GÖRÜNÜŞTE İLİŞKİSİZ REGRESYON DENKLEMLERİNİN KESTİRİMİNDE KULLANILAN YÖNTEMLERİN KARŞILAŞTIRILMASI

Alper BEKKİ

Doktora Tezi

İstatistik Anabilim Dalı

Haziran 2007

JÜRİ VE ENSTİTÜ ONAYI

Alper BEKKİ'nin **"Görünüşte İlişkisiz Regresyon Denklemlerinin Kestiriminde Kullanılan Yöntemlerin Karşılaştırılması"** başlıklı **İstatistik** Anabilim Dalındaki, Doktora Tezi 29.06.2007 tarihinde, aşağıdaki jüri tarafından Anadolu Üniversitesi Lisansüstü Eğitim - Öğretim ve Sınav Yönetmeliğinin ilgili maddeleri uyarınca değerlendirilerek kabul edilmiştir.

		Adı-Soyadı	İmza
Üye (Tez Danışmanı)	:	**Prof. Dr. Embiya AĞAOĞLU**	………………
Üye (II. Danışman)	:	**Prof. Dr. Memmedağa MEMMEDLİ**	………………
Üye	:	**Prof. Dr. Aydın ERAR**	………………
Üye	:	**Prof. Dr. Hasan DURUCASU**	………………
Üye	:	**Prof. Dr. Ali Fuat YÜZER**	………………

Anadolu Üniversitesi Fen Bilimleri Enstitüsü Yönetim Kurulu'nun ………………… tarih ve ………….. sayılı kararıyla onaylanmıştır.

Enstitü Müdürü

ÖZET

Doktora Tezi

GÖRÜNÜŞTE İLİŞKİSİZ REGRESYON DENKLEMLERİNİN KESTİRİMİNDE KULLANILAN YÖNTEMLERİN KARŞILAŞTIRILMASI

Alper BEKKİ

Anadolu Üniversitesi
Fen Bilimleri Enstitüsü
İstatistik Anabilim Dalı

I. Danışman : Prof. Dr. Embiya AĞAOĞLU
II. Danışman : Prof. Dr. Memmedağa MEMMEDLİ
2007, 93 sayfa

Görünüşte ilişkisiz regresyon modellerinin tahmini ekonometrinin temel konularından birisidir. Bu modeli meydana getiren farklı denklemler için bağımsız değişkenler arasında bir ilişki olmadığı varsayımı altında, aynı zaman noktasında elde edilen hata terimleri arasında ilişki olduğu durumlarda görünüşte ilişkisiz regresyon modeli tahminlerinin en küçük kareler tahminlerinden daha etkin olduğu bilinmektedir.

Bu çalışmada, geniş bir uygulama alanına sahip olan görünüşte ilişkisiz regresyon modellerine ilişkin katsayı tahminlerinin hesaplama açısından daha etkin bir şekilde gerçekleştirilmesi için Matlab paket programı aracılığıyla algoritmalar oluşturulmuş ve deneysel veriler kullanılarak hesaplama açısından etkinlikleri araştırılmıştır.

Anahtar Kelimeler: Görünüşte İlişkisiz Regresyon Modeli, QR ayrışımı, RQ (veya QL) Ayrışımı, Genelleştirilmiş QR Ayrışımı, Matlab.

ABSTRACT

PhD Dissertation

COMPARISON OF THE METHODS THAT ARE USED TO ESTIMATE SEEMINGLY UNRELATED REGRESSION EQUATIONS

Alper BEKKİ

Anadolu University
Graduate School of Sciences
Statistics Program

Supervisor : Prof. Dr. Embiya AĞAOĞLU
Co-Supervisor : Prof. Dr. Memmedağa MEMMEDLİ
2007, 93 pages

Estimation of seemingly unrelated regression models are one of the main subjects of econometrics. It is known that seemingly unrelated regression model estimates are more efficient than the least square estimations, under the assumption that the independent variables of different equations which constitute that model are not related with each other, for the related error terms that are obtained for the same time point.

In this study, algorithms for Matlab are constructed in order to obtain coefficient estimates of seemingly unrelated regression equations which are commonly used in application and more efficient in the manner of calculation, and using empirical data the efficiency of the coefficients in the manner of calculation is examined.

Keywords: Seemingly Unrelated Regression Model, QR Decomposition, RQ (or QL) Decomposition, Generalized QR Decomposition, Matlab.

TEŞEKKÜR

Bu çalışmanın her aşamasında bilgi ve desteklerini esirgemeyen, yönlendirmeleriyle çalışmanın tamamlanmasında büyük pay sahibi olan danışman hocalarım Prof. Dr. Embiya AĞAOĞLU ve Prof. Dr. Memmedağa MEMMEDLİ'ye, katkı ve yol göstermelerinden dolayı tez izleme komitesi üyeleri Prof. Dr. Ali Fuat YÜZER ve Prof. Dr. Hasan DURUCASU'ya, Anadolu Üniversitesi Fen Fakültesi İstatistik Bölümü'nün tüm öğretim elemanlarına ve çalışanlarına teşekkür ederim.

Ayrıca şimdiye kadar sevgi ve sabırla tüm zorlukların aşılmasında her daim yanımda olan hayat arkadaşım Hülya TAN BEKKİ'ye ve yapılan hiçbir şeyin farkında olmasa da sonsuz mutluluk kaynağı kızım Miray'a teşekkür ederim.

Alper BEKKİ

Haziran - 2007

İÇİNDEKİLER

Sayfa

ŞEKİLLER DİZİNİ

ÇİZELGELER DİZİNİ

1. GİRİŞ

Görünüşte İlişkisiz Regresyon modeli (Seemingly Unrelated Regression Model) fikri ilk olarak Zellner (1962) tarafından ortaya atılmıştır. Bu çalışmada Zellner, görünüşte ilişkisiz regresyon modelini meydana getiren farklı denklemler için bağımsız değişkenler arasında bir ilişki olmadığı varsayımı altında, aynı zaman noktasında elde edilen hata terimleri arasında ilişki olduğu durumlarda görünüşte ilişkisiz regresyon tahminlerinin en küçük kareler tahminlerinden daha etkin olduğunu göstermiştir.

