Concours ES - Epreuve de Mathématiques Générales
Durée 3 heures

Portables, calculatrices et documents interdits
Les exercices sont indépendants et peuvent être traités séparément.
Un soin particulier devra être apporté à la justification des réponses et il sera apprécié dans la notation.

Exercice 1 Soit $(u_n)_{n\geq 0}$ la suite de nombres réels définie par la donnée de u_0 et de u_1 et la relation de récurrence :

$$2u_{n+2} - u_{n+1} - u_n = 0.$$

On pose pour tout $n \in \mathbb{N}$

$$v_n = u_{n+1} - u_n \text{ et } w_n = 2u_{n+1} + u_n.$$

1. Montrer que $(v_n)_{n\in\mathbb{N}}$ est une suite géométrique de raison $-\frac{1}{2}$. Exprimer v_n en fonction de n, u_0 et u_1.
2. Montrer que $(w_n)_{n\in\mathbb{N}}$ est une suite constante. Exprimer w_n en fonction de u_0 et u_1.
3. En calculant $-2v_n + w_n$ de deux façons différentes, exprimer u_n en fonction de n, u_0 et u_1.
4. On pose pour tout $n \in \mathbb{N}$

$$S_n = \sum_{k=0}^{n} u_k.$$

   (a) Calculer S_1 et S_2 en fonction de u_0 et u_1.
   (b) Calculer S_n en fonction de n, u_0 et u_1.
   (c) Pour quelles valeurs de u_0 et u_1, la suite $(S_n)_{n\in\mathbb{N}}$ admet-elle une limite finie et dans ce cas exprimer cette limite en fonction de u_0.

Exercice 2 Soit $I_n = \int_0^1 \frac{x^n}{1+x}dx$, où $n \in \mathbb{N}$.

1. Calculer I_0, I_1 et I_2.
2. En majorant la fonction intégrée, montrer que $lim_{n\to\infty} I_n = 0$.
3. Calculer $I_n + I_{n+1}$ pour $n \in \mathbb{N}$.
4. Soit $S_n = \sum_{k=1}^{n} \frac{(-1)^{k+1}}{k}$.
   (a) Exprimer S_2, S_3, S_4 en fonction de I_j, $j \in \mathbb{N}$.
   (b) Exprimer S_n en fonction de I_j, $j \in \mathbb{N}$.
   (c) Déterminer $lim_{n\to\infty} S_n$.

Exercice 3 Considérons $P \neq 0$ un polynôme de degré $n \in \mathbb{N}$ à coefficients réels vérifiant l'équation :

$$(X-1)P'(X) + XP(X) = 1 + \frac{X^3}{2},$$

où P' est la dérivée de P.

1. Justifier que P est un polynôme de degré $n = 2$.
2. Trouver P.

Exercice 4 Soit A une matrice $\mathcal{M}_{2\times 2}$ définie par $A = \begin{pmatrix} 1 & 0 \\ 2 & 1 \end{pmatrix}$.

1. Calculer A^2 et A^3.
2. Exprimer A^n en fonction de n pour tout $n \in \mathbb{N}^*$.
3. Justifier que A est inversible et calculer A^{-1}.
4. Exprimer A^{-n} en fonction de n pour tout $n \in \mathbb{N}^*$.

Exercice 5 Soit
$$u = 1 + i \text{ et } v = -1 + i\sqrt{3}.$$

1. Rappeler la définition du module et de l'argument d'un nombre complexe $a + bi$ avec $a, b \in \mathbb{R}^*$.
2. Déterminer les modules de u et v.
3. Déterminer un argument de u et un argument de v.
4. En déduire le module et un argument pour chacune des racines cubiques de u.
5. Déterminer le module et un argument de $\frac{u}{v}$.
6. En déduire les valeurs de $\cos\left(-\frac{5\pi}{12}\right)$ et $\sin\left(-\frac{5\pi}{12}\right)$.

