Fen Bilimleri Enstitüsü Dergisi, 22 (2010) 1-18
Marmara Üniversitesi

K-MEANS VE AŞIRI KÜRESEL C-MEANS ALGORITMALARI İLE BELGE MADENCİLİĞİ

Meltem IŞIK[1] ve Ali Yılmaz ÇAMURCU [2*]
[1] *Şişli Endüstri Meslek Lisesi, Bilgisayar Öğretmeni, Şişli, İstanbul*
[2] *Marmara Üniversitesi, Teknik Eğitim Fakültesi, Elektronik ve Bilgisayar Eğitimi Bölümü, 34722 Göztepe, İstanbul*

Alındığı Tarih: 11 Kasım 2008 *Kabul Tarihi: 14 Haziran 2010*

ÖZET: İnternetin gittikçe yaygınlaşması ve boyutlarının çok genişlemesi web sayfalarının büyük bir veri deposu haline gelmesine ve karmaşıklığının artmasına neden olmuştur. Bu nedenlerle web'de arama yapma ve kullanıcı profili çıkarma alanlarında veri madenciliğine ilgi artmıştır. Web sayfalarında bulunan belgeler içinde gerekli bilgiyi elde etmede kullanılan veri madenciliği yöntemlerinden birisi de belge madenciliğidir. Bu çalışmada, web belgesi içeren üç ayrı veri seti kullanılarak k-means ve aşırı küresel bulanık c-means algoritmalarının kümeleme başarıları karşılaştırılmalı olarak incelendi. Aşırı küresel bulanık c-means algoritmasının kümeleme başarısı, k-means algoritmasından daha iyi çıkmıştır.

Anahtar sözcükler: Veri Madenciliği, Belge Madenciliği, kümeleme, k-means, Aşırı Küresel Bulanık c-means

[*] Fax: (216) 337 8987 ve e-posta: camurcu@marmara.edu.tr

DOCUMENT CLUSTERING USING K-MEANS AND HYPERSPHERICAL FUZZY C-MEANS ALGORITHMS

ABSTRACT: Web pages have became a big data repository, with rapid grow in Internet. For these reason, interest to data mining in the field of searching in web pages and analyzing user profile is increased. Document mining is preferred to get necessary knowledge from documents on web pages. In this study, k-means and hyperspherical fuzzy c-means algorithms were applied to web documents and clustering performances were investigated comparatively using three data sets which have web documents. Our results show that clustering feature of hyperspherical fuzzy c-means algorithm is better than k-means algorithm.

Keywords: Data mining, Document mining, clustering, K-means, Hyperspherical Fuzzy c-means.

GİRİŞ

Kümeleme analizi, bir veri kümesindeki bilgileri belirli yakınlık ölçütlerine göre gruplara ayırma işlemidir. Kümeleme işleminde küme içindeki elemanların benzerliği fazla, kümeler arası benzerlik ise az olmalıdır. Kümeleme analizi, bireylerin ya da nesnelerin sınıflandırılmasını ayrıntılı bir şekilde açıklamak amacıyla geliştirilmiştir. Bu amaca yönelik olarak, ele alınan örnekte yer alan varlıklar aralarındaki benzerliklere göre gruplara ayrılır, daha sonra bu gruplara giren bireylerin görünüşü ortaya konur. Diğer bir hedef ise, benzer elemanların gruplanmasıyla veri setini küçültmektir[1-4].

İnternetteki web(örün) sayfaları boyutlarının gittikçe genişlemesi ve içeriğinin dinamik bir yapıya sahip olmasından dolayı, web sayfalarının otomatik olarak organize edilmesi ihtiyacı ortaya çıkmıştır. İnternet arama motorlarındaki ilerleme ile birlikte belge kümeleme analizine ilgi oldukça artmıştır[2]. Belge kümeleme analizinin amacı, bir belge içinde yer alan benzer belgeleri bulmaktır. İyi bir belge kümeleme analizinde, küme içindeki belgeler arasındaki benzerlik uzaklığı az, kümeler arası belgelerde de belge benzerliğinin büyük olması gerekir[2,3].

Bu çalışmadaki belge madenciliği, çok boyutlu vektörlerle temsil edilen web belgelerinin bulunduğu üç ayrı veri setinde uygulandı. Web belgelerinin kümelenmesi, Kosinüs Uzaklık ölçütü kullanılarak gerçekleştirildi. Çok boyutlu vektörler üzerinde işlemler yapmak için k-means ve aşırı küresel bulanık c-means algoritmaları üzerinde çeşitli değişiklikler yapılıp, bu algoritmaların da performansları değerlendirildi.

