T.C.
BAHÇEŞEHİR ÜNİVERSİTESİ

# ÇOK KERNELLİ EN BÜYÜK ARALIK ÖBEKLEMESİ PROBLEMİNİN YARI – SONSUZ OPTİMİZASYON İLE MODELLENMESİ

**Yüksek Lisans Tezi**

GÜLNUR SEICHAN OGLOU

İSTANBUL, 2012

T.C.
BAHÇEŞEHİR ÜNİVERSİTESİ

FEN BİLİMLERİ ENSTİTÜSÜ
MATEMATİK ANA BİLİM DALI
UYGULAMALI MATEMATİK

# ÇOK KERNELLİ EN BÜYÜK ARALIK ÖBEKLEMESİ PROBLEMİNİN YARI – SONSUZ OPTİMİZASYON İLE MODELLENMESİ

**Yüksek Lisans Tezi**

**GÜLNUR SEICHAN OGLOU**

**Tez Danışmanı: Yrd. Doç. Dr. Süreyya ÖZÖĞÜR AKYÜZ**

**İSTANBUL, 2012**

**T.C.**
**BAHÇEŞEHİR ÜNİVERSİTESİ**

**FEN BİLİMLERİ ENSTİTÜSÜ**
**MATEMATİK ANA BİLİM DALI**
**UYGULAMALI MATEMATİK**

| | |
|---|---|
| Tezin Adı | : Çok Kernelli En Büyük Aralık Öbeklemesi Probleminin Yarı – Sonsuz Optimizasyon ile Modellenmesi |
| Öğrencinin Adı Soyadı | : Gülnur SEICHAN OGLOU |
| Tez Savunma Tarihi | : 24 / 04 / 2012 |

Bu tezin Yüksek Lisans tezi olarak gerekli şartları yerine getirmiş olduğu Fen Bilimleri Enstitüsü tarafından onaylanmıştır.

Doç. Dr. F. Tunç BOZBURA
Enstitü Müdürü
İmza

Bu tezin Yüksek Lisans tezi olarak gerekli şartları yerine getirmiş olduğunu onaylarım.

Prof. Dr. Nuri KURUOĞLU
Program Koordinatörü
İmza

Bu Tez tarafımızca okunmuş, nitelik ve içerik açısından bir Yüksek Lisans tezi olarak yeterli görülmüş ve kabul edilmiştir.

| Jüri Üyeleri | İmzalar |
|---|---|
| Tez Danışmanı<br>Yrd. Doç. Dr. Süreyya ÖZÖĞÜR AKYÜZ | ----------------------------------- |
| Üye<br>Doç. Dr. Atabey KAYGUN | ----------------------------------- |
| Üye<br>Yrd. Doç. Dr. Semra AĞRALI | ----------------------------------- |

# ÖNSÖZ

Bana her zaman destek olan ve üzerimde büyük emekleri bulunan canım annem Nermin ŞAHİN'e ve canım babam Barbaros ŞAHİN'e; çok uzaklardan beni her zaman çalışmaya teşvik eden, sevgisini ve desteğini üzerimden hiç eksiltmeyen sevgili kardeşim Halil İbrahim ŞAHİN'e ve aileme bana verdikleri sevgi ve değer için teşekkürler ederim. Kısa zaman önce aralarına katıldığım kocaman bir aile olmamızı sağlayan, yardım ve desteklerini esirgemeyen eşimin çok değerli ailesine, beni kızları olarak bağırlarına basan, manevi katkılarını her zaman hissettiğim kayınbabam Seyhan MUSTAFA ve kayınvalidem Sevgi MUSTAFA'ya da sevgi ve şükranlarımı sunarım. Ve benim kocaman yürekli, sabırlı, anlayışlı hayat arkadaşım, biricik eşim Akay SEICHAN OGLOU' na çalışmalarım süresince bana verdiği moral ve destek, gösterdiği sabır ve anlayıştan dolayı çok teşekkür ederim.

Bu tez çalışmam boyunca bana yardımcı olan, bu alanda çalışmam için beni teşvik eden ve desteğini esirgemeden önerileri ile beni yönlendiren tez danışmanım değerli hocam Yrd. Doç. Dr. Süreyya AKYÜZ'e ve beraber çalışmalar yürüttüğümüz İrem KARADUMAN'a çok teşekkür ederim. Ayrıca Yrd. Doç. Dr Atabey KAYGUN, Yrd. Doç. Dr. Semra AĞRALI ve Öğr. Gör. Burcu GÜNGÖR'e de önerileri ve katkılarından dolayı teşekkür ederim.

Son olarak, yüksek lisans öğrenimim ve tez hazırlığı aşamasında beni anlayışla karşılayan ve destekleyen can dostlarıma, yardımlarını esirgemeyen Matematik bölüm hocalarım ve dönem arkadaşlarıma, İSOV – Dinçkök Teknik ve Endüstri Meslek Lisesi'ndeki çalışma arkadaşlarıma da teşekkür ederim.

# ÖZET

ÇOK KERNELLİ EN BÜYÜK ARALIK ÖBEKLEMESİ PROBLEMİNİN YARI – SONSUZ OPTİMİZASYON İLE MODELLENMESİ

Gülnur SEICHAN OGLOU

Matematik Ana Bilim dalı

Uygulamalı Matematik

Tez Danışmanı: Yrd. Doç. Dr. Süreyya ÖZÖĞÜR AKYÜZ

Nisan 2012, 105 sayfa

Makine öğrenmesi, yapay zekâ bilim dalının alt konusu olup, çok sayıda verinin insan kapasitesi ile analiz edilemeyeceği durumlarda, veri analizinde ve veriden anlamlı davranışlar ve kurallar çıkararak ileriye doğru tahminler yapabilen, algoritmalar geliştiren bilim dalıdır. Günümüzde teknolojinin verdiği kolaylık ile çok sayıda veri elde edilebilmekte ve problemi çözmek için farklı kaynaklardan verilere ihtiyaç duyulmaktadır. Kernel algoritmaları verinin kendi içinde benzerliğini ölçen yöntemlerdir. Kernel yöntemleri ile öğrenme algoritmalarının sınıflandırma ve öbekleme problemlerinde başarılı oldukları, literatürde gerçel veri üzerinde kanıtlanmıştır.

Bu tezin konusu olan eğiticisiz öğrenme yöntemlerinden biri olan öbekleme yöntemi kernel algoritmaları ve optimizasyon teorisi ile geliştirilmiştir. Farklı kaynaklardan gelen verinin öbeklenmesinde, çoklu kernellerden yararlanılarak en büyük aralık ile öbekleme problemi yarı – sonsuz programlama kullanılarak modellenmiştir. Bu çalışma literatüre iki farklı katkıda bulunmuştur: 1) Çok kernelli en büyük aralık ile öbekleme probleminde kernel katsayıları $l_1$– norma kısıtlanarak seyrek (*sparse*) çözümler üretebilirken, ağırlık vektörleri $l_2$ – norma cezalandırılmıştır. 2) En büyük aralık ile çok kernelli öbekleme problemi Yarı – Sonsuz Programlama ile modellenmiştir.



**Anahtar Kelimeler:** Makine Öğrenmesi, Öbekleme, Destek Vektör Makineleri, En Büyük Aralık ile Sınıflandırma, Kernel, Optimizasyon, Yarı – Sonsuz Programlama, Çok Kernelli Öğrenme.

# ABSTRACT

MAXIMUM MARGIN MULTIPLE KERNEL CLUSTERING by SEMI - INFINITE PROGRAMMING

Gülnur SEICHAN OGLOU

Department of Mathematics

Applied Mathematics

Thesis Supervisor: Yrd. Doç. Dr. Süreyya ÖZÖĞÜR AKYÜZ

April 2012, 105 pages

Machine learning is the subfield of the artifical intelligence which finds the significant behaviours or functions from the data for future predictions where the human capacity does not allow to do. With the development of the high technology large number of data can be obtained and data from different sources are needed to model the real world problems. Kernel methods and learning algorithms are proven to be successful on real world data for the classification and clustering problems.

In this thesis, kernel clustering method is developed which is one of the unsupervised learning methods by using optimization theory. For the clustering of the data which come from different sources, maximum margin clustering is modelled with multiple kernel learning by using Semi – Infinite Programming. This study has two major contribution to the literature: 1) In maximum margin clustering by multiple kernel learning, kernel coefficients are constrained to $l_1$ – norm to produce a sparse solution while the weight vectors are penalized to $l_2$ – norm, 2) Maximum margin clustering by multiple kernel learning problem is modelled via Semi-infinite Programming.

**Keywords:** Machine Learning, Clustering, Support Vector Machines, Maximum Margin Classification, Kernel, Optimization, Semi – Infinite Programming, Multiple Kernel Learning.

## İÇİNDEKİLER

## ŞEKİLLER

## KISALTMALAR

| | | |
|---|---|---|
| CKO | : | Çok Kernelli Öbekleme |
| CURE | : | Clustering Using REpresentatives |
| DVM | : | Destek Vektör Makineleri |
| DBSCAN | : | Density – Based Spatial Clustering of Applications with Noise |
| EAO | : | En Büyük Aralık ile Öbekleme |
| EM | : | Expectation Maximization |
| GA | : | Genetik Algoritma |
| IUK | : | İkinci Derece Uygunluk Koşulu |
| KKT | : | Karush-Kuhn-Tucker |
| LYSP | : | Lineer Yarı Sonsuz Programlama |
| LBKK | : | Lineer Bağımsız Kısıt Koşulları |
| SOM | : | Self Organizing Maps |
| SP | : | Sonlu Programlama |
| VC | : | Vapnik ve Chervonenkis |
| VM | : | Veri Madenciliği |
| YSA | : | Yapay Sinir Ağı |
| YSP | : | Yarı Sonsuz Programlama |

## SEMBOLLER

| | | |
|---|---|---|
| Girdi vektörü | : | $\mathbf{x}_i$ |
| Çıkış etiketi | : | $y_i$ |
| Hiperdüzlem denklemi | : | $f(\mathbf{x})$ |
| Ağırlık terimi | : | $\mathbf{w}$ |
| Sapma terimi | : | $b$ |
| Langrange fonksiyonu | : | $L(w, b)$ |
| Langrange çarpanları | : | $\alpha_i$ |
| Serbestlik değişkenleri | : | $\xi_i$ |
| Taşıma fonksiyonu | : | $\mathbf{\Phi}$ |
| Kernel fonksiyonu | : | $K\ (.)$ |
| Sınıf sayısı | : | $k$ |
| Eğitim örnekleri sayısı | : | $n$ |
| Reel sayılar kümesi | : | $R$ |
| *n* boyutlu reel vektör uzayı | : | $R^n$ |
| $n \times n$ boyutlu reel elemanlı matrislerin kümesi | : | $R^{n\,\mathrm{x}\,n}$ |
| $f(x)$ fonksiyonunun gradyant vektörü | : | $\nabla f\ (x)$ |
| $f(x)$ fonksiyonunun hessian matrisi | : | $\nabla^2 f\ (x)$ |
| *A* matrisinin transpozesi | : | $A^T$ |
| Köşegenleştirilmiş *A* matrisi | : | $diag(A)$ |
| $n \times n$ boyutlu birim matris | : | $I_{n \times n}$ |
| Uygun bölge | : | $\mathcal{M}$ |
| Teğet uzayı | : | $T$ |
| Tanjant uzayı | : | $\mathcal{T}(\ y^*)$ |
| Yakınlık matrisi | : | $D_{ij}$ |
| Uzaklık fonksiyonu | : | $d(i, j)$ |

# 1. GİRİŞ

Makine öğrenmesi, öğrenme işinin bilgisayar tarafından gerçekleşmesinin sağlanmasıdır. Böylece, öğrenen makine geleceğe ait olayları, geçmiş yaşanmışlıkları inceleyerek karar verebilir. O halde makine öğrenmesinin temelinde, geçmiş bilgi ve gözlemlerden hareketle, gelecek hakkında bilgi vermek vardır.

Yeterince büyük veri ve parametrelerin varlığında, makine öğrenmesi, insanlar tarafından görülemeyen kuralları, kısıtları çıkarabilir. Bu da tez çalışmasında, neden makine öğrenmesi yöntemlerinin kullanıldığının sebebi olarak belirtilebilir. Makine öğrenmesi, tıpta vücuttan alınan biyolojik işaretlerin analizinde, telefonlarda veya bilgisayarlarda ses tanıma düzeneklerinde, yüz, retina veya parmak izi tanıyan güvenlik sistemlerinde, görerek objeleri tanıyan ve işlem yapan robot veya otomatlarda, uzaktan algılanan görüntülerin analizinde, sıklıkla kullanılmaktadır.

Eğiticili ve eğiticisiz öğrenme, makine öğrenmesinin tekniklerindendir. Eğiticisiz öğrenme yöntemi olan öbekleme, bir eğiticiye ihtiyaç olmadan ve verilerin sınıf etiketleri bilinmeden, öbek içindeki benzerliklerin fazla, öbek dışındaki benzerliklerin az olma prensibine göre çalışır. Öbekleme algoritmaları, büyük ölçekteki gen datalarında, mikro dizi veri analizlerinde yaygın olarak kullanılan bir yöntemdir. Bu tezin konusu olan en büyük aralık ile öbekleme problemini modellerken, ikili sınıflandırma problemi Destek Vektör Makineleri (DVM)'nden yola çıkılmıştır. Bu nedenle, makine öğrenmesi yöntemlerinden biri olan DVM üzerinde yoğunlaşılmış, lineer ve lineer olmadığı durumlarda çözüm yolları hakkında bilgiler verilmiştir. Sonrasında çok kernel ile sınıflandırma problemi tanıtılmıştır. İkili sınıflandırma problemi olan DVM'de sınıflar arası aralığın maksimum olması ile daha iyi sonuçlar alındığı gözlemlenmiştir. Buradan hareketle çok kernelli en büyük aralık öbekleme problemi oluşturulmuş, yarı – sonsuz programlama (optimizasyon) ile modellenmiştir.

Bu tez çalışması ile Zhao ve diğ. (2009) tarafından önerilen çok kernelli öbekleme yöntemi geliştirilmiş ve Yarı-sonsuz Programlama (YSP) ile modellenmiştir. Bahsedilen çalışmada Zhao ve arkadaşları geliştirilen optimizasyon problemini kesen düzlem (*cutting plane*) metodu ile çözmüşlerdir. Bu tezde yarı sonsuz olarak modellenen problem için en iyilik koşulları incelenmiştir. Bu inceleme sonucunda Ansatz azaltıcı gerek koşulu sağlandığı görülmüş, problem sonlu optimizasyon problemine dönüştürülmüştür. Problemi sağlayan yerel minimum noktasının bulunması için algoritmanın kaynak kodu Bölüm 4'de Algoritma 1 ile verilmiştir.

Çalışmanın ikinci bölümünde matematiksel hazırlık olarak optimizasyon teorisi hatırlatılmıştır. Tezin konusu olan problemin çözümünde yararlanılan Lagrange fonksiyonları, optimizasyon probleminin dual formu, yarı sonsuz optimizasyon ve kısıt koşulları da bu bölümün altında incelenmiştir.

Üçüncü bölümde makine öğrenmesi, makine öğrenmesi tekniklerinden eğiticili ve eğiticisiz öğrenme hakkında bilgi verilmiştir. Alt bölümlerde ise sınıflandırma ve öbekleme tanıtılmıştır. Sınıflandırma yöntemlerinden bu tezin modellenmesinde önemli rol oynayan Destek Vektör Makineleri ve optimizasyon probleminin çözüm yolları detaylı bir şekilde anlatılmıştır. Son olarak tezdeki problemin ana konusu olan öbeklemenin tanımı ve öbekleme yöntemlerinin bir kısmı incelenmiştir.

Dördüncü bölümde çok kernelli sınıflandırma problemi en büyük aralık ile öbekleme prensibi kullanılarak modellenmiştir. Problem, sınırlı değişkene ve sonsuz kısıta sahip olduğundan optimizasyon yöntemlerinden yarı – sonsuz programlama ile ifade edilip, lineer bağımsız kısıt koşulu, Karush Kuhn Tucker (KKT) koşulu ve ikinci dereceden koşullar incelenmiştir. Problemi sağlayan algoritmanın kaynak koduna da yine bu bölümde yer verilmiştir.

Beşinci bölümde ise sonuç ve tartışma yer almaktadır.

# 2. MATEMATİKSEL ALTYAPI

Bu bölümde, Bölüm 3 ve Bölüm 4'e hazırlık amaçlı bazı temel kavramlar, tanım ve teoremlere yer verilmiştir. Öbekleme probleminin modellenmesinde yararlanılacak, optimizasyon teorisinden, Lagrange teorisinden, yarı-sonsuz programlamadan bahsedilmiştir.

## 2.1 OPTİMİZASYON TEORİSİ

Optimum, kelimesi latince nihai, ideal anlamına gelmektedir. Optimizasyon ise bir problemde belli koşullar altında mümkün olan alternatifler içinden en iyisini seçmektir. Diğer bir ifade ile optimizasyon, verilen kısıtlar altında fonksiyonun maksimumunu veya minimumunu bulmaktır. Optimizasyon modelleri, bir problemin en iyi çözümünü bulmayı amaçlamaktadır. Bu, bir firmanın kârını maksimize etmek, zararı veya riski minimize etmek olabilir.

Optimizasyon teorisi, problemlerin çözümlerini karakterize etme üzerine yoğunlaşan ve onlar için verimli algoritmalar, modellemeler geliştiren uygulamalı matematiğin bir dalıdır. Dolayısıyla makine öğrenmesi problemi, optimizasyon teorisinin çerçevesi içinde analiz edilecek hale çevrilmiştir. Optimizasyon teorisi bize sadece algoritmik teknikler sağlamakla kalmaz, çözüm için gerekli ve yeterli şartları da açıklar. Optimizasyon problemleri kısıtlı veya kısıtsız optimizasyon problemleri olmak üzere ikiye ayrılır.

Bu tezdeki problemin kısıtlı olmasından dolayı bu bölümün devamında kısıtlı optimizasyon hakkında bilgi verilecektir. Optimizasyon problemlerinin çözüm yollarından bahsetmeden önce konu ile ilgili bazı temel tanım ve teoremler verilecektir.

**Tanım 2.1.1:** Reel tanımlı $f : R^n \to R$ olacak şekilde bir $f = f\,(x_1,\, x_2, \dots, x_n)$ fonksiyonunu ele alalım. Bu fonksiyonun ekstremumunun (maksimum veya minimum) olması için gerek şart veya birinci mertebeden şart, $\nabla f_i(x^*) = \frac{\partial f}{\partial x_i} = 0 \;\; (i = 1, \dots, n)$ eşitliğini sağlayan $x^* \in R^n$ noktasının var olmasıdır. Bu şartı sağlayan noktaya $f$

fonksiyonunun durağan *(stationary) noktası* veya *kritik noktası* adı verilir (Chiang 1999).

**Tanım 2.1.2:** Bir $A$, $n\,x\,n$ boyutlu simetrik matrisinin pozitif tanımlı oması için gerek şart $x = (\,x_1,\; x_2, \dots, x_n\,)^T$ olmak üzere, $\forall\, x\; \neq 0$ için $x^T A\, x > 0$ olmasıdır. Benzer şekilde pozitif yarı tanımlılık, negatif tanımlılık ve negatif yarı tanımlılık da tanımlanabilir. Buna göre,

a) $\forall x \neq 0$ vektörü için $x^T\; A\; x > 0$ ise pozitif tanımlıdır.

b) $\forall x \neq 0$ vektörü için $x^T\; A\; x\; \geq 0$ ise pozitif yarı tanımlıdır.

c) $\forall x \neq 0$ vektörü için $x^T\; A\; x \leq 0$ ise negatif yarı tanımlıdır.

d) $\forall x \neq 0$ vektörü için $x^T\; A\; x\; < 0$ ise negatif tanımlıdır.

e) Bazı $x\; \neq 0$ vektörü için $x^T\; A\, x\; > 0$ ve geri kalan $x\; \neq 0$ vektörü için $x^T\; A\, x < 0$ ise tanımsızdır (Nash & Sofer 1996).

Teorem 2.1.3.'de belirtildiği gibi simetrik matrisin öz değerlerinden yararlanarak da matrisin pozitif, negatif veya yarı tanımlı olup olmadığı bulunabilir.

**Teorem 2.1.3:** *A*, *n* x *n* boyutlu simetrik matris olsun. Buna göre,

a) $A$ matrisinin öz değerlerinin tümü pozitif ise pozitif tanımlıdır.
b) $A$ matrisinin öz değerlerinin tümü negatif değil ise pozitif yarı tanımlıdır.
c) $A$ matrisinin öz değerlerinin tümü negatif ise negatif tanımlıdır.
d) $A$ matrisinin öz değerlerinin tümü pozitif değil ise negatif yarı tanımlıdır.
e) $A$ matrisi içinde en az bir negatif ve en az bir pozitif özdeğer varsa tanımsızdır.

Jakobyan ve Hessian matrisi kullanılarak aşağıdaki teoreme ulaşılmaktadır.

**Teorem 2.1.4:** İkinci dereceden türevlenebilen $f(x)$ fonksiyonunun $(f : R^n \rightarrow R)$ kritik noktası $x^*$ olsun ve $\nabla^2 f\;(x)$, $f\;(x)$ fonksiyonunun Hessian matrisi olsun. Bu şartlar altında *x** ,

a) $\nabla^2 f\;(x)$ pozitif yarı tanımlı ise, $f(x)$ için mutlak (global) minimumdur.
b) $\nabla^2 f\;(x)$ pozitif tanımlı ise, $f(x)$ için kesin mutlak (global) minimumdur.
c) $\nabla^2 f\;(x)$ negatif yarı tanımlı ise, $f(x)$ için mutlak (global) maksimumdur.

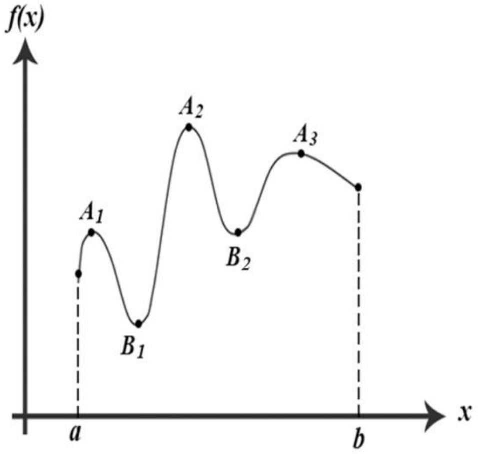
d) $\nabla^2 f\ (x)$ negatif tanımlı ise, $f(x)$ için kesin mutlak (global) maksimumdur.

e) $\nabla^2 f\ (x)$ tanımsız ise, $f(x)$ in büküm (eyer) (*saddle point*) noktasıdır.

**Teorem 2.1.5:** Reel tanımlı $f : R^n \rightarrow R$ fonksiyonunun kritik noktası $x^*$, $D \subset R^n$ bölgesinde bir iç nokta olmak üzere, öyle pozitif $\varepsilon$ sayısı vardır ki, her $x \in R^{\mathrm{n}}$ ve $\|x - x^*\| < \varepsilon$ için $f\ (x) \geq f\ (x^*)$ oluyorsa, bu nokta yerel (local) minimumdur. Eğer $\forall\, \varepsilon > 0$ , $x \in R^{\mathrm{n}}$ ve $x \neq x^*$ için $f\ (x^*) < f\ (x)$ ise $x^*$ noktasına $f$ 'nin kesin yerel (*strict local*) minimumu denir.

Teorem 2.1.4 ve Teorem 2.1.5'de açıklanan yerel ve mutlak minimumları Şekil 2.1'de incelemek mümkündür.

**Şekil 2.1: Yerel ve mutlak minimum**




| | |
|---|---|
| $A_1$, $A_2$, $A_3$ | : Yerel maksimum |
| $A_2$ | : Mutlak maksimum |
| $B_1$, $B_2$ | : Yerel minimum |
| $B_1$ | : Mutlak minimum |

*Kaynak:* KAYMAZ, Atatürk Ünv. Müh. Fak. Ders Notları Bölüm 3

### 2.1.1 Konveks Fonksiyonlar

Bu bölümde optimizasyon teorisinde önemli yeri olan konveks fonksiyonun tanımı verilmiştir. Konvekslik varsayımı, bizi fonksiyonların Hessian matrislerinin yapısını inceleme zorluğundan kurtarır (bknz Teorem 2.1.7). Konveks fonksiyon tanımından önce konveks kümeyi tanımlayalım.

**Tanım 2.1.3:** $C \in R^{\mathrm{n}}$'nin iç noktası olan her $A$ ve $G$ noktaları için, bu noktaları birleştiren doğru parçası Şekil 2.2.(a)'da gösterildiği gibi $C$ bölgesinin içinde kalıyorsa konveks kümedir. Aksi halde konveks küme değildir (Bknz Şekil 2.2.(b)).

**Tanım 2.1.4**: Konveks küme $C \subset R^{\mathrm{n}}$ ve $f : C \longrightarrow R$ fonksiyonu verilsin. Eğer, $\forall\, x, y \in C\,,\ 0 \leq \lambda \leq 1$ için

$$f\,(\lambda x + (1-\lambda)y) \leq \lambda f(x) + (1-\lambda)\, f(y) \qquad (\mathbf{2.1})$$

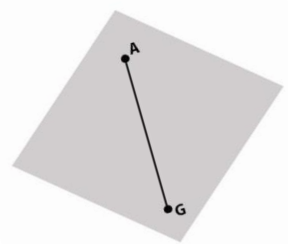
eşitsizliği sağlanıyorsa $f$ fonksiyonuna *konveks fonksiyon* denir.

Yukarıda verilen Tanım 2.1.3 ve Tanım 2.1.4'i $R^{\mathrm{n}}$ uzayında yazacak olursak eşitsizlik (2.1)'i aşağıda eşitsizlik (2.2)'de belirtildiği gibi yazmak mümkündür. Negatif olmayan $\lambda_1, \ldots,\ \lambda_k$ , sayıları için $\sum_{i=1}^{k} \lambda_i = 1$ ve $x_i \in R^{\mathrm{n}}$ de olmak üzere,

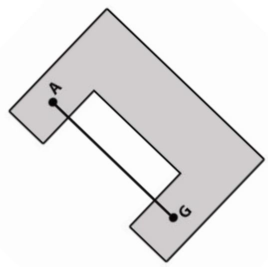
$$f\left(\sum_{i=1}^{k} \lambda_i x_i\right) \leq \sum_{i=1}^{k} \lambda_i\, f(x_i) \qquad (\mathbf{2.2})$$

eşitsizliğini yazabiliriz (Bertsimas & Tsitsiklis 1997).

**Şekil 2.2(a): Konveks küme**




**Şekil 2.2(b): Konveks olmayan küme**




**Teorem 2.1.6:** Farzedelim ki $f$ konveks bir fonksiyon olsun. Bu durumda herhangi bir yerel minimumu $x^*$ aynı zamanda $f$'nin bir mutlak minimumudur. Eğer $f$ fonksiyonunun türevi mevcut ise $f$'nin herhangi bir kritik noktası aynı zamanda bir mutlak minimumudur (Nocedal & Wright 1999).

Optimal noktayı bulmak adına bazı tanım ve teoremler, kısıtlı optimizasyon problemleri için Bölüm 2.2.1' de verilecektir.

## 2.2 KISITLI OPTİMİZASYON

Günlük hayatta karşılaşılan bir firmanın kârını maksimum veya ürettiği malın maliyetini minimum yapma ya da bir ülkenin büyüme hızını maksimum yapma vs. gibi problemler kısıtlara bağlı problemlerdir. Kârı maksimum yapma probleminde üretim zamanı bir kısıt, ürün için talep miktarı da diğer bir kısıt olarak yazılabilir. Farklı problemler için işgücü, finansman gibi koşullar kısıt örnekleri olarak verilebilir (Alan ve Yeşilyurt 1994). Amaç, değişik kısıtlar altında maksimizasyon veya minimizasyon yapmaktır. Bu işlemler, "en iyiyi aramak" anlamında optimizasyon genel başlığı altında toplanabilir. Bu bölümde kısıtlı optimizasyon problemi hakkında bilgi verilecek, optimal noktayı bulma koşulları ve dualite problemleri incelenecektir (Chiang 1999).

### 2.2.1 Kısıtlı Optimizasyon Problemi

Fonksiyonun verilen kısıtlar altında maksimumunu ya da minimumunu hesaplamak, kısıtlı optimizasyon problemi olarak adlandırılır (Nicholson 1995, Cristianini & Taylor 2000). Genel bir optimizasyon probleminin matematiksel ifadesi

$$\begin{array}{lll} \min\limits_{x} & f\,(x) & \\ \text{kısıtlar} & h_i(x) = 0 \quad (\, i = 1, 2, \dots, m\,), & \\ & g_j\,(x) \;\leq 0 \quad (\, j = 1, 2, \dots, p) & \end{array} \qquad (\boldsymbol{P})$$

ile gösterilir (Luenberger & Ye 2007).

Yukarıda $(P)$ ile tanımlanan optimizasyon probleminde $x \in R^n$, $n$ boyutlu bir vektördür, $x$ bilinmeyen olmak üzere karar değişkenini gösterir. Optimize edilecek (maksimum veya minimum) olan problemin amaç fonksiyonu $f : R^n \to R$ ile tanımlanır. Reel değerli tanımlanan, $h_i : R^n \to R$ *eşitlik kısıt* fonksiyonunu ve $g_j : R^n \to R$ *eşitsizlik kısıt* fonksiyonunu ifade eder. Kısıtları sağlayan $x \in R^n$ vektörleri arasından amaç fonksiyonunu en küçük yapacak $x^*$ vektörü bulabilirsek bu değere ***optimum değer*** adı verilir.

Kısıtları sağlayan herhangi bir noktaya uygun nokta (*feasible point*) denir. Amaç fonksiyonunun tanımlı olduğu ve tüm kısıtların sağlandığı, (bütün uygun noktaların oluşturduğu) eşitlik (2.3) ile verilen kümeye ***uygun bölge*** (*feasible region*) denir:

$$\mathcal{M} = \{ x : x \in R^n,\ h_i(x) = 0\,, g_j\,(x) \leq 0 \ (i = 1, 2, \ldots, m)\ (j = 1, 2, .., p) \}. \quad (\mathbf{2.3})$$

Bir optimizasyon probleminde amaç fonksiyonu, eşitlik ve eşitsizlik kısıtları lineer fonksiyon ise bu probleme *Lineer Programlama*, eğer amaç fonksiyonu karesel ve tüm kısıtlar lineer ise bu probleme *Karesel Programlama* adı verilir.

Kısıtlı optimizasyonda temel kavramlardan biri de ***aktif kısıt*** kavramıdır. Eğer çözüm $x^*$ ise, $x^*$ uygun noktasının aktif olabilmesi için gerek şart, eşitsizlik kısıtının $g_j(x^*) = 0$ $(j = 1,2, \ldots, p)$ eşitliğinin sağlamasıdır. Eğer $g_j(x^*) > 0$ ise eşitsizlik kısıtına ***aktif değildir*** denir. O halde, tanıma göre optimizasyon problemlerindeki $h_i(x^*) = 0$ $(i = 1,2, \ldots, m)$ eşitlik kısıtlarının her biri uygun noktada her zaman aktiftir. Bir eşitsizlik kısıtını eşitlik kısıtına dönüştürmek için ***gevşek değişken*** olarak adlandırılan $\xi_j$ değişkenlerinden yararlanılır:

$$g_j\,(x) \geq 0 \iff g_j\,(x) + \xi_j = 0\,, \quad \xi_j \geq 0.\,, \ j = 1,2, \ldots, .\, p\,. \qquad (\mathbf{2.4})$$

**Tanım 2.2.1:** Bir $x^*$ noktası $h_i(x^*) = 0$ ve $g_j(x^*) \leq 0$ kısıtlarını gerçekleyen nokta, $J$ ise $g_j(x^*) = 0$ koşulunu sağlayan $j$ indislerinin aktif kümesi olsun. $x^*$'ın düzgün (regular) bir nokta olması için gerek şart, $\nabla h_i\,(x^*)$, $\nabla g_j\,(x^*)$ ifadelerinin ($1 \leq i \leq m$, $j \in J$) lineer bağımsız olmasıdır (Luenberger & Ye 2007).

**Teorem 2.2.2:** Eşitlik kısıtı $h_i(x) = 0$ $(i = 1,2, \ldots, m)$ ile tanımlanan uygun bölge $\mathcal{M}$'nin içindeki bir $x^*$ düzgün noktasındaki teğet uzayı

$$T = \left\{ y : \sum_{i=1}^{m} \nabla h_i(x^*) y = 0 \right\} \qquad (\mathbf{2.5})$$

ile tanımlanır.

**Önerme 2.2.3**: Bir $x^*$ noktası, $h_i(x) = 0$ $(i = 1,2, \dots, m)$ kısıtının düzgün noktası (bknz Tanım 2.2.1) ve eşitlik kısıtlarına sahip bir optimizasyon probleminin yerel ekstremum (minimum yada maksimum) noktası olsun. O halde $\forall\, y \in R^n$ için

$$\nabla h\,(x^*)^T y = 0 \qquad ve \qquad \nabla f\,(x^*)^T y = 0$$

olmalıdır.

### 2.2.2 Lagrange Teorisi

Lagrange çarpanları yöntemi optimizasyon teorisinde önemli bir yer tutmaktadır. Lagrange çarpanı metodu kısıtlı optimizasyon problemlerinin çözüm yöntemlerinden birisidir (Nicholson 1995).

**Teorem 2.2.4:** Eşitsizlik kısıtlarının ve eşitlik kısıtlarının birarada verildiği problem $(P)$ için Lagrange fonksiyonu, amaç ve kısıt fonksiyonlarını içerecek biçimde denklem (2.6)'da verildiği gibi yazılır:

$$L\,(x, \lambda, \mu) := f\,(x) - \sum_{i=1}^{m} \lambda_i h_i(x) - \sum_{j=1}^{p} \mu_j g_j(x). \qquad (\mathbf{2.6})$$

Denklem (2.6)'da, $\lambda_i$ katsayıları eşitlik kısıtının, $\mu_j$ katsayıları ise eşitsizlik kısıtının Lagrange çarpanları, $h_i\,(x^*) = 0$ $(i = 1,2, \dots, m)$ eşitlik kısıtları ve $g_j(x^*) \geq 0$ $(i = 1,2, \dots, p)$ eşitsizlik kısıtlarıdır.

Optimizasyon teorisi, her problemin gerçekte özgün (*primal*) ve dual olmak üzere iki problemden ibaret olduğunu belirtmektedir. Her maksimizasyon modeline karşılık bir minimizasyon, her minimizasyon modeline karşılık bir maksimizasyon modeli vardır. Özgün (*primal*) ve dual problem arasındaki tüm ilişkiler simetrik olmalıdır. Çünkü dual problemin duali, özgün (*primal*) problemdir. İlk ele alınan modele özgün problem, buna karşılık gelen modele de dual model denir. Dolayısıyla özgün (*primal*) problemi çözmek belli koşullar altında dualinin çözümü ile aynı sonucu verecektir. Temel problemin en iyi (optimum) değeri her zaman için dual problemin en iyi değerinden büyük veya ona eşittir. Eğer kesin eşitsizlik durumu oluşursa, dual aralık (*duality gap*) oluştuğu söylenir. Dual aralık (*duality gap*) ile ilgili detaylı bilgi için Dattorro, 2005, *Convex Optimization & Euclidean Distance Geometry*, Mεβoo, incelenebilir.

