

KOCAELİ ÜNİVERSİTESİ * FEN BİLİMLERİ ENSTİTÜSÜ

**BULANIK SINIFLANDIRMA İLE BEYİN MR GÖRÜNTÜLERİNDE TÜMÖR
TESPİTİ**

YÜKSEK LİSANS TEZİ

Bilgisayar Müh. Arslan ALAN

Anabilim Dalı: Bilgisayar Mühendisliği

Danışman: Doç.Dr. Yaşar BECERİKLİ

KOCAELİ, 2010

KOCAELİ ÜNİVERSİTESİ * FEN BİLİMLERİ ENSTİTÜSÜ

BULANIK SINIFLANDIRMA İLE BEYİN MR GÖRÜNTÜLERİNDE TÜMÖR
TESPİTİ

YÜKSEK LİSANS TEZİ

Bilgisayar Müh. Arslan ALAN

Tezin Enstitüye Verildiği Tarih: 14.10.2010

Tezin Savunulduğu Tarih: 08.11.2010

Tez Danışmanı


Doç.Dr. Yaşar BECERİKLİ

Üye


Doç.Dr. Adnan KAVAK

Üye


Doç.Dr. Cemil ÖZ

KOCAELİ, 2010

ÖNSÖZ ve TEŞEKKÜR

Bu tez çalışmasında örüntü tanıma kavramı ve buna bağlı olarak kümeleme ve sınıflandırma kavramları anlatılmıştır. Kesin çizgilerde dolaşan klasik yöntemler yerine insan düşünce sistemine daha yakın olan bulanık sistemlerin bu kavramlar üzerinde sağladığı avantajlar sunulmuştur. Çalışmalar esnasında görüntü verisi kullanılmış ve bu veri üzerinde yapılan bulanık sistem tabanlı uygulamalarla mevcut sistemlere göre daha iyi sonuçlar elde edilmiştir.

Çalışmalarına yön veren ve her tez görüşmesinde bana ivme kazandıran değerli danışman hocam Doç.Dr.Yaşar BECERİKLİ'ye teşekkürlerimi sunar, bugüne kadar maddi manevi desteklerini hiç esirgemeyen değerli aileme ve biricik eşime sonsuz teşekkürler ederim.

İÇİNDEKİLER

| | |
|--|------|
| ÖNSÖZ..... | i |
| İÇİNDEKİLER..... | ii |
| ŞEKİLLER DİZİNİ..... | iv |
| TABLolar DİZİNİ..... | vi |
| SEMBOLLER..... | vii |
| ÖZET..... | viii |
| İNGİLİZCE ÖZET..... | ix |
| 1. GİRİŞ..... | 1 |
| 2. ÖRÜNTÜ TANIMA..... | 4 |
| 2.1. Örüntü Tanıma Nedir?..... | 4 |
| 2.2. Kümeleme..... | 8 |
| 2.2.1. Kümeyi temsil eden örnek nokta (ilk örnek) ile kümeleme modelleri..... | 11 |
| 2.2.1.1. K (C)-ortalama modeli..... | 12 |
| 2.2.1.2. Yarı eğitilmiş kümeleme modelleri..... | 15 |
| 2.3. Sınıflandırma..... | 17 |
| 2.4. Bölütleme..... | 18 |
| 2.4.1. Nokta, çizgi ve kenar bulma..... | 19 |
| 2.4.1.1. Nokta bulma..... | 20 |
| 2.4.1.2. Çizgi bulma..... | 21 |
| 2.4.1.3. Kenar bulma..... | 21 |
| 2.4.1.3.1. Sobel kenar bulma yöntemi..... | 24 |
| 2.4.1.3.2. Prewitt kenar bulma yöntemi..... | 24 |
| 2.4.1.3.3. Roberts kenar bulma yöntemi..... | 25 |
| 2.4.1.3.4. LoG (Laplacian of Gaussian) kenar bulma yöntemi..... | 25 |
| 2.4.1.3.5. Canny kenar bulma yöntemi..... | 26 |
| 2.4.2. Eşik değeri ile bölütleme (thresholding)..... | 27 |
| 2.4.3. Alan tabanlı bölütleme (Region-based segmentation)..... | 28 |
| 2.4.3.1. Alan büyütme (region growing)..... | 28 |
| 2.4.3.2. Alan bölme ve birleştirme..... | 30 |
| 2.4.4. Watershed dönüşümü ile bölütleme..... | 31 |
| 3. BULANIK KÜMELEME..... | 33 |
| 3.1. Bulanık c-ortalama kümeleme modeli..... | 34 |
| 4. BULANIK SINIFLANDIRMA..... | 37 |
| 4.1. Sınıflandırıcı Eğitimi..... | 38 |
| 4.1.1. Yinelemesiz çevrimdışı eğitim..... | 39 |
| 4.1.2. Yinelemeli statik çevrimdışı eğitim..... | 39 |
| 4.1.3. Dinamik çevrimiçi yinelemeli eğitim..... | 39 |
| 4.2. Sınıfı Temsil Eden Örnek Nokta (İlk Örnek) ile Sınıflandırıcı Tasarımı..... | 39 |
| 4.2.1. En yakın ilk örnek nokta sınıflandırıcı..... | 40 |
| 4.2.2. Çoklu ilk örnek (prototip) tasarımı..... | 41 |
| 4.3. En Yakın Komşu Sınıflandırıcı..... | 41 |
| 4.4. Bulanık Kural Tabanlı Sınıflandırıcı..... | 42 |
| 4.4.1. Klasik karar ağaçları..... | 42 |
| 4.4.2. Klasik karar ağaçlarından kural çıkarımı..... | 43 |

| | |
|---|----|
| 4.4.3. Klasik karar ağacı tasarımı..... | 45 |
| 4.4.4. Bulanık sistem modelleri ve fonksiyon kestirimi..... | 45 |
| 4.4.5. Bulanık sistemler için kural çıkarımı | 47 |
| 5. BULANIK BÖLÜTLEME | 49 |
| 5.1. Eşik Değeri ile Bölütleme | 49 |
| 5.2. Kümeleme ile Bölütleme | 50 |
| 5.3. Kural-Tabanlı Bölütleme..... | 54 |
| 6. BULGULAR VE TARTIŞMA..... | 56 |
| 6.1. Veri Kümeleri | 56 |
| 6.1.1. MR veri kümesi | 56 |
| 6.1.2. Iris veri kümesi (Fisher, 1936)..... | 60 |
| 6.1.3. X30 veri kümesi | 63 |
| 6.2. Bulanık Çıkarım Sistemiyle Sınıflandırıcı Tasarımı..... | 64 |
| 6.3. Bulanık C-Ortalama ve K-Ortalama Algoritmalarının Iris ve X30 Veri Kümelerine Uygulanması..... | 68 |
| 6.4. Iris Verisi ile Bulanık Sistem Tabanlı Sınıflandırma | 70 |
| 6.5. Bulanık Sistem Tabanlı Tümör Tespiti..... | 72 |
| 7. SONUÇLAR VE ÖNERİLER | 80 |
| KAYNAKLAR | 82 |
| EKLER..... | 85 |
| ÖZGEÇMİŞ | 93 |

ŞEKİLLER DİZİNİ

| | |
|---|----|
| Şekil 2. 1: Sayısal örüntü tanımının bölümleri | 5 |
| Şekil 2. 2: Küme analizinde üç aşama | 8 |
| Şekil 2. 3: Eğitimsiz k(c)-ortalama algoritması sonucu | 16 |
| Şekil 2. 4: Yarı eğitilmiş k(c)-ortalama algoritması sonucu | 17 |
| Şekil 2. 5: 3x3'lük maske | 19 |
| Şekil 2. 6: Noktaları bulan 3x3'lük maske | 20 |
| Şekil 2. 7: Nokta bulan maskenin uygulanması | 20 |
| Şekil 2. 8: Çizgileri bulan 3x3'lük maske | 21 |
| Şekil 2. 9: Çizgi bulan maskenin uygulanması | 21 |
| Şekil 2. 10: Kenar modeli | 23 |
| Şekil 2. 11: Gri seviyesi değişiminin türevleri | 23 |
| Şekil 2. 12: Sobel kenar bulma maskesi..... | 24 |
| Şekil 2. 13: Prewit kenar bulma maskesi..... | 25 |
| Şekil 2. 14: Roberts kenar bulma maskesi | 25 |
| Şekil 2. 15: Sobel, Prewit ve Roberts kenar bulma yöntemlerinin karşılaştırılması . | 26 |
| Şekil 2. 16: LoG kenar bulma maskeleri..... | 26 |
| Şekil 2. 17: Canny ve LoG kenar bulma yöntemlerinin karşılaştırılması | 27 |
| Şekil 2. 18: Eşik değeri ile bölütleme..... | 29 |
| Şekil 2. 19: Orijinal görüntünün histogramı..... | 29 |
| Şekil 2. 20: Alan bölme işlemi | 30 |
| Şekil 2. 21: Görüntünün üç boyutlu gösterimi..... | 31 |
| Şekil 2. 22: Görüntüye Watershed algoritması uygulanması | 32 |
| Şekil 4. 1: İlk örneklere göre sınıflandırma | 40 |
| Şekil 4. 2: Üç sınıftan oluşan karar ağacı | 42 |
| Şekil 4. 3: Kuralların geometrik gösterimi | 44 |
| Şekil 4. 4: Kurallar için genişletilmiş değer aralığı | 44 |
| Şekil 4. 5: MA ve TS modellerinin yapısı | 46 |
| Şekil 5. 1: Kümeleme ile bölümlendirme | 51 |
| Şekil 5. 2: Farklı durulaştırma sonuçları | 52 |
| Şekil 5. 3: FCM için "m" ve "c" parametrelerinin kestirimi | 54 |
| Şekil 6. 1: MR görüntüsü bilgileri..... | 57 |
| Şekil 6. 2: ITK-SNAP programıyla okunan görüntü | 57 |
| Şekil 6. 3: ITK-SNAP programıyla elde edilen görüntüler | 58 |
| Şekil 6. 4: Axial görüntüdeki bölümler | 59 |
| Şekil 6. 5: Iris veri kümesinin diyagramı | 60 |
| Şekil 6. 6: X ₃₀ veri kümesinin diyagramı | 61 |
| Şekil 6. 7: Giriş vektörleri için gauss üyelik fonksiyonları..... | 66 |
| Şekil 6. 8: MA sistem için çıkış üyelik fonksiyonu | 67 |
| Şekil 6. 9: Çıkış üyelik değeri hesaplanması | 67 |
| Şekil 6. 10: Iris verisinde özellik vektörlerinin diyagramları..... | 70 |
| Şekil 6. 11: Iris verisinde özellik vektörlerinin karşılıklı diyagramları | 71 |
| Şekil 6. 12: Wiener filtresi uygulanması | 73 |
| Şekil 6.13: Bulanık c-ortalama kümeleme algoritmasıyla elde edilen yedi kümeye ait görüntüler. | 75 |

| | |
|--|----|
| Şekil 6. 14: T2 MR görüntüleri..... | 76 |
| Şekil 6. 15: Bulanık kural tabanlı sistem ile tümör tespiti | 77 |
| Şekil 6. 16: Üyelik fonksiyonları | 78 |
| Şekil 6. 17: Bulanık kural tabanlı sistem ile elde edilen sonuçlar. | 79 |

TABLolar DİZİNİ

| | |
|--|----|
| Tablo 2. 1: Klasik, Bulanık ve Olasılık etiketlerinin gösterimi | 10 |
| Tablo 2. 2: Klasik k-ortalama algoritması | 14 |
| Tablo 3. 1: Bulanık c-ortalama algoritması | 35 |
| Tablo 6. 1: Fisher (1936) tarafından sunulan iris veri kümesi | 62 |
| Tablo 6. 2: X_{30} veri kümesi | 63 |
| Tablo 6. 3: Örnek giriş verisi ve bulanık sistem ile alınan sonuçlar | 68 |

SEMBOLLER

| | |
|---------------|--|
| N_{pc} | : Olasılık etiket vektörü. |
| N_{fc} | : Bulanık etiket vektörü. |
| N_{fc} | : Klasik etiket vektörü. |
| R^p | : p adet özellikten oluşan özellik uzayı. |
| D | : Sınıflandırıcı fonksiyonu, programı. |
| H | : Olasılık veya bulanık değerleri klasik değerlere dönüştüren fonksiyon. |
| U | : Üyelik matrisi. |
| M_{pcn} | : Olasılık üyelik matrisi. |
| M_{fcn} | : Bulanık üyelik matrisi. |
| M_{hcn} | : Klasik üyelik matrisi. |
| p | : Verideki nokta sayısı. |
| C | : Küme (sınıf) sayısı. |
| V | : Küme (sınıf) merkezi. |
| e | : Kümeleme algoritması. |
| T | : Adım (iterasyon) sayısı. |
| D_{ik} | : Uzaklık ölçümü. |
| J | : Uzaklıklar toplamı. |
| $P_{i,j}$ | : Görüntü. |
| IJ | : Görüntüdeki satır ve sütunlar. |
| P | : Benzerliği ölçen fonksiyon. |
| ∇f | : Eğim. |
| $\alpha(x,y)$ | : Eğimin yönü. |
| X_{tr} | : Eğitim verisi. |
| X_{te} | : Test verisi. |
| S_x | : Bulanık sistem çıkışı. |

Kısaltmalar

| | |
|-------|--|
| MR | : Magnetic Resonans (Manyetik Rezonans) |
| FCM | : Fuzzy C-Means (Bulanık C-Ortalama) |
| FIS | : Fuzzy Inference System (Bulanık Çıkarım Sistemi) |
| TS | : Takagi-Sugeno (Takagi - Sugeno) |
| MA | : Mamdani-Assilian (Mamdani - Assilian) |
| K-NN | : K Nearest Neighbour (K En Yakın Komşu) |
| DT | : Decision Tree (Karar Ağacı) |
| ID3 | : Iterative Dichotomiser (Yinelemeli Dichotomiser) |
| CART | : Classification and Regression Tree (Sınıflandırma ve Regresyon Ağacı) |
| MHA | : Meta Image File Format (Meta Görüntü Dosyası Formatı) |
| UCNIA | : Utah Center for NeuroImage Analysis (Utah Nöro Görüntü Analiz Merkezi) |

BULANIK SINIFLANDIRMA İLE BEYİN MR GÖRÜNTÜLERİNDE TÜMÖR TESPİTİ

Arslan ALAN

Anahtar Kelimeler: Bulanık sistem, Sınıflandırma, Kümeleme, Bölütleme, Görüntü İşleme.

Özet: Bu tez çalışmasında, insan düşünce sistemine yakın olan bulanık sistemin getirdiği avantajlar kullanılarak beyin MR görüntüsünde tümör tespitinde iyileştirmeler elde edilmeye çalışılmıştır. Yapılan çalışmada görüntü verisi olarak, geliştirilen algoritmaların test edilmesi için özel hazırlanan, tümörlü bölgeleri belli MR görüntüsü kullanılmıştır. Uygulamalarda bulanık sistemin klasik sistemlere göre karar aşamasında çok daha başarılı olduğu sonucuna varılmıştır.

Beyin MR görüntüsünde tümörlü bölgenin tespiti için öncelikle kümeleme algoritmaları denenmiş ve görüntü gri seviyesine göre farklı bölgelere ayrılmaya çalışılmıştır. İlerleyen safhalarda kural tabanlı bulanık sistem ile görüntüdeki tümörlü bölge sınıflandırılmıştır. Kural çıkarımı için yine kümeleme algoritmalarından faydalanılmıştır.

Yapılan testler kümeleme algoritmaları arasında kesin bir üstünlükten söz edilemeyeceğini göstermiştir. Eldeki veriye göre farklı algoritmalar başarılı sonuçlar verebilmektedir. Kümeleme algoritmalarındaki başarıda en büyük etken algoritma için gerekli parametrelerin isabetli bir şekilde kestirilmesidir.

Bu sonuçlar çerçevesinde, uygulanan kural tabanlı bulanık sistemin en iyi performansı göstermesi için kural çıkarımında kullanılan bulanık c-ortalama kümeleme algoritması farklı parametreler ve görüntüden elde edilen farklı özellikler ile denenmiş ve başarılı sonuçlar elde edilmiştir.

TUMOR DETECTION IN MR BRAIN IMAGES USING FUZZY CLASSIFICATION

Arslan ALAN

Keywords: Fuzzy Systems, Classification, Clustering, Segmentation, Image Processing.

Abstract: The goal of this work is to improve the performance of the detection of tumor areas in MR brain images using the advantages of fuzzy system. We have made available synthetic MR brain images presenting tumor, that is generated by inserting pathology into a healthy MR brain image with known ground truth. That is provide us to testing the algorithms developed. At the end of work we have had a decision that fuzzy systems has advantages over the crisp systems.

To detection of tumor areas we have implemented clustering algorithms, and segmented the image different areas in the first section. Secondly, the tumor areas classified using rule based system. At this step, clustering algorithms were used for extracting the rules.

Experimental results show that, there is no superiority between clustering algorithms. The most important fact superioing the clustering algorithm is that estimating the parameters of algorithms properly. One more point to keep in mind is that different clustering algorithms could be useful for different data.

We have tried fuzzy c-means (fcm) algorithm with different parameters for extracting of the rules for fuzzy rule based system in order to have the best classification results.

1. GİRİŞ

Hızla gelişen teknoloji paralelinde insanların hayatına birçok yenilikler girmektedir. Günlük yaşantıda, çalışma hayatında ve benzeri birçok ortamda bu yenilikleri görmekteyiz.

Bu yeniliklerden bir tanesi belki de en önemlisi tıp alanında bulunan yeniliklerdir. Bir hastalığın tedavisi için bulunan bir ilaç veya tedavi yöntemi binlerce belki milyonlarca insana umut ışığı olmaktadır. En az bulunan bu ilaç ve yöntemler kadar önemli olan bir başka unsur ise hastalığın erkenden teşhis edilmesidir. Bu durumda belki de bu ilaçlara bile gerek kalmadan tedavi uygulanması mümkün olacaktır. Bu tez çalışmasında da bu amaçla doktorların beyin MR görüntüsünde tümör tespiti için, hızlı ve daha doğru bir tespit mekanizması geliştirilmeye çalışılmıştır.

Bilgisayar destekli geliştirilen sistemlerle yapılan tespitler hem daha doğru olacak ve bu da hastalığın tedavisine bir an önce başlanmasını sağlayacak hem de daha hızlı olacak ve böylece doktorlar kazandıkları zamanı daha iyi bir hizmet vermek için kullanabileceklerdir.

Tümör tespiti için farklı yöntemler geliştirilmektedir. Bu yöntemlerde piksel değerleri üzerinden çıkarılan özellikler çoğu durumda yetersiz kaldığından simetri ve yapısal nicelikler gibi farklı özellikler de etkili olmaktadır (Xuan ve Liao, 2007). Bu noktada da tümörlü bölgeyi en iyi ayıracak özellikleri çıkarıp bunların arasından en efektif olanlarını ayırmak önem arz etmektedir. Özellik çıkarımı için Bağımsız Bileşen Analizi Li ve diğ. (2008) ve Dalgacık Dönüşümü Kharrat ve diğ. (2009) gibi yöntemler de uygulanmaktadır. Bazı yöntemlerde MR görüntüsünde bulunan kafatasına ait kısımların atılması amacıyla görüntüye ön işlem uygulanmakta ve sadece beynin iç bölgesine ait görüntü elde edilmektedir (Gordillo ve diğ., 2010). Elde edilen özelliklerden tümör tespiti için kümeleme algoritmaları da sıkça kullanılmaktadır. Özellikle bulanık c-ortalama algoritması (FCM – Fuzzy C-Means) diğer kümeleme algoritmalarına göre oldukça başarılıdır (Hemanth ve diğ., 2009).

Farklı kümeleme algoritmaları ile de benzer veya daha başarılı sonuçlar elde edilebilmektedir (Verma ve diğ., 2009).

MR görüntü sınıflandırmada en temel sorunlardan bir tanesi geliştirilen sistemlerin test edilmesi için standart bir verinin olmamasıdır. Çoğu yeni metotta test için uzman doktorlar tarafından belirlenmiş tümör ile otomatik olarak tespit edilen tümör karşılaştırılmaktadır. Ancak bu durumda algoritmaların başarısını karşılaştırmak güç olmaktadır. Bu nedenle geliştirilen sistemde Prastawa ve diğ. (2009) tarafından sunulan tümörlü bölgesi belirli olan yapay MR görüntüsü kullanılmıştır.

Görüntüdeki tümörlü bölgelerin sınıflandırılmasına başlanmadan önce örüntü tanıma kavramı ikinci bölümde detaylı olarak incelenmiştir. Bu aşamada örüntü tanımayı etkileyen süreçler anlatılmıştır. Burada dikkat çekmemiz gereken en önemli noktalar; örüntü tanıma sistemlerinde öncelikle verinin iyi analiz edilmesi ve veriyi ayrıştıracak en uygun özelliklerin belirlenmesi gerekliliğidir. Veriyi ayrıştıracak özellikler bulunmadan en iyi kümeleme veya sınıflandırma algoritmaları bile geliştirilse istenen seviyeye ulaşmak pek mümkün görülmemiştir.

Üçüncü bölümde yine örüntü tanıma sürecinin içinde yer alan kümeleme yöntemleri incelenmiştir. İkinci bölümde klasik yöntemlerine değinilen kümeleme analiziyle ilgili insan düşünce sistemine yakın bir yaklaşım sunmak için bulanık yöntemler irdelenmiştir. Bu aşamada tespit edilen en önemli husus ise; kümeleme yöntemlerinin birbirlerine tam bir üstünlük sağlayamaması buna karşılık performansı esas etkileyen öğenin kümeleme algoritmaları için gerekli parametrelerin kestirilmesindeki başarımlı olduğudur. Hangi kümeleme algoritmasını kullandığınızdan çok hangi parametreleri kullandığınız ve bu parametreler için bulduğunuz değerler burada önem kazanmaktadır.

Dördüncü bölümde yine ikinci bölümde anlatılan klasik sınıflandırma işlemleri için bulanık yaklaşımlar sunulmuştur. Bu aşamada özellikle bulanık kural tabanlı sistem üzerinde durulmuştur. Bunun nedeni; bulanık kural tabanlı sistemin hem bulanık mantığın getirdiği avantajları sağlaması hem de oluşturulan kurallar ile her seviyedeki insanın kurallara bakarak yapılan işlem hakkında bir fikre sahip olmasının sağlanmasıdır. İkinci bahsedilen husus, sistemi sonradan inceleyen kişiler için çok bir anlam ifade etmeyen karmaşık matematiksel işlemler arasında boğulan

bir yapı yerine daha anlamlı bir yapı sunmaktadır. Burada önemli olan oluşturulan kurallarda kullanılan dilsel değişkenler altındaki sayısal değerleri, yani üyelik fonksiyonlarını en uygun şekilde oluşturabilmektir. Bu amaçla karar ağaçları, kümeleme algoritmaları gibi farklı yöntemler kullanılmakta olup bu tez çalışmasında kümeleme algoritmaları bu amaçla kullanılmıştır.

Beşinci bölümde yine ikinci bölümde de anlatılan görüntü bölütleme işleminin bulanık yaklaşımla nasıl yapıldığı incelenmiştir. Burada elde edilen en önemli bulgu klasik yöntemde görüntü bölütleme yapıldığında iki bölge arasında kalan piksellerin o an için karar verilerek bir bölgeye bağlandığı ancak aslında diğer bölgeden de olabileceği ihtimalinin göz ardı edildiğidir. Bulanık yöntemde ise en son karar aşamasına kadar piksellerin hangi oranda hangi bölgeye ait olduğu bilgisi saklanmakta ve karar aşamasında elde edilen başka sonuçlar da değerlendirilerek en uygun karar verilmektedir. Bu hususun bulanık mantığın getirdiği en önemli faydalardan biri olduğu düşünülmektedir.

Altıncı bölümde tez çalışması boyunca yapılan deneyler ve elde edilen sonuçlar ile çalışmada kullanılan veri kümeleri detaylı olarak açıklanmıştır.

Son bölümde ise yapılan tüm araştırmalar ve deneylerden elde edilen bilgiler ışığında varılan sonuçlar açıklanmış ve ileride yapılması gereken çalışmalar ile incelenmesinde fayda görülen hususlar hakkında yönlendirici bilgiler sunulmaya çalışılmıştır.

2. ÖRÜNTÜ TANIMA

2.1. Örüntü Tanıma Nedir?

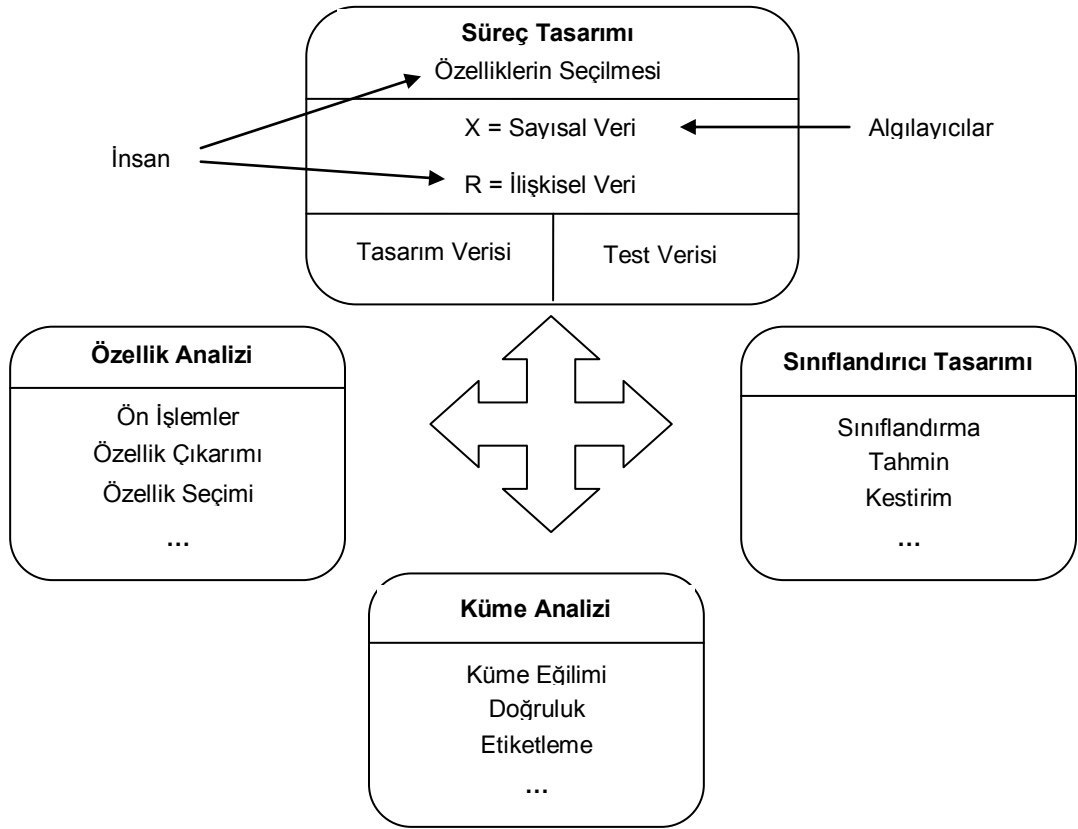
Örüntü tanıma için birçok tanım bulunmaktadır. Konunun daha iyi anlaşılabilmesi için farklı yaklaşımlar içeren bu tanımlar verilecektir.

Örüntü tanıma iki bölümden oluşmaktadır: bu bölümler çıkarılacak özelliklerin seçilmesi ve sınıflandırıcı tasarımı yapılmasıdır (Fukunaga, 1972). Örüntü tanıma, verinin, arka planda bulunan önemsiz detaylardan arındırıldıktan sonra, önemli özelliklerinin çıkarılması yoluyla farklı sınıflara ayrıştırılmasıdır (Gonzalez ve Thomason, 1978). Örüntü tanıma verinin içinde bulunan yapıların aranmasıdır (Bezdek, 1981). Örüntü tanıma, ölçümlerin tanımlanması veya sınıflandırılması ile ilgilenen bir bilim dalıdır (Schalkoff, 1992). Örüntü tanıma, sınırları net olmayan, farklı bir ifadeyle bulanık olan, çok geniş bir alanda faaliyet göstermektedir (Devijer ve Kittler, 1982).

Basit ve en doğru tanımını yapmak gerekirse, örüntü tanımının, özellik analizi, kümeleme ve sınıflandırıcı tasarımı ile ilgili olduğu, daha da ötesi bu üç disiplini içinde barındıran modellerden oluştuğu söylenebilir (Bezdek ve diğ., 2005) .

Örüntü tanıma için sayısal ve sözdizimsel yaklaşım olmak üzere iki temel yaklaşım vardır. Örneğin, özellik çıkarımında sayısal yaklaşımda hayvanlar bacak sayıları, ağırlıkları gibi sayısal nicelikleri ile sözdizimsel yaklaşımda ise dış yüzeylerine göre tüylü, kabuklu ve benzeri şekilde ifade edilebilmektedir.

Örüntü tanıma için sayısal yaklaşım Şekil 2.1'de görüldüğü gibi dört ana bölüme ayrılabilir (Bezdek ve diğ., 2005). Bu ana bölümler birbirlerinden bağımsız değildirler. Başarılı bir örüntü tanıma sisteminin geliştirilebilmesi için, belirlenen kısıtlar sağlanıncaya kadar, yinelemeli bir şekilde bu bölümler arasında dolaşım sağlanmaktadır.



Şekil 2. 1: Sayısal örüntü tanımanın bölümleri

Şekil 2.1'de gösterilen süreç tasarımı genellikle insanlar tarafından gerçekleştirilir. Bu bölümde yapılması gerekenler, süreç için bir model belirlenmesi, ölçülecek özelliklerin ve verileri toplayacak algılayıcıların seçilmesidir. Bu bölümde alınan kararlar sistemin performansına yüksek oranda etki yapacağından karar alırken dikkatli bir şekilde düşünmek gerekmektedir.

Örüntü tanımanın diğer bölümleri kısaca şöyle açıklanabilir:

Özellik Analizi, algılayıcılardan alınan veri üzerinde gürültünün giderilmesi, görüntü zenginleştirme gibi ön işlemleri, elde edilen yeni veriden özellik çıkarımını ve çıkarılan özelliklerin yeterli seviyede küme ayırımını sağlayabilecek nitelikte olanların seçimini ve benzeri işlemleri kapsamaktadır. Almeida ve Sousa (2006), verilerden elde edilen çok sayıda özelliklerden en uygun olanlarını seçmek için c-ortalama kümeleme algoritmasını kullanmışlardır.

Özellik seçimi için, genellikle mevcut olan tüm özellikler veriden elde edilir ancak bu özelliklerden bazıları eldeki problem için önemli olmayabilir. Bu aşamada sonuca büyük oranda etki eden özelliklerin saklanması diğerlerinin ise çıkarılması gerekmektedir. Böylece veri boyutu azaltılmış olmaktadır. Veri boyutu azaltılmadan önce ve azaltıldıktan sonra yapılacak analiz sonuçlarının aynı olması önem arz etmektedir (Almeida ve Sousa, 2006).

Küme Analizi, seçilen özelliklerin veriyi kümelemeye elverişli olup olmadığı, yapılan kümelemenin doğruluğu ve verinin elde edilen kümelere göre etiketlenmesi ve benzeri işlemlerdir.

Sınıflandırıcı Tasarımı ise, çıkarılan özelliklerden elde edilen ilk örneklere göre veri kümelerinin sınıflandırılması, ilk örneklere göre sınıflandırıcı fonksiyonlarının tahmin edilmesi ve elde edilen sonuçların kontrol edilmesi ve benzeri işlemlerdir.

Sayısal örüntü tanımada kullanılan iki çeşit veri vardır. Bunlar obje verisi ve ilişkisel veridir. Örnek vermek gerekirse obje veri özellik vektörleri, ilişkisel veri ise benzerlikler, yakınlıklar ve benzeri yaklaşımlardır. Objeye veri $X=\{x_1, x_2, \dots, x_n\} \subset R^p$ şeklinde ifade edilir. Bu ifade R^p özellik uzayında n adet özellik vektöründen oluşan küme anlamına gelmektedir (Bezdek ve diğ., 2005).

Örüntü tanımada en temel yapılardan birisi etiket vektörüdür. Hangi türde verinin (obje, ilişkisel vb.) elde edildiğinden bağımsız olarak üç çeşit sınıflandırma etiketinden bahsedilebilir. Bunlar klasik (0 veya 1), bulanık ve olasılık (probabilistic) etiketlerdir. Noktaların sayısı (veya özellik vektörleri) n ve sınıf sayısı da c ($1 < c < n$) olmak üzere; etiket vektörleri R^c uzayında aşağıdaki şekilde tanımlanmaktadır (Bezdek ve diğ., 2005):

$$N_{pc} = \{y \in R^c : y_i \in [0, 1] \forall i, y_i > 0 \exists i\} = [0, 1]^c - \{0\} \quad (2.1)$$

$$N_{fc} = \left\{ y \in N_{pc} : \sum_{i=1}^c y_i = 1 \right\} \quad (2.2)$$

$$N_{hc} = \{y \in N_{fc} : y_i \in \{0, 1\} \forall i\} = \{e_1, e_2, \dots, e_c\} \quad (2.3)$$

$$N_{hc} \subset N_{fc} \subset N_{pc} \quad (2.4)$$

Eğer y etiket vektörü $x \in \mathbb{R}^p$ verisi için bulanık kümeleme metotları tarafından oluşturulmuşsa (örneğin bulanık c-ortalama metodu) o zaman y vektörüne x verisi için bulanık etiket vektörü denir. Aynı şekilde olasılık yöntemleriyle elde edilirse o zaman y olasılık etiket vektörüdür denir.

Denklem (2.1)'de olasılık yöntemleriyle oluşturulan etiket vektörü ifade edilmektedir. Her objenin farklı kümelerine ait olma olasılığı hesaplanarak bulunmaktadır. Bu etiket vektörü "0" ile "1" arasındaki sayılardan oluşmaktadır. Denklem (2.2) ise bulanık etiket vektörünü ifade etmektedir. Bulanık etiketler de yine "0" ile "1" arasındaki sayılardan oluşmaktadır ancak bir obje için verilen etiket değerlerinin toplamı 1'e eşittir. Denklem (2.3) ise klasik "0" veya "1" etiketlerini ifade etmektedir. Bu etiketlere göre bir obje bir kümeye ya aittir ya da değildir. Denklem (2.4)'te ise etiket vektörlerinin aralarındaki ilişki gösterilmektedir.

Çoğu örüntü tanıma modelleri verideki kümelerin istatistiksel veya geometrik özelliklerini bulma temeline dayanmaktadır. Geometriyi tanımlayan iki ana unsurdan bir tanesi açı diğeri ise uzaklıktır. Uzaklık hesaplanması için farklı yöntemler bulunmaktadır. Euclidean, City-Block, Chessboard, Mahalanobis bunlardan bazılarıdır.

