

Örüntü Tanımada Altuzay Metriklerinin Birleřtirilmesi

řükran Künkü

DOKTORA TEZİ

Elektronik Mühendislięi Anabilim Dalı

Mayıs 2008

Combinig with The Subspace Metrics on Pattern Recognition

Sukran Kunkcu

DOKTORAL DISSERTATION

Department of Electric and Electronic Engineering

May 2008

Örüntü Tanımda Altuzay Metriklerinin Birleştirilmesi

Şükran Künkçü

Eskişehir Osmangazi Üniversitesi
Fen Bilimleri Enstitüsü
Lisansüstü Yönetmeliği Uyarınca
Elektrik-Elektronik Anabilim Dalı
Elektronik Bilim Dalında
DOKTORA TEZİ
Olarak Hazırlanmıştır

Danışman: Prof.Dr.Atalay BARKANA

Mayıs 2008

Şükran Künkçü' nün DOKTORA tezi olarak hazırladığı “Örüntü Tanımda Altuzay Metriklerinin Birleştirilmesi” başlıklı bu çalışma, jürimizce lisansüstü yönetmeliğinin ilgili maddeleri uyarınca değerlendirilerek kabul edilmiştir.

Üye : Prof.Dr. Atalay BARKANA

Üye : Prof.Dr. M.Bilginer GÜLMEZOĞLU

Üye : Yrd.Doç.Dr. Rifat EDİZKAN

Üye : Yrd.Doç.Dr. Atıf ÇAY

Üye : Yrd.Doç.Dr. Erol SEKE

Fen Bilimleri Enstitüsü Yönetim Kurulu'nun tarih ve sayılı kararıyla onaylanmıştır.

Prof. Dr. Nimetullah BURNAK

Enstitü Müdürü

ÖZET

Sınıflandırıcı birleştirmedeki amaç örüntü tanımda daha iyi başarımlar elde etmektir. Örüntü sınıflamada, sınıf içi ve sınıflar arası dağılımları birlikte kullanılmasının sınıflandırma başarımını arttırması beklenmektedir. Bu çalışmada, iki-sınıf problemi için sınıf içi ve sınıflar arası dağılımdan elde edilen ölçütleri bir yapay sinir ağı mimarisi ile birleştiren yeni bir sınıflandırıcı geliştirilmiştir. Deneysel çalışmada, yapay veriler, MNIST, TIMIT ve IRIS veri tabanları kullanılmıştır. Geliştirilen sınıflandırıcı iki-sınıf problemi göz önüne alınarak bu veri kümelerinde test edilmiş ve diğer alt uzay yöntemleri ve çoklu sınıflayıcılarla karşılaştırılmıştır. Doğru tanıma yüzde başarısı olarak; yapay veriler için % 90, MNIST veri tabanı için %91.5, TIMIT veri tabanı için %50.88, IRIS veri tabanı için %92.91 sonuçları elde edilmiştir. Elde edilen sonuçlar değerlendirildiğinde geliştirilen sınıflandırıcı birleştirme yönteminin, yapay veri üzerinde diğer altuzay ölçütlerine göre daha başarılı olduğu görülmüştür. MNIST veri tabanında daha iyi sonuçlar alınamamıştır. TIMIT ve IRIS veri tabanlarında, sınıflandırma sonuçlarından beklenen iyileştirme elde edilememiştir. Bunun nedeni kullanılan sınıflar-arası ölçütün sınıflamaya çok fazla katkısının olmamasıdır. Yapay veriler üzerinden elde edilen sonuçlar, önerilen sınıflandırıcının geliştirilerek gerçek sınıflandırma problemlerinde kullanılabileceğini göstermektedir.

Anahtar Kelimeler: Örüntü tanıma, sınıflandırıcı birleştirme, altuzay sınıflandırma yöntemleri, çoklu sınıflayıcılar, iki sınıf problemi.

SUMMARY

The purpose of classifier combination is to achieve better recognition rates in pattern recognition. It is expected that successful classifiers are achieved by using within class and between class distributions in pattern recognition. In this thesis, a new classifier is developed, which is used by combining metrics that are obtained from within_Class and between_Class distributions by using neural network architecture. In the experimental work, the artificial data, MNIST, TIMIT and IRIS data base were used for two class problem. The performance of the combined classifier was tested on these database and compared with the other subspace methods and the other multiple classifiers. The performance of the new combined classifier are 90 % and 85 % for artificial data, 91.5 % for MNIST database, 50.88 % for TIMIT database, 92.91 % for IRIS database. When the obtained results are examined, it is observed that the developed combined method on the artificial data is much more successful than the other subspace metrics. However better results could not be obtained on MNIST database. In many cases, it is observed that the developed method on TIMIT and IRIS databases give better results. In combined method, the usage of between class metric must be reviewed.

Key Words: Pattern recognition, combining classifiers, subspace classifiers, multiple classifiers, two_class problem.

TEŞEKKÜR

Doktora tez aşamasında, bana danışmanlık ederek, beni yönlendiren ve destekleyen danışmanım Prof.Dr. Atalay BARKANA'ya, danışmanlığını esirgemeyen, yol gösteren Prof.Dr. M.Bilginer Gülmezoğlu, Yrd.Doç. Rifat EDİZKAN ve Arş.Gör. Mehmet KOÇ'a teşekkür ederim.

Ayrıca doktora tezim boyunca benden maddi, manevi desteklerini esirgemeyen eşime ve sabırla annelerinin mezun olmasını bekleyen çocuklarıma da teşekkür ederim.

İÇİNDEKİLER

	<u>Sayfa</u>
ÖZET	v
SUMMARY	vi
TEŞEKKÜR.....	vii
İÇİNDEKİLER.....	viii
ŞEKİLLER DİZİNİ.....	x
ÇİZELGELER DİZİNİ.....	xii
SİMGELER VE KISALTMALAR DİZİNİ.....	xiii
1. GİRİŞ.....	1
2. ALTUZAY TEKNİKLERİNİ KULLANAN SINIFLANDIRICILAR.....	3
2.1 Ortak Vektör Yaklaşımı (OVY)	3
2.2 Temel Bileşen Analizi	6
2.3 Fisher LDA	8
3. SINIFLANDIRICI BİRLEŞTİRME.....	9
3.1 Sınıflandırıcı Birleştirme Yöntemleri	9
3.2 Sınıf İçi ve Sınıflar Arası Ölçütlerin Birleştirilmesi.....	11
3.2.1 Analitik yöntem	12
3.2.2 Steepest Descent metodu:	14
3.2.3 Sınır ağı yapısında birleştirme	15
4. DENEYSEL ÇALIŞMALAR	18
4.1 Yapay Veri-1.....	18
4.1.1 OVY ve SAM için yapay veri-1 Sonuçları.....	19
4.1.2 Yapay veri-1 için Fisher_LDA uygulanması.....	20
4.1.3 Yapay veri-1 için iki metriğin birleştirilmesi:	21
4.2 Yapay Veri-2 :	24
4.2.1 OVY ve SAM algoritmalarının yapay veri-2'ye uygulanması.....	25
4.2.2 Fisher LDA için yapay veri-2 sonuçları	26
4.2.3 İki metriğin birleştirilmesi	29

4.3	Yapay Veri-3.....	33
4.3.1	OVY ve SAM için yapay veri-3 Sonuçları.....	34
4.3.2	Yapay veri-3 için Fisher_LDA uygulanması.....	36
4.3.3	OVY ve SAM ölçütlerini oranlayarak yapay veri-3 için sınıflandırma..	36
4.3.4	Yapay veri-3 için birleştirilmiş metrik ile sınıflandırma	37
4.4	MNIST Veri Tabanı.....	39
4.5	TIMIT Veri Tabanı	43
4.6	IRIS Veri Tabanı.....	47
5.	SONUÇLAR.....	52
6.	KAYNAKLAR DİZİNİ	53

ŞEKİLLER DİZİNİ

<u>Şekil</u>	<u>Sayfa</u>
Şekil 2.1 OVY sınıflandırıcının iki sınıf probleminde yapay sinir ağı gösterimi.....	5
Şekil 2.2 SAM sınıflandırıcının iki sınıf probleminde yapay sinir ağı gösterimi.....	7
Şekil 3.1 F_W ve F_B normları ve ağırlıklar arasındaki ilişki.....	12
Şekil 3.2 Ağırlıkların bulunacağı $w_{1c_1} - w_{1c_2}$ düzlemi	14
Şekil 3.3 OVY ve SAM ölçütlerini birleştiren sınıflandırıcının yapay sinir ağı şekli....	16
Şekil 4.1 Yapay veri-1 için C_1 ve C_2 sınıflarının x-y düzleminde dağılımı.....	18
Şekil 4.2 Yapay veri-1 için OVY ile sınıflandırma	19
Şekil 4.3 Yapay veri-1 için SAM ile sınıflandırma	20
Şekil 4.4 Yapay veri-1 için LDA ile sınıflandırma.....	21
Şekil 4.5 Yapay veri-1 için OVY ve SAM ölçütleri ile ağırlıksız birleştirme.....	22
Şekil 4.6 Yapay veri-1 için OVY ve SAM'den elde edilen ölçütlerin birleştirilmesi.....	23
Şekil 4.7 Yapay veri-1 için grid üzerindeki ağırlıklarla ölçütlerin birleştirilmesi.....	23
Şekil 4.8 Yapay veri-2 için C_1 ve C_2 sınıflarını x-y düzlemindeki dağılımlar	24
Şekil 4.9 Yapay veri-2'in OVY ile sınıflandırılması	25
Şekil 4.10 Yapay veri-2'nin SAM ile sınıflandırılması	26
Şekil 4.11 Fisher LDA için yapay verinin sınıflandırılması.....	27
Şekil 4.12 Uygulama 2-a için yapay veri-2'nin sınıflandırılması.....	28
Şekil 4.13 Uygulama 2-b'ye göre yapay veri-2'nin sınıflandırılması	29
Şekil 4.14 Ağırlıksız birleştirme ile yapay veri-2'nin sınıflandırılması	30
Şekil 4.15 Yapay veri-2 için ağırlıklar kullanarak birleştirme	31
Şekil 4.16 Yapay veri-2 için grid üzerindeki ağırlıkları kullanarak birleştirme.....	32
Şekil 4.17 Yapay veri-3 için C_1 ve C_2 sınıflarının x-y düzleminde dağılımı.....	34
Şekil 4.18 Yapay veri-3 için OVY ile sınıflandırma	35
Şekil 4.19 Yapay veri-3 için SAM ile sınıflandırma	35
Şekil 4.20 Yapay veri-3 için LDA ile sınıflandırma.....	36

Şekil 4.21 Yapay veri-3 için Fisher-Oran ile sınıflandırma.....	37
Şekil 4.22 Yapay veri-3 için OVY ve SAM'den elde edilen ölçütlerin birleştirilmesi ...	38
Şekil 4.23 OVY ve SAM'den elde edilen ölçütlerin yapay sinir ağına uygulanması	49

ÇİZELGELER DİZİNİ

<u>Çizelge</u>	<u>Sayfa</u>
Çizelge 4.1 Yapay veri-1 için tanıma başarımları	24
Çizelge 4.2 Yapay veri-2 için tanıma başarımları	32
Çizelge 4.3 Yapay veri-3 için tanıma başarımları	39
Çizelge 4.4 El yazısı rakamlardan oluşan veri kümesinin rakamlara göre dağılımı.....	39
Çizelge 4.5 El yazısı karakter tanıma sonuçları.....	41
Çizelge 4.6 Sınıflayıcıların birbirlerini tamamlaması ile ilgili sonuçlar	42
Çizelge 4.7 Konuşmacıların lehçe dağılımları.....	43
Çizelge 4.8 Timit veri tabanı ünlülere göre dağılımı.....	44
Çizelge 4.9 TIMIT veri tabanının eğitim kümesi sınıflandırma sonuçları	46
Çizelge 4.10 TIMIT veri tabanının test kümesi sınıflandırma sonuçları	47
Çizelge 4.11 IRIS veri tabanı eğitim seti sınıflandırıcı sonuçları.....	50
Çizelge 4.12 IRIS veri tabanı test seti sınıflandırıcı sonuçları	51

SİMGELER VE KISALTMALAR DİZİNİ

<u>Simgeler</u>	<u>Açıklama</u>
C_1	Veri kümesinin 1. sınıfı
C_2	Veri kümesinin 2. sınıfı
$F_{w_{C_i}}$	i. Sınıf için sınıf içi dağılımdan elde edilen ölçüt
$F_{B_{C_i}}$	i. Sınıf için sınıflar arası dağılımdan elde edilen ölçüt
Φ_w^C	C sınıfının sınıf içi ortak değişinti matrisi
Φ_B^C	C sınıfının sınıflar arası ortak değişinti matrisi

Kısaltmalar

BM	: Birleştirilmiş Ölçüt
CVA	: Common Vector Approach
Değişinti	: Kovaryans
Enküçükleme	: Minimizasyon
Enbüyükleme	: Maksimizasyon
GMM	: Gaussian Mixture Model
İzdüşüm	: Projeksiyon(Projection)
LDA	: Linear Discriminant Analysis(Doğrusal Ayırma Analizi)
NN	: Neural Network
OVY	: Ortak Vektör Yaklaşımı
PCA	: Principal Component Analysis
SAM	: Sınıflar arası Metrik

1. GİRİŞ

Bugüne kadar deęişik öznitelik vektörleri ve farklı ölçütler kullanarak, farklı karar verme kriterlerine göre çalışan birçok örüntü(patern) tanıma sistemi geliştirilmiştir. Bu tanıma sistemlerinde en yaygın olarak kullanılan yöntemler HMM (Hidden Markov Model), NN(Neural Networks), GMM (Gaussian Mixture Model) ve altuzay teknikleridir. Kullanılan herhangi bir sınıflayıcı sistemin performansını, tanımda kullanılan parametreler, tanıma için kullanılan modellerde sınıfların ne kadar iyi temsil edildięi, tanıma yönteminin sınıflar arasındaki farkları ne kadar gözönüne aldığı gibi birçok faktör belirleyecektir. Performans artımı sağlamak için yöntemin iyileştirilmesi bu faktörlerin iyileştirilmesi olarak düşünülebilir (Bennani et al,1995), (Reynolds et al,1995). Sınıflayıcıların birçoęunda karar aşamasında tek bir ölçüt kullanarak karar verilir.

Özellikle altuzay teknikleri, kullanılan öznitelik vektörlerinden en etkili olan bölümleri alarak kalanları atmak yoluyla performans artımını sağlarlar. Öznitelik vektörlerinin bazı bölümlerini atmak, bir çeşit “discriminative training” yapmak anlamına gelmekte ve modellerin güvenilir olmadığı durumlarda performans artımını sağlamaktadır. OVY, SAM, LDA ve KLT gibi alt uzay teknikleri kullanan birçok sınıflayıcı bulunmaktadır (Kitler et al,1998), (Bishop, 1996). Performans artımını sağlamanın dięer yöntemlerinden birisi de karar birleştirme kurallarını ya da çoklu sınıflayıcıları kullanmaktır. Sınıflandırıcıların özellikle de karmaşık örüntü tanımda daha yüksek tanıma performansı elde edebilmesi için çoklu sınıflandırıcılar kullanılmaktadır.

Çoklu sınıflandırıcı kullanarak bu sınıflandırıcıları birleştirmek ve karar için birden fazla ölçüt kullanmak, daha iyi sonuçlar almak için uygulanacak yöntemlerden birisidir. Bu yöntemlerde esas olan, birbirinden farklı ve birbirlerini tamamlayan sınıflandırıcıları belirlemek ve bu sınıflandırıcıları aynı anda kullanan birleştirme yöntemini geliştirmektir. Böylece birden fazla sınıflayıcının üstün özellikleri,

birbirlerini tamamlayacak şekilde bir arada kullanılarak örüntü tanımada daha yüksek başarımlar elde edilebilir. Çoklu sınıflandırıcıların, literatürde yüz tanıma (Lu et al, 2003), el yazısı karakter tanım (Cao et al, 1995), (Huang et al,1995), kişisel kimlik doğrulama (Kitler et al,1998), ses tanıma (Felföldi et al, 2003) ve kişi tanıma(Radova et al, 1997) gibi örüntü tanıma problemlerine uygulandığı görülmektedir. Çoklu sınıflandırıcılara, hibrit yöntemler, karar birleştirme, çoklu uzmanlar, sınıflandırıcı topluluğu veya algılayıcı kaynaşımı da denmektedir (Koç M., 2006).

Çoklu sınıflayıcılar, mimarilerine, girdi-çıkı ilişkisine, çıkı türüne göre gibi pek çok şekilde gruplandırılabilir. Bunlar arasında en yaygın olanı çoklu sınıflandırıcıyı oluştururken doğrusal ve doğrusal olmayan birleştirmedir. Doğrusal birleştirmede her sınıflandırıcı belirli bir ağırlıkla çarpılır. Ağırlıklı ortalama (Heskes, 1997), bulanık integral (Gader et al, 1996) doğrusal birleşim yöntemlerdendir. Bunun yanında çoğunluk oyu (Ji, and Ma, 1997), enbüyük olanı seç, “Borda Count”, lojistik Regresyon Yöntemi doğrusal olmayan çoklu sınıflandırıcı yöntemleridir (Koç M., 2006).

Çoklu sınıflandırıcıları kullanmanın ve karar birleştirmenin sistem performansını artırması yanında bu artışın bazı problemlerde gerçekleşmediği de gözlenmektedir. Bu sorun literatürde yaygın olarak sınıflandırıcıların bağımsızlığı (independence) ve tamamlayıcılığı (complimentaries) kavramları ile açıklanmıştır (Saranlı A., 2000), (Kuncheva, 2004). Sınıflandırıcıların hatalarının farklı kümelerde olması performansı artıracak etkenlerdendir.

Tez dökümanının 2. bölümünde alt uzay teknikleri kullanan sınıflandırıcılardan bahsedilmiştir. 3.bölümünde bu tezde geliştirilen yeni yöntem anlatılmıştır. Bu yöntem OVY ve SAM'den türetilen ölçütleri sinir ağı yapısında birleştirmektedir. 4.Bölümde ise deneysel çalışmalar ve karşılaştırmalı sonuçlar verilmiştir. Son bölümde ise sonuç ve öneriler anlatılmaktadır.

2. ALTUZAY TEKNİKLERİNİ KULLANAN SINIFLANDIRICILAR

Bu bölümde, tez çalışmasında kullanılan bazı altuzay teknikleri ve sonuçları karşılaştırmak için üzerinde çalıştığımız bazı çoklu sınıflayıcılar anlatılmıştır.

Tez çalışmasında, sınıf içi ve sınıflar arası dağılımlara uygulanan alt uzay teknikleriyle elde edilen ölçütler birleştirilmiştir. Geliştirilen yöntemde OVY'dan elde edilen enküçükleme işlemi ve SAM'dan türetilen en büyükleme işlemi (SAM) birleştiren yeni bir ölçüt kullanılmıştır. Tez çalışması boyunca bu yöntem birleştirilmiş yöntem diye anılacaktır. Bu yöntem ile Fisher LDA kavramsal olarak bazı benzerlikler içerdiğinden sonuçların karşılaştırılmasında Fisher LDA de kullanılmıştır. Bu yüzden tüm sınıflandırıcıların içinde özellikle bu yöntemlerden aşağıda bahsedilmiştir.

2.1 Ortak Vektör Yaklaşımı (OVY)

Ortak vektör yaklaşımı örüntü tanımada kullanılan altuzay yöntemidir(Gülmezoğlu et al., 1999; Gülmezoğlu, et al., 2001). Bir sınıftaki verilerin o sınıfa has olan özellikleri ve verilerin birbirinden farklılık gösteren özellikleri vardır. Ortak vektör, bir sınıfa ait özellik vektörlerindeki farklılıklar çıkarıldıktan sonra geriye kalan ve o sınıf için değişmeyen özelliklerin oluşturduğu vektöre denir(Gülmezoğlu, et al., 2001). OVY'nin temelinde bir sınıfa ait özellik vektörlerinin ortak özelliklerini gösteren tek vektörün varlığı yatar. Ortak vektörde, yetersiz ($n \geq m$) ve yeterli ($n < m$) veri durumu olmak üzere iki durum söz konusudur. Burada m sınıftaki veri sayısı, n ise verilerin boyutunu göstermektedir.

