Dokuz Eylül Üniversitesi İktisadi ve İdari Bilimler Fakültesi Dergisi, Cilt:24, Sayı:2, Yıl:2009, ss.127-138.

Kısmi En Küçük Kareler Regresyon Yöntemi Algoritmalarından Nipals ve PLS - Kernel Algoritmalarının Karşılaştırılması ve Bir Uygulama

Elif BULUT[1]

Aylin ALIN[2]

Alınma Tarihi :Mayıs-2008, Kabul Tarihi:Haziran-2009

Özet

Kısmi en küçük kareler regresyonu, kısmi en küçük kareler analizi (KEKK) ve çoklu doğrusal regresyon analizinden oluşan çok değişkenli istatistiksel bir yöntemdir. Kısmi en küçük kareler yöntemi ile fazla sayıda olan ve aralarında çoklu doğrusal bağlantı bulunan açıklayıcı değişkenler, bağımlı ve açıklayıcı değişkendeki değişimi büyük ölçüde açıklayan daha az sayıda ve aralarında çoklu doğrusal bağlantı sorunu olmayan yeni değişkenlere (bileşen) indirgenmektedir. Elde edilen bileşenlere çoklu doğrusal regresyon analizi uygulanarak regresyon modeli oluşturulmaktadır. Bu çalışmamızda kısmi en küçük kareler regresyon yöntemi algoritmalarından NIPALS ve PLS-KERNEL algoritmalarına değinilerek, bir uygulama üzerinde sonuçlar tartışılmaktadır.

Anahtar Kelimeler: *Kısmi en küçük kareler regresyonu, NIPALS, PLS-KERNEL*

JEL Sınıflandırma Kodları: *C100, C800*

Comparison of Partial Least Squares Regression Method Algorithms: Nipals and PLS-Kernel and An Application

Abstract

Partial Least Squares Regression (PLSR) is a multivariate statistical method that consists of partial least squares and multiple linear regression analysis. Explanatory variables, $\mathbf{X}$, having multicollinearity are reduced to components which explain the great amount of covariance between explanatory and response variable. These components are few in number and they don't have multicollinearity problem. Then multiple linear regression analysis is applied to those components to model the response variable $\mathbf{Y}$. There are various PLSR algorithms. In this study NIPALS and PLS-Kernel algorithms will be studied and illustrated on a real data set.

Keywords: *Partial Least Squares Regression, NIPALS, PLS-Kernel*

JEL Classification Codes: *C100, C800*

[1] Araş. Gör. DEÜ, Fen Edebiyat Fakültesi İstatistik Bölümü, e-posta:elif.bulut@deu.edu.tr

[2] Yrd. Doç. Dr. DEÜ, Fen Edebiyat Fakültesi İstatistik Bölümü

1.Giriş

Veri kümesinde açıklayıcı değişken sayısının çok olması değişkenler arasında çoklu doğrusal bağlantı probleminin olma ihtimalini güçlendirmekte ve değişken sayısının gözlem sayısından çok olması da sıradan en küçük kareler regresyonunu kullanılmaz kılmaktadır. Çoklu doğrusal bağlantı problemi, yapılan analizler sonucunda elde edilen en küçük kareler kestiricilerinin varyans değerlerinin büyük olmasına ve tahminlerin gerçek değerlerinden uzaklaşmasına neden olmaktadır. Böyle bir durumda kullanılabilecek alternatif bir yöntem olan kısmi en küçük kareler regresyonu 1960'lı yıllarda Herman Wold tarafından geliştirilmiş olup, boyut indirgemenin temel alındığı kısmi en küçük kareler analizi ve çoklu doğrusal regresyon yöntemlerinden oluşmaktadır. Çoklu doğrusal bağlantı problemini ortadan kaldırmada, değişkenlerin gözlem sayısından çok olduğu ve gözlemlerin değişken sayısından çok olduğu durumlarda kullanılabilen istatistiksel bir yöntemdir.

KEKK analizinde, aralarında çoklu doğrusal bağlantı olan açıklayıcı değişkenler, algoritmalar yardımıyla hem bağımlı değişkendeki değişimi hemde açıklayıcı değişkenlerdeki değişimi açıklayacak, doğrusal bağlantı problemi ortadan kalkmış olan açıklayıcı değişken sayısından daha az sayıda bileşene indirgenmektedir. Bu indirgeme işleminde Lindgren ve Rännar (1998)'ın çalışmalarında da belirttiği gibi NIPALS, SIMPLS, UNIPAL, SAMPLS ve KERNEL algoritmaları kullanılan algoritmalardan bir kaçıdır.

