Projet de semestre

K-N-N : de la simplicité théorique aux défis pratiques à hautes dimensions

Étudiant : Ayemen Assadi

Abstract:

L'algorithme K-NN (K nearest neighbors) est un algorithme fondamental d'apprentissage supervisé. Bien que cet algorithme soit très fort théoriquement, il échoue systématiquement en hautes dimensions. Ce rapport a pour but de démontrer rigoureusement et empiriquement comment la concentration des distances dans des espaces de grandes dimensions invalide le principe fondamental du K-NN : l'existence de voisins réellement significatifs. Nos expériences menées sur des données artificielles et réelles affirment que la réduction des dimensions est indispensable pour pouvoir utiliser K-NN dans des applications réelles.

Université de Genève
Automne 2025

Introduction et contexte

0.1 Le K-NN: simplicité et paradoxe

K-NN est un algorithme utilisé principalement pour résoudre les problèmes de classification. On démarre de l'hypothèse que la distribution des labels préserve une forme de continuité, ceci se traduit par l'hypothèse que des données « proches » possèdent la même classification ou du moins auront presque la même distribution de probabilités a posteriori sur leurs classes respectives. La méthodologie du k-NN nécessite de donner un sens précis à la notion de "proximité". Pour cela, on introduit une mesure de distance (métrique) dans l'espace des features. Cette exigence implique que les données d'entrée soient numériques, représentées typiquement comme des vecteurs dans $\mathbf{R}^d$, où chaque dimension correspond à un attribut (feature) spécifique. Pour estimer la classification chaque nouvelle entrée, l'algorithme effectue exactement 3 étapes :

- Calculer la distance entre le nouveau point et les points d'apprentissage (training set).
- Trier ces distances et obtenir les classes des points correspondant aux k plus petites distances.
- La classe affectée au nouveau point est la classe majoritaire de k points repérés avant.

Comme tout algorithme qui se base sur le calcul des distances et qui traite les attribus comme des coordonnées, la normalisation joue un rôle crucial. Sans normalisation, les attributs ayant des grandes échelles numériques domineraient indûment les calculs de distances, et par conséquent biaiseront les prédictions. Des techniques comme standardisation (centrage-réduction) permettront d'attribuer un poids équitable à chaque attribut. Le paradoxe concernant cet algorithme réside dans sa simplicité : l'algorithme est très facile et repose sur des résultats théoriques remarquables. Notamment, le théorème de Cover et Hart (1967) [1] qui dit que l'erreur de cet algorithme est au pire deux fois l'erreur d'un classifieur optimal de Bayes. Pourtant, cette performance théorique entre en contradiction avec ses limitations pratiques, particulièrement en haute dimensionnalité, où la notion même de "voisinage" perd son sens.

Problématique: performance théorique vs dégradation pratique

La théorie développée par Cover et Hart (1967) assure le bon fonctionnement du 1-NN, mais en pratique on remarque un point de rupture critique quand la dimension de l'espace des attributs dépasse quelques dizaines ; l'existence des voisins proches (qui est l'hypothèse principale motivant l'algorithme du K-NN) s'effondre complètement à cause des conséquences d'un phénomène de l'espace qui émerge quand sa dimension devient très grande (Malédiction de la dimensionalité). Ce phénomène géométrique contre intuitif transforme la structure de l'espace comme on le peut imaginer :

- Concentration des distances : Toutes les distances entre les points deux à deux convergent vers une valeur commune
- Sparsité des données : Les points deviennent équidistants, quelle que soit leur répartition originale
- Perte de signification : La notion même de "proximité" perd son sens discriminatif

A partir de ce qu'on a annoncé précédemment, plusieurs questions se posent : « Si notre théorie est bien fondée et on sait que le classifieur 1-NN ne peut pas faire deux fois pire que le classifieur de Bayes, alors comment la malédiction de la dimensionalité invalide l'algorithme du 1-NN ? Qu'en est-il pour K-NN ? Quel est l'écart entre la performance attendue théoriquement et la performance observée empiriquement ? et à quel point cet écart devient-il critique ? Dans ce qui suit, nous développerons ces questions d'une manière rigoureuse et nous donnons aussi les résultats empiriques qui supporteront notre analyse théorique.

Démarche et méthodologie

Notre démarche dans ce rapport est biface combinant analyse théorique et validation empirique systématique:

- **Analyse théorique** : Nous donnons les fondements mathématiques qui font de l'algorithme K-NN un classifieur avec une erreur raisonnable. Nous discuterons aussi le phénomène de la malédiction de la dimensionalité et ses conséquences citées avant. Enfin, on fait le lien entre K-NN et malédiction de l'espace pour répondre à la question « comment la malédiction de l'espace invalide le K-NN ? ». L'analyse présentera aussi des résultats intermédiaires et auxiliaires qui nous permettront de bien comprendre notre problème.
- **Analyse empirique** : Nous testons notre algorithme sur des données réelles et artificielles dans des espaces de grandes et petites dimensions pour voir comment la performance du K-NN (et 1-NN) se dégrade à cause de la dimensionalité. Nous prendrons également des données en grandes dimensions et appliquerons une réduction de dimension (PCA principalement) pour voir comment K-NN se comportera dans les deux cas. Nous mettons en relief aussi le phénomène de la concentration des distances quand la dimension de l'espace de données devient très grande.

Cette méthodologie rigoureuse nous permettra de répondre de manière exhaustive à nos questions de recherche et d'établir un diagnostic précis des conditions d'application du k-NN.

Table des notations

Symbole	Signification
d	Dimension de l'espace des données
n	Nombre de points dans le jeu d'entraînement
k	Nombre de voisins dans K-NN
M	Nombre total de classes
X, x	Variable aléatoire / point de données
Y, y	Variable aléatoire / classe
R^*	Erreur du classifieur de Bayes
R_{NN}	Erreur du classifieur NN
R_{kNN}	Erreur du classifieur K-NN
$\eta_i(x)$	Probabilité a posteriori que x appartienne à la classe i
$D_{\min}$	Distance minimale entre un point requête et les points d'entraînement
$D_{\max}$	Distance maximale entre un point requête et les points d'entraînement
$L(\cdot, \cdot)$	Fonction de perte 0-1
Ω	Espace probabilisé
$S_x(\delta)$	Sphère de rayon δ centrée en x

Analyse théorique

Analyse Théorique

L'algorithme NN:

Dans cette section, nous traiterons principalement l'algorithme NN comme un cas particulier de l'algorithme K-NN (quand $K = 1$). Nous nous intéresserons à reprendre et expliquer le travail fait par les deux chercheurs T.Cover et P.Hart dans leur article « Nearest neighbor pattern classification » [1].

