

Birliktelik Kurallarının Madenciliği İçin Genetik Algoritma ve Bulanık Küme Tabanlı Yeni Bir Yaklaşım

Bilal ALATAŞ ve Ahmet ARSLAN*

Bilgisayar Mühendisliği Bölümü, Mühendislik Fakültesi, Fırat Üniversitesi, 23119, ELAZIĞ

* Bilgisayar Mühendisliği Bölümü, Mühendislik-Mimarlık Fakültesi, Selçuk Üniversitesi, KONYA
e-posta: balatas@firat.edu.tr

Özet

Bu makalede, {alındı, alınmadı} ya da {var, yok} şeklinde ikili değerler dışında kategorik ve nicel değerler de içeren veri tabanlarında birliktelik kurallarının keşfi için yapay zeka ve zeki hesaplama tekniklerinden genetik algoritma bulanık mantık tabanlı etkili, yeni bir yöntem geliştirilmiştir. Genetik algoritmalarda başlangıç popülasyonunu gelişigüzel üretmek yerine, bunu çözüm uzayına düzgün dağıtan düzenli popülasyon yöntemi kullanılmıştır. Genelde kullanılan yöntemlerin aksine yüksek destek ve güven değerlerine sahip birliktelik kuralları yoğun nesne kümeleri üretilmeden direkt olarak ve her veri tabanı için belirlenmesi güç olan minimum güven ve minimum destek eşiklerine ihtiyaç duyulmadan keşfedilmiştir. İlginç birliktelik kurallarını bulmak için uyarlamalı mutasyon ve elitizm stratejisi uygulanmıştır. Bu şekilde genetik algoritmanın son popülasyonu ilginç birliktelik kurallarını temsil etmiştir. Önerilen yöntem hem yapay bir veri tabanında hem de Fırat Üniversitesi Elektrik-Elektronik Mühendisliği lisans öğrencilerinin ders not kayıtlarında denenmiş, kullanışlı ve ilginç kurallar etkili şekilde bulunmuştur.

Anahtar Kelimeler: Veri Madenciliği, Birliktelik Kuralları, Genetik Algoritma, Bulanık Kümeler

A New Approach Based on Genetic Algorithm and Fuzzy Set for Mining of Association Rules

Abstract

In this article, a new, efficient method based on genetic algorithm and fuzzy logic, techniques of artificial intelligence and soft computing, has been developed for mining association rules within databases that may contain categorical or quantitative values except for binary values. The initial population of genetic algorithm has not been randomly generated, uniform population method that distributes the initial population in the feasible region uniformly has been used instead. Contrary to the methods used as usual, association rules that have high support and confidence have directly been mined without generating frequent itemsets and without needing minimum support and minimum confidence thresholds that are hard to determine for each database. Adaptive mutation and elitism strategy have been performed for finding interesting association rules. Thus, the last population of genetic algorithm represents the association rules. The proposed method has been tested in both artificial and real world database (records of the students of Fırat University Electrical and Electronics Engineering) and useful and interesting rules have effectively been mined.

Keywords: Data Mining, Association Rules, Genetic Algorithm, Fuzzy Sets

1. Giriş

Veri madenciliği (VM), eldeki büyük miktardaki veriden üstü kapalı, çok net olmayan, önceden bilinmeyen ancak potansiyel olarak kullanışlı bilginin istatistik, makine öğrenmesi,

yapay zeka ve örüntü tanıma yöntemleri kullanılarak çıkartılması süreci olarak tanımlanmaktadır. Birliktelik kurallarının keşfi de VM’de en çok kullanılan tekniklerden birisidir.