KEKK algoritmalarında açıklayıcı $(\mathbf{X})$ ve bağımlı $(\mathbf{Y})$ olmak üzere iki değişken matrisi ile de ilgilenilmekte olup amaç $\mathbf{X'Y}$ kovaryans matrisini en çoklayan bileşen sayısını bulmaktır. Farklı veri yapıları için bir çok algoritma geliştirilmiştir, örneğin gözlem sayısının değişken sayısından çok olduğu durumda kullanılan algoritmalar olduğu gibi değişken sayısının gözlem sayıından çok olduğu durumda kullanılan algoritmalar da mevcuttur.

Bu çalışmamızda klasik algoritma olarak da bilinen NIPALS ile değişken sayısının gözlem sayısından çok olduğu durum için geliştirilen PLS-Kernel algoritmasına değinilerek, iki algoritmanın Ondokuz Mayıs Üniversitesi Beden Eğitimi Meslek Yüksek Okulundan alınan veri kümesine uygulanması ile elde edilen sonuçlar verilmektedir. Çalışmamızda matrisler koyu ve büyük, vektörler ise koyu ve küçük harf ile gösterilmiştir. Matrisin transpozu ise " $'$ " simgesi ile gösterilmiştir.

2. NIPALS ve PLS-KERNEL Algoritmaları

2.1. NIPALS Algoritması (Non-Linear Iterative Partial Least Squares)

Klasik, standart algoritma olarak da bilinen NIPALS, KEKK’ in temelini oluşturmaktadır ve tek bağımlı değişken (KEKK1) ve çok bağımlı değişken (KEEK2) durumlarında kullanılabilmektedir. Kovaryans matrisini en çoklayan bileşenleri elde etmeyi amaçlayan algoritmada tüm bileşenler aynı anda elde edilmez. Her bir adımda tek bir bileşen ve bu bileşene ait ağırlık ve yük değerleri elde edilmektedir. Algoritma istenilen bileşen sayısı elde edilince yada $\mathbf{X}$ matrisi sıfır matrisi olunca sonlandırılır. Çalışmamızda Höskuldsson (1988) tarafından verilen algoritma ele alınmıştır. Ayrıca Helland (2001), Kowalski ve Geladi (1986) bu konuda çalışan önemli isimler arasında yer almaktadır. NIPALS yinelemeli bir algoritma olup $N \times K$ boyutlu $\mathbf{X}$, açıklayıcı değişkenler matrisi ve $N \times P$ boyutlu $\mathbf{Y}$, bağımsız değişkenler matrisi ile ilgilenmektedir. Burada, K : açıklayıcı değişken sayısını, P : bağımlı değişken sayısını vermektedir. Algoritmada a bileşen sayısını göstermekte olup $a = 1,2,\ldots,A$ dır. İlk adımda orjinal matrisler $(\mathbf{X}_1 = \mathbf{X}, \mathbf{Y}_1 = \mathbf{Y})$’in kullanıldığı algoritma aşağıdaki adımlardan oluşmaktadır.

Adım 1: Bağımlı değişken çok sayıda ise bu değişkenlerden oluşan $\mathbf{Y}$ matrisinin en yüksek varyansa sahip olan sütunu ya da ilk sütunu, bağımlı değişken sayısı tek ise direkt o değişken sütunu $\mathbf{u_a}$ vektörü olarak alınır.

Adım 2: $\mathbf{X}$’in $\mathbf{Y}$’nin ilgili bileşeni $\mathbf{u_a}$ üzerine regresyonundan $\mathbf{X}$ ve $\mathbf{u}$ arasındaki kovaryansı en çoklayan $\mathbf{w}$ ağırlık vektörü $\mathbf{w_a} = \mathbf{X}'_\mathbf{a}\mathbf{u} / (\mathbf{u}'_\mathbf{a}\mathbf{u_a})$ ile elde edilir.

Adım 3: $\mathbf{w_a} / \|\mathbf{w_a}\|$ ile $\mathbf{w_a}$ vektörü normuna bölünerek boyu 1 olacak şekilde ölçeklendirilir.

Adım 4: $\mathbf{t_a} = \mathbf{X_a}\mathbf{w_a}$ eşitliği ile $\mathbf{X}$’in ilgili bileşeni $\mathbf{t_a}$, $\mathbf{w_a}$ ağırlık vektörü ile $\mathbf{X}$’in doğrusal bir kombinasyonu olacak şekilde hesaplanır.

Adım 5: $\mathbf{t_a}$ bileşeninin $\mathbf{Y}$’yi modellemedeki katkısını açıklayan $\mathbf{c_a}$ ağırlık vektörü $\mathbf{c_a} = \mathbf{Y}'_\mathbf{a}\,\mathbf{t_a} / (\mathbf{t}'_\mathbf{a}\,\mathbf{t_a})$ ile $\mathbf{Y}$’nin $\mathbf{t_a}$ üzerine regresyonundan elde edilir.

Adım 6: $\mathbf{c_a}$ ağırlık vektörü normuna bölünerek boyu 1 olacak şekilde ölçeklendirilir. Yani $\mathbf{c_a} / \|\mathbf{c_a}\|$ hesaplanır.

