

1. GİRİŞ

Veri madenciliği ve bulanık sınıflandırma, günümüzde birçok araştırmaya tabi tutulan ve pek çok alana uygulanan en önemli uygulamalardandır. Bulanık sınıflandırmanın görevi, problem için sınıflandırma yapabilecek kapasiteye sahip kurallar üretmektir. Bir veri yığını içerisinde anlamlı olan kuralları ve örüntüleri çıkarmak için birçok çalışma yapılmıştır. Örneğin Wu ve Chen [1], bulanık kuralları eğitim örneklerinden çıkaran ve üyelik fonksiyonlarını bloklayan bir algoritma önermişlerdir. Kaya ve Alhajj [2–4], bulanık bağıntı kurallarını bulmak için etkili yöntemler önermiştir.

Bulanık sınıflandırma dilsel terimlerle ifade edilen üyelik fonksiyonlarını kullanır. Üyelik fonksiyonları çeşitli şekillerde olmakla beraber en çok kullanılan üçgensel üyelik fonksiyonlarıdır. Bulanık sınıflandırmada üyelik fonksiyonlarının taban değerleri uzmanların bilgisiyile ya da otomatik olarak genetik algoritma gibi evrimsel süreçlerle öğrenme aşamasında elde edilir.

Genetik algoritma (GA), doğadaki evrim sürecini esas alan bir arama yönetimidir. Pek çok mühendislik problemine uygulanmakla beraber, uygulanma alanı gittikçe artıp ekonomi, müzik, hastalık, planlama gibi alanlarda da başarılı sonuçlar vermiştir.

1.1. Tezin Amacı

Bu tez çalışması, hastalık teşhisi yapmak için uygun bulanık kurallar kümesini çıkarmayı amaçlamıştır. Hastalık teşhisi için, bulanık sınıflandırma ve genetik algoritma birlikte kullanılmıştır. Hastalık verilerinin sahip olduğu nicel nesnelerin üyelik fonksiyonu taban değerlerinin ayarlanması yine bu tez çalışmasının konusudur.

Bulanık sınıflandırma yapılırken hem bireysel hem de çoklu-bulanık sınıflandırıcılar beraber incelenmiştir. Çoklu-bulanık sınıflandırıcılar ile elde edilen sınıflandırma doğruluğunun, bireysel sınıflandırıcılar ile elde edilen sınıflandırma doğruluğundan daha iyi olduğunu göstermek amaçlanmıştır.

Kullanılan çok-amaçlı genetik algoritmada üç amaca göre çözüm aranır. Amacımız kural sayısı az, anlaşılabilir ve sınıflandırma doğruluğu yüksek sınıflandırıcılar elde etmektir. Çok-amaçlı genetik algoritmalarda sayısal bir değere sahip uygunluk fonksiyonu yerine daha çok kullanıcının tercihi bırakılan bir uygunluk dizisi mevcuttur.

Tez Taslağı

İkinci bölümde tezde kullanılan temel kavramlar hakkında bilgi verilmiştir. Üçüncü bölümde sınıflandırma kavramı, sınıflandırma yöntemleri ve bulanık sınıflandırma kuralları üzerinde durulmaktadır. Dördüncü bölümde çok amaçlı genetik algoritmalarından bahsedilmiştir. Beşinci bölümde çoklu bulanık sınıflandırıcıların üretilmesi için GA kullanımı ve geliştirilen uygulama yazılımının detayları bulunmaktadır. Altıncı bölümde geliştirilen uygulama yazılımı ve çoklu bulanık sınıflandırıcılar, yedinci bölümde de tez çalışması ile elde edilen sonuçlar verilmektedir.

2. TEMEL KAVRAMLAR

2.1. Veri Madenciliği

Veri madenciliği, organizasyonların karar aşamaları için yeni bilgiler üreten ya da gelecekle ilgili tahminler ve planlar yapmamızı sağlayan bir dizi teknikler ve anlayışlar bütünü olarak tanımlanmaktadır [5].

Karar aşamalarında çok kritik bazı bilgiler vardır ki, sonuçların etkileri bu bilgilerin doğruluğuyla orantılıdır. Birçok durumda cevabını tam olarak veremediğimiz sorular doğrultusunda karar verebiliriz. Müşterilerimizin ilgi alanları, bize karşı olan bakış açıları, rakip firmalara olan ilgileri, markalarımıza olan bağlılıkları, gelir düzeyleri gibi bilgiler onlara sağladığımız mal ve hizmetlerin kalitesi üzerinde çok net etkiler yapacaktır. Bu tür bilgiler teorik olarak her ne kadar sistemlerimizde kayıt altında olsa da, kullanılabilir bir şekilde açık ve net cevaplara ulaşabilmemiz mevcut kullanımdaki sistemlerle neredeyse imkânsız denecek kadar zordur.

Eskiden süper marketteki kasa basit bir toplama makinesi olup, müşterinin o anda satın almış olduğu malların toplamını hesaplamak için kullanılırdı. Günümüzde ise kasa yerine kullanılan satış noktası terminalleri sayesinde bu hareketin bütün detayları saklanabiliyor. Saklanan bu binlerce malın ve binlerce müşterinin hareket bilgileri sayesinde her malın zaman içindeki hareketleri ve eğer müşteriler bir müşteri numarası ile kodlanmışsa bir müşterinin zaman içindeki verilerine ulaşmak ve analiz etmek olasıdır.

Veri kendi başına değersizdir [6] ve istenilen amaç doğrultusunda bir bilgidir. Bilgi bir amaca yönelik işlenmiş veridir. Veriyi bilgiye çevirmeye veri analizi denir. Bilgi de bir soruya yanıt vermek için veriden çıkarılır. Veri sadece sayılar veya harflerden oluşmaz. Veri, sayı ve harfler ile onların anlamından oluşur, bu şekildeki veriye meta veri denir.

Çok büyük veri yığınları altında saklı olan bu bilgilere ulaşmak için uzun yıllar boyunca yapılan çalışmaların neticesinde bir dizi yöntem geliştirilmiştir. Veri madenciliği uzun yıllardır, özellikle batı ülkelerinde üzerinde çalışılan bir konu olmasına rağmen, gerçek hayatta yazılım endüstrisinin son yıllarda üretmiş olduğu ileri teknoloji ürünü yazılımlar ile kullanılmaya başlanmıştır [7].

Veri madenciliği Biyomedikal ve DNA veri analizlerinde, finans veri analizlerinde, perakende satış verilerinde, telekomünikasyon endüstrisinde, astronomi ve birçok alanda uygulanmaktadır [5,8]. Bu uygulamaların başlıcaları için aşağıda örnek uygulamalar verilmiştir [6].

Bağıntı: “Çocuk bezi alan müşterilerin %30’u bira da satın alır.”

Sepet analizinde (basket analysis) müşterilerin beraber satın aldığı malların analizi yapılır. Buradaki amaç mallar arasındaki pozitif veya negatif korelasyonları bulmaktır. Çocuk bezi alan müşterilerin mama da satın alacağını veya bira satın alanların cips de alacağını tahmin edebiliriz. Ancak otomatik bir analiz bütün olasılıkları göz önüne alır ve kolayca düşünülemeyecek olan çocuk bezi ve bira arasındaki bağıntıları da bulur.

Sınıflandırma: “Genç kadınlar küçük araba satın alır. Yaşlı, zengin erkekler büyük, lüks araba satın alır.”

Amaç bir malın özellikleri ile müşteri özelliklerini eşlemektir. Böylece bir müşteri için ideal ürün veya bir ürün için ideal müşteri profili çıkarılabilir. Örneğin bir otomobil satıcısı şirket geçmiş müşteri hareketlerinin analizi ile yukarıdaki gibi iki kural bulursa genç kadınların okuduğu bir dergiye reklâm verirken küçük modelinin reklâmını verir.

Regresyon: “Ev sahibi olan, evli, aynı iş yerinde beş yıldan fazladır çalışan, geçmiş kredilerinde geç ödemesi bir ayı geçmemiş bir erkeğin kredi skoru 825’dir.”

Başvuru skorlamada (application scoring) bir finans kurumuna kredi için başvuran kişi ile ilgili finansal güvenilirliğini notlayan örneğin 0 ile 1000 arasında bir skor hesaplanır. Bu skor kişinin özellikleri ve geçmiş kredi hareketlerine dayanılarak hesaplanır.

Zaman İçinde Sıralı Örüntüler: “İlk üç taksitinden iki veya daha fazlasını geç ödemiş olan müşteriler %60 olasılıkla kanuni takibe gider.”

Davranış skoru (behavioral score), başvuru skorundan farklı olarak kredi almış ve taksitleri ödeyen bir kişinin sonraki taksitlerini ödeme/geciktirme davranışını notlamayı amaçlar.

Benzer Zaman Sıraları: “X şirketinin hisse fiyatları ile Y şirketinin hisse fiyatları benzer hareket ediyor.”

Amaç zaman içindeki iki hareket serisi arasında bağıntı kurmaktır. Bunlar iki malın zaman içindeki satış miktarları olabilir. Örneğin dondurma satışları ile kola satışları arasında pozitif, dondurma satışları ile salep satışları arasında negatif bir bağıntı beklenebilir.

İstisnalar (Fark Saptanması): “Normalden farklı davranış gösteren müşterilerim var mı?”

Amaç önceki uygulamaların aksine kural bulmak değil, kurala uymayan istisnai hareketleri bulmaktır. Bu da olası sahtekârlıkların saptanmasını (fraud detection) sağlar. Örneğin Visa kredi kartı için yapılan CRIS sisteminde bir yapay sinir ağı kredi kartı hareketlerini takip ederek,

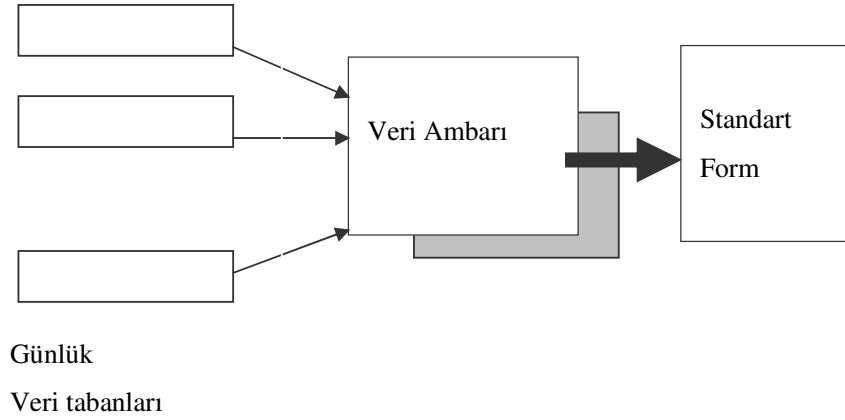
müşterinin normal davranışına uymayan hareketler için müşterinin bankası ile temasa geçerek müşteri onayı istenmesini sağlar.

Döküman Madenciliği: “Arşivimde (veya internet üzerinde) bu dokümana benzer hangi dokümanlar var?”

Amaç dokümanlar arasında ayrıca elle bir tasnif gerekmeden benzerlik hesaplayabilmektir (text mining). Bu genelde otomatik olarak çıkarılan anahtar sözcüklerin tekrar sayısı sayesinde yapılır.

2.1.1. Veri Ambarları ve Veriden Maden Elde Etme Süreci

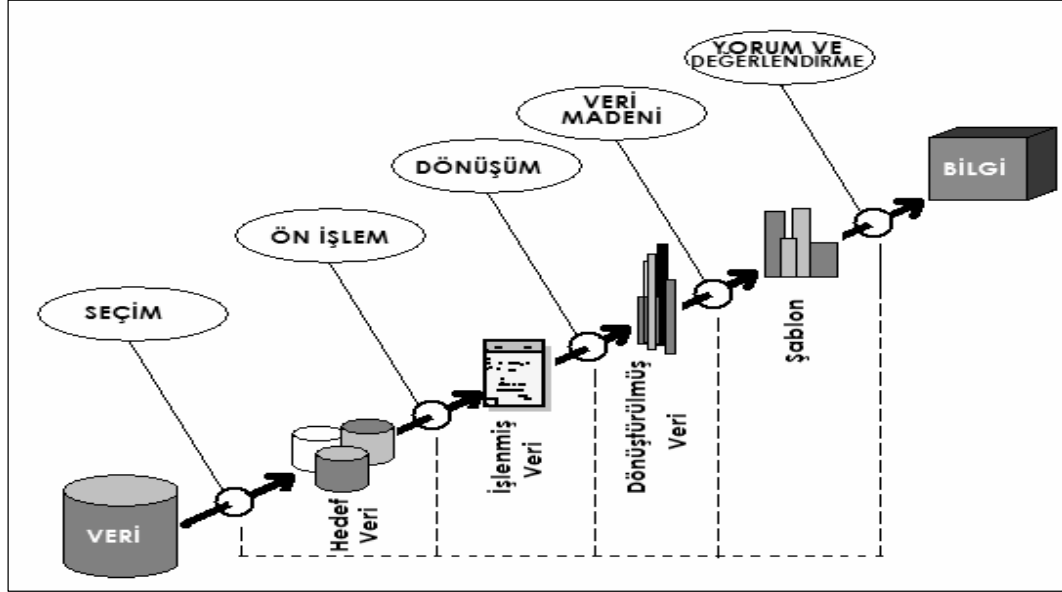
Veri madenciliği büyük miktarda veri inceleme amacı üzerine kurulmuş olduğu için veri tabanları ile yakından ilişkilidir. Gerekli verinin hızla ulaşılabilirlik şeklinde amaca uygun bir şekilde saklanması ve gerektiğinde hızla ulaşılabilmesi gerekir. Günümüzde yaygın olarak kullanılmaya başlanan veri ambarları günlük kullanılan veri tabanlarının birleştirilmiş ve işlemeye daha uygun bir özetini saklamayı amaçlar.



