet a coûté environ 300 heures de travail. Si on considère que les  $\frac{2}{3}$  de ces heures sont nécessaires à l'utilisation de deux ordinateurs consommant chacun environ 200 W/h, alors la facture d'énergie pour la consommation électrique des deux ordinateurs ayant participé au projet est de 80 000 Watts. Soient 1 300 ampoules de 60W allumées pendant 1 heure. A cette consommation il faut ajouter les quelques entretiens téléphoniques passés avec le client.

### b. L'utilisation

Nous ne connaissons pas encore l'utilisation qui sera faite par Orange Labs de notre application. Mais si on émet l'hypothèse que 2 ingénieurs génèrent et visualisent une fois par jour des logs de 100 000 films, alors la consommation électrique pour l'utilisation de notre application s'élève à 4 000 W par mois. Lors du développement du logiciel, nous avons fait attention aux performances principalement pour des questions de rapidité. En optimisant le logiciel pour sa rapidité, nous réduisons aussi sa consommation d'énergie.

### c. Le recyclage

Nous avons fait attention à bien documenter notre projet pour qu'il puisse servir de base à d'autres projets pour Orange Labs. Ainsi, les 80 000 Watts dépensés dans la fabrication de l'application pourront être économisés si un projet similaire voit le jour à Orange Labs.

## Conclusion

Finalement, cette étude environnementale de notre projet et des solutions existantes, nous a permis de réfléchir à l'impact que peut avoir un projet informatique pas seulement en utilisation, mais aussi pour sa fabrication et son recyclage.

## Bibliographie

Orange Business - Livre Blanc : réduire la consommation d'énergie

<http://www.orange-business.com/files/library/livre-blanc-reduire-conso-energie.pdf>

Wikipedia - Centre de traitement de données

[http://fr.wikipedia.org/wiki/Centre\\_de\\_traitement\\_de\\_donn%C3%A9es](http://fr.wikipedia.org/wiki/Centre_de_traitement_de_donn%C3%A9es)

Le Monde - Les centres de données informatiques

[http://www.lemonde.fr/planete/article/2013/07/01/les-centres-de-donnees-informatiques-gros-consommateurs-d-energie\\_3439768\\_3244.html](http://www.lemonde.fr/planete/article/2013/07/01/les-centres-de-donnees-informatiques-gros-consommateurs-d-energie_3439768_3244.html)

GreenPeace - Votre Cloud est-il vert ?

<http://www.greenpeace.org/france/PageFiles/300718/Votre%20cloud%20est-il%20net.pdf>

Wikipedia - Information verte

[http://fr.wikipedia.org/wiki/Informatique\\_verte](http://fr.wikipedia.org/wiki/Informatique_verte)

# Factorisation de matrices pour les systèmes de recommandation

*Bibliographie Scientifique*

Projet Transversal

Orange - Polytech'Nantes

Fabien Richard      Valentin Proust



# **Introduction**

Les systèmes de recommandation s'appuient sur des techniques de recueil de l'information variés et ont des fonctions et des objectifs divers. Mais la plupart sont basées sur le même outils mathématique : la factorisation de matrices.

Pour tenter de comprendre l'intérêt de ces factorisations et leur principe, nous commencerons dans une première partie par rappeler les deux principales stratégies des systèmes de recommandation. Puis nous distinguerons deux grands domaines dans la stratégie des systèmes de recommandation collaboratifs. Enfin nous détaillerons une méthode utilisée pour la factorisation : la décomposition en valeurs singulières.

# **Sommaire**

- I. Les deux stratégies des systèmes de recommandation**
- II. La Méthode des plus proches voisins ou des facteurs latents**
- III. La factorisation de matrices**
- IV. La décomposition en valeurs singulières (SVD)**

# I. Les deux stratégies des systèmes de recommandation

Le filtrage de contenu consiste à dresser un profil des produits en leur affectant différents attributs (genre, acteurs, popularité, etc) et des utilisateurs (âge, sexe, origine démographique, etc). A partir de ces profils, les algorithmes tentent de faire correspondre utilisateurs et produits.

La deuxième stratégie est le filtrage collaboratif. Il est basé sur l'historique comportemental des utilisateurs (ce qu'ils ont acheté ou consulté) et sur leur avis sur les produits. L'avis sur un produit peut être un simple "pouce vers le haut" ou "pouce vers le bas", une note ou un commentaire entier. La stratégie du filtrage collaboratif est confrontée à un important problème : le "démarrage à froid". En effet, sans information sur un nouvel utilisateur, il est difficile de lui proposer des produits correspondants à ses goûts.

# II. La Méthode des plus proches voisins ou des facteurs latents

Dans le filtrage collaboratif on peut distinguer deux grands domaines.

Le premier est celui des plus proches voisins. La méthode des voisins peut être centrée sur les produits ou sur les utilisateurs. On cherche à rapprocher les produits qui obtiennent des notes similaires par des utilisateurs qui ont le même profile.

Le deuxième domaine est celui des facteurs latents. On va chercher à partir d'un grand nombre de facteurs (entre 20 et 100 par exemple) à opposer et rassembler des utilisateurs et des produits. Si les produits sont des films on va par exemple pouvoir trouver des oppositions entre les films d'action et les comédies, les films lents ou rapides, etc.

# III. La factorisation de matrices

Pour pouvoir exploiter la méthode des facteurs latents, la factorisation de matrices est très efficace. En effet, elle permet d'utiliser plusieurs types de données en entrée : les données explicites comme une note ou un commentaire sur un produit et les données implicites comme l'historique de navigation, le mouvement de la souris etc. Ce système s'apparente au SVD (Singular Value Decomposition ou Décomposition en valeurs singulières) qui est une technique très utilisée pour réduire l'information et récupérer les facteurs latents. Cette technique est utile ici à cause du caractère creux des matrices due au très grand nombre de facteurs.

## IV. La décomposition en valeurs singulières (SVD)

La décomposition en valeurs singulières s'appuie sur un théorème d'algèbre linéaire qui dit qu'une matrice rectangulaire A peut être décomposée en le produit de trois matrices (une matrice orthogonale U, une matrice diagonale S et la transposée d'une matrice orthogonale V). On peut présenter ce théorème comme :

$$A_{mn} = U_{mm} \times S_{mn} \times V^T_{nn}$$

où  $U^T U = V^T V = I$ ; les colonnes de U sont les vecteurs propres de  $AA^T$ ; les colonnes de V sont les vecteurs propres orthogonaux de  $A^T A$ ; et S est une matrice diagonale qui contient les racines carrées des valeurs propres de U ou V dans l'ordre décroissant.

