MAT 2742 - INTRODUCTION À L'ALGÈBRE LINÉAIRE APPLIQUÉE UNIVERSITÉ D'OTTAWA

ALISTAIR SAVAGE

Table des matières

Note aux étudiants : Si vous remarquez une erreur dans ces notes (même les petites fautes de frappe), s'il vous plaît informer le professeur. Cela aide lui, ainsi que vos collègues étudiants.

Remerciements : Ces notes sont basées sur les notes de Michael Newman.

Info de cours

Site web du cours : www.mathstat.uottawa.ca/∼asavag2/mat2742

Professeur : Alistair Savage

Bureau : KED 207G

Heures de bureau : Lundi 10h00–11h00, mardi 13h00-14h00

Manuel de cours : David Lay, *Algèbre linéaire : théorie, exercices et applications*, 3e édition, De Boeck, 2004. ISBN-10 : 2804144089.

Syllabus : Trouvé sur le site web et mise à jour pendant le semestre.

Devoirs : 5 devoirs, à remettre à peu près chaque 2 semaines. Les dates limites peuvent être trouvées sur le site web. La première est le 29 septembre à 10h00. Si'il vous plaît, lire les instructions qui se trouvent sur le site web.

Test de mi-session : 20 octobre (en classe).

Évaluation :
- Devoirs : 15%
- Test de mi-session : 25%
- Examen final : 60%

Dans le calcul de la partie devoirs de la note de chaque étudiant(e), la note de devoirs la plus basse sera remplacée par la note à l'examen final si c'est à l'avantage de l'étudiant(e). Les notes seront affichées sur le campus virtuel.

Leçon 1 : 8 septembre 2011

1. Valeurs et vecteurs propres

1.1. **Motivation : systèmes itératifs.** Considérons un exemple (simplifié !) d'une maladie : chaque mois une personne en bonne santé a une chance de 3% de tomber malade, et une personne malade a une chance de 5% de se guérir. Décrire la situation éventuelle.

Ce modèle divise la population en deux, donc on prend la population en mois k comme un vecteur de taille deux : $\mathbf{x}_k = \begin{bmatrix} a_k \\ b_k \end{bmatrix} \in \mathbb{R}^2$, où a_k représente le nombre de personnes en bonne santé et b_k le nombre de malades (en mois k). On a donc le système d'équations suivant qui décrit la situation :

$$\begin{aligned} a_{k+1} &= 0,97a_k + 0,05b_k, \\ b_{k+1} &= 0,03a_k + 0,95b_k. \end{aligned}$$

En forme matricielle, on a

$$\mathbf{x}_{k+1} = A\mathbf{x}_k, \qquad \text{où} \qquad A = \begin{bmatrix} 0,97 & 0,05 \\ 0,03 & 0,95 \end{bmatrix}.$$

Que veut dire "la situation éventuelle" ? C'est que éventuellement la situation stabilise et on a donc $\mathbf{x}_{k+1} \approx \mathbf{x}_k$. Ceci donne que

$$\mathbf{x}_k \approx \mathbf{x}_{k+1} = A\mathbf{x}_k.$$

Donc, on veut résoudre l'équation matricielle $\mathbf{x} = A\mathbf{x}$. C'est une relation de valeur et vecteur propre.

L'exemple est simple mais les idées se généralisent.

1.2. **Définitions.**

Définition 1.1 (valeur et vecteur propre)**.** Soit A une matrice (carré) de taille $n \times n$. Soit $\mathbf{x} \neq \mathbf{0}$ un vecteur de taille n et λ un chiffre (nombre réel ou complexe). Si

$$A\mathbf{x} = \lambda\mathbf{x}$$

alors on dit que λ et un *valeur propre* (en anglais, *eigenvalue*) de A et que $\mathbf{x}$ est un *vecteur propre* (en anglais, *eigenvector*) correspondant.

Remarque. C'est important que $\mathbf{x}$ soit non-nul. Si $\mathbf{x} = \mathbf{0}$, alors $A\mathbf{0} = \mathbf{0} = \lambda\mathbf{0}$ pour *toute* valeur λ, qui n'est pas une situation intéressante.

Étant donné une matrice et un vecteur, on peut déterminer directement si c'est un vecteur propre en multipliant.

Exemple 1.2. Est-ce que $\mathbf{x} = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$ est vecteur propre de $A = \begin{bmatrix} 2 & 1 \\ 3 & 4 \end{bmatrix}$?

On calcul

$$A\mathbf{x} = \begin{bmatrix} 2 & 1 \\ 3 & 4 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 3 \\ 7 \end{bmatrix}.$$

Si $A\mathbf{x} = \lambda\mathbf{x}$ on voit que $3 = \lambda 1$ (qui implique que $\lambda = 3$) et $7 = \lambda 1$ (qui implique que $\lambda = 7$), ce qui est impossible ! Donc $\mathbf{x}$ n'est pas vecteur propre de A.

Exercice 1.3. Déterminer si les vecteurs suivants sont vecteurs propres de la matrice A d'Exemple 1.2 :

$$\begin{bmatrix} 3 \\ 4 \end{bmatrix}, \quad \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \quad \begin{bmatrix} 1 \\ -1 \end{bmatrix}, \quad \begin{bmatrix} -5 \\ 5 \end{bmatrix}.$$

Si oui, donner la valeur propre correspondante.

On peut aussi déterminer si une valeur est valeur propre d'une matrice.

Exemple 1.4. Est-ce que $\lambda = 2$ est valeur propre de $A = \begin{bmatrix} 2 & 1 \\ 3 & 4 \end{bmatrix}$?

Si oui, ce devrait être possible de trouver un vecteur propre correspondant. On note que

$$A\mathbf{x} = \lambda\mathbf{x} \iff A\mathbf{x} - \lambda\mathbf{x} = \mathbf{0} \iff (A - \lambda)\mathbf{x} = \mathbf{0}.$$

C'est une système d'équations linéaires (en forme matricielle). On peut résoudre à l'aide de la méthode de Gauss-Jordan (la réduction par rapport aux lignes) ce que vous avez vu dans MAT 1702 ou 1741.

$$\left[\begin{array}{cc|c} 0 & 1 & 0 \\ 3 & 2 & 0 \end{array}\right] \xrightarrow{R_1 \leftrightarrow R_2} \left[\begin{array}{cc|c} 3 & 2 & 0 \\ 0 & 1 & 0 \end{array}\right] \xrightarrow{R_1 \to \frac{1}{3}R_1} \left[\begin{array}{cc|c} 1 & \frac{2}{3} & 0 \\ 0 & 1 & 0 \end{array}\right] \xrightarrow{R_1 \to R_1 - \frac{2}{3}R_2} \left[\begin{array}{cc|c} 1 & 0 & 0 \\ 0 & 1 & 0 \end{array}\right]$$

(La prochaine fois, on ne va pas écrire la dernière colonne de zéros.) Si on maintenant change à la forme d'équation, on a

$$\begin{aligned} x_1 &= 0, \\ x_2 &= 0. \end{aligned}$$

Donc la seule solution est que $\mathbf{x} = \mathbf{0}$. Mais un vecteur propre n'est jamais zéro. Donc $\lambda = 2$ n'est pas valeur propre.

Exercice 1.5. Est-ce que $\lambda = 1$ est valeur propre de la matrice A dans Exemple 1.4 ?

1.3. **Méthode générale.** Chaque vecteur propre correspond à une seule valeur propre, mais chaque valeur propre possède plusieurs vecteurs propres (vous devriez avoir vu ceci dans Exercice 1.3).

Définition 1.6 (noyau, espace nul)**.** Si B est une matrice, alors

$$\ker B := \{\mathbf{x} \mid B\mathbf{x} = \mathbf{0}\}$$

est le *noyau*, ou l'*espace nul* (en anglais, *kernel*), de B.

Puisque

$$A\mathbf{x} = \lambda\mathbf{x} \iff (A - \lambda I)\mathbf{x} = \mathbf{0},$$

les vecteurs propres de A correspondant à λ sont les éléments non-nuls de $\ker(A - \lambda I)$. À l'Exemple 1.4 on n'a pas pu trouver des solutions non-nulles : autrement dit on avait trouvé que $\ker(A - 2I) = \{\mathbf{0}\}$. Pour que 2 soit valeur propre, il aurait fallu avoir $\ker(A - 2I) \neq \{\mathbf{0}\}$.

On se rappelle un théorème important en algèbre linéaire.

Théorème 1.7. *Si B est une matrice, alors*

$$\ker B \neq \{\mathbf{0}\} \iff \det B = 0.$$

Le déterminant d'une matrice se calcul par cofacteurs (voir la section 3.1 du manuel de cours). On fera donc le calcul de $\det(A - \lambda I)$. C'est le *polynôme caractéristique* de la matrice A. Les racines de ce polynôme sont exactement les valeurs propres.

Exemple 1.8. On calcule toutes les valeurs propres et vecteurs propres de la matrice

$$A = \begin{bmatrix} 0 & -1 & 4 \\ -2 & 1 & 4 \\ 0 & 0 & 2 \end{bmatrix}.$$

On fera une expansion par la deuxième rangée.

$$\begin{aligned} \det(A - \lambda I) &= \det \begin{bmatrix} 0-\lambda & -1 & 4 \\ -2 & 1-\lambda & 4 \\ 0 & 0 & 2-\lambda \end{bmatrix} \\ &= (2-\lambda)\det \begin{bmatrix} -\lambda & -1 \\ -2 & 1-\lambda \end{bmatrix} \\ &= (2-\lambda)\big((-\lambda)(1-\lambda) - (-1)(-2)\big) \\ &= (2-\lambda)(\lambda^2 - \lambda - 2) \\ &= -(\lambda-2)(\lambda-2)(\lambda+1). \end{aligned}$$

Donc les valeurs propres sont $\lambda = -1$ de multiplicité 1 et $\lambda = 2$ de multiplicité 2.

Note que ce n'est pas utile de tout multiplier toute de suite. C'est meilleur de tenter de sortir un facteur commun (le $(\lambda - 2)$ ici) avant de multiplier.

On aurait pu choisir n'importe quelle rangée ou colonne. En général, on choisit la rangée ou colonne avec le plus grand nombre de zéros.

Exercice 1.9. Calculer le déterminant ci-haut en faisant une expansion par la première colonne, ainsi que la deuxième rangée. Comparer au précédant.

Sachant les valeurs propres, on calcule les vecteurs propres. C'est la même idée qu'à l'Exemple 1.4, sauf que maintenant on sait déjà que c'est une bonne valeur propre.

Pour $\lambda = 2$, on calcule (cette fois, on n'écrit pas la dernière colonne de zéros)

$$\begin{bmatrix} -2 & -1 & 4 \\ -2 & -1 & 4 \\ 0 & 0 & 0 \end{bmatrix} \xrightarrow{R_2 \to R_2 - R_1} \begin{bmatrix} -2 & -1 & 4 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \xrightarrow{R_1 \to -\frac{1}{2}R_1} \begin{bmatrix} 1 & \frac{1}{2} & -2 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}.$$

Il y a un pivot, donc deux paramètres (correspondant aux deux colonnes sans pivot). La solution générale s'écrit comme

$$\begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = x_2 \begin{bmatrix} -\frac{1}{2} \\ 1 \\ 0 \end{bmatrix} + x_3 \begin{bmatrix} 2 \\ 0 \\ 1 \end{bmatrix}.$$

Alternativement on a une base pour l'*espace propre* :

$$\left\{ \begin{bmatrix} -\frac{1}{2} \\ 1 \\ 0 \end{bmatrix}, \begin{bmatrix} 2 \\ 0 \\ 1 \end{bmatrix} \right\}.$$

De manière similaire on traite $\lambda = -1$.

$$\begin{bmatrix} 1 & -1 & 4 \\ -2 & 2 & 4 \\ 0 & 0 & 3 \end{bmatrix} \to \cdots \to \begin{bmatrix} 1 & -1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix}$$

Solution et base :

$$\begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = x_2 \begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix}, \qquad \left\{ \begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix} \right\}.$$

Leçon 2 : 12 septembre 2011

1.4. **Diagonalisabilité.** Rappelez-vous que la *dimension* d'un espace vectoriel est exactement le nombre de vecteurs dans une base pour cette espace. Notez que la base pour l'espace propre de A correspondant à la valeur propre λ est une base pour le noyau de la matrice $A - \lambda I$. La *multiplicité* d'une valeur propre est le nombre de fois qu'elle apparaît comme racine dans le polynôme caractéristique. Chaque multiplicité est toujours égale ou inférieure à 1. Si A est $n \times n$, alors le somme des multiplicités est égal à n.

Théorème 1.10. *La dimension de chaque espace propre est toujours au plus la multiplicité de la valeur propre correspondante.*

Dans Exemple 1.8, les dimensions sont chacune égale aux multiplicités correspondantes. C'est un cas à la fois spécial et très utile (on verra bientôt une application).

Théorème 1.11 (Diagonalisation)**.** *Soit A une matrice telle que la dimension de chaque espace propre est égale à la multiplicité correspondante. Alors on a $A = PDP^{-1}$, où les colonnes de la matrice P sont les éléments des bases pour chaque espace propre, et D est une matrice diagonale formée des valeurs propres (dans l'ordre qui correspond à l'ordre des colonnes de P). (Une partie du théorème et que P est inversible.)*

Si une matrice A peut s'écrire comme $A = PDP^{-1}$ pour quelque matrice diagonale D et matrice inversible P, la matrice A est dite *diagonalisable*.

Exemple 1.12. Pour la matrice A d'Exemple 1.8 on écrit directement que

$$A = PDP^{-1} = \begin{bmatrix} 1 & -\frac{1}{2} & 2 \\ 1 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} -1 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 2 \end{bmatrix} \begin{bmatrix} 1 & -\frac{1}{2} & 2 \\ 1 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}^{-1}.$$

Observons que pour une matrice diagonalisable, on a

$$A^2 = (PDP^{-1})^2 = PDP^{-1}PDP^{-1} = PDIDP = PD^2P.$$

De la même façon on a

$$A^k = PD^kP^{-1} \quad \text{pour tout } k.$$

Puisque c'est très facile de calculer les puissances d'une matrice diagonale, cette observation est très utile pour calculer les puissances d'une matrice diagonalisable.

Exemple 1.13. Pour la matrice A précédente, calculer A^{10}.

On connaît déjà la décomposition $A = PDP^{-1}$, donc on écrit

$$\begin{aligned} A^{10} = PD^{10}P^{-1} &= \begin{bmatrix} 1 & -\frac{1}{2} & 2 \\ 1 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} -1 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 2 \end{bmatrix}^{10} \begin{bmatrix} 1 & -\frac{1}{2} & 2 \\ 1 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}^{-1} \\ &= \begin{bmatrix} 1 & -\frac{1}{2} & 2 \\ 1 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} -1 & 0 & 0 \\ 0 & 1024 & 0 \\ 0 & 0 & 1024 \end{bmatrix} \begin{bmatrix} 1 & -\frac{1}{2} & 2 \\ 1 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}^{-1}. \end{aligned}$$

Afin de finir le calcul, il faudrait calculer une inverse matricielle (quelque chose que vous avez appris dans MAT 1741 ou 1702) et deux multiplications matricielles, au lieu de neuf multiplications matricielles.

2. Systèmes dynamiques

2.1. **Un exemple : Une population de chouettes.** Considérons un exemple simple qui modèle une population de chouettes en termes d'adultes et de jeunes. On notera a_k le nombre d'adultes au temps k et b_k ("bébés") le nombre de jeunes au temps k. Supposons qu'à chaque année la moitié des adultes survivent et le quart des jeunes survivent (pour devenir adulte !). De plus, chaque adulte produit en moyenne deux jeunes chaque année. On a donc les équations suivantes :

$$\begin{aligned} a_{k+1} &= 0,5a_k + 0,25b_k \\ b_{k+1} &= 2a_k \end{aligned}$$

On a donc une équation matricielle :

$$\mathbf{x}_k = \begin{bmatrix} a_k \\ b_k \end{bmatrix}, \qquad A = \begin{bmatrix} 0,5 & 0,25 \\ 2 & 0 \end{bmatrix}, \qquad \mathbf{x}_{k+1} = A\mathbf{x}_k.$$

Ceci permet de calculer systématiquement les vecteurs de populations, sachant la population initiale (qu'on notera typiquement comme $\mathbf{x}_0$). À titre d'exemple si $\mathbf{x}_0 = \left[\begin{smallmatrix}100\\40\end{smallmatrix}\right]$ on calcule :

$$\mathbf{x}_1 = A\mathbf{x}_0 = \begin{bmatrix}60\\200\end{bmatrix}, \quad \mathbf{x}_2 = A\mathbf{x}_1 = \begin{bmatrix}80\\120\end{bmatrix}, \quad \mathbf{x}_3 = A\mathbf{x}_2 = \begin{bmatrix}70\\160\end{bmatrix}, \quad \mathbf{x}_4 = A\mathbf{x}_3 = \begin{bmatrix}75\\140\end{bmatrix}.$$

On observe deux choses.

- Il semble que $\mathbf{x}_k$ converge, c'est-à-dire, que lorsque k devient grand les populations se stabilisent (c'est plus évident si on continuer à calculer).
- Il semble avoir une oscillation.

On découvra que la théorie des valeurs et vecteurs propres est très pertinent ici.

Exercice 2.1. Montrer que $\mathbf{v}_1 = \begin{bmatrix}0,5\\1\end{bmatrix}$, $\mathbf{v}_2 = \begin{bmatrix}-0,25\\1\end{bmatrix}$ sont vecteurs propres de A avec valeurs propres $\lambda_1 = 1$, $\lambda_2 = -0,5$ (respectivement). Quelles sont les multiplicités ?

Après avoir complété Exercice 2.1, on sait que A est diagonalisable, et $A = PDP^{-1}$ où

$$P = \begin{bmatrix}0,5 & -0,25\\1 & 1\end{bmatrix}, \quad D = \begin{bmatrix}1 & 0\\0 & -0,5\end{bmatrix}.$$

On observe que les populations au temps k peuvent se calculer avec une puissance matricielle : $\mathbf{x}_k = A^k\mathbf{x}_0$. En appliquant la théorie de diagonalisation on obtient

$$\begin{aligned}\mathbf{x}_k = A^k\mathbf{x}_0 = (PDP^{-1})^k\mathbf{x}_0 = PD^kP^{-1}\mathbf{x}_0 &= \begin{bmatrix}0,5 & -0,25\\1 & 1\end{bmatrix}\begin{bmatrix}1 & 0\\0 & -0,5\end{bmatrix}^k\begin{bmatrix}0,5 & -0,25\\1 & 1\end{bmatrix}^{-1}\mathbf{x}_0\\ &= \begin{bmatrix}0,5 & -0,25\\1 & 1\end{bmatrix}\begin{bmatrix}1^k & 0\\0 & (-0,5)^k\end{bmatrix}\begin{bmatrix}0,5 & -0,25\\1 & 1\end{bmatrix}^{-1}\mathbf{x}_0\end{aligned}$$

□

Comme $k \to \infty$, on sait que $1^k \to 1$ et $(-0,5)^k \to 0$. Alors, si k est "grand", alors on a

$$\begin{aligned}\mathbf{x}_k &\approx \begin{bmatrix}0,5 & -0,25\\1 & 1\end{bmatrix}\begin{bmatrix}1 & 0\\0 & 0\end{bmatrix}\begin{bmatrix}0,5 & -0,25\\1 & 1\end{bmatrix}^{-1}\mathbf{x}_0\\ &= \begin{bmatrix}0,5 & -0,25\\1 & 1\end{bmatrix}\begin{bmatrix}1 & 0\\0 & 0\end{bmatrix}\begin{bmatrix}\alpha\\\beta\end{bmatrix}\\ &= \begin{bmatrix}0,5 & -0,25\\1 & 1\end{bmatrix}\begin{bmatrix}\alpha\\0\end{bmatrix}\\ &= \begin{bmatrix}0,5\alpha\\\alpha\end{bmatrix}\\ &= \alpha\begin{bmatrix}0,5\\1\end{bmatrix}.\end{aligned}$$

On pourrait calculer le vecteur $\left[\begin{smallmatrix}\alpha\\ \beta\end{smallmatrix}\right]$ comme $P^{-1}\mathbf{x}_0$. Mais l'important est que si k est "grand" alors $\mathbf{x}_k$ est une constante (α) fois un vecteur propre. Ceci explique la convergence numérique.

Puisque A est diagonalisable, l'ensemble $\{\mathbf{v}_1, \mathbf{v}_2\}$ forme une base pour tout l'espace $\mathbb{R}^2$. Autrement dit on peut exprimer n'importe quel vecteur en termes de cette base. Par exemple, on sait que

$$\mathbf{x}_0 = c_1\mathbf{v}_1 + c_2\mathbf{v}_2 \tag{2.1}$$

pour quelques $c_1, c_2 \in \mathbb{R}$.

Exercice 2.2. Montrer que $c_1 = \alpha$ et $c_2 = \beta$.

Prenons par exemple que $\mathbf{x}_0 = \left[\begin{smallmatrix}100\\ 40\end{smallmatrix}\right]$. Équation (2.1) devient

$$\begin{bmatrix}100\\ 40\end{bmatrix} = c_1\begin{bmatrix}0,5\\ 1\end{bmatrix} + c_2\begin{bmatrix}-0,25\\ 1\end{bmatrix} \qquad \begin{cases}0,5c_1 - 0,25c_2 = 100\\ c_1 + c_2 = 40\end{cases}$$

On solutionne ce système à l'aide de la réduction de Gauss-Jordan.

$$\left[\begin{array}{cc|c} 0,5 & -0,25 & 100\\ 1 & 1 & 40\end{array}\right] \to \cdots \to \left[\begin{array}{cc|c} 1 & 0 & 440/3\\ 0 & 1 & -320/3\end{array}\right]$$

On obtient alors $\mathbf{x}_0 = (440/3)\mathbf{v}_1 - (320/3)\mathbf{v}_2$. (Si vous avez le moindre difficulté à comprendre d'où vient cette équation, ce serait le moment de faire une révision.)

On se rappelle la relation fondamentale de vecteur et valeur propre. Si $\mathbf{v}$ est vecteur propre de A correspondant à la valeur propre λ alors $A\mathbf{v} = \lambda\mathbf{v}$. Donc

$$\begin{aligned}\mathbf{x}_k = A^k\mathbf{x}_0 &= A^k(c_1\mathbf{v}_1 + c_2\mathbf{v}_2)\\ &= c_1A^k\mathbf{v}_1 + c_2A^k\mathbf{v}_2\\ &= c_1(\lambda_1)^k\mathbf{v}_1 + c_2(\lambda_2)^k\mathbf{v}_2.\end{aligned} \tag{2.2}$$

Dans notre cas on voit que

$$\begin{aligned}\begin{bmatrix}a_k\\ b_k\end{bmatrix} &= 440/3\begin{bmatrix}0,5\\ 1\end{bmatrix} + (-0,5)^k\begin{bmatrix}-0,25\\ 1\end{bmatrix}\\ &\approx 440/3\begin{bmatrix}0,5\\ 1\end{bmatrix} \qquad \text{(si } k \text{ est grand).}\end{aligned}$$

On voit encore la convergence, mais aussi l'oscillation. Le terme dominant, correspondant à la valeur propre dominante, donne la tendance éventuelle. La deuxième valeur propre étant inférieure en valeur absolue disparaît éventuellement. L'oscillation se voit dans le fait que la deuxième valeur propre est négative.

Leçon 3 : 15 septembre 2011

L'exemple ci-haut est un *système dynamique (discret)*, c'est-à-dire, une système ou l'état $\mathbf{x}_k$ au temps k du système est décrit par une équation de récurrence de la forme $\mathbf{x}_{k+1} = A\mathbf{x}_k$ où A est une matrice (la *matrice de transition*). Pour décrire la situation éventuelle, il faut trouver les valeurs et vecteur propres.

2.2. **Proie-prédateur.** Le même format peut servir de modéliser la prédation entre deux espèces, e.g., chouettes et souris. Ici C_k mesure le nombre de chouettes et S_k mesure le nombre de rats en milliers.

$$C_{k+1} = 0,5C_k + 0,4S_k$$
$$S_{k+1} = -0,104C_k + 1,1S_k$$

Le $0,5$ et le $1,1$ indique la croissance des espèces en isolation. Les chouettes disparaissent sans nourriture tandis que les souris augmentent sans prédateurs. Les deux autres chiffres mesurent la conséquence de la prédation : positif pour les chouettes et négatif pour les souris. On a donc la *matrice de transition* (ou *matrice d'étape*)

$$A = \begin{bmatrix} 0,5 & 0,4 \\ -0,104 & 1,1 \end{bmatrix}.$$

Un calcul routine donne les valeurs et vecteurs propres.

Exercice 2.3. Déterminer les valeurs et vecteurs propres de la matrice de transition.

On obtient, comme on a fait à l'équation (2.2), une formule pour le vecteur des populations :

$$\mathbf{x}_k = \begin{bmatrix} C_k \\ S_k \end{bmatrix} = c_1(1,02)^k \begin{bmatrix} 10 \\ 13 \end{bmatrix} + c_2(0,58)^k \begin{bmatrix} 5 \\ 1 \end{bmatrix}. \tag{2.3}$$

Les chiffres c_1 et c_2 dépendent des populations initiales. Plus précisément, sachant $\mathbf{x}_0$ on les obtient en solutionnant le système :

$$\mathbf{x}_0 = c_1\mathbf{v}_1 + c_2\mathbf{v}_2 = c_1 \begin{bmatrix} 10 \\ 13 \end{bmatrix} + c_2 \begin{bmatrix} 5 \\ 1 \end{bmatrix}.$$

Par contre, sans savoir les populations initiales, on peut décrire la tendance éventuelle.

Si $c_1 \neq 0$ dans l'équation (2.3), alors le premier terme domine éventuellement, car le $(0,58)^k$ tend vers zéro. Donc dans ce cas, on aura éventuellement une croissance des deux espèces, avec un rapport d'environ 10 chouettes pour 13 souris.

Si $c_1 = 0$ alors le premier terme est zéro. La tendance éventuelle sera donné par le deuxième terme, qui diminue vers zéro à cause du $(0,58)^k$. Donc les deux espèces disparaîtront. Mais ce cas est vraiment improbable. Avoir $c_1 = 0$ veut dire que le vecteur des populations initiales est un multiple de $\mathbf{v}_2$: donc exactement 5 chouettes pour 1 souris.

2.3. **Trajectoire.** On voit que la tendance éventuelle qualitative dépend des valeurs propres. Posons λ_1 la plus grande valeur propre (en valeur absolue). Présumons que la multiplicité de λ_1 est 1 et que tout autre valeur propre est strictement inférieure à λ_1 en valeur absolue. Alors en générale on aura

$$\mathbf{x}_k = c_1(\lambda_1)^k\mathbf{v}_1 + c_1(\lambda_2)^k\mathbf{v}_2 + \cdots + c_n(\lambda_n)^k\mathbf{v}_n \tag{2.4}$$

$$\approx c_1(\lambda_1)^k\mathbf{v}_1 \qquad \text{(pour } k \text{ grand)}. \tag{2.5}$$

On a une formule exacte et une approximation simple. L'approximation est valide car en élevant les valeurs propres à des grandes puissances, le premier domine.

Exercice 2.4. Que change dans l'équation (2.5) si la multiplicité de λ_1 est plus que 1 ? Qu'arrive s'il existe des autres valeurs propres qui sont égales en valeur absolue à λ_1 (par exemple, $\lambda_1 = 1.2$ et $\lambda_2 = -1.2$) ? □

Le *trajectoire* est la suite des vecteurs $\{\mathbf{x}_0, \mathbf{x}_1, \mathbf{x}_2, \ldots\}$, où chaque vecteur est obtenu en multipliant son prédécesseur par la matrice de transition A. En deux dimensions (c'est-à-dire un modèle qui décrit deux populations) on peut représenter cette suite graphiquement. (Voir aussi la section 5.6 du manuel de cours). On verra ceci en détail au tableau en classe.

Exemple 2.5. Soit un système dynamique modélisant deux populations, avec valeurs propres $\lambda_1 > \lambda_2 > 1$.

La valeur propre dominante est plus grande que 1, donc les populations augmentent toujours. L'origine est un *point de répulsion*. Les deux valeurs propres sont positives, donc il n'y aura aucune oscillation. □

Exemple 2.6. Soit un système dynamique modélisant deux populations, avec valeurs propres $\lambda_1 = 1$ et $-1 < \lambda_2 < 0$.

La valeur propre dominante est égale à un, donc les populations se stabiliseront éventuellement. La deuxième valeur propre est négative, qui donnera une oscillation, mais une oscillation qui diminuera car elle est inférieure à un en valeur absolue. □

Exemple 2.7. Soit un système dynamique modélisant deux populations, avec valeurs propres $0 < \lambda_2 < \lambda_1 < 1$.

Toutes les valeurs propres sont inférieures à un, donc les populations diminueront vers zéro peut importe les populations initiales. L'origine est un *point d'attraction*. □

Exemple 2.8. Soit un système dynamique modélisant deux populations, avec valeurs propres $0 < \lambda_2 < 1 < \lambda_1$.

Dans la direction du vecteur propre $\mathbf{v}_1$ l'origine paraît être un point de répulsion, car $\lambda_1 > 1$. Dans la direction du vecteur propre $\mathbf{v}_2$ l'origine paraît être un point d'attraction, car $\lambda_2 < 1$. On dit que l'origine est un *point de selle*. Note que dans n'importe quelle autre direction, l'origine ressemble plutôt à un point de répulsion, car c'est la plus grande valeur propre qui domine. Donc un point de selle se comporte plus comme point de répulsion. □

2.4. **Stabilité en deux dimensions.** Un modèle général de proie-prédateur en deux dimensions possède une matrice de transition

$$A = \begin{bmatrix} r & q \\ -p & s \end{bmatrix}$$

avec $0 < r < 1$ et $s > 1$ (car en absence de proie le prédateur disparaît et en absence de prédateur le proie augmente) et $p, q > 1$. On trouve que les valeurs propres de A sont

$$\lambda = \frac{(r+s) \pm \sqrt{(r+s)^2 - 4(rs+pq)}}{2}$$

(exercice).

Afin d'avoir la stabilité éventuelle, on voudra que $\lambda_1 = 1$ et $|\lambda_2| < 1$. En posant $\lambda_1 = 1$, on obtient

$$\begin{aligned} \frac{(r+s) + \sqrt{(r+s)^2 - 4(rs+pq)}}{2} &= 1 \\ r+s &= 2 - \sqrt{(r+s)^2 - 4(rs+pq)} \\ r+s &\leq 2 \end{aligned} \tag{2.6}$$

et aussi

$$\begin{aligned} r+s &= 2 - \sqrt{(r+s)^2 - 4(rs+pq)} \\ (r+s-2)^2 &= (r+s)^2 - 4(rs+pq) \\ -(r+s)+1 &= -rs - pq \\ pq &= (1-r)(s-1). \end{aligned} \tag{2.7}$$

Note que l'équation (2.7) implique que le discriminant est positif :

$$\begin{aligned} (r+s)^2 - 4(rs+pq) &= (r+s)^2 - 4(rs + (1-r)(s-1)) \\ &= (r+s)^2 - 4(r+s) + 4 \\ &= (r+s-2)^2 \geq 0. \end{aligned}$$

Donc un système proie-prédateur donnera une stabilité éventuelle si

$$\begin{aligned} &0 < r < 1, \quad s > 1, \quad p > 0, \quad q > 0, \\ &r + s \leq 2, \\ &pq = (1-r)(s-1). \end{aligned}$$

Ce genre d'analyse devient plus difficile en $n > 2$ dimensions.

Leçon 4 : 19 septembre 2011

3. Chaînes de Markov

3.1. **Exemple de motivation.** Une *chaîne de Markov* est un cas particulier d'un système dynamique où on considère plusieurs états d'une seule population. La distinction essentielle est que la population totale est constante. Comme exemple on peut considérer une modèle d'une maladie contagieuse (voir la section 1.1). Le modèle le plus simple considère deux cas seulement : malade et bonne santé.

Exemple 3.1. Chaque mois une personne en bonne santé a une chance de 3% de tomber malade, et une personne malade a une chance de 5% de se guérir. Au départ, personne n'est malade.

On a une matrice de transition (ou matrice d'étape) de $A = \begin{bmatrix} 0,97 & 0,05 \\ 0,03 & 0,95 \end{bmatrix}$. Les colonnes correspondent au deux états au présent et les rangées correspondent au deux états à l'étape suivante.

Puisque la population totale (malade et non-malades) est constante, on considère souvent le vecteur des proportions au lieu du vecteur des populations. Donc on a $\mathbf{x}_0 = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$ et $\mathbf{x}_{k+1} = A\mathbf{x}_k$ comme avant. On dit souvent *vecteur d'état* pour $\mathbf{x}_k$. □

Note que c'est un modèle à 2 états (malade et non-malade), qui mène à une matrice de taille 2×2 et des vecteurs d'état de taille 2. En général, ayant n états on a une matrice de taille $n \times n$ et des vecteurs d'état de taille n.

3.2. **Matrice stochastique.** On observe que la somme de chaque colonne de la matrice A ci-haut est 1. Ce n'est pas par hasard, et on a même utilisé cette observation afin de trouver les composantes non-diagonales de A. On pourrait même l'adpoter comme définition.

Définition 3.2 (Matrice stochastique)**.** Une matrice A est *stochastique* si les composantes de A sont toutes non-négatives et que la somme de chaque colonne est 1. Autrement dit on exige que $A_{ij} \geq 0$ pour tous i, j, et que $\sum_i A_{ij} = 1$ pour tout j.

La matrice de transition d'une chaîne de Markov est stochastique (et toute matrice stochastique représente une chaîne de Markov).

On se rappelle que les valeurs propres d'une matrice sont les racines de $\det(A - \lambda I)$. C'est utile de considérer deux théorèmes.

Théorème 3.3. *Les valeurs propres de A sont exactement les mêmes que les valeurs propres de A^T (mais les vecteurs propres sont typiquement différents).* □

Démonstration. Les polynômes caractéristiques de A et A^T (la transposée de A) sont identiques :

$$\det(A - \lambda I) = \det(A - \lambda I)^T = \det(A^T - (\lambda I)^T) = \det(A^T - \lambda I).$$

□

Théorème 3.4. *Si la somme de chaque rangée d'une matrice est égale à r (donc, constante) alors le vecteur $\mathbf{1} = (1, 1, \ldots, 1)$ est vecteur propre avec valeur propre $\lambda = r$.* □

Vous êtes invités à démontrer ce théorème.

On applique. Soit A une matrice stochastique. Alors les rangées de A^T ont chacune une somme de 1 (pourquoi ?). Donc le théorème 3.4 nous assure que $\mathbf{1}$ est vecteur propre de A^T et que 1 est valeur propre de A^T. Donc, par le théorème 3.3, on sait que 1 est valeur propre de A (mais on ne connaît pas le vecteur propre correspondant de A).

Résultat : toute matrice stochastique a 1 comme valeur propre. Donc il existe un vecteur non-nul $\mathbf{q}$ tel que $A\mathbf{q} = \mathbf{q}$. On peut choisir $\mathbf{q}$ pour que la somme des composantes donne 1 (parce que tous les multiples non-nuls de $\mathbf{q}$ sont des vecteurs propres correspondants à la même valeur propre), donc on a une distribution d'équilibre, souvent dit *vecteur d'état stationnaire*. On a donc le suivant.

Théorème 3.5. *Toute chaîne de Markov possède un vecteur d'état stationnaire.* □

3.3. **Équilibre et tendance.** Un vecteur d'état stationnaire se trouve comme tout autre vecteur propre.

Exemple 3.6. On détermine le vecteur d'état stationnaire de l'exemple 3.1. On cherche les vecteurs propres correspondant à $\lambda = 1$, donc on solutionne $A - I = \mathbf{0}$.

$$\begin{bmatrix} 0,97 & 0,05 \\ 0,03 & 0,95 \end{bmatrix} - I = \begin{bmatrix} -0,03 & 0,05 \\ 0,03 & -0,05 \end{bmatrix} \quad \to \quad \begin{bmatrix} 1 & -5/3 \\ 0 & 0 \end{bmatrix}$$

Ceci donne la solution

$$\mathbf{q} = t \begin{bmatrix} 5/3 \\ 1 \end{bmatrix}, \quad t \in \mathbb{R}.$$

On choisit la valeur de t pour que $\mathbf{q}$ représente des proportions : c'est-à-dire on choisit t pour avoir la somme des composantes de $\mathbf{q}$ égale à 1. Donc $t = 3/8$ ou

$$\mathbf{q} = \begin{bmatrix} 5/8 \\ 3/8 \end{bmatrix} = \begin{bmatrix} 0,625 \\ 0,375 \end{bmatrix}.$$

On interprète : la situation d'avoir $62,5\%$ de la population non-malade et $37,5\%$ malade est stable. □

On se demande la tendance éventuelle des vecteurs d'état $\mathbf{x}_k$.

Exercice 3.7. Montrer que ce n'est jamais le cas que $\mathbf{x}_k$ converge vers $\mathbf{0}$. (Indice : montrer que la somme des composantes de $\mathbf{x}_k$ est égale à 1 pour $k = 1, 2, 3, \ldots$.) □

C'est déjà une différence entre les chaînes de Markov et les systèmes dynamiques en générale : ici l'extinction est impossible.

Exercice 3.8. Montrer que ce n'est jamais le cas que $\mathbf{x}_k$ augmente sans cesse ("converge vers l'infini"). (Indice : montrer que la somme des composantes de $\mathbf{x}_k$ est égale à 1 pour $k = 1, 2, 3, \ldots$.) □

Par contre ce n'est **pas** le cas que pour tout vecteur d'état $\mathbf{x}_0$ et toute matrice stochastique A, $\mathbf{x}_k = A^k\mathbf{x}_0$ converge vers un seul vecteur $\mathbf{q}$.

3.4. **Chaînes de Markov et valeurs propres.** Souvent une chaîne de Markov possède les deux propriétés suivantes.

– Il n'existe qu'un seul vecteur d'état stationnaire.
– Peu importe le vecteur d'état initial $\mathbf{x}_0$, les vecteurs $\mathbf{x}_k$ convergent vers $\mathbf{q}$.

Afin de mieux comprendre ceci, voyons quelques exemples où ce n'est pas le cas.