1.1 Théorème de Bayes:

Pour pouvoir parler de la précision de toute algorithme de classification (K-NN) en particulier, nous devons absolument avoir une référence d'un classifieur parfait et nous mesurons cette précision toujours par rapport à cette référence, et notre but sera toujours de s'approcher du parfait le maximum possible. Le classifieur de Bayes représente l'idéal théorique en classification supervisée. Contrairement aux algorithmes pratiques comme le k-NN, il s'agit d'un concept théorique qui suppose une connaissance parfaite des distributions sous-jacentes des données. Nous ne pouvons pas parler du classifieur de Bayes sans donner des éléments de la théorie de Bayes. Nous énonçons quelques définitions ainsi que le théorème de Bayes avec sa preuve. Ensuite nous nous basons sur ce théorème pour concevoir notre classifieur parfait.

Definition:
Soit $(\Omega, F, \mathbf{P})$ un espace probabilisé. Une famille $(B_i)_{i\in I} \subset F$, I fini ou dénombrable, est une partition de Ω si:

- $\forall i, j \in I : i \neq j \implies B_i \cap B_j = \emptyset$
- $\bigcup_{i\in I} B_i = \Omega$

Le thèorème suivant est simple, mais il s'agit de l'un des théorèmes les plus fondamentaux dans la théorie de probabilité:

Théorème:
Soit $(\Omega, F, \mathbf{P})$ un espace probabilisé. $(B_i)_{i\in I}$ est une partition de Ω telle que $\mathbf{P}(B_i) > 0$ pour tout $i \in I$ et soit $A \in F$:

- (**Loi de probabilité totale**): $\mathbf{P}(A) = \sum_{i\in I} \mathbf{P}(A|B_i)\mathbf{P}(B_i)$.
- (**Formule de Bayes**): Si $\mathbf{P}(A) > 0$, $\mathbf{P}(B_i|A) = \frac{\mathrm{P}(A|B_i)\mathrm{P}(B_i)}{\sum_{j\in I} \mathrm{P}(A|B_j)\mathrm{P}(B_j)}$

Preuve du théorème:

- $\mathbf{P}(A) = \mathbf{P}(\Omega \cap A) = \mathbf{P}((\bigcup_{i\in I} B_i) \cap A) = \mathbf{P}(\bigcup_{i\in I}(B_i \cap A)) = \sum_{i\in I} \mathbf{P}(B_i \cap A) = \sum_{i\in I} \mathbf{P}(A|B_i)\mathbf{P}(B_i)$
- $\mathbf{P}(B_i|A) = \frac{\mathrm{P}(B_i\cap A)}{\mathrm{P}(A)} = \frac{\mathrm{P}(A\cap B_i)}{\sum_{i\in I} \mathrm{P}(A|B_i)\mathrm{P}(B_i)} = \frac{\mathrm{P}(A|B_i)\mathrm{P}(B_i)}{\sum_{i\in I} \mathrm{P}(A|B_i)\mathrm{P}(B_i)}$

Dans la terminologie statistique, $\mathbf{P}(B_i)$ est la probabilité **à priori** de B_i et $\mathbf{P}(B_i|A)$ la probabilité **à posteriori** de B_i (sachant A). La formule de Bayes donne donc un moyen de transformer le probabilités à priori en probabilités à posteriori. Cette idée jouera un rôle essentiel dans le contexte de la classification sachant les classes des données et les points qui sont déjà classifiés.

1.2 Classifieur de Bayes:

Comment le théorème de bayes nous servira à concevoir notre classifieur parfait ? Nous répondons à cette question dans cette section.

On considère un data set $D = \{(x_1, y_1), ..., (x_n, y_n)\}$ avec $x_1, x_2, ..., x_n \in \mathbf{R}^d$ tel que d est la dimension de nos données d'entrée, et $y_1, y_2, ..., y_n \in C$ tel que $C = \{c_1, c_2, ..., c_M\}$ l'ensemble de nos classes. Notre ensemble de donnée est issu d'une distribution inconnue $\mathbf{P}(X, Y)$ et les couples de notre dataset sont choisis indépendamment entre eux. Donc:

$$\mathbf{P}(D) = \mathbf{P}((X = x_1, Y = y_1) \cap (X = x_2, Y = y_2) \cap .... \cap (X = x_n, Y = y_n)) = \prod_{i=1}^{n} \mathbf{P}(X = x_i, Y = y_i).$$

Dans un contexte pratique, on estime cette distribution. Dans notre contexte, nous travaillons avec le modèle théorique, nous ne cherchons pas à implémenter le Naive Bayes classifier, nous le considérons comme une référence parfaite pour tout classifieur supervisé. Alors, nous supposons la connaissance des distributions sous-jacentes des données.

Le classifieur de Bayes est dit naif car il repose sur une hypothèse assez forte. Bien que cette hypothèse n'est pas toujours vérifiée, le classifieur donne des résultats pratiquement raisonnables lorsque l'estimation de la distribution $\mathbf{P}(X, Y)$ reste proche de la vraie distribution. Cette hypothèse suppose que si on prend un point d'entrée $x = (x[1], ..., x[d])$, les cordonnées de x, i.e, "the features" sont indépendants conditionnellement, si on spécifie la classe à priori, les attributs sont indépendants, alors:

$$\mathbf{P}(x|y) = \mathbf{P}((x[1], x[2], ..., x[d])|y) = \prod_{i=1}^{d} \mathbf{P}(x_i|y)$$

L'idée du classifieur est de maximiser $\mathbf{P}(c|x)$. L'événement $(Y = c|X = x)$ signifie que la classe est c sachant qu'on a le point d'entrée x, alors la classification h prédite par le classifieur est :

$$\begin{aligned}
h &= \arg\max_{c \in C} \mathbf{P}(c|x) \\
&= \arg\max_{c \in C} \frac{\mathbf{P}(x|c)\mathbf{P}(c)}{\mathbf{P}(x)} \\
&= \arg\max_{c \in C} \mathbf{P}(x|c)\mathbf{P}(c) \\
&= \arg\max_{c \in C} \prod_{i=1}^{d} \mathbf{P}(x_i|c)\mathbf{P}(c) \\
&= \arg\max_{c \in C} \log(\prod_{i=1}^{d} \mathbf{P}(x_i|c)\mathbf{P}(c)) \\
&= \arg\max_{c \in C} [\sum_{i=1}^{d} \log(\mathbf{P}(x_i|c)) + \log(\mathbf{P}(c)]
\end{aligned}$$

Note de supervision: On note ici que l'hypothese du classifieur Naive Bayes (caractéristiques independantes et Gaussiennes) est un cas particulier qui simplifie l'estimation des densités *a posteriori* des classes mais qui n'est pas nécessaire pour la suite.

On rappelle donc la différence entre l'optimalité du classifieur de Bayes (connaissance abstraite des distributions) et le classifieur Naive Bayes (modèle simplifié empirique pour ces distributions)

Donc, théoriquement nous ne pouvons pas faire mieux que le classifieur naif de Bayes, car ce dernier nécessite les conditions parfaites pour la classification et tout autre classifieur pratique aura moins d'informations, en particulier **la distribution exacte** $\mathbf{P}(X, Y)$.

Le modèle théorique qu'on a présenté a joué le rôle de la référence dans l'article de Cover et Hart. Nous nous intéressons à l'efficacité théorique de cet algorithme par rapport au classifieur naif de Bayes.