İlk önerilen birliktelik kural madenciliği algoritmaları sadece iki değer alabilen yani alındı/alınmadı şeklinde tutulan market sepeti verisi üzerinden çalışmaktadır [1]. Genel olarak kuralların bulunması iki aşamadan oluşur. İlk aşamada kayıt verilerinin taranmasıyla nesne kümeleri sayılır ve aday kümeler üretilir. Kayıtlardaki nesne kümelerinin sayısı önceden tanımlı eşik değerden (minimum destek) büyükse bu nesne kümesi, yoğun nesne kümesi olarak gösterilir. Önce 1-nesne kümesi içeren nesne kümeleri işleme tabi tutulur. Sadece 2-nesne kümesi içeren yoğun nesne kümeleri, 1-nesne kümesi içeren aday nesne kümelerinden oluşturulur. Bu işlemler bütün yoğun nesne kümeleri bulunana kadar devam ettirilir. İkinci aşamada ise ilk aşamada bulunan yoğun nesne kümelerinden birliktelik kuralları oluşturulur. Bütün olası birliktelik kuralları her yoğun nesne kümesinin kombinasyonları şeklindedir ve hesaplanan güven değeri önceden tanımlı eşik değerden (minimum güven) büyük olmalıdır. Bu işlemin sonucunda birliktelik kuralları elde edilir.

Ancak günlük hayatta veriler bu şekilde ikili olabileceği gibi kategorik (posta kodu, araba modeli) ya da sürekli değerli (yaş, gelir) de olabilir. Bunun için önerilen metodun sürekli değerli verileri de içerecek şekilde genişletilmesi yapılmıştır. Nicel birliktelik kuralları ilk olarak Srikant tarafından ele alınmıştır [2]. Srikant sunduğu algoritmada, nicel özellik problemini, özellikleri bölümlere ayırıp, bir sonraki bölümle birleştirdikten sonra ikilik değere çevirerek çözer. Burada önerilen diğer bir aralık bölme yöntemi ise aralıkları üst üste binen bölgeler şeklinde seçmektir. Burada bir bölme, diğer bölmelerle üst üste binmiştir. Sonuç olarak elemanlar, bölümlenme sınırında yerleştiğinden bir bölmeden daha fazla bölmeye katkıda bulunur.

Bununla beraber geçerli nicel birliktelik kuralı madenciliği algoritmaları bazı problemler içerir. İlk problem bölümler arasında keskin sınırların bulunmasıdır. Algoritmalar bölümler arasındaki sınır elemanların etkisini ya zayıflatır ya da aşırı değerli hale getirir. Oluşturulan bölümlerin sınırlarındaki elemanlar insanların değişik cevaplar vereceği değerlerde olabilir. Başka bir problemde bu şekilde ikilik değerlere çevirmenin nitelik sayısını oldukça arttırmasıdır. Bunu önlemek için farklı yaklaşımlar önerilmiştir. Bunlar, yapay zeka ve zeki hesaplama tekniklerinden genetik algoritma (GA) yaklaşımı [3], ve bulanık mantık yaklaşımıdır [4].

Bulanık birliktelik kuralları madenciliği, bulanık küme kavramları kullanılarak birliktelik kurallarının keşfidir. Bulanık kümeler, bir kümenin aralığa üye olan kısımları ve aralığa üye olmayan kısımları arasında yumuşak bir geçiş sağlar. Ayrıca, bulanık birliktelik kuralı dilsel terimler içerdiğinden daha anlaşılardır. Bu çalışmada da bulanık kümeler ve etkili bir arama yöntemi olan GA’ların beraber kullanılması ve özellikle GA’larda çeşitli etkili yöntemlerle birliktelik kurallarının keşfi yapılmıştır.

Bu çalışmanın organizasyonu şu şekildedir: İkinci bölümde, birliktelik kurallarını bulmak için önerilen yöntem anlatılmıştır. Üçüncü bölümde, kullanılan yapay ve gerçek veritabanları hakkında bilgi verilmiştir. Dördüncü bölümde, simülasyon sonuçları sunulmuştur.

Son olarak beşinci bölümde, makale sonuçlandırılmış ve kısaca gelecekte çalışılması düşünülen konulardan bahsedilmiştir.

2. Yöntem

Bu yöntemde, daha önce önerilen yöntemlerin aksine birliktelik kuralları yoğun nesne kümeleri üretilmeden direkt olarak keşfedilmiştir. Bunun için GA kullanılmış ve GA'lardaki kromozomlar da birliktelik kurallarını temsil etmiştir. İlginç tüm kuralları hızlıca bulmak için GA'larda düzenli populasyon [5-6] ve uyarlamalı mutasyon kullanılmıştır. En son populasyondaki bireyler, yüksek destek ve güven değerlerine sahiptir.