VERİ SETLERİ(DATA SETS)

Bu çalışmada, web belgesi kümeleme için Milliyet gazetesi, Hürriyet gazetesi ve YahooNews (İndirgenmiş) veri setleri kullanıldı. Milliyet gazetesi ve Hürriyet gazetesi İnternet arşivlerinden derlenen veri setleri [5] nolu çalışmadan alınmıştır. Milliyet gazetesi veri setinde ekonomi, siyaset ve spor olarak her biri 485'er tane html belgeri içeren üç alt başlık bulunmaktadır. Hürriyet gazetesi veri seti Astroloji, Bilim, Ekonomi ve Sporu içeren dört alt başlıktan oluşmaktadır. Bu veri setindeki Astroloji klasöründe 105, Bilim klasöründe 40, Ekonomi klasöründe de 176 tane ve Spor klasöründe de 127 tane belge bulunmaktadır. YahooNews indirgenmiş veri seti, içerikleri İngilizce html belgelerini içeren 4 alt başlık oluşmaktadır. Bu veri setinin, Business klasöründe 142 belge, Politics klasöründe 114 belge, Health klasöründe 164 belge, Sports klasöründe 141 belge bulunmaktadır.

BELGE VEKTÖRÜ YAPISI VE KOSİNÜS BENZERLİK ÖLÇÜTÜ

Belge kümelemede, web sayfaları içerdikleri kelimelerin normalize edilmiş frekans değerlerini tutan vektörlerle temsil edilir. Her belge, tüm webdeki kelimelerin sadece küçük bir oranını içermektedir. Belgelerin çok boyutlu birer vektör oldukları düşünüldüğünde, buradaki kümeleme problemi klasik kümelemeden daha farklı işlemler gerektirmektedir. Belge kümeleme verisi, büyük boyutlu, seyrek ve önemli derecede sıra dışı veri içeren bir yapıda olan kelime-belge matrisidir. Veri matrisinin satırları belgeleri, sütunları ise terimleri ifade etmektedir. Bu matris oluşturulurken her kelime belge çifti için terim

sıklığı-ters belge sıklığı olarak belirtilen TF-IDF (Term Frequency–Inverse Document Frequency) değeri hesaplanır [6]. Bu değer, o terimin belgedeki ağırlığını gösterir. Bir belgede, diğer belgelere göre daha sık görülen terim, o belgenin belirleyici terimidir. Bu nedenle ağırlığı yüksektir. Diğer taraftan, birçok belgede geçen terim belgeleri ayırt edici özelliğini yitirir ve terimin ağırlığı azalır.

TF ifadesi, terimin ilgili belgede kaç tane olduğunu gösterir. Böylece o terimin ilgili belge için önemini gösterir. TF değeri, Denklem 1 ile hesaplanır Burada *n* değeri *j*. kelimenin *i*. belgedeki sayısını, *d* değeri ise *i*. belgedeki bütün kelimelerin sayısını göstermektedir [6]:

$$\mathrm{TF}_{ij} = \frac{n_{ij}}{|d_i|} \tag{1}$$

IDF ters belge sıklığı Denklem 2 ile hesaplanır. Bu denklemde *n* toplam belge sayısını, nj ise j. terimin göründüğü belgelerin sayısını belirtir (sadece $\mathrm{TF}_{ij} > 0$ olan terimler için hesaplanır) [6]:

$$\log_2\left(\frac{n}{n_j}\right) \tag{2}$$

TF-IDF değerleri hesaplanarak veri matrisi oluşturulur. Fakat, bu matrisin böyle bir şekilde kullanılması çok büyük bir veri matrisi elde edileceğinden, bellek yeterli gelmeyecektir. Veri matrisinin sütunlarında bulunan herhangi bir kelime için, kelimenin bulunmadığı belgelerdeki TF değeri sıfır olacağından, TF*IDF değeri de sıfır olacaktır. Her belgede sınırlı sayıda terim olacağı düşünüldüğünde ortaya çıkan matrisin büyük bir kısmını "0" değeri dolduracaktır. Sıfırlar çıkarılarak veri matrisi indirgenir. Bu şekilde, sıfır değerleri için gereksiz bellek kullanımı engellenerek, bellek problemi çözülmüş olmaktadır. Benzerlik hesaplamaları gerçekleştirilirken işlem yapılacak belgenin satırı bir vektöre alınır. O belgede bulunmayan terimler için "0" değeri yerleştirilerek geçici bir süre olması gereken boyuta getirilir. Sırayla her

belge için işlemler tekrarlanır. Denklem 3 ve 4 ile matris içinde her terimin belgedeki ağırlıkları hesaplanır:

$$X_{ij} = TF_{ij} * IDF_j, \tag{3}$$

$$\frac{n_{ij}}{|d_i|} \times \log_2\left(\frac{n}{n_j}\right). \tag{4}$$

Belgeler arasındaki benzerlik hesabında değişik vektör aritmetik işlemleri kullanılabilir. Belge kümelemede, Öklid Uzaklığı, Kosinüs benzerliği, Pearson ilişkisi ve genişletilmiş Jaccard benzerliği gibi benzerlik ölçütlerini hesaplayan yöntemler vardır. Bu çalışmada belge kümelemede çok kullanılan vektör tabanlı bir ölçüt olan Kosinüs benzerliği kullanılmıştır. Kosinüs benzerliğinde, iki vektör arasındaki açının Kosinüs değeri hesaplanarak vektörlerin benzerliği bulunur. Kosinüs benzerliğinin güçlü bir özelliği vektör boyutundan etkilenmemesidir. Farklı sayıda kelimeler içeren benzer içerikteki belgeleri kolaylıkla tespit eder. Denklem 5'de görüldüğü gibi, vektörlerin skaler çarpımlarının, genliklerine bölünmesiyle iki vektör arasındaki açı elde edilir. İki vektör arasındaki açı ne kadar 0'a yaklaşırsa, açının Kosinüs değeri 1'e yaklaşır ve iki vektörün birbirlerine olan benzerlikleri de artar.

d ve d^* birbirinden farklı iki belgeyi temsil eden çok boyutlu vektörlerdir ve "•" vektörlerin iç çarpımını, $|d|$ ise vektörün uzunluğunu temsil etmektedir. İki vektör arasındaki açının Kosinüs değeri aşağıdaki formülle hesaplanır [4];

$$\cos(\theta) = \frac{d \bullet d^*}{|d||d^*|} = \frac{\sum_{i=1}^{n} d_i d_i^*}{\sqrt{\sum_{i=1}^{n} (d_i)^2}\sqrt{\sum_{i=1}^{n} (d_i^*)^2}} \tag{5}$$

K-MEANS ALGORİTMASI VE YAPILAN DEĞİŞİKLİKLER

K-means algoritması, merkez noktanın kümeyi temsil etmesi ana fikrine dayalı bir metottur [1]. Eşit büyüklükte küresel kümeleri bulmaya eğilimlidir[1]. K-means algoritmasının çalışma mekanizmasına göre öncelikle her biri bir kümenin merkezini veya ortalamasını temsil etmek üzere k tane nesne seçilir. Kalan diğer nesneler, kümelerin ortalama değerlerine olan uzaklıkları dikkate alınarak, en benzer oldukları kümelere dahil edilir. Daha sonra, her bir kümenin ortalama değeri hesaplanarak yeni küme merkezleri belirlenir ve tekrar nesne-merkez uzaklıkları incelenir.

K-means kümeleme yönteminin değerlendirilmesinde en yaygın olarak toplam karesel hata kriteri SSE (Summed Squared Error) kullanılır. En düşük SSE değerine sahip kümeleme sonucu, en iyi sonucu verir. Nesnelerin bulundukları kümenin merkez noktalarına olan uzaklıklarının karelerinin toplamı Denklem 6 ile hesaplanmaktadır [4,7].

$$\mathrm{SSE} = \sum_{i=1}^{K} \sum_{x \in C_i} dist^2(m_i, x) \tag{6}$$

Burada, *dist* iki nesne arasındaki standard Öklid Uzaklığı, x değeri C_i kümesinde bulunan bir nesne, m_i değeri C_i kümesinin merkez noktasıdır. Yukarıda açıklanan k-means algoritması, iki boyutlu veriler üzerinde Öklid Uzaklık ölçütüne göre çalışmaktadır ve hiçbir nesne kümesini terk etmeyene kadar ötelenmektedir. Ancak, web uygulaması için bu k-means algoritmasının yapısı uygun değildir. Her ötelemede kümeyi terk eden nesne olup olmadığını karşılaştırmak büyük veri setlerinde zaman açısından olumsuzluk yaratacağı için amaç fonksiyonu temelli bir k-means versiyonu tercih edilmiştir ve web sayfalarını kümelemek için bu algoritma çok boyutlu veriler üzerinde çalışır hale getirilmiştir. Öncelikle, verilerin tamamını belleğe alıp işlem yapmak mümkün olmadığı için her belgeyi temsil eden vektörün sırayla çağırılması sağlanmıştır. Bu vektörlerin küme merkezlerine uzaklığını farklı yöntemlerle hesaplamak için Kosinüs Benzerlik ölçütü eklenmiştir.

BULANIK C-MEANS ALGORİTMASI VE YAPILAN DEĞİŞİKLİKLER

Bulanık c-means (FCM) algoritması, bulanık bölünmeli kümeleme tekniklerinden en iyi bilinen ve yaygın kullanılan yöntemdir. Bulanık c-means metodu nesnelerin iki veya daha fazla kümeye ait olabilmesine izin verir[8]. Bulanık mantık prensibi gereği her veri, kümelerin her birine [0,1] arasında değişen birer üyelik değeri ile aittir. Bir verinin tüm sınıflara olan üyelik değerleri toplamı "1" olmalıdır. Nesne hangi küme merkezine yakın ise o kümeye ait olma üyeliği diğer kümelere ait olma üyeliğinden daha büyük olacaktır. Çoğu bulanık kümeleme algoritması amaç fonksiyon tabanlıdır. Amaç fonksiyonun belirlenen minimum ilerleme değerine yakınsaklaşmasıyla kümeleme işlemi tamamlanır. Temel olarak k-means algoritmasına çok benzemekle beraber bulanık c-means'in, k-means'den en önemli farkı verilerin her birinin sadece bir sınıfa dahil edilme zorunluluğunun olmamasıdır.

