



Veri Madenciliğinde Otomatik Kural Üretebilen Bir Uzman Sınıflandırma Sisteminin Geliştirilmesi

Ömer Akgöbek¹

Harran Üniversitesi, Mühendislik Fakültesi, Endüstri Mühendisliği Bölümü, 63100, Şanlıurfa, Türkiye
e-mail: akgobek@harran.edu.tr

Geliş Tarihi: 31 Ekim 2005, Kabul Tarihi: 5 Nisan 2006

© Operational Research Society of Turkey / Yöneylem Araştırması Derneği

Abstract. The biggest problem for an expert system is knowledge acquisition and its evaluation. Acquiring knowledge from an expert is a time-consuming work requiring many meetings, attention to details and methodical studies. Although experts use their expert knowledge easily in daily work, they can not perform the same success in summarizing their knowledge and making them available for an expert system. The analysis of this knowledge and the creation of knowledge base from this knowledge for an expert system require another expertise. In this work, we have developed a system which can produce the rules according to the algorithms of REX (REX-1, REX-2, REX-3) family and generate a result based on the values produced by using these rules. With the help of this system, users can create their own database, enter their knowledge and produce rules from this knowledge without requiring another expert.

Key words: Data mining, expert system, knowledge acquisition, rule induction

Özet. Bir uzman sistem için en büyük problem bilgilerin elde edilmesi ve değerlendirilmesidir. Bir uzmandan bilgi elde etme, karşılıklı görüşmeler gerektiren, uzun zaman alan, dikkat isteyen ve sistematik çalışmalar gerektiren bir işlemdir. Uzmanlar, uzmanlık bilgilerini günlük çalışmalarında rahatlıkla kullanabilmelerine rağmen, bunları özetleme ve bir uzman sistemde kullanılabilir hale getirmede aynı başarıyı gösteremeyebilirler. Bu bilgilerin değerlendirilmesi ve bir uzman sistem için bilgi tabanı haline getirilmesi ayrı bir uzmanlık ister. Bu çalışmada, REX (REX-1, REX-2, REX-3) ailesi algoritmalarına göre kural üreten ve bu kuralları kullanarak verilen değerlere göre sonuç üretebilen bir sistem geliştirilmiştir. Bu sistem yardımıyla herhangi bir uzmana ihtiyaç duyulmadan kullanıcının kendi veritabanını oluşturması, bilgilerini girmesi ve bu bilgilerden kurallar üretmesi sağlanmıştır.

Anahtar kelimeler: Veri madenciliği, uzman sistem, bilgi kazanımı, kural üretme

1 Giriş

Bilgi tabanlı sistemleri hazırlama çok pahalı olmasının yanında, uzmanlardan bilgiyi elde etme ve sonucunu yararlı bir şekilde göstermenin süreci ise yavaştır. Ayrıca bu çalışmalar, hem konuyla ilgili uzmanların hem de yapay zeka uzmanlarının hizmetlerini gerektirir. Bilgi transfer sürecini bir makine tarafından doğrudan elde etme ve kendi iç gösterimini yapma, yeni uzman sistemlerin maliyetini düşürmeyi ve kendi kendini sürekli olarak geliştiren sistemler için hızlı ve verimli araçlar sağlamayı hedefler.

İnsanlar ve bilgisayarlar arasındaki büyük farklardan biri, insanlar bir faaliyete girişirken genellikle eş zamanlı geliştirme yoluyla kendi deneyimlerini kullanarak bunu gerçekleştirirler. Buna karşın, bilgisayarlar kendileri için sağlanan prosedürleri tam olarak yerine getirirler. Bilgisayarlar, programları çok verimli olarak icra edebilirler, ama bilgisayarların performansı kendi deneyimleri ile geliştirilmemiştir (Michalski ve Kodratoff, 1990).

Sistemlerin performanslarını artırmak veya bilgi esaslarını genişletmek amacıyla sistem davranışlarında meydana gelen kalıcı değişiklikler ‘öğrenme’ olarak adlandırılır. Öğrenme yeteneği, insanlar ve diğer canlı organizmalarda mevcut olup bilgisayar sistemlerinde de büyük bir değere sahiptir. Öğrenen bir sistem, bir veya daha fazla görevi gerçekleştirirken kendi bilgi veya yeteneğini de geliştirir. Gelişme, bilgi işleme aktivitelerinin bir sonucu olarak meydana gelir ve birçok şekilde ele alınmalıdır (Wang, 1997).

- Mevcut bilginin artırılması sürecinde bilgi tabanının boyutunun artırılması,
- Buluşsal yöntemle veya arama alternatiflerini kullanarak kazanım süresi boyunca problemin hızlı çözülmesi,
- Gereksiz verilerin elenmesi sürecinde problemi çözmek için ihtiyaç duyulan sürenin azaltılması,
- Çözüm bulma kalitesinin yükseltilmesi,
- Çözebildiği problemin sınıf veya örnek kümesindeki artışlardan etkilenmemesi.

Veri madenciliği ve bilgisayarla öğrenmede sonuç çıkarma için farklı yollar benimsenmiştir. Tümdengelim, tümevarım ve benzerlik oluşturma gibi ana çıkarım tipleri kullanılır. Gözlem ve deneylerden öğrenme, insan öğrenme sürecinin en önemli özelliğidir. Aynı zamanda, öğrenme bilgi kazanımının en iyi yoludur (Forsyth, 1989; Arciszewski, 1996).

2 Veri Madenciliği

Veri madenciliği, verilerin içerisindeki desenlerin, ilişkilerin, değişimlerin, düzensizliklerin, kuralların ve istatistiksel olarak önemli olan yapıların otomatik veya yarı otomatik olarak ortaya çıkarılmasıdır. Başka bir ifade ile veri madenciliği, büyük miktarda veri içerisinde gelecekle ilgili tahmin yapmamızı sağlayacak bağlantı ve kuralların bilgisayar programları kullanılarak aranması, bulunması ve kurallara dönüştürülmesidir (Alpaydm, 2000; SAS Institute Inc., 1999). Bu sayede yığın şeklindeki bir veri tabanından anlamlı ve tüm örnek kümesini ifade eden daha basit bilgilerin elde edilmesi sağlanır. Veri madenciliği; istatistik, veri tabanları, programlama teknikleri ve yüksek performanslı işlem gibi temelleri içerenin yanında, eldeki verilerden anlamlı ve kullanışlı bilgiyi çıkarmaya yarayacak tümevarım işlemlerini formüle etmeye ve uygulamaya yönelik çalışmaların bütünü de içerir.

