EGE ÜNİVERSİTESİ FEN BİLİMLERİ ENSTİTÜSÜ

(DOKTORA TEZİ)

KARMA VERİLER ÜZERİNDE ETKİN KÜMELEME ALGORİTMALARININ GELİŞTİRİLMESİ

Elvin NASİBOV

Tez Danışmanı: Doç. Dr. Burak ORDİN

Matematik Anabilim Dalı

Bilim Dalı Kodu: 619.03.03
Sunuş Tarihi: 12.10.2017

Bornova-İZMİR

2017

EGE ÜNİVERSİTESİ FEN BİLİMLERİ ENSTİTÜSÜ

ETİK KURALLARA UYGUNLUK BEYANI

EÜ Lisansüstü Eğitim ve Öğretim Yönetmeliğinin ilgili hükümleri uyarınca Doktora Tezi olarak sunduğum "Karma Veriler Üzerinde Etkin Kümeleme Algoritmalarının Geliştirilmesi" başlıklı bu tezin kendi çalışmam olduğunu, sunduğum tüm sonuç, doküman, bilgi ve belgeleri bizzat ve bu tez çalışması kapsamında elde ettiğimi, bu tez çalışmasıyla elde edilmeyen bütün bilgi ve yorumlara atıf yaptığımı ve bunları kaynaklar listesinde usulüne uygun olarak verdiğimi, tez çalışması ve yazımı sırasında patent ve telif haklarını ihlal edici bir davranışımın olmadığını, bu tezin herhangi bir bölümünü bu üniversite veya diğer bir üniversitede başka bir tez çalışması içinde sunmadığımı, bu tezin planlanmasından yazımına kadar bütün safhalarda bilimsel etik kurallarına uygun olarak davrandığımı ve aksinin ortaya çıkması durumunda her türlü yasal sonucu kabul edeceğimi beyan ederim.

03/11/ 2017

Elvin NASİBOV

Elvin NASİBOV tarafından Doktora tezi olarak sunulan **"KARMA VERİLER ÜZERİNDE ETKİN KÜMELEME ALGORİTMALARININ GELİŞTİRİLMESİ"** başlıklı bu çalışma EÜ Lisansüstü Eğitim ve Öğretim Yönetmeliği ile EÜ Fen Bilimleri Enstitüsü Eğitim ve Öğretim Yönergesi'nin ilgili hükümleri uyarınca tarafımızdan değerlendirilerek savunmaya değer bulunmuş ve 12/10/2017 tarihinde yapılan tez savunma sınavında aday oybirliği/~~oyçokluğu~~ ile başarılı bulunmuştur.

Jüri Üyeleri:		**İmza**
Jüri Başkanı	**: Prof.Dr. Urfat NURİYEV**	.....
Raportör Üye	**: Doç.Dr. Burak ORDİN**	.....
Üye	**: Doç.Dr. Murat BERBERLER**	.....
Üye	**: Yrd.Doç.Dr. Arif GÜRSOY**	.....
Üye	**: Yrd.Doç.Dr. Fidan NURİYEVA**	.....

ÖZET

KARMA VERILER ÜZERİNDE ETKİN KÜMELEME ALGORITMALARININ GELIŞTIRILMESI

NASİBOV, Elvin

Doktora Tezi, Matematik Anabilim Dalı
Tez Danışmanı: Doç. Dr. Burak ORDİN
Ekim 2017, 96 sayfa

Veri madenciliği yöntemlerinden biri olan Kümeleme Analizi, verilerin özelliklerini göz önüne alarak, birbirleri ile benzer olan verileri alt kümelere ayırmayı sağlayan çok boyutlu veri analiz yöntemidir. Kümeleme analizi yöntemleri, kümelenecek verilerin boyutu, ortamı ve özellikle de türüne göre çeşitlilik göstermektedir.

Kümeleme analizinde kullanılan veri setleri, çeşitli yöntemlerle toplanan verilerin özelliklerini içermektedir. Toplanan veriler hesaplanabilir nümerik değerlerle beraber, üzerinde matemetiksel işlemlerin kısıtlı yapılabildiği kategorik özellikler de içermektedir.

Bu tezde, hem nümerik, hem kategorik veriler içeren veri setleri için kümeleme algoritmaları ve onların geliştirilmiş versiyonları incelenmiştir. Literatürde yer alan algoritmaların lokal minimumlarda iyi sonuçlar vermesine karşılık global çözümler için yeterli olmaması nedeniyle, kümeleme analizi probleminin global çözümü için artımlı ve karma veriler ile çalışan yeni bir algoritma önerilmiştir. Önerilen artımlı yöntem C# dilinde MS SQL Server Veri Tabanı Yönetim Sistemi imkanları kullanılarak programlanıp, 16 gerçek veri seti üzerinde hesaplama denemeleri yapılmıştır. Önerilen algoritma k-Prototypes algoritması ile kıyaslandığında yöntemin yararlılığı açıkça gösterilmiştir.

Anahtar kelimeler: Kümeleme, k-Prototypes, Artımlı Kümeleme, Karma Veriler, Uzaklık ve Benzerlik, Matematiksel Programlama, Veri Madenciliği.

ABSTRACT

DEVELOPMENT OF EFFECTIVE CLUSTERING ALGORITHMS ON MIXED DATA

Nasibov Elvin

Supervisor: Assoc. Prof. Burak ORDIN

October 2017, 96 pages

The Clustering Analysis is one of the main techniques of data mining and it is also the method of analysis of multidimensional databases which divides the data set into clusters based on the similarity of data points. Clustering analysis methods vary according to the size, environment and especially the type of data to be aggregated.

Data sets used in the clustering analysis contain the characteristics of data gathering and giving in various ways collected data features include computable numerical values as well as categorical attributes on which mathematical operations can be restricted.

In this thesis, exact clustering algorithms for data sets containing both numerical and categorical data and their improved versions are investigated. Since existing algorithms provide good results at local minimums but are not sufficient for global solutions, a new algorithm for global solution of cluster analysis problem, working with incremental and mixed data, has been proposed. The proposed incremental method is programmed in C# language using MS SQL Server Database Management System facilities and calculation experiments are performed on 16 real data sets. The proposed algorithm clearly shows the usefulness of the method when compared to the k-Prototypes algorithm.

Key words: Clustering, k-Prototypes, Incremental Clustering, Mixed Data, Distance and Similarity, Mathematical Programming, Data Mining.

TEŞEKKÜR

Doktora eğitimim boyunca, bu tezi hazırlamam için benden bilgisini ve anlayışını hiçbir zaman esirgemeyen değerli hocam Doç. Dr. Burak ORDİN'e sonsuz teşekkür ederim. Ayrıca, desteğini ve ilgisini her zaman yanımda duyduğum değerli hocam Prof. Dr. Urfat NURİYEV'e teşekkür etmeği kendime bir borç bilirim.

Hayatım boyunca bana her türlü destek olan, beni her zaman anlayışla karşılayan aileme çok teşekkür ederim.

İÇİNDEKİLER

Sayfa

İÇİNDEKİLER (Devam)

Sayfa

ŞEKİLLER DİZİNİ

Sayfa

ŞEKİLLER DİZİNİ (Devam)

Sayfa

TABLOLAR DİZİNİ

Sayfa

1. GİRİŞ

Küme, aynı sınıfa ait nesnelerin grubudur. Kümeleme ise benzer nesnelerin aynı sınıflara gruplanması sürecidir. Diğer bir deyişle, benzer nesnelerin aynı kümeye ve farklı nesnelerin başka kümelerde toplanmasıdır. Veri madenciliğinde kümeleme, veri analizinde sıkca kullanılmaktadır. Verileri kümeleyerek, onları ilk bakışta farkedilmeyen özellikleri ile alt gruplara ayırıp, onlar üzerinde daha spesifik analizler yapmak mümkün olmaktadır. Elde olan verileri kümelemek için o verilere kümeleme yöntemleri uygulanır. Bu verilerin her öğesine nesne gibi bakılır.

Kümeleme işlemi ile nesne aynı anda sadece bir kümeye ait olabildiği gibi aynı anda birden fazla kümeye de ait olabilir. Nesne tamamen bir kümeye atanırsa buna *kesin kümeleme (hard clustering)* adı verilir. Diğer yandan uygulama sonrası nesne aynı anda birden fazla kümeye farklı aitlik değerleri ile ait olursa buna *esnek kümeleme (soft clustering)* denir. Kümeleme yöntemleri ile veriler üzerinde çeşitli amaçlar için işlemler yapmak mümkündür. Bu yöntemlerle nesneleri belirli özelliklerine göre tek bir kümeye atamak veya farklı kümelere farklı değerlerle ait olmalarını sağlamak, hatta grup ilişkilerine göre hiyerarşik ağaçlar inşa etmek mümkündür.

1.1 Kümeleme Analizi Uygulaması

Kümeleme analizi, zaman içerisinde çok önemli bir teknik haline gelmiştir ve bilim dünyasında çeşitli alanlarda sıkça kullanılmaktadır. Büyük veri setleri bu analiz yöntemi ile işlenmekte ve farklı türden mükemmel sonuçlar üretilmektedir. Özel veriler, alışveriş verileri, bölgesel veriler, ilgi alanları verileri, eylem verileri ve bunlar gibi sonsuz çeşitlilikte veriler ve göstergeler birbirleri ile birleştirilip, analizler gerçekleştirilebilir. Bununla çok önemli veriler elde edilebilir. Artık birçok akım bu analiz yöntemi çeşitlerini sıkca kullanmaktadır. Kümeleme analizi market araştırması, pazarlama stratejileri, model tanımlama, veri analizi, görüntü

işleme ve bunun gibi çeşitli alanlarda etkin kullanılmaktadır [Dempster, 1977; Ben ve Yankini, 1999; Xiaowei, 1998].

Kümeleme analizinin başka tanımı: Özellikler arası benzerlik ya da farklılıklara dayalı olarak hesaplanan ölçülerden yararlanarak verileri homojen gruplara bölmek, belirli prototipler tanımlamaktır.

Kümeleme yöntemleri hiyerarşik ve hiyerarşik olmayan yöntemler olarak iki sınıfa ayrılır. Hiyerarşik kümelemede veri noktaları belirli bölümleme düzeylerinde birleştirilir veya ayrıştırılır. Hiyerarşik olmayan kümeleme yaklaşımında ise, veri noktaları belirli bölümleme kriterlerine göre belirli sayıda kümelere ayrılır.

Kümeleme, pazarlamacıların müşteri bazında farklı gruplar keşfetmelerine yardımcı olmaktadır. Ayrıca bu yöntemlerle pazarlamacılar alım modellerine göre müşteri gruplarını karakterize etmektedirler. Kümeleme, biyoloji alanında bitki ve hayvan taksonomilerini ortaya çıkarmak, benzer işlevleri olan genleri kategorize etmek ve yaşamın doğası ile ilgili fikirler edinmek için kullanılmaktadır.

Kümeleme, aynı zamanda toprak gözlem veritabanlarında benzer arazilerin belirlemesinde kullanılır. Bununla birlikte evlerin tipine ve çeşitli değerlerine, coğrafi konumuna göre onları gruplamaya da yardımcı olabilir. Diğer yandan, kümeleme bilgi keşfi amacıyla web üzerinde belgeleri sınıflandırma veya kredi kartı dolandırıcılığı algılama gibi dışsal saptama uygulamalarında artık sıkça kullanılmaya başlamıştır.

Veri madenciliği fonksiyonu olarak, kümeleme analizi, her kümenin özelliklerini gözlemlemek ve veri dağıtımı hakkında fikir edinmek için bir araç olarak kullanılmaktadır.

Gerçek hayat problemlerine bakıldığında bu problemlerden elde edilen veri setlerini sadece nümerik yada sadece kategorik olarak ele alan algoritmalar olması karma veriler için geliştirilmiş etkin yöntemlerin olmaması literatürdeki önemli bir eksikliktir. Bu nedenle bu çalışmada karma veriler üzerinde etkin çalışan bir algoritmanın tasarlanmasına ihtiyaç duyulmuştur.

Bu tezin amacı karma veriler üzerinde etkili çalışabilen kümeleme algoritması geliştirmektir. Tezde, hiyerarşik olmayan kümeleme yaklaşımına

dayanan, veri setindeki veri grupları arasında kesin ayrımın söz konusu olduğu, kesin artımlı kümeleme yapan ve aynı zamanda hem kategorik, hem nümerik verilerle çalışabilen algoritmalar incelenmiştir.

Sadece nümerik ve sadece kategorik verilerle çalışan algoritmalar, veri setinde özelliklerin hem nümerik hem kategorik veriler olduğu karma verilerle çalışan kümeleme yöntemleri ile beraber ele alınıp, kümeleme probleminin çözümü için karma verilerde daha etkili ve hızlı olabilecek yeni bir algoritma önerilmiştir.

Önerilen yöntem C# dilinde MS SQL Server Veri Tabanı Yönetim Sistemi imkanları kullanılarak programlanıp, 16 gerçek veri seti üzerinde hesaplama denemeleri yapılmıştır. Önerilen algoritma *k-prototypes* algoritması ile kıyaslandığında yöntemin yararlılığı açıkça görülmektedir.

Bu tez, girişi izleyen 16 bölümden oluşmaktadır.

İkinci bölümde; kümeleme modelleri tanıtılıp, kümelemede kullanılacak algoritmanın kümelenecek olan verilere bağımlılığına değinilmiştir.

Üçüncü bölümde; veri türleri incelenmiş ve ölçüm ölçekleri hakkında bilgi verilmiştir.

Dördüncü bölümde; kümeleme analizine tabii tutulacak veri setlerindeki verilerin özellikleri detaylı incelenmiştir.

Beşinci bölümde; mesafeye dayalı kümeleme yöntemleri için sıklıkla kullanılan veriler arasında benzerlik, benzemezlik ve uzaklık kavramları incelenmiştir.

Altıncı bölümde; veri kümeleri ve kümeleme analizinin sınıflandırılması yapılmıştır. Kümeleme yöntemlerinden ileride değinilecek olan bölümlemeli yöntemlere değinilmiştir. Hazır mesafe ve benzerlik ölçekleri ve göreve özel uzaklık ve benzerlik fonksiyonları örneklerle incelenmiştir.

Yedinci bölümde; veri öğrenme algoritmaları ve bu algoritmalar ile veri arasında olan arayüz katmanı hakkında bilgi verilmiştir.

Sekizinci bölümde; karma verilerin kümelenmesi ile ilgili literatürde bulunan çalışmalar incelenmiş ve özetlenmiştir.

Dokuzuncu bölümde; kümeleme problemi tanıtılıp, problemin matematiksel modelleri incelenmiştir. Kesin kümeler, Kategorik modeller ve Artımlı modeller hakkında bilgiler verilmiştir. Sayısal ayrık ve sürekli verilere bakılmıştır.

Onuncu bölümde; sayısal verilerde kesin kümeleme problemleri için klasik ve artımlı çözüm yöntemlerine bakılıp, literatürde yer alan bazı önemli algoritmalar incelenmiştir.

Onbirinci bölümde; kategorik veriler üzerinde kümeleme problemlerine bakılıp literatürde yer alan k-modes algoritması incelenmiştir.

Onikinci bölümde; karma veriler üzerinde kümelemeye değinilmiş ve bununla ilgili literatürde bulunan k-prototypes algoritması incelenmiştir.

Onüçüncü bölümde; karma veriler üzerinde yeni geliştirilmiş artımlı kümeleme algoritması ifade edilmiştir.

Ondördüncü bölümde; UCI Veri Ambarından alınan gerçek veri kümeleri üzerinde önerilen yeni algoritma k-prototypes algoritması ile kıyaslanmış ve yapılan hesaplama denemelerinin sonuçları verilmiştir.

Onbeşinci bölümde; kümeleme yapılması için oluşturulan yazılımın özellikleri ve parametreleri hakkında bilgiler verilmiştir.

Onaltıncı bölüm sonuç bölümüdür.

2. KÜMELEME YÖNTEMİ MODELLERİ

Nesneler üzerinde kümeleme yöntemi uygulamanın çeşitli modelleri vardır. Genellikle verinin özelliklerine göre uygun modelin kullanılması daha iyi sonuçlar verir. Her model için ayrı ayrı algoritmalar mevcuttur. Bu algoritmaların özellikleri ve buna bağlı olarak sonuç değerleri birbirinden farklıdır. Uygulama modelleri amaçlarına, veriyi işlemelerine, yapılarına ve çıktılarına göre birbirinden ayrılmaktalar. En yaygınları aşağıdaki gibi sıralanabilir:

- Merkezi (Centralized) - Her küme bir tek ortalama vektör olarak temsil edilir ve kümelere aitlik bu ortalama vektör değerleri ile karşılaştırılarak oluşturulur.

- Dağıtımlı (Distributed) - Oluşturulan kümeler istatistiksel dağılımları kullanılarak inşa edilir.

- Bağlantılı (Connectivity) – Bu modelde nesneler arasındaki uzaklık dikkate alınır ve aynı kümeye aitlik değerleri zincirleme bağlantılar dikkate alınarak hesaplanır.

- Grup (Group) – Burada algoritmalar yalnızca grup bilgisine sahip olur ve hesaplama buna göre devam eder.

- Yoğunluk (Density) – Yoğunluk tabanlı modellerde küme elemanları, belli yoğunluğun üstünde olanlardan toplanırlar.

Bahsedilen modellerle verileri özelliklerine göre ayırmak için çok sayıda kümeleme yöntemi kullanmak mümkündür. Unutmamak gerekir ki, her yöntemin avantajları olduğu gibi dezavantajları da mevcuttur. Algoritma seçimi kümelenecek olan verilerin özelliklerine ve sonucu nasıl görmek istediğimize bağlı olarak değişir [Berkin, Son erişim tarihi: 4.09.2017].

Burada dikkat edilmesi gereken noktalar bulunmaktadır:

- Kümelemede verilerin oluşturduğu kümeye tek grupmuş gibi davranmak mümkündür.

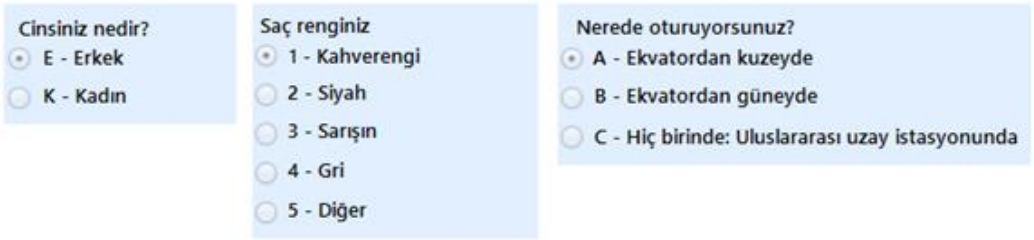
- Kümeleme analizi yaparken, ilk önce verileri benzerliklerine dayalı gruplar halinde kümelenip, sonra gruplara isimleri atamak mümkündür.
- Kümelenmenin, sınıflandırma üzerinde ana avantajı, değişikliklere adapte olması ve farklı grupları ayıran temel özellikleri ortaya çıkarmasıdır.

3. VERİ TÜRLERİ VE ÖLÇÜM ÖLÇEKLERİ

Verilerin çeşitli ölçüm ölçekleri (aynı zamanda veri türü olarak bilinir) vardır: nominal, sıralı, aralık ve oransal. Bunlar, farklı değişken türlerini karakterize etmenin basit yollarıdır. Bu konu genellikle akademik alanda daha fazla ve "gerçek dünyada" daha az sıklıkla tartışılmaktadır. Dört ölçüm ölçeği (nominal, sıra, aralık ve oransal) ve onların alt türlerini aşagıdaki örneklerle anlatalım:

3.1. Nominal (Yazılı) Değişkenler

Nominal değişkenler, niceliksel değer içermeyen değişkenlerin etiketlenmesi için kullanılır. Nominal değişkenlere basitçe "etiketler" denilebilir. Bu ölçekte veriler sıralı olmayıp her birinin aynı önemi olabilmektedir, örneğin göz rengi gibi. Aşağıda bazı örnekler verilmiştir. Bu ölçeklerin hiçbirinin karşılıklı örtüşmediğine ve herhangi bir sayısal değere sahip olmadığına dikkat etmek gerekir. (Şekil 3.1).



Şekil 3.1. Nominal (Yazılı) ölçeğe örnekler.

3.2. İkili (Binary) Değişkenler

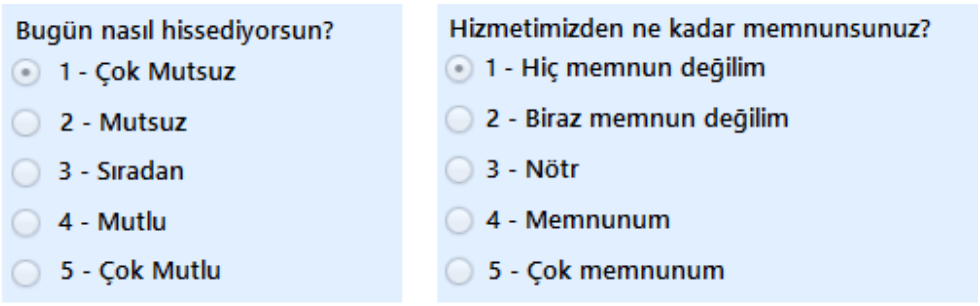
İkili değişkenler iki olasılıktan yalnızca birini tutabilir. Bu olasılıklar örneğin *Doğru* veya *Yanlış*, *Pozitif* veya *Negatif*, *Güncel* veya *Eski*, *Var* veya *Yok* ve benzerleri olabilir. İki olasılığı, *Doğru* için 1 ve *Yanlış* için 0 ile kodlamak

yaygın bir kullanımdır. Ayrıca bir değerin tanımlanan sıfırdan farklı olup olmadığını hesaplamak da mümkündür. Çoğu benzerlik veya benzemezlik yöntemleri kıyaslanan nesnelerin her iki durumu arasındaki birleşimlerin raslantılarının frekanslarını kullanır. İkili değişkenler için yaygın kullanılan metotlara Jaccard indeksi, Hamming uzaklığı örnek olarak söynebilir. İkili değişkenler, kategorik ölçeğin bir alt türüdür ve dichotomous olarak da adlandırılır.

3.3. Sıralı Değişkenler

Sıralı ölçekte, değerlerin sırası önemli ve anlamlıdır, ancak birbirleriyle aralarındaki farklar bilinmemektedir. Sıralı değişkenler genellikle bir seçimin ya da kodlanmış frekansların sırasını temsil eder. Bu ölçek türü, özellik değerlerinin, örneğin anketlerdeki sorular gibi, en az önemliden en çok önemliye sıralanması gibi net hesaplanamayan değerleri göstermek için kullanılmaktadır. Şekil 3.2'de bulunan örneklere göz atmak gerekirse, her iki örnekte de 4 numaralı seçeneğin 3 veya 2 numaralı seçenekten daha iyi olduğunu biliyoruz, ama *ne kadar* iyi olduğunu ölçemediğimiz için bilmiyoruz. Örneğin "Sıradan" seçeneği ile "Mutsuz" seçeneği arasındaki fark "Çok mutlu" ile "Mutlu" arasındaki kadardır gibi bir sonuç çıkaramayız.

Sıralı ölçekler genellikle memnuniyet, mutluluk, rahatsızlık vb. gibi sayısal olmayan kavramların ölçümünde kullanılmaktadır (Şekil 3.2).



Şekil 3.2. Sıralı ölçeğe örnekler.

Ayrıca, sıralı veriler, sayısal ve kategorik verilerin karışımından oluşabilmektedirler. Bu durumda veriler kategorilere girer, ancak kategorilere yerleştirilen değerlerin sıralı anlamı vardır. Örneğin, bir restoran derecelendirmesi için verilen puan 0'dan (en düşük) 4'e (en yüksek) aralıkta, sıralı verilerden oluşabilir. Sıralı veriler genelde kategorik olarak ele alınır, burada grafikler ve çizelgeler hazırlanarak gruplar sıralanır. Bununla birlikte, kategorik verilerin aksine, sayıların matematiksel anlamı vardır. Örneğin, 100 kişi arasında soruşturma yaparak restoran için 0-4 arası bir puan vermelerini istersek, 100 yanıtın ortalamasının alınması anlam kazanacaktır. Kategorik verilerde böyle bir durum söz konusu değildir.

Önemli not: Sıralı veri üzerinde *merkezi eğilimi* belirlemenin en iyi yolu, mod veya medyan kullanmaktır. Uzaklık kavramı olmadığı için sıralı veride ortalama bulunamaz.

3.4. Aralık Değişkenler

Aralık ölçeği, sadece sıralamayı değil aynı zamanda değerler arasındaki kesin farkları bildiğimiz sayısal ölçektir. Bu ölçek zaman veya sıcaklık gibi verilmiş bir aralığı temsil etmektedir. Aralık ölçeğinin (skalasının) klasik örneği *Celsius (selsiyus) sıcaklığıdır*, çünkü her değer arasındaki fark aynıdır. Örneğin, 60 ve 50 derece arasındaki fark, 10 derece olarak ölçülebilir ve 80 ile 70 derece arasındaki farka eşittir. Başka iyi bir örnek, aralıkları tutarlı ve ölçülebilir olduğu için, *Zaman* değerlerini söyleyebiliriz. Aralıklı ölçekler veri setleri üzerinde istatistiksel analiz alanı açtığından veri toplama aşamasında tercih edilmektedirler. Örneğin, bu verilerde mod, medyan ile *merkezi eğilim* veya ortalama ölçülebilir, standart sapma hesaplanabilir.

Aralık ölçekleri ile ilgili bir özel bir durum vadır. Bu ölçeklerde "mutlak sıfır" yani “hiçbir” değerinin olmaması durumu yoktur. Örneğin, sıfır derece var ama "hiç sıcaklık yoktur" anlamına gelmiyor. Gerçek bir sıfır olmadan, oranları hesaplamak imkansızdır. Çünkü, örneğin -10 derece sıcaklıktan 0 derece sıcaklığa geçildiğinde sıcaklık kaç oranında arttı kavramı yanlış olur. Diğer yandan, aralık

verileri ekleyebilir ve çıkartabiliriz, ancak çarpma veya bölme yapamayız. Mesela 10 derece + 10 derece = 20 derece eder ama 20 derece 10 dereceden iki defa sıcaktır söyleyemeyiz. Ayrıca, selsiyus ölçeğinde "sıcaklık yok" değeri bulunmamaktadır. Aralık ölçekleri çok kullanışlıdırlar, fakat oran hesaplanması yapılamadığı için amaca göre oransal ölçeğe yönlenebilinir (Şekil 3.3).

Şekil 3.3. Aralık ölçeğe örnek.

3.5. Oransal

Ölçüm ölçekleri söz konusu olduğunda oransal ölçekler en tercih edilenlerdir diyebiliriz. Çünkü bu ölçekler bize değerlerin sırasını ve her bir değer arasındaki farkı söyleyebilir. Ayrıca bu ölçeğin mutlak sıfır değeri vardır ve bununla bu veriler üzerinde tanımlayıcı ve çıkarımsal istatistikler uygulanabilinir. Yukarıda bahsedilen diğer veri ölçeklerinin tüm özellikleri ve buna ilave olarak sıfır kavramını oransal ölçeklere ait edebiliriz. Bu ölçeğe iyi örnek olarak uzunluk ve genişliği verebiliriz.

Oransal ölçekler, istatistiksel analiz söz konusu olduğunda çok sayıda olanaklar sağlamaktadır. Bu değişkenler üzerinde anlamını kaybetmeden ekleme, çıkarma, çarpma, bölme işlemleri yapılabilir. Verilerin merkez eğilimi mod, medyan veya ortalama ile ölçülebilir; Standart sapma ve değişim katsayısı gibi dağılım ölçütleri de oran ölçeklerinden hesaplanabilir.

Yüzdeler veya oranlar, iki bilgi parçasını, payı ve payda değerleri ile özetler. Basit oranlar (0 ve 1, yani pay ve paydanın alabileceği maksimum mümkün olan değerdir) sürekli veri olarak ele alınabilir. Oran bir değişimi temsil ettiğinde ve

değerler negatif olabildiğinde verilerin analiz edilmesi daha zor olabiliyor. Gözlem oranlarının (elde edilen değerlerin) referans değerlerle karşılaştırılması zamanı, örneğin boy yüksekliği için belirli bir cinsiyet ve yaş popülasyonun ortalaması referans olarak baz alınıyorsa değerler bu referans değerinin (%100) her iki tarafında yani hem bundan hem az hem çok olabileceğinden işlenmesi zordur.

Birçok istatistiksel yöntem sadece bazı ölçüm ölçekleri için uygundur. İstatistiksel bir yöntem seçerken, analiz edilecek verilerin nasıl ölçüldüğünü anlamak önemlidir. Ölçüm ölçeklerini belirlemek için en iyi aşama tasarım aşamasıdır. Çünkü bazı ölçüm ölçekleri tarafından etkilenen istatistiksel sınırlamalar bu aşamada veri için gözlem ve ölçüm yöntemlerini nasıl seçeceğimizi etkileyebilir. Ayrıca veriler dönüştürüldüğünde ölçüm ölçeklerinin de değiştirilmesi gereken durumlar olmaktadır (Şekil 3.4).

Şekil 3.4. Bu cihaz iki oransal ölçek (yükseklik ve ağırlık) örneği sağlar.

3.6. Özet

Özet olarak, yazılı değişkenler, bir dizi değeri "isimlendirmek" veya etiketlemek için kullanılır. Sıralı ölçekler, müşteri memnuniyeti anketinde olduğu gibi, tercih sırası hakkında bir bilgi sağlar. Aralık ölçekleri bize değerler sırasını ve ilaveten her biri arasındaki farkı belirleme becerisi kazandırır. Son olarak, oran ölçekleri nihai sıra, aralık değerlerini ve ayrıca veriler üzerinde her türlü hesaplama yapma olanağı sunar.

İstatistiksel analizde, farklı ölçüm ölçeklerini bilmek önemlidir. Ölçüm çizelgelerinin sınıflandırılmasına ek olarak veri türleri konusunu da bilmek başlıca öneme sahiptir. Bir metodu bir veri setine uygulamadan önce bazı ön kontroller yapılmalıdır. Stevens'in 'Ölçüm Ölçekleri Teorisi' [Stevens, 1946] hala tartışmalı ve genellikle yetersiz olsa da [örneğin, Velleman & Wilkinson, 1993], veri tipi, uygun metodu seçmek için önemli bir kıstastır. Veri tipleri ve bunlar üzerinde yapılabilecek işlemler Şekil 3.5'te verilmiştir.

Özellikler:	Nominal	Sıralı	Aralık	Oransal
Değerlerin sırası bilinmektedir.		✔	✔	✔
Sayısı ve Dağılma sıklığı bilinmekte.	✔	✔	✔	✔
Mod	✔	✔	✔	✔
Median		✔	✔	✔
Mean (Ortalama)			✔	✔
Tüm değerler arasındaki farklar belirlenebilinmekte			✔	✔
Değerleri toplama ve çıkarma yapmak mümkün			✔	✔
Değerleri çarpmak ve bölmek mümkün				✔
Mutlak sıfır değeri içerir				✔

Şekil 3.5. Veri tiplerinin ve ölçeklerin özeti.

4. VERİ ÖZELLİKLERİ

Veri tipinin göz önünde bulundurulmasına ek olarak bazı ilk varsayımları ve koşulları da göz önüne almak gereklidir.

4.1. Karışık Değişkenler

Aynı boyutta farklı tipte değişkenlerin kullanılması tartışmalı bir durum olsa da veri toplamada sıkça karşılaşılmaktadır. Çünkü gerçek hayatta toplanan veriler farklı araçlar ve yöntemlerle toplanmaktadır. Bu sorunla baş etmek amacıyla birçok çalışma yapılmıştır. Değişkenlere bağlı olarak her bir değişken için farklı bir ağırlık teriminin kullanılması uygulanabilir [Gower, 1971, Gower, 1987]. Hiçbir genelleştirme verilemeyeceğinden, güvenilir ve güçlü sonuçlar sunan yöntemi bulmak genellikle araştırmacıya kalır.

Yine de bir analizde farklı birimleri kullanırken sonuçları dikkatlice yorumlamak her zaman tavsiye edilir. Bu tür karmaşık verileri olabildiğince azaltmak daha uygun olur.

