

**STANDART VERİTABANI SİSTEMLERİNDE BULANIK KÜME
YAKLAŞIMI İLE ESNEK SORGULAMA**

YÜKSEK LİSANS TEZİ

Bahadır ASAR

509950070011

98429

Tezin Enstitüye Verildiği Tarih : 31 Mayıs 1999

Tezin savunulduğu Tarih : 18 Haziran 1999

Tez Danışmanı : Doç. Dr. Gazanfer ÜNAL

Diğer Jüri Üyeleri : Prof.Dr.Zekai ŞEN

Prof.Dr.Galip TEPEHAN



HAZİRAN 1999

ÖNSÖZ

Günümüzde verinin en iyi şekilde kullanımı ve yararlı bilgiye dönüştürülmesi büyük önem taşımaktadır. Bu tez çalışması, bu konuya bulanık küme teorisi yaklaşımı ile yeni bir uygulama oluşturmuştur. İlerde bu konu ile ilgili çalışmalara yardımcı olması umulmaktadır.

Bu tezin hazırlanması sırasında yardım ve güveninden ötürü Sayın Doç. Dr. Gazanfer ÜNAL'a, destek ve sabrından ötürü eşim Hülya ASAR'a teşekkür ederim.



İÇİNDEKİLER

ÖZET	IV
SUMMARY	V
1. GİRİŞ	1
1.1. Bulanıklık Kavramının Tanımı	1
1.2. Bulanık Sorgu Aracı	2
2. SIRADAN VERİTABANI SİSTEMLERİNDE BULANIK KÜME YAKLAŞIMI İLE ESNEK SORGULAMA	4
2.1. Giriş	4
2.2. Esnek Sorgulama Sistemlerinde Bulanık Küme Yaklaşımı	5
2.3. SummarySQL	9
2.4. Sonuç	13
3. BULANIK SORGU UYGULAMASI	14
3.1. Giriş	14
3.2. Bulanık Sorgulama	14
3.3. Bulanık Sorgu Tipleri	15
3.4. Programın Genel Yapısı ve Akışı	15
3.5. Bulanıklaştırma	18
3.5.1. Bulanık kümelerin tanımı	18
3.5.2. Bulanık kümelerin üyelik fonksiyonunun belirlenmesi	18
3.6. Sorgu Seçeneklerinin Belirlenmesi	21
3.7. Sorgu Cümlesinin Alınması	21
3.8. Sorgu Analizi	22
3.9. Sorgu İşlemcisi	22
3.10. Sorgu Cümlesinin Yapısı	23
3.11. Sorgu Cümlesinin Analizi ve Gerekli Verilerin Derlenmesi	25
3.11.1. BulanikOprt	26
3.11.2. BulOptHesap	26
3.11.3. BulanikKümeBul	27
3.11.4. BkümeAra	27
3.12. Sorgu İşlemcisi ve Eşleşme Derecelerinin Bulunması	28
3.12.1. Enterpolasyon	28
3.12.2. Fonksiyonların Belirlenmesi	31
3.13. Sorgu İşleminin Sonuçlandırılması	32
SONUÇLAR VE ÖNERİLER	33
KAYNAKLAR	34
EKLER	35

STANDART VERİTABANI SİSTEMLERİNDE BULANIK KÜME YAKLAŞIMI İLE ESNEK SORGULAMA

ÖZET

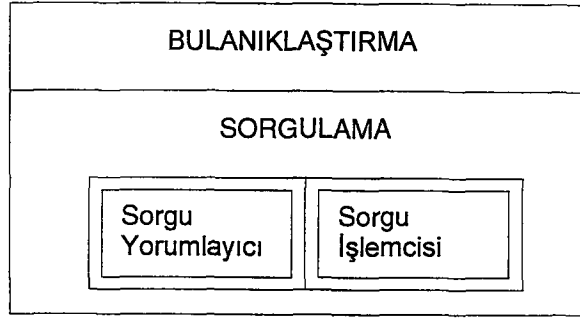
Bilgisayar kullanımının, iş dünyasında ve akademik çalışmalarda artması ile potansiyel olarak değerli bilgi içeren veri yığınları oluşmuştur. Bu bilginin oluşturulması için, esnek sorgulama ve akıllı arama yapabilen yeni nesil gelişmiş araçlar geliştirmemiz gerekmektedir. Sürekli gelişmekte olan bu konun kapandığı alanlardan ikisi Veri Madenciliği (Data Mining) ve Bilgi Keşfi (Knowledge Discovery) dir.

Bu tez çalışmasında, bulanık olmayan veritabanı sistemleri üzerinde bulanık sorgulama yapabilen bir araç geliştirilmiştir. Bu aracın geliştirilmesinde Microsoft Visual FoxPro(VFP)¹ kullanılmıştır. VFP programlanabilir bir veritabanı yönetim sistemidir ve veritabanlarına yönelik programlama için çok kullanışlıdır.

Sistem Şekil.1 de gösterildiği gibi iki ana bölümden oluşmaktadır, bulanıklaştırma ve sorgulama. Öncelikle, kullanıcı üzerinde sorgulama yapılacak veritabanının alanları için bir kereye mahsus olmak üzere bulanıklaştırma işlemi yapar. Daha sonra yapılacak sorgulama burada tanımlanan bulanık kümeler yardımı ile yapılmaktadır. Bulanıklaştırma işlemi sırasında bulanık küme bilgileri bir veritabanında saklanır.

Geleneksel bir sorgulama sisteminde, başarılı öğrencileri bulmak için “Ortalaması büyüktür eşittir 70 ve devamlılığı 10 dan büyük olan öğrencilerin listesi” şeklinde ifade edilir. Gerçekte bu tür bir sorgulamadan tam anlamı ile doğru bir sonuç alınması imkansızdır. Çünkü bu sorgu sonucunda ortalaması 69 ve devamlılığı 9 olan öğrenciler göz ardı edilmiş olacaktır. Bu durumda “Ortalaması iyi ve devamlılığı yüksek olan öğrencilerin listesi” şeklinde bir sorgulama daha doğru olacaktır.

¹ Tescilli Marka



Ŗekil.1 Sistemin genel yapısı

Bulanık sorgulama, kesin olmayan deęerleri *eski, zengin, pahalı, yüksek* gibi dilsel terimlerle belirlemeyi saęlar.

Bulanık sorgulama aracı temelde bulanık önermelere benzer şekilde oluşturulurlar. Bulanık sorgu aracında, sorgu parçaları bulanık yada bulanık olmayan özellik taşıyabilirler, fakat sistem tarafından kontrol edilen bazı sentaks ve semantic kurallara uygun olmak zorundadırlar. Bu kurallara uygun sorgu cümlesi yapısı aŖaęıdaki gibidir.

DEęİŖKEN DEęİŖTİRİCİ BULANIK KÜME DEęİL BAęLAÇ

DEęİŖKEN OPERATOR “SABİT DEęER” BAęLAÇ

Deęişken olarak belirtilenler veritabanının alanları olmak zorundadır. DeęiŖtiriciler, HAFİFÇE, AŖAęIYUKARI, ÇOK, BİRAZ olarak kullanılabilir. Bulanık kümeler bulanıklaŖtırma aŖamasında bir defaya mahsus olmak üzere kullanıcı tarafından tanımlanmalıdır. Baęlaçlar VE yada VEYA olabilir. Operatörler <, >, <=, >= ve <> olarak sıradan karşılaŖtırma operatörleri olabilir. Ayrıca YAKLAŖIK bulanık operatörde sorgulamalarda kullanılabilir. Deęilleme için Türkçe diline yakın olarak DEęİL sorgu cümlesinin sonuna konulmalıdır.

FLEXIBLE QUERYING IN STANDART DATABASE SYSTEMS WITH FUZZY SET APPROACH

SUMMARY

The increasing use of computers in businesses and academical studies has created mountains of data that contain potentially valuable knowledge. To uncover this knowledge, we have to develop a new generation of sophisticated tools with the ability to do flexible querying and intelligent searching. This rapidly growing field is known as Data Mining or Knowledge Discovery in Databases .

In this thesis, a fuzzy query tool for non-fuzzy databases developed. Microsoft Visula FoxPro²(VFP) is used for developing this application. Because VFP is a programable database management system and it is most suitable tool for developing a this kind database application.

System consist of two main part as shown figure 1, fuzzification and queryin. Firstly, user defines fuzzy sets for the attributes of database file and then run queries according to that definition. Also a fuzzy set databases created in order to store fuzzy set definition.

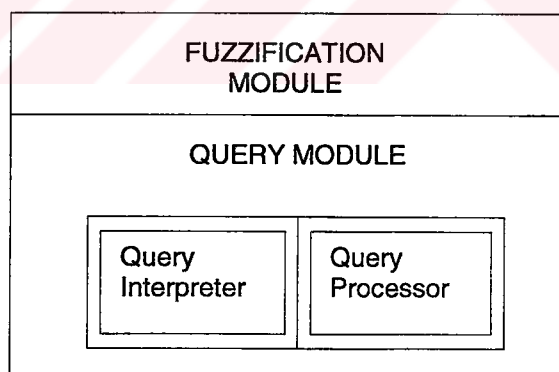


Figure 1. Main parts of system.

In a conventional querying system, one might issue such a query such a query; “List student where average more than or equal to 70 and attendance more than 10.” Actually this kind of searches does not give the exact answer of the query. Because, this query is ignore the students who have average 69 and attendance 9. It

² Treademark

would create crisp sets and ignore noteworthy fuzzy information at the edges. A vague request would be handier and more useful: "List student who have good average and high attendance."

Fuzzy query allows one to deal with the uncertainty of the values as well as vagueness of linguistic terms such as *old*, *rich*, *expensive*. Second, it permits fuzzy retrieval. When searching for 25-years-old with green eyes, one can also identify 23-years-old with green eyes and note the lower degree of match.

Fuzzy query clauses can be constructed similar to the fuzzy propositions. In the Fuzzy query tool, a query clause may be either fuzzy or non-fuzzy, but it must be consistent in some syntax and semantic rules as following.

Linguistic var. [Modifiers] *Fuzzy set* "NOT" [Conj.]

Linguistic var. *Operator* "Constant" [Conj.]

Linguistic variable must be an attribute in the file which query is done. Modifiers are fixed in program as HAFIFCE, ASAGIYUKARI, COK, BIRAZ. Fuzzy sets must be defined by user. Conjunction must be either VE (and) or VEYA (or). Operator is one of the following. =, <, >, >=, <= and <>. There is also a fuzzy operator called YAKLASIK, which converts constant value to a fuzzy number. Constant must start and end with a ". Fuzzy set precedes modifiers NOT, because in Turkish, a natural clause is constructed in that way.

1. GİRİŞ

1.1 Bulanıklık Kavramının Tanımı

Belirsizlik ve bulanıklık günlük yaşamımızın vazgeçilmez bir parçasıdır. Günlük konuşma dilimizde hemen hemen her konuda, tüm ifadelerimiz belirsizlik içermektedir. Bilgide belirsizlik farklı sebeplerden dolayı olabilir. Bu belirsizlikleti Tablo 1.1 deki gibi sınıflandıra biliriz.

Tablo 1.1 Belirsizlik tiplerinin tanımlanma şekilleri

Bilgi Ölçümü	Olasılık	Tanım
Kesin olmayan Hatalı, Yanlış Belirsiz Çelişkili	Kesin Olmayan Muhtemel	Bulanık Belirsiz Yorumlanamaz Muhtemel

Bu tablodanda anlaşıldığı gibi genellikle belirsizliğin ve kesin olmayan bilginin tanımlanmasında, bilgi ölçümünün alanları ile ilişki vardır. Ölçümde belirsizliği, bilginin yetersizliği ile birleştirmemiz mümkündür. Belirsizliğin olasılıktaki görünümü ise bir olayın gelecekte olabilmesi ile ilgili belirsizlik olarak adlandırılabilir. Belirsizlik tanımında, bulanık mantık olarak sınıflandırılan belirsizlikleri ise tümüyle olayın kendisi yada oluşumu ile ilgili belirsizliklerdir.

1965 yılında Prof. L. Zadeh tarafından tanımlanan bulanık kümeler teorisi, kesin olmayan ve belirsiz bilgiyi işleme özelliğine sahiptir. Bu durumda bulanıklık kesin olmamaktan kaynaklanan belirsizlik durumudur. Yapay zekada (YZ) kullanılan bilgi gösterim yöntemleri belirsizlik içeren bilgiyi tanımlamakta yetersiz

kalmaktadırlar. Bu durumda bulanık mantık ve bulanık küme teorisi belirsiz bilgi gösterimi için çok önemli bir rol oynamaktadırlar.

1.2 Bulanık Sorgu Aracı

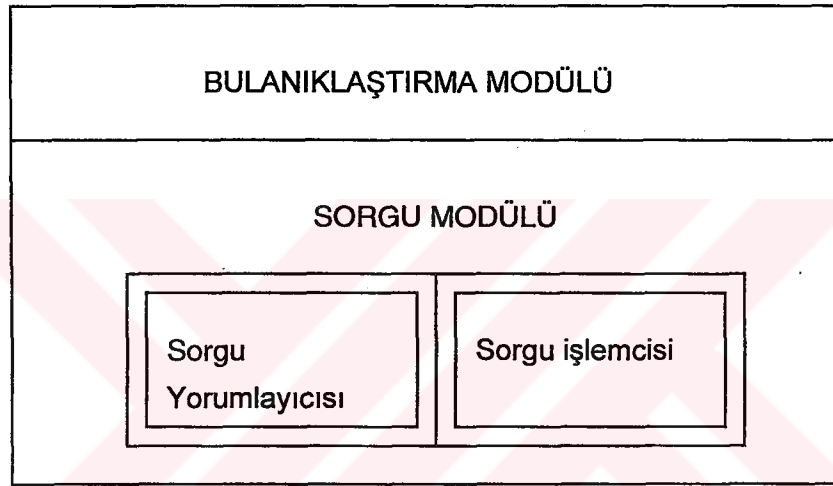
Günümüzde bilgisayarların iş hacimlerinin artması sonucunda meydana gelen veri yığınlarından değerli bilgi oluşturma çalışmaları çeşitli başlıklar altında uzun zamandan beri araştırmacıların ilgi odağı olmuştur. Bilgi keşfi (Knowledge Discovery) , veri madenciliği (Data mining) çalışmaları bu konu ile ilgili hızla gelişen alanlardan ikisidir. YZ konularının sağladığı olanaklar esnek sorgu cevaplama sistemleri (ESCS) için de kullanılmaktadır. Bunlardan en yaygın olanı, günümüze kadar büyük gelişme gösteren ve pek çok alanda kullanılan, bulanık mantık ve bulanık küme teorisidir.

Bu çalışmada, bulanık mantık ve bulanık küme teorisinden yararlanılarak, Win32 tabanlı ve Microsoft Visual FoxPro³ ile geliştirilmiş, kesin olmayan, belirsiz dilsel sorgu cümleleriyle, sıradan veritabanı sistemleri üzerinde bulanık sorgulama yapabilen bir araç oluşturulmuştur.

