

TEKNİK NOT/TECHNICAL NOTE

TEŞHİS PROBLEMLERİNDE BELİRTİLER UZAYININ KÜÇÜLTÜLMESİ Ahmet BABANLI^{1,2}, Hüseyin N. KARACA^{1,3}, Özge GÜNER^{1,4}

ÖZ

Çok sayıda belirtilerle nitelendirilen nesnelere topluluğunda belirtilerin genel sayısının azaltılarak bilgi yükü daha yüksek olanların ortaya çıkarılması sınıflandırma ve teşhis problemlerinin çözümüne olanak sağlamaktadır. Bu makalede benzer problemlerin çözümünde Boolean fonksiyonları kullanılmış, sadeleştirme süreci yerine getirilmiş, elde edilmiş normal çarpımlar toplamı formunun implikantları yeni belirtiler setinin belirlenmesinde kullanılmışlardır.

Anahtar Kelimeler: Belirtiler uzayı, Bilgi tablosu, Boolean fonksiyonların sadeleştirilmesi, Alternatif bilgi tablosu.

REDUCTION OF ATTRIBUTE SPACE IN DIAGNOSIS PROBLEMS

ABSTRACT

Reducing the number of attributes and emphasizing attributes that have more knowledge in an object community characterized by many attributes, cause to make classification and diagnosis solving problems easier. Boolean functions are used in this article for solving similar problems, also minimization process have been done and the implicants of disjunctive normal forms are used to define the equivalent small set of attributes.

Key Words: Attribute space, Information table, Minimization of Boolean functions, Alternative information table.

1. GİRİŞ

Problemlerin incelenmesinde verilerin ve onların özelliklerinin doğru olarak belirlenmesinin önemi büyüktür. Ama bazı durumlarda problemin doğru asının yeterli derecede belli olmamasından dolayı gereken bilgilerin elde edilmesi kolay olmamaktadır. Bu sebepten, araştırma sürecini tam olmayan, doğruluk derecesi şüpheli, bazı durumlarda ise aslında problemle hiç bir alakası olmayan veriler bazında sürdürmek zorunda kalınır. Verilerin çoğu zaman, pratikten deney sonucu olarak elde edildiği göz önüne alınırsa araştırılan probleme özgü özelliklerin (belirtilerin, niteliklerin) ayırt edilip çıkartılabilmesi problemin matematik modelini daha basit ve sağlam hazırlamaya olanak sağla-

yacaktır.

Bu makalede çok belirtili verilerin sınıflandırılması problemi ele alınmıştır. Veri topluluğunun sonlu olmasından ve her veri biriminin ait olduğu sınıf hakkında bilgi sahibi olunmasından dolayı "önemli" belirtilerin bulunmasına çalışılmıştır. Bu amaçla uygun Boolean fonksiyonlar oluşturulmuş, sadeleştirme işlemleri yerine getirilerek daha az sayılı değişkenler içeren implikantlar elde edilmiş ve onlar kaynak sistemin alternatifleri olarak kabul edilmişlerdir. Önemli belirtilerin bulunması derken biz teşhis için önemi az olan belirtilerin eliminasyonunu göz önüne alırız. Bu eliminasyonun yapılması için bir sadeleştirme işlemi yerine getirilmelidir. Sadeleştirilecek veri bir Boolean fonksi-

¹ Anadolu Üniversitesi, Müh. Mim. Fakültesi, Bilgisayar Müh. Bölümü, Eskişehir.

² E-posta: ababanli@anadolu.edu.tr, Tel: 222 3213550/6555

³ E-posta: hnkaraca@anadolu.edu.tr, Tel: 222 3213550/6568

⁴ E-posta: oguner@anadolu.edu.tr, Tel: 222 3213550/6569

yonla ilgili olduğu için mevcut algoritmalar içerisinde bilgisayar ortamına aktarılması kolay olan Quine-McCluskey algoritması seçilmiştir (Mano, 1991). İşlemler nöroloji hastalıkları verilerine uygulanmış, alınan sonuçların hastalıkların teşhisinde kullanılabileceği fikri önerilmiştir.

