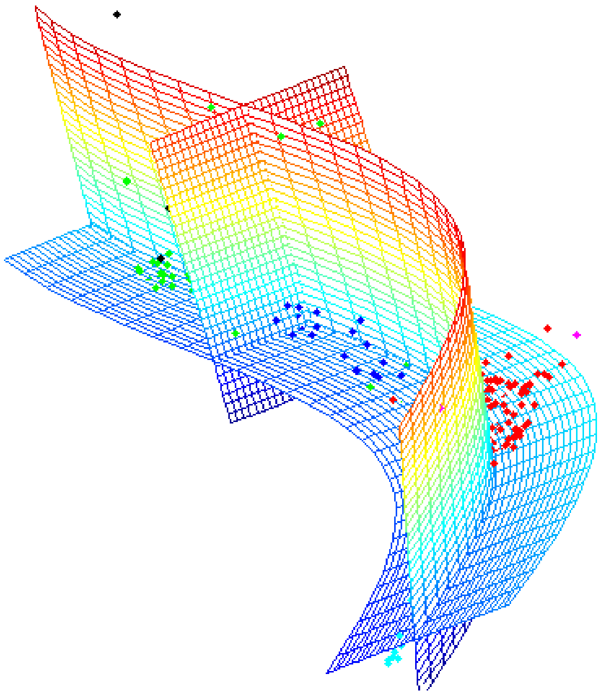
Temel Bileşenler Analizine

Genel Bir Bakış



Temel Bileşenler Analizi, yüz tanıma, resim sıkıştırma ve örüntü tanıma gibi alanlarda yaygın olarak kullanılan istatistiksel bir metottur. Bu çalışmada, temele bileşenler analizi konusuna geçmeden önce öncelikle matematiksel temele değinilmiştir. Daha sonra konunun daha rahat anlaşılmasını sağlamak amacıyla iki boyutlu bir veri seti üstünde temel bileşenler analizi adım adım uygulanmıştır. Sonraki bölümlerde metodun farklı alanlarda uygulanmasını içeren çalışmalar yer almaktadır.

İçindekiler

Şekiller ve Tablolar Dizini

1. Giriş

Bu çalışmada Temel Bileşenler Analizi konusuna değinilmiştir. Konunun anlaşılması için basit örneklere yer verilmiş ve uygulamanın kodlanması için MATLAB kullanılmıştır. Temel Bileşenler Analizi belge boyunca PCA olarak ifade edilmiştir.

Temel Bileşenler Analizi, yüz tanıma, resim sıkıştırma ve örüntü tanıma gibi alanlarda yaygın olarak kullanılan istatistiksel bir metottur. PCA ile ilgili detaylara geçmeden önce konun daha rahat anlaşılabilmesi için matematiksel temele değinilmiştir.

PCA konusunu anlayabilmek için öncelikle Tekil Değer Ayrışımı konusunun incelenmesi gerekmektedir. Özdeğer-Özvektör Hesaplamaları ve Dik İzdüşüm konuları da Temel Bileşenler Analizinin anlaşılması için incelenmesi gereken konular arasına girmektedir.

Bu nedenle çalışmada öncelikle bahsi geçen konulara göz atılmış ve matematiksel temel açıklandıktan sonra PCA ve PCA'nın uygulama alanlarına yer verilmiştir.

Notların faydalı olmasını diliyorum.

Zafer CÖMERT

2. Matematiksel Temel

Bu bölümde PCA'nın anlaşılabilmesi için gerekli olan bazı matematiksel ifadelere değinilecektir. Her bölüm birbirinden bağımsız olarak hazırlanmıştır.

2.1. İstatistik

İstatistik, belirli bir amaç için veri toplama, tablo ve grafiklerle özetleme, sonuçları yorumlama, sonuçların güven derecelerini açıklama, örneklerden elde edilen sonuçları kitle için genelleme, özellikler arasındaki ilişkiyi araştırma, çeşitli konularda geleceğe ilişkin tahmin yapma, deney düzenleme ve gözlem ilkelerini kapsayan bir bilimdir. Belirli bir amaç için verilerin toplanması, sınıflandırılması, çözümlenmesi ve sonuçlarının yorumlanması esasına dayanır [1]. İstatistiğin temel prensibi, tamamen büyük veri kümeleri üzerinde ilgilenilen alanlara bağlı olarak bu büyük veri kümesi daha küçük alt kümelerle temsil ve analiz edebilmektedir.

2.2. Standart Sapma

Standart sapma, verilerin nasıl yayıldığına (saçıldığına) dair ölçümsel olarak bilgi verir. Veri değerlerinin yayılımının özetlenmesi için kullanılan bir ölçüdür. Standart sapma varyansın kareköküdür.

$$\overline{X} = \frac{\sum_{i=1}^{n} x_i}{n}$$

$\overline{X}$, X dizisinin ortalamasını ifade etmektedir. Buna göre standart sapma:

$$s = \sqrt{\frac{\sum_{i=1}^{n} (X_i - \overline{X})^2}{n-1}}$$

2.3. Varyans

Varyans verinin yayılımı ile ilgili bir başka ölçüm bilgisi veren kavramdır. Genellikle değişimi ölçmek için kullanılır. Varyans, standart sapmanın karesidir.

$$s^2 = \frac{\sum_{i=1}^{n}(X_i - \overline{X})^2}{n-1}$$

2.4. Kovaryans

Olasılık teorisi ve istatistikte, kovaryans iki değişkenin birlikte ne kadar değiştiklerinin ölçüsüdür. Kovaryans, iki rasgele değişkenin beraber değişimlerini inceleyen bir istatistiktir [2]. Standart sapma ve varyans tek boyutlu veriler için kullanılmaktadır. Ancak çoğu zaman veri setleri birden fazla boyuta sahiptir. Kovaryans her zaman iki boyut arasında ölçüm yapmak için kullanılmaktadır.

$$var(X) = \frac{\sum_{i=1}^{n}\left(X_i - \overline{X}\right)\left(X_i - \overline{X}\right)}{n-1}$$

Yukarıdaki formüle benzer şekilde X değişkeni ve Y değişkeni arasındaki ölçüme bakılmak istendiğinde aşağıdaki formül yardımıyla bu iki değişken arasındaki ilişkiye bakılabilir.

$$cov(X,Y) = \frac{\sum_{i=1}^{n}\left(X_i - \overline{X}\right)\left(Y_i - \overline{Y}\right)}{n-1}$$

Kovaryans değeri, pozitif ise her iki değişkenin birlikte arttığı; negatif ise biri artarken diğerinin azaldığı; sıfır ise bu iki değişkenin bağımsız olduğu yorumu yapılabilir.

2.5. Kovaryans Matris

İkiden fazla değişkene bakıldığında kovaryans matris kullanılır. Kovaryans matristeki diyagonal değerler değişkenlerin varyans değerlerine eşittir. Kovaryans matris $cov(a,b) = cov(b,a)$ özelliğinden dolayı simetrik bir yapı sergilemektedir.

$$C = \begin{pmatrix} cov(x,x) & cov(x,y) & cov(x,z) \\ cov(y,x) & cov(y,y) & cov(y,z) \\ cov(z,x) & cov(z,y) & cov(z,z) \end{pmatrix}$$

X	**Y**	$\boldsymbol{X-\overline{X}}$	$\boldsymbol{Y-\overline{Y}}$	$(\boldsymbol{X-\overline{X}})(\boldsymbol{Y-\overline{Y}})$
9	39	-4,91667	-23,4167	115,1319
15	56	1,083333	-6,41667	-6,95139
25	93	11,08333	30,58333	338,9653
14	61	0,083333	-1,41667	-0,11806
10	50	-3,91667	-12,4167	48,63194
18	75	4,083333	12,58333	51,38194
0	32	-13,9167	-30,4167	423,2986
16	85	2,083333	22,58333	47,04861
5	42	-8,91667	-20,4167	182,0486
19	70	5,083333	7,583333	38,54861
16	66	2,083333	3,583333	7,465278
20	80	6,083333	17,58333	106,9653
Toplam				1352,416667
Ortalama				112,7013889

Tablo 2-1 İki Boyutlu Veri Kovaryans Hesaplaması

2.6. Özvektörler ve Özdeğerler

Bilindiği üzere boyutlar uyumlu olduğu sürece iki matris çarpılabilir ve özvektörler bu kural için özel bir durum ifade etmektedir. Bir vektör üzerine uygulanan matris o vektörün hem büyüklüğünü hem de yönünü değiştirebilir. Buna rağmen, bir matris bazı belirli vektörler üzerinde etkidiğinde onun büyüklüğünü bir çarpan kadar katlar, yani sadece büyüklüğünü değiştirir, doğrultularını değiştirmez. Doğrultusu değişmeyen bu vektörler söz konusu matrisin özvektörleri olarak ifade edilir. Özdeğerler, özvektektörler, özuzaylar bir matrisin özellikleridir ve matris hakkında önemli bilgiler vermektedir. Özvektörler ancak kare matrislerden elde edilebilir. Bu nedenle bir özdeğer ve özvektör elde etmek için kovaryans matrisler kullanılmaktadır. Ancak her kare matrisin özvektörleri yoktur [3].

2.7. Tekil Değer Ayrışımı

Tekil Değer Ayrışımı(TDA), bir matrisin çarpanlarına ayrılma türlerinden biridir ve Google'ın PageRank algoritmasından insan yüzü modellemeye, otomatik deneme notlamasından gen analizine, bilgi getirimi ve çıkarımından boyut azaltma ve veri sıkıştırmaya kadar uzanan geniş bir yelpazede kullanılan temel adımdır.

Skalerler küçük harflerle yazılır, örneğin a. Vektörler kalın küçük harflerle yazılır, örneğin **a. a** vektörünün i indisindeki elemanı a_i olarak gösterilir. Matrisler kalın büyük harflerle yazılır, Örneğin **A**. **A**'nın j. sütunu $\boldsymbol{a_j}$ vektörü olarak, (i, j) hücresindeki elemanı ise a_{ij} olarak ifade edilir. ^{H} işareti Hermit (karmaşık) transpozu göstermek üzere kullanılır. $a \in C^I$ vektörünün L_2 normu (veya Öklit normu) $\|.\|_2$ sembolleri ile gösterilir ve şöyle tanımlanır:

$$||a||_2 := \sqrt{|a_1|^2 + \cdots + |a_I|^2} = \sqrt{(a^H a)} \qquad (1)$$

$\boldsymbol{A} \in C^{IxJ}$ matrisinin spektral normu, benzer şekilde $||\,.\,||_2$sembolleri ile gösterilir ve $\boldsymbol{A^H A}$ matrisinin en büyük özdeğerinin kareköküne eşittir:

$||\boldsymbol{A}||_{\boldsymbol{2}} :=$ $\boldsymbol{A^H A}$ matrisinin en büyük özdeğerinin karekökü

$$= max_{||x||_2 \neq 0} = \frac{||Ax||_2}{||x||_2} \tag{2}$$

Dikey (orthogonal) $\boldsymbol{A} \in C^{IxJ}$ matrisinin farklı sütunlarının iç çarpımları sıfırdır. Sütun normları 1 olan dikey matrise birim dikey (orthonormal) matris denir. Aynı zamanda matris kare ise, üniter (unitary) matris olarak adlandırılır ve $\boldsymbol{A^H A} = \boldsymbol{A A^H} = \boldsymbol{I}$ denkliğini sağlar.