Programın yapısı temelde iki ana bölümden oluşmaktadır. Bulanıklaştırma modülü ve sorgu modülü. Üzerinde sorgulama yapılan veritabanı dosyaları bulanık olmadığı için öncelikle bulanıklaştırma yapılması gerekmektedir. Sorgu cümleleri Bölüm 3'te ayrıntıları ile açıklanacak olan belli başlı semantik ve sentaks kurallarına uymak zorundadır. Bu kurallar mümkün oldukça esnek tutularak sorgu cümlesinin doğal dildekine olabildiğince yakın kullanılabilmesi sağlanmaya çalışılmıştır. Bulanıklaştırma işlemi veritabanı dosyasındaki herbir alan için istenilen sayıda yapılabilmektedir. Bulanık küme bilgileri bir veritabanında saklanır dosya alanları ile ilgili bilgiler bu veritabanında, dosya adı, alan adı ve bulanık küme isimlerine göre tutlur.

³ Tescilli marka

Sorgu cümleleri bir yada birden fazla cümleden oluşabilir. Tekbir cümleden oluşan sorgu cümlelerine basit sorgu cümleleri denir. Basit sorgu cümleleri MAAS ÇOK AZ DEĞİL şeklinde bulanık bir sorgu cümlesi olabilir. Sorgu cümlesi bulanık yada sıradan bir operatörde içerebilir, bulanık operatör kullanımı durumuna örnek olarak MAAS YAKLAŞIK “150000000” verilebilir. Bu durumda sabit değer olarak girilen sayı bulanık bir sayıya çevrilir. Bu konu ile ilgili ayrıntılı bilgi Bölüm 3’ de verilecektir. Basit sorgu cümleleri VE yada VEYA bağlaçları ile birleştirilerek bileşik sorgu cümleleri oluşturulur.



Şekil 1.2.1 sistemin genel yapısı

Bu tez çalışması ilgilenenleri bilgilendirmek amacı ile sıradan veritabanı sistemlerinde esnek sorgulama araçları oluşturulurken bulanık küme teorisinden yararlanılması konusunda yapılan araştırmalar ve bu çalışmalardan oluşturulan prototip uygulamalar hakkında bilgi vermek amacı ile, tüm bunları içeren bir bölüm içermektedir.

2. SIRADAN VERİTABANI SİSTEMLERİNDE BULANIK KÜME YAKLAŞIMI İLE ESNEK SORGULAMA.

2.1 Giriş

Veritabanı sistemleri oluşumlarından beri pek çok konuya başlık olmuş ve pek çok teknolojinin gelişmesine sebep olmuşlardır. 1970'li yıllarda veritabanı teknolojilerinde temel amaç veriye bağlı sistemlerin oluşturulması, uygulanması ve yönetilmesi için kullanılmaları idi, dolayısı ile veritabanlarındaki gelişmelerde bu yönde oluyordu. Günümüzde veritabanı konusundaki araştırmalar daha genel bir bakış açısı ile yapılmaktadır. Bilgi tabanlı yönetim sistemlerinde karar verebilme yeteğinin kazandırılması için belirsiz ve kesin olmayan verilerin de saklanması sağlanmıştır.

Sıradan veritabanlarında en önemli başlıklardan biriside, kullanıcı tarafından yapılan veritabanı üzerindeki sorgulamanın verimliliğidir. Sorgulamalardaki esneklik gün geçtikçe pek çok çalışmanın konusu olmuştur. Bu konuya yaklaşımların büyük bir bölümü yöntem çalışmaları olmasına karşın bir kısımda kullanıcı arayüzleri alanındadır.

Veritabanı sorgulama sistemlerindeki esneklik kavramının önemini anlamak için daha ayrıntılı açıklamak gerekmektedir. Esneklik kavramı birkaç farklı anlamı içinde barındırmaktadır. Bu kavramları daha yakından incelemek gerekirse öncelikle, sorgulama bakımından veritabanlarının kullanımları bazı gereksinimler içermektedir. 1989 yılında Motro[2] tarafından gösterilen bu temel gereksinimler şunlardır; Kullanıcı temelde veri modelini, yani sorgulama yapacağı veritabanındaki alanları ve bunların yapısını bilmelidir. Kullanıcı sorgulama dili konusunda da yeterince bilgiye sahip olmalıdır. Ayrıca kullanıcı sorgudan verimli sonuç alabilmek için sorguyu boolean olarak iyi bir şekilde tanımlayabilmelidir. Daha esnek sistemler tüm bu gereksinimler biraz daha uygunlaştırarak kullanıcıya hem sorgunun oluşturulması

hemde sonuç alınması sırasında bazı kolaylıklar sağlamalıdır. Esnek sorgu sistemleri bu konuda farklı yaklaşımlarla, değişik çözümler üretmektedir. Bunlardan bazıları; sorgu cümlesi üzerindeki semantik ve sentaks hatalarını otomatik olarak, sorgulama sistemi tarafından düzeltilmesi, sorgu sonuçları arasında niteleyici ayrımlar kullanılması ve sorgu cümlesinin koşul bölümlerine belirsiz terimler koymak, sayılabilir.

Bu durumda bir sistemin “esnek” olabilmesi için belirsizlik içermesi en önemli koşullardan biridir. Pekçok sistem kullanıcıların hazırladıkları sorgularda belirsizlik kullanmasına olanak tanımaktadır, bunlardan bazıları yapılan araştırma çalışmaları sırasında oluşturulmuş prototiplerdir.

Sorgulama sistemlerinde belirsizlik kullanımı için farklı yaklaşımlar bulunması ile birlikte, bizim bu çalışmada üzerinde duracağımız yaklaşım bulanık mantık ve bulanık küme kullanımı ile yapılandırılanlardır.

2.2 Esnek Sorgu Sistemlerinde Bulanık Küme Yaklaşımı

Bulanık küme teorisi sorgulama sistemlerinde ilk olarak 1970 lerin sonlarında Tahani[2] tarafından kullanıldı. Tahani yaptığı çalışmada veritabanından üyelik dereceleri (μ) sayesinde bağıntılar kurmaktaydı. Buna göre R bağıntısı üyelik derecesine göre aşağıdaki gibi tanımlanmaktadır.

$X = \{ X_1, \dots, X_n \}$ kümesi üzerindeki R ilişkisi ;

$$R = \{ x/\mu \mid x \in X, \mu \text{ } x\text{'in R'ye ait olma derecesidir} \} \quad (2.2.1)$$

Bu durumda sıradan bir ilişki $\mu = 1$ olduğu zaman bulanık olmaktadır. Bulanık ilişkiler dilsel değişkenler içeren bulanık niteleyiciler ile sağlanmaktadır. Bulanık kümeler tarafından gösterilen bu dilsel değişkenlere örnek olarak “yaşlı, normal, orta, yaklaşık 50 ” yi verebiliriz. Bu durumda bulanık kümelere uygulanan sıradan işlemlerin sorgunun sonuçlandırılması için de kullanılması sözkonusu olmuştur. Bunu örneklemek için, X kümesi üzerine tanımlanmış bir bulanık R ilişkisi olduğunu

varsayalım. S bu bulanık ilişkiyi P yüklemi ile sınırlanırsın, bu durum aşağıdaki gibi göstere biliriz.

$$\forall x \in X, \mu_S(x) = \min (\mu_R(x), \mu_P(x)) \quad (2.2.2)$$

Buna bağlı olarak diğer küme işlemleri aşağıdaki gibidir.

$$\mu_{R \cup S}(x) = \max (\mu_R(x), \mu_S(x)) \quad (2.2.3)$$

$$\mu_{R \cap S}(x) = \min (\mu_R(x), \mu_S(x)) \quad (2.2.4)$$

$$\mu_{\bar{R}}(x) = 1 - \mu_R(x) \quad (2.2.5)$$

$$\mu_{R - S}(x) = \min (\mu_R(x), 1 - \mu_S(x)) \quad (2.2.6)$$

Kullanıcı boolean bir sorgu cümlesi ile bulanık bir sorgu cümlesini “AND” ve “OR” bağlaçları ile bağlayarak ortak bir sonuç alabilmektedir. Bu yaklaşımın teorisi Tahani tarafından 1976, 1977 yıllarında oluşturuldu fakat bu konuyu örnekleyen herhangi bir sorgulama aracı geliştirilmedi.

1980’li yıllarda benzer bir yaklaşımla Kacprzyk ve Ziolkowski tarafından bir çalışma yapıldı[4]. Bu çalışma Fquery1 adlı prototip bir programla örneklendi. FQuery1, MERA-400 sisteminde Polonya yapımı 16-bit bir minibilgisayar üzerinde DB-83 veritabanına bir ek olarak yapıldı. Sistem nitelenmiş sorgular içermekteydi. Sorgu, “Satışı çok iyi ve yaşı çok genç olmayan çalışanların hemen hemen hepsini bul.” şeklinde yazılabilmekteydi. Yukardaki tanımlarla temelde, kayıtlar için gerekli eşleşme derecelerinin belirlenmesinde kullanılan çalışmalar yapılmıştır. Bu durumda basit sorgular için eşleşme derecelerinin hesaplanması eğer sorgu bulanık bir operatör içermiyorsa ve kesin bir sorgu cümlesi ise, yani <, >, =, >=, <=, <> gibi operatörler içeriyorsa

$$ED = \begin{cases} 1, & \text{sorgu doğru ise} \\ 0, & \text{sorgu yanlış ise} \end{cases} \quad (2.2.10)$$

Eğer sorgu cümlesi basit ve bulanık operatör içeriyorsa, eşleşme derecesinin hesaplanması aşağıdaki gibidir,

$$ED = \mu_{\text{bulanık operatör}}(\text{Dosya alanı}) \quad (2.2.11)$$

Bileşik sorgu cümlelerindeki eşleşme derecelerinin hesaplanmasında, bileşik cümleyi oluşturan basit sorgu cümlelerinin eşleşme dereceleri tek tek hesaplandıktan sonra toplam eşleşme derecesi hesaplanır. Bu sistemde, kesin (“yaklaşık 30”) ve ilişkili (“hepsi, bir kısmı”) niteleyicilerin nasıl yorumlandığını şu şekilde açıklayabiliriz ; Bir R bağıntısı, $\{A_1, \dots, A_p\}$ özellikler kümesi üzerinde tanımlı olsun, R’nin uzantıları da $\{t_1, \dots, t_n\}$ kayıtlar kümesini oluştursun.

$v = \text{“<Q ilişkisel niteleyicisi > P bulanık yüklemine sağlayan R kayıtlar”}$, için

$$v = \mu_Q \left(\frac{1}{n} \sum_{i=1}^n \mu_{p_i}(t_i) \right) \quad (2.2.7)$$

ve g ölçümünde “ $\{P_1, \dots, P_m\}$ üzerindeki $\text{<Q ilişkisel niteleyicisi >}$ ” ni sağlayan R’ nin herhangi bir t parçası için,

$$g = \mu_Q \left(\frac{1}{m} \sum_{i=1}^m \mu_{p_i}(t) \right) \quad (2.2.8)$$

şeklinde gösterilir.

$\mu_i \in [0,1]$, P_i yüklemine önemini belirtiyor ise, t tadafından sağlanan g' değeri:

$$g' = \mu_Q \left(\sum_{i=1}^m \min(\mu_{p_i}(t), \mu_i) / \sum_{i=1}^m \mu_i \right) \quad (2.2.9)$$

olarak belirlenir. Yukarıda g değeri niteleniş birleşik sorgu cümlesinin eşleşme derecesini belirtirken, g' önemine göre nitelenmiş sorgu cümlelerinin eşleşme derecesini göstermektedir.

1980'li yılların sonlarında Patric BOSC[2] bulanık olmayan veritabanları için, esnek sorgulama sistemlerinde uygulama bakımından farklı bir bakış açısı getirdi. Bu çalışmalarda sıradan veritabanlarında kullanılmakta olan standart sorgulama dili SQL'e bulanık sorgu yapabilmesini sağlayan bir uzantı eklendi. Bosc'e göre sıradan VTYS'lerinde sorgulamanın temelde üç basamağı vardı. Bunlardan ilki, temel işlemleri yapabilmek için gerekli olan algoritmanın belirlenmesi, ikincisi sorgunun bir dizi temel işlemlere ayrıştırılması, son olarakda, sorgu daha verimli olabilmesini sağlayacak benzer sorgulara bölünmesi için semantik analiz uygulanmasıydı. Bu durumda sıradan VTYS lere bulanık sorgu uygulamak temelde yukarıda anlatılanlardan çok farklı olmasada bazı zorluklar içermekte idi. Öncelikle uygun olan etki yolları direk olarak kullanılamıyordu. Yani bulanık yüklem içeren bir sorgu cümlesi, sıradan bir sorgu gibi tanımlanamıyordu. Diğer bir zorluk ise, sorgulama sonucunda yapılan seçimlerde bulanık olanların, boolean olanlara göre daha fazla kayıt ile dönmesi idi.

Sıradan bir SQL cümlesinin yapısı bilindiği üzere ,

```
SELECT * FROM <VERİTABANI>
WHERE <ŞART1> AND/OR ...AND/OR <ŞARTn>
GROUP BY ....
HAVING .....
```

şeklinde ifade edilmektedir. Bosc tarafından geliştirilen SQL^f bulanık şartları HAVING kısmında içermektedir. İlk kısım SQL tarafından sonuçlandırılırken bulanık yüklem bulduğu ikinci kısım ise yukarıda anlatılan yöntemle hesaplanır. Bu şekilde farklı iki yöntem bir sorguda karıştırılmış olmaktadır. SQL de seçme işlemi yüklemlere bir dizi toplam fonksiyonu uygulayarak yapılmaktadır. Benzer özellikler SQL^f de farklı boyutlarda kazandırılmıştır. Toplamlar kullanılmış ve sonuçları bulanık yüklem için parametre olarak belirlenmiştir. Bunun dışında birden fazla şartlar birbirlerine bağlanmıştır. Örnek olarak "Ortalama satışları yaklaşık 1600 dolar olan satıcıların bulunduğu ilk 10 bölümü bul." Sorgusu aşağıdaki gibi ifade edilebilir ;

```
SELECT 10 BOLUM FROM CALISANLAR  
WHERE GOREV = "SATICI"  
GROUP BY BOLUM  
HAVING AVG(SATIS) = "YAKLAŞIK 1600"
```

Başka bir format olarakta kesin niteleyiciler (yaklaşık 50, birkaç , ...) yada ilgili (hiçbiri, çoğu, pekazı, ...) niteleyiciler kullanılarak nitelenmiş bulanık önermelerin SQL satırının HAVING bölümüne yazılmasıdır. Bu durumda genel sentaks ;

```
SELECT ... FROM ... WHERE ... GROUP BY ... HAVING ... <nitelenmiş  
önerme> ...,
```

şeklinde yazılabilir.

İki çeşit basit yüklem mümkündür; $Qf \rightarrow fc$; burada Qf , fc bulanık şartını sağlayan kayıtlara uygulanan kesin bir niteleyicidir.

$Qf [fc2] \rightarrow fc1$; burada ise Qf , ilk bulanık şartı sağlamış ($fc1$), buna bağlı olarak ikinci bulanık şartı da sağlayan ($fc2$) alakalı niteleyicilerdir. Eğer $fc1$ çıkartılırsa tüm kayıtlar için geçerlidir.

Tüm bunlara bağlı olarak "çalışanlarını çoğunun genç ve iyi kazananlar" şartını sağlayan 10 bölümü bulmak için ;

```
SELECT 10 BOLUM FROM CALISAN GROUP BY BOLUM  
HAVING ÇOĞU(YAS = "GENÇ") VE "İYİ KAZANAN"
```

şeklinde yazılabilir.

