

KİSMİ EN KÜÇÜK KARELER ALGORİTMALARINDAN NIPALS İLE SIMPLS'İN TANITIMI VE BİR UYGULAMA

Elif BULUT*

Aylin ALIN**

ÖZET

Kısmi en küçük kareler (KEKK) yöntemi, aralarında çoklu doğrusal bağlantı bulunan açıklayıcı değişkenler içerisinde, bu değişkenlerdeki değişimin büyük kısmını açıklayacak, dik ve daha az sayıda gizli değişkeni (bileşen) bulmakla ilgilendir. Farklı alanlarda farklı amaçlar doğrultusunda kullanımından dolayı, KEKK için farklı algoritmalar ortaya çıkarılmıştır. Bu çalışmada KEKK için en temel algoritma olan NIPALS, 1993 yılında De Jong tarafından geliştirilen ve yaygın olarak kullanılan SIMPLS algoritmalarına değinilmiş, bir örnek üzerinde uygulama gösterilmiştir.

Anahtar Kelimeler: Çoklu Doğrusal Bağlantı, Kısmi En Küçük Kareler, NIPALS, SIMPLS.

1. GİRİŞ

1960'lı yıllarda Herman Wold tarafından geliştirilen KEKK yöntemlerinin yaygın olarak kullanımına, kimya alanında oğlu Swante Wold tarafından başlanmıştır. Randall (1996)'ın da belirttiği gibi kimya alanında kalibrasyon modellerini kurmak için birçok kimyager tarafından da kullanılan bu yöntem; endüstri, tarım, tıp ve gıda gibi birçok alandada uygulanmaktadır.

KEKK yöntemi, boyut indirgemesi, sınıflama ve regresyon aşamaları gibi birçok aşamayı barındıran çok değişkenli istatistiksel bir yöntemdir. Değişken sayısının gözlem sayısından çok olduğu ve değişkenler arasında çoklu doğrusal bağlantı sorununun olduğu durumlarda rahatlıkla kullanılabilen bu yöntem, bağımlı ve açıklayıcı değişkenler matrisi olmak üzere iki veri matrisiyle de ilgilenmektedir. KEKK1 tek bağımlı değişken ve KEKK2 çok bağımlı değişken durumlarında kullanılabilir.

Burada amaç, bağımlı değişkeni tahmin etmeyi sağlayacak, bağımlı ve açıklayıcı değişkenlerdeki birlikte değişimin büyük bir kısmını açıklayacak, açıklayıcı değişkenden

* Araş. Gör., Dokuz Eylül Üniversitesi Fen-Edebiyat Fak., İstatistik Böl., Buca İzmir.

e-posta: elif.bulut@deu.edu.tr

** Yard. Doç. Dr., Dokuz Eylül Üniversitesi Fen-Edebiyat Fak., İstatistik Böl., Buca İzmir.

e-posta: aylin.alin@deu.edu.tr

daha az sayıda yeni gizli bileşenler bulmaktır. Elde edilen bileşenler aynı zamanda bağımlı değişkendeki değişimi de açıkladığından, yöntem diğer çok değişkenli yöntemlerden ayrılmaktadır.

Modellemede kullanılan bu bileşenler, $X = [X_1, X_2, \dots, X_k]$ açıklayıcı değişkenler matrisi ve $Y = [Y_1, Y_2, \dots, Y_m]$ bağımlı değişkenler matrisi olmak üzere $T = XP$ ve $U = YC$ eşitlikleri ile elde edilir. Burada T ve P sırasıyla X değişken matrisi için bileşeni ve açıklayıcı değişkenin bileşendeki yükünü ifade eden matrislerdir. U ve C ise sırasıyla Y değişken matrisi için bileşeni ve bağımlı değişkenin bileşendeki ağırlığını ifade eden matrislerdir.

KEKK yinelemeli bir süreçtir. Farklı alanlarda, farklı veri yapıları ile çalışıldığından birçok farklı algoritma geliştirilmiştir. Lindgren ve Rännar (1998)'in çalışmalarında belirttiği gibi; NIPALS, SIMPLS, UNIPAL, SAMPLS ve KERNEL algoritmaları bunlardan birkaçıdır. Çalışmada NIPALS ve SIMPLS algoritmaları kısaca anlatılarak, Ondokuz Mayıs Üniversitesi Tıp Fakültesi'nden alınan veri kümesine algoritmalar uygulanarak, sonuçlar verilmektedir. Bu çalışmada matrisler büyük, italik ve kalın harf ile vektörler küçük, italik ve kalın harf ile gösterilmiştir. “” simgesi ise matrisin veya vektörün devriğini (transpozunu) göstermektedir.

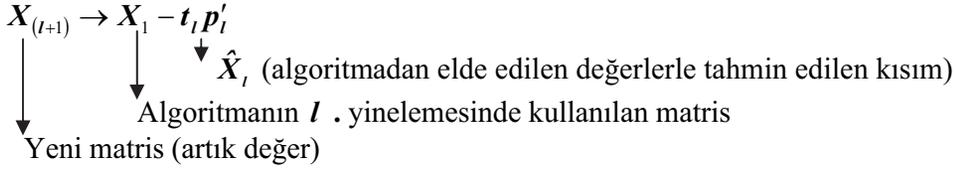
2. NIPALS VE SIMPLS ALGORİTMALARI

2.1 NIPALS Algoritması

“Klasik, standart” algoritma olarak da nitelendirilen NIPALS (Non-Linear Iterative Partial Least Squares; Doğrusal olmayan yinelemeli kısmi en küçük kareler) KEKK'nın ilk şeklini oluşturmaktadır ve 1970'lerin başında geliştirilmiştir. NIPALS algoritması farklı yazarlar tarafından ele alınmıştır. Helland (1988), Geladi (1986), Kowalski (1986) ve Höskuldsson (1988), 1980'li yıllarda bu konuda çalışan en önemli isimlerdendir. Bu çalışmada Höskuldsson (1988) tarafından verilen algoritma ele alınmıştır. Algoritmanın başlangıç noktası $n \times k$ boyutlu X açıklayıcı değişkenler matrisi ve $n \times m$ boyutlu Y bağımlı değişkenler matrisidir. Algoritmaya başlamadan önce farklı ölçümlerden kaynaklanan çeşitliliği ortadan kaldırmak için sıfır ortalama ve birim varyansa sahip olacak şekilde matrisler standartlaştırılır. Bu algoritma yinelemeli bir algoritmadır. Algoritmada yer alan l indisi yineleme sayısını göstermektedir. Yinelemeler bütün bileşenler elde edilinceye kadar sürmektedir. Bir başka ifade ile A bileşen sayısı olmak üzere, $l = 1, 2, \dots, A$ 'dır. İlk yinelemede $X_1 = X$, $Y_1 = Y$ alınarak algoritma aşağıdaki 12 adım izlenerek çalıştırılır.