4.2. Ortak Sıfırlar

Mesafe, benzemezlik ve benzerlik ölçümlerinin birçoğu ortak sıfırlara duyarlıdır. Bu, kıyaslanan nesneler arasında aynı değişkenlerde sıfırları olan veri kayıtları problemine işaret eder. Örneğin, organizma türlerinin analizinde birkaç veri toplama istasyonunda birçok türün değeri bilinse de, tek bir türün yalnızca birkaç bireyinin bulunduğu görülebilir. Çoğu durumda araştırmacı, verimli türler arasındaki farklılıkların analizi ile ilgilenir. Algoritmaya bağlı olarak böyle yaygın veri yoklukları analizde çok daha yüksek etki yaratabilir. Ortak sıfırlara duyarlı ölçümler sıklıkla ve genel olarak uygulanabilir olması için yeterince güçlü değildir [Field, 1982]. Bu durumda, *logaritmik* veya *karekök* gibi dönüşümler bile

yardımcı olmamaktadır. Örneğin, Minkowski ölçeği gibi yöntemler ortak sıfırın yüksek oranlarına duyarlı olabilir.

4.3. Negatif Değerler

Bazı mesafe, benzemezlik ya da benzerlik ölçümlerinin sonuçları, yalnızca değişkenler tamamen pozitifken tanımlıdır. Geliştirilecek yöntemlerde öncelikle oluşacak verinin negatif değerler alıp almaması netleşmelidir. Ayrıca kümeleme yapmadan önce varolan veri setini de kontrol etmek gerekir. Negatif değerler alma olasılığı varsa veya zaten negatif değerlere sahip bir veri boyutu ile çalışılacak ise geliştirilecek yöntemin bu durumu göze almasına dikkat etmek gereklidir. Birçok yaygın yöntemler, örneğin, çevrebilimde yaygınca kullanılan Bray-Curtis benzemezliği negatif değerler işlendiğinde güvenilir sonuçlar üretmez. Bu ayrıca Canberra metriği için de geçerlidir (verilerin tamamı negatif olmadıkça) [Bray, Curtis, 1957].

4.4. Eksik Değerler

Uzaklık, benzemezlik ya da benzerlik ölçümleri tam veri setleri için kurulur. Çoğu analitik yazılımlar eksik veriye izin vermez ve analizi sonlandırır, ancak sonuçlar bir ya da daha çok değer eksikken hala anlamlı olabilir. Bu sebepten, böyle veri setlerini önceden işlemek tavsiye edilmektedir. Bu sebepten ötürü, bu tür veri setlerini ön işlemden geçirmek tavsiye edilmektedir. Daha detaylı analizlerde eksik nesneleri eğer çok rastlanırsa setten çıkarmak genellikle daha çok tercih edilir. Başka bir seçenek ise elde olan verilerden ve yorumlardan eksik değerleri tahmin etmek olabilir. Ama eksik veri fazlalaştıkca bu sorun artar, analiz sonuçları sağlıklı olmamış olur.

4.5. Dönüşüm

Verileri işlemeden önce ayrıca normalleştirme de yapmak gerekmektedir. Ayrıca, aynı özellik değeri için veriler farklı kaynaklardan veya farklı yorumlarla oluştuğu durumlarda verileri aynı hizaya getirmeden incelemek anlamsız olur. Bunun için çok sayıda yöntem geliştirilmiştir. Örneğin, *Log* ya da *kök* gibi dönüşümler çarpık veriyi olumlu olarak normalleştirir ve benzemezlik indislerine dayanan prosedürlerde yüksek değerlerin etkisini azaltır [Digby ve Kempton, 1987]. Özellikle uzaklık fonksiyonlarını farklı ölçekler ya da geniş aralıklardaki değişkenlerde kullanırken analizleri saptırabilir. Bu durumda veri dönüşümlerini veya standartizasyonlarını kullanmak önerilebilir. Bazı yazarlar dominant değişkenlerin etkisini azaltmak için *kökten-köke* dönüşümünü kullanmayı önerir [Clarke ve Warwick, 1994; Field v.d.., 1982]. Bu yaklaşım, dönüşümün etkisinin, değişkenin altında yatan dağılıma bağlı olmasından dolayı zordur ve tutarsız olabilir [Quinn ve Keough, 2004]. Bu yüzden Quinn ve Keough [2004] veri standartizasyonunu tercih etmektedirler. Bu nedenle farklı aralıklardaki değişkenlerin etkisini azaltmak için veriyi ikili varlık-yokluk durumlarına indirgemek yararlı olabilir.

5. VERİ KAYITLARI ARASINDA BENZERLİK, BENZEMEZLİK VE UZAKLIKLAR

5.1. Amaç

Belli bir nesne için karakteristik özellikler tanımlarken araştırmacılar sık sık sayısal bulgularla yüzleşirler. Birçok uygulama için iki ya da daha fazla nesnenin ne kadar benzer olduğunu bulmak büyük önem arzetmektedir. *n*, nesnenin ölçülen değişkenlerinin sayısı olmak üzere, böyle bir uzayda konumu, *n*-boyutlu bir vektör ile tanımlanır. Genelleşmiş olarak bu, her nesnenin, değişken sayısıyla aynı boyuta sahip bir uzayda bir nokta olduğu anlamına gelir. Değişkenleri tek tek incelemek aralarındaki ilişkileri ihmal eder ve çok değişkenli karakteristiklerine uygun olmaz. Birden fazla nesneye sahipken uzaklık ve benzerlik indeksleri tüm değişkenleri içererek hesaplanabilir. Elde edilen değerler, nesneler arasındaki ilgili boşlukları tek bir değerle ifade eder ve benzerlikleri için bir ölçüdür. Daha geniş veri setlerinde doğal olarak oluşan iki ya da daha fazla grupta, analize dahil edilen değişkenler böyle bir uzayda nesneleri ayırıyorsa, o zaman ayırt edilebilir. Bunlar, kümelerin ve nesne gruplarının benzer ya da birbirinden ayrı olmasını tanımlayan veri sınıflandırma tekniklerinin temelidir.

Bir kısıtlama olarak, indisler, hesaplamada boş değer kabul etmezler ve her değişken için değer gerektirir. İndisler; bilimsel, endüstriyel, mühendislik ve diğer uygulamalarda yaygınca kullanılır [Legendre & Legendre, 1998]. Çevrebilimle ilgili araştırmalar için başlıca kullanılan 50'den fazla farklı ölçüm vardır.

5.2. Farklı Türden Verilerde Basit Nitelikler için Benzerlik/Benzeşmezlik.

Farklı türden verilerde basit nitelikler için benzerlik/benzeşmezlik tanımlamaları Tablo 5.1'de verilmiştir. Burada p ve q iki veri nesnesi için nitelik değerleridir.

Tablo 5.1 Nitelik türlerine göre verilerde benzerlik ve benzeşmezlik.

Nitelik Türü	**Benzerlik**	**Benzeşmezlik**
Nominal	$s = \begin{cases} 1 & eğer\ p = q \\ 0 & eğer\ p \neq q \end{cases}$	$d = \begin{cases} 0 & eğer\ p = q \\ 1 & eğer\ p \neq q \end{cases}$
Sıralı	$s = 1 - \frac{\|p - q\|}{n - 1}$ (veriler 0 tam sayısından, n-1'e kadar haritalanmıştır, n değerlerin sayısıdır)	$d = \frac{\|p - q\|}{n - 1}$
Aralık veya Oransal	$s = 1 - \|p - q\|, s = \frac{1}{1 + \|p - q\|}$	$d = \|p - q\|$

Budara uzaklık, bir benzemezlik ölçeğidir (örneğin Öklit uzaklığı) ve bazı iyi bilinen özellikleri vardır:

1. her p ve q *için* $d(p, q) \geq 0$, ve $d(p, q) = 0$ eğer sadece $p = q$ *ise*,
2. her p ve q *için* $d(p, q) = d(q,p)$,
3. tüm p, q ve r için $d(p, r) \leq d(p, q) + d(q, r)$, burada $d(p, q)$, p ve q noktaları (veri nesneleri) arasındaki uzaklıktır (benzemezliktir).

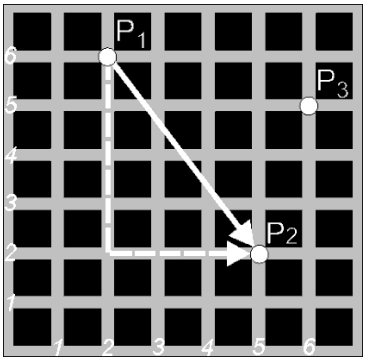
Bu özellikleri sağlayan herhangi bir fonksiyona **metrik** denir. Aşağıda bahsi geçecek olan uzaklık çeşitleri, çok değişkenli veriyi karşılaştırmak için yaygınca

kullanılan birkaç uzaklık ölçütünün bir listesidir. Bütün niteliklerin sürekli olduğunu kabul edeceğiz.

5.3. Uzaklık Çeşitleri

Nümerik olarak ifade edilen nesnelerin benzerliği için ilk örnek olarak, bir düzlemdeki noktalar düşünülebilir. Bu noktalar, örneğin, bir şehir haritasındaki koordinatlar olabilir. Her koordinatın elemanları değişkenlerdir. İlk değişken verilen bir x-ekseni boyunca, ikincisi de dikey bir y-ekseni boyunca pozisyonu tanımlar. Böylece, bu düzlemdeki bir nokta iki boyutlu bir uzayda bir nesnedir.

İki nokta arasındaki ilişkiyi ifade etmek için aralarındaki en kısa uzaklığın, Öklid uzaklığı olduğunu söyleyebiliriz. Öklid uzaklığı, bu iki noktanın bir dikdörtgenin zıt köşeleri iken köşegenini belirlemekle eşdeğerdir. Buna karşılık, verilen probleme bağlı olarak bu uzaklık yaklaşımı en iyi ölçümü verir demek yanlış olur. Başka bir seçenek iki eksen üzerindeki mutlak uzaklıkları toplamaktır. Bu, bir kent yerleşiminde olduğu gibi sokaklar dikey açıdayken bir kişinin yürümesi gereken karayoluyla kıyaslanabilir. Bu uzaklık, sokaklar boyunca iki blok arasındaki en kısa uzaklığı verir ve Manhattan uzaklığı olarak anılır. Bu durumda, bloklar doğrudan geçilemeyeceğinden, Öklid uzaklığının kullanılması uygun olmayacaktır, ama bir helikopter kullanılmasıyla Öklid uzaklığını kullanmak yararlı olabilir (Şekil 5.1).



Şekil 5.1. İki-boyutlu uzaydaki noktalar arasında Öklid uzaklığının (kalın ok) ve Manhattan uzaklığının (kesikli ok) örneği.

Yukarıdaki resimdeki noktaların konumları bir tabloda gösterilebilir. Satırın uygun sütununda değişkenler verilirken, her nokta yeni bir satırda listelenmiştir.

Bu, çeşitli nesneler için bulguları görüntülemenin yaygın bir yoludur. Yukarıdaki örnek için çizelgelenen x ve y koordinatları düzlemdeki konumu veren iki boyutlu satır vektörlerinin değerleridir (Tablo 5.2):

Tablo 5.2 Örnek veriler.

Nesne	*x*	*y*
P_1	2	6
P_2	5	2
P_3	6	5

Öklid uzaklığını hesaplamak için iki eksen boyunca farkların karelerinin toplamının kökü alınmalıdır. Böylece P_1 ve P_2 arasındaki uzaklık:

$$d^{EUD}(P_1, P_2) = \sqrt{(2-5)^2 + (6-2)^2} = \sqrt{(-3)^2 + (+4)^2} = \sqrt{25} = 5\text{'dir.}$$

Manhattan uzaklığı, eksenler boyunca koordinatların mutlak farklarının toplamı olarak tanımlanır. Bu farklılıklar, ikinci uzaklık ölçümüyle aynı olduğu için, bunları toplarız ve Manhattan uzaklığı için sonuç:

$$d^{MAD}(P_1, P_2) = |(2-5) + (6-2)| = 3 + 4 = 7\text{'dir.}$$

Her iki durumda da sonucun birimi geçilen blokların sayısıdır.

Öklid Uzaklığı

x_{ik}, $i = 1, \ldots, N$, değişkenleri üzerinde $k = 1, \ldots, p$, ölçümü (nitelikler de denir) yaptığımızı varsayalım.

Ölçülen her (i, j) çifti için i'nci ve j'nci nesne arasındaki Öklid uzaklığı

$$d_E(i,j) = \left(\sum_{k=1}^{p} (x_{ik} - x_{jk})^2 \right)^{\frac{1}{2}} \qquad (5.3.1)$$

Ağırlıklı Öklid mesafesi aşağıdaki gibi olacaktır:

$$d_{WE}(i,j) = \left(\sum_{k=1}^{p} W_k (x_{ik} - x_{jk})^2 \right)^{\frac{1}{2}} \quad (5.3.2)$$

Burada w_k , $k = 1, \ldots, p$, özelliklerin (özniteliklerin) ağırlığıdır. Özniteliklerin ölçekleri önemli ölçüde farklıysa, normalizasyon gereklidir.

Minkowski Uzaklığı

Minkowski uzaklığı, Öklid uzaklığının genellemesidir. x_{ik} , $i = 1, \ldots, N$, $k = 1, \ldots, p$, ölçümü için Minkowski uzaklığı:

$$d_M(i,j) = \left(\sum_{k=1}^{p} |x_{ik} - x_{jk}|^{\lambda} \right)^{\frac{1}{\lambda}}, \quad (5.3.3)$$

olacaktır. Burada $\lambda \geq 1$, ve bu uzaklık L_λ metriği olarak da adlanır.

- $\lambda = 1$: L_1 metriği, Manhattan veya City-block mesafesine dönüşür,
- $\lambda = 2$: L_2 metriği, Euclidean mesafesine dfönüşür,
- $\lambda \to \infty$: L_∞ metriği, Supremum mesafesine dönüşür, yani

$$\lim \lambda \to \infty = \left(\sum_{k=1}^{p} |x_{ik} - x_{jk}|^{\lambda} \right)^{\frac{1}{\lambda}} = \max \left(|x_{i1} - x_{j1}|, \ldots, |x_{ip} - x_{jp}| \right) \quad (5.3.4)$$

Mahalanobis Uzaklığı

X 'ın N × p matrisi olduğunu varsayalım. O halde, **X** 'ın i'nci satırı

$$x_i^T = (x_{i1}, \ldots, x_{ip}) \quad (5.3.5)$$

olur. Bununla Mahalanobis Uzaklığı

$$d_{MH}(i,j) = \left((x_i - x_j)^T \Sigma^{-1} (x_i - x_j)\right)^{\frac{1}{2}} \quad (5.3.6)$$

şeklinde gösterilir. Burada ∑ , p×p boyunda olan kovaryans matristir.

Uzaklık ölçüsü, herhangi *i.* ve *j.* nesneler arasındaki bir benzemezlik ölçüsü olarak ele alınabilir. Belli $d(i,j)$ uzaklık ölçüsü, çeşitli dönüşümlerle $s(i,j)$ benzerlik ölçüsüne de dönüştürülebilir:

$$s(i,j) = \max\{0,\ 1 - d(i,j)\} \quad (5.3.7)$$

veya

$$s(i,j) = \frac{1}{1+d(i,j)} \quad (5.3.8)$$

5.4. Uygulama Örneği

Mesafenin, benzeşmezliğin ya da benzerliğin hesaplanması için bir hayli algoritma mevcuttur. Her birinin kendi yetenekleri ve sınırları vardır. Bütün bu yöntemlerde ortak olan, nesneler arasındaki ilişkileri tek bir değer ile temsil etme yeteneğidir.

Bu tür hesaplamalar, birçok nesne için ilk tahminlerde ilgi çekici olabilir. Bir veri setinin bütün nesneleri arasında yapıldığı zaman ise, benzerliklerin, benzeşmezliklerin ve mesafelerin hesaplanması daha da anlamlı olur. Sonuçlar nesneler arasındaki farklı mesafelerin değerleri sayısına eşit sıra ve satırları içeren bir matriste özetlenebilir. Bu, sonuç tablosu ya da matrisinin kare olması anlamına gelir. Her nesne bir kez sırasıyla hem satır, hem sütunda olmuş olacaktır. (i,j) hücresinde, i nesnesi ile j nesnesi arasındaki mesafe değeri olur. i nesnesi j nesnesi ile ve tam tersi, j nesnesi i nesnesi ile kıyaslandığında aralarında fark olmadığından, yani aynı olduğundan matris simetrik ve köşegeninden ayna yansımalı olur. Bu tasarım, bilgi kitapçıklarında bulunan, örneğin şehirler arasındaki uzaklığı veren mesafe tabloları ile aynıdır (Tablo 5.3).

Tablo 5.3 Uzaklık matrisi örneği.

Öklid Uzaklığı	Durum*1*	Durum*2*	Durum*3*	Durum*4*	Durum*5*	Durum*6*
Durum1	0	1	1.732	15.588	17.321	9.899
Durum*2*	1	0	2.449	16.186	17.916	9.849
Durum*3*	1.732	2.449	0	13.856	15.588	8.775
Durum*4*	15.588	16.186	13.856	0	1.732	11.180
Durum*5*	17.321	17.916	15.588	1.732	0	12.570
Durum*6*	9.899	9.849	8.775	11.180	12.570	0

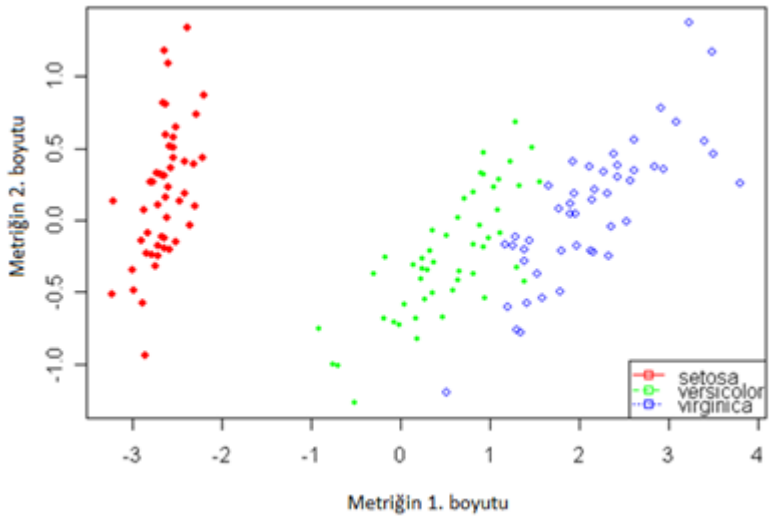
Köşegen boyunca Öklid mesafe değerlerinin daima sıfır olduğu görülebilir. Bu, mesafe matrisleri için ortak bir durumdur ve bir nesne kendisi ile karşılaştırıldığında tam eşleşildiği anlamına gelir. 0'a daha yakın değerlerde bu nesnelerin daha çok benzer oldukları anlamına gelir. Benzerlik matrisinde ise, matris kendisi simetrik olsa da, köşegen boyunca 1 değeri bulunabilir.

Matris simetrik olduğunda, köşegenin alt veya üst yarısı genellikle yok sayılır. Matrisin köşegen değerleri de eğer bir köşesinde bilgi olarak verilmiş ise tamamı gösterilmeyebilir. Bunun gibi üçgen matrisler, farklı veri işleme teknikleri için bir ön koşuldur. İsteğe bağlı üstteki veya alttaki bölüme geçmekle üçgen matrislere çevrilen bu yöntemler, bütün matrisin yaratımı için faydalı olmaktadırlar. Böylece yalnızca bir tarafı hesaplanarak, sonuçlar diğer tarafa ayna gibi yansıtılabilir.

Veri işleme teknikleri, bir n-boyutlu veri setinin özelliklerini daha düşük boyutlarda ele almayı amaçlar [Gauch, 1982]. Rastgele bir başlangıçta, yapılandırılan nesneler, ölçülen ilişkilere göre, hedef olacak boyutta düzenlenir. Bu, veri nesnelerini temsil eden küreler modeli olarak düşünülebilir. Her küre tüm diğerleri ile yaylarla birbirine bağlı olur. Yayların nötr uzunlukları, ilgili objeler arasındaki ilişkiyi gösterir. Rastgele başlangıca göre yaylar gerilmişler ve kendi nötr uzunluklarında değildirler. Tüm yerinden oynamaların sonucu olarak global gerilim, genellikle stres olarak tanımlanır. İşleme zamanı, yaylar boyunca toplam stresin en düşük olduğu, hedef alanın boyutunda bir düzen bulunmaya çalışılır. Bu tekrarlanan süreç boyunca, nesneler arasında göreceli bir mesafe oluşturmaya ve adım adım stresin düşürülmesine çalışılır. Mükemmel eşleşmeler sayılmazsa kapsamlı stres nadiren sıfır olur. Özellikle daha büyük veri setlerinde, bütün nesneler arasındaki göreceli mesafeler güçlükle bağdaştırılır. Bununla beraber,

daha yüksek boyutlu ve kompleks verilerin insan aklıyla algılanabilen 1, 2 veya 3 boyutlu bir alanda gösterilmesine izin vermiş olur.

Analiz boyunca verilen değerlerle ayırt edilebilir nesneler, kümeler oluşturmaya doğru bir eğilim gösterir. Grafik çizimleriyle bu tür veri setleri genellikle görsel olarak da ayırt edilebilir. Böylece, nesneleri, onların mesafesi, benzeşmezliği ya da benzerliğine göre bir gruba ayırmak ve daha büyük veri setlerindeki daha yüksek sıralı yapıları ifade etmek mümkündür. Bu, karmaşık ilişkileri anlamaya ve gruplar arasındaki benzerlikleri ifade etmeye yardımcı olabilir. Yeni bir nesnenin hangi bilinen gruba ait olduğu sayısal olarak grupların ağırlık merkezlerine uzaklıkları ile belirlenebilir. Yine de, eğim üzerinde nesnelerin kıyaslanmasının başarısız olma olasılığı vardır, çünkü nesneler arasındaki karşılıklı-ikili testler eğimin farklı kenarlarında farklı sonuçlar verebilir [Somerfield, 2002].



Şekil 5.2. Fisher-Anderson'un IRIS veri setinin Çok Boyutlu Ölçüm Metriği

Yukarıdaki şekil (Şekil 5.2), Fisher tarafından onun sayısal sınıflandırma bilgileri üzerine yazdığı makalesindeki, ünlü Iris veri setinin (Iridaceae bitki familyası) [Anderson, 1935] Çok Boyutlu Ölçüm (Multi-Dimensional-Scaling/MDS) sınıflandırmasını gösterir [Fisher, 1936]. Çok Boyutlu Ölçüm (MDS) ya da Sammon haritalaması yaygın karar alma yöntemlerindendir.

Şekildeki veri seti, *Iris setosa, Iris versicolor ve Iris virginica* olarak adlandırılan bitki türünün her üçü için 50, toplamda 150 örneğe sahiptir.

Her örnek için 4 değişken ölçülmüştür. Bu değişkenler her öğenin çanak yaprakları ve taç yapraklarının uzunluk ve genişlik değerleridir. Yukarıdaki MDS için, ayrı ayrı nesneler arasındaki ilişki ölçüsü olarak Öklid uzunluğu seçilmiştir. Öğelerin iki geniş gruba ayrıldığı görülmektedir. İlk grup Iris Setosa türünden oluşurken ikinci grup Iris Vesicolor ve Iris Virginica türlerinden oluşturulmuştur. Iris Setosa'nın Iris Versicolor ve Iris Virginica'dan ayrıldığı görülebilir. Bu, Iris Setosa türünün değişkenler aracılığıyla diğer iki türden ayırt edilebilmesinin mümkün olduğu anlamına gelir. Ayrıca Iris Versicolor ve Iris Virginica'nın türleri aralarındaki mesafe kısa olsa da, ölçülen parametrelerin boyutunda bir takım karakteristik farklılıklar olduğu görülmektedir. Türler arasındaki bu farkı da analizlerle hesaplamak mümkündür. Grafik üzerindeki noktalar, kendi gruplarına göre renklendirilmiştir. Bu analiz, tek başına değişkenler hakkında bir şey söylememekle beraber, veri setini bütün olarak inceleme imkânı sağlamaktadır.

6. MESAFEYE BAĞLI VERİ KÜMELEME

6.1. Veri Kümesindeki Nesneler Arası Mesafe

Dijital olarak kaydedilen veriler mevcut araştırma ortamımızda önemli bir doğal kaynak haline gelmiştir. Bu gerçeklik onlarca yıllık bilgisayar mühendisliği, bilgisayar bilimi, elektronik ve iletişim araştırmalarının muazzam zaferlerinden biridir. Bu senaryo gelecekte de geçerli olmaya devam ederken, çağımız dijital olarak depolanan veriyle yakından ilintili durdurulamayan bir başka faaliyetle, bu tür verilerden bilgi ve enformasyon çıkarmanın başlangıcını da belirlemiştir. Dijital veriler farklı şekillerde ve çeşitli ölçeklerde kaydedilir. Örnekler, her türden veritabanı tablolarını içerir; Tweetler; E-postalar ve metin belgeleri; ses ve konuşma sinyalleri; sismik veriler (zamansal çok boyutlu tensörler olarak kaydedilir); video verileri (büyük oranla ses ve altyazı gibi diğer yöntemlerle depolanır); farklı varlıklar arasındaki ilişkileri ve etkileşimleri temsil eden grafikler (Web dokümanları arasındaki bağlantılar, sosyal ağlardaki kullanıcılar, bilgisayar ağlarındaki cihazlar veya gen etkileşim ağları); ve fonksiyonel manyetik rezonans görüntüleme (fMRI) gibi resimleri ve daha fazlasını içeren görüntüleri. Farklı biçim ve yapılarından dolayı, burada, bir veri kümesindeki her veriye *nesne* denilecektir.

Büyüklükleri ve karmaşık yapıları nedeniyle, veri kümeleri genelde istatistik algoritmaları, makine öğrenimi ve sinyal işleme yöntemleri tarafından az çok benzer hedefler için işlenir (verileri temizlemek, düzleştirmek veya filtrelemek; çıkarımsal ve analitik tahminleme görevleri gerçekleştirmek; verilerdeki yapıları ve ilişkileri belirlemek ve/veya görselleştirmek için). Bu tür görevler için geliştirilen algoritmalar, boyut ve/veya karmaşıklık nedeniyle insanlar tarafından ele alınamayan verilerden yeni bilgi ve enformasyon çıkardıklarından öğrenme algoritmaları olarak da adlandırılırlar. Örneğin, hava durumu tahminleri, tıbbi arşivlerin tıbbi bilgiye dönüştürülmesi ve genomik verilerin analizi, kendi tecrübelerinden öğrenebilen bilgi işlem makineleri yardımı olmadan insanlar tarafından ele alınamayan bu tür öğrenme görevlerinin yalnızca birkaç örneğidir.

Veritabanındaki veriler kümelere ayrılırken, benzerlik ve uzaklık kavramlarından yararlanılır. Bu, veritabanındaki her bir kaydın diğer bir kayıtla olan benzerliği ya da her bir kaydın veritabanındaki diğer kayıtlardan olan uzaklığı olduğu gibi oluşturulan gerçek ve aday kümeler arasındaki mesafe ve benzerliği de içerir. Sözgelimi, veriler birbirilerine olan uzaklığa göre başlangıçta 8 ayrı kümeye ayrıldılarsa, bu 8 ayrı kümenin gerçekten farklı özelliklere sahip birer küme olup olmadığının da belirlenmesi gerekecektir. Bu durumda, oluşturulmuş bu kümeler arasındaki mesafe / benzerlik de ölçülmelidir. Birbirinden pek de farkı olmayan kümeler birleştirilerek tek bir küme haline dönüştürülebilir. Bu işlem, tüm veriler taranıp kümeler ortaya çıktıktan sonra yapılabileceği gibi veritabanının taranması ve veriler arasındaki benzerlik ve mesafenin ölçümü esasında da yapılabilir. Bu nedenle kümeler arasındaki mesafenin ölçülmesiyle iki veya daha fazla kümenin birleştirilmesi söz konusu olduğu gibi aynı zamanda, bir kümeden birden fazla küme üretilmesi de söz konusu olacaktır. Bunun için de sürekli olarak kümelerin büyüklüğü ve çapı ölçülmelidir.

Çeşitli öğrenme algoritmaları dolaylı veya açıkça veri setindeki nesneler arasındaki mesafe / benzerlik kavramına dayanabilir. Bu tür algoritmaları belirli bir uygulama için kullanırken veya özelleştirirken, algoritma tasarımcısı veya uyarlama yapanlar genellikle aşağıdaki sorunlar ile yüz yüze kalmaktadırlar. Eldeki veriler için en uygun mesafe/benzerlik ölçüsünün ne olduğunu analize başlamadan veya analiz esnasında belirlemek gerekmektedir. Uygun mesafe / benzerlik ölçütü yalnızca verilerin doğal gruplarını ve yapısını ortaya çıkarmakla kalmaz, veri analizi, çıkarsama, tahminleme veya karar alma süreçlerinin daha sonraki aşamalarının etkinliğini ve performansını da geliştirir. Bazı durumlarda, veriler, uygulama alanı, önceliğe ilişkin bilgi, sorunun fiziği veya veri üretme süreci aşamasında benzerlik ya da mesafe kavramıyla donatılmış oluyor. Bununla birlikte, bu tür bilgilerin mevcut olmadığı durumlar da vardır. Dolayısıyla, tasarımcı ve uygulayıcı, eldeki veri seti için uygun bir benzerlik veya mesafe ölçümünün nasıl tanımlanacağı konusunda önemli bir soruyla yüzleşmek zorundadır.

6.2. Kümeleme Algoritmalarının Sınıflandırılması

Literatürde birçok kümeleme algoritmasının adı geçmektedir. Algoritmalar birbirlerinden kümelemenin oluşturuluş şekline göre ayrıldıkları gibi kullanılan veri türüne, yapılacak olan çalışmanın amacına göre de farklılıklar gösterirler [Han ve Kember, 2001]. Kümeleme algoritmaları, genel olarak hiyerarşik ve bölümlemeli olarak ikiye ayrılırken, bu konuda yapılmış bir literatür taraması bu algoritmaların daha alt bölümlere ayrılabileceğini göstermektedir [Berkin, Son erişim tarihi: 4.09.2017].

Hiyerarşik Yöntemler

- Toparlamalı (Agglomerative) Kümeleme Algoritmaları.
- Ayrıştırmalı (Divisive) Kümeleme Algoritmaları.

Bölümlemeli (Partitioning) Yöntemler

- Yer değiştiren Algoritmalar.
- Olasılıksal Algoritmalar.
- k-Medoid Yöntemler.
- k-Means Yöntemler.
- Yoğunluğa Dayalı Algoritmalar.
  - Yoğunluğa Dayalı Bağlantılı Kümeleme.
  - Yoğunluk Fonksiyonlu Kümeleme.

Grid Temelli Yöntemler

Kategorik verinin yinelenmesine Dayanan Yöntemler

Kısıtlara Dayanan Yöntemler

Makina Öğrenmesi Alanında Kullanılan Yöntemler

- Gradyent İnme ve Yapay Sinir Ağları.
- Ölçeklenebilir Kümeleme Yöntemleri.

Bu yöntemlere ait olan algoritmaların amaca göre birçok varyasyonları mevcuttur. Örneğin, klasik k-Means algoritmasının bulanık hali Fuzzy c-Means, veya global çözümlere ulaşmak için geliştirilmiş Global k-Means algoritmalarını söyleyebiliriz. Bu tezde bakacağımız yöntem bölümlemeli yöntemlerden olacaktır.

6.3. Bölümlemeli Yöntemler

Bölümlemeli yöntemlerde n adet nokta önceden verilen k küme sayısına ($k < n$) göre kümelere ayrılır. Hiyerarşik yöntemlerin tersine kullanıcı tarafından verilen bazı kriterlere uygun kümeler yaratılırken, yaratılacak küme sayısı önceden belirlidir. Kullanıcı algoritmaya kümeler arasındaki minimum/maksimum mesafeyi ve kümelerin iç benzerlik kriterilerini de vermek zorundadır [Giudici, 2004].

Bölümlemeli algoritmalar genel olarak hiyerarşik algoritmalardan daha hızlı çalışırlar, çünkü hiyerarşik algoritmalardaki gibi bir benzerlik/mesafe matrisi kullanmak zorunda değillerdir. Bundan dolayı da büyük veritabanlarının kümelenmesinde hiyerarşik yöntemlere göre daha uygundur. Bununla beraber önceden verilen kritere uygun birden fazla sonuç çıkarmak mümkün olabilir. Bu durumda algoritmanın gerçekten en uygun çözümü bulup bulamadığı ise hiçbir zaman bilinmeyebilir [Dunham, 2003]. Bunun öğrenilebilmesi için verilerin dağıtılarak, sıra ve yerleri değiştirilerek, algoritmanın tekrar çalıştırılması gerekecek ve çıkan sonuçların birbiriyle kıyaslanması gerekecektir. Bu da zaman maliyetini oldukça artıracaktır.

