Exercices sur la partie SON du cours de traitement numérique du signal

Gabriel Dauphin

November 27, 2024

Exercice 1 *rrx129 On considère un signal x_n d'une durée $d = 1s$ et échantillonné à $fe = 8kHz$. Ce signal est découpé en $K = 5$ trames avec un taux de recouvrement de $\alpha = 0.2$. Une fois découpé il est noté y_{kn} avec $k \in \{0, \dots, 4\}$. On cherche à simuler la deuxième trame $k = 1$. On note N_K le nombre d'échantillons dans chaque trame.*

1. *Combien y a-t-il d'échantillons dans le signal ?*
2. *Donnez en fonction de N_K le nombre d'échantillons dans la première trame qui ne sont pas présents dans la trame suivante. On suppose ici que la dernière trame ne contient que ce nombre d'échantillons. En déduire une relation entre K, α, N_K, N ? Montrez que $N_K = \frac{N}{K(1-\alpha)}$. Combien vaut N_K ?*
3. *Donnez une formule calculant les temps associées aux différents indices de la seconde trame.*
4. *Calculez $t^{(de)}$, $t^{(ce)}$ et $t^{(fin)}$ correspondant au temps associé au premier indice, à l'indice du milieu et au dernier indice de la trame.*
5. *Donnez le pseudo-code permettant à partir de x_n de trouver y_{1n}.*

Solution :

1. Le nombre d'échantillons est $N = \frac{d}{T_e} = dfe$.
2. La deuxième trame commence un peu avant la fin de la première trame, elle recouvre αN_K échantillons, il en reste $(1-\alpha)N_K$ non-recouverts. Chacune des K trames ont exactement $(1-\alpha)N_K$ échantillons non-recouverts par la suivante et comme la dernière est ici un peu plus courte on a $\frac{K(1-\alpha)}{N_K} = N$. D'où $N_K = \frac{N}{K(1-\alpha)} = \frac{df_e}{K(1-\alpha)} = 2000$.
3.
   - $t^{(\text{de})} = (1-\alpha)N_K T_e = \frac{0.8\times 2000}{8000} = 0.2\text{s}$.
   - $t^{(\text{fin})} = t^{(\text{de})} + (N_K - 1)T_e = 0.2 + \frac{2000-1}{8000} = 0.2 + 0.25 - 1.25\times 10^{-4} = 0.45 - 1.25\times 10^{-4}$.
   - $t^{(\text{ce})} = t^{(\text{de})} + 0.5 N_K T_e = 0.2 + 0.125 = 0.325\text{s}$.
4.
```
alpha = 0.2
N_K = 2000
FOR n = 0 TO 1999 DO
y_1n[n] = x_n[n+(1-alpha)*N_K]
END_FOR
```

Exercice 2 (x130) *On considère une base de sons notées* (x_n, y_n) *où chaque valeur de* n *désigne un élément de cette base,* x_n *est un son et* y_n *est une étiquette qui vaut un, deux ou trois,* N *est le nombre total de sons. Dans cet exercice, on s'intéresse à une autre technique de classification qui n'a pas été vue en cours, qui est parfois appelée arbre aléatoire. L'apprentissage consiste à choisir aléatoirement* G *attributs distincts et pour chaque attribut choisi, de choisir un seuil. Ensuite les éléments de la base d'apprentissage sont répartis dans les* 2^G *regroupements en fonction des résultats de seuillage vis-à-vis des attributs et des seuils. Pour chaque élément de la base, on note* $\mathbf{z}_n$ *le vecteur binaire de taille* G *contenant les résultats de ces seuillages (*$\mathbf{z} = 1 \dots 1$ *signifie que pour chaque seuil, l'attribut est inférieur à ce seuil). Pour chaque regroupement, on note l'étiquette majoritaire au sein du regroupement. L'étiquette majoritaire du regroupement* $\mathbf{z}$ *est notée* $L(\mathbf{z})$*. La prédiction consiste d'abord à déterminer le vecteur binaire* $\mathbf{z}$ *associée à la requête* x*, l'étiquette prédite est alors celle donnée par* $L(\mathbf{z})$*. On se place ici dans le cas d'une base de* $N = 20$ *sons avec trois attributs* PMCT*,* ZCR *et* FMOY *pour moyenne de la puissance moyenne, moyenne de la première harmonique et moyenne de la fréquence moyenne.*

1. *Donnez un pseudo-code de l'algorithme d'apprentissage.*
2. *Donnez un pseudo-code de la prédiction.*

Solution :

1. Ici le choix aléatoire des attributs n'est pas utile car l'ordre dans lequel ces attributs sont choisis n'a pas d'importance.

```
Choisir aléatoirement les seuils S_PMCT, S_ZCR, S_FMOY.
FOR n = 1 TO 20 DO
  Calculer PMCT(x_n)
  IF PMCT(x_n)<S_PMCT, z_n[0] = 1, ELSE z_n[0] = 0, END_IF
  Calculer ZCR(x_n)
  IF ZCR(x_n)<S_ZCR,
  z_n[1] = 1, ELSE z_n[1] = 0, END_IF
  Calculer FMOY(x_n)
  IF FMOY(x_n)<S_FMOY,
  z_n[2] = 1, ELSE z_n[2] = 0, END_IF
END_FOR
FOR b = 000 TO 111 DO
  v_1 = 0, v_2 = 0, v_3 = 0
  FOR n = 1 TO 20 DO
    IF v_1 == y_n, v_1 += 1
    ELSE_IF 2 == y_n, v_2 += 1
    ELSE v_3 += 1
    END_IF
  END_FOR
  IF v_1 >= v_2 AND v_1 >= v_3, L(b) = 1, END_IF
  IF v_2 >= v_1 AND v_2 >= v_3, L(b) = 2, END_IF
  IF v_3 >= v_1 AND v_3 >= v_2, L(b) = 3, END_IF
END_FOR
```

2. On note x le son appelé requête dont on cherche à déterminer l'étiquette y. On dispose de S_PMCT, S_ZCR, S_FMOY et de L.

```
Calculer PMCT(x), ZCR(x), FMOY(x).
IF PMCT(x) <= S_PMCT, b[0] = 1, ELSE b[0] = 0, END_IF
IF ZCR(x)  <= S_ZCR,  b[1] = 1, ELSE b[1] = 0, END_IF
IF FMOY(x) <= S_FMOY, b[2] = 1, ELSE b[2] = 0, END_IF
Déterminer y = L(b)
```

Exercice 3 (x127) *On considère un son décrit par les valeurs suivantes et utilisant une fréquence d'échantillonnage de* 8kHz *:*

$$\begin{array}{ccccc} x_0 = 2 & x_1 = 1 & x_2 = 2 & x_3 = 3 & x_4 = 2 \\ x_5 = 1 & x_6 = 1 & x_7 = 3 & x_8 = 1 & x_9 = 2 \\ x_{10} = 1 & x_{11} = 3 & & & \end{array} \tag{1}$$

On réalise un découpage de ce son en trois trames sans chevauchement.