1. Y matrisinin ilk sütunu veya en yüksek varyansa sahip olan sütunu u_l vektörü olarak tanımlanır.
2. X 'in, Y 'nin ilgili bileşeni (u) üzerine regresyonundan X ve u arasında en büyük kovaryansı sağlayacak w ağırlık vektörü $w_l = X'_l u_l / (u'_l u_l)$ eşitliği ile elde edilir.

3. $\mathbf{w}_i / \|\mathbf{w}_i\|$ ile \mathbf{w} ağırlık vektörü normuna bölünerek boyu bir olacak şekilde ölçeklendirilir. $\|\mathbf{w}_i\|$, \mathbf{w}_i ağırlık vektörü için normu ifade edip $\|\mathbf{w}_i\| = \sqrt{\mathbf{w}_i' \mathbf{w}_i}$ ile hesaplanmaktadır.
 4. \mathbf{Y} değişkenini tahmin etmede kullanılan ve $\mathbf{t}_i = \mathbf{X}_i \mathbf{w}_i$ eşitliği ile \mathbf{w} ağırlıkları ve \mathbf{X} matrisinin doğrusal birleşiminden oluşan \mathbf{t} bileşen vektörü tanımlanır.
 5. \mathbf{Y} 'nin, \mathbf{X} 'in ilgili bileşeni (\mathbf{t}) üzerine regresyonundan \mathbf{Y} matrisi için ağırlıkları verecek olan \mathbf{c} vektörü $\mathbf{c}_i = \mathbf{Y}_i \mathbf{t}_i / (\mathbf{t}_i' \mathbf{t}_i)$ eşitliği ile elde edilir.
 6. $\mathbf{c}_i / \|\mathbf{c}_i\|$ ile \mathbf{c} ağırlık vektörü normuna bölünerek boyu bir olacak şekilde ölçeklendirilir. \mathbf{c}_i 'nin normu $\|\mathbf{c}_i\| = \sqrt{\mathbf{c}_i' \mathbf{c}_i}$ ile hesaplanır.
 7. Önceki adımda hesaplanan \mathbf{c} ağırlık vektörü, $\mathbf{Y}_i \mathbf{c}_i / (\mathbf{c}_i' \mathbf{c}_i)$ bölümüyle \mathbf{u}_{yeni} bileşen vektörünü tanımlamakta kullanılmaktadır. Elde edilen değerler, \mathbf{Y} matrisi ile \mathbf{c} ağırlık vektör değerleri nin doğrusal bir kombinasyonudur.
 8. Yedinci adımda elde edilen \mathbf{u}_{yeni} bileşen vektörü ile ilk adımda kullanılan \mathbf{u} vektörü arasında bir yakınsama sağlanıp sağlanmadığına bakılır. Eğer bu iki değer arasındaki farkın normu 10^{-6} gibi bir değere yakınsıyorsa, algoritma sonraki adımlara devam edilerek bitirilir. Aksi takdirde yedinci adımda elde edilen \mathbf{u}_{yeni} vektörü, ikinci adımda yerine koyularak tekrar adım adım algoritmaya devam edilir.
 9. $\mathbf{X}_i' \mathbf{t}_i / (\mathbf{t}_i' \mathbf{t}_i)$ ile \mathbf{X} değişken matrisinin ilgili bileşeni üzerine regresyonundan \mathbf{p}_i yük vektörü elde edilir. Bu değerler ilgili temel bileşenin \mathbf{X} değişken matrisini ne kadar açıkladığını belirlemekte kullanılır. Bir başka ifade ile \mathbf{X}_i , $i = 1, 2, \dots, k$ değişkeninin ilgili bileşen üzerindeki yükünü göstermektedir.
 10. $\mathbf{Y}_i' \mathbf{u}_i / (\mathbf{u}_i' \mathbf{u}_i)$ ile \mathbf{Y} değişken matrisinin ilgili bileşeni üzerine regresyonundan \mathbf{q}_i yük vektörü elde edilir. Bu değerler ilgili temel bileşenin \mathbf{Y} değişken matrisini ne kadar açıkladığını belirlemekte kullanılır. Bu değer, \mathbf{Y}_i değişkeninin ilgili bileşen üzerindeki yükünü göstermektedir.
 11. Bileşenler arasında içsel bir ilişki tanımlayan b katsayısı, $b = \mathbf{u}_i' \mathbf{t}_i / (\mathbf{t}_i' \mathbf{t}_i)$ ile \mathbf{u} 'nun \mathbf{t} üzerine regresyonundan tahmin edilir. Bu sabit bir değerdir ve tahmin etmede önem taşımaktadır.
 12. Son adımda ise $(l+1)$. yinelemede kullanılacak olan artık vektörler açıklayıcı değişkenler için $\mathbf{X}_{(l+1)}$ ve bağımlı değişkenler için $\mathbf{Y}_{(l+1)}$ sırasıyla $\mathbf{X}_{l+1} \rightarrow \mathbf{X}_l - \mathbf{t}_l \mathbf{p}_l'$ ve $\mathbf{Y}_{l+1} \rightarrow \mathbf{Y}_l - b \mathbf{t}_l \mathbf{c}_l'$ eşitlikleri ile elde edilir. Bu matrisler l . yinelemede kullanılan matrislerin açıklayamadığı değişimlerden oluşan matrislerdir.
- Gösterim ile ifade edilmek istenirse;



Aynı işlem Y değişken matrisi içinde tanımlanmaktadır.

Algoritmaya X değişken matrisi sıfır matrisi oluncaya kadar başka bir deyişle açıklayıcı değişkenlerinin büyük bir kısmı açıklanuncaya kadar devam edilir. Algoritmanın yineleme sayısı ihtiyaç duyulan bileşen sayısını vermektedir.