Bu anlamda bakıldığında görünüşte ilişkisiz regresyon modelinin tahmini ekonometrik modellerin tahmininde ve analizinde geniş bir uygulanabilirliğe sahiptir. Zaman serileri, panel veri modelleri ve eşanlı denklem sistemleri gibi birçok alanda karşımıza çıkmaktadır (Baltagi, 2001; Dhrymes, 1994; Lütkepohl, 1993; Srivastava ve Giles, 1987). Görünüşte ilişkisiz regresyon modelleri için tahminleme teorisi ve özel durumlarda görünüşte ilişkisiz regresyon modelleri için etkin tahmincilerin elde edilme süreci 40 yıldan uzun bir süre zarfında aktif olarak incelenmesine karşın hesaplamaya dayalı nümerik tahmin yöntemleri son zamanlarda ele alınmaktadır (Dinenis ve Kontoghiorghes, 1997a; Dinenis ve Kontoghiorghes, 1997b, Kontoghiorghes, 2000a; Kontoghiorghes, 2000b; Kontoghiorghes, 2000c). En çok kullanılan tahmin yöntemleri, teorik formülasyonun direkt olarak uygulanması temeline dayanır ki bu küçük hacimli örnek problemlerde dahi hesaplama yükünün aşırı büyük olmasına neden olmaktadır. Bunun yanı sıra hesaplama esnasında ortaya çıkabilecek kötü-koşullandırılmış (ill-conditioned) matrisler model sonuçlarının anlamsız olmasını da beraberinde getirir (Belsley, 1992; Söderkvist, 1996).

Artıkların kovaryans matrisi bilindiğinde, görünüşte ilişkisiz regresyon modeli için ortak olarak en çok kullanılan tahminci Genelleştirilmiş En Küçük Kareler tahmincisidir. Bu tahminci, en iyi doğrusal sapmasız tahminciyi verir. Diğer taraftan kovaryans matrisi bilinmediğinde, tahmin için Uygun (Feasible) Genelleştirilmiş En Küçük Kareler ve En Çok Benzerlik yöntemleri kullanılır. Bu yöntemlere ilişkin tahminciler normal denklemlerin çözümünden elde edilmektedir ki bu da Kronecker çarpımı ve Direkt toplam gibi hesaplamaları bünyesinde barındırmaktadır (Andrews ve Kane, 1970; Regalia ve Mitra, 1989; Srivastava ve Tivari, 1978).

Görünüşte ilişkisiz regresyon modellerinin çözümlemesinde nümerik ve hesaplama araçlarının eşit düzeydeki gelişimleri ekonometrik alanda yapılan teorik ilerlemelerin gerisinde kalmaktadır. Bu nedenle, görünüşte ilişkisiz regresyon modelleri için en iyi doğrusal sapmasız tahmincilerin kestirimine yönelik yapılacak algoritmalar hata varyans-kovaryans matrisinin tekil olmamasını gerektirir ki bu birçok uygulamada mümkün değildir (Judge ve ark., 1985; Theil, 1971).

Aslında görünüşte ilişkisiz regresyon modelleri geniş bir uygulama alanına sahip olmasına rağmen, yapısı itibariyle aşırı büyük boyutlara ulaşan bir problem niteliğinde olduğundan dolayı yapılan uygulama sayısı çok azdır. Örnek verecek olursak, her biri 8 adet bağımsız değişken içeren 10 adet regresyon denkleminden meydana gelen ve her bir değişkenin de 100'er adet gözlem değeri içerdiği bir görünüşte ilişkisiz regresyon modeli tahminleme problemini, 80 adet bağımsız değişken ve 1000 gözlem değerine sahip genel doğrusal modelin tahminleme problemine denk olarak düşünebiliriz. Bundan dolayı bu çalışmada görünüşte ilişkisiz regresyon modellerinin çözümleme sürecinde karşılaşılan zorlukları aşabilmek ve hesap yükünü azaltmak amacıyla oluşturulan ve QR ayrışımı temeline dayanan algoritmalar sunulmuştur. Bu algoritmalar, Matlab paket programı kullanılarak kodlanmış ve kodlanan bu algoritmaların deneysel verilerle hesaplama etkinlikleri karşılaştırılmıştır.

2. REGRESYON MODELLERİNDE KULLANILAN BAZI LİNEER CEBİR KAVRAMLARI

Bu bölümde çalışmanın amacına uygun olarak, görünüşte ilişkisiz regresyon modellerinin çözümleme sürecinin temelini oluşturan bazı lineer cebir kavramları ve bunların özelliklerine yer verilmiştir.

2.1. Kronecker Çarpımı ve Özellikleri

Kronecker çarpımını ilk olarak Alman matematikçi Leopold Kronecker tanımlanmış ve kullanılmıştır. Kronecker çarpımı $\otimes$ simgesiyle ifade edilir ve aslında tensor çarpımının matrislere uygulanan özel bir durumudur. Bu çarpım keyfi boyutlardaki iki matrisin özel blok yapısına sahip daha büyük bir matris şeklinde ifade edilmesidir. Kronecker çarpımı, bütünüyle farklı bir işlem olduğu için bilinen matris çarpımı ile karıştırılmamalıdır. Verilen $m \times n$ boyutlu bir $\mathbf{A}_{m\times n}$ ve $p \times q$ boyutlu bir $\mathbf{B}_{p\times q}$ matrisi için;

$$\mathbf{A} = \begin{bmatrix} a_{11} & \cdots & a_{1n} \\ \vdots & \ddots & \vdots \\ a_{m1} & \cdots & a_{mn} \end{bmatrix}_{m\times n} \quad , \quad \mathbf{B} = \begin{bmatrix} b_{11} & \cdots & b_{1q} \\ \vdots & \ddots & \vdots \\ b_{p1} & \cdots & b_{pq} \end{bmatrix}_{p\times q}$$

bu matrislerin kronecker çarpımı $\mathbf{A} \otimes \mathbf{B}$ ile gösterilir ve $mp \times nq$ boyutlu blok matris;

$$\mathbf{A} \otimes \mathbf{B} = \begin{bmatrix} a_{11}\mathbf{B} & \cdots & a_{1n}\mathbf{B} \\ \vdots & \ddots & \vdots \\ a_{m1}\mathbf{B} & \cdots & a_{mn}\mathbf{B} \end{bmatrix}_{mp\times nq}$$