Şekil 2.1. Veri ambarlarının kullanımı

Şekil 2.1’de günlük veri tabanlarından istenen özet bilgi seçilerek ve gerekli ön işlemeden sonra veri ambarında saklanır. Ardından amaç doğrultusunda gerekli veri ambardan alınarak veri madenciliği çalışması için standart bir forma çevrilir.

Şekil 2.2’de veri için ön hazırlığın nasıl yapıldığını göstermektedir. Burada verinin istenilmeyen kısımlarının atılması, başka veri tabanından veri birleştirilmesi, verinin istenilen şekilde dönüştürülmesi ve verinin indirgenmesi aşamaları resmedilmektedir.



Şekil 2.2. Verinin hazırlanması

Şekil 2.2’de gösterilen aşamalar aşağıda açıklanmıştır:

Seçim: Verinin bazı kısıtlara göre seçilmesi, gerekli olan bir kısımdır. Birçok veri tabanı temelde birbirleriyle ilgisi olan verilerden oluşmasına rağmen, özelde önemli verilerin belirlenerek seçimlerinin yapılması gerekmektedir. Örneğin, araba sahibi olanlar derken, seçim için araba sahibi olma şartını öne sürmemiz bir seçim kısıtıdır.

Ön İşlem: Bu aşama, verinin ayıklanması işleminin yapıldığı kısımdır. Gerekli olmadığı düşünülen ve sorguyu yavaşlatabilecek belli bilgiler kaldırılır. Örnek verecek olursak, hamilelik üzerine çalışıldığında hasta cinsiyetini not almak gereksizdir. Bunun yanı sıra verinin birçok kaynaktan temininden dolayı istemediğimiz ya da uygun olmayan formatlarda bulunma ihtimali düşünülerek, belirlenen bir formata göre düzenlenmelidir. Cinsiyet verisi için, E veya K ya da 1 veya 0 seçilmesi gibi. Bu durumda veriler arasında ortak bir dil tanımlanmış olacaktır. Böylece sistemimiz hangi başlık altında hangi verinin bulunduğunu kolayca yorumlayabilecektir.

Dönüşüm: Veri sadece transfer edilmez aynı zamanda dönüştürülür. Buna örnek olarak çoğunlukla market araştırmalarında kullanılan müşteri portföyü gibi ekstra bilgilerin mevcut verilere eklenmesini gösterebiliriz. Bilgi kullanılabilir ve sevk edilebilir yapıya getirilir.

Veri Madeni: Bu evre ise veri içerisinden örüntülerin elde edilmesiyle ilgilidir.

Yorum ve Değerlendirme: Burada, sistem tarafından tanımlanan örüntüler, insanların karar vermesine yardımcı olmada, önceden haber vermeden ve konuların sınıflanmasında, veritabanının içeriğini özetlemede veya gözlenen olayların açıklamasında daha sonra kullanılabilecek bilgiler şekline dönüştürülür [9].

2.2. Genetik Algoritmalar

Genetik algoritmalar (GA) doğadaki evrim yöntemlerini kullanan bir arama yöntemidir. Genetik Algoritma tekniği, Michigan Üniversitesinde yer alan John Holland ve arkadaşlarının liderliğinde yapılan çalışmalar sonucu 1970’li yıllarda ortaya çıkmış ve 1975’de Holland, Doğal ve Yapay Sistemlerin Uygulanması adlı kitabını yayınlamıştır. Mekanik öğrenme konusunda çalışan Holland, Darwin’in evrim kavramından etkilenerek canlılarda yaşanan genetik süreci bilgisayar ortamında gerçekleştirmeyi düşündü. 1985’te Holland’ın öğrencisi olarak doktorasını veren David E. Goldberg adlı inşaat mühendisi, 1989’da konusunda bir klasik sayılan kitabı yayınlanana karar, genetik algoritmaların pek yararı olmayan bir araştırma konusu olduğu düşünüüyordu.

Bugün bilgisayar yöntemleri biyolojik değerlendirmeden esinlenerek evrimsel hesaplama olarak adlandırılan bir şemsiye altında gruplandırılmıştır.

Evrimsel hesaplamanın ana elemanlar aşağıda tanımlanmaktadır [10].

- (1) Değerlendirme stratejileri
- (2) Evrimsel programlama
- (3) Genetik algoritmalar

Bu üç tekniğin her biri doğal değerlendirmedeki gözlemlenen süreci taklit eder ve verilen problem için aday çözümlerin değerlendirme popülasyonlarıyla etkili arama motorlar sağlar. GA’lar genel olarak evrimsel hesaplama alanındaki en göze çarpan teknik olarak düşünülebilir.

GA, evrimsel programlamanın en yaygın ve en çok kullanılan dalıdır. Türkiye dâhil dünyada pek çok araştırmacı bu konuda çalışmaktadır. Son yıllarda genetik algoritmalara ilgi büyüyerek artmaktadır.

GA, hem problem çözmek hem de modelleme için kullanılmaktadır. Günümüzde genetik algoritmaların uygulama alanları genişlemektedir. Bunlardan bazıları: Atölye Çizelgeleme, Yapay Sinir Ağları Tasarımı, Görüntü Kontrolü, Elektronik Devre Tasarımı, Optimizasyon, Uzman Sistemler, Paketleme Problemleri, Makine ve Robot Öğrenmesi, Gezgin Satıcı Problemi, Ekonomik Model Çıkarma sayılabilir [11]. Canlıların yapılarında var olan bir takım özellikler sanal ortamlarda taklit edilerek modeller geliştirilmeye ve bu modellerle de karşılaşılan problemlere çözümler bulunmaya çalışılmaktadır. GA, geleneksel sezgisel yöntemlerden daha etkili ve çözüm yaklaşımında yapılacak küçük değişikliklerle halledilebildiklerinden dolayı da daha esnekler. Bu sebeple, GA araştırmacıların ilgisini çekmektedir. Bilindiği üzere optimizasyondaki temel amaç optimal bir noktaya ulaşabilmek, daha doğrusu mümkün oldukça yaklaşımdır. Bunu gerçekleştirmek için bilinen pek çok klasik yöntem vardır. Bu yöntemlerin başarısı optimum noktaya ulaşp ulaşmadıkları veya ne kadar ulaşabildikleri ile ölçülür. Genetik algoritmalar, klasik optimizasyon algoritmalarından dört temel noktada ayrılır:

- GA, bir noktadan değil bir arama uzayını kullanarak aramaya başlar. Yani GA çalışmaya başladığında popülasyonda olası çözümler mevcuttur.
- GA, olasılık kurallarına göre çalışır.
- GA, her problem yapısına göre parametrelerin kodlanmış haliyle çalışır.
- GA, zor matematiksel işlemler kullanmaz. Her problemin kendi yapısına uygun olarak oluşturulan bir değerlendirme fonksiyonu kullanır.

GA'ların parametreleri; çaprazlama oranı, mutasyon oranı, popülasyon büyüklüğü, seçim, kodlama, çaprazlama ve mutasyon tipi gibi genel parametrelerdir. Çaprazlama oranı yüksek olmalıdır. Buna karşılık mutasyon oran da çok düşük olmalıdır.

Seçim için genellikle rulet tekerleği kullanılır, bunun yanı sıra rank seçimi, kararlı durum (steady state) ve elitizm gibi seçim yöntemleri de kullanılmaktadır.

GA yöntemi, yıllar boyu süregelen genetik mühendisliği ve biyoloji alanında yapılan çalışmalar sonucu ortaya atılmış bir tekniktir ve her sisteme uygulanamayabilir [12]. Standart bir GA yönteminin adımları aşağıdaki gibi verilebilir:

1. Başlangıç: n adet kromozom içeren popülasyonun oluşturulması (problemin potansiyel çözümü),
2. Uygunluk: her x kromozomu için uygunluğun $f(x)$ değerlendirilmesi,
3. Yeni popülasyon: Yeni popülasyon oluşuncaya kadar aşağıdaki adımların tekrar edilmesi,
 - 3.1. Seçim: İki ebeveyn kromozomun uygunluklarına göre seçimi (daha iyi uyum seçilme şansını artırır),
 - 3.2. Çaprazlama: Yeni bir fert oluşturmak için ebeveynlerin bir çaprazlama olasılığına göre çaprazlanması. Eğer çaprazlama yapılmazsa, yeni fert anne veya babanın kopyası olacaktır.
 - 3.3. Mutasyon: Yeni ferdin mutasyon olasılığına göre kromozom içindeki konumu (lokus) değiştirilir.
 - 3.4. Ekleme: Yeni bireyin yeni popülasyona eklenmesi.
4. Değiştirme: Algoritmanın yeniden çalıştırılmasında oluşan yeni popülasyonun kullanılması,
5. Test: Eğer sonuç tatmin ediyorsa, algoritmanın sona erdirilmesi ve son popülasyonun çözüm olarak sunulması.
6. Döngü: 2. adıma geri dönülmesi.

2.2.1. Genetik Operatörler

Kullanılan genetik operatörler, var olan popülasyon üzerine uygulanan işlemlerdir. Bu işlemlerin amacı daha iyi özelliğe sahip yeni nesiller üretmek ve arama algoritmasının alanını genişletmektir. Farklı uygulamalarda farklı operatörler kullanılmakla birlikte GA’da üç standart operatör kullanılır. Bu operatörler:

- Seçme ve Yeniden Üretim
- Çaprazlama
- Mutasyon
- Çaprazlama ve mutasyon olasılığı

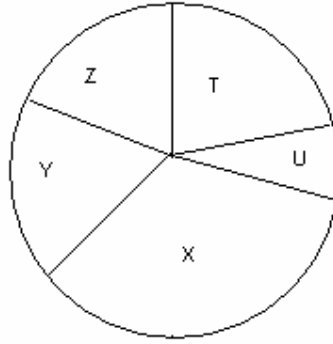
(i) Seçme ve Yeniden Üretim, Nesil Üretimi :

Genetik algoritmalarda bir sonraki nesli oluşturmak için iyi olan bireylerin seçilmesi gerekmektedir. Popülasyondan iyi olan genlerin seçimi için birkaç yöntem vardır. Bu yöntemler

içinden en çok kullanılanı, bireylerin uygunluk değerlerine bakarak en iyi uygunluk değerine sahip olan bireyleri seçmektir. Seçilen bireyler kendi aralarında çaprazlama işlemine tabi tutulur. Çaprazlama işleminden sonra elde edilen evlat bireyler daha kötü uygunluk değerine sahip bireylerin yerine yerleştirilir. Diğer seçim yöntemleri ise şunlardır:

- Rulet tekerleği seçimi
- Sıralı (rank) seçim
- Turnuva seçimi
- Kararlı durum seçimi
- Oransal seçim
- Seçkinlik (Elitizm)

Rulet Tekerleği Seçimi: Bu seçim yönteminde tüm bireylerin uygunluk değerleri bir tekerleğe yerleştirilir. Uygunluk değeri yüksek olan bireylerin seçilme şansı daha yüksek, uygunluk değeri kötü olan bireylerin seçilme şansı ise daha düşüktür. Yalnız burada amaç kötü uygunluğa sahip bireyleri tamamen yok etmek değil, onların da çaprazlamaya az da olsa katılma ihtimalinin olmasıdır.



Şekil 2.3. Rulet tekerleği uygunluk yerleşimi

$$P_{\text{seçilen}} = f_i / \sum_{i=1}^N f_i$$

Bağıntısıyla, bireyin seçilme olasılığı hesaplanabilir. Burada;

f_i : Bireyin popülasyondaki uygunluk değeri

f_k : popülasyondaki toplam uygunluk değeridir.

Şekil 2.3'deki rulet tekerleği yerleşiminde, x bireyinin seçilme şansı diğerlerinden daha yüksektir. Çünkü sahip olduğu uygunluk değeri diğer bireylerden daha iyidir.

Sıralı (rank) Seçim: Rulet seçimi eğer uygunluk çok fazla değişiyorsa sorun çıkartabilir. Örneğin en iyi kromozomun uygunluğu %90 ise diğer kromozomların seçilme şansı azalacaktır. Bunu önlemek için sıralı seçim kullanılabilir. Sıralı seçimde en kötü uygunlukta olan kromozoma 1 değeri sonrakine 2 değeri verilir ve böylelikle seçilmede bunlara öncelik tanınmış olur. En iyi kromozom ise N değerini alacaktır (N popülasyondaki kromozom sayısıdır). Bu şekilde kötü bireylerin de seçilme şansı artar fakat bu çözümün daha geç yakınsamasına neden olabilir.