1.3 Erreur de l'algorithme 1-NN pour la classification binaire (M=1):

Soit x un point issu d'une variable aléatoire X qu'on veut classifier à l'aide de notre algorithme NN. On suppose que x_k est le plus proche voisin de x avec une classe y_k, on note y la vraie classe de x issu de la variable aléatoire Y. On considère les variables aléatoires engendrant le dataset D: $(X_1, Y_1), (X_2, Y_2), ..., (X_n, Y_n)$,on suppose que ces variables aléatoires sont indépendants et identiquement distribuées.
On définit l'erreur de classification ainsi : $R_{NN}(n) = \mathbf{E}[L(Y, Y_k)]$ avec L une fonction décrivant "the loss" la pénalisation de l'erreur et n désignant le nombre de points de notre dataset. $R_{NN} = \lim_{n\to\infty} R_{NN}(n)$ est l'erreur quand le cardinal du training set (D) tend vers l'infini. On note aussi R^* l'erreur du classifieur de Bayes parfait.
La fonction "loss" L que nous considérons est : $L(h(X), Y) = 1_{\{h(X)\neq Y\}}$ avec h le classifieur 1-NN est Y la vraie classe du point donné par X. Nous énonçons le théorème essentiel qui valide le classifieur NN.

Théorème: Soit X un espace métrique séparable et $x \in X$ est un point qu'on veut classifier.
f_1 et f_2 les densités de probabilité des distributions des classes 1 et 2 respectivement telles que: f_1 et f_2 sont continues en x ou bien $f_1(x) \neq 0$ et $f_2(x) \neq 0$.
Alors, l'erreur R de l'algorithme NN à M=2 classes est borné par :

$$R^* \leq R_{NN} \leq 2R^*(1 - R^*)$$

Avant démontrer ce théorème, nous aurons besoin d'un lemme:

Lemme:(Convergence du plus proche voisin)
Soit x et $x_1, x_2, ...$ des variables aléatoires indépendant et identiquement distribués qui prennent valeurs dans un espace métrique séparable X.
On note $x_k \in \{x_1, x_2, ...\}$ le voisin le plus proche de x. Alors, $n \to \infty \implies x_k \to x$ avec une probabilité égale à 1. Autrement dit, quand la taille de notre dataset tend vers l'infini, le voisin le plus proche de x devient arbitrairement proche à x lui même.

Preuve du lemme:
Nous définissons $S_x(\delta) = \{y \in X : d(x, y) \leq \delta\}$ la sphère centré en x de rayon δ.
Premier cas : soit x un bon point de X, c'est à dire que pour tout δ, la probabilité qu'un élément du dataset se trouve dans $S_x(\delta)$ n'est pas nulle, on note : $\mathbf{P}(S_x(\delta)) > 0$.
Alors la probabilité que tous les points soient en dehors de $S_x(\delta)$ pour un δ donné est : $P = (1 - \mathbf{P}(S_x(\delta)))^n \to 0$ quand le nombre de points n de notre dataset tend vers l'infini. Donc, plus de points dans notre dataset plus qu'on force que le plus proche voisin soit arbitrairement proche de x.
Deuxième cas : on définit l'ensemble $N = \{x \in X : \exists r_x > 0 : \mathbf{P}(S_x(r_x)) = 0\}$ qui décrit les points isolés de notre distribution; si x appartient à N cela veut dire que x possède une sphère dans laquelle les points de notre dataset ne peuvent pas se trouvrer.
Soit $x \in N$, par définition de la séparabilité de X, il existe un sous ensemble dénombrable A qui est dense dans X, alors : $\forall x \in N : \exists a_x \in A : a_x \in S_x(\frac{r_x}{3})$, donc, il existe une sphère $S_{a_x}(\frac{r_x}{2}) \subset S_x(r_x)$,
alors $\mathbf{P}(S_{a_x}(\frac{r_x}{2})) \leq \mathbf{P}(S_x(r_x)) = 0$, d'où $\mathbf{P}(S_{a_x}(\frac{r_x}{2})) = 0$.
On a $N \subset \bigcup_{x\in N} S_{a_x}(\frac{r_x}{2})$ avec $\forall x \in N : \mathbf{P}(S_{a_x}(\frac{r_x}{2})) = 0$, alors : $\mathbf{P}(N) = 0$.

Synthèse de la preuve: le point x est soit un bon point, c'est à dire il n'appartient pas à N alors, on déduit que le plus proche voisin est arbitrairement proche. Soit x est dans N avec probabilité 0.

Preuve du théorème :

Soit $x \in X$ un point qu'on veut classifier à l'aide du 1-NN,y sa vraie classe et x' son plus proche voisin avec y' la classe de ce dernier.

On définit $\hat{\eta} = \mathbf{P}(y = c_1|X = x)$ la probabilité à priori de la classe c_1 et $\mathbf{P}(y = c_2|X = x) = 1 - \hat{\eta}(x)$ est la probabilité que x soit dans la classe c_2 c'est à dire que $y = c_2$.

On définit $\epsilon(x, x') := \mathbf{P}(L(y, y') = 1|x, x')$ l'erreur conditionnelle de la classification de x comme la probabilité d'avoir une erreur en considérant x et son plus proche voisin x'.

On a :

$$\begin{aligned}\epsilon(x, x') &= \mathbf{P}(L(y, y') = 1|x, x') \\ &= \mathbf{P}(y = c_1, y' = c_2|x, x') + \mathbf{P}(y = c_2, y' = c_1|x, x') \\ &= \mathbf{P}(y = c_1|x)\mathbf{P}(y' = c_2|x') + \mathbf{P}(y = c_2|x)\mathbf{P}(y' = c_1|x') \\ &= \hat{\eta}(x)(1 - \hat{\eta}(x')) + (1 - \hat{\eta}(x))\hat{\eta}(x')\end{aligned}$$

Puisque x' est le plus proche voisin alors par le lemme: $x \to x'$ quand le nombre de points de notre dataset est très grand. Les densités f_1, f_2 sont continues au point x donc $\hat{\eta}$ l'est aussi. Donc : $x \to x' \implies \hat{\eta}(x) \to \hat{\eta}(x')$.

Donc $\epsilon(x, x') \to 2\hat{\eta}(x)(1 - \hat{\eta}(x))$ quand le nombre n de données tend vers l'infini.

Si $\epsilon^*(x)$ est l'erreur de classification pour le classifieur de Bayes.

Alors par symétrie :

$$\begin{aligned}\epsilon_{NN}(x) &= 2\hat{\eta}_1(x)(1 - \hat{\eta}_1(x)) \\ &= 2\epsilon^*(x)(1 - \epsilon^*(x))\end{aligned}$$

Nous venons de démontrer que si on a un nombre infini de points dans notre dataset, le classifieur NN prédit la classe d'un nouveau point à erreur $\epsilon^*(x)(1 - \epsilon^*(x))$ avec $\epsilon^*(x)$ l"erreur faite si on classifie ce même point x en utilisant le classifieur de Bayes.