2. NESNELER VE BELİRTİLER

H veriler kümesini ve onların özelliklerini temsil eden B belirtiler kümesini ele alalım. Her belirti için $\{v_{jk}\}$ domenler kümesi vardır, yani belirti belli bir değerler kümesinden değer alabilir. Bu bilgilere dayanarak söz konusu veriler B belirtiler uzayında gösterilebilir. Genelde H bir sınıflar kümesi gibi ele alırsa, her bir eleman uzayda bir sınıf (klaster) olarak düşünülebilir. Sınıfın her bir noktası ise gerçek bir veri (örneğin, deney sayesinde elde edilmiş) gibi ele alınabilir.

Bunları matematiksel olarak şu şekilde gösterebiliriz:

$$H = \{h_p\}, p = [1, 9]; X = \{X_i\}, i = [1, n]$$

$$B = \{b_j\}, j = [1, m]; V = \{v_{jk}\}, k = [1, l]$$

Burada

H — sınıflar kümesi, p — sınıfların sayısı (1 ile 9 arasında değer alan indis);

X — nesnelere kümesi, n — nesnelere sayısı;

B — belirtiler kümesi, m — belirtilerinin sayısı;

V — domenler kümesi, v_{jk} — j-ci belirtinin değerler domeni, k - bu domende k-cı değer, l — domeni oluşturan değerler sayısıdır.

Örnek olarak bir tıbbi hastalık dalı ele alırsa; H — hastalıkları, X — teşhisi ve tedavisi bitmiş hastaları, B — bu hastalık dalında olan tüm belirtileri, v_{jk} ise her bir belirtinin alabileceği gerçek değerleri gösterecektir.

Problemin tanımı şu şekilde yapılır: Elimizde sınırlı sayıda (örneğin $n = 50$) gerçek pratik X_i 'lerin olduğunu varsayalım. Her bir X_i 'nin belirtileri (gerçek değerler) ve hangi sınıfa (h_p) ait olduğunu hakkında bilgilere sahibiz. Bu bilgilere dayanarak yeni meydana çıkan X_i 'nin hangi sınıfa ait olduğunu söylemek gerekir. Bu, bir örneklerin tanımlanması problemi ve bu yüzden çok sayıda çözüm metodu mevcuttur (Russel ve Norvig 1995). Bizim amacımız, yeni bir çözüm metodu bulmak değil, olan çözüm metodları için gereken temel bilgilerin kapsamını azaltmaktır. Diğer bir deyişle, elde olan gerçek verilere dayanarak daha "yükümlü" belirtileri bulmak üzere basit çözüm kuralları oluşturmaktır.

Elimizde olan gerçek verileri Tablo 1 şeklinde gösterebiliriz (Warczak ve Massart, 1999).

Tablo 1. Bilgi Tablosu

	b_1	b_2	b_m	h_p
X_1	v_{1k}	v_{2k}	v_{mk}	1
X_2	v_{1k}	v_{2k}	v_{mk}	1
:	:	:	:	:
:	:	:	:	:
:	:	:	:	:
:	:	:	:	:
X_n	v_{1k}	v_{2k}		v_{mk}	P

Tablonun satırlarında nesnelere (örneğin, gerçek hastanın verileri), sütunlarında ise belirtiler yer almaktadır. Satırlarla sütunların kesiştiği yerde bu nesnenin ilgili belirtisinin hangi değeri aldığı gösterilmiştir (sütunda şartlı olarak aynı değerler (v_{jk}) gösterilmiştir; aslında onlar birkaç nesne için aynı, birkaç veya hepsi için farklı olabilir). Tabloya ek olarak h_p sütunu eklenmiştir ve o, şu nesnenin hangi sınıfa ait olduğunu göstermektedir (örneğin, tabloda X_1 ve X_2 'nin h_1 'e ait olduğunu gözükmektedir).

