KENDİ KENDİNİ DÜZENLEYEN HARİTALAR (SOM) İLE X-MEANS KÜMELEME METODUNUN KARŞILAŞTIRILMASI

Öğr. Gör. Serpil SEVİMLİ DENİZ

Van Yüzüncü Yıl Üniversitesi, Bilgisayar Programlama Bölümü, Van, Türkiye

sdeniz@yyu.edu.tr

Prof. Dr. H. Eray ÇELİK

Van Yüzüncü Yıl Üniversitesi, Ekonometri Bölümü, Van, Türkiye

herayçelik@yyu.edu.tr

ÖZET

Kümeleme analizi, verilerin belirli bir model üzerinden geçip farklı gruplara ayrılmasıdır. Bu gruplar küme olarak adlandırılır. Kümeleme işlemleri verilerin içerdiği nesnelerin özelliklerine göre yapılır. Kendi kendini düzenleyen haritalar (SOM) temelinde yapay sinir ağlarının kullanıldığı soyut matematiksel bir modeldir. Boyut azaltma ve veri kümeleme amacıyla denetimsiz öğrenme algoritması kullanılarak eğitilen bir tür yapay sinir ağıdır. Kümeleme problemlerine etkin bir çözüm sunmaktadır. Bölümleyici küme analizi algoritmalarından k-means, kümeleme konusunda popüler olmasına rağmen, yetersiz hesaplama yapması, küme sayısının kullanıcı tarafından tanımlanması ve aramada yerel minimuma eğilimli olması açısından üç temel eksikliğe sahiptir. X-means algoritması, k-means ın her çalışmasından sonra, mevcut merkezlerin hangi alt kümelerinin, verilere daha iyi uyacak şekilde bölünmesi gerektiği konusunda yerel kararlar verebilen, k sayısını belirtilen aralıkta bulabilen alternatif bir algoritmadır. X-means ile Bayes Bilgilendirme Ölçütünü (BIC) optimize ederek kümelenme yerlerinin ve kümelenmelerin sayısını verimli bir şekilde araştıran yeni bir algoritma önerilmiştir. Bu çalışmada WEKA (Waikato Environment for Knowledge Analysis) yardımıyla aynı veri setine SOM ve x-means kümeleme uygulaması yapılarak kümeleme algoritmaları karşılaştırılmıştır.

Anahtar Kelimeler: Kendi kendini düzenleyen haritalar, SOM, kümeleme, x-means,k- means

1-Giriş

Küme analizi, amacı nesneleri kümeler halinde sınıflandırmak olan bir yöntem grubunu temsil eder. Kümeleme çalışmalarında, klasik istatistiksel yöntemler yerine yapay sinir ağları kullanılabilmektedir. Çeşitli nöral ağ mimarileri ve öğrenme algoritmaları arasında, Kohonen'in haritası (SOM) en popüler sinir ağ modellerinden biridir. İlişkilendirilmiş bir bellek modeli için geliştirilen, denetlenmeyen bir öğrenmedir(Kohonen, 1982).

2-Kendi Kendini Düzenleyen Haritalar (SOM)

Kendi Kendini Düzenleyen Haritalar (SOM) uyarıcı ve cevaplar arasındaki içsel ilişkilerin, potansiyel olarak yanlı ya da öznel bir dış etki olmaksızın öğrenildiği, temel bir kalıp tanıma sürecidir. SOM, girişten çıkış alanlarına topolojik olarak korunmuş haritalama sağlayabilir. Bu ağlarda kullanılan öğrenme algoritması denetimsizdir. Yani, ağ eğitilirken bağımlı değişken kullanılmaz. Veri setindeki giriş vektörleri ağa girildikçe ağ kendi kendini düzenler ve referans vektörleri oluşur. SOM ağları, veri setindeki birimleri hem kümelendirebilir hem de görsel olarak haritalandırabilir. Bu sebeple SOM ağları, klasik istatistikteki k-ortalamalar ile çok boyutlu ölçekleme yöntemlerinin her ikisinin de işlevlerini yerine getirebilmektedir. SOM ağları, hem verilerin kümelenmesi, hem de görselleştirilmesi için tercih edilmektedirler (Zontul vd., 2004). SOM ağları, tek katmanlı bir ağ olup giriş ve çıkış nöronlarından oluşur. Giriş nöronlarının sayısını veri setindeki değişken sayısı belirler. Diğer yapay sinir ağlarından farklı olarak SOM ağlarında, çıkış katmanındaki nöronların dizilimi çok önemlidir. Bu dizilim doğrusal, dikdörtgensel, altıgen veya küp şeklinde olabilir. En çok dikdörtgensel ve altıgen şeklindeki dizilimler tercih edilmektedir. Pratikte, çoğu kez dikdörtgensel dizilim karesel dizilim olarak uygulanır. Buradaki dizilim topolojik komşuluk açısından önemlidir. Aslında, çıkış nöronları arasında doğrudan bir bağlantı yoktur. Giriş nöronları ile her bir çıkış nöronu arasındaki bağlantıyı referans vektörleri (code-book vectors) gösterir. Bu vektörler bir katsayılar