Turnuva Seçimi: Turnuva seçiminde bir eşleştirme havuzu vardır ve rasgele seçilen genler bu eşleştirme havuzuna atılır.

Popülasyonda bulunan bireyler içerisinde en yüksek uygunluk değerine sahip olanlar eşleştirme havuzuna alınırlar ve geriye kalan bireyler içerisinde yeni bir turnuva seçimi daha yapılır. Turnuvanın boyutu kadar bu işleme devam edilir. Sonuçta havuzda oluşan bireylerin uygunluk değeri yükseltilmiş olur.

Kararlı Hal (Steady-State) Seçimi: Bu seçme yöntemi ise popülasyonda bulunan en iyi bireylerin kötü olan bireylerin yerine yerleştirilmesidir. Burada iyi olan bireyler herhangi bir işleme tabi tutulmaz, iyi olan birey varlığını koruyarak bir sonraki nesilde hayatını devam ettirir.

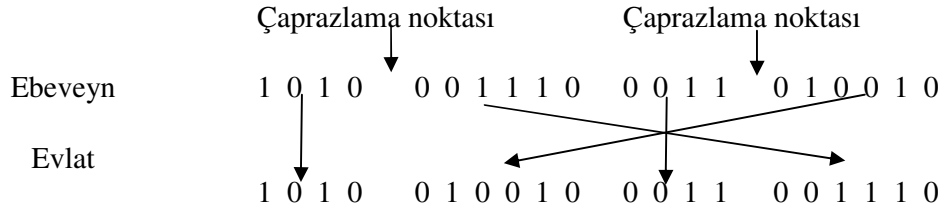
Oransal Seçim: Bu yöntemle, bir bireyden seçilecek kopya sayısı belirlenir. Bireyin popülasyondaki uygunluk değeri, popülasyonun ortalama uygunluk değerine bölünür. Elde edilen değer kadar o birey kopyalanarak yeni nesle aktarılır.

Seçkinlik (Elitizm): Çaprazlama ve Mutasyon yöntemleriyle yeni bir nesil oluştururken, en iyi kromozomları kaybetme olasılığımız vardır. Seçkinlik, en iyi kromozomların (ya da bir kısmının) ilk önce kopyalanıp yeni nesle aktarıldığı yöntemin adıdır. Seçkinlik GA'nın başarımını hızlı bir şekilde arttırabilir, çünkü bulunan en iyi çözümün kaybolmasını önler.

(ii) **Çaprazlama:** Çaprazlama operatörü GA'lardaki en önemli operatördür. Çözümün yapılar kullanılarak yeni bir çözüm oluşturulması esasına dayanır. Çaprazlama işlemi genel olarak ikili

dizilerin parçalarının deęiř tokuřu řeklinde gerekleřtirilir. Farklı uygulamalarda farklı kodlama yöntemleri kullanıldıęı için farklı aprazlama yöntemleri kullanılır. Bunlardan en ok kullanılanları ařaęıda verilmiřtir:

- Tek noktalı aprazlama,
- ok noktalı aprazlama
- Üniform aprazlama



řekil 2.4. aprazlama

Ama, ebeveyn kromozom genlerinin yerini deęiřtirerek ocuk kromozomları üretmek ve böylece var olan uygunluk deęeri daha yüksek olan kromozomlar elde etmektedir.

(iii) Mutasyon: Mutasyon GA’lardaki operasyonda karar verici olarak ikinci derecede rol oynar. Ama, var olan bir kromozomun genlerinin bir ya da bir kaçının yerlerini deęiřtirerek yeni kromozom oluřturmaktır. Yeniden ve sürekli yeni nesil üretimi sonucunda belirli bir süre sonra nesildeki kromozomlar birbirini tekrarlama konumuna gelebilir ve bunun sonucunda farklı kromozom üretimi durabilir veya ok azalabilir. İřte bu sebeple nesildeki kromozomların eřitlilięini arttırmak için kromozomlardan bazıları mutasyona uğratılır.



řekil 2.5. Mutasyon

(iii) aprazlama ve Mutasyon Olasılıęı: aprazlama olasılıęı ile bireyler yeni bireyleri oluřturmak için birbirleriyle eřleřtirilir. Eęer aprazlama yapılmazsa, evlat bireyler ebeveyn

bireylerin tıpatıp aynısı olacaktır. Mutasyon olasılığı ile yeni evlatlar üzerinde mutasyon işlemi yapılacaktır.

2.2.2. Uygunluk Değerlendirmesi

Genetik algoritma süreci, başlangıç popülasyonunun oluşturulmasıyla başlar ve bundan sonra evrim geçirir. Bu evrim süresince iyi bireyler uygunluk değerlerine göre ayırt edilirler. Uygunluk, potansiyel çözümün iyilik derecesini belirleyen bir kıstastır. Bir bireyin (genotip veya kromozom) uygunluk değeri, onun iyilik derecesini belirleyen bir pozitif sayıdır. Yani bütün uygunluk değerleri pozitif olmak zorundadır, negatif olmasına izin verilmez. Kromozom, kombinasyonel optimizasyon problemleri için bir çözüm oluşturduğunda, uygunluk değeri bu çözümün maliyetini gösterir. Minimizasyon problemlerinde, düşük maliyetli çözümlerle daha uygun olan bireylerden seçilir. Uygunluğu iyi olan bireylerin seçilme olasılıkları yüksek olacaktır ve bu sayede bu bireylere genetik operatörlerin uygulanması ihtimali de yüksek olacaktır.

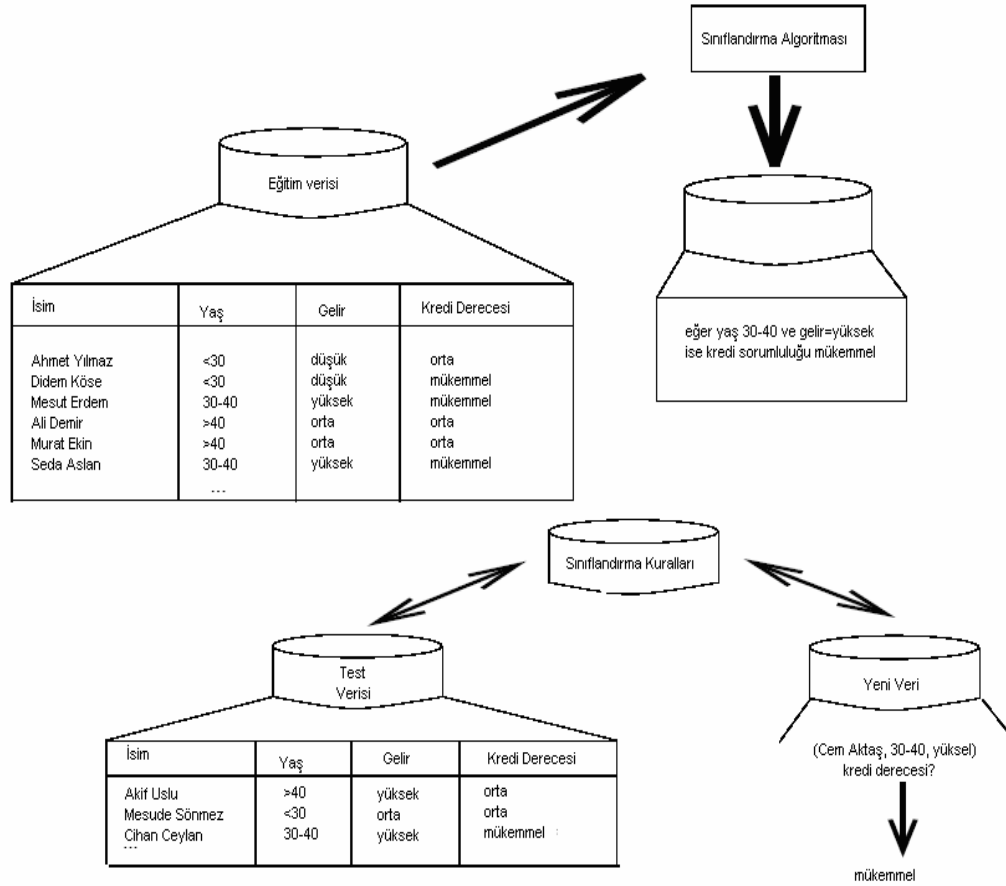
Genetik algoritma doğal olarak uygunluğu maksimum yaparak çalışır. birçok optimizasyon probleminde amaç maliyeti minimum yapmaktır. Bu nedenle maliyet fonksiyonu bir uygunluk fonksiyonu ile tasarlanır.

Uygunluk fonksiyonu genelde probleme özgüdür. Bir problem için oluşturulan uygunluk fonksiyonu başka bir problemde kullanılamaz. Problem incelenerek uygunluk fonksiyonunun ne olacağına karar verilir.

3. SINIFLANDIRMA

3.1. Sınıflandırma ve Tahmin Kavramları

Veritabanları akıllı kararlar verebilmek için kullanılacak zengin gizli bilgiler içerirler [13]. Sınıflandırma ve tahmin, önemli veri sınıflarını tanımlamak ve gelecekteki trendi tahmin etmeye yarayacak modelleri oluşturmak için kullanılan veri analizi türleridir. Sınıflandırma kategorik etiketleri tahmin etmekte kullanılırken, tahmin modelleri ise sürekli değer alan fonksiyonlar için kullanılırlar. Örneğin, bir sınıflandırma modeli banka kredilerinin güvenli mi yoksa riskli mi olduğunu kategorize etmek için kullanılabilir. Bir tahmin modeli de potansiyel müşterilerin gelir ve meslek bilgilerini kullanarak bilgisayar donanım harcamalarını tahmin etmek için kullanılabilir. Nörobiyoloji, istatistik, uzman sistemler ve makine öğrenmesi alanlarında birçok sınıflandırma ve tahmin algoritmaları kullanılmıştır.



Şekil 3.1. Sınıflandırma Aşamaları

Veri sınıflandırması iki aşamalı bir süreçtir. Birinci aşamada, önceden belirlenmiş veri sınıfları ve kavramlar kümesini ifade eden modeller oluşturulur. Model nesnelerle tanımlanan veritabanı parçacıkları analiz edilerek oluşturulurlar. Her parçacığın önceden tanımlanmış, bir nesne tarafından belirlenen bir sınıfa ait olduğu varsayılır. Sınıflandırma alanında veri parçacıkları örnekler ve nesnelerdir. Eğitim veri kümesinde yer alan parçacıklar toplu olarak analiz edilerek model oluşturulur. Eğitim veri kümelerini oluşturan parçacıklara eğitim örnekleri denir ve örnek popülasyondan rasgele elde edilirler. Her bir eğitim örneğinin sınıf etiketi bilindiğinden bu aşamada eğitimi öğrenmedir.

Tipik olarak, öğrenilmiş model sınıflandırma kuralları, karar ağaçları ve matematiksel formüllerle ifade edilir. Örneğin, müşterinin kredi bilgileri kullanılarak, kredi puanları sınıflandırma kuralları öğrenilebilir. Bu kurallar ileride karşılaşılabilecek veri kümelerini sınıflandırmak ve veritabanı içeriğini daha rahat anlamak için kullanılırlar.

İkinci aşamada model sınıflandırmak için kullanılır. Öncelikle sınıflandırıcının tahmin doğruluğu hesaplanır. Bu metot sınıflanarak etiketlenmiş örnekleri kullanan basit bir metottur. Bu örnekler eğitim örneklerinden bağımsız olarak ve rasgele seçilirler. Bir modelin bir test kümesindeki doğruluk oranı, model tarafından doğru olarak sınıflandırılan örnek test kümelerinin yüzdesel oranıdır. Her bir test örneği için bilinen sınıf etiketi bu örnek için öğrenilmiş modelin sınıf tahminiyle karşılaştırılır.

Sınıflandırma ve tahmin için veri hazırlanır. Aşağıda yer alan adımlar uygulanarak sınıflandırma ve tahmin sürecinin doğruluk ve etkinliği geliştirilebilir.

Veri Temizliği: Veride yer alan kirliliğin giderilmesi ve eksik değerlerin giderilmesi anlamına gelir. Bütün sınıflandırma algoritmalarının veri kirliliği ve eksik veriler konusunda bazı metotlara sahip olmasına rağmen, bu adım öğrenme sürecinde karmaşıklığı azaltır.

İlişki Analizi: Veri içinde yer alan nesnelerin birçoğu sınıflandırma veya tahmin ile ilgisiz olabilir. Örneğin, banka kredi başvurusunun haftanın hangi günü yapıldığı kaydı başvurunun sonucu ile ilgisizdir. Ayrıca, diğer nesneler gereksiz olabilir. Bu nedenle, gereksiz ve ilişkisi olmayan nesnelerden kurtulmak için ilişki analizi yapılabilir. Makine öğrenmesinde bu adım özellik seçilmesidir. Aksi halde bu nesnelerin dahil edilmesi durumunda öğrenme süreci yavaşlayabilir ve yanlış yönlendirilebilir.

Veri Dönüşümü: Veri genelleştirilerek daha yüksek seviyede kavramlarla ilişkilendirilebilir. Kavram hiyerarşileri bu amaçla kullanılabilirler. Bu sürekli değer

alan nesneler için kısmen faydalıdır. Örneğin, gelir nesnesinin sayısal değerleri düşük, orta ve yüksek gibi kesikli sınıflara genelleştirilebilirler. Benzer şekilde, nominal değerler alan sokak gibi nesneler, şehir gibi daha yüksek kavramlara genelleştirilebilirler. Genelleştirme orijinal veriyi sıkıştırdığı için daha az işlemle öğrenme süreci yürütülebilir.