Tablo 1'in verilerini inceleyerek, farklı sınıflara ait olan iki nesnenin hangi belirtilere göre birbirinden farklılaştırılabileceğini görebiliriz (aynı sınıfa ait olan nesnelere birbirinden farklılaştırılmasına bir ihtiyacın olmaması malumdur). X_1, X_2, X_3 nesnelere için h_1 'e, diğer nesnelere için ise diğer sınıflara ait olduğunu varsayalım. O zaman, tablonun verilerine dayanarak X_1 'in X_4 'den hangi belirtilerinin hangi değerlerine göre farklılandığını belirleyebiliriz. Şu belirtilerinin değerlerinin X_1 ve X_4 için farklı olduklarını varsayalım:

$$(1, 4) \rightarrow (b_1, b_2, b_4, b_7, \dots)$$

Yani $b_1, b_2, b_4, b_7, \dots$ belirtilerinin değerleri X_1 ve X_4 için farklıdır, ama b_3, b_5, b_6, \dots 'nin değerleri aynıdır. Bunu matematiksel olarak bir Boolean fonksiyonu dolayısıyla ifade edebiliriz:

$$f_{1,4} = b_1 \vee b_2 \vee b_4 \vee b_7 \vee \dots$$

$f_{1,4}$ fonksiyonunun yorumunu şu şekilde yapabiliriz.: X_1 'i X_4 'den ayırt edilebilmesi için b_1 'in veya b_2 'nin veya b_4 'ün veya b_7 'nin veya... değeri gerekir. X_1 'in diğer nesnelere ayırt edilebilmesi için de benzer fonksiyonları yazabiliriz: $f_{1,5}, f_{1,6}, \dots$. Şimdi X_1 'in tüm diğer nesnelere ayırt edilebilmesi için gereken fonksiyonu şu yerel fonksiyonların mantıksal çarpımı gibi yazabiliriz:

$$f_1 = f_{1,4} \wedge f_{1,5} \wedge \dots \wedge f_{1,n}$$

Aynı ifadeyi diğ er nesnelere için de yazabiliriz: f_2, f_3, \dots, f_n . Şimdi tüm nesnelere birbirinden ayırt edilebilmesi için gereken genel Boolean fonksiyonunu oluştururuz:

$$f(B) = f_1 \wedge f_2 \wedge \dots \wedge f_n$$

Burada $f(B)$ ifadenin B uzayında geçerli olduğu unu göstermektedir.

Bu normal toplamlar çarpımı formülüdür (Conjunctive Normal Form). Onu normal çarpımlar toplamına (Disjunctive Normal Form) dönüştürürsek sonuç ifadenin her bir implikanti daha az sayıda belirti içermek üzere Tablo 1'in daha az sütunlu alternatifini oluşturmak imkanı verecektir.

3. BOOLEAN FONKSİYONLARIN SADELEŞTİRİLMESİ

Mantıksal tasarımda basit Boolean fonksiyonların sadeleştirilmesinde kullanılan en temel algoritma Karnaugh haritası algoritmasıdır. Karnaugh haritası karelerden oluşan bir diyagramdır ve her bir kare bir mintermi ifade eder. Boolean ifadeler mintermlerin toplamları şeklinde yazılabildikleri için Karnaugh haritasında görsel olarak incelenebilirler ve sadeleştirme işlemleri bu diyagram üzerinde yapılabilir (Karnaugh, 1953).

6'dan fazla değ işken içeren fonksiyonların görsel olarak Karnaugh haritasında sadeleştirilmesi ya da bu algoritmanın bir programlama dili aracılığı ile bilgisayar ortamına aktarılması oldukça zordur. Bu sebepten dolayı Karnaugh haritası algoritmasına alternatif bir algoritma Quine-McCluskey algoritması olarak seçilmiştir. Quine-McCluskey algoritması prime implikantların bulunmasına ve sadeleştirilmesine sistematik bir yaklaşım getirir. Aynı zamanda Quine-McCluskey algoritmasının bir programlama dili ile modellenmesi daha kolaydır. Quine-McCluskey algoritmasının heuristik bir yaklaşım yerine kesin sonuç üretmesi elimizdeki verileri sadeleştirmek için en büyük tercih sebebidir (Mano, 1991).