Adım 7: $\mathbf{Y}$ için ilgili bileşen $\mathbf{u}_{\mathbf{a(yeni)}}$, $\mathbf{c}_\mathbf{a}$ ağırlık vektörü ile $\mathbf{Y}$'nin doğrusal bir kombinasyonunu oluşturacak şekilde $\mathbf{Y}_\mathbf{a}\,\mathbf{c}_\mathbf{a}/(\mathbf{c}'_\mathbf{a}\,\mathbf{c}_\mathbf{a})$ ile hesaplanır.

Adım 8: Adım 2' de kullanılan $\mathbf{u}_\mathbf{a}$ değeri ile Adım 7'de kullanılan $\mathbf{u}_{\mathbf{a(yeni)}}$ değeri arasında bir yakınsama sağlanıp sağlanmadığına bakılır. Bu yakınsama, iki vektörün farkının normunun 10^{-6} gibi sıfıra çok yakın bir değer olması ile tespit edilir. Bu yakınsama sağlanır ise sonraki adımlara geçilerek algoritma sonlandırılır, aksi taktirde Adım 7'de elde edilen $\mathbf{u}_{\mathbf{a(yeni)}}$ değeri Adım 2'de yerine koyularak algoritmaya devam edilir.

Adım 9: $\mathbf{X}$' in ilgili bileşeni $\mathbf{t}_\mathbf{a}$ üzerine regresyonundan, bileşenin açıklayıcı değişken üzerindeki etkisini ifade eden yük vektörü $\mathbf{p}_\mathbf{a}$, $\mathbf{X}'_\mathbf{a}\,\mathbf{t}_\mathbf{a}/(\mathbf{t}'_\mathbf{a}\,\mathbf{t}_\mathbf{a})$ ile elde edilir.

Adım 10: $\mathbf{Y}$'nin ilgili bileşeni $\mathbf{u}_\mathbf{a}$ üzerine regresyonundan, bileşenin bağımlı değişken üzerindeki etkisini ifade eden yük vektörü $\mathbf{q}_\mathbf{a}$, $\mathbf{Y}'_\mathbf{a}\,\mathbf{u}_\mathbf{a}/(\mathbf{u}'_\mathbf{a}\,\mathbf{u}_\mathbf{a})$ ile elde edilir.

Adım 11: Hem $\mathbf{X}$ hem de $\mathbf{Y}$ için bileşenler ayrı hesaplandığından bileşenler arasında zayıf bir ilişki olmakta. Bu durumu ortadan kaldırmak için her bir bileşen için $\mathbf{Y}$'nin ilgili bileşeni $\mathbf{u}_\mathbf{a}$'nın $\mathbf{X}$'in ilgili bileşeni $\mathbf{t}_\mathbf{a}$ üzerine regresyonundan elde edilen içsel bir ilişkiyi tanımlayan b_a katsayısı $\mathrm{b}_\mathbf{a} = \mathbf{u}'_\mathbf{a}\,\mathbf{t}_\mathbf{a}/(\mathbf{t}'_\mathbf{a}\,\mathbf{t}_\mathbf{a})$ ile hesaplanır.

Adım 12: Elde edilen bileşenler ve yükler bağımlı ve açıklayıcı değişkeni modellemede kullanılmaktadır. Sırasıyla açıklayıcı ve bağımlı değişken $\mathbf{X} = \mathbf{TP}'$ ve $\mathbf{Y} = \mathbf{BTC}'$ ile modellenmektedir. Algoritmanın bu adımında bir sonraki bileşeni elde etmek için kullanılacak olan $\mathbf{X}_{\mathbf{a}+1}$ ve $\mathbf{Y}_{\mathbf{a}+1}$ artık matrisleri $\mathbf{X}_{\mathbf{a}+1} \rightarrow \mathbf{X}_\mathbf{a} - \mathbf{t}_\mathbf{a}\mathbf{p}'_\mathbf{a}$ ve $\mathbf{Y}_{\mathbf{a}+1} \rightarrow \mathbf{Y}_\mathbf{a} - \mathbf{bt}_\mathbf{a}\mathbf{c}'_\mathbf{a}$ ile hesaplanmaktadır.

Algoritmaya açıklayıcı değişkenlerdeki ve bağımlı değişkenlerdeki değişimin büyük bir kısmı açıklanıncaya kadar devam edilir. Algoritma ihtiyaç duyulan en az sayıda bileşen sayısını vermektedir.

