Recommandation - Rapport partiel

Ken Chanseau–Saint-Germain & Vincent Vidal

26 janvier 2014

Table des matières

Notation

On notera pour $p > 0$ un réel, X un vecteur et M une matrice quelconque :

$$\|X\|_p = \left(\sum_i |x_i|^p \right)^{\frac{1}{p}} \qquad \|M\|_p = \left(\sum_{i,j} |m_{i,j}|^p \right)^{\frac{1}{p}}$$

$$|||M|||_p = \sup_{\|x\|_p = 1} \|Mx\|_p$$

Et on posera $\|X\|_0$ le nombre de composantes non nulles de X.

1 Introduction

On se donne ici n personnes donnant des notes à m objets.
On notera $a_{i,j}$ la note de l'individu i sur l'objet j ainsi que $A = (a_{i,j})_{\substack{1 \leq i \leq n \\ 1 \leq j \leq m}}$ la matrice des notes.

On suppose ici que l'on n'a accès qu'à une matrice incomplète B obtenue en annulant certaines composantes de A. Le but est alors de trouver une bonne approximation $\widehat{A}$ de A à partir de B.

On prendra comme mesure d'approximation, l'erreur moyenne suivante :

$$\mathrm{RMSE}\left(\widehat{A}\right) = \sqrt{\frac{\sum_{i,j \in S} \left(a_{i,j} - \widehat{a}_{i,j}\right)^2}{\mathrm{Card}\,(S)}}$$

où S dont les éléments correspondent aux composantes oubliées dans la matrice A.

En pratique, les données que l'on utilisera pour tester les algorithmes correspondront à des matrices A incomplètes, et on utilisera une matrice B plus incomplète que A. La matrice d'approximation $\widehat{A}$ obtenue donne alors des coefficients dont on ne peut vérifier la précision. L'erreur moyenne ne sera alors calculée que sur les coefficients non nuls de A.

2 Approximation basique

2.1 Par moyenne

On suppose ici que la note d'un objet est de la forme $\widehat{a}_{i,j} = m_{\widehat{A}} + p_i + o_j$, avec $m_{\widehat{A}}$ la moyenne des valeurs de $\widehat{A}$, p_i la moyenne des notes recentrées qu'a donné la personne i et o_j la moyenne des notes recentrées qu'a obtenu l'objet j. C'est à dire :

$$m_{\widehat{A}} = \frac{\sum_{i,j} b_{i,j}}{\|B\|_0} \qquad p_i = \frac{\sum_j b_{i,j} - m_{\widehat{A}}}{\|B^{\mathrm{T}} \mathrm{e}_i\|_0} \qquad o_j = \frac{\sum_i b_{i,j} - m_{\widehat{A}}}{\|B \mathrm{e}_i\|_0}$$

Algorithm 1: Recommandations naïves par moyenne

Data: La matrice incomplète B.
Result: La matrice d'approximation $\widehat{A}$.
$m \longleftarrow \text{mean}(B)$;
for $i \in [\![1,\, n]\!]$ **do**
  $p[i] \longleftarrow 0$;
  $card \longleftarrow 0$;
  for $j \in [\![1,\, m]\!]$ **do**
    $p[i] \longleftarrow p[i] + B[i,\, j]$;
    if $b[i,\, j] \neq 0$ **then**
      $card \longleftarrow card + 1$;
  $p[i] \longleftarrow p[i]/card - m$;
On calcul de même les $o[j]$;
for $i \in [\![1,\, n]\!]$ **do**
  for $j \in [\![1,\, m]\!]$ **do**
    $\widehat{A}[i,\, j] \longleftarrow m + p[i] + o[j]$;
return $\widehat{A}$;

2.2 Par minimisation de problème convexe

De manière plus précise, on peut résoudre le problème suivant de minimisation convexe pénalisée sur les p_i et o_j pour trouver des p_i et o_j plus adaptés :

$$\mathcal{P}(p,\, o) = \sum_{i,j} \left(a_{i,j} - m_{\widehat{A}} + p_i + o_j\right)^2 + \lambda\left(\|o\|_1 + \|p\|_1\right)$$

Problème que l'on peut écrire :

$$\mathcal{P}(x) = \|Mx + b\|_2^2 + \lambda \|x\|_1 .$$

Il suffit alors de résoudre ce problème, pour un λ donné et de calculer l'approximation à partir des coefficients p_i et o_j obtenus.

On utilisera ici un algorithme de minimisation qui descend de l'algorithme de Douglas-Rachford [**?**], se rapprochant d'une méthode de descente de gradient. Cet algorithme prend un paramètre supplémentaire, μ, dont dépend la vitesse de convergence. La suite calculée par cet algorithme converge vers un minimum lorsque μ est suffisamment proche de 0.