1. Calculer la transposée de A
2. Multiplier A par sa transposée
3. Trouver les valeurs propres de  $AA^T$  en résolvant  $\det(AA^T - \lambda I_d) = 0$
4. Pour chaque valeur propre, trouver le vecteur propre correspondant
5. Ces vecteurs propres deviennent les vecteurs en colonne d'une matrice. Ils sont classés dans l'ordre décroissant de leur valeur propre correspondante
6. Pour obtenir la matrice orthogonale U, il suffit enfin d'appliquer le processus d'orthonormalisation de Gram-Schmidt
7. Le calcul de  $V^T$  est identique en se basant sur  $A^T A$  au lieu de  $AA^T$  comme pour et en prenant la transposée
8. Pour trouver S, on prend simplement la racine carrée des valeurs propres non nulles et on les place dans l'ordre décroissant en diagonale :  $s_{11}$  est la plus grande valeur, puis  $s_{22}$ , etc.

Finalement on a réussi à décomposer A en un produit de trois matrices. Les données dans la diagonale de A sont les valeurs singulières de A, les colonnes de U sont les vecteurs singuliers gauches de A et les colonnes de V les vecteurs singuliers droits de A.

## **Conclusion**

Nous avons pu voir que l'intérêt pour les grandes matrices creuses, telles qu'utilisent les systèmes de recommandation, est de réduire leur dimension pour pouvoir extraire des tendances et fournir les meilleures recommandations possibles.

## Références

BAKER Kirk, Singular Value Decomposition Tutorial, 2005 révisé en 2013

KOREN Yehunda, BELL Robert, VOLINSKY Chris, Matrix factorization techniques for recommender systems, IEE Computer society 2009

# Bibliographie technique

*Java 2D pour la génération d'images au format PNG*

*Documentation officielle sur Java 2D*

<http://docs.oracle.com/javase/tutorial/2d/index.html>

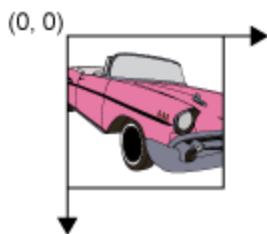
**Cette bibliographie présente les classes de java2D utiles à la réalisation du projet**

## Présentation de l'API Java 2D

Java 2D est une bibliothèque graphique java qui vient en extension de l'AWT qui présentait des manques et des déficiences. Les classes de Java 2D ne sont pas regroupées en un package unique mais ajoutées aux classes préexistantes de l'AWT (Abstract Windows Toolkit).

## Système de coordonnées

L'origine du repère est en haut à gauche de la fenêtre. Les coordonnées sont des nombres entiers qui correspondent à un nombre de pixels.



## La classe Font

Cette classe contient toutes les informations nécessaires pour appliquer une police de caractères à une chaîne.

Un constructeur de la classe Font est : `Font(String name, int style, int size)`

## BufferedImage

Dans l'API Java ™ une image 2D est une matrice bidimensionnelle rectangulaire de pixels, où chaque pixel représente la couleur à cette position de l'image et où les dimensions représentent la largeur et la hauteur de l' image telle qu'elle est affichée.

La classe la plus importante pour représenter ces images est `java.awt.image.BufferedImage`. L'API Java 2D stocke le contenu de ces images en mémoire afin qu'elles puissent être directement accessibles. La classe `BufferedImage` fournit des méthodes pour lire et écrire des images dans la mémoire tampon.

```
// Créer une image en mémoire tampon à partir des images types prédéfinis :  
new BufferedImage(width, height, type, colorModel)
```

## Graphics 2D

La classe `BufferedImage` ne fournit pas directement des méthodes de dessin. Il faut créer un objet `Graphics` avec la méthode `BufferedImage.createGraphics()`

```
//Créer un objet Graphics2D qui peut être utilisé pour écrire dans l'objet BufferedImage :  
createGraphics()
```

Plusieurs méthodes de la classe `graphics2D` sont intéressantes :

Appliquer la police de caractères au composant :  
`setFont(Font font)`

Positionner la chaîne de caractères passée en paramètre sur le composant `graphics2D` :  
`drawString(String str, float x, float y)`

## Point 2D

La classe `point` crée un point représentant un emplacement dans l'espace de coordonnées (x,y). Les sous-classes `Point2D.Float` et `Point2D.Double` permettent d'utiliser des coordonnées respectivement de type `float` et `double`.

```
// Crédation Point2D.Double  
Point2D.Double points = new Point2D.Double (x, y);
```

En outre la classe Point2D a des méthodes de calcul de distances.

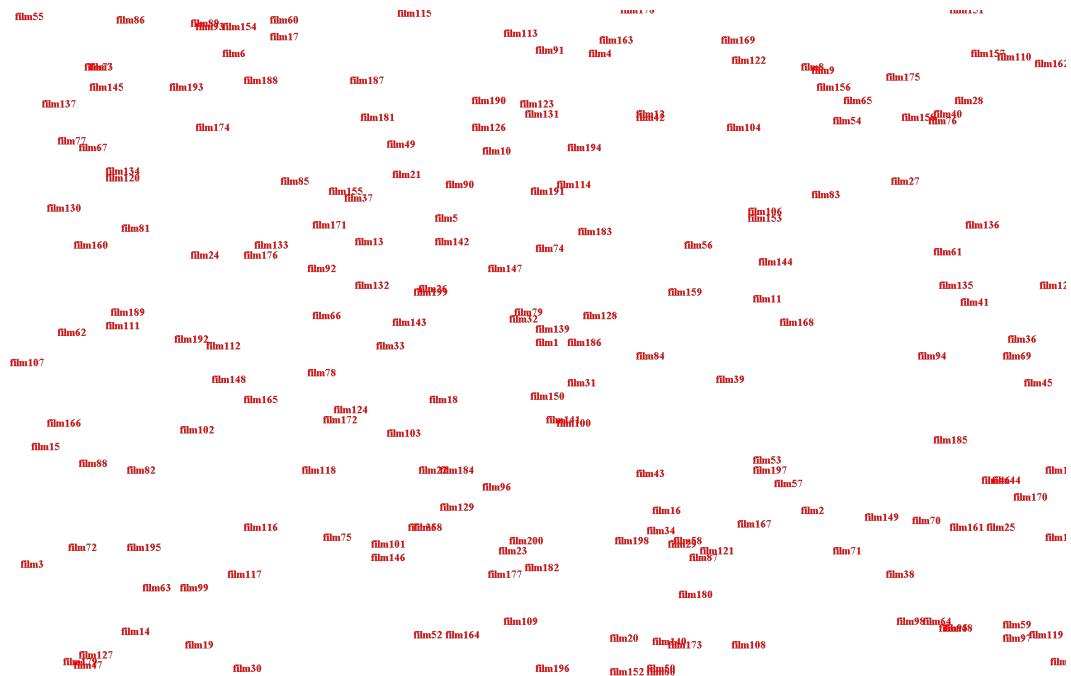
## Ellipse 2D

Créer une ellipse définie par un rectangle.

`new Ellipse2D.Double(double x, double y, double w, double h)`

## Premiers tests

Grâce à cette librairie, nous avons pu produire l'image présentée ci-dessous. Elle représente les films sur une carte à partir des coordonnées fournies en entrée.