2.2. PLS-KERNEL Algoritması

Değişken sayısı gözlem sayısından çok olduğu ya da gözlem sayısı değişken sayısından çok olduğu durumda NIPALS algoritması çok fazla zaman alabilmektedir. Bu durumlarda kullanılması önerilen Kernel algoritmaları mevcuttur. Lindgren *vd.* (1993), De Jong ve Ter Braak (1994) gözlem sayısının değişken sayısından çok olduğu durumda kullanılmak amacı ile kernel algoritmaları geliştirmişlerdir. Rännar *vd.* (1994) ise değişken sayısının gözlem sayısından fazla olduğu durumda kullanılacak olan bir kernel algoritması geliştirmişlerdir. Bu çalışmada bu kernel algoritması üzerinde durulacaktır.

PLS-Kernel algoritması, K ' nın N ' ye göre çok fazla olduğu durum $(\mathrm{K} >> \mathrm{N})$ için geliştirilmiş hızlı bir KEKK regresyon algoritmasıdır. Algoritmada $\mathbf{XX}'$ ve $\mathbf{YY}'$ birliktelik matrisleri ve bu matrislerin çarpımından elde edilen $\mathbf{XX}'\mathbf{YY}'$ kernel matrisi kullanılmaktadır. Bu matris değişken sayısından bağımsız olup, daha az sayıdaki gözlem sayısına bağlı olduğu için bize daha küçük bir matris ile çalışma imkanı vermektedir. Algoritma da $\mathrm{N} \times \mathrm{N}$ boyutlu $\mathbf{XX}'$ ve $\mathbf{YY}'$ birliktelik matrislerinin indirgenmesi temel alınmakta ve bu özellik her seferinde daha büyük matris olan $\mathbf{X}$ ve $\mathbf{Y}$ matrislerinin indirgenmesine dayanan NIPALS algoritmasına kıyasla PLS-Kernel algoritmasını daha hızlı ve bellekte daha az yer kaplayan bir algoritma haline getirmektedir.

Orijinal $\mathbf{X}$ ve $\mathbf{Y}$ matrisleri ile başlanan algoritmada aşağıdaki adımlar izlenerek bileşenler ve ağırlıkları elde edilmektedir. NIPALS algoritmasına benzer olarak a bileşen sayısını göstermekte olup $\mathrm{a} = 1,2,\ldots,\mathrm{A}$ 'dır.

Adım 1: $\mathbf{XX}'$ ve $\mathbf{YY}'$ birliktelik matrislerinin çarpımından $\mathbf{XX}'\mathbf{YY}'$ kernel matrisi elde edilir.

Adım 2: Bu matrise özdeğer ayrıştırması uygulanması ile elde edilen en yüksek öz değere karşılık gelen özvektör, $\mathbf{X}$ değişkeninin a 'ncı bileşen değerini $(\mathbf{t}_\mathbf{a})$ oluşturmaktadır.

Adım 3: Adım 2'de elde edilen $\mathbf{t}_\mathbf{a}$ bileşeni normuna bölündükten sonra elde edilen yeni bileşen $\mathbf{t}_{\mathbf{a-yeni}}$ ile $\mathbf{YY}'$ birliktelik matrisinin çarpımından, bağımlı değişkene ait $\mathbf{u}_\mathbf{a}$ bileşeni elde edilir.

Adım 4: Klasik algoritmadakine benzer olacak şekilde bileşen değerlerini elde etmek için, Adım 2 ve Adım 3 ‘de elde edilen bileşen değerleri aşağıdaki şekilde tekrar ölçeklendirilir.

$$\mathbf{u}_{\mathbf{a\text{-geçiş}}} = \frac{\mathbf{u}_{\mathbf{a}}}{\left(\mathbf{t}'_{\mathbf{a}}\mathbf{F}_{\mathbf{a}-1}\mathbf{F}'_{\mathbf{a}-1}\mathbf{t}_{\mathbf{a}}\right)}$$

$$\mathbf{w}'\mathbf{w} = \mathbf{u}'_{\mathbf{a\text{-geçiş}}}\mathbf{E}_{\mathbf{a}-1}\mathbf{E}'_{\mathbf{a}-1}\mathbf{u}_{\mathbf{a\text{-geçiş}}}$$

$$\mathbf{t}_{\mathbf{a\text{-ölçek}}} = \mathbf{t}_{\mathbf{a}}\sqrt{\mathbf{w}'_{\mathbf{a}}\mathbf{w}_{\mathbf{a}}}$$

$$\mathbf{u}_{\mathbf{a\text{-ölçek}}} = \mathbf{u}_{\mathbf{a\text{-geçiş}}}\sqrt{\left(\mathbf{w}'_{\mathbf{a}}\mathbf{w}_{\mathbf{a}}\right)}$$

Tekrar ölçeklendirmeyi sağlamak için $\mathbf{u}_{\mathbf{a-geçiş}}$ olarak tanımlandırılan geçiş vektörü kullanılmaktadır.