Normalizasyon: Bir nesneye ait bütün değerleri belirli bir tanımlanmış aralıkta yer alacak şekilde boyutlandırılmasıdır. Uzaklık ölçümü metotları nesnelerin başlangıçta çok yüksek değerler almasını başlangıçta verilen daha küçük aralık değerlerini kullanarak engeller.

Sınıflandırma ve tahmin metotları aşağıda yer alan kıstaslara göre karşılaştırılabilirler:

Tahmin Doğruluğu: Modelin yeni veya daha önce hiç görülmemiş bir veriye ait sınıf değerini doğru olarak tahmin etme yeteneğini ifade eder.

Hız: Modelin kullanımı ve oluşturulması ile ilgili işlem maliyetlerini ifade eder.

Sağlamlık: Modelin kirli veya eksik değerler içeren verilerle doğru tahminler yapabilme yeteneğidir.

Ölçeklenirlik: Çok sayıda veri ile çalışırken verimli olma yeteneğidir.

Anlaşılabilirlik: Öğrenilmiş modelin anlaşılabilirliği ve kavrayışını ifade eder.

3.2. Sınıflandırma Yöntemleri

3.2.1. Karar Ağacı ile Sınıflandırma

Karar ağacı her bir düğümün nesne için yapılan bir testi temsil ettiği, her bir dalın testin sonucunu ifade ettiği ve yaprakların sınıfları veya sınıf dağılımını ifade ettiği, ağaç yapısında bir akış şemasıdır.

Karar ağacı metodunun temel algoritması, yukarıdan aşağıya özyinelemeli ilerleyen böl ve yönet mantığını kullanan bir algoritmadır. Algoritmanın temel stratejisi aşağıda verilmiştir.

- Ağaç eğitim örneklerini temsil eden bir düğüm ile başlar.
- Eğer örnekler, aynı sınıfa aitse, düğüm bu sınıf ile adlandırılan bir yaprak olur.
- Aksi takdirde, algoritma entropi tabanlı, bilgi kazanımı olarak bilinen ve örnekleri en iyi şekilde sınıflara ayıran, nesneleri seçen bir ölçü kullanır. Bu nesne düğüm

geçişlerinde test veya karar nesnesi olarak kullanılır. Algoritmanın bu sürümünde, tüm nesneler kategorik yani ayrık değerler alır. Sürekli değerlere sahip nesneler ayrık değerlere dönüştürülmelidir.

- Test nesnesinin her bilinen değeri için bir dal oluşturulur ve örnekler uygun şekilde parçalanır.
- Algoritma, her bölmeleme için karar ağacından özyinelemeli olarak aynı işlemi tekrarlar. Bir düğümde bir nesne ortaya çıktığında, düğümün alt kısmı için dikkate alınmasına gerek yoktur.
- Özyinelemeli bölmeleme sadece aşağıdaki durumlardan herhangi biri doğru olduğu zaman durur:
 - o Verilen bir düğüme ait tüm örnekler aynı sınıfta yer alırsa.
 - o Örneklerde bölmelenmeyen hiçbir nesne yoksa bu durumda çoğunluk oylaması kullanılır. Bu oylama, verilen düğümün bir yaprağa dönüştürülmesi ve örneklerde yer alan çoğunluk sınıfı adıyla etiketlenmesini içerir. Alternatif olarak, örneklerin sınıf dağılımları saklanabilir.
 - o Dal test nesnesi için hiçbir örnek yoksa örneklerdeki çoğunluk sınıfı ile bir yaprak oluşturulur.

3.2.2. Bayes Sınıflandırma

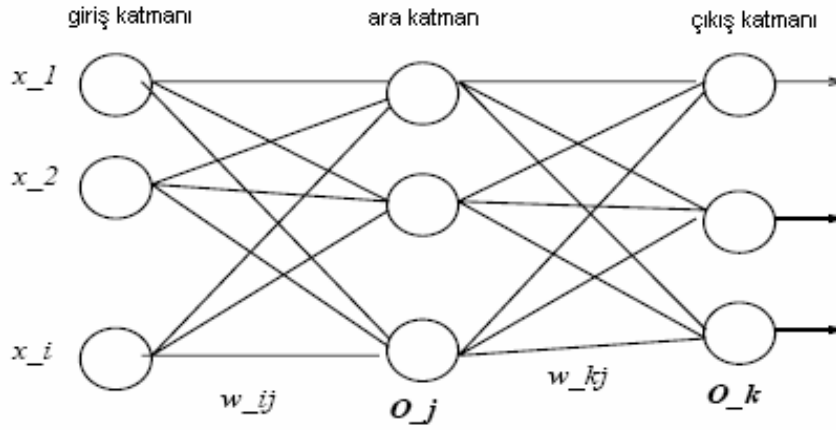
Bayes sınıflandırma istatistiksel bir sınıflandırmadır. Onlar olası sınıf üyeliklerini tahmin edebilirler. Bayes sınıflandırma Bayes teoremini temel alır. Bayes sınıflandırma nöral sınıflandırıcılar ve karar ağaçlarıyla performans olarak karşılaştırıldığında, Bayes sınıflandırıcıların geniş veri kümelerine uygulandığında yüksek doğruluk ve hızla sahip olduğu görülür.

3.2.3. Geri Yayılımlı Nöral Ağlar ile Sınıflandırma

Geri yayılımlı nöral ağ bir öğrenme algoritmasıdır. Nöral ağlar nörobiyolojiden esinlenerek oluşmuştur. Kabataslak olarak bir nöral ağ, her bağlantının bir ağırlık değerine sahip olduğu, bağlantılı giriş/çıkış birim kümelerinden oluşur. Öğrenme aşamasında, ağ, girilen örnekleri doğru sınıflandırabilmek için bağlantılardaki ağırlıkları ayarlamayı öğrenir. Nöral ağ öğrenmesi, düğümler arasındaki bağlantılardan dolayı bağlantısal öğrenme olarak da bilinir. Nöral ağlar uzun eğitim zamanı gerektirir. Bu yüzden uygun problemlerde kullanılabilir. Nöral ağ öğrenmesi başlarken bağlantılara bir başlangıç değeri verilir. Nöral ağlar veri gürültüsüne

karşı yüksek toleransa sahiptir. Son zamanlarda öğrenmiş nöral ağlardan kural çıkarmak için bazı algoritmalar geliştirilmiştir.

En yaygın nöral ağ algoritması ise geri yayımlı nöral ağıdır. Aşağıda ileri-beslemeli bir nöral ağ görülmektedir. Burada $X=(x_1, x_2, \dots, x_i)$ girişleri, w_{ij} i katmanından j katmanına olan bağlantının ağırlığını temsil eder. w_{ij} ağırlığı bir sonraki katman olan ara katmanı besler. Ara katmanın çıkışları başka ara katmanları besleyebilir. Ara katman sayısı isteğe bağlı olarak artırılabilir olduğu halde pratikte genelde tek ara katman kullanılır. Son ara katmanın çıkışı, çıkış katmanına giriş olarak verilir ve çıkış katmanının sonucu ise verilen örnek için ağın tahminini ifade eder.



Şekil 3.2. 3 Katmanlı bir nöral ağ

3.2.4. k-En Yakın Komşuluk Sınıflandırıcıları

Bu sınıflandırıcılar benzerlik (örnekleme) öğrenmesini temel alır. Öğrenme örnekleri n boyutlu sayısal nesnelerle ifade edilir. Her örnek n boyutlu uzayda bir nesneyi ifade eder. Böylece tüm öğrenme örnekleri n boyutlu bir örüntü uzayında saklanır. Bilinmeyen bir örnek geldiği zaman, k-en yakın komşuluk sınıflandırıcısı, gelen örneğe en yakın olan k eğitim örneğini bulmak için örüntü uzayını arar. Bu yakınlık ölçüsü, 2 nokta arasındaki mesafeyi bulan öklit uzaklığı formülüyle bulunur.

$X=(x_1, x_2, \dots, x_n)$, $Y=(y_1, y_2, \dots, y_n)$ olmak üzere, X ve Y arasındaki uzaklık öklit uzaklığıyla

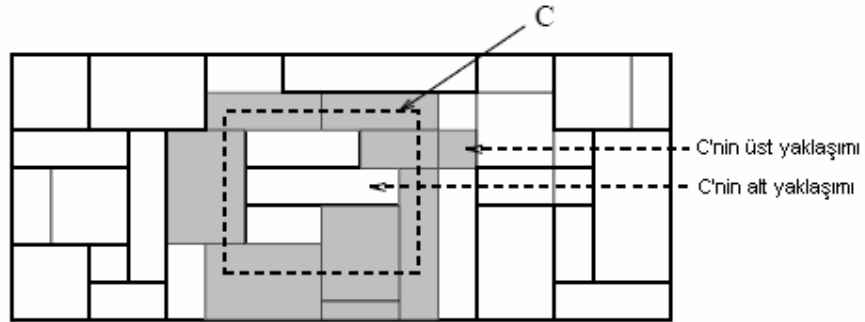
$$d(X, Y) = \sqrt{\sum_{i=1}^n (x_i - y_i)^2}$$

formüle edilir.

Bilinmeyen örnek, onun k en yakın komşulukları içerisinde en yaygın olan sınıfa dahil edilir.

3.2.5. Kaba (rough) Küme Teorisi:

Bu teorem, eksik veya gürültülü veri içerisindeki yapısal ilişkileri bulup sınıflandırmak için kullanılabilir. Ayırık değerli nesnelere uygulanır. Sürekli değerlere sahip olan nesneler kullanılmadan önce ayrıştırılmış olmalıdır. Kaba küme teorisi, verilen eğitim verisi içerisinde, denk sınıfları oluşturmayı esas alır. Gerçek dünya problemlerinde, bazı sınıflar uygun nesneler açısından ayırt edilemeyebilir. Şekil 3.3'te kaba küme teorisi yaklaşımı görülmektedir. C sınıfının örneklerinin kaba küme yaklaşımları görülmektedir. Dikdörtgensel alanlar denk sınıfları temsil eder. Verilen bir C sınıfı için kaba küme tanımlaması 2 küme tarafından tanımlanır. Bu iki küme, C 'nin üst ve alt yaklaşımlarıdır. C 'nin alt yaklaşımı, belirsizlik içermeden C 'ye ait olan tüm kesin veri örneklerinden oluşur. Üst yaklaşım, C 'ye ait olup olmadığı tanımlanamayan belirsiz veri örneklerinden oluşur. Şekil 3.3'teki üst ve alt yaklaşım kümeleri çıkarıldıktan sonra bir karar kural kümesi oluşturulur.



Şekil 3.3. Kaba küme teorisi için karar kuralları

3.3. Bulanık Sınıflandırma ve Bulanık Kurallar

3.3.1. Bulanık Küme Teorisi

Zadeh 1965'te [14], bulanık küme teorisini ilk öneren bilim adamlarıdır. U bir dilsel evren olsun. $U=\{U_1, U_2, \dots, U_n\}$. A, dilsel evrenin bulanık bir kümesi ise şöyle temsil edilebilir.

$$A = \sum_{i=1}^n \mu_A(u_i) / u_i \\ = \mu_A(u_1) / u_1 + \mu_A(u_2) / u_2 + \dots + \mu_A(u_n) / u_n$$

$1 \leq i \leq n$ için, μ_A A bulanık kümesinin üyelik fonksiyonunu, $\mu_A(u_i)$ A bulanık kümesindeki u_i 'nin üyelik derecesini gösterir. (+) sembolü, birleşim operatörü ve '/' ayırıcısı gösterir.