şeklinde $\mathbf{A}$ matrisinin her bir a_{ij} elemanının, $p \times q$ boyutlu $a_{ij}\mathbf{B}$ matrisi ile yer değiştirmesiyle elde edilir (Harville, 1997; Graham, 1981; Andrews ve Kane, 1970). Açıkça görüldüğü üzere $\mathbf{A}$ ve $\mathbf{B}$ matrislerinin kronecker çarpımı parçalanmış (parçalara ayrılmış, partitioned) bir matristir ve her bir elemanı $p \times q$ boyutlu bloklar olan m adet satır n adet sütuna sahip bir blok matris yapısındadır. Şunu özellikle belirtmek gerekir ki $\mathbf{A}$ ve $\mathbf{B}$ matrislerinin sıradan matris çarpımları - ki bu ancak $n = p$ durumunda geçerlidir - yani $\mathbf{AB}$'nin aksine $\mathbf{A} \otimes \mathbf{B}$ kronecker çarpımı için matrislerin boyutları arasında herhangi bir ilişkinin olması gerekmez.

Henderson ve Searle (1981), kronecker çarpımında yer alan matrisler ile sonuç matrisi arasındaki ilişkileri yapısal matris özellikleri açısından detaylı bir şekilde incelemişlerdir. Uygulama ve hesaplama açısından büyük önem taşıyan bu ilişkilerden bazılarını şu şekilde ifade etmek mümkündür:

$$\mathbf{A} \text{ ve } \mathbf{B} \left\{ \begin{matrix} \text{tekil olmayan} \\ \text{alt (üst) üçgen} \\ \text{bant} \\ \text{simetrik} \\ \text{pozitif tanımlı} \\ \text{stokastik} \\ \text{Toeplitz} \\ \text{permütasyon} \\ \text{ortogonal} \end{matrix} \right\} \text{ matrisler ise,} \quad \mathbf{A} \otimes \mathbf{B} \left\{ \begin{matrix} \text{tekil olmayan} \\ \text{alt (üst) üçgen} \\ \text{blok bant} \\ \text{simetrik} \\ \text{pozitif tanımlı} \\ \text{stokastik} \\ \text{blok Toeplitz} \\ \text{permütasyon} \\ \text{ortogonal} \end{matrix} \right\} \text{ matristir.}$$

Şekil 2.1. Kronecker çarpımında yer alan matrisler ile sonuç matrisi arasındaki ilişkiler

Özellikleri

$\mathbf{A}, \mathbf{B}, \mathbf{C}$ ve $\mathbf{D}$ gerçek değerli matrisler olmak üzere Kronecker çarpımı aşağıdaki özelliklere sahiptir. Ancak bu özelliklerden bazılarında adı geçen matrislerin yapılan işlemlere göre boyutlarının uygun olduğu kabul edilmiştir. Ayrıca, bu çalışmada yoğun olarak kullanılan özellikler ispatları ile beraber verilirken diğer özellikler tanım olarak sunulmuştur.

i. Kronecker çarpımı çifte doğrusal (bi-linear) bir operatördür. Verilen $\alpha \in \mathbb{R}$ keyfi sabiti için,

$$\mathbf{A} \otimes (\alpha \mathbf{B}) = \alpha \left(\mathbf{A} \otimes \mathbf{B} \right)$$
$$(\alpha \mathbf{A}) \otimes \mathbf{B} = \alpha \left(\mathbf{A} \otimes \mathbf{B} \right)$$

ii. Kronecker çarpımının toplama üzerine dağılma özelliği vardır.

$$\left(\mathbf{A} + \mathbf{B} \right) \otimes \mathbf{C} = \left(\mathbf{A} \otimes \mathbf{C} \right) + \left(\mathbf{B} \otimes \mathbf{C} \right)$$
$$\mathbf{A} \otimes \left(\mathbf{B} + \mathbf{C} \right) = \left(\mathbf{A} \otimes \mathbf{B} \right) + \left(\mathbf{A} \otimes \mathbf{C} \right)$$

iii. Kronecker çarpımının birleşim özelliği vardır.

$$\left(\mathbf{A} \otimes \mathbf{B} \right) \otimes \mathbf{C} = \mathbf{A} \otimes \left(\mathbf{B} \otimes \mathbf{C} \right)$$

iv. Kronecker çarpımının yer değiştirme özelliği yoktur.

$$\left(\mathbf{A} \otimes \mathbf{B} \right) \neq \left(\mathbf{B} \otimes \mathbf{A} \right)$$

v. Kronecker çarpımının transpozunun açılımı, normal çarpımdaki transpozun açılımındaki gibi ters sırada değildir.

$$\left(\mathbf{A} \otimes \mathbf{B} \right)^T = \mathbf{A}^T \otimes \mathbf{B}^T$$

vi. Matris çarpımı (boyutlar uygun olmak koşuluyla),

$$\left(\mathbf{A} \otimes \mathbf{B} \right) \left(\mathbf{C} \otimes \mathbf{D} \right) = \left(\mathbf{AC} \right) \otimes \left(\mathbf{BD} \right)$$

İspatı : Herhangi $m \times n$ boyutlu $\mathbf{A} = \left\{ a_{ij} \right\}$, $p \times q$ boyutlu $\mathbf{B} = \left\{ b_{ij} \right\}$, $n \times u$ boyutlu $\mathbf{C} = \left\{ c_{ij} \right\}$ ve $q \times v$ boyutlu $\mathbf{D} = \left\{ d_{ij} \right\}$ matrisleri için;