Adım 5: $\mathbf{X}_{\mathbf{a}+1}\mathbf{X}'_{\mathbf{a}+1} \rightarrow \mathbf{G}_{\mathbf{a}}\mathbf{X}_{\mathbf{a}}\mathbf{X}'_{\mathbf{a}}\mathbf{G}_{\mathbf{a}}$ ve $\mathbf{Y}_{\mathbf{a}+1}\mathbf{Y}'_{\mathbf{a}+1} \rightarrow \mathbf{G}_{\mathbf{a}}\mathbf{Y}_{\mathbf{a}}\mathbf{Y}'_{\mathbf{a}}\mathbf{G}_{\mathbf{a}}$ eşitlikleri ile indirgenmiş birliktelik matrisleri elde edilmektedir. Burada $\mathbf{G}_{\mathbf{a}} = \mathbf{I} - \mathbf{t}_{\mathbf{a}}\mathbf{t}'_{\mathbf{a}}$ dır.

Adım 6: Ağırlık matrisi $\mathbf{W}$ ve yük matrisleri $\mathbf{P}$ ve $\mathbf{C}$'yi oluşturacak vektörler aşağıdaki eşitliklerle elde edilir.

$$\mathbf{w}_{\mathbf{a}} = \mathbf{X}'_{\mathbf{a}}\mathbf{u}_{\mathbf{a}}$$

$$\mathbf{p}_{\mathbf{a}} = \left(\mathbf{X}'_{\mathbf{a}}\mathbf{t}_{\mathbf{a}}\right)\left(\mathbf{t}'_{\mathbf{a}}\mathbf{t}_{\mathbf{a}}\right)^{-1}$$

$$\mathbf{c}_{\mathbf{a}} = \left(\mathbf{Y}'_{\mathbf{a}}\mathbf{t}_{\mathbf{a}}\right)\left(\mathbf{t}'_{\mathbf{a}}\mathbf{t}_{\mathbf{a}}\right)^{-1}$$

Bu adımlarla, A bileşen saysı elde edilinceye kadar algoritmaya devam edilir.

PLS-Kernel algoritması değişken sayısının gözlem sayısından çok olduğu durumda NIPALS algoritmasına kıyasla daha hızlı bir algoritmadır. Bağımlı değişken sayısının tek olduğu durumda ise $\mathbf{u}$ vektörlerinin yakınsaması tek yinelemede sağlandığı için NIPALS algoritması PLS-Kernel algoritmasına tercih edilmektedir.

3. Uygulama

Çalışmamızda kullanılan veri kümesi 2006 yılında 30 sporcu ile yapılan bir çalışma olup Ondokuz Mayıs Üniversitesi Beden Eğitimi Meslek Yüksek Okulu'ndan alınmıştır. Ölçümler vücut ikiye bölünerek sağ ve sol taraf için elde edilmiştir. Açıklayıcı değişkenler matrisi, kol çevre genişliği, ön kol çevre genişliği, el çevresi, uyluk çevresi, diz çevresi, bacak çevresi, ayak çevresi, kol uzunluğu, el uzunluğu, ayak uzunluğu, karın yağ kıvrımı kalınlığı, kürek kemiği uzunluğu gibi sağlı sollu elde edilen 71 değişkenden oluşmaktadır. Bağımlı değişkenlerimiz ise dikey sıçrama ve çift ayak üzerinde ileriye doğru sıçrama olarak iki tanedir.

Bu durumda X matrisi (30×71), Y matrisi (30×2) olmaktadır.

Uygulamada amaç, hem açıklayıcı değişkendeki hemde bağımlı değişkendeki değişimin büyük kısmını açıklayacak, birbirinden bağımsız bileşenler elde ederek regresyon analizini yapmaktır.

Algoritmalar MATLAB programında yazılarak sonuçlar hesaplatılmıştır.

3.1. NIPALS Algoritması

Yapılan analiz sonucunda algoritmanın $\mathbf{X}$ açıklayıcı değişkenler matrisi sıfır matrisi oluncaya kadar devam etmesiyle, 71 açıklayıcı değişkenden ve 2 bağımlı değişkenden elde edilen 30 bileşenin açıklayıcı değişkenlerdeki ve bağımlı değişkenlerdeki değişimin tamamını açıkladığı görülmüştür.

NIPALS algoritması ile elde edilen 30 bileşene ait bileşen, ağırlık ve yük değerleri Tablo1-Tablo 5 ile verilmektedir.

Tablo 1. $\mathbf{X}$ değişken matrisi için bileşen matrisi $(\mathbf{T})$.

	$\mathbf{t}_1$	$\mathbf{t}_2$	$\mathbf{t}_3$	$\cdots$	$\mathbf{t}_{28}$	$\mathbf{t}_{29}$	$\mathbf{t}_{30}$
1	3.5948	1.6812	-0.6424	…	-0.3273	0.2943	0,0000
2	-3.9780	2.6898	-0.6334	…	0.0107	1.1663	0,0000
3	-1.9234	0.8154	-1.5040	…	-0.3063	-0.7564	0,0000
⋮	⋮	⋮	⋮	…	⋮	⋮	⋮
28	-2.6707	-0.6930	-0.8778	…	0.1427	0.2756	0,0000
29	-1.5199	1.6288	-1.3328	…	0.0715	-0.4336	0,0000
30	-0.3193	0.7215	2.0071	…	-0.5041	0.5501	0,0000

Bileşenler sayıca açıklayıcı değişkenden daha az ve birbirlerine dik olmaktadır.

