İSTANBUL TEKNİK ÜNİVERSİTESİ
ELEKTRİK-ELEKTRONİK FAKÜLTESİ

WEB SAYFASI SINIFLANDIRMA YÖNTEMLERİ ve BENZERLİK ÖLÇÜTLERİ

Bitirme Ödevi

Eser Aygün
040010431

Bölüm: Bilgisayar Mühendisliği
Anabilim Dalı: Bilgisayar Bilimleri

Danışman: Yrd. Doç. Dr. Zehra Çataltepe

Mayıs 2006

İSTANBUL TEKNİK ÜNİVERSİTESİ
ELEKTRİK-ELEKTRONİK FAKÜLTESİ

WEB SAYFASI SINIFLANDIRMA YÖNTEMLERİ ve BENZERLİK ÖLÇÜTLERİ

Bitirme Ödevi

Eser Aygün
040010431

Bölüm: Bilgisayar Mühendisliği
Anabilim Dalı: Bilgisayar Bilimleri

Danışman: Yrd. Doç. Dr. Zehra Çataltepe

Mayıs 2006

Özgünlük Bildirisi

1. Bu çalışmada, başka kaynaklardan yapılan tüm alıntıların, ilgili kaynaklar referans gösterilerek açıkça belirtildiğini,
2. Alıntılar dışındaki bölümlerin, özellikle projenin ana konusunu oluşturan teorik çalışmaların ve yazılım/donanımın benim tarafımdan yapıldığını

bildiririm.

İstanbul, 16 Mayıs 2006

Eser Aygün

WEB SAYFASI SINIFLANDIRMA YÖNTEMLERİ ve BENZERLİK ÖLÇÜTLERİ

(ÖZET)

Son yıllarda bilişim ve iletişim teknolojilerinin hızlı yükselişiyle birlikte sayısı hızla artan web sayfalarının sağladığı bilgiye erişmek giderek zorlaşmaktadır. Bu sorunun üstesinden gelebilmek için, verilen anahtar sözcüklerden yola çıkarak ilgili sayfaları bulmaya çalışan arama motorları ya da web sayfalarını hiyerarşik bir yapı içerisinde, kolayca anlaşılabilecek bir düzende tutmayı amaçlayan dizinler kullanılır. Web sayfası sınıflandırma teknikleri, verilen sayfaları belli kategoriler altında toplayarak otomatik dizin oluşturmayı mümkün hâle getirirler. Bu projenin amacı, mevcut web sayfası sınıflandırma yöntemlerini inceleyip kıyaslamak ve yeni yöntemler önermektir.

Çalışmada, web sayfası sınıflandırma sorununun üstesinden gelmek için ortaya atılmış olan önemli sınıflandırma yöntemleri ve bu yöntemlerde kullanılabilecek farklı benzerlik ölçütleri incelenmiş ve yeni yöntemler önerilmiştir. Ayrıca, incelenen ve önerilen tüm yöntem ve ölçütler MATLAB dilinde gerçeklenmiş ve gerçek hayattan veriler kullanılarak birbirleriyle kıyaslanmıştır. Böylece hangilerinin ele alınan sorun için daha uygun olduğuna karar verilmiş ve önerilen yeni yöntemler ile bir gelişme kaydedilip edilmediği gözlenmiştir.

Özetle denebilir ki; bu çalışmada örüntü tanıma ve web sayfası sınıflandırma alanları için gereken temel kavram ve yöntemlerin açıklanmış olmasının yanı sıra, web sayfası sınıflandırmak için kullanılan yöntemler ve benzerlik ölçütleri kıyaslanmış ve bu yöntemlerden birisi olan Ağırlık Merkeziyle Temsil yönteminin geliştirilmesiyle oluşturulmuş ve hem kuramsal olarak hem de deneysel olarak verimli olduğu gösterilmiş olan yeni bir yöntem ortaya atılmıştır. Son olarak, tüm bunların üzerine atılabilecek adımlardan birkaçına değinilmiş ve web sayfası sınıflandırma konusunun henüz kapanmamış bir konu olduğu sonucuna varılmıştır.

WEB-PAGE CLASSIFICATION METHODS and SIMILARITY METRICS

(SUMMARY)

During recent years, with the fast advance of computer and communication technology, the amount of information on the WWW (World Wide Web) is steeply increased. This situation makes the access to the information harder than ever. There are two methods that are adopted to overcome this problem: Search engines and directories.

A search engine tries to locate a web-page using given keywords. To be able to do this, search engines use keyword indices, which are populated by a automatic web crawler program. However, even large indices are becoming insufficient. One reason of this is that junk pages that pollutes the indices with incorrect entries. Furthermore, because of the increasing number of web-pages, it is becoming more and more a necessity not only to find the relevant pages but also to sort them by their relevance.

The other method is to categorize the web-pages using directories. These hierarchical directories let the users to scan the pages in a top-down manner and find more easily what they are looking for. Though, these directories are created and kept up to date by human efforts. As a result of manual categorization, besides objectivity of directories, the construction and maintenance costs also becomes a problem. So, one can ask "How can we automate the process of web-page categorization?" By collecting the given pages under some specific categories, web-page classification techniques make automatic web-page categorization possible.

This work is on the classification of web-pages after feature extraction. The project consists of the investigation of some important classification methods, similarity metrics and some improvements on them. Also, these methods and metrics were implemented using MATLAB and compared to each other.

There are so many approaches to classify web-pages; statistical, graph-based, artificial neural network-based, etc. But, only the spatial ones are taken into account during this work. These are Nearest Neighbor, K Nearest Neighbor, Nearest Cluster and Centroid-Based methods. In addition to these, three new methods with names Representative Addition Using K Means, Representative Addition Using Subsets and Centroid-Based With Weighted Data, were introduced.

There are also numerous metrics to measure the similarity between two objects. But, four of them are especially important in web-page classification area: Euclid Similarity, Cosine

Similarity, Extended Jaccard Similarity and Pearson Correlation. The exploration of these metrics was another step of the work.

To be able to compare these methods and metrics, it is required to implement them. Because of its high level matrix operation support and visualization capabilities, MATLAB was used during implementation. To permit new classification methods and similarity metrics to be added easily to the system, the code is written modularly.

During the comparison step, many different real world data were needed. So, large amounts of data are collected from different sources. Finally, all the experiments were done using these data and the results were interpreted.

In the deeper steps of the work, first of all, general pattern recognition and web-page classification concepts were defined and formulated mathematically so that the constructed formal system may be used in future works. The classifier and the similarity metric notions and some evaluation tools (confusion matrix and accuracy) were also defined precisely using this formal system, besides the training and application algorithms of classification methods.

Secondly, the four classification methods were defined and they were compared theoretically by the means of computational complexity. The results indicated that all of the methods that are discussed in this work behave nearly the same while training, where the complexities of application algorithms differ: Nearest Neighbor and K Nearest Neighbor methods are asymptotically slower than the others while applying on test data.

Third step was the exploration of the four similarity metrics. Euclid similarity, which is derived from Euclid distance, is sensitive to the scale of the vectors and insensitive to the transposition. On the contrary, cosine similarity is sensitive to the transposition and insensitive to the scale. Extended Jaccard similarity is like a transition between cosine similarity and Euclid similarity, where Pearson correlation has statistical meanings more than geometric ones.

After these investigations, three experiments were followed. First experiment targeted to compare the four existing classification methods and the results indicated that the Centroid-Based method is performed best in spite of its simplicity. The second experiment was done to compare the four similarity metrics. Pearson correlation, extendend Jaccard similarity and cosine metrics are performed well during this experiment, while Euclid similarity had nothing to say. Finally, the third experiment aimed to compare the suggested methods with existing ones and it showed that the newly introduced Centroid-Based With Weighted Data method performed very well.

To sum up; in addition to defining the basic concepts and methods of pattern recognition and web-page classification, the classification methods and similarity metrics were compared with each other and three new methods deriving from Centroid-Based method were introduced and also it was shown both theoretically and empirically that on of these

new methods is more efficient than existing ones in many cases. Finally, some future work over this work was proposed, and it was realized that there are still uncovered parts on the web-page classification subject.

İÇİNDEKİLER

1 GİRİŞ 1
2 PROJENİN TANIMI ve PLANI 2
3 KURAMSAL BİLGİLER 3
3.1. Örüntü Tanıma 3
3.2. Örüntü Tanıma Uygulaması Olarak Web Sayfası Sınıflandırma 4
3.3. Temel Matematik Tanımları 4
3.3.1. Sınıflandırıcı 4
3.3.2. Benzerlik Ölçütü 5
3.3.3. Değerlendirme Araçları 5
4 SINIFLANDIRMA YÖNTEMLERİ 7
4.1. K Orta Algoritması 8
4.2. En Yakın Komşu Yöntemi 9
4.3. En Yakın K Komşu Yöntemi 9
4.4. En Yakın Küme Yöntemi 10
4.5. Ağırlık Merkeziyle Temsil Yöntemi 12
4.6. Yeni Yöntemler 13
4.6.1. K Orta Kullanarak Temsilci Ekleme 13
4.6.2. Alt Kümeleri Kullanarak Temsilci Ekleme 15
4.6.3. Ağırlıklandırılmış Verili Ağırlık Merkeziyle Temsil 15
5 BENZERLİK ÖLÇÜTLERİ 18
5.1. Öklid Benzerliği 18
5.2. Kosinüs Benzerliği 19
5.3. Genişletilmiş Jaccard Benzerliği 20
5.4. Pearson Korelasyonu 21
6 GERÇEKLEME 22
6.1. Genel Amaçlı Fonksiyonlar 22
6.2. Değerlendirme Fonksiyonları 22
6.3. Veri Yükleme Fonksiyonları 22
6.4. Veri Sağlayıcılar 23
6.5. Benzerlik Ölçütleri 23
6.6. Eğitim Fonksiyonları 23
6.7. Uygulama Fonksiyonları 23
7 DENEYSEL SONUÇLAR 25
7.1. Kullanılan Veriler 25
7.1.1. Yahoo! News 25
7.1.2. Reuters-21578 26
7.1.3. TREC 26

7.2. Kıyaslamalar 26
7.2.1. Deney 1: Mevcut Yöntemlerin Kıyaslanması 27
7.2.2. Deney 2: Benzerlik Ölçütlerinin Kıyaslanması 28
7.2.3. Deney 3: Yeni Yöntemlerin Kıyaslanması 29
8 SONUÇ ve ÖNERİLER 30
9 KAYNAKLAR 33

1 GİRİŞ

Son yıllarda bilişim ve iletişim teknolojilerinin hızlı yükselişiyle birlikte WWW (World Wide Web - Dünya Çapında Ağ) üzerindeki sayfaların sayısı ve bu sayfaların güncellenme sıklığında şaşırtıcı bir artış meydana gelmiştir. 1998 yılında 2.851.000 olan web sitesi sayısı 2002'de 9 milyonu aşmıştır ve bunların 3 milyondan fazlasını halka açık siteler oluşturmaktadır [1]. Bu sitelerin taşıdığı sayfaların milyarları bulduğu düşünülürse sunulan bilgiye erişimi hızlandırma gereksinimi kolayca fark edilebilir.

Hızlı erişimin bir kolunu arama motorları oluşturur. Arama motorları, verilen anahtar sözcüklerden yola çıkarak ilgili sayfaları bulmaya çalışırlar. Bu işlemin verimli yapılabilmesi için kelime indeksleri kullanılır. Kelime indeksleri WWW üzerindeki sayfaların bir uygulama aracılığıyla otomatik olarak taranması ile oluşturulur. [2]'deki verilere göre, 1994'te ilk arama motorlarından biri olan World Wide Web Worm (WWWW) 110.000 sayfayı indekslemişken, 1997'de bu sayı bazı arama motorları için 100 milyona ulaşmıştır. 2004'te ise Google arama motorunun yaklaşık 8,1 milyar sayfayı indeksinde barındırdığı düşünülmektedir [3]. Bu şaşırtıcı rakamlara rağmen, maalesef büyük indeksler aranan veriye ulaşmak için yeterli olmamaktadır. Bunun en önemli nedeni kullanıcıları kandırmak isteyen reklâm amaçlı yalancı içeriklerdir. Ayrıca artan sayfa sayısı nedeniyle sadece doğru sayfaları bulmak yeterli olmamakta, aynı zamanda bu sayfaları ilgililiklerine göre sıralamak gerekmektedir. Bu sorunları aşmak için sayfalar arası bağlantıların incelenmesi ve bağlantı metinlerinin taranması gibi ek yöntemler geliştirilmiştir.

Bir diğer yöntem ise sayfaları dizinlemektir. Bu yöntemde sayfalar hiyerarşik bir yapı içerisinde, kolayca anlaşılabilecek bir düzende tutulurlar; böylece kullanıcılar bu yapıyı aşağıya doğru tarayarak istedikleri bilgilere ulaşabilirler. DMOZ Open Directory Project (ODP) bu dizinlere güzel bir örnek teşkil eder. ODP, 72.535 editörün 7 milyondan fazla sayfayı 590.000 kategori altında toplaması ile oluşmuştur [4]. ODP'de olduğu gibi mevcut yaklaşımlarla dizinleme yöntemi, akla yatkınlığına karşın, büyük oranda insan emeği gerektirir. İnsan eliyle oluşturulan dizinler özneldir ve bu dizinleri oluşturmak ve güncel tutmak yüksek maliyet gerektirir. Bu durum dizinleme işlemini otomatikleştirme düşüncesini akla getirir. Web sayfası sınıflandırma teknikleri, verilen sayfaları belli kategoriler altında toplayarak otomatik dizin oluşturmayı mümkün hâle getirirler. Bu projenin amacı, mevcut web sayfası sınıflandırma yöntemlerini inceleyip kıyaslamak ve yeni yöntemler önermektir.