2.3 SummarySQL

1997 yılında Dan Rasmussen ve Ronald R. Yager tarafından[6] , SummarySQL adında , sıradan veritabanları üzerinde, bilinen SQL'e bir ek olarak geliştirmiş bulanık sorgu yapan başka bir araç geliştirildi. SummarySQL'in Bosc'un geliştirdiği SQL^f den farkı, SQL cümlesinin belirsizlik bulunan kısmını içeren bulanık yüklemelerin, dilsel bir özet olarak cümlelerin WHERE ile başlayan koşul kısmında bulunmasıydı. SummarySQL de dilsel özetler doğruluk değerinin ölçümünde

kullanıldıkları gibi bulanık sorguların yüklemeleri olarakta kullanılabilirler. Bir yüklem işleme alındığı zaman aynı bir dilsel özet gibi birim bir aralık değeri almaktadır. Bu sebepten dolayı dilsel değişkenler bulanık yüklemeler, yorumlayıcı tarafından dilsel değişkenler olarak belirlenirler. Dilsel değişkenler bir dizi standart SQL fonksiyonları olan toplam fonksiyonları (AVG, COUNT, ...) ile de ilişkilidirler. Dilsel değişkenler bu fonksiyonlardan farklı olarak birim aralıkta sürekli doğru değeri alırlar. Prototip bir yorumlayıcı olarak geliştirilen SummarySQL'in standart ifade şekli aşağıdaki gibidir.

Select *alan listesi*

From *tablo listesi*

Where *koşullar*

Sorgu sonucu tüm kayıtların ve doğruluk derecelerinin bulunduğu bir tablo şeklinde verilir. Sorgunun from bölümünde belirtilen birleşik-tablolar üzerinde sorgulama yapılan tabloları göstermektedir. Bu bölüme başka bir select ifadesi ile bir altsorgu daha eklenebilir. Koşul kısmında ise format "ALAN is BULANIK TERİM" şeklinde yazılır, bu kısım birleşim "AND", kesişim "OR" yada değilleme "NOT" ifadelerinde içerebilir. Bulanık ifade birleşik-tablodaki herbir kayıt için değerlendirilir ve sonuç doğruluk değerine atanır(μ).

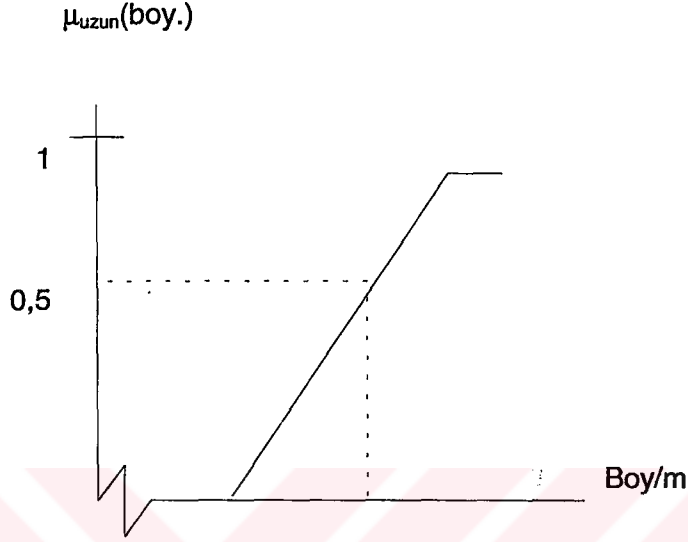
Select *

from Persons

Where Boy is uzun

Ad	Boy	Kilo	μ
John	195	96	1
Ben	193	101	1
Jake	191	97	1
Harry	190	99	1
Jerry	187	96	0,85
Sam	185	97	0,75
Burt	185	96	0,75
Dan	180	93	0,50
Dustin	177	87	0,35
Adam	173	83	0,15
Michael	170	78	0
Eric	168	75	0
Paul	166	70	0

Yukarıda SummarySQL'in basit bir sorgu ile yapılan örneğini görülmektedir. Bu sorguda standart SQL de olduğu gibi sorgulama yapılan tablonun tüm alanlarını almak için '*' ifadesi kullanılmıştır.



Şekil 2.3.1 “uzun” bulanık teriminin tanımı.

SummarySQL deki summary ifadeside aşağıdaki formatda kullanılmaktadır.

Summary *niteleyici*

from *tablo listesi*

where *koşullar*

Bu kısımda “*niteleyici*”, bulanık bir niteleyicidir ve “from” ve “where” ifadeleri daha önceki tanımlandıkları şekilde kullanılırlar. Bu ifadede from kısmında kullanılan birleşik-tablo bir bulanık alt sorgu olabilir.

Summary çoğu

From (select * from Persons where boy is uzun)

Where kilo is zayıf ;

Result : 0.943878

Summary ifadesi örnek.

Şekil 3 deki örnekte olduğu gibi içi içe bulunan iki sorgu cümlesi görülmektedir. En içteki sorgu cümlesi dıştaki sorgu cümlesinin from ifadesi için bir birleşik tablo oluşturmaktadır. Altsorgu cümlesindeki sorgu uygulandıktan sonra kendi tblosundaki alanları otomatik olarak dış sorgu cümlesine bırakır. Sonuç olarak summary ifadesi ile çoğu ifadesinin doğruluk derecesi $\tau = 0,94$ olarak hesaplanır.

Summary ifadesi, sorgunun dilsel özeti olarak adlandırılır. Bir dilsel özet, veritabanındaki bilgi ile ilgili genel tanımı içermekte ve veri hakkındaki ilişkisel bilginin gösteriminde kullanılmaktadır. Dilsel özete bir örnek “Veritabanındaki uzun insanların bir kısmı” şeklinde verilebilir. Biçimsel olarak tanımlandığında dilsel özet format olarak sorgu cümlesi içinde, “VT S nesne Q” için S özetleyici, Q ise sorgunun uyuşmasındaki ölçü olarak adlandırılabilir. Ayrıca dilsel özet bir doğruluk derecesine yani $\tau \in [0,1]$ atanır. S olarak belirtilen dilsel genel kavramın bulanık ifadesidir ve sorgulamak üzere seçilen veritabanının bir alanında tanımlanır. Bir diğer dilsel özet bileşeni ise bağdaşma ölçümü olarak adlandırılan ve Zadeh tarafından dilsel niteleyiciler olarak adlandırılan bölümdür.

Bir dilsel özetin τ 'nu hesaplayabilmek için, $VT = \{o_1, \dots, o_n\}$ veritabanını oluşturan katıylar kümesi olsun. S özetleyici olarak kullanılan bulanık bir ifade olsun, Q bulanık niteleyici, A'da VT veritabanında bit alan olarak tanımlansın. VT veritabanındaki kayıtlar için yapılacak dilsel özetleme işlemi aşağıdaki gibi tanımlanabilir;

1. Her o_i VT için $S(o_i)$ hesaplanır
2. $r = \frac{1}{n} \sum_{i=1}^n S(o_i)$, S'yi sağlayan VT'deki kayıtların oranları bulunur. (2.3.1)
3. Sonuç olarak $\tau = Q(r)$, r'nin üyelik derecesi bağdaşma ölçümü olarak hesaplanır.

Bir başka dilsel özetleme örneğinde “VT'deki çoğu uzun kişi gençtir.” şeklinde verilebilir. Bu gibi sorgularda sözkonusu olan, *uzun'lar* kümesinin altkümeleridir. Bu tür sorgular veritabanlarından bulanık kuralların çıkartılmasında kullanılmaktadır. Bu

sorguda, **Q** çoğu, **R** uzun, genç **S** olarak tanımlanabilir. τ 'nın hesaplanması I ikinci basamak hariç yukarıdaki gibidir. Bu durumda r ise aşağıdaki gibi hesaplanabilir,

$$r = \frac{\sum_{i=1}^n T(S(o_i), R(o_i))}{\sum_{i=1}^n R(o_i)} \quad \text{olarak hesaplanabilir.} \quad (2.3.2)$$

S ve R özet kısmında belirtilen kavramın bulanık alt kümeleri ve T de [2] bir t-norm işlemidir. T-norm, minimum ve çarpım altında tanımlı ve *and* operatörünü sağlayan bir grup ikili operatördür.

2.4 Sonuç

Tüm diğer çalışmalarda da görüldüğü gibi bulanık sorgu araçlarını hemen hemen hepsi, veritabanı sistemlerinde bir standart olan SQL sorgulama diline bir uzantı olarak geliştirilmiştir. Bu tez çalışmasının yapıldığı sırada, Amerika Birleşik Devletlerinde Sonalysts inc. şirketi tarafından benzer özelliklerde bulanık sorgulama yapabilen ticari bir araç oluşturulmuştur. Sonalysts şirketi tarafından web sayfasında yapılan program tanıtımından elde edilen bilgiler ışığında bu sorgulama aracınında yöntem olarak diğerleri ile benzer bir yol izlediği gözlenmiştir. Kullanım olarak bu tez çalışması sırasında oluşturulan BSorguWin aracı ile benzerlikler göstermesine rağmen büyük farklılıklar içermektedir.

BSorguWin bulanık sorgulama aracı, yöntem olarak tüm bu çalışmalardan farklı bir yol izlemesinin yanında, Kacprzyk ve Ziolkowski[4] tarafın oluşturulan eşleşme derecesi bulma yöntemlerini hareket noktası alarak, 1996 yılında Taner MUTLU ve Doç. Dr. Gazanfer ÜNAL tarafından yapılan yüksek lisans tez çalışmasının bir devamı olarak oluşturulmuştur.

Bölüm 3 de programın geliştirilmesi ve akışı ile ilgili ayrıntılı bilgi verilmiştir.

3. BULANIK SORGU UYGULAMASI

3.1 Giriş

Bilgisayarların kullanımlarının artması ile ağ bağlantıları, yüksek hızlı ve kapasiteli veri depolama sistemleri geliştirilmiş ve işlem sayısı her geçen gün artmıştır. Bunun sonucu olarak işlenmesi gereken veri miktarı ve esnek sorgu cevaplama sistemlerinin önemi artmıştır. Verinin yararlı bir şekilde işlenmesi için çeşitli yöntemler ile birçok araç geliştirilmiştir. YZ yöntemleride bu alanda kullanılanların başında gelir. Özellikle bulanık mantık ve bulanık küme yaklaşımının pek çok başarılı örneği bulunmaktadır. Bu çalışmada bulanık mantık ve bulanık küme kullanımı ile sıradan veritabanlarında sözsel sorgu yapan *BsorguWin* adlı bir araç geliştirilmiştir.

1. Bölümde, bulanık sorgulamanın tanımlanması ve bulanık sorgu tipleri açıklanacak ve sıradan, sorgulama ile arasındaki farklılıklar örneklerle açıklanacaktır.
2. Bölümde, Bulanık sorgu aracının genel yapısı ve işleyişi alt bölümler halinde açıklanacaktır.

3.2 Bulanık Sorgulama

Kullanıcı elindeki verilerle değerli bilgi elde etmek için öncelikle uygun yapıda güvenilir bir veritabanına gereksinim duyar. Daha sonra bu veritabanından değerli bilgiyi çıkartabilecek uygun bir sorgulama aracı gerekmektedir. Sıradan veritabanlarında bulunan veya şu anda kullanılmakta olan pek çok bağımsız sorgu araçlarının temel özeliği sorguyu sadece belirli kriterler yardımı ile kesin kıyaslamalar veya aralıklarda yapabilmeleridir. Oysa gerçek hayattaki sorular bu kadar kesin sınırlarla çizilmez. Örneğin herhengi bir sınıftaki öğrencilerin başarı durumlarını sorgularken “Ortalaması ≥ 70 ve devamı < 10 olan öğrenciler

kimlerdir?” şeklinde bir sorgulama kullanmayız. Bu tür sorgulamalar genelde daha esnek ve açık aralıklarda sorulur. Aynı örneği günlük dilde kullanıldığı şekile çevirirsek *“Ortalaması iyi olan ve devamı normal öğrenciler”* yada sorgu alanını kısıtlamak istediğimizde her zaman kullandığımız değiştiricilerle anlamı biraz daha kuvvetlendirerek *“Ortalaması çok iyi olan öğrenciler”* şeklinde sorulabilir. Bu tür belirsiz niteleyiciler içeren sorgulama türleri bulanık sorgulama olarak adlandırılır.

Bulanık sorgulamanın yararlarından ilki, yapılan sorgudan daha yararlı bilgi almaktır. Kesin sorguda alınacak bir sonuç yeterli derecede bilgi veremeyebilir. Örneğin, yukarıda sözü edilen sorgu cümlesi ele alırsak bu sorgu sonucu , ortalaması 69 ve devamı 11 olan öğrenciler başarısız olarak belirlenecektirler. Dolayısı ile başarı için belirlediğimiz kriterlere çok yakın olan öğrenciler değerlendirmeye alınmayacaktır. Bu durumda standart sorgu mekanizması sağlıklı bir sonuç vermiş sayılamaz. Bulanık sorgulama sayesinde, kullanıcı sorgu yapılan veritabanındaki tüm kayıtlar için bir bilgiye sahip olur bu da sorgunun esnekliğini büyük ölçüde geliştirir. Diğer bir yararıda, bilgisayar kullanımında kullanıcıya kolaylık sağlamasıdır.

3.3 Bulanık Sorgu Tipleri

Bulanık sorgu tiplerini temel olarak iki ölüme ayırabiliriz ;

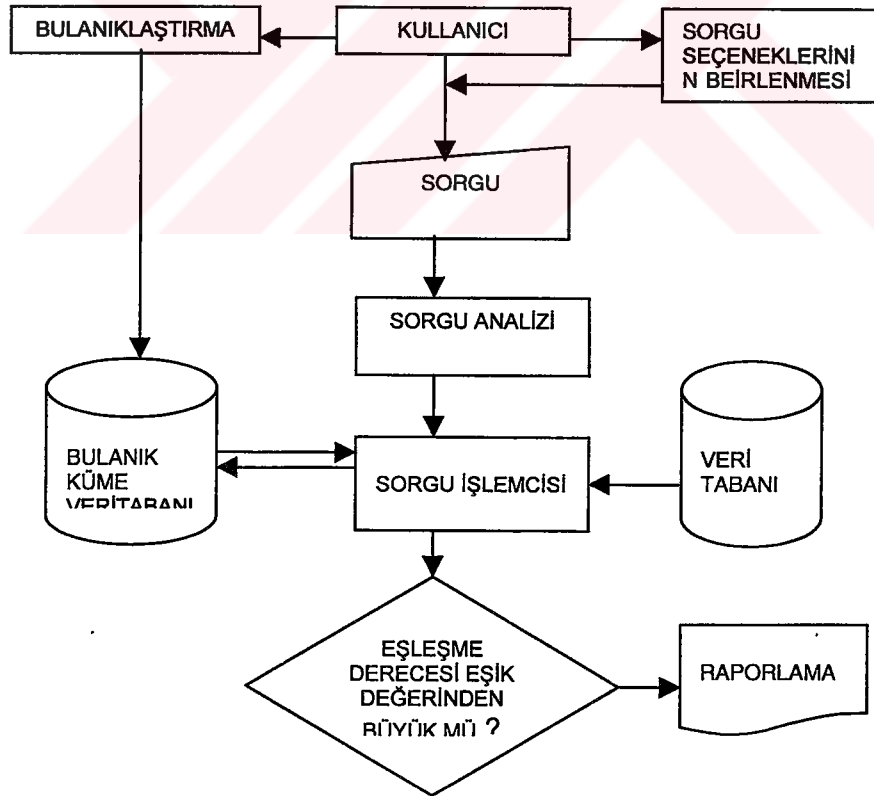
- i. Basit Cümleler
 - a) Bulanık olmayan basit sorgular
 - b) Bulanık operatörlü basit sorgular
- ii. Bileşik Cümleler

3.4 Programın Genel Yapısı ve Akışı

Tüm bunların ışığında sıradan veritabanları üzerinde doğal dildekine benzer sorgu cümleleri ile sorgu yapılabilen bir araç geliştirilmiştir. Program Windows 9X işletim sistemi üzerinde Visual FoxPro (VFP) ortamında geliştirilmiştir. VFP temelde XBASE üzerine kurulmuş ticari bir veritabanı uygulamaları geliştirme ve yönetim sistemidir. DBASE benzeri tüm veritabanı dosyalarına doğal erişim

yapabilmesi yanında, “Open Database Connectivity”(ODBC) olanakları ile diğer veritabanı sistemleri üzerinde de işlem yapabilmektedir. Programın geliştirilmesinde VFP’nun nesne yönelimli programlama yapısı, sınıf (class) yapıları, dinamik diziler ve ActiveX gibi özellikleri kullanılmıştır.