6.4. Hazır Mesafe ve Benzerlik Ölçekleri

Veriler vektörler ya da matrisler biçimindeyse ya da özellik çıkarımı ile bu yapılara dönüştürülmüşse, çeşitli hazır mesafe ve benzerlik ölçekleri kullanılabilir. Örnekler arasında Minkowski mesafesi, Mahalanobis mesafesi, Matsushita

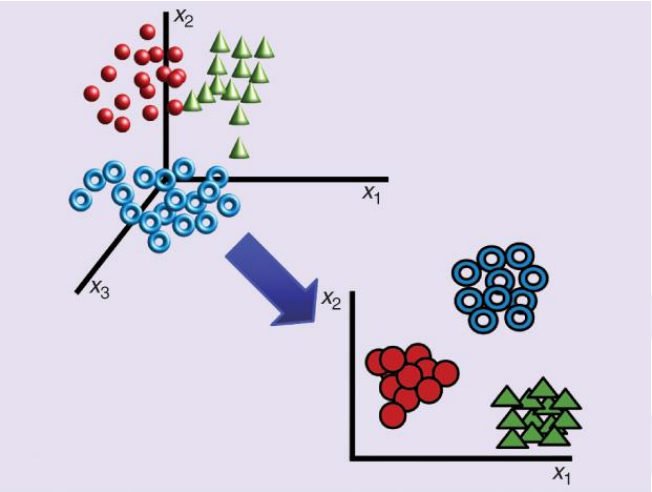
mesafesi, histogramlar için ki-kare mesafesi (Hamogramlar için), Hamming mesafesi, kosinüs benzerliği, skalar çarpım, çekirdekler, altuzaylar arasındaki açılar ve Grassmann mesafesi sayılabilir. Özellik çıkarma hazır mesafe fonksiyonlarının kullanılması, bugüne kadar çeşitli uygulamalarda başarılı olarak kullanılmıştır. Bununla birlikte, bilim ilerledikçe ve daha karmaşık veri türleri kaydedildikçe, böyle naif yaklaşımların her tür veri için uygun olup olmayacağı sorgulanmaya başladı. Bakılan veriye uyumsuz tanımlanan bir mesafe fonksiyonunun verinin öz nitelikli yapısını yakalaması beklenemez. Dolayısıyla, verilerin özündeki belirli yapıları yakalayabilen, veriye özel ve veri bağımlı mesafe işlevleri tasarlamanın mümkün olup olmadığını sorgulamak gereklidir. Bu tür uyarlanabilir mesafe işlevleri için örnek olarak, dinamik programlama tabanlı mesafeleri, örneğin zaman serileri ve ardışık veriler için dinamik zaman bükme (dynamic time warping, DTW) mesafesini göstermek mümkündür. Daha önce listelenen mesafe işlevlerinin aksine, DTW hesaplama işlemi, herhangi bir sıralı veri için üç işlemi tanımlamayı gerektirir: ekleme, silme ve eşleme işlemleri. Dolayısıyla, DTW, tanım gereği, bu işlemler aracılığıyla her bir veri kümesi için uyarlanabilir.

6.5. Göreve Özel Uzaklık ve Benzerlik Fonksiyonları

İlk önce daha önceki basit yaklaşıma kıyasla önemli gelişmeler kaydeden iki yaklaşım sunabiliriz: 1) mesafe ve benzerlik ölçeklerine örnek olan özellik çıkarımını iyileştirmenin daha iyi performans sağladığı yaklaşım ve 2) göreve özel mesafe ve benzerlik işlevlerini öğrenme yaklaşımı. Bu iki yaklaşımdan anlaşılıyor ki, öğrenme görevinin önceden bilinmesi gereklidir. Başka bir deyişle, mesafe (veya benzerlik) fonksiyonu amaca ve veriye bağımlıdır. Dahası, bu yaklaşımlar denetimli (supervised) öğrenmenin ayarları ile sınırlıdır.

Denetimsiz (unsupervised) öğrenmede bu iki yaklaşım kümeleme, boyut azaltma, bu problemleri çözen algoritmalar ve model seçme yaklaşımlar gibi bazı problemler için yararlı olabilir. Bu yaklaşımların çoğunun vektör biçimindeki verilerle de sınırlı olduğunu unutmamak gerekir. Bu iki yaklaşımın yanı sıra,

çekirdek yöntemler ve bu yöntemlerin çeşitli öğrenme algoritmalarında kullanımını da gözden geçirmek gerekir. Çekirdek yöntemlerin gücü modülerlik ve esnekliğinden kaynaklanmaktadır, çünkü denetimli ve denetimsiz olan herhangi bir biçimde veya yapıdaki veriyle çalışabilirler.



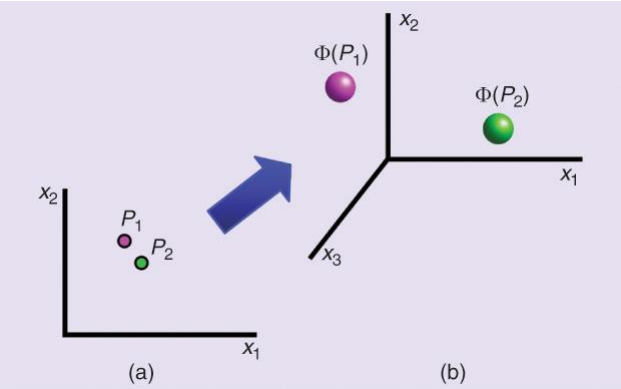
Şekil 6.1. Verilerin üç boyutlu uzaydan (resimdeki sol üst köşe) iki boyutlu uzaya (resimdeki sağ alt köşe) geçirilmesi örneği.

Birinci yaklaşım için LDA (Doğrusal diskriminant analizi, Linear discriminant analysis) iyi bilinen bir tekniktir [Shalev ve Ben, 2014]. Farklı sınıflardan örnekleri ayırt etmeye çalışan mesafeye dayalı bir öğrenme algoritması, Öklid uzaklığı veya Mahalanobis uzaklığı gibi piyasada bulunan uzaklık fonksiyonlarını kullanarak iki boyutlu alanda daha iyi çalışacaktır. LDA doğrusal boyut azaltma tekniği (linear dimensionality reduction) olarak da bilinir. Metrik öğrenme algoritmaları prensip olarak farklı formülasyonlara sahip olsalar da, LDA'ya çok benzemektedirler. Şekil 6.1'de, LDA verileri yüksek üç boyutlu uzaydan (resimdeki sol üst köşe) daha düşük iki boyutlu uzaya (resimdeki sağ alt köşe) geçiriyor. LDA'nın amacı, aynı sınıftan (diskler, daireler ve üçgenler) olan noktalar arasındaki uzaklıkların (Öklid veya Mahalanobis) minimize edildiği ve farklı sınıflar arasındaki uzaklığın en üst düzeye çıkarıldığı düşük boyutlu bir altuzay bulunmasıdır. Yani LDA, düşük boyutlu uzayda noktaların kompakt kümelerini tanımlamaya çalışmaktadır.

LDA, bazı varsayımlar altında gözlemlenen iki grup arasında en iyi ayırım yapabilen bir alt uzayı öğrenir: Kontrol edilen hastalara karşı gerçek hastaları, A kişisinin yüz görüntülerinin B kişisine, spam maillerin spam olmayanlara vb. kıyaslamaları gibi örnekler söylenebilir. Verileri bu alt uzaya yansıtmak ve Öklid veya Mahalanobis uzaklığına dayalı bir sınıflandırıcı uygulamak olası en düşük sınıflandırma hatasını vermiş olur. Filtreler ve paketleyiciler gibi diğer yaklaşımların da etkili olduğu bilinmektedir.

İkinci yaklaşım için kullanılabilen yöntem Uzaklık Öğrenme (Distance Metric Learning, DML) algoritmalarıdır [B. Kulis, 2013]. DML, LDA'nın amacına ve kullanımına çok benzer ve LDA'nın ikili formülasyonu olarak görülebilir. Bununla birlikte, DML'deki yenilik, LDA'nın yaptığı gibi, ayrımcı bir alt uzayı öğrenmek yerine, bir mesafe fonksiyonunun öğrenilmesi sorununu doğrudan ele almasıdır. Bu farklı görüş, genelleştirilmiş kuadratik mesafe fonksiyonlarını öğrenmek için çok modern ve verimli algoritmaların oluşturulmasını sağlamıştır.

Makine öğrenimi üzerinde büyük etkisi olan bir yaklaşım, çekirdek yöntemlerinin tanıtılmasıdır [Scholkopf ve Smola, 2001]. Çekirdek yöntemlerin temel bir avantajı, modüler olmalarıdır; Öğrenme algoritmasını veriden ayırırlar. Öğrenme algoritması, veri üzerinde yalnızca, özellik alanındaki veri nesneleri arasındaki çekirdek işlemleri yoluyla çalışır. Çekirdek işlevinin kendisi görev bağımlı değildir ve bu nedenle, denetimli ve denetimsiz (denetlenmeyen) öğrenme algoritmalarında kullanılabilir. Prensip olarak, belirli bir öğrenme görevi için çekirdeğe dayalı bir algoritma, uygun çekirdek fonksiyonu ile donatıldığı sürece, herhangi bir veride de çalışabileceğini vurgulamak gerekir. Bu durum ise iki zorluk oluşturur: 1) Yalnızca çekirdek üzerinden veri üzerinde çalışabilen öğrenme algoritmaları tasarlama ihtiyacı ve 2) her farklı veri kümesi türü için uygun çekirdek dizayn etme ihtiyacı. Bunlara veri bağımlı veya veriye özel çekirdek fonksiyonları denir. Ayrıca, genellikle iyi tanımlanmış mesafe fonksiyonlarından çekirdeklerin elde edilmesi mümkündür. Örneğin, Şekil 6.2'deki çekirdek fonksiyonu iki boyutlu bir uzayda bulunan P1 ve P2 noktalarını, daha yüksek boyutlu bir uzaya eşleyen bir haritalama fonksiyonu Φ ile ilişkilendirilir.



Şekil 6.2. İki boyutlu uzaydan üçboyutlu uzaya taşıyan çekirdek fonksiyonu örneği

Manifold öğrenme algoritmaları, özel çekirdek yöntemlerinin bir türüdür. Özellikle çok yönlü öğrenme algoritmaları, istatistik ve psikometri alanlarındaki iki klasik doğrusal boyut azaltma algoritmasının çekirdekleştirilmiş haline (temel bileşen analizi ve klasik çok boyutlu ölçekleme) dayanır. Bununla birlikte, hazır çekirdek fonksiyonları (radyal temel işlevi çekirdeği, polinom vb.) kullanmak yerine, veri bağımlı çekirdekler kullanılır. Çok yönlü öğrenme algoritmalarının çeşitli türleri (versiyonları) arasındaki farklılıklar, her bir algoritma tarafından farklı veri bağımlı çekirdek fonksiyonunun kullanılmasının sonucudur. Bu algoritmalar, denetimsiz doğrusal olmayan boyut azaltma yöntemleri olarak tanıtılır ve verilerin doğrusal olmayan biçimde yansıtıldığı daha düşük boyutlu Öklid alt uzayında öğrenme gerçekleştirirler. Böyle düşük boyutlu alt uzay, bir özellik çıkarma gibi de algılanabilir. Veri bağımlı çekirdek fonksiyonunun verilerin özlü yapısını da içerdiğini varsayarsak, bu tür düşük boyutlu Öklid alt uzayları kümeleme ve görselleştirme için de uygun olabilir, çünkü verilerdeki doğal gruplamayı ve yapıyı ortaya çıkarmak eğilimindedirler. Yine, bu tür algoritmaların performansı uygun veriye bağımlı çekirdek fonksiyonlarına bağlıdır.

6.6. Ortalama (Mean), Ortanca (Median) ve Mod (Mode)

Büyük bir veri kümesiyle çalışırken, tüm veri kümesini, kümenin "orta" veya "ortalama" değerini tanımlayan tek bir değerle göstermek yararlı

olabilmektedir. İstatistikte bu değer duruma göre, merkezi eğilim, ortalama, medyan ve mod olarak adlandırılır. Mean, yani ortalama değeri bulmak için veri kümesindeki değerler toplanıp toplanan değer sayısına bölünür.

Medyan, yani ortanca değeri bulmak için veri kümesinin değerlerini sayısal sırada listeleyip, listenin ortasındaki değeri almak gereklidir. Bu yöntem veriler üzerinde toplama ve çıkarma gibi işlemlerin yapılamadığı, fakat verilerin mantıksal olarak sıralanabildiği durumlarda işe yaramaktadır.

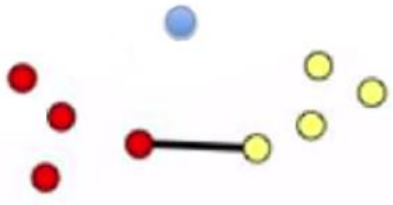
Mod değerini bulmak için, veri kümesindeki en sık rastlanan değeri seçmek gereklidir. Bu yöntem verilerin üzerinde hesaplama yapmaktan ziyade, bu verilerin sıralanamadığı durumlarda kullanılmaktadır. Sıralanamayan kategorik verilerde bu yöntem yapı gereği oratama bulmak için daha sık kullanılmaktadır.

6.7. Tek-İlişki (Single-Linkage), Tam-İlişki (Complete-Linkage) ve Ortalama-İlişki (Average-Linkage) Kümeleme Algoritmaları

Hiyerarşik kümeleme, her veri noktasını ayrı bir küme olarak ele alır ve sonra tüm noktaları tek bir kümede birleştirinceye kadar kümeleri birbiri ardına birleştirir. Hiyerarşik kümeleme sonucu genellikle dendrogram şeklinde verilir [Manning vd., 1999].

Single-Link algoritması tek bağlantı ya da en yakın komşu tekniğini kullanır [Sibson, 1973]. Bu teknikte, kümeler arasındaki mesafe ölçülürken, iki küme içinde birbirine en yakın iki elemanın uzaklığı ya da başka bir deyişle iki kümeyi en yakın kılan elemanların mesafesi kümeler arası mesafe olarak kabul edilir (Şekil 6.3). Algoritmanın zaman karmaşıklığı $O(n^2)$ dir.

$$d(c_1, c_2) = \min_{x_1 \in c_1, x_2 \in c_2} d(x_1, x_2) \qquad (6.7.1)$$



Şekil 6.3. Single Linkage algoritmasında, kümeler arası mesafe için en yakın noktaların alınması.

Öncelikle eldeki verilerden mesafe/benzerlik matrisi çıkartılır; bu matris bir ağaç haline dönüştürülür. Şebeke modellerinden en küçük maliyetli ağaç çıkartılarak, verilen eşik değerine göre kümeler oluşturulur.

Single-Linkage algoritması, k-en yakın komşu algoritmasının bir türevidir. Bu teknik literatürde en yakın komşu kümelemesi olarak da adlandırılır [Dunham, 2003]. Hatırlanacağı gibi, en yakın komşu algoritmasında k sayısının değeri önemlidir. Örneğin k=1 iken sadece iki nokta arasındaki mesafe göz önüne alınırken, k=2 için her bir kümeden iki ayrı, toplamda dört ayrı noktanın mesafeleri göz önüne alınır. Single-Linkage algoritmasında iki en yakın elemanın uzaklığı ya da başka bir deyişle iki kümeyi en yakın kılan elemanların mesafesi kümeler arası mesafe olarak kabul edilir.

Algoritmanın adımları şöyle özetlenebilir:

Girdiler:

Veri Kümesi

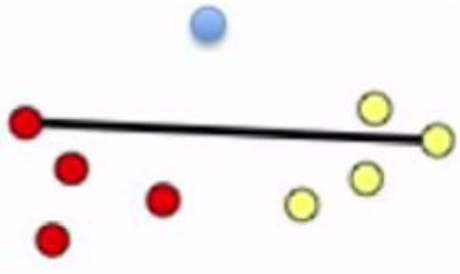
Bağlantı eşikdeğeri

Adımlar:

1. Eldeki verilerin mesafe/benzerlik matrisini çıkart.
2. Bu matrisi bir ağaç (*dendogram*) haline dönüştür.
3. Şebeke modellerinden en küçük maliyetli ağacı uygula ve çıkart.
4. Verilen eşik değerinin altında kalan dalları kopar ve kümeleri oluştur.
5. Dur.

Bu sınıftan olan diğer yöntemlerden Complete-Linkage (Tam bağlantı) algoritması ise iki küme arasındaki mesafeyi hesaplarken bu kümelerin birbirine en uzak iki noktasının uzaklık değerini ele almaktadır. Bunun amacı kümedeki tek bir noktanın değil, tüm kümede olan noktaların uzaklığa etkisini ele almaktır (Şekil 6.4).

$$d(c_1, c_2) = \max_{x_1 \in c_1, x_2 \in c_2} d(x_1, x_2) \qquad (6.7.2)$$

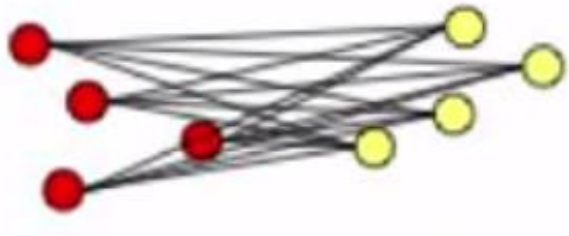


Şekil 6.4. Complete Linkage algoritmasında, kümeler arası mesafe için en uzak noktaların alınması.

Complate-Linkage algoritmasının zaman karmaşıklığı en kötü durum için $O(n^2 \log n)$ dir.

Average-Linkage (Ortalama bağlantı) algoritması, her iterasyonda en yüksek kohezyona sahip olan kümeler çiftini birleştirir (Şekil 6.5). Veri noktalarımız Öklid uzayında normalleştirilmiş vektörler olarak gösterilirse, C kümesinin G kohezyonunu noktaların ortalama çarpımı olarak tanımlayabiliriz:

$$d(c_1, c_2) = \frac{1}{|c_1|}\frac{1}{|c_2|} \sum_{x_1 \in c_1} \sum_{x_2 \in c_2} d(x_1, x_2) \qquad (6.7.3)$$

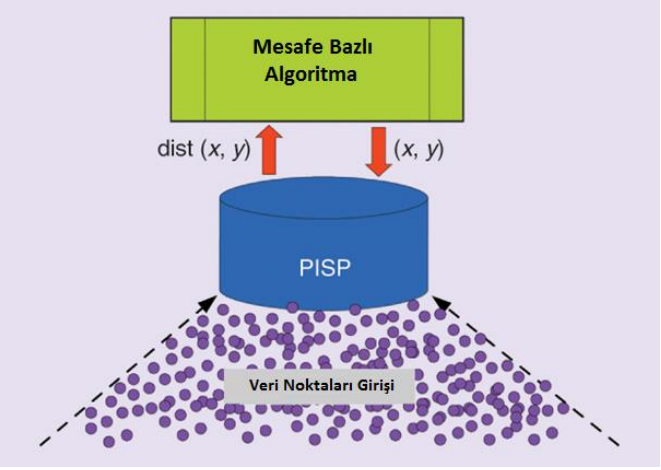


Şekil 6.5. Kümeler arasında Average Linkage uzaklığı.

İç-içe iterasyonlardan dolayı bu algoritmanın zaman karmaşıklığı $O(n^2 \log n)$ dir. Bu yaklaşımın avantajı olarak, gürültülü verilerden çok etkilenmemesini gösterebiliriz [Defays, 1977].

7. VERİ VE ÖĞRENME ALGORİTMALARININ ARASINDA ARAYÜZ KATMANI

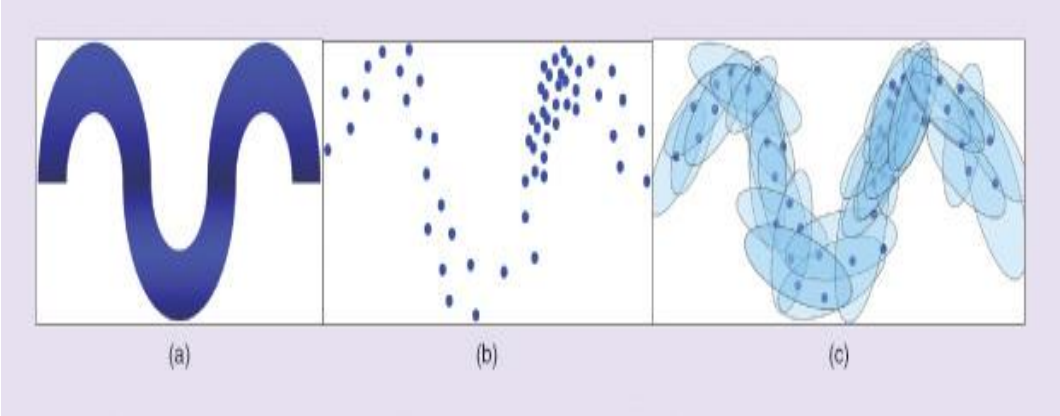
Önceki bölümde tanıtılan kurulum ve perspektif, bizi veri bağımlı mesafe fonksiyonlarını tanımlamak için yaklaşım düşünmeye teşvik etmektedir. Bu doğrultuda güncel araştırma çalışmaları, sonlu bir veri kümesinden jeodezik mesafeyi yaklaşıklaştırmaya doğru atılmış bir adımdır. Bu yaklaşım, eldeki seyrek veri kümesindeki veri manifoltunun yaklaştırılmasına odaklanmaktadır. Bu yaklaşım, giriş veri seti ile herhangi bir mesafeye dayalı algoritma arasında bir arayüz katmanı getirilerek sağlanır (Şekil 7.1). Algoritma, iki nesne arasındaki mesafeyi (veya benzerliği) sorarsa, bu sorguyu, veri manifoltuna yaklaşıklaştıran arayüz katmanına gönderir. Böylece, arayüz katmanı tarafından sağlanan mesafe, verilen veri kümesinin nesneleri arasındaki mesafe olarak ele alınacaktır. Teknik dilde ve rolünden dolayı arayüz katmanı pilot giriş alanı (PISP) olarak adlandırılır. PISP ana fikri, mesafeye dayalı bir algoritma olarak, bir arayüz katmanı aracılığıyla veri ile etkileşime geçer. PIPS katmanı girilen verilere göre veri-bağımlı mesafe ölçeklerini öğrenir. Algoritma, x ve y arasındaki mesafeye gerek duyduğunda, o, mesafeyi PIPS‘dan sorar ve PIPS kendi sırasında algoritmaya gereken mesafe değerini geri döndürür. (Şekil 7.1).



Şekil 7.1. PISP katmanının etkileşim şeması.

Şekil 7.1, PISP'nin "dikişli" elipsoidal yamalar aracılığıyla veri manifolduna nasıl yaklaştırdığını göstermektedir. PISP katmanı, uzaklık tabanlı algoritmalara,

noktaların yoğunluğu ve yaklaşık eğriliği (verinin altta yatan manifoldu) gibi, verilerin içsel bilgilerini dikkate alan yeni bir mesafeyi iletir. Bu arabirim katmanı, kümeleme algoritmaları için sistematik iyileştirme ve çeşitli gerçek veri kümelerinde boyutun azalmasını sağlar [Abou-Moustafa vd., 2013]. Burada sunulan PISP üzerine yapılan araştırmanın temel sonucu, yoğunluk ve eğrilik gibi, içsel bilgileri dikkate alan mesafelerin, uzaklığa dayalı veri analizi ve öğrenme algoritmalarının doğruluğunu arttırabilir.



Şekil 7.2. Giriş noktaları arasındaki mesafeyi tanımlamak için PISP tarafından kullanılan mekanizmanın bir gösterimi.

Şekil 7.2'de, giriş noktaları arasındaki mesafeyi tanımlamak için PISP tarafından kullanılan mekanizmanın bir gösterimi verilmiştir. (a): Gözlenlenmemiş orijinal veri manifoldu. (b): (a) daki manifolddan gözlemlenen sonlu nokta örnekleri, (c) her bir giriş noktası için, PISP birkaç en yakın komşusunu uygun mesafe metriğini kullanarak bulur. Bu, her noktanın komşuluğu olarak bilinir. Her mahalle için, PISP komşu noktalara tam bir kovaryans matrisi ile Gauss dağılımı uygular. Tek bir Gauss dağılımı, tam kovaryans matrisinden dolayı bir elips ile temsil edilir. Elips, veri noktalarının bulunduğu giriş alanının küçük bir bölümünü (bu durumda bir komşuluğu) kapsayan eliptik bir yama olarak görülebilir. Bu mekanizmanın toplam etkisi, örtüşen eliptik yamaların (a) 'daki orijinal manifoldun parçalı bir bölüm haline yaklaşık olmasıdır. Dolayısıyla, PISP'deki iki nokta arasındaki mesafe iki Gauss dağılımı arasındaki mesafedir ve bilgi dağılım ölçüleri kullanılarak ölçülebilir.

Bazı diğer çalışmalar ise, noktalar arasındaki jeodezik mesafeye yaklaşık olarak farklı mesafe türlerini öneriyor. Örneğin, izometrik yerleştirme (izometrik

haritalama) [Tenenbaum vd., 2000] algoritma sonlu örneklerin (vektörler) elemanlarını (veya nesnelerini) tüm veri setini birbirine bağlayan bir grafta tepeler olarak kullanır. Bu, *veri grafı* veya her nokta, en yakın komşusuna bağlıysa *komşuluk grafı* olarak bilinir. Veri kümesindeki herhangi iki öğe arasındaki mesafe, bu iki öğeyi veri grafı üzerinde birleştiren en kısa yolun uzunluğu olarak alınır. Bu ayarda, veri grafının manifoldun alttaki topolojisine yaklaştığı kabul edilirken, en kısa yol mesafesinin jeodezik mesafeye yaklaştığı varsayılmaktadır. En kısa yol mesafesi sezgisel olarak cazip olsa da, grafın kenarlarındaki ağırlıklara bağımlı ve duyarlıdır. Yoğunluğa dayalı mesafeler (Density-based distances, DBD) ve graf yoğunluğuna dayalı mesafeler [Sajama ve Orlitsky, 2005], verilerin altında yatan yoğunluğu göz önüne alan diğer mesafelere örneklerdir. Burada, manifold üzerindeki iki nokta arasındaki uzaklık, bu iki noktayı birbirine bağlayan yol olarak tanımlanır. Böylece, yüksek yoğunluklu bölgelerden geçen yollar, düşük yoğunluklu bölgelerden geçen yollardan çok daha kısa olmalıdır. İki noktayı birbirine bağlayan çeşitli yollar bulunduğundan, DBD bu iki noktayı birleştiren en kısa yol olarak tanımlanır. Buradaki sezgi, eğer bir nokta ile birkaç yakın komşusu arasındaki mesafe azsa komşuluğun yüksek yoğunluklu bir bölge olması beklenir.

Yoğunluk ve eğrilik bilgisini göz önüne alan daha temel yaklaşımlar, bilgi geometrisi ve hesaplamalı bilgi geometrisi üzerine literatürde bulunabilir. Bu literatürde bilginin ıraksama ölçümleri, böyle uzaklıklarını tanımlamak için koşum atlarıdır. Bu ölçümler, bir değişkenler kümesinde tanımlanmış olasılık dağılımları arasındaki uyumsuzluğu niceler [Amari ve Nagoka, 2000; Nielsen ve Bhatia, 2013].

Önceki kısımda sunulan fikirler, bu tezde daha önceden tartışılmış olanlardan farklı, yeni bir uzaklık ölçüm trendi tanımlar. Böyle uzaklık ölçümleri, mevcut sınırlı örneklerden denetlenmemiş bir tutumla öğrenilmiştir ve yoğunluk ve eğim gibi esas olan bilgiyi yakalar. Buradaki ana avantaj uzaklığın göreve bağımlı değil, veriye bağımlı olmasıdır ve dolayısıyla, geniş bir kapsamda, analitik, veri madenciliği ya da öğrenme için herhangi bir uzaklık tabanlı algoritma ile kullanılabilir.

Ek olarak, böyle uzaklıkları öğrenmek için mekanizma, uzaklık ölçümüne spesifik görevler ya da analiz tipleri için ince ayar yapılması yoluyla, şeffaf bir şekilde göreve bağımlı olmaya uydurulabilir.

Bu alanda araştırmalar, farklı alanlardaki farklı araştırma toplulukları arasında görece yeni ve kısıtlıdır. Bu kısa genel değerlendirme gerçekte ne tamdır ne de kapsamlı ve çok daha fazlası eklenebilir. Yine de hedef, farklı uygulama alanlarından bu fikirleri öne çıkarmaktır. Farklı uygulama alanlarından zorlayıcı taleplerle bu araştırma daha ileriye taşınabilir.

8. KARMA VERİLER İLE İLGİLİ YAPILAN ÇALIŞMALAR

Çalışmalarımızda sayısal verileri işlemekle sınırlı olmayıp kategorik verilerle de çalışan ve etkinliğini koruyan, aynı zamanda artımlı olup global çözüme daha iyi yakınsayan bir algoritma tasarlanmıştır. Verilerin nümerik olduğu durumda algoritma k-means tabanlı algoritma şeklinde çalışacaktır. Fakat verilerin karma (hem nümerik, hem kategorik) olduğu durumda kategorik veriler için nesnelerin benzerlik (uzaklık) tanımı farklı şekilde hesaplanmaktadır.

Standart hiyerarşik kümeleme yöntemleri, sayısal ve kategorik değerlere sahip verileri işleyebilmelerine rağmen, kuadratik hesaplama maliyeti nedeniyle büyük veri setleri için kullanılması kabul edilemez hale gelmektedir. k-means tabanlı yöntemler büyük verisetlerinde etkili olduğu için veri madenciliğinde kullanılmasını cazip kılar. Bu yöntemler ne yazık ki sadece sayısal veriler üzerinde çalışmaktadır. Bunun nedeni, bu algoritmalar veri noktaları ve kümelerin merkezleri arasındaki mesafeyi çeşitli uzaklık fonksiyonlarına göre (Manhattan, Öklid uzaklık vb..) tanımlanan maliyet fonksiyonu ile eniyilemeye çalışmaktadırlar. Bu türlü maliyet fonksiyonunu hesaplayarak minimize etmek algoritmaların sadece sayısal verilerle kullanılmasına neden olur. Karma verilerin de kümelemede etkin şekilde kullanılması için çeşitli çalışmalar yapılmıştır.

2005 senesinde Zengyou vd. karma – nümerik ve kategorik özelliklerden oluşan veri setlerinin nümerik ve kategorik parçalara ayırarak kümelenmesini gösteren bir makale yayınladılar. Makalede ana fikir olarak ele alınan veri setini özelliklerine göre iki yere, tamamen nümerik ve tamamen kategorik değerlere sahip olacak şekilde iki hisseye ayırmak ve sonra ayrı şekilde her kısım için uygun kümeleme yöntemi uygulamaktır. Çalışmanın sonrasında kümeleme sonuçlarını kategorik veri gibi bir araya getirmek ve sonuç üzerinde son bir kategorik kümeleme yöntemi uygulamakla sonuç kümeleri oluşturmayı hedeflemişlerdi [Zengyou vd., 2005].

2007 senesinde Ahmad ve Dey, k-means tabanlı kümeleme yöntemlerinin geliştirilerek karma veriler üzerinde çalışabilen farklı bir türünü yayınladılar. Ahmad ve Dey yeni karma veriler için yeni yakınlık ve amaç fonksiyonu sunmuşlardır. Bu çalışmadaki ana yaklaşımlardan biri karma veriler arasındaki uzaklığı hesaplamak için sadece o verilerin sıklığına (veri setinin tamamında

rastlanmasına) birebir bağlı olmayıp diğer veri özellikleri ile beraber rastlanma sıklıkları kullanılmıştır [Ahmad ve Dey, 2007].

Diğer bir çalışmada, Sally vd., 2010 senesinde nörobilim alanında kümeleme yöntemleri üzerine çalışma yapmışlardır. Bu çalışmada, kümeler arası korelasyon ve kümelenmiş verileri analiz etmek için bir dizi yöntem araştırılmış ve istatistiksel bir model sunulmuştur. Makale, veri analizi ve veri kümelemesinin kritik önemini vurgular ve veri kümelemesini hesaplamak için kullanılabilecek uygun istatistiksel yaklaşımları önerir. Çalışmanın amacı, veri kümelemesini çevreleyen bazı konular ele almak, bunları nörobilimcilerin erişebileceği bir biçimde sunmak ve kümelemede nörobilimde yaygın olarak karşılaşılan veri türlerine yaklaşımları incelemektir. Nöröbilimde bazı veriler ele alınmış ve bunlarla çalışmak için veri analizinin kritik önemine değinilmiştir [Sally vd., 2010].

