

Dynamique des connaissances

Raisonnement à partir de cas

Yohan Michelland et Marion Vertessen

Introduction

Ce rapport se rapporte à un travail pratique réalisé pour le cours de dynamique des connaissances. Il a pour but de créer un système de raisonnement à partir de cas, c'est-à-dire un système capable de résoudre de nouveaux problèmes en utilisant des solutions de problèmes antérieurs résolus. On réalise ici un système permettant de déterminer le montant du loyer d'un appartement.

I - Architecture

On retrouve la structure globale du système sur la figure 1.

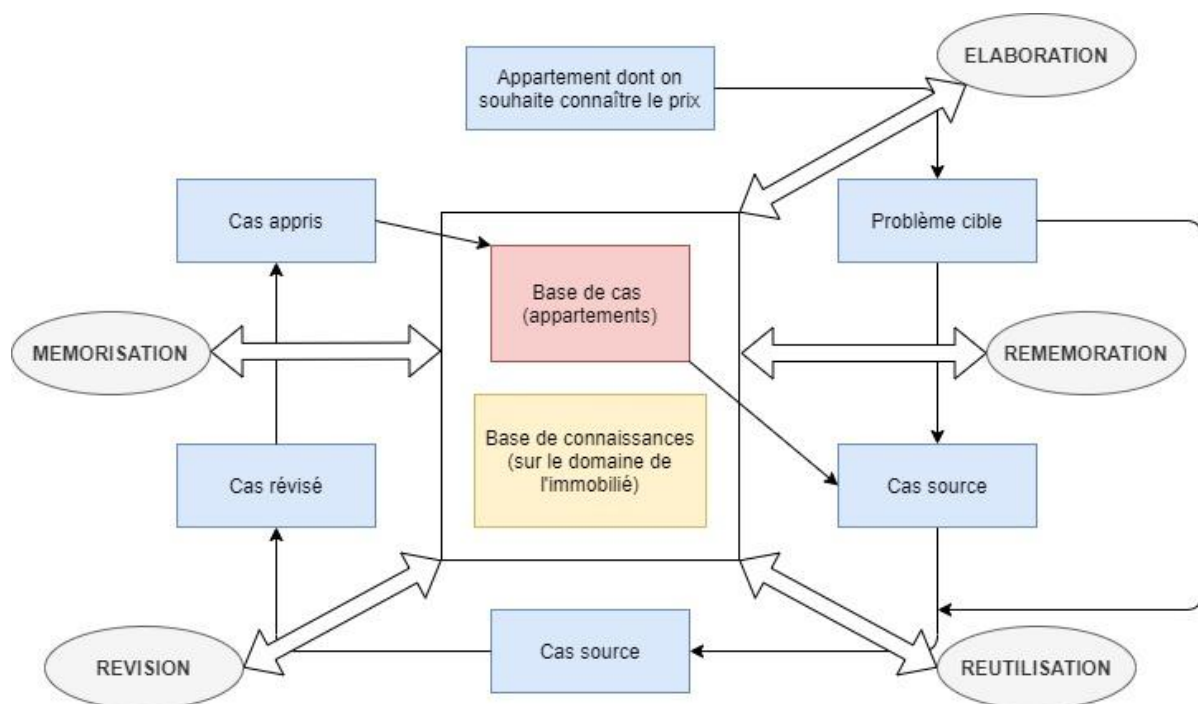


Figure 1 : Système global

Afin de construire notre base de cas, nous nous sommes basés sur de la location d'appartement dans la ville de Lyon. Nous avons donc réalisé un fichier de type csv contenant différents appartement ainsi que leur prix. Ces appartements sont des appartements réellement mis en location sur <https://www.seloger.com/>.

Afin d'avoir une base de cas contenant assez d'informations pour produire une simulation intéressante, nous avons décidé de sélectionner les descripteurs suivants :

- L'arrondissement
- Le nombre de pièces
- Si l'appartement est meublé ou non
- Le nombre de balcons
- L'étage
- Le nombre de parkings
- La surface
- Si l'appartement contient une cuisine équipée ou non
- Le standing de l'appartement (vivable, moyen, bon, très bon)
- Si l'immeuble possède un ascenseur
- Si l'appartement possède une cave
- Le prix de l'appartement

On réalise un arbre de décision à partir des différents cas de notre base de données. Cependant, on ne crée des branches que pour l'arrondissement et le nombre de pièces. En effet, notre base de données est assez petite (17 cas) et ne sera donc pas très pertinente sur un arbre contenant l'ensemble des descripteurs. Nous avons sélectionné l'arrondissement et le nombre de pièces pour la construction de ce-dernier.

Sur la figure 2, on retrouve un aperçu de notre arbre de recherche.