Exemple 3.9. Supposons qu'une chaîne de Markov possède comme matrice de transition

$$A = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}.$$

En addition, supposons que l'état initial est $\mathbf{x}_0 = \begin{bmatrix} 0,2 \\ 0,8 \end{bmatrix}$. Alors,

$$\mathbf{x}_1 = A\mathbf{x}_0 = \begin{bmatrix} 0,8 \\ 0,2 \end{bmatrix}, \quad \mathbf{x}_2 = A\mathbf{x}_1 = \begin{bmatrix} 0,2 \\ 0,8 \end{bmatrix}, \quad \ldots.$$

On voit que les vecteurs de convergent pas. Cependant, il existe un état stationnaire $\mathbf{x} = \begin{bmatrix} 0,5 \\ 0,5 \end{bmatrix}$. Les vecteurs convergent si et seulement si l'état initial est $\mathbf{x}_0 = \begin{bmatrix} 0,5 \\ 0,5 \end{bmatrix}$.

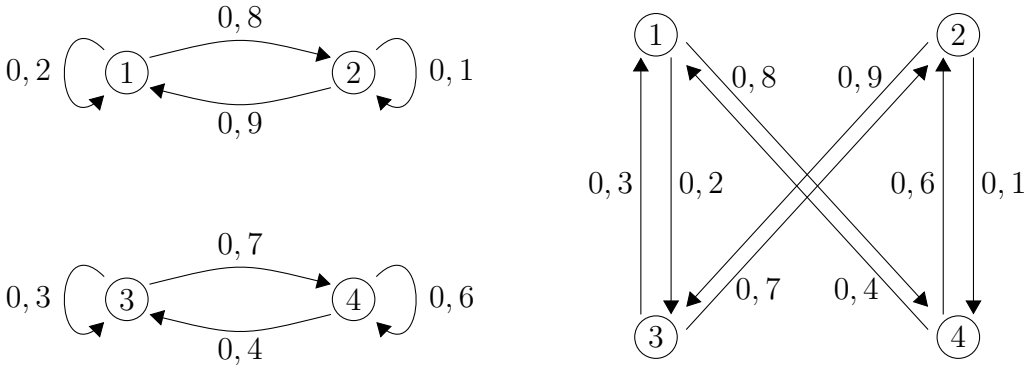
Maintenant, deux exemples un peu plus compliqués.

Exercice 3.10. Vérifier que la première matrice de transition possède plus qu'un vecteur d'état stationnaire. Vérifier que pour la deuxième matrice les vecteurs d'état ne convergent pas.

$$\begin{bmatrix} 0,2 & 0,9 & 0 & 0 \\ 0,8 & 0,1 & 0 & 0 \\ 0 & 0 & 0,3 & 0,4 \\ 0 & 0 & 0,7 & 0,6 \end{bmatrix} \qquad \begin{bmatrix} 0 & 0 & 0,3 & 0,4 \\ 0 & 0 & 0,7 & 0,6 \\ 0,2 & 0,9 & 0 & 0 \\ 0,8 & 0,1 & 0 & 0 \end{bmatrix}$$

(Indice : les chiffres ne sont pas tellement importants, c'est plutôt les zéros) □

Ce serait utile de considérer les graphes qui correspondent à ces matrices. Pour une chaîne de Markov on construit le graphe en mettant un sommet pour chaque état et une flèche pour chaque transition possible. Voici les graphes correspondants.



Les valeurs propres se montrent utiles ici.

Théorème 3.11. *Soit A une matrice stochastique diagonalisable tel que $\lambda_1 = 1$ est valeur propre de multiplicité 1, et que $|\lambda_j| < 1$ pour toute autre valeur propre λ_j. Alors il n'existe qu'un seul vecteur d'état stationnaire $\mathbf{q}$ et peu importe le vecteur d'état initial $\mathbf{x}_0$, les vecteurs $\mathbf{x}_k$ convergent vers $\mathbf{q}$.* □

La démonstration de ce théorème est déjà faite : c'est l'équation (2.5). En principe il ne faut que calculer les valeurs propres de A et vérifier que la multiplicité de $\lambda = 1$ est un et que les autres sont tous inférieures à 1 en valeur absolue. En pratique ce n'est si facile : trouver directement les valeurs propres est difficile (parfois impossible ?) pour une grande matrice. De plus, il existe des matrices stochastiques qui ne sont pas diagonalisables.

Qu'est-ce qui se passe dans l'exercice 3.10 ? Pour la première matrice, la multiplicité de $\lambda = 1$ est 2 (exercice !). Le vecteur d'état stationnaire n'est pas unique. Pour la deuxième, on voit que $\lambda = 1$ et $\lambda = -1$ sont valeurs propres. Donc l'équation (2.4) est valide mais il faut modifier l'équation (2.5) (voir l'exercice 2.4). Donc on voit que, dans les deux exemples de l'exercice 3.10, le théorème 3.11 ne s'applique pas.

3.5. **Chaînes de Markov régulières.** Il y a une autre approche. On a un théorème qui garantit la convergence désirée, sans devoir calculer les valeurs propres. Une matrice stochastique A est *régulière* s'il existe un entier ℓ tel que toute composante de A^ℓ est positive.

Théorème 3.12. *Soit A une matrice stochastique régulière. Alors il existe un seul vecteur d'état stationnaire $\mathbf{q}$. De plus pour tout vecteur d'état initial $\mathbf{x}_0$ les vecteurs d'état $\mathbf{x}_k$ convergent vers $\mathbf{q}$.* □

La démonstration est un peu technique. Parfois c'est directement utile.

Exemple 3.13. Considerez une chaîne de Markov avec matrice de transition

$$A = \begin{bmatrix} 0,8 & 0,3 & 0 \\ 0 & 0,2 & 0,5 \\ 0,2 & 0,5 & 0,5 \end{bmatrix}.$$

Alors

$$A^2 = \begin{bmatrix} 0,64 & 0,3 & 0,15 \\ 0,1 & 0,29 & 0,35 \\ 0,26 & 0,41 & 0,5 \end{bmatrix}.$$

Donc A est régulière (on prends $\ell = 2$ dans la définition) et on sait qu'il existe un seul vecteur d'état stationnaire et pour tout vecteur d'état initial, les vecteurs d'état convergent vers ce vecteur d'état stationnaire.

Exercice 3.14. Montrer que la chaîne de Markov de l'exemple 3.1 converge toujours vers le même vecteur d'état stationnaire. (indice : il faut simplement trouver une valeur de ℓ tel que...) □

Leçon 5 : 22 septembre 2011

Parfois il semble utile, mais pas autant qu'on voudrait.

Exemple 3.15. Tenter de calculer le carré de la première matrice A de l'exercice 3.10. On peut montrer, sans calculer tout, que A^2 possède des zéros au même endroits que A (exercice !). Dans la même manière, on peut voir qu'il n'existe aucun entier ℓ tel que toute composante de A^ℓ est positif (car A^ℓ possède les mêmes zéros que A).

Par contre le théorème 3.12 ne permet pas de conclure qu'il y a plus qu'un vecteur d'état stationnaire. Ce théorème ne s'applique tout simplement pas, car il s'applique seulement dans les cas ou un ℓ existe. □

Mais ce n'est pas toujours pratique. C'est vrai que $\ell = 1$ ne suffit pas pour les matrices de l'exercice 3.10, mais peut-être que $\ell = 437$ suffit ? ou $\ell = 142857$? Le problème c'est qu'on ne connaît pas la valeur de ℓ.

Considérons un graphe (par exemple, le graphe d'une matrice stochastique). Si c'est possible de trouver un chemin de n'importe quel état à n'importe quel autre état, alors on dit que le graphe est *fortement connexe*. Un chemin, dans ce sens, doit suivre les flèches dans la "bonne" direction. Un *cycle* est un chemin qui aboutit à son point de départ. La *longueur* d'un chemin ou d'un cycle est le nombre de flèches dans le chemin ou cycle (on peut traverser une flèche plus qu'une fois et, dans cette situation, on la compte plus qu'une fois dans la calcule de la longueur). Si les longueurs de tous les cycles dans un graphe ont un facteur en commun, on dit que le graphes est *périodique* (le plus grand facteur commun est la période). Sinon il est *apériodique*.

Ces définitions mènent au théorème suivant.

Théorème 3.16. *Soit A une matrice stochastique tel que son graphe est fortement connexe et apériodique. Alors A est régulière. Donc, il existe un seul vecteur d'état stationnaire $\mathbf{q}$, et pour tout vecteur d'état initial $\mathbf{x}_0$ les vecteurs d'état $\mathbf{x}_k$ convergent vers $\mathbf{q}$.* □

3.6. **Puissances.** Posons A une matrice stochastique régulière (peut-être qu'on a vérifié que le graphe est fortement connexe et apériodique). Posons $\mathbf{q}$ pour le vecteur d'état stationnaire. Il reste une observation qui est parfois très utile.

On sait que $A^k\mathbf{x}_0 \approx \mathbf{q}$ si k est "grand", peu importe le vecteur initial $\mathbf{x}_0$. Pour faciliter la présentation, imaginons un modèle à $n = 3$ états, donc A est de taille 3×3. On peut choisir $\mathbf{x}_0$ comme on veut, donc par exemple :

$$A^k \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} \approx \mathbf{q}, \qquad A^k \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} \approx \mathbf{q}, \qquad A^k \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \approx \mathbf{q}, \qquad \text{(pour } k \text{ grand)}.$$

Exercice 3.17. Que représente les trois produits matriciels ? Comme conséquence, que peut-on dire au sujet de A^k ? (Indice : considérer les colonnes de A^k) □

3.7. **Attention.** Un avertissement s'impose. On n'a pas du tout complété l'analyse de la théorie des chaînes de Markov. En particulier, c'est possible d'avoir une chaîne de Markov qui converge toujours vers un vecteur d'état stationnaire, mais n'est pas régulière.

Exercice 3.18. Montrer que la matrice $A = \begin{bmatrix} 0 & 0 \\ 1 & 1 \end{bmatrix}$ n'est pas régulière mais on a encore que pour tout vecteur d'état initial $\mathbf{x}_0$, les vecteurs $\mathbf{x}_k = A\mathbf{x}_0$ convergent vers un vecteur d'état stationnaire unique $\mathbf{q}$. Déterminer aussi $\mathbf{q}$.

Pouvez-vous donner d'autres exemples similaires ? □

4. Équations de récurrence

4.1. **Introduction.** Une *équation de récurrence* est un peu comme un système dynamique. Mais au lieu d'avoir *plusieurs* variables qui dépendent de leurs valeurs à *une* étape précédente, on a *une* variable qui dépend de *plusieurs* étapes précédentes.

Une récurrence de la forme générale serait de la forme

$$f_k = \alpha_1 f_{k-1} + \alpha_2 f_{k-2} + \cdot + \alpha_n f_{k-n},$$

ou $n \geq 1$ est un entier. On dit que *l'ordre* de la récurrence est la différence maximale des indices : ici c'est n, car on a f_k en termes des n valeurs précédentes. Pour calculer f_k il faut connaître n valeurs précédentes.

Exemple 4.1. La population mondial est environ $6,8$ milliards, et a augmenté d'environ 1.1% depuis l'année dernière. Si on présume que cette augmentation continue, on a le modèle suivant :

$$p_k = 1.011 p_{k-1}, \qquad p_0 = 6,8.$$

Ceci est un système dynamique avec une seule population. C'est aussi une récurrence d'ordre 1. □

4.2. **Solution matricielle : exemple.**

Exemple 4.2. Les *nombres de Fibonacci* sont obtenu par récurrence $f_k = f_{k-1} + f_{k-2}$, avec $f_0 = 0$ et $f_1 = 1$. Donc on obtient

$$0, 1, 1, 2, 3, 5, 8, 13, 21, 34, \ldots$$

C'est une récurrence d'ordre 2. □

On pose deux questions : Comment calculer $f_{1000000}$ directement, c'est-à-dire sans devoir calculer toutes les valeurs intermédiaires ? Plus généralement, est-ce qu'on peut trouver une expression explicite (c.-à.-d. directe) pour f_k en termes de k (et pas en termes les valeurs de f_j pour $j < k$). Que peut-on dire sur la tendance éventuelle de f_n ?

La solution est essentiellement la même que ce qu'on a fait pour un système dynamique, mais il faut reformuler le problème. On pose

$$\mathbf{x}_k = \begin{bmatrix} f_k \\ f_{k-1} \end{bmatrix}$$

et on cherche une matrice A telle que $\mathbf{x}_k = A\mathbf{x}_{k-1}$ (c'est la même chose que $\mathbf{x}_{k+1} = A\mathbf{x}_k$, mais la forme présente sera plus claire ici).

$$\mathbf{x}_k = A\mathbf{x}_{k-1} \quad \Longleftrightarrow \quad \begin{bmatrix} f_k \\ f_{k-1} \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \begin{bmatrix} f_{k-1} \\ f_{k-2} \end{bmatrix} \quad \Longleftrightarrow \quad \begin{cases} f_k = a_{11} f_{k-1} + a_{12} f_{k-2} \\ f_{k-1} = a_{21} f_{k-1} + a_{22} f_{k-2} \end{cases}$$

La première équation suggère de prendre $a_{11} = 1$ et $a_{12} = 1$ pour donner $f_k = f_{k-1} + f_{k-2}$. Que faire de la deuxième ? Afin d'avoir $\mathbf{x}_k = A\mathbf{x}_{k-1}$ il faut que cette deuxième équation soit valide, mais on a déjà la récurrence, donc il nous faut rien de plus. La solution c'est d'ajouter une équation vraie mais redondante : $f_{k-1} = f_{k-1}$, ce qui s'accompli en mettant $a_{21} = 1$ et $a_{22} = 0$. Donc

$$\mathbf{x}_k = \begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix} \mathbf{x}_{k-1}.$$

Comme système algébrique, c'est exactement pareil à un système dynamique avec deux variables. La seule différence est *l'interprétation* du vecteur des "populations". On le résout de la même manière, en calculant les valeurs et vecteurs propres.

Exercice 4.3. Monter que les valeurs propres de $A = \begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix}$ sont $\lambda_1 = \frac{1+\sqrt{5}}{2}$ et $\lambda_2 = \frac{1-\sqrt{5}}{2}$, avec vecteurs propres $\mathbf{v}_1 = \begin{bmatrix} \frac{1+\sqrt{5}}{2} \\ 1 \end{bmatrix}$ et $\mathbf{v}_2 = \begin{bmatrix} \frac{1-\sqrt{5}}{2} \\ 1 \end{bmatrix}$. □

Exercice 4.4. Montrer que le vecteur $\mathbf{x}_1 = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$ peut s'écrire comme $\mathbf{x}_1 = \frac{1}{\sqrt{5}}\mathbf{v}_1 - \frac{1}{\sqrt{5}}\mathbf{v}_2$. □

Sachant ceci, on peut écrire la solution générale pour $\mathbf{x}_k$. Un détail : on ne commence pas avec $\mathbf{x}_0$, mais plutôt avec $\mathbf{x}_1$. C'est parce que $\mathbf{x}_1 = \begin{bmatrix} f_1 \\ f_0 \end{bmatrix}$ mais $\mathbf{x}_0 = \begin{bmatrix} f_0 \\ f_{-1} \end{bmatrix}$. Donc on obtient

$$\begin{aligned} \mathbf{x}_k &= A^{k-1}\mathbf{x}_1 \\ &= A^{k-1}\left(\frac{1}{\sqrt{5}}\mathbf{v}_1 - \frac{1}{\sqrt{5}}\mathbf{v}_2\right) \\ &= \frac{1}{\sqrt{5}}(\lambda_1)^{k-1}\mathbf{v}_1 - \frac{1}{\sqrt{5}}(\lambda_2)^{k-1}\mathbf{v}_2. \end{aligned}$$

En termes des nombres f_k on obtient

$$\begin{bmatrix} f_k \\ f_{k-1} \end{bmatrix} = \frac{1}{\sqrt{5}}\left(\frac{1+\sqrt{5}}{2}\right)^{k-1}\begin{bmatrix} \frac{1+\sqrt{5}}{2} \\ 1 \end{bmatrix} - \frac{1}{\sqrt{5}}\left(\frac{1-\sqrt{5}}{2}\right)^{k-1}\begin{bmatrix} \frac{1-\sqrt{5}}{2} \\ 1 \end{bmatrix}$$

$$\iff \begin{cases} f_k = \frac{1}{\sqrt{5}}\left(\frac{1+\sqrt{5}}{2}\right)^{k-1}\frac{1+\sqrt{5}}{2} - \frac{1}{\sqrt{5}}\left(\frac{1-\sqrt{5}}{2}\right)^{k-1}\frac{1-\sqrt{5}}{2} \\ f_{k-1} = \frac{1}{\sqrt{5}}\left(\frac{1+\sqrt{5}}{2}\right)^{k-1} - \frac{1}{\sqrt{5}}\left(\frac{1-\sqrt{5}}{2}\right)^{k-1} \end{cases}$$

$$\iff \begin{cases} f_k = \frac{1}{\sqrt{5}}\left(\frac{1+\sqrt{5}}{2}\right)^{k} - \frac{1}{\sqrt{5}}\left(\frac{1-\sqrt{5}}{2}\right)^{k} \\ f_{k-1} = \frac{1}{\sqrt{5}}\left(\frac{1+\sqrt{5}}{2}\right)^{k-1} - \frac{1}{\sqrt{5}}\left(\frac{1-\sqrt{5}}{2}\right)^{k-1} \end{cases}$$

On obtient deux fois la même relation. Ce n'est pas surprenant, car f_{k-1} et f_k ne sont pas deux populations différentes, mais la même population à deux intervalles successifs. Aussi, on a construit l'équation matricielle en ajoutant une équation redondante : on n'est pas surpris d'en recevoir dans la réponse une équation redondante aussi.

On a donc une formule exacte pour f_k, sans devoir calculer les valeurs intermédiaires. On se rappelle que f_k est toujours un entier positif. Ceci n'est pas évident dans la formule exacte. Pourtant, toutes les $\sqrt{5}$ s'annulent dans le calcul de f_k. Même si on ne cherche à calculer que des suites d'entiers, on entraîne des nombres irrationnels. Parfois, on entraîne même le calcul avec des nombres complexes (par exemple, si la matrice possède des valeurs propres complexes). Ceci se produit pour des applications très concrets et réelles.

Leçon 6 : 26 septembre 2011

On a aussi une approximation de la tendance éventuelle, car une des valeurs propres est plus grande que l'autre ($\lambda_1 \approx 1,618$ et $\lambda_2 \approx -0,618$). Donc pour k "grand" on a

$$f_k \approx e_k := \frac{1}{\sqrt{5}}\left(\frac{1+\sqrt{5}}{2}\right)^{k}.$$

L'approximation e_k n'est certainement pas un entier. On compare :

f_k :	0	1	1	2	3	5	8	13	21	34
e_k :	0,45	0,72	1,17	1,89	3,07	4,96	8,02	12,98	21,01	33,99

4.3. **Solution matricielle : générale.** En générale, la même méthode s'applique. On commence avec une récurrence d'ordre n :

$$f_k = \alpha_1 f_{k-1} + \alpha_2 f_{k-2} + \cdots + \alpha_n f_{k-n}.$$

On définie un vecteur (de taille n) :

$$\mathbf{x}_k = \begin{bmatrix} f_k \\ f_{k-1} \\ \vdots \\ f_{k-n} \end{bmatrix}.$$

et on cherche une matrice A tel que $\mathbf{x}_k = A\mathbf{x}_{k-1}$. La première rangée de cette équation matricielle est exactement la récurrence, à laquelle on ajoute $n-1$ équations redondantes : $f_{k-1} = f_{k-1}$, $f_{k-2} = f_{k-2}$, …, $f_{k-n} = f_{k-n}$. Ceci donne la matrice (de taille $n \times n$) :

$$A = \begin{bmatrix} \alpha_1 & \alpha_2 & \cdots & \alpha_{n-1} & \alpha_n \\ 1 & 0 & \cdots & 0 & 0 \\ 0 & 1 & \cdots & 0 & 0 \\ 0 & 0 & \ddots & 0 & 0 \\ 0 & 0 & \cdots & 1 & 0 \end{bmatrix}$$

On mentionne un résultat théorique utile ici. La matrice A a une forme spéciale : la première rangée contient la récurrence, et en bas on a une matrice identité et une colonne de zéros. On peut montrer que le polynôme caractéristique de A est

$$\det(xI - A) = x^n - \alpha_1 x^{n-1} - \alpha_2 x^{n-2} - \cdots - \alpha_n x^0.$$

On pourrait combiner un système dynamique avec une récurrence.

Exemple 4.5. On considère deux populations, qui dépendent de deux temps précédentes.

$$c_k = 0,2c_{k-1} + 0,4c_{k-2} + 0,3r_{k-1}$$
$$r_k = -0,104c_{k-1} + 0,8r_{k-1} + 0,3r_{k-2}$$

On aura un vecteur

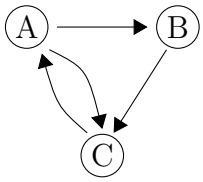
$$\mathbf{x}_k = \begin{bmatrix} c_k \\ c_{k-1} \\ r_k \\ r_{k-1} \end{bmatrix}$$

et une matrice A telle que $\mathbf{x}_k = A\mathbf{x}_{k-1}\ldots$ □

5. Mettre tout en ordre

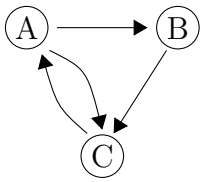
5.1. **Introduction.** On cherche un algorithme pour déterminer un *ordre d'importance* dans un réseau.

Exemple 5.1. Une ligue de football comprend trois équipes : A, B et C. En tout, ils ont joué quatre jeux : A a perdu contre B et C, B a perdu contre C, et C a perdu contre A. Ceci se résume dans un graphe où $X \to Y$ indique que X et Y ont joué une partie, et que Y a gagné. **Attention :** En classe, j'ai fait une erreur. J'ai dit que $X \to Y$ indique que X a gagné (et donc j'ai fait des autres changement à la question qui étaient incorrects aussi). Les flèches pointent toujours vers l'équipe qui a gagné.



S'il y aurait une équipe qui avait gagné contre toutes les autres, on pourrait facilement l'identifier comme la meilleure. Mais ce n'est pas le cas (et en générale ce ne serait pas le cas). Comment choisir ? □

Exemple 5.2. Un internet comprend trois pages web : A, B et C (c'est un exemple simplifié. . .). Les liens sont indiqués dans le graphe suivant, où $X \to Y$ indique que la page X fait un lien vers la page Y.



Déterminer la page web "le plus important" ; mieux, donner un ordre aux pages. (On va voir que ce n'est pas le même problème, même si les graphes sont les mêmes.) □

5.2. **Équipes.** Au lieu de déterminer la meilleure, identifions plutôt un "score" pour chaque équipe. L'ordre des équipes correspondra à l'ordre des scores. La meilleure équipe sera celle avec la plus grande score, la deuxième équipe sera celle avec la deuxième score, etc. Comment déterminer les scores ?

On pose w_A, w_B et w_C les trois scores. Chaque victoire devrait avancé le score, mais de combien ? On pourra simplement dire que le score est le nombre de victoires de l'équipe qui donnera $w_A = 2$, $w_B = 1$ et $w_C = 1$. Même sur ce petit exemple, on voit déjà des difficultés : on aura que A est meilleur et que B et C sont égales, pourtant C a gagné contre la meilleure équipe

et B n'a pas. Les équipes B et C sont classées comme égales, mais la victoire de C était "plus importante que celle de B". On pose donc plutôt que le score devrait être égale à la somme des scores de toutes les équipes qui ont été défaites. Pour des raisons techniques, on permet un facteur multiplicatif, donc on pose

$$w_X = \alpha\,(\text{somme de toutes les } w_Y \text{ où } Y \text{ a perdu contre } X)\,. \tag{5.1}$$

Le graphe de la ligue de l'exemple 5.1 peut s'écrire en termes de matrice, òu $A_{ij} = 1$ si $j \to i$, c'est-à-dire si l'équipe j a perdu contre i. On peut aussi mettre les scores en vecteur. Pour l'exemple 5.1 on obtient alors

$$A = \begin{bmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 1 & 1 & 0 \end{bmatrix}, \qquad \mathbf{w} = \begin{bmatrix} w_A \\ w_B \\ w_C \end{bmatrix}.$$

Alors la condition (5.1) s'écrit comme $\mathbf{w} = \alpha A\mathbf{w}$. En autres mots, on a une équation de valeur et vecteur propre $A\mathbf{w} = \frac{1}{\alpha}\mathbf{w}$ avec valeur propre $\frac{1}{\alpha}$ et vecteur propre $\mathbf{w}$. Les scores sont exactement les composantes du vecteur propre.

Comment choisir le vecteur propre ? Il y a peut-être plusieurs. On s'inspire des idées des chaînes de Markov.

Exercice 5.3. Est-ce que la matrice A ci-haut est la matrice de transition d'une chaîne de Markov ? □

Bien que la réponse à l'exercice précédent est "non", on a encore un espoir n'est pas perdu, car le théorème 3.16 reste encore valide.

Théorème 5.4. *Soit A une matrice avec toutes les composantes non-négatives. Si le graphe correspondante est fortement connexe et apériodique alors la valeur propre dominante λ_1 est positive et de multiplicité un, toute autre valeur propre λ_j satisfait $|\lambda_j| < \lambda_1$, et toute composante du vecteur propre correspondant à λ_1 est positif.* □

Le théorème 3.16 est le cas spécial de celui-ci où la valeur propre dominante est égale à 1.

Donc pour résoudre la question de l'exemple 5.1, on devrait calculer le vecteur propre dominante, et *c'est exactement l'ordre d'importance des équipes.*

On détermine avec un peu de calcul que la valeur et vecteur propre dominante sont

$$\lambda_1 \approx 1,32 \quad , \qquad \mathbf{v}_1 \approx \begin{bmatrix} 1 \\ 0,75 \\ 1,32 \end{bmatrix}.$$

Donc l'équipe C est la meilleure, suivie de A, suivie de B.

On se rappelle que calculer toutes les valeurs et vecteurs propres est une tâche onéreuse, surtout pour des grandes matrices. Par contre, on ne cherche qu'une seule chose : le vecteur propre dominante (on pourrait presque dire vecteur d'état stationnaire sauf que...).

On s'inspire donc d'une technique qu'on a vu pour les chaînes de Markov : on considère les *puissances* de A. Si on prend k suffisamment grande, alors les colonnes de A^k seront toutes des multiples de ce vecteur propre dominante (approximativement). On se rappelle que pour une matrice stochastique régulière P, les colonnes de P^k sont approximativement toutes égales, et égales au vecteur d'état stationnaire (pour k suffisamment grand). Ici, la matrice n'est pas stochastique qui donne qu'on aura des *multiples* du vecteur propre dominante.

En calculant (par ordinateur bien sûr !) :

$$A^{10} = \begin{bmatrix} 7 & 4 & 5 \\ 5 & 3 & 4 \\ 9 & 5 & 7 \end{bmatrix}, \qquad A^{20} = \begin{bmatrix} 114 & 65 & 86 \\ 86 & 49 & 65 \\ 151 & 86 & 114 \end{bmatrix}.$$

Exercice 5.5. Vérifier si les colonnes de A^{10} sont toutes des multiples du vecteur propre $\begin{bmatrix} 1 \\ 0{,}75 \\ 1{,}32 \end{bmatrix}$ (e.g., en divisant chaque colonne par sa première valeur). Faire de même pour A^{20}. □

5.3. **Pages web.** On voit que l'exemple 5.2 est essentiellement la même que l'exemple 5.1. Une "perte" est maintenant un "lien vers", mais dans un sens c'est la même chose. Dans les deux exemples, les flèches indiquent la progression : soit vers la meilleure équipe, soit vers la prochaine page.

Il y a une différence technique. Dans la ligue, chaque jeux a une certaine influence : l'information totale est le nombre de jeux. Mais sur l'internet, un page qui fait plusieurs liens donne moins d'importance à chacun. La solution c'est de créer une chaîne de Markov. Dans chaque page, on accorde une probabilité égales à chacun des liens. Ceci est équivalent à diviser chaque colonne de A par sa somme.

$$A = \begin{bmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 1 & 1 & 0 \end{bmatrix} \qquad \longrightarrow \qquad P = \begin{bmatrix} 0/2 & 0/1 & 1/1 \\ 1/2 & 0/1 & 0/1 \\ 1/2 & 1/1 & 0/1 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 \\ 0.5 & 0 & 0 \\ 0.5 & 1 & 0 \end{bmatrix}$$

La matrice P représente la matrice de transition du "surfeur aléatoire" : à chaque page web, il choisit par hasard un des liens et le suit. Au long terme, les surfeurs aléatoires seront décrits par le vecteur d'état stationnaire.

On cherche donc le vecteur d'état stationnaire de P (pour une matrice stochastique, le vecteur d'état stationnaire et le vecteur propre dominante sont exactement la même chose). Quelques calculs donnent la valeur et vecteur propre dominante :

$$\lambda_1 = 1, \qquad \mathbf{v}_1 = \begin{bmatrix} 0,4 \\ 0,2 \\ 0,4 \end{bmatrix}.$$

On a ici que les pages A et C sont d'importance égale, et que B est moins importante. Note que ce n'est pas la même chose que la ligue. C'est raisonnable que les deux approches donnerait pas exactement la même importance. Dans la ligue, si C aurait perdu contre A 15 fois, on aurait peut-être changé notre opinion sur l'ordre des équipes. Mais qu'une page web fait 15 liens vers une autre page ne devrait pas compter 15 fois.

Exercice 5.6. Comment saviez-vous sans calcul que la valeur propre dominante est 1 ? □

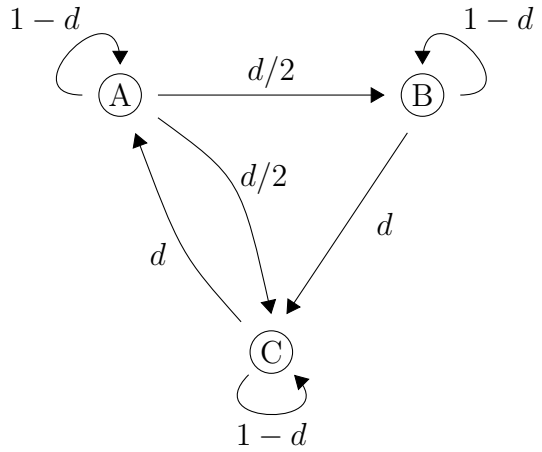
On pourra aussi calculer des puissances de P afin de voir le vecteur d'état stationnaire dans les colonnes.

$$P^{10} = \begin{bmatrix} 0,40625 & 0,4375 & 0,37500 \\ 0,18750 & 0,1875 & 0,21875 \\ 0,40625 & 0,3750 & 0,40625 \end{bmatrix}, \qquad P^{20} = \begin{bmatrix} 0,3994140625 & 0,400390625 & 0,4003906250 \\ 0,2001953125 & 0,199218750 & 0,2001953125 \\ 0,4003906250 & 0,400390625 & 0,3994140625 \end{bmatrix}$$

Exercice 5.7. Est-ce que $k = 10$ est "grand" ? C'est-à-dire, est-ce que toutes les colonnes de P^{10} sont à peu près égales ? Et $k = 20$? En vous fiant aux puissances données, donner une approximation du vecteur d'état stationnaire et la comparer au vecteur exact ci-haut. □

Leçon 7 : 29 septembre 2011

On peut faire mieux, en introduisant un peu de paresse : à chaque page web, on se permet l'option de ne rien faire, c'est-à-dire de rester sur la page présente. Posons que la probabilité de se déplacer est d. Cette probabilité se divise également entre les liens de la page, et le $1 - d$ restant s'attache à la page présente. Voici le graphe et la matrice de transition Q.



$$Q = \begin{bmatrix} 1-d & 0 & d \\ 0,5d & 1-d & 0 \\ 0,5d & d & 1-d \end{bmatrix}$$

Les deux matrices de transition, P et Q, sont fortement reliés.

Exercice 5.8. Montrer que $Q = dP + (1 - d)I$, où I représente la matrice d'identité. □

Exercice 5.9. Montrer que P et Q possèdent exactement les mêmes vecteurs propres (indice : calculer $I\mathbf{v}$ où $\mathbf{v}$ est vecteur propre de P). Quelles sont les valeurs propres de Q, en termes de celles de P ? En particulier, montrer que si $\mathbf{v}$ est vecteur propre de P avec valeur propre 1, alors $\mathbf{v}$ est aussi vecteur propre de Q avec valeur propre 1. qed

La conséquence est que le vecteur d'état stationnaire de P est exactement la même que le vecteur d'état stationnaire de Q. On pourra utiliser l'une ou l'autre matrice. La distinction,

c'est que la paresse est plus *rapide* : une puissance de Q converge typiquement plus rapidement vers le vecteur d'état stationnaire qu'une puissance de P. Par exemple, si $d = 0,5$, on a

$$Q^{10} \approx \begin{bmatrix} 0,399994 & 0,399963 & 0,400024 \\ 0,200012 & 0,200012 & 0,199982 \\ 0,399994 & 0,400024 & 0,399994 \end{bmatrix}.$$

On voit clairement le vecteur d'état stationnaire, à quatre décimales. À comparer avec P^{10}.

Il reste une autre optimisation (qui s'applique également aux chaînes de Markov et même au systèmes dynamiques).

Principe. Soit A une matrice. Les colonnes de A^k sont toutes approximativement multiples d'un même vecteur $\mathbf{v}$ si et seulement si $A^k\mathbf{x}_0$ est approximativement multiple de $\mathbf{v}$ pour tout vecteur $\mathbf{x}_0$. □

Donc au lieu de calculer A^k, ou aurait pu calculer $A^k\mathbf{x}_0$ pour n'importe quel vecteur de départ $\mathbf{x}_0$. L'avantage c'est au plan technique : calculer A^k entraîne multiplier matrice par matrice k fois, tandis que $A^k\mathbf{x}_0$ ne requiert que multiplier matrice par vecteur k fois.

Exercice 5.10. Soit $\mathbf{x}_0 = \begin{bmatrix} 0.5 \\ 0.5 \\ 0 \end{bmatrix}$. Calculer $P^{10}\mathbf{x}_0$, $P^{20}\mathbf{x}_0$ et $Q^{10}\mathbf{x}_0$ pour les matrices P et Q ci-haut, avec $d = 0,5$ (en utilisant les calculs de puissances déjà donnée). Vérifier qu'on obtient des approximations des vecteurs dominantes. Choisir d'autres vecteurs de départ et refaire.

Calculer à bras P^2, et ensuite calculer $P\mathbf{x}_0$, et $P(P\mathbf{x}_0)$. Expliquer pourquoi le calcul de $P^2\mathbf{x}_0$ et plus rapide que le calcul de P^2. □

Le calcul du vecteur d'état stationnaire de Q est essentiellement le calcul de PageRank de Google. Voir *The anatomy of a large-scale hypertextual search engine* sur l'internet...

6. Programmes linéaires

6.1. Introduction.

Exemple 6.1. Une usine fabrique deux sortes de vélos, X et Y. La main-d'oeuvre est distribuée parmi trois usines, A, B, et C. Les vélos de type X nécessitent 2 heures de travail dans l'usine A, 1 heure dans B et 1 heure dans C. Les vélos de type Y nécessitent 1 heure de travail dans l'usine A, 1 heure dans B et 3 heures dans C. Chaque vélo X remporte \$40 de profit et chaque vélo Y remporte \$60. Le temps disponible chaque semaine pour fabriquer des vélos est 70 heures dans l'usine A, 40 heures dans B et 90 heures dans C. Combien de vélos de chaque type devrait-on fabriquer afin de maximiser le profit ? □

Bien que les chiffres sont peut-être artificiels dans cet exemple, on peut voir un problème général. On cherche à optimiser une certaine fonction (ici, maximiser le profit total). En principe,

c'est simple : on augmente la production. Mais on a aussi des contraintes sur les ressources disponibles (ici, des limites sur le nombre d'heures au total).

Avant de passer aux détails, on observe que dans l'exemple 6.1, il y a deux variables : le nombre de vélos de type X et le nombre de type Y. Il y a aussi trois contraintes : chacune des trois usines a une limite sur le nombre d'heures disponible. Si on considère que chaque variable doit être positive, on a deux autres contraintes. On pourrait facilement imaginer une situation avec beaucoup plus de variables, et beaucoup plus de contraintes. Donc on ne cherche pas seulement une réponse à l'exemple 6.1, mais une réponse qu'on pourrait généraliser et même automatiser.

On demande plusieurs questions :
– Est-ce qu'il existe une solution optimale ?
– Est-ce que la solution optimale est unique ?
– Est-ce qu'on peut trouver une solution optimale d'une manière efficace ?

6.2. **Programmes linéaires : forme.** On commence en formalisant en peu de notation.

Dans l'exemple 6.1, on identifie deux paramètres qu'on peut modifier : le nombre de vélos de chaque type. Ce sont les variables. Notons x_1 pour le nombre de vélos de type X et x_2 pour le nombre de type Y. Étant donné des valeurs pour x_1 et x_2, le profit est

$$40x_1 + 60x_2$$

On dit que c'est la *fonction objective*, ou parfois l'*objectif*.

Il y a aussi trois limites sur les ressources ; autrement dit, trois raisons pourquoi on ne peut pas simplement fabriquer une infinité de vélos (pour un profit infini). Ce sont les *contraintes* du problème.

$$\begin{aligned} 2x_1 + 1x_2 &\leq 70 \\ 1x_1 + 1x_2 &\leq 40 \\ 1x_1 + 3x_2 &\leq 90 \end{aligned}$$

C'est souvent utile de présenter l'information en forme matricielle. On écrit alors

$$\max \ \mathbf{c}^T\mathbf{x} \quad \text{s.c.} \quad A\mathbf{x} \leq \mathbf{b}, \ \mathbf{x} \geq \mathbf{0}. \tag{6.1}$$

Ceci indique qu'on cherche à maximiser la quantité $\mathbf{c}^T\mathbf{x}$, sujet à la contrainte de $A\mathbf{x} \leq \mathbf{b}$, $\mathbf{x} \geq \mathbf{0}$. Note que $\mathbf{c}^T\mathbf{x}$ est une quantité (c'est-à-dire un chiffre et non pas un vecteur). Aussi, la notation de "$\leq$" et "$\geq$" s'applique à chaque coordonnée.

Pour l'exemple 6.1, on aurait

$$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}, \qquad \mathbf{c} = \begin{bmatrix} 40 \\ 60 \end{bmatrix}, \qquad A = \begin{bmatrix} 2 & 1 \\ 1 & 1 \\ 1 & 3 \end{bmatrix}, \qquad \mathbf{b} = \begin{bmatrix} 70 \\ 40 \\ 90 \end{bmatrix}.$$

Exercice 6.2. Vérifier que ces matrices donnent vraiment l'objectif et les contraintes de l'exemple 6.1. □

On peut imaginer d'autres variations : min au lieu de max, ou peut-être $A\mathbf{x} \geq \mathbf{b}$. Ce serait bien sûr une autre application : par exemple on cherche à minimiser les dépenses du gouvernement sujet à la nécessité de fournir un minimum de services aux citoyens. On pourrait exiger des valeurs négatives pour les variables, ou bien permettre n'importe quelle valeur.

