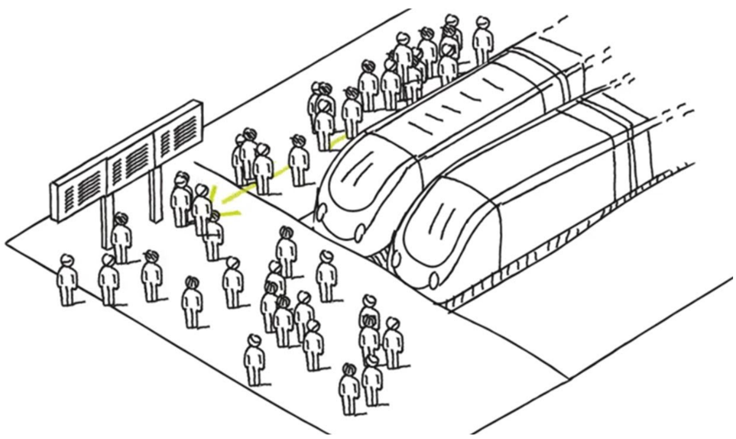
Mini-rapport sur l'estimation de la charge à la voiture

Formalisation du problème d'optimisation et implémentation en R



Lab' Mass Transit Academy
SNCF - Transilien
Auteur : Joshua Wolff
Supervision : Rémi Coulaud, Mathilde Vimont
Mai 2021

Table des matières

1 Objectif du document 1

2 Préambule 1

3 Méthode Hector 2

4 Méthode Weights 4

5 Solveur d'optimisation 5

1 Objectif du document

Ce document est un commentaire détaillé de la partie optimisation du code du projet *Optimove*. Nous poserons les problèmes d'optimisation à résoudre et les décrirons leurs formes matricielles. Une dernière partie est dédiée à une brève présentation du solveur d'optimisation employé dans ce projet.

2 Préambule

Précisons quelques notations qui seront d'usage dans ce document :

Notation	Domaine	Description
D	$\mathbb{N}^*$	Nombre de portes pour les trains observés
C	$\mathbb{N}^*$	Nombre de voitures pour les trains observés
K	$\mathbb{N}^*$	Nombre d'observations
a_{ik}	$\mathbb{N}$	Nombre de voyageurs descendus par la porte i pour l'observation k
A	$\mathbb{N}^{K\times D}$	Matrice des observations en descente dont le coefficient (k,i) est a_{ik}
b_{ik}	$\mathbb{N}$	Nombre de voyageurs montés par la porte i pour l'observation k
B	$\mathbb{N}^{K\times D}$	Matrice des observations en montée dont le coefficient (k,i) est b_{ik}
w_{ik}^{in}	$\mathbb{R}_+$	Charge de la voiture i de l'observation k en nombre de passagers en arrivée dans la gare considérée
w_{ik}^{out}	$\mathbb{R}_+$	Charge de la voiture i de l'observation k en nombre de passagers au départ de la gare considérée

TABLE 1 – Table des notations

Dans le cadre du formalisme *Hector*, les trajets sont agrégés. Ainsi a_{ik} désigne le nombre de voyageurs descendus par la porte i du train k sur l'ensemble de son trajet (de même pour b_{ik}). Dans le formalisme *Weights*, on se focalise à l'échelle d'une gare donc a_{ik} désigne le nombre de voyageurs descendus par la porte i du train k dans une gare fixée (de même pour b_{ik}).

Avant d'entrer dans le vif du sujet, intéressons nous à quelques aspects probabilistes différenciant les deux méthodes *Hector* et *Weights*. Considérons trois variables aléatoires X_1, X_2 et X_3. Elles représentent respectivement la porte de montée, la voiture d'installation et la porte de descente d'un voyageur quelconque. L'objectif du projet *Optimove* est de déterminer $\mathbb{P}(X_2|X_1)$ (*ie* la probabilité de s'installer dans une voiture sachant la voiture de montée dans le train). Dans le cas de la méthode *Weights*, le résultat du problème d'optimisation donne immédiatement $\mathbb{P}(X_2|X_1)$. Dans le cas de la méthode *Hector*, le résultat du problème d'optimisation donne $\mathbb{P}(X_3|X_1)$, il faut donc un peu ruser et remarquer que :

$$\mathbb{P}(X_2, X_3|X_1) = \frac{\mathbb{P}(X_2, X_3, X_1)}{\mathbb{P}(X_1)} = \frac{\mathbb{P}(X_2, X_3, X_1)}{\mathbb{P}(X_3, X_1)} \frac{\mathbb{P}(X_3, X_1)}{\mathbb{P}(X_1)} = \mathbb{P}(X_2|X_1, X_3).\mathbb{P}(X_3|X_1)$$