2 PROJENİN TANIMI ve PLANI

Bu çalışma, web sayfası sınıflandırma sürecinin üç adımından sonuncusu olan, nitelikleri belirlenmiş web sayfalarının kategorilere ayrılması üzerinedir. Çalışmada, bu işin üstesinden gelmek için ortaya atılmış olan önemli sınıflandırma yöntemleri ve bu yöntemlerde kullanılabilecek farklı benzerlik ölçütleri incelenecek ve yeni yöntemler önerilecektir. Ayrıca, incelenen ve önerilen tüm yöntem ve ölçütler MATLAB dilinde gerçeklenecek ve gerçek hayattan veriler kullanılarak birbirleriyle kıyaslanacaktır. Böylece hangilerinin ele alınan sorun için daha uygun olduğuna karar verilecek ve önerilen yeni yöntemler ile bir gelişme kaydedilip edilmediği gözlenecektir.

Web sayfası sınıflandırma için pek çok yaklaşım bulunmaktadır. Kimi yöntemler istatistikî çözümler önerirken, kimileri çizgelerden faydalanır. Ayrıca, yapay sinir ağları da bu alanda kullanılmakta olan bir diğer yaklaşımdır. Ancak, bu çalışmada, nitelik vektörlerine geometrik anlamda yaklaşan uzaysal yöntemler incelenecektir. Bu yöntemler; En Yakın Komşu Yöntemi, En Yakın K Komşu Yöntemi, En Yakın Küme Yöntemi ve Ağırlık Merkeziyle Temsil Yöntemi'dir. Bunlara ek olarak; K Orta Kullanarak Temsilci Ekleme, Alt Kümeleri Kullanarak Temsilci Ekleme ve Ağırlıklandırılmış Verili Ağırlık Merkeziyle Temsil adında üç yeni yöntem önerilecektir.

İki varlığın benzerliğini tanımlamakta kullanılabilecek benzerlik ölçütlerinin sayısı da en az sınıflandırma yöntemleri kadar çoktur. Ancak, web sayfası sınıflandırma sorunuyla ilgili olan ve uzaysal yöntemlerle bağdaşanlara bakıldığında dört adet ölçüt göze çarpar: Öklid Benzerliği (veya genel olarak geometrik uzaklıktan türeyen benzerlikler), Kosinüs Benzerliği, Genişletilmiş Jaccard Benzerliği ve Pearson Korelasyonu. Bu ölçütlerin incelenmesi, çalışmanın diğer bir bölümü oluşturmaktadır.

Çalışmada, kıyaslamaların yapılabilmesi için, incelemelerin yanı sıra gerçeklemeye de ağırlık verilecektir. Bol miktarda matris matematiğine ve görselleştirme olanaklarına ihtiyaç duyulması nedeniyle, gerçekleme için MATLAB dili tercih edilmiştir. Her an yeni yöntem ve ölçüt ekleyebilmek ve eldeki yöntemler ile eldeki ölçütleri rahatça bir araya getirebilmek için modüler bir alt yapı kurulacaktır. Kurulan alt yapının web sayfası sınıflandırma dışındaki örüntü tanıma uygulamalarıyla da kullanılabilir olmasını sağlamak da bir hedeftir.

Kıyaslama aşamasında ise, incelenen yöntemleri ve ölçütleri doğru bir şekilde karşılaştırabilmek için, gerçek hayattan alınmış bol miktarda veriye ihtiyaç vardır. Çalışmanın bir adımı da bu verilerin elde edilmesi ve işlenmeye uygun hâle getirilmesidir. Daha sonra, bu veriler kullanılarak kıyaslamalar yapılacak ve sonuçlar incelenip yorumlanacaktır.

3 KURAMSAL BİLGİLER

Web sayfası sınıflandırma, verilen web sayfalarının konularına göre önceden belirlenmiş konu başlıkları altında sınıflandırılmasını amaçlar. Bir takım varlıkların belirli niteliklerine göre kategorilerine ayrılması işlemi örüntü tanıma adını alır. İlerleyen bölümlerde genel olarak örüntü tanıma kavramı tanımlanacak ve örüntü tanıma alanına ilişkin kavramların ele alınan sorundaki karşılıkları belirlenerek sorun somutlaştırılacaktır.

3.1. Örüntü Tanıma

Örüntü tanıma, verilen varlıkları kategorilere ayırmayı amaçlayan bilgisayar bilimi alanıdır. Bu alan, ele alınan varlıklara ve varlıkların ayrılacağı kategorilere göre farklılaşmış pek çok uygulama alanına sahiptir. Örnek birkaç uygulama alanı Tablo 1'de verilmiştir.

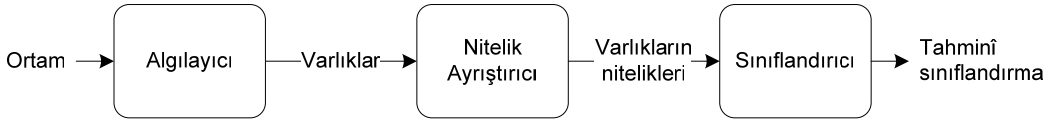
Tablo 1. Örüntü tanımanın örnek uygulama alanlarında varlıkların ve kategorilerin tanımları.

Uygulama alanları	*Varlıklar*	*Kategoriler*
Yazı Tanıma	Yazı görüntüsünden parçalar	Karakterler
Konuşma Tanıma	Konuşma kaydından parçalar	Kelimeler
Belge Sınıflandırma	Belgeler	Konu başlıkları

Örüntü tanıma, kategorilerin önceden belirlenmiş olup olmamasına göre iki kola ayrılır. Kategorilerin önceden belli ve tanımlı olduğu duruma *sınıflama* denir. Örneğin yazı tanımada kategoriler karakterlerdir ve karakterler insanlar tarafından önceden belirlenmişlerdir. Bir yazı tanıma sisteminden beklenen, insanların zaten bildiği karakter tanımının bir modelini barındırmak ya da mekanik öğrenme ile böyle bir modele ulaşmak suretiyle, verilen yazı görüntüsü parçalarındaki karakterleri tıpkı bir insanın yapacağı gibi - ama muhtemelen daha verimli bir şekilde- belirlemesidir.

İkinci kola kümeleme adı verilir. *Kümeleme*, kategorilerin önceden belli olmadığı durumdur. Yazı tanıma örneğinden devam etmek gerekirse; kümeleme, bilinmeyen bir alfabenin karakterlerini belirlemeye karşılık düşer. Bu durumda örüntü tanıma sistemini kuranın kategoriler hakkında sisteme gömebileceği ya da öğretebileceği bir ön bilgisi bulunmadığından, model, verilen varlıklar arasındaki benzerlik ilişkilerine göre kurulur.

Her durumda örüntü tanıma işlemi üç adımda gerçekleşir: (i) varlıkların algılanması, (ii) varlıkların niteliklerinin ayrıştırılması ve (iii) sınıflandırma (bkz. Şekil 1). Birinci adımda, algılayıcı varlıkları algılar ve bu algıyı ham veriye dönüştür. İkinci adımda, nitelik ayrıştırıcı bu ham varlık verilerinden, sınıflandırma için gerekli olan önceden belirlenmiş nitelikleri ayrıştırır. Son adımda, sınıflandırıcı bu nitelikleri kullanarak her varlık için bir kategori tahmininde bulunur.



Şekil 1. Örüntü tanımanın adımları ve birimleri.

3.2. Örüntü Tanıma Uygulaması Olarak Web Sayfası Sınıflandırma

Web sayfası sınıflandırma sorunu bir örüntü tanıma sorunu olarak ele alınacaksa, örüntü tanıma bölümünde bahsedilen kavramların bu sorun için somutlaştırılması gerekir. Tablo 2'de örüntü tanıma kavramlarının web sayfası sınıflandırma sorunundaki karşılıkları görülebilir.

Tablo 2. Örüntü tanıma kavramlarının web sayfası sınıflandırma sorunundaki karşılıkları.

Örüntü tanıma kavramı	*Web sayfası sınıflandırmadaki karşılığı*
Varlıklar	Web sayfaları
Kategoriler	Konu başlıkları
Ortam	WWW
Algılama	WWW üzerinden web sayfalarının çekilmesi
Nitelikler	Terim sıklıkları
Nitelik ayrıştırma	Web sayfalarının metinlerindeki terimlerin sayılması

Terimler, önceden belirlenmiş ve sınıflandırılmak istenen web sayfalarının özünü temsil edeceği varsayılan kelime ya da kelime kökleridir. *Terim sıklığı* ise, bir terimin bir sayfada kaç kere tekrar edildiğinin ölçüsüdür. Terim olarak çoğunlukla, dildeki kelimelerden bağlaç, edat gibi her metinde geçebilecek olanların atılması ve çekimli kelimelerin köklerine indirgenmesiyle elde edilen kelimeler kullanılır. İlerleyen bölümlerde bahsedileceği gibi; bu şekilde elde edilen terimler, sorunu basitleştirmek adına ayrıca bir elemeden geçirilebilir.

3.3. Temel Matematik Tanımları

3.3.1. Sınıflandırıcı

Her bir web sayfasının, terimlerin bu sayfa metnindeki sıklığı ile nitelendirildiği belirtilmişti. Buna dayanarak, terimlerin sayısı d, girdideki sayfa sayısı n ve i. terimin j. sayfadaki sıklığı x_i^j olmak üzere, her bir $j \in \{1,2,\ldots,n\}$ varlığı $\mathbf{x}_j = [x_1^j \quad x_2^j \quad \ldots \quad x_d^j]^T$ biçimindeki bir vektör ile, tüm girdi ise aşağıdaki matris ile ifade edilir:

$$\mathbf{X} = \begin{bmatrix} x_1^1 & x_1^2 & \cdots & x_1^n \\ x_2^1 & x_2^2 & \cdots & x_2^n \\ \vdots & \vdots & \ddots & \vdots \\ x_d^1 & x_d^2 & \cdots & x_d^n \end{bmatrix}.$$

Bu matris, sınıflandırıcının girdisini oluşturur. Sınıflandırıcının çıktısı ise *tahminî sınıflandırma*dır. Bir sınıflandırıcı, kategori sayısı g olmak üzere, her bir j varlığını bir $y_j \in \{1,2,\ldots,g\}$ kategorisine yerleştirir. Buna dayanarak, tahminî sınıflandırma da aşağıdaki vektör ile ifade edilir:

$$\mathbf{Y} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix}.$$

Bu gösterim kullanılarak bir sınıflandırıcı $C : (\mathbf{X} \in \mathrm{R}^{d \times n}) \rightarrow (\mathbf{Y} \in \{1,2,\ldots,g\}^n)$ biçimindeki bir dönüşüm olarak tanımlanır.

3.3.2. Benzerlik Ölçütü

Benzerlik ölçütleri, sınıflandırma sırasında, iki varlığın birbirine ne kadar benzediğine dair fikir veren fonksiyonlardır. İki varlığın benzerliğine, aynı kategoride bulunma ihtimalleri olarak bakılabilir ki; standart bir benzerlik ölçütü de iki nitelik vektörünü 0'dan 1'e kadar bir reel sayıya eşler. Şu hâlde, bir benzerlik ölçütü de $s : ((\mathbf{x}_a, \mathbf{x}_b) \in \mathrm{R}^{d \times 2}) \rightarrow [0,1]$ biçimindeki bir dönüşüm olarak tanımlanabilir.

3.3.3. Değerlendirme Araçları

Bir sınıflandırma işleminin önceden bilinen ve beklenen çıktısına *hedef sınıflandırma* denir. Her bir j varlığının yerleştirilmesi beklenen kategori $t_j \in \{1,2,\ldots,g\}$ olmak üzere hedef sınıflandırma aşağıdaki vektör ile ifade edilir:

$$\mathbf{T} = \begin{bmatrix} t_1 \\ t_2 \\ \vdots \\ t_n \end{bmatrix}.$$

Sık sık kullanılan bir değerlendirme aracı olan *karışıklık matrisi*, h olarak sınıflandırılması beklenirken l olarak sınıflandırılmış olan varlıkların sayısı f_h^l olmak üzere şu şekilde ifade edilir:

$$\mathbf{F} = \begin{bmatrix} f_1^1 & f_1^2 & \cdots & f_1^g \\ f_2^1 & f_2^2 & \cdots & f_2^g \\ \vdots & \vdots & \ddots & \vdots \\ f_g^1 & f_g^2 & \cdots & f_g^g \end{bmatrix}.$$

Matrisin satırları beklenen kategorilerdeki varlıkları gösterirken, sütunları tahmin edilen kategorilerdeki varlıları gösterir. Karışıklık matrisinin bir faydası, birbiriyle karıştırılan kategorileri fark etmeyi kolaylaştırmasıdır [5]. Tablo 3'te, 12 kedi, 12 köpek ve 12 fareden oluşan 36 hayvanlık bir varlık takımının sınıflandırılması sonucu oluşabilecek örnek bir karışıklık matrisi verilmiştir.

