KOCAELİ ÜNİVERSİTESİ * FEN BİLİMLERİ ENSTİTÜSÜ

ÖĞRENCİ BİLGİ SİSTEMİNDE VERİ MADENCİLİĞİNİN UYGULANMASI

YÜKSEK LİSANS TEZİ

Bilgisayar. Müh. Umut ALTINIŞIK

Anabilim Dalı : Elektronik ve Bilgisayar Eğitimi

Tez Danışmanı : Prof. Dr. Kadir ERKAN

MAYIS 2006

KOCAELİ ÜNİVERSİTESİ * FEN BİLİMLERİ ENSTİTÜSÜ

ÖĞRENCİ BİLGİ SİSTEMİNDE VERİ MADENLİĞİNİN UYGULANMASI

YÜKSEK LİSANS TEZİ

Bilgisayar. Müh. Umut ALTINIŞIK

Tezin Enstitüye Verildiği Tarih : 26 / 05 / 2006

Tezin Savunulduğu Tarih : 09 / 06 / 2006

Tez Danışmanı	Üye	Üye
Prof. Dr. Kadir ERKAN	Yrd. Doç. Dr. Ferdi BOYNAK	Yrd. Doç. Dr. Mehmet YILDIRIM
(.....)	(.....)	(.....)

MAYIS 2006

ÖĞRENCİ BİLGİ SİSTEMİNDE VERİ MADENCİLİĞİNİN UYGULANMASI

Umut ALTINIŞIK

Anahtar Kelimeler : Veri Madenciliği, Birliktelik Kuralları, Apriori Algoritması, Öğrenci Bilgi Sistemi, Ders Birliktelikleri.

Özet : Günümüzde veritabanlarında muhafaza edilmekte olan verilerin değerlendirilerek her hangi bir konuda anlamlı sonuçlara ulaşılması büyük bir önem kazanmaktadır. Bu süreç içinde Veri Madenciliği en önemli safhalardan birisidir.

Birliktelik kuralları Veri Madenciliği tekniklerinden birisidir. Birliktelik kuralları içerisinde en yaygın olarak kullanılanı ve en bilineni Apriori algoritmasıdır.

Bu tez çalışmasında, Kocaeli Üniversitesi ÖBS (Öğrenci Bilgi Sistemi) üzerinde Apriori algoritması kullanılarak veri madenciliği uygulaması gerçekleştirilmiştir. Veritabanı olarak Microsoft YSD (Yapısal Sorgu Dili) 2000 ve uygulama ara yüzü olarak ise DELPHİ 6.0 kullanılmıştır.

Bu çalışma sonucunda, öğrencilerin başarısız oldukları dersler değerlendirilerek bu dersler arasında birliktelikler oluşturulmuştur. Bu birliktelikler kullanılarak derslerin birbirleri ile bağlantıları ortaya konulmuştur. Böylece derslerin bağıntı kurallarına göre başarısız öğrencilerin ders seçiminde danışmana rehber olabilecek bir uygulama programı geliştirilmiştir.

A DATA MINING APPLICATION on A STUDENT INFORMATION SYSTEM

Umut ALTINIŞIK

Keywords : Data Mining, Association Rules, Apriori Algorithm, Student Information System, Association of Lessons

Abstract : Nowadays, it is very important that to reach meaningful results from evaluating data which is kept by databases. In this process, data mining is the most important phase.

Associative Rules is one of the techniques for data mining. Apriori is the most important and common technique in associative rules.

On the thesis, data mining application is executed by using Apriori on Kocaeli University Student Information System. Microsoft SQL (Structured Query Language) is used for database and DELPHİ 6.0 as an application interface.

On the result of this study, associations is made between lessons for evaluating unsuccessful students. The relationship is created by using associtions of lessons Therefore, on the result of relationship rules, an application was built up for unsuccessful student on advisor guide.

ÖNSÖZ

Dünya üzerinde toplanmakta olan veri miktarı büyük bir hızla her geçen gün artmaktadır. Bu süreç içinde işlenmemiş olan bu verilerin kullanımı büyük bir önem taşımaktadır. Günümüzde, Veri Madenciliği konusu hayatın her alanına nüfus etmektedir. Kamu kurumları, ticari ve kar amacı gütmeyen işletmelerde muhafaza edilen veriler işlenerek değerli bir kaynak haline getirilmektedir.

Veri Madenciliğinin gelişmesi ile birlikte çeşitli Veri Madenciliği teknikleri ortaya çıkarılmıştır. Bu tekniklerden birliktelik kuralları tekniği kullanılarak, üniversitemiz bünyesinde kullanılmakta olan Öğrenci Bilgi Sistemi (ÖBS) üzerinde madencilik uygulaması yapılmıştır.

Bu tez çalışması süresince her türlü desteğini esirgemeyen danışmanım Prof. Dr. Kadir Erkan'a teşekkürlerimi sunarım.

Ayrıca, ÖBS' nin kullanılması aşamasında yardımlarını esirgemeyen başta Okt. Mustafa GÜNDOĞDU olmak üzere, diğer tüm arkadaşlarıma teşekkür ederim.

İÇİNDEKİLER

SİMGELER DİZİNİ ve KISALTMALAR

ÖBS	: Öğrenci Bilgi Sistemi
SIS	: Student Information System
YSD	: Yapısal Sorgu Dili
SQL	: Structured Query Language
SC	: Stock Chain
VM	: Veri Madenciliği
VTBK	: Veri Tabanlarında Bilgi Keşfi
T	: Veri tabanındaki her bir kayıt
I	: Öğe kümeleri
D	: Veri tabanı
TID	: Transaction ID
DHP	: Direct Hashing and Pruning
DIC	: Dynamic Itemset Counting
OCA	: Online Combinatorial Approximation
CRM	: Customer Relation Management
AIS	: Artificial Immune Systems
SETM	: Set-Oriented Mining
CART	: Classification and Regression Trees
CHAID	: Chi-square Automatic Interaction Detector
SEAR	: Sequential Efficient Association Rule
ECLAT	: Equivalence Class-Based
PEAR	: Partitioned Efficient Association Rule

ŞEKİLLER DİZİNİ

TABLOLAR DİZİNİ

BÖLÜM 1. GİRİŞ

Günümüz teknolojilerinde verilerin toplama ve saklama işlemleri bir sorun olmaktan çıkmıştır. Dünyadaki bilgi miktarı her geçen gün katlanarak artmakta ve bu verilerin belirli bir süzgeçten geçirilip özümsendikten sonra çeşitli alanlarda kullanılmak üzere hazırlanması ihtiyacı ortaya çıkmaktadır.

Veriler belirli bir amaca yönelik işlenmediği sürece herhangi bir değer içermezler. Verinin işlenerek bilgi haline getirilmesi işlemini veri analizi olarak adlandırabiliriz.

Veri Madenciliği, büyük ölçekli veriler arasından anlamlı örüntülerin keşif ve analiz edilmesi sürecidir (Bery and Linoff 1997). Veri madenciliği, büyük miktarlardaki veriler içerisinden gelecekle ilgili tahmin yapmamızı sağlayacak bağıntı ve kuralların bilgisayar programları kullanarak aranması olarak da tanımlanabilir (Alpaydın 2000).

Olasısal veri nedenleşmede veri madenciliği, istatistik alanındaki bir çok metodu kullanmasına rağmen, nesnelerin nitelik değerlerine bağlı çıkarsama yapmada bilinen statiksel metotlardan ayrılmaktadır (Ziarko 1991, Elder and Pregipon 1995).

Tez uygulamasında da kullanılan ve ilk olarak Agrawal et al'da (1993) ortaya çıkartılmış olan birliktelik kuralları veri madenciliğinde en çok tercih edilen tekniklerden birisidir.

Birliktelik kurallarının çıkarımı problemi ilk olarak 1993 yılında Market Sepet Analizi (Market Basket Analysis) ile verilerin eşleştirilmesiyle yapılmıştır (Agrawal et al 1993). Birliktelik sorgusu, ilişki içindeki bir niteliğin aldığı değerler arasındaki bağımlılıkları, anahtarda yer almayan diğer niteliklerin gruplanmış olan verilerini kullanarak keşfeder.

Birliktelik kuralları, büyük veri yığınları arasındaki ilginç birliktelikleri veya da birliktelik ilişkilerini keşfeden veri madenciliği tekniğinidir (Han and Kamber 2000).

Birliktelik kuralı minimum güvenirlik ve destek ölçütlerini gözeterek veri analizi problemini ikiye bölmüştür (Zaki and Ogihara 1998, Srikant and Agrawal 1995).

Hong et al (1999), birliktelik kuralları ve Apriori algoritmasının nicel veriler üzerinde uygulandığı bir algoritma geliştirmişler ve bunun performansını I-Shou üniversitesi öğrencilerinin notlarının testinde kullanmışlardır.

Zaki and Ogihara (1998), Apriori algoritmasından farklı olarak sık geçen aday kümelerini DHP algoritmasını kullanarak azaltmışlardır. Kullanılan bu algoritma da veri tabanının bir çok kez taranması problemi sorunu ile karşımıza çıkmaktadır.

Savasere et al (1995), bölümleme (partitioning)algoritması kullanılarak daha önceden bahsetmiş olduğumuz veri tabanının bir çok kez taranması sorununu ortadan kaldırır ve veritabanının sadece iki kez okunması ile bellekte kullanılabilecek küçük parçalar oluşturulur. Birinci seferde, her kısımdaki yerel olarak kullanılan sık geçen öğe kümeleri okunur. İkinci seferde ise bu öğe kümelerinin tüm veri tabanındaki destek değerleri hesaplanarak en azından tek bir kısımda olması gereken potansiyel sık kullanılan öğe kümeleri taranır.

Brin et al (1997), DIC algoritması ile dinamik olarak küme setlerinin veri tabanı içinde birden çok bloklar halinde kısımlara ayrılmasını gerçekleştirmişlerdir. Apriori algoritmasından farklı olarak bu algoritmada, veritabanının en baştan yeniden taranması yerine başlangıç noktaları saptanıp bu noktalardan itibaren taranma işlemi gerçekleştirilir.

Toivonen (1996), Apriori algoritmasının sık kullanılan öğe kümelerinin daha kısa bir sürede elde edilmesi amacı ile en sık kullanılan aday öğe kümeleri yeni bir algoritma olan örnekleme (sampling) algoritması kullanarak belirlenmiştir. Geliştirilmiş olan bu algoritma ile veritabanı iki kez taranmaktadır. İlk seferinde veri tabanından örnek bir aday küme seti oluşturulmaktadır. Veri tabanının ikinci kez taranmasında ise bu

sık kullanılan öğelerin destek ve negatif destek sınırları göz önüne alınarak sonuçlar elde edilmektedir.

Liu (1968), Kombinasyon tekniğini kullanılarak varsayılan bir eşik değeri olmaksızın büyük miktarlardaki öğelerin esnek eşik değeri ile daha kısa bir zamanda elde edildiği görülmüştür.

Jea et al (2004), Internet üzerindeki büyük miktarlardaki öğe kümelerinin etkili ve esnek bir şekilde keşfedilmesi için yukarıda bahsettiğimiz kombinasyon tekniği esnek eşik değeri ile beraber Internet ortamında kullanılarak OCA algoritması geliştirilmiştir.

Park et al (1995), Apriori uygulamasının her bir tekrarı ile oluşan aday öğe kümesi sayılarının azaltılması için olasısal hesaplama kullanmışlardır. Bu işlem budama (Hashing) olarak da adlandırılır.

Ramaswamy et al (1998), Kütüksel (Calendric) Sepet analizi tekniğini geliştirerek kullanıcı tanımlı bir zaman aralığındaki tüm sık kullanılan küme setlerini bulmuşlardır.

Seno and Karypis (2001), Uzunluk budama madenciliği (LP Miner) algoritmasını geliştirerek, destek sınırının uzunluğunu kısaltmışlar ve böylece küme setlerinin keşfedilmesi için gereken ortalama işlem süresini azaltmışlardır.

Tsay and Chiang (2004), Birliktelik kurallarında rastlanan zorlukları aşabilmek için kümeleme temelli birliktelik kuralını (CBAR) ortaya çıkarmışlardır. Bu algoritmada Microsoft SQL sunucusunda bulunmakta olan FOODMART veritabanını deneysel olarak kullanılmış ve Apriori algoritmasının geliştirildiği gözlenmiştir. Burada keşfedilen örüntülerin sayı ve boyutu arttıkça performans aralığının daha aşikar olduğu görülmüştür.

Chen et al (2005), sepet analizi yöntemini çoklu biriktirme ortamlarında sorunsuz bir şekilde kullanılmasını sağlamak için SC (Stok zinciri- Stock Chain) algoritmasını

geliştirmişlerdir. SC algoritmasının geleneksel kurallardan farkı, kural uygulanırken stok yeri ve zamanı ile ilgili bilgilerinde kullanılmasıdır.

Yapılan bu çalışmada, birliktelik kurallarından olan Apriori algoritması kullanılarak öğrencilerin başarısız olduğu dersler analiz edilerek bu derslerin birbirleri ile ilişkili olan birliktelik kuralları ortaya çıkarılmıştır.

Bu tez beş bölümden oluşmaktadır. Birinci bölümde veri madenciliği ve birliktelik kuralları hakkında genel bilgi verilmiş, konu ile ilgili bilimsel dergiler, konferans bildirileri, sempozyum bildirileri ve ilgili kitaplardan literatür taraması yapılmış ve tez çalışmasının amacı hakkında genel bir tanımlama yapılmıştır.

İkinci bölümde, veri madenciliği ile ilgili temel kavramlar , veri madenciliği bilgi keşfi, ve veri madenciliği teknikleri hakkında bilgiler verilmiştir.

Üçüncü bölümde ise birliktelik kuralları analizi detaylıca anlatılmıştır. Ayrıca birliktelik kurallarında kullanılan algoritmalar incelenmiş ve bunların birbirlerine göre avantaj ve dezavantajları açıklanmıştır.

Dördüncü bölümde birliktelik kurallarından Apriori algoritması kullanılarak derslerin birbirleriyle olan ilişkileri irdelenmiştir.

Son bölümde, yapılan bu çalışmadan elde edilen sonuçlar ve bu çalışmanın, sonraki çalışmalara nasıl yön verebileceği konusuna değinilmiştir.

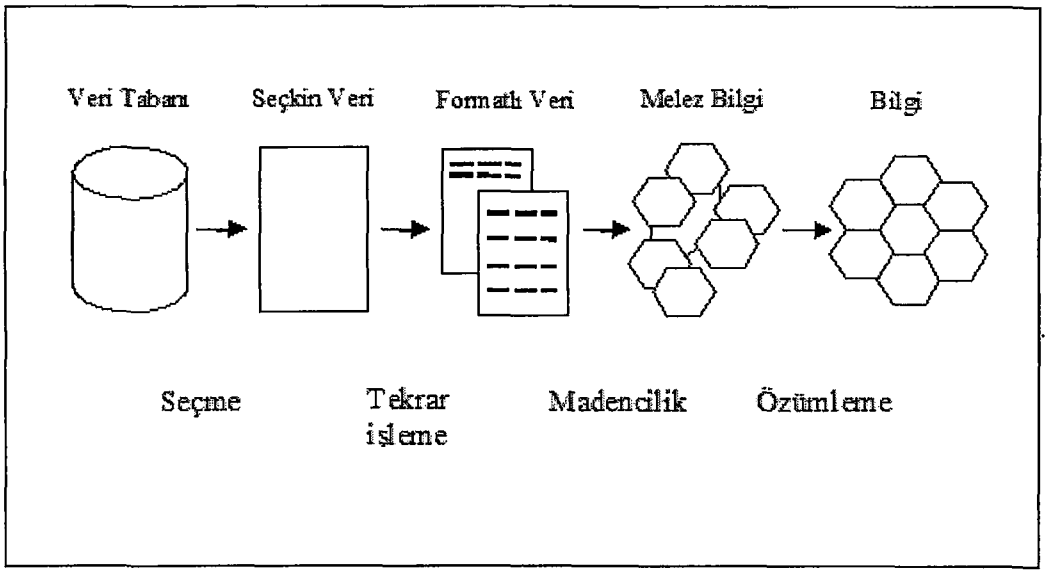
BÖLÜM 2. VERİ MADENCİLİĞİ ve TEMEL KAVRAMLAR

2.1. Veri Madenciliği Tanımı

Veri Madenciliği, büyük miktarlardaki veriler içerisinden gelecekle ilgili tahmin yapmamızı sağlayacak bağıntı ve kuralların bilgisayar programları kullanarak aranmasıdır (Alpaydın 2000).

Veri madenciliği, çeşitli veri analiz araçlarının kullanılmasıyla veriler üzerinde gelecek ile ilgili geçerli tahminler yapmamız için kullanılacak örüntü ve ilişkilerin keşfedilmesi süreci olarak da Two Crows Corparation (2006) tarafından tanımlanmıştır.

Veri madenciliği, yazılım mühendisliği kullanılarak elde edilen veri kümelerindeki önceden bilinmeyen kullanışlı verilerin çıkarılması sürecidir. Burada kullanılan yazılım mühendisliği veri kümesi, veri tabanlarında bulunan gereksiz bilgilerden bir sonuç çıkarmaktır (Mendonca and Sunderhaft 1999). Verinin veritabanından seçilip işlenerek bilgi haline getirilmesi Şekil 2.1.'de veri madenciliği süreci olarak gösterilmektedir.



Şekil 2.1. VM (Veri madenciliği) süreci.

2.2. Veri Madenciliği Uygulamaları

Günümüzde VM kavramına sıkça rastlamaktayız. Artık VM çeşitli alanlarda uygulanmaktadır. Bu uygulama alanları aşağıda özetlenmiştir (Akpınar 2004).

1. Pazarlama ve Perakendecilik : Müşterilerin satın alma örüntülerine göre bir müşteri portföyü belirlemek, mevcut müşterileri kaybetmeden yeni müşteriler kazanmak, pazar sepeti analizi (Market Basket Analysis) ile müşterilerin aldığı ürünler arasında bağıntı kurmak, müşteri ilişkilerini düzenlemek, satış tahmini ve satış noktaları belirleme gibi konularda kullanılır.

2. Bankacılık, Sigortacılık, Borsa : Kredi kartı dolandırıcılıklarının önlenmesi, kredi başvurularının puanlanması ve risk algılamasının yapılması, Sigorta poliçesi talep edebilecek müşterilerin belirlenmesinde, borsadaki tahvil ve hisse senetlerinde fiyat değerlendirmesi yapma gibi başlıca konularda kullanılmaktır.

3. Telekomünikasyon : Hatların yoğunluk durumuna göre düzenlemeler yapılarak kalite ve verimliliğin arttırılması gibi çalışmalarda kullanılmaktadır.

4. Sağlık ve İlaç : Hastalıkların teşhis edilmeleri sürecinde, tedavi yönteminin belirlenmesi, ilaç üretiminin ve geliştirilmesinin sağlanması amacı ile sağlık sektöründe kullanılmaktadır.

5. Biyoloji ve Genetik : Bitki türlerinin belirlenmesi ve ıslahında, genetik hastalıkların tespiti ve insan geni haritasının çıkarılmasında kullanılmaktadır.

6. Endüstri ve Mühendislik : Ürün üretimi iyileştirmesi ve bu ürünlerin kalite kontrol analizlerinin yapılması sürecinde, bilimsel ve teorik problemleri çözmek için kullanılmaktadır.