Uzaklık ölçüm normlarının birbirlerine tam bir üstünlükleri bulunmamaktadır. Elde edilen veriye göre hangi uzaklık normunun kullanılacağına karar vermek daha uygun gözükmektedir. Hiç şüphe yok ki en çok bilinen ve tercih edilen uzaklık normu Öklit uzaklığıdır. Eğer veride bulunan noktaların değer aralığı yoğun şekilde çeşitlilik gösterirse Mahalanobis uzaklığının kullanımı daha uygundur.

Sınıflandırıcıyı $D: \mathbb{R}^p \rightarrow N_{pc}$ şeklinde bir fonksiyon olarak tanımlamak mümkündür. Yani sınıflandırıcı özellik uzayından noktaların hangi kümeye ait olduğunu gösteren etiket uzayına bir geçiş fonksiyonu olarak ifade edilebilir. $y = D(z)$ ifadesi \mathbb{R}^p özellik uzayında tanımlı z için etiket vektörüdür. Sınıflandırıcılar da oluşturduğu etiket vektörüne göre klasik, bulanık veya olasılık sınıflandırıcı olarak gruplandırılabilir. Sınıflandırıcı tasarımı iyi bir D fonksiyonu oluşturabilmek için parametrelerin bulunmasıdır. Bu parametreler verilerden veya deneyimlerden elde edilebilir. Parametrelerin elde edildiği veri etiketliyse D fonksiyonunu bulmak eğitilmiş öğrenme

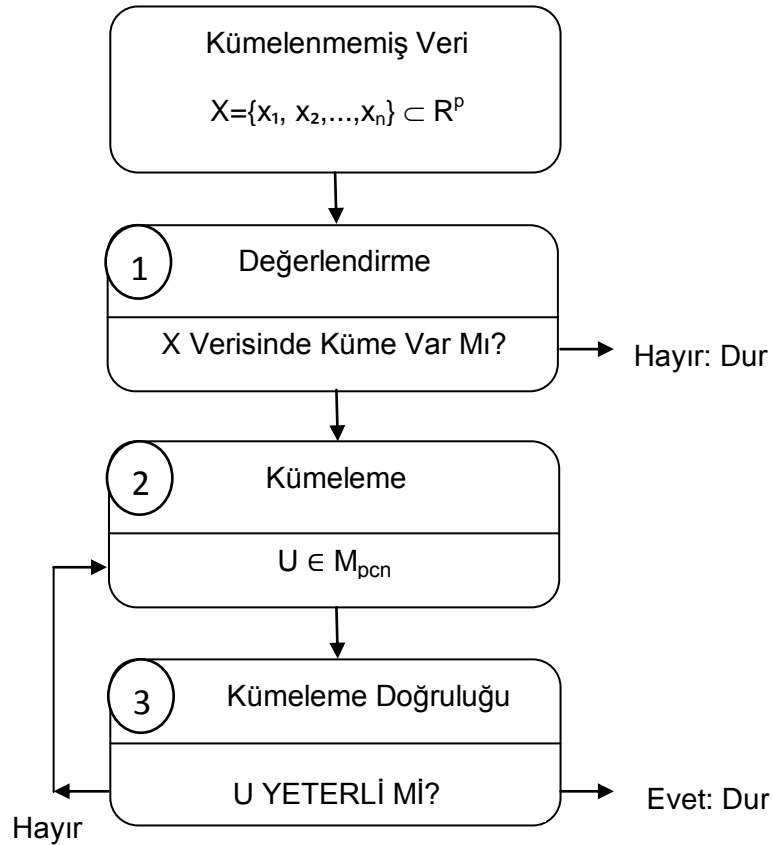
(supervised learning), değilse eğitimsiz öğrenme (unsupervised learning) olarak adlandırılır.

Sınıflandırma veya kümeleme işleminin amacı son aşamada kesin bir sınıf atamasına karar vermek olduğundan sınıflandırma algoritmaları tarafından üretilen N_{pc} ve N_{fc} etiket vektörleri klasik etiketlere dönüştürülür. Çoğu klasik olmayan sınıflandırıcılar $H: N_{pc} \rightarrow N_{hc}$ fonksiyonuyla klasik hale getirilirler.

$$H(y) = e_i \leftrightarrow \|y - e_i\| < \|y - e_j\| \leftrightarrow y_i > y_j ; j \neq i \quad (2.5)$$

H fonksiyonu öklit uzaklığı hassasiyetiyle y 'e en yakın üyelikleri alarak klasik etiket vektörünü oluşturur. Klasik hale dönüştürülecek etiketler eğer bulanık ise bu işleme durulaştırma (defuzzification) adı verilir.

2.2. Kümeleme



Şekil 2. 2: Küme analizinde üç aşama

Küme analizinin amacı objeleri aralarında bulunan benzerliklere göre sınıflandırma ve veriyi gruplara ayırma olarak tanımlanabilir (Almeida ve Sousa, 2006).

Şekil 2.2 kümeleme analizini tanımlamaktadır. Kümeleme analizinde öncelikle, alınan veride alt kümeler olup olmadığı sorusuna yanıt aranmaktadır. Eğer veride alt kümeler varsa veri kümelemeye eğilimlidir denir ve kümeleme işlemine başlanır. Literatürde veride alt kümeler olup olmadığı ile ilgilenen az sayıda metod bulunmaktadır (Bezdek ve diğ., 2005).

Kümeleme işlemine başlanmasına karar verildikten sonra, insanların algılama hassasiyetine benzer şekilde, yapıları bulabilen bir matematiksel benzerlik ölçüm modeli seçilmesine ihtiyaç duyulmaktadır. Burada hangi benzerlik kıstaslarının kullanılacağına belirlenmesi kümeleme modellerinin en önemli kısmıdır. Benzerlik kıstasları için genel kıstas ve lokal kıstas teknikleri bulunmaktadır.

Farklı kümeleme algoritmaları veriye uygulandığında farklı sonuçlar elde edilebilir. Hangi sonucun daha iyi olduğu ise belirsiz bir durumdur. Şekil 2.2'de belirtilen kümeleme aşaması, farklı bir ifadeyle eğitimsiz öğrenme (unsupervised learning), klasik (0 veya 1) veya bulanık (0-1 aralığında) etiket vektörlerinin (label vectors) $X=\{x_1, x_2, \dots, x_n\} \subset R^p$ verisindeki $\{x_k\}$ noktalarına atanarak kümeleme işleminin yapılmasıdır. Buradaki öğrenme (learning) kelimesi her küme için iyi örneklerin öğrenilmesini ifade etmektedir (Bezdek ve diğ., 2005).

Veri kümesi X 'in c kümeden oluştuğu varsayılırsa U üyelik matrisi $c \times n$ boyutlu (c verideki küme, n nokta sayısı) $U=[U_1 U_2 \dots U_n]=[U_{ik}]$ olacaktır. U_k , U matrisindeki n adet sütundan k 'inci sütunu ifade etmektedir. Bu aşamada farklı etiket vektörü modellerinden söz etmek mümkündür. Bunlar klasik etiketlerden oluşan (0 veya 1) vektörler, bulanık (0-1 aralığında) vektörler veya olasılık (0-1 aralığında) vektörleridir. Bu vektörlere ait üyelik matrisleri aşağıda verilmiştir (Bezdek ve diğ., 2005).

Aşağıdaki denklemler X verisine ait c adet kümeyi farklı etiketleme modellerine göre tanımlamaktadır. U matrisinin her sütunu etiket vektörüdür. Burada M üyelik matrisini, h , 0 veya 1 olan klasik üyelik etiketlerini, f bulanık üyelik etiketlerini, p

olasılık üyelik etiketlerini, c sınıf (küme) sayısını ve n ise X'te kaç adet nokta olduğunu göstermektedir.

$$M_{pcn} = \{U \in R^{cn} : U_k \in N_{pc} \forall k; 0 < \sum_{k=1}^n U_{ik} \forall i\} \quad (2.6)$$

$$M_{fcn} = \{U \in M_{pcn} : U_k \in N_{fc} \forall k\} \quad (2.7)$$

$$M_{hcn} = \{U \in M_{fcn} : U_k \in N_{hc} \forall k\} \quad (2.8)$$

Yukarıda denklemleri verilen etiketleme modellerinin daha iyi anlaşılabilmesi için (Bezdek ve diğ., 2005) tarafından sunulan aşağıdaki örnek incelenecektir.

X verisi $\{x_1 = \text{şeftali}, x_2 = \text{erik}, x_3 = \text{nektarin}\}$ olarak verilsin ve bu verinin iki kümeye ayrılması istensin ($c=2$). Bu üç obje şu şekilde kümelenecektir:

Tablo 2. 1: Klasik, Bulanık ve Olasılık etiketlerinin gösterimi

| | $U_1 \in M_{h23}$ (Klasik etiket gösterimi) | | | $U_2 \in M_{f23}$ (Bulanık Etiket gösterimi) | | | $U_3 \in M_{p23}$ (Olasılık etiket gösterimi) | | |
|----------------|--|-------|-------|---|-------|-------|--|-------|-------|
| objeler | x_1 | x_2 | x_3 | x_1 | x_2 | x_3 | x_1 | x_2 | x_3 |
| Şeftali | 1.0 | 0.0 | 0.0 | .0 | 0.2 | 0.4 | 1.0 | 0.2 | 0.5 |
| Erik | 0.0 | 1.0 | 1.0 | 0.0 | 0.8 | 0.6 | 0.0 | 0.8 | 0. |

Tablo 2.1 incelendiğinde; etiketleme işlemi klasik yöntemle (0 veya 1, M_{h23}) yapılırsa nektarin meyvesine (x_3) şeftali sınıfı için 0 üyelik erik sınıfı için ise 1 üyelik değeri verildiği görülmektedir (algoritma gereği bir tercih yapılması zorunludur). Nektarin meyvesinin erik grubuna girmediği aşikârdır, dolayısıyla bu etiketleme işleminin hatalı sonuç ürettiği anlaşılmaktadır. Diğer modellere bakıldığında (M_{f23} ve M_{p23}) ise; bulanık modelde (M_{f23}) nektarin meyvesinin (x_3) 0.4 üyelik derecesiyle şeftali sınıfından olduğu, 0.6 üyelik derecesiyle erik sınıfından olduğu, olasılık modelinde M_{p23} ise üyelik derecelerinin 0.5 ve 0.6 olduğu görülmektedir. Peki, gerçekte durum nedir? Nektarin meyvesi dışı tüysüz olduğu için bir miktar erik sınıfına benzemekte ve bir miktar da şeftali sınıfına benzemektedir. Yani bulanık ve olasılık modelleri gerçeğe daha yakın sonuçlar üretmekte, en azından detay bilgileri saklamaktadır.

Bulanık ve olasılık modellerindeki üyelik değerleri en büyük üyelik derecesine göre düzenlenirse (örneğin x_3 objesinin bulanık olan üyelik dereceleri en büyük olanın seçilmesi şeklinde düzenlenirse (defuzzification, durulaştırma) 0.4 ve 0.6 olan üyelik dereceleri $0.4 < 0.6$ olduğu için $0.4 = 0$ ve $0.6 = 1$ olacaktır) birinci modelle aynı sonuçların elde edildiği görülmektedir. Yeniden düzenlenen üyelik derecelerinde artık ince detaylar bulunmayacaktır (çünkü sadece 0 ve 1 olacak). Dolayısıyla, bu detaylar kaybedilmeden kullanılarak, veriden bilgi çıkarımı işlemi gerçekleştirildikten sonra en son aşamada bulanık değerlerin durulaştırılması sağlanmalıdır.

Bulanık ve olasılık üyelik dereceleri, objeyi bir sınıfa atamaktansa, farklı sınıflar için üyelik dereceleri birbirine yakınsa o objeye farklı bir açıdan bakılması gerekliliğini ortaya koymaktadır. Bu örnekte de nektarin ne şeftali ne de eriktir, dolayısıyla U_2 ve U_3 'teki üyelik dereceleriyle sınıfların belirtilmesi gerçeğe daha uygundur.

M_{pcn} ve M_{fcn} , gerçek obje sınıflarındaki sınırların belirsizliğinden dolayı (bulanık olmasından dolayı), M_{ncn} 'ye göre daha gerçekçi bir yaklaşım sunmaktadır (Bezdek ve diğ., 2005).

2.2.1. Kümeyi temsil eden örnek nokta (ilk örnek) ile kümeleme modelleri

Konu için kullanılan başlığı daha anlamlı hale getirmek için biraz açıklamak gerekirse, örnek noktadan kastedilen anlam bir kümeyi en iyi temsil edebilen, o kümenin tam merkezinde bulunan noktanın belirlenmesidir. Bu modele uygun kümeleme algoritmaları belirlenen bu nokta etrafında şekillenmektedir.

Kümeleme algoritmaları X verisinde bulunan c adet küme için U üyelik değeri matrisini oluştururlar. Çoğu kümeleme algoritması U matrisine ilave olarak farklı parametreler de üretir. En genel parametreler $V = \{v_1, v_2, \dots, v_c\} \subset R^p$ ile ifade edilen vektörlerdir. v_i vektörü i kümesi ile ilişkilendirilen noktalar için ilk örnek (küme merkezi, imza, şablon) olarak tanımlanabilir. Yani kümeyi en iyi temsil eden noktalardır. İlk örnekler, küme yapılarının sıkıştırılmış bir gösterimi olarak kabul edilebilir. Yinelemeli (iterative) örnek nokta algoritmalarının temel formu şöyledir (Bezdek ve diğ., 2005):

$$(U_t, V_t) = e(X: U_{t-1}, V_{t-1}), t > 0 \quad (2.9)$$

ϵ kümeleme algoritmasını, t ise yinelemeli algoritmada hangi adımda bulunduğunu belirtir. Kümeleme algoritmalarında başlangıç ve bitiş kısıtları belirlenirken U veya V kullanılabilir. Daha açık ifade etmek gerekirse; algoritma başlarken üyelik değerleri rastgele atanarak bu değerlerle V vektörü (örneğin küme merkezleri) oluşturulur, sonraki adımda V vektörü kullanılarak U üyelik değerleri yeniden hesaplanır ve U matrisi istenilen kısıtları sağlayıncaya kadar algoritma devam eder. Aynı şekilde başlangıçta U üyelik değerleri yerine V vektörü rastgele atanarak bu vektöre göre U üyelik matrisi oluşturulur ve bitiş kısıtı V vektörü ile belirtilir. U ve V parametrelerinin birlikte kullanıldığı modellere örnek olarak klasik ve bulanık c-ortalama modelleri verilebilir.

2.2.1.1. K (C)-ortalama modeli

K (c)-ortalama modelleri en çok bilinen ve gelişim gösteren modellerdir. Amaç, veri kümesindeki objeler (noktalar, pikseller) ile küme merkezleri (V) arasındaki uzaklıklar toplamının minimize edilmesidir. Uzaklık ölçümleri D ile, etiket matrisi U ile ve uzaklıklar toplamı ise J ile gösterilmektedir. Bu modeller için optimizasyon problemi klasik etiketleme yöntemine göre Denklem (2.10) ile aşağıdaki şekilde tanımlanır (Bezdek ve diğ., 2005):

$$\min_{(U, V)} \{J(U, V) = \sum_{i=1}^c \sum_{k=1}^n u_{ik} D_{ik}^2\} \quad (2.10)$$

$U \in M_{hcn}$, klasik üyelik matrisi

$V = (v_1, v_2, \dots, v_c) \in R^{cp}$; $v_i \in R^p$, i nci obje için örnek nokta (küme merkezi)

$$D_{ik}^2 = \|x_k - v_i\|_A^2 \quad (2.11)$$

Bu problemi açıklamak gerekirse; U üyelik matrisinde her nokta için hangi kümeye ait olduğunu gösteren etiketler bulunmaktadır. Böylece, her obje için ait olduğu küme merkezine uzaklığını hesaplamak mümkündür. Uzaklık hesaplanması için farklı uzaklık normları kullanılabilir. Burada Denklem (2.11) ile ifade edilen öklit uzaklık normu kullanılmaktadır. Bu problemde amaç hesaplanan bu uzaklıklar toplamının minimize edilmesidir.

U üyelik matrisi ve V küme merkezlerinin hesaplanması Denklem (2.12) ve (2.13) ile aşağıda verilmiştir.

$$U_{ik} = \begin{cases} 1; & D_{ik} \leq D_{ij}, j \neq i \\ 0; & \text{değilse} \end{cases} \quad \forall i, k; \quad (2.12)$$

$$v_i = \frac{\sum_{k=1}^n u_{ik} x_k}{\sum_{k=1}^n u_{ik}} = \frac{\sum_{x_k \in X_i} x_k}{n_i} = \bar{v}_i \quad \forall i \quad (2.13)$$

Etiket matrisi (U) oluşturulurken öncelikle noktanın tüm küme merkezlerine uzaklığı ölçülür. Ölçülen uzaklıklar sonucunda nokta hangi kümeye yakınsa etiket matrisinde (U) nokta ile o kümenin kesişim noktasındaki değer "1" yapılır, objenin diğer kümelerle kesiştiği noktaların değerleri ise "0" yapılır. Böylece tüm objeler en yakın olduğu kümeye dâhil edilmiş olur (Denklem (2.12)).

Daha sonra Denklem (2.13) ile oluşturulan etiket matrisinde(U) "1" ile işaretlenen objeler her küme için ayrı ayrı kullanılarak küme merkezleri(V) tekrar hesaplanır.

Tüm bu anlatılanlardan sonra k-ortalama algoritması Tablo 2.2'de özet olarak sunulmuştur.

Algoritmanın adımlarını açıklamak gerekirse, öncelikle kümelenecek veri algoritmaya kaydedilir. Veride her nokta (piksel) için kaç adet özellik mevcutsa p sayısı bu adedi ifade eder. Daha sonra algoritma tarafından kullanılacak olan birtakım değerler belirlenir.

Bu değerleri açıklamak gerekirse, öncelikle algoritmanın bulacağı küme sayısı belirtilmelidir. Bu sayı 1'den büyük ve nokta sayısı (piksel sayısı) olan n'den küçük olmalıdır. İkinci olarak, algoritmanın sonlanabilmesi için yeni bulunan küme merkezleri ile bir öncekiler arasındaki farkın belirli bir değerden (ϵ) küçük olması gerekmektedir veya bu değere ulaşılamasa da belirli bir sayıda adımdan (T) sonra algoritma son bulmalıdır. Bahsedilen ϵ ve T değerleri belirlenmelidir.

Tablo 2. 2: Klasik k-ortalama algoritması

| | |
|--------|--|
| Adım 1 | Etiketlenmemiş veri girişi $X \subset R^p$ |
| Adım 2 | Küme sayısı (c) belirlenir: $1 < c < n$ Maksimum iterasyon sayısı belirlenir : T J_m için uzaklık ölçüm normu belirlenir: $\ x\ _A^2 = x^T A x$ Sonlandırma ölçümü: $E_t = \ V_t - V_{t-1}\ $ Sonlandırma için eşik değeri belirlenir: $0 < \varepsilon$ |
| Adım 3 | Başlangıç noktaları girilir: $V_0 = (v_{1,0}, \dots, v_{c,0}) \in R^p$ |
| Adım 4 | $t = 0$ Repeat U_t , Denklem (2.20) ile önceki adımdaki örnek noktalar (V_{t-1}) kullanılarak hesaplanır. V_t , U_t kullanılarak Denklem (2.21) ile yeniden hesaplanır. Until ($t=T$ veya $E_t \leq \varepsilon$) |
| Adım 5 | $U, V = U_t, V_t$ |

Algoritma çalışmaya başladığında küme merkezleri olarak kullanacağı (V) değerlere ihtiyaç duymaktadır. Başlangıç küme merkezlerinin belirlenmesi aslında algoritmanın başarısını da etkilemektedir. Tam net olmasa da gerçek merkezlere yakın küme merkezleri belirlemek algoritmayı daha başarılı kılacaktır. Küme merkezleri rastgele seçilebileceği gibi verinin ilk değerinden başlayarak küme sayısı kadar obje merkez olarak seçilebilir. Literatürde küme merkezlerini belirlemek için farklı yöntemler de uygulanmaktadır. Kimi bilim adamları tarafından basit bölümlenme algoritmaları ile küme merkezleri tahmini olarak belirlenerek kümeleme algoritmasına giriş değeri olarak verilmektedir (Bezdek ve diğ., 2005).

Algoritmanın son kısmında ise sonlanma kistası sağlanıncaya kadar her adımda noktaların yeniden etiketlenmesi ve küme merkezlerinin yapılan bu etiketlemeye göre yeniden hesaplanması işlemleri bulunmaktadır. Burada algoritmanın başında belirlenen ε ve T değerleri kullanılarak; küme merkezleri ile bir önceki adımda hesaplanan küme merkezleri arasındaki fark ε değerinden küçük veya bu değere eşit olacaktır kistası sağlanıncaya kadar veya bu kistas sağlanamazsa belirlenen T adım sayısına ulaşıncaya kadar algoritma devam edecektir.

2.2.1.2. Yarı eğitilmiş kümeleme modelleri

Her kümeyle ait olan üye noktaları, hataların karesi en düşük değerde olacak şekilde, en uygun seviyeye getirmek için J fonksiyonunu kullanmak iyi bilinen bir yöntemdir. Ancak burada can sıkıcı bir durum söz konusudur. J fonksiyonunun en küçük değere ulaştığı her durum X veri kümesinin en uygun şekilde kümelendiği anlamına gelmemektedir. Bu durumda algoritma için sonlanma kriteri olarak belirlenen J'nin en düşük değerde olması kriteri bazı durumlarda istenen sonucu elde etmeyi sağlayamamaktadır (Bezdek ve diğ., 2005).

Yarı eğitilmiş k(c)-ortalama kümeleme modelleri bu kriteri ortadan kaldırmak için geliştirilmektedir. Bu modellerde ilk etapta hangi kümeyle ait olduğu bilinen etiketli veri bulunmaktadır. Bu veriler geriye kalan verilerin doğru şekilde kümelenebilmesine yön vermektedirler (bu duruma özellikle tıbbi görüntülerde rastlanmaktadır). Denklem (2.14) bu durumu ifade etmektedir. Buna göre X giriş verisi hangi kümeyle ait olduğu bilinen X^d etiketli veri ile hangi kümeyle ait olduğu bilinmeyen X^u etiketsiz veriden oluşmaktadır.

$$X = \{ X^d_1, X^d_2, \dots, X^d_{nd} \} + \{ X^u_1, X^u_2, \dots, X^u_{nu} \} = X^d + X^u \quad (2.14)$$

etiketli veri etiketsiz veri

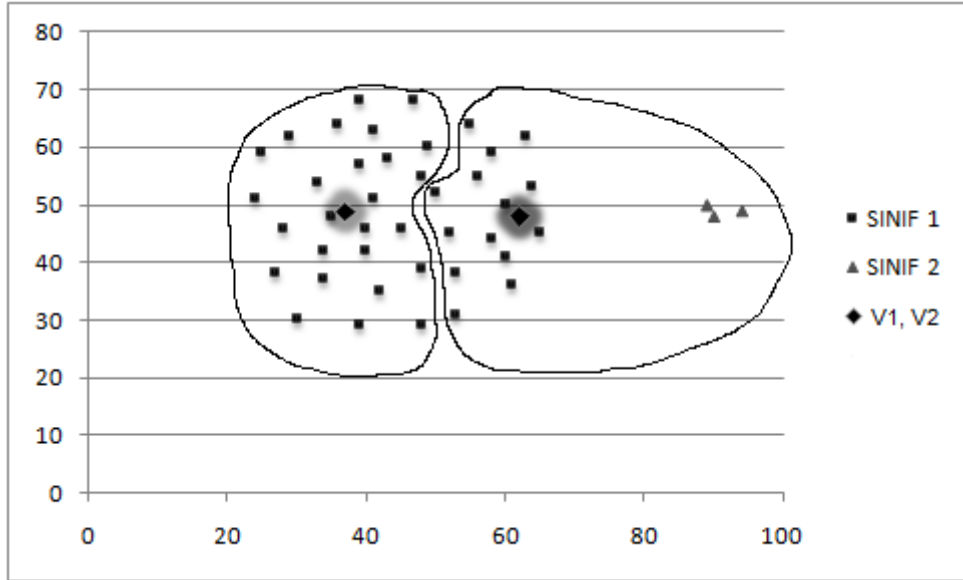
Burada yapılan işlemin sınıflandırma ile karıştırılmaması önemlidir. Eğitimsiz algoritmalarda başlangıçta oluşturulan U üyelik matrisindeki bazı noktaların üyelik değerleri, yarı eğitilmiş algoritmalarda ait oldukları kümelere uygun olarak algoritma başlamadan U üyelik matrisine girilmektedir. Böylece algoritma birinci adımda küme merkezlerini hesaplarken gerçek merkezlere yakın bir değer elde etmekte ve bu faktör algoritmanın başarımını artırmaktadır (Bezdek ve diğ., 2005).

Bazı yarı eğitilmiş kümeleme algoritmalarında eğitilmiş U matrisi kullanılarak küme merkezleri hesaplanmakta, bazılarında ise sadece eğitim için girilen etiketli veriler kullanılarak küme merkezleri hesaplanmaktadır.

Konunun daha iyi anlaşılabilmesi için Bezdek ve diğ. (2005) tarafından sunulan örneği buraya aktarmak faydalı olacaktır. Örnekte Bensaid ve diğ. (1996a) tarafından X_{43} adı verilen veri üzerinde $c=2$ parametresiyle (aranan küme sayısı iki)

eđitimsiz ve yarı eđitilmiş k(c)-ortalama kümeleme algoritmaları denenmiştir. X_{43} verisinin özelliđi 43 objenin 3 adedi bir kümeye ait diđer 3 adedi ise ayrı bir kümeye aittir. Veri kümesinin net bir şekilde iki kümeye ayrıldıđı Şekil 2.3'te gözükmemektedir.

Veri üzerinde eđitimsiz k(c)-ortalama kümeleme algoritması uygulandıđında net şekilde gözükmemek üzere iki kümenin birbirinden ayrıştırılmadıđı gözükmemektedir. Daha önce de belirtildiđi şekilde başlangıç deđerleri oluşturulurken rastgele seđilen küme merkezleri burada önemli bir etken durumundadır. Ayrıca J fonksiyonunun en düşük deđere ulaştırılmasının da en iyi kümeleme sonucuna ulaşıldıđı anlamına gelmeyeceđi de önceki açıklamalarda belirtilmişti. Tüm bu etkenler göz önüne alındıđında X_{43} benzeri bir veri ile karşılaşıldıđında istenilen sonuçların alınamadıđı gösterilmektedir. Bu durum Şekil 2.3'te gösterilmektedir (Bezdek ve diđer., 2005).

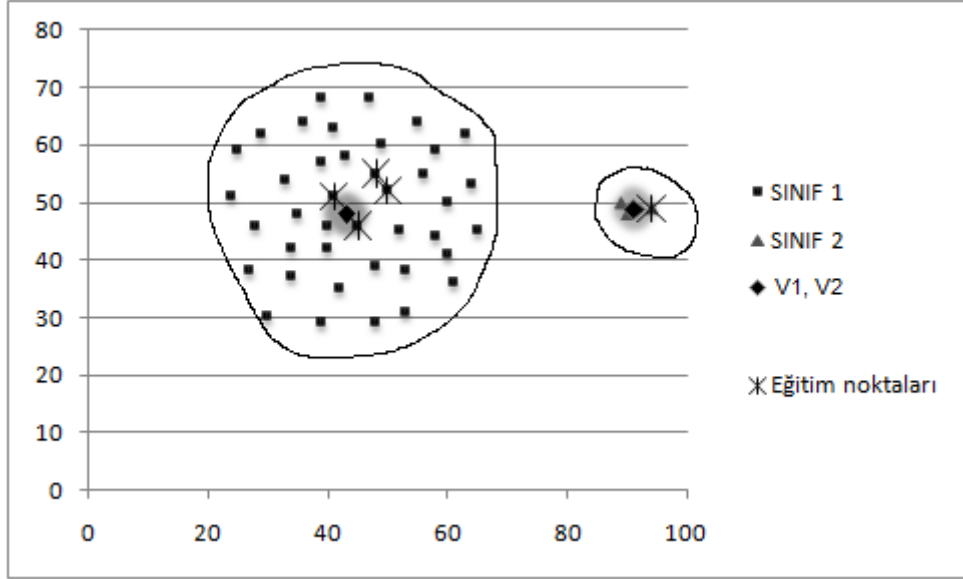


Şekil 2. 3: Eđitimsiz k(c)-ortalama algoritması sonucu

Yarı eđitilmiş k(c)-ortalama algoritmaları veri üzerine uygulandıđında 43 noktanın da dođru şekilde kümelendiđi Şekil 2.4'te gözlemlenmektedir. Yarı eđitilmiş k(c)-ortalama algoritmasında eđitim için 3 üyeli birinci kümeden bir, 40 üyeli ikinci kümeden ise dört nokta eđitim için seđilmiştir. Başlangıç küme merkezlerinin seđilen bu eđitim noktalarıyla oluşturulması algoritmanın başarımında önemli rol oynamaktadır. Eđitim için seđilen noktalar deđiştirildiđinde bazı durumlarda sonuçların kötüye gittiđi, farklı algoritmalar ve aynı eđitim noktalarıyla aynı sonuçların elde edilemediđi ve

parametreler deęiştirilerek bu algoritmaların da istenen seviyeye yakın sonuçlar üretebildiđi gözlemlenmiştir.

Buradan şöyle bir sonuç çıkarmak yanlış olmayacaktır. Çođu kümeleme algoritmaları dođru parametreler bulunduđunda benzer sonuçlar üretecektir. Ancak bu parametrelerin bulunması da ayrı ve zorlu bir aşamadır.



Şekil 2. 4: Yarı eğitimli k(c)-ortalama algoritması sonucu

2.3. Sınıflandırma

Sınıflandırma işlemini bir nevi kümeleme işlemindeki gibi veriyi gruplandırma olarak algılayabiliriz. Buradaki fark, veride özellikle belirli grupların aranmasıdır. Bu gruplara ait özellikler önceden belirlidir ve veri kümesi bu özelliklerle karşılaştırılarak aranılan gruba ait olan örnekler etiketlenir. Böylece veri kümesinde istenilen obje sınıflandırılmış olur.

Aranılan veri grubuna ait özelliklerin belirlenerek sınıflandırıcıya tanıtılmasına sınıflandırıcının eğitim safhası diyebiliriz. Bu safhada bir nesneyi en iyi belirten özellikler ve bu özelliklere ait deđer aralığı belirlenerek sınıflandırıcıya eğitim verisi olarak verilmektedir. Sonraki aşamalarda sınıflandırıcıya gelen veriler (test verisi veya gerçek veri) sınıflandırıcı tarafından eğitim safhasında oluşturduđu parametrelere göre sınıflandırılmaktadır.

Kümeleme işlemine aslında eğitimsiz sınıflandırma diyebiliriz. Çünkü kümeleme işleminde veride farklı gruplara ait nesnelere belirlenmektedir ancak bu nesnelere adı belli değildir. Sadece farklı gruptan olan nesnelere farklı etiketlerle etiketlenmektedir.

Sınıflandırma işleminde aranan sınıftan nesne bulunduğunda o nesne "1" ile diğer nesnelere ise "0" ile etiketlenmektedir. Böylece veri kümesindeki nesnelere sınıflandırılmaktadır. Ancak klasik "1" ve "0" birçok durumda yetersiz kalmaktadır. Çünkü bazı nesnelere aranan sınıfa tam olarak uymakta, bazıları ise %49, %60, %51 gibi oranlarla aranan sınıfa benzemektedir. Klasik yöntemlerde %51, %60 ve %100 aynı kabul edilmekte, %49 ise %51'e çok benzemesine rağmen farklı sınıftan kabul edilmektedir. Bu detay bilgilerin son karar aşamasına kadar saklanabilmesi için bulanık sınıflandırma yöntemleri geliştirilmiştir. Bulanık sınıflandırma dördüncü bölümde incelenecektir.

2.4. Bölütleme

Görüntü bölütleme, bilgisayar görmesi algoritmalarında önemli bir basamaktır. Bölütlemenin amacı görüntüyü anlamlı bölgelere ayırmaktır. Bölütleme adımında yapılacak hatalar üst seviyedeki görme algoritmalarını olumsuz yönde etkilemektedir. Otomatik bir sistemde istenen objenin arka plandan tamamen ayrılması bilgisayar görmesinin sonraki aşamaları için performansa çok büyük katkı yapmaktadır. Bu nedenle obje ve bölge tanımlamalarındaki belirsizlikler üzerine geliştirilen metotlar ile obje(bölge)leri ifade eden özelliklerin doğruluğu önem arz etmektedir (Bezdek ve diğ., 2005).

Bölütlenmiş görüntüde, idealde her bölüm gri seviyesi gibi karakteristik özellik bakımından benzer özelliklerde olmalı, komşu bölgeler ise farklı karakteristik özelliklere sahip olmalıdır.

Bölütlemeyi formüle ederek göstermek gerekirse, P_{ij} görüntüyü, I_j ise görüntüdeki satır ve sütunları temsil etmek üzere; bölütleme, P_{ij} görüntüsünü c bölüme ayıran ve her R_i (c bölümden i 'nci bölüm) bölümünün p uzayında homojen özelliklere sahip olduğu bir işlemdir. Bu gösterim Denklem (2.15)'te verilmiştir (Fu, 1982).

Gri seviyeli görüntülerde uygulanan bölütleme algoritmaları, genelde gri seviyelerindeki benzerlik ve değişim özellikleri temeline dayanmaktadır. Değişim özelliğini kullanan algoritmalar gri seviyesindeki ani değişimlere (örneğin görüntüdeki kenarlar) göre görüntüyü bölümlere ayırmaktadır. Benzerlik özelliğini kullanan algoritmalarda ise önceden belirlenen kıstaslara benzerlik durumuna göre görüntü bölümlere ayrılmaktadır.

$$\bigcup_{i=1}^c R_i = P_{I,J} \quad (2.15)$$

$$R_i \cap R_j = \emptyset \quad \forall i, j, i \neq j$$

$R_i, 1, \dots, c$ komşu bölgeler

$P(R_i) = \text{doğru} \quad \forall i, R_i$ bölgesi için ortak bir özellik bulunmaktadır.

$P(R_i \cup R_j) = \text{yanlış}$ eğer $i \neq j$, farklı bölgelerde ortak özellik bulunmamaktadır.

Görüntüdeki ani değişimlere göre çalışan algoritmalarla özellikle kenar bulma algoritmaları bölütleme algoritmaları arasında önemli bir yere sahiptir.