1. *Donnez les valeurs pour ce signal de* $K, T_K, N_K, t_1^{(de)}, t_1^{(fin)}, t_1^{(ce)}, t_{1,2}, \alpha$ *À défaut des valeurs exactes, vous pouvez donner les opérations qu'il y aurait à effectuer sur une calculatrice pour obtenir les valeurs.*

2. *On effectue ici une normalisation de l'amplitude du son par une constante sans y inclure un centrage du signal ni une détection du début et de la fin du son, avec pour objectif que la puissance moyenne de l'ensemble du son soit de* 1*. Indiquez les nouvelles valeurs du signal.*

3. *On considère un deuxième son* $y_0 \dots y_{11}$ *qui est tout le temps égale à* 1*. On considère une distance* $\mathbf{d}$ *entre les sons* x *et* y *définis par*

$$\mathbf{d}(x,y) = \sqrt{\frac{1}{K}\sum_{k=0}^{K-1} \left(PMCT_{x,k} - PMCT_{y,k}\right)^2}$$

où PMCT *est la puissance moyenne à court terme calculée sur chaque trame.*

$$PMCT_{x,k} = \frac{1}{N_K}\sum_{n=0}^{N_K-1} x_{k,n}^2$$

Normalement cette distance est calculée sur les signaux normalisés notés $\tilde{x}$ *et* $\tilde{y}$*, mais ici pour simplifier, on calcule cette distance sur les signaux non-normalisés. Calculez une valeur numérique de* $\mathbf{d}(x,y)$

Simulation :

```
tab=ceil(3*rand(1,12));
for k=1:length(tab)
  if 0==mod(k,5) str='\\'; else str='&'; end
  disp(['x_',num2str(k-1),'=',num2str(tab(k)),str]),
end
N_K=4; fe=8e3; T_k=N_K/fe,
2*T_K-T_e,
(T_k+(2*T_K-T_e))/2,
T_k+2/fe,
```

Solution :

1. $N = 3$, $K = 3$, $T_K = N_K T_e = 5{\times}10^{-4}$s. $t_1^{(\text{de})} = T_K = 5{\times}10^{-4}$s, $t_1^{(\text{fin})} = t_2^{(\text{de})} - T_e = 2T_K - T_e = 8.75{\times}10^{-4}$s. $t_1^{(\text{ce})} = (t_1^{(\text{de})} + t_1^{(\text{fin})})/2 = 6.875{\times}10^{-4}$s, $t_{1,2} = T_k + 2T_e = 7.5{\times}10^{-4}$s. $\alpha = 0$

2. La puissance est donnée par

$$P = \frac{1}{12}\sum_{n=0}^{11} x_n^2 = \frac{1}{12}\left(9{\times}3 + 4{\times}4 + 5{\times}1\right) = 4$$

La normalisation est obtenue en divisant par deux.

$$\begin{array}{ccccc} \tilde{x}_0 = 1 & \tilde{x}_1 = 0.5 & \tilde{x}_2 = 1 & \tilde{x}_3 = 1.5 & \tilde{x}_4 = 1 \\ \tilde{x}_5 = 0.5 & \tilde{x}_6 = 0.5 & \tilde{x}_7 = 1.5 & \tilde{x}_8 = 0.5 & \tilde{x}_9 = 1 \\ \tilde{x}_{10} = 0.5 & \tilde{x}_{11} = 1.5 & & & \end{array} \tag{2}$$

3.

$$\begin{array}{ccc} \mathrm{PMCT}_{x,1} = 4.5 & \mathrm{PMCT}_{x,2} = 3.75 & \mathrm{PMCT}_{x,3} = 3.75 \\ \mathrm{PMCT}_{y,1} = 1 & \mathrm{PMCT}_{y,2} = 1 & \mathrm{PMCT}_{y,3} = 1 \end{array}$$

$$\mathbf{d}(x,y) = \sqrt{\frac{1}{3}\left(3.5^2 + 2.75^2 + 2.75^2\right)} \approx 3.02$$

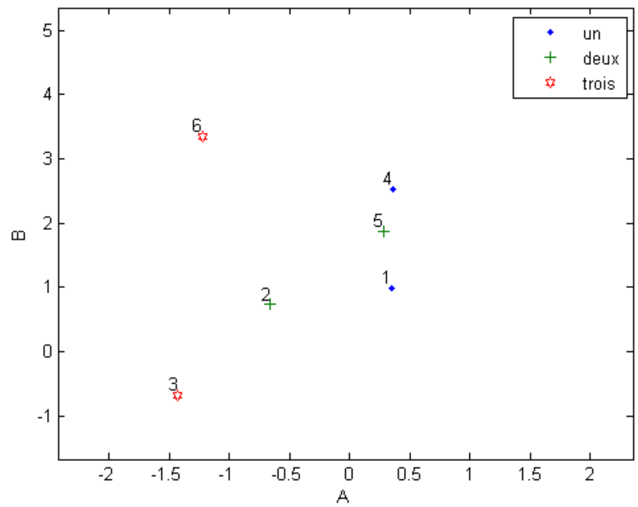

Figure 1: Visualisation des valeurs prises par les descripteurs pour chacun des six sons. Exercice 4