**Teorem 2.2.5 (İkinci Mertebeden Gerek Şart):** İkinci dereceden türevlenebilir sürekli fonksiyonlar $f, g, h \in C^2$ ve $x^*$ düzgün bir nokta olsun. Eğer $x^*$ noktası problem $(P)$ için bir yerel minimumu ise $\lambda \in R^m$ ve $\mu \in R^p$, $\lambda \geq 0$, $\mu \geq 0$ vektörleri için,

$$\nabla f\,(x^*) + \sum_{i=1}^{m} \lambda_i\, \nabla h_i\,(x^*) + \sum_{j=1}^{p} \mu_j\, \nabla g_j\,(x^*) = 0,$$

$$\mu_j\, g_j\,(x^*) = 0,$$

koşulları sağlanır. Lagrange denkleminin Hessian fonksiyonu

$$\nabla^2 L(x^*) := \nabla^2 f\,(x^*) - \sum_{i=1}^{m} \lambda_i\, \nabla^2 h_i\,(x^*) - \sum_{j=1}^{p} \mu_j\, \nabla^2 g_j\,(x^*)$$

ile yazılmış olur. Eşitsizlik kısıtlarının Hessian matrisi $\nabla^2 g_j(x^*)$ olmak üzere, $x^*$ noktasındaki aktif kısıtlar teğet alt uzayında pozitif yarı tanımlı olmalıdır.

**Teorem 2.2.6 (İkinci Mertebeden Yeter Şart):** İkinci dereceden türevlenebilir sürekli fonksiyonlar $f, g, h \in C^2$ olsun. $x^*$ noktasının problem $(P)$ için yerel minimum olması için yeter şart, $\lambda \in R^m$ ve $\mu \in R^p$ vektörleri için,

$$\mu \geq 0,$$

$$\mu_j\, g_j\,(x^*) = 0,$$

$$\nabla f\,(x^*) - \sum_{i=1}^{m} \lambda_i\, \nabla h_i\,(x^*) - \sum_{j=1}^{p} \mu_j\, \nabla g_j\,(x^*) = 0$$

ve indis kümesi $J = \{ j\!: \ g_j(x) = 0,\ \mu_j > 0 \}$ olmak üzere Lagrange denkleminin Hessian matrisi

$$\nabla^2 L(x^*) = \nabla^2 f\,(x^*) - \sum_{i=1}^{m} \lambda_i\, \nabla^2 h_i\,(x^*) - \sum_{j=1}^{p} \mu_j\, \nabla^2 g_j\,(x^*),$$

tanjant uzayı $\mathcal{T} = \{ y\colon \nabla h\,(x^*)y = 0,\ \nabla g_j(x^*)y = 0,\ \forall j \in J \}$ üzerinde pozitif tanımlı olmalıdır.

**Teorem 2.2.7 (Karush Kuhn Tucker Koşulları):** Farz edelim ki $x^*$ noktası problem $(P)$'nin minimumu ve verilen kısıtlar üzerinde düzgün bir nokta olsun. O halde öyle $\lambda \in R^m$ ve $\mu \in R^p$ vektörleri vardır ki,

a) $\mu \geq 0,$

b) $\nabla f\,(x^*) - \sum_{i=1}^{m} \lambda_i \nabla h_i\,(x^*) - \sum_{j=1}^{p} \mu_j \nabla g_j\,(x^*) = 0,$

c) $\mu_j\, g_j\,(x^*) = 0$

olmalıdır.

## 2.3 YARI SONSUZ PROGRAMLAMA – YSP (Semi – Infinite Programming)

Bölüm 4.2.3'de öbekleme problemi, yarı sonsuz problem ile modelleneceği için bu bölümde Yarı-Sonsuz Programlama (YSP) tanıtılmış, gerekli tanım ve teoremlerden bahsedilmiştir. YSP optimizasyon problemlerinin bir sınıfı olup, "Yarı-sonsuz" isminden de anlaşılacağı gibi sonsuz sayıda kısıtları ve sonlu sayıda değişkenleri vardır.

**Tanım 2.3.1:** Yarı sonsuz programlama (YSP) sonlu tane eşitlik kısıtlar ve sonsuz tane eşitsizlik kısıtları tarafından tanımlanan ve sonlu boyutlu değişkene ($x \in R^n$) sahip olan bir optimizasyon problemidir:

$$\begin{array}{lll} \min\limits_{x} & f\,(x) & \\ \text{kısıtlar} & h_i(x) = 0 & (i \in I), \\ & g(x,y) \geq 0 & (y \in Y). \end{array} \qquad (\boldsymbol{YSP})$$

Burada, $f\colon R^n \to R$, $g\colon R^n \text{ x } R^m \to R$, $I := \{1,2,\dots,s\}$ sonlu indeks kümesi, $X \subseteq R^n$, $Y \subseteq R^m$ sonsuz indeks kümesi ve $y := (y_1, y_2, \dots, y_m)^T$'dir.

$\mathcal{M}$ uygun (*feasible*) kümeyi göstermek üzere,

$$\mathcal{M}:= \{ x \in R^n \mid h_i(x) = 0 \ (i \in I) \ \ ve \ \ g(x,y) \geq 0 \ \ (y \in Y)\}, \qquad (\mathbf{2.7})$$

(YSP) probleminin minimum değerlerinin oluşturduğu küme $\mathcal{G} := \{x^* \in \mathcal{M} \mid f(x^*) = v\}$ ve optimum değer $v := \ inf \ \ \{ f(x) \mid x \in \mathcal{M} \}$ ile gösterilsin.

**Kısıt koşulları:**

Bu alt bölümde, Sonlu Programlama (SP), Yarı-Sonsuz Programlama (YSP) ve Lineer Yarı-Sonsuz Programlama'nın (LYSP) uygun kümeleri (*feasible set*) $\mathcal{M}$ tanımlanmış, sonlu programlama ve yarı sonsuz programlama arasındaki farkı göstermek için her ikisinin de kısıt koşullarından bahsedilmiştir.

Yarı sonsuz programlamada (YSP), uygun küme $\mathcal{M}$,

$$\mathcal{M} := \{ x \in R^n \mid h_i(x) = 0 \ (i \in I) \quad g(x,y) \geq 0 \ \ (y \in Y)\}, \qquad (\mathbf{2.8})$$

şeklinde tanımlanır.

Problem (YSP)'de verilen $I$' nın sonlu indeks kümesi, $Y \subset R^m$ ise sonsuz indeks kümesini ifade eder. Farz edelim ki $g \in C^2: R^n \times R^n \ \to R$ ve $h \in C^2: R^n \ \to R$ olsun.

Lineer yarı-sonsuz programlama – LYSP (L*inear semi – infinite programming–LSIP) da* uygun küme $\mathcal{M}$,

$$\mathcal{M} = \left\{ x \in R^m \ \middle| \ a_y^T x \geq b_y \ (y \in Y) \right\}, \qquad (\mathbf{2.9})$$

şeklinde tanımlanır.

Denklem (2.9)'da $y \ \in \ Y$, $a_y \in R^n$ ve $b_y \in R$ fonksiyonlarıyla birlikte $Y \ \subset R^m$ sonsuz indeks kümesidir. Yukarıda denklem (2.9) ile verilen tanımda, eşitlik kısıtları problem tanımının dışında tutulmuştur.

Sonlu progralamada (SP) ise uygun küme $\mathcal{M}$, denklem (2.10) ile tanımlanır:

$$\mathcal{M} = \{ x \in R^n \mid h_i(x) = 0 \ (i \in I) \ \ ve \ g_j(x) \geq 0 \ (j \in J) \}, \qquad (\mathbf{2.10})$$

öyle ki $J = \{1,2,\dots,m\}$ ve $I = \{1,2,\dots,s\}$ sonlu indeks kümeleri ve $h_i \ (i \in I)$ ve $g_j : R^n \rightarrow R \ \ (j \in J)$ ikinci dereceden türevlenebilir sürekli fonksiyonlardır.

Şimdi sonlu ve yarı sonsuz programlama için kısıt koşullarını tanımlayalım.

### 2.3.1 Sonlu Programlamada (SP) Kısıt Koşulları

**Tanım 2.3.2:** Eğer $x^* \in \mathcal{M}$ ve $\nabla h_i(x^*) \ \ (i \in I)$, $\nabla g_j(x^*) \ \ (j \in J_0(x^*))$, gradyant vektörleri lineer bağımsız bir aileden ise, Lineer Bağımsız Kısıt Koşulları (LBKK) uygun bir $x^*$ noktasında sağlanır. Burada $J_0(x^*)$ aktif indeks kümesi

$$J_0(x^*) := \{ j \in J \mid g_j(x^*) = 0 \}, \qquad (\mathbf{2.11})$$

ile tanımlanır.

### 2.3.2 Yarı Sonsuz Programlamada (YSP) Kısıt Koşulları

Eğer $x^* \in \mathcal{M}$ noktasında, $\nabla h_i(x^*) \ \ (i \in I)$, $\nabla_x g(x^*, y) \ \ \big(y \in Y_0(x^*)\big)$ vektörleri, lineer bağımsız bir aileden ise LBKK sağlanır. Öyle ki, aktif indeks kümesi

$$Y_0(x^*) := \{ y \in Y \mid g(x^*, y) = 0 \} \qquad (\mathbf{2.12})$$

ile tanımlanır. Dikkat edelim ki LBKK $x^*$ noktasından sağlanırsa, aktif $Y_0(x^*)$ kümesinin $n$ sayıda elemandan daha fazlasını bulunduramayacağını biliyoruz. Burada $n, R^n$ vektör uzayının boyutudur.

Aşağıda Bölüm 2.3.4'de, YSP'nin ilk optimum koşulları için yerel ve mutlak küçültücülerin tanımları ve YSP için optimal noktanın varlığını Ansatz Azaltıcı koşulları ile garantileyen ve YSP problemini sonlu probleme indirgeyerek çözmemizi sağlayacak teorem ile devam edeceğiz (Weber 1992, Still 2004).

### 2.3.3 Birinci Dereceden En İyilik Koşulları (First – Order Optimality Conditions)

**Tanım 2.3.3:** Eğer bazı $\varepsilon > 0$ ve $x^* \in \mathcal{M}$ uygun (feasible) noktası için ($\|\cdot\|_2$ öklid normu olmak üzere),

$$f(x) - f(x^*) \geq 0 \qquad x \in \mathcal{M} \quad ve \quad \| x - x^* \|_2 < \varepsilon \quad \forall x^* \in \mathcal{M} \qquad (\mathbf{2.13})$$

ise, $x^*$ noktası YSP' nin yerel minimumu olarak adlandırılır.

**Tanım 2.3.4:** Eğer denklem (2.13) herhangi bir $\varepsilon > 0$ için sağlanıyorsa, $x^*$ mutlak minimumdur.

**Tanım 2.3.5:** Kritik nokta $x^* \in \mathcal{M}, p > 0$ olmak üzere, $f(x) - f(x^*) \geq q \| x - x^* \|_2{}^p$, $\forall x \in \mathcal{M}$ ve $\| x - x^* \|_2 \leq \varepsilon$ eşitsizliklerini sağlayacak bazı $q > 0$ ve $\varepsilon > 0$ var ise $x^* \in \mathcal{M}$ problem (YSP)'nin mutlak yerel minimumudur.

**Teorem 2.3.6 (Birinci Dereceden Yeterlilik Koşulu – First-Order Sufficient Condition):** Varsayalım ki, $x^* \in \mathcal{M}$, problem (YSP) için uygun nokta olsun öyle ki aşağıdaki koşulu karşılayan $0 \neq d \in R^n$ vektörü var olmasın:

$$\nabla f(x^*)d \leq 0, \; \nabla_x^T h_i(x^*)d = 0 \; (i \in I), \qquad \nabla_x^T g(x^*, y)d \geq 0 \; \big(y \in Y_0(x^*)\big).$$

O zaman $x^*$, problem (YSP)' nin $p = 1$ için kesin yerel minimumudur.

**Teorem 2.3.7 (Birinci Dereceden Gereklilik Koşulu – First-Order Necessary Condition):** Varsayalım ki, $x^* \in \mathcal{M}$, (YSP)'nin lokal minimumu olsun. O halde takip eden KKT koşulu aşağıdaki ifade ile verilebilir.

**KKT Koşulu:** Eğer Mangasarian Fromovitz Kısıt Yeterliliği (MFCQ) (*Mangasarian Fromovitz Constraint Qualification*) (Hettich & Kortanek 1993, Weber 2003, Still 2004) $x^* \in \mathcal{M}$ noktasında sağlanıyor ise, mevcut çarpanlar $\lambda_1, \lambda_2, \ldots, \lambda_s \in$ R ve $\mu_1, \mu_2, \ldots, \mu_k \geq 0$ ve $y_1, y_2, \ldots, y_k \in Y_0(x^*)$, $k \leq n$ olmak üzere,

$$\nabla f(x^*) - \sum_{i=1}^{s} \lambda_i \nabla_x^T h_i(x^*) - \sum_{j=1}^{k} \mu_j \nabla_x^T g\left(x^*, y_j\right) = 0 \qquad \mathbf{(2.14)}$$

eşitliği sağlanır.

### 2.3.4 İkinci Dereceden Uygunluk Koşulları – IUK *(Second-Order Optimality Conditions – SOC)*

Bu alt bölümde, indirgeme (*reduction*) yaklaşımını uygulayarak problem (YSP)'de verilen yarı – sonsuz denklemler için **İkinci Derece Uygunluk Koşulu** (IUK) tanıtılacaktır.

Problem (YSP)'nin uygun noktası $x^* \in \mathcal{M}$ olduğunu varsayalım öyle ki $u, v, f, g, h \in C^2$ ve sonsuz indeks kümesi $Y$, $u_k$ eşitlik fonksiyonu, $v_l$ eşitsizlik fonksiyonları ile birlikte çözüm kümesi olarak aşağıdaki gibi tanımlansın:

$$Y = \{ y \in R^m \mid u_k(y) = 0 \ (k \in K), v_l(y) \geq 0 \ (l \in L)\}. \qquad \mathbf{(2.15)}$$

Burada $K := \{1,2,\ldots,r\}$ ve $L := \{1,2,\ldots,q\}$ olmak üzere tanım gereği, $Y_0(x^*)$ dan her hangi bir aktif $y^*$ noktası aşağıda denklem (2.16) ile gösterilen parametrik optimizasyon probleminin bir mutlak minimumudur:

$$\min_{y} \quad g(x^*, y) \qquad \boldsymbol{Q(x^*)}$$

$$\text{kısıtlar} \quad u_k(y) = 0 \ (k \in K), \ v_l(y) \geq 0 \ (l \in L). \qquad \mathbf{(2.16)}$$

Problem $Q(x^*)$ ***düşük düzeyli problem*** (*lover level problem*) olarak adlandırılır. Burada $x^*$ parametresine bağlı aktif indeks kümesi $L_0(y^*)$ ile gösterilsin ve problem $Q(x^*)$ in aktif kısıtlarının kümesi olarak tanımlansın. Örneğin,

$$L_0(y^*) = \{l \in L \mid v_l(y^*) = 0\}.$$

Her $x^* \in \mathcal{M}$ noktası ve $y^* \in Y_0(x^*)$ olmak üzere $y^*$ düşük düzey (lower level) probleminin lokal minimumu iken $Q(x^*)$ için Lagrange fonksiyonunu yazalım:

$$\mathcal{L}(x,y,\zeta,\gamma) := g(x,y) - \sum_{k\in K} \zeta_k u_k(y) - \sum_{l\in L_0(y^*)} \gamma_l v_l(y) \qquad \mathbf{(2.17)}$$

öyle ki, $\zeta_k$ $(k \in K)$ ve $\gamma_l$ $\big(l \in L_0(y^*)\big)$ Lagrange çarpanları olsun.

Şimdi, optimum ve tek bir çözüm elde etmek için çok önemli varsayımlar içeren Ansatz Azaltıcısı (*Reduction Ansatz*) tanımlanacaktır.

**Tanım 2.3.8 (Ansatz Azaltıcısı (*Reduction Ansatz*)):** Herhangi bir $y^* \in Y_0(x^*)$ olmak üzere problem $Q(x^*)$'in yerel minimumu $y^*$ iken tek optimum noktanın varlığı için aşağıdaki Ansatz Azaltıcı koşulları sağlanmalıdır:

**i.** LBKK (LICQ): $\nabla_y u_k(y^*)$, $\nabla_y v_l(y^*)$ $(k \in K)$, $l \in L_0(y^*)$ lineer bağımsız kümedir,

**ii.** KKT Koşulları: LBKK (LICQ) nin sağlanması durumunda,

$$\nabla_y \mathcal{L}(x^*, y^*, \zeta^*, \gamma^*) = 0$$

eşitliğini sağlayan tek bir $\zeta^*$ ve $0 \le \gamma^* \in R^{|L_0(y^*)|}$ çarpanları vardır (Burada varsayalım ki, $\gamma_l > 0$ $(l \in L_0(y^*))$ olsun (kesin tümleyici gevşeklik)),

**iii.** İkinci mertebe uygunluk koşulu (IUK) (SOC): Özellik ii.'deki $\gamma^*$ ile

$\eta^T \nabla_y^2 \mathcal{L}(x^*, y^*, \zeta^*, \gamma^*)\eta > 0$, her $\eta \in \mathcal{T}(y^*)\backslash\{0\}$ için sağlanır,

öye ki $y^*$ noktasında tanjant uzayı,

$$\mathcal{T}(y^*) = \{\gamma \in R^m \,|\, \nabla_y^T u_k(y^*)\eta = 0 \ (k \in K), \quad \nabla_y^T v_l(y^*)\eta = 0 \quad \big(l \in L_0(y^*)\big)\}$$

ile tanımlanır.

Ansatz Azaltıcısındaki varsayımlar altında aşağıdaki teoremi verebiliriz (Hettich &

Jongen 1978, Hettich & Kortanek 1993, Hettich & Zencke 1982, Weber 2003, Still 2004).

**Teorem 2.3.9 (**Hettich & Jongen 1978, Hettich & Kortanek 1993, Hettich & Zencke 1982, Weber 2003, Still 2004, Wetterling 1970**):** Ansatz Azaltıcı koşulları uygun $x^*$ noktasında problem (YSP) için $Y_0(\,x^*)$ aktif indeks kümesinde ya da başka bir ifade ile problem $Q(\,x^*)$'in yerel minimum kümesinde sağlanmış olsun. Problem (2.15)'de verilen *Y* sonsuz indeks kümesi tıkız (compact) ise aşağıdaki sonuçlar elde edilir:

**1.** $Y_0(\,x^*) := \{\mathrm{y}_1^*,\ \mathrm{y}_2^*, \dots, \mathrm{y}_r^*\}$ aktif indeks kümesinin sonlu sayıda elemanı vardır ve sırası ile $x^*$ ve $y_j^*$ nin mevcut komşuları $U_{x^*}$ ve $V_{y_j^*}$ vardır. Bunlara karşılık gelen Lagrange çarpanları $\zeta_j^*$ ve $\gamma_k^*$ için,

$$y_j : U_{x^*} \rightarrow V_{y_j^*},\ \text{ile}\ \ y_j(\,x^*) = y_j^* \quad (\,j\ =\ 1,2, \dots, r\,),$$

$$\zeta_k\ :\ U_{x^*}\ \rightarrow\ R,\ \ \zeta_k^j(\,x^*)\ =\ \zeta_k^*\ \ (k\ \in\ K)\ \text{ve}$$

$$\gamma_j : U_{x^*}\ \rightarrow\ R\ \ \text{ile}\ \ \ \gamma_l^j(\,x^*)\ =\ \gamma_j^*,\ (\,l\ \in\ L_0(\,x^*)\,)$$

bağıntıları sağlanır.

Böylece, Lagrange çarpım vektörleri $\gamma_l^j(x)$ ve $\zeta_k^j(x)$ ile her $x\ \in\ U_{x^*}$ için $y_j(x)$ değeri $V_{y_j^*}$ komşuluğunda $Q(x)$'in benzersiz (tek) yerel minimumudur.

**2.** Birinci koşulda elde edilen fonksiyonlarla birlikte $x\ \in U_{x^*}\ \cap\ \mathcal{M}$ problem (YSP)'nin yerel minimumudur ancak ve ancak $x$, indirgenmiş problem $(P_{red}(\,x^*))$'in yerel çözümü ise:

$$\min_{x\in U_{x^*}} f(x) \qquad \boldsymbol{(P_{red}(\,x^*)}$$

$$\text{kısıtlar,}\quad h_i(x) = 0\ \ (i \in I),$$

$$\tilde{g}_j(x) := \ g\left(\,x, y_j(x)\right) \geq\ 0 \quad (j\ =\ 1,2, \dots, r).$$

**3.** $h_i(x)\ (i\ \in\ I)$ ve $\tilde{g}_j(x)\ =\ g(\,x, y_j(x))\ \ (\,j\ =\ 1,2, \dots, r)$ fonksiyonları $U_{x^*}$ de (ya da içinde) $C^2$ – fonksiyonlarıdır.

# 3. MAKİNE ÖĞRENMESİ

Bu bölümde, makine öğrenmesi kavramı üzerinde durularak, öğrenme stratejilerinden genel olarak bahsedilmiştir. Bundan sonraki bölümlerde, vektörlerin gösterimi kalın harf ile yapılacaktır.

## 3.1 GİRİŞ

Bilgisayarları akıllı yapma bilimi diye kısaca özetleyebileceğimiz yapay zekâ (*Artificial Intelligence*), Bilgisayar Bilimleri, Psikoloji, Felsefe, Dilbilim, Nöroloji vb. bilimsel disiplinlerin ortak çalışma alanıdır. Makine ögrenmesi ise bilgisayarlara (makine) ögrenme yetisi kazandırma amaçlı algoritmalar geliştiren akademik disiplin ve yapay zekânın bir alt alanıdır.

1945 yıllında I. nesil bilgisayarların ortaya çıkmasıyla bilim insanları makinelerin zekâ ile donatılmasıyla ilgilenmeye başladılar. 1950'de matematikçi Alan Mathison Turing, ‘Makineler düşünebilir mi?’ sorusu için geliştirdiği Turing testi ile bilgisayarların düşünme yetileri konusunda bir kriter öne sürmüştür. Böylece Turing makinesi denilen algoritma ile modern bilgisayarların kavramsal temeli atılmıştır (Nabiyev 2005).

Makine öğrenmesindeki ilk araştırmalar daha çok *örüntü tanıma* alanı ile ilgiliydi. İlk öğrenen makine modeli *Perceptron* adı ile bilinen tek katmanlı ilk *Yapay Sinir Ağı* (YSA) modelidir. Basit nöron modellerine dayalı hesaplama modeli olup 1950’lerde Frank Rosenblatt tarafından önerilmiştir (Rosenblatt 1959).

1965’de ise ilk makine öğrenmesi kitabı yayınlanmıştır (Nilson 1965). 1968 yılında Vapnik ve Chervonenkis (VC) *İstatistiksel öğrenme* teorisini geliştirmiş, (örüntü tanıma problemi için) *VC* entropi ve *VC boyutu* temel kavramları, gösterge (*indikatör*) fonksiyonlar kümesi için tanıtılmıştır (Vapnik & Chervonenkis 1968). 1971’de bu kavramlar kullanılarak, fonksiyonel uzayda büyük sayılar kanunu bulunmuş, öğrenme işlemiyle ilişkisi tanımlanmıştır.

1972’lerde farklı disiplinlerle çalışan elektrik mühendisi Kohonen ve nöropsikolojist Anderson daha sonra geliştirilecek eğiticisiz öğrenmenin temeli olan çağrışımlı bellek (*associative memory*) konusunda birbirine benzer çalışmalar yayınlamışlardır (Kohonen 1972, Anderson 1972). 1975 – 1980 döneminde bilim insanları, dil, biyoloji ve psikoloji gibi diğer bilim alanlarından da faydalanabileceklerini görmüşler ve yapay zekâ da 1987 yılında bilim dalı haline gelmiştir. Günümüzde ekonomik olarak daha uygun yazılımlar ve araçlar sayesinde, daha geniş kullanım alanları ortaya çıkmıştır.

1982 yılında Kohonen kendi kendine öğrenme nitelik haritaları (*self organzinig feature maps* - SOM) konusunda çalışma yayınlamıştır. 1982 ve 1984 yıllarında Hopfield, YSA'ların genelleştirilebileceği ve çözümü zor problemlere çözüm üretebileceğini göstermiş, geleneksel *Gezgin Satıcı Problemini (Travelling Salesman Problem)* çözmüştür (Hopfield 1982). 1988 yılında, Broomhead ve Lowe *Radyal Tabanlı Fonksiyonlar Modelini (Radial Basis Functions RBF)* geliştirmişler ve özellikle filtreleme konusunda başarılı sonuçlar elde etmişlerdir (Broomhead & Lowe 1988). Daha sonra Specht**,** bu ağların daha gelişmiş şekli olan *Probabilistik Ağlar (PNN)* ve *Genel Regresyon Ağlarını (GRNN)* geliştirmiştir (Specht (1988) ve (1991)), (Öztemel 2003).

### 3.1.1 Öğrenme Nedir?

Öğrenme, günümüzde hala çok yönlü araştırmaların yapıldığı merak konusudur. Öğrenme, anlama ve zekâ birbirleri ile ilintili kavramlar olduğundan öğrenmenin tanımı yapılırken biraz zekâdan da bahsetmek gerekir. Latince “*intellectus*” kelimesinin karşılığı olan zekâyı, Lenat ve Feigenbaum, geniş arama alanı içinde gizlenmiş olan çözümü, hızlı bulabilen güç olarak tanımlarken (Kocabaş 1991), Feigenbaum buna ek olarak; “Zekâ arama yapmaktan ziyade belli bir amaç için gereken yararlı bilgileri kurabilme anlamına, yani “bilgi kurma” anlamına gelmektedir” der (Kocabaş 1991). Fatmi ve Young ise zekâyı fiziksel ve biyolojik açıdan değerlendirip, “Zekâ, daha önce düzensiz algılanan bir sistemi düzenli bir şekilde algılayabilme yeteneğidir.” şeklinde tanımlarlar (Kocabaş 1991). Zekâyı, biyologlar çevreye uyum kabiliyeti, eğitimciler öğrenme, psikologlar ilişkileri anlama, bilgisayarcılar bilgiyi işleme kabiliyeti şeklinde değerlendirmişlerdir.

Öğrenmenin tanımı bilim insanlarınca farklı şekillerde yapılmıştır. Michalski: ”Deneyimlerin gösterim biçimlerini oluşturma ve bunları değiştirmeye öğrenme denir.” derken, Simon öğrenmeyi, “zaman içinde yeni bilgilerin keşfedilmesi yoluyla davranışların iyileştirilmesi süreci” olarak tanımlar (Simon 1983). Zaman içinde iyileşmeden kasıt, bilgisayarın da insan gibi zamanla tecrübe kazanmasıdır. İnsan öğrenmesi ile makine öğrenmesi benzerdir. Makine öğrenmesi (*Machine Learning*) önceki örnekleri ve sonuçları inceler, bu işleri nasıl yeniden yapacağını öğrenip yeni durumlar hakkında genellemeler yapar.

Sistemin önceden bilmediği işlemleri, eğitimden sonra yapabiliyor olması, sistemin öğrendiğini düşündürür. Bu da öğrenmenin hiyerarşik yapıya sahip olduğunu gösterir. Makine öğrenimi geniş bir sahadır, sadece örneklerden öğrenme değil takviyeli öğrenme ve eğiticisiz öğrenme de bu sahaya girmektedir. Makine öğrenim sistemi tek bir gözlemleyici kullanmaz. *Eğitim kümesi* adı verilen bir sistem kullanır. Bu küme içinde örnek gözlem kodları bulunan ve makine tarafından okunabilen bazı formlar bulunur.

Makine öğrenmesi, *“bilgisayarın bir olay ile ilgili bilgileri ve tecrübeleri öğrenerek, gelecekte oluşacak benzer olaylar hakkında kararlar verebilmesi ve problemlere çözümler üretebilmesidir”* denilebilir.

Makine öğrenmesi, istatistik, olasılık kuramı, veri madenciliği, örüntü tanıma, yapay zekâ, gibi alanlarla ilişkilidir. Makine öğrenimi algoritmalarında önemli bir unsur da verinin ifade ya da gösterim biçimidir. Ayrık geometrik şekiller, özellikle verinin doğrusal olarak ayrilabilirliği makine öğrenimi yöntemlerinde önemli rol oynamaktadır.

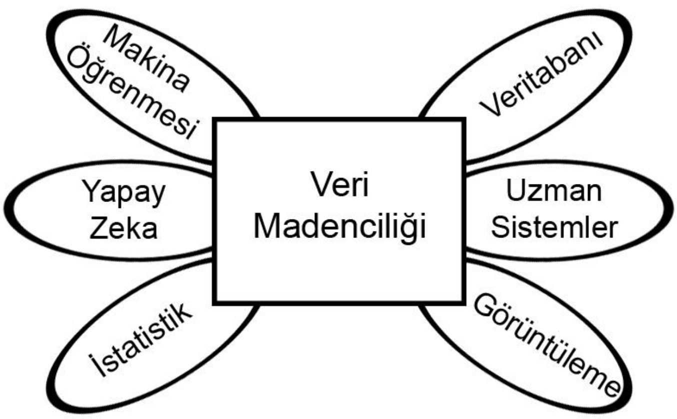
Makine öğrenmesinin temelinde; geçmiş bilgiler ve gözlemlerden yararlanarak gelecekte başımıza gelecekler hakkında bilgiye ulaşma, geleceğe ait durumları geçmişte karşılaştığımız durumları inceleyerek karar verme yatmaktadır. Karar vermede en gerekli unsur bilgidir. Bilgi, bir amaca yönelik işlenmiş veridir. Verinin bilgiye veya anlamlı hale dönüştürülmesi noktasında da *veri madenciliği* kavramı ile karşılaşılmaktadır. Veriyi bilgiye çevirmeye de veri *analizi* denir. Veri Madenciliği

(VM), verideki gizli kurallar veya örüntülerin bulunması ile ilgilenir. Bağımsız değişken sayısının çok fazla olduğu verilerde, VM veya makine öğrenmesi teknikleri sıklıkla kullanılmaktadır. Makine öğrenimi, istatistik ve VM arasında yakınlık vardır. Makine öğrenimi yöntemleri, VM algoritmalarında kullanılan yöntemlerin çekirdeğini oluşturur.

J. Han ve M. Kamber'e göre veri madenciliği; (*KDD-Knowledge Discovery in Databases* – Veri Tabanları Bilgi Keşfi – VTBK) veri tabanlarında, veri ambarlarında veya diğer veri depolarındaki büyük miktarlardaki veri içindeki ilginç bilgileri keşfetme sürecidir (Han & Kamber 2006).

Veri madenciliği, Şekil 3.1'de de gösterildiği gibi veritabanı teknolojisi, istatistik, makine öğrenmesi, örüntü tanıma, yapay sinir ağları, verilerin görselleştirmesi, uzaysal veri analizi gibi farklı disiplinlerde yer alan tekniklerin birleşimini içermektedir. Kuşkusuz bunlar içerisinde en büyük katkıyı istatistik, makine ögrenmesi ve veri tabanı teknolojisi sağlamaktadır.

**Şekil 3.1: Veri madenciliği ile diğer disiplinlerin gösterimi**




İstatistiğin amacı ana kütle hakkında anlamlı bilgiler elde etmek ve yorum yapmak, veri madenciliğinin amacı da anlamlı bilgiler elde etmek ve bunu eyleme dönüştürecek kararlar için kullanmaktır.

Veri madenciliği, veri tabanı teknolojisi, istatistik, yapay zekâ (*artificial intelligence*), makine ögrenmesi (*machine learning*), örüntü tanımlama (*pattem recognition*) ve veri görselleştirmesi (*data visualization*) gibi pek çok teknik alan arasında köprü görevi gören çok disiplinli bir alandır. Veri madenciliği astronomi, biyoloji, finans, pazarlama, sigorta, tıp gibi birçok dalda uygulanmaktadır.

Makine öğrenmesinin günlük yaşamdaki başlıca uygulama alanları; kredi riski hesaplaması, astronomik görüntülerin kataloglanması, üretim hatalarının tespiti, teknik direktörler için sporcu performansı analizi, örüntü tanıma, arama motorları, DNA dizilerinin sınıflandırılması, borsa çözümlemesi, konuşma ve el yazısı tanıma, bilgisayarlı görmede nesne tanıma, irisin sahibini bulma, yazılım mühendisliği, Biyoinformatik, test sonuçlarının tahmini, ürün geliştirme, tıbbi teşhiste, tedavi sürecinin belirlenmesi vs.'dir.

## 3.2 MAKİNE ÖĞRENMESİNİN ÇEŞİTLERİ

Makine öğrenmesinin yanıtlamaya çalıştığı sorulardan birisi de geçmiş deneyimleri öğrenmede nasıl kullanırız sorusudur. Gelecek bölümde öğrenme türlerinden bahsederken bu soruya da yanıt bulmuş olacağız.

Makine öğrenmesi, kullandığı algoritmaları öğrenim yöntemlerine göre (eğitim verisinden öğrenmek için) eğiticili öğrenme, eğiticisiz öğrenme, yarı eğiticili öğrenme, takviyeli öğrenme, ödüllü öğrenme, öğrenme ile öğrenme, uyum sağlama ile öğrenme olmak üzere alt gruplara ayrılır.

Farklı uygulamalarda faklı beklentiler olabilmektedir. Makine öğrenmesi yöntemlerini bu beklentilere göre de sınıflandırma, öbekleme, eğri uydurma (*regresyon*), ilişki belirleme vs. şeklinde guruplamak mümkündür (Alpaydın 2004).

Bu tezin konusu olan öbekleme problemini modellerken, sınıflandırma yöntemlerinden yola çıkıldığı için, bu bölümün alt başlıkları olarak sadece eğiticili ve eğiticisiz öğrenmeye yer verilecektir.

## 3.3 EĞİTİCİLİ ÖĞRENME (SUPERVISED LEARNINIG)

Eğiticili öğrenmede, sistemin öğrenmesine dışardan bir eğitici müdahale etmektedir. Eğitici, sistemin öğrenmesi için ilgili örnekleri girdi – çıktı kümesi ile vermektedir. Burada amaç; girdi ve çıktı değerleri arasındaki ilişkiyi öğrenmektir. Böylece yeni girdi değeri için çıktı değeri tahmin edilebilmektedir. Girdi – çıktı işlevselliğinin örnekleri eğitim verilerini göstermektedir. *Eğiticili öğrenmede* tüm eğitim verisi etiketlenmiştir. Girilen değer ve istenen değer arasındaki fark, hata değeri olarak önceden belirlenen değerden küçük oluncaya dek eğitime devam edilir. Sistem girdiler için istenen doğruluğa ulaştığı zaman, eğitim işi tamamlanmış olur ve süreç sona erer.

Sınıflandırma algoritmaları, regresyonun dâhil olduğu geleneksel istatistik teknikleri, yapay sinir ağları, destek vektör makineleri, eğiticili öğrenmeye örneklerdir. Bu tür öğrenmeye, el yazısı harf ve rakamların sınıflandırılması, borsa hisse değerlerinin öngörüsü, hava tahmini ve haber ajansındaki haberlerin sınıflandırması örnekleri verilebilir.

### 3.3.1 Sınıflandırma (Classification)

Bu bölümde önce sınıflandırmanın tanımı yapılacak, ardından sınıflandırma alanında kullanılan terimlerden, sonra da tezdeki modellemeye ışık tutan sınıflandırma yöntemlerinden biri olan Destek Vektör Makineleri anlatılacaktır.

Sınıflandırma, daha önceden sınıfları bilinen örnekler kullanılarak bir model oluşturulması ve bilinmeyen yeni bir verinin hangi sınıfa ait olduğunun tahmin edilmesidir. Tanımdan da anlaşılacağı gibi sınıflandırma tahminleyici bir modeldir; *havanın bir sonraki gün nasıl olacağı ya da bir kutuda ne kadar mavi top olduğunun tahmin edilmesi aslında bir sınıflandırma işlemidir* (Dunham 2003). Sınıflandırma, makine öğrenmesi literatüründe eğiticili öğrenmeye dâhil olur.

Sınıflandırmanın yaygın kullanım alanları; görüntü işleme, örüntü tanıma, hastalık tanıları, dolandırıcılık tespiti, banka kredisi onaylama işlemi, kredi kartı sahteciliği tespiti, hastalık teşhisi, ses tanıma, karakter tanıma, gazete haberlerini konularına göre ayırma, kullanıcı davranışları belirleme, kalite kontrol çalışmaları, pazarlama konuları, risk analizi vb.'dir.