matrisinin sütunları olarak da düşünülebilir. SOM sinir ağları eğitilirken bu topolojik komşuluk referans vektörlerinin yenilenmesinde kullanılır (Zontul vd., 2004).

Kohonen ağında, giriş katmanına ek olarak, birbiriyle topolojik olarak ilişkili sinirlerden oluşan tek bir çıkış katmanı vardır. Her bir giriş, çıkış katmanındaki her bir sinire bağlıdır. Ağ rastgele ağırlıklarla çalışmaya başlar. Herhangi bir giriş uygulandığında, giriş vektörüne Öklid uzaklığı en az olan sinir seçilir ve bu sinire gelen bağlantı giriş ağırlıkları giriş vektörüne yaklaşacak şekilde yenilenir. Bu kazanan sinirle birlikte, onun topolojik komşuluğunda bulunan belli sayıda sinire gelen ağırlıklar da benzeri şekilde değiştirilir. İki boyutlu Kohonen ağında *i* siniri kazanır ise ağırlıklar;

$$w_i(t)+\eta(t)[X(t)-w_i(t)] \qquad i \in A_i(t)$$
$$w_i(t+1)=w_i(t) \qquad i \notin A_i(t)$$

olacak şekilde yenilenir. Burada x giriş vektörü, w_i, *i* sinirine gelen giriş ağırlık vektörü, η öğrenme hızıdır. $A_i(t)$ ise merkezi kazanan sinir olan komşuluk işlevidir ve *t* anında *i* sinirine komşu olan sinirler kümesini tanımlar. Kazanan sinirlerin ne büyüklükte bir komşuluktaki diğer sinirleri etkileyeceği zaman içerisinde değişiklik gösterir. Bu komşuluk başlangıçta büyük tutulup zaman içerisinde azalır. Böylece giriş vektörlerine tek sinirlerin değil sinir öbeklerinin tepki vereceği bir ön örnekleme sağlanmış olur. Topolojik olarak komşu olan sinirler birer çizgi ile birleştirilmiştir. Çıkış katmanında en sık kullanılan topolojiler bir ve iki boyutludur. SOM algoritması rekabete dayalı öğrenme yöntemi ile çalışır. Bu yöntemde ağa ait çıktı nöronları aktifleştirmek, yani kazanan nöron olmak için rekabet halindedir ve sonuçta her bir zaman diliminde sadece bir nöron kazanabilmektedir(Keim, 2002) . Girdi uzayındaki veri ile düşük boyutlu harita üzerinde yer alan her bir nöron arasında sinaptik bağlantılar oluşturulmakta ve bu süreç sonucunda, girdi uzayındaki örneğe (vektöre) en yakın olan ya da başka bir deyişle en fazla benzerliğe sahip olan çıktı düğümü aktifleşmektedir. Aktifleşen nöron Kazanan Nöron (Best-Matching Unit, BMU) olarak anılır. Ağın ilk olarak kurulmasından itibaren gerçekleşen süreç, üç ana süreçte incelenmektedir; rekabet, ortak çalışma ve sinaptik uyum(Haykin, 1999). Rekabet sürecinde, her bir girdi vektörüne ait kazanan düğümler (nöronlar) belirlenmekte, daha sonra ortak çalışma sürecinde komşu nöronlar ile ilişkilerin ele alınması açısından her bir aktif düğüme ait komşu düğümlerin SOM üzerindeki koordinatları belirlenmekte ve son olarak da sinaptik uyum sürecinde, komşu nöronların ağırlık vektörlerinde güncellemeler gerçekleştirilerek bir sonraki aşamada girdi kalıbı için daha uygun bir uygulama gerçekleştirilmektedir (Weiss ve Indurkhya, 1998).