# La recherche des plus proches voisins

*Bibliographie Scientifique*

Projet Transversal 2013-2014

Orange – Polytech’Nantes

Fabien Richard                    Valentin Proust

Février 2014



## Introduction

Une partie du programme que nous avons à réaliser pour Orange Labs dans le cadre de notre projet transversal, consiste à mettre en valeur le plus proche voisin pour l'ensemble des films. C'est pourquoi il est important de comparer et d'étudier les différentes techniques de recherche de voisins les plus proches qui existent pour pouvoir choisir celle qui sera la plus adaptée à notre cas. Pour cela, nous commencerons par définir ce que doit faire l'algorithme et nous détaillerons ses différentes applications. Puis nous soulignerons l'intérêt que cette étude a pour notre projet. Enfin, nous étudierons les méthodes de recherche exacte puis les méthodes de recherche approchée.

## **Sommaire**

I. Définition

II. Applications générales

III. Intérêt pour notre projet

IV. Algorithme naïf

V. Algorithmes efficaces

## I. Définition

La recherche du plus proche voisin est un problème d'optimisation pour trouver le point le plus proche d'un point donné.

Considérons :

- un espace  $E$  de dimension  $D$  ;
- un ensemble  $A$  de  $N$  points dans cet espace ;

La recherche du plus proche voisin consiste, étant donné un point  $x$  de  $E$  n'appartenant pas nécessairement à  $A$ , à déterminer quel est le point de  $A$  le plus proche de  $x$ .

## II. Applications générales

La recherche de voisinage est utilisée dans de nombreux domaines :

- la reconnaissance de formes
- le clustering (ou regroupement en classes)
- l'approximation de fonctions
- la prédiction de séries temporelles et même les algorithmes de compression (recherche d'un groupe de données le plus proche possible du groupe de données à compresser pour minimiser l'apport d'information).
- détection de plagiat
- systèmes de recommandations

## III. Intérêt pour notre projet

Les ingénieurs d'Orange Labs qui travaillent sur les algorithmes pour le système de recommandation du service de vidéo à la demande, veulent pouvoir visualiser, pour chaque film A, le film B qui lui est le plus proche. On peut imaginer que dans un système de recommandation, la première suggestion est la plus importante, puisque c'est celle-ci que l'utilisateur regardera en premier. Si elle n'est pas pertinente, alors il y a des chances que l'utilisateur ne regarde pas les suggestions suivantes. Cette étude bibliographique des techniques existantes pour le calcul des plus proches voisins nous est essentielle. En effet, le nombre important de données peut ralentir considérablement la génération des images.

## IV. Algorithme naïf

Recherche linéaire des  $k$  plus proches voisins (Wikipédia) :

```

pour i allant de 1 à k
    mettre le point D[i] dans proches_voisins
fin pour
pour i allant de k+1 à N
    si la distance entre D[i] et x est inférieure à la distance d'un des points de proches_voisins à x
        supprimer de proches_voisins le point le plus éloigné de x
        mettre dans proches_voisins le point D[i]
    fin si
fin pour
proches_voisins contient les k plus proches voisins de x

```

## V. Algorithmes efficaces

Il existe deux types d'algorithmes de recherche des plus proches voisins :

- les algorithmes de recherche exacte qui ne sont pas beaucoup plus performants que les algorithmes linéaires
- les algorithmes de recherche approchée qui acceptent de manquer des points voisins de la requête, mais qui, en contrepartie, sont beaucoup plus rapides.

Les méthodes que nous présenterons ici sont basées sur une structure de données appelée “arbre kd”.

### 1. Présentation des arbres kd

Un arbre kd est une structure de donnée pour diviser de manière hiérarchique l'espace étudié. Ils sont des cas particuliers des arbres à partitionnement binaire de l'espace.

- Chaque nœud contient un point en dimension k.
- Chaque nœud non terminal divise l'espace en deux demi-espaces.
- Le noeud courant contient les points situés dans chacun des deux demi-espaces dans ses branches gauche et droite.

Afin d'avoir un arbre équilibré, le point inséré dans l'arbre à chaque étape est celui qui a la coordonnée médiane dans la direction considérée.

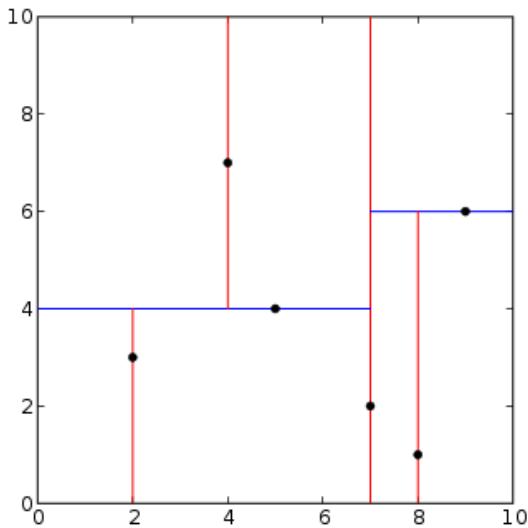


Fig. 1 Décomposition en arbre kd pour le jeu de données : (2,3), (5,4), (9,6), (4,7), (8,1), (7,2).

On en déduit l'arbre kd suivant:

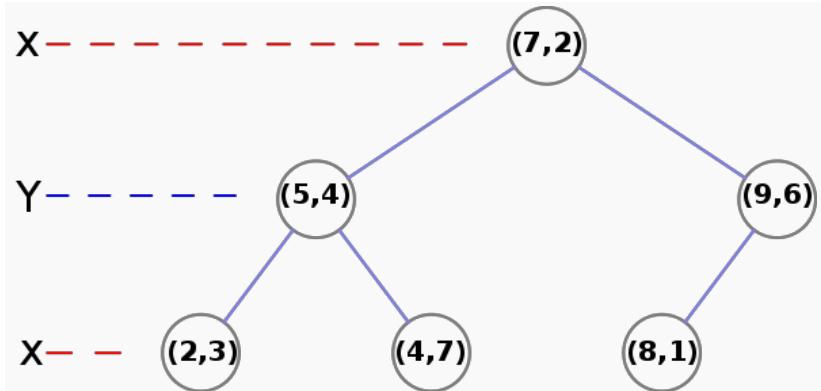


Fig. 2 Arbre kd résultant

Construction de l'arbre dans l'exemple ci-dessus :

- L'espace est divisé en deux parties par le point de coordonnées (7,2) selon un axe orthogonal à (Ox).
- Les deux sous espaces sont ensuite divisés par les points (5,4) et (9,6) selon un axe orthogonal à (Oy).
- Et ainsi de suite jusqu'à ce que tous les points divisent une partie de l'espace.

## 2. Méthode de recherche exacte des plus proches voisins basée sur les arbres kd.

Cette algorithme privilégie la profondeur (depth-first). En effet, on a plus de chance de trouver les plus proches voisins au début, et donc de limiter le nombre de branches à visiter ensuite.

Les étapes de recherche sont les suivantes :

- Recherche du noeud (c'est à dire la région) dans lequel se trouve le point choisi.
- Sélection des points de la feuille atteinte.(c'est à dire la région)
- A chaque noeud visité précédemment, on regarde si la branche non visitée peut contenir des voisins, si c'est le cas, on visite cette branche.