Exercice 4 (x118) *On considère six sons numérotés de 1 à 6. Le véritable sens de chacun des sons est décrit* $y_i \in \{1, 2, 3\}$ *:* $y_i = 1$ *signifie que le son i correspond au mot* un, $y_i = 2$ *correspond à* deux *et* $y_i = 3$ *correspond à* trois. *On suppose ici qu'après un traitement numérique complexe, on est parvenu à donner deux descripteurs* A_i *et* B_i *et à calculer leur valeur numérique pour chacun des sons (il s'agit ici pour chaque descripteur, d'une valeur pour l'ensemble du son). On considère ici comme technique de prédiction, la sélection du plus proche voisin au sens de la norme euclidienne en utilisant le descripteur A d'abord puis A et B ensuite. L'objet de cet exercice est de déterminer l'utilité des classifieurs obtenus. Le tableau suivant ainsi que la figure 1 montre les valeurs des descripteurs pour chaque son.*

son	A	B	y
son 1	0.34663	0.99315	1
son 2	-0.66365	0.73033	2
son 3	-1.4282	-0.67675	3
son 4	0.35464	2.5326	1
son 5	0.27736	1.8714	2
son 6	-1.227	3.3521	3

1. *On utilise les trois premiers sons pour la base d'apprentissage et les trois sons suivants pour la base de test. Pour la classification, on considère ici que le descripteur A. Déterminez la prédiction de chaque son de la base de test. Déterminez la matrice de confusion, la sensibilité globale. Déterminez aussi la précision et le rappel pour chaque classe.*

2. *On utilise maintenant les deux descripteurs et on réalise une validation croisée en deux lots, calculez la sensibilité globale.*

Simulation Matlab ayant permis de fabriquer le graphique.

```
y=[1:3 1:3]';
A=zeros(6,1); B=zeros(6,1);
A(y==1)=-0.2+0.5*randn(2,1);
A(y==2)=0.2+randn(2,1);
A(y==3)=1+2*randn(2,1);
B(y==1)=1+randn(2,1);
B(y==2)=1.5+randn(2,1);
```

```
B(y==3)=3*randn(2,1);
for i=1:6
  ligne=['son ',num2str(i),' & ',num2str(A(i)),' & ',num2str(B(i)),' & ',num2str(y(i)),'\\[0.2cm]'];
  disp(ligne),
end
figure(1);
plot(A(y==1),B(y==1),'.',A(y==2),B(y==2),'+',A(y==3),B(y==3),'h');
legend('un','deux','trois');
xlabel('A'),ylabel('B');
ea=0.04*(max(A)-min(A)); eb=0.04*(max(B)-min(B));
for i=1:6
  text(A(i)-ea,B(i)+eb,num2str(i));
end
axis([min(A)-1 max(A)+2 min(B)-1 max(B)+2]);
```

1. Les sons 4,5 et 6 font partie de la base de test. Pour déterminer les valeurs prédites, on utilise que l'indicateur A (c'est l'énoncé) et on cherche le son dans la base d'apprentissage dont la valeur A est la plus proche de celle prise par chacun des sons tests.

$$\widehat{x}_4 = 1,\ \widehat{x}_5 = 1,\ \widehat{x}_6 = 3$$

   Ces prédictions permettent de déterminer exactement la matrice de confusion obtenue qui est

$$C = \begin{bmatrix} 1 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

   La sensibilité globale OA $= \frac{C_{11}+C_{22}+C_{33}}{\sum_{ij} C_{ij}} = \frac{2}{3}$.
   Pour la classe associée à *un*, $P_1 = \frac{C_{11}}{C_{11}+C_{21}+C_{31}} = \frac{1}{2}$. $R_1 = \frac{C_{11}}{C_{11}+C_{12}+C_{13}} = 1$.
   Pour la classe associée à *deux*, $P_2 = \frac{C_{22}}{C_{12}+C_{22}+C_{32}}$ n'est pas défini. $R_2 = \frac{C_{22}}{C_{21}+C_{22}+C_{23}} = 0$.
   Pour la classe associée à *trois*, $P_3 = \frac{C_{33}}{C_{13}+C_{23}+C_{33}} = 1$. $R_3 = \frac{C_{33}}{C_{31}+C_{32}+C_{33}} = 1$.

2. La validation croisée se fait d'abord en considérant pour la base d'apprentissage les trois premiers sons et pour la base de test les trois autre sons.

$$\widehat{x}_4 = 1,\ \widehat{x}_5 = 1,\ \widehat{x}_6 = 2$$

   Il y a ici une bonne prédiction, la sensibilité globale est donc : $\mathrm{OA}_1 = \frac{1}{3}$.

   Ensuite on considère les sons 4, 5 et 6 comme base d'apprentissage et les autres sons comme base de test.

$$\widehat{x}_1 = 2,\ \widehat{x}_2 = 2,\ \widehat{x}_3 = 2$$

   Il y a ici une bonne prédiction, la sensibilité globale est donc : $\mathrm{OA}_2 = \frac{1}{3}$.

   La sensibilité globale finale est donc égale à : $\mathrm{OA} = \frac{1}{2}(\mathrm{OA}_1 + \mathrm{OA}_2) = \frac{1}{3}$.

Exercice 5 (101) *Un collègue informaticien ayant de vague souvenir de traitement du signal vous demande comment enlever le silence qui se trouve au début et à la fin d'un signal sonore. Ce signal sonore est échantillonné à 40kHz, il contient juste un mot d'une durée de 3 secondes et il est stocké sous la forme d'un fichier texte avec à chaque ligne un entier entre −128 et 127. Expliquez-lui en détail et de façon qu'il comprenne en une quinzaine de lignes, comment il peut fabriquer un nouveau fichier texte en utilisant C correspondant à ce signal sonore sans les parties silencieuses.*

Solution : On découpe le signal en 100 trames de 30ms sans chevauchement. Chaque trame contient $N_K = 3 * 40000/100 = 1200$ échantillons notés x_{kn} avec $k \in \{0 \dots 99\}$. On note E_k l'énergie contenue dans chaque trame.

$$E_k = \sum_{n=0}^{N_K-1} (x_{kn})^2 \tag{3}$$

On définit le seuil

$$s = \frac{1}{10} \max_k E_k \tag{4}$$

On calcule `k_1` et `k_2` les numéros de la première trame après le silence et de la dernière trame avant le dernier silence au moyen de ce pseudo-code.

```
k = 0
WHILE (E\_k <= s)
  k += 1
END_WHILE
k\_1 = k
k = K-1
WHILE (E\_k <= s)
  k -= 1
END_WHILE
k\_2 = k
```

On reconstruit le signal sonore en considérant toutes les trames entre k_1 et k_2.

