Département de Mathémathiques et Informatique
Faculté des Sciences Exactes et Appliquées
Université de N'Djamèna

—

Simplonline.co
Tech4Tchad

Machine Learning
t-SNE : t-distributed Stochastic Neighor Embedding

HAMIDE Mahamat
HASSANE MOUSTAPHA Ousmane
Honoré PAYANG
MAHAMAT NIL Hassan
MBA LE IRAN Ezechiel

27 mai 2023

Table des matières

Table des figures

Introduction

Le Machine Learning ou apprentissage automatique est un domaine scientifique, et plus particulièrement une sous-catégorie de l'intelligence artificielle. Elle consiste à laisser des algorithmes découvrir des « patterns », à savoir des motifs récurrents, dans les ensembles de données. Ces données peuvent être des chiffres, des mots, des images, des statistiques, ...

Le t-Distributed Stochastic Neighbor Embedding (t-SNE) est un algorithme d'apprentissage automatique. C'est une technique non linéaire de réduction de dimensions pour la visualisation des données multidimensionnelles dans un espace de visualisation en deux ou trois dimensions. Cependant, comment fonctionne cet algorithme ? Quelles sont ces forces et limites ? Comment l'implémenter ? Ces intérogations seront répondues tout au long de ce document.

1 Algorithme

L'algorithme t-SNE se base sur la probabilité conditionnelle que deux points de données soient voisins dans l'espace d'origine et dans l'espace de projection. Cette probabilité est calculée en utilisant une distribution t-student. Ensuite, l'algorithme minimise la divergence de Kullback-Leibler [1] entre la distribution de probabilité conditionnelle dans l'espace d'origine et dans l'espace de projection.

Plus spécifiquement, l'algorithme t-SNE suit les étapes suivantes :

1. Calcul des distances entre les points de données dans l'espace d'origine.
2. Calcul des probabilités conditionnelles de similarité entre chaque point et ses voisins les plus proches dans l'espace d'origine.
3. Calcul des probabilités conditionnelles de similarité entre chaque point et ses voisins les plus proches dans l'espace de projection.
4. Minimisation de la divergence de Kullback-Leibler entre les deux distributions de probabilité conditionnelle à l'aide d'une méthode de descente de gradient stochastique.
5. Projection des points de données dans l'espace de projection final.

1. En théorie des probabilités et en théorie de l'information, la divergence de Kullback-Leibler (ou divergence K-L ou encore entropie relative) est une mesure de dissimilarité entre deux distributions de probabilités.

2 Avantages et Inconvéniants du t-SNE

2.1 Avantages du t-SNE

Le t-SNE convertit les affinités des points de données en probabilités. Les affinités dans l'espace original sont représentées par des probabilités conjointes gaussiennes et les affinités dans l'espace intégré sont représentées par des distributions t de Student. Cela permet au t-SNE d'être particulièrement sensible à la structure locale. L'utilisation de l'algorithme t-SNE présentes de nombreux d'autres avantages :

- Visualisation efficace des données grâce à la réduction dimentionnelle ;
- Conservation des relations entre les points : les points qui sont proches dans l'espace d'origine restent proches dans l'espace de projection. Cela permet de mieux comprendre la structure des données ;
- Capacité de gérer des données de types variés (textuelles, numériques, ...) ainsi que des données volumineuses ;
- Facilité d'utilisation : t-SNE est assez simple à utiliser et ne nécessite pas de paramètres complexes à régler.
- Performances élevées : t-SNE est capable de produire des projections de données de haute qualité en un temps raisonnable, ce qui le rend pratique pour une utilisation en temps réel.
- Révéler la structure à plusieurs échelles sur une seule carte

2.2 Inconvénients du t-SNE

Bien que l'algorithme t-SNE à de nombreux avantages, son usage présente également quelques inconvénients (qui sont grossiers) :

- Le coût de calcul est très élevé : pred beaucoup de temps et de ressources informatiques pour le traitement de grandes quantités de données ;
- Il est stochastique (et à un fonction de coût qui n'est pas convexe), c'est-à-dire qu'avec des initialisations différentes, on peut obtenir des résultats différents ;
- La structure globale des données n'est pas explicitement préservée. Ce problème est atténué en initialisant les points avec l'ACP (Analyse de la Composante Principale) en utilisant init='pca' ;
- Incapacté à traiter les données manquantes : si une variable de l'ensemble de données contient des valeurs manquantes, il est nécessaire de les remplacer par une valeur avant d'utiliser t-SNE.

3 Application de l'algorithme t-SNE

L'algorithme t-SNE est généralement utilisé pour explorer les données, comprendre les modèles sous-jacents et détecter des structures, des clusters ou des relations entre les données qui ne sont pas facilement perceptibles dans l'espace d'origine.