Veri madenciliğinin iş dünyasındaki faydalarını ele alan bir çalışmayı da Chidanand vd. yapmışlardır. Makalede doğrudan müşteriye e-posta gönderimlerinde, perakende sektöründe, otomobil sigortası ve sağlık hizmetleri olmak üzere dört iş dünyası uygulama alanında veri madenciliği uygulamalarının nasıl yapıldığı ve ne gibi faydalar sağladığı ele alınmıştır. Veri madenciliğinin bireysel ve toplu davranışları belirlemeye ve tahmin etmeye nasıl yardımcı olduğunu göstermişler. Bu çalışmada, karar destek yapısının kurulması için veri analizine geleneksel yaklaşım ile beraber alan uzmanlığını istatistiksel modelleme teknikleriyle birleştirerek, belirli problemler için elle hazırlanmış çözümlerin geliştirilmesi sunulmuştur. Veri madenciliğinde kümelemenin önemi de vurgulanmıştır [Chidanand vd., 2002].

Carlos ve arkadaşları 2009 senesinde hastaların mekanik nefes alma makinesinden ayrılma denemeleri kayıtları üzerinde veri madenciliği ve sinir ağları çalışmaları yaptılar. Bu çalışmanın temel amacı, mekanik nefes alma denemelerinde, küme analizi ve sinir ağları uygulamalarını kullanarak hastalarda solunum yolu değişkenliğini analiz etmektir. Veri madenciliği süreçlerini kullanmakla, basit veri analizinden daha fazlasının yapılabildiği, verilerdeki detaylı eğilimleri tanımlamanın mümkün olduğu gösterilmiştir. Mekanik nefes almadan ayrılma süreci yoğun bakımda karşılaşılan zorluklardan biridir. Çalışmalarında ekstübasyon (çoğunlukla solunum yetmezliğindeki hastaları solunum cihazına bağlayabilmek için ağızdan nefes borusuna ulaşan bir boru

takmak gerekir. Bu boru takma işlemine "entübasyon" denir) işlemi altındaki 149 hasta incelenmiştir. Her hastanın değerleri 8 farklı zaman aralığındaki ölçümlerle karakterize edilmiştir. Solunum ve kardiyo sinyallerden 6 çeşit istatistik alınmıştır. Her bir hasta için 48 özellikten oluşan bir dizi izlenerek hareketli pencere (bant üzerinde kayma yapılmayla çizim çizme) istatistiksel analizi uygulanmıştır. Bir küme analizi uygulayarak, çoğunluğa göre iki grup elde edilmiştir. Gruplardan hastalar arasında ayrım yapmak için sinir ağları uygulanmıştır. Elde edilen en iyi performans, özellik seçme prosedürüyle (48 özellikten 19'u seçen) eğitilmiş doğrusal bir algılayıcı kullanılarak iyi sınıflandırılmış hastalarda ve girdi olarak ana küme merkezi kullanılarak 84.0% değeri elde edilmiştir. Çalışmada k-ortalamalar (k-means) yöntemi kullanılmıştır [Carlos A. vd., 2009].

Payam ve Owoc, 2011 yılındaki çalışmasında son birkaç on yılda veri madenciliği üzerine yapılan araştırmaların büyük ilerleme kat ettiğini belirtmişlerdir. Hastane bilgi sistemlerinin bir parçası olan hasta kayıtlarının artık daha kullanabilir, içeriğinin daha kapsamlı ve yayılma oranının geliştiği anlatılmış ve bu verileri ele alan matematiksel analiz modeller hakkında çalışma yapılmıştır. Makalede, güncel veri analizi akımlarına genel bakış sağlanmış ve hastane bilgi sistemlerinde hasta kayıtları ile analiz çalışmalarının tıbbi alan üzerindeki etkisi gösterilmiştir. Hasta kayıtları, heterojen bilgi sistemlerinde ve heterojen kaynaklardan heterojen veriler içerir. Sadece bilgi teknolojisi, elektronik hasta dosyalarındaki veri madenciliğinin karmaşıklığını artırmakla kalmaz, aynı zamanda bir hastanın hastalığını tanımlamaya çalışan doktorların tıbbi açıklamasının da göze alınması ihtiyacından dolayı analizlerin boyutu çok karmaşık olabilmektedir. Veri madenciliğinin özellikle bu alanda özellikle yararlı olduğu belirtilmiştir [Payam ve. Owoc, 2011].

Bellazzi ve Zupan, 2008, çalışmasında tahmini veri madenciliği araştırma alanının kapsamını ve rolünü tartışmak ve klinik tıp alanındaki veri madenciliği modellerini oluşturma, değerlendirme ve kullanma sorunlarıyla başa çıkmak için bir çerçeve önermişlerdir. Klinik tıpta prediktif veri madenciliği alanında yayınlanan son önemli çalışmalar gözden geçirilmiş, kritik konuları vurgulamış ve yaklaşımları bir dizi öğrenilmiş ders içinde özetlenmiştir. Makale, klinik tıpta prediktif veri madenciliği tekniğinin son durumunu kapsamlı bir şekilde incelemekte ve bu alandaki veri madenciliği çalışmalarını yürütmek için

yönergeler sunmaktadır. Genetik tıp içerisinde yer alan moleküler ve klinik verilerin entegrasyonu sayesinde, bu alan yakın zamanda yeni bir ivme ile beraber ele alınması gereken yeni karmaşık sorunlar dizisi yaratmıştır [Bellazzi ve Zupan, 2008].

1999 senesinde Lavrač tıpta uygulanabilen seçilmiş veri madenciliği teknikleri ve özellikle tıbbi veritabanlarının analizini daha uygun hale getiren mekanizmalar da dâhil olmak üzere bazı makine öğrenme teknikleri sunan bir çalışma yapmıştır. Makalede veri analizinin sonuçlarının yorumlanabilirliğinin önemi, seçilen tıbbi uygulamalar üzerinde tartışılmış ve anlatılmıştır. Çalışma veri madenciliği teknikleri ile ilgili olup teşhis, tarama, öngörü, izleme, tedavi desteği veya genel hastaların araştırılması için tıbbi bilgiyi elde etme yaklaşımları sunmaktadır. Makalede tıpta uygulanabilen seçilmiş veri madenciliği teknikleri ve özellikle tıbbi veritabanlarının analizi için uygun teknikler sunulmaktadır [Lavrač, 1999].

Literatürde bunlar gibi birçok çalışma yer almaktadır. Çalışmaların bir kısmı yeni yöntemlerin bulunması ve varolan yöntemlerin geliştirilmesi [Aranganayagi S. vb. 2009; Huang Z. vb. 2003; Khan S., Kant S. IJCAI-07, p. 2784-2789; Mastrogiannis N. vb. 2009; Moth'd Belal. vb. 2007; Peters M. vb. 2004; Rezanková H., 2014; Šulc Z. vb. 2014; Syal R. vb. 2012; Tripathy B. K.vb. 2011; Zengyou H. 2006; Zengyou He vb. 2008], diğer kısmı ise yeni uygulama alanları ve yaklaşımları sunmaktadır [Alexandros N. 2001; Jain A. K. vb. 1999; Murtagh F. 1983; William H. Day vb. 1984].

9. KÜMELEME PROBLEMİ

Verilerin kümelenmesi işlemi veri madenciliğinin önemli kısımlarından bir tanesidir. Büyük veri setlerini homojen kümeler halinde verimli bölümleme işi, veri madenciliği için temel problemlerdendir. Standart hiyerarşik kümeleme yöntemleri hesaplama verimsizliği nedeniyle bu probleme etkin çözümler sunamamaktadır. Büyük veri kümelerini işlemek için k-means tabanlı yöntemler daha verimli olmaktadır. Buna karşılık, bu tür yöntemlerin kullanımı genellikle sayısal veriler ile sınırlı olmaktadır. Ayrıca k-means tabanlı algoritmalar başlangıçta rastgele veriler ile işleme başlamalarından dolayı lokal sonuçlar üretirler. Bu algoritmalar deterministik olmayan algoritmalarıdır. Diğer yandan k-means algoritmasının artımlı versiyonları mevcuttur. Bu türden artımlı algoritmalar kümeleme sonuçlarında global çözümlere ulaşmayı sağlarlar ve deterministik olmaları önemli farklarındandır.

Ele alınan verilerin tipi ve kümeleme amacına göre kullanılan yöntemler farklılık göstermektedir. Sayısal yada kategorik veri türleri üzerinde bugüne kadar geliştirilmiş farklı kümeleme yöntemleri bulunmaktadır. Özellikle sayısal veriler üzerinde çalışan pek çok yöntem bulunmakta iken, kategorik veriler üzerinde çalışan fazla yöntem bulunmamaktadır. Kümeleme yöntemlerinden bazıları daha hızlı çalışırken, bazıları daha iyi amaç fonksiyon değeri hesaplamaktadır. Bu iki performans göstergesi baz alınıp her ikisini de iyileştirmeye çalışan rekabetçi algoritmalar geliştirilmeye çalışılmaktadır.

Gerçek hayat verilerine bakıldığında ele alınan verilerin sadece sayısal yada sadece kategorik olması gibi bir kısıtlamaya gitmek ele alınan veri kümesi üzerinde veri analizinin başarılı yapılması için bir engeldir. Bu nedenle karma veriler üzerinde çalışabilen ve yukarıda sözü edilen performans değerlerini arttırabilecek algoritmaların geliştirilmesi önemlidir. Bu bölümde kümeleme probleminin matematiksel modelleri ifade edilmiştir.

9.1. Kümeleme Probleminin Tanımı

m-boyutlu n noktadan oluşan R^m uzayında sonlu sayıda elemana sahip $\boldsymbol{X}$ kümesini ele alalım:

$$\boldsymbol{X} = \{X_1, X_2, \ldots, X_n\} \text{ , ve } X_i \in \mathbb{R}^m, \ \ i = 1, \ldots, n.$$

Buna göre kısıtsız ve tam bölümlemeli kümeleme probleminin amacı, $\boldsymbol{X}$ kümesindeki noktaları, verilen k adet ayrık $X_j, \ j = 1, \ldots, k$ alt kümeye, önceden tanımlanmış aşağıdaki kurallara göre ayırmaktır.

1) $X_j \neq \emptyset, \ \ j = 1, \ldots, k;$
2) $X_j \cap X_l = \emptyset, \ \ j, l = 1, \ldots, k, \ \ j \neq l;$
3) $X = \bigcup_{j=1}^{k} X_j;$
4) $X_j, \ \ j = 1, \ldots, k$ kümeleri için ilave koşul bulunmamaktadır.

$X_j, \ j = 1, \ldots, k$ alt kümelerine, küme (cluster) adı verilir. Burada oluşacak her bir kümenin boş küme olmaması, hiçbir küme çiftinin ortak bir elemana sahip olmaması, kümelerin birleşiminin veri kümesine eşit olması ve hiçbir küme için bir kısıt bulunmaması kuralları esas alınmaktadır.

Aynı kümeden olan noktalar olabildiğince birbirlerine benzer ve farklı kümeden olan noktalar ise birbirlerinden farklıdırlar. Veri noktalarının benzerlikleri benzerlik ölçüsü (similarity measure) ile tanımlanır. Bu ölçüm incelenen noktanın ait olduğu kümenin merkezine uzaklığı ile tanımlanır.

$d(x, y)$: x ve y noktaları arasındaki uzaklık olsun. Bu durumda kümeleme problemi aşağıdaki optimizasyon problemine dönüşür:

Min:

$$\psi_k(x, w) = \frac{1}{m} \sum_{i=1}^{m} \sum_{j=1}^{k} w_{ij} d\left(x^j, a^i\right) \qquad (9.1.1)$$

Burada,

$$x = (x^1, \dots, x^k) \in \mathbb{R}^{n \times k}, \tag{9.1.2}$$

$$\sum_{j=1}^{k} w_{ij} = 1\,, i = 1, \dots, m, \tag{9.1.3}$$

$$w_{ij} = 0 \text{ veya } 1, \;\; i = 1, \dots, m, \;\; j = 1, \dots, k \tag{9.1.4}$$

Burada w_{ij} : a^i örneğinin j kümesine aşağıdaki koşullara göre ait olma ağırlığıdır.

$$w_{ij} = \begin{cases} 1, & \text{eğer } a^i \text{ elemanı } j \text{ kümesine atandıysa,} \\ 0, & \text{aksi halde} \end{cases}$$

ve

$$x^j = \frac{\sum_{i=1}^{m} w_{ij} a^i}{\sum_{i=1}^{m} w_{ij}}, \qquad j = 1, \dots, k. \tag{9.1.5}$$

burada w, m x k boyutlu bir matristir.

$d(x, y)$ uzaklığı farklı normlarla tanımlanabilir. Bu çalışmada aşağıdaki şekilde tanımlanan Öklid normu kullanılmıştır:

$$d(x, y) = \left(\sum_{\mathrm{i}=1}^{\mathrm{n}} |\mathrm{x_i} - \mathrm{y_i}|^2 \right)^{1/2} \tag{9.1.6}$$

(9.1.1)-(9.1.4) problemi kümeleme probleminin karma tamsayılı doğrusal olmayan programlama modelidir. (9.1.1)-(9.1.4) deki kümeleme probleminin Pürüzlü Dışbükey olmayan (Nonsmooth Nonconvex) formülasyonu ise aşağıdaki şekilde verilebilir [Bagirov vd., 2002; Bagirov vd.,2003; Bock, 1998]:

$$\text{Min: } f_k(x^1, \dots, x^k) = \frac{1}{m} \sum_{i=1}^{m} min_{j=1,\dots,k}\, d\left(x^j, a^i\right) \tag{9.1.7}$$

$$(x^1, \dots, x^k) \in \mathbb{R}^{n \times k}, \tag{9.1.8}$$

Hem ψ_k hem de f_k fonksiyonları küme (cluster) fonksiyonları olarak adlandırılır. Yukarıda ifade edilen model ile (9.1.1)-(9.1.4) denklemlerinde ifade edilen model kıyaslandığında:

1. ψ_k amaç fonksiyonu, w_{ij} , $i = 1, \dots, m$, $j = 1, \dots, k$ (0 veya 1 olan katsayılar) ve $x^1, x^2, \dots, x^k$, $x^j \in \mathbb{R}^n$, $j = 1, \dots, k$ (sürekli değişkenler olan küme merkezleri) değişkenlerine bağlıdır. Diğer yandan f_k fonksiyonu yalnızca $x^1, x^2, \dots, x^k$ değişkenlerine bağlıdır. Buna dayanarak her iki fonksiyonun dışbükey olmadığını söyleyebiliriz.

2. (9.1.1)-(9.1.4) problemindeki değişkenlerin sayısı $(m + n) * k$ iken (9.1.7)-(9.1.8) probleminde bu sayı sadece $n * k$ dır ve dolayısıyla değişken sayısı veri sayısına bağlı değildir. Dikkat edilmelidir ki, gerçek hayattaki veri kümelerinde verilerin sayısı olan m, özelliklerin sayısı olan n değerinden çok daha büyüktür.

3. (9.1.1)-(9.1.4) ve (9.1.7)-(9.1.8) problemleri global enküçükleyicilerinin aynı olması anlamında birbirlerine denktirler. (9.1.1)-(9.1.4) modeli üzerinde kullanılan k-ortalamalara karşın (9.1.7)-(9.1.8) modeli üzerinde kullanılan global k-ortalamalar algoritması global optimum yada global optimuma yakın sonuçlar vermektedir.

9.2. Kategorik Modeller.

Kümeleme uygulamalarında verinin türüne bağlı olarak uygulanacak algoritma seçilmektedir. Algoritmaların çoğu verilerin nümerik değerler oluştuğunu varsayarak çalışmaktadır. Gerçek veri setlerinde veriler bu şekilde değildir. Çoğunlukla nümerik verilerle beraber kategorik veriler de gerçek veri setlerinde yaygınca bulunmaktadır.

Kategorik veri setleri veya verilerdeki kategorik özellikler cins, ırk bilgisi, posta kodları vb. gibi bilgiler diskret olup sıralamaları yapılamamaktadır. Genellikle veri setlerindeki verilerin özellikleri karışık olmakta, örneğin maaş

bilgisi nümerik olmasına rağmen cinsiyet veya posta kodu kategorik olabilmektedir. Kategorik veri setine örnek olan özel formlardan biri de market alışveriş sepeti veri setidir.

Kategorik veriler kümelemede çeşitli sorunlar yaratmaktadırlar:

- Uygulanan kümeleme yöntemi eğer benzerlik ve uzaklık metrikleri ile çalışıyor ise kategorik veriler için artık bu kullanılamazdır. Bunun yerine artık yeni uzaklık tanımı kullanmak gerekmektedir.
- *k*-means veya *k*-medians gibi birçok algoritma verilerde merkez, median temsilleri oluşturarak ilerlemekteler. Çoğunlukla bu yöntemler nümerik verilerden hesaplanarak oluşturulmaktadır. Bu aşamada hesaplamaların kategorik-diskret veriler ile de yapılması gereği oluşmaktadır.

Veri kümeleme yöntemleri kümelenecek verilerin türü ile ilişkilidir. Ölçekleme, normalleştirme ve yakınlık kavramlarını anlamak veri madenciliği sonuçlarını doğru hesaplama ve yorumlamada önemlidir. Veri türleri kümelemenin niceliğine etki etmektedir. Tek bir veri özelliği ikili, diskret veya devamlı olabilir. İkili özellik sadece iki değere sahip olabilir, örneğin doğru veya yanlış gibi. Diskret veri türü sonlu sayıda farklı değerlere sahip olabilmektedir. İkili veri türü diskret veri türünün özel bir alt kümesidir.

Değerlerin önemini belirleyen veri skalası veya ölçekleme de kümelemede çok büyük öneme sahiptir. Ölçekleme niteliksel ve niceliksel olabilmektedir. Önceki bölümlerde de belirtildiği gibi, nitel ölçekler arasında nominal ölçekler ve sıra ölçekleri bulunmaktadır. Niceliksel ölçekler, aralık ölçekleri ve oran ölçeklerini içerir.

Veri karışık olduğunda kümeleme daha da karmaşıklaşır, çünkü çeşitli veri özelliklerini heterojen olarak ele almak ve yeni veriye özel uzaklık fonksiyonu tanımlamak gerekmektedir.

Bazı kümeleme yöntemlerinin çeşitli veriler için daha uygun olduğunu söylemek gerek. Örneğin, bazı yöntemler sadece veriler arasındaki mesafe (benzerlik) fonksiyonuna bağımlılardır. Başka bir deyişle, veriler arasındaki benzerlik fonkiyonu doğru tanımlanırsa o zaman kümeleme yönteminin etkin şekilde çalışacağı söylenebilir. Spektral kümeleme öyle bir kümeleme türüdür ki, burada uygun benzerlik fonksiyonu olduğu sürece her tür veri ile çalışmak

mümkündür. Bu yaklaşımın olumsuz yanı yöntem skalası benzerlik matrisinin karesi kadar büyümesidir. Genellenen modeller de benzer biçimde veri setinin her bir bileşeni için uygun genellenmiş bir model oluşturulabilirse, her tür veriyle çalışabilmektedir. Kategorik veri kümelemesi için en yaygın algoritmalar k-modes [Chaturvedi vd., 2001], k-prototypes [Huang, 1998], CACTUS ve STIRR [Ganti vd., 1999], ROCK [Guha vd., 2000] ve LIMBO [Andritsos vd., 2003] algoritmalarıdır.

9.3. Artımlı (Incremental) Modeller.

k-means ve benzer sınıftan kümeleme algoritmalarında sonuç çıktıları başlangıçta alınan rastgele değerlerden büyük ölçüde etkilenmektedir. Bu özelliklerinden dolayı sözü geçen algoritmalar lokal çözümler üretmektedirler. Bu sorunun çözümü için literatürde farklı yaklaşımlar önerilmiştir. Alınacak sonuçların rastgelelikten etkilenmemesi ve ele alınan veri seti için tek global çözüm istendiği durumlarda artımlı yaklaşımlar kullanılmaktadır. Bu yaklaşımlarda, örneğin k-means algoritması için, baştan bir küme merkezi alınıp, adım-adım gereken küme sayısına kadar artımlı şekilde kümeleme yapılmaktadır. Artımlı model uygulandığı zaman, kümeleme sonuçları her seferinde aynı olacaktır. Bu da sonucun global sonuç olduğu anlamına gelmektedir.

Diğer yandan, modelin dezavantajı çalışma zamanıdır. Yapısı gereği daha çok kombinasyon kontrol ettiğinden dolayı bu modeller daha geç çalışmaktadır.

Zaman probleminin de çözülmesi için son yıllarda literatürde çeşitli öneriler verilmektedir [Bagirov, 2008].

9.4. İstatistiksel Veriler.

İstatistiklerle çalışırken, farklı veri türlerini tanımak önemlidir. Veriler, çalışmalar yoluyla toplanılan bilgilerin gerçek parçalarıdır. Örneğin, arkadaşlarınızdan beşine kaç adet evcil hayvanı olduğunu sorduğunuzda, size aşağıdaki verileri verebilirler: 0, 2, 1, 4, 18 (Beşinci arkadaş akvaryumdaki her

balığını ayrı bir evcil hayvan olarak sayabilir). Her veri sayısal olmayabilir. Diyelim ki, arkadaşlarınızın her birinin cinsiyetini de kaydetmek istediniz. Bu durumda, aldığımız veriler örneğin: erkek, erkek, kadın, erkek, kadın gibi olacaktır. Verilen genellikle iki gruba ayrılırlar: sayısal ve kategorik.

İki seçenek (dichotomous) ve ikiden fazla seçeneği tanımlayan verilere gözlemciler kategorik veriler demektedirler. Nümerik tanımlanmış yöntemlerle sayılabilen ve hesaplanabilen veriler ise nümerik olarak adlandırılmaktadır.

Kategorik özellikler niteliksel özellikler gibi ele alınabilinir. Basit bir örnekle, araba markaları veya banka şube isimleri gibi isimleri tanımlamak için kullanılabilirler. Veri kümesinin sonlu sayıda verilerden oluştuğunu dikkate alırsak, verilerin niteliksel özellik sayısı da sonlu olmuş olacaktır. Niteliksel veri türü için ise diskret veri türünün özel bir durumu olduğunu söyleyebiliriz.

Bu veriler, kişinin cinsiyeti, medeni hali, memleketi veya sevdiği film türleri gibi özellikleri temsil eder. Kategorik veriler, sayısal değerler de (örneğin '1' erkek, '2' kadın) alabilir, ancak bu sayıların matematiksel anlamı yoktur. Örneğin bu değerleri toplayamazsınız. Kategorik verilerin diğer adları niteliksel veriler veya Evet / Hayır verileridir.

Aşağıda kategorik verilerin tanımını, onlarla ilgili sembolik tabloları ve sıklık tablolarını vereceğiz.

$D = \{x_1, x_1, \dots x_n\}$ nin n veriden oluşan kategorik veri seti olduğunu ve her bir verinin d sayıda $v_1, v_2, \dots, v_d$ şeklinde kategorik özelliği olduğunu kabul edelim. $E(v_j)$, v_j özelliğinin alabileceği değerler kümesini ifade etmektedir (Tablo 9.1).

Tablo 9.1: Kategorik veri seti örneği.

Veriler	*Değerler*
$\mathbf{x}1$	(A, A, A, A, B, B)
$\mathbf{x}2$	(A, A, A, A, C, D)
$\mathbf{x}3$	(A, A, A, A, D, C)
$\mathbf{x}4$	(B, B, C, C, D, C)
$\mathbf{x}5$	(B, B, D, D, C, D)

Tablo 9.1 de gösterilen kategorik veri setinde uygun olarak v_1 için $E(v_i) = \{A, B\}$ ve v_4 için $E(v_4) = \{B, C, D\}$ olacaktır.

Verilen D kategorik veri seti için $E(v_i) = \left\{A_{j1}, A_{j2}, \dots A_{jn_j}\right\}$, $j = 1,2, \dots, d$ dir.

Biz $A_{j1}(1 \leq l \leq n_j)$ ye v_i kategorik özelliğin değeri diyeceğiz ve n_j ise v_i özelliğinin D verisetinde olası değerlerinin sayısı olacaktır. O zaman semboller tablosu olan T_s alttaki şekilde olacaktır.

$$T_s = (\boldsymbol{s_1}, \boldsymbol{s_2}, \dots, \boldsymbol{s_d}). \tag{9.4.1}$$

Burada $s_j, (1 \leq j \leq d)$, $s_j = (A_{j1}, A_{j2}, \dots A_{jn_j})^T$ şeklinde tanımlanan vektördür.

Her bir özellik için birden fazla değer olası olduğu için sembol tablosu genelde benzersiz değildir. Örneğin Tablo 9.1 için Tablo 9.2 ve Tablo 9.3 onun sembol tablolarıdır.

Tablo 9.2: Tablo 9.1 deki veri setinin sembol tablolarından biri.

$$\begin{pmatrix} A & A & A & A & B & B \\ B & B & C & C & C & C \\ & & D & D & D & D \end{pmatrix}$$

Tablo 9.3: Tablo 9.1 deki veri setinin diğer sembol tablosu.

$$\begin{pmatrix} A & B & D & A & B & C \\ B & A & C & C & C & B \\ & & A & D & D & D \end{pmatrix}$$

Tablo 9.4: Tablo 9.2 deki sembol tablosundan hesaplanan sıklık tablosu.

$$\begin{pmatrix} 3 & 3 & 3 & 3 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 \\ & & 0 & 0 & 1 & 1 \end{pmatrix}$$

Tablo 9.5: Tablo 9.3 deki sembol tablosundan hesaplanan sıklık tablosu.

$$\begin{pmatrix} 3 & 0 & 0 & 3 & 1 & 1 \\ 0 & 3 & 0 & 0 & 1 & 1 \\ & & 3 & 0 & 1 & 1 \end{pmatrix}$$

Sıklık tablosu da sembol tablolarından oluşmaktadır ve tamamen sembol tablolarının boyutlarındadır. C , bir küme olsun. O zaman bu kümenin $T_f(C)$ sıklık tablosu

$$T_f(C) = (\boldsymbol{f}_1(C), \boldsymbol{f}_2(C), \dots \boldsymbol{f}_d(C)) \tag{9.4.2}$$

olacaktır. Burada $\boldsymbol{f_1}(C)$

$$\boldsymbol{f}_1(C) = (f_{j1}(C), f_{j2}(C), \dots f_{jn_j}(C))^T \tag{9.4.3}$$

şeklinde tanımlanan vektördür. Burada $f_{jr}(C), (1 \le j \le d, 1 \le r \le n_j)$, C kümesinde j-nci boyutta A_{jr} değerini alan veri numarasıdır.

$$f_{jr}(C) = \left|\{\mathrm{x} \in C : x_j = A_{jr}\}\right|, \tag{9.4.5}$$

Burada x_j , x ın j - nci bileşenidir.

Verilmiş veri seti için her bir kümenin sıklık tablosu bu sembol tablosu için tekildir. Örneğin Tablo 9.1 de gösterilen veriler için, $C = \{x_1, x_2, x_3\}$ bir küme olsun. O zaman eğer Tablo 9.2 de gösterilen sembol tablosunu kullanırsak C kümesi için sıklık tablosu Tablo 9.4 deki gibi olacaktır. Eğer Tablo 9.3 deki sembol tablosunu kullanırsak C kümesi için sıklık tablosu Tablo 9.5 deki gibi olacaktır.

Verilen D veri seti için $T_f(C)$ sıklık tablosunun tüm veri setinin dikkate alınarak hesaplandığını görüyoruz. D veri setinin k sayıda örtüşmeyen $C_1, C_1, \dots, C_k$ kümelere ayrıldığını varsayalım. O zaman tüm $r = 1,2,\dots, n_j$ ve $j = 1,2,\dots, d$ için

$$f_{jr}(D) = \sum_{i=1}^{k} f_{ir}(C_i) \tag{9.4.6}$$

olur.

Nümerik veriler ise bir kişinin boyu, kilosu, IQ veya kan basıncı gibi değerlerini, veya bir kişinin sahip olduğu hisse senedi sayısı, bir köpeğin kaç dişi olduğu ya da uykuya dalmadan önce en çok sevdiğiniz kitaptan kaç sayfa okuyabilirsiniz vb. gibi değerleri göstermektedir. İstatistikçiler, sayısal verileri nicel veri olarak da adlandırırlar.

9.5. Sayısal Veriler: Kesikli (Diskret) ve Sürekli.

Kesikli veriler sayılabilecek öğeleri temsil eder, örneğin bir yılda kliniğe başvuran çocuk veya hasta sayısı gibi sadece belirli sayısal değerleri alabilen gözlemlerden ortaya çıkmaktadır. Bu veri türü listelenebilir değerler kümesine ait değerleri alabilmektedir. Olası değerlerin listesi sabitlenebilir (sonlu olarak da adlandırılır) veya 0, 1, 2 şeklinde sayılabilir sonsuza kadar rakamlar şeklinde de gidebilir. Örneğin, bir madeni parayı 100 defa attığımızda yazı veya tura olma sayısı 0'dan 100'e kadar olan sonlu sayıda değer alır. Ama madeni para atarak 100 defa yazı veya tura atma sayısı en kısa senaryoyla 100 den başlayarak sonsuz sayıda (yeterli sayıda atılmadığı sürece) devam edebilir. O zaman alınabilinecek değerler sonsuz sayılabilen (101, 102, 103, ...) olacaktır.

Sıralı kategorik veriler bazen diskret (ayrık) veriler olarak ele alınır, bu yanlıştır. Örneğin, sosyal sınıf sınıflandırmasında 1. sınıfın 5. sınıftan beş defa daha iyi sosyo-ekonomik durumda olduğunu söylemek yanlış olur, çünkü bu kategoriler arasında katı bir sayısal ilişki bulunmamaktadır. Dolayısıyla, ortalama değerinin de anlamsız olduğunu söyleyebiliriz. Böylece, sıralanmış kategorik veriler, istatistiksel analiz için diskret (ayrı) veriler olarak ele alınmamalıdır. Diskret (Ayrık) veriler istatistiksel analizde sıralanmış olarak ele alınabilir, ancak bunu yaparken bazı bilgiler kaybolur.

Sürekli veriler ölçümleri temsil eder; olası değerleri hesaplanamaz ve sadece gerçek sayılar çizgisinde aralıklarla tanımlanabilir. Örneğin, 20 litrelik tanklar için istasyondan alınan tam gaz miktarı, [0, 20] aralığı ile temsil edilen 0 litre'den 20 litreye kadar sürekli veriler olacaktır. 8.40 litre veya 8.41 veya 8.414863 litre pompalanabilir, 0'dan 20'ye kadar olası tüm reel sayılar kadar pompalanabilir. Bu sayede, sürekli verilerin sayılamaz sonsuz olduğu kabul edilmektedir. Kayıt tutma kolaylığı için istatistikçiler genellikle uygun sayılar belirleyerek yuvarlama yaparlar. Bir başka örnek, bir AA pilinin kullanım ömrü, 0 saatten sonsuza kadar (sonsuza kadar sürerse), teknik olarak bu aralıkta olası tüm değerler olabilir. Bataryanın ömrünün birkaç yüz saatten uzun sürmesi beklenmiyordur, ancak kimse ne kadar süre gidebileceğine bir sınır koyamaz.

Bu tip veriler, teorik olarak sonsuz küçük birimler halinde ölçülebilen sayısal verilerdir. Örneğin, kan basıncı genellikle 2mm Hg'ye yakın değerlerle ölçülür. Ancak çok daha büyük değerlerle de ölçülebilir. Aralık ölçeği, sürekli

veriler için tasarlanmıştır. Bazen sürekli veriler belirli eşiklerde ayrı değerler alır, örneğin, doğum günü ayrıktır ama yaşın kendisi sürekli bir değerdir. Bu durumlarda, ayrık değerleri sürekli veri türü olarak ele almak mantıklıdır. Genelde sıralı veriler üzerinde istatistiksel analizler yapmak kategorik verilere göre daha iyi sonuçlar verecektir.

10. SAYISAL VERİLER ÜZERİNDE KÜMELEME

10.1. k-Ortalamalar (k-means) algoritması

k-ortalama (k-means) algoritması kesin (hard) kümeleme için en çok kullanılan algoritmalardan biridir. Bu algoritma sürekli olarak kümelerin yenilendiği ve en uygun çözüme ulaşana kadar devam eden döngüsel bir algoritmadır. Bölümlemeli algoritmaların tipik özelliklerini taşır. Bu alandaki benzer algoritmaların çoğu ya *k-ortalama (k-means)* algoritmasından esinlenerek ya da bu algoritmanın geliştirilmesiyle ortaya çıkmıştır. Dolayısıyla bu algoritmanın anlaşılması bundan sonraki algoritmaların mantığının kavranmasında önemli bir rol oynayacaktır [Han J. ve Kamber M., 2001].