$$y_n = [x_{k\_10} \dots x_{k\_1N_K} \quad \dots \quad x_{k\_20} \dots x_{k\_2N_K}] \tag{5}$$

Exercice 6 (x131) *On cherche à écrire les équations d'un banc de filtre avec une échelle logarithmique et composé de trois filtres. On suppose ici qu'on dispose d'un algorithme qui calcule la réponse fréquentielle d'un filtre passe-bas en fonction de $f_r = \frac{f_c}{f_e}$ où f_c est la fréquence de coupure. Ce filtre est à réponse impulsionnelle finie et compte N termes successifs notés $h_n(f_r)$. Ce banc de filtre s'applique aux signaux de chaque trame, ceux-ci sont ici composés de $N_K = 1200$ échantillons. La fréquence d'échantillonnage est de $f_e = 40kHz$.*

1. *Proposez des valeurs pour les fréquences de coupures des trois filtres.*

2. *Déduisez les réponses impulsionnelles des trois filtres.*

3. *Donnez les équations des trois filtres.*

Solution :

1. On note a la raison de la suite géométrique qui donne les fréquences de coupures. Le premier filtre est un passe-bas de fréquence de coupure $f_c^{(1)} = af_0$ où $f_0 = \frac{1}{N_K T_e} = \frac{f_e}{N_K}$, parce qu'il n'est pas pertinent de considérer une fréquence plus faible pour un signal d'une durée assez courte. Les deuxième et le troisième filtres sont des passe-bande de fréquences de coupures $[f_c^{(1)}, f_c^{(2)}]$ et $[f_c^{(2)}, f_c^{(3)}]$ où $f_c^{(2)} = af_c^{(1)} = a^2 f_0$ et $f_c^{(3)} = af_c^{(2)} = a^3 f_0$ Par ailleurs la dernière fréquence de coupure $f_c^{(3)}$ doit coïncider avec $\frac{f_e}{2}$. La valeur de a est alors déterminée par

$$\frac{f_e}{2} = a^3 \frac{f_e}{N_K} \quad \Rightarrow \quad a = \left(\frac{N_K}{2}\right)^{1/3} = 600^{1/3} = 8.43 \tag{6}$$

   Cela donne $f_c^{(1)} = \frac{af_e}{N_K} = 281\text{Hz}$, $f_c^{(2)} = 2371\text{Hz}$ et $f_c^{(3)} = 20000\text{Hz}$.

2. Les réponses impulsionnelles des trois filtres sont notées $h_n^{(1)}$, $h_n^{(2)}$ et $h_n^{(3)}$. Le premier est $h_n^{(1)} = h_n\left(\frac{a}{N_K}\right)$. Le deuxième est $h_n^{(2)} = h_n\left(\frac{a^2}{N_K}\right) - h_n\left(\frac{a}{N_K}\right)$. Et le troisième $h_n^{(3)} = h_n\left(\frac{1}{2}\right) - h_n\left(\frac{a^2}{N_K}\right)$.

3. Les relations entrée-sortie des trois filtres sont :

$$\begin{aligned} y_n^{(1)} &= \sum_{k=0}^{N-1} h_k^{(1)} x_{n-k} \\ y_n^{(2)} &= \sum_{k=0}^{N-1} h_k^{(2)} x_{n-k} \\ y_n^{(3)} &= \sum_{k=0}^{N-1} h_k^{(3)} x_{n-k} \end{aligned} \tag{7}$$

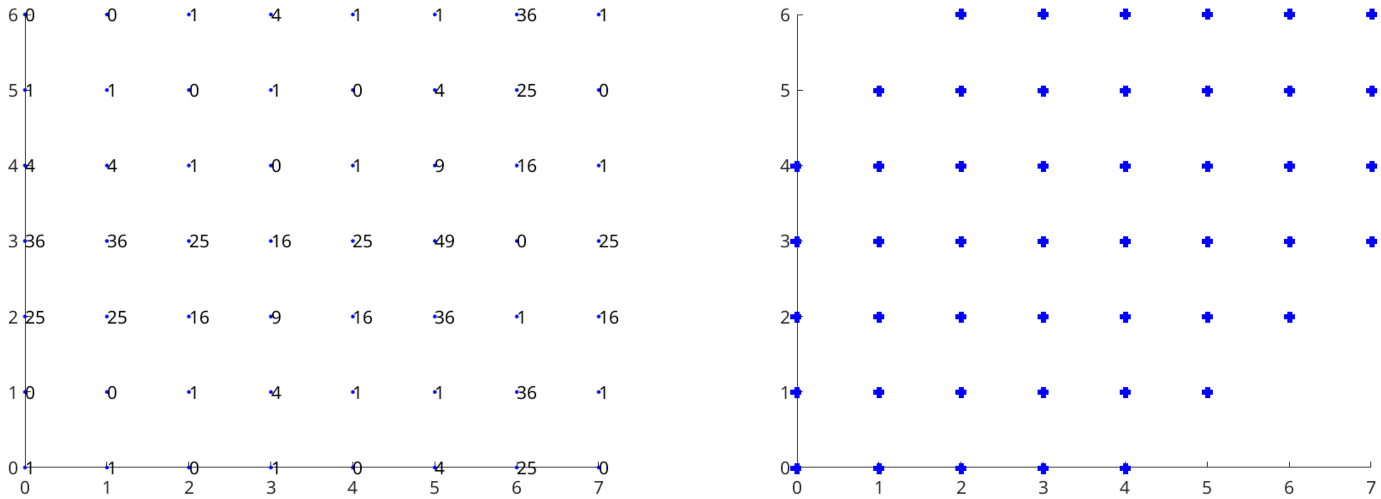

Figure 2: Réponse à gauche à la question 1 de l'exercice 7, et à droite pour la question 2.