Veri madenciliği ve bilgi keşfi (data mining & knowledge discovery), özellikle mühendislik, ticaret, tıp ve eğitim alanlarındaki uygulamalarda yeni ve temel bir araştırma sahası olarak ortaya çıkmaya başlamıştır. Veri madenciliği teknikleri, geniş veri kümelerinden anlamlı bilgileri, düzensizlikleri ve veriler arasındaki ilişkileri ortaya çıkarmakta kullanılır. Bu sayede, web üzerinde filtrelemeler, DNA sıraları içerisinde genlerin tespiti, ekonomideki eğilim ve düzensizliklerin tespiti,

İnternet üzerinden alışveriş yapan müşterilerin alışkanlıkları gibi karar verme mekanizmaları için önemli bulgular elde edilebilir (SAS Institute Inc., 1999; Vahaplar ve İnceoğlu, 2001).

Veri madenciliğinde kullanılacak olan veritabanları geniş, yüksek hacimli ve/veya dağıtık şekilde bulunabilir. Kullanılacak olan tekniklerin bu yapılarla uygun olarak tasarlanması çok önemlidir. Bu tekniklerin bilgisayar belleğine sığmayacak kadar büyük veya farklı coğrafi konumlardaki bilgiyi işleme yeteneğine de sahip olması gerekir (Vahaplar ve İnceoğlu, 2001).

2.1 Veri Madenciliğinde Kullanılan Teknikler

2.1.1 İstatistiksel yöntemler. Veri madenciliği çalışması temel olarak bir istatistik uygulaması olarak görülür. İstatistik, eldeki veriden anlamlı bilgiler elde etmek için farklı teknikler kullanır: Sınıflandırma, regresyon öbekleme, hipotez testi, varyans analizi, bağıntı kurma, boyut azaltma (Alpaydın, 2000).

2.1.2 Yapay sinir ağları (YSA). Son zamanlarda yaygınlaşan yapay sinir ağlarında amaç fonksiyonu birbirine bağlı basit işlemci ünitelerinden oluşan bir ağ üzerine dağıtılmıştır. Yapay sinir ağlarında kullanılan öğrenme algoritmaları, veriden üniteler arasındaki bağlantı ağırlıklarını hesaplar. YSA istatistiksel yöntemler gibi veri hakkında parametrik bir model varsaymadığından uygulama alanı daha geniştir ve bellek tabanlı yöntemler kadar yüksek işlem ve bellek gerektirmez (Alpaydın, 2000).

2.1.3 Kümeleme modelleri. Kümeleme modellerinde amaç, üyelerin birbirlerine çok benzediği, ancak özellikleri birbirlerinden çok farklı olan kümelerin bulunması ve veritabanındaki kayıtların bu farklı kümelere bölünmesidir. Kümeleme analizinde; veritabanındaki kayıtların hangi kümelere ayrılacağı veya kümelemenin hangi değişken özelliklerine göre yapılacağı konunun uzmanı olan bir kişi tarafından belirtilebileceği gibi bilgisayar programları tarafından da yapılabilmektedir.

2.1.4 Karar ağaçları ve kural kümeleri. İstatistiksel yöntemlerde veya yapay sinir ağlarında veriden bir fonksiyon öğrenildikten sonra bu fonksiyonun insanlar tarafından anlaşılabilir bir kural olarak yorumlanması zordur. Oysa, karar ağaçları oluşturulduktan sonra kökten yaprığa doğru inilerek her dal bir kural oluşturacak şekilde kurallar yazılabilir. Bu şekilde kural çıkarma veri madenciliği çalışmasının sonucunun doğrulanmasını sağlar (Alpaydın, 2000).

Karar ağacı tabanlı algoritmalar genellikle karar ağacını oluşturmak için en büyük bilgi kazancı veren özelliklerden arama yapmak için bilginin entropi ölçüsünü kullanırlar (Michalski ve Kodratoff, 1990; Wang, 1997). Entropisi en düşük olan karakteristik, en iyi olarak kabul edilir ve bu karakteristik karar ağacının kökünü oluşturur. Örnek kümesi bu karakteristiğe göre küçük alt kümelere bölünür ve ağacın her dalı bir sınıf değerine karşılık gelecek şekilde dallandırılır. Karar ağacı geliştirme prosedürü, eğitime örnekleri kullanıcının belirlediği sınıflandırma ölçütüne göre doğru sınıflandırılmaya kadar devam eder.

Karar ağacı tabanlı algoritmalar Böl-ve-Fethet metoduna göre işlem yaparak örnekleri alt kümelere ayırırlar. Böl-ve-Fethet öğrenme metodu, bir örnek kümesini, her bir alt kümede sadece tek bir sınıf kalıncaya kadar alt kümelere ayırır. Doğrudan kural üreten algoritmalar ise kapsama metoduna göre işlem yaparlar. Bu metot ile verilen örnek kümesinde daha genel kurallar elde etmek için örnek kümesi sınıf değerlerine göre alt kümelere ayrılır ve verilen alt kümede arama yöntemleri kullanılarak kural üretilmeye çalışılır (Akgöbek, 2003).

2.2 Veri Madenciliğinde Karşılaşılan Problemler

Küçük veri kümeleriyle hızlı ve doğru bir biçimde çalışabilen bir veri madenciliği sistemi, çok büyük veritabanlarına uygulandığında tamamen farklı davranabilir. Ayrıca, tutarlı veri üzerinde mükemmel çalışırken, aynı veriye gürültü eklendiğinde önemli bir biçimde kötüleşebilir. Veri madenciliğinde karşılaşılan problemler aşağıda sıralanmıştır (Sever ve Oğuz, 2002; Chen v.d., 2005).

2.2.1 Veritabanı boyutu. Bir veritabanına kayıt girilmesi durumunda buna paralel olarak boyutu da artacaktır. Ancak pek çok algoritma oldukça küçük örneklemeleri ele alabilecek biçimde geliştirilmiştir. Aynı algoritmaların yüzbinlerce kat büyük örneklemelerde kullanılabilmesi için azami dikkat ve ileri programcılık tekniklerinin kullanılması gerekmektedir. Örneklemin büyük olması, durumun gerçekten var olduğunu göstermesi açısından bir avantajdır. Ancak böyle bir örneklemde elde edilebilecek olası durum sayısı çok büyüktür. Bu yüzden veri madenciliği sistemlerinin karşı karşıya olduğu en önemli sorunlardan biri, veritabanı boyutunun çok büyük olabileceğidir. Dolayısıyla veri madenciliği yöntemleri ya sezgisel/buluşsal bir yaklaşımla arama uzayını taramalıdır ya da ileri programlama teknikleri yardımıyla uygun sorgulama yöntemlerini kullanarak indirgemelidir (Sever ve Oğuz, 2002; Wu v.d., 2005; Lavington v.d., 2005).