Yetersiz veri durumu($n \geq m$) için bir sınıfa ait herhangi bir vektörün fark uzayına izdüşümünün vektörün kendisinden çıkarılması ile elde edilen vektör, daima ortak vektörü verir. Yeterli veri durumu ($n < m$) için ise sınıfa ait vektörün farksızlık uzayına izdüşümü ortak vektöre yakın olacaktır.

Birleştirme yöntemindeki enküçükleme OVY'den türetilmiştir. Sınıf içi dağılımda bir sınıfın elemanlarının kendi sınıfının ortalamasına yakın olması gerekir. Bu durumda aşağıdaki ölçüt enküçük olmalıdır. Bu ölçüt şu şekilde yazılabilir:

$$F_{w_{C_i}} = \sum_{C_i} \left\| \mathbf{P}_{w_{C_i}}^\perp (\vec{a}_i - \vec{a}_{\text{ort}_{C_i}}) \right\|^2 \quad (2.1)$$

Burada w alt indisi sınıf içi dağılımı göstermektedir. Bu metriği C_1 ve C_2 sınıfları için ayrı ayrı yazarsak;

$$F_{w_{C_1}} = \sum_{C_1} \left\| \mathbf{P}_{w_{C_1}}^\perp (\vec{a}_i - \vec{a}_{\text{ort}_{C_1}}) \right\|^2 \quad \text{ve}$$

$$F_{w_{C_2}} = \sum_{C_2} \left\| \mathbf{P}_{w_{C_2}}^\perp (\vec{a}_i - \vec{a}_{\text{ort}_{C_2}}) \right\|^2 \quad (2.2)$$

olacaktır.

\vec{a}_x bilinmeyen vektör olmak üzere $F_{w_{C_i}}$ 'lerden hangisi küçük ise \vec{a}_x o sınıfa aittir.

Herhangi bir C sınıfındaki vektörler \vec{a}_i^C ($i=1,2,\dots,m$) ile gösterilmek üzere sınıf içi ortak değişinti matrisi aşağıdaki gibi yazılır.

$$\Phi_w^C = \sum_{i=1}^m (\vec{a}_i^C - \vec{a}_{\text{ort}}^C)(\vec{a}_i^C - \vec{a}_{\text{ort}}^C)^T \quad (2.3)$$

Burada \vec{a}_{ort}^C , C sınıfının ortalamasıdır.

İç içe geçmiş sınıfların bulunabileceği göz önüne alındığında, her sınıfın elemanları kendi sınıfının ortalamasından çıkarılmış, farklı sınıfların biraz daha birbirinden ayrılarak kendi sınıf elemanlarına daha yakın olması sağlanmıştır. Enküçükleme işlemi her sınıf için ayrı ayrı yapılır. Böylelikle, sınıf elemanları sınıf ortalamasına yakın hale getirilir.

Sınıf içi ortak değışinti matrisin m tane özdeğerinden, en küçük m_1 tanesi ve bunlara karşılık gelen özvektörler (\vec{z}) kullanılarak m_1 boyutlu farksızlık altuzayı izdüşüm matrisi elde edilir :

$$P_{w_i}^\perp = \sum_{i=1}^{m_1} \vec{z}_i \vec{z}_i^T \quad (2.4)$$

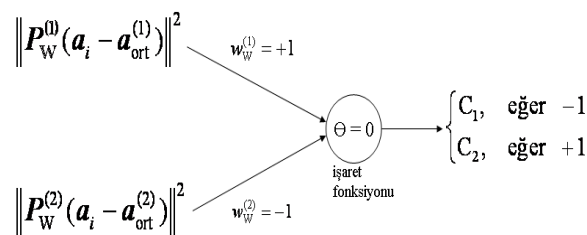
Burada $P_{w_{c1}}^\perp$ ve $P_{w_{c2}}^\perp$ farksızlık altuzayı izdüşüm matrislerini göstermektedir. $P_{w_{ci}}^\perp$ 'ler ortak değışinti matrisin en küçük özdeğerlerine karşılık gelen -ki bunlar farksızlık uzayını ifade ederler, özvektörlerden elde edilir. Buradan elde edilecek altuzayda -ki bu uzay sınıf-içi dağılımların oluşturacağı alt uzaydır, enküçükleme yapılarak bilinmeyen vektörün hangi sınıfa ait olduğu bulunur. Karar kriteri yazılacak olursa;

$$F_{w_{C_1}} = \left\| P_{w_{C_1}}^\perp (\vec{a}_i - \vec{a}_{\text{ort}_{C_1}}) \right\|^2 \quad \text{ve} \quad F_{w_{C_2}} = \left\| P_{w_{C_2}}^\perp (\vec{a}_i - \vec{a}_{\text{ort}_{C_2}}) \right\|^2 \quad (2.5)$$

\vec{a}_x bilinmeyen vektör olmak üzere F 'lerden hangisi küçük ise \vec{a}_x , o sınıfa aittir.

$$c_* = \arg \min_k \left\| P_{w_{C_i}} (\vec{a}_x - \vec{a}_{i,\text{ort}}) \right\|^2 = \arg \min (F_{w_{C_1}}, F_{w_{C_2}}) \quad i, k = 1, 2 \quad (2.6)$$

Şekil 2.1'de sınıf-içi değışinti ölçütünü kullanan sınıflandırıcının, iki sınıflı sınıflandırma problemi için sinir ağı formunda gösterimi verilmiştir.



Şekil 2.1 OVY sınıflandırıcının iki sınıf probleminde yapay sinir ağı gösterimi

2.2 Temel Bileşen Analizi

Temel bileşen analizi, sinyal işleme uygulamalarında kullanılan(Zhang, et al, 2001; Duin, et al, 2000), çok boyutlu verileri, az boyutlu verilere en az bilgi kaybıyla indirgeyen bir dönüşüm tekniğidir(Kramer, 1991, KirBM et al., 1990, Wang et al., 2002). Bu yönteme aynı zamanda Karhunen-Loeve dönüşümü de denir(Bishop, 1996). SAM aynı zamanda sınıflandırma için de kullanılmaktadır(Oja, 1983, Günal et al, 2005).

Bu dönüşüm yapılırken ortak değişinti matrisinin özvektörleri özdeğerlerine göre büyükten küçüğe doğru sıralanır. İlk ana bileşen, özdeğeri en büyük olan özvektör yönündedir ve değişintinin en büyük olduğu doğrultuyu gösterir. İkinci ana bileşen, özdeğeri en büyük ikinci özvektördür ve bir sonraki en büyük değişintinin doğrultusunu gösterir ve ilk ana bileşenden bağımsızdır.

Kullanılan yöntem tam olarak PCA yöntemi olmamakla beraber temel yaklaşım ve türetme PCA tabanlı olduğu için, tezde bir çok uygulamada PCA olarak adı geçen tanımlama aslında SAM (Sınıflar Arası MetrikMetrik)'dir. Sınıflar arası dağılım ölçütünü kullanmaktaki amaç, bir sınıfın elemanlarının diğer sınıfın ortalamasından uzak olmasını sağlamaktır. SAM, değişinti matrisinin en küçük özdeğerleri yönündeki özellik vektörleri bileşenlerini yok eder. Böylece ortak yönleri yok eder, farklılığı öne çıkarır. Bunun için aşağıda verilen ölçütün maksimizasyonu gerekir.

$$F_{B_{C_i}} = \sum_{C_i} \left\| P_{B_{C_i}} (\vec{a}_x - \vec{a}_{d,ort}) \right\|^2 \quad (2.7)$$

Burada B alt indisi sınıflar arası ölçütünü göstermek için kullanılmaktadır. \vec{a}_x , bilinmeyen vektör olmak üzere $F_{B_{C_i}}$ 'lerden hangisi büyük ise \vec{a}_x , o sınıfa aittir.

Denklem 2.7'de P_B matrisi farklılık altuzayına izdüşüm matrisini göstermektedir. Bu ölçütde sınıflar arası farklılıklar enbüyüklenmeye çalışılmaktadır.

Herhangi bir C sınıfındaki vektörler \vec{a}_i^C ($i=1,2,\dots,m$) ile gösterilmek üzere sınıflar arası değişinti matrisi şu şekilde hesaplanır ;

$$\Phi_B^C = \sum_{i=1}^m (\vec{a}_i^C - \vec{a}_{d,ort}^C)(\vec{a}_i^C - \vec{a}_{d,ort}^C)^T \quad (2.8)$$

Değişinti matrisinin en büyük özdeğerlerine karşılık gelen özvektörlerle izdüşüm matrisi, aşağıdaki şekilde hesaplanır;

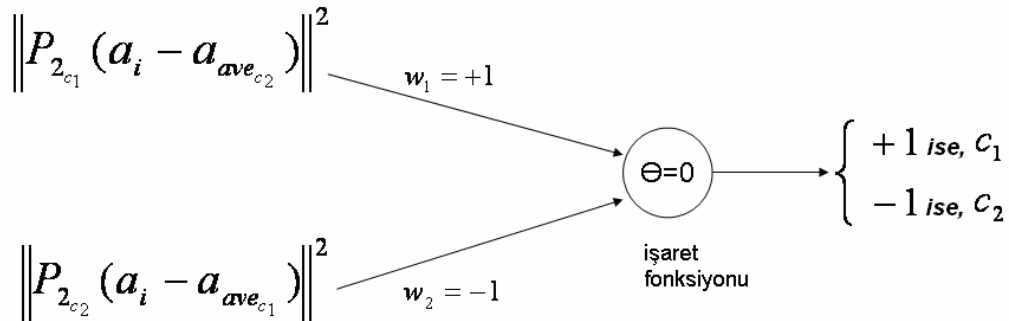
$$\mathfrak{R}^n = \text{span}\left\{ \underbrace{\vec{v}_1, \vec{v}_2, \dots, \vec{v}_{m_2}}_{\substack{\text{en küçük özdeğere} \\ \text{karşı gelen} \\ \text{özvektörler}}}, \underbrace{\vec{v}_{m_2+1}, \dots, \vec{v}_m}_{\substack{\text{en büyük özdeğere} \\ \text{karşı gelen} \\ \text{özvektörler}}} \right\} \quad (2.9)$$

$$P_B = \sum_{i=m_2+1}^m \vec{v}_i \vec{v}_i^T \quad (2.10)$$

Bu matrisi kullanarak, sınıflar arası dağılımların oluşturacağı altuzayda, daha önce verilen metrik ile enbüyükleme yapılır. Karar kuralı ise şöyledir:

$$c_* = \arg \max_k \left\| P_{B C_i} (\vec{a}_x - \vec{a}_{d,ort}) \right\|^2 \quad ; \quad i, d, k = 1, 2 \text{ ve } i \neq d \quad (2.11)$$

Bu ölçütü de yapay sinir ağı şeklinde gösterebiliriz.



Şekil 2.2 SAM sınıflandırıcının iki sınıf probleminde yapay sinir ağı gösterimi

2.3 Fisher LDA

Fisher LDA örüntü tanımada kullanılan bir altuzay sınıflandırma yöntemidir (Bishop, 1996). Doğrusal ayırtaç analizindeki amaç sınıflar arası dağılımı en büyük yaparak farklı grupların örneklerini birbirinden ayırırken, sınıf içi değişintiyi en küçük yapmaktır. LDA'de sınıflandırma ölçütünü oluşturmak için sınıf içi ve sınıflar arası değişinti matrisi kullanılır.

Sınıflar arası dağılım için değişinti matrisi,

$$S_b = \sum_j (\mu_j - \mu_g)(\mu_j - \mu_g)^T \quad (2.12)$$

formülüyle hesaplanır. Burada μ_j , j inci sınıfın ortalamasını; ve μ_g , bütün verilerin ortalamasını göstermektedir.

Fisher LDA sınıflar arası ve sınıf içi değişinti J_r ölçütü ile birleştirilmektedir (Bishop, 1996):

$$J_r = (S_w)^{-1} S_b \quad (2.13)$$

formülüyle hesaplanır. J_r 'nin sıfırdan farklı özdeğerlerine karşılık gelen özvektörler dönüşüm matrisini oluşturur. Çünkü sıfır özdeğere karşılık gelen özvektörler farksızlık altuzayını gererler ve doğrusal bağımlılığı gösterirler. Bu özvektörlerin sınıflandırmada bir katkısı olmaz. A dönüşüm matrisi olmak üzere dönüşüm

$$\vec{y} = A\vec{x} \quad (2.14)$$

ile elde edilir. Fisher LDA'de aşağıda ölçüt enbüyüklenir:

$$J(W) = \frac{A^T S_w A}{A^T S_b A} \quad (2.15)$$

3. SINIFLANDIRICI BİRLEŞTİRME

Sınıflayıcılardan gelen lerin birleştirilmesindeki amaç, her ikisinin ayrı ayrı elde ettikleri başarıdan daha iyi sonuç elde edebilmektir. Teorik olarak böyle bir iyileşmenin beklenmesi doğaldır. Dietterich (2000) sınıflandırıcı birleştirmenin, tek sınıflandırıcıdan niçin daha iyi çalışabileceği hakkında üç ana sebep sunmuştur (Koç M.,2006; Kuncheva, 2004):

- İstatistiksel (Statistical): Bir veri kümesi üzerinde iyi çalışan sınıflandırıcı kümemiz olsun. Sınıflandırıcılardan herhangi birini seçmek yerine bunların ortak kararını almak yanlış sınıflandırıcıyı seçme riskini azaltır.
- Hesapsal (Computational): Bazı eğitim algoritmaları (tepe-inme, rastgele arama) farklı yerel eniyi değere yakınsayabilirler. Fakat bunların farklı şekillerdeki toplamları genel eniyiye daha iyi yakınsayabilir.
- Temsili (Representational): Örüntü tanımada sınıflandırma için kullanılacak sınıfların dağılımı itibariyle bu uzaydaki sınıflandırıcılardan hiçbiri eniyi sınıflandırıcı olmayabilir. Ama bu sınıflandırıcıların birleşimi ile en iyi sınıflandırıcı elde edilebilir.

3.1 Sınıflandırıcı Birleştirme Yöntemleri

Çoklu sınıflandırıcılar için çeşitli yöntemler geliştirilmiştir (Xu, et al., 1992). Bunlardan en iyi bilinen yöntem çoğunluk oyu (Ji, and Ma, 1997) yöntemidir. Diğer yöntemler ise enküçük, enbüyük, ortanca, ortalama (Taniguchi and Tresp, 1997), çarpım (Tax, et al.2000), ağırlıklı ortalama (Heskes, 1997), Borda sayısı, Bayes birleşimi, bulanık integral, bulanık bağlayıcılar, bulanık şablonlar, Dempster Shafer teorisi ve olasılıksal şablonlardır (Koç M., 2006; Kuncheva 2004; Rogova 1994).

- Ortalama

Bu yöntemde tekil sınıflandırıcıların çıkışlarının ortalaması alınır. Sınıflandırmada ise ortalamanın en yüksek olduğu sınıfa yapılır.

$$Q(x) = \arg \max \left(\frac{1}{K} \sum_{i=1}^K y_{ij}(x) \right), (j = 1..N) \quad (3.1.)$$

Burada N sınıf sayısı, $y_{ij}(x)$ ise x girişi için j . sınıfın i . sınıflandırıcıdan çıkış değeridir.

- Ağırlıklı Ortalama

Ağırlıklı ortalama, ortalama yöntemiyle benzerdir, farklı olarak sınıflandırıcıların çıkışları ağırlıklarla çarpılır. Yani

$$Q(x) = \arg \max \left(\frac{1}{K} \sum_{i=1}^K w_i y_{ij}(x) \right), (j = 1..N) \quad (3.2)$$

- Çoğunluk Oyu

Burada doğru sınıf farklı sınıflandırıcılar tarafından en çok seçilen sınıftır. Eğer bütün sınıflandırıcılar farklı sınıfları seçer ise veya eşitlik olursa doğru sınıf olarak çıkış değeri en yüksek olan seçilir.

- Enbüyük

Doğru sınıf olarak en büyük çıkış değeri veren seçilir.

$$Q(x) = \arg \max y_i(x), i = 1..K \quad (3.3.)$$

Burada K sınıflandırıcı sayısı, $y_i(x)$ ise x vektörünün i . sınıflandırıcıdaki çıkış değeridir.

- Borda Sayısı

Herhangi bir j sınıfı için, Borda sayısı, herbir sınıflandırıcı tarafından sıralamada j inci sınıfın ardına yerleştirilen sınıf sayısının toplamıdır. Eğer $B_i(j)$ i . sınıflandırıcı tarafından j 'den sonraya yerleştirilen sınıf sayısı ise j sınıfı için Borda sayısı

$$B(j) = \sum_{i=1}^K B_i(j) \quad (3.4)$$

ile gösterilir. Örüntü en büyük Borda sayısına sahip sınıfa atanır.

- Bayes Birleşimi

c^i , i . sınıfa ait hata matrisi olsun. Bu hata matrisinin elemanları c_{jk}^i 'ler, j . sınıfta oldukları halde k . sınıfta tespit edilen veri sayısı olsun. Bu durumda bir i sınıflandırıcısının j sınıfına ait bir x vektörünü k sınıfına atanmasının koşullu olasılığı

$$P(x \in q_j | \lambda_i(x) = j_i) = \frac{c_{jk}^i}{\sum_{j=1}^N c_{jk}^i} \quad (3.5)$$

Buradan, eğer sınıflandırıcıların birbirinden bağımsız olduğunu varsayarsak, x vektörünün j sınıfına ait güven değeri yaklaşık olarak

$$B(j) = \frac{\prod_{i=1}^K P(x \in q_j | \lambda_i(x) = j_i)}{\sum_{j=1}^N \prod_{i=1}^K P(x \in q_j | \lambda_i(x) = j_i)} \quad (3.6)$$

olarak hesaplanabilir. Sınıflandırma yapılırken x vektörü güven değeri en büyük olan sınıfa atanır.

3.2 Sınıf İçi ve Sınıflar Arası Ölçütlerin Birleştirilmesi

Ölçütlerin bu birleştirilmesindeki amaç, her ikisinin ayrı ayrı elde ettikleri başarıdan daha iyi sonuç elde edebilmektir. Teorik olarak böyle bir iyileşmenin beklenmesi doğaldır. İdealde beklenen, iki ölçütten elde edilen ölçümler birleştirildiğinde yapılacak olan enküçükleme işlemi sonucunda toplam hatanın 0 olmasıdır.

Bu çalışmada, sınıflar-arası ve sınıf-içi değişintiden elde edilen ölçütler Fisher LDA'de kullanılan eniyileme ölçütünden farklı bir ölçüt ile birleştirilmiştir:

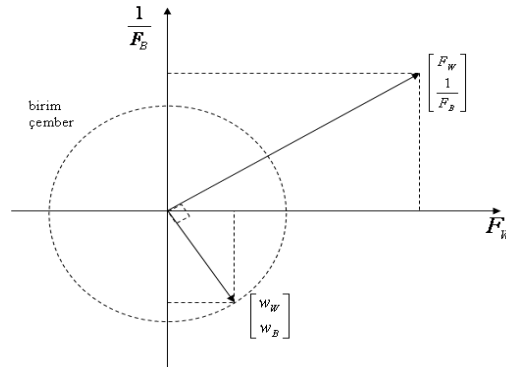
$$F_w w_w + \frac{1}{F_B} w_B = 0 \quad (3.7)$$

Denklem (3.7)'de F_w ve F_B sırasıyla sınıf içi ve sınıflar arası ölçütü; ve w_w ve w_B ise bu ölçütler için ağırlık değerlerini göstermektedir. Ağırlıklar için;

$$\left|w_{1_{c_i}}\right| \leq 1, \left|w_{2_{c_i}}\right| \leq 1 \quad \text{ve} \quad w_{1_{c_1}}^2 + w_{2_{c_1}}^2 = 1, w_{1_{c_2}}^2 + w_{2_{c_2}}^2 = 1 \quad (3.8)$$

kısıtları vardır.

Önerilen sınıflandırıcı için F_W ve F_B normlarıyla ağırlıklar arasındaki ilişki Şekil 3.1 'de gösterilmektedir.