Tezin ilerleyen bölümlerinde kullanacağımız *örüntü* (*pattern*), herhangi bir çizim, görüntü, ses, resim, parmak izi, bir kimsenin yaptığı işler ya da sınıflandırılması istenen başka bir işaret olabilir. Bir sınıflandırma probleminde *eğitim* kümesi (*training set*) ve *sınıf etiketleri* (*class label*) bilinmektedir. Kullanılan algoritmadan bağımsız, sınıflandırma modelinin oluşturulmasında kullanılan $n$ boyutlu $N$ adet örüntü, *veri kümesi*; veri kümesinin bir alt kümesi *eğitim kümesi* (*training set*); veri kümesinden eğitim kümesinin çıkarılması ile oluşan küme ise *test kümesi*dir. Eğitim kümesi $D = \{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \ldots, (\mathbf{x}_m, y_m)\}$ olmak üzere $(\mathbf{x}_1, \mathbf{x}_2, \ldots, \boldsymbol{x}_n)$ girdi (*input*) öyle ki $\mathbf{x}_i \in R^n$, $(y_1, y_2, \ldots, y_n)$ $y_i \in \{-1, +1\}$ çıktı (*output*) ya da etiket olarak adlandırılır (Bu tanımda sınıf sayısı 2 alınmıştır).

Sınıflandırma problemlerinde yaygın olarak kullanılan yöntemler: Karar Ağaçları (*Decision Trees*), Yapay Sinir Ağları (*Artificial Neural Network*), Destek Vektör Makineleri (*Support Vector Machines*), Naive Bayes, k-En Yakın Komşu (*k-Nearest Neighbor*) ve Genetik Algoritma yöntemleridir. Ancak gelecek alt bölümde sadece tezdeki modellemeye ışık tutan Destek Vektör Makineleri hakkında bilgi verilecektir (Duda et al. 2001, Cristianini & Taylor 2000, Han & Kamber 2006, Alpaydın 2004).

### 3.3.2 Destek Vektör Makineleri – DVM (Support Vector Machines – SVM)

Literatürde Destek Vektör Makineleri (DVM) konusundaki araştırmaların büyük bir kısmı sınıflandırma problemlerine yöneliktir. Bu tez çalışmasında da DVM yöntemi kullanılarak yeni bir öbekleme tekniği geliştirilmiştir ve detayları Bölüm 4'de anlatılmıştır. Burada kısaca DVM ile sınıflandırma, tekniği hakkında bilgi verilecektir.

DVM, 1960'ların sonunda matematikçi Vladimir Vapnik tarafından bulunan istatistiksel bir metottur. DVM, eğiticili veya yarı eğiticili olarak çalışabilen ve sınıflandırma ile regresyon uygulamalarında kullanılan bir makine öğrenmesi yöntemidir (Vapnik 1998). Günümüze kadar DVM, veri madenciliğinde (Burbidge & Buxton 2001), finans sektöründe (Chen & Shih 2006), çeşitli mühendislik problemlerinde (Chiang ve diğ.

2004, Samanta 2004), görüntü işleme uygulamalarında (Sun ve diğ. 2004), tıp (Übeyli 2007), (Çomak ve diğ. 2007a, 2007b) ve gıda sektörü (Du & Sun 2005) gibi çeşitli alanlarda başarıyla kullanılmıştır. El yazısı tanıma, ses ve yüz tanıma, metin karakterizasyonu, imaj tanıma, tıpta meme kanseri tanıma, protein bağı tanıma, biyoinformatik – gen ve protein sınıflandırması, kanser hücrelerinin tanınması ve uzaysal veri analizi gibi birçok alanda kullanılmaktadır.

DVM, güçlü bir algoritmik alt yapıya sahiptir. Küçük eğitim verisiyle öğrenebilir ve genel optimuma ulaşabilir. Hatta yüksek boyutlu az sayıda veri içeren eğitim kümelerinde de başarılıdır (Shen 2005, Shen ve diğ. 2004). Veriden öğrenme için üretici, eğitici ve öğrenme makinesi temel bileşenlerdir. Öğrenme makinesi girdi ve çıktı kümeleri arasındaki ilişkiyi, hipotez uzayı denilen karar verici fonksiyonların kümesi yardımıyla öğrenir. Veriden öğrenme problemi, eğiticinin yanıtlarını en iyi tahmin edebilen, hipotez uzayından bir fonksiyonun seçimidir. Eğiticinin cevapları da öğrenme makinesinin çıktı etiketleridir. İkili sınıflandırma (*Binary* SVM) problemi için $y \in \{-1,1\}$ olur.

DVM ile sınıflandırma probleminin en temeli, lineer olarak ayrılabilme durumudur. Temel mantığı, girdi uzayı üzerinde lineer olarak ayrılabilen veriler için en iyi hiperdüzlemin (*hyperplane*) belirlenmesidir. Doğrusal olarak ayrılabilir durumlarda bu veriler hiperdüzlem ile ayrılabilirler. Diğer bir durum ise doğrusal olarak ayrılamama durumudur. Bu durumda DVM, kernel fonksiyonları yardımıyla daha yüksek boyutlu bir uzaya taşınır ve öznitelik uzayında doğrusal ayrılabilr duruma gelir.

#### 3.3.2.1 Lineer destek vektör makineleri

En büyük aralık ile lineer ayrılabilir DVM (*Maksimal Marj SVM*), eğitim kümesinin hatasız (*hard margin*) ve hatalı (*soft margin*) ayrılma durumu olmak üzere iki alt başlıkta incelenecektir.

***i. Eğitim kümesinin hatasız ayrılma durumu (Hard Margin SVM)***

DVM, veriyi sınıflandırmak için eğitim kümesi $(\mathbf{x}_i\,, y_i)$ $(i = 1, 2, \ldots\ldots, N)$ kullanarak model oluşturmaktadır. $N$ adet veri vektörü ve n boyutlu uzayda $\mathbf{x}_i \in R^n$ $(i = 1, 2, \ldots, N)$ olmak üzere, veri vektörlerine karşılık gelen sınıf etiketi $y_i \in \{-1,1\}$ olsun. Sınıfları ayıracak olan hiperdüzlem, $\mathbf{w} \in R^n$ ağırlık vektörü (hiperdüzlemin normali), $b \in R$ sapma terimi (*bias*) olmak üzere denklem (3.1) ile tanımlanır. İki sınıftan oluşan veri kümesindeki örnekler, sınıflandırma probleminde denklem (3.1) kullanılarak gruplanmakta ve hiperdüzlem üzerindeki bütün noktalar denklem (3.1)'i sağlamaktadır. Lineer sınıflandırma için karar fonksiyonu, $f : R^n \rightarrow R$ olmak üzere,

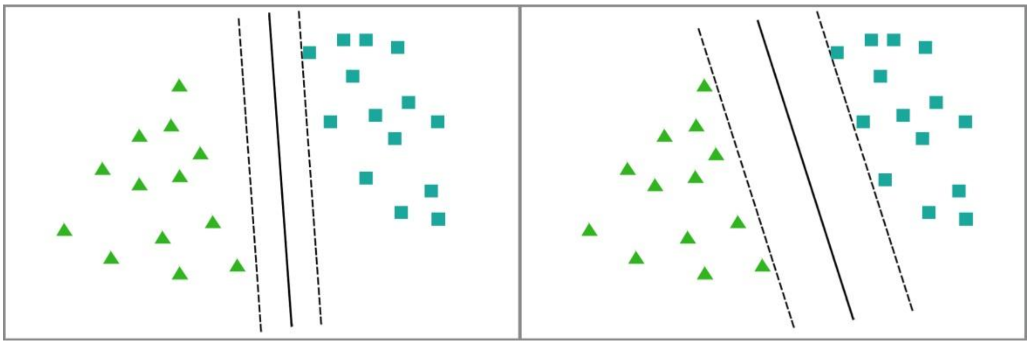
$$f(\mathbf{x}) = \langle \boldsymbol{w}, \boldsymbol{x} \rangle + b \qquad (\mathbf{3.1})$$

$$= \sum_{i=1}^{N} \mathbf{w}_i \mathbf{x}_i + b \text{ ile tanımlanır.}$$

Oluşturulmaya çalışılan hiperdüzlem, problemin eğitim verisini hatasız ayırabilen lineer fonksiyondur. Eğitim kümesinin elamanları $\mathbf{x}$ ve $y_i$'ler bilindiğine göre, en uygun hiperdüzlemi bulmak için $\mathbf{w}$ ve $b$ değerlerinin hesaplanması gerekir (Cherkassky 1998).

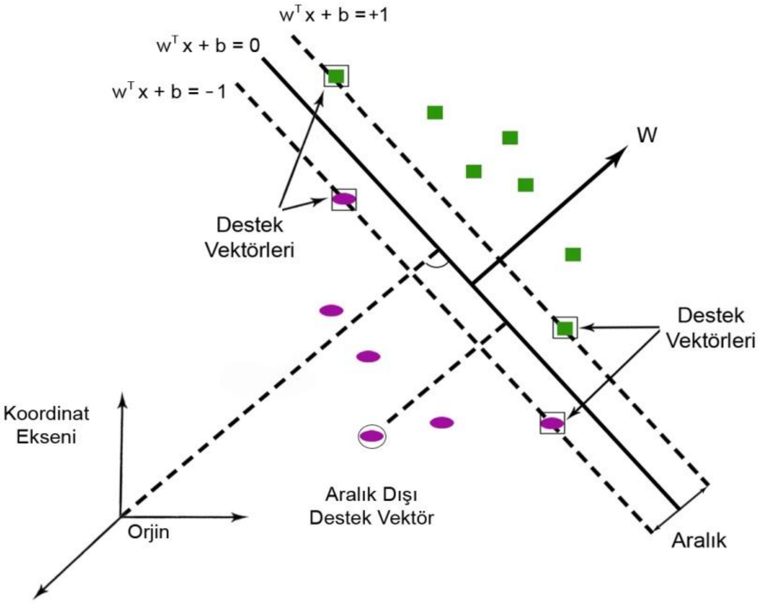
Hiperdüzlemden en yakın veri noktasına, dik olan maksimum uzaklığa ***aralık (margin)*** denir. Optimal (en uygun) hiperdüzlem, iki sınıf arasındaki aralığı maksimum yapan hiperdüzlemdir. Daha büyük aralık ile daha iyi bir genelleme yapılabileceğinden, Şekil 3.2'de gösterildiği gibi aralık maksimum uzunlukta alınarak, hiperdüzlem optimum (en uygun) hiperdüzlem olur (Cherkassky 1998). İki sınıf arasındaki aralığın (*margin*) büyük olması sınıflandırmanın iyi olduğunu göstermektedir.

**Şekil 3.2: Sınıflandırma örneği**



İkili sınıflandırma (*Binary* SVM) için {–1, +1} sınıf etiketleri kullanılabilir. Girdi vektörü $\mathbf{x} = (x_1, \dots, x_n)^T$ için, $f(\mathbf{x}) \geq 0$ veya $f(\mathbf{x}) < 0$ bulunur ve $sgn\left(f(\mathbf{x})\right)$ fonksiyonu ile verinin pozitif ya da negatif sınıfa ait olduğu belirlenir. Bu durumda, $y_i = +1$ için $\mathbf{w}^T\mathbf{x} + b \geq +1\ (*)$ yazılabilir ve $\mathbf{x}_i$ pozitif sınıfa, $y_i = -1$ için $\mathbf{w}^T\mathbf{x} + b \leq -1\ (**)$ yazılabilir ve $\mathbf{x}_i$ negatif sınıfa dahil edilir. Bu iki eşitsizliği birleştirirsek, $y_i(\mathbf{w}^T\mathbf{x}_i + b) \geq 1\ (i = 1, \dots, N)$ eşitsizliği elde edilir. DVM ile lineer sınıflandırma yöntemi Şekil 3.3' de gösterilmiştir.

**Şekil 3.3: DVM yöntemi ile lineer sınıflandırma**




*Kaynak*: Melgani & Bruzzone 2004

Şekil 3.3'de pozitif taraftaki hiperdüzleminin orjine dik uzaklığı, $|1-b|/\|w\|$, negatif taraftaki hiperdüzleminin orjine dik uzaklığı, $|-1-b|/\|w\|$ kullanılarak iki hiperdüzlemin optimal (en uygun) hiperdüzleme uzaklıkları $1/\|w\|$ bulunur. Buradan hareketle iki örnek küme arasındaki uzaklık da hiperdüzlemler biribirine paralel olduğu için $2/\|w\|$ olarak hesaplanır. Ağırlık vektörü, $\mathbf{w}$'nun öklit normu; $\|\mathbf{w}\| = \sqrt{\mathbf{w}^T\mathbf{w}}$ olmak üzere, hiperdüzlemler arasındaki en büyük aralığı veren **w**, başka bir deyişle, $1/\|w\|$ ifadesini maksimum yapan **w**, (hiperdüzlemler arasındaki aralığın (*margin*) maksimum olması için) $\|\mathbf{w}\|^2$ değerini minimum yapan **w** ile eşdeğerdir. O halde sınıflandırma problemini modellerken optimizasyon problemi $(\boldsymbol{P_P})$'nin amaç fonksiyonunda $\|\mathbf{w}\|^2$ terimi kullanılacaktır. Eğitim kümesinin hatasız ayrılma durumu, en büyük aralık ile hiperdüzlemi bulmak için aşağıdaki optimizasyon problemi çözülebilir:

*Özgün (Primal) Problem*

$$\min_{w,b} \quad \frac{1}{2}\|w\|^2 \qquad (\boldsymbol{P_P})$$

$$\text{kısıtlar } y_i(\, w^T x_i \, + \, b) \; \geq 1 \quad (i = 1, \dots, N). \qquad (\mathbf{3.2})$$

Optimizasyon problemi $(P_P)$ bir konveks karesel programlama problemidir. Kısıtlar, eğitim örneklerinin hiperdüzlemin doğru tarafına düşmesini formüle etmektedir. Problem $(P_P)$'nin, Bölüm 2.2.2'de anlattığımız Lagrange tekniği ile çözülmesi mümkündür. O halde, problem $(P_P)$ in Lagrange fonksiyonu,

$$L_P(\mathbf{w}, b, \boldsymbol{\alpha}) = \frac{1}{2}\|\boldsymbol{w}\|^2 - \sum_{i=1}^{N} \alpha_i[y_i(\, \mathbf{w}^T\mathbf{x}_i \, + \, b) - 1] \qquad (\mathbf{3.3})$$

şeklinde yazılabilir. Denklem (3.3)'de $\alpha_i \geq 0$ pozitif Lagrange çarpanlarıdır. KKT şartları gereğince Lagrange fonksiyonunun **w** ve $b$ değişkenlerine göre türevleri alınıp, sıfıra eşitlenirse,

$$\frac{\partial L_P(\mathbf{w}, b, \boldsymbol{\alpha})}{\partial \mathbf{w}} = 0 \quad \Rightarrow w = \sum_{i=1}^{N} \alpha_i y_i \mathbf{x}_i,$$

$$\frac{\partial L_P(\mathbf{w}, b, \boldsymbol{\alpha})}{\partial b} = 0 \quad \Rightarrow \quad \sum_{i=1}^{N} \alpha_i y_i = 0$$

eşitlikleri bulunur. Bulunan $\mathbf{w}$ ve $b$ değişkenlerinin dual formları denklem (3.3)'de yerine yazıldığında aşağıdaki Lagrange fonksiyonu elde edilir:

$$L_D = \sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{N} \alpha_i \alpha_j y_i y_j \mathbf{x}_i^T \mathbf{x}_j . \qquad \mathbf{(3.4)}$$

Elde edilen Lagrange fonksiyonunun maksimizasyonu bize aşağıdaki dual problem $(P_D)$'i verir:

$$\max_{\alpha} \quad \sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_i \alpha_j y_i y_j \langle \mathbf{x}_i , \mathbf{x}_j \rangle \qquad (\boldsymbol{P_D})$$

kısıtlar $\quad \alpha_i \ \geq \ 0 \quad (i = 1, \dots, N),$

$$\sum_{i=1}^{N} \alpha_i y_i = 0.$$

Lagrange çarpanları $\alpha_i \geq 0$ pozitiftir. Eğitilen sınıflandırıcı ile yeni örneklerin sınıflarının bulunabilmesi için, hiperdüzlemin hangi tarafında olduğuna bakılması gerekir. Bunun için örneğin hiperdüzlem fonksiyonundaki değeri hesaplanarak işaretine bakılır. Sonuç olarak, $f(x) = sgn\ (\langle \mathbf{w}, \mathbf{x} \rangle + b)$ eşitliği ile verilen karar fonksiyonunda dual değişkenler kullanarak,

$$f(x) = sgn \left( \sum_{i=1}^{N} \alpha_i y_i \langle \mathbf{x}_i , x \rangle + b \right) \qquad \mathbf{(3.5)}$$

yazılabilir ve fonksiyonunun işareti bize $\mathbf{x}$ noktasının hangi sınıfa ait olduğunu verir.

KKT kullanılarak:

$$\alpha_i[y_i(\mathbf{w}^T\mathbf{x}_i + b) - 1] = 0 \qquad (i = 1, \ldots, N),$$

$$\alpha_i > 0 \Rightarrow y_i(\mathbf{w}^T\mathbf{x}_i + b) - 1 = 0,$$

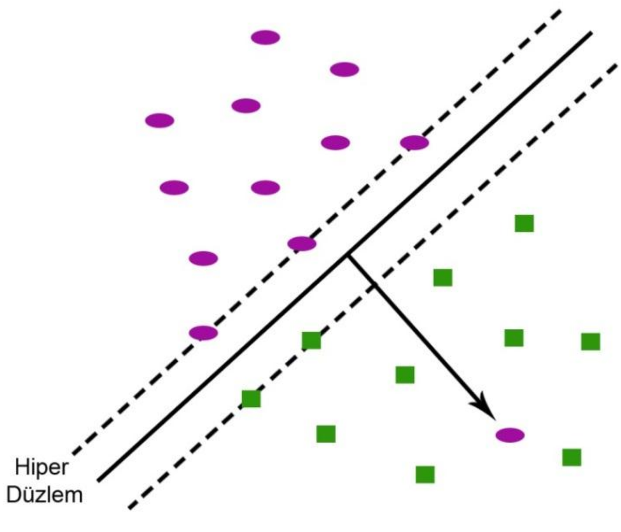
$$\alpha_i = 0 \Rightarrow y_i(\mathbf{w}^T\mathbf{x}_i + b) - 1 > 0$$

ifadeleri sağlanır. Problem $(P_D)$'de her eğitim örneği için bir Lagrange çarpanı vardır, çözümden elde edilen Lagrange çarpanlarının çoğu sıfır olacaktır ve her bir örnek için $\alpha_i = 0$ ya da $y_i(\mathbf{w}^T\mathbf{x}_i + b) = 1$ olacaktır. $\alpha_i = 0$ olan örnekler denklem (3.5)'de toplamda etkisiz olacak ve buna ek olarak yeni örneklerin sınıf etiketlerinin tahmininde de önemi olmayacaktır. Geriye kalan $\alpha_i \geq 0$ değerli $\mathbf{x}_i$ örnekleri *destek vektörleri* olup, $y_i(w^T\mathbf{x}_i + b) = 1$ eşitşiğini sağlar. Destek vektörleri pozitif veya negatif taraftaki hiperdüzlemlerin üzerinde yer alırken, $\alpha_i = 0$ olan örnekler ise pozitif veya negatif taraftaki hiperdüzlemlerin arka taraflarında yer alırlar.

### *ii. Eğitim kümesinin hata ile ayrılma durumu (Soft Margin SVM)*

Bazı durumlarda, lineer bir düzlem ile veriler birbirinden ayrılmayabilir. O zaman da verilerin minimum hata ile ayrılması tercih edilebilir. Veriler (eğitim verileri) denklem (3.2) kısıtını sağlamadıklarında maximum aralığın içine düşmekte veya sınıf etiketinin hatalı olduğu bölgeye atanmaktadır. Hatalı sınıflandırma, veri noktasının yanlış bölgeye atandığını göstermektedir. Aşağıda Şekil 3.4'de hatalı sınıflandırılmaya yer verilmiştir.

**Şekil 3.4: Hatalı sınıflandırma örneği**



Literatürde esnek aralık (*soft margin*) olarak geçen bu teknikte amaç, hatalı sınıflandırmaya (*allow misclassification*) izin vermektir. O halde, eğitim kümesinin hatasız ayrılma durumunda anlatılan DVM problemini bir miktar hataya izin verebilecek şekilde yeniden düzenlemek gerekir (bknz Şekil 3.5). Bunun için, öncelikle $(*)$ ve $(**)$ daki kısıtlamalar gevşetilerek hiperdüzlemin koşulları:

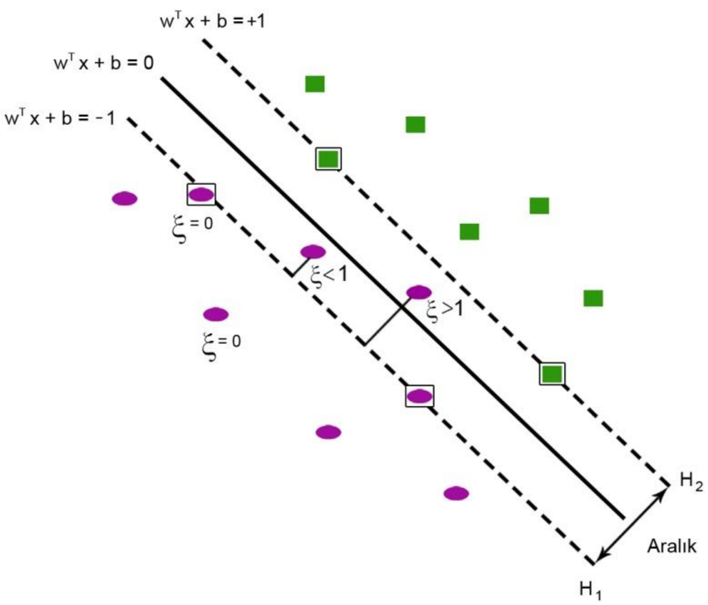
$$y_i = +1 \text{ için, } \mathbf{w}^T\mathbf{x} + b \geq +1 - \xi_i,$$

$$y_i = -1 \text{ için, } \mathbf{w}^T\mathbf{x} + b \leq -1 + \xi_i,$$

$$\xi_i \geq 0 \quad \forall i$$

biçiminde yeniden tanımlanabilir (Cristianini & Taylor 2000).

**Şekil 3.5: Gevşek değişkenlerin gösterimi**




*Kaynak*: Gönen 2007

Şekil 3.5'de görüldüğü üzere, $\xi_i$ (*slack variable*) negatif olmayan gevşek değişkenlerdir. Bu değişkenlerin $\xi_i = 0$ olması durumunda $\mathbf{x}_i$ eğitim örneği doğru sınıflandırılmış, $\xi_i \geq 1$ ise hatalı sınıflandırılmıştır. Buna göre, $\xi_i = |y_i - (\mathbf{w}^T\mathbf{x}_i + b)|$ için, $\mathbf{x}_i$ karar sınırının üzerinde ise $\mathbf{w}^T\mathbf{x}_i + b = 0$ dolayısıyla $\xi_i = 1$ olacak; $\xi_i > 1$ olan $\mathbf{x}_i$'ler için hatalı sınıflama olacaktır. $0 < \xi_i < 1$ durumunda ise $\mathbf{x}_i$'ler hiperdüzlemlerin arasında (en büyük aralık bölgesinde) yer alacaktır.

Aralığın en büyük, sınıflandırma hatalarının en küçük olabilmesi için *C* ile gösterilen, pozitif değerler alabilen $(0 < C < \infty)$ düzenleme (regularization) parametresi kullanılabilir. Bu parametre hataları kontrol altına almak için kullanılır. Örneğin, iki sınıf arasındaki aralığa esneklik kazandırarak daha iyi sınıflandırma yapılmasını sağlar. Ayrıca kategorileri ayırma işleminde esnekliğe de izin verir. *C* parametresi kullanıcı tarafından girilir. *C* değeri arttığında daha fazla örnek hatalı sınıflandırılacak ve sınıflandırıcının karmaşıklığı düşecek, *C* değeri düşük tutulduğunda ise daha az örnek hatalı sınıflandırılacak ve sınıflandırıcının karmaşıklığı artacaktır. Dolayısıyla *C* parametresi sınıflandırma hatası ile model karmaşıklığı arasındaki çelişkiyi düzenleme görevini üstlenir.

Lineer olarak belirli oranda hata ile ayrılabilen veri kümelerinde, en uygun hiperdüzlemi bulmak için, düzenleme parametresi ve gevşek değişken kullanılarak esnek aralıklı (*soft margin*) temel optimizasyon problemi $(P_{SMP})$ aşağıdaki gibi oluşturulabilir:

*Özgün (Primal) Problem*

$$\min_{w,b,\xi} \quad \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{N}\xi_i \tag{$\boldsymbol{P_{SMP}}$}$$

$$\text{kısıtlar} \quad y_i(\,\mathbf{w}^T\mathbf{x}_i + b) \geq 1 - \xi_i, \tag{3.6}$$

$$\xi_i \geq 0 \qquad (i = 1, \ldots, N). \tag{3.7}$$

Problem $(P_{SMP})$' in ilk terimi lineer ayrılabilen (*hard margin* DVM) temel DVM problemi ile aynı iken, ikinci terim hatalı sınıflandırma noktasını kontrol etmektedir. Ayrıca *C* parametresi Lagrange çarpanlarının alabilecekleri maksimum değeri de göstermektedir. Dolayısıyla $\alpha_i \in [0, C]$ aralığında bulunmaktadır. Eşitsizlik (3.6) ve (3.7)'nin problem$(P_{SMP})$' inin kısıtları olduğunu dikkate alarak Lagrange denklemi,

$$L(\boldsymbol{w}, b, \xi, \boldsymbol{\alpha}, \boldsymbol{\mu}) =$$

$$= \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{N}\xi_i - \sum_{i=1}^{N}\alpha_i[y_i(\mathbf{w}^T\mathbf{x}_i + b) - 1 + \xi_i] - \sum_{i=1}^{N}\mu_i\xi_i \qquad (\mathbf{3.8})$$

ile yazılabilir. Pozitif $\alpha_i$, $\mu_i$ katsayıları, $\xi_i$' nin pozitif olması için kullanılan Lagrange çarpanlarıdır. Çözüm için KKT koşullarını yazalım:

$$\frac{\partial L(\boldsymbol{w}, b, \xi, \boldsymbol{\alpha}, \boldsymbol{\mu})}{\partial \mathbf{w}} = 0 \quad \Rightarrow \mathbf{w} = \sum_{i=1}^{N}\alpha_i y_i \mathbf{x}_i, \qquad (\mathbf{3.9})$$

$$\frac{\partial L(\boldsymbol{w}, b, \xi, \boldsymbol{\alpha}, \boldsymbol{\mu})}{\partial b} = 0 \quad \Rightarrow \sum_{i=1}^{N}\alpha_i y_i = 0, \qquad (\mathbf{3.10})$$

$$\frac{\partial L(\boldsymbol{w}, b, \xi, \boldsymbol{\alpha}, \boldsymbol{\mu})}{\partial \xi_i} = 0 \quad \Rightarrow C = \alpha_i + \mu_i. \qquad (\mathbf{3.11})$$

Yukarıdaki denklem (3.9) ve (3.11), Lagrange denklemi (3.8)'de yerine yazıldığında ve denklem (3.10)'u da kısıt olarak aldığımızda aşağıdaki dual problemi elde etmiş oluruz:

*Dual Problem*

$$\max_{\boldsymbol{\alpha}} \quad \sum_{i=1}^{N}\alpha_i - \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}\alpha_i\alpha_j y_i y_j\langle\mathbf{x}_i, \mathbf{x}_j\rangle \qquad (\boldsymbol{P_{SMD}})$$

öyle ki $\quad 0 \leq \alpha_i \leq C \quad (i = 1, \dots, N),$

$$\sum_{i=1}^{N}\alpha_i y_i = 0.$$

O halde dual değişkenler kullanılarak karar fonksiyonu

$$f(\mathbf{x}) = sgn\left(\sum_{i=1}^{N}\alpha_i y_i \langle\mathbf{x}_i, \mathbf{x}\rangle + b\right)$$

bulunur.

Lineer DVM için bulunan konveks karesel programlama $(P_D)$ problemi ile karar fonksiyonunun değişmediği görülmektedir. Tek fark, $\alpha_i$ Lagrange çarpanları için $C$ üst sınırının getirilmesidir. Eğer $C = \infty$ alınırsa, ilk durum elde edilir. Aşağıda $\alpha_i$ katsayılarının sınırda aldığı değerlere göre eşitsizlik kısıtının durumları incelenmiştir:

$$\alpha_i[y_i(\mathbf{w}^T\mathbf{x}_i + b) - 1 + \xi_i] = 0 \quad (i = 1, \dots, N),$$

$$\beta_i\xi_i = 0,$$

$$\alpha_i = C \Rightarrow y_i(\mathbf{w}^T\mathbf{x}_i + b) - 1 + \xi_i = 0 \quad \xi_i > 0,$$

$$0 < \alpha_i < C \Rightarrow y_i(\mathbf{w}^T\mathbf{x}_i + b) - 1 + \xi_i = 0 \qquad \xi_i = 0,$$

$$\alpha_i = 0 \Rightarrow y_i(\mathbf{w}^T\mathbf{x}_i + b) - 1 + \xi_i > 0.$$

Sonuç olarak, daha öncede anlatıldığı gibi, $\alpha_i = C$ olan $x_i$'ler $(\xi_i > 0)$ gevşek değişken ile sınırlandırılmış destek vektörleri olarak tanımlanırlar ve en geniş aralık alanının içerisinde uzanırlar. $0 < \alpha_i < C$ olan $(\xi_i = 0)$, $\mathbf{x}_i$'ler aralık üzerinde olan sınırlandırılmamış destek vektörleridir.

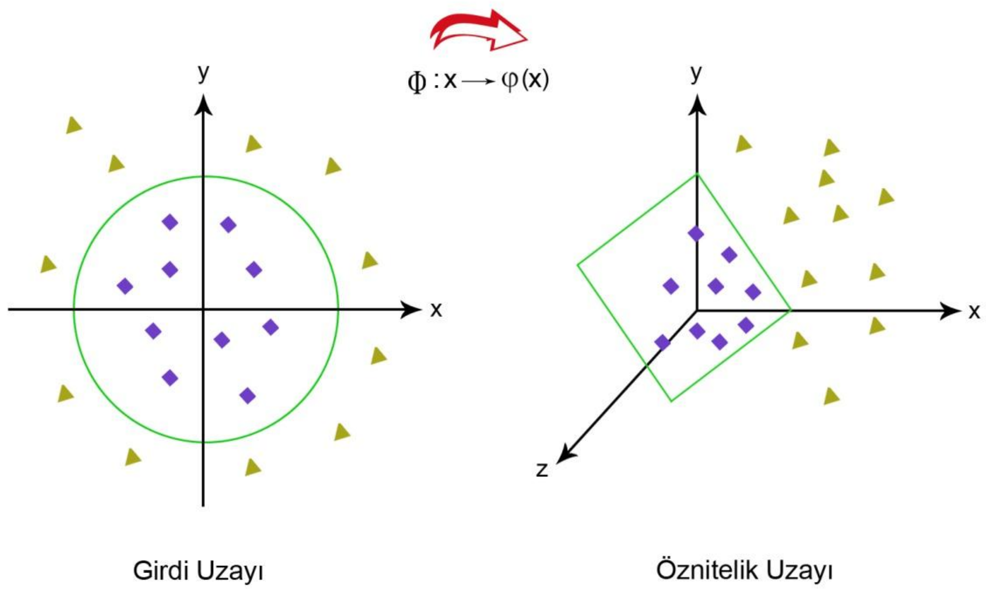
### 3.3.2.2 Lineer olmayan destek vektör makineleri

Lineer olarak ayrılamayan durumlarda örnekler (Şekil 3.6), kernel fonksiyonları ile lineer olarak ayrılabildikleri daha yüksek boyutlu başka bir uzaya lineer olmayan $\mathbf{\Phi}(x)$ fonksiyonu ile taşınır ve sınıflandırma o uzayda yapılır. Yüksek boyutta oluşturulan öznitelik uzayını $F$ ile gösterelim. Lineer olmayan $\mathbf{\Phi} : X \to F$ ile tanımlanır öyle ki, $F = \{\Phi(\mathbf{x}) | \mathbf{x} \in X\}$ olsun. Artık $\mathbf{x}_i$ noktaları yerine öznitelik uzayından alınacak olan $\mathbf{\Phi}(\mathbf{x})$ noktaları vardır. Dolayısıyla karar fonksiyonu öznitelik uzayında yazılırsa,

$$f(\mathbf{x}) = sgn\left(\sum_{i=1}^{N} \boldsymbol{w}_i\Phi_i(\mathbf{x}) + b\right) \qquad \mathbf{(3.12)}$$

şeklinde bir fonksiyon oluşacaktır.

**Şekil 3.6: Girdi uzayından öznitelik uzayına dönüşüm**




*Kaynak:* Zien 2008, Multiple Kernel Learning

Denklem (3.1) ile tanımlanan karar fonksiyonunda $\mathbf{x}$ yerine, $\mathbf{\Phi}(\mathbf{x})$ yazılarak, öznitelik uzayında tanımlanmış karar fonksiyonu elde edilir:

$$f(\mathbf{x}) = sgn\left(\sum_{i=1}^{N} \alpha_i y_i \langle \mathbf{\Phi}(\mathbf{x}_i), \mathbf{\Phi}(\mathbf{x})\rangle + \mathrm{b}\right). \qquad \mathbf{(3.13)}$$

Denklem (3.13)'de iç çarpım yerine $K\ (\mathbf{x}_i, \mathbf{x})$ yazılırsa

$$f(\mathbf{x}) = \ sgn\left(\sum_{i=1}^{N} \alpha_i y_i \, K\ (\mathbf{x}_i, \mathbf{x}) + \mathrm{b}\right)$$

elde edilir, öyle ki $K\left(\mathbf{x}_i, \mathbf{x}_j\right) = \langle \Phi(\mathbf{x}_i)\,.\,\Phi\left(\mathbf{x}_j\right)\rangle$ fonksiyonuna ***Kernel fonksiyonu*** adı verilir. Dikkatle incelenirse, kernel fonksiyonu kullanıldığında $\Phi(.)$ fonksiyonunu hesaplamaya gerek kalmamaktadır. Yapılması gereken iç çarpımları giriş uzayı yerine kernel fonksiyonları ile kernel uzayında hesaplamaktır. Bir fonksiyonun kernel fonksiyonu olabilmesi için Mercer koşullarını (Pöyhönen 2004, Mercer 2003) sağlamalı ve simetrik kesin pozitif olmalıdır.

Başarılı bir sınıflandırma için uygun kernel fonksiyonunun seçilmesi gerekmektedir. Literatürde çeşitli kernel fonksiyonları mevcuttur, aşağıda en yaygın Kernel fonksiyonlarına yer verilmiştir:

1. Doğrusal kernel fonksiyonu: $K\left(\mathbf{x}_i, \mathbf{x}_j\right) = \left(\mathbf{x}_i^T \mathbf{x}_j\right),$
2. Polinom kernel fonksiyonu: $K\left(\mathbf{x}_i, \boldsymbol{x}_j\right) = \left(\gamma \mathbf{x}_i^T \mathbf{x}_j\right)^d,$
3. Radyal tabanlı (RBF)kernel fonksiyonu (Gauss Kerneli):

$$K\left(\mathbf{x}_i , \mathbf{x}_j\right) = \exp\left( -\frac{\| \mathbf{x}_i - \mathbf{x}_j \|^2}{2\sigma^2}\right).$$

## 3.4 EĞİTİCİSİZ ÖĞRENME (UNSUPERVISED LEARNING)

Eğiticisiz öğrenmede, sistemin olayı öğrenmesine yardımcı olan bir eğitimci yoktur. Sisteme sadece girdi değerleri verilip, örnekler arasındaki ilişkiyi öğrenmesi beklenir. Doğru çıktılar hakkında hiçbir bilgi verilmez. Amaç, girdi değerlerinin öznitelikleri kullanılarak öğrenme yapmaktır.

