Örneklemeli K-ortalama Algoritması
K-means with Sampling

Mehmet Fatih Amasyalı[1]

[1]Bilgisayar Mühendisliği Bölümü
Yıldız Teknik Üniversitesi
mfatih@ce.yildiz.edu.tr

Özet

K-ortalama algoritması, kümeleme problemlerinin çözümünde en çok kullanılan yöntemlerden biridir. Ancak K-ortalama'nın sonucu, ilk değer atamalarına bağlı olduğundan lokal minimumlara takılma problemi vardır. Optimizasyon literatüründe, lokal minimumlara takılma probleminin çözümü için rastgeleliğin kullanımı önemli bir yer tutmaktadır. Bu çalışmada K-ortalama algoritmasına bir örnekleme adımı eklenmiştir. Örnekleme yapmak arama uzayında küme merkezlerini yönlendirilmiş bir rastgelelikle hareket ettirmek olarak düşünülebilir. Bu hareketler sayesinde lokal minimumlardan kurtulmak mümkün olabilmektedir. Çalışmada ayrıca önerilen yöntemin orijinal algoritmadan daha iyi çalışabilmesi için gerekli olan bölgesel seçimleri yapma dinamikleri de incelenmiştir. Yapılan deneylerde önerilen algoritmanın, orijinalinden hem daha başarılı hem de daha hızlı olduğu görülmüştür.

Abstract

One of the most popular clustering algorithms is K-means. However, it can get stuck in a local minimum, because K-means result relies on its random initial points. In literature, randomness is often used to avoid local minimums. In this paper, a random sampling step is added to the K-means. Sampling can be thought as a directed random movement. These movements make it possible to avoid local minimums. Dynamics of local sampling (necessary for avoiding local minimums) is also investigated. Our experimental results showed that the proposed algorithm (K-means with sampling) is very fast and has more local minimum avoidance capability than the original K-means.

1. Giriş

Küme, birbirine bir şekilde benzer örneklerden oluşan topluluktur. Diğer bir deyişle bir kümedeki elemanlar birbirlerine başka bir kümenin elemanlarına göre daha çok benzerler. Bir firmanın elindeki müşteri bilgilerini kullanarak benzer müşteri profilleri oluşturması ya da bir resimdeki farklı renk sayısını azaltarak resmin sıkıştırılması verileri kümelemenin yaygın uygulamalarıdır. Bu uygulamalarda eldeki verilerin etiketleri / sınıfları yoktur. Bu sebeple bu tür verilerin kümelemesi eğiticisiz öğrenme olarak da anılır. Burada öğrenilen, kümelerin merkezleridir. Kümeleme algoritmaları temelde temsil edenle (küme merkezleri), temsil edilen (o kümedeki örnekler) arasındaki farkların ortalamasını (*H*) minimize etmeye çalışırlar. *H*'ın hesaplanması Eşitlik 1'de verilmiştir [1].

$$H = \frac{\sum_{t=1}^{N}\sum_{i=1}^{k} b_i^t \left\| x^t - m_i \right\|^2}{N} \quad (1)$$

$$b_i^t \leftarrow \begin{cases} 1 & if \left\| x^t - m_i \right\| = \min_j \left\| x^t - m_j \right\| \\ 0 & else \end{cases} \quad (2)$$

Eşitlik 1'de; *N*: örnek sayısını, *k*: küme sayısını, x^t : t.örneği, m_i : *i.* merkezi, b_i^t : *t.* örneği *i.* merkezin temsil edip etmediğini göstermektedir. *t.* örneğe en yakın merkez *i.* merkez ise b_i^t değeri 1, aksi durumda 0 olmaktadır.

Eşitlik 1'deki hata fonksiyonu sadece küme merkezlerinin değerine bağlıdır. Dolayısıyla kümeleme algoritmaları, küme merkezlerinin değerlerinin olduğu bir arama uzayın Eşitlik 1'in minimum değerini aramaktadırlar. Arama uzayının lokal minimumlara sahip olduğu durumlarda optimal sonuca erişilemeyebilmektedir.

Kümeleme algoritmalarında kaç adet küme merkezinin olacağını ya kullanıcı tarafından belirlenir ya da algoritma kendisi belirler. Küme sayısını otomatik belirleyen algoritmalarda kullanıcıdan bir eşik değeri beklemektedir. Dolayısıyla kümeleme algoritmalarında küme sayısını kullanıcı doğrudan ya da dolaylı olarak belirlemektedir.

