

**YILDIZ TEKNİK ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ**

**TANECİKLİ HESAPLAMA YÖNTEMİ İLE KAVRAM
ÇÖZÜMLEME**

Mat. Yük. Müh. Mert BAL

**F.B.E Matematik Mühendisliği Anabilim Dalında
Hazırlanan**

DOKTORA TEZİ

Tez Savunma Tarihi: 15 Şubat 2008

Tez Danışmanı : Prof. Dr. Oya KALIPSIZ (Yıldız Teknik Üniversitesi)

Jüri Üyeleri : Prof. Dr. Eşref ADALI (İstanbul Teknik Üniversitesi)

: Prof. Dr. Mustafa SİVRİ (Yıldız Teknik Üniversitesi)

: Prof. Dr. A.Coşkun SÖNMEZ (Yıldız Teknik Üniversitesi)

: Prof. Dr. Behiç ÇAĞAL (Kültür Üniversitesi)

İSTANBUL, 2008

İÇİNDEKİLER

	Sayfa
SİMGE LİSTESİ	v
KISALTIMA LİSTESİ.....	vi
ŞEKİL LİSTESİ	vii
ÇİZELGE LİSTESİ	viii
ÖNSÖZ.....	ix
ÖZET	x
ABSTRACT	xi
1. GİRİŞ.....	1
2. BAYES AĞLARI (BAYES İNANÇ AĞLARI, OLASILIKSAL AĞLAR).....	6
2.1 Bayes Ağları Hakkında Ön Bilgiler.....	10
2.1.1 Çizge Kuramının Bazı Temel Kavram ve Tanımları.....	11
2.1.2 Olasılık Kuramının Bazı Temel Kavram ve Tanımları	14
2.1.2.1 Olasılık Ölçüsü	14
2.1.2.2 Olasılık Dağılımları	16
2.1.2.2.1 Koşullu Olasılık	16
2.1.2.3 Bağımlılık ve Bağımsızlık	17
2.1.2.3.1 İki Değişkenin Bağımsızlığı	17
2.1.2.3.2 Değişkenler Kümesinin Bağımsızlığı.....	18
2.1.2.3.3 Koşullu Bağımlılık ve Bağımsızlık	18
2.1.2.4 Bayes Teoremi	19
2.2 Bayes Ağlarının Oluşturulması	20
2.2.1 Bayes Ağ Tabanlı Teşhis Modeli	22
2.3 Bayes Ağlarında Çıkarım Yöntemleri	23
2.3.1 Tam (Kesin) Çıkarım.....	23
2.3.2 Yaklaşık Çıkarım	29
2.3.2.1 Stokastik Benzetim Algoritmaları	29
2.3.2.2 Model Basitleştirme Yöntemleri.....	31
2.3.2.3 Arama Tabanlı Yöntemler	31
2.3.2.4 Döngülü İnanç Yayılımı	31
3. TANECİKLİ HESAPLAMA YÖNTEMİ ve KABA KÜMELER KURAMI	32
3.1 Tanecikli Hesaplama Yöntemi	32
3.1.1 Tanecikli Hesaplamanın Temel Kavramları.....	32
3.1.2 Tanecikli Hesaplama ve Veri Madenciliği	33
3.1.2.1 Bilgi Tanecikleri	35
3.1.2.2 Yapısal Bilgi	36
3.1.2.3 Madenleme Görevi	36
3.2 Kaba Kümeler Kuramı.....	37
3.2.1 Kaba Kümeler Kuramında Temel Kavramlar.....	39

3.2.1.1	Çapraşıklık ve Sınır Bölgesi	39
3.2.1.2	Belirsizlik ve Kaba Üyelik Fonksiyonu.....	40
3.2.1.3	Çapraşık Kavramların Sınıflandırılması	43
3.2.1.4	Bilgi Sistemleri	44
3.2.1.5	Ayrıtedilemezlik	45
3.2.1.6	Ayrıtedilebilirlik Matrisi.....	46
3.2.1.7	Ayrıtedilebilirlik Fonksiyonu	46
3.2.2	Küme Yaklaşımları	47
3.2.2.1	Kavramların Yakınlık Ölçüleri	50
3.2.2.1.1	Yaklaşımın Doğruluğu.....	50
3.2.2.1.2	Pozitif Bölge ve Ölçüsü	51
3.2.2.1.3	Kaba Üyelik Fonksiyonu	51
3.2.2.1.4	Özelliklerin Bağımlılığı	51
3.2.3	İndirgenmiş Özellik Kümeleri	52
3.2.4	Karar Sistemleri ve Karar Kuralları.....	53
3.3	Kaba Kümeler Kuramı ve Bayes Ağları Arasındaki İlişki	54
3.3.1	Akış Çizgeleri	54
3.3.2	Kesinlik ve Örtme Faktörleri	56
3.3.3	Karar Algoritmaları	58
3.3.3.1	Karar Algoritmalarının Özellikleri	60
3.3.4	Akış Çizgelerindeki Bağımlılıklar.....	61
3.3.5	Karar Yüzeyi.....	62
4.	BİÇİMSEL KAVRAM ÇÖZÜMLEME.....	63
4.1	Biçimsel Kavram Çözümlemede Temel Kavramlar.....	64
4.2	Problemin Formüle Edilmesi	67
4.3	Tanımlanabilir Kavramlar	68
4.4	Tanımlanamaz Kavramlar	69
4.5	Tanımlanabilir Kavramların Matematiksel Özellikleri	73
4.6	Kavram Yaklaşımlarına Kaba Kümeler Yaklaşımı	73
4.6.1	Mevcut Yaklaşım.....	74
4.6.2	Nesneler Kümesine Yaklaşım.....	75
4.6.3	Özellikler Kümesine Yaklaşım.....	77
4.6.4	Bir Tanımlanamayan Kavrama Yaklaşım	78
5.	BELİRTİ-HASTALIK İLİŞKİSİNİN KAVRAM KAFESİ ile MODELLENMESİ ve DENEYSEL ÇALIŞMA.....	82
5.1	Belirti-Hastalık İlişkisinin Kavram Kafesi ile Modellenmesi	82
5.1.1	Kavram Kafesinin Oluşturulması	84
5.1.2	Kafes Yapısındaki Olasılıkların Hesaplanması	85
5.2	Veri Kümesi ve ALARM Ağ Yapısı	86
5.3	Özellik Verilerinin İndirgenmesinde Kaba Kümelerin Kullanımı	90
5.4	DeneySEL Çalışma	92
5.4.1	Kafes Tabanlı Modelin Makine Öğrenme Yöntemleriyle Karşılaştırılması.....	94
6.	SONUÇ ve DEĞERLENDİRME.....	96
	KAYNAKLAR.....	98
	EKLER.....	105

Ek 1 Modeling the Symptom-Disease Relationship by Using Rough Set Theory and Formal Concept Analysis.....	106
Ek 2 Veri Madenciliğinde Birliktelik Kuralları.....	107
ÖZGEÇMİŞ.....	108

SİMGE LİSTESİ

p	Olasılık ölçümü
V	Bayes ağındaki düğümlerin kümesi
E	Bayes ağındaki kenarların kümesi
G	Yönlü döngüsüz çizge
$Adj(V_i)$	V_i 'nin komşuluk kümesi
S	V düğümler kümesinin bir alt kümesi
C	Düğümlerin bir tam kümesi
$Nbr(V_i)$	V_i düğümüne komşu olan düğümlerin kümesi
$Bnd(S)$	S düğümlerinin sınırı
S	Örnek uzay
A	Atomik önerme kümesi
X_i	Ayrık rasgele değişkenler kümesi
x_i	Ayrık rasgele değişkenler kümesinin olası örnekler kümesi
U	Evrensel küme
I	Evrensel küme üzerinde ikili bağıntı
A	Bilgi sistemi
$\mu_x^l(x)$	Kaba üyelik fonksiyonu
A	Bilgi sisteminde özelliklerin boş olmayan sonlu bir kümesi
a	Boş olmayan sonlu özellikler kümesinin bir elemanı
d	Karar özelliği
V_a	a Özelliklerinin değer kümesi
M_A	Ayırtedilebilirlik matrisi
f_A	Ayırtedilebilirlik fonksiyonu
ε	Değişken duyarlı kaba küme modelinde eşik değeri
$IND_A(B)$	B – ayırtedilemezlik bağıntısı
$[x]_B$	B – ayırtedilemezlik bağıntısının denklik bağıntıları
$X_A^{r(d)}$	A Bilgi sisteminin karar sınıfı
$\alpha_B(X)$	Kaba kümeler kuramında yaklaşım doğruluğunun ölçüsü
ζ	B ve V 'deki formüller kümesinin bir elemanı
N	Akış çizgesindeki düğümlerin kümesi
\mathcal{B}	Akış çizgesindeki yönlü dalların kümesi
Φ	Bir karar kuralının koşul kümesi
Ψ	Bir karar kuralının karar kümesi
$\varphi(x, y)$	x 'den y 'ye bir akış yolu
σ	(x, y) 'nin normalize edilmiş akışı
$\delta(x, y)$	n boyutlu karar uzayında x ve y noktaları arasındaki mesafe
G	Nesneler kümesi
M	Özellikler kümesi
g	G nesneler kümesinin bir elemanı
m	M özellikler kümesinin bir elemanı
A	G nesneler kümesinin bir alt kümesi
B	M özellikler kümesinin bir alt kümesi
(β, α)	Biçimsel içeriğinde bir Galois ikilisi

KISALTMA LİSTESİ

ALARM	A Logical Alarm Reduction Mechanism
BA	Bayes Ağları (Bayesian Networks)
BKÇ	Biçimsel Kavram Çözümleme (Formal Concept Analysis)
ÇKY	Çok Katmanlı Yayılım (Multi Layer Perceptron)
DVM	Destek Vektör Makinesi (Support Vector Machine)
GA	Genetik Algoritma (Genetic Algorithm)
KDS	Karar Destek Sistemleri (Decision Support Systems)
TH	Tanecikli Hesaplama (Granular Computing)
TKDS	Tıbbi Karar Destek Sistemleri (Medical Decision Support Systems)
VM	Veri Madenciliği (Data Mining)
VTBK	Veri Tabanlarından Bilgi Keşfi (Knowledge Discovery in Databases)
YDÇ	Yönlü Döngüsüz Çizge (Directed Acyclic Graph)
YSA	Yapay Sinir Ağları (Artificial Neural Networks)
YZ	Yapay Zekâ (Artificial Intelligence)
ZVA	Zeki Veri Analizi (Intelligent Data Analysis)

ŞEKİL LİSTESİ

	Sayfa
Şekil 2.1	Bir Bayes ağı oluşturma adımları 8
Şekil 2.2	Dinamik Bayes ağının şematik olarak gösterilmesi..... 9
Şekil 2.3	Dinamik Bayes ağı için gerekli uygulama adımları 10
Şekil 2.4	Bir çizge örneği..... 11
Şekil 2.5	Üçgenleştirilmiş bir çizge 14
Şekil 2.6	Üçgenleştirilmemiş bir çizge 14
Şekil 2.7	Örnek bir Bayes ağı 21
Şekil 2.8	Klasik bir teşhis problemini modelleyen bir Bayes ağı örneği..... 22
Şekil 2.9	Bir G ikili çizge örneği..... 25
Şekil 2.10	G çizgesi için üçgenleştirilmiş H moral çizge örneği..... 25
Şekil 2.11	H çizgesi için klik ağacı örneği 27
Şekil 3.1	Bir kaba küme örneği..... 48
Şekil 3.2	Bir kaba kümenin pozitif, negatif ve sınır bölgeleri 50
Şekil 4.1	Çizelge 4.1' de verilen içeriğin kavram kafesi..... 68
Şekil 4.2	Kavramlara genel bir bakış..... 69
Şekil 4.3	Algoritma 1: Bir A nesnelere kümesine yaklaşım için sözde kod 76
Şekil 4.4	Algoritma 2: Bir B özellikler kümesine bir yaklaşım için sözde kod 78
Şekil 4.5	Algoritma 3: Bir tanımlanamaz (A, B) kavramına yaklaşım için sözde kod 81
Şekil 5.1	Belirti-Hastalık bağlamına ait kafes yapısı..... 83
Şekil 5.2	Çizgedeki düğümlerin bulunması için algoritma..... 84
Şekil 5.3	Çizgedeki kenarların bulunması için algoritma 85
Şekil 5.4	ALARM ağ yapısı ve ağda tanımlı değişkenler 88
Şekil 5.5	Alarm.dnet Catechol değişkeni için koşullu olasılık çizelgesi 90
Şekil 5.6	ROSETTA süreci..... 91

ÇİZELGE LİSTESİ

	Sayfa
Çizelge 4.1 Bir bağlam örneği	65
Çizelge 5.1 Belirti-Hastalık ilişkisi için bağlam	82
Çizelge 5.2 10 kayıtlık veri kümesine ait bir çıktı kümesi.....	92
Çizelge 5.3 100 kayıtlık veri kümesine ait bir çıktı kümesi.....	93
Çizelge 5.4 1000 kayıtlık veri kümesine ait bir çıktı kümesi.....	93
Çizelge 5.5 2000 kayıtlık veri kümesine ait bir çıktı kümesi.....	94
Çizelge 5.6 Geliştirilen modele ait test sonuçları.....	94
Çizelge 5.7 Kafes tabanlı yöntemin diğer makine öğrenme yöntemleri ile karşılaştırılması	95

ÖNSÖZ

Bu tez çalışmam sırasında benden desteğini ve yardımlarını hiçbir zaman esirgemeyen çok değerli hocalarım ve tez danışmanlarım Sayın Prof. Dr. Oya KALIPSIZ ve Sayın Prof. Dr. Hayri SEVER' e en içten teşekkürlerimi sunarım.

Akademik hayatımda bana destek olan Sayın Prof. Dr. Durul ÖREN, Sayın Prof. Dr. Sabriye PİŞKİN, Sayın Prof. Dr. Murat DEMİRCİOĞLU ve Sayın Prof. Dr. Coşkun SÖNMEZ' e çok teşekkür ederim.

Tez çalışmam sırasında değerli görüşlerini ve desteklerini aldığım Sayın Öğr. Gör. Güven KÖSE ve Sayın Uğur Kâşif BOYACI' ya teşekkürü bir borç bilirim.

Tüm yaşamım boyunca gerek maddi gerekse manevi olarak her zaman bana destek olan sevgili annem, babam ve kardeşime en içten teşekkürlerimi ve sevgilerimi sunarım.

Tez çalışmalarım boyunca benden desteklerini esirgemeyen eşimin anne ve babasına çok teşekkür ederim.

Kendisini tanıdığım ilk andan itibaren sevgisi ve sabrı ile daima yanımda olan, akademik yaşamımda bana güç vererek, beni cesaretlendiren ve tez çalışmalarım boyunca yardımlarını aldığım çok değerli eşim Sayın Yasemin BAL' a kalbimin derinliklerinden en içten teşekkürlerimi ve sevgilerimi sunarım.

Mert BAL

Şubat 2008

ÖZET

Tıbbi Karar Destek Sistemleri (TKDS) eksik bilgi ve belirsizlik altında çıkarsama yapabilen akıllı yazılım sistemleridir. Bu sistemlerde belirsizliğin modellenmesi için Bayes ağları, kaba kümeler, yapay sinir ağları, bulanık mantık, tümevarımsal mantık programlama, genetik algoritmalar gibi çeşitli esnek hesaplama yöntemleri ile bu yöntemlerin birkaçının birleşiminden oluşan melez yöntemler kullanılmaktadır. Bayes ağları TKDS' de sıklıkla kullanılan veri güdümlü bir yöntemdir.

Bu çalışmada, Bayes ağları ile Biçimsel Kavram Çözümlemedeki kavram kafes yapısı arasındaki benzerliklerden yola çıkılarak Belirti-Hastalık arasındaki ilişkileri yansıtan yeni bir veri güdümlü model önerilmiştir. Bu modelde hastalıklar nesnelere ve belirtiler de özellikler olarak modellenmiştir.

Geliştirilen modelin test edilmesi için veri kümesi bulmadaki teknik problemler nedeniyle gerçek durumlardan çıkarılmış ve gerçek hasta verileri kullanılarak hazırlanmış olan ALARM ağ yapısı kullanılmıştır. Çeşitli boyutlardaki bu veri kümelerinin üretilmesi NETICA yazılımı ile gerçekleştirilmiştir. Hesaplama karmaşıklığını önlemek amacıyla, özellik verilerine kaba kümeler kuramının temelini oluşturan ve ayırtedilebilirliğe dayalı bir yazılım sistemi olan ROSETTA kullanılmıştır.

Belirti-Hastalık bağlamının, ilgili algoritmalar ile kavram kafesi yapısı oluşturulmuştur. Kavram kafesindeki kenarlar, belirtiler ve hastalıklar arasındaki olasılıksal ilişkileri yansıtmakta ve Bayes teoremi ile hesaplanmaktadır. Bu işlemler Oracle JDeveloper yazılımı kullanılarak gerçekleştirilmiştir. Hesaplanan koşullu olasılıklar modelin doğruluk oranını vermektedir. Ağdaki gerçek durum ile programın ürettiği sonuç aynı olduğunda doğru teşhise ulaşılmaktadır. Oluşturulan model, farklı boyutlar da veri kümelerine uygulandığında ortalama yaklaşık %51 doğruluk oranı ile hastalığı doğru teşhis edebilmektedir.

Ayrıca bu veri kümeleri C4.5, Destek Vektör Makinesi ve Çok Katmanlı Yayılım gibi makine öğrenmesi yöntemlerine de uygulanmıştır. En iyi sonucu yaklaşık %75 ortalama doğruluk oranı ile Destek Vektör Makinesi yöntemi vermiştir.

Anahtar Kelimeler: Kaba kümeler kuramı, tanecikli hesaplama, biçimsel kavram çözümleme, Bayes ağları, makine öğrenmesi.

ABSTRACT

Medical Decision Support Systems (MDSS) are intelligent software systems that are able to make deduction in case of lack of information and in the presence of ambiguity. In order to model the uncertainty in these systems, different soft computing methods like Bayesian networks, rough sets, neural networks, fuzzy logic, inductive logic programming, genetic algorithms and/or hybrid systems that are combination of several of these mentioned methods are used. Bayesian network is a data oriented method that is frequently used in MDSS.

In this study, based on the similarity between Bayesian network and concept lattice in formal concept analysis, a data oriented model reflecting the relations between Symptom-Disease is proposed, in which diseases and symptoms are modeled as objects and attributes respectively.

In order to test the developed model, with the technical problems of finding data sets, an ALARM network based on real situations and real patient data is used. Data sets of various size is implemented using NETICA software. In order to reduce computational complexity, ROSETTA software, which is a fundamental tool in rough set theory and based on discernibility, is used.

Lattice of symptom-disease concept is constructed using related algorithms by Oracle JDeveloper software and later the conditional probabilities that are indicator of the models accuracy are calculated by Bayes theorem. Correct diagnosis is achieved when the actual situation in the network and the result of the program is the same. The proposed model applied to various sized data set can give the correct diagnosis with an average of 51% accuracy.

Also, these datasets are applied to the machine learning methods like C4.5, Support Vector Machines (SVM) and Multi Layer Perceptron (MLP). Support Vector Machine has given the best result as approximately 75% average accuracy.

Keywords: Rough set theory, granular computing, formal concept analysis, Bayesian networks, machine learning.

1. GİRİŞ

Tanecikli Hesaplama (TH) (Granular Computing) yöntemi, problem çözümede tanecikleri (granules) kullanan kuram, yöntem bilim, teknik ve araçları kapsayan, bilgisayar bilimlerinde problem çözümünde kullanılan ve son yıllarda önemi giderek artan bir kavramdır. Bu kavram ilk defa Zadeh (1979) tarafından yayınlanan bilgi tanecikleri adlı çalışmada ortaya çıkmış olup, Lin (1997) tarafından getirilen bir öneri ile hayata geçmiştir. Tanecikli hesaplamanın temel bileşenleri alt kümeler, sınıflar ve bir evrenin kümeleridir. TH yöntemi yeni olmasına rağmen, bu yöntemin temel fikirleri ve özellikleri geniş alanda uygulanmaktadır. Bu alanlardan bazılarında, kaba küme kuramı (rough set theory), bilgi keşfi (knowledge discovery), Dempster Shafer inanç fonksiyon kuramı (Dempster Shafer belief function theory), yapay zekâ (YZ) (artificial intelligence), kümeleme analizi (clustering analysis), veri sıkıştırma (data compression), veri tabanları (databases), karar ağaçları (decision tree), böl ve fethet (divide and conquer), aralık hesaplama (interval computing), makine öğrenmesi (machine learning), yapısal programlama (structural programming) ve bilgi erişimi (information retrieval) örnek olarak verilebilir. Bilgi tanecikleştirme ve kavram yaklaşımı tanecikli hesaplamanın temel konularından bazılarıdır. Bilginin tanecikleştirilmesi mevcut bilgiye dayalıdır. Bir evrenin tanecikleştirilmesi benzer elemanları evrenin işlenmemiş parçaların görüşlerini oluşturabilecek taneciklere gruplamayı içerir. Kavramların yaklaşımı evrenin alt kümeleri ile temsil edilir ve tanecikleri kullanarak kavramların tanımlanması ile ilgilenir.

Bir tanecik normal olarak ayırt edilemezlik, benzerlik veya fonksiyonellik ile bir araya gelen elemanlardan oluşur (Yao, 2001). Bu kavramlardan ilerideki bölümlerde ayrıntılı bir şekilde bahsedilmiştir.

TH çalışmak için birçok neden vardır ve bu konuda farklı alanlarda birçok araştırmalar yapılmaktadır. Problem çözümedeki basitlik ve pratiklik gerekliliği en önemli nedenlerden bazılarıdır. Bir problem tamamlanmamış, belirsiz veya kesin olmayan bilgi içerdiğinde elemanları belirgin bir şekilde farklılaştırmak zor ve tanecikleri dikkate almak gerekli olabilmektedir. Ayrıntılı bilgi mevcut olduğunda bile verimli ve pratik çözüm elde edebilmek için tanecikleri kullanmak etkili olabilir. Çoğu pratik problem için çok kesin çözümler gerekmez. Tanecikli hesaplamanın verimli ve pratik zeki bilgi sistemlerinin tasarım ve uygulamasında önemli bir rol oynayacağı beklenmektedir.

Tanecikli hesaplamaadaki birçok çalışma bulanık sistemler (fuzzy systems), sinir ağları (neural networks), kaba kümeler (rough sets), makine öğrenmesi ve kaba mantık çıkarımı (rough

logic reasoning) gibi esnek hesaplama (soft computing) sistemleri ile somut modellere odaklanmaktadır. Bir makine öğrenme tekniği olan tümevarımsal mantık programlama (inductive logic programming) örnekler ve bilginin alt yapısında (geçmişinde) birinci sıralı cümlecik kuramların yapılandırılması için kullanılmaktadır. Bu yöntemin amacı, daha önce sınıflandırılmamış, verilen örnekler kümesinden yüksek tahminleyici güce sahip bir sınıflandırma kuralı kümesi keşfetmektir. PROGOL (Muggleton, 1995) ve FOIL (Quinlan, 1990) sınıflandırma algoritmaları, bu yönteme dayalı olarak çoğu alanda başarı ile uygulanmıştır. Son zamanlarda, birçok tümevarımsal programlamaya dayalı ölçekli tümevarım algoritmaları örneğin, TILDE (Blockeel ve De Raedt, 1997) ve GOLEM (Muggleton, 1990) geliştirilmiştir. Veri madenciliği (VM) sürecinde kullanılan yukarıda verilen yöntemlerin her birinin bir diğerine göre avantaj ve dezavantajları bulunmaktadır. Bu yöntemlerin üstün olan yönlerini birleştirmek suretiyle elde edilen melez algoritmalar (örneğin, sinir ağları ve bulanık mantığın birleştirilmesi ile oluşturulan sinirsel bulanık hesaplama gibi) ile yararlı örüntülerin verimli bir şekilde madenlenmesi mümkündür. Veri Madenciliği geniş veri kümesi içinde gizli ve yararlı olan örüntüleri ve ilişkileri bulma sürecidir. Veri madenciliği sürecinin çok çeşitli alanlarda başarılı uygulamaları mevcuttur. Örneğin tıp alanında, hastane yönetim sistemlerinde depolanan birçok teşhis edici bilgi ve test sonuçlarının değerlendirilmesi ile hastalıkların belirlenmesi ve hastalar için uygun tedavinin belirlenmesi (Zhang ve Zhang, 2002), milyonlarca hasta kayıtlarının incelenerek hastalıkların gizli belirtilerinin ortaya çıkarılmasında, bilimsel veri analizinde ve genetik veri madenciliği alanında başarılı bir şekilde uygulanmaktadır (Pal ve Mitra, 2004). Tıp alanında yukarıda verdiğimiz örnekler Karar Destek Sistemlerinin (Decision Support Systems) (KDS) bir uygulaması olan Tıbbi Karar Destek Sistemleri (Medical Decision Support Systems) (TKDS) ile gerçekleştirilir. TKDS, hekimlere ve sağlık alanında çalışan personellere alacakları tıbbi kararlarda destek sağlayan karmaşık akıllı bir yazılım sistemidir. Bu sistemler tasarlanırken dikkat edilmesi gereken en önemli nokta, karar desteğinin verileceği tıp alanının çok iyi incelenmesi geniş ve kapsamlı bir bilgi tabanının hazırlanmasıdır. TKDS tasarımında en küçük bir ayrıntının göz ardı edilmesi bile sistemin hatalı sonuçlar üretmesine ve istenilmeyen sonuçlara neden olabilir. Tüm yukarıda anlatılanlardan VM sürecinin KDS ile yakından ilişkili olduğu görülmektedir. Belirsizliğin ve eksik bilginin varlığında karar vermeyi destekleyen bu akıllı sistemlerin kullanımı giderek artmaktadır. Karar Destek Sistemlerini, veri güdümlü ve kural tabanlı sistemler olmak üzere iki temel grupta incelemek mümkündür. Veri güdümlü sistemler geniş veri yığınları içinde çalışır ve VM yöntemleri ile karar sürecine

destek verirler. Bu sistemler, sistem tarafından üretilen sistemin karakteristiklerini açıklamak için “*aşağıdan yukarıya*” mantığını kullanırlar (Yücebaş, 2006).

TKDS alanında bilinen en eski uygulama MYCIN sistemidir. YZ çalışmalarının geliştiği bir dönemde geliştirilen MYCIN sisteminde kural tabanı üzerinde geri zincirleme yöntemi kullanılmaktaydı. TKDS’lerinde kullanılan yöntemleri incelediğimizde KDS’lerinde olduğu gibi iki yöntem (veri güdümlü ve kural tabanlı) bulunmaktadır. Bu yöntemlerden veri güdümlü sistemlerde yapay sinir ağları (artificial neural networks), genetik algoritmalar (GA) (genetic algorithms), bulanık mantık (fuzzy logic), durum tabanlı nedenleme (case based reasoning) ve Bayes Ağları (BA) (Bayesian Networks), örüntü tanıma (pattern recognition) yöntemleri, vs. kullanılabilir. TKDS’de en çok kullanılan yöntemler Bayes ağları ve kural tabanlı yöntemlerdir. Literatürdeki çalışmalara baktığımızda Bayes ağlarının daha yaygın kullanıldığı görülmektedir. Bayes ağlarının TKDS’de yaygın olarak kullanılmasının nedeni karar vermedeki belirsizlik için çok önemli bir model olmasından kaynaklanmaktadır.

Burada anlatılanlardan tanecikli hesaplamaların TKDS ile yakından ilişkili olduğu görülmektedir.

Araştırmalar TH konularının ve problem çözmedeki önemini göstermektedir. Ayrıca literatürde tanecikli hesaplamaların birçok özellikli modelleri de incelenmiştir. Pawlak (1998), Polkowski ve Skowron (1998), Skowron ve Stepaniuk (1998) tanecikli hesaplamayı kaba küme kuramı ile bağlantılı olarak incelemişlerdir. Yao (1999a) katmanlaştırılmış kaba küme (stratified rough set) yaklaşımlarının çalışılması için hiyerarşik taneciklerin kullanılmasını önermiştir. Lin (1998) ve Yao (1999b; 1999c) tanecikli hesaplamayı komşuluk sistemleri kullanarak çalışmışlardır. Klir (1998) tanecikli olasılıklar ile hesaplanan bazı temel konularını araştırmıştır (Yao, 2001).

TKDS’de sıklıkla kullanılan Bayes ağlarının Zeki Veri Analiz (intelligent data analysis) (ZVA) yöntemlerinden biri olan kaba kümeler kuramı ile ilişkili olduğu Pawlak (2003) tarafından gösterilmiştir. Kaba kümeler kuramında herhangi bir karar algoritması, Bayes kuramını klasik Bayes metodolojisi ile birleştirilen önceki veya sonraki olasılıkları kullanmadan sağlar. Aynı zamanda, her karar algoritması ile bir akış çizgesinin birleştirilmesi mümkündür. Bayes kuramı çizgedeki bilgi akışı ile yakından ilişkilidir.

Bayes kuramı ile kaba küme yöntemlerinin birlikte ele alındığı diğer çalışmalar Slezak ve Ziarko (2002, 2003a, 2003b, 2005), Greco vd. (2004) makalelerinde incelenmiştir.

ZVA yöntemlerinden bir diğeri de, Biçimsel Kavram Çözümlemesidir (Formal Concept Analysis) (BKÇ). BKÇ gerçek dünyayı nesnelere ve özellikler (nitelikler) olarak modelleyen bir yöntemdir ve nesnelere taşıdıkları özellikler arasında bir ilişki kurulabileceğini varsayar. BKÇ kavram adı verilen birimler tanımlar. Bir kavram özellik ve nesne kümelerinden oluşan bir ikili ile ifade edilir. BKÇ ve Kaba Küme Kuramı veri analizi için ilgili ve birbirini tamamlayan iki yöntemdir. Bu yöntemler, kavramlar fikrine farklı bakış açılarından bakar. (Yao, 2004a; 2004b) BKÇ’ de tanımlanamayan kavramlar söz konusu olduğunda kaba kümeler kuramı bu kavramlara alt ve üst yaklaşımlar ile yaklaşır.

BKÇ’ ye TKDS açısından bakarsak örneğin, doktor bir hastayı muayene ettiğinde hastada mevcut olan belirtilere en yakın olan belirtiyeye sahip bir hastalığı bulur. Bu durumda belirtileri özellikler ve hastalıkları da nesnelere olarak modelleyebiliriz. Bu nedenle küçük bir bilginin varlığı bile kavram yaklaşımlarını bulmaya imkân vermede temel bir öneme sahiptir (Saquer ve Deogun, 2001).

Bu tez çalışmasında, Belirti-Hastalık ilişkileri, hastalıklar nesnelere ve belirtiler özellikler olarak modellenmiştir. Veri kümesi için gerçek hasta verileri ile gerçek durumlardan çıkarılmış bir ağ yapısı kullanılmış ve NETICA yazılımı ile sentetik veri kümeleri üretilmiştir. Bu veri kümeleri ayrıca, makine öğrenme yöntemlerinden C4.5 karar ağacı (decision tree) (Quinlan, 1993), Destek Vektör Makinesi (DVM) (Support Vector Machine, SVM) (Cortes ve Vapnik, 1995), (Vapnik, 1995;1998) ve Çok Katmanlı Yayılım (ÇKY), (Multi Layer Perceptron, MLP) (Hornik vd., 1989) yöntemleri üzerinde denenmiştir.

Hesaplama karmaşıklığının azaltılması amacıyla özellik verilerine kaba kümeler kuramında ayırtedilebilirliğe dayalı bir yazılım sistemi olan ROSETTA uygulanmıştır. Daha sonraki aşamada belirti ve hastalık bağlamına ait kafes yapısı oluşturulmuştur. Bu kafes yapısındaki kenarlar, belirtiler ve hastalıklar arasındaki olasılıksal ilişkileri göstermektedir. Kafes yapısındaki bu ilişkiler Bayes teoremi ile hesaplanmaktadır. Kafes yapısının oluşturulması ve olasılıksal ilişkilerin hesaplanması için Oracle JDeveloper yazılımı kullanılmıştır.

Tez çalışmasının diğer bölümleri aşağıdaki şekilde düzenlenmiştir.

Çalışmanın 2. bölümünde, Çizge Kuramının temel kavram ve tanımları, Bayes ağlarının oluşturulması ve ALARM ağındaki iki çıkarım yöntemlerinden Pearl ve Lauritzen-Spiegelhalter’ in algoritmaları ile Bayes ağlarındaki çıkarım yöntemleri ayrıntılı bir şekilde incelenmiştir.

3. bölümde Tanecikli Hesaplama yöntemi ve temel kavramları ile Kaba Kümeler Kuramı ve Bayes Ağları ile Akış Çizgeleri arasındaki ilişkiler ayrıntılı bir şekilde anlatılmıştır. Ayrıca, özellik verilerinin indirgenerek minimal veri kümelerinin bulunması işleminde kullanılan ROSETTA yazılımının temelini teşkil eden ayırtedilebilirliğe ait tanımlar bu bölümde verilmiştir.
4. bölümde, kavram kafesi oluşturmak için kullanılan Biçimsel Kavram Çözümlemenin temel kavram ve tanımları açıklanmıştır. Ayrıca kaba kümeler kuramı ile biçimsel kavram çözümü arasındaki ilişkilere bu bölümde ayrıntılı olarak değinilmiştir.
5. bölümde Belirti-Hastalık ilişkisinin kavram kafes yapısı ile modellenmesi gerçekleştirilmiş ve kurulan model ALARM ağı kullanılarak, NETICA yazılımı ile üretilen sentetik veri kümeleri üzerinde denenmiştir. Ayrıca bu veri kümeleri, makine öğrenme yöntemlerinden C4.5, DVM ve ÇKY yöntemleri üzerinde denenmiştir.
6. bölüm olan sonuç ve değerlendirmeler bölümünde ise, çalışmada elde edilen sonuçların değerlendirilmesi verilmiştir.

2. BAYES AĞLARI (BAYES İNANÇ AĞLARI, OLASILIKSAL AĞLAR)

Bayes Ağları (Bayesian Networks) (BA), istatistiksel sonuç çıkarımı ve karar vermedeki belirsizlik için çok önemli bir model olup istatistiksel bağımlılıkların grafiksel temsilini sunan güçlü ve güvenilir bir biçimselleştirme (Milho ve Fred, 2000; Aktaş ve Ülengin, 2004).

BA, YZ alanındaki araştırma ve uygulamaların gerçekleştirilmesinde son yıllarda önemi gittikçe artan bir konu haline gelmiştir. Bu konuda yapılan pek çok araştırma Bayes ağlarının TKDS' de etkin bir biçimde kullanılabileceğini göstermiştir. BA, $p(D|T)$ 'yi hesaplamak için Bayes kuramını kullanırlar ki burada "T" ve "D" sırasıyla belirtiler ve teşhisler rassal kümelerini ve $p(D|T)$ belirtilerden tanılara giden koşullu olasılığı gösterir. Bu teorem yeni bilgiler temelinde olasılıkları güncelleyen matematiksel formüller sağlar. Yeni bilginin (T) sonucuna bağlı olarak, verilen hastalıkla ilgili daha sonraki olasılık olan $p(D|T)$ olasılığı hesaplanır. Bu yeni olasılık değeri hareket için temel oluşturur. Bu teorem Bayesian çıkarsama motorunun önemli kısmını oluşturur. Bayes hipotezinde bir kanıt değerlendirebilmek için Boolean mantığı, koşullu bağımsızlık (her hasta için sadece bir teşhis), marjinalleşme teknikleri (gereksiz ve kullanışsız parametrelerden arınma) önemli araçlardır. Bu araçları uygulayabilmek için, bir önceki olasılıklar ve olabilecek(olacağı muhtemel) olasılıklar sağlanmalıdır. Olabilirlik oranı bu olasılıkların oranıdır. Bir kanıtın olabilirlik oranı iki hipotez altında önceki ihtimallerin birleştirilerek sonrakileri oluşturması şeklinde üretilir. Bir olayın ihtimali diğer olaylardan daha yüksek ise olabilirlik şansı çoktur. BA üzerinde çıkarsama yapan karar analiz sistemlerinin elde edilebilirliği, KDS' nin kesinliğini artırabilir ve Bayes ağındaki bilgiyi kapsamlı hale getirebilir.

BA, incelenen sistemlerdeki nedensel ilişkileri ortaya koyan, uzman bilgisinden ve nitel değerlendirmelerden büyük ölçüde yararlanan araçlardır. Bayes yöntemleri belirsizlik koşulları altındaki kısmi inançlar hakkında çıkarım için bir biçimselleştirme sağlar. Bu biçimselleştirmede, önermeler sayısal parametreler şeklinde verilir ve bilginin bazı topluluğu altında inancın derecesini gösterir ve parametreler birleştirilmiş olarak olasılık kuramının kurallarına göre kullanılır. Bu nedenle modellenecek sisteme belirsizlik hâkim ise ve sistem hakkında eksik bilgiler mevcut ise BA yararlanabilecek kullanışlı bir yöntemdir (Pearl, 1988; Aktaş ve Ülengin, 2004).

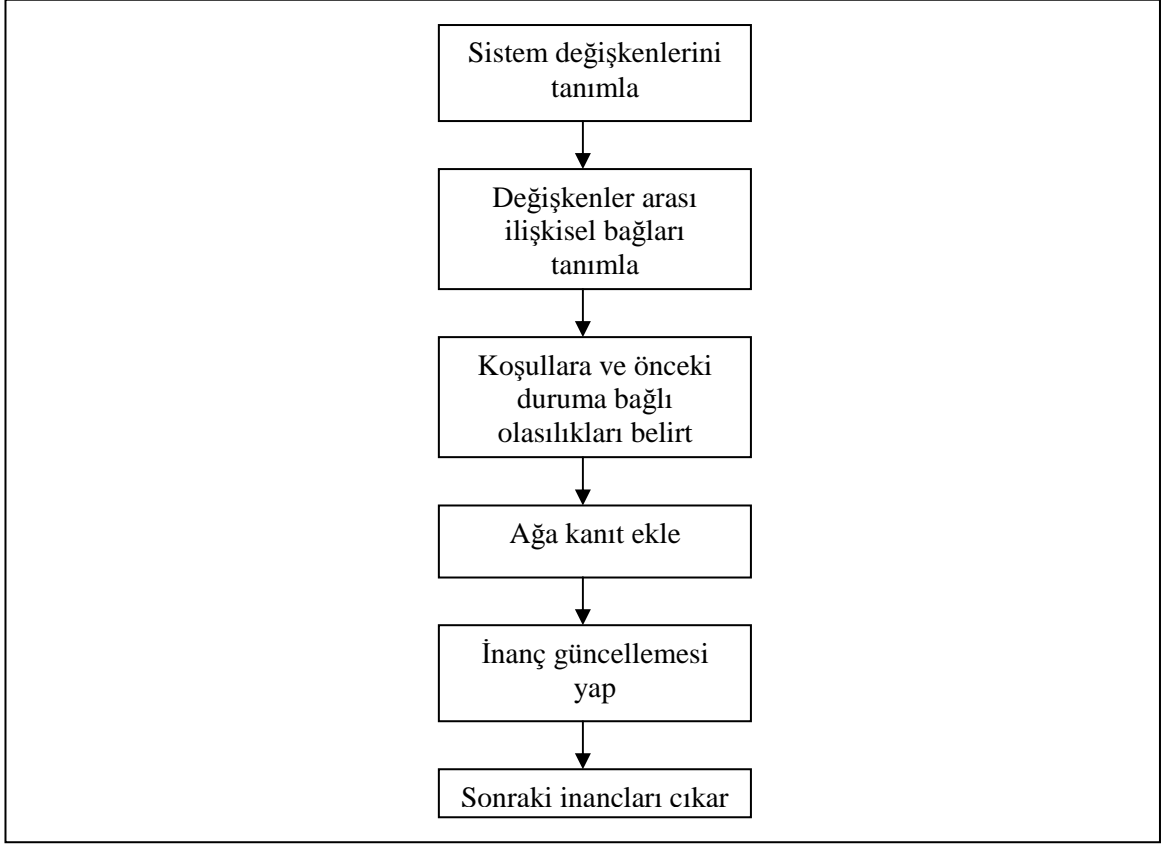
Bayes ağlarının diğer geleneksel yaklaşımlara göre avantajları aşağıdaki gibi sıralanabilir:

- Kayıp ve eksik değerlerde oldukça iyi sonuçlar verir.
- Anlaşılabilirlik kolaylığı oldukça zor olan Yapay Sinir Ağları (YSA) gibi temsillere göre anlaşılabilirliği kolaydır.

- Birleşik olasılık dağılımlarını kodlama yeteneği mevcuttur. Buna ilave olarak, grafiksel yapı yoluyla, birleşik dağılımları oluşturan koşullu dağılımları tanımlar.
- Önceki bilgiyi birleştirme becerisi mevcuttur ve özellikle de nedensel bilgi biçiminde olan bilgi, Bayes ağının yapısını oldukça basitleştirir ve aynı zamanda sonuç olarak meydana gelen modelin anlaşılabilirliğini geliştirir.
- Diğer yaklaşımlara göre karmaşık karar verme modelleri olarak etkindirler.
- Bayes ağlarındaki değişkenlerin alacağı her bir değer için doğru/yanlış'tan daha fazla değere sahip olabilir ve tüm ilişkilerin deterministik olmasının gerekmemesi diğer önemli özelliğidir.

Tüm bu avantajlardan dolayı BA' nın gerçek yaşamda TKDS' nin yanı sıra haberleşme, bilgi erişimi, sistem hataları gibi birçok uygulamaları mevcuttur. BA çizge kuramına güçlü bağlantılar ile olasılıksal modellerin bir sınıfından oluşur. Aslında, BA deterministik tahmin kurallarına dayalı olarak yüklem mantığının bir genişlemesi olarak geliştirilmiştir (Hansen, 2006).

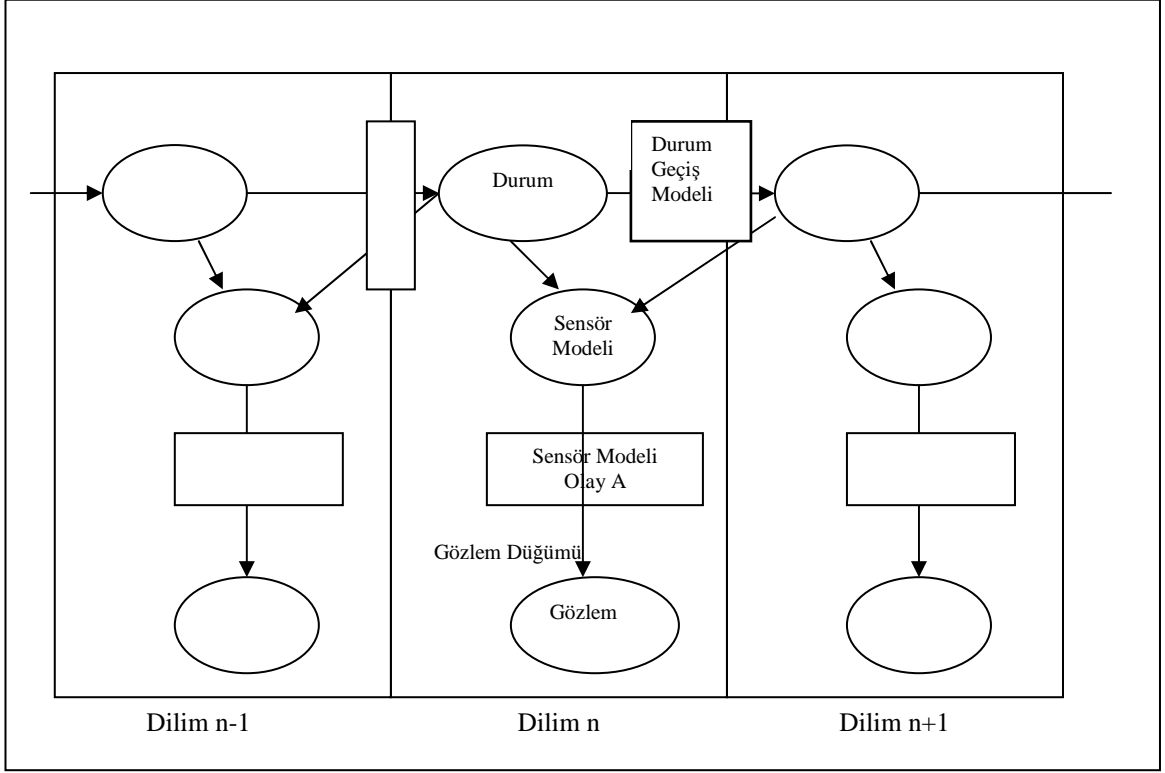
Bir Bayes ağı oluşturulurken, başlangıç tasarımı için modellenecek alanın bir uzmanından yardım alınması gereklidir. Bunun nedeni, nedensel ilişkilerin ve koşullu olasılıkların iyice ve doğru olarak tanımlanmasıdır. Bu ağları oluşturmak için bilgisayarlı öğrenim tabanlı teknolojiler de kullanılabilir. Test verisi kullanılarak bir yazılımın ağ için gerekli parametreleri belirlemesi sağlanabilir. Bu parametrelerden kasıt, koşullu ve olasılıksal dağılımlardır. Yeteri kadar karmaşık bir yazılımda değişkenler arasındaki nedensel ilişkileri belirleyebilir, böylece otomatik olarak uygun bir ağ yapısı oluşturulur. Bir Bayes ağı oluşturulması adımları aşağıdaki Şekil 2.1' de akış diyagramında verilmiştir.