Quine-McCluskey algoritması $A + A' = 1$ kanununun sadeleşecek ifadeye tekrar tekrar uygulanmasıdır. A bir boolean değ işken ise doğ ruluk tablosu ele alındığında A, 0 veya 1 değ erlerinden birini alabilir. $A + A'$ ifadesi; A'nın 0 değ eri için $0 + 1 = 1$ ve A'nın 1 değ eri için $1 + 0 = 1$ sonucunu verecektir. Algoritmanın hedefi fonksiyondaki bütün mintermleri temsil eden minimum sayıda terim bulmaktır. Bunun için öncelikle prime implikantlar bulunur. Daha sonra essential prime implikantlar bulunur. Son olarak kalan prime implikantları temsil eden minimum küme seçilir ve sonuca ulaşılır.

Quine-McCluskey algoritmasına girdi standart çarpımların toplamı şeklinde olmalıdır. Ancak bizim elimizdeki veri standart olmayan toplamların çarpımı şeklinde olabilir. Bu yüzden öncelikle bu veriler algoritmaya girdi olabilecek uygun forma getirilmelidir. Bunun için iki farklı metot bilgisayar ortamında uygulanmıştır.

İlk metotta birkaç öz yinelemeli fonksiyon yardımıyla veri istenen hale sokulmuştur. Bizim elimizdeki veriler:

$$(b_1 \vee b_2 \vee b_3) \vee (b_1 \vee b_2) \dots$$

biçimindedir. İlk metotta \wedge işleminin \vee işlemine göre dağı lma özelliğ i kullanılır ve elde edilen ara sonuç standart olmayan çarpımların toplamı şeklinde olur. Bu ara sonuç kolaylıkla bilgisayar ortamında çarpımların toplamına çevrilir ve algoritmaya girdi olarak hazır konuma gelir. Ancak çok fazla değ işken ve çarpım içeren ifadelerde bu metodun kullanılması oldukça verimsiz sonuçlar verecektir. Örneğ in 18 değ işkenli 20 çarpım olan bir ifadenin hesaplanmasının sonucunda 20^{18} adet terim hesaplanacaktır ve bunun hesaplanması saatler alabilir.

İkinci metot ise Boolean ifadelerde standart çarpımların toplamı ile standart toplamların çarpımının birbirine dönüştürülmesi kuralları esas alınarak bilgisayar ortamına aktarılmasına dayanır. Bir Boolean fonksiyon eğer mintermlerin toplamı olarak yazılmışsa bu fonksiyonun en uygun gösterimi aşağı daki gibi olur:

$$F(b_i) = \sum (m_i)$$

Örneğ in aşağı daki 3 değ işkenli bir fonksiyon için

$$F = b_1' b_2' b_3 + b_1 b_2' b_3' + b_1 b_2' b_3 + b_1 b_2 b_3' + b_1 b_2 b_3$$

bu ifadeyi yazabiliriz:

$$F = m_1 + m_4 + m_5 + m_6 + m_7$$

Sonuç olarak

$$F(b_1, b_2, b_3) = \sum (1, 4, 5, 6, 7)$$

Eğer Boolean fonksiyonumuz mintermlerin değ il de maxtermlerin toplamı olsaydı bu fonksiyonun en uygun gösterimi aşağı daki gibi olurdu:

$$F(b_i) = \prod (M_i)$$

Örneğ in aşağı daki 3 değ işkenli fonksiyon için:

$$F = (b_1' + b_2' + b_3') (b_1' + b_2 + b_3') (b_1' + b_2 + b_3)$$

bu ifadeyi yazabiliriz:

$$F = M_0 M_2 M_3$$

Not: Sadelik için burada mantıksal işlemler uygun aritmetik işaretler dolayısıyla gösterilmiştir.