Doğru küme etiketleri ya da doğru küme merkezleri elde olmadığından kümeleme algoritmalarının performanslarının birbiriyle karşılaştırılması sınıflandırma ya da regresyon problemlerinde olduğu kadar kolay değildir. Aynı sayıda küme üreten algoritmaların performanslarının karşılaştırılması mantıklıdır. Literatürde Eşitlik 1'deki formül, siluet genişliği [2], Davies-Bouldin Index [3] gibi çeşitli metrikler önerilmiştir.

K-ortalama algoritması basitliği ve kabul edilebilir performansı sebebiyle en yaygın kullanılan kümeleme algoritmasıdır. Toplu (Batch) ve tekli (online) olmak üzere

iki türü bulunan algoritmanın toplu hali Şekil 1'de verilmiştir. Tekli hali için [1] incelenebilir.

1-Küme merkezlerinin m_i {i=1..k} ilk değerlerini belirle

2-For a=1:maksimum iterasyon sayısı

For t=1:N

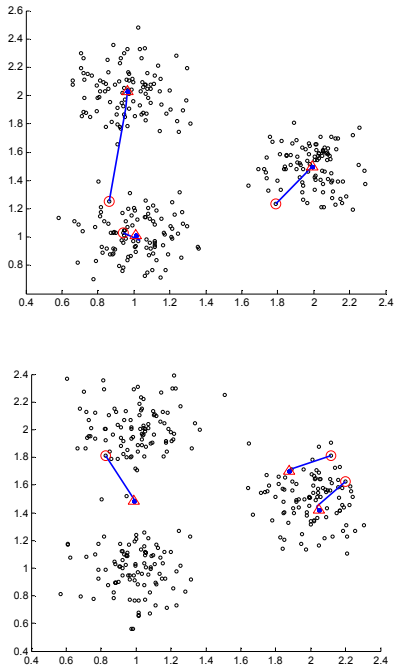
b_i^t leri Eşitlik 2'ye göre hesapla

For i=1:k

$$m_i = \frac{\sum_{t=1}^{N} b_i^t x^t}{\sum_{t=1}^{N} b_i^t}$$

Şekil 1: Toplu K-ortalama Algoritması.

Algoritma incelendiğinde küme merkezlerinin ilk değerleri atandıktan sonra herhangi bir rastgelelik içermediğinden sonuçtaki küme merkezlerinin değerlerinin deterministik olarak belirlendiği görülmektedir. İlk değerlerin farklı seçilmesi durumunda ise arama uzayının şekline göre farklı sonuçlara varılabilir. Şekil 2'de aynı dağılımdan üretilmiş 2 veri kümesi üzerinde farklı 2 ilk değerlerle erişilen sonuçlar görülmektedir. Şekillerde küme merkezlerinin başlangıç (o) ve bitiş (Δ) değerleri ve iterasyonlar boyunca aldıkları değerler (mavi çizgiler) gösterilmiştir.



Şekil 2: Farklı başlangıç değerleri için k-ortalama algoritmasının 2 çalışması

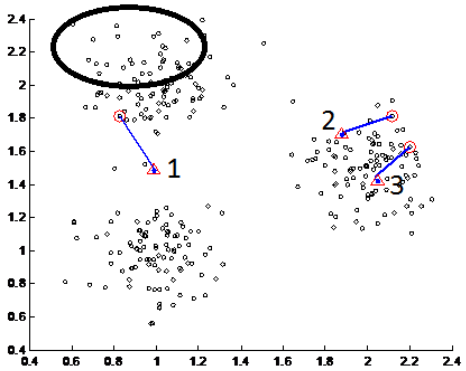
Şekil 2 incelendiğinde alttaki ilk değerlerin lokal bir minimuma takıldığı görülmektedir.