Remarquons par ailleurs que :

$$\mathbb{P}(X_2|X_1) = \mathbb{P}(\bigcup_{i=1}^{D}(\{X_3 = i\} \cap X_2)|X_1) = \sum_{i=1}^{D} \mathbb{P}(X_2, X_3 = i|X_1)$$

D'où, en utilisant la première remarque :

$$\mathbb{P}(X_2|X_1) = \sum_{i=1}^{D} \mathbb{P}(X_2|X_1, X_3 = i).\mathbb{P}(X_3 = i|X_1)$$

La distribution suivie par la variable aléatoire $X_2|X_1, X_3$ est fixée "à la main", tandis que la distribution suivie par $X_3|X_1$ est déterminée par la résolution du problème d'optimisation. Ainsi, le terme p_{ij} désigne $\mathbb{P}(X_3 = j|X_1 = i)$ dans la méthode *Hector* et p^b_{ij} (respectivement p^a_{ij}) désigne directement $\mathbb{P}(X_2 = i|X_1 = j)$ (respectivement $\mathbb{P}(X_2 = i|X_3 = j)$) dans la méthode *Weights*.

3 Méthode Hector

Le problème d'optimisation à résoudre dans la méthode *Hector* est le suivant :

$$\min_p \sum_k \sum_j (a_{jk} - \sum_i b_{ik} p_{ij})^2$$

Sous contraintes :

$$0 \leq p_{ij} \leq 1$$
$$\sum_j p_{ij} = 1$$

On souhaite mettre ce problème sous la forme matricielle suivante :

$$\min_p ||Mp - d||^2$$

Sous contraintes :

$$0 \leq p \leq 1 \text{ (inégalités valables pour chaque coefficient de } p\text{)}$$

$$Rp = 1 \text{ (égalité valable pour chaque coefficient de } Rp\text{)}$$

On introduit alors les notations suivantes :

$$p = \begin{pmatrix} p_{11} \\ p_{12} \\ \vdots \\ p_{DD} \end{pmatrix}, \, d = \begin{pmatrix} a_{11} \\ a_{21} \\ \vdots \\ a_{DK} \end{pmatrix}$$

On construit finalement la matrice M, où $\otimes$ désigne le produit de Kronecker et I_D l'identité de $\mathbb{R}^{D \times D}$:

$$M = B \otimes I_D \in \mathbb{R}^{DK \times D^2}$$

Assemblons désormais la matrice des contraintes R. On note $e_1, ..., e_D$ les vecteurs formant la base canonique de $\mathbb{R}^D$, R est définie par les colonnes comme suit :

$$R = [\underbrace{e_1, ..., e_1}_{D fois}, \underbrace{e_2, ..., e_2}_{D fois}, ..., \underbrace{e_D, ..., e_D}_{D fois}] \in \mathbb{R}^{D \times D^2}$$

Voici un exemple plus parlant en petite dimension avec $D = 2$ portes et $K = 3$ observations :

$$B = \begin{pmatrix} b_{11} & b_{21} \\ b_{12} & b_{22} \\ b_{13} & b_{23} \end{pmatrix} \text{ et } A = \begin{pmatrix} a_{11} & a_{21} \\ a_{12} & a_{22} \\ a_{13} & a_{23} \end{pmatrix}$$

$$p = \begin{pmatrix} p_{11} \\ p_{12} \\ p_{21} \\ p_{22} \end{pmatrix} \text{ et } d = \begin{pmatrix} a_{11} \\ a_{21} \\ a_{12} \\ a_{22} \\ a_{13} \\ a_{23} \end{pmatrix}$$

$$M = \begin{pmatrix} b_{11} & 0 & b_{21} & 0 \\ 0 & b_{11} & 0 & b_{21} \\ b_{12} & 0 & b_{22} & 0 \\ 0 & b_{12} & 0 & b_{22} \\ b_{13} & 0 & b_{23} & 0 \\ 0 & b_{13} & 0 & b_{23} \end{pmatrix} \text{ et } R = \begin{pmatrix} 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \end{pmatrix}$$

4 Méthode Weights

Le problème d'optimisation à résoudre dans la méthode *Weights* est le suivant :

$$\min_p \sum_k \sum_i (w_{ik}^{out} - w_{ik}^{in} - \sum_j (p_{ij}^b b_{jk} - p_{ij}^a a_{jk}))^2$$