Şekil 3.1 F_W ve F_B normları ve ağırlıklar arasındaki ilişki

Tez çalışmasında, kullanılan ağırlıkların hesaplanmasında kullanılan yöntemler aşağıdaki alt bölümlerde anlatılmaktadır.

3.2.1 Analitik yöntem

Elde edilecek yeni ölçüt de minimizasyon için kullanılacağından sınıflar-arası dağılım için kullanılan ölçüt, birleştirme sırasında $\frac{1}{F_B}$ şeklinde kullanılarak minimizasyon haline getirilmiştir (Koç, 2006). Kullanılan ağırlıklara w_1 ve w_2 diyecek

olursak enküçükleme için $[w_1 \ w_2]$ ile $\left[F_W \ \frac{1}{F_B} \right]$ 'nin ortogonal olması gerekir.

Herhangi bir vektör için yeni metriği yazacak olursak;

$$F = \min \left\{ \left[\left\| \frac{P_{W C_i}^\perp (\vec{x}_i - \vec{x}_{ort, C_i})}{a_i} \right\|^2 w_1 + \frac{1}{\left\| \frac{P_{B C_i} (\vec{x}_i - \vec{x}_{ort, C_{di}})}{b_i} \right\|^2} w_2 \right]^2 \right\} \quad (3.9)$$

olur.

$$F = \min \left\{ (a_i w_1 + b_i w_2)^2 \right\} \quad (3.10)$$

Buradaki karar kriterinden tekrar metriği yazacak olursak, herhangi bir C sınıfındaki N eleman için enküçüklenecek edilecek yeni ölçüt F şöyle olacaktır:

$$F = \min \left\{ \underbrace{\sum_{i=1}^N (a_i w_1 + b_i w_2)^2}_F \right\} = \min \left\{ \sum_{i=1}^N (a_i^2 w_1^2 + 2a_i b_i w_1 w_2 + b_i^2 w_2^2) \right\} \quad (3.11)$$

$$F = \min \left\{ w_1^2 \sum_{i=1}^N a_i^2 + 2w_1 w_2 \sum_{i=1}^N a_i b_i + w_2^2 \sum_{i=1}^N b_i^2 \right\} = \min \{ A w_1^2 + 2C w_1 w_2 + B w_2^2 \} \quad (3.12)$$

$F = \{ A w_1^2 + 2C w_1 w_2 + B w_2^2 \}$ bu eşitlikteki ağırlıkların birim çember üzerinde olması zorlanırsa,

$$F_a = A w_1^2 + 2C w_1 w_2 + B w_2^2 + \lambda (w_1^2 + w_2^2 + 1) \quad (3.13)$$

F_a 'nın w_1 ve w_2 'ye göre kısmi türevleri alarak ilgili denklemler çözülürse;

$$\lambda_{1,2} = \frac{-(A+B) \pm \sqrt{A^2 + B^2 + 4C^2 - 2AB}}{2} \text{ ve}$$

$$w_1^2 = \frac{B + \lambda}{A + B + 2\lambda}, \quad w_2^2 = \frac{A + \lambda}{A + B + 2\lambda} \quad (3.14)$$

bulunacaktır.

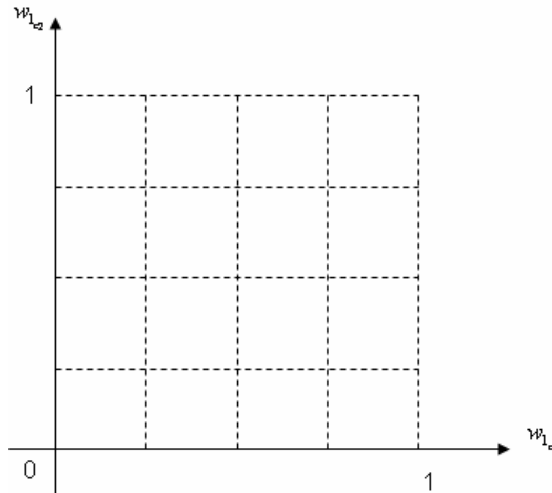
Birleştirilmiş F kriterinde kullanılan ölçüt ve karar kuralı şöyle olacaktır:

$$\begin{aligned} F_{C_1} &= w_{1,C_1} F_{W,C_1} + w_{2,C_1} F_{B,C_1} ; \\ F_{C_2} &= w_{1,C_2} F_{W,C_2} + w_{2,C_2} F_{B,C_2} \end{aligned} \quad c = \operatorname{argmin}(F_{C_1}, F_{C_2}) \quad (3.15)$$

3.2.2 Steepest Descent metodu:

Analitik olarak ağırlıklar bulunabileceği gibi Steepest Descent Methodu ile de $w_{1,C_i}^2 + w_{2,C_i}^2 = 1$ koşulunu sağlamak şartıyla, $w_{1,C_1} - w_{1,C_2}$ düzleminde 0-1 aralığında bir grid üzerinde de bulunabilir.

Arama bölgesi için $w_{1,C_1} - w_{1,C_2}$ düzleminin ilk bölgesinde bir ızgara oluşturulduğunda küçük adımlarla ilerleyerek, gerekli ağırlıklar hesaplanabilir. w_{2,C_1} ve w_{2,C_2} ağırlıkları ise kısıt ilişkilerinden hesaplanabilir.



Şekil 3.2 Ağırlıkların bulunacağı $w_{1,C_1} - w_{1,C_2}$ düzlemi

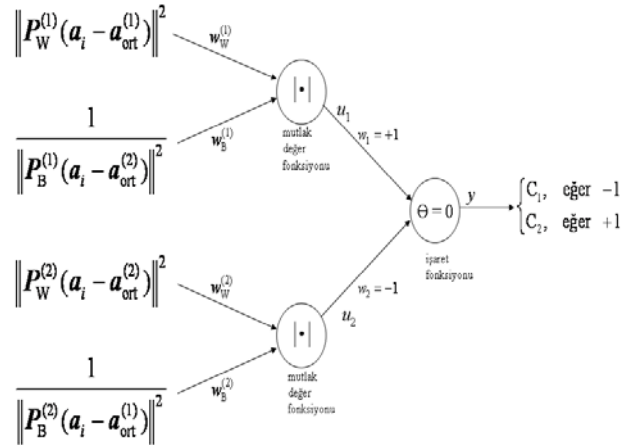
Elbette bulunacak w 'lar hiç bir zaman genel minimumu sağlamayı garanti etmez. Yerel bir minimumu da sağlamış olabilirler. Uygun şartları sağlayacak birden

çok ağırlık bulunabilmektedir. Diğer bir sorun da ne kadar hasasiyetle w ağırlıkları hesaplamak gerektiğidir. Hasassiyet arttıkça algoritmanın işlem süresi artmaktadır.

Burada yapılabilecek diğer kullanışlı bir uygulama da analitik yöntemle bulunacak w 'ların olduğu yerde grid üzerinde gezinmeye başlamak olacaktır. Analitik yöntemle bulunacak w 'ları merkez alacak küçük çerçeveler boyunca daha minimum yapacak w ağırlıkları bulunabilir mi diye aramak daha uygun olacaktır.

3.2.3 Sinir ağı yapısında birleştirme

Sınıf içi ve sınıflar arası dağılımlardan elde edilen ölçütler sinir ağı yapısında birleştirilerek çıkıştaki toplam hatanın minimum yapılması amaçlanmış ve böylece birleştirilmiş ölçütün her ikisinden de daha iyi tanıma başarısı elde etmesi amaçlanmıştır. Şekil 3.3'te önerilen sınıflandırıcının iki-sınıf problemi için sinir ağı şeklinde gösterimi verilmiştir. Burada sinir ağının girişleri, \mathbf{a}_x vektörünün birinci ve ikinci sınıfın fark ve farksızlık altuzaylarına izdüşümlerinin normlarıdır. Bu girişlerden farksızlık altuzayına izdüşümü alınanlar $w_w^{(1)}$ ve $w_w^{(2)}$ ağırlıklarıyla, fark altuzayına izdüşümü alınanlar ise $w_B^{(1)}$ ve $w_B^{(2)}$ ile çarpılır. Birleştirilen ölçüt enküçültme ölçütü olduğundan sinir ağının ilk katmanından elde edilen u_1 ve u_2 çıkışlarından en küçük olana göre çıkış vektörünün sınıflandırılması yapılır. Sinir ağının çıkışı sıfırdan küçük ise \mathbf{a}_x birinci sınıfa, sıfırdan büyük ise ikinci sınıfa atanır.



Şekil 3.3 OVY ve SAM ölçütlerini birleştiren sınıflandırıcının yapay sinir ağı şekli

Denklem (3.16)'da, w_W sıfırdan büyük seçildiğinden w_B sıfırdan küçük olmak zorundadır. Şekil 3.3 'te u_1 ve u_2 çıkışlarının sıfırdan küçük olmaları ölçütün çalışmasını olumsuz etkileyebileceğinden sinir ağının ilk katmanına mutlak değer fonksiyonu eklenmiştir.

Şekil 3.3'te görüleceği gibi, C_1 sınıfının bir elemanı C_2 'ye sınıflandırılırsa yapay sinir ağı $+1$ çıkışı verecektir. Bu durumda C_1 sınıfının elemanlarının yanlış sınıflandırılmasından dolayı oluşacak toplam hata

$$\mathcal{E}_1 = \frac{1}{2} \sum_{i=1}^{N_1} \left[\operatorname{sgn} \left\{ \left\| P_W^{(1)}(a_i - a_{ort}^{(1)}) \right\|^2 w_W^{(1)} + \frac{1}{\|P_B^{(1)}(a_i - a_{ort}^{(2)})\|^2} w_B^{(1)} \right\} - \left\| P_W^{(2)}(a_i - a_{ort}^{(2)}) \right\|^2 w_W^{(2)} + \frac{1}{\|P_B^{(2)}(a_i - a_{ort}^{(1)})\|^2} w_B^{(2)} \right\} + 1 \right] \quad (3.16)$$

ifadesiyle hesaplanır. Benzer şekilde C_2 sınıfının elemanlarının yanlış sınıflandırılmasından dolayı oluşacak toplam hata aşağıdaki ifadeden bulunur:

$$\mathcal{E}_2 = -\frac{1}{2} \sum_{i=1}^{N_2} \left[\operatorname{sgn} \left\{ \left\| P_W^{(1)}(a_i - a_{ort}^{(1)}) \right\|^2 w_W^{(1)} + \frac{1}{\|P_B^{(1)}(a_i - a_{ort}^{(2)})\|^2} w_B^{(1)} \right\} - \left\| P_W^{(2)}(a_i - a_{ort}^{(2)}) \right\|^2 w_W^{(2)} + \frac{1}{\|P_B^{(2)}(a_i - a_{ort}^{(1)})\|^2} w_B^{(2)} \right\} - 1 \right] \quad (3.17)$$

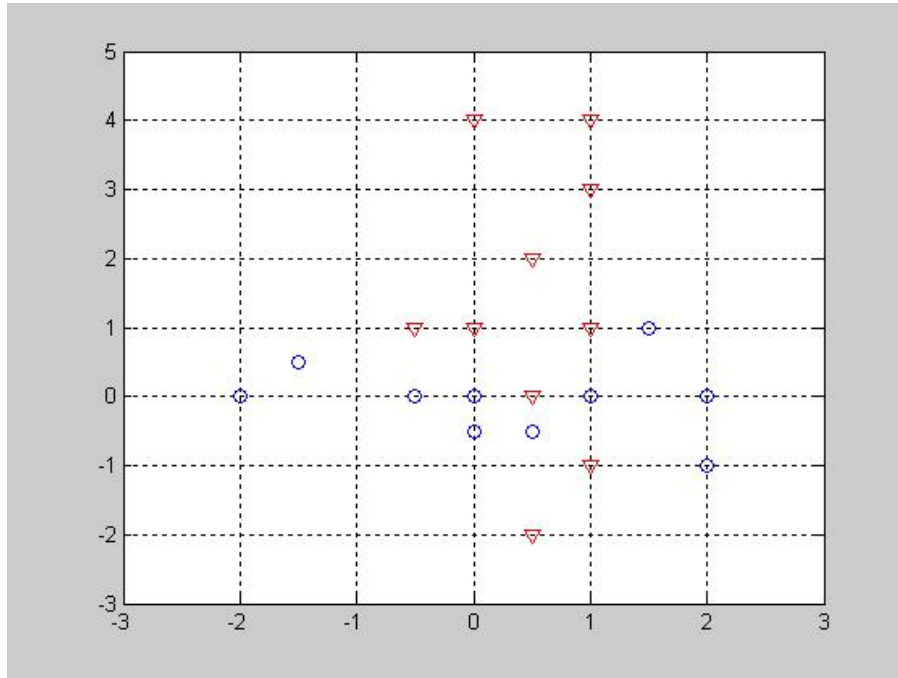
Denklem (3.16) ve (3.17)'de verilen sınıflama hatalarını enküçükleyen ağırlıklar, verilen kısıtlar altında belirli adımlarla $w_W^{(1)} - w_W^{(2)}$ düzlemi taranarak bulunabilir.

4. DENEYSEL ÇALIŞMALAR

4.1 Yapay Veri-1

Tez çalışmasında geliştirilen yöntem, önce yapay olarak oluşturulmuş iki boyutlu düzlemdeki veri setlerinde denenmiştir. Bu veri setleri tamamen rastgele oluşturulmuş herhangi bir kural belirlenmemiştir. Sadece, sınıflandırma sonrası grafik olarak da görebilmek için iki boyutlu seçilmiştir.

Bu veri kümesinde her bir sınıfta 10 eleman bulunmaktadır. C_1 ve C_2 sınıflarının dağılımı aşağıdaki şekilde verilmiştir.



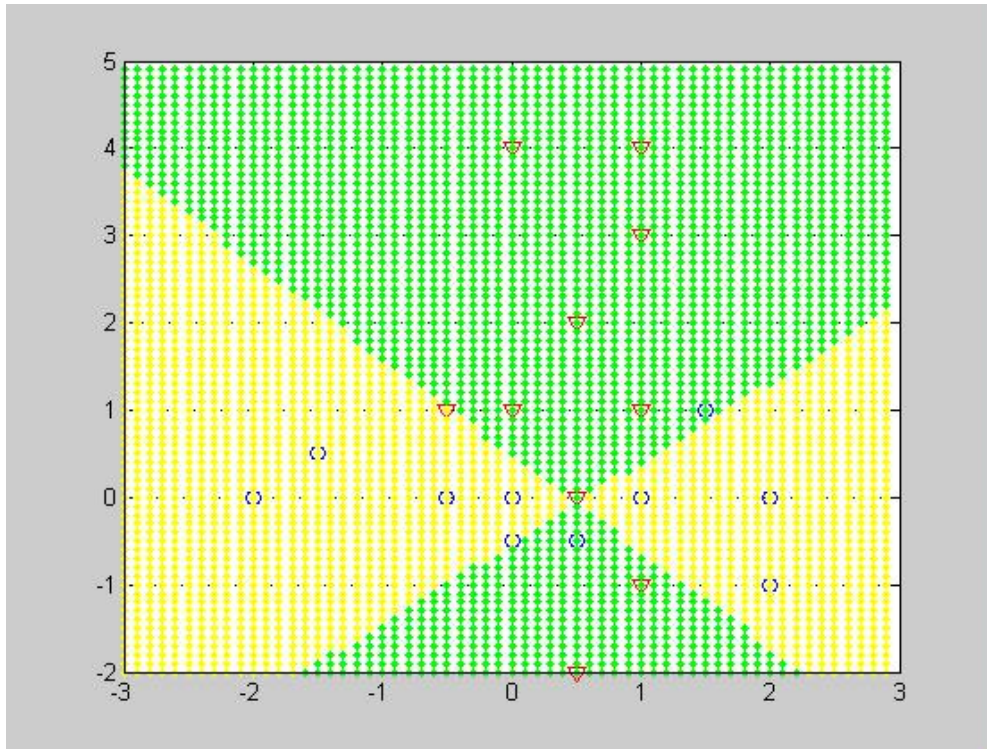
Şekil 4.1 Yapay veri-1 için C_1 ve C_2 sınıflarının x-y düzleminde dağılımı

4.1.1 OVY ve SAM için yapay veri-1 Sonuçları

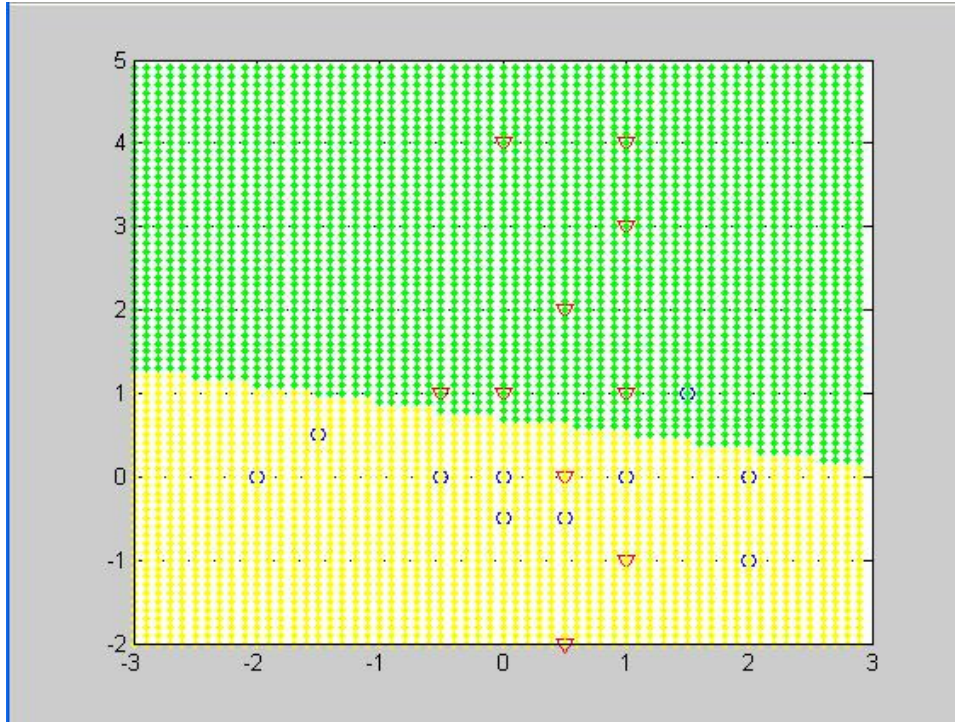
OVY ve SAM'da iki boyutlu uzayda tek özdeğer kullanarak aşağıdaki ayırma bölgeleri elde edilmiştir.

Sınıflara ait örneklere gözle de bakıldığında OVY yöntemi için iki sınıf için toplam 20 örnekten 3 tanesinin yanlış ayırma bölgesinde kaldığı yani hatalı tanıma oranının % 15, başarılı tanıma oranının ise %85 olduğu görülmektedir.

Aynı şekilde SAM algoritması yapay veriye uygulandığında elde edilen ayırıcı bölgelerde, toplam 20 örnekten 4 tane hatalı sınıflama olduğu görülür. Buradaki yanlış tanıma % 20, başarılı tanıma ise % 80 olmuştur.