Tablo 3. Örnek karışıklık matrisi.

		Tahminî		
		Kedi	**Köpek**	**Fare**
Beklenen	**Kedi**	7	3	2
	Köpek	2	9	1
	Fare	1	0	11

Karışıklık matrisi, bir sınıflandırmanın başarısına ilişkin oldukça fazla bilgi veriyorsa da, karşılaştırılabilir bir değer sunmadığı için kıyaslamalarda kullanılamaz. Bundan dolayı, karışıklık matrisinden, kıyaslamalarda kullanabilecek reel bir değerlendirme terimi üretilmiştir. *Doğruluk* adı verilen bu terim, beklendiği gibi sınıflandırılmış varlık sayısının toplam varlık sayısına oranı olarak tanımlanır ve aşağıdaki gibi ifade edilir:

$$\mathrm{acc}(\mathbf{F}) = \frac{\sum_{h=1}^{g} f_h^h}{n}.$$

Doğruluk, beklendiği gibi sınıflandırılan hiç varlık olmadığı durumda 0, tüm varlıkların beklendiği gibi sınıflandırıldığı durumda 1 değerini alır.

4 SINIFLANDIRMA YÖNTEMLERİ

Bu çalışmanın en önemli aşamalarından birisini web sayfası sınıflandırmada kullanılan sınıflandırma yöntemlerinin incelenmesi teşkil etmektedir. Önceki bölümde tanımlandığı gibi, bir sınıflandırıcı, verilen nitelik vektörlerini kullanarak tahminî sınıflandırma üreten bir fonksiyondur. Sınıflandırıcılar neredeyse her zaman mekanik öğrenme yöntemleriyle oluşturulurlar. Mekanik öğrenmede, sınıflandırıcıya beklenen sınıflandırması bilinen bir varlık kümesine ait nitelik vektörleri verilir. Sınıflandırıcı, bu vektörleri kullanarak sınıflandırma işlemi için genel bir model üretmeye çalışılar. Bir kere bu model üretildikten sonra, sınıflandırıcı uygulama verileriyle çalıştırılabilir.

Çalışmadan ele alınan sınıflandırma yöntemleri uzaysaldır. Yani nitelik vektörlerinin geometrik anlamları olduğunu varsayarlar. Genel yaklaşım, eğitim verisinden bir miktar temsilci seçmek ya da oluşturmak ve uygulama sırasında varlıkları kendilerine en çok benzeyen temsilci ile ilişkili olan sınıfa yerleştirmektir. En Yakın Komşu yöntemi, eğitim verisindeki tüm varlıkları temsilci kabul ederek saklarken, Ağırlık Merkeziyle Temsil yöntemi her sınıf için yalnızca bir temsilci, En Yakın Küme yöntemi ise toplamda k adet temsilci üretir. En Yakın K Komşu yöntemi, eğitim aşamasında En Yakın Komşu yöntemi ile benzerlik gösterse de, uygulama aşamasında farklılaşır: Bu yöntemin uygulanmasında, varlıklar kendilerine en çok benzeyen temsilcinin sınıfına değil, kendilerine en çok benzeyen k temsilci arasında baskın olan sınıfa yerleştirilirler.

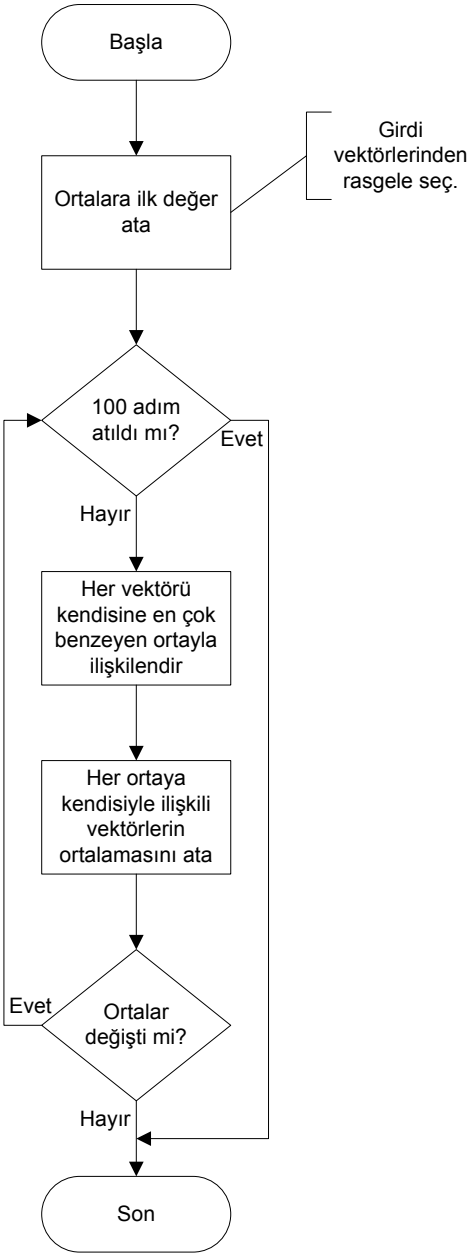
Bu mevcut yöntemlerin yanı sıra, Ağırlık Merkeziyle Temsil (A.M.T.) yönteminden hareketle üç yeni yöntem önerilmiştir. Bunların ilk ikisi, A.M.T. gibi sınıf başına bir temsilci ile başlayarak, doğruluğu arttıracak şekilde yeni temsilciler eklemeyi amaçlar. Üçüncü yöntem ise, doğruluğu arttırmak için A.M.T. ile üretilen temsilcilere yenilerini eklemektense, bu temsilcileri hareket ettirmeyi dener.

Tablo 4'te bahsedilen yöntemlerin basit bir karşılaştırılması görülebilir. İlerleyen bölümlerde her yöntem ayrıntılı olarak incelenecektir.

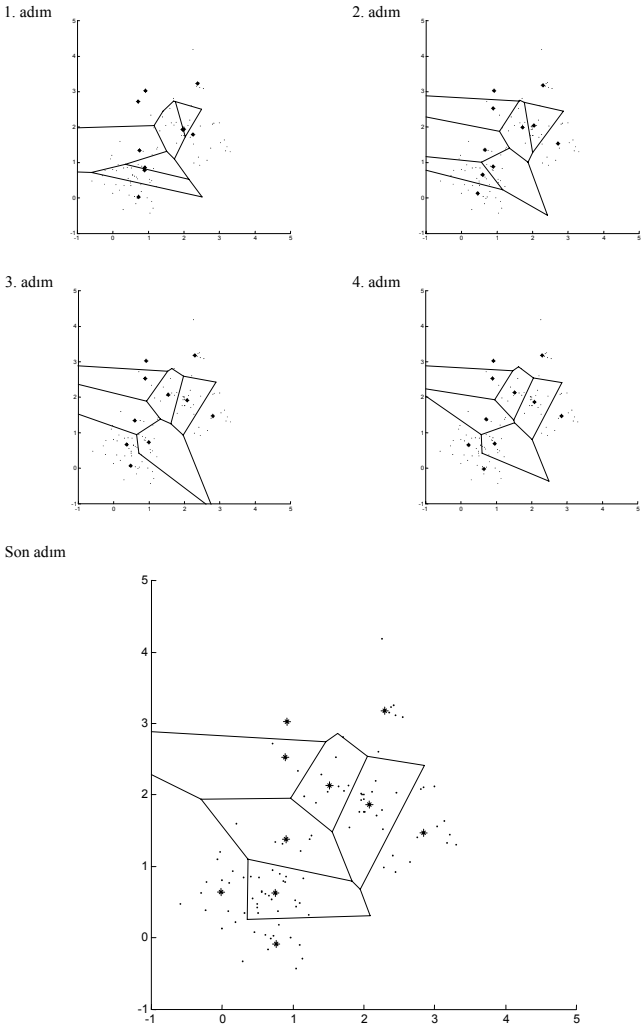
Tablo 4. Ele alınan sınıflandırma yöntemleri.

Yöntem	*Prm'ler*	*Eğitim Algo.*	*Uygulama Algo.*
En Yakın Komşu	-	$O(d \cdot n)$	$O(d \cdot n \cdot \hat{n})$
En Yakın K Komşu	k	$O(d \cdot n)$	$O(d \cdot n \cdot \hat{n} \cdot \log \hat{n})$
En Yakın Küme	k	$O(d \cdot n \cdot k)$	$O(d \cdot n \cdot k)$
Ağırlık Merkeziyle Temsil (A.M.T.)	-	$O(d \cdot n)$	$O(d \cdot n \cdot g)$
Temsilci Ekleme Yöntemleri	m	$O(d \cdot n \cdot m \cdot (g+m))$	$O(d \cdot n \cdot (g+m))$
Ağırlıklandırılmış Verili A.M.T.	m	$O(d \cdot n \cdot m \cdot g)$	$O(d \cdot n \cdot g)$

4.1. K Orta[1] Algoritması



Şekil 2. K Orta algoritmasının akış diyagramı.



Şekil 3. K Orta algoritması: $d = 2$, $n = 100$, $k = 10$. **Düz çizgiler küme sınırlarını temsil eden Voronoi diyagramını göstermektedir.**

K Orta algoritması, verilen vektörler için öyle k adet orta vektörü üretmeyi amaçlar ki; bu k adet vektörün girdi vektörlerine olan benzerliklerinin toplamı en küçük olsun. Bu algoritma, daha çok kümeleme yöntemlerinde kullanılmasına rağmen, bir sınıflama yöntemi olan En Yakın Küme yönteminde de başvurulmaktadır.

[1] İngilizce kaynaklarda *K Means* adıyla anılmaktadır.

4.2. En Yakın Komşu[2] Yöntemi

İlk akla gelebilecek sınıflandırma yöntemlerinden birisi olan En Yakın Komşu yönteminde, eğitim verisi doğrudan saklanır ve uygulama için verilen her varlık eğitim verisindeki en yakın komşunun bulunduğu sınıfa yerleştirilir.

Eğitim

Bütün varlıklar ve bulundukları sınıflar eğitim verisini temsil etmek üzere doğrudan saklanır. Hesap karmaşıklığı $O(d \cdot n)$ 'dir.

$$\text{NEAREST}-\text{NEIGHBOR}-\text{TRAIN}(\mathbf{X},\mathbf{T}) \rightarrow \aleph :$$
$$\aleph \leftarrow (\mathbf{X},\mathbf{T})$$

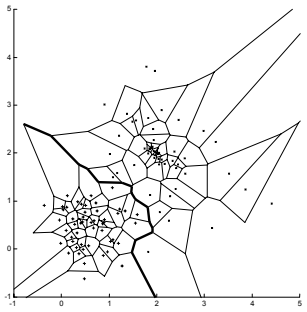
Şekil 4. En Yakın Komşu yöntemi için eğitim algoritması.

Uygulama

Her varlık, eğitim verisi içerisinden kendisine en çok benzeyen varlığın bulunduğu sınıfa yerleştirilir. Hesap karmaşıklığı, $\hat{n}$ eğitim verisinin boyutu olmak üzere, $O(d \cdot n \cdot \hat{n})$ 'dır.

$$\text{NEAREST}-\text{NEIGHBOR}-\text{APPLY}(\aleph,\mathbf{X}) \rightarrow \mathbf{Y} :$$
$$(\hat{\mathbf{X}},\hat{\mathbf{T}}) \leftarrow \aleph$$
$$\forall i \in \{1,2,\ldots,n\} :$$
$$z_i \leftarrow \underset{j\in\{1,2,\ldots,\hat{n}\}}{\arg\max}\, s(\mathbf{x}_i,\hat{\mathbf{x}}_j)$$
$$y_i \leftarrow \hat{t}_{z_i}$$

Şekil 5. En Yakın Komşu yöntemi için uygulama algoritması.



Şekil 6. En Yakın Komşu yöntemi: $d = 2$, $n = 100$, $g = 2$.

4.3. En Yakın K Komşu[3] Yöntemi

Sınıflara ait varlıkların nitelik vektörü uzayında birbirine geçtiği durumda, En Yakın Komşu yöntemi kararsız sonuçlar üretir. Bunun üstesinden gelmek için, en yakın komşunun sınıfına bakmak yerine, en yakın komşular arasında baskın olan sınıfa bakmak önerilmiştir.

Parametre

k : Uygulama sırasında dikkate alınacak komşu sayısı.

---

[2] İngilizce kaynaklarda *Nearest Neighbor* adıyla anılmaktadır.

[3] İngilizce kaynaklarda *K Nearest Neighbor* adıyla anılmaktadır.

Eğitim

Bütün varlıklar ve bulundukları sınıflar eğitim verisini temsil etmek üzere doğrudan saklanır. Hesap karmaşıklığı $O(d \cdot n)$ 'dir.

$$\text{K}-\text{NEAREST}-\text{NEIGHBOR}-\text{TRAIN}(\mathbf{X}, \mathbf{T}, k) \rightarrow \aleph :$$
$$\aleph \leftarrow (\mathbf{X}, \mathbf{T}, k)$$

Şekil 7. En Yakın K Komşu yöntemi için eğitim algoritması.