Bulanık c-means algoritması 1973 yılında Dunn tarafından ortaya atılmış ve 1981' de Bezdek tarafından geliştirilmiştir[9]. Bulanık c-means algoritması da amaç fonksiyonu temelli bir metottur. Algoritma, en küçük kareler yönteminin genellemesi olan aşağıdaki amaç fonksiyonunu öteleyerek minimize etmek için çalışır [10];

$$J_m = \sum_{i=1}^{N}\sum_{j=1}^{C} u_{ij}^{m} \left\| x_i - c_j \right\|^2, \qquad 1 \le m < \infty . \tag{7}$$

U üyelik matrisi rasgele atanarak algoritma başlatılır. İkinci adımda ise merkez vektörleri hesaplanır. Merkezler, Denklem 8 ile hesaplanır [10].

$$c_j = \frac{\sum_{i=1}^{N} u_{ij}^{m} x_i}{\sum_{i=1}^{N} u_{ij}^{m}} \tag{8}$$

Hesaplanan küme merkezlerine göre **U** matrisi Denklem 9 kullanılarak yeniden hesaplanır. Eski **U** matrisi ile yeni **U** matrisi karşılaştırılır ve fark ε'dan küçük olana kadar işlemler devam eder [10].

$$u_{ij} = \frac{1}{\sum_{k=1}^{C} \left(\frac{\|x_i - c_i\|}{\|x_i - c_k\|} \right)^{2/(m-1)}} \tag{9}$$

Kümeleme işlemi sonucunda bulanık değerler içeren **U** üyelik matrisi kümelemenin sonucunu yansıtır. İstenirse, berraklaştırma yapılarak bu değerler yuvarlanıp 0 ve 1'lere dönüştürülebilir. Yukarıda açıklanan bulanık c-means algoritması da k-means gibi iki boyutlu veriler üzerinde çalışmaktadır. Aynı şekilde bu algoritma da çok boyutlu veriler üzerinde çalışır hale getirilmiştir. Mendes ve Sacks tarafından geliştirilen aşırı küresel bulanık c-means algoritmasının formülleri kullanılarak, MATLAB'in bulanık(fuzzy) toolbox'ında bulunan Roger Jang'ın[8] hazırladığı bulanık c-means fonksiyonları web-belgesi kümeleme için uygun hale getirilmiştir. Aşırı küresel bulanık c-means normalize edilmiş belge vektörleri ile çalıştığı için tüm veri algoritmaya verilmeden önce normalize edilir. Aşırı küresel bulanık c-means formüllerinde Kosinüs benzerliği ölçüt olarak seçilmiş ve Denklem 10'da görülen benzersizlik ölçütüne çevrilmiştir [11].

$0 \leq S(X\alpha, X\beta) \leq 1$, V α, β -

$S(X\alpha, X\alpha) = 1$ V α -

Xi = [wi1 w i2 w i3……w ik]

Benzersizlik matrisi fonksiyonu: [12]

$$D(x_\alpha, x_\beta) = 1 - S(x_\alpha, x_\beta) = 1 - \sum_{j=1}^{k} w_{\alpha i} w_{\beta j} \tag{10}$$

Aşırı küresel bulanık c-means amaç fonksiyonu: [12]

$$J_m(U,V)=\sum_{i=1}^{N}\sum_{\alpha=1}^{c}u_{\alpha i}^{m}D_{\alpha i}=\sum_{i=1}^{N}\sum_{\alpha=1}^{c}u_{\alpha i}^{m}\left(1-\sum_{j=1}^{k}x_{ij}v_{\alpha j}\right). \tag{11}$$

Üyelik matrisi formülü: [12]

$$u_{\alpha i}=\sum_{\beta=1}^{c}\left(\frac{D_{\alpha i}}{D_{\beta i}}\right)^{-\frac{1}{m-1}}=\sum_{\beta=1}^{c}\left(\frac{1-\sum_{j=1}^{k}x_{ij}v_{\alpha j}}{1-\sum_{j=1}^{k}x_{ij}v_{\beta j}}\right)^{-\frac{1}{m-1}}. \tag{12}$$

Denklem 12 deki kısıdı gerçekleştirmek ve amaç fonksiyonu minimize etmek için küme merkezi formülüne Lagrange Çarpanı (multiplier) metodu uygulanmıştır[12]. Böylece Denklem 13'deki küme merkezleri formülü elde edilmiştir. Bu formülle elde edilen küme merkezleri normalize edilmiş vektörlerdir.

$$D(v_\alpha,v_\alpha)=1-\sum_{j=1}^{k}v_{\alpha j}\cdot v_{\alpha j}=1-\sum_{j=1}^{k}v_{\alpha j}^{2}=0,\quad \forall\alpha \tag{13}$$

$$v_\alpha=\sum_{i=1}^{N}u_{\alpha i}^{m}x_i\cdot\left[\sum_{j=1}^{k}\left(\sum_{i=1}^{N}u_{\alpha i}^{m}x_{ij}\right)^{2}\right]^{-1/2} \tag{14}$$

Denklem 14'de, w_i belge vektörünün bir boyutunu, v_i küme merkezini, N veri setindeki nesne sayısını, x_i α kümesindeki herhangi bir nesneyi, m bulanıklık katsayısını ifade etmektedir.