Kullanılan yöntemde bulanıklaştırma işlemi kullanıcı tarafından yapılmış ve daha sonra bu bulanıklaştırılmış değerler üzerinden GA çalıştırılmıştır. Yani bulanıklaştırma aşamasında, GA yerine kullanıcı tanımlı, hazır üyelik fonksiyonları kullanılmıştır. Algoritmanın temel adımları Şekil 1'de gösterilmiştir. Yöntemin ayrıntıları alt bölümlerde açıklanmıştır.

```
t=0;
P(t) := DüzenliPopulasyonuBaslat(nesne_sayisi);
while(t < maksimum_nesil_sayisi) do
    Degerlendir(P(t), veri_kumesi);
    t := t+1;
    P(t) := Secim(P(t-1));
    Caprazlama(P(t));
    Mutasyon(P(t));
end while;
```

Şekil 1. Algoritmanın temel adımları

2.1. Kodlama

Kromozomlar kuralları temsil etmektedir ve kromozomdaki her bir gen de, bir nitelik-değer çiftinden ve bir konum değerinden oluşmaktadır. Kodlamadaki konum değeri K_j , ilgili niteliğin kuralda yer alıp almadığını, eğer kuralda yer almışsa kuralın sağında mı solunda mı olduğunu belirtmektedir. K_j üç farklı değer almaktadır: “0”, “1” ve “*”. “0” değeri ilgili niteliğin kuralın solunda bulunacağını, “1” değeri ise sağında bulunacağını göstermektedir. “*” değeri ise niteliğin kuralda yer almayacağını belirtmektedir. Nitelik-değer çifti $A_i=D_{ij}$ şeklindedir ve A_i , i . niteliği D_{ij} de A_i 'nin j . değerini temsil eder. Kodlamayı basitleştirmek için i . niteliğin i . gende kodlandığı bir konumsal kodlama kullanılmıştır. İkili değerliler için D_{ij} , 1 ya da 0 olarak gösterilir. Sürekli değerli nitelikler için ise bu, bulanık kümelerdeki dilbilimsel değerlerden oluşur. Kromozom kodlaması Şekil 2'de gösterilmiştir.

Gen ₁		Gen ₂		...	...	Gen _m	
K_1	D_{1j}	K_2	D_{2j}	...	...	K_m	D_{mj}

Şekil 2. Kodlama

2.2. Başlangıç popülasyonu

Genetik süreç içerisinde başlangıç popülasyonu gelişigüzel üretildiğinden, her zaman çözüme ulaşmak yerine bazen yerel çözümde kalınırken, bazen de çözümden uzaklaşma durumu ile karşılaşılabilir. Çünkü üretilen popülasyon gelişigüzel bir yapıya sahiptir ve çözümden uzakta bir araya toplanmış kromozomlardan oluşabilir. Hatta, bütün kromozomları çözümden uzakta bulunabilir. Bu durumda çözüme ulaşmak çok zor olabilir ve fazla zaman alabilir. Bundan dolayı, başlangıç popülasyonun bu dezavantajlı durumunun giderilmesi gerekir. Bu çalışmada bireyleri çözüm uzayına düzgün dağıtan düzenli popülasyon yöntemi [5-6] kullanılmıştır.

Bu yöntemin bir r parametresi vardır ve $r = 1$ durumunda bir kromozom gelişigüzel üretilir, bu kromozomun tümleyeni alınır ve bu tümleyen kromozom da popülasyon içerisinde bir kromozom olarak yerini alır. $r = 2$ olduğu durumda popülasyon biraz daha düzenli hale gelmektedir. Bu durumda gelişigüzel üretilen kromozom iki parçaya bölünür. İlk olarak sağ yarının tümleyeni ile sol yarının birbirine eklenmesi ile bir kromozom elde edilir. Ondan sonra sol yarının tümleyeni ile sağ yarının birbirine eklenmesi ile başka bir kromozom elde edilir. Son olarak gelişigüzel üretilen kromozomun tümleyeni alınarak başka bir kromozom elde edilir. Kısaca gelişigüzel olarak üretilen bir kromozomdan 3 tane daha kromozom elde edilmektedir. İlk seçilen birey $\vec{x}$ ise $\vec{y}, \vec{z}, \vec{t}$ türetilen bireyler