Her $\boldsymbol{A} \in C^{I \times J}$ matrisini

$$\boldsymbol{A} = \boldsymbol{U \Sigma V^H} \tag{3}$$

Biçiminde alttaki maddeleri sağlayacak şekilde çarpanlarına ayrılabilir.

- $\boldsymbol{U} = [\boldsymbol{u_1} \quad \boldsymbol{u_2} \quad \cdots \quad \boldsymbol{u_I}] \in C^{IxJ}$ üniter bir matristir.
- $\boldsymbol{V} = [\boldsymbol{v_1} \quad \boldsymbol{v_2} \quad \cdots \quad \boldsymbol{v_I}] \in C^{IxJ}$ üniter bir matristir.
- $\Sigma \in C^{IxJ}$ sözdeköşegen (pseudodiagonal) bir matristir.

$\Sigma =$ köşegen$(\sigma_1, \sigma_2, \ldots \sigma_{\min(I,J)})$ ve sıralıdır $(\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_{\min(I,J)} \geq 0$ [4]

A: nxp tipinde bir matris, $rank(A) = r$ olmak üzere $U : nxn$, $V : pxp$ ortogonal matrisleri vardır öyle ki:

$$A = U \begin{bmatrix} D & 0 \\ 0 & 0 \end{bmatrix} V'$$

biçiminde yazılır. Burada, U matrisi AA' nın ve V matrisi $A'A$ nın normlanmış özvektörlerinin matrisleridir. D matrisi, AA' matrisinin ($A'A$ matrisinin) sıfırdan farklı d_i $(d_i > 0, i = 1,2, \ldots, r)$özdeğerlerinin

$$D = \begin{bmatrix} \sqrt{d_1} & \cdots & \\ \vdots & \ddots & \vdots \\ & \cdots & \sqrt{d_r} \end{bmatrix}$$

köşegen matrisidir.

U matrisinin ilk r sütunundan oluşan matris $U_1 : nxr$ $(U_1{}'U_1) = I_r$ V matrisinin ilk r satırından oluşan matris $V_1 : rxp$ $(V_1 V_1{}') = I_r$ V olmak üzere,

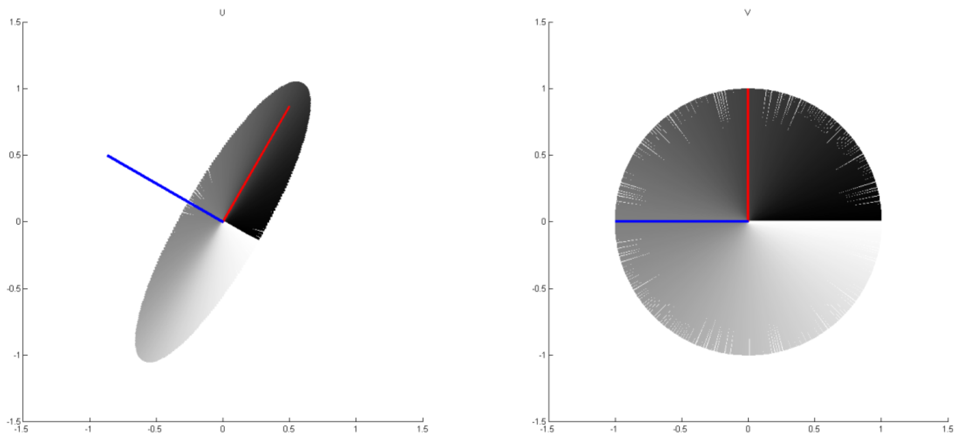
$$A = U_1 D V_1{}'$$

Tersine, bu gösterimden yukarıdakine geçmek için D matrisi sıfır matrisleri ile genişletilecek ve U_1ile V_1matrisleri ortogonal matrisler olacak şekilde genişletileceklerdir. A matrisinin böyle bir gösterimine tekil değer ayrışımı ve D matrisinin elemanlarına A'nın tekil değerleri denir [5].

Örnek

```
%% Tekil Değer Ayrısımı
clc, clear all
A=[1 2 3 4 5;2 2 2 2 2;3 4 5 6 7;5 6 7 8 9; 8 10 12 14 16]
[U D V]=svd(A)
%% A Matrisinin Değerleri Elde Edildi
AA = U*D*V'
U1=U(:,1:2)
V1=V(:,1:2)
D1=D(1:2,1:2)
%% A Matrisinin Değerlerine Ayrışımlarla Geri Dönüldü
AAA = U1*D1*V1'
```

Örnek



Şekil 2.1 Tekil Değer Ayrışımı

Bu bölümde $\boldsymbol{A} \in R^{2x2}$ için bunu görselleştirmek için örnek bir Matlab uygulamasına göz atılacaktır. 2 boyut için düşünülürse, v_j vektörleri bir birim çemberin iki dik vektörü olacak, dönüşümde elde edilecek u_i vektörleri ise bir elipsin eksenlerini oluşturacaktır. Daha büyük boyutlarda hiperkürenin hiperelipsoide dönüşümünü de benzer şekilde gözlemlenebilir.

```
% Matrisi tanımı
A = [ 0.3*cos(pi/6)   1.2*sin(pi/6)
      0.3* -sin(pi/6) 1.2*cos(pi/6)];

% Figür ayarları
fig1 = subplot(1,2,1);
hold on; axis equal; title('U')
fig2 = subplot(1,2,2);
hold on; axis equal; title('V')
linkaxes([fig1, fig2]);
axis([-1.5 1.5 -1.5 1.5])

for theta = 0:pi/150:2*pi

    % Her açı için birim çember üstünde bir x alınıp, nereye düştüğüne
    % bakılır
    x = [cos(theta); sin(theta)];
    y = A*x;

    % Gittikçe beyazlaşacak şekilde çizim
    plot(fig1, [0 y(1)], [0 y(2)], ...
        'Color', [theta/(2*pi) theta/(2*pi) theta/(2*pi)], ...
        'LineWidth', 3);
    plot(fig2, [0 x(1)], [0 x(2)], ...
        'Color', [theta/(2*pi) theta/(2*pi) theta/(2*pi)], ...
        'LineWidth', 3);
end

% Matlab'ın bulduğu değerler
[U, S, V] = svd(A);

% Sol ve sağ tekil vektörleri çizelim
plot(fig1,[0 U(1,1)], [0 U(2,1)], 'Color', 'red', 'LineWidth', 3);
plot(fig1,[0 U(1,2)], [0 U(2,2)], 'Color', 'blue', 'LineWidth', 3);
plot(fig2,[0 V(1,1)], [0 V(2,1)], 'Color', 'red', 'LineWidth', 3);
plot(fig2,[0 V(1,2)], [0 V(2,2)], 'Color', 'blue', 'LineWidth', 3);
```

Örnek uygulamaya bakıldığında, birim çemberin bir elipse dönüştüğünü ve aranılan tekil değerlerin bu elipsin eksen uzunluklarının yarıları olduğu ortaya çıktı. Örnekte bu değerler 1.2 ve 0.3 olarak elde edildi. Soldaki çizimde kırmızı ve mavi doğru parçaları birim uzunlukta (bunlar u_1 ve u_2 vektörleri), elipsin dik eksenleri ise görüldüğü gibi 1.2 ve 0.3 katları. Zaten matris önce ölçekleyip sonra döndürecek şekilde özel olarak seçilmiştir, sonuçta da bunlar elde edildi.

Matematikte bir probleme yaklaşımda üç temel soru başat konumdadır.

1. Çalıştığımız uzayın tüm elemanları için bu ayrışım var mıdır? (Existence)
2. Bir eleman için ayrışım tek midir? (Uniqeuness)

3. Değerlerdeki küçük oynamalar (Perturbation) ayrışım sonuçlarını ne denli etkiler?

Gürültülü bir veri için sonuçlara ne denli güvenilir? Yani değerler hafif farklı olsaydı tamamen farklı sonuçlar mı alınırdı, yoksa sonuçlarda küçük değişiklikler mi gözlenirdi? Özetle yöntem kararlı mıdır? Ya da başka bir deyişle problem kötü konumlanmış bir problem midir? Üstteki üç özelliği sağlayan bir problem iyi konumlanmış olarak da tanımlanmaktadır.

Yukarıdaki sorular ışında örnek uygulama ele alındığında: Herhangi bir matris için tekil değer ayrışımı yapılabileceği görülmüştü, dolayısıyla birinci soru çözümlendi. Bir matrisin tek bir tekil değer ayrışımı mı var, yoksa birden çok çözüm olabilir mi? Şekil 2.1'de sağdaki çemberin kırmızı ve mavi ile gösterilen eksenleri $\boldsymbol{v_1}$ ve $\boldsymbol{v_2}$ idi. Soldaki elipsinkiler ise $\boldsymbol{u_1}$ ve$\boldsymbol{u_2}$. Bu ayrışımda kırmızı ile gösterilen eksenleri ters yöne çevirseydik, yani $\boldsymbol{U} = [\boldsymbol{u_1} \; -\boldsymbol{u_2}]$ ve $\boldsymbol{V} = [\boldsymbol{v_1} \; -\boldsymbol{v_2}]$ olsaydı $\boldsymbol{A} = \boldsymbol{U\Sigma VH}$ eşitliği korunurdu. Dolayısıyla tek bir ayrışım yoktur. Aynı indekse ait sağ ve sol tekil vektörleri birlikte ters yöne çevirince aynı çarpım elde edilecektir.

Bunun dışında birim matrise bakmakta da aydınlatıcı olacaktır.

$$\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \cos(\theta) & \sin(\theta) \\ -\sin(\theta) & \cos(\theta) \end{bmatrix} \tag{4}$$

görüldüğü üzere ne yazık ki, birim çember için de tek bir çözüm yok, hatta sonsuz çözüm var. Yani tekil değer ayrışımı için ikinci madde sağlanamamaktadır. Fakat bu uygulamalarda sorun yaşatmayacak çünkü tekil vektörlerin yönleri değil doğrultuları önemli olacaktır. Birim matriste olduğu gibi tekil değerlerin aynı olması durumunda sonsuz çözüm olması da sorun oluşturmayacak çünkü seçilen herhangi bir çözüm uygulamadaki ihtiyaçlara eşdeğer güçte cevap verecektir.

TDA'yı güçlü kılan kararlı bir ayrışım yöntemi olduğunu bilmektedir. Deneylerdeki küçük nümerik farklar sonuçlarda beklenmedik sıçramalara sebep olmayacak ve analizleri güvenilir kılacaktır [4].