2.2.2 Gürültülü veri. Büyük veritabanlarında karakteristiklere ait değerler yanlış veya eksik olabilir. Bu durum; veri girişi sırasında yapılan insan hataları, değerlerin yanlış girilmesi, ölçülmesi veya gözlemlerin yanlış yapılmasından kaynaklanır. Veri girişi ya da veri toplanması sırasında oluşan sistem dışı hatalara gürültü adı verilir. Hatalı veri, gerçek veritabanlarında ciddi problemler oluşturabilir. Bu durum, bir veri madenciliği yönteminin kullanılan veri kümesinde bulunan gürültülü verilere karşı daha az duyarlı olmasını gerektirir. Gürültülü verinin yol açtığı problemler tümevarımsal karar ağaçlarında uygulanan metotlar yardımıyla kapsamlı bir biçimde araştırılır. Eğer veri kümesi gürültülü ise sistem bozuk veriyi tanımalı ve ihmal etmelidir. Deneysel sonuçlar, gürültünün öğrenme algoritmasını doğrudan etkileyerek performansının düşmesine sebep olduğunu göstermiştir (Sever ve Oğuz, 2002).

2.2.3 Boş değerler. Bir veritabanında boş değer, veritabanını oluşturan karakteristiklerden herhangi birine ait değerlerin boş olmasıdır. Boş değer, tanımı gereği kendisi de dahil olmak üzere hiç bir değere eşit olmayan değerdir. Eğer bir örneğin bir karakteristiğine ait değeri boş ise o değer bilinmeyen ve uygulanamaz bir değer anlamına gelir. Örneğin, bir hasta veritabanında bir hastaya ait bazı karakteristiklerin boş olması kaçınılmazdır. Boş değerler veri kümesinde bulunuyorsa, ya bu değerler tamamıyla ihmal edilmeli ya da bu değere en yakın değer atanmalıdır (Sever ve Oğuz, 2002).

2.2.4 Eksik veri. Evrendeki her nesnenin ayrıntılı bir biçimde tanımlandığı ve bu nesnelerin alabileceği değerler kümesinin belirli olduğu varsayılın. Verilen bir bağlamda her bir nesnenin tanımı kesin ve yeterli olsa idi, sınıflama işlemi basitçe nesnelerin alt kümelerinden faydalanılarak yapılardı. Bununla birlikte, veriler kurum ihtiyaçları göz önünde bulundurularak düzenlenip toplandığından, mevcut veri gerçek hayatı yeterince yansıtmayabilir. Örneğin, hastalığın tanısını koymak için kurallar sadece çok yaşlı insanların belirtilerinin bulunduğu bir veri kümesi kullanılarak üretilseydi, bu kurallara dayanarak bir çocuğa tanı koymak pek doğru olmazdı. Bu gibi koşullarda bilgi keşfi modeli belirli bir güvenlik (ya da doğruluk) derecesinde tahmini kararlar alabilmelidir (Tolun v.d., 2005).

2.2.5 Artık veri. Verilen veri kümesi, eldeki probleme uygun olmayan veya artık nitelikler içerebilir. Bu durum pek çok işlem sırasında karşımıza çıkabilir. Örneğin, eldeki problem ile ilgili veriyi elde etmek için iki ilişkiyi ortak nitelikler üzerinden birleştirirsek sonuç ilişkide kullanıcının farkında olmadığı artık nitelikler bulunur. Artık nitelikleri elemek için geliştirilmiş teknikler ‘karakteristik seçimi’ olarak adlandırılır. Karakteristik seçimi yalnızca arama uzayını küçültmekle kalmayıp, sınıflama işleminin kalitesini de artırır (Sever ve Oğuz, 2002).

2.2.6 Dinamik veri. Kurumsal çevrimiçi veritabanları dinamiktir, sürekli olarak bilgi girişi ve güncelleme yapıldığından içeriği de sürekli olarak değişir. Bu durum, bilgi keşfi metodları için önemli sakıncalar doğurmaktadır. İlk olarak sadece okuma yapan ve uzun süre çalışan bilgi keşfi metodu bir veritabanı uygulaması olarak mevcut veritabanı ile birlikte çalıştırıldığında mevcut uygulamanın da performansı ciddi ölçüde düşer. Diğer bir sakınca ise, veritabanında bulunan verilerin kalıcı olduğu varsayıлып, çevrimdışı veri üzerinde bilgi keşif metodu çalıştırıldığında, değişen verinin elde edilen örüntülere yansımaları gerekmektedir. Bu işlem, bilgi keşfi metodunun ürettiği örüntülerin, zaman içinde değişen veriye göre sadece ilgili örüntüleri güncelleme yeteneğine sahip olmasını gerektirir (Sever ve Oğuz, 2002).

2.2.7 Farklı tipteki veriler. Gerçek hayattaki uygulamalar makine öğreniminde olduğu gibi yalnızca sembolik veya kategorik veri türleri değil, fakat aynı zamanda tamsayı, kesirli sayılar ve çoklu ortam verisi gibi farklı tipteki veriler üzerinde işlem yapılmasını gerektirir. Kullanılan verinin saklandığı ortam, düz bir veri kütüğü veya ilişkisel veritabanında yer alan tablolar olacağı gibi, nesneye yönelik veritabanları, çoklu ortam veritabanları, coğrafi veritabanları vb. olabilir. Saklandığı ortama göre veri, basit tipte olabileceği gibi karmaşık veri tipleri (çoklu ortam verisi, zaman içeren veri, yardımcı metin, coğrafi vb.) de olabilir. Bununla birlikte veri tipi çeşitliliğinin fazla olması bir veri madenciliği algoritmasının tüm veri tiplerini ele alabilmesini olanaksız hale getirebilir. Bu yüzden veri tipine özgü algoritmalar geliştirilmektedir (Sever ve Oğuz, 2002).