7. Sosyal bilimler ve Davranış bilimleri : Anketler düzenlenerek toplumun belirli bir konu hakkındaki eğilimleri belirlenerek anlamlı sonuçlar çıkarılmaktadır. Örnek olarak seçim anketlerini verebiliriz.

8. Metin Madenciliği (Text Mining) : Tek başına herhangi bir anlam içermeyen büyük miktarlardaki metin yığınları arasından anlamlı ilişkiler elde etmek için kullanılmaktadır.

9. Meteoroloji : Hava durumu tahminlerinin yapılması, ekolojik dengenin gözetilmesi amacı ile kullanılmaktadır.

10. Uzay Bilimleri : Gezegen ve galaksideki diğer tüm yıldızlar hakkında gerekli araştırmaların yapılmasında kullanılmaktadır.

11. Görüntü tanıma ve Robot görüş sistemleri : Algılayıcılar yardımı ile elde edilmiş çeşitli örüntülerin kullanılması ile engel ve yol tanıma işlemlerinde, adli alanda kullanılan yüz ve parmak izi tespiti gibi uygulamaları sayabiliriz.

12. Web Madenciliği (Web Mining) : Internet üzerindeki verilerin çeşit ve büyüklüğü her geçen gün artmaktadır. Internet üzerinde düz metin ve resim haricinde akan (streaming) ve sayısal verilerde bulunmaktadır. Bu verilerin

sınıflandırılarak en kısa sürede ulaşabilmemizi sağlamak için web madenciliği kullanılmaktadır.

2.3. Veri Madenciliğinde Karşılaşılan Problemler

Küçük veritabanlarında hızlı ve doğru olarak çalışan bir VM uygulaması, çok büyük bir veritabanına uygulandığında farklı sonuçlar üretebilir. Bu kısımda VM'de karşılaşılan problemler özet olarak açıklanmaktadır (Sever ve Oğuz 1999).

1. Veri Tabanı Boyutu : Veritabanlarının boyutu her geçen gün büyük bir hızla artmaktadır. VM uygulamalarının bir çoğu az sayıdaki veriyi değerlendirecek şekilde geliştirilmiştir. Örnekleme büyüdükçe aynı algoritmanın kullanılabilmesi için çok büyük bir dikkat göstermemiz gerekmektedir. Örneklemenin büyümesi ile birlikte kullanılan örüntü sayısı artmaktadır. Böyle bir durumda örnekleme yatay olarak indirgenmelidir. Yatay indirgeme çeşitli biçimlerde yapılabilmektedir. Bunlardan ilkinde belirli bir niteliğin alan değerleri önceden kategorize edilir. Daha sonra ilgili niteliğin değerleri aşağıdan yukarıya doğru değiştirilir ve tekrarlı çokluklar çıkartılır (Han et al 1992). Bir diğer yöntem ise, sürekli değerlerden oluşan bir alanı önceden belirlenmiş aralık değerlerine kesikleştirme tekniğinin uygulanmasıyla dönüştürülmesidir (Fayyad and Irani 1993).

2. *Gürültülü Veri : Büyük veritabanlarında çoğu niteliğin değeri yanlış olabilir.* Örnek olarak, kişinin doğum tarihinin 1975 yerine kullanıcı tarafından 1978 girilmesi veya girilecek değerin yanlış ölçüldüğü durumları sayabiliriz. Veri girişi yada veri toplanması sırasında oluşan sistem dışı hatalara gürültü adı verilmektedir (Sever ve Oğuz 1999). Gürültülü veri ile ilgili problemler tümevarımsal karar ağaçlarında kullanılan metotlarda araştırılmıştır (Quinlan, 1986b). Chan and Wong (1991) ise gürültü verisini veri tabanından çıkarılmasını sağlamak için istatistiksel yöntemler kullanmışlardır.

3. Boş (Null) Değerler : Veritabanında bulunmakta olan boş değerli nitelikler ise tamamen ihmal edilmeli yada bu niteliklere olası en uygun değerler atanmalıdır (Quinlan 1986a). Lee(1992), tarafından boş değerler bilinmeyen, uygulanamaz ve bilinmeyen ve uygulanamaz olmak üzere üçe ayırtmıştır.

4. Eksik Veri : Kullanmakta olduğumuz veriler bazı durumlarda yeterli olmayabilir.Verilerin kurumun ihtiyaçları gözetilerek yapılan değerlendirmede doğru analiz sonuçlarını yansıtmadığı görülebilir (Piatetsky - Shapiro 1991). Örneğin, sadece yaşlı hastalardan alınan veriler ışığında yapılan bir sorgulamaya göre konulan teşhis genç ve çocuk hastalar için geçerli olmayacaktır.

5. Dinamik Veri : Üniversitemizde kullanılan Öğrenci Bilgi Sistemindeki verilerin devamlı olarak değişmekte olmasını dinamik veriye örnek olarak gösterebiliriz. Eğer veri madenciliği programı sistem çalışırken kullanılırsa öncelikli olarak ÖBS' in yavaşlamasına neden olur.

2.4. Veri Tabanı ve Veri Ambarı Kavramları

Veritabanı, bir çeşit elektronik doldurma kabini olarak düşünülebilir. Başka bir deyişle bilgisayarla işlenmiş veri dosyalarının toplanmış olduğu bir ambardır (Date 1994).

Veritabanları hem okuma hem de yazma amaçlı olarak verilerin dinamik olarak değiştirildiği yapılardır. Veritabanı yönetim sistemleri çevrim içi hareket işlenmesi (OLTP) düşünülerek geliştirilmişlerdir.

Veri Ambarı (Data Warehouse), birden çok veri tabanı tarafından toplanan bilgilerin, gelecekte değerlendirilmek üzere arka plandaki başka bir yapıda birleştirilmesiyle oluşan veri deposudur. Veri Ambarları veri tabanlarının tersine sadece okuma amaçlı olarak kullanılırlar.

Veri ambarları sayesinde verinin analizi ve ileriye yönelik tahmin ve raporlamalar yapılır. Veri ambarlarında Codd et al (1993) tarafından geliştirilmiş olan verinin çevrim içi analitik işlenmesi yada kısaca OLAP, müşteri ilişkileri metodu (CRM) ve istatistiksel analiz ve raporlama işlemleri yapılır.

2.5. Veri Tabanlarında Bilgi Keşfi

Veri Madenciliği VTBK (Veri Tabanlarında Bilgi Keşfi) sürecinde yer alan bir adımdır (Fayyad et al 1996a). Fayyad et al (1996b)'a göre VTBK, veri içindeki geçerli, potansiyel olarak kullanılabilir ve en sonunda anlaşılabilen örüntülerin saptamaya yarayan önemli bir süreçtir ve VTBK sürecinde yer alan aşamalar aşağıdaki gibidir :

1. Veri Seçimi ve Örnekleme (Data Selection and Sampling): Veri ambarından seçilen birden çok veri kümesi birleştirilerek, sorgulamaya uygun örnekleme kümesi ortaya çıkarılır.

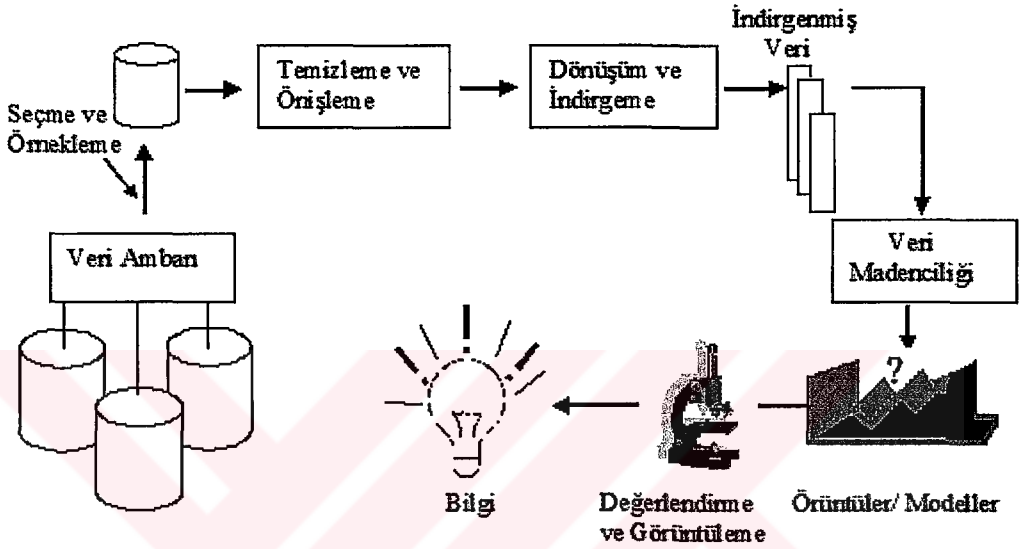
2. Veri Temizleme ve Önişleme (Data Cleaning &Preprocessing): Ortaya çıkarılan örnekleme kümesinde bulunan hatalı ve eksik nitelik değerlerinin ön işleme ile temizlenip değiştirildiği aşamadır.

3. Veri Dönüşümü ve İndirgeme (Data Transformation and Reduction): Örnekleme kümesinde bulunmakta olan ilgisiz ve tekrarlı nitelikler atılarak veri madenciliği sorgusunun daha hızlı çalışması sağlanır.

4. Veri Madenciliği (Data Mining): Belirlemiş olduğumuz bir VM tekniğinin kullanıldığı aşamadır.

5. Değerlendirme ve görüntüleme (Evaluation and Visualization): Keşfedilmiş olan bilginin ilginçlik, yararlılık, geçerli olma, basitlik ölçütlerine göre değerlendirilip rapor olarak görüntülenip kişilere bilgi halinde sunulmasıdır.

Yukarıdaki aşamaların gerçekleştirilmesi esnasında, Veri Madenciliği Sorgulama Dili kullanılır. Standart bir Veri madenciliği Sorgulama Dili (DMQL) olmamasına rağmen Han (1996) tarafından ilişkisel veri tabanları için bir dil oluşturulmuş ve kitabında bu dilin tüm yapısı açıklanmıştır.Şekil 2.2.' de VTBK sürecinde yer alan aşamalar gösterilmiştir.



Şekil 2.2. VTBK (Veri tabanlarında bilgi keşfi) süreci.

2.6. Veri Madenciliği Teknikleri

Veri madenciliği süreci sonunda elde ettiğimiz örüntüler kullanmış olduğumuz VM tekniğine göre farklılıklar gösterir. VM algoritmalarını iki grupta toplayabiliriz (Simoudis 1996). Bunlar, doğrulamaya dayalı VM ve keşfe dayalı tekniklerdir.

Doğrulamaya dayalı tekniklerde kullanıcı tarafından bir hipotez öne sürülür ve bu ispatlanmaya çalışılır. Bu tekniklerin genel olarak kullanıldığı alanlar ise istatistiksel ve çok boyutlu analizlerdir.

Keşfe dayalı tekniklerde otomatik olarak yeni bilgiler çıkarırlar. Bu teknikler sembolik ve sinir kümeleme, birliktelik keşfi, tümevarımsal denetim gibi araçlar kullanılarak çıkarılırlar.

VM teknikleri ve bunların gruplandırılması ile ilgili olarak yapılan çalışmalar bir çok kaynakta farklılıklar içermektedir. Hangi metodun hangi gruba ait olduğu kesinlik içermemektedir.

2.6.1. Birliktelik kuralları (association rules)

VM en çok kullanılan yöntemlerden birisi birliktelik kurallarıdır Birliktelik analizi kullanılarak veri tabanındaki veri kümelerinin hareketleri takip edilerek belirli sonuçlara varılır. Bankacılık ve Market sepeti işlemleri analizinde bu yöntemin kullanıldığını görmekteyiz. Market Sepeti verisi ile büyük süper marketlerdeki satış noktalarından elde edilen veriler kullanılarak müşterilerin alışveriş profili çıkarılabilmektedir. Örnek: Çocuk maması alanların %40'ı makarna da satın alır. Çocuk maması alanların çocuk bezi, makarna alanların ket çap alacağını tahmin etmek kolaydır. Fakat vermiş olduğumuz örnekte ise sonucu çıkarmak için bütün olasılıkları göz önüne alarak kolayca aklımıza gelmeyen ürün birliktelikleri ortaya çıkartılmaktadır.

Birliktelik analizi sadece mal satın alma işlemlerinde değil ayrıca günlük hayatımızın her alanında kullanılabilmektedir. Örneğin, bir ilköğretim okulundaki öğrencilerin beslenme durumları ile ilgili bir araştırma yaptığımızı varsayarsak. Karşımıza yaşları 10 ile 15 arasındaki öğrencilerden ailesinin geliri 800 YTL ile 1000 YTL arasında olanlardan balık satın alabildiğini görebiliriz. Bu durumu birliktelik kuralı ile şu şekilde ifade edebiliriz.

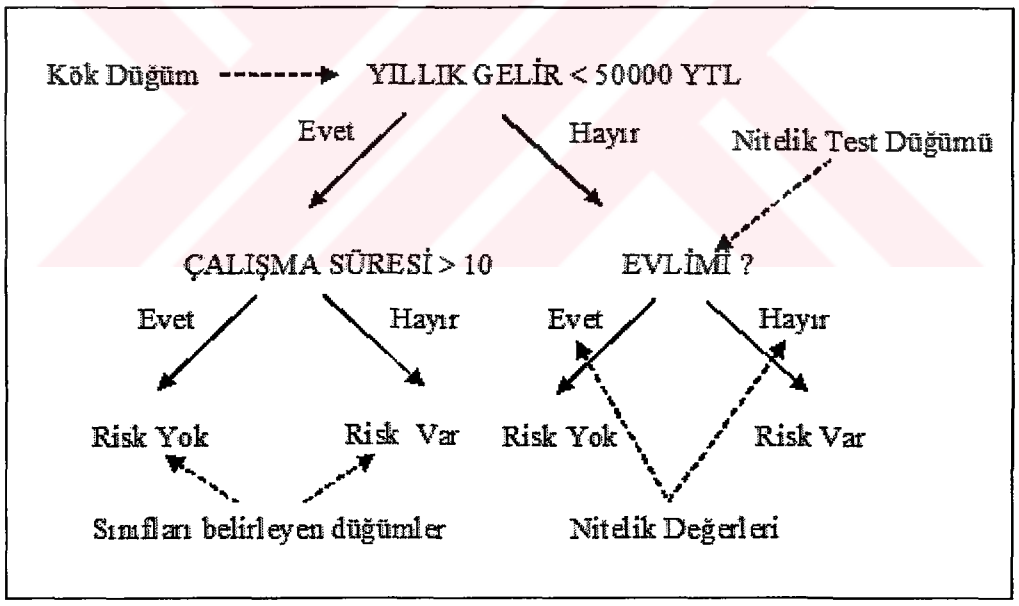
Yaş (X , "10..15") ^ Gelir (X , "800..1000") $\Rightarrow$ Alır (X , "Balık")

Bu Tezde VM tekniği olarak birliktelik kuralını kullandığımız için, birliktelik kuralları Bölüm 3'te detaylı olarak anlatılacaktır.

2.6.2. Sınıflandırma ve tahmin etme

Sınıflandırma, daha önceden belirlenmiş olan sınıflara herhangi bir sınıfa ait olmayan yeni verilerin atanması amaçlanır (Weiss et al 1991). Sınıflandırma metodu keşfe dayalı bir tekniktir.

Sınıflandırma Modeli değişik şekillerde olabilir. (IF-THEN) kuralları, karar ağaçları, matematiksel formüller veya yapay sinir ağları gibi. Karar ağacı, ağaç yapısında olan bir akış şeması şeklindedir. Düğümler üzerinde niteliklerin test işlemi yapılır.Test işleminin sonucuna göre ise dallanma meydana gelir. Sonuç olarak ağaç sınıflar ile son bulur. Karar ağaçları genellikle banka kredisinin onaylanması, kredi kartı ve sigorta işlemlerinde risk analizi yapmak gibi işlemler için kullanılır. Karar ağaçlarını kullanarak bir kişiye verilen kredinin risk değerlendirmesinin yapıldığı örnekle ilgili karar ağacı yapısı Şekil 2.3.'te verilmiştir.



Şekil 2.3. Karar ağacı örneği.

Karar ağaçları ile ilgili çeşitli algoritmalar geliştirilmiştir. Bery and Linoff (1997) tarafından yayınlan bu algoritmalar aşağıda kısaca açıklanmıştır:

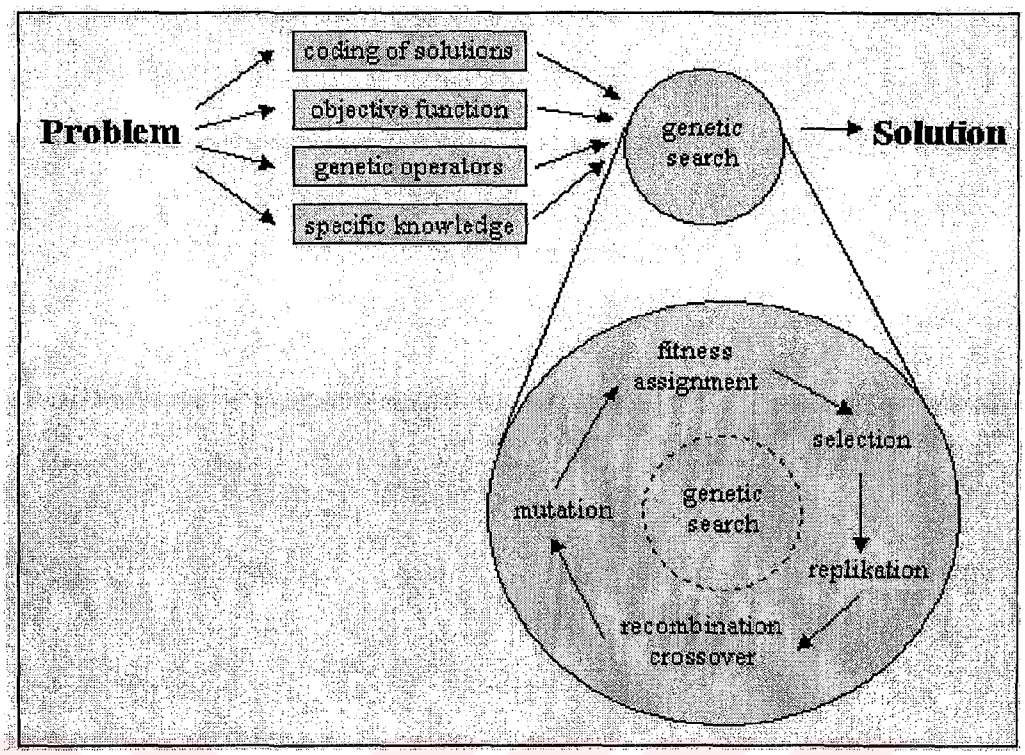
CART algoritması Briemen tarafından 1984 yılında , C4.5 algoritması Quinlan tarafından ve CHAID algoritması da Hartigan tarafından 1975 yılında bulunmuştur (Bery and Linoff 1997). CHAID algoritmasının diğer iki algoritmadan farklı olarak sadece sınıflandırılmış verilere uygulanmasından dolayı ağacın budanmasına gerek yoktur.