Şekil 4.2 Yapay veri-1 için OVY ile sınıflandırma

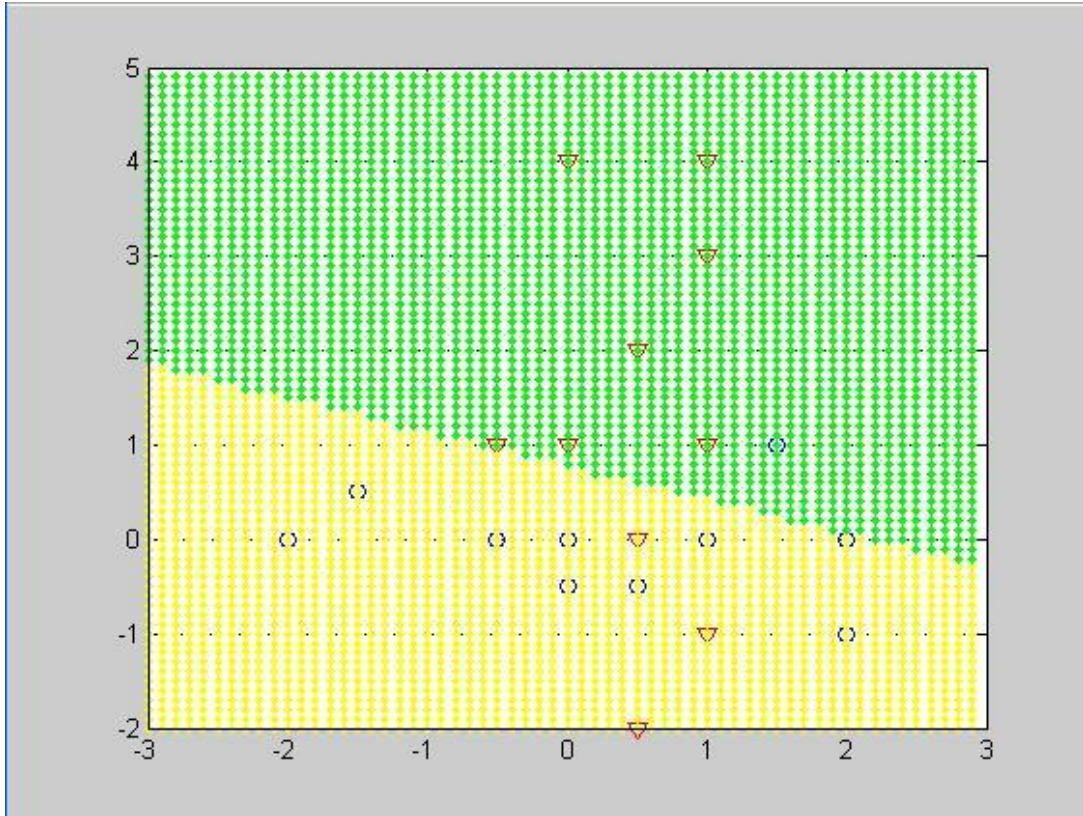


Şekil 4.3 Yapay veri-1 için SAM ile sınıflandırma

4.1.2 Yapay veri-1 için Fisher_LDA uygulanması

$F_{\text{FISHER}} = \text{Tr} \{ (W^T S_W W)^{-1} (W^T S_B W) \}$ kriteri $S_w^{-1} S_B$ nin ait olduğu alt uzayda en büyük özdeğerleri ile enbüyüklenmiştir. W matrisleri izdüşüm matrisini, S_w sınıf-içi dağılım matrisini ve S_B sınıflar-arası dağılım matrisini göstermektedir (Bishop,1996).

Bu uygulama ile grafiksel olarak kriterin ayırıcılığı aşağıda verilmiştir. Algoritma, %80 başarılı tanıma sonucunu vermiştir. Fisher'in kullandığı Doğrusal Ayırma Analizi(LDA) iki boyutlu uzayda tek boyutlu bir ayırıcı elde edebilir ki bu da elde edilen doğrudur. Bu şekilde dağılmış bir veri kümesinin bir doğru ile ayrılması mümkün değildir.



Şekil 4.4 Yapay veri-1 için LDA ile sınıflandırma

4.1.3 Yapay veri-1 için iki metriğin birleştirilmesi:

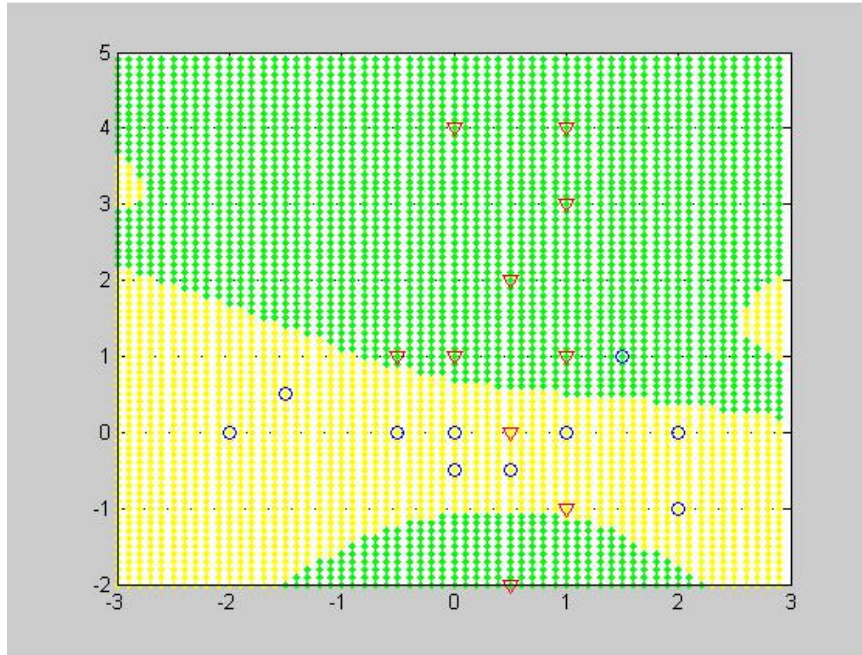
4.1.3.1 Ağırlıksız birleştirme

Sınıf-içi ve sınıflar-arası dağılımlardan elde edilen ölçütler ağırlıklar kullanmadan birleştirilirse ya da başka bir deyişle ağırlıkların etkisi ihmal edilip hepsi 1 alınarak birleştirilirse karar kuralı aşağıdaki gibi olacaktır.

$$F_{C_1} = F_{W,C_1} + F_{B,C_1};$$

$$F_{C_2} = F_{W,C_2} + F_{B,C_2}, \quad c = \text{argmin}(F_{C_1}, F_{C_2}) \quad (4.1)$$

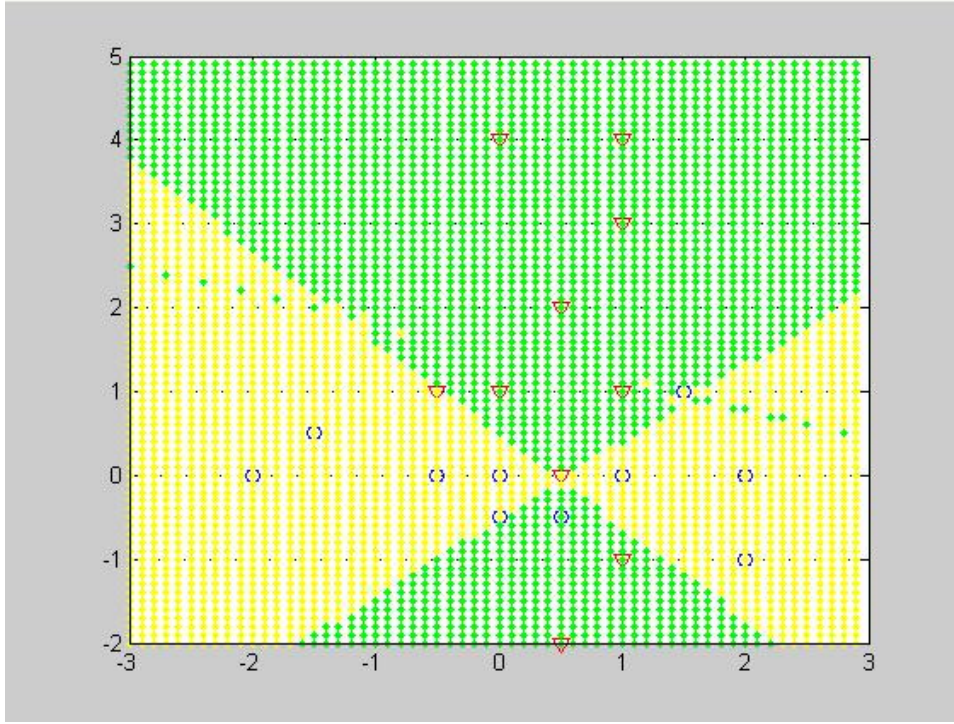
Ağırlıklar 1 alınarak birleştirildiğinde elde edilen ayırıcı bölgeler şekil 4.5'te verilmiştir. Burada da görüleceği gibi toplam 20 örnekten 3 tanesi hatalı sınıfa atanmış olup tanıma başarısı % 85'tir. Ancak ayırıcı bölgelerin sınıfları verimli bir şekilde ayırmadığı da görülmektedir.



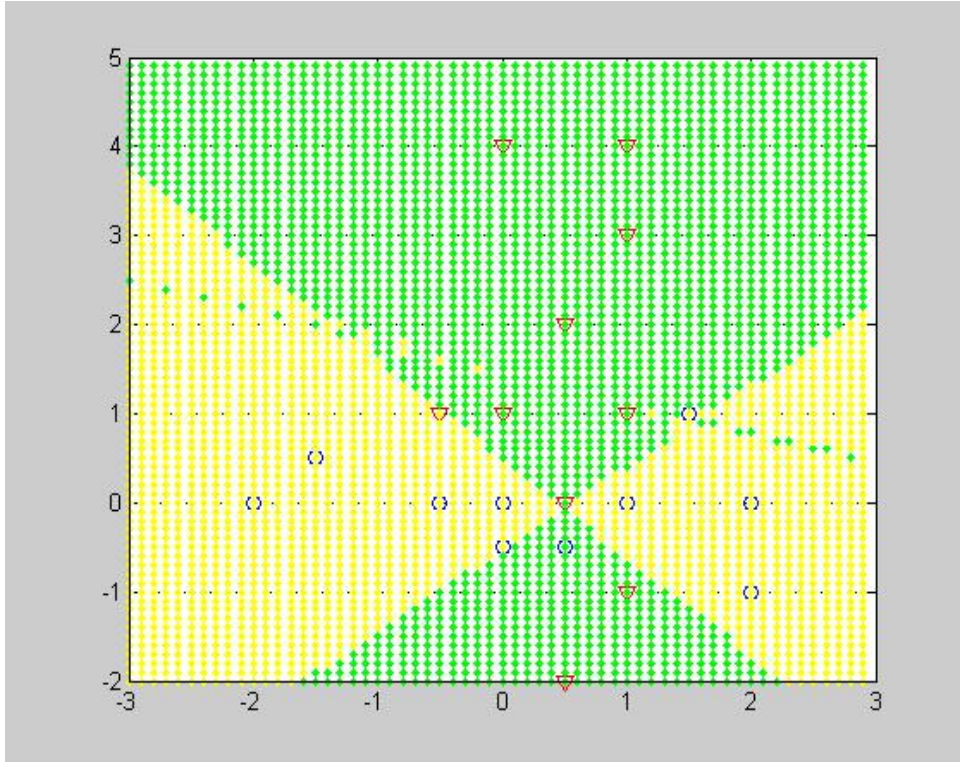
Şekil 4.5 Yapay veri-1 için OVY ve SAM ölçütleri ile ağırlıksız birleştirme

4.1.3.2 Ağırlıklar kullanarak birleştirme

Daha sonraki çalışmada, sınıflara ait veriler kullanılarak Şekil 3.3'te verilen yapay sinir ağı yapısı için ağırlıklar, analitik yöntemle $w_w^{C_1} = 0.999976$, $w_B^{C_1} = -0.000023$ ve $w_w^{C_2} = 0.999997$, $w_B^{C_{21}} = -0.0024$ olarak hesaplanmıştır. Bu ağırlıklar kullanarak elde edilen tanıma başarısı % 85'dir. Sınıflandırma bölgeleri de aşağıdaki şekilde gösterilmiştir. Bu ağırlıklar çıkış noktası alınarak, 0.0001 adımla grid üzerinde ilerlenecek olursa yeni bulunacak ağırlıklar $w_w^{C_1} = 1$, $w_B^{C_1} = 0$ ve $w_w^{C_2} = 0.999992$, $w_B^{C_{21}} = -0.0039$ olacaktır. Bu ağırlıklar kullanarak elde edilen tanıma başarısı ise %90 olmuştur. Her birinde 10 eleman bulunan iki sınıf için tanıma başarımlarının bu yüzdelerde verilmesi çok da anlamlı değildir. Böyle bir çalışma, eleman sayısının azlığı sebebiyle işlemlerin elle sağlanmasının yapılmasına olanak vermektedir. Grafik olarak da, geliştirilen yöntem ve diğer yöntemlerin karşılaştırılabilmesini sağlamaktadır.



Şekil 4.6 Yapay veri-1 için OVYve SAM'den elde edilen ölçütlerin birleştirilmesi



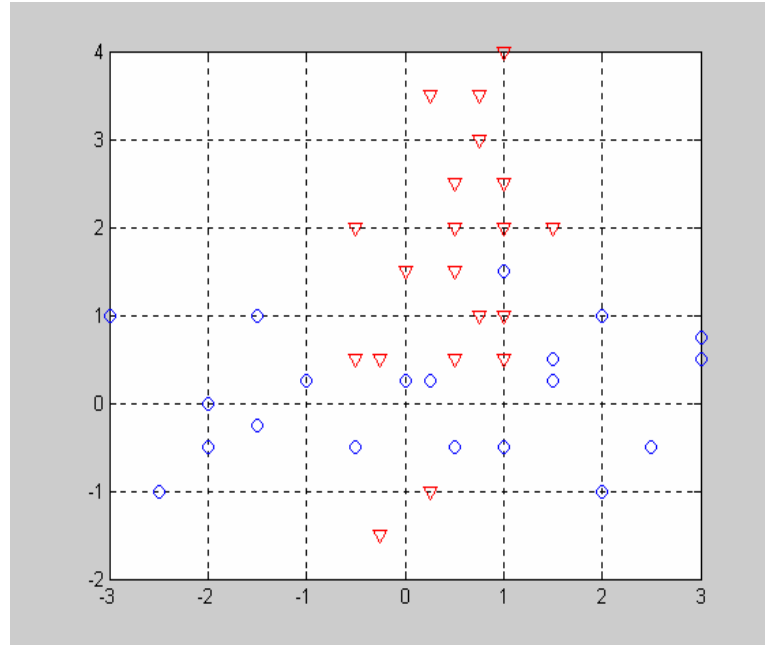
Şekil 4.7 Yapay veri-1 için grid üzerindeki ağırlıklarla ölçütlerin birleştirilmesi

Çizelge 4.1 Yapay veri-1 için tanıma başarımları

Sınıflandırıcı	Ortalama Tanıma Oranları(%)
F_w	85
F_B	80
Fisher-LDA	80
Birleştirilmiş F_B ve F_w	85
Grid üzerindeki ağırlıklar ile birleştirilmiş F_B ve F_w	90

4.2 Yapay Veri-2 :

C_1 ve C_2 sınıfları aşağıdaki şekilde gösterildiği gibi dağılmaktadır. Her sınıfta 20 eleman bulunmaktadır. Hem eğitim hem de test aşamasında aynı veriler kullanılmıştır.

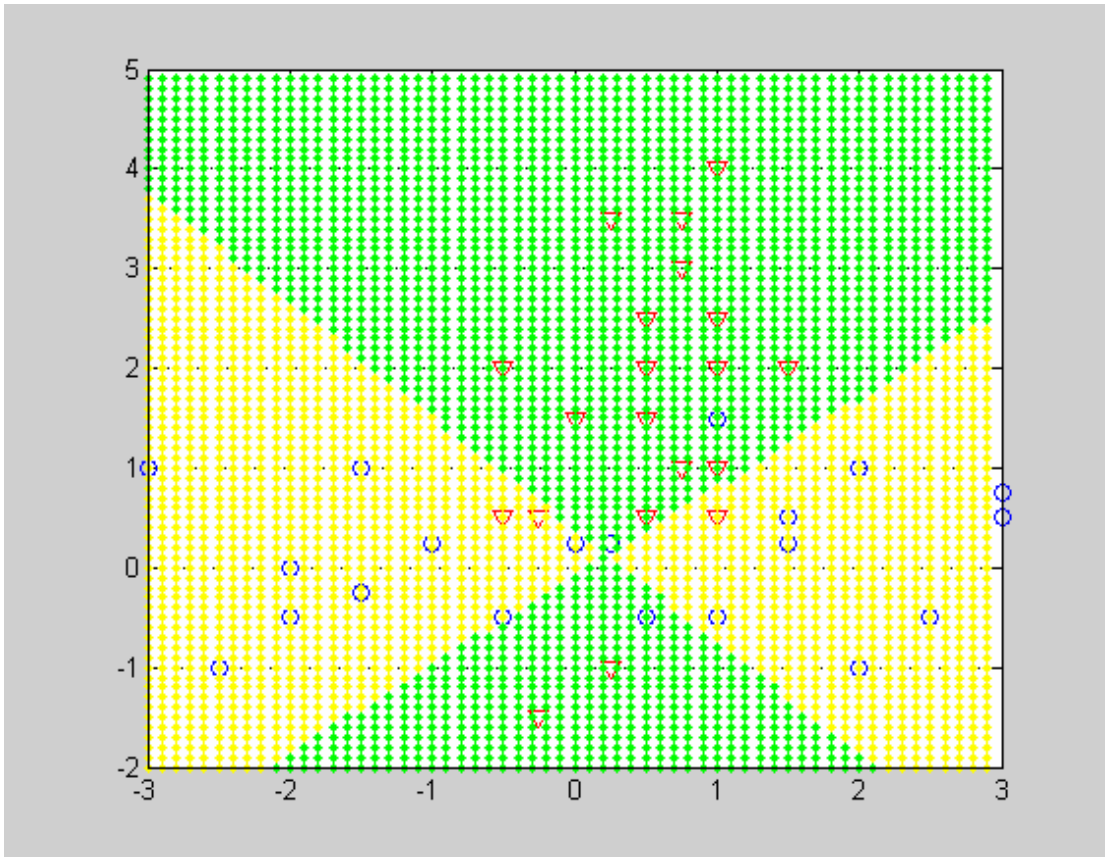
Şekil 4.8 Yapay veri-2 için C_1 ve C_2 sınıflarını x-y düzlemindeki dağılımlar

4.2.1 OVY ve SAM algoritmalarının yapay veri-2'ye uygulanması

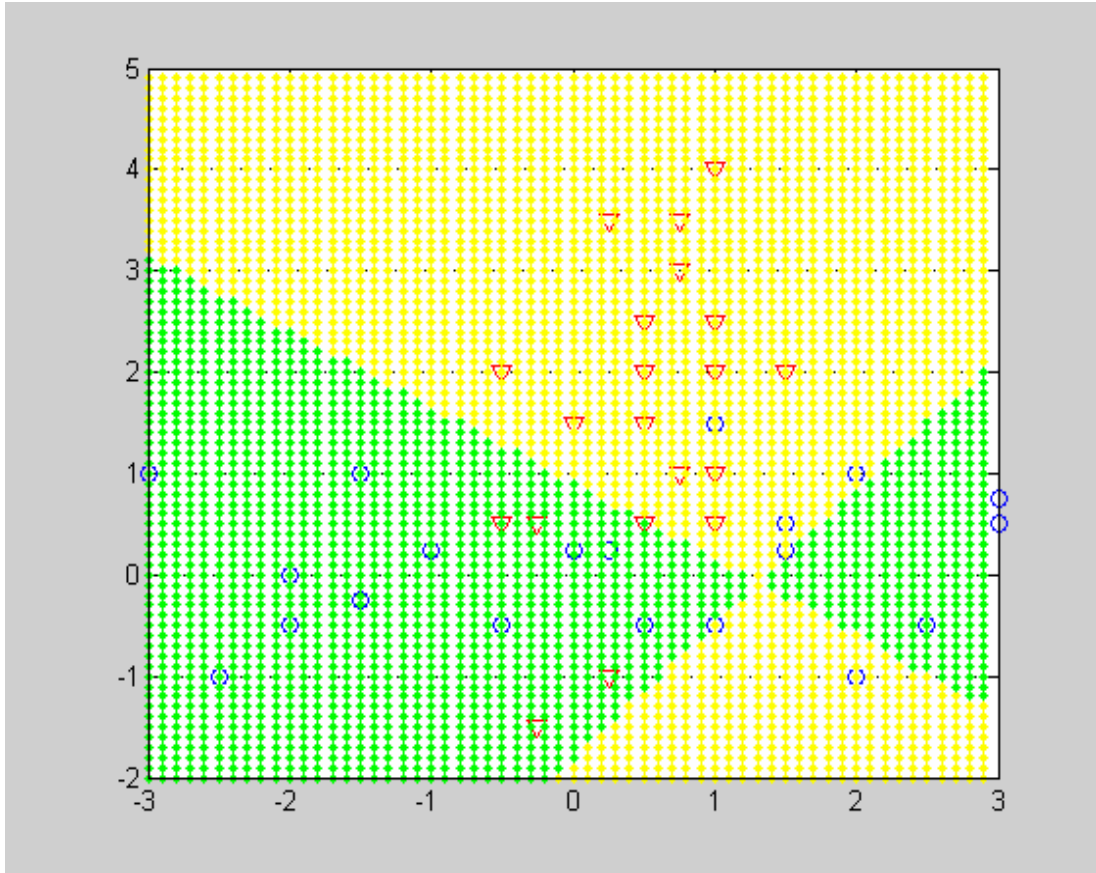
OVY ve SAM 'da iki boyutlu uzayda tek özdeğer kullanarak aşağıdaki ayırma bölgeleri elde edilmiştir.