İlk olarak 1967 yılında ortaya atılan [MacQueen, 1967] k-ortalama algoritması eldeki verileri k adet kümeye ayırır. Veri herhangi bir kümeye atanırken kümelerin ortalamalarına yakınlık dikkate alınır. k küme sayısı kullanıcı tarafından verilir. Burada kastedilen ortalama daha önce belirtilen küme merkezidir.

Aşağıda algoritmanın kaba kodu verilmiştir:

Girdiler:

$\boldsymbol{X} = \{x_1, x_2, \dots x_n\}$ // eldeki veritabanı.

k // verilen küme sayısı.

Adımlar:

1. Keyfi olarak $q_1, q_2, \dots q_k$ merkezleri seçilir (merkezlerin $\boldsymbol{X}$ kümesine ait olması gerekmez).
2. Herbir x_i en yakın olduğu q_i nin kümesine atanır.
3. Kümelere ait $q_1, q_2, \dots q_k$ değerleri yeniden hesaplanır.
4. Küme elemanlarında herhangi değişiklik yoksa durulur.
5. İlk adımdan itibaren tekrar edilir.

Çıktılar:

k adet küme.

Bu algoritma, bahsedilen başlangıç noktalarının seçimine oldukça duyarlıdır ve büyük veri kümeleri için, bu başlangıç noktalarına bağlı olarak, global çözümden oldukça farklı yerel çözümler bulabilmektedir.

Adımlardan da görüldüğü gibi k-ortalama algoritmasının sonlanma kriteri iki defa üst üste aynı kümeleme yapısının bulunmasıdır. Bu algoritmanın zaman karmaşıklığı *O(tkn)* olarak hesaplanmıştır [Dunham, 2001]. Burada *t* iterasyon sayısını *k* küme sayısını ve *n* ise veritabanındaki veri sayısını temsil etmektedir. K-ortalama algoritması sadece sayısal verilerde kullanılabilir; kategorik verilerde kullanılamaz. Çünkü kategorik verilerde elde edilecek ortalamalar bir kümeyi diğer kümelerden ayıran anlamlı sayılar olmayacaktır. Bunun dışında, bu algoritma sadece dışbükey şeklindeki kümelerin tespit edilmesinde kullanılabilinir ve gürültülü ve uçtaki verilerden oldukça etkilenir.

10.2. Artımlı Kümeleme - Global k-Ortalamalar Algoritması (Global k-Means)

k-means algoritması global değil, lokal çözümler sunmaktadır ve algoritmanın sonuç değerleri başlangıçta rastgele olarak seçilen noktalardan ciddi ölçüde etkilenir. Bu sorunları çözmek için global optimizasyona dayalı çeşitli çalışmalar yapılmıştır. Genellikle bu çalışmalar çok popüler olmamış ve çözüm olarak yine kümeleme için k-means algoritması tekrar tekrar çalıştırılıp sonuçları öyle analiz edilmiştir.

2001 yılında Likas ve arkadaşları tarafından Global k-means algoritması adında bir kümeleme algoritması geliştirildi. Bu algoritma artımlı algoritmadır ve $k \leq m$ için aşağıdaki şekilde ifade edilir:

Adım 1 (Başlangıç): ilk iterasyonda $\boldsymbol{X}$ kümesinin Q_1 merkezi aşağıdaki şekilde hesaplanır:

$$Q_1 = \frac{1}{n}\sum_{i=1}^{n} x_i, \quad x_i \in \boldsymbol{X}, \ i = 1, \ldots, n \tag{10.2.1}$$

ve $u = 1$ alınır. Burada *n* veritabanındaki nesne sayısıdır.

Adım 2 (Durma Kriteri): $u = u + 1$. Eğer $u > k$ ise durulur.

Adım 3: $q_1, q_2, \dots, q_{l-1}$ merkezleri önceki iterasyondan alınır ve $\boldsymbol{X}$ kümesindeki herbir x elemanı önceki $u - 1$ sayıda q küme merkezleri ile beraber bir merkez noktası gibi ele alınır. Buna göre $(q_1, q_2, \dots, q_{u-1}, x)$ şeklinde, her biri u boyutlu, n adet (her bir veri noktası için bir başlangıç çözüm) başlangıç noktaları oluşturularak bunların her biri için k-means algoritması uygulanır ve elde edilen en iyi kümeleme ile (burada her k-means hesaplaması için oluşan amaç fonksiyon değerleri kontrol edilir) bu kümelemeye karşılık gelen sonuç merkezler saklanır $(y^1, y^2, \dots, y^u)$.

Adım 4: $q_i = y_i, \ \ i = 1, \dots, u$ ataması yapılır ve adım 2 ye gidilir.

Global k-means algoritması, global en iyiyi yada globale yakın çözümleri bulabilmesi sebebiyle, k-means algoritmasından pozitif anlamda farklılaşmaktadır. Gerçek hayat uygulamalarında, verideki doğal gruplamayı ortaya çıkarabilecek olan en uygun k değerinin bilinmesi, genellikle mümkün olamasa da, çalışılan alanda uzmanlaşmış kişilerin tecrübeleri bu değerin belirlenmesinde kullanılmaktadır. Fakat global k-means yönteminin doğası gereği, k değerinin baştan belirlenmesi gerekli değildir; çünkü artımlı bir algoritma söz konusudur. Bu algoritmada amaç, kümeleme problemlerine her aşamada en iyi olan, yeni bir küme merkezini eklemektir. Veri kümesinin k-bölünmesini hesaplamak için, $(k-1)$-kümeleme problemindeki, $k-1$ merkezli başlangıç durumundan yola çıkılarak, k. merkez için en uygun noktayı seçmek amaç edinilir [Bagirov ve Yearwood, 2006; Hansen vd., 2002].

11. KATEGORİK VERİLER ÜZERİNDE KÜMELEME

11.1. k-Modes

Çok sayıda veritabanı uygulamasında verinin homojen kümelere ayrılması gerekir. Bununla veri içinde ilgi çekecek gruplar tespit edilmiş olur. Bu gibi analizlerin yapılması için en azından iki problem çözülmüş olmalı: (1) verilerin etkili şekilde homojen gruplara ayrılması ve (2) kümelerin etkili şekilde yorumlanması. Bakılan kümeleme yöntemi birinci problemin çözümü için önerilmektedir.

İlk problem için çeşitli yöntemler kullanılmaktadır. Veri hakkında fazla bilginin olmadığı durumlarda sıklıkla kümeleme yöntemleri kullanılır. Diğer yandan veri madenciliğinde veriler hem kategorik hem nümerik özellik değerlerine sahip olmaktadır. Kategorik özelliklerin nümerik özellikler gibi işlenmesi her zaman mantıklı sonuçlar vermemektedir çünkü kategorik değerler sıralı değildir.

2001 senesinde Chaturvedi ve arkadaşları tarafından kategorik verileri kümelemek için k-means tabanlı k-modes algoritması geliştirilmiştir. Bu algoritma, kategorik verilerin benzerlik özelliği göz önüne alınarak hesaplama yapmaktadır. İlgili prosedür k-means kümeleme prosedürüne benzemektedir. Algoritmanın modeli aşağıdaki gibidir:

$$C_{MxN} = S_{MxK} \cdot W_{KxN} + error \qquad (11.1.1)$$

Burada C_{MxN} M sayıda N boyutlu veri matrisidir. S her bir M verisinin K kümesine ait olma matrisidir (0 ve 1 değerini almaktadır). W ise aday ve kesin merkez noktalar matrisidir.

Algoritma çalışma prensibi: ilk başta C veri matrisi verilmektedir. S ve W matrisleri baştan belli değildir ve hesaplama ile sırasıyla yer değiştirerek oluşturulurlar. C veri matrisi tamamen kategorik verilerden oluşmaktadır.

Problemin amacı L_P – norma dayalı amaç fonksiyonunu minimize etmektir (kısaca L_P– norm bazlı amaç fonksiyonu denilir).

$$L_P = \sum_{m=1}^{M}\sum_{n=1}^{N}|c_{mn} - \hat{c}_{mn}|^p \qquad (11.1.2)$$

Burada $\hat{c}_{mn}$; $\hat{C} = SW$ nin (m, n)-inci elemanıdır ve pozitif $p \to 0$ değerleri için S ve W yi bulmakta kullanılır. Sonlu $p \to 0$ koşulunda L_P – norma dayalı amaç fonksiyonu basitçe C ve $\hat{C}$ matrislerindeki uyumsuzlukları sayar. Matematiksel olarak L_P fonksiyonu norm sayı metriğini kullanır ve bu yaklaşımı kategorik verilerde de kullanmak mümkündür (kategorik verilerde sayma işlemi mümkün olduğu için).

S ve Wmatrisleri sırasıyla hesaplanır (S nin bilinmesiyle W ve W nin bilinmesiyle de yeni S değerleri bulunur). Bu işlem sırasıyla L_P fonksiyonunun artık iyileşmediği ana kadar devam eder.

İlk adımda verilen W değerlerine göre S aşağıdaki gibi bulunmaktadır:

$$C = \begin{bmatrix} 1 & 5 & 0 & 3 \\ 2 & 6 & 1 & 3 \\ 3 & 6 & 0 & 3 \\ 2 & 7 & 0 & 4 \end{bmatrix}, \quad S = \begin{bmatrix} s_{11} & s_{12} \\ s_{21} & s_{22} \\ s_{31} & s_{32} \\ s_{41} & s_{42} \end{bmatrix} \text{ ve } \quad W = \begin{bmatrix} 2 & 6 & 1 & 3 \\ 1 & 5 & 0 & 4 \end{bmatrix}.$$

W matrisinin (i, j) elemanı i kümesinin j kategorik özellik değerlerinin merkezine karşılık gelir. Burada amaç S ye dayalı en iyi L_P fonksiyon değerine ulaşmaktır. Burada $C = SW + error$ ve S; 0 veya 1 değerlerini almaktadır.

k-means, k-medoids ve buna benzer algoritmalar gibi bu algoritma da lokal optimal çözümleri üretmektedir.

12. KARMA VERİLER ÜZERİNDE KÜMELEME

12.1. k-Prototypes

$\boldsymbol{X} = \{X_1, X_2, \ldots, X_n\}$, n adet nesneyi temsil eden küme olsun ve $X_i = [x_{i1}, x_{i2}, \ldots, x_{im}]$ bu kümenin *m* özellikli nesnesi olsun. *k* pozitif tam sayı olsun. $\boldsymbol{X}$ 'ı kümelemenin amacı, $\boldsymbol{X}$ 'ın nesnelerini, *k* sayıda kümelere ayıran bölünmeyi bulmaktır.

Verilen *n* için, mümkün bölümlemelerin sayısı sonlu fakat oldukça çoktur. Her bölümlemeyi incelemek ve bunların içinden en iyisini seçmek oldukça elverişsizdir. Bilindik çözüm, uygun bölümlemeyi bulmaya yardımcı olacak bir kümeleme kriteri belirlemek olacaktır. Kümeleme kriteri aşağıdaki gibi *amaç fonksiyonu (cost function)* olarak adlandırılır.

Amaç Fonksiyonu

Sık kullanılan amaç fonksiyonu küme içi dağılım matrisi şeklindedir. Bu fonksiyonu aşağıdaki şekilde tanımlamak mümkündür:

$$E = \sum_{l=1}^{k} \sum_{i=1}^{n} y_{il} d(X_i, Q_l) \qquad (12.1.1)$$

Burada, $Q_l = [q_{l1}, q_{l2}, \ldots, q_{lm}]$, l kümesinin *temsili vektörü* veya *prototipidir (merkezidir)* ve y_{il} , Y_{nxl} ait olma matrisinin elemanıdır. *d* benzerlik ölçeğidir ve genellikle öklid mesafesi olarak da tanımlanır. *Y* 'nin iki özelliği var: (1) $0 \leq y_{il} \leq 1$ ve (2) $\sum_{l=1}^{k} y_{il} = 1$. Burada *Y*, eğer $y_{il} \in \{0,1\}$ ise, *kesin kümeleme* olarak adlanır. Değilse *bulanık kümeleme* olmuş olacaktır. Kesin kümelemede, $y_{il} = 1$ eşitliği, ilgili X_i verisinin *l* kümesine *Y* ye göre ait olduğunu gösterir. Bu algoritmada *kesin kümeleme* bakılmıştır. (12.1.1) denklemindeki $E_l = \sum_{i=1}^{n} y_{il} d(X_i, Q_l)$ terimi, $\boldsymbol{X}$ 'ı l kümesine atanmasının toplam maliyetidir, yani, l kümesindeki nesnelerin onların Q_l prototipine toplam dağılımıdır. Aşağıdaki koşul sağlanırsa E_l minimize olur:

$$q_{lj} = \frac{1}{n_l}\sum_{i=1}^{n} y_{il}x_{ij}\,, \qquad j = 1, \dots, m \quad \text{için} \qquad (12.1.2)$$

Burada, $n_l = \sum_{i=1}^{n} y_{il}$, l kümesindeki nesne sayısıdır.

$\boldsymbol{X}$ 'da kategorik özellikler olduğunda, benzerlik denklemini aşağıdaki gibi yazabiliriz:

$$d(X_i, Q_l) = \sum_{j=1}^{m_r}\left(x_{ij}^r - q_{lj}^r\right)^2 + \gamma_l \sum_{j=1}^{m_c}\delta\left(x_{ij}^c, q_{lj}^c\right) \qquad (12.1.3)$$

Burada, $\delta(p,q) = 0$, $p = q$ için ve $\delta(p,q) = 1$, $p \neq q$ için. x_{ij}^r ve q_{lj}^r nümerik özellik değerleridir. x_{ij}^c ve q_{lj}^c ise i nesnesinin ve küme merkezinin kategorik değerleridir. γ_l , l kümesi için kategorik özelliklerin ağırlık değeridir.

E_l 'i aşağıdaki şekilde yazabiliriz.

$$E_l = \sum_{i=1}^{n} y_{il}\sum_{j=1}^{m_r}\left(x_{ij}^r - q_{lj}^r\right)^2 + \gamma_l \sum_{i=1}^{n} y_{il}\sum_{j=1}^{m_c}\delta\left(x_{ij}^c, q_{lj}^c\right) = E_l^r + E_l^c \qquad (12.1.4)$$

Burada, E_l^r , l kümesindeki tüm nümerik özelliklerin toplam maliyet değeridir. Eğer q_{lj}^r , (12.1.2) denklemine göre hesaplanırsa, E_l^r minimize olur.

C_j nin , j özelliği için tüm tekrar etmeyen değerlerinin kümesi olduğunu kabul edelim ve $p\left(c_j \in C_j | l\right)$ ise, c_j değerinin l kümesinde rastlanma olasılığı olsun. (12.1.4) deki E_l^c ni aşağıdaki gibi yazabiliriz:

$$E_l^c = \gamma_l \sum_{j=1}^{m_c} n_l \left(1 - p\left(c_j \in C_j | l\right)\right) \qquad (12.1.5)$$

Burada, n_l , l kümesindeki nesne sayısıdır. E_l^c ni minimize etmek için, çözüm olarak aşağıdaki Lemma verilmiştir:

Lemma 1: Sadece ve sadece tüm kategorik özellikler için $q_{lj}^{c} \neq c_j$ olduğunda, $p\left(q_{lj}^{c} \in C_j | l\right) \geq p\left(c_j \in C_j | l\right)$ olursa verilen l kümesi için, E_l^c minimize olur.

Son olarak E'ni aşağıdaki şekilde yazabiliriz:

$$E = \sum_{l=1}^{k} (E_l^r + E_l^c) = \sum_{l=1}^{k} E_l^r + \sum_{l=1}^{k} E_l^c = E^r + E^c \qquad (12.1.6)$$

(12.1.6) eşitliliği nümerik ve kategorik değerleri olan veri setinin kümelemesindeki amaç fonksiyonudur. E^r ve E^c negatif değerler olmadığı için E nin minimize edilmesi için E^r ve E^c nümerik ve kategorik özelliklerin tüm veri setinde minimize edilmesi gerekir. E^r değeri (12.1.2) denklemi ile k küme prototipinin nümerik değerlerinin hesaplanması ile minimize olur. E^c ise k kümesinin kategorik elemanlarının prototipinin Lemma 1'e göre seçilmesi ile minimize edilebilir. Başka bir deyişle, (12.1.2) denklemi ve Lemma 1, (12.1.5) amaç fonksiyonunun minimize edilmesi için küme prototiplerinin seçilmesine yardımcı olur. Buna *k-prototypes* algoritmasının temeli diyebiliriz.

Benzerlik ölçeği

(12.1.3) denkleminde tanımlanan (12.1.6) amaç fonksiyonu, veri nesneleri ve küme merkezleri arasında nümerik ve kategorik özellikler için ortak uzaklık ölçeğini verir. Nümerik özellikler için benzerlik ölçeği Öklid mesafesidir. Kategorik özellikler için ise veri nesneleri ve küme prototipleri arasındaki kategorik özelliklerdeki eşitsizliklerinin sayısıdır söyleyebiliriz. γ_l ağırlık değeri, özelliklerin önemini ayarlamak için kullanılmaktadır.

Eğer $\gamma_l = 0$ ise o zaman kümeleme sadece nümerik verilere bağlı olacaktır. Eğer $\gamma_l > 0$ ise artık kategorik veriler de sonuçlara etki etmeye başlayacaktır. γ_l nin seçimi nümerik özelliklerin dağılımına bağlıdır. Genel olarak söylemek gerekirse γ_l nin seçimi nümerik özelliklerin l kümesindeki ortalama standart dağılımı olan σ_l ye bağlıdır. Pratikte σ_l, γ_l değerini bulmak için kullanılır. σ_l değeri kümeleme öncesi kümeler daha belirlenmediği için bilinmez olduğundan

tüm nümerik özelliklerin toplam ortalama standart dağılımı olan σ değerini tüm σ_l ler için kullanmak mümkündür. Artımlı algoritmalarda σ_l değeri her iterasyon için bir önceki iterasyonla hesaplanabilinir.

k-prototypes **Algoritması**

Algoritmada 3 ana fonksiyon çalışmaktadır: (1) *İlk prototiplerin seçilmesi*, (2) *ilk atama* ve (3) *yeniden atama* işlemleri. Dikkat edilmesi gereken detay, ilk prototiplerin seçilmesi aşamasında seçilen *k* sayıda rastgele prototip veri seti nesnelerinin içinden seçiliyor olmasıdır.

k-prototypes algoritması aşağıdaki adımlarla gösterilebilir:

(1) $\boldsymbol{X}$ veri setinden *k* sayıda, her biri bir küme için, başlangıç prototip seç.

(2) $\boldsymbol{X}$ veri setindeki her bir nesneyi (12.1.3) denklemine göre prototipine en yakın olan kümeye ata. Her atamadan sonra o kümenin prototipini güncelle.

(3) Tüm nesneler kümelere atandıktan sonra, oluşmuş sonuç prototiplere göre nesnelerin benzerliklerini bir daha hesapla. Bu aşamada, eğer nesnenin o anki prototipinden daha yakın bir prototip olursa nesneyi yeni prototipe ata ve her iki prototipi son duruma göre yeniden güncelle.

(4) Adım (3)'ü tüm $\boldsymbol{X}$ veri setinden geçildiğinde hiçbir nesne yer değişmediği ana kadar tekrarla.

13. KARMA VERİLER ÜZERİNDE ARTIMLI KÜMELEME ALGORİTMASI

k-means algoritmasının lokal çözümlerine karşılık olarak geliştirilen global k-means algoritmasından yukarıda bahsedildi. Bu yaklaşımın global çözüm veya global çözüme yakın sonuçlar çıkardığı gözlemlenmektedir. Bununla beraber, bu yöntem sadece nümerik verilerde çalışan k-means algoritmasının geliştirilmiş halidir. k-means algoritmasının kategorik verilerle de çalışmasını sağlayan k-prototypes algoritması da yine lokal çözümler üretmekte ve başlangıçta seçilen rastgele k merkez prototipine oldukça bağlıdır.

k-prototypes algoritmasının karma verilerle çalışması avantajını ve global k-means algoritmasının global veya globale yakın çözümler üretme yeteneğini göz önünde bulundurup, hem nümerik, hem kategorik verilerle çalışan ve artımlı olup başlangıç rastgele seçilmiş merkezlerine bağımlı olmayan bir algoritma geliştirildi. Bu algoritma veri kümeleme ihtiyaçlarında sıklıkla rastlanan karma veriler için etkin çözüm bulmaktadır.

13.1. Artımlı Global k-Prototypes Algoritması

$\boldsymbol{X} = \{X_1, X_2, \ldots, X_n\}$, n adet nesneyi temsil eden küme olsun ve $X_i = [x_{i1}, x_{i2}, \ldots, x_{im}]$ bu kümenin m özellikli nesnesi olsun. k pozitif tam sayı olsun. Artımlı kümelemede de $\boldsymbol{X}$ 'ı kümelemenin amacı, $\boldsymbol{X}$ 'ın nesnelerini , k sayıda kümelere ayıran bölünmeyi bulmaktır. Amaç fonkiyonu

$$E = \sum_{l=1}^{k} \sum_{i=1}^{n} y_{il} d(X_i, Q_l) \qquad (13.1.1)$$

şeklindedir. Burada, y_{il} , Y_{nxl} ait olma matrisinin elemanıdır. $d(X_i, Q_l)$, bakılan X_i veri noktasının ait olduğu l kümesinin Q_l merkez-prototipine uzaklığıdır. Verilerin nümerik özellikleri için uzaklık Öklid mesafesi olarak, kategorik uzaklığı için aynı değer olup olmaması ile ölçülmektedir. Uzaklık ölçeğini aşağıdakı gibi yazabiliriz.

$$d(X_i, Q_l) = \sum_{j=1}^{m_r} \left(x_{ij}^r - q_{lj}^r\right)^2 + \sum_{j=1}^{m_c} \gamma_{lj} \cdot \delta\left(x_{ij}^c, q_{lj}^c\right) \qquad (13.1.2)$$

Burada x_{ij}^r ve q_{lj}^r bakılan veri ve merkez noktasının nümerik özelliklerini, x_{ij}^c ve q_{lj}^c ise kategorik özelliklerini göstermektedir.

Kategorik verilerin benzerlik ölçeği $\delta(p, q) = 1$, $p = q$ için ve $\delta(p, q) = 0$, $p \neq q$ için olacaktır. Aynı kategorik veri özellikleri arasında mesafe 0 ve farklılar için 1 olacaktır.

k-prototypes algoritmasından farklı olarak bu algoritmada l kümesi için kategorik verilerin γ_l ağırlık değeri kullanılmamıştır. Bunun yerine verideki nümerik özellik değerlerini [0,1] arasında normalleştirip kategorik verilerin benzerliği ile aynı ölçeğe dönüştürüp, bu şekilde hesaplama yapılmıştır.

Burada önemli bir detay, eğer kategorik özelliklerden biri diğerlerine göre daha etkin ise o kategorik özellik için γ_{lj} ağırlığı kullanılabilinir. Kategorik özellikler için varsayılan ağırlık değeri 1 olacaktır. Ağırlık ayarını tüm kategorik özellikler için tek değer kullanılmayıp, kategorik özellik bazında değişmek mümkün. Bununla, özelliğin önemine göre farklı ağırlık katsayıları kullanılabilir.

E^r ve E^c ni ayrı ayrı verilerdeki nümerik ve kategorik özellikler için ayrı ayrı amaç fonksiyonları gibi söyleyebiliriz.

$$E = E^r + E^c \qquad (13.1.3)$$

Amaç fonksiyonunu minimize etmek için E^r ve E^c nümerik ve kategorik özelliklerin tüm veri setinde minimize edilmesi gerekir. E^r değerinin minimize edilmesi için nümerik verilerde olabildiğince verilerin kendi merkez noktalarına uzaklıkları toplamı minimize edilmelidir. E^c kategorik özellikler için ise her bir kategorik özellik için en çok rastlanan değerlerin merkez noktasındaki o özelliğe atanması ile elde edilebilir.

Artımlı k-prototypes algoritmasındaki önemli kısım verinin başlangıçta hemen *k* sayıda kümeye atanmayıp sadece *k*=1 şeklinde tanımla tüm verisetinin ortasını almaktan başlamaktadır. Daha sonrasında her iterasyonla beraber en uygun sıradaki aday prototip belirlenip, kümelemenin verilen k küme sayısına ulaşıncaya kadar global çözümleri bularak ilerlemesi şeklindedir. Algoritma adımlarını aşağıdaki şekilde sıralayabiliriz:

Algoritma:

Adım 1 (Başlangıç): Tüm verisetinin ağırlıklı merkezi bulunur. Bunun için q_1 merkezi hesaplanır. Bu merkezin nümerik özellikleri q_{1j}^r ile, kategorik özellikleri ise q_{1j}^c ile gösterilmektedir. Bu merkez için nümerik özellik değeri veri setindeki tüm verilerin o özellik değerlerinin ortalaması, yani mean'ı seçilir. Kategorik özellikler için en çok rastlanan kategorik özellik değeri, yani mod'u seçilmektedir. C_j ni, j özelliği için tüm tekrar etmeyen değerlerinin kümesi olduğunu kabul edersek, $p\left(c_j \in C_j | l\right)$, c_j değerinin l kümesinde rastlanma olasılığı olacaktır. Böylece, q_{1j} ilk merkez noktası aşağıdaki gibi gösterilebilir:

$$q_{1j}\left(q_{1j}^r \middle|\, q_{1j}^c\right) = \left(\frac{1}{n}\sum_{i=1}^{n} x_{1j}^r \;\middle|\; \max\left(p\left(c_j \in C_j | l\right)\right)\right) \quad x_i \in X,\ \ i = 1, \dots, n \tag{13.1.4}$$

ve $u = 1$ alınır. Burada n, X kümesindeki eleman sayısıdır.

Adım 2 (Durma Kriteri): $u = u + 1$. Eğer $u > k$ ise durdurulur.

Adım 3: $q_1, q_2, \dots, q_{u-1}$ merkezleri önceki iterasyondan alınır. ve X kümesindeki herbir x elemanı u'ncu küme merkezi için bir başlangıç noktası olarak ele alınır. Buna göre $(q_1, \dots, q_{u-1}, x)$ şeklinde u boyutlu, n adet başlangıç çözümleri oluşturularak her bir merkez noktasının çözümü için amaç fonksiyonu hesaplanır. Elde edilen en iyi amaç fonksiyon değerini veren çözüm noktaları ile *k-prototypes* algoritması uygulanır ve bu kümelemenin sonuç çıktı merkezleri saklanır $(y_1, y_2, \dots, y_u)$.

Adım 4: $q_i = y_i,\ \ i = 1, \dots, u$ ataması yapılır ve adım 2 ye gidilir.

14. HESAPLAMA DENEMELERİ – KIYASLAMALAR

Geliştirilen algoritmanın testleri için 16 veri seti üzerinde k-prototypes algoritması ile, tezde geliştirilen artımlı k-prototypes algoritması kıyaslanarak hesaplamalar yapıldı. k-prototypes algoritması başlangıçta aldığı k sayıda merkez prototipini rastgele aldığından algoritmanın sonuçları bu değerlere bağımlıdır ve böylece lokal minimumlar almaktadır. Bu farkı kapatmak için ele alınan veri setleri üzerinde k-prototypes algoritması her bir *k* ve *gamma* değerleri için 10 defa çalıştırılıp bu çalışmalarda bulunan minimum ve ortalama amaç fonksiyon değerleri ele alındı. Diğer yandan veriler üzerinde aynı *k* ve *gamma* değerleri ile artımlı k-prototypes algoritması çalıştırılarak sonuçlar hesaplandı ve bu iki algoritmanın farkı tablolarla verildi. Tüm veri setleri için hesaplamalar geniş alan kapladığı için sadece tek bir veri seti örneği verilmiştir.

Hesaplamalarda *k* değerine 2 den 50 ye kadar değerler verildi. Bazı durumlarda daha küçük *k* değerinde işlem durduruldu (sunulan verisetinden başka birkaç veri setinde), çünkü veri sayısından ve gamma değerinden dolayı fazla küme olması artık anlamsızlaşmaya başladı.

Kategorik verilerin önem katsayısı olarak söyleyebileceğimiz γ değeri 0.5 , 0.7 ve 1.0 alındı.