On dit *programme linéaire* pour un problème qui cherche à optimiser une fonction linéaire sujet à une contrainte linéaire. Donc l'objectif est soit $\max \mathbf{c}^T\mathbf{x}$ ou $\min \mathbf{c}^T\mathbf{x}$. Les contraintes peuvent êtres une combinaison de $A\mathbf{x} \leq \mathbf{b}$, $A\mathbf{x} \geq \mathbf{b}$, $\mathbf{x} \geq \mathbf{0}$, $\mathbf{x} \leq \mathbf{0}$ ou $\mathbf{x}$ libre. On dit qu'un programme linéaire de la forme précise de l'équation (6.1) est en *forme canonique*.

Exemple 6.3. Voici quelques exemples de programmes linéaires.

$$\min \begin{bmatrix} 2 & 3 & -1 \end{bmatrix} \mathbf{x} \quad \text{s.c.} \quad \begin{bmatrix} 1 & 1 & 0 \\ 0 & 1 & -1 \end{bmatrix} \mathbf{x} \leq \begin{bmatrix} 10 \\ 10 \end{bmatrix}, \ \mathbf{x} \geq \mathbf{0}$$

$$\max \begin{bmatrix} 2 & 3 \end{bmatrix} \mathbf{x} \quad \text{s.c.} \quad \begin{bmatrix} 1 & 2 \\ 2 & 1 \end{bmatrix} \mathbf{x} \leq \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \ \mathbf{x} \geq \mathbf{0}$$

$$\max \begin{bmatrix} 1 & 1 \end{bmatrix} \mathbf{x} \quad \text{s.c.} \quad \begin{bmatrix} -1 & 1 \\ 1 & -2 \\ 2 & 2 \end{bmatrix} \mathbf{x} \leq \begin{bmatrix} 1 \\ 1 \\ 10 \end{bmatrix}, \ \mathbf{x} \geq \mathbf{0}$$

□

Exercice 6.4. Pour chaque programme linéaire ci-haut, identifier le nombre de variables ; c'est-à-dire, la taille du vecteur $\mathbf{x}$ dans chaque cas. Aussi identifier le nombre de contraintes, et écrire explicitement la fonction objective et chacune des contraintes. □

Bien que l'application serait en forme de maximiser ou minimiser, il n'y a pas de différence fondamental. On peut transformer un programme linéaire pour donner un système équivalent.

Exemple 6.5. Soit le programme linéaire $\min 2x_1 - x_2$ s.c. $x_1 - x_2 \geq 3$, $x_1 + x_2 \leq 4$.

Minimiser $2x_1 - x_2$ équivaut à maximiser la négative, c'est-à-dire maximiser $-2x_1 + x_2$. Aussi, exiger que $x_1 - x_2 \geq 3$ équivaut à exiger que $-x_1 + x_2 \leq -3$. Donc les deux programmes suivants sont équivalents :

$$\min \begin{bmatrix} 2 & -1 \end{bmatrix} \mathbf{x} \quad \text{s.c.} \quad \begin{cases} x_1 - x_2 \geq 3 \\ x_1 + x_2 \leq 4 \end{cases} \qquad \max \begin{bmatrix} -2 & 1 \end{bmatrix} \mathbf{x} \quad \text{s.c.} \quad \begin{bmatrix} -1 & 1 \\ 1 & 1 \end{bmatrix} \mathbf{x} \leq \begin{bmatrix} -3 \\ 4 \end{bmatrix}$$

6.3. **Région faisable : solution graphique.** Étant donné un programme linéaire, l'ensemble de toutes les valeurs permissibles des variables donne la *région faisable*. Formellement, pour un programme linéaire de la forme (6.1), la région faisable est l'ensemble de tout $\mathbf{x}$ avec $A\mathbf{x} \leq \mathbf{b}$ et $\mathbf{x} \geq \mathbf{0}$. C'est un sous-ensemble de $\mathbb{R}^n$, donc on peut imaginer une représentation graphique

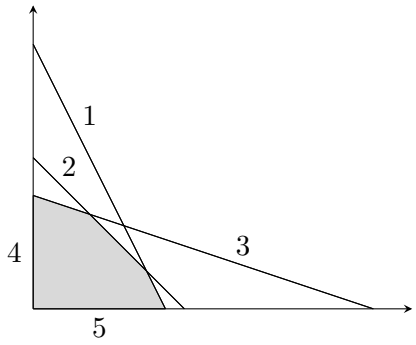
— du moins pour $n = 2$! Un point dans la région faisable est dit *solution faisable*. C'est une solution au programme linéaire qui n'est peut-être pas optimale, mais du moins légal !

Reprenons l'exemple 6.1. Pour chaque contrainte, on obtient une droite en remplaçant chaque inégalité par une égalité. L'inégalité correspond à l'une ou l'autre côté de la droite : c'est un demi-plan. La région faisable correspond à l'intersection de toutes les demi-plans.

Pour l'exemple 6.1, il y a cinq contraintes : on compte aussi $x_1 \geq 0$ et $x_2 \geq 0$ comme contraintes.

$$\begin{aligned} 2x_1 + 1x_2 &\leq 70 \\ 1x_1 + 1x_2 &\leq 40 \\ 1x_1 + 3x_2 &\leq 90 \\ x_1 &\geq 0 \\ x_2 &\geq 0 \end{aligned}$$

Donc afin de déterminer la région faisable on aurait cinq droites. Les contraintes sont numérotés dans l'ordre donnée. La région faisable est indiquée en gris.



Exercice 6.6. Déterminer les points d'intersection de chaque pair de droites. Note qu'afin de faire un graphique précis, c'est souvent utile de calculer les interceptes : ce sont les points d'intersection d'une des "vraies" droites avec $x_1 = 0$ ou $x_2 = 0$. □

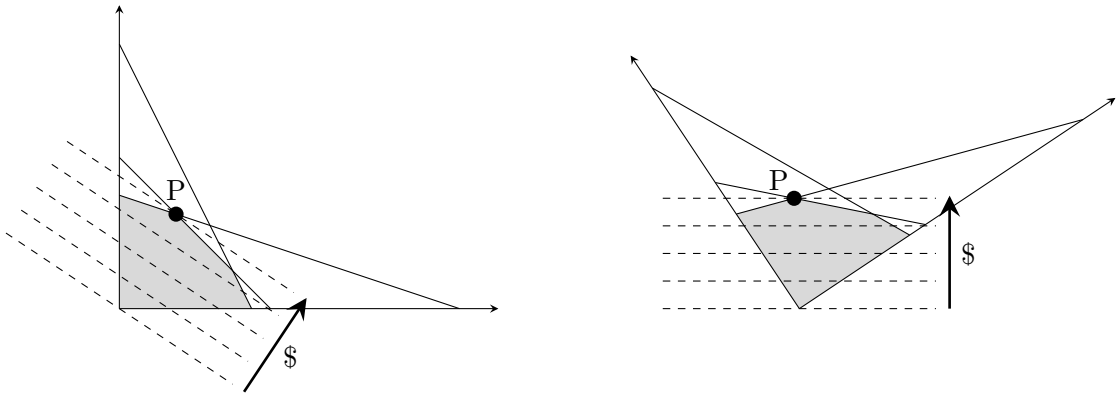
Pour chaque point dans le plan, on pourrait calculer un profit. Par exemple, si $x_1 = -100$ et $x_2 = 20$ on obtient un profit de $40x_1 + 60x_2 = -\$3880$. Mais ce calcul est illusoire, car le point $(-100, 20)$ n'est pas faisable. On peut voir que ce n'est pas faisable de deux façons : soit en le plaçant sur le graphique, ou mieux, en calculant $A\mathbf{x}$ et comparant à $\mathbf{b}$.

Exercice 6.7. Pour chaque point donné, déterminer le profit correspondant. Lesquelles sont des solutions faisables ? En comparant les profits à chaque point, déterminé lesquels ne sont *pas* optimales.

Points : $(-20, 100)$, $(0, 0)$, $(10, 10)$, $(15, 15)$, $(30, 0)$, $(0, 25)$. □

On peut solutionner un programme linéaire graphiquement, en suivant la tendance de la fonction objective sur la région faisable. On identifie les surfaces de valeur constant de l'objectif. Soit $M = \mathbf{c}^T\mathbf{x}$ la valeur de l'objective. On trace une série de courbes correspondant à des valeurs croissantes de M, afin d'identifier le point optimale dans la région faisable.

Exemple 6.8. Pour l'exemple 6.1, on a comme départ $40x_1 + 60x_2 = 0$: c'est une droite passant par l'origine. En traçant une série de droites de la forme $40x_1 + 60x_2 = M$ pour divers valeurs de M, on déduit que la direction de croissance est exactement normale à cette droite, c'est-à-dire, dans la direction $(40, 60)$. La flèche "\$" indique la direction de croissance du profit. C'est indiqué dans le graphique à gauche. C'est peut-être utile de faire une rotation, comme à droite.



On trouve que la solution optimale se trouve au point $P = (15, 25)$. On y retrouve un profit de $40(15) + 60(25) = \$2100$. □

Exercice 6.9. Pour chaque programme linéaire de l'exemple 6.3, faire un graphique de la région faisable. Déterminer la direction de croissance de la fonction objective. Tenter de solutionner en suivant la direction de croissance de la fonction objective. Que remarquez-vous pour chaque programme ? □

Leçon 8 : 3 octobre 2011

6.4. **Solution analytique : sommets.** La solution graphique de l'exemple 6.1 est raisonnable. La difficulté est que le nombre de dimensions est en générale égale au nombre de variables.

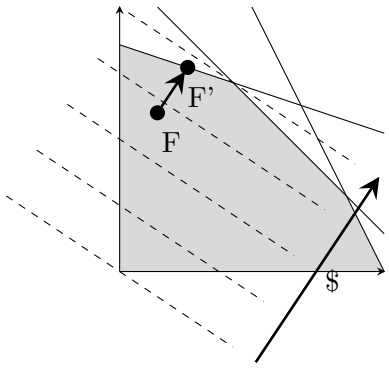
Une contrainte linéaire en n variables donne en général un *hyperplan* : une surface de $n-1$ dimensions. Donc en $\mathbb{R}^2$, on a des droites comme on a vu. En $\mathbb{R}^3$ les contraintes donnent des plans. De même avec les surfaces d'objectif constant : on a une série de plans parallèles en $\mathbb{R}^3$.

L'idée de la solution graphique reste valide, donc on s'inspire pour développer quelques idées.

On observe que la solution optimale s'est produit à un sommet, c'est-à-dire l'intersection de deux droites. En générale, un sommet est l'intersection de n hyperplans en $\mathbb{R}^n$: donc deux droites en $\mathbb{R}^2$, trois plans en $\mathbb{R}^3$, etc. [1]

Proposition 6.10. *Si un programme linéaire possède une solution optimale, alors, parmi toutes les solutions optimales, on peut trouver une solution optimale à un sommet.* □

Démonstration. Voici une idée de la preuve. Considérons un point F strictement à l'intérieur de la région faisable d'un programme linéaire. Puisque c'est strictement à l'intérieur, on pourra déplacer F dans la direction de croissance de l'objectif pour obtenir le point faisable F'. Donc F n'est pas une solution optimale, car le point F' est meilleur.



On conclut que les points intérieurs ne sont jamais optimales. En allant vers F' on atteint une des contraintes. Dans cet exemple, la contrainte atteinte n'est pas parallèle aux droites de valeur constant de l'objectif, donc on peut continuer à augmenter jusqu'au sommet P. Si la contrainte aurait été parallèle, on aurait pu continuer à gauche ou à droite en conservant l'objectif, donc on aurait trouvé un sommet ayant la même valeur objective que F'. □

On a donc l'algorithme suivant pour solutionner un programme linéaire.

Algorithme 6.11. Soit un programme linéaire qui possède une solution optimale. On peut la trouver comme suit.

(a) Déterminer tous les sommets.

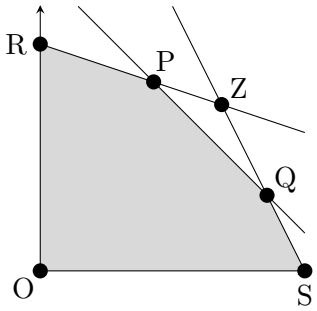
(b) Évaluer la fonction objective à chaque sommet de la région faisable.

(c) Choisir une des sommets qui donne la valeur optimale. □

En $\mathbb{R}^2$, un sommet est l'intersection de deux droites qui est aussi un point faisable. Donc on calcule l'intersection de chaque paire de droites et on élimine les points non-faisables.

1. On exige de plus que les hyperplans sont en *position générale*, ou autrement dit, indépendants. Donc on ne considère pas l'intersection de deux droites parallèles, ni de trois plans qui partageant une droite, etc.

Exemple 6.12. On retourne à l'exemple 6.1. On identifie les sommets : P, Q, R et S. Le point Z est l'intersection de deux contraintes, mais ce n'est pas un sommet faisable. On calcule l'objectif à chaque sommet, et on voit que P est optimale parmi les sommets, donc c'est une solution optimale du programme linéaire.



point	coord.	objectif
P	$(15, 25)$	\$2100
Q	$(30, 10)$	\$1800
R	$(0, 30)$	\$1800
S	$(35, 0)$	\$1400
O	$(0, 0)$	\$0
Z	$(24, 22)$	pas un sommet

Exercice 6.13. Vérifier les coordonnées des point P, Q, R, S, et Z, ainsi que les calculs de l'objectifs à chacun. Aussi, vérifier que Z n'est pas dans la région faisable en vérifiant les contraintes $A\mathbf{x} \leq \mathbf{b}$ pour $\mathbf{x} = Z$. □

Exercice 6.14. Solutionner chaque programme linéaire de l'exemple 6.3 en utilisant l'algorithme 6.11. Comparer avec vos résultats de l'exercice 6.9. Que remarquez-vous pour chaque programme ? □

6.5. **Difficultés.** Solutionner un programme linéaire en suivant algorithme 6.11 est meilleur que faire un graphique, mais il reste encore des difficultés. En $\mathbb{R}^2$, on voit que la région faisable est un polygone, donc il y a au plus p sommets, où p représente le nombre de contraintes. En $\mathbb{R}^n$, le nombre de sommets peut être exponentiel : même faire une liste est quasi-impossible. Il nous faut une méthode qui ne considère pas *chaque* sommet, mais plutôt une méthode qui *choisit* un sommet optimale.

Il y a une autre difficulté : afin d'appliquer l'algorithme 6.11, il faut savoir qu'il y a une solution optimale. Il faudrait donc détecter ceci.

Vous avez déjà vu un exemple d'un programme linéaire qui n'a pas de solution optimale parmi l'exemple 6.3. Cette situation correspond à une région faisable qui n'est pas borné dans la direction de croissance de l'objective.

Vous avez aussi vu un exemple d'un programme linéaire qui n'a pas de solution unique. En générale, l'ensemble des solutions optimales est soit un point (unique !), un segment de droite, un segment d'un plan, etc.

7. Méthode simplex

7.1. **Introduction.** On cherche un algorithme qui pourra trouver un sommet optimal, sans devoir tester chaque sommet. C'est parce que un programme linéaire peut avoir un nombre énorme de sommets : on trouve aisément des programmes linéaires dont la région faisable possède plus de sommets que le nombre d'électrons dans l'univers.

7.2. **Transformer en tableau.** On considère un programme linéaire en forme canonique :

$$\max \mathbf{c}^T\mathbf{x} \quad \text{s.c.} \quad A\mathbf{x} \leq \mathbf{b},\ \mathbf{x} \geq \mathbf{0}.$$

Donc on présume que chaque inégalité est écrit du forme "variables au plus petite ou égale à constante". On dénote par n le nombre de variables et p le nombre de contraintes (pas incluant les contraintes $\mathbf{x} \geq \mathbf{0}$). Donc A est de taille $n \times p$.

De plus, on suppose que le vecteur $\mathbf{b}$ est complètement non-négatif, c'est-à-dire que chaque constante à droite est non-négative. C'est une restriction technique : on verra bientôt comment traiter les autres cas.

La première étape consiste en transformer chaque inégalité en égalité. Ceci s'accomplit en introduisant une nouvelle variable auxiliaire pour chaque contrainte. Donc pour la contrainte i (la rangée i de $A\mathbf{x} \leq \mathbf{b}$) on obtient

$$a_{i1}x_1 + a_{i2}x_2 + \cdots a_{in}x_n \leq b_i \quad \rightsquigarrow \quad a_{i1}x_1 + a_{i2}x_2 + \cdots a_{in}x_n + y_i = b_i$$

On exige que $y_i \geq 0$ pour $1 \leq i \leq p$. On a maintenant un système avec $n + p$ variables et p égalités.

On introduit une autre variable M pour l'objectif. Ce n'est pas une variable "libre", car on connaît que $M = \mathbf{c}^T\mathbf{x}$. En forme standard, on écrit $M - \mathbf{c}^T\mathbf{x} = \mathbf{0}$.

On a maintenant $n+p+1$ variables et $p+1$ équations. On cherche une solution de ce système qui maximise la valeur de la variable M. On peut donner la matrice augmentée de ce système comme pour n'importe quel autre. Le voici en schéma :

$$\left[\begin{array}{c||c|c|c} 1 & -\mathbf{c}^T & \mathbf{0}^T & 0 \\ \hline\hline \mathbf{0} & A & I & \mathbf{b} \end{array}\right] \tag{7.1}$$

On observe que ce tableau possède $n + p + 1$ variables, $p + 1$ rangées et $p + 1$ pivots, donc n variables libres. Cette matrice est appelée le *tableau de simplex*.

Ce serait utile de considérer un exemple !

Exemple 7.1. On se rappelle l'exemple 6.1.

$$\max 40x_1 + 60x_2 \quad \text{s.c.} \quad \begin{cases} 2x_1 + x_2 \leq 70 \\ x_1 + x_2 \leq 40 \\ x_1 + 3x_2 \leq 90 \end{cases} \qquad x_1, x_2 \geq 0$$

On a $n = 2$ variables et $p = 3$ contraintes ; on introduit alors $p = 3$ variables $y_1, y_2, y_3 \geq 0$ pour obtenir

$$\begin{aligned} 2x_1 + x_2 + y_1 &= 70 \\ x_1 + x_2 + y_2 &= 40 \\ x_1 + 3x_2 + y_3 &= 90 \end{aligned}$$

On écrit la fonction objective comme $M - 40x_1 - 60x_2 = 0$. On a donc le tableau suivant. La première rangée représente l'équation de l'objectif, les autres correspondent aux contraintes. Les colonnes correspondent aux variables : les vraies variables, les variables auxiliaires, et la valeur de l'objectif. Voici les équations :

$$\begin{array}{cccccccc} M & -40x_1 & -60x_2 & & & & & = 0 \\ & 2x_1 & +x_2 & +y_1 & & & & = 70 \\ & x_1 & +x_2 & & +y_2 & & & = 40 \\ & x_1 & +3x_2 & & & +y_3 & & = 90 \end{array}$$

On a le tableau de simplex du programme linéaire original, qui n'est rien d'autre que la matrice augmentée de ce nouveau système. On indique aussi, comme aide visuel, les variables qui correspondent à chaque colonne.

$$\begin{array}{c} \begin{array}{ccccccc} M & x_1 & x_2 & y_1 & y_2 & y_3 & \end{array} \\ \left[\begin{array}{c|cccccc} 1 & -40 & -60 & 0 & 0 & 0 & 0 \\ \hline 0 & 2 & 1 & 1 & 0 & 0 & 70 \\ 0 & 1 & 1 & 0 & 1 & 0 & 40 \\ 0 & 1 & 3 & 0 & 0 & 1 & 90 \end{array}\right] \end{array} \tag{7.2}$$

Comparer ceci avec la forme générale de l'équation (7.1) : trouver la matrice A, I, $\mathbf{c}^T$, $\mathbf{b}$. On observe que dans cet exemple les constantes (la dernière colonne) sont non-négatives. □

7.3. **Sommets et pivots : trouver l'optimum.** Observons que le tableau de simplex est en forme échelonnée, pourvu qu'on se permet de permuter les colonnes. Les pivots de l'équation (7.2) sont des les colonnes 1, 4, 5 et 6. Donc les variables libres sont x_1 et x_2. Il y a exactement $n = 2$ variables libres. De façon générale, il y aurait toujours exactement n variables libres.

On cherche une solution qui correspond à un sommet, autrement dit, une solution qui correspond à avoir une égalité dans n des contraintes : soit les p contraintes du problème ou les n contraintes de la forme $x_j \geq 0$. Mais une égalité dans une des contraintes originales correspond exactement à avoir une des variables auxiliaires égale à zéro. On a le résultat important suivant :

Proposition 7.2. *Un sommet de la région faisable est exactement une solution du tableau de simplex avec n des variables (réelles ou auxiliaires) égales à zéro.*

Donc on interprète le tableau de simplex comme suit. Les pivots indiquent les variables de base, et les variables libres (les n variables libres) seront zéro. Dans le tableau actuel, on lit la solution suivante :

$$x_1 = x_2 = 0$$
$$M = 0, y_1 = 70, y_2 = 40, y_3 = 90$$

C'est une solution faisable : fabriquer rien du tout pour un profit nul. Mais ce n'est pas optimal. Comment l'améliorer ?

On cherche une solution parmi les sommets, donc parmi les solutions ayant n variables nulles. L'idée c'est de changer les pivots pour permettre un meilleur choix. Autrement dit, on doit choisir une des variables libres, et modifier le tableau pour que sa colonne possède un pivot. Autrement dit, on doit choisir une des variables qui est zéro dans la solution actuelle, et l'augmenter.

Exercice 7.3. Pourquoi est-ce qu'on ne peut pas diminuer une des variables qui est zéro dans la solution actuelle ? □

Comment choisir ?

La première rangée donne l'équation de M. On lit que $M-40x_1-60x_2 = 0$ ou $M = 40x_1+60x_2$ (attention au signes !). On a le choix d'augmenter x_1 ou x_2 (correspondant aux valeurs négatives dans la première rangée). Augmenter x_2 a une plus forte influence sur M, donc on choisit d'augmenter x_2, ce qui équivaut à choisir la colonne de x_2 comme colonne qui aura un pivot. On identifie x_2 comme la *variable qui entre* dans la solution : c'est la colonne qu'on choisit.

Il faut maintenant choisir la rangée, c'est-à-dire la position dans cette colonne où sera le pivot. L'idée c'est que, en augmentant x_2, il faut que les autres variables restent positives. Donc au pire la contribution de x_2 ne peut pas dépasser la constante dans la colonne **b**. En regardant les trois rangées correspondant aux contraintes, on voit qu'on a des limites en augmentant x_2 :

$$x_2 \leq 70$$
$$x_2 \leq 40$$
$$3x_2 \leq 90$$

Ces limites garantissent que x_2 reste faisable. La limite la plus restrictive est $3x_2 \leq 90$, donc c'est dans cette rangée qu'on aura le pivot.

Pour résumer, on a décider de faire à ce que la valeur en boîte devienne un pivot.

$$\left[\begin{array}{c||cccccc} 1 & -40 & -60 & 0 & 0 & 0 & 0 \\ \hline\hline 0 & 2 & 1 & 1 & 0 & 0 & 70 \\ 0 & 1 & 1 & 0 & 1 & 0 & 40 \\ 0 & 1 & \boxed{3} & 0 & 0 & 1 & 90 \end{array}\right]$$

Afin d'accomplir ceci, on multiplie la troisième rangée par 1/3, et on la soustrait des autres rangées. Voici le tableau qui en résulte.

$$\left[\begin{array}{c||cccccc|c} 1 & -20 & 0 & 0 & 0 & 20 & 1800 \\ \hline\hline 0 & \frac{5}{3} & 0 & 1 & 0 & -\frac{1}{3} & 40 \\ 0 & \frac{2}{3} & 0 & 0 & 1 & -\frac{1}{3} & 10 \\ 0 & \frac{1}{3} & 1 & 0 & 0 & \frac{1}{3} & 30 \end{array}\right]$$

De ce tableau on lit la solution de la même manière. On identifie les variables libres et on leur donne la valeur zéro. Les autres se lisent directement du tableau.

$$x_1 = y_3 = 0$$
$$M = 1800, x_2 = 30, y_1 = 40, y_2 = 10$$

C'est une solution préférable, car M a augmenté. Mais on peut faire mieux, en observant que dans la première rangée il y a encore une valeur négative. Donc on peut encore augmenter M en introduisant la variable x_1 dans la solution. En augmentant x_1 il faudrait avoir

$$\frac{5}{3}x_1 \leq 40$$
$$\frac{2}{3}x_1 \leq 10$$
$$\frac{1}{3}x_1 \leq 30$$

La condition la plus restrictive est la deuxième. Ceci indique la position du prochain pivot : colonne de x_1 et la rangée de $\frac{2}{3}x_1 \leq 10$.

$$\left[\begin{array}{c||cccccc|c} 1 & -20 & 0 & 0 & 0 & 20 & 1800 \\ \hline\hline 0 & \frac{5}{3} & 0 & 1 & 0 & -\frac{1}{3} & 40 \\ 0 & \boxed{\frac{2}{3}} & 0 & 0 & 1 & -\frac{1}{3} & 10 \\ 0 & \frac{1}{3} & 1 & 0 & 0 & \frac{1}{3} & 30 \end{array}\right]$$

On multiplie cette rangée par $\frac{3}{2}$ et ensuite on utilise le 1 pour annuler les autres éléments de la colonne. Ceci donne :

$$\left[\begin{array}{c||cccccc|c} 1 & 0 & 0 & 0 & 30 & 10 & 2100 \\ \hline\hline 0 & 0 & 0 & 1 & -\frac{5}{2} & \frac{1}{2} & 15 \\ 0 & 1 & 0 & 0 & \frac{3}{2} & -\frac{1}{2} & 15 \\ 0 & 0 & 1 & 0 & -\frac{1}{2} & \frac{1}{2} & 25 \end{array}\right]$$

Dans ce tableau on lit la solution

$$y_2 = y_3 = 0$$
$$M = 2100, x_1 = 15, x_2 = 25, y_1 = 15$$

C'est encore plus préférable, car M a augmenté.

Pour trouver une prochaine solution, on a l'option d'introduire la variable y_2 ou y_3. Mais les deux éléments correspondants dans la première rangée sont non-négatifs. Autrement dit, on a maintenant $M = 2100 - 30y_2 - 10y_3$. On voit alors que augmenter y_2 ou y_3 diminuerait la valeur de M.

On conclut que la solution actuelle est optimal.

Exercice 7.4. Dans le développement précédent, on a trouvé trois solutions (c'est-à-dire trois sommets). Vérifier que ces trois solutions sont, en ordre, les point O, R et P de l'exemple 6.12. □

Leçon 9 : 6 octobre 2011

7.4. **Conditions sur la variable qui entre.** Il se peut qu'un programme linéaire possède des solutions, mais ne possède aucune solution optimale. On verra comment la méthode de simplex permet de détecter ce phénomène.

Exercice 7.5. Pouvez-vous donner une raison pourquoi on pourrait ne pas avoir une solution optimal ? □

Exemple 7.6. Trouver la solution optimale de

$$\min -x_1 - x_2 \quad \text{s.c.} \quad \begin{cases} x_1 - x_2 \geq -1 \\ x_1 - 2x_2 \leq 4 \end{cases}, \quad \mathbf{x} \geq \mathbf{0}.$$

□

On commence en récrivant le programme linéaire en forme canonique :

$$\max x_1 + x_2 \quad \text{s.c.} \quad \begin{cases} -x_1 + x_2 \leq 1 \\ x_1 - 2x_2 \leq 4 \end{cases}, \quad \mathbf{x} \geq \mathbf{0}.$$

□

On note que $\mathbf{b} \geq \mathbf{0}$, donc on connaît comment solutionner. On obtient le tableau initial

$$\left[\begin{array}{c||cccc|c} 1 & -1 & -1 & 0 & 0 & 0 \\ \hline\hline 0 & -1 & 1 & 1 & 0 & 1 \\ 0 & 1 & -2 & 0 & 1 & 4 \end{array}\right].$$

La solution actuelle est

$$x_1 = x_2 = 0,$$
$$M = 0,\ y_1 = 1,\ y_2 = 4.$$

On a le choix d'augmenter x_1 ou x_2. Les valeurs correspondantes dans la première rangée sont égales, donc on choisira au hasard x_1. En augmentant x_1, on doit respecter les conditions

$$-x_1 \leq 1,$$
$$x_1 \leq 4.$$

Ici, on voit que la première "condition" n'est pas une condition du tout. On cherche à *augmenter* x_2. On peut l'augmenter autant qu'on veut, on aura toujours $-x_1 \leq 1$. Donc la seule condition restrictive est que $x_1 \leq 4$. C'est donc la rangée correspondante qui indique l'endroit du prochain pivot.

$$\left[\begin{array}{c||cccc|c} 1 & -1 & -1 & 0 & 0 & 0 \\ \hline\hline 0 & -1 & 1 & 1 & 0 & 1 \\ 0 & \boxed{1} & -2 & 0 & 1 & 4 \end{array}\right]$$

On fait des opérations de rangée pour que cette position devienne pivot :

$$\left[\begin{array}{c||cccc|c} 1 & 0 & -3 & 0 & 1 & 4 \\ \hline\hline 0 & 0 & -1 & 1 & 1 & 5 \\ 0 & 1 & -2 & 0 & 1 & 4 \end{array}\right]$$

La solution actuelle est

$$x_2 = y_2 = 0,$$
$$M = 4,\ x_1 = 4,\ y_1 = 5.$$

On peut augmenter M, car il y a une valeur négative dans la première rangée. Donc on augmente x_2. Les conditions restrictives sont

$$-x_2 \leq 5$$
$$-2x_2 \leq 4$$

Aucune de ces conditions impose une restriction sur x_2. On cherche à *augmenter* x_2 ; on peut augmenter à volonté ! Donc il n'y a aucune limite sur la valeur de M.

Conclusion : ce programme linéaire ne possède aucune solution optimale. Le programme linéaire original ne possède pas de solution non plus : on peut diminuer l'objectif original (le min) autant qu'on veut.

Si c'était un problème réel, alors on aurait probablement oublié une contrainte (normalement on ne pourrait pas produire une quantité illimité de quoi que ce soit). Mais ce problème, tel que présenté, ne possède aucune solution optimale.

Exercice 7.7. On a fait le choix d'augmenter x_1 au début. Refaire l'exemple avec le choix de x_2. Est-ce que le résultat final est pareille ? Est-ce que les résultats intermédiaires sont pareilles ? □

Exercice 7.8. Faire un graphique de ce programme linéaire, et tenter de trouver la solution optimale graphiquement. Est-ce que vous pouvez voir pourquoi il n'y a pas de solution optimal ? De plus, trouver les solutions intermédiaires trouvées par la méthode de simplex et expliquer graphiquement le fait qu'on peut "augmenter x_2 autant qu'on veut". □

7.5. **Constantes négatives.** Considérons l'exemple suivant.

Exemple 7.9. Trouver la solution optimale de

$$\min 2x_1 - 3x_2 \quad \text{s.c.} \quad \begin{cases} x_1 - x_2 \geq 1 \\ x_1 + x_2 \leq 2 \end{cases}, \quad \mathbf{x} \geq \mathbf{0}$$

□

On récrit en forme canonique pour obtenir

$$\max -2x_1 + 3x_2 \quad \text{s.c.} \quad \begin{cases} -x_1 + x_2 \leq -1 \\ x_1 + x_2 \leq 2 \end{cases}, \quad \mathbf{x} \geq \mathbf{0}$$

On observe une des constantes est négative. Voyons ce qui se passe. On forme le tableau initial.

$$\left[\begin{array}{c|cccc|c} 1 & 2 & -3 & 0 & 0 & 0 \\ \hline 0 & -1 & 1 & 1 & 0 & -1 \\ 0 & 1 & 1 & 0 & 1 & 2 \end{array}\right]$$

On lit la solution initiale.

$$x_1 = x_2 = 0$$
$$M = 0, y_1 = -1, y_2 = 2$$

Le problème ici, c'est que cette "solution" n'est pas faisable, car $y_1 < 0$. N'étant pas faisable, ce n'est certainement pas un sommet. Que faire ?

Exercice 7.10. Montrer que si $\mathbf{b} \geq \mathbf{0}$ alors la solution obtenue du tableau initial est toujours faisable. Montrer que si $\mathbf{b} \ngeq \mathbf{0}$ alors la solution obtenue du tableau initial n'est pas faisable. □

On comprend que la vraie raison d'exiger que $\mathbf{b} \geq \mathbf{0}$ est que c'est une façon de garantir que la solution initiale est faisable. Pour la méthode de simplex en forme canonique, la solution initiale est toujours l'origine ; ici l'origine n'est pas faisable. Donc le vrai problème c'est comment trouver une solution initiale, c'est-à-dire un sommet de la région faisable, afin de pouvoir démarrer la méthode de simplex.

Exercice 7.11. Vérifier que l'origine n'est pas un point faisable de l'exemple 7.9. □

On considère la contrainte problématique :

$$-x_1 + x_2 + y_1 = -1$$

Note qu'on veut que y_1 soit variable de base et $x_1 = x_2 = 0$, mais on ne peut pas, car on aura alors $y_1 < 0$. L'exemple est suffisamment "petite" que vous voyez peut-être la résolution, mais on cherche une méthode plutôt qu'une réponse, donc on continue de façon générale.

La solution c'est d'introduire une *variable artificielle*, qui est, comme les autres, non-négative.

$$-x_1 + x_2 + y_1 - s = -1$$

Si on exige que s soit variable de base on voit qu'il y a une solution : $s = 1$. La variable artificielle permet "d'absorber" la négativité. Mais le fait que $s \neq 0$ veut dire qu'on n'a pas une solution faisable au programme linéaire original. Donc on cherche une solution qui minimise s, c'est-à-dire qui maximise $-s$.

Afin de trouver une solution faisable au programme linéaire original, on résout le programme suivant :

$$\max -s \quad \text{s.c.} \quad \begin{cases} -x_1 + x_2 - s \leq -1 \\ x_1 + x_2 \leq 2 \end{cases}, \quad x_1, x_2, s \geq 0$$

On a donc le tableau suivant, mais on veut que s soit une variable de base, c'est-à-dire qu'on veut que la colonne de s possède un pivot.

$$\begin{array}{c} M' \quad x_1 \quad x_2 \quad y_1 \quad y_2 \quad s \\ \left[\begin{array}{c||ccccc|c} 1 & 0 & 0 & 0 & 0 & 1 & 0 \\ \hline\hline 0 & -1 & 1 & 1 & 0 & -1 & -1 \\ 0 & 1 & 1 & 0 & 1 & 0 & 2 \end{array}\right] \end{array}$$

(Ici, $M' = -s$ est la fonction qu'on veut maximiser.)

La colonne de s doit être ajusté pour que le pivot soit réellement un pivot.

$$\left[\begin{array}{c||ccccc|c} 1 & 0 & 0 & 0 & 0 & 1 & 0 \\ \hline\hline 0 & -1 & 1 & 1 & 0 & \boxed{-1} & -1 \\ 0 & 1 & 1 & 0 & 1 & 0 & 2 \end{array}\right] \quad \to \quad \left[\begin{array}{c||ccccc|c} 1 & -1 & 1 & 1 & 0 & 0 & -1 \\ \hline\hline 0 & 1 & -1 & -1 & 0 & 1 & 1 \\ 0 & 1 & 1 & 0 & 1 & 0 & 2 \end{array}\right]$$

Note qu'en exigeant que la colonne de s possède un pivot (au lieu de y_1) on rend la constante non-négative. On lit la solution initiale :

$$x_1 = x_2 = y_1 = 0,$$
$$M' = -1,\ y_2 = 2,\ s = 1.$$

On savait ceci déjà, n'est-ce pas ? (On se souvient que l'objectif M' est égale à $-s$ et non le "vrai" objectif.)

Les constantes sont toutes non-négatives : ce serait toujours le cas, car en faisant à ce que s devient variable de base on multiplie sa rangée par -1, qui "répare" la constante négative.

On peut maintenant faire une méthode simplex ordinaire. On identifie la variable x_1 comme variable qui entre, et on trouve la position du prochain pivot. Ensuite on fait des opérations de rangées pour que ça devienne pivot.

$$\left[\begin{array}{c||ccccc|c} 1 & -1 & 1 & 1 & 0 & 0 & -1 \\ \hline\hline 0 & \boxed{1} & -1 & -1 & 0 & 1 & 1 \\ 0 & 1 & 1 & 0 & 1 & 0 & 2 \end{array}\right] \to \left[\begin{array}{c||ccccc|c} 1 & 0 & 0 & 0 & 0 & 1 & 0 \\ \hline\hline 0 & 1 & -1 & -1 & 0 & 1 & 1 \\ 0 & 0 & 2 & 1 & 1 & -1 & 1 \end{array}\right]$$

On a la solution actuelle :

$$x_2 = y_1 = s = 0,$$
$$M' = 0,\ x_1 = 1,\ y_2 = 1.$$

Encore une fois, le M' représente l'objectif de $-s$. Note qu'on ne peut pas augmenter M', car il n'y a aucune valeur négative dans la première rangée. Par contre, on a atteint notre but : une solution ayant $s = 0$. On voit ceci de deux façons : la colonne de s ne possède pas de pivot, donc on peut la mettre égale à zéro. De plus, l'objectif $M' = -s$ a atteint la valeur de zéro.

Si on avait trouvé une solution optimale avec $M' < 0$ (ou $s > 0$) alors on aurait conclut qu'il n'existe aucune solution faisable avec $s = 0$. Autrement dit, le programme linéaire original n'aurait eu aucune solution faisable du tout : une région faisable vide !

Ici ce n'est pas le cas : on a trouvé une solution faisable. Elle se produit avec x_1 et y_2 comme variables de base. C'est exactement la raison d'avoir introduit une variable artificielle : pour trouver ce que *devrait* être les pivots dans le tableau initial. Donc on retourne au tableau initial, mais maintenant on exige que les pivots soient dans les colonnes indiqués par notre dernier tableau "artificielle", c'est-à-dire dans les colonnes de x_1 et y_2.

$$\left[\begin{array}{c|c|cccc|c} 1 & 2 & -3 & 0 & 0 & 0 \\ \hline 0 & -1 & 1 & 1 & 0 & -1 \\ 0 & 1 & 1 & 0 & 1 & 2 \end{array}\right] \to \left[\begin{array}{c|c|cccc|c} 1 & 0 & -1 & 2 & 0 & -2 \\ \hline 0 & 1 & -1 & -1 & 0 & 1 \\ 0 & 0 & 2 & 1 & 1 & 1 \end{array}\right]$$

On observe que toutes les constantes sont non-négatives. Ce serait toujours le cas : le fait d'avoir identifier les bonnes colonnes pour les variables de base garantit que les nouvelles constantes seront toutes non-négatives. On peut maintenant retourner à la méthode se simplex standard. On lit la solution initiale (qui est maintenant faisable !) :

$$x_2 = y_1 = 0$$
$$M = -2, x_1 = 1, y_2 = 1$$

Leçon 10 : 13 octobre 2011

On voit que c'est possible d'augmenter M en introduisant la variable x_2. Note qu'une des conditions restrictives sur x_2 n'est pas une restriction.

