Compte Rendu TP3 Apprentissage non-supervisé - Réduction de la dimensionnalité

Introduction

Ce TP explore les techniques de réduction de dimensionnalité (PCA et t-SNE) appliquées à la base de données MNIST des chiffres manuscrits. L'objectif est de comprendre l'impact de ces techniques sur la visualisation des données et sur les performances de deux algorithmes de classification : RandomForest et Softmax (régression logistique).

I - Affichage des données MNIST

1. Utilisation de la méthode PCA

Question 6 : Vérification de la nouvelle dimension

Oui, la nouvelle dimension des données réduites est bien de 2. Le résultat affiche : `Shape of reduced data: (10000, 2)`, ce qui signifie 10000 échantillons avec 2 composantes principales.

Question 8 : Le résultat d'affichage permet-il de donner une idée claire sur la distribution des instances de MNIST ?

Non, le résultat d'affichage avec PCA ne donne pas une idée très claire de la distribution. On observe un nuage de points assez compact où les différents chiffres se chevauchent énormément. Il est difficile de distinguer des groupes bien séparés pour chaque chiffre. La PCA réduit les dimensions en préservant la variance maximale, mais ne prend pas en compte la structure des groupes, ce qui explique cette visualisation peu lisible.

2. Utilisation de la méthode t-SNE

Question 12 : Existe-t-il des chevauchements entre quelques chiffres ? Si oui, lesquels ?

Oui, même avec t-SNE qui donne une bien meilleure séparation, on observe quelques zones de chevauchement entre certains chiffres qui se ressemblent visuellement :

- Les chiffres 4 et 9 (formes proches avec une boucle)
- Les chiffres 3 et 8 (courbes similaires)
- Les chiffres 7 et 1 (formes allongées)
- Parfois les chiffres 5 et 6 (courbes similaires)

Question 13 : Comparaison des résultats d'affichage PCA et t-SNE

La différence est très importante :

- PCA : Les points sont très mélangés, avec beaucoup de chevauchements. On ne distingue pas clairement 10 groupes séparés. La visualisation ne permet pas d'identifier facilement les différentes classes de chiffres.
- t-SNE : Formation de clusters bien distincts pour chaque chiffre. On observe clairement 10 groupes qui correspondent aux 10 chiffres (0 à 9). Les clusters sont bien séparés avec peu de chevauchement.

Conclusion : t-SNE est bien plus efficace que PCA pour visualiser des données complexes comme MNIST car elle préserve les relations de proximité locale entre les points, permettant de former des groupes cohérents.

II - PCA sur les données MNIST

1. Résultats de RandomForest SANS réduction de dimensionnalité

Résultats obtenus :

- Temps d'apprentissage : 62.07 secondes
- Taux de classification : 0.9705 (97.05%)

RandomForest obtient d'excellents résultats sur les données complètes avec 784 dimensions (28x28 pixels).

2. Résultats de RandomForest AVEC réduction de dimensionnalité

Réduction des dimensions :

- Dimensions originales : (60000, 784)

- Dimensions réduites : (60000, 154)
- Le PCA a réduit de 784 à 154 dimensions tout en conservant 95% de la variance

Résultats obtenus :

- **Temps d'apprentissage** : 192.85 secondes
- **Taux de classification** : 0.9488 (94.88%)

Question 20 : Le temps d'apprentissage est-il plus rapide que celui du II-1 ? Est-ce le résultat attendu ?

Non, c'est surprenant ! Le temps d'apprentissage est **plus lent** avec PCA (192.85 secondes contre 62.07 secondes). Ce n'est **pas** le résultat attendu.

Explication : Normalement, avec moins de dimensions, l'apprentissage devrait être plus rapide. Cependant, RandomForest fonctionne différemment des autres algorithmes. Il peut parfois être plus rapide sur des données brutes car il utilise des sous-ensembles aléatoires de features. La réduction PCA peut aussi ajouter du temps de calcul initial.

Question 22 : Comparaison du taux de classification

Le taux de classification a légèrement baissé : 97.05% sans PCA contre 94.88% avec PCA. On perd environ 2.17% de précision.

Question 23 : L'application du PCA était-elle fructueuse pour RandomForest ?

Non, l'application du PCA n'est **pas fructueuse** pour RandomForest dans ce cas :

- **Inconvénient majeur** : Le temps d'apprentissage est 3 fois plus long (192s vs 62s)
- **Inconvénient** : Perte de précision de 2.17%

Pour RandomForest sur MNIST, il vaut mieux utiliser les données complètes sans PCA.

3. Résultats de Softmax SANS réduction de dimensionnalité

Résultats obtenus :

- **Temps d'apprentissage** : 42.55 secondes
- **Taux de classification** : 0.9255 (92.55%)

Note : Un warning apparaît indiquant que l'algorithme n'a pas complètement convergé (limite d'itérations atteinte). Softmax est plus rapide que RandomForest mais moins précis.

4. Résultats de Softmax AVEC réduction de dimensionnalité

Résultats obtenus :

- **Temps d'apprentissage** : 16.51 secondes
- **Taux de classification** : 0.9201 (92.01%)

Question 28 : Le temps d'apprentissage est-il plus rapide que celui du II-3 ?

Oui ! Le temps d'apprentissage est beaucoup plus rapide : 16.51 secondes avec PCA contre 42.55 secondes sans PCA. On gagne **26 secondes**, soit une réduction de plus de 60% du temps.

Question 30 : Comparaison du taux de classification

Le taux de classification baisse légèrement : 92.55% sans PCA contre 92.01% avec PCA. La perte est minime (seulement 0.54%).

Question 31 : L'application du PCA était-elle fructueuse pour Softmax ?

Oui, l'application du PCA est **très fructueuse** pour Softmax :

- **Avantage majeur** : Réduction du temps d'apprentissage de 60% (de 42.55s à 16.51s)
- **Avantage** : Perte de précision négligeable (seulement 0.54%)

C'est un excellent compromis temps/précision. Pour Softmax, le PCA améliore considérablement l'efficacité sans sacrifier significativement la performance.

Conclusion générale

Question 32 : L'application du PCA contribue-t-elle toujours à accélérer le temps de calcul ?

Non, pas toujours. Les résultats montrent que cela dépend de l'algorithme :

- **Pour RandomForest** : Le PCA a **ralenti** l'apprentissage (de 62s à 192s). RandomForest gère efficacement les données de haute dimensionnalité grâce à sa structure d'arbres, donc le PCA n'apporte pas d'avantage ici.

- **Pour Softmax** : Le PCA a **accéléré** l'apprentissage (de 42s à 16s). Les algorithmes linéaires comme la régression logistique bénéficient beaucoup de la réduction de dimensionnalité.