Eşik değerine göre bölütleme yapan algoritmalar özellikle hızın önemli olduğu durumlarda oldukça popülerdir. (Gonzalez ve Woods, 2002)

2.4.1. Nokta, çizgi ve kenar bulma

Görüntüdeki ani değişimler bulunarak nokta, çizgi veya kenarlar tespit edilebilmektedir. Görüntüdeki ani değişimleri bulmanın en temel yolu görüntü ile Şekil 2.5'teki gibi bir maskeyi konvolüsyon işlemi işlemine sokmaktır (Gonzalez ve Woods, 2002).

| | | |
|-------|-------|-------|
| w_1 | w_2 | w_3 |
| w_4 | w_5 | w_6 |
| w_7 | w_8 | w_9 |

Şekil 2. 5: 3x3'lük maske

Görüntünün bu maske ile konvolüsyon işleminde sonuç değeri Denklem (2.16) ile hesaplanmaktadır.

$$R = w_1z_1 + w_2z_2 + \dots + w_9z_9 \quad (2.16)$$

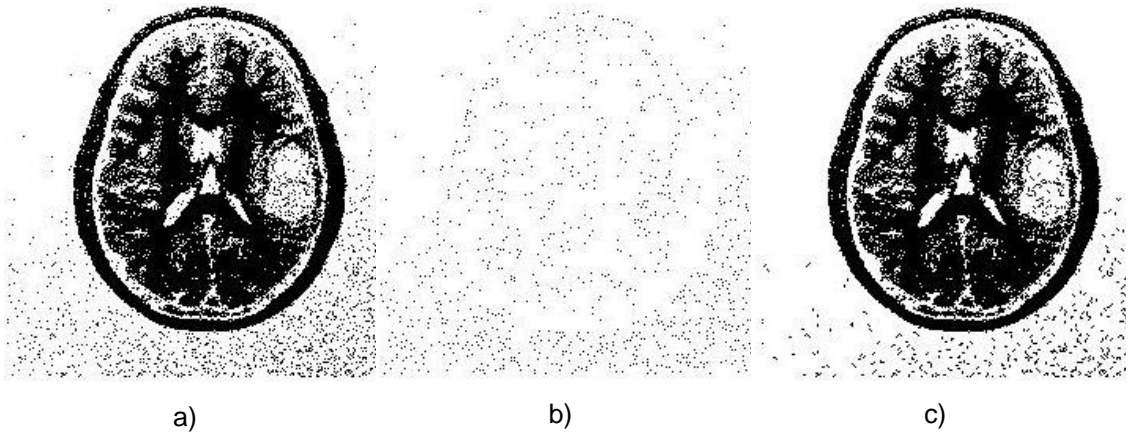
Burada z_i , w_i konumunda bulunan pikselin gri seviyesini ifade etmektedir. Elde edilen R değeri pikselin yeni değerini oluşturmaktadır.

2.4.1.1. Nokta bulma

| | | |
|----|----|----|
| -1 | -1 | -1 |
| -1 | 8 | -1 |
| -1 | 1 | -1 |

Şekil 2. 6: Noktaları bulan 3x3'lük maske

Görüntünün Gonzalez ve diğ., (2004) tarafından sunulan Şekil 2.6'daki maske ile konvolüsyon işleminden elde edilen değer, tespit edilen eşik değerinden ($|R| \geq T$) büyükse maskenin merkezine denk gelen pikselin bir nokta olduğu belirlenmektedir. Maskeden elde edilen değer sıfıra yakınsa düzensizliğin az olduğu, sıfırdan çok büyükse düzensizliğin fazla olduğu anlamı çıkmaktadır. Şekil 2.7a'daki görüntüye maske uygulandıktan sonra 2.8b'deki nokta olarak belirlenen pikseller silinmiştir. Şekil 2.7c'den görüldüğü gibi küçük noktalar yok olmuş ancak daha büyük boyutlu parçacıklar korunmuştur.



Şekil 2. 7: Nokta bulan maskenin uygulanması. a) Orijinal görüntü (gürültülü). b) Tespit edilen noktalar. c) Orijinal görüntüden noktalar çıkarıldıktan sonra elde edilen görüntü.

2.4.1.2. Çizgi bulma

Şekil 2.8'deki maskeler herhangi bir çizginin içinde bulunan piksele geldiğinde en yüksek değeri verirler (Gonzalez ve diğ., 2004). Görüntüye bu maskeler uygulandıktan sonra elde edilen görüntü matrisinde yüksek değerli pikseller çizgileri ifade etmektedir. Şekil 2.9'da yatay ve dikey çizgileri tespit eden maskenin görüntüye uygulanmasıyla elde edilen sonuçlar gösterilmektedir.

| | | |
|----|----|----|
| -1 | -1 | -1 |
| 2 | 2 | 2 |
| -1 | -1 | -1 |

a) Yatay çizgi

| | | |
|----|----|----|
| -1 | -1 | 2 |
| -1 | 2 | -1 |
| 2 | -1 | -1 |

b) Sağa yatık çizgi

| | | |
|----|---|----|
| -1 | 2 | 1 |
| -1 | 2 | -1 |
| -1 | 2 | -1 |

c) Dikey çizgi

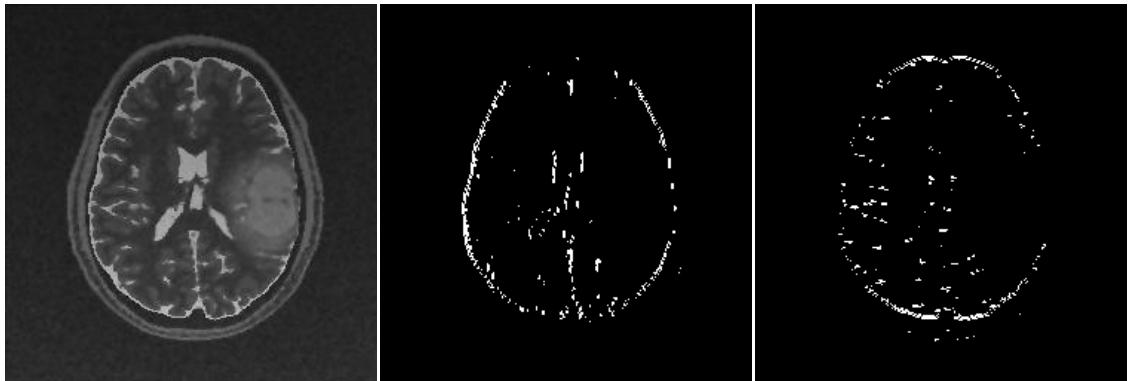
| | | |
|----|----|----|
| 2 | -1 | -1 |
| -1 | 2 | -1 |
| -1 | -1 | 2 |

d) Sola yatık çizgi

Şekil 2. 8: Çizgileri bulan 3x3'lük maske

2.4.1.3. Kenar bulma

Görüntüyü bölümlere ayırırken nokta ve çizgi bulma önemlidir ancak kenar bulma yöntemleri, gri seviyesindeki ani değişimleri bulmak için en genel yöntemlerdir. Görüntüde kenarları belirlemek için birçok yöntem bulunmaktadır. Bu yöntemler Gonzalez ve diğ. (2004) ile Gonzalez ve Woods, (2002)'un kitaplarından faydalanılarak kısaca açıklanmaya çalışılacaktır.



a) Orijinal görüntü.

b) Dikey çizgiler.

c) Yatay çizgiler.

Şekil 2. 9: Çizgi bulan maskenin uygulanması

Görüntüde bulunan kenarları daha iyi ifade edebilmek için Gonzalez ve Woods, (2002) tarafından kullanılan gösterim Şekil 2.10'da sunulmaktadır. Şeklin birinci bölümünde kenar için ideal olan durum sunulmaktadır ki bu durum gerçekte pek mümkün olmamaktadır. İkinci bölümde ise kenar için gerçek bir yaklaşım sunulmaktadır. Görüntülerde kenar bölgelerde genelde bu şekilde bir geçiş bulunmaktadır. Eğim (gradient) hesaplayarak kenar bulan yöntemlerde şeklin ikinci bölümünde gösterilen kenar bölgenin eğimi bulunmaktadır. Bu yöntemlerde eğimin yönünü de bulmak mümkündür. Eğim hesaplamak için geliştirilen maskeler 0 (sıfır) sonucunu ürettiğinde pikselin kenar piksel olmadığı, maskenin ürettiği değer arttıkça ise pikselin kuvvetli bir ihtimalle kenar pikseli olduğu anlamı çıkarılmaktadır. Bu aşamada hesaplanan eğim değeri belirlenen bir t eşik değeri ile karşılaştırılmakta ve eğim değeri t eşik değerinden büyükse piksel kenar piksel olarak kabul edilmektedir.

Şekil 2.11'de ise dikey bir çizgiyle iki bölgeye ayrılan görüntünün gri seviyesi değişimi ve bu değişim noktalarındaki birinci ve ikinci türevler gösterilmektedir (Gonzalez ve Woods, 2002).

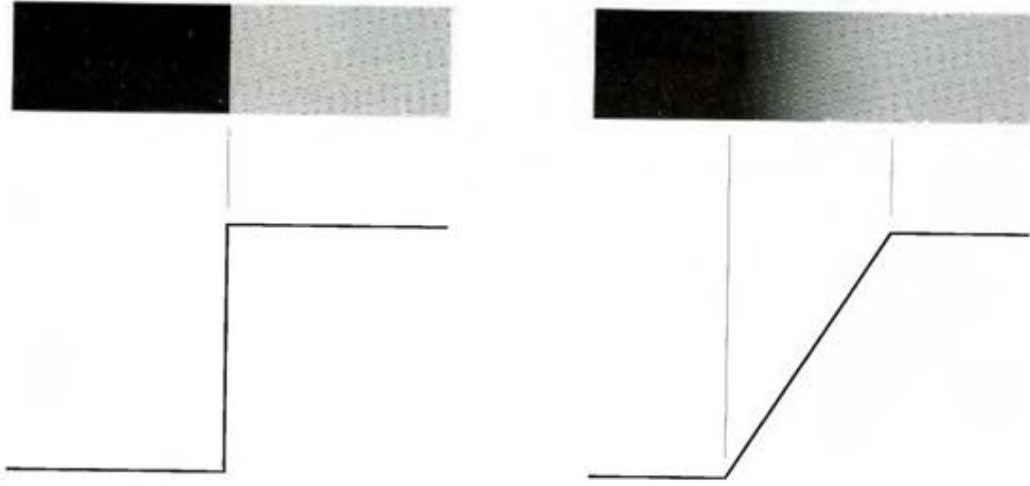
Görüntüde herhangi bir (x,y) konumundaki eğim (gradient) f(x,y) Denklem (2.17) ile hesaplanmaktadır. Eğimin büyüklüğü ise Denklem (2.18) ile hesaplanmaktadır. Eğimin büyüklüğünün yanında yönü de önem arz etmektedir. $\alpha(x,y)$ eğimin yönünü göstermek üzere; eğimin yönü, Denklem (2.19) ile hesaplanmaktadır (Gonzalez ve Woods, 2002) :

$$\nabla f = \begin{bmatrix} G_x \\ G_y \end{bmatrix} = \begin{bmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial y} \end{bmatrix} \quad (2.17)$$

$$\left[G_x^2 + G_y^2 \right]^{1/2} \quad (2.18)$$

$$\alpha(x,y) = \tan^{-1} \left(\frac{G_y}{G_x} \right) \quad (2.19)$$

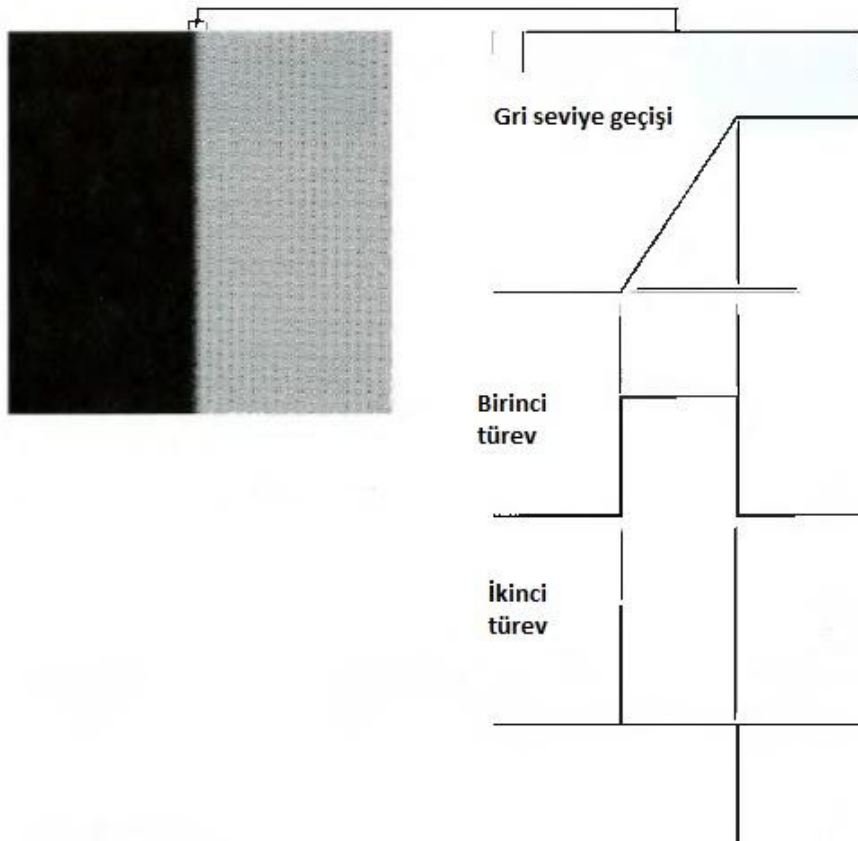
Birinci türevi kullanarak kenar bulan yöntemler olan Sobel, Prewit ve Roberts yöntemleri sırasıyla sunulacaktır.



a) Kenar için ideal model

b) Yumuşak geçişli kenar modeli

Şekil 2. 10: Kenar modeli (Gonzalez ve diğ., 2004)



Şekil 2. 11: Gri seviyesi değişiminin türevleri (Gonzalez ve diğ., 2004)

2.4.1.3.1. Sobel kenar bulma yöntemi

Bu yöntemde Şekil 2.12'deki maskeler kullanılmaktadır (Gonzalez ve Woods, 2002).

Merkez pikseldeki eğim değeri Denklem (2.20) ile hesaplanmaktadır. Denklem açılırsa Denklem (2.21) elde edilmektedir. Denklem ile elde edilen g değeri belirli bir t eşik değerinden büyük veya bu değere eşit olursa maskenin üzerinde bulunduğu piksel kenar piksel olarak kabul edilmektedir.

| | | |
|----------------|----------------|----------------|
| Z ₁ | Z ₂ | Z ₃ |
| Z ₄ | Z ₅ | Z ₆ |
| Z ₇ | Z ₈ | Z ₉ |

Görüntü komşulukları

| | | |
|----|----|----|
| -1 | -2 | -1 |
| 0 | 0 | 0 |
| 1 | 2 | 1 |

$$G_x = \frac{(Z_7 + 2Z_8 + Z_9) - (Z_1 + 2Z_2 + Z_3)}{(Z_1 + 2Z_2 + Z_3)}$$

| | | |
|----|---|---|
| -1 | 0 | 1 |
| - | 0 | 2 |
| -1 | 0 | 1 |

$$G_y = \frac{(Z_3 + 2Z_6 + Z_9) - (Z_1 + 2Z_4 + Z_7)}{(Z_1 + 2Z_4 + Z_7)}$$

Şekil 2. 12: Sobel kenar bulma maskesi

$$g = \left[G_x^2 + G_y^2 \right]^{1/2} \quad (2.20)$$

$$g = \left\{ \left[(Z_7 + 2Z_8 + Z_9) - (Z_1 + 2Z_2 + Z_3) \right]^2 + \left[(Z_3 + 2Z_6 + Z_9) - (Z_1 + 2Z_4 + Z_7) \right]^2 \right\}^{1/2} \quad (2.21)$$

2.4.1.3.2. Prewit kenar bulma yöntemi

Bu yöntemde Şekil 2.13'teki maske kullanılmaktadır. Merkez pikseldeki eğim değeri Sobel'de olduğu gibi Denklem (2.20) ile hesaplanmaktadır. Bu denklem Prewit için açıldığında Denklem (2.22) elde edilmektedir (Gonzalez ve Woods, 2002).

$$g = \left\{ \left[(Z_7 + Z_8 + Z_9) - (Z_1 + Z_2 + Z_3) \right]^2 + \left[(Z_3 + Z_6 + Z_9) - (Z_1 + Z_4 + Z_7) \right]^2 \right\}^{1/2} \quad (2.22)$$

Hesaplanması Sobel'e göre daha basittir. Ancak gürültüye daha hassastır. Sobel maskesinde bulunan iki (2) katsayısı maskenin gürültüden etkilenme oranını düşürmektedir.

| | | |
|----------------|----------------|----------------|
| Z ₁ | Z ₂ | Z ₃ |
| Z ₄ | Z ₅ | Z ₆ |
| Z ₇ | Z ₈ | Z ₉ |

Görüntü komşulukları

| | | |
|----|----|----|
| -1 | -1 | -1 |
| 0 | 0 | 0 |
| 1 | 1 | 1 |

$$G_x = (Z_7 + Z_8 + Z_9) - (Z_1 + Z_2 + Z_3)$$

| | | |
|----|---|---|
| -1 | 0 | 1 |
| -1 | 0 | 1 |
| -1 | 0 | 1 |

$$G_y = (Z_3 + Z_6 + Z_9) - (Z_1 + Z_4 + Z_7)$$

Şekil 2. 13: Prewit kenar bulma maskesi

2.4.1.3.3. Roberts kenar bulma yöntemi

En eski kenar bulma yöntemlerinden biridir. Sobel ve Roberts'a göre daha az kullanılmaktadır. Basitlik ve hız önemliyse kullanılabilir. Hesaplanması Şekil 2.14'teki maskeler kullanılarak yapılmaktadır (Gonzalez ve Woods, 2002).

Sobel, Prewit ve Roberts kenar bulma yöntemlerinin beyin MR görüntüsüne uygulanması sonucu elde edilen görüntüler Şekil 2.15'te sunulmaktadır. Burada kullanılan görüntüde gürültü bulunmadığı için Sobel ve Prewit yöntemlerinin çok yakın sonuçlar verdiği, Roberts kenar bulma yönteminin sonucunun ise diğer iki yöntemle göre daha zayıf olduğu görülmektedir.

| | |
|----|---|
| -1 | 0 |
| 0 | 1 |

$$G_x = Z_9 - Z_5$$

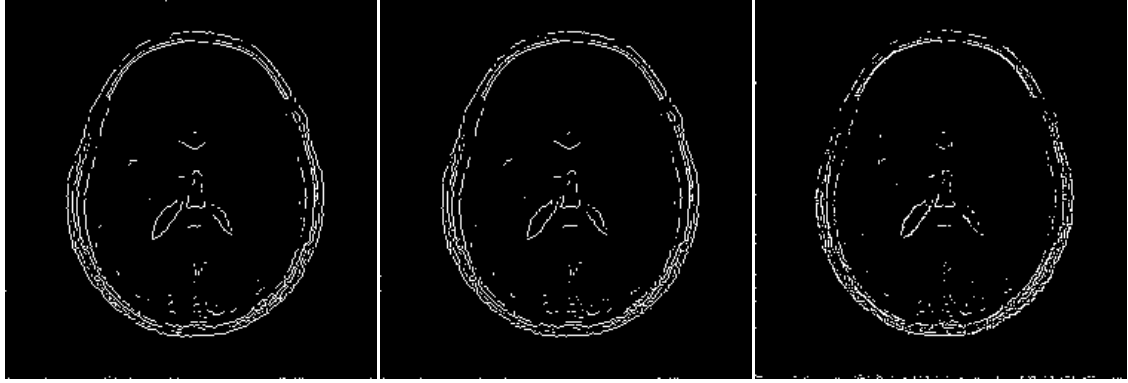
| | |
|---|----|
| 0 | -1 |
| 1 | 0 |

$$G_y = Z_8 - Z_6$$

Şekil 2. 14: Roberts kenar bulma maskesi

2.4.1.3.4. LoG (Laplacian of Gaussian) kenar bulma yöntemi

Bu yöntem de türev tabanlı bir yöntemdir. Bu yöntemlerde en çok kullanılan maskeler Şekil 2.16'da sunulmaktadır (Gonzalez ve Woods, 2002).



a)

b)

c)

Şekil 2. 15: Sobel, Prewit ve Roberts kenar bulma yöntemlerinin karşılaştırılması. a) Sobel kenar bulma yöntemi sonucu. b) Prewit kenar bulma yöntemi sonucu. c) Roberts kenar bulma yöntemi sonucu

| | | |
|----|----|----|
| 0 | -1 | 0 |
| -1 | 4 | -1 |
| 0 | -1 | 0 |

Laplacian

| | | |
|----|----|----|
| -1 | -1 | -1 |
| -1 | 8 | -1 |
| -1 | -1 | -1 |

Laplacian

| | | |
|----|----|----|
| 0 | -1 | 0 |
| -1 | 5 | -1 |
| 0 | -1 | 0 |

Laplacian of Gaussian (LoG)

Şekil 2. 16: LoG kenar bulma maskeleri

2.4.1.3.5. Canny kenar bulma yöntemi

Canny kenar bulma yöntemi türev tabanlı bir yöntem olmakla birlikte, anlatılan diğer yöntemlere göre daha karmaşık bir yapıya sahiptir. Şekil 2.17'de Canny ve Log kenar bulma yöntemlerinin beyin MR görüntüsüne uygulanması ile elde edilen görüntüler sunulmaktadır. Canny kenar bulma yönteminin diğer yöntemlere göre daha iyi sonuç verdiği görülmektedir.

Canny kenar bulma yöntemi dört adımda açıklanabilir (Canny, 1986):

Birinci adım gürültünün azaltılması amacıyla Gaussian filtresi kullanılarak görüntünün bulanıklaştırılmasıdır.

İkinci adımda her nokta için Denklem (2.18) ile ifade edilen lokal eğim ve Denklem (2.19) ile belirtilen eğimin yönü hesaplanmaktadır. G_x ve G_y hesaplanırken Sobel, Prewitt veya Roberts yöntemlerinden birisi kullanılabilir. Eğimle aynı yönde lokal maksimum değere sahip olan nokta kenar nokta olarak belirlenmektedir.



Şekil 2. 17: Canny ve LoG kenar bulma yöntemlerinin karşılaştırılması. a) Canny kenar bulma yöntemi sonucu. b) LoG kenar bulma yöntemi sonucu.

İkinci adımda bulunan kenar noktalar eğim büyüklüğünü gösteren görüntüde tepe noktasına doğru artış göstermektedir. Algoritma bu tepe noktalarının haricindeki noktaları sıfıra eşitler ve çıkışta kalın bir çizgi sonucunu verir. Bu işleme maksimum olmayan noktaların bastırılması da diyebiliriz. Daha sonra bu tepe noktalar T_1 ve T_2 ($T_1 < T_2$) eşik değerleriyle bölütlenir. T_2 'den büyük olan noktalar güçlü kenar noktası, T_1 'den küçük olan noktalar ise zayıf kenar noktasıdır.

Son olarak algoritma zayıf kenar noktaları da dahil ederek güçlü kenar noktalara doğru sekiz komşuluklu (8-connected) kenar birleştirme yaparak bütünlüğü sağlamaya çalışmaktadır (Gonzalez ve diğ., 2004).

2.4.2. Eşik değeri ile bölütleme (thresholding)

Görüntünün, eşik değeri kullanılarak bölümlere ayrılması tek boyutlu görüntüler için (gray level) en basit bölütleme yöntemlerinden biridir. Eşik değeri görüntüye uygulandıktan sonra ikili (binary) bir görüntü elde edilmektedir. Bu görüntüde "1" değeri objeleri, "0" değeri ise arka planda bulunan değerleri ifade etmektedir. X_{ij} , i, j

konumundaki piksel deęerini ifade etmekte olup, ikili grnt Denklem (2.23) ile elde edilmektedir (Bezdek ve dię., 2005).

$$I_T(X_{ij}) = \begin{cases} 1 & ; I(X_{ij}) \geq t \\ 0 & ; I(X_{ij}) < t \end{cases} \quad (2.23)$$

Doęru eřik deęerini bulmak iin birok klasik yntem bulunmaktadır. Ortalama deęer alınması, en byk deęer ile en kk deęer ortalaması, histograma bakılması ve benzeri birok yntem sayılabilmektedir.

Őekil 2.19'da histogramı verilen Őekil 2.18a'daki orijinal beyin MR grntsnn, Őekil 2.18b'de k-ortalama kmeleme algoritmasıyla bulunan 0.2326 eřik deęeri ile oluřturulan ikili (binary) grnts, 2.18c'de ise histograma bakılarak bulunan 0.25 eřik deęeri ile oluřturulan ikili grnt sunulmaktadır.

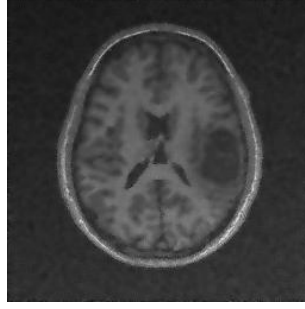
Grntdeki piksellerin gri seviyesi, pikselin obje iindeki yelik derecesiyle baęlantılıdır. Orneęin obje arka plana gre daha aık tonlarda bulunuyorsa o zaman pikselin gri seviyesi ne kadar yksekse yelik deęeri de o kadar yksek olmaktadır.

2.4.3. Alan tabanlı bltleme (Region-based segmentation)

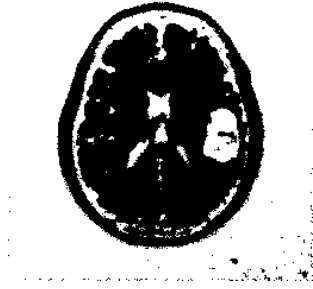
2.4.3.1. Alan bytme (region growing)

Bu yntemde ncelikle ekirdek bir piksel grubu belirlenmektedir. Daha sonra gri seviyesi, renk bilgisi ve benzeri zelliklere gre ekirdek pikselin komřuluęunda bulunan pikseller eklenerek alan (blęe) geniřletilmektedir (Gonzalez ve dię., 2004).

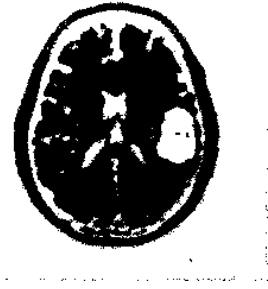
Algorri ve dię., (2004), bltleme algoritmasında alan bytme metodunu kullanmıřlardır. Uyguladıkları yntemde nce bir ekirdek piksel seilmekte ve bu piksel ile aynı alanda olan pikseller belirlenmektedir. Alana dahil edilecek pikseller belirlenirken; pikselin, 8 komřuluęu ile olan gri seviyesi benzerlięine ve grntnn ortalama gri seviyesi ile olan gri seviyesi benzerlięine bakılmaktadır.



a) Orijinal görüntü

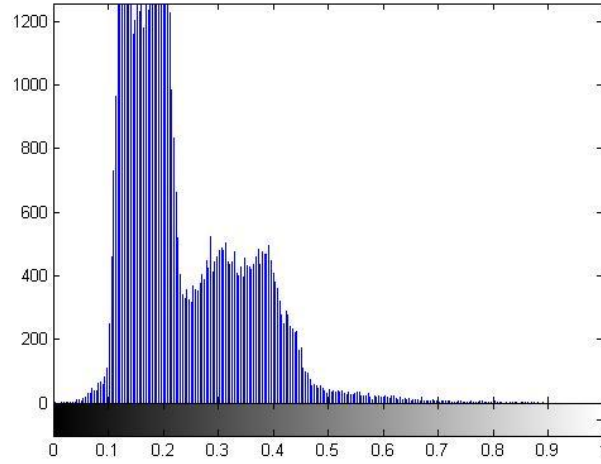


b) k-means algoritmasıyla bulunan 0.2326 eşik değeri uygulanmıştır.



c) Histogramdan bulunan 0.25 eşik değeri uygulanmıştır.

Şekil 2. 18: Eşik değeri ile bölütleme



Şekil 2. 19: Orijinal görüntünün histogramı

Her iki durum için de gri seviyesi farkları belirli bir eşik seviyesinden küçük olan komşu pikseller çekirdek pikselin alanına dahil edilmektedir.

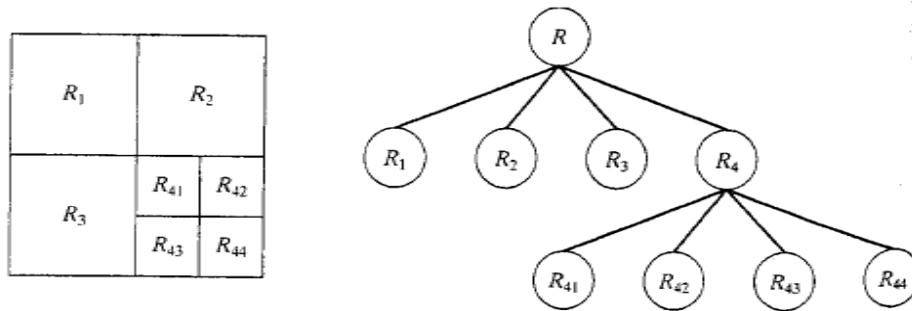
2.4.3.2. Alan bölme ve birleştirme

Bu yöntemde görüntü başlangıçta rastgele parçalara ayrılmakta ve daha sonra bu parçalar belirli bir kurala göre birleştirilmektedir.

R tüm görüntüyü ifade eden bir simge ve P ise ayrılan parçaların kendi içinde homojen olup olmadığını, daha fazla bölünmeye gerek olup olmadığını, belirleyen bir çıkarım yöntemi olmak üzere; görüntüde ilk bölümler oluşturulduktan sonra her bölüm için $P(R_i)$ hesaplanmaktadır. $P(R_i)=\text{doğru}$ ise artık o bölüm tekrar bölünmemekte, $P(R_i)=\text{yanlış}$ ise R_i parçası tekrar bölünmektedir. Görüntünün R_i bölgelere bölünmesi ve bölünme işleminin ağaç yapısında gösterimi Şekil 2.20'de şematik olarak sunulmaktadır (Gonzalez ve diğ., 2004).

Bölünme işlemi tüm R_i bölgeleri kendi içinde homojen oluncaya kadar, yani $P(R_i)=\text{doğru} \forall i$ oluncaya kadar, devam etmektedir.

Bölünme işleminden sonra birleştirme işlemi başlamaktadır. Birleştirme işleminde $P(R_i \cup R_j)=\text{doğru} \ i \neq j$ ise i ve j bölgeleri birleştirilmektedir. Bu aşamada farklı birleştirme metotları uygulanabilmektedir. Örneğin P , tüm bölgeler için karşılıklı kontrol edilerek birleşim işlemi gerçekleştirilebileceği gibi, sadece ağaç yapısında aynı düğüme bağlı bölgeler de birleştirilebilmektedir (Gonzalez ve diğ., 2004).



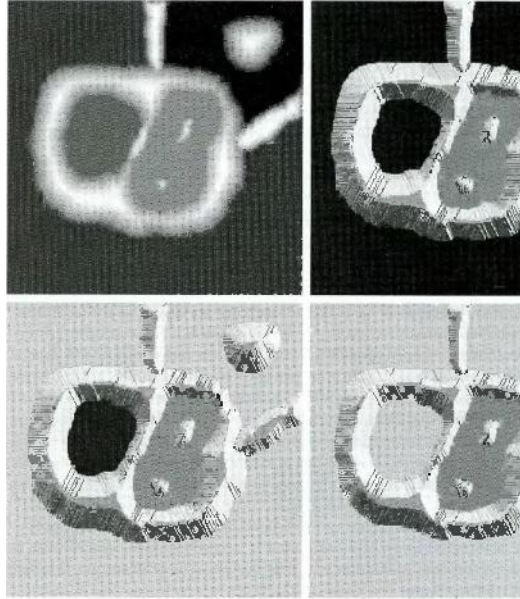
Şekil 2. 20: Alan bölme işlemi (Gonzalez ve diğ., 2004)

2.4.4. Watershed dönüşümü ile bölütleme

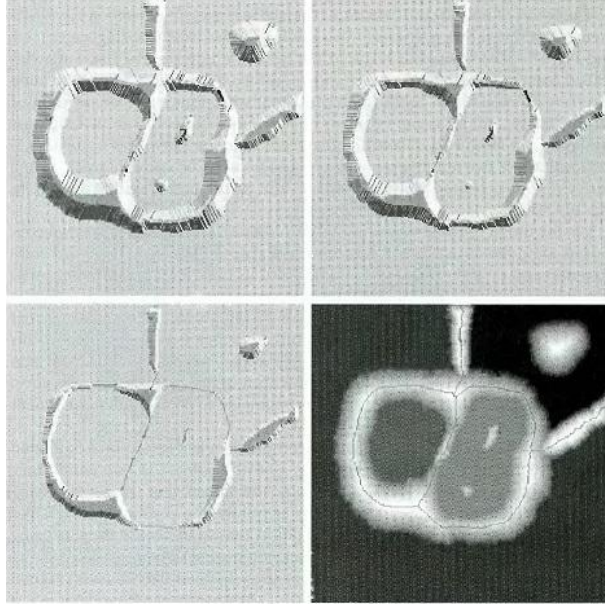
Coğrafyada watershed (iki havza arasındaki set), akarsularla beslenen bölge(havza)leri bölen sırtlara denilmektedir. Havza (catchments basin) ise dağ ve tepelerle çevrili, suları aynı yere akan bölge anlamına gelmektedir. Su doldukça havzalar suyun altında kalacak ve sırt bölgeler ise suyun üstünde bulunacaktır. Belirli bir seviyede su doldurulduktan sonra artık dipteki detaylar kaybolacak, sadece yüzeydeki sırtlardan oluşan bir şekil kalacaktır (Gonzalez ve diğ., 2004).

Aynı mantık görüntü üzerine uygulandığında objeleri gösteren pikseller su yüzeyinde kalırmışçasına belirginleşmekte, diğer pikseller soyutlanmaktadır.

Şekil 2.21 ile sunulan görüntüde sol üstte orijinal görüntü, diğer üç bölümde ise orijinal görüntünün üç boyutlu hali görülmektedir. Aynı kitapta bulunan ve Şekil 2.22 ile sunulan ikinci görüntüde ise watershed algoritmasının uygulanmış hali görülmekte olup, görüntünün sağ alt köşesi görüntünün son halini göstermektedir.



