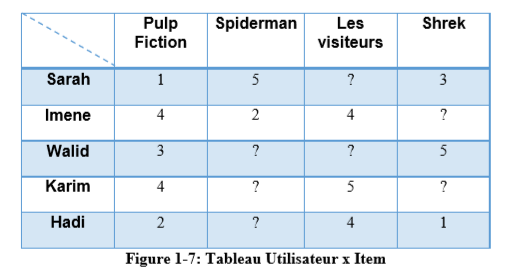
**FACTORISATION MATRICIELLE**

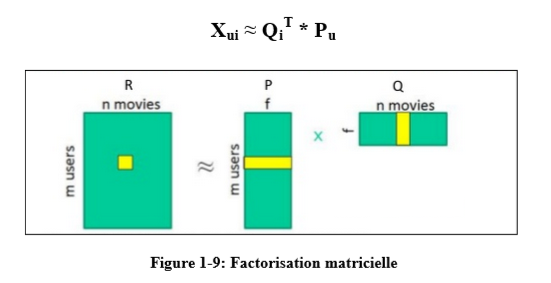
**La factorisation matricielle** est une classe d’algorithmes de **filtrage collaboratif** utilisée dans les systèmes de recommandation. Les algorithmes de factorisation matricielle fonctionnent en décomposant la matrice d’interaction Utilisateur-élément en le produit de deux matrices rectangulaires de dimensionnalité inferieure.



Cette famille de méthodes est largement connue grâce au défi du prix Netflix en raison de son efficacité, comme l’a rapporté Simon Funk dans son article de Blog 2006.

L’idée derrière la factorisation matricielle est de représenter les utilisateurs et les éléments (items) dans un espace latent de dimension inférieure.

La factorisation de matrice ou décomposition de matrice, consiste à décomposer une matrice en plusieurs autres matrices. Pour approximer la matrice observée, il suffira de calculer le produit de ces matrices. Ce produit aboutit à la matrice prédite.



Il existe plusieurs types de factorisation matricielles, en ce qui nous concerne nous nous attarderons sur la factorisation avec la méthode SVD qui comporte plusieurs approches :

1. **Funk SVD**

Il s’agit de l’algorithme original proposé par Simon Funk, qui consiste à factoriser la matrice notation Utilisateur-Item comme le produit de 02 matrices de dimension inférieure, la première a une colonne pour chaque utilisateur et la deuxième matrice a une colonne pour chaque item.

Les notes prédites peuvent être calculées comme suite **~R = HW** où ~R est de dimension Users\*Items et H est de dimension Users\*Facteurslatents et W est de dimension Facteurslatents\*Items

1. **SVD++**

Malgré Funk SVD soit capable de fournir des recommandations de bonne qualité, sa capacité à n’utiliser que des évaluations numériques explicites en tant qu’interactions utilisateur-item constitue une limitation. Car les systèmes de recommandation devraient exploiter les évaluations aussi bien explicites que implicites, SVD++ a donc été conçu pour prendre également en compte les évaluations implicites, il prend donc en compte le biais de l’utilisateur et de l’article.

La note prévue pour que l’utilisateur U attribue à l’item i sera calculée comme suit :

**~rui = HWui (comme avec funk Svd) + µ + bi + bu** où µ représente la moyenne de toutes les notes attribuées aux items, bi est la moyenne de toutes les notes attribuées à l’item i moins µ, bu est la moyenne de toutes les notes attribuées par l’utilisateur u moins µ.

**Exemple avec Funk SVD :**