Sonuç olarak:

$$F(b_1, b_2, b_3) = \Pi(0, 2, 3).$$

Burada dikkat edilirse her iki örnekte de verilen Boolean ifadelerinin doğruluk tabloları aynıdır:

$$F(b_1, b_2, b_3) = \Sigma(1, 4, 5, 6, 7).$$

Yukarıdaki ifadenin komplementi şöyle ifade edilecektir:

$$F'(b_1, b_2, b_3) = \Sigma(0, 2, 3) = m_0 + m_2 + m_3.$$

DeMorgan teoremine göre tekrar komplement işlemi yapılmış olursa:

$$F = (m_0 + m_2 + m_3)' = m_0' m_2' m_3' = M_0 M_2 M_3 =$$

$$\Pi(0, 2, 3).$$

Bu işlemlerden sonra aşağıda görülen eşitlikteki sonuca kolaylıkla varabiliriz:

$$m_i' = M_i.$$

Yani, i indeksli minterm, i indeksli maxterm komplementine eşittir. Bu kuraldan yararlanılarak veriler bilgisayar ortamında oldukça hızlı bir şekilde Quine-McCluskey algoritmasına uygun hale getirilmiştir ve bilgisayar ortamına taşınan algoritma sayesinde hızlı bir şekilde sadeleştirilmiştir.

4. SADELEŞTİRME İŞLEMİNİN BAŞAĞRISI

HASTALIKLARI VERİLERİNE UYGULANMASI

Baş ağ rısı hastalıklarının teşhisinde doktorlar çok sayıda belirti (septom) kullanmaktadırlar. Belirtilerin bir kaçının bulanık nitelikli olması, bazılarının ise aslında hastalıkla hiçbir ilgisinin olmaması teşhis süresini

olumsuz yönde etkilemektedir. Bu durumda daha "yükümlü" belirtilerin tahmin edilmesi doktorun işinin kolaylaşmasına ve teşhisin doğruluğuna olanak sağlayacağı söyleyebiliriz.

Tablo 2'de 11 hastanın 9 hastalık belirtilerini içeren verileri gösterilmiştir. Bu veriler, hastaların teşhis ve tedavi sürecinden geçmiş gerçek verileridir. Veriler, Osmangazi Üniversitesi Tıp Fakültesi Nöroloji ana bilim dalı doktorlarının katkıları ile hazırlanmıştır.

Veriler 3 hastalık a (sınıfa) aittirler. Belirtiler genellikle sözel niteliklidirler ve bu nedenle onların kodlanması gerekir. Her bir sözel değer bir tamsayı ile kodlanmıştır. Örneğin, b_1 belirtisi (baş ağrısının şiddeti) şu şekilde kodlanmıştır:

1 — Hafif

2 — Orta

3 — Orta şiddetli

4 — Şiddetli

Hastaların (nesnelerin) ait olduğu sınıflar (hastalıklar) ise aşağıdaki gibidir:

Sınıf 1: X_1, X_2, X_3, X_4

Sınıf 2: X_5, X_6, X_7, X_8

Sınıf 3: X_{10}, X_{11}

Tablo 2'nin verilerini kullanarak 3. bölümde sözü geçen Boolean fonksiyonlarının oluşturulması gerekir. Hazırlanan yazılım, fonksiyonlar oluşturulması, bir formdan diğerine dönüştürülmesi ve sadeleştirilmesi işlemlerini yerine getiren birkaç modül içermektedir. Yazılım Visual Basic ortamında gerçekleştirilmiştir.

Yazılımın ilk modülü öncelikle farklı sınıflardaki nesnelere teker teker birbiri ile karşılaştırarak farklı be-

Tablo 2. Hastalık Bilgileri

	b1	b2	b3	b4	b5	b6	b7	b8	b9	h
X_1	1	4	1	4	3	1	1	4	6	1
X_2	1	2	1	3	1	1	7	3	2	1
X_3	3	1	2	2	2	2	4	2	2	1
X_4	1	8	1	2	1	3	3	2	4	1
X_5	1	2	1	2	3	3	5	3	8	2
X_6	4	3	1	3	1	1	2	3	5	2
X_7	1	7	1	3	3	1	5	2	2	2
X_8	1	5	1	3	3	1	6	3	7	2
X_9	1	7	1	4	3	1	2	3	2	2
X_{10}	2	8	2	5	1	3	8	2	3	3
X_{11}	2	6	1	1	1	1	5	1	1	3

lirtileri saptar.