Orijinal bulanık c-means algoritması ve web belgelerini kümelemek için modifiye edilmiş aşırı küresel bulanık c-means algoritması belgeler arasındaki uzaklıkları hesaplamada ve küme merkezi bulunması konusunda farklılıklar göstermekle birlikte benzer şekilde çalışmaktadır.

KÜMELEMENİN DEĞERLENDİRİLMESİ

Kümelemenin değerlendirilmesi için kullanılan saflık (purity), entropi ve ortak bilgi (mutual information) ölçütleri kümelemenin sonucuna uygulanmıştır. Aynı klasörde bulunan belgeler aynı etiket numarasına sahiptir. Örneğin, Milliyet veri setindeki ekonomi klasörünün altında bulunan 485 adet belge "0" numaralı etikete, siyaset klasörünün altında bulunan 485 adet belge "1" numaralı etikete, spor klasöründe bulunan 485 adet belge ise "2" numaralı etikete sahiptir. Değerlendirme ölçütlerinin hesaplanabilmesi için kümeleme işlemlerinin sonucunda kümelerdeki belgelerin hangi etiket numaralarına sahip olduklarının bilinmesi gerekmektedir. Bu etiketler sayesinde hangi kümede hangi kategorilerden kaçar belge olduğu tespit edilir ve Tablo I' deki gibi bir karmaşıklık matrisi (confusion matrix) oluşturulur. Bu matris kullanılarak saflık, entropi ve ortak bilgi hesaplanır.

Saflık, küme elemanları içindeki baskın sınıfın kümedeki eleman sayısına oranını verir. Bir kümedeki elemanların hepsi aynı sınıfa aitse saflık maksimumdur. Denklem 15 ile hesaplanır [13].

$$\Phi^{(A)}(C_l) = \frac{1}{n_l} \max_h (n_l^{(h)}) \tag{15}$$

Burada, C_l her bir küme ve $n_l^{(h)}$, C_l kümesindeki her bir h kategorisine ait nesne sayısıdır.

Entropi, saflıktan daha kapsamlı bir ölçüttür. Saflık sadece baskın sınıfın içerisinde olan ve olmayan nesne sayılarıyla ilgilenirken, entropi tüm dağılımdaki düzensizlikle ilgilenir. Entropi, her bir sınıfa ait belgelerin bir küme içerisinde nasıl dağıldığına bakar. Kümenin içerdiği elemanlarının hepsi aynı sınıfa aitse entropi "0" olur. Denklem 16 ile hesaplanır [13].

$$\Phi^{(B)}(C_l) = -\sum_{h=1}^{g} \frac{n_l^{(h)}}{n_l} \log_g \left(\frac{n_l^{(h)}}{n_l} \right) \tag{16}$$

Saflık ve entropi büyük sayıda kümeleri değerlendirmek üzere kullanılmaktadır. Genel kümelemeyi değil de her bir kümenin kendi içindeki değerlendirilmesini yansıtırlar. Her küme tek bir belgeden oluştuğu zaman optimum değeri üretmektedirler. Bu nedenle kümelemenin genel başarısını gösteremezler.

Ortak Bilgi (Mutual Information): Teorik olarak en iyi sonuç veren nitelik ölçütüdür. Tarafsız bir değerdir. [0,1] arasında değerler almaktadır. Sınıflar dengeli olduğu durumlarda kümeleme başarılı ise 1'e doğru bir değer üretir. Saflık ve entropinin etkilendiği olumsuzluklardan etkilenmez. Denklem 17 ile hesaplanır [13].

$$\Phi^{(B)}(\lambda, K) = \frac{2}{n} \sum_{l=1}^{k} \sum_{h=1}^{g} n_l^{(h)} \frac{1}{\log(k \cdot g)} \left(\frac{n_l^{(h)} n}{\sum_{i=1}^{k} n_i^{(h)} \sum_{i=1}^{g} n_l^{(i)}} \right) \tag{17}$$

KÜMELEME ALGORİTMALARI TEST SONUÇLARI VE TARTIŞMA

Tablo II'de Milliyet gazetesi veri seti, Tablo III'de YahooNews (indirgenmiş) veri seti, Tablo IV'de Hürriyet gazetesi veri seti üzerinde k-means ve bulanık c-means algoritmaları için yapılan test sonuçları görülmektedir. Yapılan testlerde

Tablo I. Üç kümeye ayrılmış Milliyet veri setinin karmaşıklık matrisi

	Etiket		
Küme	1 (ekonomi)	2 (siyaset)	3 (spor)
1	460	3	15
2	10	17	462
3	15	465	8

iyi sonuç üretmesi ve işlemsel karmaşıklığının az olması nedeniyle Kosinüs benzersizliği kullanılmıştır.