Tablo 2. $\mathbf{T}$ için ağırlık matrisi $(\mathbf{W})$.

	$\mathbf{w}_1$	$\mathbf{w}_2$	$\mathbf{w}_3$	$\cdots$	$\mathbf{w}_{28}$	$\mathbf{w}_{29}$	$\mathbf{w}_{30}$
1	-0.1624	-0.0958	-0.1574	…	0.2249	-0.0179	-0.0338
2	-0.0321	-0.1283	-0.0887	…	0.0070	-0.2646	0.0914
3	0.0384	0.1374	-0.1777	…	-0.0909	-0.3321	0.0726
⋮	⋮	⋮	⋮	…	⋮	⋮	⋮
69	0.1056	0.0780	-0.0015	…	-0.1516	0.1178	0.0058
70	0.1006	0.0103	-0.0429	…	0.1253	-0.0947	-0.0014
71	0.1836	0.1776	0.1610	…	-0.0398	0.0058	-0.0017

$\mathbf{T} = \mathbf{XW}$ eşitliğinde de ifade edildiği gibi, bileşenler $\mathbf{W}$ ağırlık matrisi ile $\mathbf{X}$'in doğrusal birleşimini oluşturmaktadır. Ağırlık değeri ister negatif ister pozitif olsun büyüklüğü, açıklayıcı değişkenin bileşene katkısını göstermektedir.

Tablo 3. Bileşenler için yük matrisi $(\mathbf{P})$.

	$\mathbf{p}_1$	$\mathbf{p}_2$	$\mathbf{p}_3$	$\cdots$	$\mathbf{p}_{28}$	$\mathbf{p}_{29}$	$\mathbf{p}_{30}$
1	-0.0817	-0.1269	-0.1638	…	0.2289	-0.0179	-0.0338
2	0.0657	-0.1421	-0.1466	…	0.0672	-0.2646	0.0914
3	-0.0496	0.2197	-0.2118	…	-0.0153	-0.3321	0.0726
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
69	0.0520	0.0497	0.0728	…	-0.1784	0.1178	0.0259
70	0.0913	0.1164	0.0261	…	0.1468	-0.0947	-0.0014
71	0.0460	0.1879	0.1943	…	-0.0412	0.0058	-0.0017

Yük matrisi, bileşen matrisi ile çarpılarak $\mathbf{X} = \mathbf{TP}$ eşitliği ile $\mathbf{X}$'i modellemede kullanılmaktadır. Yük değerleri, bileşenin açıklayıcı değişkeni açıklama miktarını göstermektedir.

Tablo 4. $\mathbf{Y}$ değişken matrisi için bileşen matrisi $(\mathbf{U})$.

	$\mathbf{u}_1$	$\mathbf{u}_2$	$\mathbf{u}_3$	$\cdots$	$\mathbf{u}_{28}$	$\mathbf{u}_{29}$	$\mathbf{u}_{30}$
1	1.7137	0.7858	0.0084	…	0.0000	0.0000	0.0000
2	-0.2503	0.7901	0.0999	…	0.0001	0.0001	0.0000
3	-0.2586	0.2183	0.2255	…	-0.0001	0.0000	0.0000
⋮	⋮	⋮	⋮	…	⋮	⋮	⋮
28	-1.2698	-0.6911	0.6606	…	0.0000	0.0000	0.0000
29	-0.9421	-0.6116	-0.0697	…	0.0000	0.0000	0.0000
30	0.6660	0.7146	0.5465	…	0.0000	0.0000	0.0000

Tablo 5. $\mathbf{U}$ için ağırlık matrisi $(\mathbf{C})$.

	$\mathbf{c}_1$	$\mathbf{c}_2$	$\mathbf{c}_3$	$\cdots$	$\mathbf{c}_{28}$	$\mathbf{c}_{29}$	$\mathbf{c}_{30}$
1	0.6487	0.5547	0.9982	…	0.9952	0.8081	0.9989
2	0.7610	0.8321	-0.0609	…	-0.0978	0.5891	0.0462

$\mathbf{t}_\mathbf{a}$ bileşenlerinin aksine $\mathbf{u}_\mathbf{a}$ bileşenleri kendi içlerinde birbirlerine dik olmamakla birlikte bir ya da daha önceki adımda elde edilen $\mathbf{t}$ bileşenlerine diktir. $(\mathbf{u}'_\mathbf{a}\mathbf{t}_\mathbf{b} = 0, \ \ a > b \text{ için})$. $\mathbf{u}_\mathbf{a}$ bileşenleri ile $\mathbf{c}_\mathbf{a}$ ağırlıkları ise $\mathbf{Y} = \mathbf{UC}$ eşitliğinde de ifade edildiği gibi bağımlı değişkeni modellemede kullanılmaktadır. Ağırlık değerinin fazla olması bileşenin bağımlı değişkeni modellemedeki katkısının fazla olduğunu ifade etmektedir.