Örneğin, burada yapılacak ilk iş; X_1 nesnesi ile sırasıyla $X_5, X_6, X_7, X_8, X_9, X_{10}$ ve X_{11} nesnelerini karşılaştırmak olacaktır. X_1 ile X_5 nesnesi karşılaştırıldığı anda $b_2, b_4, b_6, b_7, b_8, b_9$ belirtilerinin içerdiği değerlerin farklı olduğu gözlenir.

Bu işlem aşağıdaki şekilde ifade edilir:

$$(b_2 + b_4 + b_6 + b_7 + b_8 + b_9).$$

Bütün bu karşılaştırmalar sonucunda elde edilen standart olmayan toplamların çarpımı biçimindeki sonuç aşağıdaki gibi olacaktır:

$$(b_2 + b_4 + b_6 + b_7 + b_8 + b_9)(b_1 + b_2 + b_4 + b_5 + b_7 + b_8 + b_9)(b_2 + b_4 + b_7 + b_8 + b_9)(b_2 + b_4 + b_7 + b_8 + b_9)(b_2 + b_7 + b_8 + b_9)(b_1 + b_2 + b_3 + b_4 + b_5 + b_6 + b_7 + b_8 + b_9)(b_1 + b_2 + b_4 + b_5 + b_7 + b_8 + b_9)(b_4 + b_5 + b_6 + b_7 + b_9)(b_1 + b_2 + b_7 + b_9)(b_2 + b_5 + b_7 + b_8)(b_2 + b_5 + b_7 + b_9)(b_2 + b_4 + b_5 + b_7)(b_1 + b_2 + b_3 + b_4 + b_6 + b_7 + b_8 + b_9)(b_1 + b_2 + b_4 + b_7 + b_8 + b_9)(b_1 + b_2 + b_3 + b_5 + b_6 + b_7 + b_8 + b_9)(b_1 + b_2 + b_3 + b_4 + b_5 + b_6 + b_7)(b_1 + b_2 + b_3 + b_4 + b_5 + b_6 + b_7 + b_8 + b_9)(b_1 + b_2 + b_3 + b_4 + b_5 + b_6 + b_7 + b_8 + b_9)(b_1 + b_2 + b_3 + b_4 + b_5 + b_6 + b_7 + b_8 + b_9)(b_2 + b_5 + b_7 + b_8 + b_9)(b_1 + b_2 + b_4 + b_6 + b_7 + b_8 + b_9)(b_2 + b_4 + b_5 + b_6 + b_7 + b_9)(b_2 + b_4 + b_5 + b_6 + b_7 + b_8 + b_9)(b_2 + b_4 + b_5 + b_6 + b_7 + b_8 + b_9)(b_1 + b_3 + b_4 + b_7 + b_9)(b_1 + b_2 + b_4 + b_6 + b_7 + b_8 + b_9)(b_1 + b_2 + b_3 + b_4 + b_5 + b_6 + b_7 + b_8 + b_9)(b_1 + b_2 + b_4 + b_5 + b_6 + b_8 + b_9)(b_1 + b_2 + b_3 + b_4 + b_6 + b_7 + b_8 + b_9)(b_1 + b_2 + b_4 + b_7 + b_8 + b_9)(b_1 + b_2 + b_3 + b_4 + b_5 + b_6 + b_7 + b_9)(b_1 + b_2 + b_3 + b_4 + b_5 + b_6 + b_7 + b_8 + b_9)(b_1 + b_2 + b_4 + b_5 + b_7 + b_8 + b_9)(b_1 + b_2 + b_3 + b_4 + b_5 + b_6 + b_7 + b_8 + b_9)(b_1 + b_2 + b_4 + b_5 + b_7 + b_8 + b_9)(b_1 + b_2 + b_3 + b_4 + b_5 + b_6 + b_7 + b_8 + b_9)(b_1 + b_2 + b_4 + b_5 + b_7 + b_8 + b_9)(b_1 + b_2 + b_4 + b_5 + b_7 + b_8 + b_9).$$