Şekil 2.1 Bir bayes aęı oluřturma adımları

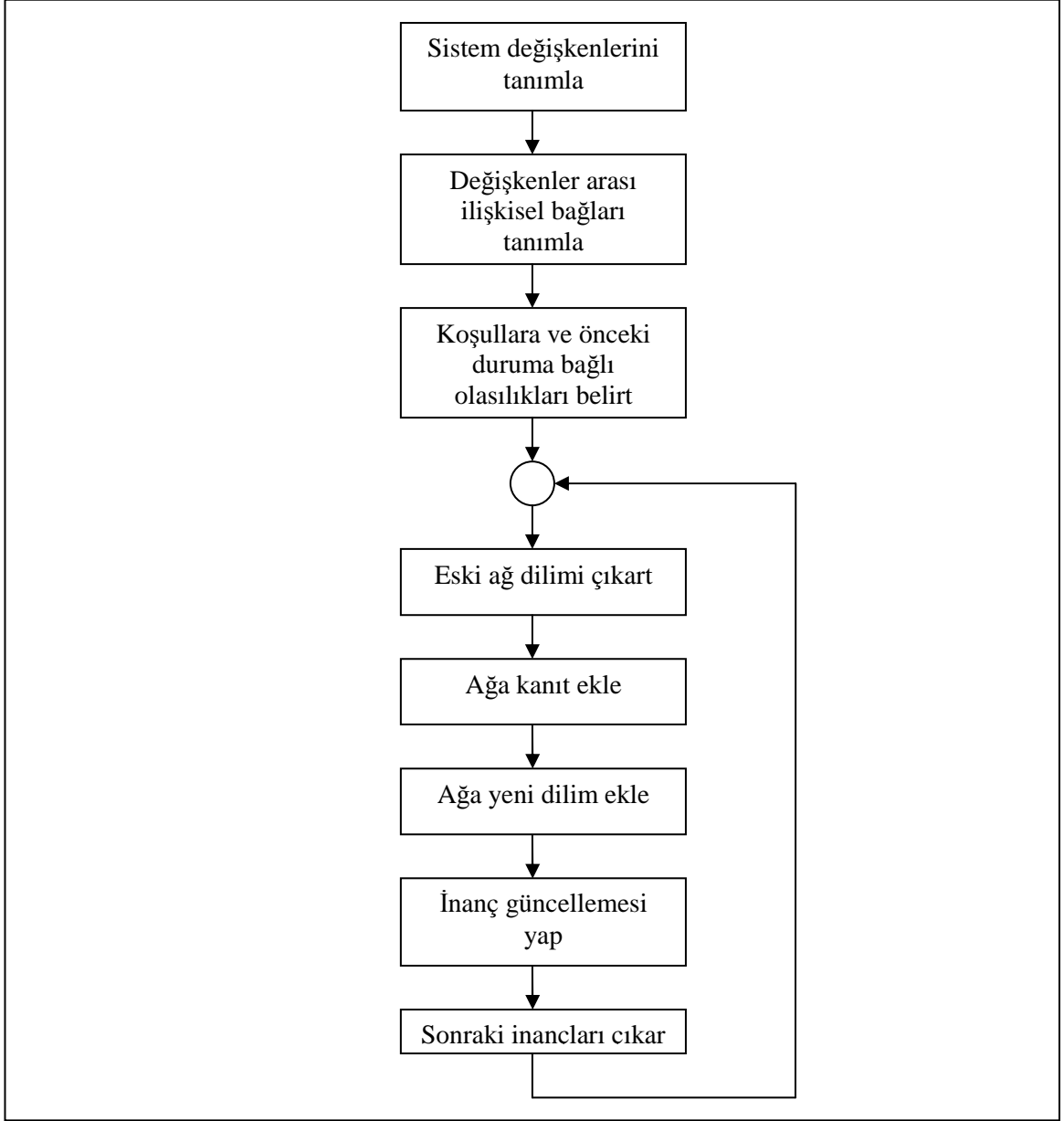
Eęer Bayes aę yapısı daha da geliřtirilerek kullanılmak istenirse dinamik bir yapıdan faydalanılabılır.

Duraęan BA tek bir andaki kanıt ve inançlara dayanarak alıřır. Sonu olarak, bu tr aęlar zamana yayılan ve sre iinde evrimleřen sistemler iin uygun deęildir. Bazen gemiřteki kanıtları ve inanları hatırlamak sistemin bugnk durumu ve gelecekteki ıkarsamaları iin kullanıřlı olabilir. Dinamik BA geici evreleri modellemede kullanılır ve Bayes aęlarının bir uzantısıdır. Daha kk statik Bayes aęlarının kk zaman dilimlerinde birbirleriyle baęlanmasından oluřur. Kanıtlar ve nceki zaman dilimlerinden ıkarılan inanlar řu anki ve gelecekteki zaman dilimlerinin tahmini iin kullanılır. Deęiřik zaman dilimindeki dęmler ve bu dęmlerin ana dęmlerinin olasılıksal daęılımları Durum Evrimleřme Modeli ile belirtilir. Bu yapı sistemin zamana yayılarak nasıl geliřip evrimleřtięini gsterir. Dilimler arasındaki nedensel baęlar ise geici baęlar olarak adlandırılır. Durum ve gzlem dęmlerinin kořullu olasılıkları bir sensr modeli ile belirtilir. Bu model sistemin asıl durumunun okunması sırasında kořullu olasılıksal daęılımı saęlayarak sensrn somutlařtırılmasını saęlar. Ařaęıda Şekil 2.2’ de Dinamik bir Bayes aęı řematik olarak gsterilmiřtir.



Şekil 2.2 Dinamik bayes ağıнын şematik olarak gösterilmesi

Yeni kanıtlar sisteme eklendikçe bunun için yeni bir zaman dilimi eklenir. Hesaplamasal karmaşıklığı (computational complexity) azaltmak için eski zaman dilimleri genellikle kaldırılır ve içlerindeki bilgiler bu dilimleri takip eden diğer dilimlere özet olarak eklenir. Bu durum “*kayan-pencere*” yapısı oluşturur. Çıkarımsama ise sanki ağda normal bir Bayes ağı varmış gibi yapılır. Dinamik BA genellikle daha büyük ve karmaşık ağlar olarak sonuçlandıkları için daha çok hesaplama ve güncellemeye ihtiyaç duyulur. Bu durum gerçek zamanlı (yeni kanıtların sıklığına bağlı olarak) bir güncellemeye ihtiyaç duyulması anlamına gelir. Böylece yöntem aynı olsa da çıkarımsama için daha hızlı ve etkili bir algoritma gereklidir. Dinamik BA, Bayes ağlarına benzer şekilde oluşturulurlar ancak ek olarak zaman dilimleri arasındaki geçici ilişkiler de göz önüne alınmalıdır. Genellikle Dinamik BA ağları bir referans ağ alınarak yapılır, sonrasında yazılım geliştirilir. Dinamik BA için gerekli uygulama adımları aşağıdaki Şekil 2.3’ de verilmiştir.



Şekil 2.3 Dinamik bayes aęı için gerekli uygulama adımları

2.1 Bayes Aęları Hakkında Ön Bilgiler

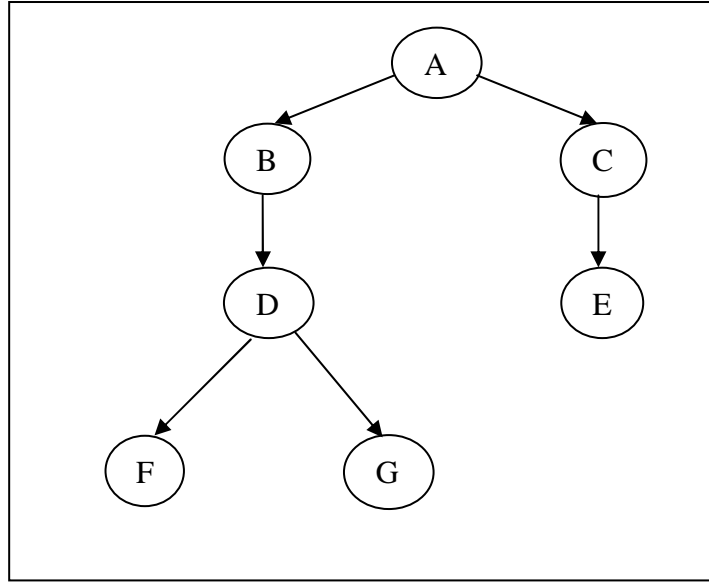
Bir Bayes aę, koşullu baęımsızlık özelliklerin yapısını tanımlayan Yönlü Döngsüz Çizgelerdir (Directed Acyclic Graph) (YDC). Bayes aę, deęişkenler ve deęişkenler arası yönlü kenarların kümesinden oluşur. Her bir deęişken karşılıklı baęımsız durumların sonlu bir kümesine sahiptir. Her köşe rasgele deęişkenler ile gösterilir. Kenarlar, deęişkenler arası olasılık baęımlılıkları gösterir. Bu baęımlılıklar koşullu olasılıkların kümesinden oluşur. Her bir deęişkenlerin ebeveynleri verildiğinde deęişkenin koşullu olasılığı belirlenir. Bir köşenin ebeveynleri olmadığı zaman, bir deęişken koşulsuz bir olasılığa sahiptir. Köşenin ebeveynleri

üzerinde koşullandırılma yapılarak, her düğüm için koşullu olasılıklar hesaplanır (Bender, 2000).

Buradan görüldüğü gibi BA, olasılık kuramı ve çizge kuramı ile çok yakından ilgilidir. Aşağıda bu kuramların bazı temel kavramları ve tanımları verilecektir.

2.1.1 Çizge Kuramının Bazı Temel Kavram ve Tanımları

$V = \{V_1, V_2, \dots, V_n\}$ olası ilişkili nesnelere kümesine sahip olduğumuzu varsayalım. V kümesi düğümler veya köşeler kümesi tarafından grafik biçiminde temsil edilebilir. V 'nin her bir elemanı için bir düğüm mevcuttur. Bu düğümler bağlantılar veya kenar olarak belirlenen doğru parçaları, yaylar veya oklar tarafından bağlanır. Eğer V_i ve V_j iki düğüm arasında bir bağlantı varsa bu bağlantı V_{ij} şeklinde gösterilir. Tüm bağlantıların kümesi $E = \{E_{ij} \mid V_i \text{ ve } V_j \text{ bağlantılı}\}$ ile gösterilir. V ve E kümeleri bir çizgeyi tanımlar. Aşağıdaki Şekil 2.4' de 7 düğümden oluşan bir çizge örneği verilmiştir. Bu düğümlerin her biri bir daire ile ve $E = \{E_{AB}, E_{AC}, E_{BD}, E_{CE}, E_{DF}, E_{DG}\}$ şeklinde 6 bağlantılı küme ile temsil edilir. Her bir bağlantı 2 düğümü bağlayan bir doğru parçası tarafından temsil edilir.



Şekil 2.4 Bir çizge örneği (Castillo vd., 1997)

Aşağıda çizge kuramına ait bazı tanımlara yer verilmiştir.

Çizge veya Ağ: $G = (V, E)$ çizgesi V ve E kümeleri tarafından tanımlanır. Burada V , $V = \{V_1, V_2, \dots, V_n\}$ düğümlerinin sonlu bir kümesidir ve E farklı düğümlerin tüm olası sıralı çiftlerinin bir alt kümesi olan bağlantıların kümesidir.

Uzman sistemler (expert systems) alanında çizgeler önermesel değişkenlerin (düğümler) kümesi ve onlar arasındaki bağımlılık ilişkilerini (bağlantıları) temsil etmek için kullanılır. Bir çizgenin bağlantıları içerdiği düğümlerin sırasına bağlı olarak yönlü veya yönsüz olabilir.

Yönlü Bağlantı: $G = (V, E)$ bir çizge olsun. $E_{ij} \in E$ ve $E_{ji} \notin E$ olduğundan E_{ij} bağlantısı yönlü bağlantı olarak adlandırılır. V_i ve V_j düğümleri arasındaki bir yönlü bağlantı $V_i \rightarrow V_j$ şeklinde gösterilir.

Yönsüz Bağlantı: $G = (V, E)$ bir çizge olsun. $E_{ij} \in E$ ve $E_{ji} \in E$ olduğunda V_i ve V_j düğümleri arasında bir yönsüz bağlantı $V_i - V_j$ veya $V_j - V_i$ şeklinde gösterilir.

Yönlü veya Yönsüz Çizge: Tüm bağlantıların yönlü olduğu bir çizge yönlü çizge ve tüm bağlantıların yönsüz olduğu çizgede yönsüz çizge olarak adlandırılır. Böylece bir yönlü çizgede bağlantı tanımlayan düğümlerin sırası önemli fakat yönsüz çizgede önemsizdir.

Komşuluk Kümesi: $G = (V, E)$ çizgesi ve V_i düğümü verildiğinde V_i 'nin komşuluk kümesi V_i 'den doğrudan elde edilebilen düğümlerin kümesidir ve $Adj(V_i) = \{V_j \in V \mid E_{ij} \in E\}$ şeklinde gösterilir. Bu tanım V düğümlerinin kümesini ve V 'de ki her düğüm için komşuluk kümelerini belirleyerek bir çizgenin alternatif tanımını sağlar. Yani, (V, E) çizgesi (V, Adj) olarak temsil edilebilir. Burada, $V = \{V_1, V_2, \dots, V_n\}$ düğümler kümesidir ve $Adj = \{Adj(V_1), Adj(V_2), \dots, Adj(V_n)\}$ tüm komşu kümelerin kümesidir.

İki Düğüm Arasındaki Yol: V_i düğümünden V_j düğümüne bir yol $\{V_{i_1}, V_{i_2}, \dots, V_{i_r}\}$ düğümlerinin sıralı kümesidir. $V_{i_1} = V_i$ 'den başlar ve $V_{i_r} = V_j$ 'de biter. Şöyle ki, V_{i_k} 'den $V_{i_{k+1}}$ 'e bir bağlantıda olduğu gibi. Yani, $V_{i_{k+1}} \in Adj(V_{i_k})$ 'dir. ($k = 1, 2, \dots, r-1$)

Bu yolun uzunluğu $(r-1)$, yani içerdiği bağlantıların sayısıdır. Yönsüz çizge durumunda $(V_{i_1}, V_{i_2}, \dots, V_{i_r})$ yolu $V_{i_1} - V_{i_2} - \dots - V_{i_r}$ olarak temsil edilebilir. Bu da bağlantıların yönsüz karakterini gösterir. Benzer şekilde, $V_{i_1} \rightarrow V_{i_2} \rightarrow \dots \rightarrow V_{i_r}$ yönlü çizgede bir yolu temsil etmenin bir yoludur.

Kapalı Yol: Eğer $V_{i_1} = V_{i_r}$ ise yani başlangıç ve sonlanma düğümleri aynı düğümlere sahip ise V_{i_1} 'den V_{i_r} 'ye yolu kapalıdır.

Tam Çizge: Bir yönsüz çizgenin tüm düğüm çiftleri arasında bir bağlantı varsa bu çizge tam çizge olarak adlandırılır.

Tam Küme: Eğer S 'deki düğümlerin her çifti arasında bağlantılar varsa bir G çizgesinin S düğümlerinin bir alt kümesinin tam olduğu söylenir.

Klik: C düğümlerinin bir tam kümesi, eğer azami ise yani diğer tam kümenin uygun bir alt kümesi değilse klik (clique) olarak adlandırılır.

Döngü: Bir döngü, yönsüz çizgedeki kapalı bir yoldur.

Bir Düğümün Komşuları: Bir yönsüz çizgedeki V_i düğümüne komşu olan düğümler kümesi V_i 'nin komşuları olarak belirlenir ve $Nbr(V_i) = \{V_j \mid V_j \in Adj(V_i)\}$ ile ifade edilir.

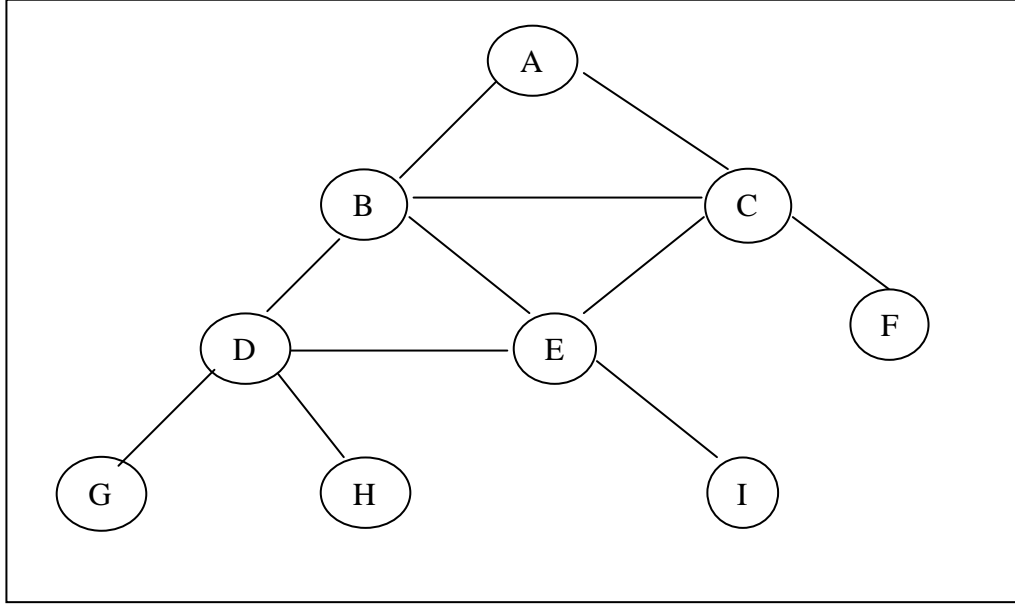
Düğüm Kümesinin Sınırı: Verilen S alt kümesinde düğümlerin komşu kümelerinin birleşimi, S 'deki düğümler hariç, S 'nin sınırı olarak adlandırılır ve $Bnd(S)$ olarak gösterilir ve $Bnd(S) = (\bigcup_{V_i \in S} Nbr(V_i)) \setminus S$ ile ifade edilir. Burada $V \setminus S$, S 'dekiler hariç, V 'deki tüm düğümlerin kümesini gösterir.

Ebeveynler ve Çocuklar: V_i 'den V_j 'ye $V_i \rightarrow V_j$ bir yönlü bağlantısı olduğu zaman V_i , V_j 'nin ebeveyni, V_j 'de V_i 'nin çocuğu olarak adlandırılır.

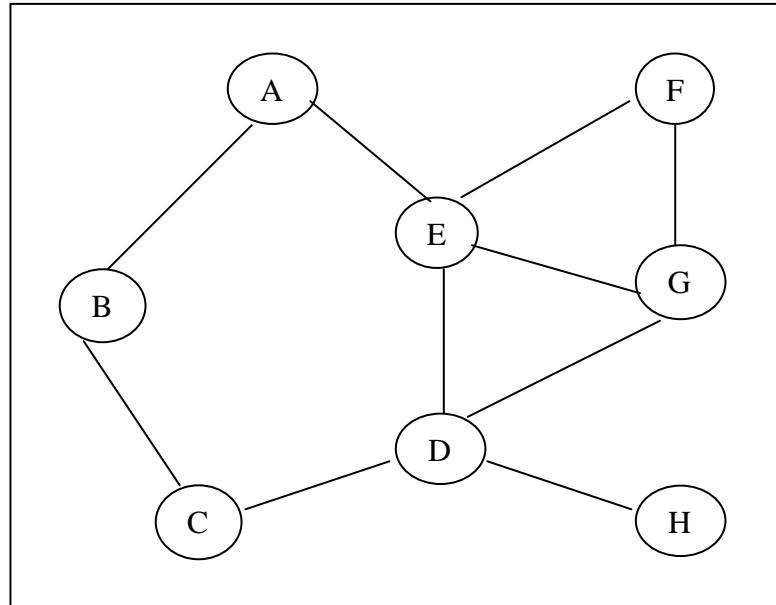
Moral Çizge: Yönlü çizgedeki bir ortak çocuk ile her düğüm çifti arasına bir bağlantı ekleyen ve daha sonra bağlantıların yönünü iptal ederek elde edilen çizgeye, moral çizge (moral graph) denir.

Bir Döngünün Bağlaması (Chord): Bir bağlama, döngüde yer almayan iki düğüm arasındaki bağlantıdır.

Üçgenleştirilmiş (Triangulated) Çizge: Bir çizgenin, 4 uzunluklu veya daha fazla uzunlukta olan her döngüsü en az bir bağlamaya sahip ise bir yönsüz çizgenin üçgenleştirilmiş olduğu söylenir. Üçgenleştirilmiş çizgeler birçok alanda ilginç uygulamalara sahip olan yönsüz çizgelerin özel bir türüdür. Şekil 2.5, 4 uzunlukta veya daha fazla uzunlukta olan tüm döngülerin en az bir bağlamaya sahip olduğu üçgenleştirilmiş bir çizgeyi gösterir ve $A-B-E-C-A$ ve $B-C-E-D-B$ çizgelerinde olduğu gibi 4 uzunluktaki iki döngüyü içerir ve 5 uzunluktaki çizgenin bir döngüsü $A-B-C-D-E-A$ şeklindedir. Bu döngülerin her biri en az bir bağlamaya sahiptir. Diğer yandan, şekil 2.6' deki çizge üçgenleştirilmiş değildir. Bunun nedeni, 5 uzunluktaki bir çizgenin döngüsü $A-B-C-D-E-A$ bir bağlamaya sahip değildir.



Şekil 2.5 Üçgenleştirilmiş bir çizge (Castillo vd., 1997)



Şekil 2.6 Üçgenleştirilmemiş bir çizge(Castillo vd., 1997)

2.1.2 Olasılık Kuramının Bazı Temel Kavram ve Tanımları

2.1.2.1 Olasılık Ölçüsü

Belirsizliğin ölçülmesi için öncelikle verilen bir S hükmünü ele alalım. Bir olasılıksal model, S hükmünün olasılığını hesaplamaya izin veren olasılıksal bilginin kodlamasıdır. Örneğin, A, B, C üç atomik önerme (atomic proposition) kümesi olmak üzere, S hükmü bu önermeleri içeren tüm Boolean formüllerinden oluşur. Bu üç atomik önerme için, birleşik

dağılım fonksiyonu negatif olmayan ağırlıkları sekiz kombinasyona atmalıdır. Bunlar, $(\mathbf{A} \wedge \mathbf{B} \wedge \mathbf{C}), (\mathbf{A} \wedge \mathbf{B} \wedge \neg \mathbf{C}), \dots, (\neg \mathbf{A} \wedge \neg \mathbf{B} \wedge \neg \mathbf{C})$ şeklindedir. Ayrıca aşağıdaki Aksiyom 1'de de görüleceği gibi bu ağırlıkların toplamaları 1'e eşit olmalıdır. S kümesi ayrıca örnek uzay (sample space) olarak da adlandırılır. O halde burada amacımız, S 'in her alt kümesine onun varlığı hakkındaki belirsizliğin derecesini ölçen bir gerçek değer atamaktır. Açık fiziksel ve pratik anlamlar ile ölçümler elde edebilmek için bazı gerçek ve sezgisel özellikler olasılık ölçümleri (probability measures) olarak bilinen ölçüm sınıfını tanımlamada kullanılır.

$A \subseteq S$ herhangi bir alt kümesini $[0,1]$ aralığına eşleyen bir p fonksiyonu eğer, aşağıdaki aksiyomları sağlarsa olasılık ölçümü olarak adlandırılır.

Aksiyom 1: (Sınır) $p(S) = 1$

Aksiyom 2: (Toplanabilirlik) S kümesinin ayrık alt kümelerinin herhangi bir sonsuz A_1, A_2, \dots , dizisi için

$$p\left(\bigcup_{i=1}^{\infty} A_i\right) = \sum_{i=1}^{\infty} p(A_i) \text{ dir.} \quad (2.1)$$

Aksiyom 1 gösterir ki, belirsizlik derecemize rağmen S evrensel kümemizde en az bir eleman vardır.

Aksiyom 2, ayrık alt kümelerin bir birleşiminin olasılığını hesaplamada kullanılan bir toplam formülüdür ve verilen alt kümenin belirsizliği bunun ayrık bölümlerinin belirsizliklerinin toplamı olduğunu gösterir. Bu özellik ayrıca, sonlu diziler içinde geçerlidir. Yukarıda ki aksiyomlardan olasılık ölçümünün aşağıdaki özellikleri elde edilebilir.

Özellik 1: (Sınır) $p(\emptyset) = 0$

Özellik 2: (Monotonluk) Eğer $A \subseteq B \subseteq S$ ise, $p(A) \leq p(B)$ dir.

Özellik 3: (Süreklilik ve Tutarlılık) S 'in alt kümelerinin her artan dizisi $A_1 \subseteq A_2 \subseteq \dots$ veya azalan dizisi $A_1 \supseteq A_2 \supseteq \dots$ için aşağıdaki ifadeye sahibiz.

$$\lim_{i \rightarrow \infty} p(A_i) = p\left(\lim_{i \rightarrow \infty} A_i\right)$$

Özellik 4: (Kapsama-Hariç tutma) S 'in A ve B alt küme çifti verildiğinde aşağıdaki eşitlik her zaman geçerlidir.

$$p(A \cup B) = p(A) + p(B) - p(A \cap B) \quad (2.2)$$

Özellik 1, tam bir bilgi eksikliği ile ilişkili olan kanıtın, 0 olarak tanımlandığını gösterir. Özellik 2 ise, kümedeki bir elemanın üyeliğinin kanıtı en az elemanın ait olduğu kendi alt kümelerinin kanıtı kadar büyük olması gerektiğini gösterir. Diğer bir ifadeyle, A kümesine ait olan elemanın kesinliği A 'ya elemanların eklenmesi ile azalmamalıdır. Özellik 3,

tutarlılık veya süreklilik özelliği olarak görülebilir, eğer S 'nin aynı alt kümesine yakınsayan iki diziyi seçersek aynı limit belirsizliği elde edilmelidir. Özellik 4'den, $A, B, A \cap B, A \cup B$ kümelerinin olasılıkları bağımsız değildir ve (2.2) denklemi ile ilişkilidir.

2.1.2.2 Olasılık Dağılımları

$\{X_1, X_2, \dots, X_n\}$ ayrık rasgele değişkenler kümesi olsun ve $\{x_1, x_2, \dots, x_n\}$ ayrık rasgele değişkenlerin olası örneklerin kümesi olsun. Burada, bir değişken büyük harf ile gösterilir ve değişkene karşılık gelen örnekler küçük harfler ile gösterilir. Örneğin, eğer X_i bir ikili değişken ise bu durumda, x_i , 1 veya 0 değerini alabilir. Aşağıdaki sonuçlar eğer değişkenler sürekli ise geçerli olmaya devam eder. Ancak bu durumda toplam sembollerinin yerini integral sembolleri alır.

$$p(x_1, x_2, \dots, x_n), \quad X \text{ 'deki değişkenlerin birleşik olasılık dağılımını gösterebilir. Yani,}$$

$$p(x_1, x_2, \dots, x_n) = p(X_1 = x_1, X_2 = x_2, \dots, X_n = x_n) \quad (2.3)$$

O halde, i . değişkenin marjinal olasılık dağılımı aşağıdaki formül ile ifade edilir.

$$p(x_i) = p(X_i = x_i) = \sum_{x_1, \dots, x_{i-1}, x_{i+1}, \dots, x_n} p(x_1, x_2, \dots, x_n) \quad (2.4)$$

Olayın varlığı hakkındaki bilgi diğer olayların olasılıklarını değiştirebilir. Örneğin, bir hasta için verilen hastalık olasılığı, kan testinin sonuçları elde edildikten sonra değişebilir. Böylece, yeni bilgi mevcut hale geldiği her zaman olayların olasılıkları değişebilir. Buda koşullu olasılık kavramlarını ortaya çıkarır.

2.1.2.2.1 Koşullu Olasılık

X ve Y , $p(y) > 0$ olmak üzere değişkenlerin iki ayrık alt kümesi olsun. O halde, $Y = y$ verildiğinde X 'in koşullu olasılık dağılımı aşağıdaki denklem ile verilir.

$$p(X = x | Y = y) = p(x | y) = \frac{p(x, y)}{p(y)} \quad (2.5)$$

(2.5) denklemi, X ve Y değişken kümelerinin olasılık dağılımının aşağıdaki gibi yazılabileceğini gösterir.

$$p(x, y) = p(y) \cdot p(x | y) \quad (2.6)$$

(2.5) denkleminin bir özel durumu X değişken kümesinde tek bir değişken olduğunda ve Y değişkenlerinin bir alt kümesi olduğunda elde edilir. Bu durumda (2.5) denklemi şu hale gelir.

$$p(x_i | x_1, \dots, x_k) = \frac{p(x_i, x_1, \dots, x_k)}{p(x_1, x_2, \dots, x_k)} = \frac{p(x_i, x_1, \dots, x_k)}{\sum_{x_i} p(x_i, x_1, \dots, x_k)} \quad (2.7)$$

Bu da, i . deęişken X_i 'nin koşullu olasılık dağılımıdır. $\{X_1, X_2, \dots, X_k\}$ deęişkenlerinin alt kümesi verildiğinde (2.7) denkleminin paydasının toplamı X_i 'nin tüm olası deęerlerini içerir. (2.4) denklemindeki her iki marjinal olasılık ve (2.7) denklemindeki koşullu olasılık eęer tek deęişkenli X_i , tüm deęişkenler ayırık olduęu sürece deęişkenlerin bir alt kümesi ile yer deęiştirebilir. Ayrıca, (2.7) denkleminde eęer $\{X_1, X_2, \dots, X_k\}$ kümesi boş küme ile yer deęiştirdiğinde (2.7) denklemi $p(x_i)$ 'e indirgenir. Bu nedenle, marjinal olasılık koşullu olasılığın özel bir durumu olarak düşünülebilir.

2.1.2.3 Baęımlılık ve Baęımsızlık

Bu bölümde olasılık kuramındaki baęımlılık ve baęımsızlık kavramlarından bahsedilecektir.

2.1.2.3.1 İki Deęişkenin Baęımsızlığı

X ve Y rasgele deęişkenleri $\{X_1, X_2, \dots, X_n\}$ kümesinin iki ayırık alt kümesi olsun.

$$p(x | y) = p(x) \quad (2.8)$$

Denklem (2.8)'den X ve Y 'nin tüm olası x, y deęerleri için X 'in Y 'den baęımsız olduęu söylenebilir, aksi takdirde X, Y 'ye baęımlı olur.

Eęer x ve y , X ve Y 'nin olası deęerleri ise o halde $p(x) > 0$ ve $p(y) > 0$ olacaktır. (2.8) denkleminin anlamı, eęer X, Y 'den baęımsız ise Y hakkında mevcut olan bilginin, X hakkındaki bilgiyi etkilemeyeceęini ifade eder. Yani, Y deęişkeni, X hakkında bilgiye sahip deęildir. Ayrıca, X, Y deęişkeninden baęımsız ise o halde, (2.5) ve (2.8) denklemleri birleştirebilir ve $p(x, y) / p(y) = p(x)$ elde edilir.

Buradan

$$p(x, y) = p(x).p(y) \quad (2.9)$$

(2.9) denklemi elde edilir. Bu denklemin anlamı, X, Y deęişkeninden baęımsız olduęunda X 'in ve Y 'nin birleşik olasılık dağılımının marjinal olasılıklarının çarpımına eęit olduęudur. Baęımsızlık ilişkisinin önemli bir özellięi onun simetrisi olmasıdır. Yani, X, Y 'den baęımsız ise Y 'de X 'den baęımsızdır. Bunun nedeni,

$$p(y | x) = \frac{p(x, y)}{p(x)} = \frac{p(x).p(y)}{p(x)} = p(y) \quad (2.10)$$

olduęudur.

Simetri özellięinden dolayı, X ve Y deęişkenleri baęımsızdır veya karşılıklı baęımsızdır. Simetrimin uygulama anlamı şudur, eęer Y 'nin mevcut bilgisi ile X ilgili ise (ilgisiz ise) o halde X 'in Y ile olan bilgisi de ilgili (ilgisiz) olacaktır.

İki rasgele değişkenlerin bağımlılık ve bağımsızlık kavramları ikiden fazla rasgele değişken durumuna aşağıdaki gibi genişletilebilir.

2.1.2.3.2 Değişkenler Kümesinin Bağımsızlığı

Sadece ve sadece X_1, X_2, \dots, X_m 'in tüm olası x_1, x_2, \dots, x_m değerleri için $\{X_1, X_2, \dots, X_m\}$ rasgele değişkenlerinin bağımsız olduğu söylenebilir. Bu durumda,

$$p(x_1, x_2, \dots, x_m) = \prod_{i=1}^m p(x_i) \quad (2.11)$$

(2.11) denklemini geçerli olacaktır.

Diğer bir ifadeyle, sadece ve sadece değişkenlerin birleşik olasılık dağılımı, marjinal olasılık dağılımının çarpımına eşit ise $\{X_1, X_2, \dots, X_m\}$ değişkenlerinin bağımsız olduğu söylenebilir. Burada (2.11) denklemini (2.9) denkleminin genelleştirilmiş bir halidir. Ayrıca, eğer X_1, X_2, \dots, X_m değişkenleri koşullu olarak birbirinden bağımsız ise diğer bir Y_1, Y_2, \dots, Y_n alt kümesi verildiğinde aşağıdaki (2.12) denklemini geçerlidir.

$$p(x_1, x_2, \dots, x_m | y_1, y_2, \dots, y_n) = \prod_{i=1}^m p(x_i | y_1, y_2, \dots, y_n) \quad (2.12)$$

Bağımsızlığın önemli bir anlamı, bağımsız (ilgisiz) değişkenler hakkında bilgi toplamaya gerek olmadığıdır, yani bağımsızlık ilgisizlik anlamına gelir (Castillo vd., 1997).

2.1.2.3.3 Koşullu Bağımlılık ve Bağımsızlık

X, Y ve Z değişkenlerin üç ayrık kümesi olsun. X, Y ve Z 'nin tüm olası x, y, z değerleri için, X ve Z verildiğinde Y 'den koşullu olarak bağımsızdır, aksi takdirde X ve Y, Z verildiğinde koşullu olarak bağımlıdır. X ve Y, Z 'den bağımsız olduğu zaman $I(X, Y | Z)$ şeklinde ifade edilir. $I(X, Y | Z)$ ifadesi koşullu bağımsızlık ifadesi olarak adlandırılır. $I(X, Y | Z)$ gösterimi yerine $I(X, Z, Y)$ 'de kullanılabilir. Benzer şekilde, X ve Y, Z 'ye koşullu bağımlı olduğu zaman $D(X, Y | Z)$ şeklinde yazılır. Bu da koşullu bağımlılık ifadesi olarak adlandırılır. Bazı durumlarda ifadenin p olasılığı (birleşik olasılık dağılımı) ile ilişkili olasılıksal modelden veya model tarafından elde edildiğini belirtmek için $I(X, Y | Z)_p$ ve $D(X, Y | Z)_p$ şeklinde ifade edilir. Koşullu bağımsızlığın tanımının ana fikri, Z değişken kümesinin bilinmesi durumunda, Y 'yi bilmenin X 'in olasılığını etkilemeyeceğidir. Diğer bir ifadeyle, eğer Z değişken kümesi zaten biliniyorsa Y 'nin bilgisi X hakkında yeni bir bilgi vermez. Koşullu bağımsızlığın alternatif fakat denk tanımı aşağıdaki denklem ile verilir.

$$p(x, y | z) = p(x | z) \cdot p(y | z) \quad (2.13)$$

Burada, bağımsızlık koşullu bağımsızlığın belirli bir durumu olarak görülebilir. Örneğin, $I(X, Y | \emptyset)$ ifadesi X ve Y 'nin koşulsuz bağımsız olduğunu gösterir. Burada \emptyset boş kümedir. Ayrıca, X ve Y koşulsuz olarak bağımsız olabilir ancak, Z değişken kümesi verildiğinde koşullu bağımlılık $I(X, Y | \emptyset)$ ve $D(X, Y | Z)$ aynı anda geçerli olabilir.

Aşağıdaki denklemler, $I(X, Z, Y)$ koşullu bağımsızlık ilişkisi tarafından sağlanan denk özelliklerin kısmi bir listesidir.

$$I(X, Z, Y) \Leftrightarrow p(x, y | z) = p(x | z)p(y | z) \quad (2.14)$$

$$I(X, Z, Y) \Leftrightarrow \exists f, g : p(x, y, z) = f(x, z)g(y, z) \quad (2.15)$$

$$I(X, Z, Y) \Leftrightarrow p(x, y, z) = p(x | z)p(y | z) \quad (2.16)$$

Teorem 2.1: X, Y ve Z değişkenlerin üç ayrık alt kümeleri olsun. Eğer $I(X, Z, Y)$ bazı p olasılıksal modelinde Z verildiğinde X, Y 'den bağımsızdır ilişkisi mevcut ise bu durumda aşağıdaki dört bağımsız koşulu sağlamak gerekir (Pearl, 1988).

- Simetri Koşulu

$$I(X, Z, Y) \Leftrightarrow I(Y, Z, X) \quad (2.17)$$

- Ayrışma Koşulu

$$I(X, Z, Y \cup W) \Rightarrow I(X, Z, Y) \& I(X, Z, W) \quad (2.18)$$

- Zayıf Birleşim Koşulu

$$I(X, Z, Y \cup W) \Rightarrow I(X, Z \cup W, Y) \quad (2.19)$$

- Küçülme (Büzülme) Koşulu

$$I(X, Z, Y) \& I(X, Z \cup Y, W) \Rightarrow I(X, Z, Y \cup W) \quad (2.20)$$

- Kesişme Koşulu

$$I(X, Z \cup W, Y) \& I(X, Z \cup Y, W) \Rightarrow I(X, Z, Y \cup W) \quad (2.21)$$

2.1.2.4 Bayes Teoremi

Olasılık kavramı belirsizlik altında çıkarsama yaparken faydalanabileceğimiz bir kavramdır. Belirsizliğin en temel nedenlerinden birisi, belirsizliğin ilgili olduğu alanın tüm değişkenlerini gözlemlemenin zorluğundan kaynaklanır. Bunun yanı sıra, gözlemlenebilen değişkenlerin ait oldukları dünya deterministik olsa da, rasgele davranış gösterir. Bu nedenden dolayı, ilgili dünyanın tüm değişkenlerini belirlemek ve aralarında mevcut olan ilişkileri modellemek zaman ve maliyet açısından çoğu zaman imkânsızdır. Bayes teoremi, koşullu olasılıkları hesaplayan basit bir formüldür. Bayes teoremi, yeni bir kanıtın göstergesinde o ana kadar olan

inançlarımızı nasıl değiştirmemiz gerektiğini açıklayan bir matematiksel kuraldır. Diğer bir ifadeyle, yeni bilgiler ile hali hazırda bulunan verilerin ve bilgilerin birleştirilmesini sağlar (Yücebaş, 2006). Bayes teoremi aşağıdaki formül ile ifade edilir.

$$p(x_i | x_1, \dots, x_k) = \frac{p(x_i, x_1, \dots, x_k)}{\sum_{x_i} p(x_i, x_1, \dots, x_k)} = \frac{p(x_i) \cdot p(x_1, x_2, \dots, x_k | x_i)}{\sum_{x_i} p(x_i) \cdot p(x_1, x_2, \dots, x_k | x_i)} \quad (2.22)$$

2.2 Bayes Ağlarının Oluşturulması

Biçimsel olarak, BA her bir düğümün rasgele değişkeni veya daha fazla değer alan belirsiz bir nicelik temsil ettiği yönlü döngüsüz çizgelerdir. Yaylar, bağlantılı değişkenler arasındaki doğrudan nedensel etkilerin varlığını ifade eder ve bu etkilerin gücü koşullu olasılıklar tarafından ölçülür.

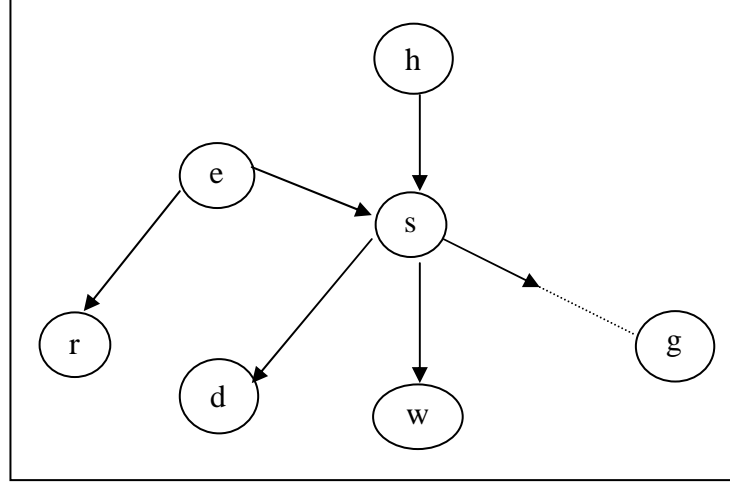
Biçimsel olmayarak, Bayes ağ yapısı basit bir yinelemeli prosedür ile belirlenebilir: Alandaki her bir değişkene bir nokta (köşe) ataması yapılır ve X_i ' nin doğrudan nedenleri olarak algılanan noktaların (köşelerin) \prod_{X_i} seçilmiş kümesinden her bir X_i noktasına doğru oklar çizeriz. Bu doğrudan etkilerin gücü daha sonra her bir X_i değişkeninin bağlantı matrisi $p(x_i | \prod_{X_i})$ koşullu olasılığına atayarak ölçülür ki bu da \prod_{X_i} ebeveyn kümesinin herhangi değer kombinasyonu verildiğinde $X_i = x_i$ olayının koşullu olasılıklarının hükümsel tahminlerini temsil eder. Bu yerel tahminlerin birleşimi (mantıksal çarpım, Ve işlemi) tüm olasılıksal sorguların cevaplanabildiği bir tabanda ve uyumlu genel (global) modeli (örneğin, bir birleşik dağılım fonksiyonu) açıkça belirtir. Sonuç olarak, birleşik dağılım fonksiyonu X_1, X_2, \dots, X_n değişkenleri üzerinde aşağıdaki çarpım ile verilir.

$$p(x_1, x_2, \dots, x_n) = \prod_{i=1}^n p(x_i | \prod_{X_i}) \quad (2.23)$$

Bu nedenle örneğin, Şekil 2.7' de ağa karşılık gelen birleşik dağılım aşağıdaki denklem ile verilir.

$$p(h, e, r, s, d, w, g) = p(h)p(e)p(r|e)p(s|e, h)p(d|s)p(w|s)p(g|s) \quad (2.24)$$

Burada, küçük harf sembolleri değişkenlerin karşılık gelen belirli değerlerini (örneğin doğru veya yanlış gibi) gösterir.



Şekil 2.7 Örnek bir bayes ağı (Pearl, 1988)

Ağ temsiline avantajı, insanlara “doğrudan bağımlılığın” temel nitelik ilişkisini doğrudan ifade etmeye izin verir. Ağ, doğrudan ve dolaylı olarak bağımlılıkların uyumlu bir kümesini gösterir ve bunu sayısal tahminlerden bağımsız olarak modelin sabit bir parçası olarak korur.

$p(x_1, x_2, \dots, x_n)$ olasılık dağılımı ve değişkenlerin sıralaması verildiğinde YDÇ ataları \prod_{X_i} minimal kümesindeki ve X_i ' nin ebeveynleri olarak gösterilir ve aşağıdaki (2.25) denklemini sağlar.

$$p(x_i | \prod_{X_i}) = p(x_i | x_1, x_2, \dots, x_{i-1}), \quad \prod_{X_i} \subseteq \{X_1, X_2, \dots, X_{i-1}\} \quad (2.25)$$

Denklem (2.25) deki ifade p ' nin bir Bayes ağını gösterir.