Sous contraintes :

$$0 \leq p_{ij} \leq 1$$
$$\sum_j p_{ij} = 1$$

On souhaite mettre ce problème sous la forme matricielle suivante :

$$\min_p ||Mp - d||^2$$

Sous contraintes :

$$0 \leq p \leq 1 \text{ (inégalités valables pour chaque coefficient de } p\text{)}$$
$$Rp = 1 \text{ (égalité valable pour chaque coefficient de } Rp\text{)}$$

On introduit alors les notations suivantes :

$$p = \begin{pmatrix} p_{11}^b \\ p_{21}^b \\ \vdots \\ p_{CD}^b \\ p_{11}^a \\ p_{21}^a \\ \vdots \\ p_{CD}^a \end{pmatrix}, \ d = \begin{pmatrix} w_{11}^{out} - w_{11}^{in} \\ w_{21}^{out} - w_{21}^{in} \\ \vdots \\ w_{CK}^{out} - w_{CK}^{in} \end{pmatrix}$$

On construit finalement la matrice M, où $\otimes$ désigne le produit de Kronecker, $[.|.]$ la concaténation matricielle par les colonnes et I_C l'identité de $\mathbb{R}^{C \times C}$:

$$M = [B \otimes I_C | - A \otimes I_C] \in \mathbb{R}^{KC \times 2CD}$$

Assemblons désormais la matrice des contraintes R. On note $e_1, ..., e_D$ les vecteurs formant la base canonique de $\mathbb{R}^D$, R est définie comme suit :

$$R = \begin{pmatrix} \Delta & 0_{\mathbb{R}^{D \times CD}} \\ 0_{\mathbb{R}^{D \times CD}} & \Delta \end{pmatrix} \text{ avec } \Delta = [\underbrace{e_1, ..., e_1}_{C fois}, \underbrace{e_2, ..., e_2}_{C fois}, ..., \underbrace{e_D, ..., e_D}_{C fois}] \in \mathbb{R}^{D \times CD}$$

Voici un exemple plus parlant en petite dimension avec $C = 3$ voitures, $D = 2$ portes et $K = 3$ observations :

$$B = \begin{pmatrix} b_{11} & b_{21} \\ b_{12} & b_{22} \\ b_{13} & b_{23} \end{pmatrix} \text{ et } A = \begin{pmatrix} a_{11} & a_{21} \\ a_{12} & a_{22} \\ a_{13} & a_{23} \end{pmatrix}$$

$$W^{in} = \begin{pmatrix} w_{11}^{in} & w_{21}^{in} & w_{31}^{in} \\ w_{12}^{in} & w_{22}^{in} & w_{32}^{in} \\ w_{13}^{in} & w_{23}^{in} & w_{33}^{in} \end{pmatrix} \text{ et } W^{out} = \begin{pmatrix} w_{11}^{out} & w_{21}^{out} & w_{31}^{out} \\ w_{12}^{out} & w_{22}^{out} & w_{32}^{out} \\ w_{13}^{out} & w_{23}^{out} & w_{33}^{out} \end{pmatrix}$$

$$p = \begin{pmatrix} p_{11}^{b} \\ p_{21}^{b} \\ p_{31}^{b} \\ p_{12}^{b} \\ p_{22}^{b} \\ p_{32}^{b} \\ p_{11}^{a} \\ p_{21}^{a} \\ p_{31}^{a} \\ p_{12}^{a} \\ p_{22}^{a} \\ p_{32}^{a} \end{pmatrix} \text{ et } d = \begin{pmatrix} w_{11}^{out} - w_{11}^{in} \\ w_{21}^{out} - w_{21}^{in} \\ w_{31}^{out} - w_{31}^{in} \\ w_{12}^{out} - w_{12}^{in} \\ w_{22}^{out} - w_{22}^{in} \\ w_{32}^{out} - w_{32}^{in} \\ w_{13}^{out} - w_{13}^{in} \\ w_{23}^{out} - w_{23}^{in} \\ w_{33}^{out} - w_{33}^{in} \end{pmatrix}$$