Bu veriler Quine-McCluskey algoritmasına girdi olmak için uygun biçimde değerlerdir. Bu noktada verileri uygun biçime sokmak için iki farklı metot kullanılabilir. Birincisi yukarıdaki çarpımların yapılmasıdır. Bu çarpımlar yapıldıktan sonra standart olmayan çarpımların toplamı biçiminde veriler elde edilir ve bu veriler standart biçime çevrildikten sonra Quine-McCluskey algoritmasına girdi olabilir. İkinci metot ise yukarıdaki standart olmayan toplamların çarpımı biçimindeki verilerin standart biçime çevrilmesi ve daha sonra standart toplamların çarpımının, standart çarpımların toplamına dönüştürülmesi ile Quine-McCluskey algoritmasına hazır hale getirilmesidir.

İlk metot uygulandığında yukarıda görülen çarpma işlemleri yazılımın ikinci modülü ile yapılır ve sonuçta standart olmayan çarpımların toplamı şeklinde bir ifade elde edilir. Daha sonra bu toplam, standart çarpımların toplamına üçüncü modül ile dönüştürülür. Elde edilen sonuç artık Quine-McCluskey algoritmasına girdi olabilecek biçimdedir. Yukarıdaki ifade her biri 5-6 değer içeren 38 terimin çarpımından oluştuğu için ilk metodun uygulanması çok uygun olmayabilir. Çünkü 5^{38} ile 6^{38} arasında terimle işlem yapılmalıdır.

İkinci metot uygulandığında ise yukarıdaki standart olmayan toplamların çarpımı biçimindeki ifade çarpım işlemi yapılmaksızın yazılımın üçüncü modülü kullanılarak standart biçime dönüştürülür. Bu durumda 342 adet terim elde edilir. 5^{38} ile 6^{38} terimle işlem yapmaktansa 342 adet terimle çalışmak daha anlamlıdır. Bu terimler bir metin dosyasında aşağıdaki gibi saklanır:

```
342
9
111111111
111101111
110111111
110101111
011111111
011101111
010111111
010101111
111111111
111110111
:
```

İlk satırda kaç adet terim olduğu, ikinci satırda ise değer içeren sayısı gösterilmiştir. Örneğin, üçüncü ve dördüncü satır aşağıdaki ifadeyi temsil etmektedir:

$$(b_1 + b_2 + b_3 + b_4 + b_5 + b_6 + b_7 + b_8 + b_9)(b_1 + b_2 + b_3 + b_4 + b_5 + b_6 + b_7 + b_8 + b_9)$$

Daha sonra dördüncü modül kullanılarak standart toplamların çarpımı biçimindeki terimler standart çarpımların toplamına dönüştürülür. Bu durumda ise 414 adet terim elde edilir. Bu terimlerde bir metin dosyasında aşağıdaki gibi saklanır. Buradaki M , minterm sayısını V ise değer (attribute) sayısını göstermektedir.

M 414

V 9

000000000

000000001

000000010

000000011

000000100

000000101

000000110

000000111

000001000

000001001

:

:

:

:

Örneğin, üçüncü ve dördüncü satır aşağıdaki ifadeyi temsil etmektedir:

$$b_1'b_2'b_3'b_4'b_5'b_6'b_7'b_8'b_9'+b_1'b_2'b_3'b_4'b_5'b_6'b_7'b_8'b_9'$$

Bu işlemler sonucunda diğer metotta olduğu gibi Quine-McCluskey algoritmasına girdi olarak kullanılabilir terimler elde edilir.