Sınıflara ait örnekler ayrılan bölgelerde kalmış mı yoksa yanlış bölge sınıflanmış mı diye bakılacak olursa gözle de OVY yönteminde iki sınıf için, toplam 40 örnekten 6 tanesinin yanlış ayırma bölgesinde kaldığı yani hatalı tanıma oranının % 15, başarılı tanıma oranının ise % 85 olduğu görülmektedir.

Aynı şekilde SAM algoritması yapay veriye uygulandığında elde edilen ayırıcı bölgelerdeki yanlış sınıflanan elemanlara bakıldığında her iki sınıf için toplamda 11 olduğu görülür. Buradaki yanlış tanıma % 27.5, başarılı tanıma ise % 72.5 olmuştur.



Şekil 4.9 Yapay veri-2'in OVY ile sınıflandırılması



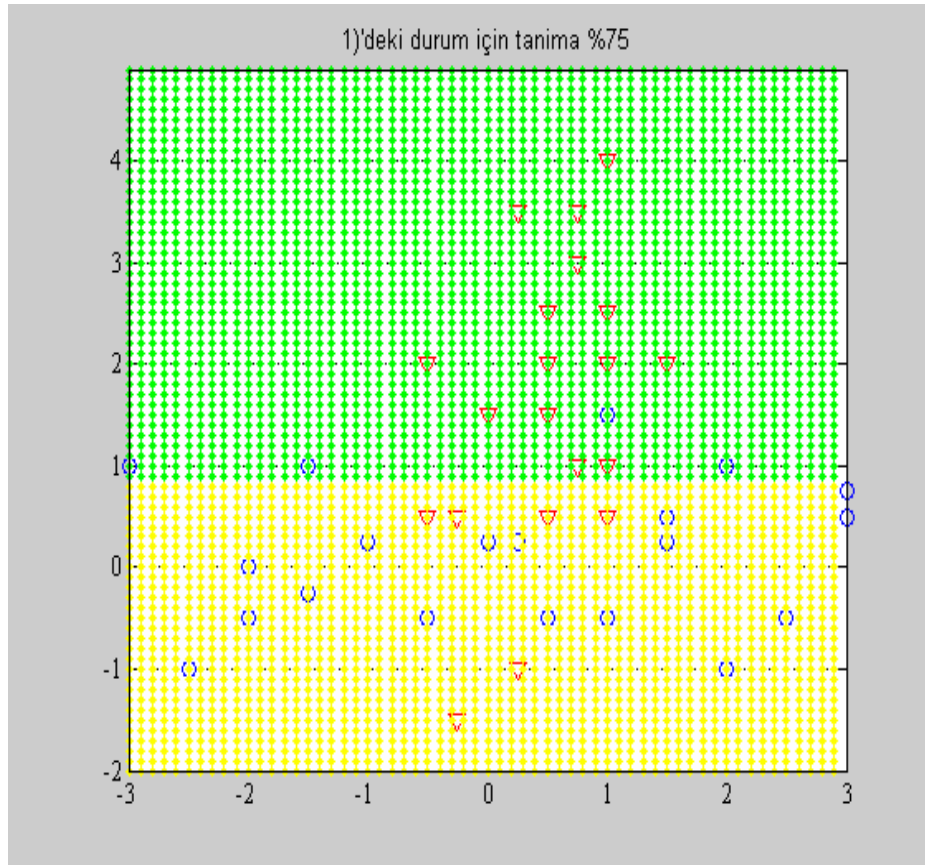
Şekil 4.10 Yapay veri-2'nin SAM ile sınıflandırılması

4.2.2 Fisher LDA için yapay veri-2 sonuçları

4.2.2.1 Uygulama 1:

$F_{\text{FISHER}} = \text{Tr} \{ (W^T S_W W)^{-1} (W^T S_B W) \}$ kriteri $S_w^{-1} S_B$ nin ait olduğu alt uzayda en büyük özdeğerleri ile enbüyüklenmiştir. W matrisleri izdüşüm matrisini göstermektedir (Bishop,1996).

Bu uygulama ile grafiksel olarak kriterin ayırıcılığı aşağıda verilmiştir. Bu ayırıcı bölge, matematik olarak doğrusal ayırıcı doğru denklemi bulunarak çizildiğinde de aynı bölgeyi vermektedir. Algoritma, 10 yanlış sınıflanan örnek ile % 75 başarılı tanıma sonucunu vermiştir. Fisher'in kullandığı Doğrusal Ayırma Analizi (LDA) iki boyutlu uzayda tek boyutlu bir ayırıcı elde edebilir ki bu da elde edilen doğrudur. Bu şekilde dağılmış bir veri kümesinin bir doğru ile ayrılması mümkün değildir.



Şekil 4.11 Fisher LDA için yapay verinin sınıflandırılması

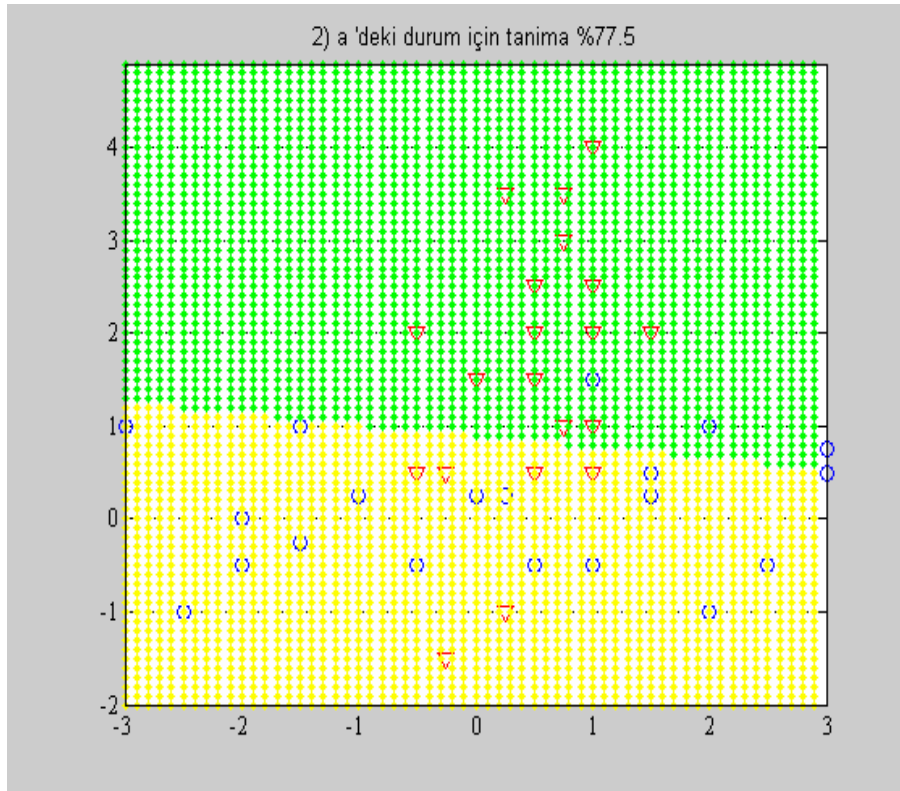
4.2.2.2 Uygulama 2(a)

Daha önceki çalışmalarımızla paralellik kurabilmek için Fisher kriteri, $S_B^{-1} S_W$ ifadesi C_1 ve C_2 sınıfı için tek altuzay kullanmak yerine ayrı ayrı alt uzaylar olarak hesaplanmıştır. Her bir sınıf için Fisher kriteri ait olduğu alt uzaylarda en küçük özdeğerleri ile minimize edilmiştir. Tanıma kriteri her iki sınıf için ayrı ayrı yazılırsa;

$$F_{C_1} = \frac{\|P_{C_1}(\vec{a}_x - \vec{a}_{C_1,ort})\|^2}{\|P_{C_1}(\vec{a}_x - \vec{a}_{C_2,ort})\|^2}, \quad F_{C_2} = \frac{\|P_{C_2}(\vec{a}_x - \vec{a}_{C_2,ort})\|^2}{\|P_{C_2}(\vec{a}_x - \vec{a}_{C_1,ort})\|^2} \quad (4.2)$$

P_{C_1} ; C_1 sınıfı için $S_B^{-1} S_W$ ifadesinden elde edilen tek alt uzay üzerine izdüşüm matrisi

P_{C_2} ; C_2 sınıfı için $S_B^{-1} S_W$ ifadesinden elde edilen tek alt uzay üzerine izdüşüm matrisi

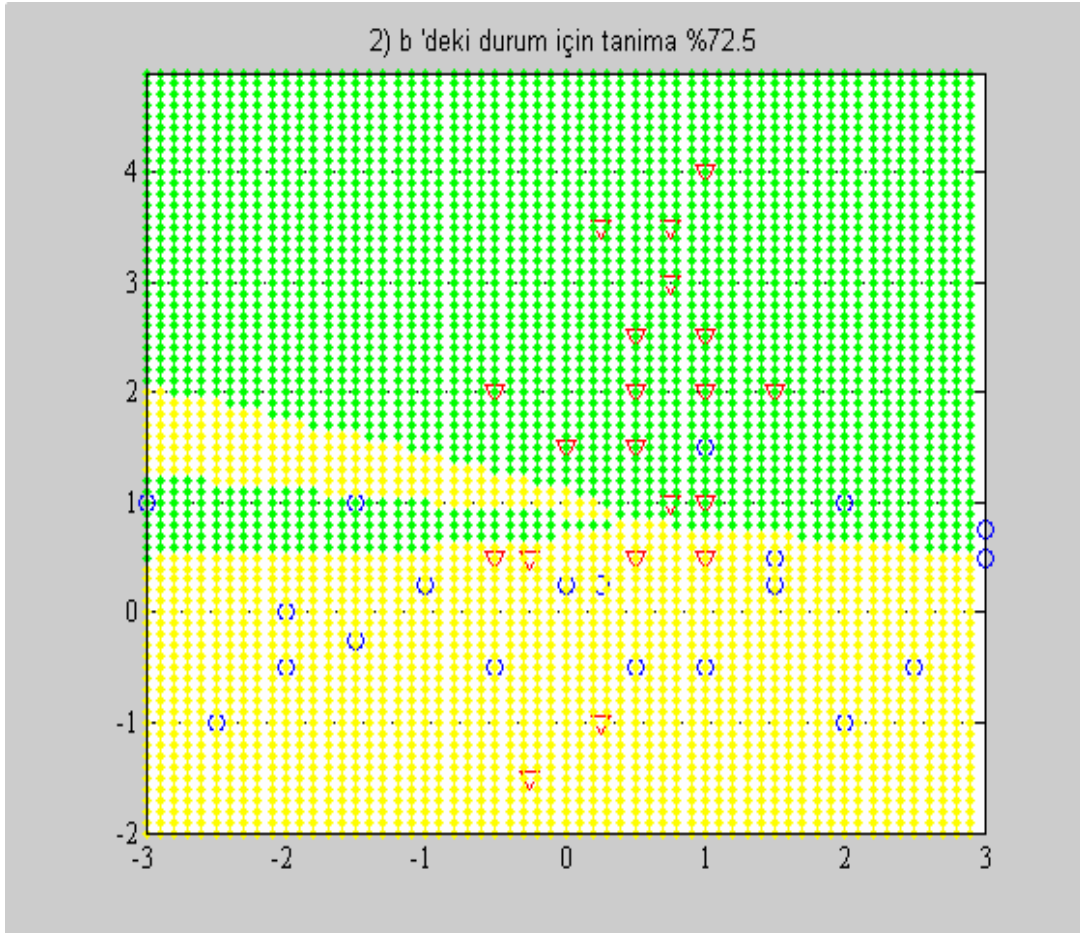


Şekil 4.12 Uygulama 2-a için yapay veri-2'nin sınıflandırılması

4.2.2.3 Uygulama 2 (b)

Sınıflar-arası değişinti matrisinin hesabında diğer sınıfın ortalamasından uzaklaştırmak yerine iki sınıf ortalamasından (\mathbf{a}_{ort}) uzaklaştırılırsa altuzaylar ve izdüşüm matrisleri değişecektir (P_1 ve P_2). Tanıma kriteri her iki sınıf için ;

$$F_{C_1} = \frac{\|P_{C_1}(\bar{\mathbf{a}}_x - \bar{\mathbf{a}}_{C_1,ort})\|^2}{\|P_{C_1}(\bar{\mathbf{a}}_x - \bar{\mathbf{a}}_{ort})\|^2}, \quad F_{C_2} = \frac{\|P_{C_2}(\bar{\mathbf{a}}_x - \bar{\mathbf{a}}_{C_2,ort})\|^2}{\|P_{C_2}(\bar{\mathbf{a}}_x - \bar{\mathbf{a}}_{ort})\|^2} \quad (4.3)$$



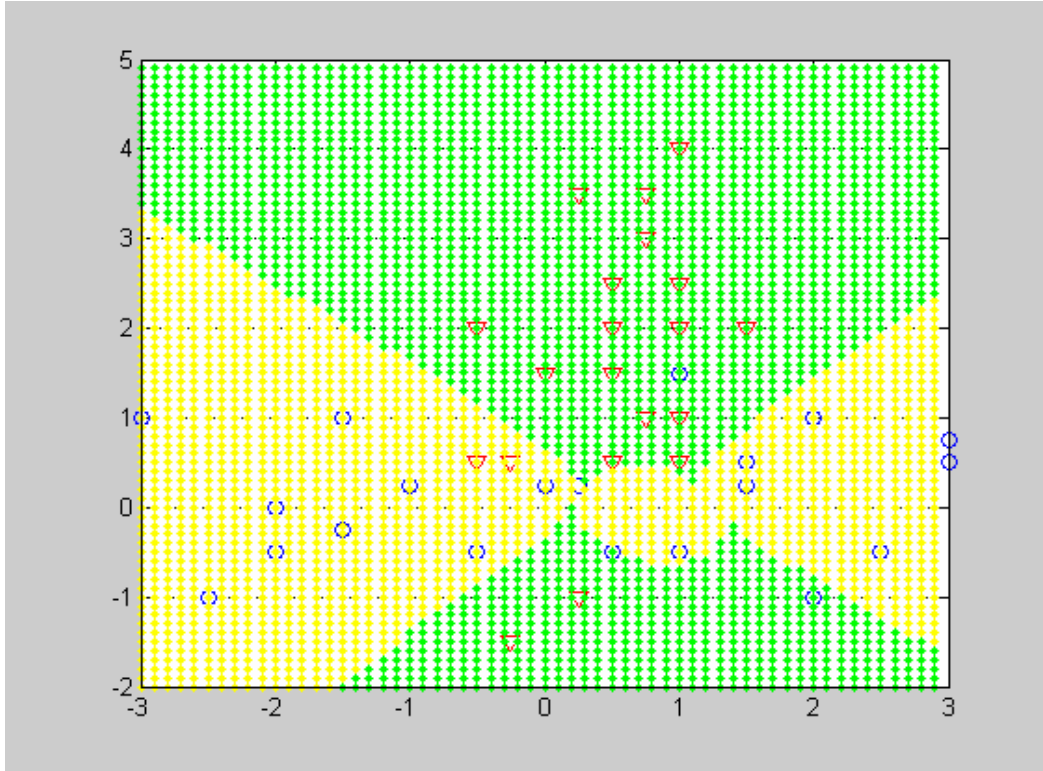
Şekil 4.13 Uygulama 2-b'ye göre yapay veri-2'nin sınıflandırılması

4.2.3 İki metriğin birleştirilmesi

4.2.3.1 Ağırlıksız birleştirme (Fisher Oran)

Sınıf içi ve sınıflar arası dağılımlar kullanılarak elde edilen F_w ve F_B ölçütleri Fisher'dekine benzer bir şekilde sadece oranlanarak birleştirilmeye çalışılırsa yeni bulunacak ölçüt için karar kriteri her iki sınıf için ayrı ayrı 4.3'te olduğu gibidir.

Kriterde tanıma oranı % 85 bulunmuştur. Bu sonuç OVY sonucu ile aynıdır çünkü paydadın gelen SAM katkısı çok çok küçük olmaktadır.



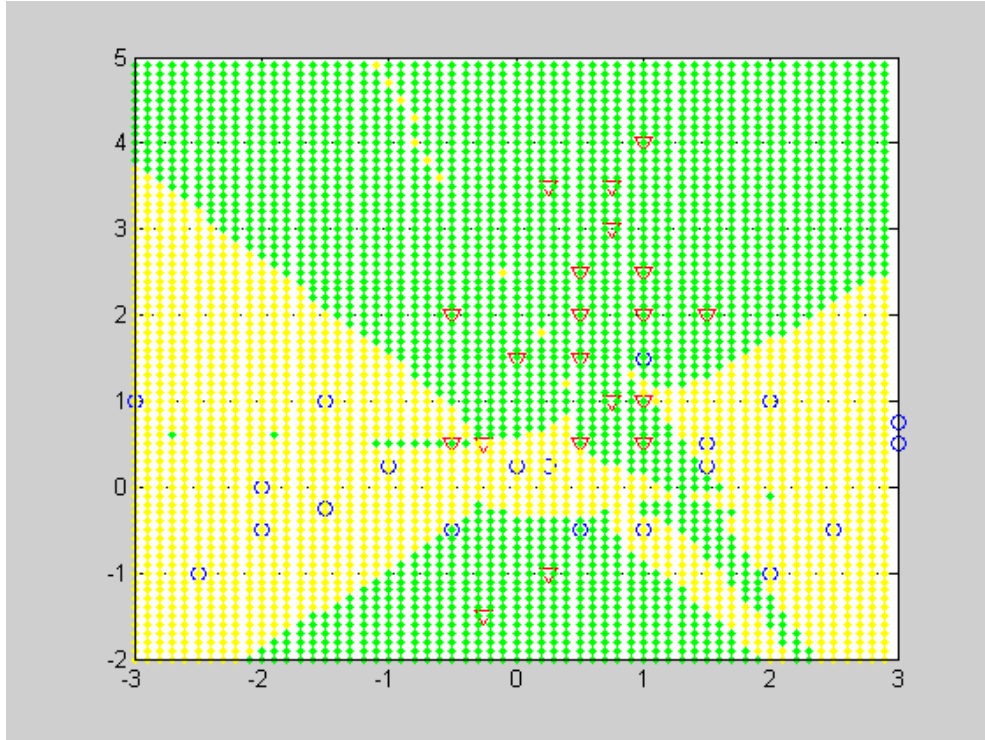
Şekil 4.14 Ağırlıksız birleştirme ile yapay veri-2'nin sınıflandırılması

4.2.3.2 Ağırlıklar kullanarak birleştirme (Birleştirilmiş OVY ve SAM):

Tez çalışmasında, sınıf-içi ve sınıflar-arası dağılımlardan elde edilen ölçütler Fisher_LDA'ye benzeyen ancak ondan farklı bir eniyileme ölçütü ile birleştirilmiştir. Birleştirmede w_w ve w_B ağırlıkları kullanılmıştır.