2.8. Dikey Bazı Matrisler

Eldeki matrisi ifade etmek üzere Denklem **(5)** kullanılabilir.

$$A = U\Sigma V^H = \Sigma_{i=1}^{n} \sigma_i u_i v_i^H \tag{5}$$

Yukarıdaki formülde n. dereceden bir matris, 1. dereceden ve birbirine dikey n tane matrisin doğrusal birleşimi (lineer kombinasyonu) biçiminde yazıldı. $\boldsymbol{A}$ matrisine en çok katkıda bulunan matris $\sigma_1 u_1 v_1^H$ matrisi, çünkü en büyük tekil değer σ_1 ve $u_i v_i^H$ matrislerinin Frobenius normlarıdır.

Eldeki matrisi eşit büyüklükteki farklı legolardan oluşmuş bir oyuncak gibi düşünülürse, bir biçimde bu oyuncakta kullanılan parçaları bulundu $(u_i v_i^H)$ ve parçaların hangi oranda kullanıldığını (σ_1) belirlendi [4].

2.9. Dik İzdüşüm

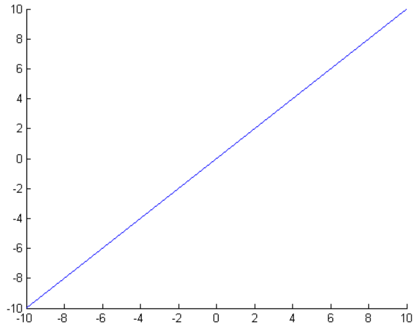
$P_{[A]} = AA^+$, yani AA^+matrisi A'nın sütun vektörlerinin gerdiği $[A] \subset R^{nx1}$alt uzayı üzerine dik izdüşüm matrisidir. Ayrıca, A'ın sütun vektörlerinin gerdiği uzaya dik olan alt uzay $[A]^\perp(R^n = [A] \oplus [A]^\perp)$ olmak üzere, bu alt uzay üzerine dik izdüşüm matrisi $P_{[A]^\perp} = I - AA^+$dır [5].

Örnek

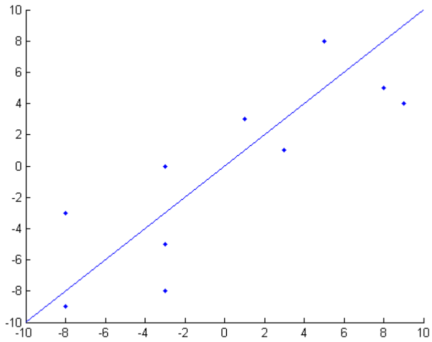
İki boyutlu vektör uzayındaki vektörlerin (iki boyutlu koordinat sistemindeki noktaların) aşağıdaki mavi çizgi ile gösterilen doğru üzerine dik iz düşürülmesi:

Mavi çizgi $\begin{bmatrix}1\\1\end{bmatrix} \in R^{2x1}$ vektörü tarafından gerilen alt vektör uzayındaki noktaların bir kısmıdır.

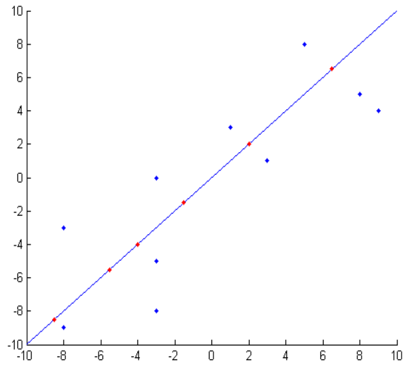
$$\left\{v : v = c\begin{bmatrix}1\\1\end{bmatrix}, c \in [-10,10]\right\}$$



Şekil 2.2 Öncelikle bir doğru tanımlanmıştır.



Şekil 2.3 Daha sonra noktalar bu doğruya göre çizdirilmiştir.



Şekil 2.4 Her bir noktanın doğruya göre iz düşümü alınmıştır.

```
close all, clear all, clc
hold on
grid off
plot([-10 10],[-10 10])
Noktalar=[1 3; 3 1; 5 8; 8 5; 9 4; -3 0; -3 -5; -3 -8; -8 -3; -8 -9]
plot(Noktalar(:,1),Noktalar(:,2),'.')
A=[1 1]'
DikizdA=A*pinv(A)
izdNoktalar=Noktalar*DikizdA
plot(izdNoktalar(:,1),izdNoktalar(:,2),'.r')
B=[1 3]'
plot([3 -3],[9 -9])
izdNoktalar=Noktalar*(B*pinv(B))
plot(izdNoktalar(:,1),izdNoktalar(:,2),'.g')
```

3. Temel Bileşenler Analizi

Temel Bileşen Analizi[1] nedir? Hali hazırda elde var olan bir veri için öyle bir dönüşüm yapılmak isteniyor ki, bu dönüşümün sonunda elde edilecek örnekler birbirinden mümkün olduğunca ayrılmış, dağılmış, saçılmış olsun [6].

Temel bileşenler yaklaşımı bağımlılık yapısını yok etme ve boyut indirgeme amaçları için kullanılmaktadır. Tanıma, sınıflandırma, boyut indirgenmesi ve yorumlanmasını sağlayan, çok değişkenli bir istatistik yöntemidir. Bu yaklaşım verinin içindeki en güçlü örüntüyü bulmaya çalışır. Bu yüzden örüntü bulma tekniği olarak da kullanılabilir. Çoğunlukla verinin sahip olduğu çeşitlilik, tüm boyut takımından seçilen küçük bir boyut setiyle yakalanabilir. Verideki gürültüler, örüntülerden daha güçsüz olduklarından, boyut küçültme sonucunda bu gürültüler temizlenebilir.

PCA'nın üç temel amacı vardır:

1. Verilerin boyutunu azaltma
2. Tahminleme yapma
3. Veri setini, bazı analizler için görüntülemek

PCA uygulandığında p boyutlu uzayın gerçek boyutu belirlenir. Bu gerçek boyuta temel bileşenler adı verilir. Temel bileşenlerin üç özelliği vardır:

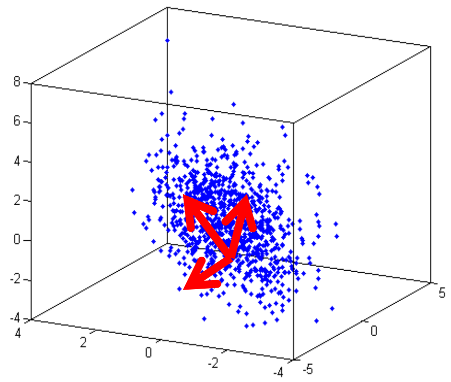
1. Kolerasyonsuzlardır.
2. Birinci temel bileşen toplam değişkenliği en çok açıklayan değişkendir.
3. Bir sonraki temel bileşen kalan değişkenliği en çok açıklayan değişkendir. [7]

[1] Principal component analysis (PCA) is a statistical procedure that uses an orthogonal transformation to convert a set of observations of possibly correlated variables into a set of values of linearly uncorrelated variables called principal components. The number of principal components is less than or equal to the number of original variables. This transformation is defined in such a way that the first principal component has the largest possible variance (that is, accounts for as much of the variability in the data as possible), and each succeeding component in turn has the highest variance possible under the constraint that it is orthogonal to (i.e., uncorrelated with) the preceding components. The principal components are orthogonal because they are the eigenvectors of the covariance matrix, which is symmetric. PCA is sensitive to the relative scaling of the original variables.

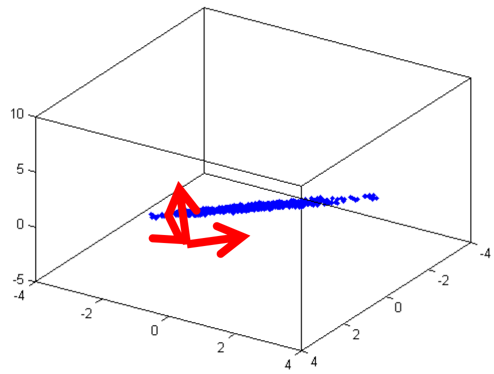
Temel Bileşen Analizi bu soruna şu şekilde yaklaşmaktadır:

a. Çok boyutlu verilere doğru açıdan bakarak genellikle verideki ilişkiler açıklanabilir.

b. PCA'nın amacı bu "doğru açıyı" bulmaktadır.

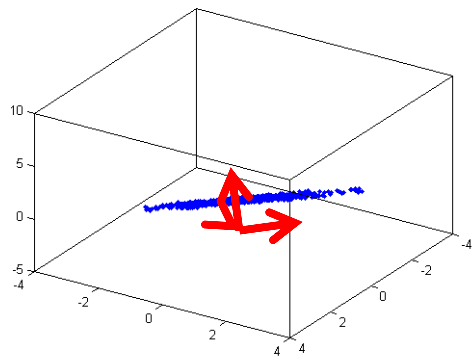


Şekil 3.1 Veri çok boyutlu ve ilişkiler açık değildir.

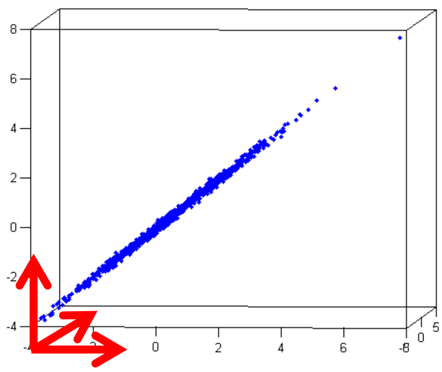


Şekil 3.2 Doğru açıdan bakıldığında çok boyutlu karmaşık veri setindeki ilişkinin lineer olduğu görülmektedir.

PCA kilit noktası, problemi çözmek üzere görsel inceleme için uygun bir "açı" yani uygun bir koordinat sistemi seçmektir.



Şekil 3.3 Karmaşık veri seti lineer hale getirilmiştir.



Şekil 3.4 Yeni koordinat sisteminde verinin görüntüsü elde edilmiştir.

Uygun "açıdan" verilere bakmak, bu koordinat sistemi kullanarak verileri incelemek demektir.

PCA'da, uygun koordinat sistemi aşağıdaki şekilde aranmaktadır:

- 1. eksen olarak, verilerin en büyük değişiminde olan yön seçilir.
- 2. eksen olarak, önceki 1. eksene dikey olan ve verilerin en büyük değişiminde olan yön seçilir.
- 3. eksen olarak, önceki 1. ve 2. eksene dikey olan ve kalan verilerin en büyük değişiminde olan yön seçilir.
- Böyle – her zaman *yeni eksen olarak verilerindeki en büyük kalan değişimde* olan yön seçilmektedir.

Böyle seçilmiş dikey olan "en büyük değişim" yönlerine **"temel bileşenler"** denir. PCA yönleri, verilerin değişimi ile ilgili en büyük katkıda olan yönü ilk önce belirtmekte, daha sonra da daha az katkıda olan yönleri açıklamaktadır.

Gerçek uygulamalarda "çok boyutlu ve ilk bakıştan çok karmaşık verilerde çok az temel etki olduğu" düşünülebilir. Bu etkiler birkaç ilk PCA yönü olarak bulunur. Bu anlamda, PCA yönleri, çok boyutlu karmaşık verileri sıkça 2-3 yeni "özellikle" açıklayıp gösterebilmektedir.