2. Lokal Minimumlardan Kurtulmak

Optimizasyon literatüründe, lokal minimumlardan kurtulmak için sisteme rastgelelik katılması en yaygın kullanılan yöntemlerdendir. Benzetimli Tavlama'da [4], olası güncellemelerden her zaman en iyisinin seçilmesi yerine belirli bir olasılıkla daha kötüsünün seçilmesine izin verilir. Genetik algoritmalardaki mutasyon işlemi de sisteme rastgelelik katma işlemidir [5]. Bagging [6] algoritması da, eğiticili öğrenmede, tüm verilerle eğitilen tek bir öğrenici yerine rastgele örnekleme ile seçilmiş alt veri kümeleriyle eğitilen çok sayıda öğrenicinin kararlarının birleştirilmesini önerir. Bu sayede karar sınırı bölgelerindeki marjin arttırılmakta ve test kümesi üzerinde daha başarılı sonuçlar alınabilmektedir.

Sisteme rastgelelik katmanın bu başarılı örnekleri sebebiyle, bu çalışmada, aynı mantığın (rastgele örnekleme yapmak) lokal minimumlara takılabilen K-ortalama algoritması için de bir iyileşme sağlama potansiyeli incelenmiştir.

Lokal minimumdan kurtulmak için ne yapmalı? Şekil 3'te küme merkezlerinin lokal minimumdan kurtulabilmesi için nasıl bir örneklemeye ihtiyaç olduğu gösterilmiştir.



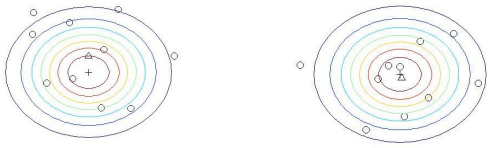
Şekil 3: Şekil 2(b)'deki lokal minimumdan kurtulmak için olası seçimlerden biri (siyah elipsin olduğu bölgeden yoğun bir seçilim, diğer bölgelerden seyrek bir seçilim)

Merkezlerin mevcut değerleri (Δ), Şekil 3'teki gibi olduğunda eğer iterasyonda tüm örnekler kullanılırsa küme merkezlerinde anlamlı bir değişim olmayacak 1 nolu merkez soldaki iki kümeyi birden temsil etmeye devam edecektir. Ancak siyah elips ile gösterilen bölgeden yoğun, diğer bölgelerden seyrek bir seçilim yapılırsa, 1 nolu küme merkezi seçimin yoğun yapıldığı bölgeye yukarıya doğru kayacaktır. Bu sayede sol alttaki kümenin örnekleri 2 ya da 3 nolu merkezlerle temsil edilebileceklerdir.

Şekil 3'teki seçilim incelendiğinde birörnek (uniform) bir seçilim olmadığı görülmektedir. Bu sebeple böylesi seçilim türüne bölgesel seçilim diyelim. Bölgesel seçilimin özelliği, küme merkezini hareket ettirebilmesidir. Özel seçilme olasılıkları vermeden, Bagging'de önerilen tamamen rastgele bir örnekleme böyle bölgesel seçimleri üretebilir mi? Üretme

kapasitesi nelere bağlıdır? Bu 2 soruya cevap vermek için Gaussian dağılıma sahip yapay bir veri kümesinde denemeler yapılmıştır.

Bölgesel seçilimin yapıldığının ölçütü (*S*), seçilen örneklerin ortalamasının, gerçek ortalamadan sapma büyüklüğüdür. Eğer büyük bir sapma olduysa, bölgesel seçilim yapılmıştır diyebiliriz. Şekil 4'te rastgele iki örnekleme sonucunda seçilen örnekler ve her birindeki sapma miktarları verilmiştir. Dağılımın ortalaması (+), seçilen örneklerin ortalaması (Δ), örnekler (o) ile gösterilmiştir.



Şekil 4: Aynı dağılımdan üretilmiş 2 rastgele örnek kümesi

Şekil 4'te soldaki seçilim, küme merkezini diğerine göre daha fazla hareket ettirecektir. Şekil 4'teki seçilimler, tamamen rastgele üretilmiştir. Buna göre 1. sorumuza cevap olarak, tamamen rastgele bir seçilimin bölgesel seçilimler üretebildiği söylenebilir. Rastgele seçilimin bölgesel seçilim üretme dinamikleri nelere bağlıdır? Bunun için, dağılımları üretmekte kullandığımız tüm parametreler incelenmelidir.

