Sınıflandırma Yöntemleri Üzerinde Lineer Boyut İndirgeme Yöntemlerinin Karşılaştırılması

Comparison Of Linear Dimensionality Reduction Methods On Classification Methods

Eray YILDIZ[1], Yusuf SEVİM[1]

[1]Elektrik-Elektronik Mühendisliği
Karadeniz Teknik Üniversitesi
erayyildiz@ktu.edu.tr, ysevim@ktu.edu.tr

Özet

Yüksek boyutlu verinin analiz edilmesi ile birçok alanda karşılaşılır. Bu tür verilerin analizinde, daha az sayıda boyut ile çalışmak için yaygın bir şekilde boyut indirgeme yöntemleri kullanılmaktadır. Genellikle lineer boyut indirgeme yöntemleri daha çok tercih edilmektedirler. Bu bildiride popüler lineer boyut indirgeme yöntemleri ve performansları incelenecektir. Bu yöntemler, temel bileşen analizi (TBA), doğrusal diskriminant analizi (DDA), yerellik koruyan iz düşüm (YKİ), komşuluk koruyan gömme (KKG) ve yerellik duyarlı diskriminant analizidir (YDDA).

Abstract

The analysis of high dimensional data is encountered in many areas. In the analysis of that kind of data, dimensionality reduction methods are used to work with fewer dimensions. Generally, linear dimensionality reduction methods are more preferred. In this paper, popular linear dimensionality reduction methods and their performance are investigated. These methods are principal component analysis (PCA), linear discriminant analysis (LDA), locality preserving projection (LPP), neighborhood preserving embedding (NPE) and locality sensitive discriminant analysis (LSDA).

1. Giriş

Yüz tanıma, konuşma sinyali işleme gibi birçok uygulamada, yüksek boyutlu veri ile işlem yapmak gerekmektedir. Bu tür verilerde, büyük veri boyutu işlem yükü ve zaman açısından verinin etkin ve hızlı bir şekilde işlenmesini zorlaştırmaktadır. Bu zorluğun üstesinden gelebilmek için boyut indirgeme yöntemleri yaygın bir şekilde kullanılmaktadır. Boyutu indirgenmiş verinin işlenmesi daha kolaydır.

Boyut indirgeme işlemi ile yapılan, yüksek boyutlu verinin daha düşük boyutlu bir uzayda anlamlı bir şekilde ifade edilmesidir. İstatistik, makine öğrenmesi, veri madenciliği ve ilgili alanlarda kullanılan birçok boyut indirgeme yöntemleri vardır. Bu yöntemler, lineer ve lineer olmayan olmak üzere iki kategoriye ayrıştırılabilir.

Lineer boyut indirgeme yöntemleri, işlem yükü ve çalışma zamanı açısından lineer olmayan yöntemlere göre daha performanslıdır. Bu yüzden, lineer olmayan yöntemlere göre uygulanması daha kolay ve basit oldukları için yüksek boyutu veri analizinde daha çok tercih edilmektedirler.

Bu bildiride lineer boyut indirgeme yöntemlerinin sınıflandırma yöntemleri üzerindeki performansları karşılaştırılmaktadır. Bildirinin ikinci bölümde lineer boyut indirgemenin genel özellikleri ve yöntemleri kısaca tanıtılacaktır. Üçüncü bölümde lineer boyut indirgeme yöntemlerini karşılaştırmak için seçilen veri kümesi tanıtılarak ve bu veri kümesi kullanılarak elde edilen sonuçlar verilecektir. Son bölümünde ise elde edilen sonuçlar değerlendirilecektir.

2. Lineer Boyut İndirgeme Yöntemleri

Lineer boyut indirgeme yöntemleri, yüksek boyutlu verinin daha düşük boyutlu bir uzaya lineer olarak iz düşüm yapılması fikrine dayanır [1]. Bu yöntemlerde amaç, *n* tane örneğe sahip *d* boyutlu veriyi $\boldsymbol{x}_1, \boldsymbol{x}_2, ..., \boldsymbol{x}_n$, örnek sayısı *n*'i sabit tutarak *r* boyutlu $\boldsymbol{y}_1, \boldsymbol{y}_2, ..., \boldsymbol{y}_n$ veriye dönüştüren bir *dxr* boyutlu dönüşüm matrisi *P* bulmaktır. *P* dönüşüm matrisi kullanılarak, yüksek boyutlu veri nesnelerini düşük boyutlu nesnelere aşağıdaki gibi dönüştürülür:

$$\boldsymbol{y}_i = P^T \boldsymbol{x}_i \tag{1}$$

Eğer yüksek boyutlu veriyi, *dxn* veri matrisi $X = [\boldsymbol{x}_1, \boldsymbol{x}_2, ..., \boldsymbol{x}_n]$ şeklinde tanımlarsak, boyutu indirgenmiş veri *rxn* matrisi $Y = [\boldsymbol{y}_1, \boldsymbol{y}_2, ..., \boldsymbol{y}_n]$ olmak üzere, boyut indirgeme işlemi aşağıdaki gibi tanımlanabilir:

$$Y = P^T X \tag{2}$$

Farklı lineer boyut azaltma yöntemleri, farklı *P* dönüşüm matrisi oluştururlar**.** Yöntemler arasındaki fark ise, verinin sahip olduğu özelliklerden hangisinin yansıtılmak ve korunmak istediğine bağlıdır.

2.1. Temel Bileşen Analizi (TBA)

TBA [2,3], istatistik, makine öğrenmesi, veri madenciliği gibi birçok alanda sıkça kullanılan bir lineer boyut azaltma yöntemidir. TBA, veri kümesinin muhtemel ilintili değişkenlerini, temel bileşenler olarak adlandırılan doğrusal ilintisiz değişkenlere dönüştürmek için ortogonal dönüşüm kullanan bir istatistiksel yöntemdir.

TBA yöntemi, *d* boyutlu *n* örnekli bir veri kümesi $X = [\boldsymbol{x}_1, \boldsymbol{x}_2, ..., \boldsymbol{x}_n]$ için, kovaryans matrisinin Σ bulunması ile başlar**:**

$$\Sigma = \frac{1}{n}\sum_{i=1}^{n}(\boldsymbol{x}_i - \bar{\boldsymbol{x}})(\boldsymbol{x}_i - \bar{\boldsymbol{x}})^T = \frac{1}{n}XX^T \quad (3)$$

Buradaki $\bar{\boldsymbol{x}}$ terimi veri nesnelerinin ortalama değerini ifade etmektedir:

$$\bar{\boldsymbol{x}} = \frac{1}{n}\sum_{i=1}^{n}(\boldsymbol{x}_i) \quad (4)$$

Öz değer-öz vektör ayrıştırma yöntemi ile kovaryans matrisin öz değerleri ve öz vektörleri bulunur. En büyük değere sahip *r* tane öz değer seçilir ve *P* dönüşüm matrisi bu öz değerlere karşılık gelen *r* tane öz vektör kullanılarak oluşturulur.

2.2. Doğrusal Diskriminant Analizi (DDA)

DDA, veri nesnelerinin sınıf bilgisini de kullanarak sınıfların en iyi ayrıştığı uzayda yer alan vektörleri bulmaya amaçlayan bir lineer boyut indirgeme yöntemidir [2]. TBA yöntemi ile en sık kullanılan yöntemler arasındadır. DDA yöntemi için, boyut indirgeme işleminde sınıflar arası saçılım matrisi S_B ve sınıf içi saçılım matrisi S_W göz önünde bulundurulur. *c* tane sınıfa sahip bir veri kümesi için $\bar{\boldsymbol{x}}$ veri nesnelerinin ortalama değerini ifade ederse, bu matrisler aşağıdaki şekilde tanımlanır:

$$S_B = \sum_{i=1}^{c}(\bar{\boldsymbol{x}}_i - \bar{\boldsymbol{x}})(\bar{\boldsymbol{x}}_i - \bar{\boldsymbol{x}})^T \quad (5)$$

$$S_W = \sum_{i=1}^{c}\sum_{j=1}^{N_i}(\boldsymbol{x}_{i,j} - \bar{\boldsymbol{x}})(\boldsymbol{x}_{i,j} - \bar{\boldsymbol{x}})^T \quad (6)$$

Buradaki $\bar{\boldsymbol{x}}_i$, i.sınıfın ortalama değerinin ifade etmektedir. *i*.sınıfın *j*.örneği $\boldsymbol{x}_{i,j}$ ile ifade edilmektedir. N_i , *i*.sınıfındaki örnek sayısıdır.

S_B ve S_W matrislerin bulunmasından sonra denklem 7'deki genelleştirilmiş öz değerler probleminin çözülür:

$$S_B\boldsymbol{a} = \lambda S_W\boldsymbol{a} \quad (7)$$

Bu genelleştirilmiş öz değerler probleminin çözümü olan en büyük *r* tane öz değere karşılık gelen *r* tane öz vektör ile *P* dönüşüm matrisi oluşturulur.