Exercice 6

1. Enoncer le théorème de Taylor-Lagrange (on notera $n+1$ l'ordre du reste dans la formule).
2. Soit la fonction $f : \mathbb{R}_+ \longrightarrow \mathbb{R}$ définie par $f(t) = \ln(1 + t)$.
   (a) Calculer les dérivées successives de f jusqu'à l'ordre 4.
   (b) Appliquer le théorème de Taylor-Lagrange sur l'intervalle $[0, 1]$ et avec un reste d'ordre 4.
   (c) En déduire l'encadrement de $\ln(2)$ suivant
   $$\frac{7}{12} \leq \ln(2) \leq \frac{157}{192}.$$

Concours ES - Epreuve de Probabilités
Durée 3 heures
Portables, calculatrices et documents interdits
Les exercices sont indépendants et peuvent être traités séparément.
Un soin particulier devra être apporté à la justification des réponses et il sera apprécié dans la notation.

Exercice 1 Une variable aléatoire discrète suit une loi de Poisson de paramètre $\lambda > 0$ si

$$\mathbb{P}(X = k) = \frac{\lambda^k}{k!} e^{-\lambda},$$

k étant un entier naturel, $k = 0, 1, 2. \ldots$

1. Calculer (ou rappeler directement le résultat) $\sum_{k=0}^{\infty} \frac{\lambda^k}{k!}$.
2. Calculer l'espérance et la variance de X.
3. Calculer $\mathbb{P}(X < 3)$ et $\mathbb{P}(X \geq 2)$.
4. Soient X_1 et X_2 deux variables aléatoires indépendantes de loi de Poisson de paramètre $\lambda = 1$. Calculer $\mathbb{P}(X_1 \leq 1)$ et $\mathbb{P}(X_1 + X_2 \leq 2)$.
5. Ecrire $e^{-n} \sum_{k=0}^{n} \frac{n^k}{k!}$ comme la probabilité d'une somme de variables aléatoires indépendantes de loi Poisson.
6. Rappeler le TCL (Théorème Central Limite).
7. Calculer $\lim_{n\to\infty} e^{-n} \sum_{k=0}^{n} \frac{n^k}{k!}$.

Exercice 2 Soit Γ une matrice $\mathcal{M}_{3\times 3}$ définie par $\Gamma = \begin{pmatrix} 1 & 0 & -1 \\ 0 & 2 & 2 \\ -1 & 2 & 5 \end{pmatrix}$ avec $a \in \mathbb{R}_+$.

1. Rappeler la définition d'une matrice définie positive.
2. Montrer que la matrice Γ est définie positive. En déduire qu'il existe $X = (X_1,\ X_2,\ X_3)^t$ un vecteur gaussien de moyenne $m = (0, 1, 1)^t$ et de matrice de covariance Γ (où x^t est la transposée du vecteur x).
3. Préciser les relations d'indépendance entre les coordonnées de X.
4. On pose

$$Y_1 = X_1 + X_3, \qquad Y_2 = \alpha X_1 + 2X_2 + \beta X_3 \quad \text{et} \quad Y_3 = -X_1 - X_2 + X_3,$$

où $\alpha, \beta \in \mathbb{R}$.

(a) Montrer que $Y = (Y_1,\ Y_2,\ Y_3)^t$ est un vecteur gaussien dont on donnera les paramètres (vecteur moyenne et matrice de covariance en fonction de α, β).

(b) Les variables aléatoires $Y_1,\ Y_2$, et Y_3 sont-elles mutuellement indépendantes ?

(c) Pour quelles valeurs de (α, β), les variables Y_2 et Y_3 sont-elles indépendantes ?

Exercice 3 Soit X une variable aléatoire à valeurs réelles positives. La variable X est absolument continue et a sa densité définie sur $[0, \infty[$ par

$$f(x) = \lambda e^{-\lambda x}.$$

La loi de X est appelée loi exponentielle de paramètre λ ($\lambda > 0$).