Exercice 7 (x132) *On considère deux signaux sonores formés de $K_x = 8$ trames et $K_y = 7$ trames observées avec un seul descripteur. On note x_k et y_k les valeurs de ce descripteur pour chaque signal sonore. On cherche à déterminer $d(x, y)$ la distance entre ces deux sons en prenant en compte cette déformation temporelle. On note $\Phi_x[k]$ et $\Phi_y[k]$ les deux tables d'indexations.*

$$\mathbf{d}(x, y) = \min_{\Phi_x, \Phi_y} \sqrt{\sum_{k=0}^{K-1} \left(x_{\Phi_x[k]} - y_{\Phi_x[k]} \right)^2} \tag{8}$$

1. *Représentez le graphique permettant d'appliquer l'algorithme de Dijkstra pour le calcul de cette distance. L'axe des abscisses porte les valeurs de $\Phi_x[k]$ et l'axe des ordonnées porte celles de $\Phi_y[k]$. Mettez les valeurs sur chaque nœud.*

2. *Représentez avec des plus les nœuds que l'on peut atteindre avec des tables d'indexations qui respectent les contraintes suivantes.*

$$\begin{cases} \Phi_x[k+1] - \Phi_x[k] \in \{0, 1\} \\ \Phi_y[k+1] - \Phi_y[k] \in \{0, 1\} \\ \Phi_x[0] = 0 \\ \Phi_y[0] = 0 \\ \Phi_x[K] = K_x - 1 \\ \Phi_y[K] = K_y - 1 \\ |\Phi_y[k] - \Phi_x[k]| \leq \delta \end{cases} \tag{9}$$

   avec $\delta = 4$.

3. *Représentez l'exploration à la première itération, à la deuxième itération et à la dernière itération. Sur chaque nœud doit figurer à gauche le coût et à droite les coordonnées du nœud d'origine permettant d'atteindre ce nœud avec ce coût.*

4. *Représentez le chemin atteint et son coût. Donnez la valeur de la distance.*

Solution :

1. Gauche de la figure 2.

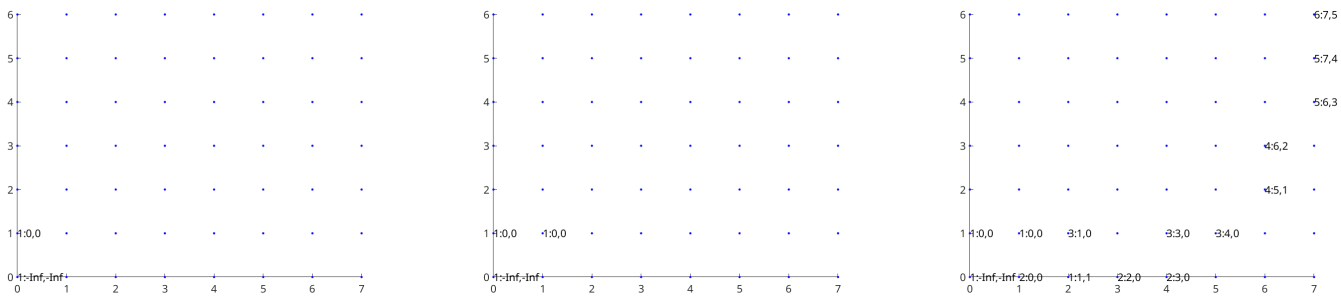

Figure 3: Réponses à la question 3 de l'exercice 7.

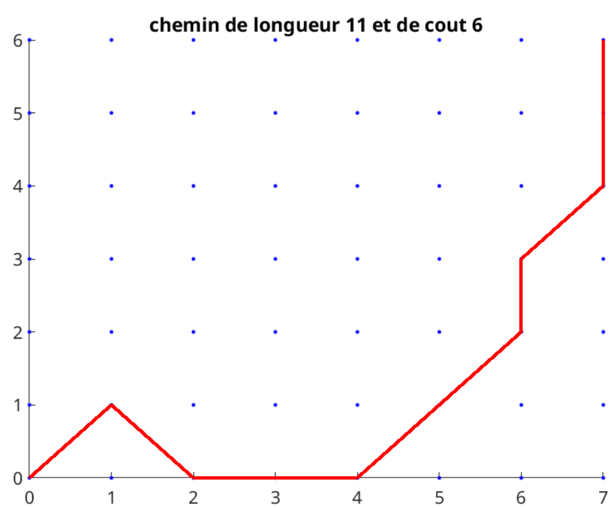

Figure 4: Réponse à la question 4 de l'exercice 7.

```
clear;
x=[1,1,2,3,2,0,7,2]; Kx=length(x);
y=[2,1,6,7,3,2,1]; Ky=length(y);
figure(1); hold on;
for kx=0:Kx-1
  for ky=0:Ky-1
    plot(kx,ky,'b.','linewidth',3);
    tab(kx+1,ky+1)=(x(kx+1)-y(ky+1))^2;
  end
end
text(kx,ky,num2str(tab(kx+1,ky+1)));
hold off;
saveas(1,'fig_ex120.png');
```

2. Droite de la figure 2.

```
tab_ok=zeros(size(tab));
delta=4;
figure(2); hold on;
for kx=0:Kx-1
  for ky=0:Ky-1
    if ~(abs(ky-kx)<=delta) continue, end
    tab_ok(kx+1,ky+1)=1;
  end
plot(kx,ky,'b+','linewidth',3);
end
hold off;
saveas(2,'fig_ex120b.png');
```