$\mathbf{A} \otimes \mathbf{B}$ parçalanmış matrisi, ij inci elemanı $a_{ij}\mathbf{B}$ şeklinde $p \times q$ boyutlu bloklardan oluşan m satır ve n sütuna sahip bir matristir. Aynı şekilde $\mathbf{C} \otimes \mathbf{D}$ parçalanmış matrisi ise, jr inci elemanı $c_{jr}\mathbf{D}$ şeklinde $q \times v$ boyutlu bloklardan oluşan n satır ve u sütuna sahip bir matristir. Buna göre $\left(\mathbf{A} \otimes \mathbf{B} \right) \left(\mathbf{C} \otimes \mathbf{D} \right)$

parçalanmış matrisi, ir inci elemanı $\sum_{j=1}^{n}\left(a_{ij}\mathbf{B}\right)\left(c_{jr}\mathbf{D}\right)=\sum_{j=1}^{n}\left(a_{ij}c_{jr}\right)\mathbf{BD}$ şeklinde $p\times v$ boyutlu bloklardan oluşan m satır ve u sütuna sahip bir matristir.

$\left(\mathbf{AC}\right)\otimes\left(\mathbf{BD}\right)$ parçalanmış matrisi, ir inci elemanı $f_{ir}\mathbf{BD}$ şeklinde $p\times v$ boyutlu bloklardan oluşan m satır ve u sütuna sahip bir matristir. Burada f_{ir}, $\mathbf{AC}$ matrisinin ir inci elemanıdır ve $f_{ir}=\sum_{j=1}^{n}a_{ij}c_{jr}$ olduğu kolayca görülebilir.

Eşitliğin her iki tarafı için de ir inci blok elemanlarının aynı olması ile ispat tamamlanmış olur (Harville, 1997).

vii. $\mathbf{A}$ ve $\mathbf{B}$ kare ve tekil olmayan matrisler olmak koşuluyla,

$$\left(\mathbf{A}\otimes\mathbf{B}\right)^{-1}=\mathbf{A}^{-1}\otimes\mathbf{B}^{-1}$$

<u>*İspatı*</u> *:* Herhangi $m\times m$ boyutlu $\mathbf{A}$ ve $p\times p$ boyutlu $\mathbf{B}$ tekil olmayan matrisleri için;

Eşitliğin her iki tarafı da soldan $\left(\mathbf{A}\otimes\mathbf{B}\right)$ ile çarpıldığında,

$$\left(\mathbf{A}\otimes\mathbf{B}\right)\left(\mathbf{A}\otimes\mathbf{B}\right)^{-1}=\left(\mathbf{A}\otimes\mathbf{B}\right)\left(\mathbf{A}^{-1}\otimes\mathbf{B}^{-1}\right)$$

vi. özellik yardımıyla;

$$\mathrm{I}_{mp}=\left(\mathbf{A}\otimes\mathbf{B}\right)\left(\mathbf{A}^{-1}\otimes\mathbf{B}^{-1}\right)$$
$$\mathrm{I}_{mp}=\left(\mathbf{AA}^{-1}\right)\otimes\left(\mathbf{BB}^{-1}\right)$$
$$\mathrm{I}_{mp}=\mathrm{I}_{m}\otimes\mathrm{I}_{p}$$
$$\mathrm{I}_{mp}=\mathrm{I}_{mp}$$

şeklinde ispat tamamlanmış olur (Harville, 1997).

viii. Kronecker çarpımının determinantı,

$$\det\left(\mathbf{A}_{m\times m}\otimes\mathbf{B}_{n\times n}\right)=\det\left(\mathbf{A}\right)^{m}.\det\left(\mathbf{B}\right)^{n}$$

ix. Kronecker çarpımının izi ve dolayısıyla rankı,

$$iz(\mathbf{A}\otimes\mathbf{B}) = iz(\mathbf{A})\, iz(\mathbf{B})$$
$$rank(\mathbf{A}\otimes\mathbf{B}) = rank(\mathbf{A})\, rank(\mathbf{B})$$

2.2. Vec Operatörü ve Özellikleri

Vec operatörü, bir matrisin sütunlarını alt alta istiflemek (yığmak) suretiyle sütun vektörü halinde ifade etmeye yarayan bir operatördür. Benzer şekilde matrisin satırlarını da yan yana istifleyerek bir satır vektörü elde edilebilir ancak uygulamalarda bu çok fazla tercih edilmemektedir. Sütunları $a_i = (a_{1i}, \ldots, a_{mi})^T$ şeklinde ifade edilen $m \times n$ boyutlu bir $\mathbf{A}_{m\times n} = (a_1, \ldots, a_n)$ matrisi için Vec operatörü uygulandığında;

$$Vec(A) = \begin{pmatrix} a_1 \\ \vdots \\ a_n \end{pmatrix}$$

$mn \times 1$ boyutlu bir sütun vektörü elde edilmiş olur. Matrislerle ilgili yapılacak hesaplamalarda özellikle Vec operatörü ile Kronecker çarpımı arasındaki ilişki oldukça önemlidir (Harville, 1997; Fackler, 2005).

Özellikleri

$\mathbf{A}, \mathbf{B}$ ve $\mathbf{C}$ gerçek değerli matrisleri için Vec operatörü ile ilgili özellikler şu şekilde özetlenebilir. Burada verilen özelliklerde, yapılan işlemlere göre bu matrislerin boyutlarının uygun olduğu kabul edilmiştir.

i. Verilen $\alpha \in \mathbb{R}$ keyfi sabiti için,

$$Vec(\alpha\mathbf{A}) = \alpha Vec(\mathbf{A})$$

ii. $Vec(\mathbf{A}+\mathbf{B}) = Vec(\mathbf{A}) + Vec(\mathbf{B})$

iii. $Vec(\mathbf{ABC}) = (\mathbf{C}^T \otimes \mathbf{A})Vec(\mathbf{B})$

<u>*İspatı*</u> *:* Herhangi $m \times n$ boyutlu $\mathbf{A}$, $n \times p$ boyutlu $\mathbf{B}$ ve $p \times q$ boyutlu $\mathbf{C}$ matrisleri için;