Şekil 2. 21: Görüntünün üç boyutlu gösterimi (Gonzalez ve diğ., 2004)



Şekil 2. 22: Görüntüye Watershed algoritması uygulanması (Gonzalez ve diğ., 2004)

3. BULANIK KÜMELEME

Bulanık yöntemlerin klasik yöntemlere göre avantajlarına önceki bölümde değinilmiştir. Klasik yöntemlerde giriş verisi algoritma ile belirlenen kurallara göre bir kümeye ya aittir ya da değildir. Bu sonuç algoritma tarafından üretilerek sunulmaktadır. Verinin kümeye ne oranda ait olduğu bilgisi bilinmemektedir. Farklı veri kümelerine aynı uzaklıkta olan bir veri için durum ne olacaktır? Klasik yöntemde mecburen bir küme seçilecektir. Bu durumun getirdiği bir çok dezavantaj bulunmaktadır. Bu nedenle daha esnek bir yapı sunan bulanık kümeleme yöntemleri geliştirilmiştir. Bu yöntemlerde hangi kümeye ait olunacağından çok her kümeye ne kadar ait olunabileceği hesaplanmaktadır. Sunulan bu değerler doğrultusunda son karar aşamasında sağlıklı bir karar verilmesi mümkün hale getirilmektedir.

İkinci bölümde klasik yöntemlerine değinilen kümeleme analiziyle ilgili insan düşünce sistemine yakın bir yaklaşım sunmak için bulanık yöntemler irdelenmiştir. Bu aşamada tespit edilen en önemli husus; kümeleme yöntemlerinin birbirlerine tam bir üstünlük sağlayamaması buna karşılık performansı esas etkileyen öğenin kümeleme algoritmaları için gerekli parametrelerin kestirilmesindeki başarımlıdır. Hangi kümeleme algoritmasını kullandığınızdan çok hangi parametreleri kullandığınız ve bu parametreler için bulduğunuz değerler burada önem kazanmaktadır.

Bulanık c-ortalama algoritması (fuzzy c-means algorithm), Gustafson-Kessel algoritması ve Gath-Geva algoritması bulanık kümeleme algoritmalarına örnek olarak verilebilir.

Bulanık kümeleme yaklaşımıyla ilgili tanımlamalara ikinci bölümde değinilmiştir. Bu tanımlamalara göre Denklem (2.2) bulanık etiket vektörünü ifade etmektedir. Bulanık etiketler de yine "0" ile "1" arasındaki sayılardan oluşmaktadır ancak bir obje için verilen etiket değerlerinin toplamı 1'e eşittir. Bulanık kümeleme yöntemleri ile diğer yöntemler arasındaki ilişkiyi irdelemek için ikinci bölümde verilen örneğin incelenmesinde fayda bulunmaktadır.

3.1. Bulanık c-ortalama kümeleme modeli

Klasik modelden farklı olarak üyelik matrisi (U) ve örnek nokta (V) hesaplanması aşağıdaki şekilde değiştirilmektedir.

$$U_{ik} = \left[\sum_{j=1}^c \left(\frac{D_{ik}}{D_{jk}} \right)^{\frac{2}{m-1}} \right]^{-1} \quad \forall i, k; \quad (3.1)$$

$$v_i = \left(\sum_{k=1}^n u_{ik}^m x_k / \sum_{k=1}^n u_{ik}^m \right) \quad \forall i \quad (3.2)$$

Denklem (3.1) ile (3.2) klasik modellerle aynı işlemleri bulanık yaklaşımla göstermektedir. Burada "1" ve "0" kullanmak yerine her objeye küme merkezlerine olan uzaklıklarına göre [0-1] aralığında değerler verilir ve küme merkezleri yeniden hesaplanırken de bu değerler kullanılır.

Bulanık c-ortalama algoritması Tablo 3.1'de verilmiştir. Klasik k-ortalama algoritmasından farklı üyelik matrisi ve örnek nokta hesaplanmasında bulanıklık katsayısının (m) devreye girmesidir. Üyelik matrisi oluşturulurken noktanın küme merkezine uzaklığıyla orantılı ve bulanıklık katsayısı ile bulanıklık derecesi değişen üyelik dereceleri hesaplanacaktır. Bir nokta (piksel) için üyelik dereceleri toplamı 1(bir)'e eşit olacaktır.

Algoritmanın adımlarını açıklamak gerekirse, öncelikle kümelenecek veri algoritmaya kaydedilir. Veride her obje (piksel) için kaç adet özellik mevcutsa p sayısı bu adedi ifade eder. Daha sonra algoritma tarafından kullanılacak olan birtakım değerler belirlenir.

Bu değerleri açıklamak gerekirse, öncelikle algoritmanın bulacağı küme sayısı belirtilmelidir. Bu sayı 1'den büyük ve obje sayısı (piksel sayısı) olan n'den küçük olmalıdır. İkinci olarak, algoritmanın sonlanabilmesi için yeni bulunan küme merkezleri ile bir öncekiler arasındaki farkın belirli bir değerden (ϵ) küçük olması gerekmektedir veya bu değere ulaşılmasa da belirli bir sayıda adımdan (T) sonra algoritma son bulmalıdır. Bahsedilen ϵ ve T değerleri belirlenmelidir. Son olarak

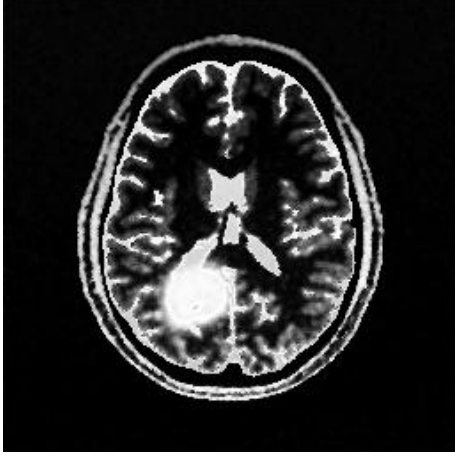
bulanıklık katsayısı (m) de belirlenmelidir. Klasik yaklaşımda $m=1$ olarak kabul edilmektedir.

Tablo 3. 1: Bulanık c-ortalama algoritması

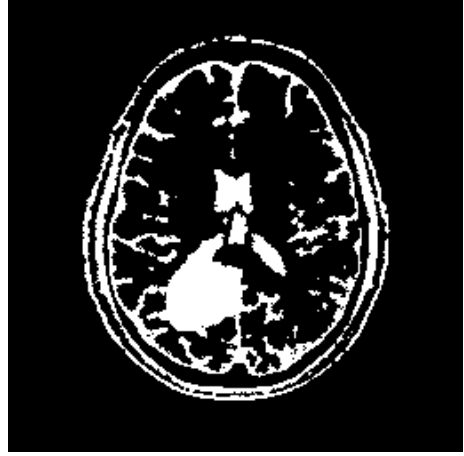
| | |
|--------|--|
| Adım 1 | Etiketlenmemiş veri girişi $X \subset \mathbb{R}^p$ |
| Adım 2 | <p>Küme sayısı (c) belirlenir: $1 < c < n$</p> <p>Maksimum iterasyon sayısı belirlenir : T</p> <p>Bulanıklık katsayısı m belirlenir: $1 \leq m < \infty$</p> <p>J_m için uzaklık ölçüm normu belirlenir: $\ x\ _A^2 = x^T A x$</p> <p>Sonlandırma ölçümü: $E_t = \ V_t - V_{t-1}\$</p> <p>Sonlandırma için eşik değeri belirlenir: $0 < \varepsilon$</p> |
| Adım 3 | Başlangıç noktaları girilir: $V_0 = (v_{1,0}, \dots, v_{c,0}) \in \mathbb{R}^p$ |
| Adım 4 | <p>$t = 0$</p> <p>Repeat</p> <p>U_t, Denklem (2.20) ile önceki adımdaki örnek noktalar (V_{t-1}) kullanılarak hesaplanır.</p> <p>V_t, U_t kullanılarak Denklem (2.21) ile yeniden hesaplanır.</p> <p>Until ($t=T$ veya $E_t \leq \varepsilon$)</p> |
| Adım 5 | $U, V = U_t, V_t$ |

Algoritmanın başında belirlenen ε ve T değerleri kullanılarak; küme merkezleri ile bir önceki adımda hesaplanan küme merkezleri arasındaki fark ε değerinden küçük veya bu değere eşit olacaktır kistası sağlanıncaya kadar veya bu kistas sağlanamazsa belirlenen T adım sayısına ulaşıncaya kadar algoritma devam edecektir.

Şekil 3.1'de klasik ve bulanık yöntemlerle iki kümeye ayrılan beyin MR görüntüsü gösterilmektedir. İki görüntü de incelendiğinde klasik yöntemle yapılan kümelemede sınırların kesin hatlarla çizildiği ancak bulanık yöntemle kümelenen görüntüde geçiş hatlarının belirgin olduğu görülmektedir. Klasik görüntüde görülmeyen birçok ayrıntı bulanık görüntüde bulunmaktadır. Görüntülere bakıldığında bulanık görüntünün bize çok daha fazla detay sunduğu ve gerçeğe çok daha yakın olduğu aşikardır. Gözle bile rahat bir şekilde görülebilen bu detay bulanık sistemlerin ne kadar efektif olduğunu göstermektedir.



a) Bulanık c-ortalama



b) Klasik k-ortalama

Şekil 3. 1: Klasik ve bulanık yöntemle kümeleme. a) Bulanık yöntemle iki kümeye ayrılmış MR görüntüsü. b) Klasik yöntemle iki kümeye ayrılmış MR görüntüsü.

4. BULANIK SINIFLANDIRMA

İkinci bölümde sınıflandırıcı $D: R^p \rightarrow N_{pc}$ şeklinde bir fonksiyon olarak tanımlanmıştır. $y = D(z)$ değeri R^p uzayında tanımlı z için etiket vektörüdür.

Sınıflandırma için oluşturulan algoritmaların son aşamaya kadar klasik karar verme (0 veya 1) yerine bulanık bir karar mekanizmasıyla işlemesi önem arz etmektedir. Klasik yapıya dönüştürülen kararların tekrar bulanık yapıya dönüştürülmesi ise çok zordur, hatta imkânsızdır. Algoritma klasik yapıda ilerlediğinde detay bilgilerin işleme sokulması mümkün olmayacaktır. Özellikle otomatik hedef tanıma sistemleri, bilgisayar destekli medikal analiz gibi karmaşık yapılarda bu detay bilgiler büyük önem taşımaktadır. Bunun nedeni bu gibi sistemlerde karar sürecini etkileyen birçok faktör vardır. Bu faktörlere örnek olarak görüntüdeki gürültü piksellerinin ön işlem için sınıflandırılması, gürültü arındırıldıktan sonra görüntüdeki objelerin belirlenmesi (segmentasyon), özellik çıkarımı ve objelerin sınıflandırılması ve son olarak tüm görüntünün elde edilen sonuçlara göre etiketlenmesi verilebilir.

Wang ve diğ. (2009), Landsat TM uydusundan alınan görüntünün 5(beş) kategoriye (nehir, göl, pirinç alanı, yeşillik ve yerleşim) ayrılmasında bulanık yöntemleri kullanmıştır. Uygulama sonucunda klasik yöntemlere göre daha iyi sonuçlar elde edilmiştir.

Yine ikinci bölümde klasik olmayan sınıflandırıcıların elde ettikleri değerleri $H: N_{fc} \rightarrow N_{hc}$ fonksiyonuyla klasik hale getirdikleri belirtilmiştir. $H \circ D(y) = H(D(y)) \in \{e_1, \dots, e_c\}$.

Sınıflandırıcı tasarımı iyi bir D fonksiyonu bulmak anlamına gelmektedir. D fonksiyonu oluşturulurken etiketlenmiş veri kullanılırsa bu işleme eğitilmiş öğrenme (supervised learning) adı verilmektedir.

D bir fonksiyon olarak tanımlanabileceği gibi (örneğin Bayes sınıflandırıcı) bir bilgisayar programı (örneğin computational neural network veya fuzzy input-output system) olarak da tanımlanabilir. Her iki sınıflandırıcı da parametrelere sahiptir. D fonksiyon olarak tanımlandığında bazı sabit değerlerin eğitim süresince öğrenilmesi

gerekmektedir. D bilgisayar programı olarak tanımlanmışsa hem kontrol parametrelerinin hem de sabitlerin eğitim süresince öğrenilmesi gerekmektedir. Her iki durumda da öğrenme kelimesi D için iyi parametrelerin bulunması anlamına gelmektedir.

Eğitilmiş sınıflandırma tasarımında X verisi genellikle klasik yöntemle etiketlenmiş eğitim verisinden X_{tr} ve eğitim sonucunda elde edilen parametrelerin sınıflandırıcı ile test edilmesi için test verisinden X_{te} oluşmaktadır. Sınıflandırıcının test edilmesi hata oranının bulunmasıdır. Bunun için standart metod sınıflandırıcının X_{te} verisine uygulanması ve yanlış yapılan sınıflandırmaların sayılmasıdır. Hatalı sınıflandırılan noktalar karışıklık matrisi (confusion matrix) ile gösterilmektedir. Hata oranı belirli bir değerin altına düşene kadar sınıflandırıcının (D) eğitilmesine devam edilir (Bezdek ve diğ., 2005).

Bazı sınıflandırıcı sistemlerinde eğitim ve test verisinin elde edilmesi için çeşitli metotlar kullanılmaktadır. Bunlara örnek olarak kümeleme algoritmaları, bulanık entegral sınıflandırıcı (fuzzy integral classifier), karar ağacı modelleri, bulanık sistemler ve benzerleri verilebilir.

Eğitim için klasik yöntemle etiketlenen verilerde bazen sınıflar arasında örtüşme olabilir. Bunu anlamak için veri üzerinde kümeleme algoritmaları çalıştırılarak eğitim için verilen verinin doğru sınıflandırılmış olduğu test edilebilir. Kümeleme algoritması kullanılmasının bir başka etkisi ise etiketli veride “bilinmeyen” veya “arada” olarak işaretlenen objelerin belirli bir sınıfa aktarılması ve böylece sınıflandırıcı performansında artış sağlanabilmesidir. House ve arkadaşları (1999) bu tekniği kullanarak iki sınıf etiketiyle etiketlenen eğitim verisinde (c=2) bulanık c-ortalama algoritmasını c=3 parametresiyle çalıştırarak ara bir sınıf elde etmişlerdir (Bezdek ve diğ., 2005).

4.1. Sınıflandırıcı Eğitimi

Sınıflandırıcının eğitim verisiyle eğitilmesi için farklı yaklaşımlar bulunmaktadır. Bu yaklaşımlara özet olarak değinilecektir.

4.1.1. Yinelemesiz çevrimdışı eğitim

Sınıflandırıcı fonksiyonunu (D) bulmak için eğitim verisi X_{tr} sadece bir kez kullanılmaktadır. Sınıflandırıcı için gerekli parametreler X_{tr} eğitim verisi kullanılarak elde edilmekte ve sistem tasarımı sona ermektedir. Yinelemeli bir işlem söz konusu değildir (Bezdek ve diğ., 2005).

4.1.2. Yinelemeli statik çevrimdışı eğitim

Bu yöntemde eğitim verisi X_{tr} sınıflandırıcının (D) performansını arttıracak parametrelerin daha iyi kestirilebilmesi için yinelemeli bir şekilde kullanılmaktadır. Bulanık sistem ve sinir ağlarında uygulanan bir yöntemdir (Bezdek ve diğ., 2005).

4.1.3. Dinamik çevrimiçi yinelemeli eğitim

Bu yöntemde ise başlangıçta ilk iki yöntemden biri ile sınıflandırıcı oluşturulur. Yeni veriler geldikçe, parametreler sınıflandırıcı performansını arttıracak şekilde değiştirilerek sınıflandırıcı sürekli olarak güncellenmektedir (Bezdek ve diğ., 2005).

Sınıflandırıcı performansı eğitim verisinin X_{tr} kalitesine bağlıdır. Eğer X_{tr} yeterli büyüklükte seçilir ve içinde bulunan kümeler iyi belirlenirse o zaman bu veri ile eğitilen sınıflandırıcıların az hata yapacağı söylenebilir. Burada karşılaşılabilecek bir sorun çok büyük X_{tr} için işlemci zamanı ve disk alanıdır. Bu sorunların aşılması için birçok kişi tarafından eğitim verisinin incelenmesi ve sınıflandırıcı performansını düşürmeyecek şekilde azaltılması üzerine de çalışmalar yapılmaktadır.

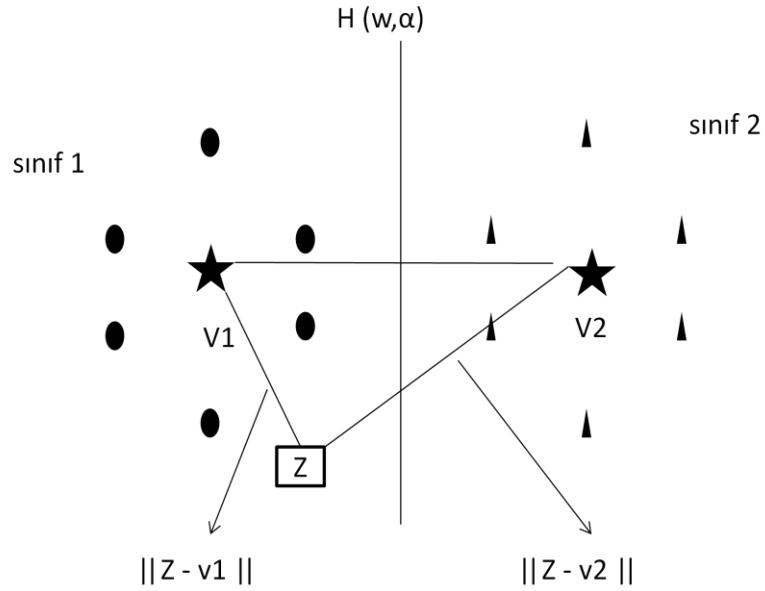
4.2. Sınıfı Temsil Eden Örnek Nokta (İlk Örnek) ile Sınıflandırıcı Tasarımı

Sınıflandırma için seçilen özelliklerden her sınıf için o sınıfı en iyi temsil eden örnek bir nokta belirlenmektedir. Sınıflandırma algoritması bu örnek nokta etrafında şekillenmektedir.

4.2.1. En yakın ilk örnek nokta sınıflandırıcı

Veri dizisindeki noktaların sınıf ilk örneklerine olan uzaklığı hesaplanmakta ve hangi sınıfın ilk örneğine yakınsa nokta o sınıfın etiketiyle etiketlenmektedir. Burada kullanılan uzaklık ölçüm metotları birinci bölümde bahsedilen metotlardır. Eğer her örnek nokta için dairesel bir komşuluk o sınıfa ait kabul edilecekse öklit normu, elmas şeklinde bir komşuluk o sınıfa ait kabul edilecekse 1-norm uzaklık ölçüm normu olarak seçilebilir.

Sınıf ilk örnekleri (prototip) eğitim için verilen veriden elde edilmektedir. Örneğin ilk örnek olarak her sınıfa ait ortalama değer belirlenmişse bu durumda eğitim için verilen veride o sınıfın etiketiyle etiketli noktaların ortalama değeri alınarak ilk örnek oluşturulmaktadır. İlk örnekler oluşturulduktan sonra geriye sadece uzaklık ölçüm metodunu seçmek kalmaktadır. Böylece sınıflandırıcı tasarımı tamamlanmaktadır. Oluşturulan sınıflandırıcıya test için verilen veri en yakın olduğu ilk örneğin sınıfına ait etiketle etiketlenecektir. Şekil 4.1'de v_1 ve v_2 ilk örnek noktaları kullanılarak z objesinin sınıflandırılması gösterilmiştir. z objesi v_1 sınıfına daha yakın olduğu için v_1 sınıfı için daha yüksek üyelik derecesine sahip olacaktır (Bezdek ve diğ., 2005).



Şekil 4. 1: İlk örneklere göre sınıflandırma

4.2.2. Çoklu ilk örnek (prototip) tasarımı

Sınıfları temsil etmesi için seçilen tek bir noktadan oluşan ilk örnek çoğunlukla yetersiz kalmaktadır. İlk örnek olarak belirlenen o noktaya yakın başka sınıfların da olma ihtimali yüksektir. Bu durumda birden fazla sayıda nokta ilk örnek olarak belirlenmektedir.

En yakın ilk örnek sınıflandırıcılarını kullanmak basit ve efektif bir yöntemdir. Ancak buradaki en büyük problem ilk örneklerin oluşturulmasıdır ve bu hiç de kolay bir işlem değildir.

Çoklu ilk örnek oluşturmanın en basit yollarından biri etiketli veri üzerinde X_{tr} her defasında bir sınıf için herhangi bir kümeleme algoritması çalıştırmaktır. Böylece o sınıfa ait bir yada daha fazla ilk örnek üretilmiş olacaktır. Bu ilk örnekler sınıflandırıcı tasarımı için kullanılmaktadır. Eğitim verisinin X_{tr} tamamına tüm sınıflar için kümeleme algoritması uygulanarak da ilk örnekler elde edilebilmektedir (Bezdek ve diğ., 2005).

4.3. En Yakın Komşu Sınıflandırıcı

İlk örnek sınıflandırıcıdan sonra sınıflandırıcı tasarımında çok kullanılan bir başka yöntem k-en yakın komşu (k-nn) kuralıdır. Bu yöntem her sınıf için etiketlenmiş örneklere ihtiyaç duymaktadır. Bu yöntemde belirlenmesi gereken en önemli parametre k değeridir. Sınıflandırma yapılırken etiketlenmemiş bir obje için en yakınında bulunan k adet objeye bakılır ve en çok hangi sınıftan komşu varsa obje o sınıfın etiketi ile etiketlenir. Bulanık yaklaşımla düşünüldüğünde objeye her sınıf için k komşuluğunda bulunan sınıf yoğunluğuyla orantılı bir üyelik derecesi verilir.

Bulanık k-en yakın komşu (k-nn) modeli her obje için sınıflara aitliğini gösteren bulanık üyelik dereceleri üreten bir algoritmadır. Bu üyelik dereceleri daha sonra klasik (0 veya 1) değerlere dönüştürülmektedir (Bezdek ve diğ., 2005).

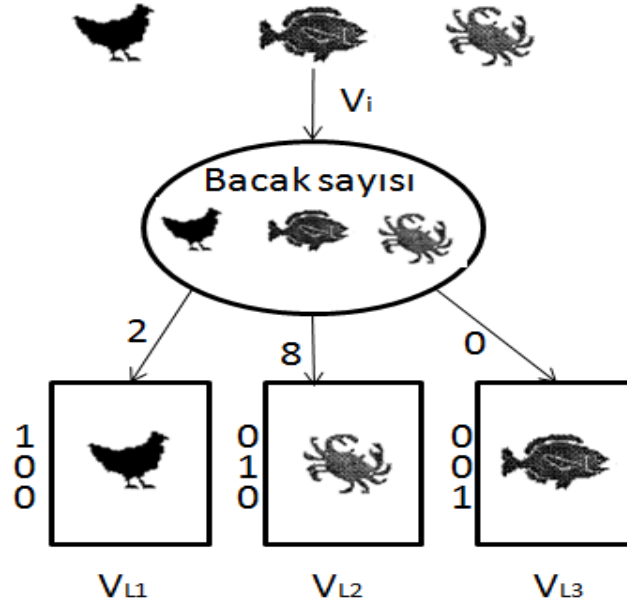
4.4. Bulanık Kural Tabanlı Sınıflandırıcı

Bulanık kural tabanlı sınıflandırıcı kullanmanın bir avantajı kural tabanlı işlemler sürerken obje etiketleri bulanık değerleriyle kalmakta ve en son aşamada klasik değerlere çevrilmektedir.

Eğitim için verilen veriden kural çıkarımı yapmak için birçok yöntem bulunmaktadır. Bunlara örnek olarak karar ağaçları, kümeleme ve deneysel yöntemler verilebilir.

4.4.1. Klasik karar ağaçları

Karar ağaçları, sınıflandırıcı tasarımı için kural tabanlı ağ yaklaşımını tanımlayan basit ve doğal bir yöntemdir. Bu yöntemde en önemli unsur iyi bir sınıflandırma ağacının nasıl oluşturulacağıdır. Karar ağaçları düğüm noktaları (nodes) ve uç noktalardan (leaves) oluşmaktadır.



Şekil 4. 2: Üç sınıftan oluşan karar ağacı (Bezdek ve diğ., 2005)

Şekil 4.2'de örnek bir karar ağacı görülmektedir (Bezdek ve diğ., 2005). Elips şeklinde olanlar düğüm noktalarını dikdörtgen şekiller ise uç noktaları ifade etmektedir. Sınıflandırma yapmak için sadece canlılara ait bacak sayıları kullanılmıştır. Burada V_1 düğüm noktasında bacak sayısına göre yönlendirme

yapılmaktadır. Her düğüm noktasında bir soru sorulmakta ve cevaba göre yönlendirme yapılmaktadır. Uç noktalarda (V_{L1} , V_{L2} , V_{L3}) herhangi bir sorgulama yapılmamakta sadece o noktaya gelen obje uç noktanın etiketi ile etiketlenmektedir.

Bu örnekte sayısal özellik kullanılmaktadır. Sayısal özellik yanında kategorik özellikler de kullanılabilir. Kategorik özellikler kural tabanlı sistemlerde kullanılan dilsel özelliklere benzemektedir. Dilsel değişkenlerin farkı üyelik fonksiyonları ile ifade edilmeleri ve her dilsel değişken karşılığında sayısal bir değer aralığı bulunmasıdır.

Yukarıda belirtilen örneği kategorik özellikler ile de gerçekleştirmek mümkündür. Bacak sayısı yerine derinin türü olarak kategorik bir özellik belirlenebilir. Bu sayede $V1$ düğüm noktasında bacak sayısı sorusu yerine ne tür bir deriye sahip olduğu sorusu yöneltilebilir. Cevap olarak tavuk için tüylü, yengeç için sert ve balık için pullu kategorik değerleri verilerek ilk düğüm noktasında yönlendirme yapılabilir. Böylelikle kategorik özellik ile sınıflandırma yapılabilir (Bezdek ve diğ., 2005).

4.4.2. Klasik karar ağaçlarından kural çıkarımı

Yukarıdaki örnekte belirtilen karar ağacı ile sınıflandırma yapılabileceği aşikar bir durumdur. Bu tür bir sınıflandırıcı tasarımının önceki açıklanan sınıflandırıcılardan farkı oluşumundaki farklılıklardır. Örneğe göre aşağıdaki kurallar çıkarılabilir (Bezdek ve diğ., 2005):

Eğer ($n_L = 2$) o halde $D_{DT}(n_L) = e_1$

Eğer ($n_L = 8$) o halde $D_{DT}(n_L) = e_2$

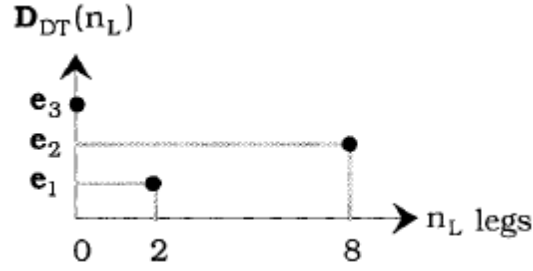
Eğer ($n_L = 0$) o halde $D_{DT}(n_L) = e_3$

Burada D_{DT} karar ağacı (Decision Tree - DT) ile oluşturulan sınıflandırıcıyı ifade etmektedir. n_L ise objeye ait özelliği (burada bacak sayısı) göstermektedir. Örneğin birinci kural için "Eğer obje 2 bacaklıysa sınıfı e_1 'dir" anlamı çıkmaktadır.

Kural tabanlı sistemlerin birinci avantajı diğer sistemlerde yapılan matematiksel hesaplar insanlara anlamlı gelmemektedir, ancak kural tabanlı sistemde örnekte de görüldüğü gibi sınıflandırmanın başlangıcından sonuna kadar kolayca anlaşılabilir

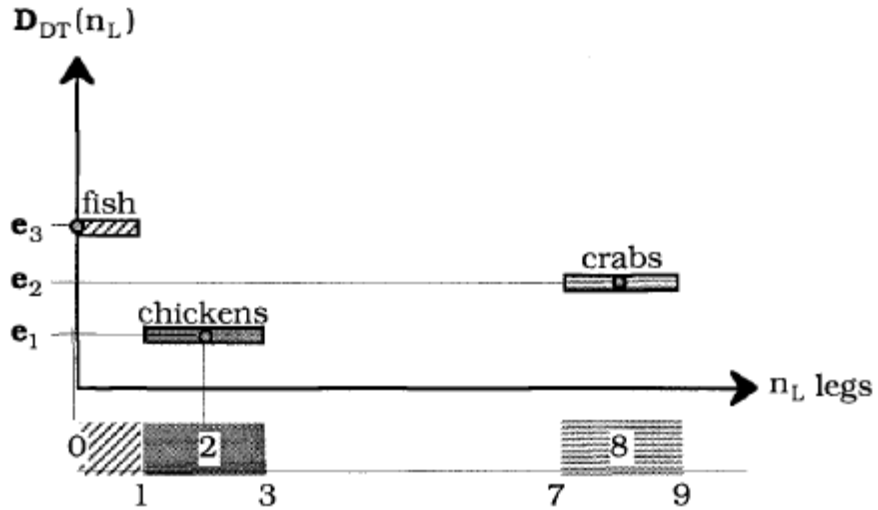
bir yapı söz konusudur. Sistem tarafından oluşturulan her kural insanlar tarafından kolaylıkla anlaşılabilir.

Kural tabanlı sınıflandırıcıların bir başka avantajı ise geometrik olarak kolaylıkla gösterilebilir olmalarıdır. Şekil 4.3'te oluşturulan kurallar geometrik olarak gösterilmektedir (Bezdek ve diğ., 2005).



Şekil 4. 3: Kuralların geometrik gösterimi (Bezdek ve diğ., 2005)

Bu örnekte 6 bacaklı bir canlı geldiğinde sınıflandırma işlemi hata ile sonuçlanacaktır veya bir insan iki bacaklı olduğu için tavuk olarak sınıflandırılacaktır. Bu durum klasik karar ağacının bir eksikliğidir. Bu problem her sınıf için kabul edilebilir değer aralığı genişletilerek çözülmeye çalışılmaktadır.



Şekil 4. 4: Kurallar için genişletilmiş değer aralığı (Bezdek ve diğ., 2005)

Şekil 4.4'te her sınıf için süreklilik arz eden bir değer aralığı verilmiştir. Bu işlem genellikle orta değerlerin her iki tarafı için bir bitiş değeri belirtilerek yapılmaktadır (Bezdek ve diğ., 2005).

4.4.3. Klasik karar ağacı tasarımı

Karar ağacı oluşturmak için en çok kullanılan algoritmalar Quinlan (1983, 1986) tarafından geliştirilen ID3 (Iterative Dichotomiser) metodu ve bu metodun geliştirilmiş olan C4.5 ile Breiman ve diğ. (1984) tarafından geliştirilen CART (Classification and Regression Tree) yaklaşımıdır. ID3 daha çok küçük boyutlu kategorik veriler için geliştirilmiştir. C4.5 algoritması ise daha gelişmiş ve sık kullanılan bir algoritmadır.

Chang ve Pavlidis (1977) karar ağaçları için bulanık yaklaşımı geliştirmişlerdir. Bulanık karar ağacında düğüm noktalarından başka bir düğüme veya uç noktaya geçiş yapılırken bir ağırlık değeri bulunmaktadır. Uç noktalara gelindiğinde bu ağırlık değerlerine göre bir üyelik derecesi oluşmaktadır.

4.4.4. Bulanık sistem modelleri ve fonksiyon kestirimi



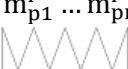
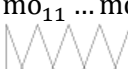

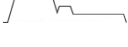
Bu bölümde bulanık kural tabanlı sistemlerin iki ana türü olan Mamdani-Assilian, (1975) ve Takagi-Sugeno (1985) modelleri anlatılacaktır.

Giriş verisi $X = \{x_1, \dots, x_n\} \subset R^p$, çıkış ise $Y = \{y_1, \dots, y_n\} \subset R^q$ olsun. S fonksiyonunun $S : R^p \rightarrow R^q$, $y_k = S(x_k)$, $k=1, \dots, n$ yani $Y = S[X]$ sağlayan bilinmeyen bir fonksiyon olduğu varsayılmaktadır. X ve Y giriş-çıkış verisi, $XY = \{ \{x_k, y_k\}^T = \{x_{1k}, \dots, x_{pk}, y_{1k}, \dots, y_{qk}\}^T; k=1, \dots, n \}$ ise X ve Y'de bulunan giriş ve çıkış vektörlerini birleştiren bir sistemdir. XY'yi kullanarak iyi bir S oluşturmak eğitilmiş öğrenme (supervised learning, interpolation, collocation, function approximation) olarak adlandırılmaktadır. Bulanık sistemde, $p=1$ olduğunda tekli giriş (single input), $p>1$ olduğunda çoklu giriş (multiple input); $q=1$ olduğunda tekli çıkış (single output), $q>1$ olduğunda ise çoklu çıkış (multiple output) bulunmaktadır.

S fonksiyonunun kestirimi için üç ana basamak bulunmaktadır. Birinci olarak sistemin yapısı tanımlanmalıdır. Örneğin karar ağacı modeli seçilmişse ağacın kaç

seviyeden oluşacağı, her seviyede kaç düğüm noktası olacağı, uç nokta sayısı, ağırlık değerleri ve benzeri unsurlar yapıyı tanımlamaktadır. Bulanık sistem için yapının tanımlanması her dilsel değişken için değer aralığı belirlenmesi ve benzeri unsurları kapsamaktadır. İkinci olarak sistem yapısında tanımlanan parametrelerin kestirilmesi bulunmaktadır. Son olarak oluşturulan sistemin test edilmesi bulunmaktadır (Bezdek ve diğ., 2005).

