ALIGNEMENT DE DONNÉES HÉTÉROGÈNES BASÉ SUR LA MÉTHODE DE KERNELIZED SORTING

Ibrahim RIAZI

Parcours IAAA

RESUMÉ

Ce TER se concentre sur l'alignement de données hétérogènes en utilisant une méthode basée sur le **"Kernelized Sorting"**, une approche puissante pour aligner des ensembles d'éléments en comparant leurs distributions. Nous avons développé une implémentation de cette méthode avec une interface de visualisation intuitive. Pour cela, nous avons utilisé deux types de noyaux, le noyau Gaussien et le noyau basé sur le produit de noyaux, pour mesurer la similarité entre les éléments des ensembles. En ajustant la valeur de σ du noyau, nous avons obtenu des valeurs de similarité appropriées dans la matrice de similarité. Nous avons utilisé **"l'algorithme hongrois"** comme solution du problème d'appariement linéaire pour trouver la permutation optimale pour l'alignement des données. Nous avons appliqué notre méthode à un ensemble de 128 images représentant le mot **"AMU"** et nous avons observé des résultats satisfaisants, où les images similaires étaient regroupées à des emplacements proximaux dans la grille.

Mots clés— Alignement de données, Kernelized Sorting, Similarité, Noyau gaussien, Produit de noyaux, Interface de visualisation, Appariement linéaire, Problème d'appariement linéaire, Images, Grille.

1. INTRODUCTION

L'alignement de données hétérogènes est un enjeu majeur dans de nombreux domaines, tels que la bio-informatique, la vision par ordinateur, traitement automatique des langues ou encore la reconnaissance de la parole. Les méthodes d'alignement sont utilisées pour identifier les correspondances entre les éléments de deux ensembles de données ou plus. Elles nous permettent ainsi de mieux comprendre les relations entre ces ensembles.

Notre objectif dans ce TER est de proposer une implémentation de la méthode d'alignement proposée dans l'article **"Kernelized Sorting"**[1], présentée lors de la conférence **NIPS** en 2008. Cette méthode consiste à ajuster des données hétérogènes en essayant d'aligner les distributions et en comparant les éléments homogènes entre eux de chaque côté.

TER M1 encadré par Francois xavier DUPE.

Nous proposons également une interface de visualisation pour cette méthode qui permet à l'utilisateur de mieux comprendre les résultats de l'alignement. Cette interface de visualisation sera basée sur l'utilisation de petites images pour former un mot.

Ce rapport de projet présente donc notre approche pour mettre en œuvre la méthode d'alignement proposée et son interface de visualisation associée. Nous procédons également à une évaluation de notre méthode en utilisant des exemples de données hétérogènes, tout en discutant des limites de cette méthode ainsi que des perspectives pour des travaux futurs.

2. KERNELIZED SORTING

La méthode d'alignement proposée par **"Kernelized Sorting"**[1] vise à résoudre le problème de l'alignement de données hétérogènes en se basant sur des techniques de comparaison et d'alignement de distributions. L'idée principale est de trouver des correspondances entre les éléments de deux ensembles de données en alignant leurs distributions respectives.

L'algorithme de **"Kernelized Sorting"** est basé sur l'utilisation de noyaux **kernels** pour comparer les distributions des ensembles de données. Les noyaux sont utilisés pour mesurer la similarité entre deux éléments ou ensembles de données. Ils sont particulièrement adaptés pour capturer les structures non linéaires et les relations complexes entre les données.

La méthode commence par la construction d'une matrice de similarité entre les ensembles de données à aligner. Cette matrice est calculée en utilisant les noyaux appropriés pour mesurer la similarité entre les éléments des ensembles. Ensuite, une fonction de coût basée sur cette matrice est définie pour évaluer la qualité de l'alignement.

L'objectif de l'algorithme est de trouver une permutation optimale des éléments dans chaque ensemble de données, de manière à minimiser la fonction de coût. Cela revient à trouver l'alignement qui maximise la similarité entre les distributions des ensembles.

Pour résoudre ce problème d'optimisation, l'algorithme de **"Kernelized Sorting"** utilise des techniques d'optimisation et de recherche exhaustive pour explorer l'espace des permutations possibles et trouver la permutation optimale.

L'avantage de cette méthode est qu'elle permet d'aligner des données hétérogènes sans avoir besoin de connaître a priori les correspondances exactes entre les éléments des ensembles. De plus, en utilisant des noyaux, elle peut capturer des relations complexes et non linéaires entre les données, ce qui en fait une méthode flexible et puissante pour l'alignement de données.