Algorithm 2: Minimisation convexe

Data: x, λ, le nombre d'itération N et un paramètre μ.
Result: une meilleur approximation x.
for $k \in [\![1,\ N]\!]$ **do**
    $x \longleftarrow x - \mu.M^{\mathrm{T}}\left(Mx + b\right)$;
    `/* Seuillage doux sur chaque composante */`
    for $i \in [\![1,\ n]\!]$ **do**
        $x\left[i\right] \longleftarrow \mathrm{signe}\left(x\left[i\right]\right).\max\left(\ \left|x\left[i\right]\right| - \mu\lambda,\ 0\ \right)$;
return x;

3 Approximation par voisinage

Nous noterons par la suite P_i la i-ième colonne de B^{T}, représentant la personne i et O_j la j-ième colonne de B, représentant l'objet j.

3.1 Plus proches voisins

La méthode précédente suppose qu'il n'existe qu'un seul type de personne et d'objet : un objet est bien ou mauvais dans l'ensemble et une personne donne de manière générale des notes bonnes ou des notes mauvaises. Ceci apparaît comme très naïf.

On considère ici une méthode différente. Pour une personne i et un objet j dont l'information est manquante, nous allons chercher parmi les personnes ayant noté j les k qui se rapprochent le plus i. En notant S les indices de ces personnes, on peut alors prendre comme approximation la moyenne pondérée des notes :

$$\widehat{a}_{i_0,j} = \frac{\sum\limits_{i \in S} \left(s\left(i_0,\ i\right) \cdot b_{i,j}\right)}{\sum\limits_{i \in S} \left|s\left(i_0,\ i\right)\right|}.$$

Où $s(i,j)$ correspond à la distance entre deux personnes et représente leur ressemblance. Il reste à définir une distance pour sélectionner le plus proche voisin. Si on cherche le plus proche voisin d'un vecteur X de support S parmi un ensemble de vecteurs E, on pourrait considérer $E_{|S}$ les vecteurs de E projetés sur le support S et on prendrait alors le plus proche voisin pour la norme euclidienne sur $\mathbb{R}^{\mathrm{Card}(S)}$.

Nous prendrons ici la valeur du cosinus de l'angle entre les deux personnes :

$$s\left(i_1,\ i_2\right) = \frac{P_{i_1}^{\mathrm{T}} \cdot P_{i_2}}{\left\|P_{i_1}\right\|_2 \left\|P_{i_2}\right\|_2}.$$

Algorithm 3: Recommandation par k-plus proches voisins

Data: La matrice B et le nombre k de voisins pris en compte.
Result: La matrice approchée $\widehat{A}$.
calcul des $s(i_1, i_2)$;
for $i \in [\![1,\ n]\!]$ **do**
  for $j \in [\![1,\ m]\!]$ **do**
    $S \longleftarrow$ indices des k-plus proches voisins de i ayant noté j ;
    $\widehat{a}_{i,j} \longleftarrow \sum_{t \in S} (s\,(i,\ t) \cdot b_{t,j}) / \sum_{t \in S} |s\,(i,\ t)|$;
return $\widehat{A}$;

3.2 Implantation matricielle pour $k = n$

Dans le cas où $k = n$, les calculs peuvent s'effectuer plus efficacement à l'aide de calculs matriciels. On pourra considérer l'algorithme suivant :

Algorithm 4: Recommandation par voisinage

Data: La matrice B.
Result: La matrice approchée $\widehat{A}$.
$S \longleftarrow BB^{\mathrm{T}}$;
for $i \in [\![1,\ n]\!]$ **do**
  $norm\,[i] \longleftarrow \left\| B^{\mathrm{T}} \mathrm{e}_i \right\|_2$
for $i \in [\![1,\ n]\!]$ **do**
  $somme\,[i] \longleftarrow 0$;
  for $k \in [\![1,\ n]\!]$ **do**
    $S\,[i,\ k] \longleftarrow S\,[i,\ k]\,/\,(norm\,[i]\,.norm\,[k])$;
    $somme\,[i] \longleftarrow somme\,[i] + |S\,[i,\ k]|$;
$\widehat{A} \longleftarrow S.B$;
for $i \in [\![1,\ n]\!]$ **do**
  for $j \in [\![1,\ m]\!]$ **do**
    $\widehat{A}\,[i,\ j] \longleftarrow \widehat{A}\,[i,\ j]\,/somme\,[i]$;
return $\widehat{A}$;

4 Décomposition en valeurs singulières

Le but est ici de faire apparaître k modèles de personnes, dont toutes les autres seront des combinaisons linéaires.

Autrement dit, on souhaite faire apparaître la matrice $\widehat{A}$ de la manière suivante :

$$\widehat{A} = P \cdot O$$

avec $P \in \mathcal{M}_{n,\,k}(\mathbb{R})$ de colonnes libres que l'on peut prendre orthonormées et $O \in \mathcal{M}_{k,\,m}(\mathbb{R})$.

On peut facilement accéder à une telle décomposition à l'aide de la décomposition en valeurs singulières.

Algorithm 5: Recommandation par décomposition en valeurs singulières

Data: La matrice B et un entier k.
Result: La matrice approchée $\widehat{A}$.
$[U,\, S,\, V] \longleftarrow \mathrm{SVD}(B)$;
for $i \in [\![k,\, \mathrm{size}(S)]\!]$ **do**
  $S[i,\, i] \longleftarrow 0$;
$\widehat{A} \longleftarrow USV^{\mathrm{T}}$;
return $\widehat{A}$;