3 Rex Ailesi Algoritmaları

REX ailesi (Akgöbek, 2003) algoritmaları, Rules ailesi algoritmaları gibi örnekte yer alan değerlerin kombinasyonlarına göre kural üretir. Rules ailesi algoritmalarından farklı olarak REX-1 algoritması, kural üretme aşamasında sadece arama yaparak kural üretmez aynı zamanda karakteristik ve değerlerin entropilerini de hesaplar ve düzensizliği en az olan karakteristiğe öncelik verir. REX-2 algoritması, karakteristik ve değerlerin entropilerini kullanarak bilgi kazançlarını hesaplar ve bu değere göre bilgi kazancı yüksek olan karakteristiklere öncelik verir. REX-3 algoritması ise bilgi kazancına göre karakteristikleri sıraladıktan sonra, örnek kümesi üzerinde arama yaparak mevcut kombinasyonlara göre üretilebilecek tüm kuralları üretir ve ardından en çok örneği sınıflandırabilen en genel kuralları seçer.

REX ailesi algoritmalarının kural üretme yöntemleri karar ağacı üreten algoritmaların kural üretme yöntemlerinden farklıdır. Karar ağacı üreten algoritmalar öncelikle bilgi değerine göre seçilen bir karakteristiği ağacın kökü, diğer karakteristikleri ağacın dalları olarak kabul eder. Karar ağacının oluşturulmasından sonra ağacın her bir dalı bir kural oluşturacak şekilde kurallara dönüştürülür. REX ailesi algoritmalarından REX-2'nin kural üretme metodu Şekil-1'de gösterilmiştir.

REX-2 algoritması örnek kümesini yeniden düzenlemek için her karakteristiğin Bilgi Kazancı hesaplar. Hesaplanan bilgi kazancı, karakteristiklerin Kısmi Entropisine bölünerek Bilgi Kazanç Oranları bulunur. Bu oranlar büyükten küçüğe doğru sıralanarak, örnek kümesi bu sıralamaya

göre yeniden düzenlenir. Bu şekilde bilgi kazanç oranı en yüksek olan karakteristiklere öncelik verilmesi sağlanır. Her aşama sonunda tüm örneklerin sınıflandırılıp sınıflandırılmadığına bakılarak işleme devam edilip edilmeyeceği belirlenir.

Örnek kümesinin bilgi değeri 1 no'lu denklem yardımıyla hesaplanır.

$$\text{Bilgi}(S) = - \sum_{i=1}^m \frac{S_i}{|S|} \cdot \log_2 \left(\frac{S_i}{|S|} \right) \quad (1)$$

Buradaki, m örnek kümesindeki sınıf sayısını, örnek kümesindeki örnek sayısını, S_i ise i . sınıfa ait örnek sayısını göstermektedir.

Bir karakteristiği oluşturan değerlerin entropisi 2 no'lu denklem yardımıyla hesaplanır (Klinkenberg, 1996). Denklemde kullanılan, T karakteristikte yer alan bir değeri, $\text{freq}(C_i, T)$ T değerinin geçtiği C_i sınıfına ait örneklerin sayısını, T değerinin geçtiği toplam örnek sayısını, k bu örneklerdeki sınıf sayısını göstermektedir.

$$E(T) = - \sum_{i=1}^k \frac{\text{freq}(C_i, T)}{|T|} \cdot \log_2 \frac{\text{freq}(C_i, T)}{|T|} \quad (2)$$

(N : karakteristik sayısı, KombSay : Kombinasyon sayısı).

Adım-1. Sayısal karakteristikler için aralık tanımlamaları yapılır;

Adım-2. Verilen örnek kümesindeki her değer ve her karakteristiğin entropisi ile her karakteristiğin bilgi kazancı, kısmi bilgi ve bilgi kazanç oranları hesaplanır;

Adım-3. Karakteristikler bilgi kazanç oranlarına göre büyükten küçüğe doğru sıralanır ve örnek kümesi bu sıralamaya göre yeniden düzenlenir;

Adım-4. $n = 1$;

Adım-5. DO

ÇIKIŞ=FALSE;

FOR $I = 1$ **TO** ÖrnekSayısı

IF I . örnek sınıflandırılmamış **THEN**

Örnekteki her karakteristik değerinden birer tane almak şartıyla n 'li kombinasyonlar oluşturulur;

$K = 0$;

DO

$K = K + 1$;

Kural_Uretildi = FALSE;

K . kombinasyon örnek kümesindeki tüm örneklerle uygulanır;

IF n adet değerden oluşan K . Kombinasyon tüm örneklerde tek bir sınıfa karşılık geliyor **THEN**

K . Kombinasyon kural haline getirilir;

Sınıflandırılan örnekler işaretlenir;

Kural_Uretildi =TRUE;

WHILE ($K \leq \text{KombSay}$) **OR** (**NOT** Kural_Uretildi);

END; // **FOR**

$n = n + 1$;

IF tüm örnekler sınıflandırılmış **THEN** ÇIKIŞ = TRUE;

WHILE ($n \leq N$) **OR** (**NOT** ÇIKIŞ)

Adım-6. Aynı örnekleri temsil eden birden fazla kural varsa en genel kurallar seçilir;

Şekil 1. REX-2 algoritmasının kural üretme metodu

Bir karakteristiğe ait entropi 3 no'lu denklem yardımıyla hesaplanır. Bir karakteristiğin entropisi; karakteristikteki değerlerin entropilerinin ilgili değerlerin olasılıkları ile çarpımlarının toplamına eşittir (Klinkenberg, 1996).

$$E(A) = \sum_{i=1}^n \frac{T_i}{|T|} \cdot E(T_i) \quad (3)$$

Denklemden; A bir karakteristiği, n karakteristikteki değer sayısını ve $E(T_i)$ ise i . değer entropisini göstermektedir.

Bir karakteristiğe ait bilgi kazancı, örnek kümesinin bilgi değeri ile karakteristiğe ait entropi değerleri arasındaki farka eşittir. S örnek kümesindeki A karakteristiğine ait bilgi kazancı 4 no'lu denklem yardımıyla hesaplanır.

$$Kazanc(S, A) = Bilgi(S) - E(A) \quad (4)$$

S örnek kümesindeki A karakteristiğine ait kısmi bilgi 5 no'lu denklem yardımıyla hesaplanır.

$$KısmiBilgi(S, A) = - \sum_{i=1}^{A_n} \frac{S_i}{S} \log_2 \frac{S_i}{S} \quad (5)$$

Bu denklemde kullanılan; A_n A karakteristiğinin değer sayısını, S_i A karakteristiğindeki i . değer geçtiği örnek sayısını ve S örnek kümesindeki toplam örnek sayısını göstermektedir.