Pour l'erreur globale:

$$\begin{aligned}R_{NN} &= \lim_n \mathbf{E}[\epsilon(x, x')] \\ &= \mathbf{E}[2\hat{\eta}(x)(1 - \hat{\eta}(x))] \\ &= \mathbf{E}[2\epsilon^*(x)(1 - \epsilon^*(x))] \\ &= 2\mathbf{E}[\epsilon^*(x)] - 2\mathbf{E}[(\epsilon^*(x))^2] \\ &\leq 2\mathbf{E}[\epsilon^*(x)] - 2(\mathbf{E}[\epsilon^*(x)])^2 && (\mathbf{E}[X^2] \geq (\mathbf{E}[X])^2 \text{ Cauchy-Schwartz}) \\ &= 2R^* - 2(R^*)^2 = 2R^*(1 - R^*)\end{aligned}$$

Pour l'autre côté de l'inégalité, on a :

$$\begin{aligned}R_{NN} &= \mathbf{E}[2\epsilon^*(x)(1 - \epsilon^*(x))] \\ &= \mathbf{E}[\epsilon^*(x) + \epsilon^*(x)(1 - 2\epsilon^*(x))] \\ &= \mathbf{E}[\epsilon^*(x)] + \mathbf{E}[\epsilon^*(x)(1 - 2\epsilon^*(x))] \\ &= R^* + \mathbf{E}[\epsilon^*(x)(1 - 2\epsilon^*(x))] \geq R^*\end{aligned}$$

Maintenant on traite le cas où on a plusieurs classes ($M \geq 2$):

Théorème:(Théorème d'extension)
Soit X un espace métrique séparable et $x \in X$ un point qu'on veut classifier.
$f_1, f_2, ..., f_M$ les densités de probabilité des distributions des classes $1, 2, .., M$ respectivement telles que:
$f_1, f_2, ..., f_M$ sont continues en x ou bien $f_i(x) \neq 0 \quad \forall i \in [[1, M]]$
Alors, l'erreur R_{NN} de l'algorithme NN à M classes est borné par :

$$R^* \leq R_{NN} \leq R^*(2 - \frac{M}{M-1}R^*)$$

Preuve du théorème d'extension:

On travaille avec les même données et on généralise les notations et définitions de la preuve précédente.
On a $x \to x'$ avec probabilité 1. Alors, $\hat{\eta}_1(x) \to \hat{\eta}_1(x')$ et $\hat{\eta}_2(x) \to \hat{\eta}_2(x')$ aussi avec probabilité 1. Donc:

$$\begin{aligned}
\epsilon(x, x') &= \mathbf{P}[L(y, y') = 1 | x, x'] \\
&= \mathbf{P}[y \neq y' | x, x'] \\
&= \sum_{i=1}^{M} \hat{\eta}_i(x)(1 - \hat{\eta}_i(x')) && (x' \text{ ne doit pas être dans la même classe que x}) \\
&= \sum_{i=1}^{M} \hat{\eta}_i(x) \quad - \sum_{i=1}^{M} \hat{\eta}_i(x')\hat{\eta}_i(x) \\
&= 1 - \sum_{i=1}^{M} \hat{\eta}_i(x')\hat{\eta}_i(x)
\end{aligned}$$

Pour un classifieur de Bayes, si on note : $\max_{i \in [[1,M]]}\{\hat{\eta}_i(x)\} = \hat{\eta}_k(x)$ alors, l'erreur de classification est :

$$\epsilon^*(x) = 1 - \max_{i \in [[1,M]]}\{\hat{\eta}_i(x)\} = 1 - \hat{\eta}_k(x)$$

On a:

$$(M-1)\sum_{i \neq j}^{M}(\hat{\eta}_i(x))^2 \geq [\sum_{i \neq j}^{M}\hat{\eta}_i(x))^2] \qquad (M > 1 \text{ et on a utilisé Cauchy Scwartz pour la somme})$$

Donc:

$$(M-1)\sum_{i \neq j}^{M}(\hat{\eta}_i(x))^2 \geq [1 - \hat{\eta}_k(x)]^2 = (\epsilon^*(x))^2$$

Alors:

$$\begin{aligned}
&(M-1)\sum_{i=1}^{M}(\hat{\eta}_i(x))^2 \geq \epsilon^*(x) + (M-1)(\hat{\eta_k}(x))^2 = (\epsilon^*(x))^2 + (M-1)(1 - \epsilon^*(x))^2 \\
&\iff \sum_{i=1}^{M}(\hat{\eta}_i(x))^2 \geq \frac{(\epsilon^*(x))^2}{M-1} + (1 - \epsilon^*(x))^2 \\
&\iff 1 - \epsilon_{NN}(x) \geq \frac{(\epsilon^*(x))^2}{M-1} + 1 - 2\epsilon^*(x) + (\epsilon^*(x))^2 \\
&\iff \epsilon_{NN}(x) \leq 2\epsilon^*(x) - \frac{(M}{M-1}\epsilon^*(x))^2
\end{aligned}$$

On fait la même transition qu'on a faite pour le théorème précédent :

$$R_{NN} \leq 2R^* - \frac{M}{M-1}(R^*)^2$$

Nous avons donné les bases théoriques qui valident l'algorithme 1-NN. Nous avons conclu que notre algorithme simple est au pire deux fois moins performant le classifieur parfait dans le cas de deux classes seulement et l'erreur de ce dernier devient plus petite que $2R^*$ si le nombre de classes est supérieur à 2.
La prochaine étape est de faire le lien avec le K-NN, quelle est la performance théorique atteinte par notre classifieur si nous augmentons le nombre de voisins considérés ?

Le classifieur K-NN

Le classifieur K-NN ou l'algorithme K-NN suit le même principe que l'algorithme NN, en fait le NN est un cas particulier de l'algorithme du K-NN, car dans le premier algorithme nous considérons spécifiquement le plus proche voisin, or dans le K-NN nous travaillons avec les k voisins les plus proches, choisir k impair empêche le cas d'égalité. k est appelé un **hyperparamètre** de l'algorithme, le choix de ce dernier affecte bien la performance de l'algorithme, il y a des méthodes et des conseils pour le choix de ce dernier mais il n'y a pas une règle à suivre, nous discuterons l'impact du choix de k aussi dans cette section.