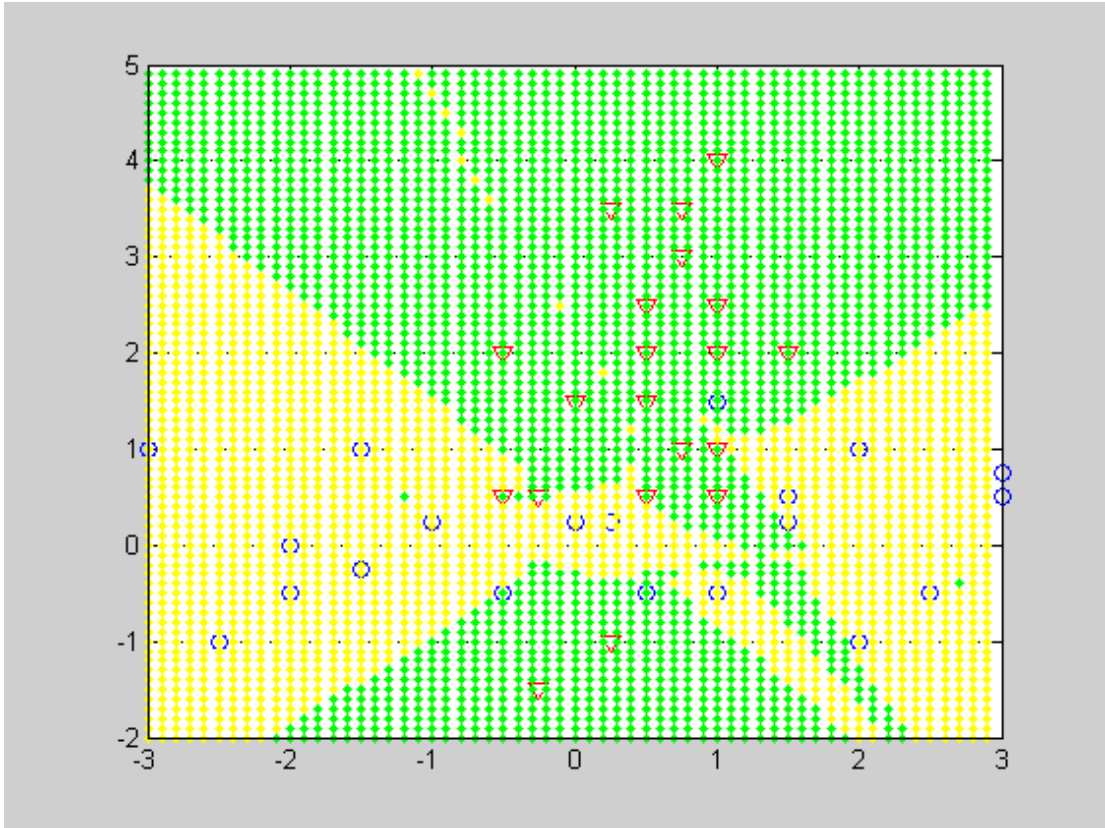
$$F_{C_i} = F_w^{C_i} w_w^{C_i} + \frac{1}{F_B^{C_i}} w_B^{C_i} \quad (4.4)$$

Sınıflara ait veriler kullanılarak Şekil 3.3'te verilen yapay sinir ağı yapısı için ağırlıklar analitik yöntemle $w_w^{(1)} = 0.914$, $w_w^{(2)} = 0.999$, $w_B^{(1)} = -0.40571$ ve $w_B^{(2)} = -0.04471$ olarak hesaplanmıştır.



Şekil 4.15 Yapay veri-2 için ağırlıklar kullanarak birleştirme

Aynı çalışma da ağırlıkları belirlemek için bölüm 3.2.2’de anlatılan grid yöntemi de kullanılmıştır. Grid 0-1 aralığında 0.00001 hassaslıkta adım adım taratılmıştır. Ancak başlangıç noktası olarak, analitik olarak belirlenen w ağırlıklarından yola çıkılmıştır. Adım adım ilerlendiğinde, $w_w^{C_1}=0.999$, $w_B^{C_1}=-0.0282$, $w_w^{C_2}=0.9951$, $w_B^{C_2}=-0.0988$ ağırlıklarında en iyi tanıma başarımı elde edildiği tesbit edilmiştir. Bulunan bu ağırlıklarla tanıma başarımına bakıldığında % 92.5’tur.



Şekil 4.16 Yapay veri-2 için grid üzerindeki ağırlıkları kullanarak birleştirme

Çizelge 4.2’de sınıf içi, sınıflar arası, Fisher-LDA ve önerilen sınıflandırıcının sınıflandırma başarımları verilmiştir. Çizelge 4.2’den görüldüğü gibi yapay veri üzerinde, önerilen sınıflandırıcı ile diğer sınıflandırıcılara göre daha iyi sınıflandırma başarımları elde edilmiştir. Özellikle grid üzerinde ağırlık bulma yöntemiyle, analitik olarak ağırlık bulma yönteminden daha iyi sonuçlar elde edilmiştir.

Çizelge 4.2 Yapay veri-2 için tanıma başarımları

Sınıflandırıcı	Ortalama Tanıma Oranları(%)
F_B	72.5
F_W	85
Fisher-LDA	75
Birleştirilmiş F_B ve F_W	90
Grid üzerindeki ağırlıklar ile birleştirilmiş F_B ve F_W	95

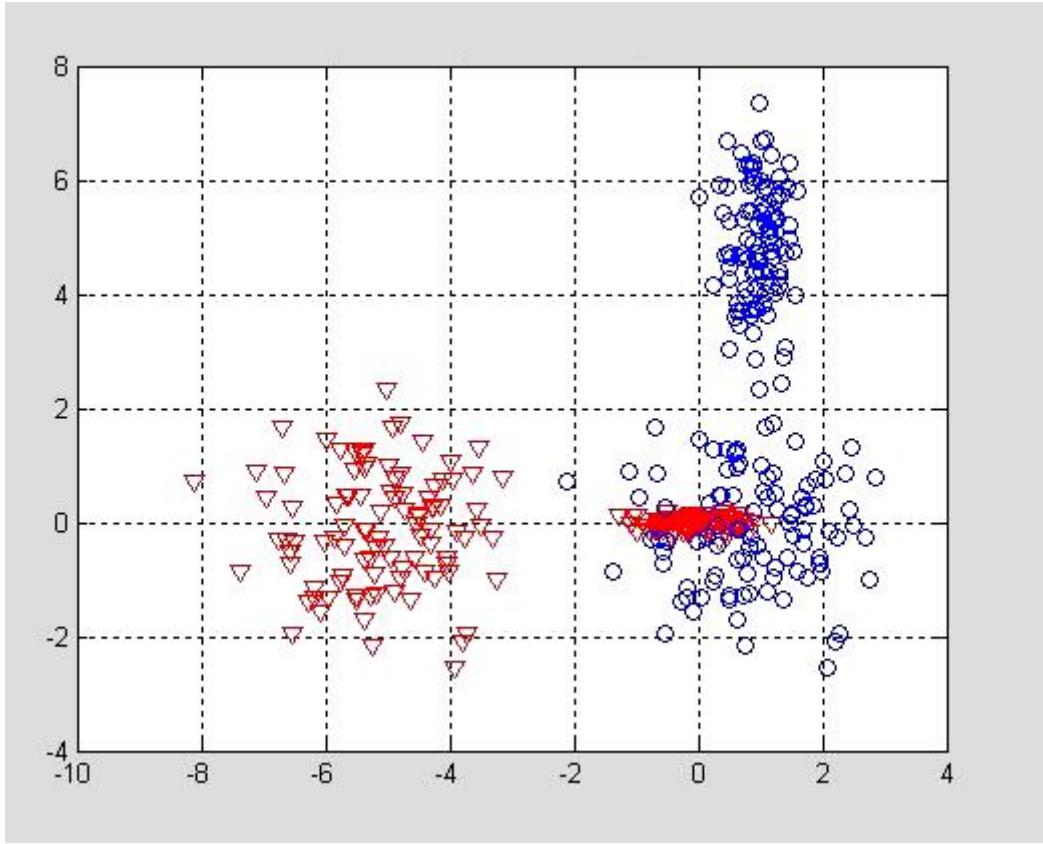
4.3 Yapay Veri-3

Tez çalışmasında geliştirilen yöntem, yapay olarak oluşturulmuş, literatürde elde edilen sınıflandırma sonuçlarının yayınlandığı üçüncü bir veri setinde de denenmiştir (Juang B.,et al,1992). Makale incelendiğinde, bu veri setinin sınıflandırıcı tasarımında geliştirilen yeni bir yöntemi test etmek üzere hazırlanmış olduğu ve yine iki boyutlu vektörlerden oluşan iki sınıf tanıma problemi üzerinde çalışıldığı gözlenmiştir

Sınıflar oluşturulurken şu kriterler gözetilmiştir:

- 1.sınıf, ortalaması $[-5 \ 0]$ ve değişinti matrisi $[1 \ 0; 0 \ 1]$ olan gauss dağılımlı 700 örnek ile ortalaması $[0 \ 0]$ ve değişinti matrisi $[0.25 \ 0; 0 \ 0.01]$ olan gauss dağılımlı 300 örneğin oluşturduğu sınıf ve
- 2.sınıf, ortalaması $[1 \ 5]$ ve değişinti matrisi $[0.1 \ 0; 0 \ 1]$ olan gauss dağılımlı 500 örnek ile ortalaması $[1 \ 0]$ ve değişinti matrisi $[1 \ 0; 0 \ 0.1]$ olan gauss dağılımlı 500 örneğin oluşturduğu sınıftan oluşmaktadır.

Makalede her iki sınıftan 1000 vektör alınmış olmakla beraber tez çalışması için grafiklerde daha görünür olması açısından her sınıf için 200 vektör kullanılmıştır. C_1 ve C_2 sınıflarının dağılımı aşağıdaki şekilde verilmiştir.



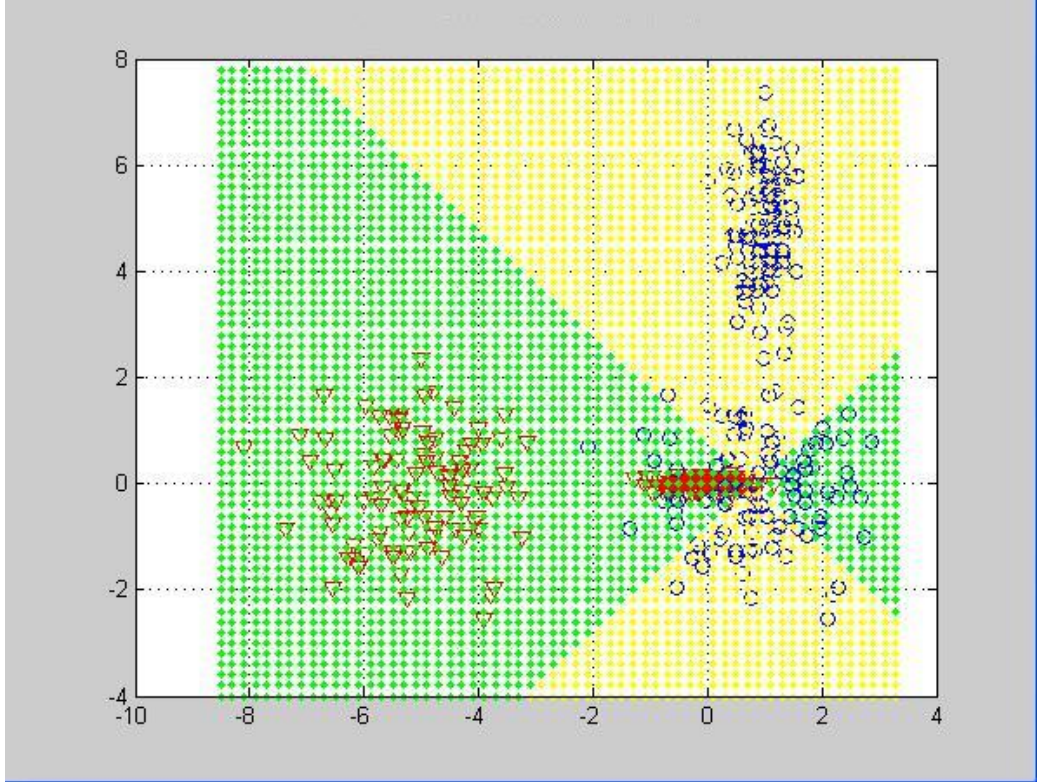
Şekil 4.17 Yapay veri-3 için C_1 ve C_2 sınıflarının x-y düzleminde dağılımı

4.3.1 OVY ve SAM için yapay veri-3 Sonuçları

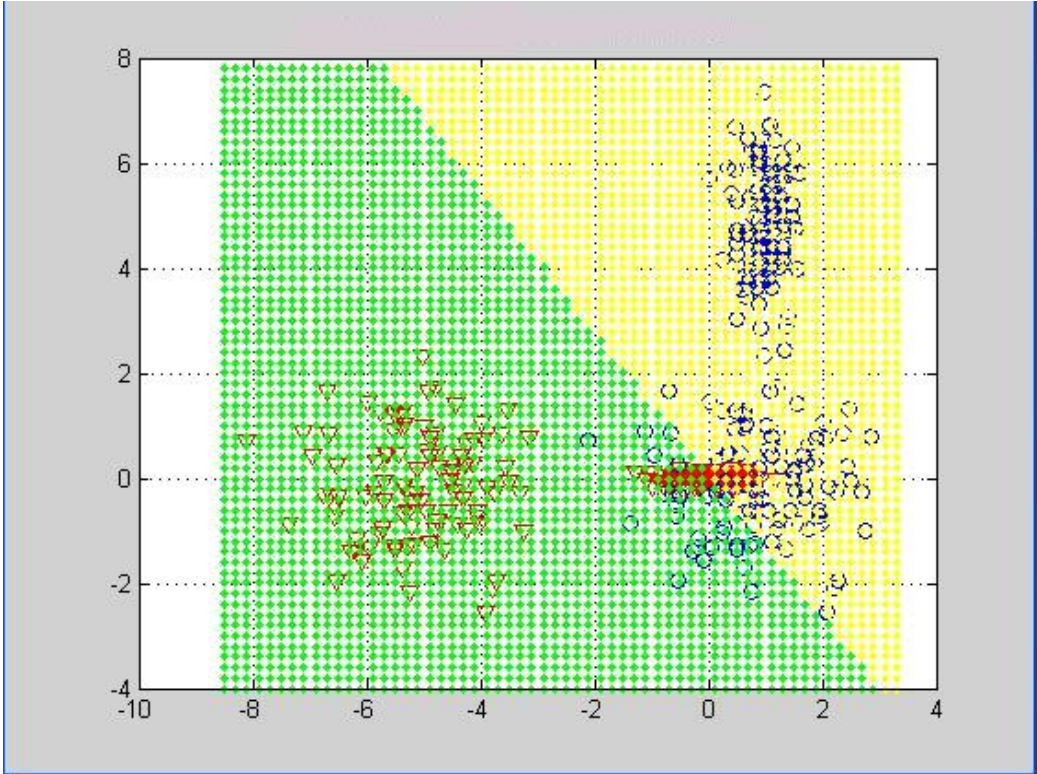
OVY ve SAM'da iki boyutlu uzayda tek özdeğer kullanarak aşağıdaki ayırma bölgeleri elde edilmiştir.

OVY yöntemi için iki sınıf için hatalı tanıma oranının % 12, başarılı tanıma oranının ise % 88 olduğu görülmektedir.

Aynı şekilde SAM algoritması yapay veriye uygulandığında elde edilen ayırıcı bölgelerde, yanlış tanıma % 18.25, başarılı tanıma ise % 81.75 olmuştur.



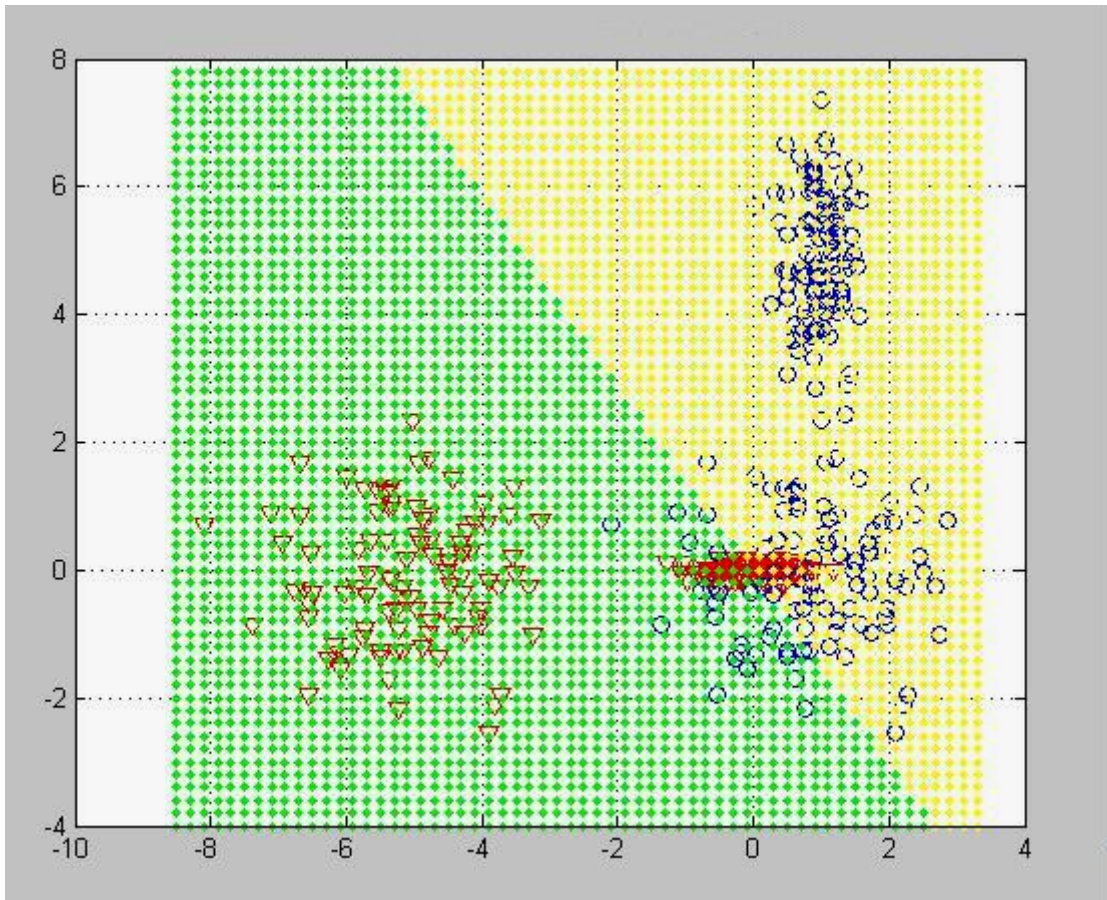
Şekil 4.18 Yapay veri-3 için OVY ile sınıflandırma



Şekil 4.19 Yapay veri-3 için SAM ile sınıflandırma

4.3.2 Yapay veri-3 için Fisher_LDA uygulanması

Bu uygulama ile grafiksel olarak kriterin ayırıcılığı aşağıda verilmiştir. Algoritma, % 81 başarılı tanıma sonucunu vermiştir. Fisher'in kullandığı Doğrusal Ayırma Analizi(LDA) iki boyutlu uzayda tek boyutlu bir ayırıcı elde edebilir ki bu da elde edilen doğrudur.