### 3. Méthodes approchées

L'Université de Maryland aux États-Unis a développé la librairie ANN (Approximate Nearest Neighbor Searching) qui permet d'utiliser différentes structures de données basées sur les arbres kd. Cette librairie permet l'utilisation de nombreuses distances (comme la distance Euclidienne, la distance de Manhattan ou encore la distance max) et permet de résoudre le problème des plus proches voisins de manière exacte ou approchée.

Source : <http://www.cs.umd.edu/~mount/ANN/>

#### 3.1 Locality-Sensitive Hashing (LSH)

Cette méthode consiste à utiliser une fonction de hachage afin de regrouper les points qui sont dans des zones similaires dans des mêmes "paquets" (*buckets*). On détaillera ici cette méthode pour la similarité cosinus et la distance Euclidienne.

##### a. Similarité cosinus

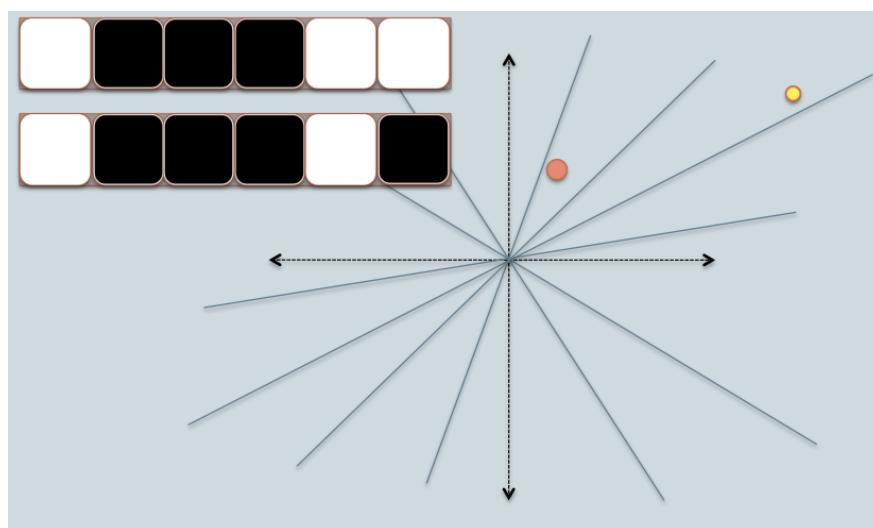
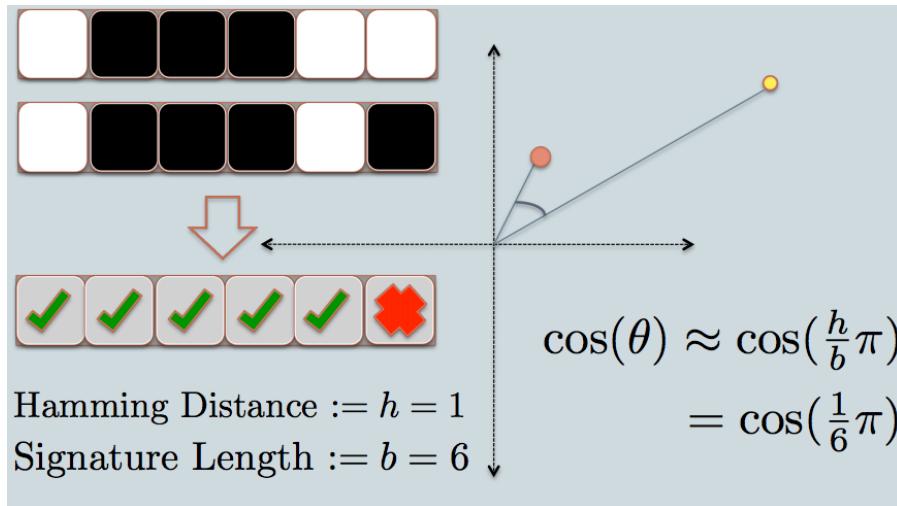


Fig. 3 Illustration de la clé de hachage

Sur la figure ci-dessus, les deux points (jaune et rouge) représentent deux données dans un espace à deux dimensions. L'objectif est de trouver leur similarité cosinus avec la méthode de LSH. Les axes gris sont pris au hasard uniformément. En fonction que le point se trouve au dessus ou en dessous de l'axe, il est marqué par un 1 (blanc), sinon par un 0 (noir) dans la signature – *sketch* – (les blocks en haut à gauche sur la figure). Pour être précis dans le calcul de la différence entre les deux points, un grand nombre d'axes est nécessaire puisque seuls les axes situés entre les deux points expliqueront la différence.



*Fig. 4 Explication du calcul de la similarité cosinus*

Dans cet exemple, 6 bits sont utilisés pour représenter les points. Ils correspondent au hachage LSH des données de départ. Les points qui ont une clé de hachage identique ont une forte probabilité de se trouver dans la même région de l'espace de recherche. La similarité cosinus est calculée en faisant :

$$\cos(\theta) \approx \cos(h/b * \pi)$$

avec  $h$  la distance de Hamming entre les deux clés (nombre de bits différents, ici 1) et  $b$  la longueur de la signature (ici 6 bits)

On peut alors trouver les plus proches voisins d'un point de manière plus rapide en ne cherchant que parmi les points qui ont des clés de hachage similaires. La précision du résultat dépend de la longueur de la clé :

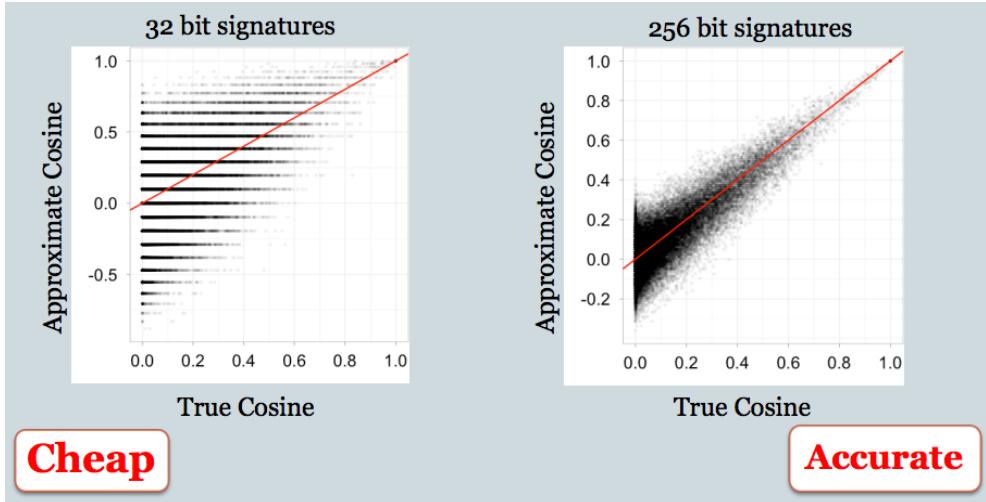


Fig. 5 Précision de la méthode LSH pour la similarité cosinus en fonction de la taille de la clé de hachage

Source des images : <http://www.cs.jhu.edu/~vandurme/papers/VanDurmeLallACL10-slides.pdf>

### b. Distance Euclidienne

Nous expliquerons cette méthode seulement pour un espace à deux dimensions, mais elle fonctionne aussi pour des espaces de plus grande dimension.