3.2. PLS-Kernel algoritması

Pls-Kernel algoritması sonucu elde edilen değerler aşağıda verilmektedir.

Tablo 6. $\mathbf{X}$ değişken matrisi için bileşen matrisi $(\mathbf{T})$.

Sporcu	$\mathbf{t}_1$	$\mathbf{t}_2$	$\mathbf{t}_3$	…	$\mathbf{t}_{28}$	$\mathbf{t}_{29}$	$\mathbf{t}_{30}$
1	0.2356	0.1416	-0.0589	…	-0.0924	0.1086	0.1826
2	-0.2607	0.2265	-0.0581	…	0.00319	0.4304	0.1826
3	-0.1260	0.0687	-0.1379	…	-0.0865	-0.2792	0.1826
⋮	⋮	⋮	⋮	…	⋮	⋮	⋮
28	-0.1750	-0.0584	-0.0805	…	0.03966	0.1018	0.1826
29	-0.0996	0.1371	-0.1222	…	0.0209	-0.1601	0.1826
30	-0.0209	0.0607	0.1840	…	-0.1426	0.2030	0.1826

PLS-Kernel algoritmasında $\mathbf{X}$ değişken matrisi için elde edilen bileşen vektörleri normuna bölünerek boyları bir olacak şekilde ölçeklendirilmektedir. Ölçeklendirilmeyen bileşen değerleri ile bileşenler için elde edilen ağırlık ve yük değerleri NIPALS algoritmasında elde edilen değerlerle aynı olmaktadır.

Tablo 7. $\mathbf{Y}$ değişken matrisi için bileşen matrisi $(\mathbf{U})$.

	$\mathbf{u}_1$	$\mathbf{u}_2$	$\mathbf{u}_3$	…	$\mathbf{u}_{28}$	$\mathbf{u}_{29}$	$\mathbf{u}_{30}$
1	6.9132	2.8961	0.0256	…	0.0000	0.0000	0.0000
2	-1.0097	2.9119	0.3033	…	0.0001	0.0001	0.0000
3	-1.0433	0.8044	0.6840	…	-0.0001	0.0000	0.0000
⋮	⋮	⋮	⋮	…	⋮	⋮	⋮
28	-5.1225	-2.5471	2.0037	…	0.0000	0.0000	0.0000
29	-3.8007	-2.2540	-0.2114	…	0.0000	0.0000	0.0000
30	2.6866	2.6335	1.6575	…	0.0000	0.0000	0.0000

Tablo 8. $\mathbf{U}$ için ağırlık matrisi $(\mathbf{C})$.

	$\mathbf{c}_1$	$\mathbf{c}_2$	$\mathbf{c}_3$	…	$\mathbf{c}_{28}$	$\mathbf{c}_{29}$	$\mathbf{c}_{30}$
1	0.1715	0.1721	0.2776	…	0.0002	0.0000	0.06768
2	0.2012	0.2582	-0.0169	…	0.0000	0.0000	0.0031

Tablo 7 ve Tablo 8 ile verilen matrislerin yorumu NIPALS algoritmasındaki gibi yapılmaktadır.

Bileşenleri elde ettikten sonra ikinci adım regresyon analizidir. Regresyon aşamasında ilk 10 bileşen alınarak, hangi bileşenlerin modelde kalması gerektiğini belirlemek için tekli çapraz geçerlilik yöntemi uygulanmıştır. Her bir bağımlı değişken için, bileşenlere ait R^2 değerleri, hata kareler toplamları, PRESS (Predicted REsidual Sum of Squares) değerleri ve 10 bileşen için gerçekleştirilen varyans analizi sonuçları Tablo 9'da verilmektedir.

Tablo 9a. Kısmi en küçük kareler regresyonuna ait ANOVA sonuçları.