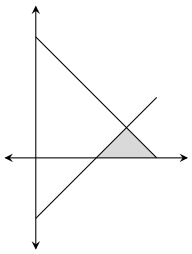
$$\left[\begin{array}{c|c|cccc|c} 1 & 0 & -1 & 2 & 0 & -2 \\ \hline 0 & 1 & -1 & -1 & 0 & 1 \\ 0 & 0 & \boxed{2} & 1 & 1 & 1 \end{array}\right] \to \left[\begin{array}{c|c|cccc|c} 1 & 0 & 0 & \frac{5}{2} & \frac{1}{2} & -\frac{3}{2} \\ \hline 0 & 1 & 0 & -\frac{1}{2} & \frac{1}{2} & \frac{3}{2} \\ 0 & 0 & 1 & \frac{1}{2} & \frac{1}{2} & \frac{1}{2} \end{array}\right]$$

On lit la solution actuelle.

$$y_1 = y_2 = 0$$
$$M = -\frac{3}{2}, x_1 = \frac{3}{2}, x_2 = \frac{1}{2}$$

On ne peut plus augmenter M, donc c'est une solution optimale.

Exercice 7.12. Voici un graphique du programme linéaire de l'exemple 7.9. Trouver sur ce graphique chaque solution intermédiaire qu'on a trouvé en résolvant ce programme linéaire, ainsi que la solution optimale.



Exercice 7.13. Solutionner le programme linéaire suivant en utilisant la méthode de simplex.

$$\min 4x_1 - 5x_2 \quad \text{s.c.} \quad \begin{cases} x_1 - x_2 \geq 2 \\ x_1 + x_2 \leq 1 \end{cases}, \quad \mathbf{x} \geq \mathbf{0}$$

Faire un graphique de ce programme linéaire. Bien indiquer la région faisable et les sommets. Indiquer sur ce graphique les solutions intermédiaires de votre méthode (méthodes ?) de simplex, ou alternativement, tenter d'expliquer de façon géométrique la solution obtenue. □

7.6. **Méthode de simplex : algorithme.** On résume avec une description algorithmique de la *méthode de simplex*. C'est une description formelle de ceux qu'on a découvert dans les exemples.

Algorithme 7.14 (Méthode de simplex avec $\mathbf{b} \geq \mathbf{0}$)**.** On commence avec

$$\max \mathbf{c}^T\mathbf{x} \quad \text{s.c.} \quad A\mathbf{x} \leq \mathbf{b},\ \mathbf{x} \geq \mathbf{0}$$

avec $\mathbf{b} \geq \mathbf{0}$.

On forme le tableau initial de simplex $\begin{bmatrix} 1 & -\mathbf{c}^T & \mathbf{0}^T & 0 \\ \mathbf{0} & A & I & \mathbf{b} \end{bmatrix}$.

On répète ensuite les étapes suivantes.

(a) La solution actuelle est obtenue en mettant chaque variable libre égale à zéro, et en lisant la solution pour les variables de base (celles avec pivot), y inclus la variable M.

(b) Si chaque valeur dans la première rangée est non-négatif (sauf peut-être la dernière), on ne peut pas augmenter M. On arrête : la solution actuelle est optimale.

(c) Sinon, on identifie la valeur la plus négative : c'est la colonne du prochain pivot, et c'est la variable qu'on va introduire dans la solution, i.e., la variable qu'on va augmenter.

(d) On identifie les limites imposés sur cette variable. Ceci ce fait en ignorant toute autre variable et en lisant chaque rangée comme une inégalité. La condition la plus restrictive donne la rangée du prochain pivot.

(e) S'il n'y a aucune condition restrictive alors on pourra augmenter M autant qu'on veut. On arrête : il n'y a aucune solution optimale.

(f) On connaît maintenant la position du prochain pivot. On multiplie cette rangée par une constante afin que le pivot devienne 1, et on l'utilise pour annuler les autres éléments dans sa colonne. □

Si $\mathbf{b} \ngeq \mathbf{0}$, c'est que l'origine n'est pas une solution faisable (et donc pas un sommet). On présente ici un algorithme qui commence en trouvant une solution faisable, c'est-à-dire un sommet. On peut la modifier pour obtenir une méthode itérative qui fonctionne pour plusieurs constantes négatives, mais on la présentera ici pour une seule valeur négative.

Algorithme 7.15 (Méthode de simplex avec $\mathbf{b} \ngeq \mathbf{0}$)**.** On commence avec

$$\max \mathbf{c}^T\mathbf{x} \quad \text{s.c.} \quad A\mathbf{x} \leq \mathbf{b},\ \mathbf{x} \geq \mathbf{0}.$$

(a) Introduire une variable artificielle s pour la valeur négative dans $\mathbf{b}$. Mettre $M' = -s$.

(b) Former le tableau initial avec les variables ordinaires, auxiliaires et la variable artificielle et la fonction objective $M' = -s$. Utiliser des opérations de rangées afin que la variable artificielle devienne pivot.

(c) Faire maintenant une méthode de simplex ordinaire.

(d) Au cas où la solution optimale donne $M' < 0$, il n'existe aucune solution faisable avec $s = 0$, donc aucune solution faisable du programme linéaire original. On arrête, car il n'y a aucune solution du tout.

(e) Au cas où la solution optimale donne $M' = 0$, on a trouvé une solution faisable avec $s = 0$. Noter les colonnes avec pivot : ce sont les variables de base désirées.

(f) Reprendre le tableau du programme original, avec la "vraie" fonction objective. Faire des opérations de rangée pour que les pivots soient dans les colonnes des variables de base désirées, celles indiquées par le tableau final de la première méthode de simplex.

(g) On a maintenant un tableau de simplex avec toutes les constantes non-négatives. Autrement dit, on a trouvé une solution initiale, c'est-à-dire, un sommet. On peut donc faire une méthode de simplex ordinaire.

7.7. **Entraînement.**

Exercice 7.16. Solutionner chaque programme linéaire avec la méthode simplex. Ensuite, tracer un graphique afin de repérer les solution intermédiaires et/ou expliquer le résultat de façon géométrique.

(a) $\max x_1 + 3x_2 \quad \text{s.c.} \quad \begin{cases} x_1 + x_2 \leq 10 \\ x_2 \leq 8 \\ x_1 - x_2 \leq 0 \end{cases}$

(b) $\min -2x_1 + x_2 \quad \text{s.c.} \quad \begin{cases} x_1 + x_2 \leq 10 \\ x_2 \leq 8 \\ x_1 - x_2 \leq 0 \end{cases}$

(c) $\max x_1 + 3x_2 \quad \text{s.c.} \quad \begin{cases} x_2 \geq x_1 - 1 \\ x_2 \leq 2x_1 + 3 \\ x_1 + x_2 \geq 2 \end{cases}$

(d) $\min x_1 + 3x_2 \quad \text{s.c.} \quad \begin{cases} x_2 \geq x_1 - 1 \\ x_2 \leq 2x_1 + 3 \\ x_1 + x_2 \geq 2 \end{cases}$

8. Programmes linéaires et dualité

8.1. **Définition.** Soit le programme linéaire suivant en forme canonique.

$$\mathcal{P}: \quad \max \mathbf{c}^T\mathbf{x} \quad \text{s.c.} \quad A\mathbf{x} \leq \mathbf{b},\ \mathbf{x} \geq \mathbf{0}.$$

On définit le programme linéaire *dual* comme suit :

$$\mathcal{P}^*: \quad \min \mathbf{b}^T\mathbf{y} \quad \text{s.c.} \quad A^T\mathbf{y} \geq \mathbf{c},\ \mathbf{y} \geq \mathbf{0}.$$

Exemple 8.1. Voici le programme de l'exemple 6.1 :

$$\mathcal{P}: \quad \max 40x_1 + 60x_2 \quad \text{s.c.} \quad \begin{bmatrix} 2 & 1 \\ 1 & 1 \\ 1 & 3 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \leq \begin{bmatrix} 70 \\ 40 \\ 90 \end{bmatrix},\ x_1, x_2 \geq 0.$$

Le dual du programme est donc

$$\mathcal{P}: \quad \min 70y_1 + 40y_2 + 90y_3 \quad \text{s.c.} \quad \begin{bmatrix} 2 & 1 & 1 \\ 1 & 1 & 3 \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix} \geq \begin{bmatrix} 40 \\ 60 \end{bmatrix},\ y_1, y_2, y_3 \geq 0.$$

□

On se rappelle que c'est toujours possible de récrire un programme linéaire en forme canonique, donc on peut en principe donner le dual de n'importe quel programme.

8.2. **Dualité et optimalité.** Le dual est fortement relié au programme original (parfois on dit *primal* pour celui-ci). On écrit $\mathcal{F}$ pour la région faisable du primal et $\mathcal{F}^*$ pour la région faisable du dual. Formellement,

$$\mathcal{F} = \{\mathbf{x} \mid A\mathbf{x} \leq \mathbf{b},\ \mathbf{x} \geq \mathbf{0}\}$$
$$\mathcal{F}^* = \{\mathbf{y} \mid A^T\mathbf{y} \geq \mathbf{c},\ \mathbf{y} \geq \mathbf{0}\}$$

Proposition 8.2. *Soit $\mathcal{P}$ et $\mathcal{P}^*$ comme ci-haut. Si $\mathbf{x}$ est un point de $\mathcal{F}$ et $\mathbf{y}$ est un point de $\mathcal{F}^*$, alors $\mathbf{c}^T\mathbf{x} \leq \mathbf{b}^T\mathbf{y}$.*

L'idée du preuve repose sur la multiplication de matrices et les transposes :

$$\mathbf{c}^T\mathbf{x} = \mathbf{x}^T\mathbf{c} \leq \mathbf{x}^TA^T\mathbf{y} = \mathbf{y}^TA\mathbf{x} \leq \mathbf{y}^T\mathbf{b} = \mathbf{b}^T\mathbf{y}.$$

Donc $\mathbf{c}^T\mathbf{x} \leq \mathbf{b}^T\mathbf{y}$. On a utiliser ici le fait que la transposée d'une matrice de taille 1×1 est égale à elle-même. Par exemple, $\mathbf{c}^T\mathbf{x}$ est une matrice de taille 1×1. Donc $\mathbf{c}^T\mathbf{x} = (\mathbf{c}^T\mathbf{x})^T = \mathbf{x}^T\mathbf{c}$. (Rappelez-vous que $(AB)^T = B^TA^T$ pour deux matrices A et B telles que AB est définit.)

Leçon 11 : 17 octobre 2011

Exemple 8.3. Pour l'exemple 8.1, on vérifie que $\mathbf{y} = (30, 30, 30)$ est un point de $\mathcal{F}^*$ (un point faisable du dual). On calcule que pour ce point, $\mathbf{b}^T\mathbf{y} = 70(30) + 40(30) + 90(30) = 6000$. Donc pour le primal, on sait que le maximum de $\mathbf{c}^T\mathbf{x}$ est au plus 6000. Pour cet example, on connaît mieux, mais le principe est qu'un point faisable pour le dual donne une borne sur la valeur de l'objectif primal.

Alternativement, on pourrait prendre le point $\mathbf{x} = (10, 10)$ comme point de $\mathcal{F}$. On aura $\mathbf{x}^T\mathbf{c} = 1000$. Donc pour le dual on sait que le minimum de $\mathbf{b}^T\mathbf{y}$ est au moins 1000.

Exercice 8.4. Trouver des autres points faisable du dual de l'exemple 8.1, et calculer la valeur de l'objectif dual pour ces points. Conclure que la valeur de l'objectif primal serait toujours au plus ces valeurs. Tenter de trouver un point qui donne un objectif aussi petit que possible.

Aussi, trouver des points faisable du primal de l'exemple 8.1, pour donner des bornes sur le dual. □

On voit que, en principe, la meilleure réponse à l'exercice précédant est de trouver les points faisables optimales.

Exemple 8.5. On peut vérifier que pour l'exemple 8.1, $\mathbf{x} = (15, 25)$ est un point faisable de $\mathcal{P}$ et que $\mathbf{y} = (0, 30, 10)$ est un point faisable de $\mathcal{P}^*$. Les valeurs des deux fonctions objectives à ces points sont les deux 2100. En se référant à la proposition 8.2, on peut conclure que ces points donnent les solutions optimales aux deux programmes linéaires. □

Le résultat de l'exercice précédant est typique.

Théorème 8.6. *Soit les deux programmes linéaires $\mathcal{P}$ et $\mathcal{P}^*$ ci-haut. Il y a exactement quatre possibilités :*

(a) Les deux régions faisables $\mathcal{F}$ et $\mathcal{F}^*$ sont non-vides. C'est-à-dire qu'il existe des points faisables pour le primal et pour le dual. Dans ce cas, le maximum M de $\mathcal{P}$ existe, le minimum m de $\mathcal{P}^*$ existe et $M = m$.

(b) La région faisable $\mathcal{F}$ est non-vide et $\mathcal{F}^*$ est vide. Dans ce cas, le maximum de $\mathcal{P}$ n'existe pas.

(c) La région faisable $\mathcal{F}^*$ est non-vide et $\mathcal{F}$ est vide. Dans ce cas, le minimum de $\mathcal{P}^*$ n'existe pas.

(d) Les deux régions faisable $\mathcal{F}$ et $\mathcal{F}^*$ sont vides. Il n'y a aucun point faisable pour ni l'un ni l'autre des deux programmes linéaires. □

La dernière possibilité n'est typiquement pas intéressant : il n'y a rien à optimiser car rien n'est possible. Les deuxième et troisième possibilités sont parfois utiles dans le sens contraire : on peut prouver qu'un programme linéaire ne possède aucun point faisable du tout en montrant que son dual ne possède pas de solution optimale. Mais typiquement c'est la première possibilité qui nous intéresse.

Si les deux régions faisables ne sont pas vides, alors les deux programmes ont la même solution optimale. On peut donc choisir de solutionner soit le primal ou le dual.

8.3. **Dualité et simplex.** Il y a une conséquence pratique du théorème 8.6. Ayant trouvé une solution optimale du primal, on connaît la valeur optimale de la fonction objective du dual aussi. On résout deux programmes à la fois. Mais comment trouver les valeurs des *variables* dual ?

Soit le tableau de simplex original d'un programme linéaire, et le tableau final :

$$\left[\begin{array}{c||c|c|c} 1 & -\mathbf{c}^T & \mathbf{0}^T & 0 \\ \hline\hline \mathbf{0} & A & I & \mathbf{b} \end{array}\right] \quad \to \cdots \to \quad \left[\begin{array}{c||c|c|c} 1 & \mathbf{d}^T & \mathbf{u}^T & M \\ \hline\hline \mathbf{0} & B_1 & B_2 & \mathbf{s} \end{array}\right]$$

Les matrices B_1 et B_2 sont les résultats finals des opération de rangées. Le vecteur $\mathbf{s}$ donne la solution optimale pour les variables de base (afin de connaître quelles variables, il faudrait connaître B_1 et B_2). La valeur M donne le maximum de l'objectif.

Qu'en est-il de $\mathbf{d}$ et $\mathbf{u}$? Ce sont des vecteurs non-négatifs (pourquoi ?). On peut montrer que $\mathbf{d}^T = -\mathbf{c}^T + \mathbf{u}^T A$. Donc

$$0 \leq \mathbf{d} \leq \left(-\mathbf{c}^T + \mathbf{u}^T A\right)^T = -\mathbf{c} + A^T \mathbf{u}$$

et $A^T \mathbf{u} \geq \mathbf{c}$. On peut aussi montrer que $\mathbf{b}^T \mathbf{u} = M$. Donc $\mathbf{u}$ est un vecteur faisable pour le dual qui donne la même valeur objective que la valeur optimale du primal. On résume :

Théorème 8.7. *Soit le tableau de simplex original et final comme ci-haut, d'un programme linéaire qui possède une solution optimale. Alors la solution optimale du programme dual est exactement* $\mathbf{y} = \mathbf{u}$*, avec la valeur objective* $m = M$*. C'est-à-dire, on peut lire la solution optimale du dual dans la première rangée, dans les colonnes correspondant aux variables auxiliaires.* □

Exemple 8.8. On se rappelle l'exemple 6.1. En particulier, voici le tableau de simplex original et final :

$$\left[\begin{array}{c||cccccc} 1 & -40 & -60 & 0 & 0 & 0 & 0 \\ \hline\hline 0 & 2 & 1 & 1 & 0 & 0 & 70 \\ 0 & 1 & 1 & 0 & 1 & 0 & 40 \\ 0 & 1 & 3 & 0 & 0 & 1 & 90 \end{array}\right] \quad \to \cdots \to \quad \left[\begin{array}{c||cccccc} 1 & 0 & 0 & 0 & 30 & 10 & 2100 \\ \hline\hline 0 & 0 & 0 & 1 & -\frac{5}{2} & \frac{1}{2} & 15 \\ 0 & 1 & 0 & 0 & \frac{3}{2} & -\frac{1}{2} & 15 \\ 0 & 0 & 1 & 0 & -\frac{1}{2} & \frac{1}{2} & 25 \end{array}\right]$$

Dans le tableau final, on lit la solution primal parmi les pivots : $x_1 = 15, x_2 = 25$. On trouve dans la première rangée la solution pour le dual, dans les colonnes des variables auxiliaires : $y_1 = 0, y_2 = 30, y_3 = 10$. □

Exercice 8.9. Vérifier que $x_1 = 15, x_2 = 25$ est solution faisable du primal, et que $y_1 = 0, y_2 = 30, y_3 = 10$ est solution faisable de dual. Vérifier aussi que ces deux solutions donnent la même valeur objective. □

8.4. **Dualité et dualité.** On note que le dual ressemble pas mal au primal. On voit aussi une symétrie dans le théorème 8.6. Ceci s'explique par le fait suivant :

Théorème 8.10. *Le dual du dual est le primal :* $(\mathcal{P}^*)^* = \mathcal{P}$.

Soit le primal

$$\mathcal{P} : \quad \max \mathbf{c}^T\mathbf{x} \quad \text{s.c.} \quad A\mathbf{x} \leq \mathbf{b},\ \mathbf{x} \geq \mathbf{0}$$

et le programme dual

$$\mathcal{P}^* : \quad \min \mathbf{b}^T\mathbf{y} \quad \text{s.c.} \quad A^T\mathbf{y} \geq \mathbf{c},\ \mathbf{y} \geq \mathbf{0}.$$

On peut exprimer le dual en forme canonique comme

$$\mathcal{P}^* : \quad \max -\mathbf{b}^T\mathbf{y} \quad \text{s.c.} \quad (-A)^T\mathbf{y} \leq (-\mathbf{c}),\ \mathbf{y} \geq \mathbf{0}.$$

Maintenant le dual du dual s'écrit comme

$$(\mathcal{P}^*)^* : \quad \min (-\mathbf{c})^T\mathbf{z} \quad \text{s.c.} \quad (-A)^T\mathbf{z} \geq (-\mathbf{b}),\ \mathbf{z} \geq \mathbf{0}.$$

On exprime ce dual du dual en forme canonique pour obtenir le programme linéaire originale.

$$(\mathcal{P}^*)^* = \mathcal{P} : \quad \max \mathbf{c}^T\mathbf{z} \quad \text{s.c.} \quad A\mathbf{z} \leq \mathbf{b},\ \mathbf{z} \geq \mathbf{0}$$

Exercice 8.11. Considérer le programme linéaire de l'exemple 7.9. Donner le programme linéaire dual.

On a déjà vu la solution de l'exemple 7.9 à l'aide de la méthode de simplex. En profiter pour "lire" la solution du dual. Vérifier que la solution est faisable pour le dual et aussi que l'objectif optimal dual (m) et égale à l'objectif optimal primal (M).

Faire une graphique du programme dual afin de vérifier les calculs. □

Exercice 8.12. Dans l'exemple 7.6 on a vu que il n'y avait aucune solution optimale, car c'était possible d'augmenter l'objectif sans limite. En vous référant au théorème 8.6, que peut-on conclure pour le dual, en particulier pour la région faisable du dual ? Donner le programme dual, et vérifier votre conclusion. □

Exercice 8.13. Pour les problèmes de l'exercice 7.16, écrire le dual. Ensuite, donner une solution optimale du dual, en vous référant à vos tableau finals de simplex pour les programmes primals. □

9. Projections

9.1. **Introduction.** On se dirige vers la solution approximative de systèmes linéaires. Un exemple commun est de trouver la meilleure droite étant donné un ensemble de points : c'est la *droite de régression*. On verra que l'approche est également applicable à des fonctions plus générales (eg, polynômes, exponentielles, ...).

L'outil fondamental serait la projection, et donc on commence en développant un peu de théorie des projections.

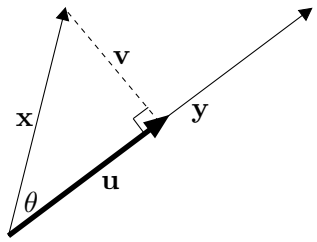
9.2. **Produit scalaire.** Le produit scalaire de deux vecteurs en $\mathbb{R}^n$ est définie comme

$$\mathbf{x} \cdot \mathbf{y} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} \cdot \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix} = x_1y_1 + x_2y_2 + \cdots + x_ny_n.$$

Quelques faits utiles du produit scalaire :
- La longueur d'un vecteur $\mathbf{x}$ est $\|x\| = \sqrt{\mathbf{x} \cdot \mathbf{x}}$. En deux dimensions c'est le théorème de Pythagore.
- C'est un produit linéaire, dans le sens qu'on peut enlever des facteurs communs : $(k\mathbf{x}) \cdot \mathbf{y} = k(\mathbf{x} \cdot \mathbf{y}) = \mathbf{x} \cdot (k\mathbf{x})$.
- Deux vecteurs $\mathbf{x}$ et $\mathbf{y}$ sont *orthogonaux* si l'angle entre $\mathbf{x}$ et $\mathbf{y}$ est $90°$; on écrit alors $\mathbf{x} \perp \mathbf{y}$. On peut détecter l'orthogonalité en calculant le produit scalaire, car $\mathbf{x} \perp \mathbf{y}$ si et seulement si $\mathbf{x} \cdot \mathbf{y} = 0$.
- De façon plus générale, on peut déterminer l'angle θ entre deux vecteurs, car $\mathbf{x} \cdot \mathbf{y} = \|x\| \, \|y\| \cos\theta$.

Donc le produit scalaire sert à mesurer et la longueur et la direction des vecteurs.

9.3. **Projection.** La projection d'un vecteur $\mathbf{x}$ sur un autre vecteur $\mathbf{y}$ est la partie de $\mathbf{x}$ qui est dans la direction de $\mathbf{y}$. On peut décomposer $\mathbf{x}$ en deux parties : l'une dans la direction de $\mathbf{y}$, l'autre orthogonal à $\mathbf{y}$.



Dans le graphique, le vecteur $\mathbf{u}$ représente la projection de $\mathbf{x}$ sur $\mathbf{y}$. On écrit $\mathbf{u} = \text{proj}_{\mathbf{y}}(\mathbf{x})$. Le vecteur $\mathbf{v}$ représente la partie de $\mathbf{x}$ qui est orthogonal à $\mathbf{y}$. On a $\mathbf{x} = \mathbf{u} + \mathbf{v}$. Sachant la projection, $\mathbf{u}$, on pourrait calculer la partie orthogonale comme $\mathbf{v} = \mathbf{x} - \mathbf{u}$.

Leçon 12 : 31 octobre 2011

On peut calculer la projection à l'aide de la formule suivante.

Proposition 9.1. *La projection de* $\mathbf{x}$ *sur* $\mathbf{y}$ *est* $\text{proj}_{\mathbf{y}}(\mathbf{x}) = \frac{\mathbf{x}\cdot\mathbf{y}}{\mathbf{y}\cdot\mathbf{y}}\mathbf{y}$. □

Exemple 9.2. Soit $\mathbf{x} = \begin{bmatrix}3\\6\end{bmatrix}$ et $\mathbf{y} = \begin{bmatrix}3\\2\end{bmatrix}$. On calcule la projection comme

$$\text{proj}_{\mathbf{y}}(\mathbf{x}) = \frac{\begin{bmatrix}3\\6\end{bmatrix}\cdot\begin{bmatrix}3\\2\end{bmatrix}}{\begin{bmatrix}3\\2\end{bmatrix}\cdot\begin{bmatrix}3\\2\end{bmatrix}}\begin{bmatrix}3\\2\end{bmatrix} = \frac{21}{13}\begin{bmatrix}3\\2\end{bmatrix} \approx \begin{bmatrix}4,8\\3,2\end{bmatrix}.$$

On peut alors calculer le "reste" de $\mathbf{x}$, c'est-à-dire la partie orthogonale à $\mathbf{y}$, en calculant

$$\mathbf{x} - \text{proj}_{\mathbf{y}}(\mathbf{x}) = \begin{bmatrix}3\\6\end{bmatrix} - \frac{21}{13}\begin{bmatrix}3\\2\end{bmatrix} \approx \begin{bmatrix}-1,8\\2,8\end{bmatrix}.$$

On obtient alors une décomposition de $\mathbf{x}$ en deux parties : la première dans la direction de $\mathbf{y}$ (c'est $\text{proj}_{\mathbf{y}}(\mathbf{x})$) et la deuxième perpendiculaire à $\mathbf{y}$:

$$\begin{bmatrix}3\\6\end{bmatrix} = \begin{bmatrix}63/13\\42/13\end{bmatrix} + \begin{bmatrix}-24/13\\36/13\end{bmatrix} \approx \begin{bmatrix}4,8\\3,2\end{bmatrix} + \begin{bmatrix}-1,8\\2,8\end{bmatrix}.$$

□

Exercice 9.3. Vérifier que la décomposition donnée dans l'exercice précédant a les bonnes directions : le premier vecteur devrait être un multiple de $\mathbf{y}$ et le deuxième vecteur devrait être perpendiculaire à $\mathbf{y}$. □

Exercice 9.4. Soit $\mathbf{x} = \begin{bmatrix}1\\1\\1\end{bmatrix}$ et $\mathbf{y} = \begin{bmatrix}2\\-2\\1\end{bmatrix}$. Calculer $\text{proj}_{\mathbf{y}}(\mathbf{x})$, et donner une décomposition de $\mathbf{x}$ en deux vecteurs : l'un dans la direction de $\mathbf{y}$ et l'autre perpendiculaire à $\mathbf{y}$.

Calculer aussi $\text{proj}_{\mathbf{x}}(\mathbf{y})$, et donner une décomposition de $\mathbf{y}$ en deux vecteurs : l'un dans la direction de $\mathbf{x}$ et l'autre perpendiculaire à $\mathbf{x}$. □

9.4. **Bases.** Une *base* d'un sous-espace U de $\mathbb{R}^n$ est un ensemble indépendant de vecteurs qui engendre U.

On peut aussi avoir des bases pour des sous-espaces : c'est un ensemble indépendant de vecteurs qui engendre le sous-espace.

On a déjà vu des exemples : on a vu comment trouver des bases pour des espaces propres d'une matrice. En prenant l'union de toutes les bases propres, on a pu trouver une base pour $\mathbb{R}^n$ (du moins dans le cas des matrices diagonalisables). C'était une base très utile pour étudier des systèmes dynamiques.

La *dimension* d'un sous-espace est le nombre de vecteurs dans une base pour le sous-espace. Par exemple, $\dim \mathbb{R}^n = n$.

Une base est utile car ça permet d'exprimer tout vecteur uniquement.

Théorème 9.5. *Soit* $\{\mathbf{v}_1, \mathbf{v}_2, \cdots, \mathbf{v}_k\}$ *une base pour un sous-espace* U*, et* $\mathbf{x}$ *n'importe quel vecteur dans* U*.*

Alors il existe des valeurs uniques $\alpha_1, \alpha_2, \cdots, \alpha_k$ *qui donnent*

$$\mathbf{x} = \alpha_1\mathbf{v}_1 + \alpha_2\mathbf{v}_2 + \cdots + \alpha_k\mathbf{v}_k$$

Ces valeurs sont les coordonnées *de* $\mathbf{x}$ *par rapport à la base* $\{\mathbf{v}_1, \ldots, \mathbf{v}_k\}$. □

Exemple 9.6. On calcule les coordonnées de $\begin{bmatrix} 3 \\ 4 \end{bmatrix}$ par rapport à la base $\left\{ \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 1 \end{bmatrix} \right\}$. On cherche alors des valeurs tel que

$$\begin{bmatrix} 3 \\ 4 \end{bmatrix} = \alpha_1 \begin{bmatrix} 1 \\ 0 \end{bmatrix} + \alpha_2 \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

Ici, on voit directement que $\alpha_1 = 3$ et $\alpha_2 = 4$.

On calcule les coordonnées de $\begin{bmatrix} 3 \\ 4 \end{bmatrix}$ par rapport à la base $\left\{ \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \begin{bmatrix} 1 \\ 1 \end{bmatrix} \right\}$. En générale, on aurait la matrice augmentée, qúon résout à l'aide de la méthode Gauss-Jordan

$$\left[\begin{array}{cc|c} 1 & 1 & 3 \\ 0 & 1 & 4 \end{array}\right] \quad \to \quad \left[\begin{array}{cc|c} 1 & 0 & -1 \\ 0 & 1 & 4 \end{array}\right].$$

Les coordonnées sont alors $\alpha_1 = -1$ et $\alpha_2 = 4$.

On observe que le même vecteur a des coordonnées différents par rapport à des bases différentes. De plus, le coordonnée qui a changé est le coordonnée qui correspond au vecteur du base qui n'a *pas* changé. Résumé : les coordonnées dépendent de la base *entière*. □

9.5. **Bases orthogonales.** L'orthogonalité n'est pas simplement une caractéristique géométrique : c'est une condition algébrique aussi.

Théorème 9.7. *Soit* $\{\mathbf{v}_1, \mathbf{v}_2, \cdots, \mathbf{v}_k\}$ *un ensemble orthogonal ; c'est-à-dire un ensemble avec* $\mathbf{v}_i \perp \mathbf{v}_j$ *pour tout* $1 \leq i < j \leq k$, *et* $\mathbf{v}_i \neq \mathbf{0}$ *pour tout* $1 \leq i \leq n$.

Alors $\{\mathbf{v}_1, \mathbf{v}_2, \cdots, \mathbf{v}_k\}$ *est en ensemble de vecteurs indépendent.* □

Note que ce n'est pas valide dans l'autre sens : un ensemble indépendant n'est pas nécessairement orthogonal.

La conséquence c'est que si on a n vecteurs orthogonaux dans $\mathbb{R}^n$, alors ce serait n vecteurs indépendants dans $\mathbb{R}^n$, et donc une base pour $\mathbb{R}^n$: une *base orthogonale*. De même, si on a k vecteurs orthogonaux dans un sous-espace U de dimension k, ce serait une base orthogonale pour le sous-espace U.

Exercice 9.8. Vérifier que $\left\{ \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 1 \end{bmatrix} \right\}$ est une base orthogonale pour $\mathbb{R}^2$. Vérifier que $\left\{ \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \begin{bmatrix} 1 \\ 1 \end{bmatrix} \right\}$ n'est pas une base orthogonale pour $\mathbb{R}^2$.

Est-ce que $\left\{ \begin{bmatrix} 2 \\ 2 \\ -1 \end{bmatrix}, \begin{bmatrix} -1 \\ 1 \\ 0 \end{bmatrix} \right\}$ est une base orthogonale pour un sous-espace de $\mathbb{R}^3$?

Exercice 9.9. Trouver des valeurs a, b tel que $\left\{ \begin{bmatrix} 1 \\ 1 \end{bmatrix}, \begin{bmatrix} a \\ b \end{bmatrix} \right\}$ soit une base orthogonale. Est-ce que les valeurs a, b sont uniques ? □

Une *base orthonormale* est une base orthogonale qui a aussi la propriété que chaque vecteur est de longueur 1. On peut transformer une base orthogonale à une base orthonormale en divisant chaque vecteur par sa norme.

Exercice 9.10. Vérifier que $\{\mathbf{v}_1, \mathbf{v}_2\} = \left\{ \begin{bmatrix} 2 \\ 2 \\ -1 \end{bmatrix}, \begin{bmatrix} -1 \\ 1 \\ 0 \end{bmatrix} \right\}$ est une base orthogonale pour le sous-espace engendré par ces deux vecteurs. □

Exemple 9.11. Pour l'exercice précédant, on trouve une base orthonormale.

On calcule que $\|\mathbf{v}_1\| = \sqrt{(2)^2 + (2)^2 + (-1)^2} = 3$ et que $\|\mathbf{v}_2\| = \sqrt{(-1)^2 + (1)^2 + (0)^2} = \sqrt{2}$. Donc on obtient la base orthonormale

$$\left\{ \frac{1}{3}\mathbf{v}_1, \frac{1}{\sqrt{2}}\mathbf{v}_2 \right\} = \left\{ \begin{bmatrix} 2/3 \\ 2/3 \\ -1/3 \end{bmatrix}, \begin{bmatrix} -1/\sqrt{2} \\ 1/\sqrt{2} \\ 0 \end{bmatrix} \right\}$$

□

Si $\{\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_n\}$ est une base orthonormale, on a

$$\mathbf{v}_i \cdot \mathbf{v}_j = \begin{cases} 0, & i \neq j, \\ 1, & i = j. \end{cases}$$

Donc, si on prend une base orthonormale, et on met les vecteurs comme colonnes d'une matrice U, alors $U^T U = I$. Une telle matrice est dite *matrice orthogonale*. [2]

9.6. **Bases orthogonales et projections.** On cherche à calculer des projections de manière efficace. Les bases orthogonales servent exactement à ceci. Notre but immédiat est d'obtenir une formule pour la projection d'un vecteur sur un sous-espace. On verra que c'est essentiellement la même chose que la projection d'un vecteur sur un autre vecteur.

Théorème 9.12. *Soit* $\{\mathbf{v}_1, \mathbf{v}_2, \cdots \mathbf{v}_n\}$ *une base orthogonale pour* $\mathbb{R}^n$*, et* $\mathbf{x}$ *n'importe quel vecteur dans* $\mathbb{R}^n$*. Alors*

$$\mathbf{x} = \frac{\mathbf{x} \cdot \mathbf{v}_1}{\mathbf{v}_1 \cdot \mathbf{v}_1}\mathbf{v}_1 + \frac{\mathbf{x} \cdot \mathbf{v}_2}{\mathbf{v}_2 \cdot \mathbf{v}_2}\mathbf{v}_2 + \cdots + \frac{\mathbf{x} \cdot \mathbf{v}_n}{\mathbf{v}_n \cdot \mathbf{v}_n}\mathbf{v}_n.$$

□

Autrement dit, $\mathbf{x}$ est égale à la somme des projections de $\mathbf{x}$ sur chaque vecteur de la base. C'est à comparer avec l'exemple 9.6 : on a $\alpha_j = \frac{\mathbf{x} \cdot \mathbf{v}_j}{\mathbf{v}_j \cdot \mathbf{v}_j}$.

Ce théorème s'applique aux sous-espaces aussi. Si on a un vecteur $\mathbf{x}$ dans $\mathbb{R}^n$ et un sous-espace, on pourrait demander quelle partie de $\mathbf{x}$ est "dans" le sous-espace. On voudrait trouver deux vecteurs, $\mathbf{u}$ et $\mathbf{v}$, tel que $\mathbf{u}$ est dans U, $\mathbf{v}$ est orthogonal à U, et $\mathbf{x} = \mathbf{u} + \mathbf{v}$.

"Orthogonal à U" veut dire que $\mathbf{v}$ est orthogonal à chaque vecteur dans U. On dit que le *complément orthogonal* de U, dit $U^\perp$, est l'ensemble des vecteurs qui sont orthogonaux à chaque vecteur de U.

$$U^\perp = \{\mathbf{y} \mid \mathbf{y} \perp \mathbf{x} \text{ pour tout } \mathbf{x} \in U\}.$$

La théorie des espaces orthogonaux donne la suivante.

Théorème 9.13. *Soit* U *un sous-espace quelconque de* $\mathbb{R}^n$*. Soit* $U^\perp$ *son complément orthogonal. Alors* $\dim(U) + \dim(U^\perp) = n$*. De plus, tout vecteur* $\mathbf{x}$ *peut s'écrire comme* $\mathbf{x} = \mathbf{u} + \mathbf{v}$*, où* $\mathbf{u} \in U$ *et* $\mathbf{v} \in U^\perp$*.* □

On reconnaît l'idée de la projection d'un vecteur sur une autre. D'ailleurs, le graphique ci-haut illustre la projection de $\mathbf{x}$ sur l'espace engendré par $\mathbf{y}$. Une autre conséquence est la suivante : le complément orthogonal est unique.

Corollaire 9.14. *Si* U *et* U' *sont deux sous-espaces de* $\mathbb{R}^n$ *avec* $\dim(U) + \dim(U') = n$*, et que chaque vecteur de* U *est orthogonal à chaque vecteur de* U'*, alors* $U' = U^\perp$*.* □

2. La terminologie est un peu bizarre : on aurait pensé "matrice orthonormale". Mais c'est comme ça !

Ceci se facilite avec des bases orthogonaux. On a trouvé la formule de projection sur un sous-espace, et aussi sur son complément orthogonal.