Le principe est le suivant : chaque fonction de hachage  $f$  dans notre famille  $F$  sera associée à une ligne prise au hasard dans l'espace de recherche. On choisit une constante  $a$  et on divise la ligne en segments de longueur  $a$ . Ces segments correspondent aux “packets” (*buckets*) des fonctions de hachage par leur projection sur la ligne. Une fois les fonctions de hachage appliquées, pour trouver les plus proches voisins d'un point, on peut restreindre l'espace de recherche aux points qui ont une clé de hachage identique pour accélérer la recherche.

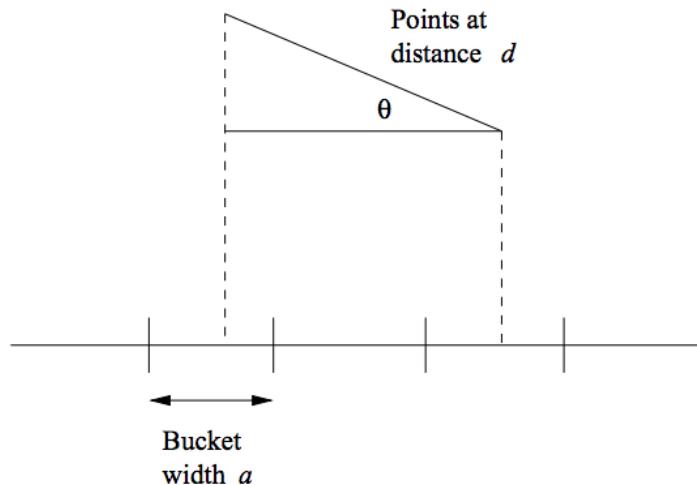


Fig. 6 Les points éloignés ( $d>>a$ ) sont rarement dans le même packet.

*Source de l'image : <http://infolab.stanford.edu/~ullman/mmds/ch3.pdf>*

### **3.2 Best Bin First (BBF)**

La méthode de Best Bin First est une variante des arbres kd. Le principe est le suivant : comme pour la méthode LSH, on recherche seulement dans des sous espaces de recherche (*bins*) par ordre croissant de la distance au point interrogé. La distance entre le sous espace et le point interrogé est la distance minimale entre ce point et tout point situé sur la limite du sous espace de recherche.

*Source : <http://image.ntua.gr/iva/files/ann.pdf>*

## Conclusion

Nous avons pu comparer et étudier les différentes méthodes de recherche des plus proches voisins pour pouvoir choisir une technique adaptée à notre projet. L'objectif de cette comparaison était de trouver une méthode de recherche qui donne un résultat précis en un temps raisonnable. Nous avons opté pour un algorithme linéaire. En effet, la génération des mosaïques implique déjà de faire un tri des films en fonction de leur position (coordonnées  $x,y$ ). On peut alors réduire l'espace de recherche aux films appartenant à la même mosaïque (voire aux mosaïques voisines pour les cas où le film le plus proche se trouve sur une mosaïque différente).

## Références

GORISSE David, K Nearest Neighbours Search, Janvier 2009,  
<http://perso-etis.ensea.fr/~davigori/master/knn.pdf>

FLEURY Cédric, Le kd-Tree : une méthode de subdivision spatiale, novembre 2007,  
[http://cedric.fleu.free.fr/media/rapportCTR\\_cfleury.pdf](http://cedric.fleu.free.fr/media/rapportCTR_cfleury.pdf)

RAJARAMAN Anand and ULLMAN Jeff, Mining of Massive Datasets, juillet 2012,  
<http://infolab.stanford.edu/~ullman/mmds.html>

VAN DURME Benjamin & LALL Ashwin, Online Generation of Locality Sensitive Hash Signatures, 2010, <http://www.cs.jhu.edu/~vandurme/papers/VanDurmeLallACL10-slides.pdf>

KYBIC Jan and VNUCKO Ivan, Approximate Best Bin First k-d Tree All Nearest Neighbor Search with Incremental Updates, juillet 2010,  
<ftp://cmp.felk.cvut.cz/pub/cvl/articles/kybic/Kybic-CAK-2010-40.pdf>

# Les systèmes de recommandations

*Bibliographie Scientifique*

Projet Transversal

Orange - Polytech'Nantes

Fabien Richard

Valentin Proust



# Introduction

Définition wikipédia :

*“Les systèmes de recommandation sont une forme spécifique de filtrage de l’information visant à présenter les éléments d’information (films, musique, livres, news, images, pages Web, etc) qui sont susceptibles d’intéresser l’utilisateur.”*

Pour comprendre et approfondir cette définition, nous présenterons dans un premier temps les différentes fonctions des moteurs de recommandation. Puis nous verrons en quoi les moteurs de recommandations peuvent apporter des informations plus ou moins ciblées au travers des techniques utilisées. Enfin, nous tenterons d’expliquer leur principe de fonctionnement, sans entrer dans les détails techniques. Mais tout d’abord, demandons nous pour quelles raisons autant de moyens sont mis en oeuvre pour recommander de l’information.

## Pourquoi les systèmes de recommandation ?

De plus en plus de choix en contenu (films, livres, musiques, articles, etc) et en équipement (vêtements, électronique, etc) est offert aux consommateurs sur les sites marchands et d’information. Il est dans l’intérêt du consommateur et du vendeur de pouvoir proposer du contenu personnalisé qui correspond à ses goûts, ses humeurs, les tendances etc. L’intérêt pour le consommateur est de disposer de l’information qui l’intéresse le plus et d’accéder rapidement à ce qu’il désirait trouver en se rendant sur le système d’information ou même de trouver ce qu’il n’avait pas encore considéré. L’intérêt pour le vendeur est de cibler le comportement de ses clients pour pouvoir lui vendre plus de produits ou encore augmenter le niveau de satisfaction de l’utilisateur pour le fidéliser ou augmenter le bouche à oreille.

# **Sommaire**

**I) Outils d'aide à l'utilisation des systèmes d'information basés sur un système de recommandation**

**II) Techniques qui implémentent ces outils**

**III) Principe du fonctionnement des moteurs de recommandation**

# I) Outils d'aide à l'utilisation des systèmes d'information basés sur un système de recommandation

Différents outils d'aide à l'utilisation des systèmes d'information sont basés sur des moteurs de recommandation. Nous nous intéressons ici à leur principe et à la valeur ajoutée pour le système d'information, mais aussi à la façon dont la recommandation est visualisée.

## 1. Aide à la navigation

Le principe d'aide à la navigation est de proposer à l'utilisateur de visualiser d'autres informations (souvent en rapport à l'information qu'il est actuellement en train de consulter). Cet outil est notamment utilisé dans le e-commerce. L'utilisateur navigue ainsi au gré des recommandations. L'information se présente pour la plupart du temps sous forme d'imagettes avec une brève description, l'objectif étant d'attirer l'oeil du lecteur.