Yapay sinir ağlarında öğrenme (artificial neural networks) insanın deneyimleriyle öğrenme yolu gibidir (Bigus 1996). Diğer bir deyişle, bir bebeğin büyüme süreci içerisindeki adımlardan olan yürümesinden önce emeklemesi, sonra tutunarak ayakta durması ve en son aşamada yürümeye başlamasını gösterebiliriz. Bu süreç içinde olaylar gözlemlenmekte ve burada kazanılan deneyimler daha sonra kullanılmak üzere bir veritabanına kaydedilmektedir.

Yapay sinir ağlarında önceden sonuçları bilinen veri kümeleri eğitilerek yeni gelecek verilerin işlenmesi için gereken ağırlıklar hesaplanır. Veri kümelerinin eğitiminde kullanılan değerlerin sayısal değerler olması gerekmektedir.

Genetik Algoritmalar (Genetic Algorithms) sinir ağlarının kullanmış olduğu fikir yapısına dayanırlar. Genler içerisinden farklı kombinasyonlarda olanları bulunur ve bunlara çaprazlama, mutasyon ve eleme gibi genetik algoritma teknikleri uygulanarak gelecek nesil elde edilir.

Genetik algoritma doğal eleme ve genetik evrimin öncül olarak deneye dayalı uyarlanabilen bir şekilde araştırılmış olduğu bir algoritmadır (Pohlheim 1997). Genetik ile ilgili olarak, genetik algoritma sürecini gösteren tipik bir genetik yapısı Şekil 2.4.'te gösterilmiştir.



Şekil 2.4. Genetik algoritma süreci.

Bayesian kuramı, rastlantısal bir olay değerlendirilirken tüm olasılıkların göz önüne alınarak sonuçların güncellenmesidir. Örneğin, paranın havaya atıldığında yazı veya tura gelmesi olasılığını %50 olarak değerlendirmek yerine, paranın yazı veya tura geleceğine diğer faktörleri de göz önüne alınarak bir sonucun çıkarılmasıdır. Burada havanın o anki rüzgar, yağmur ve paranın durumu gibi durumları diğer faktörler olarak düşünebiliriz. Bu teorem Bayes öldükten iki yol sonra makale olarak yayınlanmıştır (Bayes 1764).

K-En Yakın Komşu (K-Nearest Neighbor) metin sınıflandırması için geliştirilmiş olan makine öğrenme algoritmasıdır (Yang 1999).

Katı Küme Yaklaşımı (Rough Set Approach) bir veri kümesinin başka bir veri kümesi ile birlikte sınıflandırıldığında,bu sınıflandırmalar çok büyük olabilir.Bu durumda azami sınıflandırma kuralı (maximal association) kullanılır (Feldman et al 1997, 1998). Katı küme yaklaşımı ile azami sınıflandırmalar bulunur.

Bulanık Küme Yaklaşımı (Fuzzy Set Approach) karasızlık durumlarında daha doğal bir durum olan insanın bakış açısı ile hüküm verme işlemidir (Zadeh 1965).

2.6.3. Kümeleme (Clustering)

Bu algoritmada veri tabanında bulunan aynı nitelikler gruplanarak alt kümelere ayrılır. Geleneksel olarak kümeleme algoritmaları iki ana kategoriye ayrılır. Bunlar, bölümlemeli (partitional) ve aşamalı (hierarchical) algoritmalardır (Jain and Dubes 1988).

Bölümlemeli Kümeleme algoritması, aynı zamanda K-Means metodu olarak bilinir (MacQueen 1967). Bu metotta izlenen aşamalar aşağıda sırası ile verilmiştir.

1. K sayısınca istekli kümeleme seçilir.
2. Bu seçilen kümelemelerin her biri için bir merkez kaydı alınır.
3. Veri setine gidilir ve en yakın kümelemeye atanması işlemi yapılır.
4. Yeni kümelemeler için tekrardan merkez hesaplanır.
5. Kümeleme ile kayıtlar arasındaki mesafe minimum oluncaya dek 4 ve 5. adımlar tekrarlanır.

Bölümlemeli küme metoduna dayanarak sırasıyla, Karınca sistemleri (Dorigo et al 1991), Birch (Zhang et al 1996) ve K Harmonic Means (Zhang et al 1999) gibi değişik algoritmalarda geliştirilmiştir.

Aşamalı Kümeleme yönteminde ise birleştirici (agglomerative) ve bölücü (divisive) kümeleme algoritmaları kullanılır (Milligan 1980).Bu algoritmada izlenen aşamalar aşağıda sırası ile verilmiştir.

1. Veri kümesi içindeki her bir kayıt için kümeleme oluşturulur.
2. En yakın kümeleme en büyük olana karıştırılır.
3. Tek bir kümeleme kalana dek işleme devam edilir.

BÖLÜM 3. BİRLİKTELİK KURALLARI

3.1. Birliktelik Kuralları Nelerdir ?

Birliktelik kuralları ile bir ilişkide yer alan niteliklerin değerleri arasındaki bağımlılıklar, anahtarda yer almayan diğer niteliklerin gruplandırılması ile bulunur. Bu kurallar ilk olarak Agrawal (1994) tarafından geliştirilmiştir.

Birliktelik kurallarının analizi süreci market sepeti analizi olarak da adlandırılır (Agrawal et al 1993). Market sepeti analizinde müşteri ile ilgili veri hareketlerinden gelecekte müşterinin nasıl bir tercih yapacağına dair sonuçlar tahmin edilir. Birliktelik kuralı ilgili basit bir örnek vermek istersek, eğer kişi restoranda biftek sipariş etmişse aynı zamanda içecek olarak kola istemiştir.

Birliktelik kuralları destek ve güven parametrelerine dayanır. İki küme seti arasındaki destek düşükse bir sonuca ulaşma olasılığımız oldukça zayıflar. A veya B öğe kümesi nitelikleri içindeki işlem oranı bize destek miktarını verir. Güven ise sepet içindeki B ve A niteliklerinin birlikte bulunma olasılığına dayanır.

Örnek olarak, A $\Rightarrow$ B (destek= %10, güven=80), burada belirtilen %10 destek değeri ile tüm alışverişlerde A ve B ürünlerinin %10 oranında birlikte satıldığını gösterir. %80 güven oranı ise A ürünü satın alanların aynı alışverişte B ürününü de aldığını gösterir.

Veri madenciliğinde oluşturulan tüm birliktelik kurallarının veri tabanı içinde oluşturulmuş olan minimum destek (sık kullanılma kuralı) ve minimum güven eşik değerlerinden (güven kuralı) büyük olması gerekir. Bu işlem iki adımda gerçekleşir (Zaki 1999). Bunlar;

1. Minimum desteğe sahip olan sık kullanılmış küme setleri bulunur. Sık tekrarlanan öğelerin arama uzayındaki sayısı 2^m'dir.

2. Sık tekrarlanan öğelerden minimum güven değerine sahip güçlü birliktelikler oluşturulur.

3.2. Birliktelik Kurallarının Zafiyet ve Üstünlükleri

Birliktelik kurallarının sağlamış olduğu en büyük avantajlardan birisi, veri madenciliği sonucunda ortaya çıkan sonuçların kolayca anlaşılabilir olmasıdır (Mena 1999). Bu duruma uygun bir çok örnek verebiliriz. Örneğin, Cips alan müşterilerin kola alması gibi. Bunu baz alarak işletmeler gelecekle ilgili nasıl bir strateji uygulayacaklarına dair kararlar verirler.

Veri Madenciliği aşamasında yapılması gereken hesaplamaların karmaşık olmaması bu metodun diğer bir üstünlüğüdür (Berry et al 1997). Klasik bir birliktelik kuralında destek ve güven kuralları kolayca tanımlanır.

Diğer taraftan birliktelik kurallarının kullanılması ile ilgili engel ve zorluklarda vardır. Hesaplamaların basit olmasına rağmen arama uzayı az miktardaki küme sayısı için bile oldukça büyüktür. Kümeler ve arama uzayı arasındaki ilişki üssel bir büyüklüğe dayanır (Cabena et al 1997).

Birliktelik kurallarının bir diğer zafiyeti ise, az rastlanan kümelerin ihmal edilmesidir. Az rastlanan kümelerin ihmal edilmesi işlemi sonucunda sık kullanılan küme setlerinden bu kümeler çıkartılır. Böylece burada kullanmış olduğumuz destek kuralı yavaşlar.

Sonuç olarak, algoritmaların uyarlanması çok büyük sayıdaki birliktelik kuralına yol gösterme eğilimidir. Buradaki eğilimin dikkatlice sorgulanması ve karışıklıkların giderilmesi sıklıkla rastlanan bir durumdur.

3.3. Birliktelik Kurallarında Problem Tanımı

Birliktelik kurallarının matematiksel olarak ifade edilmesi Agrawal et al (1993) tarafından yapılmıştır.

Bu ifade şeklinde I= $\{I_1,I_2,\ldots,I_n\}$ verilen problemde kullanılan öğeler, D ise veritabanı hareketlerini ,T ise veritabanındaki her bir hareketi, *tid* her harekete ait biricik numarayı, k-öğe kümesi (k-item set) k adet ürün içeren öğe kümesini temsil etmektedir. A öğe kümesi için destek $\sigma(A)$ şeklinde gösterilir. T hareketi A ürün kümesini $A \subseteq T$ şartını sağlıyorsa içerir. $A \Rightarrow B$ biçiminde bir birliktelik kuralı tanımlamamız için A ve B öğe setleri I ürün kümesinin bir alt kümesi ($A \subset I$ ve $B \subset I$) ve $A \cap B = \emptyset$ olması gerekir. Bu ifade ile A B' yi kapsar veya B kümesinin olması A kümesinin varlığı ile ilişkilidir diyebiliriz.Buradan elde edilen bağımlılık ilişkisinin yüzde yüz doğru olması söz konusu değildir.

Oluşturulmuş olan kuralın kullanmış olduğumuz hareketler tarafından desteklenmesi gerekir. Bu sebepten, $A \Rightarrow B$ birliktelik kuralı minimum değeri belirlenmiş bir güven ve destek kuralları sağlanacak şekilde oluşturulur. Buradaki destek kuralı $\sigma(A \cup B)$ ve güven kuralı ise $\sigma(A \cup B)/\ \sigma(A)$ şeklinde ifade edilir (Zaki 1999).

3.4. Birliktelik Kuralı Algoritmaları

Birliktelik kuralı algoritmaları sıralı ve paralel algoritmalar olmak üzere ikiye ayrılır. Sıralı ve paralel algoritmaların neler olduğu ve nasıl çalıştıkları bu kısımda özetle anlatılmıştır.

3.4.1. Sıralı algoritmalar

Sıralı algoritmaların başlıca kullanılanları, AIS, Apriori, SETM, DHP, Bölümleme, DIC, SEAR, ECLAT, PEAR algoritmalarıdır. Bu tezde birliktelik kurallarından Apriori algoritması kullanıldığından dolayı Apriori algoritmasına detaylıca değinilmiştir.

3.4.1.1. AIS algoritması

Sıralı algoritmalar içerisinde en ilkel olanı AIS' tir (Agrawal et al 1993). Kullanılan bu algoritmanın en büyük özelliği birden fazla elemana sahip kuralların oluşturulamamasıdır. Burada birliktelik kuralının ifade şekli $X \Rightarrow I_k$ biçimindedir.

AIS algoritmasının en büyük dezavantajı, daha sonradan küçük öğe kümeleri olacak çok sayıda aday öğe küme üretmesi ve kuralın sağ tarafında sadece bir tek öğe ile sınırlı olunmasıdır.

3.4.1.2. Apriori algoritması

Veri Madenciliğinde kullanılmakta olan en yaygın tekniklerden biridir. Bu algoritmada veritabanı birden fazla taranarak sık kullanılan öğe kümeleri bulunur. Tarama iki aşamadan oluşur, bunlar aday üretme ve aday hesaplamadır (Ganti et al 1999).

İlk taramada üretilmiş olan aday öğe kümelerinden minimum destek ölçütünü sağlayan 1 elemanlı öğe kümeleri bulunur. Daha sonraki taramalarda ise bir önceki taramada bulunan sık geçen aday kümeleri budanarak yeni sık geçen öğe kümeleri üretilir.

Aday kümelerin destek değerleri tarama sırasında hesaplandıktan sonra minimum destek ölçütünü sağlayan öğe kümeleri, o tarama için üretilmiş olan sık geçen öğe kümeleri olurlar. Bir sonraki taramada bu işlem sonucunda elde edilen öğe kümeleri aday kümeler olur ve bu işlem yeni bir sık geçen öğe kümesi bulunmayana kadar devam eder.

Apriori algoritmasında kullanmış olduğumuz değişkenler Tablo 3.1.'de özet olarak verilmiştir.

Tablo 3.1. Apriori algoritması değişkenleri.

k-öğe küme	K adet öğe içeren küme
L_k	Sık geçen K öğe kümesi
C_k	Aday K öğe kümesi

Apriori algoritmasının çalışmasını Tablo 3.2'de verilen veritabanına göre bir örnekle anlatılacaktır (Han 2000).

Tablo 3.2. Elektronik işlem bilgileri.

İNO	Öğe Listesi NO
İ100	I_1, I_2, I_5
İ200	I_2, I_4
İ300	I_2, I_3
İ400	I_1, I_2, I_4
İ500	I_1, I_3
İ600	I_2, I_3
İ700	I_1, I_3
İ800	I_1, I_2, I_3, I_5
İ900	I_1, I_2, I_3

Tablo 3.2.'de elektronik işlem bilgilerini içeren D veritabanı görülmektedir. Veritabanında yapılan işlem numaraları İNO sütununda görülmektedir. Her bir işlemde kullanılan öğeler ise Öğe Listesi No sütununda görülmektedir. Tablo 3.2. kullanılarak oluşturulan apriori algoritmasında izlenen adımlar aşağıda sırası ile gösterilmektedir (Han 2000).

1. Algoritmanın ilk adımında, her öğe C_1 aday kümesinin elemanıdır. Her öğeden kaç tane olduğunu bulmak için algoritma tarafından tüm veritabanı taranır. Şekil 3.1.'de görüldüğü gibi D veritabanında 6 adet I_1, 7 adet I_2, 6 adet I_3, 2 adet I_4 ve 2 adet I_5 öğesi bulunmaktadır.

2. Minimum işlem destek sayısının 2 olduğunu kabul edelim (min_destek =2/9=22). En sık tekrar edilen 1 elemanlı öğe kümeleri L_1 kümesi olarak Şekil 3.1.'de gösterilmektedir. Burada minimum desteği sağlayan bir elemanlı aday öğe kümeleri bulunur. Verilen örnekte tüm öğeler minimum destek değeri olan 2'den büyük olduğu için herhangi bir budanma işlemi olmamıştır.

3. En sık tekrarlanan 2 öğeli kümeler olan L_2'yi bulmak için, L_1 kümesindeki öğelerin ikili kombinasyonları bulunarak C_2 kümesi oluşturulur ($C_2=L_1 \bowtie L_1$).

4. Daha sonra, C_2 kümesindeki her bir aday öğe kümesinin destek değerlerini bulmak için D veritabanı taranır. Burada bulunan değerler şekil 3.2.'de gösterildiği gibi C_2 kümesine Destek sayısı sütunu eklenerek oluşturulur.

5. C_2 kümesindeki minimum destek şartını sağlayan iki öğeli kümeler L_2 kümesine aktarılır.

6. 3 elemanlı sık tekrarlanmış olan öğe kümelerini C_3 oluşturmak için önce L_2 kümesindeki öğelerin 3'lü kombinasyonları bulunur. ($C_3=L_2 \bowtie L_2$)={{ I_1, I_2, I_3},{ I_1, I_2, I_5},{ I_1, I_3, I_5},{ I_2, I_3, I_4},{ I_2, I_3, I_5},{ I_2, I_4, I_5}}. Fakat Apriori algoritmasına göre, sık tekrarlanmış olan öğelerin alt kümeleri de sık tekrarlanmış olması gerektiğinden dolayı C_3 kümesi, C_3={{ I_1, I_2, I_3},{ I_1, I_2, I_5}} olur. Bu işlemler yapılırken 2 aşamada yapılır.

- Kombinasyon (Join): Apriori özelliği kullanılarak iki kümenin kartezyen işlemi yapılır. Aşağıda bu kartezyen işleminin nasıl yapıldığı gösterilmiştir.
  $C_3=L_2 \bowtie L_2$={{ I_1, I_2},{ I_1, I_3},{ I_1, I_5},{ I_2, I_3},{ I_2, I_4}{ I_2, I_5}}$\bowtie${{ I_1, I_2},{ I_1, I_3},{ I_1, I_5},{ I_2, I_3},{ I_2, I_4},{ I_2, I_5}}={{ I_1, I_2, I_3},{ I_1, I_2, I_5},{ I_1, I_3, I_5},{ I_2, I_3, I_4},{ I_2, I_3, I_5},{ I_2, I_4, I_5}}

- Budama (Prune): Apriori özelliği kullanılarak budama işlemi yapılır. { I_1, I_2, I_3}'ün 2 öğeli alt kümeleri olan { I_1, I_2},{ I_1, I_3} ve { I_2, I_3}'ün hepside 2 öğeli kümeleri içeren L_2 üyesi oldukları için { I_1, I_2, I_3} C_3'te bulunur.

- { I_1, I_2, I_5}'ün 2 öğeli alt kümeleri olan { I_1, I_2},{ I_1, I_5} ve { I_2, I_5}'ün hepside 2 öğeli kümeleri içeren L_2 üyesi oldukları için { I_1, I_2, I_5} C_3'te bulunur.

- { I_1, I_3, I_5}'ün 2 öğeli alt kümeleri olan { I_1, I_3},{ I_1, I_5} ve { I_3, I_5}'ten { I_3, I_5} L_2 üyesi olmadığı için { I_1, I_3, I_5} C_3'ten çıkarılır.

- { I_2, I_3, I_4}'ün 2 öğeli alt kümeleri olan { I_2, I_3},{ I_2, I_4} ve { I_3, I_4}'ten { I_3, I_4} L_2 üyesi olmadığı için { I_2, I_3, I_4} C_3'ten çıkarılır.

