

Travaux d'Études et de Recherche

Rapport Intermédiaire

Julien Henry
Nicolas Lacourte-Barbadaux
Alexandre Masson
Léo Rousseau

14 Janvier 2013

Table des matières

1	Résumé du projet	3
2	Introduction du domaine	4
3	Analyse de l'existant	5
4	Besoins non fonctionnels	6
5	Besoins fonctionnels	7
6	Description des prototypes	8
7	Planning, affectation des taches	9
8	Bibliographie	10

1 Résumé du projet

R est un logiciel de statistiques open source. De notre côté, nous développons depuis plusieurs années un logiciel de visualisation de données en java doté d'une interface graphique 3D interactive.

Il s'agit ici d'étudier le portage de cette application vers R, en se focalisant notamment sur les capacités d'interaction des librairies 3D proposées par ce système.

2 Introduction du domaine

La classification La classification est une forme d'apprentissage automatique. Elle consiste, étant donné un ensemble d'éléments, à les répartir dans différentes catégories. Il existe deux courants majoritaires de classification.

La classification supervisée Dans la classification supervisée on cherche à déterminer la étant-donné un ensemble d'apprentissage dont on connaît déjà la classification. Par exemple si on classe des animaux on doit pouvoir déterminer que le cobra est un reptile parce qu'il a une description proche des reptiles déjà présents dans l'ensemble initial.

La classification non-supervisée La classification non-supervisée, ou encore clustering, a un but différent. On cherche à classer un ensemble de données en plusieurs classes sans les connaître à l'avance. On calcule une répartition pour l'ensemble. C'est ce que fait par exemple l'algorithme des K-Moyennes[1].

Le logiciel R Le logiciel R est à la fois un environnement interactif de calcul, en lignes de commandes, et un langage de programmation. Il existe sur différentes plateformes (Linux, MacOS et Windows). Il peut être utilisé pour des traitements de données ou des analyses statistiques. R est un logiciel libre, ouvert à la participation de développeurs extérieurs. En effet, l'environnement s'enrichit par l'intégration de packages provenant de la communauté ajoutant des modules et des fonctionnalités complémentaires au logiciel initial.

Par ailleurs il existe dans R une fonction mean, mais c'est un bon exemple de programmation en R.

3 Analyse de l'existant

R est un logiciel de statistiques open source qui fonctionne par modules [http ://www.r-project.org/](http://www.r-project.org/).

La communauté R est très active et de nombreux modules sont régulièrement proposés. R permet facilement de charger des données, de sortir des informations statistiques, de visualiser diverses courbes extraites des données, etc.

Par défaut, R se présente sous la forme d'une ligne de commande. Il est cependant possible de développer des interfaces utilisateur pour contrôler l'application ainsi que pour visualiser des résultats de traitement.

De notre côté, nous développons depuis plusieurs années un logiciel de visualisation de données doté d'une interface graphique 3D interactive. Ce logiciel est écrit en Java. [http ://www.univ-orleans.fr/lifo/software/Explorer3D/](http://www.univ-orleans.fr/lifo/software/Explorer3D/)

Nous avons acquis une bonne maîtrise de la structuration d'un tel outil afin de le rendre évolutif. Nous avons également mis en place un certain nombre de fonctionnalités interactives. Nous étudions maintenant les possibilités d'interactions entre ce logiciel et la plateforme R, afin d'intégrer rapidement divers outils existant sous R, mais également de diffuser notre logiciel vers cette communauté.

Au moins deux approches sont possibles : 1/ intégrer des invocations de R depuis java. 2/ migrer notre logiciel vers R.

Concernant le premier point, l'invocation de R depuis java est relativement simple. Elle ne sera pas abordée dans ce TER. Concernant le second point, R propose la création de fenêtres de contrôle et de fenêtres graphiques 2D et 3D (voir par exemple [http ://rgl.neoscientists.org/](http://rgl.neoscientists.org/)). Toutefois, nous ne connaissons pas les possibilités d'interaction réelles de l'interface 3D.

L'objectif de ce TER consiste à :

- étudier le mécanisme de développement d'interfaces utilisateur sur le plan 1/ des fenêtres de contrôle, et 2/ des fenêtres de visualisation 3D et 2D ;
- en produire une synthèse ;
- réaliser une maquette dont le contour précis sera discuté avec l'enseignant (charger un jeu de données, visualiser en 3D après préparation des données, sélectionner des objets de la scène 3D grâce à la souris, afficher des images).

4 Besoins non fonctionnels

- Montée en charge
- Interaction
- Documentation des technologies à utiliser

Montée en charge Comme expliqué plus haut, nous allons différer de la méthode utilisée dans Explorer3D pour toute la partie affichage. Pour savoir si les solutions choisies sont acceptables, il sera nécessaire de faire des tests de montée en charge.

Pour cela nous allons tester la réactivité du système aux différents événements, que ce soit des demandes de calculs, ou des interactions avec la scène 3D (rotation, zoom, déplacement, etc...).

Pour réaliser ces tests, nous procéderons à la main aux différentes manipulations, et nous nous baserons à notre jugement en terme de vitesse de réponse, car il n'est pas facile de chronométrer la réactivité de la scène 3D, nous allons bien entendu refaire ces tests avec un nombre de plus en plus grand d'objets à afficher, pour pouvoir déterminer une limite en terme de quantité d'objets représentable.

5 Besoins fonctionnels

Les principaux besoins fonctionnels sont les fonctionnalités d'Explorer3D qu'il est nécessaire de porter sous R, du moins autant qu'il sera possible d'en porter.

Nous allons lister les différentes fonctionnalités souhaitées et nous les commenterons.

Gestion de fenêtres Nous souhaitons pouvoir fournir à l'utilisateur une interface graphique lui permettant d'agir sur les données, offrant diverses possibilités, telles que le chargement de données, ou la demande de calcul. Il doit aussi avoir la possibilité d'ouvrir une nouvelle fenêtre 3D interactive.

Nous allons donc déterminer si il est possible de créer ces différentes fenêtres, aussi nous chercherons à savoir comment pourrions nous agir depuis cette interface utilisateur sur la ou les fenêtres 3D.

Nous allons maintenant préciser ce que nous attendons des fenêtres 3D.

Nous l'avons évoqué dans les besoins non fonctionnels, nous voulons avoir une fenêtre qui nous permettent d'afficher un très grand nombre d'objets, nous ne donnerons pas d'ordre de grandeur pour le moment, car nous en saurons plus après les premiers tests. - gestion de fenêtres de dialogue permettant le chargement de données tabulaire, le calcul de projections, l'ouverture d'une fenêtre 3D interactive.

Nous souhaitons aussi avoir la possibilité de modifier la scène 3D, plus précisément nous souhaitons connaître les possibilités de zoom et de déplacement de la scène dans la fenêtre 3D. Nous nous intéressons aussi à la rotation de la scène, nous aimerions pouvoir faire tourner la scène et ainsi avoir un autre point de vue de la répartition des objets.

- fenêtre 3D interactive, permettant notamment : - zoom et déplacement - mais aussi sélection d'objet par clic - mise en surbrillance des objets sélectionnés - affichage dynamique du repositionnement spatial des objets - ajout au besoin d'objets supplémentaires (e.g. ellipses) - coexistence de plusieurs fenêtres 3D - mécanisme similaire à la "loupe" de Explorer3D (étudier des pistes permettant d'obtenir...) - AJOUT : possibilité de se mettre en écoute sur un port pour réception de commandes de type jeu de données + affichage.

6 Description des prototypes

7 Planning, affectation des taches

8 Bibliographie