Eğiticisiz öğrenmede nesnelerin eğitim örnekleri verilmemektedir. Dolayısıyla herbir veri, niteliklerin değerlerini içerirken, *etiket içermemektedir*. Eğiticili öğrenmeden de farkı budur. Eğiticisiz öğrenme, genellikle *öbekleme* için kullanılan bir öğrenme çeşididir. Yarışmacı öğrenme (*competitive learning*), Kohonen'in kendini düzenleyen (*self-organizing*) haritaları (Kohonen 1984, Kohonen 2001), Hebbian öğrenme (Hebb 1949), Grossberg öğrenme (Grossberg 1986) gibi öğrenme kuralları eğticisiz öğrenme örnekleridir (Elmas 2003).

### 3.4.1 Öbekleme (Clustering)

#### 3.4.1.1 Literatür özeti

Öbekleme konusundaki çalışmalar 1963 yılında Sokal ve Sneath nin yazdığı "Principles of Numerical Taxonomy" kitabı yayınlandıktan sonra hızlanmıştır (Sokal & Sneath 1963). Öbekleme üzerine yazılan ilk kitaplardan biri, fakat tek bir yaklaşımın kullanıldığı “Cluster Analysis” (Tryon & Bailey 1970) ardından, daha çok öbekleme işleminin matematik kısmının incelendiği “Mathematical Taxonomy” (Jardine & Sibson 1971), öbekeleme üzerine yazılmış olan en kapsamlı kitap “Cluster Analysis for Applications” (Anderberg 1973), değişik projelerin toplandığı, “Clustering Algorithms” (Hartigan 1975), geniş kapsamlı “Algorithms for Clustering Data” (Jain & Dubes 1988) ve 1990’dan itibaren konu ile ilgili olarak çok farklı uygulamalar ve kitaplar ortaya konulmuştur.

Farklı uygulamalarda kullanılabilen çok çeşitli öbekleme algoritmaları mevcuttur. Literatürde yeni öbekleme algoritmaları da geliştirilmektedir. Genel olarak bu algoritmalar iki başlıkta toplanmaktadır: Veri kümesindeki öbekler *bölümleme (partitioning)* ya da *aşama sıralı (hierarchical*) *öbekleme* yöntemleri ile belirlenebilir. Var olan verileri öbeklere ayıracak tek bir algoritma bulunmamaktadır. Dolayısıyla çeşitli algoritmalar denenmelidir.

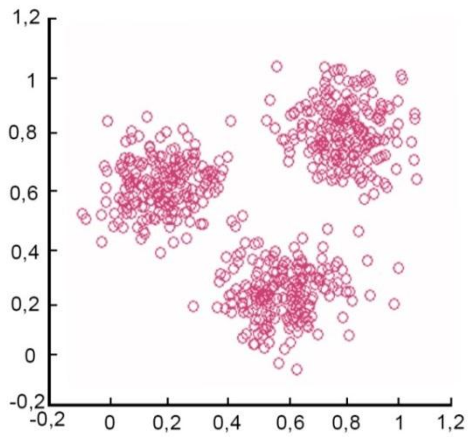
Uygulama alanları; müşteri değerlendirme ve çapraz satış analizleri (pazarlama), risk analizleri, usulsüzlüklerin tespiti, ana giderlerin azaltılması, poliçe fiyatlarının belirlenmesi (sigortacılık), satış noktası veri analizleri, alış-veriş sepeti analizleri (perakendecilik), hisse senedi fiyat tahmini, genel piyasa analizleri, en iyi alım-satım stratejilerinin belirlenmesi (borsa), hatların yoğunluk tahminleri (haberleşme), test sonuçlarının tahmini, ürün geliştirme, ilaçlarda kullanılan maddelerin sınıflandırılması (ilaç sanayi), tıbbi teşhis, uygun tedavi sürecinin belirlenmesi (sağlık), kalite kontrol, lojistik, üretim süreçlerinin en iyileştirilmesi (endüstri) gözlemsel veriler üzerinde modeller kurularak bilimsel ve teknik problemlerin çözümlenmesi, çeşitli tahminler ve sınıflandırma problemlerinin ayrıştırılarak çözümlenmesidir (Hartigan 1975).

#### 3.4.1.2 Giriş

Öbek, benzer elemanların bir grubudur. *Öbekleme ise en basit tanımıyla benzer özellik gösteren veri elemanlarının kendi aralarında gruplara ayrılmasıdır.* Literatürde öbekleme analizini açıklayan birçok tanım bulunmaktadır (Berkhin 2002, Bilgin 2003, Boutsinas & Gnardellis 2002, Han & Kamber 2006, Jain & Dubes 1988, Jain ve diğ. 1999, Karypis ve diğ. 1999, Mercer 2003, Witten & Frank 1999). Öbekleme de amaç, Şekil 3.9'da görüldüğü gibi öbekler arasındaki uzaklıkları en büyük, her bir öbek içindeki gözlemler arasındaki uzaklıkları en küçük yapmaktır. Öbekleri belirlemek için genellikle öbekler içindeki verilerin nasıl benzer olduklarını ya da öbekler arasındaki verilerin nasıl farklı olduklarını modelleyen fonksiyon önemli bir kriterdir ve öbekleme sonuçlarının değerlendirilmesinde önem kazanacaktır. Öbekleme algoritmaları veri tabanındaki kayıtları alt kümelere ayırır. Başlangıçta veritabanındaki kayıtların hangi öbeklere ayrılacağı veya öbeklemenin hangi değişken özelliklerine göre yapılacağı, oluşacak öbek sayısı, öbekler hakkında ön bilgi bilinmemekte olup herbiri problem için birer parametredir.

Şekil 3.7'de görüldüğü gibi her bir öbekteki elemanlar kendi grubunu diğer gruplardan ayıran benzer özelliklere sahiptir ve buradaki benzerlik fonksiyonu mesafedir.

**Şekil 3.7: Öbekleme için örnek veri kümesi**




Bu bölümde anlatılanlardan, öbekleme bize "benzerlik, uzaklık, örüntü vs. kavramlarını çağrıştırmaktadır. Alt bölümde ise öbekleme içinde kullanacağımız bazı genel tanımlardan bahsedilecektir.

### 3.4.2 Tanımlar ve Kullanılan Semboller

**Tanım 3.4.1 (Örüntü Matrisi - Pattern Matrix):** Örüntü (*pattern*) öbekleme algoritmasınca kullanılan veri öğeleridir ve yapılan ölçümlerin sonuçlarını içermektedir.

Örüntü vektörünün her bir sayısal elemanı da ($\mathbf{x}_i$), verilerin uzaklık bileşenleri, öznitelik (*attribute*) olarak tanımlanmaktadır.

Girdi vektörü,

$$\mathbf{x}_i = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_d \end{bmatrix} \quad (i = 1, 2, \dots, n),$$

$d$ öznitelik ve $n$ örüntü sayısını göstermek üzere, $n$ x $d$ boyutlu örüntü matrisi X aşağıdaki şekilde gösterilmektedir:

$$\mathrm{X} = [\mathbf{x}_1 \quad \mathbf{x}_2 \quad \dots\dots \quad \mathbf{x}_n\,] \Rightarrow \mathrm{X} = \begin{bmatrix} x_{11} & x_{12} & \cdots & x_{1d} \\ x_{21} & x_{22} & \cdots & x_{2d} \\ \vdots & \vdots & \ddots & \vdots \\ x_{n1} & x_{n2} & \cdots & x_{nd} \end{bmatrix}.$$

Bu matrisin her satırı bir veri vektörünü (örüntüyü) ve vektörlerin bileşenleri de öznitelikleri ifade etmektedir (Jain & Dubes 1988). Örüntü matrisinde sütunları oluşturan bu özellikler için bir hastanedeki hastaların yaş, boy, ağırlık, cinsiyet gibi kişisel özellikleri ve tahlil sonuçları örnek olarak verilebilir.

**Tanım 3.4.2 (Yakınlık Matrisi (Proximity Matrix)):** Yakınlık matrisi $D_{ij}$, denklem (3.14) ile gösterildiği gibi, her satır ve sütununda bir örüntünün bulunduğu, nesnelerin karşılıklı benzerlik (yakınlık) bilgilerinin tutulduğu, $n$ x $n$ boyutlu ve köşegen üzerinde bulunan tüm değerlerin sıfır olduğu bir matristir. Nesneler arasındaki yakınlık değişme özelliğine sahip olduğundan yakınlık matrisleri simetriktir (Jain ve diğ. 1999).

Yakınlık matrisi,

$$D_{ij} = \begin{bmatrix} d_{11} & d_{12} & \cdots & d_{1m} \\ d_{21} & d_{22} & \cdots & d_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ d_{n1} & d_{n2} & \cdots & d_{nm} \end{bmatrix} \qquad (i = 1,2,\dots \mathrm{n}), \quad (\mathrm{j} = 1 \dots\dots m). \qquad \mathbf{(3.14)}$$

ile ifade edilir. Burada $d_{ij}$ ; *i* ve *j* nesneleri arasında ölçülen yakınlık veya uzaklıktır.

**Tanım 3.4.3 (Yakınlık İfadeleri):** Verilen X veri matrisinde $i$. ve *j. ö*rüntüler (i. ve j. satırlar) arasındaki yakınlık değeri $d(i,j)$ ile gösterilmektedir ve $d(i,j)$, tüm $(i,j)$ değerleri için,

i. $d(i,i) = 0$,

ii. $d(i,j) = d(j,i)$,

iii. $d(i,j) \geq 0$,

iv. $d(i,j) = 0 \Leftrightarrow i = j$,

v. $d(i,k) \leq d(i,j) + d(j,k) \quad (i,j = 1,2, \dots n), (k = 1,2, \dots, p)$ (üçgen eşitsizliği),

şartlarını sağlıyorsa $d(i,j)$'ye *uzaklık fonksiyonu* denir.

Örüntüleri öbeklendirmede benzerlik (*similarity*) ve benzememezlik (*dissimilarity*) ölçüsü olan sayısal değer, *uzaklık ölçüsü (distance measure)* olarak adlandırılmaktadır. Uzaklıkların hesaplanması ve yakınlık matrisinin oluşturulması için en çok kullanılan uzaklık ölçüleri aşağıdaki gibi verilmiştir (Tatlıdil 1996):

1. *Öklit Uzaklığı (Euclidean Distance)*

$$d(i,j) = \sqrt{\sum_{k=1}^{p} \left(x_{ik} - x_{jk}\right)^2}\,, \qquad \mathbf{(3.15)}$$

( ‖•‖ ile de gösterilmektedir.)

2. *City Blok Uzaklığı (Manhattan Uzaklığı)*

$$d(i,j) = \sum_{k=1}^{p} \left|x_{ik} - x_{jk}\right|\,, \qquad \mathbf{(3.16)}$$

3. *Minkowski Uzaklığı*

$$d(i,j) = \left\{ \sum_{k=1}^{p} \left| x_{ik} - x_{jk} \right|^{r} \right\}^{1/r}, \qquad (\mathbf{3.17})$$

4. *Supremium Uzaklığı*

$$d(i,j) = \max_{1 \le j \le d} \left| x_{ik} - x_{jk} \right|.$$

Minkowski uzaklığı denklem (3.17)'de $r = 2$ seçilirse Öklit uzaklığı; $r = 1$ seçilirse Manhattan uzaklığı elde edilir; $r = \infty$ seçilirse Supremium uzaklığı (*Sup distance*) elde edilir.

Bunlar arasında en yaygın kullanılan uzaklık fonksiyonu, Öklit uzaklık ölçümüdür. Manhattan ve Öklit uzaklık fonksiyonları çoğunlukla benzerliklerin bulunmasında kullanılır. Uzaklık fonksiyonu büyük bir değer ise benzerlik az, küçük bir değer ise benzerliğin fazla olduğu anlaşılır. Belirttiğimiz uzaklık ölçüleri[1] değişkenlerin ölçü birimlerinden etkilenmektedirler. Kullanılan bir ölçü biriminde iki birey birbirine uzakken, ölçü birimi değiştirildiğinde birbirine en yakın hale gelebilir, bireyler arasındaki sıralama değişebilir bu sebeple, uzaklık hesabından önce değişkenlerin standartlaştırılması gerekmektedir (Aldenderfer & Blashfield 1984).

Bir nesnenin bazı özellikleri için farklı ölçekler kullanılıyorsa, Öklit uzaklık fonksiyonu kullanılarak büyük ölçeklerle ölçülen nitelikler küçük ölçekle ölçülen niteliğe baskın gelebilir. Bu tip bir sorun ile karşılaşmamak için, nitelik değerleri genellikle 0 ile 1 arasında normalleştirilir. Öbekleme için veri kümesinden uzaklık ölçülerinden birinin kullanılmasıyla yakınlık matrisi oluşturularak öbekleme yöntemlerinden birisi ile bireyler öbeklere ayrılır (Bölüm 3.4.3'de öbekleme yöntemleri hakkında detaylı bilgi verilmiştir).

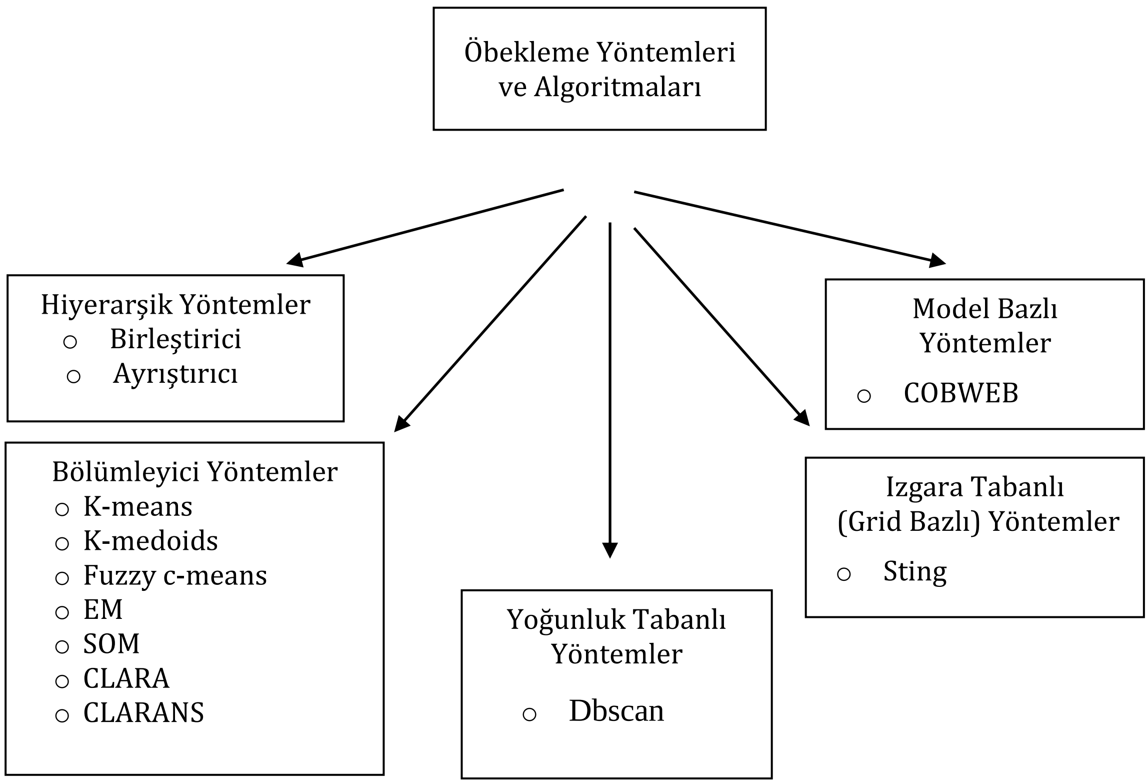
[1] Daha fazla uzaklık yöntemi için bakınız: (http://mathworld.wolfram.com)

### 3.4.3 Öbekleme Yöntemleri

Tez konusu öbekleme problemi olmasından dolayı bu bölümde öbeklemede kullanılan yöntem ve algoritmaların kısa tanıtımı yapılacaktır.

Öbekleme yöntemleri, yakınlık matrisinden yararlanarak değişkenleri kendi içinde homojen ve kendi aralarında heterojen gruplara ayıran yöntemlerdir. Öbeklemenin fen bilimleri ve sosyal bilimlerde yaygın kullanım alanlarının olması ve bilgisayar teknolojisinin gelişmekte olması, öbekleme yöntemlerinin artmasında büyük rol oynamaktadır. Literatürde birçok öbekleme metodu geliştirilmiştir (Han & Kamber 2006, Jain & Dubes 1988, Mercer 2003, Witten & Frank 1999). Milligan & Cooper öbekleme yöntemlerini dört farklı kategoriye ayırırken, Anderberg (1973) en kabul gören halini ortaya koymuş, bölümleyici (*partitioning*) ve hiyerarşik (*hierarchical*) öbekleme metotları olarak sınıflandırmıştır (Özdamar 2004). Çeşitli öbekleme metotlarının yer aldığı farklı sınıflamalar da yapılabilir. Buna ilişkin bir sınıflama örneğini öbekleme metot ve uygun algoritmalarını Şekil 3.8'de bulabilrsiniz.

**Şekil 3.8: Öbekleme yöntemleri**



Çok bilinen öbekleme algoritmaları k-means, bölümleyici, hiyerarşik, SOM, DBSCAN (*Density – Based Spatial Clustering of Applications with Noise),* EM (*Expectation Maximization*) ve CURE (*Clustering Using REpresentatives*) olarak sıralanabilir.

### 3.4.4 Bölümleyici (Partitioning) Öbekleme Yöntemleri

Bu metotlar, verilerin doğal olarak oluşturdukları grupları yeniden gruplamayı amaçlar. Bu nedenle verileri, örüntü matrisleri olarak alıp işleme sokmaktadırlar. Genellikle öznitelikler oransal değerlerdir ve $n$ tane nesnenin bulunduğu $d$ boyutlu bir uzayda (veri setinde), her bölümün bir öbeği temsil ettiği $k$ tane $(k \leq n)$ gruba ayıran metotlardır. Öbekler, nesneler arasındaki *benzerliklere* göre oluşturulur. *Öbekleme işlemi sonucunda aynı öbekteki nesneler (örüntüler) arası benzerlik maksimum, farklı öbeklerdeki nesneler arası benzerlik minimum olur.*

Bir kümenin merkezi ya kümedeki bütün noktaların bir ortalaması (*centroid*) ya da kümeyi en iyi temsil edecek nokta olan en merkez noktadır (*medoid*). Bütün teknikler merkez noktanın öbeği temsil etmesi esasına dayanmaktadır. Ugulanabilirliği kolay olan bölünmeli öbekleme algoritmalarında $k$ öbek sayısı *giriş parametresi* olarak verilir.

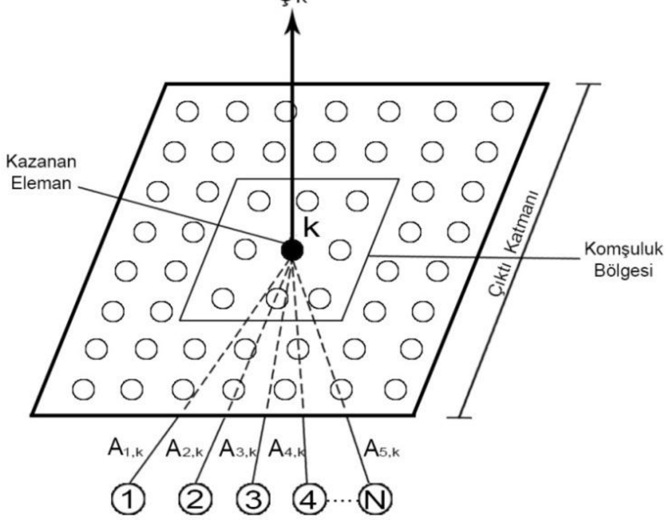
### 3.4.5 k – Ortalama (k – Means) Yöntemi

En eski, en basit ve en yaygın kullanılan eğiticisiz öğrenme algoritmalarından biri olan $k$ – ortalama yönteminde $k$ oluşacak *öbek sayısını, ortalama* ise öbeği oluşturacak nesnelerin ağırlıklı ortalamasını ifade eder. Rasgele bir paylaşım ($k$ tane nokta seçimi) ile başlayan algoritma, nesneleri öbek merkezlerine olan uzaklıkları dikkate alınarak en benzer oldukları öbeklere atar. Daha sonra her öbeğin ortalama değerleri hesaplanır, yeni öbek merkezleri bulunur ve bir önceki adım tekrarlanır. Herhangi bir yakınsama gerçekleşinceye kadar (öbeklerde bir değişiklik olmayıncaya kadar) adımlar tekrarlanır. Uzaklık için daha çok Öklit uzaklık ölçümü tercih edilmektedir. Başlangıç öbek merkezi seçimi algoritmanın sonucunu etkilemektedir. Bu algoritmada amaç, öbek merkezlerinin en uzak, öbek içi nesnelerin en yakın olmasıdır. İyi bir öbekleme için deneme yanılma yöntemi gerekebilir. Daha fazla bilgi için Duda R. , Hart P. , Stork D. Pattern Calssification sf 526 – 527 ve Everitt B. , Landau S. Cluster Analysis sf 122 – 130 incelenebilir.

### 3.4.6 Kendi Kendini Düzenleyen Haritalar (Self Organizing Maps-SOM) Yöntemi

Öbekleme çalışmalarında, yapay sinir ağları da kullanılabilir. Bir veri kümesindeki eleman ve değişken sayısının çok fazla olması sinir ağları için bir zorluk yaratmaz. Öbekleme çalışmalarında en çok kullanılan yapay sinir ağları, Kohonen (1981) tarafından ortaya koyulan kendi kendine öğrenen haritalar olarak bilinen (*Self-Organizing Maps* – SOM) sinir ağlarıdır (Kohonen 2001). Bu sebeple, Kohonen SOM ağları olarak da bilinir. Literatürde SOM ağı olarak geçen algoritma hem verilerin öbeklenmesi hem de görselleştirmesi yönünden tercih edilir. Dolayısıyla *k* – ortalamanın işlevlerini yerine getirebilir. Tek katmanlı bir ağ olan SOM, giriş ve çıkış nöronlarından oluşur. Giriş nöronlarının sayısı veri setindeki değişken sayısına göre belirlenir. Çıkış nöronlarının her biri bir öbeği temsil eder. Şekil 3.9'da bir SOM ağı görülmektedir. Diğer yapay sinir ağlarından farklı olarak, çıkış katmanındaki nöronların dizilimi çok önemlidir. Bu dizilim doğrusal, dikdörtgensel, altıgen veya küp şeklinde olabilir. En çok dikdörtgensel ve altıgen şeklindeki dizilimler tercih edilmektedir. Pratikte, çoğu kez dikdörtgensel dizilim karesel dizilim olarak uygulanır. Buradaki dizilim topolojik komşuluk açısından önemlidir. Aslında, çıkış nöronları arasında doğrudan bir bağlantı yoktur. Giriş nöronları ile her bir çıkış nöronu arasındaki bağlantıyı referans vektörleri (*code-book vectors*) gösterir. Bu vektörler bir katsayılar matrisinin sütunları olarak da düşünülebilir. SOM sinir ağları eğitilirken, bu topolojik komşuluk referans vektörlerinin yenilenmesinde kullanılır.

**Şekil 3.9: Som ağının gösterimi**




*Kaynak:* Kohonen 1984.

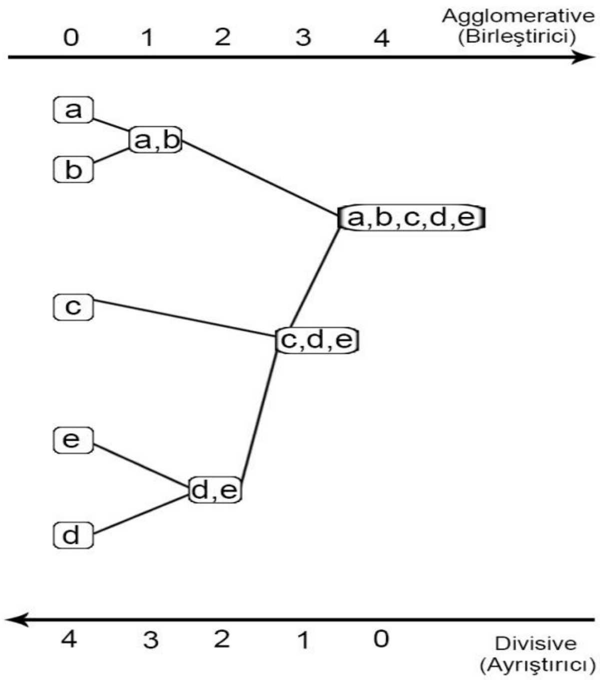
Şekil (3.9)'da sadece kazanan eleman ile girdi elemanlarının arasındaki bağlantılar gösterilmiştir. Aslında girdi elemanları çıktı elemanlarının tamamına bağlıdır.

### 3.4.7 Hiyerarşik (Hierarchical) Yöntemler

Hiyerarşik öbeklemede iki farklı kümeyi birleştirmek veya ayırmak için *yakınlık* matrisleri kullanılır. Şekilce ağaca benzer bir yapının (dendogram) oluşturulmasını sağlayan algoritmalara dayanmaktadırlar. Bu dendogramlar iki farklı teknikle üretilmektedir: Birincisi veri nesnelerini veya küçük grupları birleştirerek oluşturulan Birleştirici Öbekleme (*Agglomerative Clustering*), ikincisi büyük grupları bölerek oluşturulan Bölücü Öbekleme (*Divisive Clustering*). AGNES (*AGglomerative Nesting*) algoritması aşağıdan yukarı doğru, DIANA (*DIvisive ANAlysis*) algoritması yukarıdan aşağı doğru çalışan bir inşa yapısı izler.

AGNES algoritması, başlangıçta örneklerin hepsini ayrı ayrı birer öbek kabul edip her bir adımında öbeklerin birbirlerine olan uzaklıklarına bakarak en yakın iki öbeği birleştirir. DIANA algoritması ise başlangıçta bütün örnekleri aynı öbekte kabul edip her bir adımda birbirine en uzak iki öbeği bulmaya çalışır. Diğer algoritmalardan farkı örneklerin koordinatlarına ihtiyaç duymayıp yakınlık matrisinin bu algoritma için yeterli olmasıdır (yakınlık matrisi ile yetinmesidir).

**Şekil 3.10: Birleştirici ve ayrıştırıcı hiyerarşik öbekleme**




*Kaynak:* Kaufman & Rousseeuw 1990

Hiyerarşik öbekleme algoritmaları ve daha fazla bilgi için Duda, Hart and Stork 2001 incelenebilir.

# 4. ÇOK KERNELLİ ÖĞRENME YÖNTEMLERİ

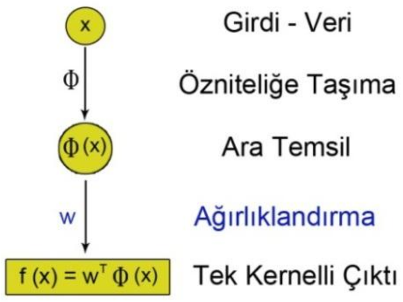
## 4.1 GİRİŞ

Lineer olarak ayrılamayan veri kümelerinde, kernel fonksiyonlarının kullanılmasıyla veriler yüksek boyutlu öznitelik uzayına taşınarak lineer ayrılabilir hale getirilmektedir. Tek kernel fonksiyonu ile yapılan bu sınıflandırma işlemleri, genellikle gerçek hayat problemlerinde karşılaştığımız (görüntü tanıma, biyoinformatik veri kümeleri gibi farklı kaynaklardan gelen veriler) çok boyutlu ve heterojen kaynaklı veri kümelerinde lineer olmayan veriyi sınıflandırmakta yetersiz kalmaktadır. Bu tür veriyi sınıflandırmak için kernellerin konveks kombinasyonlarından oluşan çok kernelli öğrenim (*multiple kernel learning*) yöntemi geliştirilmiştir (Bach ve diğ. 2004). Bu yöntem ile farklı kaynaklardan gelen farklı kernel fonksiyonlarını Şekil 4.1 (b)'de gösterildiği gibi DVM yöntemi ile sınıflandırmak mümkündür. El yazısı tanıma, yüz tanıma ve tıp alanındaki görüntülerin sınıflandırılması çalışmalarında çok kernelli algoritmalar kullanılabilir.

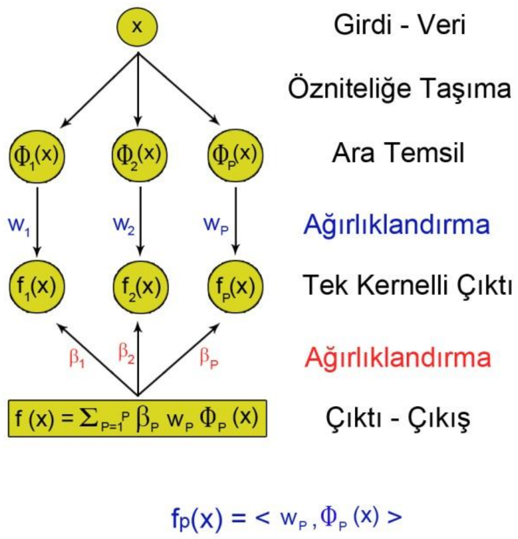
Son yıllarda yapılan birçok çalışmada çok kernelli yöntemler önerilmiştir (Nash & Sofer 1996, Joachims 1999, Bach ve diğ. 2004, Nemirovski 2004, Fan ve diğ. 2005, Rubinstein 2005, Sonnenburg ve diğ. 2005 and 2006, Gönen ve diğ. 2008, Akyüz 2009, Xu ve diğ. 2009, Kloft ve diğ. 2011). El yazısı tanıma, yüz tanıma ve tıp alanındaki görüntülerin sınıflandırılması çalışmalarında çok kernelli algoritmalar kullanılabilir.

Farklı kerneller değişik benzerlik ölçütleri tanımlamaktadır ve farklı kaynaklardan gelen veya farklı özelliklerdeki verileri birleştirmek için de kullanılabilir.

**Şekil 4.1.(a): Tek Kernelli DVM**




**Şekil 4.1.(b): Çok Kernelli DVM**




*Kaynak*: Zien 2008

Bölüm 4.2'de bu tezdeki modelleme için yol gösterici olan çok kernelli öğrenme probleminin özgün (*primal*) formu verilecek ardından optimizasyon problemi incelenecektir.

## 4.2 ÇOK KERNELLİ ÖĞRENME – (MULTİPLE KERNEL LEARNING-MKL)

### 4.2.1 Çok Kernelli Sınıflandırma

Destek vektör makineleri gibi kernel tabanlı yöntemlerin sınıflandırma problemleri için etkili olduğu kanıtlanmıştır. Son yıllarda yapılan çalışmalar, farklı kaynaklardan gelen lineer olmayan verilerin sınıflandırılması için, çok kernelli öğrenme yönteminin daha etkili olduğu yönündedir. Sezgisel olarak $\mathbf{x}_i$ ve $\mathbf{x}_j$ gibi iki örnek arasındaki benzerliği hesaplayan $K\left(\mathbf{x}_i, \mathbf{x}_j\right)$ kernel fonksiyonu kullanılmaktadır. Heterojen kaynaklı verilerin kernel ağırlıklı kombinasyonu farklı kaynaklardan gelen verilerin her biri için ayrı ayrı benzerlik ölçümü tanımlamamızı sağlar. Kernellerin konveks kombinasyonları aşağıda verilen denklem (4.1)'de olduğu gibi ifade edilir (Sonnenburg ve diğ. 2006 ):

$$K\left(\mathbf{x}_i, \mathbf{x}_j\right) = \sum_{\kappa=1}^{K} \beta_\kappa K_\kappa\left(\mathbf{x}_i, \mathbf{x}_j\right), \qquad \mathbf{(4.1)}$$

öyle ki, $\beta_\kappa \geq 0$, kernel katsayısı olmak üzere ve $\sum_{\kappa=1}^{K} \beta_\kappa = 1$'dir.

Girdi vektörü $\mathbf{x_i}$ $(i = 1, \dots, n)$, $K$ tane taşıma fonksiyonu $\Phi_\kappa$ ile taşınarak öznitelik uzayında $\Phi_\kappa : x \mapsto \Phi_\kappa(\mathbf{x}) \in R^{D_\kappa}$ $(\kappa = 1, \dots, K)$ halini alır. Girdi uzayı $R^n$'den, $R^{D_\kappa}$ içine, $\kappa$. öznitelik uzayında

$$K_\kappa(\mathbf{x}_i, \mathbf{x}_j) = \langle \Phi_\kappa(\mathbf{x}_i), \Phi_\kappa(\mathbf{x}_j) \rangle \qquad \mathbf{(4.2)}$$

elde edilmiş olur (Akyüz 2009).

Sapma terimi (*bias*) $b$, $y_i \in \{\pm 1\}$, $\mathbf{x}_i$ girdi vektörlerinin sınıf etiketi olacak biçimde, hiperdüzlemin fonksiyonu

$$f(\mathbf{x}) = \sum_{\kappa=1}^{K} \langle \mathbf{w}_\kappa, \Phi_\kappa(\mathbf{x}_i) \rangle + b \qquad \mathbf{(4.3)}$$

ile tanımlanır. Literatürde çok kernelli öğrenme özgün problemi denklem (4.4) ile tanımlanmıştır (Bach ve diğ. 2004, Sonnenburg ve diğ. 2006):

***Özgün (Primal) Problem***

$$\min_{w,\ \xi,\ b} \quad \frac{1}{2}\left(\sum_{\kappa=1}^{K} \|\mathbf{w}_\kappa\|_2\right)^2 + C \sum_{i=1}^{n} \xi_i \qquad \mathbf{(4.4)}$$

öyle ki $\quad \mathbf{w}_\kappa \in R^{D_\kappa}$ $(\kappa = 1, \dots, K)$, $\quad b \in R$,

kısıtlar $\quad \boldsymbol{\xi} \in \mathbb{R}_+^n$,

$$y_i\left(\sum_{\kappa=1}^{K} \mathbf{w}_\kappa^T \Phi_\kappa(\mathbf{x_i}) + b\right) \geq 1 - \xi_i \quad (i = 1, \dots, n), \qquad \mathbf{(4.5)}$$

$$\sum_{\kappa=1}^{K} \beta_\kappa = 1.$$

Problem (4.4)'de amaç fonksiyonunda $(\sum_{\kappa=1}^{K}\|\mathbf{w}_{\boldsymbol{\kappa}}\|_2)^2$ terim ile her biri ayrı ayrı bloklarda ağırlık vektörü $\mathbf{w}_\kappa = \beta_\kappa\, \mathbf{w}'_\kappa$, $l_2$ – norma cezalandırılırken, $\boldsymbol{\beta}$ vektörü $l_1$ – norma kısıtlanmıştır. Burada $\boldsymbol{\beta}$ vektörü $l_1$ – norm kısıtından dolayı seyrek gösterime sahip olur ve seyrek optimal çözüm verir. Bu işlem makinenin sadece katsayısı $\beta_{\boldsymbol{i}}$ sıfırdan farklı kernelleri hafızada tutmasına, böylece uygulamada kolaylık sağlanmasına sebep olur. Dolayısıyla problemin çözümü kolaylaşır. Amacımız problemde optimal $\boldsymbol{\beta}$ ve $\mathbf{w}_\kappa$ ları bulmaktır.