Bulanık sistemler S fonksiyonunu kural kümesi $R = \{R_1, \dots, R_M\}$ ile kestirirler. Temel MA ve TS modelleri Şekil 4.5'te özetlenmektedir (Bezdek ve diğ., 2005).

| Adım 1 | Adım 2 | Adım 3 |
|--|--|--|
| Giriş X: Özellik vektörleri girişe verilir. | Bulanıklaştırma (Fuzzify) Her kural için üyelik fonksiyonları belirlenir. | Kural Birleştirme Üyelik fonksiyonları birleştirilir. |
| X_1 → X_2 → X_3 → | $m_{11}^i \dots m_{1r}^i$  $D_1 \leftrightarrow L_1$ $m_{k1}^i \dots m_{kr}^i$  $D_k \leftrightarrow L_k$ $m_{p1}^i \dots m_{pr}^i$  $D_p \leftrightarrow L_p$ | $\alpha_1(x) = \cap (m^1(x))$ \vdots $\alpha_M(x) = \cap (m^M(x))$ |
| Adım 4 | | |
| Çıkış | | |
| Çıkış değeri seçilen sistem modeline göre hesaplanır. | | |
| <p style="text-align: center;">TS Modeli</p> $\{u_i(x) : 1 \leq i \leq M\}$ $S_{TS}(x) = \frac{\sum_{i=1}^M \alpha_i(x) u_i(x)}{\sum_{j=1}^M \alpha_j(x)}$ | <p style="text-align: center;">MA modeli</p> $z^i(x) = \psi(\alpha_i(x), mo^i); i = 1, \dots, M$ <div style="display: flex; flex-direction: column; align-items: center;"> <div style="display: flex; align-items: center; margin-bottom: 10px;"> <div style="font-size: 2em; margin-right: 10px;">↓</div> <div style="text-align: center;"> $mo_{11} \dots mo_{1s}$  $D_{o1} \leftrightarrow L_{O1}$ </div> </div> <div style="display: flex; align-items: center; margin-bottom: 10px;"> <div style="font-size: 2em; margin-right: 10px;">↓</div> <div style="text-align: center;"> $mo_{q1} \dots mo_{qs}$  $D_{os} \leftrightarrow L_{Os}$ </div> </div> <div style="display: flex; align-items: center; margin-bottom: 10px;"> <div style="font-size: 2em; margin-right: 10px;">↓</div> <div style="text-align: center;">  </div> </div> <div style="display: flex; align-items: center;"> <div style="font-size: 2em; margin-right: 10px;">↓</div> <div style="text-align: center;"> $S_{MA}(x) = \theta(\alpha(x), Z(x), U, D_F)$ </div> </div> </div> | |

Şekil 4. 5: MA ve TS modellerinin yapısı

Birinci (1) adımda R^p uzayında (özellik uzayı) x vektörünü giriş olarak almaktadır. Örneğin yükseklik, genişlik ve benzeri özelliklerin X için değer vektörleri giriş verilmektedir. İkinci (2) adıma her giriş değişkeninin değer aralığı belirlenerek başlanmaktadır. Örneğin k 'inci giriş vektörü (x_k) için D_k değer aralığında her dilsel değişken L_k için bir alt değer aralığı (r_k) belirlenmektedir. m ifadesi örneğin m_{11} birinci özellik vektörü (x_1) için belirtilen değer aralığının birinci alt aralığının üyelik fonksiyonunu belirtmektedir. m ifadesinin üzerinde bulunan i ise kural kümesindeki i 'nci kuralı (R_i) belirtmektedir. Şekilde üyelik fonksiyonu olarak üçgen üyelik fonksiyonu belirtilmiştir ancak farklı türlerde üyelik fonksiyonları da bulunmaktadır. İkinci adım bulanıklaştırma (fuzzification) olarak adlandırılmaktadır. Üçüncü (3) adımda, oluşturulan üyelik fonksiyonlarına göre kuralların ilk kısmı (şart koşma kısmı, LHS oluşturulan kuralda eşitliğin sol tarafı anlamına gelmektedir) oluşturulmaktadır. α her kural için ağırlıkları ifade etmektedir. Eşitliğin sol tarafı oluşturulurken üyelik fonksiyonları arasında kullanılacak T normları (çarpım, en büyük olanı al ve benzeri) da belirlenmektedir. Dördüncü (4) adımda, $S(x)$ çıkış vektörü oluşturulmaktadır. TS modeli için çıkış değerleri M üyelik fonksiyonu ve α ağırlık değerlerine göre Denklem (4.1) ile hesaplanmaktadır.

$$S_{TS}(x) = \frac{\sum_{i=1}^M \alpha_i(x) \cdot u_i(x)}{\sum_{j=1}^M \alpha_j(x)} \quad (4.1)$$

$$S_{MA}(x) = \theta(\alpha(x), Z(x), U, D_F) \quad (4.2)$$

MA modeli TS modeline göre daha karmaşıktır. MA modelin çıkışı Denklem (4.2) ile hesaplanmaktadır (Bezdek ve diğ., 2005).

Sınıflandırıcı tasarımı için iki modelden hangisi kullanılırsa kullanılsın sonuç olarak etiket vektörü çıkmaktadır.

4.4.5. Bulanık sistemler için kural çıkarımı

Bulanık sistemler için kuralların çıkarılmasında farklı yöntemler kullanılabilir. Bu yöntemlerden karar ağaçları ve kümeleme yöntemleri ile kural çıkarılması üzerinde durulacaktır.

Klasik karar ağaçları ve klasik karar ağaçlarından kural çıkarımı Bölüm 4.4.1 ve 4.4.2'de açıklanmıştır. Klasik karar ağaçlarında düğüm noktalarından ayrılan kollara "0" veya "1" ağırlık değeri verilmekteyken bulanık yöntemde bu ağırlıklar [0-1] aralığında değerlerden oluşmaktadır. Kural çıkarımında bu yeni değerler kullanılmakta ve böylece bulanık kurallar oluşturulmaktadır.

Kural çıkarımında diğer yöntem ise kümeleme algoritmalarıyla kural çıkarılmasıdır. Burada kümeleme algoritmaları her sınıftan örnekler içeren veri üzerinde yürütülmekte ve böylece sınıflara ait küme merkezleri bulunmakta ve her sınıf için bir üyelik fonksiyonu oluşturulmaktadır. Bu üyelik fonksiyonlarına sınıfa aitlik oranına göre bir ağırlık değeri verilmekte ve bu ağırlık değeri de kullanılarak bulanık kurallar her sınıf için oluşturulmaktadır.

5. BULANIK BÖLÜTLEME

Klasik bölütlemelerde pikseller bölgelere ayrılırken, eğer piksel aranan bölgeden ise "1", değilse "0" olarak işaretlenmektedir. Başka bir ifadeyle pikseller için şu şekilde bir üyelik söz konusudur:

$$m_{R_i}(i, j) = \begin{cases} 1 & ;(i,j) \in R_i \\ 0 & ;(i,j) \in \text{değil } R_i \end{cases} \quad (5.1)$$

Denklem (5.1)'de i, j konumundaki piksel R_i bölgesinde ise (görüntüde gri seviyesine göre bölütleme yapıldığında R_i bölgesi belirli bir gri seviye aralığını ifade etmektedir) piksel değerini "1" yapan, değilse "0" yapan bir üyelik fonksiyonu verilmiştir. Çoğu durumda pikselin bir bölgeye ait olup olmadığını belirlemek kolay olmamaktadır. Bunun nedeni obje kenarlarında keskin bir geçiş bulunmamasıdır. Keskin bir geçiş bulunmamasına rağmen görüntü üzerinde koşturulan bir maske neticesinde pikselin R_i bölgesinde olup olmadığına karar verilmektedir. Ancak gerçekte bu piksel sınırda bulunmakta ve her iki bölgeden de olma ihtimali bulunmaktadır. Bu noktada bulanık küme kavramı devreye girmektedir. Bulanık bölütlemelerde pikselin her bölgeye ne ölçüde ait olduğunu gösteren bir üyelik derecesi bulunmaktadır. Klasik yöntemde "0" ve "1" ile ifade edilen üyelik dereceleri, bulanık yöntemde [0-1] arasındaki sayılarla ifade edilmektedir (Bezdek ve diğ., 2004).

5.1. Eşik Değeri ile Bölütleme

Eşik değerini bulmak için bulanık yöntemler iki ana kategoriye ayrılmaktadır; birincisi görüntüdeki piksellere atanan üyelik derecelerine göre bulunan bulanıklık indeksini minimum veya maksimum yapan t değerini bulan yöntemler, ikincisi ise gri değerleri iki kümeye ayıran kümeleme yöntemleridir.

Pikseller için üyelik derecelerini belirlemek üzere gri değerlerini üyelik derecesine çeviren üyelik fonksiyonları $m_{R_o}(X_i)$ kullanılmaktadır. Bu fonksiyonlar gri değerler

[0-1] aralığına çevrilerek elde edilebilmektedir. Bu çevirme işlemi için kullanılan popüler bir yöntem aşağıda sunulmaktadır (Bezdek ve diğ., 2005).

$$m_{R_0}(I(X)) = S(I(X)) = \begin{cases} 0 & ; I(X) \leq a \\ 1/2 \left(\frac{I(X)-a}{b-a} \right)^2 & ; a < I(X) \leq b \\ 1 - 1/2 \left(\frac{I(X)-c}{c-b} \right)^2 & ; b < I(X) \leq c \\ 1 & ; I(X) > c \end{cases} \quad (5.2)$$

Görüntüdeki gri değerler Denklem (5.2) ile gösterilen üyelik fonksiyonu ile [0-1] aralığında üyelik derecelerine dönüştürülmektedir. Fonksiyon tanımında a ve c değerleri genelde sabit kalmakta, b değeri değiştirilerek her b için bulanıklık indisi hesaplanmaktadır. Böylece en uygun b değeri bulunmaktadır. Eşik değeri bulunan b değeri olarak belirlenmektedir. I(X) gri değeri ifade etmektedir.

Bulanıklığı ölçmek için birçok formül bulunmaktadır. Burada bulanıklığın lineer ölçüm indeksi verilecektir (Kaufmann, 1975):

$$Y_q(P_{IJ}) = \frac{2}{n} \sum_{i=1}^n \min(m_{R_0}(X_i), 1 - m_{R_0}(X_i)) \quad (5.3)$$

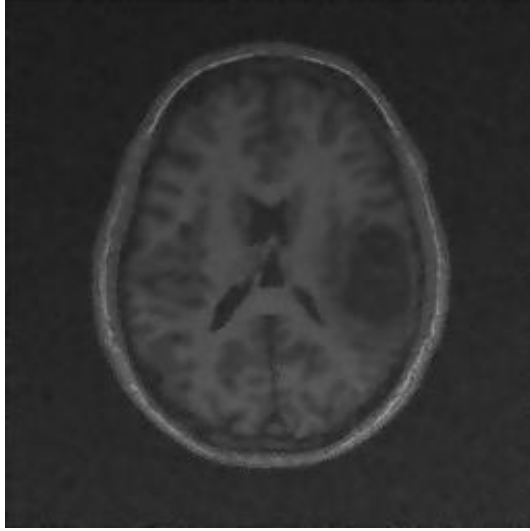
Denklem (5.3) ile her pikselin maksimum üyelik derecesi kullanılarak bir hesap yapılmaktadır. Denklem (5.3) en optimum değeri en uygun b değeri ile bölütleme yapıldığında verecektir.

Gri seviyeli görüntülerde obje sınırları genelde bulanık yapıdadır ve görüntü işleme esnasında bu sınırlar zarar görmektedir. Görüntüye eşik değeri uygulandığında obje sınırlarındaki belirsizlikler yok olmakta, objelerin şekil ve boyutları değişmektedir. Bu nedenle obje sınırlarındaki belirsizliklerin son karar aşamasına kadar saklanması önem arz etmektedir. Bulanık yöntemler bu imkanı sağlamaktadırlar (Bezdek ve diğ., 2005).

5.2. Kümeleme ile Bölütleme

Kümeleme algoritmaları giriş olarak verilen değerleri belirli kıstaslara göre gruplandıran algoritmalarıdır. Bu kıstaslar verinin kaç kümeye ayrılacağı, uzaklık

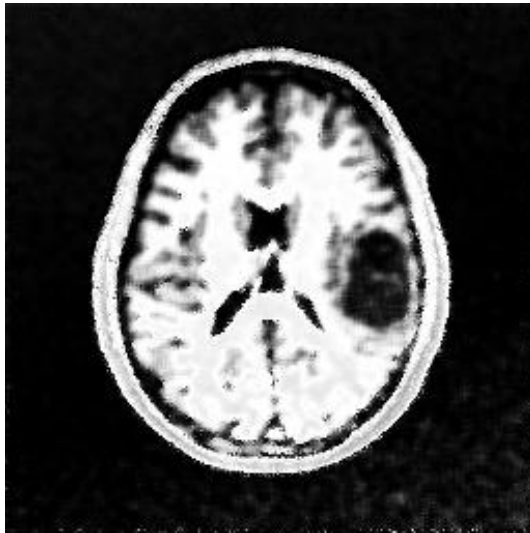
kriteri ve benzeri olabilmektedir. Bölütleme algoritmalarında kümeleme metotlarını kullanarak görüntü gri seviyesine göre gruplara ayrılabilir. Kümeleme algoritmasına görüntüyü iki kümeye ayırması kistas olarak verildiğinde sonuç olarak ikili (binary) bir görüntü elde edilebilir. Ayrılan kümelerin biri arka plan diğeri ise obje olarak nitelendirilebilir. Böylece kümeleme algoritması görüntü bölütleme için kullanılabilir.



a) Orijinal görüntü



b) K-ortalama ile bölütleme ($k=2$)



c) Bulanık C-ortalama (FCM) ile bölütleme ($c=2$)

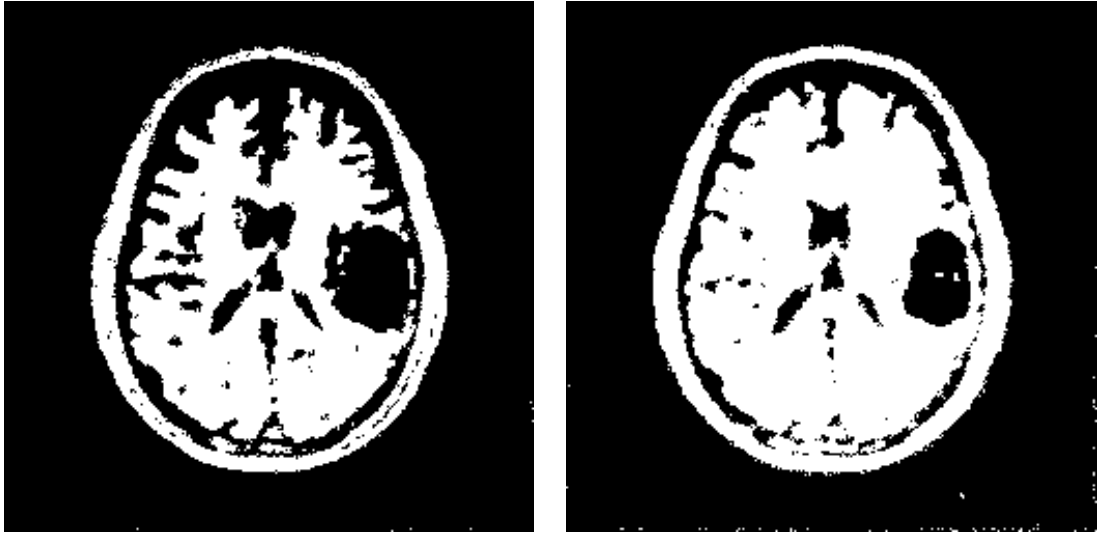


d) FCM ile kümelene görüntünün durulaştırılmış hali

Şekil 5. 1: Kümeleme ile bölütleme

Şekil 5.1’de k-ortalama (k-means) ve bulanık c-ortalama (FCM – Fuzzy C-Means) kümeleme algoritmalarının beyin MR görüntüsüne uygulanması sonucu elde edilen görüntüler bulunmaktadır. Bulanık algoritmayla elde edilen görüntüde obje sınırlarında yumuşak bir geçiş olduğu gözükmemektedir. Aslında bu bize bulanık algoritmalarda son aşamaya (durulaştırma aşaması) kadar detay bilgilerin saklandığını göstermektedir. Bulanık c-ortalama algoritmasıyla görüntüdeki her piksele gri seviyesine göre iki farklı küme için üyelik derecesi verilmiş ve şekildeki görüntü elde edilmiştir. Durulaştırma işleminde üyelik derecesi 0.5 değerinden büyük veya eşit olan pikseller birinci kümeye diğerleri ikinci kümeye dahil edilmiştir. Durulaştırma işleminde farklı yöntemler uygulanarak daha iyi sonuçlar elde etmek mümkündür.

Şekil 5.2’de durulaştırma işleminde 0.5 yerine 0.3 ve 0.6 üyelik değerleri kullanılmıştır. Görüldüğü gibi bulanık kümeleme algoritmalarında detay bilgiler saklandığı için son aşama olan karar aşamasında farklı varyasyonlar uygulayarak farklı sonuçlar elde edilebilmektedir. Bu yetenek birçok uygulamada sonuca büyük katkı yapmaktadır.



a) Eşik değeri 0.6 ile durulaştırma

b) Eşik değeri 0.3 ile durulaştırma

Şekil 5. 2: Farklı durulaştırma sonuçları

Kümeleme algoritmalarına görüntü sadece piksel değeriyle verilebileceği gibi (özellikle çok bantlı görüntüler, örneğin uydu görüntüleri) piksel değerlerinin çeşitli fonksiyonları halinde de verilebilmekte veya her ikisi birden de giriş değerini

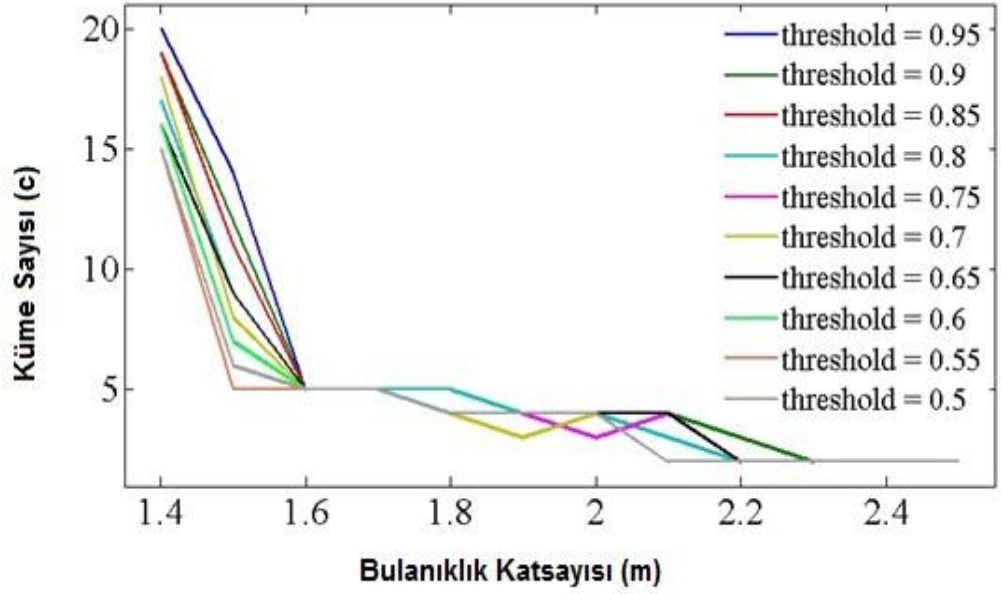
oluşturabilmektedir. Böyle durumlarda her piksel için birden fazla özellik bulunmakta ve bölütleme işlemi bu özelliklere göre yapılmaktadır (Gonzalez ve diğ., 2004).

Sebiskveradze ve diğ. (2009), IR görüntüsünde kanserli bölgelerin tespiti için FCM algoritmasını kullanmıştır. Yaptıkları araştırmada FCM algoritmasına parametre olarak girilen küme sayısı(c) ve bulanıklık katsayısını (m) kestirebilmek için yeni bir yöntemden söz etmişlerdir. Yayınladıkları makalede k-ortalama algoritmasının da tümörlü bölgelerin tespitinde kullanıldığını yazmışlardır. FCM'de m ve c'nin kestirimi için geliştirdikleri yöntem kısaca şöyledir.

Algoritmada, Şekil 5.3'te de gösterildiği gibi ilk adımda tüm m değerleri için, belirli bir eşik değerine göre bölüt sayısı belirlenir (burada eşik değeri oluşturulan bölütler arasındaki korelasyonu karşılaştırmak için kullanılır. Korelasyon eşik değerinden büyükse iki bölüt aynı kabul edilir). Sonuçta aralarındaki korelasyon belirli bir korelasyondan az olan bölüt sayısı elde edilmiş olunur. Bu adım farklı eşik değerleri için de tekrarlanır. Böylece elde edilen verilerle, tüm eşik değerleri için her m değerinde kaç tane küme (c) oluşacağını eğrisi çizilir. Belirli bir eşik değerine sahip her eğri algoritmaya göre sürekli azalmaktadır (küme sayısı azaltılmaktadır). Artık belirli bir küme sayısından sonra azalma işlemi durmaktadır. Her eşik değeri için çizilen eğriler için azalmanın durduğu noktalara bakıldığında genelde her eğrinin aynı noktadan sonra azalmayı kestiği görülmektedir. İşte bu noktadaki küme sayısı optimum küme sayısı olarak seçilir. Bu küme sayısı her eğri için belirli bir m aralığında oluşmaktadır. Buradan da m değeri belirlenir. Böylece FCM için gerekli iki parametre de kestirilmiş olunur (Sebiskveradze ve diğ., 2009).

Kümeleme algoritmalarının ne tür bir sonuç ortaya çıkaracağını kestirmek zordur. Görüntü kümeleme algoritmasıyla bölümlere ayrıldığında birbiriyle bağlantılı olmayan objeler aynı bölümmüş gibi gözükülebilmektedir. Bir başka sorun ise görüntünün kaç kümeye ayrılacağını tespit edilmesidir

Elde edilecek bölütleme kalitesi kümeleme algoritmasında kullanılan uzaklık ölçüm birimi ile bağlantılıdır. Öklit uzaklığı bölümler birbirinden iyi bir şekilde ayrılabilirdiği ve objeler yaklaşık boyutlarda olduğu zaman efektif bir sonuç vermektedir (Bezdek ve diğ., 2005).



Şekil 5. 3: FCM için “m” ve “c” parametrelerinin kestirimi (Sebiskveradze ve diğ., 2009)

5.3. Kural-Tabanlı Bölütleme

Birçok durumda eğitimsiz bölütleme (kümeleme – clustering ile yapılan bölütleme) yeterince iyi sonuç vermemektedir. Bunun nedeni farklı nesnelere ait özellik vektörlerinin yeterince belirleyici olamamasıdır. Bu nedenle giriş verisinde bazı pikseller etiketlenerek bu piksellere benzer olanların aynı gruba alınması sağlanabilmektedir. Yarı eğitilmiş kümeleme algoritmalarında bu hususa değinilmiştir.

Kural tabanlı bölütlemeye ise böyle bir sorundan bahsetmek mümkün değildir. Etiketli giriş görüntüleri kullanılarak özellik vektörlerinden farklı nesnelere en iyi şekilde ayırabilecek kuralların oluşturulması mümkündür. Böylece test amacıyla verilen yeni görüntülerde çok daha başarılı bölütleme sonuçları elde etmek mümkün olacaktır.

Otomatik bölütlemenin en zor yönü eğitim için kullanılan veri ne kadar iyi olursa olsun, görüntüde yeni objeler olması veya objelerin eğitim verisindekinden farklı olmasının sonuca yanlış yönde etki etmesidir. Örneğin farklı hastalara ait MR görüntülerinde (örneğin beyin tümörü görüntüsü) birbirinden çok farklı anatomik yapılar bulunmaktadır ve bu nedenle bu görüntüleri otomatik olarak bölümlenecek ve

sınıflandıracak bir sistem oluşturmak çok zordur. Çoğu sistemde görüntüdeki bölümler için sadece normal veya normal değil bilgisi çıkarılmakta, normal olmayan durumlar için uzman yardımı beklenmektedir. Bunun gibi durumlarda insan düşünce sistemine yakın olan kural tabanlı bulanık sistemlerin kullanılması daha başarılı sonuçlar elde edilmesini sağlayacaktır. Tez çalışmasında yapılan tümör tespiti için bu sistem kullanılmıştır. Uygulamanın detayları sonraki bölümde anlatılacaktır.

6. BULGULAR VE TARTIŞMA

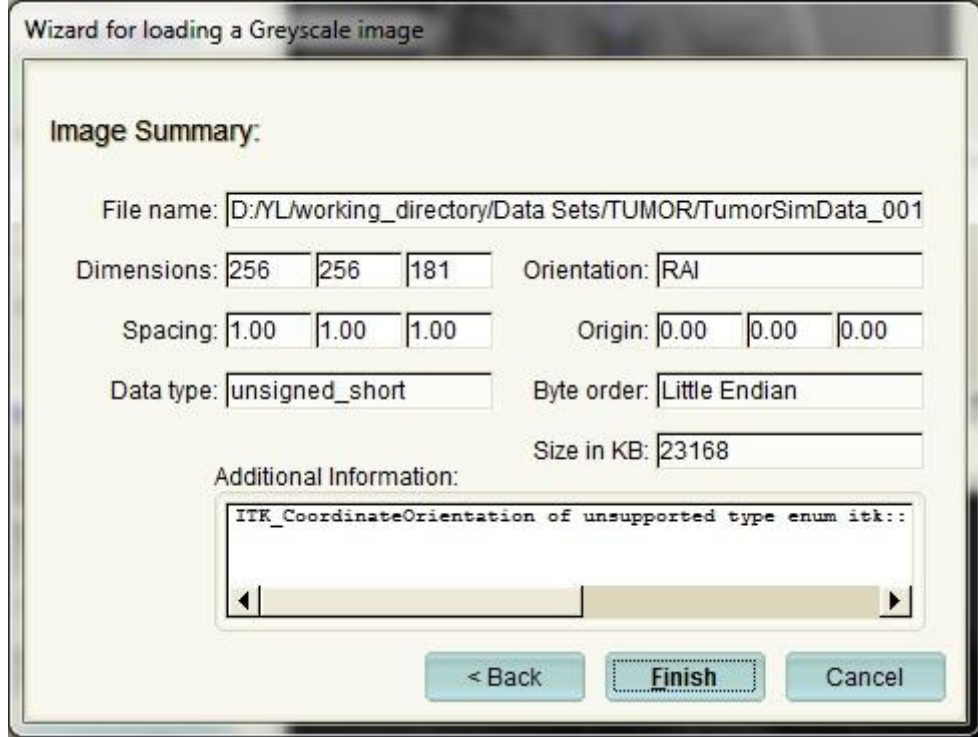
6.1. Veri Kümeleri

6.1.1. MR veri kümesi

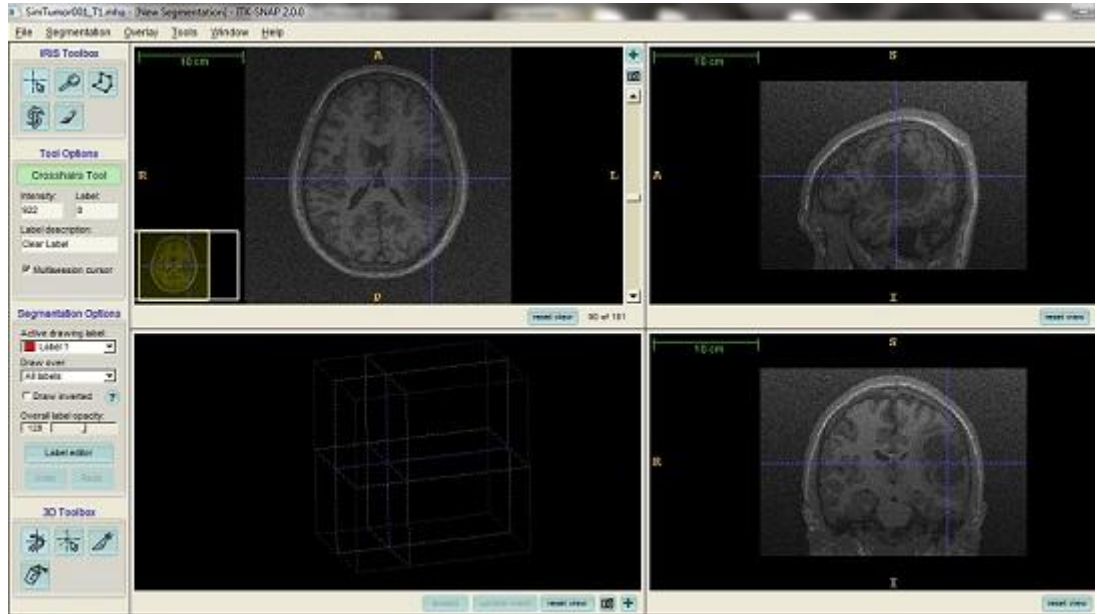
MR (Magnetik Rezonans) görüntüleme tekniğinde, görüntüler farklı dokuların manyetik alanlara farklı tepki süresi göstermesi üzerinden elde edilmektedir. Bu da istenilen doku hakkında oldukça yararlı bilgiler sunmaktadır. Bu teknik zararlı X ışınları içermediği için diğer tekniklere göre güvenilirdir. Bu nedenlerle özellikle yumuşak dokulu organlardaki teşhislerde sıkça kullanılmaktadır (Özkurt ve Özkurt, 2007)

MR görüntü sınıflandırmada en temel sorunlardan bir tanesi geliştirilen sistemlerin test edilmesi için standart bir verinin olmamasıdır. Çoğu yeni metotta test için uzman doktorlar tarafından belirlenmiş tümör ile otomatik olarak tespit edilen tümör karşılaştırılmaktadır. Ancak bu durumda algoritmaların başarısını karşılaştırmak güç olmaktadır. Bu nedenle; geliştirilen sistemde, Prastawa ve diğ. (2009) tarafından sunulan tümörlü bölgesi belirli olan yapay MR görüntüsü kullanılmıştır. Görüntü UCNIA (Utah Center for NeuroImage Analysis) web sitesinden indirilmiştir. Görüntüler mha (meta image file format) dosya formatında alınmıştır. Alınan görüntü bilgileri Şekil 6.1 'de sunulmuştur.

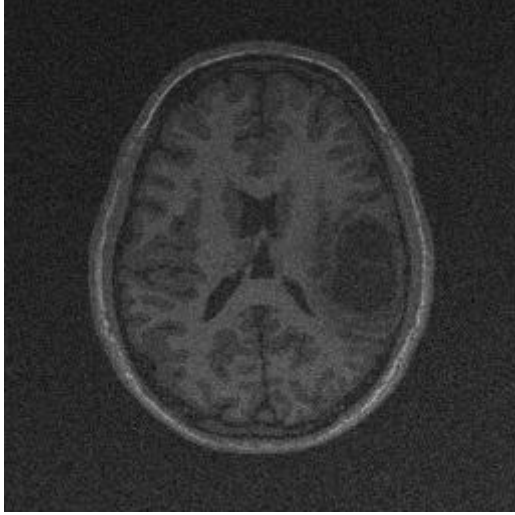
Görüntü ITK-SNAP versiyon 2.0.0 programıyla okunarak "tif" formatına çevrilmiştir. Programla görüntünün okunması Şekil 6.2 'de, T1 bandından oluşturulan "tif" dosyaları ise Şekil 6.3 'te, axial görüntüdeki yapılar ise Şekil 6.4'te sunulmuştur.