Temel bileşenlerin yeterli sayısını belirtmek için "tutulan varyans[2]" kavramı kullanılmaktadır. Kullanılacak ilk temel bileşenlerin toplam varyansı orijinal verilerin toplam varyansının %90-%95'i olmalıdır. Genel uygulamalarda, 1000 boyutlu veriler için genelikle10-20 ilk temel bileşen verilerin %90-%95 değişimini vermektedir. Bir başka ifadeyle, orijinal veriler %95 doğrululukla temsil etmek için 10-20 PCA bileşeni yeterli olabilir. Örneğin, 1000 özellikli veriyi kaydetmek için bütün 1000 özelliği kaydetme yerine 10-20 ilk temel bileşen kaydedilip diğer bileşenlerin değerleri için ortalama olarak (çünkü onlar aşağı yukarı değişmez) depolanabilir. Orijinal veriler, %1-2 bellekle kaydedilebilir.

[2] Recovered varians

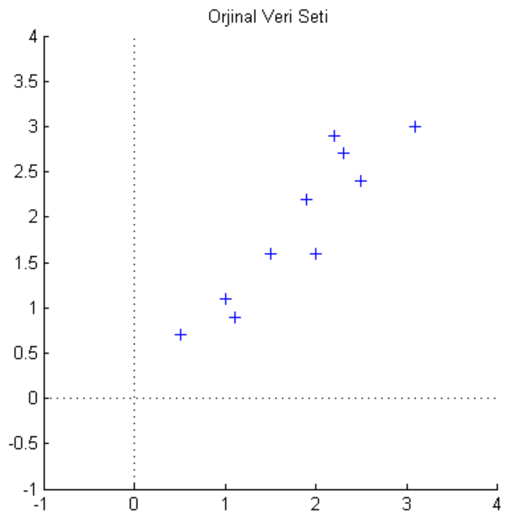
Sonuç olarak:

- PCA, boyut azaltmada çok faydalı bir yöntemdir.
- PCA, çok boyutlu verileri yaklaşık olarak ve daha az boyutlu veriyle temsil eder.
- PCA, orijinal veriler için dik-olan-en-büyük-varyans-yönleri bulup orijinal verileri bu koordinat sisteminde gösterir.
- PCA, çok boyutlu verilerin görsel gösterilmesi ve incelenmesi için kullanılabilir.
- PCA, makine öğrenmesi olarak, verilerin boyutu azaltabilir–az değişen PCA özellikleri modelleme için önemsiz olabilir, bu şekilde modelleme ile ilgili hesaplama hızlandırabilir.
- PCA, veri sıkıştırma için de kullanılabilir.

Özetle PCA, istatistiksel bir metottur. Bir veri setindeki örüntünün tanımlanmasında, veri setinin açıklanmasında, veri içindeki benzer ve farklı desenlerin tanımlanmasında kullanılabilir. PCA verinin sıkıştırılmasına boyut azaltarak imkân vermektedir. Üstelik boyut azaltılırken veri kaybı da yaşanmamaktadır. Bu teknik bilgisayar bilimleri içerisinde görüntü işleme alanında sıkça kullanılmaktadır. Çalışmanın ilerleyen bölümünde bir veri seti üzerinde PCA uygulanacak ve adım adım her aşamada neler yapıldığına değinilecektir.

3.1. Veri Setinin Hazırlanması (1.Adım)

Hazırlanan basit örnekte, PCA iki boyutlu bir veri setine uygulanacaktır. Veri seti Tablo 3-1'de gösterilmiştir.



Şekil 3.5 Veri Seti

x	y
2,5	2,4
0,5	0,7
2,2	2,9
1,9	2,2
3,1	3
2,3	2,7
2	1,6
1	1,1
1,5	1,6
1,1	0,9

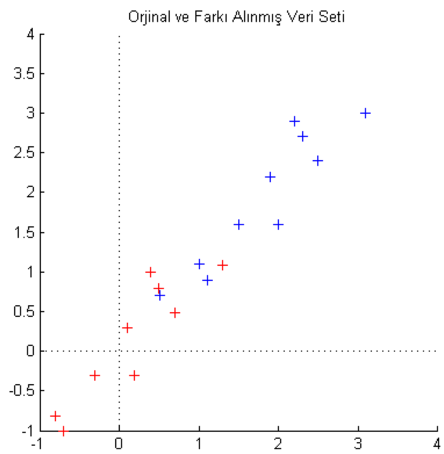
Tablo 3-1 Veri Seti

3.2. Ortalamadan Çıkarma (2.Adım)

x	y	$(x-\overline{x})$	$(y-\overline{y})$
2,5	2,4	0,69	0,49
0,5	0,7	-1,31	-1,21
2,2	2,9	0,39	0,99
1,9	2,2	0,09	0,29
3,1	3	1,29	1,09
2,3	2,7	0,49	0,79
2	1,6	0,19	-0,31
1	1,1	-0,81	-0,81
1,5	1,6	-0,31	-0,31
1,1	0,9	-0,71	-1,01

Tablo 3-2 Yeni Veri Seti

PCA ile çalışabilmek için veri setinin her bir boyutunu kendi ortalamasından çıkarmak gereklidir. Böylelikle veri setinde yer alan x ve y değerleri için $(x-\overline{x})$ ve $(y-\overline{y})$ değerleri elde edilecektir. Tablo 3-2'de elde edilen yeni veri seti görülmektedir.



Şekil 3.6 Orijinal Veri Seti ve Ortalamalarından Farkları Alınmış Yeni Veri Seti

Şekil 3.6'de orijinal veri setine ve veri setindeki her bir boyutun ortalamasından farkı alınmış yeni veri setine yer verilmiştir.

3.3. Kovaryans Matris (3.Adım)

Kovaryans matris, özdeğerler ve özvektörlerin elde edilmesi için kullanılmaktadır. Veri iki boyutlu olduğundan kovaryans matris de *2x2* boyutlu olacaktır.

$$cov(Data) = \begin{pmatrix} 0.6166 & 0.6154 \\ 0.6154 & 0.7166 \end{pmatrix}$$

Kovaryans matris hesaplandıktan sonra özvektörler ve özdeğerler bu matristen elde edilebilir.

3.4. Özdeğerler ve Özvektörler (4.Adım)

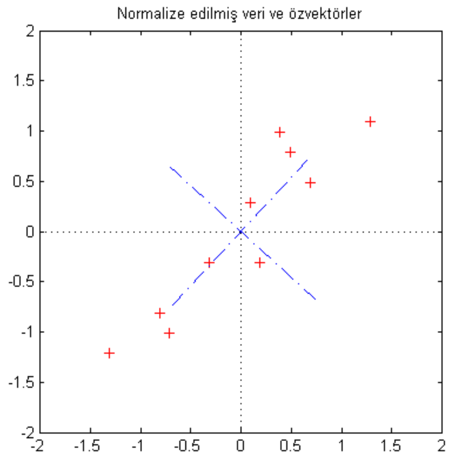
Kovaryans matris karesel bir matristir. Özvektörler ve özdeğerler bu matristen elde edilebilir. Burada önemli olan nokta bu matrisin veri seti ile ilgili verdiği bilgidir.

$$D\ (\ddot{o}zde\breve{g}erler) = \begin{pmatrix} 0.0491 \\ 1.2840 \end{pmatrix}$$

$$V\ (\ddot{o}zvekt\ddot{o}rler) = \begin{pmatrix} -0.7352 & 0.6779 \\ 0.6779 & 0.7352 \end{pmatrix}$$

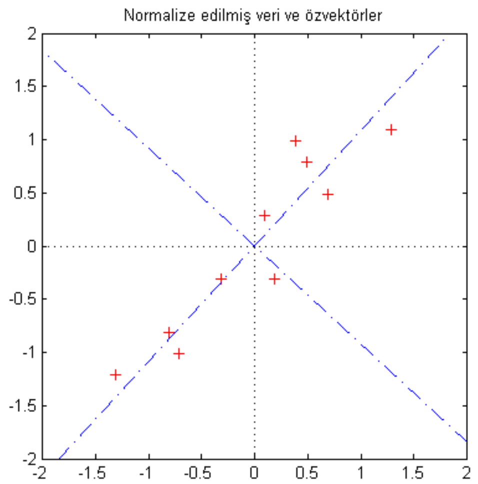
3.5. Bileşenlerin Seçilmesi ve Özellik Vektörünün Oluşturulması (5.Adım)

Kovaryans matristen elde edilen özvektör ve özdeğerler yardımıyla eksenler ve özellik vektörü elde edilecektir. Şekil 3.7'de normalize edilmiş veriler kırmızı + karakteri ile gösterilmektedir. Mavi çizgiler ile özvektörlerden elde edilmiş eksenler göstermektedir.



Şekil 3.7 Normalize Edilmiş Veri ve Özvektörler

Özvektörler hazırlanan programda 10 ile çarpılmıştır. Böylelikle eksenlerin daha belirgin olması sağlanmıştır.



Şekil 3.8 Özvektörlerin Genişletilmesi

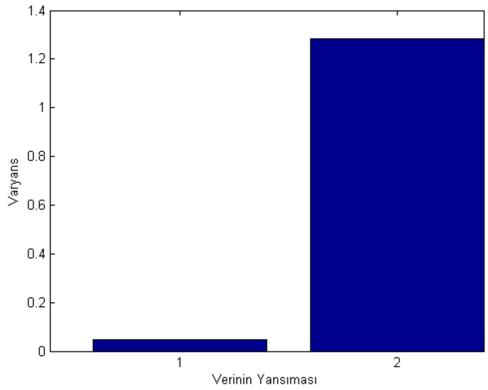
PCA, bileşenler yardımıyla veri setini temsil etmektedir. Başka bir ifadeyle veri setinin temel bileşenleri en yüksek özdeğerler ile özvektörlerden oluşmaktadır. Genellikle, özvektörler öncelikli olarak kovaryans matristen elde edilir ve daha sonra yüksek değerden düşük değere doğru sıralanır. Amaç bileşenleri veriyi temsil etme oranına göre sıralamaktır. Böylelikle en önemli bileşenden en az önemli bileşene doğru bir sıralama yapılır. Eğer bazı bileşenler atılırsa sonuçta elde edilecek veri seti orijinal veri setinden daha az boyuta sahip olabilir. Örneğin n boyutlu bir veri setinden n özvektör ve özdeğer elde edildikten sonra p kadar özvektör seçilirse sonuçta elde edilecek veri seti sadece p boyutlu olacaktır.

$$\text{Özellik Vektörü} = (\text{özvektör}_1, \text{özvektör}_2, \dots, \text{özvektör}_n)$$

3.6. Yeni Veri Setinin Türetilmesi (6.Adım)

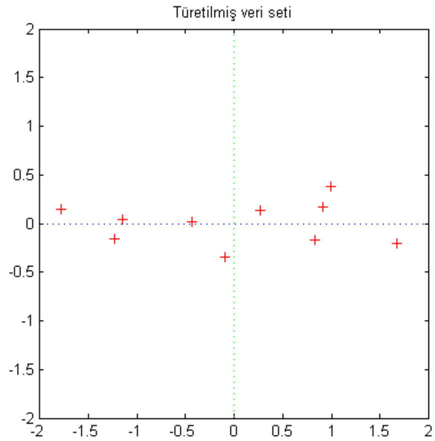
Temel bileşenler analizinin son aşamasında yeni veri seti türetilir ve en basit aşamada burasıdır. Öncelikle veri setini en iyi temsil edecek, önem derecesi en yüksek olarak seçilen bileşenler ile normalize edilmiş verinin transpozu alınarak çarpılması sonucu yeni veri seti türetilebilir.

$$\text{Yeni Veri Seti} = \text{Özellik Vektörü } x \text{ Normalize Edilmiş Veri}$$



Şekil 3.9 Özdeğerler ve Varyans

Şekil 3.9'de özdeğerler görülmektedir. Görüldüğü üzere ikinci değerin varyansı daha yüksektir. Bir başka ifadeyle özvektörlerin iki numaralı bileşeni veri daha iyi temsil etmektir. Bu nedenle öncelikle özvektörlerin 2 numaralı kolonu işleme alınacak daha sonra da bir numaralı kolonu işleme alınacaktır. Bu bilgi bileşenlerin önem sırasının belirlenmesi açısından önemlidir.