Tablo II'de Milliyet veri seti kelimelerinin tamamı alınarak ve tohum değeri "7" verilerek uygulanan sonuçlar görülmektedir. %100 kelime için algoritma karşılaştırmasında k-means'in genel kümeleme başarısı 0.9955 bulanık c-means'in ise bir ("1") bulunmuştur. Her iki algoritma bu testte çok başarılı olmuştur. K-means 0.0045' lik çok küçük hata oranına sahiptir, ancak bulanık c-means kümeleri tamamen doğru ayırmıştır. Oluşan kümelerin saflığı ve entropisi incelenerek ayrı ayrı değerlendirmesi yapıldığında k-means'te sadece birinci kümenin farklı kategorilerden eleman içerdiği görülmektedir. Birinci kümenin saflığının 0.9955 ve entropisinin ise 0.0135 olması nedeniyle farklı kategorilere ait eleman sayısının oldukça az olduğu anlaşılmaktadır. İkinci ve üçüncü küme ise sadece tek bir kategoriye ait eleman içermektedir. Bulanık c-means'te ise bütün kümeler tamamen doğru ayrılmıştır. Kümelerin saflığının "1", entropisinin ise "0" olması her kümenin sadece tek bir kategoriye ait eleman içerdiğini göstermektedir. K-means algoritması'nın öteleme sayısı, bulanık c-means'e göre daha düşük çıkmıştır. K-means'in bulanık c-means'e göre daha çabuk yakınsaklaştığı görülmektedir. Hem öteleme sayısı hem de işlemsel karmaşıklığa bağlı olarak k-means'in geçen zaman değeri de bulanık c-means'den az çıkmıştır. Milliyet %100'lük veri seti sonuçları ortak bilgi, kümelerin saflığı ve entropisi açısından kıyaslandığında bulanık c-means algoritmasının k-means algoritmasından daha başarılı olduğu ancak k-means'in öteleme sayısı ve geçen zaman değerleri açısından daha avantajlı olduğu görülmektedir.

Milliyet veri setinin toplam kelimelerinin %50'si alınarak uygulanan sonuçlara göre kelime sayısında düşüş olmasına karşın, iki algoritmanın hem ortak bilgi hem de küme saflıkları ve entropileri, Milliyet %100'lük veri seti ile aynı çıkmıştır. Bulanık c-means'in öteleme sayısı biraz azalmıştır. Belgeleri temsil eden kelimelerin ayırt ediciliğine bağlı olarak algoritmanın

yakınsaklaşma süreci değişir. Kelime sayısındaki azalma nedeniyle geçen zaman değerleri de düşmüştür. Milliyet %50'lik veri seti sonuçları, %100'lük veri seti sonuçlarıyla benzer nitelikte çıkmıştır.

Tablo II. Milliyet Gazetesi veri setinde tohum 7 için algoritmaların karşılaştırılması

Veri Seti	L	Algoritma	Ortak Bilgi	Kümelerin Saflığı	Kümelerin Entropisi	Öteleme Sayısı	Geçen Zaman(sn)
Milliyet veri setinde	100	k-means	0.9955	0.9979 1 1	0.0135 0 0	6	24.7
100% kelime	100	Bulanık c-means	1	1 1 1	0 0 0	33	76.9
Milliyet veri setinde	50	k-means	0.9955	0.9979 1 1	0.0135 0 0	6	50
50% kelime	50	Bulanık c-means	1	1 1 1	0 0 0	30	50
Milliyet veri setinde	25	k-means	0.9473	0.9876 0.9815 0.9979	0.0610 0.0838 0.0135	9	17
25% kelime	25	Bulanık c-means	0.9605	0.9817 0.9937 1.0000	0.0833 0.0346 0	128	431

Milliyet veri setinin toplam kelimelerinin %25'i alınarak yapılan algoritma karşılaştırmasında k-means algoritmasının genel kümeleme başarısı 0.9473, bulanık c-means algoritmasının ise 0.9605 bulunmuştur. Her iki algoritmanın başarısında düşüş olmuştur, ancak bulanık c-means'nin başarısı daha az oranda düşmüştür. Her iki algoritma sonucunda oluşan kümeler farklı kategorilerden nesneler içermektedir. Ancak, bulanık c-means ile oluşturulan kümelerdeki farklı kategorilerden eleman sayısı k-means'le oluşan kümelerdekinden daha az çıkmıştır. Öteleme sayısı değerlerinin her iki algoritma için artması belgelerdeki

ayırt edici kelime sayısının oldukça azaldığını göstermektedir. Bu testte, bulanık c-means'in k-means'ten daha iyi sonuçlar üreterek daha başarılı kümeleme gerçekleştirdiği ancak öteleme sayısı ve geçen zaman açısından k-means'ten dezavantajlı olduğu görülmektedir.