3. Figure 3. Dans la phase exploratoire, on commence par un ensemble ne contenant que le point de départ et on prend la valeur minimale parmi ses voisins.

```
function [tab_S,tab_orx,tab_ory]=exploration(tab,tab_ok,tab_S,tab_orx,tab_ory)
  Kx=size(tab,1); Ky=size(tab,2);
  if 2==nargin
    tab_S=+Inf*ones(size(tab));
    tab_orx=-Inf*ones(size(tab)); tab_ory=-Inf*ones(size(tab));
  end
tab_S(1,1)=tab(1,1);
  %recherche_voisins
  voisins = [];
  for kx=0:Kx-1
    for ky=0:Ky-1
      if ~tab_ok(kx+1,ky+1)
        continue, end
      if ~(tab_S(kx+1,ky+1)==Inf) continue, end
      if (kx>0)&(ky>0)&~(tab_S(kx,ky)==Inf)
        voisins = [voisins,[kx;ky;tab(kx+1,ky+1);kx-1;ky-1]]; end
      if (ky>0)&~(tab_S(kx+1,ky)==Inf)
        voisins = [voisins,[kx;ky;tab(kx+1,ky+1);kx  ;ky-1]]; end
      if (kx<Kx-1)&(ky>0)&~(tab_S(kx+2,ky)==Inf)
        voisins = [voisins,[kx;ky;tab(kx+1,ky+1);kx+1;ky-1]]; end
      if (kx>0)&~(tab_S(kx,ky+1)==Inf)
        voisins = [voisins,[kx;ky;tab(kx+1,ky+1);kx-1;ky]]; end
      if (kx<Kx-1)&~(tab_S(kx+2,ky+1)==Inf)
```

```
        voisins = [voisins,[kx;ky;tab(kx+1,ky+1);kx+1;ky]]; end
      if (kx>0)&(ky<Ky-1)&~(tab_S(kx,ky+2)==Inf)
        voisins = [voisins,[kx;ky;tab(kx+1,ky+1);kx-1;ky+1]]; end
      if (ky<Ky-1)&~(tab_S(kx+1,ky+2)==Inf)
        voisins = [voisins,[kx;ky;tab(kx+1,ky+1);kx  ;ky+1]]; end
      if (kx<Kx-1)&(ky<Ky-1)&~(tab_S(kx+2,ky+2)==Inf)
      voisins = [voisins,[kx;ky;tab(kx+1,ky+1);kx+1;ky+1]]; end
end
  end
  %choix d'un voisin
  [val,ind]=min(voisins(3,:));
  voisin=voisins(:,ind);
  %actualisation des informations
  if tab_S(voisin(4)+1,voisin(5)+1)+val < tab_S(voisin(1)+1,voisin(2)+1)
    tab_S(voisin(1)+1,voisin(2)+1)=tab_S(voisin(4)+1,voisin(5)+1)+val;
    tab_orx(voisin(1)+1,voisin(2)+1)=voisin(4);
tab_ory(voisin(1)+1,voisin(2)+1)=voisin(5);
  end
end
%affichage
function affiche(tab_S,tab_ok,tab_orx,tab_ory,num)
  figure(num); hold on;
  Kx=size(tab_S,1); Ky=size(tab_S,2);
  for kx=0:Kx-1
    for ky=0:Ky-1
      plot(kx,ky,'b.','linewidth',2);
      if tab_S(kx+1,ky+1)<Inf
        text(kx,ky,[num2str(tab_S(kx+1,ky+1)),':',...
  num2str(tab_orx(kx+1,ky+1)),',',num2str(tab_ory(kx+1,ky+1))]);
  end
end
end
  saveas(num,['fig_ex120c_',num2str(num),'.png']);
end
%vérification
function verif(tab,tab_ok,tab_S,tab_orx,tab_ory)
  for exp = 1:100
    Kx=size(tab,1); Ky=size(tab,2);
    kx=floor(rand(1)*Kx);
    ky=floor(rand(1)*Ky);
    if ~tab_ok(kx+1,ky+1)
      assert(tab_S(kx+1,ky+1)==Inf);
    end
    if ~(tab_orx == -Inf)
      assert(tab_ory == -Inf);
      kx1 = tab_orx(kx+1,ky+1);
      ky1 = tab_ory(kx+1,ky+1);
      assert(tab_S(kx1+1,ky1+1)+tab(kx+1,ky+1) == tab_S(kx+1,ky+1));
    end
  end
end
%programme
close all
num=1;
```

```
while(1)
  if 1==num
    [tab_S,tab_orx,tab_ory]=exploration(tab,tab_ok);
  else
    [tab_S,tab_orx,tab_ory]=exploration(tab,tab_ok,tab_S,tab_orx,tab_ory);
  end
  verif(tab,tab_ok,tab_S,tab_orx,tab_ory);
  affiche(tab_S,tab_ok,tab_orx,tab_ory,num);
  if tab_S(Kx,Ky)<Inf break; end
  num = num+1;
end
```

4. Le coût du chemin est de 6, donc la distance est $\sqrt{6}$ Figure 4.

```
%%affichage
function affiche2(chemin,cout,tab_S,num)
  figure(num); hold on;
  Kx=size(tab_S,1); Ky=size(tab_S,2);
  for kx=0:Kx-1
    for ky=0:Ky-1
      plot(kx,ky,'b.','linewidth',2);
end
  end
  title(['chemin de longueur ',num2str(size(chemin,2)),...
  ' et de cout ',num2str(cout)]),
  K=size(chemin,2);
  for k=1:K-1
    plot([chemin(1,k),chemin(1,k+1)],[chemin(2,k),chemin(2,k+1)],...
  'r-','linewidth',2);
  end
saveas(num,['fig_ex120d_chemin',num2str(num),'.png']);
end
%%programme
kx=Kx-1; ky=Ky-1;
chemin=[]; cout=0;
while(1)
  chemin = [[kx;ky],chemin];
  cout = cout+tab(kx+1,ky+1);
  if (0 == kx)&(0 == ky) break; end
  verif2(chemin,cout,tab,tab_ok,tab_S,tab_orx,tab_ory);
  kx1 = tab_orx(kx+1,ky+1);
  ky1 = tab_ory(kx+1,ky+1);
  kx = kx1; ky = ky1;
end
affiche2(chemin,cout,tab_S,num+1);
```

Exercice 8 (x133) *On considère deux sons x_n et y_n et deux descripteurs, PMCT et ZCR. On suppose que préalablement les deux sons ont été préparés, c'est-à-dire que les silences en début et en fin ont été retirés et qu'ensuite ils ont été normalisés, de sorte que la puissance des deux signaux est identique. On note $PMCT_k^{(x)}$, $PMCT_k^{(y)}$, $ZCR_k^{(x)}$, $ZCR_k^{(y)}$, les valeurs des descripteurs pour la trame k et pour le son x_n ou y_n. f_e est la fréquence d'échantillonnage et K_x et K_y sont les nombres de trames de x_n et y_n. La distance proposée est ainsi définie*

$$\mathbf{d}(x_n, y_n) = \sqrt{\left(\frac{\sigma_x - \sigma_y}{\sigma_x + \sigma_y}\right)^2 + \left(\frac{\eta_x - \eta_y}{f_e}\right)^2} \tag{10}$$

avec les notations suivantes :

$$\begin{aligned}
\mu_x &= \tfrac{1}{K_x}\textstyle\sum_{k=0}^{K_x-1} PMCT_k^{(x)}\\
\sigma_x &= \sqrt{\tfrac{1}{K_x}\textstyle\sum_{k=0}^{K_x-1}\left(PMCT_k^{(x)} - \mu_x\right)^2}\\
\mu_y &= \tfrac{1}{K_y}\textstyle\sum_{k=0}^{K_y-1} PMCT_k^{(y)}\\
\sigma_y &= \sqrt{\tfrac{1}{K_y}\textstyle\sum_{k=0}^{K_y-1}\left(PMCT_k^{(y)} - \mu_y\right)^2}\\
\eta_x &= \tfrac{1}{K_x}\textstyle\sum_{k=0}^{K_x-1} ZCR_k^{(x)}\\
\eta_y &= \tfrac{1}{K_y}\textstyle\sum_{k=0}^{K_y-1} ZCR_k^{(y)}
\end{aligned} \tag{11}$$