Herhangi bir metodun kullanılmasından sonra elde edilen veriler, son modül olan Quine-McCluskey algoritmasını gerçekleyen modüle girildiğinde sadeleşme işlemi gerçekleşir ve aşandaki sonuç elde edilir:

$$b_2b_4 + b_2b_9 + b_5b_9 + b_1b_7 + b_2b_7 + b_4b_7 + b_5b_7 + b_7b_8 + b_7b_9 + b_1b_2b_5 + b_1b_2b_6 + b_1b_5b_8 + b_4b_8b_9 + b_2b_3b_5 + b_2b_3b_6.$$

Başka bir yazılım ile de sonucun doğruluğu test edilmiştir. Bu programda, standart olmayan toplamların çarpımı şeklindeki sadeleşmemiş veriler ile en son elde edilen sonucun doğruluk tabloları hesaplanmış, karşılaştırılmış ve iki ifadenin denk ifadeler olduğuna karar verilmiştir.

Sonucun her bir implikanti Tablo 2'nin bilgilerini alternatif olarak ifade etmektedir. Diğer bir deyişle b_2b_4 implikanti bazında oluşturulmuş Tablo 3 ile Tablo 2'nin bilgileri sözü geçen üç hastalığın bir birinden ayır edebilmesi açısından aynıdır. Yani, teşhis için 9 belirtinin yerine 2 tanesinin (b_2 ve b_4 veya b_2 ve b_9 veya b_5 ve b_7 veya vs..) kullanılması yeterlidir.

Tablo 3. {b2, b4} Bilgi Tablosu

	b2	b4
X ₁	4	4
X ₂	2	3
:	:	:
:	:	:
X ₁₁	6	1

5. SONUÇLAR VE TARTIŞMA

Büyük kapsamlı bilgi sistemlerinde belirleyici özelliklerin ortaya çıkarılabilmesi nesnelere sınıflandırılmasını kolaylaştırır ve böylece sistemin genel analizine olanak sağlar. Örneğin tıbbi teşhis problemlerinde çok sayıda bulanık nitelikli belirtiler topluluğundan daha "yükümlü"lerin seçilmesi daha basit teşhis kurallarının oluşturulmasına yardım eder. Bilgili belirtilerin seçilmesi uygun bilgi tablosu bazında Boolean fonksiyonların oluşturulması ve onların sadeleştirilerek normal çarpımlar toplamı formuna dönüştürülmesine dayalı bir algoritma dolayısı ile yerine getirilebilir. Bu algoritma genel nitelikli olup, bilgi tablosu biçiminde gösterilebilir istenilen bilgi topluluğuna uygulanabilir özelliğe sahiptir.

Teşhis kurallarının oluşturulması açısından, örneğin, 9 belirti yerine 2 belirtinin kullanılmasının önemi büyüktür. Genelde azaltılan verilerin sayısı kaynak verilerin iç yapısı ve kapsamı ile ilgilidir. Kaynak verilerin tam olması, gerçeği yansıtmaya derecesinin yüksekliği ve ön sınıflandırmanın dengeli oluşturulması alternatif bilgi tablosunun tüm veri topluluğunu daha iyi temsil etmesine olanak sağlayacaktır. Makalede verilen bilgiler gerçek niteliklidir, ama tüm nöroloji hastalıklarını temsil etmemektedir (nöroloji hastalıklarının genel sayısı ~30'dur). Bu durumda sınıflar hiyerarşisi oluşturularak, sadeleştirme algoritmasının her bir alt sınıfa uygulanması yoluna başvurulabilir. Bu yönde araştırmalar devam etmektedir.

KAYNAKÇA

- Komorowski J., (1999). Rough Sets: A Tutorial. In: S.K.Pal, A.Skowron(Eds), *Rough Fuzzy Hybridization, A New Trend in Decision-Making*, Springer-Verlag. pp. 1-97
- Russel S., Norvig P. (1995). *Artificial Intelligence, A modern Approach*, Prentice Hall International, Inc.
- Walczak B., Massart D.L. (1999). *Rough Sets Theory, Chemometrics and Intelligent Laboratory Systems*, 47, pp.1-16, Elsevier Science BV, Netherlands.
- Maurice Karnaugh, (1953). The Map Method for Synthesis of Combinational Logic Circuits, *Trans. AIEE*. pt I, 72(9):593-599.
- M. Morris Mano.(1991). *Digital Design, 2nd Edition*, Prentice-Hall International, Inc.