2.3. Yerellik Koruyan İz Düşüm (YKİ)

YKİ [4], verinin yerel komşuluk bilgisini en ideal biçimde korumaya çalışan lineer bir yöntemdir. Bu yöntemde, *d* boyutlu *n* örnekli bir veri kümesi $X = [\boldsymbol{x}_1, \boldsymbol{x}_2, ..., \boldsymbol{x}_n]$ için, *n* tane düğümden oluşan komşuluk grafiğinin *G* oluşturulması ile başlar**.** $\boldsymbol{x}_i$ veri örneği *i*.düğüme karşılık gelir. Komşuluk grafiği oluşturulurken bir veri örneğinin ε-komşuluğu veya *k* yakın komşuları kullanılır. Birbirine komşu olan veri örneklerinin düğümlerini birbirine bağlayan bir kenar koyulur. Oluşturulan komşuluk grafiği kullanılarak, bir simetrik *n*x*n* ağırlık matrisi *W* bulunur. Ağırlık matrisinin elemanları W_{ij} iki farklı yol ile bulunabilir:

1. Eğer $\boldsymbol{x}_i$ ve $\boldsymbol{x}_j$ komşuluk grafiğinde birbirine bağlı ise ($t \in \boldsymbol{R}$):

$$W_{ij} = e^{-\frac{\|\boldsymbol{x}_i - \boldsymbol{x}_j\|^2}{t}} \quad (8)$$

2. Eğer $\boldsymbol{x}_i$ ve $\boldsymbol{x}_j$ komşuluk grafiğinde birbirine bağlı ise:

$$W_{ij} = 1 \quad (9)$$

Bu işlemlerden sonra, aşağıdaki genelleştirilmiş öz değerler problemi çözülür:

$$XLX^T\boldsymbol{a} = \lambda XDX^T\boldsymbol{a} \quad (10)$$

L matrisi laplacian matrisi olarak adlandırılır ve $L = D - W$ şeklinde bulunur. *D* matrisi diyagonal bir matristir ve her bir diyagonal eleman, *W* ağırlık matrisinin her bir kolonunun toplamına eşittir:

$$D_{ii} = \sum_j W_{ij} \quad (11)$$

Denklem 10'daki genelleştirilmiş öz değerler probleminin çözümü olan en küçük değere sahip *r* tane öz değerlere karşılık gelen *r* tane öz vektör, *P* dönüşüm matrisini oluşturur.

2.4. Yerellik Duyarlı Diskriminant Analizi (YDDA)

DDA yöntemin en büyük dezavantajı, veri sınıflarının global geometrik yapısını dikkate alarak verideki yerel geometrik yapıyı keşfetmekte başarısız olmasıdır. DDA yönteminin doğasında bulunana bu dezavantajı yok etmek için YDDA [5] yöntemi geliştirilmiştir. YDDA, yerel veri yapısını keşfederek her yerel bölgedeki farklı sınıflara ait veri örnekleri arasındaki marjini en büyük yapan iz düşümü bulan bir yöntemdir.

YDDA yöntemi, *d* boyutlu *n* örnekli bir veri kümesi $X = [\boldsymbol{x}_1, \boldsymbol{x}_2, ..., \boldsymbol{x}_n]$ için, her ikisi de *n* tane düğümden oluşan sınıf içi komşuluk grafiği G_w ve sınıf arası komşuluk grafiğinin G_b oluşturulması ile başlar**.** Bu grafiklerin oluşturulmasında öncelikle her bir $\boldsymbol{x}_i$ veri örneği için, en yakın *k* komşularından oluşan $N(\boldsymbol{x}_i)$ kümesi, $\boldsymbol{x}_i$ veri örneği ile aynı sınıfa ait veri örnekleri kümesi $N_w(\boldsymbol{x}_i)$ ve $\boldsymbol{x}_i$ veri örneği ile aynı sınıfta olmayan veri örnekleri kümesi $N_b(\boldsymbol{x}_i)$ olmak üzere iki alt kümeye bölünür. Bu iki alt kümeleri $N_w(\boldsymbol{x}_i)$ ve $N_b(\boldsymbol{x}_i)$, $l(\boldsymbol{x}_i)$ ifadesi $\boldsymbol{x}_i$ veri noktasının sınıf etiketi olmak üzere:

$$N_w(\boldsymbol{x}_i) = \left\{\boldsymbol{x}_i^j \middle| l(\boldsymbol{x}_i) = l(\boldsymbol{x}_j), 1 \le j \le k\right\} \quad (12)$$

$$N_b(\boldsymbol{x}_i) = \left\{\boldsymbol{x}_i^j \middle| l(\boldsymbol{x}_i) \ne l(\boldsymbol{x}_j), 1 \le j \le k\right\} \quad (13)$$

olarak buluruz. Sınıf içi komşuluk grafiği G_w ve sınıf arası komşuluk grafiği G_b oluşturulurken, $\boldsymbol{x}_i$ veri örneği ve $N_w(\boldsymbol{x}_i)$ ve $N_b(\boldsymbol{x}_i)$ kümelerinde yer alan komşuları için *i*.düğüm ile bu kümelerdeki komşuları arasına kenar koyulur. Daha sonra sınıf içi komşuluk grafiği G_w ve sınıf arası komşuluk grafiğinin G_b ağırlık matrisleri olan W_w ve W_b aşağıdaki denklemlere göre oluşturulur:

$$W_{w,ij} = \begin{cases} \boldsymbol{x}_i \in N_w(\boldsymbol{x}_j) yada\, \boldsymbol{x}_j \in N_w(\boldsymbol{x}_i) ise, 1 \\ yoksa, 0 \end{cases} \quad (14)$$

$$W_{b,ij} = \begin{cases} \boldsymbol{x}_i \in N_b(\boldsymbol{x}_j) yada\, \boldsymbol{x}_j \in N_b(\boldsymbol{x}_i) ise, 1 \\ yoksa, 0 \end{cases} \quad (15)$$

Bu işlemlerden sonra, aşağıdaki genelleştirilmiş öz değerler problemi çözülür:

$$X(\alpha L_b + (1-\alpha) W_w) X^T \boldsymbol{a} = \lambda X D_w X^T \boldsymbol{a} \quad (16)$$

α , 0 ile 1 arasında uygun bir sabit olarak seçilir. L_b , G_b matrisinin laplacian matrisidir ve $L_b = D_b - W_b$ şeklide bulunur. D_w ve D_b matrisleri diyagonal matrislerdir ve her bir diyagonal eleman W_w ve W_b ağırlık matrisinin her bir kolonunun toplamına eşittir:

$$D_{w,ii} = \sum_j W_{w,ij} \quad (17)$$

$$D_{b,ii} = \sum_j W_{b,ij} \quad (18)$$

Denklem 16'daki genelleştirilmiş öz değerler probleminin çözümü olan en büyük değere sahip *r* tane öz değerlere karşılık gelen *r* tane öz vektör *P* dönüşüm matrisini oluşturur.

2.5. Komşuluk Koruyan Gömme (KKG)

KKG [6], verinin yerel komşuluk yapısını korumayı amaçlayan lineer bir yöntemidir. YKİ yöntemine benzer bir yöntemdir, ama amaç fonksiyonları tamamen birbirinden farklıdır [5].

KKG yönteminde, *d* boyutlu *n* örnekli bir veri kümesi $X = [\boldsymbol{x}_1, \boldsymbol{x}_2, ..., \boldsymbol{x}_n]$ için, *n* düğümlü komşuluk grafiğinin *G* oluşturulması ile başlar**.** Komşuluk grafiği *G* oluşturulurken bir veri örneğinin ε-komşuluğu veya *k* en yakın komşuları kullanılır. YKİ yönteminde bahsedilen şekilde oluşturulur. Oluşturulan bu komşuluk grafiği kullanılarak, simetrik *nxn* ağırlık matrisi *W* bulunur. Eğer komşuluk grafiğinde $\boldsymbol{x}_i$ ve $\boldsymbol{x}_j$ komşuluk grafiğinde birbirine bağlı değil ise bu komşuluklar için $W_{ij} = 0$ olur. Eğer komşuluk grafiğinde $\boldsymbol{x}_i$ ve $\boldsymbol{x}_j$ komşuluk grafiğinde birbirine bağlı ise, bu W_{ij} ağırlık değerleri aşağıda kısıtlaması ile verilen amaç fonksiyonunu minimize eden değerler olur:

$$\min \sum_i \left\| \boldsymbol{x}_i - \sum_j W_{ij} \boldsymbol{x}_j \right\| \quad (19)$$

$$\sum_j W_{ij} = 1, j = 1,2,...,m \quad (20)$$

Bu işlemlerden sonra, aşağıdaki genelleştirilmiş öz değerler problemi çözülür:

$$XMX^T \boldsymbol{a} = \lambda X X^T \boldsymbol{a} \quad (21)$$

M matrisi için aşağıdaki denklem kullanılır:

$$M = (I - W)^T (I - W) \quad (22)$$

Denklem 21'nin çözümü olan en küçük değere sahip *r* tane öz değerlere karşılık gelen *r* tane öz vektör ile *P* dönüşüm matrisini oluşturur.