$$\vec{x} = (x_1, x_2, \dots, x_n), \vec{y} = (\bar{x}_1, \bar{x}_2, \dots, \bar{x}_n) \quad \vec{z} = (\bar{x}_1, \bar{x}_2, \dots, \bar{x}_{\frac{n}{2}}, x_{\frac{n}{2}+1}, \dots, x_n)$$

$$\vec{t} = (x_1, x_2, \dots, x_{\frac{n}{2}}, \bar{x}_{\frac{n}{2}+1}, \dots, \bar{x}_n)$$

şeklinde temsil edilir. İndislerde görülen $n/2$ ifadesinde bölme işleminin tam kısmı alınır. Eğer ϕ tane gelişigüzel üretilen kromozom varsa ve $r = m$ ise, başlangıç popülasyonunun boyutu aşağıdaki formül ile hesaplanabilir:

$$(2^m - 1)\phi + \phi = \phi 2^m \quad (1)$$

2.3. Uygunluk fonksiyonu

Direkt birliktelik kurallarını bulacağımız için, GA'daki kromozomların veritabanında sık olarak geçen birlikteliklerin ne kadar sıklıkla geçtiğini ve bu sıklığın güven değerini gösteren bir uygunluğa sahip olmaları gerekmektedir. Buna göre uygunluk olarak aşağıdaki ifade seçilmiştir:

$$\text{Uygunluk} = \alpha \times \text{kapsama} + \beta \times \text{güven} + \gamma \times \text{nitelik sayısı} \quad (2)$$

kapsama değeri, veritabanında kromozomdaki ilgili konum ve değerlerin içerdiği tüm kayıtlar için üyelik derecelerinin toplamının veritabanındaki kayıt sayısına bölümüdür. Bu değer literatürde kullanılan güven değeri olarak düşünebilir. Üyelik derecesi bir kayıt için

$$\min_{i \in X} \mu_i \quad (3)$$

olur. Burada μ_i , i . nitelik için derece, X ise konum değerleri “0” ya da “1” olan niteliklerin kümesidir (nesne kümesi). Yani kromozom, kapsama değerini genlerindeki konum değeri “*” olmayan niteliklerinden alır. Bir kromozomun herhangi bir kayıttan alacağı *kapsama* değeri, kaydın ilgili nitelik değerlerinin üyelik derecelerinin min operatörüne tabi tutulmasıyla elde edilecek değerdir. Mesela bir kromozomun temsil ettiği kuralın sol kısmının yani kromozomun konum değeri “0” olan geninin (*Türk_Dili=Orta*) ve sağ kısmının yani kromozomun konum değeri “1” olan geninin (*İngilizce=İyi*) olduğunu düşünelim. Verilen bir kaydın *Türk_Dili* niteliği için 38, *İngilizce* niteliği için 69 değerlerine sahip olduğunu varsayalım. Aynı zamanda *Türk_Dili*’nin *Orta* dilbilimsel terimi için 0.4, *İngilizce*’nin *İyi* dilbilimsel terimi için 0.6 değerini verdiğini varsayalım. O zaman bu kaydın bu kromozoma vereceği kapsama değeri $\min(0.4, 0.6)=0.4$ olur. Bir kromozomun tüm kayıtlardan elde edeceği ve uygunluk fonksiyonunda kullanılacak olan *kapsama* değeri de şu şekilde hesaplanır:

$$\text{kapsama} = \frac{\sum_{t_j \in T} \min_{i \in X} \mu_i}{N} \quad (4)$$

Burada T , tüm veritabanı; N kayıt sayısı; t_j ise her bir kayıttır.