Tanım: A ve B sırasıyla μ_A ve μ_B üyelik fonksiyonlu U dilsel evreninin iki bulanık kümesi olsun. A ve B bulanık kümelerinin birleşimi şöyle tanımlanır:

$$\mu_{A \cup B}(u) = \max \{ \mu_A(u), \mu_B(u) \} \quad \forall u \in U$$

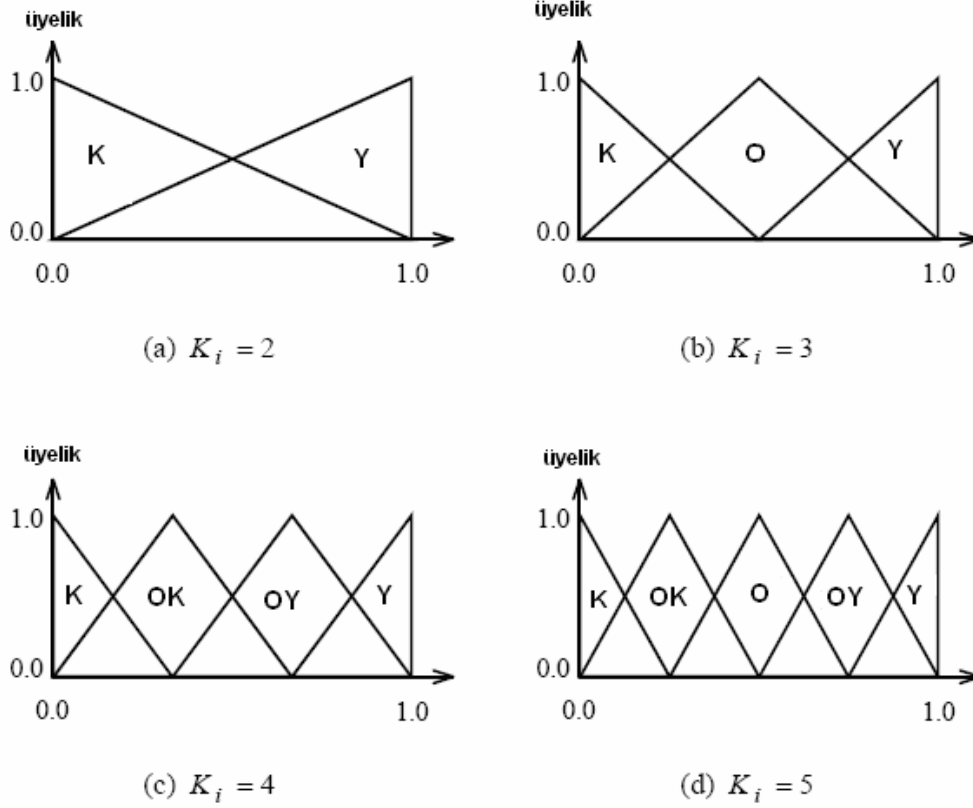
A ve B'nin kesişimi, $A \cap B$ olarak gösterilir ve şöyle tanımlanır:

$$\mu_{A \cap B}(u) = \min \{ \mu_A(u), \mu_B(u) \} \quad \forall u \in U$$

A'nın tümleyeni $\bar{A}$ olarak gösterilir ve şöyle tanımlanır.

$$\mu_{\bar{A}}(u) = 1 - \mu_A(u) \quad \forall u \in U$$

Şekil 3.4'te çeşitli bulanık kümeler görülmektedir. (a)'da "küçük" ve "yüksek" olmak üzere 2, (b)'de "küçük", "orta" ve "yüksek" olmak üzere 3, (c)'de farklı olarak "orta küçük" ve "orta yüksek" eklenerek 4 ve (d)'de ise 5 tane bulanık küme vardır. Bulanık küme sayısı isteğe göre belirlenebilmektedir.



Şekil 3.4. Çeşitli sayıda bulanık kümeler

3.3.2. Bulanık Sınıflandırma

Veri madenciliği teknikleri veri analizi ve keşfiyle yararlı örüntüleri bulmaya alışıktır. Bu yüzden veri madenciliği tekniklerinin içine sınıflandırma işlemleri dahil edilebilir. Bulanık sınıflandırma bulanık mantığın önemli uygulamalarından biridir. Bulanık sınıflandırma sistemleri sınıflandırma problemlerinin içerdiği anlam bulanıklığı ve belirsizliği gibi algısal kararsızlıkları işletebilir.

Bir bulanık sınıflandırma sistemini başarmadaki en önemli görev bir belirli sınıflandırma problemi için uygun bulanık kurallar kümesini bulmaktır.

Bulanık sınıflandırma sistemleri, sınıflandırma problemlerindeki anlam belirsizliği gibi algısal kararsızlığı yapabilir. Bulanık kuralların kümesini bulmada iki yaklaşım vardır.

Birinci yaklaşım, uzmanlardan bilgiyi elde etme ve onların bilgisini doğrudan bulanık kurallara çevirmedir. Bununla birlikte bilgi edinme ve doğrulama işlemi zordur ve zaman alır. Uzmanın, bilgisinin açık ve doğru olmama ihtimali vardır.

Diğer yaklaşım, örneklerden veya örnek durumlardan otomatik çıkarılabilen bilginin bir makine öğrenmesi işleminden geçirilerek bulanık kuralları üretmektir.

Castro ve Zurita [15] bulanık sistemlerde endüktif öğrenme algoritması sunmuşlardır. Chen ve Yeh [16] boş değerleri tahmin etmek için, ilişkisel veri tabanı sistemlerinden bulanık kurallar üretmek için bir metot önermişlerdir. Hayashi ve Imura [17] bir eğitilmiş sinir ağından bulanık if-then kurallarını otomatik çıkarmak için bir metot sunmuşlardır. Hong ve Lee [18] eğitilmiş örneklerden üyelik fonksiyonları ve bulanık kuralları çıkarmak için bir algoritma sunmuşlardır. Klawonn ve Kruse [19] verilen datadan bir bulanık kontrolör bloklamak için bir metot sunmuşlardır. Nozaki, Ishibuchi ve Tanaka [20] sayısal verilerden bulanık kurallar üretmek için bir sezgisel metot sunmuşlardır. Wang ve Mendel [21] örneklerden öğrenmeyle bulanık kuralların üretimi için bir algoritma sunmuşlardır. Wu ve Chen [1] eğitim örneklerinden, bulanık kurallar ve üyelik fonksiyonları bloklamak için bir metot önermişlerdir. Yuan and Zhuang [22] eğitim örneklerinden bulanık sınıflandırma kurallarının üretimi için bir GA sunmuşlardır.

Sınıflandırma metotları uygun bilgiden sınıflandırma kuralları elde etmeden oluşan bir öğrenme aşamasıyla karakterize edilir.

Bu aşamada tümevarım ve tümdengelim yaklaşımları kullanılır. Tümevarım yaklaşımıyla, sınıflandırma kuralları örneklerden elde edilir ve her örnek iyi bilinen bir sınıfa aittir. Bu yaklaşımın amacı sınıflara yeni örnekler atamak için sınıflandırma kuralları üretmektir. Tümevarım yaklaşımını kullanan birçok metot vardır. Örnek olarak k-en yakın komşuluk kuralı, Bayes teknikleri, diskriminant analiz, nöral network ve karar ağacı verilebilir [23]. Tümdengelim yaklaşımıyla kurallara bir uzman veya kural oluşturucu tarafından bir öncelik verilir. Bu kurallar sayesinde nesnelerin hangi sınıfa atanacağı belirlenir. Uzman sistemler ve kaba küme (rough set) metotları bu tip yaklaşımları içerir [23].

Bir bulanık sınıflandırma sisteminde, bir durum veya bir nesne, onun özelliklerinin dilsel değerlerine dayanan bulanık kuralların bir kümesinin uygulanmasıyla sınıflandırılabilir. Uygun bulanık kuralların nasıl elde edilebildiği bir bulanık sınıflandırma işlemi çok önemlidir. Bu problemin üstesinden gelmek için başlıca iki yaklaşım vardır: biri doğrudan işletmenin veya uzmanın deneyimlerini özetleme ve onların bilgisini bulanık kuralların içine çevirmedir. Bilgi edinme ve doğrulama işlemi zordur ve zaman alır. Diğer yaklaşım, örnek durum ve örneklerden bilgi çıkaran veya otomatik olarak üreten makine öğrenmesi içinden bulanık kuralların elde edilmesidir. Makine öğrenmesinin popüler yaklaşımlarından biri karar ağacı atamadır [24]. Fakat sınıflandırma problemlerinde bilgi çıkarımı için kullanılan çoğu karar ağacı atama metotları, insan düşüncesi ve algısı ile ilişkilendirilmiş anlam bulanıklığı ve belirsizliği gibi kavramsal kararsızlıklar ile ilgilenmez.

Diğer makine öğrenmesi metodu GA'dır. GA çok tanınan ve yoğun hesaplama içeren bir arama algoritmasıdır.

3.4. Bulanık Kurallar

$T = \{t_1, t_2, \dots, t_n\}$ işlem kümesi; her işlem T 'de j . değerle temsil edilir. $I = \{i_1, i_2, \dots, i_m\}$ T 'de görülen tüm nesneleri temsil eder. Her nesne i_k ikilik, kategorik veya D i_k nicel ile ifade edilen nicel bir alt küme olabilir. Bunun yanı sıra, her nicel nesne i_k , birkaç bulanık kümeyle ilişki halindedir.

Açıkça, i_k nesnesi için her biri üyelik fonksiyonlarına sahip olan bulanık kümeler tanımlanabilir. Öyleki i_k nesnesinin her nicel değeri bu i_k için tanımlı bulanık kümelerin bir veya birkaçında yer alabilir.

i_k nesnesi için tanımlı bulanık kümelerin birinde, i_k nesnesinin her değerinin üyelik derecesi, i_k nesnesinin bir girdi olarak kullanıldığı bulanık kümelerin üyelik fonksiyonlarının değerlendirmesine doğrudan bağlıdır.

Verilen bir T işlem veritabanında, I kümesi bulanık kümelerle ilgili nicel değerlere sahip nesne kümesi olmak üzere, amaç bazı ilginç, yararlı olabilecek kuralları bulmaktır. Mesela, yeteri kadar desteğe ve yüksek güven değerine sahip kurallar. Her t_j işlemi I kümesinden bazı nesneler içerir ve I 'daki her nicel nesne bir veya daha fazla bulanık kümeler ile ilişkilidir.

Bütün bu tanımlamalar ışığında bir bulanık birliktelik kuralı aşağıdaki gibi gösterilebilir:

Kural: EĞER $Q = \{u_1, u_2, \dots, u_p\}$, $F1 = \{f_1, f_2, \dots, f_p\}$ İSE $R = \{v_1, v_2, \dots, v_q\}$, $F2 = \{g_1, g_2, \dots, g_q\}$ 'dir.

Burada Q ve R ayrık nesne kümeleridir. Yani $Q \subset I$, $R \subset I$ ve $Q \cap R = \emptyset$ 'dir.

$F1$ ve $F2$, sırasıyla Q ve R deki nesnelerle ilgili bulanık kümeleri içerir. Örneğin, f_i , u_i nesnesine ilişkin bulanık kümelerin bir kümesidir ve g_j , v_j nesnesinin ilişkin bulanık kümelerin bir kümesidir.

Sonuç olarak: “ Q , $F1$ ise” kuralın öncülü olarak adlandırılırken; “ R , $F2$ 'dir” kuralın sonculu olarak adlandırılır.

Yukarıdaki tanıma göre bir bulanık kural örneği şöyle olur:

“EĞER birikim geniş İSE yıllık gelir orta'dır”

Burada, birikim ve yıllık gelir, nesne; geniş ve orta ise nesnelerle ilişkili bulanık kümelerdir. Bir kuralın ilginç olabilmesi için, kullanıcı tarafından tanımlanmış belirli eşik değerlerinden daha fazla destek ve güven değerlerine sahip olması gerekir.

4. ÇOK-AMAÇLI GENETİK ALGORİTMALAR

4.1 Çok-Amaçlı Optimizasyon

Çok amaçlı optimizasyon tek amaçlı optimizasyondan farklı olarak, aynı anda birbirleriyle çelişen amaçların (performans ve maliyet gibi) olduğu problemlerle uğraşır. Örneğin, karmaşık bir donanım tasarımı düşünüldüğünde en yüksek performans hedeflenirken maliyetlerin de minimize edilmesi gerekmektedir. Yukarıda bahsedildiği gibi birden fazla amaç olduğunda bunlardan bazıları kısıt olarak tanımlanabilir. Örneğin, bir sistem yüksek performans ve düşük maliyet ile optimize edilirken, sistemin belirli boyutları geçmemesi ayrı bir optimizasyon kriteri olarak tanımlanabilir. Böylece, bir çok amaçlı optimizasyon problemi aşağıdaki gibi formüle edilebilir [25] ve bir amaçlı optimizasyon problemi genel olarak şunları içerir:

- bir parametre kümesi(karar değişkenleri), a
- amaç fonksiyonları kümesi, b
- kısıtlar kümesi, c
- ve amaç fonksiyonları ve kısıtlar karar değişkenlerinin fonksiyonlarıdır.

Matematiksel olarak bir optimizasyon problemi aşağıdaki şekilde gösterilebilir:

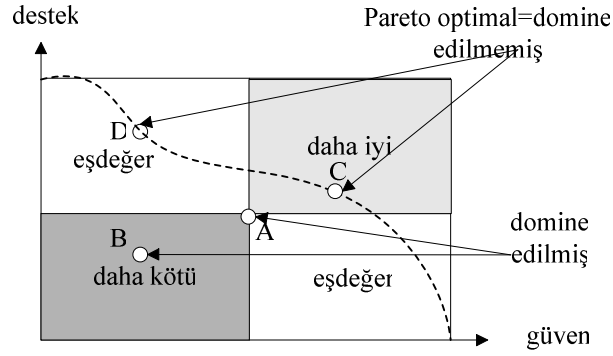
$$\text{Min/max } y:f(x) = (f_1(x), f_2(x), \dots, f_b(x))$$

$$\text{Kısıtlar } e(x) = (e_1(x), e_2(x), \dots, e_c(x)) \leq 0$$

$$x = (x_1, x_2, \dots, x_a) \in X$$

$$y = (y_1, y_2, \dots, y_b) \in Y,$$

Burada x karar vektörü, y amaç vektörü, X karar uzayı, Y amaç uzayı olarak isimlendirilir ve $e(x) \leq 0$ kısıtları olası çözümleri belirler.