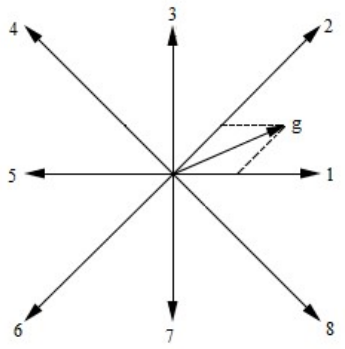
3. Deneysel Çalışmalar

3.1. Veri Kümesi

Bu deneysel çalışmada, USPS veri kümesi kullanılmaktadır. USPS, zarflar üzerindeki elle yazılmış rakamları tanımak için oluşturulmuş bir veri kümesidir [7]. Her bir nesne 16x16 pikselden oluşmaktadır. Eğitim kümesi ve test kümesiyle sırayla 7291 ve 2007 örneğe sahiptir.

3.2. Öznitelik Çıkarılması

Bu deneysel çalışmada, rakam veya karakter tanımada sıkça kullanılan gradyan öznitelikleri [8] tercih edilmiştir. Gradyan öznitelikleri, Sobel operatörü ile hesaplanır. Sobel operatörünü veri kümesindeki her görüntüye uygularsak, her görüntü pikseli için gradyanın x-eksen bileşenini ve y-eksen bileşenini elde ederiz. Daha sonra her gradyan vektörü standart sekiz yönden kendisine yakın olan iki yönde bileşenlere ayrılır. Şekil 1'de *g* gradyan vektörünün bileşenlerine ayrılması için bir örnek gösterilmektedir.



Şekil 1. Gradyan standart yönler ve bir vektörü bileşenlere ayırma

Her bir görüntü, genellikle 4x4 veya 5x5 bölgelere ayrılır. Her bölgedeki her bir standart yön için, toplam bileşen vektörler hesaplanır. Bütün bölgelerdeki standart yönlerin şiddetleri, görüntü örnekleri için öznitelik vektörünü oluşturur. Dolayısıyla her bir bölge için 8 tane öznitelik elde edilir.

Bu deneysel çalışmada 16x16 piksele sahip görüntüler, 4x4 bölgeye ayrılır. Her bölge için 8 standart yön şiddeti bir öznitelik olduğu için, 4x4x8=128 boyutlu öznitelik vektörleri oluşur.

3.3. Boyut İndirgeme

Veri kümesine ait örneklerinin öznitelikleri elde edildikten sonra, lineer boyut indirgeme yöntemleri uygulanır. Boyut indirgeme işleminin süresi önemlidir. Süre göz önüne alındığında, boyutu azaltılmış verinin boyut indirgeme ve sınıflandırma işlemi boyutu azaltılmamış verinin sınıflandırma işleminden daha kısa olması beklenir. Aksi halde zaman açısından boyut indirgeme işleminin kazancı olmaz.

Bildiride incelenen lineer boyut indirgeme yöntemleri, öznitelikleri çıkarılmış veriye uyguladığında bu yöntemlerin süreleri çizelge 1'de verilmiştir.

Çizelge 1: Lineer boyut azaltma yöntemlerinin işlem süreleri(saniye)

	TBA	DDA	YKİ	YDDA	KKG
Süre	0.058	0.065	0.75	3.59	1.84

3.4. Sınıflandırma

Bu kısımda boyut indirgeme yöntemlerinin, sınıflandırmanın doğruluğu ve süresi açısından etkileri incelenecektir. Boyut indirgeme yöntemleri ile sınıflandırmada kullanılacak olan boyut sayısı azaltılarak işlem yükü azaltılacak ve dolayısıyla işlem süresini azaltması sağlanacaktır. Bunun yanına ek olarak, sınıflandırma doğruluğunun kabul edilebilir seviyede olmasını da istenmektedir. Sınıflandırma işlemi için *k* en yakın komşu (*k*-EK) [9], destek vektör makineleri (DVM) [9] ve yapay sinir ağları (YSA) [9] kullanılmıştır.