güven değeri, aynı kromozomun *kapsama* değerinin, kromozomda kuralın sol tarafını oluşturan niteliklerin, yani konum değeri “0” olan genlerin temsil ettiği niteliklerin sahip olduğu *kapsama* değerine bölünmesiyle elde edilir. Yani X ve Y nesne kümeleri ise ve kromozom “ $X=A \Rightarrow Y=B$ ” şeklinde bir kuralı temsil ediyorsa, kromozomdaki X ve Y ’nin ikisinin birden (4) ifadesinden elde edeceği kapsama değerinin, sadece X ’in (4)’ten elde edeceği kapsama değerine bölünmesiyle güven değeri hesaplanır. Yani:

$$\text{güven} = \frac{\sum_{t_j \in T} \min_{i \in X} \mu_i}{\sum_{t_j \in T} \min_{i \in Z} \mu_i} \quad (5)$$

Burada Z , konum değerleri “0” olan niteliklerin kümesidir.

nitelik sayısı değeri ise, kromozomdaki nitelik sayısıdır. Yani kromozomda konum değeri “0” ya da “1” olan gen sayısıdır. Birlikte kuralı için en az iki konum değerinin “*”’dan farklı olması ve birinin “0” diğerinin ise “1” olması, yani kuralın sol ve sağ tarafının bulunması gerekmektedir. Bu şartı sağlamayan kromozomların uygunluğuna çok küçük bir değer atanır. Bu şekilde kromozomun hayatta kalması engellenir. Bu terim kuralın içerdiği yoğun nesne

kümesinin boyutu anlamına gelir ve ne kadar büyükse o kadar nesnenin sık olarak veritabanı içinde yer aldığını gösterir. Uygunluğun değerini arttırıcı yönde etki etmektedir.

2.4. Genetik operatörler

Seçim için rulet tekerleği, çaprazlama olarak iki noktalı çaprazlama kullanılmıştır ve oranı %50 seçilmiştir. Mutasyon için uyarlamalı bir mutasyon oranı seçilmiştir. Bilindiği gibi GA'lar çalıştırılıp sonuç elde edildiğinde son popülasyondaki bireylerin birbirine benzediği görülmektedir. Yani tek bir sonuç elde edilmektedir. Burada ise ilginç kuralları bulmak istenmekte yani birden fazla çözüm aranmaktadır. Bu yüzden en iyi birey popülasyonda baskın olduğundan mutasyon oranı da orantılı olarak arttırılmıştır. Kısaca en iyi bireyin popülasyonda ne kadar kopyası varsa o oranda mutasyon oranı da arttırılmıştır. Ayrıca elitist strateji de izlenerek en iyi bireylerin kaybedilmesi önlenmiştir.

2.5. Algoritmanın karmaşıklığı

Kromozomların veri kayıtlarıyla karşılaştırılması algoritmanın karmaşıklığının en kritik kısmıdır. Algoritma karmaşıklığının bu kısmı dört parametreye çok bağlıdır: veritabanındaki kayıt sayısı (KS), popülasyon boyutu (PB), veritabanındaki nitelik sayısı (NS) ve verilen bir genetik süreç içindeki iterasyon sayısı (IS). Her iterasyonda, önerilen algoritma veriyi sadece bir kez okur ve tüm kromozomları okunan kayıtlarla karşılaştırır. Popülasyon boyutu genellikle veri örnek boyutundan çok küçüktür, bu yüzden popülasyonda birkaç tarama gerçekleştirmek veriyi birçok kere gözden geçirmekten çok daha etkilidir. Burada veritabanı genellikle bir diskte depolanır ve popülasyon bilgisayar hafızasında yer alır. Popülasyondaki her kromozomun veritabanındaki her kayıta karşılık bir değer-gen karşılaştırması yapıldığından toplam karşılaştırma sayısı (K),

$$K = KS \times PB \times NS \times IS \quad (6)$$

olarak bulunur. Bu yaklaşımının toplam karmaşıklığı ilk bakışta oldukça yüksek görülebilir. Ancak nicel birliktelik kurallarının keşfi, çok fazla arama uzayı içerir. Ayrıca önerilen yöntem, ekstra bir hesapsal maliyeti olan yoğun nesne kümelerinin üretimini yapmadan birliktelik kurallarını direkt olarak bulur. Arama uzayı çok fazla sınırlandırıldığında, pratikte önerilen algoritmanın etkili olarak çalışacağı unutulmamalıdır.