Dağılımdaki örnek sayısının sonsuz olduğu durumda, üretilen örneklerin ortalaması, dağılımın gerçek ortalamasının (μ) aynısı olacaktır (Eşitlik 3).

$$\mu = \lim_{N\to\infty} \frac{1}{N} \sum_{t=1}^{N} x^t \qquad (3)$$

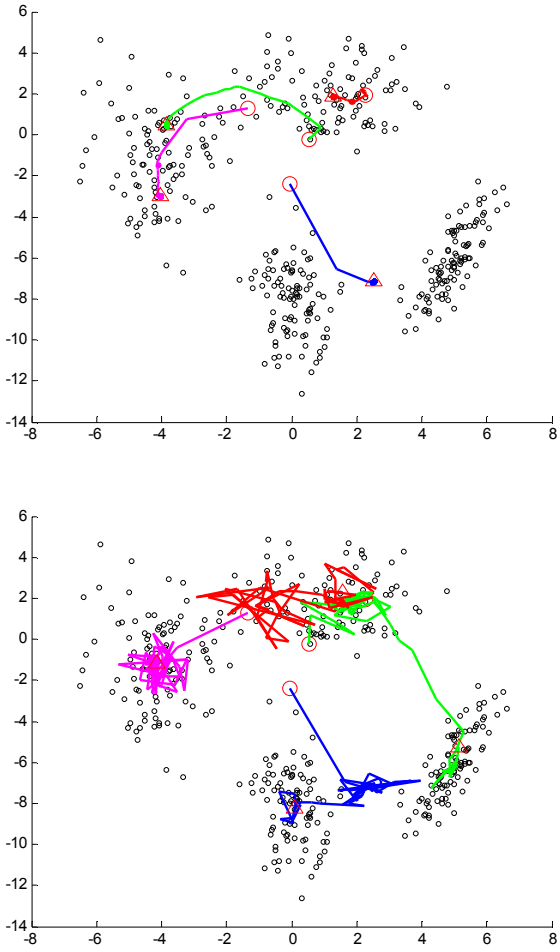
Eşitlik 3'te x^t, $D(\mu,\sigma^2)$ normal dağılımından üretilen örnekleri göstermektedir. Sonuç olarak az sayıda örnek varken bölgesel seçilim olma olasılığı daha yüksektir.

Rastgele örnekleme oranının (*SS*) etkisi, dağılımdaki örnek sayısıyla aynıdır. Ne kadar az oranda örnekleme yapılırsa doğal olarak bölgesel seçilim olasılığı o kadar artacaktır. Özelliklerin standart sapması ne kadar fazla olursa örnekler o kadar dağınık olacak ve bölgesel seçilim olasılığı o kadar artacaktır. Özelliklerin korelasyonlarının fazla olması örneklerin rastgeleliğini azaltacak ve bölgesel seçilim olasılığını azaltacaktır. Dağılıma her yeni eklenen özellik, öncekilere dik bir sapma daha ekleyeceğinden toplam sapma (gerçek ortalamadan sapma) artacaktır. Diğer bir ifadeyle özellik sayısı arttıkça, bölgesel seçilim olma ihtimali de artacaktır. Küme (dağılım) sayısının artışı örneklerin standart sapmasını arttıracak ve dolayısıyla bölgesel seçilim ihtimalini arttıracaktır.

Elde edilen sonuçlar özetlenirse, bölgesel seçilim olasılığı dağılımdaki örnek sayısı, örnekleme oranı, özelliklerin korelasyonuyla ters, özelliklerin sayısı ve küme sayısıyla doğru orantılıdır.

3. Örneklemeli K-ortalama Algoritması

Verilen bir veri kümesinde örnek sayısı, özellik sayısı, özelliklerin korelasyonu, küme sayısı değiştirilemez. Ancak veri kümesi üzerinde örnekleme yapılabilir. Bu sebeple orijinal K-ortalama algoritmasına bir örnekleme adımı eklenmiş ve böylece lokal minimumlardan kurtulma ihtimali arttırılmıştır. Örnekleme işleminde, her iterasyonun başında, orijinal örneklerden SS oranında yerine koymalı bir seçilim yapılmaktadır. Kümelere ait örneklerin belirlenmesi ve küme merkezlerinin güncellenmesi orijinal örneklere göre değil, bu seçilen örneklere göre yapılmaktadır. Sadece son iterasyonda, küme merkezlerinin daha iyi belirlenmesi için örnekleme yapılmamaktadır. Şekil 5'te aynı veri kümesi ve aynı başlangıç değerleri için orijinal ve örneklemeli k-ortalama algoritmalarının eriştikleri sonuç küme değerleri verilmiştir.