2.1 Bias-Variance Tradeoff:

Le compromis Biais Variance est un phénomène qui se produit quand on essaie d'entraîner un estimateur quelconque (Le classifieur K-NN dans notre cas qui estime la classe du nouveau point qu'on souhaite classifier). Nous donnons dans cette section une justification mathématique ainsi que des exemples réels de ce phénomène.
Nous considérons dans cette partie le classifieur K-NN comme un **ERM** (Empirical Risk Minimization predictor), c'est à dire un estimateur qui essaie toujours de minimiser l'erreur empirique. Pour ce faire, nous définissons :

X : l'espace de données $\quad Y = [[1, M]]$: l'espace des labels $\quad S_n = \{(x_1, y_1), \ldots, (x_n, y_n)\}$: training set
La classe des hypothèses du KNN est : $\mathcal{H}_{KNN} = \{h : X \to Y \mid h(x) = \text{classe majoritaire de } N_k(x)\}$
avec : $N_k(x)$: les k plus proches voisins de x dans S_n

Le classifieur K-NN minimize l'erreur 0-1 empiriquement, alors on cherche:

$$h^*_{KNN} = \arg \min_{h \in \mathcal{H}_{KNN}} \frac{1}{n} \sum_{i=1}^{n} 1_{\{h(x_i) \neq y_i\}}$$

Et puisque les voisins sont tous traités de manière égale, alors:

l'estimation de la probabilité à posteriori de la classe i est donnée par : $\mathbf{P}(Y = i | X = x) = \hat{\eta}_i(x) = \frac{1}{k} \sum_{j \in N_k(x)} 1_{\{y_j = i\}}$

La question du choix entre traiter les voisins d'une manière égale ou utiliser des coefficients dont la somme est 1 est toute à fait logique, mais nous ne la couvrons pas dans ce travail.
Pour traiter la question du compromis, nous prenons une hypothèse h_{S_n} et nous décomposons notre **Loss** en deux

parties:

$$L(h_{S_n}) = \epsilon_{app} + \epsilon_{est} \quad \text{where : } \epsilon_{app} = \min_{h \in \mathcal{H}} L(h) \text{ (approximation error)}, \quad \epsilon_{est} = L(h_{S_n}) - \epsilon_{app} \text{ (estimation error)}$$

- **The approximation error:** Cette quantité est égale au minimum d'erreur possible sur notre classe $\mathcal{H}$. Il s'agit de l'erreur engendrée par la restriction de nos estimateurs sur une classe particulière d'estimateurs. Cette erreur ne dépend pas de la taille du training set. Dans notre cas, nous pouvons considérer notre erreur d'approximation comme la différence entre le k optimal et le k qu'on a choisi, ajouter d'autres points n'influencera pas cette erreur car elle est liée à la nature de la distribution des points et non à la taille du set.
- **Estimation Error:** Cette quantité représente la différence entre l'approximation de l'erreur et l'erreur atteint par l'ERM. L'erreur d'estimation est le résultat de la différence entre la vraie erreur et l'erreur estimé car en travaillant avec un dataset fini, nous pouvons qu'estimer empiriquement la vraie erreur de notre estimateur. Cette erreur dépend de la taille de notre échantillon.

Notre objectif est de minimiser l'erreur $L(h_{S_n})$. Une classe d'hypothèses H très riche (par exemple un k très petit dans le cas du K-NN) permet de réduire l'erreur d'approximation, mais augmente l'erreur d'estimation, car le modèle dépend fortement des données, ce qui conduit à un sur-apprentissage (overfitting). À l'inverse, une classe d'hypothèses trop restreinte (un k très grand) réduit l'erreur d'estimation au prix d'une augmentation de l'erreur d'approximation, entraînant un sous-apprentissage (underfitting). Ce compromis entre sur-apprentissage et sous-apprentissage est appelé compromis biais–variance.

2.2 Bornes théoriques pour KNN:

La discussion précédente était dans le cadre fini, c'est à dire nous travaillons avec des datasets de taille finie (le cas réelle), dans cette section nous s'intéressons à quelques résultats théoriques intéressants spécifique pour la classification binaire (quand on a deux classes). Nous considérons aussi que k est impaire pour éviter les cas d'églité (la théorie là dessus n'est pas discuté dans ce travail). Nous considérons $\eta(X)$ la probabilité que X (valeur aléatoire qui décrit l'entrée qu'on veut classifier) soit dans la première classe, $\eta(X) = \mathbf{P}(Y = 1|X)$.

Théorème:
Si on définit R^* l'erreur du classifieur parfait de Bayes et L_{KNN} l'erreur de l'algorithme K-NN.
Alors,

$$R^* \leq \cdots \leq R_{2k+1} \leq R_{2k-1} \leq \cdots \leq R_{3NN} \leq R_{NN} \leq 2R^*$$

Preuve du théorème:

On note : $R_{KNN} = \mathbf{E}[\alpha(\eta(X))]$ l'erreur de notre classifieur KNN, avec :

$$\forall p \in [0,1] : \alpha_K(p) = \mathbf{P}(\text{erreur K-NN} \mid \eta(X) = p)$$

On introduit aussi les variables aléatoires $X_1, X_2, ..., X_n$ décrivant les points de notre dataset et $Y_1, Y_2, ..., Y_n$ leurs classes respectives. On note $X_{(1)}, X_{(2)}, ..., X_{(K)}$ les K plus proches voisins de X et $Y_{(1)}, Y_{(2)}, ..., Y_{(K)}$ leurs classes respectives.

On a pour tout $p \in [0,1]$:

$$\begin{aligned}
\alpha_K(p) &= P(\sum_{i=1}^{K} Y_{(i)}(x) > \frac{K}{2}\ ,\ Y = 0) + \mathbf{P}(\sum_{i=1}^{K} Y_{(i)}(x) \leq \frac{K}{2}\ ,\ Y = 1) \\
&= (1-p)\mathbf{P}(Bin(K,p) > \frac{K}{2}) + p\ \mathbf{P}(Bin(K,p) \leq \frac{K}{2}\ (\textbf{Bin} \text{ désigne l'expérience binomiale}) \\
&= (1-p)\ \mathbf{P}(Bin(K,p) > \frac{K}{2}) + p\ (1 - \mathbf{P}(Bin(K,p) > \frac{K}{2})) \\
&= p + (1-2p)\ \mathbf{P}(Bin(K,p) > \frac{K}{2}) \text{ (Cette écriture est pratique si } p < \frac{1}{2}) \\
&= (1-p) + (2p-1) + (1-2p)\mathbf{P}(Bin(K,p) > \frac{K}{2} \\
&= (1-p) + (2p-1)[1 - \mathbf{P}(Bin(K,p) \leq \frac{K}{2})] \\
&= (1-p) + (2p-1)\mathbf{P}(Bin(K,1-p) > \frac{K}{2})
\end{aligned}$$

Donc:

$$\alpha_K(p) = \min\{p, 1-p\} + |2p-1|\ \mathbf{P}(Bin(K, \min\{p,1-p\}) > \frac{K}{2}) \quad \forall p \in [0,1]$$

On observe clairement que : $\alpha_K(p) = \alpha_K(1-p) \quad \forall p \in [0,1]$.

Nous avons déjà montré que $R_{NN} \leq R^*$, et on sait bien que le classifieur de Bayes est parfait donc R^* sert comme une borne inférieur de toutes les erreurs qu'on manipule. Donc nous avons besoin juste de montrer que : $R_{2k+1} \leq R_{2k-1}$.