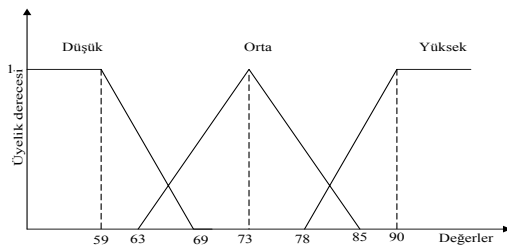
3. Veri Kümesi

3.1. Yapay veritabanı

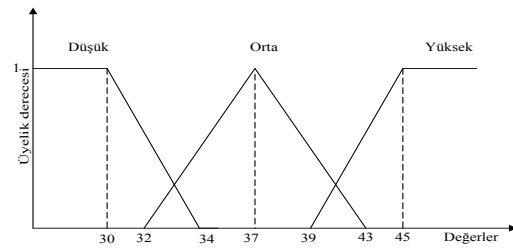
Önerilen yöntemi test için tüm nitelikleri (N_i , $1 \leq i \leq 5$) sürekli değer alan 10 kayıtlık bir veri tabanı oluşturulmuştur ve Tablo 1'de gösterilmiştir. Sürekli değer alan bu niteliklerin her biri, bir dilbilimsel terimle ilişkilendirilmiştir. Bu çalışmada kullanılan veritabanında, N_2 0-50, diğer nitelikler de 0-100 aralığında sayısal değerler alabilmektedir. Buna göre, kullanılan üyelik fonksiyonu N_1 , N_3 , N_4 ve N_5 için Şekil 3(a)'da, N_2 için ise Şekil 3(b)'de gösterilmiştir.

Tablo 1. Yapay veri tabanı

Kayıt	N ₁	N ₂	N ₃	N ₄	N ₅	Kayıt	N ₁	N ₂	N ₃	N ₄	N ₅
1	83	34	65	85	89	6	86	24	64	84	88
2	73	43	79	84	62	7	71	43	75	71	80
3	75	33	86	86	79	8	65	33	86	61	87
4	86	38	86	71	68	9	70	43	87	72	79
5	61	44	89	77	80	10	84	45	86	79	89



(a)

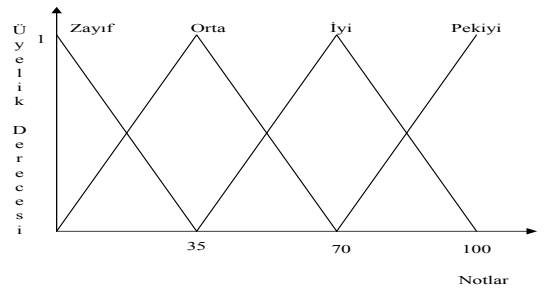


(b)

Şekil 3. Üyelik fonksiyonları (a) N₁, N₃, N₄ ve N₅ için (b) N₂ için

3.2. Gerçek veritabanı

Gerçek veri olarak, Fırat Üniversitesi Elektrik-Elektronik bölümü lisans öğrencilerinin ders notları kullanılmıştır. Derslerle ilgili ayrıntılı bilgiler <http://www.firat.edu.tr/> adresinden alınabilir. Program, ortak olarak alınan dersler üzerinde çalıştırılmıştır. Kayıt dondurma, öğrencilerin ayrılması, seçmeli dersler, derslerin değiştirilmesi vb. nedenlerden dolayı çoğu öğrenci tarafından alınmayan ders problemini ortadan kaldırmak için kurallar, dönem dönem çıkarılmıştır. Yani birinci sınıfın birinci dönem kuralları, üçüncü sınıfın ikinci dönem kuralları gibi. Ders notu olarak geçme notu kullanılmıştır. Nitelikler, dersler; değerler de vize ve final notlarından hesaplanan geçme notudur. Tüm nitelikler 0-100 aralığında sayısal değer alacağından hepsi için kullanılan üyelik fonksiyonu Şekil 4'te gösterilmiştir. Her dönemdeki öğrenci kayıt sayısı da Tablo 2'de gösterilmiştir.



Şekil 4. Gerçek veri tabanı için kullanılan bulanık küme ve üyelik fonksiyonu.