Les cas d'utilisation courants de l'algorithme t-SNE incluent :

1. Analyse de regroupement (clustering) : en réduisant la dimensionnalité des données, t-SNE peut faciliter l'analyse de regroupement en visualisant les clusters potentiels et en aidant à évaluer la cohérence des regroupements dans l'espace réduit.
2. Détection d'anomalies : en identifiant les structures inhabituelles ou les points isolés dans l'espace réduit, t-SNE peut aider à repérer les anomalies ou les points de données atypiques. Cela peut être utile dans des domaines tels que la détection de fraudes, la cybersécurité ou la surveillance des systèmes.
3. Prétraitement pour l'apprentissage automatique : t-SNE peut également être utilisé comme une étape de prétraitement avant d'appliquer d'autres techniques d'apprentissage automatique. En réduisant la dimensionnalité des données, il peut améliorer les performances des modèles en réduisant la complexité et en éliminant les redondances dans les données.

Voici quelque cas d'utilisation spécifiques :

— L'analyse de données de génomique, de protéomique et de transcriptomique pour visualiser les relations entre les différents gènes ou protéines : ceci est un exemple typique du cas d'ustilisation 1.
— L'analyse de données de vision par ordinateur pour visualiser les caractéristiques de différentes images.
— L'analyse de données textuelles pour visualiser les relations entre différents documents ou mots.
— La visualisation de données de réseaux sociaux pour comprendre les communautés et les relations entre différents utilisateurs.

Il convient également de noter que l'algorithme t-SNE n'est pas limité à ces cas d'utilisation peut être appliqué à différents types de données pour mieux comprendre les relations entre les variables.

4 Implémentation

L'implémentation du t-SNE est très facile en python avec la bibliothèque "scikit-learn". Voici la signature de la classe TSNE :
class sklearn.manifold.TSNE(*n_components=2, *, perplexity=30.0, early_exaggeration=12.0, learning_rate='auto', n_iter=1000, n_iter_without_progress=300, min_grad_norm=1e-07, metric='euclidean', metric_params=None, init='pca', verbose=0, random_state=None, method='barnes_hut', angle=0.5, n_jobs=None, square_distances='deprecated'*) [2]

```
In [1]: # Importer les bibliothèques nécessaires :
        import numpy as np
        import matplotlib.pyplot as plt
        from sklearn import datasets
        from sklearn.manifold import TSNE

In [2]: # Charger les données de Iris (à partir de la bibliothèque `sklearn.datasets`)
        iris = datasets.load_iris ()

        # Récuperer les données dans x, et les étiquettes dans y.
        x = iris.data
        y = iris.target

In [3]: # Réduire la dimensionnalité (à 2 dimensions) avec t-SNE : n_components = 2 par défaut
        tsne = TSNE ()

        # Ajuster x dans un espace intégré et récupérer la sortie transformée dans x_tsne
        x_tsne = tsne.fit_transform (x)

In [4]: # Visualiser les résultats
        plt.figure (figsize= (10, 6))
        colors = ['red', 'green', 'blue']
        for i in range (3):
            plt.scatter (x_tsne [y == i, 0], x_tsne [y == i, 1], c = colors [i], label = iris.target_names [i])
        plt.title ("t-SNE sur les données Iris")
        plt.legend (loc = 'best')
        plt.xlabel ("Dimension 1")
        plt.ylabel ("Dimension 2")
        plt.show ()
```

FIGURE 1 – Exemple d'implémentation de t-SNE

Pour exécuter et visualiser ce scritp, veillez cliquer ici.

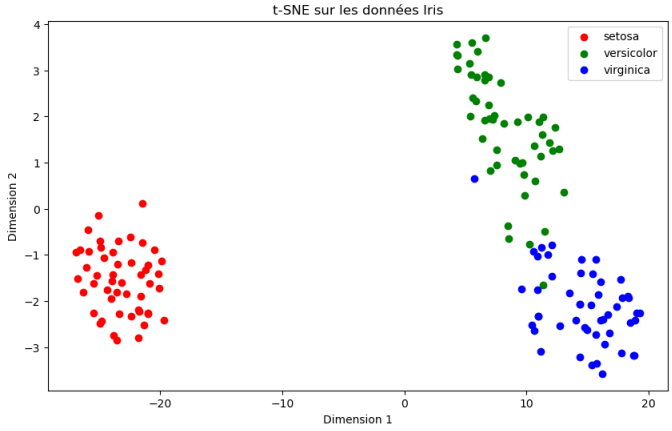

FIGURE 2 – Plot

2. Source

Conclusion

Bref, l'algorithme t-SNE repose sur le principe de regrouper des points similaires dans l'espace d'origine pour les projeter ensuite dans un espace de dimension inférieure. Il utilise une distribution de probabilité pour mesurer la similarité entre les points dans l'espace d'origine et dans l'espace de projection. Les points similaires ont une probabilité plus élevée d'être choisis comme voisins dans l'espace de projection. Cet algorithme a des limites dont l'Analyse de la Composante Principale (ACP) apporte des solutions.

Reférences

— Le notebook des modeules étudiés en salle
— La documentation officielle de scikit-learn.
— La documentation officielle de scikit-learn (t-SNE).
— Wikipédia