Program iki ana bölümden oluşmaktadır. Öncelikle üzerinde sorgu yapılacak veritabanının istenilen alanları bulanıklaştırılır ve bu işlemle ilgili bilgiler bulanık küme veritabanında saklanır. Sorgu cümlesi alındıktan sonra semantic ve sentaks kurallarına göre analiz edilir. Bileşik bir sorgu cümlesi basit alt cümlelere bölünerek analiz edilir. Analiz sonuçları sorgu işlemine hazırlanması için dinamik bir diziye yerleştirilir. Daha sonra dizideki verilere göre sorgu işlemcisi tarafından veritabanındaki tüm kayıtlar için her alt cümle için eşleşme derecesi bulunur. Bulunan alt cümle eşleşme dereceleri ile toplam eşleşme dereceleri hesaplanır. Eğer toplam eşleşme derecesi sorgu öncesi belirlenen eşik değerinden büyük ise ekrana raporlanır. Bu işlem dosyadaki tüm kayıtlar için yapılır. Programın akış diagramı Şekil 3.4.1 de gösterilmiştir.



Şekil 3.4.1 Program akış diagramı

Programın basamakları aşağıdaki gibi sıralanabilir.

1. Bulanıklaştırma
2. Bulanık kümelerin tanımı
3. Bulanık kümelerin üyelik fonksiyonlarının belirlenmesi
4. Sorgu ile ilgili seçeneklerin belirlenmesi
5. Sorgu cümlesinin alınması
6. Sorgu cümlesinin analizi
7. Sentaks ve semantik kuralların uygulanması
8. Analiz sonuçların derlenmesi
9. Sorgu işlemcisinin çalışması
10. Veritabanındaki bütün kayıtlar için alt sorgu cümlelerinin eşleşme derecelerinin bulunması
11. Veritabanındaki bütün kayıtlar için toplam eşleşme derecelerinin bulunması
12. Eşik değerini sağlayan kayıtların raporlanması.

Basamakları basit bir veritabanında örnekleyelim. Veritabanımız, Tablo 3.4.1 de gösterildiği gibi, bir sınıftaki öğrencilerin ortalamaları ve devamlarını içermektedir.

Tablo 3.4.1 Örnek veritabanı

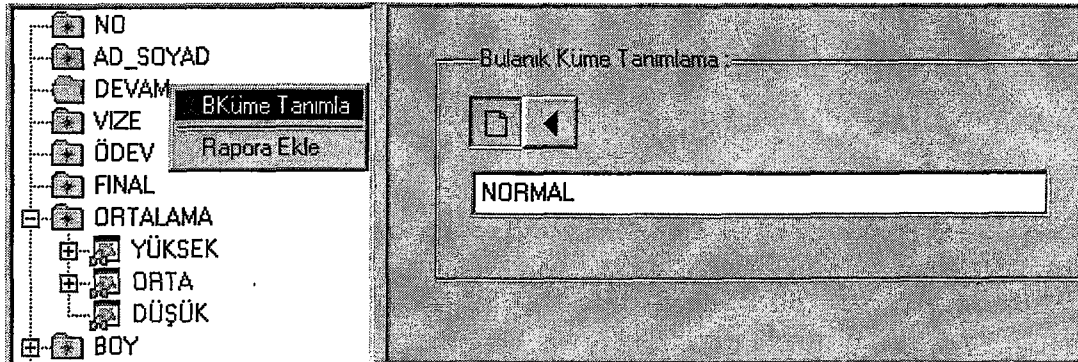
AD_SOYAD	DEVAM ORTALAMA	
HÜSEYİN ETİKAN	3	20
İSMAİL EKİNCİ	4	28
BEKİR İNCE	8	30
TEOMAN DAĞCI	12	40
BELKİS EMREM	1	44
CELAL ÖZTÜRK	19	50
NİMET DOĞRU	17	65
MUSTAFA YILDIZ	10	67
RECEP ŞAHOĞLU	18	76

3.5 Bulanıklaştırma

3.5.1 Bulanık kümelerin tanıtımı :

Sorgulama yapılan veritabanları bulanık olamadığından sorgulamadan önce bulanıklaştırma işlemi yapılmalıdır. Bulanık kümeler sorgu için seçilen veritabanı ve üzerinde bulunulan alan ile belirlenirler. Bulanık küme bilgileri BNMST adlı bulanık küme veritabanında saklanır. Bulanık küme veritabanında üzerinde sorgu yapılacak olan veritabanını adı, bulanık kümenin ait olduğu veritabanı alanı, alanın tipi ve uzunluğu ile ilgili bilgiler bulunmaktadır. Veritabanı dosyasındaki alanların oluşturduğu bulanık küme elemanları ve üyelik dereceleri BNDET dosyasında saklanır .BNMAS ve BNDET dosyaları bir kod alanı ile ilişkilendirilmişlerdir.

Dosyadaki DEVAM ve ORTALAMA alanları için sırası ile NORMAL ve YÜKSEK bulanık kümelerini tanımlayalım. Bulanık kümelerin tanımlanması için sorgulama yapılacak dosya seçilerek yüklendikten sonra, ekranın sol tarafında dosyanın alanları listelenir. Bulanık küme tanımlanmak istenen alanın üzerine gelindiğinde açılan kısayol menüsünden “Bküme Tanımla” seçeneği seçilir. Ekranın sağ tarafında açılan alandan bulanık kümelerin tanımlanması yapılır. “Ekle” tuşuna basılarak bulanık kümenin oluşturulması tamamlanır.



Şekil 3.5.1.1. Bulanık küme tanımlama ekran kesiti.

Bu şekilde veritabanının her alanı için istenildiği kadar bulanık küme tanımlanabilir. Bulanık kümeler tanımlandığı dosya ve alan ile belirlenirler. Bu şekilde günlük dilde kullanıldığı gibi farklı alanlar için aynı isimli bulanık kümeler tanımlanabilir. Örneğin veritabanımızdaki DEVAM ve ORTALAMA alanlarının her ikisi için NORMAL, YÜKSEK gibi bulanık kümeler tanımlanabilir.

3.5.2 Bulanık kümelerin üyelik fonksiyonunun belirlenmesi

Üyelik fonksiyonlarının belirlenmesinde iki farklı yol kullanılmaktadır. Bunlardan ilki, üyelik fonksiyonunun 3.5.2.1 denkleminde gösterildiği gibi eleman ve üyelik derecesi çifti olarak tanımlanmasıdır.

$$\sum_{i=1}^n \mu_A(x_i)/x_i = \mu_A(x_1)/x_1 + \mu_A(x_2)/x_2 + \dots + \mu_A(x_n)/x_n \quad (3.5.2.1)$$

Bunun için veritabanında seçilen alan sayısal ise alt ve üst sınırları program tarafından kullanıcıya gösterilir. Buna göre kullanıcı bu değerler arasında istediği kadar bulanık küme elemanı ve üyelik derecelerini tanımlar. Üyelik dereceleri [0,1] aralığında olmalıdır. Üyelik fonksiyonları sürekli kabul edilmiştir. Eğer veritabanı alanı sayısal değer içeriyorsa ve üyelik derecesi kullanıcı tarafından belirlenmemişse enterpolasyon ile hesaplanır. Enterpolasyon yöntemi kullanıcı tarafından belirlenir. Eğer üyelik derecesi hesaplama yöntemi olarak fonksiyon kullanımı belirlenmişse, sistem üyelik derecelerini, bulanık küme tanımları ile BNMAS dosyasına kaydedilen üyelik fonksiyonu ile hesaplar ve elde edilen üyelik derecelerini BNDET dosyasına kaydeder. Veritabanı alanı karakter değeri içeriyorsa üyelik derecesi 1 olarak alınır.

Program, sorgu yapılan veritabanının üyelik derecelérinin belirlenmesi için seçilen alanın, sayısal değér içeriyor ise en yüksek ve en düşük değérini hesaplar ve kullanıcıya gösterir. Kullanıcı bu değérlerde dahil olamk üzere toplam beş eleman- üyelik derecesi çifti tanımlayabilir. Bu işlemin yapıldığı ekranın kesiti şekil 3.4.2.1 de gösterilmiştir.

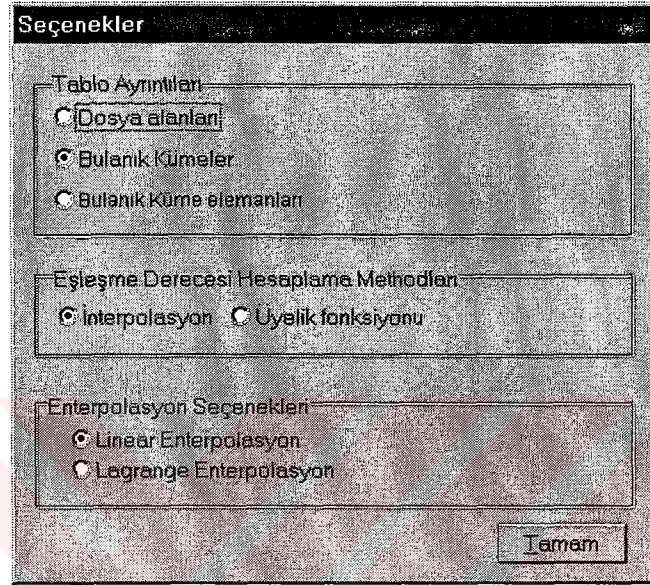
İkinci olarak üyelik fonksiyonları, şekil 3.5.2.2’de gösterildiği gibi programda doğrudan fonksiyon kodu olarak tanımlana bilirler. Bu kod sıradan işlemlerin bilgisayarda tanımlanması gibi temel matematiksel fonksiyonların ve parantezlerin kullanılması ile yapılır. Değişken olarak “x” kullanılır. Ekrandan kullanıcı tarafından girilen üyelik fonksiyonu bulanık küme bilgilerinin tutulduğu veritabanında bir alanda saklanır. Kullanıcı tarafından programa tanımlanan üyelik fonksiyonu, seçenekler modülünde eşleşme derecesi hesaplama methodlarında belirlenerek işler hale getirilir.

Şekil 3.5.2.1 Bulanık küme elemanları ve üyelik dereceleri tanımlama ekran kesiti

Şekil 3.5.2.2. Bulanık küme ve üyelik fonksiyonu tanımlama ekran kesiti

3.6 Sorgu Seçeneklerinin Belirlenmesi

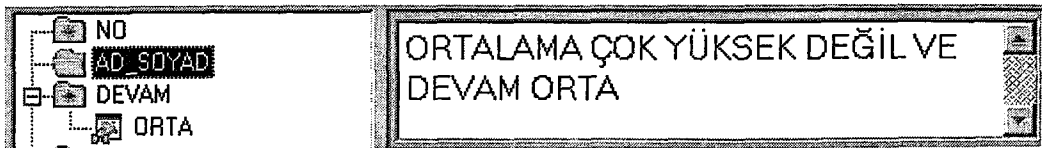
Sorgulama aracı sorgulama işlemi ile ilgili bazı seçimleri bir arayüz ile kullanıcıya bırakmıştır. Bu seçenekler içinde üyelik derecelerinin belirlenme yöntemleri ve bazı raporlama seçenekleri vardır. Bu arayüz şekil 3.6.1 de gösterilmiştir.



Şekil 3.6.1 Seçenekler ekran kesiti.

3.7 Sorgu Cümlesinin Alınması

Temel olarak programda yapılacak sorgunun herhangi bir sınırlaması yoktur. Bağlaçlarla bağlanmış istenilen sayıda sorgu cümlesi girilebilir. Programın bu kısmındada kullanımı kolaylaştırmak ve hatayı azaltmak için iki farklı yöntem geliştirilmiştir. Sorgu cümleleri, Şekil 3.7.1’de gösterildiği gibi, ekranın sağ üst kısmındaki alandan tuşlanarak yazılabilir yada windows ortamının bir özelliği olan “çek ve bırak” ile tamamen fare kullanarak oluşturulabilir.



Şekil 3.7.1 Sorgu cümlesi alınması ekran kesiti.

Örnek veritabanımız için aşağıdaki gibi bir sorgu oluşturalım.

“ORTLAMA ÇOK YÜKSEK DEĞİL VE DEVAM ORTA”

3.8 Sorgu Analizi

Sorgu cümlesi alındıktan sonra alt cümlelere bölünerek dinamik bir tabloya yerleştirilir. Tablodaki her alt cümle semantik ve sentaks kurallarına göre kontrol edilir. Sorgu işlemcisi devreye girmeden önce, alt cümlelerden herhangi birinde bir hata bulunursa program ekrana hata mesajı verir. Sorgu işlemcisi devreye girmeden önce altsorgu cümleleri ile ilgili tüm veriler dizi aracılığı ile belleğe yüklenir. Sorgu işlemcisi, icrası sırasında her alt cümlelerin eşleşme derecesini bulurken bu verilerden yararlanır. Sorgu cümlesinin bilgisayarda gösterimi ve analizi için kurulan veri yapıları bölüm 3.11’de daha ayrıntılı bir şekilde açıklanacaktır.

3.9 Sorgu İşlemcisi

Sorgu cümlesi ile ilgili analiz işlemleri bittikten sonra sorgu işlemcisi veritabanındaki her kayıt için eşleşme derecelerini hesaplar. Eğer eşleşme dereceleri eşik değeri olarak belirlenen kabul edilebilirlik şartını sağlıyorsa ekrana raporlanır.

Cümlede “YAKLAŞIK” bulanık operatörü bulunuyorsa, operatörden sonra belirtilen sabit değer kısmındaki sayı bulanık sayıya çevrilir. Bu işlemin yapılması için veritabanında sorgu yapılan alanın kayıtlarının en büyük ve en küçük değerleri bulunur ve bunların farkı alınır. Daha sonra veritabanındaki kayıt sayısının oranı kadarlık bir kısmı (örneğimiz için bu değer $\pm 1/10$ dur) $\alpha = 0$ olmak üzere α -düzey kümesi olarak kullanılır.