$j = 1,\ldots,p$ için, $\mathbf{B}$ matrisinin j inci sütun vektörü $\mathbf{b}_j$ ve $p \times p$ boyutlu $\mathbf{I}_p$ birim matrisinin j inci satır vektörü $\mathbf{e}_j$ olmak üzere $\mathbf{B} = \sum_{j=1}^{p} \mathbf{b}_j \mathbf{e}_j$ olarak ifade edilebilir. Buna göre;

$$
\begin{aligned}
Vec(\mathbf{ABC}) &= Vec\left[\mathbf{A}\left(\sum_j \mathbf{b}_j \mathbf{e}_j\right)\mathbf{C}\right] \\
&= \sum_j Vec\left(\mathbf{A}\mathbf{b}_j \mathbf{e}_j \mathbf{C}\right) \\
&= \sum_j \left[\left(\mathbf{C}^T \mathbf{e}_j^T\right) \otimes \left(\mathbf{A}\mathbf{b}_j\right)\right] \\
&= \sum_j \left(\mathbf{C}^T \otimes \mathbf{A}\right)\left(\mathbf{e}_j^T \otimes \mathbf{b}_j\right) \\
&= \sum_j \left(\mathbf{C}^T \otimes \mathbf{A}\right) Vec\left(\mathbf{b}_j \mathbf{e}_j\right) \\
&= \left(\mathbf{C}^T \otimes \mathbf{A}\right) Vec\left(\sum_j \mathbf{b}_j \mathbf{e}_j\right) \\
&= \left(\mathbf{C}^T \otimes \mathbf{A}\right) Vec(\mathbf{B})
\end{aligned}
$$

şeklinde ispat tamamlanmış olur (Harville, 1997).

iv. $Vec(\mathbf{AB}) = (\mathbf{I} \otimes \mathbf{A})Vec(\mathbf{B}) = (\mathbf{B}^T \otimes \mathbf{I})Vec(\mathbf{A}) = (\mathbf{B}^T \otimes \mathbf{A})Vec(\mathbf{I})$

v. $Vec\left(\mathbf{B}^T\right)^T Vec(\mathbf{A}) = iz(\mathbf{AB}) = iz(\mathbf{BA}) = Vec\left(\mathbf{A}^T\right)^T Vec(\mathbf{B})$

vi. $iz(\mathbf{ABC}) = Vec\left(\mathbf{A}^T\right)^T \left(\mathbf{C}^T \otimes \mathbf{I}\right)Vec(\mathbf{B}) = Vec\left(\mathbf{A}^T\right)^T (\mathbf{I} \otimes \mathbf{B})Vec(\mathbf{C})$

2.3. Permütasyon Matrisi ve Özellikleri

Permütasyon matrisi aslında permütasyonun matris gösterimidir ve her bir satır ve sütununda sadece bir tane "1" değeri olan ve diğer tüm elemanları "0" olan kare matris olarak tanımlanabilir. Permütasyon matrisi bir birim matrisin satır vektörlerinin veya sütun vektörlerinin birbirleri arasında yer değiştirmeleri sonucunda oluşan bir matristir ve uygulanan matrisin satır veya sütunlarının yerlerinin değiştirilmesi için kullanılır. $\mathbf{P}$, $n \times n$ boyutlu bir permütasyon matrisi olsun. Buna göre blok gösterimle;

$$\mathbf{P} = \begin{pmatrix} \mathbf{p}_1 \\ \mathbf{p}_2 \\ \vdots \\ \mathbf{p}_n \end{pmatrix} = \begin{pmatrix} \mathbf{p}_1^T & \mathbf{p}_2^T & \cdots & \mathbf{p}_n^T \end{pmatrix}$$

şeklinde ifade edilir.

Özellikleri

- Permütasyon matrisinin determinantı +1 veya -1'dir. $\text{Yani}, \det(\mathbf{P}) = \pm 1$.
- Permütasyon matrisi tekil değildir ve tersi transpozuna eşittir. Yani, $\mathbf{P}^{-1} = \mathbf{P}^T$.
- Permütasyon matrisinin transpozu ile sağdan ya da soldan çarpımı birim matrise eşittir. Yani, $\mathbf{P}\mathbf{P}^T = \mathbf{P}^T\mathbf{P} = \mathbf{I}_n$.

$\mathbf{P}$ permütasyon matrisinin $n \times n$ boyutlu bir $\mathbf{A}$ kare matrisi ile soldan çarpımı ($\mathbf{PA}$), $\mathbf{A}$ matrisinin satırlarının birbirleriyle yer değiştirmesine neden olurken sağdan çarpımı ($\mathbf{AP}$) ise $\mathbf{A}$ matrisinin sütunlarının birbirleriyle yer değiştirmesine neden olur (Björk, 1996; Golub ve Van Loan, 1996; Harville, 1997; Horn ve Johnson, 1985).

2.4. Ortogonal Ayrışımlar

Bir matrisin, kanonik (standart, bilinen) matris yapılarının çarpımı olarak ifade edilmesi matris ayrışımlarının temel nedenidir. Ayrışımlar özellikle lineer cebir ve uygulamalı istatistikte sıklıkla kullanılan yöntemlerdir. Matris ayrışımlarının yapılmasının temel nedeni kanonik yapılara sahip matris özelliklerinden faydalanarak hesaplamaların kolaylaştırılması ve çözümlemenin basitleştirilmesidir. Ayrıca uygulamalarda ele alınan matrislerin birçoğu matrisin tersi, determinantı, doğrusal sistem çözümlemesi ve en küçük kareler tahminleri gibi bilinen algoritmalara sahip çözümlemeler için uygun değildir. Bu nedenle farklı ayrışım yöntemleri etkin algoritmalar geliştirmek için kullanılırlar.