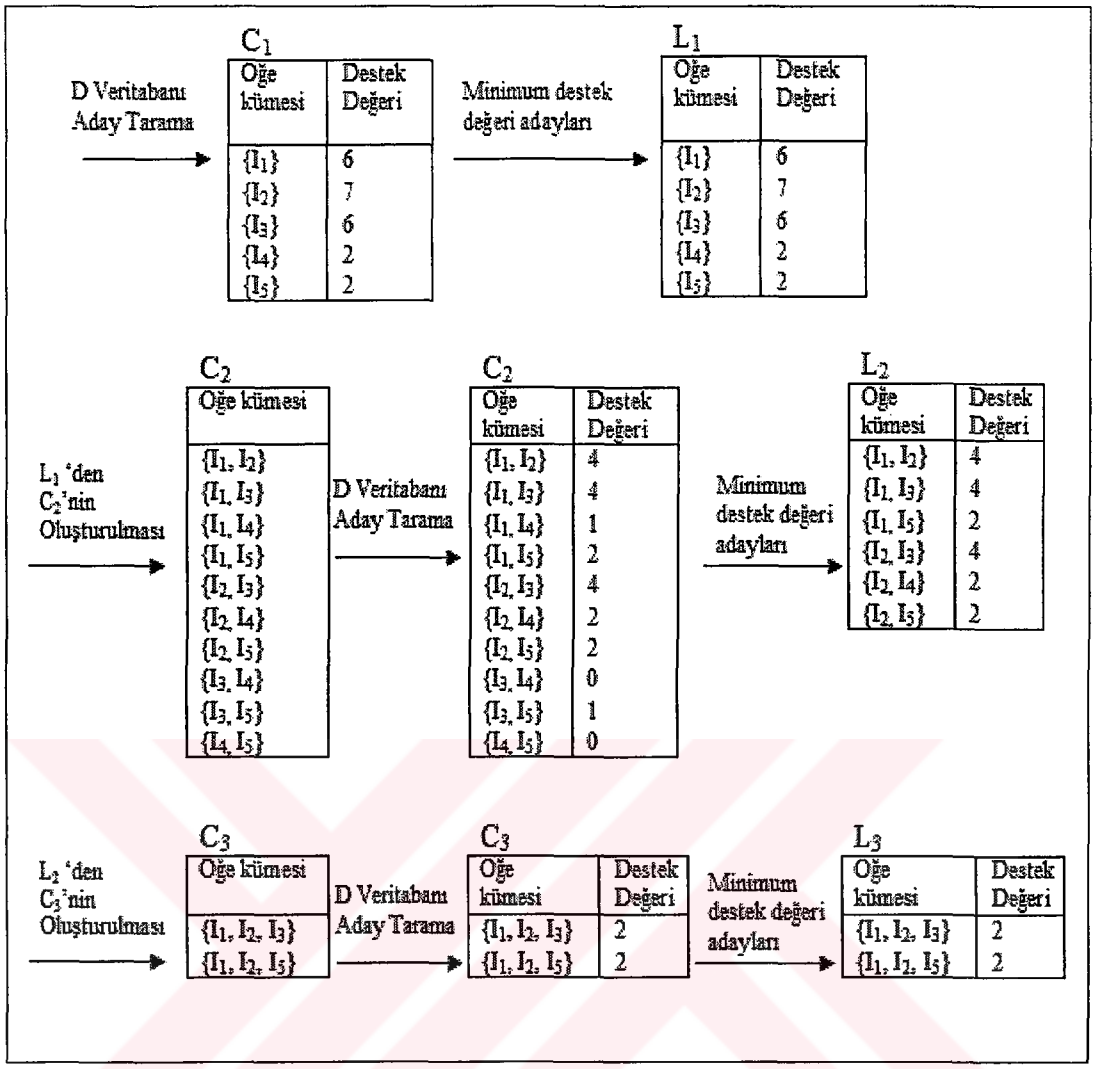
- { I_2, I_3, I_5}'in 2 öğeli alt kümeleri olan { I_2, I_3},{ I_2, I_5} ve { I_3, I_5}'ten { I_3, I_5} L_2 üyesi olmadığı için { I_2, I_3, I_5} C_3'ten çıkarılır.

- { I_2, I_4, I_5}'in 2 öğeli alt kümeleri olan { I_2, I_4},{ I_2, I_5} ve { I_4, I_5}'ten { I_4, I_5} L_2 üyesi olmadığı için { I_1, I_3, I_5} C_3'ten çıkarılır.

7. C_3 kümesindeki öğelerin destek değerlerini bulmak için D veritabanı taranır ve L_3 oluşturacak olan 3 elemanlı minimum destek koşulu sağlayan aday kümesi destek sayısı sütunu eklenerek oluşturulur.

8. 4 elemanlı öğe kümelerini oluşturmak için algoritma kullanılarak $L_3 \bowtie L_3$ işlemi yapılır. Bu yapılan dörtlü tek kombinasyondan sonra $C_4=\{I_1, I_2, I_3, I_5\}$ olur. Fakat burada bulunan öğe kümesinin alt kümesi ($\{I_2, I_3, I_5\}$) sık tekrarlanan olmadığı için $C_4=\phi$ olur ve böylece sık kullanılan tüm öğeler Apriori tarafından bulunduğu için tarama işlemi sonlanır.

Minimum destek değeri 2'ye göre aday ve sık tekrarlanan öğe küme kümelerinin oluşturulmasında izlenen adımlar ile ilgili yapı Şekil 3.1.'de gösterilmiştir.



Şekil 3.1. Aday ve sık tekrarlanan öğe küme kümelerinin oluşturulması.

Şekil 3.2.'de Han (2000) tarafından geliştirilmiş olan algoritma kesiti, Şekil 3.3.'te ise burada kullanılan apriori-gen işlevinin YSD (Yapısal Sorgu Dili)'den faydalanılarak oluşturulmuş olan algoritma kesiti verilmiştir.

Burada ilk adımda sık tekrarlanan bir öğeli kümeler bulunur. Adım 2-10 arasında, k-1 adet öğeye sahip L_{k-1} öğe kümesi kullanılarak k adet öğeye sahip olan L_k öğe kümesi bulunur. Adım 3'te apriori-gen işlevi kullanılarak sık tekrarlanmayan alt öğe kümeleri budanır. Adım 4'te tüm adaylar için veritabanı taranır. Adım 5'te adayların alt kümeleri elde edilir. Adım 6 ve 7'de her aday kümesi için hesaplama yapılır. En son olarak ise tüm elemanlar için minimum desteği sağlayanlar elde edilerek C_k bulunmuş olur.

```
L1={sık geçen 1-öğe kümelerinin D veritabanında bulunması}
FOR (k=2;Lk-1 ≠ φ;k++){
      Ck=apriori_gen(Lk-1,min_destek);
            FOR (Tüm t ∈ D hareketleri için) //D taranır{
               Ct=subset(Ck,t);
               FOR (Tüm c ∈ Ct hareketleri için) //D taranır{
               c.değer=c.değer+1; //aday kümeler elde edilir.}
      Lk={c ∈ Ck | c.değer≥min_destek
                        }
return Ck
}
```

Şekil 3.2. Apriori algoritma kesiti.

```
INSERT INTO Ck
      SELECT l1[1], l1[2],... l1[k-1], l2[1], l2[2],... l2[k-1]
      FROM L1 ⋈ L2
      WHERE l1[1]∧l2[1]= l1[2]∧l2[2] ∧... ∧ l1[k-1]∧l2[k-1]
Tüm c∈ Ck aday kümeleri için
      Tüm c kümesinin (k-1) öğeli alt kümeleri olan s için
            Eğer (s∉ Lk-1) ise
                  DELETE c FROM Ck //Budama adımı
```

Şekil 3.3. Apriori-gen algoritma kesiti

D veritabanındaki sık tekrarlanan öğeler bulunduktan sonra sık kullanılan öğeler için birliktelik kuralları oluşturulur. Bu kuralların oluşturulması için elde edilen destek değerleri kullanılarak güven oranı bulunur. Aşağıdaki formülde bu oranın nasıl hesaplandığı gösterilmiştir (Han 2000).

$$\text{Güven } (A \Rightarrow B) = P(B \backslash A) = \frac{\text{destekdeğeri } (A \cup B)}{\text{destekdeğeri } (A)} \qquad (3.1)$$

Formül 3.1'de kullanmış olduğumuz destekdeğeri $(A \cup B)$, $A \cup B$ öğe kümelerinin işlem sayısıdır. Destekdeğeri (A) ise A öğe kümesinin içermiş olduğu işlem sayısıdır. Bu eşitlik kullanılarak birliktelik kuralı şöyle oluşturulur:

- Tüm sık kullanılan l öğe kümeleri için, l'nin boş olmayan tüm alt kümeleri oluşturulur.

- Tüm boş olmayan l'nin alt kümesi s için, $\frac{\text{destekdeğeri(l)}}{\text{destekdeğeri(s)}} \geq$ minimum güven ise sonuç kuralı $s \Rightarrow$ (l-s)' dir.

Tablo 3.2.'deki veritabanı kullanılarak elde edilmiş olan $l=\{I_1, I_2, I_5\}$ bağıntısı için yukarıdaki formül kullanılarak Tablo 3.3.'teki kuralları çıkarabiliriz. Burada boş olmayan alt kümeler $\{I_1, I_2\},\{I_1, I_5\},\{I_2, I_5\},\{I_1\},\{I_2\},\{I_5\}$'tir.

Tablo 3.3. Birliktelik kuralları.

Birliktelik	Güven
$I_1 \wedge I_2 \Rightarrow I_5$	güven=2/4=50%
$I_1 \wedge I_5 \Rightarrow I_2$	güven=2/2=100%
$I_2 \wedge I_5 \Rightarrow I_1$	güven=2/2=100%
$I_1 \Rightarrow I_2 \wedge I_5$	güven=2/6=33%
$I_2 \Rightarrow I_1 \wedge I_5$	güven=2/7=29%
$I_5 \Rightarrow I_1 \wedge I_2$	güven=2/2=100%

Eğer minimum güven eşik değeri 70% olarak varsayılırsa, Tablo 3.3.'teki birliktelik kurallarından ikinci, üçüncü ve altıncı kurallar güçlü birliktelikler olarak dikkate alınır (Han, 2000).

Apriori algoritmasını daha etkili olarak kullanabilmemiz için bu algoritmanın çeşitli varyasyonları geliştirilmiştir. Bunlar bu bölümde bahsetmiş olduğumuz bölümleme (partitioning), budama (hashing), örnekleme (sampling), dinamik öğe kümesi sayma (DIC - dynamic item set counting) yöntemleridir.

3.4.1.3. SETM algoritması

SETM Swami and Houtsma (1993) tarafından bulunmuştur. Bu algoritma YSD kullanılarak basitçe çalıştırılabilir. Bu algoritmada sık kullanılan öğe kümelerinin her öğesi TID öğe kümesi şeklindedir.

SETM algoritmasının en büyük dezavantajı her bir aday kümesine biricik anahtar olan TID verme zorunluluğudur. Bu zorunluluk aday öğe kümelerini saklamak için daha büyük bir depolama alanı ihtiyacı ortaya çıkarmaktadır.

3.4.1.4. DHP algoritması

DHP Park et al (1995) tarafından bulunmuştur.Bu algoritma hash tablolarını kullanarak arama uzayını azaltır. Apriori algoritmasındaki k öğeli sık geçen küme adayları hash tablolarının kullanılması ile ilk taramada yeniden hesaplanır. İkinci taramada sadece minimum desteğe sahip olan bu hash hücrelerindeki küme adayları kullanılarak veritabanı taranır.

Bu algoritmanın dezavantajı ise veri tabanının bir çok kez taranması problemini ortadan kaldıramamasıdır.

3.4.1.5. Bölümleme (Partition) algoritması

Bölümleme algoritması Savasere et al (1995) tarafından bulunmuştur.Bu algoritma kullanılarak veritabanının mantıksal olarak iç içe gelmemiş olarak yatayda bölünmesi işlemi yapılır.

Bu algoritmanın avantajları, veritabanını sadece iki kez okuyarak giriş ve çıkış işlemlerini azaltması ve veritabanının bellekte yer alabilecek en küçük parçalara bölünmesidir.

3.4.1.6. DIC algoritması

DIC algoritması Apriori algoritmasının Brin et al (1997) tarafından geliştirilmiş diğer bir genellemesidir. Bu algoritma, öğe kümelerinin veri tabanındaki tarama sayısını azaltmak için kullanılmakta olan etkili bir yöntemdir. Yalnız burada kullanılan verilerin homojen bir yapıda olması gerekmektedir. Aksi takdirde, yerelde sık fakat genelde sık olmayan yanlış öğe kümeleri üretilir ve bu veritabanın Apriori algoritmasından bile daha yavaş taranmasına neden olur.

3.4.1.7. Diğer sıralı algoritmalar

Örnekleme (Sampling) algoritması Toivonen (1996) tarafından bulunmuştur. Bu algoritmada Apriori algoritmasının sık kullanılan öğe kümelerinin bulunması amacı ile yeni bir yöntem belirlenmiştir Bu yöntemde sık kullanılan öğe kümelerini bulmak için destek ve negatif destek değerleri dikkate alınarak oluşturulan örnek aday kümesi kullanılmaktadır.

SEAR (Sırasal Etkili - Sequential Efficient Association Rule) ve Spear algoritmaları Muller (1995) tarafından bulunmuştur.

ECLAT (Sınıf Temelli Denk - Equivalence Class-Based), MaxEclat ve MaxClique algoritmaları Zaki (1997) tarafından bulunmuştur.

3.4.1.8. Sıralı algoritmaların karşılaştırılması

Sıralı algoritmaların Tablo 3.4.'te birbirleri ile karşılaştırılmaları özet olarak gösterilmiştir (Zaki 1999).

Tablo 3.4.'te veritabanı tarama sayısı 2^x tane sık geçen öğe kümesi düşünülerek hesaplanmıştır.

Tablo 3.4. Sıralı algoritmaların karşılaştırılması.

Algoritma	Veri Yapısı	Arama Şekli	Sayma	Veri Tabanı Tarama Sayısı
AIS	Hash Tree	Aşağıdan - Yukarıya	Hepsi	X+1
Apriori	Hash Tree	Aşağıdan - Yukarıya	Hepsi	X+1
SETM	Hash Tree	Aşağıdan - Yukarıya	Hepsi	X+1
DHP	Hash Tree	Aşağıdan - Yukarıya	Hepsi	X+1
Partition	Yok	Aşağıdan - Yukarıya	Hepsi	2
DIC	Prefix Tree	Aşağıdan - Yukarıya	Hepsi	<= X+1
SEAR	Prefix Tree	Aşağıdan - Yukarıya	Hepsi	X+1
Spear	Prefix Tree	Aşağıdan - Yukarıya	Hepsi	2
Eclat	Yok	Aşağıdan - Yukarıya	Hepsi	>= 3
MaxClique	Yok	Melez	En fazla ve en az	>= 3

3.4.2. Paralel algoritmalar

Paralel Algoritmalar belleği kullanım şekline göre statik ve dinamik olmak üzere iki temel sınıfa ayrılırlar. Statik bellek kullanımında dağıtılmış ve paylaşılmış bellek yapısı kullanılırken, dinamik bellek kullanımında ise sadece sınıfsal bir yapı şekli kullanılır. Paralel algoritmalar bellek türüne göre incelenirken yine kendi içlerinde veri paralelliği ve işgücü paralelliği olmak üzere iki sınıfa ayrılırlar (Chattratichat 1997).

3.4.2.1. Dağıtılmış bellek algoritmaları

Dağıtılmış bellek sisteminde her bir işlemci için diğer işlemcilerden bağımsız özel bir bellek yapısı kullanılmaktadır. Dağıtılmış bellek sistemin kullanılmakta olan her bir işlemcinin sistem üzerindeki herhangi bir işlemi gerçekleştirme aşamasında bir birlerine karşı üstünlükleri yoktur.

1. Veri paralel algoritmalar:

- PEAR (Etkili Bölümlü Birliktelik Kuralları- Partitioned Efficient Association Rules) (Mueller 1995)

- PDM (Paralel Veri Madenciliği- Paralel Data Mining) (Park et al 1995)

- CD (Sayma Dağılımı – Count Distribution) (Agrawal and Shafer 1996)

- NPA (Bölümlenmemiş Apriori - Non Partitioned Apriori) (Shintani and Kitsuregawa 1996)

- FDM (Hızlı Dağıtılmış Madencilik - Fast Distributed Mining) (Cheung et al 1996)

- FPM (Hızlı Paralel Madencilik - Fast Parallel Mining) (Cheung and Xiao 1998)

2. İşgücü paralel algoritmalar:

- PPAR (Paralel Bölümlü Birliktelik Kuralları- Partitioned Parallel Association Rules) (Mueller 1995)

- DD (Veri Dağılımı – Data Distribution) ve CandDist (Aday Dağılımı – Candidate Distribution) (Agrawal and Shafer 1996)

- SPA (Basit Bölümlü Apriori - Simply Partitioned Apriori) ve HPA (Budama Bölümlü Apriori – Hash Partitioned Apriori) (Kitsuregawa and Shintani 1996)

- IDD (Zeki Veri Dağılımı – Intelligent Data Distribution) ve HD (Melez Dağılım – Hybrid Distribution) (Han et al. 1997)

3.4.2.2. Paylaşımlı bellek algoritmalar

Paylaşımlı bellek sisteminde dağıtılmış bellekten farklı olarak her bir işlemci için ayrı bir bellek yerine tüm işlemciler için ortak bir bellek kullanılır. Kullanılmakta olan bu sistemde sistem içindeki her bir işlemcinin sisteme erişirken eşit hakları vardır.

1. Veri paralel algoritmalar:

- CCPD (Ortak Aday Bölümlü Veritabanı - Common Candidate Partitioned Database) (Zaki et al 1996)

2. İşgücü paralel algoritmalar:

- PCCD (Bölümlü Aday Ortak Veritabanı - Partitioned Candidate Common Database) (Zaki et al 1996)

- APM (Eşzamanlı olmayan Paralel Madencilik - Asynchronous Parallel Mining) (Cheung 1998)

3.4.2.3. Sınıfsal algoritmalar

Bu sistemde ise dağıtılmış ve paylaşımlı bellek modellerinin her ikisi de kullanılır. Yalnız sınıfsal algoritmalar arasında veri paralel algoritmalar bulunmayıp sadece işgücü paralel algoritmalar kullanılmaktadır.

1. İşgücü paralel algoritmalar:

- ParEclat ve ParMaxEclat (Parallel Association Rules Equivalence Class-Based) (Zaki et al 1997)

- ParClique ve ParMaxClique (Parallel Association Rules Clique) (Zaki et al 1997)

3.4.2.4. Paralel algoritmaların özellikleri

Paralel algoritmaların özellikleri Şekil 3.4.'te özet olarak gösterilmiştir (Zaki 1999).

ALGORİTMA	ÖZELLİKLER
CD	Apriori temelli
PEAR	Aday ön ekli ağacı (Prefix tree)
PDM	Hash tablo
NPA	Sadece asıl toplamda indirgeme yapar.
FDM	Yerel ve genel budama, oy sayma
FPM	Yerel ve genel budama, taşıma eğriliği
CCPD	Paylaşımlı Bellek
DD	Tüm veritabanı tekrarlama başına değiştirilir.
SPA	DD ile aynıdır.
IDD	Halka temelli
PCCD	Paylaşımlı Bellek
HD	Sayma ve Veri dağıtımının birleştirir
Candidate D	Eşzamansız,Veri tekrarı olabilir
HPA	Veri tekrarı olmaz
HPA-ELD	Sık kullanılan öğeler tekrarlanır
ParEclat	Eclat temelli, eşzamansız,sınıfsal
ParMaxEclat	MaxEclat temelli, eşzamansız,sınıfsal
ParClique	Clique temelli, eşzamansız,sınıfsal
ParMaxClique	MaxClique temelli, eşzamansız,sınıfsal
APM	DIC temelli, paylaşımlı bellek, eşzamansız
PPAR	Bölümleme temelli, yatay veritabanı

Şekil 3.4. Paralel Algoritmaların Özellikleri.

BÖLÜM 4. ÖĞRENCİ BİLGİ SİSTEMİ DERS BİRLİKTELİKLERİ

Veri Madenciliğinde birliktelik kuralları kullanılarak büyük ölçekli veri tabanlarının incelenmesiyle bu veri tabanlarındaki ilginç birliktelikler ortaya çıkartılmaktadır.

Bu tez çalışmasında Kocaeli Üniversitesi Bilgi İşlemi tarafından yapılmış ve tüm üniversite tarafından kullanılmakta olan Öğrenci Bilgi Sistemi kaynak olarak seçilmiştir.