Tablo 14.1: Hesaplama sonuçları tablolarındaki kolonların açıklamaları.

k (Küme Sayısı)	***k***
k-prototypes ortalama çalışma süresi	***k-p. avg(t)***
k-prototypes minimum amaç fonksiyon değeri	***k-p. min($F_{amaç}$)***
k-prototypes ortalama amaç fonksiyon değeri	***k-p. avg($F_{amaç}$)***
Artımlı k-prototypes çalışma süresi	***G. k-p. t***
Artımlı k-prototypes amaç fonksiyon değeri	***G. k-p. $F_{amaç}$***
k-prototypes minimum değeri ile Artımlı k-prototypes değeri farkı	***G. k-p. $F_{amaç}$ - k-p. min($F_{amaç}$)***
k-prototypes ortalama değeri ile Artımlı k-prototypes değeri farkı	***G. k-p. $F_{amaç}$ - k-p. avg($F_{amaç}$)***

Tablo 14.2: Veri setleri detayları tablosu (içerdiği veri sayısına göre sıralanmıştır):

İsmi	Özellik Türleri	Veri Sayısı	Özellik Sayısı	Nümerik	Kategorik	ID Kolonu
Post-Operative Patient	Kategorik , Tamsayı	**90**	8	1	8	0
Zoo	Kategorik , Tamsayı	**101**	17	1	33	1
Acute Inflammations	Kategorik , Tamsayı	**120**	6	1	7	0
Japanese Credit Screening	Kategorik , Reel, Tamsayı	**125**	16	6	9	1
Teaching Assistant Evaluation	Kategorik , Tamsayı	**151**	5	1	5	0
Servo	Kategorik , Tamsayı	**167**	4	4	1	0
Flags	Kategorik , Tamsayı	**194**	30	2	27	1
Auto imports 85	Kategorik , Tamsayı , Reel	**205**	26	14	11	0
Statlog (Heart)	Kategorik , Real	**270**	13	5	8	1
Liver Disorders	Kategorik , Tamsayı , Reel	**345**	7	6	1	0
Dermatology	Kategorik , Tamsayı	**366**	33	33	1	1
Auto MPG	Kategorik , Reel	**398**	8	5	3	1
Meta-data	Kategorik , Tamsayı, Reel	**528**	22	20	2	1
Statlog (Australian Credit Approval)	Kategorik , Tamsayı , Reel	**690**	14	6	8	1
Statlog (German Credit Data)	Kategorik , Tamsayı	**1000**	20	7	13	0
Contraceptive Method Choice	Kategorik , Tamsayı	**1473**	10	2	8	0

Tablo 14.3.1 Hesaplama parametreleri (Örnek 1.1)

Veriseti	K_MEANS_AUTO_IMPORTS
Gamma Değeri	**0.5**
Nümerik Özellik Sayısı	14
Kategorik Özellik Sayısı	11

Tablo 14.3.2 Hesaplama sonuçları detayları (Örnek 1.2)

k	*k-p. avg(t)*	*k-p. min(Famaç)*	*k-p. avg(Famaç)*	*G. k-p. t*	*G. k-p. Famaç*	*G. k-p. Famaç - k-p. min(Famaç)*	*G. k-p. Famaç - k-p. avg(Famaç)*
2	**00:00:00.7885060**	434.9547612	453.4110623	**00:00:01.1033999**	439.6343905	**-4.679629273**	**13.77667186**
3	**00:00:00.8383530**	395.3820429	406.7413951	**00:00:01.0345902**	399.6625725	**-4.280529521**	**7.078822684**
4	**00:00:00.8311520**	359.5707743	383.81318	**00:00:01.2707135**	371.145944	**-11.57516969**	**12.66723592**
5	**00:00:00.8263280**	343.6529851	362.6002258	**00:00:01.6653553**	344.0406473	**-0.387662153**	**18.55957849**
6	**00:00:00.7917120**	335.2653695	346.0948645	**00:00:02.1748522**	325.8926022	**9.372767325**	**20.20226229**
7	**00:00:00.7645600**	308.5598375	327.0861781	**00:00:02.8214904**	309.4888694	**-0.929031883**	**17.59730867**
8	**00:00:00.7984850**	300.796381	311.3582589	**00:00:03.5281578**	295.6407355	**5.155645527**	**15.71752344**
9	**00:00:00.7699330**	291.6847625	300.5501104	**00:00:04.1711683**	280.5097881	**11.17497442**	**20.04032235**
10	**00:00:00.8734870**	282.4807599	294.0632008	**00:00:04.9291508**	270.2848853	**12.19587462**	**23.77831548**
11	**00:00:00.7887900**	276.7737647	285.5482064	**00:00:05.7612686**	259.4375231	**17.33624155**	**26.11068331**
12	**00:00:00.8096640**	271.9713008	284.222479	**00:00:06.8600416**	251.5167717	**20.45452918**	**32.70570735**
13	**00:00:00.8038560**	255.388271	267.1581534	**00:00:07.9229820**	240.5617595	**14.82651156**	**26.5963939**
14	**00:00:00.8406920**	250.3198648	263.5912895	**00:00:08.9211292**	233.2609005	**17.05896424**	**30.33038896**
15	**00:00:00.8287920**	244.8012182	257.4236706	**00:00:10.1261795**	226.6571621	**18.14405607**	**30.7665085**
16	**00:00:00.8495800**	231.9399408	246.650725	**00:00:11.4187243**	218.2167004	**13.72324037**	**28.43402461**
17	**00:00:00.8388870**	236.9233845	249.5131479	**00:00:12.8084904**	211.8891216	**25.03426296**	**37.62402634**
18	**00:00:00.8327760**	229.26752	247.3633109	**00:00:14.2574847**	206.1288696	**23.13865038**	**41.23444136**
19	**00:00:00.8620760**	217.2128276	229.9093461	**00:00:15.8029579**	200.2084147	**17.00441289**	**29.70093145**
20	**00:00:00.8880040**	220.538321	232.063573	**00:00:17.4506992**	193.1835367	**27.35478425**	**38.88003628**
21	**00:00:00.8721620**	207.3387724	221.2341468	**00:00:19.0176202**	187.7121107	**19.62666166**	**33.52203611**
22	**00:00:00.8819040**	204.834666	221.5596895	**00:00:20.7589895**	181.9313004	**22.90336551**	**39.62838908**
23	**00:00:00.9137050**	202.9579053	216.6854627	**00:00:22.5764749**	177.0041506	**25.95375471**	**39.68131215**
24	**00:00:00.9138550**	194.7361284	208.8918135	**00:00:24.5080448**	172.9604677	**21.7756607**	**35.93134581**
25	**00:00:00.9683110**	188.8966056	207.0309016	**00:00:26.6294242**	167.4889086	**21.40769698**	**39.54199301**
26	**00:00:00.9483490**	189.3596842	206.8984398	**00:00:28.5956346**	163.9126648	**25.44701937**	**42.98577502**
27	**00:00:00.9658700**	196.1169778	201.6170102	**00:00:30.6437463**	160.3056379	**35.81133986**	**41.31137234**
28	**00:00:00.9861730**	180.6857545	198.0968002	**00:00:32.8831068**	156.9093144	**23.77644009**	**41.18748584**
29	**00:00:01.0130110**	179.3011492	193.8211037	**00:00:35.1773890**	153.1041254	**26.19702377**	**40.71697822**
30	**00:00:01.0463040**	166.2877181	187.6144471	**00:00:37.5685371**	149.7644826	**16.52323551**	**37.84996458**
31	**00:00:01.0382610**	176.9357389	191.4700936	**00:00:40.1873220**	145.8048591	**31.13087981**	**45.66523451**
32	**00:00:01.0889760**	168.9551637	181.9950331	**00:00:42.5191068**	142.2955813	**26.65958237**	**39.6994518**
33	**00:00:01.0902770**	173.615984	184.6423393	**00:00:45.1227003**	139.1186965	**34.49728747**	**45.52364277**
34	**00:00:01.1051600**	161.2316651	175.7402798	**00:00:49.0117906**	135.8746122	**25.35705289**	**39.86566752**
35	**00:00:01.1553800**	164.4060541	177.8383054	**00:00:51.6048697**	132.8800611	**31.52599297**	**44.95824434**
36	**00:00:01.1057960**	157.6497223	170.310302	**00:00:53.6776688**	130.5084685	**27.14125387**	**39.80183359**
37	**00:00:01.1232140**	151.2588727	166.0613469	**00:00:56.3765567**	128.0612791	**23.19759367**	**38.00006786**
38	**00:00:01.1306180**	155.2330887	169.799125	**00:01:00.3437571**	125.418005	**29.81508364**	**44.38111994**
39	**00:00:01.1685540**	150.822403	160.7317699	**00:01:04.6239256**	122.6910736	**28.13132941**	**38.04069634**
40	**00:00:01.2352330**	145.0166441	159.5674825	**00:01:05.6610448**	120.1513363	**24.86530775**	**39.41614614**
41	**00:00:01.2124630**	143.3898392	158.6378893	**00:01:08.8977470**	117.7954842	**25.59435507**	**40.84240513**
42	**00:00:01.2524430**	148.6207419	161.4371922	**00:01:12.2453264**	115.2658925	**33.35484938**	**46.17129963**
43	**00:00:01.2644460**	144.7456452	152.111393	**00:01:15.6010825**	112.9916988	**31.75394636**	**39.11969419**
44	**00:00:01.2802650**	136.101842	153.4966659	**00:01:18.9304192**	110.6241734	**25.47766857**	**42.8724925**
45	**00:00:01.3342800**	135.1521096	151.9664697	**00:01:22.8987854**	108.5653792	**26.58673045**	**43.40109054**
46	**00:00:01.3195060**	142.2501118	154.2635343	**00:01:26.2528385**	106.6962293	**35.55388249**	**47.56730496**
47	**00:00:01.3715320**	127.6220094	147.0478343	**00:01:30.0376227**	104.1146367	**23.50737267**	**42.93319759**
48	**00:00:01.3752560**	133.3560917	142.9847807	**00:01:33.7119029**	102.3254105	**31.0306812**	**40.65937021**
49	**00:00:01.3977690**	134.7056198	142.2561799	**00:01:37.5410062**	100.5723623	**34.13325756**	**41.68381761**
50	**00:00:01.4808820**	127.9556163	142.2683258	**00:01:41.4502065**	98.77544471	**29.18017158**	**43.49288109**

Tablo 14.3.3 Hesaplama parametreleri (Örnek 2.1)

Veriseti	K_MEANS_AUTO_IMPORTS
Gamma Değeri	**0.7**
Nümerik Özellik Sayısı	14
Kategorik Özellik Sayısı	11

Tablo 14.3.4 Hesaplama parametreleri (Örnek 2.2)

k	*k-p. avg(t)*	*k-p. min(Famaç)*	*k-p. avg(Famaç)*	*G. k-p. t*	*G. k-p. Famaç*	*G. k-p. Famaç - k-p. min(Famaç)*	*G. k-p. Famaç - k-p. avg(Famaç)*
2	**00:00:00.8710430**	558.2952211	593.5091907	**00:00:00.9388943**	569.2343905	**-10.93916938**	**24.27480024**
3	**00:00:00.8010990**	513.9751877	547.9818041	**00:00:01.0086797**	516.6625725	**-2.687384738**	**31.31923162**
4	**00:00:00.7766320**	479.9054449	503.0001924	**00:00:01.2570990**	480.9123899	**-1.006944978**	**22.08780256**
5	**00:00:00.7623090**	454.6871316	467.839157	**00:00:01.7275012**	445.0831068	**9.604024757**	**22.75605022**
6	**00:00:00.7581660**	423.7815418	440.5192082	**00:00:02.3088554**	420.9399342	**2.841607579**	**19.57927397**
7	**00:00:00.7703330**	404.7984208	423.1006366	**00:00:02.7928649**	397.7483028	**7.050117956**	**25.35233382**
8	**00:00:00.8029540**	387.0323639	417.2890557	**00:00:03.6081606**	380.4407355	**6.591628408**	**36.84832017**
9	**00:00:00.7730180**	374.5875369	400.5861609	**00:00:04.2959967**	360.5657822	**14.02175472**	**40.02037865**
10	**00:00:00.7814510**	353.0208028	378.8941245	**00:00:04.8461246**	346.4848853	**6.53591751**	**32.40923921**
11	**00:00:00.7913310**	346.3535412	363.0037067	**00:00:05.7042573**	331.8375231	**14.51601811**	**31.1661836**
12	**00:00:00.8047580**	340.685544	355.4667858	**00:00:06.7625497**	321.1167717	**19.5687723**	**34.35001413**
13	**00:00:00.8156600**	328.0198294	353.3229087	**00:00:07.7434141**	307.7330182	**20.28681127**	**45.58989056**
14	**00:00:00.8304730**	319.4025506	334.3778771	**00:00:08.9118174**	296.3603792	**23.04217143**	**38.01749792**
15	**00:00:00.8613100**	319.6693209	336.2637509	**00:00:09.9850391**	286.6108965	**33.05842436**	**49.65285437**
16	**00:00:00.8430850**	306.9923821	325.4322306	**00:00:11.3325303**	278.4167004	**28.57568164**	**47.01553014**
17	**00:00:00.8432030**	291.6008467	310.2448156	**00:00:12.6738280**	270.2891216	**21.31172513**	**39.95569405**
18	**00:00:00.8368880**	284.5009965	298.6275827	**00:00:14.1670141**	262.7288696	**21.77212689**	**35.89871315**
19	**00:00:00.8503690**	282.1218266	297.5724622	**00:00:15.8481960**	255.0725521	**27.04927449**	**42.49991011**
20	**00:00:00.9320400**	272.9879283	289.9564398	**00:00:17.2998370**	247.5628094	**25.42511887**	**42.39363042**
21	**00:00:00.8699550**	274.2299499	285.1311207	**00:00:18.8630258**	238.1138185	**36.11613138**	**47.01730213**
22	**00:00:00.9058510**	261.1641803	281.478305	**00:00:20.6857225**	231.5650274	**29.59915288**	**49.91327767**
23	**00:00:00.8977450**	239.5188055	270.3589809	**00:00:22.3443427**	224.4041506	**15.1146549**	**45.95483035**
24	**00:00:00.9098480**	241.5945239	268.644842	**00:00:24.3583678**	218.7604677	**22.83405619**	**49.88437425**
25	**00:00:00.9752860**	244.7928918	267.2922019	**00:00:26.2452135**	211.0889086	**33.70398321**	**56.20329335**
26	**00:00:00.9447600**	238.0998201	261.7684529	**00:00:28.2501773**	206.3126648	**31.78715526**	**55.45578812**
27	**00:00:00.9875420**	234.2482261	252.9876007	**00:00:30.3798235**	201.7056379	**32.54258817**	**51.28196278**
28	**00:00:00.9926280**	218.6131287	248.7493835	**00:00:32.6075152**	197.1325525	**21.48057626**	**51.61683101**
29	**00:00:01.0293260**	233.621659	247.6807078	**00:00:34.9905052**	192.632527	**40.98913203**	**55.04818081**
30	**00:00:01.0629480**	234.932039	245.86558	**00:00:37.3358629**	188.0830111	**46.84902792**	**57.78256895**
31	**00:00:01.0355640**	206.7145924	238.554584	**00:00:39.6955317**	183.5810083	**23.13358407**	**54.97357572**
32	**00:00:01.0491980**	220.2030733	234.3819435	**00:00:42.1670968**	179.589831	**40.61324223**	**54.79211245**
33	**00:00:01.0709400**	209.0095056	230.546035	**00:00:44.8230106**	174.97035	**34.03915558**	**55.57568495**
34	**00:00:01.0754070**	216.8433361	226.6257883	**00:00:47.5273555**	170.9495089	**45.89382727**	**55.67627947**
35	**00:00:01.1511350**	212.024346	223.4186169	**00:00:50.2234880**	167.340709	**44.68363702**	**56.07790791**
36	**00:00:01.1055420**	197.7071044	218.5715331	**00:00:53.3645547**	163.8787682	**33.82833615**	**54.69276482**
37	**00:00:01.1244800**	195.179273	213.7000093	**00:00:56.0320129**	160.5688409	**34.6104321**	**53.1311684**
38	**00:00:01.1460960**	197.3637689	214.7215609	**00:01:00.4105227**	158.3035594	**39.06020954**	**56.4180015**
39	**00:00:01.1654220**	194.8656624	209.0062164	**00:01:02.1180432**	155.1229052	**39.74275721**	**53.88331119**
40	**00:00:01.2220280**	190.6752208	210.3664502	**00:01:05.2950779**	151.8796312	**38.79558964**	**58.48681907**
41	**00:00:01.2145710**	176.7709457	198.1543386	**00:01:08.5426061**	148.5526997	**28.21824598**	**49.60163886**
42	**00:00:01.2581910**	189.4489251	202.8052046	**00:01:11.9483311**	145.4775529	**43.97137221**	**57.32765171**
43	**00:00:01.2460760**	179.8752548	194.1605727	**00:01:16.3095225**	142.4375458	**37.43770895**	**51.72302692**
44	**00:00:01.2804580**	181.4906691	194.2056058	**00:01:18.3137461**	139.7890133	**41.70165586**	**54.41659253**
45	**00:00:01.3639290**	176.4322233	188.5629913	**00:01:21.7348437**	137.3619121	**39.07031119**	**51.20107926**
46	**00:00:01.3555860**	171.0556849	189.812335	**00:01:25.6436081**	134.6552699	**36.40041496**	**55.15706514**
47	**00:00:01.3674120**	168.1317889	187.183807	**00:01:29.3322864**	132.2675454	**35.86424356**	**54.91626158**
48	**00:00:01.4002390**	169.6065349	188.4440463	**00:01:32.9655830**	129.9728617	**39.63367325**	**58.47118459**
49	**00:00:01.4239950**	170.5089438	186.4509698	**00:01:36.8020433**	127.0524772	**43.4564666**	**59.39849253**
50	**00:00:01.4929790**	160.8488135	179.1417431	**00:01:40.8362622**	124.899429	**35.94938446**	**54.24231406**

Tablo 14.3.5 Hesaplama parametreleri (Örnek 3.1)

Veriseti	K_MEANS_AUTO_IMPORTS
Gamma Değeri	**1.0**
Nümerik Özellik Sayısı	14
Kategorik Özellik Sayısı	11

Tablo 14.3.6 Hesaplama parametreleri (Örnek 3.2)

k	*k-p. avg(t)*	*k-p. min(Famaç)*	*k-p. avg(Famaç)*	*G. k-p. t*	*G. k-p. Famaç*	*G. k-p. Famaç - k-p. min(Famaç)*	*G. k-p. Famaç - k-p. avg(Famaç)*
2	**00:00:00.8169140**	756.61933	799.9136889	**00:00:01.0544707**	763.6343905	**-7.015060419**	**36.27929842**
3	**00:00:00.7967760**	689.1061357	714.966217	**00:00:00.9459451**	692.1625725	**-3.056436745**	**22.80364458**
4	**00:00:00.7597140**	628.6577139	662.4936068	**00:00:01.2218332**	645.0480228	**-16.39030895**	**17.44558391**
5	**00:00:00.7625800**	594.3356952	630.4452106	**00:00:01.5438552**	596.4052033	**-2.069508094**	**34.04000725**
6	**00:00:00.7617250**	556.9266458	592.563796	**00:00:02.1211862**	562.3576599	**-5.431014039**	**30.2061361**
7	**00:00:00.7885640**	532.056589	558.3207227	**00:00:02.6489323**	531.0644985	**0.992090542**	**27.25622424**
8	**00:00:00.7865190**	527.2782832	547.1027044	**00:00:03.3389569**	507.7845989	**19.49368426**	**39.31810545**
9	**00:00:00.7822520**	484.9704664	519.6655068	**00:00:03.9386129**	480.1659559	**4.804510553**	**39.49955091**
10	**00:00:00.7881870**	480.283	511.8626878	**00:00:04.7828601**	460.156418	**20.12658198**	**51.70626972**
11	**00:00:00.8031910**	465.8238655	494.9934209	**00:00:05.5368913**	440.5778485	**25.24601705**	**54.41557248**
12	**00:00:00.8128420**	452.4079145	470.5618509	**00:00:06.5544239**	425.5167717	**26.89114281**	**45.04507922**
13	**00:00:00.8376450**	440.5949769	468.4770567	**00:00:07.4247083**	407.0330182	**33.56195876**	**61.44403852**
14	**00:00:00.8436830**	414.4802847	449.749707	**00:00:08.8186227**	392.5875736	**21.89271113**	**57.16213341**
15	**00:00:00.8583200**	409.0757925	430.5802078	**00:00:10.2049984**	380.5980456	**28.47774686**	**49.98216219**
16	**00:00:00.8275990**	402.1260943	432.6330368	**00:00:11.0988219**	368.4401234	**33.68597087**	**64.1929134**
17	**00:00:00.8376420**	383.8777005	416.273318	**00:00:12.2979208**	357.8548361	**26.02286444**	**58.41848186**
18	**00:00:00.8514110**	381.5623699	413.6557048	**00:00:13.7292671**	347.0272572	**34.53511264**	**66.62844752**
19	**00:00:00.8713600**	369.9176128	387.3643147	**00:00:15.1331519**	336.7906307	**33.12698206**	**50.57368399**
20	**00:00:00.8994870**	347.6489105	389.6439714	**00:00:16.7562873**	326.2831239	**21.36578661**	**63.36084755**
21	**00:00:00.8716240**	344.1919878	367.8197513	**00:00:18.3090341**	314.853076	**29.33891178**	**52.96667526**
22	**00:00:00.8795820**	347.1194681	369.0534462	**00:00:19.9365410**	305.235801	**41.8836671**	**63.8176452**
23	**00:00:00.9044250**	343.9170937	365.5063958	**00:00:21.7139000**	293.1067436	**50.81035005**	**72.39965212**
24	**00:00:00.9055760**	336.3572787	372.3866397	**00:00:23.6173450**	284.4143573	**51.9429214**	**87.9722824**
25	**00:00:00.9594680**	326.5913478	353.6379351	**00:00:25.4615710**	276.5446633	**50.04668452**	**77.09327176**
26	**00:00:00.9367160**	312.004231	336.5409146	**00:00:27.4684492**	269.4501335	**42.55409751**	**67.09078111**
27	**00:00:00.9497340**	297.8293397	321.9860051	**00:00:29.5312018**	263.0506695	**34.7786702**	**58.93533564**
28	**00:00:00.9774620**	291.5760861	325.3942086	**00:00:31.7545739**	256.750644	**34.82544205**	**68.64356458**
29	**00:00:00.9880960**	299.9814107	323.5477953	**00:00:33.9629339**	250.4525202	**49.52889046**	**73.09527511**
30	**00:00:01.0565970**	290.7587868	314.8485395	**00:00:36.2169762**	244.6117657	**46.14702116**	**70.23677382**
31	**00:00:01.0267400**	279.9379665	310.9096299	**00:00:38.5803145**	239.9062665	**40.03170005**	**71.00336343**
32	**00:00:01.0625090**	282.4293276	309.0260649	**00:00:41.0350774**	234.1029242	**48.32640343**	**74.92314077**
33	**00:00:01.0876590**	282.2373782	302.570723	**00:00:43.7776536**	228.2161533	**54.0212249**	**74.3545697**
34	**00:00:01.0804790**	272.4002205	294.9387775	**00:00:46.1235503**	223.0985309	**49.3016896**	**71.84024659**
35	**00:00:01.1597840**	276.7082097	300.6386311	**00:00:48.8280226**	217.8776897	**58.83052005**	**82.76094139**
36	**00:00:01.1295000**	266.8649363	287.4754186	**00:00:51.6827123**	213.1321467	**53.73278962**	**74.3432719**
37	**00:00:01.1562340**	262.4269424	282.0785187	**00:00:54.4320091**	208.8451689	**53.58177349**	**73.23334975**
38	**00:00:01.1667990**	251.6203533	278.7607971	**00:00:57.2757727**	204.567764	**47.05258932**	**74.19303318**
39	**00:00:01.1874580**	263.5803067	276.6434775	**00:01:00.9993576**	200.3508803	**63.22942643**	**76.29259717**
40	**00:00:01.2337350**	246.0638812	266.5132094	**00:01:03.2816348**	196.5774857	**49.48639549**	**69.93572371**
41	**00:00:01.2145050**	242.4290085	263.1902073	**00:01:06.3969148**	192.4829508	**49.9460577**	**70.7072565**
42	**00:00:01.2510160**	238.1960363	256.5840174	**00:01:09.5563371**	188.7149744	**49.48106193**	**67.86904303**
43	**00:00:01.2665030**	222.272487	255.3322877	**00:01:13.0002306**	184.7295886	**37.54289837**	**70.60269905**
44	**00:00:01.3233530**	238.8892389	251.8570688	**00:01:16.2827505**	181.5330325	**57.35620642**	**70.32403629**
45	**00:00:01.3681650**	228.1973786	253.2359201	**00:01:21.0606323**	178.3448103	**49.85256829**	**74.89110977**
46	**00:00:01.3396560**	234.5676465	255.9889692	**00:01:23.7245496**	177.2761738	**57.29147272**	**78.71279542**
47	**00:00:01.3776540**	212.4154087	239.6756314	**00:01:26.5293447**	177.2761738	**35.13923498**	**62.3994576**
48	**00:00:01.4035930**	200.9207034	236.2748585	**00:01:30.3796816**	177.2761738	**23.64452959**	**58.99868471**
49	**00:00:01.4236640**	200.128157	238.7153515	**00:01:33.7393885**	177.2761738	**22.85198324**	**61.43917773**
50	**00:00:01.4745780**	211.8169259	229.8963202	**00:01:37.7312292**	177.2761738	**34.54075209**	**52.62014648**

14.3 tablolarında “AUTO IMPORTS 85” isimli Amerikan araba ithalat verileri için çalıştırma sonuçları verilmiştir. Tablolarda gamma değerinin 0.5, 0.7 ve 1 değerleri ile 2’den 50 kümeye kadar hesaplama sonuçları verilmiştir.

Hesaplama sonuçlarından elde ettiğimiz sonuç, geliştirilen algoritma ile yapılan kümeleme sonunda verilerin kendi küme merkezleri ile oluşturduğu kümelerde amaç fonksiyon değerinin daha düşük olduğu ve bu sonuca tek seferde hesaplama yaparak ulaşıldığını görüyoruz. Örneğin 14.4 tablosuna bakalım:

Tablo 14.4. Hesaplama tablolarından örnek.

k	*k-p. avg(t)*	*k-p. min(Famaç)*	*k-p. avg(Famaç)*	*G. k-p. t*	*G. k-p. Famaç*	*G. k-p. Famaç - k-p. min(Famaç)*	*G. k-p. Famaç - k-p. avg(Famaç)*
2	**00:00:00.8710430**	558.2952211	593.5091907	**00:00:00.9388943**	569.2343905	**-10.93916938**	**24.27480024**
3	**00:00:00.8010990**	513.9751877	547.9818041	**00:00:01.0086797**	516.6625725	**-2.687384738**	**31.31923162**
4	**00:00:00.7766320**	479.9054449	503.0001924	**00:00:01.2570990**	480.9123899	**-1.006944978**	**22.08780256**
5	**00:00:00.7623090**	454.6871316	467.839157	**00:00:01.7275012**	445.0831068	**9.604024757**	**22.75605022**
6	**00:00:00.7581660**	423.7815418	440.5192082	**00:00:02.3088554**	420.9399342	**2.841607579**	**19.57927397**
7	**00:00:00.7703330**	404.7984208	423.1006366	**00:00:02.7928649**	397.7483028	**7.050117956**	**25.35233382**
8	**00:00:00.8029540**	387.0323639	417.2890557	**00:00:03.6081606**	380.4407355	**6.591628408**	**36.84832017**
9	**00:00:00.7730180**	374.5875369	400.5861609	**00:00:04.2959967**	360.5657822	**14.02175472**	**40.02037865**
10	**00:00:00.7814510**	353.0208028	378.8941245	**00:00:04.8461246**	346.4848853	**6.53591751**	**32.40923921**
11	**00:00:00.7913310**	346.3535412	363.0037067	**00:00:05.7042573**	331.8375231	**14.51601811**	**31.1661836**
12	**00:00:00.8047580**	340.685544	355.4667858	**00:00:06.7625497**	321.1167717	**19.5687723**	**34.35001413**
13	**00:00:00.8156600**	328.0198294	353.3229087	**00:00:07.7434141**	307.7330182	**20.28681127**	**45.58989056**
14	**00:00:00.8304730**	319.4025506	334.3778771	**00:00:08.9118174**	296.3603792	**23.04217143**	**38.01749792**
15	**00:00:00.8613100**	319.6693209	336.2637509	**00:00:09.9850391**	286.6108965	**33.05842436**	**49.65285437**

Burada *“k-p. min($F_{amaç}$)”* isimli ve *“k-p. avg($F_{amaç}$)”* isimli kolonlar sırasıyla veriseti üzerinde k-prototypes algoritmasının 10 defa çalıştırılmasıyla alınan amaç fonksiyonu değerlerinin minimum ve ortalama değerlerini ifade etmektedir. Bu değerler örneğin k=9, yani 9 küme oluşturulduğunda yuvarlak olarak en iyi sonuç, yani minimum 374,588 ve 10 çalıştırmada ortalama 400,587 dir. Buna karşılık *“G. k-p. $F_{amaç}$”* kolonu, yani, global – artımlı k-prototypes ile tek çalıştırmada alınan amaç fonksiyon değeri 360,566 dır. Bu şekilde elde edilen sonuçlarla sunulan algoritmanın klasik algoritmadan ne kadar iyi olduğunu rakamlarla görebilmekteyiz.

Aşağıdaki tabloda tüm verisetleri için γ değerinin 0.5 ve k değerinin 15 olduğu (Tablo 14.5) ve γ değerinin 1 ve k değerinin 10 olduğu (Tablo 14.6) durumdaki hesaplama sonuçları verilmiştir.

Tablo 14.5. γ değerinin 0.5 ve k değerinin 15 olduğu sonuçlardan alıntı özet tablo.

γ=0.5 ve k=15	Veri Sayısı	Nümerik özellik sayısı	Kategorik özellik sayısı	k-p. avg(t)	k-p. min(Famaç)	k-p. avg(Famaç)	G. k-p. t	G. k-p. Famaç	G. k-p. Famaç - k-p. min(Famaç)	G. k-p. Famaç - k-p. avg(Famaç)
Post-Operative Patient	90	2	8	**00:00:00.3147390**	61.86428571	69.54710336	**00:00:00.7713924**	59.6737494	**2.190536316**	**9.873353961**
Zoo	101	2	16	**00:00:00.3710290**	45.3015873	57.95451823	**00:00:01.6187130**	39.27380952	**6.027777778**	**18.68070871**
Acute Inflammations	120	2	7	**00:00:00.4210960**	7.229537037	16.60185436	**00:00:01.0555127**	5.716132756	**1.513404281**	**10.88572161**
Japanese Credit Screening	125	6	9	**00:00:02.0834620**	663.3319039	683.241988	**00:00:54.2020039**	612.3565515	**50.97535237**	**70.88543646**
Teaching Assistant Evaluation	151	2	5	**00:00:00.5337670**	101.5992613	105.9903373	**00:00:01.4869173**	93.87879198	**7.720469304**	**12.11154531**
Servo	167	2	4	**00:00:00.6892580**	108.7899452	116.2239762	**00:00:02.0915187**	102.77285	**6.017095239**	**13.45112628**
Flags	194	2	27	**00:00:00.8246830**	561.5028906	580.8724235	**00:00:06.1265273**	537.7823908	**23.72049976**	**43.09003274**
Auto imports 85	205	14	11	**00:00:00.8287920**	244.8012182	257.4236706	**00:00:10.1261795**	226.6571621	**18.14405607**	**30.7665085**
Statlog (Heart)	270	5	8	**00:00:01.1748530**	251.3898274	263.6823983	**00:00:09.9662338**	248.128759	**3.261068409**	**15.55363937**
Liver Disorders	345	6	1	**00:00:01.4138950**	66.90947511	68.5489187	**00:00:13.9737381**	66.89768117	**0.011793945**	**1.651237528**
Dermatology	366	2	33	**00:00:01.5745270**	1223.700914	1252.602223	**00:00:22.6331794**	1200.204153	**23.49676148**	**52.39806966**
Auto MPG	398	5	3	**00:00:01.8352180**	200.7821024	210.6387279	**00:00:21.8572598**	192.6670845	**8.115017953**	**17.97164338**
Meta-data	528	20	2	**00:00:01.6076870**	473.0069022	520.1673794	**00:01:14.0731963**	434.6634389	**38.34346329**	**85.5039405**
Statlog (Australian Credit Approval)	690	6	8	**00:00:03.2927020**	669.4265529	695.8782762	**00:01:24.6574717**	638.9518488	**30.47470411**	**56.92642742**
Statlog (German Credit Data)	1000	7	13	**00:00:02.5101640**	1774	1808.6	**00:01:47.4250742**	1667	**107**	**141.6**
Contraceptive Method Choice	1473	2	8	**00:00:06.8159260**	1468.299888	1505.040263	**00:03:38.0817274**	1364.175495	**104.1243937**	**140.8647684**

Tablo 14.6. γ değerinin 1 ve k değerinin 10 olduğu sonuçlardan alıntı özet tablo.

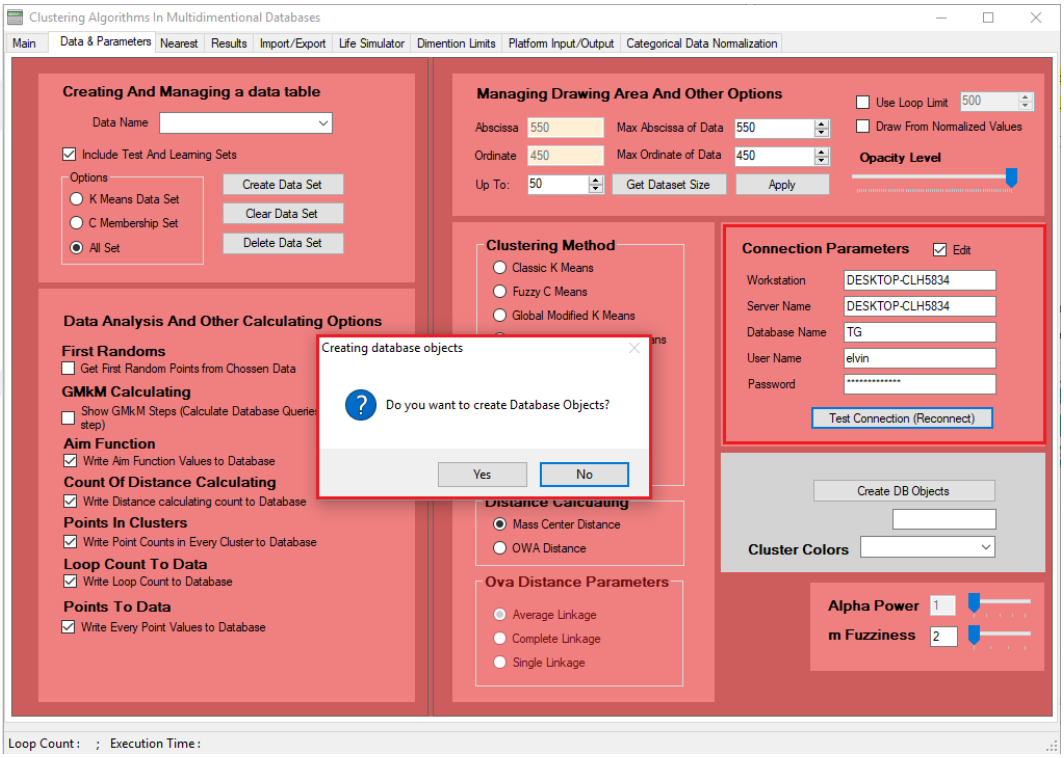
γ=1.0 ve k=10	*Veri Sayısı*	*Nümerik özellik sayısı*	*Kategorik özellik sayısı*	*k-p. avg(t)*	*k-p. min(Famaç)*	*k-p. avg(Famaç)*	*G. k-p. t*	*G. k-p. Famaç*	*G. k-p. Famaç - k-p. min(Famaç)*	*G. k-p. Famaç - k-p. avg(Famaç)*
Post-Operative Patient	90	2	8	**00:00:00.2934590**	143.4584055	152.8123909	**00:00:00.4702090**	134.2425019	**9.215903541**	**18.56988894**
Zoo	101	2	16	**00:00:00.3376570**	119.4850529	143.6644288	**00:00:00.8427341**	101.4604314	**18.02462153**	**42.20399744**
Acute Inflammations	120	2	7	**00:00:00.3720040**	28.53561135	52.94237547	**00:00:00.5774846**	9.359338624	**19.17627272**	**43.58303685**
Japanese Credit Screening	125	6	9	**00:00:02.1103820**	1266.411959	1343.595965	**00:00:25.1682100**	1209.915845	**56.49611408**	**133.6801198**
Teaching Assistant Evaluation	151	2	5	**00:00:00.4834660**	210.8923988	225.7292995	**00:00:00.7928781**	209.3348259	**1.557572902**	**16.39447357**
Servo	167	2	4	**00:00:00.6452460**	245.6999426	255.9803474	**00:00:01.1447410**	231.010655	**14.68928761**	**24.96969241**
Flags	194	2	27	**00:00:00.8078410**	1221.502366	1255.071759	**00:00:03.0258724**	1200.798438	**20.70392799**	**54.27332133**
Auto imports 85	205	14	11	**00:00:00.7881870**	480.283	511.8626878	**00:00:04.7828601**	460.156418	**20.12658198**	**51.70626972**
Statlog (Heart)	270	5	8	**00:00:01.1101850**	496.6621874	511.6722443	**00:00:04.9979811**	471.2938041	**25.3683833**	**40.37844019**
Liver Disorders	345	6	1	**00:00:01.3621320**	72.73809023	74.96972092	**00:00:06.9476283**	73.06068912	**-0.32259889**	**1.909031798**
Dermatology	366	2	33	**00:00:01.7771030**	2535.115771	2662.630563	**00:00:10.0704404**	2533.436908	**1.67886245**	**129.1936552**
Auto MPG	398	5	3	**00:00:01.7646360**	365.8278863	384.2852596	**00:00:11.4591477**	360.8111331	**5.016753203**	**23.47412651**
Meta-data	528	20	2	**00:00:01.7626460**	1031.801384	1070.954085	**00:00:40.9163048**	1024.852826	**6.948557501**	**46.10125897**
Statlog (Australian Credit Approval)	690	6	8	**00:00:02.2656760**	1322.855275	1373.628332	**00:00:26.7638341**	1248.389251	**74.46602356**	**125.2390811**
Statlog (German Credit Data)	1000	7	13	**00:00:02.7709880**	3789	3837.7	**00:00:53.3536075**	3602	**187**	**235.7**
Contraceptive Method Choice	1473	2	8	**00:00:07.2004430**	2963.506575	3009.010347	**00:01:46.0863767**	2709.629648	**253.8769269**	**299.3806982**

Her iki özet tabloda da hesaplamaların çalışma sürelerini ve sonuç amaç fonksiyonu değerlerini görebilmekteyiz. Sonuçları özetlersek sunulan algoritmanın çalışma süresi klasik yönteme göre (saniyeler bazında) daha yavaş olsa da sunduğu çözümler daha iyi sonuç vermektedir.