Yukarıda bahsedilen Bayes ağ oluşturma işlemini burada açık bir şekilde ifade edelim. Öncelikle, X_1 kök olarak seçilir ve bu $p(x_1, x_2, \dots, x_n)$ ile yazılan $p(x_1)$ marjinal olasılığına atanır. Daha sonraki aşamada, X_2 ' yi temsil etmesi için bir düğüm oluşturulur. Eğer X_2 , X_1 ' e bağımlı ise, X_1 ' den X_2 ' ye bir bağlantı kurulur ve bu bağımlılık $p(x_2 | x_1)$ olasılığı ile ölçülür. Diğer taraftan, X_1 ve X_2 bağlantısız bırakılır ve $p(x_2)$ önceki olasılığı X_2 düğümüne atanır. i . aşamada X_i düğümü oluşturulur ve \prod_{X_i} ebeveyn kümesinden X_i düğümüne yönlü doğru bağlantılar grubu çizilir. Bu da yukarıdaki denklem (2.25) de ifade edilmişti ve bağlantıların bu grubu $p(x_i | \prod_{X_i})$ koşullu olasılığı ile ölçülür. Diğer taraftan YDÇ'lerin bağlantıları üzerindeki $p(x_i | \prod_{X_i})$ koşullu olasılıkları orijinal dağılım fonksiyonunu yeniden yapılandırmak için tüm gerekli bilgiyi içermelidir. Denklem (2.25) kullanılarak aşağıdaki denklem (2.26) elde edilir.

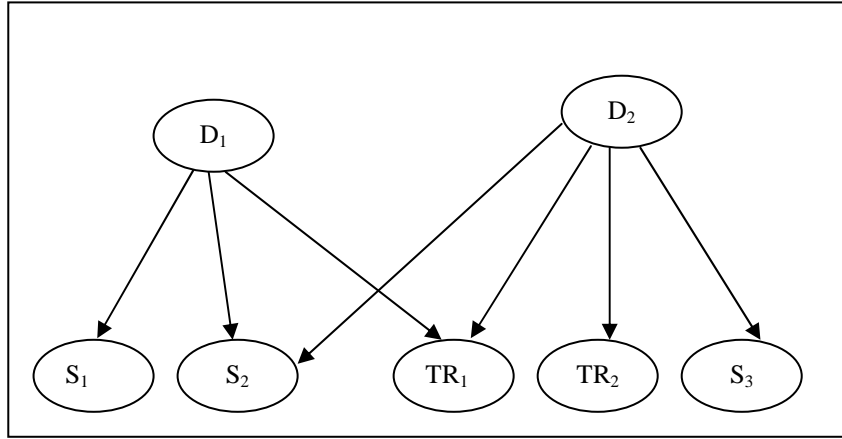
$$p(x_1, x_2, \dots, x_n) = p(x_n | x_{n-1}, \dots, x_1) p(x_{n-1} | x_{n-2}, \dots, x_1) \dots p(x_3 | x_2, x_1) p(x_2 | x_1) p(x_1)$$

$$p(x_1, x_2, \dots, x_n) = \prod_i p(x_i | \prod_{X_i}) \quad (2.26)$$

Burada, X_i 'nin ebeveynleri X_i 'nin doğrudan nedenleri veya X_i 'de doğrudan etkileri olan değişkenlerdir (Pearl, 1988).

2.2.1 Bayes Ağ Tabanlı Teşhis Modeli

Tıbbi teşhis problemi belirtilerin (klinik veriler), işaretlerin veya test sonuçlarının (hastaya uygulanan testler) bir kümesi verildiğinde hangi hastalıkların bu belirli bulguları gösterdiğini tanımlayan patolojik durumları değerler. Tıbbi teşhisin bağlamı ile ilgilenerek bir Bayes Ağ hastalıkların ve belirtilerin de test sonuçları gibi grafiksel olarak düğüm şeklinde temsil edildiği döngüsüz bir çizge ile ifade edilir.



Şekil 2.8 Klasik bir teşhis problemini modelleyen bir bayes ağı örneği (Milho ve Fred, 2000)

Şekil 2.8' de, kök düğümler (D- düğümleri) hastalıkları temsil ederken, torunları (soyundan gelenler) ise belirtileri ve test sonuçlarını (S- düğümleri ve TR- düğümleri) modeller. Bir D- düğümü eğer S-düğümündeki belirti için ilk olasılıksal neden ise o halde S-düğümüne bağlanır. Model nedensel bir yapıda oluşturulduğunda (ağdaki oklar nedenlerden sonuçlara doğru gider) teşhis probleminin grafiksel temsilini tasarlamak mantıksal olarak doğrudur ve etki alanındaki tıbbi bilgiyi kullanır. Teşhis sistemi için temel görev olasılıksal bir çerçeve altında sorgu değişkenleri kümesi için sonra gelen olasılık dağılımını hesaplamaktır. Bazı kanıt değişkenleri için belirli değerler verildiğinde, yani sistem $p(\text{Sorgu}|\text{Kanıt})$ olasılığını hesaplar. Şekil 2.8'deki örnekte D düğümü D_1 olası bir sorgu değişkenidir ve S_1 ile TR_1 'de kanıt değişkenleri olarak kullanılabilir, tipik teşhis çıkarımı olarak Bayes ağlarında her hangi

bir düğüm bir sorgu olarak veya bir kanıt değişkeni olarak kullanılabilir. Dört tür farklı çıkarım görülebilir:

- Teşhis çıkarımı: (etkilerden nedenlere), örneğin, S_1 belirtisi verildiğinde, patoloji D_1 'in olasılığını ifade eder. ($p(D_1|S_1)$)
- Nedensel çıkarım: (nedenlerden etkilere), örneğin, D_2 hastalığı verildiğinde en olası belirtiler bulunur. ($p(S_i|D_2)$)
- Nedenler arası çıkarım: (genel bir etkinin nedenleri arasında), örneğin, S_2 verildiğinde $p(D_1|S_2)$ çıkarımı yapılır ancak D_2 'nin doğru olduğu kanıtı eklenirse, D_1 'in olasılığını düşürür. D_1 ve D_2 düğümleri bağımsız olmasına rağmen, birinin varlığı diğerini daha az olası yapar.
- Karışık çıkarım: Yukarıdaki çıkarımlarının iki veya daha fazlasını birleştiren çıkarımdır (Milho ve Fred, 2000; Russell ve Norvig, 2003).

2.3 Bayes Ağlarında Çıkarım Yöntemleri

Bayes ağlarında kullanılan iki tür çıkarım yöntemi vardır. Bunlar, tam çıkarım ve yaklaşık çıkarım yöntemleridir.

Bayes ağlarına dayalı modellerde değişkenlerin sayısı yüzlerce boyuta ulaşabilmektedir. BA için birçok tam çıkarım algoritmaları geliştirilmiş olmalarına rağmen Bayes ağlarında inanç (belief) güncellemesi NP-zordur (NP-Hard) (Cooper, 1990) ve bu algoritmaların çok geniş ve karmaşık modellere uygulanmaları pratik değildir. Bu nedenle, yaklaşık çıkarım algoritmalarına odaklanmak önemlidir (Lin ve Druzdzel, 1999).

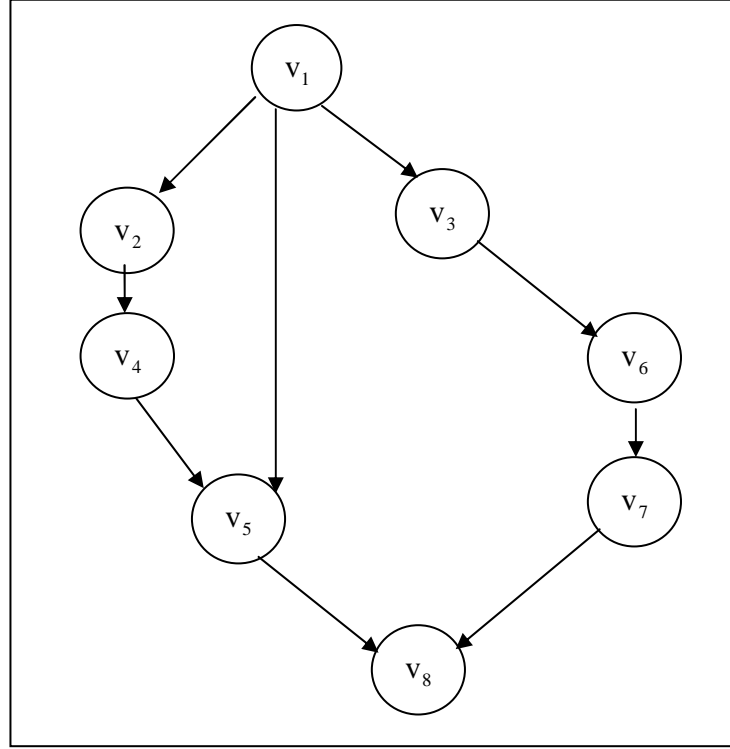
2.3.1 Tam (Kesin) Çıkarım

Doktora tezi kapsamında, kullanacağımız ALARM Ağ yapısında çıkarım yöntemleri olarak Pearl'in (1986a; 1986b) "*döngü kesi kümesi koşullaması algoritması*" (*loop cutset conditioning algorithm*) ve Lauritzen ve Spiegelhalter' in (1988) "*klik ağacı yayılım algoritması*" (*clique-tree propagation algorithm*) kullanılır. Her iki yöntemde tam çıkarım yöntemleri olarak bilinir. Bu bölümde bu iki algoritma ayrıntılı olarak incelenecektir. Diğer yöntemlerden ise kısaca bahsedilecektir. Olasılıksal çıkarım için Pearl' in algoritması hesaplama mimarisi için bir Bayes ağının ikili çizgesini ele alır. Çizgedeki köşeler, bağımsız nesnelere ve yaylar ise iki yönlü iletişim kanalları olarak görülür. Her bir köşenin, basit önceden tanımlanmış hesaplamalar ve kendi ilişkili olasılıkların depolandığı bir yerel hafızası olan bir yerel işlemcisi vardır. İletişim kanalları yoluyla köşeler temsil edilen olasılık dağılımı ile bilgi sağlayan mesajları birbirlerine yollarlar. Her bir köşe hafızasında depolanan

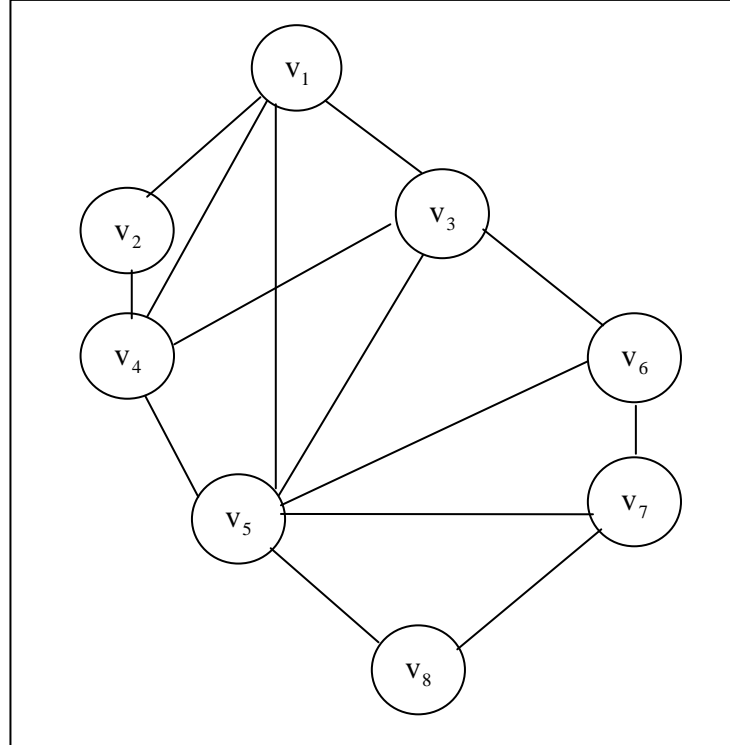
olasılıklardan kendi değerlerinin olasılıklarını ve komşularından aldığı bilgiyi hesaplayabilir. Bir köşenin gerçek değeri bilinir duruma geldiğinde bu köşenin komşularına gönderdiği mesaj kanıtı yansıtmak için güncellenir ve komşularının da güncellenmiş mesajlarını hesaplamalarını zorlar. Kanıtın etkisi böylece komşular arasında mesaj geçişi yoluyla çizgede yayılır. Komşuluğu olan düğümler arasında mesaj geçişi bir Bayes ağı ile doğru olasılıksal çıkarım için yeterli olur ve bir tek bağlantılı ikili çizgeyi ihtiva eder. Böyle bir ağ için Pearl algoritmasının hesaplama karmaşıklığı $O(n.d.2^d)$ şeklinde gösterilir. Burada, n ağın ikili çizgesindeki köşelerin sayısıdır ve d ikili çizgenin (azami) mertebesidir. Eğer ikili çizgenin mertebesi $\Omega(n)$ ise, algoritmanın karmaşıklığı köşe sayılarının üsseli şeklindedir. Bununla beraber eğer Bayes ağının ikili çizgesinin mertebesi bir sabit ile sınırlı ise algoritma lineer zaman alır. Çoklu bağlantılı ikili çizgeyi ihtiva eden BA için kendi temelini teşkil eden çizgesinde bir veya daha fazla döngülere sahip komşuluğu olan köşeler arasında mesaj geçişi doğru olasılıksal çıkarım için yeterli olmaz. Köşeler kendi komşularına güncellenmiş mesajları belirsiz olarak gönderebilir. Bu ağlar için mesaj geçişi, döngü kesi kümesi koşulları olarak adlandırılan bir yöntem ile tamamlanır (Pearl, 1988; Suermondt ve Cooper, 1990). Bu yöntemin temel fikri ikili çizgedeki tüm döngüleri kesmek ve böylece bunların tek bağlantılıymış gibi davranmalarına izin vermektir.

$G = (V, E)$ bir çizge olsun. Eğer G çizgesinde her bir c döngüsü için c üzerinde ayrılan bir kenar ile $V_i \in L$ köşesi varsa $L \subseteq V$ kümesi G çizgesi için bir döngü kesi kümesidir.

Genel olarak, çoklu bağlantılı ikili çizge birçok farklı döngü kesi kümesine imkân verir.



Şekil 2.9 Bir G ikili çizge örneği (Gaag ve Bodlaender, 1997)



Şekil 2.10 G çizgesi için üçgenleştirilmiş H moral çizge örneği (Gaag ve Bodlaender, 1997)

Çoklu bağlantılı ikili çizgeyi ihtiva eden bir Bayes ağı için Pearl' in hesaplama karmaşıklığı $O(n.d.2^{d+l})$ ye eşittir. Burada, n ağın ikili çizgesindeki köşelerin sayısıdır ve d ikili çizgenin (azami) mertebesidir ve l ise ikili çizge için seçilmiş olan döngü kesi kümesindeki köşelerin sayısıdır. Algoritmanın karmaşıklığı kullanılan döngü kesi kümesinin boyutunda üsseldir. Pearl algoritmasının hesaplama karmaşıklığı döngü kesi kümesinin boyutu ile üssel olduğundan pratik uygulamalarda kullanılan en iyi döngü kesi kümesi asgari sayıda köşeler ile bir döngü kesi kümesidir. Çoklu bağlantılı ikili çizgeyi bulma probleminde en küçük döngü kesi kümesi NP-zor olarak bilinir (Suermondt ve Cooper, 1990). Bu nedenle çoklu bağlantılı bir ağ için döngü kesi kümesinin boyutunu minimize etmek hesaplama karmaşıklığını azaltmak için oldukça önemlidir.

Çizgedeki tüm döngüleri kesmek için gerekli olan köşelerin asgari sayısının en çok iki kez çoklu bağlantılı ikili çizge için bir döngü kesi kümesi bulan verimli bir sezgisel algoritma mevcuttur (Becker ve Geiger, 1996; Gaag ve Bodlaender, 1997).

Lauritzen ve Spiegelhalter' in "*klik ağacı yayılım algoritmasında*" olasılıksal çıkarım için Bayes ağının ikili çizgesi, Pearl' in algoritmasında olduğu gibi hesaplamalı bir mimari olarak doğrudan bir şekilde işletilmemektedir. Onun yerine ikili çizge ilk olarak yönsüz üçgenleştirilmiş çizgeye dönüştürülür. Yani dört veya daha fazla uzunlukta, hiçbir dairesel olmayan yönsüz bir çizgedir.

$G = (V, E)$ döngüsüz bir ikili çizge olsun. G için üçgenleştirilmiş bir moral çizge, yönsüz bir üçgenleştirilmiş bir $H = (V, F)$ çizgesidir.

Şöyle ki,

- Eğer, $v_i \rightarrow v_j \in E$ ise o zaman, $v_i - v_j \in F$,
- Eğer, $v_i \rightarrow v_j, v_k \rightarrow v_j \in E$ ise o zaman, $v_i - v_k \in F$.

Genel olarak bir döngüsüz çizge birçok farklı üçgenleştirilmiş moral çizgeye izin verir.

Şekil 2.9' deki G ikili çizgesini dikkate alalım. Şekil 2.10'de gösterilen H yönsüz çizgesi, G için bir üçgenleştirilmiş moral çizgedir. Herhangi bir köşe için, öncelinin orijinal kümesi H 'nin aynı (azami) kliğine dahildir.

Bayes ağının üçgenleştirilmiş moral çizgesinde bir klik ağacı veya bağlantı ağacı oluşturulur (Jensen, 1996).

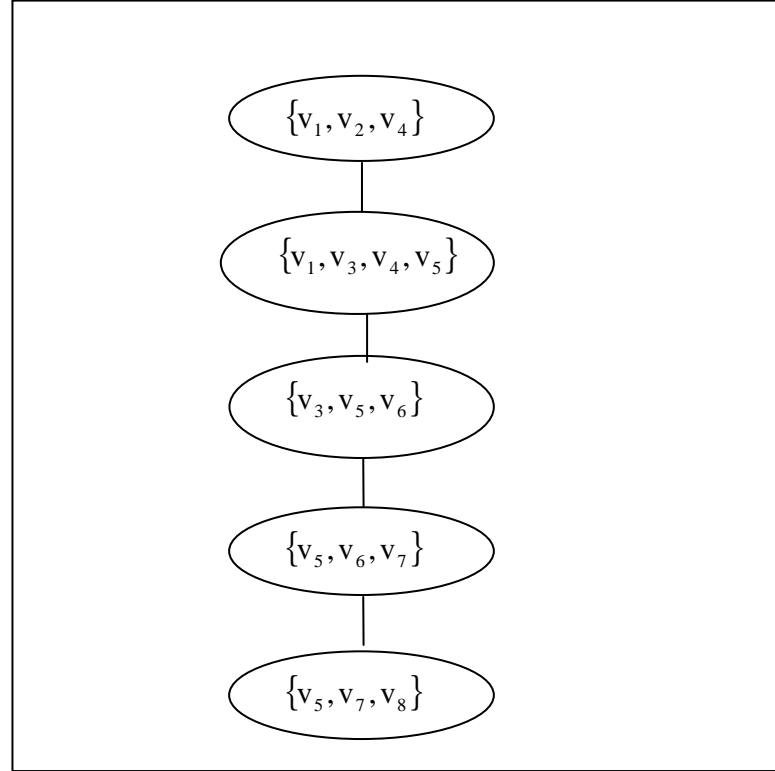
Bir klik ağacında düğümler yapılandırıldığı çizgenin azami kliklerini temsil eder ve klikler arasındaki kesişimler ağacın kenarlarını meydana getirir.

$G = (V, E)$ döngüsüz bir ikili çizge ve H , G için bir üçgenleştirilmiş moral çizge olsun, $C = \{C_i | i=1,2,\dots,n\}$ H 'nin klikleri kümesi olsun. Her bir $C_i \in C$ klik için V_i bu

kliklerin düğüm kümesi olsun. G için bir klik ağacı öyle bir ağaçtır ki, burada $T = (C, F)$ F kenarların kümesi aşağıdaki özelliği sağlar:

Herhangi $C_i, C_j \in C$ klikleri için ve yoldaki her bir C_k klik için T ağacında C_i 'den C_j 'ye doğru $V_i \cap V_j \subseteq V_k$ kümesi elde edilir.

Genel olarak, üçgenleştirilmiş moral çizge çeşitli farklı klik ağaçlarına izin verir. Üçgenleştirilmiş bir çizgede bir klik ağacı oluşturmak için verimli algoritmalar mevcuttur (Pearl, 1988; Jensen, 1996).



Şekil 2.11 H çizgesi için klik ağacı örneği (Gaag ve Bodlaender, 1997)

Şekil 2.9' daki G ikili çizgesini ele alalım. Şekil 2.12, Şekil 2.11'deki H üçgenleştirilmiş moral çizgede oluşturulan G için bir klik ağacı gösterir. Lauritzen ve Spiegelhalter' in algoritması hesaplama mimarisi için bir Bayes ağının ikili çizgesinde oluşturulan bir klik ağacını alır. Ağaçtaki klikler bağımsız nesnelere ve ağacın kenarları da iki yönlü iletişim kanalları olarak görülür. Her bir klik basit önceden tanımlanmış olasılıksal hesaplamaları yapabilen ve köşelerindeki marjinal dağılımın depolandığı yerel bir hafızaya sahip olan bir yerel işlemciye sahiptir. İletişim kanalları yoluyla klikler temsil edilen birleşik olasılık dağılımları hakkında bilgi sağlayan mesajları birbirlerine gönderirler. Her bir klik yerel hafızasında depolanan marjinal dağılımdan köşelerindeki güncellenmiş marjinal dağılımı

hesaplayabilir ve komşularından aldığı bilgiyi hesaplayabilir. Lauritzen ve Spiegelhalter algoritmasının hesaplama karmaşıklığı $O(n.2^c)$ 'e eşittir. Burada, n inanç ağının ikili çizgesindeki köşelerin sayısıdır ve c ikili çizgeden oluşturulan klik ağacındaki en büyük klikteki köşelerin sayısıdır. Algoritmanın karmaşıklığı en büyük cliğin boyutuyla üsseldir. Eğer, kullanılan klik ağacında klik boyutları bir sabit ile sınırlandırılmışsa algoritma lineer zaman alır. Lauritzen ve Spiegelhalter algoritmasının hesaplama karmaşıklığı klik boyutu ile üssel olarak ilişkili olduğundan pratik uygulamalarda kullanılacak en iyi klik ağacı azami sayıda köşeler ile kliklerin oluşturduğu bir ağaçtır veya daha kesin olarak en küçük durum uzayından çıkarılan bir klik ağacıdır. Bu ikili çizge bulma probleminde, böyle bir klik ağacını bulma problemi NP-Zor olarak bilinir. Problem, bir çizge için ağaç genişliğini belirleme problemi ile yakından ilgilidir. Bir döngüsüz ikili çizgenin bir klik ağacını bulmak için çeşitli verimli sezgisel algoritmalar mevcuttur. Bununla beraber bu algoritmalar en uygunluk özellikleri göstermezler (Gaag ve Bodlaender, 1997).

Tam çıkarım algoritmalarının birçok sınıfı mevcut bulunmaktadır. Örneğin, “*düğüm indirgeme*” (*node reduction*) yaklaşımı ilk olarak geliştirilen tam çıkarım algoritmalarından biridir, diğerleri de kümeleme ve koşullama gibi yöntemlerdir. “*Sembolik olasılıksal çıkarım*”, (*symbolic probabilistic inference*) olasılıksal çıkarımı kombinatoriyal bir optimizasyon (combinatorial optimization) problemini, en uygun çarpımlama problemi olarak görür. Olasılıksal çıkarım verilen olasılıksal dağılım kümesinde en uygun bir çarpımlama bulma problemidir. Sembolik olasılıksal çıkarım algoritması sembolik ve sorgu güdümlü bir algoritmadır.

“*Türevsel yaklaşım*” (*differential approach*) algoritması bir Bayes ağını çok değişkenli polinomsal olarak derler ve daha sonra bu polinomsalın kısmi türevlerini her bir değişkenle ilgili olarak hesaplar. Bu tür türevler bir kere mevcut olduğunda sabit bir zamandaki olasılıksal sorguların çok geniş sınıflarına cevaplar hesaplanabilir. BA çıkarım algoritmalarının her bir sınıfları için birçok farklı, saf, melez, genelleştirmeler ve sezgisel çözümler olabilir. Örnek olarak, Koşullama algoritmalarının sınıfında “*yerel koşullama*”, (*local conditioning*), “*genel koşullama*”(*global conditioning*), “*dinamik koşullama*” (*dynamic conditioning*) ve “*yineleyen koşullama*”, (*recursive conditioning*) Kümeleme algoritmalarının sınıfında ise “*Shenoy-Shafer*”, “*Hugin*” , ve “*tembel yayılım*” (*lazy propagation*), eleminasyon (eleme) sınıfında ise “*kova elemesi*” (*bucket elimination*) ve “*değişken eleme*”(*variable elimination*) yöntemleri verilebilir (Guo ve Hsu, 2002).

2.3.2 Yaklaşık Çıkarım

Yaklaşık Bayes Ağları çıkarım algoritmaları stokastik benzetim algoritmalarını, model basitleştirme yöntemlerini, arama tabanlı yöntemleri ve döngülü inanç yayılımlarını kapsamaktadır.

2.3.2.1 Stokastik Benzetim Algoritmaları

Stokastik Benzetim Algoritmaları (Stochastic Simulation Algorithms) ayrıca Stokastik Örnekleme (Stochastic Sampling) veya Monte Carlo algoritmaları olarak da adlandırılır ve en iyi bilinen yaklaşık Bayes Ağı çıkarım algoritmalarıdır. Bunlar, rasgele seçilen örnekler kümesini veya modeldeki birleşik olasılık tablolarına göre ağın somutlaştırılmış örneklerini üretir ve daha sonra örnekteki görüntülerin sıklıkları ile sorgu değişkenlerinin yaklaşık olasılıklarını üretirler. Doğruluğu, örneklerin boyutuna bağlıdır. Stokastik Benzetim Algoritmaları büyük bir ailedir. Bunlar iki temel kategoriye ayrılabilir: Önem (büyüklük) Örnekleme (Importance Sampling) ve Markov Zinciri Monte Carlo (Markov Chain Monte Carlo) yöntemleridir.

“*Olasılıksal mantık örnekleme*” (*probabilistic logic sampling*) ilk ve en basit ileri örnekleme algoritmasıdır. Mantık örneklemede, etki oklarını takiben BA tarafından tarif edilen dünyanın tekrarlı benzetimleri işletilir ve kayıt değerler ile uyumsuz olan örnekler atılır ve örnekte benzer şekilde meydana gelen olaylar ile sıklıkları sayılarak sorgu düğümlerinin olasılıklarını tahmin edilir. Hiçbir kanıt gözlemlenmediği zaman, mantık örnekleme çok iyi çalışmaktadır, ancak eğer kanıt varsa üretilen çoğu örnekler uyumsuz ve çok büyük olacaktır. Faydalı örneklerin parçaları (kesirleri) kanıt değişkenlerinin sayısı ile üssel olarak azalır. Birçok kanıt düğümlü geniş bir ağ için kanıtın önceki olasılığı genellikle çok küçüktür ve bu nedenle mantık örnekleme zayıf bir şekilde işler.

“*Olabilirlik ağırlıklandırma*” (*likelihood weighting*) veya “*kanıt ağırlıklandırma*” (*evidence weighting*) mantık örneklemedeki problemi ortadan kaldırmak için tasarlanmıştır. Olabilirlik Ağırlıklandırma da bir kanıt düğümüne her zaman ulaşılabilir, ulaşılan düğüm örneklendirilmez ve uyumsuz örnekler atılır ve örnekler üzerindeki koşullu kanıtın olabilirliği tarafından örnek ağırlıklandırılır. Olabilirlik Ağırlıklandırma genellikle mantık örneklemeden daha hızlı yakınsar ve daha geniş ağlar ile baş edebilir. Olabilirlik Ağırlıklandırma algoritmasının temel zorluğu ve aslında her stokastik örneklemenin zorluğu ihtimal dahilinde olmayan olaylar için yakınsamanın uzun zaman almasıdır. Hem mantıksal örnekleme hem de olabilirlik ağırlıklandırma'nın her ikisi de ileri örnekleme (forward sampling) yöntemleridir. Geri örnekleme (backward sampling) ağdaki düğümlerin topolojik sırasına ters yöndeki kanıt

düğümlerden örnekler üretmeye izin verir. Geri örnekleme düşük olabilirlik kanıtlarla ileri örnekleme göre daha iyi çalışır. Bununla beraber her ikisinde daha zayıf işleyebileceği bazı durumlar vardır. Çünkü bunlar sonraki doğru dağılımlara ulaşmada başarısız olur.

Bu örnekleme yaklaşımlarını geliştirmek için iyi bilinen bir yöntem sonraki dağılımlara bir yaklaşım olarak örnekleme için gözden geçirilmiş “önem (büyüklük)” dağılımı kullanmaktır. Önem (büyüklük) dağılımları iki yolla üretilebilir. Shachter ve Peot iki tür önem örnekleme algoritması tanıtmıştır. Bunlardan biri “*özellik önem örnekleme*” (*self-importance sampling*) ve diğeri de “*sezgisel önem örnekleme*” (*heuristic importance sampling*) yöntemidir. Özellik Önem Örnekleme algoritması, üretilen sonuçları sık olmayan bir şekilde kullanarak kendi önem fonksiyonunu günceller. Sezgisel önem örnekleme kendi önem fonksiyonu kanıt yayılım algoritması ile bağlantılı değiştirilmiş versiyonu gerçekleştirerek önem fonksiyonunu hesaplar. Tüm deneysel sonuçlar bunların olabilirlik ağırlıklandırmalarında daha iyi uygulandığını göstermektedir.

“*Sınır değişimi*” (*bounded variance*), Olabilirlik Ağırlıklandırma nın farklı bir çeşididir. Bu algoritmalar, Olabilirlik Ağırlıklandırma algoritması ve “Durma Kuralı” (Stopping Rule) teoremine dayalıdır. Bunlar Olabilirlik Ağırlıklandırma’ dan daha iyi çalışmaktadır. Bugüne kadar sunulan en etkin stokastik örnekleme algoritması BA için, “*uyarlanabilen önem örnekleme*” (*adaptive importance sampling*) dir. Uyarlanabilen Önem Örnekleme-BA örnekleme varyansını (değişimini), en uygun önem örnekleme fonksiyonuna mümkün olduğunca yakın olan bir örnekleme dağılımı öğrenerek azaltır. Uyarlanabilen Önem Örnekleme-BA algoritması farklı öğrenme seviyelerinde üretilen örnekler için farklı ağırlıklar tanıtır. Uyarlanan Önem Örnekleme- BA geniş ağlar için daha iyi çalışan diğer iki örnek algoritmasına yol açacak şekilde yeni “Durma Kuralları” ile birleştirir. Yukarıdaki tüm stokastik örnekleme algoritmaları, örneklerin birbirine bağımsız olduğu önem örnekleme çerçevesinde genelleştirilebilir. Diğer bir grup stokastik örnekleme algoritmaları, Markov Zincir Monte Carlo yöntemleridir ve burada örnekler bağımlıdır. Bu yöntemler “*Markov Zincir Monte Carlo*” yöntemleri “*Gibbs örnekleme*” (*Gibbs sampling*), “*Metropolis örnekleme*” (*Metropolis sampling*) ve “*Melez Monte Carlo örnekleme*” (*Hybrid Monte Carlo sampling*). Bu algoritmalar koşullu olasılık tablolarında uçdeğer olasılıklar olmadan iyi çalışır. Uçdeğer koşullu olasılıklar olduğu zaman Markov Zincir Monte Carlo örnekleme algoritmalarının yakınsaması çok yavaş olabilir. Diğer örnekleme teknikleri, “*katmanlı (tabakalı) örnekleme*” (*stratified sampling*), “*hiperküp örnekleme*” (*hypercube sampling*), “*kuasi (yarı) Monte Carlo*” (*quasi Monte Carlo*) yöntemleridir. Bu teknikler, düzgün dağılımlardan rasgele örnekler üretme konusunda farklıdır.

2.3.2.2 Model Basitleştirme Yöntemleri

Model basitleştirme yöntemleri (model simplification methods) ilk olarak modeli tam yöntemler ile uygun duruma gelinceye kadar basitleştirir ve daha sonra tam bir algoritma işletir. Bazı yöntemler model karmaşıklığını küçük olasılıkları iptal ederek azaltır. Diğer yöntemler ise zayıf bağımlılıkların çıkarılmasını içerir.

“*Yerelleştirilmiş kısmi değerlendirme*” (*localized partial evaluation*) algoritması ağlardan seçili düğümleri çıkarır.

“*Durum uzay soyutlama*” (*state space abstraction*) modeli basitleştirmek için koşullu olasılık tablolarının kardinalitesini azaltır.

“*Varyasyonel yaklaşım*” (*variational approach*) varyasyonel dönüşümleri tanıtır ve çizgeden düğümleri tek tek uygulanabilir olacak şekilde çizge yeteri kadar seyrek olana kadar bağlantıyı çözer.

“*Sarkar’ in algoritması*” Bayes ağına gerçek ağa en yakın olan en uygun ağaç ayrıştırılabilir temsili bularak yaklaşır.

“*Bağlama özgü yaklaşım*” (*context-specific approximation*) algoritması genel durumdaki yapıyı hesaba katar ve ağı olasılıklardaki farklılıkları kaldırarak basitleştirir.

Diğer kısmi değerlendirme yöntemleri “*artımlı sembolik olasılıksal çıkarım*” (*incremental symbolic probabilistic inference*) algoritması ve “*mini kova*” (*mini bucket*) algoritmasını içerir.

2.3.2.3 Arama Tabanlı Yöntemler

Arama tabanlı yöntemler (search based methods), birleşik olasılık uzayının göreceli küçük parçanın olasılık kütesinin bir büyüklüğünü içerdiğini varsayar. Bu algoritmalar yüksek olasılıklı örnekleri arar ve daha sonra makul bir yaklaşım elde etmek için bunları kullanır. Bu yöntemler, “*Top-N*” arama tabanlı yöntem, “*deterministik yaklaşımı*” (*deterministic approximation*) ve “*örnekle ve topla*” (*sample and accumulate*) yöntemleridir.

2.3.2.4 Döngülü İnanç Yayılımı

Son yıllarda, “*döngülü inanç yayılımı*” (*loopy belief propagation*), Pearl’ in döngüler ile Bayes Ağındaki çokluağaç (polytree) yayılım algoritmasının kullanımı önemli bir konu haline gelmiştir. Araştırmacılar deneysel olarak, döngülü inanç yayılımının hata düzeltme kodlarının içeriğinde ve bilgisayarlı görü alanında etkin olarak uygulandığını göstermiştir. Fakat döngülü bazı diğer çizgeler için inanç yayılımı zayıf sonuçlar verebilir ve hatta yakınsamada başarısız da olabilir (Guo ve Hsu, 2002).

3. TANECİKLİ HESAPLAMA YÖNTEMİ ve KABA KÜMELER KURAMI

3.1 Tanecikli Hesaplama Yöntemi

Tanecikli Hesaplama (TH) (Granular Computing) yönteminin uygulama alanlarından birisi Kaba Kümeler (Rough Sets) kuramıdır. Bu kuramın tanımları ve özellikleri verilmeden önce aşağıda tanecikli hesaplama yönteminin temel kavram ve tanımlarından bahsedilecektir.

3.1.1 Tanecikli Hesaplamanın Temel Kavramları

Herhangi bir sistemi anlayabilmek için, gerekli kilit noktalar; sistemin temel elemanlarını ve onların işlemlerini anlamaktır. Tanecikli hesaplamanın temel bileşenleri taneciklerdir. Tanecikler üzerindeki işlemler ise tanecikleştirme olarak adlandırılmaktadır. Burada bu kavramlar kısaca açıklanacaktır.

Tanecik(Granule): Bir taneciğin sözlük anlamı “*küçük bir parçacık, özellikle büyük birimleri oluşturan sayısız parçacıklardan biri*” olarak tanımlanmaktadır. Tanecikli hesaplamadaki tanecik tanımı ise yukarıdaki bu tanıma oldukça benzerdir. Özellikle, herhangi bir alt küme, sınıf, nesne veya evrenin kümelenmesi tanecik olarak adlandırılmaktadır. Bu tanecikler, ayırt edilemezlik (indistinguishability), benzerlik (similarity) ve fonksiyonellik (functionality) ile bir araya gelen daha küçük taneciklerden oluşmaktadır. Kavramlar veya nesnelere grubu, uzaysal komşuluk (spatial neighborhood), kapalılık (closeness), uyuşma (cohesion) vb. ile tanecik olarak dikkate alınırlar. Tanecikli hesaplamanın, kesin olmayan, belirsizlik ve kısmi gerçek ile ilgilenmesi beklense de, tanecikler kesin veya bulanık biçimde olabilmektedir. Bir tanecik, bir parçacık modelde kullanıldığı zaman farklı bir biçime ve anlama sahip olabilir. En düşük seviyedeki tanecikler, kullanılan parçacıklar modelin temel parçacıklarından veya elemanlarından oluşmaktadır. Örneğin, bir makale evrenindeki en iyi tanecikler kelimelerdir. Bunlar, temel parçacıklardan oluşmuştur; harfler bazı özel durumlarda tek çocuklu (singleton) tanecikler olarak adlandırılır.

Tanecikleştirme (Granulation): Sözlük anlamı ile tanecikleştirme teriminin orijinal anlamı “*taneciklere yapılandırma süreci veya eylemidir*”. Bu büyük bir nesneyi daha küçük parçalara ayırma sürecini içerir. L.A. Zadeh bu fikri, yayınladığı tanecikli hesaplama makalesinde evreni taneciklere ayırma olarak benimsemiştir. Zadeh tanecikleştirmeyi “*tanecikleştirme bir bütünü parçalara ayırmayı içerir*” şeklinde tanımlamaktadır. Bunun aksine, organizasyon parçaları bütünleştirmeyi içermektedir. Bu tanıma dayalı olarak tanecikli hesaplama iki işlem bulunmaktadır bunlar, tanecikleştirme ve organizasyon işlemleridir. Tanecikli hesaplama tanecikleştirme yapılandırma ve ayırma işlemi içermektedir.

Yapılandırma ise, daha küçük ve daha düşük seviyedeki alt tanecikler ile daha büyük ve daha yüksek seviyedeki taneciği oluşturma sürecini içeren bir aşağıdan yukarı süreçtir. Ayırıştırma ise daha geniş bir taneciği sözlük tanımındaki ve Zadeh' in tanecikleştirme tanımına benzer şekilde daha küçük ve daha alt seviyedeki taneciklere bölme sürecini içeren yukarıdan aşağı bir süreçtir. Yapılandırma ve ayırıştırma birbiri ile sıkı bir şekilde ilişkili süreçlerdir (J.T.Yao, 2005).

3.1.2 Tanecikli Hesaplama ve Veri Madenciliği

Karar Destek Sistemlerinin iki temel grupta incelendiğine ikinci bölümde değinilmişti. Bunlardan biri olan veri güdümlü sistemler, organizasyon içindeki, veri ambarları gibi çok geniş veri havuzlarını inceler ve VM yöntemleri ile karar sürecinde yer alırlar. Tanecikli hesaplama yöntemi giderek gelişen bir araştırma alanı olarak veri madenciliğindeki birçok konuyu çalışmak için kavramsal bir çerçeve sağlar.

VM veride gömülü olan bilgiyi keşfetmeyi amaçlar. Kurallar, en çok kullanılan bilgi temsili yöntemlerinden birisidir. Karakteristiklerine dayalı olarak farklı türde kurallar oluşturulabilir. VM için tanecikli hesaplama ile ilgili olarak birçok çalışma vardır (Lin, 1999; Yao ve Zhong, 1999; Yao, 2001; Tsumoto, v.d., 2002; Li ve Zhong, 2003).

Mevcut bazı yöntemler çeşitli bakış açılarından incelenebilir. Burada kural madenciliğinden bahsedilecektir.

Kural Temsili/Yorumu: Bulanık küme kuramının kilit bir fikri dilsel değişkenlerdir. Bir bulanık tanecik, genelleştirilmiş kısıtları ile ilgili olarak tanımlanabilir ve bulanık tanecikler doğal bir dilin kelimeleri ile temsil edilebilir (Zadeh, 1997). Bir kural iki tanecik arasındaki bağlantıyı özetler. Sonuç olarak, kuralların insanlar tarafından anlaşılabilir ve doğal bir yorumu elde edilir (Hirota ve Pedrycz, 1999; Mitra, vd., 2002).

Tanecikli hesaplamanın somut bir örneği olarak kaba küme kuramı VM alanına uygulanmıştır (Polkowski ve Skowron, 1998). Bu bağlamda kurallar, tanımlanabilir tanecikler olarak ifade edilir. Kuralların özellikleri, kurallar tarafından içerilen taneciklere dayalı olarak yorumlanabilir. Kaba kümeler kuramının VM, bilgi keşfi, ZVA alanında faydalı olduğu kanıtlanmıştır. Örneğin, Polkowski ve Skowron (1998), Tsumoto(1998; 1999), Skowron ve Stepianiuk (1999), Yao ve Zhong (1999), Peters vd. (2002), Zhang vd. (2003) ve diğer birçok araştırmacı kuralları, tanecik özelliklerine ve tanecikler arasındaki ilişkilerin içerimine dayalı olarak yorumlamıştır. Bu kuram, VM uygulamaları için sağlam bir temel oluşturur. Kuram, verideki gizli örüntüleri keşfetmeye yarayan etkin matematiksel araçlar sağlar. Örneğin, verilerin temsili ve sınıflandırılması veya bilgiyi tanımlama buna örnek olarak verilebilir. Veri

tabanlarındaki kısmi veya toplam bağımlılıkların (sebeup sonu) ilişkilerini tanımlar. Artık veriyi elimine eder ve boş deęerleri, kayıplı veriye, dinamik veriye ve dięerlerine yaklařtırır. ok geniř veri tabanlarındaki, VM yntemlerinin kaba kmeleri kullanması son zamanlarda en ok arařtırılan konuların bařında gelmektedir. Ařaęıda verilen problemlerin yapılanması ile ilgili kavram yaklařımlarının birleřiminde bazı nemli adımlar vardır.

- Daha karmařık kavramların yapılandırılacaęı yaklařımlar ile ilgili ilkel kavramlar
- Yakın kavramlar arasındaki benzerlik lmleri
- İlkel olan kavramlardan daha karmařık olan kavramların yapılandırılması iin iřlemler

Bu problemler klasik Kaba Kme yaklařımı ile Kaba Kme kuramının yeni aılımlarının birleřtirilmesi ile özlebilir, rneęin VM baęlamında, Deęiřken Duyarlıklı Kaba Kme (Ziarko, 1993) modelinin yaklařım blgesinin tanımlarını esnek bir Őekilde kontrol edebilme becerisi veride mevcut olan olasılıksal ilişkileri etkili bir Őekilde yakalamaya izin verir.

Bu adımlardaki oluřan problemleri özmek iin yntemler aynı zamanda Bilgi Keřfi ve VM iin ok nemlidir. Veriden rntleri ıkarmak iin yeni yntemler, karar tablolarının ayrıřtırılması, VM iin yeni bir metodoloji ve oklu etmenli sistemler geliřtirilmiřtir (Komorowski, vd.,1998).

Kural Madencilięi: TH teknikleri kural madencilięine (rule mining) uygulanabilir. Daha genel veya anlamlı kurallar madenleyebilmek iin zellik deęerleri arasındaki anlamsal ilişkiler dikkate alınarak taneciklere gruplanabilir veya tanecikler hiyerarřisine (rneęin kavram hiyerarřisi) gruplanabilir (Han, vd., 1993). rneęin, Zhong (2000) kural madencilięinde zellik deęerlerinin taneciklerini kullanmıřtır.

Lin, (1999; 2000) tanecikli hesaplamaya dayalı kural madencilięini yeniden formle etmiřtir ve makine ynelimli bir ereve sunmuřtur. Verilen bir deęer zellięi, bit katarı olarak dnřtrlerek kodlanan deęere sahip olan nesnelere kmesi tarafından temsil edilir. VM sreci daha sonra bit katarlarındaki iřlemler yoluyla yrtlr. Hiyerarřik yapılar kullanılarak tanecikli hesaplama verinin hiyerarřik olarak yorumlanmasını saęlar. Bylece veri tanecięin farklı seviyelerine dnřtrlebilir. rneęin Hirota ve Pedrycz (1999) tanecięin farklı seviyelerinin dilsel tanecikler yolu ile veri dnřtm yoluyla elde edilebildięi VM' nin piramit yapısını ele almıřtır.