Kocaeli Üniversitesinde, öğrencilerin kayıt işlemlerinin yapılması, sınav sonuçlarının açıklanması, askerlik tecil belgesi, mezuniyet belgesi, diploma ve diğer matbu işlemlerde kullanılan belgelerin hazırlanması gibi işlemler internet üzerinden öğrencilerin her an erişilebileceği bir şekilde ÖBS' de mevcuttur.

ÖBS programında kullanılmakta olan veriler Macromedia ColdFusion ara yüz programının yardımıyla Microsoft SQL 2000 veritabanında muhafaza edilmek üzere işlenmektedir. Bu uygulamada mevcut veri tabanında bulunmakta olan veriler kullanılarak birliktelik kuralları gerçekleştirilmiştir.

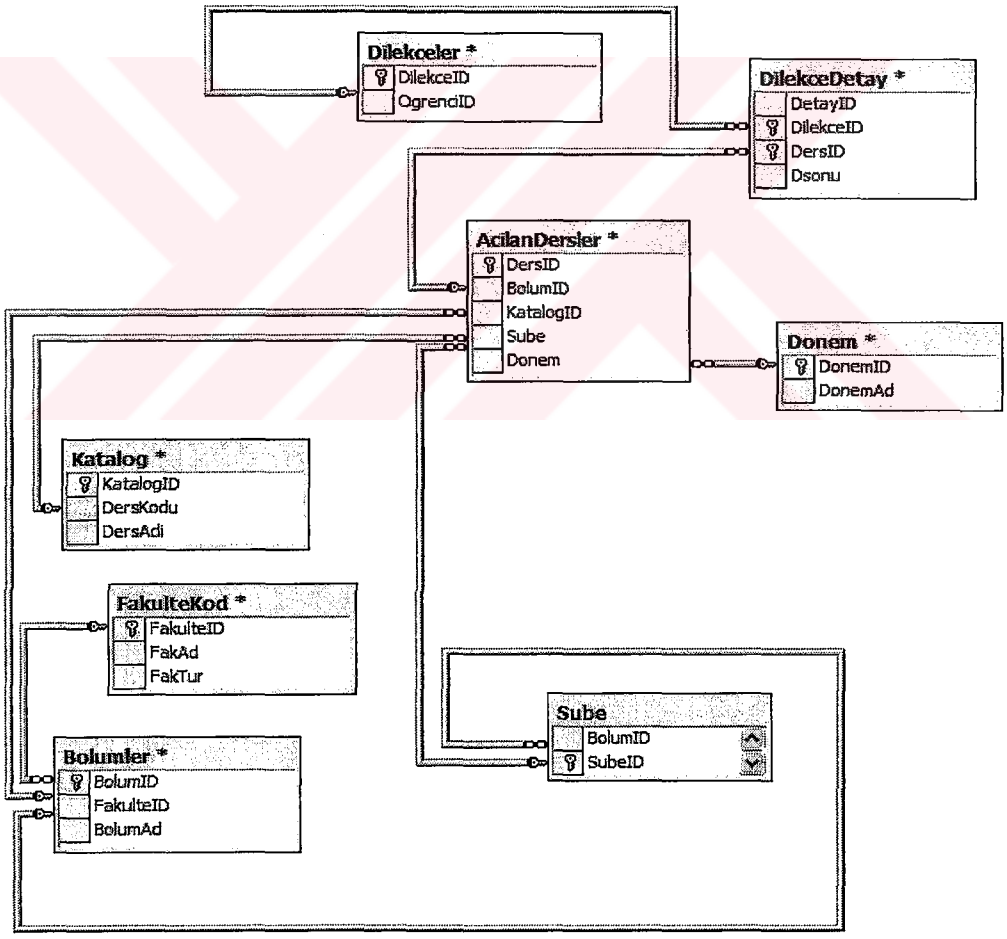
Tez çalışmamda ara yüzün hazırlanması aşamasında Delphi 6.0 programı kullanılmıştır. Veri tabanı olarak ÖBS' de olduğu gibi Microsoft SQL 2000 kullanılmış ve veriler burada oluşturulmuş olan ogrmining veritabanına aktarılmıştır.

Tez çalışmasında birliktelik kuralları oluşturulurken Srikant and Agrawal (1995) tarafından keşfedilmiş olan Apriori algoritması kullanılmıştır. Apriori algoritmasının kullanılmasında Han (2000) tarafından oluşturulmuş ve bölüm 3'te bahsetmiş olduğumuz Tablo 3.2'de izlenen yollar takip edilerek birliktelikler oluşturulmuştur.

Birliktelik kuralları kullanarak Kocaeli Üniversitesindeki öğrencilerin kalmış oldukları dersler hakkında mantıklı sonuçlar çıkarılarak bu derslerin birbirleri ile ilişkilendirilmeleri sağlanmıştır.

4.1. ÖBS Sistem Mimarisi

Tez çalışmasında ÖBS veritabanı ogrkou kullanılmıştır. Öncelikli olarak ogrkou veri tabanında bulunmakta gerekli tablolar, tezimde kullanmış olduğum ogrmining veritabanına aktarılmıştır. ÖBS' den almış olduğumuz işlenmemiş veri ile ilgili tablolar ve bunların varlık ilişki diyagramı (entity-relationship diagram) aşağıdaki Şekil 4.1.'de verilmiştir.



Şekil 4.1. ÖBS ilişki diyagram kesiti.

Şekil 4.1.'de kullanılmakta olan verilerin bulunduğu tabloların ilişkisel şeması verilmektedir. Bu şekilde sadece tez uygulamasında kullanılmakta olan alanlar gösterilmiştir.

FakulteKod ve Bolumler tablosu kullanılarak öğrencinin Üniversite bünyesinde okumakta olduğu fakülte ve bölüm adı bilgileri elde edilmektedir. Donem tablosunda dönem ile ilgili veriler, Sube tablosunda ise sınıf bilgisi bulunmaktadır.

AcilanDersler tablosunda bölümler tarafından açılmış olan dersler dönem bazında dersid olarak tutulmaktadır. Bu derslerin isim ve kodları ise Katalog tablosunda bulunmaktadır.

Bölümler tarafından açılmış olan dersler Dilekceler tablosu kullanılarak öğrenciler tarafından her dönem seçilmektedir. Bu alınan derslerden öğrencilerin almış oldukları notlar ise DilekceDetay tablosunda bulunmaktadır.

4.2. Verilerin Aktarılması

Bu çalışmada birliktelik kuralları oluşturulurken ÖBS' de bulunmakta olan 2002 ve 2003 yılı girişli öğrencilerin 1'inci ve 2'nci sınıfta başarısız oldukları dersler göz önüne alınarak bir değerlendirme yapılmıştır.

Apriori algoritması kullanılarak aday üretme işleminin daha hızlı yapılabilmesi için Şekil 4.2. ve 4.3.'deki YSD komutları kullanılarak oluşturulmuş olan sanal tablo (view)'dan elde edilmiş olan veriler Tablo 4.1.'teki Tblmine tablosuna aktarılmıştır. Böylece sık tekrarlanan aday kümelerinin elde edilme işlemi esnasında bu tablodaki verilerin kullanılması ile uygulamanın daha hızlı bir şekilde çalışması sağlanmıştır.

Verilerin Tablo 4.1.'e atanırken ön işleme sürecinden geçirilip tekrarlı olan veriler bu tabloya konulmamıştır. Örneğin, eğer öğrenci 1. sınıfta kaldığı dersten bir sonraki senede kalmışsa bu olay veri tekrarına neden olacağından dolayı bu veriler

silinmiştir. Çünkü burada amaç, bir öğrencinin bir dersten sadece ilk alışındaki başarısızlık durumunu göz önünde tutarak bir sonuça ulaşmaktır.

Tablo 4.1. TblMine tablosu

Alan ismi	**Veri tipi**
OgrID	char (9)
KatalogId	char (7)
BolumId	char (4)
YilId	int (4)

Yukarıdaki Tablo 4.1.'deki alanları inceleyecek olursak,

- OgrID: Öğrenci Bilgi Sistemindeki öğrenci numarası
- KatalogId: Bölümler tarafından açılmış olan dersin katalog numarası
- BolumId: Öğrencinin okumakta olduğu bölüm
- YilId: Öğrencinin giriş yılını belirtmektedir.

```
SELECT      Katalog.KatalogID AS Expr1, Ogrenciler.OgrenciID AS Expr2,
            Katalog.DersAdi, MIN(AcilanDersler.Donem) AS Expr3,
            Katalog.DersKodu, AcilanDersler.Sube
FROM   DilekceDetay INNER JOIN
    AcilanDersler ON DilekceDetay.DersID = AcilanDersler.DersID INNER JOIN
    Katalog ON AcilanDersler.KatalogID = Katalog.KatalogID INNER JOIN
    Dilekceler ON DilekceDetay.DilekceID = Dilekceler.DilekceID INNER JOIN
    Ogrenciler ON Dilekceler.OgrenciID = Ogrenciler.OgrenciID
WHERE  (DilekceDetay.DSonuHarf = 'F') AND (Dilekceler.OgrenciID LIKE '02%') AND
(SUBSTRING(AcilanDersler.Sube, 5, 1) IN ('1', '2'))
GROUP BY Katalog.DersAdi,AcilanDersler.Sube,Katalog.KatalogID,
Ogrenciler.OgrenciID,Katalog.DersAdi,Katalog.DersKodu,AcilanDersler.Sube
HAVING (MIN(AcilanDersler.Donem) = '0304G') OR
          (MIN(AcilanDersler.Donem) = '0304B') OR
          (MIN(AcilanDersler.Donem) = '0203G') OR
          (MIN(AcilanDersler.Donem) = '0203B')
ORDER BY Ogrenciler.OgrenciID, Katalog.KatalogID, MIN(AcilanDersler.Donem)
```

Şekil 4.2. 2002 Verilerinin elde edilmesi.

```
SELECT      Katalog.KatalogID AS Expr1, Ogrenciler.OgrenciID AS Expr2,
            Katalog.DersAdi, MIN(AcilanDersler.Donem) AS Expr3,
            Katalog.DersKodu, AcilanDersler.Sube
FROM   DilekceDetay INNER JOIN
    AcilanDersler ON DilekceDetay.DersID = AcilanDersler.DersID INNER JOIN
    Katalog ON AcilanDersler.KatalogID = Katalog.KatalogID INNER JOIN
    Dilekceler ON DilekceDetay.DilekceID = Dilekceler.DilekceID INNER JOIN
    Ogrenciler ON Dilekceler.OgrenciID = Ogrenciler.OgrenciID
WHERE  (DilekceDetay.DSonuHarf= 'F') AND (Dilekceler.OgrenciID LIKE '03%')
AND (SUBSTRING(AcilanDersler.Sube, 5, 1) IN ('1', '2'))
GROUP BY Katalog.DersAdi,AcilanDersler.Sube,Katalog.KatalogID,
Ogrenciler.OgrenciID,Katalog.DersAdi,Katalog.DersKodu,AcilanDersler.Sube
HAVING (MIN(AcilanDersler.Donem) = '0304G') OR
           (MIN(AcilanDersler.Donem) = '0304B') OR
           (MIN(AcilanDersler.Donem) = '0405G') OR
           (MIN(AcilanDersler.Donem) = '0405B')
ORDER BY Ogrenciler.OgrenciID, Katalog.KatalogID, MIN(AcilanDersler.Donem)
```

Şekil 4.3. 2003 Verilerinin elde edilmesi.

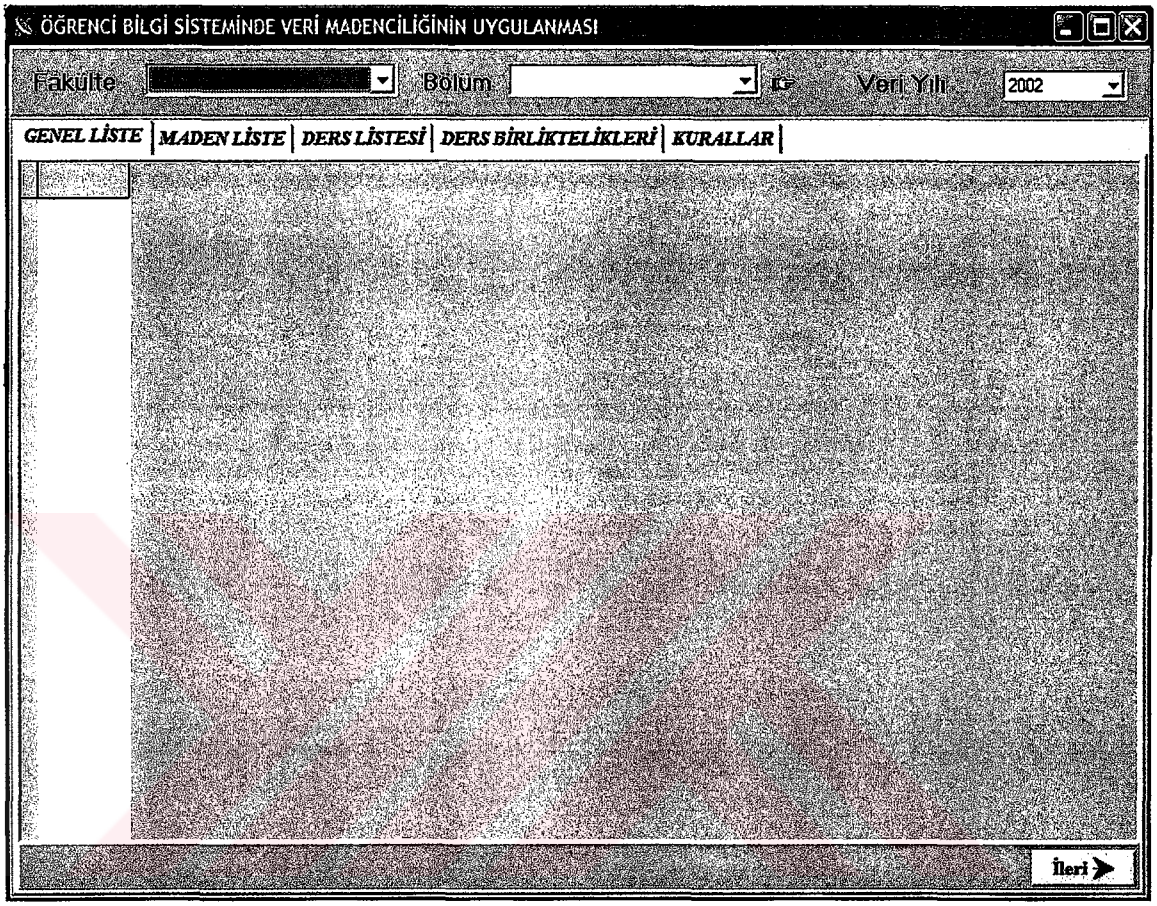
4.3. Veri Madenciliği Uygulama Platformu

Birliktelik kurallarının oluşturmak için Delphi 6.0'da hazırlanan uygulama ara yüzü bu bölümde adım adım anlatılmaktadır. Uygulama ara yüzünün giriş ekranı Şekil 4.4. gösterilmiştir. Giriş ekranında öğrencinin okumakta olduğu fakülte, bölüm ve öğrencinin giriş yılı olan veri yılı seçenekleri seçilerek uygulama programı başlatılmaktadır. Uygulama programında ara yüzün giriş sayfası 5 ayrı bölümden oluşmaktadır. Bunlar, Genel Liste, Maden Liste, Ders Listesi, Ders Birliktelikleri ve Kurallar bölümleridir.

Bu bölümlerde yapılan işlemler ve Algoritmanın nasıl çalıştığı Teknik Eğitim Fakültesi Bilgisayar Öğretmenliğinde okumakta olan 2002-2003 giriş yıllı öğrencilerin 1. ve 2.sınıf verileri kullanılarak anlatılmıştır.

Dersler arasındaki birlikteliklerin oluşturulmasında bölüm bazında işlem yapılarak daha sağlıklı verilerin elde edilerek kullanılabilir yargılara ulaşmak hedef seçilmiştir.

Ayrıca 2002 ve 2003 yılı girişli öğrenciler ayrıca değerlendirilerek ulaşılan sonuçların doğruluk payları sınanmıştır. Bu açıklamalar ışığında Şekil 4.4.'teki Fakülte, Bölüm ve Veri Yılının seçilmesi ile oluşan bir giriş ekranı tasarlanmıştır.

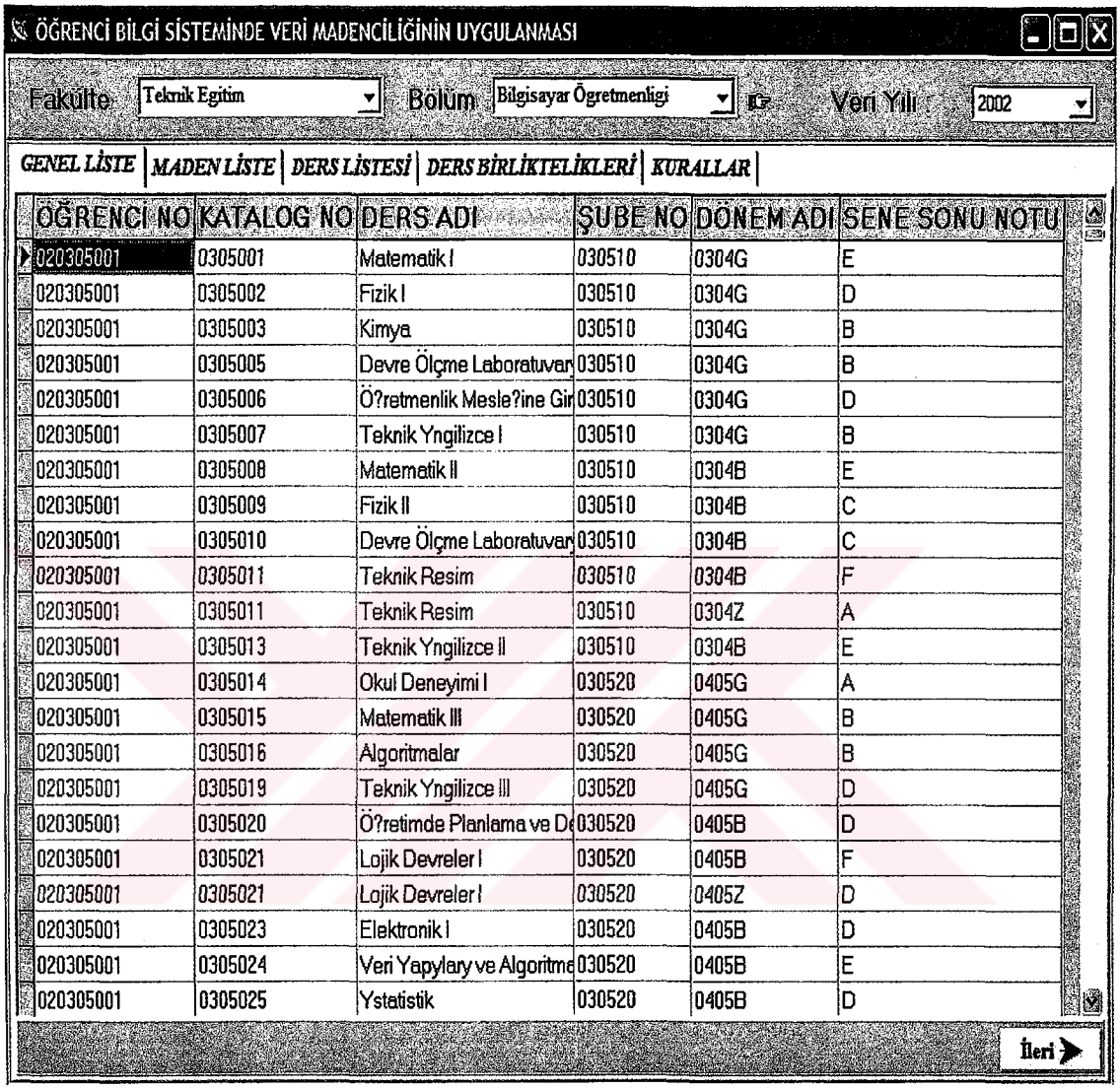


Şekil 4.4. Giriş ekranı.