Örnek veritabanımız için yapılacak ORTALAMA YAKLAŞIK “50” sorgulamasının işleyişi yukarıdaki açıklamalar ışığında şöyle olacaktır.

$$\text{Max ortalama} = 90 \quad (3.9.1)$$

$$\text{Min ortalama} = 20 \quad (3.9.2)$$

$$d = (90-20)/10 = 7 \quad (3.9.3)$$

$$43 \leq \text{ORTALAMA} \leq 57 \quad (3.9.4)$$

bu aralık dışındaki tüm ORTALAMA ların eşleşme dereceleri 0 olarak alınacaktır.

Aralık içerisindeki herhangi bir χ ortalamasının eşleşme dereceleri ise;

$$ED = 1 - (|50 - \chi| / d) \quad (3.9.5)$$

şekilde hesaplanır .

Bulanık sorgu cümlelerinin, işlemci tarafından icrası şu şekilde basamaklandırılabilir. İlk alt sorgu cümlesi ORTALAMA ÇOK YÜKSEK DEĞİL in eşleşme derecesi ed_1 , ikincisi DEVAM ORTA 'nın ed_2 ve toplam eşleşme derecesi Σed olsun. ÇOK değiştiricisinin etki değeri değiştiricilerin tanımlandığı modülde tanımlanmıştır. Buna göre ÇOK değiştiricisinin uygulandığı bulanık kümenin veritabanındaki kayıt için eşleşme derecesi;

$$\text{ÇOK}(ed) = ed^2, \text{ olarak hesaplanır.} \quad (3.5.6)$$

Eğer dorgu alt cümlesi değilme içeriyorsa değilme işlemi,

$$ed_1 = 1 - ed_1^2, \text{ şeklinde hesaplanır.} \quad (3.5.7)$$

Veritabanındaki her kayıt için toplam eşleşme derecesi ise aşağıdaki gibi hesaplanmaktadır;

$$\Sigma ed = ed_1 \wedge ed_2 = \min(ed_1, ed_2) \quad (3.5.8)$$

3.10 Sorgu Cümlesinin Yapısı

Yorumlayıcı tarafından analiz edilebilmesi için sorgu cümlelerinin bazı standart şekillerde alınması gerekmektedir. Bu standart sorgu yapısı bulanık olan ve bulanık olmayan sorgu tipleri için aşağıdaki gibi gösterilebilir.

DEĞİŞKEN DEĞİŞTİRİCİ BULANIK KÜME DEĞİL BAĞLAÇ

DEĞİŞKEN OPERATOR “SABİT DEĞER” BAĞLAÇ

Koyu olarak belirtilen kısımlar zorunlu olarak sorgu cümlesinde bulunması gerekenlerdir. Bulanık bir sorgu cümlesinde en azından değişken yani sorgu yapılacak veritabanından bir alan ve bu alan için tanımlanmış bir bulanık küme bulunmalıdır. Bulanık sorgularda isteğe bağlı olarak değiştiricilerde kullanılabılır. Değiştiriciler programın başka bir modülünden tanımlanabilmektedirler. Eğer sorguda kullanılan değiştirici, değiştiricilerin saklandığı veritabanında bulunamazsa sorgu hata mesajı ile sonuçlanır.

Benzer bir şekilde bulanık olmayan ve bulanık operatörlü sorgularda bulunması zorunlu olan üç kısım vardır, bunlar sorgulanan veritabanı alanı, bulanık yada sıradan bir operatör ve karşılaştırma yapılacak bir sabit değerdir.

Bulanık olmayan, sıradan sorgu cümleleri herhangi bir veritabanı üzerinde bulunana geleneksel sorgu operatörlerini kullanır. Bu tür sorgularda $<$, $>$, $\diamond$, $=$, $\leq$ yada $\geq$ gibi semboller operatör olarak kullanılır. Eğer operatör YAKLAŞIK operatörü ise sabit kısmında belirtilen sayı bulanık sayıya çevrilir. Tüm sabit değerler çift tırnak arasında gösterilmelidir. Bu işlemle ilgili ayrıntı bölüm 3.9’da verimiştir. Sorgu cümleleri bağlaç olarak VE yada VEYA kullanılarak birbirlerine bağlanmalıdırlar. Sorgu cümleleri bağlaçlarına göre sorgu işlemcisi tarafından parçalanarak işleme alınırlar. Sorgu işlemcisi öncelikle VE ile bağlanmış olan alrsorgu cümlelerinin eşleşme derecelerini bulur daha sonra bu eşleşme dereceleri VEYA ile bağlanan ana sorgu cümlesinin eşleşme derecesini oluşturur. İki alt sorgu cümlesi arasında sadece bir bağlaç kelimesi bulunabilir.

Sorgu cümlesinde bulunan bulanık küme daha önceden bulanıklaştırma modülü ile tanımlanmış olmalıdır, aksi taktirde sorgu sonucu hata mesajı olarak döner.

3.11 Sorgu Cümlesinin Analizi ve Gerekli Verilerin Derlenmesi

Sorgu cümlesi sorgu analiz fonksiyonları tarafından parçalanarak alt cümlelere bölünür. Parçalanmış alt cümleler kendi içlerinde analiz edilirler. Analiz sonuçları dinamik bir diziye yerleştirilirler. İki boyutlu olan bu diziler alt sorgu cümlelerinin sayısına göre dinamik bir yapıya sahiptirler. Şekil 3.4.1 de gösterildiği gibi satır sayısı alt sorgu cümlesi kadar olurken sütun sayıları, analiz fonksiyonları sayısı kadardır.

1	2	3	4	5	6	..	..	..	18	19
...										
...										

Şekil 3.10.1 dinamik dizi yapısı

Sorgu işlemcisinin devreye girmesinden önce BSORGU formunun *Ön_işlem* fonksiyonu altsorgu cümlelerini yukardaki yapıya uygun olarak diziye yerleştirir. Dizideki herbir sütun için BSORGU formunun bir nesnesi olarak oluşturulan BSORGU1'in içerdiği fonksiyonlardan yararlanır. Yerleştirme, bu fonksiyonların çağrılarak parametre olarak altsorgu cümlesi verilmesi ve sonuçlarının ilgili hücrelere atanması ile yapılır.

BSORGU sınıfının önemli fonksiyonlarından bazıları aşağıda açıklamaları ile verilmiştir. Fonksiyonların tümünün kaynağı kodları EK-C de buluna bilir.

3.11.1 BulanikOprr :

```
LParameters OPRBul
If This.TirnakBul(OPRBul) = .t. AND OCCURS("YAKLAŞIK",OPRBul)>0
    RETURN .t.
Else
    RETURN .f.
EndIf
```

BulanikOprr : Bu fonksiyon altsorgu cümlesinde bulanık operatör olup olmadığını bulur. Bulanık operatörün program içinde işleme alınması için öncelikle sorgu cümlesinde tırnak işareti olup olmadığı kontrol edilir bu sayede sorgu cümlesinin sabit bir değer içerip içermediği anlaşılmış olur. Eğer bu fonksiyonun sonucu mantık operasyonu olarak “.T.” yani TRUE dönerse , sorgu işlemcisi bu sorguyu “bulanık operatörlü” olarak algılar ve işleme koyar.

3.11.2 BulOptHesap :

```
Lparameters Alan,Deger,AlanDeg,d
Private Ed
if BETWEEN(AlanDeg,INT(VAL(Deger))-d,INT(VAL(Deger))+d)
    Ed = 1-(ABS(INT(VAL(Deger))-AlanDeg)/d)
Else
    Ed = 0.000
EndIf

RETURN Ed
```

BulOptHesapla fonksiyonu, bölüm 3.2.4 te tanımlanan denkleme göre alfa-kesmelerini bulur ve bulanık operatörlü sorgu cümlesindeki sabit sayıyı bulanık sayıya dönüştürür. Eğer YAKLAŞIK operatörü bulunamazsa, sorgu cümlesi basit sorgu cümlesi olarak adlandırılır ve basit sorgu uygulanır. Bunun anlamı veritabanındaki kayıtlar eğer bu basit sorguyu sağlıyorsa eşleşme derecesi olarak 1.000, sağlamıyorlarsa 0.000 değeri alırlar. Cümle tırnak işareti yani sabit değer içermiyorsa bulanık sorgu cümlesi olarak algılanır ve aşağıdaki fonksiyonların sonuçları devreye girer.

3.11.3 BulanikKumeBul :

```
LParameters BKBul
Private Boşluk, Değil, TirnakX
Boşluk = This.BosBul(BKBul)
Değil = This.NotBul(BKBul)
TirnakX = This.TirnakBul(BKBul)
Do Case
Case Boşluk > 2 and Değil = .t. and TirnakX = .f.
RETURN substr(BKBul,at_c(' ',BKBul,2)+1,at_c(' ',BKBul,3)-at_c(' ',BKBul,2))
Case Boşluk > 1 and Değil = .t. and TirnakX = .f.
RETURN substr(BKBul,at_c(' ',BKBul,1)+1,at_c(' ',BKBul,2)-at_c(' ',BKBul,1))
Case Boşluk > 1 and Değil = .f. and TirnakX = .f.
RETURN ALLT(substr(BKBul,at_c(' ',BKBul,2)+1,50))
Case Boşluk = 1 and Değil = .f. and TirnakX = .f.
RETURN ALLT(substr(BKBul,at_c(' ',BKBul,1)+1,50))
Otherwise
RETURN ''
EndCase
```

3.11.4 BkumeAra :

```
Lparameters Bkume,DosyaAd,DegAlan
Private durum,Kod,Tip
durum = .f.
If !EMPTY(Bkume)
Select BNMST
SCAN FOR Allt(Mfq)=Allt(Bkume) and
Allt(Mfile)=Allt(DosyaAd) and Allt(Mfld)=Allt(DegAlan)
durum = .t.
Kod = Msq
Tip = Mtyp
ENDSCAN
if durum = .t.
RETURN Allt(Kod+', '+Tip)
Else
RETURN ''
Endif
Else
RETURN ''
EndIf
```

BulanikKumeBul fonksiyonu, altsorgu cümlesini alır ve içindeki bulanık kümeyi bulur. Buradan elde edilen sonuç daha sonra eşleşme derecesinin bulunmasında ve BkumeAra fonksiyonunda parametre olarak kullanılır. BkumeAra fonksiyonu bulanık küme veritabanında bulanık küme adına göre arama yapar ve sonuç olarak bulanık kümenin veritabanındaki kodu ile alan tipini verir. Bu veriler sorgu işlemcisinin icrasında kullanılmak üzere sorgu cümlelerinin tutulduğu dizide ait oldukları hücreye yerleştirilirler.

3.12 Sorgu İşlemcisi ve Eşleşme Derecelerinin Bulunması

Herhangi bir hata içermeyen sorgu cümlesi, daha sonra sorgu işlemcisi tarafından işleme alınır. Sorgu işlemcisi veritabanındaki herbir kayıt için altsorgu cümlelerinin eşleşme derecelerini bulur. Daha sonra toplam eşleşme derecesini hesaplar ve bunu eşik değeri ile karşılaştırır. Eşik değerini sağlayan kayıtlar raporlanır. Eşik değeri kullanıcı tarafından ekrandan belirlenir. Herhangi bir belirlenme yapılmazsa eşik değeri 0.000 olarak alınır, bunun anlamı dosyadaki tüm kayıtların raporlanmasıdır.

Sorgu işlemcisi, sorgu cümlesinin özelliklerine, önceliği bulanık operatörlü sorgu cümlelerine verir. Daha sonra basit sorgu cümleleri gelir. Eğer sorgu cümlesi sabit değer içermiyorsa bulanık sorgu için gerekli fonksiyonlar devreye girer. Sorgu cümlesindeki alan için bulanık küme veritabanında bir arama yapılır eğer bulunursa ara eşleşme derecesi olarak alınır. Eğer bulunamazsa ara eşleşme derecesi seçenekler modülünden belirlenen methoda göre hesaplanır. Eğer formül seçeneği seçilmiş ise alanın aldığı değer formülde yerine koyulur ve bu işlem sonucunda bulunan değer ara eşleşme derecesi olarak alınır.

3.12.1 Enterpolasyon :

```
PARAMETERS KOD,DEG
PRIVATE DURUM,i,DEGVAL,SK,SB,SECIM
DURUM = .f.
DEGVAL = 0.000
SK = -1
SB = -1
SECIM = BSORGU.ENTERPOLAS
SELECT BNDET
GO TOP
SCAN FOR ALLT(BNDET.DSQ)==kod AND ALLT(BNDET.DVAL)==DEG
    DURUM = .t.
ENDSCAN
IF DURUM = .f.
    CALCULATE CNT() FOR ALLT(BNDET.DSQ)==kod TO cntx
    DIMENSION Arrx(cntx)
    i = 1
    GO TOP
    DO While !EOF()
IF ALLT(BNDET.DSQ)=kod
    Arrx(i) = INT(VAL(BNDET.DVAL))
    i = i + 1
ENDIF
Skip
ENDDO
```

```

=ASORT(Arrx)
For j=1 to cntx
If Arrx(j) < INT(VAL(DEG))
SK = Arrx(j)
EndIf
EndFor
For k=cntx to 1 STEP -1
    If Arrx(k) > INT(VAL(DEG))
        SB = Arrx(k)
    EndIf
EndFor
ENDIF
if SK > 0 and SB > 0
SCAN FOR INT(VAL(BNDET.DVAL)) = SK
    STORE DMEM TO SKVAL
ENDSCAN
SCAN FOR INT(VAL(BNDET.DVAL)) = SB
    STORE DMEM TO SBVAL
ENDSCAN
DO CASE
    CASE SECIM = 1
        DEGVAL = SKVAL+((SBVAL-SKVAL)/(SB-SK))*(INT(VAL(DEG))-SK)
    CASE SECIM = 2
        DEGVAL = (( ( INT(VAL(DEG)) - SB ) / ( SK - SB ))
* SKVAL ) + ;
        (( ( INT(VAL(DEG)) - SK ) / ( SB - SK )) * SBVAL )
OTHERWISE
=MESSAGEBOX('Enterpolasyon başarı ile uygulanamadı .',16,'Dikkat
!..')
RETURN
ENDCASE
ELSE
    DEGVAL = 1.000
ENDIF
RETURN DEGVAL

```

Enterpolasyon seçeneği işaretlendi ise belirlenen enterpolasyon yöntemine göre enterpolasyon uygulanır ve bulunan değer ara eşleşme derecesi olur. Tüm bu işlemler ve seçenekler gözden geçirilip ara eşleşme derecesi bulunduğunda sorgu cümlesinde değiştirici ve değilleme olup olmadığı kontrol edilir. Değiştirici var ise değiştiriciler veritabanından cümledeki değiştiricinin etki değeri alınır ve daha önce elde edilmiş olan ara eşleşme derecesine uygulanır. Değilleme varsa elde edilen ara eşleşme derecesi 1 den çıkartılarak değilleme uygulanmış olur. Bu işlem programda aşağıdaki gibi kodlanmıştır.

```

IF Test_Arr(xi,14) <> 1.000          &&
    Mem    = AraMem**Test_Arr(xi,14)  && Değiştirici varsa
ELSE                                     && değiştirici uygula...
    Mem    = AraMem                    &&
ENDIF                                   &&
IF Test_Arr(xi,8) = .t.                &&
    Mem = 1 - Mem                      && Değilleme varsa hesapla...
ENDIF                                  &&

```

Yukarıda bahsedilen işlemler dizisi tüm altsorgu cümleleri için yapılır. Daha sonra VE ile bağlı olanlar cümlelerin kısmi eşleşme dereceleri bulunur. Sonuç olarak sorgu cümlesi VEYA bağlaçları ile bağlanmış tek bir sorgu cümlesine dönüşür. Son olarak toplam eşleşme derecesi bu cümleden bulunur.

```

For xj = 1 To VeyaLen
    Tot1= '1.000'
    f = 1
    Do While f <= VeLen
        if VeyaArr(xj,2) = VeArr(f,3)
            Tot1 = Tot1+', '+ Allt(Str(VeArr(f,4),5,3))
        EndIf
        f = f+1
    EndDo
    VeyaArr(xj,3) = Min(&Tot1)
EndFor
Tot2 = '0.000'
For zi = 1 to VeyaLen
    Tot2 = Tot2+', '+Allt(Str(VeyaArr(zi,3),5,3))
EndFor
ED = Max(&Tot2)

```

Bu hesaplamalar veritabanındaki tüm kayıtlar için yapılır. Yukarıda da anlatıldığı gibi eşik değerini sağlayanlar raporlanır.