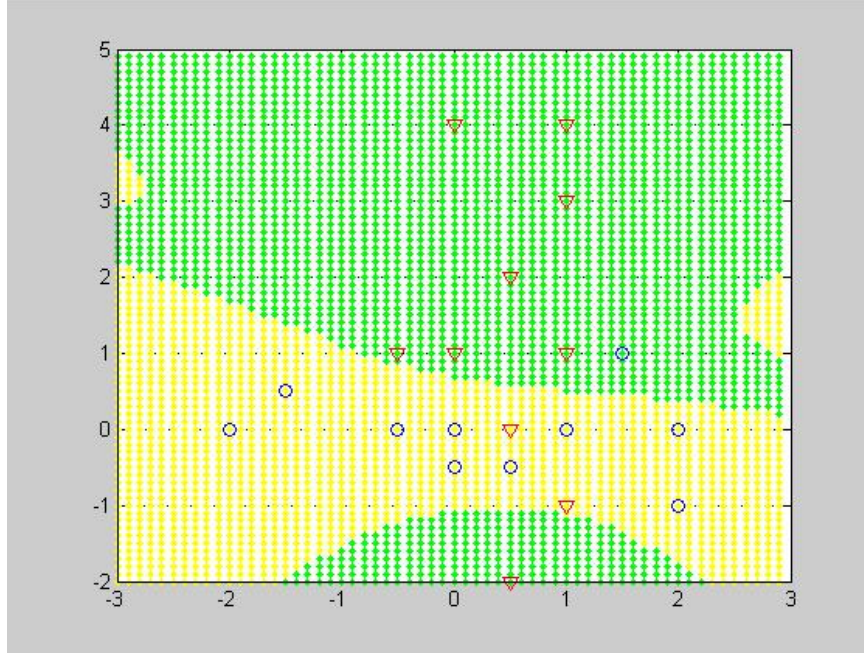


Şekil 4.20 Yapay veri-3 için LDA ile sınıflandırma

4.3.3 OVY ve SAM ölçütlerini oranlayarak yapay veri-3 için sınıflandırma

Bu çalışmada Fisher LDA gibi sınıf içi ve sınıflar arası metrikler doğrudan oranlanarak kullanılmıştır. Fisher LDA den farkı, her iki metrik için tek bir altuzay kullanmak yerine, sınıf içi ve sınıflar arası metrikler için ayrı ayrı altuzaylar bulup elde

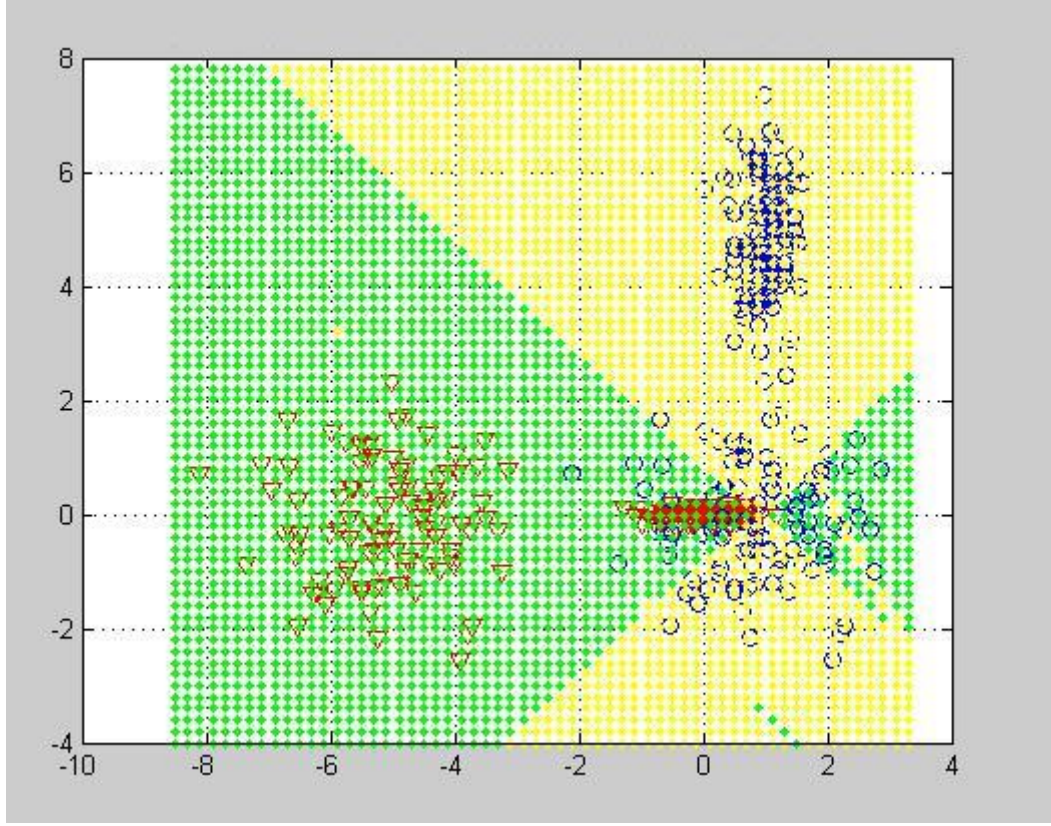
edilen metrikleri oranlamaktır. Diğer çalışmalara ait sonuçlarda da bu çalışma Fisher-Oran olarak anılmaktadır. Burada elde edilen tanıma başarısı da % 92 olmuştur.



Şekil 4.21 Yapay veri-3 için Fisher-Oran ile sınıflandırma

4.3.4 Yapay veri-3 için birleştirilmiş metrik ile sınıflandırma

Bu çalışmada, sınıflara ait veriler kullanılarak Şekil 3.3'te verilen yapay sinir ağı yapısı için ağırlıklar, analitik yöntemle hesaplanmıştır. Bu ağırlıklar kullanılarak elde edilen tanıma başarısı % 88.25'dir. Sınıflandırma bölgeleri de aşağıdaki şekilde gösterilmiştir.



Şekil 4.22 Yapay veri-3 için OVY ve SAM'den elde edilen ölçütlerin birleştirilmesi

Aşağıdaki çizelge de tez çalışmasında elde edilen sonuçlarla beraber makaledeki sonuçlarda aynı çizelge de verilmiştir. Birleştirilmiş metrik makaledeki iki yöntemden daha iyi sonuçlar vermiştir.

Makalede kullanılan kısaltmaların anlamları ise şöyledir

- LC : Lineer Classification
- PE : Perceptron Error
- MCE: Minimum Classification Error
- MSE: Minimum Squared Error

Çizelge 4.3 Yapay veri-3 için tanıma başarımları

		% Başarılı Tanıma
Tez Çalışmasındaki Sonuçlar	OVY	88
	SAM	81.75
	Fisher LDA	81
	Fisher-Oran	92
	Birleştirilmiş Metrik	88.25
Makaledeki Sonuçlar	LC+PE	87.75
	LC+MSE	84.75
	LC+MCE	90

4.4 MNIST Veri Tabanı

Gerçek veri tabanındaki bu ilk uygulamada toplam 10 sınıf bulunan MNIST veri tabanı kullanılmıştır. Deney seti 0 ile 9 arasındaki el yazısı karakterlere ait görüntü dosyalarıdır. Her bir sınıftaki eleman sayıları şöyledir:

Çizelge 4.4 El yazısı rakamlardan oluşan veri kümesinin rakamlara göre dağılımı

	Test Kümesi	Eğitim Kümesi
“ 0 ”	1135	6742
“ 1 ”	1032	5958
“ 2 ”	1010	6131
“ 3 ”	982	5842
“ 4 ”	892	5421
“ 5 ”	958	5918
“ 6 “	1028	6265
“ 7 “	974	5851
“ 8 “	1009	5949
“ 9 “	980	5923

Uygulamada 10 sınıflık veri tabanı sanki iki sınıfmış gibi ele alınmıştır. İki sınıf gibi uygulandığında, rakamlardan birisi ilk sınıf, geri kalan diğer dokuz sınıfın vektörleri de ikinci sınıf olarak olarak kullanılmıştır.

El yazısı rakamlara ait veriler doğrudan ham veri olarak kullanılmamıştır. Özellik vektörleri olarak, daha önce yapılan çalışmalarda tanıma başarımı olarak iyi sonuçlar elde edilen 47 boyutlu Zernik_Moment vektörler kullanılmıştır (Çetişli B.,2003).

Eğitim seti olarak herbir rakama ait 5000, test seti olarak herbir rakama ait 850 vektör kullanılmıştır. Eğitim ve test setleri birbirinden farklıdır.

Çizelge 4.5 El yazısı karakter tanıma sonuçları

Biri ve diğerleri N(Eğitim seti) = 5000 M(test seti) = 850						
EĞİTİM (5000 örnek)						
	OVY	SAM	Fisher_Oran	FISHER	Borda_Count	Birleştirilmiş M
0	96,932	95,429	97,121	96,711	96,520	96,932
1	87,116	77,104	87,299	72,359	81,360	87,116
2	89,192	85,162	89,793	75,454	86,181	89,192
3	95,377	84,607	94,904	88,579	87,829	95,377
4	85,820	81,142	87,576	84,233	83,993	85,820
5	87,306	84,177	88,202	86,220	84,962	87,306
6	92,269	85,436	91,707	87,299	88,771	92,269
7	92,804	85,301	93,194	87,778	86,262	92,804
8	88,163	82,724	88,296	87,450	82,598	88,163
9	96,633	95,326	97,411	95,754	95,846	96,633
ORT	91,161	85,641	91,550	86,184	87,432	91,161
TEST (850 örnek)						
	OVY	SAM	Fisher_Oran	FISHER	Borda_Count	Birleştirilmiş M
0	97,536	97,098	98,059	97,500	97,157	97,536
1	86,837	76,065	86,654	70,320	80,111	86,837
2	88,987	84,204	89,575	75,791	85,935	88,987
3	95,856	85,261	95,405	87,850	88,725	95,856
4	84,758	82,417	87,105	84,673	83,817	84,758
5	88,333	85,072	89,078	87,595	85,797	88,333
6	92,275	85,222	91,392	86,059	88,850	92,275
7	93,177	85,431	93,549	87,601	86,850	93,177
8	87,549	82,850	88,405	86,994	82,425	87,549
9	96,242	96,503	97,909	97,340	95,608	96,242

El yazısı rakamlardan oluşan 10 sınıflı veri tabanında sadece “sıfır” rakamı ve diğerleri olacak şekilde şöyle tamamlayıcı bir çalışma da yapılmıştır. Bu durumda ilk sınıf 500 elemanlı “sıfır” rakamı ve ikinci grup 4500 elemanlı “bir-dokuz” rakamları görüntülerinden oluşmaktadır. Oluşturulan sistemle veri tabanı önce OVY sınıflandırıcısına girilerek sonuçlar elde edilmiş sonra burada tanınmayan örnekler

SAM sınıflandırıcısından geçirilmiştir. İşlem bir kerede önce SAM, sonra OVY olacak şekilde tekrarlanmış, sistem performanslarına bakılmış, sonuçlar çizelge 4.5 olarak verilmiştir. Bu çalışmanın yapılmasının amacı OVY ve SAM sonuçlarının birbirinin tamamlayıcı olabileceğini görmektir. Gerçektende oluşturulan sistemlerde bir sınıflayıcının tanıyamadığı elemanın diğer sınıflayıcı tarafından doğru olarak tanındığı örnekler elde edilebilmiştir.

Birleştirilecek sınıfların seçilmesinde dikkat edilmesi gereken önemli hususlardan biri, birleştirilen sınıflandırıcıların birbirlerini tamamlamasıdır. Sınıflandırıcılardan birisi kusursuz sınıflandırma yapıyor ise sınıflandırıcı birleştirmeye gerek yoktur. Ama sınıflandırıcılar hata yapıyorsa bu hataların farklı kümelerde olmasına dikkat edilmelidir (Kuncheva, 2004).

Tabiki çalışma çok detaylandırılmamıştır. Genellemeler yapmak doğru olmayabilir ancak örnekte görüldüğü üzere birinin başarısız olduğu durumlarda diğeri onu destekleyebilmektedir.

Çizelge 4.6 Sınıflayıcıların birbirlerini tamamlaması ile ilgili sonuçlar

	OVY	SAM	OVY → SAM	SAM → OVY
1.sınıf için doğru tanınan eleman sayısı (500 içinde)	490	461	490	490
2.sınıf için doğru tanınan eleman sayısı(4500 içinde)	4365	4431	4474	4481
1.sınıf için tanıma yüzdeleri	% 98	% 92.2	% 98	% 98
2.sınıf için tanıma yüzdeleri	% 97	% 98.46	% 99.42	% 99.5

4.5 TIMIT Veri Tabanı

TIMIT veri tabanı otomatik ses tanıma sistemlerinin geliştirilmesinde kullanılmak üzere ses verisi sağlamak amacıyla tasarlanmış literatürde de kullanılan standart bir veri tabanıdır.

TIMIT toplamda 6300 cümle içerir. ABD'nin 8 ana lehçe bölgesinden, 630 konuşmacı tarafından 10'ar cümle söylenmiştir.

TIMIT deki yazı materyali Stanford Araştırma Enstitüsü (SRI) tarafından hazırlanmış 2 “sibboleth” cümle, Massachusetts Teknoloji Enstitüsü (MIT) tarafından hazırlanmış 450 tane fonetik olarak kompakt cümle ve Texas Instruments (TI) tarafından hazırlanmış fonetik olarak çeşitli 1890 cümle içermektedir. Konuşmacıların lehçesel farklılıkları da gözönüne alınmıştır. Bölgeler ve cinsiyete göre dağılım çizelge 4.6'da verilmiştir. Yapılan deneylerde TIMIT veri tabanındaki 12 ünlü kullanılmıştır.

Çizelge 4.7 Konuşmacıların lehçe dağılımları

Bölge (dr)	Lehçe Bölgeleri	#Erkek	#Bayan	Toplam
1	New England	31 (63%)	18 (27%)	49 (8%)
2	Northern	71 (70%)	31 (30%)	102 (16%)
3	North Midland	79 (67%)	23 (23%)	102 (16%)
4	South Midland	69 (69%)	31 (31%)	100 (16%)
5	Southern	62 (63%)	36 (37%)	98 (16%)
6	New York City	30 (65%)	16 (35%)	46 (7%)
7	Western	74 (74%)	26 (26%)	100 (16%)
8	Army Brat(taşınanlar)	22 (67%)	11 (33%)	33 (5%)
	Toplam	438 (70%)	192 (30%)	630 (100%)

Çizelge 4.8 Timit veri tabanı ünlülere göre dağılımı

ÜNLÜ	Eğitim setindeki veri sayısı	Test setindeki veri sayısı
'iy'	4626	1810
'ih'	4245	1438
'eh'	3277	1247
'ey'	2271	802
'ae'	2292	772
'aa'	2256	846
'aw'	728	216
'ay'	1934	686
'ah'	2265	860
'ao'	1865	761
'oy'	304	127
'ow'	1653	600

Deneylerde toplam 13 parametreden oluşan öznitelik vektörleri kullanılmıştır. Ses örnekleri üzerinde ön vurgulama yapıldıktan sonra öznitelik vektörleri uzunluğu 10 ms ve ardışık iki pencerenin örtüşmesi 2.5 ms olacak şekilde 19 adet pencereye ayrılmıştır. Türetim sırasında 19 çerçeveden uzun olanlar ağırlık merkezine göre sağından ve solundan 9'ar çerçeve alınmıştır. 19 Çerçeveden kısa olanlara ise ağırlık merkezine göre sağından ve solundan sıfır eklenmiştir (Koç M.,2006).

Her bir pencereden 12 MFCC ve çerçeve enerjisinin logaritması olmak üzere toplam 13 parametre elde edilmiştir. Her hecenin herbir söyleşi için toplam $19 \times 13 = 247$ boyutlu özvektörler kullanılmıştır.

Deneylerde sınıf içi ölçütü kullanıldığında, farklılık altuzayına izdüşüm matrisi elde edilirken veri sayısı boyut sayısından fazla olduğundan yeterli veri durumu ortaya

çıkılmaktadır (Gülmezoğlu, 2001). Bu durumda sınıf içi ölçütünü enküçük yapan özdeğerler sayısı, seçilen bir yüzde enerji değerine göre seçilir (Koç M., 2006).

Eğitim setinde her hece için 300, test setinde ise 125 söyleyiş kullanılmıştır. Eğitim setinde özdeğer sayısı olarak OVY 'da 3 ve SAM'da 7 özdeğer kullanılmıştır. OVY ve SAM ile veri kümesinin tanıma başarımları elde edilirken bu özdeğer sayıları belirlenmiştir. Tutarlılığı sağlamak için Fisher ve birleştirilmiş yöntemde de buradaki özdeğer sayıları alınmıştır. Test setinde OVY için 125, SAM için 12 özdeğer kullanılmıştır. Yine Fisher ve Combined yöntemde de sınıf içi için 125, sınıflar arası için 12 özdeğer kullanılmıştır.

Sonuçları karşılaştırabilmek için “Borda Count” yöntemi de bu veri kümesine uygulanmıştır. OVY ve SAM sınıflayıcıların verdikleri ham skorları ayrı ayrı bulunup toplanarak yeni skorlar elde edilmiş ve en yüksek skora sahip olan sınıfı son karar olarak seçilmiştir. Literatürde yaygın olarak bilinen birleştirme yöntemlerden biri olmakla birlikte probleme has bir birleştirme algoritması değildir.

Tablolarda GMM ile ilgili sonuçlar da verilmektedir.. Bu çalışma GMM'in kullanılmasındaki amaç çoğu sistemlerde iyi çalıştığı bilinen bu sınıflandırıcı sonuçlarıyla, tezde geliştirilen yöntemin sonuçlarını karşılaştırarak performansı karşılaştırabilmektir. HMM de kullanılabilirdi ancak bu tez çalışmasında ele alınmamıştır.

GMM'de her sınıf 5 Gaussian ile modellenmiş model parametreleri çıkarılmış ve oluşturulan sisteme gelen girdiler bu model parametreleri uyuşmalarına göre o sınıfa ait veya değil diye değerlendirilmiştir. Sonuçlar 5 gaussian karışım kullanılarak elde edilmiştir.

Çizelge 4.9 TIMIT veri tabanının eğitim kümesi sınıflandırma sonuçları

EĞİTİM								
	OVY	SAM	Fisher_Oran	Fisher	Borda_Count	Birleştirilmiş Ölçüt		GMM
						λ_2	λ_1	
iy'	88.256	66.356	86.3	82.667	77.544	88.256	85.911	87.222
ih'	72.5	71.167	74.222	62.222	72.5	50	75.878	75.556
eh'	77.911	65.567	77.667	49.889	74.367	77.911	83.356	65.589
ey'	92.844	59.167	91.8	86.133	73.544	92.844	88.322	83.9
ae'	94.089	60.522	92.633	69.144	74.067	94.089	88.556	78.444
aa'	91.622	57.267	91.756	77.489	69.2	91.622	88.7	74.122
aw'	93.633	69.089	88.011	65.111	75.456	93.633	89.656	85.189
ay'	93.911	68.789	87.967	83.289	75.333	93.911	90.178	80.833
ah'	76.944	65.467	77.711	62.644	76.644	76.889	79.922	74.256
ao'	91.822	56.733	91.678	80.9	68.589	91.822	89.3	76.511
oy'	92.956	76.133	83.322	86.489	80.122	92.956	87.133	89.2
ow'	92.933	58.044	90.867	55.1	72.522	92.933	89.767	74.367
uh'	75.878	71.711	77.811	53.111	75.878	50	82.211	72.5
uw'	92.056	57.533	91.178	78.389	77.956	92.056	80.822	82.589
ux'	90.744	59.6	89.233	60.867	78.633	90.744	81.156	76.633
er'	93.511	56.367	93.756	89.389	68.444	93.511	92.022	83.844
ORT	88.226	63.7195	86.6195	71.4271	74.42494	85.19856	85.8056	78.7972

Çizelge 4.10 TIMIT veri tabanının test kümesi sınıflandırma sonuçları

TEST							
	OVY	SAM	Fisher_Oran	Fisher	Borda_Count	Birleştirilmiş Ölçüt (BM)	
						λ_2	λ_1
iy'	62.88	71.627	72.88	71.04	71.947	62.88	56.96
ih'	45.013	48.027	47.173	50.32	47.92	45.013	47.653
eh'	52.64	64.053	62.453	36.507	63.84	52.64	53.387
ey'	49.36	41.2	44.747	51.307	41.12	49.36	49.173
ae'	49.707	47.68	45.653	51.04	48.053	49.707	46.747
aa'	49.093	34.347	47.387	45.493	34.08	49.093	50.267
aw'	48.053	52.8	51.707	46.507	52.8	48.053	50.72
ay'	49.867	55.893	58.533	47.68	55.893	49.867	55.04
ah'	49.28	45.84	46.693	50.4	46	49.28	48.853
ao'	49.787	54.773	50.24	52.613	54.72	49.787	50.347
oy'	50.56	53.067	53.84	46	53.04	50.56	55.84
ow'	49.787	52.32	51.173	47.147	52.32	49.787	49.44
uh'	47.36	47.173	48.347	53.12	47.147	47.36	48.453
uw'	50.133	47.84	49.787	47.413	47.813	50.133	50.267
ux'	49.04	39.867	42.213	60	39.813	49.04	49.173
er'	52.24	60.907	53.44	47.04	59.68	52.24	51.813
ORT	50.3	51.08838	51.64163	50.2267	51.01163	50.3	50,883

4.6 IRIS Veri Tabanı

IRIS veri tabanı, örüntü sınıflandırıcıların başarımlarını karşılaştırmada kullanılmaktadır (Juang B. et al, 1992). IRIS veri tabanı 3 sınıftan oluşmaktadır. Her sınıf 150 adet örneğe sahiptir ve her örnek 4 boyutludur (Petal length, petal width, sepal length, sepal width).

