ELM472 Makine Öğrenmesinin Temelleri

Ödev3

Ayşe Serra Şimşek
asimsek2019@gtu.edu.tr
[2]Elektronik Mühendisliği Bölümü, GTÜ, Kocaeli, Türkiye

I. GİRİŞ

Bu çalışmanın konusu, ABD'de bulunan Salinas Vadisi'nin, sensörler ile toplanan veri setlerinin haritasının k-means ile öbeklenip öbekleme hatasının hesaplanmasıdır. Bunun için iki farklı geliştirilen kod ile kıyaslama yapılması istenmiştir.

Salinas_gt.mat dosyası(ground truth) gerçek verilerin öbeklenmiş halini içermektedir. Veri seti 512 satırda 217 örnek olarak verilmiştir. Salinas_corrected.mat dosyası k-means ile öbeklenip ground truth dosyası ile karşılaştırılarak öbekleme hatası hesaplanmıştır. Analitik hesabın yanı sıra confusion matrix çizdirilerek karşılaştırma görsel olarak da yapılmıştır.

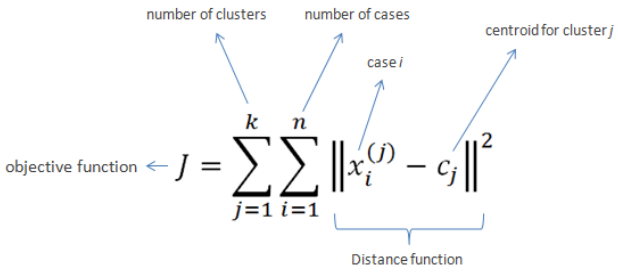
II. TEORİK BİLGİ

K-Means algoritması, kümeleme problemlerinde kullanılan ve veri noktalarını belirli bir sayıda küme veya gruplara bölen unsupervised bir makine öğrenimi algoritmasıdır. Başlangıçta rastgele seçilen küme merkezlerine dayanarak, her veri noktasını en yakın küme merkezine atayarak ve ardından bu merkezlerin yerini güncelleyerek çalışır. Bu atama ve güncelleme adımları, veri noktalarının benzerliklerine göre kümelere ayrılmasını sağlar. İteratif bir yaklaşımla devam eden algoritma, genellikle küme merkezlerinin değişmemesi veya belirli bir eşik değeri altına inmesi durumunda sonlanır, böylece veri kümeleme işlemi tamamlanır.

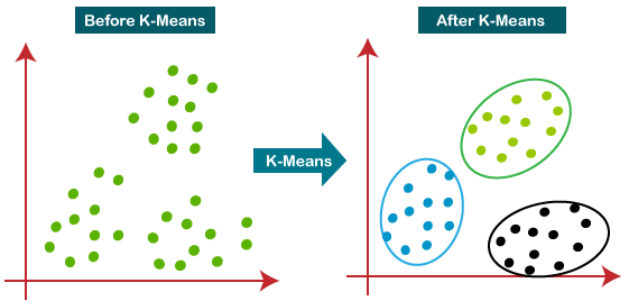
Algoritma, başlangıçta rastgele seçilen küme merkezlerine dayandığı için, başlangıç merkezlerinin seçimi algoritmanın sonuçları üzerinde etkili olabilir. İyi bir başlangıç noktası seçimi, algoritmanın daha hızlı ve daha istikrarlı bir şekilde sonlanmasına veya daha iyi kümeleme sonuçlarına yol açabilir.

K-Means algoritması genellikle sayısal ve doğrusal ölçeklenebilirlik gösteren verilerin kümeleme işlemlerinde tercih edilir. Özellikle, benzer ölçülerdeki veri setleri üzerinde etkili çalışır; örneğin, pazarlama verileri, müşteri segmentasyonu, finansal veriler ve bu çalışmada ele alındığı üzere coğrafi verilerin kümelendirilmesi gibi.

K-means kümelemesinin amacı, küme içi toplam varyansı veya kare hata fonksiyonunu en aza indirmektir:



$$J = \sum_{j=1}^{k} \sum_{i=1}^{n} \left\| x_i^{(j)} - c_j \right\|^2 \quad (1)$$



Şekil 1. K-means uygulaması

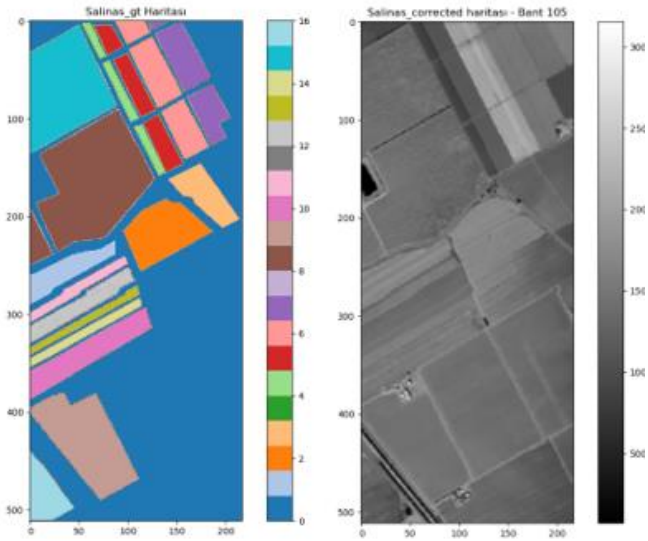
Algoritmanın aşamaları:

- İlk adımda, veri setinden belirli sayıda küme merkezi (centroid) rastgele seçilir. Bu küme merkezleri, başlangıçta veri setindeki noktalardan birkaçı olarak atanır.
- Her veri noktası, en yakın olan küme merkezine atanır.
- Atama aşamasından sonra, her küme için yeni bir merkez hesaplanır. Küme merkezleri, kümeye ait olan veri noktalarının geometrik ortalaması (ortalama konumu) olarak güncellenir.
- Atama ve yeniden hesaplama adımları tekrar edilir. Veri noktaları tekrar tekrar kümelere atanır ve küme merkezleri güncellenir.
- Genellikle, küme merkezlerinin konumu belirli bir eşik değerine veya sabitlendiğinde veya belirli bir iterasyon sayısına ulaştığında algoritma sonlanır.

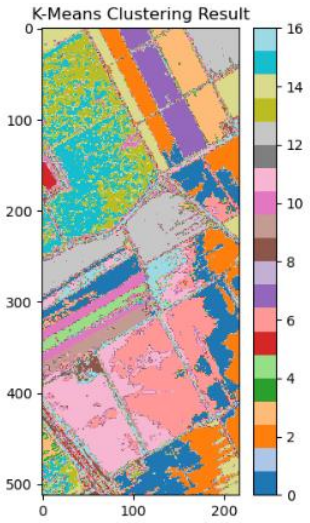
III. ÇALIŞMA

- ground_truth = loadmat('Salinas_gt.mat')['salinas_gt']: 'Salinas_gt.mat' dosyasından doğrulama verisi yüklenmiştir.
- plt.imshow(ground_truth, cmap='tab20', vmin=0, vmax=16), plt.colorbar(), plt.title("Salinas_gt Haritası"), plt.show(): 'ground_truth' verisini renkli bir harita olarak görselleştirilmiştir. Bu iki adım Salinas_corrected.mat dosyası için de gerçekleştirilmiştir.
- n_clusters = 17: Küme sayısı belirlenmiştir ve kmeans = KMeans(n_clusters=n_clusters, n_init=10, random_state=100): K-Means algoritması belirtilen küme sayısıyla başlatılmıştır.
- clusters = kmeans.fit_predict(reshaped_data): Veri seti üzerinde K-Means kümeleme algoritması çalıştırılır ve her örneğin hangi kümeye ait olduğu tahmin edilmiştir ve kümeleme sonuçları, orijinal görüntü şekline (cluster_labels) dönüştürülüp görselleştirilmiştir.
- Daha sonra, plot_confusion_matrix(): Oluşturulan karmaşıklık matrisi görselleştirilmiştir. Hata oranı hesaplanıp ve calculate_error_rate() ile ekrana yazdırılmıştır.
- Son olarak, Salinas_corrected.mat dosyasının kümelenmesi ve kümeleme hatasının bulunması, sklearn kütüphanesi kullanılarak ve kullanılmaksızın olmak üzere iki ayrı kodla denenmiştir.

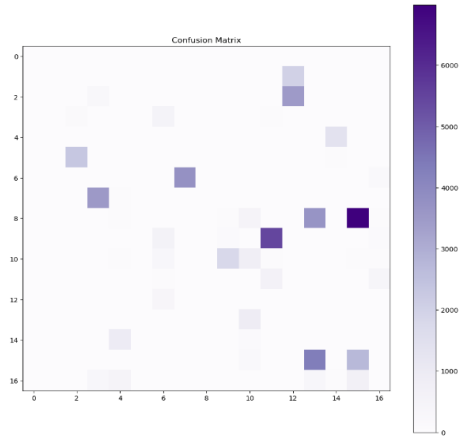
IV. SONUÇLAR



Şekil 2. Salinas_gt ve Salinas_corrected dosyaları



Şekil 3. Kümelenmiş Salinas_corrected dosyası



Şekil 4. Hata matrisi

- Confusion matrix ve hata hesaplama algoritması kullanılarak sklearn kütüphanesinin bulunduğu algoritmada kümeleme hatası 0.4789 olarak bulunmuştur. Kütüphanenin kullanılmadığı algoritmada ise hata oranı yaklaşık 0.4777'dir. Her iki kod da birbirine oldukça yakın değerler bulmuştur. Kodlar birden fazla kez çalıştırıldığında ± 2 gibi değişkenlikler gösterebilmektedir. Sebebi ise k-means algoritmasına göre başlangıç noktalarının rastgele seçilmesidir. Yarı yarıya başarılı bir kümeleme işleminin gerçekleştirildiği görülmektedir.
- Özellikle sklearn kütüphanesinin kullanılmadığı algoritmada, iyileştirmeler yapmak adına DBSCAN algoritması ve PCA indirgeme yöntemi denenmiştir. Ancak bu işlemlerin, kümeleme hatasını daha çok arttırdığı gözlenmiştir.

KAYNAKÇA

[1] Shaoguang Huang 1,Hongyan Zhang 2,Qian Du 3 andAleksandra Pižurica 4, "Sketch-Based Subspace Clustering of Hyperspectral Images"(2020)