Şekil 3.10 Türetilmiş Veri Seti

Veri setindeki varyans değerleri dikkate alınarak bileşenler seçildikten sonra yeni eksen üzerine veriler yansıtılmıştır. Böylelikle iki boyutlu bir veri setine PCA uygulanmıştır.

3.7. Matlab Kodları

Uygulama MATLAB ortamında yazılmıştır. Kaynak kodlar aşağıdaki gibidir.

```
clear all,close all, clc
%% Adım 1: Veri Setinin hazırlanması
Data = [2.5 2.4;
    0.5 0.7;
    2.2 2.9;
    1.9 2.2;
    3.1 3.0;
    2.3 2.7;
    2 1.6;
    1 1.1;
    1.5 1.6;
    1.1 0.9];

%% Adım 2: Ortalamadan Çıkarma
figure
m = mean(Data); % Veri setinin her bir boyutunun ortaması alındı.
hold on
plot([0 0],[-1 4],'k:');
plot([-1 4], [0 0],'k:');
title('Orjinal Veri Seti');

plot(Data(:,1),Data(:,2)','+');
axis equal
axis([-1 4 -1 4])

DA(:,1)=Data(:,1)-m(1);
DA(:,2)=Data(:,2)-m(2);

plot(DA(:,1),DA(:,2),'r+');
title('Orjinal ve Farkı Alınmış Veri Seti');
%% Adım 3: Kovaryans matrisin hesaplanması
CM = cov(Data);
%% Adım 4: Özvektör ve özdeğerlerin hesaplanması
[V D] = eig(CM);

%% Adım 5: Bileşenlerin seçilmesi ve özellik vektörünün oluşturulması
figure
plot(DA(:,1),DA(:,2),'r+');
A = 10* V;
hold on
axis equal
axis([-2 2 -2 2])
plot([0 0],[-2,2],'k:');
plot([-2 2],[0,0],'k:');
plot([-A(1,1) A(1,1)],[-A(1,2) A(1,2)],'b-.')
plot([-A(2,1) A(2,1)],[-A(2,2) A(2,2)],'b-.')
title('Normalize edilmiş veri ve özvektörler');
%% Adım 6: Yeni Verinin Türetilmesi
f1 = V(:,2)';
% f1 = [-0.677873399 -0.735178656];
PC1 = f1*DA'
PC1 = PC1'
f2 = V(:,1)';
%f2 =[-0.73518656 0.677873399];
PC2 = f2*DA';
PC2 = PC2'

F = [f1; f2];
Y = [PC1 PC2];
figure
```

```
plot(Y(:,1),Y(:,2),'r+')
axis equal
axis([-2 2 -2 2])

Cy = cov(Y);
[Vy Dy] = eig(Cy);
hold on
A = 10 * Vy';
plot([-A(1,1) A(1,1)],[-A(1,2) A(1,2)],'g:');
plot([-A(2,1) A(2,1)],[-A(2,2) A(2,2)],'b:');
title('Türetilmiş veri seti');
% Variance
figure;
hold off
bar(diag(D));
xlabel('Veri boyutunun yansıması');
ylabel('Varyans');
```

4. Temel Bileşenler Analizi ile Beyaz Gürültülü Sinüzoidal Dalganın Gürültü Probleminin Giderilmesi

Bu çalışmada bir sinüzoidal sinyal üretilmiş ve üretilen sinyale bir miktar gürültü eklenmiştir. Satır şeklindeki sinyal, temel bileşenler analizinde kullanılmak üzere bir yığın yapısına dönüştürülmüş ve bileşen sayısı belirlenerek temel bileşenler analizinin uygulanması sağlanmıştır. Son olarak seçilen bileşen sayısına bağlı olarak satır sinyal formatına dönülmüş ve sinyal gürültüsünün giderilmesi sağlanmıştır.

4.1. Giriş

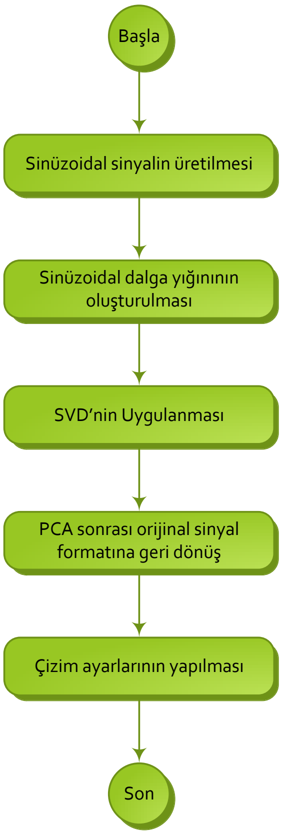
Sinyal (işaret): zaman, uzay ya da başka bir veya birkaç bağımsız değişken ile değişiklik gösteren fiziksel nicelik olarak tanımlanabilir. Matematiksel olarak, bir sinyal bir ya da daha fazla bağımsız değişkenin fonksiyonu olarak tanımlanabilir. Bir başka ifadeyle işaretler, fiziksel bir durum hakkında bilgi taşıyan, bir veya daha fazla değişkene bağlı fonksiyonlardır.

Doğadaki pek çok sinyal fonksiyonel bir ifadeyle tanımlanamaz. Genel olarak bir konuşma sinyali için farklı genlikler ve farklı frekanslarda birkaç sinüzoidal frekansın toplamı olarak aşağıdaki gibi yazılabilir:

$$\sum_{i=1}^{N} A_i(t)\sin[2\pi F_i(t)t + \theta_i(t)]$$

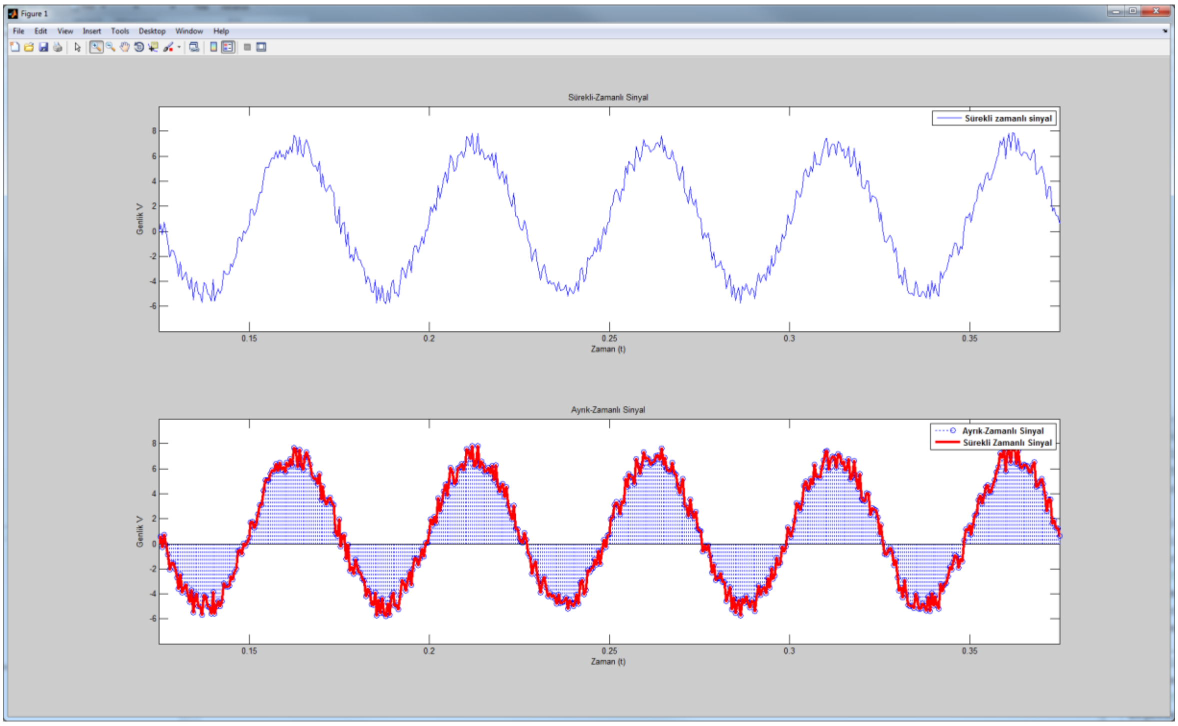
- $A_i(t)$: Sinüzoidal dalganın genliği
- $F_i(t)$: Sinüzoidal dalganın frekansı
- $\theta_i(t)$: Sinüzoidal dalganın faz kümesini temsil etmektedir.