Şekil 5: Aynı veri kümesi ve aynı başlangıç değerleri için orijinal (yukarıda) ve örneklemeli (aşağıda) k-ortalama algoritmalarının eriştikleri sonuç küme değerleri.

Şekil 5'teki veri kümesi her biri 100'er örnek içeren 4 normal dağılımdan oluşmaktadır. 50 iterasyon sonundaki değerler verilmiştir. Örnekleme oranı olarak 0.2 alınmıştır. Orijinal algoritma lokal bir minimuma takılmışken, örneklemeli k-ortalama kurtulmuştur. Bölgesel seçilimle, bu kurtuluşun ilgisi düşünüldüğünde şu sonuca ulaşılmıştır: Temsil etmemesi gerekenleri temsil eden küme merkezlerinin örneklerinde bölgesel seçilim olasılığı, iyi oluşmuş kümelerin örneklerine göre daha fazladır. Çünkü

örneklerinin standart sapması daha fazladır. Dolayısıyla, kötü küme merkezlerinin, iyilere göre kayma olasılığı daha fazladır ki bu da lokal minimumlardan kurtulmanın anahtarıdır.

Literatürde, yaklaşımımıza en yakın çalışma, Li ve arkadaşları [7] tarafından yapılan çalışmadır. Bu çalışmada, K-ortalama algoritması Bagging ile birlikte kullanarak (örnekleme oranı = 1, topluluk boyutu = küme sayısı) bir kümeleyici topluluğu oluşturmuşlardır. 5 veri kümesi üzerinde orijinal k-ortalamadan daha iyi sonuçlar üretmişler ancak işlem zamanı olarak topluluğun boyutu oranında bir artışa sebep olmuşlardır. Başarılı sonuçlar elde etmelerine rağmen, algoritma küme sayısı kez K-ortalama algoritmasının çalıştırılmasına ihtiyaç duyduğundan hesaplama karmaşıklığı yüksektir.

Bizim çalışmamızda ise, çok düşük bir oranda örnekleme yapıldığından, orijinal K-ortalama algoritmasından bile daha hızlı çalışmaktadır.

4. Deneysel Sonuçlar

Bu bölümde "K-ortalamanın lokal minimuma takılma olasılığı nedir? Bu olasılığın bağlı olduğu parametreler nelerdir? Örneklemeli K-ortalama, orijinal K-ortalamaya göre ne kadar iyileşme sağlıyordur?" sorularına cevap bulmak için deneyler tasarlanmıştır. Soruların cevapları düşünüldüğünde şu hipotezlere ulaşılmıştır: Küme sayısıyla, lokal minimuma takılma ihtimalinin doğrudan ilişkili olduğu görülecektir. Çünkü ne kadar çok lokal minimum varsa o kadar çok bunlara takılma ihtimali olacaktır. Dağılımlardaki örnek sayısı ne kadar çok olursa, o dağılımın algılanma olasılığı o kadar fazladır. Bu nedenle örnek sayısı azalınca, lokal minimumlara takılma olasılığı artacaktır. Dağılımlardaki özellik sayısı da aynı şekilde düşünülebilir. Eşit örnek sayısına sahip dağılımlarda ne kadar çok özellik olursa dağılımın algılanma olasılığı o kadar düşecek ve lokal minimumlara takılma ihtimali artacaktır.

Bu hipotezlerin doğrulanması için çeşitli sayılarda Gaussian dağılım içeren veri kümeleri üretilmiştir. Dağılımların 2 ve 3 özelliklileri için kovaryans matrisi olarak sırasıyla [0.025 0; 0 0.025] ve [0.025 0 0; 0 0.025 0; 0 0 0.025] matrisleri kullanılmıştır. Bu veri kümelerinde küme merkezlerinin 100 farklı başlangıç değerinin her biri için 50 iterasyonluk denemeler yapılmıştır. Her iki algoritmanın da aynı ilk küme merkezi değerlerinden başlaması sağlanmıştır. Her iki algoritmanın da sonuçta eriştiği hatalar bulunmuş ve ortalamaları alınmıştır. Ölü bir merkez oluşma durumunda her iki algoritmada da, ölü merkez, orijinal örneklerden rastgele seçilen birinin değerine eşitlenmiştir.