1. Calculer $\mathbb{P}(a \leq X \leq b)$ en fonction de a et b, où $a, b \in \mathbb{R}$ tels que $a \leq b$.
2. Déterminer la valeur de λ telle que la probabilité $\mathbb{P}(1 \leq X \leq 2)$ soit égale à $\frac{1}{4}$.

Exercice 4 Soit X est une variable aléatoire réelle.

1. Rappeler la formule de la variance de X.
2. On sait que si $\phi(.)$ est une fonction affine,

$$\mathbb{E}(\phi(X)) = \phi(\mathbb{E}(X)).$$

   A quelle condition sur la loi de probabilité de X cette égalité reste-t-elle vraie si ϕ est la fonction définie par $\phi(x) = x^2$?

Exercice 5 On lance simultanément deux pièces de monnaie (équilibrées) et on note

(X, Y)= (nombre de « piles » obtenues, nombre de « faces » obtenues).

1. Déterminer la loi de probabilité du vecteur (X, Y).
2. Déterminer la loi de probabilité de chacune des variables aléatoires réelles

$$X, \quad X + Y, \quad X - Y, \quad \frac{X - Y + 2}{2}.$$

Concours d'entrée en Master Econométrie et Statistiques
Epreuve de Micro-économie
Durée 3 heures

Tout matériel électronique (téléphone, calculette, montre connectée, ….) est interdit.
Il sera tenu compte de la rédaction et de la calligraphie

Exercice 1

On considère un duopole produisant un bien homogène. La firme 1 produit une quantité du bien à partir d'une unité de main d'œuvre et d'une unité de matériel. La firme 2 produit le bien à partir de deux unités de main d'œuvre et d'une unité de matériel. Le coût d'une unité de main d'œuvre est w et le coût d'une unité de matériel est r. La demande inverse s'écrit $p = 1 - q_1 - q_2$ où q_1 et q_2 représentent les quantités des firmes 1 et 2. Les deux firmes se font concurrence en quantités (à la Cournot).

1. Calculez l'équilibre de concurrence à la Cournot. Quelles sont les quantités à l'équilibre des firmes 1 et 2 ?
2. Montrez que le profit de la firme 1 ne dépend pas du prix de la main d'œuvre w. Comment interprétez-vous ce résultat ?

Exercice 2

Un agent Uqui se comporte selon la Théorie de l'Utilité Espérée possède une richesse initiale $x_0$et a des préférences représentées par la fonction d'utilité $u(x) = \sqrt{x}$, avec x la richesse définie sur $\mathbb{R}^+$. On lui propose le jeu suivant: avec une probabilité de 1/3 ; il peut tripler sa richesse initiale et avec une probabilité de2/3 il peut tout perdre.

1. Quelle est l'attitude vis-à-vis du risque de l'agent U? U choisit-il de participer à ce jeu ?
2. Calculer l'équivalent certain de ce jeu et en déduire la prime de risque.

Exercice 3

On considère une économie sans production à deux biens, notés 1 et 2, et deux agents, notés A et B. Les biens 1 et 2 sont disponibles en quantités respectives de 4 et 4 unités. Les préférences des agents sont représentées par les fonctions d'utilité suivantes:

$$U(x_{a1}; x_{a2}) = x_{a1}^2 . x_{a2} \text{ et } U(x_{b1}; x_{b2}) = 2. x_{b1} . x_{b2}^2$$

1. Quelles sont les conditions caractérisant un optimum de Pareto?
2. En déduire l'équation de la courbe des contrats. Représenter cette courbe dans la boîte d'Edgeworth.
3. Calculez l'équilibre général en supposant que les allocations initiales sont

$(x_{a1}; x_{a2}) = (1; 2)$ et $(x_{a1}; x_{a2}) = (3; 2)$. Le premier théorème du bien-être est-il vérifié ?

Questions

1. L'eau correspond à un besoin vital et pourtant le diamant est plus cher : commentez.
2. Pourquoi n'est-il pas neutre en termes de consommation de mettre une taxe de 10% sur tous les biens et d'augmenter votre revenu de 10% ?