2.2 SIMPLS Algoritması

De Jong (1993)'un geliştirdiği SIMPLS algoritmanın temeli NIPALS algoritmasına dayanmaktadır. İsmi İngilizce olarak “**S**traightforward **I**mplementation of a **S**tatistically Inspired Modification of PLS Method” dan alan bu algoritma, yaygın olarak kullanılmakta olup, KEKK yönteminin istatistiksel bir uyarlamasıdır.

SIMPLS algoritması ile NIPALS algoritması arasında benzerlikler ve farklılıklar bulunmaktadır. Lindgren ve Rännar (1998)'in ifade ettiği gibi SIMPLS, NIPALS algoritmasına kıyasla daha hızlı, daha az zaman gerektiren ve hafızada daha az yer kaplayan bir algoritmadır. Diğer bir benzerlik ve fark ise De Jong (1993)'un makalesinde de belirttiği gibi bağımlı değişken sayısının bir olduğu durumda algoritmaların aynı sonucu vermesi, bağımlı değişken sayısının birden fazla olduğu durumda ise sonuçlar arasında çok fazla olmayan bir farklılığın mevcut olmasıdır.

NIPALS algoritmasında ağırlık vektörleri hesaplanırken her seferinde X ve Y değişkenlerinin azaltılmış veri matrisleri kullanılmaktadır. Bu da faktörlerin yorumlanmasını zorlaştırmaktadır. Çünkü yineleme sayısı ilerledikçe alt yinelemelerde kullanılan azaltılmış veri matrisindeki katkının ne olduğundan şüphe duyulabilmektedir. Bazı X değişkenleri ilk faktörde yer alırken, diğer değişkenler sonraki faktörlerde kullanılmaktadır. Bu nedenle bileşen ve değişkenler arasındaki ilişki, yükler (p) aracılığıyla daha iyi anlatılabilmektedir. SIMPLS algoritmasında ise, bileşenler sıfır ortalamaya sahip X orijinal veri matrisi üzerinden tanımlanmaktadır.

$$t_l = Xr_l \quad (1)$$

$$u_l = Yc_l \quad l = 1, \dots, A \quad (2)$$

Yukarıdaki eşitliklerde gösterilen değerler, her bir yinelemede elde edilen vektörlerdir. Algoritmanın sonunda bu vektörler matris olarak aşağıdaki şekilde ifade edilmektedir.

$$T = XR \quad (3)$$

$$U = YC \quad (4)$$

SIMPLS algoritması da değişkenler arasındaki kovaryansı en çoklama temeline dayanmaktadır. Bu algortmada kullanılan r ve c ağırlıkları, t ve u skor vektörleri arasındaki kovaryansı en çok yapacak şekilde aşağıdaki kısıtlar altında elde edilir.

1. $r'_l r_l = 1$ ağırlıkların normalize edilmesi.
2. $c'_l c_l = 1$ ağırlıkların normalize edilmesi.
3. t skorlarının dikliği. Bir başka ifade ile, $t'_{l+1} t_l = 0$ koşulunun sağlanması.
4. $u'_l t_l$ kovaryansının en çoklanması.

SIMPLS algoritmasında $S = X'Y$ kovaryans matrisine tekil değer ayrıştırması yöntemi uygulanarak, l . adım için ağırlık vektörleri r_l ve c_l elde edilir. r_l bu ayrıştırmada en yüksek tekil değere karşılık gelen sol tekil vektör iken, c_l sağ tekil vektördür. Dik t bileşen vektörleri elde etmek için ilk adımdan sonra ($l > 1$ için) $S_l = S_{l-1} - v_l(v'_l S_{l-1})$ matrisi hesaplanarak bu matris üzerinden algortmaya devam edilir. Ortonormal v_l vektörü ilk adımda $v_l \propto p_1$ şeklinde hesaplanırken, $l = 2, 3, \dots, A$ için $v_l \propto p_l - V_{l-1}(V'_{l-1} p_l)$ şeklinde hesaplanmaktadır. Burada V_{l-1} matrisi, $(l+1)$. adıma kadar elde edilen vektörlerden oluşmaktadır. Bir başka ifade ile, $V_{l-1} = [v_1, v_2, \dots, v_{l-1}]$ 'dir. \propto simgesi oransallığın yanı sıra ilgili vektörün uzunluğu bir olacak şekilde normalleştirilmesini de ifade etmektedir. Bu algortmada NIPALS algortmasına karşı önerilen değişiklik R ağırlıklarının doğrudan hesaplanmasıdır. Bu yolla X ve Y matrisleri her bir yinelemede indirgenmemiş olmaktadır.

SIMPLS algortması aşağıdaki adımlardan oluşmaktadır. Algortmada l indisi NIPALS' ta olduğu gibi yineleme sayısını göstermekte olup $l = 1, 2, 3, \dots, A$ 'dır.

1. Orijinal veri matrisleri üzerinden $S = X'Y$ kovaryans matrisi tanımlanır ve bu matrise tekil değer ayrıştırması uygulanır. En yüksek tekil değer elde edilir.
2. En yüksek tekil değere karşılık gelen sol tekil vektör r_l , bir başka ifade ile X değişkeninin ilgili bileşeni (t) için ağırlık vektörü hesaplanır.
3. X değişkeninin ilgili bileşen değeri $t_l = Xr_l$ ile orijinal X matrisi ile bu bileşen için ağırlıkların çarpımından elde edilir.
4. t bileşeni normuna bölünerek boyu bir olacak şekilde ölçeklendirilir. Yeni t_l bileşeni $t_l = t_l / \|t_l\|$ ile elde edilir ve bu eşitlikte $\|t_l\|$, $\sqrt{t'_l t_l}$ ile hesaplanmaktadır.
5. t bileşeninin boyu bir olacak şekilde ölçeklendirildiği için $r_l = r_l / \|t_l\|$ ile ağırlık vektörü r , t 'nin normuna bölünerek, ölçeklendirilen t bileşen vektörüne göre uyarlanır.

6. X değişkeninin ilgili bileşeni için, X değişkeninin bileşen üzerine yükü p_l , $p_l = X't_l$ eşitliği ile hesaplanır.
7. Adım 6'ya benzer olarak Y değişkeninin ilgili bileşeni için ağırlık değeri c_l , $c_l = Y't_l$ eşitliği ile elde edilir.
8. Bir önceki adımda elde edilen c_l değeri $u_l = Yc_l$ eşitliğinde yerine konularak, Y değişkeninin ilgili bileşen değeri hesaplanır.

9. $l = 1$ için $v_1 = p_1$ alınarak dik yükler algoritmaya dahil edilir.