4.3.1. Genel liste

Giriş ekranında seçilen bölüm ve yıla ait olan öğrencilerin okumuş oldukları süre içinde gördükleri derslerden almış oldukları tüm notlar gösterilmektedir. Bu ekranda verilerin eleme süreci öncesindeki ham veri hali verilmektedir. Bir başka deyişle, teknik eğitim fakültesi bilgisayar öğretmenliğinde okumakta olan 2002-2003 yılları girişli öğrencilerin şimdiye kadar almış oldukları tüm notlar dönem bazında gösterilmiştir. Şekil 4.5.'de Genel Liste ekranının bir kesiti verilmiştir. Burada öğrencilerin kaldıkları dersler değil şimdiye kadar almış oldukları tüm notlar

listelenmiştir. Yine sadece 1 ve 2'nci sınıf notları değil tüm notları ve bu aldıkları derslerin katalog numarası ve ders adları da listelenmiştir.



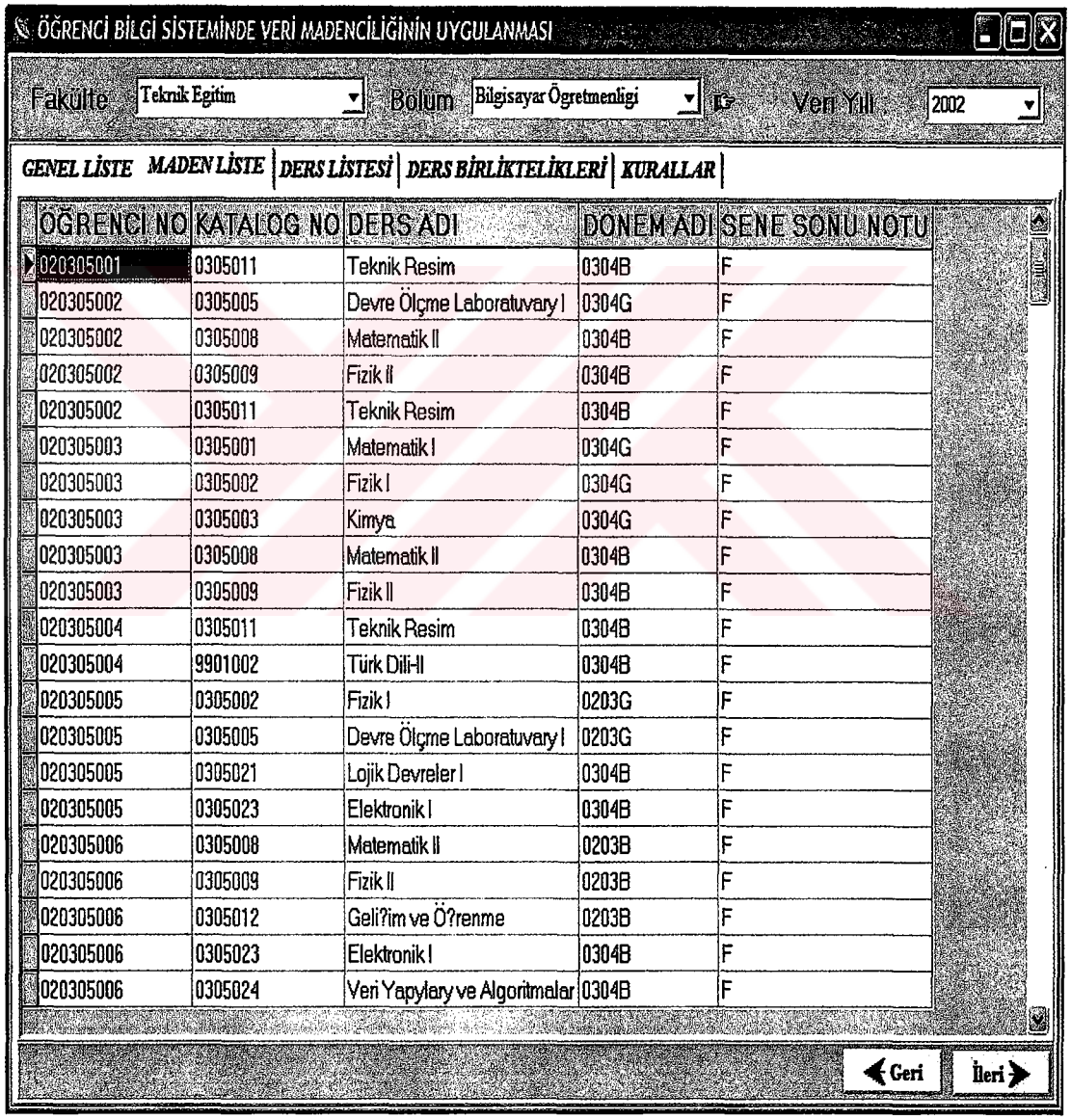
ÖĞRENCİ NO	KATALOG NO	DERS ADI	ŞUBE NO	DÖNEM ADI	SENE SONU NOTU
020305001	0305001	Matematik I	030510	0304G	E
020305001	0305002	Fizik I	030510	0304G	D
020305001	0305003	Kimya	030510	0304G	B
020305001	0305005	Devre Ölçme Laboratuvarı	030510	0304G	B
020305001	0305006	Ö?retmenlik Mesle?ine Gir	030510	0304G	D
020305001	0305007	Teknik Yngilizce I	030510	0304G	B
020305001	0305008	Matematik II	030510	0304B	E
020305001	0305009	Fizik II	030510	0304B	C
020305001	0305010	Devre Ölçme Laboratuvarı	030510	0304B	C
020305001	0305011	Teknik Resim	030510	0304B	F
020305001	0305011	Teknik Resim	030510	0304Z	A
020305001	0305013	Teknik Yngilizce II	030510	0304B	E
020305001	0305014	Okul Deneyimi I	030520	0405G	A
020305001	0305015	Matematik III	030520	0405G	B
020305001	0305016	Algoritmalar	030520	0405G	B
020305001	0305019	Teknik Yngilizce III	030520	0405G	D
020305001	0305020	Ö?retimde Planlama ve De	030520	0405B	D
020305001	0305021	Lojik Devreler I	030520	0405B	F
020305001	0305021	Lojik Devreler I	030520	0405Z	D
020305001	0305023	Elektronik I	030520	0405B	D
020305001	0305024	Veri Yapylary ve Algoritma	030520	0405B	E
020305001	0305025	Ystatistik	030520	0405B	D

Şekil 4.5. Genel liste.

4.3.2. Maden liste

Maden Liste ekranında, giriş ekranında seçilmiş olan bölüme ait olan veri yılındaki öğrencilerin 1'nci ve 2'nci sınıfta sadece başarısız oldukları dersler dönem olarak verilmektedir.

Maden Liste ekranında veri tabanında bulunmakta olan veriler bir süzgeçten geçirilerek ön eleme işlemi yapılmıştır. Bu ön eleme işlemi sonucunda, birliktelik kurallarını oluşturma esnasında kullanacağımız veriler elde edilmiştir. Bu verilerin elde edilme aşamasında öğrenci sadece dersi ilk defa almış olduğu zaman ve dönem değerlendirilmiştir. Öğrencinin bütünleme ve yaz okulundaki almış olduğunu notlar verinin güvenilirliğini azaltacağından dikkate alınmamıştır. Maden Liste ekranı aşağıdaki Şekil 4.6.'da gözükmektedir.

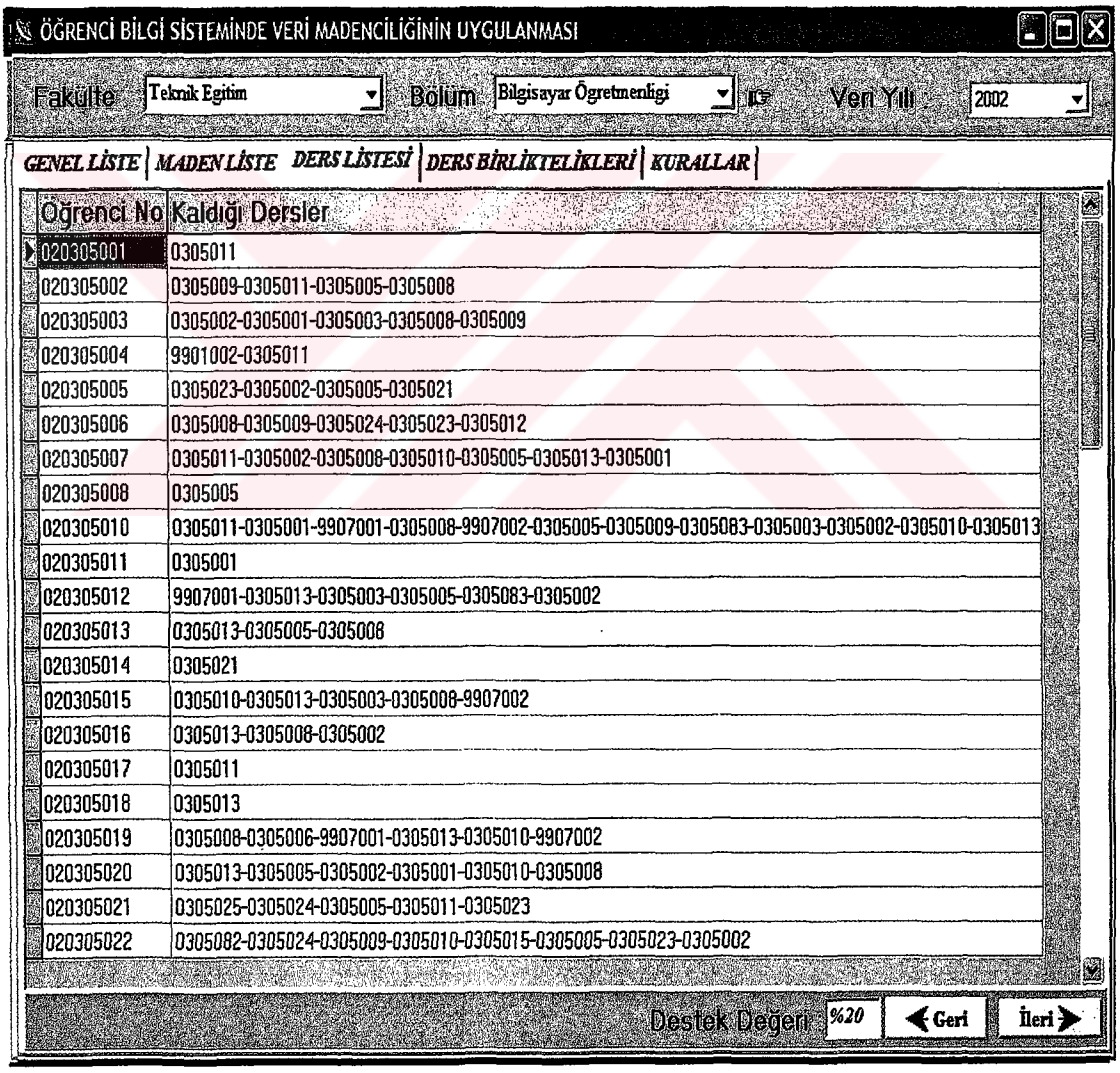


ÖĞRENCİ NO	KATALOG NO	DERS ADI	DÖNEM ADI	SENE SONU NOTU
020305001	0305011	Teknik Resim	0304B	F
020305002	0305005	Devre Ölçme Laboratuvary I	0304G	F
020305002	0305008	Matematik II	0304B	F
020305002	0305009	Fizik II	0304B	F
020305002	0305011	Teknik Resim	0304B	F
020305003	0305001	Matematik I	0304G	F
020305003	0305002	Fizik I	0304G	F
020305003	0305003	Kimya	0304G	F
020305003	0305008	Matematik II	0304B	F
020305003	0305009	Fizik II	0304B	F
020305004	0305011	Teknik Resim	0304B	F
020305004	9901002	Türk Dili-II	0304B	F
020305005	0305002	Fizik I	0203G	F
020305005	0305005	Devre Ölçme Laboratuvary I	0203G	F
020305005	0305021	Lojik Devreler I	0304B	F
020305005	0305023	Elektronik I	0304B	F
020305006	0305008	Matematik II	0203B	F
020305006	0305009	Fizik II	0203B	F
020305006	0305012	Geli?im ve Ö?renme	0203B	F
020305006	0305023	Elektronik I	0304B	F
020305006	0305024	Veri Yapylary ve Algoritmalar	0304B	F

Şekil 4.6. Maden liste.

4.3.3. Ders listesi

Ders Listesi ekranında maden listede elde ettiğimiz sonuçlar kullanılarak her bir öğrencinin kalmış olduğu tüm dersler katalog numarası ile birlikte listelenmektedir. Böylece ders listesi ekranında sık kullanılan aday öğe kümelerinin elde edileceği veri tablosu oluşturulmuştur. Bu tablo üzerinde destek değeri girilerek sık kullanılan aday kümeleri destek değerleri ile birlikte elde edilmiştir. Şekil 4.7.'de Teknik Eğitim Fakültesi Bilgisayar Öğretmenliğine 2002 yılında girmiş olan öğrencilerin numaraları ve kalmış oldukları derslerin katalog numaraları sırasıyla listelenmiştir.



Öğrenci No	Kaldığı Dersler
020305001	0305011
020305002	0305009-0305011-0305005-0305008
020305003	0305002-0305001-0305003-0305008-0305009
020305004	9901002-0305011
020305005	0305023-0305002-0305005-0305021
020305006	0305008-0305009-0305024-0305023-0305012
020305007	0305011-0305002-0305008-0305010-0305005-0305013-0305001
020305008	0305005
020305010	0305011-0305001-9907001-0305008-9907002-0305005-0305009-0305083-0305003-0305002-0305010-0305013
020305011	0305001
020305012	9907001-0305013-0305003-0305005-0305083-0305002
020305013	0305013-0305005-0305008
020305014	0305021
020305015	0305010-0305013-0305003-0305008-9907002
020305016	0305013-0305008-0305002
020305017	0305011
020305018	0305013
020305019	0305008-0305006-9907001-0305013-0305010-9907002
020305020	0305013-0305005-0305002-0305001-0305010-0305008
020305021	0305025-0305024-0305005-0305011-0305023
020305022	0305082-0305024-0305009-0305010-0305015-0305005-0305023-0305002

Şekil 4.7. Ders listesi.

Şekil 4.7.'deki ders listesi ekranında görmüş olduğumuz kesitteki toplam 49 öğrenci en az bir dersten kalmıştır. Başka bir deyişle kullanılmakta olan toplam işlem adeti 47'dir. Şekil 4.7.'deki gibi destek değeri yüzdesini %20 olarak varsaydığımızda daha sonradan sık kullanılan aday kümelerini belirlemede kullanacağımız destek değerine ulaşırız. Destek değerinin hesaplanması aşağıda verilmiştir.

- Destek Değeri = Destek Değeri Yüzdesi * İşlem Adeti
- Destek Değeri = 0.2 * 47 = 9

Ders Listesi ekranında destek değerini ne kadar yüksek olarak belirleyebilirsek oluşturacak olduğumuz kurallarımızın güvenirliği de o oranda artmış olacaktır.

4.3.4. Ders birliktelikleri

Ders Listesi ekranında seçmiş olduğumuz %20 destek değerine göre, ders birliktelikleri ekranında en az 9 öğrenci tarafından kalınmış olan dersler göz önünde bulundurularak aday öğe kümeleri oluşturulur. Aşağıda bu işlemlerin nasıl gerçekleştirildiği örneklenerek anlatılmıştır.

Ders Listesinde daha önceden oluşturmuş olduğumuz dersler veritabanın ilk taranması sonucunda ilk olarak 1 elemanlı aday öğe kümesini üretilmesinde kullanılır. Burada kullanmış olduğumuz örnekte 47 kayıt içinde en az bir kere bulunan toplam 27 adet dersin katalog numarası ve destek değerleri aşağıdaki Tablo 4.2'de 1 elemanlı aday kümesi olarak verilmiştir.

Tablo 4.2.'deki derslerden olan {0305003, 0305006, 0305007, 0305012, 0305015, 0305016, 0305017, 0305019, 0305021, 0305023, 0305024, 0305025, 0305082, 0305083, 9901002, 9905001, 9905002, 9907001, 9907002} katalog numaralı dersler destek değeri olan 9'un altında olduklarından 1 elemanlı aday kümesinden elenmişlerdir.

Tablo 4.2. 1 Elemanlı aday kümesi.

Öğe Kümesi	Destek Değeri
0305001	18
0305002	17
0305003	6
0305005	18
0305006	5
0305007	3
0305008	20
0305009	15
0305010	13
0305011	14
0305012	3
0305013	14
0305015	6
0305016	2
0305017	2
0305019	1
0305021	4
0305023	6
0305024	8
0305025	3
0305082	3
0305083	6
9901002	3
9905001	2
9905002	2
9907001	6
9907002	7

Budama işlemi sonrasında elde edilen 1 Elemanlı sık geçen aday kümesi Tablo 4.3.'de gösterilmektedir.

Tablo 4.3. 1 Elemanlı sık geçen aday kümesi.

Öğe Kümesi	Destek Değeri
0305001	18
0305002	17
0305005	18
0305008	20
0305009	15
0305010	13
0305011	14
0305013	14

2 Elemanlı Aday kümesini elde etmek için Tablo 4.3'deki 1 elamanlı sık geçen aday kümesi öğelerinin çaprazlama işlemi yapılır. ({0305001, 0305002, 0305005, 0305008, 0305009, 0305010, 0305011, 0305013} ⋈ {0305001, 0305002, 0305005, 0305008, 0305009, 0305010, 0305011, 0305013}={0305001 0305002, 0305001 0305005, 0305001 0305008, 0305001 0305009, 0305001 0305010, 0305001 0305011, 0305001 0305013, 0305002 0305005, 0305002 0305008, 0305002 0305009, 0305002 0305010, 0305002 0305011, 0305002 0305013, 0305005 0305008, 0305005 0305009, 0305005 0305010, 0305005 0305011, 0305005 0305013, 0305008 0305009, 0305008 0305010, 0305008 0305011, 0305008 0305013, 0305009 0305010, 0305009 0305011, 0305009 0305013, 0305010 0305011, 0305010 0305013, 0305011 0305013}).

Bu iki tablonun çaprazlaması işlemi sonucunda yukarıda elde edilmiş olan toplam 29 adet 2 elemanlı aday kümesi aşağıda Tablo 4.4.'de destek değerleri ile birlikte verilmiştir.

Tablo 4.4. 2 Elemanlı aday kümesi.