Concours d'entrée
Épreuve d'informatique

Durée : 3 heures
Tous dispositifs électroniques, calculatrices et documents interdits

Pseudo-code :
À plusieurs reprises dans ce sujet, il vous est demandé de décrire un traitement informatique à l'aide de pseudo-code, c'est à dire en décrivant les opérations de ce programme, cette description étant basée sur un mélange de langage naturel et de structures de contrôle classiques (instructions conditionnelles, boucles, etc). Des exemples sont donnés pages 1 et 2.

Imputation de valeurs pour des données manquantes

Big Data, *Data Mining*, analyse de données. . .ces termes évoquent l'exploitation d'informations relatives à des individus d'une population. En amont de toute exploitation des données, la qualité de ces données doit être contrôlée. Si pour certains individus une partie des données n'est pas renseignée, certains traitements ne seront pas applicables à l'ensemble des individus.

Retirer les individus pour lesquels les données ne sont pas connues est le moyen le plus trivial pour évacuer ce problème, mais cela peut dégrader les analyses réalisées. Il faut donc essayer de proposer des valeurs là où elles manquent. Une solution simple consiste à utiliser la moyenne ou la médiane (déterminées à partir des individus complets) comme valeur. Une autre approche consiste à identifier, pour un individu, ses k plus proches voisins et à construire à partir des valeurs de ces derniers une valeur vraisemblable pour la donnée manquante de l'individu. Ce sont les deux étapes de la méthode des *k-nn* (pour *nearest neighbours* – voisins les plus proches en anglais). Dans cette première partie, nous allons aborder naïvement l'étape de recherche des plus proches voisins.

Considérons un ensemble de N individus. Le i^{e} individu est associé à un point de $\mathbb{R}^p$ noté x_i. Nous noterons d l'application de $\mathbb{R}^p \times \mathbb{R}^p$ dans $\mathbb{R}^+$ utilisée pour mesurer la distance entre deux points de $\mathbb{R}^p$. Soit $k > 0$ un entier petit relativement à N ($k \ll N$). Enfin, notons y le point de $\mathbb{R}^p$ associé à un nouvel individu dont nous souhaitons déterminer les k plus proches voisins. Précisons pour lever tout ambiguïté que y correspond à la partie des informations connues pour l'individu. La description de la recherche des k plus proches voisins de y est la suivante :

> INITIALISER L'ENSEMBLE K AVEC $\{x_i, \forall i \in [\![1, k]\!]\}$.
> POUR i ALLANT DE $k+1$ À N, FAIRE
>   SI $d(y, x_i) < \max_{x \in K} d(y, x)$
>   ALORS
>   RETIRER UN x DE K TEL QUE $d(y, x) = \max_{x \in K} d(y, x)$
>   AJOUTER x_i À K
> RENVOYER K

Ci-dessus, l'ensemble K est mis à jour au fil des essais avec les x_i et contient après l'examen de tous les x_i les k plus proches de y. Pour l'implémentation de cet algorithme, diverses structures classiques sont envisageables pour l'ensemble K. Considérons l'utilisation de listes à travers le pseudo-code suivant.

```
KNN_1(X,k,y) {
  Pour i allant de 1 à k
  Faire
    D[i]<-d(y,X[i]); K[i]<-X[i]

  M<-PosMax(D)
  Pour i allant de k+1 à N
  Faire
    Si d(y,X[i]) < D[M]
    Alors
      K[M]<-X[i]; D[M]<-d(y,X[i]); M<-PosMax(D)

  Renvoyer K
}
```

Q.1 Proposez un pseudo-code pour définir la fonction `PosMax` qui reçoit en argument une liste `L` et qui retourne la valeur d'un indice correspondant à la plus grande valeur de `L`.

Q.2 Le pseudo-code définissant la fonction `KNN_1` ne correspond pas exactement à la description de la recherche qui est donnée précédemment. Quels éléments ont été introduits et à quelle fin ?