4.2. Programın Akış Şeması



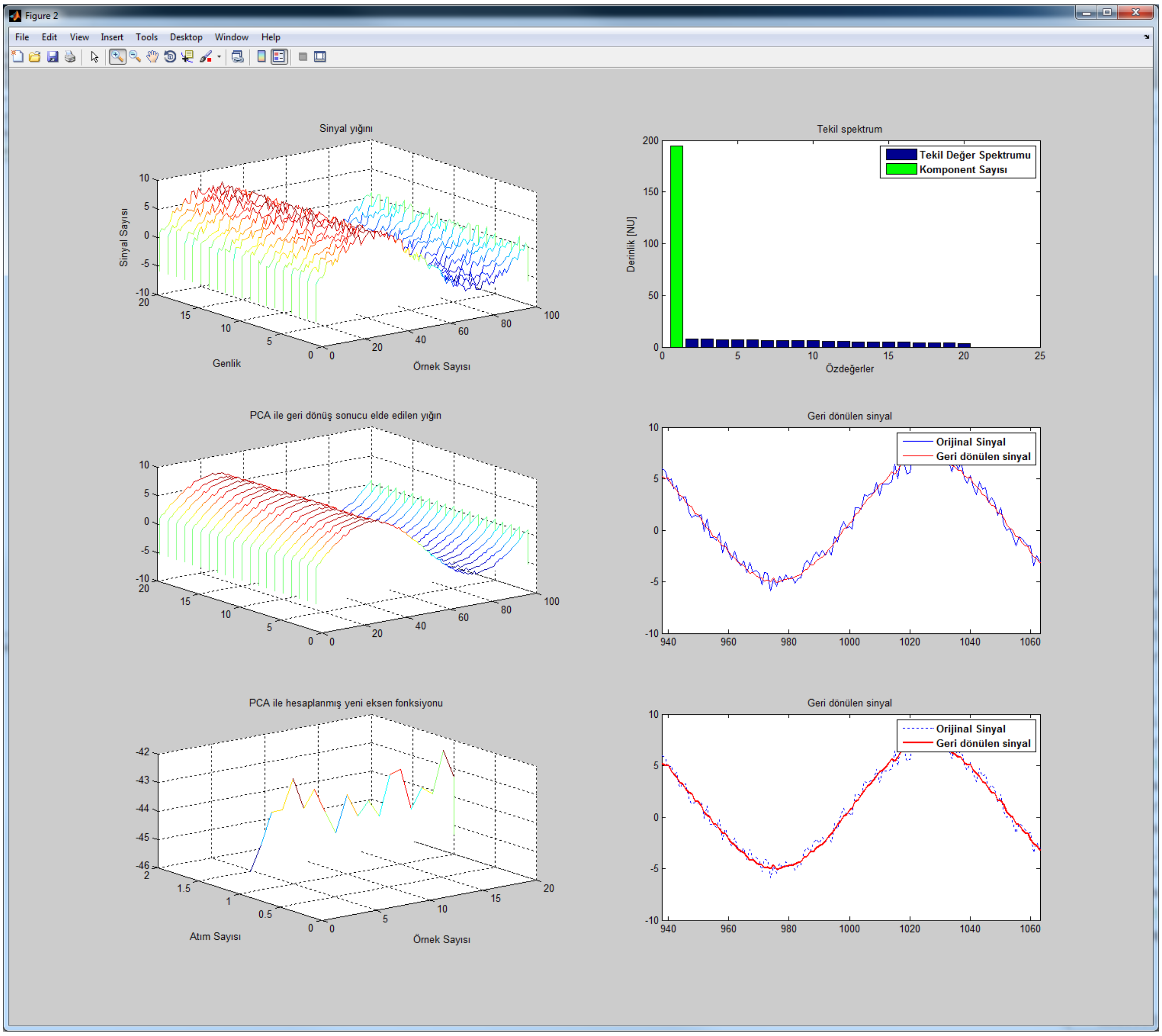
Şekil 4.1 Programın akış şeması

Çalışmada yukarıda tanımlı olan her bir parametrenin kullanıcı tercihine bağlı olarak alındığı bir yapı oluşturulmuş ve bir sinüzoidal sinyalin gürültü ile birlikte oluşturulması sağlanmıştır. Geliştirilen programa ait akış şeması Şekil 4.1'de görülmektedir.



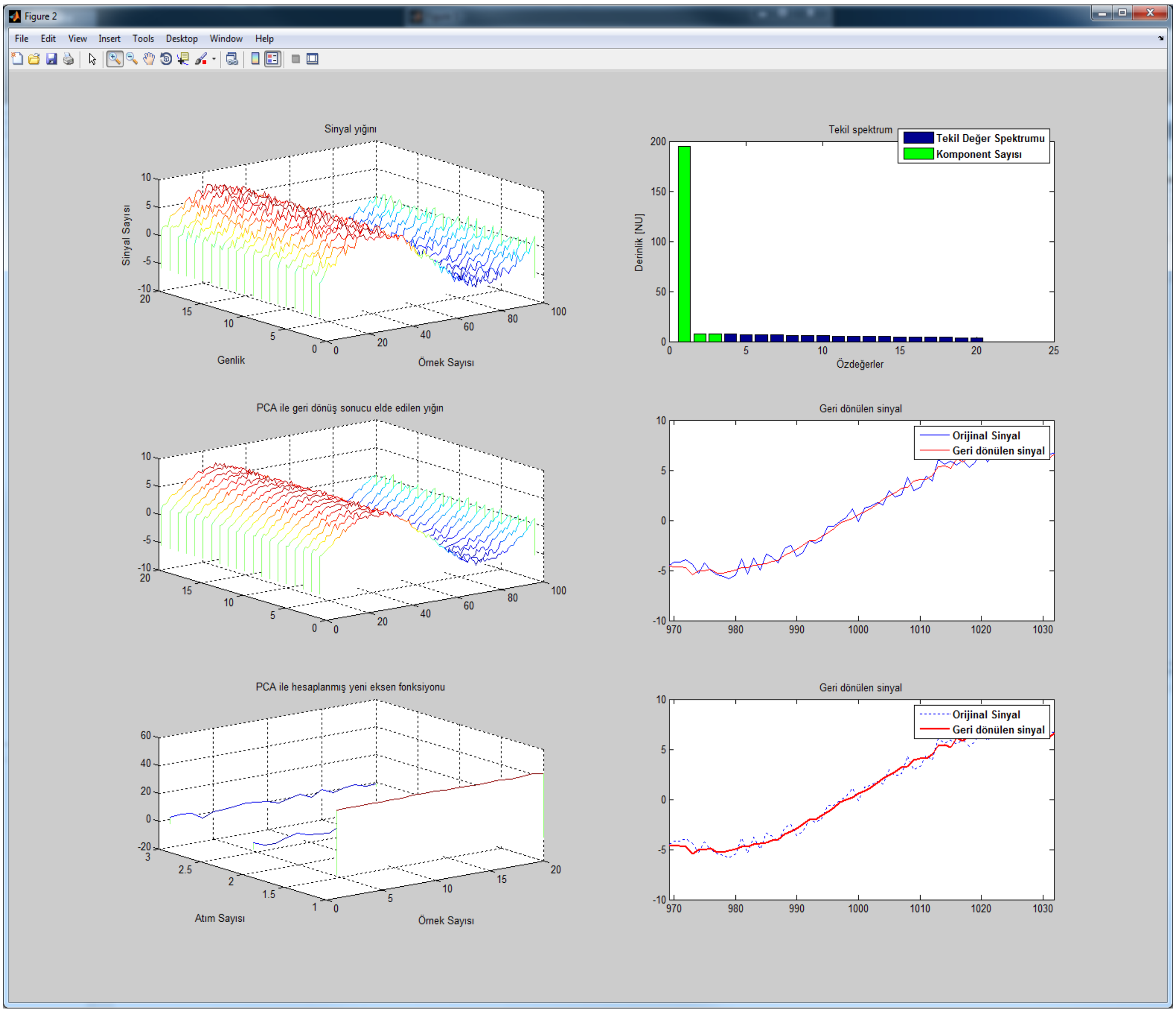
Şekil 4.2 Üretilen gürültülü sinüzoidal sinyal

Şekil 4.2'de üretilen sinyal görülmektedir. Sinyale ait tanımlamalara bağlı olarak gürültülü bir sinyal oluşturulmuştur.



Şekil 4.3 PCA'nın tek bileşen kullanılması ile elde edilen uygulanma sonuçları

Şekil 4.3'de ilk figür gürültülü sinyal yığınını göstermektedir. Görüldüğü gibi 20 yığından oluşan sinyalin tek bir bileşeni alınarak geriye kalan bileşenlerin atılması sağlanmıştır. Bileşenlerin her biri veriyi temsil etme gücünü göstermektedir ve azalan bir sıralayama sahiptir. Eğer tüm bileşenler alınarak geri dönüşüm yapılırsa kayıpsız olarak orijinal sinyal yeniden elde edilecektir. Amacımız sinyal üzerindeki gürültüleri atmak olduğunda ve birinci bileşenin sinyali ciddi derecede temsil ettiği görüldüğünden sadece tek bileşen kullanılarak sonuçlar elde edilmiştir. PCA ile hesaplanmış yeni eksen fonksiyonu çizdirilmiştir. Son olarak orijinal sinyal ve gürültüsü temizlenmiş sinyal gösterilmiştir. Şekil 4.4'de PCA hesabı için üç bileşen kullanılmıştır. Sonuçlar kıyaslandığında tek bileşenli uygulamanın daha fazla veri kaybı yaşadığı ve sonuç olarak elde edilen sinyalin daha temiz olduğu söylenebilir.



Şekil 4.4 PCA'nın üç bileşen kullanılması ile elde edilen uygulanma sonuçları

5. Sentetik Elektrokardiyografi Sinyalleri Üzerinde QT Aralığının Belirlenmesi

Elektrokardiyografi, kardiyovasküler rahatsızlıklara tanı koyma ve değerlendirme amacıyla yaygın olarak kullanılmaktadır. Kalp hızı, ventriküler aksiyon potansiyelinin en önemli belirleyicisidir. Bu nedenle QT aralığı kalp hızıyla ters orantılı olacak şekilde değişmektedir. QT aralığının hesaplanma ve düzeltme gereği bu durumdan kaynaklanmaktadır. Bu çalışmada, sentetik elektrokardiyografi sinyalleri, bir önişleme sürecinden geçirilmiş ve temel bileşenler analizi ile sinyal üzerindeki gürültülerin azaltılması sağlanmıştır. Daha sonra sinyal özellikleri çıkarılıp son olarak QT aralıkları Bazett yöntemiyle ile belirlenmiştir.

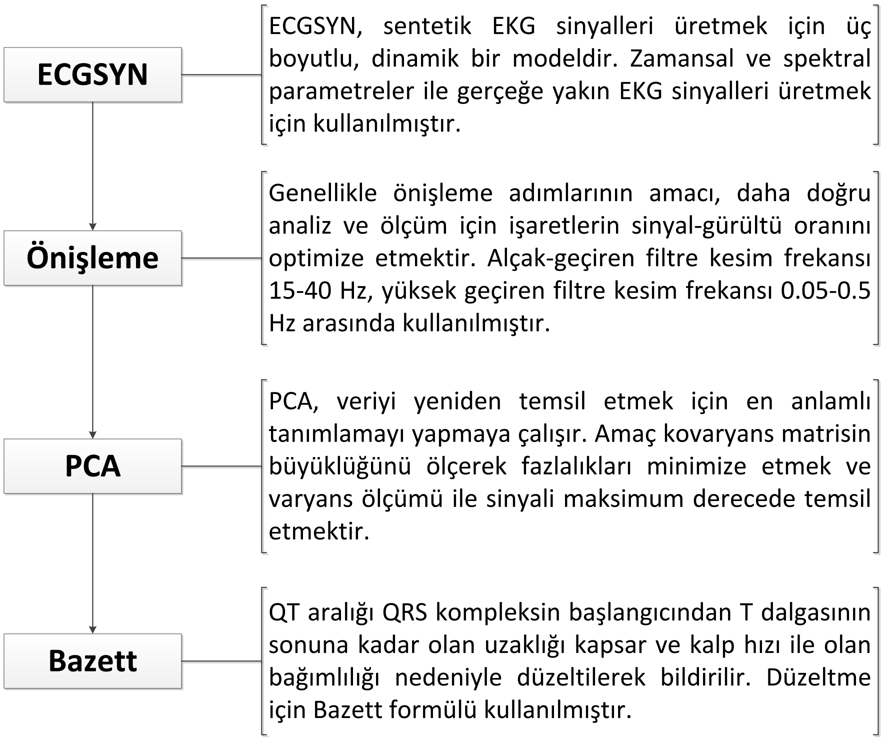
Anahtar Kelimeler — QT Analizi, Bazett, Temel Bileşenler Analizi

Bu bölümde IEEE 23. Sinyal İşleme ve İletişim Uygulamaları Kurultayı'nda bildiri olarak sunduğumuz ve temel bileşenler analizini kullandığımız bilimsel bir çalışmanın bir bölümüne yer vereceğim.