15. GELİŞTİRİLEN YAZILIMIN ÖZELLİKLERİ, PARAMETRELERİ VE KULLANIMI

15.1. Veri Hazırlama ve Normalizasyon

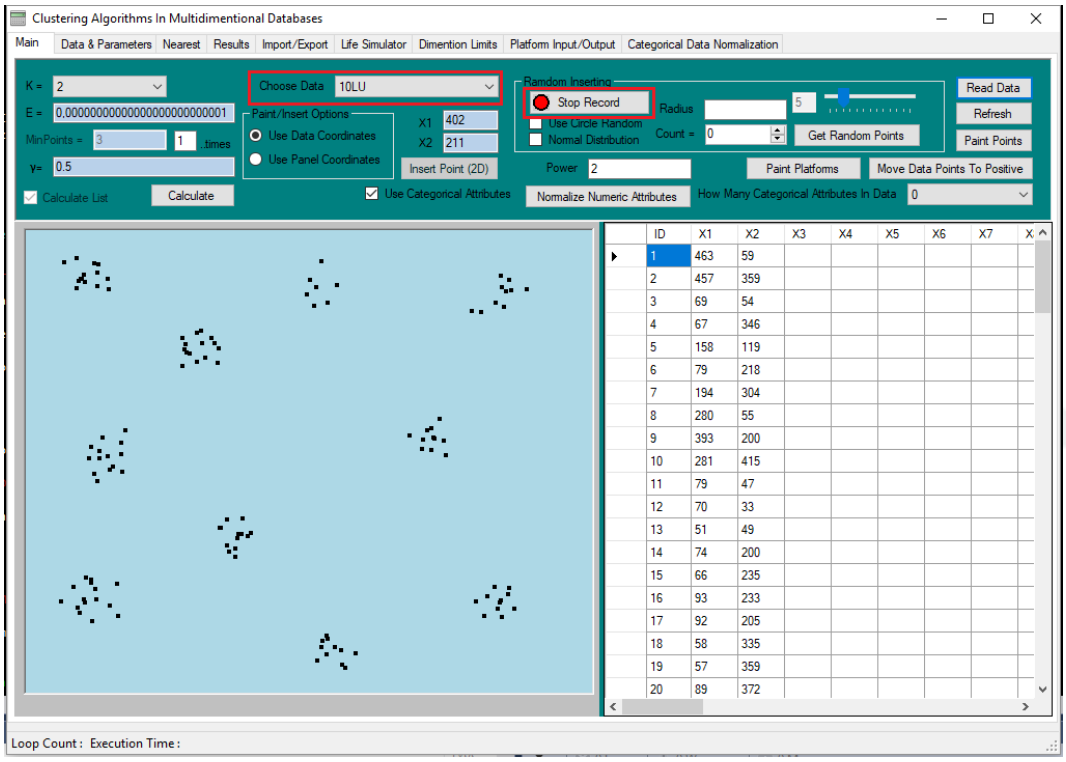
Uygulama açıldıktan sonra verilerin çekileceği, hesaplanacağı ve sonuçların çeşitli detaylarda tutulacağı veritabanını seçip bağlanıyoruz. Bu aşamada isteğe bağlı olarak uygulamanın çalışması için veritabanında ihtiyaç duyulan nesnelerin oluşturulmasını da onaylayıp sağlayabiliriz (Şekil 15.1).



Şekil 15.1. Veritabanı seçimi ve bağlantısı. Uygulama altyapısı için ihtiyaç duyulan nesnelerin veritabanında oluşturulması sorusu.

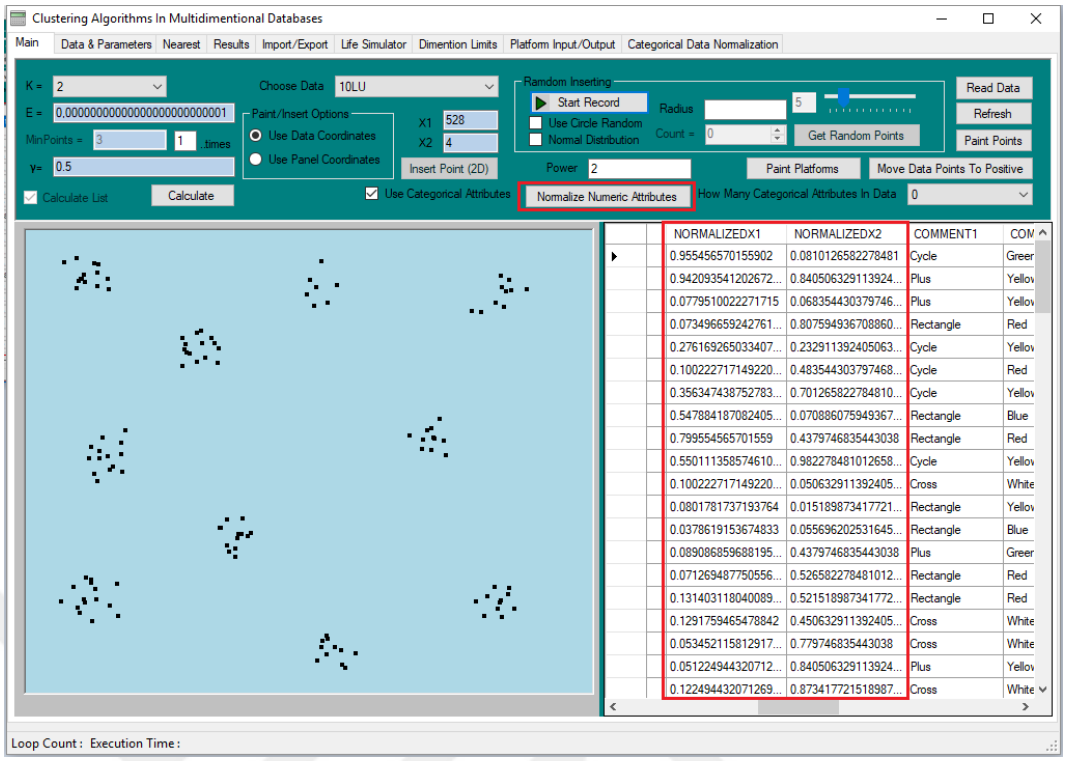
Sonra istenir ise hazır veriden, istenir ise anlık veri üreterek üzerinde çalışma yapılacak verisetimizi ayarlıyoruz. Bunun için eğer dışarıdan hazır veri seti ile çalışılacak ise sadece bu veriseti için uygulama içinden tablo oluşturup içini gerekli kümelenecek veri ile doldurup hesaplama işlemlerine başlayabiliriz. Anlatımda ise sıfırdan küçük bir veriseti oluşturup, onun üzerinden göstereceğiz.

10LU isimli ve iki nümerik ve 3 kategorik özellikten oluşan verisetimizi uygulama yardımı ile oluşturduk. Bu veri setinin özelliği, iki nümerik özellik değerleri ile iki boyutlu yüzeyde gösterilirken, bir-birinden ayrılmış 10 küme gözlemleyebilmekteyiz (Şekil 15.2).



Şekil 15.2. Seçilmiş verisetimizin iki nümerik özellik değerine göre ikiboyutlu yüzeyde görünüşü.

Oluşan veri dağılımını normalize ediyoruz (Şekil 15.3).

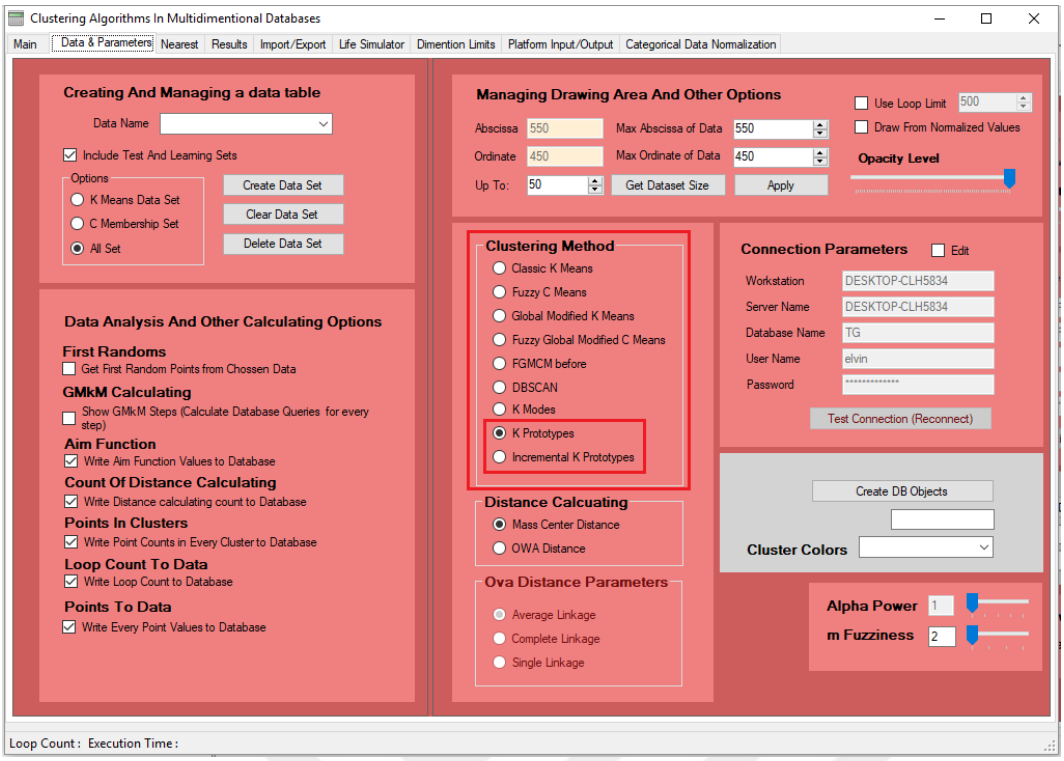


Şekil 15.3. Verinin Normalize edilmesi

15.2. Algoritmaların Ayarlanması ve Farklı Parametrelerle Çalıştırılması.

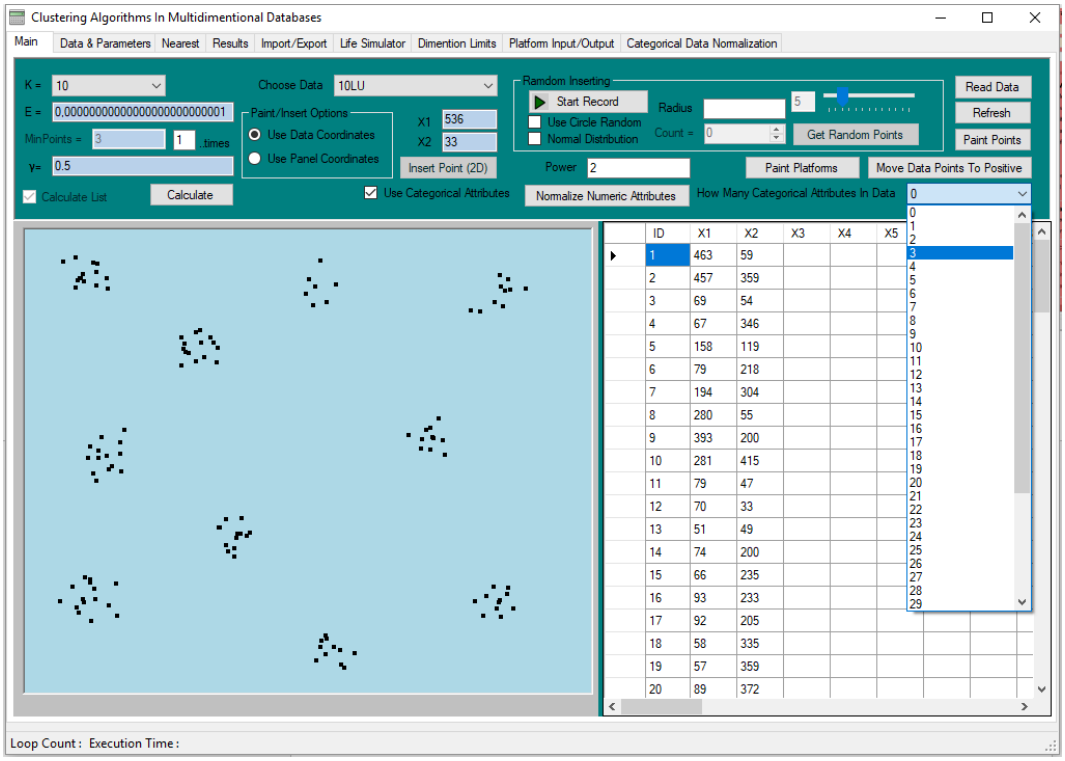
Veri hazırlandıktan sonra verinin kaç nümerik, kaç kategorik özellikten ibaret olması, kaç kümeye ayrılacağı (*k* değeri) ve γ-*gamma* (kategorik verilerin önem katsayısı) değerleri ayarlanıp hesaplama işlemi başlatılmalıdır.

Hangi algoritma ile kümeleme yapılacağı seçiliyor (Şekil 15.4).



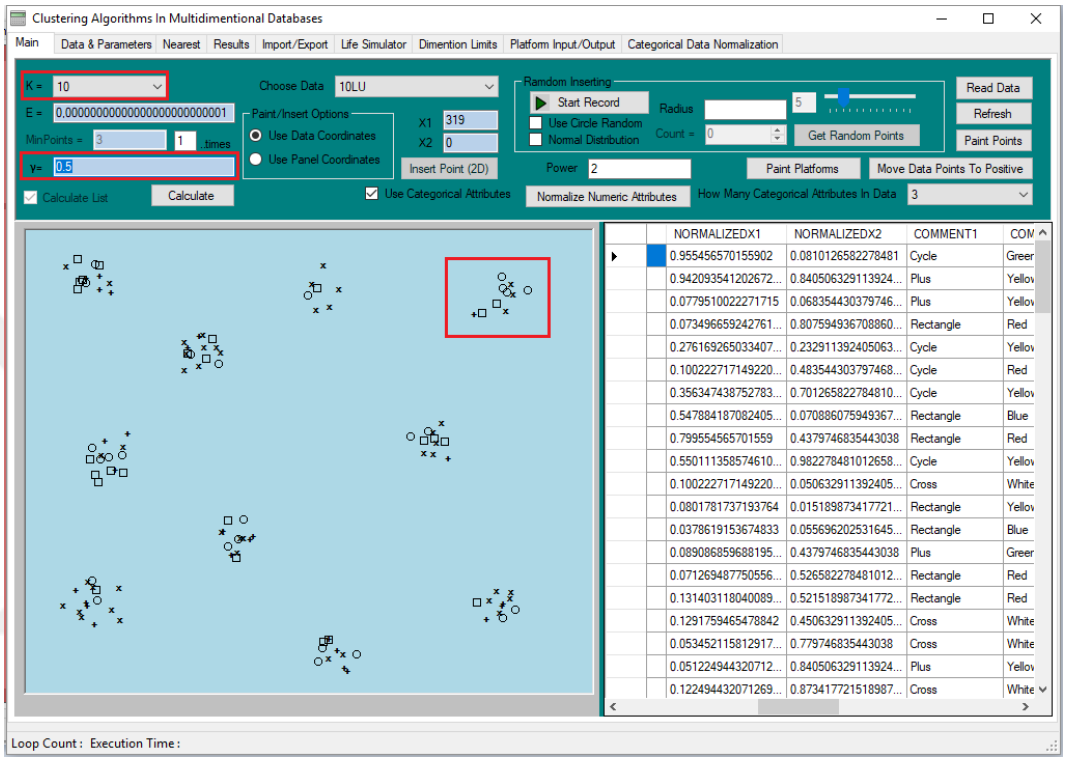
Şekil 15.4. Algoritma seçimi

Sonra özellik sayıları seçilmektedir. Program bu değeri direkt veri setinden anlayabildiği gibi manuel olarak da girilebilmektedir (Şekil 15.5).



Şekil 15.5. Kategorik özellik sayısını ayarlama

k ve γ değerleri de seçildikten sonra (Şekil 15.6) işleme başlanabilir. Özellik sayısı olarak birden fazla değer girildiğinde, ikiboyutlu ölçekte gösterilen verilerin artık biçimini de (“kare”, “yuvarlak”, “x” ve “+” olarak rastgele veya isteğe bağlı olarak veriseti oluşturma aşamasında yeni noktalar tanımlanabilmektedir) görebilmekteyiz. Bununla gamma değerine bağlı, bu biçimlerin de kümelemeye nasıl etki ettiğini görsel olarak da görebilmiş oluyoruz.

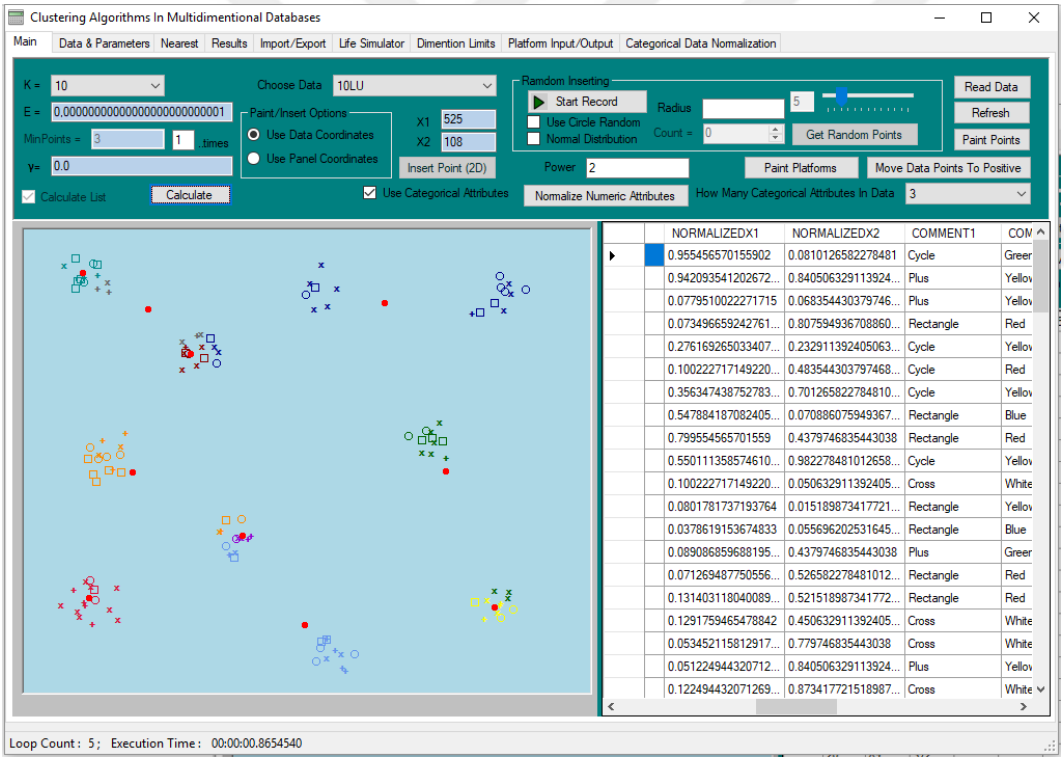


Şekil 15.6. k ve γ değerlerini ayarlama. Verinin kategorik özelliğine göre biçiminin görünmesi.

15.3. Çeşitli Parametreler ile Kümeleme ve Sonuçların Yorumu.

Ele alınan veriseti için nümerik özelliklerle iki boyutlu görüntüsünün 10 seçilebilir kümeye ayrılmasından dolayı *k* değerini örneklerimizde 10 tutmaktayız. Bununla beraber verisetinin 3 özelliğinin kategorik olmasıyla γ değerini 0-dan 1-e kadar birkaç farklı değer verip sonucu görsel olarak izleyebilmekteyiz. Aynı parametre seçimleri ile sonrasında geliştirilmiş artımlı k-prototypes algoritması ile de sonuçlar gösterilmiştir.

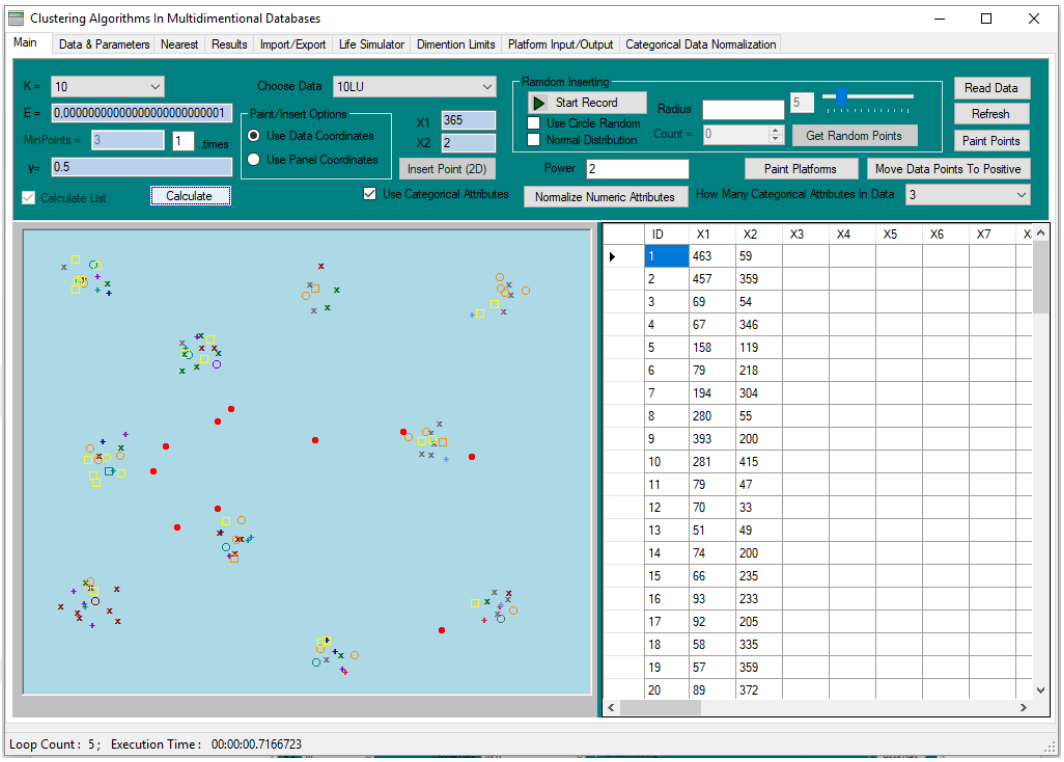
İlk önce k-prototypes algoritması ile ve γ –nın 0 olduğu birkaç çalışma sonucunu görebiliriz. γ değerinin sıfır olması algoritmayı k-means gibi çalıştırmakta ve sonuçlar rastgele olmaktadır (Şekil 15.7).



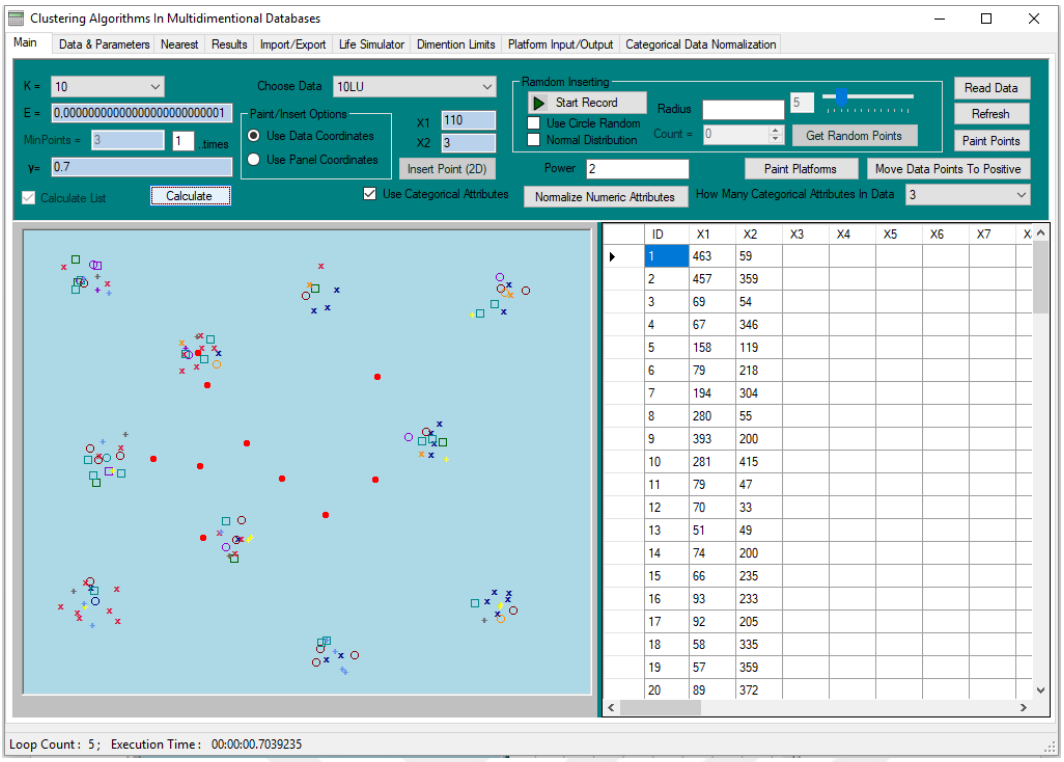
Şekil 15.7. k-prototypes algoritması ile γ değerinin sıfır olduğu koşullarda çeşitli sonuçlardan biri. Her renk ile ayrı bir küme gösterilmiştir.

γ değeri arttıkça verilerdeki kategorik özelliklerin farklı olmaları veri noktaları arasındaki mesafeye daha büyük etki etmektedir. Bununla kümelenen verilerin atanacağı kümeye de daha büyük etki etmiş olmaktadır.

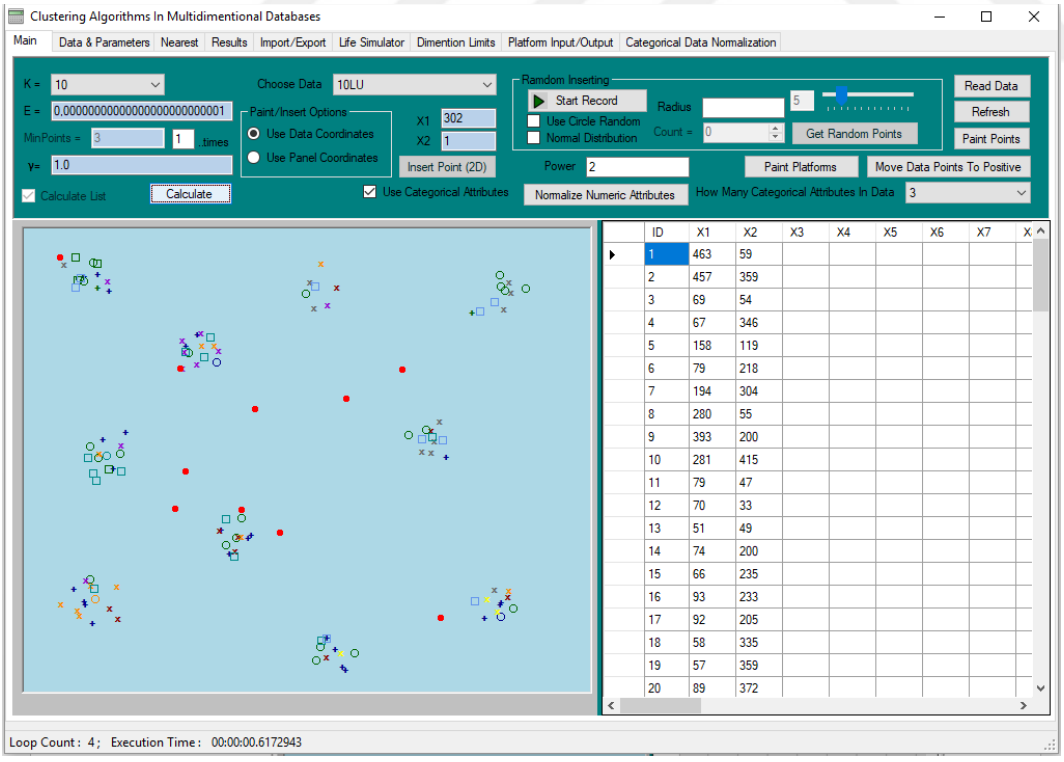
Aynı k-prototypes algoritması ile γ değerinin 0.5 , 0.7 ve 1.0 olduğu sonuçlar aşağıdaki resimlerde verilmiştir (Şekil 15.8; 15.9; 15.10). Sonuçlarda görüldüğü üzere artık veriler kümelere sadece nümerik enlem ve boylam değerleri ile değil, biçimlerine göre de atanıyorlar.



Resim 15.8. k-prototypes algoritması ile γ değerinin 0.5 olduğu koşullarda sonuçlardan biri. Her renk ile ayrı bir küme gösterilmiştir.

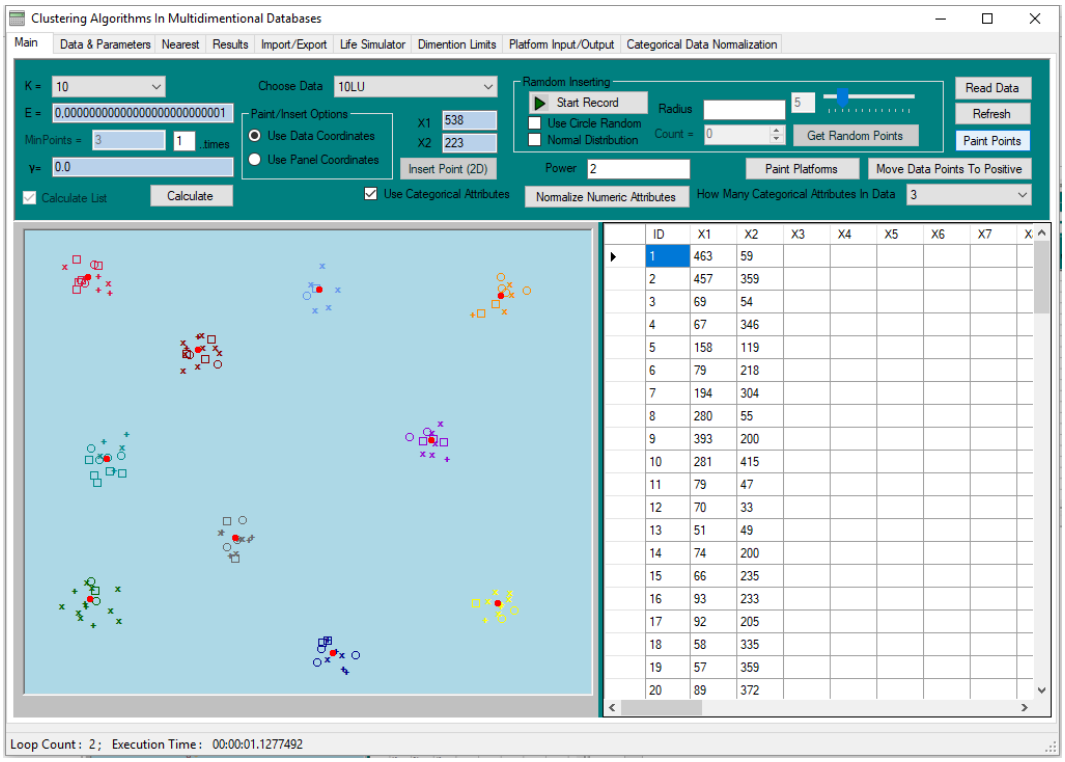


Resim 15.9. k-prototypes algoritması ile γ değerinin 0.7 olduğu koşullarda sonuçlardan biri. Her renk ile ayrı bir küme gösterimiştir.