En 2009, l'outil d'aide à la navigation a généré 30% des bénéfices d'Amazon. Cette part importante du chiffre d'affaire montre bien qu'un grand nombre d'utilisateurs *naviguent* sur les systèmes d'information sans réellement savoir ce qu'ils cherchent, à la manière du "lèche-vitrine" du monde réel. Ceci révèle l'importance de la mise en valeur des produits, comme pour la vitrine d'un magasin, à la différence qu'un système d'information peut posséder un très grand nombre de renseignements sur l'utilisateur et donc orienter sa "mise en vitrine" en fonction de son profile, ce qu'un magasin réel ne peut pas faire.

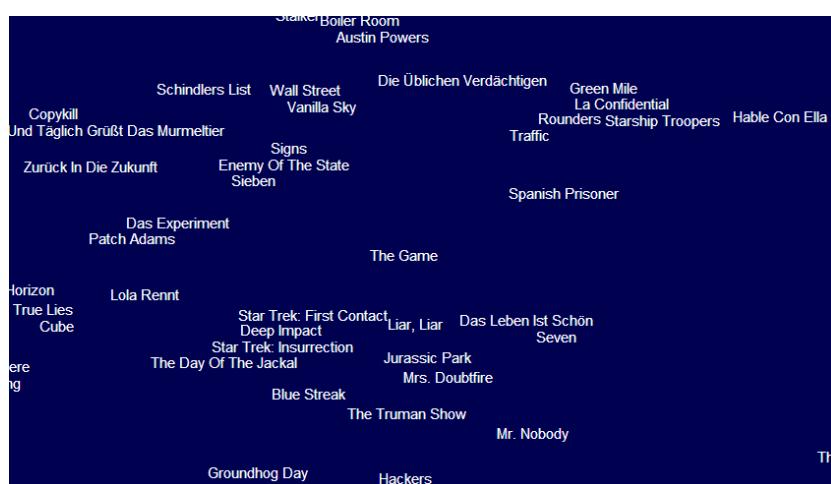
La visualisation peut aussi se faire à l'aide d'une simple liste comme sur les sites de presse.

le 19/11/2013 à 20h19	<b>EN DIRECT.</b> Fusillades à Paris : le photographe de Libé est «réveillé et conscient»
le 19/11/2013 à 13h08	<b>VIDEOS.</b> Fusillades à Paris : une nouvelle photo du tireur solitaire
le 19/11/2013 à 09h52	<b>VIDEO.</b> Fiscalité : Jean-Marc Ayrault annonce une «remise à plat»
le 18/11/2013 à 23h12	Kevin, 230 kg, a finalement été accepté à bord d'un avion Virgin
le 19/11/2013 à 16h36	Remise à plat de la fiscalité : «Juste une réforme technique», estime Copé

*Sur LeParisien.fr, le lecteur peut naviguer d'articles en articles*

## 2. Aide à la décision

L'outil d'aide à la décision permet de faire des prédictions sur les valorisations ou préférences qu'un utilisateur attribuerait à un objet ou à un élément social qu'il n'a pas encore considéré. La valeur ajoutée pour le système d'information est d'aider l'utilisateur dans sa prise de décision, l'incitant à la consommation. Les moyens de visualisation peuvent être des listes (ordonnées ou non), des cartes, des imagettes, sans oublier les commentaires laissés par les autres utilisateurs qui peuvent aider dans la prise de décision.



*“Les gens qui ont aimé le film +The Game+ sont susceptibles d'aimer ces autres films. Plus deux films sont proches sur la carte et plus ils se ressemblent.” Movie Map*

The screenshot shows a movie review by a user named 'sqweegly'. The profile picture is a blue silhouette of a person's head. The review text is:  
★★★☆☆ 3 - Pas mal  
film avec de beaux effets visuels. Dommage que le scénario ne soit pas plus imaginatif. On sait d'avance le déroulement du film et son dénouement.  
Ajoutée le 07 nov. 2013 à 13h15

Below the review, there are interaction icons: a smiley face with the number 3, a red heart with the number 0, and a thumbs-down icon. To the right, there is a link 'Signaler un abus'.

*Forum sur le site Allociné pour la question “Devrais-je aller voir Gravity ?”*

Les commentaires des autres utilisateurs de Allociné peuvent aider l'utilisateur dans sa prise de décision.

### 3. Aide à la comparaison

Cet outil permet d'ordonnancer ou réordonnancer les résultats d'une recherche effectuée par un utilisateur pour lui proposer les résultats les plus pertinents. La visualisation se fait souvent à l'aide d'une liste.

Par exemple, si l'on recherche des contacts professionnels sur LinkedIn, le moteur de recherche nous proposera d'abord des contacts avec qui on a des relations en commun, qui parlent la même langue, etc..

The screenshot shows the LinkedIn search interface with the query 'polytech'nantes' entered. It displays four search results:

- Alexandre Séjourné** (2e)  
Consultant fonctionnel chez Bureau Veritas  
Région de Paris, France  
Technologies et services de l'information  
3 relations en commun · Similaire  
Entreprise précédente : Engineering student in Computer Scien...
- Johan Salmon** (2e)  
2011/2012 en formation MAQSE à Polytech'Nantes  
Région de Nantes, France  
Technologies et services de l'information  
1 relation en commun · Similaire  
Entreprise actuelle : 2011/2012 : D.U. Management Associé Qua...
- Dobromir Manchev** (2e)  
Functional and Technical Test Designer  
Région de Nantes, France  
Technologies et services de l'information  
1 relation en commun · Similaire  
Formation: Polytech'Nantes
- Fabien Richard** (1er)  
Research Assistant chez Orange Business Services  
Région de Nantes, France  
Technologies et services de l'information  
8 relations en commun · Similaire · 62

*Fonctionnalité de recherche sur le réseau social professionnel LinkedIn*

### 4. Aide à la découverte

Cet outil permet à l'utilisateur de consulter des informations auxquelles il n'aurait pas pensé, de lui recommander un objet ou une personne qu'il n'avait pas considéré auparavant. C'est le principe de beaucoup de systèmes de recommandation.

On peut ainsi trouver sur le service de clips musicaux en ligne de Vevo une rubrique "staff picks" (*sélection du personnel*) qui recommande aux utilisateurs des clips qui a priori ne sont pas personnalisés. On peut en effet questionner la question de neutralité de ces suggestions puisque rien nous permet de vérifier que la sélection des vidéos de cette rubrique n'est en fait pas biaisée par les préférences de l'utilisateur.