m, girdi uzayının boyutunu temsil etmek üzere, girdi uzayından rastgele olarak bir *x* vektörü,

$x=[x_1, x_2, x_3..., x_m]^T$ seçilmiş olsun. Harita yapısında bulunan her bir nörona ait sinaptik-ağırlık vektörlerinin boyutları da girdi uzayının boyutu ile aynı olmak durumundadır. Örneğin, ağ üzerinden seçilmiş olan bir *j* nöronuna ait sinaptik-ağırlık vektörü şu şekilde tanımlanabilir;

$$w_j=[w_{j1}, w_{j2}, w_{j3}..., w_{jm}]^T, \qquad j=1,2,3,...,l$$

Eşitlikte belirtilen *l*, SOM örgüsü üzerinde bulunan toplam nöron sayısını belirtmektedir. *x* girdi vektörüne ait kazanan nöronun (BMU) belirlenmesi için tüm ağırlık vektörleri sırasıyla *x* vektörü ile iç çarpım (skaler çarpım) $(w_j^T x)$ işlemine tabi tutulurlar. İç çarpım sonucu en büyük değeri hangi ağırlık vektörü oluşturuyor ise, kazanan nöron olarak seçilmektedir. $(w_j^T x)$ iç çarpım değerinin en büyültülmesi, *x* vektörü ile (w_j) ağırlık vektörleri arası öklid uzaklığının en küçük değere yaklaşması anlamına gelmektedir. İki vektör arası öklid uzaklığı azaldıkça da, iki vektör arası benzerliğin aynı derece artması anlamına gelir. Burada, vektörler arası iç çarpım

uygulanmakta ve en yüksek iç çarpım değeri seçilerek girdi vektörü x 'e SOM harita üzerinde bulunan nöronlardan en yakın olanı, başka bir deyişle en benzer olan nöron seçilir(Haykin, 1999a).

Basit SOM Algoritması;

$x = [x_1, x_2, x_3..., x_m]^T$, m boyutlu girdi uzayında bir örnek ve j nöronuna ait ağırlık vektörü de, $w_j = [w_{j1}, w_{j2}, w_{j3}..., x_{jm}]^T$ olmak üzere;

Bir boyutlu uzayda tanımlı olan SOM için algoritma şu şekildedir:

Adım 1: İlk atamaların yapılması Sinaptik-ağırlık vektörlerine ilk değerleri ata.

Öğrenme katsayısını $\alpha(t)$ ata. Komşuluk derecesini(K) ve komşuluk fonksiyonunu $(h_{j,i})$ ata.

Adım 2: Öklid uzaklığını kullanarak girdi örneği ile her nöron arası uzaklığı hesapla.

$$d(j) = \sum_{i=1}^{N} (w_{i,j} - x_i)$$

Adım 3: Girdi verisine en yakın nöronu (BMU) bul.

(Adım 2'de en küçük değere sahip j indisi bu değere sahiptir.)

Adım 4: Verilmiş olan parametrelere göre vektör güncelleştirmesini,

$$w_j(k+1) = w_j(k) + \eta(n) h_{j,i(x)}(k)(x(k) - w_j(k))$$

göre yap.

Adım 5: Her girdi verisi için Adım 2-4 ü gerçekleştir.

Adım 6: Öğrenme katsayısını güncelle.

Adım 7: Topolojik komşuluk katsayısını güncelle.

Adım 8: Çalışmanın sonlandırılmasını kontrol et.

Adım 9: Sonlandırma olmadığı sürece Adım 2-8 gerçekleştir.

İlk Değerlerin Atanması

Eğitim sürecinin öncesinde, prototip vektörlere ilk değerlerin atanması gerekir. Bu işlem gerçekleştirilmesi için üç farklı yoldan birisi tercih edilebilir;

Rastgele Değer Atama (Random Initialization): Ağırlık vektörleri, rassal olarak küçük değerlerle yüklenir.

Girdi Referanslı Değer Atama (Sample Initialization): Ağırlık vektörleri, girdi verilerden rassal olarak çekilen değerlerle yüklenir.