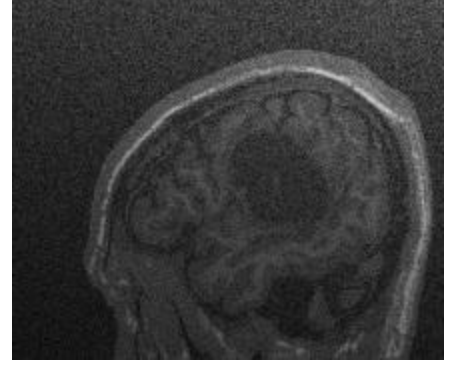
Şekil 6. 1: MR görüntüsü bilgileri



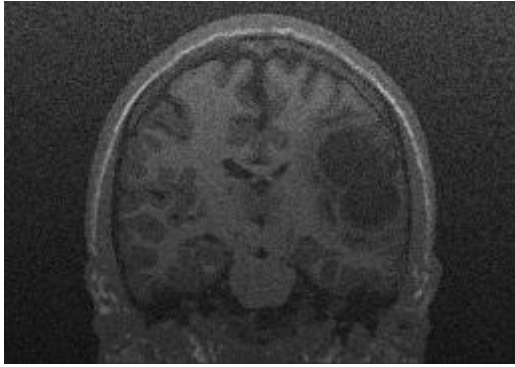
Şekil 6. 2: ITK-SNAP programıyla okunan görüntü



a) T1 Axial

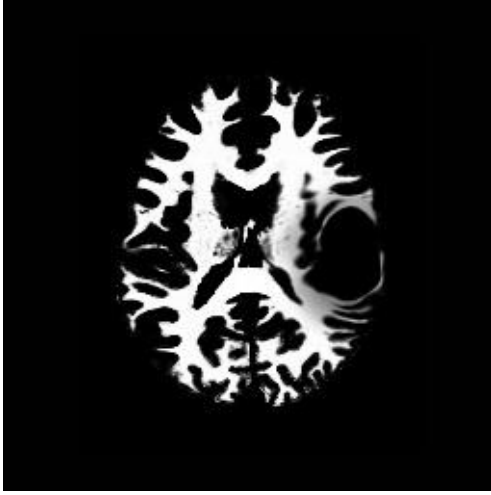


b) T1 Sagittal

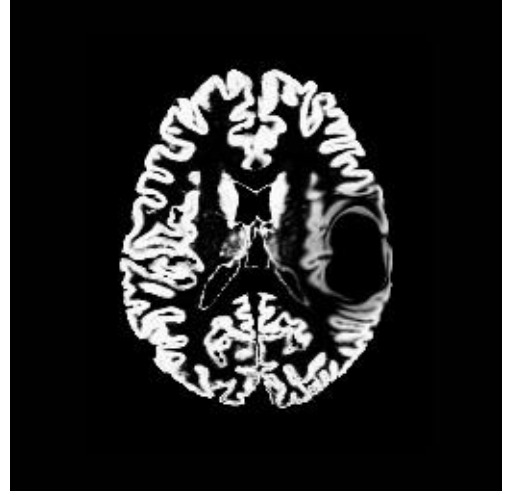


c) T1 Coronal

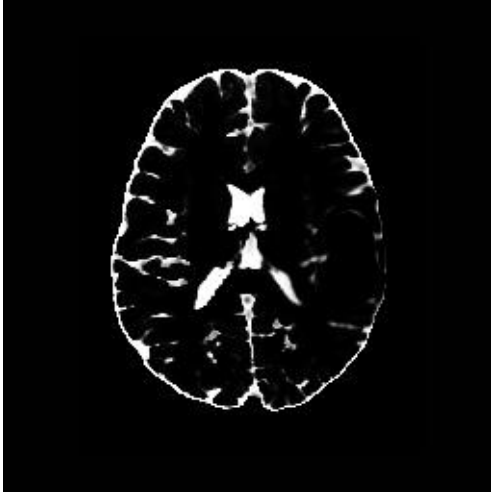
Şekil 6. 3: ITK-SNAP programıyla elde edilen görüntüler



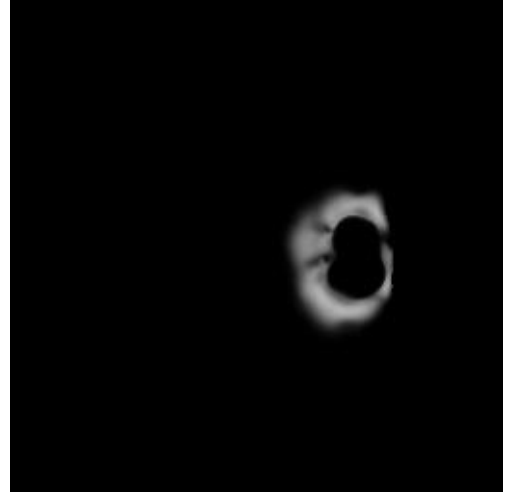
a) White matter



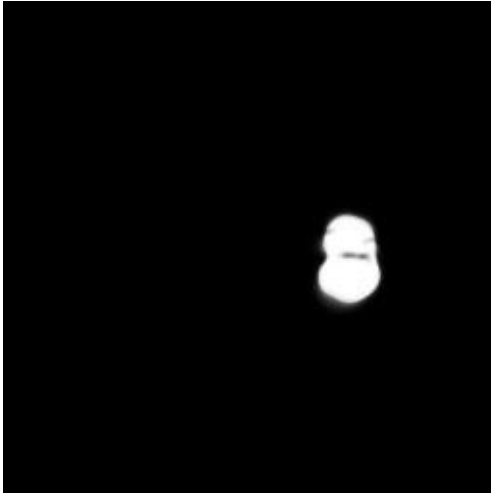
b) Gray matter



c) csf



d) Ödem



e) Tumor

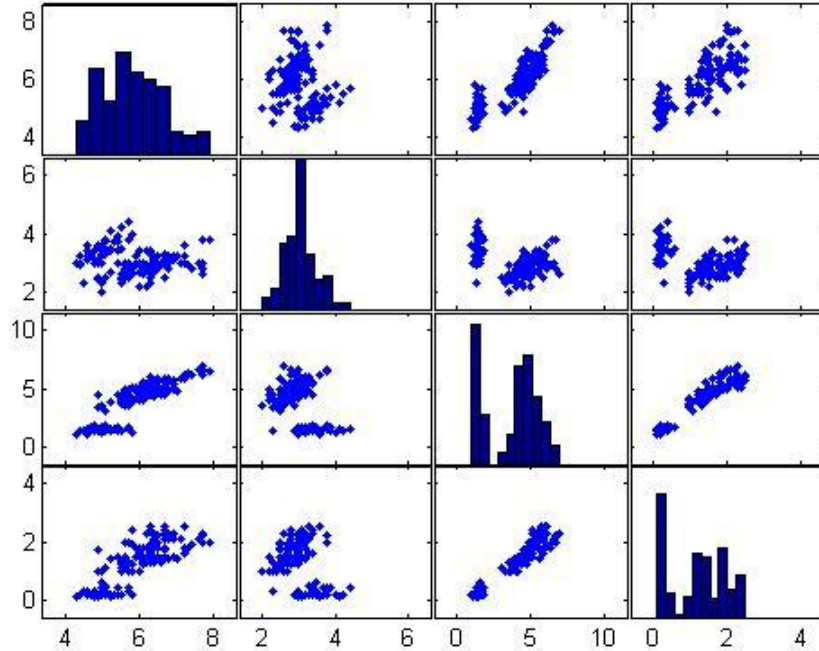
Şekil 6. 4: Axial görüntüdeki bölümler

6.1.2. Iris veri kümesi (Fisher, 1936)

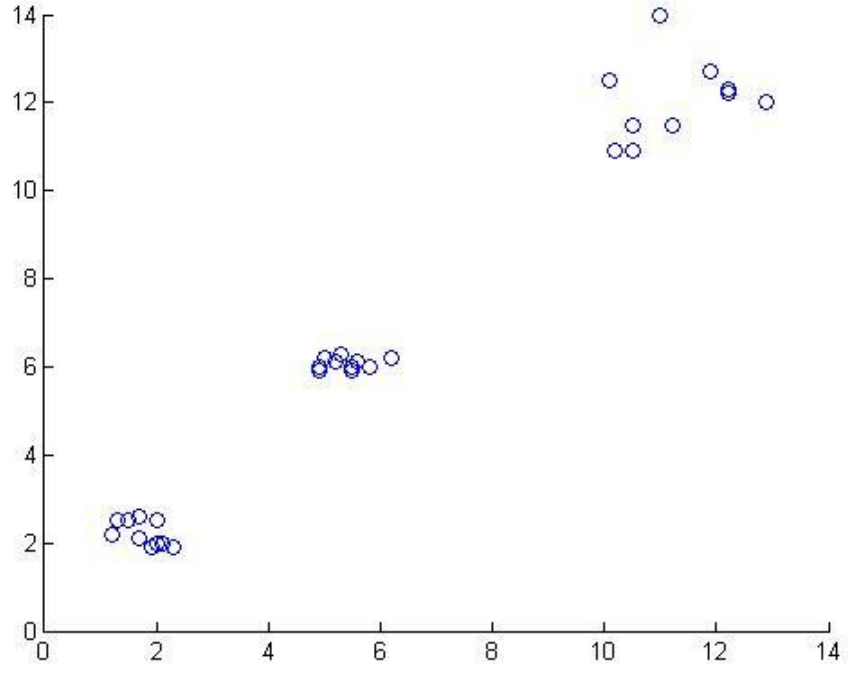
Tablo 6.1’de verilen iris veri kümesi (Fisher, 1936) herkes tarafından bilinen bir veri kümesidir. Bu veri kümesinde “sestosa” sınıfı diğer iki sınıftan (versicolor ve virginica) doğrusal bir şekilde ayrılmaktadır. Yani “sestosa” sınıfını diğer sınıflardan ayırt etmek oldukça kolaydır. Ancak “versicolor” ve “virginica” sınıfları birbirlerinden doğrusal bir şekilde ayrılmamaktadır. Bu iki sınıfı birbirinden ayırt etmek için birçok kişi farklı yöntemler geliştirmiştir (Bezdek ve diğ., 2005).

Tez çalışmasında bu veri üzerinde bulanık c-ortalama ve k-ortalama algoritmaları denenmiştir. Ayrıca bulanık kural tabanlı sistemin test edilmesinde de bu veriden yararlanılmıştır.

Şekil 6.5’te veri kümesinde bulunan dört özellik vektörünün (Sepal Length (Sepal Uzunluk), Sepal Width (Sepal Genişlik), Petal Length (Petal Uzunluk) ve Petal Width (Petal Genişlik)) karşılıklı histogramları ve diyagramları sunulmuştur. Dikkat edilirse “sestosa” sınıfının diğer iki sınıftan net bir şekilde ayrıldığı görülecektir. Diğer iki sınıfın ise iç içe girmiş olduğu ortadadır.



Şekil 6. 5: Iris veri kümesinin diyagramı



Şekil 6. 6: X_{30} veri kümesinin diyagramı

Tablo 6. 1: Fisher (1936) tarafından sunulan iris veri kümesi

| Iris setosa | | | | Iris versicolor | | | | Iris virginica | | | |
|-------------|-------------|-------------|-------------|-----------------|-------------|-------------|-------------|----------------|-------------|-------------|-------------|
| Sepal Leng. | Sepal Width | Petal Leng. | Petal Width | Sepal Leng. | Sepal Width | Petal Leng. | Petal Width | Sepal Leng. | Sepal Width | Petal Leng. | Petal Width |
| 5.1 | 3.5 | 1.4 | 0.2 | 7.0 | 3.2 | 4.7 | 1.4 | 6.3 | 3.3 | 6.0 | 2.5 |
| 4.9 | 3.0 | 1.4 | 0.2 | 6.4 | 3.2 | 4.5 | 1.5 | 5.8 | 2.7 | 5.1 | 1.9 |
| 4.7 | 3.2 | 1.3 | 0.2 | 6.9 | 3.1 | 4.9 | 1.5 | 7.1 | 3.0 | 5.9 | 2.1 |
| 4.6 | 3.1 | 1.5 | 0.2 | 5.5 | 2.3 | 4.0 | 1.3 | 6.3 | 2.9 | 5.6 | 1.8 |
| 5.0 | 3.6 | 1.4 | 0.2 | 6.5 | 2.8 | 4.6 | 1.5 | 6.5 | 3.0 | 5.8 | 2.2 |
| 5.4 | 3.9 | 1.7 | 0.4 | 5.7 | 2.8 | 4.5 | 1.3 | 7.6 | 3.0 | 6.6 | 2.1 |
| 4.6 | 3.4 | 1.4 | 0.3 | 6.3 | 3.3 | 4.7 | 1.6 | 4.9 | 2.5 | 4.5 | 1.7 |
| 5.0 | 3.4 | 1.5 | 0.2 | 4.9 | 2.4 | 3.3 | 1.0 | 7.3 | 2.9 | 6.3 | 1.8 |
| 4.4 | 2.9 | 1.4 | 0.2 | 6.6 | 2.9 | 4.6 | 1.3 | 6.7 | 2.5 | 5.8 | 1.8 |
| 4.9 | 3.1 | 1.5 | 0.1 | 5.2 | 2.7 | 3.9 | 1.4 | 7.2 | 3.6 | 6.1 | 2.5 |
| 5.4 | 3.7 | 1.5 | 0.2 | 5.0 | 2.0 | 3.5 | 1.0 | 6.5 | 3.2 | 5.1 | 2.0 |
| 4.8 | 3.4 | 1.6 | 0.2 | 5.9 | 3.0 | 4.2 | 1.5 | 6.4 | 2.7 | 5.3 | 1.9 |
| 4.8 | 3.0 | 1.4 | 0.1 | 6.0 | 2.2 | 4.0 | 1.0 | 6.8 | 3.0 | 5.5 | 2.1 |
| 4.3 | 3.0 | 1.1 | 0.1 | 6.1 | 2.9 | 4.7 | 1.4 | 5.7 | 2.5 | 5.0 | 2.0 |
| 5.8 | 4.0 | 1.2 | 0.2 | 5.6 | 2.9 | 3.6 | 1.3 | 5.8 | 2.8 | 5.1 | 2.4 |
| 5.7 | 4.4 | 1.5 | 0.4 | 6.7 | 3.1 | 4.4 | 1.4 | 6.4 | 3.2 | 5.3 | 2.3 |
| 5.4 | 3.9 | 1.3 | 0.4 | 5.6 | 3.0 | 4.5 | 1.5 | 6.5 | 3.0 | 5.5 | 1.8 |
| 5.1 | 3.5 | 1.4 | 0.3 | 5.8 | 2.7 | 4.1 | 1.0 | 7.7 | 3.8 | 6.7 | 2.2 |
| 5.7 | 3.8 | 1.7 | 0.3 | 6.2 | 2.2 | 4.5 | 1.5 | 7.7 | 2.6 | 6.9 | 2.3 |
| 5.1 | 3.8 | 1.5 | 0.3 | 5.6 | 2.5 | 3.9 | 1.1 | 6.0 | 2.2 | 5.0 | 1.5 |
| 5.4 | 3.4 | 1.7 | 0.2 | 5.9 | 3.2 | 4.8 | 1.8 | 6.9 | 3.2 | 5.7 | 2.3 |
| 5.1 | 3.7 | 1.5 | 0.4 | 6.1 | 2.8 | 4.0 | 1.3 | 5.6 | 2.8 | 4.9 | 2.0 |
| 4.6 | 3.6 | 1.0 | 0.2 | 6.3 | 2.5 | 4.9 | 1.5 | 7.7 | 2.8 | 6.7 | 2.0 |
| 5.1 | 3.3 | 1.7 | 0.5 | 6.1 | 2.8 | 4.7 | 1.2 | 6.3 | 2.7 | 4.9 | 1.8 |
| 4.8 | 3.4 | 1.9 | 0.2 | 6.4 | 2.9 | 4.3 | 1.3 | 6.7 | 3.3 | 5.7 | 2.1 |
| 5.0 | 3.0 | 1.6 | 0.2 | 6.6 | 3.0 | 4.4 | 1.4 | 7.2 | 3.2 | 6.0 | 1.8 |
| 5.0 | 3.4 | 1.6 | 0.4 | 6.8 | 2.8 | 4.8 | 1.4 | 6.2 | 2.8 | 4.8 | 1.8 |
| 5.2 | 3.5 | 1.5 | 0.2 | 6.7 | 3.0 | 5.0 | 1.7 | 6.1 | 3.0 | 4.9 | 1.8 |
| 5.2 | 3.4 | 1.4 | 0.2 | 6.0 | 2.9 | 4.5 | 1.5 | 6.4 | 2.8 | 5.6 | 2.1 |
| 4.7 | 3.2 | 1.6 | 0.2 | 5.7 | 2.6 | 3.5 | 1.0 | 7.2 | 3.0 | 5.8 | 1.6 |
| 4.8 | 3.1 | 1.6 | 0.2 | 5.5 | 2.4 | 3.8 | 1.1 | 7.4 | 2.8 | 6.1 | 1.9 |
| 5.4 | 3.4 | 1.5 | 0.4 | 5.5 | 2.4 | 3.7 | 1.0 | 7.9 | 3.8 | 6.4 | 2.0 |
| 5.2 | 4.1 | 1.5 | 0.1 | 5.8 | 2.7 | 3.9 | 1.2 | 6.4 | 2.8 | 5.6 | 2.2 |
| 5.5 | 4.2 | 1.4 | 0.2 | 6.0 | 2.7 | 5.1 | 1.6 | 6.3 | 2.8 | 5.1 | 1.5 |
| 4.9 | 3.1 | 1.5 | 0.2 | 5.4 | 3.0 | 4.5 | 1.5 | 6.1 | 2.6 | 5.6 | 1.4 |
| 5.0 | 3.2 | 1.2 | 0.2 | 6.0 | 3.4 | 4.5 | 1.6 | 7.7 | 3.0 | 6.1 | 2.3 |
| 5.5 | 3.5 | 1.3 | 0.2 | 6.7 | 3.1 | 4.7 | 1.5 | 6.3 | 3.4 | 5.6 | 2.4 |
| 4.9 | 3.6 | 1.4 | 0.1 | 6.3 | 2.3 | 4.4 | 1.3 | 6.4 | 3.1 | 5.5 | 1.8 |
| 4.4 | 3.0 | 1.3 | 0.2 | 5.6 | 3.0 | 4.1 | 1.3 | 6.0 | 3.0 | 4.8 | 1.8 |
| 5.1 | 3.4 | 1.5 | 0.2 | 5.5 | 2.5 | 4.0 | 1.3 | 6.9 | 3.1 | 5.4 | 2.1 |
| 5.0 | 3.5 | 1.3 | 0.3 | 5.5 | 2.6 | 4.4 | 1.2 | 6.7 | 3.1 | 5.6 | 2.4 |
| 4.5 | 2.3 | 1.3 | 0.3 | 6.1 | 3.0 | 4.6 | 1.4 | 6.9 | 3.1 | 5.1 | 2.3 |
| 4.4 | 3.2 | 1.3 | 0.2 | 5.8 | 2.6 | 4.0 | 1.2 | 5.8 | 2.7 | 5.1 | 1.9 |
| 5.0 | 3.5 | 1.6 | 0.6 | 5.0 | 2.3 | 3.3 | 1.0 | 6.8 | 3.2 | 5.9 | 2.3 |
| 5.1 | 3.8 | 1.9 | 0.4 | 5.6 | 2.7 | 4.2 | 1.3 | 6.7 | 3.3 | 5.7 | 2.5 |
| 4.8 | 3.0 | 1.4 | 0.3 | 5.7 | 3.0 | 4.2 | 1.2 | 6.7 | 3.0 | 5.2 | 2.3 |
| 5.1 | 3.8 | 1.6 | 0.2 | 5.7 | 2.9 | 4.2 | 1.3 | 6.3 | 2.5 | 5.0 | 1.9 |
| 4.6 | 3.2 | 1.4 | 0.2 | 6.2 | 2.9 | 4.3 | 1.3 | 6.5 | 3.0 | 5.2 | 2.0 |
| 5.3 | 3.7 | 1.5 | 0.2 | 5.1 | 2.5 | 3.0 | 1.1 | 6.2 | 3.4 | 5.4 | 2.3 |
| 5.0 | 3.3 | 1.4 | 0.2 | 5.7 | 2.8 | 4.1 | 1.3 | 5.9 | 3.0 | 5.1 | 1.8 |

6.1.3. X_{30} veri kümesi

Tablo 6. 2: X_{30} veri kümesi

| Obje | Giriş1 | Giriş2 | Sınıf |
|------|--------|--------|-------|
| 1 | 1,5 | 2,5 | 1 |
| 2 | 1,7 | 2,6 | 1 |
| 3 | 1,2 | 2,2 | 1 |
| 4 | 2 | 2 | 1 |
| 5 | 1,7 | 2,1 | 1 |
| 6 | 1,3 | 2,5 | 1 |
| 7 | 2,1 | 2 | 1 |
| 8 | 2,3 | 1,9 | 1 |
| 9 | 2 | 2,5 | 1 |
| 10 | 1,9 | 1,9 | 1 |
| 11 | 5 | 6,2 | 2 |
| 12 | 5,5 | 6 | 2 |
| 13 | 4,9 | 5,9 | 2 |
| 14 | 5,3 | 6,3 | 2 |
| 15 | 4,9 | 6 | 2 |
| 16 | 5,8 | 6 | 2 |
| 17 | 5,5 | 5,9 | 2 |
| 18 | 5,2 | 6,1 | 2 |
| 19 | 6,2 | 6,2 | 2 |
| 20 | 5,6 | 6,1 | 2 |
| 21 | 10,1 | 12,5 | 3 |
| 22 | 11,2 | 11,5 | 3 |
| 23 | 10,5 | 10,9 | 3 |
| 24 | 12,2 | 12,3 | 3 |
| 25 | 10,5 | 11,5 | 3 |
| 26 | 11 | 14 | 3 |
| 27 | 12,2 | 12,2 | 3 |
| 28 | 10,2 | 10,9 | 3 |
| 29 | 11,9 | 12,7 | 3 |
| 30 | 12,9 | 12 | 3 |

Tablo 6.2'de sunulan X_{30} veri kümesi de yine bulanık c-ortalama, k-ortalama kümeleme algoritmaları ile kural tabanlı çıkarım sisteminin test edilmesinde kullanılmıştır.

Şekil 6.6'da veri kümesine ait diyagram sunulmuştur. Görüldüğü gibi veride bulunan üç sınıf da net bir şekilde birbirinden ayrılmaktadır.

6.2. Bulanık Çıkarım Sistemiyle Sınıflandırıcı Tasarımı

Bulanık mantık son zamanlarda geniş bir kullanım alanı bulmaktadır. Yeni bir teori olmasına karşın oldukça geniş bir kullanım alanı bulunmaktadır. Örnek olarak süreç kontrol sistemleri, yönetim ve karar verme, örüntü tanıma ve sınıflandırma verilebilir. Klasik yöntemlerdeki "0" ve "1" in yeterli olmadığı birçok durumda Prof. Lotfi A. Zadeh (1965) tarafından sunulan üyelik derecesi yöntemi problem çözümünde yeni bir olanak sunmaktadır. Bu yöntemde klasik kümeler yerine, üyeleri "üyelik derecesi"ne sahip bulanık kümeler bulunmaktadır. Bulanık kümedeki elemanlar tam "1" veya sıfır "0" üyelik derecesine sahip olabileceği gibi "0" ile "1" arasında herhangi bir değerde üyelik derecesine de sahip olabilmektedir. Bulanık kümedeki elemanların üyelik derecelerini tanımlayan matematiksel fonksiyona "üyelik fonksiyonu" denilmektedir. Problemlerin, sayısal değerler arasındaki ilişkilerle tanımlanması yerine dilsel değişkenler kullanılarak sözselsel bir yapıda tanımlanması bulanık mantığın en büyük avantajıdır (Nedeljkovic, 2004).

Bu bölümde Sugeno ve Mamdani bulanık çıkarım sistemleriyle (FIS- Fuzzy Inference System) sınıflandırıcı tasarımının nasıl uygulandığı incelenecektir. Önceki bölümlerde de anlatıldığı gibi bulanık sistemde kural çıkarımı için farklı yöntemler kullanılabilir. Bu yöntemlere örnek olarak karar ağaçları ve kümeleme algoritmaları verilebilir.

Bu uygulamada bulanık sistem için üyelik fonksiyonları oluşturulurken kümeleme algoritması kullanılmış ve oluşturulan üyelik fonksiyonları ile bulanık kurallar oluşturulmuştur. Kümeleme algoritması olarak bulanık c-ortalama (fuzzy c-means - fcm) algoritması kullanılmıştır.

Basit ve anlaşılır olması açısından uygulamada Bezdek ve diğ. (2005) tarafından da kullanılan Tablo 6.2'de sunulan veri kümesi kullanılmıştır. Üyelik fonksiyonları oluşturulurken öncelikle bu veri üzerinde bulanık c-ortalama algoritması, giriş verisi için üç adet üyelik fonksiyonu oluşturulacağı için, üç küme bulunacak şekilde (c=3 parametresiyle) çalıştırılmış ve veri kümesindeki küme merkezleri bulunmuştur.

Üyelik fonksiyonları gauss eğrisiyle oluşturulmuş ve eğrinin merkezi olarak da c-ortalama algoritmasıyla bulunan küme merkezleri girilmiştir. Gauss eğrisi her kümedeki objeleri kapsayacak şekilde genişletilmiştir. Her iki giriş için de oluşturulan gauss üyelik fonksiyonları Şekil 6.7'de gösterilmektedir.

Birinci giriş için bulanık c-ortalama algoritmasıyla bulunan küme merkezleri 1.77, 5.39 ve 11.27'dir. Şekil 6.7a'da da gauss eğrisinde merkez noktalar bu değerlere karşılık gelmektedir. Merkezden uzaklaştıkça o küme için verilen üyelik derecesi de azalmaktadır. Örneğin 5.8 (veri kümesinde 16'ncı nokta) noktası yüksek miktarda orta kümesine çok düşük oranlarda ise düşük ve yüksek kümelerine aittir.

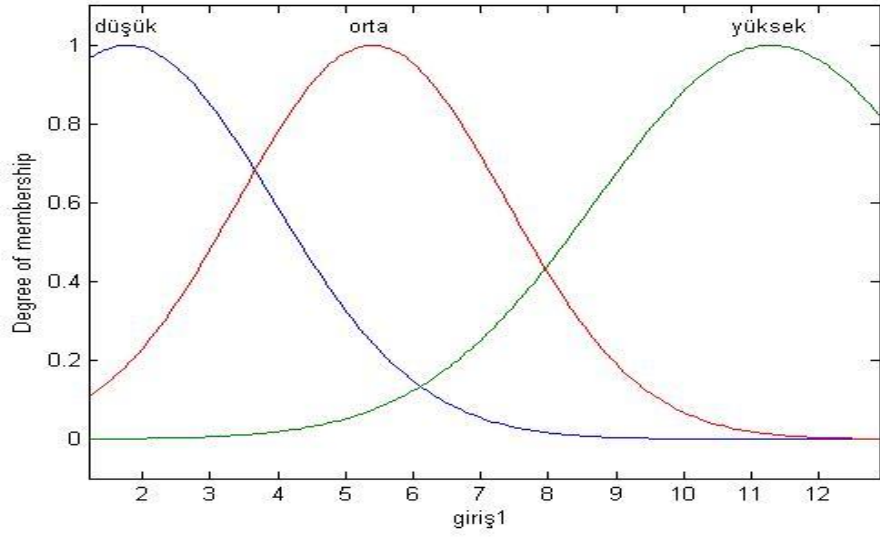
Şekil 6.7b'de ise ikinci giriş için üyelik fonksiyonları görülmektedir. İkinci giriş için bulanık c-ortalama algoritmasıyla bulunan küme merkezleri ise 2.22, 6.07 ve 12.05'tir. Bu girişte 16'ncı noktaya baktığımızda 6 değerine sahip olduğu bu değer ise yine yüksek oranda orta kümesine ait olduğu görülmektedir.

Şekil 6.8'de MA bulanık çıkarım sisteminin çıkış üyelik fonksiyonlarına (sınıflara) ait eğriler görülmektedir. Giriş üyelik fonksiyonlarından elde edilen sonuçlara göre objenin çıkışta hangi sınıfa ne oranda ait olduğu hesaplanmaktadır. Şekil 6.9'da örnek olarak seçtiğimiz 16'ncı objenin (5.8, 6) çıkış üyelik değeri hesaplanması gösterilmektedir.

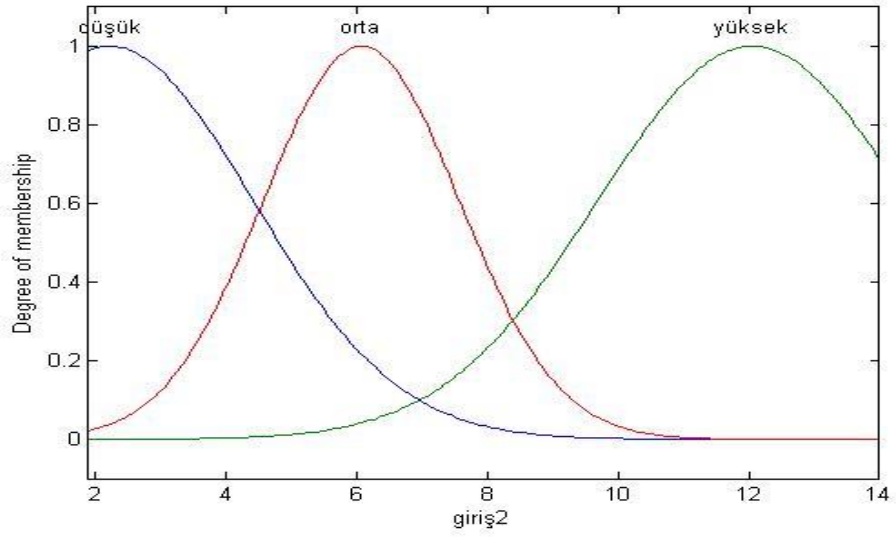
Şekil 6.9'da 5.8 ve 6 giriş değerleri için çıkış üyelik derecesinin 1.92 olarak hesaplandığı görülmektedir. Burada durulaştırma (defuzzification) işlemi yapıldığında 1.92 değerinin ikinci sınıfa (sınıf2) ait olduğu sonucuna varılacaktır.

Tablo 6.3'te örnek giriş verisi ile oluşturulan sugeno bulanık çıkarım sisteminin yine aynı veri ile yapılan test sonuçları görülmektedir. Örnek veride ilk on obje birinci sınıfa, ikinci on obje ikinci sınıfa ve üçüncü on obje ise üçüncü sınıfa aittir. Sisteme uygulanan test sonucunda yine aynı sınıflandırmaların elde edildiği görülmektedir.

Bu uygulamada yapılan, sınıfları belli 30 adet veri noktası kullanılarak sınıflandırıcı oluşturulmasıdır. Bu veri noktaları sistemin eğitilmesi için kullanılmıştır. Veri noktaları birbirlerinden kolayca ayrıştırılabilir olduğundan yapılan testlerde yüzde yüz başarı sağlanmıştır.

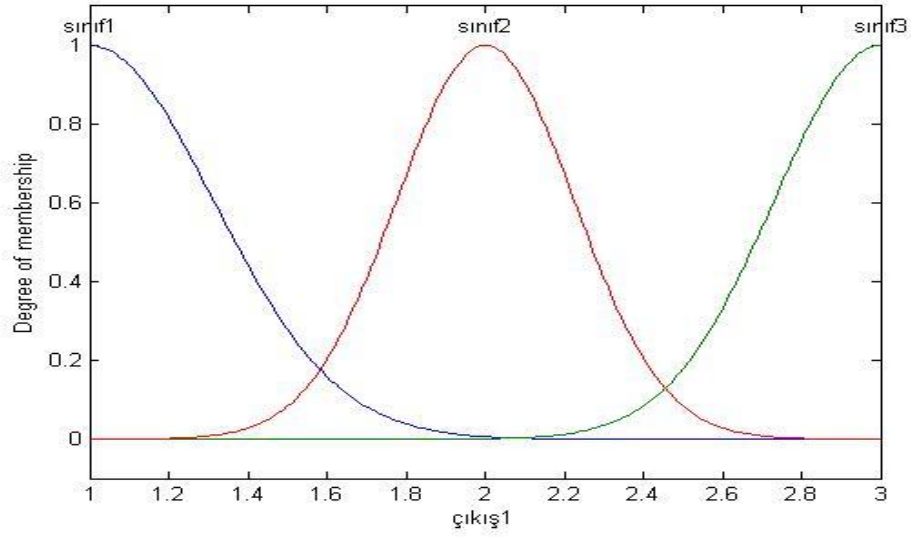


a) Birinci giriş için üyelik fonksiyonları

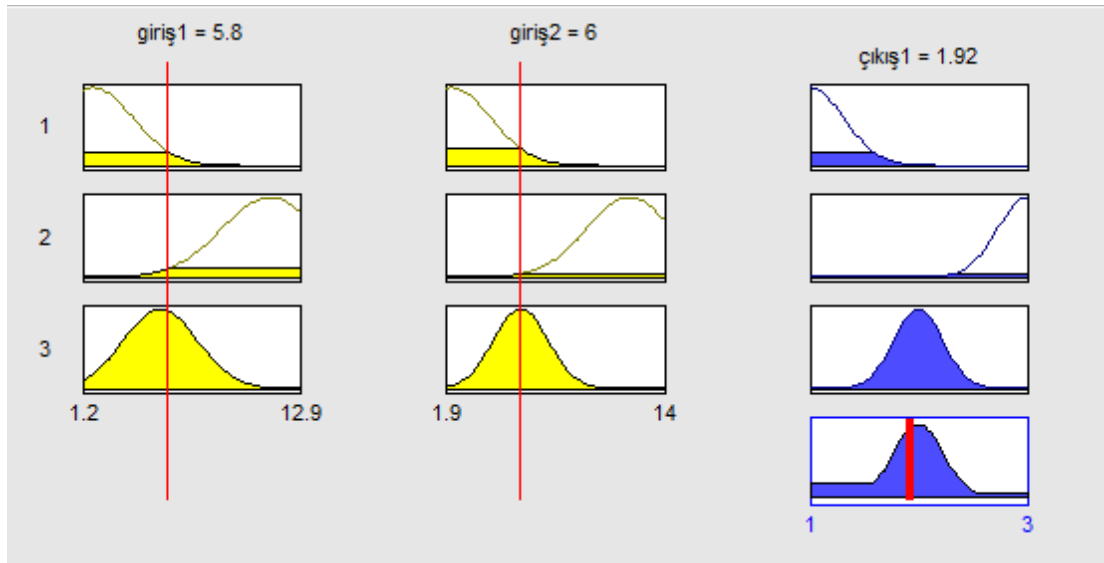


b) İkinci giriş için üyelik fonksiyonları

Şekil 6. 7: Giriş vektörleri için gauss üyelik fonksiyonları



Şekil 6. 8: MA sistem için çıkış üyelik fonksiyonu



Şekil 6. 9: Çıkış üyelik değeri hesaplanması

Tablo 6. 3: Örnek giriş verisi ve bulanık sistem ile alınan sonuçlar

| Obje | Giriş1 | Giriş2 | Çıkış |
|------|--------|--------|----------|
| 1 | 1,5 | 2,5 | 1,008766 |
| 2 | 1,7 | 2,6 | 1,012365 |
| 3 | 1,2 | 2,2 | 1,004028 |
| 4 | 2 | 2 | 1,005966 |
| 5 | 1,7 | 2,1 | 1,005398 |
| 6 | 1,3 | 2,5 | 1,007283 |
| 7 | 2,1 | 2 | 1,006529 |
| 8 | 2,3 | 1,9 | 1,006554 |
| 9 | 2 | 2,5 | 1,013815 |
| 10 | 1,9 | 1,9 | 1,00457 |
| 11 | 5 | 6,2 | 1,94088 |
| 12 | 5,5 | 6 | 1,953651 |
| 13 | 4,9 | 5,9 | 1,918818 |
| 14 | 5,3 | 6,3 | 1,957682 |
| 15 | 4,9 | 6 | 1,925043 |
| 16 | 5,8 | 6 | 1,964356 |
| 17 | 5,5 | 5,9 | 1,949331 |
| 18 | 5,2 | 6,1 | 1,94549 |
| 19 | 6,2 | 6,2 | 1,981853 |
| 20 | 5,6 | 6,1 | 1,961192 |
| 21 | 10,1 | 12,5 | 2,999993 |
| 22 | 11,2 | 11,5 | 2,999979 |
| 23 | 10,5 | 10,9 | 2,99975 |
| 24 | 12,2 | 12,3 | 2,999999 |
| 25 | 10,5 | 11,5 | 2,999941 |
| 26 | 11 | 14 | 3 |
| 27 | 12,2 | 12,2 | 2,999999 |
| 28 | 10,2 | 10,9 | 2,999619 |
| 29 | 11,9 | 12,7 | 3 |
| 30 | 12,9 | 12 | 3 |

6.3. Bulanık C-Ortalama ve K-Ortalama Algoritmalarının Iris ve X₃₀ Veri Kümelerine Uygulanması

K-ortalama ve bulanık c-ortalama algoritmalarının veri kümelerine uygulanması ile elde edilen sonuçlar EK-A ve EK-B'de gösterilmiştir.

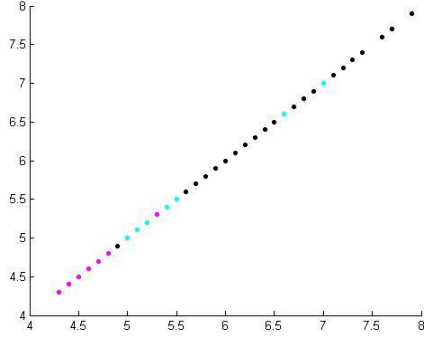
Algoritmaların X_{30} verisine uygulanması ile elde edilen sonuçlarda yüzde yüz başarı sağlanması oldukça normal bir durumdur. Bu veri kümesinde bulunan üç küme de birbirinden doğrusal bir çizgi ile ayrılabilmekte ve verideki kümelerin boyutları eşit seviyede bulunmaktadır. Dolayısıyla kümeleme algoritmalarının bu tür bir veride başarısız olma ihtimali yok denecek kadar azdır.

Başarısızlık ihtimalinin her ne kadar az olduğunu söylesek de burada bir hususun üzerinde durmak faydalı olacaktır. Kümeleme algoritmalarında başlangıç değerleri algoritmanın başarısına etki etmektedir. Başlangıç değerleri rastgele seçilebildiği gibi her kümeden bir örnek içerecek şekilde önceden değerlerin girilmesiyle de oluşturulabilmektedir. Giriş parametrelerinin rastgele belirlendiği bir durumda sonuçların istenen seviyede olmaması ihtimalini de göz ardı etmemek gerekmektedir. Yapılan denemelerin bazılarında giriş değerleri rastgele oluşturulduğunda yanlış gruplandırma sonuçları elde edilmiştir. Bunun nedeni algoritmanın başlangıçta rastgele seçimde sadece bir kümeye ait objeleri alması ve bu değerlerle elde ettiği küme merkezlerinin gerçek merkezlerden çok uzak olması ve dolayısıyla algoritmanın istenen değerlere ulaşamamasıdır. Özellikle küçük boyutlu verilerde bu duruma rastlamak hiç de sürpriz değildir.