SEARCH

BROWSE

## Staff Picks



Papaoutai  
Stromae  
76,880,050 views



J'me tire (Official  
Video)  
Maître Gims  
38,976,549 views



Ailleurs  
Black M  
4,651,523 views



I Love You (Official  
Video)  
Woodkid  
5,489,665 views

*Catégorie "recommandation du personnel" sur le site de clips musicaux Vevo*

On retrouve un système similaire d'aide à la découverte sur le réseau social Facebook pour la suggestion d'amitiés :

The screenshot shows the Facebook interface with a blue header bar containing the 'facebook' logo and a search bar with the placeholder 'Trouvez des personnes, des lieux ou d'autres choses'. Below the header, a white box contains the text 'Vous connaissez peut-être...'. It lists two suggestions:

- Jeanne Yan**  
Nantes  
5 amis en commun + Ajouter
- Chloe Shang**  
Université de Nantes  
18 amis en commun + Ajouter

*Fonctionnalité de suggestion d'amitiés sur le réseau social Facebook*

## **II) Techniques qui implémentent ces outils**

Nous avons vu qu'il existe différents outils basés sur des systèmes de recommandations. Nous allons maintenant aborder quelles techniques sont mises en oeuvre pour effectuer ces recommandations.

### **1. Recommandation “éditoriale”**

C'est la méthode de recommandation la plus simple. Elle consiste pour l'administrateur du système d'information à mettre des informations en avant. Par exemple des promotions, Top 10 des ventes, Top 10 apprécié. Cette recommandation n'est pas personnalisée pour l'utilisateur. L'avantage de cette technologie est qu'elle est simple à mettre en oeuvre, en revanche elle ne permet pas de cibler directement l'utilisateur.

### **2. Recommandation “sociale”**

Cette technologie consiste à mettre à disposition de l'utilisateur les avis des autres utilisateurs sur une information. Par exemple coupler un catalogue de produits à un forum. Si l'utilisateur est connecté à un réseau social, il peut consulter l'avis de ses contacts. L'avantage de cette technologie est qu'elle est simple à mettre en oeuvre et qu'elle est personnalisée à l'utilisateur. En revanche, l'inconvénient est que la recommandation n'est pas “automatique”.

### **3. Recommandation contextuelle**

Cette technologie consiste à proposer des informations en rapport avec l'information visualisée. C'est cette technologie qui est utilisée dans les outils d'aide à la navigation. La recommandation se fait en Item-to-Item (mots clés communs, catégorie commune, etc..) ou Item-to-Users ( “les personnes qui ont acheté ces produits ont également aimé ceux ci”).

### **4. Recommandation automatique personnalisée**

C'est la méthode la plus efficace. C'est aussi la méthode la plus difficile à mettre en oeuvre. Elle permet de sélectionner l'information pour l'utilisateur en se basant sur son profil. L'avantage est que la recommandation est ciblée et donc plus pertinente. L'inconvénient est que l'utilisateur doit être connecté au système d'information, que son profil doit contenir des informations.

## III) Principe du fonctionnement des moteurs de recommandation

Dans cette partie, nous commencerons par traiter des deux façons de recueillir de l'information sur les produits et sur les utilisateurs pour pouvoir générer des recommandations pertinentes. Puis nous proposons une explication accompagnée d'exemples, d'avantages et d'inconvénients pour les deux types de moteur de filtrage : collaboratif et thématique.

### 1. Le recueil de l'information

#### 1.1 Recueil actif

La première méthode de recueil est le recueil actif. Elle consiste à recueillir l'information sur les utilisateurs et les objets de manière explicite. Ce type de recueil est très rare car l'utilisateur ne souhaite en général pas consacrer de temps à donner ses goûts au système d'information mais préfère plutôt fouiller dans le site pour rechercher ce qu'il désire. Pourtant, ce type de recueil est le plus précis et le moins biaisé puisqu'on demande explicitement à l'utilisateur ses goûts sans faire aucune déduction erronée.

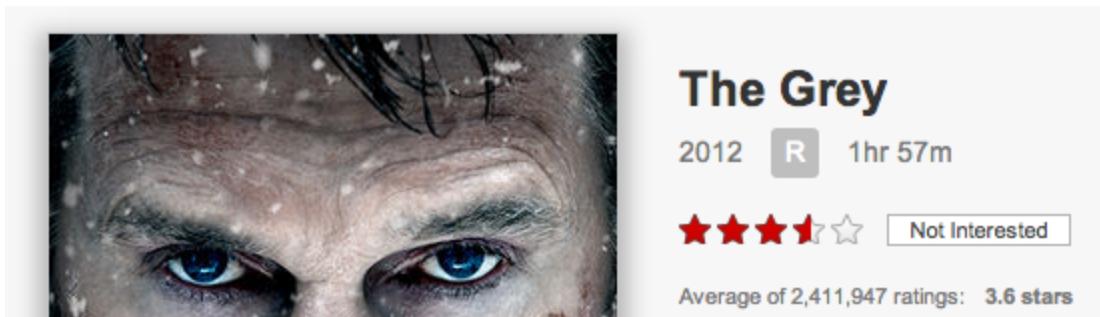
Par exemple, pour recueillir de l'information sur les utilisateurs on peut leur demander leurs goûts. L'application du journal *Le Monde* est l'un des rares systèmes d'information à proposer à l'utilisateur d'indiquer ses préférences en matière de contenu :



*Choix des préférences pour les alertes sur l'application iPhone du journal *Le Monde**

De la même façon, le recueil actif pour les informations sur un objet peut se faire de plusieurs manières. Certains sites comme Amazon, Netflix ou encore le Google Play pour les applications

Android ont choisi un système de notation allant de 0 étoile pour signifier que l'utilisateur n'aime pas du tout le contenu ou l'objet à 5 étoiles qui veut dire que l'utilisateur l'a adoré.



Système de notation du site américain de visionnage de films en ligne Netflix

Ce système de notation simple permet à l'utilisateur de rapidement connaître l'avis des autres utilisateurs sur ce film. Mais il permet surtout au système d'information d'affiner le profile de l'utilisateur en connaissant mieux ses goûts pour pouvoir lui faire des recommandations toujours plus pertinentes. Certains systèmes d'information accompagnent la notation d'un commentaire ou avis qui permet de justifier la note attribuée à l'objet. C'est le cas des trois systèmes d'information cités ci-dessus.

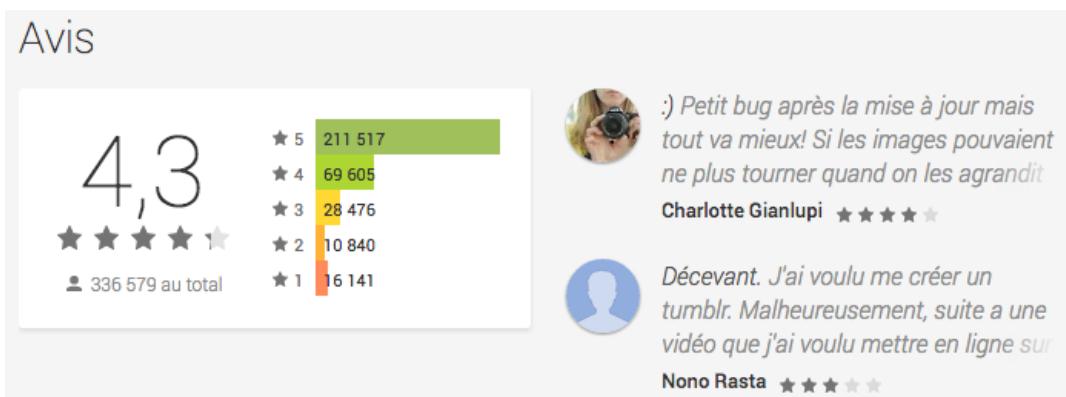


Illustration de la possibilité d'ajouter un avis à la note attribuée à une application sur le Google Play