Uygulama

Her varlık, eğitim verisi içerisinden kendisine en çok benzeyen k varlık arasında baskın olan sınıfa yerleştirilir. Hesap karmaşıklığı $O(d \cdot n \cdot \hat{n} \cdot \log \hat{n})$ 'dır.

$$\text{K}-\text{NEAREST}-\text{NEIGHBOR}-\text{APPLY}(\aleph, \mathbf{X}) \rightarrow \mathbf{Y} :$$
$$(\hat{\mathbf{X}}, \hat{\mathbf{T}}, k) \leftarrow \aleph$$
$$\forall i \in \{1,2,\ldots,n\} :$$
$$Z_i \leftarrow \underset{j=\{1,2,\ldots,\hat{n}\}}{\arg\max} -k\, s(\mathbf{x}_i, \hat{\mathbf{x}}_j)$$
$$y_i \leftarrow \text{dom}(\hat{\mathbf{T}}_{Z_i})$$

Şekil 8. En Yakın K Komşu yöntemi için uygulama algoritması.

4.4. En Yakın Küme[4] Yöntemi

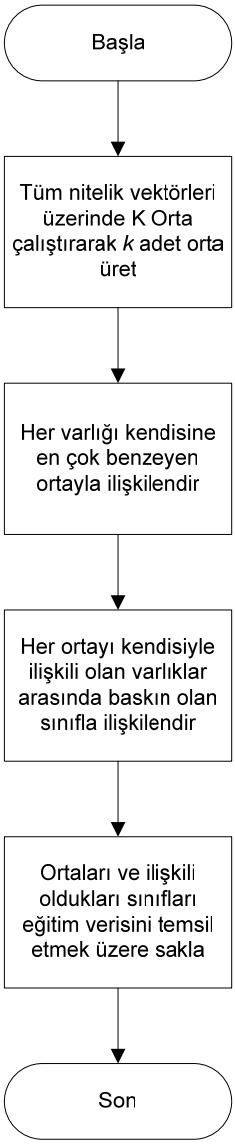
En Yakın Küme yöntemi, K Orta kümeleme yönteminin sınıflandırmadaki uzantısıdır. Eğitim için, öncelikle tüm veriler üzerinde K Orta uygulanarak k adet temsilci seçilir ve ardından bu temsilcilere, çevrelerindeki varlıklar arasında baskın olan sınıfa göre birer sınıf atanır. Temsilciler ve sınıfları belirlendikten sonra, yöntem, En Yakın Komşu yöntemi gibi uygulanır.

Parametre

k : Temsilci sayısı.

[4] İngilizce kaynaklarda *Nearest Cluster* adıyla anılmaktadır.

Eğitim



$$\text{NEAREST}-\text{CLUSTER}-\text{TRAIN}(\mathbf{X},\mathbf{T},k) \rightarrow \aleph :$$

$$\hat{\mathbf{X}} \leftarrow \text{K}-\text{means}(\mathbf{X},k)$$

$$\forall i \in \{1,2,\ldots,n\} : z_i \leftarrow \underset{j\in\{1,2,\ldots,k\}}{\arg\max}\, s(\mathbf{x}_i, \hat{\mathbf{x}}_j)$$

$$\forall j \in \{1,2,\ldots,k\} : \hat{t}_j \leftarrow \text{dom}(\mathbf{T}_{\{h|z_h=j\}})$$ [5]

$$\aleph \leftarrow (\hat{\mathbf{X}},\hat{\mathbf{T}},k)$$

Şekil 9. En Yakın Küme yöntemi için eğitim algoritması.

Hesap karmaşıklığı $O(d \cdot n \cdot k)$ 'dir.

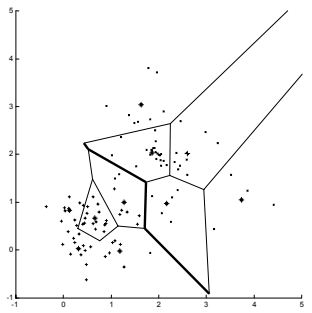
Uygulama

Her varlık, kendisine en çok benzeyen ortayla ilişkili olan sınıfa yerleştirilir. Hesap karmaşıklığı $O(d \cdot n \cdot k)$ 'dir.

[5] $\text{dom(T)}:\ \tau_i$ değerleri arasında baskın olan değer.

$$\text{NEAREST}-\text{CLUSTER}-\text{APPLY}(\aleph, \mathbf{X}) \rightarrow \mathbf{Y}:$$
$$(\hat{\mathbf{X}}, \hat{\mathbf{T}}, k) \leftarrow \aleph$$
$$\forall i \in \{1,2,\ldots,n\}:$$
$$z_i \leftarrow \arg\max_{j \in \{1,2,\ldots,k\}} s(\mathbf{x}_i, \hat{\mathbf{x}}_j)$$
$$y_i \leftarrow \hat{t}_{z_i}$$

Şekil 10. En Yakın Küme yöntemi için uygulama algoritması.



Şekil 11. En Yakın Küme yöntemi: $d = 2$, $n = 100$, $g = 2$, $k = 10$.

4.5. Ağırlık Merkeziyle Temsil[6] Yöntemi

Ağırlık Merkeziyle Temsil yönteminin yaklaşımı çok basittir: Her sınıf için tek bir temsilci tutulur; o da sınıfa ait varlıkların ağırlık merkezidir. Bu basit işleyişine karşın, deneysel sonuçlarda görüleceği gibi, doğruluk açısından diğer yöntemlerden geri kalır yanı yoktur.

Eğitim

Her sınıf için bir ağırlık merkezi hesaplanır. Bu ağırlık merkezleri ve ilişkili oldukları sınıflar eğitim verisini temsil etmek üzere saklanır. Ağırlık merkezleri, ilişkili oldukları sınıfa ait varlıkların nitelik vektörlerinin ortalamasıdır. Hesap karmaşıklığı $O(d \cdot n)$'dir.

$$\text{CENTROID}-\text{TRAIN}(\mathbf{X}, \mathbf{T}) \rightarrow \aleph:$$
$$\forall j \in \{1,2,\ldots g\}:$$
$$\hat{\mathbf{x}}_j \leftarrow \frac{1}{|C_j|} \sum_{i \in C_j} \mathbf{x}_i, \quad C_j = \{h \mid t_h = j\}$$
$$\hat{t}_j \leftarrow j$$
$$\aleph \leftarrow (\hat{\mathbf{X}}, \hat{\mathbf{T}})$$

Şekil 12. Ağırlık Merkeziyle Temsil yöntemi için eğitim algoritması.

Uygulama

Her varlık, kendisine en çok benzeyen ağırlık merkezi ile ilişkili olan sınıfa yerleştirilir. Hesap karmaşıklığı $O(d \cdot n \cdot g)$'dir.

[6] İngilizce kaynaklarda *Centroid-Based* adıyla anılmaktadır.

$$\text{CENTROID}-\text{APPLY}(\aleph, \mathbf{X}) \rightarrow \mathbf{Y}:$$

$$(\hat{\mathbf{X}}, \hat{\mathbf{T}}) \leftarrow \aleph$$

$$\forall i \in \{1,2,\ldots,n\}:$$

$$z_i \leftarrow \underset{j \in \{1,2,\ldots,g\}}{\arg\max}\, s(\mathbf{x}_i, \hat{\mathbf{x}}_j)$$

$$y_i \leftarrow \hat{t}_{z_i}$$

Şekil 13. Ağırlık Merkeziyle Temsil yöntemi için uygulama algoritması.

4.6. Yeni Yöntemler

4.6.1. K Orta Kullanarak Temsilci Ekleme

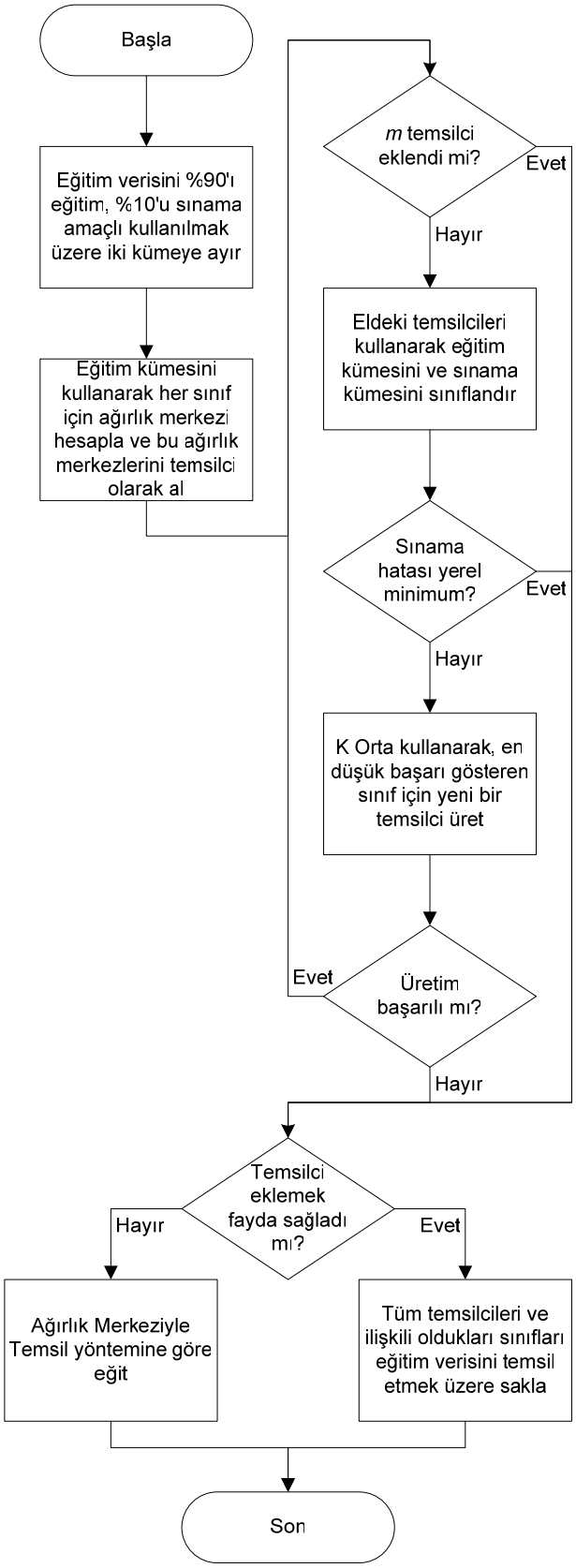
Ağırlık Merkeziyle Temsil yöntemi, sade olmasının yanında, çok az temsilci vektör kullanmasına rağmen (tam olarak sınıf sayısı kadar) gayet başarılı sonuçlar vermektedir. Bu durum, temsilci sayısının bir miktar arttırılmasıyla sınıflandırma doğruluğunun daha da yükseltilebileceği fikrini uyandırır.

Temsilci sayısının adımlı olarak arttırılması durumunda, yeni temsilcinin hangi sınıfa ekleneceğine karar vermek gerekir. Bunun için her adımda öncelikle eldeki temsilciler ile bir sınıflandırma yapılır ve sınıflandırma hatası en yüksek olan sınıf yeni temsilci eklenmek üzere seçilir. Yeni temsilcinin konumuna karar vermek için ise, bu sınıfın varlıkları üzerinde, sınıfın eski temsilci sayısı r olmak üzere, $k = r + 1$ için K Orta algoritması çalıştırılır. Algoritmanın oluşturduğu $r + 1$ adet orta, eski r adet temsilcinin yerine saklanır. Sınama hatasında bir yerel minimuma ulaşılana ya da m temsilcinin tamamı eklenene kadar bu sınıflandırma-ekleme işlemleri devam eder.

Parametreler

m : Eklenecek temsilci sayısı için üst değer.

Eğitim



Şekil 14. K Orta Kullanarak Temsilci Ekleme yöntemi için eğitim algoritması.

Uygulama

Her varlık, kendisine en çok benzeyen temsilci ile ilişkili olan sınıfa yerleştirilir.

4.6.2. Alt Kümeleri Kullanarak Temsilci Ekleme

K Orta Kullanarak Temsilci Ekleme gibi çalışır. Farklı olarak, yeni temsilciler sınıflar için K Orta çalıştırılarak üretilmez. Bunun yerine, her adımda yapılan sınıflandırmadan sonra, yanlış sınıflandırılmış alt kümelerin en kalabalık olanında bulunan varlıkların ağırlık merkezi temsilci olarak eklenir. Yanlış sınıflandırılmış alt kümelerin en kalabalığı, karışıklık matrisinin köşegen haricinde kalan hücrelerinin en büyüğüdür.