Çizelge 2: Boyut indirgeme işlemi uygulanmadan yapılan sınıflandırma işleminin doğrulukları(%) ve süreleri(saniye)

	k-EK	DVM	YSA
Doğruluk(%)	96.41	97.11	96.51
Süre(saniye)	4.50	4.68	578

Çizelge 3: Boyut indirgeme ile yapılan sınıflandırma işleminin doğrulukları(%)

	k-EK	DVM	YSA
TBA	93.02	93.82	93.12
DDA	95.57	95.96	95.91
YKİ	94.87	94.92	94.62
KKG	95.22	95.22	94.87
YDDA	94.67	95.12	94.67

Çizelge 4: Boyut indirgeme ile yapılan sınıflandırma işleminin süreleri(saniye)

	k-EK	DVM	YSA
TBA	0.52	1.1	115
DDA	0.52	0.6	115
YKİ	0.52	0.7	115
KKG	0.52	0.7	115
YDDA	0.52	0.85	115

Kısım 3.2'de veriden elde edilen özniteliklere boyut indirgeme işlemi uygulanmadan yapılan sınıflandırma işlemlerinin doğrulukları ve süreleri çizelge 2'de gösterilmiştir.

Kısım 3.3'te boyut indirgeme işlemiyle elde edilen öznitelikler ile yapılan sınıflandırma işlemlerinin doğrulukları çizelge 3'te ve süreleri çizelge 4'te gösterilmiştir.

4. Sonuçlar

Veri analizinde, boyut indirgeme yöntemlerine sıkça başvurulur. Birçok boyut indirgeme yöntemi olmasına rağmen, lineer yöntemler daha çok tercih edilir. Bu bildiride popüler lineer boyut indirgeme yöntemleri incelenmiştir. Lineer boyut indirgeme yöntemlerinin işlem süreleri açısından en başarılı yöntem TBA ve DDA olarak gözükmektedir. Bu konuda en kötü performansı YDDA yöntemi göstermiştir. Elde edilen sonuçlara göre, incelenen lineer boyut indirgeme yöntemleri sınıflandırma sürelerini kayda değer bir şekilde azaltmıştır. Özellikle YSA yöntemi ile sınıflandırma işleminin süresi çok ciddi bir azalma göstermiştir. Lineer boyut indirgeme yöntemleri ile sınıflandırma doğrululuklarında azalma görülmüştür. USPS veri kümesi üzerinde incelenen boyut indirgeme yöntemleri arasında sınıflandırma doğruluğu açısından en başarılı yöntem DDA olmuştur. Bu konuda en kötü performansı ise TBA yöntemi göstermiştir. Genel olarak bu deneysel çalışmada, veri boyutu indirgenerek yapılan sınıflandırma işlemlerinin doğrulukları kabul edilebilir seviyededir.

5. Kaynaklar

[1] Martinez, W.L., Martinez, A.R., Martinez, A. ve Solka, J., *Exploratory Data Analysis with MATLAB,* CRC Press, 2010.

[2] Martinez, M., ve Kak, A.C., "PCA versus LDA," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Cilt No. 23, 228-233, 2001.

[3] Belhumuer, P.N., Hespanha, J.P. ve Kriegman, D.J., "Eigenfaces vs. Fisherfaces: Recognition Using Class Specific Linear Projection", *IEEE Transactions on Pattern Analysis and Machine Intelligence,* Cilt No. 19, 711-720, 1997.

[4] He, X. ve Niyogi, P., "Locality Preserving Projections", *Proc. Conf. Advances in Neural Information Processing Systems*, 2003.

[5] Cai, D., He, X., Zhou, K., Han, J., ve Bao, H., "Locality Sensitive Discriminant Analysis", *International Joint Conference on Artificial Intelligence*, 2007, 708-713.

[6] He, X, Cai, D., Yan, S., ve Zhang, H.J., "Neighborhood Preserving Embedding", *International Conference on Computer Visions,* 2005, 2208-1213.

[7] Hastie, T., Tibshirani, R. ve Friedman, J.H., *The Elements of Statistical Learning*, Springer, 2003.

[8] Liu, C.L., Nkashima, K., Sako, H. ve Fujisawa, H., "Handwritten digit recognition: investigation of normalization and feature extraction techniques", *Pattern Recognition*, Cilt No.36, 265-279, 2003.

[9] Han, J., Kamber, M. Ve Pei, J., *Data Mining : Concepts and Techniques,* Morgan Kaufmann, 2011.