İkinci dereceden konik (*second – order cone* [2]) ve $\frac{1}{2}\gamma^2 \longrightarrow \gamma$ dönüşümünden yararlanarak özgün problemin dual formu problem (4.6) elde edilir (Bach ve diğ. 2004):

***Dual Problem***

$$\min_{\gamma\in R\,,\boldsymbol{\alpha}\in R^n} \; -\gamma - \sum_{i=1}^{n} \alpha_i \qquad (\mathbf{4.6})$$

kısıtlar $\quad 0 \le \alpha_i \le C,\ \sum_{i=1}^{n} \alpha_i y_i = 0,$

$$\sum_{i,j=1}^{n} \alpha_i\alpha_j y_i y_j K_\kappa(\mathbf{x}_i, \mathbf{x}_j) \le \gamma \qquad (\kappa = 1,2, \dots, K).$$

Denklem (4.2)'de $K_\kappa(\mathbf{x}_i, \mathbf{x}_j)$ verilmişti. Problem (4.6)'da $\gamma - \sum_{i=1}^{n} \alpha_i \rightarrow \gamma$ dönüşümü yapılırsa problem (4.7)'de verilen dual* problem elde edilir:

[2] Second – order cone: $U \subseteq R^n$ konveks olsun. U tarafından üretilen hafif konik,

$$\mathcal{K}_2 = \left\{ \begin{bmatrix} x \\ t \end{bmatrix} \in R^n \times R \,\middle|\, \|x\|_2 \le t \right\}$$

dışbükey bir konidir. (Dattorro J. 2011)

***Dual* Problem***

$$\min_{\gamma\in R,\boldsymbol{\alpha}\in R^n} \quad \gamma \tag{4.7}$$

$$\text{kısıtlar} \quad 0 \le \alpha_i \le C, \qquad \sum_{i=1}^{n} \alpha_i y_i = 0,$$

$$\sum_{i,j=1}^{n} \alpha_i\alpha_j y_i y_j K_\kappa(\mathbf{x}_i,\mathbf{x}_j) - \sum_{i=1}^{n}\alpha_i \le \gamma \qquad (\kappa = 1,2,\dots,K).$$

Problem (4.7)'de verilen eşitsizlik kısıtını işlem ve gösterim kolaylığı açısından,

$$S_\kappa(\alpha) \coloneqq \frac{1}{2}\sum_{i,j=1}^{n} \alpha_i\alpha_j y_i y_j K_\kappa(\mathbf{x}_i,\mathbf{x}_j) - \sum_{i=1}^{n}\alpha_i \tag{4.8}$$

ile tanımlayalım. Problem (4.7)'i çözmek için denklem (4.9) ile verilen büküm noktası problemi çözülebilir:

$$\max_{\beta} \quad \min_{\alpha} \quad \gamma + \sum_{\kappa=1}^{K} \beta_\kappa (S_\kappa(\alpha) - \gamma), \tag{4.9}$$

öyle ki $\gamma \in R$, $\boldsymbol{\alpha} \in R^n$ $(0 \le \alpha_i \le C,\ \sum_{i=1}^{n}\alpha_i y_i = 0)$. Denklem (4.9) da $\gamma$' ya göre türev alır, sıfıra eşitlersek, $\boldsymbol{\beta} \in R^K$, $\beta \ge 0$ için $\sum_{\kappa=1}^{K}\beta_\kappa = 1$ olduğu bulunur. Böylece denklem (4.9) daha basit hale gelir ve $S(\alpha,\beta) \coloneqq \sum_{\kappa=1}^{K}\beta_\kappa S_\kappa(\alpha)$ olarak tanımlanabilir. Buradan fonksiyonu minimum yapan $\boldsymbol{\alpha}$ ve aynı anda maksimum yapan $\boldsymbol{\beta}$ değerini bulmak için min – max problemi yazılabilir:

***Min – Max Problem***

$$\max_{\boldsymbol{\beta}\in R^K} \quad \min_{\boldsymbol{\alpha}\in R^n} \quad \sum_{\kappa=1}^{K} \beta_\kappa S_\kappa(\alpha) \tag{4.10}$$

$$\text{kısıtlar} \quad 0 \le \alpha_i \le C, \qquad 0 \le \beta, \quad \sum_{i=1}^{n}\alpha_i y_i = 0 \ \ ve \ \sum_{\kappa=1}^{K}\beta_\kappa = 1.$$

Optimum çözüm, $\boldsymbol{\alpha}^*$ için $\Theta^* := S\,(\boldsymbol{\alpha}^*,\ \boldsymbol{\beta})$ ve her $\boldsymbol{\alpha}$ için $S\,(\boldsymbol{\alpha}, \boldsymbol{\beta})\ \geq\ \Theta^*$ olduğunu varsayalım. Böylece problem (4.11) ile verilen lineer yarı sonsuz programlama elde edilir.

***Lineer - Yarı - Sonsuz Programlama LYSP (Linear-semi - infinite programming LSIP)***

$$\max_{\Theta\in R, \beta\in R^K} \quad \Theta \tag{4.11}$$

$$\text{kısıtlar} \quad 0\ \leq \beta, \quad \sum_{\kappa=1}^{K} \beta_\kappa = 1 \quad ve \quad \sum_{\kappa=1}^{K} \beta_\kappa S_\kappa(\alpha)\ \geq\ \Theta,$$

$$\forall\, \boldsymbol{\alpha} \in R^n, \qquad 0\ \leq \alpha_i\ \leq C, \qquad \sum_{i=1}^{n} \alpha_i y_i = 0.$$

Bölüm 4.2.2’de Bach (2004) ve Sonnenburg (2006)’un çalışmalarından yola çıkarak, en büyük aralık ile öbekleme YSP ile modellenecektir.

### 4.2.2 En Büyük Aralık İle Öbekleme (Maximum Margin Clustering - MMC)

Bu bölümde kısaca DVM’nin en büyük aralık prensibinden yola çıkarak, en büyük aralık ilkesinin çok kernel ile öbeklemeye uygulaması anlatılacaktır.

Öbekleme algoritmaları verileri önceden belirlenen kurallara göre gruplamaktadır. DVM ile öbeklendirme yaparken, lineer olmayan girdi verileri, uygun kernel fonksiyonları ile daha yüksek boyutlu öznitelik uzayına taşınmakta, öbekleme bu uzayda yapılmaktadır. Veriler merkezden en büyük aralık ile ayırabilecek hiperdüzlem aracılığı ile öbeklere ayrılmaktadır. Bu işlemler sırasında önemli olan kısım, makinenin eğitim sırasında hiçbir sınıf bilgisini kullanmamasıdır. Anlaşılacağı gibi problemde $y_i$ etiket değerleri yoktur.

Son yıllarda, Eltoft ve de Figueiredo (1998) tarafından YSA tabanlı ve Ben-Hur ve diğ. (2001) tarafından da DVM tabanlı karmaşık yapılı öbekleme metotları geliştirilmiştir.

Bu metotlar lineer olmayan verileri öbekleyebilmekte ve etiket bilgisine de gerek duymayan karmaşık yapılı metotlardır**.**

İki sınıfın öbeklenmesi, veri kümesi $X = \{x_1,\ x_2, \dots, x_n\}\ (x_i \in R^n)$ için en iyi etiket kombinasyonunu $y = \{y_1, \dots, y_n\} \in \{-1,1\}^n$ bulmayı amaçlamaktadır. Bölüm 3.3.2'de verilen DVM ilkeleri kullanılarak, aşağıda problem ($\mathrm{P}_A$) ile verildiği üzere En Büyük Aralık ile Öbekleme (EAO) optimizasyon modeli, öbekler arası en büyük aralığı veren etiketleri ve hiperdüzlem parametreleri **w** ve b'yi bulacaktır (Zhao ve diğ. 2009):

$$\min_{y \in \{\pm 1\}} \ \min_{w,b,\xi_i} \quad \frac{1}{2}\mathbf{w}^{\mathbf{T}}\mathbf{w} + \mathrm{C}\sum_{i=1}^{n} \xi_i \qquad (\mathbf{P}_{\boldsymbol{A}})$$

$$\text{kısıtlar} \qquad y_i(\mathbf{w}^{\mathrm{T}}\Phi(\mathbf{x}_i) + b) \ \geq \ 1 - \xi_i\,, \quad \xi_i \geq 0 \ \ \forall\, i,$$

$$-l \ \leq \ \sum_{i=1}^{n} y_i \ \leq \ l \quad (i = 1, \dots, n). \qquad (\mathbf{4.12})$$

Problem ($\mathrm{P}_A$), $y \in \{\pm 1\}$ ve $w, b, \xi$ reel uzayda tanımlı değişkenler oldukları için *karışık tamsayılı programlama (mixed integer programming)* adını alır. Burada X veri örnekleri, **Φ** fonksiyonu ile yüksek boyutlu öznitelik uzayında lineer olmayan dönüşüm yaparlar. Problem ($\mathrm{P}_A$)'da eşitsizlik (4.12) ile verilen son kısıt, bu sınıf için denge kısıtıdır. Bütün sınıfları tek bir grupta sınıflayan aşikar optimal sınıflandırmayı engellemek için tanıtılmış olup sınıflar arası dengesizliği kontrol etmek için $\boldsymbol{l} > 0$ parametresi alınmıştır.

Problem ($\mathrm{P}_A$)'yı tamsayı programı ile çözmek, ikinci dereceden (kuadratik) program ile çözmekten daha zordur. Buradan hareketle EAO problemindeki tamsayı (*binary*: 1 ve $-1$ gibi) ikili değişkeni mutlak değer ile problem (4.13) gibi yazılarak sürekli optimizasyon problemi haline getirilebilir:

$$\min_{w,b,\xi_i} \quad \frac{1}{2}\mathbf{w}^{\mathbf{T}}\mathbf{w} + \mathrm{C}\sum_{i=1}^{n}\xi_i \qquad (\mathbf{4.13})$$

$$\text{kısıtlar} \quad |\mathbf{w}^{\mathrm{T}}\Phi(\mathbf{x}_i) + b| \;\geq\; 1 - \xi_i\,, \quad \xi_i \geq 0 \quad (i\,\ldots, 2, \ldots, \mathrm{n}),$$

$$-l \;\leq\; \sum_{i=1}^{n}[\mathbf{w}^{\mathrm{T}}\Phi(\mathbf{x}_i) + b] \;\leq\; l.$$

Yukarıda verilen problem (4.13)'ün çözümünden, veri örneklerinin çıkış etiketleri $y_i = sgn(\mathrm{w}^{\mathrm{T}}\Phi(\mathbf{x}_i) + b)$ hesaplanmış olur (Zhao ve diğ. 2009).

Bin Zhao ve diğ. (2009) tarafından yukarıda problem (4.13) ile tanıtılan EAO (MMC) problemi, ikinci dereceden konik program (Second order cone program) kullanılarak kesen düzlem (cutting plane) algoritması ile çözülmüştür.

### 4.2.3 Çok Kernelli En Büyük Aralık İle Öbekleme (Multiple Kernel Maximum Margin Clustering)

Bu bölümde, en büyük aralığı bulan hiperdüzlem ile iki sınıfı birbirinden ayıran, çok kernelli öbekleme algoritması incelenecektir. Genel olarak, Bölüm 4.1'de bahsedildiği gibi, lineer olmayan $\boldsymbol{\Phi}$ fonksiyonu veri örneklerini öznitelik uzayına $\left(\Phi_\kappa : \mathbf{x} \longrightarrow \Phi_\kappa(\mathbf{x})\right)$ taşırlar. Sınıflandırma probleminde, çok kernelli öğrenme çalışmalarından hareketle (Bach ve diğ. 2004, Akyüz 2009, Gönen ve Alpaydın 2008, Lanckriet ve diğ. 2004a, 2004b, Rakotomamonjy ve diğ. 2007, Sonnenburg ve diğ. 2006) kernel kombinasyonları EAO yöntemi için önerilmiştir.

Girdi uzayından, öznitelik uzayına negatif olmayan ve lineer olmayan $\Phi_1, \ldots, \Phi_M$ fonksiyonlarının lineer kombinasyonlarını ele alalım (Zhao ve diğ. 2009):

$$\Phi(\mathbf{x}) = \sum_{\kappa=1}^{K}\beta_\kappa\Phi_\kappa(\mathbf{x}) \qquad (\mathbf{4.14})$$

öyle ki, $\beta_\kappa \geq 0$ ve bazı $p$ tamsayıları için $\sum_{\kappa=1}^{M}\beta_\kappa^p \leq 1$ olsun.

Zhao ve diğ. (2009), çok kernelli öbekleme problemini problem (4.13)'e eşdeğer olan problem $(P_Z)$'deki gibi yazmışlardır. Amaç fonksiyonunu optimize ederek hem hiperdüzlem parametreleri (**w**,b), hem de lineer kombinasyonun parametreleri $(\beta_\kappa)$ elde edilir:

$$\min_{\beta,w,b,\xi} \quad \frac{1}{2}\sum_{\kappa=1}^{K} \beta_\kappa \|\mathbf{w}_\kappa\|^2 + \mathrm{C}\sum_{i=1}^{n} \xi_i \tag{$\boldsymbol{P_Z}$}$$

$$\left|\sum_{\kappa=1}^{K} \beta_\kappa \mathbf{w}_\kappa{}^T\, \Phi_\kappa(\mathbf{x_i}) + b\right| \geq 1 - \xi_i\,, \quad \xi_i \geq 0 \qquad (i = 1, \dots, n),$$

$$\beta_\kappa \geq 0 \qquad (\kappa = 1, \dots, K),$$

$$\sum_{\kappa=1}^{K} \beta_\kappa^p \leq 1,$$

$$-l \leq \sum_{i=1}^{n}\left[\sum_{\kappa=1}^{K} \beta_\kappa \mathbf{w}_{\boldsymbol{\kappa}}{}^T\, \Phi_\kappa(\mathbf{x_i}) + b\right] \leq l, \qquad l > 0,$$

öyle ki, $\boldsymbol{\beta} = (\beta_1, \dots, \beta_\kappa)^T$ olmak üzere, $p = 2$ alınarak katsayılar $l_2$ norm ile kısıtlanmıştır.

Oysaki bu tezde $\sum_{\kappa=1}^{K} \beta_\kappa = 1$ ve amaç fonksiyonunun parantez karesini alınmıştır (bknz problem (CP)). Amaç fonksiyonunun parantez karesinin alınması, **w** vektörlerinin blok içinde seyrek olmamasını (*non sparse*), **β** vektörlerinin $l_1$norma kısıtlanması ise **β** vektörlerini blok seviyesinde seyrek (*sparse*) olmasını sağlayacaktır. Seyrek $\boldsymbol{\beta}$ vektörü bularak, karar fonksiyonu daha kolay ve rahat ifade edilebilecek hale gelir, bu da hesaplamayı kolaylaştıracaktır. Bu nedenle Bölüm 4.1, denklem (4.4) ile verilen modelde, en büyük aralık öbeklemesi için optimizasyon problemi, $\beta_\kappa \geq 0$ ,$\forall\ \kappa \in \{1, \dots, K\}$ için $\sum_{\kappa=1}^{K} \beta_\kappa = 1$ ve $\mathbf{w}_\kappa = \beta_\kappa\, \mathbf{w'}_\kappa$ eşitlikleri dikkate alınarak, aşağıda problem (CP)'deki gibi ifade edilebilir:

$$\min_{w,\ b\ ,\ \xi} \quad \frac{1}{2}\left(\sum_{\kappa=1}^{K}\|\mathbf{w}_\kappa\|_2\right)^2 + \mathrm{C}\sum_{i=1}^{n}\xi_i \qquad (\boldsymbol{CP})$$

$$\left|\sum_{\kappa=1}^{K}\mathbf{w}_\kappa{}^T\,\Phi_\kappa(x_i)+b\right| \geq\ 1-\xi_i,\quad \xi_i\geq 0\quad (i=1,\dots,n), \qquad (\mathbf{4.15})$$

$$-l\ \leq \sum_{i=1}^{n}\left[\sum_{\kappa=1}^{K}\mathbf{w}_\kappa{}^T\,\Phi_\kappa(x_i)+b\right] \leq\ l, \qquad l>0.$$

Yukarıda tanımladığımız problem (CP)'nin birinci kısıtı (4.15) mutlak değerden dolayı konveks değildir. Birinci kısıt mutlak değerden çıkartılırsa aşağıda $(CP_A)$ ile verilen "**veya**" kısıtlı konveks olmayan problemi verecektir:

$$\min_{w,\ b\ ,\ \xi} \quad \frac{1}{2}\left(\sum_{\kappa=1}^{K}\|\mathbf{w}_\kappa\|_2\right)^2 + \mathrm{C}\sum_{i=1}^{n}\xi_i \qquad (\boldsymbol{CP_A})$$

kısıtlar $\quad \xi_i \geq 0 \quad (\forall\ i=1,\dots,n),$

$$\sum_{\kappa=1}^{K}\mathbf{w}_\kappa{}^T\,\Phi_\kappa(\mathbf{x_i})+b\leq\ \ \xi_i-1\ \ \textbf{veya}\ \ \sum_{\kappa=1}^{K}\mathbf{w}_\kappa{}^T\,\Phi_\kappa(\mathbf{x_i})+b\leq\xi_i-1\ \ \forall i,$$

$$-l\ \leq \sum_{i=1}^{n}\left[\sum_{\kappa=1}^{K}\mathbf{w}_\kappa{}^T\,\Phi_\kappa(\mathbf{x_i})+b\right] \leq\ l, \qquad l>0.$$

Problem $(CP_A)$' nın "**veya**" kısıtları yeterince büyük *D* sayısı ve yeni tanıtılacak olan ikili değişken $z\in\{0,1\}$ ile "**veya**" kısıtından kurtararak aşağıdaki Karışık Tamsayı (*Mixed Integer Problem*) problemine dönüştürülebilir:

$$\min_{w,\ b,\ \xi,\ y} \quad \frac{1}{2}\left(\sum_{\kappa=1}^{K}\|\mathbf{w}_\kappa\|_2\right)^2 + \mathrm{C}\sum_{i=1}^{n}\xi_i \qquad (\boldsymbol{CP_B})$$

$$\text{kısıtlar} \quad \xi_i \geq 0, \quad \forall\ i = \{1, \dots, n\},$$

$$-\sum_{\kappa=1}^{K}\mathbf{w}_\kappa{}^T\,\Phi_\kappa(\mathbf{x_i}) + b \leq \ \xi_i - 1 + Dz \qquad \forall i,$$

$$\sum_{\kappa=1}^{K}\mathbf{w}_\kappa{}^T\,\Phi_\kappa(\mathbf{x_i}) + b \leq \ \xi_i - 1 + D\,(1 - z) \qquad \forall i,$$

$$z \ \in \{0,1\},$$

$$-l \ \leq \sum_{i=1}^{n}\left[\sum_{\kappa=1}^{K}\mathbf{w}_\kappa{}^T\,\Phi_\kappa(\mathbf{x_i}) + b\right] \leq \ l\,, \qquad l > 0.$$

Dikkat edilecek olunursa, $y$ değişkeni ikili (*binary*) olması sebebi ile sürekli optimizasyon problemini Karışık Tamsayı problemine dönüştürmüştür. Karışık Tamsayı problemlerini çözmek sürekli optimizasyon problemlerini çözmekten daha karmaşık olması ve problemimiz $(CP_B)$'ye özel sadece bir değişkenin tamsayı alması sebebi ile $z$ değişkenini $[0,1]$ aralığına gevşeteceğiz. O halde problem $(CP_B)$ aşağıdaki forma dönüşecektir:

$$\min_{w,\ b,\ \xi,\ y} \quad \frac{1}{2}\left(\sum_{\kappa=1}^{K}\|\mathbf{w}_\kappa\|_2\right)^2 + \mathrm{C}\sum_{i=1}^{n}\xi_i \qquad (\boldsymbol{CP_C})$$

$$\text{kısıtlar} \quad \xi_i \geq 0, \quad (i = 1, \dots, n),$$

$$-\sum_{\kappa=1}^{K}\mathbf{w}_\kappa{}^T\,\Phi_\kappa(\mathbf{x_i}) + b \leq \ \xi_i - 1 + Dz \qquad (i = 1, \dots, n),$$

$$\sum_{\kappa=1}^{K}\mathbf{w}_\kappa{}^T\,\Phi_\kappa(\mathbf{x_i}) + b \leq \ \xi_i - 1 + D\,(1 - z) \quad (i = 1, \dots, n),$$

$$0 \ \leq z \ \leq 1,$$

$$-l \ \leq \ \sum_{i=1}^{n} \left[ \sum_{\kappa=1}^{K} \mathbf{w}_{\kappa}{}^{T} \ \Phi_{\kappa}(\mathbf{x_i}) + b \right] \leq \ l, \qquad l > 0.$$

Yukarıda tanımlanan problem $(CP_C)$'nin Çok Kernelli Problem olduğunu hatırlayalım. O halde $\boldsymbol{\beta} = (\ \beta_1, .., \beta_\kappa\ )^T$ olmak üzere, $\mathbf{w}_\kappa = \beta_\kappa\, \mathbf{w'}_\kappa$ ... $\forall\ \kappa = 1, \ldots, K$ için $\beta_\kappa \geq 0$ ve $\sum_{\kappa=1}^{K} \beta_\kappa = 1$ eşitliklerini sağlayan $\beta_\kappa$ kernel katsayıları vardır.

Problem $(CP_C)$ *ikinci dereceden konik programlama* kullanılarak, $u \ \in R$ olmak üzere, $\sum_{\kappa=1}^{K} \|\mathbf{w}_\kappa\|_2 \ = u$ olarak tanımlanırsa problem $(CP_C)$'ye eşdeğer aşağıdaki problem $(CP_D)$ elde edilir:

$$\min_{u,\ \xi,\ y} \ \ \frac{1}{2}\mathrm{u}^2 + \mathrm{C} \sum_{i=1}^{n} \xi_i \qquad\qquad (\boldsymbol{CP_D})$$

kısıtlar $\ \xi_i \geq 0 \ \ (\ i = 1, \ldots, n),$

$$-\sum_{\kappa=1}^{K} \mathbf{w}_{\kappa}{}^{T} \ \Phi_{\kappa}(\mathbf{x_i}) + b \leq \ \ \xi_i - 1 + Dz \qquad\qquad (i = 1, \ldots, n),$$

$$\sum_{\kappa=1}^{K} \mathbf{w}_{\kappa}{}^{T} \ \Phi_{\kappa}(\mathbf{x_i}) + b \leq \ \ \xi_i - 1 + D\ (1 - z) \quad (i = 1, \ldots, n),$$

$$0 \ \leq \ z \ \leq 1\ ,$$

$$-l \ \leq \ \ \sum_{i=1}^{n} \left[ \sum_{\kappa=1}^{K} \mathbf{w}_{\kappa}{}^{T} \ \Phi_{\kappa}(\mathbf{x}_i) + b \right],$$

$$\sum_{i=1}^{n} \left[ \sum_{\kappa=1}^{K} \mathbf{w}_{\kappa}{}^{T} \ \Phi_{\kappa}(\mathbf{x}_i) + b \right] \leq l\ ,$$

$$\sum_{\kappa=1}^{K} \|\mathbf{w}_\kappa\| \leq u.$$

Problem $(CP_D)$, $t_\kappa \in R$, $(\kappa = 1, \dots, K)$ değişkeni kullanılarak aşağıdaki probleme basitleştirilebilir:

$$\min_{\substack{u \in R \\ \xi \in R \\ t \in R^K \\ \mathbf{w}_\kappa \in R^{D_K} \\ y}} \quad \frac{1}{2}\mathrm{u}^2 + \mathrm{C}\sum_{i=1}^{n} \xi_i \qquad (\boldsymbol{CP_E})$$

$$\xi_i \geq 0 \qquad (i = 1, \dots, n),$$

$$-\sum_{\kappa=1}^{K} \mathbf{w}_\kappa{}^T\, \Phi_\kappa(\mathbf{x}_\mathbf{i}) + b \leq \; \xi_i - 1 + Dz \qquad \forall i,$$

$$\sum_{\kappa=1}^{K} \mathbf{w}_\kappa{}^T\, \Phi_\kappa(\mathbf{x}_\mathbf{i}) + b \leq \; \xi_i - 1 + D\,(1 - z) \qquad \forall i,$$

$$0 \;\leq\; z \;\leq 1,$$

$$-l \;\leq\; \sum_{i=1}^{n} \left[\sum_{\kappa=1}^{K} \mathbf{w}_\kappa{}^T\, \Phi_\kappa(\mathbf{x}_i) + b\right],$$

$$\sum_{i=1}^{n} \left[\sum_{\kappa=1}^{K} \mathbf{w}_\kappa{}^T\, \Phi_\kappa(\mathbf{x}_i) + b\right] \leq l,$$

$$\|\mathbf{w}_\kappa\| \leq t_\kappa\,, \quad \sum_{\kappa=1}^{K} t_\kappa \;\; \leq u. \qquad (\mathbf{1})$$

Bölüm 4.1'de dip not [2] de tanımlanan ikinci dereceden koniğin (*second order cone*) boyutu D ile gösterilsin. Öyle ki, $\left(w_{\kappa\,,} t_\kappa\right) \in K_D$ için $K_D = \{(x, c) \in R^D x R\,,\;\; \|x\|_2 \leq c\}$ konisinde $D\; = 2$ alınarak $(CP_E)$ deki (1) numaralı $\|\mathbf{w}_\kappa\|_2 \leq t_\kappa$ kısıtı yazılabilir.

Böylece, ikinci dereceden koni programı kullanılarak özgün (*primal*) problem $(CKO)$ ile ifade edilebilir.

***Özgün (Primal) Problem***

$$\min_{\substack{u \in R \\ \xi \in R \\ t \in R^K \\ y}} \quad \frac{1}{2}\mathrm{u}^2 + \mathrm{C}\sum_{i=1}^{n} \xi_i \qquad (\boldsymbol{CKO})$$

öyle ki, $(\mathbf{w}_\kappa, t_\kappa) \in K_{D_K} \quad (\forall \kappa = 1, \ldots\ldots, K),$

$$\xi_i \geq 0,$$

$$-\sum_{\kappa=1}^{K} \mathbf{w}_\kappa{}^T \, \Phi_\kappa(\mathbf{x_i}) + b \leq \; \xi_i - 1 + Dz \qquad (i = 1, \ldots, n),$$

$$\sum_{\kappa=1}^{K} \mathbf{w}_\kappa{}^T \, \Phi_\kappa(\mathbf{x_i}) + b \leq \; \xi_i - 1 + D\,(1 - z) \quad \forall i,$$

$$0 \; \leq \; z \; \leq 1,$$

$$-l \; \leq \; \sum_{i=1}^{n} \left[ \sum_{\kappa=1}^{K} \mathbf{w}_\kappa{}^T \, \Phi_\kappa(x_i) + b \right],$$

$$\sum_{i=1}^{n} \left[ \sum_{\kappa=1}^{K} \mathbf{w}_\kappa{}^T \, \Phi_\kappa(x_i) + b \right] \leq l,$$

$$\sum_{\kappa=1}^{K} t_\kappa \quad \leq u.$$

Yukarıdaki problem $(CKO)$'da verilen eşitsizlikler kısıtlı bir optimizasyon problemidir. Dual koniğin temel değişkenlerini elde etmek için Lagrange denklemini yazdığımızda kernel fonksiyonunu $(\text{denklem}(4.2))$ doğrudan iç çarpım ile elde etmemize de olanak sağlar. Bu nedenle her bir eşitsizlik kısıtı için negatif olmayan Lagrange çarpanları $\propto_i,\ \widetilde{\beta_\iota},\ \mu_i,\ \lambda, \gamma, \sigma, \upsilon, \varkappa, \vartheta_\kappa, \xi_\kappa, \zeta_\kappa\ ,\ (i = 1, \dots, n), (\kappa = 1, \dots, K)$ vardır.

Bu noktada kuralın tekrar hatırlatılması faydalı olacaktır. Lagrange denklemini oluşturmak için, $g_i \geq 0$ formundaki kısıtlara ait kısıt denklemleri pozitif Lagrange çarpanlarıyla çarpılır ve amaç fonksiyonundan çıkarılır. Özgün problem için Lagrange denklemini $L(w, b, t, u, l, \xi, \propto, \mu, \lambda, \gamma, \tilde{\beta}, \sigma, \theta, \zeta, \upsilon, \varkappa) = L^P$ ile gösterirsek aşağıda denklem (4.16)'da verilen biçimde ifade edilebilir:

$$L^P = \frac{1}{2}\mathrm{u}^2 + \mathrm{C}\sum_{i=1}^{n} \xi_i - \sum_{i=1}^{n} \propto_i \left[\left(\sum_{\kappa=1}^{K} \mathbf{w}_\kappa{}^T \Phi_\kappa(\mathbf{x}_i) + b\right) - 1 + \xi_i + Dz\right] \qquad \mathbf{(4.16)}$$

$$+ \sum_{i=1}^{n} \mu_i \left[\left(\sum_{\kappa=1}^{K} \mathbf{w}_\kappa{}^T \Phi_\kappa(\mathbf{x}_i) + b\right) - \xi_i + 1 - D\,(1 - z)\right]$$

$$-\lambda \left[\sum_{i=1}^{n} \left(\sum_{\kappa=1}^{K} \mathbf{w}_\kappa{}^T \Phi_\kappa(\mathbf{x}_i) + b\right) + l\right]$$

$$+\gamma \left[\sum_{i=1}^{n} \left(\sum_{\kappa=1}^{K} \mathbf{w}_\kappa{}^T \Phi_\kappa(\mathbf{x}_i) + b\right) - l\right] - \sum_{i=1}^{n} \tilde{\beta}_i\, \xi_i - \sigma \left(u - \sum_{\kappa=1}^{K} t_\kappa\right)$$

$$- \sum_{\kappa=1}^{K} \theta_\kappa{}^T\, \mathrm{w}_\kappa - \sum_{\kappa=1}^{K} \zeta_\kappa t_\kappa - \upsilon(1 - z) - \varkappa z.$$

Özgün problem (CKO) artık bir konveks kuadratik programlama problemidir, çünkü amaç fonksiyonunun kendisi konvekstir ve kısıtları sağlayan bu noktalar da aynı zamanda konveks bir küme oluşturmaktadır (herhangi bir doğrusal kısıt konveks bir küme tanımlar ve N ardışık doğrusal kısıtlar kümesi, N konveks kümenin kesişimini tanımlar ki; bu da konveks bir kümedir). Bu problemimizin dual ile denk olarak çözülebileceği anlamına gelmektedir.

Lagrange fonksiyonu $(L^P)$'de temel değişkenler $w, b, l, t, u, \xi\ ve\ z$' ye göre türevlerini alıp sıfıra eşitleyelim:

$$\frac{\partial L}{\partial \mathbf{w}_\kappa} = 0,$$

$$-\sum_{i=1}^{n} \propto_i \Phi_\kappa(\mathbf{x}_i) + \sum_{i=1}^{n} \mu_i \Phi_\kappa(\mathbf{x}_i) - \lambda \sum_{i=1}^{n} \Phi_\kappa(\mathbf{x}_i) + \gamma \sum_{i=1}^{n} \Phi_\kappa(\mathbf{x}_i) - \Phi_\kappa = 0,$$

$$\Rightarrow \theta_\kappa = \sum_{i=1}^{n} \Phi_\kappa(\mathbf{x}_i)\,(-\propto_i + \mu_i - \lambda + \gamma), \tag{4.17}$$

$$\frac{\partial L}{\partial b} = 0,$$

$$-\sum_{i=1}^{n} \propto_i + \sum_{i=1}^{n} \mu_i - \sum_{i=1}^{n} \lambda + \sum_{i=1}^{n} \gamma = 0,$$

$$\sum_{i=1}^{n} \gamma - \sum_{i=1}^{n} \lambda = \sum_{i=1}^{n} \propto_i - \sum_{i=1}^{n} \mu_i\,,$$

$$n\gamma - n\lambda = \sum_{i=1}^{n} (\propto_i - \mu_i)\,,$$

$$\Rightarrow \sum_{i=1}^{n} (\propto_i - \mu_i) = n(\gamma - \lambda), \tag{4.18}$$

$$\frac{\partial L}{\partial l} = 0,$$

$$\Rightarrow \lambda - \gamma = 0,$$

$$\Rightarrow \gamma = -\lambda, \tag{4.19}$$

$$\frac{\partial L}{\partial t_\kappa} = 0,$$

$$\sigma - \zeta_\kappa = 0,$$

$$\Rightarrow \zeta_\kappa = \sigma, \qquad (\mathbf{4.20})$$

$$\frac{\partial L}{\partial u} = 0,$$

$$\frac{1}{2}.2u - \sigma = 0,$$

$$\Rightarrow u = \sigma, \qquad (\mathbf{4.21})$$

$$\frac{\partial L}{\partial \xi_i} = 0,$$

$$\Rightarrow C - \propto_i - \mu_i - \tilde{\beta}_i = 0,$$

$$\Rightarrow C = \propto_i + \mu_i + \tilde{\beta}_i, \qquad (\mathbf{4.22})$$

$$\frac{\partial L}{\partial z} = 0,$$

$$\Rightarrow -\sum_{i=1}^{n} \propto_i D + \sum_{i=1}^{n} \mu_i D + \upsilon - \varkappa = 0,$$

$$\Rightarrow \left(-\sum_{i=1}^{n} \propto_i + \sum_{i=1}^{n} \mu_i\right) D = -\upsilon + \varkappa . \qquad (\mathbf{4.23})$$

Yukarıda bulduklarımızı Lagrange denklemi $L^P$'de yerine yazarak, aşağıda problem $(CKO_D)$ ile verilen dual forma ulaşmak mümkündür ancak, önce $L^P$'de parantezleri açıp, benzer terimleri yanyana getirerek, $(4.17)-(4.23)$ aralığındaki eşitlikleri yerlerine yazalım:

$$L^P = \frac{1}{2}\mathrm{u}^2 + \mathrm{C}\sum_{i=1}^{n}\xi_i - \sum_{i=1}^{n}\propto_i \xi_i - \sum_{i=1}^{n}\mu_i\xi_i - \sum_{i=1}^{n}\tilde{\beta}_i\xi_i$$

$$-\sum_{i=1}^{n}\propto_i\left[\sum_{\kappa=1}^{K}\mathbf{w}_\kappa{}^T\Phi_\kappa(\mathbf{x}_i) + b\right] + \sum_{i=1}^{n}\propto_i - \sum_{i=1}^{n}\propto_i Dz$$

$$+\sum_{i=1}^{n}\mu_i\left(\sum_{\kappa=1}^{K}\mathbf{w}_\kappa{}^T\Phi_\kappa(\mathbf{x}_i) + b\right) + \sum_{i=1}^{n}\mu_i - \sum_{i=1}^{n}\mu_i\left(D(1-z)\right)$$

$$-\lambda\sum_{i=1}^{n}\left(\sum_{\kappa=1}^{K}\mathbf{w}_\kappa{}^T\Phi_\kappa(\mathbf{x}_i) + b\right) - \lambda\sum_{i=1}^{n} l - \gamma\sum_{i=1}^{n} l + \gamma\sum_{i=1}^{n}\left(\sum_{\kappa=1}^{K}\mathbf{w}_\kappa{}^T\Phi_\kappa(\mathbf{x}_i) + b\right)$$

$$-\sigma u + \sigma\sum_{\kappa=1}^{K} t_\kappa - \sum_{\kappa=1}^{K}\theta_\kappa{}^T\mathbf{w}_\kappa - \sum_{\kappa=1}^{K}\zeta_k t_\kappa - \upsilon(1-z) - \varkappa z.$$

Denklem $(4.17)-(4.23)$'den gelen eşitlikleri yerlerine yazalım:

$$= \frac{1}{2}\sigma^2 + \left(\propto_i + \mu_i + \tilde{\beta}_i\right)\sum_{i=1}^{n}\xi_i - \sum_{i=1}^{n}\left(\propto_i + \mu_i + \tilde{\beta}_i\right).\xi_i$$

$$-\sum_{i=1}^{n}\propto_i\left[\sum_{\kappa=1}^{K}\mathbf{w}_\kappa{}^T\Phi_\kappa(\mathbf{x}_i)\right] + \sum_{i=1}^{n}\mu_i\left[\sum_{\kappa=1}^{K}\mathbf{w}_\kappa{}^T\Phi_\kappa(\mathbf{x}_i)\right] - \lambda\sum_{i=1}^{n}\left[\sum_{\kappa=1}^{K}\mathbf{w}_\kappa{}^T\Phi_\kappa(\mathbf{x}_i)\right]$$

$$+\gamma\sum_{i=1}^{n}\left[\sum_{\kappa=1}^{K}\mathbf{w}_\kappa{}^T\Phi_\kappa(\mathbf{x}_i)\right] - \sum_{i=1}^{n}\propto_i b + \sum_{i=1}^{n}\mu_i.b - \lambda\sum_{i=1}^{n} b + \gamma\sum_{i=1}^{n} b + \sum_{i=1}^{n}\propto_i$$

$$+\sum_{i=1}^{n}\mu_i - \lambda\sum_{i=1}^{n} l - \gamma\sum_{i=1}^{n} l - \sigma.\sigma + \sigma\sum_{\kappa=1}^{K} t_\kappa - \sum_{\kappa=1}^{K}\theta_\kappa{}^T\mathbf{w}_\kappa - \sum_{\kappa=1}^{K}\sigma t_\kappa$$

$$-\sum_{i=1}^{n} \propto_i Dz + \sum_{i=1}^{n} \mu_i Dz + \upsilon z - \varkappa z - \sum_{i=1}^{n} \mu_i D - \upsilon.$$

Benzer terimler yanyana gelecek şekilde tekrar düzenlersek:

$$= \frac{1}{2}\sigma^2 - \sigma^2 + \sum_{i=1}^{n}\sum_{\kappa=1}^{K} \mathbf{w}_\kappa{}^T \Phi_\kappa(\mathbf{x}_i)\,[-\propto_i + \mu_i - \lambda + \gamma] - \sum_{\kappa=1}^{K} \theta_\kappa{}^T \mathbf{w}_\kappa$$

$$+b\left[-\sum_{i=1}^{n} \propto_i + \sum_{i=1}^{n} \mu_i - n\lambda + n\gamma\right] + \sum_{i=1}^{n} \propto_i + \sum_{i=1}^{n} \mu_i - \lambda nl - \gamma nl$$

$$-\sum_{i=1}^{n} \mu_i D - \upsilon$$

bulunmuş olur. Son olarak denklem (4.17) ve (4.19) kullanılarak aşağıdaki forma ulaşılır:

$$L^P = -\frac{1}{2}\sigma^2 + \sum_{\kappa=1}^{K} \mathbf{w}_\kappa{}^T \underbrace{\sum_{i=1}^{n} \Phi_\kappa(\mathbf{x}_i)\ (-\propto_i + \mu_i - \lambda + \gamma)}_{\theta_\kappa} - \sum_{\kappa=1}^{K} \theta_\kappa \mathbf{w}_\kappa{}^T$$

$$+\sum_{i=1}^{n} \propto_i + \sum_{i=1}^{n} \mu_i - \sum_{i=1}^{n} \mu_i D + b\left[-\sum_{i=1}^{n} \propto_i + \sum_{i=1}^{n} \mu_i + \sum_{i=1}^{n} \propto_i - \sum_{i=1}^{n} \mu_i\right] - \lambda n$$

$$-(-\lambda)nl - \upsilon.$$

Burada da uygun sadeleştirmeler yapılırsa aşağıdaki denklem ($L^P$) elde edilir:

$$L^P = -\frac{1}{2}\sigma^2 + \sum_{i=1}^{n} \propto_i + \sum_{i=1}^{n} \mu_i - \sum_{i=1}^{n} \mu_i D - \upsilon.$$

Lagrange denklemi ($L^P$)'de $\sum_{i=1}^{n} \mu_i$ parantezine alındığında ($L^P$)'nin denklem (4.24) ile verilen formu bulunmuş olur:

$$L^P = -\frac{1}{2}\sigma^2 + \sum_{i=1}^{n} \propto_i + \sum_{i=1}^{n} \mu_i(1-D) - \upsilon. \qquad (\mathbf{4.24})$$

Öbekleme problemi $(CP_E)$'de (1) numaralı $\|w_\kappa\| \le t_k$ eşitsizliği elde edilmişti. Lagrange denklemi $L^P$'de $w_\kappa$ ve $t_\kappa$ vektörlerinin Lagrange çarpanları $\theta_\kappa$ ve $\xi_\kappa$ için, denklem (4.17)'deki eşitlik kullanılarak $\|\theta_\kappa\|_2 \le \xi_\kappa$ eşitsizliği yazılabilir.