Öğe Kümesi	Destek Değeri
0305001 0305002	11
0305001 0305005	9
0305001 0305008	11
0305001 0305009	8
0305001 0305010	7
0305001 0305011	5
0305001 0305013	6
0305002 0305005	12
0305002 0305008	12
0305002 0305009	8
0305002 0305010	9
0305002 0305011	4
0305002 0305013	8
0305005 0305008	11
0305005 0305009	8
0305005 0305010	7
0305005 0305011	6
0305005 0305013	8
0305008 0305009	10
0305008 0305010	12
0305008 0305011	6
0305008 0305013	11
0305009 0305010	6
0305009 0305011	5
0305009 0305013	4
0305010 0305011	4
0305010 0305013	8
0305011 0305013	4

Tablo 4.4.'teki 2 elemanlı toplam 29 adet olan aday kümelerinden destek değeri olarak daha önceden hesaplamış olduğumuz bir dersten en az 9 kişinin kalması eşik değerinin altında olanların budanması ile elde edilen 2 elemanlı sık geçen aday öğe kümesi aşağıda Tablo 4.5.'de verilmiştir.

Tablo 4.5. 2 Elemanlı sık geçen aday kümesi.

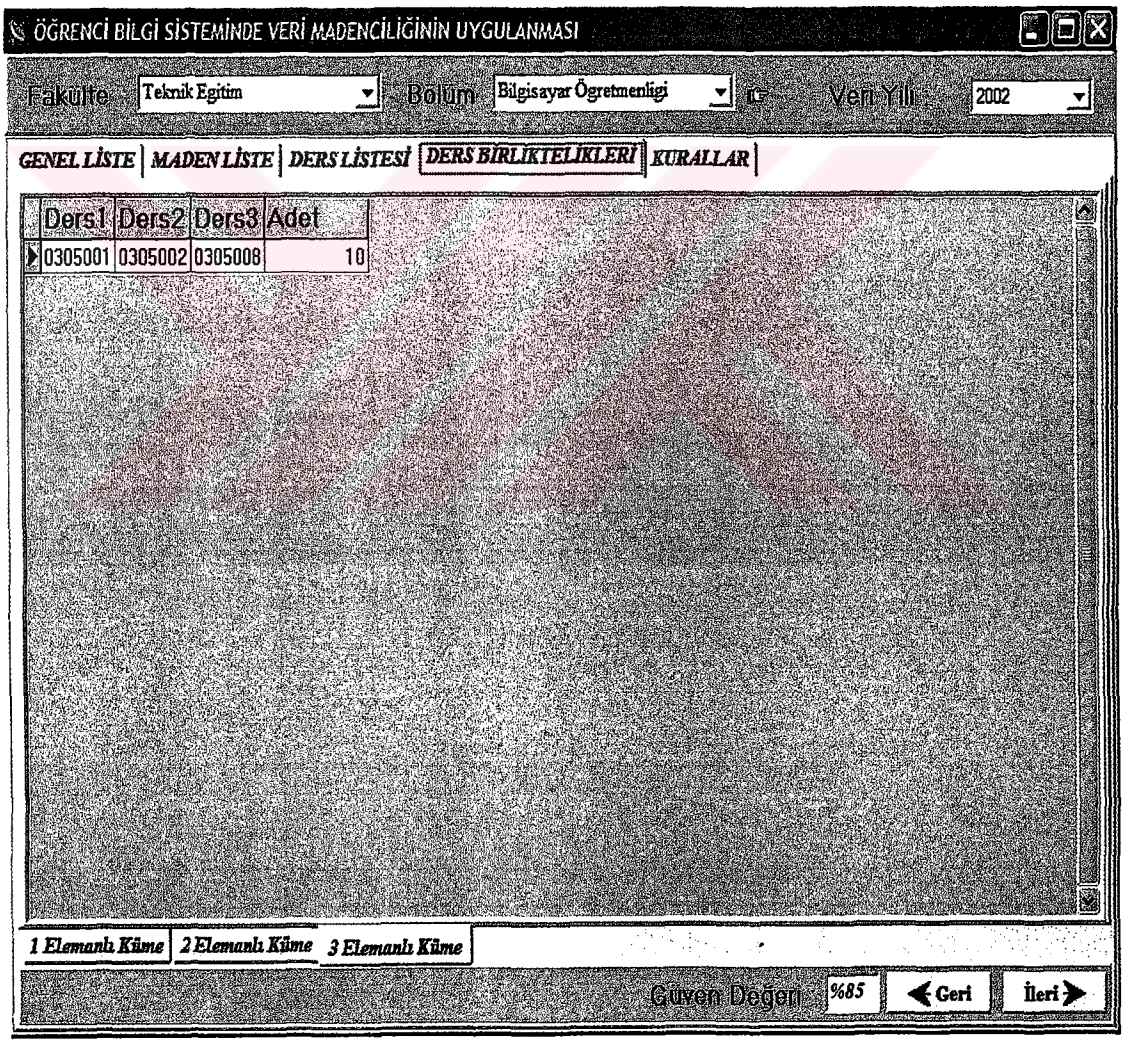
Öğe Kümesi	Destek Değeri
0305001 0305002	11
0305001 0305005	9
0305001 0305008	11
0305002 0305005	12
0305002 0305008	12
0305002 03050010	9
0305005 0305008	11
0305008 0305009	10
0305008 0305010	12
0305008 0305013	11

2 Elemanlı Aday kümesinin çaprazlama işlemi sonucunda toplam 55 adet 3 elemanlı aday kümesi elde edilmiştir. Bu çaprazlama sonucunda elde edilen 3 elemanlı aday kümelerinden sadece 0305001 0305002 0305005, 0305001 0305002 0305008, 0305001 0305005 0305008 ve 0305002 0305005 0305008 katalog numaralı 3 elemanlılarının alt kümeleri 2 elemanlı sık geçen aday kümesinde bulunduğundan diğerleri budanır. Budanma sonucunda destek değeri sırasıyla 7,8,8 olan 0305001 0305002 0305005, 0305001 0305005 0305008, 0305002 0305005 0305008 katalog numaralı 3 elemanlı aday kümeleri minimum destek değeri olan 9 adetin altında olduğundan elenirler. Minimum destek değeri 9'un üzerinde olan 0305001 0305002 0305008 katalog numaralı 3 elemanlı sık geçen öğe kümesi ve destek değeri Tablo 4.6.'da gösterilmiştir.

Tablo 4.6. 3 Elemanlı sık geçen aday kümesi.

Öğe Kümesi	Destek Değeri
0305001 0305002 0305008	10

Tablo 4.6.'da sadece 1 adet 3 elemanlı öğe kümesi olduğundan çaprazlama yapılarak 4 elemanlı aday kümesi üretilememiştir. Sonuç olarak 4 elemanlı aday kümesi ϕ olduğundan aday üretme işlemi sonlandırılır ve Şekil 4.8.'de görüldüğü gibi güven değeri girilerek kuralların çıkarılması işlemi yapılır. Bu örnekte güven değeri 85% olarak girilmiştir.



Şekil 4.8. Ders birliktelikleri.

4.3.5. Kurallar

Ders Birliktelikleri ekranında elde edilen 3 elemanlı sık geçen aday öğe kümesindeki 0305001 0305002 0305008 bağıntısı kullanılarak Tablo 4.7.'deki birliktelik kurallarını çıkarırız. 3 elemanlı sık geçen aday kümesi için toplam 6 adet birliktelik sonucu çıkmaktadır. Burada elde ettiğimiz bağıntı kuralları aşağıda Tablo 4.7.'de verilmiştir.

Tablo 4.7. Birliktelik kuralları.

Birliktelik	**Güven**
$0305001 \wedge 0305002 \Rightarrow 0305008$	güven=10/11=91%
$0305001 \wedge 0305008 \Rightarrow 0305002$	güven=10/11=91%
$0305002 \wedge 0305008 \Rightarrow 0305001$	güven=10/12=83%
$0305001 \Rightarrow 0305002 \wedge 0305008$	güven=10/18=56%
$0305002 \Rightarrow 0305001 \wedge 0305008$	güven=10/17=59%
$0305008 \Rightarrow 0305001 \wedge 0305002$	güven=10/20=50%

Bu birliktelik kurallarında katalog numaraları verilen derslerin adları ise aşağıdaki Tablo 4.8.'de verilmiştir.

Tablo 4.8. Katalog tablosu

Katalog ID	**Ders Adı**
0305001	Matematik-I
0305002	Fizik-I
0305008	Matematik-II

Güven değeri % 85'i geçen kurallar ogrmining veritabanında oluşturulmuş olan Tablo 4.9.'daki TblKural tablosunda muhafaza edilmektedir. Tablo 4.9.'da kullanılan alanlar ve ne için kullanıldıkları aşağıda kısaca özetlenmiştir:

- KuralID : Oluşturulmuş olan kural numarasını
- G1 : Kuralın sol tarafını
- G2 : Kuralın sağ tarafını
- Oran : Güven değerini temsil etmektedir.

Tablo 4.9. TblKural tablosu

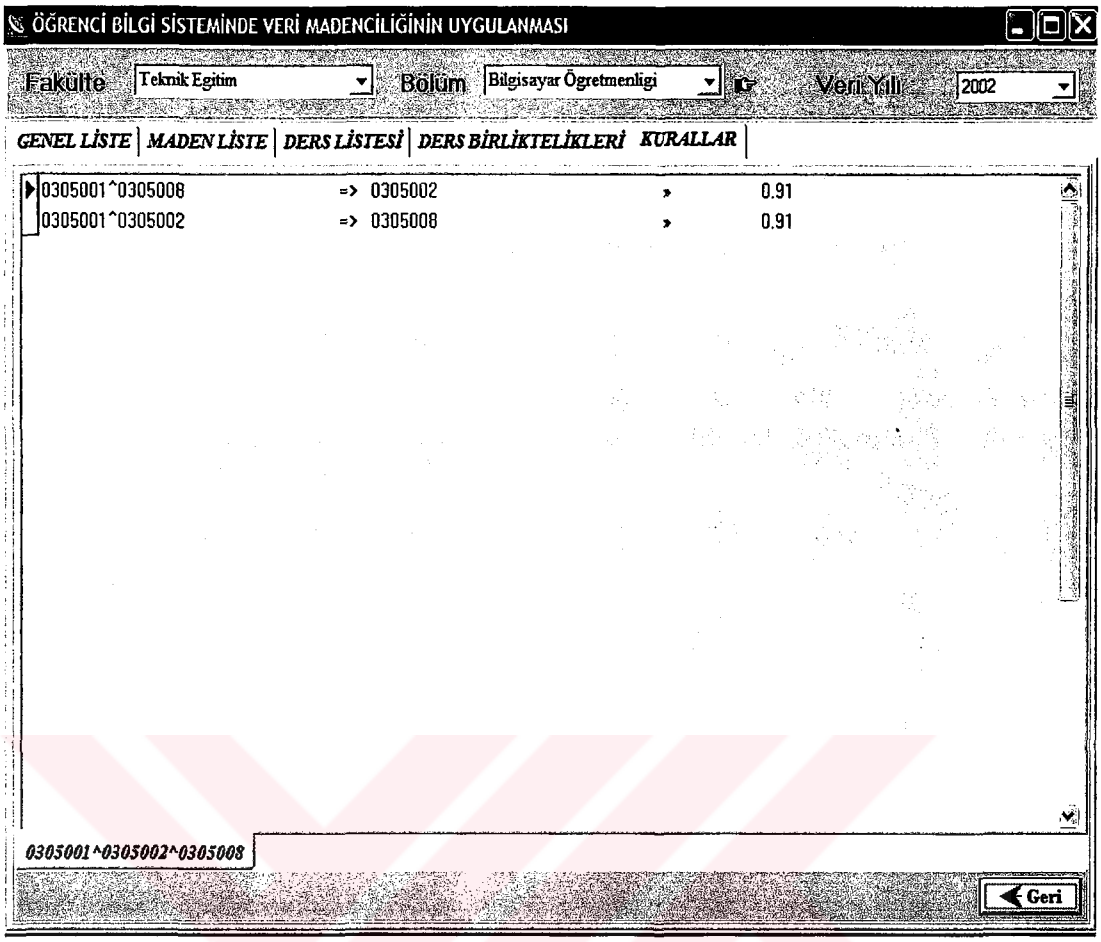
Alan ismi	**Veri tipi**
KuralID	int (4)
G1	varchar (200)
G2	varchar (200)
Oran	numeric (5)

Ders birliktelikleri ekranında yazılmış olan 85% güven değerine göre elde edilen sonuçlar Şekil 4.9.'da kurallar ekranında gösterilmiştir. Şekil 4.9.'da elde edilen birliktelik kurallarını katalog numarası yerine ders adı ile Tablo 4.10.'daki gibi gösterebiliriz.

Tablo 4.10. Ders bağıntıları

Matematik-I $\wedge$ Matematik-II $\Rightarrow$ Fizik-I
Matematik-I $\wedge$ Fizik-I $\Rightarrow$ Matematik-II

Tablo 4.10. ve Şekil 4.9.'u' beraberce değerlendirdiğimizde buradaki iki bağıntıdan ikincisinden mantıklı bir sonuça ulaşmamız mümkündür. Sonuçta, 2002 yılı girişli Teknik Eğitim Fakültesi öğrencilerinin almış oldukları sene sonu notlarına göre, I. dönem dersi olan Matematik-I ve Fizik-I derslerinden kalan öğrencilerin %91'inin II. dönem dersi olan Matematik-II' den kaldıkları görülmektedir.

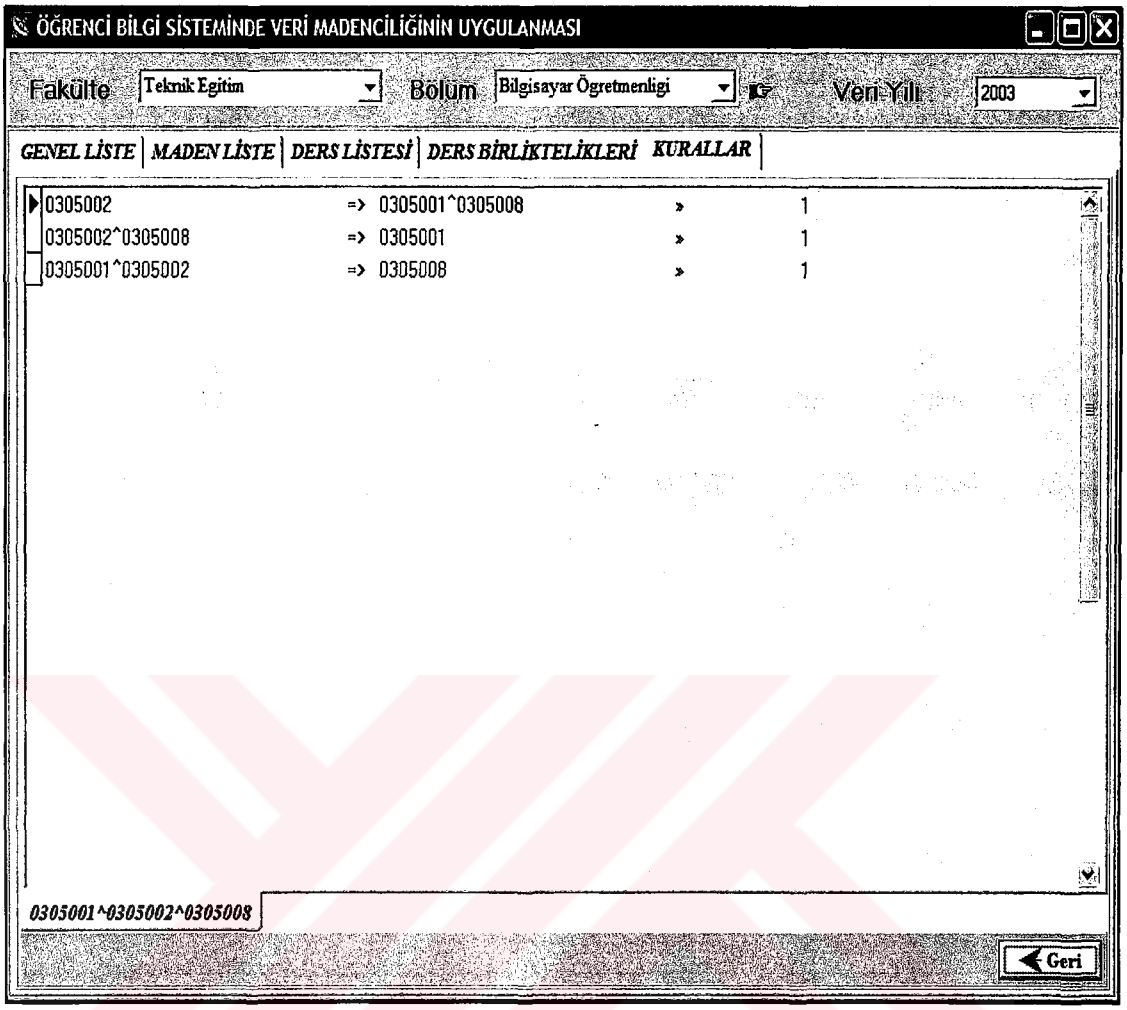


Şekil 4.9. Kurallar.

4.3.6. 2002 ve 2003 yılları karşılaştırması

Teknik Eğitim Fakültesi Bilgisayar Öğretmenliğinde okumakta olan 2003 yılı girişli öğrenciler için aynı destek ve güven değerlerini kullanarak oluşturmuş olduğumuz birliktelik kurallarını incelediğimizde, 2002 yılı verilerine göre elde ettiğimiz 0305001 0305002 0305008 katalog numaralı I. Dönem dersi olan Matematik-I ve Fizik-I, II. Dönem dersi olan Matematik-II dersleri arasında bağıntı oluştuğunu görmekteyiz. Bu üç ders arasındaki oluşan yeni bağıntı kuralları Şekil 4.10.'da gösterilmektedir.

2003 yılı sonuçlarını değerlendirdiğimizde yine 2002'de olduğu gibi Matematik-I ve Fizik-I derslerinden kalan öğrencilerin 100% oranında Matematik-II dersinden kalmış olduklarını gözükmektedir.



Şekil 4.10. Kurallar (2003).

Bu birliktelik kuralını seçerek, bu bölümde okuyan herhangi bir öğrenci II. dönem derslerini seçerken eğer kredi sorunu ile karşılaşmışsa Matematik-II dersi yerine başka bir ders seçmesinin daha mantıklı olacağı yargısına danışmanı tarafından karar verilip öğrencinin II.dönemdeki ders yükü azaltılarak daha başarılı olması sağlanabilir.

BÖLÜM 5. SONUÇ ve ÖNERİLER

Dünyamızda bilişim sektöründeki gelişmeler büyük bir hızla artmakta ve bilgi teknolojisi ile hayatımızın her aşamasında karşılaşmaktayız. Sağlık, eğitim, ticaret, askeri alanlar, alışveriş, devlet sektörü, özel sektör ve burada sayamadığımız çeşitli alanlarda artık verilerin işlenmesi ve bu verilerin değerlendirilerek bilgi haline getirilmesi bir zaruret haline gelmiştir. Böylece bu alanlardaki gelişmeler daha hızlı ve verimli bir şekilde gerçekleşebilmektedir.

Bu tez çalışmasında birliktelik kuralları analiz yöntemlerimden en sık ve yaygın olarak kullanılmakta olan Apriori algoritması kullanılarak bir uygulama programı yapılmıştır. Genellikle Market sepeti analizi yönteminde kullanılan birliktelik analizi burada Kocaeli Üniversitesi ÖBS' deki kayıtlar göz önüne alınarak bir çalışma gerçekleştirilmiştir.