Dięer yntemler ile birleřimler: TH yeni veya daha verimli madencilik yntemleri ile retilenler iin dięer yntemler ile birleřtirilebilir. Hesaplamalı zek (computational intelligence) baęlamında Hirota ve Pedrycz (1999) nrohesaplama (sinirsel hesaplama)

(neurocomputing), evrimsel hesaplama (evolutionary computing) ve tanecikli hesaplamanın (özellikle kaba kümeler ve bulanık kümeler) birbiri ile değerlendirilebileceğini belirtmiştir. Çoğu araştırmacı veri madenciliği için tanecikli hesaplama ve diğer yöntemleri birleştirmeye çalışmıştır. Örneğin, Zhang vd.(2000) VM için tanecikli hesaplama ve nörohesaplamanın (sinirsel hesaplama) birleşimini ele almıştır.

VM' nin çerçevesine dayalı olarak bir tanecikli hesaplama oluşturmak için aşağıdaki varsayımlar benimsenir:

- **Bilgi taneciği:** Her bir tanecik bir bilgi parçasını temsil eder.
- **Yapısal Bilgi:** Bilgi taneciklerinin bir ağ arasındaki bağlantıları yapısal bilgiyi temsil eder.
- **Madencilik Görevi:** Anlamlı bilgi taneciklerini ve yapısal bilgiyi aramak veri madenciliğinin temel görevidir.

Bunlar tanecikli hesaplamanın temel görevi ve temel bileşenleri ile ilişkilidir.

3.1.2.1 Bilgi Tanecikleri

Bir taneciğe bir bilgi parçası olarak bakabilmemiz için bir taneciği isimlendirmemiz veya temsil etmemiz gerekir. Bu amaçla, tanecikleri iyi bilinen kavram fikri ile ilişkilendiririz. Kavramlar, insan zekâsı ve hesaplamasının altında yatan düşüncenin temel birimleridir. Kavramların, kavram oluşumunun ve öğrenmenin birçok kuramsal görüşü vardır. Klasik görüş, kavramlara iyi tanımlanmış sınır hattı olan birleşik yeterli ve tek gerekli koşulların kümesi tarafından tarif edilebilir bir varlık olarak davranır. Diğer görüşler, prototip, örnekleme, çerçeve ve kuram görüşünü içerir. Her bir görüş kavramların belirli yönlerini yakalar ve kavram oluşumu ve öğrenimi için farklı bir anlama sahiptir. Tümevarımsal veri analizi (inductive data analysis) için farklı görüşlerin uygulaması birçok araştırmacı tarafından incelenmiştir. Her kavram, içerik ve kapsam olarak iki bölümden oluşur ve buna dayalı olarak bir dil tanımlanabilir. Böylece, bir kavramın içeriği dilin bir formülü olarak ve kapsamı da formülü tatmin eden nesnelere kümesi olarak ifade edilir (Yao ve Liau, 2002; Yao, 2004a).

VM için kavramların kapsamı örneklerin belirli bir eğitim kümesi ile ilgili olarak tanımlanır. Bu durumda taneciklerin tanımlanabilirliği ve tanımlanamayan taneciklerin yaklaşımlarının dikkate alınması gerekir (Yao, 2004b).

3.1.2.2 Yapısal Bilgi

Bilgi tanecikleri yapısal bilgiyi elde etmek için yapı taşları olarak kullanılır. İnsan bilgisi kavramsaldır ve bütünü birleştirerek oluşturur. İnsan bilgisini karakterize etmede bağlam ve hiyerarşi durumları ele alınmalıdır. Bu nedenle yapısal bilgi, bilgi taneciklerinin bir ağı arasındaki bağlantılar ile ifade edilir. Tanecikli yapılar, yapısal bilgiyi temsil etmede akla yatkın bir yol sağlar. İçerikler hakkında çıkarım mantığına dayalıdır. VM için, kavramların kapsamları arasındaki ilişkilere dayalı olan kavramların içerikleri arasındaki ilişkilerin elde edilmesi gerekir. Kavramların kapsamları arasındaki bağlantılar yoluyla kavramlar arasında ilişkiler kurulabilir (Yao, 2001; 2003). Tanecikli yapıların sonuçları, yapısal bilgiye kolaylıkla uygulanabilir.

3.1.2.3 Madenleme Görevi

Hem bilgi tanecikleri hem de yapısal bilgi yararlıdır. Kavramsal olarak veri kümesine gizlenmiş olan bilgi, sadece kavramların içerikleri kullanılarak tanımlanan tüm bilgi uzayında gömülüdür. Bu bilgi uzayının sadece küçük bir kısmıdır. Bu nedenle madenlemenin görevi en uygun bilgi taneciklerini ve yapısal bilgiyi aramaktır. Mitchell (1982) tarafından önerilen araştırma olarak öğrenme fikri burada uygulanabilir. Pratik algoritmalara sahip olmak için yönetilebilir büyüklük ile bir araştırma uzayına sahip olmak gerekir. Verilen bir veri kümesi için bilginin farklı türleri bir arada bulunabilir. VM için birçok görüş yaklaşımlarına sahip olmak gerekir. Örneğin, yapısal bilginin iki türü biçimsel kavram kafesleri (Ganter ve Wille, 1999) ve hiyerarşik sınıflardır.

Üç temel konuyu yani, bilgi tanecikleri, madenleme görevi ve yapısal bilgi yeni bir çerçeve olarak VM için bir temel sağlayabilir. Gereksiz ve ilgisiz ayrıntılar ile uğraşmak yerine bu çerçeve güçlü bir kavramsal bakış açısı sunar. Bu çerçeve göz önünde bulundurulmak suretiyle VM' nin mevcut modellerini yorumlamak basit olacaktır.

TH disiplinler arası bir çalışmadır. Her bir disiplin sadece belirli yönleri ele alırken, TH ile çalışmak ise bütünleştirilmiş bir bakış açısı sağlar. Felsefi düzeyde tanecikli hesaplama yapılanmış düşünce ile ilgilenir. Uygulama düzeyinde bunun işletilmesi ise yapılandırılmış bir sorunu çözmeye yol açar. Bu güçlü bakış bize VM için somut bir temel kurmamızı sağlar. Tanecikli hesaplamanın VM için uygulaması bize iki temel noktayı gösterir. Birincisi, tanecikli hesaplama birçok problemi modellemek için kullanılan güçlü bir yöntem olması, diğeri ise birçok alan gibi VM, tanecikli hesaplamanın özelliklerini taşır. VM' nin yeni çerçevesi ilgisiz ayrıntıları ihmal ederek yüksek kavramsal düzeye odaklanır. Bilimsel bir çalışma alanı olarak veri madenciliğine daha fazla bakış açısı sağlar. Üç temel varsayım a)

bilgi tanecikleri olarak tanecikler, b) yapısal bilgi olarak tanecikli yapılar ve c) araştırma olarak madenleme mantıklı olarak görülür. Buradaki varsayımlar aslında örtülü olarak çoğu VM modeli tarafından kullanılır (Yao, 2006).

3.2 Kaba Kümeler Kuramı

TH yönteminin uygulama alanlarından birinin kaba kümeler olduğuna önceki bölümde değinilmişti. Bu bölümde kaba kümeler kuramının genel tanımları, avantajları, uygulama alanları ve özelliklerine değinilmiştir.

Kaba kümeler kuramı, 1980' lerin başlarında Z. Pawlak (1982) tarafından önerilmiştir ve evrendeki her bir nesnenin bilgi elde edebileceğimiz varsayımı üzerine kuruludur (Nguyen ve Slezak, 1999; Pawlak, 2002a).

Birçok kuramsal araştırmalar kaba küme kuramından ilham almıştır. Çoğu araştırmacılar, kaba kümelerin, cebrik ve topolojik özellikleri ile çalışmıştır. Bunun yanı sıra, çeşitli mantıksal araştırmalar, yaklaşık çıkarımı ele alan mantıksal araçların oluşturulmasına yönelmiştir. Birçok görüşe göre, kaba küme kavramı, çapraşıklık (vagueness) ve belirsizlik (uncertainty) ile uğraşmak için farklı matematiksel fikirlerden ortaya çıkmıştır. Özellikle birçok yazarlar, bulanık kümeler ve kaba kümeler arasındaki ilişkilerin aydınlatılmasına yönelik çalışmalar yapmışlardır (Dubois ve Prade, 1992; Pawlak ve Skowron, 1994). Kanıt kuramı ve kaba küme arasındaki ilişkinin belirlenmesine yönelik çalışmalar son zamanlarda Skowron ve Grzymala-Busse (1994) tarafından açıklanmıştır. Veri analizindeki kaba küme felsefesi, istatistiksel yaklaşıma yakındır. Bu iki yöntemin karşılaştırılmaları, Krusinska, vd., (1992) de bulunabilir. Kaba kümelerin istatistiksel ilişkilerin diğer görüşleri, Wong, vd., (1986) göz önüne alınmıştır. Boolean kullanılmada kaba küme kuramı ilişkilerinin önemli konuları Skowron ve Rauszer (1992) tarafından ayrıntılı bir şekilde incelenmiştir (Pawlak, 1994a).

Kaba kümeler kuramında aynı bilgi ile karakterize edilen nesnelere onların mevcut bilgileri ile aynıdır, yani ayırt edilemezdir. Bu yolla üretilen ayırt edilemezlik ilişkisi kaba küme kuramının matematiksel tabanını oluşturmaktadır. Tüm ayırt edilemez aynı nesnelere kümeleri "*temel küme*" olarak adlandırılır ve evren hakkında bilginin temel, yani en küçük yapı taşını (atom) oluşturur. Bazı temel kümelerin herhangi birleşimi "*keskin küme*" olarak adlandırılır, aksi takdirde küme "*kaba küme*" olmaktadır. Her bir kaba küme sınır çizgilere sahiptir. Örneğin, kesinlik ile sınıflandırılmayan nesnelere gibi. Belirgin olarak kaba kümeler, duyarlı kümelerin aksine, elemanları ile ilgili bilgiler olarak karakterize edilemezler. Herhangi bir kaba küme ile bir duyarlı küme çifti, ilişkili olan kaba kümenin alt ve üst yaklaşımı olarak

adlandırılır. Alt yaklaşım (lower approximation), tamamen kümeyle ait olan tüm nesnelere içerir ve üst yaklaşım (upper approximation) ise, olası olarak kümeyle ait olan nesnelere içerir. Alt ve üst yaklaşımlar arasındaki farklar kaba kümenin sınır alanını (boundary region) oluşturur. Alt ve üst yaklaşımlar kaba küme kuramında ki iki temel işlemdir.

Kaba küme yaklaşımı, YZ ile bilişsel bilim (cognitive science), özellikle makine öğrenimi (machine learning), bilgi kazanımı (information gain), karar analizi (decision analysis), veri tabanlarından bilgi keşfi (knowledge discovery in databases), uzman sistemler, tümevarımsal nedenleme (inductive reasoning) konuları içinde büyük önem taşımaktadır. Kaba küme kuramı, mühendislik, finans, bankacılık, tıp gibi birçok alanda günlük yaşam problemlerine uygulanmakta ve bunun yanı sıra, diğer önemli disiplinler arasındaki bağlantıları mevcut bulunmaktadır. Örneğin, matematiksel morfoloji (mathematical morphology), çatışma teorisi (conflict theory), Petri ağları (Petri nets), YSA, GA buna örnek olarak verilebilir (Pawlak, 1994a; 2002a). Veri analizinde kaba küme yaklaşımının birçok önemli avantajı bulunmaktadır. Bunların bazılarını aşağıdaki gibi sıralayabiliriz.

- Minimal veri kümeleri bulur (veri indirgeme) ve veriden karar kuralı kümeleri üretir.
- Elde edilen sonuçların açık bir şekilde yorumlanmasını gerçekleştirir ve verinin anlamlılığını değerlendirir.
- Kaba küme kuramına dayalı çoğu algoritmalar paralel işleme için bilhassa uygundur (Pawlak, 2002a).
- Büyük hacimli ve herhangi türdeki verilerin ele alınması yeteneği, birçok alt sistemin birçok değişkenler tarafından yönetildiği, alt sistemler arasında ilişki olan ve birbirlerinin performansını etkileyen bütün sistem için birkaç en uygun çözümün olduğu mühendislik analiz ve modellenmesinde çok faydalıdır.
- Doğrusal olmayan veya sürekli olmayan fonksiyonel bağıntılar modelleme yeteneği karmaşık, çok boyutlu örüntüleri nitelendirmek için güçlü bir yöntem sağlar. Üretilen kurallar ve kullanılan özellikler gereğinden fazla olmadığından örüntüler çok özlü, güçlü ve sağlamdır. Ayrıca, verideki gizli örüntüleri bulmak için verimli algoritmalar sağlar.
- Kaba kümeler belirli olmayan sistemleri tanımlayabilir ve karakterize edebilirler. Karar vermede olasılıklı bilgiyi içerirler. Bu özellikler teşhise yönelik uygulamalarda (tıbbi teşhis gibi) veya mevcut bilginin hedef kararları tam olarak nitelendirmek için uygun olmadığı durumlarda uzman sistemlerin oluşturulması için çok yararlıdır.

- Kaba kümelerin bilgiyi anlaşılması kolay mantık örüntüleri olarak göstermeleri, bilginin denetlenmesi ve geçerliliğinin ortaya konulması gereken yerlerde veya kararların kurallara ve destekleyen durumlara uyumlu olduğu yerlerde bu yöntemin başarılı olmasını sağlamıştır (Binay, 2002).

3.2.1 Kaba Kümeler Kuramında Temel Kavramlar

Bu bölümde Kaba Kümeler kuramının temel kavramları ve sınıflandırmalar ele alınacaktır.

3.2.1.1 Çapraşıklık ve Sınır Bölgesi

Boş olmayan sonlu bir U evrensel kümesi verilsin ve I 'nin U üzerinde bir ikili bağıntı olduğunu varsayalım. $I(x)$ ile yIx olacak şekilde bütün $y \in U$ kümesi belirtilsin. Eğer, I yansıyan ve simetrik şöyle ki, xIx ise ve her $x, y \in U$ ve xIy için her $x, y \in U$ için yIx ise o halde I bağıntısı bir tolerans bağıntısıdır. Burada eğer I bağıntısının ayrıca geçişken özelliği mevcut ise şöyle ki xIy ve yIz olduğunda xIz ise o halde, I bir denklik bağıntısıdır. Bu durum, $I(x) = [x]_I$ şeklinde ifade edilir. Yani, $I(x)$, x elemanını içeren I bağıntısının bir denklik sınıfıdır. Eğer, I bağıntısı bir tolerans bağıntısı ve xIy ise x, y 'ye I 'ya göre benzer (I -benzer) denir. Bunun yanında eğer, I bir denklik bağıntısı ve xIy ise x, y 'ye göre ayırtedilemez (I -ayırtedilemez) olarak adlandırılır (Pawlak, 1994b). Biçimsel olarak, U evreninin herhangi bir X alt kümesi verildiğinde X 'in alt ve üst yaklaşımları aşağıdaki gibi tanımlanır:

$$I_*(X) = \{x \in U \mid I(x) \subseteq X\} \quad (3.1)$$

$$I^*(X) = \{x \in U \mid I(x) \cap X \neq \emptyset\} \quad (3.2)$$

Burada $I(x)$, x ile ayırtedilemez olan nesnelere kümesini göstermektedir. X 'in sınır bölgesinin kümesi aşağıdaki gibi ifade edilir:

$$BN_I(X) = I^*(X) - I_*(X) \quad (3.3)$$

Eğer, X 'in sınır bölgesi boş küme yani, $BN_I(X) = \emptyset$ ise o halde, X kümesi I ile ilgili kesin olarak adlandırılır Karşıt olarak, eğer $BN_I(X) \neq \emptyset$ ise o halde, X kümesi I ile ilgili kaba olarak adlandırılır (Pawlak, 1994b). Böylece, kaba kümeler, çapraşık (vague) kavramların doğal bir matematiksel modeli olarak görülebilir. Çapraşık kavramların bazı özellikleri aşağıdaki gibidir.

$$i. \quad I_*(X) \subseteq X \subseteq I^*(X) \quad (3.4)$$

$$ii. \quad I_*(\emptyset) = I^*(\emptyset) = \emptyset, \quad I_*(U) = I^*(U) = U \quad (3.5)$$

$$iii. \quad I^*(X \cup Y) = I^*(X) \cup I^*(Y) \quad (3.6)$$

$$iv. \quad I_*(X \cap Y) = I_*(X) \cap I_*(Y) \quad (3.7)$$

$$v. \quad X \subseteq Y, \quad I_*(X) \subseteq I_*(Y) \text{ ve } I^*(X) \subseteq I^*(Y) \quad (3.8)$$

$$vi. \quad I_*(X \cup Y) \supseteq I_*(X) \cup I_*(Y) \quad (3.9)$$

$$vii. \quad I^*(X \cap Y) \subseteq I^*(X) \cap I^*(Y) \quad (3.10)$$

$$viii. \quad I_*(-X) = -I^*(X) \quad (3.11)$$

$$ix. \quad I^*(-X) = -I_*(X) \quad (3.12)$$

$$x. \quad I_*(I_*(X)) = I^*(I_*(X)) = I_*(X) \quad (3.13)$$

$$xi. \quad I^*(I^*(X)) = I_*(I^*(X)) = I^*(X) \quad (3.14)$$

Çapraşıklık sayısal olarak aşağıdaki katsayı tanımlanarak karakterize edilebilir:

$$\alpha_I(X) = \frac{|I_*(X)|}{|I^*(X)|} \text{ veya } \alpha_I(X) = \frac{\text{card}(I_*(X))}{\text{card}(I^*(X))} \quad (3.15)$$

şeklinde ifade edilir.

Burada, $|I_*(X)|$, $I_*(X)$ kümesinin eleman sayısını, $|I^*(X)|$ ise $I^*(X)$ kümesinin eleman sayısını gösterir. Açıkça görülmektedir ki, $\alpha_I(X)$ 0 ile 1 aralığında değer alacaktır, şöyle ki, $0 \leq \alpha_I(X) \leq 1$. Eğer, $\alpha_I(X) = 1$ ise o halde X kümesi I ile ilgili olarak kesindir. Aksi takdirde, eğer $\alpha_I(X) < 1$ ise o halde, X kümesi I ile ilgili olarak kabadır (Pawlak, 1994a). Böylece, $\alpha_I(X)$ katsayısı, X kümesinin yaklaşımının doğruluğu olarak anlaşılabilir.

3.2.1.2 Belirsizlik ve Kaba Üyelik Fonksiyonu

Bir çapraşık kavram sınır çizgisi durumlarına sahiptir, şöyle ki, evrenin kesinlikle kavramın elemanları olarak sınıflandırılmayan elemanları mevcuttur. Bu nedenle belirsizlik elemanların küme üyeliği ile ilgilidir. Dolayısıyla, kaba küme bakış açısından belirsizlik problemini tartışmak için kaba küme kavramı ile ilgili olarak üyelik fonksiyonu (kaba üyelik fonksiyonu) tanımlanmalıdır. Kaba üyelik fonksiyonu, I ayırtedilemezlik bağıntısı kullanılarak aşağıdaki gibi tanımlanabilir:

$$\mu_X^I(x) = \frac{|X \cap I(x)|}{|I(x)|} = \frac{\text{card}(X \cap I(x))}{\text{card}(I(x))} \quad (3.16)$$

Burada açıkça görülebilir ki, kaba üyelik fonksiyonu 0 ile 1 arasında (0 ve 1 dahil) bir değer alacaktır. ($0 \leq \mu_X^I(x) \leq 1$)

Kaba üyelik fonksiyonu, yaklaşımlar ve bir kümenin sınır bölgesi kullanılarak aşağıdaki gibi tanımlanabilir:

$$I_*(X) = \{x \in U \mid \mu_X^I(x) = 1\} \quad (3.17)$$

$$I^*(X) = \{x \in U \mid \mu_X^I(x) > 0\} \quad (3.18)$$

$$BN_I(X) = I^*(X) - I_*(X) = \{x \in U \mid 0 < \mu_X^I(x) < 1\} \quad (3.19)$$

Yukarıdaki tanımlardan görülebilir ki, kaba küme kuramında belirsizlik ve çapraşıklık arasında sıkı bir ilişki vardır. Çapraşıklık kümeler ile ilgili iken, belirsizlik ise kümenin elemanları ile ilişkilidir. Bu nedenle, belirsiz veriler dikkate alındığında kaba üyeliğin gerekli olduğu gibi, çapraşıklık kavramlarından bahsederken yaklaşımlar gereklidir (Pawlak, vd., 1995).

Kaba üyelik fonksiyonunun aşağıdaki özelliklere sahip olduğu Pawlak (1994b) tarafından gösterilmiştir:

- i. Sadece ve sadece $x \in I_*(X)$ ise, $\mu_X^I(x) = 1$
- ii. Sadece ve sadece $x \in U - I^*(X)$ ise, $\mu_X^I(x) = 0$
- iii. Sadece ve sadece $x \in BN_I(X)$ ise, $0 < \mu_X^I(x) < 1$
- iv. Eğer, $I = \{(x, x) \mid x \in U\}$ ise o halde, $\mu_X^I(x)$, X 'in karakteristik fonksiyonudur.
- v. Eğer, xIy ise o halde, $\mu_X^I(x) = \mu_X^I(y)$ olduğunda I denklik bağıntısı sağlanır.
- vi. Her $x \in U$ için, $\mu_{U-X}^I(x) = 1 - \mu_X^I(x)$
- vii. Her $x \in U$ için, $\mu_{X \cup Y}^I(x) \geq \max(\mu_X^I(x), \mu_Y^I(x))$
- viii. Her $x \in U$ için, $\mu_{X \cap Y}^I(x) \leq \min(\mu_X^I(x), \mu_Y^I(x))$
- ix. Eğer, \mathbf{X} U 'nin çift ayrık kümelerinin bir ailesi ise o halde, herhangi $x \in U$ için $\mu_{\cup \mathbf{X}}^I(x) = \sum_{x \in \mathbf{X}} \mu_X^I(x)$, I denklik bağıntısını sağlar (Pawlak, 1996).

Yukarıdaki fikirler Ziarko (1993) tarafından önerilen değişken duyarlıklı kaba küme modeli ile genelleştirilebilir. ε , $0 \leq \varepsilon < 0.5$ gibi reel bir sayı olsun. Bu durumda alt ve üst yaklaşımları aşağıdaki gibi tanımlayabiliriz.

$$I_{*\varepsilon}(X) = \{x \in U \mid \mu_X^I(x) \geq 1 - \varepsilon\} \quad (3.20)$$

$$I_*^\varepsilon(X) = \{x \in U \mid \mu_X^I(x) > \varepsilon\} \quad (3.21)$$

Burada, eğer $\varepsilon = 0$ ise önceki durum elde edilir.

Ayrıca varsayalım ki, $0.5 < \varepsilon \leq 1$ ise sonuç olarak, aşağıdaki (3.22) ve (3.23) eşitlikleri elde edilir.

$$I_{*\varepsilon}(X) = \{x \in U \mid \mu_X^I(x) \geq \varepsilon\} \quad (3.22)$$

$$I_\varepsilon^*(X) = \{x \in U \mid \mu_X^I(x) > 1 - \varepsilon\} \quad (3.23)$$

Sınır bölgesi ise aşağıdaki gibi ifade edilir.

$$BN_I^\varepsilon(X) = I_\varepsilon^*(X) - I_{*\varepsilon}(X) = \{x \in U \mid \varepsilon < \mu_X^I(x) < 1 - \varepsilon\} \quad (3.24)$$

Aşağıdaki özellikler açıkça görülebilir:

- i. Eğer, $x \in I_{*\varepsilon}(X)$ ise o halde, $I(x) \subseteq X$ olması gerekli değildir.
- ii. Eğer, $x \in U - I_\varepsilon^*(X)$ ise o halde, $X \cap I(x) = \emptyset$ olması gerekli değildir.
- iii. Eğer, $X \cap I(x) \neq \emptyset$ ise o halde, $x \in I_\varepsilon^*(X)$ olması gerekli değildir.
- iv. Eğer, $I(x) \not\subseteq X$ ise o halde, $x \notin I_\varepsilon^*(X)$ olması gerekli değildir.

Bunların yanı sıra aşağıdaki (3.25) ve (3.26) eşitsizliklerine sahibiz.

$$I_*(X) \subseteq I_{*\varepsilon}(X) \quad (3.25)$$

$$I^*(X) \supseteq I_\varepsilon^*(X) \quad (3.26)$$

Değişken duyarlıklı kaba kümeler fikri yaklaşımların tanımlarına yüklenen katı koşul durumlarını daha esnek bir duruma getirir ve orijinal tanımların daha zayıf hali olarak görülür.

Kaba kümelerin değişken duyarlıklı modeli pratik problemlere kaba küme yaklaşımının uygulanabilirliğini geliştirmeyi amaçlayan orijinal kaba küme yaklaşımının bir uzantısı olup yaklaşım bölgelerini tanımlamadaki temel kümeler ile ilişkili koşullu olasılığın parametre kontrollü derecesine dayalıdır. VM bağlamında bu model yaklaşım bölgesinin tanımlarını esnek bir şekilde kontrol edebilme becerisi veride mevcut olan olasılıksal ilişkileri etkili bir şekilde yakalamaya imkân sağlar. Bazı uygulamalarda parametrelerin nasıl ayarlanacağı açık değildir ve bunların yaklaşım bölgelerini tanımlamak için kullanılmasına bazen gerek

duyulmaz. Mevcut bilgiye dayalı olarak, elde edilmesi imkânsız olabilecek şekilde olayın meydana gelmesini tahmin etmek için olasılıksal kural yerine bu uygulamalarda genel amaç bir olayın oluşabileceği veya oluşmayacağı tahmin doğruluğunu arttırmaktır. Örneğin, tıbbi alanda tıbbi testlerin sonuçları belirli bir hastalığın artan olasılığını veya hastalığın muhtemel olmadığını gösterebilir. Testler olmadan meydana gelen hastalığın olasılıkları genel popülasyondaki hastalığın önceki meydana gelme özelliği ile verilir, yani istatistiksel hesaplamaların sonucu popülasyondaki bireylerin tıbbi durumu hakkında herhangi spesifik bilgiyi hesaba katmaz. Bununla beraber, önceki olasılık tek durumlarda olayın (hastalığın) meydana gelme gerçek olasılığını doğru bir şekilde tahmin etmek için yeterli değildir. Bu durumlarda Bayes Kaba Küme Modeli (Slezak ve Ziarko, 2002; 2005) kullanılır. Bayes Kaba Küme modelinin parametre edilmiş bir uyarlaması Değişken Duyarlıklı Bayes Kaba Küme Modelidir. Genel olarak, Değişken Duyarlıklı Bayes Kaba Küme Modeli VM uygulamalarında Değişken Duyarlıklı Kaba Küme Modeli ile Bayes çıkarımının avantajlarını birleştirmeye imkân verir (Slezak ve Ziarko, 2003a; 2003b).

3.2.1.3 Çapraşık Kavramların Sınıflandırılması

Kaba kümelerin dört temel sınıfı mevcuttur. Bunlar aşağıdaki gibidir:

- i. Eğer, $0,1 \in \{\mu'_x(x) | x \in U\}$ ise o halde, X kabaca I gözlemlenebilir.
- ii. Eğer $1 \notin \{\mu'_x(x) | x \in U\}$ ve $0 \in \{\mu'_x(x) | x \in U\}$ ise o halde, X içten I gözlemlenemez.
- iii. Eğer $0 \notin \{\mu'_x(x) | x \in U\}$ ve $1 \in \{\mu'_x(x) | x \in U\}$ ise o halde, X dıştan I gözlemlenemez.
- iv. Eğer, $0,1 \notin \{\mu'_x(x) | x \in U\}$ ise o halde, X tamamen I gözlemlenemez.

Doğrudan yukarıdaki tanımlardan aşağıdaki özelliklere sahibiz:

1. Sadece ve sadece X kabaca I gözlemlenebilir ise $x, y \in U$ ve $I(x) \subseteq X$ ve $I(y) \cap X = \emptyset$ 'dir.
2. Sadece ve sadece X içten I gözlemlenemez ise herhangi $x \in U$ için $I(x) \not\subseteq X$ ve bazı $y \in U$ için $I(y) \cap X = \emptyset$ 'dir.
3. Sadece ve sadece X dıştan I gözlemlenemez ise herhangi $x \in U$ için $I(x) \cap X \neq \emptyset$ ve bazı $y \in U$ için $I(y) \subseteq X$ 'dir.
4. Sadece ve sadece X tamamen I gözlemlenemez ise herhangi $x \in U$ için $I(x) \not\subseteq X$ ve $I(x) \cap X \neq \emptyset$ 'dir

Yukarıdaki kümeler yaklaşımların kavramları kullanılarak da aşağıdaki gibi tanımlanabilir:

- Sadece ve sadece X kabaca I gözlemlenebilir ise, $I_*(X) \neq \emptyset$ ve $I^*(X) \neq U$ 'dir.
- Sadece ve sadece X içten I gözlemlenemez ise, $I_*(X) = \emptyset$ ve $I^*(X) \neq U$ 'dir.
- Sadece ve sadece X dıştan I gözlemlenemez ise, $I_*(X) \neq \emptyset$ ve $I^*(X) = U$ 'dir.
- Sadece ve sadece X tamamen I gözlemlenemez ise, $I_*(X) = \emptyset$ ve $I^*(X) = U$ 'dir.

Yukarıdaki sınıflandırmaların sezgisel anlamı aşağıdaki gibidir:

Eğer X kümesi kabaca I gözlemlenebilir ise, bu durumda, U kümesinin bazı elemanları için X veya $-X$ 'e ait olup olmadığına karar verilebileceği anlamına gelir.

Eğer X kümesi içten I gözlemlenemez ise, bu durumda, U kümesinin bazı elemanları için $-X$ 'e ait olup olmadığına karar verilebileceği fakat U kümesinin herhangi bir elemanının X 'e ait olup olmadığına karar verilemeyeceği anlamına gelir.

Eğer X kümesi dıştan I gözlemlenemez ise, bu durumda, U kümesinin bazı elemanlarının X 'e ait olup olmadığına karar verilebileceği fakat U kümesinin herhangi bir elemanının $-X$ 'e ait olup olmadığına karar verilemeyeceği anlamına gelir.

Eğer X kümesi tamamen I gözlemlenemez ise, bu durumda, U kümesinin hiçbir elemanının X veya $-X$ 'e ait olup olmadığına karar verilemeyeceği anlamına gelir.

Bunun anlamı, X kümesinin, pozitif olarak sınıflandırılabilen evrendeki bazı elemanları mevcut ise kabaca gözlemlenebilir olduğudur. Bu tanım ayrıca, X kümesinin dışında herhangi bir çapraşıklık olmaksızın sınıflandırılabilen diğer bazı elemanların olduğunu vurgular. Bir kümenin dış I gözlemlenemezliği bazı elemanlar için pozitif sınıflandırma olduğu durumu ifade eder. Ancak X 'e ait olamayan bir elemanı belirlemek imkânsızdır. Yukarıdaki sınıflandırma çapraşıklık ve karşılık gelen belirsizliklerin dört doğal sınıfı olduğu sonucuna götürür. Ayrıca kuramsal öneminin yanında bu sınıflandırma, kaba küme kuramını veri analizinde kullanmada önemli bir pratik öneme sahiptir (Pawlak, 1994c).

3.2.1.4 Bilgi Sistemleri

Bir veri kümesi bir tablo olarak temsil edilir, her bir satır bir durumu, bir olayı, bir hastalığı veya basit olarak bir nesneyi temsil eder. Her sütun, her bir nesne için ölçülebilir bir özelliği (bir değişken, bir gözlem, vb.) temsil eder. Bu tablo bir bilgi sistemi olarak adlandırılır. Daha biçimsel olarak, bir bilgi sistemi, $A = (U, A)$ ikilisi ile temsil edilir. U , evren olarak

adlandırılan nesnelerin boş olmayan sonlu bir kümesi ve A , özelliklerin boş olmayan sonlu bir kümesidir. Burada, $\forall a \in A$ için $a : U \rightarrow V_a$. V_a kümesi, a 'nın değer kümesi olarak adlandırılır. Bilgi sistemlerinin bir diğer çeşidi ise karar sistemleri olarak adlandırılır. Bir karar sistemi, herhangi bir bilgi sisteminin $A = (U, A \cup \{d\})$ biçimidir. Burada, $d \notin A$ karar özellikleridir. Diğer özellikler $a \in A - \{d\}$ koşullu özellikler olarak adlandırılır. Karar özellikleri birçok değer alabilir, fakat genellikle doğru, yanlış gibi ikili değerler alırlar (Komorowski, vd., 1998; Hui, 2002).

3.2.1.5 Ayırtedilemezlik

Bilgi sistemlerinin özel bir hali olan karar sistemleri bir model (olay, durum) hakkındaki tüm bilgiyi içerir. Karar sistemlerinde aynı veya ayırtedilemeyen nesnelere birden fazla temsil edilmiş olabilir veya özellikler gereğinden fazladır. Bu durumda karar sistemini temsil eden tablo gereğinden büyük olacaktır. Aşağıda ayırtedilemezliğe (indiscernibility) ait bağıntı tanımlanacaktır.

Bir ikili bağıntı $R \subseteq X \times X$, yansımali (yani bir nesne kendi kendisi ile ilişkili ise xRx), simetrik (eğer xRy ise o zaman yRx), ve geçişli ise (eğer xRy ve yRz ise o zaman xRz) o zaman bir denklik bağıntısıdır. Bir $x \in X$ elemanının denklik sınıfları tüm $y \in X$ nesnelere içerir, şöyle ki, xRy 'dir. $A = (U, A)$ bir bilgi sistemi olsun o zaman, herhangi bir $B \subseteq A$ ile bir $IND_A(B)$ denklik bağıntısı ilişkisi mevcuttur:

$$IND_A(B) = \{(x, y) \in U^2 \mid \forall a \in B a(x) = a(y)\} \quad (3.27)$$

$IND_A(B)$, B -ayırtedilemezlik bağıntısı olarak adlandırılır. Eğer $(x, y) \in IND_A(B)$ ise o zaman x ve y nesnelere, B 'deki özellikler ile birbirlerinden ayırtedilemezdir. B -ayırtedilemezlik bağıntısının denklik sınıfları $[x]_B$ olarak gösterilir (Komorowski vd., 1998; 1999). $IND_A(B)$ ayırtedilemezlik bağıntısı, ikili bir denklik bağıntısı olarak verilen bir U evrensel kümesini bir $\{X_1, X_2, \dots, X_r\}$ denklik sınıfları ailesine ayırır. U kümesindeki $IND_A(B)$ bağıntısı ile tanımlanan tüm $\{X_1, X_2, \dots, X_r\}$ denklik sınıflarının ailesi U kümesinin bir bölümlendirmesini oluşturur ve B^* olarak ifade edilir. B^* denklik sınıflarının ailesi sınıflama olarak da adlandırılır ve $U / IND_A(B)$ ile gösterilir. Aynı X_i denklik sınıflarına ait nesnelere ayırtedilemezdir, aksi takdirde nesnelere B özellikler alt kümesine göre ayırtedilebilirler. $IND_A(B)$ bağıntısının X_i , $(1, 2, \dots, r)$ denklik sınıflarına bir A bilgi sistemindeki B temel (elementer) kümeler olarak adlandırılır.

$[x]_B$, bir x elemanını içeren bir B temel kümesini gösterir ve aşağıdaki (3.28) eşitliği ile tanımlanır:

$$[x]_B = \{y \in U \mid xIND_A y\} \quad (3.28)$$

Bir $(U, IND_A(B))$ sıralı çifti yaklaşım uzayı olarak adlandırılır. Bir yaklaşım uzayındaki temel (elemanter) kümelerin herhangi bir sonlu birleşimine yaklaşım uzayında tanımlı küme veya birleşik küme denir. Bir $A = (U, A)$ bilgi sisteminin A temel (elemanter) kümelerine A bilgi sisteminin atomları denir (Binay, 2002).

3.2.1.6 Ayırtedilebilirlik Matrisi

Nesnelerin ayırtedilebilirliği ile ilgili çalışma Skowron ve Rauszer (1992) tarafından yapılmıştır. Bu çalışmada, verilen bir bilgi sistemindeki bütün kavramları tanımlamak için yeterli en az özellik alt kümelerinin oluşturulması ile ilgili etkin algoritmalar oluşturmaya yönelik ayırtedilebilirlik matrisi (indiscernibility matrix) ve ayırtedilebilirlik fonksiyonu sunulmuştur.

A , n adet nesnesi olan bir bilgi sistemi olsun. A bilgi sistemi için M_A ayırtedilebilirlik matrisi aşağıda verilen c_{pq} elemanları olan simetrik bir $n \times n$ matristir. Bu matrisin her c_{pq} elemanı x_p ve x_q nesnelerini farklı kılan özellikler kümesinden oluşmaktadır.

$$c_{pq} = \{a \in A \mid a(x_p) \neq a(x_q)\}, \quad (p, q = 1, 2, \dots, n) \quad (3.29)$$

Kavramsal olarak, M_A ayırtedilebilirlik matrisi bir $|U| \times |U|$ matristir. Ayırtedilebilirlik matrisini oluşturabilmemiz için farklı nesne çiftlerini göz önünde bulundurmamız gerekmektedir. Tüm x_p ve x_q nesneleri için $c_{pq} = c_{qp}$ ve $c_{pp} = \emptyset$ olduğundan dolayı M_A ayırtedilebilirlik matrisinin oluşturulmasında elemanların yarısının hesaplanması gerekmeyecektir. Bu da, hesaplama karmaşıklığının azalmasına neden olacaktır.

3.2.1.7 Ayırtedilebilirlik Fonksiyonu

Ayırtedilebilirlik fonksiyonu (discernibility function) bir nesne veya nesneler kümesinin, nesneler evreninin belirli bir alt kümesinden nasıl ayırtedilebileceğini açıklayan bir fonksiyondur. Ayırt edilebilirlik fonksiyonu Boolean toplamlarının bir çarpımıdır. M_A ayırtedilebilirlik matrisi her hangi bir $x \in U$ nesnesi için ayırtedilebilirlik fonksiyonu aşağıda verildiği şekilde oluşturulur. Bir bilgi sistemi için f_A ayırtedilebilirlik fonksiyonu a_1, a_2, \dots, a_m özelliklerine karşılık gelen m adet $a_1^*, a_2^*, \dots, a_m^*$ Boolean

değişkenlerinin bir boolean fonksiyonudur. f_A ayırtedilebilirlik fonksiyonu aşağıdaki gibi ifade edilir:

$$f_A(a_1^*, a_2^*, \dots, a_m^*) = \bigwedge \{ \bigvee c_{pq}^* \mid 1 \leq q \leq p \leq n, c_{pq}^* \neq \emptyset \} \quad (3.30)$$

Burada,

$$c_{pq} = \{ a \in A \mid a(x_p) \neq a(x_q) \} \quad (p, q = 1, 2, \dots, n) \quad (3.31)$$

$$c_{pq}^* = \{ a^* \mid a \in c_{pq} \} \quad (3.32)$$

elde edilir.

Bir M_A ayırtedilebilirlik matrisinden bir $x \in U$ nesnesi ile ilgili olarak bir ayırt edilebilirlik fonksiyonu oluşturmak mümkün olabilir. $f_A(x)$ fonksiyonu $|A|$ Boolean değişkenlerinin, toplamlarının çarpım fonksiyonu olup a^* değişkeni ise a özelliğine karşılık gelmektedir. $f_A(x)$ 'in her bir birleşimi x 'in ayırtedilebildiği bir $y \in U$ nesnesinden çıkar ve birleşimdeki her bir terim bu nesnelere birbirinden ayırt eden bir özelliği temsil eder.

$$f_A(x) = \prod_{y \in U} \{ \sum a^* \mid a \in M_A(x, y) \vee M_A(x, y) \neq \emptyset \} \quad (3.33)$$

$f_A(x)$ 'in temel içerenleri U evreninde nesnelere x nesnesinden ayırt etmek için ihtiyaç duyulan A 'nın en küçük alt kümelerini gösterir.

3.2.2 Küme Yaklaşımları

Kaba kümeler kuramının altında yatan temel düşünce, $IND_A(B)$ ikili bağıntısını kullanmak suretiyle kümelerin yaklaşımlarını oluşturmaktır. Eğer X , A özellikleri kullanılarak kesin bir biçimde tanımlanmıyorsa o halde, alt ve üst yaklaşımlar ile ifade edilirler. $A = (U, A)$ bir bilgi sistemi ve $B \subseteq A$ ve $X \subseteq U$ olsun. X sırasıyla, $\underline{B}X$ ve $\overline{B}X$ ile gösterilen X 'in B -alt ve B -üst yaklaşımlarını oluşturarak sadece B 'de içeren bilgi kullanılarak yaklaştırılabilir. Burada, alt ve üst yaklaşımlar sırasıyla aşağıdaki gibi tanımlanır:

$$\underline{B}X = \{ x \mid [x]_B \subseteq X \} \quad (3.34)$$

$$\overline{B}X = \{ x \mid [x]_B \cap X \neq \emptyset \} \quad (3.35)$$

$\underline{B}X$ 'deki nesnelere, B 'deki bilginin temeli üzerine kesin X 'in elemanları olarak sınıflandırılır. $\overline{B}X$ 'deki nesnelere, B 'deki bilginin temeli üzerine sadece X 'in olası elemanları olarak sınıflandırılabilir.

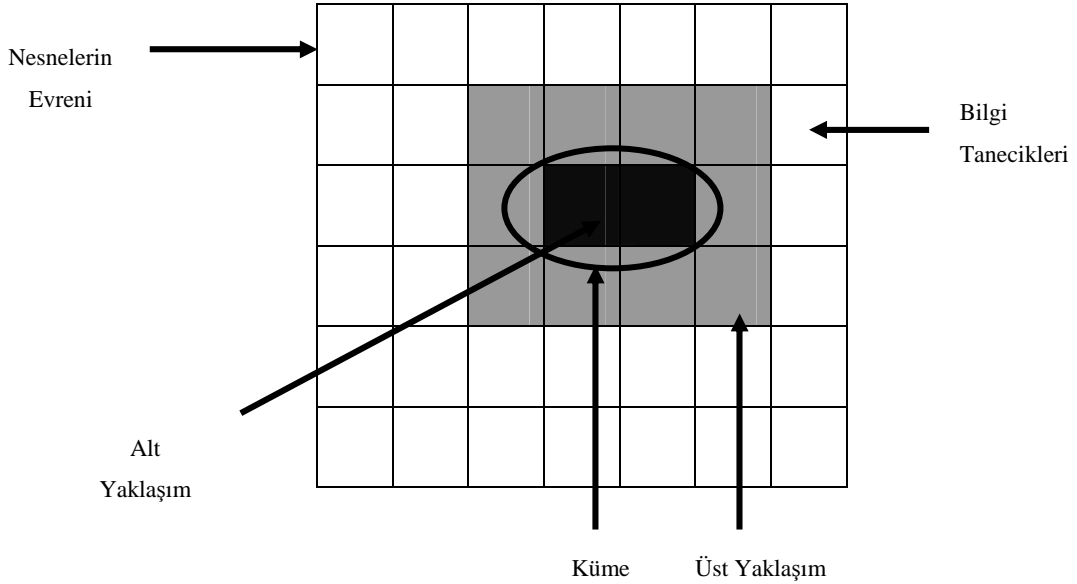
$$BN_B(X) = \overline{BX} - \underline{BX} \quad (3.36)$$

(3.36) eşitliği X 'in B -sınır bölgesi olarak adlandırılır ve böylece, B 'deki bilginin temeli üzerine kesin olarak X 'e ait olarak sınıflanamayan nesnelere oluşur.

$U - \overline{BX}$ kümesi X 'in B -dış bölgesi olarak adlandırılır ve B 'deki bilginin temeli üzerine kesinlikle X 'e ait olmayan nesnelere oluşur.

Eğer, $BN_B(X) = \overline{BX} - \underline{BX} = \emptyset$, yani $\overline{BX} = \underline{BX}$ ise B kümesi kesin küme olarak adlandırılır. $BN_B(X) \neq \overline{BX} - \underline{BX} \neq \emptyset$, yani $\overline{BX} \neq \underline{BX}$ ise B kümesi kaba küme olarak adlandırılır. Bu durumda, B kümesi sadece alt ve üst yaklaşımlar ile nitelendirilebilir.

Aşağıdaki Şekil 3.1' de bir kaba küme örneği verilmiştir.