Q.3 Proposez un pseudo-code pour définir la fonction `d` qui reçoit en arguments deux listes `a` et `b` donnant chacune les coordonnées d'un point de $\mathbb{R}^p$ et qui renvoie le carré de la distance euclidienne entre ces deux points.

Q.4 Nous nous intéressons à la complexité, en temps et en mémoire, de la fonction `KNN_1`. Faites une estimation du décompte des différentes opérations suivantes : affectation, comparaison, calcul de distance. De même, quelle utilisation mémoire est occasionnée par un appel de cette fonction.

Q.5 En supposant que le calcul de distance soit l'opération la plus coûteuse, proposez une complexité en temps pour la fonction `KNN_1`.

Dans l'approche précédente, nous travaillons sur des objets de taille k qui sont mis à jour au fil de l'examen des x_i. Une autre approche basée sur l'algorithme *QuickSelect* est possible :

> Assigner à D l'ensemble $\{d(y, x_i), \forall i \in [\![1, N]\!]\}$
> Déterminer v la k^{e} plus petite valeur de D (QuickSelect)
> Pour i allant de 1 à N, Faire
>   Si D[i]$\leq v$
>   Alors placer x_i dans K
> Renvoyer K

Le pseudo-code ci-dessous décrit l'algorithme *QuickSelect* qui permet l'identification de la k^{e} valeur la plus petite d'une serie de valeurs :

```
QuickSelect(X,k) {
  Choisir dans X une valeur v.
  Pour i allant de 1 à N
  Faire
    Si X[i] < v
    Alors X1[Longueur(X1)+1]<-X[i]
    Sinon Si X[i] > v
          Alors X3[Longueur(X3)+1]<-X[i]
          Sinon X2[Longueur(X2)+1]<-X[i]

  Si k<=Longueur(X1)
  Alors
    Renvoyer QuickSelect(X1,k)
  Sinon
    Si k> Longueur(X1)+Longueur(X2)
    Alors
      Renvoyer QuickSelect(X3,k-Longueur(X1)-Longueur(X2))
    Sinon
      Renvoyer v
}
```

Q.6 Explicitez le déroulement de l'algorithme afin d'établir qu'il renvoie bien la valeur attendue.

Q.7 Appliquez étape par étape cet algorithme à la série suivante afin de déterminer le 5^{e} plus petit élément :

15 37 35 39 9 22 2 15 2 15

Q.8 Proposez un pseudo-code pour définir la fonction `KNN_2` qui reçoit en arguments la liste `X`, l'entier k et le point y et qui correspond à la recherche de k plus proches voisins basée sur *QuickSelect*.

Q.9 Nous nous intéressons à la complexité, en temps et en mémoire, de la fonction `KNN_2`. L'algorithme *QuickSelect* est réputé pour sa bonne complexité en temps. Que pensez-vous des besoins en occupation mémoire de la fonction `KNN_2` ?

Méthodes d'imputation basées sur les k plus proches voisins

Les données caractérisant nos N individus correspondent à des variables quantitatives notées X_1 à X_p. Pour ces indidivus, nous disposons également d'une variable additionnelle notée Z.

Considérons un nouvel individu pour lequel nous disposons de ses valeurs pour les X_i mais pas pour Z. Nous noterons y le point de $\mathbb{R}^p$ associé à cet individu. Nous pourrions utiliser les valeurs de Z prises par ses plus proches voisins pour lui associer une valeur vraisemblable.

Considérons dans un premier temps le cas pour lequel Z est une variable quantitative. Le cas qualitatif sera envisagé plus loin.

Q.10 Proposez un pseudo-code pour définir la fonction `ImpMoy` qui prendra en argument une liste d'indices et qui renverra la moyenne de Z pour les individus dont les indices sont donnés en arguments.

Il est possible de pondérer les valeurs des voisins en fonction de leur proximité à y : plus un voisin est proche et plus sa valeur compte dans la moyenne.