Her karakteristik için bilgi kazanç oranları ise 6 no'lu denklem yardımıyla hesaplanır.

$$KazancOranı(S, A) = \frac{Kazanc(S, A)}{KısmiBilgi(S, A)} \quad (6)$$

Bilgi kazanç oranlarının büyükten küçüğe doğru sıralanmasından sonra örnek kümesi bu sıralamaya göre yeniden düzenlenir ve kural üretme işlemi başlatılır.

Kural üretme işlemi sırasında örnekler sırasıyla ele alınır. Her örnekteki karakteristik değerlerinin oluşturduğu tekli kombinasyonların sadece bir sınıfta geçip geçmediği kontrol edilir. Kombinasyon tüm örneklerde sadece bir sınıfa ait ise bu kombinasyon kurala dönüştürülür. Aksi halde kombinasyondaki değer sayısı bir artırılarak, kombinasyonun sadece bir sınıfta geçip geçmediği kontrol edilir. Bu işlem ilgili örnek için kural üretilinceye kadar devam eder. Son kombinasyon tüm değerlerden oluşacağından ilgili örneği mutlaka sınıflandıracaktır.

3.1 Algoritmaların Performanslarının Karşılaştırması

Bilgi keşfi için kullanılan algoritmaların performansı ürettikleri kural sayısı ve doğruluk oranlarına göre test edilir. Algoritmalarından istenen, örnek kümesinden az sayıda kural üretmesi ve test veri kümesindeki örneklerin hangi sınıfa ait olduğunu yüksek doğruluk oranında sağlamasıdır. Bu bölümde, bu iki ölçü kullanılarak REX ailesi algoritmaları ile diğer algoritmalar arasında performans karşılaştırması yapılacaktır.

REX ailesi algoritmaları, RULES-3 algoritması üzerine inşa edilmişlerdir. Bundan dolayı öncelikli olarak RULES ailesinin diğer algoritmaları ile hem kural sayıları hem de doğruluk oranlarına göre IRIS veri kümesini kullanarak karşılaştırma yapılmıştır. REX ailesi ve Rules-3 Plus algoritmalarının IRIS veri kümesine uygulanması sonucunda elde edilen kurallar Tablo 1a, 1b ve 1c'de gösterilmiştir. Bu sonuçlara göre REX-1 algoritması tarafından üretilen kural sayısı 8 ve şart sayısı 11 iken, REX-2 ve REX-3 algoritmaları tarafından üretilen kural sayısı 7 ve şart sayısı ise 10'dur. Buna karşın, Rules-3 Plus tarafından üretilen kural sayısı 10 ve şart sayısı 14'tür (Pham ve Dimov, 1997a).

Tablo 1a. REX-1 tarafından elde edilen kurallar

Kural	Kural Tanımı
1	EĞER $1.3 \leq PW < 1.7$ VE $3.95 \leq PL < 4.93$ İSE IRIS=Iris-versicolor
2	EĞER $0 \leq PW < 0.5$ İSE IRIS =Iris-setosa
3	EĞER $1.7 \leq PW < 2.1$ İSE IRIS =Iris-virginica
4	EĞER $0.9 \leq PW < 1.3$ İSE IRIS =Iris-versicolor
5	EĞER $2.1 \leq PW < 2.5$ İSE IRIS =Iris-virginica
6	EĞER $4.93 \leq PL < 5.91$ VE $2.8 \leq SW < 3.2$ İSE IRIS =Iris-virginica
7	EĞER $2.1 \leq PW < 2.5$ İSE IRIS =Iris-virginica
8	EĞER $1.3 \leq PW < 1.7$ VE $2.4 \leq SW < 2.8$ İSE IRIS =Iris-versicolor

Tablo 1b. REX-2 ve REX-3 tarafından elde edilen kurallar

Kural	Kural Tanımı
1	EĞER $1 \leq PL < 1.98$ İSE KARAR=Iris-setosa
2	EĞER $1.7 \leq PW < 2.1$ İSE KARAR=Iris-virginica
3	EĞER $0.9 \leq PW < 1.3$ İSE KARAR=Iris-versicolor
4	EĞER $2.1 \leq PW < 2.5$ İSE KARAR=Iris-virginica
5	EĞER $3.95 \leq PL < 4.93$ VE $1.3 \leq PW < 1.7$ İSE KARAR=Iris-versicolor
6	EĞER $4.93 \leq PL < 5.91$ VE $2.8 \leq SW < 3.2$ İSE KARAR=Iris-virginica
7	EĞER $1.3 \leq PW < 1.7$ VE $2.4 \leq SW < 2.8$ İSE KARAR=Iris-versicolor

Tablo 1c. Rules-3 Plus tarafından elde edilen kurallar [15]

Kural	Kural Tanımı
1	EĞER $1 \leq PL < 1.98$ İSE IRIS=Iris-setosa
2	EĞER $3.95 \leq PL < 4.93$ VE $1.3 \leq PW < 1.7$ İSE IRIS =Iris-versicolor
3	EĞER $5.91 \leq PL < 6.9$ İSE IRIS =Iris-virginica
4	EĞER $4.93 \leq PL < 5.91$ İSE IRIS =Iris-virginica
5	EĞER $2.4 \leq SW < 2.8$ VE $1.7 \leq PW < 2.1$ İSE IRIS =Iris-virginica
6	EĞER $2.96 \leq PL < 3.95$ İSE IRIS =Iris-versicolor
7	EĞER $0.9 \leq PW < 1.3$ İSE IRIS =Iris-versicolor
8	EĞER $2.1 \leq PW < 1.5$ İSE IRIS =Iris-virginica
9	EĞER $3.2 \leq SW < 3.6$ VE $3.95 \leq PL < 4.93$ İSE IRIS =Iris-versicolor
10	EĞER $2.8 \leq SW < 3.2$ VE $1.7 \leq PW < 2.1$ İSE IRIS =Iris-virginica

Tablo 2. IRIS test veri kümesi kullanılarak elde edilen kural sayıları ve doğruluk oranları

Algoritma	Kural Sayısı	Doğruluk Oranı
RULES-3	11	%91.50
ID3	8	%92.50
RULES-3 PLUS	10	%97.50
RULES-4	9	%93.60
REX-1	8	%93.75
REX-2	7	%100.00
REX-3	7	%100.00

REX ailesi ile RULES ailesi algoritmaları arasında, ürettikleri kural sayıları ve doğruluk oranlarına göre yapılan karşılaştırmalar Tablo 2'de verilmiştir. RULES ailesi algoritmalarından en iyi sonucu veren Rules-4 algoritması tarafından üretilen kural sayısı 9 ve şart sayısı 12'dir (Pham ve Dimov, 1997a; Pham ve Dimov, 1997b). REX algoritmalarının ürettiği kural sayıları ve kurallardaki şart sayıları diğer RULES ailesi algoritmalarından daha azdır. Ayrıca, Rules-4 algoritması %93.60, REX-1 algoritması %93.75 ve REX-2 algoritması ise %100 doğruluk oranında sınıflandırma yapmıştır (Pham ve Dimov, 1997a; Pham ve Dimov, 1997b; Pham v.d., 2000). Bu bilgilere göre REX ailesi algoritmaları diğer RULES ailesi algoritmalarına göre hem az sayıda kural üretmekte hem de yüksek doğruluk oranında sınıflandırma yapabilmektedir.