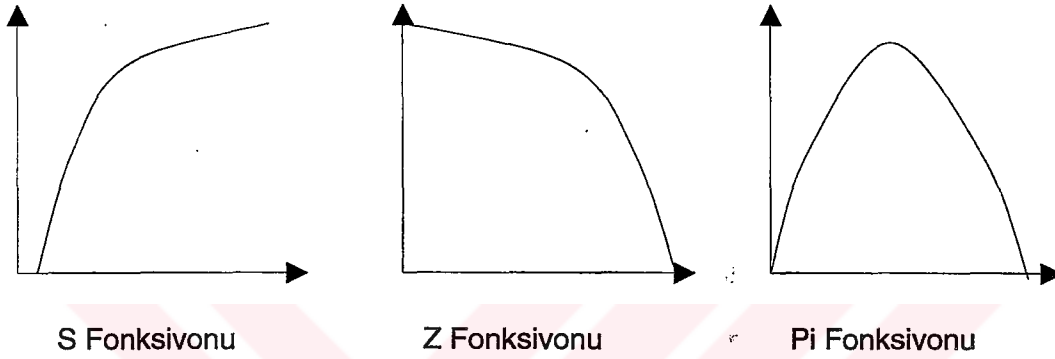
```

IF ED >= Thisform.iotoolbar.eşik.Value
oItem = Thisform.pg.p1.oList1.ListItems.Add ;
(,,Allt(Str(ED,5,3)),1,8)
    SELECT YeniDosya
    lCnt = 1
    DO while lCnt<=20 and !EMPTY(RapArr(lCnt))
        oItem.SubItems(lCnt) = &RapArr(lCnt)
        lCnt = lCnt + 1
    ENDDO
    StatuX = .f.
ENDIF

```

3.12.2 Fonksiyonların Belirlenmesi

Sistem bulanık kümelerin üyelik fonksiyonlarının belirlenmesinde bulanık mantık ve bulanık küme teorisinin çok kullanıldığı bir yöntemle üç adet standart fonksiyon kullanır, bunlar bulanık kümelerin belirlenmesi sırasında kullanıcı tarafından bulanık kümenin özelliğine uygun olarak tanımlanır. Bu fonksiyonlar S, Z ve Pi fonksiyonlarıdır. Fonksiyonların tanımları aşağıdaki gibidir.

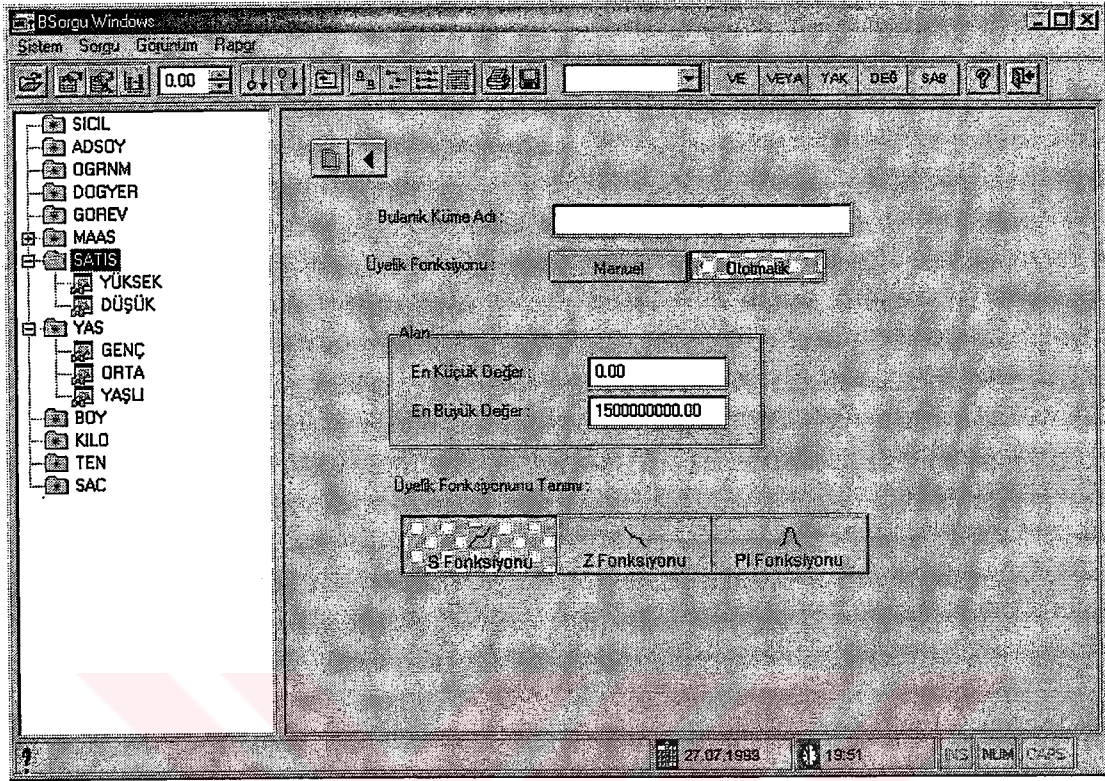


Şekil 3.12.2.1 Fonksiyon grafikleri

$$S(x; \alpha, \beta, \gamma) = \begin{cases} 0 \rightarrow x \leq \alpha \\ 2 \left(\frac{(x-\alpha)}{(\alpha-\beta)} \right)^2 \rightarrow \alpha \leq x \leq \beta \\ 1 - 2 \left(\frac{(x-\gamma)}{(\gamma-\alpha)} \right)^2 \rightarrow \beta \leq x \leq \gamma \\ 1 \rightarrow \alpha \geq \gamma \end{cases} \quad (3.12.2.1)$$

$$Z(x; \alpha, \beta, \gamma) = 1 - S(x; \alpha, \beta, \gamma) \quad (3.12.2.2)$$

$$\Pi(x; \beta, \gamma) = \begin{cases} S\left(x; \gamma - \beta, \gamma - \frac{\beta}{2}, \gamma\right) \rightarrow x \leq \gamma \\ Z\left(x; \gamma, \gamma + \frac{\beta}{2}, \gamma + \beta\right) \rightarrow x \geq \gamma \end{cases} \quad (3.12.2.3)$$



Şekil 3.12.2.2 Fonksiyon tanımları ekran kesiti

Bu fonsiyonlar sisteme Şekil 3.12.2.2 de gösterildiği gibi veritabanı alanlarına ait bulanık kümeler tanımlanırken bir defaya mahsus olmak üzere belirlenir. Fonksiyonların sınırları ve parametreleri otomatik olarak veritabanındaki ilgili alanın içeriğinden alınır.

3.13 Sorgu İşleminin Sonuçlandırılması

Program tüm aşamaları tamamladıktan sonra, araç çubuğundaki “sorgu” düğmesine basılması ile sorgu işlemcisini çalıştırır ve veritabanındaki kayıtları eşleşme dereceleri ile ekrana raporlar. Eğer eşik değeri olarak bir sayı belirlenmiş ise raporlama bu kritere göre yapılır. Sorgu sonuçları eşleşme derecelerine artan yada azalan sıralama yapılarak görülebilir. Raporlama işlemi şekil 3.13.1 deki ekran kesitinde görülmektedir.

BSorgu Windows

Sistem Sorgu Görünüm Rapor

0:00

VE VEYA YAK DES SAB

SICIL

ADSOY

OGRNM

DOGYER

GOREV

MAAS

AZ

ORTA

YETERLI

SATIS

YAS

YASLI

GENC

ORTA

BOY

KILO

ZAYIF

TEN

SAC

MAAS ÇOK AZ VE YAS GENÇ

Uyelik Drc	ADSOY	MAAS	YAS
0.872	NIHAL MEMIS	30000000	22
0.752	YASEMIN OKYANUS	35000000	23
0.752	DILEK YENILMEZ	35000000	24
0.752	ALI KIRAL	35000000	22
0.752	OYA TANYOLAC	35000000	26
0.813	AYSE TAVADOGLU	42500000	25
0.813	MUKADDES BAYAZIT	42500000	30
0.588	CEM GOKCE	45000000	26
0.584	MUSTAFA YENIGUN	45500000	32
0.673	NURCIHAN SAHIN	46500000	24
0.558	SERIFE CELIKTAS	48000000	24
0.490	FUNDA CINCI	55000000	28
0.490	CUMHUR ONEN	55000000	31
0.490	MEHMET SELIMOGLU	55000000	29
0.460	TAYFUN PARPUCU	135000000	38
0.430	SEFER GUGLU	35000000	39
0.423	FILIZ TOPTAS	62500000	28
0.423	ILKER EMIROGLU	62500000	25
0.401	RUHI BERIK	65000000	28

SORGU TAMAMLANDI

25.05.1999

12:17

Şekil 3.13.1 Sorgu raporlama ekran kesiti.

SONUÇLAR VE ÖNERİLER

Bu tez çalışmasında sıradan veritabanlarında bulanık küme teorisi kullanılarak günlük dildekine yakın cümleler ile sorgu yapılabilen bir sorgulama aracı geliştirilmiştir.

Bulanık sorgulama günümüzde bilgi keşfi, veri madenciliği gibi YZ ye dayalı akıllı veri değerlendirme yöntemlerine yeni bir bakış açısı getirmiştir. Bu tezin geliştirilmesi sırasında bile dünyanın pek çok yerinde bu konu ile ilgili araştırmalar devam etmektedir.

İçinde bulunduğumuz yüzyılın sonlarının ve önümüzdeki yüzyılın en değerli teknolojisinin, bilgi teknolojisi olduğunu göz önünde bulundurursak dünyanın her yerinde yapılan araştırmaların zamana karşı bir yarış olduğunu söylemek yanlış olmaz.

Bu çalışmadan çıkartılan sonuçların ve geliştirilen uygulamanın bundan sonra ülkeizde bu konuda yapılacak olan çalışmalara ivme kazandırması umulmaktadır.

KAYNAKLAR

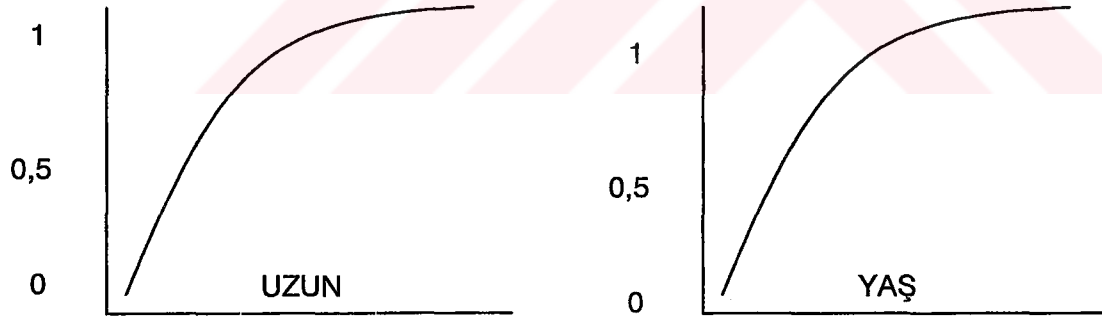
- [1] Andersen, T., Christiansen, H. and Larsen, H.L., 1997., Flexible Query Answering Systems, 45-61, 187-209, 247-277. Kluwer Academic Publishers,Boston.
- [2] Zadeh, L.A. and Kacprzyk, J., 1992. Fuzzy Logic For The Management of Uncertainty, 645-672 . Library of Congress Cataloging-in-Publication Data Press, New York.
- [3] Mutlu, T., 1996. Bulanık olmayan veritabanlarında bulanık sorgulama, *Yüksek Lisans Tezi*, İ.T.Ü. Fen Bilimleri Enstitüsü, İstanbul.
- [4] Kacprzyk, J. and Ziolkowski A., 1986. Database Queries with Fuzzy Linguistic Quantifiers, IEEE Transactions on Systems, Man, and Cybernetics, Vol SMC-16, No. 3 May/June 1986 . 474-478.
- [5] Takahashi, Y., 1991. A Fuzzy Query Language for Relational Databases. IEEE Transactions on Systems, Man and, Cybernetics. Vol 21. No 6. November-December 1991, 1576-1579.
- [6] Rasmussen, D. and Yager, R.R., 1996, SummarySQL – A Fuzzy Tool For Data Mining, Intelligent Data Analysis.

EKLER

EK A - BULANIK KÜMELER

Doğal diller belirsiz ve birden fazla anlama gelebilen kelimeler içerir. Özellikle pek çok sıfat bize kesin nicel bilgiler vermez. Örneğin uzun adam dediğimizde kesin olarak kimin uzun kimin uzun olmadığını belirtmiş olmayız. Benzer şekilde yaşlı adam tanımında da yaş kelimesinden dolayı bir belirsizlik vardır. Bu kelimeler bize yükseklik miktarı ve yaş miktarı ifade ettiği halde günlük yaşamda aynı katagoriye farklı yaştaki insanları sokabiliriz.

Şimdi , miktar olarak uzun ve yaşlının bize ne ifade ettiğine bakalım. 140 cm. ile 200 cm arasındaki yüksekliklerin uzunluk derecelerine μ diyelim . Yani belirlenen aralıktaki her x değerine karşılık bir μ ($0 \leq \mu \leq 1$) tanımlıyalım. Yatay eksene x dikey eksene μ dersek aşağıdaki grafiği elde edebiliriz. Benzer şekilde yaşlı kelimesini de nicel olarak ifade edebiliriz. Bu ifadeler subjektif olarak oluşturulmuştur. Fakat zaten bu kelimelerin anlamları kişiden kişiye değişebilir. Bulanık küme teorisinin önemli noktalarından biri budur.



Miktarla ilgili pek çok sıfat ve hatta *küçük* sayı, *yakın arkadaş* gibi soyut ifadeler bile nicelleştirilebilir. Bu durumda, miktarla bağlantı olmaksızın, küme elemanları yatay eksene yerleştirilir ve üyelik dereceleri belirlenir.