2.4.1. QR Ayrışımı

QR ayrışımı regresyonda kullanılan önemli hesaplama araçlarından birisidir. Temelde doğrusal sistemlerin çözümlenmesinde kullanılır (Björk, 1996; Fausett ve Ark., 1997; Golub ve Van Loan, 1996; Lawson ve Hanson, 1974). QR ayrışımı, LU ayrışımından ve diğer ayrışımlardan daha doğru çözümler verir ve tekil değer ayrışımından (Singular Value Decomposition) daha az hesaplama gerektirir. İstatistik, ekonometri, optimizasyon ve sinyal işleme gibi ismini sayamadığımız daha birçok alandaki çeşitli uygulamalarda en küçük kareler problemlerinin çözümü ile birlikte karşımıza çıkmaktadır (Lütkepohl, 1993). Bu ayrışımın yapılabilmesi için farklı yöntemler kullanılmaktadır ki bunlar arasında en yaygın kullanılanları Gram-Schmidt süreci, Householder dönüşümü ve Givens rotasyonu yöntemleridir. Ancak özellikle bilgisayar ortamında yapılacak hesaplamalar açısından Householder dönüşüm yöntemi, Gram-Schmidt sürecine göre nümerik açıdan daha istikrarlı sonuçlar verdiği için ve de diğer yöntemlere nazaran daha az yuvarlama hatalarına yol açtığı için en sık kullanılan yöntemdir (Golub ve Van Loan, 1996; Kontoghiorghes, 2000b; Trefethen ve Bau, 1997). Verilen $m \times n$ boyutlu bir A matrisinin QR ayrışımı;

$$\mathbf{A} = \mathbf{QR}$$

şeklindedir ki burada $\mathbf{Q} \in \mathbb{R}^{m\times m}$ ortogonal matristir ve $\mathbf{R}$ matrisi m ve n'in durumuna göre aşağıdaki formlara sahiptir:

$m \geq n$ için
$$\mathbf{R} = \begin{array}{cc} n & \\ \begin{pmatrix} \mathbf{R}_1 \\ \mathbf{0} \end{pmatrix} & \begin{array}{c} n \\ m-n \end{array} \end{array}$$

veya

$m < n$ için
$$\mathbf{R} = \begin{array}{cc} \begin{array}{cc} m & n \end{array} & \\ \begin{pmatrix} \mathbf{R}_{11} & \mathbf{R}_{12} \end{pmatrix} & m \end{array}$$

Burada $n\times n$ boyutlu $\mathbf{R}_1$ ve $m\times m$ boyutlu $\mathbf{R}_{11}$ üst üçgen matrislerdir.

Uygulamalarda en çok karşılaşılan ve en çok bilinen durum $m \geq n$ olduğu durumdur. Bu durum için $\mathbf{A}$ matrisi tam sütun ranklı olduğunda, $\mathbf{R}_1$ matrisi tekil olmayan bir matris olacaktır ve elde edilecek $\mathbf{Q}$ ortogonal matrisinin ilk n sütunu $rank(\mathbf{A})$'nın ortonormal bir tabanı olacaktır (Golub ve Van Loan, 1996). Buna göre $\mathbf{Q} = \begin{pmatrix} \mathbf{Q}_1 & \mathbf{Q}_2 \end{pmatrix}$ şeklinde ilk n sütunu $\mathbf{Q}_1$ ve kalan $m-n$ sütunu $\mathbf{Q}_2$ olacak şekilde parçalara ayrılabilir ki bu durumda QR ayrışımı;

$$\mathbf{A} = \begin{pmatrix} \mathbf{Q}_1 & \mathbf{Q}_2 \end{pmatrix} \begin{pmatrix} \mathbf{R}_1 \\ \mathbf{0} \end{pmatrix} = \mathbf{Q}_1\mathbf{R}_1$$

şekline indirgenmiş olur (Anderson ve Ark., 1995; Björk, 1996).

2.4.2. Householder Dönüşüm Yöntemi

Householder dönüşümü ilk olarak 1958 yılında Alston Scott Householder tarafından öne sürülmüştür (Householder, 1958). Householder vektörü $\mathbf{v} \in \mathbb{R}^m$ sıfır olmamak koşuluyla, $\mathbf{x} \in \mathbb{R}^m$ sütun vektörünü $\mathbf{Hx} = (a, 0, \ldots, 0)^T$ ($a \neq 0$) şekline getiren $\mathbf{H}$ matrisine Householder dönüşümü (Householder matrisi veya Householder yansıması) adı verilir ve şu şekilde hesaplanabilir:

$$\mathbf{H} = \mathbf{I}_m - 2\frac{\mathbf{v}\mathbf{v}^T}{\mathbf{v}^T\mathbf{v}} = \mathbf{I}_m - 2\frac{\mathbf{v}\mathbf{v}^T}{\|\mathbf{v}\|^2}$$

Burada yapılması gereken şey $\mathbf{v}$ Householder vektörünü elde etmektir. Sonuçta elde edilecek $\mathbf{H}$ Householder matrisi simetrik ($\mathbf{H}^T = \mathbf{H}$) ve ortogonaldir ($\mathbf{H}^{-1} = \mathbf{H}^T$). Householder dönüşümü bir vektörün veya matrisin belirli elemanlarını yok etmek ya da diğer bir ifadeyle sıfır yapmak için kullanılır.

$\mathbf{x} \neq 0$ vektörü için $\mathbf{Hx}$ dönüşümünü $\mathbf{e}_1 = \mathbf{I}_{\mathrm{m}}(:,1)$ şeklinde birim matrisin birinci sütun vektörü olarak ifade etmeye çalışalım. Buna göre;

$$\mathbf{Hx} = \left(\mathbf{I}_m - 2\frac{\mathbf{v}\mathbf{v}^T}{\mathbf{v}^T\mathbf{v}} \right)\mathbf{x} = \mathbf{x} - 2\frac{\mathbf{v}^T\mathbf{x}}{\mathbf{v}^T\mathbf{v}}\mathbf{v}$$