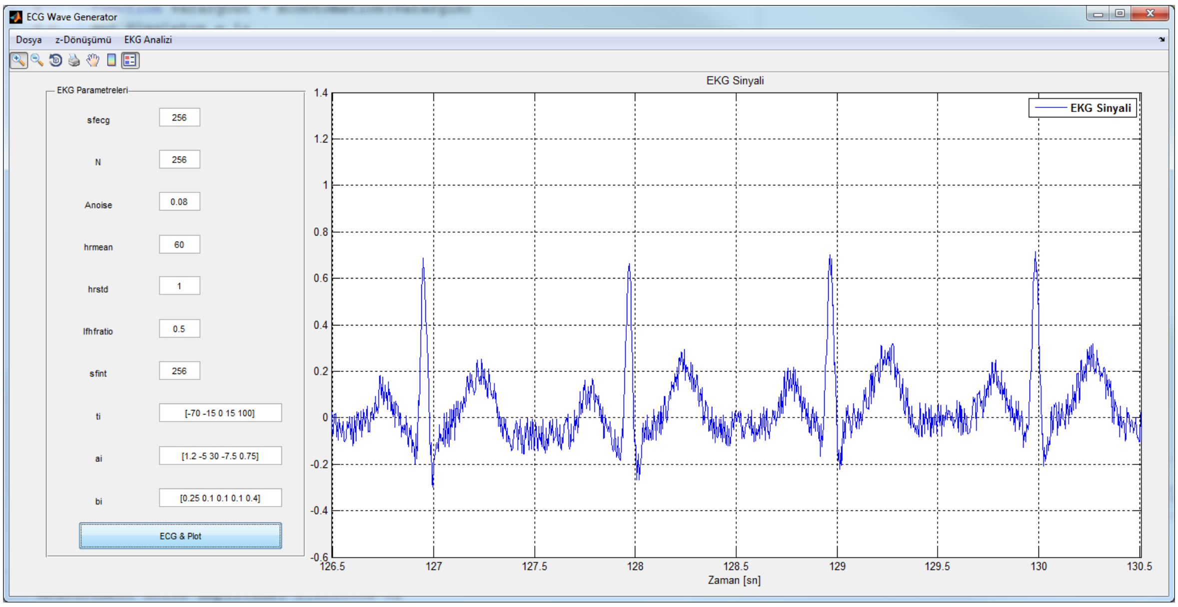
Şekil 5.1'de programın genel fonksiyonların blok diyagram olarak gösterilmiş ve her bir blok içerisinde yapılanlar kısaca özetlenmiştir.

Şekil 5.2'de sentetik elektrokardiyografi (EKG) sinyalleri üretmek için tasarlanan zamansal ve spektral parametrelerin girilebildiği bir EKG dalga üreteci gösterilmektedir. Bu arayüz işlenecek sinyalin üretildiği yeri ifade etmektedir.

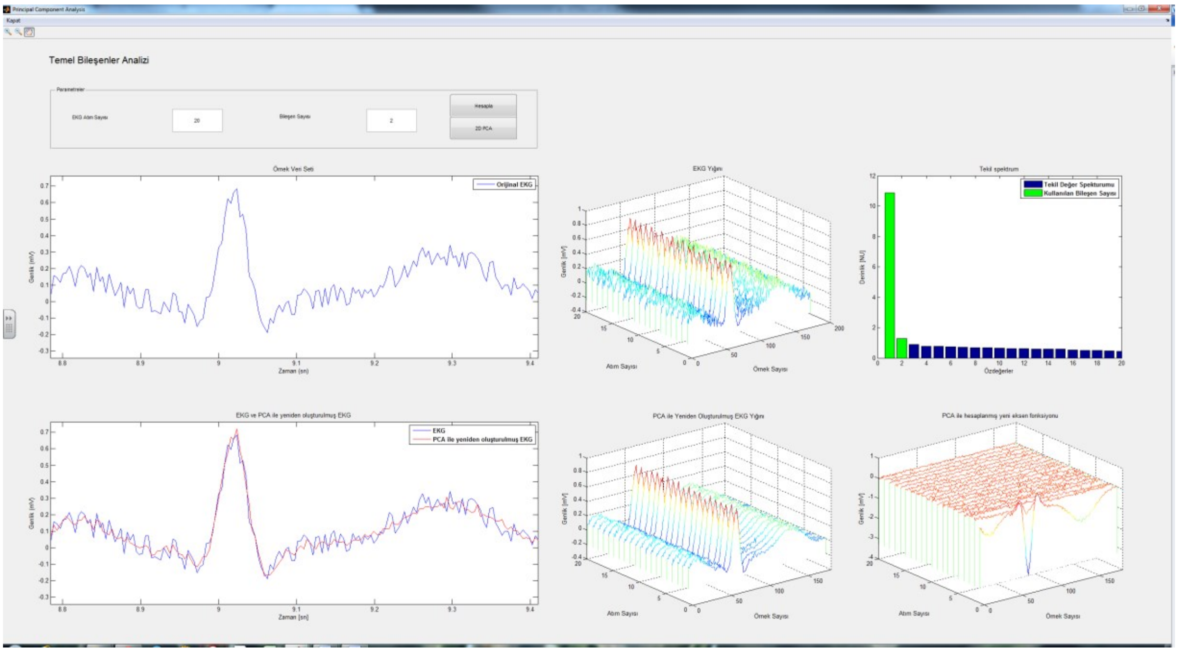
Üretilen sinyal bir önişleme sürecinden geçirildikten sonra PCA ile sinyal üzerindeki gürültülerin giderilmesi sağlanmıştır ve Şekil 5.3'de gösterilmiştir.



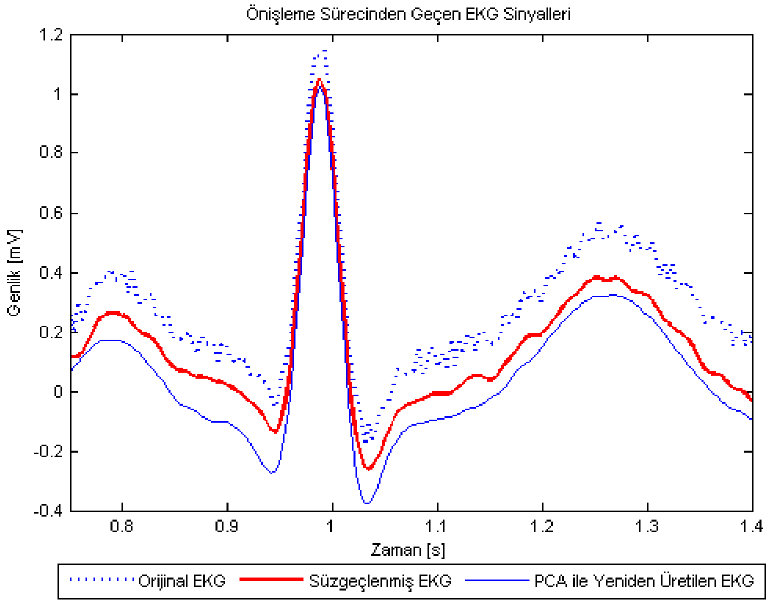
Şekil 5.1 Program blokları



Şekil 5.2 Sentetik EKG dalga üreteci



Şekil 5.3 EKG'de gürültü giderme için PCA kullanımı



Şekil 5.4 Önişleme ve PCA sonrası EKG sinyalinin durumları

$$\mathrm{EKG} = \mathrm{USV}^T$$

Denklemde EKG, PCA sonrası elde edilmiş sinyali, S giriş sinyali ile aynı boyutta, elemanların büyükten küçüğe sıralandığı ve pozitif değerler taşıdığı

($[S_1, S_2, \ldots, S_n]_{diag}$, $S_1 > S_2 > \cdots > S_n$) diyagonal matrisi, U ve V ise üniter matrisleri temsil etmektedir.

Bir EKG sinyalinin ilk üç-beş temel bileşeni alınarak aşağıdaki denklemde hesaplandığı gibi işaret %90'ların üzerinde temsil edilebilmektedir. Şekil 5.3'de PCA ile yeniden üretilmiş EKG yığını görülmektedir ve orijinal EKG sinyalini temsil etme oranı 0.976 olarak hesaplanmıştır. PCA'dan orijinal sinyale dönüş için sadece dört temel bileşen kullanılmıştır.

$$\frac{s_1^2 + s_2^2 + s_3^2 + s_4^2}{\sum_{i=1}^{n} s_i^2} > 0.97$$

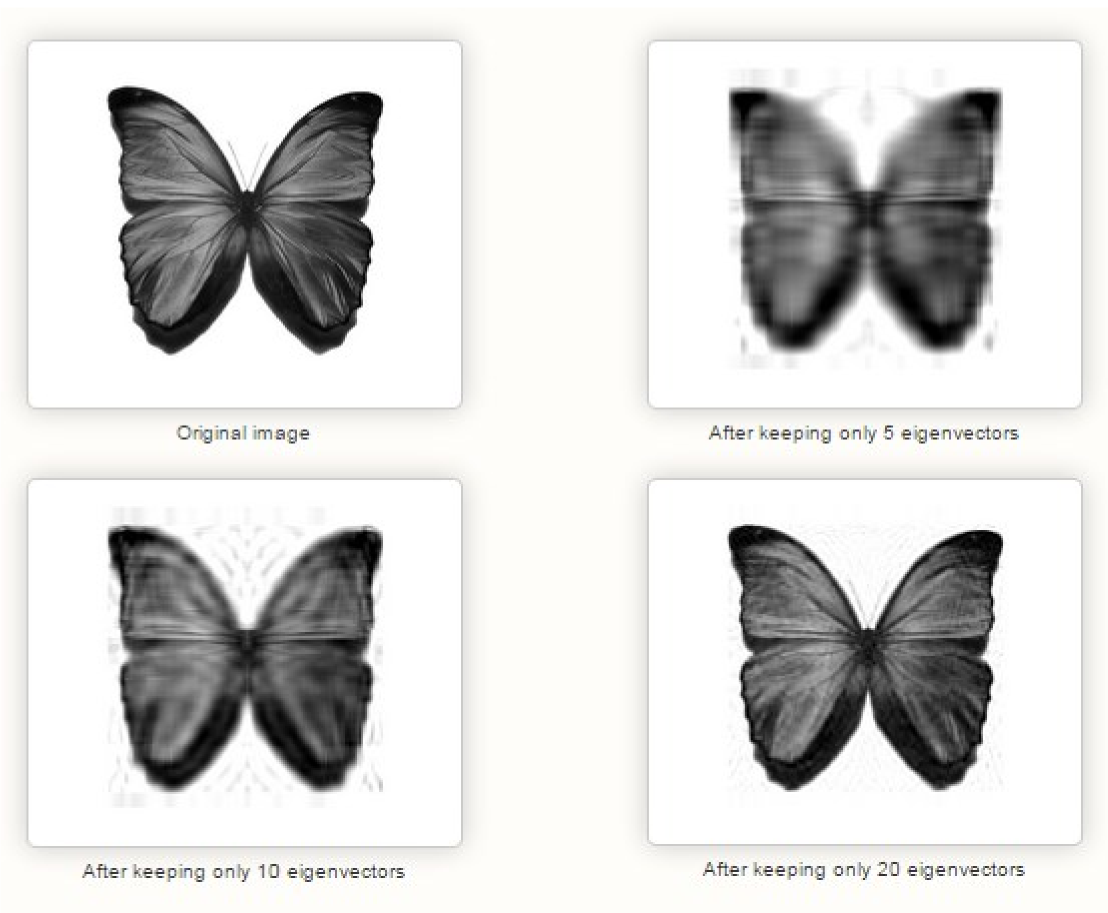
Denklemde s_i değerlerinin her biri S matrisinin elemanlarını temsil etmektedir ve tekil değerleri tutmaktadır.