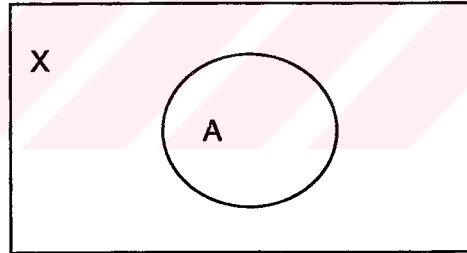
Bulanık küme teorisi, küme kavramı yardımıyla, anlamı kesin belirli olmayan kelimeleri nicel olarak ifade etmeye çalışır. Bulanık kümeler teorisi, küme teorisine bazı eklemeler yapılarak oluşturulduğundan, sıradan kümeler bulanık kümelerin özel bir şekli olarak görülür.

Bulanık küme teorisinde kullanılan temel notasyon şu şekildedir;

X	tüm küme
E	X 'in alt kümesi
$\emptyset$	boş küme
$\{0,1\}$	0 ve 1 den oluşan küme
$[0,1]$	0 ve 1 arasındaki reel sayılar kümesi
χ_E	E kümesinin karakteristik fonksiyonu
$a \wedge b$	a ve b 'den en küçüğü
$a \vee b$	a ve b 'den en büyüğü

Bulanık Küme : Bir X kümesini ele alalım. X kümesinde tanımlı bir A bulanık altkütmesi, $\forall x \in X$ için $\mu_A(x):X \rightarrow [0,1]$ üyelik fonksiyonu ile tanımlanır. X kümesine A 'nın *destek kümesi* denir.

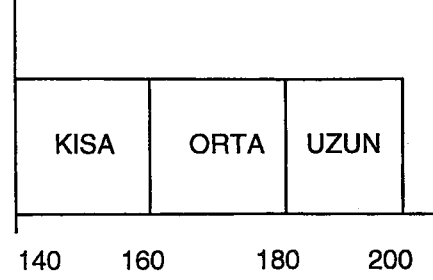
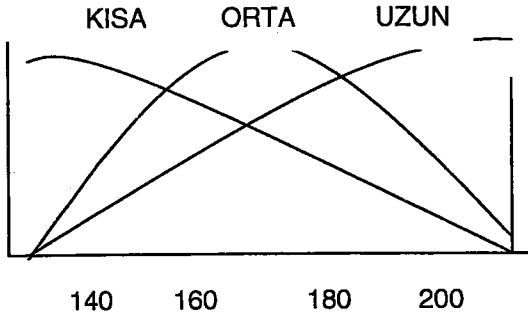
Bir X kümesinin bulanık altkütmesi A 'yı aşağıdaki gibi gösterebiliriz. Dikdörtgen çerçeve tüm kümeyi ve sınırları kesin belli olmayan daire bulanık altkümeyi temsil etmektedir. Bir bulanık küme daima başka bir kümenin alt kümesi olduğundan , bulanık altkümelerden , çoğu kaynakta olduğu gibi kısaca bulanık küme olarak bahsedeceğiz.



Bulanık A Kümesi

Bulanık kümelerle sıradan kümeler arasındaki fark üyelik fonksiyonunda kendini gösterir. X 'in sıradan altkütmesi olan bir E kümesini ele alalım. Aşağıdaki gibi tanımlanan $\chi_E(x)$ fonksiyonuna E 'nin karakteristik fonksiyonu denir.

Bu fonksiyon sıradan altkümenin üyelik fonksiyonuna karşılık gelir. Tanımdan da görüldüğü gibi $\chi_E(x):X \rightarrow \{0,1\}$ 'dir. Sıradan kümelerde bir nesne , ya kümeye dahildir yada değildir. Yani üyelik derecesi 0 yada 1 değerini alabilir. Bulanık kümelerde ise üyelik derecesi $[0,1]$ aralığında herhangi bir olabilir. Aşağıdaki şekillerde 140 cm ile 200 cm arasındaki yükseklikler kümesi üzerinde tanımlanan uzun , orta , kısa bulanık kümeleri ve bulanık olmayan küme farkı göstermektedir.



Bulanık Küme İşlemleri

Bulanık kümelerde de , sıradan kümelerdeki gibi en temel üç işlem birleşme , kesişme ve tülemedir. Bulanık kümeler üyelik fonksiyonları ile tanımlandıklarından , bu kümeler üzerindeki işlemler de üyelik fonksiyonları yardımıyla yapılır.

Birleşme : A ve B bulanık kümelerinin birleşimi $A \cup B$ kümesi aşağıdaki üyelik fonksiyonu ile tanımlanır.

$$\mu_{A \cup B}(x) = \mu_A(x) \vee \mu_B(x)$$

Birleşme işlemi $\vee: [0,1] \times [0,1] \rightarrow [0,1]$ bir fonksiyondur ve bazı özellikleri aşağıdaki gibidir .

$$a \vee 0 = a$$

$$a \vee b = b \vee a$$

$$a \vee b \geq c \vee d, a \geq c, b \geq d \text{ ise}$$

$$a \vee b \vee c = a \vee (b \vee c) = (a \vee b) \vee c$$

Tanımından anlaşıldığı gibi A ve B kümelerinin birleşimi hem A hemde B 'yi içeren en küçük bulanık kümedir. Daha açık bir ifadeyle , eğer D bir bulanık küme ise ve A ve B 'yi içeriyorsa $A \cup B$ 'yide içerir. Birleşme işlemi semantik olarak VEYA'YA karşılık gelir.

Kesişme : A ve B bulanık kümelerin kesişimi $A \cap B$ kümesi aşağıdaki üyelik fonksiyonu ile tanımlanır.

$$\mu_{A \cap B}(x) = \mu_A(x) \wedge \mu_B(x)$$

Kesişme işlemi de birleşme özelliğine sahiptir ve bize hem A hemde B 'yi içeren en büyük bulanık kümeyi verir. Kesişme işlemi semantik olarak VE'ye karşılık gelir. Birleşme işlemine benzer şekilde , a) $a \wedge 1 = a$ 'dır ve diğer b-d şıklarındaki özellikleri sağlar.

Tümleme : A kümesinin tümleyeni $\bar{A}$ aşağıdaki üyelik fonksiyonu ile tanımlanır.

$$\mu_{\bar{A}}(x) = 1 - \mu_A(x)$$

Tümleme işlemi semantik olarak *DEĞİL* 'e karşılık gelir. Bulanık kümelere ait diğer bazı ilişki ve özellikler şöyledir;

Boş küme : Eğer bir A bulanık kümesinin üyelik fonksiyonu $\mu_A(x) \equiv 0$ ise A bulanık kümesi boştur denir.

Normal Bulanık Küme : bir A bulanık kümesinin üyelik fonksiyonu en az bir eleman için 1 değerini almışsa A normal bulanık kümedir .

Eşitlik : Eğer $x \in X$ için $\mu_A(x) = \mu_B(x)$ ise A ve B bulanık kümeleri eşittir denir ve $A = B$ şeklinde gösterilir.

Kapsama : Eğer $x \in X$ için $\mu_A(x) \subseteq \mu_B(x)$ ise B bulanık kümesi A bulanık kümesini kapsar denir ve $A \subset B$ şeklinde gösterilir.

Çifte Değilleme : Bir bulanık kümenin tümleyeninin tümleyeni yine kendisidir .

$$(A')' = A$$

De Morgan Kuralları :

$$(A \cup B)' = A' \cap B'$$

$$(A \cap B)' = A' \cup B'$$

Bu bağlamda ,

$$\mu_X(x) = 1$$

$$\mu_a(x) = 0$$

Bulanık kümelerin üyelik fonksiyonları ile tanımlandığını söylemiştik. Fakat destek kümesi eğer $X = \{x_1, x_2, x_3, \dots, x_n\}$ şeklinde sonlu sayıda eleman içeriyorsa , X 'in A bulanık altkümesi şu şekilde ifade edilebilir.

$$A = \sum_{i=1}^n \mu_A(x_i) / x_i = \mu_A(x_1) / x_1 + \mu_A(x_2) / x_2 + \dots + \mu_A(x_n) / x_n$$

Bu gösterimde üyelik derecesi 0 olan elemanlar ihmal edilir. + simgesi $\vee$ yani *veya* işlemine karşılık gelir . Örneğin,

$$a/x_1 + b/x_1 = a \vee b / x_1$$

İki bulanık kümenin birleşimini , kümeleri + ile bağlayarak bulabiliriz. Örneğin 1 ile 10 arasındaki tamsayılar kümesi için *büyük* ve *orta* şeklinde iki bulanık kümeyi aşağıdaki gibi tanımlayalım.

$$büyük = 0.2/5+0.4/6+0.7/7+0.9/8+1/10$$

$$orta = 0.4/3+0.8/4+1./5+0.8/6+0.4/7$$

Bu kümelerin birleşimi büyük $\cup$ orta ,

$$Büyük \cup orta = 0.4/3+0.8/4+1/5+0.8/6+0.7/7+0.9/8+1/10$$

Eğer destek kümesi sonsuz sayıda eleman içeriyorsa eşitliği aşağıdaki gibi genelleştirilebilir .

$$A = \int_X \mu_A(X) / X$$

Bu ifade yöntemi kullanılırsa kesişme , birleşme ve tümlene işlemleri aşağıdaki gibi ifade edilebilir.

$$A \cup B = \int_X (\mu_A(x) \vee \mu_B(x)) / x$$

$$A \cap B = \int_X (\mu_A(x) \wedge \mu_B(x)) / x$$

$$A' = \int_X (1 - \mu_A(x)) / x$$

Yukarıdaki işlemlere ek olarak , bulanık kümelerin çeşitli kombinasyonlarını oluşturan veya birbirleriyle ilişkilendirilen pekçok işlem tanımlanabilir. Bunlardan en önemli olanları inceleyelim.

Cebrik Çarpım : A ve B bulanık kümelerinin cebrik çarpımı $A . B$ şeklinde gösterilir ve aşağıdaki üyelik fonksiyonu ile tanımlanır.

$$\mu_{A.B}(x) = \mu_A(x) \mu_B(x)$$

Görüldüğü gibi $A . B \subset A \cap B$.

Cebrik Toplam : A ve B bulanık kümelerinin cebrik toplamı $A \oplus B$ şeklinde gösterilir ve aşağıdaki üyelik fonksiyonu ile tanımlanır.

$$\mu_{A \oplus B}(x) = \mu_A(x) + \mu_B(x) - \mu_A(x) \mu_B(x)$$

Doğrudan Çarpım : A ve B sırasıyla X ve Y kümelerinin bulanık altkümeleri ve $x \in X$ ve $y \in Y$ için , A ve B'nin doğrudan çarpımı $A \times B$ şeklinde gösterilir ve aşağıdaki üyelik fonksiyonu ile tanımlanır .

$$\mu_{A \times B}(x) = \mu_A(x) \wedge \mu_B(x)$$

Sınırlı Toplam : A ve B bulanık kümelerinin sınırlı toplam $A+B$ şeklinde gösterilir ve aşağıdaki üyelik fonksiyonu ile tanımlanır. Görüldüğü gibi sınırlı toplam $\forall x \in X, \mu_A(x) + \mu_B(x) \leq 1$ şartı sağlandığında anlamlıdır.

$$\mu_{A+B}(x) = (\mu_A(x) + \mu_B(x)) \wedge 1$$

Sınırlı Fark : A ve B bulanık kümelerinin sınırlı farkı $A \oplus B$ şeklinde gösterilir ve aşağıdaki üyelik fonksiyonu ile tanımlanır.

$$\mu_{A \oplus B}(x) = (\mu_A(x) + \mu_B(x)) \vee 0$$

Mutlak Fark : A ve B bulanık kümelerinin mutlak farkı $|A - B|$ şeklinde gösterilir ve aşağıdaki üyelik fonksiyonu ile tanımlanır.

$$\mu_{|A-B|}(x) = |\mu_A(x) - \mu_B(x)|$$

Mutlak fark, sıradan kümelerde $A \cup B$ 'nin $A \cap B$ 'deki izafi tümleyene indirgenir.

Konveks Kombinasyon : İki f ve g vektörünün konveks kombinasyonu $0 \leq \lambda \leq 1$ olmak üzere $\lambda f + (1 - \lambda)g$ şeklinde lineer bir kombinasyon olarak tanımlanır.[3] Bu kombinasyon bulanık kümelerde uygulanabilir.

A, B ve Λ herhangi üç bulanık küme olsun. A, B ve Λ nın konveks kombinasyonu (A, B, Λ) şeklinde gösterilir ve aşağıdaki ilişki ile tanımlanır.

$$(A, B, \Lambda) = \Lambda A + \Lambda' B$$

Üyelik fonksiyonu ile ifade edersek ,

A, B ve Λ 'nın konveks kombinasyonun temel özelliği şu şekilde ifade edilebilir .

$$A \cap B \subset (A, B; \Lambda) \subset A \cup B$$

Bu özellik üyelik fonksiyonları cinsinden bize aşağıdaki eşitsizliği verir .

$A \cup B \subset C \subset A \cup B$ şartını sağlayan bir C bulanık kümesi için daima $C = (A, B; \Lambda)$ olacak şekilde bir Λ bulanık kümesi bulunabilir . Bu kümenin üyelik fonksiyonu aşağıdaki gibidir.

$$\mu_{\Lambda}(x) = \frac{\mu_C(x) - \mu_B(x)}{\mu_A(x) - \mu_B(x)}$$

Bulanık Kümelerde Konvekslik

Üyelik fonksiyonu ... olan bir A bulanık kümesi ele alalım . Eğer aşağıdaki gibi tanımlanmış Γ_α kümeleri $(0, 1]$ aralığında her α için konveks ise A bulanık kümesi de konvektir.

$$\Gamma_\alpha = \{x \mid \mu_A(x) \geq \alpha\}$$

Daha başka bir ifade ile $\forall x_1, x_2 \in X, \lambda \in [0,1]$ için ,

$$\mu_A(x)[\lambda x_1 + (1 - \lambda)x_2] \geq \mu_A(x_1) \wedge \mu_A(x_2)$$

2.4 Bulanık Kümenin α Kesimi

Bir A bulanık kümesi için ,

$$A_\alpha = \{x \mid \mu_A(x) > \alpha, \alpha \in [0,1]\}$$

$$A_{\bar{\alpha}} = \{x \mid \mu_A(x) \geq \alpha, \alpha \in (0,1]\}$$

kümelerine sırasıyla zayıf α kesimi ve kuvvetli α kesimi denir. [2] α kesimi anlam olarak , üyelik derecesi belirli bir sınırın üstünde olan nesnelerin kümesidir. Eğer üyelik fonksiyonu sürekliyse , kuvvetli ve zayıf olarak iki kesimin tanımlanmasına gerek duyulmadığından doğrudan zayıf α kesimi kullanılır.

Bir bulanık kümenin üyelik fonksiyonunu α - kesimi yardımıyla ifade edebiliriz. Şöyleki , zayıf ve kuvvetli α kesimleri için , A_α 'nın karakteristik fonksiyonuna $\chi_{A_\alpha}(x)$ dersek,

$$\mu_A(x) = \sup_{\alpha \in (0,1]} [\alpha \wedge \chi_{A_\alpha}(x)] \text{ olur.}$$

Bulanık Sayılar

Normal ve konveks bir A bulanık kümesinin α -düzey kümesi kapalı bir aralık ise , bu aralık bulanık sayı olarak adlandırılır.[2] Bulanık sayılar aşağıdaki şekildeki gibi üyelik fonksiyonlarına sahiptirler. Üyelik fonksiyonu sürekli ise α kesimi zaten kapalı bir aralıktır. Fakat kapalı bir α kesimi elde etmek için üyelik fonksiyonunun sürekli olması sorun değildir.