$l > 1$ için ise

$v_l = v_l - V(V'p_l)$ eşitliği ile önceki yüklere dik olacak şekilde v vektörü elde edilir.

10. $u_l = u_l - T(T'u_l)$ eşitliği ile de önceki t değerlerine dik olacak şekilde u vektörü elde edilir.

11. $v_l = v_l / \|v_l\|$ ile elde edilen yeni v yük vektörü normuna bölünerek boyu 1 olacak şekilde ölçeklendirilir.

12. Son adımda ise önceki adımlardan elde edilen değerler $S_{l+1} = S_l - v_l(v_l'S_l)$ eşitliğinde kullanılarak indirgenmiş kovaryans matrisi elde edilir ve bu değer adım 1'de yerine konularak algoritmaya devam edilir.

3. UYGULAMA

Uygulamada kullanılan veriler 'Ondokuz Mayıs Üniversitesi Tıp Fakültesi'nden alınmıştır. Değişkenler 75 hasta için ölçülmüş olup, aşağıda Tablo 1'de verilmiştir. Uygulamada iki bağımlı değişken, Y matrisi, 15 açıklayıcı değişken, X matrisi, olarak alınmıştır.

Tablo 1. Değişkenler tablosu

Bağımlı Değişkenler	Bağımsız Değişkenler
Yüksek kan basıncı (siskb)	Yaş
Düşük kan basıncı (diaskb)	Boy
	Kilo
	Bme (vücut-kütle endeksi)
	Bç (bel çevresi)
	Kç (kalça çevresi)
	Bko (bel kalça oranı)
	Lpa (lipoprotein a)
	Trigliserid (kandaki yağ miktarı, mg/dl)
	Totkol (toplam kolesterol, mg/dl)
	Ldl (kötü huylu kolesterol, mg/dl)
	Hdl (iyi huylu kolesterol, mg/dl)
	Glikoz (mg/dl)
	İnsülin (ulU/ml)
	Homo (mg/dl)

Fazla sayıda açıklayıcı değişken olması değişkenler arasında çoklu doğrusal bağlantı probleminin ortaya çıkmasını kolaylaştırmaktadır. Açıklayıcı değişkenler arasında çoklu doğrusal bağlantı problemini saptamak için VIF (Variance Inflation Factor) varyans şişirme faktör değerlerine bakılmıştır. VIF değerinin 10'dan büyük olması, değişkenler arasında çoklu doğrusal bağlantı probleminin olduğunu göstermektedir. VIF hakkında daha ayrıntılı bilgi için bakınız "Rawlings (1988)". Çalışmada bazı açıklayıcı değişkenler için VIF değerleri boy için 13.69, bç için 500 ve insülin için 55.56 olarak hesaplanmıştır. Bu değerler, açıklayıcı değişkenler arasında çoklu doğrusal bağlantı olduğunun bir göstergesidir.

Çalışmanın amacı, açıklayıcı değişkenler arasındaki çoklu doğrusal bağlantı problemini ortadan kaldırmak ve hem açıklayıcı değişkenlerdeki değişimi, hem de bağımlı değişkendeki değişimi açıklayacak az sayıda bileşenleri elde etmek için kullanılan KEKK algoritmalarından NIPALS ve SIMPLS algoritmalarını uygulamak, analizler sonucunda elde edilen değerlerin benzerliklerini ve farklılıklarını ortaya koymaktır. Algoritmalar MATLAB programında yazılarak sonuçlar hesaplatılmıştır. Algoritmalara ait MATLAB kodları ekte yer almaktadır.

3.1 NIPALS Algoritması

Algoritma sonucunda elde edilen değerler Tablo 2-Tablo 9'da verilmektedir.

Tablo 2. X değişken matrisi için bileşen matrisi (T)

Hasta	t_1	t_2	t_3	...	t_{13}	t_{14}	t_{15}
1	0.0386	-0.1045	-0.0224	...	0.1134	-0.0191	0.0015
2	0.0197	-0.0285	-0.1182	...	0.0503	-0.0054	-0.1169
3	0.2036	0.1603	-0.0805	...	0.1637	-0.0553	-0.1350
⋮	⋮	⋮	⋮	...	⋮	⋮	⋮
73	-0.1241	0.2232	-0.0234	...	0.1569	-0.1055	0.0921
74	-0.1457	-0.0789	0.0291	...	0.0938	-0.1787	-0.2559
75	-0.1443	-0.0909	-0.0657	...	0.0474	-0.0496	0.0240

Tablo 3. Y değişken matrisi için bileşen matrisi (U)

Hasta	u_1	u_2	u_3	...	u_{13}	u_{14}	u_{15}
1	-0.0739	-0.4086	-0.0898	...	-0.1447	-0.2206	-0.2943
2	0.9376	0.6809	0.6188	...	0.9141	-0.6283	0.5569
3	4.2439	2.6269	1.8934	...	2.1983	-0.0612	1.9070
⋮	⋮	⋮	⋮	...	⋮	⋮	⋮
73	0.7980	1.7232	0.8776	...	0.9779	-0.0521	0.8027
74	-2.1875	-1.0830	-0.7753	...	-0.4710	-0.1517	-0.5122
75	-0.6249	0.4090	0.6395	...	1.0173	-0.2066	0.8360

Tablo 2 ve Tablo 3’de verilen t_a ve u_a , $a = 1,2,\dots,15$, vektörleri yeni bileşen matrislerini oluşturmaktadır. Tablo 2 ile verilen t_a bileşenleri X değişken matrisini modellemede ve Y ’yi tahmin etmede kullanılmaktadır.