Şekil 3.1 Bir kaba küme örneği

Alt ve üst yaklaşımlar aşağıdaki özelliklere sahiptir:

$$1. \underline{BX} \subseteq X \subseteq \overline{BX} \quad (3.37)$$

$$2. \underline{B}(\emptyset) = \overline{B}(\emptyset) = \emptyset, \underline{B}(U) = \overline{B}(U) = U \quad (3.38)$$

$$3. \overline{B}(X \cup Y) = \overline{B}(X) \cup \overline{B}(Y) \quad (3.39)$$

$$4. \underline{B}(X \cap Y) = \underline{B}(X) \cap \underline{B}(Y) \quad (3.40)$$

$$5. X \subseteq Y \text{ olsun, bu durumda } \underline{B}(X) \subseteq \underline{B}(Y) \text{ ve } \overline{B}(X) \subseteq \overline{B}(Y) \quad (3.41)$$

$$6. \underline{B}(X \cup Y) \supseteq \underline{B}(X) \cup \underline{B}(Y) \quad (3.42)$$

$$7. \overline{B}(X \cap Y) \subseteq \overline{B}(X) \cap \overline{B}(Y) \quad (3.43)$$

$$8. \underline{B}(-X) = -\overline{B}(X) \quad (3.44)$$

$$9. \overline{B}(-X) = -\underline{B}(X) \quad (3.45)$$

$$10. \underline{B}(\underline{B}(X)) = \overline{B}(\overline{B}(X)) = \underline{B}(X) \quad (3.46)$$

$$11. \overline{B}(\overline{B}(X)) = \underline{B}(\underline{B}(X)) = \overline{B}(X) \quad (3.47)$$

Burada, $-X$, $U - X$ anlamına gelmektedir. Kaba kümelerin dört temel sınıfı (çapraşıklıkla dört kategorisi) aşağıdaki gibi tanımlanır.

- Sadece ve sadece $\underline{B}(X) \neq \emptyset$ ve $\overline{B}(X) \neq U$ ise o halde X kaba olarak B tanımlanabilir.
- Sadece ve sadece $\underline{B}(X) = \emptyset$ ve $\overline{B}(X) \neq U$ ise o halde X içten B tanımlanamaz.
- Sadece ve sadece $\underline{B}(X) \neq \emptyset$ ve $\overline{B}(X) = U$ ise o halde X dıştan B tanımlanamaz.
- Sadece ve sadece $\underline{B}(X) = \emptyset$ ve $\overline{B}(X) = U$ ise o halde X tamamen B tanımlanamaz.

Bu sınıflandırmaların anlamları aşağıdaki gibidir:

Eğer, X kaba olarak B tanımlanabilir ise o halde, B 'yi kullanmak suretiyle U kümesinin bazı elemanlarının X 'e ve bazı elemanlarının ise $-X$ 'e ait olduğuna karar verilebileceği anlamına gelir.

Eğer, X içten B tanımlanamaz ise o halde, B 'yi kullanmak suretiyle U kümesinin bazı elemanlarının $-X$ 'e ait olduğuna karar verilebileceği fakat U kümesinin herhangi bir elemanının X 'e ait olup olmadığına karar verilemeyeceği anlamına gelir.

Eğer, X dıştan B tanımlanamaz ise o halde, B 'yi kullanmak suretiyle U kümesinin bazı elemanlarının X 'e ait olduğuna karar verilebileceği fakat U kümesinin herhangi bir elemanının $-X$ 'e ait olup olmadığına karar verilemeyeceği anlamına gelir.

Eğer, X tamamen B tanımlanamaz ise o halde, B 'yi kullanmak suretiyle U kümesinin hiçbir elemanının X veya $-X$ 'e ait olup olmadığına karar verilemeyeceği anlamına gelir (Komorowski, vd., 1998).

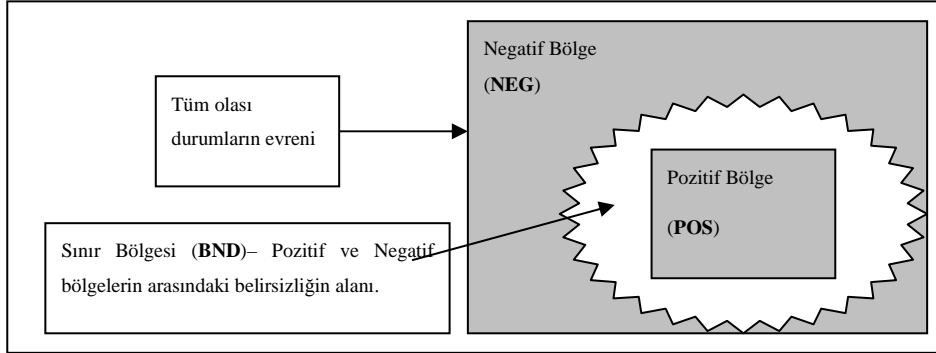
$X \subseteq U$ herhangi alt kümesi ile ilgili olarak evren alt ve üst yaklaşımlar kullanılarak üç ayrık bölgeye bölünebilir. Bu üç bölge pozitif, negatif ve sınır bölgesi aşağıdaki gibi ifade edilir.

$$BND(X) = \overline{B}(X) - \underline{B}(X) \quad (3.48)$$

$$POS(X) = \underline{B}(X) \quad (3.49)$$

$$NEG(X) = U - \overline{B}(X) \quad (3.50)$$

$NEG(X)$ negatif bölgesinin bir elemanı X 'e ait değildir. $POS(X)$ pozitif bölgesinin bir elemanı X 'e aittir ve sınır bölgesi $BND(X)$ 'in sadece bir elemanı X 'e aittir (Allam vd., 2005). Aşağıdaki Şekil 3.2'de bu bölgeler gösterilmiştir.



Şekil 3.2 Bir kaba kümenin pozitif, negatif ve sınır bölgeleri

3.2.2.1 Kavramların Yakınlık Ölçüleri

Bu bölümde bazı yakınlık ölçüleri yaklaşımları anlatılacaktır. Bu ölçüler, yaklaşımın doğruluğu, pozitif bölgenin ölçüsü, kaba üyelik fonksiyonları ve bir dereceye göre bağımlılıklardır. Bu düşünceler makul yöntemlerin belirlenmesinde uygulanabilir olmasının yanı sıra kuralların gücünü ve kavramların yakınlığını değerlendirmede kullanılabilirler.

3.2.2.1.1 Yaklaşımın Doğruluğu

Bir kaba küme sezgisel olarak aşağıdaki katsayı ile karakterize edilebilir:

$$\alpha_B(X) = \frac{|\underline{B}(X)|}{|\overline{B}(X)|} = \frac{card(\underline{B}(X))}{card(\overline{B}(X))} \quad (3.51)$$

Burada $\alpha_B(X)$ katsayısı, yaklaşımın doğruluğu olarak adlandırılır ve $|\underline{B}(X)|$ ile $|\overline{B}(X)|$ kümesinin eleman sayısı, $|\underline{B}(X)|$ ile $|\overline{B}(X)|$ kümesinin eleman sayısı gösterilir. Açıkça görülebilir ki, $0 \leq \alpha_B(X) \leq 1$ 'dir. ($\alpha_B(X) \in [0,1]$)

Eğer, $\alpha_B(X) = 1$ ise X , B ile ilgili kesin olarak adlandırılır, aksi takdirde eğer, $\alpha_B(X) < 1$ ise X , B ile ilgili kaba (X , B ile ilgili olarak çapraşıktır) olarak adlandırılır.

3.2.2.1.2 Pozitif Bölge ve Ölçüsü

Eğer, $X_A^1, X_A^2, \dots, X_A^{r(d)}$ A bilgi sisteminin karar sınıfları ise o halde, $\underline{B}X_1 \cup \underline{B}X_2 \cup \dots \cup \underline{B}X_{r(d)}$ kümesi A bilgi sisteminin B-pozitif bölgesi olarak adlandırılır ve $POS_B(d)$ ile gösterilir. $|POS_B(d)|/|U|$ sayısı ile karar ile tanımlanan bölüntü ile B'den özellikler ile tanımlanan yaklaşımın yakınlığını ölçer.

3.2.2.1.3 Kaba Üyelik Fonksiyonu

Klasik küme kuramında, bir eleman bir kümeye ya aittir ya da değildir. Üyelik fonksiyonu, kümenin karakteristik fonksiyonudur. Örneğin, sırasıyla 1 ve 0 değerlerini alan fonksiyon gibi. Bu durumda kaba kümelerde üyelik fikri farklıdır. Kaba üyelik fonksiyonu, X kümesi x'in ait olduğu denklik sınıfı arasındaki göreceli örtüşmenin derecesini ölçer ve aşağıdaki gibi tanımlanır.

$$\mu_x^B(x): U \rightarrow [0,1] \text{ ve } \mu_x^B(x) = \frac{|[x]_B \cap X|}{|[x]_B|} \quad (3.52)$$

Üst ve alt küme yaklaşımları için formüller, değişken duyarlılık kaba üyelik fonksiyonları yoluyla $\pi \in \left[\frac{1}{2}, 1\right]$ duyarlılık seviyesi ile aşağıdaki gibi genelleştirilebilir:

$$\underline{B}_\pi X = \{x \mid \mu_x^B(x) \geq \pi\} \quad (3.53)$$

$$\overline{B}_\pi X = \{x \mid \mu_x^B(x) > 1 - \pi\} \quad (3.54)$$

Burada ki, $\underline{B}_\pi X$ ve $\overline{B}_\pi X$ alt ve üst yaklaşımları “*değişken duyarlılık*” kaba küme olarak adlandırılır. Orijinal alt ve üst yaklaşımlar $\pi = 1.0$ olan özel bir durum olarak elde edilir. Kaba kümeler π parametresine dayanarak değiştirilen kesinlik ile hastalar, olaylar, sonuçlar, vb. kümesini yaklaşık olarak tanımlar. Verilerin gürültüden etkilendiği varsayılarak “en iyi” kavram tahminini bulmak için π eşiği ayarlanabilir.

3.2.2.1.4 Özelliklerin Bağımlılığı

Veri analizindeki diğer bir önemli konu, özellikler arasındaki bağımlılıkların keşfedilmesidir. Sezgisel olarak, eğer bir D özellikler kümesindeki özellikler tek olarak bir C özellikler kümesinden, özelliklerin değerleri tarafından belirleniyorsa, D tamamen C'ye bağımlıdır denir ve $C \Rightarrow D$ olarak gösterilir. Diğer bir ifadeyle, eğer D ve C'nin değerleri arasında bir fonksiyonel bağımlılık mevcut ise o zaman, D tamamen C'ye bağımlıdır denir. Bağımlılık biçimsel olarak aşağıdaki yol ile de tanımlanabilir.

A 'nin alt kümeleri C ve D olsun. D 'nin C 'ye k ($0 \leq k \leq 1$) derecesinde bağlı olduğu söylenebilir ve $C \Rightarrow_k D$ şeklinde gösterilir. Eğer,

$$k = \gamma(C, D) = \frac{|POS_C(D)|}{|U|} \quad (3.55)$$

ise (3.55) denkleminde açıkça görülebilir ki,

$$\gamma(C, D) = \sum_{X \in U/D} \frac{|C(X)|}{|U|} \quad (3.56)$$

olacaktır.

Burada,

$$POS_C(D) = \bigcup_{X \in U/D} C(X) \quad (3.57)$$

denklemini elde edilir.

Eğer, $k = 1$ ise D 'nin C 'ye tamamen bağlı olduğu söylenir ve eğer, $k < 1$ ise D 'nin C 'ye kısmen (k derecesinde) bağlı olduğu söylenir. $\gamma(C, D) = U/D$ bölüntüsünün yakınlığını ve yaklaşımının C 'deki durumunu belirler. k katsayısı C özelliklerini kullanarak U/D bölüntüsünün bloklarına düzgün bir şekilde sınıflandırılabilen evrenin tüm elemanlarının oranını ifade eder. Bu da bağımlılık derecesi olarak adlandırılacaktır.

Yukarıdaki tüm yakınlık ölçüleri, mevcut geçmiş bilgi temeline dayalı olarak yapılandırılmıştır. İki önemli problem, ilgili yakınlık ölçülerinin çıkarılması ve üretilmesi yöntemleridir (Komorowski, vd., 1999).

3.2.3 İndirgenmiş Özellik Kümeleri

Kaba küme uygulamalarında önemli bir nokta A özelliklerinin tam kümesi ile aynı nesne sınıflamasının yapılabilmesine imkân sağlayabilmesine göre bilgi sistemindeki özelliklerden bazılarının gereğinden fazla olup olmadığıdır. Eğer, özellik sayısı fazla ise hesaplama karmaşıklığı artacaktır, bu durum kaba kümelere bir darboğaz oluşturur. Bir $B \subseteq A$ özellik alt kümesi $IND_A(A)$ ayırtedilemezlik bağıntısını ve küme yaklaşımlarını biçimlendirme kavramını muhafaza ediyorsa $A - B$ özelliklerinin atılabilir olduğu sonucuna varılır. Bunun nedeni, bu özelliklerin fazlalık olmaları ve atılmalarının sınıflamayı daha kötü yönde etkilemeyecek olmasıdır. Tipik olarak bir bilgi sistemi birçok B özellik alt kümesine sahip olabilir. Tüm en küçük atılabilir özellik içermeyen alt kümelere, indirgenmiş özellik kümeleri

denir. Şöyle ki, verilen bir $A = (U, A)$ bilgi sisteminde bir indirgenmiş özellik kümesi $IND_A(B) = IND_A(A)$ olacak şekilde bir $B \subseteq A$ en küçük özellikler kümesidir. Tüm indirgenmiş özellik kümelerinin arasında en küçük eleman sayısına sahip olanı bulmak NP-Zor problemdir. Tüm indirgenmiş özellik kümelerinin kesişimi çekirdek olarak adlandırılır. Çekirdek boş küme (\emptyset) olabilir. Bir $A = (U, A)$ bilgi sisteminin tüm indirgenmiş özellik kümelerinin kümesi $RED(A)$ ile gösterilir. Belirli bir nesneye göre indirgenmiş özellik kümeleri, belirli nesneyi diğer bütün ayırtedilebilir nesnelere ayırt etmek için ihtiyaç duyulan minimum miktarda bilgiyi gösterirler. Bir $A = (U, A)$ bilgi sisteminin bir x nesnesine göre olan tüm indirgenmiş özellikler kümesi $RED(A, x)$ ile gösterilir (Binay, 2002).

3.2.4 Karar Sistemleri ve Karar Kuralları

Bilgi sistemlerinin bir diğer çeşidi karar sistemleri olarak adlandırılır. Bir karar sistemi, herhangi bir bilgi sisteminin $A = (U, A \cup \{d\})$ biçimindedir. $A = (U, A \cup \{d\})$ bir karar sistemi ve $V = \bigcup \{V_a \mid a \in A\} \cup V_d$ olsun. Burada, $d \notin A$ karar özellikleridir. Diğer özellikler $a \in A - \{d\}$ koşullu özellikler olarak adlandırılır. $B \subseteq A \cup \{d\}$ üzerindeki ve V 'deki çok küçük formüller $a = v$ şeklindeki ifadeler olup bunlar B ve V 'deki tanımlayıcılar olarak adlandırılır. Burada, $a \in B$ ve $v \in V_a$ 'dır. B ve V 'deki formüllerin $F(B, V)$ kümesi B ve V 'deki bütün küçük (atomik) formülleri kapsayan en küçük kümedir ve önerilen bağlaçlara (\wedge (Ve), \vee (Veya), \neg (Değil)) göre kapalıdır. Formüllerin anlamı birbirini takip eden terimler ile tanımlanır. $\zeta \in F(B, V)$ olsun. U 'daki \emptyset özelliği olan bütün nesnelere kümesi $|\zeta|_A$, A karar tablosundaki ζ 'nin anlamını gösterir. Bu kümeler aşağıdaki gibi tanımlanırlar:

i) Eğer ζ , $a = v$ biçiminde ise o halde $|\zeta|_A = \{x \in U \mid a(x) = v\}$ dir.

ii) $|\zeta \wedge \zeta'|_A = |\zeta|_A \cap |\zeta'|_A$, $|\zeta \vee \zeta'|_A = |\zeta|_A \cup |\zeta'|_A$, $|\neg \zeta|_A = U - |\zeta|_A$.

$F(B, V)$ kümesi A 'nin koşul formülleri olarak adlandırılır ve $C(B, V)$ şeklinde gösterilir. (Komorowski, vd., 1998)

$v_i \in V_{a_i}$ ($i = 1, 2, \dots, l$) için ve $P = \{a_1, a_2, \dots, a_l\} \subseteq A$ ise, $v_i \in V_{a_i} (a_1 = v_1) \wedge (a_2 = v_2) \wedge \dots \wedge (a_l = v_l)$ biçimindeki herhangi bir formüle A 'nin bir P temel formülü denir. Eğer, ζ A 'nin bir P temel formülü ise ζ/Q ile $a \in P/Q$ olacak şekilde ζ 'den bütün temel alt formülleri ($a = v_a$) çıkarılarak elde edilen Q temel formülü

ifade edilir. A için bir karar kuralı $\zeta \Rightarrow d = v$ biçimindeki herhangi bir ifadedir.

Burada, $\zeta \in C(B, V)$, $v \in V_d$ ve $|\zeta|_A \neq \emptyset$ 'dir. ζ ve $d = v$ formüllerine sırasıyla, $\zeta \Rightarrow d = v$ karar kuralının önceli ve ardılı olarak adlandırılır. Eğer, r A 'daki bir karar kuralı ise $Pr ed(r)$ ile r 'nin önceli $Succ(r)$ ile r 'nin ardılı olarak ifade edilir. Bir $u \in U$ nesnesi ancak ve ancak $u \in |\zeta|_A$ ise, bir $\zeta \Rightarrow d = v_d^k$, ($1 \leq k \leq r(d)$) karar kuralı ile eşleştirilir.

Eğer u , $\zeta \Rightarrow d = v_d^k$ tarafından eşleştirilirse kuralın u 'yu X_k karar sınıfına sınıfladığı söylenir (Binay, 2002). Karar özellikleri birçok farklı değerler alabilir, fakat genelde ikili değerler alırlar, örneğin, doğru ve yanlış, evet ve hayır değerleri gibi (Hui, 2002).

3.3 Kaba Kümeler Kuramı ve Bayes Ağları Arasındaki İlişki

Bu bölümde Bayes kuramı ile Kaba kümeler arasında bir ilişkinin mevcut olduğu gösterilecektir. Herhangi bir karar algoritması, Bayes kuramını klasik Bayes metodolojisi ile esas olarak birleştirilen önceki veya sonraki olasılıkları kullanmadan sağlar. Aynı zamanda, her karar algoritması ile bir akış çizgesinin birleştirilmesi mümkündür. Bayes kuramı çizgedeki bilgi akışı ile yakından ilişkilidir. Bunun yanında, her akış çizgesi, Eucklidean uzayı ile ifade edilebilir. Burada uzayın noktaları karar algoritması tarafından belirlenen kararları temsil ederken noktalar arasındaki mesafede karar algoritmasındaki kararlar arasındaki mesafeyi gösterir (Pawlak, 2003). Koşul ve karar özelliklerinin birbirinden ayırt edildiği bilgi tablosuna karar tablosu denir. Karar tablosundan “Eğer.....İse.....” şeklinde karar kuralları biçiminde bazı örüntüler çıkarılabilir (Greco vd., 2004).

Karar algoritması, “Eğer.....İse.....” karar kurallarının sonlu bir kümesidir ve her karar kuralı üç katsayı ile ifade edilir. Bunlar kuralın gücü, kesinliği ve örtme faktörleridir. Katsayılar veriden hesaplanabilir veya öznel bir değerlendirme ile belirlenebilir. Bu katsayılar Bayes formülünü sağlar. Bayes çıkarım metodolojisi önceki olasılıkları güncellemeden oluşur ve bunları da sonraki olasılıklar yoluyla gerçekleştirir. Bu da veri mevcut olduğunda güncellenmiş bilgiyi ifade eder. Güç, kesinlik ve örtme faktörleri, sınır hattındaki olasılıklar veya gerçeklik derecesi olarak yorumlanabilir. Ayrıca bunlar karar algoritmaları ile birleştirilmiş olarak akış çizgelerindeki bir deterministik akış dağılımı olarak yorumlanabilir.

3.3.1 Akış Çizgeleri

Bir akış çizgesi (flow garph) yönlü, döngüsüz, sonlu çizge $G = \{ N, \mathcal{B}, \varphi \}$ 'dir. Burada N düğümlerin kümesi $\mathcal{B} \subseteq N \times N$ yönlü dalların kümesi, $\varphi: \mathcal{B} \rightarrow R^+$ bir akış fonksiyonu ve R^+ pozitif gerçel sayılar kümesidir. $(x, y) \in \mathcal{B}$ ise o halde x , y 'nin girişi ve y 'de x 'in çıkışıdır.

Eğer, $x \in N$ ise bu durumda $I(x)$, x 'in tüm girişlerinin ve $O(x)$ 'de x 'in tüm çıkışlarının kümesidir G çizgesinin giriş ve çıkışı aşağıdaki gibi tanımlanır.

$$I(G) = \{x \in N : I(x) = \emptyset\} \quad (3.58)$$

$$O(G) = \{x \in N : O(x) = \emptyset\} \quad (3.59)$$

G çizgesinin giriş ve çıkışları G 'nin dış düğümleridir, diğer düğümler ise G 'nin iç düğümleri olarak adlandırılır. Eğer, $(x,y) \in \mathcal{B}$ ise o halde $\varphi(x,y)$ x 'den y 'ye bir akış yoludur. Varsayalım ki, her $(x,y) \in \mathcal{B}$ için $\varphi(x,y) \neq 0$ olsun. G akış çizgesinin her x düğümü ile giriş ve çıkış akışını birleştirelim bu durumda aşağıdaki denklemler elde edilir.

$$\varphi_+(x) = \sum_{y \in I(x)} \varphi(y,x) \quad (3.60)$$

$$\varphi_-(x) = \sum_{y \in O(x)} \varphi(x,y) \quad (3.61)$$

Benzer şekilde, tüm G akış çizgesi için aşağıdaki gibi giriş ve çıkış akışı tanımlanır.

$$\varphi_+(G) = \sum_{x \in I(G)} \varphi_-(x) \quad (3.62)$$

$$\varphi_-(G) = \sum_{x \in O(G)} \varphi_+(x) \quad (3.63)$$

Herhangi x içsel düğümü için, $\varphi_+(x) = \varphi_-(x) = \varphi(x)$ 'dir ve burada $\varphi(x)$, x düğümünün akış yoludur. Açık bir şekilde, $\varphi_+(G) = \varphi_-(G) = \varphi(G)$ olur ve burada $\varphi(G)$, G çizgesinin akış yoludur.

Yukarıdaki formüller, akış korunum denklemleri olarak ele alınabilir.

Bir normalize edilmiş akış çizgesi bir yönlü, döngüsüz, sonlu çizge, $G = \{N, \mathcal{B}, \sigma\}$ dir. Burada, N düğümler kümesi, $\mathcal{B} \subseteq N \times N$ yönlü dalların bir kümesi ve $\sigma : \mathcal{B} \rightarrow \langle 0,1 \rangle$, (x,y) 'nin normalize edilmiş akışıdır ve

$$\sigma(x,y) = \frac{\varphi(x,y)}{\varphi(G)} \quad (3.64)$$

(3.64) denklemi (x,y) 'nin gücüdür ve 0 ile 1 arasında değerler alacaktır ($0 \leq \sigma(x,y) \leq 1$) Dalın gücü, dal boyunca toplam akışın basitçe yüzdesini ifade eder. G akış çizgesinin her x düğümü ile onun normalize edilmiş giriş ve çıkış akışı birleştirilebilir. Bu da aşağıdaki gibi tanımlanır.

$$\sigma_+(x) = \frac{\varphi_+(x)}{\varphi(G)} = \sum_{y \in I(x)} \sigma(y, x) \quad (3.65)$$

$$\sigma_-(x) = \frac{\varphi_-(x)}{\varphi(G)} = \sum_{y \in O(x)} \sigma(x, y) \quad (3.66)$$

Açıkça görülebilir ki, her x içsel düğümü için $\sigma_+(x) = \sigma_-(x) = \sigma(x)$ 'dir. Burada, $\sigma(x)$ normalize edilmiş akış yoludur.

Ayrıca,

$$\sigma_+(G) = \frac{\varphi_+(G)}{\varphi(G)} = \sum_{x \in I(G)} \sigma_-(x) \quad (3.67)$$

$$\sigma_-(G) = \frac{\varphi_-(G)}{\varphi(G)} = \sum_{x \in O(G)} \sigma_+(x) \quad (3.68)$$

olsun. (3.67) ve (3.68) denklemlerinden açıkça görülebilir ki, $\sigma_+(G) = \sigma_-(G) = \sigma(G) = 1$ olacaktır.

3.3.2 Kesinlik ve Örtme Faktörleri

Bir G akış çizgesinin her (x, y) dalı ile kesinlik ve örtme faktörleri birleştirilebilir. (x, y) 'nin kesinlik ve örtme faktörleri aşağıdaki (3.69) ve (3.70) denklemleri ile tanımlanır.

$$\text{cer}(x, y) = \frac{\sigma(x, y)}{\sigma(x)} \quad (3.69)$$

$$\text{cov}(x, y) = \frac{\sigma(x, y)}{\sigma(y)} \quad (3.70)$$

Burada sırasıyla, $\sigma(x) \neq 0$ ve $\sigma(y) \neq 0$ olacaktır.

Yukarıda verilen tanımlardan aşağıdaki sonuçlar elde edilir.

$$\sum_{y \in O(x)} \text{cer}(x, y) = 1 \quad (3.71)$$

$$\sum_{x \in I(y)} \text{cov}(x, y) = 1 \quad (3.72)$$

$$\sigma(x) = \sum_{y \in O(x)} \text{cer}(x, y) \sigma(x) = \sum_{y \in O(x)} \sigma(x, y) \quad (3.73)$$

$$\sigma(y) = \sum_{x \in I(y)} \text{cov}(x, y) \sigma(y) = \sum_{x \in I(y)} \sigma(x, y) \quad (3.74)$$

$$\text{cer}(x, y) = \frac{\text{cov}(x, y)\sigma(y)}{\sigma(x)} \quad (3.75)$$

$$\text{cov}(x, y) = \frac{\text{cer}(x, y)\sigma(x)}{\sigma(y)} \quad (3.76)$$

Yukarıdaki özellikler olasılıksal bir niteliğe sahiptir. Örneğin, (3.73) ve (3.74) denklemleri toplam olasılıksal teorem yapısına sahiptir ve (3.75) ve (3.76) formülleri ise Bayes kurallarıdır. Burada ki yaklaşımda bu özellikler deterministik bir yol ile tanımlanır ve ağdaki dallar boyunca akış dağılımını tanımlar.

x 'den y 'ye bir yönlü yol G çizgesinde $x \neq y$ olmak üzere her i için $(1 \leq i \leq n-1)$ $x_1 = x$, $x_n = y$ ve $(x_i, x_{i+1}) \in \mathcal{B}$ gibi x_1, x_2, \dots, x_n ' e düğümler dizisidir. x 'den y 'ye bir yol $[x \dots y]$ ile gösterilir. $[x_1, x_2, \dots, x_n]$ yolunun gücü, kesinliği ve örtmesi sırasıyla aşağıdaki gibi tanımlanır.

$$\text{cer}[x_1, x_2, \dots, x_n] = \prod_{i=1}^{n-1} \text{cer}(x_i, x_{i+1}) \quad (3.77)$$

$$\text{cov}[x_1, x_2, \dots, x_n] = \prod_{i=1}^{n-1} \text{cov}(x_i, x_{i+1}) \quad (3.78)$$

$$\sigma[x \dots y] = \sigma(x)\text{cer}[x \dots y] = \sigma(y)\text{cov}[x \dots y] \quad (3.79)$$

G çizgesinde, x 'den y 'ye tüm yollar kümesi $\langle x, y \rangle$ ile gösterilir. Bu da G 'de x 'den y 'ye bir bağlantı olarak adlandırılır. Diğer bir ifadeyle, $\langle x, y \rangle$ bağlantısı x ve y düğümleri ile belirlenen G 'nin bir alt çizgesidir. Her $\langle x, y \rangle$ bağlantısı için onun güç, kesinlik ve örtmesi aşağıdaki gibi gösterilir.

$$\text{cer}\langle x, y \rangle = \sum_{[x \dots y] \in \langle x, y \rangle} \text{cer}[x \dots y] \quad (3.80)$$

$\langle x, y \rangle$ bağlantısının örtmesi,

$$\text{cov}\langle x, y \rangle = \sum_{[x \dots y] \in \langle x, y \rangle} \text{cov}[x \dots y] \quad (3.81)$$

$\langle x, y \rangle$ bağlantısının gücü,

$$\sigma\langle x, y \rangle = \sum_{[x \dots y] \in \langle x, y \rangle} \sigma[x \dots y] = \sigma(x)\text{cer}\langle x, y \rangle = \sigma(y)\text{cov}\langle x, y \rangle \quad (3.82)$$

$[x \dots y]$ bir yol olsun. Burada, x ve y sırasıyla G çizgesinin giriş ve çıkışıdır. Böyle bir yol tam olarak adlandırılır. x 'den y 'ye tüm yolların kümesi G çizgesinde x 'den y 'ye tam bağlantı olarak adlandırılacaktır. x ve y sırasıyla, G 'nin giriş ve çıkışı olsun. Eğer G 'deki her $\langle x, y \rangle$ tam bağlantısı için, $\sigma(x, y) = \sigma\langle x, y \rangle$ ile yer değiştirirsek $cer(x, y) = cer\langle x, y \rangle$ ve $cov(x, y) = cov\langle x, y \rangle$ olur. O halde, $\sigma(G) = \sigma(G')$ gibi yeni bir G' akış çizgesi elde ederiz. Yeni akış çizgesi, birleştirilmiş akış çizgesi olarak adlandırılır. Verilen bir akış çizgesi için birleştirilmiş akış çizgesi onun giriş ve çıkışları arasındaki ilişkiyi temsil eder.

3.3.3 Karar Algoritmaları

Bir karar kuralı (decision rule), $\Phi \rightarrow \Psi$ biçiminde ifade edilir ve “Eğer Φ ise o halde Ψ ” şeklinde okunur. Burada Φ ve Ψ kuralın koşul ve kararı olarak adlandırılan mantıksal formüllerdir. Bu mantıksal formüller çizgenin düğüm kümesini oluşturur. Mantıksal formüller, önermesel fonksiyonlar olarak ifade edilir ve her kenar (Φ, Ψ) karar kuralı olarak görülebilir. $|\Phi|$, U evreninden tüm nesnelere kümesini gösterir ve bunlar Φ özelliğine sahiptir. Eğer, $\Phi \rightarrow \Psi$ bir karar kuralı ise o zaman $supp(\Phi, \Psi) = card(\Phi \wedge \Psi)$ karar kuralının desteği olarak adlandırılır ve karar kuralı, bu kuralın gücü ($\sigma(\Phi, \Psi)$), kesinliği ($cer(\Phi, \Psi)$) ve örtme ($cov(\Phi, \Psi)$) faktörleri ile karakterize edilir. Her $[\Phi_1, \Phi_2, \dots, \Phi_n]$ yolu, $\Phi_1 \rightarrow \Phi_2, \Phi_2 \rightarrow \Phi_3, \dots, \Phi_{n-1} \rightarrow \Phi_n$ karar kuralları dizisini oluşturur ve karar kurallarının bu dizisi tek bir karar kuralı $\Phi_1 \Phi_2 \dots \Phi_{n-1} \rightarrow \Phi_n$ olarak yorumlanır ve kısaca $\Phi^* \rightarrow \Phi_n$ olarak yazılır. Burada, $\Phi^* = \Phi_1 \wedge \Phi_2 \wedge \dots \wedge \Phi_{n-1}$ dir ve aşağıdaki gibi karakterize edilir.

$$cer(\Phi^*, \Phi_n) = cer[\Phi_1, \Phi_2, \dots, \Phi_n] \quad (3.83)$$

$$cov(\Phi^*, \Phi_n) = cov[\Phi_1, \Phi_2, \dots, \Phi_n] \quad (3.84)$$

$$\sigma(\Phi^*, \Phi_n) = \sigma(\Phi_1) cer[\Phi_1, \Phi_2, \dots, \Phi_n] = \sigma(\Phi_n) cov[\Phi_1, \Phi_2, \dots, \Phi_n] \quad (3.85)$$

Burada, $\sigma(\Phi)$ formülün doğruluk değeri ve $\sigma(\Phi, \Psi)$, $\Phi \rightarrow \Psi$ karar kuralının gücüdür. Benzer şekilde, her $\langle \Phi, \Psi \rangle$ bağlantısı tek karar kuralı $\Phi \rightarrow \Psi$ olarak yorumlanabilir ve aşağıdaki (3.86), (3.87) ve (3.88) bağıntıları ile gösterilir.

$$cer(\Phi, \Psi) = cer\langle \Phi, \Psi \rangle \quad (3.86)$$

$$cov(\Phi, \Psi) = cov\langle \Phi, \Psi \rangle \quad (3.87)$$

$$\sigma(\Phi, \Psi) = \sigma(\Phi) \text{cer} \langle \Phi, \Psi \rangle = \sigma(\Psi) \text{cov} \langle \Phi, \Psi \rangle \quad (3.88)$$

$[\Phi_1, \Phi_2, \dots, \Phi_n]$ bir yol olsun. Şöyle ki, Φ_1 bir giriş ve Φ_n G akış çizgesinin bir çıkışıdır. Böyle bir yol, uygun $\langle \Phi_1, \Phi_n \rangle$ bağlantısı tam olarak adlandırılır. Tüm karar kuralları $\Phi_{i_1} \Phi_{i_2} \dots \Phi_{i_{n-1}} \rightarrow \Phi_{i_n}$ tüm tam yollar $\Phi_{i_1} \Phi_{i_2} \dots \Phi_{i_n}$ ile ilişkilidir ve akış çizgesi tarafından çıkarılan karar algoritması olarak adlandırılır.

Bir karar kuralının gücü, kesinlik ve örtme faktörleri aşağıdaki gibi tanımlanır.

Karar kuralının gücü,

$$\sigma(\Phi, \Psi) = \frac{\text{supp}(\Phi, \Psi)}{\text{card}(U)} \quad (3.89)$$

Karar kuralının kesinlik faktörü,

$$\text{cer}(\Phi, \Psi) = \frac{\text{supp}(\Phi, \Psi)}{\text{card}(|\Phi|)} \quad (3.90)$$

Karar kuralının örtme faktörü,

$$\text{cov}(\Phi, \Psi) = \frac{\text{supp}(\Phi, \Psi)}{\text{card}(|\Psi|)} \quad (3.91)$$

Karar tablolarından “Eğer.....İse.....” şeklinde karar kuralları örüntülerinin çıkarılabileceğine yukarıda değinilmişti. Örneğin, hasta örneklerinden tıbbi bilgi toplayan bir veri tablosunda belirtiler kümesi $S = \{s_1, s_2, \dots, s_n\}$ 'yi koşul özellikleri ve hastalıklar kümesi $D = \{d_1, d_2, \dots, d_m\}$ 'yi de koşul özellikleri olarak ele alalım. Karar tablosunda karar kurallarını “Eğer $s_{i_1}, s_{i_2}, \dots, s_{i_h}$ belirtileri görülürse o halde d_j hastalığı vardır” biçiminde karar kurallarını çıkarabiliriz. Burada $s_{i_1}, s_{i_2}, \dots, s_{i_h} \in S$ ve $d_j \in D$ 'dir. Bir karar tablosundan çıkarılan her karar kuralı ile geleneksel olarak üç katsayı (güç, kesinlik ve örtme) ilişkilidir. Örneğin, karar kuralı “Eğer $s_{i_1}, s_{i_2}, \dots, s_{i_h}$ belirtileri görülürse o halde d_j hastalığı vardır” şeklinde ise bu aşağıdaki şekilde karakterize edilebilir: (% sayısı, varsayıma dayalı veri tablosundan hesaplanır.)

- $s_{i_1}, s_{i_2}, \dots, s_{i_h}$ belirtilerine sahip olan hastalar ve d_j hastalığı örnekteki tüm hastaların %15'ini oluşturur. Bu durumda kuralın gücü %15'dir.
- $s_{i_1}, s_{i_2}, \dots, s_{i_h}$ belirtilerine sahip olan hastaların %91'i ayrıca d_j hastalığına sahiptir. bu durumda kuralın kesinlik faktörü %91'dir.

- d_j hastalığına sahip hastaların %52'si ayrıca $s_{i1}, s_{i2}, \dots, s_{ih}$ belirtilerine sahiptir.

Bu durumda kuralın örtme faktörü %52'dir (Greco vd.,2004).

Güç, kesinlik ve örtme katsayıları, uzun zamandır veri tabanlarında ve makine öğreniminde kullanılmaktadır. Eğer $\text{cer}(\Phi, \Psi) = 1$ ise, $\Phi \rightarrow \Psi$ karar kuralı kesin olarak adlandırılır. Aksi takdirde karar kuralı, kesin olmayan şekilde adlandırılır. Karar kurallarının bir kümesi,

$$\text{Dec}(\Phi, \Psi) = \{\Phi_i \rightarrow \Psi_i\}_{i=1}^n \quad (3.92)$$

şeklindedir. Eğer (3.92) de verilen ifadenin tüm karar kuralları aşağıdaki özellikleri sağlarsa $n \geq 2$ için bir karar algoritması olarak adlandırılır.

- $\text{supp}(\Phi_i, \Psi_i) \neq \emptyset$ (her $1 \leq i \leq n$ için)
- Karşılıklı olarak bağımsız, örneğin, her $\Phi_i \rightarrow \Psi_i$ ve $\Phi_j \rightarrow \Psi_j$ için, $\Phi_i = \Psi_j$ veya $\Phi_i \wedge \Psi_j = \emptyset$ ve $\Phi_j = \Psi_i$ veya $\Psi_i \wedge \Phi_j = \emptyset$.
- $|\bigvee_{i=1}^n \Phi_i| = |\bigvee_{i=1}^n \Psi_i| = U$.

Eğer, $\text{Dec}(\Phi, \Psi) = \{\Phi_i \rightarrow \Psi_i\}_{i=1}^n$ bir karar algoritması ise,

$$\text{Dec}(\Psi, \Phi) = \{\Psi_i \rightarrow \Phi_i\}_{i=1}^n \quad (3.93)$$

ifadesi $\text{Dec}(\Phi, \Psi)$ 'nin ters karar algoritması olarak adlandırılır.

3.3.3.1 Karar Algoritmalarının Özellikleri

$\text{Dec}(\Phi, \Psi)$ bir karar algoritması ve $\Phi \rightarrow \Psi$, karar algoritmasındaki bir karar kuralı olsun. $D(\Phi)$ ve $C(\Psi)$, Φ 'nin tüm kararların kümesini ve $\text{Dec}(\Phi, \Psi)$ 'deki Ψ 'nin tüm koşullarının kümesini gösterir (Pawlak, 2002b).

Her karar algoritmasının aşağıdaki olasılıksal özelliklere sahip olduğu gösterilebilir.

$$\sum_{\Phi' \in C(\Psi)} \text{cov}(\Phi', \Psi) = 1 \quad (3.94)$$

$$\sum_{\Psi' \in D(\Phi)} \text{cer}(\Phi, \Psi') = 1 \quad (3.95)$$

$$\pi(\Psi) = \sum_{\Phi' \in C(\Psi)} \text{cer}(\Phi', \Psi) \cdot \pi_S(\Phi') = \sum_{\Phi' \in C(\Psi)} \sigma(\Phi', \Psi) \quad (3.96)$$

$$\pi(\Phi) = \sum_{\Psi' \in D(\Phi)} \text{cov}(\Phi, \Psi') \cdot \pi_S(\Psi') = \sum_{\Psi' \in D(\Phi)} \sigma(\Phi, \Psi') \quad (3.97)$$

$$\text{cer}(\Phi, \Psi) = \frac{\text{cov}(\Phi, \Psi) \cdot \pi(\Psi)}{\sum_{\Psi' \in D(\Phi)} \sigma(\Phi, \Psi')} = \frac{\sigma(\Psi, \Phi)}{\pi(\Phi)} \quad (3.98)$$

$$\text{cov}(\Phi, \Psi) = \frac{\text{cer}(\Phi, \Psi) \cdot \pi(\Phi)}{\sum_{\Phi' \in C(\Psi)} \sigma(\Phi', \Psi)} = \frac{\sigma(\Phi, \Psi)}{\pi(\Psi)} \quad (3.99)$$

Burada, $\pi(\Psi)$ ve $\pi(\Phi)$ aşağıdaki gibi tanımlanır.

$$\pi(\Psi) = \frac{\text{card}(|\Psi|)}{\text{card}(U)} \quad (3.100)$$

$$\pi(\Phi) = \frac{\text{card}(|\Phi|)}{\text{card}(U)} \quad (3.101)$$

(3.94) ve (3.99) formülleri olasılıksal bir çerçeveye sahiptir. (3.96) ve (3.97) formülleri toplam olasılık formülleri olarak anlaşılabilir. (3.98) ve (3.99) formülleri ise Bayes kuramı biçimindedir. Daha öncede belirtildiği gibi eğer, güç, kesinlik ve örtme faktörleri, olasılıklar olarak yorumlarsak o halde, bu formüller karar algoritmasındaki veri arasında bazı ilişkileri açıklar. Bununla beraber, bu faktörleri bir gerçeklik derecesi olarak deterministik bir yolla yorumlamak da mümkündür.

3.3.4 Akış Çizgelerindeki Bağımlılıklar

$(x, y) \in \mathcal{B}$ olsun. $\sigma(x, y) = \sigma(x) \cdot \sigma(y)$ olduğunda x ve y düğümleri birbirinden bağımsızdır.

Sonuç olarak,

$$\frac{\sigma(x, y)}{\sigma(x)} = \text{cer}(x, y) = \sigma(y) \quad (3.102)$$

$$\frac{\sigma(x, y)}{\sigma(y)} = \text{cov}(x, y) = \sigma(x) \quad (3.103)$$

$$\text{cer}(x, y) > \sigma(y) \quad (3.104)$$

$$\text{cov}(x, y) > \sigma(x) \quad (3.105)$$

yukarıdaki (3.102)-(3.105) ifadelerinin gerçekleşmesi durumunda x veya y 'nin birbirine pozitif olarak bağımlı olduğu söylenebilir.

Benzer şekilde, eğer

$$\text{cer}(x, y) < \sigma(y) \quad (3.106)$$

$$\text{cov}(x, y) < \sigma(x) \quad (3.107)$$

(3.106) ve (3.107) eşitsizlikleri gerçekleşirse, x veya y birbirine negatif olarak bağımlıdır.

Bağımsızlık ve bağımlılık ilişkilerinin simetrik ve istatistikte kullanılan benzer olduğu söylenebilir.

Her $(x, y) \in \mathcal{B}$ için, $\eta(x, y)$ bağımlılık faktörü aşağıdaki gibi tanımlanır.

$$\eta(x, y) = \frac{\text{cer}(x, y) - \sigma(y)}{\text{cer}(x, y) + \sigma(y)} = \frac{\text{cov}(x, y) - \sigma(x)}{\text{cov}(x, y) + \sigma(x)} \quad (3.108)$$

Eğer, $\eta(x, y) = 0$ ise x veya y birbirinden bağımsızdır. Eğer, $-1 < \eta(x, y) < 0$ ise, x veya y negatif olarak bağımlı ve $0 < \eta(x, y) < 1$ ise x veya y birbirine pozitif olarak bağımlıdır (Pawlak, 2004).

3.3.5 Karar Yüzeyi

n değerli her karar algoritması ile n boyutlu Euklidean uzayını birleştirmek mümkündür. Burada, kararların değerleri çizgenin n apsisini belirler ve koşul özellik değerleri (denklik sınıfları) uzayın noktalarını belirler. Karar kurallarının güçleri, karşılık gelen noktaların koordinatları olarak görülmelidir. $\delta(x, y)$ mesafesi n boyutlu karar uzayında x ve y noktaları arasındaki mesafe

$$\delta(x, y) = \sqrt{\sum_{i=1}^n (x_i - y_i)^2} \quad (3.109)$$

şeklinde tanımlanır. Burada, $x = (x_1, x_2, \dots, x_n)$ ve $y = (y_1, y_2, \dots, y_n)$ karşılık gelen karar kurallarının vektörleridir (Pawlak, 2003).