Pour chaque affirmation, dites si elle est vraie ou fausse. Dans le cas où elle est vraie, donnez une démonstration et dans le cas où elle est fausse, donnez un contre-exemple. Pour la dernière affirmation, l'idée est de s'appuyer sur l'affirmation suivante.

$$x \geq 0 \text{ et } y \geq 0 \quad \Rightarrow \quad \left|\frac{1-xy}{1+xy}\right| \leq \left|\frac{1-x}{1+x}\right| + \left|\frac{1-y}{1+y}\right| \tag{12}$$

Sa démonstration n'est pas demandée, elle est faite plus bas.

1. $\mathbf{d}(x_n, y_n) = \mathbf{d}(2x_n, 2y_n)$.
2. $\mathbf{d}(x_n, y_n) = \mathbf{d}(y_n, x_n)$.
3. $\mathbf{d}(x_n, y_n) = 0 \quad \Rightarrow \quad x_n = y_n$.
4. $\mathbf{d}(x_n, y_n) \leq \frac{\sqrt{5}}{2}$.
5. $\mathbf{d}(x_n, z_n) \leq \mathbf{d}(x_n, y_n) + \mathbf{d}(y_n, z_n)$

Solution :

1. L'affirmation est vraie. Je note x'_n et y'_n les signaux obtenus lorsque x_n et y_n sont multipliés par deux. D'après le cours $\text{PMCT}_k^{(x)} = \frac{1}{N_k}\sum_{n=0}^{N_K-1} x_k^2[n]$. D'autre part $f_k^{(ZCR)} = \frac{f_e}{2}\left(\frac{1}{N_{K-1}}\sum_{n=0}^{N_K-2} \mathbf{1}(x_k[n+1]x_k[n] < 0)\right)$, que l'on note ici $\text{ZCR}_k^{(x)}$. On a donc

$$\begin{aligned}
&\text{PMCT}_k^{(x')} = 4\text{PMCT}_k^{(x)} \text{ et } \text{PMCT}_k^{(y')} = 4\text{PMCT}_k^{(y)}\\
&\text{ZCR}_k^{(x')} = \text{ZCR}_k^{(x)} \text{ et } \text{ZCR}_k^{(y')} = 4\text{ZCR}_k^{(y)}
\end{aligned} \tag{13}$$

Du coup, on a

$$\left\{\begin{array}{lcl}
\mu_{x'} = 4\mu_x & & \mu_{y'} = 4\mu_y\\
\sigma_{x'} = 4\sigma_x & \text{et} & \sigma_{y'} = 4\sigma_y\\
\eta_{x'} = \eta_x & & \eta_{y'} = \eta_y
\end{array}\right. \quad \Rightarrow \quad \left\{\begin{array}{l}
\frac{\sigma_{x'}-\sigma_{y'}}{\sigma_{x'}+\sigma_{y'}} = \frac{\sigma_x-\sigma_y}{\sigma_x+\sigma_y}\\
\frac{\eta_{x'}-\sigma_{y'}}{f_e} = \frac{\eta_x-\eta_y}{f_e}
\end{array}\right. \tag{14}$$

Et finalement on a bien $\mathbf{d}(x_n, y_n) = \mathbf{d}(2x_n, 2y_n)$

2. L'affirmation est vraie. On observe que

$$\left(\frac{\sigma_x - \sigma_y}{\sigma_x + \sigma_y}\right)^2 = \left(\frac{\sigma_y - \sigma_x}{\sigma_y + \sigma_x}\right)^2 \tag{15}$$

et que

$$\left(\frac{\eta_x - \eta_y}{f_e}\right)^2 = \left(\frac{\eta_y - \eta_x}{f_e}\right)^2 \tag{16}$$

On en déduit que $\mathbf{d}(x_n, y_n) = \mathbf{d}(y_n, x_n)$.

3. L'affirmation est fausse. Je propose deux signaux découpé en une seule trame de longueur $N_K = 2$.

$$\begin{aligned} x_n &= 2\delta_n + 1\delta_{n-1} \\ y_n &= 1\delta_n + 2\delta_{n-1} \end{aligned} \tag{17}$$

Les valeurs des descripteurs sont alors

$$\text{PMCT}_0^{(x)} = \text{PMCT}_0^{(y)} = \frac{5}{2} \text{ et } \text{ZCR}_0^{(x)} = \text{ZCR}_0^{(y)} = 0 \tag{18}$$

On a alors $\sigma_x = \sigma_y = 0$, $\mu_x = \mu_y = 5$ et $\eta_x = \eta_y = 0$ et par suite $\mathbf{d}(x_n, y_n) = 0$ et pourtant $x_n \neq y_n$.

4. On a une première majoration en observant la décroissance de $x \mapsto \frac{1}{1+x}$ pour $x \in \Re$.

$$\frac{\sigma_x - \sigma_y}{\sigma_x + \sigma_y} = \frac{1 - \frac{\sigma_y}{\sigma_x}}{1 + \frac{\sigma_y}{\sigma_x}} = \frac{-1 - \frac{\sigma_y}{\sigma_x} + 2}{1 + \frac{\sigma_y}{\sigma_x}} = -1 + \frac{2}{1 + \frac{\sigma_y}{\sigma_x}} \in -1 + [0, 2] = [-1, 1] \tag{19}$$

On a une deuxième majoration en utilisant la définition de ZCR_k

$$0 \leq \text{ZCR}_k = \frac{f_e}{2}\left(\frac{1}{N_K - 1}\sum_{n=0}^{N_K-2} \mathbf{1}(x_k[n+1]x_k[n] < 0)\right) \leq \frac{f_e}{2} \tag{20}$$