6. PCA ile İmge Sıkıştırma

Bu çalışmada imge sıkıştırmak için Matlab kaynak kodlarına yer verilmiştir. İmge sıkıştırma yöntemi olarak temel bileşenler analizi kullanılmıştır.

6.1. Giriş

Çalışma Ammar's Technical Realm[3]'den alınmıştır. Çalışmanın orijinal haline ilgili web sayfasından erişilebilir.



Şekil 6.1 Öz vektör sayısına göre imgeden elde edinilen sonuçlar

[3] http://ecammar.blogspot.com.tr/2013/04/image-compression-using-principal-of.html

6.2. İmge Sıkıştırma Matlab Kodları

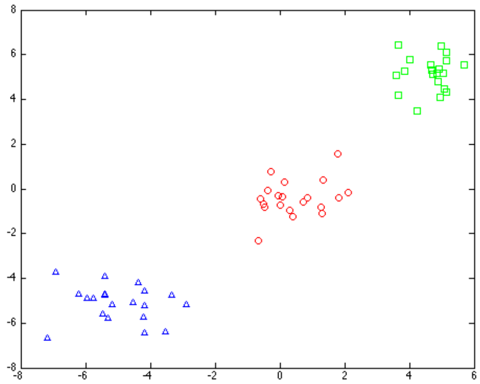
```
%% Source
% http://ecammar.blogspot.com/2013/04/image-compression-using-principal-of.html
clear all;clc;

% Start of PCA code,
Data = imread('butterfly.jpg');
Data_gray = rgb2gray(Data);
Data_grayD = im2double(Data_gray);
figure,
set(gcf,'numbertitle','off','name','Grayscale Image'),
imshow(Data_grayD)
Data_mean = mean(Data_grayD);
[a b] = size(Data_gray);
Data_meanNew = repmat(Data_mean,a,1);
DataAdjust = Data_grayD - Data_meanNew;
cov_data = cov(DataAdjust);
[V, D] = eig(cov_data);
V_trans = transpose(V);
DataAdjust_trans = transpose(DataAdjust);
FinalData = V_trans * DataAdjust_trans;
% End of PCA code

% Start of Inverse PCA code,
OriginalData_trans = inv(V_trans) * FinalData;
OriginalData = transpose(OriginalData_trans) + Data_meanNew;
figure,
set(gcf,'numbertitle','off','name','RecoveredImage'),
imshow(OriginalData)
% End of Inverse PCA code

% Image compression
PCs=input('Enter number of PC colomuns needed?  ');
PCs = b - PCs;
Reduced_V = V;
for i = 1:PCs,
Reduced_V(:,1) =[];
end
Y=Reduced_V'* DataAdjust_trans;
Compressed_Data=Reduced_V*Y;
Compressed_Data = Compressed_Data' + Data_meanNew;
figure,
set(gcf,'numbertitle','off','name','Compressed Image'),
imshow(Compressed_Data)
% End of image compression
```

7. PCA'nın Makine Öğrenmesinde Kullanımı

Bu çalışmada, bir PCA örneğinin makine öğrenmesinde kullanımına dair uygulama yer almaktadır.

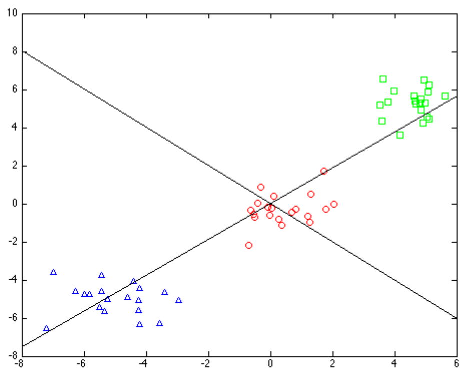
7.1. Çalışmanın Adımları

1. Verilerin oluşturulması
2. Orijinal verilerin çizdirilmesi



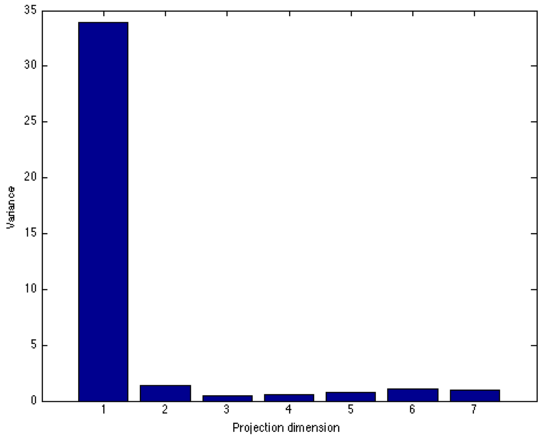
Şekil 7.1 Orijinal veri seti

3. PCA'nın uygulanması
4. İlk iki bileşenin orijinal veri setine yansıtılması



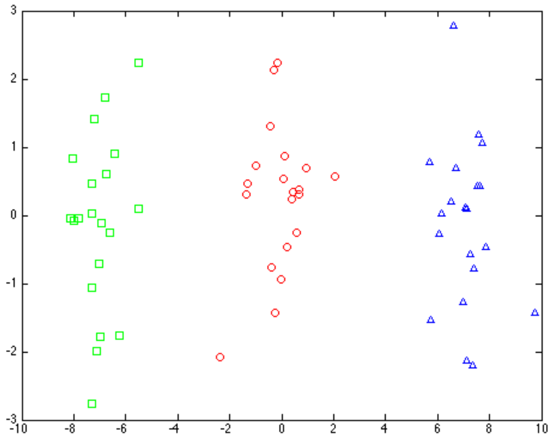
Şekil 7.2 Orijinal veri setine ilk iki bileşenin yansıtılması

5. Özdeğerlerin çubuk grafik şeklinde verilmesi



Şekil 7.3 Özdeğerlerin çubuk grafik şeklinde verilmesi

6. Verilerin ilk iki bileşendeki eksen yansıtılması



Şekil 7.46. Verilerin ilk iki bileşendeki eksen yansıtılması

Çalışmanın kaynak kodları için ilgili web sayfasını ziyaret edilebilir[4].

[4] http://www.dcs.gla.ac.uk/~srogers/firstcourseml/matlab/chapter7/pcaexample.html#2

7.3. Matlab Kodları

```
% Source
%http://www.dcs.gla.ac.uk/~srogers/firstcourseml/matlab/chapter7/pcaexample.html
clear all;close all;
% Generate the data
Y = [randn(20,2);randn(20,2)+5;randn(20,2)-5];
% Add 5 random dimensions
N = size(Y,1);
Y = [Y randn(N,5)];

% labels - just used for plotting
t = [repmat(1,20,1);repmat(2,20,1);repmat(3,20,1)];

% Plot the original data
symbs = {'ro','gs','b^'};
figure(1);hold off
for k = 1:3
    pos = find(t==k);
    plot(Y(pos,1),Y(pos,2),symbs{k});
    hold on
end
% Do the PCA
% Subtract the means

Y = Y - repmat(mean(Y,1),N,1);
% Compute covariance matrix
C = (1/N)*Y'*Y;
% Find the eigen-vectors/values
% columns of w correspond to the projection directions
[w,lam] = eig(C);
% Plot the first two components on to the original data
figure;
hold off
for k = 1:3
    pos = find(t==k);
    plot(Y(pos,1),Y(pos,2),symbs{k});
    hold on
end

xl = xlim;
for k = 1:2
    plot(xl,xl*w(1,k)/w(2,k),'k');
end

% Bar plot of the eigenvalues
figure;
hold off
bar(diag(lam));
xlabel('Projection dimension');

% Plot the data projected into the first two dimensions
 X = Y*w(:,1:2);
figure;
hold off
for k = 1:3
    pos = find(t==k);
    plot(X(pos,1),X(pos,2),symbs{k});
    hold on
end
ylabel('Variance');
```

Kaynakça

[1] Vikipedi, «Vikipedi,» 01 12 2013. [Çevrimiçi]. Available: http://tr.wikipedia.org/wiki/%C4%B0statistik. [%1 tarihinde erişilmiştir25 03 2014].

[2] Vikipedi, «Vikipedi,» 20 05 2013. [Çevrimiçi]. Available: http://tr.wikipedia.org/wiki/Kovaryans. [%1 tarihinde erişilmiştir25 03 2014].

[3] L. I. Smith, «A tutorial on Principal Components Analysis,» 2002.

[4] İ. Arı, «Tekil Değer Ayrışımı,» 4 10 2010. [Çevrimiçi]. Available: http://ismailari.com/blog/tekil-deger-ayrisimi-1/. [%1 tarihinde erişilmiştir12 12 2013].

[5] Ankara Üniversitesi, «Ankara Üniversitesi,» [Çevrimiçi]. Available: http://80.251.40.59/science.ankara.edu.tr/ozturk/Dersler/ist312/Ders4/Matris2.pdf. [%1 tarihinde erişilmiştir25 12 2013].

[6] B. E. Demiröz, «Temel Bileşenler Analizi (PCA) çıkarımı,» 05 02 2012. [Çevrimiçi]. Available: http://www.anlak.com/temel-bilesenler-analizi-pca-cikarimi/. [%1 tarihinde erişilmiştir25 12 2013].

[7] İstatistik Eğitim ve Bilgilendirme Sitesi, «Stat,» 2013. [Çevrimiçi]. Available: http://www.stat.gen.tr/index.php?istek=sinif&dersid=ist01&konuid=pca&max=1. [%1 tarihinde erişilmiştir25 12 2012].

Belge Geçmişi

21.05.2015	Belge ilk kez oluşturuldu.