4. BİÇİMSEL KAVRAM ÇÖZÜMLEME

Son yıllarda ilgi çeken araştırma konuları arasında yer alan Biçimsel Kavram Çözümleme (Formal Concept Analysis), bilimsel verilerin matematiksel analizi için kuramsal bir yöntemdir. Biçimsel Kavram Çözümleme (BKÇ), 80'li yılların ortalarında Alman bilim adamı Wille (1982) tarafından kafes kuramının uygulamaları için bir çerçeve çalışması geliştirilirken ortaya çıkmıştır. BKÇ' nin ana fikrinin, Galois bağlantılarının cebrik prensiplerine dayanarak özelliklerin kümelenmesi olduğu söylenebilir. Bu da kavram kafesi olarak adlandırılan kısmi sıralı küme oluşturmaktadır. BKÇ gerçek dünyayı nesnelere ve özelliklere (niteliklere) olarak modelleyen bir yöntemdir. BKÇ nesnelere taşıdıkları özellikler arasında bir ilişki kurulabileceğini varsayar ve kavram adı verilen birimler tanımlar. Bir kavramı tanımlamak için özellik ve nesne kümelerinden oluşan bir ikili kullanılır. Bu ikilide yer alan nitelik kümesi ortak nesnelere taşıdıkları en büyük nesnelere kümesidir. Kavramlar arasındaki alt kavram, üst kavram ilişkisi bir kavramın sıra düzensizliği elde edilmesine yol açar. Gerçek hayatta da hiyerarşiler sıklıkla karşımıza çıkar. Kısmi sıralı kümeler ve kafes kuramları BKÇ ve gerçek hayatta karşılaştığımız hiyerarşilerin çözümlemesi için temel sağlar. BKÇ, kavramları verilen içerik içinde tanımlar ve kavramlar arasındaki kesin ilişkiyi, içeriğe karşılık gelen kafes yapısını kullanarak inceler. Kavram, matematiksel kavram olarak kökünü biçimsel mantıktan almıştır. Bununla birlikte kavram çeşitli disiplinlerde genel bir mekanizma olarak karşımıza çıkmaktadır. Bu genel tanım *kapsam (extent)* ve *içerik (intent)* olmak üzere iki türlü yapılabilir. İçerik kavramın özelliklerini, kapsam ise kavramda yer alan nesnelere verir. Birkaç örneği şöyle sıralayabiliriz. Ele alınan veri modeline bağlı olarak ilişkisel veritabanında ilişkisel şemayı içerik, bilgi erişim sistemlerinde ise sunulan sorgu için döndürülen ilişkili belgelerde kapsama karşılık gelmektedir. Veri madenciliğinde, kavramın içeriği, etiketli sınıflama veya etiketsiz kümeleme teknikleri ile verilen nesnelere kümesinden elde edilir. Makine öğreniminde ise, gösterim biçimindeki farklılığa rağmen (ilişkisel, kavram ağaçları vs.) ya ortak özellikler ile ya da ortak özelliklere sahip nesnelere ile tanımlanabilir. Nesnelere taşıdıkları özelliklere göre gruplanmasına *kavramlaştırma* denir.

Son yirmi yıldır BKÇ, kümeleme (clustering), veri analizi (data analysis), bilgi erişimi (information retrieval), bilgi keşfi (knowledge discovery), yazılım mühendisliği (software engineering), durum tabanlı nedenleme ve ontoloji mühendisliği (ontology engineering) için çok güçlü bir araç haline gelmiş ve birçok bilimsel projede kullanılmıştır. BKÇ' nin bu önemli gelişimi altında yatan neden, BKÇ' nin birçok yorumunun kolayca etkilenebilir olması böylece bağlantıların farklı alanlar ile yapılabilir olması ve farklı disiplinlerden araştırmacılar

tarafından kullanılabilir olmasıdır. (Zhang, 2004).

4.1 Biçimsel Kavram Çözümlemede Temel Kavramlar

BKÇ, biçimsel içerik ve kavram üzerine inşa edilmiştir. Bu yöntem, sembolik/kategorik verilerin matematiksel analizi için kafes tabanlı (Zhang vd., 2005) olup çok büyük verilerdeki kavramsal yapıların keşfi için bir öğrenme tekniğidir (Alonso vd., 2004).

BKÇ' nin matematiksel temeli Birkhoff' un kafes kuramına dayanmaktadır. Birkhoff bir kafesin aşağıdaki gibi olabileceğini göstermiştir:

1. Orijinal ilişkinin örtülü yapısını ortaya çıkaran belirli nesnelere ve özellikler arasındaki her ikili ilişkiyi inşa edilebilir.

2. Orijinal ilişkiye geri dönüştürülebilir.

Bu kafes, tablolulu veride mevcut olan örtülü ilişkilerin bir çizge tabanlı görselleştirilmesini sağlamaktadır (Tam, 2004).

Aşağıda BKÇ ile ilgili temel tanımlara yer verilmiştir.

Tanım 4.1: Biçimsel içerik, G , M kümeler ve I alt küme $I \subseteq G \times M$, G ve M arasında bir ikili bir bağıntı olmak üzere (G, M, I) üçlüsüne denir. G ' nin elemanlarına nesnelere, M ' nin elemanlarına da bu nesnelere sahip olduğu özellik veya nesnenin özelliği olarak bakılabilir. g nesnesi ve m özelliği için $(g, m) \in I$ veya $g \in I_m$ şu demektir: g nesnesi m özelliğine sahiptir. Aşağıda verilen örnek tabloda, $G = \{N_1, N_2, \dots, N_8\}$, 8 öğrenci ve $M = \{S\text{-Siyah}, S\text{-Sarı}, S\text{-Kırmızı}, G\text{-Mavi}, G\text{-Kahverengi}\}$ özellikleri saç ve göz rengi olan 5 tane özellik olan bir küme olsun. Tablo 4.1' de i . satır ve j . sütunun olduğu hücredeki X ' in anlamı, i . satırdaki nesnenin j . sütundaki özelliğe sahip olduğudur. Aşağıdaki tabloda bir içerik örneği gösterilmektedir (Saquer, 2000; Saquer ve Deogun, 2001).

Çizelge 4.1 Bir bağlam örneği (Saquer, 2000)

Öğrenci	Saç Rengi			Göz Rengi	
	Siyah	Sarı	Kırmızı	Mavi	Kahverengi
N_1			X		X
N_2		X		X	
N_3		X		X	
N_4		X		X	
N_5		X			X
N_6	X				X
N_7	X				X
N_8	X				X

Tanım 4.2: (G, M, I) biçimsel içerik olsun. $A \subseteq G$ ve $B \subseteq M$ olmak üzere, α ve β operatörleri aşağıdaki gibi tanımlanır.

$$\beta: 2^G \rightarrow 2^M, \beta(A) = \{ m \in M \mid g \text{ Im } \forall g \in A \} \quad (4.1)$$

$$\alpha: 2^M \rightarrow 2^G, \alpha(B) = \{ g \in G \mid g \text{ Im } \forall m \in B \} \quad (4.2)$$

$\beta(A)$, A kümesindeki bütün nesnelere ortak olan özellikler kümesini ifade eder. Benzer şekilde, $\alpha(B)$ fonksiyonu, B 'deki bütün özelliklere sahip G 'nin elemanlarını ifade eder. Diğer bir ifadeyle, $\beta(A)$, A 'daki bütün nesnelere tarafından paylaşılan özelliklerin azami kümesine ve $\alpha(B)$, B 'deki, bütün özelliklere sahip azami nesne kümesini vermektedir. (β, α) ikilisi, 2^G ve 2^M arasında bir Galois bağlantısı oluşturacaktır. (β, α) ikilisinin bir Galois ikilisi olduğunda aşağıdaki Yardımcı Teorem 4.1 geçerlidir.

Yardımcı Teorem 4.1: (G, M, I) bir içerik olsun. Bu durumda aşağıdaki önermeler geçerlidir: (Saquer ve Deogun, 2001)

$$1. \forall A_1, A_2 \subseteq G \text{ için, } A_1 \subseteq A_2 \Rightarrow \beta(A_1) \supseteq \beta(A_2) \quad (4.3)$$

$$\forall B_1, B_2 \subseteq M \text{ için, } B_1 \subseteq B_2 \Rightarrow \alpha(B_1) \supseteq \alpha(B_2) \quad (4.4)$$

$$2. \text{ Tüm } A \subseteq G \text{ için, } A \subseteq \alpha(\beta(A)) \text{ ve } \beta(A) = \beta(\alpha(\beta(A))) \quad (4.5)$$

$$\text{Tüm } B \subseteq M \text{ için, } B \subseteq \beta(\alpha(B)) \text{ ve } \alpha(\beta) = \alpha(\beta(\alpha(B))) \quad (4.6)$$

Tanım 4.3: $A \subseteq G$ ve $B \subseteq M$ olduğunda (G, M, I) içeriğindeki bir atomik kavram bir (A, B) ikilisidir. Öyle ki $\beta(A) = B$ ve $\alpha(B) = A$ olur. (A, B) kavramı için A 'ya kapsam B 'ye içerik denir.

Örneğin, yukarıdaki Tablo 4.1' deki içerik örneği dikkate alınacak olursa, (A, B) ikilisi $A = \{N_6, N_7, N_8\}$ ve $B = \{S\text{-Siyah, G-Kahverengi}\}$ bir atomik kavramdır. Diğer yandan, (A, B) ikilisi $A = \{N_1, N_2, N_3\}$ ve $B = \{S\text{-Sarı, G-Mavi}\}$ bir atomik kavram değildir. Çünkü $\alpha(B) = \{N_2, N_3, N_4\} \neq A$ dir.

$A = \{N_6, N_7, N_8\}$ kümesine, yukarıdaki örneğin kapsamı, $B = \{S\text{-Siyah, G-Kahverengi}\}$ kümesine içeriği denir. Diğer yandan, $\{N_1, N_5, N_6\}$ hiçbir kavramın dışı değildir. Yukarıdaki örnekten açıkça görülmektedir ki, $\forall A \subseteq G$ bir kavrama karşılık gelmemektedir. Yani, bazı A 'lar herhangi bir kavramın dışı olmayabilir. Aynı şekilde B 'de herhangi bir kavramın içi olmayabilir ya da $B \subseteq M$ 'ye bir kavram karşılık gelmemektedir. Kavram tanımlayan alt kümelere uygun kümeler denir.

Tanım 4.4: $B \subseteq M$ kümesine eğer, $\beta(\alpha(B)) = B$ ise diğer bir ifadeyle, B kümesi sadece $(\alpha(B), B)$ kavramının içeriği ise uygun iç (veya B 'ye uygun özellikler kümesi) olarak adlandırılır. Benzer şekilde, $A \subseteq G$ kümesi $\alpha(\beta(A)) = A$ veya diğer bir deyişle, $(A, \beta(A))$ bir kavram ve A kümesi sadece bu kavramın kapsamı ise A 'ya uygun küme (veya uygun nesnelere kümesi) denir. Uygun olmayan kümeye, elverişsiz küme denir.

Örneğin, Tablo 4.1' deki içerik göz önüne alınırsa, $\{N_6, N_7, N_8\}$ 'e uygun dış, $B = \{S\text{-Siyah, G-Kahverengi}\}$ 'e uygun iç denir. $\{N_1, N_5, N_6\}$ 'ya elverişsiz dış ve $B = \{S\text{-Siyah, G-Mavi}\}$ 'e elverişsiz iç denir.

(G, M, I) içeriğinin bütün kavramlar kümesini, $\mathcal{C}(G, M, I)$ olarak gösterelim. $\mathcal{C}(G, M, I)$ üzerinde aşağıdaki gibi bir sıralama bağıntısı tanımlanabilir.

(A_1, B_1) ve (A_2, B_2) , $\mathcal{C}(G, M, I)$ 'de iki kavram olsun.

$$(A_1, B_1) \leq (A_2, B_2) \Leftrightarrow A_1 \subseteq A_2 \text{ (veya } \cong B_1 \supseteq B_2) \quad (4.7)$$

(4.7) deki ifade de (A_1, B_1) 'e (A_2, B_2) 'nin alt kavramı denir. (A_2, B_2) 'ye ise (A_1, B_1) 'in üst kavramı denir. $\mathcal{L}(G, M, I) \leq$ kısmi sıralama bağıntısı ve (G, M, I) 'nin bütün kavramlarının kümesi olan $\mathcal{C}(G, M, I)$ 'nin oluşturduğu kısmi sıralama kümesini gösterir. $\mathcal{L}(G, M, I)$, $(\mathcal{C}(G, M, I), \leq)$ şeklinde gösterilir. Wille' nin kavram kafesleri üzerine temel teoremi $\mathcal{L}(G, M, I)$ 'nin bir tam kafes olduğunu ifade eder. Bu kafese (G, M, I) içeriğinin kavram kafesi adı verilir. Kavramlar arasındaki, alt kavram üst kavram bağıntısı kavram hiyerarşisine yol açtığından sıralı kümeler kuramı ve kafesler, kavram hiyerarşilerinin biçimsel çözümlemesi için uygun bir çalışma ortamı oluşturmaktadır. Şekil 4.1' de Tablo 4.1'in kavram kafesi gösterilmektedir. Veriyi içerik olarak yorumlayan bir uygulamada, veri birleştirme, özellik bağımlılıklarının çözümlenmesi, nesnelerin sınıflandırılması, verinin anlaşılması veya açıklanması için BKÇ kullanılabilir (Saquer, 2000).

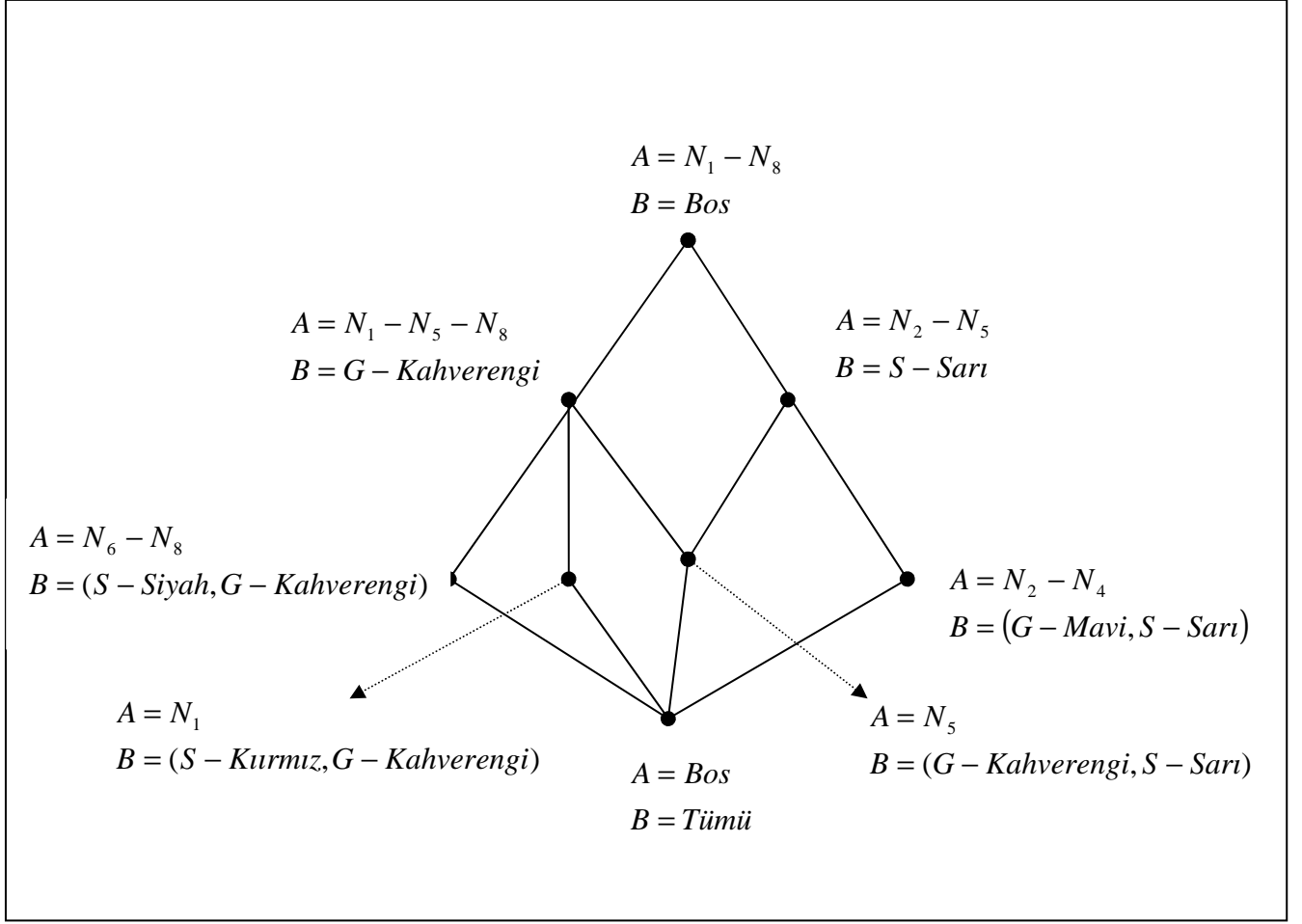
Teorem 4.1: (G, M, I) bir içerik olsun. (G, M, I) üzerindeki $\mathcal{C}(G, M, I)$ tanımlanabilir kavramlar kümesi bir tam kafes oluşturur. O halde, $(\mathcal{C}(G, M, I), \leq)$ bir tam kafestir. Suprema ve Infima aşağıdaki şekilde tanımlanır (Deogun ve Saquer, 2004).

$$\bigvee_{j \in J} (A_j, B_j) = \left(\alpha \left(\bigcap_{j \in J} B_j \right), \bigcap_{j \in J} B_j \right) \quad (4.8)$$

$$\bigwedge_{j \in J} (A_j, B_j) = \left(\bigcap_{j \in J} A_j, \beta \left(\bigcap_{j \in J} A_j \right) \right) \quad (4.9)$$

4.2 Problemin Formüle Edilmesi

Bir tam kafes veri çözümleme için bir kavramsal araç sağlar. BKÇ' nin temel hipotezlerinden birisi, verinin kavram kafesi (concept lattice) adı verilen bir kafes şeklinde temsil edilebileceğidir. Daha sonra bu kafes yapısı, verinin çözümlenmesi için kullanılır. Wille' nin kavram tanımı, bu tanımla modellenemeyecek sayısız uygulama olduğu için çok kısıtlayıcıdır. Wille' nin kavram tanımının modelleyebildiği uygulama türleri uygun nesne ve uygun özellik kümelerine ilişkin uygulamalardır. Örneğin, bilgi erişimi gibi uygulamalarda, belgeleri nesnelere, belgelerin içindeki terimleri özellikler olarak modellemek mümkündür. Bu modelleme ortak terim içeren dokümanların (belgelerin) ancak ve ancak bu ortak terimler bir uygun iç oluşturmuyorsa erişilmesine imkân sağlamaktadır. Ne var ki, uygun bir iç oluşturmayan bir ortak terimler kümesini paylaşan belgelerinde erişilmesine izin verilirse daha faydalı olması söz konusu olabilir. Üstelik daha genel bir kavram tanımı verilirse kavram kafesi daha fazla uygulamalara imkân tanıyabilir.



Şekil 4.1 Çizelge 4.1’de verilen içeriğin kavram kafesi (Saquer, 2000)

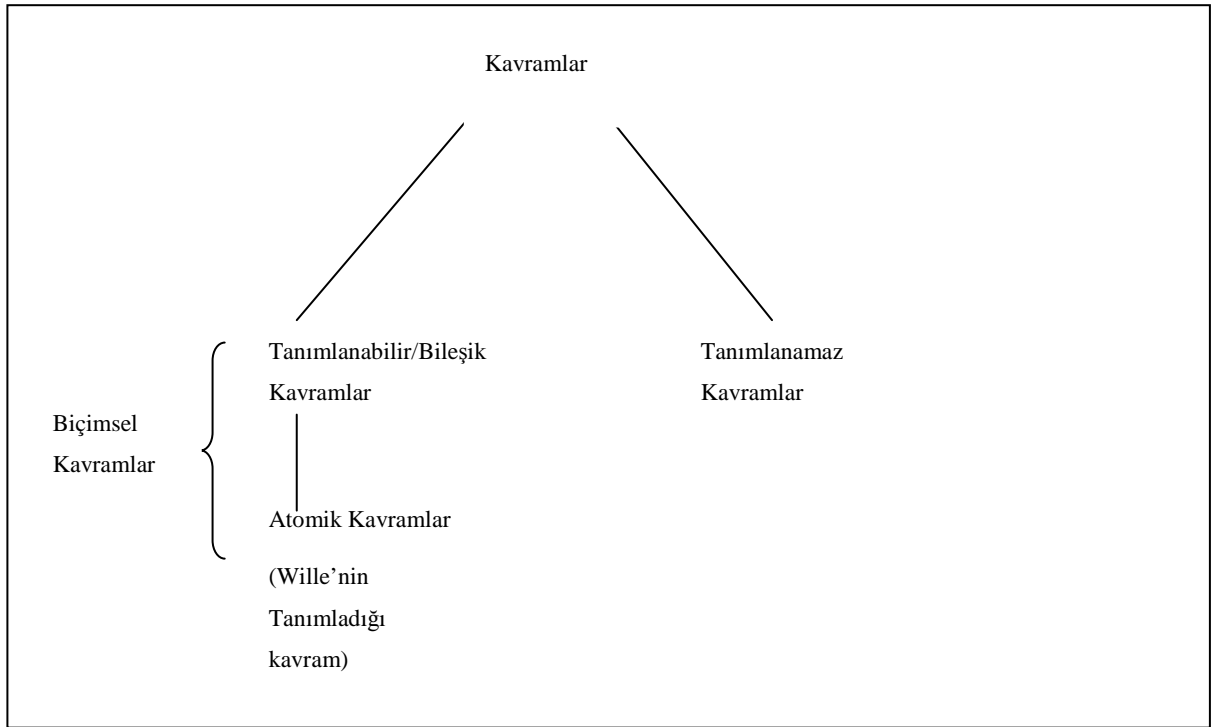
4.3 Tanımlanabilir Kavramlar

Bilgi erişim kuramında eğer her bir sınıf, sınıf içerisindeki tüm nesnelere tarafından paylaşılan gerekli ve yeterli özellikler kümesini tarif eden bir kural tanımlanmış ise bu sınıflama monotetik sınıflama olarak adlandırılır. Yani bir monotetik sınıflamanın her bir sınıfı bir atomik kavramdır. Diğer taraftan, sınıfların her biri sınıf içerisindeki nesnelere tanımlamaya yeterli kurallar kümesi varsa bu sınıflama politetik sınıflama olarak adlandırılır. Buradan da anlaşılacağı üzere politetik sınıflandırma, atomik kavramlardan daha geneldir ve atomik kavramlar cinsinden doğrudan temsil edilmeyebilir. Ne var ki, gerçek hayattaki birçok uygulama politetik benzeri kavramları içerir. Tanımlanabilir kavramlar, Wille’ nin atomik kavram olarak adlandırdıklarının genelleştirilmesini temsil eden biçimsel kavramlardır (Saquer, 2000).

4.4 Tanımlanamaz Kavramlar

Belge ve görüntü erişimi gibi birçok uygulamalarda, nesnelerin taşıdıkları özellikler belirsiz veya kesin değildir. Örneğin, aynı belge kümesi içinde bir kullanıcı ile ilgili olan belgeler ile diğer bir kullanıcı ile ilgili olan belgeler kümesi, her iki kullanıcı aynı konular hakkında araştırma yapıyor olsalar dahi tamamen örtüşmeyebilir. Bu nedenle kullanıcı olurlu olmayan bir nesnelere kümesinin özelliklerini sorgulamak gerekebilir. Yani, kullanıcının sorgusuna mevcut kavram sıra düzeninde cevap vermek mümkün olmayabilir. Bu durumda bir kavram çifti elde etmek mümkün değildir, buna tanımlanamaz kavram adı verilir. Tanımlanamaz kavram terimi ilk olarak kaba küme kuramı kullanılarak ifade edilmiştir (Sever ve Oğuz, 2003).

Tanımlanamaz kavramlara yaklaşmak için kaba küme yaklaşımı bu bölümün ilerleyen alt bölümlerinde ayrıntılı bir şekilde incelenecektir.



Şekil 4.2 Kavramlara genel bir bakış (Sever ve Oğuz, 2003)

(G, M, I) bir içerik olsun. B 'nin elemanlarının kesişimlerini temsil eden bir Boolean kesişim ifadesini her $B \subseteq M$ özellikler kümesi ile ilişkilendirelim. Örneğin; $B = \{a, b, c\}$ ise, B 'nin Boolean kesişim ifadesi $\{a \wedge b \wedge c\}$ olur. Bir Boolean kesişim ifadesi, eğer karşılık gelen özellikler kümesi uygun ise buna da uygun denir. Boolean kesişim ifadelerinin birleşimine bir Boolean formül veya kısaca formül denir. Öyle ki, eğer B_1, B_2, \dots, B_n Boolean kesişim

ifadesi ise,

$$F = B_1 \vee B_2 \vee \dots \vee B_n = \bigvee_{i=1}^n B_i \quad (4.10)$$

bir formüldür. Bir F formülü eğer, her bir B_i uygun ise, $F = \bigvee_{i=1}^n B_i$ formülüne uygun denir.

Aksi takdirde, F formülü elverişsizdir.

Örneğin, Tablo 4.1' de verilen aşağıdaki üç farklı F_1 , F_2 ve F_3 içerikleri uygun formüllerdir.

$$F_1 = \{ \{ \text{S-Kırmızı} \wedge \text{G-Kahverengi} \} \vee \{ \text{S-Sarı} \wedge \text{G-Kahverengi} \} \}$$

$$F_2 = \{ \text{S-Sarı} \wedge \text{G-Kahverengi} \} \text{ ve}$$

$$F_3 = \{ \{ \text{S-Siyah} \wedge \text{G-Kahverengi} \} \vee \{ \text{S-Kırmızı} \wedge \text{G-Kahverengi} \} \vee \{ \text{S-Sarı} \wedge \text{G-Mavi} \} \}$$

\wedge yerine kolaylık olsun diye (\cdot) kullanacağız. Bundan dolayı F_1 , F_2 ve F_3 formülleri, aşağıdaki gibi basit biçimde yazılabilir:

$$F_1 = \{ \{ \text{S-Kırmızı}, \text{G-Kahverengi} \} \vee \{ \text{S-Sarı}, \text{G-Kahverengi} \} \}$$

$$F_2 = \{ \text{S-Sarı}, \text{G-Kahverengi} \} \text{ ve}$$

$$F_3 = \{ \{ \text{S-Siyah}, \text{G-Kahverengi} \} \vee \{ \text{S-Kırmızı}, \text{G-Kahverengi} \} \vee \{ \text{S-Sarı}, \text{G-Mavi} \} \}.$$

Tanım 4.6: B_1 ve B_2 , iki kesişimsel Boolean ifadesi olsun. Bunun anlamı, G 'de B_1 'i sağlayan bütün nesnelere aynı zamanda B_2 'yi de sağlar. (B_1, B_2 'nin alt kümesi demek değildir.)

Örneğin, $\{ \text{G-Mavi} \} \rightarrow \{ \text{S-Sarı}, \text{G-Mavi} \}$

Yardımcı Teorem 1'den herhangi bir B kesişim Boolean ifadesi için $\beta(\alpha(B)) \rightarrow B$ olduğunu çıkartabiliriz.

Tanım 4.7: Bir kesişimsel Boolean ifade B için $\delta(B)$ 'yi, $\delta(B) = \{ g \in G \mid g \text{ Im}, \forall m \in B \text{ için } \}$ şeklinde tanımlayalım. Yani, $\delta(B)$, B kesişim Boolean ifadesi ile ilişkili bütün özellikleri içeren bütün nesnelere kümesidir. $\delta(B) = \alpha(B)$ her kesişim Boolean ifade için eğer, $g \in \delta(B)$ ise g 'ye B kesişim Boolean ifadesini sağlıyor denir. Benzer şekilde, F formülü oluşturan en az bir kesişim Boolean ifadesini sağlıyorsa g 'ye F formülünü sağlıyor denir. $\delta(F)$, F 'yi sağlayan bütün nesnelere kümesi olarak tanımlanır.

Tanım 4.8: Bir Boolean $F = \bigvee_{i=1}^n B_i$ formülü için $\delta(F)$ aşağıdaki gibi tanımlanır:

$$\delta(F) = \bigcup_i \delta(B_i) \quad (4.11)$$

B_i, F 'dedir. Yani, $\delta(F) = \{ g \in G \mid 1 \leq i \leq n \text{ için } g, B_i \text{ 'leri sağlar.} \}$ F formülünü sağlayan bütün nesnelerin kümesidir. Örneğin, Tablo 4. 1' de verilen içerik için, $B = \{ \text{S-Kırmızı, G-Kahverengi} \}$ ve $F = \{ \{ \text{S-Kırmızı, G-Kahverengi} \} \vee \{ \text{S-Sarı, G-Kahverengi} \} \}$ olsun. Hem B , hem de F uygundur. Bunun yanında $\delta(B) = \{N_1\}$ ve $\delta(F) = \{N_1, N_5\}$ dir.

Tanım 4.9: $A \subseteq G$ nesnelere kümesi eğer atomik dışların sonlu birleşimi olarak yazılabiliyorsa buna tanımlanabilir dış denir. Ancak ve ancak $1 \leq i \leq n$ için $A = \bigcup_{i=1}^n A_i$, $\forall A_i$ bir atomik dış ise, A kümesi tanımlanabilir dıştır. Benzer şekilde, $B \subseteq M$ özellikler kümesi eğer, atomik içlerin sonlu birleşimi olarak yazılabiliyorsa, tanımlanabilir iç denir. A nesnelere kümesi uygun olduğu zaman (bir atomik kavramın dışı ise) A hem tanımlanabilir, hem de uygundur. Dolayısıyla, uygun dış, tanımlanabilir dışın özel bir halidir. Benzer bir şekilde, uygun iç, tanımlanabilir için özel bir halidir. Her atomik A_i dışı, A_i 'deki her nesnenin B_i 'yi sağladığı bir kesişim Boolean ifadesi ile ilişkilendirilir. Benzer şekilde, $A = \bigcup_{i=1}^n A_i$ tanımlanabilir dışı, $1 \leq i \leq n$ için her bir (A_i, B_i) 'nin bir atomik kavram olduğu $F = \{B_1, B_2, \dots, B_n\}$ Boolean formülü ile ilişkilendirilebilir.

Tanım 4.10: $A_i \subseteq G$ nesnelere kümesi için $\gamma(A_i)$, $\beta(A_i)$ 'lerin ilişkilendirildiği kesişim Boolean ifadesi olarak tanımlanır.

$A = \bigcup_{i=1}^n A_i$ için $\gamma(A) = \bigvee_{i=1}^n \gamma(A_i)$ şeklinde tanımlanır. γ 'yi, δ operatörünün duali (veya tersi) olarak düşünebiliriz. Yani, eğer A ve F bir önceki paragraftaki gibi ise $\delta(F) = A$ ve $\gamma(A) = F$ 'dir.

Tanım 4.11: F_1 ve F_2 iki Boolean formül olsun. Eğer, $\delta(F_1) \subseteq \delta(F_2)$ ise, F_1, F_2 'yi kapsar denir ve $F_1 \rightarrow F_2$ olarak ifade edilir. F_i ve F_j 'lerin her biri tek bir kesişim Boolean ifadesi içeriyorsa Tanım 4.11, Tanım 4.6' ya indirgenir. Bu tanımdan, G 'de F_1 'i sağlayan bütün nesnelere, aynı zamanda F_2 'yi de sağlıyorsa $F_1 \rightarrow F_2$ olduğu açıktır. Örneğin,

$F_1 = \{ \text{S-Kırmızı, G-Kahverengi} \}$, $F_2 = \{ \text{G-Kahverengi, S-Sarı} \}$, $F_3 = \{ \text{G-Kahverengi} \}$ ve $F_4 = \{ \{ \text{S-Kırmızı, G-Kahverengi} \} \vee \{ \text{S-Sarı, G-Kahverengi} \} \}$ ise

O halde, $\delta(F_1) = \{N_1\}$, $\delta(F_2) = \{N_5\}$, $\delta(F_3) = \{N_1, N_5, N_6, N_7, N_8\}$ ve $\delta(F_4) = \{N_1, N_5\}$.
Dolayısıyla yukarıdaki ifadelerden aşağıdaki çıkarımlar geçerlidir:

$$F_1 \rightarrow F_3, F_2 \rightarrow F_3, F_1 \rightarrow F_4, F_2 \rightarrow F_4 \text{ ve } F_4 \rightarrow F_3.$$

Tanım 4.12: A tanımlanabilir dış, F uygun Boolean formül $\delta(F) = A$, $\gamma(A) = F$, $1 \leq i \leq m$ için her (A_i, B_i) bir atomik kavram olmak üzere, (G, M, I) içeriğindeki (A, F) ikilisi tanımlanabilir kavram olarak adlandırılır. (A, F) tanımlanabilir kavramı için A 'ya tanımlanabilir kavramın dışı, F 'ye tanımlanabilir kavramın içi denir. Örneğin $C_1 = (A_1, F_1)$, $C_2 = (A_2, F_2)$ ve $C_3 = (A_3, F_3)$ Tablo 4.1' de tarif edilen içerik için üç tanımlanabilir kavramlardır. Bunların dışı,

$$A_1 = \{\{N_1\}, \{N_5\}\} = \{N_1, N_5\}, A_2 = \{N_5\} \text{ ve}$$

$$A_3 = \{\{N_6, N_7, N_8\}, \{N_1\}, \{N_2, N_3, N_4\}\} = \{N_1, N_2, N_3, N_4, N_6, N_7, N_8\} \text{ ve içleri}$$

$$F_1 = \{\{ \text{S-Kırmızı}, \text{G-Kahverengi} \} \vee \{ \text{S-Sarı}, \text{G-Kahverengi} \} \},$$

$$F_2 = \{ \text{S-Sarı}, \text{G-Kahverengi} \} \text{ ve}$$

$$F_3 = \{\{ \text{S-Siyah}, \text{G-Kahverengi} \} \vee \{ \text{S-Kırmızı}, \text{G-Kahverengi} \} \vee \{ \text{S-Sarı}, \text{G-Mavi} \} \}$$

Boolean formülleridir.

Atomik kavramın dış kümesinin bir atomik dış ve iç kümesinin tek bir kesişim Boolean ifadesinden oluşan bir Boolean formülü olduğu durumda, tanımlanabilir kavramın özel bir durumu söz konusudur.

$\mathcal{C}(G, M, I)$, (G, M, I) içeriğinin bütün tanımlanabilir kavramlar kümesini gösterebilir. (G, M, I) üzerinde aşağıdaki şekilde bir sıralama bağıntısı tanımlanabilir.

Tanım 4.13: (A_1, F_1) ve (A_2, F_2) (G, M, I) 'de iki tanımlanabilir kavramlar olsunlar.

$$(A_1, F_1) \leq (A_2, F_2) \Leftrightarrow A_1 \subseteq A_2 \text{ (veya } \cong F_1 \rightarrow F_2) \quad (4.12)$$

(A_1, F_1) 'e (A_2, F_2) 'nin alt kavramı, (A_2, F_2) 'ye (A_1, F_1) 'in üst kavramı olarak adlandırılır (Saquer, 2000).

4.5 Tanımlanabilir Kavramların Matematiksel Özellikleri

Bu bölümde tanımlanabilir kavramların matematiksel özellikleri verilecektir.

Yardımcı Teorem 4.2: (G, M, I) bir içerik, $A_1, A_2, A \subseteq G$ ve $F_1, F_2, F \subseteq M$ olsun. O halde aşağıda verilen sonuçlar geçerlidir (Saquer, 2000).

$$A_1 \subseteq A_2 \Rightarrow \gamma(A_1) \rightarrow \gamma(A_2) \quad (4.13)$$

$$F_1 \rightarrow F_2 \Rightarrow \delta(F_1) \subseteq \delta(F_2) \quad (4.14)$$

$$A \subseteq \delta(\gamma(A)) \quad (4.15)$$

$$\gamma(\delta(F)) \rightarrow F \quad (4.16)$$

Yukarıdaki Yardımcı Teorem' den görülebilir ki, γ ve δ operatörleri G ve M ' nin güç kümeleri arasında bir Galois bağlantısı oluşturur. Yardımcı Teoremin ispatı için Saquer' in (2000) çalışmasına bakılabilir.

Önerme 4.1: (G, M, I) bir içerik olsun. $A = \bigcup_{i=1}^n A_i$ ve $F = \bigvee_{i=1}^m B_i$ o halde, aşağıda verilen sonuçlar geçerli olacaktır.

$$\gamma(A) = \gamma(\delta(\gamma(A))) \quad (4.17)$$

$$\delta(F) = \delta(\gamma(\delta(F))) \quad (4.18)$$

$$A \subseteq \delta(F) \Leftrightarrow \gamma(A) \rightarrow F \quad (4.19)$$

Bu önermede verilen sonuçlar Galois bağlantılarının özelliklerinden çıkarılabilir.

4.6 Kavram Yaklaşımlarına Kaba Kümeler Yaklaşımı

BKÇ ve kaba kümeler kuramı veri analizi ile ilgili ve birbirini tamamlayan iki yöntemdir. Bu yöntemler, kavramlar fikrine farklı bakış açılarından bakar. Kaba kümeler kuramının ana fikri, nesnelerin ayırtelemesizliği ile ilgili olarak bir kümenin özellikleri ve yaklaşım operatörlerinin indirgenmesidir. Bunun yanında BKÇ' nin ana fikri biçimsel kavramlar ve kavram kafesleridir (Yao, 2004c; 2004d). Bu bölümde biçimsel kavramlara yaklaşmak için kaba kümeler yaklaşımı incelenecektir. Burada incelenecek yaklaşımlar (Saquer ve Deogun, 2001) birçok yeni özelliğe sahiptir ve Kent' in (1994; 1996) çalışmasında verilen mevcut yaklaşıma göre birçok nedenden dolayı belirgin olarak daha gelişmiştir. İlk olarak yaklaşımda kullanılan ilişkiler her zaman aynı cevabın verileceğini garanti eden bir yol ile tanımlanır ve cevap uzman tarafından sağlanan denklik bağıntısına bağlıdır.

İkincisi, kullanılan denklik bağıntısı kullanıcıdan bir müdahale olmadan sistem tarafından otomatik olarak tanımlanır ve bu da yaklaşım yöntemini tamamen otomatik ve kullanıcı bağımsız yapar.

Üçüncü olarak, hem nesnel kümesi, hem de özellikler kümesi tanımlanamayan kavramlara yaklaşmak için kullanılır. Bununla beraber, Kent'in (1994; 1996) çalışmasında sadece özellikler kümesi kullanılır. Bu da aynı nesnel kümesi ile olan çiftlerin kullanılan denklik ilişkisini göz ardı ederek aynı yaklaşımla sonuçlanır.

Dördüncü olarak, içerik kavram yaklaşımı için doğrudan kullanılır. Ancak Kent'in çalışmasında içeriğe ilk olarak alt ve üst yaklaşımlar ile yaklaşılır ve daha sonra kavram yaklaşımı için kullanılır.

Beşinci olarak, yaklaşım tek nesnel kümesine veya tek özellikler kümesine yaklaşan biçimsel kavramları bulmak için yeterlidir.

4.6.1 Mevcut Yaklaşım

Kavramlara yaklaşmak için mevcut yaklaşım (existing approach) Kent'in (1996) çalışmasına dayalıdır. Bu yaklaşım Kaba Kavram Çözümleme olarak adlandırılır ve bir E denklik ilişkisinin varlığına dayanır. G nesnel kümesi üzerinde bir uzman tarafından sağlanır. E 'nin G üzerinde bir denklik bağıntısı olduğu (G, E) çifti yaklaşım uzayı olarak adlandırılır. G nesnelinin, E tanımlanabilir biçimsel içeriği ve M özellikleri bir (G, M, I) biçimsel içeriğidir ve bunun temel kapsamları $\{Im \mid m \in M\}$, $Im = \{g \in G \mid gIm\}$ olduğunda G nesnelinin E tanımlanabilir alt kümeleridir. (G, E) çifti ile ilgili olarak, I 'nin alt ve üst yaklaşımları \underline{I}_E ve \bar{I}_E olarak gösterilir ve sırasıyla aşağıdaki şekilde ifade edilir.

$$\underline{I}_E = \{(g, m) \mid [g]_E \subseteq Im\} \quad (4.20)$$

$$\bar{I}_E = \{(g, m) \mid [g]_E \cap Im \neq \emptyset\} \quad (4.21)$$

(G, M, I) biçimsel içeriğine (G, M, \underline{I}_E) ve (G, M, \bar{I}_E) alt ve üst içerikleri tarafından yaklaşılır. \underline{I}_E ve \bar{I}_E ile ilgili olarak $B \subseteq M$ özellik kümesinin kaba kapsamları aşağıdaki gibi tanımlanır.

$$\alpha(\underline{B}_E) = \underline{\alpha(B)}_E = \bigcap_{m \in B} \underline{Im}_E \quad (4.22)$$

$$\alpha(\bar{B}^E) = \overline{\alpha(B)}^E = \bigcap_{m \in B} \bar{Im}^E \quad (4.23)$$

$(A, B) \in \mathcal{C}(G, M, I)$ herhangi biçimsel kavramına \underline{I}_E ve \bar{I}_E yolu ile yaklaşılabılır. (A, B) çiftinin alt ve üst E yaklaşımları sırasıyla aşağıdaki gibi verilir (Saquer ve Deogun, 1999).

$$\underline{(A, B)}_E = (\alpha(\underline{B}_E), \beta(\alpha(\underline{B}_E))) \quad (4.24)$$

$$\overline{(A, B)}_E = (\alpha(\overline{B}^E), \beta(\alpha(\overline{B}^E))) \quad (4.25)$$

4.6.2 Nesnelere Kümesine Yaklaşım

Bu bölümde Saquer ve Deogun (2001) tarafından sunulan yaklaşımlar incelenecektir.

$A \subseteq G$ nesnelere kümesi verildiğinde kapsam yaklaşımları A olan bir biçimsel kavramın varlığı önemlidir. Bu tür bir kavram A kümesine yaklaşır. Bu durumda aşağıdaki durumlar mevcuttur.

Durum 1: A Kümesi Uygun

$(A, \beta(A))$ bir biçimsel kavram olduğundan dolayı, $(A, \beta(A))$ en iyi yaklaşımdır.

Durum 2: A Kümesi Elverişsizdir

Uygun olmayan A kümesine yaklaşan biçimsel kavramları bulmak kolay değildir. Burada ki yaklaşım, A kümesini bir kaba küme olarak düşünmektir. İlk olarak, A 'nin alt ve üst yaklaşımlarını temsil eden \underline{A} ve \bar{A} tanımlanabilir küme çifti bulunur. Daha sonra, A 'ya yaklaşan iki biçimsel kavramı bulmak için \underline{A} ve \bar{A} alt ve üst yaklaşımları kullanılır. $gI = \{m \in M \mid gIm\}$, g nesnesi tarafından sahip olunan tüm özellikler kümesini gösterebilir. g 'deki R ilişkisi aşağıdaki gibi tanımlanır.

Sadece ve sadece $g_1I = g_2I$ ise, g_1Rg_2 'dir. Burada $g_1, g_2 \in G$ 'dir. Örneğin, iki nesne sadece ve sadece aynı özellikler kümesine sahipler ise, ilişkilidirler. Açıkça, R , G 'de bir denklik ilişkisidir. Bu nedenle, G 'de bir bölümlendirmeye neden olur. G/R , G 'de R tarafından çıkarılan tüm denklik sınıflarının kümesi olsun. (Ayrıca bölüm küme olarak da bilinir) G/R 'nin denklik sınıfları temel kümeler olarak adlandırılır. Temel kümelerin herhangi sonlu birleşimi tanımlanabilir küme olarak adlandırılır. Önerme 4.2 gösterir ki, iki nesne sadece ve sadece eğer aynı nesne kavramına sahipse ve burada g nesnesinin nesne kavramı g 'yi içeren en küçük biçimsel kavram ise ve $(\alpha(\beta(g)), \beta(g))$ verilirse o halde, bu iki nesne aynı denklik sınıfına aittir (Ganter ve Wille, 1999).

Önerme 4.2: Herhangi iki g_1 ve g_2 nesnelere için, sadece ve sadece g_1 'in nesne kavramı g_2 'nin nesne kavramına eşit ise $g_1 R g_2$ olur.