Tablo 4. T için ağırlık matrisi (W)

	w_1	w_2	w_3	...	w_{13}	w_{14}	w_{15}
Yaş	0.0850	0.1582	-0.2183	...	0.0014	-0.0166	0.0022
Boy	-0.0488	0.1289	0.0626	...	0.1633	-0.0571	-0.0551
Kilo	0.3900	0.0152	0.0070	...	-0.4550	0.1309	0.2035
⋮	⋮	⋮	⋮	...	⋮	⋮	⋮
Glikoz	0.2780	0.2156	0.1067	...	-0.0329	0.0452	0.0060
İnsulin	0.2814	-0.1444	0.3769	...	-0.3092	0.0259	0.0160
Homo	0.2831	-0.1806	0.1830	...	0.3283	-0.0532	-0.0209

Tablo 5. U için ağırlık matrisi (C)

	c_1	c_2	c_3	...	c_{13}	c_{14}	c_{15}
Siskb	0.7264	0.8086	0.8980	...	0.5554	0.8209	0.8439
Diaskb	0.6873	0.5884	0.4401	...	0.8316	-0.5711	0.5365

Tablo 4 ve Tablo 5 ile verilen ağırlık matrisleri $T = XW$ ve $U = YC$ eşitlikleri ile bileşenleri oluşturmakta kullanılmaktadır. w_a ve c_a ağırlık vektörleri t_a ve u_a bileşenlerinin yorumlanmasını sağlamaktadır. w_{ia} ve c_{ja} katsayıları, $i = 1,2,\dots,15$ ve $j = 1,2$ için değişkenlerin bileşenlerini oluşturmak için nasıl bir araya geldiklerini göstermektedir. t_a bileşenine katkısı yüksek olan X_i değişkenine ait w_{ia} değeri yüksek olacaktır. t_a bileşeninin Y_i ’yi modellemedeki önemi ise c_{ja} ile ölçülmektedir. Eğer t_a bileşeni Y_i ’yi modellemede önemli ise c_{ja} değeri yüksek olacaktır. Örneğin; Tablo 4’e bakıldığında kilo değişkeninin 13. bileşene katkısı -0.4550 katsayısı ile daha yüksek iken, insülin ve homo değişkenlerinin ilgili bileşene katkılarının eşit olduğu söylenebilir. Ayrıca yaş değişkeninin de ilgili bileşene katkısı oldukça düşüktür. Tablo 5’e bakıldığında ise, 3. bileşenin siskb değişkenini modellemedeki katkısının diaskb değişkenini modellemedeki katkısına göre daha fazla olduğu görülmektedir.

Tablo 6. X değişken matrisini açıklayan bileşenler için yük matrisi (P)

	p_1	p_2	p_3	...	p_{13}	p_{14}	p_{15}
Yaş	1.2407	1.6976	-4.1808	...	0.0004	-0.0059	0.0007
Boy	-1.3673	0.8199	3.0663	...	0.0947	-0.0168	-0.0167
Kilo	7.9740	0.1404	0.5088	...	-0.2703	0.0342	0.0617
⋮	⋮	⋮	⋮	...	⋮	⋮	⋮
Glikoz	4.9478	0.9437	0.1482	...	-0.0182	0.0154	0.0018
İnsulin	6.2710	-1.9534	3.3867	...	-0.1687	0.0081	0.0048
Homo	6.4084	-1.6746	3.2505	...	0.1795	-0.0174	-0.0063

Yük matris değerleri, bileşenlerin açıklayıcı değişkendeki değişimi açıklamaya olan katkılarından oluşmaktadır. p_a değerleri açıklayıcı değişkende var olan değişimin en iyi özetleyicisidirler. X_i 'yi modellemede önemli olan t_a vektörüne ait p_{ia} değeri yüksek olacaktır.

Elde edilen bileşenlerin tek tek ve birikimli olarak değişkenlerdeki değişimi açıklama yüzdeleri merak edilebilir. Açıklayıcı değişkendeki değişimi açıklama yüzdelerini bulmak için $P'P$ matrisinin köşegen değerleri SS_X 'e bölünmektedir. $P'P$ matrisin köşegen değerleri aşağıda verilmektedir.

Tablo 7. $P'P$ matrisinin köşegen değerleri

430.9972	54.0879	143.4910	63.4381	77.1674	57.0910	49.9939
58.4382	43.3200	57.3475	64.4502	9.6541	0.3070	0.1247
						0.0922

Açıklayıcı değişkenler için kareler toplamını gösteren SS_X , $X'X$ matrisinin iz değerine eşittir ve yapılan hesaplama sonucunda bu değer 1110 olarak elde edilmiştir. Bileşenlerin açıklayıcı değişkenlerdeki değişimi açıklama yüzdeleri aşağıda Tablo 8'de verilmektedir.

Tablo 8. Bileşenlerin açıklayıcı değişkendeki değişimi açıklama yüzdeleri

Bileşen	X 'deki değişimi açıklama yüzdesi	Açıklanan değişimin birikimli yüzdesi
1	38.83	38.83
2	4.87	43.70
3	12.93	56.63
4	5.71	62.34
5	6.95	69.29
6	5.15	74.44
7	4.50	78.94
8	5.27	84.21
9	3.90	88.11
10	5.17	93.28
11	5.80	99.08
12	0.87	99.95
13	0.03	99.98
14	0.01	99.99
15	0.01	100.0

Tablodan da görüleceği gibi 15 açıklayıcı değişkenden, NIPALS algoritması sonucunda elde edilen 11 bileşen X 'deki değişimin %99'unu açıklamaktadır. 11. bileşen ise tek başına açıklayıcı değişkendeki değişimin %5.80'ini açıklamaktadır. Bu bileşenden sonraki bileşenlerin değişimi açıklama yüzdeleri yok denilebilecek kadar azdır.

Aynı tablo bağımlı değişken için de verilebilir. Bağımlı değişkendeki değişimi açıklama yüzdeleri $C'C$ matrisinin köşegen değerlerinin $SS_Y = iz(Y'Y) = 148$ 'e bölünmesi ile elde edilir. SS_Y bağımlı değişkenler için kareler toplamını göstermektedir. Hesaplamalar sonucunda elde edilen açıklama yüzdeleri Tablo 9'da verilmektedir.

Tablo 9. Bileşenlerin bağımlı değişkendeki değişimi açıklama yüzdeleri

Bileşen	Y'deki değişimi açıklama yüzdesi	Açıklanan değişimin birikimli yüzdesi
1	38.67	38.67
2	8.55	47.22
3	0.59	47.81
4	0.58	48.39
5	0.28	48.67
6	0.22	48.89
7	0.39	49.28
8	0.72	50.00
9	0.79	50.79
10	0.13	50.92
11	0.18	51.10
12	0.99	52.09
13	0.37	52.46
14	0.31	52.77
15	0.08	52.85

Tablo 9'dan da görüldüğü gibi 15 değişkenden elde edilen bileşenlerin 11 tanesi Y 'deki değişimin %51'ini açıklamaktadır. Bu değişim 15 bileşen için yaklaşık olarak %53 olarak elde edilmiştir.

3.2 SIMPLS Algoritması

SIMPLS algoritması sonucunda elde edilen değerler, Tablo 10-Tablo 15'de verilmektedir.