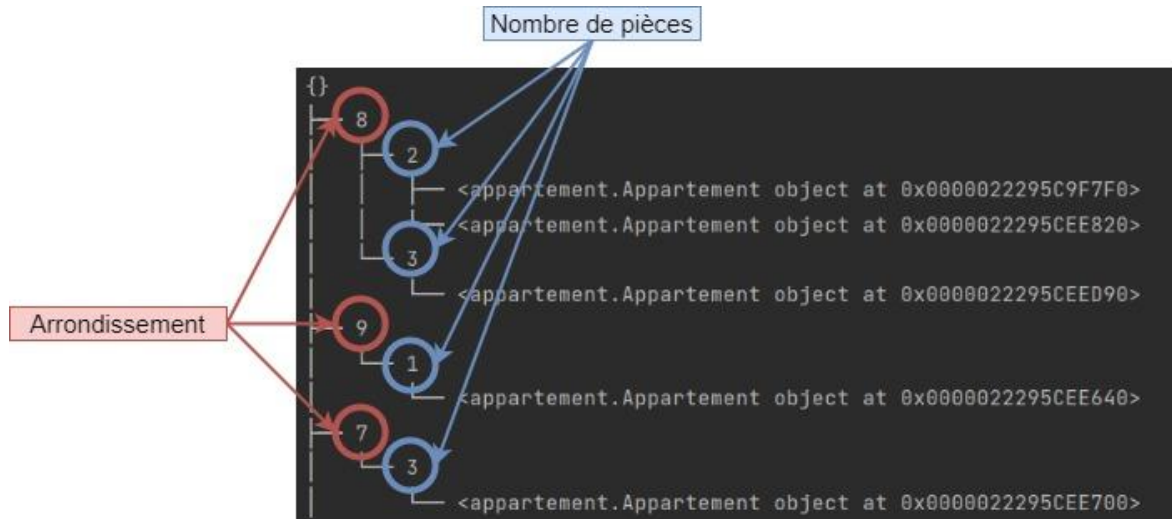


Figure 2 : Arbre de recherche

II - Implémentation d'un appartement

1 - Descripteur numérique

Les descripteurs '*arrondissement*', '*nb_pieces*', '*balcon*', '*parking*', et '*surface*' prennent des valeurs numériques (int ou float) et ne nécessitent pas de traitement supplémentaire

2 - Descripteur catégoriel

Le standing de l'appartement est un descripteur catégoriel. Il prend les valeurs 'Vivable', 'Moyen', 'Bon', 'Très bon'. A l'instanciation de l'appartement, cette valeur est convertie en entiers allant de 0 à 3.

3 - Descripteur booléen

Les descripteurs '*meuble*', '*cuisine_equipe*' et '*cave*' sont des booléens prenant la valeur *True* si l'appartement est meublé (respectivement possède une cuisine équipée ou une cave), et *False* si l'appartement n'est pas meublé (respectivement ne possède pas de cuisine équipée ou une cave). A l'instanciation de l'appartement, cette valeur est convertie en entier entre 0 et 1

4 - Descripteur combiné

Les descripteurs '*etage*' et '*ascenseur*' sont combinés en un seul descripteur. En effet, un étage élevé est souhaitable si l'immeuble est équipé d'un ascenseur, mais non souhaitable en l'absence d'ascenseur. Nous avons donc choisis d'attribuer un signe négatif à l'étage dans le cas où l'immeuble n'est pas équipé d'un ascenseur.

III - Fonction de remémoration

Tout d'abord, la fonction de remémoration prend en entrée un appartement cible (pour lequel on aimerait prédire le prix) et renvoie un appartement source similaire trouvé dans la base de cas (dont le prix est connu).

Pour cela, on parcourt simplement l'arbre construit précédemment, afin de lister l'ensemble des appartements ayant le même arrondissement et le même nombre de pièces que l'appartement cible. Si aucun appartement ne remplit les conditions (par exemple si un arrondissement inexistant a été choisi), on arrête l'algo et on précise à l'utilisateur qu'aucun appartement comparable n'a été trouvé.

Dans un second temps, on choisit dans cette liste l'appartement qui se rapproche le plus de la cible. Pour cela, on peut se servir du delta (décrit plus bas) comme mesure de similarité. L'appartement cible est l'appartement de la liste ayant le delta le plus petit avec la source

IV - Calcul du delta

Le delta entre une source et une cible représente la valeur à ajouter au prix de la source pour déduire le prix de la cible. Il se calcule à partir des descripteurs d'un appartement cible et des descripteurs d'un appartement source. Chaque descripteur d possède un poids $poids(d)$, représentant l'impact de la variation de ce descripteur sur le prix de l'appartement. Un poids positif indique qu'une valeur du descripteur plus grande entraîne une augmentation du prix de l'appartement (par exemple la surface). Ces poids ont des valeurs empiriques, qui sont déterminées par nos connaissances du marché de la location d'appartement. Pour chaque descripteur d , on calcul :

$$dif = (d(cible) - d(source)) * poids(d)$$

et on ajoute cette dif au delta total.

V - Fonction d'adaptation

La fonction d'adaptation prend en entrée un appartement cible (dont on cherche le prix), et l'appartement source (trouvé grâce à la fonction de remémoration et dont on connaît le prix). Elle calcul ensuite simplement le prix de la cible de cette manière

$$prix(cible) = prix(source) + delta(source, cible)$$

VI - Utilisation de l'application

Tout en haut du fichier *main.py*, l'utilisateur peut choisir les descripteurs de l'appartement dont il cherche à prédire le prix. (dans le dictionnaire *Appartement_cible*). Dans le *main*, sont alors appelé :

- *init_base_cas* : remplit la base de case à partir de *Appartement.csv*
- *rememoration* : trouve l'appartement source le plus similaire à l'appartement cible
- *adaptation* : adapte le prix de la source à l'aide du delta, et renvoie le prix prédit de la cible.