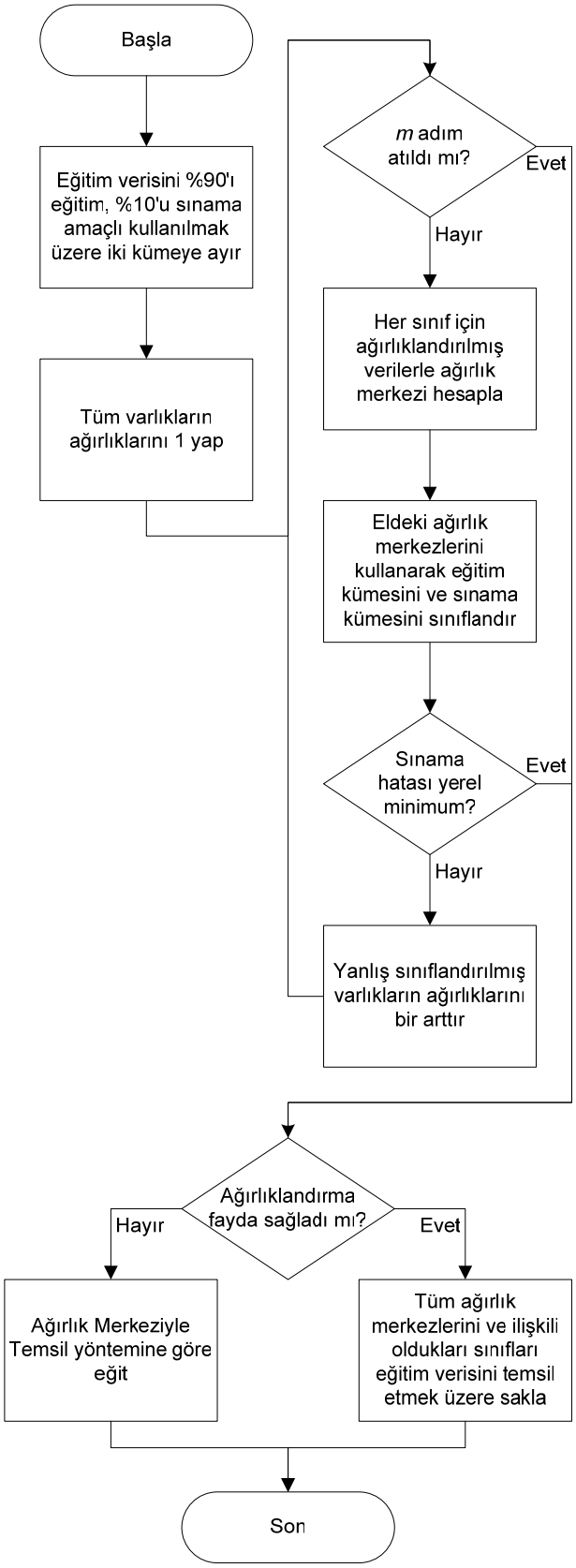
4.6.3. Ağırlıklandırılmış Verili Ağırlık Merkeziyle Temsil

A.M.T. yönteminin geliştirilmesi için akla gelen ikinci bir yol da, ağırlık merkezlerinin sınıflandırma doğrulunu arttıracak şekilde oynatılmasıdır. Bunun için çalışmada önerilen yöntem, eğitim verilerini ağırlıklandırmak ve yanlış sınıflandırılanların ağırlıklarının arttırılmasıyla, ağırlık merkezlerinin onlara daha benzer bir duruma gelmesini sağlamaktır.

Parametreler

m : Adım sayısı için üst değer.

Eğitim



Şekil 15. Ağırlıklandırılmış Verili Ağırlık Merkeziyle Temsil yöntemi için eğitim algoritması.

Klasik A.M.T.'den farklı olarak, bu yöntemde, ağırlık merkezleri ağırlıklandırılmış veri ile hesaplanır. Ağırlıklandırılmış verili ağırlık merkezi hesabı, i. varlığın ağırlığı w_i olmak üzere, aşağıdaki formüle göre yapılır:

$$\widetilde{\mathbf{x}}_C = \frac{1}{\sum_{i \in C} w_i} \sum_{i \in C} w_i \mathbf{x}_i \ .$$

Ağırlıkların değiştirilmesinde ise son derece basit bir höristik kullanılmaktadır: Yanlış sınıflandırılmış olan varlıkların ağırlıkları bir birim arttırılır. Bundaki amaç, ağırlık merkezlerinin yanlış sınıflandırılmış varlıklarla olan benzerliklerini arttırarak, bu varlıkların doğru sınıflandırılması ihtimalini yükseltmektir.

Uygulama

Ağırlık Merkeziyle Temsil yöntemi gibi uygulanır.

5 BENZERLİK ÖLÇÜTLERİ

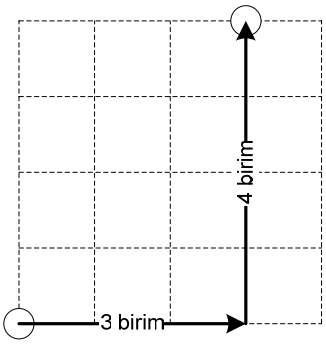
Kuramsal Bilgiler bölümünde tanımlandığı üzere, benzerlik ölçütleri, yapılmak istenen sınıflandırmanın doğası açısından iki varlığın birbirine ne kadar benzediğine, diğer bir değişle, bu sınıflandırmaya göre aynı kategoride bulunma ihtimallerine işaret eder. Ele alınan, varlıkların nitelik vektörleri ile ifade edildiği durumda, ölçütler de vektör işlemleri olarak tanımlanırlar. İlerleyen bölümlerde, web sayfası sınıflandırma için önemli olan bazı benzerlik ölçütleri incelenecektir.

5.1. Öklid Benzerliği

Geometri sorunları için standart uzaklık ölçütleri, Minkowski uzaklıklarıdır:

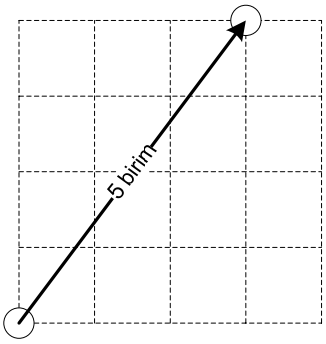
$$D_p(\mathbf{x}_a, \mathbf{x}_b) = \left(\sum_{i=1}^{d} \left| x_{a,i}, x_{b,i} \right|^p \right)^{1/p}.$$

$p = 1$ için Minkowski uzaklığı, Manhattan uzaklığı adını alır. Manhattan uzaklığı, bir kentin birbirini dik kesen caddelerinde hareket ederek alınması gereken yola benzetilebilir.



Şekil 16. Manhattan uzaklığı: $3 + 4 = 7$ **birim.**

$p = 2$ alındığında ise, en çok bilinen geometrik uzaklık ölçütü olan Öklid uzaklığına ulaşılır. Öklid uzaklığı, klasik evrende iki nokta arasındaki en kısa mesafeye denk düşer. Bu yüzden, özellikle görüntü sınıflandırma ile ilgili alanlarda sıklıkla kullanılır.



Şekil 17. Öklid uzaklığı: $(3^2 + 4^2)^{1/2} = 5$ **birim.**

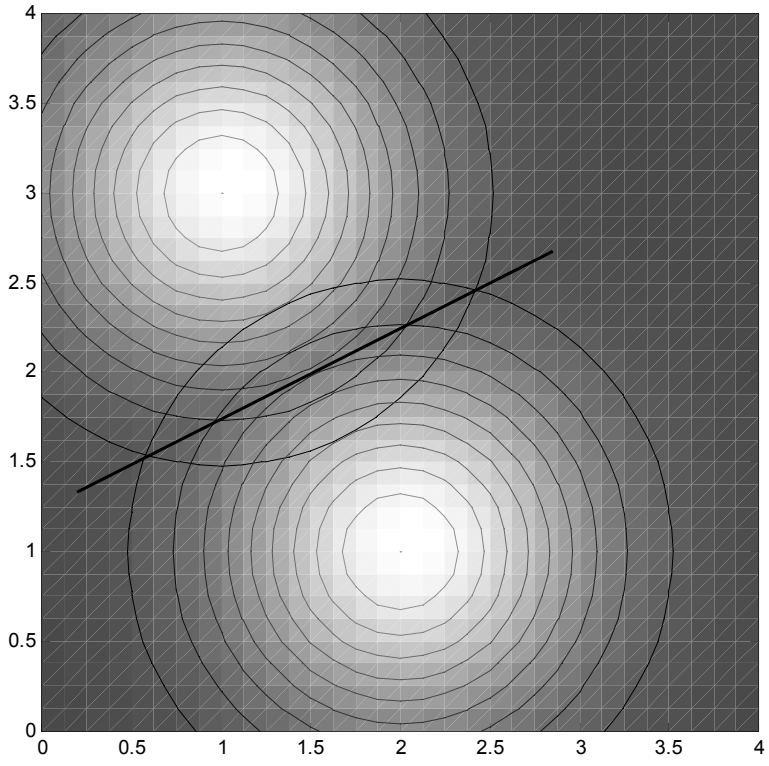
$[0,\infty)$ aralığında değer alan Öklid uzaklığı d, [0,1] aralığında değer alan ve d azaldıkça artan s'ye $s = e^{-d^2}$ bağıntısı ile bağlanabilir [6]. Buradan hareketle, Öklid benzerliği şu şekilde tanımlanır:

$$s^{(\ddot{O})}(\mathbf{x}_a, \mathbf{x}_b) = e^{-\|\mathbf{x}_a - \mathbf{x}_b\|_2^2}.$$

Öklid benzerliği, iki vektörün aynı miktarda ötelenmesi durumunda değerini korur, yani ötelemeye karşı duyarsızdır:

$$\begin{aligned} s^{(\ddot{O})}(\mathbf{x}_a + \mathbf{t}, \mathbf{x}_b + \mathbf{t}) &= e^{-\|(\mathbf{x}_a + \mathbf{t}) - (\mathbf{x}_b + \mathbf{t})\|_2^2} \\ &= e^{-\|(\mathbf{x}_a - \mathbf{x}_b) + (\mathbf{t} - \mathbf{t})\|_2^2} \\ &= e^{-\|\mathbf{x}_a - \mathbf{x}_b\|_2^2} \\ &= s^{(\ddot{O})}(\mathbf{x}_a, \mathbf{x}_b). \end{aligned}$$

İki vektör aynı miktarda ölçeklendiğinde ise Öklid benzerliğinin değeri değişir, yani Öklid benzerliği ölçeklemeye karşı duyarlıdır.



Şekil 18. Öklid benzerliği: Eş benzerlik eğrileri eş merkezli çemberlerdir.

5.2. Kosinüs Benzerliği

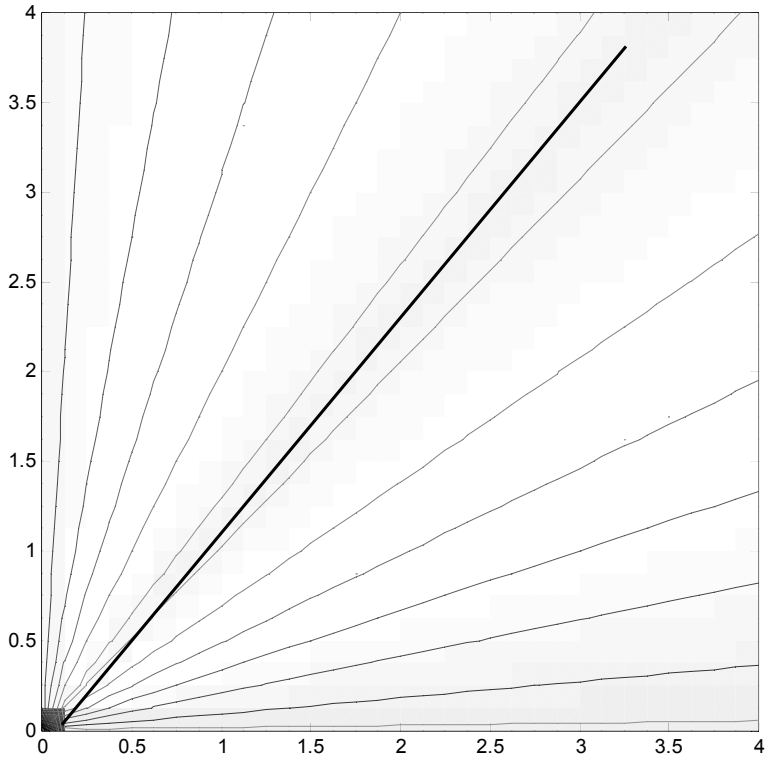
Bir başka benzerlik ölçütü de kosinüs benzerliğidir. Kosinüs benzerliği, iki vektör arasındaki açının kosinüsü olarak tanımlanır:

$$s^{(K)}(\mathbf{x}_a, \mathbf{x}_b) = \frac{\mathbf{x}_a^T \mathbf{x}_b}{\|\mathbf{x}_a\|_2 \|\mathbf{x}_b\|_2} .$$

Kosinüs benzerliği, ölçeklemeye karşı duyarsızdır:

$$\begin{aligned} s^{(K)}(f\mathbf{x}_a, f\mathbf{x}_b) &= \frac{(f\mathbf{x}_a)^T (f\mathbf{x}_b)}{\|f\mathbf{x}_a\|_2 \|f\mathbf{x}_b\|_2} \\ &= \frac{f^2 (\mathbf{x}_a^T \mathbf{x}_b)}{f^2 \|\mathbf{x}_a\|_2 \|\mathbf{x}_b\|_2} \\ &= \frac{\mathbf{x}_a^T \mathbf{x}_b}{\|\mathbf{x}_a\|_2 \|\mathbf{x}_b\|_2} \\ &= s^{(K)}(\mathbf{x}_a, \mathbf{x}_b). \end{aligned}$$

Bu özelliği kosinüs benzerliğini belge sınıflandırma için uygun hâle getirir. Çünkü belgelerin içerdikleri terimlerin sıklıklarının mutlak değerlerinden çok, göreli değerlerini ön plana çıkarır; bu sayede, aynı terimleri toplamda farklı sayıda ama aynı oranda içeren belgeleri benzer olarak değerlendirir.