R ile ilgili $A \subseteq G$ 'nin alt ve üst yaklaşımları aşağıdaki gibi tanımlanır.

$$\underline{A} = \{g \in G \mid [g] \subseteq A\} = \bigcup \{X \in G/R \mid X \subseteq A\} \quad (4.26)$$

$$\overline{A} = \{g \in G \mid [g] \cap A \neq \emptyset\} = \bigcup \{X \in G/R \mid X \cap A \neq \emptyset\} \quad (4.27)$$

Şimdi, A 'ya yaklaşan iki biçimsel kavram bulunabilir. Alt ve üst kavram yaklaşımları sıra ile $(\alpha(\beta(\underline{A})), \beta(\underline{A}))$ ve $(\alpha(\beta(\overline{A})), \beta(\overline{A}))$ şeklinde verilir. Eğer A tanımlanabilir ise, $\underline{A} = A = \overline{A}$ eşitliği geçerlidir. Bu durumda, A 'nın alt ve üst yaklaşımları çakışır. Ayrıca, $\underline{A} = \overline{A} = A \subseteq \alpha(\beta(A)) = \alpha(\beta(\underline{A})) = \alpha(\beta(\overline{A}))$ (4.28)

olduğundan elde edilen yaklaşım bir üst yaklaşımdır. Bir nesnelere kümesine yaklaşmak için sözde kod Algoritma 1'de verilmiştir. Giriş, A nesnelere kümesi ve bölüm küme G/R 'dir. Çıkış ise, A kümesi için kavram yaklaşımıdır. Algoritma 1'in zaman karmaşıklığı, $O(|G|^2 |M|^2)$ dir.

1.	eğer $(\alpha(\beta(A)) = A)$ ise // Yani A kümesi uygun
2.	Cevap $\leftarrow (A, \beta(A))$
3.	değilse
4.	$\underline{A} = \bigcup \{X \in G/R \mid X \subseteq A\}$
5.	$\overline{A} = \bigcup \{X \in G/R \mid X \cap A \neq \emptyset\}$
6.	eğer $(\underline{A} = \overline{A})$ // Yani A kümesi tanımlanabilir
7.	Cevap $\leftarrow (\alpha(\beta(\overline{A})), \beta(\overline{A}))$
8.	değilse {
9.	Alt $\leftarrow (\alpha(\beta(\underline{A})), \beta(\underline{A}))$
10.	Üst $\leftarrow (\alpha(\beta(\overline{A})), \beta(\overline{A}))$
11.	Cevap \leftarrow Alt ve Üst }
12.	son

Şekil 4.3 Algoritma 1: bir A nesnelere kümesine yaklaşım için sözde kod (Saquer ve Deogun, 2001)

4.6.3 Özellikler Kümesine Yaklaşım

$B \subseteq M$ özellikler kümesine yaklaşım nesnelere kümesine yaklaşıma benzerdir. Burada da aşağıdaki durumlar söz konusudur.

Durum 1: B Kümesi Uygun

$(\alpha(B), B)$ bir biçimsel kavramdır. Bu nedenle, $(\alpha(B), B)$ en iyi yaklaşımdır.

Durum 2: B Kümesi Elverişsiz

$Im = \{g \in G \mid gIm\}$, m özelliğine sahip olan tüm nesnelere kümesi olsun. M 'deki R' ilişkisi aşağıdaki gibi tanımlanır.

Sadece ve sadece $Im_1 = Im_2$ ise, $m_1 R' m_2$ 'dir. Burada, $m_1, m_2 \in M$ 'dir. Örneğin iki özellik sadece ve sadece aynı nesnelere kümesi tarafından sahip olunursa o halde ilişkilidir denir. R' ilişkisi, M 'de denklik ilişkisidir. M/R' , M 'de R' tarafından çıkarılan tüm denklik sınıflarının kümesini gösterebilir. M/R' denklik sınıfları temel kümeler olarak adlandırılır ve temel kümelerin herhangi sonlu birleşimi bir tanımlanabilir küme olarak adlandırılır. $m \in M$ özellik kavramı, m özelliğini içeren en büyük kavramdır ve $(\alpha(m), \beta(\alpha(m)))$ ile verilir (Ganter ve Wille, 1999).

Aşağıdaki Önerme 4.3, R' ve özellik kavramları arasında bir ilişkiyi göstermektedir.

Önerme 4.3: m_1 ve m_2 herhangi iki özelliği için, sadece ve sadece m_1 'in özellik kavramı, m_2 'nin özellik kavramına eşit ise $m_1 R' m_2$ 'dir.

$B \subseteq M$ 'nin alt ve üst yaklaşımları sırasıyla aşağıdaki gibi tanımlanır.

$$\underline{B} = \{m \in M \mid [m] \subseteq B\} = \bigcup \{Y \in M/R' \mid Y \subseteq B\} \quad (4.29)$$

$$\overline{B} = \{m \in M \mid [m] \cap B \neq \emptyset\} = \bigcup \{Y \in M/R' \mid Y \cap B \neq \emptyset\} \quad (4.30)$$

$B \subseteq M$, uygun olmayan özellikler kümesinin alt ve üst yaklaşımları sırasıyla $(\alpha(\overline{B}), \beta(\alpha(\overline{B})))$ ve $(\alpha(\underline{B}), \beta(\alpha(\underline{B})))$ biçimsel kavramlarıyla verilir. $(\alpha(\overline{B}), \beta(\alpha(\overline{B})))$, $(\alpha(\underline{B}), \beta(\alpha(\underline{B})))$ 'nin alt kavramıdır. Bunu ispatlamak için, $\underline{B} \subseteq B \subseteq \overline{B}$ 'nin $\alpha(\overline{B}) \subseteq \alpha(B) \subseteq \alpha(\underline{B})$ 'yi ifade ettiğini belirtelim. Bu yüzden, B 'nin alt yaklaşım kavramının kapsamı, B 'nin üst yaklaşım kavram kapsamının bir alt kümesidir. Bununla beraber, B tanımlanabilir olduğunda $\underline{B} = B = \overline{B}$ eşitliği mevcuttur. Ayrıca,

$$B \subseteq \beta(\alpha(B)) = \beta(\alpha(\underline{B})) = \beta(\alpha(\overline{B})) \quad (4.31)$$

dir. Bu nedenle, $B \subseteq \beta(\alpha(B))$ olur. Bunun anlamı, yaklaşım formüllerinin ilişkisinde B için alt yaklaşım verdiğiidir.

Özellikler kümesine yaklaşım için sözde kod Algoritma 2’de verilmiştir. Bu algorithmada, B özellikler kümesini ve M/R' bölüm kümesini giriş olarak alır ve B kümesi için kavram yaklaşımını çıktı olarak verir. Algoritma 2, Algoritma 1’e benzerdir. Algoritma 2 için zaman karmaşıklığı, B kümesi uygun olduğunda, $O(|G||M||B||\alpha(B)|)$ 'dir ve B kümesi uygun olmadığıda $O(|G||M|(|\underline{B}||\alpha(\underline{B})| + |\overline{B}||\alpha(\overline{B})|))$ olacaktır. Ancak herhangi nesnelere kümesinin boyutu $|G|$ ile sınırlı olduğundan ve özellikler kümesi $|M|$ ile sınırlı olduğundan dolayı, Algoritma 2 için zaman karmaşıklığı $O(|G|^2|M|^2)$ şeklinde olacaktır.

1.	eğer $(\beta(\alpha(B)) == B)$ ise // Yani B kümesi uygun
2.	Cevap $\leftarrow (\alpha(B), B)$
3.	değilse
4.	$\underline{B} = \bigcup \{Y \in M/R' \mid Y \subseteq B\}$
5.	$\overline{B} = \bigcup \{Y \in M/R' \mid Y \cap B \neq \emptyset\}$
6.	eğer $(\underline{B} == \overline{B})$ // Yani B kümesi tanımlanabilir
7.	Cevap $\leftarrow (\alpha(\overline{B}), \beta(\alpha(\overline{B})))$
8.	değilse {
9.	Alt $\leftarrow (\alpha(\overline{B}), \beta(\alpha(\overline{B})))$
10.	Üst $\leftarrow (\alpha(\underline{B}), \beta(\alpha(\underline{B})))$
11.	Cevap \leftarrow Alt ve Üst }
12.	son

Şekil 4.4 Algoritma 2: bir B özellikler kümesine bir yaklaşım için sözde kod (Saquer ve Deogun, 2001)

4.6.4 Bir Tanımlanamayan Kavrama Yaklaşım

Bir (A, B) çifti verildiğinde $A \subseteq G$ ve $B \subseteq M$ olduğunda kapsam yaklaşımları A ve içerik yaklaşımları B olan biçimsel kavram veya kavramlar bulunmalıdır. Bu tür kavramların (A, B) çiftine yaklaştığı söylenebilir. Burada aşağıdaki dört durum dikkate alınmalıdır.

- Hem A hem de B kümesi uygun.
- A kümesi uygun, B kümesi elverişsiz.
- B kümesi uygun, A kümesi elverişsiz.

- Hem A hem de B kümesi elverişsiz.

Durum 1: Hem A Hem de B Kümesi Uygun

Hem A hem de B kümesinin uygun olması durumunda aşağıdaki gibi iki alt durum mevcuttur.

Durum 1.1: $\beta(A) = B$.

Eğer, $\beta(A) = B$ ise o halde, $\alpha(B)$, A 'ya eşit olmalıdır. Çünkü hem A hem de B kümesi uygundur. Aşağıdaki Önerme 4.4 benzer bir sonuç verir. Böylece (A, B) çifti bir biçimsel kavramdır ve hiçbir yaklaşıma gerek duymaz. Önerme 4.4, Algoritma 3'de kullanılan tanımlanamayan kavramlara yaklaşmak için bir yaklaşım geliştirmede kullanılır.

Önerme 4.4: $A \subseteq G$ ve $B \subseteq M$ olsun. Eğer, A kümesi uygun ve $\beta(A) = B$ ise, o halde B kümesi de uygundur ve $\alpha(B) = A$ 'dir.

Durum 1.2: $\beta(A) \neq B$.

Eğer, $\beta(A) \neq B$ ise (ve bu nedenle, $\alpha(B) \neq A$) o halde $\beta(A) = A'$ ve $\alpha(B) = B'$ olur. A ve B kümelerinin ikisinin de uygun olması durumunda (A, A') ve (B, B') çiftlerinin her ikisinin de (G, M, I) biçimsel kavramlar olduğu görülür.

(A, B) çifti için alt ve üst yaklaşımlar BKÇ' nin temel teoremi kullanılarak ve (A, A') , (B, B') çiftleri için infima ve suprema ile bulunur. Alt ve üst yaklaşımlar için aşağıdaki formüller elde edilir.

$$\underline{(A, B)} = (A \cap B', \beta(A \cap B')) \quad (4.32)$$

$$\overline{(A, B)} = (\alpha(A' \cap B), A' \cap B) \quad (4.33)$$

Durum 2: A Kümesi Uygun, B Kümesi Elverişsiz

A uygun olduğundan, A kümesine $C_1 = (A, \beta(A))$ biçimsel kavramı ile yaklaşılabilir. B uygun olmadığından, B kümesine kaba küme gibi davranılır ve B 'ye yaklaşan biçimsel kavramlar bulunur. Ayrıca, B kümesinin tanımlanamaz olduğu varsayılır.

$C_2 = (\alpha(\overline{B}), \beta(\alpha(\overline{B})))$ ve $C_3 = (\alpha(\underline{B}), \beta(\alpha(\underline{B})))$ sırasıyla B 'nin alt ve üst yaklaşımlarını temsil eden biçimsel kavramları gösterir.

(A, B) çifti için alt ve üst yaklaşımlar aşağıdaki formüller ile bulunur.

$$\underline{(A, B)} = \inf\{C_1, C_2\} = (A \cap \alpha(\overline{B}), \beta(A \cap \alpha(\overline{B}))) \quad (4.34)$$

$$\overline{(A, B)} = \sup\{C_1, C_3\} = (\alpha(\beta(A) \cap \beta(\alpha(\underline{B}))), \beta(A) \cap \beta(\alpha(\underline{B}))) \quad (4.35)$$

Burada infimum ve supremum BKÇ' nin temel teoremi kullanılarak elde edilir (Ganter ve Wille, 1999).

Durum 3: B Kümesi Uygun, A Kümesi Elverişsiz

Bu durum için gerekli işlemler bir önceki duruma göre benzerdir. B kümesine, $C_1 = (\alpha(B), B)$ kavramı ile yaklaşılır ve A kümesine sırasıyla $C_2 = (\alpha(\beta(\underline{A})), \beta(\underline{A}))$ ve $C_3 = (\alpha(\beta(\overline{A})), \beta(\overline{A}))$ alt ve üst yaklaşım kavramları ile yaklaşılır. (A, B) çifti için alt ve üst yaklaşımlar aşağıdaki gibi verilir.

$$\underline{(A, B)} = \inf\{C_1, C_2\} = (\alpha(\beta(\underline{A})) \cap \alpha(B), \beta(\alpha(\beta(\underline{A})) \cap \alpha(B))) \quad (4.36)$$

$$\overline{(A, B)} = \sup\{C_1, C_3\} = (\alpha(\beta(\overline{A}) \cap B), \beta(\overline{A}) \cap B) \quad (4.37)$$

Durum 4: Hem A Kümesi Hem de B Kümesi Elverişsiz

Bu durumda A ve B kümelerine kaba kümeler olarak davranılır. A ve B kümelerinin alt ve üst kavram yaklaşımlarını temsil eden biçimsel kavramlar bulunur.

C_1 ve C_2 , A kümesinin alt ve üst kavram yaklaşımlarını ve C_3 ve C_4 'de sırasıyla, B kümesinin alt ve üst kavram yaklaşımlarını gösterebilir. $C_1 = (\alpha(\beta(\underline{A})), \beta(\underline{A}))$, $C_2 = (\alpha(\beta(\overline{A})), \beta(\overline{A}))$, $C_3 = (\alpha(\overline{B}), \beta(\alpha(\overline{B})))$ ve $C_4 = (\alpha(\underline{B}), \beta(\alpha(\underline{B})))$ sırasıyla nesnelere ve özelliklere kümelerine yaklaşımlardan bulunur.

(A, B) çiftinin alt ve üst yaklaşımları sırasıyla aşağıdaki biçimsel kavramlar ile verilir.

$$\underline{(A, B)} = \inf\{C_1, C_3\} = (\alpha(\beta(\underline{A})) \cap \alpha(\overline{B}), \beta(\alpha(\beta(\underline{A})) \cap \alpha(\overline{B}))) \quad (4.38)$$

$$\overline{(A, B)} = \sup\{C_2, C_4\} = (\alpha(\beta(\overline{A}) \cap \beta(\alpha(\underline{B}))), \beta(\overline{A}) \cap \beta(\alpha(\underline{B}))) \quad (4.39)$$

(A, B) çiftine yaklaşım için sözde kod Algoritma 3'de verilmiştir. Algoritmada giriş, nesnelere A kümesi, özelliklerin B kümesi, G/R ve M/R' bölüm kümeleridir. Çıkış ise, (A, B) çifti için kavram yaklaşımıdır.

Algoritma 3 için zaman karmaşıklığı, $O(|G|^2 |M|^2)$ şeklindedir (Saquer ve Deogun, 2001).

1. eğer $(\alpha(\beta(A)) = A)$ ise // Yani A kümesi uygun
2. eğer $(\beta(A) = B)$ // Durum 1.1
3. Cevap $\leftarrow (A, B)$
4. değilse eğer $(\beta(\alpha(B)) = B)$ // Durum 1.2
5. Alt $\leftarrow (A \cap \alpha(B), \beta(A) \cup B)$
6. Üst $\leftarrow (A \cup \alpha(B), \beta(A) \cap B)$
7. Cevap \leftarrow Alt ve Üst
8. değilse // Durum 2: A kümesi uygun fakat B kümesi değil
9. $\underline{B} \leftarrow \bigcup \{Y \in M / R' \mid Y \subseteq B\}$
10. $\overline{B} \leftarrow \bigcup \{Y \in M / R' \mid Y \cap B \neq \emptyset\}$
11. Alt $\leftarrow (A \cap \alpha(\overline{B}), \beta(A) \cup \beta(\alpha(\overline{B})))$
12. Üst $\leftarrow (A \cup \alpha(\underline{B}), \beta(A) \cap \beta(\alpha(\underline{B})))$
13. Cevap \leftarrow Alt ve Üst
14. son eğer
15. değilse eğer $(\beta(\alpha(B)) = B)$ // Durum 3: B kümesi uygun fakat A kümesi değil
16. $\underline{A} \leftarrow \bigcup \{X \in G / R \mid X \subseteq A\}$
17. $\overline{A} \leftarrow \bigcup \{X \in G / R \mid X \cap A \neq \emptyset\}$
18. Alt $\leftarrow (\alpha(\beta(\underline{A})) \cap \alpha(B), \beta(\underline{A}) \cup B)$
19. Üst $\leftarrow (\alpha(\beta(\overline{A})) \cup \alpha(B), \beta(\overline{A}) \cap B)$
20. Cevap \leftarrow Alt ve Üst
21. değilse // Durum 4: Hem A kümesi hemde B kümesi elverişsiz
22. $\underline{A} \leftarrow \bigcup \{X \in G / R \mid X \subseteq A\}$
23. $\overline{A} \leftarrow \bigcup \{X \in G / R \mid X \cap A \neq \emptyset\}$
24. $\underline{B} \leftarrow \bigcup \{Y \in M / R' \mid Y \subseteq B\}$
25. $\overline{B} \leftarrow \bigcup \{Y \in M / R' \mid Y \cap B \neq \emptyset\}$
26. Alt $\leftarrow (\alpha(\beta(\underline{A})) \cap \alpha(\overline{B}), \beta(\underline{A}) \cup \beta(\alpha(\overline{B})))$
27. Üst $\leftarrow (\alpha(\beta(\overline{A})) \cup \alpha(\underline{B}), \beta(\overline{A}) \cap \beta(\alpha(\underline{B})))$
28. Cevap \leftarrow Alt ve Üst
29. son eğer

Şekil 4.5 Algoritma 3: bir tanımlanamaz (A, B) kavramına yaklaşım için sözde kod (Saquer ve Deogun, 2001)

5. BELİRTİ-HASTALIK İLİŞKİSİNİN KAVRAM KAFESİ ile MODELLENMESİ ve DENEYSEL ÇALIŞMA

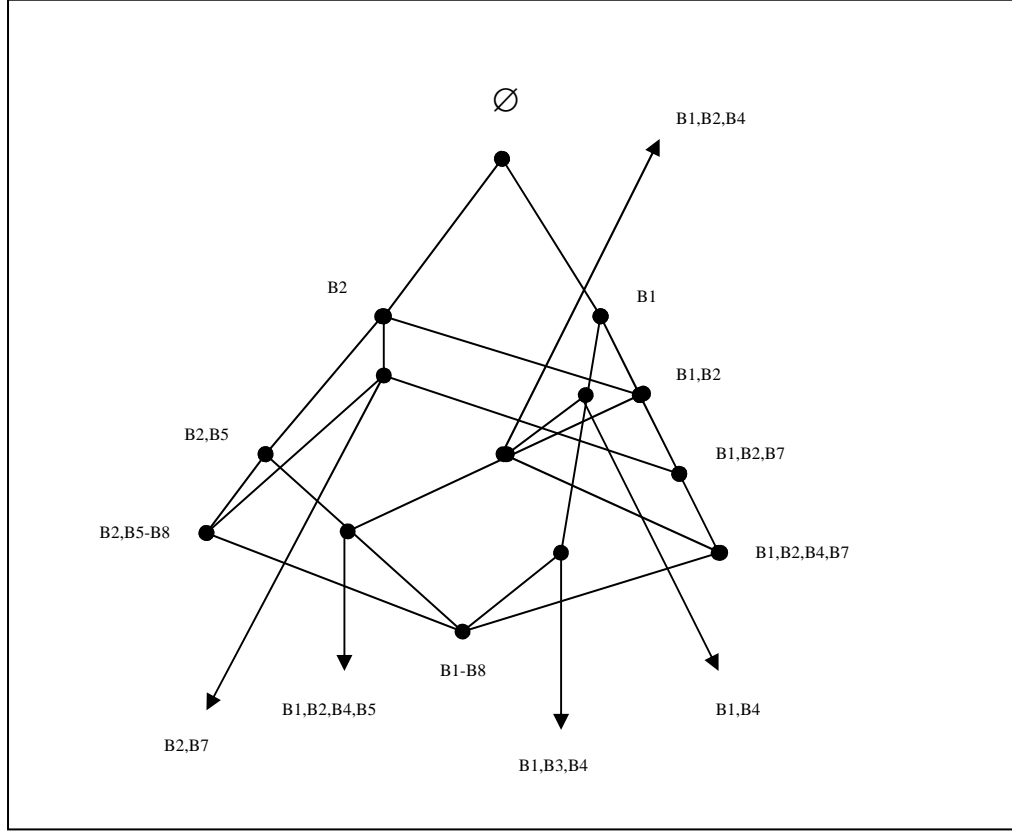
Tıbbi Karar Destek Sistemlerinde sıklıkla kullanılan Bayes ağları ile kafes yapıları çok benzer bir yapıya sahiptir (Ye vd., 2007). Bundan dolayı bu tez çalışmasında TKDS' de Bayes ağları yerine yeni bir yöntem önerilmiştir. Önerilecek model için ilk olarak Belirti-Hastalık ilişkisinin matematiksel olarak modellenmesi gerçekleştirilecek ve daha sonra bu model üretilen sentetik veriler üzerinde denenecektir.

5.1 Belirti-Hastalık İlişkisinin Kavram Kafesi ile Modellenmesi

Bu bölümde ilk olarak Belirti-Hastalık ilişkilerinin verilen bir bağlam içinde kafes yapısı olarak modellenmesi ve daha sonrada bu kafes yapısındaki olasılıksal hesaplamaların gerçekleştirilmesi anlatılacaktır. Burada aşağıdaki çizelgeyi göz önüne alırsak hastalıkları nesnel kümesi $G = \{H_1, H_2, H_3, H_4, H_5, H_6\}$ ve belirtileri de bu nesnelere karşılık gelen özellikler kümesi $M = \{B_1, B_2, B_3, B_4, B_5, B_6, B_7, B_8\}$ olarak düşünülebilir. Biçimsel kavram çözümleme ve kavram kafesi konuları bir önceki bölümde ayrıntılı olarak incelenmiştir.

Çizelge 5.1 Belirti-Hastalık ilişkisi için bağlam

HASTALIKLAR	BELİRTİLER							
	B1	B2	B3	B4	B5	B6	B7	B8
H1	X	X		X				
H2	X		X	X				
H3	X	X		X	X			
H4		X			X	X	X	X
H5	X	X					X	
H6	X	X		X			X	



Şekil 5.1 Belirti-Hastalık bağlamına ait kafes yapısı

Öncelikle hastalıklar ve belirtiler arasında bir $\mathfrak{S}(H, B)$ ilişkisi tanımlayalım. Örneğin, Çizelge 5.1'deki bağlamı dikkate alırsak hastalıklar ve belirtiler arasındaki ilişki $\mathfrak{S}(H, B) = \{\langle H_1, B_1 \rangle, \langle H_1, B_3 \rangle, \langle H_1, B_4 \rangle, \dots, \langle H_6, B_1 \rangle, \langle H_6, B_2 \rangle, \langle H_6, B_4 \rangle, \langle H_6, B_7 \rangle\}$ şeklinde gösterilebilir. Burada $\mathfrak{S}_1(B') = \{\langle B_1, B_2, B_4 \rangle, \langle B_1, B_3, B_4 \rangle, \dots, \langle B_1, B_2, B_4, B_7 \rangle\}$ şeklinde yeni bir ilişki tanımlayalım. Bu biçimde tanımladığımız kümeleri Belirti-Hastalık ilişkisi olarak adlandıralım. Belirti-Hastalık problemi kayıtları 1'ler ve 0'lar dan oluşan bir Boolean ilişkiye dönüştürülebilir. Belirti sayısı n , $\mathfrak{S}_1(B')$ ilişkisindeki kayıt sayısı m olsun. Aşağıda verilen dönüşüm işlemini göz önüne alalım.

$$f : X \in \mathfrak{S}(B) \rightarrow \langle D_1, D_2, \dots, D_n \rangle \text{ öyle ki, } D_i = \{i \in X \Rightarrow 1, i \notin X \Rightarrow 0\}$$

Yukarıda tanımlı f fonksiyonu, n adet özelliğe (niteliğe) sahip ve her bir özelliğin değeri i belirtisi X ' in bir ögesi ise 1 değilse 0 olacak şekilde bir kayıt döndürür. Örneğin, $\mathfrak{S}_1(B')$ ilişkisinde yer alan $\langle B_2, B_5, B_6, B_7, B_8 \rangle$ kaydına f fonksiyonunu uygularsak $f(\langle B_2, B_5, B_6, B_7, B_8 \rangle) = \langle 0, 1, 0, 0, 1, 1, 1, 1 \rangle$ sonucu elde edilir. Sonuç olarak f fonksiyonu $\mathfrak{S}_1(B')$ ilişkisini $\mathfrak{S}_2(D_1, D_2, \dots, D_n)$ ilişkisine çevirir. Elde edilen \mathfrak{S}_2 ilişkisi n adet özellikten ve m adet kayıttan oluşur.

5.1.1 Kavram Kafesinin Oluşturulması

Verilen bir bağlamda kavram kafesi $C = \langle V, E \rangle$ yönsüz çizgesiyle modellenir. Burada, V düğümleri, E kenarları temsil etmektedir. Çizgenin her bir düğümü özellikler (nitelikler) kümesinden oluşmaktadır. $v \in V$ düğümünde yer alan özellikleri döndüren bir $a: V \rightarrow \mathfrak{K}(B)$ fonksiyon olsun. V düğümleri üzerinde J dizin kümesi tanımlayalım. Çizgede bulunan her bir kenar iki düğüm arasındaki ilişkiyi gösterir. $(v_i, v_j) \in E$ ise v_i ve v_j düğümlerinin özellikleri arasında bir R eşleştirmesi mevcuttur. C yönsüz çizgesi üzerinde bir R eşleştirme ilişkisi yansımali, geçişli, karşı simetrik bir sıra verir.

$C = \langle V, E \rangle$ yönsüz çizgesinin kavram kafesini oluşturmak için öncelikle V ve E kümelerinin bulunması gerekir. Kavram kafesi oluşturma algoritması v_i ile v_j düğümü arasında bir kenar varsa ya $a(v_i) \subset a(v_j)$ ya da $a(v_j) \subset a(v_i)$ fikrine dayanır. $C = \langle V, E \rangle$ çizgesinin tam kafes olması özelliği kullanılarak V düğümler kümesini elde etmek için $\mathfrak{S}_1(B')$ ilişkisindeki her bir tutanak için bir düğüm yaratılır ve tam kafes oluşana kadar düğümler eklenir. Çizgedeki düğümlerin bulunması için gerekli Algoritma Şekil 5.2' de verilmiştir.

Algoritma: Çizgedeki düğümlerin $V(C)$ bulunması

Girdiler: $\mathfrak{S}_1(B')$ ilişkisinde yer alan tutanaklar

Çıktılar: $V(C), J$

DUGUMBUL ($\mathfrak{S}_1(B'), V(C), J$) **YORDAMI;**

DISYAPILAR;

$\mathfrak{S}_1(B')$ ilişkisindeki tutanaklar, GİRİŞ

$V(C)$, C çizgesini oluşturan düğümler kümesi, ÇIKIŞ;

J , $V(C)$ üzerinde sınırları 1 ile $V(C)$ kümesinin öge sayısına kadar değişen dizin, ÇIKIŞ

ICYAPILAR;

KOMUTLAR;

1. $\mathfrak{S}_1(B')$ ilişkisindeki tutanakları $V(C)$ kümesine ekle;
2. $\emptyset, \{1, 2, \dots, n\}$ kümelerini $V(C)$ kümesine ekle;
3. $J := |V(C)| + 2$;
4. ONCE YINELE, $V(C)$ KUMESINDE DEĞİŞİKLİK YOKSA ÇIK
5. EGER ($i \neq j$) VE $(a(v_i) \cap a(v_j)) \notin V(C)$ İSE [
6. $V(C) := V(C) \cup a(v_i) \cap a(v_j)$
7. $J := J + 1$;
8.]

DUGUMBUL BİTTİ

Şekil 5.2 Çizgedeki düğümlerin bulunması için algoritma (Oğuz vd., 2000)

DUGUMBUL algoritmasının en iyi durumdaki algoritma karmaşıklığı $O(m^2)$ dir. Algoritmanın en kötü çalışma zamanı karmaşıklığı $O(2^n)$ dir. Yönsüz çizgede düğümler arasındaki $E(C)$ kenarlar kümesini bulmak için $V(C)$ düğümler kümesinde birbiriyle doğrudan ilişkisi olan düğüm çiftlerinin bulunması gerekmektedir. Eğer $a(v_i) \subset a(v_j)$ ve düğümler arasında $a(v_i) \subset a(v_k) \subset a(v_j)$ koşulunu sağlayan bir $v_k \in V(C)$ düğümü yoksa v_i ile v_j arasında bir kenar mevcuttur. Bu kapsama özelliği olarak adlandırılır. Eğer, v_i, v_j kapsama özelliğini taşıyorsa v_i düğümü v_j düğümünü tam anlamıyla kapsar olarak adlandırılır. $E(C)$ ' de yer alan tüm kenarlar, $V(C)$ kapsama özelliğini sağlayan tüm düğüm çiftleridir. Çizgedeki kenarların bulunması için gerekli Algoritma Şekil 5.3'de verilmiştir. KENARBUL algoritmasının, karmaşıklığı $O(h^3)$ dir (Deogun vd., 1998; Oğuz vd., 2000).

Algoritma: Çizgedeki kenarların $E(C)$ bulunması

Girdiler: $V(C)$

Çıktılar: $E(C)$

KENARBUL ($V(C)$) YORDAMI;

DISYAPILAR;

$V(C)$, C çizgesini oluşturan düğümler kümesi, GİRİŞ;

ICYAPILAR;

h , $V(C)$ kümesinin öge sayısı

KOMUTLAR;

1. $h := |V(C)|;$

2. $E(C) := \emptyset;$

3. SAYARAK YINELE ($i:=1, i \leq h, i:=i+1$)

4. SAYARAK YINELE ($j:=1, j \leq h, j:=j+1$)

5. EGER ($i \neq j$) VE $a(v_i) \subset a(v_j)$ İSE [

6. kenarekle=DOGRU;

7. SAYARAK YINELE ($k:=1, kenarekle \text{ VE } k \leq h, k+1$)

8. EGER ($i \neq j$) VE ($k \neq j$) VE $a(v_i) \subset a(v_k) \subset a(v_j)$

9. kenarekle=YANLIS;

10. EGER kenarekle İSE

11. $E(C) := E(C) \cup \{(v_i, v_j)\}$

12.]

KENARBUL BITTI

Şekil 5.3 Çizgedeki kenarların bulunması için algoritma (Oğuz, vd., 2000)

5.1.2 Kafes Yapısındaki Olasılıkların Hesaplanması

Kafes yapısındaki kenarlar, özellikler arasındaki olasılıksal ilişkileri temsil etmektedir. Belirti-Hastalık ilişkisine ait kafes yapısı oluşturulduktan sonra bu yapıdaki olasılıkların hesaplanması için Bayes teoreminden faydalanılacaktır. Örneğin bir kişi hasta veya sağlıklı olsun veya m ($\{H_1, H_2, \dots, H_m\}$) hastalıklarından (nesnelere) birine sahip olsun.

Gösterimin basitliği için H , m olası değerlerinin birini alabilen rasgele bir değişken $\{h_1, h_2, \dots, h_m\}$ olsun. Burada, $H=h_i$ 'nin anlamı hastanın h_i hastalığına sahip olduğu ve $H=h_m$ 'nin anlamı da kişinin hiçbir hastalığa sahip olmadığıdır. Ayrıca, varsayalım ki, n ilişkili özellikler (belirtiler) $\{B_1, B_2, \dots, B_n\}$ olsun. $\{b_1, b_2, \dots, b_t\}$ belirtiler kümesine sahip bir hasta verildiğinde, hastanın h_i hastalığına sahip olduğunun ($H=h_i$) olasılığı Bayes Teoremi kullanılarak aşağıdaki formül ile hesaplanır.

$$p(h_i | b_1, b_2, \dots, b_t) = \frac{p(h_i) \cdot p(b_1, b_2, \dots, b_t | h_i)}{\sum_{b_i} p(h_i) \cdot p(b_1, b_2, \dots, b_t | h_i)} \quad (5.1)$$

Yukarıda verilen (5.1) denklemi ile ilgili olarak aşağıdaki açıklamaları yapalım.

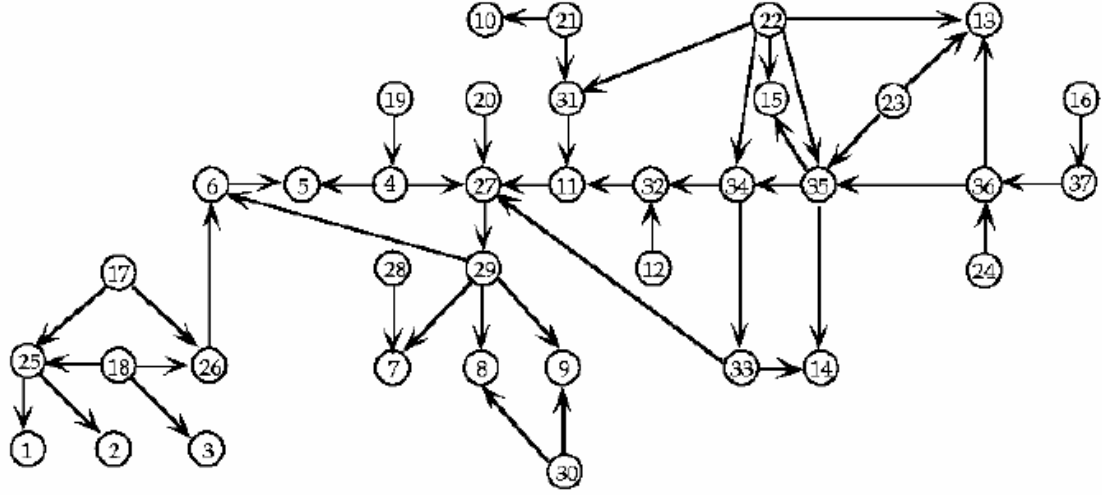
- $p(h_i)$ olasılığı $H=h_i$ hastalığının marjinal, önceki veya başlangıç olasılığı olarak adlandırılır. Çünkü herhangi belirtiyi bilmeden önce elde edilebilir.
- $p(h_i | b_1, b_2, \dots, b_t)$ olasılığı, $H=h_i$ hastalığının sonraki veya koşullu olasılığıdır. Bunun nedeni, $B_1 = b_1, B_2 = b_2, \dots, B_t = b_t$ belirtilerini bildikten sonra hesaplanmasıdır.
- $p(b_1, b_2, \dots, b_t | h_i)$ olasılığı, $H=h_i$ hastalığı ile bir hasta $B_1 = b_1, B_2 = b_2, \dots, B_t = b_t$ belirtilerini gösterecektir. Bu nedenle Bayes teoremi, önceki olasılıkları kullanarak sonraki olasılık dağılımını güncellemede kullanılabilir.

5.2 Veri Kümesi ve ALARM Ağ Yapısı

Doktora tezi kapsamında oluşturulacak modelin test edileceği veri kümesi için bilimsel araştırmalarda kullanılan ve literatürde ALARM (A Logical Alarm Reduction Mechanism) Network (Beinlich, vd., 1989) olarak geçen ağ yapısı kullanılmıştır. ALARM Ağı birçok değişken için gerçek hasta verileri kullanılarak hazırlanmış ve gerçek durumlardan çıkarılmış olasılıkları gösteren bir ağ yapısıdır. ALARM Ağı mevcut kanıtlara dayanarak farklı teşhisler için olasılıkları hesaplamakta ve son yıllarda özellikle Bayes ağlarının yapısını tahmin etmek için birçok araştırmacı tarafından kullanılmaktadır. ALARM Ağında toplam 37 adet düğüm (değişken) bulunmakta ve bunların birbiri ile ilişkileri ve aralarındaki koşullu olasılıklar tanımlanmaktadır. Ayrıca tıbbi bilgi 8 adet teşhis problemini temsil eden 46 adet yay, 16 adet tetkik sonuçları ve tetkik sonuçlarını teşhis problemlerine ilişkilendiren 13 adet ara değişkeni bağlayan grafiksel bir yapıda kodlanmıştır. Bu Bayes ağına iki algoritma uygulanmıştır, bunlardan biri koşullama yöntemini kullanarak çeşitli bağlı ağlardaki olasılık güncellemesi için Pearl tarafından geliştirilen bir mesaj geçiş algoritması ve diğeri de, grafiksel yapıdaki

yerel olasılık hesaplamaları için Lauritzen-Spiegelhalter tarafından geliştirilen tam çıkarım algoritmasıdır. Bu algoritmalar ve Bayes ağları ile ilgili ayrıntılı bilgiler 3. bölümde verilmektedir. ALARM ağında üç tür değişken temsil edilir. **Teşhis** ve nitel bilgi ağın en üst düzeyindedir. Bu değişkenler hiçbir öncele (ataya) sahip değildir ve bunların öncekinden karşılıklı olarak bağımsız oldukları varsayılır. Tüm düğümler belirli bir hastalığın ciddiyetini veya varlığını veya yokluğunu temsil eden karşılıklı özel ve ayrıntılı değerler kümesi ile birleştirilmiştir. **Ölçümler**, herhangi mevcut nicel bilgiyi temsil eder. Tüm sürekli değişkenler değer kümesini bölen ayrık aralıklar kümesi ile kategorik olarak temsil edilir. **Ara değişkenler**, doğrudan ölçülemeyen öğeleri gösterir. Bu Bayes ağındaki olasılıklar öznel olduğu kadar nesnel bilgiyi de temsil edebilir. ALARM Ağı önceki olasılıklardaki istatistiksel veriyi, değişkenler ile ilgili denklemlerden hesaplanan mantıksal koşullu olasılıkları ve belirli sayıda öznel değerlemeleri içerir ve çoğunlukla, sentetik veriler üzerinden ağ yapısının oluşturulması amacıyla kullanılır.

Verilen tüm farklı ata düğümlerin durumlarında, bir düğüm için koşullu olasılık elde etmek gereklidir. ALARM ağının yapısı ve tanımlı değişkenler Şekil 5.4' de verilmiştir.



- | | |
|---|---|
| 1 – central venous pressure | 20 – insufficient anesthesia or analgesia |
| 2 – pulmonary capillary wedge pressure | 21 – pulmonary embolus |
| 3 – history of left ventricular failure | 22 – intubation status |
| 4 – total peripheral resistance | 23 – kinked ventilation tube |
| 5 – blood pressure | 24 – disconnected ventilation tube |
| 6 – cardiac output | 25 – left-ventricular end-diastolic volume |
| 7 – heart rate obtained from blood pressure monitor | 26 – stroke volume |
| 8 – heart rate obtained from electrocardiogram | 27 – catecholamine level |
| 9 – heart rate obtained from oximeter | 28 – error in heart rate reading due to low cardiac output |
| 10 – pulmonary artery pressure | 29 – true heart rate |
| 11 – arterial-blood oxygen saturation | 30 – error in heart rate reading due to electrocautery device |
| 12 – fraction of oxygen in inspired gas | 31 – shunt |
| 13 – ventilation pressure | 32 – pulmonary-artery oxygen saturation |
| 14 – carbon-dioxide content of expired gas | 33 – arterial carbon-dioxide content |
| 15 – minute volume, measured | 34 – alveolar ventilation |
| 16 – minute volume, calculated | 35 – pulmonary ventilation |
| 17 – hypovolemia | 36 – ventilation measured at endotracheal tube |
| 18 – left-ventricular failure | 37 – minute ventilation measured at the ventilator |
| 19 – anaphylaxis | |

Şekil 5.4 ALARM ağ yapısı ve ağda tanımlı değişkenler (Cheng vd., 1997)

Toplam 37 farklı değişkenden oluşan ALARM ağ yapısında 12 belirti (tetikleyici etken), 14 ara değişken ve 11 farklı hastalık bulunmaktadır. Ağ yapısından da anlaşılacağı gibi 12 belirti (tetikleyici etken) şunlardır;

- 12 → Fraction of Oxygen Saturation
- 16 → Minute Volume, Calculated
- 17 → Hypovolemia
- 18 → Left-Ventricular Failure
- 19 → Anaphylaxis
- 20 → Insufficient Anesthesia or Analgesia
- 21 → Pulmonary Embolus
- 22 → Intubation Status

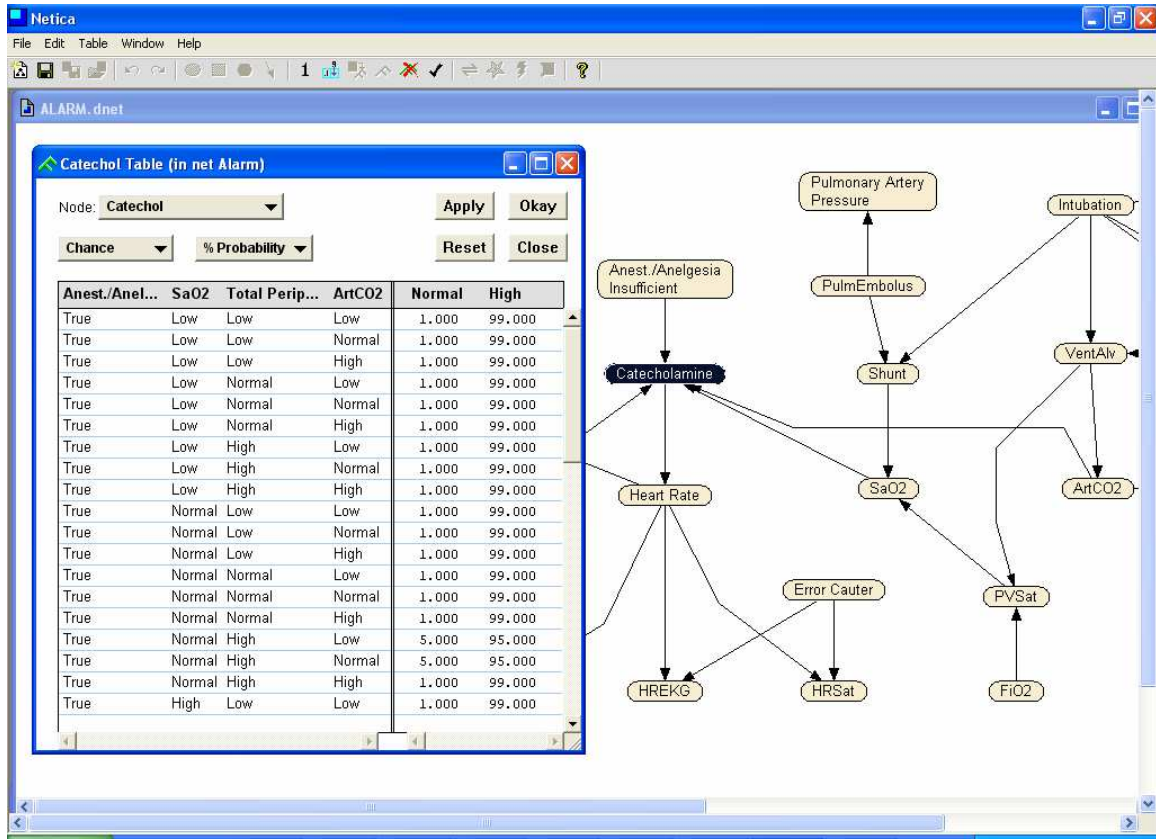
- 23 → Kinked Ventilation Tube
- 24 → Disconnected Ventilation Tube
- 28 → Error In Heart Rate Reading Due To Low Cardiac Output
- 30 → Error In Heart Rate Reading Due To Electrocautery Device

Bu ağda gösterilen ve oluşabilecek 11 farklı hastalıklar aşağıdaki gibidir.

- 1 → CVP (Central Venous Pressure)
- 2 → PCWP (Pulmonary Capillary Wedge Pressure)
- 3 → History (History of left ventricular failure)
- 5 → BP (Blood Pressure)
- 7 → HRBP (Heart Rate obtained from Blood Pressure monitor)
- 8 → HREKG (Heart Rate obtained from Electrocardiogram)
- 9 → HRSat (Heart Rate obtained from oximeter)
- 10 → PAP (Pulmonary Artery Pressure)
- 13 → Press (Ventilation Pressure)
- 14 → ExpCO2 (Carbon-Dioxide content of expired gas)
- 15 → MinVol (Minute Volume, measured)

Deneysel çalışma kapsamında kullanılmak üzere ALARM ağ yapısı üzerindeki olasılıkları yansıtacak biçimde 10, 100, 1000 ve 2000 kayıttan oluşan sentetik test verileri üretilmiştir. Bu işlemler için ALARM ağ yapısı temel alınarak NETICA 3.18 yazılımı kullanılmıştır.. ALARM ağ yapısı ve ağda tanımlı bir değişken için koşullu olasılık Çizelgesi Şekil 5.5' de gösterilmektedir.