Théorème 9.15. *Soient* $\{\mathbf{u}_1, \mathbf{u}_2, \cdots, \mathbf{u}_k\}$ *une base orthogonale pour un sous-espace* U *de* $\mathbb{R}^n$*, et* $\{\mathbf{v}_1, \mathbf{v}_2, \cdots \mathbf{v}_{n-k}\}$ *une base orthogonale pour* $U^\perp$*. Alors*

$$\begin{aligned}
\mathbf{x} &= \text{proj}_U(\mathbf{x}) + \text{proj}_{U^\perp}(\mathbf{x}),\\
\text{proj}_U(\mathbf{x}) &= \frac{\mathbf{x}\cdot\mathbf{u}_1}{\mathbf{u}_1\cdot\mathbf{u}_1}\mathbf{u}_1 + \frac{\mathbf{x}\cdot\mathbf{u}_2}{\mathbf{u}_2\cdot\mathbf{u}_2}\mathbf{u}_2 + \cdots + \frac{\mathbf{x}\cdot\mathbf{u}_k}{\mathbf{u}_k\cdot\mathbf{u}_k}\mathbf{u}_k\\
&= \text{proj}_{\mathbf{u}_1}(\mathbf{x}) + \text{proj}_{\mathbf{u}_2}(\mathbf{x}) + \cdots + \text{proj}_{\mathbf{u}_k}(\mathbf{x}),\\
\text{proj}_{U^\perp}(\mathbf{x}) &= \frac{\mathbf{x}\cdot\mathbf{v}_1}{\mathbf{v}_1\cdot\mathbf{v}_1}\mathbf{v}_1 + \frac{\mathbf{x}\cdot\mathbf{v}_2}{\mathbf{v}_2\cdot\mathbf{v}_2}\mathbf{v}_2 + \cdots + \frac{\mathbf{x}\cdot\mathbf{v}_{n-k}}{\mathbf{v}_{n-k}\cdot\mathbf{v}_{n-k}}\mathbf{v}_{n-k}\\
&= \text{proj}_{\mathbf{v}_1}(\mathbf{x}) + \text{proj}_{\mathbf{v}_2}(\mathbf{x}) + \cdots + \text{proj}_{\mathbf{v}_{n-k}}(\mathbf{x}).
\end{aligned}$$

□

Leçon 13 : 3 novembre 2011

Exemple 9.16. Soit $\left\{\begin{bmatrix}2\\2\\-1\\0\end{bmatrix}, \begin{bmatrix}-1\\1\\0\\0\end{bmatrix}\right\}$ et $\left\{\begin{bmatrix}1\\1\\4\\0\end{bmatrix}, \begin{bmatrix}0\\0\\0\\1\end{bmatrix}\right\}$ des bases pour un espace U et son complément orthogonal, respectivement. Si $\mathbf{x} = \begin{bmatrix}1\\1\\1\\1\end{bmatrix}$, alors on peut calculer les deux projections comme

$$\text{proj}_U(\mathbf{x}) = \frac{3}{9}\begin{bmatrix}2\\2\\-1\\0\end{bmatrix} + \frac{0}{2}\begin{bmatrix}-1\\1\\0\\0\end{bmatrix} = \begin{bmatrix}2/3\\2/3\\-1/3\\0\end{bmatrix},$$

$$\text{proj}_{U^\perp}(\mathbf{x}) = \frac{6}{18}\begin{bmatrix}1\\1\\4\\0\end{bmatrix} + \frac{1}{1}\begin{bmatrix}0\\0\\0\\1\end{bmatrix} = \begin{bmatrix}1/3\\1/3\\4/3\\1\end{bmatrix}.$$

Note que la somme de ces deux projections donne $\mathbf{x}$. Donc, on aurait pu calculer une projection, et obtenir l'autre en soustrayant de $\mathbf{x}$ (e.g. $\text{proj}_{U^T}(\mathbf{x}) = \mathbf{x} - \text{proj}_U(\mathbf{x})$). □

Exercice 9.17. Vérifier que les deux bases données dans l'exemple précédant sont chacune des ensembles orthogonaux. Vérifier que chaque vecteur de la première est orthogonal à chaque vecteur de la deuxième. Conclure que les quatre vecteurs ensemble forment une base orthogonale pour $\mathbb{R}^4$. □

Exercice 9.18. Soit un ensemble orthogonal de k vecteurs dans $\mathbb{R}^n$, et un autre ensemble orthogonal de $n-k$ vecteurs dans $\mathbb{R}^n$, tel que chaque vecteur du premier ensemble est orthogonal à chaque vecteur du deuxième. Expliquer pourquoi la combinaison des deux donne une base orthogonale pour $\mathbb{R}^n$. □

Exercice 9.19. Pour les bases de l'exemple 9.16, trouver $\text{proj}_U(\mathbf{y})$ et $\text{proj}_{U^\perp}(\mathbf{y})$ pour $\mathbf{y} = \begin{bmatrix}1 & 0 & 1 & 0\end{bmatrix}^T$. Ensuite calculer $\mathbf{y} - \text{proj}_U(\mathbf{y})$ et comparer. □

Il reste deux questions techniques. Étant donné un sous-espace, comment trouver une base orthogonale ? Et aussi, comment trouver une base pour le complément orthogonale ?

9.7. **Gram-Schmidt.** On s'inspire de la première graphique de projection ci-haut. Les vecteurs $\mathbf{x}$ et $\mathbf{y}$ forment une base pour $\mathbb{R}^2$, mais pas une base orthogonale. On pourrait remplacer $\mathbf{x}$ avec la partie de $\mathbf{x}$ qui est orthogonal à $\mathbf{y}$, pour obtenir une base orthogonale. C'est-à-dire, $\{\mathbf{y}, \mathbf{x} - \text{proj}_{\mathbf{y}}(\mathbf{x})\}$ est une base orthogonale pour $\mathbb{R}^2$.

L'idée c'est de construire la base orthogonale un vecteur à la fois, chaque fois enlevant toutes les projections des vecteurs qui sont déjà dans la base orthogonale.

Algorithme 9.20 (Algorithme Gram-Schmidt)**.** Soit $\{\mathbf{v}_1, \mathbf{v}_2, \cdots, \mathbf{v}_k\}$ un ensemble qui engendre un sous-espace. On peut la transformer en base orthogonale comme suit.

On commence avec une base orthogonale vide, et on répète les étapes suivantes.

(a) Choisir un vecteur de l'ensemble donné.
(b) Soustraire de ce vecteur sa projection sur chaque vecteur qui est déjà dans la base orthogonale.
(c) Mettre le résultat (s'il est non-nul) dans la base orthogonale.

Exemple 9.21. Les vecteurs $\left\{ \begin{bmatrix}1\\1\\2\\0\end{bmatrix}, \begin{bmatrix}2\\0\\2\\0\end{bmatrix}, \begin{bmatrix}5\\1\\0\\2\end{bmatrix} \right\}$ forment une base pour un sous-espace. Trouver une base orthogonale pour ce sous-espace.

On commence avec la base orthogonale vide, donc $\{\}$.

On choisit un vecteur dans l'ensemble ; disons $\begin{bmatrix}1\\1\\2\\0\end{bmatrix}$. On soustrait sa projection sur les autres vecteurs dans la base orthogonale. Il n'y a rien dans la base orthogonale (à date !), donc on soustrait rien. On met le résultat dans la base orthogonale.

À date on a la base orthogonale : $\left\{ \begin{bmatrix}1\\1\\2\\0\end{bmatrix} \right\}$. Note que tout ce qu'on a fait c'était de mettre un vecteur dans la base orthogonale : le premier vecteur serait toujours ainsi.

On répète ! On choisit maintenant un vecteur ; disons $\begin{bmatrix}2\\0\\2\\0\end{bmatrix}$. On soustrait sa projection sur chaque vecteur qui est déjà dans la base orthogonale. Donc

$$\begin{bmatrix}2\\0\\2\\0\end{bmatrix} - \frac{\begin{bmatrix}2\\0\\2\\0\end{bmatrix}\cdot\begin{bmatrix}1\\1\\2\\0\end{bmatrix}}{\begin{bmatrix}1\\1\\2\\0\end{bmatrix}\cdot\begin{bmatrix}1\\1\\2\\0\end{bmatrix}}\begin{bmatrix}1\\1\\2\\0\end{bmatrix} = \begin{bmatrix}2\\0\\2\\0\end{bmatrix} - \frac{6}{6}\begin{bmatrix}1\\1\\2\\0\end{bmatrix} = \begin{bmatrix}1\\-1\\0\\0\end{bmatrix}.$$

On ajoute le résultat à la base orthogonale.

À date on a la base orthogonale : $\left\{\begin{bmatrix}1\\1\\2\\0\end{bmatrix}, \begin{bmatrix}1\\-1\\0\\0\end{bmatrix}\right\}$.

On choisit maintenant le dernier vecteur, $\begin{bmatrix}5\\1\\0\\2\end{bmatrix}$. On soustrait les projections :

$$\begin{bmatrix}5\\1\\0\\2\end{bmatrix} - \frac{\begin{bmatrix}5\\1\\0\\2\end{bmatrix}\cdot\begin{bmatrix}1\\1\\2\\0\end{bmatrix}}{\begin{bmatrix}1\\1\\2\\0\end{bmatrix}\cdot\begin{bmatrix}1\\1\\2\\0\end{bmatrix}}\begin{bmatrix}1\\1\\2\\0\end{bmatrix} - \frac{\begin{bmatrix}5\\1\\0\\2\end{bmatrix}\cdot\begin{bmatrix}1\\-1\\0\\0\end{bmatrix}}{\begin{bmatrix}1\\-1\\0\\0\end{bmatrix}\cdot\begin{bmatrix}1\\-1\\0\\0\end{bmatrix}}\begin{bmatrix}1\\-1\\0\\0\end{bmatrix} = \begin{bmatrix}5\\1\\0\\2\end{bmatrix} - \frac{6}{6}\begin{bmatrix}1\\1\\2\\0\end{bmatrix} - \frac{4}{2}\begin{bmatrix}1\\-1\\0\\0\end{bmatrix} = \begin{bmatrix}2\\2\\-2\\2\end{bmatrix}.$$

À date on a la base orthogonale : $\left\{\begin{bmatrix}1\\1\\2\\0\end{bmatrix}, \begin{bmatrix}1\\-1\\0\\0\end{bmatrix}, \begin{bmatrix}2\\2\\-2\\2\end{bmatrix}\right\}$. Il ne reste aucun vecteur dans l'ensemble original, donc l'algorithme se termine et c'est une base orthogonale pour l'espace engendré par les trois vecteurs originaux.

On se rappelle qu'en soustrayant les projections, on considère les projections sur les *nouveaux* vecteurs à date, et non les originaux. Aussi, on voit que puisqu'on obtient un ensemble orthogonal, alors c'est nécessairement un ensemble indépendant, donc une base pour l'espace engendré. Donc dans l'exemple 9.21 on a *prouvé* que l'ensemble original était une base. Sinon, en soustrayant on aurait obtenu un vecteur $\mathbf{0}$, qu'on aurait rejeté.

Exercice 9.22. Dans l'exemple 9.21, on a choisit les vecteurs dans un autre ordre particulier. Répéter l'exemple, mais avec un autre ordre. Est-ce qu'on obtient encore une base orthogonale ? Est-ce la même base ? □

Exercice 9.23. Dans l'exemple 9.21, on a trouvé une base de trois vecteurs orthogonaux. Vérifier directement que $\left\{\begin{bmatrix}1\\1\\2\\0\end{bmatrix}, \begin{bmatrix}1\\-1\\0\\0\end{bmatrix}, \begin{bmatrix}1\\1\\-1\\1\end{bmatrix}\right\}$ est aussi un ensemble orthogonal. □

On voit une principe parfois utile : on peut multiplier un vecteur par une constante (non-nul) sans changer sa direction. Donc si on a une base orthogonale on peut multiplier chaque vecteur par des constantes (non-nuls) et encore avoir une base orthogonale. Note que ce n'est pas la

même base : lorsqu'on multiplie un vecteur par $\frac{1}{2}$ ça change ! Mais c'est une autre base qui est encore orthogonale.

9.8. **Le complément orthogonal : compléter une base.** Sachant un sous-espace, comment trouver son complément orthogonal ? On peut accomplir ceci avec une réduction Gauss-Jordan.

Algorithme 9.24. Soit un ensemble $\{\mathbf{v}_1, \mathbf{v}_2, \cdots, \mathbf{v}_k\}$ qui engendre un sous-espace U de $\mathbb{R}^n$.

On met les vecteurs comme colonnes d'une matrice A, et on fait une méthode de Gauss sur la matrice $[A|I]$. Le résultat est $[R|B]$.

Les colonnes de R avec pivots indiquent quels vecteurs de l'ensemble on choisit pour la base du sous-espace U.

Les rangées de B correspondant aux rangées nulles de R donnent une base pour $U^\perp$.

Si on combine la base de U avec la base de $U^\perp$ on obtient une base de $\mathbb{R}^n$. On a "complété" la base de U à une base de $\mathbb{R}^n$. □

Exemple 9.25. Un sous-espace U est engendré par les vecteurs $\left\{ \begin{bmatrix} 1 \\ 0 \\ 2 \\ 1 \end{bmatrix}, \begin{bmatrix} 1 \\ 1 \\ 0 \\ 3 \end{bmatrix}, \begin{bmatrix} 2 \\ 1 \\ 2 \\ 4 \end{bmatrix} \right\}$. Trouver une base pour $U^\perp$.

On fait la réduction suivante.

$$\left[\begin{array}{cccc|cccc} 1 & 1 & 2 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 & 0 \\ 2 & 0 & 2 & 0 & 0 & 1 & 0 \\ 1 & 3 & 4 & 0 & 0 & 0 & 1 \end{array}\right] \to \cdots \to \left[\begin{array}{cccc|cccc} 1 & 1 & 2 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & -2 & 2 & 1 & 0 \\ 0 & 0 & 0 & -1 & -2 & 0 & 1 \end{array}\right]$$

À gauche, il y a des pivots dans la première et deuxième colonne. Donc on prend les vecteurs correspondantes (dans notre liste originale) comme base de U.

À gauche, il y a deux rangées nulles. Donc on prend les rangées correspondantes à droite comme base de $U^\perp$.

$$\text{base de } U : \left\{ \begin{bmatrix} 1 \\ 0 \\ 2 \\ 1 \end{bmatrix}, \begin{bmatrix} 1 \\ 1 \\ 0 \\ 3 \end{bmatrix} \right\} \qquad \text{base de } U^\perp : \left\{ \begin{bmatrix} -2 \\ 2 \\ 1 \\ 0 \end{bmatrix}, \begin{bmatrix} -1 \\ -2 \\ 0 \\ 1 \end{bmatrix} \right\}$$

$$\text{base de } \mathbb{R}^4 : \left\{ \begin{bmatrix} 1 \\ 0 \\ 2 \\ 1 \end{bmatrix}, \begin{bmatrix} 1 \\ 1 \\ 0 \\ 3 \end{bmatrix}, \begin{bmatrix} -2 \\ 2 \\ 1 \\ 0 \end{bmatrix}, \begin{bmatrix} -1 \\ -2 \\ 0 \\ 1 \end{bmatrix} \right\}$$

□

Notez que les bases trouvées avec l'algorithme 9.24 ne sont pas orthogonales en général. Si vous avez besoin des bases orthogonales, vous pouvez utiliser l'algorithme Gram-Schmidt (l'algorithme 9.20) avec des bases de U et U^T trouvées avec l'algorithme 9.24.

Exercice 9.26. Vérifier que chaque vecteur dans la base pour $U^\perp$ est orthogonal à chaque vecteur dans la base pour U. □

Exercice 9.27. Dans l'exemple 9.25, vérifier que la base pour U n'est pas une base orthogonale. La transformer en base orthogonale. Faire de même pour la base pour $U^\perp$. Montrer que si on combine les deux bases orthogonales obtenues, on obtient une base orthogonale pour $\mathbb{R}^4$. Pourquoi est-ce qu'on n'a pas du faire une méthode de Gram-Schmidt avec quatre vecteurs ? Qu'arrive si on fait une méthode de Gram-Schmidt avec les quatre vecteurs ? □

Pour ceux qui s'intéressent, la méthode de l'algorithme 9.24 repose sur l'idée des matrices élémentaires. Une réduction de Gauss consiste en une multiplication par une séquence de matrices élémentaires. Donc on peut comprendre toute la réduction par une multiplication à gauche par une matrice inversible E. Donc la matrice finale $[R|B]$ est exactement $E[A|I] = [EA|E]$. Le fait d'avoir des rangées nulles à gauche veut dire que certaines rangées de E sont orthogonales à toutes les colonnes de A. Donc ce sont exactement ces rangées qu'on veut. Elles se trouvent à droite, car $EI = E$.

10. Approximations

10.1. **Motivation.** On propose de modeler un ensemble de données expérimentales avec une fonction. On connaît la *forme* de la fonction mais pas la fonction exacte. Par exemple, on connaît peut-être que la fonction devrait être une droite sans savoir les coefficients exactes.

On a donc un système d'approximations : chaque point représente une approximation de la droite. Les inconnues, c'est-à-dire les variables, sont alors les coefficients de la droite.

Exemple 10.1. Pour chaque valeur de x, on a mesuré une valeur y. Les valeurs y sont sujets à divers erreurs. On pense que les valeurs devraient être relié par une fonction $y = \alpha x + \beta$. Voici les données ; le but est de déterminer α et β.

x	0	1	2	3	4	5
y	1	2	2	3	3	4

□

En voulant que les données de l'exemple 10.1 suivent la fonction, on cherche une solution au système suivant.

$$\begin{cases} (1) = \alpha(0) + \beta \\ (2) = \alpha(1) + \beta \\ (2) = \alpha(2) + \beta \\ (3) = \alpha(3) + \beta \\ (3) = \alpha(4) + \beta \\ (4) = \alpha(5) + \beta \end{cases} \qquad \begin{bmatrix} 0 & 1 \\ 1 & 1 \\ 2 & 1 \\ 3 & 1 \\ 4 & 1 \\ 5 & 1 \end{bmatrix} \begin{bmatrix} \alpha \\ \beta \end{bmatrix} = \begin{bmatrix} 1 \\ 2 \\ 2 \\ 3 \\ 3 \\ 4 \end{bmatrix}$$

Le système $A\mathbf{x} = \mathbf{b}$ à droite ne possède aucune solution, donc on cherche la meilleure approximation $A\mathbf{x} \approx \mathbf{b}$. Notre but est de comprendre comment résoudre $A\mathbf{x} \approx \mathbf{b}$.

Exercice 10.2. Vérifier que le système $A\mathbf{x} = \mathbf{b}$ provenant de l'exemple 10.1 ne possède aucune solution exacte. □

Exercice 10.3. Faire un graphique des données de l'exemple 10.1 et faire une estimation visuelle de la droite. □

10.2. **Approximation de droites I : projections.** Afin de résoudre $A\mathbf{x} \approx \mathbf{b}$, on peut commencer avec la question suivante : quelle est la meilleure approximation $\hat{\mathbf{b}} \approx \mathbf{b}$ tel que $A\mathbf{x} = \hat{\mathbf{b}}$ possède une solution exacte ?

"Meilleure approximation" veut dire que $\left\|\mathbf{b} - \hat{\mathbf{b}}\right\|$ est un vecteur aussi petit que possible, sujet à la contrainte que $A\mathbf{x} = \hat{\mathbf{b}}$ possède une solution exacte. Que $A\mathbf{x} = \hat{\mathbf{b}}$ possède une solution exacte équivaut à dire que $\hat{\mathbf{b}}$ est dans l'espace engendré par les colonnes de A, $\text{col}(A)$. On reconnaît

$$\mathbf{b} = \hat{\mathbf{b}} + \left(\mathbf{b} - \hat{\mathbf{b}}\right) = \text{proj}_{\text{col}(A)}(\mathbf{b}) + \left(\mathbf{b} - \text{proj}_{\text{col}(A)}(\mathbf{b})\right)$$

Le vecteur $\hat{\mathbf{b}}$ est exactement la projection de $\mathbf{b}$ sur l'espace engendré par les colonnes de A.

Afin de calculer la projection, on transforme les colonnes de A en base orthogonale. . . méthode de Gram-Schmidt bien sur ! Posons $\mathbf{a}_1 = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix}^T$ et $\mathbf{a}_2 = \begin{bmatrix} 0 & 1 & 2 & 3 & 4 & 5 \end{bmatrix}^T$. La base orthogonale serait alors $\{\mathbf{u}_1, \mathbf{u}_2\}$ avec

$$\begin{aligned} \mathbf{u}_1 &= \mathbf{a}_1 = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix}^T \\ \mathbf{u}_2 &= \mathbf{a}_2 - \frac{\mathbf{a}_2 \cdot \mathbf{u}_1}{\mathbf{u}_1 \cdot \mathbf{u}_1}\mathbf{u}_1 = \mathbf{a}_2 - \frac{15}{6}\mathbf{u}_1 = \frac{1}{2}\begin{bmatrix} -5 & -3 & -1 & 1 & 3 & 5 \end{bmatrix}^T \end{aligned}$$

Exercice 10.4. Dans la méthode de Gram-Schmidt, on a pris les colonnes de A dans la "mauvaise" ordre. En réalité, tout ordre est bon. Refaire Gram-Schmidt en commençant avec l'autre colonne de A. □

Sachant une base orthogonale pour $\mathrm{col}(A)$, on peut calculer $\hat{\mathbf{b}}$ comme projection :

$$\begin{aligned}
\hat{\mathbf{b}} &= \mathrm{proj}_{\mathrm{col}(A)}(\mathbf{b}) \\
&= \mathrm{proj}_{\mathbf{u}_1}(\mathbf{b}) + \mathrm{proj}_{\mathbf{u}_2}(\mathbf{b}) \\
&= \frac{\mathbf{b}\cdot\mathbf{u}_1}{\mathbf{u}_1\cdot\mathbf{u}_1}\mathbf{u}_1 + \frac{\mathbf{b}\cdot\mathbf{u}_2}{\mathbf{u}_2\cdot\mathbf{u}_2}\mathbf{u}_2 \\
&= \frac{15}{6}\mathbf{u}_1 + \frac{19/2}{70/4}\mathbf{u}_2 \\
&= \frac{5}{2}\begin{bmatrix}1 & 1 & 1 & 1 & 1 & 1\end{bmatrix} + \frac{19}{70}\begin{bmatrix}-5 & -3 & -1 & 1 & 3 & 5\end{bmatrix}^T \\
&= \frac{1}{35}\begin{bmatrix}40 & 59 & 78 & 97 & 116 & 135\end{bmatrix}^T
\end{aligned}$$

Si on remplace les valeurs y dans les données originales par celles-ci, les points seront tous exactement aligné sur la droite optimale. On résout directement $A\mathbf{x} = \hat{\mathbf{b}}$.

$$\left[\begin{array}{cc|c} 0 & 1 & 40/35 \\ 1 & 1 & 59/35 \\ 2 & 1 & 78/35 \\ 3 & 1 & 97/35 \\ 4 & 1 & 116/35 \\ 5 & 1 & 135/35 \end{array}\right] \to \cdots \to \left[\begin{array}{cc|c} 1 & 0 & 19/35 \\ 0 & 1 & 40/35 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{array}\right]$$

Ceci donne la solution

$$\mathbf{x} = \begin{bmatrix}\alpha \\ \beta\end{bmatrix} = \begin{bmatrix}19/35 \\ 8/7\end{bmatrix}$$

Exercice 10.5. La matrice augmentée précédente possède beaucoup plus de rangées que de colonnes. On voit que — heureusement ! — les rangées sans pivots sont toutes complètement nulles. Il n'y a eu aucune "rangée contradictoire". Expliquer pourquoi ceci n'est pas dû à l'hasard, mais serait toujours le cas pour le système $A\mathbf{x} = \hat{\mathbf{b}}$. En autres mots, expliquer pourquoi le système $A\mathbf{x} = \hat{\mathbf{b}}$ est toujours consistant. □

La droite optimale est alors

$$y = \frac{19}{35}x + \frac{8}{7}.$$

Exercice 10.6. Faire un graphique, incluant les données originales, les données ajustées ($\hat{\mathbf{b}}$ au lieu de $\mathbf{b}$) et la droite optimale. □

Leçon 14 : 7 novembre 2011

Pour résumer :

Algorithme 10.7 (Résoudre $A\mathbf{x} \approx \mathbf{b}$ par projections)**.** On cherche à trouver la meilleure solution approximative de $A\mathbf{x} \approx \mathbf{b}$.

(a) Transformer les colonnes de A en base orthogonale, à l'aide de la méthode Gram-Schmidt.
(b) Calculer $\hat{\mathbf{b}} = \text{proj}_{\text{col}(A)}(\mathbf{b})$ en utilisant la base orthogonale. Le vecteur $\hat{\mathbf{b}}$ représente les données "ajustées", qui suivent la droite exactement.
(c) Solutionner $A\mathbf{x} = \hat{\mathbf{b}}$, afin de trouver $\mathbf{x}$. Ceci donne directement la droite.

□

10.3. **Approximation de droites II : équations normales.** Il y a une autre méthode, qui évite la méthode de Gram-Schmidt et la projections, et qui donne aussi une matrice plus petite pour réduire.

On a déjà vu que

$$\mathbf{b} = \text{proj}_{\text{col}(A)}(\mathbf{b}) + \left(\mathbf{b} - \text{proj}_{\text{col}(A)}(\mathbf{b})\right).$$

Le premier vecteur à droite est dans l'espace $\text{col}(A)$: c'est l'observation qui a motivée la méthode des projections. Le *deuxième* vecteur à droite est orthogonal à $\text{col}(A)$, voulant dire que le produit scalaire du deuxième vecteur avec chaque colonne de A est 0, voulant dire que A^T multiplié par le deuxième vecteur donne $\mathbf{0}$:

$$\mathbf{0} = A^T\left(\mathbf{b} - \text{proj}_{\text{col}(A)}(\mathbf{b})\right) = A^T\left(\mathbf{b} - \hat{\mathbf{b}}\right) = A^T(\mathbf{b} - A\mathbf{x}) = A^T\mathbf{b} - A^TA\mathbf{x}.$$

Donc on cherche à résoudre $A^TA\mathbf{x} = A^T\mathbf{b}$. Note que c'est une égalité : c'est un système linéaire ordinaire. On calcul

$$A^TA = \begin{bmatrix} 0 & 1 & 2 & 3 & 4 & 5 \\ 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} 0 & 1 \\ 1 & 1 \\ 2 & 1 \\ 3 & 1 \\ 4 & 1 \\ 5 & 1 \end{bmatrix} = \begin{bmatrix} 55 & 15 \\ 15 & 6 \end{bmatrix} \qquad A^T\mathbf{b} = \begin{bmatrix} 0 & 1 & 2 & 3 & 4 & 5 \\ 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ 2 \\ 2 \\ 3 \\ 3 \\ 4 \end{bmatrix} = \begin{bmatrix} 47 \\ 15 \end{bmatrix}$$

On peut maintenant résoudre le système d'*équations normales*.

$$\left[\begin{array}{cc|c} 55 & 15 & 47 \\ 15 & 6 & 15 \end{array}\right] \to \cdots \to \left[\begin{array}{cc|c} 1 & 0 & 19/35 \\ 0 & 1 & 8/7 \end{array}\right]$$

On a la solution

$$\mathbf{x} = \begin{bmatrix} \alpha \\ \beta \end{bmatrix} = \begin{bmatrix} 19/35 \\ 8/7 \end{bmatrix}.$$

On peut maintenant calculer $\hat{\mathbf{b}}$, car $A\mathbf{x} = \hat{\mathbf{b}}$, et on connaît maintenant $\mathbf{x}$.

$$\hat{\mathbf{b}} = A\mathbf{x} = \begin{bmatrix} 0 & 1 \\ 1 & 1 \\ 2 & 1 \\ 3 & 1 \\ 4 & 1 \\ 5 & 1 \end{bmatrix} \begin{bmatrix} 19/35 \\ 8/7 \end{bmatrix} = \frac{1}{35} \begin{bmatrix} 40 \\ 59 \\ 78 \\ 97 \\ 116 \\ 135 \end{bmatrix}$$

Note qu'on a solutionné le système $A\mathbf{x} = \hat{\mathbf{b}}$, *sans connaître* $\hat{\mathbf{b}}$. Ceci permet de trouver $\hat{\mathbf{b}}$ directement.

Pour ceux qui veulent comprendre la différence entre ces deux méthodes, on propose l'exercice (la discussion !) suivante.

Exercice 10.8. Cette méthode semble beaucoup plus simple. Mais regarder de près les calculs de A^TA et $A^T\mathbf{b}$, et comparer avec la méthode de Gram-Schmidt et le calcul de $\hat{\mathbf{b}}$ dans la méthode des projections. Est-ce qu'on a *vraiment* évité la méthode de Gram-Schmidt ? De plus, la matrice, bien que petite, possède des "grands" chiffres, donc la réduction est moins facile qu'elle ne paraît. La matrice augmentée de la méthode de projections est plus grande, mais ce n'est pas nécessaire de compléter la réduction, grâce à l'exercice 10.5 (n'est-ce pas ?). Est-ce que ce sont vraiment deux méthodes différentes ? □

Il reste une difficulté. Ici le système $A^TA\mathbf{x} = A^T\mathbf{b}$ était consistant. Serait-ce toujours le cas ? Si les colonnes de A sont indépendants, alors oui, grâce au résultat suivant. On se rappelle que le *rang* d'une matrice est le nombre de pivots dans sa forme échelonnée.

Théorème 10.9. *Pour toute matrice A, les rangs de A, A^TA, et AA^T sont égaux.*

On ne démontrera pas ce théorème, mais on verra plus tard une version plus générale. Pour le moment on observe simplement que si les colonnes de A sont indépendantes, alors le rang de A est égale à n (nombre de colonnes de A), et alors A^TA a aussi n pivots ; étant une matrice carré $(n \times n)$, ceci garantit que $A^TA\mathbf{x} = \mathbf{d}$ possède une solution unique pour tout $\mathbf{d}$ (en particulier, $\mathbf{d} = A^T\mathbf{b}$).

Mais alors comment sait-on que les colonnes de A sont indépendantes ? La réponse est que parfois, elles ne le sont pas ! Si les colonnes ne sont pas indépendantes, ceci veut dire qu'il n'y a pas une seule droite optimale, mais plusieurs. Par exemple, si l'exemple 10.1 n'avait qu'un seul point, on aurait une matrice A avec des colonnes dépendantes.

Exercice 10.10. Montrer que si les données originales ont au moins deux points, alors les colonnes de A seront indépendantes. Postuler une explication géométrique de ceci. □

En bref : le système $A^TA\mathbf{x} = A^T\mathbf{b}$ serait "toujours" consistant.

Pour résumer :

Algorithme 10.11 (Résoudre $A\mathbf{x} \approx \mathbf{b}$ par équations normales)**.** On cherche à trouver la meilleure solution approximative de $A\mathbf{x} \approx \mathbf{b}$.

(a) On calcule A^TA et $A^T\mathbf{b}$.
(b) On résout (exactement) $A^TA\mathbf{x} = A^T\mathbf{b}$. Ceci donne la droite.
(c) On calcule $\hat{\mathbf{b}}$ par $\hat{\mathbf{b}} = A\mathbf{x}$. Ceci donne les données "ajustées", qui suivent la droite exactement.

□

10.4. **Approximation de droites III : matrices de projection.** Il y a une autre approche qui est parfois utile.

On a observé que le rang de A^TA est égal au nombre de colonnes ; de plus c'est une matrice carré. Donc c'est une matrice inversible. Ceci permet une variante de l'approche précédente. Afin de résoudre $A^TA\mathbf{x} = A^T\mathbf{b}$ on peut utiliser l'inverse.

$$\left(A^TA\right)^{-1}\left(A^TA\right)\mathbf{x} = \left(A^TA\right)^{-1}A^T\mathbf{b} \implies \mathbf{x} = (A^TA)^{-1}A^T\mathbf{b}.$$

Note que cette expression ne peut pas se simplifier ! Ensuite on calcule $\hat{\mathbf{b}}$ directement.

$$\hat{\mathbf{b}} = A\mathbf{x} = A\left(A^TA\right)^{-1}A^T\mathbf{b}$$

Donc on pose $M = \left(A^TA\right)^{-1}A^T$ et $N = AM = A\left(A^TA\right)^{-1}A^T$, et on a alors

$$\mathbf{x} = M\mathbf{b}, \qquad \hat{\mathbf{b}} = N\mathbf{b}.$$

Ceci montre que si A est constante, alors les calculs de $\mathbf{x}$ et $\hat{\mathbf{b}}$ sont des transformations linéaires.

Typiquement cette méthode, bien qu'elle paraît moins compliquée, requiert plus de calculs. Mais il y a une situation où elle se montre utile. Les valeurs x dans les données correspondent souvent à l'organisation de l'expérience (sondage, investigation, etc), tandis que les valeurs y correspondent souvent aux valeurs mesurées. Donc si on répète l'expérience, la matrice A peut demeurer constante, et donc on pourra calculer M et N une fois, et les réutiliser pour chaque nouvelle $\mathbf{b}$.

10.5. **Approximations de fonctions générales.** On peut comprendre l'équation $A\mathbf{x} \approx \mathbf{b}$ comme étant une représentation matricielle des données et de la droite désirée à la fois. Le vecteur $\mathbf{b}$ est exactement les valeurs y. Le produit matricielle $A\mathbf{x}$ représente l'évaluation des valeurs x dans la droite. [3]

Cette méthode s'adapte à des fonctions plus générales.

Exemple 10.12. Considérons encore les données de l'exemple 10.1, mais cette fois on trouvera une fonction de la forme $\alpha x^2 + \beta x + \gamma$. Revoici les données :

x	0	1	2	3	4	5
y	1	2	2	3	3	4

Chaque rangée de $A\mathbf{x} \approx \mathbf{b}$ représente une des données. Par exemple, la première est $(0, 1)$, qui est supposé de respecter la fonction $y = \alpha x^2 + \beta x + \gamma$. Donc on a

$$\alpha(0)^2 + \beta(0) + \gamma = 1.$$

3. Rappel : les valeurs x ne sont pas la même chose que le vecteur $\mathbf{x}$.

Pour la deuxième rangée on a la donnée $(1, 2)$, donc

$$\alpha(1)^2 + \beta(1) + \gamma = 2.$$

En total, on forme la version matricielle.

$$\begin{bmatrix} (0)^2 & (0) & 1 \\ (1)^2 & (1) & 1 \\ (2)^2 & (2) & 1 \\ (3)^2 & (3) & 1 \\ (4)^2 & (4) & 1 \\ (5)^2 & (5) & 1 \end{bmatrix} \begin{bmatrix} \alpha \\ \beta \\ \gamma \end{bmatrix} \approx \begin{bmatrix} 1 \\ 2 \\ 2 \\ 3 \\ 3 \\ 4 \end{bmatrix} \quad \longrightarrow \quad \begin{bmatrix} 0 & 0 & 1 \\ 1 & 1 & 1 \\ 4 & 2 & 1 \\ 9 & 3 & 1 \\ 16 & 4 & 1 \\ 25 & 5 & 1 \end{bmatrix} \begin{bmatrix} \alpha \\ \beta \\ \gamma \end{bmatrix} \approx \begin{bmatrix} 1 \\ 2 \\ 2 \\ 3 \\ 3 \\ 4 \end{bmatrix}$$

Exercice 10.13. On cherche à trouver, pour les mêmes données, la meilleure équation de la forme $y = \alpha x^3 + \beta x^2 + \gamma x + \delta$. Donner la forme matricielle $A\mathbf{x} \approx \mathbf{b}$ qui convient. NB : si vous voulez le résoudre, un solveur serait utile ! □

Exercice 10.14. On cherche à trouver, pour les mêmes données, la meilleure équation de la forme $y = \alpha 2^x + \beta x + \gamma$. Donner la forme matricielle $A\mathbf{x} \approx \mathbf{b}$ qui convient. NB : si vous voulez le résoudre, un solveur serait utile ! □

10.6. **Erreur de l'approximation.** En trouvant une solution approximative, on a trouvé des valeurs $\hat{\mathbf{b}}$ qui suivent la fonction désirée exactement, au lieu des valeurs $\mathbf{b}$ originales. Donc l'erreur de l'approximation est dans ce terme. En fait, on a présumé que dans les données, les x étaient toujours exacte, et les y douteux. C'est souvent le cas, ou plutôt c'est souvent le cas que l'erreur soit concentré dans une partie. Un traitement plus générale dépasse notre cours.

Donc on a trouvé un $\hat{\mathbf{b}}$ qui est aussi proche de $\mathbf{b}$ que possible. Ceci minimise la distance entre $\hat{\mathbf{b}}$ et $\mathbf{b}$, soit $\left\|\mathbf{b} - \hat{\mathbf{b}}\right\|$. On pose $\mathbf{e} = \mathbf{b} - \hat{\mathbf{b}}$; donc chaque composant de $\mathbf{e}$ représente l'erreur entre la valeur mesurée et la valeur de la droite. Précisément, chaque composant représente la distance entre les données originales et les données ajustées dans votre graphique de l'exercice 10.6. En minimisant $\left\|\mathbf{b} - \hat{\mathbf{b}}\right\|$, on minimise également $\left\|\mathbf{b} - \hat{\mathbf{b}}\right\|^2$, et donc

$$\left\|\mathbf{b} - \hat{\mathbf{b}}\right\|^2 = e_1^2 + e_2^2 + \cdots + e_m^2,$$

où m est le nombre de données (nombre de rangées dans A). On minimise la somme des carrées : c'est la raison pourquoi cette méthode s'appelle parfois la méthode des *moindres carrées*.

On pourra donc comprendre la solution de $A\mathbf{x} \approx \mathbf{b}$ comme

$$\min \left\|\mathbf{b} - \hat{\mathbf{b}}\right\|^2 \quad \text{s.c.} \quad \hat{\mathbf{b}} \in \text{col}(A).$$

C'est une optimisation quadratique.

Leçon 15 : 10 novembre 2011

11. Formes quadratiques

11.1. **Introduction.** On a vu comment optimiser une fonction linéaire sujet à des contraintes linéaires : les programmes linéaires et la méthode de simplex.

La solution des systèmes approximatifs est aussi une optimisation. Il s'agit de minimiser $\left\|\mathbf{b}-\hat{\mathbf{b}}\right\|^2$ sujet à la contrainte que $\hat{\mathbf{b}} \in \text{col}(A)$. C'est une fonction objective quadratique et une contrainte linéaire.

On étudiera comment optimiser une fonction quadratique sujet à une contrainte quadratique.

11.2. **Formes quadratiques.** Une *forme quadratique* est un polynôme $Q(\mathbf{x}) = Q(x_1, \dots, x_n)$ en plusieurs variables, dont chaque terme est de degré deux.

Exemple 11.1. Voici trois exemples de formes quadratiques :

$$x_1^2 + x_2^2, \qquad x_1^2 - x_1x_2 + 4x_2x_3 + 2x_3^2, \qquad x_1x_2 + x_2x_3 + x_3x_1.$$

□

Les formes quadratiques s'écrivent naturellement en matrices. Posons $\mathbf{x} = \begin{bmatrix} x_1 & x_2 & \cdots & x_n \end{bmatrix}^T$, un vecteur des variables et $A = [a_{ij}]$ une matrice $n \times n$. Alors :

$$Q(\mathbf{x}) = \mathbf{x}^T A \mathbf{x} = \sum_{1 \le i,j \le n} a_{ij} x_i x_j.$$

Par exemple $\begin{bmatrix} x_1 & x_2 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = x_1^2 + x_2^2$: la forme $x_1^2 + x_2^2$ correspond à la matrice $\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$. Ici, la matrice est unique ; en général on a un peu de choix.