Burada $\mathbf{v}$ vektörünü $\mathbf{e}_1$ cinsinden $\mathbf{v} = \mathbf{x} + \alpha\mathbf{e}_1$ şeklinde belirleyelim ve gereken hesaplamaları inceleyelim. Buna göre;

$$\mathbf{v}^T\mathbf{x} = \left(\mathbf{x} + \alpha\mathbf{e}_1\right)^T \mathbf{x} = \mathbf{x}^T\mathbf{x} + \alpha\mathbf{x}_1$$

ve

$$\mathbf{v}^T\mathbf{v} = \left(\mathbf{x} + \alpha\mathbf{e}_1\right)^T \left(\mathbf{x} + \alpha\mathbf{e}_1\right) = \mathbf{x}^T\mathbf{x} + 2\alpha\mathbf{x}_1 + \alpha^2$$

olacaktır ki buradan da,

$$\mathbf{Hx} = \left(1 - 2\frac{\mathbf{x}^T\mathbf{x} + \alpha\mathbf{x}_1}{\mathbf{x}^T\mathbf{x} + 2\alpha\mathbf{x}_1 + \alpha^2} \right)\mathbf{x} - 2\alpha\frac{\mathbf{v}^T\mathbf{x}}{\mathbf{v}^T\mathbf{v}}\mathbf{e}_1$$

olur. İstenilen sonuca ulaşmak için $\mathbf{x}$'in katsayısının sıfır olması gerekmektedir. Dolayısıyla;

$$1 - 2\frac{\mathbf{x}^T\mathbf{x} + \alpha\mathbf{x}_1}{\mathbf{x}^T\mathbf{x} + 2\alpha\mathbf{x}_1 + \alpha^2} = 0 \quad \Rightarrow \quad \alpha = \pm\|\mathbf{x}\|_2$$

olarak elde edilir ki buna göre Householder vektörü;

$$\mathbf{v} = \mathbf{x} \pm \|\mathbf{x}\|_2\, \mathbf{e}_1$$

olarak belirlenmiş olur. Sonuçta,

$$\mathbf{Hx} = \left(\mathbf{I}_m - 2\frac{\mathbf{v}\mathbf{v}^T}{\mathbf{v}^T\mathbf{v}} \right)\mathbf{x} = \pm\|\mathbf{x}\|_2\, \mathbf{e}_1$$

olarak alındığında $\mathbf{x} \in \mathbb{R}^m$ sütun vektörünün ilk elemanı sıfırdan farklı diğer elemanları sıfır olacak şekilde bir vektör elde edilmiş olur (Golub ve Van Loan, 1996).

$$\mathbf{H}\begin{bmatrix} \mathrm{x} \\ \mathrm{x} \\ \mathrm{x} \\ \mathrm{x} \\ \mathrm{x} \end{bmatrix} = \begin{bmatrix} \mathrm{x} \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

Şekil 2.2. $m = 5$ boyutlu bir sütun vektörü için Householder dönüşümü

Verilen bir $\mathbf{x} \in \mathbb{R}^m$ sütun vektörü için Householder dönüşümünün hesaplanmasına ilişkin Matlab kodu Ek-1'de verilmiştir.

Bir $\mathbf{A} \in \mathbb{R}^{m \times n}$ ($m \geq n$) matrisi için Householder dönüşüm yöntemi ile QR ayrışımı elde edilmek istendiğinde yukarıda belirtilen adımlar matrisin her bir sütun vektörü için tekrarlanmak suretiyle (toplam n defa) ve her bir adımda ele alınan sütun vektörünün boyutu üstteki bir birim atılmak suretiyle küçültülerek yapılır. Sonuçta elde edilen n adet Householder matrisinin çarpımı bize ortogonal $\mathbf{H}_1\mathbf{H}_2 \ldots \mathbf{H}_n = \mathbf{Q} \in \mathbb{R}^{m \times m}$ matrisini verirken bu matrisin transpozunun $\mathbf{A}$ matrisi ile soldan çarpımı da $\mathbf{Q}^T\mathbf{A} = \mathbf{R} \in \mathbb{R}^{m \times n}$ üst üçgen matrisini verecektir. Burada,

$$\mathbf{H}_i = \mathbf{I}_\mathrm{m} - 2\frac{\mathbf{h}_i\mathbf{h}_i^T}{\mathbf{h}_i^T\mathbf{h}_i} \text{ ve } \quad \mathbf{h}_i = \begin{pmatrix} \mathbf{0} \\ \tilde{\mathbf{h}}_i \end{pmatrix} \begin{matrix} i-1 \\ m-i+1 \end{matrix} \qquad (i = 1 \ldots n)$$

şeklindedir.

$$\mathbf{H}_3\mathbf{H}_2\mathbf{H}_1 \underbrace{\begin{bmatrix} \mathrm{x} & \mathrm{x} & \mathrm{x} \\ \mathrm{x} & \mathrm{x} & \mathrm{x} \\ \mathrm{x} & \mathrm{x} & \mathrm{x} \\ \mathrm{x} & \mathrm{x} & \mathrm{x} \end{bmatrix}}_{\mathbf{A}} = \begin{bmatrix} \mathrm{x} & \mathrm{x} & \mathrm{x} \\ 0 & \mathrm{x} & \mathrm{x} \\ 0 & 0 & \mathrm{x} \\ 0 & 0 & 0 \end{bmatrix} = \mathbf{R}$$

Şekil 2.3. $m = 4$, $n = 3$ boyutlu bir $\mathbf{A} \in \mathbb{R}^{m \times n}$ matrisi için Householder dönüşümü ile QR ayrışımı

Verilen bir $\mathbf{A} \in \mathbb{R}^{m \times n}$ matrisine Householder dönüşümü yardımıyla QR ayrışımının hesaplanmasına ilişkin Matlab kodu Ek-2'de verilmiştir.

2.4.3. Givens Rotasyon Yöntemi

Householder dönüşüm yöntemi, bir vektörün ilk değeri haricindeki tüm değerleri topluca sıfır yapmak için oldukça kullanışlı bir yöntemdir. Ancak bu yöntem ile büyük ölçekte bir yok etme gerçekleştirilmektedir. Bu yüzden daha seçimsel bir yöntem olan Givens rotasyon yöntemi 1950'li yıllarda James Wallace Givens tarafından öne sürülmüştür (Givens, 1954; Givens, 1958). Bu yöntem Householder dönüşümünden farklı olarak matrisin sadece belirlenen iki değerini (koordinatını) değiştirir ve bunlardan sadece bir tanesini sıfıra çevirir. Diğer elemanların değerleri değişmez (Golub ve Van Loan, 1996).