Şekil 19. Kosinüs benzerliği: Eş benzerlik eğrileri orijinden geçen doğrulardır.

5.3. Genişletilmiş Jaccard Benzerliği

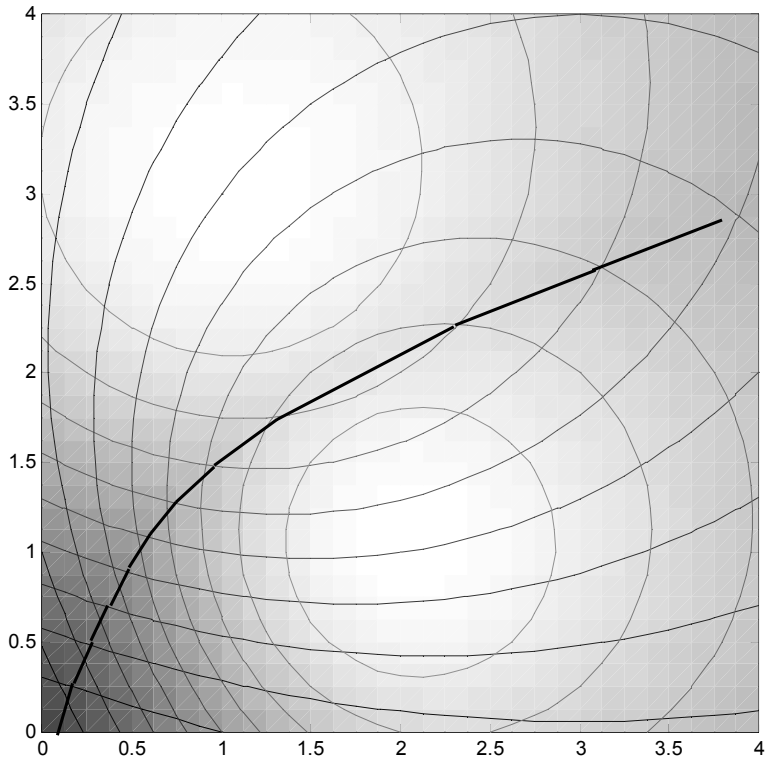
Klasik Jaccard benzerliği, ikili nitelik[7] vektörleri için, hem $\mathbf{x}_a$'da hem $\mathbf{x}_b$'de olan niteliklerin sayısının $\mathbf{x}_a$'da veya $\mathbf{x}_b$'de olan niteliklerin sayısına oranı olarak

[7] Var ve yok gibi, olası iki değeri olan nitelikler.

tanımlanmıştır ($|\mathbf{x}_a \text{ VE } \mathbf{x}_b| / |\mathbf{x}_a \text{ VEYA } \mathbf{x}_b|$). Genişletilmiş Jaccard benzerliği ise, bu benzerliğin sürekli nitelik vektörlerine uyarlanması ile elde edilmiştir [6]:

$$s^{(J)}(\mathbf{x}_a, \mathbf{x}_b) = \frac{\mathbf{x}_a^T \mathbf{x}_b}{\|\mathbf{x}_a\|_2^2 + \|\mathbf{x}_b\|_2^2 - \mathbf{x}_a^T \mathbf{x}_b}.$$

Genişletilmiş Jaccard benzerliği, hem ötelemeye hem de ölçeklemeye duyarlıdır. Buna karşılık, kosinüs benzerliği ile Öklid benzerliği arasında ilginç bir geçiş özelliği sergiler. Orijine yakın noktalarda kosinüs benzerliği gibi, orijinden sonsuza doğru uzaklaştıkça ise Öklid benzerliği gibi davranır.



Şekil 20. Genişletilmiş Jaccard benzerliği: Eş benzerlik eğrileri eş merkezli olmayan çemberlerdir.

5.4. Pearson Korelasyonu

Pearson korelasyonu, istatistikte iki değişken arasındaki doğrusal ilişkiyi yansıtmak üzere kullanılır. $\bar{x}$, $\mathbf{x}$ vektörünün tüm boyutlardaki değerlerinin ortalamasını belirtmek üzere, $\frac{(\mathbf{x}_a - \bar{x}_a)^T (\mathbf{x}_b - \bar{x}_b)}{\|\mathbf{x}_a - \bar{x}_a\|_2 \|\mathbf{x}_b - \bar{x}_b\|_2}$ şeklinde tanımlanır. $[-1,1]$ aralığında değer üreten bu bağıntı $[0,1]$ aralığına normalleştirilirse aşağıdaki benzerlik ölçütüne ulaşılır [6]:

$$s^{(P)}(\mathbf{x}_a, \mathbf{x}_b) = \frac{1}{2}\left(\frac{(\mathbf{x}_a - \bar{x}_a)^T (\mathbf{x}_b - \bar{x}_b)}{\|\mathbf{x}_a - \bar{x}_a\|_2 \|\mathbf{x}_b - \bar{x}_b\|_2} + 1 \right).$$

6 GERÇEKLEME

Gerçeklemenin tamamı MATLAB dilinde yapılmıştır. Üretilen kütüphanenin daha sonraki çalışmalarda, belki örüntü tanımanın farklı uygulama alanlarında kullanılabilmesi için olabildiğince soyut bir yapı kurulmuştur. Ayrıca kodun sadeliğinin ve okunabilirliğinin arttırılması için çoğu yerde MATLAB'a ait çok biçimli veri yapıları ve fonksiyon değişkenleri gibi özelliklerden faydalanılmış; böylece, tasarımda modülerlik sağlanmıştır.

İlerleyen alt bölümlerde, yazılan fonksiyonlar hakkında kısa bilgiler verilecektir.

6.1. Genel Amaçlı Fonksiyonlar

D = classdist(T, nclass): Verilen sınıflandırmadaki sınıf dağılımını verir.

datastat(): Yüklü olan veriye ilişkin sayısal değerler verir.

I = filterbyfreq(X, fmin, fmax): Verilen varlık-nitelik matrisindeki nitelikleri ortalama sıklıklarına göre filtreler.

IDF = idf(X): Ters belge sıklıklarını hesaplar: $IDF_i = \log(n/DF_i)$.

S = mutualsim(Xa, Xb, metric): Verilen varlık-nitelik matrislerindeki varlıklar arasındaki benzerlikleri verilen ölçüte göre hesaplar: $S_i^j = s^{(metric)}(\mathbf{x}_{a,i}, \mathbf{x}_{b,i})$.

[Itrain Itest Ieval] = partition(T, nclass, ptest, peval): Bir veriyi eğitim, sınama ve gerekiyorsa değerlendirme kümelerine böler. Bölme işlemi sırasında verilen sınıflandırmadaki sınıf dağılımını korumaya çalışır.

Y = tiebreak(X, n): X matrisindeki değerlere çok küçük değerler ekleyerek eşitlik durumlarını bozmaya çalışır. Sıralama ve en küçük/büyük bulma işlemlerinden önce kullanılır.

6.2. Değerlendirme Fonksiyonları

e = accuracy(Y, T): Verilen tahminî sınıflandırmanın doğruluğunu hesaplar.

CM = confmat(Y, T, nclass): Verilen tahminî ve beklenen sınıflandırmalara göre karışıklık matrisini hesaplar.

6.3. Veri Yükleme Fonksiyonları

loaddata(prefix, n): Verilen dosya ön ekine sahip dosyalardan veri yükler ve veri üzerinde gerekli ön işlemleri yapar.

6.4. Veri Sağlayıcılar

[X T nclass] = big(n): Bellekte yüklü olan veriden n varlık ve bu varlıklar için beklenen sınıflandırmayı döner.

[X T nclass] = toy1(n): Rasgele oluşturulmuş, iki boyutlu nitelik vektörlerine sahip, iki sınıflı, n varlığa sahip bir veri döner. Bu oyuncak veri, basit görselleştirmelerde kullanılır.

6.5. Benzerlik Ölçütleri

s = cosine(A, B): Kosinüs benzerliği.

s = correl(A, B): Pearson korelasyonu.

s = euclid(A, B): Öklid benzerliği.

s = jaccard(A, B): Jaccard benzerliği.

6.6. Eğitim Fonksiyonları

net = centroidTrain(Xtrain, Ttrain, nclass, params): Ağırlık Merkeziyle Temsil yöntemi için eğitim fonksiyonu.

net = knnTrain(Xtrain, Ttrain, nclass, params): En Yakın K Komşu yöntemi için eğitim fonksiyonu.

net = ncTrain(Xtrain, Ttrain, nclass, params): En Yakın Küme yöntemi için eğitim fonksiyonu.

net = nnTrain(Xtrain, Ttrain, nclass, params): En Yakın Komşu yöntemi için eğitim fonksiyonu.

net = randclassTrain(Xtrain, Ttrain, nclass, params): Rasgele sınıflandırıcı için eğitim fonksiyonu.

net = kmcentroidTrain(Xtrain, Ttrain, nclass, params): K Orta Kullanarak Temsilci Ekleme yöntemi için eğitim fonksiyonu.

net = sscentroidTrain(Xtrain, Ttrain, nclass, params): Alt Kümeleri Kullanarak Temsilci Ekleme yöntemi için eğitim fonksiyonu.

net = wdcentroidTrain(Xtrain, Ttrain, nclass, params): Ağırlıklandırılmış Verili Ağırlık Merkeziyle Temsil yöntemi için eğitim fonksiyonu.

6.7. Uygulama Fonksiyonları

Y = knn(net, X): En Yakın K Komşu yöntemi için uygulama fonksiyonu.

Y = nn(net, X): En Yakın Komşu yöntemi ve diğer tüm temsilcili yöntemler için uygulama fonksiyonu.

Y = randclass(net, X): Rasgele sınıflandırma için uygulama fonksiyonu.

7 DENEYSEL SONUÇLAR

7.1. Kullanılan Veriler

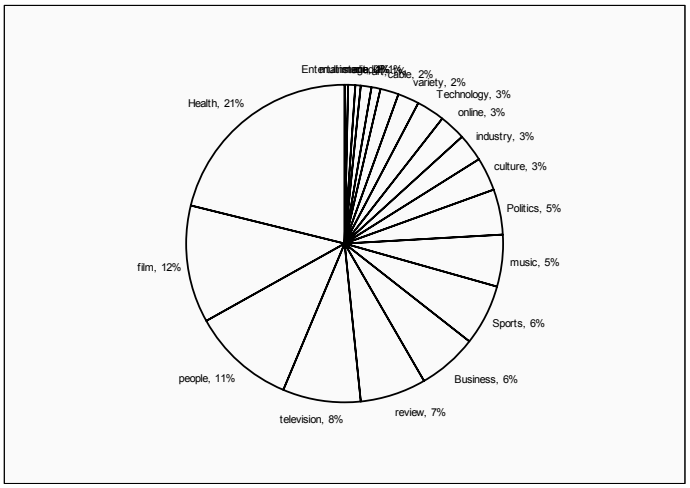
Anlamlı kıyaslamalar yapabilmek için bol miktarda ve gerçekçi veriye ihtiyaç vardır. Daha önceki çalışmaların bir sonucu olarak, web sayfası sınıflandırma deneylerinde kullanılabilecek bol miktarda veriye ulaşılmıştır. Bunlardan Yahoo! News sayfaları [6]'dan, geri kalanlar ise [7]'den elde edilmiştir.

Tablo 5. Çalışmada kullanılan verilerle ilgili sayısal değerler.

Veri Adı	*Belge Sayısı*	*Filtrelenmemiş Terim Sayısı*	*Filtrelenmiş Terim Sayısı*	*Sınıf Sayısı*
Yahoo! News	2340	21389	2903	20
Reuters-21578 re0	1504	2886	1007	13
Reuters-21578 re1	1657	3758	1313	25
TREC tr11	414	6429	3736	9
TREC tr12	313	5804	3557	8
TREC la1	3204	31472	3236	6
TREC la2	3075	31472	3329	6
TREC la12	6279	31472	3273	6

7.1.1. Yahoo! News

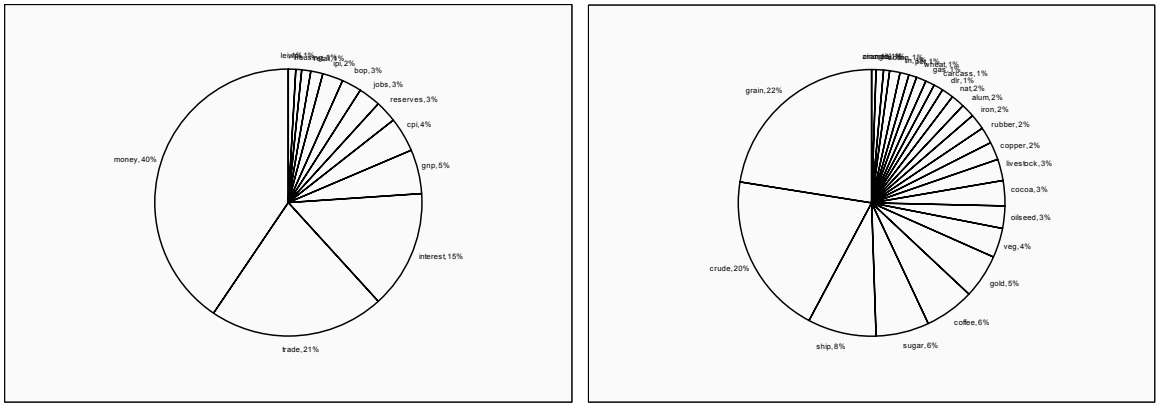
Yahoo! haber kategorileri altından toplanmış web sayfalarından oluşur [6]. 20 adet sınıf barındırır: *Business, Entertainment, art, cable, culture, film, industry, media, multimedia, music, online, people, review, stage, television, variety, Health, Politics, Sports, Technology* (küçük harflerle başlayan kategoriler *Entertainment* kategorisi altında yer almaktadır). Veride 2340 belge ve filtrelenmeden 21389, filtrelendikten sonra 2903 terim bulunmaktadır.