Exercice 11.2. Vérifier que $\mathbf{x}^T A \mathbf{x} = x_1x_2 + x_2x_3 + x_3x_1$ si $\mathbf{x} = \begin{bmatrix} x_1 & x_2 & x_3 \end{bmatrix}^T$ et A est n'importe quelle des matrices suivantes :

$$\begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{bmatrix} \qquad \begin{bmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \qquad \begin{bmatrix} 0 & 1 & 1 \\ 0 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \qquad \begin{bmatrix} 0 & 1/2 & 1/2 \\ 1/2 & 0 & 1/2 \\ 1/2 & 1/2 & 0 \end{bmatrix}$$

Combien d'autres matrices sont possibles (indice : beaucoup !). □

Pour des raisons qu'on verra très bientôt, on préfère toujours la matrice symétrique. On dit que la *matrice d'une forme quadratique* sur les variables $x_1, x_2, \cdots, x_n$ est la matrice de taille $n \times n$ telle que la position A_{ii} de la matrice est le coefficient de x_i^2, et la position A_{ij} de la matrice est la moitié du coefficient de x_ix_j (pour $i \neq j$).

Exercice 11.3. Donner la matrice (symétrique, bien sur !) pour chaque forme :

$$x_1^2 + 4x_1x_2 + 6x_2^2, \qquad 4x_1^2 + 6x_2x_3, \qquad -4x_1^2 + +4x_2^2 + 6x_1x_2.$$

□

Exercice 11.4. Donner les formes quadratiques qui correspondent aux matrices suivantes.

$$\begin{bmatrix} 1 & 2 \\ 2 & 1 \end{bmatrix} \qquad \begin{bmatrix} -1 & 1 \\ 1 & -1 \end{bmatrix} \qquad \begin{bmatrix} 2 & 1 & 2 \\ 1 & 3 & 1/2 \\ 2 & 1/2 & 5 \end{bmatrix}$$

□

11.3. **Matrices symétriques.** Pourquoi choisir une matrice symétrique pour une forme quadratique ? À cause du résultat suivant.

Théorème 11.5. *Toute matrice symétrique est diagonalisable.* □

On se rappelle qu'un matrice est diagonalizable si et seulement si la dimension de chaque espace propre est égale à la multiplicité de la valeur propre correspondante.

Il y a une autre raison.

Théorème 11.6. *Soit A une matrice symétrique, avec $\mathbf{v}$ et $\mathbf{w}$ des vecteurs propres avec valeurs propres distinctes λ et μ. Alors $\mathbf{v}$ et $\mathbf{w}$ sont orthogonaux.*

Démonstration. Vérifier les étapes suivants. C'est utile de commencer au centre et de procéder à gauche et à droite.

$$\mu \mathbf{w}^T \mathbf{v} = \left(\mathbf{w}^T A^T\right) \mathbf{v} = \mathbf{w}^T A^T \mathbf{v} = \mathbf{w}^T A \mathbf{v} = \mathbf{w}^T (A\mathbf{v}) = \lambda \mathbf{w}^T \mathbf{v}$$

Donc $\mu \mathbf{w}^T \mathbf{v} = \lambda \mathbf{w}^T \mathbf{v} \implies (\mu - \lambda)\mathbf{w}^T \mathbf{v} = 0$. Puisque $\mu \neq \lambda$ il faut que $\mathbf{w}^T \mathbf{v} = 0$. □

La conséquence est importante : les différentes espaces propres sont automatiquement orthogonales. On peut transformer chaque base propre à une base propre orthogonale avec Gram-Schmidt. Donc on obtient le résultat suivant.

Théorème 11.7 (Le Théorème des Axes Principaux)**.** *Soit A une matrice symétrique. Alors $A = PDP^T$ pour une matrice diagonale D et une matrice P avec $P^{-1} = P^T$. Donc $\mathbf{x} = P\mathbf{y}$ est un changement de variable orthogonal qui transforme la forme quadratique $\mathbf{x}^T A \mathbf{x}$ en une forme quadratique $\mathbf{y}^T D \mathbf{y}$ où D est diagonale.* □

C'est en combinant les deux résultats précédents. On obtient une diagonalization comme d'habitude. Ensuite on applique Gram-Schmidt à chaque espace propre, et on normalise les vecteurs propres obtenus, ce qui donne une base orthonormale. La matrice P alors est automatiquement tel que $P^{-1} = P^T$. Puis on voit que

$$\mathbf{x}^T A \mathbf{x} = (P\mathbf{y})^T A P \mathbf{y} = \mathbf{y}^T P^T A P \mathbf{y} = \mathbf{y}^T D \mathbf{y}.$$

C'est utile de voir qu'il y a peu de nouveau ici en pratique : les valeurs propres et les bases propres s'obtiennent comme d'habitude. On est garantit que les espaces sont déjà orthogonales entre eux, donc il ne faudrait faire Gram-Schmidt que *dans* chaque base propre (et non pas à tout l'ensemble).

Exemple 11.8. Trouver une décomposition $A = PDP^T$ pour $A = \begin{bmatrix} 3 & 1 \\ 1 & 3 \end{bmatrix}$.

On cherche les valeurs propres en factorisant le polynôme caractéristique.

$$\det \begin{bmatrix} 3-\lambda & 1 \\ 1 & 3-\lambda \end{bmatrix} = (3-\lambda)^2 - 1 = \lambda^2 - 6\lambda + 8 = (\lambda - 4)(\lambda - 2)$$

Ensuite on trouve une base pour chaque espace propre.

$$\lambda = 4: \qquad A - (4)I = \begin{bmatrix} -1 & 1 \\ 1 & -1 \end{bmatrix} \to \begin{bmatrix} 1 & -1 \\ 0 & 0 \end{bmatrix} \qquad \text{base pour } E_4: \left\{ \begin{bmatrix} 1 \\ 1 \end{bmatrix} \right\}$$

$$\lambda = 2: \qquad A - (2)I = \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix} \to \begin{bmatrix} 1 & 1 \\ 0 & 0 \end{bmatrix} \qquad \text{base pour } E_2: \left\{ \begin{bmatrix} -1 \\ 1 \end{bmatrix} \right\}$$

La dimension de chaque espace propre est un (on savait ceci avant de calculer les espaces propres : pourquoi ?). Donc chaque base propre est déjà une base orthogonale. Il ne faut que normaliser, c'est-à-dire diviser chaque vecteur par sa longueur (ici, $\sqrt{2}$). Les deux vecteurs forment les colonnes de P, et les deux valeurs propres forment les éléments diagonales de D.

$$\begin{aligned} A = PDP^T &= \begin{bmatrix} 1/\sqrt{2} & -1/\sqrt{2} \\ 1/\sqrt{2} & 1/\sqrt{2} \end{bmatrix} \begin{bmatrix} 4 & 0 \\ 0 & 2 \end{bmatrix} \begin{bmatrix} 1/\sqrt{2} & -1/\sqrt{2} \\ 1/\sqrt{2} & 1/\sqrt{2} \end{bmatrix}^T \\ &= \begin{bmatrix} 1/\sqrt{2} & -1/\sqrt{2} \\ 1/\sqrt{2} & 1/\sqrt{2} \end{bmatrix} \begin{bmatrix} 4 & 0 \\ 0 & 2 \end{bmatrix} \begin{bmatrix} 1/\sqrt{2} & 1/\sqrt{2} \\ -1/\sqrt{2} & 1/\sqrt{2} \end{bmatrix} \end{aligned}$$

□

Exercice 11.9. Vérifier que les deux espaces propres sont orthogonales dans l'exemple précédent. C'est-à-dire, vérifier que $\begin{bmatrix} 1 \\ 1 \end{bmatrix}$ et $\begin{bmatrix} -1 \\ 1 \end{bmatrix}$ sont orthogonales. C'est l'orthogonalité *entre* les deux espaces.

Expliquer pourquoi les deux bases sont chacune déjà des bases orthogonales. C'est l'orthogonalité *dans* chaque espace. □

Exemple 11.10. Les valeurs propres de $A = \begin{bmatrix} 3 & -2 & 4 \\ -2 & 6 & 2 \\ 4 & 2 & 3 \end{bmatrix}$ sont $\lambda = -2$ de multiplicité un et $\lambda = 7$ de multiplicité deux. Trouver une décomposition $A = PDP^T$.

$$\lambda = -2: \quad A - (-2)I = \begin{bmatrix} 5 & -2 & 4 \\ -2 & 8 & 2 \\ 4 & 2 & 5 \end{bmatrix} \to \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 1/2 \\ 0 & 0 & 0 \end{bmatrix} \qquad \text{base pour } E_{-2}: \left\{ \begin{bmatrix} -2 \\ -1 \\ 2 \end{bmatrix} \right\}$$

$$\lambda = 7: \quad A - (7)I = \begin{bmatrix} -4 & -2 & 4 \\ -2 & -1 & 2 \\ 4 & 2 & -4 \end{bmatrix} \to \begin{bmatrix} 1 & 1/2 & -1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \qquad \text{base pour } E_7: \left\{ \begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix}, \begin{bmatrix} -1 \\ 2 \\ 0 \end{bmatrix} \right\}$$

Comme prévue, la dimension est égale à la multiplicité chaque fois. Aussi on voit que chaque vecteur dans la base de E_{-2} est orthogonale à chaque vecteur dans la base de E_7. Il faut

appliquer la méthode de Gram-Schmidt à la base pour E_7.

$$\mathbf{u}_1 = \begin{bmatrix}1\\0\\1\end{bmatrix} \qquad \mathbf{u}_2 = \begin{bmatrix}-1\\2\\0\end{bmatrix} - \operatorname{proj}_{\mathbf{u}_1}\begin{bmatrix}-1\\2\\0\end{bmatrix} = \begin{bmatrix}-1\\2\\0\end{bmatrix} - \frac{-1}{2}\begin{bmatrix}1\\0\\1\end{bmatrix} = \begin{bmatrix}-1/2\\2\\1/2\end{bmatrix}$$

Il faut normaliser les deux vecteurs $\mathbf{u}_1$ et $\mathbf{u}_2$ pour obtenir une base orthonormale de E_7.

$$\frac{\mathbf{u}_1}{\|\mathbf{u}_1\|} = \frac{1}{\sqrt{2}}\begin{bmatrix}1\\0\\1\end{bmatrix} \qquad \frac{\mathbf{u}_2}{\|\mathbf{u}_2\|} = \frac{1}{\sqrt{18}}\begin{bmatrix}-1\\4\\1\end{bmatrix} = \frac{1}{3\sqrt{2}}\begin{bmatrix}-1\\4\\1\end{bmatrix}$$

On normalise aussi le vecteur dans la base pour E_{-2} :

$$\frac{1}{3}\begin{bmatrix}-2\\-1\\2\end{bmatrix}.$$

Donc

$$A = \begin{bmatrix}-2/3 & 1/\sqrt{2} & -1/3\sqrt{2}\\ -1/3 & 0 & 4/3\sqrt{2}\\ 2/3 & 1\sqrt{2} & 1/3\sqrt{2}\end{bmatrix}\begin{bmatrix}-2 & 0 & 0\\ 0 & 7 & 0\\ 0 & 0 & 7\end{bmatrix}\begin{bmatrix}-2/3 & 1/\sqrt{2} & -1/3\sqrt{2}\\ -1/3 & 0 & 4/3\sqrt{2}\\ 2/3 & 1\sqrt{2} & 1/3\sqrt{2}\end{bmatrix}^T.$$

Exercice 11.11. Vérifier que dans l'exemple précédent, chaque vecteur dans la base de E_{-2} est orthogonale à chaque vecteur dans la base de E_7.

Leçon 16 : 14 novembre 2011

Définition 11.12. Une forme quadratique $Q(\mathbf{x})$ est

(a) *définie positive* si $Q(\mathbf{x}) > 0$ pour tout $\mathbf{x} \neq \mathbf{0}$,

(b) *définie négative* si $Q(\mathbf{x}) < 0$ pour tout $\mathbf{x} \neq \mathbf{0}$,

(c) *indéfinie* si $Q(\mathbf{x})$ prend des valeurs positives et négatives.

Théorème 11.13. *Soit A une matrice symétrique. La forme quadratique $\mathbf{x}^T A\mathbf{x}$ est*

(a) définie positive si et seulement si les valeurs propres de A sont toutes strictement positives,

(b) définie négative si et seulement si les valeurs propres de A sont toutes strictement négatives,

(c) indéfinie si et seulement si A a des valeurs propres négatives et positives.

Démonstration. D'après Le Théorème des Axes Principaux (le théorème 11.7), il existe une changement de variable orthogonal $\mathbf{x} = P\mathbf{y}$ tel que

$$Q(\mathbf{x}) = \mathbf{x}^T A\mathbf{x} = \mathbf{y}^T D\mathbf{y} = \lambda_1 y_1^2 + \lambda_2 y_2^2 + \cdots + \lambda_n y_n^2,$$

où $\lambda_1, \ldots, \lambda_n$ sont les valeurs propres de A. Comme P est inversible, il y a une correspondance bijective entre tous les $\mathbf{x}$ non nuls et tous les $\mathbf{y}$ non nuls. Donc les signes des valeurs de $Q(\mathbf{x})$ pour $\mathbf{x} \neq \mathbf{0}$ sont contrôlées par les signes des valeurs propres $\lambda_1, \ldots, \lambda_n$ des trois façons décrites dans l'énoncé du théorème. □

11.4. **Optimisation quadratique.** On applique la théorie des matrices symétriques pour optimiser des formes quadratiques.

Théorème 11.14. *Soit A une matrice symétrique. Alors le maximum de $\mathbf{x}^T A\mathbf{x}$ sujet à la contrainte que $\|\mathbf{x}\| = 1$ est la valeur propre maximale de A. Ce maximum est atteint lorsque $\mathbf{x}$ est un vecteur propre correspondant (avec $\|\mathbf{x}\| = 1$).*

Démonstration. Puisque A est symétrique, on peut trouver une base orthonormale pour $\mathbb{R}^n$ formée de vecteurs propres de A : disons $\{\mathbf{v}_1, \mathbf{v}_2, \cdots, \mathbf{v}_n\}$, avec valeurs propres correspondantes $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_n$. On peut alors écrire $\mathbf{x}$ comme $\mathbf{x} = \alpha_1\mathbf{v}_1 + \alpha_2\mathbf{v}_2 + \cdots + \alpha_n\mathbf{v}_n$. Ceci donne :

$$\begin{aligned}
\mathbf{x}^T A\mathbf{x} &= \mathbf{x}^T A\left(\alpha_1\mathbf{v}_1 + \alpha_2\mathbf{v}_2 + \cdots + \alpha_n\mathbf{v}_n\right) \\
&= \mathbf{x}^T\left(\alpha_1\lambda_1\mathbf{v}_1 + \alpha_2\lambda_2\mathbf{v}_2 + \cdots + \lambda_n\alpha_n\mathbf{v}_n\right) \\
&= \left(\alpha_1\mathbf{v}_1^T + \alpha_2\mathbf{v}_2^T + \cdots + \alpha_n\mathbf{v}_n^T\right)\left(\alpha_1\lambda_1\mathbf{v}_1 + \alpha_2\lambda_2\mathbf{v}_2 + \cdots + \lambda_n\alpha_n\mathbf{v}_n\right) \\
&= \alpha_1^2\lambda_1 + \alpha_2^2\lambda_2 + \cdots + \alpha_n^2\lambda_n \\
&\leq \alpha_1^2\lambda_1 + \alpha_1^2\lambda_1 + \cdots + \alpha_n^2\lambda_1 \\
&= \lambda_1\left(\alpha_1^2 + \alpha_2^2 + \cdots + \alpha_n^2\right) \\
&= \lambda_1\|\mathbf{x}\|^2 \\
&= \lambda_1
\end{aligned}$$

Note que dans la multiplication des deux grandes parenthèses, on a utilisé le fait que $\mathbf{v}_i^T\mathbf{v}_i = 1$ et $\mathbf{v}_i^T\mathbf{v}_j = 0$ si $i \neq j$. C'est parce que c'est une base orthonormale.

Si on a $\mathbf{x}^T A\mathbf{x} = \lambda_1$, c'est que le "$\leq$" est "$=$". Il faudrait avoir $\mathbf{x}$ un vecteur propre correspondant au valeur propre λ_1. □

Il y a un résultat similaire pour minimiser.

Théorème 11.15. *Soit A une matrice symétrique. Alors le minimum de $\mathbf{x}^T A\mathbf{x}$ sujet à la contrainte que $\|x\| = 1$ est la valeur propre minimale de A. Ce minimum est atteint lorsque $\mathbf{x}$ est un vecteur propre correspondant.* □

Exemple 11.16. Quel est le maximum de $2x_1^2 + 2x_1x_2 + 3x_2^2$, sujet à la contrainte $x_1^2 + x_2^2 = 1$?

On met $\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$ et $A = \begin{bmatrix} 2 & 1 \\ 1 & 3 \end{bmatrix}$. Note que $\|\mathbf{x}\| = \sqrt{x_1^2 + x_2^2}$, et donc la condition équivaut à $\|\mathbf{x}\|^2 = 1$, ce qui équivaut à $\|\mathbf{x}\| = 1$. On cherche donc :

$$\max\ \mathbf{x}^T A\mathbf{x} \quad \text{s.c.} \quad \|\mathbf{x}\| = 1$$

Selon le théorème, le maximum est la valeur propre maximale de A, qui est $(5+\sqrt{5})/2$ (exercice). Cette valeur est atteinte si le vecteur $\mathbf{x}$ est vecteur propre correspondant à cette valeur propre (exercice : trouver ces vecteurs – il y en a deux !). □

Comment maximiser des formes quadratiques selon des contraintes quadratiques *arbitraires* ?

Exemple 11.17. Quel est le maximum de x_1x_2 selon la contrainte $3x_1^2 + 2x_1x_2 + 3x_2^2 = 4$?

On écrit la fonction objective et la contrainte en forme matricielle.

$$\max \mathbf{x}^T \begin{bmatrix} 0 & 1/2 \\ 1/2 & 0 \end{bmatrix} \mathbf{x} \quad \text{s.c.} \quad \mathbf{x}^T \begin{bmatrix} 3 & 1 \\ 1 & 3 \end{bmatrix} \mathbf{x} \leq 4$$

□

On a transformé le problème en problème matriciel. On voudrait que la réponse soit la valeur propre maximale de la fonction objective, mais ceci n'est valide que lorsque la contrainte est en forme standard. Donc on tente de récrire la contrainte.

Exemple 11.18. Effectuer un changement de variable pour transformer $\mathbf{x}^T \begin{bmatrix} 3 & 1 \\ 1 & 3 \end{bmatrix} \mathbf{x} = 4$ en forme sans terme "rectangle" : c'est-à-dire, en forme avec une matrice diagonale.

Premièrement, on obtient une décomposition PDP^T de la matrice. C'est toujours possible, car on a une matrice symétrique. On obtient

$$\mathbf{x}^T \begin{bmatrix} 3 & 1 \\ 1 & 3 \end{bmatrix} \mathbf{x} = 4$$

$$\mathbf{x}^T \begin{bmatrix} 1/\sqrt{2} & -1/\sqrt{2} \\ 1/\sqrt{2} & 1/\sqrt{2} \end{bmatrix} \begin{bmatrix} 4 & 0 \\ 0 & 2 \end{bmatrix} \begin{bmatrix} 1/\sqrt{2} & -1/\sqrt{2} \\ 1/\sqrt{2} & 1/\sqrt{2} \end{bmatrix}^T \mathbf{x} = 4$$

Si on met $\mathbf{y} = P^T\mathbf{x}$, on obtient la forme $\mathbf{y}^T \begin{bmatrix} 4 & 0 \\ 0 & 2 \end{bmatrix} \mathbf{y} = 4$, ou $4y_1^2 + 2y_2^2 = 4$. □

C'est une forme plus simple, mais on peut faire mieux, en transformant la contrainte à la forme standard $z_1^2 + z_2^2 = 1$.

Exemple 11.19. Effectuer un changement de variable pour transformer $4y_1^2 + 2y_2^2 = 4$ à la forme $z_1^2 + z_2^2 = 1$.

$$4y_1^2 + 2y_2^2 = 4 \qquad \longrightarrow \qquad y_1^2 + \frac{y_2^2}{2} = 1 \qquad \longrightarrow \qquad z_1^2 + z_2^2 = 1$$

où $z_1 = y_1$ et $z_2 = y_2/\sqrt{2}$. C'est un changement de variable qui renormalise seulement. □

La stratégie de maximiser les formes quadratiques est donc de transformer la contrainte en forme standard, et de considérer alors la fonction objective transformée en forme matricielle.

Algorithme 11.20. On cherche à optimiser une forme quadratique selon une contrainte quadratique, donc

$$\max \mathbf{x}^T M \mathbf{x} \quad \text{s.c.} \quad \mathbf{x}^T A \mathbf{x} = k.$$

(a) On obtient une décomposition $A = PDP^T$. La contrainte devient $\mathbf{y}^T D \mathbf{y} = k$ avec $\mathbf{y} = P^T\mathbf{x}$, ou $\mathbf{x} = P\mathbf{y}$.

(b) La forme $\mathbf{x}^T A\mathbf{x}$ équivaut à $\frac{\lambda_1}{k}y_1^2 + \cdots \frac{\lambda_n}{k}y_n^2$.

(c) On renormalise les variables $\mathbf{y}$ pour donner $z_1^2+\cdots+z_n^2 = 1$, à l'aide de la transformation $z_j = \sqrt{\frac{\lambda_j}{k}}y_j$, ou $y_j = \sqrt{\frac{k}{\lambda_j}}z_j$.

(d) On écrit la forme objective $\mathbf{x}^T M\mathbf{x}$ en termes de $\mathbf{z}$, et en forme matricielle, obtenant alors $\mathbf{z}^T N\mathbf{z}$.

(e) La valeur maximale de la fonction objective est la valeur propre maximale de N ; la valeur minimale de la fonction objective est la valeur propre minimale de N.

□

Note qu'en pratique, on n'a pas besoin de connaître la matrice M. Il n'y a aucune raison d'écrire la fonction objective originale en forme matricielle, c'est plutôt la fonction objective transformée qu'on devrait écrire en forme matricielle.

Exemple 11.21. Quel est le maximum de x_1x_2 selon la contrainte $3x_1^2 + 2x_1x_2 + 3x_2^2 = 4$?

L'approche est donc de transformer la contrainte en forme standard. On a déjà fait le travail : voici le résumé.

$$\begin{aligned}
3x_1^2 + 2x_1x_2 + 3x_2^2 &= 4 \\
\mathbf{x}^T \begin{bmatrix} 3 & 1 \\ 1 & 3 \end{bmatrix} \mathbf{x} &= 4 \\
\mathbf{y} \begin{bmatrix} 4 & 0 \\ 0 & 2 \end{bmatrix} \mathbf{y} &= 4 \qquad \text{avec } \mathbf{x} = \begin{bmatrix} 1/\sqrt{2} & -1/\sqrt{2} \\ 1/\sqrt{2} & 1/\sqrt{2} \end{bmatrix} \mathbf{y} \\
y_1^2 + \frac{y_2^2}{2} &= 1 \qquad \text{avec } \mathbf{x} = \begin{bmatrix} 1/\sqrt{2} & -1/\sqrt{2} \\ 1/\sqrt{2} & 1/\sqrt{2} \end{bmatrix} \mathbf{y} \\
z_1^2 + z_2^2 &= 1 \qquad \text{avec } \mathbf{x} = \begin{bmatrix} 1/\sqrt{2} & -1/\sqrt{2} \\ 1/\sqrt{2} & 1/\sqrt{2} \end{bmatrix} \mathbf{y}, \text{ et } \begin{cases} y_1 = z_1 \\ y_2 = \sqrt{2}z_2 \end{cases}
\end{aligned}$$

La transformation de variable qui "standardise" la contrainte est alors :

$$\begin{aligned}
\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} &= \begin{bmatrix} 1/\sqrt{2} & -1/\sqrt{2} \\ 1/\sqrt{2} & 1/\sqrt{2} \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \end{bmatrix} \\
&= \begin{bmatrix} 1/\sqrt{2} & -1/\sqrt{2} \\ 1/\sqrt{2} & 1/\sqrt{2} \end{bmatrix} \begin{bmatrix} z_1 \\ \sqrt{2}z_2 \end{bmatrix} \\
\text{ou } &\begin{cases} x_1 = z_1/\sqrt{2} - z_2 \\ x_2 = z_1/\sqrt{2} + z_2 \end{cases}
\end{aligned}$$

Le problème est donc

$$\begin{aligned}
&\max\ (x_1x_2) \quad \text{s.c.} \quad 3x_1^2+2x_1x_2+3x_2^2=4\\
&\max\ \left(z_1/\sqrt{2}-z_2\right)\left(z_1/\sqrt{2}+z_2\right) \quad \text{s.c.} \quad z_1^2+z_2^2=1\\
&\max\ \left(z_1^2/2-z_2^2\right) \quad \text{s.c.} \quad z_1^2+z_2^2=1\\
&\max\ \mathbf{z}^T\begin{bmatrix}1/2 & 0\\ 0 & -1\end{bmatrix}\mathbf{z} \quad \text{s.c.} \quad \|\mathbf{z}\|=1
\end{aligned}$$

Puisque la contrainte est en forme standard, le maximum de la forme quadratique est la valeur propre maximale de la matrice de la fonction objective. Donc, le maximum est $1/2$. Ce maximum se produit lorsque $\mathbf{z}$ est vecteur propre normalisé de $\lambda = 1/2$, donc $\mathbf{z} = \left[\begin{smallmatrix}\pm 1\\ 0\end{smallmatrix}\right]$, donc $x_1 = 1/\sqrt{2}-0 = 1/\sqrt{2}$ et $x_2 = 1/\sqrt{2}+0 = 1/\sqrt{2}$, ou $x_1 = -1/\sqrt{2}$, $x_2 = -1/\sqrt{2}$. □

Leçon 17 : 17 novembre 2011

12. Décomposition en valeurs singulières

12.1. **Introduction.** Une matrice A de taille $m \times n$ représente une *transformation linéaire* de $\mathbb{R}^n$ à $\mathbb{R}^m$, en considérant la fonction $\mathbf{x} \mapsto A\mathbf{x}$.

Pour des matrices carrées, cette transformation se comprend très bien en considérant les valeurs et vecteurs propres de A : c'est la théorie qui soutient notre analyse de systèmes dynamiques et chaînes de Markov, entre autres. Cette théorie dépend d'une diagonalization de A, qui n'est pas toujours possible. Certainement ce n'est jamais possible pour une matrice non-carrée.

12.2. **Valeurs singulières.** Soit A une matrice complètement arbitraire, de taille $m \times n$. Pour un vecteur $\mathbf{x} \in \mathbb{R}^n$ avec $\|\mathbf{x}\| = 1$, on considère $\|A\mathbf{x}\|$. Cette norme mesure le changement en longueur entre $\mathbf{x}$ et $A\mathbf{x}$. Note que

$$\|A\mathbf{x}\|^2 = (A\mathbf{x})^T(A\mathbf{x}) = \mathbf{x}^TA^TA\mathbf{x}.$$

Ceci suggère de considérer un vecteur propre de A^TA, c'est-à-dire un $\mathbf{x}$ avec $A^TA\mathbf{x} = \lambda\mathbf{x}$.

La matrice A^TA est carrée ($n \times n$) et symétrique (puisque $\left(A^TA\right)^T = A^T\left(A^T\right)^T = A^TA$). Donc on peut trouver une décomposition $A^TA = PDP^T$ et alors une base orthonormale de $\mathbb{R}^n$ formée de vecteurs propres de A^TA. Dénotons cette base par $\{\mathbf{v}_1, \mathbf{v}_2, \cdots, \mathbf{v}_n\}$, où $A^TA\mathbf{v}_j = \lambda_j\mathbf{v}_j$.

Théorème 12.1. *Les valeurs propres de A^TA sont réelles et non-négatives.*

Démonstration. Si $A^TA\mathbf{x} = \lambda\mathbf{x}$, $\mathbf{x} \neq \mathbf{0}$, alors

$$\|A\mathbf{x}\|^2 = \mathbf{x}^TA^TA\mathbf{x} = \lambda\mathbf{x}^T\mathbf{x} = \lambda\|\mathbf{x}\|^2,$$

et donc $\lambda = \|A\mathbf{x}\|^2 / \|\mathbf{x}\|^2$. Les deux normes sont non-négatives, et le dénominateur n'est pas nul, donc $\lambda \geq 0$. □

On considère les valeurs propres (avec multiplicités) en ordre décroissant, $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_n$. Si le rang de A^TA est r, alors $\lambda_j = 0$ pour $j > r$: les dernières $n - r$ valeurs propres sont 0.

Les *valeurs singulières* de A sont $\sigma_1, \sigma_2, \cdots, \sigma_r$, où $\sigma_i = \sqrt{\lambda_i}$. Ce sont les racines carrées des valeurs propres non-nulles de A^TA. Note que la preuve du théorème 12.1 implique que si $\mathbf{v}_i$ est un vecteur propre unitaire de A^TA correspondant à la valeur propre non-nulle λ_i, alors $\|A\mathbf{v}_i\| = \sigma_i$.

Exemple 12.2. Soit $A = \begin{bmatrix} 3 & -1 \\ 1 & 3 \\ 1 & 1 \end{bmatrix}$. Calculer les valeurs singulières de A, et de A^T.

Pour A, on a $A^TA = \begin{bmatrix} 11 & 1 \\ 1 & 11 \end{bmatrix}$. On calcul

$$\begin{aligned} \det\left(A^TA - \lambda I\right) &= \det\begin{bmatrix} 11-\lambda & 1 \\ 1 & 11-\lambda \end{bmatrix} \\ &= (\lambda - 11)(\lambda - 11) - 1 \\ &= \lambda^2 - 22\lambda + 120 \\ &= (10 - \lambda)(\lambda - 12) \end{aligned}$$

Les valeurs propres sont $12, 10$. Les valeurs singulières de A^T sont alors $\sigma_1 = \sqrt{12}$ et $\sigma_2 = \sqrt{10}$.

Pour A^T on a $(A^T)^TA^T = AA^T = \begin{bmatrix} 10 & 0 & 2 \\ 0 & 10 & 4 \\ 2 & 4 & 2 \end{bmatrix}$. On calcul

$$\begin{aligned} \det\left(AA^T - \lambda I\right) &= \begin{bmatrix} 10-\lambda & 0 & 2 \\ 0 & 10-\lambda & 4 \\ 2 & 4 & 2-\lambda \end{bmatrix} \\ &= (10-\lambda)\det\begin{bmatrix} 10-\lambda & 4 \\ 4 & 2-\lambda \end{bmatrix} + 2\det\begin{bmatrix} 0 & 10-\lambda \\ 2 & 4 \end{bmatrix} \\ &= (10-\lambda)\left((10-\lambda)(2-\lambda) - 16\right) + 2\left((10-\lambda)(2) - 0\right) \\ &= (10-\lambda)\left((10-\lambda)(2-\lambda) - 16 + 4\right) \\ &= (10-\lambda)(\lambda)(\lambda - 12) \end{aligned}$$

Les valeurs propres sont $12, 10, 0$. Les valeurs singulières de A sont $\sigma_1 = \sqrt{12}$ et $\sigma_2 = \sqrt{10}$.

12.3. **Décomposition.** On observe dans l'exemple 12.2 que A et A^T avaient les mêmes valeurs singulières. Ce n'est pas par hasard...

Théorème 12.3. *Soit A une matrice de taille $m \times n$. Les matrices A^TA et AA^T ont exactement les mêmes valeurs propres non-nulles.*

Démonstration. Soit $\mathbf{x}$ un vecteur propre de A^TA avec $A^TA\mathbf{x} = \lambda\mathbf{x}$ et $\lambda \neq 0$. Alors

$$AA^T(A\mathbf{x}) = A(A^TA\mathbf{x}) = A(\lambda\mathbf{x}) = \lambda(A\mathbf{x}).$$

Aussi, $A\mathbf{x} \neq \mathbf{0}$, car sinon on aurait $\lambda\mathbf{x} = A^T(A\mathbf{x}) = A\mathbf{0} = \mathbf{0}$. Donc $A\mathbf{x}$ est vecteur propre de AA^T avec valeur propre λ. Si $\mathbf{y}$ est vecteur propre de AA^T, l'analyse est similaire. □

Ce théorème est à comparer avec le théorème 10.9. C'est un théorème où la preuve est encore plus utile que le résultat. On y voit que étant donné des vecteurs propres de A^TA, on obtient des vecteurs propres de AA^T en multipliant par A. Il y a encore plus.

Leçon 18 : 21 novembre 2011

Théorème 12.4. *Soit A une matrice de taille $m \times n$. Soient $\mathbf{x}$ et $\mathbf{y}$ des vecteurs propres de A^TA avec $A^TA\mathbf{x} = \lambda\mathbf{x}$, $A^TA\mathbf{y} = \mu\mathbf{y}$ et $\lambda, \mu \neq 0$. Si $\mathbf{x} \perp \mathbf{y}$, alors $(A\mathbf{x}) \perp (A\mathbf{y})$.*

Démonstration. On note que $A\mathbf{x} \neq \mathbf{0}$, car sinon on aurait $\lambda = 0$; de la même manière $A\mathbf{y} \neq \mathbf{0}$. Être perpendiculaire veut donc dire que le produit scalaire est zéro. On calcule directement :

$$(A\mathbf{x}) \cdot (A\mathbf{y}) = (A\mathbf{x})^T(A\mathbf{y}) = \mathbf{x}^TA^TA\mathbf{y} = \lambda\left(\mathbf{x}^T\mathbf{y}\right) = \lambda\left(\mathbf{x} \cdot \mathbf{y}\right) = \lambda(0) = 0$$

□

La conséquence est qu'un ensemble orthogonal de vecteurs propres de A^TA devient un ensemble orthogonal de vecteurs propres de AA^T en multipliant chaque vecteur par A — pourvu que les valeurs propres correspondants sont non-nulles (autrement on obtient des vecteurs nuls en multipliant par A). Cette observation est la base de notre décomposition. La décomposition implique une matrice "diagonale" de taille $m \times n$:

$$\Sigma = \begin{bmatrix} D & 0 \\ 0 & 0 \end{bmatrix}, \tag{12.1}$$

où D est une matrice diagonale de taille $r \times r$ ($r \leq \min(m, n)$), le 0 en haut à droite est la matrice zéro de taille $r \times (n - r)$, le 0 en bas à gauche est la matrice zéro de taille $(m - r) \times r$, et le 0 en bas à droite est la matrice zéro de taille $(m - r) \times (n - r)$.

Théorème 12.5 (Décomposition en valeurs singulières)**.** *Soit A une matrice de taille $m \times n$. Alors il existe des matrices U, Σ, V telles que $A = U\Sigma V^T$ avec Σ de type (12.1), U orthogonale de taille $m \times m$, et V orthogonale de taille $n \times n$.* □

La décomposition $A = U\Sigma V^T$ est la *décomposition en valeurs singulières* de A.

Que U est orthogonale (c.-à-d. $U^TU = I$) signifie que les colonnes de U forment une base orthonormale de $\mathbb{R}^m$; que V est orthogonale (c.-à-d. $V^TV = I$) signifie que les colonnes de V forment une base orthonormale de $\mathbb{R}^n$. Comme preuve on présente l'algorithme suivant, qui *construit* les matrices requises.

Algorithme 12.6 (Décomposition en valeurs singulières)**.**
Soit A une matrice de taille $m \times n$.

(a) On calcule A^TA, et on trouve une diagonalisation orthogonale $A^TA = PDP^T$.
(b) On identifie les valeurs propres non nulles de A^TA : $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_r$.
(c) Les valeurs singulières de A sont $\sigma_1, \sigma_2, \cdots \sigma_r$, où $\sigma_j = \sqrt{\lambda_j}$.
(d) On pose Σ la matrice de taille $m \times n$ avec les valeurs singulières sur le diagonal.
(e) On pose V la matrice P.
(f) On identifie les colonnes $\mathbf{v}_1, \mathbf{v}_2, \cdots, \mathbf{v}_r$ de V, correspondant aux valeurs singulières.
(g) On pose $\mathbf{u}_j = A\mathbf{v}_j / \|A\mathbf{v}_j\| = A\mathbf{v}_j/\sigma_j$, $1 \leq j \leq r$.
(h) Si $r < m$ on étend l'ensemble $\{\mathbf{u}_1, \mathbf{u}_2, \cdots, \mathbf{u}_r\}$ à une base orthonormale de $\mathbb{R}^m$.
(i) On pose U la matrice avec les colonnes $\mathbf{u}_1, \cdots, \mathbf{u}_m$.

On a alors $A = U\Sigma V^T$. □

Il faudrait démontrer que cette algorithme fonctionne, c'est-à-dire, que les matrices U, Σ et V ont les bonnes propriétés. On voit directement que $U^TU = I$ et $V^TV = I$ car les colonnes de U forment une base orthonormale pour $\mathbb{R}^m$ et les colonnes de V forment une base orthonormale pour $\mathbb{R}^n$. Considérons le produit AV. La colonne j est $A\mathbf{v}_j$, qui est $\sigma_j\mathbf{u}_j$, qui est la colonne j du produit $U\Sigma$. Donc $AV = U\Sigma$. Multipliant par V^T, on obtient $A = U\Sigma V^T$.

Les étapes à suivre pour faire une décomposition en valeurs singulières sont des étapes qu'on connaît déjà. On souligne que la dernière étape peut se faire de deux manières. Les vecteurs $\mathbf{u}_{r+1}, \cdots \mathbf{u}_m$ forment une base pour l'espace nul de AA^T. Alternativement, c'est une base pour le complément orthogonale de $\{\mathbf{u}_1, \mathbf{u}_2, \cdots, \mathbf{u}_r\}$.

Exemple 12.7. Trouver une décomposition en valeurs singulières de $A = \begin{bmatrix} 3 & -1 \\ 1 & 3 \\ 1 & 1 \end{bmatrix}$.