Çizelge 1'de çeşitli kriterlere göre orijinal algoritma ve önerilen algoritmanın hata değerlerinin ortalamaları ve standart sapmaları "ortalama hata(standart sapma)" formatında verilmiştir. Hata ölçümü Eşitlik 1'de verilen *H* ile hesaplanmıştır. Örnekleme oranı (*SS*), 5 no'lu deneyde (örnek sayısı çok az olduğundan) 0.95, diğer tüm deneylerde 0.1 olarak kullanılmıştır. SS, 0.1 olarak seçildiğinde her adımda toplam örnek sayısının %10'u işleme katıldığından, örneklemeli k-ortalama, orijinal k-ortalamaya göre yaklaşık 10 kat daha hızlı çalışmaktadır.

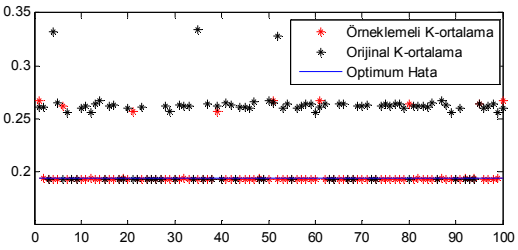
Çizelge 1: Çeşitli Kriterlere Göre Algoritmaların Performans Karşılaştırması

Deney no	İncelenen kriter	Kriter değeri	Orijinal K-ortalama	Örneklemeli K-ortalama	İyileşme oranı (%)
1	**Küme sayısı**	9	0.2409 (0.0362)	0.1999 (0.0202)	19
2		16	0.2592 (0.0337)	0.213 (0.0202)	19
3	**İlk değer belirleme yöntemi**	Orijinalden seç	0.2592 (0.0337)	0.213 (0.0202)	19
4		Rastgele seç.	0.2553 (0.0303)	0.2086 (0.0195)	19
5	**Küme başına örnek sayısı**	10	0.2596 (0.0321)	0.1945 (0.0219)	18
6		100	0.2592 (0.0337)	0.213 (0.0202)	20
7	**Özellik sayısı**	2	0.2592 (0.0337)	0.213 (0.0202)	20
8		3	0.2882 (0.0392)	0.2526 (0.0123)	25

Tablo 6'da, 1 ve 2 no'lu deneylerde, küme merkezlerinin ilk değer belirleme yöntemi olarak orijinal örneklerden seçilim, küme başına örnek sayısı 100, özellik sayısı 2 olarak belirlenmiştir. 3 ve 4 no'lu deneylerde, küme sayısı 16, küme başına örnek sayısı 100, özellik sayısı 2 olarak belirlenmiştir. 5 ve 6 no'lu deneylerde, küme sayısı 16, küme merkezlerinin ilk değer belirleme yöntemi olarak orijinal örneklerden seçilim, özellik sayısı 2 olarak belirlenmiştir. 7 ve 8 no'lu deneylerde, küme sayısı özellik sayısı 2 için 16, özellik sayısı 3 için 8, küme merkezlerinin ilk değer belirleme yöntemi orijinal örneklerden seçilim, küme başına örnek sayısı 100 olarak belirlenmiştir.

Çizelge 1 incelendiğinde tüm durumlarda önerilen algoritmanın daha az hataya (daha başarılı) ve standart sapmaya (daha güvenilir) sahip olduğu görülmektedir. Tablonun son sütununda orijinal algoritmaya göre % kaç oranında iyileşme sağlandığı verilmiştir. Bu iyileşmenin istatistiksel olarak anlamlı olup olmadıklarının ölçümü için 1 ve 2, 3 ve 4, 5 ve 6, 7 ve 8 nolu deneyler arasında ikili t-test yapılmış, tüm testlerde %5 anlamlılık ölçüsünde önerilen algoritmanın orijinal algoritmadan daha iyi olduğu görülmüştür. Bu bölümün başında sunduğumuz hipotezlerimiz, deney sonuçlarıyla tutarlıdır. Optimal hata değerleri; 2, 3, 4, 6 ve 7. deneyler için 0.1961, 1. deney için 0.1937, 5. deney için 0.1765, 8. deney için 0.2504'tür. Optimal hata, örneklerin gerçekte ait oldukları küme merkezine olan uzaklıkları kullanılarak hesaplanmıştır.