Şekil 4.1. Pareto'nun optimalite kavramı

Yukarıdaki tanımlamalar dikkate alınarak, amaçlarımız performans (f_1) ve maliyet fonksiyonunun zıttı olan ucuzluk (f_2) fonksiyonlarının sınırlayıcı şartları (e_1) altında maksimize edilmesidir.

O zaman, optimum dizayn, maksimum performansı, minimum maliyet ile sağlar ve kısıtları aşmaz. Eğer böyle bir çözüm olsaydı, yani her bir amaç için optimum olan çözüm diğeri için de optimum olsaydı aslında sadece tek amaçlı optimizasyon problemi çözmemiz gerekirdi. Fakat çok-amaçlı optimizasyonu zorlaştıran genelde bu amaç fonksiyonlarının optimum çözümlerinin birbirinden farklı olmasıdır.

Şekil 4.1'de amaç fonksiyonları olarak bulanık kuralların destek ve güven değerleri gösterilmektedir. Şekil'de gösterildiği gibi bir çözümde yer alan karar vektörü amaç değerleri açısından bir diğerinden daha iyi, daha kötü, eşit veya farksız olabilir. Daha iyi olması demek çözümün hiçbir amaç değerinin diğer çözümlerin amaç değerlerinden düşük olmaması ve en az bir amaç değerinde daha yüksek olması demektir. Örneğin şekilde B ile temsil edilen çözüm A çözümünden daha kötü ve C çözümü ise A çözümünden daha iyidir. Fakat C ile D çözümleri biri diğerinden daha iyi diyemeyiz. Bunun sebebi amaç değerlerinden birinin bir çözümde, diğerinin ise diğer çözümde yüksek olmasıdır.

Bu kavramı kullanarak optimum çözümü arama uzayında hiçbir diğer çözüm tarafından domine edilemeyen çözüm olarak tanımlayabiliriz. Böyle bir çözümü Pareto'nun optimum çözümü olarak tanımlayabiliriz. Şekilde noktalı çizgiyle gösterilen çözümler ise Pareto'nun optimum çözüm kümesidir.

Böyle bir optimizasyon probleminde amaçlar birbiriyle çakışır ve aynı anda optimize edilemezler. Bunun yerine tatmin edici bir tercih çözümü bulunmalıdır. Bu nedenle uygun tercih seçimi için bir karar verme süreci gerekir.

4.2. Çok-Amaçlı Genetik Algoritmalar

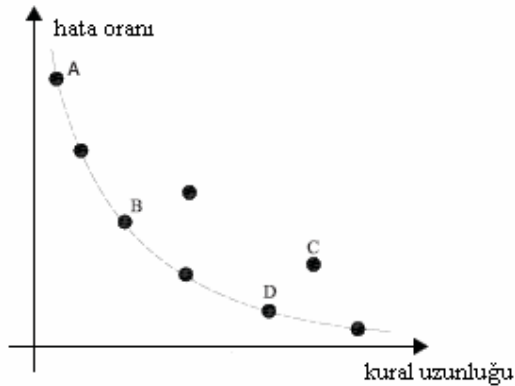
Gerçek dünya problemleri karmaşıktır ve iyi bir çözüm için birden fazla amacın sağlanması gerekmektedir. Birçok proje tek bir fonksiyona dayalı bir yaklaşımla çözüm getirmektedir. Bu basit yaklaşım pek çok durumda çok etkin değildir. İlk olarak amaçlar sık sık birbirleriyle çatışabilmektedir. İkinci olarak amaçlar sık sık aday çözüm kalitesine uygun olmayabilir ve farklılıklar gösterebilir. Çok amaçlı GA'larda bu iki durum da minimize edilmektedir. Genetik algoritmanın doğal ve evrimsel yapısı çok amaçlı yapı için de uygundur.

Pareto en uygun çözümü, çok amaçlı GA'nın doğasına çok uygundur [26]. Goldberg'in [27] seçme algoritması ise [28, 29]'de önerilen çok amaçlı evrimsel algoritmayı kullanır.

Çok amaçlı genetik algoritmalarda birden fazla amaca aynı anda bakılır. Yani tek bir en uygun çözüm yoktur. Amaçlar arasında bir tercih yapılarak bir çözüm kümesi seçilebilir [29]. Bu yöntemle kullanıcı spesifik problem için olası çözümlerden birini seçebilir. Böylece kullanıcı yüksek kaliteli çözümleri inceleyerek amaçlar arasında bir tercih yaparak bir çözüm kümesi seçme fırsatı bulacaktır. Bu yöntem kullanıcıyı tek bir çözüm kümesine zorlamaktan daha iyi bir yöntemdir.

Çok amaçlı GA'ların tercih edilmesinin nedenleri şunlardır:

- (i) Genetik algoritmalar arama uzayının geniş olmasından dolayı güçlü bir algoritmadır.
- (ii) Genetik algoritmalar global bir arama yöntemidir ve greedy arama metotlarından daha kolay bir şekilde nesneler ile iletişime girmektedir.
- (iii) Genetik algoritmalar, çok amaçlı problemleri çözmek için bir aday çözüm kümesiyle çalışır [29].



Şekil 4.2. İki amaçlı problem örneği

Şekil 4.2’de A çözümü düşük kural uzunluğuna sahip ama hata oranı yüksek, D çözümü ise düşük hata oranına ancak yüksek kural uzunluğuna sahiptir. Her iki amaç önemli olduğundan; A çözümü D’den daha iyidir ya da tam tersi söylenemez. Diğer taraftan C çözümü D’den daha kötüdür denebilir.

4.2.1. Üç Amaçlı Bir Problem

Bu tez konusunun temelinde üç amaçlı bir problem vardır. N tane aday “EĞER-İSE” kuralımızın olduğunu varsayalım. Burada amaç yüksek sınıflandırma performansına sahip az sayıda nesne içeren az sayıda “EĞER-İSE” bulanık kurallar kümesi bulmaktır. Bu işlem sınıflandırma doğruluk oranını maksimize etmek, seçilen kural sayısını minimize etmek ve toplam kural uzunluğunu minimize etmektir.

5. ÇOK-AMAÇLI GENETİK ALGORİTMA KULLANARAK ÇOKLU-BULANIK SINIFLANDIRICILARIN GELİŞTİRİLMESİ

5.1. Giriş

Bu tezde çok amaçlı GA kullanılarak hastalık teşhisi yapmak amaçlanmıştır. Hastalık teşhisi için bulunacak kurallarda yüksek sınıflandırma doğruluk oranı, az sayıda kural ve az sayıda nesne şartları aranmaktadır. Kural sayısının az olmasının istenme sebebi, az sayıda kural içeren kural seti ile verilerin daha hızlı sınıflandırılmasıdır. Yani ne kadar az kural olursa, veriler o kadar az zamanda sınıflandırılır. Nesne sayısı azaldıkça kuralın anlaşılabilirliği artmaktadır. Yani kural setinin az sayıda nesne içermesi, o kural setinin daha az karmaşık olduğunu gösterir. Yapılan birçok çalışmada üyelik fonksiyonu taban değerleri uzman bilgisine dayanarak alınmıştır. Bu tez çalışmasında ise hastalık teşhisi için bulanık sınıflandırma kural setleri bulunurken, aynı zamanda nicel değerlere sahip nesneler için üyelik fonksiyonu taban değerleri ayarlanmaktadır. Üretilcek olan kurallar nesne için önerilen üyelik taban değerleri üzerinden hesaplanarak bulunur. Bulunacak olan kurallarda sadece nicel değerli nesneler değil, aynı zamanda iki değerli nesneler de yer almaktadır. Kural çıkarımı için geliştirilen algoritma hepatit ve diyabet veri kümeleri uygulanmıştır.

Hepatit ve diyabet veri kümelerinin kullanılma nedenleri arasında, her iki veri kümesinde de nesne sayıları ve özellikle de nicel nesne sayılarının yüksek olması ve her ikisinin de daha önceki çalışmalarda yüksek hata oranı ile sınıflandırılmaları sayılabilir. Nicel nesne sayısı az olan ve internette mevcut diğer gerçek verilerle yaptığımız tecrübelerde %97'lere ulaşan başarı elde edilmiştir.

5.2. Hepatit ve Diyabet Veritabanlarının Özellikleri

Hastalık teşhisi için, <http://www.ics.uci.edu/~mllearn/MLRepository.html> sitesinden alınan hepatit veri kümesi kullanılmıştır.

Hepatit veri kümesinde 6 tane nicel değerli, 13 tane de 2 değere sahip(1 veya 2) nesne olmak üzere toplam 19 nesne vardır. Sınıf sayısı “die”(1) ve “live”(2) olmak üzere 2 tanedir. Üzerinde çalışılan veri seti 155 örnek içerir bunlardan 32'si die, 125'i live sınıfına aittir. Aşağıda nesneler ile ilgili detaylar gösterilmektedir:

1. Class: Die, Live
2. Age: 10, 20, 30, 40, 50, 60, 70, 80
3. Sex: male, female
4. Steroid: no, yes
5. Antivirals: no, yes
6. Fatigue: no, yes
7. Malaise: no, yes
8. Anorexia: no, yes
9. Liver Big: no, yes
10. Liver Firm: no, yes
11. Spleen Palpable: no, yes
12. Spiders: no, yes
13. Ascites: no, yes
14. Varices: no, yes
15. Bilirubin: 0.39, 0.80, 1.20, 2.00, 3.00, 4.00
16. Alk Phosphate: 33, 80, 120, 160, 200, 250
17. Sgot: 13, 100, 200, 300, 400, 500,
18. Albumin: 2.1, 3.0, 3.8, 4.5, 5.0, 6.0
19. Protine: 10, 20, 30, 40, 50, 60, 70, 80, 90
20. Histology: no, yes

Diyabet veri kümesi de, <http://www.ics.uci.edu/~mlearn/MLRepository.html> sitesinden alınmıştır. Diyabet verileri en az 21 yaşında olan ve USA Arizonada yaşayan bayanlardan alınmıştır. Diyabet veri kümesinde sağlıklı ve diyabetli diye 2 sınıf ve 8 tane nicel değerli nesne vardır. Diyabet veri kümesi 768 örnek içerir.

Aşağıda diyabet veri kümesine ait nesneler ve açıklamaları verilmektedir:

- Nesne 1: Hamilelik sayısı
- Nesne 2: Glikoz konsantrasyonu
- Nesne 3: Kan basıncı
- Nesne 4: Kas katlaması
- Nesne 5: Serum insülin
- Nesne 6: Vücut kütlesi
- Nesne 7: Diyabet soyağacı
- Nesne 8: Yaş

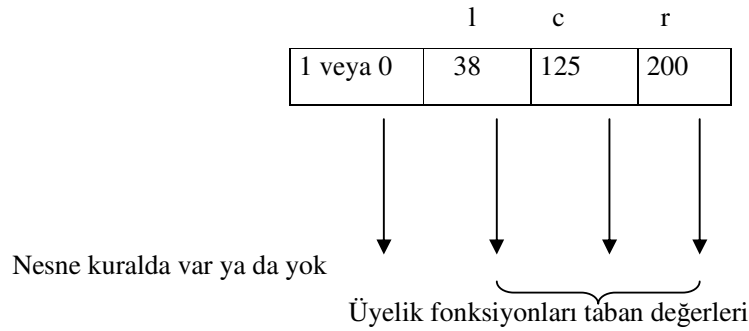
5.3. Önerilen Yöntemin Hepatit Veritabanına Uygulanması

Hepatit veritabanında bulunan nicel değerli nesneler için minimum ve maksimum değerler verilmiştir. Üyelik fonksiyonu taban değerleri üretilirken buradaki minimum ve

maksimum değerlere göre üretilmektedir. Nicel değerli nesneler için minimum ve maksimum değerler içerisinde bir bulanıklaştırma işlemi yapılır. Bulanıklaştırma yapılırken üyelik fonksiyonu sayısı; küçük, orta ve yüksek olmak üzere 3 belirlenmiştir. Yani bu minimum ve maksimum değerler arasında olmak kaydıyla üç üçgensel üyelik fonksiyonunun taban değerleri için l, c, r değerleri üretilir. Cinsiyet(sex) nesnesi için bayan=1 ve erkek=2 ve aynı şekilde, hayır(no) ve evet(yes) değerleri alan diğer nesneler için hayır=1 ve evet=2 ile temsil edilir. Benzer şekilde “die” sınıfı 1 ve “live” sınıfı 2 ile gösterilmektedir.