L'avantage de ce type de recueil d'information est qu'il est très précis et très proche de la réalité puisque l'information est explicitement demandée à l'utilisateur. En revanche, il peut être difficile de recueillir un grand volume d'information de cette manière car il faut réussir à encourager les utilisateurs à vouloir mettre une note et/ou un avis sur un produit qu'il a consommé. Des techniques ont été mises en place pour faciliter la notation comme le système de notation en un clic de Netflix qui n'implique pas le rechargement de la page web. Amazon a fait le choix d'envoyer un email après la réception de la commande qui invite à laisser une note au produit :



So Richard Fabien, how did this item meet your expectations?



Disney Assorted Advent Calendar  
3Pack 1.76oz each

Frankford

[Start by rating it](#)



*Email envoyé par le site marchand Amazon invitant à noter le produit consommé*

Un inconvénient au recueil d'information actif est qu'il peut être biaisé par l'échelle subjective de notation de chaque utilisateur. En effet, certains utilisateurs auront tendance à donner facilement 5 étoiles sur 5 à un produit consommé alors que certains n'attribueront cette note qu'en cas de satisfaction totale. On utilise alors des outils mathématiques comme le recentrage des valeurs pour rapporter ces notations à une échelle "standard" de notation. Un autre paramètre important à prendre en compte dans cette technique de recommandation est le temps. En effet, si un utilisateur a aimé un produit ou un contenu à un instant t, cela ne veut pas dire qu'il l'aimera toujours à un instant t+1. Ce changement de goût peut être du par exemple au fait que l'utilisateur a grandi et donc que ces centres d'intérêts ont évolué. Il peut également être du à une simple évolution de sa manière de penser, en fonction de l'environnement dans lequel il vit ou des tendances. Il faut donc garder à l'esprit que les modèles utilisateurs qui représentent leurs goûts doivent être mis à jour régulièrement.

## 1.2 Recueil passif

En opposition au recueil actif, le recueil passif consiste à recueillir de l'information sans la demander explicitement à l'utilisateur. On peut par exemple se baser sur son historique de consultation ou d'achat, sur son comportement de navigation. Le site internet de visionnage de films en ligne Netflix analyse par exemple en temps réel le déplacement de la souris de l'utilisateur sur la page web. Ainsi, si un utilisateur survole plusieurs fois avec sa souris un titre de film ou son illustration, on pourra en déduire qu'il a hésité à cliquer dessus et donc qu'il a une préférence pour ce film. Amazon, quant à lui, conserve un historique des commandes passées par un utilisateur pour pouvoir établir son profile.

Une fois l'information recueillie, il faut utiliser un moteur de filtrage pour pouvoir recommander du contenu ou des objets de manière pertinente et personnalisée.

## 2. Les moteurs de filtrage

### 2.1 Moteur de filtrage collaboratif

Le filtrage de contenu consiste à dresser un profil des produits en leur affectant différents attributs (genre, acteurs, popularité, etc) et des utilisateurs (âge, sexe, origine démographique, etc à partir de leurs usages. A partir de ces profils, les algorithmes tentent de faire correspondre utilisateurs et produits. Les données viennent pour la plupart de logs d'usage. Pour améliorer son moteur de recommandation collaboratif, Netflix avait mis en place un prix, le *Netflix Prize*, qui invitait les concurrents à tenter de trouver des algorithmes capables de recommander les films mieux que le fait le moteur de recommandation actuel de Netflix. Pour cela, les concurrents travaillaient tous sur le même jeu de données. Le prix de 1M de dollars récompensait seulement l'équipe avec le meilleur algorithme et sa valeur dépendait de la qualité de l'algorithme. Ainsi, Netflix pouvait profiter du travail de recherche de milliers de personnes mais ne récompenser que la meilleure équipe.

## 2.2 Moteur de filtrage thématique

Le filtrage thématique ne fait pas intervenir les habitudes d'usage de l'utilisateur du système d'information. Il fonctionne à partir de thèmes ou caractéristiques des informations. Ce moteur de recommandation est plus simple et son fonctionnement est transparent. Il sert en général de complément. L'inconvénient est que le système d'information nécessite un catalogue riche et bien renseigné et de mettre à jour régulièrement le modèle utilisateur pour prendre en compte ses changements de goûts au cours du temps.

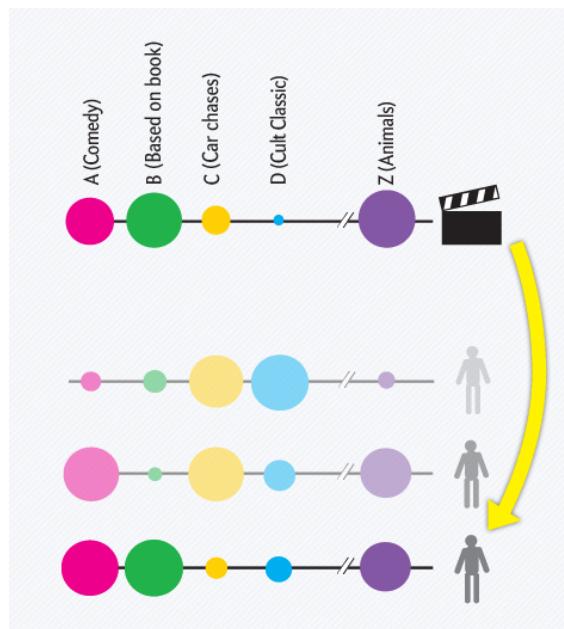


Illustration du principe de filtrage thématique. Source : [scientificamerican.com](http://scientificamerican.com)

## **Conclusion**

Les systèmes d'informations prennent une part de plus en plus conséquente dans les systèmes d'information, notamment dans le e-commerce. Il existe différents outils de recommandation qui se basent sur des informations aussi différentes que des "mots-clé" sur des produits, des logs d'achat ou des notes d'utilisateurs.

## Références bibliographiques

FRANCESCO Ricci, LIOR Rokach and BRACHA Shapira, Introduction to Recommender Systems Handbook

KOREN Yehuda, BELL Robert, VOLINSKY Chris, Matrix factorization techniques for recommender systems, IEE Computer society 2009

MATHIEU N. Les Algorithmes de recommandations - Podcast [en ligne]. In : Podcast Science, France. Site disponible sur : <http://www.podcastscience.fm/dossiers/2012/04/25/les-algorithmes-de-recommandation/> (Page consultée le 10/11/2013)

MEYER Franck. Recommender systems in industrial contexts. Thèse de doctorat d'université. Grenobles : Université de Grenoble, 2012.