On calcule premièrement $A^TA = \begin{bmatrix} 11 & 1 \\ 1 & 11 \end{bmatrix}$. On obtient $A^TA = PDP^T$ avec

$$A = PDP^T = \begin{bmatrix} 1/\sqrt{2} & 1/\sqrt{2} \\ -1/\sqrt{2} & 1/\sqrt{2} \end{bmatrix} \begin{bmatrix} 12 & 0 \\ 0 & 10 \end{bmatrix} \begin{bmatrix} 1/\sqrt{2} & 1/\sqrt{2} \\ -1/\sqrt{2} & 1/\sqrt{2} \end{bmatrix}^T$$

(voir Exemple 12.2). On pose donc

$$V = P = \begin{bmatrix} 1/\sqrt{2} & 1/\sqrt{2} \\ -1/\sqrt{2} & 1/\sqrt{2} \end{bmatrix}, \qquad \Sigma = \begin{bmatrix} \sqrt{12} & 0 \\ 0 & \sqrt{10} \\ 0 & 0 \end{bmatrix}.$$

On calcule les vecteurs $\mathbf{u}_1$ et $\mathbf{u}_2$.

$$\mathbf{u}_1 = \frac{A\mathbf{v}_1}{\sigma_1} = \frac{1}{\sqrt{12}} \begin{bmatrix} 2/\sqrt{2} \\ 4/\sqrt{2} \\ 2/\sqrt{2} \end{bmatrix} = \frac{1}{\sqrt{6}} \begin{bmatrix} 1 \\ 2 \\ 1 \end{bmatrix} \qquad \mathbf{u}_2 = \frac{A\mathbf{v}_2}{\sigma_2} = \frac{1}{\sqrt{10}} \begin{bmatrix} -4/\sqrt{2} \\ 2/\sqrt{2} \\ 0 \end{bmatrix} = \frac{1}{\sqrt{5}} \begin{bmatrix} -2 \\ 1 \\ 0 \end{bmatrix}$$

Ici on voit que $r = 2 < m = 3$. Donc il manque le vecteur $\mathbf{u}_3$. On peut procéder de deux manières. On peut chercher un vecteur orthogonale à $\mathbf{u}_1$ et $\mathbf{u}_2$, donc on met ces deux vecteurs comme rangées d'une matrice B et on cherche l'espace nul de cette matrice. On peut prendre des multiples de chaque vecteur pour simplifier les calculs.

$$B = \begin{bmatrix} 1 & 2 & 1 \\ -2 & 1 & 0 \end{bmatrix} \to \begin{bmatrix} 1 & 0 & 1/5 \\ 0 & 1 & 2/5 \end{bmatrix} \qquad \text{base pour } \ker(B) : \left\{ \begin{bmatrix} -1/5 \\ -2/5 \\ 1 \end{bmatrix} \right\}$$

Cette première méthode à une version alternative, pour ceux qui n'aiment pas lire des solutions générales dans des matrices. On forme la matrice C ayant $\mathbf{u}_1$ et $\mathbf{u}_2$ comme colonnes et on réduit la matrice $[C|I]$. Les rangées nulles à gauche indiquent quelles rangées à droite forment la base de $\ker(C^T) = \ker(B)$.

$$\left[\begin{array}{cc|ccc} 1 & -2 & 1 & 0 & 0 \\ 2 & 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 1 \end{array}\right] \to \left[\begin{array}{cc|ccc} 1 & -2 & 1 & 0 & 0 \\ 0 & 5 & -2 & 1 & 0 \\ 0 & 0 & -1/5 & -2/5 & 1 \end{array}\right] \qquad \text{base : } \left\{ \begin{bmatrix} -1/5 \\ -2/5 \\ 1 \end{bmatrix} \right\}$$

Cette base est une base orthogonale, car ce n'est qu'un seul vecteur. Donc Gram-Schmidt n'est pas requis. On normalise pour obtenir

$$\mathbf{u}_3 = \frac{1}{\sqrt{30}} \begin{bmatrix} -1 \\ -2 \\ 5 \end{bmatrix}.$$

Alternativement, on pourrait trouver $\mathbf{u}_3$ en cherchant une base orthonormale pour $\ker(AA^T)$.

$$AA^T = \begin{bmatrix} 10 & 0 & 2 \\ 0 & 10 & 4 \\ 2 & 4 & 2 \end{bmatrix} \to \begin{bmatrix} 1 & 0 & 1/5 \\ 0 & 1 & 2/5 \\ 0 & 0 & 0 \end{bmatrix} \qquad \text{base pour } \ker(AA^T) : \left\{ \begin{bmatrix} -1/5 \\ -2/5 \\ 1 \end{bmatrix} \right\}$$

Encore une fois c'est déjà une base orthogonale, donc aucun Gram-Schmidt. On normalise pour obtenir

$$\mathbf{u}_3 = \frac{1}{\sqrt{30}} \begin{bmatrix} -1 \\ -2 \\ 5 \end{bmatrix}.$$

On a maintenant la matrice U, formée des trois colonnes $\mathbf{u}_1$, $\mathbf{u}_2$ et $\mathbf{u}_3$.

$$U = \begin{bmatrix} 1/\sqrt{6} & -2\sqrt{5} & 1/\sqrt{30} \\ 2/\sqrt{6} & 1\sqrt{5} & 2/\sqrt{30} \\ 1/\sqrt{6} & 0 & -5/\sqrt{30} \end{bmatrix}$$

Ceci donne la décomposition en valeurs singulières

$$\begin{bmatrix} 3 & -1 \\ 1 & 3 \\ 1 & 1 \end{bmatrix} = \begin{bmatrix} 1/\sqrt{6} & -2\sqrt{5} & 1/\sqrt{30} \\ 2/\sqrt{6} & 1\sqrt{5} & 2/\sqrt{30} \\ 1/\sqrt{6} & 0 & -5/\sqrt{30} \end{bmatrix} \begin{bmatrix} \sqrt{12} & 0 \\ 0 & \sqrt{10} \\ 0 & 0 \end{bmatrix} \begin{bmatrix} 1/\sqrt{2} & 1/\sqrt{2} \\ -1/\sqrt{2} & 1/\sqrt{2} \end{bmatrix}^T$$

□

L'exemple était un peu long. Par contre on a calculé le vecteur $\mathbf{u}_3$ de trois manières différentes. Typiquement, une fois suffit...

La décomposition en valeurs singulières n'est pas unique. On aurait pu prendre ici $-\mathbf{u}_3$ au lieu de $\mathbf{u}_3$. Aussi, on aurait pu prendre $-\mathbf{v}_1$ au lieu de $\mathbf{v}_1$ (ceci aurait changé $\mathbf{u}_1$). En générale, si on a une valeur singulière de multiplicité plus grand que un, on pourrait avoir plusieurs bases orthogonales différentes pour l'espace propre correspondant de A^TA.

Exercice 12.8. Recalculer la décomposition en valeurs singulières de l'exemple précédent en utilisant $-\mathbf{v}_1$ au lieu de $\mathbf{v}_1$. Comparer votre décomposition avec celle ci-haut. (indice : les calculs sont très similaires, on peut recycler...). □

On se rappelle qu'on peut calculer les valeurs singulières de A en calculant les valeurs propres de la matrice A^TA ou la matrice AA^T. On peut dire plus : on peut calculer toute la décomposition en valeurs singulières par l'une ou l'autre de ces matrices. Il s'agit de "commencer à la gauche" au lieu de "commencer à la droite". Voici la version "gauche" de l'algorithme 12.9.

Algorithme 12.9 (Décomposition en valeurs singulières, version gauche)**.**

Soit A une matrice de taille $m \times n$.

(a) On calcule AA^T, et on trouve une diagonalisation orthogonale $AA^T = PDP^T$.
(b) On identifie les valeurs propres non nulles de AA^T : $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_r$.
(c) Les valeurs singulières de A sont $\sigma_1, \sigma_2, \cdots \sigma_r$, où $\sigma_j = \sqrt{\lambda_j}$.
(d) On pose Σ la matrice de taille $m \times n$ avec les valeurs singulières sur le diagonal.
(e) On pose U la matrice P.
(f) On identifie les colonnes $\mathbf{u}_1, \mathbf{u}_2, \cdots, \mathbf{u}_r$ de U, correspondant aux valeurs singulières.
(g) On pose $\mathbf{v}_j = A^T\mathbf{u}_j / \left\|A^T\mathbf{u}_j\right\| = A\mathbf{u}_j/\sigma_j$.
(h) Si $r < n$ on étend l'ensemble $\{\mathbf{v}_1, \mathbf{v}_2, \cdots, \mathbf{v}_r\}$ à une base orthonormale de $\mathbb{R}^n$.
(i) On pose V la matrice avec les colonnes $\mathbf{v}_1, \cdots, \mathbf{v}_n$.

On a alors $A = U\Sigma V^T$. □

Exercice 12.10. Calculer une décomposition en valeurs singulières de $A = \begin{bmatrix} 3 & -1 \\ 1 & 3 \\ 1 & 1 \end{bmatrix}$ en utilisant la méthode "gauche". Note que puisque $m > n$, alors la première étape est plus longue et la dernière méthode est plus courte, en comparaison avec la méthode "droite". □

12.4. **Approximation : composantes principales.** La décomposition en valeurs singulières permet une représentation plus compacte d'une matrice. Voici une première observation.

Théorème 12.11. *Soit $A = U\Sigma V^T$ une décomposition en valeurs singulières. Soit U_r la matrice formée des colonnes $\mathbf{u}_1, \cdots, \mathbf{u}_r$, V_r la matrice formée des colonnes $\mathbf{v}_1, \cdots, \mathbf{v}_r$, et Σ_r la matrice diagonale $r \times r$ formée des valeurs singulières.*

Alors $A = U_r \Sigma_r V_r^T$. □

C'est une observation basée sur le fait que des rangées nulles de Σ "annulent" les colonnes correspondant de U, et les colonnes nulles de Σ "annulent" les colonnes correspondant de V (rangées de V^T).

Exemple 12.12. On considère la décomposition ci-haut (Exemple 12.7). On vérifie que

$$\begin{bmatrix} 3 & -1 \\ 1 & 3 \\ 1 & 1 \end{bmatrix} = \begin{bmatrix} 1/\sqrt{6} & -2\sqrt{5} & 1/\sqrt{30} \\ 2/\sqrt{6} & 1\sqrt{5} & 2/\sqrt{30} \\ 1/\sqrt{6} & 0 & -5/\sqrt{30} \end{bmatrix} \begin{bmatrix} 12 & 0 \\ 0 & 10 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} 1/\sqrt{2} & 1/\sqrt{2} \\ -1/\sqrt{2} & 1/\sqrt{2} \end{bmatrix}^T$$
$$= \begin{bmatrix} 1/\sqrt{6} & -2\sqrt{5} \\ 2/\sqrt{6} & 1\sqrt{5} \\ 1/\sqrt{6} & 0 \end{bmatrix} \begin{bmatrix} 12 & 0 \\ 0 & 10 \end{bmatrix} \begin{bmatrix} 1/\sqrt{2} & 1/\sqrt{2} \\ -1/\sqrt{2} & 1/\sqrt{2} \end{bmatrix}^T$$

□

C'est une première "approximation" : on simplifie le produit matriciel en enlevant les parties qui contribuent absolument rien. Ce qui est plus utile c'est d'enlever les parties qui contribuent peu. Un résultat technique est nécessaire.

Théorème 12.13. *Si $A = U\Sigma V^T$ est une décomposition en valeurs singulières, alors $A = \sigma_1 \mathbf{u}_1 \mathbf{v}_1^T + \sigma_2 \mathbf{u}_2 \mathbf{v}_2^T + \cdots + \sigma_r \mathbf{u}_r \mathbf{v}_r^T$ (où r est le rang de A).* □

Note que le produit $\mathbf{u}_1 \mathbf{v}_1^T$ est le produit d'une matrice $m \times 1$ par une matrice $1 \times n$, donnant une matrice $m \times n$. La matrice $\sigma_j \mathbf{u}_j \mathbf{v}_j^T$ est la *j-ième composante principale* de A. On dénote par

$$\widehat{A}_t = \sigma_1 \mathbf{u}_1 \mathbf{v}_1^T + \sigma_2 \mathbf{u}_2 \mathbf{v}_2^T + \cdots + \sigma_t \mathbf{u}_t \mathbf{v}_t^T$$

la somme des t premières composantes principales. Note que $\widehat{A}_r = A$ exactement. De plus, le rang de $\widehat{A}_t$ est t. Dans un sens technique, $\widehat{A}_t$ est la meilleure approximation de A de rang t.

Leçon 19 : 24 novembre 2011

Exemple 12.14. Une expérience donne un résultat en forme matriciel $\begin{bmatrix} 1,02 & 2,03 & 4,20 \\ 0,25 & 0,51 & 1,06 \\ 1,74 & 3,46 & 7,17 \end{bmatrix}$. On cherche à savoir le rang de cette matrice. On fait une méthode de Gauss, qui donne que le

rang est 3, mais en faisant cette réduction on voit que les chiffres deviennent très petits. Est-ce que c'est peut-être de l'erreur expérimentale ? Si oui, quelle est la "vraie" matrice ?

On calcule les valeurs singulières : $\sigma_1 = 9,5213$, $\sigma_2 = 0,0071$ et $\sigma_3 = 0,0023$. Il semble que peut-être le rang est vraiment un, et que il y a eu de l'erreur expérimentale. On calcul

$$\widehat{A}_1 = \sigma_1 \mathbf{u}_1 \mathbf{v}_1^T = \begin{bmatrix} 1,0193 & 2,0280 & 4,2011 \\ 0,2567 & 0,5107 & 1,0580 \\ 1,7395 & 3,4610 & 7,1696 \end{bmatrix}.$$

C'est la matrice de rang un qui est la meilleur approximation de A. □

Exercice 12.15. Appliquer la méthode de Gauss à la matrice de l'exemple précédent. Montrer qu'on obtient deux rangées "presque" nulles.

En continuant avec la méthode de Gauss, on multiplie ces rangées par des très grands chiffres : si ces deux rangées s'agissait des erreurs expérimentales, alors on multiplierait des erreurs par des très grands chiffres ! Pour cette raison, la méthode de Gauss-Jordan n'est pas *numériquement stable*. □

Exemple 12.16. Une satellite prend plusieurs images digitales du même secteur de la Terre, chaque image captant une différente longueur d'onde (i.e., une différente "couleur"). On voudrait transmettre une image compacte (c.-à-d. transmettre la plus petite quantité de données que possible). Il y a c couleurs au total.

On forme une matrice A de taille $c \times 1000000$, où chaque rangée représente toutes les pixels d'une des images, donc une couleur. On calcule une décomposition en valeurs singulières de A. Si il y a quelques (disons t) valeurs singulières qui dominent, alors on pourrait être confiant que la matrice $\widehat{A}_t$ est une bonne approximation de A.

Le satellite pourrait transmettre alors $\widehat{A}_t$ au lieu de A. L'avantage est que

$$\widehat{A}_t = \sigma_1 \mathbf{u}_1 \mathbf{v}_1^T + \sigma_2 \mathbf{u}_2 \mathbf{v}_2^T + \cdots + \sigma_t \mathbf{u}_r \mathbf{v}_t^T.$$

Donc on pourrait transmettre t valeurs singulières, t vecteurs de taille c (les $\mathbf{u}_j$) et t vecteurs de taille 1000000 (les vecteurs $\mathbf{v}_j$). Il faut donc transmettre $t \times (1000000 + c + 1)$ chiffres au lieu de $c \times 1000000$. □

Exemple 12.17. On pourrait aussi considérer une image comme une matrice de taille 1000×1000 (chaque élément est un pixel). Si une décomposition en valeurs singulières montre que les premières t valeurs singulières dominent, alors $\widehat{A}_t$ est une bonne approximation à l'image. Il faudrait transmettre t valeurs singulières et $2t$ vecteurs de taille 1000, au lieu de 1000000 valeurs. □

13. Codes

13.1. **Introduction.** Alice et Bob veulent communiquer, mais leurs appareils (téléphone, cellulaire, radio, ordinateur...) sont imparfaites. Est-ce qu'ils peuvent communiquer sans compromettre leur conversation ? Par exemple, ils veulent se parler au téléphone, mais la ligne est imparfaite, donnant parfois du "bruit". Ou bien Alice veut transmettre à Bob ses photos de vacances sur son cellulaire, sachant que quelques bits seront renverser par l'interférence. C'est un modèle très général : si "Alice" est un disque compacte et "Bob" est un haut-parleur, alors on cherche à jouer correctement la musique même si le disque est endommagé.

Toutes ces situations se comprennent comme un désir de transmettre un message, sachant qu'il serait modifiée de façon inconnue, mais le transmettre d'une manière à ce que le récepteur peut encore correctement comprendre.

Ce n'est pas un problème nouveau : on pourrait dire que c'est le problème fondamental de communication entre humains. Les langues modernes représentent une solution : vuos pouvvait encore comprenrde le sens malgrè des ereurs, car la langue française inclut suffisamment de redondance à pouvoir détecter et même corriger ces erreurs.

La solution est donc simple, en principe : la redondance. Le problème c'est qu'on veut aussi que le message soit aussi petit que possible (e.g., la transmission d'images digitales).

13.2. **Mots, distance, boules.** On écrit les messages avec un *alphabet* de q symboles, typiquement $\{0, 1, 2, \cdots, q-1\}$. Par exemple, si $q = 2$, alors chaque symbole est un "bit". On forme des *mots* avec n symboles. Par exemple, si $q = 2$ un mot de $n = 8$ bits serait un "byte".

Les mots possibles sont tous les mots de longueur n avec q symboles ; il y a donc au total q^n mots possibles. Un *code* C est un sous-ensemble des mots possibles.

La *distance* entre deux mots est le nombre de positions dans lesquelles ils diffèrent : on écrit $d(x, y)$ pour la distance entre les deux mots x et y. La *distance minimale* du code est le minimum de la distance entre tout pair de mots dans le code ; on le dénote par δ.

Pour chaque mot x du code et chaque entier positif t, on définit la *boule* de rayon t autour de x comme l'ensemble de tous les mots à distance au plus t de x. Autrement dit, c'est l'ensemble de tous les mots qui sont à t erreurs ou moins de x. Formellement, $B_t(x) = \{z \mid d(x, z) \leq t\}$.

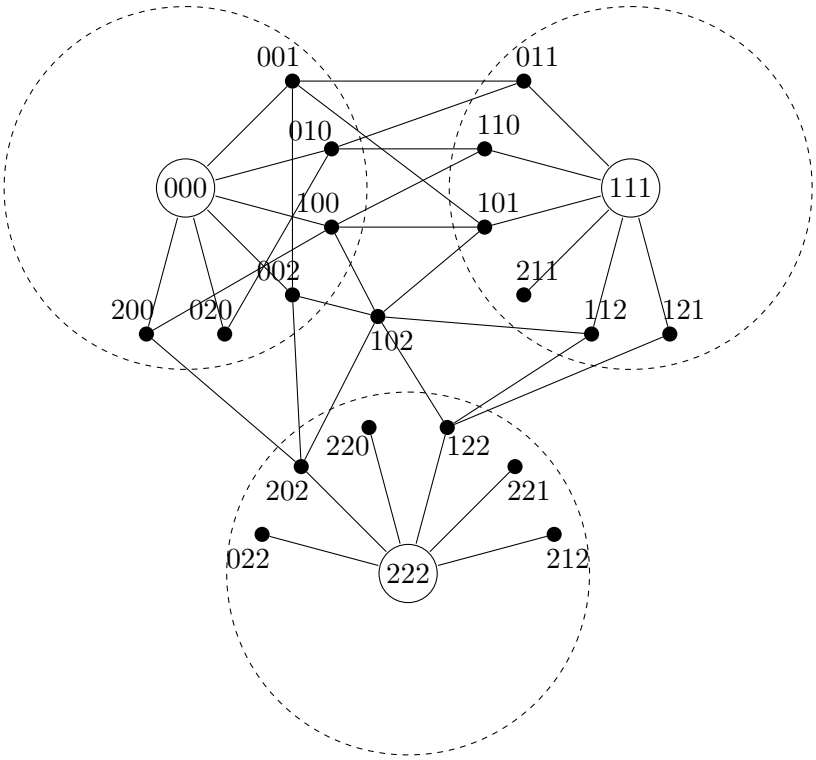
Exemple 13.1. Considérons le code suivant. On décrira ses paramètres.

$$C = \{000, 111, 222\}$$

On a $q = 3$, et $n = 3$: c'est un code de longueur 3 avec 3 symboles. On a $d(000, 222) = 3$, car ces deux mots diffèrent en trois positions. Aussi $d(000, 222) = d(111, 222) = 3$. La distance minimale est $\delta = 3$. Ici, on a $d(x, y) = 3$ pour tout mot $x \neq y$ dans C !

Avec $q = 3$ symboles et des mots de longueur $n = 3$, il y a $q^n = 3^3 = 27$ mots possibles. Il y a seulement 3 mots "permis" : le code C. La boule de rayon 1 autour de 000 est $B_1(000) = \{000, 001, 010, 100, 002, 020, 200\}$: tous les mots qui sont à une erreur proche du mot 000. □

On peut représenter un code par un graphe. Ici on indique les mots du code par un gros cercle, et les autres mots par des points. Deux mots sont reliés si leur distance est 1. On montre les trois boules $B_1(000)$, $B_1(111)$ et $B_1(222)$ comme cercles pointillés. NB : le dessin n'est *pas* complet, pour ne pas le rendre trop compliqué !



Exercice 13.2. Pour le graphe ci-haut, il manque certains mots et arrêts. Ajouter quelques-uns. Combien de mots sont à l'intérieur de chaque boule de rayon 1 ? Combien de mots ne sont contenus dans aucune boule de rayon 1 ? □

On peut comprendre ce graphe comme suit. Certains mots sont "corrects" : ce sont les mots du code : 000, 111 et 222. Certains sont incorrects, mais avec une erreur : par exemple 001. Certains sont incorrects, mais avec deux erreurs : par exemple 102.

Le graphe montre que pour le code de l'exemple 13.1 les trois boules $B_1(000)$, $B_1(111)$ et $B_1(222)$ sont disjointes. Si on transmet un mot du code, et on sait que la transmission est assez fiable pour avoir au plus une erreur, alors on sait que le résultat, même corrompu, serait encore dans la bonne boule. On pourra alors "corriger" l'erreur : un message corrompu peut être compris parfaitement par le récepteur !

On peut généraliser ces observations.

Théorème 13.3. *Si un code à une distance minimale $\delta \geq 2t+1$, alors les boules de rayon t sont toutes disjointes. Un tel code peut alors détecter et corriger jusqu'à t erreurs par mot.* □

Le code de l'exemple 13.1 a $\delta = 3$, et $3 \geq 2(1)+1$, donc ce code peut corriger 1 erreur.

Pour le code de l'exemple 13.1 il y a certains mots qui sont à distance deux de chaque mot du code (par exemple 102). Si on reçoit ce mot, on sait qu'il y a eu au moins deux erreurs. On peut, dans ce cas, détecter le fait qu'il y a eu (au moins) deux erreurs. Par contre, on ne peut pas toujours détecter deux erreurs : il se peut que 100 a soumis deux erreurs, mais on ne pourra savoir, car ce mot incorrect et plus proche à 000 (une erreur).

On peut généraliser.

Théorème 13.4. *Si un code a une distance minimale $\delta \geq 2t$, alors les boules de rayon t sont toutes disjointes sauf possiblement pour les mots à distance t du code. Un tel code peut alors détecter (mais pas nécessairement corriger) jusqu'à t erreurs par mot.* □

Le code de l'exemple 13.1 a $\delta = 3$, et $3 \geq 2(1)$, donc ce code peut détecter 1 erreur.

Exemple 13.5. Soit le code $C = \{00, 11, 22\}$, avec $n = 2$ et $q = 3$.

La distance minimale est $\delta = 2$: ceci se voit directement en calculant la liste de toutes les distances dans le code :

$$d(00,11) = 2 \qquad d(00,22) = 2 \qquad d(11,22) = 2$$

Les boules de rayon 1 ne sont pas disjointes, car $B_1(00) = \{00, 01, 10, 02, 20\}$ et $B_1(11) = \{11, 01, 10, 21, 12\}$. L'intersection est non vide : les mots $\{01, 10\}$ sont dans les deux boules.

On a $\delta = 2 \geq 2(1)$, donc ce code peut détecter une erreur. Par contre on a $\delta = 2 \geq 2(0)+1$, donc se peut corriger 0 erreur. On pourra savoir qu'un message est corrompu, mais on pourra savoir comment l'interpréter. C'est consistant avec les observations sur les boules : un mot comme 01 est visiblement en erreur, mais on ne peut pas décider *quelle* est l'erreur. □

Exercice 13.6. Faire un graphe du code de l'exemple précédant. Inclure tous les mots possibles et les boules de rayon 1. □

Exercice 13.7. Soit le code $C = \{0000, 1111\}$, avec $n = 4$ et $q = 2$. Déterminer la distance minimale. Faire un graphe des mots possibles, incluant les boules de rayon 1. (Si le graphe est trop grand, donner une partie seulement.) Est-ce que les boules de rayon 1 sont disjointes ? De rayon 2 ? Combien d'erreurs est-ce que ce code peut détecter ? Combien d'erreurs est-ce que ce code peut corriger ? □

Exercice 13.8. Soit le code $C = \{0000, 0011, 1122\}$, avec $n = 4$ et $q = 3$. Déterminer la distance minimale. Faire un graphe des mots possibles, incluant les boules de rayon 1. (Si le graphe est trop grand, donner une partie seulement.) Est-ce que les boules de rayon 1 sont disjointes ? De rayon 2 ? Combien d'erreurs est-ce que ce code peut détecter ? Combien d'erreurs est-ce que ce code peut corriger ? □

Exercice 13.9. Donner un exemple d'un code de distance minimale 5. Préciser n et q pour votre exemple. Faire un graphique (partiel au besoin !) des mots possibles. Inclure les boules de rayon t, ou t est aussi grand que possible pour que les boules soient disjointes. Combien d'erreurs est-ce que ce code peut détecter ? Combien d'erreurs est-ce que ce code peut corriger ? □

Exercice 13.10. Soit un code de distance minimale 5. Combien d'erreurs peut-il détecter ? Combien d'erreurs peut-il corriger ?

Si un code peut détecter 3 erreurs, mais peut corriger seulement 2 erreurs, alors quelle est sa distance minimale ? □

13.3. **Borne de Hamming.** Dans le code C de l'exemple 13.1, la distance minimale est $\delta = 3$ qui donne que les boules de rayon 1 sont disjointes. Chaque boule contient 7 mots (un mot du code et 6 mots à une erreur proche). Donc il y a au moins $3 \times 7 = 21$ mots au total (en réalité il y a 27 mots).

Considérons un code C de longueur n avec q symboles. Pour chaque mot du code il y a $n(q-1)$ mots qui sont à une erreur proche (n positions qui pourront changer à $q-1$ symboles différents). De plus il y a $\binom{n}{2}(q-1)^2$ mots qui sont à deux erreurs proche ($\binom{n}{2}$ combinaisons de 2 positions qui pourront chacun changer à $q-1$ symboles différents). En générale, le nombre de mots dans une boule est

$$|B_t(x)| = 1 + n(q-1) + \binom{n}{2}(q-1)^2 + \cdots + \binom{n}{t}(q-1)^t.$$

Si toutes les boules sont disjointes, alors la somme de $|B_t(x)|$ pour chaque x dans le code est au plus q^n. Le nombre de boules est exactement égal au nombre de mots dans le code, $|C|$.

Définition 13.11. Si $z \in \mathbb{R}$, alors $\lfloor z \rfloor$ est l'entier le plus grand qui est inférieur ou égal à z.

Théorème 13.12. *Soit C un code de longueur n avec q symboles et de distance minimale δ. Soit $t = \lfloor (\delta-1)/2 \rfloor$. Alors*

$$|C| \leq \frac{q^n}{1 + n(q-1) + \binom{n}{2}(q-1)^2 + \cdots + \binom{n}{t}(q-1)^t}.$$

L'expression à la droite est la Borne de Hamming *pour le code.*

Démonstration. Si un code a une distance minimale δ, alors les boules de rayon t seront disjointes, où $t = \lfloor (\delta-1)/2 \rfloor$. Le dénominateur est donc exactement le nombre de mots dans une boule. Le numérateur est le nombre de mots possibles. Le nombre de boules, multiplié par le nombre de mots dans une boule, ne peut pas dépasser le nombre de mots possibles. □

Un code tel que l'inégalité dans théorème 13.12 est une égalité est un *code parfait*.

Exemple 13.13. Considérons le code de l'exemple 13.1. Ici on a $\delta = 3$, donc avec $t = 1$ on a les boules de rayon 1 disjointes. De plus, $|B_1(x)| = 1 + 3(3-1) = 7$. Donc

$$|C| \leq \frac{q^n}{|B_1(x)|} \qquad \rightsquigarrow \qquad 3 \leq \frac{3^3}{7} = \frac{27}{7}.$$

Puisque $3 < 27/7$ ce code n'est pas parfait. □

Si un code n'est pas parfait, alors il existe des mots ambigus : on ne sait pas comment les interpréter. Dans le code de l'exemple 13.1, le récepteur ne sait pas comment interpréter le mot 012 : évidemment il y a eu au moins deux erreurs, mais si on accepte la possibilité de deux erreurs dans un mot alors le mot 001 devient ambigu aussi.

Exercice 13.14. Pour les codes des exercices ci-haut, lesquels sont parfaits ? □

Exercice 13.15. Donner un exemple d'un code parfait de distance minimale 3, et un code parfait de distance minimale 5. (Indice : $q = 2$.) □

Leçon 20 : 28 novembre 2011

14. Corps finis

14.1. **Codes efficaces.** Si Alice et Bob communiquent avec un code, alors Alice envoie un mot y du code à Bob, qui ne reçoit pas exactement y, mais plutôt z, qui est proche à y. Que fait Bob ? Il a une liste de tous les mots du code, et de toutes les boules et toutes les mots possibles (incluant son z). Il regarde donc dans sa liste pour savoir dans quelle boule se trouve z. Cette boule correspond à y, et donc il sait que Alice lui a envoyé y et non z.

Le travail de Alice est simple : elle envoie son mot. Bob a une tâche plus difficile : il doit chercher pour z dans une liste complète, chaque fois qu'il reçoit un mot d'Alice. Il y a une méthode beaucoup plus efficace, mais pour le comprendre il faut comprendre l'algèbre linéaire sur un corps fini.

14.2. **Corps.** Un corps et un ensemble de "nombres" avec lesquels on peut faire de l'arithmétique. Quelques exemples que vous connaissez déjà sont $\mathbb{R}$ (les réels), $\mathbb{C}$ (les complexes), $\mathbb{Q}$ (les rationnels). Il y a (beaucoup !) d'autres.

Formellement un ensemble $\mathbb{K}$ avec une addition et une multiplication est un *corps* si les propriétés suivantes sont valides pour tout $a, b, c \in \mathbb{K}$.

1) $a + b \in \mathbb{K}$
2) $ab \in \mathbb{K}$

3) $a + b = b + a$
4) $ab = ba$
5) $a + (b + c) = (a + b) + c$
6) $a(bc) = (ab)c$
7) $a(b + c) = ab + ac$

En addition, les propriété suivantes doivent être valides.

8) Il existe un élément $0 \in \mathbb{K}$ tel que $0 + a = a$ pour tout $a \in \mathbb{K}$.
9) Il existe un élément $1 \in \mathbb{K}$ tel que $1a = a$ pour tout $a \in \mathbb{K}$.
10) Pour chaque a, il existe un élément $-a \in \mathbb{K}$ avec $a + (-a) = 0$ ($-a$ est l'*inverse additif* de a).
11) Pour chaque $a \neq 0$, il existe un élément $a^{-1} \in \mathbb{K}$ avec $a\,(a^{-1}) = 1$ ($-a$ est l'*inverse multiplicatif* de a).

Par exemple, ai a et b sont des fractions, alors $a + b$ est une fraction aussi : c'est Propriété 1 pour $\mathbb{Q}$. Si on considère deux nombres complexes a et b, alors $ab = ba$: c'est Propriété 4 pour $\mathbb{C}$. Une inverse additif est aussi connue comme la négative d'un chiffre ; l'inverse multiplicatif est la réciproque.

Exemple 14.1. L'ensemble des entiers $\mathbb{Z}$ n'est *pas* un corps. Par exemple, 2 n'a pas d'inverse multiplicatif dans $\mathbb{Z}$. Quelles propriétés sont valides ? Lesquelles ne sont pas ? □

14.3. **Arithmétique modulo n.** Soit n un entier positif fixe. Pour $s, r \in \mathbb{Z}$, on dit que s et r sont *équivalents modulo* n, et on écrit "$s \equiv r \mod n$", si $s - r$ est divisible par n. Si s est n'importe quel entier, on peut trouver t et r tels que $s = tn + r$ avec $0 \leq r < n$. C'est exactement la division : t est le quotient et r et le reste, obtenu en divisant s par n. On voit que s est équivalent à r modulo n. On peut comprendre ceci d'une autre manière : en commençant avec s, on soustrait (ou additionne) des multiples de n afin de la réduire à un chiffre positif inférieur à n. On parle donc parfois de *réduction* modulo n.

Exemple 14.2. Voici quelques calculs modulo 7 à titre d'exemple. On fait les calculs avec des entiers ordinaires, pour ensuite "réduire" modulo 7.

$$\begin{aligned} 3 + 6 = 9 = 1(7) + 2 \equiv 2 \mod 7 \\ 3 \times 5 = 15 \equiv 2(7) + 1 \equiv 1 \mod 7 \\ 5 \times (2 + 4) = 5 \times 6 = 30 = 4(7) + 2 \equiv 2 \mod 7 \\ 2 \times (1 - 6) = 2 \times (-5) = -10 = -2(7) + 4 \equiv 4 \mod 7 \end{aligned}$$

Typiquement on réserve "=" pour une égalité entre entiers (réels) et $\equiv$ pour une égalité entre entiers modulo n. Parfois on n'écrit pas le " mod 7" si le contexte le rend clair. □

On dénote par $\mathbb{Z}_n$ l'ensemble de chiffres non négatifs inférieurs à n, avec l'arithmétique modulo n. Donc $\mathbb{Z}_n = \{0, 1, 2, \cdots, n - 1\}$, avec les calculs modulo n.

Exercice 14.3. Est-ce que $\mathbb{Z}_n$ est un corps, avec l'addition et la multiplication modulo n ? Est-ce que les propriétés sont satisfaits ? (On verra la réponse très bientôt.) □

Exemple 14.4. On peut construire les tables d'addition et de multiplication pour $\mathbb{Z}_n$. Ce serait des tables *au complet*, car $\mathbb{Z}_n$ ne contient que n éléments ($\mathbb{R}$ contient une infinité d'éléments, donc une table au complet et impossible !). Voici pour $\mathbb{Z}_5$.

$+$	0	1	2	3	4
0	0	1	2	3	4
1	1	2	3	4	$\boxed{0}$
2	2	3	4	0	1
3	3	4	0	1	2
4	4	0	1	2	3

$\times$	0	1	2	3	4
0	0	0	0	0	0
1	0	1	2	3	4
2	0	2	4	$\boxed{1}$	3
3	0	3	1	4	2
4	0	4	3	2	1

Par exemple, les valeurs en boîte s'expliquent avec les calculs suivants.

$$1+4=5=1(5)+0\equiv 0 \mod 5$$
$$2\times 3=6=1(5)+1\equiv 1 \mod 5$$

On découvre que $1+4\equiv 0 \mod 5$. Donc 4 est la négative de 1, car $-1\equiv 4 \mod 5$. On a aussi que $2\times 3\equiv 1 \mod 5$. Donc 3 est la réciproque de 2, car $2^{-1}\equiv 3 \mod 5$. □

Exercice 14.5. Vérifier les tables d'addition et de multiplication données pour $\mathbb{Z}_5$. Trouver l'inverse additif de 2 dans la table d'addition (dans la rangée de 2, chercher la valeur 0 : la colonne correspondante donne l'inverse additif de 2). Trouver l'inverse additif de chaque élément de $\mathbb{Z}_5$, si possible. Trouver l'inverse multiplicatif de chaque élément non-nul de $\mathbb{Z}_5$, si possible. □

Exercice 14.6. Est-ce que $\mathbb{Z}_5$ est un corps ? L'exercice précédant se montre utile, car il faudrait que tout élément de $\mathbb{Z}_5$ possède une inverse additif (Propriété 10), et que tout élément non-nul de $\mathbb{Z}_5$ possède une inverse multiplicatif (Propriété 11). □

L'ensemble $\mathbb{Z}_2$ est un cas intéressant. On a $\mathbb{Z}_2=\{0,1\}$ avec l'addition et la multiplication modulo 2. Note qu'il n'y a pas beaucoup d'arithmétique modulo 2.

Exemple 14.7. Si a est un entier pair, alors $a\equiv 0 \mod 2$, et si b est un entier impair, alors $b\equiv 1 \mod 2$. Donc on peut comprendre l'arithmétique en $\mathbb{Z}_2$ comme l'arithmétique de la parité : si le résultat est pair alors c'est zéro, et si le résultat est impair alors c'est 1. En particulier on a $1+1\equiv 0 \mod 2$.

Exercice 14.8. Construire les tables d'addition et de multiplication pour $\mathbb{Z}_2$. Trouver les inverses additifs de chaque élément de $\mathbb{Z}_2$. Trouver les inverses multiplicatifs de chaque élément non-nul de $\mathbb{Z}_2$. Montrer que "additionner" et "soustraire" sont des synonymes pour des éléments de $\mathbb{Z}_2$. Montrer que "multiplier" et "diviser" (c.-à-d. multiplier par l'inverse multiplicatif) sont des synonymes pour les éléments non-nuls de $\mathbb{Z}_2$. □

L'ensemble $\mathbb{Z}_2$ représente l'arithmétique des ordinateurs : c'est une chose très pratique. De plus, c'est un corps.

Exercice 14.9. Montrer directement que $\mathbb{Z}_2$ est un corps. C'est-à-dire, pour chaque propriété, montrer directement selon vos tables que la propriété est satisfaite pour tout $a,b,c\in\mathbb{Z}_2$. (Note que "tout $a,b,c\in\mathbb{Z}_2$ n'est pas beaucoup...). □

Par contre l'arithmétique modulo n ne donne pas toujours un corps.

Exemple 14.10. L'ensemble $\mathbb{Z}_4$ n'est pas un corps. Si c'était un corps, on aurait

$$0 \equiv 4 = 2 \cdot 2 \implies 2^{-1} \cdot 0 = 2^{-1}(2 \cdot 2) \implies 0 = (2^{-1} \cdot 2)2 \implies 0 = 1 \cdot 2 = 2,$$

qui est une contradiction. On voit que Propriété 11 n'est pas satisfait (l'élément 2 ne peut pas avoir un inverse). Est-ce que ce contre-exemple est unique ? □

Un nombre n est *premier* si il possède exactement deux diviseurs positifs : 1 et n. Donc $2, 3, 17$ sont premiers, tandis que $1, 4, 437$ ne sont pas.

Théorème 14.11. *L'ensemble Z_p avec l'arithmétique modulo p est un corps lorsque p est un nombre premier. Si n n'est pas un nombre premier, alors $\mathbb{Z}_n$ n'est pas un corps, car Propriété 11 n'est pas satisfaite (un diviseur non-trivial de n ne possède pas d'inverse multiplicatif).* □

En général, il existe un corps fini avec q éléments si et seulement si $q = p^k$ pour un nombre premier k. Donc il existe un corps ayant 4 éléments, mais ce n'est pas $\mathbb{Z}_4$! On ne démontrera pas se résultat fondamental : on se contente de travailler avec les corps $\mathbb{Z}_p$.