1. Dikey Sıçrama için varyans analizi

	sd	Kareler Toplamı	Kareler Ortalaması	F	P
Regresyon	9	1247,77	138,641	135,94	0,000
Hata	20	20,40	1,020		
Toplam	29	1268,17			

Dikey sıçrama için model seçimi ve geçerlilik

Bileşen	X Değişim	Hata Kareler Toplamı	R^2	PRESS
1	0,169899	968,687	0,236152	1509,99
2	0,264025	785,922	0,380269	1787,71
3	0,350130	385,090	0,696341	2215,97
4	0,418499	288,615	0,772416	2088,67
5	0,476348	205,358	0,838067	1907,43
6	0,529145	123,271	0,902796	1627,99
7	0,564710	51,968	0,959021	1486,68
8	0,602974	47,271	0,962725	1469,12
9	0,636995	20,398	0,983916	1382,33
10		7,008	0,994474	1388,20

Tekli çapraz geçerlilik yöntemi, sırası ile her bir gözlemin modelden çıkarılarak geri kalan gözlemler ile model kurmaya dayanmaktadır. Ancak, çıkarılan gözlemler tekrar yerine konmaktadır. Öneğin, 1. gözlem çıkarılarak model kurulduktan sonra, 2. gözlem çıkarıldığı zaman 1. gözlem tekrar yerine konmaktadır. Çıkarılan her bir gözlemin o gözlem olmadan kurulan model yardımı ile tahmin edilerek, bu tahminlere ait hata kareler değerlerinin toplanması ile o modele ait PRESS değeri elde edilmektedir.

Tablo 9a-9b de verilen PRESS değerlerinden dolayı bileşenin her 2 bağımlı değişkeni de tahmin etmede yeterli olduğu görülmektedir. Bu bileşenler açıklayıcı değişkenlerdeki değişimin % 63.7 sini, dikey sıçrama ve çift ayak üzerinde öne sıçrama değişkenlerindeki değişimin ise sırası ile % 98.4 ve %97.4 ünü açıklamaktadır.

Tablo 9b. Kısmi en küçük kareler regresyonuna ait ANOVA sonuçları.

2. Çift ayak üzerinde öne sıçrama için varyans analizi

	sd	Kareler Toplamı	Kareler Ortalaması	F	P
Regresyaon	9	0,577597	0,0641774	83,24	0,000
Hata	20	0,015420	0,0007710		
Toplam	29	0,593017			

Çift ayak üzerinde öne sıçrama için model seçimi ve geçerlilik

Bileşen	X Değişim	Hata Kareler Toplamı	R^2	PRESS
1	0,169899	0,400272	0,325024	0,610212
2	0,264025	0,207980	0,649284	0,757862
3	0,350130	0,207284	0,650459	0,726520
4	0,418499	0,118098	0,800853	0,775235
5	0,476348	0,066685	0,887549	0,718414
6	0,529145	0,045358	0,923513	0,712970
7	0,564710	0,044363	0,925190	0,671187
8	0,602974	0,020641	0,965194	0,737527
9	0,636995	0,015420	0,973998	0,758906
10		0,015339	0,974134	0,753898

4. Sonuç

Bu çalışmada açıklayıcı değişkenler arasında çoklu doğrusal bağlantı problemi olduğunda kullanılan KEKK regresyonu incelenmiştir. KEKK regresyon algoritmalarından NIPALS ve değişken sayısının gözlem sayısından çok olduğu durumda önerilen PLS-Kernel algoritmaları seçilerek, çalışmada kısaca bu algoritmaların tanımı verilmiş ve bir veri kümesine uygulanması sonucunda elde edilen değerler tartışılmıştır. Her iki algoritmada $\mathbf{X'Y}$ kovaryans matrisini en çoklamaya dayanırken ve benzer sonuçları verirken aralarında ki fark hız ve zaman olarak ifade edilmektedir. Rännar vd. (1994)'un belirttiği gibi, değişken sayısının gözlem sayısından çok olduğu veri kümesi için PLS-Kernel algoritması NIPALS algoritmasına kıyasla daha hızlı çalışmaktadır.

Kaynakça

De Jong, S., Ter Braak, C.J.F. (1994). Comments on the Kernel Algorithm. Journal of Chemometrics, 8, 169-174.

Helland, S. I. (2001). Some Theoretical Aspects of Partial Least Squares Regression. Chemometrics and Intelligent Laboratory Systems, 58, 97-107.

Höskuldsson, A. (1988). PLS Regression Methods. Journal of Chemometrics, 2, 211-228.

Kowalski, B.R., Geladi, P., (1986). Partial Least Squares Regression-A Tutorial. Analytica Chimica Acta, 185, 1-17.

Lindgren, F., Geladi, P., Wold, S. (1993). The Kernel Algorithm for PLS. Journal of Chemometrics, 7, 45-59.

Lindgren, F., & Rannar, S. (1998). Alternative Partial Least-Squares (PLS) Algorithm. Perspective in Drug Discovery and Design, 12/13/14. 105-113.

Rännar, S., Lindgren, F., Geladi, P., Wold, S. (1994). A PLS Kernel Algorithm For Data Sets With Many Variables and Fewer Objects. Part1: Theory and Algorithm. Journal of Chemometrics, 8, 111-125.

Wold, H. (1985). Partial Least Squares. Encyclopedia of Statistical Sciences. New York: Wiley, 6, 581-591.