Doğruluk oranlarına göre performans karşılaştırmaları için en yaygın kullanılan 7 algoritma ve toplam 10 adet veri kümesi seçilmiştir (Tolun v.d., 2005; Wu, 1998; Tolun ve Abu-Soud, 1998; An, 2003; Thrun v.d., 1991; Bramer, 2000). Bu veri kümeleri günlük hayattan alınan gerçek veri kümeleridir. Bunların kural üretmek için bir eğitim kümesi ve bu kural kümesinin doğruluğunu test etmek için bir test kümesi bulunmaktadır (Blake ve Merz, 1994).

Tablo 3'deki örnek kümelerinin ortalama doğruluk oranlarına göre en yüksek doğruluk oranı %86.92 ile REX-1, ikinci en yüksek doğruluk oranı %85.42 ile REX-3 ve üçüncü en yüksek doğruluk oranı ise %84.07 ile CN2 algoritmasına aittir. Tablodaki REX algoritmaları için verilen değerlerin tamamı geliştirilen bu sistem tarafından elde edilmiştir.

Tablo 3. Algoritmalar tarafından elde edilen doğruluk oranları

Örnek	C4.5 Rules	C4.5 Prune	ID3	ILA	ILA-2	OC1	CN2	REX-1	REX-2	REX-3
Vote	95.6	97.0	94.1	94.8	97.0	96.3	95.6	97.0	89.6	89.6
Parity5+5	50	50.0	50.8	51.2	50.0	52.4	53.0	64.7	63.8	64.7
Monk1	93.5	75.7	81.0	100	100	91.2	98.6	97.2	97.2	96.8
Monk2	66.2	65.0	69.9	78.5	59.7	96.3	75.4	76.4	72.2	76.9
Monk3	96.3	97.2	91.7	88.2	100	94.2	90.7	93.5	89.4	91.0
Zoo	85.3	85.3	97.1	91.2	88.2	73.5	82.4	97.1	85.3	88.2
Lenses	62.5	62.5	62.5	50.0	62.5	37.5	62.5	62.5	62.5	62.5
Tic-Tac-Toe	98.1	82.2	80.9	98.1	84.1	85.6	98.0	97.1	97.1	98.0
Splice	92.7	90.4	89.0	67.9	73.4	91.2	84.5	83.7	83.2	89.0
Promoter	97.5	95.0	100	100	97.5	87.5	100	100	100	97.5
Ortalama	83.77	80.03	81.7	81.99	81.24	80.57	84.07	86.92	84.03	85.42

4 Geliştirilen Uzman Sınıflandırma Sistemi

Veri madenciliğinin yukarıda sayılan problemlerini ortadan kaldırmak için doğrudan kural üreten REX ailesi algoritmaları kullanılarak örnek kümesinden kural üreten bir sistem geliştirilmiştir. Sistem, veri madenciliği problemlerini gidermek için örnek kümesindeki eksik ve boş veriyle, sayısal ve alfabetik değerli karakteristiklerle, çok büyük veri kümeleriyle ve dilsel değişkenlerle işlem yapabilmektedir. Ayrıca, sayısal değer içeren karakteristikler için aralık tanımlamalarının yapılmasına izin vermektedir.

Geliştirilen uzman sistemde karar verme mekanizması olarak geriye doğru zincirleme tekniği kullanılmaktadır. Bir uzman sistemde yeni bir durumla karşılaşıldığında sonucun ne olacağı, mevcut kural kümesinden buna cevap verecek kuralların bulunmasına bağlıdır. Kural kümesinde arama yapılırken kurallar sırasıyla araştırılır. Verilen karakteristik değerlerinin tamamını doğrulayan kural seçilir. Uyan bir kuralın bulunmaması durumunda en yakın doğruluğu sağlayan kural seçilir. Bunun için, karakteristik değerlerinden ilk önce bir tanesi elenir, sonuç bulunmaması durumunda sonuç bulununcaya kadar diğer karakteristikler birer birer elenir. Buna rağmen sonuç bulunamaması durumunda verilen örnek için herhangi bir sınıflandırma yapılmaz.

Uzman sınıflandırma sisteminde, REX ailesi algoritmalarının her üyesi ile ayrı ayrı kural üretmek mümkündür.

S1	S2	S3	S4	S5	S6	Karar
round	octagon	yes	flag	green	yes	no
round	octagon	no	sword	red	yes	no
round	octagon	no	sword	red	no	no
round	octagon	no	sword	yellow	yes	no
round	octagon	no	sword	blue	no	no
round	octagon	no	balloon	green	no	no
round	octagon	no	balloon	blue	no	no
round	octagon	no	flag	blue	yes	no
square	round	yes	sword	red	yes	yes
square	round	yes	sword	red	no	yes
square	round	yes	sword	blue	yes	no
square	round	yes	sword	blue	no	no
square	round	yes	balloon	red	yes	yes
square	round	yes	balloon	red	no	yes

Şekil 2. Örnek kümesi bilgilerinin girişi için kullanılan arayüzü

4.1 Uzman Sınıflandırma Sisteminin Kullanıcı Arayüzü

Kullanıcının, geliştirilen uzman sistemi kolay bir şekilde kullanmasını sağlamak için uygun programlama teknikleri kullanılmıştır. Bu şekilde kullanıcının kolaylıkla kendi veritabanını oluşturması ve bilgi girişlerini daha rahat bir şekilde yapması sağlanmıştır.

4.2 Yeni Veritabanının Oluşturulması

Bu bölümde örnek kümesinin kısa bir tanımı yapılır. Örnek kümesi için bir veritabanı ismi ve bir açıklama bilgisi ile birlikte örnek kümesinde yer alacak olan karakteristik tanımlamalarının yapılması sağlanır. Örnek kümesinde yer alacak olan karakteristiklerin metin mi, yoksa sayısal mı olduğu ve sayısal alanların kaç sınıftan oluşacağı da bu bölümde belirlenir. Veritabanına yeni karakteristikler ekleme ve mevcut karakteristikleri silme gibi işlemler yapılabilir.