Tablo 10. X değişken matrisi için bileşen matrisi (T)

Hasta	t_1	t_2	t_3	...	t_{13}	t_{14}	t_{15}
1	0.0386	-0.1046	-0.0256	...	0.1133	0.0191	-0.0017
2	0.0198	-0.0286	-0.1180	...	0.0503	0.0064	0.1168
3	0.2036	0.1603	-0.0795	...	0.1637	0.0565	0.1345
⋮	⋮	⋮	⋮	...	⋮	⋮	⋮
73	-0.1241	0.2233	-0.0194	...	0.1563	0.1055	-0.0929
74	-0.1457	-0.0789	0.0298	...	0.0933	0.1812	0.2547
75	-0.1443	-0.0909	-0.0639	...	0.0474	0.0499	-0.0244

Tablo 11. Y değişken matrisi için bileşen matrisi (U)

Hasta	u_1	u_2	u_3	...	u_{13}	u_{14}	u_{15}
1	-0.5590	-1.4540	-0.0909	...	-0.1069	0.1534	0.0959
2	7.0931	2.4214	0.5493	...	0.6682	0.4210	-0.1863
3	32.1060	9.3446	1.7319	...	1.6106	0.0204	-0.6280
⋮	⋮	⋮	⋮	...	⋮	⋮	⋮
73	6.0368	6.1301	0.8059	...	0.7156	0.0270	-0.2642
74	-16.5490	-3.8528	-0.7154	...	-0.3473	0.1100	0.1683
75	-4.7277	1.4544	0.5791	...	0.7448	0.1314	-0.2761

Bu algoritma, Bölüm 2.1'de verilen NIPALS algoritmasından farklı olarak \boldsymbol{w} ağırlık vektörü yerine \boldsymbol{r} ile gösterilen dikleştirilmiş ağırlık vektörü ile çalışmaktadır. Her adımda \boldsymbol{w} ağırlık vektörü azaltılmış \boldsymbol{X} matrisi üzerinden hesaplanırken, \boldsymbol{r} vektörü orijinal \boldsymbol{X} matrisi üzerinden hesaplanmaktadır. NIPALS algoritmasında da \boldsymbol{w} ağırlık vektörü yerine \boldsymbol{r} ağırlık vektörü hesaplanabilmektedir (Bakınız Wold vd., 2001).

Tablo 12. \boldsymbol{T} için ağırlık matrisi (\boldsymbol{R})

	\boldsymbol{r}_1	\boldsymbol{r}_2	\boldsymbol{r}_3	...	\boldsymbol{r}_{13}	\boldsymbol{r}_{14}	\boldsymbol{r}_{15}
Yaş	0.0850	0.1693	-0.1349	...	-0.0023	-0.0081	-0.0029
Boy	-0.0488	0.1202	0.1165	...	0.1821	0.0551	0.0180
Kilo	0.3900	0.0747	0.0406	...	-0.5937	-0.1471	-0.0768
⋮	⋮	⋮	⋮	...	⋮	⋮	⋮
Glikoz	0.2781	0.2549	0.1702	...	-0.0030	-0.0187	-0.0076
İnsulin	0.2815	-0.0994	0.3190	...	-0.1110	-0.0674	0.0009
Homo	0.2831	-0.1351	0.1244	...	0.1225	0.0704	0.0015

\boldsymbol{X} matrisi için ağırlık ve yük \boldsymbol{R} ve \boldsymbol{P} , Tablo 12 ve Tablo 13'te verilmektedir. \boldsymbol{r}_i , $i = 1, 2, \dots, 15$, ağırlık değerleri orijinal \boldsymbol{X} değişkenlerinin \boldsymbol{t}_i bileşenleri üzerine ağırlığını belirtmektedir. \boldsymbol{r}_a vektörleri orijinal \boldsymbol{X} matrisi üzerinden hesaplandığı için, \boldsymbol{w}_{ia} değerleri yerine \boldsymbol{r}_{ia} değerlerinin kullanılması ilgili değişkenin \boldsymbol{t}_a bileşeni üzerine etkisini yorumlamada kolaylık sağlamaktadır.

Tablo 13. \boldsymbol{X} değişken matrisini açıklayan bileşenler için yük matrisi (\boldsymbol{P})

	\boldsymbol{p}_1	\boldsymbol{p}_2	\boldsymbol{p}_3	...	\boldsymbol{p}_{13}	\boldsymbol{p}_{14}	\boldsymbol{p}_{15}
Yaş	1.2407	1.6940	-4.2186	...	0.0008	0.0170	-0.0007
Boy	-1.3673	0.8250	3.1576	...	0.0934	0.0117	0.0166
Kilo	7.9740	0.1419	0.5407	...	-0.2696	-0.0389	-0.0613
⋮	⋮	⋮	⋮	...	⋮	⋮	⋮
Glikoz	4.9478	0.9393	0.0282	...	-0.0213	-0.0194	-0.0017
İnsulin	6.2710	-1.9504	3.4267	...	-0.1687	-0.0008	-0.0047
Homo	6.4084	-1.6726	3.2649	...	0.1792	0.0308	0.0061

NIPALS algoritmasındaki benzer olarak SIMPLS algoritması sonucunda da bulunan bileşenlerin bağımlı ve açıklayıcı değişkenlerdeki değişimi açıklama yüzdeleri hesaplanabilir. Bu değerler, NIPALS algoritmasında olduğu gibi hesaplanmaktadır.

Tablo 14. Bileşenlerin açıklayıcı değişkendeki değişimi açıklama yüzdeleri

Bileşen	X'deki değişimi açıklama yüzdesi	Açıklanan değişimin birikimli yüzdesi
1	38.83	38.83
2	4.86	43.69
3	12.94	56.63
4	5.74	62.37
5	6.94	69.31
6	5.73	75.04
7	4.93	79.97
8	4.40	84.37
9	2.55	86.92
10	4.60	91.52
11	8.00	99.52
12	0.43	99.95
13	0.03	99.98
14	0.01	99.99
15	0.01	100.0

Tablo 15. Bileşenlerin bağımlı değişkendeki değişimi açıklama yüzdeleri

Bileşen	Y'deki değişimi açıklama yüzdesi	Açıklanan değişimin birikimli yüzdesi
1	38.67	38.67
2	8.55	47.22
3	0.59	47.81
4	0.58	48.39
5	0.28	48.67
6	0.08	48.75
7	0.24	48.99
8	0.47	49.46
9	1.10	50.56
10	0.36	50.92
11	0.08	51.00
12	1.09	52.09
13	0.37	52.46
14	0.31	52.77
15	0.08	52.85

Tablo 10 sonuçlarına bakıldığında, daha öncede bahsedildiği gibi t_1 bileşen değerlerinin NIPALS algoritması ile aynı ve t_2, \dots, t_{15} değerlerinden ise çok da büyük olmayan farklılıklar olduğu görülmektedir. Aynı yorum Tablo 12 ve Tablo 13 ile verilen ağırlık ve yük değerleri içinde geçerlidir. Bileşenlerin değişimi açıklama yüzdeleri incelendiğinde ise sonuçların genel olarak NIPALS algoritması ile yakın hatta aynı olduğu söylenebilir.