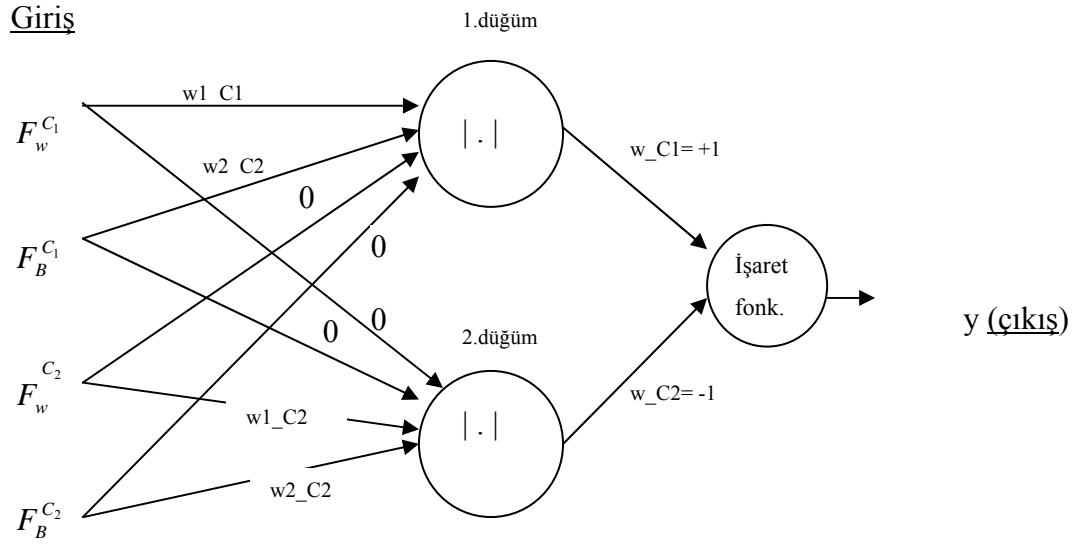
Deneylerde 1, 2, 3 ve 4 özdeğer için model eğitimi ve sınıflandırma testi yapılmıştır.

Deneysel çalışmada, IRIS veri tabanı 3 sınıftan oluşmakla birlikte sanki iki sınıfmış gibi ele alınmıştır. İki sınıf gibi uygulandığında, gruplardan birisi ilk sınıf, geri kalan diğer iki sınıfın vektörleri de ikinci sınıf olarak ele alınmıştır.

Eğitim seti olarak her sınıfa ait 1-30 arasındaki 30 örnek ve test seti olarak da sonraki 20 örnek kullanılmıştır. Böylece eğitim ve test setleri birbirinden farklı iki set olarak oluşturulmuştur.

IRIS veri tabanında yine iki sınıf problemi üzerinde sınıflayıcı olarak NN (Neural Network) algoritması da uygulanmıştır. NN'ün girdileri IRIS veri tabanı datalarıdır. Ağırlıklar ile ilgili hiç bir kısıtlama kullanılmamıştır. Eğitim ve test kümelerinde elde edilen tanıma başarımları Çizelge 4.10 ve 4.11 de verilmiştir (Çizelge de BM olarak kullanılan kısaltma “Birleştirilmiş Metrik” ifadesi için kullanılan bir kısaltmadır). Tez çalışmasında geliştirilen yöntemin (2 ya da daha fazla özdeğer ile altuzaylar bulunarak uygulandığı durumda) NN'den daha iyi sonuçlar verdiği gözlenmiştir.

Geliştirilen yöntem anlatılırken birleştirme işleminin sinir ağı yapısındaki şematik gösterimi verilmişti. Bu yapıyı temel alarak birleştirilmiş yöntem, NN algoritması şeklinde uygulanmıştır. NN ile ağırlıklar belirlenip sınıflama yapıldığındaki başarımlar da tabloda yer almıştır. NN'ün girdileri, iki sınıf için ayrı ayrı elde edilen sınıf içi ve sınıflar arası dağılımdan elde edilen ölçütlerdir. Ağırlıklar ve nodlara gelen girdiler için kısıtlamalar getirilmiştir. Aşağıdaki şekilde (Şekil 4.17) uygulamanın nasıl yapıldığı gösterilmektedir. Bu yapıda NN uygulandığında elde edilen tanıma başarımlarının düşük olduğu gözlemlenmiştir. Bu sonuçlarda ağırlıklara getirilen kısıtlamaların etkisi olduğu düşünülmektedir.



Şekil 4.23 OVY ve SAM'den elde edilen ölçütlerin yapay sinir ağına uygulanması

Çizelge 4.11 IRIS veri tabanı eğitim seti sınıflandırıcı sonuçları

Vektör sayısı = 30 Vektör boyutu = 4											
EĞİTİM											
Özdeğer sayısı = 1									SAM için kendi ortalamasını kullanarak		NN yapısında BM
	CVA	SAM	Fisher Oran	Fisher	Borda Count	NN	BM	GMM	Fisher	BM	
1--> 2.3	90.00	99.17	100.00	100.00	99.17	100.00	100.00	74.16	50.00	88.33	81.00
2--> 1.3	76.67	65.00	76.67	75.00	65.83	78.80	64.17	96.67	50.00	76.67	69.00
3--> 1.2	77.50	82.50	87.50	90.00	71.67	96.60	81.67	97.50	50.00	58.33	67.00
ORT	81.39	82.22	88.06	88.33	78.89	91.80	81.95	89.44	50.00	74.44	72.33
Özdeğer sayısı = 2									SAM için kendi ortalamasını kullanarak		NN yapısında BM
	CVA	SAM	Fisher Oran	Fisher	Borda Count	NN	BM	GMM	Fisher	BM	
1--> 2.3	100.00	99.17	100.00	100.00	99.17	100.00	100.00	--	50.00	100.00	94.00
2--> 1.3	90.00	66.67	86.67	58.33	73.33	78.80	90.00	--	50.00	90.00	62.00
3--> 1.2	88.33	84.16	96.67	90.00	84.17	96.60	88.33	--	50.00	88.33	73.00
ORT	92.78	83.33	94.45	82.78	85.56	91.80	92.78	--	50.00	92.78	76.33
Özdeğer sayısı = 3									SAM için kendi ortalamasını kullanarak		NN yapısında BM
	CVA	SAM	Fisher Oran	Fisher	Borda Count	NN	BM	GMM	Fisher	BM	
1--> 2.3	100.00	99.17	100.00	100.00	99.17	100.00	100.00	--	50.00	100.00	89.00
2--> 1.3	88.33	66.67	85.83	50.83	73.34	78.80	90.83	--	50.00	88.33	68.00
3--> 1.2	86.67	82.50	94.17	82.50	80.33	96.60	86.67	--	50.00	86.67	75.00
ORT	91.67	82.78	93.33	77.78	84.28	91.80	92.50	--	50.00	91.67	77.33
Özdeğer sayısı = 4									SAM için kendi ortalamasını kullanarak		NN yapısında BM
	CVA	SAM	Fisher Oran	Fisher	Borda Count	NN	BM	GMM	Fisher	BM	
1--> 2.3	99.17	99.17	99.17	100.00	99.17	100.00	99.17	--	50.00	99.17	100.00
2--> 1.3	66.67	66.67	66.67	51.67	66.67	78.80	87.50	--	50.00	66.67	66.00
3--> 1.2	82.50	82.50	82.50	83.33	82.50	96.60	85.00	--	50.00	82.50	65.00
ORT	82.78	82.78	82.78	78.33	82.78	91.80	90.56	--	50.00	82.78	77.00

Çizelge 4.12 IRIS veri tabanı test seti sınıflandırıcı sonuçları

Vektör sayısı = 30 Vektör boyutu = 4										
TEST										
Özdeğer sayısı = 1								SAM için kendi ortalamasını kullanarak		NN yapısında BM
	CVA	SAM	Fisher Oran	Fisher	Borda_Count	NN	BM	Fisher	BM	
1--> 2,3	86.25	97.5	100	100	96.25	98.30	97.5	50	86.25	83
2--> 1.3	76.25	60	68.75	77.5	62.5	75.00	65	50	73.75	63
3--> 1.2	72.5	91.25	92.5	93.75	76.25	91.60	81.25	50	56.25	68
ORT	78.33	82.92	87.08	90.42	78.33	88.30	81.25	50.00	72.08	71.33
Özdeğer sayısı = 2								SAM için kendi ortalamasını kullanarak		NN yapısında BM
	CVA	SAM	Fisher Oran	Fisher	Borda_Count	NN	BM	Fisher	BM	
1--> 2.3	97.5	97.5	100	100	97.5	--	100	50	97.5	91
2--> 1.3	85	65	83.75	65	72.5	--	86.25	50	85	59
3--> 1.2	85	91.25	96.25	87.5	91.25	--	85	50	85	72
ORT	89.17	84.58	93.33	84.17	87.08	86.11	90.42	50.00	89.17	74
Özdeğer sayısı = 3								SAM için kendi ortalamasını kullanarak		NN yapısında BM
	CVA	SAM	Fisher Oran	Fisher	Borda_Count	NN	BM	Fisher	BM	
1--> 2.3	100	97.5	100	97.5	97.5	--	100	50	100	85
2--> 1.3	91.25	65	58.33	58.75	75	--	91.25	50	91.25	61
3--> 1.2	87.5	91.25	90	83.75	88.75	--	92.5	50	87.5	70
ORT	92.92	84.58	82.78	80.00	87.08	86.11	94.58	50.00	92.92	72
Özdeğer sayısı = 4								SAM için kendi ortalamasını kullanarak		NN yapısında BM
	CVA	SAM	Fisher Oran	Fisher	Borda_Count	NN	BM	Fisher	BM	
1--> 2.3	97.5	97.5	97.5	97.5	97.5	--	97.5	50	97.5	98
2--> 1.3	65	65	65	60	65	--	88.75	50	65	62
3--> 1.2	91.25	91.25	91.25	83.75	91.25	--	96.25	50	91.25	64
ORT	84.58	84.58	84.58	80.42	84.58	86.11	94.17	50.00	84.58	74.67

5. SONUÇLAR

Bu çalışmada sınıf içi ve sınıflar arası dağılımlardan elde edilen ölçütleri birleştiren yeni bir ölçüt kullanılmıştır. Bu ölçütü kullanan sınıflandırıcının iki-sınıf problemine uygulanması yapay sinir ağı ile gösterilmektedir. Bölüm 4.1 ve 4.2 'de anlatıldığı gibi, geliştirilen sınıflandırıcı önce iki boyutlu yapay veri üzerinde test edilmiş ve OVY, SAM ve Fisher LDA 'den daha iyi sonuçlar vermiştir. Bu aşamadan sonra yöntem, gerçek veri tabanlarında test edilmiştir.

Test edilen ilk veri tabanı olan MNIST veri tabanında başarılı sonuçlara ulaşılamamıştır. Geliştirilen birleştirilmiş sınıflandırıcı, OVY gibi davranmakta sınıflar arası dağılımlardan gelecek katkıyı gözardı etmektedir. Daha sonra kullanılan TIMIT veri tabanında, birleştirilmiş sınıflayıcının MNIST veri tabanından daha iyi çalıştığı gözlemlenmiştir. OVY ve SAM'dan daha iyi sonuçlar verdiği gibi Fisher LDA ve Borda Count yönteminden de daha iyi sonuçlar elde edilmiştir. Ancak yöntem bu veri tabanının tüm sınıflarında iyi sonuçlar vermeyip, bazı sınıf gruplarında iyi çalışmaktadır. Bu veri tabanında tamamlayıcılığa bakmak yerinde olabilir. Daha küçük boyutlu olması açısından yapılanların sağlamlasını da elde etmek adına IRIS veri tabanında da çalışılan yöntem test edilmiştir. Orada da başarılı sayılabilecek sonuçlar alınmıştır.

Yapay veri üzerinde edilen sonuçlar, önerilen sınıflandırıcı birleştirme yönteminin gerçek örüntü sınıflandırmada kullanılabileceğini göstermektedir. Geliştirilen birleştirilmiş sınıflayıcı, sınıflar arası dağılımında katkısını artıracak yeni ölçütler bulunup, yüksek performans elde edilecek şekilde iyileştirilmelidir.

6. KAYNAKLAR DİZİNİ

- Bennani Y., Gallinari P., Neural Networks for Discrimination and Modellization of Speakers. *Speecch Communication.Cilt 17.s.159-175.1995*
- Bishop C.M., Neural networks for Pattern Recognition. Aston University. Birmingham. UK. 1996. sayfa 310-313
- Cao J., Ahmadi. M., Shridhar. M., Recognition of Handwritten Numerals with Multiple Feature and Multi-stage Classifier. *Pattern Recognition. Vol. 28. No. 2. pp. 153-160. 1995*
- Dietterich T. G., Ensemble Methods in Machine Learning. *Lecture Notes in Computer Science. Cagliari.Italy. 2000. Springer. pp. 1-15*
- Duin. R.P.W., Loog. M., and Umbach. R.H., Multi-class Linear Feature Extraction BM Nonlinear SAM. *ICPR15. Proc. 15th Int. Conference on Pattern Recognition. 2000*
- Felföldi. L., Kocsor. A., Tóth. L.: Classifier Combination in Speech Recognition. *Periodica Polytechnica. Electrical Engineering. Vol. 47. No. 1-2. pp. 125-140. 2003*
- Gader. P., Mohamed. M., and Keller. J., Fusion of handwritten word classifiers. *Pattern Recognition Letters. vol. 17. no. 6. pp. 577–584. 1996.*
- Gülmezoğlu. M. B., Dzhafarov. V., Keskin. M., and Barkana. A., A Novel Approach To Isolated Word Recognition. *IEEE Trans. Speech and Audio Processing. vol. 7. pp. 620-628. 1999 3. 6*
- Gülmezoğlu. M.B., Dzhafarov. V., and Barkana. A., The Common Vector Approach and its Relation to Principal Component Analysis. *IEEE Trans. Speech and Audio Processing. vol. 9. no. 6. 2001*
- Günel. S., Ergin S., Gerk Ö.N., Spam e_mail recognition BM subspace analysis.*Proc.INISTA 2005. Istanbul. 2005.*
- Huang Y.S., Suen C.Y., A Method of Combining Multiple Experts for the Recognition of Unconstrained Handwritten Numerals. *IEEE Transactions on Pattern Analysis and Machine Intelligence.cilt 17. No. 1. s. 90-94. 1995.*
- Ji. C. and Ma. S. Combinations of weak classifiers. *IEEE Transactions on Neural Networks. vol. 8. no. 1. pp. 32–42. 1997.*

- Juang B. and Katagiri S., Discriminative learning for Minimum Error Classification, IEEE Transaction On Signal Processing, Vol. 40, No.12, December 1992.
- KirBM. M., Sirovich. L., Application of the Karhunen_Loeve Procedure and the Characrerization of Human Faces.IEEE Transactions on Pattern Analysis and Machine Intelligence.pp. 103-108. 1990
- Kittler. J., Hatef. M., Ruin. R.P.W., Matas. J., On Combining Classifiers. IEEE Trans. on Pattern Analysis and Machine Intelligence. vol. 20. no. 3. 1998
- Koç M., Sınıf İçi ve Sınıflar Arası Dağılımlardan Elde Edilen Ölçütlerin Birleştirilerek Fonem Tanımda Kullanılması. Eskişehir Osmangazi Üniversitesi. Fen Bilimleri Enstitüsü. Yüksek Lisans Tezi. 2006
- Kramer. M. A. . Nonlinear Principal Component Analysis Using Autoassociative Neural Networks. Neural Computation 9 (7). 1493—1516
- Kuncheva. L.I., Combining Pattern Classifiers: Methods and Algorithms. John Wiley and Sons. Inc. 2004. pp 103.295
- Lu. X., Wang. Y., and Jain. A.K., Combining Classifiers for Face Recognition. Proceedings of International Con-ference on Multimedia and Expo. Vol. 3. pp. 13-16. 2003
- Oja. E., Subspace methods of pattern recognition. John Wiley and Sons Inc., New York. 1983
- Radova. V., Psutka. J., An Approach to Speaker Identification Using Multiple Classifiers. Proc. of the IEEE Int. Conf. on Acoustics. Speech and Signal Processing. 1997. pp. 1135–1138
- Reynolds D.A., Rose R.C., Robust Text-Independent Speaker Identification Using Gaussian Mixture Speaker Models. IEEE Transactions on Speech and Audio Processing.cilt3. No. 1. s. 72-83. 1995.
- Rogova G., Combining The results of Several Neural Network Classifiers. Neural Networks. pp 7:777-781.1994.
- Saranlı A., A Unifying Theory for Rank Based Multiple Classifier Systems. Doktora Tezi. ODTÜ. Ocak 2000.
- Taniguchi. M. and Tresp. V., Averaging regularized estimators. Neural Computation. vol. 9. pp. 1163–1178. 1997.
- Tax. D., Breukelen. M. V., Duin. R., and Kittler. J., Combining multiple classifiers BM averaging or BM multiplying?. Pattern Recognition. vol. 33. pp. 1475–1485. 2000.

- Wang. X., Paliwal. K.K., A modified minimum classification error (MCE) training algorithm for dimensionality reduction. *Journal of VLSI Signal Processing Systems*. Vol. 32. pp. 19-28. 2002.
- Xu. L., Krzyzak. A., and Suen. C., Methods of combining multiple classifiers and their applications to handwriting recognition. *IEEE Transactions on Systems, Man, and Cybernetics*. vol. 22. no. 3. pp. 418–435. 1992.
- Zhang. B., Fu. M., Yan. H., A Nonlinear Neural Network Model of Mixture of Local Principal Component Analysis: Application to Handwritten Digits Recognition. *Pattern Recognition*.

ÖZGEÇMİŞ

Şükran Künkü, 1970 yılında Kastamonu’da doğdu. İlk, orta ve lise öğrenimini Çankırı’da tamamladı. 1988 yılında Anadolu Üniversitesi Mühendislik-Mimarlık Fakültesi Elektrik-Elektronik Mühendisliği bölümüne girdi. 1992 yılında bölüm birincisi olarak mezun oldu. Aynı yıl Elek.-Elektro. Müh. Bölümünde araştırma görevlisi olarak çalışmaya başladı. 1992-1995 yılları arasında aynı üniversitede Fen Bilimleri Enstitüsü Elektronik Anabilim Dalında yüksek lisans yaptı. Yüksek lisans eğitimi boyunca sinyal işleme üzerinde çalıştı. Yüksek Lisans tezi olarak “0-9 Arası Rakamlar Ve Bazı Basit Emir Kelimelerinin Dinamik Programlama İle Tanınması” isimli çalışmayı gerçekleştirdi. 1995 ‘te Osmangazi Üniversitesi Fen Bilimleri Enstitüsünde doktora eğitimine başladı. Sinyal işlemleri ve Uygulamaları üzerine dersler aldı. Bölümdeki görevinde, dijital sistemler, devre analizi, yüksek frekans tekniği, haberleşme, bilgisayar programlama derslerine yardımcı oldu. 1998’de doktora yeterlilik sınavını verdi. 1999 yılında tez aşamasındaki doktora çalışmalarını, işi gereği yurt dışına çıkışı nedeniyle dondurdu. 1999 yılında üniversitedeki görevinden ayrılarak Havelsan A.Ş.’ne yazılım mühendisi olarak girdi. 1.5 sene yurt dışında bulundu. Yazılım ve sistem mühendisliği eğitimleri aldı. Aviyonik sistemler ve modernizasyon projelerinde çalışmaya başladı. 2005 yılında tekrar doktora çalışmalarına başladı. 2008 yılı mayıs ayında tezini tamamladı.

Halen Havelsan A.Ş. Eskişehir ekibinde uzman yazılım mühendisi olarak çalışmaktadır. İngilizce bilmektedir. Evli ve iki çocuk annesidir.