Tablo III. Yahoo News veri setinde tohum 7 için algoritmaların karşılaştırılması

Veri Seti	L	Algoritma	Ortak Bilgi	Kümelerin Saflığı	Kümelerin Entropisi	Öteleme Sayısı	Geçen Zaman
YahooNews veri setinde %100 kelime	100	k-means	0.6581	0.5745 0.9464 0.9063 0.9776	0.6025 0.1753 0.2709 0.0773	8	40
	100	Bulanık c-means	0.8874	0.9145 0.9760 1.0000 0.9489	0.2500 0.0815 0 0.1455	33	315
YahooNews veri setinde %50 kelime	50	k-means	0.6581	0.5745 0.9464 0.9063 0.9776	0.6025 0.1753 0.2709 0.0773	8	40
	50	Bulanık c-means	0.8873	0.9145 0.9760 1.0000 0.9489	0.2500 0.0815 0 0.1455	33	311
YahooNews veri setinde %25 kelime	25	k-means	0.6884	0.7737 0.9896 0.7673 0.9683	0.5258 0.0418 0.4402 0.1015	13	55
	25	Bulanık c-means	0.8783	0.9279 0.9930 0.9758 0.9371	0.2160 0.0302 0.0944 0.1695	36	255

Tablo III' de YahooNews (indirgenmiş) veri seti üzerinde uygulanan algoritma test sonuçları görülmektedir. Bu sonuçlarda, k-means ve bulanık c-means arasındaki performans farkı daha belirgin olarak görülmektedir. Üç tablonun da sonuçları birbirine paralel çıkmıştır. K-means'in genel başarısı 0,65 gibi çok iyi sayılamayacak değerlerde çıkmıştır. Ayrıca bazı kümelerdeki farklı kategorilerden eleman sayısı oldukça fazla çıkmıştır. Bulanık c-means'in ise genel başarısı 0,88 gibi iyi değerlerde çıkmıştır. Ayrıca kümelerin saflığı oldukça iyidir, kümeler az oranda farklı kategorilerden elemanlar içermektedir. Daha önceki testlerde görüldüğü gibi k-means'in öteleme sayısı ve geçen zaman değeri bulanık c-means'ten düşüktür. Ancak ortak bilgi değeri öteleme sayısından ve geçen zamandan daha önemlidir. Genel kümeleme sonucunun başarılı olmadığı durumda öteleme sayısının ve geçen zamanın az olmasının bir önemi yoktur.

Tablo IV' de Hürriyet gazetesi veri seti üzerinde uygulanan algoritma test sonuçları görülmektedir. Bu sonuçlarda k-means ve bulanık c-means arasındaki performans farkı YahooNews veri setinde olduğu gibi net olarak görülmektedir. K-means'in genel başarısı 0,55 civarlarında iyi sayılamayacak değerlerde çıkmıştır. Ayrıca bazı kümelerdeki farklı kategorilere ait eleman sayısı oldukça fazladır ve kümeler iyi ayrılamamıştır. Bulanık c-means'in ise genel başarısı 0,77 civarlarında orta seviyede çıkmıştır. Ayrıca kümelerin saflığı, ikinci kümelerde 0.73 gibi biraz düşük seviyede çıkmıştır ancak diğer kümelerin saflığı oldukça iyidir ve kümeler az oranda farklı kategorilerden elemanlar içermektedir. Öteleme sayısı ve geçen zaman değeri önceki testlerle benzer çıkmıştır. Hürriyet veri setinde %25'lik değerler daha iyi çıkmıştır. Bu da rasgele seçilen kelimelerin ayırt ediciliğine ve sıra dışılık yaratan kelimelerin seçim içerisinde yer almamasına bağlı olarak değişiklik gösterir.

DEĞERLENDİRME

Gerçek veri seti olarak web belgeleri seçilmiştir. Web belgelerinin her biri içerdikleri kelimelerle ifade edildikleri için çok boyutlu vektörlerden oluşurlar.