4. TARTIŞMA VE SONUÇ

KEKK, açıklayıcı değişkenin gözlem sayısından çok olduğu durumda veya açıklayıcı değişkenler arasında çoklu doğrusal bağlantı olduğu durumda kullanılabilen bir yöntemdir. Burada amaç açıklayıcı değişkeni açıklayan az sayıda bileşeni seçerken bağımlı değişkeni de en iyi şekilde tahmin etmektir. Uygulamada kullanılan veri kümesinde açıklayıcı değişkenler arasında çoklu doğrusal bağlantı olduğundan bu yöntem tercih edilmiştir.

Bu çalışmada kısaca KEKK yönteminde bileşenleri elde etmek için kullanılan NIPALS ve SIMPLS algoritmaları arasındaki benzerlik ve farklılıklar anlatılarak, bir örnek üzerinde uygulaması yapılmıştır. Her iki algoritma da $X'Y$ kovaryans matrisini en çoklamaya dayanmaktadır. Fakat NIPALS algoritması bileşenleri X ve Y veri matrislerini indirgeyerek elde ederken, SIMPLS algoritması $S = X'Y$ kovaryans matrisini indirgeme temeline dayanmaktadır. Uygulama sonuçlarına bakıldığında her iki algoritma sonucunda elde edilen ilk 11 bileşen açıklayıcı ve bağımlı değişkenlerdeki değişiminin sırası ile %99 ve %51'ini açıklamaktadır. Bu iki algortmada aralarında çoklu doğrusal bağlantı olan 15 açıklayıcı değişken, açıklayıcı değişkenlerdeki değişimin neredeyse tamamını ve bağımlı değişkendeki değişimin yarısını açıklayacak şekilde aralarında çoklu doğrusal bağlantı problemi olmayan 11 bileşene indirgenmektedir. Tablolar incelendiğinde, her iki algoritma sonucunda ilk bileşen için elde edilen değerlerin aynı olduğu, diğer bileşenler ($a > 2$) için ise az da olsa bir farklılığın mevcut olduğu görülmektedir. SIMPLS algoritmasında c vektörünün boyu 1 olacak şekilde ölçeklendirilmediğinden bağımlı değişken için elde edilen bileşen değerleri NIPALS'dan farklı çıkmaktadır.

5. KAYNAKLAR

De Jong, S., 1993. SIMPLS: An alternative approach to partial least squares regression. *Chemometrics and Intelligent Laboratory Systems* 18, 251-263.

Geladi, P., Kowalski, R., 1986. Partial least squares regression: A tutorial. *Analytica Chimica Acta* 185, 1-17.

Helland, I. S., 1988. On the structure of partial least squares regression. *Communications in Statistics* 17(2), 581-607.

Höskuldsson, A., 1988. PLS regression methods. *Journal of Chemometrics* 2, 211- 228.

Lindgren, F., Rännar, S., 1998. Alternative partial least-squares (PLS) algorithms. *Perspectives in Drug Discovery and Design* 12/13/14, 105-113.

Randall, D. Tobias., 1996. An introduction to partial least squares regression. SAS Institute Inc., Cary, NC. <http://support.sas.com/techsup/technote/ts509.pdf>, 11 Şubat 2008.

Rawlings, J. O., 1988. Applied regression analysis: A research tool. Cole Advanced Books&Software. California.

Wold, S., Sjostrom, M., Eriksson, L., 2001. PLS-regression: A basic tool of chemometrics. *Chemometrics and Intelligent Laboratory Systems* 58, 109-130.

INTRODUCTION TO PARTIAL LEAST SQUARES ALGORITHMS NIPALS AND SIMPLS AND AN APPLICATION

ABSTRACT

Partial least squares (PLS) is used for obtaining the smallest number of orthogonal latent variables (component) that explain the great amount of variation in the original independent variables which have multicollinearity. Since it is applicable to various disciplines different algorithms have been developed for PLS. In this study, NIPALS which is the first and basic algorithm of PLS and SIMPLS developed by De Jong in 1993 will be described and illustrated on an health data set.