Tablo 2. Her dönemdeki kayıt sayısı

Dönem	Kayıt Sayısı	Dönem	Kayıt Sayısı
1. Sınıf 1. Dönem	279	3. Sınıf 1. Dönem	170
1. Sınıf 2. Dönem	272	3. Sınıf 2. Dönem	166
2. Sınıf 1. Dönem	278	4. Sınıf 1. Dönem	157
2. Sınıf 2. Dönem	150	4. Sınıf 2. Dönem	161

4. Simülasyon Sonuçları

Yapay veritabanı için 4 bireylik bir başlangıç popülasyonu gelişigüzel üretilen bir bireyden $r = 2$ durumu için oluşturulmuştur. Düzenli popülasyon için kullanılacak tümeleme işlemi ise aşağıdaki Tablo 3'te gösterilen kurala göre yapılmıştır. Yani kromozomdaki konum değerlerinden “*”ın tümleyeni “0”, “0”ın “1” ve “1”in de “*” olarak belirlenmiştir. Nitelik değerlerinde, karşılıklı olarak “Zayıf”ın tümleyeni “İyi”, “Orta”nın ise “Pekiyi” olarak belirlenmiştir.

Tablo 3. Tümeleme kuralları

	Konum			Nitelik			
Değerler	*	0	1	Zayıf	Orta	İyi	Pekiyi
Tümleyeni	0	1	*	İyi	Pekiyi	Zayıf	Orta

Uyarlamalı mutasyon kullanılırken dikkat edilmesi gereken bir konu vardır. Konum değerlerinde “*” olan genlerin nitelik değerleri, farklı değerler olsa da bu kromozomlar aslında aynı kuralı temsil etmektedir. Mesela $[*|Düşük|0|Orta|1|Orta|*|Yüksek|1|Yüksek]$ ile $[*|Orta|0|Orta|1|Orta|*|Orta|1|Yüksek]$ aynı kuralı yani “ $N_2=Orta \Rightarrow N_3=Orta$ VE $N_5=Yüksek$ ” kuralını temsil etmektedir. Bu durumda bunlar aynı kromozom olarak işlem görür ve uyarlamalı mutasyon ilgili değeri alır. Mutasyon hem konum değerlerine hem de nitelik değerlerine ayrı ayrı uygulanmıştır. Uygunluk fonksiyonunda kullanılan katsayılar $\alpha=1$, $\beta=1$ ve $\gamma=0.1$ olarak seçilmiştir.

Yapay veri tabanı için 10 iterasyon sonucu elde edilen kurallar Tablo 4'te gösterilmiştir. Dikkat edilirse keşfedilen kurallar bir nevi destek değeri olarak yorumlanabilen yüksek kapsama değerlerine sahiptir. Ayrıca güven değerleri de yüksektir.

Fırat Üniversitesi Elektrik-Elektronik bölümü lisans öğrencilerinin ders not kayıtları için gelişigüzel üç kromozom $r = 4$ için düzenli popülasyon yöntemiyle 48 kromozomluk bir başlangıç popülasyonu oluşturulmuştur. Uyarlamalı mutasyon oranı, en iyi kromozom popülasyonda tek bir taneyse %5 olarak seçilmiş, popülasyondaki oranı arttıkça %100'e kadar çıkabilecek şekilde ayarlanmıştır. Yani %5-%100 aralığında değişen oranlarda uygulanmıştır. 40 iterasyon sonunda elde edilen sonuçlar Tablo 5'te gösterilmiştir. Keşfedilen kuralların yüksek güven değerlerine sahip olduğu görülmektedir.

Tablo 4. Yapay veritabanı için keşfedilen kurallar

Kurallar	Kapsama	Güven
$N_4=Orta \Rightarrow N_3=Yüksek$	0,28	0,72
$N_2=Yüksek \Rightarrow N_4=Orta$	0,27	0,71
$N_4=Orta \Rightarrow N_2=Yüksek$	0,27	0,69
$N_3=Yüksek \Rightarrow N_4=Orta$	0,28	0,61