Doğrusal Değer Atama (Linear Initialization): Ağırlık vektörleri, girdi verisine ait en büyük iki özdeğere (eigenvalue) karşılık gelen iki özvektörün (eigenvector) yine girdi uzayına dağıtılması sonucu yüklenir(Schatzmann,2003).

3- X-Means Algoritması

K-means, verileri sayısı araştırıcı tarafından belirlenen kümelere ayırmakta kullanılan bir algoritmadır. Veriler kümeler arasındaki değişkenliğin maksimum, kümeler içi değişkenliğin minimum olması amaçlanarak bölütlere ayrılır (D. Pelleg, and A. Moore, 2000)

k-means kümeleme yönteminin algoritması şu biçimdedir:

1. Ayırmak istedigimiz k küme sayısı belirlenir.

2. k tane veri başlangıçta küme merkezi olması amaçlanarak rastgele veya özel belirlenir.

3. Küme merkezi olmayan verilerin belirlenen uzaklık ölçülerine göre küme merkezlerine uzaklıkları hesaplanır.

4. Her veri en yakın olduğu küme merkezine atanır.

5. Yeni küme merkezleri, k adet baslangıç kümesindeki değişkenlerin ortalamaları alınarak oluşturulur.

6. Veriler belirlenen uzaklık ölçülerine göre en yakın oldukları oluşturulan yeni küme merkezlerine atanırlar.

7. Verilerin, yeni oluşturulan küme merkezlerine olan uzaklıkları öncekilerle kıyaslanır.

8. Uzaklık uygun oranda azalmış ise 6. adıma dönülür.

9. Eğer çok büyük bir değişiklik yok ise iterasyon sona erdirilir (Bilen 2004; Martinez ve Martinez 2005).

x-means sınırsız bir Gaussian EM algoritmasında bir model araştırmasını yönlendirmek için BIC'nin uygulanması olarak tanımlanabilir. Çok büyük veri setlerinde çok büyük ölçekli gözlemler kullanarak yapılan çalışmalara yardımcı olacak geniş bir algoritma sınıfı için bir fırsat sunmaktadır. k-means algoritması geliştirilerek, x-means ile model seçimi için yeni bir k-means tabanlı algoritma sunulmaktadır. İstatistiksel temelli kriterler kullanan bu model üzerinde yapılan uygulamalar k-means'tan hızlı ve daha iyi performanslı olduğunu göstermektedir. X-means; küme sayısını kendi belirlemektedir. Verileri analiz ederek min/max küme sayısını belirleyebilir. Dictance metric ve veri yapısı kendisine özeldir. Nominal veri alamaz(D. Pelleg, and A. Moore, 1999).

4. Uygulama

Uygulamada literatürde çok bilinen Iris Plants Database verileri WEKA (Waikato Environment for Knowledge Analysis) yardımıyla SOM ve X-Means kümeleme algoritmalarıyla kümelenmiştir. Küme sayısı 2,3,4,5 ve 6 olarak denenerek SOM ve x-means algoritmalarının performansları çizelge 1 deki gibi gösterilmiştir. Yapılan denemelerden sonra her iki algoritmada kümeleme başarımları benzerlik gösterse de x-means algoritması verileri kümelemede SOM algoritması kadar hassas davranamamıştır. En büyük fark ise algoritmaların çalışma zamanları ve adımlar(iterasyon) arasındadır. Çalışmalar sonucunda x-means algoritması genel olarak aslında 3 küme olduğunu zaten bildiğimiz iris flower veri setinde yapılan çalışmada 3 özelliği 0,01 sn de kümelemiştir. Toplam adım(iterasyon) sayısı ise 1 dir. 3 küme için BIC Değeri 65.87 dir. Küme sayısı arttıkça BIC değeri düşmektedir.

SOM kümeleme algoritması ise daha uzun sürede sonuca ulaşmaktadır. Aynı veriler için yapılan çalışmalarda küme sayısının 3 seçilmesiyle beraber süre olarak 2 sn de kümeleme gerçekleşmiştir.