$f : X \rightarrow Y$ bir fonksiyon olsun . X 'in bulanık altkümeleri A'ya karşılık Y'de aşağıdaki üyelik fonksiyonu ile tanımlı bir $f(A)$ bulanık kümesi elde edebilir .

$$\mu_{f(x)}(x) = \sup_{y=f(x)} \mu_A(x)$$

Verilen bir y için eşitliğin sağ tarafı $y = f(x)$ olacak şekilde ... en büyük değerini alacağımızı ifade eder. $f(A)$, A 'nın f içinden görüntüsüdür. $f(A)$ doğrudan aşağıdaki gibi tanımlanabilir.

$$f(A) = \int_y \mu_A(x) / f(x)$$

Örneğin $(\chi) = 2x+1$ olsun $x = 3$ iken bu fonksiyon bize 7 değerini verir. Farzedelim ki x bir sayı değil, $[2,4]$ aralığı olsun. Bu durumda fonksiyon bize $[5,9]$ sayı aralığını verir. Şayet bir sayı yada sayı aralığı yerine *yaklaşık 3* yada *5 civarı* gibi bir bulanık sayı varsa,

$$f(3) = 2 \times 3 + 1 = 7$$

elde ederiz. 3 bulanık sayısını şekil 2.x'ten yararlanarak α - kesimi ile ifade edersek,

$f(3)$ 'ün üyelik fonksiyonunu bu sonuçlara göre çizersek şekil 2.x.b'yi elde ederiz.

Bulanık İlişkiler

Günlük yaşamda "x ve y aşağıyukarı eşit" veya "x y'den daha güzel" gibi belirsiz ilişkilerle sıklıkla karşılaşırız. Belirsiz kelimeleri geleneksel teorilerle ifade edilmenin güçlüğünden daha önce bahsedilmişti. Aynı zorluk bu çeşit ilişkilerle de karşımıza çıkmaktadır. Bu gibi belirsiz ilişkileri, sıradan ilişkilerin genelleştirilmiş şekli olarak ele alınan bulanık ilişkilerle ifade edebiliriz.

A ve B bulanık kümeleri arasında tanımlı bir R bulanık ilişkisi A ve B 'nin direk çarpımıdır, $A \times B = \{ (a, b) \mid a \in A, b \in B \}$, ve üyelik fonksiyonu aşağıdaki gibidir.[2]

$$\mu_R(a, b): A \times B \rightarrow [0,1]$$

Eğer $A = B$ ise R A üzerinde tanımlı bir bulanık ilişkidir. Genel olarak $X_1 \times X_2 \times X_3 \times \dots \times X_n$ çarpım uzayında n 'li bir R bulanık ilişkisi tanımlanabilir. Böyle bir ilişkinin üyelik fonksiyonu

$$\mu_R(x_1, x_2, x_3, \dots, x_n)$$

şeklindedir.

Bulanık İlişkiler Kompozisyonu : R , $X \times Y$ 'de $S, Y \times Z$ 'te tanımlı bulanık ilişkiler olsunlar. R ve S 'nin kompozisyonu $R \circ S$ $X \times Z$ 'te aşağıdaki gibi tanımlı bir bulanık ilişkidir.[2]

$$R \circ S \leftrightarrow \mu_{R \circ S}(x, z) = \bigvee_y \{ \mu_R(x, y) \wedge \mu_S(y, z) \}$$

EK -B

BULANIK MANTIK

Bulanık mantık , adından da anlaşılacağı gibi kesin çıkarımlardan çok yaklaşık çıkarımlar elde edilen mantıktır. Önergeler doğal dille ifade edildiğinde , çoğunlukla doğal dilin yapısı gereği belirsizlik taşırlar. Dolayısı ile bu çeşit önermelerden kesin çıkarım yapmak olanaklı olmaz. İnsanlar çoğu zaman kesin çıkarım yapmak yerine yaklaşık çıkarım yaparlar. Örneğin insanların boyunun kaç cm ve kaç mm olduğunu bilmeden ve ilgilenmeden , insanları uzun boylu , orta boylu , kısa boylu vs . gibi kümelerle ayırırlar. Yani insan düşüncüsü ve çıkarımı bulanıktır demek hemen hemen doğrudur.

- Bulanık mantığın bazı karakteristik özelliklerini şöyle sıralayabiliriz.[4]
- Bulanık mantıkta , kesin çıkarım yaklaşık çıkarımın özel bir hali olarak görülür.
- Herhangi bir mantıksal sistem bulanıklaştırılabilir.
- Bulanık mantıkta bilgi , bir değişkenler kümesi üzerinde tanımlı olan bulanık kısıtlar olarak yorumlanır.
- Çıkarım mekanizması , bulanık kısıtların sağlanması işlemi olarak görülür.

Bulanık mantık , geleneksel mantık sistemlerinden hem ana düşünüş şekli olarak hemde detayda farklılıklar gösterir. Bu farklılıkları şöyle özetleyebiliriz.[4]

Doğruluk (Truth) : İki değerli sistemlerde doğruluk sadece iki değer alabilir. Çok değerli sistemlerde ise doğruluk genellikle sonlu bir kümenin elemanıdır. Bulanık mantıkta doğruluk , verilen kümenin $[0,1]$ aralığındaki bir bulanık altkümesidir veya daha basitçe $[0,1]$ aralığında bir noktadır. Dilsel olarak tanımlanan *doğru* , *çok doğru* , *pek doğru değil* gibi doğruluk değerleri de birim aralıkta bulanık kümeler olarak ele alınır.

Yüklemler (*Predicates*) : İki değerli sistemlerde yüklemeler kesindir , örneğin *ölümlü* , *çift* , *daha büyük* vs. Dikkat edilirse doğal dildeki yüklemeler , kesin olmaktan çok bulanıktırlar.

Yüklem Değiştiriciler (*Predicate modifiers*) : Geleneksel sistemlerde kullanılan tek modifikatör *değil* değiştiricisidir. Bulanık mantıkta buna ek olarak pek çok değiştirici kullanılabilir. Örneğin *çok* , *aşağıyukarı* , *aşırı* vs. Bu gibi yüklem değiştiricileri çeşitli dilsel değişkenler oluşturulurken önemli bir rol oynarlar. Örneğin *çok genç* , *çok genç değil* vs.

Niteleyiciler (*Quantifiers*) : Geleneksel sistemlerde sadece iki niteleyen kullanılır, *her* ve *bazı*. Bulanık mantıkta pek çok niteleyici kullanılabilir. Örneğin *birkaç* , *çoğu* , *siksik* , *pekçok* vs. Bulanık mantıkta niteleyiciler bulanık küme olarak ele alınırlar.

Olasılık (*Probabilities*) : Geleneksel mantıksal sistemlerde olasılık bir sayı yada bir sayı aralığıdır. Bulanık mantıkta olasılık dilsel terimlerle ifade edilebilir.

Örneğin *hemen hemen* , *yaklaşık* vs. Bu gibi olasılıklar bulanık sayı olarak olarak ele alınır ve bulanık aritmetik kullanılarak hesaplamaya katılır. Pratikte genellikle bulanık olasılık içeren önermeler bulanık niteleyici içerecek bir eşdeğer forma getirebilir.

Olabilirlik (*Possibilities*) : Geleneksel mantık sistemlerinin tersine , bulanık mantıkta olabilirlik kavramı iki değerli olmak yerine derecelendirilmiştir. Örneğin *büyük ihtimal olabilir*, *hemen hemen imkansız* vs. Bu değerler bulanık kümeler olarak ele alınır .

Olasılık fonksiyonları bulanık mantıkta önemli bir yer tutar. Bir U uzayında tanımlı bir X değişkeninin dağılım fonksiyonu $\prod x$ ise $\prod x$ X 'in üyelik fonksiyonuna eşdeğerdir.

Bulanık mantık , görüldüğü gibi , klasik mantıksal sistemlere bazı ekler getirilerek elde edilir. Bu nedenle diğer mantıksal sistemlerle çelişen bir sistem değil onların bir uzantısı olarak görülmelidir.

Dilsel Değişkenler

Bulanık mantık ve yaklaşık çıkarıma ihtiyaç duyulmasının nedeni daha önce bahsettiğimiz gibi kesin belirli bir anlamı olmayan veya birden fazla anlama gelebilen dilsel değişkenlerle karşılaşmamızdır. Dilsel değişkenler belirli bir kümenin bulanık altkümeleri olarak ele alınır. Bir dilsel değişkeni $(\times T(x), U, G, M)$ ile karakterize edebiliriz. [5] Burada x dilsel değişkeni $T(x)$ bu değişkenle ilgili terim kümesini , U dilsel değişkenin tanımlı olduğu kümeyi , G x değişkeninden üretilen terimlerin yazım kurallarını ve M üretilen terimlerle ilgili semantik kuralları göstermektedir.

Örneğin X yıl olarak yaşı temsil eden bir dilsel değişken olsun. $U = [0,100]$ yani 0 ile 100 arasında değişen yaşlar kümesini ele alalım. Bir dilsel değişken için tanımlı terimler örneğin yaşlı , çok yaşlı , genç vs. olabilir. Bu terimlerden örneğin yaşlıya anlam yükleyen $Myaşlı = \{(x, \mu_{yaşlı}(x)) | x \in U\}$ şeklindedir. Burada $\mu_{yaşlı}(x)$ U ' da tanımlı yaş değişkeninin yaşlı bulanık kümesine üyelik derecesini belirten üyelik fonksiyonudur.

Ürettiğimiz dilsel terimleri yine dilsel değiştiricilerle anlamlarını değiştirip yeni bulanık kümeler elde edebiliriz. Eğer A kümesi m değiştiricisi ile değiştirilmişse yeni bir $B = m(A)$ terimi veya bulanık kümesi elde edilir.

Değiştiriciler için sıkça kullanılan ve kabul görmüş bazı matematiksel modellerden yoğunlaştırma , biraz yoğunlaştırma ve yayma modelleri sırasıyla aşağıdaki gibidir.[4]

$$\mu_{yog}(A)(x) = (\mu_A(x))^2$$

$$\mu_{bir}(A)(x) = (\mu_A(x))^{1.25}$$

$$\mu_{yay}(A)(x) = (\mu_A(x))^{1/2}$$

Bu modelleri kullanarak şu deęiřtircileri oluřturabiliriz. A bulanık kumesi iin ,

$$\text{OK } A = \text{yog}(A)$$

$$\text{AřAđIYUKARI } A = \text{yay}(A)$$

$$\text{BİRAZ } A = \text{bir}(A)$$

$$\text{HAFİFE } A = (\text{BİRAZ } A \text{ VE OK DEđİL})$$

Bulanık Önergeler

Bulanık yüklem ieren önergeler bulanık önerme denir. Örneđin " Ayře *ince* yapılıdır" veya "*x küçük* bir sayıdır" gibi. Bu örneklerdeki bulanık deęiřkenler italik karakterle belirtilmiřtir. Genel olarak ,

$$X F' \text{ dir.}$$

řeklinde gösterilirler. Burada X bir nesneler kumesi ve F bulanık bir yüklem ve *bulanık deęiřken* olarak ta adlandırılır. Bulanık deęiřkenler *dilsel deęiřken* olarak ta adlandırılır ve bulanık kümeler yardımıyla ifade edilirler. Her $x \in X$ iin $x F' \text{ dir}$ önergelerinin dođruluđu bilirse önermenin dođruluđu saptanabilir. $X = \{....\}$ ise,

$$\text{dođruluk}(x F' \text{ dir}) = \mu$$

ve önermenin dođruluk derecesi r , dir.

Bulanık küme herhangi bir m deęiřtircisi ile deęiřtirilebilir. Deęiřtirciler daha önce deđindiđimiz gibi bulanık kümeyi deęiřtirerek yeni bir bulanık küme ortaya çıkarılır. Dolayısıyla dođruluk hesapları yukarıdaki gibi yapılabilir. Bir m deęiřtircisi ile deęiřtirilmiř bir önermenin genel yapısı ise,

$$X mF' \text{ dir.}$$

řeklinde dir. Örneđin "*x ok küçük* bir sayıdır" gibi. Deęiřtirciler de bulanık deęiřkenler gibi subjektiftir, genel bir formları yoktur. Fakat *deđil* deęiřtircisi objektif bir deęiřtirici olarak deđerlendirilebilir.

Önermenin nesnesi bir Q niteleyicisi ile nitelenebilir. İki deđerli mantıkta kullanılan *her* ve *bazı* niteleyicilerinin yanında bulanık mantıkta ok eřitli niteleyiciler tanımlanıp kullanılabilir. Örneđin *ođu*, *birkaçı* vs. Böyle bir cümlenin genel yapısı,

$$QX mF' \text{ dir.}$$

řeklinde gösterilebilir. Örneđin "*ođu ocuklar ok yaramazdır*" gibi. Niteleyicilerin bulanık kümeler olarak ele alındıđını söylemiřtik. Bu tip önergelerin dođruluklarını saptamak iin önce $X mF' \text{ dir}$ önermesinin dođruluk derecesi r hesaplanır. Önermenin dođruluk derecesi r' ise,

$$r' = \mu_Q(r)$$

şeklindedir.

Birden fazla önerme *ve* yada *veya* mantıksal operatörleriyle bağlanırsa aşağıdaki önermeler elde edilir.

$$X \text{ A'dır } \text{ ve } X \text{ B'dir}$$

$$X \text{ A'dır } \text{ veya } X \text{ B'dir}$$

Örneğin “ x çok büyük bir sayı değildir, x orta büyüklükte bir sayıdır” önermesini ele alalım. *Büyük* ve *orta* bulanık kümeleri belirli destek kümelerinde her x için tanımlı olsun ve *çok* değıştiricisini üyelik derecesinin karesi olarak tanımlamış olalım. Bu durumda önermenin doğruluk derecesi,

$$(1 - \text{BÜYÜK}^2) \cap \text{ORTA}$$

şeklindedir. [2] Fakat bileşik bulanık önerme farklı dilsel değışkenlere ait bulanık kümeler içeriyorsa, örneğin *uzun* ve *ağır* gibi, doğruluk hesabını bu şekilde yapamayız. Böyle bir durumda bulanık ilişkileri kullanmamız gerekir. [2] Örneğin,

$$X \text{ C'dir } \text{ ve } Y \text{ D'dir} = (x, y) \in C \times D \text{ dir}$$

$$X \text{ C'dir } \text{ veya } Y \text{ D'dir} = (x, y) \in C \times Y \cup X \times D \text{ dir.}$$

ÖZGEÇMİŞ

Bahadır ASAR 1972 yılında İstanbulda doğdu. İlköğrenimini Mersin Çankaya ilkokulunda tamamladı. Ortaöğrenimini İstanbul Göztepe Ortaokulunda ve lise eğitiminde İstanbul İntaş Mehmet Akif Ersoy Lisesinde tamamladı. 1990 yılında Doğu Akdeniz Üniversitesi Fen Bilimleri Enstitüsü Uygulamalı Matematik ve Bilgisayar Bilimleri bölümüne girdi. 1995 yılında İstanbul Teknik Üniversitesi Fen Bilimleri Enstitüsü Mühendislik Bilimleri Anabilim Dalı, Sistem Analizi Programı'nda yüksek lisansa başladı.