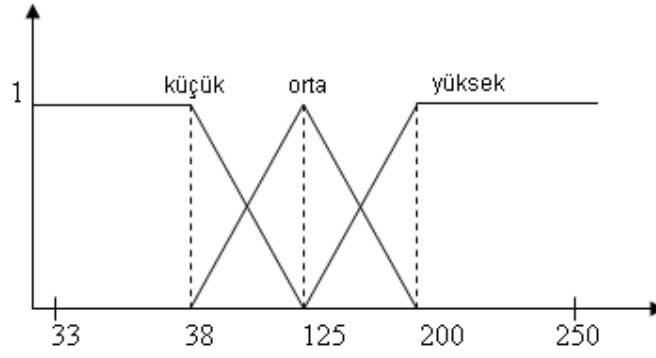
5.3.1. Kromozom Yapısı

Genetik popülasyonun boyutu 150 olarak belirlenip, her 2 sınıf için de üretilen kurallar aynı kromozomda yer almaktadır. Popülasyon dizisindeki her bir kromozomun ilk 51 geni sınıf 1’i (die) sonraki 51 geni ise sınıf 2’yi (live) temsil eder. Gende nicel değerli nesneler 4, 2 değere sahip (hayır-evet, erkek-bayan) nesneler ise 2 gen ile temsil edilir. Örneğin “alk phosphate” nesnesi için minimum ve maksimum değerler 33-250’dir. “Alk phosphate” nesnesi için 4 haneli gen şöyle üretilebilir:



Şekil 5.1 Nicel değerli bir nesnenin gen yapısı

Yukarıda rasgele olarak üretilen taban değerleri için üyelik fonksiyonları şöyle olmaktadır:



Şekil 5.2. Üyelik fonksiyonu taban değerlerinin ayarlanması

Diğer değerler için üyelik fonksiyonu olmayıp onlar için gen yapısı şu şekilde tanımlanmıştır:

0 veya 1	1 veya 2
a	b

Şekil 5.3. Cinsiyet nesnesi için tanımlanan gen yapısı

a değeri: 0 veya 1, bu nesne için kuralda bulunup bulunmama durumunu gösterir.

b değeri: 1 veya 2 ise sırasıyla bayan ve erkek cinsiyetlerini temsil eder.

Böylece bir kromozom yapısını bir sınıf için şöyle özetleyebiliriz:

1...5	1 veya 0	1	c	r	1 veya 0	1	c	r		1 veya 0	1 veya 2	1 veya 0	1 veya 2
Kural sayısı	Nesne kuralda mı?	Alt vartaban değeri	Orta taban değeri	Üst taban değeri						Nesne kuralda var mı?	Evet Hayır		
	1. nicel nesne			2. nicel nesne					2 değerli nesne				
Sınıf 1 için oluşturulacak kural sayısı													
Sınıf 1 için kromozom yapısı													

Şekil 5.4. Kullanılan kromozom yapısı

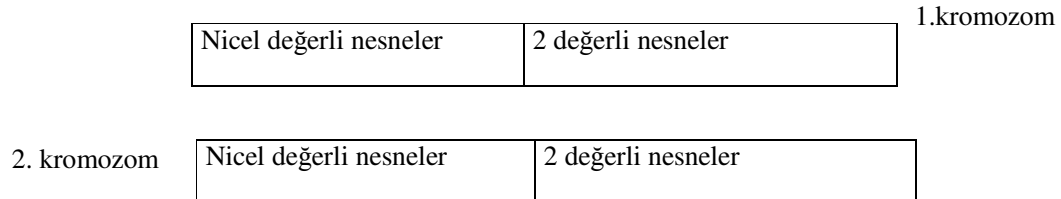
Yukarıda bir kromozomda sınıf 1 için oluşturulan kodlama görülmektedir. Uygulama yazılımında kullanılan hepatit veritabanı 2 sınıf içerdiğinden aynı kodlama 2. sınıf için de

yapılır. Hepatit veritabanında nicel değerli 6 nesne ve 2 değerli 13 nesne vardır. Sınıf 1 ve 2 için kromozomda toplam 102 gerçek değerli alan mevcuttur.

5.3.2. Genetik Operatörler

5.3.2.1. Çaprazlama

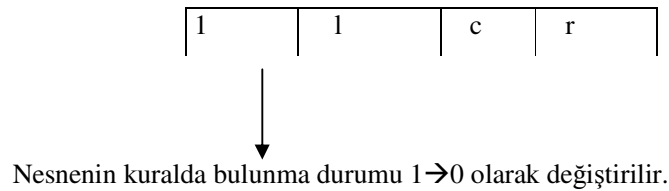
Bu tez çalışmasında kullanılan çaprazlama yöntemi tek noktalı çaprazlamadır. 0.8 olasılıkla çaprazlama işlemi gerçekleştirilmektedir. İyi olan iki kromozom rasgele bir yerden kesilen parçaları karşılıklı olarak yer değiştirilir. Kesilen parçalar aşağıdaki şekilde yer değiştirmektedir:



Şekil 5.5. Çaprazlama işlemi

5.3.2.2. Mutasyon

Mutasyon olasılığı 0.01 olarak belirlenmiştir. Mutasyon işlemi esnasında bir bit değişecek şekilde işlem yapılır. Burada mutasyon işlemi nesnelerin kuralda bulunup bulunmama durumlarını değiştirmektedir. Nesne kuralda var (1) ise yok (0), kuralda yok (0) ise var (1) olacak şekilde değiştirilir.



Şekil 5.6. Mutasyon işlemi

5.3.2.3. Uygunluk Değerlendirmesi

GA'larda kullanılan uygunluk fonksiyonları, üretilen çözümün ne derece iyi bir çözüm olduğunu değerlendirmek için kullanılır. GA'larda, genelde uygunluk değeri yüksek olan çözüm daha iyi bir çözüm olmaktadır.

Bu tez çalışmasında çok-amaçlı GA kullanıldığından uygunluk bir fonksiyon değildir. Burada kullanılan uygunluk, her bir çözümün 3 değere sahip olduğu bir dizi tarafından temsil edilmektedir. Uygunluk dizisinde bulunan 3 değer; sınıflandırma doğruluğu, kural setindeki toplam kural sayısı ve kural setlerinde bulunan toplam nesne sayısıdır. Burada uygunluk değerleri bir fonksiyona tabi tutulup sonuç olarak tek bir değer elde edilmez. Kullanıcının giriş ekranından girdiği kural sayısı ve doğruluk değerini aşan kurallar kullanıcıya seçenek olarak sunulur. Aşağıda örnek bir uygunluk dizisi verilmiştir:

Doğruluk %	Toplam kural	Toplam nesne
75	6	45

Şekil 5.7. Örnek uygunluk dizisi

5.3.3. Algoritma

Başlangıç değerleri:

Popülasyon boyutu: 150

Çaprazlama olasılığı: 0.8

Mutasyon olasılığı: 0.01

Durdurma kistası: 1000 iterasyon

Minimum doğruluk: %70

Maksimum kural sayısı:5

1.....for 1 to 1000 do

2.....genetik_algoritma ile başlangıç popülasyonu üretilir

3.....üretilen başlangıç popülasyonuna göre kural setleri üretilir

4.....kural setlerinin üyelik değerleri hesaplanır

5.....kural setlerinde bulunan toplam kural uzunluğu (kural sayısı) hesaplanır

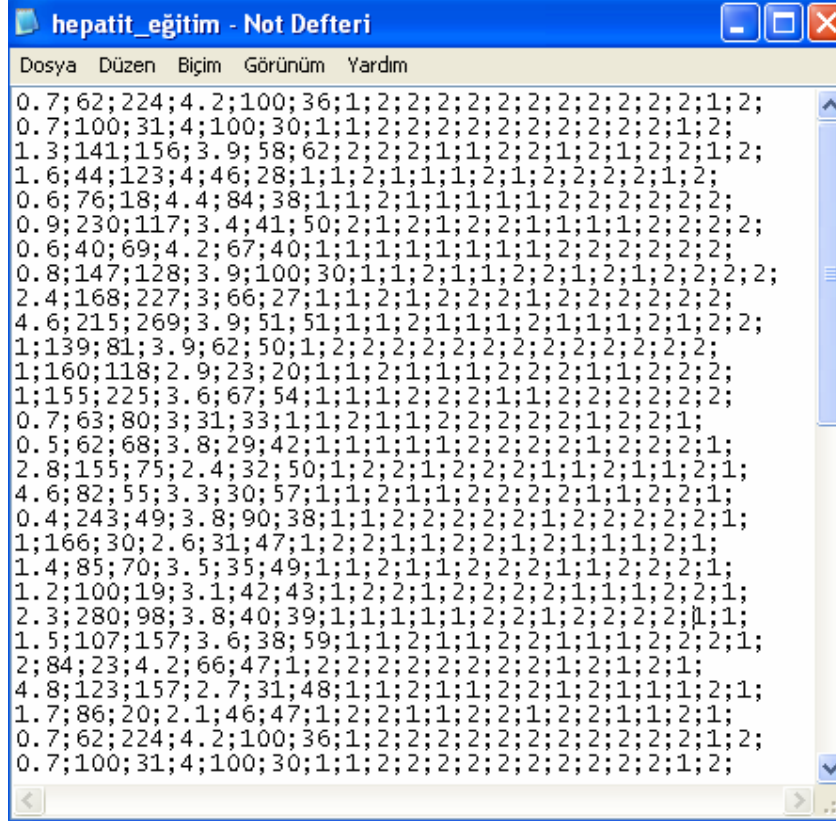
6.....seçme kriterlerine uygun olan kromozomlar ve kurallar seçilir

- 7.....iyi olan genler yoksa 2. adıma dönülür
- 8.....iyi genler çaprazlanarak kötü olan genlerin yerine yerleştirilir,
- 9.....mutasyon yapılır
- 10.....istenen doğruluk ve kural sayısına uygun kromozomlar kural setine alınır
- 11.....istenen değerler bulununca veya döngü bitince işlem bitirilir.

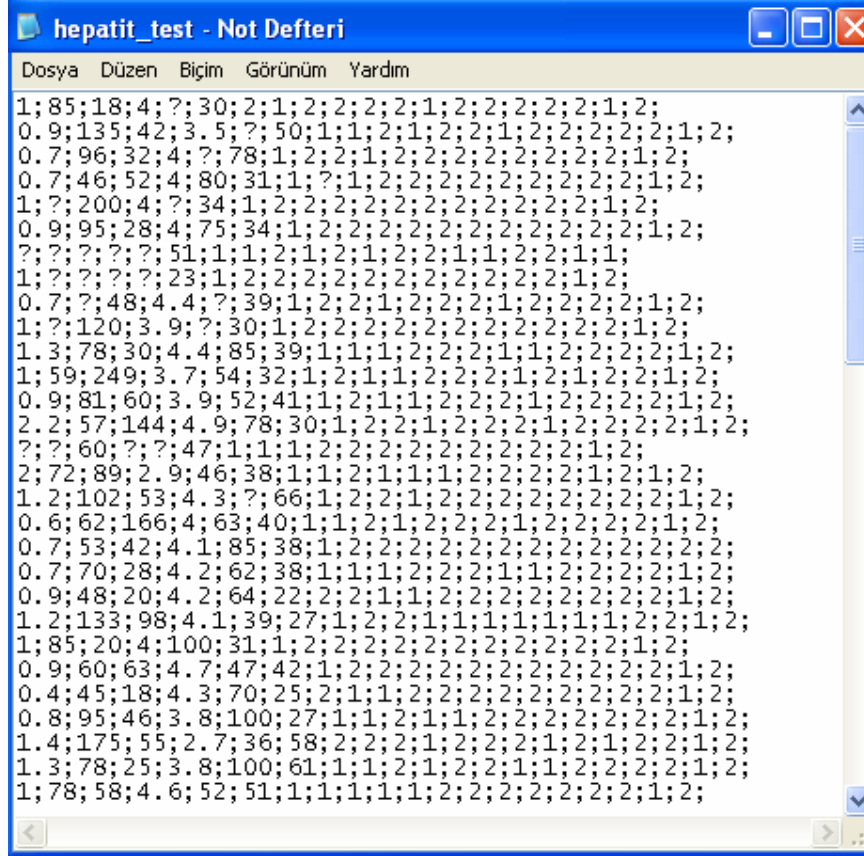
Şekil 5.8. Önerilen algoritma

6. UYGULAMA YAZILIMI

Aşağıda uygulama yazılımının kullandığı hepatit eğitim ve test veri kümeleri görülmektedir.



Şekil 6.1. Hepatit eğitim verileri



Şekil 6.2. Hepatit test verileri

Yukarıdaki hepatit eğitim ve test verileri görülmektedir. ? olan değerler kayıp verileri temsil eder. Eğitim verileri için daha çok temiz ya da az kayıp içeren veriler seçilmiştir. Ancak test verileri içerisinde kayıp veriler oldukça fazladır. Kayıp verilerden oluşan bir giriş sınıflandırılırken, kayıp veriler etkisiz elaman olarak dikkate alınır. Tüm verilerin 3/2'si eğitim, 3/1'i test için kullanılmıştır.



Şekil 6.3. Program ana ekranı

Maksimum kural sayısı girişi kısmında, bulunacak kural setlerinde maksimum kaç kuralın olmasını istiyorsak buradan seçilebilir. Maksimum 5 kural olarak girilmiştir. Çaprazlama olasılığı 0.8 ve mutasyon olasılığı 0.01 olarak girilmiştir. Doğruluk oranı kısmında ise %70'in üzerinde doğruluk sağlayan kuralların getirilmesi istenmiştir.

6.1. Uygulama Ortamı

Uygulama yazılımı gerçekleştirilirken Delphi 6.0 programlama dili kullanılmıştır. Testler Pentium III 2.4GHz işlemciye ve 1 GB RAM'e sahip bir bilgisayarda yapılmıştır.