Pour cela on a :

$$\begin{aligned}
&\alpha_{2K+1}(p) \leq \alpha_{2K-1}(p) \text{ (On suppose que p< } \frac{1}{2} \text{ sinon on choisit 1-p)} \\
&\Leftrightarrow p + (1-2p)\mathbf{P}(Bin(2K+1,p) > \frac{2K+1}{2}) \leq p + (1-2p)\ \mathbf{P}(Bin(2K-1,p) > \frac{2K-1}{2}) \\
&\Leftrightarrow \mathbf{P}(Bin(2K+1,p) > K + \frac{1}{2}) \leq \mathbf{P}(Bin(2K-1,p) > K - \frac{1}{2})
\end{aligned}$$

La dernière ligne des équivalence est vraie pour la simple raison suivante (interprétation combinatoire): Si $p < \frac{1}{2}$, atteindre plus que $K + \frac{1}{2}$ succès avec $2K+1$ essais, est peu probable que atteindre $K - \frac{1}{2}$ succès avec $2K-1$ essais.

Donc, en passant aux espérances:

$$R_{2K+1} = \mathbf{E}[\alpha_{2K+1}(\eta(X))] \leq \mathbf{E}[\alpha_{2K-1}(\eta(X))] = R_{2K-1}$$

Conclusion :

$$R^* \leq \cdots \leq R_{2k+1} \leq R_{2k-1} \leq \cdots \leq R_{3NN} \leq R_{NN} \leq 2R^*$$

Malédiction de l'espace

On pourra penser que l'espace à hautes dimensions se comporte à peu près de la même façon que l'espace imaginable (< 4 dimensions), c'est à dire que les règles qui régissent l'espace imaginable ont juste besoin d'une extnesion de quelques dimensions. En vérité ce n'est pas le cas, et c'est cela qu'on discutera dans cette section.

3.1 Rareté des données (Data sparsity):

Data sparsity est un phénomène qui apparaît lorsqu'on traite des données dans des espaces à hautes dimensions. Il s'agit simplement du fait que lorsque la dimension de l'espace des données croît, il devient difficile de trouver des échantillons de données (samples) dans l'espace. Formellement, cette idée sera claire.
On se donne un hypercube de l'espace de données $\Omega = [0,1]^d$ avec d la dimension de cet espace. On suppose que l'échantillon de points est tiré uniformément de ce cube. On note l la longueur du plus petit hypercube qui contient par exemple les 10 plus proches voisins d'un poit $x \in \Omega$ et n le nombre des points d'entraînement.

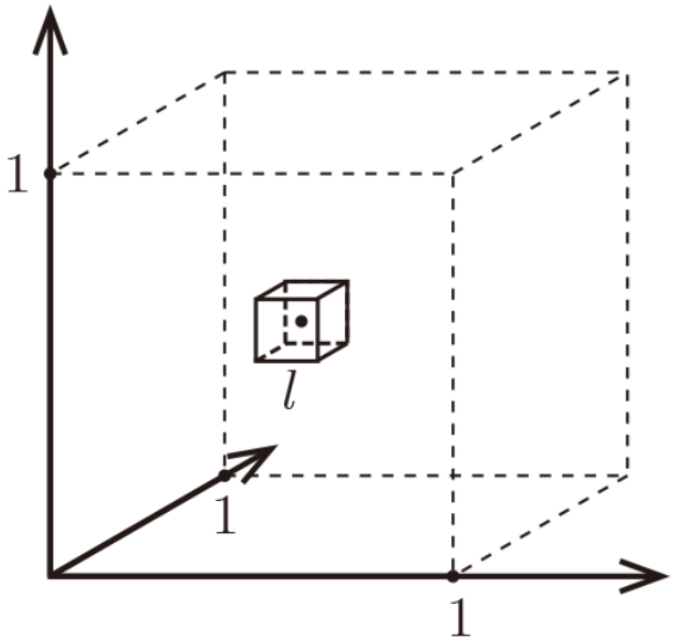

Figure 1: Illustration de l'hypercube

Puisque le volume est égal à 1 et les points sont uniformément distribués, on peut faire l'approximation suivante :

$$n\, l^d \approx k \implies l^d \approx \frac{k}{n} \implies l \approx (\frac{k}{n})^{1/d}$$

Donc, la conclusion immédiate pour notre cas est: Si la dimension d est très grande alors, il nous faut un nombre immense de points d'entraînement pour avoir l raisonnable. Exemple: Si on fixe $l = 0.1$ alors on aura besoin de $n = \frac{k}{l^d} = k\ 10^d$. Notons que le nombre de points d'entraînement requis croît exponentiellement avec la dimension. C'est l'une des raisons majeures pour laquelle le classifieur KNN devient inefficace dans les grandes dimensions. Tout simplement, il n'y a pas assez de proches voisins pour dire que les points proches ont les même caractéristiques. La notion de **proche** devient **rare**!

3.2 Concentration des distances:

Un autre phénomène qui apparaît dans l'espace quand sa dimension devient considérablement grande est : **La concentration des distances**. Il s'agit du fait que la distance entre les paires de points de l'espace converge vers la même valeur, en particulier la distance au plus proche voisin devient égale à la distance au plus loin, ce qui rend les algorithmes reposant sur la distances inefficaces.
La distance traité dans ce rapport est en particulier **La distance de Minkowski**.
On commence par quelque définitions avant d'énoncer le théorème qui illustre la concentration des distances sous certaines conditions théoriques.

- m : paramètre de la dimension $\in \mathbf{N}$
- $F_{data_1}, F_{data_2}, ...$: densités des distributions des points d'entrées à chaque dimension m
- $F_{query_1}, F_{query_2}, ...$: densités de distributions des points à classifier
- n : Nombre de points d'entraînement.
- $\forall m : P_{m,1}, P_{m,2}, ..., P_{m,n}$: les points d'entraînement qui sont tous indépendants et suivent la même loi P_m
- $Q_m \sim F_{query_m}$: un point à classifier tiré de la distribution de F_{query_m}
- $0 < p < \infty$ une constante
- d_m: représente la distance. Elle s'agit d'une fonction qui prend un point du domaine de F_{data_m} et un point du domaine de Q_m et donne un nombre positif.
- $D_{\min}^{(m)} = \min\{d_m(P_{m,i}, Q_m) | 1 \leq i \leq n\}$
- $D_{\max}^{(m)} = \max\{d_m(P_{m,i}, Q_m) | 1 \leq i \leq n\}$

Théorème:
On garde les définitions d'avant, Si:

$$\lim_{m\to\infty} \mathbf{Var}[\frac{(d_m(P_{m,1}, Q_m))^p}{\mathbf{E}[(d_m(P_{m,1}, Q_m))^p]}] = 0$$

Alors,

$$\forall \epsilon > 0 : \lim_{m\to\infty} \mathbf{P}[D_{\max}^{(m)} \leq (1+\epsilon) D_{\max}^{(m)}] = 1$$

On a choisi $P_{m,1}$ car tous les points $P_{m,i}$ suivent la même distribution P_m donc il n'y a pas de différence.
Preuve:
Soit $\mu_m = \mathbf{E}[(d_m(P_{m,i}, Q_m))^p]$ (μ_m est indépendant de i pour la même raison qu'on a dit avant)
et $V_m = \frac{(d_m(P_{m,1}, Q_m))^p}{\mu_m}$.
Partie 1: Prouver que $V_m \to_p 1$ (Convergence en probabilité)
On a $\mathbf{E}[V_m] = \frac{\mathrm{E}[(d_m(P_{m,i}, Q_m))^p]}{\mu_m} = 1$, donc $\lim_{m\to\infty} \mathbf{E}[V_m] = 1$.
La condition du théorème nous dit que : $\lim_{m\to\infty} \mathbf{Var}[V_m] = 0$, donc par l'inégalité de Benyamé-Tchebyshev :

$$\forall \epsilon > 0 : \lim_{m\to\infty} \mathbf{P}[|V_m - 1| < \epsilon] = 0 \implies V_m \to_p 1$$

Partie 2:

On montre que le résultat de la partie 1 implique bien la conclusion du théorème.