Key words: *Multicollinearity, Partial Least Squares, NIPALS, SIMPLS.*

6. EKLER

Ek 1. NIPALS Algoritması için MATLAB Kodu

[X]=x;	Açıklayıcı değişkenler matrisi tanımlanır.
[Y]=y;	Bağımlı değişkenler matrisi tanımlanır.
hata2=1;	Dış döngü ve birikimli varyans açıklama başlangıç değerleri tanımlanır.
tekrar=0;	
cumvarx=0;	
cumvary=0;	
ictekrar=[];	Açıklayıcı değişkenler için $X'X$ varyans-kovaryans matrisi oluşturulur.
SSX=X'*X;	
SSY=Y'*Y;	Bağımlı değişkenler için $Y'Y$ varyans-kovaryans matrisi oluşturulur.
while hata2>10^-6	Y matrisinin ilk sütunu u vektörü olarak alınır. Dış döngüdeki yinelemeyi artırır.
var(x)=0	
u=Y(:,1);	İç döngü için başlangıç değeri (u vektörlerinin yakınsaması ile ilgili olan döngü)
tekrar=tekrar+1;	
hata=1;	İç döngüdeki yinelemeyi artırır.
tekrar2=0;	
while hata>10^-6	X matrisinin ilgili bileşeni için ağırlık vektörü oluşturulur.
tekrar2=tekrar2+1;	
w=X'*u/(u'*u);	Ağırlık vektörü boyu bir olacak şekilde ölçeklendirilir.
w=w/norm(w);	
t=X*w;	X matrisi için bileşen vektörü elde edilir.
t=t/norm(t);	
c=Y'*t/(t'*t);	Bileşen vektörü boyu bir olacak şekilde ölçeklendirilir.
c=c/norm(c);	
u_yeni=Y*c/(c*c);	Y matrisinin ilgili bileşeni için ağırlık vektörü oluşturulur.
d=u-u_yeni;	
hata=norm(d);	Ağırlık vektörü boyu bir olacak şekilde ölçeklendirilir.
u=u_yeni;	
end;	Y matrisi için bileşen vektörü elde edilir.
ictekrar=[ictekrar tekrar2];	
p=X'*t/(t'*t);	İlk adımdaki u vektörü ile elde edilen u vektörü arasındaki fark hesaplanır.
q=Y'*u/(u'*u);	
b=u'*t/(t'*t);	Fark vektörünün boyu hesaplanır.
X_yeni=X-(t*p);	
Y_yeni=Y - b*t*c';	Elde edilen yeni u vektörü u vektörü olarak alınır.
hata2=sum(sum(abs(X_yeni)));	
X=X_yeni;	X'in ilgili bileşeni için yük vektörü hesaplanır.
Y=Y_yeni;	
varx=(p'*p)/trace(SSX);	Y'nin ilgili bileşeni için yük vektörü hesaplanır.
	Bileşenler arası içsel ilişkiyi tanımlayan b katsayısı hesaplanır.
	X matrisi için artık değer matrisi hesaplanır.
	Y matrisi için artık değer matrisi hesaplanır.
	Yeni artık değer matrisi X matrisi olarak alınır.
	Yeni artık değer matrisi Y matrisi olarak alınır.
	Her bileşenin açıklayıcı değişkenlerin varyansını açıklama yüzdeleri elde edilir.

```

vary=(b*b)/trace(SSY)
u_lar(:,tekrar)=u;
t_lar(:,tekrar)=t;
b_lar(:,tekrar)=b;

c_lar(:,tekrar)=c;
w_lar(:,tekrar)=w;
q_lar(:,tekrar)=q;
varxpercler(:,tekrar)=varx;

varypercler(:,tekrar)=vary;

cumvarperxler(:,tekrar)=cumvarx;

cumvarperyler(:,tekrar)=cumvary;

end;
u_lar
t_lar
b_lar
p_lar
c_lar
w_lar
q_lar
varxpercler
varypercler
cumvarperxler
cumvarperyler
tekrar
ictekrar

```

Her bileşenin bağımlı değişkenlerin varyansını açıklama yüzdeleri elde edilir.

Elde edilen vektörler ilgili matrislere atanır. $p_lar(:,tekrar)=p;$

Elde edilen vektörler ilgili matrislere atanır.

Her bir bileşen için açıklayıcı değişkenlerin varyansı açıklama yüzdeleri matrise atanır.

Her bir bileşen için bağımlı değişkenlerin varyansı açıklama yüzdeleri matrise atanır.

Tüm bileşenlerin açıklayıcı değişkenlerin varyansı açıklamada birikimli açıklama yüzdeleri matrise atanır.

Tüm bileşenlerin bağımlı değişkenlerin varyansı açıklamada birikimli açıklama yüzdeleri matrise atanır.

Matrisler yazdırılır.

Açıklama yüzdeleri yazdırılır.

Algoritma yineleme sayısını verir.
u vektörlerinin birbirine yakınsaması ile ilgili olan iç döngüdeki toplam yineleme sayısını verir.

Ek 2. SIMPLS Algoritması için MATLAB Kodu

[X]=x;	Açıklayıcı değişkenler matrisi tanımlanır.
[Y]=y;	Bağımlı değişkenler matrisi tanımlanır.
S=X'*Y;	X'Y varyans-kovaryans matrisi oluşturulur.
SSX=X'*X;	Açıklayıcı değişkenler için X'X varyans-kovaryans matrisi oluşturulur.
SSY=Y'*Y;	Bağımlı değişkenler için Y'Y varyans-kovaryans matrisi oluşturulur.
cumvarx=0;	} Birikimli varyans açıklama başlangıç değeri tanımlanır.
cumvary=0;	
for a=1:15	
[r,s,c]=svds(S,1);	X'Y matrisinin özdeğer ve özvektörü hesaplatılır.
t=X*r;	X matrisi için bileşen vektörü elde edilir.
t=t-mean(t);	Bileşen vektör değerleri ortalamasından çıkarılarak merkezileştirilir.
t=t/norm(t);	Bileşen vektörü normuna bölünerek ölçeklendirilir.
r=r/norm(t);	Ağırlık vektörü bileşen vektörünün normuna bölünerek ölçeklendirilir.
p=X*t;	Bileşen için yük vektörü elde edilir.
c=Y*t;	Y matrisinin ilgili bileşeni için ağırlık vektörü oluşturulur.
u=Y*c;	Y matrisi için bileşen vektörü elde edilir.
v=p;	
varx=(p'*p)/trace(SSX);	Bileşenlerin açıklayıcı değişkenlerin varyansını açıklama yüzdeleri elde edilir.
vary=(c'*c)/trace(SSY);	Bileşenlerin bağımlı değişkenlerin varyansını açıklama yüzdeleri elde edilir.
if a>1	
v=v-V*(V'*p);	} v vektörü önceki yük vektörüne ve u vektörü önceki t vektörlerine dik olacak şekilde tanımlanır.
u=u-T*(T'*u);	
end;	
v=v/norm(v);	v vektörü normuna bölünerek ölçeklendirilir.
S=S-v*(v'*S);	S matrisi indirgenir.
R(:,a)=r; T(:,a)=t;	} Elde edilen vektörler ilgili matrislere atanır.
P(:,a)=p; C(:,a)=c;	
U(:,a)=u; V(:,a)=v;	
varxpercler(:,a)=varx;	Her bir bileşen için açıklayıcı değişkenlerdeki varyansı açıklama yüzdeleri matrise atanır.
varypercler(:,a)=vary;	Her bir bileşen için bağımlı değişkenlerdeki varyansı açıklama yüzdeleri matrise atanır.
cumvarperxler(:,a)=cumvarx;	} Birikimli varyans açıklama yüzdeleri matrise atanır.
cumvarperyler(:,a)=cumvary;	
end;	
varxpercler;	} Açıklama yüzdeleri yazdırılır.
varypercler;	
cumvarperxler;	
cumvarperyler;	