6.2. Geliştirilen Yazılımla Elde Edilen Kurallar

Aşağıda bulunan kurallar ve taban değerleri verilmiştir. Her bir kural seti bir bütün yani sınıflandırıcı kümesi olarak kullanılmaktadır. Bulunan en iyi 3 sınıflandırıcı aşağıda gösterilmektedir. Bunların her biri hem kendi başına yani bireysel olarak sınıflandırma yapmıştır hem de üçü kural seti beraber çoklu sınıflandırma yapmışlardır.

Kural Seti 1 (1 nolu Bulanık Sınıflandırıcı):

Kurallar:

- 1-EĞER bilirubin orta VE alk_phosphate yüksek VE sgot küçük VE albumin yüksek VE antivirals evet VE anorexia evet VE liver_big evet İSE sınıf 1(Die)
- 2-EĞER bilirubin küçük VE alk_phosphate küçük VE sgot yüksek VE protime yüksek VE age orta VE varices evet İSE sınıf 2 (Live)
- 3-EĞER bilirubin küçük VE alk_phosphate orta VE sgot orta VE protime orta VE age orta VE varices evet İSE sınıf 2 (Live)

1 nolu Kural Setinin Uygunluk Dizisi:

Tablo 6.1. 1 nolu bulanık sınıflandırıcının uygunluk dizisi

	Doğruluk	Kural sayısı	Toplam nesne
Eğitim	%73	3	19
Test	%72.5	3	19

1 nolu Kural setinin taban değerleri

Tablo 6.2. 1 nolu bulanık sınıflandırıcının nicel nesneler için önerdiği taban değerleri

Nicel nesneler	Taban değerleri (l,c,r)			Taban değerleri (l,c,r)		
	Sınıf 1(die) için			Sınıf 2(live) için		
Bilirubin	0.43	0.77	3.43	0.5	2.97	3.37
Alk phosphate	43	111	147	61	102	126
Sgot	193	206	256	74	304	454
Albümin	2.7	3.8	3.9	2.3	4.5	5.4
Prottime	78	80	86	20	21	47
Age	17	21	22	13	61	74

Kural Seti 2 (2 nolu Bulanık Sınıflandırıcı):

Kurallar:

- 1-EĞER alk_phosphate küçük VE protime yüksek VE age yüksek VE sex bayan VE anorexia evet VE ascites hayır İSE sınıf 1(Die)
- 2-EĞER alk_phosphate küçük VE protime orta VE age yüksek VE sex bayan VE anorexia evet VE ascites hayır İSE sınıf 1(Die)
- 3-EĞER albumin orta VE protime orta VE age orta VE varices evet İSE sınıf 2 (Live)

2 nolu kural setinin uygunluk dizisi:

Tablo 6.3. 2 nolu bulanık sınıflandırıcının uygunluk dizisi

	Doğruluk	Kural sayısı	Toplam nesne
Eğitim	%70	3	14
Test	%80	3	14

2 nolu kural setinin taban değerleri:

Tablo 6.4 2 nolu bulanık sınıflandırıcının nicel nesneler için önerdiği taban değerleri

Nicel nesne	Taban değerleri (l,c,r)	Taban değerleri (l,c,r)
	Sınıf 1(die) için	Sınıf 2(live) için
Bilirubin	0.41 1.33 3	0.57 1.86 3.41
Alk phosphate	67 167 207	47 74 143
Sgot	38 374 447	103 334 349
Albumin	2.8 3.1 3.7	2.2 3 5.3
Protime	15 17 62	19 45 56
Age	46 59 68	18 21 79

Kural Seti 3 (3 nolu Bulanık Sınıflandırıcı):

Kurallar:

- 1-EĞER alk_phosphate orta VE albumin. küçük VE age yüksek VE sex bayan VE fatigue hayır VE liver_big evet VE ascites hayır İSE sınıf 1 (Die)
- 2-EĞER albumin yüksek VE age orta İSE sınıf 2 (Live)
- 3-EĞER albumin orta VE age orta İSE sınıf 2 (Live)

3 nolu kural setinin uygunluk dizisi:

Tablo 6.5. 3 nolu bulanık sınıflandırıcının uygunluk dizisi

	Doğruluk	Kural sayısı	Toplam nesne
Eğitim	%70	3	11
Test	%78	3	11

3 nolu kural setinin taban değerleri:

Tablo 6.6. 3 nolu bulanık sınıflandırıcının nicel nesneler için önerdiği taban değerleri

Nicel nesneler	Taban değerleri (l,c,r)	Taban değerleri (l,c,r)
	Sınıf 1(die) için	Sınıf 2(live) için
Bilirubin	1.02 1.89 3.27	0.39 3.86 3.91
Alk phosphate	65 140 168	105 113 216
Sgot	98 265 382	63 270 451
Albumin	2.4 3 4.1	2.5 3.8 5.3
Prottime	61 64 82	15 35 37
Age	41 46 72	19 45 67

Bir kural setine ait kural sayısının az olması istenen bir durumdur çünkü sınıflandırma süresi ile kural sayısı doğru orantılıdır. Yani sınıflandırma doğruluğu eşit olan iki kural setinden kural sayısı az olan daha kısa sürede sınıflandırma yapabilir. Kural setindeki nesne sayısının az olması ise yine istenen bir durumdur. Bu da kuralın anlaşılabilirliğini artırır ki bu sınıflandırma işlemi için önemlidir

6.3. ÇOKLU SINIFLANDIRICILAR

Çoklu sınıflandırıcı, sınıflandırma işlemine birkaç tane kural setinin katılmasıyla oluşmaktadır. Bu tezde çoklu sınıflandırma için, sınıflandırıcılar içerisindeki kuralların sol tarafına bakılır. Destek değeri en yüksek olan sınıf o kural seti için kazanan sınıf olur. Veri sınıflandırıcılar içerisinde kazanan sınıflardan en yüksek destek değerine sahip olan sınıfa dahil edilir.

Ishibuchi ve Tanaka [30]'da çoklu sınıflandırıcıların bir uygulamasını yapmışlardır. Burada;

$$S=S_1S_2S_3.....S_N$$

S'ler sınıflandırıcıları; N ise aday kural sayısını gösterir. $s_j=0$ ve $s_j=1$ olmak üzere, s değişkeni 2 değer alır. Bunun anlamı j. aday kuralı S kural setinde bulunur ya da bulunmaz demektir. Örneğin, her sınıf kural için 300 kural üretilecekse ve M tane sınıf varsa 300M tane kural olur. Üç sınıflı bir örneği düşünürsek, $300 \times M=900$ tane kural demektir ve tüm kurallar $S=S_1S_2S_3.....S_{900}$ olur. Burada mevcut olabilecek tüm kurallara bakıldığı için hesaplamalar oldukça karmaşık olmakta ve sınıflandırma oldukça zaman almaktadır. Bu şekilde elde edilen S sınıflandırıcılarından çok-amaçlı genetik algoritma kullanarak seçme yapılmakta ve uygun S sınıflandırıcıları verilmektedir.

Bölüm 6.2'de elde edilen kural setlerinin her biriyle hem bireysel olarak hem de çoklu sınıflandırıcı olarak tamamıyla sınıflandırma işlemi yapılmıştır. Bireysel olarak eğitim ve test verisinde verdiği başarı ile çoklu sınıflandırıcı olarak verdiği başarı farklılıklar göstermektedir. Bireysel olarak test verisi sınıflandırmaya tabi tutulduğunda her bir kural seti tüm test verisini teker teker sınıflandırır. Çoklu sınıflandırıcılar ile sınıflandırma yapıldığında ise 3 kural setinde bulunan kuralların her bir test verisi için destek değeri bulunarak destek değeri en fazla olan sınıflandırıcıdaki sınıfa dahil edilmektedir. Kuralın destek değeri değeri şöyle hesaplanmaktadır:

EĞER bilirubin orta VE alk_phosphate yüksek VE sgot küçük VE albumin yüksek VE antivirals evet VE anorexia evet VE liver_big evet İSE sınıf 1(Die)

Kuralını ele alalım. Bu kuralda 4 tane nicel 3 tane de iki değere sahip nesneler bulunmaktadır. Kuralın destek değeri hesaplanırken o kural için önerilen üyelik fonksiyonu taban değerlerini de kullanılmaktadır.

Yukarıdaki kural için önerilen taban değerleri aşağıda gösterilmektedir:

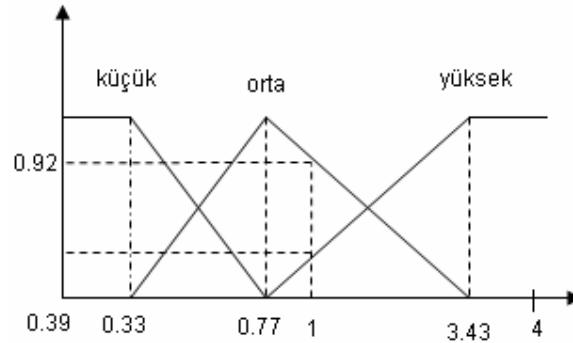
Tablo 6.7. Örnek kuralın taban değerleri

Nicel nesneler	Taban değerleri (l,c,r)			Taban değerleri (l,c,r)		
	Sınıf 1(die) için			Sınıf 2(live) için		
Bilirubin	0.43	0.77	3.43	0.5	2.97	3.37
Alk phosphate	43	111	147	61	102	126
Sgot	193	206	256	74	304	454
Albumin	2.7	3.8	3.9	2.3	4.5	5.4

Gelen bir test verisi örneği de aşağıda verilmektedir:

1;85;18;4;?;30;2;1;2;2;2;2;1;2;2;2;2;1;2;

Yukarıdaki test verisinin en sonunda bulunan 2, sınıf 2’yi göstermektedir. “Bilirubin” nesnesi için üyelik fonksiyonları aşağıdaki gibi olmaktadır:



Şekil 6.4. Örnek kurala göre elde edilen destek değeri

Test verisinde bilirubin nesnesine ait değer 1’dir. Kuralda ise “bilirubin orta” dilsel terimi olduğundan, “yüksek” dilsel terimine sahip üyelik üzerinde hiçbir işlem yapmıyoruz. Aynı şekilde “alk phosphate”, “sgot” ve “albumin” için üyelik değerleri bulunup “VE” işlemi yapılarak destek değeri bulunmuş olur. Test verisi kural setleri arasında destek değeri maksimum olan sınıfa dahil edilir.

Aşağıda bireysel, çoklu ve “c4.5 [31]” sınıflandırıcılar için elde edilen en iyi sonuçlar verilmiştir.

Tablo 6.8. Hepatit verisi için en iyi sonuçlar

	Test verisi için
Bireysel Sınıflandırıcı	%80
Çoklu Sınıflandırıcı	%84
C 4.5	%78

Tablo 6.9. Diyabet verisi için en iyi sonuçlar

	Test verisi için
Bireysel Sınıflandırıcı	%75
Çoklu Sınıflandırıcı	%78
C 4.5	%75

Önerilen yöntemle yapılan testler sonucunda çoklu-sınıflandırıcıların bireysel ve C4.5’ten daha iyi sonuç verdiği görülmektedir. Bireysel olarak elde edilen kural setleri önce bireysel olarak sonra da birleştirilerek çoklu olarak test edilmiştir. Bireysel olarak elde edilen iyi kurallar birleştirilerek çoklu-sınıflandırıcı olarak kullanıldığında bireyselden daha iyi sonuç verdiği görülmüştür.

7. SONUÇLAR

Bu tez çalışmasında bireysel ve çoklu-sınıflandırıcılar üzerinde durulmuştur. Sınıflandırıcılar geliştirilirken çok-amaçlı GA kullanılarak, birden fazla amacın optimizasyonu sağlanmaktadır. Klasik GA'da kullanılan uygunluk fonksiyonu çok amaçlı-GA'da üç boyutlu bir uygunluk dizisine dönmektedir. Dolayısıyla bu uygunluk değerlendirmesinde bir çözüm kümesinin uygunluk değeri bir sayısal büyüklük değil, üç sayısal büyüklükten oluşmaktadır. Uygunluk değerlendirilmesindeki amaçlar, kural sayısının az, sınıflandırma doğruluğunun yüksek ve nesne sayısının az olmasıdır. Çok amaçlı-GA kullanarak üretilen çözüm kümesinin esnek olması, yani bir tek çözüm değil, birden fazla çözümün kullanıcıya önerilmesi sağlanmıştır.

Önerilen algoritma ile hepatit ve diyabet hastalık veri kümeleri için bireysel sınıflandırıcılar elde edilmiştir. Bireysel sınıflandırıcıların birleştirilmesiyle elde edilen çoklu-sınıflandırıcılar ile örnekler tekrar sınıflandırılmıştır. Yapılan testler sonucunda çoklu-sınıflandırıcıların bireysel ve C4.5 sınıflandırıcılardan daha iyi olduğu görülmüştür.

Bu tez çalışması aynı zamanda, genelde uzman bilgisine dayalı olarak elde edilen bulanık küme taban değerlerini nesnelerin alabileceği değer aralığında olacak şekilde her sınıf için farklı ve otomatik olarak önermektedir.

Bundan sonra veri kümelerinde bulunan eksik değerlerin “?” giderilmesi üzerinde çalışılabilir. Eksik değerler, veri kümesinin geneli ele alınarak tahmin edildikten sonra bireysel ve çoklu-sınıflandırıcılar ile tekrar testler yapıp sonuçları tartışılabilir.