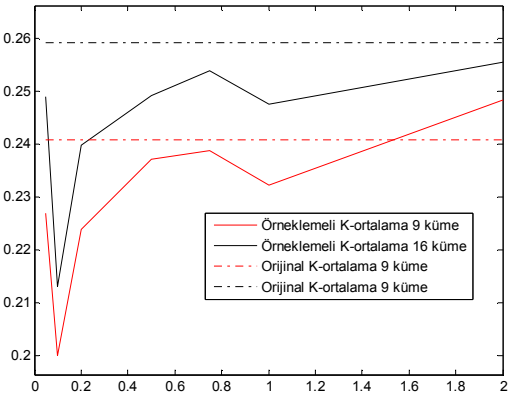
Elde edilen ortalama sonuçlara bakıldığında örneklemeli K-ortalamanın orijinal K-ortalamaya göre lokal minimumlardan daha kolay kaçabildiği görülmektedir. Algoritmaların denemelerin yüzde kaçında lokal minimum'a takıldığını görebilmek için Şekil 6'da 1.Deney'de elde edilen 100'er sonuç verilmiştir.



Şekil 6: Lokal minimuma takılma oranları

Şekil 6 incelendiğinde orijinal K-ortalamanın 100 denemenin 66'sında optimal hataya erişemediği (lokal bir minimuma takıldığı), örneklemeli K-ortalamanın ise 100 denemenin sadece 9'unda lokal bir minimuma takıldığı görülmektedir.

Algoritmanın üst-parametresi olan örnekleme oranının (*SS*), performansa etkisini incelemek için küme sayısı 9 ve 16, küme başına örnek sayısı 100, küme merkezlerinin ilk değer belirleme yöntemi olarak orijinal örneklerden seçilim, özellik sayısı 2 olarak belirlenerek 50 iterasyonluk, 100 farklı başlangıç değeriyle denemeler yapılmıştır. Şekil 7'de, farklı örnekleme oranları {0.05, 0.1, 0.2, 0.5, 0.75, 1, 2} için yapılan denemelerin ortalama değerleri verilmiştir.



Şekil 7: Örnekleme oranının (SS), hataya etkisi

Şekil 7 incelendiğinde, her 2 küme sayısı içinde, örnekleme oranının 0.1 (örneklerin rastgele %10'unun seçilmesi) olduğu durumda en iyi performansın elde edildiği görülmektedir. Örnekleme oranı arttıkça örneklemeli K-ortalamanın performansı, orijinal K-ortalamaya yaklaşmaktadır.

5. Sonuç

K-ortalama algoritmasının lokal minimumlara takılma problemine bir çözüm olarak, bu çalışmada orijinal algoritmaya bir örnekleme adımı eklenmiştir. Sonuçlara göre önerilen algoritma (örneklemeli K-ortalama) hem daha başarılı hem daha hızlıdır. Çalışmada bu avantajların dinamikleri de incelenmiştir.

Algoritmanın tek dezavantajı orijinal algoritmaya göre fazladan bir hiper-parametreye (örnekleme oranı) sahip olmasıdır.

Gelecek bir çalışma olarak, genel optimizasyon yöntemlerinde (tepe tırmanma, benzetimli tavlama, genetik algoritma vb.) o anki durum ya da durumların değerlendirilmesinde tüm örneklerin kullanılması yerine, bir alt kümesinin kullanımı düşünülebilir.

6. Kaynaklar

[1] Alpaydın, E., "*Introduction to Machine Learning*", The MIT Press, 2004..

[2] Rousseeuw P.J., "Silhouettes: a graphical aid to the interperation and validation of cluster analysis", *Journal of Computational and Applied Mathematics*, 20, 1987, pp. 53-65.

[3] Davies D.L., Bouldin D.W., "A cluster separation measure", *IEEE Trans. Pattern Anal. Machine Intell.*, 1979, pp.224-227.

[4] Van Laarhoven, Peter JM, ve Emile HL Aarts. "*Simulated annealing*". Springer Netherlands, 1987.

[5] Goldberg, D. E., "*Genetic algorithm in search. Optimization and Machine Learning*", 1989.

[6] Breiman, L., "Bagging predictors", *Machine Learning*, 24(2), 1996.

[7] Li, Hai-Guang, et al. "K-means clustering with bagging and mapreduce", *System Sciences (HICSS), 44th Hawaii International Conference on. IEEE*, 2011.