Şekil 21. *ynews* sınıf dağılımı.

7.1.2. Reuters-21578

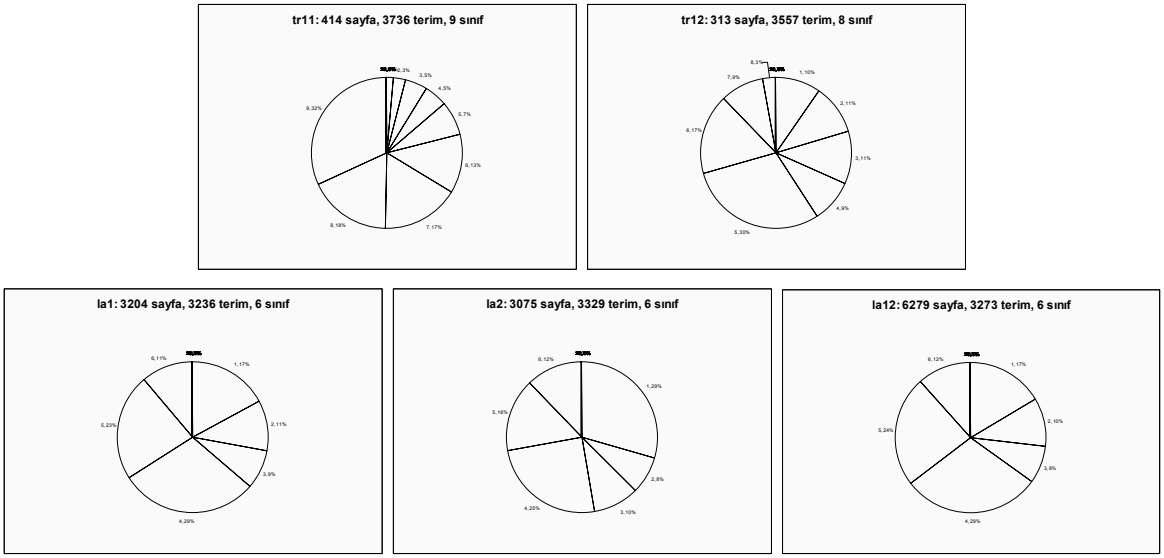
Reuters-21578 Text Categorization Test Collection Distribution 1.0 içerisindeki belgelerden oluşur [7]. İki parçaya ayrılmıştır: re0 ve re1. re0 içerisinde 13 sınıf, 1504 belge, filtrelenmeden 2886, filtrelendikten sonra 1007 terim; re1 içerisinde ise 25 sınıf, 1657 belge, filtrelenmeden 3758, filtrelendikten sonra 1313 terim bulunmaktadır.



Şekil 22. *re0* ve *re1* için sınıf dağılımları.

7.1.3. TREC

TREC (Text REtrivial Conference) için hazırlanmış verilerden oluşur [7]. Beş parçadan oluşmaktadır: tr11, tr12, la1, la2, la12. Bu verilerlerle ilgili ayrıntılı bilgiler Şekil 23'de verilmiştir.



Şekil 23. TREC verileri için sayısal değerler ve sınıf dağılımları.

7.2. Kıyaslamalar

Kıyaslamalarda veriler, %90'ı sınıflandırıcıların eğitiminde, %10'u değerlendirilmesinde kullanılmak üzere iki kümeye ayrılmıştır. Bu ayrım sırasında sınıf dağılımlarının

bozulmamasına dikkat edilmiştir. Sonuçta her sınıflandırıcı için değerlendirme kümesi sınıflandırmasındaki doğruluk ölçülmüştür.

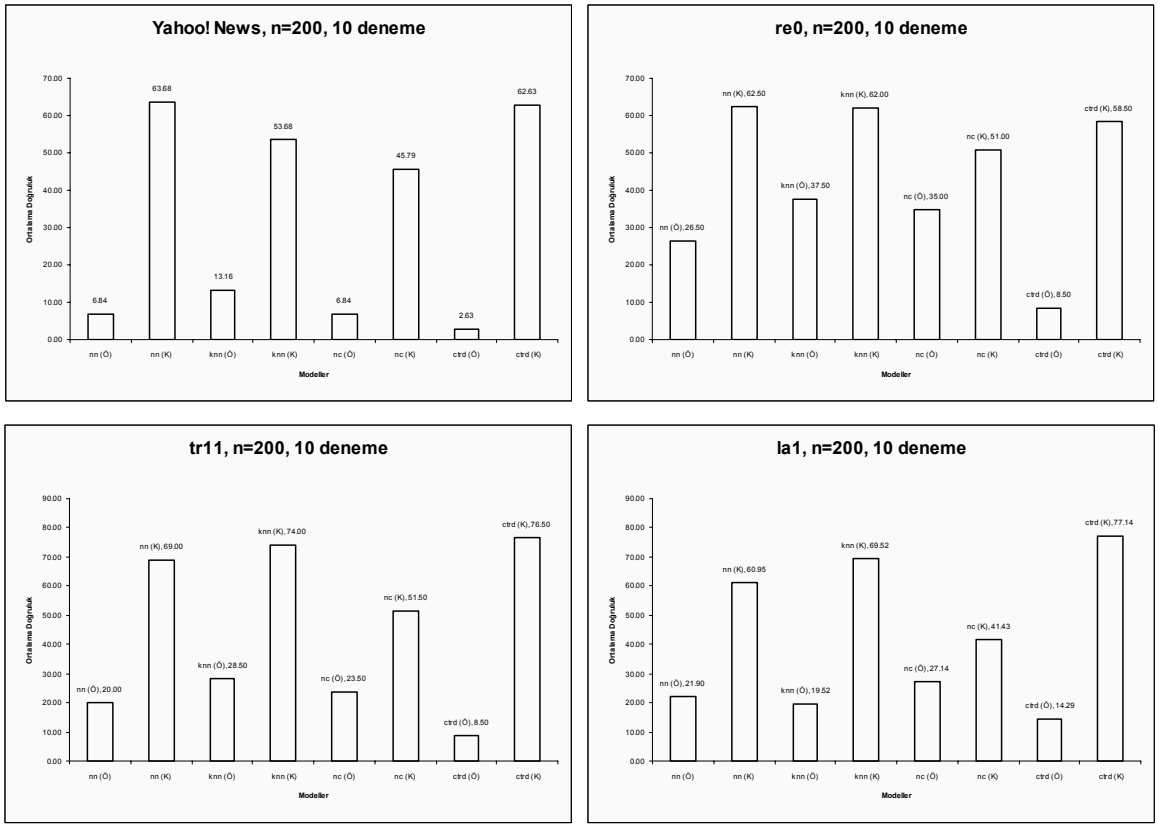
7.2.1. Deney 1: Mevcut Yöntemlerin Kıyaslanması

Birinci deneyde, incelenmiş olan dört mevcut yöntem kıyaslanmıştır. Kullanılmış olan sınıflandırma modelleri Tablo 6'da açıklanmıştır.

Tablo 6. Deney 1 için kullanılan modeller.

Model Adı	*Sınıflandırma Yöntemi*	*Benzerlik Ölçütü*	*Parametreler*
nn (Ö)	En Yakın Komşu	Öklid benzerliği	-
nn (K)		Kosinüs benzerliği	-
knn (Ö)	En Yakın K Komşu	Öklid benzerliği	$k = 10$
knn (K)		Kosinüs benzerliği	$k = 10$
nc (Ö)	En Yakın Küme	Öklid benzerliği	$k = 2g$
nc (K)		Kosinüs benzerliği	$k = 2g$
ctrd (Ö)	Ağırlık Merkeziyle Temsil	Öklid benzerliği	-
ctrd (K)		Kosinüs benzerliği	-

Kıyaslamalar, tüm verilerden 200 sayfa seçilerek 10 deneme yapılması suretiyle gerçekleştirilmiştir. Sonuçlar aşağıda görülebilir.



Şekil 24. Deney 1: Dört veri için sonuçların grafikleri.

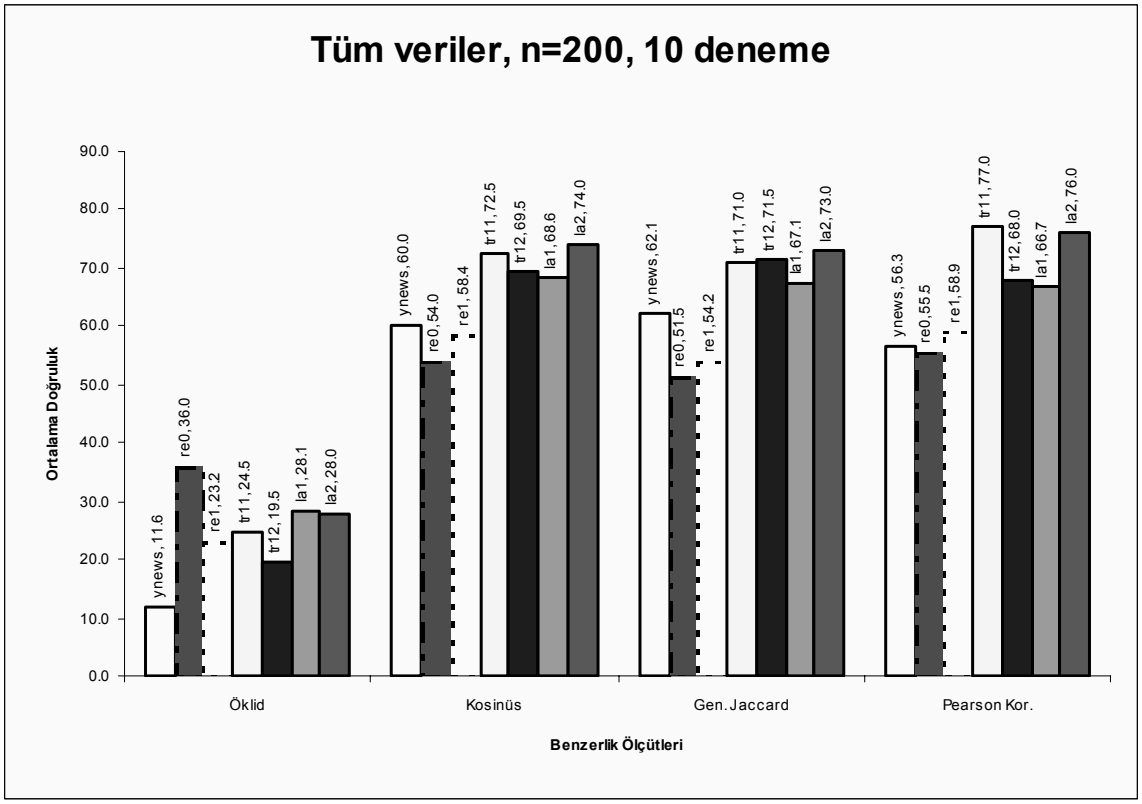
Tablo 7. Deney 1: Tüm veriler için en iyi sonuçlar.

Model	*ynews*	*re0*	*re1*	*tr11*	*tr12*	*la1*	*la2*
nn (K)	**63.7**	**62.5**	**61.1**	69.0	61.0	61.0	66.5
knn (K)	53.7	62.0	59.5	74.0	70.5	69.5	75.5
nc (K)	45.8	51.0	52.1	51.5	49.0	41.4	49.5
ctrd (K)	62.6	58.5	60.5	**76.5**	**74.0**	**77.1**	**79.0**

Sonuçlara göre, Öklid benzerliği kullanılan modeller kosinüs benzerliği kullanılanlara göre çok düşük başarı göstermiş, En Yakın Komşu yöntemi üç veride, Ağırlık Merkeziyle Temsil yöntemiyse dört veride en iyi sonucu vermiştir. Bu noktada, Ağırlık Merkeziyle Temsil yönteminin eğitim ve uygulama algoritmalarının karmaşıklığının diğer yöntemlerinkine göre az olduğunu hatırlamakta fayda vardır.

7.2.2. Deney 2: Benzerlik Ölçütlerinin Kıyaslanması

İkinci deneyde, ortalamada iyi sonuçlar veren En Yakın K Komşu yöntemi kullanılarak dört benzerlik ölçütü kıyaslanmıştır. Kıyaslamalar, tüm verilerden 200 sayfa seçilerek 10 deneme yapılması suretiyle gerçekleştirilmiştir.



Şekil 25. Deney 2: Tüm veriler için sonuçların grafiği.