2.1. Construction de la matrice de similarité

La matrice de similarité est une matrice qui mesure la similarité entre chaque paire d'éléments des ensembles à l'aide de noyaux appropriés. On peut utiliser différents types de noyaux pour calculer cette matrice, en fonction des caractéristiques des données. Par exemple, le noyau linéaire peut être utilisé pour des données linéairement séparables, tandis que le noyau gaussien peut être utilisé pour des données non linéaires.

Noyaux

Pour la construction de la matrice de similarité, nous avons essayé deux types de noyaux pour mesurer la similarité entre les éléments des ensembles. Le premier noyau utilisé était le noyau **Gaussien**, qui est défini par l'équation suivante :

$$k_1(x, x') = \exp\left(\frac{-||x - x'||^2}{2\sigma}\right) \tag{1}$$

Dans cette équation, le paramètre sigma σ contrôle la largeur de la distribution Gaussienne. Une valeur plus grande de sigma augmente la similarité entre les éléments, tandis qu'une valeur plus petite la réduit.

Le deuxième noyau que nous avons utilisé est basé sur une approche de produit de noyaux. Ce noyau est défini par l'équation suivante :

$$k_2(S1, S2) = \frac{1}{\sqrt{|S1|}\sqrt{|S2|}} \sum_{s\in S1} \sum_{s'\in S2} \exp\left(\frac{\|s-s'\|^2}{2\sigma}\right) \tag{2}$$

Dans cette équation, les ensembles de données $S1$ et $S2$ sont comparés en utilisant une combinaison des distances entre les paires d'éléments. Le paramètre sigma σ contrôle également la largeur de la distribution Gaussienne dans ce noyau.

Cependant, pour assurer une comparaison équitable entre les ensembles, il est important de normaliser ce noyau. Nous utilisons la fonction de normalisation suivante :

$$k_{\text{norm}}(x, y) = \frac{k_2(x, y)}{\sqrt{k_2(x, x) \cdot k_2(y, y)}} \tag{3}$$

Cette fonction de normalisation divise le noyau non normalisé par la racine carrée du produit des noyaux entre chaque ensemble et lui-même.

Les valeurs de similarité obtenues à l'aide de ces deux noyaux représentent la mesure de similitude entre deux éléments. Les valeurs de similarité sont généralement comprises entre 0 et 1, où une valeur plus proche de 1 indique une forte similarité entre les éléments, tandis qu'une valeur proche de 0 indique une faible similarité.

2.2. La permutation optimale

Cette étape consiste à trouver la permutation optimale des éléments dans chaque ensemble de données. L'objectif est de minimiser la différence entre les distributions alignées en trouvant la meilleure correspondance entre ces éléments des ensembles. Cela peut être formulé comme un problème d'optimisation combinatoire, où différentes méthodes de résolution peuvent être appliquées, telles que la programmation dynamique ou les algorithmes de recherche exhaustive.

La méthode proposée dans l'article **"Kernelized Sorting"**[1] pour trouver la permutation optimale repose sur le problème d'appariement linéaire[2]. L'objectif est de trouver une matrice de permutation π qui maximise la similarité entre les éléments des ensembles à aligner.

L'algorithme commence par initialiser une matrice de permutation initiale π_0. Ensuite, à chaque itération i, la matrice de permutation π_{i} est mise à jour en utilisant l'équation suivante :

$$\pi_{i+1} = (1 - \lambda)\pi_i + \lambda \cdot \text{argmax}_{\pi\in P_m}[\text{tr}(K\pi^T \cdot L\pi_i)] \tag{4}$$

Avec, K est la matrice de similarité calculée à partir des noyaux, L est la matrice de coûts qui capture les différences entre les éléments des ensembles calculée à partir des noyaux aussi, et P_m est l'ensemble de toutes les matrices de permutations π de m éléments. Le terme $\text{argmax}_{\pi\in P_m}[\text{tr}(K\pi^T \cdot L\pi_i)]$ représente le problème d'appariement linéaire, où nous cherchons la matrice de permutation π qui maximise la $trace$ de la multiplication des matrices $K\pi^T$ et $L\pi_i$.

Le facteur de pondération λ contrôle l'importance de la mise à jour de la permutation à chaque itération. Il permet de réguler la convergence vers la permutation optimale en combinant la permutation actuelle π_i avec la permutation obtenue en résolvant le problème d'appariement linéaire.

Lemme 1 *L'algorithme décrit en* (4) *pour* $\lambda = 1$ *se termine en un nombre fini d'étapes [1].*

Pour résoudre ce problème d'appariement linéaire $\text{argmax}_{\pi\in P_m}[\text{tr}(K\pi^T \cdot L\pi_i)]$, nous avons utilisé **"l'algorithme hongrois"**[3] également connu sous le nom d'algorithme de l'affectation optimale. L'algorithme hongrois est une méthode efficace pour résoudre des problèmes d'appariement linéaire en utilisant une approche de programmation dynamique.