Denklem (4.17) ve (4.20) de dikkate alınarak ikinci dereceden koni kısıtlamaları:

$$\left\| \sum_{i=1}^{n} \Phi_\kappa(\mathbf{x}_i)\ (-\propto_i + \mu_i - \lambda + \gamma) \right\|_2 \le \sigma \qquad (\mathbf{4.25})$$

$\forall\ \kappa = 1, \dots\dots, K$ için,

bulunur ki, bu da problemi aşağıda belirtildiği gibi yazma kolaylığı sağlar:

$$\max \quad -\frac{1}{2}\sigma^2 + \sum_{i=1}^{n} \propto_i + \sum_{i=1}^{n} \mu_i\,(1-\mathrm{D}) - \upsilon$$

öyle ki $\sigma, \upsilon \in R$, $\quad \propto, \mu \in R^n$,

kısıtlar $\sigma, \upsilon \ge 0$, $\lambda > 0$ ve $0 \le \propto \le C.1 \quad 0 \le \mu \le C.1$.

Denklem (4.22) yardımı ile $C = \propto_i + \mu_i + \tilde{\beta}_i$ ve Lagrange çarpanlarının negatif olmamasından yararlanılarak, $C - \propto_i - \mu_i = \tilde{\beta}_i \ge 0$ yazılabilir. O halde, $C - \propto_i - \mu_i \ge 0$ ve $C \ge \propto_i + \mu_i$'dir. Öyle ki, $0 \le \propto \le C.1$, $0 \le \mu \le C.1$ yazılabilir.

Yukarıda kısıt (4.25) ile verilen eşitsizlikte normu iç çarpım olarak yazarsak

$$\left( \left\langle \sum_{i=1}^{n} \Phi_\kappa(\mathbf{x}_i)\ (-\propto_i + \mu_i - \lambda - \lambda), \sum_{j=1}^{n} \Phi_\kappa(\mathbf{x}_j)\ \left(-\propto_j + \mu_j - \lambda - \lambda\right) \right\rangle \right)^{1/2} \le \sigma,$$

$$\left( \sum_{i,j=1}^{n} (-\alpha_i + \mu_i - 2\lambda) \left(-\alpha_j + \mu_j - 2\lambda\right) \langle \Phi_\kappa(\mathbf{x}_i), \Phi_\kappa\left(\mathbf{x}_j\right) \rangle \right)^{1/2} \leq \sigma,$$

$$\left( \sum_{i,j=1}^{n} (-\alpha_i + \mu_i - 2\lambda) \left(-\alpha_j + \mu_j - 2\lambda\right) K_\kappa\left(\mathbf{x}_i, \mathbf{x}_j\right) \right)^{1/2} \leq \sigma \qquad (\mathbf{4.26})$$

elde edilir. Eşitsizlik (4.26)'da her iki tarafın karesi alınıp $\frac{1}{2}$ ile çarpılırsa

$$\sum_{i,j=1}^{n} (-\alpha_i + \mu_i - 2\lambda) \left(-\alpha_j + \mu_j - 2\lambda\right) K_\kappa\left(\mathbf{x}_i, \mathbf{x}_j\right) \leq \sigma^2,$$

$$\Rightarrow \frac{1}{2} \sum_{i,j=1}^{n} (-\alpha_i + \mu_i - 2\lambda) \left(-\alpha_j + \mu_j - 2\lambda\right) K_\kappa\left(\mathbf{x}_i, \mathbf{x}_j\right) \leq \frac{1}{2}\sigma^2$$

eşitsizliği yazılabilir. Dolayısıyla, $\frac{1}{2}\sigma^2 \rightarrow \sigma$ dönüşümünden, Özgün (*primal*) problemin dual formu, problem $(CKO_A)$'da verildiği gibi ifade edilir:

***Dual Problem (Conic Dual)***

$$\min_{\substack{\sigma,\upsilon \in R \\ \alpha,\mu \in R^n}} \sigma - \sum_{i=1}^{n} \alpha_i - \sum_{i=1}^{n} \mu_i (1 - \mathrm{D}) + \upsilon \qquad (\boldsymbol{CKO_A})$$

kısıtlar $\sigma, \upsilon \geq 0, \ \lambda > 0, \ 0 \leq \alpha \leq C.1, \ 0 \leq \mu \leq C.1,$

$$\frac{1}{2} \sum_{i,j=1}^{n} (-\alpha_i + \mu_i - 2\lambda) \left(-\alpha_j + \mu_j - 2\lambda\right) K_\kappa\left(\mathbf{x}_i, \mathbf{x}_j\right) \leq \sigma \quad \forall \kappa = 1, .., K.$$

Hatırlayalım; denklem (4.2)'de, $K_\kappa\left(\mathbf{x}_i, \mathbf{x}_j\right) = \langle \Phi_\kappa(\mathbf{x}_i), \Phi_\kappa\left(\mathbf{x}_j\right) \rangle$ idi. Her bir kernel için problem $(\mathrm{CKO}_A)$'da *ikinci dereceden kısıt* bulunmaktadır. Yukarıdaki problem $\kappa = 1$ durumunda orjinal (tek kernelli) DVM dual formudur. Bir optimizasyon probleminin dual formunu bulabilmenin birden çok yolu vardır. Bu tezde özgün problem (CKO), ikinci dereceden konik program kullanılarak dual form ($\text{CKO}_A$) haline getirilmiştir. Dual formun *KKT en iyilik koşulları* aşağıda sırası ile incelenmiştir. Karush – Kuhn – Tucker (KKT) en iyilik koşulları, aşağıdaki $(4.27) - (4.30)$'da belirtilen tümler gevşeklik eşitliklerini verir:

$$a)\quad \left.\begin{array}{c} \propto_i \left[\left(\sum_{\kappa=1}^{K} \mathbf{w}_\kappa{}^T \Phi_\kappa(\mathbf{x}_i) + b\right) - 1 + \xi_i + Dy\right] = 0, \\ \mu_i \left[\left(\sum_{\kappa=1}^{K} \mathbf{w}_\kappa{}^T \Phi_\kappa(\mathbf{x}_i) + b\right) - \xi_i + 1 - \mathrm{D}(1-y)\right] = 0, \\ \lambda \left[\left(\sum_{\kappa=1}^{K} \mathbf{w}_\kappa{}^T \Phi_\kappa(\mathbf{x}_i) + b\right) + l\right] = 0, \\ \gamma \left[\left(\sum_{\kappa=1}^{K} \mathbf{w}_\kappa{}^T \Phi_\kappa(\mathbf{x}_i) + b\right) + l\right] = 0, \\ \upsilon\,(1-y) = 0, \\ \varkappa y = 0, \end{array}\right\} \forall\, i = 1,..,n \quad (\mathbf{4.27})$$

$$b)\ (C - \propto_i - \ \mu_i).\ \xi_i = 0,\ \ \forall\, i = 1, \dots\dots, n\,,\ \ 0\ \leq \propto \leq C,\ \ 0\ \leq \mu\ \leq C, \quad (\mathbf{4.28})$$

$$c)\left(\frac{\mathbf{w}_\kappa}{\|\mathbf{w}_\kappa\|_2}\right)^T \left(\frac{-\sum_{i=1}^{n} \Phi_\kappa(\mathbf{x}_i)\ (-\propto_i + \ \mu_i - \ 2\lambda)}{\sigma}\right) = 0, (\kappa = 1,..,K), \quad (\mathbf{4.29})$$

$$d)\ \ \sigma\left(\sum_{\kappa=1}^{K} t_\kappa - \sigma\right) = 0. \quad (\mathbf{4.30})$$

Eşitlikler (4.27) ve (4.28), veri noktalarını, $I_0 = \{i,\ \propto_i = 0\}$, $I_M = \{i,\ \propto_i \in (0, \mathrm{C})\}$, $I_C = \{i, \propto_i = \mathrm{C}\}$ olmak üzere üç ayrık kümeye ayırmaktadır.

$K_2 = \{w = (u, v) \in R^2 \mathrm{x}\, R,\ \ \|u\|_2 \leq v\}$ olmak üzere, ikinci dereceden koniğin, sıfırdan farklı $(u, v)$ ve $(u', v')$ iki elemanı ortogonaldir ancak ve ancak her ikisi de sınıra ait ve *ters orantılı* ise (Lobo ve diğ. 1998). O halde, $\exists\, \eta > 0$ öyle ki, $\|u\|_2 = v$, $\|u'\|_2 = v'$ ve $(u, v) = \eta\left(-u', v'\right)$ eşitlikleri sağlanır. Bu bilgilere göre,

eğer $\sigma > 0$ ise:

**i.** $$\left\|\sum_{i=1}^{n} \Phi_\kappa(\mathbf{x}_i)\ (-\propto_i + \ \mu_i -\ 2\lambda)\right\|_2 \ \le \sigma \ \Rightarrow \ \mathbf{w}_\kappa = 0,$$

**ii.** $$\left\|\sum_{i=1}^{n} \Phi_\kappa(\mathbf{x}_i)\ (-\propto_i + \ \mu_i -\ 2\lambda)\right\|_2 \ = \sigma$$ öyleyse $\exists\, \eta_\kappa > 0$ öyle ki,

$$\boldsymbol{w}_\kappa = \eta_\kappa \sum_{i=1}^{n} \Phi_\kappa(\mathbf{x}_i)\,(-\propto_i + \ \mu_i -\ 2\lambda),$$

$$\|\mathbf{w}_\kappa\|_2 = \eta_\kappa \sigma, \qquad (\mathbf{4.31})$$

bağıntıları vardır.

Böylece optimizasyon probleminin seyrek çözümü olduğu görülür. Aktif küme $\mathcal{J}$

$$\mathcal{J}\,(\propto, \mu, \lambda, \sigma) = \left\{\kappa\colon \left\|\sum_{i=1}^{n} \Phi_\kappa(\mathbf{x}_i)\ (-\propto_i + \ \mu_i -\ 2\lambda)\right\|_2 \ = \sigma\right\}$$

ile tanımlansın. En iyilik koşullarını tekrar yazacak olursak denklem $(4.30)$'da, $\kappa \notin \mathcal{J}$ ise ,

$$\forall\, \kappa, \qquad \mathbf{w}_\kappa = \eta_\kappa \sum_{i=1}^{n} \Phi_\kappa(\mathbf{x}_i)\,(-\propto_i + \ \mu_i -\ 2\lambda) \ \text{ ile } \ \eta_\kappa = 0, \qquad (\mathbf{4.32})$$

bulunur. Dolayısıyla, denklem $(4.31)$ kullanılarak,

$$\sum_{\kappa} \|\mathbf{w}_\kappa\|_2 = \sum_{\kappa} \eta_\kappa \sigma,$$

$$\Rightarrow \sum_{\kappa \in \mathcal{J}} \eta_\kappa = 1$$

elde edilir. Elde edilen eşitlik $(CP)$ problemindeki $\sum_{\kappa=1}^{K} \beta_\kappa = 1$'e karşılık gelir. O halde, denklem (4.32)'de $\mathbf{w}_\kappa$ nın dual formu:

$$\mathbf{w}_\kappa = \boldsymbol{\beta}_\kappa \sum_{i=1}^{n} \Phi_\kappa(\mathbf{x}_i)\,(-\propto_i + \ \mu_i - 2\lambda) \qquad \mathbf{(4.33)}$$

şeklinde yazılabilir.

Dual problem $(CKO_A)$'daki amaç fonksiyonunda $\sigma - \sum_{i=1}^{n} \propto_i - \sum_{i=1}^{n} \mu_i(1-\mathrm{D})$ terimleri yerine $\sigma$ yazılarak problem $(CKO_A)$'nın kısıtlarının ker iki tarafına da $-\sum_{i=1}^{n} \propto_i - \sum_{i=1}^{n} \mu_i(1-\mathrm{D})$ eşdeğerini eklenirse aşağıda $(CKO_B)$ ile verilen problem elde edilir:

$$\min_{\substack{\sigma \in R, \\ \lambda, \upsilon \in R^+, \\ \propto, \mu \in R^n}} \quad \sigma + \upsilon \qquad (\boldsymbol{CKO_B})$$

kısıtlar $\quad \sigma, \upsilon \ \geq 0 \ \ , \ \ 0 \ \leq \propto \leq C.1, \ \ 0 \ \leq \mu \ \leq C.1,$

$$\frac{1}{2} \sum_{i,j=1}^{n} (-\propto_i + \ \mu_i - 2\lambda)\left(-\propto_j + \ \mu_j - 2\lambda\right) K_\kappa\left(\mathbf{x}_i\,, \mathbf{x}_j\right) - \sum_{i=1}^{n} \propto_i - \sum_{i=1}^{n} \mu_i(1-\mathrm{D}) \leq \sigma$$

$\forall\, \kappa = 1, \ldots\ldots, K.$

İfade kolaylığı açısından son kısıtı aşağıdaki şekilde ifade edelim:

$$S_\kappa(\propto, \mu, \lambda\,) = \frac{1}{2} \sum_{i,j=1}^{n} (-\propto_i + \ \mu_i - 2\lambda)\left(-\propto_j + \ \mu_j - 2\lambda\right) K_\kappa\left(\mathbf{x}_i\,, \mathbf{x}_j\right)$$

$$-\sum_{i=1}^{n} \propto_i - \sum_{i=1}^{n} \mu_i(1-\mathrm{D}) \ \leq \sigma.$$

Problem $(CKO_B)$'de amaç fonksiyonu yerine

$$= \ \sigma + \ \upsilon + \sum_{\kappa=1}^{K} \beta_\kappa S_\kappa(\propto, \mu, \lambda\,) - \ \sigma$$

$$= \sum_{\kappa=1}^{K} \beta_\kappa S_\kappa(\propto, \mu, \lambda\,) + \ \upsilon$$

ifadesi yazılırsa aşağıdaki $(CKO_C)$ problemini çözmek eşdeğerdir:

$$\min_{\substack{\lambda, \upsilon\, \in R^+ \\ \propto, \mu\, \in R^n}} \quad \sum_{\kappa=1}^{K} \beta_\kappa S_\kappa(\propto, \mu, \lambda\,) + \upsilon \qquad (\boldsymbol{CKO_C})$$

kısıtlar $\lambda\ > 0,\ 0\ \leq \propto \leq C.1,\ \ 0\ \leq \mu\ \leq \text{C}.1\,,\ \ 1 =\ (1,1,1,\ldots,1)^T \in\ R^n,$

$$\sum_{\kappa=1}^{K} \beta_\kappa = 1.$$

Problemin gösterimde kolaylık olması için $S\,(\propto, \mu, \lambda, \beta, \upsilon) := \sum_{\kappa=1}^{K} \beta_\kappa S_\kappa(\propto, \mu, \lambda) + \upsilon$ ile tanımlayalım. Sonrasında bu problemi $\propto, \mu, \lambda, \upsilon$ değerleri için minimum, $\boldsymbol{\beta}$ değeri için maksimum yapacak aşağıdaki forma ulaşılmış olur:

***Min – Max Problem***

$$\max_{\beta \in R_+^K} \quad \min_{\substack{\propto, \mu \in R_+^n \\ \lambda, \upsilon\, \in R^+}} \quad \sum_{\kappa=1}^{K} \beta_\kappa S_\kappa(\propto, \mu, \lambda\,) +\ \upsilon \qquad (\boldsymbol{CKO_D})$$

kısıtlar $\quad 0\ \leq \propto \leq C.1, \quad 0\ \leq \mu\ \leq C.1, \quad 0\ \leq \beta$ ve $\displaystyle\sum_{\kappa=1}^{K} \beta_\kappa = 1.$

Farz edelim ki, $\propto^*,\ \mu^*,\ \lambda^*,\ \upsilon^*$ ideal çözüm olsun. O halde $\Theta^* :=\ S\,(\propto^*, \mu^*, \lambda^*, \upsilon^*, \beta)$ ile tanımlanabilir. Bu tanımdan yararlanarak Lineer Yarı – Sonsuz Programlama (bknz Bölüm 2.2.1) problemi yazılabilir.

***Lineer Yarı – Sonsuz Program (LYSP) – (Linear Semi – Infinite Program - LSIP)***

$$\max_{\substack{\Theta\in \mathrm{R} \\ \boldsymbol{\beta}\in R^K}} \quad \Theta \qquad (\boldsymbol{CKO_E})$$

kısıtlar $\quad 0 \leq \boldsymbol{\beta}, \quad \sum_{\kappa=1}^{K}\beta_\kappa = 1,$

$$\sum_{\kappa=1}^{K}\beta_\kappa S_\kappa(\propto,\mu,\lambda) + \upsilon \geq \Theta \quad \forall \propto,\mu, \quad \forall\lambda,\upsilon \in R^+,$$

$$0 \leq \propto \leq C.1 \;,\; 0 \leq \mu \leq C.1.$$

Yukarıda problem $(CKO_E)$ ile verilen maximum problemi minimizasyon problemine dönüştürüldüğünde denklem $(CKO_F)$ halini alır:

$$\min_{\substack{\Theta\in \mathrm{R} \\ \beta\in R^K}} \quad -\Theta \qquad (\boldsymbol{CKO_F})$$

kısıtlar $\quad 0 \leq \beta, \quad \sum_{\kappa=1}^{K}\beta_\kappa = 1,$

$$\sum_{\kappa=1}^{K}\beta_\kappa S_\kappa(\propto,\mu,\lambda) - \Theta + \upsilon \geq 0 \;\forall \propto,\mu, \quad \forall\lambda,\upsilon \in R^+,$$

$$0 \leq \propto \leq C.1, \quad 0 \leq \mu \leq C.1.$$

Buradan hareketle öbeklere ayıracak (iki öbek) en uygun hiperdüzlem denklem (4.34) ile yazılabilir:

$$f(\mathbf{x}) = \left(\sum_{\kappa=1}^{K}\langle \mathbf{w}_\kappa, \Phi_\kappa(\mathbf{x}) \rangle + b\right). \qquad (\mathbf{4.34})$$

Denklem (4.33)'de $\mathbf{w}_\kappa$vektörünün dual formu yerine yazılacak olursa,

$$f(\mathbf{x}) = \left(\sum_{\kappa=1}^{K}\sum_{i=1}^{n}\beta_\kappa \langle \Phi_\kappa(\mathbf{x}_i), \Phi_\kappa(\mathbf{x}) \rangle \; (-\alpha_i + \mu_i - 2\lambda) + b\right),$$

$$= \left( \sum_{i=1}^{n} \sum_{\kappa=1}^{K} \beta_\kappa \;\; K_\kappa(\mathbf{x}_i, \mathbf{x}) \;\; (-\,\alpha_i + \;\mu_i - 2\lambda) + b \right),$$

$$= \left( \sum_{i=1}^{n} K(\mathbf{x}_i, \mathbf{x}) \;\; (-\,\alpha_i + \;\mu_i - 2\lambda) + b \right),$$

$$f(\mathbf{x}) \quad = \left( \sum_{i=1}^{n} K(\mathbf{x}_i, \mathbf{x}) \;\; (-\,\alpha_i + \;\mu_i - 2\lambda) + b \right)$$

elde edilir.

Verilen **x** girdi noktasının öbeğini belirlemek için $f(\mathbf{x})$ fonksiyonunun işaretine bakılır:

$$y = sign \left( \sum_{i=1}^{n} K(\mathbf{x}_i, \mathbf{x}) \;\; (-\,\alpha_i + \;\mu_i - 2\lambda) + b \right).$$

Bulunan değerin pozitif ya da negatif olmasına göre, hiperdüzlemin hangi tarafında olacağına karar verilir.

Lineer Yarı – sonsuz Programlama problemi, problem $(CKO_F)$ ile minimize edilmişti. Bölüm 2.3.5'de detayları verilen düşük düzeyli problemi yazmak için, verilen $(\Theta, \beta)$ parametresi ile $g$ fonksiyonu yazılırsa:

$$g\left((\Theta, \beta)\,, \propto, \mu, \lambda, \upsilon\,\right) \coloneqq \sum_{\kappa=1}^{K} \beta_\kappa S_\kappa(\propto, \mu, \lambda\,) - \Theta + \upsilon$$

ifadesi ile tanımlanabilir. Varsayalım ki, $(\Theta, \beta)$ verilmiş olsun. O halde, düşük düzeyli özgün problem $(\mathrm{CKO_G})$ aşağıdaki gibi yazılabilir.

***Düşük Düzeyli Özgün Problem (Lower Level Primal Problem)***

$$\min_{\substack{\propto,\mu \in R^n \\ \lambda,\upsilon \in R^+}} \; g\left((\Theta,\beta),\propto,\mu,\lambda,\upsilon\right) \qquad (\boldsymbol{CKO_G})$$

kısıtlar $\propto,\mu,\lambda,\upsilon \in A$,

$$A = \{\propto,\mu \in R^n,\ \lambda,\upsilon \in R \mid 0 \le \propto \le C.1\,,\ 0 \le \mu \le C.1\,,\ 0 \le \lambda,\upsilon\}.$$

Dikkat edilirse, $A$ kümesinin bu hali tıkız değildir. O halde $A$ kümesini tıkız yapacak yeterince büyük Γ sayısı alalım, öyle ki $0 \le \lambda,\upsilon \le \Gamma$ olsun. O halde artık $A$ kümesi tıkızdır. Bölüm 2.3.5'de olduğu gibi *A* sonsuz indeks kümesini tanımlayan fonksiyonları yazalım:

$$\upsilon_r^1\left((\Theta,\beta),\propto,\mu\right) = \alpha_r \ \ge 0 \quad \upsilon_s^1\left((\Theta,\beta),\propto,\mu\right) = -\alpha_{4n-s} + C \ge 0, \qquad (\boldsymbol{A_1})$$

$$\upsilon_r^2\left((\Theta,\beta),\propto,\mu\right) = \ \mu_r \ \ge 0 \quad \upsilon_s^2\left((\Theta,\beta),\propto,\mu\right) = -\mu_{4n-s} + C \ge 0, \qquad (\boldsymbol{A_2})$$

$$\upsilon^2(\lambda) = \lambda, \qquad (\boldsymbol{A_3})$$

$$\upsilon^3(\upsilon) = \upsilon, \qquad (\boldsymbol{A_4})$$

$$(r \in \{1,2,\ldots,2n\},\ \ s \in \{2n+1, 2n+2, \ldots, 4n\}).$$

**Not:** $\upsilon_r^1$, $\upsilon_s^1$, $\upsilon^2$, $\upsilon^3$ gösterimlerindeki 1, 2 ve 3 üstel değildir, sadece notasyon amaçlı kullanılmıştır.

Yukarıda $(A_1)$ ve $(A_2)$ ile tanımlanan fonksiyonlar kullanım kolaylığı açısından aşağıda belirtildiği gibi yazılabilir:

$$\upsilon_r^1\left((\Theta,\beta),\propto,\mu\right) =: \upsilon_r^1(\propto,\mu) \ge 0, \qquad \upsilon_s^1\left((\Theta,\beta),\propto,\mu\right) =: \upsilon_s^1(\propto,\mu) \ge 0,$$

$$\upsilon_r^2\left((\Theta,\beta),\propto,\mu\right) =: \upsilon_r^2(\propto,\mu) \ge 0, \qquad \upsilon_s^2\left((\Theta,\beta),\propto,\mu\right) =: \upsilon_s^2(\propto,\mu) \ge 0.$$

Gösterimde kolaylık olması bakımından $\tilde{\alpha} := \begin{bmatrix} \propto \\ \mu \end{bmatrix}$ ile tanımlanacak olursa sonsuz indeks kümesi $A$' nın fonksiyonları aşağıda verildiği üzere denklem$(B)$'ye indirgenmiş olur:

$$v_r^1\big((\Theta,\beta)\,,\tilde{\alpha}\,\big) = \tilde{\alpha}_r \ \geq 0, \quad v_s^1\big((\Theta,\beta)\,,\tilde{\alpha}\,\big) = -\,\tilde{\alpha}_{4n-s} + C \ \geq 0. \qquad (\boldsymbol{B})$$

Anlatımda kolaylık açısından denklem $(B)$ ile gösterilen fonksiyonları

$$v_r^1\big((\Theta,\beta)\,,\tilde{\alpha}\,\big) =: v_r(\tilde{\alpha}\,)\ ,\ v_s^1\big((\Theta,\beta)\,,\tilde{\alpha}\,\big) =: v_s(\tilde{\alpha}\,)$$

biçiminde yazarak basitleştirelim. O halde, sonsuz indeks kümesi

$$A = \{\propto, \mu\ \in R^n,\ \ \lambda, \upsilon \in R \mid v_r(\tilde{\alpha}\,) \geq\ 0,\ \ v_s(\tilde{\alpha}\,) \geq\ 0,\ v^2(\,\lambda) \geq\ 0,\ v^3(\upsilon) \geq\ 0\}$$

şeklinde ifade edilir.

Aktif indislerin kümesi $L_0\big(\bar{\tilde{\alpha}},\ \ \bar{\lambda},\ \ \bar{\upsilon}\big)$ olmak üzere,

$$L_0\big(\bar{\tilde{\alpha}},\ \ \bar{\lambda},\ \ \bar{\upsilon}\big) = \big\{l\ \in L\ \mid\ v_l^1(\bar{\tilde{\alpha}}) = 0,\ \ v^2\big(\bar{\lambda}\,\big) = 0,\ \ v^3(\,\bar{\upsilon}) = 0\big\}\ (L =\ \{1, 2, \dots\ ,\ 4n\})$$

ile tanımlansın. Burada, $\bar{\tilde{\alpha}}\ , \bar{\lambda}\ ve\ \bar{\upsilon}$, aktif yapan noktalar olsun. O halde özgün (*primal*) problem $(CKO_G)$'nin Lagrange fonksiyonu,

$$\mathcal{L}^P(\Theta,\beta;\tilde{\alpha},\tilde{\gamma},\lambda,\upsilon\,) := g\,\big((\Theta,\beta),\tilde{\alpha},\lambda,\upsilon\,\big) - \sum_{l\ \in L_0(\bar{\tilde{\alpha}},\bar{\lambda},\bar{\upsilon})} \tilde{\gamma_l}^1\ v_l^1(\tilde{\alpha})$$

$$-\tilde{\gamma_l}^2\ v^2\big(\,\bar{\lambda}\,\big) - \tilde{\gamma_l}^3 v^3(\,\bar{\upsilon}\,)$$

biçiminde yazılabilir. Lagrange fonksiyonu $\mathcal{L}^P(\tilde{\alpha},\lambda,\gamma,\upsilon) := \mathcal{L}^P(\Theta,\beta;\ \tilde{\alpha},\lambda,\gamma,\upsilon)$ ile tanımlansın.

$A$ kümesi tıkız (kompakt) küme ve $g$ sürekli fonksiyon ($g\ \in C^2$) olduğundan $(\Theta,\beta)$ için $(CKO_G)$ probleminin yerel minimumu vardır. Her $(\Theta,\beta)$ ve her bir $(\bar{\tilde{\alpha}} \in A)$ adayı için koşulları inceleyelim:

***1. Lineer Bağımsız Kısıt Koşulu (LBKK):***

***(Linear Independence Constraint Qualification - LICQ)***

Öncelikle aktif eşitsizlik kısıtlarının $\nabla v_r(\tilde{\alpha}\,)$ ile $\nabla v_s(\tilde{\alpha}\,)$'nin $(r \in \{1,2,\dots,2n\},\ s \in \{2n+1, 2n+2\,,\dots,4n\})$ lineer bağımsız olup olmadıkları kontrol edilecektir.

Öyle ki $\tilde{\alpha}$, $v_r(\tilde{\alpha}\,) = \tilde{\alpha}_r$ veya $v_s(\tilde{\alpha}\,) = -\tilde{\alpha}_{4n-s} + C$ eşitliğini sağlar ve Jakobyenleri $\nabla v_r(\bar{\tilde{\alpha}}\,) = (0,0,\dots,1,0,\dots,0)^T$, $\nabla\, v_s(\bar{\tilde{\alpha}}) = (0,0,\dots,-1,0,\dots,0)^T$'dir. Aktif eşitsizliklerin Jacobyen matrisleri aşağıda denklem (4.35) ile hesaplanabilir. Kolaylık açısından tüm aktif kısıtlar $A(\tilde{\alpha})$ vektörü ile tanımlanacak olursa

$$A(\tilde{\alpha},\lambda,v) = \begin{bmatrix} v_{l_1}(\tilde{\alpha}\,) \\ v_{l_2}(\tilde{\alpha}\,) \\ \vdots \\ v_{l_k}(\tilde{\alpha}\,) \\ v^2(\lambda) \\ v^3(v) \end{bmatrix} \qquad \mathbf{(4.35)}$$

matrisi elde edilir öyle ki $L_0\big(\bar{\tilde{\alpha}},\ \bar{\lambda},\ \bar{v}\big) = \{l_1\,, l_2, \dots, l_k\}$ ve aktif kümelerin eleman sayısı $|L_0(\bar{\tilde{\alpha}})| = k$'dır. Dikkat edilecek olunursa, Jacobyen matrisi $A(\tilde{\alpha})\ k$ x $(2n+2)$ lik bir matristir.

Jacobyen matris, denklem (4.36)'da, $k$ satır sayısını, ilk $n$ tane sütun $\propto$'ya göre türevi, $n+1$'den $2n$'e kadar olan sütun $\mu$'ye göre türevi, $2n+1$.ci sütun $\lambda$' ya göre türevi ve son sütun ise $v$'ye göre türevi göstermektedir:

$$DA(\tilde{\alpha}\,,\lambda,v) = \begin{bmatrix} 1 & 0 & \cdots & 0 & 1 & 0 & \cdots & 1 & 1 \\ 0 & 0 & 1 & \cdots & 1 & \cdots & 0 & 1 & 1 \\ 0 & 1 & \cdots & 0 & 1 & 0 & \cdots & 1 & 1 \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ 0 & 0 & -1 & \cdots & 0 & -1 & & 1 & 1 \\ -1 & 0 & \cdots & 0 & -1 & \cdots & 0 & 1 & 1 \\ 0 & 0 & \cdots & \cdots & -1 & 0 & \cdots & 1 & 1 \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ 0 & 0 & 0 & \cdots & -1 & 0 & \cdots & 1 & 1 \end{bmatrix}_{k\,x\,(2n+2)} \qquad \mathbf{(4.36)}$$

Eğer, $2n+2 < k$ ise $rank\left(DA(\tilde{\alpha}, \lambda, \upsilon)\right) = 2n+2$ olur ki rank, satır sayısından küçük olduğu için lineer bağımsızlık koşulu sağlanamamış olur. Bu da LBKK'nın sağlanamadığını gösterir. Halbuki sonsuz indeks kümesinin fonksiyonları $\upsilon_r(\tilde{\alpha}) = \tilde{\alpha}_r$ veya $\upsilon_s(\tilde{\alpha}) = -\tilde{\alpha}_{4n-s} + C$ eşitliklerini sağlayacağı için $\upsilon^2(\lambda) = \lambda$ ve $\upsilon^3(\upsilon) = \upsilon$ de göz önünde bulundurulursa, *2n* tane $\upsilon_r(\tilde{\alpha})$'dan ya da maksimum *2n* tane $\upsilon_s(\tilde{\alpha})$'dan aktif eşitlik gelir. O halde aktif indeks kümesinin eşitsizlikleri maksimum *2n+2* tane noktada aktif olabilirler. O halde, $k$ maksimum *2n+2* değerini alacaktır ve LBKK her durumda sağlanacaktır.

***2. Karush Kuhn Tucker Koşulu ile Pozitif Lagrange Çarpanları (aktif eşitsizlikler için):***

Çarpan vektörü $\tilde{\gamma}_l \in R^{|L_0(\bar{\tilde{\alpha}},\ \bar{\lambda},\ \bar{\upsilon})|}$ denklem $(CKO_G)$ ile tanımlanan $g\left((\Theta,\beta), \tilde{\alpha}, \lambda, \upsilon\right)$ fonksiyonundaki Lagragne vektörleri kullanılarak aşağıdaki gibi yazılabilir:

$$\tilde{\gamma} := \left[\left(\tilde{\gamma}_l^{\,1}\right)_{l \in L_0(\bar{\tilde{\alpha}},\bar{\lambda},\bar{\upsilon})} \quad \left(\tilde{\gamma}_l^{\,2}\right)_{l \in L_0(\bar{\tilde{\alpha}},\bar{\lambda},\bar{\upsilon})} \quad \left(\tilde{\gamma}_l^{\,3}\right)_{l \in L_0(\bar{\tilde{\alpha}},\bar{\lambda},\bar{\upsilon})}\right]^T \qquad (\mathbf{4.37})$$

öyle ki,

$$\nabla_{(\tilde{\alpha},\lambda,\upsilon)} \mathcal{L}^P(\Theta,\beta;\ \tilde{\alpha},\lambda,\gamma,\upsilon)\tilde{\gamma}_l = 0 \ \text{ ve } \ \tilde{\gamma}_l > 0 \qquad \left(l \in L_0(\bar{\tilde{\alpha}},\bar{\lambda},\bar{\upsilon})\right)$$

biçimindedir.