Tablo 8. Deney 2: Tüm veriler için en iyi sonuçlar.

Model	*ynews*	*re0*	*re1*	*tr11*	*tr12*	*la1*	*la2*
knn (Ö)	11.6	36.0	23.2	24.5	19.5	28.1	28.0
knn (K)	60.0	54.0	58.4	72.5	69.5	**68.6**	74.0
knn (J)	**62.1**	51.5	54.2	71.0	**71.5**	67.1	73.0
knn (P)	56.3	**55.5**	**58.9**	**77.0**	68.0	66.7	**76.0**

Deneyin sonucunda, kosinüs benzerliği bir, genişletilmiş Jaccard benzerliği iki, Pearson korelasyonu ise dört veride en iyi sonucu vermiştir. Öklid benzerliği ise diğerlerine göre bir hayli kötü sonuçlar üretmiştir.

7.2.3. Deney 3: Yeni Yöntemlerin Kıyaslanması

Son deneyde, önerilen yeni yöntemler incelenmiş olan tüm eski yöntemlerle kıyaslanmıştır. Kıyaslamalar, tüm veriler ve verilerdeki tüm sayfalar için yapılmış ve tüm modellerde kosinüs benzerliği kullanılmıştır. Ele alınan modeller Tablo 9'de görülebilir.

Tablo 9. Deney 2 için kullanılan modeller.

Model Adı	*Sınıflandırma Yöntemi*	*Benzerlik Ölçütü*	*Parametreler*
nn (K)	En Yakın Komşu	Kosinüs benzerliği	-
knn (K)	En Yakın K Komşu	Kosinüs benzerliği	$k = 10$
nc (K)	En Yakın Küme	Kosinüs benzerliği	$k = 2g$
ctrd (K)	Ağırlık Merkeziyle Temsil	Kosinüs benzerliği	-
wdatactrd (K)	Ağırlıklandırılmış Verili A.M.T.	Kosinüs benzerliği	$m = 10$

Deneyin sonucunda, önerilen iki temsilci ekleme yönteminin de Ağırlık Merkeziyle Temsil yöntemine üstün gelemediği görülmüş ve bu yöntemlerin kullanıldığı modeller dikkate alınmamıştır. Sonuçlar Tablo 10'da görülebilir.

Tablo 10. Deney 2: Tüm veri kümeleri için en iyi sonuçlar.

Model	*ynews*	*re0*	*re1*	*tr11*	*tr12*	*la1*	*la2*
nn (K)	68.6	**71.2**	**74.6**	71.7	71.0	73.3	75.2
knn (K)	73.6	70.4	73.6	78.3	72.6	77.9	80.4
nc (K)	54.2	49.6	50.2	52.4	43.5	56.7	61.5
ctrd (K)	79.1	68.2	73.4	**80.7**	**79.0**	84.8	86.2
wdatactrd (K)	**79.2**	68.8	73.6	78.5	78.1	**85.6**	**86.7**

Sonuçlara göre, En Yakın Komşu yöntemi iki ve Ağırlık Merkeziyle Temsil yöntemi iki, bu çalışmada önerilmiş olan Ağırlıklandırılmış Verili A.M.T. yöntemi ise üç veride en iyi sonucu vermiştir.

8 SONUÇ ve ÖNERİLER

Yaklaşık üç aylık bir sürede, genel olarak örüntü işleme ve sınıflandırma ile bazı web sayfası sınıflandırma yöntemleri ve bu yöntemlerde kullanılabilecek bazı benzerlik ölçütleri üzerine kapsamlı bir araştırma yapılmış, mevcut yöntemlere ek olarak üç yeni yöntem önerilmiş, tüm bu yöntemlerin uygulanabilmesi için MATLAB dilinde yaklaşık olarak 30 fonksiyondan oluşan bir kütüphane yazılmış ve bu kütüphane kullanılarak kıyaslamalar yapılmıştır.

Kuramsal Bilgiler bölümünde, örüntü tanıma kavramları önce sözle daha sonra matematik diliyle tanımlanarak başka çalışmalarda da kullanılabilecek formel bir sistem inşa edilmiştir. Bu sistem kullanılarak sınıflandırma, benzerlik ölçütü ve değerlendirme araçları kesin bir biçimde tanımlanmış ve daha sonraki aşamalarda değinilen algoritmalar yine bu formel sisteme bağlı kalınarak açıklanmıştır.

Bir sonraki aşamada, sınıflandırma yöntemleri ayrı ayrı ele alınmıştır. Kıyaslamaya geçmeden önce, her yöntemin eğitim ve uygulama algoritmalarının karmaşıklığı kuramsal olarak hesaplanmıştır. Buna göre; eğitim algoritmaları için $O(d \cdot n)$ karmaşıklıkla, En Yakın Komşu, En Yakın K Komşu ve Ağırlık Merkeziyle Temsil yöntemleri ön plana çıkmaktadır. Ancak, En Yakın Küme yöntemi için k'nın, temsilci ekleme yöntemleri ve Ağırlıklandırılmış Verili A.M.T yöntemi için de m'nin ve g'nin, n'den çok küçük olduğu varsayılırsa, eğitim algoritmalarının karmaşıklığı açısından ele alınan yöntemler arasında çok büyük fark olmadığı görülür. Dolayısıyla asıl farkı yaratan, uygulama algoritmalarının karmaşıklığıdır. O hâlde, g ile m'nin asimptotik olarak aynı derecelere sahip olduğu durumda, $O(d \cdot n \cdot k)$ ve $O(d \cdot n \cdot g)$ karmaşıklıklarına sahip olan En Yakın Küme, Ağırlık Merkeziyle Temsil, Ağırlıklandırılmış Verili A.M.T ve temsilci ekleme yöntemlerinin, En Yakın Komşu ve En Yakın K Komşu yöntemlerinden düşük karmaşıklığa sahip olduğu söylenebilir.

Bir diğer aşama, benzerlik ölçütlerinin incelenmesi olmuştur. İncelenen dört farklı ölçütün kimi özelliklerine bakıldığında kıyaslamada alınan sonuçlar kısmen öngörülebilir: Öklid benzerliği oranlamaya duyarlıdır; hâlbuki web sayfası sınıflandırmada nitelik vektörlerinin oranlarının değeri yoktur. Nitekim kosinüs benzerliği bu oranları tamamen ortadan kaldırır ve tüm nitelik vektörlerini birim hiper-küre üzerine normalleştirir. Genişletilmiş Jaccard benzerliği ise, doğal benzerlik ölçütü olan Öklid benzerliği ile web sayfalarında uygulamaya elverişli olan kosinüs benzerliği arasında bir özellik sergiler. Diğer taraftan, geometrik bir yorumu olmamasına rağmen, nitelik vektörlerinin doğrusal korelasyonundan ibaret olan Pearson korelasyonunun da sınıflandırmada başarılı sonuçlar verebileceği görülmüştür.

Deneysel sonuçları yorumlamaya gelmek gerekirse; ilk deney, mevcut yöntemler arasında basitliğine rağmen Ağırlık Merkeziyle Temsil yönteminin çoğunlukla en iyi sonucu

verdiğini göstermiştir. En yakın rakibi olan En Yakın Komşu yönteminin uygulama algoritmasının karmaşıklığı da düşünüldüğünde A.M.T.'den verimsiz olduğu söylenebilir. Dikkati çeken bir başka nokta da En Yakın Küme yönteminin çok başarısız sonuçlar vermiş olmasıdır.

İkinci deney ise benzerlik ölçütlerini kıyaslamak amaçlıdır. Bunun için En Yakın K Komşu yöntemi temel alınmış ve dört benzerlik ölçütü bu yöntemde kullanılarak kıyaslanmıştır. Geometrik bir yaklaşım olmamasına rağmen, Pearson korelasyonu, deneyde kullanılan dört veri için en başarılı ölçüt olmuş; Öklid benzerliği ise bahsedilen özelliklerinden dolayı en kötü sonuçları vermiştir. Genişletilmiş Jaccard benzerliği ise çoğunlukta kosinüs benzerliğine göre kötü sonuçlar vermiş olmasına rağmen iki veri kümesi için en yüksek doğruluğu sağlamıştır. Bu sonuçlar, benzerlik ölçütlerini web sayfası kümeleme yöntemleri üzerinde kıyaslamış ve Öklid benzerliği dışındaki üç ölçütün bu alanda kullanılabilir olduğu sonucuna varmış olan [6] ile uyumluluk göstermektedir.

Son deney yeni önerilen yöntemlerin mevcut yöntemlerle kıyaslanması için yapılmıştır. Tam doğru bir sonuca varabilmek adına verilerin tamamıyla yapılan kıyaslamalar sonucunda, temsilci ekleme temeline dayanan K Orta Kullanarak Temsilci Ekleme ve Alt Küme Kullanarak Temsilci Ekleme yöntemlerinin her seferinde A.M.T.'den daha kötü sonuçlar verdiği görülmüş, bu yüzden bu yöntemler kıyaslamaya dâhil edilmemişlerdir. Bu başarısızlık, eklenen temsilcilerin eğitim verisi için fazla özelleşmiş olmalarına, dolayısıyla bir ezberleme sorunu doğuruyor olmalarına bağlanabilir. Diğer taraftan, Ağırlıklandırılmış Verili A.M.T. yöntemi, beklenildiğinden de yüksek başarı göstererek verilerin üçünde en iyi sonucu vermiştir. Klasik A.M.T. ve En Yakın Komşu yöntemleri iki veri için en yüksek doğruluğu sağlarken, En Yakın K Komşu yine ortalama bir başarı sergilemiş, En Yakın Küme ise tüm verilerde en başarısız yöntem olmuştur.

Bundan sonra, yeni önerilmiş yöntemlerden temsilci eklemeli olanlar üzerinde daha fazla durularak, yeni temsilcileri ezberlemek yerine genelleyecek şekilde eklemeleri sağlanabilir. Ayrıca, temsilci ekleme yöntemi, akla bir de temsilci çıkarma yöntemi olabileceği düşüncesini getirir. Bu sefer, en az sayıda temsilciyle başlayarak yenilerini eklemek yerine, en fazla sayıda temsilciyle başlayıp, fazla olanların atılmasıyla genelleme sağlanabilir. Ağırlıklandırılmış Verili A.M.T. yöntemi için önerilebilecek değişiklikler de bulunmaktadır. Bunlardan ilk akla geleni, varlık ağırlıklarının nasıl değiştirileceğini belirleyen höristik üzerinde oynamaktır. Bir diğeri ise, bu yöntemi bir şekilde temsilci ekleme yöntemleriyle birleştirerek, yeni eklenen temsilcilerin de daha doğru konumlara hareket etmesini sağlamak olabilir.

Öyleyse denebilir ki; bu çalışmada örüntü tanıma ve web sayfası sınıflandırma alanları için gereken temel kavram ve yöntemlerin açıklanmış olmasının yanı sıra, web sayfası sınıflandırmak için kullanılan yöntemler ve benzerlik ölçütleri kıyaslanmış ve bu yöntemlerden birisi olan Ağırlık Merkeziyle Temsil yönteminin geliştirilmesiyle oluşturulmuş ve hem kuramsal olarak hem de deneysel olarak verimli olduğu gösterilmiş

olan yeni bir yöntem ortaya atılmıştır. Son olarak, tüm bunların üzerine atılabilecek adımlardan birkaçına değinilmiş ve web sayfası sınıflandırma konusunun henüz kapanmamış bir konu olduğu sonucuna varılmıştır.

9 KAYNAKLAR

[1] E. T. O'Neill, B. F. Lavoie, R. Bennett. 2003. Trends in the Evolution of the Public Web 1998–2003. *D-Lib Magazine,* 9, 4. (http://www.dlib.org/dlib/april03/lavoie/04lavoie.html)

[2] S. Brin, L. Page. The Anatomy of a Large-Scale Hypertextual Web Search Engine. http://www-db.stanford.edu/~backrub/google.html.

[3] D. Sullivan. Search Engine Size Wars V Erupts. http://blog.searchenginewatch.com/blog/041111-084221.

[4] DMOZ Open Directory Project. http://www.dmoz.org/.

[5] Wikipedia. Confusion Matrix. http://en.wikipedia.org/wiki/Confusion_matrix.

[6] A. Strehl, J. Ghosh, R. Mooney. 2000. Impact of Similarity Measures on Web-page Clustering. AAAI–2000: Workshop of Artificial Intelligence for Web Search, Austin.

[7] E. S. Han, G. Karypis. 2000. Centroid-Based Document Classification: Analysis & Experimental Results. Technical Report TR-00-017, Department of Computer Science, University of Minnesota, Minneapolis.

[8] E. Pekalska, R. P. W. Duin. 2000. Classifiers for Dissimilarity-based Pattern Recognition. *Pattern Recognition, 2000. Proceedings. 15th International Conference on,* 2, pp. 12-16.

[9] A. K. Jain, M. N. Murty, P. J. 1999. Flynn. Data Clustering: A Review. *ACM Computing Surveys (CSUR),* 31, 3, pp. 264–323.

[10] K. Beyer, J. Goldstein, R. Ramakrishnan, U. Shaft. 1999. When is "Nearest Neighbor" Meaningful? *Database Theory - ICDT'99: 7th International Conference, Jerusalem, Israel, January 1999. Proceedings,* p. 217.