4.2.1 Örnek kümesine bilgi girişi. Örnek kümesinin tanımı yapıldıktan sonra bir veritabanı boş olarak oluşturulur ve bilgi girişine hazır hale getirilir. Girilen her kayıt bir örnek olarak adlandırılır. Tüm örnekler örnek kümesini oluşturur. Örnek girişleri, farklı zamanlarda veritabanına kayıt ilavesi şeklinde yapılabilir. Bilgilerin son şekli kalıcı olarak saklanır. Bilgi giriş ekranı Şekil-2'de gösterilmiştir.

4.2.2 Kural üretme ve doğruluk oranının hesaplanması. Sistemde, kural üretme işlemi REX-1, REX-2 veya REX-3 algoritmalarından birine göre yapılabilir. Her algoritmanın ürettiği kurallar, bir kural kümesi olarak, test veri kümesi ile doğruluk testi yapmak veya verilen değerlerin hangi sınıfa ait olduğunu bulmak üzere bir yazı (text) dosyasında saklanır. Test işlemi yardımıyla doğru olarak sınıflandırılabilen örnek sayısı, sınıflandırılmayan örnek sayısı ve doğruluk oranı değerleri kullanıcıya yansıtılır.

Kural üretme işleminin kullanıcı arayüzü Şekil-3'te gösterilmiştir. Kullanıcı aktif örnek kümesine hangi algoritmanın uygulanacağına ilgili pencereden karar verir. Bu pencerede, üretilen kurallar ve örnek kümesindeki karakteristik ve değerlere ait bilgi ölçüm değerleri görüntülenir. Kullanıcı, Test düğmesi yardımıyla test veri kümesini kullanarak örnek kümesinin doğruluğunu test edebilir. Test işlemi ile test örnek kümesindeki örneklerden sınıflandırılabilen ve sınıflandırılmayanların sayısı ve örnek kümesinin doğruluk oranı hesaplanır. Kaydet düğmesi yardımıyla kural kümesini kaydedebilir ve eğer aralık tanımlamaları yapılmış ise Sayısal düğmesi yardımıyla bu değerlerin dilsel karşılıklarını görebilir. Şekil 3'te, seçilen algoritma yardımıyla örnek kümesinden kurallar üretildikten sonra test düğmesi yardımıyla doğruluk oranı hesaplanmakta ve bu bilgiler görüntülenmektedir.

5 Sonuçlar

Geliştirilen uzman sınıflandırma sistemi yardımıyla herhangi bir uzmana gerek kalmadan ilgili meslek sahibi kendi veritabanını kolaylıkla oluşturabilmekte, bilgi girişlerini yapabilmekte ve bilgilerden kural üretebilmektedir. Üretilen kuralların doğruluğunu test ederek, bilinmeyen bir durum ile karşılaşılması halinde hangi sonucun hangi doğruluk oranında elde edileceğini görebilmektedir. Sistemin avantajları şunlardır:

- Eksik ve boş veri ile çalışabilmekte: Girilen bilgilerde bazı karakteristiklere ait değerlerin bulunmaması sistemin çalışmasını engellememekte ve bu değerler yok kabul edilmektedir.
- Gürültülü verileri ayırt edebilmekte: Kullanılan bilgi ölçüleri (entropi, bilgi kazancı) sayesinde gürültülü veriler elenmekte ve bu veriler kural üretme aşamasında dikkate alınmamaktadır.
- Dilsel değişkenler kullanılabilme: Sayısal değer içeren bir tanımlama yerine dilsel değişkenlerin kullanılması, bilginin daha kolay anlaşılmasının yanında büyük ölçüde bilgi indirgenmesi sağlamaktadır.
- Sorgulama teknikleri yardımıyla hızlı sonuç üretebilmekte: Kombinasyonların örnek kümesi üzerinde tek bir sınıfa ait olup olmadıklarının test edilmesi aşamasında, sorgulama işlemleri SQL komutları yardımıyla yapılarak işlemlerin daha hızlı yapılması sağlanmaktadır.
- Sistem bütün veri tipleri ile çalışabilmektedir: Karakteristiklere ait değerlerin sayısal veya metin formatında olması kaçınılmazdır. Sayısal değerler de kendi içerisinde tamsayı veya ondalıklı sayı olabilmektedir. Karakteristik tanımlamaları sırasında karakteristiğin metin veya sayısal veri tipinde olduğu belirtilir.
- Gerektiğinde sayısal değerler için aralık tanımlamaları yapabilmekte: Sayısal değerlerin doğrudan ele alınması yerine verilen aralıklara göre dilsel değişkenlerin kullanılması daha yararlıdır. Bu sayede daha genel kurallar üretilir. Örneğin, bir örnek kümesinde sıcaklık karakteristiği için 18 ve 19 değerlerinin bir farkı yoksa, bu değerlerin ayrı ayrı ele alınması yerine, $15 < \text{Sıcaklık} < 28$ (sıcaklık 15 ile 28 değerleri arasında) gibi bir değer aralığının kullanılması daha doğru olacaktır. Aksi halde sayısal değer içeren her örnek için bir özel kuralın üretilmesi kaçınılmazdır.

S1	S2	S3	S4	S5	S6	Karar
square	octagon	no	sword	blue	yes	no
square	octagon	no	balloon	green	yes	no
square	octagon	no	balloon	blue	no	no
square	octagon	no	flag	red	yes	no
square	octagon	no	flag	yellow	yes	no
square	octagon	no	flag	blue	no	no
octagon	round	yes	sword	red	yes	yes
octagon	round	yes	sword	yellow	yes	yes
octagon	round	yes	sword	green	yes	yes
octagon	round	yes	balloon	blue	no	no

Üretilen Kurallar

Kural 19: EĞER S2=octagon VE S5=green VE S4=sword İSE Karar=yes [2]
Kural 20: EĞER S2=octagon VE S4=balloon İSE Karar=no [13]
Kural 21: EĞER S2=octagon VE S4=flag İSE Karar=no [14]
Kural 22: EĞER S3=no VE S2=octagon İSE Karar=no [22]
Kural 23: EĞER S2=round VE S5=green VE S1=octagon İSE Karar=yes [2]
Kural 24: EĞER S2=round VE S5=yellow VE S1=octagon İSE Karar=yes [2]
Kural 25: EĞER S2=octagon VE S5=green VE S1=octagon İSE Karar=yes [3]
Kural 26: EĞER S2=octagon VE S5=blue VE S1=octagon VE S4=sword İSE Karar=yes [1]
...