Q.11 Etant donné un ensemble d'indices I, proposez un système de pondérations $\{p_i \geq 0, i \in I\}$ reprenant le principe précédent et tel que :

1. $\sum_{i \in I} p_i = 1$
2. $\forall i, j \in I, d(y, x_i) < d(y, x_j) \Rightarrow p_i > p_j$

Prenez garde à gérer correctement les éventuels cas pour lesquels $d(y, x_i) = 0$.

Q.12 Proposez un pseudo-code pour définir la fonction `ImpMoyPond` qui prendra en argument une liste d'indices et qui renverra une moyenne pondérée de Z utilisant la pondération que vous avez proposé à la question 11.

Considérons à présent le cas pour lequel Z est une variable qualitative. Il est possible de mettre en place un mécanisme de vote permettant de définir la modalité de Z qui sera renvoyée. Si le principe est simple (la modalité la plus rencontrée chez les individus sélectionnés est choisie), la gestion de ex-æquo est plus délicate. Plusieurs pistes sont envisagées ci-après.

Q.13 Supposons que pour chaque modalité soit calculé le nombre d'individus (de la sélection des k plus proches voisins) qui la possède et que plusieurs modalités correspondent à l'effectif le plus grand (situation d'ex-æquo pour la première place). Il est possible de prendre en compte un voisin supplémentaire pour essayer de départager les modalités. Faites la critique de cette proposition en considérant d'une part son coût et d'autre part sa capacité à proposer une solution.

Q.14 Revenons sur l'idée de prendre en compte la distance des voisins à y pour pondérer les votes. La somme des poids pour chaque modalité pourrait être utilisée pour choisir la modalité renvoyée. Comme dans le cas précédent, faites la critique de cette proposition.

Vous disposez de la fonction `RAND(min,max)` qui renvoie une valeur tirée pseudo-aléatoirement selon une loi uniforme sur l'intervalle [`min`, `max`].

Q.15 Proposez un pseudo-code pour définir la fonction `ImpVotePond` qui prendra en argument une liste d'indices et qui renverra la modalité obtenue selon un procédé reposant sur le système de pondérations évoqué à la question 11 et la fonction `RAND`.

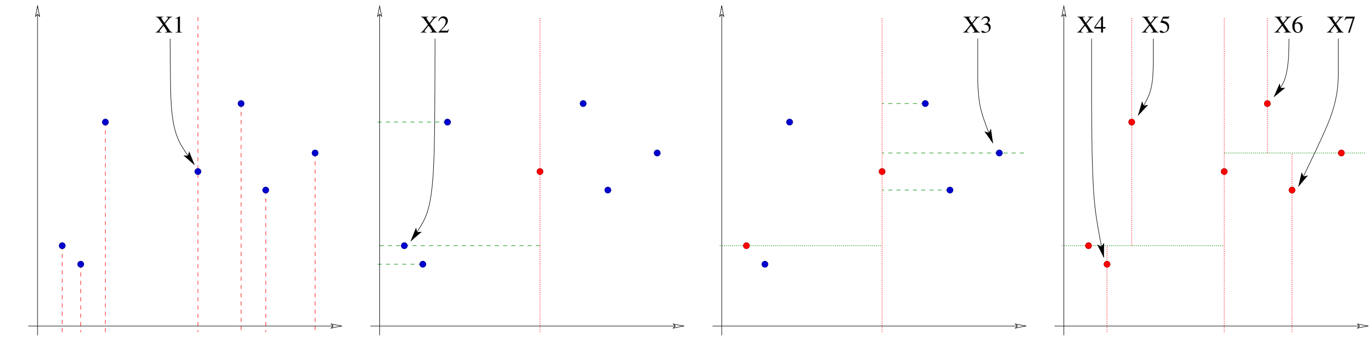
Une structure pour accélerer les recherches

Dans la première partie, les fonctions `KNN_1` et `KNN_2` ont mis l'accent sur l'efficacité de la recherche directement à partir des données, de k et de y. Mais si la recherche des k plus proches voisins doit être répétée pour un grand nombre d'individus, il est possible de tirer parti d'une structure particulière pour accélerer l'identification des k plus proches voisins. Cela implique un surcoût pour la mise à point de cette structure, mais il peut être rapidement amorti si le nombre de recherche est important. Bien que des avancées récentes aient été proposées[1], nous allons nous pencher sur l'ancêtre de ces structures : l'arbre k-d ou k-d *tree*.