Tablo 5. Keşfedilen kurallar

Dönem	En İyi ve İkinci En İyi	Kurallar	Güven	Dönem	En İyi ve İkinci En İyi	Kurallar	Güven
1	1	ENF101=iyi $\Rightarrow$ TRD109=iyi	0.7541	5	1	EMÜ351=iyi $\Rightarrow$ EMÜ333=iyi	0.6557
	2	YDİ107=iyi $\Rightarrow$ ENF101=iyi	0.7373		2	EMÜ313=iyi $\Rightarrow$ EMÜ331=iyi	0.726
2	1	KİM106=iyi $\Rightarrow$ TRD110=iyi	0.6857	6	1	EMÜ316=iyi $\Rightarrow$ EMÜ312=iyi	0.6608
	2	FİZ112=iyi $\Rightarrow$ TRD110=iyi	0.683		2	EMÜ332=iyi $\Rightarrow$ EMÜ314=iyi	0.6913
3	1	AİT209=iyi $\Rightarrow$ MAT265=iyi	0.702	7	1	EMÜ413=iyi $\Rightarrow$ EMÜ451=iyi	0.6908
	2	YDİ207=iyi $\Rightarrow$ MMÜ207=iyi	0.6694		2	EMÜ425=iyi $\Rightarrow$ EMÜ413=iyi	0.708
4	1	YDİ208=iyi $\Rightarrow$ AİT210=iyi	0.6506	8	1	EMÜ408=iyi $\Rightarrow$ KİM300=iyi	0.6818
	2	EMÜ232=iyi $\Rightarrow$ EMÜ210=iyi	0.6944		2	EMÜ400=iyi $\Rightarrow$ EMÜ422=iyi	0.713

5. Sonuç

Yeni veri tiplerinin madenciliği için yeni algoritma, teknik ve sistemler sürekli geliştirilmektedir. Bu çalışmada da VM tekniklerinden birliktelik kurallarının keşfi için bulanık mantık ve GA tabanlı bir yeni bir yöntem geliştirilmiştir. GA’da bazen rastlanan yerel çözümde kalma ve çözümden uzaklaşma durumlarının önlenmesi ve gerçek çözüme hızlı şekilde ulaşılabilmesi için düzenli popülasyon yöntemi ve etkili birçok kuralın bulunabilmesi için uyarlamalı mutasyon kullanılmıştır. Deneysel sonuçlardan, yöntemin nicel ve kategorik değerler de içerebilen veritabanlarında birliktelik kurallarının keşfi için etkili bir yöntem olduğu görülmüştür.

İlerleyen çalışmalarda birliktelik kural madenciliğinden özellikle negatif, nicel, genelleştirilmiş ve ağırlıklı birliktelik kurallarının keşfi için önerilen yöntem kullanılacaktır. Bunların birbiriyle karşılaştırılarak performans değerlendirilmesi de yapılacaktır. Algoritmaların paralel ve dağıtık versiyonlarıyla ve GA’nın genetik operatörlerindeki ince ayarlamalarla da daha hızlı şekilde farklı veritabanlarında kuralların bulunması hedeflenmektedir.

Kaynaklar

1. R. Agrawal, T. Imielinski ve A. Swami, Mining Association Rules Between Sets of Items in Large Databases, ACM SIGMOD Conf. Management of Data, 207-216, 1993.
2. R. Srikant ve R. Agrawal, Mining Quantitative Association Rules in Large Relational Tables, Proceedings of 25th ACM SIGMOD International Conference on Management of Data, 1-12, 1996.
3. J.M. Vázquez, J.L.A. Macías ve J.C.R. Santos, Discovering Numeric Association Rules via Evolutionary Algorithm, PAKDD 2002, 40-51, 2002.
4. W. Zhang, Mining Fuzzy Quantitative Association Rules, 11th IEEE International Conference on Tools with Artificial Intelligence, 99-102, 1999.
5. A. Karcı ve A. Arslan, Uniform Population in Genetic Algorithms, İstanbul University Journal of Electrical & Electronics (IU-JEEE), 2, 2, 495-504, 2002.
6. B. Alataş ve A. Karcı, Genetik Algoritmelerde Düzenli Populasyon ve Düzenli Operatör, Afyon Kocatepe Üniversitesi Fen Bilimleri Dergisi, 3, 2, 11-26, 2003.