En utilisant l'algorithme décrit en (4), nous parvenons à trouver la matrice de permutation optimale qui maximise la similarité et minimise les différences entre les éléments des ensembles à aligner. Cette matrice garantit un meilleur alignement global des données hétérogènes, facilitant ainsi l'analyse et la comparaison des éléments entre les ensembles.

2.3. L'alignement des données

Une fois la permutation optimale trouvée, les données peuvent être alignées en utilisant cette correspondance entre les éléments des ensembles. Les éléments correspondants sont regroupés et alignés selon les critères définis par la méthode. Par exemple, si les ensembles de données contiennent des images, les images correspondantes peuvent être alignées pour former des mots ou d'autres structures visuelles.

3. APPLICATION

Notre algorithme d'alignement basé sur la méthode de **"Kernelized Sorting"** sera appliqué à un ensemble de 128 images dans le but de former le mot **"AMU"**. L'objectif est de trouver une permutation optimale qui aligne les images de manière à former le mot souhaité.

Pour ce faire, nous avons utilisé un ensemble de 128 images provenant de **"The CIFAR-10 and CIFAR-100 datasets"**[4]. Pour représenter cet ensemble d'images, nous avons extrait des histogrammes de couleurs qui capturent les caractéristiques distinctives de chaque image.

Ensuite, nous avons construit une grille correspondant à la matrice L de l'algorithme l'algorithme décrit en (4), qui représente les lettres du mot **"AMU"** et indique l'emplacement où les images doivent être arrangées. Chaque lettre est associée à une grille spécifique dans laquelle les images seront placées.

La valeur de σ du noyau a été ajustée afin d'obtenir des valeurs de similarité appropriées dans la matrice de similarité K de l'algorithme décrit en équation (4). L'objectif était de réduire l'échelle de la similarité mesurée entre les éléments des ensembles et d'augmenter les valeurs de similarité obtenues. Afin d'explorer différentes valeurs de σ, une analyse a été effectuée pour évaluer son impact sur la matrice de similarité K de notre ensemble d'images.

La courbe suivante (Figure 1) illustre l'effet de la valeur de σ sur les valeurs de la matrice de similarité K de notre ensemble d'images. Différentes valeurs de σ ont été testées, allant de faibles à élevées, et les valeurs correspondantes de similarité ont été enregistrées. L'objectif était de trouver un compromis qui permette d'obtenir des valeurs de similarité distinctes tout en évitant une échelle de similarité trop large. Cette étape était cruciale pour garantir une représentation adéquate des différences de similarité entre les éléments de notre ensemble.

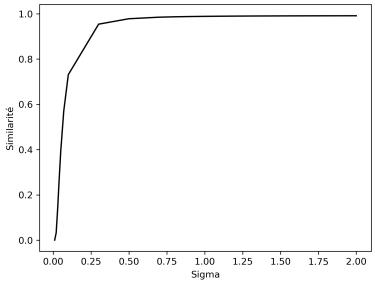

Fig. 1. Effet de Sigma sur la matrice de similarité

Nous avons expérimenté différentes valeurs de σ et calculer la distance moyenne entre toutes les paires d'images adjacentes en utilisant l'équation suivante :

$$Distance_{moyenne} = \frac{\sum_{i=1}^{N-1} \sum_{j=i+1}^{N} distance(I_i, I_j)}{N-1} \quad (5)$$

où :

- N est le nombre total d'images dans notre ensemble.
- $distance(\mathrm{I}_i,\mathrm{I}_j)$ est la fonction de calcul de la distance entre les images I_i et I_j (par exemple, distance euclidienne, distance de Manhattan, etc.).

Cette distance moyenne nous a permis d'évaluer la qualité de l'alignement en fonction de la valeur de sigma utilisée. Plus précisément, nous avons calculé la distance moyenne entre toutes les paires d'images adjacentes dans l'ensemble aligné, pour chaque valeur de σ testée.

La courbe suivante (Figure 2) illustre l'effet de la valeur de σ sur l'alignement de nos images. Pour chaque valeur de σ, nous avons mesuré la distance moyenne entre les images voisines après l'alignement. Une valeur plus faible de σ peut conduire à un alignement plus précis avec des distances moyennes plus faibles, tandis qu'une valeur plus élevée de σ peut entraîner un alignement moins précis avec des distances moyennes plus élevées.