Un arbre k-d est une structure construite en réalisant des dichotomies successives des individus en utilisant les valeurs médianes de sous-groupes d'individus le long des variables prises tour à tour. Ainsi, un x_i médian le long de la

1. Consulter par exemple `http://arxiv.org/abs/1509.06957` pour une référence de 2015.

première variable est choisi pour séparer le reste des individus en deux groupes équilibrés. Ce point sera associé à la racine de l'arbre. Chacun des deux groupes est à son tour scindé en choisissant un de ses individus médian le long de la 2ᵉ variable. Ces individus seront associés aux 2ᵉ et 3ᵉ nœuds. Ce processus est poursuivi ainsi de suite en bouclant sur les variables jusqu'à ce que tous les individus soient affectés à un nœud de l'arbre.



(a) Première dichotomie, identification de la racine

(b) Découpage 1ᵉ sous-groupe

(c) Découpage 2ᵉ sous-groupe

(d) Découpage final

FIGURE 1 – Illustration d'un construction de k-d *tree* dans $\mathbb{R}^2$

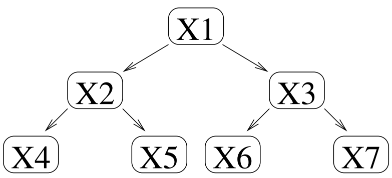


FIGURE 2 – arbre k-d construit

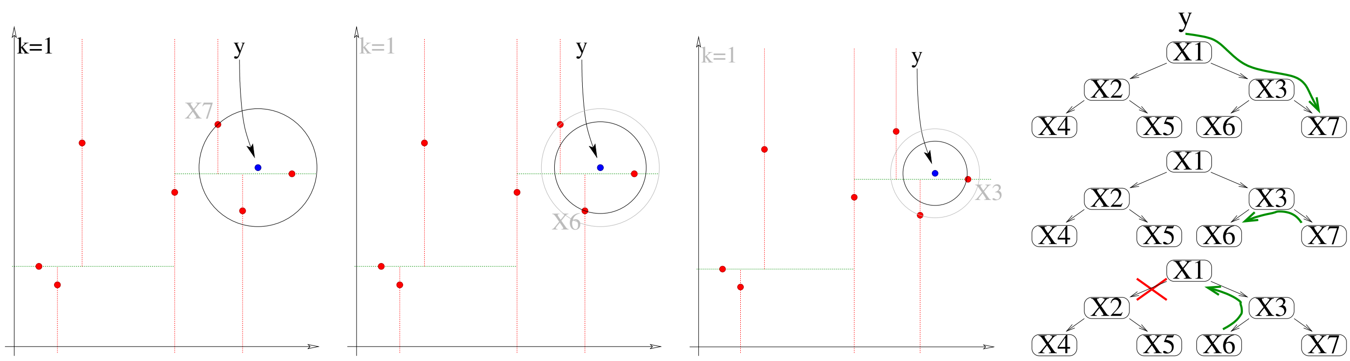
Les figures 1a à 1d illustrent les dichotomies successives sur les points représentés. Le point `X1` est le point médian le long du premier axe utilisé pour réaliser la première dichotomie. Il sert de racine à l'arbre en construction. Les points situés à gauche (resp. droite) de `X1` sont ensuite séparés à l'aide `X2` (resp. `X3`). Chaque quart de population ainsi formé ne contient qu'un seul élément qui forme ainsi une feuille de l'arbre. L'arbre final est représenté sur la figure 2.
Pour qu'un nouveau point y traverse l'arbre à partir d'un nœud donné, la coordonnée du `Xi` qui a été utilisée pour réaliser la dichotomie est comparée à la coordonnée correspondante du point y.