Oluşturulan bu veriler üzerindeki her bir kayıt ağ üzerinde tanımlı olan 37 değişkene ait olası değerleri göstermektedir. Her bir kayıta 12 adet giriş ve 11 adet sonuç değişkeni ile beraber ara değişkenler için de ilgili değerler tutulmaktadır. Gerçekleştirilen testler, her bir kayıttaki giriş değişkenlerinin değerlerini ilgili modüle göndermekte ve bu işlem sonunda oluşturulan listeyi ayrı bir dosya olarak saklamaktadır. Ulaşılan sonuçların doğruluğuna, test verilerindeki ilgili kayıtlardaki değişken değerlerinin karşılaştırılması ile karar verilmektedir. Her bir kayıt için 11 olası sonuca ulaşılmıştır.



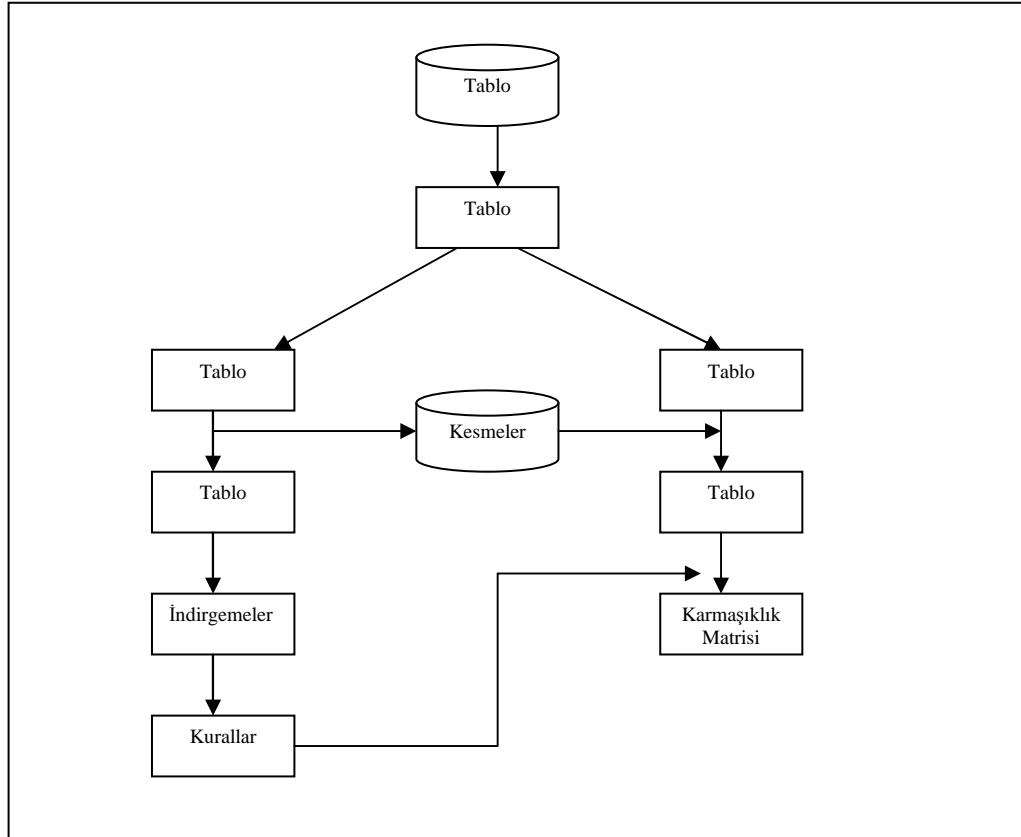
Şekil 5.5 Alarm.dnet Catechol değişkeni için koşullu olasılık çizelgesi

5.3 Özellik Verilerinin İndirgenmesinde Kaba Kümelerin Kullanımı

Üretilen bu veri kümelerindeki özellik (tetikleyici etkenler, giriş değişkenleri) verilerinin indirgenmesi için kaba kümeler kuramının temelini teşkil eden ayırtedilebilirlik kavramına dayalı yazılım aracı olan ROSETTA (A ROUGH SET Toolkit for Analysis Data) (Ohn, 1999) kullanılmıştır. ROSETTA sistemi veri analizine dayalı, mevcut diğer benzer sistemleri içinde en esnek ve ileri düzeyde bir kaba küme yazılım sistemi olduğu için tercih edilmiştir. Bu yazılım kullanılarak tanılar arındırılmış ve özelliklerin indirgenmesi gerçekleştirilmiştir. Böylece gereksiz verilerin kullanılması önlenerek hesaplama karmaşıklığı azaltılmıştır. ROSETTA grafik kullanıcı arabirimi, ağacın düğümleri nesnelere temsil eden bir ağaç yapısı şeklinde temsil edilir. ROSETTA' da ki nesnelere, gösterilen hareket sonuçlarının tablolı veri yapısında veya harekette bir çıktı içeren dosya veya üretilen kurallar kümesi yapısında olabilir. Bu ağaç yapısı aynı nesne üzerindeki farklı hareketleri yönetebilmenin organize edilmiş bir görünümünü sunar bu da sonuçların daha verimli bir şekilde taranması ve nesnelere analiz araçlarının daha kolay uygulanmasını sağlar.

Ayırtedilebilirliğe dayalı kaba kümeler ve yöntemler VM ve VTBK için bilimsel çerçevede büyük bir öneme sahiptir. Bu arabirim, model yapılandırma ve değerlemenin tekrarlanan doğasının tüm yönlerini destekleyen karmaşık araçlar ve çevreler sağlamak ile

zenginleştirilebilir. Bu ihtiyaçlar bağlamında VTBK için kaba kümeleri kullanma süreci sorgulanmakta, araştırılmakta ve kaba kümeler VTBK deneylerine benzer şekilde süreç örüntüleri kurulmaktadır. Hesaplamalı çekirdek ve ROSETTA' nın kullanıcıya elverişli yüzü arasında bir farklılaştırma yapmak kullanışlı olmaktadır. Hesaplamalı çekirdek kaba küme yöntem bilimi ile VTBK için genel bir C++ sınıf kütüphanesidir ve araştırmacılara yeni algoritmaları ve fikirleri deneyebilecekleri ve hızlı bir şekilde makine diline çevirebilen avantajlı bir kod tabanı sunmaktadır. Kullanıcıya elverişli yüz Windows işletim sistemi altında çalışan en gelişmiş teknolojik grafik kullanıcı arayüzüdür. Çekirdek ve kullanıcıya elverişli yüz beraber pratik ayırtehdilebilirlik ile ilgili araştırma ve uygulamalar için oldukça güçlü bir araç oluşturur. ROSETTA çekirdeği, grafik kullanıcı arayüzü kullanıcıya elverişli yüz ile beraber ve kendi başına komut satır programı olarak iki şekilde kullanılabilir. İlki kullanıcı dostu bir çevrede hesaplamalı motora ulaşmayı sağlarken diğeri ise başka bir yerden çağrılan bir hesaplamalı motor olarak ROSETTA' nın kullanılmasına imkân verir. Aşağıdaki Şekil 5.6' da ROSETTA süreci gösterilmektedir (Hui, 2002).



Şekil 5.6 ROSETTA süreci

5.4 Deneysel Çalışma

Minimal veri kümelerinin üretilmesi için kaba kümeler kuramına dayalı yazılım kullanılarak veri indirgeme işlemi gerçekleştirildikten sonra ilgili modelin kurulması için yukarıda verilen algoritmalar ile kafes yapısı oluşturulmuş ve kafes üzerindeki koşullu olasılıklar hesaplanmıştır. Bu işlemler için yazılım geliştirme aracı olarak, yeterli dokümantasyona sahip yönetilebilir olması ve desteklediği veri biçimleri göz önünde bulundurulduğunda Oracle JDeveloper 10.1.3.2.0 seçilmiştir. Oracle JDeveloper uygulaması yazılım geliştiricilerin ihtiyaç duyacakları geniş teknik dokümana ve ayrıntılı kullanıcı yardım dosyalarına sahiptir. Geliştirilen yazılımın üretilen sentetik veri kümelerinin her birine ayrı ayrı uygulanması ile elde edilen sonuçlar aşağıdaki çizelgelerde verilmiştir. Burada, Çizelgelerin her birinde 1 kayıt için 11 çıktı değişkeni gösterilmektedir.

Bu Çizelgelerde 1. sütun değişken adı (hastalık adı), 2. sütun doğruluk derecesini göstermekte ve program tarafından Bayes teoremi ile hesaplanmaktadır, 3. sütun ALARM ağındaki gerçek durumları, 4.sütun programın ürettiği sonucu, 5. Sütun ise gerçek durum ve programın ürettiği sonucun karşılaştırmasını göstermektedir. 5. sütunda gerçek durum ile programın ürettiği sonuç aynı ise POZITIF, farklı ise NEGATIF sonucunu üretecektir. POZITIF' ler doğru teşhisleri NEGATIF' ler ise yanlış teşhisleri göstermektedir.

Çizelge 5.2 10 kayıtlık veri kümesine ait bir çıktı kümesi

Değişken Adı (Hastalık)	Doğruluk Derecesi	Gerçekteki Durum	Programın Ürettiği Sonuç	Gerçek Durum ve Üretilen Sonucun Karşılaştırılması
History	0,9900	False	False	POZITIF
Press	0,2500	Zero	Zero	POZITIF
MinVol	0,2500	Zero	Zero	POZITIF
ExpCO2	0,2618	Zero	High	NEGATIF
PAP	0,9000	Normal	Normal	POZITIF
HRBP	0,4601	Normal	Normal	POZITIF
HREKG	0,5364	Normal	High	NEGATIF
HRSat	0,5364	Normal	High	NEGATIF
CVP	0,7075	Low	Normal	NEGATIF
PCWP	0,6970	Low	Normal	NEGATIF
BP	0,4342	Normal	Low	NEGATIF

Örneğin, Çizelge 5.2' de MinVol değişkeninin program tarafından doğruluk derecesi 0,2500 olarak hesaplanmıştır. Bu değer ağıdaki gerçek durum ile aynı olduğundan doğru teşhise ulaşılmış olmaktadır. Benzer olarak, HREKG değişkeni için, programın ürettiği doğruluk

değeri 0,5364 olarak hesaplanmıştır. Bu değer ise ağdaki gerçek durum ile aynı olmadığından doğru teşhise ulaşılamamıştır. Benzer yorumlar aşağıdaki çizelgeler içinde geçerlidir.

Çizelge 5.3 100 kayıtlık veri kümesine ait bir çıktı kümesi

Değişken Adı (Hastalık)	Doğruluk Derecesi	Gerçekteki Durum	Programın Ürettiği Sonuç	Gerçek Durum ve Üretilen Sonucun Karşılaştırılması
History	0,9900	False	False	POZITIF
Press	0,9412	Normal	Zero	NEGATIF
MinVol	0,9136	Normal	Zero	NEGATIF
ExpCO2	0,9136	Normal	Zero	NEGATIF
PAP	0,9000	Normal	Normal	POZITIF
HRBP	0,8229	High	High	POZITIF
HREKG	0,8229	High	High	POZITIF
HRSat	0,8229	High	High	POZITIF
CVP	0,7075	Normal	Normal	POZITIF
PCWP	0,6970	Normal	Normal	POZITIF
BP	0,4052	Low	Low	POZITIF

Çizelge 5.4 1000 kayıtlık veri kümesine ait bir çıktı kümesi

Değişken Adı (Hastalık)	Doğruluk Derecesi	Gerçekteki Durum	Programın Ürettiği Sonuç	Gerçek Durum ve Üretilen Sonucun Karşılaştırılması
History	0,9900	False	False	POZITIF
Press	0,2500	Zero	Zero	POZITIF
MinVol	0,2500	Zero	Zero	POZITIF
ExpCO2	0,2618	Zero	High	NEGATIF
PAP	0,9000	Normal	Normal	POZITIF
HRBP	0,4601	Normal	Normal	POZITIF
HREKG	0,5364	Normal	High	NEGATIF
HRSat	0,5364	Normal	High	NEGATIF
CVP	0,7075	Low	Normal	NEGATIF
PCWP	0,6970	Low	Normal	NEGATIF
BP	0,4352	Normal	Low	NEGATIF

Çizelge 5.5 2000 kayıtlık veri kümesine ait bir çıktı kümesi

Değişken Adı (Hastalık)	Doğruluk Derecesi	Gerçekteki Durum	Programın Ürettiği Sonuç	Gerçek Durum ve Üretilen Sonucun Karşılaştırılması
History	0,9900	False	False	POZITIF
Press	0,9412	Normal	Zero	NEGATIF
MinVol	0,9136	Normal	Zero	NEGATIF
ExpCO2	0,9136	Normal	Zero	NEGATIF
PAP	0,9000	Normal	Normal	POZITIF
HRBP	0,8229	Normal	High	NEGATIF
HREKG	0,8229	Normal	High	NEGATIF
HRSat	0,8229	Normal	High	NEGATIF
CVP	0,7075	Low	Normal	NEGATIF
PCWP	0,6970	Low	Normal	NEGATIF
BP	0,4052	Low	Low	POZITIF

Tüm veri kümeleri göz önüne alındığında, “Doğruluk Oranı”, doğru teşhis sayısının toplamının, toplam teşhis sayısına oranı şeklinde hesaplanır.

$$\text{Doğruluk Oranı} = \frac{\text{Doğru Teşhis Sayısı}}{\text{Toplam Teşhis Sayısı}}$$

Aşağıdaki çizelgede tüm veri kümelerine ait doğru teşhis sayısı, toplam teşhis sayısı ve hesaplanan doğruluk oranları gösterilmektedir.

Çizelge 5.6 Geliştirilen modele ait test sonuçları

Veri Kümesi	Doğru Teşhis Sayısı	Toplam Teşhis Sayısı	Doğruluk Oranı
10 kayıtlık sentetik veri kümesi	55	110	%50
100 kayıtlık sentetik veri kümesi	576	1100	%52,36
1000 kayıtlık sentetik veri kümesi	5550	11000	%50,46
2000 kayıtlık sentetik veri kümesi	11012	22000	%50

Yukarıdaki deneysel çalışma sonuçları göz önünde bulundurulduğunda 5 veri kümesine ait ortalama doğruluk oranının %50,7 olduğu görülmektedir.

5.4.1 Kafes Tabanlı Modelin Makine Öğrenme Yöntemleriyle Karşılaştırılması

Geliştirilen kafes tabanlı modele ait sonuçlar diğer makine öğrenme yöntemlerinden, C4.5 karar ağacı, DVM ve geriye yayımlı yapay sinir ağı olan ÇKY yöntemleri ile karşılaştırılmıştır. Bunun için WEKA programı kullanılmıştır. Karşılaştırma sonuçları aşağıda çizelge 5.7’ de verilmiştir.

Çizelge 5.7 Kafes tabanlı yöntemin diğer makine öğrenme yöntemleri ile karşılaştırılması

Veri Kümesi (Dataset)	Kafes Tabanlı Yöntem	C4.5 Karar ağacı	DVM (SVM) Destek Vektör Makinesi	ÇKY (MLP) Çok Katmanlı Yayılm
10	%50,00	%62,72	%65,45	%66,36
100	%52,36	%74,36	%75,36	%75,18
1000	%50,46	%78,72	%78,92	%78,04
2000	%50,00	%78,71	%78,75	%78,31
Ortalama Doğruluk Oranları	%50,70	%73,62	%74,62	%74,47

Çizelge 5.7' den de görüleceği gibi en iyi sonucu yaklaşık %75 ortalama doğruluk oranı ile DVM yöntemi vermiştir.

6. SONUÇ ve DEĞERLENDİRME

Sağlık kuruluşları günümüzde bilişim sistemlerinden, yönetim hizmetleri, hastalıkların teşhis edilmesi, hekimlerin hasta ile ilgili vereceği kararların desteklenmesi, hekimlerin ve diğer sağlık personelinin yapacağı işlerde rehberlik, sinyal yorumlama, laboratuvar hizmetleri ve hasta yönetimi gibi çok çeşitli alanlarda faydalanmaktadır. Bireylerin giderek daha bilgili, yüksek beklentili ve hareketli olduğu, kurumların ise giderek uzmanlaştığı bir ortamda, çeşitli basamaklardaki bakım hizmetlerinin bütünleşmiş ve sürekli bir hizmetler bütünü olarak alınabilmesi günümüz toplumlarının öncelikli talepleri arasındadır. Öte yandan, gerek bireylerin daha sağlıklı kılınmaları gerekse hastaların daha iyi tedavi ve bakım görmeleri ancak yetkili sağlık çalışanlarının doğru ve eksiksiz sağlık bilgilerine gereken yerde ve gereken zamanda ulaşmaları ile mümkündür.

Bilgisayar yazılım ve donanım teknolojisindeki hızlı gelişmeler ile birlikte daha fazla veriyi saklama ve bu verileri faydalı bilgiye dönüştürme imkânına sahibiz. Buna paralel olarak, modelleme ile sınıflandırma uygulamalarındaki gelişmeler, Karar Destek Sistem'lerinin tıp alanında da kullanılmasına olanak sağlamıştır. Bu amaçla kullanılan sistemlerin başında ise Tıbbi Karar Destek Sistemleri (TKDS) gelmektedir. Teknolojideki bu gelişmeler TKDS' de büyük bir öneme sahiptir. Bunun nedeni, çok küçük bir bilginin bile hastalığın teşhis edilmesinde sahip olduğu etkidir. Bayes ağları TKDS' de belirsizlik ile baş edilmede sıklıkla kullanılan veri güdümlü bir yöntemdir.

Doktora tezi kapsamında Bayes ağlarının kafes yapısına benzer bir yapıya sahip olmasından yola çıkılarak yeni bir veri güdümlü model önerilmiştir.

Modelin deneneceği veri kümesi için, veri kümesi bulmadaki teknik sıkıntılar nedeniyle gerçek hasta verilerinden oluşturulmuş ve gerçek durumları yansıtan ALARM ağı kullanılarak 10, 100, 1000 ve 2000 kayıttan oluşan sentetik veri kümeleri NETICA yazılımı kullanılarak oluşturulmuştur.

Üretilen sentetik veriler üzerinde hesaplama karmaşıklığının azaltılması amacıyla öncelikle bu veri kümelerindeki özellik değerlerine (belirtilere, giriş değişkenlerine) kaba kümeler kuramının temelini teşkil eden ayırte diledibilirliğe dayalı bir yazılım olan ROSETTA uygulanmıştır. Bu yazılımda özellik verilerinin indirgenmesi için kısa hesaplama zamanı gerektiren Johnson algoritması kullanılmıştır. Böylelikle tanıların rafine edilmesi gerçekleştirilmiş ve gereksiz verilerin kullanılması engellenmiştir.

Geliştirilen modelde, Biçimsel Kavram Çözümleme yöntemi kullanılarak, Belirti-Hastalık ilişkileri, hastalıklar nesnelere ve belirtiler de özellikler olarak kafes yapısı ile modellenmiştir. Kafes yapısı, belirti ve hastalıklar arasındaki olasılıksal ilişkileri yansıtmaktadır. Bu ilişkiler kafes yapısındaki kenarlar ile gösterilmektedir.

Belirti-Hastalık ilişkisinin bağlamından kafes yapısının oluşturulması ve bu yapıdaki koşullu olasılıkların hesaplanması Oracle JDeveloper yazılımı ile gerçekleştirilmiştir. Kafes yapısındaki koşullu olasılıklar doğruluk derecesini göstermektedir. Üretilen bu doğruluk dereceleri göz önüne alınarak ALARM ağındaki gerçek durum ve program tarafından üretilen sonuçlar karşılaştırılmıştır. Eğer model tarafından üretilen değer ile gerçekteki durum birbirine yakın ise doğru teşhise ulaşılmaktadır.

Geliştirilen model, veri güdümlü bir modeldir ve farklı boyutlarda veri kümelerine uygulandığında yaklaşık ortalama %51 doğruluk oranı ile doğru teşhise ulaşılmıştır.

Ayrıca bu verileri makine öğrenme yöntemlerinden C4.5 karar ağacı, destek vektör makinesi ve çok katmanlı yayılım ağına uyguladığımızda en iyi sonucun yaklaşık ortalama %75 ile destek vektör makinesi yöntemine ait olduğu görülmüştür.

Veri güdümlü sistemlerde kaba kümeler kuramının dışında ayrıca, genetik algoritmalar ve tümevarımsal mantık programlama, vs. çok çeşitli algoritmalar kullanılmaktadır. Bu algoritmaların her birinin birbirlerine göre üstün yönleri bulunmaktadır. Dolayısıyla bu çalışmada önerilen modelin doğruluk oranının yükseltilmesi amacıyla diğer yöntemler ile birleştirilmesi ve melez bir yöntem oluşturulması mümkündür.

Bu çalışma ayrıca, tanecikli hesaplamanın uygulama alanlarından birisi olan kaba kümeler kuramı ile bilginin temsil edilmesinde güçlü bir yöntem olan biçimsel kavram çözümleme yönteminin tıbbi karar destek sistemlerinin tasarımında oynayacağı rolün önemini göstermiştir.

Ayrıca, bu tez çalışması kapsamında kaba kümeler kuramının Bayes ağları ve akış çizgeleri ile biçimsel kavram çözümleme konuları arasındaki kuramsal ilişkileri de ayrıntılı bir şekilde incelenmiştir.

KAYNAKLAR

- Aktaş, E., Ülengin, F., (2004), "Sağlık Hizmetleri Yönetiminde bir Bayes Ağları Uygulaması", XXIV Ulusal YA/EM 2004 Kongresi, Gaziantep-Adana.
- Allam, A.A., Bakeir, M.Y., Abo-Tabl, E.A., (2005), "New Approach for Basic Rough Set Concepts", International Conference on Rough Sets, Fuzzy Sets, Data Mining, Granular Computing, 64-73.
- Alonso, J.A., Borrego, J., Hidalgo, M.J., Martin-Mateos, F.J., Ruiz-Reina, J.L., (2004), "Verification of the Formal Concept Analysis", RACSAM, Vol.98(1).
- Becker A., Geiger, D., (1996), "Optimization of Pearl's Method of Conditioning and Greedy Like Approximation Algorithms for the Vertex Feedback Set Problem", Artificial Intelligence, Vol.83, 167-188.
- Beinlich, I., Suermondt, H.J., Chavez, R.M., Cooper, G.F., (1989), "The ALARM Monitoring System: A Case Study with Two Probabilistic Inference Techniques for Belief Networks", In Proceedings of the 2nd European Conference on Artificial Intelligence in Medicine, 38, 247-256.
- Bender, E. A., (2000), "Mathematical Methods in Artificial Intelligence", IEEE Computer Society Press, California.
- Binay, H.S., (2002), "Yatırım Kararlarında Kaba Küme Yaklaşımı", Ankara Üniversitesi, Doktora Tezi, Ankara (yayımlanmamış).
- Blockeel, H., De Raedt, L., (1997), "Lookahead and Discretization in ILP", In Proceedings of Inductive Logic Programming 1997, 77-85.
- Castillo E., Gutierrez, J.M, Hadi, A.S., (1997), "Expert Systems and Probabilistic Network Models", Springer Verlag, New York.
- Cooper, G.F., (1990), "The Computational Complexity of Probabilistic Inference Using Bayesian Belief Networks", Artificial Intelligence, Vol.42, 393-405.
- Cortes, C., Vapnik, V.N., (1995), "Support Vector Networks", Machine Learning, vol.20, 173-297.
- Deogun J.S., Raghavan V.V., Sever, H., (1998), "Association Mining and Formal Concept Analysis", In Proceedings Sixth International Workshop on Rough Sets, Data Mining and Granular Computing, 335-338.
- Deogun, J.S., Saquer, J.M., (2004), "Monotone Concepts for Formal Concept Analysis", Discrete Applied Mathematics, Vol.144, 70-78.
- Dubois, D., Prade, H. (1992), "Putting Rough Sets and Fuzzy Sets Together", In: Slowinski, R. (Ed.) Intelligent Decision Support System, Handbook of Applications and Advances of the Rough Set Theory, Kluwer Academic Publishers, 203-232.
- Gaag Van der, L.C., Bodlaender H.L., (1997), "Comparing Loop Cutsets and Clique Trees in Probabilistic Inference", Technical Report.
- Ganter W., Wille R., (1999), "Formal Concept Analysis: Mathematical Foundations" Springer Verlag, Berlin.
- Greco, S., Pawlak, Z., Slowinski, R., (2004), "Can Bayesian Confirmation Measures be Useful for Rough Set Decision Rules?", Engineering Applications of Artificial Intelligence, 17, 345-361.

- Guo, H., Hsu, W., (2002), "A Survey of Algorithms for Real-Time Bayesian Network Inference", In: In the Joint AAAI-02/KDD-02/UAI-02 Workshop on Real-Time Decision Support and Diagnosis Systems, Edmonton, Alberta, Canada.
- Han, J., Cai, Y, Cercone, N., (1993), "Data-Driven Discovery of Quantitative Rules in Data Bases", IEEE Transactions on Knowledge and Data Engineering, Vol.5, 29-40.
- Hansen, A.,F.,(2006), "Bayesian Networks as a Decision Support Tool in Marine Applications", Ph. D. Thesis, Denmark.
- Hirota, K., Pedrycz, W., (1999), "Fuzzy Computing for Data Mining", Proceedings of the IEEE, Vol. 87, 1575–1600.
- Hornik, K., Stinchcombe M., White, H., (1989), "Multilayer Feedforward Networks are Universal Approximators", Neural Networks, 359-366.
- Hui, S., (2002), "Rough Set Classification of Gene Expression Data", Bioinformatics Group Project, 2002.
- Jensen, F.V., (1996), "An Introduction to Bayesian Networks", UCL Press, London.
- Johnson, S.D., (1974), "Approximation Algorithms for Combinatorial Problems", Journal of Computer and System Sciences, Vol.9, 256-278.
- Kent, R., (1994), "Rough Concept Analysis", Proceedings International Workshop Rough Sets and Knowledge Discovery", Banff, Canada, 245-253.
- Kent, R., (1996), "Rough Concept Analysis: A Synthesis of Rough Sets and Formal Concept Analysis", Fundamenta Informaticae, Vol.27, No.2, 169-181.
- Klir, G.J., (1998), "Basic Issues of Computing with Granular Probabilities", Proceedings of 1998 IEEE International Conference on Fuzzy Systems, 101-105, 1998.
- Komorowski, J., Polkowski, L., Skowron, A. (1998), "Rough Sets: A tutorial", Rough-Fuzzy Hybridization: A new Method for Decision Making, Spriger Verlag.
- Komorowski, J., Pawlak, Z., Polkowski, L., Skowron, A., (1999), "A Rough Set Perspective on Data and Knowledge".
- Krusinska, E., Slowinski, R., Stefanowski, J., (1992), "Discriminant Versus Rough Set Approach to Vague Data Analysis", Journal of Applied Statistics and Data Analysis, 8, 43-56.
- Lauritzen, S.L., Spiegelhalter, D.J., (1988), "Local Computations with Probabilities on Graphical Structures and Their Applications to Expert Systems, Proceedings of the Royal Statistical Society, 154-227.
- Li, Y.F., Zhong, N., (2003), "Interpretions of Association Rules by Granular Computing", Proceedings of 2003 IEEE International Conference on Data Mining, 593-596.
- Lin T.Y., (1997), "Granular Computing", Announcement of the BISC Special Interest Group on Granular Computing.
- Lin, T.Y., (1998), "Granular Computing on Binary Relations I: Data Mining and Neighborhood Systems, II: Rough Sets Representations and Belief Functions", In: Rough Sets in Knowledge Discovery I, Physica Verlag, 107-140.
- Lin T.Y., (1999), "Data Mining: Granular Computing Approach Approach", Methodologies for Knowledge Discovery and Data Mining Proceedings of Pacific Asia Knowledge Discovery and Data Mining 1999, 24-33.

- Lin T.Y., (2000), "Data Mining and Machine Oriented Modeling: A Granular Computing Approach", *Journal of Applied Intelligence*, Vol.13, 113-124.
- Lin, Y., Druzdzel, M.J., (1999), "Stochastic Sampling and Search in Belief Updating Algorithms for Very Large Bayesian Networks", In *Working Notes of the AAAI Spring Symposium on Search Techniques for Problem Solving Under Uncertainty and Incomplete Information*, 77-82.
- Milho, I., Fred, A., (2000), "A User Friendly Development Tool for Medical Diagnosis Based on Bayesian Networks", *Proceedings of the Second International Conference on Enterprise Information Systems, ICEIS 2000, Stafford*.
- Mitchell, T.M., (1982), "Generalization as Search", *Artificial Intelligence*, Vol.18, 203-226.
- Mitra, S., Pal, S.K., Mitra, P. (2002), "Data Mining in Software Computing Framework: A Survey", *IEEE Transactions on Neural Network*, Vol.13, 3-14.
- Muggleton, S., (1990), "Efficient Induction of Logic Programs", In *Proceedings of Algorithmic Learning Theory*, 368-381.
- Muggleton, S., (1995), "Inverse Entailment and Progol", *New Generation Computing* 13, 245-286.
- Nguyen, H.S., Slezak, D., (1999), "Approximate Reducts and Association Rules Correspondence and Complexity Results", *Proceedings of the 7th International Workshop on New Directions in Rough Sets, Data Mining and Granular Computing*, Vol.1711, LNCS, Springer Verlag, 137-145.
- Oğuz, B., Sever, H., Tolun, M., (2000), "Eşleştirme Sorgularının Modellenmesi", *The Ninth Turkish Symposium on Artificial Intelligence and Neural Networks 2000, İzmir*.
- Ohrn, A.,(1999), "Discernibility and Rough Sets in Medicine, Tools and Applications", *Ph.D. Thesis, Norway*,
- Pal, S., Mitra, P., (2004), "Pattern Recognition Algorithms for Data Mining", *Chapman Hall, Boca Raton*.
- Pawlak, Z., (1982), "Rough Sets", *International Journal of Computer and Information Sciences* 11, 341-356.
- Pawlak, Z. (1994a), "Rough Sets Present State and Further Prospects", In the *Third International Workshop on Rough Set and Soft Computing (RSSC'94)*, 72–76.
- Pawlak, Z., (1994b), "An Inquiry into Vagueness and Uncertainty".
- Pawlak, Z., (1994c), "Vagueness and Uncertainty: A Rough Set Perspective", *Journal of Computational Intelligence*, Vol.11, No.2, 227-232.
- Pawlak, Z., Skowron, A., (1994), "Rough Membership Functions", *Advances in the Dempster Shafer Theory of Evidence*, John Wiley and Sons, 251-271.
- Pawlak, Z., Grzymala Busse, J., Slowinski, R., Ziarko, W., (1995), "Rough Set Approach to Knowledge Based Decision Support", *Journal of Communications of the ACM*, Vol 38, No 11, 88-95.
- Pawlak, Z., (1996), "Rough Sets, Rough Relations and Rough Functions", *Fundamenta Informaticae*, Vol 27, No:2/3, 103-108.
- Pawlak, Z., (1998), "Granularity of Knowledge, Indiscernibility and Rough Sets", *Proceedings of 1998 IEEE International Conference on Fuzzy Systems*, 106-110.

- Pawlak, Z., (2002a), "Rough Sets and Intelligent Data Analysis", Information Sciences, Vol.147, No.1, 1-12.
- Pawlak, Z., (2002b), "Rough Set Theory", Journal of Telecommunications and Information Technology, Vol.3, 7-10.
- Pawlak, Z., (2003), "Decision Algorithms and Flow Graphs: A Rough Set Approach", Journal of Telecommunications and Information Technology, Vol.3, 98-101.
- Pawlak, Z., (2004), "Data Analysis and Flow Graphs", Journal of Telecommunications and Information Technology, Vol.3, 1-5.
- Pearl, J., (1986a), "A Constraint-Propagation Approach to Probabilistic Reasoning", In: Kanal, L.N., Lemmer, J.F. (Eds.), Uncertainty in Artificial Intelligence, 371-382.
- Pearl, J., (1986b), "Fusion, Propagation and Structuring in Belief Networks", UCLA Computer Science Department Technical Report , Artificial Intelligence Vol.29, No.3.
- Pearl, J., (1988), "Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference", Morgan Kaufmann Publishers, California.
- Peters, J.F., Pawlak, Z., Skowron, A., (2002), "A Rough Set Approach to Measuring Information Granules", Proceedings of COMPSAC 2002, 1135-1139.
- Polkowski, L., Skowron, A., (1998), "Towards Adaptive Calculus of Granules", Proceedings of 1998 IEEE International Conference on Fuzzy Systems, 111-116.
- Quinlan, J., (1990), "Learning Logical Definitions from Relations", Machine Learning, 239-266.
- Quinlan, J., (1993), "C4.5: Programs for Machine Learning", Morgan Kaufmann, San Mateo, California.
- Russell, S., Norvig, P., (2003), "Artificial Intelligence A Modern Approach", Second Edition, Pearson Education, New Jersey.
- Saquer, J.M., Deogun, J.S., (1999), "Formal Rough Concept Analysis" New Directions in Rough Sets, Data Mining and Granular-Soft Computing, Yamaguchi, Japan, Springer Verlag, 91-99.
- Saquer, J.M., (2000), "Formal Concept Analysis and Applications", Ph. D. Thesis, Canada.
- Saquer, J.M., Deogun, J.S., (2001), "Concept Approximations Based on Rough Sets and Similarity Measures", International Journal App. Math. Computer Science, Vol.11, No.3, 2001, 655-674.
- Sever, H., Oğuz, B., (2003), "Veri Tabanlarında Bilgi Keşfine Formel Bir Yaklaşım: Kısım II- Eşleştirme Sorgularının Biçimsel Kavram Analizi ile Modellenmesi", Bilgi Dünyası, 15-44.
- Skowron, A. ve Rauszer, C. (1992), "The Discernibility Matrices and Functions in Information Systems", Intelligent Decision Support Handbook of Advances and Applications of the Rough Set Theory. Kluwer Academic Publishers, 311-362.
- Skowron, A., Grzymala-Busse, J., (1994), "From the Rough Set Theory to Evidence Theory", Advances in the Dempster Shafer Theory of Evidence, John Wiley and Sons, 193-235.
- Skowron, A., Stepaniuk, J., (1998), "Information Granules and Approximation Spaces", Manuscript, 1998.

- Skowron, A., Stepaniuk, J., (1999), "Towards Discovery of Information Granules", Proceedings of Pacific Asia Knowledge Discovery and Data Mining 1999, 542-547.
- Slezak, D., Ziarko, W., (2002), "Bayesian Rough Set Model", Proceedings of the International Workshop on Foundation of Data Mining, Maebashi, Japan, 131-135.
- Slezak, D., Ziarko, W., (2003a), "Attribute Reduction in the Bayesian Version of Variable Precision Rough Set Model", Proceedings of the International Workshop on Rough Sets and Knowledge Discovery. Elsevier, ENTCS, Vol.82, No.4.
- Slezak, D., Ziarko, W., (2003b), "Variable Precision Bayesian Rough Set Model", 9 th International Conference on Rough Sets, Fuzzy Sets, Data Mining, Granular Computing 2003, Chongqing, China, 312-315.
- Slezak, D., Ziarko, W., (2005), "The Investigation of the Bayesian Rough Set Model", International Journal of Approximate Reasoning, 81-91.
- Suermondt H.J., Cooper, G.F., (1990), "A Combination of Exact Algorithms for Inference on Bayesian Belief Networks", International Journal of Approximate Reasoning, Vol.4, 283-306.
- Tam, G.K., (2004), "FOCAS- Formal Concept Analysis and Text Similarity", B.Sc. Thesis, Australia.
- Tsumoto, S., (1998), "Modelling Medical Diagnostic Rules Based on Rough Sets", Rough Sets and Current Trends in Computing, LNAI 1424, 475-482.
- Tsumoto, S., (1999), "Automated Discovery of Plausible Rules Based on Rough Sets and Rough Inclusion", Proceedings of International Conference on Pacific Asia Knowledge Discovery and Data Mining 1999, LNAI 1574, 210-219.
- Tsumoto, S., Lin, T.Y., Peters, J.F., (2002), "Foundations of Data Mining via Granular and Rough Computing", Proceedings of the 26th Annual International Computer Software and Applications Conference 2002, 1123-1125.
- Vapnik, V.N., (1995), "The Nature of Statistical Learning Theory", Springer Verlag, New York.
- Vapnik, V.N., (1998), "Statistical Learning Theory", John Wiley&Sons, New York.
- Wille G., (1982), "Restructuring Lattice Theory: An Approach Based on Hierarchies on Concepts", In: Ordered Sets, Dordrecht, 445-470.
- Wong, S.K.M., Ziarko, W., Ye, R.L., (1986), "Comparison of Rough Set and Statistical Methods in Inductive Learning", International Journal of Man-Machine Studies, 24, 53-72.
- Yao, J.T., (2005), "Information Granulation and Granular Relationship", Proceedings of the IEEE Conference on Granular Computing", 326-329.
- Yao, Y.Y., Zhong, N., (1999), "Potential Applications of Granular Computing in Knowledge Discovery and Data Mining", Proceedings of World Multiconference on Systemics, Cybernetics, Informatics, Vol.5, Computer Science and Engineering, 573-580.
- Yao, Y.Y., (1999a), "Rough Sets, Neighborhood Systems and Granular Computing", Proceedings of the 1999 IEEE Canadian Conference on Electrical and Computer Engineering. IEEE Pres, 1553-1558.
- Yao, Y.Y., (1999b), "Granular Computing Using Neighborhood Systems", In: Advances in Soft Computing: Engineering and Manufacturing", Springer Verlag, 539-553.

- Yao, Y.Y., (1999c), "Stratified Rough Sets and Granular Computing", Proceedings of the 18 th International Conference of the North American Fuzzy Information Processing Society, IEEE Pres, 800-804.
- Yao, Y.Y., (2001), "Information Granulation and Rough Set Approximation", International Journal of Intelligent Systems, Vol.16, No.1, 87-104.
- Yao, Y.Y., Liao, C.J., (2002), "A Generalized Decision Logic Language for Granular Computing", Proceedins of FUZZ-IEEE'02, 1092-1097.
- Yao, Y.Y., (2003), "Information Granulation and Approximation in a Decision-Theoretical Model of Rough Sets", Rough-Neural Computing: Techniques for Computing Words, SpringerVerlag, Berlin, 491-518.
- Yao, Y.Y., (2004a), "Concept Formation and Learning: A Cognitive Informatics Perspective", Proceedings of the Third IEEE International Conference on Cognitive Informatics, 42-51.
- Yao, Y.Y., (2004b), "A Partiton Model of Granular Computing", Lecture Notes in Computer Science Transactions on Rough Sets, Vol.1, 232-253.
- Yao, Y.Y., (2004c), "A Comparative Study of Formal Concept Analysis and Rough Set Theory in Data Analysis", Rough Sets and Current Trends in Computing 4th International Conference 2004, 59-68.
- Yao, Y.Y., (2004d), "Concept Lattices in Rough Set Theory", Proceedings of 2004 Annual Meeting of the North American Fuzzy Information Processing Society, 796-801.
- Yao, Y.Y., (2006), "Granular Computing for Data Mining", Proceedings of SPIE Conference on Data Mining, Intrusion Detection, Information Assurance and Data Networks Security, 1-12.
- Ye, J., Coyle, L, Dobson, S., Nixon, P., (2007), "Using Situation Lattices to Model and Reason About Context", Proceedings of the Fourth International Workshop Modeling and Reasoning in Context 2007, Denmark,
- Yücebaş, S.C., (2006), "Hipokrat-I: Bayes Ağı Tabanlı Tıbbi Teşhis Destek Sistemi", Başkent Üniversitesi, YL Tezi, Ankara, (Yayımlanmamış)
- Zadeh, L.A., (1979), "Fuzzy Sets and Information Granulation", Advances in Fuzzy Set Theory and Applications, North Holland, 3-18.
- Zadeh, L.A., (1997), "Towards a Theory of Fuzzy Information Granulation and its Centrality in Human Reasoning and Fuzzy Logic", Fuzzy Sets and Systems, Vol.19, 111-117.
- Zhang, C., Zhang, S., (2002), "Association Rule Mining", Springer Verlag, Berlin.
- Zhang, G.Q., (2004), "Chu Spaces Concept Lattices and Domains", Electronic Notes in Theoretical Computer Science.
- Zhang, G.Q., Shen, G., Tian, Y., Sun, J., (2005), "Concept Analysis as a Formal Method for Menu Design", 12th International Workshop on Design, Specification, and Verification of Interactive Systems, England, 2005, 173-187.
- Zhang, M., Xu, L.D., Zhang, W.X., Li, H.Z., (2003), "A Rough Set Approach to Knowledge Reduction Based on Inclusion Degree and Evidence Reasoning Theory", Expert Systems, Vol.20, 298-304.
- Zhang, Y.Q., Fraser, M.D., Gagliano, R.A., Kandel, A., (2000), "Granular Neural Networks for Numerical-Linguistic Data Fusion and Knowledge Discovery", IEEE

Transactions on Neural Networks, Vol.11, 658-667.

Zhong, N., (2000), "Multi-Database Mining: A Granular Computing Approach", Proceedings of the Fifth Joint Conference on Information Sciences 2000, 198-201.

Ziarko, W., (1993), "Variable Precision Rough Set Model", Journal of Computer and System Sciences, 40(1), 39-59.

INTERNET KAYNAKLARI

[1] <http://www.cs.ualberta.ca/~jcheng/Doc/report98.pdf>

EKLER

Ek1

“International Conference on Intelligent Systems-ICIS” konferansında 2007’de yayınlanan “Modeling the Symptom-Disease Relationship by Using Rough Set Theory and Formal Concept Analysis” isimli bildirinin özeti

Ek2

“Bilimde Modern Yöntemler Sempozyumu-BMYS” sempozyumunda 2005’de yayınlanan “Veri Madenciliğinde Birlikte Kuralları” isimli bildirinin özeti

Ek 1 Modeling the Symptom-Disease Relationship by Using Rough Set Theory and Formal Concept Analysis

Abstract. Medical Decision Support Systems (MDSSs) are sophisticated, intelligent systems that can provide inference due to lack of information and uncertainty. In such systems, to model the uncertainty various soft computing methods such as Bayesian networks, rough sets, artificial neural networks, fuzzy logic, inductive logic programming and genetic algorithms and hybrid methods that formed from the combination of the few mentioned methods are used. In this study, symptom-disease relationships are presented by a framework which is modeled with a formal concept analysis and theory, as diseases, objects and attributes of symptoms. After a concept lattice is formed, Bayes theorem can be used to determine the relationships between attributes and objects. A discernibility relation that forms the base of the rough sets can be applied to attribute data sets in order to reduce attributes and decrease the complexity of computation.

Ek 2 Veri Madenciliğinde Birliktelik Kuralları

Özet. Bilginin özgün bir formu olarak birliktelikler, veritabanlarındaki kelimelerin aralarındaki ilişkilerine yansımaktadır. Ayrıca birliktelik kuralları veritabanlarından bilgi keşfi ve veri madenciliği alanlarında da geniş çapta araştırmalara konu olmaktadır. Veri madenciliği, büyük miktarda veri içinden gelecek ile ilgili tahmin yapmamızı sağlayacak bağıntı ve kuralların bilgisayar programları kullanılarak aranmasıdır. Veri tabanlarından bilgi keşfi ise, değişik örüntülerin anlamlı, kısa ve özgün bir şekilde tanımlanması ile ilgilidir.

Bu çalışmada, birliktelik kuralları madenciliğinin genel tanımları ve uygulamalarına değinilmiş, bununla ilgili olarak bir örnek verilmiş ve çok bilinen apriori algoritması anlatılmıştır.

ÖZGEÇMİŞ

Doğum tarihi	04.04.1974	
Doğum yeri	İstanbul	
Lise	1988-1991	Şehremini Lisesi
Lisans	1994-1999	Yıldız Teknik Üniversitesi, Kimya-Metalurji Fakültesi, Matematik Mühendisliği Bölümü
Yüksek Lisans	1999-2001	Yıldız Teknik Üniversitesi, Fen Bilimleri Enstitüsü, Matematik Mühendisliği Anabilim Dalı
Doktora	2001-	Yıldız Teknik Üniversitesi, Fen Bilimleri Enstitüsü, Matematik Mühendisliği Anabilim Dalı

Çalıştığı kurum(lar)

1999-Devam ediyor YTÜ, Kimya-Metalurji Fakültesi,
Matematik Müh. Bölümü'nde Araştırma Görevlisi