On considère:

$$Y_m = (\frac{d_m(P_{m,1}, Q_m)}{\mu_m^{1/p}}, \frac{d_m(P_{m,2}, Q_m)}{\mu_m^{1/p}}, ..., \frac{d_m(P_{m,n}, Q_m)}{\mu_m^{1/p}}) = (V_m^{1/p}, V_m^{1/p}, ..., V_m^{1/p})$$

. Puisque la fonction $x \to x^{1/p}$ avec p une constante est continue, alors $V_m^{1/p} \to_p 1$,

et par conséquent $Y_m \to_p (1, 1, ..., 1)$.

les fonctions min et max sont aussi continues alors on conclut de même : $\max(Y_m) \to_p \max(1, 1, ..., 1) = 1$ et $\min(Y_m) \to_p \min(1, 1, ..., 1) = 1$.

Alors:

$$\frac{\max(Y_m)}{\min(Y_m)} \to_p \frac{1}{1} = 1$$

On note que $D_{\min}^{(m)} = \mu_m^{1/p} \ \min(Y_m)$ et $D_{\max}^{(m)} = \mu_m^{1/p} \max(Y_m)$ Alors:

$$\frac{D_{\max}^{(m)}}{D_{\min}^{(m)}} \to_p 1$$

D'où:

$$\lim_{m\to\infty} \mathbf{P}[D_{\max}^{(m)} \leq (1+\epsilon) D_{\min}^{(m)}] = \lim_{m\to\infty} \mathbf{P}[|\frac{D_{\max}^{(m)}}{D_{\min}^{(m)}} - 1| \leq \epsilon] = 0$$

Dans la pratique, nous observons dans nombreux cas (images et vidéos, textes et NLP,...) que les points d'entrées sont en distance moyenne entre eux dû à la haute dimension de l'espace. C'est à dire la distance relative à la moyenne des distances ne varie pas autant (~ 0), c'est ce qui rend ce théorème très fort (la condition est à peu près toujours présente). Donc notre classifieur KNN qui prédit les sorties en se basant sur la comparaison des distances devient inefficace, comme $D_{\max}^{(m)}$ et $D_{\min}^{(m)}$ deviennent à peu près les mêmes.

Analyse Empirique

Analyse Empirique

Expérience 1: Performance vs Dimension

Dans la première expérience, nous illustrons la dégradation de la performance du classifieur KNN quand la dimension de notre espace d'entrée devient de plus en plus grande.
Les vecteurs sont échantillonés i.i.d et les dimensions sont indépendantes, pour chaque dimension, nous générons des données artificielles (test set et training set) qui ont des composantes informatives contribuant à l'identification de la classe des données et d'autres qui sont soit répétées soit sans intérêt à la classification (irrelevant). Nous entraînons notre classifieur sur le training set, et nous prenons **the accuracy score** de notre classifieur sur l'ensemble de validation (testing set).
On obtient la figure suivante: Le phénomène est très clair, la dimension représente le nombre de composantes, la

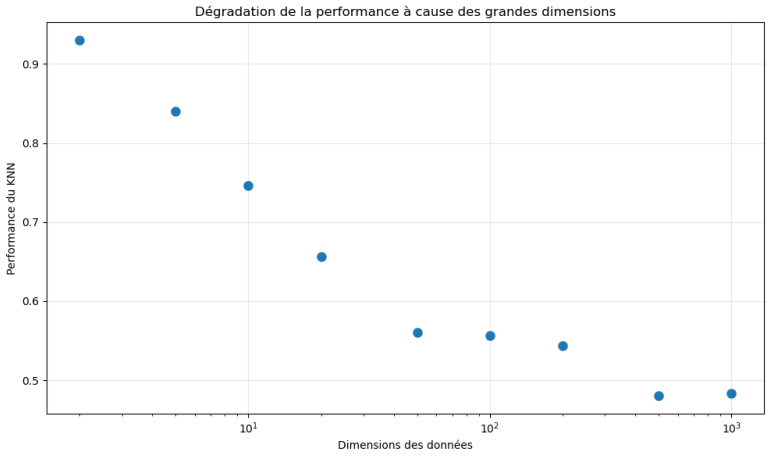

Figure 2: Expérience 1

plupart des composantes de notre ensemble de données s'agit du bruit (similaire aux cas réels), donc le classifieur est perturbé, ce qui explique la dégradation de la performance.

Expérience 2: Concentration des distances

Dans cette expérience, nous validons le théorème qui illustre la concentration des distances quand la dimension devient plus grande et la variance des distance de leur moyenne est minimale (la condition du théorème), cette condition est vraie dans la plupart des cas pratiques.
Comme l'expérience précédente, nous générons des données artificielles (training set et query set), **query set** s'agit de l'ensemble des points pour lesquels on considère les voisins, c'est pour ces points-là qu'on trace le rapport $D_{\text{max}}^{(m)}/D_{\text{min}}^{(m)}$ en fonction de la dimension. Cette fois-ci, les donées sont conçues de façon telle qu'il y a deux clusters représentant les deux classes des points.
Nous obtenons le graphique suivant:

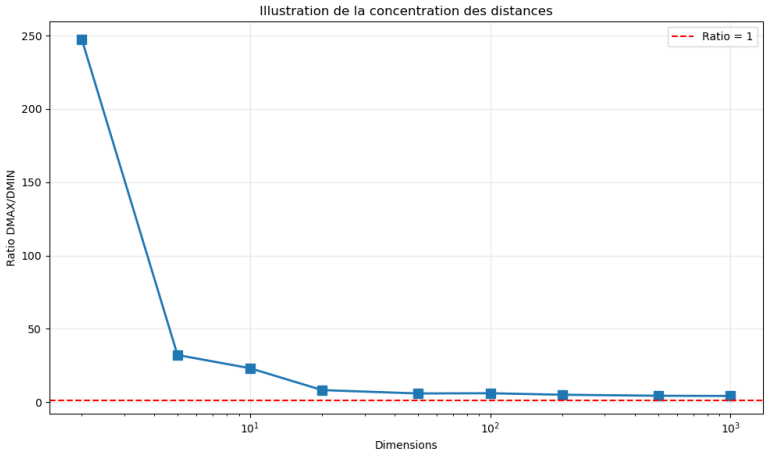

Figure 3: Expérience 2

La figure illustre bien le problème de la concentration des distances, l'un des problèmes majeurs qui rendent le KNN inefficace pour tel type de points.