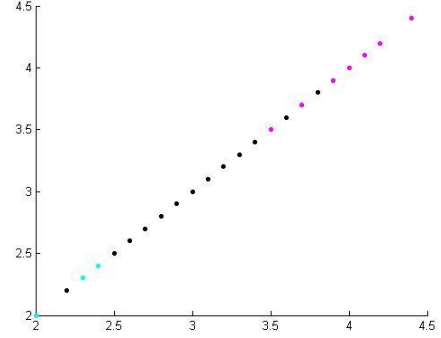
Algoritmalar iris veri kümesine uygulandığında her iki algoritmada veride bulunan 150(yüz elli) noktanın 16(on altı) tanesini yanlış kümelenmiştir. Veri kümesinde bulunan "sestosa" sınıfına ait ilk 50 nokta diğerlerinde doğrusal bir şekilde ayrılabilirdi için yüzde yüz başarıyla doğru kümelenmiştir. Yanlış kümelenmelere diğer iki sınıftan noktalarda rastlanmıştır. Yapılan denemelerde k-ortalama ve bulanık c-ortalama algoritmaları ile aynı sonuçların elde edilmesine karşın bulanık algoritmaların objelerin hangi oranda hangi kümeye ait olduğu bilgisini sakladığı için çok daha avantajlı olduğu gözlemlenmiştir. Bulanık c-ortalama algoritmasıyla elde edilen üyelik dereceleri en büyük üyelik derecesine sahip olanın kabul edilmesi yöntemiyle durulaştırıldığında k-ortalama algoritmasıyla aynı sonuçları vermektedir. Ancak durulaştırma aşamasında farklı yöntemler uygulayarak daha iyi sonuçların elde edilebileceği düşünülmektedir. En azından son karar aşamasına kadar objelerin kümelere ne oranda ait olduğu bilgisine sahip olmanın büyük bir avantaj olduğu sonucuna varılmıştır.

6.4. Iris Verisi ile Bulanık Sistem Tabanlı Sınıflandırma

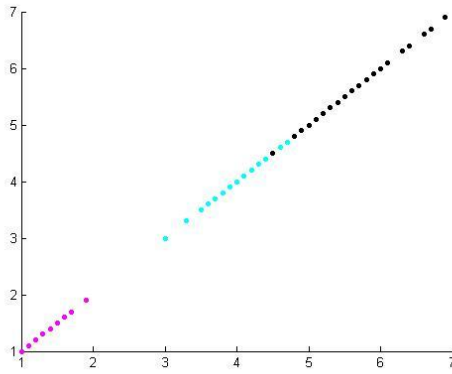
Mor (magenta) Iris Sestosa (1-50), açık mavi (cyan) Iris Versicolor (51-100) ve siyah Iris Virginica (101-150) sınıflarını göstermektedir.



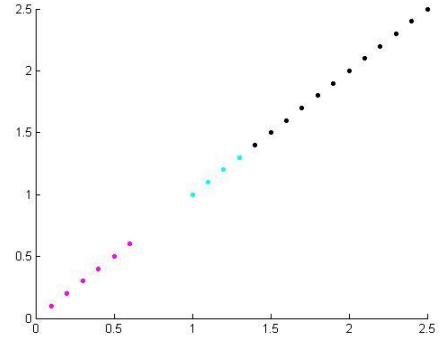
a) Sepal Uzunluk



b) Sepal Genişlik



c) Petal Uzunluk

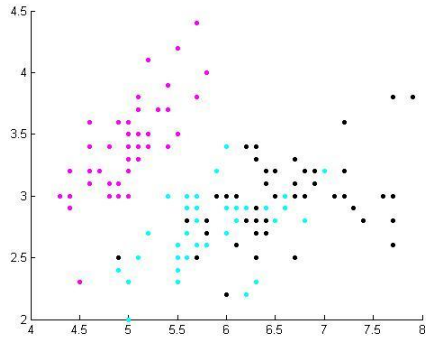


d) Petal Genişlik

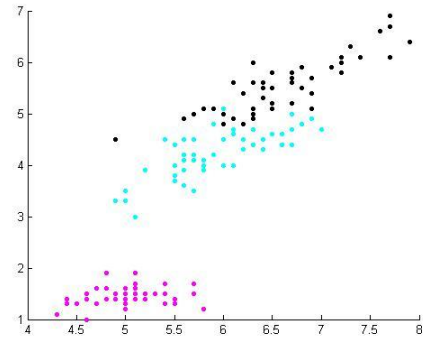
Şekil 6. 10: Iris verisinde özellik vektörlerinin diyagramları

Şekil 6.10'da görüldüğü gibi Petal Uzunluk ve Petal Genişlik özelliklerinde Iris Sestosa sınıfı (mor renkli) rahat bir şekilde ayrıştırılabilmektedir. Iris Versicolor (siyah) ve Iris Virginica (açık mavi) sınıflarının ise dört özellikte de iç içe girmiş olduğu ya da çok yakın bir sınıra sahip oldukları görülmektedir.

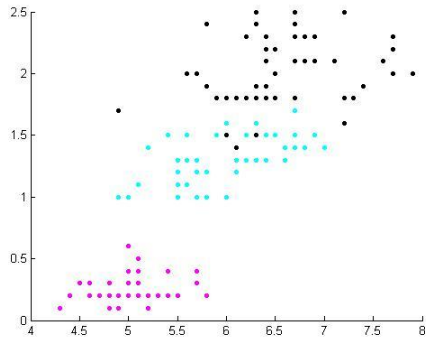
Özellikler Şekil 6.11'deki gibi ikişerli karşılıklı olarak incelendiğinde yine Iris Sestosa sınıfının (mor renkli) rahat bir şekilde ayrıştırılabildiği, Iris Versicolor (siyah) ve Iris Virginica (açık mavi) sınıflarının ise iç içe girmiş olduğu ya da çok yakın bir sınıra sahip oldukları görülmektedir.



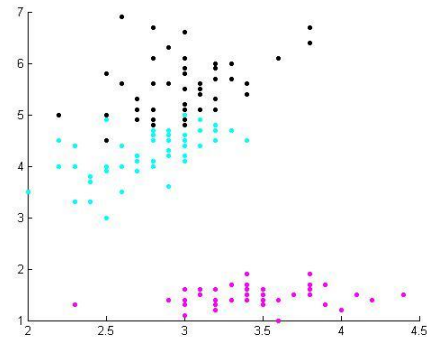
Sepal Uzunluk & Sepal Genişlik



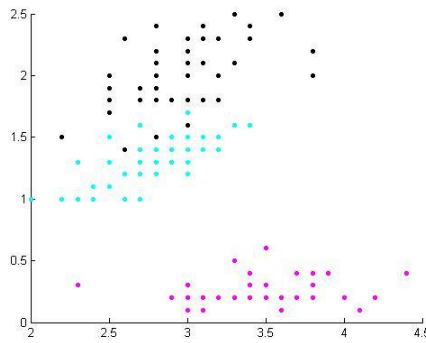
Sepal Uzunluk & Petal Genişlik



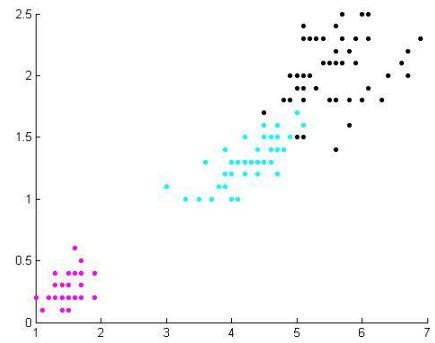
Sepal Uzunluk & Petal Genişlik



Sepal Genişlik & Petal Uzunluk



Sepal Genişlik & Petal Genişlik



Petal Uzunluk & Petal Genişlik

Şekil 6. 11: Iris verisinde özellik vektörlerinin karşılıklı diyagramları

Uygulama Mamdani bulanık modeli ile gerçekleştirilmiştir. Bulanık c-ortalama (FCM) algoritmasıyla her giriş vektörü için küme merkezleri bulunmuş ve bulunan bu küme merkezleri üçgen üyelik fonksiyonlarının oluşturulmasında kullanılmıştır. Bulanık c-ortalama algoritmasıyla bulunan küme merkez değerleri üçgen üyelik fonksiyonlarının merkez noktasını oluşturmuştur.

Üyelik fonksiyonlarının oluşturulmasından sonra özellik vektörlerinin histogramları incelenerek kural tablosu oluşturulmuştur. Örneğin histogramlardan açık bir şekilde görüldüğü gibi Petal Uzunluk ve Petal Genişlik özelliklerinin düşük değerli olduğu bölgelerde obje sestosa sınıfında bulunmaktadır. O halde şu kuralı yazmak gayet mantıklıdır:

Eğer (Petal Uzunluk kısa ise) veya (Petal Genişlik dar ise) o zaman (output1 sestosa sınıfıdır) (1)

Buradaki (1) sayısı kuralın ağırlığını göstermektedir. Varsayılan değer olarak 1 almakta olup değer aralığı [0-1] arasında değişmektedir.

Yazılan diğer kurallar ise şöyledir;

Eğer (Sepal Uzunluk orta ise) ve (Petal Uzunluk orta ise) ve (Petal Genişlik orta ise) o zaman (output1 versicolor sınıfıdır) (1)

Eğer (Petal Uzunluk uzun ise) ve (Petal Genişlik geniş ise) o zaman (output1 virginica sınıfıdır) (1)

Yazılan bu üç kuralla yapılan sınıflandırmada 150 örnek içinden 146 doğru, 4 yanlış sınıflandırma sonucu elde edilmiştir. Elde edilen sonuçlar EK-C'de gösterilmiştir.

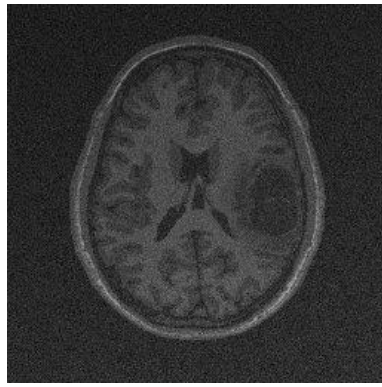
Yapılan eğitilmiş sınıflandırmada elde edilen deneyimler kısaca şöyle özetlenebilir. Öncelikle sınıfların ayırt edilmesini sağlayacak en uygun özellikler belirlenmeli, belirlenen özellikler için eğitim amacıyla verilen objelere ait özellik vektörlerinde küme merkezleri bulunmalı ve bu merkezler ve örneklerin dağılımı ile orantılı olarak üyelik fonksiyonları oluşturulmalıdır.

6.5. Bulanık Sistem Tabanlı Tümör Tespiti

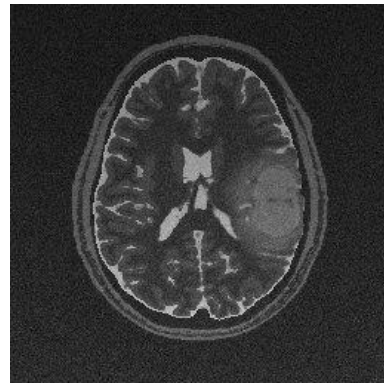
Önceki bölümlerde anlatılan hususlar doğrultusunda MR görüntüsünün T1 ve T2 bantları kullanılarak tümörlü bölgenin tespit edilmesine çalışılmıştır. Bu işlem için Mamdani bulanık sistemi kullanılmıştır. Elde edilen sonuçların karşılaştırılabilmesi için MR görüntüsü ile birlikte indirilen tümörlü bölgeyi gösteren görüntü kullanılmıştır.

Tümörlü bölgenin sınıflandırılması için T1 ve T2 bantlarının piksel değerleri, beyin MR görüntüsünün simetrik olması ve T2 bandından çıkarılan görüntüdeki objelerin yatay ve dikey uzunlukları ile alan bilgisi kullanılmıştır. Yapılan işlem detayları ilerleyen paragraflarda açıklanmıştır.

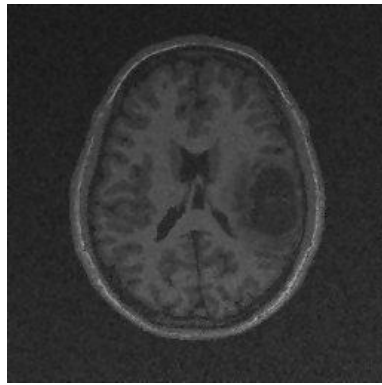
Öncelikle görüntüde bulunan gürültünün giderilebilmesi maksadıyla T1 ve T2 görüntülerine Wiener alçak geçiren filtresi uygulanmıştır. Elde edilen görüntüler orijinal halleriyle birlikte Şekil 6.12a-d'de gösterilmiştir.



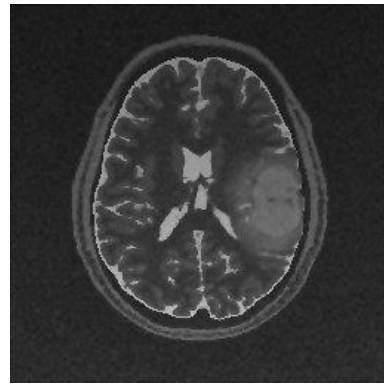
a) T1 bandı



b) T2 bandı



c) Filtre uygulanmış T1 bandı



d) Filtre uygulanmış T2 bandı

Şekil 6. 12: Wiener filtresi uygulanması. Birinci sütun kaynak görüntüleri, ikinci sütun ise wiener filtresi uygulanması sonucu elde edilen görüntüleri göstermektedir.

T2 bandı incelendiğinde tümörlü bölgenin belirli bir piksel değeri aralığında olduğu, tümörlü bölgeye ait olmayan piksel aralığının büyük oranda arka plan verisine ait olduğu görülmüştür. Bu nedenle bulunacak efektif bir eşik değeri ile görüntüde istenmeyen bölgelerin atılabileceği ve tümör tespitinde karmaşıklığın azaltılabileceği

düşünülmüştür. Bu düşünce bulanık c-ortalama algoritmasından faydalanılarak gerçekleştirilmiştir.

Bölütleme işlemi yapılırken, MR görüntüsü bulanık c-ortalama algoritmasıyla yedi bölgeye bölünmüştür. Yedi küme için elde edilen veri kümeleri Şekil 6.13'te gösterilmektedir.

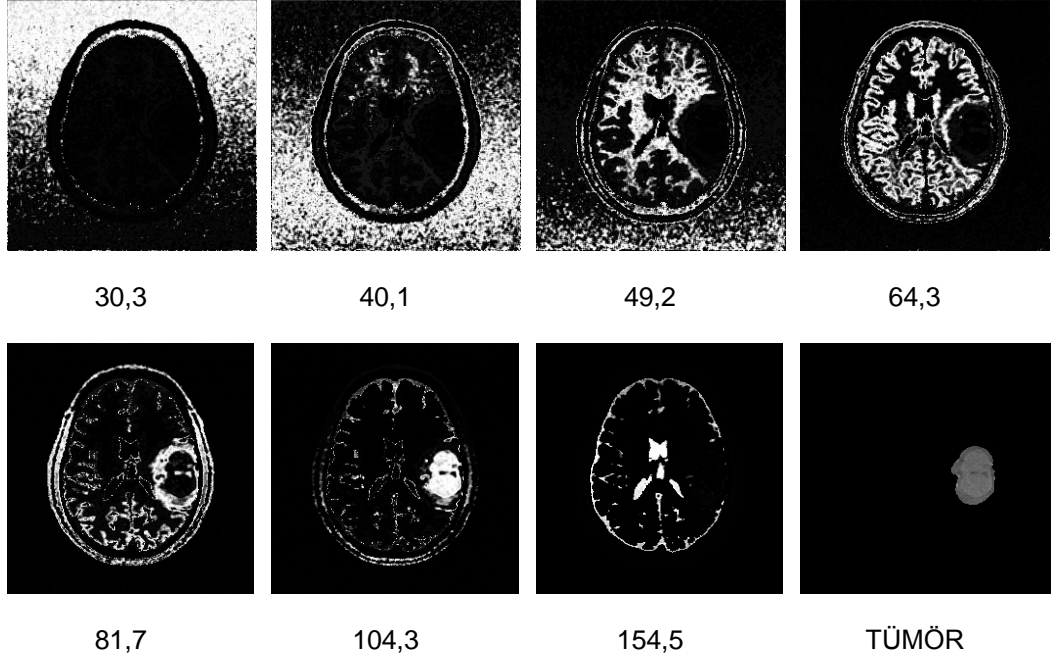
Şekil 6.13'te de görüldüğü gibi bulanık c-ortalama algoritmasıyla elde edilen ilk dört küme tümörlü bölgenin dışında bulunmaktadır. Bu nedenle beşinci küme merkezi eşik değeri olarak belirlenerek bu değer altında kalan piksellere "0" değeri verilmektedir. Bu sayede oluşturulan yeni görüntü Şekil 6.14.a'da gösterilmektedir.

Şekil 6.14.a'da görüldüğü gibi tümörlü bölge hala görüntüde bulunmakta, buna karşılık gereksiz kısımlar görüntüden kaldırılmaktadır.

Elde edilen bu görüntüden dört özellik çıkarılmıştır. Bu özellikler tümör olasılık hesabı, alan bilgisi ile genişlik ve uzunluk bilgileridir.

Beyin görüntüsü dikey bir çizgiyle ortadan bölündüğünde sağ ve sol kısımların simetrik olduğu görülmektedir. Buradan hareketle eğer sağ ve sol kısım simetrik değilse bu durum simetriyi bozan bir etken olduğunu göstermektedir. Böylece simetrinin bozulduğu kısımda anormal bir durum olduğu ve bu kısımda tümör olma ihtimalinin yüksek olduğu sonucu çıkacaktır. Piksel değerlerinden tümörlü bölgeyi ayırmak mümkün olmadığından bu özellik bazı uygulamalarda kullanılmıştır (Gordillo ve diğ., 2010).

Bu uygulamada da beyin görüntüsü dikey ve yatay çizgilerle dörde bölünmüştür. Şekil 6.14a'da bulunan T2 görüntüsüne bakıldığında dörde bölünecek görüntüde simetrik bölgelerde piksel değeri sıfırdan büyük olan piksel sayılarının ortalama olarak birbirine yakın olması beklenecektir. Piksel sayılarında anormal bir fark varsa bu bölgede tümör olma ihtimali yüksektir. Buradan hareketle her piksel için tümörlü bölgede olma olasılığı hesaplanmıştır. Tümör olasılık görüntüsü Şekil 6.14.b'de gösterilmektedir. Dikkat edilirse tümörün büyük oranda bulunduğu sağ alt kısımda görüntü en parlak, sağ üstte daha az parlak ve sol tarafta ise mattır. Yani sol tarafta tümör bulunma olasılığı yok denecek kadar az, sağ tarafta ise yüksektir.

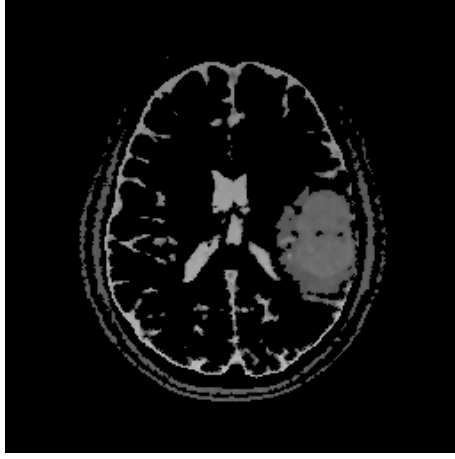


Şekil 6.13: Bulanık c-ortalama kümeleme algoritmasıyla elde edilen yedi kümeye ait görüntüler. Görüntülerin altında küme merkezi değeri de verilmiştir. Sağ alttaki son görüntü tümörlü bölgeyi göstermektedir.

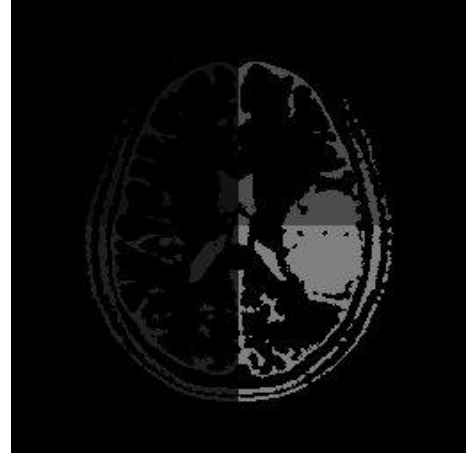
Alan özelliğinin çıkarılmasının amacı nokta ve benzeri küçük parçaların sonuca karışmasını önlemektir. Boyut özelliği ise görüntüde dış kısımda bulunan ince uzun tabakanın atılması için çıkarılmıştır. Aslında bazı uygulamalarda bu kısımlar ön işlem aşamasında atılmaktadır. Literatürde dış kısmın atılmasına (skull-stripping) yönelik farklı yöntemler kullanılmaktadır (Gordillo ve diğ., 2010).

Alan ve boyut özellikleri hesaplanırken Şekil 6.14.a'da bulunan bağımsız objeler dört komşuluğuna (4-connected) bakılarak belirlenmiştir. Bu objelerin alan değerleri, yatay ve dikeydeki uzunlukları hesaplanarak objede bulunan piksellere yazılmış ve böylece üç özellik matrisi (alan, en ve boy özellik matrisleri) elde edilmiştir. T2 görüntüsünde bulunan bağımsız objeler Şekil 6.14c ve d'de gösterilmektedir.

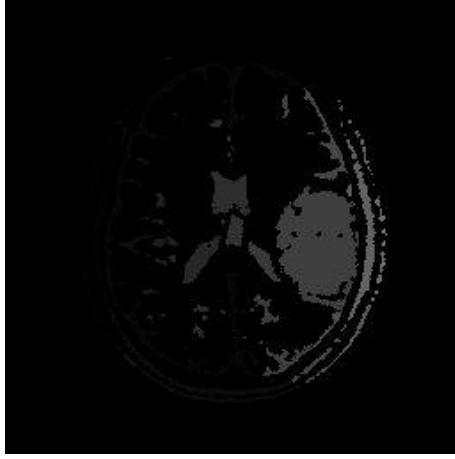
Şekil 6.14d'den de görüldüğü gibi tümörlü bölge diğer bölgelerden ayrılmıştır. Dolayısıyla bu bölgede hesaplanacak alan ve boyut bilgileri diğer bölgelere göre farklılık göstereceğinden tümör tespitinde ek bir fayda sağlayacaktır.



a) Bulanık c-ortalama algoritmasıyla elde edilen eşik değeri ile bölütlenmiş görüntü.



b) Tümör bulunma olasılığı.



c) T2 görüntüsünde dört komşuluğa göre bulunan bağımsız objeler (gri seviyeli).



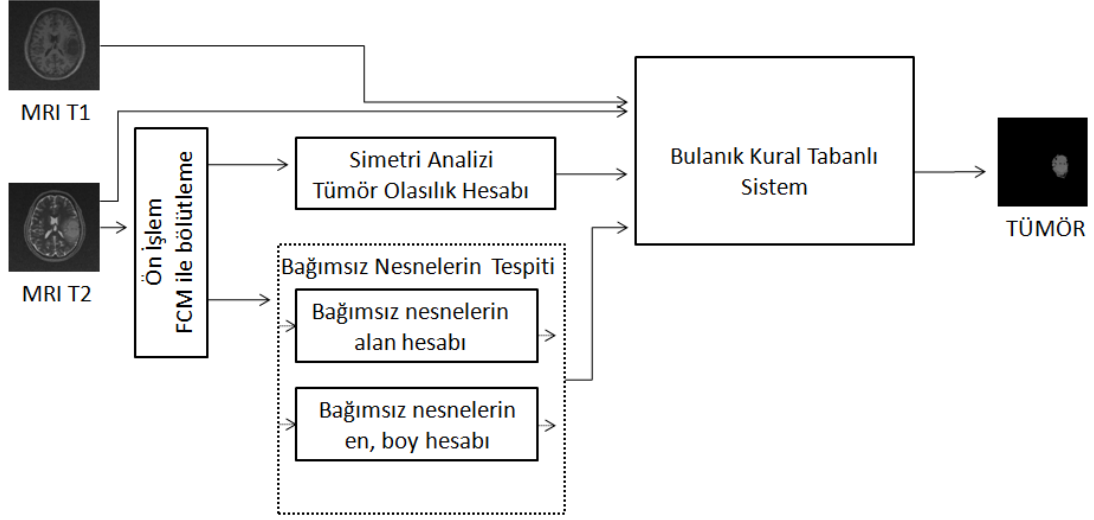
d) T2 görüntüsünde dört komşuluğa göre bulunan bağımsız objeler (RGB).

Şekil 6. 14: T2 MR görüntüleri

Bulanık kural tabanlı sistem tasarımında bulanık sisteme T1 ve T2 MR görüntüleri, simetri özelliği kullanılarak çıkarılan tümör olasılık matrisi ve T2 görüntüsünde bulunan bağımsız nesnelerin alan ve boyut bilgileri giriş olarak verilmektedir. Oluşturulan sistem tasarımı Şekil 6.15'te gösterilmektedir.

Bulanık kural tabanlı sistemde öncelikle altı özellik matrisi (T1, T2, tümör olasılık, alan, uzunluk ve genişlik) için üyelik fonksiyonları oluşturulmuştur. Üyelik fonksiyonları oluşturulurken özellik matrisinde tümörlü bölgenin minimum ve maksimum değerleri, ortalama değeri ve nasıl bir dağılım gösterdiği önem arz

etmektedir. Üyelik fonksiyonları sistemin başarısında önemli bir rol oynamaktadır. Uygulamada oluşturulan üyelik fonksiyonları Şekil 6.16'da gösterilmektedir.



Şekil 6. 15: Bulanık kural tabanlı sistem ile tümör tespiti

Üyelik fonksiyonları oluşturulduktan sonra sistem çıkışında tümörlü bölgeyi ayırt edebilecek bulanık kurallar oluşturulmuştur.

Birinci kural:

T1 ile T2 piksel değerleri tümörlü bölge olarak tanımlı üyelik fonksiyonu içinde yer almalıdır. Piksel değeri üyelik fonksiyonu merkezine ne kadar yakınsa tümör olma olasılığı o kadar yüksektir.

Bu kuralda öncelikle tümörlü bölgenin piksel değerine yakın olmayan pikseller görüntüden ayrılmaktadır. Ancak tümörlü bölgeye ait olmayıp benzer piksel değerlerine sahip olan pikseller henüz ayırt edilmemiştir.

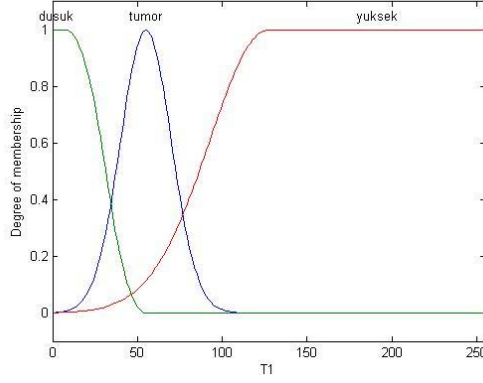
İkinci kural:

Tümör olasılık değeri düşükse tümör olma olasılığı çok düşük, yüksekse tümör olma olasılığı da çok yüksektir.

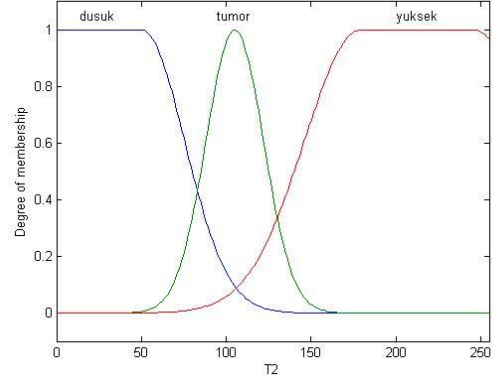
Bu kural ise dört bölgeye ayrılan beyin MR görüntüsünde hangi bölgede tümör olasılığı yüksekse o bölgeyi öne çıkarmakta diğer bölgeleri ayırmaktadır.

Üçüncü kural:

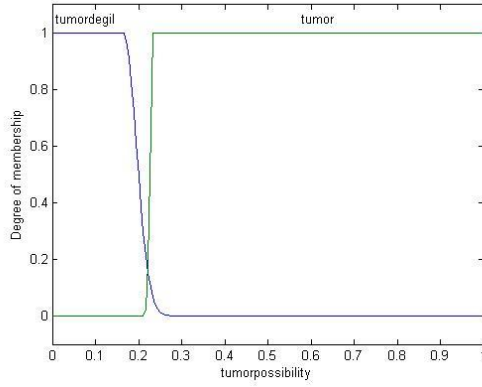
Küçük alana sahip olan pikseller nokta ve benzeri parçalara ait olabilir ancak tümörlü bölgeye ait olma olasılıkları düşüktür.



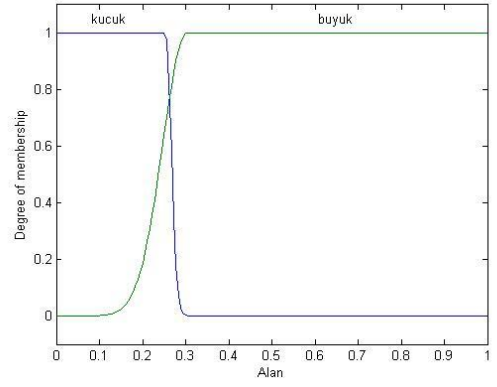
a) T1 MR görüntüsü



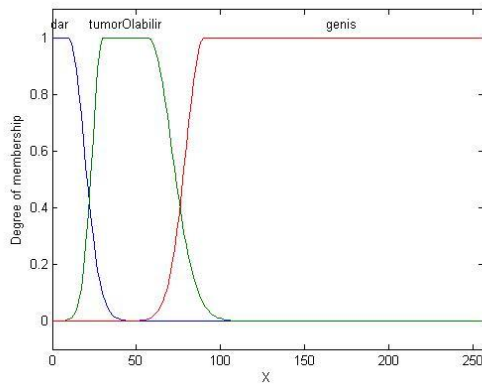
b) T2 MR görüntüsü



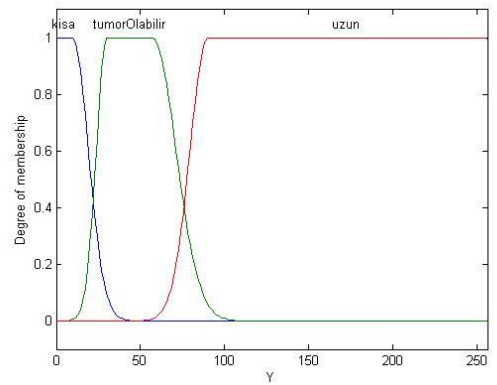
c) Tümör olasılık değeri



d) Bağımsız nesnelerin alan bilgisi



e) Bağımsız nesnelerin genişlik bilgisi



f) Bağımsız nesnelerin uzunluk bilgisi

Şekil 6.16: Üyelik fonksiyonları

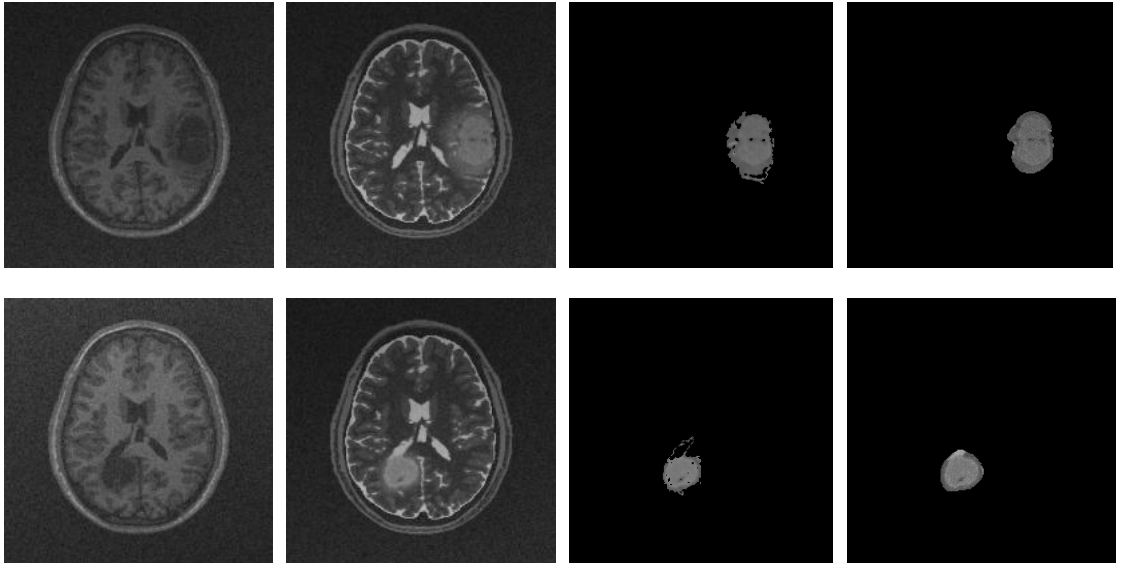
Bu kural ile görüntüde ana objelere ait olmayan bağımsız küçük bölgeler ayrıştırılmaktadır.

Dördüncü kural:

Şekilsel olarak ince uzun veya kısa ama geniş olan nesnelerin tümör olma olasılığı düşüktür.

Beyin MR görüntüsünde piksel değeri olarak tümörlü bölgeye benzeyen ancak şekilsel olarak ayırt edilebilen bölgeler bulunmaktadır. Özellikle görüntünün dış kısmını çevreleyen kısımlar buna örnek olarak verilebilir. Bu kısımları ayırt etmek için ön işlem aşamasında çeşitli yöntemlerin kullanıldığına alan ve boyut özelliği kısmında değinilmiştir. Oluşturulan kuralla ön işleme algoritma ile bu kısımların çıkarılması yerine şekilsel özellikler ile çıkarma yoluna gidilmiştir.

Kural tabanlı sistem ile elde edilen sonuçlar ise Şekil 6.17’de verilmiştir.



Şekil 6. 17: Bulanık kural tabanlı sistem ile elde edilen sonuçlar. Sırasıyla birinci sütun kaynak T1 görüntüsü, ikinci sütun kaynak T2 görüntüsü, üçüncü sütun elde edilen tümör, dördüncü sütun ise gerçek tümör görüntüsüdür.

7. SONUÇLAR VE ÖNERİLER

Bu çalışmada beyin MR görüntüsünde tümörlü bölgelerin tespit edilmesi için bulanık mantığın getirdiği avantajlardan faydalanılmıştır.