Toplam Kural Sayısı :26
Toplam Şart Sayısı :73
Kural Başına Şart Ortalaması :2.81
Sınıflandırılmayan Örnek Sayısı:0

Doğruluk Oranı

Test Kümesindeki Örnek Sayısı	:432
Sınıflandırılan Örnek Sayısı	:404
Sınıflandırılmayan Örnek Sayısı	:28
Doğruluk Oranı (%)	:93.52

Karakteristik Değer	Entropi	D.Entropi
S6 no	0.995	
S6 yes	0.990	
S6 > > >	0.993	1.985
S3 yes	1.000	

Şekil 3. Kural üretme işleminin kullanıcı arayüzü

- Büyük hacimli veri kümeleri üzerinde işlem yapabilmekte: Veritabanındaki kayıtların çok sayıda olması herhangi bir probleme sebep olmamaktadır. Sistem her veri kümesi için en fazla 255 karakteristikten oluşan 2 milyara kadar örnek girişine izin vermektedir.

Sistemin en büyük dezavantajı ise dinamik veri girişinden sonra her defasında sistemin yeniden çalıştırılmasının gerekliliğidir. Veritabanına her bilgi girişinden sonra girilen bilginin kural kümesini etkileyip etkilemediğinin bilinmesi ve buna göre kural kümesinin yeniden düzenlenmesi pratiklik ve zaman tasarrufu yanında kural kümesinin sürekli olarak güncellenmesini sağlayacaktır.

Kaynaklar

- Akgöbek, Ö. (2003), "Endüktif Öğrenmede Otomatik Bilgi Kazancı İçin Yeni Algoritmalar", Doktora Tezi, Sakarya Üniversitesi, Fen Bilimleri Enstitüsü, Sakarya.
- Alpaydın, E. (2000), "Zeki Veri Madenciliği: Ham Veriden Altın Bilgiye Ulaşma Yöntemleri", Bilişim 2000 Eğitim Semineri, İstanbul.
- An, A. (2003), "Learning Classification Rules From Data", *Computers and Mathematics with Applications* 45, 737-748.
- Arciszewski, T. (1996), "Learning engineering: The Key to Automatic Knowledge Acquisition", *Artificial Intelligence for Engineering Design, Analysis and Manufacturing*, Cambridge University Press, Cambridge, MA.

- Blake, C.L., Merz, C.J. (1994), “UCI Repository of Machine Learning Databases”, [<http://ftp.ics.uci.edu/pub/ml-repos/machine-learning-databases/>]. Irvine, CA: University of California, Department of Information and Computer Science.
- Bramer, M. A. (2000), “Inducer: A Rule Induction Workbench for Data Mining”, In: *IFIP World Computer Congress Conference on Intelligent Information Processing*, Beijing, Publishing House of Electronics Industry, 499-506.
- Chen M., Huang, C., Chen, K., Wu, H. (2005), “Aggregation of Orders in Distribution Centers Using Data Mining”, *Expert Systems with Applications* 28, 453–460.
- Forsyth, R. (1989), *Machine Learning - Principle and Technology*, Chapman and Hall, London.
- Klinkenberg, R. (1996), Rule Set Quality Measures For Inductive Learning Algorithms, M.S. Thesis, University of Missouri – Rolla.
- Lavington, S., Dewhurst, N., Wilkins, E., Freitas, A. (1999), “Interfacing Knowledge Discovery Algorithms to Large Database Management Systems”, *Information and Software Technology* 41, 605–617.
- Michalski, R.S., Kodratoff, Y. (1990), “Research in Machine Learning: Recent Progress, Classification of Methods, and Future Directions”, *Machine Learning* 3, Morgan Kaufmann, San Mateo, CA, 3-30.
- Pham D. T., Dimov S. S. (1997a), “An Efficient Algorithm for Automatic Knowledge Acquisition”, *Pattern Recognition* 30 (7), 1137-1143
- Pham, D. T., Dimov, S. S. (1997b), “The RULES-4 Incremental Inductive Learning Algorithm”, *Applications of Artificial Intelligence in Engineering XII*, R.A. Adey G. Rzevski and R. Teti (Eds), Computational Mechanics Publications, Southampton, Boston, 163-166.
- Pham, D. T., Dimov S. S., Salem Z. (2000), “Technique for Selecting Examples in Inductive Learning”, *ESIT 2000*, Aachen, Germany.
- SAS Institute Inc. (1999), The Data Mining Challenge: Turning Raw Data Into Business Gold. www.sas.com/software/data_mining/.
- Sever H., Oğuz, B., (2002), “Veritabanlarında Bilgi Keşfine Formal Bir Yaklaşım, Kısım I: Eşleştirme Sorguları ve Algoritmalar”, *Bilgi Dünyası*, 3 (2), 173-204.
- Thrun, S.B., Bala, J., Bloedorn, E., Bratko, I., Cestnik, B., Cheng, J., De Jong, K., Dzeroski, S., Fahlman, S.E., Fisher, D., Hamann, R., Kaufman, K., Keller, S., Kononenko, I., Kreuziger, J., Michalski, R.S., Mitchell, T., Pachowicz, P., Reich, Y., Vafaie, H., Welde, W. V., Wenzel, W., Wnek, J., Zhang, J. (1991), “The MONK’s Problems: A Performance Comparison of Different Learning Algorithms”, Carnegie Mellon University, CMU-CS-91-197.
- Tolun, M. R., Abu-Soud S.M. (1998), “ILA: An Inductive Learning Algorithm For Rule Extraction”, *Expert Systems with Applications* 14, 361-370.
- Tolun, M. R., Sever, H., Uludağ, M. (1998), “Improved Rule Discovery Performance on Uncertainty”, *The Second Pacific-Asia Conference on Knowledge Discovery and Data Mining (PAKDD-98)*, Melbourne, Australia.
- Vahaplar, A., İnceoğlu, M. M. (2001), “Veri Madenciliği ve Elektronik Ticaret”, *inet-tr2001 Türkiye’de internet konferansları VII*, İstanbul.
- Wang, X. (1997), “Inductive Learning Algorithms”, Ph.D. Thesis, University of Wales Cardiff.
- Wu, X. (1998), “Rule Induction with Extension Matrices”, *Journal of the American Society for Information Science* 49 (5), 435-454.
- Wu, X., Zhang C., Zhang, S. (2005), “Database Classification for Multi-database Mining”, *Information Systems* 30, 71–88.