Çizelge 1. SOM ve X-means algoritmalarının performansları

	x-means		***SOM***	
Küme Sayısı	Kümelenmiş örnekler	Süre	Kümelenmiş örnekler	Süre
2	67% 33%	0.01	65% 35%	1.43
3	35% 32% 33%	0.01	33% 33% 34%	2
4	35% 32% 16% 17%	0.01	19% 28% 19% 33%	2.55
5	20% 18% 16% 17% 29%	0.01	33% 3% 19% 27% 18%	3.13
6	18% 7% 16% 17% 17% 24%	0.01	17% 14% 19% 17% 15% 19%	3.67

5-Sonuç

Bölümlemeli kümeleme algoritmaları giriş parametresini alarak n tane nesneyi k tane kümeye böler. Nesneler birbirlerine benziyorlarsa ve başka kümelerdeki nesnelere benzemiyorlarsa, aynı kümeye alınır. Bölümlemeli kümeleme algoritmalarında yöntemler hem uygulanabilirliğinin kolay olması hem de verimli olması nedeniyle iyi sonuçlar üretir (Işık, 2006). Çalışmada zaten 3 küme olduğu bilinen bir veri setinin gruplara ayırılması SOM ve X-Means algoritmaları kullanılarak gerçekleştirildi. Her iki algoritma için küme sayısı 2,3,4,5,6 için denenerek kümeler arası ilişkiler incelendi. Çalışmada küme içi benzerlikler maksimum, kümeler arası benzerlikler ise minimuma indirgenmiştir. Her iki algoritma da defalarca çalıştırılarak optimum sonuca ulaşmak mümkündür. Bu çalışma ile SOM ve X-Means algoritmalarının çalışma zamanlarına bakılarak bir karşılaştırma yapılmış ve SOM algoritmasının X-Means'e göre daha yavaş çalıştığı gözlemlenmiştir. Aynı zamanda SOM algoritmasının ise kümeleme başarımında X-means algoritmasına göre daha etkili olduğu da yapılan çalışmalar sonucunda elde edilmiştir. Sonuç olarak SOM kümeleme algoritması verileri kümelemede x-means algoritmasına göre zaman açısından bakıldığında daha yavaş çalışsa bile daha iyi sonuç verdiği gözlemlenmiştir.

6-Kaynaklar

Bilen, Ö., (2004), ÖSS Sınav Sonuçlarının Okul Bazında Veri Madenciligi ile incelenmesi, Yüksek Lisans Tezi, FEN Bilimleri Enstitüsü, Yıldız Teknik Üniversitesi, İstanbul(Yayınlanmamıs).

D. Pelleg, and A. Moore: Accelerating exact X-means algorithms with geometric reasoning, *KDD-99*, 1999, 277-281.

D. Pelleg, and A. Moore, _x-means: Extending X- means with efficient estimation of the number of clusters, *17th International Conf. on Machine Learning*, 2000, 727–734

Haykin, S., 1994, Neural networks: A comprehensive foundation, MacMillan College Publishing Comp. Inc., New York.

Haykin, S., 1999a, Learning processes; single – layer perceptrons; multilayer perceptrons. Neural Networks: A Comprehensive Foundation. 2‟nd edition, Prentice Hall International Inc, USA, p. 14-68.

Keim D. A., (2002), "Information Visualization and Visual Data Minind", IEEE on Transactions on Visualizations and Computer Graphics, 8:100-107

Martinez, W.L. ve Martinez A. R., (2005), Exploratory Data Analysis with MATLAB, Boca Raton : CRC Press, USA.

Schatzmann, J., (2003), Using Self-Organizing Maps to Visualize Clusters and Trends in Multidimensional Datasets, Final Year Individual Project Report, Department of computing Data Mining Group, Imperial College, London.

T.Kohonen,Self-organizedformationoftopologicallycorrectfeaturemaps,Biol. Cybern.43(1982)59–69.

Toledo, M.D.G., (2005), A Comparison in Cluster Validation Techniques, Yüksek Lisans Tezi, University of Puerto Rico Matematics Department, Puerto Rico

Zontul, M., Kaynar, O. ve Bircan, H., (2004), "SOM Tipimde Yapay Sinir Ağlarını Kullanarak Türkiye' nin İthalat Yaptığı Ülkelerin Kümelenmesi Üzerine Bir Çalışma", Cumhuriyet Üniversitesi, İktisadi ve İdari Bilimler Dergisi, 5(2):47-68.

Weiss, S., ve Indurkhya, N., 1998, Predictive data mining: A practical guide. *Morgan Kaufmann,* DMSK Software: www.data-miner.com., [Erişim Tarihi : 13.02.2018].