Öncelikle kümeleme algoritmaları kullanılarak MR görüntüsünde bulunan farklı küme grupları tespit edilerek tümörlü bölgenin ayırt edilmesine çalışılmıştır. Bu yöntemde görüntü verisi doğal gruplarına ayrıştırılmıştır. Ancak bu yöntemlerin görüntüde bulunan kümeler doğrusal bir çizgiyle ayrıştırılabilir nitelikte değilse yani iç içe geçmiş durumdaysa veya kümeler arasında ayırt edilmesi zor az miktarda farklılıklar bulunuyorsa hatalı sonuçlar üretmesi kaçınılmazdır. Yine bu algoritmalarda doğal bir gruplandırma yapıldığı için çok küçük boyutlu kümelerin fark edilememesi sonuçları oluşabilmektedir. Yapılan denemelerde bulanık c-ortalama algoritmasının (veya klasik k-ortalama) 40(kırk) objeli bir veride 3(üç) objeli kümeyi ayırt edemediği görülmüştür. Denemeler sonucunda elde edilen bir başka sonuç ise kümeleme algoritmalarında başlangıç değerlerinin sonuca önemli etkiler yaptığıdır. Bu nedenle bu algoritmalara en uygun başlangıç değerlerinin verilmesi önem arz etmektedir.

Yukarıda bahsedilen iki husus bize kümeleme algoritmalarında bir eğitim safhasının algoritmanın doğruluk oranını artıracaklarını göstermektedir. Bulanık c-ortalama algoritmasının (veya klasik k-ortalama) 40(kırk) objeli bir veride 3(üç) objeli kümeyi bulabilmesi için algoritmaya her iki kümeden de örnekler içerecek (3 objeli ve 37 objeli küme) bir başlangıç değeri verildiğinde küçük olan kümenin de algoritma tarafında bulunabilmesi sağlanmış olacaktır. Bu da bize kümeleme algoritmalarında eğitim safhasının önemini göstermektedir.

Kümeleme algoritmaları eğitimsiz sınıflandırıcı (unsupervised classifier) olarak adlandırılırsa yukarıda bahsedilen eğitim safhasının eklenmesiyle yarı eğitici bir sınıflandırıcı (semi-supervised classifier) elde edilmiş olacaktır.

Beyin MR görüntülerine bakıldığında kümeler arasında çok küçük farklar bulunduğu, kümeler arasında bulunan sınırların net olmadığı (bulanık olduğu) ve bu nedenle

görüntüdeki objelerin ayırt edilmesinin çok zor olduğu görülmektedir. Doğal kümeleme yöntemleriyle böyle bir görüntüde bulunan objeleri ayırt etmek oldukça zordur. Bu nedenle insanların düşünce sistemine benzer bir mantıkla işleyen bulanık sistemlerin kullanılması burada avantajlı bir durum oluşturacaktır. Bulanık sistemlerle bulanık sınırlara sahip görüntü objeleri tanımlanarak son karar aşamasına kadar bu bulanıklık kullanılmaktadır. Durulaştırma işleminde kullanılan farklı yöntemlerle objeler en uygun şekilde sınıflandırılabilir.

Yarı eğitilmiş kümeleme algoritmalarının küçük boyutlu kümeleri de bulabildiğinden bahsedilmiştir. O halde tam eğitilmiş ve zamanla kendini eğitmeye devam eden (adaptif) bir sistemin ne kadar etkili olabileceğini tahmin etmek zor değildir. Burada önemli olan oluşturulan sınıflandırıcı sisteminin iyi bir şekilde eğitilmesidir. Tümörlü bölgeyi tespit edecek doğru parametreler doğru değerleriyle bulanık sisteme öğretildiğinde sınıflandırma işleminde yüksek bir başarı oranı elde etmek sürpriz olmayacaktır.

Bu amaçla oluşturulan kural tabanlı bulanık sistem ile kümeleme algoritmalarına göre çok daha iyi sonuçlar elde edilmiştir. Kural tabanlı sistemde önemli olan kuralların oluşturulmasıdır. Doğru bir şekilde oluşturulan kurallar ve bu kuralları oluşturan dilsel değişkenlerin altındaki değer aralıkları sistemin başarısında önemli rol oynamaktadır.

Tüm bu anlatılanlar doğrultusunda iyi bir sınıflandırıcı tasarımı için aranılan objeyi en iyi ifade eden parametrelerin özellik vektörlerinden çıkarılması, bu parametreler için uygun değer aralığının belirlenmesi en önemli hususlardır. Buna yönelik olarak özellikle özellik vektörlerinden bulanık kural çıkarımı konusunda yapılacak çalışmalar ile sınıflandırıcı sistemi için en doğru kuralların oluşturulması sağlanabilir. Böylece oluşturulacak sınıflandırıcı sistemi ile tümörlü bölgelerin tespit edilmesinde çok daha iyi sonuçlar elde edilebilir.

KAYNAKLAR

Algorri, M.E., Flores-Mangas, F., "Classification of Anatomical Structures in MR Brain Images Using Fuzzy Parameters", **IEEE Transactions On Biomedical Engineering**, Vol. 51, No. 9, 1599-1608, (2004).

Almeida, R.J., Sousa, J.M.C., "Comparison of Fuzzy Clustering Algorithms for Classification", **Evolving Fuzzy Systems, 2006 International Symposium on**, 112 - 117, (2006).

Bezdek, J. C, "Pattern Recognition with Fuzzy Objective Function Algorithms", **Plenum**, (1981)

Bezdek, J.C., Keller, J., Krisnapuram, R., Pal, N.R., "Fuzzy Models and Algorithms for Pattern Recognition and Image Processing", 1st edition, 1-75, 130-137, 183-201, 241-253, 268-370, 523-547, 580-601, 663-681, 759 ,**Springer Science+Business Media, Inc.**, (2005).

Breiman, L., Freidman, J.H., Olshen, R.A., Stone, C. J. "Classification and Regression Trees", **Wadsworth, Inc.**, Belmont, (1984).

Canny, J., "A Computational Approach to Edge Detection," **IEEE Transactions on Pattern Analysis and Machine Intelligence**, Vol. PAMI-8, No. 6, 679-698, (1986) .

Chang, R.L.P., Pavlidis, T. "Fuzzy Decision Tree Algorithms", **IEEE Trans. Syst. Man and Cybems.**, 28-35, (1977).

Devijver, P., Kittler, J., "Pattern Recognition: a Statistical Approach", **Prentice-Hall, Englewood Cliffs**, (1982).

Fisher, R.A., "The Use of Multiple Measurements in Taxonomic Problems", **Annual Eugenics**, 179-188, (1936).

Fu, K.S., "Syntactic Pattern Recognition and Applications", **Prentice Hall, Englewood Cliffs, NJ.**, (1982).

Fukunaga, K., "Statistical Pattern Recognition, Second Edition (First Edition, 1972)", **Academic Press**, San Diego, (1991).

Gonzalez, R. C., Thomason, M. G., "Syntactic Pattern Recognition : an Introduction", **Addison Wesley, Reading**, (1978).

Gonzalez, R.C., Woods, R.E., "Digital Image Processing 2nd Edition (DIP/2e)", 75-134, 220-277, 519-750, **Prentice Hall**, (2002).

Gonzalez, R.C., Woods, R.E., Eddins, S.L., "Digital Image Processing Using MATLAB", 1st edition, 1-109, 141-195, 335-514 , **Prentice Hall**, (2004).

Gordillo, N, Montseny, E., Sobrevilla, P., "A New Fuzzy Approach To Brain Tumor Segmentation", **Fuzzy Systems (FUZZ), 2010 IEEE International Conference on**, 1-8, (2010).

Hemanth, D.J., Selvathi, D., Anitha, J., "Effective Fuzzy Clustering Algorithm for Abnormal MR Brain Image Segmentation", **Advance Computing Conference, 2009. IACC 2009. IEEE International**, 609-614, (2009).

House, J. M., Lee, Y. W., Shin, D. R., "Classification Techniques for Fault Detection and Diagnosis of an Air Handling Unit", **Trans. ASHRAE**, (1999).

Kaufmann, A., "Introduction to the Theory of Fuzzy Subsets Fundamental Theoretical Elements", 1, **Academic Press, NY.**, (1975).

Mamdani, E.H., Assilian, S. "An Experiment in Linguistic Synthesis with a Fuzzy Logic Controller", **Int. J. of Man-Machine Studies**, 7(1),1-13, (1975).

Nedeljkovic, I., "Image Classification Based on Fuzzy Logic", **International Society for Photogrammetry and Remote Sensing**, İstanbul, (2004).

Özkurt, N., Özkurt, A., "Unsupervised Segmentation of MRI Using Independent Component Analysis", **Signal Processing and Communications Applications, 2007. SIU 2007. IEEE 15th**, 1-4, (2007).

Quinlan, J. R., "Learning Efficient Classification Procedures and Their Application to Chess End-Games", **Machine Learning**, (1983).

Quinlan, J. R., "Induction of Decision Trees", **Machine Learning**, 1, 81-106, (1986).

Kharrat, A., Benamrane, N., Ben Messaoud, M., Abid, M., "Detection of Brain Tumor in Medical Images", **Signals, Circuits and Systems (SCS), 2009 3rd International Conference on**, 1-6, (2009).

Li, Z., Nie, S., Chen, Z., "Application of Independent Component Analysis in MR Image Segmentation", **Bioinformatics and Biomedical Engineering, 2008. ICBBE 2008. The 2nd International Conference on**, 2686-2689, (2008).

Prastawa, M., Bullitt, E., Gerig, G., "Simulation of Brain Tumors in MR Images for Evaluation of Segmentation Efficacy", **Medical Image Analysis**, 297-311, (2009).

Sebiskveradze, D., Ly, E., Gobinet, C., Piot, O., Manfait, M., Jeannesson, P., Vrabie, V., "From Preprocessing to Fuzzy Classification of MR Images of Paraffin Embedded Cancerous Skin Samples", **Hyperspectral Image and Signal Processing: Evolution in Remote Sensing, 2009. WHISPERS '09. First Workshop on**, 1-4, (2009).

Schalkoff, R., "Pattern Recognition: Statistical, Structural and Neural Approaches", **John Wiley and Sons**, (1992).

Takagi, T., Sugeno, M., "Fuzzy Identification of Systems and Its Application to Modeling and Control", **IEEE Trans. Syst, Man and Cybems.**, 116-132, (1985).

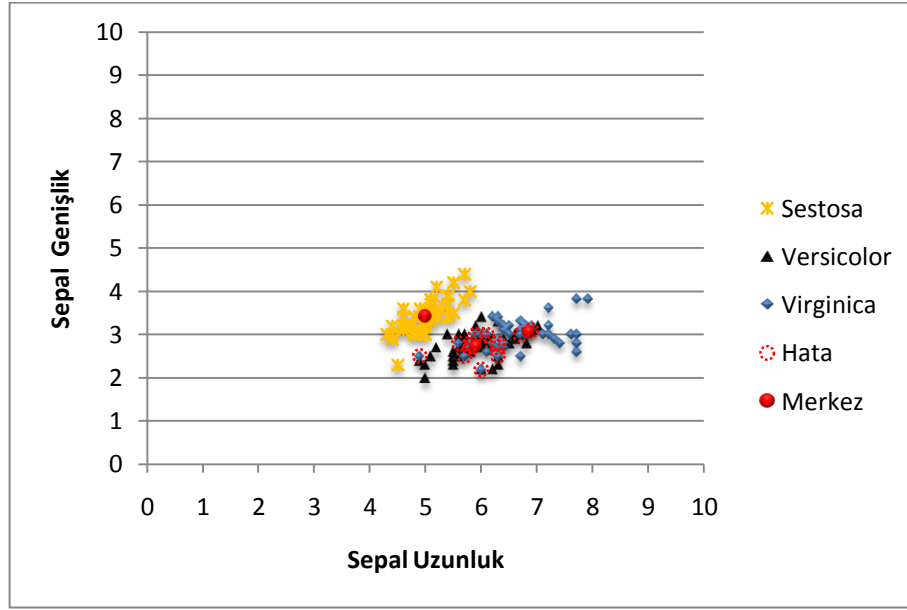
Wang, K., Wan, Y., Shen S., "Classification of Remote Sensing Images Using Fuzzy Multi-Classifiers", **Intelligent Computing and Intelligent Systems, 2009. ICIS 2009. IEEE International Conference on**, Shanghai, Vol. 4, 411-414, (2009).

Xuan, X., Liao, Q., "Statistical Structure Analysis in MRI Brain Tumor Segmentation", **Image and Graphics, 2007. ICG 2007. Fourth International Conference on**, 421-426, (2007).

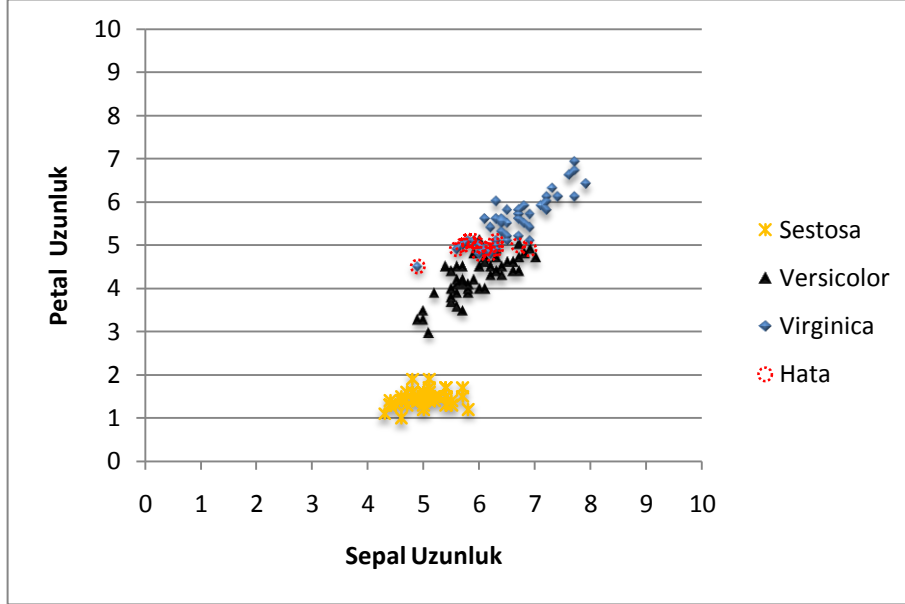
Verma, N.K., Gupta, P., Agrawal, P., Yan Cui, "MRI Brain Image Segmentation For Spotting Tumors Using Improved Mountain Clustering Approach", **Applied Imagery Pattern Recognition Workshop (AIPRW)**, 1-8, (2009).

EKLER

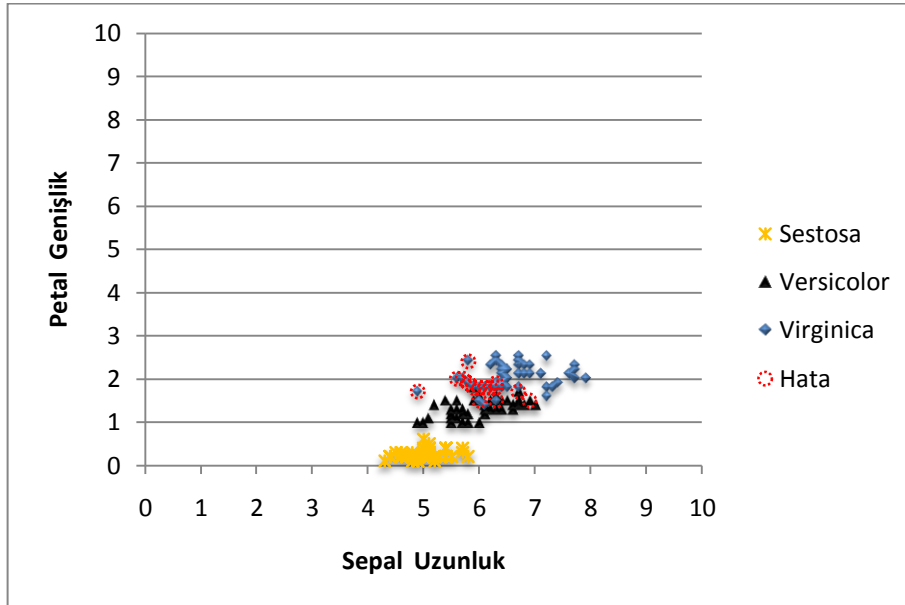
EK-A: KLASİK K-ORTALAMA VE BULANIK C-ORTALAMA ALGORİTMALARININ IRIS VERİ KÜMESİNE UYGULANMASI İLE ELDE EDİLEN SONUÇLAR



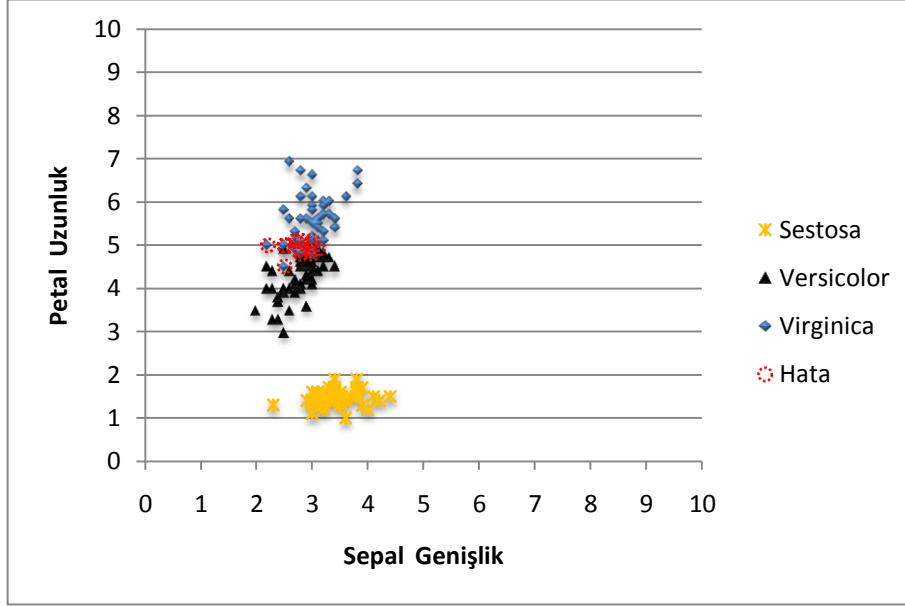
Şekil A-1



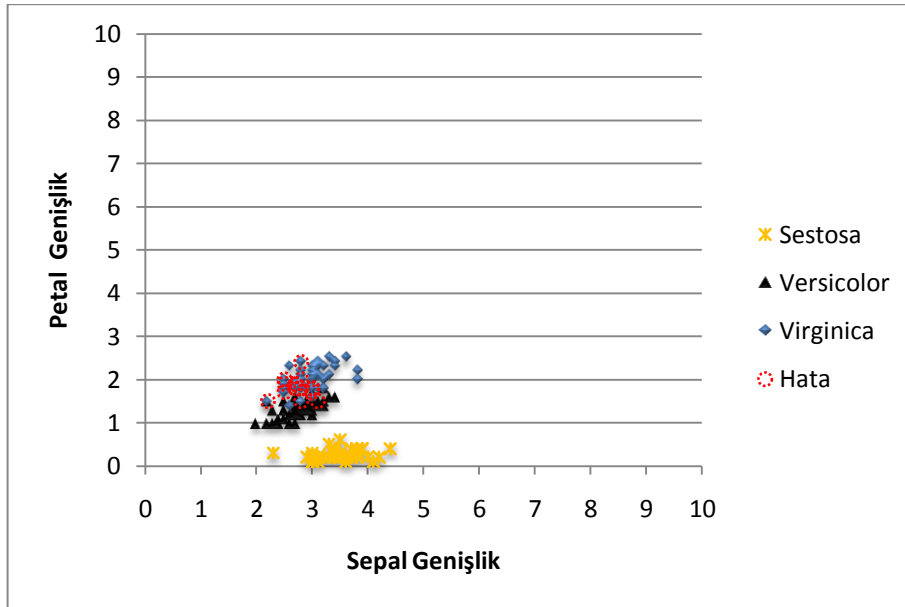
Şekil A-2



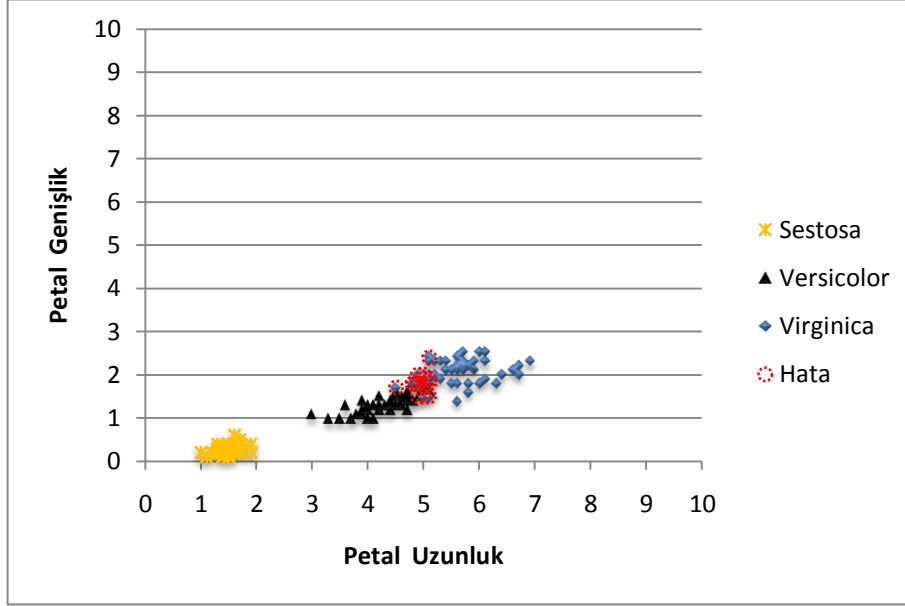
Şekil A-3



Şekil A-4

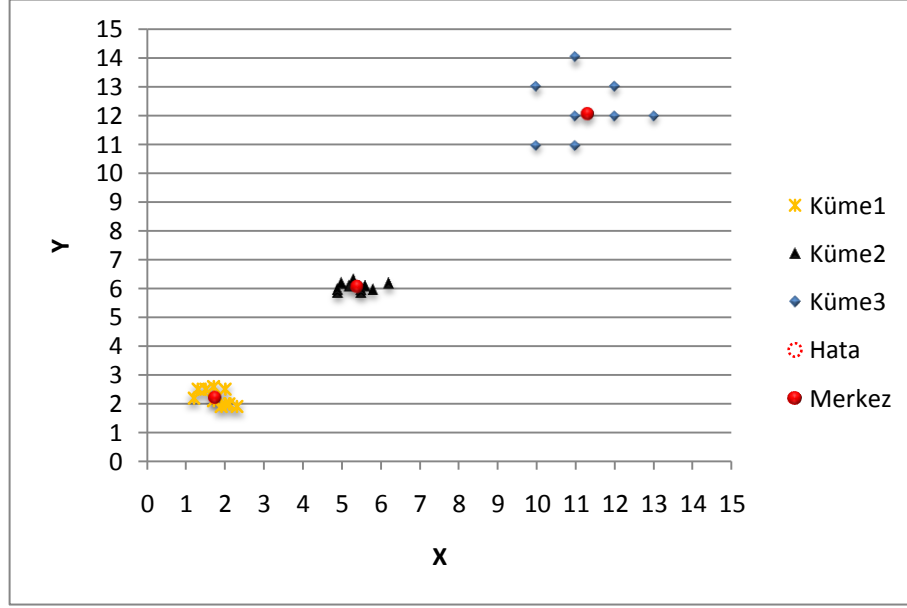


Şekil A-5



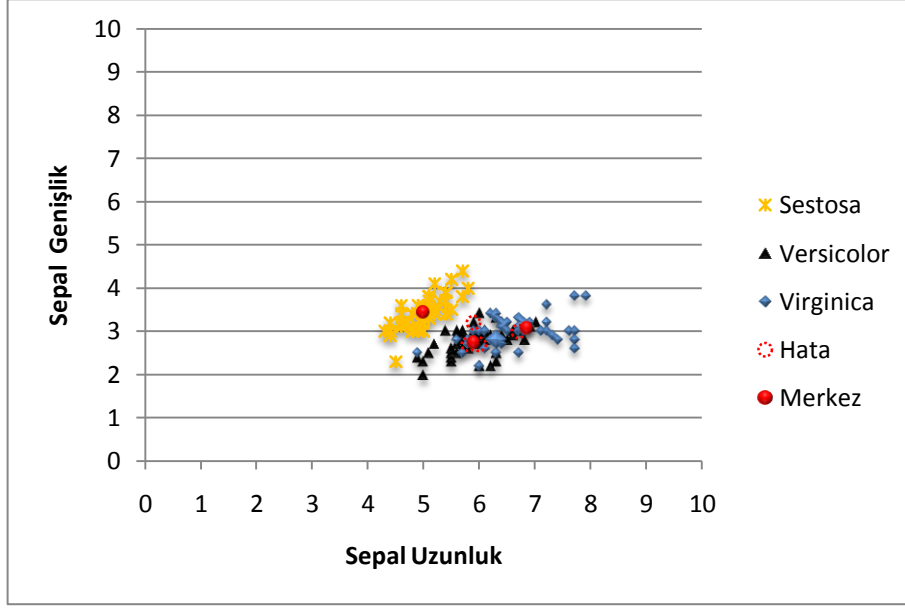
Şekil A-6

**EK-B: KLASİK K-ORTALAMA VE BULANIK C-ORTALAMA
ALGORİTMALARININ X_{30} VERİ KÜMESİNE UYGULANMASI İLE ELDE EDİLEN
SONUÇLAR**

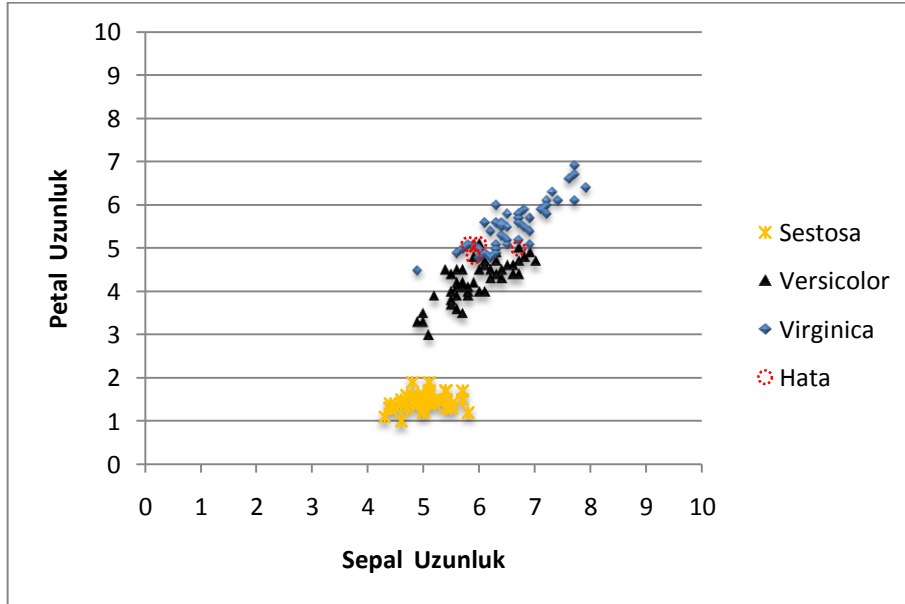


Şekil B-1

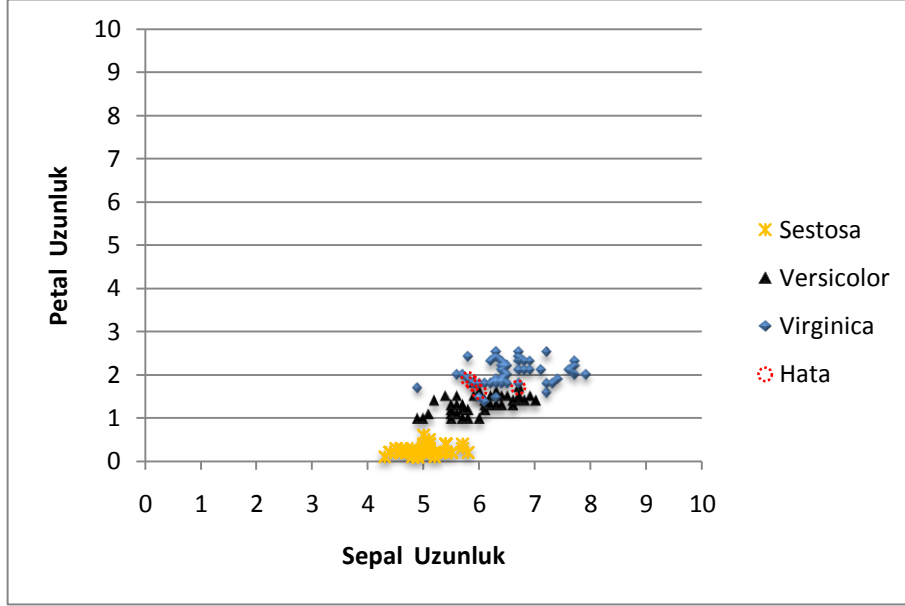
EK-C: MAMDANI BULANIK SİSTEMİ İLE IRIS VERİ KÜMESİNİN SINIFLANDIRILMASI



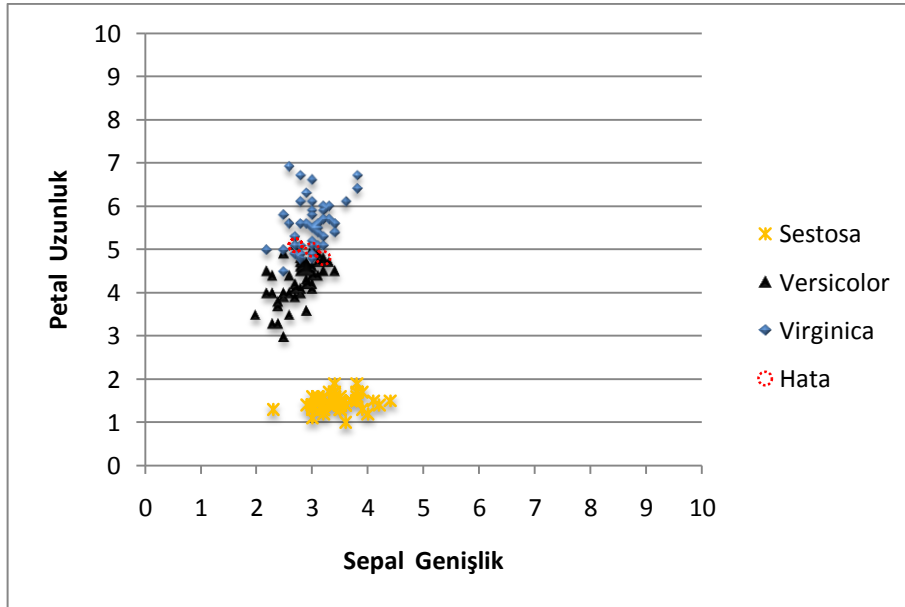
Şekil C-1



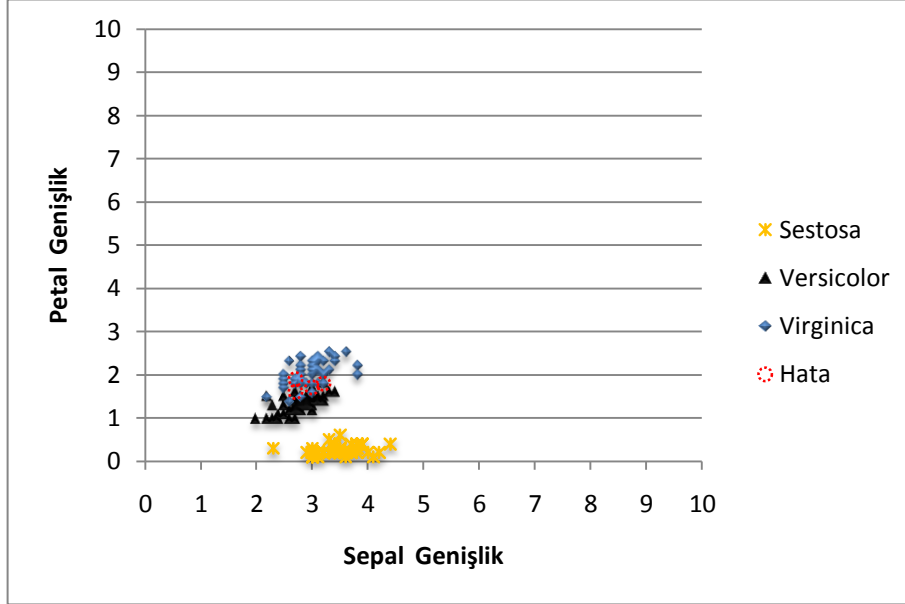
Şekil C-2



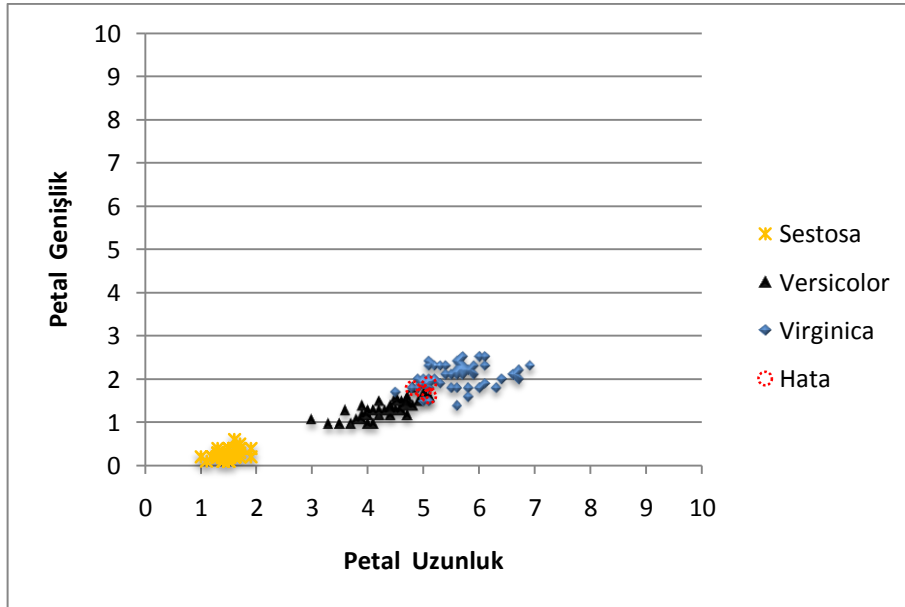
Şekil C-3



Şekil C-4



Şekil C-5



Şekil C-6

ÖZGEÇMİŞ

1981 yılında Trabzon'da doğdu. İlk, orta ve lise öğrenimini İstanbul'da tamamladı. 2002 yılında girdiği Kocaeli Üniversitesi Mühendislik Fakültesi Bilgisayar Mühendisliği Bölümünden 2007 yılında mezun oldu. Aynı yıl Kocaeli Üniversitesi Fen Bilimleri Enstitüsü Bilgisayar Mühendisliği Anabilim Dalı'nda Yüksek Lisans eğitimine başladı.