Pozitif Lagrange çarpanları için tüm durumları inceleyelim:

*1. Durum:* Eğer, $L_0(\bar{\tilde{\alpha}},\bar{\lambda},\bar{\upsilon}) \neq \emptyset$ ise, $g\left((\Theta,\beta), \propto, \mu, \lambda, \upsilon\right)$ yi yeniden yazalım:

$$g\left((\Theta,\beta), \propto, \mu, \lambda, \upsilon\right) := \sum_{\kappa=1}^{K} \beta_\kappa S_\kappa(\propto, \mu, \lambda) - \Theta + \upsilon$$

$$= \sum_{\kappa=1}^{K} \beta_\kappa \left[\frac{1}{2} \sum_{i,j=1}^{n} (-\propto_i + \mu_i - 2\lambda)\left(-\propto_j + \mu_j - 2\lambda\right) K_\kappa\left(\mathbf{x}_i, \mathbf{x}_j\right)\right]$$

$$-\sum_{\kappa=1}^{K}\beta_\kappa\left[\sum_{i=1}^{n}\propto_i+\sum_{i=1}^{n}\mu_i(1-\mathrm{D})\right]-\Theta+\upsilon$$

$$=\frac{1}{2}\sum_{i,j=1}^{n}(-\propto_i+\ \mu_i-\ 2\lambda)\left(-\propto_j+\ \mu_j-2\lambda\right)\underbrace{\sum_{\kappa=1}^{K}\beta_\kappa K_\kappa\left(\mathbf{x}_i\,,\mathbf{x}_j\right)}_{=:K_{ij}}$$

$$-\sum_{\kappa=1}^{K}\sum_{i=1}^{n}\beta_\kappa\propto_i-\sum_{\kappa=1}^{K}\sum_{i=1}^{n}\beta_\kappa\mu_i(1-\mathrm{D})-\ \Theta+\upsilon$$

$$=\ \frac{1}{2}\sum_{i,j=1}^{n}(-\propto_i+\ \mu_i-\ 2\lambda)\left(-\propto_j+\ \mu_j-2\lambda\right)K_{ij}-\sum_{\kappa=1}^{K}\sum_{i=1}^{n}\beta_\kappa\propto_i$$

$$-\sum_{\kappa=1}^{K}\sum_{i=1}^{n}\beta_\kappa\mu_i+\sum_{\kappa=1}^{K}\sum_{i=1}^{n}\beta_\kappa\mu_i\,D-\ \Theta+\upsilon$$

şeklinde yazabileceğimiz $g\left((\Theta,\beta)\,,\propto,\mu,\lambda,\upsilon\right)$ fonksiyonunda $K_{ij}$, $\propto,\mu,\lambda,\upsilon,\Theta$'dan bağımsız ancak $\beta$'ya bağımlıdır. $\sum_{\kappa=1}^{K}\beta_\kappa=1$ özel durumunu göz önüne alarak Lagrange fonksiyonunu yazarsak denklem (4.38)'i elde ederiz:

$$\mathcal{L}_1^P(\Theta,\beta;\ \tilde{\alpha},\tilde{\gamma},\lambda,\upsilon)=$$

$$g\left((\Theta,\beta),\tilde{\alpha},\lambda,\upsilon\right)-\sum_{l\,\in L_0\left(\bar{\tilde{\alpha}},\bar{\lambda},\bar{\upsilon}\right)}\tilde{\gamma}_l^{\,1}\,v_l^1(\bar{\tilde{\alpha}})-\tilde{\gamma}_l^{\,2}\,v^2(\,\bar{\lambda}\,)-\tilde{\gamma}_l^{\,3}v^3(\,\bar{\upsilon}\,).\qquad(\mathbf{4.38})$$

Lagrange fonksiyonu denklem (4.38)'de $g\left((\Theta,\beta)\,,\propto,\mu,\lambda,\upsilon\right)$'yı yerine yazarak asağıdaki denklem (4.39)'u elde ederiz:

$$\mathcal{L}_1^P(\Theta,\beta;\ \tilde{\alpha},\tilde{\gamma},\lambda,\upsilon)=\ \frac{1}{2}\sum_{i,j=1}^{n}(-\propto_i+\ \mu_i-\ 2\lambda)\left(-\propto_j+\ \mu_j-2\lambda\right)K_{ij}-\sum_{\kappa=1}^{K}\sum_{i=1}^{n}\beta_\kappa\propto_i$$

$$- \sum_{\kappa=1}^{K}\sum_{i=1}^{n} \beta_\kappa \mu_i + \sum_{\kappa=1}^{K}\sum_{i=1}^{n} \beta_\kappa \mu_i D - \Theta + \upsilon$$

$$- \sum_{l \in L_0(\bar{\tilde{\alpha}},\bar{\lambda},\bar{\upsilon})} \tilde{\gamma}_l^{\,1}\; v_l^1(\bar{\tilde{\alpha}}) \; - \tilde{\gamma}^2\, v^2(\,\bar{\lambda}\,) - \tilde{\gamma}^3 v^3(\,\bar{\upsilon}\,)$$

$$= \frac{1}{2}\sum_{i=1}^{n} (-\propto_i + \mu_i - 2\lambda)^2\; K_{ii} + \frac{1}{2}\sum_{\substack{i,j=1 \\ i \neq j}}^{n} (-\propto_i + \mu_i - 2\lambda)\left(-\propto_j + \mu_j - 2\lambda\right) K_{ij}$$

$$- \sum_{\kappa=1}^{K}\sum_{i=1}^{n} \beta_\kappa \propto_i - \sum_{\kappa=1}^{K}\sum_{i=1}^{n} \beta_\kappa \mu_i + \sum_{\kappa=1}^{K}\sum_{i=1}^{n} \beta_\kappa \mu_i D - \Theta + \upsilon$$

$$- \sum_{l \in L_0(\bar{\tilde{\alpha}},\bar{\lambda},\bar{\upsilon})} \tilde{\gamma}_l^{\,1}\; v_l^1(\bar{\tilde{\alpha}}) \; - \tilde{\gamma}^2\, v^2(\,\bar{\lambda}\,) - \tilde{\gamma}^3 v^3(\,\bar{\upsilon}\,). \qquad \mathbf{(4.39)}$$

İşlemleri basitleştirebilmek için, eşitlik (4.39)'da Gauss Kernel fonksiyonu tercih edilmiştir. Diğer kernel fonksiyonları da uygulanabilir. Gauss kernel,

$$K(x_i\,, x_j, w) = \exp\left(-w\,\left\|x_i - x_j\right\|_2^2\right) \quad (*)$$

ile tanımlanır. Denklem $(*)$'da $i = j$ için $K(x_i\,, x_i, w) = 1'$ dir. [3]

Negatif olmayan Lagrange çarpanlarını bulmak için, $\nabla_{\tilde{\alpha}} \mathcal{L}_1^P(\Theta, \beta;\; \tilde{\alpha}, \tilde{\gamma}, \lambda, \upsilon) = 0$ lineer olmayan denklemin çözülmesi gerekir. Denklem (4.39) ile verilen Lagrange fonksiyonu aşağıdaki forma dönüşür:

$$\mathcal{L}_1^P(\Theta, \beta;\; \tilde{\alpha}, \tilde{\gamma}, \lambda, \upsilon) =$$

$$= \frac{1}{2}\sum_{i=1}^{n} (-\propto_i + \mu_i - 2\lambda)^2 .1 + \frac{1}{2}\sum_{\substack{i,j=1 \\ i \neq j}}^{n} (-\propto_i + \mu_i - 2\lambda)\left(-\propto_j + \mu_j - 2\lambda\right) K_{ij}$$

[3] $i = j$ için, $K_{ii}\;(\mathbf{x}_i\,, \mathbf{x}_i) = \sum_{\kappa=1}^{K} \beta_\kappa\; \underbrace{e^{-w\,\|x_i - x_j\|_2^2}}_{1} = \sum_{\kappa=1}^{K} \beta_\kappa = 1$'dir.

$$-\sum_{\kappa=1}^{K}\sum_{i=1}^{n}\beta_{\kappa}\propto_i-\sum_{\kappa=1}^{K}\sum_{i=1}^{n}\beta_{\kappa}\,\mu_i+\sum_{\kappa=1}^{K}\sum_{i=1}^{n}\beta_{\kappa}\mu_i\,D-\Theta+\upsilon$$

$$-\sum_{l\in L_0(\bar{\tilde{\alpha}},\bar{\lambda},\bar{\upsilon})}\tilde{\gamma}_l^{\,1}\;\upsilon_l^1(\bar{\tilde{\alpha}})\;-\tilde{\gamma}^2\,\upsilon^2(\,\bar{\lambda}\,)-\tilde{\gamma}^3\upsilon^3(\,\bar{\upsilon}\,). \qquad \mathbf{(4.40)}$$

Denklem (4.40)'da verilen Lagrange fonksiyonunun, Jakobyen matrisini sıfıra eşitleyelim:

$$\nabla_{(\tilde{\alpha},\lambda,\upsilon)}\mathcal{L}_1^P(\Theta,\beta;\;\tilde{\alpha},\tilde{\gamma},\lambda,\upsilon)\;=0$$

$$\nabla_{(\tilde{\alpha},\lambda,\upsilon)}\,\mathcal{L}_1^P(\Theta,\beta;\;\tilde{\alpha},\tilde{\gamma},\lambda,\upsilon)=\begin{bmatrix}\dfrac{\partial\,\mathcal{L}_1^p(\Theta,\beta;\;\tilde{\alpha},\tilde{\gamma},\lambda,\upsilon)}{\partial\propto_1}\\ \dfrac{\partial\,\mathcal{L}_1^p(\Theta,\beta;\;\tilde{\alpha},\tilde{\gamma},\lambda,\upsilon)}{\partial\propto_2}\\ \vdots\\ \dfrac{\partial\,\mathcal{L}_1^p(\Theta,\beta;\;\tilde{\alpha},\tilde{\gamma},\lambda,\upsilon)}{\partial\propto_i}\\ \vdots\\ \dfrac{\partial\,\mathcal{L}_1^p(\Theta,\beta;\;\tilde{\alpha},\tilde{\gamma},\lambda,\upsilon)}{\partial\propto_n}\\ \dfrac{\partial\,\mathcal{L}_1^p(\Theta,\beta;\;\tilde{\alpha},\tilde{\gamma},\lambda,\upsilon)}{\partial\mu_1}\\ \dfrac{\partial\,\mathcal{L}_1^p(\Theta,\beta;\;\tilde{\alpha},\tilde{\gamma},\lambda,\upsilon)}{\partial\mu_2}\\ \vdots\\ \dfrac{\partial\,\mathcal{L}_1^p(\Theta,\beta;\;\tilde{\alpha},\tilde{\gamma},\lambda,\upsilon)}{\partial\mu_{\mathrm{i}}}\\ \vdots\\ \dfrac{\partial\,\mathcal{L}_1^p(\Theta,\beta;\;\tilde{\alpha},\tilde{\gamma},\lambda,\upsilon)}{\partial\mu_n}\\ \dfrac{\partial\,\mathcal{L}_1^p(\Theta,\beta;\;\tilde{\alpha},\tilde{\gamma},\lambda,\upsilon)}{\partial\lambda}\\ \dfrac{\partial\,\mathcal{L}_1^p(\Theta,\beta;\;\tilde{\alpha},\tilde{\gamma},\lambda,\upsilon)}{\partial\,\upsilon}\end{bmatrix}=0 \qquad (i=1,2,\ldots,n),$$

$$\frac{\partial\,\mathcal{L}_1^P(\Theta,\beta;\;\tilde{\alpha},\tilde{\gamma},\lambda,\upsilon)}{\partial\propto_i}=0\;\;\text{için}$$

$$\frac{\partial\, \mathcal{L}_1^P(\Theta,\beta;\ \tilde{\alpha},\tilde{\gamma},\lambda,\upsilon)}{\partial \propto_i} = -\frac{1}{2}.\,2 \propto_i (-\propto_i + \ \mu_i -\ 2\lambda) - \frac{1}{2}.\,1 \sum_{\substack{i,j=1 \\ i\neq j}}^{n} \left(-\propto_j + \ \mu_j - 2\lambda\right) K_{ij}$$

$$-\underbrace{\sum_{\kappa=1}^{K} \beta_\kappa}_{=1\ idi.} - \sum_{l\, \in L_0(\bar{\tilde{\alpha}},\bar{\lambda},\bar{\upsilon})} \widetilde{\gamma_l}^{\,1}\ \frac{\partial\, v_l^1(\bar{\tilde{\alpha}})}{\partial \propto_i}$$

ifadesinin sıfıra eşitlenmesi ile,

$$= -\propto_i (-\propto_i + \ \mu_i -\ 2\lambda) - \frac{1}{2} \sum_{\substack{i,j=1 \\ i\neq j}}^{n} \left(-\propto_j + \ \mu_j - 2\lambda\right)\ K_{ij} - 1$$

$$- \sum_{l\, \in L_0(\bar{\tilde{\alpha}},\bar{\lambda},\bar{\upsilon})} \widetilde{\gamma_l}^{\,1}\ \frac{\partial\, v_l^1(\bar{\tilde{\alpha}})}{\partial \propto_i} = 0 \qquad \mathbf{(4.41)}$$

elde edilir.

$$\frac{\partial\, \mathcal{L}_1^P(\Theta,\beta;\ \tilde{\alpha},\tilde{\gamma},\lambda,\upsilon)}{\partial \mu_i} = 0 \text{ için}$$

$$\frac{\partial\, \mathcal{L}_1^P(\Theta,\beta;\ \tilde{\alpha},\tilde{\gamma},\lambda,\upsilon)}{\partial \mu_i} = \ \frac{1}{2}.\,2\, \mu_i(-\propto_i + \ \mu_i -\ 2\lambda) + \frac{1}{2}.\,1 \sum_{\substack{i,j=1 \\ i\neq j}}^{n} \left(-\propto_j + \ \mu_j - 2\lambda\right) K_{ij}$$

$$-\underbrace{\sum_{i=1}^{n} \beta_\kappa}_{=1\ idi.} + \sum_{i=1}^{n} \beta_\kappa D - \sum_{l\, \in L_0(\bar{\tilde{\alpha}},\bar{\lambda},\bar{\upsilon})} \widetilde{\gamma_l}\, \frac{\partial\, v_l^1(\bar{\tilde{\alpha}})}{\partial \mu_i}$$

$$= \ \mu_i(-\propto_i + \ \mu_i -\ 2\lambda) + \frac{1}{2} \sum_{\substack{i,j=1 \\ i\neq j}}^{n} \left(-\propto_j + \ \mu_j - 2\lambda\right) K_{ij} - 1 + \mathrm{D}$$

$$- \sum_{l\, \in L_0(\bar{\tilde{\alpha}},\bar{\lambda},\bar{\upsilon})} \widetilde{\gamma_l}\, \frac{\partial\, v_l^1(\bar{\tilde{\alpha}})}{\partial \mu_i} = 0 \qquad \mathbf{(4.42)}$$

elde edilir.

$$\frac{\partial\, \mathcal{L}_1^P(\Theta,\beta;\ \tilde{\alpha},\tilde{\gamma},\lambda,\upsilon)}{\partial\lambda} = 0 \text{ için}$$

$$\frac{\partial\, \mathcal{L}_1^P(\Theta,\beta;\ \tilde{\alpha},\tilde{\gamma},\lambda,\upsilon)}{\partial\lambda} = -\frac{1}{2}.2.2(-\propto_i+\ \mu_i-2\lambda)\ -\frac{1}{2}.2\sum_{\substack{i,j=1\\ i\neq j}}^{n}\left(-\propto_j+\ \mu_j-2\lambda\right)K_{ij}$$

$$-\frac{1}{2}.2\sum_{\substack{i,j=1\\ i\neq j}}^{n}(-\propto_{\mathrm{i}}+\ \mu_{\mathrm{i}}-2\lambda)\,K_{ij}-\ \tilde{\gamma}^2,$$

$$=-2\lambda(-\propto_i+\mu_i-2\lambda)-\sum_{\substack{i,j=1\\ i\neq j}}^{n}\left(-\propto_j+\ \mu_j-2\lambda\right)K_{ij}$$

$$-\sum_{\substack{i,j=1\\ i\neq j}}^{n}(-\propto_{\mathrm{i}}+\mu_{\mathrm{i}}-2\lambda)\,K_{ij}-\tilde{\gamma}^2=0 \qquad (\mathbf{4.43})$$

elde edilir.

$$\frac{\partial\, \mathcal{L}_1^P(\Theta,\beta;\ \tilde{\alpha},\tilde{\gamma},\lambda,\upsilon)}{\partial\upsilon} = 0 \text{ için}$$

$$\frac{\partial\, \mathcal{L}_1^P(\Theta,\beta;\ \tilde{\alpha},\tilde{\gamma},\lambda,\upsilon)}{\partial\upsilon} =\ 1-\ \tilde{\gamma}^3,$$

$$1-\ \tilde{\gamma}^3=0 \qquad (\mathbf{4.44})$$

elde edilir.

Lineer olmayan denklem sistemleri (4.41), (4.42), (4.43) ve (4.44)'ü nümerik yollarla çözmek mümkündür (örneğin, Newton metodu). Ancak lineer olmayan denklem sistemini çözmek bu tezin konusu olmadığından detaylara değinilmeyecektir.

2. ***Durum***: Eğer, $L_0\left(\bar{\tilde{\alpha}},\ \bar{\lambda},\ \bar{\upsilon}\ \right)=\emptyset$ olması halinde aktif kısıtlar sadece eşitlik fonksiyonlarında yer alacaktır. Bu tezdeki problem (CP)'de eşitlik kısıtları

olmadığından, Lagrange fonksiyonu, denklem $(\mathrm{CKO}_G)$ ile verilen düşük düzeyli özgün problemin amaç fonksiyonuna eşit olur:

$$\mathcal{L}_2^P(\Theta,\beta;\ \tilde{\alpha},\tilde{\gamma},\lambda,\upsilon) = \ g\left((\Theta,\beta)\ ,\propto,\mu,\lambda,\upsilon\ \right)$$

$$= \frac{1}{2}\sum_{i,j=1}^{n}(-\propto_i+\ \mu_i-\ 2\lambda)\left(-\propto_j+\ \mu_j-2\lambda\right)K_{ij}-\sum_{\kappa=1}^{K}\sum_{i=1}^{n}\beta_\kappa\propto_i$$

$$-\sum_{\kappa=1}^{K}\sum_{i=1}^{n}\beta_\kappa\,\mu_i+\ \sum_{\kappa=1}^{K}\sum_{i=1}^{n}\ \beta_\kappa\mu_i\,D-\ \Theta+\upsilon,$$

$$=\frac{1}{2}\sum_{i=1}^{n}(-\propto_i+\ \mu_i-\ 2\lambda)^2\ K_{ii}+\ \frac{1}{2}\sum_{\substack{i,j=1\\ i\neq j}}^{n}(-\propto_i+\ \mu_i-\ 2\lambda)\left(-\propto_j+\ \mu_j-2\lambda\right)K_{ij}$$

$$-\sum_{\kappa=1}^{K}\sum_{i=1}^{n}\beta_\kappa\propto_i-\sum_{\kappa=1}^{K}\sum_{i=1}^{n}\beta_\kappa\,\mu_i+\sum_{\kappa=1}^{K}\sum_{i=1}^{n}\beta_\kappa\mu_i\,D-\ \Theta+\upsilon. \qquad \mathbf{(4.45)}$$

Burada durum 1'de olduğu gibi kernel fonksiyonumuzun Gauss kerneli olduğu varsayılmaktadır. Lagrange fonksiyonunda, $\mathcal{L}_2^P(\Theta,\beta;\ \tilde{\alpha},\tilde{\gamma},\lambda,\upsilon)=0$ denklemini sağlayan $\Theta,\beta,\tilde{\alpha},\tilde{\lambda},\tilde{\gamma}$ değerleri aranmaktadır. Bunun için denklem (4.45) ile verilen Lagrange fonksiyonunun, Jakobyen matrisini sıfıra eşitleyelim:

$$\nabla_{(\tilde{\alpha},\lambda,\upsilon)}\mathcal{L}_2^P(\Theta,\beta;\ \tilde{\alpha},\tilde{\gamma},\lambda,\upsilon)\ =0$$

$$\nabla_{(\tilde{\alpha},\lambda,\upsilon)}\,\mathcal{L}_2^P(\Theta,\beta;\ \tilde{\alpha},\tilde{\gamma},\lambda,\upsilon)=\begin{bmatrix}\dfrac{\partial\,\mathcal{L}_2^p(\Theta,\beta;\ \tilde{\alpha},\tilde{\gamma},\lambda,\upsilon)}{\partial \propto_1}\\ \dfrac{\partial\,\mathcal{L}_2^p(\Theta,\beta;\ \tilde{\alpha},\tilde{\gamma},\lambda,\upsilon)}{\partial \propto_2}\\ \vdots\\ \dfrac{\partial\,\mathcal{L}_2^p(\Theta,\beta;\ \tilde{\alpha},\tilde{\gamma},\lambda,\upsilon)}{\partial \propto_i}\\ \vdots\\ \dfrac{\partial\,\mathcal{L}_2^p(\Theta,\beta;\ \tilde{\alpha},\tilde{\gamma},\lambda,\upsilon)}{\partial \propto_n}\\ \dfrac{\partial\,\mathcal{L}_2^p(\Theta,\beta;\ \tilde{\alpha},\tilde{\gamma},\lambda,\upsilon)}{\partial\mu_1}\\ \dfrac{\partial\,\mathcal{L}_2^p(\Theta,\beta;\ \tilde{\alpha},\tilde{\gamma},\lambda,\upsilon)}{\partial\mu_2}\\ \vdots\\ \dfrac{\partial\,\mathcal{L}_2^p(\Theta,\beta;\ \tilde{\alpha},\tilde{\gamma},\lambda,\upsilon)}{\partial\mu_\mathrm{i}}\\ \vdots\\ \dfrac{\partial\,\mathcal{L}_2^p(\Theta,\beta;\ \tilde{\alpha},\tilde{\gamma},\lambda,\upsilon)}{\partial\mu_n}\\ \dfrac{\partial\,\mathcal{L}_2^p(\Theta,\beta;\ \tilde{\alpha},\tilde{\gamma},\lambda,\upsilon)}{\partial\lambda}\\ \dfrac{\partial\,\mathcal{L}_2^p(\Theta,\beta;\ \tilde{\alpha},\tilde{\gamma},\lambda,\upsilon)}{\partial\upsilon}\end{bmatrix}=0\qquad (i=1,2,\dots,n),$$

$$\frac{\partial\,\mathcal{L}_2^P(\Theta,\beta;\ \tilde{\alpha},\tilde{\gamma},\lambda,\upsilon)}{\partial \propto_i}=-\propto_i\,(-\propto_i+\mu_i-2\lambda).K_{ii}-\frac{1}{2}.1\sum_{\substack{i,j=1\\ i\neq j}}^{n}\left(-\propto_j+\ \mu_j-2\lambda\right)\,K_{ij}-1$$

$K_{ii}=1$ iken sistem denklemi aşağıdaki gibi olur:

$$\frac{\partial\,L_2^P(\Theta,\beta;\ \tilde{\alpha},\tilde{\gamma},\lambda,\upsilon)}{\partial \propto_i}=0 \text{ için,}$$

$$\propto_i\,(-\propto_i+\ \mu_i-\ 2\lambda)+\frac{1}{2}\sum_{\substack{i,j=1\\ i\neq j}}^{n}\left(-\propto_j+\ \mu_j-2\lambda\right)\,K_{ij}+1=0 \qquad \mathbf{(4.46)}$$

ile ifade edilir.

$$\frac{\partial \mathcal{L}_2^P(\Theta,\beta;\ \tilde{\alpha},\tilde{\gamma},\lambda,\upsilon)}{\partial \mu_i}$$

$$= \mu_i(-\propto_i + \mu_i - 2\lambda).K_{ii} + \frac{1}{2}.1\sum_{\substack{i,j=1\\ i\neq j}}^{n}\left(-\propto_j + \mu_j - 2\lambda\right) K_{ij} - \underbrace{\sum_{\kappa=1}^{K}\beta_\kappa}_{=1}$$

$K_{ii} = 1$ iken,

$$\frac{\partial L_2^P(\Theta,\beta;\ \tilde{\alpha},\tilde{\gamma},\lambda,\upsilon)}{\partial \mu_i} = 0 \text{ için,}$$

$$\mu_i(-\propto_i + \mu_i - 2\lambda) + \frac{1}{2}\sum_{\substack{i,j=1\\ i\neq j}}^{n}\left(-\propto_j + \mu_j - 2\lambda\right) K_{ij} - 1 + D = 0 \qquad \textbf{(4.47)}$$

bulunur.

$$\frac{\partial \mathcal{L}_2^P(\Theta,\beta;\ \tilde{\alpha},\tilde{\gamma},\lambda,\upsilon)}{\partial \lambda} = \frac{1}{2}.2.2(-\propto_i + \mu_i - 2\lambda).\underbrace{K_{ii}}_{=1}$$

$$+ \frac{1}{2}\sum_{\substack{i,j=1\\ i\neq j}}^{n}\left[(-2).\left(-\propto_j + \mu_j - 2\lambda\right) - 2.(-\propto_i + \mu_i - 2\lambda)\right] K_{ij}$$

$$= 2(-\propto_i + \mu_i - 2\lambda) - \frac{1}{2}.2\sum_{\substack{i,j=1\\ i\neq j}}^{n}\left(-\propto_j + \mu_j - 2\lambda - \propto_i + \mu_i - 2\lambda\right) K_{ij}$$

$$\frac{\partial \mathcal{L}_2^P(\Theta,\beta;\ \tilde{\alpha},\tilde{\gamma},\lambda,\upsilon)}{\partial \lambda} = 0 \text{ için,}$$

$$2(-\propto_i + \mu_i - 2\lambda) - \sum_{\substack{i,j=1\\ i\neq j}}^{n}\left(-\propto_i + \mu_i - \propto_j + \mu_j - 4\lambda\right) K_{ij} = 0 \text{ dan,}$$

$$\sum_{\substack{i,j=1 \\ i\neq j}}^{n} \left(-\propto_i + \mu_i - \propto_j + \mu_j - 4\lambda\right) K_{ij} = 2(-\propto_i + \mu_i - 2\lambda) \qquad \mathbf{(4.48)}$$

bulunur.

$$\frac{\partial \mathcal{L}_2^P(\Theta, \beta;\ \tilde{\alpha}, \tilde{\gamma}, \lambda, \upsilon)}{\partial \upsilon} = 1 \qquad \mathbf{(4.49)}$$

bulunur.

Karşımıza tekrar durum 1'deki gibi, lineer olmayan denklem sistemi çıkmıştır. Lineer olmayan denklem sistemleri (4.46), (4.47), (4.48) ve (4.49)'u nümerik yollarla çözmek mümkündür (örneğin, Newton metodu). Ancak lineer olmayan denklem sistemini çözmek bu tezin konusu olmadığından detaylara değinilmeyecektir.

**3. *İkinci Dereceden Uygunluk Koşulları – IUK (SOC):***

Daha önce (4.37) ile tanımlanan $\tilde{\gamma}$ değeri için ,

$$\eta^T \nabla^2_{(\tilde{\alpha},\lambda,\upsilon)^T} \mathcal{L}^P(\Theta, \beta;\ \tilde{\alpha}, \tilde{\gamma}, \lambda, \upsilon)\eta > 0 \qquad \forall\, \eta \in \mathcal{T}^p\left(\bar{\tilde{\boldsymbol{\alpha}}}, \lambda, \upsilon\right) - \{0\}$$

tanjant uzayını $\bar{\tilde{\boldsymbol{\alpha}}}, \bar{\boldsymbol{\lambda}}, \bar{\boldsymbol{\upsilon}}$ noktalarında tanımlarsak:

$$\mathcal{T}^p(\bar{\tilde{\alpha}}, \lambda, \upsilon) := \left\{\tau, \varsigma, \rho \in R^n \,\middle|\, \nabla^{\mathrm{T}} \upsilon_l(\bar{\alpha})\tau = 0, \nabla^T \upsilon^2(\bar{\lambda})\varsigma = 0\,, \nabla^T \upsilon^3(\bar{\upsilon})\rho = 0, \left(l \in L_0\left(\bar{\tilde{\alpha}}, \bar{\lambda}, \bar{\upsilon}\right)\right)\right\}$$

elde edilir. Sağlanması gereken tüm durum ve koşullar için tanjant uzayını ifade edelim. Gösterimde kolaylık olması bakımından $\Omega := \begin{bmatrix} \tau \\ \varsigma \\ \rho \end{bmatrix}$ ile tanımlansın.

<u>1. Durum</u>: Eğer $L_0\left(\bar{\tilde{\alpha}}, \bar{\lambda}, \bar{\upsilon}\right) \neq \emptyset$ ise tanjant uzayı,

$$\mathcal{T}_1{}^p(\bar{\tilde{\alpha}}, \lambda, \upsilon) := \left\{\Omega \in R^n \,\middle|\, \mathrm{DA}\left(\bar{\tilde{\alpha}},\ \bar{\lambda},\ \bar{\upsilon}\right)\Omega = 0 \quad \left(l \in L_0\left(\bar{\tilde{\alpha}},\ \bar{\lambda},\ \bar{\upsilon}\right)\right)\right\} \qquad \mathbf{(4.50)}$$

koşulu ile tanımlanır.

Örnek Jacobyen matrisi tanjant uzayındaki tanım (4.50)'de yerine yazalım:

$$DA(\bar{\tilde{\alpha}},\ \bar{\lambda},\ \bar{\upsilon})\Omega = \begin{bmatrix} 1 & 0 & \cdots & 0 & 1 & 0 & \cdots & 1\,1 \\ 0 & 0 & 1 & \cdots & 1 & \cdots & 0 & 1\,1 \\ 0 & 1 & \cdots & 0 & 1 & 0 & \cdots & 1\,1 \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ 0 & 0 & -1 & \cdots & 0 & -1 & \cdots & 1\,1 \\ -1 & 0 & \cdots & 0 & -1 & \cdots & 0 & 1\,1 \\ 0 & 0 & \cdots & \cdots & -1 & 0 & \cdots & 1\,1 \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ 0 & 0 & 0 & \cdots & -1 & 0 & \cdots & 1\,1 \end{bmatrix} \begin{bmatrix} \Omega_1 \\ \Omega_2 \\ \Omega_3 \\ . \\ . \\ . \\ . \\ . \\ \Omega_l \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ . \\ . \\ . \\ . \\ . \\ . \\ 0 \end{bmatrix}, \qquad (\mathbf{4.51})$$

$\nabla v_l\ (\tilde{\alpha}) = \ (0, \ldots, \pm 1\ ,0 \ldots 0)^T \quad l\ \in L_0(\bar{\tilde{\alpha}},\ \bar{\lambda},\ \bar{\upsilon}), \quad \nabla v^2\ (\lambda) = \ (0, \ldots, 1\ ,0 \ldots 0)^T,$

$\nabla v^3\ (\upsilon) = \ (0, \ldots, 1\ ,0 \ldots 0)^T$ ve $k = \left|L_0(\bar{\tilde{\alpha}},\ \bar{\lambda},\ \bar{\upsilon})\right|$'dır.

Denklem (4.51)'de aktif kısıtlar,

$$\Omega_r = 0 \qquad \left(r \in L_0(\bar{\tilde{\alpha}},\ \bar{\lambda},\ \bar{\upsilon})\ \cap\ \{1,2, \ldots, 2n\}\right),$$

$\Omega_s = 0 \qquad \left(s \in L_0(\bar{\tilde{\alpha}},\ \bar{\lambda},\ \bar{\upsilon})\ \cap\ \{2n,\ 2n+1, \ldots, 4n\}\right)$ ile yazılabilir.