Expérience 3: Data Manifold

Cette expérience illustre le cas où le classifieur KNN fonctionne très bien même si nous nous trouvons dans un espace à dimension 784 ! Il s'agit du dataset **MNIST** qui s'agit des nombres de 0 à 9 écrits à la main.
Dans cette expérience nous testons la performance du KNN sur MNIST directement, après nous effectuons une réduction de dimension (En utilisant PCA) et nous reclassifions les points à dimension réduite à l'aide du KNN.
Nous obtenons le graphique suivant:

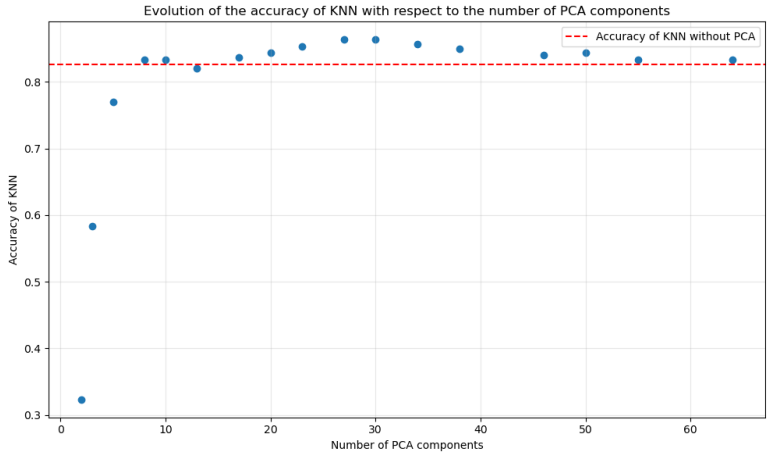

Figure 4: Expérience 3

Ce graphique nous dit que nous pouvons atteindre une grande précision en se contentant juste d'un petit nombre de composantes au lieu de travailler avec 784, ce qui implique que notre dataset en fait se trouve dans un sous espace à petite dimension du grand espace de dimensions 784. Cela explique pourquoi le KNN performe très bien. N'oublions pas que les données sont bien structuré, c'est à dire les '1' se ressemblent, les '2' se ressemblent, ainsi de suite. Il y aura évidemment des cas où le classifieur se trompera, mais en moyenne sa précision est bonne.

Expérience 4: Choix du k

Cette expérience illustre l'impact du choix de l'hyperparamètre k (nombre de voisins à considérer) sur la performance du classifieur pour les deux ensembles, ensemble d'entraînement et l'ensemble de validation. Nous prenons des point du dataset MNIST et nous conparons les performances du classifieur sur l'ensemble d'entraînement et l'ensemble de validation.
On obtient le graphique suivant:

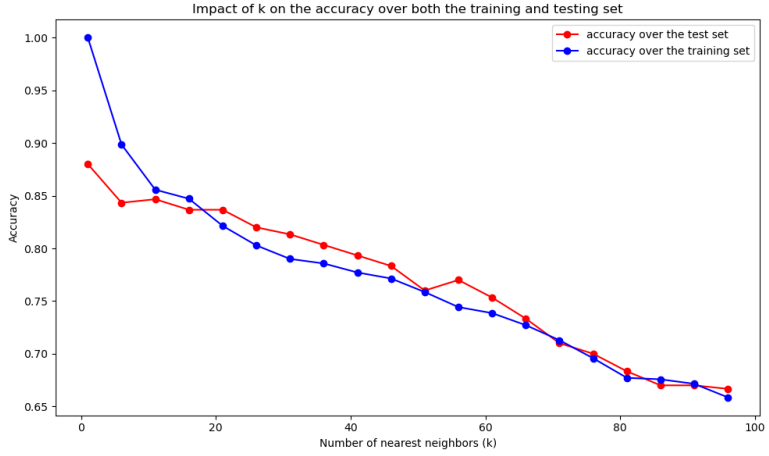

Figure 5: Expérience 4

Le graphique montre clairement que les performances sur les deux ensembles décroissent quand k devient de

plus en plus grand, mais nous observons aussi que l'erreur de l'ensemble de validation devient plus grand que l'erreur sur l'ensemble d'entraînement.
Le choix du paramètre k impacte directement la performance du classifieur.

Conclusion

Ce projet a permis d'explorer l'algorithme K-NN sous un double angle : théorique et empirique. Théoriquement, nous avons rappelé les garanties de performance remarquables de K-NN, notamment le résultat fondamental de Cover et Hart [1] qui borne son erreur à deux fois celle du classifieur de Bayes optimal dans le cas binaire. Ces fondements théoriques expliquent l'attrait conceptuel de l'algorithme.

Cependant, nos analyses ont révélé le fossé entre théorie et pratique. La **malédiction de la dimensionalité** se manifeste par deux phénomènes critiques : la **rareté des données** (nécessitant un nombre exponentiel de points d'entraînement) et la **concentration des distances** (rendant la notion de voisinage peu discriminante en haute dimension). Nos expériences empiriques ont validé ces limitations, montrant une dégradation rapide des performances au-delà de quelques dizaines de dimensions.

Paradoxalement, notre expérience sur MNIST a démontré que K-NN peut exceller même en dimension nominale élevée (784) lorsque les données résident sur une **variété de faible dimension**. Ce résultat clé justifie l'utilisation de techniques de réduction de dimension comme l'ACP avant d'appliquer K-NN.

Le choix du paramètre k illustre également le **compromis biais-variance** inhérent : un k petit favorise la flexibilité au risque du sur-apprentissage, tandis qu'un k grand favorise la stabilité au risque du sous-apprentissage.

En pratique, le K-NN ne doit pas être appliqué naïvement à des données de haute dimension sans prétraitement. Il gagne à être intégré dans des pipelines incluant :

- Réduction de dimension (ACP, t-SNE, UMAP)
- Sélection de caractéristiques pertinentes
- Apprentissage de métriques adaptées

En définitive, le K-NN illustre parfaitement le défi de concilier théorie élégante et contraintes pratiques. Son extrême simplicité, son interprétabilité et ses fondements théoriques solides en font un algorithme précieux, à condition d'être appliqué avec discernement et dans des conditions appropriées.

Références

1. T. Cover et P. Hart, *Nearest neighbor pattern classification*, IEEE Transactions on Information Theory, 13(1):21–27, 1967.
2. C. M. Bishop, *Pattern Recognition and Machine Learning*, Springer, 2006.
3. R. O. Duda, P. E. Hart et D. G. Stork, *Pattern Classification*, 2nd edition, Wiley, 2001.
4. J. Friedman, T. Hastie et R. Tibshirani, *The Elements of Statistical Learning*, Springer, 2001.
5. Y. LeCun et al., *MNIST handwritten digit database*, 1998.