Tablo IV. Hürriyet veri setinde tohum 13 için algoritmaların karşılaştırılması

Veri Seti	L	Algoritma	Ortak Bilgi	Kümelerin Saflığı	Kümelerin Entropisi	Öteleme Sayısı	Geçen Zaman
Hürriyet veri setinde 100% kelime	100	k-means	0.5403	1.0000 0.5730 0.8814 0.7849	0 0.7366 0.3139 0.4063	6	5.6
	100	Bulanık c-means	0.7639	0.9775 0.7410 1.0000 0.9362	0.0776 0.4599 0 0.1712	26	31
Hürriyet veri setinde 50% kelime	50	k-means	0.5432	1.0000 0.5730 0.8814 0.7935	0 0.7366 0.3139 0.3981	5	4.1
	50	Bulanık c-means	0.7666	0.9775 0.7357 1.0000 0.9462	0.0776 0.4638 0 0.1511	25	25.8
Hürriyet veri setinde 25% kelime	25	k-means	0.5610	1.0000 0.6071 0.8667 0.7629	0 0.6915 0.3373 0.4257	7	5
	25	Bulanık c-means	0.7829	0.9770 0.7305 1.0000 0.9681	0.0790 0.4440 0 0.1020	26	24

Her bir veri setinde yüzlerce belge ve her belgede de yüzlerce kelime olduğundan dolayı çeşitli bellek sorunlarıyla karşılaşıldı. Bir belgede geçen herhangi bir kelime birçok belgede bulunmadığından, diğer belgelerdeki ağırlığı sıfır olarak değerlendirilmektedir. Bellek problemini gidermek için bu sıfırlar indirgenmiştir. Ayrıca verilerin çok boyutlu olmasından dolayı algoritmaların bu veriler üzerinde işlem yapabilmesi için algoritmalarda da çeşitli değişiklikler

de yapılmıştır. K-means ve bulanık c-means algoritmasında ise Kosinüs benzerliği uygulanmıştır.

Test sonuçlarında da görüldüğü gibi bulanık c-means algoritması kümeleme işlemlerinde daha az hata oranına sebep olmuştur. Oluşan kümelerin saflıkları ve entropileri k-means'le oluşan kümelerin değerlerinden daha iyi çıkmıştır. Ayrıca daha kararlı sonuçlar üretmiştir. Ancak k-means algoritmasına göre işlemsel karmaşıklığı oldukça yüksek olduğundan daha fazla döngü gerçekleşmiş, ayrıca öteleme sayıları k-means'in öteleme sayılarından fazla çıkmış, bu nedenlerle geçen zaman değerleri k-means'ten yüksek çıkmıştır. Oluşan küme saflıklarında iyi sonuç elde edilmesi istenen çalışmalar için bulanık c-means algoritmasının kullanımı uygun olacaktır.

KAYNAKLAR

[1] Fayyad, U.M.; Piatetsky-Shapiro, G.; Smyth, R.; Uthurusamy, R.: "Advances in Knowledge Discovery and Data Mining", *AAAI/MIT Pres, CA*, **1996**.

[2] Han, J.; Kamber, M.: "Data Mining Concepts and Techniques", *Morgan Kauffmann Publishers Inc*., **2006**.

[3] Pang-Ning Tan, P.N.; Steinbach, M.; Kumar, V.: "Introduction to Data Mining", *Addison Wesley*, Mart **2006**.

[4] Jain, A.K.; Murty, M.N.; Flynn, P.J.: "Data Clustering: A Review", *ACM Computing Surveys*, Vol. 31, No 3, September **1999**.

[5] Gerçek veri setlerinin kaynağı : Işık, D.; Dolu, O.; Özbek, U.: "Web Sayfalarının Özelliklerini Elde Eden ve Web Sayfaları Benzerlik Ölçütlerini Karşılaştıran Uygulama", *Lisans Tezi*, İstanbul Teknik Üniversitesi, (**2006**)

[6] Robertson, S.E.; Jones, K. Sparck: "Simple, proven approachesto text retrieval", *Technical Report Number 356, Computer Laboratory, UCAM-CL-TR-356*, **1994**.

[7] Kaufman, L.; Rousseeuw, P. J.: "Finding Groups in Data: an Introduction to Cluster Analysis", *John Wiley and So*ns, 1990.

[8] Kruse, R.; Borgelt, C.; Nauck, D.: "Fuzzy Data Analysis: Challenges and Perspectives", *IEEE Int. Conf. on Fuzzy Systems 1999 (FUZZIEEE99), Seoul*, 1211-1216, **1999**.

[9] Höppner, F.; Klawonn, F.; Kruse, R.; Runkler, T.: "Fuzzy Cluster Analysis", *John Wiley&Sons, Chichester,* **2000**.

[10] Moertini, V.S.: "Introduction To Five Clustering Algorithms", *Integral,* Vol. 7, No. 2, Ekim 2002.

[11] Salem, S.A.; Nandi, A.K.: "New Assessment Criteria for Clustering Algorithms", *Proceedings of the IEEE International Workshop on Machine Learning for Signal Processing (MLSP-2005),* Mystic, CT, USA, (Eylül **2005**), 285-290.

[12] Mendes, M.E.S.; Sacks, L.: "A scalable hierarchical fuzzy clustering algorithm for text mining", *4th International Conference on Recent Advances in Soft Computing,* 269-274, **2004**.

[13] Strehl, A.; Ghosh, J.; Money, R.: "Impact of Similarity Measures on Web-page Clustering", *AAAI Workshop on AI for Web Search*, 58-64, **2000**.