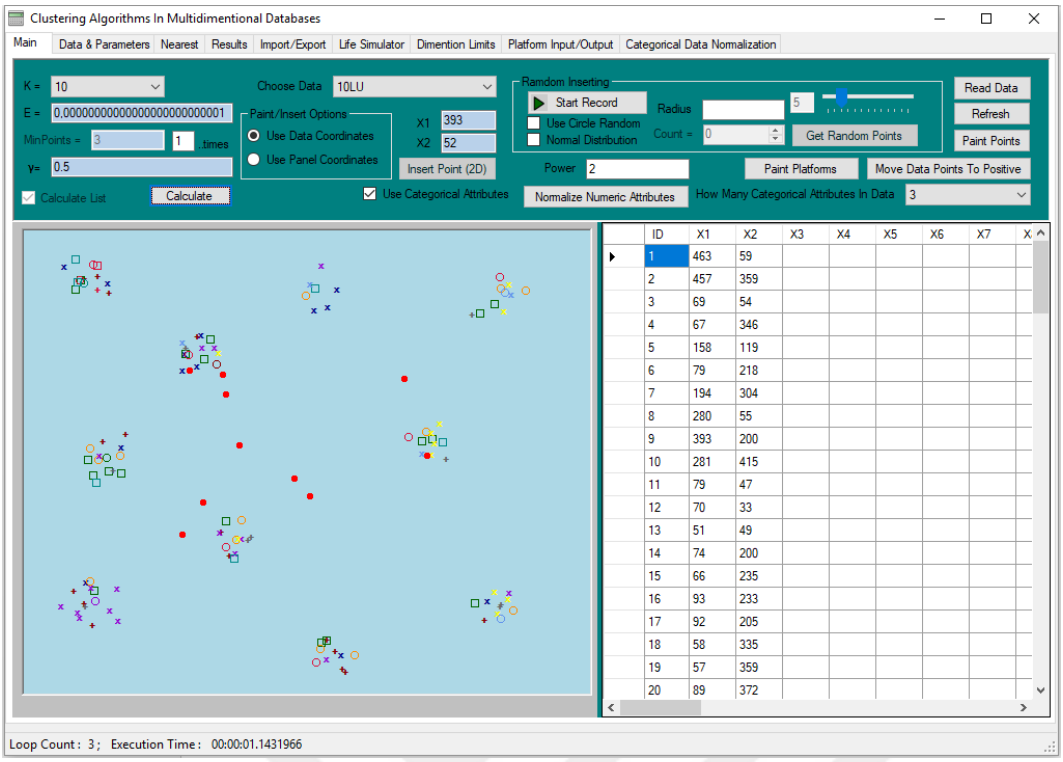


Resim 15.10. k-prototypes algoritması ile γ değerinin 1.0 olduğu koşullarda sonuçlardan biri. Her renk ile ayrı bir küme gösterimiştir.

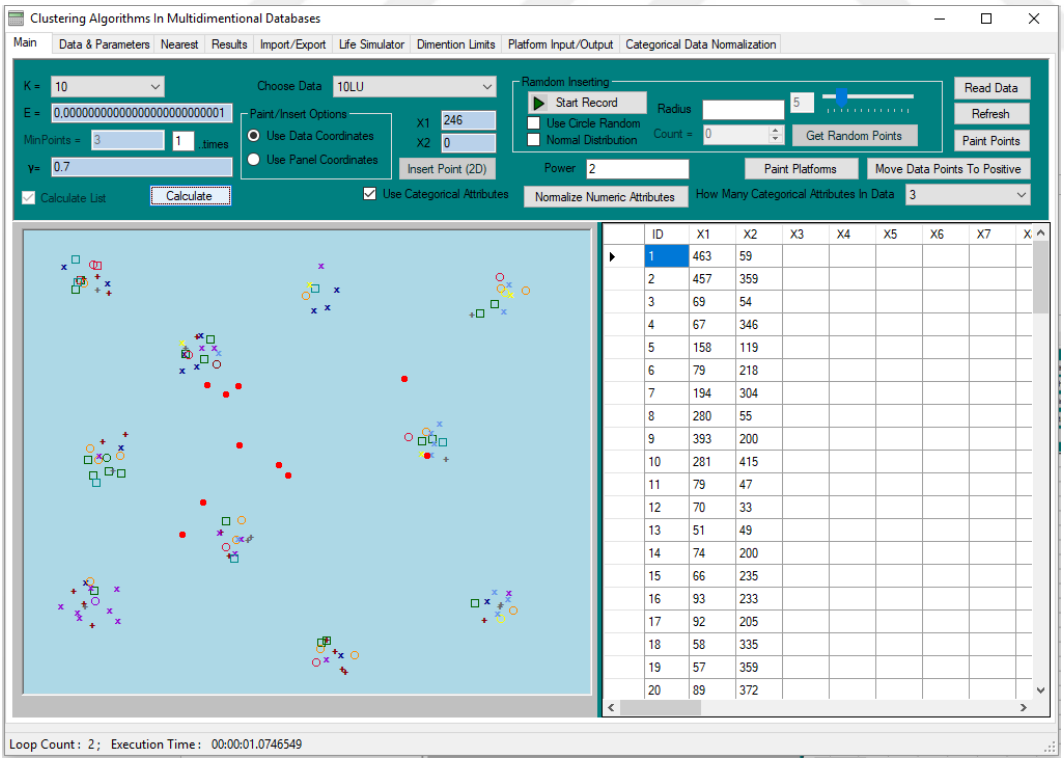
Geliştirilen Artımlı k-prototypes algoritması için de γ değerinin önemi aynıdır. Fakat bu algoritmanın önemli avantajı, sonucun global çözüm olması ve hesaplamanın deterministikliğidir. γ değerinin 0, 0.5, 0.7 ve 1.0 olduğu sonuçlar aşağıdaki resimlerde verilmiştir (Şekil 15.11 - 15.14). Şekillerde her renk ile ayrı bir küme gösterilmiştir ve kümeler nümerik özelliklerine göre görsel olarak rahatlıkla görülebilir şekilde kümelenmiştir.



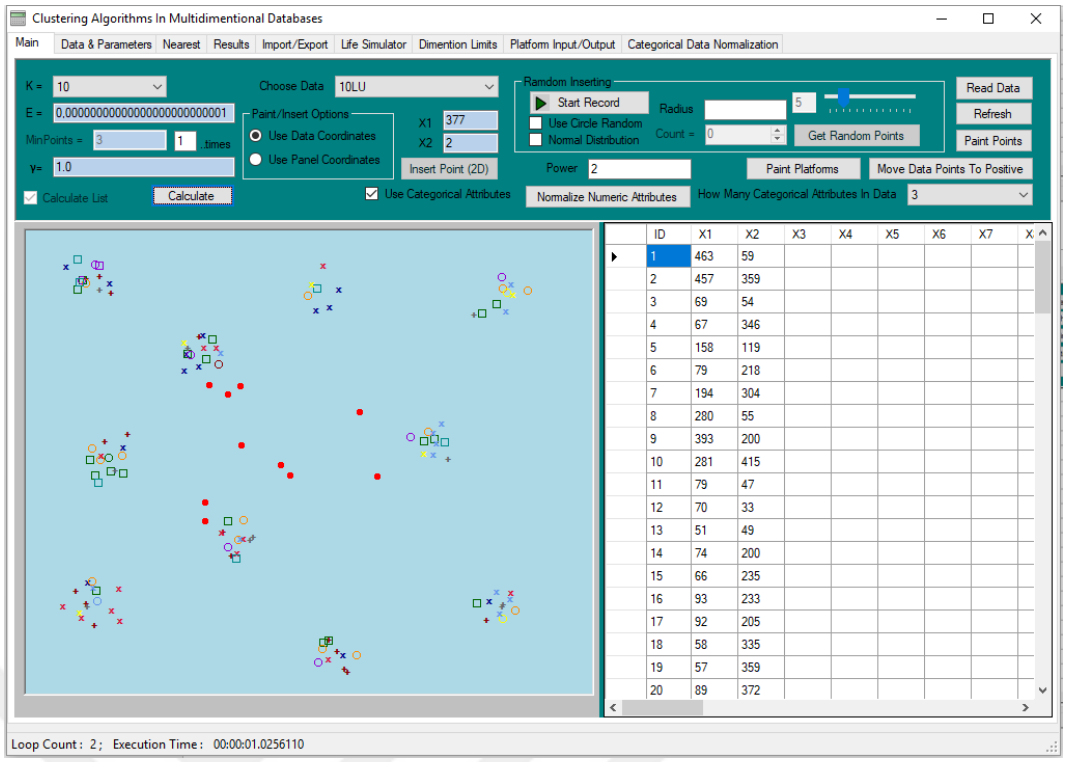
Resim 15.11. Artımlı k-prototypes algoritması ile γ değerinin 0 olduğu koşulda sonuç.



Resim 15.12. Artımlı k-prototypes algoritması ile γ değerinin 0.5 olduğu koşulda sonuç.



Resim 15.13. Artımlı k-prototypes algoritması ile γ değerinin 0.7 olduğu koşulda sonuç.



Resim 15.14. Artımlı k-prototypes algoritması ile γ değerinin 1.0 olduğu koşulda sonuç.

Böylece aynı veriseti k=10 için ve $\gamma \in \{0; 0.5; 0.7; 1.0\}$ için k-prototypes algoritmasıyla ikişer defa ve artımlı k-prototypes algoritmasıyla birer defa çalıştırıldı. Alınan sonuç amaç fonksiyon değerleri aşağıdaki tablolarda verilmiştir (Tablo 15.1 ve Tablo 15.2).

Tablo 15.1. Hesaplama sabit girdileri

Veriseti	K_MEANS_10LU
Küme Sayısı	10
Nümerik Özellik Sayısı	2
Kategorik Özellik Sayısı	3

Tablo 15.2. k-prototypes ve Artımlı k-prototypes algoritmaları ile hesaplama sonuçları.

Çalıştırılan Yöntem	Gamma değeri	Kümeleme sonucu Amaç değeri	Şekil No.
KP (k-Prototypes) ilk	0	**9.051444586**	15.7.1
KP (k-Prototypes)	0	**9.54494264**	15.7.2
KP (k-Prototypes) ilk	0.5	**86.58721254**	15.8.1
KP (k-Prototypes)	0.5	**88.87121236**	15.8.2
KP (k-Prototypes) ilk	0.7	**98.15875708**	15.9.1
KP (k-Prototypes)	0.7	**109.1314412**	15.9.2
KP (k-Prototypes) ilk	1	**128.4639695**	15.10.1
KP (k-Prototypes)	1	**147.2403494**	15.10.2
GKP (Global k-prototypes)	0	**5.147610414**	15.11
GKP (Global k-prototypes)	0.5	**80.91317215**	15.12
GKP (Global k-prototypes)	0.7	**95.26009486**	15.13
GKP (Global k-prototypes)	1	**115.0958666**	15.14

Sonuçlardan da görüldüğü gibi klasik k-prototypes algoritması her gamma değeri için iki defa çalıştırıldı ve her defasında lokal çözümler üretti. Buna karşılık Artımlı k-prototypes algoritması tek seferde global çözüm üretti ve sonuç amaç değerleri de k-prototypes algoritması değerlerinden daha küçük oldu.

Ele alınan 16 veri seti için, bu bölümde anlatılan yöntemler ile hesaplamalar yapılmıştır. 14. bölümdeki tablolarda 14 nümerik ve 11 kategorik özelliği olan "Auto IMPORTS-85" isimli amerikan araba ithalatı veriseti için hesaplamaların daha detaylı bilgileri gösterilmiştir.

16. SONUÇ

Kümeleme analizi günümüzde büyük öneme sahiptir. Bankacılıktan tıpa, sigortacılıktan ticarete, jeolojiden kimyaya her alanda elde olan verilerin kümelenme ile analizi ve sonuçların kullanılması artık büyük değer kazanmaktadır.

Çeşitli alanlarda, çeşitli kaynaklardan toplanan verilerin türleri farklılık göstermektedir. Veriler, üzerinde hesaplamaların daha rahat olduğu nümerik veriler gibi, daha özel yaklaşımlar gerektiren kategorik verilerden oluşmaktadır. Verilerin türüne göre kümeleme yöntemleri farklılık göstermektedir. Literatürde sadece nümerik veriler üzerinde etkin çalışan algoritmalar yer almaktadır. Bu algoritmalardan yaygın olan k-means algoritması veriler arasındaki uzaklığı baz alarak kümeleme yapmaktadır. Fakat bu uzaklık değerinin elde edilmesi için verileri oldukları boyutta bir nokta olarak ele almak gerekir. Bunun için de verilerin özellikleri nümerik olmalıdır. Bununla da, sözü geçen yaklaşımla kategorik verilerin kümelenmesi zorlaşmaktadır.

Literatürde aynı zamanda kategorik veriler üzerinde kümeleme yöntemleri yer almaktadır. Bu yöntemlerdeki yaklaşımlardan biri de kategorik özelliklerin benzerlik ve yakınlık kavramıdır. k-prototypes algoritması bu yaklaşımı kullanmaktadır. Fakat k-prototypes algoritması k-means algoritması gibi rastgeleliğe dayanmaktadır ve lokal çözüm olarak sunduğu sonuçlar aynı olmayıp en iyi oldukları her zaman söylenemez.

Zamanla k-means algoritmasının global çözümünü sağlayan global k-means algoritması ve onun geliştirilmiş versiyonu global modified k-means algoritmaları geliştirildi. Fakat bu algoritmalar kategorik özelliklerle çalışmamaktalar.

Bu tezde, kategorik ve nümerik özellikleri bir arada olan veriler üzerinde kümeleme probleminin çözümü için yeni bir artımlı Global k-prototypes (GKP) algoritması geliştirilmiştir. Algoritmanın özelliği 1) daha doğal veri yapıları ile, yani, hem kategorik hem de nümerik özellikleri olan verilerle çalışabilmesi, 2) gürültülerden etkilenmemesi, 3) k-means tabanlı çoğu algoritma gibi rastgeleliğe dayanmayıp global çözümler üretmesi ve 4) bu avantajları ile beraber hız açısından da yeterli olmasıdır. Ayrıca, geliştirilen yöntemin nümerik ve kategorik

özellikleri bir arada olan verisetleri için çalıştırılabilmesi, algoritmanın uyarlanacağı alanı oldukça genişletmektedir. Artımlı kümeleme özelliği ise, artımlı olmayan klasik yöntemlerden global çözümleri bulma ve başlangıçtaki rastgelelik olmadığından, determenistik olması ile seçiliyor.

Sunulan algoritmaların yazılımı C# programlama dilinde geliştirilmiştir. Elde olan ve hesaplama sonuçlarındaki verileri tutmak için MS SQL veritabanı kullanılmıştır. Tutulan veriler her adım için toplam amaç fonksiyon değeri, hesaplanan mesafeler, çalışma süreleri, döngü sayıları gibi çok detaylı toplanmıştır.

Geliştirilen algoritma k-prototypes algoritması ile beraber UC Irvine Machine Learning Repository (http://archive.ics.uci.edu/ml/) veri ambarından seçilen 16 gerçek veri seti üzerinde uygulanmış ve sonuçların analizi yapılmıştır.

Bölüm 14'de sunulan algoritma ile k-prototypes algoritması aynı girdilerle aynı verisetleri üzerinde çalıştırılmış ve sonuçları sunulmuştur. Ele alınan tüm veri setleri için sonuçlar fazla yer kaplayacağından, tez çalışmasında yalnızca bir veri seti için 0.5 ve 0.7 gamma değeri girdisinin sonuçları tablolarla sunulmuştur. Algoritmanın k-prototypes algoritmasından daha iyi sonuç vermesi ve bunu tek çalıştırmada yapması hesaplamalarla gösterilmiştir.

KAYNAKLAR DİZİNİ

Abou-Moustafa K. T., Schuurmans D., and Ferrie F. P., 2013,*"Learning a metric space for neighborhood topology estimation: Application to manifold learning,"* J. Mach. Learn. Res. Workshops Conf. Proc., vol. 29, pp. 341–356, Nov.

Ahmad A., Lipika D., 2007, *"A k-mean clustering algorithm for mixed numeric and categorical data"*, Data & Knowledge Engineering 63, pp. 503–527

Alexandros N.. 2001, *"C2P: Clustering based on Closest Pairs"*, 27. VLDB Conference, İtaly.

Amari S.-I. and Nagoka H., 2000,*"Methods of information geometry,"* in Translations of Mathematical Monographs, vol. 191, American Mathematical Society. New York: Oxford Univ. Press, 2000.

Amir Ben-Dor. and Zohar Yankini., 1999, *"Clustering Gene Expression Patterns"*, 3rd International Computational Molecular Biology Conference, Leon, Fransa, pp.11-14.

Anderson E., 1935: *"The Irises of the Gaspe Peninsula"*. Bulletin of the American Iris Society, 59:2–5.

Andritsos P. vd. 2004, *"LIMBO: Scalable clustering of categorical data"*. EDBT Conference.

Andritsos P., Tsaparas P., Miller R., Kenneth J., Sevcik C., 2003, *"LIMBO: A Scalable Algorithm to Cluster Categorical Data"*, University of Toronto, Department of Computer Science, CSRG Technical Report 467, July 7, 2003.

Aranganayagi S., and Thangavel K., 2009, *"Improved K-Modes for Categorical Clustering Using Weighted Dissimilarity Measure"*, International Journal of Information and Mathematical Sciences 5:2 2009

Bagirov A.M., 2008, *"Modified global k-means algorithm for minimum sum-of-squares clustering problems"*, , Pattern Recognition 41, pp. 3192- 3199.

Bagirov A.M., Rubinov A. M., Soukhoroukva N.V., 2003, *"Unsupervised and Supervised Data Classification via Nonsmooth and Global Optimization",* TOP, pp. 1-22.

Bagirov A.M., Rubinov A. M., Yearwood J., 2002, *"A global optimisation approach to classification",* Optimization and Engineering, 3(2), 129-155.

Bagirov A.M., Yearwood J., 2006, *"A new nonsmooth optimization algorithm for minimum sum-of-squares clustering problems",* Eur J Oper Res 170(2):578-596.

KAYNAKLAR DİZİNİ (DEVAMI)

Bellazzi R., Zupan B., 2008, *"Predictive data mining in clinical medicine: Current issues and guidelines"*, International Journal of Medical Informatics, Volume 77, Issue 2, February 2008, Pages 81–97

Berkin P., *"Survey of Clustering Data Mining Techniques",* https://www.cc.gatech.edu/~isbell/reading/papers/berkhin02survey.pdf (Son erişim tarihi: 4.09.2017)

Bock, H.H., 1998, *"Clustering and neural networks",* In: Rizzi, A., Vichi, M. & Bock, H.H. (eds), Advances in Data Science and Classification, Springer-Verlag, Berlin, pp. 265-277.

Bray J.R., Curtis J.T., 1957, *"An ordination of the upland forest communities of Southern Wisconsin".* Ecological Monographies 27:325-349.

Carlos A., Enrique R., René A., Pere C., Ivan D., Salvador B. ve Beatriz F.G., 2009, Member, IEEE, *"Data Mining of Patients on Weaning Trials from Mechanical Ventilation Using Cluster Analysis and Neural Networks"*, 31st Annual International Conference of the IEEE EMBS, Minneapolis, Minnesota, USA, September 2-6, pages 4343-4346, 2009.

Chaturvedi A., Woods K., Green P.E. and Carroll J. D, 2001, *"K-Modes Clustering"*, Journal of Classification 18: 35-55.

Chidanand A., Bing L., Edwin P.D.P. ve Padhraic S., 2002, *"Business Applications of Data Mining",* Communıcations of the ACM, August 2002/Vol. 45, No. 8.

Christopher D. Manning and Hinrich Schütze. 1999. *"Foundations of Statistical Natural Language Processing".* MIT Press.

Clarke K.R., Somerfield P.J., Chapman M.G. 2006, *"On resamblance measures for ecological studies, including taxonomic dissimilarities and a zero-adjusted Bray-Curtis coefficient for denuded assemblages"*. Journal of Experimental Marine Biology and Ecology 330:55-80

Clarke K.R., Warwick R.M. 1994, *"Change in marine communities: An approach to statistical analysis and interpretation"*. Natural Environment Research Council, UK.

Defays D.. 1977. *"An Efficient Algorithm for a Complete Link Method".* The Computer Journal. Volume 20, Number 4, pp. 364-366.

Dempster Arthur. P. vd., *"Maximum Likelihood from Incomplate Data via the EM Algorithm"*, journal of the Royal Statistical Society Agglomerative, B Serisi, Vol 39, 1977, p. 1-38.

KAYNAKLAR DİZİNİ (DEVAMI)

Digby P.G.N., Kempton R.A. ,1987, *"Multivariate analysis of ecological communities".* Chapman & Hall, London.

Dunham Margaret H., 2003, *"Data Mining Introductory and Advanced Topics"*, Prentice Hall, Pearson Education Inc., New Jersea, 2003, s.8.

Field J.G., Clarke K.R., Warwick R.M., 1982, *"A practical strategy for analysing multispecies distribution patterns"*. Marine Ecology Progress Series 8:37-52.

Fisher R. A., 1936, "The use of multiple measurements in taxonomic problems," Ann Eugen., vol. 7, no. 2, pp. 179–188, Sept. 1936.

Ganti V., Gehrke J., and Ramakrishnan R.. , 1999, *"CACTUS-clustering categorical data using summaries"*. ACM KDD Conference.

Gauch H.G., 1982, *"Multivariate analysis in community ecology"*. Cambridge University Press, New York.

Giudici Paolo, 2004, *"Applied Data Mining: Statistical Methods For Business and Industry"*, Wiley, 2004, s.83.

Gower J.C., 1971, *"A general coefficient of similarity and some of its properties"*. Biometrics 27:857-871.

Gower J.C., 1987, *"Introduction to ordination techniques. In: Developments in Numerical Ecology"*. Edited by Legendre and Legendre. NATO ASI Series G 14.

Guha S., Rastogi R. and Shim K.. 2000, *"ROCK: A robust clustering algorithm for categorical attributes"*. Information Systems, 25(5):345–366, 2000.

Han J. and Kamber M., 2001, *"Data Mining Concepts and Techniques"*, Morgan Kaufman Publishers, Academic Press, p. 106.

Huang Z., *"Clustering Large Data Sets With Mixed Numeric And Categorical Values",* CSIRO Mathematical and Information Sciences, GPO Box 664 Canberra ACT 2601, AUSTRALIA

Huang Z., 1998,*"Extensions to the k-Means Algorithm for Clustering Large Data Sets with Categorical Values"*, 1998 Kluwer Academic Publishers. Manufactured in The Netherlands, Data Mining and Knowledge Discovery 2, 283–304.

Huang Z., Ng M.K., 2003, *"A Note on K - modes Clustering"*, Journal of Classification 20:257-261, DOI: 10.1007/s00357-003-0014-4

Jain A. K., Murty M. N., Flynn P. J.. 1999. "*Data Clustering: A Review",* ACM Computing Surveys (CSUR), Volume 31, Issue 3, ACM Press, New York, pp. 264-323.

KAYNAKLAR DİZİNİ (DEVAMI)

Khan S., Kant S., *"Computation of Initial Modes for K-modes Clustering Algorithm using Evidence Accumulation"*, IJCAI-07, p. 2784-2789

Kulis B., 2013, *"Metric learning: A survey,"* Found. Trends Mach. Learn., vol. 5, no. 4, pp. 287–364.

Lavrač N., 1999, *"Selected techniques for data mining in medicine"*, Artificial Intelligence in Medicine, Volume 16, Issue 1, May 1999, Pages 3–23, (Data Mining Techniques and Applications in Medicine).

Legendre P., Legendre L., 1998, *"Numerical ecology".* 2nd Edition, Elsevier, Amsterdam.

Likas A., Vlassis N. and Verbeek J., 2001, *"The global k-means clustering algorithm"*, Pattern Recognition 36, pp. 451 – 461.

MacQueen, J., 1967, *"Some Methods for Classification and Analysis of Multivariate Observations"*, 5th Berkeley Mathematical Statistics and Probability Semposium, University of California Press, s. 281-297.

Mastrogiannis N., Giannikos I., Boutsinas B. and Antzoulatos G., 2009, *"CL.E.KMODES: A modified k-modes clustering algorithm"* , Journal of the Operational Research Society, 1-11

Moth'd Belal. Al-Daoud, 2007, *"A New Algorithm for Cluster Initialization"*, World Academy of Science, Engineering and Technology 4.

Murtagh F., 1983. *"A survey of recent advances in hierarchical clustering algorithms"*. The Computer Journal. Volume 26, Number 4, pp. 354-359.

Nielsen F. and Bhatia R., 2013, *"Matrix Information Geometry".* New York: Springer.

Payam H. ve Mieczyslaw L. Owoc, 2011, *"Data mining Research Trends in Computerized Patient Records"*, Proceedings of the Federated Conference on Computer Science and Information Systems, pp.133-139.

Peters M. and Zaki M.J., 2004, *"CLICK: Clustering Categorical Data using K-partite Maximal Cliques"*, Computer Science Department, Rensselaer Polytechnic Institute, Troy NY 12180.

Quinn G.P., Keough M.J. , 2004, *"Experimental design and data analysis for biologists".* Cambridge University Press, Reprinted.

Rezanková H., *"Cluster analysis and categorical data"*, STATISTICA – p. 216-232

KAYNAKLAR DİZİNİ (DEVAMI)

Rezanková H., 2014 *"Cluster Analysis of Economic Data"*, STATISTICA 2014, 94(1) – p. 73-86

Sajama S., Orlitsky A., 2005, *"Estimating and computing density based distance metrics"* in Proc. 22nd Int. Conf. Machine Learning, 2005, pp. 760–767.

Sally G., James A.D. ve Bryce V., 2010, *"A Study of Clustered Data and Approaches to Its Analysis"*, The Journal of Neuroscience, August 11, 2010 • 30(32), pages 10601–10608, 2010.

Scholkopf B. and Smola A. J., 2001, *"Learning with Kernels, Support Vector Machines, Regularization, Optimization, and Beyond"*. Cambridge, MA: MIT Press.

Shalev-Shwartz S. and Ben-David S., 2014, *"Understanding Machine Learning: From Theory to Algorithms"*. New York: Cambridge Univ. Press, 2014.

Sibson R., 1973, *"An Optimlly Efficient Algorithm for the Single Link Cluster Method"*, The Comuter Journal, Vol 16, Issue 1.

Somerfield P.J., Clarke K.R., Olsgard F. , 2002, *"A comparison of the power of categorical and correlational tests applied to community ecology data from gradient studies".* Journal of Animal Ecology 71:581-593.

Stevens S.S., 1946, *"On the Theory of Scales of Measurements".* Science 103(2684):677-680.

Šulc Z., Řezanková H, 2014,*"Evaluation Of Selected Approaches To Clustering Categorical Variables"*, Statistics In Transition new series, Autumn 2014, Vol. 15, No. 4, pp. 591–610

Syal R., Kumar V., 2012, *"Innovative Modified K-Mode Clustering Algorithm"*, International Journal of Engineering Research and Applications, ISSN: 2248-9622, Vol. 2, Issue 4, July-August 2012, pp.390-398

Tenenbaum J., Silva V. and Langford J., 2000,*"A global geometric framework for nonlinear dimensionality reduction,"* Science, vol. 290, no. 5500, pp. 2319–2323.

Tripathy B. K. and Ghosh A, 2011, *"SSDR: An Algorithm for Clustering Categorical Data Using Rough Set Theory"*, Pelagia Research Library, Advances in Applied Science Research, 2 (3): 314-326

UC Irvine Machine Learning Repository (http://archive.ics.uci.edu/ml/) (Erişim tarihi Temmuz 2017)

Velleman P.F., Wilkinson L. , 1993, *"Nominal, ordinal, interval and ratio typologies are misleading"*. The American Statistician 47:65-72.

KAYNAKLAR DİZİNİ (DEVAMI)

William H. Day ve Herbert Edelsbrunner. 1984. *"Efficient Algorithms for Agglomerative Hierarchical Clustering Methods"*. Journal of Classification. Volume 1, pp. 1-24.

Xiaowei Xu vd., 1998, *"A Distribution-Based Clustering Algorithm for Mining in Large Spatial Databases"*, ICDE, sayı: 14, 1998, s. 324-331.

Zengyou H., 2006, *"Approximation Algorithms for K-Model Clustering",* CoRR abs/cs/0603120.

Zengyou H., Xiaofe X., Shengchun D., 2005, *"Clustering Mixed Numeric and Categorical Data. A Cluster Ensemble Approach"*, Departament of Computer Science and Engineering, Harbin Institute of Technology, Harbin 150001, P. R. China, (5 Sep 2005)

Zengyou He, Xiaofei Xu, Shengchun Deng, 2008, *"K-ANMI A Mutual Infomation Based Clustering Algorithm for Categorical Data"*, Department of Computer Science and Engineering, Harbin Institute of Technology, Information Fusion 9(2), pages 223-233, 2008.

ÖZGEÇMİŞ

27.08.1988 tarihinde Baküde (Azerbaycan Cümhuriyyeti) doğmuştur. İlk, orta ve lise öğrenimini yine aynı ilde tamamladıktan sonra Azerbaycan Devlet Petrol Akademisi Uygulamalı Matematik Bölümünüde burslu eğitim almaya başlamıştır. Öğrencilik zamanı 2007 senesinden özel eğitim kurumunda yarı zamanlı bilgisayar eğitmenliği ve banka sektöründe yazılım uzmanı görevlerinde çalışmıştır. Bu bölümden 2009 yılında mezun olduğunda aynı ilde Baku Devlet üniversitesi Matemetik Bölümünde Yüksek lisans eğitimi almaya hak kazanmıştır.

Fakat aynı sene askere gitmiş ve 2010 senesinde döndüğünde Baküde kazandığı bölüme devam etmeyip Türkiye Cümhurriyetinin İzmir şehrine gelmiştir. Burada aynı sene Netsis yazılım şirketinde işe başlamıştır.

2012 yılında işle beraber Ege üniversitesi Matematik bölümünde Yüksek Lisans eğitim hakkı kazanmış ve 2014 senesinde tamamlamıştır. 2015 senesinde aynı bölümde Doktora eğitimine başlamıştır.

Yapılan çalışmalar aşağıda sunulmuştur:

Projeler

- TüBİTAK 3001 Projesi, Araştırmacı, Matematiksel Optimizasyon Tabanlı Veri Madenciliği Yöntemleri, 2015.
- Çokboyutlu Veritabanlarında Kümeleme Yöntemleri zerine, BAP Projesi, 2013.

Konferans, Kongre vb.

- Ordin B., Nasibov E., Bulanık Kümelemede Yeni Bir Yaklaşım, Uluslararası Yöneylem Araştırması ve Endüstri Mühendisliği Kongresi 2013, İstanbul.
- Ordin B., Nasibov E., Matematik Tabanlı Kümeleme Algoritmaları üzerine, YEAM 2015 Ulusal Kongresi, Ankara.

Makaleler

Nasibov, E., Ordin, B., An Incremental Fuzzy Algorithm For Data Clustering Problems, Numerical Algebra, Control and Optimization (NACO), 2017 (kabul edildi).

Nasibov, E., Ordin, B., An Incremental Algorithm For Data Clustering Problems In Datasets With Mixed Attributes (hazırlık aşamasında).