Bu çalışmada, Kocaeli Üniversitesinde okumakta olan 2002-2003 yılı girişli tüm öğrencilerin 1. ve 2.sınıfta almış oldukları notlar değerlendirilmiştir. Öğrencilerin başarısız oldukları dersler değerlendirilerek bu dersler arasındaki birliktelik kuralları ortaya çıkarılmıştır.

Birliktelik kurallarının ortaya çıkarılmasında çok önemli iki aşama karşımıza çıkmaktadır. Bunlardan birincisi, daha önceden belirlemiş olduğumuz destek ve güven değerlerine göre birliktelik kurallarını oluşturmaktır. Destek değeri ve güven değerinin artması ile birliktelik kuralının güvenilirliği de artmaktadır. İkincisi ise, ortaya çıkan birliktelikleri değerlendirerek bu birliktelik kurallarından kullanılabilir olanları ayıklamaktadır. Çünkü Veri Madenciliği sonucunda oluşturmuş olduğumuz tüm kurallar kesinlikle kullanılabilir veya kullanılamazdır diye bir yargıya önceden varmamız mümkün değildir.

Bu çalışmada ortaya çıkarılan birliktelik kuralı öğrencinin danışmanı tarafından değerlendirilip öğrenciler için gelecekte ders seçimi aşamasında bir kıstas olarak değerlendirilebilir. Örneğin, öğrenci 3. sınıfa geçmiş ve alttan bir çok dersi olduğu için bu derslerden bazılarını almaması gerektiği bir durumda oluşturulmuş olan ders birlikteliklerinden biri kullanılarak bazı dersleri seçmemesi için danışmanı tarafından yönlendirilebilir. Öğrenci başarısız olabileceği bir dersi seçmek yerine başka bir derse yönlendirilir ve böylece klasik olarak belirlenen öncelikli olarak alttan olan dersi alma gerekliliği ortadan kalkmış olur.

Ayrıca bu tez çalışmasındaki birliktelikler gözetilerek Bölümler tarafından öğrencilerin başarısız oldukları derslerdeki birliktelikler önceden değerlendirilerek bu durumun ortadan kaldırılmasına yönelik değişik stratejiler belirlenmesi hususunda bir karar verilebilir.

Bu tezin en önemli özelliklerinden birisi sadece bir bölüm tarafından değil Kocaeli Üniversitesindeki tüm bölümler tarafından uygulanabilir olmasıdır. Üniversite bünyesindeki tüm bölümlere ait veriler burada kullanılan veritabanına aktarılırmıştır.

Kocaeli Üniversitesinin farklı bölümlerindeki öğrencilerin aynı seviyede olmaları beklenemez. Örneğin, Türk Dili Edebiyatı ve Matematik bölümündeki iki farklı öğrencinin Matematik dersindeki başarı durumlarının farklılık göstermesi muhtemeldir. Yani oluşturulmuş olan bir birliktelik kuralının farklı bölümlerde farklı sonuçlar vermesi olağandır. Bu uygulamada ders birliktelikleri bölüm bazında yapılarak her bir bölüm için ayrı bir birliktelik kuralının ortaya çıkarılması esnekliği sağlanmıştır.

Birliktelik kurallarının oluşturulması aşamasında sadece 2002-2003 yılı girişli öğrencilerin 1. ve 2.sınıfta almış oldukları notların değerlendirilmiştir. Kocaeli Üniversitesi ÖBS' deki veriler ileride 4 yılı içerecek şekilde olduğunda uygulama veritabanına bu yıllardaki verilerin eklenmesi gerekecektir. Diğer yandan öğrencilerin 4 sene boyunca almış oldukları notların eklenmesi ile tüm dersler için birliktelik kuralları oluşturulmuş olacaktır. Örneğin 2002 ve 2003 yılı öğrencilerin 4 yıl boyunca almış oldukları derslerin değerlendirilip karşılaştırılmaları sonucunda

dersler arasında tam bir birliktelik sağlanmış olacaktır. Daha sonradan bu birliktelikler 2004-2005 girişli öğrencilerde göz anına alınarak yapılan bir değerlendirme sonucunda tam olarak test edilebileceklerdir.

Bu tez çalışmasında birliktelik kuralları oluşturulurken dersi veren öğretim üyelerine göre sonuçların değişim durumu gözlenmemiştir. Örneğin bir bölümde 2002 yılında Fizik dersini veren öğretim üyesindeki sonuçlar ile 2003 yılında başka bir öğretim üyesi tarafından verildiğinde farklılıklar olabilir. Bu çalışmada birliktelik kuralları oluşturulurken bu konu dikkate alınmamıştır. Dersi veren kişinin değişmesi halini göz önüne alabilecek eklentilerin kuralların daha tutarlı olması için eklenmesi beklenebilir.

KAYNAKLAR

1. AGRAWAL,R., 1994. Fast algorithms for mining association rules. Proceedings of the 20th VLDB conference, 487-499.

2. AGRAWAL,R., IMIELINSKI,T., SWAMI,A., 1993. Mining association rules. Between sets of items in large databases. Proceedings of ACM SIGMOD international conference on management of data, 207-216.

3. AGRAWAL,R., SHAFER,J., 1996. Parallel Mining of Association Rules. IEEE Transactions on Knowledge and Data Engineering,Vol.8, No.6, 962-969.

4. AKPINAR,H., 2004. Veri Tabanlarında Bilgi Keşfi ve Veri Madenciliği, İ.Ü. İşletme Fakültesi Dergisi, C:29, S: 1, s.1-22.

5. ALPAYDIN,E., 2000. Zeki Veri Madenciliği: Ham Veriden Altın Bilgiye Ulaşma Yöntemleri, Bilişim 2000 Veri Madenciliği Eğitim Semineri.

6. BAYES,T., 1764. An Essay Toward Solving a Problem in the Doctrine of Chances. Philosophical Transactions of the Royal Society, London, Vol.53, 370-418.

7. BERRY,M.J.A., LINOFF,G., 1997. Data Mining Techniques: For Marketing, Sales and Customer Support, John Wiley and Sons Ltd., ISBN: 0-4171-17980-9, USA.

8. BIGUS,J.P., 1996. Data Mining with Neural Networks: Solving Business Problems – From Application Development to Decision Support, McGraw Hill Book Co Ltd, ISBN: 0070057796.

9. BRIN,S., MOTWANI,R., ULLMAN,J.D., TSUR,S., 1997. Dynamic Item set Counting and Implication Rules for Market Basket Data. Proceedings ACM SIGMOD International Conference on Management of Data, ACM Pres, New York, 1997, 255-264.

10. CABENA,P., HADJNIAN, STADLER, VERHEES, ZANASI, 1997.Discovering Data Mining From Concept to Implementation, Prentice Hall.

11. CHAN,K.C.C., WONG,A.K.C., 1991.A statistical technique for extracting classificatory knowledge from databases. In Piatetsky-Shapiro, G., and Frawley, W.J., Knowledge and Discovery in Databases, Cambridge, MA:AAAI/MIT Press, 107-123.

12. CHATTRATICHAT,J., DARLINGTON,J., GHANEM,M., GUO,Y., HUNING,H., KOHLER,M., SUTIWARAPHUN,J., TO,H.W., YANG,D., 1997. Large Scale Data Mining: Challenge and Responses. 3th International Conference on Knowledge Discovery and Data Mining, AAAI Press, 143-146.

13. CHEN,Y.L., TANG,K., SHEN,R.J., HU,Y.H., 2005. Market basket analysis in a multiple store environment. Elsevier, Decision Support Systems 40 (2005), 339-354.

14. CHEUNG,D., HAN,J., NG,V., FU,W., FU,Y., 1996. A Fast Distributed Algorithm for Mining Association Rules. Proceedings 4th International Conference Parallel and Distributed Information Systems, IEEE CS. Press, Los Alamitos, California, 31-42.

15. CHEUNG,D., XIAO,Y., 1998.Effect of Data Skew ness in Parallel Mining of Association Rules. Proceedings Pasific-Asia Conference Knowledge Discovery and Data Mining, Lecture notes in Computer Science, Vol.1394, Springer, New York, 48-60.

16. CHEUNG,D., HU,K., XIA,S., 1998. Asynchronous Parallel Algorithm for Mining of Association Rules on Shared-Memory Multi-Processors. Proceedings 10th ACM Symp. Parallel Algorithms and Architectures, ACM Press, New York, 279-288.

17. CODD,E.F., CODD,S.B., SALLEY,C.T., 1993. Providing OLAP (On-line Analytical Processing) to User Analysts: An IT Mandate. E.F Codd Associates.

18. DATE,C.J., 1994. An Introduction to Database Systems. Addison Wesley Publishing Company, USA.

19. DORIGO,M., MANIEZZO,V., COLORNI,A., 1991. The Ant System: An autocatalytic optimizing process. Technical Report, Politecnico, Milano, Italy, 91-106.

20. ELDER,J.F., PREGIPON,D., 1995. A statistical perspective on KDD. U.Fayyad and R.Uthurusamy, The 1st International Conference on Knowledge Discovery and Data Mining, Montreal, Canada, 87-93.

21. FAYYAD,U.M., IRANI,K.B. 1993.Multi interval discretization of continuous attributes for classification learning. R. Bajcsy, 13th International Joint Conference on Artificial Intelligence, 1022-1027, Morgan Kauffmann, New York.

22. FAYYAD,U.M., PIATETSKY - SHAPIRO., SYMTH,P., 1996. The KDD process for extracting useful knowledge from volumes of data. Communications of ACM, Vol.39, No.11, 27-34.

23. FAYYAD,U.M., PIATETSKY - SHAPIRO., UTHURUSAMY,R., 1996. Advances in Knowledge Discovery and Data Mining. Cambridge, MA: MIT Press.

24. FELDMAN,R., AUMANN,Y., AMÝR,A., ZILBERSTAIN,A., KLOESGEN,W., BEN-YEHUDA,Y., 1997. Maximal association rules: a new tool for mining for keyword co-occurrences in document collection, in Proceedings of the 3rd International Conference on Knowledge Discovery (KDD 1997), 167-170.

25. FELDMAN,R., FRESKO,M., KINAR,Y., LINDELL,Y., LIPHSAT,O., RAJMAN,M., SCHLER,Y., ZAMIR,O., 1998. Text mining at the term level, in Proceedings of the 2nd European Symposium on Knowledge Discovery in Databases, PKDD'98, Nantes, France, 23-26 September 1998, Lecture Notes in Artificial Intelligence 1510: Principles of Data Mining and Knowledge Discovery, Jan M Zytkow Mohamed Quafafou eds., Springer 65-73.

26. GANTI,V., GEHRKE,J., RAMAKRISHNAN,R.,1999. Mining Very Large Databases. In IEEE Computer, Vol.32, Issue.8, 38-45.

27. HAN,E.H., KARYPIS,G., KUMAR,V., 1996. Scalable Parallel Data Mining for Association Rules. Proceedings ACM Conference Management of Data, ACM Pres, New York, 277-288.

28. HAN,J., CAI,Y., CERCONE,N., 1992. Knowledge Discovery in databases: An attribute oriented approach. 18th VLDB Conference, Vancouver, Canada, 547-559.

29. HAN,J., KAMBER,M., 2000., Data Mining: Concepts and Techniques, Morgan Kauffmann Publishers, San Francisco, USA, 2000.

30. HONG,T.P., KUO,C.S., CHI,S.C., 1999. Mining Association Rules From Quantitive Data, Intelligent Data Analysis, 363-376.

31. Introduction to Data Mining and Knowledge Discovery, Two Crows Corporation. "http://www.twocroes.com/intro-dm.pdf", Erişim Tarihi: Mart 2005.

32. JAIN,A.K., DUBES,R.C., 1988. Algorithms for Clustering Data. Prentice Hall, New Jersey.

33. JEA,K.F., CHANG,M.Y., LIN,K.C., 2004. An efficient flexible algorithm for online mining of large item sets. Elsevier, Information Processing Letters 92 (2004), 311-316.

34. LEE,S.K., 1992. An extended relational database model for uncertain and imprecise information. 18th International Conference on Very Large Databases, VLDB'92, Vancouver, Canada, 211-218.

35. LIU,C.L., 1968. Introduction to Combinatorial Mathematics, McGrawHill, New York, 1968.

36. MACQUEEN,J.B., 1967. Some Methods for Classification and Analysis of Multivariate Observations. In L. M. LeCam and J.Neyman, Proceedings 5th Berkeley Symposium on Mathematical Statistic and Probability, 281-297.

37. MENA,J., 1999. Data Mining Your Website, Digital Pres, ISBN: 1-55558-222-2, USA.

38. MENDONCA,M., SUNDERHAFT,N.L., 1999. A State of the Art Report Mining Software Engineering Data: A Survey, DoD Data and Analysis Center for Software (DACS).

39. MILLIGAN,G.W., 1980. An Examination of the Effect of Six Types of Error Perturbation of Fifteen Clustering Algorithms Psychometrika, Vol.45, No.3, 325-342.

40. MULLER,A., 1995. Fast Sequential and Parallel Algorithms for Association Rule Mining: A Comparison, Technical Report CS-TR-3515, University of Maryland, College Park.

41. PARK,J.S., CHEN,M., PHILIP,S.Y., 1995. An Effective Hash Based Algorithm for Mining Association Rules. Proceedings ACM SIGMOD International Conference Management of Data, ACM Pres, New York, 175-186.

42. PARK,J.S., CHEN,M., YU,P.S., 1995. Efficient Parallel Data Mining for Association Rules. Proceedings ACM International Conference Information and Knowledge Management, ACM Pres, New York, 31-36.

43. PIATETSKY – SHAPIRO,G. 1991. Discovery, analysis, and presentation of strong rules. G. Piatetsky – Shapiro and W.J. Frawley, Knowledge Discovery in Databases, Cambridge, MA:AAAI/MIT Press, 229-238.

44. POHLHEIM,H., 1997. Genetic and Evolutionary Algorithms: Principles, Methods and Algorithms. "http://www.systemtechnik.tuilmenau.de/~pohlheim/GA_Toolbox/algindex.htm", Erişim Tarihi: Mart 2005.

45. QUINLAN,J.R., 1986. Induction of decision trees. Machine Learning, cilt.1, 81-106.

46. QUINLAN,J.R., 1986. The effect of noise on concept learning. R. Michalski, J. Carbonell, T. Mitchell, Machine Learning: An Artificial Intelligence Approach, San Mateo, Morgan Kauffman, cilt2, 149-166.

47. RAMASWAMY,S., MAHAJAN,S., SILBERSHATZ,A., 1998. On the Discovery of Interesting Patterns in Association Rules. Proceedings 24th International Conference Very Large Databases, Morgan Kaufmann, San Francisco, 1998, 368-379.

48. SAVASERE,A., OMIECINSKI,E., NAVATHE,S., 1995. An Efficient Algorithm for Mining Association Rules in Large Databases. Proceedings 21st International Conference Very Large Databases, Morgan Kaufmann, San Francisco, 432-444.

49. SENO,M., KARYPIS,G., 2001. LPMiner: An Algorithm for Finding Frequent Item sets Using Length-Decreasing Support Constraint. IEEE Conference on Data Mining, 505-512.

50. SEVER,H., OĞUZ,B., 1999. Veritabanlarında Bilgi Keşfine Formal Bir Yaklaşım. Kısım I: Eşleştirme Sorguları ve Algoritmalar. Bilgi Dünyası, 173-204.

51. SHINTANI,T., KITSUREGAWA,M., 1996. Hash Based Parallel Algorithms for Mining Association Rules. Proceedings 4th International Conference, Parallel and Distributed Information Systems, IEEE CS. Press, Los Alamitos, California, 19-30.

52. SIMOUDIS,E., 1996. Reality Check for Data Mining. IEEE Expert: Intelligent Systems and Their Applications, Vol.11, No.5, 26-33.

53. SRIKANT,R., AGRAWAL,R., 1995. Mining Generalized Association Rules. 21st International Conference on Very Large Databases, VLDB'95, Zurich, Switzerland.

54. SWAMI,A., HOUTSMA,M., 1993. Set oriented mining for association rules in relational databases, Technical Report R.J 9567, IBM Alwaden Research Center, San Jose, California.

55. TOIVONEN,H., 1996. Sampling Large Databases for Association Rules. Proceedings 22nd International Conference Very Large Databases, Morgan Kaufmann, San Francisco, 1996, 134-145.

56. TSAY,Y.J., CHIANG,J.Y., 2004. CBAR: an efficient method for mining association rules, Elsevier, Knowledge-Based Systems 18 (2005), 99-105.

57. WEISS,S.M., KULIKOWSKI,C.A., 1991. Computer Systems that learn: Classification and Prediction Methods from Statistics. Neural Net, Machine Learning, and Expert Systems, Morgan Kauffman

58. YANG,Y., 1999. An evaluation of statistical approaches to text categorization. Information Retrial, 76-78.

59. ZADEH,L.A., 1965. Fuzzy Sets, Information and Control. Vol.8, 338-353.

60. ZAKI,M.J., OGIHARA,M., PARTHASARATHY,S.,LI,W., 1996. Parallel Data Mining for Association Rules on Shared-Memory Multi-Processors. Proceedings Supercomputing, IEEE CS. Press, Los Alamitos, California.

61. ZAKI,M.J., OGIHARA,M., PARTHASARATHY,S.,LI,W., 1997. Parallel Algorithms for Fast Discovery of Association Rules,Data Mining and Knowledge Discovery: An International Journal, Vol.1, No.4, 343-373.

62. ZAKI,M.J., OGIHARA,M., 1998. Theoretical foundations of association rules. In Proceedings of 3rd SIGMOD's Workshop on Research Issues in Data Mining and Knowledge Discovery, Seatle, 85-93.

63. ZAKI,M.J., 1999. Parallel and Distributed Association Mining: A Survey, IEEE Concurrency, special issue on Parallel Mechanisms for Data Mining. Vol.7, No.5, 14-25.

64. ZHANG,T., RAMASWAMY,S., LIVNY,M., 1996. Birch: An Efficient Data Clustering Method for Large Databases. Proceedings ACM SIGMOD International Conference on Management of Data, ACM Pres, New York, 1996, 103-114.

65. ZHANG,T., HSU,M.C, DAYAL,U., 1999. K-Harmonic Means – A Data Clustering Algorithm. Software Technology Laboratory, HPL (Hewlett Packard Laboratory), 1999, 124

66. ZIARKO,W., 1991. The discovery, analysis, and representation of data dependencies in databases. G. Piatetsky – Shapiro and W.J. Frawley, Knowledge Discovery in Databases, Cambridge, MA: AAAI/MIT.

ÖZGEÇMİŞ

1972 yılında Trabzon, Of'ta doğdu. İlk ve orta öğrenimini Sakarya'da tamamladı. 1990 yılında Adapazarı Ali Dilmen Lisesi'nden mezun olarak lise öğrenimini tamamladı. 1998 yılında Marmara Üniversitesi Bilgisayar Mühendisliği Bölümü'nden mezun oldu. 2000 yılında, Kocaeli Üniversitesi Enformatik Bölümü'nde, öğretim görevlisi olarak çalışmaya başlamıştır.

Eylül 2003'te Kocaeli Üniversitesi Fen Bilimleri Enstitüsü Elektronik ve Bilgisayar Eğitimi Anabilim Dalı'nda yüksek lisans öğrenimine başladı.