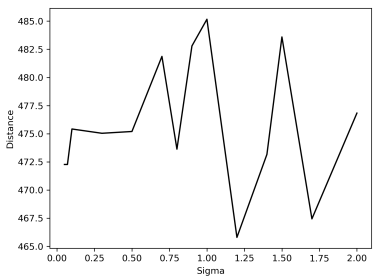

Fig. 2. L'impact du paramètre sigma sur l'alignement des images

Cette analyse nous a permis de déterminer la valeur de σ qui donne les meilleurs résultats en termes d'alignement des images. Nous avons cherché à trouver un équilibre entre une valeur de σ qui capture les différences de similarité entre les images et une valeur qui évite une échelle de similarité trop large, conduisant à un alignement moins précis.

Après le tri, nous affichons les images en fonction de leurs coordonnées correspondantes (Figure 3). Nous pouvons voir des images avec une composition de couleur similaire se trouvent à des emplacements proximaux.

Fig. 3. Affichage de 128 images alignées dans une grille de lettres **"AMU"** en utilisant *Kernelized Sorting*

4. ANALYSE CRITIQUE

Les résultats obtenus avec l'algorithme **"Kernelized Sorting"** sont prometteurs et démontrent son efficacité pour l'alignement des images dans le contexte spécifique de notre étude. En utilisant des noyaux définis par les équations (2) et (3) pour calculer la similarité entre les images, nous avons pu obtenir des valeurs de similarité appropriées dans la matrice de similarité K. Cependant, étant donné que l'algorithme repose sur une approche de produit de noyaux, il nécessite un peu plus de temps de calcul. En ajustant la valeur de σ, nous avons pu réduire l'échelle de similarité et améliorer les valeurs de similarité obtenues.

L'application de l'algorithme sur un ensemble de 128 images représentant les classes de CIFAR-10 a donné lieu à une grille alignée du mot **"AMU"**. L'alignement des images en fonction de leurs similarités a permis de regrouper les images présentant une composition de couleurs similaire à des emplacements proches dans la grille, démontrant ainsi la capacité de l'algorithme à capturer les similarités visuelles et à les organiser de manière cohérente. Toutefois, en ce qui concerne l'alignement en fonction des classes, les résultats n'ont pas été aussi efficaces que prévu.

Malgré les résultats positifs, il est important de souligner que l'algorithme **"Kernelized Sorting"** présente certaines limites. Il peut être sensible à la qualité des données en entrée, notamment à la diversité des images et à la présence de bruit. De plus, le temps de calcul de l'algorithme peut augmenter de manière significative avec un nombre croissant d'images, ce qui peut devenir un facteur limitant dans certaines situations. La courbe suivante (Figure 4) illustre l'impact du nombre d'images sur le temps d'exécution..

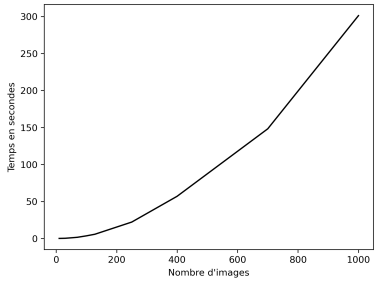

Fig. 4. La relation entre le nombre d'images et le temps d'exécution.

5. CONCLUSION ET PERSPECTIVES

En conclusion, notre étude propose une méthode d'alignement basée sur l'algorithme de **"Kernelized Sorting"** pour organiser un ensemble d'images selon une grille prédéfinie. Les résultats obtenus démontrent l'efficacité de notre approche pour l'alignement des images représentées par des histogrammes de couleurs. Cette méthode permet de regrouper les images similaires et de visualiser les motifs de composition de couleur à travers la grille, offrant ainsi une représentation cohérente et pratique des collections d'images. Cependant, des perspectives d'amélioration demeurent. Parmi celles-ci, l'extension de la méthode à d'autres modalités de représentation, et l'exploration des noyaux de similarité plus avancés permettant ainsi de capturer des relations plus complexes entre les éléments des ensembles. Ces perspectives ouvrent la voie à des recherches futures visant à améliorer la précision et la généralisation de notre méthode d'alignement pour une utilisation plus étendue dans divers domaines d'application.

6. REFERENCES

[1] Novi Quadrianto, Le Song, and Alex J. Smola, "Kernelized sorting," *Part of Advances in Neural Information Processing Systems 21 (NIPS 2008)*, vol. 2, pp. 4–6, 2008.

[2] "Assignment problem," `https://en.wikipedia.org/wiki/Assignment_problem`.

[3] "Algorithme hongrois," `https://fr.wikipedia.org/wiki/Algorithme_hongrois`.

[4] "The CIFAR-10 and CIFAR-100 datasets," `https://www.cs.toronto.edu/~kriz/cifar.html`.