$$M = \begin{pmatrix} b_{11} & 0 & 0 & b_{21} & 0 & 0 & -a_{11} & 0 & 0 & -a_{21} & 0 & 0 \\ 0 & b_{11} & 0 & 0 & b_{21} & 0 & 0 & -a_{11} & 0 & 0 & -a_{21} & 0 \\ 0 & 0 & b_{11} & 0 & 0 & b_{21} & 0 & 0 & -a_{11} & 0 & 0 & -a_{21} \\ b_{12} & 0 & 0 & b_{22} & 0 & 0 & -a_{12} & 0 & 0 & -a_{22} & 0 & 0 \\ 0 & b_{12} & 0 & 0 & b_{22} & 0 & 0 & -a_{12} & 0 & 0 & -a_{22} & 0 \\ 0 & 0 & b_{12} & 0 & 0 & b_{22} & 0 & 0 & -a_{12} & 0 & 0 & -a_{22} \\ b_{13} & 0 & 0 & b_{23} & 0 & 0 & -a_{13} & 0 & 0 & -a_{23} & 0 & 0 \\ 0 & b_{13} & 0 & 0 & b_{23} & 0 & 0 & -a_{13} & 0 & 0 & -a_{23} & 0 \\ 0 & 0 & b_{13} & 0 & 0 & b_{23} & 0 & 0 & -a_{13} & 0 & 0 & -a_{23} \end{pmatrix}$$

$$R = \begin{pmatrix} 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \end{pmatrix}$$

5 Solveur d'optimisation

Le solveur employé pour résoudre les problèmes d'optimisation posés précédemment est nommé *lsqlincon* (Linear Least-Squares Fitting with linear constraints) et est implémenté dans la librairie R *pracma*. Il permet de résoudre des problèmes de moindres carrés linéaires (ie que l'on peut mettre sous forme matricielle). On peut par ailleurs ajouter des contraintes linéaires d'égalité / inégalité sur les paramètres.

Compte tenu du formalisme précédemment développé, il suffit de passer en argument de *lsqlincon* les matrices M et R, le vecteur d et enfin les bornes inférieures et supérieures des paramètres (à savoir 0 et 1 dans notre cas). Le solveur retourne alors p sous la forme vectorielle décrite dans ce document.

On donne ci-dessous quelques exemples reprenant les mêmes notations que ce document :

```
library(pracma)

###########
# HECTOR #
###########

D <- 5 # 5 portes
K <- 100 # 100 observations

# Donnees aleatoires (D portes, K observations)

B <- matrix(rnorm(D*K), nrow = K, ncol = D) # montees
A <- matrix(rnorm(D*K), nrow = K, ncol = D) # descentes

# Mise sous forme matricielle

d <- as.vector(t(A))
M <- B %x% diag(D)
R <- diag(D)[,rep(1:D,each=D)]

# Resolution du probleme d'optimisation
# Il est d'usage de normaliser par la norme matricielle de M
# pour eviter les problemes d'overflow (ne change pas la solution)

res <- lsqlincon(C=M/norm(M,"F"), d=d/norm(M,"F"),
                 Aeq = R, beq = rep(1,D),
                 lb = 0, ub = 1)

# Mise en forme matricielle du resultat

mat_res <- t(matrix(res, nrow = D))

print(rowSums(mat_res))
[1] 1 1 1 1 1
```

```
############
# WEIGHTS #
############

C <- 7 # 7 voitures
D <- 5 # 5 portes
K <- 100 # 100 observations

# Donnees aleatoires

B <- matrix(rnorm(D*K), nrow = K, ncol = D) # montees
A <- matrix(rnorm(D*K), nrow = K, ncol = D) # descentes

Win <- matrix(rnorm(C*K), nrow = K, ncol = C)
Wout <- matrix(rnorm(C*K), nrow = K, ncol = C)

# Mise sous forme matricielle

d <- as.vector(t(Wout-Win))
M <- cbind(B %x% diag(C), -A %x% diag(C))
Delta <- diag(D)[, rep(1:D,each=C)]
Zero <- matrix(0, nrow=D, ncol=D*C)
R <- rbind(cbind(Delta,Zero),
           cbind(Zero,Delta))

# Resolution du probleme d'optimisation

res <- lsqlincon(C=M/norm(M,"F"), d=d/norm(M,"F"),
                 Aeq = R, beq = rep(1,2*D),
                 lb = 0, ub = 1)

# seul res_b nous interesse en pratique !
res_b <- res[1:(length(res) %/% 2)]
res_a <- res[(length(res) %/% 2 + 1):length(res)]

# Mise en forme matricielle du resultat

mat_res_b <- t(matrix(res_b, nrow = C))

print(rowSums(mat_res_b))
[1] 1 1 1 1 1
```