$\nabla^2_{(\tilde{\alpha},\lambda,\upsilon)^T} {\mathcal{L}_1}^P\left(\Theta, \beta;\ \tilde{\alpha}, \tilde{\lambda}, \tilde{\upsilon}, \tilde{\gamma}\right)$ matrisini yazalım:

$$\nabla^2_{(\tilde{\alpha},\lambda,\upsilon)^T} {\mathcal{L}_1}^P\left(\Theta, \beta;\ \tilde{\alpha}, \tilde{\lambda}, \tilde{\upsilon}, \tilde{\gamma}\right) =$$

$$\begin{bmatrix} \frac{\partial^2\, \mathcal{L}_1^p(\Theta,\beta;\,\tilde{\alpha},\lambda,\upsilon,\tilde{\gamma})}{\partial^2 \propto_1} & \cdots & \frac{\partial^2\, \mathcal{L}_1^p(\Theta,\beta;\,\tilde{\alpha},\lambda,\upsilon,\tilde{\gamma})}{\partial \propto_1 \partial \propto_n} & \frac{\partial^2\, \mathcal{L}_1^p(\Theta,\beta;\,\tilde{\alpha},\lambda,\upsilon,\tilde{\gamma})}{\partial \propto_1 \partial \mu_1} & \cdots & \frac{\partial^2\, \mathcal{L}_1^p(\Theta,\beta;\,\tilde{\alpha},\lambda,\upsilon,\tilde{\gamma})}{\partial \propto_1 \partial \mu_n} & \frac{\partial^2\, \mathcal{L}_1^p(\Theta,\beta;\,\tilde{\alpha},\lambda,\upsilon,\tilde{\gamma})}{\partial \propto_1 \partial \lambda} & \frac{\partial^2\, \mathcal{L}_1^p(\Theta,\beta;\,\tilde{\alpha},\lambda,\upsilon,\tilde{\gamma})}{\partial \propto_1 \partial \upsilon} \\ \frac{\partial^2\, \mathcal{L}_1^p(\Theta,\beta;\,\tilde{\alpha},\lambda,\upsilon,\tilde{\gamma})}{\partial \propto_2 \partial \propto_1} & \cdots & \frac{\partial^2\, \mathcal{L}_1^p(\Theta,\beta;\,\tilde{\alpha},\lambda,\upsilon,\tilde{\gamma})}{\partial \propto_2 \partial \propto_n} & \frac{\partial^2\, \mathcal{L}_1^p(\Theta,\beta;\,\tilde{\alpha},\lambda,\upsilon,\tilde{\gamma})}{\partial \propto_2 \partial \mu_1} & \cdots & \frac{\partial^2\, \mathcal{L}_1^p(\Theta,\beta;\,\tilde{\alpha},\lambda,\upsilon,\tilde{\gamma})}{\partial \propto_2 \partial \mu_n} & \frac{\partial^2\, \mathcal{L}_1^p(\Theta,\beta;\,\tilde{\alpha},\lambda,\upsilon,\tilde{\gamma})}{\partial \propto_2 \partial \lambda} & \frac{\partial^2\, \mathcal{L}_1^p(\Theta,\beta;\,\tilde{\alpha},\lambda,\upsilon,\tilde{\gamma})}{\partial \propto_2 \partial \upsilon} \\ & & \vdots & & \vdots & & \vdots & \\ \frac{\partial^2\, \mathcal{L}_1^p(\Theta,\beta;\,\tilde{\alpha},\lambda,\upsilon,\tilde{\gamma})}{\partial \propto_n \partial \propto_1} & \cdots & \frac{\partial^2\, \mathcal{L}_1^p(\Theta,\beta;\,\tilde{\alpha},\lambda,\upsilon,\tilde{\gamma})}{\partial^2 \propto_n} & \frac{\partial^2\, \mathcal{L}_1^p(\Theta,\beta;\,\tilde{\alpha},\lambda,\upsilon,\tilde{\gamma})}{\partial \propto_n \partial \mu_1} & \cdots & \frac{\partial^2\, \mathcal{L}_1^p(\Theta,\beta;\,\tilde{\alpha},\lambda,\upsilon,\tilde{\gamma})}{\partial \propto_n \partial \mu_n} & \frac{\partial^2\, \mathcal{L}_1^p(\Theta,\beta;\,\tilde{\alpha},\lambda,\upsilon,\tilde{\gamma})}{\partial \propto_n \partial \lambda} & \frac{\partial^2\, \mathcal{L}_1^p(\Theta,\beta;\,\tilde{\alpha},\lambda,\upsilon,\tilde{\gamma})}{\partial \propto_n \partial \upsilon} \end{bmatrix},$$

$$\frac{\partial\, \mathcal{L}_1^p(\Theta, \beta;\ \tilde{\alpha}, \lambda, \tilde{\gamma})}{\partial \propto_i} = \ -\propto_i\ (-\propto_i +\ \mu_i -\ 2\lambda) - \frac{1}{2} \sum_{\substack{i,j=1 \\ i\, \neq j}}^{n} \left(-\propto_j +\ \mu_j - 2\lambda\right)\ K_{ij}$$

$$-1 - \sum_{l\, \in L_0(\bar{\tilde{\alpha}},\bar{\lambda},\bar{\upsilon})} \tilde{\gamma}_l^{\ 1}\ \frac{\partial\ v_l^1(\bar{\tilde{\alpha}})}{\partial \propto_i},$$

$$\frac{\partial^2 \mathcal{L}_1^p(\Theta,\beta;\ \tilde{\alpha},\lambda,\tilde{\gamma})}{\partial^2 \propto_i} = -(-\propto_i + \mu_i - 2\lambda) - (-\propto_i) - \underbrace{\sum_{l\in L_0(\bar{\tilde{\alpha}},\bar{\lambda},\bar{v})} \widetilde{\gamma_l}\,\frac{\partial^2 v_l^1(\bar{\tilde{\alpha}})}{\partial^2 \propto_i}}_{=0\,.}$$

$$= \propto_i - \mu_i + 2\lambda + \propto_i$$

$$= 2\propto_i + 2\lambda - \mu_i,$$

$$\frac{\partial^2 \mathcal{L}_1^p(\Theta,\beta;\ \tilde{\alpha},\lambda,\tilde{\gamma})}{\partial \propto_i\ \partial \propto_j} = 0 - \frac{1}{2}.(-1).\sum_{\substack{i,j=1\\ i\neq j}}^{n} K_{ij} - \sum_{l\in L_0(\bar{\tilde{\alpha}},\bar{\lambda},\bar{v})} \widetilde{\gamma_l}\,\frac{\partial^2 v_l^1(\bar{\tilde{\alpha}})}{\partial \propto_i\ \partial \propto_j}$$

$$= \frac{1}{2}\sum_{\substack{i,j=1\\ i\neq j}}^{n} K_{ij}\,,$$

$$\frac{\partial^2 \mathcal{L}_1^p(\Theta,\beta;\ \tilde{\alpha},\lambda,\tilde{\gamma})}{\partial \propto_i\ \partial \mu_i} = (-1).\propto_i - \sum_{l\in L_0(\bar{\tilde{\alpha}},\bar{\lambda},\bar{v})} \widetilde{\gamma_l}\,\frac{\partial^2 v_l^1(\bar{\tilde{\alpha}})}{\partial \propto_i\ \partial \mu_i}$$

$$= -\propto_i - \sum_{l\in L_0(\bar{\tilde{\alpha}},\bar{\lambda},\bar{v})} \widetilde{\gamma_l}\,\frac{\partial\, v_l^1(\bar{\tilde{\alpha}})}{\partial \propto_i\ \partial \mu_i},$$

$$\frac{\partial^2 \mathcal{L}_1^p(\Theta,\beta;\ \tilde{\alpha},\lambda,\tilde{\gamma})}{\partial \propto_i\ \partial \mu_j} = -\frac{1}{2}.1.\sum_{\substack{i,j=1\\ i\neq j}}^{n} K_{ij} - \sum_{l\in L_0(\bar{\tilde{\alpha}},\bar{\lambda},\bar{v})} \widetilde{\gamma_l}\,\frac{\partial^2 v_l^1(\bar{\tilde{\alpha}})}{\partial \propto_i\ \partial \mu_j}$$

$$= -\frac{1}{2}\sum_{\substack{i,j=1\\ i\neq j}}^{n} K_{ij} - \sum_{l\in L_0(\bar{\tilde{\alpha}},\bar{\lambda},\bar{v})} \widetilde{\gamma_l}\,\frac{\partial^2 v_l^1(\bar{\tilde{\alpha}})}{\partial \propto_i\ \partial \mu_j},$$

$$\frac{\partial^2 \mathcal{L}_1^p(\Theta,\beta;\ \tilde{\alpha},\lambda,\tilde{\gamma})}{\partial \propto_i\ \partial \lambda} = -\propto_i(-2) + \left(-\frac{1}{2}\right).(-2)\sum_{\substack{i,j=1\\ i\neq j}}^{n} K_{ij} - 0$$

$$= 2 \propto_i + \sum_{\substack{i,j=1 \\ i \neq j}}^{n} K_{ij} \,.$$

IDK' nın bu durumu için, hessian matrisinin pozitif yarı tanımlı olup olmaması veriye bağlıdır. Bu durumda kesin bir sonuca varamayız.

2. *Durum:* Eğer $L_0(\bar{\bar{\alpha}}, \bar{\lambda}, \bar{\upsilon}) = \emptyset$ ise tanjant uzayı,

$\mathcal{T}_2{}^p(\bar{\bar{\alpha}}, \lambda, \upsilon)$'de boş küme $(\mathcal{T}_2{}^p(\bar{\bar{\alpha}}, \lambda, \upsilon\,) := \emptyset)$ olduğu için, ikinci dereceden koşullar IDK (SOC) her durumda sağlanır.

Böylece Bölüm 2, Tanım 2.3.6 ile verilen Ansatz Azaltıcı koşulları IDK'da, ilk durum yukarıda da anlatıldığı üzere, Hessian matrisinin pozitif tanımlı olması veriye bağlıdır. Buna karşın diğer maddelerin sağlandığı görülmüştür. O halde, Teorem 2.3.9 gereğince, Yarı Sonsuz Programlama (YSP)'nın düşük düzey problemi, yarı sonsuz iken sonlu problem haline getiririlmiştir.

Yarı – Sonsuz Programlama ile modellenen problem (*CKO$_F$*)'i aşağıdaki algoritma ile iteratif şekilde çözmek mümkündür:

**Algoritma 1:** Yarı Sonsuz Programlama ile Çok Kernelli Öbekleme Algoritması (CKO$_{YSP}$)

Girdi:

$(\Theta_0, \boldsymbol{\beta}^0)$ : optimal çözüm için ilk tahmin (uygunluk şartı gözardı edilebilir.)

$\varepsilon_0$ : algoritmanın durma/ sonlandırma kriteri için yeterince küçük pozitif sayı

Çıktı:

$\Theta_k$ : minimizasyon poblemi için bilinmeyen değişken

$\boldsymbol{\beta}^k$ : kernel katsayısı

$\widetilde{\boldsymbol{\alpha}}_k, \lambda_k, \upsilon_k$ : dual değişkenler $(CKO_G)$

CKO$_{YSP}$ $(\Theta_k, \boldsymbol{\beta}^k, \widetilde{\boldsymbol{\alpha}}_k, \lambda_k, \upsilon_k, \Theta_0, \boldsymbol{\beta}^0)$

Algoritma:

1. k:= 0
2. Aşağıdaki problemin bütün yerel minimumlarını $\widetilde{\alpha}_k^1, \widetilde{\alpha}_k^2, \ldots, \widetilde{\alpha}_k^r, \lambda_k, \upsilon_k$ bul.

$$\min_{\widetilde{\boldsymbol{\alpha}}, \lambda, \upsilon \in \mathrm{A}} \; g\left((\Theta_k, \boldsymbol{\beta}^k), \widetilde{\boldsymbol{\alpha}}_k, \lambda_k, \upsilon_k\right)$$

3. DO aşağıdaki problemin $(\Theta^*, \boldsymbol{\beta}^*)$ sonucunu bul.

$$\min_{\substack{\Theta \in R \\ \boldsymbol{\beta} \in R^k}} \; -\Theta$$

kısıtlar

$$g\left((\Theta, \beta), \widetilde{\alpha}_k^l, \lambda_k, \upsilon_k\right) = \sum_{\kappa=1}^{K} \beta_\kappa S_\kappa(\widetilde{\alpha}_k^l, \lambda_k^l, \upsilon_k^l) - \Theta + \upsilon \geq 0 \;\; (l = 1,2, \ldots, r),$$

$$\sum_{\kappa=1}^{K} \beta_\kappa = 1.$$

4. Eğer $\varepsilon_0 > 0$ sabiti için,

$$\|(\Theta_{k+1}, \boldsymbol{\beta}^{k+1}) - (\Theta_k, \boldsymbol{\beta}^k)\|_2 \leq \varepsilon_0 \text{ ise}$$

5. STOP
6. else
7. $(\Theta_{k+1}, \boldsymbol{\beta}^{k+1}) := (\Theta^*, \boldsymbol{\beta}^*)$
8. k := k+1
9. end if
10. END DO

**($\mathbf{P}_Z$) ve ($CP$) Problemlerinin Karşılaştırılması:**

Bin Zhao ve diğ. (2009), problem ($P_Z$)'yi oluştururken $\boldsymbol{\beta} = (\beta_1, .., \beta_\kappa)^T$ olmak üzere, $p = 2$ alarak katsayıları $l_2\ norm$'a kısıtlamışlardır. Oysaki bu tezde problem (CP)'yi oluştururken $\sum_{\kappa=1}^{K} \beta_\kappa = 1$ ve amaç fonksiyonunun parantez karesini alarak $l_1$ norm'a kısıtlama yapılmıştır. Bu kısıtlamayı yaparak, $\boldsymbol{\beta}$ vektörlerini blok seviyesinde seyrek (sparse), blok içinde ise seyrek olmayan (*non sparse*) $\mathbf{w}$ vektörleri bulmak amaçlanmıştır. Bu nedenle en büyük aralık ile öbekleme problemi (CP)'yi oluştururken, $\beta_\kappa \geq 0,\ \forall\ \kappa \in \{1, \dots, K\}$ için $\sum_{\kappa=1}^{K} \beta_\kappa = 1$ ve $\mathbf{w}_\kappa = \beta_\kappa\, \mathbf{w}'_\kappa$ eşitlikleri göz önünde bulundurulmuştur.

$$\min_{\beta, \mathbf{w}, b, \xi} \quad \frac{1}{2} \sum_{\kappa=1}^{K} \beta_\kappa \|\mathbf{w}_\kappa\|^2 + C \sum_{i=1}^{n} \xi_i \qquad (P_Z)$$

$$\left| \sum_{\kappa=1}^{K} \beta_\kappa \mathbf{w}_\kappa^T\, \Phi_\kappa(\mathbf{x_i}) + b \right| \geq 1 - \xi_i\ ,$$

$$\xi_i \geq 0 \quad \forall\ i = 1, \dots, n,$$

$$\beta_\kappa \geq 0 \qquad \forall\ \kappa = 1, \dots, K, \qquad \sum_{\kappa=1}^{K} \beta_\kappa^p \leq 1,$$

$$-l \leq \sum_{i=1}^{n} \left[ \sum_{\kappa=1}^{K} \beta_\kappa \mathbf{w_\kappa}^T\, \Phi_\kappa(\mathbf{x_i}) + b \right] \leq l\ , l > 0$$

$$\min_{w,\ b,\ \xi} \quad \frac{1}{2} \left( \sum_{\kappa=1}^{K} \|\mathbf{w}_\kappa\|_2 \right)^2 + C \sum_{i=1}^{n} \xi_i \quad (CP)$$

$$\left| \sum_{\kappa=1}^{K} \mathbf{w}_\kappa^T\, \Phi_\kappa(x_i) + b \right| \geq 1 - \xi_i\ ,$$

$$\xi_i \geq 0 \quad \forall\ i = 1, \dots, n,$$

$$-l \leq \sum_{i=1}^{n} \left[ \sum_{\kappa=1}^{K} \mathbf{w}_\kappa^T\, \Phi_\kappa(x_i) + b \right] \leq l\ , l > 0$$

Bin Zhao ve diğ. (2009) tarafından yukarıda ($P_Z$) ile tanıtılan EAO (MMC) problemi, İkinci Dereceden Konik Program (*Second order cone program*) kullanılarak Kesen Düzlem (*Cutting Plane*) algoritması ile modellenmiştir.

Bu tezde ise yukarıda ($CP$) ile tanıtılan EAO (MMC) problemi Yarı – Sonsuz Programlama ile modellenmiştir.

# 5. SONUÇ ve TARTIŞMA

Bu tezde, makine öğrenmesi yöntemlerinden ağırlıklı olarak öbeklemeye yer verilmiş ve makine öğrenmesinin temelini oluşturan optimizasyon teorisinin önemli teoremleri ve en iyilik koşulları tanıtılmıştır. Literatürde kullanılan öbekleme yöntemleri özetlenmiş ve en büyük aralık ile öbekleme yönteminde en son geliştirilen algoritma (Zhao ve diğ. 2009) hakkında detaylı bilgi verilerek, Zhao ve diğ. (2009)'nıın çalışmasındaki eksiklikler göz önünde bulundurularak en büyük aralık ile öbekleme yöntemi çok kernelli Yarı – Sonsuz Programlama (YSP) ile modellenmiştir. Çoklu kernel ile öbekleme ya da sınıflandırma yaparken kernellerin alacağı katsayıların, ($\boldsymbol{\beta}_\kappa$) vektörel yapısı algoritmamın verimliliği ve hızı açısından önem teşkil etmektedir. Bu tezde Zhao ve diğ. (2009)'den farklı olarak kernel katsayıları $l_1$norm'a kısıtlanarak, seyrek hale getirilmiş, aynı zamanda ağırlık vektörleri amaç fonksiyonunda kendi içinde $l_2$ norm'a cezalandırılmıştır.

Bu tezin literatüre ikinci bir katkısı ise öbekleme probleminin YSP ile şimdiye kadar modellenmemiş olmasıdır. Literatürde çok kernelli öbekleme algoritmaları yarı tanımlı programlama (*semi definite programming*) (Boyd & Vandenberghe 2004) ve kesen düzlem (*cutting plane*) algoritması ile çözülmüştür (Zhao ve diğ. 2009). Yarı tanımlı programlama (*semi definite programming*) sınıflandırma programlarında da kullanılmış fakat YSP'nin çoklu kernel ile sınıflandırma problemlerinde daha iyi sonuçlar verdiği ispatlanmıştır (Sonnenburg 2006, Bach 2004). Bu nedenle bu tezde, en büyük aralık ile çok kernelli öbekleme YSP ile modellenmiş ve optimizasyon probleminin en iyilik koşulları olabilecek bütün durumlar göz onunda bulundurularak, her durum için ayrı ayrı incelenmiş, LBKK, KKT ve IUK (SOC) analizleri verilmiştir. İncelenen her koşul için Bölüm 2'de verilen Teorem 2.3.9 gerek koşulu Ansatz azaltıcı maddelerinin IUK birinci durumu hariç, sağlanmış olup, yarı sonsuz olarak modellenen problem sonlu optimizasyon problemine dönüşmüştür.

Algoritma 1 ile verilen yöntemin reel veri üzerinde uygulanması ve diğer öbekleme algoritmaları ile karşılaştırılması gelecek çalışmaya bırakılmıştır.

# KAYNAKÇA

## *Kitaplar*


Alpaydın, E., 2004. *Introduction to Machine Learning (Adaptive Computation and MachineLearning)*. Cambridge: Mass. MIT Press,

Anderberg, M. R., 1973. *Cluster Analysis for Applications*. New York: Academic Press.

Apostol, T., 1967. *Calculus*, John Wiley & Sons, 1. cilt, s.189.

Berberian, S., 1994. *A First Course in Real Analysis*. Springer-Verlag, s.128.

Bertsekas, D.P., 1999. *Nonlinear Programming*, Second Edition. Nashua, NH: Athena Scientific. ISBN: 1-886529-00-0

Bertsimas, D. & Tsitsiklis, J.N., 1997. *Introduction to Linear Optimization*. Athena Scientific Series, Belmont, Massachusetts. ISBN: 978-1-886529-19-9

Boyd, S. & Vandenberghe, L., 2004. *Convex Optimization*. Cambridge: Cambridge University Press.

Broomhead, D.S., Lowe, D., 1988. *Radial basis – functions, multi – variable functional interpolation and adaptive networks*. Rsre Memorandum 4148. Royal signals & radar establishment. Auithors: Broomnhead D. S. and Lowe D.

Bulgak, A., Bulgak, H., 2001. *Lineer Cebir*. Konya, Türkiye: Uygulamalı Matematik Araştırma Merkezi Yayınları, Selçuk Üniversitesi Vakfı.

Chiang, A. C, 1994, *Fundamental Methods of Mathematical Economics*, McGraw-Hill, Inc.,s.317. ISBN: 0-07-010813-7.

Cherkassky, V. & Muller F., 1998. *Learning from Data: Concepts, Theory and Methods.* New York: John Wiley & Sons.

Christodoulos, A. F., 1995. *Nonlinear and Mixed-Integer Optimization Fundamentals and Applications.* New York : Oxford Press.

Chong, E.K.P. & Zak, S.H. 2001. *An Introduction to Optimization.* Second Edition, New York: John Wiley & Sons, Intersicence Series in Discrete Mathematics and Optimization.

Cristianini, N. & Taylor, J.S., 2000. *An Introduction to Support Vector Machines and other kernel – based learning methods.* UK: Cambridge, Cambridge University press.

Dattorro, J., 2005. *Convex Optimization & Euclidean Distance Geometry,* USA: California, Meboo Publishing.

Duda, R. O., Hart, P.E. & Stork D.G., 2001. *Pattern Classification.* Second Edition, John Wiley & Sons– Interscience Publication. ISBN: 0-471-39126-3.

Dunham, M. H., 2003. *Data Mining Introductory and Advanced Topics*. Pearson Education, Inc., , New Jersey: Upper Saddle River.

Elmas, Ç., 2003. *Yapay Sinir Ağları (Kuram, Mimari, Eğitim, Uygulama).* İstanbul: Seçkin Yayıncılık.

Han, J. & Kamber, M., 2006. *Data Mining: Concepts and Techniques*. Second Edition, USA, San Francisco: Morgan Kaufmann Publishers Inc.

Hartigan, J. A., 1975. *Clustering Algorithms.* New York: John Wiley & Sons Inc., ISBN: 0-471-35645-X, Hebb, D., 1949. *The organization of behavior*. New York: Wiley & Sons.

Hettich, R. & Jongen, H.Th 1978. *Semi-infinite programming: Conditions of optimality and applications*. In J. Stoer, editor, Optimization Techniques, Lecture notes in Control and Information Sciences. Springer, Berlin: Heidelberg.

Hettich, R. & Zencke, P., 1982. *Numerische Methoden der Approximation und semi-infiniten Optimierung*. Teubner Studienbücher Mathematik, Stuttgart.

Jain, A. K. & Dubes, R. C., 1988. *Algorithms for Clustering Data*. New Jersey: Prentice Hall, Englewood Cliffs. ISBN:0-13-022278-X.

Jardine, N. & Sibson, R., 1971. *Mathematical Taxonomy*. London: Wiley.

Joachims, T., 1999. *Making large-scale SVM learning practical*. In Schölkopf, Burges, B., C. J. C. and Smola, A. J. (eds.), *Advances in Kernel Methods—Support Vector Learning*, Cambridge: MA, MIT Press. pp. 169—184.

Kaufman, L. & Rousseeuw, P. J., 1990. *Finding Groups in Data: An Introduction to Cluster Analysis*. New Jersey: John Wiley & Sons, Inc.

Kohonen, T., 1984. *Self – Organization and Associative Memory*. Berlin: Springer-Verlag. MIT Press.

Kohonen, T., 2001. *Self-Organizing Maps*. Germany: Springer – Verlag Berlin Heidelberg.

Kolman, B., Hill, D. R., 2002. *Uygulamalı Lineer Cebir*. Ö. Akın (Çev.), Ankara: Palme Yayıncılık.

Luenberger, D.G. & Ye, Y., 2007. *Linear and Nonlinear Programming*. International Series in Operations Research & Management Science. Third Edition, New York: Springer.

Mangasarian, O.L., 1994. *Nonlinear Programming*, New York: SIAM. p.39.

Mitchell, T., 1997. *Machine Learning*. USA: WCB / McGraw – Hill Series in Computer Science / Engineering / Math. p. 367.

Nabiyev, V. V., 2005. *Yapay Zekâ - Problemler-Yöntemler- Algoritmalar*, Ankara : Seçkin Yayıncılık.

Nash, S. & Sofer A., 1996. *Linear and nonlinear programming*. New York: McGraw-Hill, Inc.

Nicholson, W. 1995. *Microeconomic Theory, Basic Principles and Extensions*. 6nd ed. USA: The Dryden Press, Fort Worth.

Nilson, N.J., 1965. *Learning Machines.* New York: McGraw - Hill

Nocedal, J. & Wright, S.J., 1999. *Numerical Optimization.* New York: Springer-Verlag.

Özdamar, K., 2004. *Paket programlama ile İstatistiksel Veri Analizi (Çok Değişkenli Analizler).* Eskişehir: Kaan Kitapevi, s.502.

Öztemel, E., 2003. *Yapay Sinir Ağları.* İstanbul: Papatya Yayıncılık.

Pöyhönen, S., 2004. *Support Vector Machine Based Classification in Condition Monitoring of Induction Motors*. Finland: Helsinki University of Technology Control Engineering Laboratory.

Rosenblatt, F., 1959. *Principles of Neuradynamics,* New York: Spartan Books.

Simon, H.A., 1983. *Why Should Machines Learn?* In: Machine Learning – An Artificial Intelligence Approach. Michalski, R. S., Carbonell, J.G., & Mitchell, T.M. (Ed). Palo Alto, California: Tioga

Still, G., 2004. *Semi-infinite programming*: An introduction, preliminary version. Technicalreport, University of Twente Department of Applied Mathematics, P.O.Box 217 7500 AE Enschede, The Netherlands.

Sokal, R.R. & Sneath P.H.A., 1963. *Principles of numerical taxonomy*. San Francisco: &c., W. H. Freeman & Company.

Tryon, R. C. & Bailey, D. E, 1970. *Cluster Analysis*, New York: McGraw-Hill Book Company.

Vajda, S., 1974, Theory of Linear and Nonlinear Programming**,** London: Longman. s.4.

Vapnik, V. N., 2000. *The Nature of Statistical Learning Theory*. Second Ed., New York: Springer.

Weber, G.-W.,1992. *Charakterisierung struktureller stabilit at in der nichtlinearen optimierung.* In H.Th. Jongen H.H. Bock andW. Plesken, editors, *Aachener Beitraege zur Mathematik 5*. Augustinus publishing house (now: Mainz publishing house), Aachen.

Wetterling, W.W.E., 1970. *Definit the its bedingungen f¨ur relative extrema bei optimierungsund approximation* saufgaben. *Numer. Math.*, 12 pp.122–136.

Witten, I. H. & Frank, E., 1999. *Data Mining: Practical machine learning tools with Java implementations.* San Francisco: Morgan Kaufmann.

***Süreli Yayınlar***


Alan, M. A. ve Yeşilyurt, C., 1994. Doğrusal Programlama Problemlerinin Excel ile Çözümü. *Cumhuriyet Üniversitesi İktisadi ve İdari Bilimler Dergisi*, **5**, (1)

Aldenderfer, M. S. & Blashfield, R. K., 1984. Cluster Analysis, *Sage University Paper series on Quantitative Applications in the Social Science*, series no 07 – 044. Newbury Park, California: Sage Publications.a

Anderson, J. A., 1972. A simple neural network generating on interactive memory. *Mathematical Biosciences,* **14**, pp. 197-220

Bach F., Lanckriet G. R. G. & Jordan M. I., 2004. Multiple Kernel Learning, Conic Duality, and the SMO Algorithm. *Proceedings of the Twenty-first International Conference on Machine Learning.*

Ben-Hur A., Horn D., Siegelmann, H. T., Vapnik, V., 2001. Support Vector Clustering. *Journal of Machine Learning Research*, **2** , pp.125-137.

Boutsinas, B. & Gnardellis, T., 2002. On Distributing the Clustering Process. Pattern Recognition Letters, *Elsevier Science Publishers B.V.*, **23** (8), pp. 999-1008.

Burl, M.C., Fayyad, U. M., Perona, P., Smyth, P. & Burl, M.P. 1994. Automating the hunt for volcanes on venus. *In Procee*

Chen, W. H. & Shih, J. Y., 2006. A study of Taiwan’s issuer credit rating systems using support vector machines. *ELSEVIER Expert Systems with Applications* **30,** pp. 427-435.

Chiang, L. H., Kotanchek, M. E., & Kordon A. K., 2004. Fault diagnosis based on Fisher discriminant analysis and support vector machines. *Computers and Chemical Engineering* **28** (8), pp.1389–1401.

Çomak, E., Arslan, A. ve Türkoğlu, İ., 2007. A decision support system based on support vector machines for diagnosis of the heart valve diseases. *Computers in Biology and Medicine* **37,** ss. 21-27.

Çomak, E., Polat, K., Güneş, S. ve Arslan, A., 2007. A new medical decision making system: Least square support vector machine (LSSVM) with Fuzzy Weighting Preprocessing. Expert Systems with Applications **32** (2), ss. 409-414.

Du, C. J. & Sun, D. W., 2005. Pizza sauce spread classification using colour vision and support vector machines. *Journal of Food Engineering* **66**, pp.137–145.

Eltoft, T. & Figueiredo, R. J. P., 1998. A New Neural Network for Cluster-Detection and- Labeling. *IEEE Transactions on Neural Networks*, **9** (5), pp.1021-1035.

Fan, R.- E., Chen P.-H. & Lin C. – J., 2005. Working set selection using the second order information for training support vector machines. *Journal of Machine Learning Research*, **6**, pp.1889–1918.

Gönen M. ve Alpaydın E., 2008. Localized multiple kernel learning. In ICML ’08: Proceedings of the 25th international conference on Machine learning, ss. 352–359, New York, NY, USA. ACM. ISBN 978-1-60558-205-4. [online]: http://doi.acm.org/10.1145/1390156.1390201.

Grossberg, S., 1986. *The Adaptive Brain I:Cognition, Learning, Reinforcement, and Rhytm, and The Adaptive Brain.* II:Visions, Speech, Language, and Motor Control, Elsevier/North-Holland, Amsterdam.

Hettich, R. & Kortanek, O., 1993. Semi-infinite programming: Theory, methods and applications. *SIAM Review*, **35** (3), pp. 380–429

Hopfield, J.J. 1982. Neural networks and physical systems with emergent collective computational abilities. *Proceedings of the National Academy of Sciences, USA*, Biophysics, **79**, pp. 2554 – 2558.

Hopfield, J.J. 1982. Neurons with graded response have collective computational properties like those of two state neurons. *Proceedings of the National Academy of Science USA*, **81**, pp. 3088-3092.

Jain, A. K., Murty, M. N. & Flynn, P. J., 1999. Data Clustering: A Review. *ACM Computing Surveys*. **31** (3), pp. 264-323.

Karypis G., Han E. H. & Kumar V., 1999. CHAMELEON: A Hierarchical Clustering Algorithm Using Dynamic Modeling, *IEEE Computer* **32** (8), pp. 68-75.

Kloft M., Brefeld U., Sonnenburg S. & Zien A., 2011. $l_p$ *– Norm Multiple Kernel Learning* Journal of Machine Learning Research **12**, pp. 953-997.

Kocabaş, Ş., 1991. *A Review of learning*. The Knowledge Engineering Review, **6** (3), ss. 195-222.

Kohonen, T., 1972. Correlation matrix memories. *IEEE Transaction on Computers,* C-**21** (4), pp. 353-359.

Kohonen, T., 1982. Self – organazied formation of topologically correct feature maps. *Biological Cybernetics.* **43** (1), pp. 59-69.

Lanckriet, G., Bie, T. De, Cristianini, N., Jordan, M. & Noble, W., 2004. A statistical framework for genomic data fusion. *Bioinfomatics*, **20** (16), pp. 2626-2635

Lanckriet, G., Cristianini, N., Ghaoui, L., Bartlett, P. & Jordan, M., 2004. Learning the kernel matrix with semidefinite programming. *JMLR*, **5**, pp. 27-72.

Lobo, M. S., Vandenberghe, L., Boyd, S., & L´ebret, H., 1998. Applications of second-order cone programming. *Elsevier, Linear Algebra and its Applications*. **284** (1-3), pp. 193–228.

Melgani, F., Bruzzone, L., 2004. Classification of Hyperspectral Remote-Sensing Images with Support Vector Machines. *IEEE Transactions on Geoscience and Remote Sensing*, **42** (8), pp. 1778-1790.

Nemirovski, A., 2004. Prox-method with rate of convergence $O(1/T)$ for variational inequalities with lipschitz continuous monotone operators and smooth convex-concave saddle point problems. *SIAM Journal on Optimization*, **15**, pp.229–251.

Özöğür-Akyüz, S. & Weber, G.-W., 2010. Infinite Kernel Learning via infinite and semi-infinite programming. *In the special issue of OMS (Optimization Methods and Software).* **25** (6), pp. 937-970.

Özöğür-Akyüz, S., ve Weber, G.-W., 2010. On Numerical Optimization theory of Infinite Kernel Learning, *Journal of Global Optimization*. **48** (2), pp. 215 -239.

Samanta, B., 2004. *Gear fault Detection Using Artificial Neural Network and Support Vector Machines with Genetic Algorithms*. Mechanical Systems and SignalProcessing, **18** (3), pp. 625-644.

Sonnenburg, S., Ratsch, G. & Schafer, C., 2005. Learning Interpretable SVMs for Biological Sequence Classification. In *RECOMB, LNBI 3500*, pp. 389–407. Berlin Heidelberg: Springer-Verlag.

Sonnenburg, S., Ratsch, G., Schafer, C. & B. Scholkopf., 2006. Large scale multiple kernel learning. *Journal of Machine Learning Research*, **7**, pp.1531–1565.

Specht, D.F., 1988. Probabilistic neural networks for classification, mapping, or associative memory. *IEEE Conference on Neural Neetworks*, **1**, San Diego, pp.525-532.

Specht, D.F., 1991. A general regression neural network. *IEEE Transactions on Neural Networks*, **2** (6), pp. 568-576.

Übeyli, E. D., 2007. *Combining eigenvector methods and support vector machines for detecting variability of Doppler ultrasound signals*. Computer Methods and Programs in Biomedicine **86** (2). ss. 181-190.

Vapnik, V. N. & Chervonenkis A.Ya., 1968. On the uniform convergence of relative frequencies of events to their probabilities. *Soviet Math. Dokl*. **9**, pp.915-918.

Weber, G.W., 2003. Generalized Semi-Infinite Optimization and Related Topics. *Research and Exposition in Mathematics*. **29**, Heldermann Verlag, Germany.

Xu Z., Jin R., King I., ve Lyu. M,, 2009. *An extended level method for efficient multiple kernel learning*. In D. Koller, D. Schuurmans, Y. Bengio, and L. Bottou, editors, *Advances in Neural Information Processing Systems 21*, pp. 1825–1832.

***Diğer Yayınlar***

http://www.ahmetkakici.com/yapay-sinir-aglari/yapay-sinir-aglarinin-siniflandirilmasi/ [Erişim tarihi: 08/01/2012]

http://www.google.com.tr/books?hl=tr&lr=&id=sna9BaxVbj8C&oi=fnd&pg=PR7&dq=Vapnik,+The+Nature+of+Statistical+Learning+Theory&ots=onIcPSood7&sig=WXAsmu_3eIJHvdWmfTGsItKvJ4&redir_esc=y#v=onepage&q=Vapnik%2C%20The%20Nature%20of%20Statistical%20Learning%20Theory&f=false [Erişim tarihi: 23/12/2011]

Bilgin T., (2003). Veri Madenciliğinde Kümeleme Analizi Yöntemi Uygulaması. *Yüksek Lisans Tezi.* Marmara Üniversitesi, Fen Bilimleri Enstitüsü, Bilgisayar ve Kontrol Eğitimi, İstanbul.

Berkhin, P., 2002. *Survey of Clustering Data Mining Techniques.* In: Accrue Software, Technical Report, San Jose, California: USA.

Burbidge, R. & Buxton, B. 2001. An Introduction to Support Vector Machines for Data Mining. *Technical Report*. Young OR 12, University of Nottingham, 3-15. http://datamining.martinsewell.com/BuBu.pdf pp.5-10

Kaymaz İ., Atatürk Ünv. Müh. Fak. Ders Notları Bölüm3 s. 3

Mercer D. P., (2003), ”Clustering Large Datasets”, http://www.stats.ox.ac.uk/~mercer/documents/transfer.pdf, [Erişim tarihi: 13/01/2012]

Rubinstein E., (2005). Support vector machines via advanced optimization techniques. *Master’s thesis*, Faculty of Electrical Engineering, Technion.

Shen, J., Pei, Z. J., Lee, E. S., 2004. Support Vector Regression in the Analysis of Soft-Pad Grinding of Wire-Sawn Silicon Wafers. CITSA 2004/ISAS, *International*

*Conference on Cybernetics and Information Technologies, Systems and Applications/The 10th International Conference on Information Systems Analysis and Synthesis*, pp. 19-24.

Shen, J., (2005). Fusing Support Vector Machines and Soft Computing for Pattern Recognition and Regression. *Doktora Tezi*. Kansas: Kansas State University Manhattan.

Tolun S., (2008). Destek Vektör Makineleri: Banka Başarısızlığının Tahmini üzerine Bir Uygulama. *Doktora Tezi*. İstanbul: İstanbul Üniversitesi Sosyal Bilimler Enstitüsü.

http://www.webhatti.com/egitim-kultur/107273-optimizasyon-nedir.html [Erişim tarihi: 23/12/2011]

Zhao, B., Kwok, J. & Zhang, C., 2009. Multiple Kernel Clustering. *Proceedings of the SIAM International Conference on Data Mining*, SDM. Sparks, Nevada, USA. SIAM. pp.638-649.

Zien A., 2008, Multiple Kernel Learning, Summer School on Neural Networks, Fraunhofer FIRST.IDA, Berlin, Germany, p. 35

## ÖZGEÇMİŞ

**Adı Soyadı** : Gülnur SEICHAN OGLOU

**Sürekli Adresi** : Cihannüma Mah. Cihannüma Sok. Kılıç I Apt. 8/5 Beşiktaş – İST

**Doğum Yeri ve Yılı** : İstanbul – 1979

**Yabancı Dili** : İngilizce

**İlk Öğretim** : Bakırköy Ortaokulu – 1992

**Orta Öğretim** : Bakırköy Lisesi – 1995

**Lisans** : İ.Ü. Fen Fakültesi Matematik Bölümü – 1999

**Yüksek Lisans** : Bahçeşehir Üniversitesi

**Enstitü Adı** : Fen Bilimleri Enst.

**Program Adı** : Uygulamalı Matematik

**Çalışma Hayatı** :

| | | |
|---|---|---|
| 30/03/2010 – ….. | İSOV – Dinçkök Teknik ve EML | (Matematik Öğretmenliği) |
| 27/10/2009 – 29/03/2010 | İSOV – Dinçkök Anadolu Teknik Lisesi | (Müdür Yardımcılığı) |
| 11/09/2008 – 26/10/2009 | İSOV – Dinçkök Anadolu Teknik Lisesi | (Matematik Öğretmenliği) |
| 02/09/2008 – 11/09/2008 | Kırımlı İsmail Rüştü Olcay Lisesi | (Matematik Öğretmenliği) |
| 19/09/2003 – 02/09/2008 | Bahçelievler Kocasinan Teknik ve EML. | (Matematik Öğretmenliği) |
| 04/10/2000 – 19/09/2003 | Bahçelievler Kazım Beyaz Mesleki Eğitim Merkezi | (Matematik Öğrt) |