14.4. **Algèbre linéaire en $\mathbb{Z}_p$.** On peut faire l'algèbre linéaire avec n'importe quel corps. Il s'agit de l'algèbre "ordinaire", mais avec l'arithmétique du corps.

Exemple 14.12. Résoudre $2x = 3$ sur le corps $\mathbb{R}$, sur le corps $\mathbb{Z}_2$ et sur le corps $\mathbb{Z}_5$.

$$\begin{aligned}
\mathbb{R}: &\quad 2x = 3 \implies (2^{-1})2x = (2^{-1})3 \implies x = 3/2 && \text{solution : } x = 3/2\\
\mathbb{Z}_2: &\quad 2x \equiv 3 \implies 0x \equiv 1 \implies 0 \equiv 1 && \text{aucune solution}\\
\mathbb{Z}_5: &\quad 2x \equiv 3 \implies (2^{-1})2x \equiv (2^{-1})3 \implies (3)2x \equiv (3)3 \implies 1x \equiv 4 && \text{solution : } x \equiv 4
\end{aligned}$$

On s'étonne peut-être qu'il n'y a aucune solution en $\mathbb{Z}_2$: une équation linéaire a "toujours" une solution, non ? En fait, non : l'équation $0x = 1$ n'a pas de solution en $\mathbb{R}$. L'équation donnée n'est pas vraiment en $\mathbb{Z}_2$, car "2" et "3" ne sont pas dans $\mathbb{Z}_2$. Donc il a fallu réduire premièrement, et on découvre que cette équation est donc "$0x = 1$" (qui ne possède aucune solution peut importe le corps). □

Exercice 14.13. Solutionner $3x - 4 = x$ sur $\mathbb{Z}_5$, et aussi sur $\mathbb{Z}_2$.

Donner la liste de toutes les équations linéaires de la forme $ax + b = 0$ sur $\mathbb{Z}_2$. Solutionner chaque équation individuellement. □

On peut manipuler des matrices sur $\mathbb{Z}_p$ comme sur $\mathbb{R}$. Les opérations de rangées, les pivots, le rang, la dimension, les combinaisons linéaires, l'indépendance... tout est "pareil" sauf qu'on utilise l'arithmétique modulo p au lieu de l'arithmétique de $\mathbb{R}$.

Exemple 14.14. Soit la matrice $A = \begin{bmatrix} 1 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 1 \end{bmatrix}$. Donner sa forme échelonnée réduite sur $\mathbb{R}$, sur $\mathbb{Z}_5$ et sur $\mathbb{Z}_2$.

$$\mathbb{R}: \begin{bmatrix} 1 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 1 \end{bmatrix} \xrightarrow[R_2 \to R_2 + (-1)R_1]{} \begin{bmatrix} 1 & 1 & 0 \\ 0 & -1 & 1 \\ 0 & 1 & 1 \end{bmatrix} \xrightarrow[R_3 \to R_3 + (1)R_2]{R_1 \to R_1 + (1)R_2} \begin{bmatrix} 1 & 0 & 1 \\ 0 & -1 & 1 \\ 0 & 0 & 2 \end{bmatrix}$$
$$\xrightarrow[R_3 \to (\frac{1}{2})R_3]{R_2 \to (-1)R_2} \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & -1 \\ 0 & 0 & 1 \end{bmatrix} \xrightarrow[R_1 \to R_1 + (-1)R_3]{R_2 \to R_2 + (1)R_3} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$\mathbb{Z}_5: \begin{bmatrix} 1 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 1 \end{bmatrix} \xrightarrow[R_2 \to R_2 + (-1)R_1]{} \begin{bmatrix} 1 & 1 & 0 \\ 0 & 4 & 1 \\ 0 & 1 & 1 \end{bmatrix} \xrightarrow[R_3 \to R_3 + (-4)R_2]{R_1 \to R_1 + (-4)R_2} \begin{bmatrix} 1 & 0 & 1 \\ 0 & 4 & 1 \\ 0 & 0 & 2 \end{bmatrix}$$
$$\xrightarrow[R_3 \to (2^{-1})R_3]{R_2 \to (4^{-1})R_2} \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 4 \\ 0 & 0 & 1 \end{bmatrix} \xrightarrow[R_1 \to R_1 + (-1)R_3]{R_2 \to R_2 + (1)R_3} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$\mathbb{Z}_2: \begin{bmatrix} 1 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 1 \end{bmatrix} \xrightarrow[R_2 \to R_2 + R_1]{} \begin{bmatrix} 1 & 1 & 0 \\ 0 & 1 & 1 \\ 0 & 1 & 1 \end{bmatrix} \xrightarrow[R_1 \to R_1 + R_2]{R_3 \to R_3 + R_2} \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 1 \\ 0 & 0 & 0 \end{bmatrix}$$

Note que pour les opérations sur $\mathbb{Z}_5$, "$_{R_1 \to R_1 + (-4)R_2}$" est la même chose que "$_{R_1 \to R_1 + 1R_2}$". Sur le corps $\mathbb{Z}_2$, il y a seulement deux sortes d'opérations : "$_{R_j \to R_j + R_i}$" et "$_{R_i \rightleftharpoons R_j}$".

On a que le rang de A sur $\mathbb{R}$ ou $\mathbb{Z}_5$ est 3, tandis que le rang de A sur $\mathbb{Z}_2$ est 2. □

Exercice 14.15. Expliquer pourquoi les seules opérations de rangée sur $\mathbb{Z}_2$ sont d'ajouter une rangée à une autre, ou d'échanger deux rangées : "$_{R_j \to R_j + R_i}$" et "$_{R_i \rightleftharpoons R_j}$". □

Exemple 14.16. Est-ce que $\mathbf{u} = \begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix}$ est vecteur propre de $A = \begin{bmatrix} 1 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 1 \end{bmatrix}$ sur $\mathbb{R}$? Sur $\mathbb{Z}_2$?

$$\mathbb{R}: A\mathbf{u} = \begin{bmatrix} 1 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 1 \\ 2 \\ 1 \end{bmatrix} \neq \lambda \mathbf{u}$$

$$\mathbb{Z}_2: A\mathbf{u} = \begin{bmatrix} 1 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix} = 1\mathbf{u}$$

Le vecteur $\mathbf{u}$ n'est pas un vecteur propre de A sur $\mathbb{R}$; par contre, c'est un vecteur propre avec valeur propre $\lambda = 1$ sur $\mathbb{Z}_2$. □

Leçon 21 : 1 decembre 2011

15. Codes linéaires

15.1. **Introduction.** Considérons les mots de longueur n en symboles $\{0, 1, \cdots, p-1\}$, où p est premier (on utilise p au lieu de q ici pour souligner le fait que ce nombre est premier). Dénotons par V l'ensemble de tous les mots possibles. Pour $n = 3$ et $p = 3$ on aura

$$\begin{aligned} V &= \{abc \mid a, b, c \in \{0, 1, 2\}\} \\ &= \{000, 001, \cdots, 102, 110, \cdots, 222\}. \end{aligned}$$

L'ensemble V est un espace vectoriel de dimension n sur le corps $\mathbb{Z}_p$. On peut voir ceci en écrivant les mots comme vecteurs. On écrira les vecteurs en rangée au lieu d'en colonne. C'est pour des raisons pratiques : on aura souvent à écrire des "grands" vecteurs, qui sont plus compactes en rangée qu'en colonne.

$$\begin{aligned} V &= \left\{ \begin{bmatrix} a & b & c \end{bmatrix} \mid a, b, c \in \mathbb{Z}_3 \right\} \\ &= \left\{ \begin{bmatrix} 0 & 0 & 0 \end{bmatrix}, \begin{bmatrix} 0 & 0 & 1 \end{bmatrix}, \cdots, \begin{bmatrix} 1 & 0 & 2 \end{bmatrix}, \begin{bmatrix} 1 & 1 & 0 \end{bmatrix}, \cdots, \begin{bmatrix} 2 & 2 & 2 \end{bmatrix} \right\} \end{aligned}$$

On peut construire des espaces vectoriels sur le corps $\mathbb{Z}_p$ comme sur le corps $\mathbb{R}$. Pour $n = 3$ et $p = 3$ on a $V = {\mathbb{Z}_p}^n = {\mathbb{Z}_3}^3$.

Le code $C = \{000, 111, 222\}$ de l'exemple 13.1 est en sous-espace de ${\mathbb{Z}_3}^3$. On peut dire même plus : $C = \operatorname{span}\left\{ \begin{bmatrix} 1 & 1 & 1 \end{bmatrix} \right\}$ sur le corps $\mathbb{Z}_2$, car :

$$\begin{aligned} \operatorname{span}\left\{ \begin{bmatrix} 1 & 1 & 1 \end{bmatrix} \right\} &= \left\{ t \begin{bmatrix} 1 & 1 & 1 \end{bmatrix} \mid t \in \mathbb{Z}_3 \right\} \\ &= \left\{ t \begin{bmatrix} 1 & 1 & 1 \end{bmatrix} \mid t \in \{0, 1, 2\} \right\} \\ &= \left\{ 0 \begin{bmatrix} 1 & 1 & 1 \end{bmatrix}, 1 \begin{bmatrix} 1 & 1 & 1 \end{bmatrix}, 2 \begin{bmatrix} 1 & 1 & 1 \end{bmatrix} \right\} \\ &= \left\{ \begin{bmatrix} 0 & 0 & 0 \end{bmatrix}, \begin{bmatrix} 1 & 1 & 1 \end{bmatrix}, \begin{bmatrix} 2 & 2 & 2 \end{bmatrix} \right\} \end{aligned}$$

On voit qu'un "mot" n'est nul autre qu'un vecteur écrit sans les parenthèse et les espaces : "$\begin{bmatrix} 1 & 1 & 1 \end{bmatrix}$" est équivalent à "111". Pour cet raison on écrira parfois des vecteurs comme des mots, et parlera de faire des opérations algébriques sur des "mots".

Le fait que C est un sous-espace d'un espace vectoriel permet une approche plus efficace pour Alice et Bob.

15.2. **Codes linéaires : matrice génératrice.** Un *code linéaire* est un code qui est un sous-espace vectoriel de l'espace vectoriel ${\mathbb{Z}_p}^n$. Comme tout autre sous-espace, un code possède des bases. Si on écrit les vecteurs de la base d'un code comme rangées d'une matrice, on a une *matrice génératrice* pour ce code. Note que typiquement il y aura plusieurs bases pour un sous-espace ; on préfère une base qui donne une matrice génératrice en forme échelonnée réduite. Une telle base, écrite comme rangées d'une matrice, donne une *matrice génératrice standard*.

Un code linéaire est donc l'espace rangée d'une matrice. On peut faire des opérations de rangée sur une matrice sans changer son espace-rangée, donc on peut toujours trouver une matrice génératrice standard en faisant une réduction par rapport aux lignes.

Une matrice génératrice "génère" le code : le code est exactement l'ensemble des vecteurs obtenus comme combinaison linéaire des rangées de G. Donc le code C généré par la matrice G (de taille $m \times n$) est exactement $C = \{\mathbf{x}G \mid \mathbf{x} \in {\mathbb{Z}_p}^m\}$.

Exemple 15.1. Soit le code $C = \{0000, 1100, 0011, 1111\}$, avec $n = 4$, $p = 2$.

On peut voir que c'est un code linéaire, car on peut trouver un ensemble engendrant. Il y en a plusieurs, par exemple, $\{1100, 1111\}$ comme montre les calculs suivants.

$$\begin{aligned} C &= \operatorname{span}\{1100, 1111\} \\ &= \{a(1100) + b(1111) \mid a, b \in \mathbb{Z}_2\} \\ &= \{0(1100) + 0(1111),\ 0(1100) + 1(1111),\ 1(1100) + 0(1111),\ 1(1100) + 1(1111)\} \\ &= \{0000, 1111, 0011, 0011\} \end{aligned}$$

De plus, $\{1100, 1111\}$ est linéairement indépendant, donc c'est une base pour ce code. On écrit ici "1100" pour le vecteur "$\begin{bmatrix}1 & 1 & 0 & 0\end{bmatrix}$".

On a donc une matrice génératrice pour ce code :

$$\begin{bmatrix}1 & 1 & 0 & 0\\ 1 & 1 & 1 & 1\end{bmatrix}$$

On peut trouver une matrice génératrice standard en faisant une opération de rangée :

$$\begin{bmatrix}1 & 1 & 0 & 0\\ 1 & 1 & 1 & 1\end{bmatrix} \xrightarrow[R_2 \to R_2 + 1R_1]{} \begin{bmatrix}1 & 1 & 0 & 0\\ 0 & 0 & 1 & 1\end{bmatrix}$$

On a la matrice génératrice standard $G = \begin{bmatrix}1 & 1 & 0 & 0\\ 0 & 0 & 1 & 1\end{bmatrix}$.

Le code est généré par G, car $C = \{\mathbf{x}G \mid \mathbf{x} \in \mathbb{R}^2\}$. Les quatre possibilités pour $\mathbf{x}$ sont $\{00, 01, 10, 11\}$; en multipliant par G on obtient les quatre mots du code.

$$\begin{bmatrix}0 & 0\end{bmatrix}\begin{bmatrix}1 & 1 & 0 & 0\\ 0 & 0 & 1 & 1\end{bmatrix} = \begin{bmatrix}0 & 0 & 0 & 0\end{bmatrix}$$

$$\begin{bmatrix}0 & 1\end{bmatrix}\begin{bmatrix}1 & 1 & 0 & 0\\ 0 & 0 & 1 & 1\end{bmatrix} = \begin{bmatrix}0 & 0 & 1 & 1\end{bmatrix}$$

$$\begin{bmatrix}1 & 0\end{bmatrix}\begin{bmatrix}1 & 1 & 0 & 0\\ 0 & 0 & 1 & 1\end{bmatrix} = \begin{bmatrix}1 & 1 & 0 & 0\end{bmatrix}$$

$$\begin{bmatrix}1 & 1\end{bmatrix}\begin{bmatrix}1 & 1 & 0 & 0\\ 0 & 0 & 1 & 1\end{bmatrix} = \begin{bmatrix}1 & 1 & 1 & 1\end{bmatrix}$$

□

Exercice 15.2. Montrer que le code généré par $\begin{bmatrix}1 & 1 & 0 & 0\\ 1 & 1 & 1 & 1\end{bmatrix}$ est le même code que celui généré par la matrice génératrice standard ci-haut. □

Exemple 15.3. Soit le code généré par la matrice $\begin{bmatrix}1&1&1&0&0&0\\0&0&0&1&1&1\end{bmatrix}$.

La liste de tous les mots du code est obtenue en pré-multipliant G par tous les vecteurs dans ${\mathbb{Z}_2}^2$. On obtient alors le code $\{000000, 000111, 111000, 111111\}$. La distance minimale est $\delta = 3$. Donc les boules de rayon 1 sont toutes disjointes, et ce code peut détecter et corriger une erreur par mot. □

Une matrice génératrice indique comment transmettre des messages. Dans le code d'Exemple 15.3, la matrice génératrice standard et

$$G = \begin{bmatrix}1&1&1&0&0&0\\0&0&0&1&1&1\end{bmatrix}.$$

Donc, le code est

$$C = \{\mathbf{x}G \mid \mathbf{x} \in {\mathbb{Z}_2}^2\}.$$

Alice écrit son message en termes des mots $\{00, 01, 10, 11\}$ dans ${\mathbb{Z}_2}^2$. Elle veut transmettre son message à Bob. Pour chaque mot $\mathbf{x}$, elle envoie le mot correspondant du code $\mathbf{y} = \mathbf{x}G$. On imagine que les mots $\mathbf{x}$ représentent le message original, sans aucune redondance : tout pixel compte. Les mots $\mathbf{y}$ représentent la même information, mais plus dispersé : il y a une redondance.

Exemple 15.4. Alice veut envoyer le message "00, 11, 01, 01" à Bob. Alice et Bob ont déjà décidé que leurs communications se feront avec le code C généré par $\begin{bmatrix}1&1&1&0&0&0\\0&0&0&1&1&1\end{bmatrix}$. Alice fait les calculs suivants.

$$\mathbf{x} = \begin{bmatrix}0&0\end{bmatrix} \rightsquigarrow \mathbf{y} = \begin{bmatrix}0&0\end{bmatrix}\begin{bmatrix}1&1&1&0&0&0\\0&0&0&1&1&1\end{bmatrix} = \begin{bmatrix}0&0&0&0&0&0\end{bmatrix}$$

$$\mathbf{x} = \begin{bmatrix}1&1\end{bmatrix} \rightsquigarrow \mathbf{y} = \begin{bmatrix}1&1\end{bmatrix}\begin{bmatrix}1&1&1&0&0&0\\0&0&0&1&1&1\end{bmatrix} = \begin{bmatrix}1&1&1&1&1&1\end{bmatrix}$$

$$\mathbf{x} = \begin{bmatrix}0&1\end{bmatrix} \rightsquigarrow \mathbf{y} = \begin{bmatrix}0&1\end{bmatrix}\begin{bmatrix}1&1&1&0&0&0\\0&0&0&1&1&1\end{bmatrix} = \begin{bmatrix}0&0&0&1&1&1\end{bmatrix}$$

$$\mathbf{x} = \begin{bmatrix}0&1\end{bmatrix} \rightsquigarrow \mathbf{y} = \begin{bmatrix}0&1\end{bmatrix}\begin{bmatrix}1&1&1&0&0&0\\0&0&0&1&1&1\end{bmatrix} = \begin{bmatrix}0&0&0&1&1&1\end{bmatrix}$$

Donc Alice envoie "000000, 111111, 000111, 000111". □

Il y a une observation utile pour les codes linéaires. Si $\mathbf{x}_1, \mathbf{x}_2$ sont deux mots (vecteurs) dans un code linéaire (sous-espace) alors $d(\mathbf{x}_1, \mathbf{x}_2) = d(\mathbf{x}_1 - \mathbf{x}_2, \mathbf{0})$. La distance entre deux mots et égale à la distance entre leur différence et le vecteur zéro. Si $d(\mathbf{x}_1, \mathbf{x}_2) = \delta$, la distance minimale du code, alors $d(\mathbf{x}_1 - \mathbf{x}_2, \mathbf{0}) = \delta$ aussi. Puisqu'il s'agit d'un code linéaire, $\mathbf{x}_1 - \mathbf{x}_2$ est également un mot du code. Donc afin de trouver la distance minimale, il suffit de considérer la distance entre un mot et $\mathbf{0}$. La distance $d(\mathbf{x}, \mathbf{0})$ est le nombre de symboles non-nuls dans $\mathbf{x}$, dit le *poids* de $\mathbf{x}$. On écrit $w(\mathbf{x})$ pour le poids de $\mathbf{x}$. Le *poids minimal* du code est le minimum du poids de tous les mots non-nuls du code.

Théorème 15.5. *Si C est un code linéaire, alors la distance minimale du code est égale au poids minimale du code.* □

L'avantage de ce résultat est qu'on peut plus rapidement calculer la distance minimale : ce n'est que nécessaire de considérer tous les mots, et non tous les paires de mots. (Souvent on peut calculer la distance minimale d'un code linéaire plus directement, en comprenant sa structure particulière.)

Exemple 15.6. Pour le code de l'exemple 15.3, on peut calculer toutes les distances :

$$d(000000, 000111) = 3 \qquad d(000000, 111000) = 3 \qquad d(000000, 111111) = 6$$
$$d(111000, 111111) = 3 \qquad d(000111, 111111) = 3 \qquad d(000111, 111000) = 6$$

On voit que la distance minimale est 3. On aurait pu aussi calculer la liste de tous les poids.

$$w(000111) = 3 \qquad w(111000) = 3 \qquad w(111111) = 6$$

Le poids minimale est 3, donc la distance minimale est aussi 3. □

15.3. **Matrice de contrôle.** Bob ne reçoit pas exactement les mots du code que Alice envoie (à cause du "bruit" dans la transmission). Il a besoin d'une méthode d'interpréter les mots reçus.

Si G est une matrice génératrice standard, alors on peut la comprendre comme étant une matrice de la forme $[I_m|A]$: les colonnes de I_m correspondent au colonnes de G avec pivot (ce ne sont pas nécessairement au début), la matrice A représente le "reste" de G. Si G est une matrice de taille $m \times n$, alors A est de taille $m \times (n-m)$. Donc la matrice identité est de taille $m \times m$, ce qui explique la notation I_m.

La *matrice de contrôle*, H, est obtenue en écrivant la matrice $-A^T$ dans les colonnes correspondant aux pivots de G, et une matrice identité dans les colonnes correspondant aux non-pivots de G. Puisque A^T est de taille $(n-m) \times m$, alors H a $n-m$ rangées est la matrice identité ici est de taille $(n-m) \times (n-m)$. Donc on a $H = [-A^T|I_{n-m}]$, de taille $(n-m) \times n$.

La matrice R (la *matrice de "récupérage"*) est obtenue en remplaçant la matrice A dans G avec des zéros. C'est la même taille que G, donc $m \times n$.

Exemple 15.7. On considère le code de l'exemple 15.3. On construit la matrice H en mettant $-A^T$ dans le colonnes pivot de G, et en remplissant le reste avec une identité. On indique les

colonnes correspondant au pivots de G en **gras**.

$$G = \begin{bmatrix} \mathbf{1} & 1 & 1 & \mathbf{0} & 0 & 0 \\ \mathbf{0} & 0 & 0 & \mathbf{1} & 1 & 1 \end{bmatrix}$$

$$A = \begin{bmatrix} & 1 & 1 & & 0 & 0 \\ & 0 & 0 & & 1 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \end{bmatrix}$$

$$-A^T = \begin{bmatrix} \mathbf{1} & & & \mathbf{0} & & \\ \mathbf{1} & & & \mathbf{0} & & \\ \mathbf{0} & & & \mathbf{1} & & \\ \mathbf{0} & & & \mathbf{1} & & \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 1 & 0 \\ 0 & 1 \\ 0 & 1 \end{bmatrix}$$

$$H = \begin{bmatrix} \mathbf{1} & 1 & 0 & \mathbf{0} & 0 & 0 \\ \mathbf{1} & 0 & 1 & \mathbf{0} & 0 & 0 \\ \mathbf{0} & 0 & 0 & \mathbf{1} & 1 & 0 \\ \mathbf{0} & 0 & 0 & \mathbf{1} & 0 & 1 \end{bmatrix}$$

$$R = \begin{bmatrix} \mathbf{1} & 0 & 0 & \mathbf{0} & 0 & 0 \\ \mathbf{0} & 0 & 0 & \mathbf{1} & 0 & 0 \end{bmatrix}$$

Note que les matrices G, H, R ont tous $n = 6$ colonnes. Les matrices G et R sont de la même taille, 2×6. Mais la matrice H est de taille 4×6.

Ce code est un code sur le corps $\mathbb{Z}_2$. Donc la négation de A^T se fait par rapport au corps $\mathbb{Z}_2$. Mais dans ce corps $-1 \equiv 1$, donc $-A^T = A^T$ (voir l'exercice 14.8). □

Les matrices H et R sont utiles pour interpréter les mots reçus.

Théorème 15.8. *Si H est la matrice de contrôle qui correspond à la matrice génératrice G, alors $HG^T = \mathbf{0}$. Autrement dit, les rangées de H sont tous orthogonales aux rangées de G. Comme conséquence, si $\mathbf{y} = \mathbf{x}G$ est n'importe quel mot du code, alors $H\mathbf{y}^T = \mathbf{0}$.*

Démonstration. On observe que $HG^T = [-A^T|I]\left([I|A]\right)^T = -A^T I + IA^T = \mathbf{0}$ (ici on utilise le fait que les positions des colonnes de la matrice $-A^T$ dans H correspondent aux positions des colonnes de la matrice I dans G). De plus, $H\mathbf{y}^T = H\left(\mathbf{x}G\right)^T = HG^T\mathbf{x}^T = \mathbf{0}\mathbf{x}^T = \mathbf{0}$. □

Une façon de comprendre ceci est que le noyau de H est l'espace rangée de G.

L'application est directe. Bob reçoit un mot $\mathbf{z}$, qui n'est peut-être pas exactement le mot $\mathbf{y}$ envoyé par Alice. Il calcul $H\mathbf{z}^T$: si c'est $\mathbf{0}$, alors $\mathbf{z}$ *est* un mot du code : aucune erreur. Si $H\mathbf{z}^T \neq \mathbf{0}$ alors il y a eu une erreur.

Exemple 15.9. Rappelant l'exemple 15.4, Bob reçoit le message suivant : "000001, 111111, 000111, 010111". Il dénote ces quatre mots par $\mathbf{z}_1$, $\mathbf{z}_2$, $\mathbf{z}_3$, $\mathbf{z}_4$.

Il connaît la matrice génératrice du code, et aussi la matrice de contrôle (car il a bien étudié l'exemple 15.7). Donc il calcul :

$$H\mathbf{z}_1^T = \begin{bmatrix}0\\0\\0\\1\end{bmatrix} \implies \mathbf{z}_1 \text{ est incorrect} \qquad H\mathbf{z}_2^T = \begin{bmatrix}0\\0\\0\\0\end{bmatrix} \implies \mathbf{z}_2 \text{ est correct}$$

$$H\mathbf{z}_3^T = \begin{bmatrix}0\\0\\0\\0\end{bmatrix} \implies \mathbf{z}_3 \text{ est correct} \qquad H\mathbf{z}_4^T = \begin{bmatrix}1\\0\\0\\0\end{bmatrix} \implies \mathbf{z}_4 \text{ est incorrect}$$

Bob sait alors que $\mathbf{z}_1$ et $\mathbf{z}_4$ sont en erreur, mais $\mathbf{z}_2$ et $\mathbf{z}_3$ sont corrects. □

Leçon 22 : 5 decembre 2011

15.4. **Syndrome.** La matrice de contrôle sert à reconnaître les mots incorrects, mais elle sert aussi à les corriger. On imagine que Alice envoie $\mathbf{y}$ à Bob, mais Bob reçoit $\mathbf{z}$. On peut écrire $\mathbf{z}$ comme $\mathbf{z} = \mathbf{y} + \mathbf{e}$, où $\mathbf{e}$ est le vecteur de l'erreur. En fait $\mathbf{e} = \mathbf{z} - \mathbf{y}$, mais Bob ne connaît (pour l'instant) ni $\mathbf{e}$ ni $\mathbf{y}$. On voit que

$$H\mathbf{z}^T = H\left(\mathbf{y} + \mathbf{e}\right)^T = H\mathbf{y}^T + H\mathbf{e}^T = \mathbf{0} + H\mathbf{e}^T = H\mathbf{e}^T.$$

On dit que $H\mathbf{z}^T$ est le *syndrome* de $\mathbf{z}$. Bob ne peut pas voir l'erreur $\mathbf{e}$, mais il peut connaître le syndrome *de l'erreur*, car $H\mathbf{e}^T = H\mathbf{z}^T$.

Le nombre d'erreurs possibles est relativement petit : Il s'agit de tous les vecteurs $\mathbf{e}$ avec au plus t éléments non-nuls, pour un code qui peut détecter et corriger t erreurs par mot.

Exemple 15.10. Si un code de longueur n sur le corps $\mathbb{Z}_p$ peut détecter et corriger t erreurs, alors le nombre de vecteurs d'erreur est

$$n(p-1) + \binom{n}{2}(p-1)^2 + \cdots + \binom{n}{t}(p-1)^t$$

On à déjà vu cette formule : c'est le nombre de mots dans la boule de rayon t (sans inclure le mot du code lui-même). Voir le théorème 13.12 et la discussion qui la précède. □

Exemple 15.11. Pour le code de l'exemple 15.3, la distance minimale est $\delta = 3$, donc ce code peut corriger une erreur par mot. C'est un code sur le corps $\mathbb{Z}_2$, donc les erreurs possibles sont les vecteurs dans ${\mathbb{Z}_2}^6$ ayant un élément non-nul. Il y en a six. On donne tous les erreurs possibles, ainsi que les syndromes correspondants (les vecteurs sont écrits comme mots pour

conserver l'espace).

$$\begin{aligned}
\mathbf{e}_1 = 000001 \quad &\to \quad \text{syndrome} : H\mathbf{e}_1 = 0001 \\
\mathbf{e}_2 = 000010 \quad &\to \quad \text{syndrome} : H\mathbf{e}_2 = 0010 \\
\mathbf{e}_3 = 000100 \quad &\to \quad \text{syndrome} : H\mathbf{e}_3 = 0011 \\
\mathbf{e}_4 = 001000 \quad &\to \quad \text{syndrome} : H\mathbf{e}_4 = 0100 \\
\mathbf{e}_5 = 010000 \quad &\to \quad \text{syndrome} : H\mathbf{e}_5 = 1000 \\
\mathbf{e}_6 = 100000 \quad &\to \quad \text{syndrome} : H\mathbf{e}_6 = 1100
\end{aligned}$$

□

L'exemple précédant montre une chose très pratique : si un code sur $\mathbb{Z}_2$ corrige une erreur, alors les syndromes sont exactement les colonnes de H. La tâche de Bob est simplifié : il calcul le syndrome de chaque mot reçu. Si le syndrome est $\mathbf{0}$, alors le mot est correct. Si le syndrome n'est pas zéro, alors le syndrome est une des colonnes de H. De plus, la colonne de H correspond exactement à l'erreur ! Donc il sait $\mathbf{e}$ et il peut donc calculer $\mathbf{y} = \mathbf{z} - \mathbf{e}$ et récupérer le message original.

Le principe est pareil en général, par contre les erreurs possibles sont tous les vecteurs de ${\mathbb{Z}_p}^n$ ayant poids au plus $\lfloor(\delta - 1)/2\rfloor$. Donc il y a plus de syndromes ! Pour les codes à grande distance minimales, on cherche des autres méthodes de "décoder", mais pour nous, l'approche simple suffira.

Exemple 15.12. Rappelant l'exemple 15.9, Bob a reçu le message "000001, 111111, 000111, 010111" et il a calculé les syndromes "0001, 0000, 0000, 1000".

Le premier syndrome 0001 est exactement la dernière colonne de H. Donc c'est le dernier symbole de ce mot qui est en erreur. Donc Bob sait que 000001 était vraiment 000000.

Le deuxième syndrome est 0000, donc le deuxième mot est correct.

Le troisième syndrome est 0000, donc le troisième mot est correct.

Le quatrième syndrome est 1000, qui est deuxième colonne de H. Donc c'est le deuxième symbole de ce mot qui est en erreur. Donc Bob sait que 010111 était vraiment 000111.

Bob sait alors que les mots transmis par Alice étaient "000000, 111111, 000111, 000111". □

Bob a corrigé le message. Il veut maintenant récupérer le message original d'Alice. C'est la matrice R qui fait ceci.

Théorème 15.13. *Soit un code généré par G, avec matrice de contrôle H et matrice de "récupérage" R. Si $\mathbf{y} = \mathbf{x}G$ est un mot du code, alors $\mathbf{x}^T = R\mathbf{y}^T$ (ou $\mathbf{x} = \mathbf{y}R^T$).*

Démonstration. On a directement que $RG^T = I$. Donc $R\mathbf{y}^T = R\left(\mathbf{x}G\right)^T = RG^T\mathbf{x}^T = I\mathbf{x}^T = \mathbf{x}^T$.
□

Exemple 15.14. Rappelant l'exemple 15.12, Bob a reçu le message "000001, 111111, 000111, 010111", il a calculer les syndromes "0001, 0000, 0000, 1000", et il a corrigé pour trouver les mots transmis par Alice "000000, 111111, 000111, 000111". Il peut maintenant calculer le message original.

$$\begin{aligned} \mathbf{y}_1 = 000000 \quad &\Longrightarrow \quad \mathbf{x}_1 = \mathbf{y}_1 R^T = 00 \\ \mathbf{y}_2 = 111111 \quad &\Longrightarrow \quad \mathbf{x}_2 = \mathbf{y}_1 R^T = 11 \\ \mathbf{y}_3 = 000111 \quad &\Longrightarrow \quad \mathbf{x}_3 = \mathbf{y}_1 R^T = 01 \\ \mathbf{y}_4 = 000111 \quad &\Longrightarrow \quad \mathbf{x}_4 = \mathbf{y}_1 R^T = 01 \end{aligned}$$

Bob sait alors que le message original de Alice était "00, 11, 01, 01". □

Algorithme 15.15. Soit un code linéaire avec matrice génératrice G de taille $m \times n$ sur un corps $\mathbb{Z}_p$. On calcule la matrice de contrôle H et la matrice de "récupérage" R.

Si Alice veut envoyer un message à Bob, elle suit les étapes suivants.

(a) Alice écrit son message original en termes de mots de longueur m sur $\mathbb{Z}_p$: chaque mot est un vecteur $\mathbf{x}$ dans l'espace vectoriel ${\mathbb{Z}_p}^m$.

(b) Pour chaque mot $\mathbf{x}$, Alice calcul $\mathbf{y} = \mathbf{x}G$; c'est un mot du code, sous-espace de ${\mathbb{Z}_p}^n$.

(c) Alice envoie chaque $\mathbf{y}$ à Bob.

Si Bob reçoit un message de Alice, il suit les étapes suivants.

(a) Bob reçoit des mots de Alice ; ce sont des mots $\mathbf{z}$ en ${\mathbb{Z}_p}^n$.

(b) Pour chaque mot $\mathbf{z}$, Bob calcul le syndrome $H\mathbf{z}^T$.

(c) Si le syndrome est $\mathbf{0}$, alors le mot est correct, et $\mathbf{y} = \mathbf{z}$.

(d) Si le syndrome n'est pas $\mathbf{0}$, alors le mot est incorrect. Il connaît tous les erreurs possibles. Parmi ces erreurs, il choisi celui qui à le même syndrome que $\mathbf{z}$. C'est cette erreur $\mathbf{e}$ qui a perturbé le mot $\mathbf{y}$, donc il peut maintenant calculer $\mathbf{y} = \mathbf{z} - \mathbf{e}$.

(e) Bob connaît maintenant $\mathbf{y}$, le mot correct transmis par Alice. Il obtient $\mathbf{x} = \mathbf{y}R^T$. □

Note que Bob doit passer à travers une liste de tous les erreurs possibles. C'est beaucoup plus rapide que de passer à travers une liste de tous les mots possibles. Mais si le nombre d'erreurs par mot est plus élevé, on a besoin d'autres techniques.

Le message original est écrit en ${\mathbb{Z}_p}^m$; les mots du code sont écrit en ${\mathbb{Z}_p}^n$. On dit que m est la *dimension* du code et que n et la *longueur* du code. On a déjà vu la longueur, mais dans le contexte des codes linéaires on comprend que "dimension" mesure la quantité d'information dans le message original. Le nombre de mots possibles dans le code est exactement p^m. La "longueur" mesure la quantité de redondance ajoutée pour donner une message transmissible (et corrigeable, au besoin).

Exercice 15.16. En utilisant encore le même code que l'exemple 15.3, Bob renvoie un message à Alice. Alice reçoit "100111, 000101, 111111, 110111". Quel est le message original de Bob ? Donner les syndromes, les mots corrigés et le message original. □

Exercice 15.17. Les photos prises par les sondes spatiales *Voyager* ont été transmises à la Terre en utilisant un code de Golay. Ce code prend des mots en ${\mathbb{Z}_2}^{12}$ (donc 12 bits) et les transforme en mot du code qui est un sous-espace de ${\mathbb{Z}_2}^{24}$ (donc 24 bits). La distance minimale de ce code est 8. Donner les tailles des matrices G, H et R. Combien d'erreurs par mot est-ce que ce code peut détecter ? Combien d'erreurs par mot est-ce que ce code peut corriger ?

Combien de syndromes distinctes sont possibles avec ce code ? Bob devrait passer à travers cette liste pour chaque mot qu'il reçoit. Est-ce que vous pensez que c'est une méthode efficace pour un code de distance minimale $\delta = 8$? Note que c'est encore beaucoup plus rapide que de passer à travers une liste de tous les mots possibles dans ${\mathbb{Z}_2}^{24}$. Néanmoins, il existe des méthodes même plus rapides de "décoder" ce code. □

15.5. **Codes de Hamming.** Un *code de Hamming* est un code linéaire parfaite. On considère les codes de Hamming *binaire*, voulant dire sur le corps $\mathbb{Z}_2$. On obtient sa matrice de contrôle en écrivant tous les vecteurs non-nuls de ${\mathbb{Z}_2}^k$ comme colonnes, pour un entier k. On peut alors déterminer G et R.

Exemple 15.18. Pour $k = 3$, la matrice de contrôle du code de Hamming est

$$H = \begin{bmatrix} 0 & 1 & 1 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 & 1 \end{bmatrix}.$$

On calcule alors que

$$G = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{bmatrix},$$

$$R = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix}.$$

On voit que la dimension est $m = 4$ et la longueur est $n = 7$ (c'est la taille de G, 4×7). En général, $k = n - m$, $n = 2^k - 1$ et $m = 2^k - k - 1$.

Ici, on a choisi d'écrire la matrice H avec la matrice identité à la fin. Ce n'est pas strictement nécessaire, il suffit de l'écrire dans un ordre où l'on peut "voir" la matrice identité. Mais dans l'ordre choisi, on aura automatiquement une matrice génératrice standard. □

On peut montrer que le poids minimal du code de Hamming est 3. C'est vrai en toute dimension. Donc ce code peut détecter et corriger une erreur par mot. Ce code comporte 2^m mots. On a que $|C|\,|B_1| = (2^m)\,(1+n) = 2^{n-k}2^k = 2^n$. La borne d'un code parfaite est atteinte.

Exercice 15.19. Alice veut envoyer le message "0001, 0100, 0110" à Bob en utilisant un code de Hamming binaire de dimension $m = 4$. Donner les mots du code qu'elle transmet. □

Exercice 15.20. Bob reçoit le message "0011010, 1100110, 1110100", envoyé selon le code Hamming binaire de dimension $m = 4$. Déterminer si les mots sont corrects. Si nécessaire, les corriger. Donner le message original (en ${\mathbb{Z}_2}^4$). □

Exercice 15.21. Donner la dimension m et la longueur m pour le code Hamming avec $k = 2$. Donner les matrices G, H et R pour ce code. Faire un graphe de ce code, montrant les boules de rayon 1. Vérifier que les boules sont toutes disjointes et que le code est parfait. □

Exercice 15.22. Donner la dimension m et la longueur m pour le code Hamming avec $k = 4$. Donner les tailles de matrices G, H et R. Les décrire (c'est peut-être un peu long de les écrire explicitement). □

Index