L'algorithme ci-dessous décrit la procédure à suivre pour identifier le plus proche voisin (i.e. pour $k = 1$) d'un nouveau point y.

FAIRE TRAVERSER y DEPUIS LA RACINE DE L'ARBRE JUSQU'À ATTEINDRE UNE FEUILLE Xi
INTIALISER K AVEC LE POINT Xi ET r AVEC $d(y, Xi)$, PLUS PETITE DISTANCE CONNUE
TANT QU'IL EXISTE UN NŒUD PÈRE
FAIRE
    REMONTER AU NŒUD PÈRE - SOIT Xp LE POINT ASSOCIÉ
    SI Xp EST PLUS PROCHE QU'UN POINT CONSERVÉ DANS K
    ALORS
        SUBSTITUER Xp À CE POINT DANS K ET METTRE r À JOUR
    SI LA BRANCHE SŒUR N'A PAS ENCORE ÉTÉ EXAMINÉE <u>ET</u>
    SI LA BOULE CENTRÉE SUR y ET DE RAYON r INTERSECTE LA ZONE ASSOCIÉE À SA BRANCHE SŒUR
    ALORS
        FAIRE TRAVERSER y DEPUIS LA RACINE DE LA BRANCHE SŒUR JUSQU'À ATTEINDRE UNE FEUILLE Xs
        SI Xs EST PLUS PROCHE QU'UN POINT CONSERVÉ DANS K
        ALORS
            METTRE À JOUR K ET r
RENVOYER K

Tester l'intersection de la boule centrée sur y et de rayon r avec la zone associée à sa branche sœur (par rapport au nœud que l'on occupe) permet de détecter la possitilité qu'un point de cette zone améliore la solution temporaire.

Les figures 3a à 3d montrent des éléments d'utilisation de l'arbre pour explorer les parties permettant l'identification du plus proche voisin de y selon l'algorithme présenté.



(a) Recherche du point le plus proche (b) Exploration de la branche voisine (c) Remontée vers le nœud supérieur (d) Exploration

FIGURE 3 – Illustration de la recherche du plus proche voisin ($k = 1$) de y sur l'arbre et du parcours associé

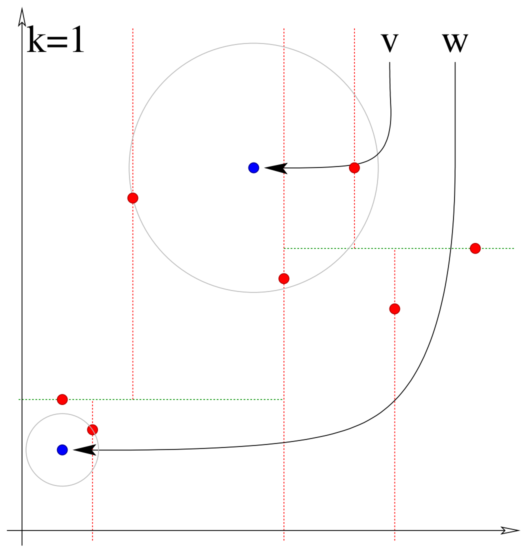


FIGURE 4 – Points v et w

Q.16 Nous considérons les points v et w représentés sur la figure 4. Appliquez pour chacun l'algorithme de recherche du plus proche voisin. Vous détaillerez les étapes pour les deux applications.

Q.17 Comparez les deux explorations précédentes. En reprenant l'hypothèse que nous travaillons sur notre population bien connue de N points, combien de points sont considérés pour identifier le plus proche voisin d'un point dans le meilleur cas ? Et dans le pire cas ?

Q.18 Proposez une adaptation de l'algorithme présenté en page 4 pour la recherche du plus proche voisin qui permette la recherche des k plus proches voisins.

Q.19 Quel type de structure de données proposeriez vous pour implémenter l'arbre et rendre les opérations de parcours les plus directes possibles ?