On a donc $\eta_x \in [0, \frac{f_e}{2}]$, $\eta_y \in [0, \frac{f_e}{2}]$ et par suite

$$-\frac{1}{2} \leq \left(\frac{\eta_x - \eta_y}{f_e}\right) \leq \frac{1}{2} \tag{21}$$

Finalement on trouve

$$\mathbf{d}(x_n, y_n) \leq \sqrt{1^2 + \left(\frac{1}{2}\right)^2} = \sqrt{\frac{5}{4}} \leq \frac{\sqrt{5}}{2} \tag{22}$$

5. Je définit $h(x, y)$ sur $\mathbb{R}_+^2$ en utilisant $f(x) = \left|\frac{1-x}{1+x}\right|$ défini dans la preuve du lemme 1.

$$h(x, y) = \left|\frac{x - y}{x + y}\right| = \begin{cases} f\left(\frac{y}{x}\right) & \text{si} \quad x \neq 0 \\ 1 & \text{si} \quad x = 0 \end{cases} \tag{23}$$

Cette définition permet d'écrire d sous la forme

$$\mathbf{d}(x_n, y_n) = \sqrt{h^2(\sigma_x, \sigma_y) + \left(\frac{\eta_x - \eta_y}{f_e}\right)^2} \tag{24}$$

Le lemme 1 montre que

$$h(x, z) \leq h(x, y) + h(y, z) \tag{25}$$

- Si $x \neq 0$ et $y \neq 0$, $h(x, z) = f\left(\frac{z}{x}\right) = f\left(\frac{z}{y}\frac{y}{x}\right) \leq f\left(\frac{z}{y}\right) + f\left(\frac{y}{x}\right) = h(x, y) + h(y, z)$.

- Si $x \neq 0$ et $y = 0$, $h(x,z) = f\left(\frac{z}{x}\right) \leq 2 = f\left(\frac{y}{x}\right) + 1 = h(x,y) + h(y,z)$.
- Si $x = 0$ et $y \neq 0$, $h(x,z) = 1 = h(x,y) \leq h(x,y) + h(y,z)$.
- Si $x = y = 0$, $h(x,z) = 1 \leq 2 = h(x,y) + h(y,z)$.

La norme euclidiennes sur $\mathbb{R}^2$ respecte l'inégalité triangulaire aussi

$$\sqrt{(a_1+a_2)^2 + (b_1+b_2)^2} \leq \sqrt{a_1^2 + b_1^2} + \sqrt{a_2^2 + b_2^2} \tag{26}$$

Je considère trois signaux, x_n, y_n, z_n et je note μ_x, μ_y, μ_z, $\sigma_x, \sigma_y, \sigma_z$ et η_x, η_y, η_z les descripteurs correspondants.

$$\begin{aligned}\mathbf{d}(x_n, z_n) = \sqrt{h^2(\sigma_x, \sigma_z) + \left(\frac{\eta_x - \eta_z}{f_e}\right)^2} \leq \sqrt{(h(\sigma_x, \sigma_y) + h(\sigma_y, \sigma_z))^2 + \left(\frac{\eta_x - \eta_y}{f_e} + \frac{\eta_y - \eta_z}{f_e}\right)^2} \\ \leq \sqrt{h^2(\sigma_x, \sigma_y) + \left(\frac{\eta_x - \eta_y}{f_e}\right)^2} + \sqrt{h^2(\sigma_y, \sigma_z) + \left(\frac{\eta_y - \eta_z}{f_e}\right)^2} = \mathbf{d}(x_n, y_n) + \mathbf{d}(y_n, z_n)\end{aligned} \tag{27}$$

Lemma 1 *Pour $x, y \in \mathbb{R}$,*

$$\left|\frac{1-xy}{1+xy}\right| \leq \left|\frac{1-x}{1+x}\right| + \left|\frac{1-y}{1+y}\right| \tag{28}$$

Proof. Je note f la fonction

$$\begin{aligned} f(x) = \left|\frac{1-x}{1+x}\right| = \text{signe}(x-1)\frac{x-1}{1+x} = \text{signe}(x-1)\left(1 - \frac{2}{1+x}\right) \\ f(x) = f\left(\frac{1}{x}\right) \text{ pour } x \neq 0 \\ \frac{d}{dx} f(x) = \frac{2\text{signe}(x-1)}{(1+x)^2} \text{ pour } x \neq 1 \end{aligned} \tag{29}$$

Cette fonction est donc décroissante sur $[0,1]$ et croissante sur $[1, +\infty[$.

Je note g la fonction

$$\begin{aligned} g(x,y) = f(x) + f(y) - f(xy) \\ \frac{\partial}{\partial y} g(x,y) = f'(y) - x f'(xy) = \frac{2\text{signe}(y-1)}{(1+y)^2} - \frac{2x\text{signe}(xy-1)}{(1+xy)^2} \text{ pour } x \neq 1,\ y \neq 1,\ \text{et } xy \neq 1 \end{aligned} \tag{30}$$

Le lemme est équivalent à montrer que

$$\forall x \in \mathbb{R}_+,\ \inf_{y \in \mathbb{R}} g(x,y) \geq 0 \tag{31}$$

Pour $x = 0$, $\inf_{y\in\mathbb{R}} g(x,y) = g(x,1) = f(x) - f(x) - f(1) = 0$.

Supposons que $y \in \mathbb{R}_+ \backslash \{0,1\}$ soit une valeur minimale alors

$$0 = \frac{1}{(1+y)^2} - \frac{x}{(1+xy)^2} = \frac{1 + 2xy + x^2y^2 - x\left(1 + 2y + y^2\right)}{(1+y)^2(1+xy)^2} = \frac{(1-xy^2)(1-x)}{(1+y)^2(1+xy)^2} \tag{32}$$

Aussi une nouvelle valeur potentiellement minimale est $y = \frac{1}{\sqrt{x}}$

$$\begin{aligned} \inf_{y\in\mathbb{R}_+} g(x,y) = \min\left(g(x,0), g(x,1), g\left(x, \frac{1}{\sqrt{x}}\right), g\left(x, \frac{1}{x}\right), \lim_{y\to+\infty} g(x,y)\right) \\ = \min\left(f(x), 0, f(x), 2f(x), f(x)\right) = 0 \end{aligned} \tag{33}$$