$m \times m$ boyutlu bir Givens rotasyonu birim matrisin rank-iki düzeltmesi (rank-two correction) olarak tanımlanır ve

$$\mathbf{G}_{i,j} = \begin{array}{c} \begin{array}{ccccc} & i & & j & \\ & \downarrow & & \downarrow & \end{array} \\ \begin{bmatrix} \mathbf{I}_{i-1} & 0 & 0 & 0 & 0 \\ 0 & c & 0 & s & 0 \\ 0 & 0 & \mathbf{I}_{j-i-1} & 0 & 0 \\ 0 & -s & 0 & c & 0 \\ 0 & 0 & 0 & 0 & \mathbf{I}_{m-j-1} \end{bmatrix} \end{array} \begin{array}{l} \\ \\ \\ \leftarrow i \\ \\ \leftarrow j \\ \\ \end{array}$$

yapısal biçimine sahip ortogonal bir matristir. Burada bazı θ değerleri için $c = \cos(\theta)$, $s = \sin(\theta)$ olarak hesaplanır ve $c^2 + s^2 = 1$'dir. $\mathbf{G}_{i,j}^T$ rotasyonu bir matrisin soluna uygulandığında sadece i inci ve j inci satırlara etki eder ve j inci satırdaki i inci elemanı yok eder.

Örneğin $\mathbf{A} \in \mathbb{R}^{m \times n}$ matrisi için ve $\tilde{\mathbf{A}} = \mathbf{G}_{i,j}^T \mathbf{A}$ iken $\tilde{\mathbf{A}}$ 'nın p inci satırı;

$$\tilde{\mathbf{A}}_{p,:} = \begin{cases} c\mathbf{A}_{i,:} - s\mathbf{A}_{j,:} & , \; p = i \quad \text{ise} \\ s\mathbf{A}_{i,:} + c\mathbf{A}_{j,:} & , \; p = j \quad \text{ise} \\ \mathbf{A}_{p,:} & , \; p \neq i, j \text{ ise} \end{cases}$$

şeklinde değiştirilmiş olur. Böylece şayet $\mathbf{A}$ matrisinin (j,k) ıncı elemanı olan $a_{j,k}$ sıfırdan farklı bir değerse, $c = a_{i,k}/r$, $s = -a_{j,k}/r$ ve $r = \sqrt{a_{i,k}^2 + a_{j,k}^2}$ olarak belirlenmek koşuluyla Givens rotasyon yöntemi kullanılarak yok edilebilir (Golub ve Van Loan, 1996). Bu rotasyon aşağıdaki şekilde ifade edilir.

$$\begin{bmatrix} c & s \\ -s & c \end{bmatrix}^T \begin{bmatrix} a_{i,k} \\ a_{j,k} \end{bmatrix} = \begin{bmatrix} r \\ 0 \end{bmatrix}$$

a ve b şeklinde verilen iki değerden istenilen değeri yok etmek ya da diğer bir deyişle sıfır yapmak için kullanılan Givens rotasyonu hesaplanmasına ilişkin Matlab kodu Ek-3'de verilmiştir.

Givens rotasyon yöntemi sütun veya satır tabanlı olmak üzere iki farklı şekilde hesaplanabilir. Örneğin sütun tabanlı stratejide her bir sütun için ana köşegen altındaki değerlerin her birisi için Givens rotasyonu ard arda uygulanmak suretiyle QR ayrışımı hesaplanabilir. Sonuçta $\mathbf{Q}$ ortogonal matrisi,

$$\mathbf{Q} = \left(\underbrace{\mathbf{G}_{m-1,m} \cdots \mathbf{G}_{1,2}}_{\text{1. SÜTUN}} \right) \left(\underbrace{\mathbf{G}_{m-1,m} \cdots \mathbf{G}_{2,3}}_{\text{2. SÜTUN}} \right) \cdots \left(\underbrace{\mathbf{G}_{m-1,m} \cdots \mathbf{G}_{n,n+1}}_{\text{n. SÜTUN}} \right)$$

şeklinde elde edilir. Burada $\left(\mathbf{G}_{m-1,m} \ldots \mathbf{G}_{i,i+1}\right)$ rotasyonu sırasıyla $(m,i),\ldots,(i+1,i)$ elemanları yok eder. Yani i inci sütunun $m-i$ tane elemanını sıfır yapar ve daha önceden sıfır yapılan elemanların değerlerini korur.

$$\begin{bmatrix} x & x & x \\ x & x & x \\ x & x & x \\ x & x & x \end{bmatrix} \xrightarrow{(3,4)} \begin{bmatrix} x & x & x \\ x & x & x \\ x & x & x \\ 0 & x & x \end{bmatrix} \xrightarrow{(2,3)} \begin{bmatrix} x & x & x \\ x & x & x \\ 0 & x & x \\ 0 & x & x \end{bmatrix} \xrightarrow{(1,2)}$$

$$\begin{bmatrix} x & x & x \\ 0 & x & x \\ 0 & x & x \\ 0 & x & x \end{bmatrix} \xrightarrow{(3,4)} \begin{bmatrix} x & x & x \\ 0 & x & x \\ 0 & x & x \\ 0 & 0 & x \end{bmatrix} \xrightarrow{(2,3)} \begin{bmatrix} x & x & x \\ 0 & x & x \\ 0 & 0 & x \\ 0 & 0 & x \end{bmatrix} \xrightarrow{(3,4)} \mathbf{R}$$

Şekil 2.4. $m = 4$, $n = 3$ boyutlu bir $\mathbf{A}$ matrisi için Givens rotasyonu ile QR ayrışımı

Verilen bir $\mathbf{A} \in \mathbb{R}^{m \times n}$ $(m \geq n)$ matrisine Givens rotasyon yöntemi yardımıyla QR ayrışımının hesaplanmasına ilişkin Matlab kodu Ek-4'de verilmiştir.

2.5. RQ veya QL Ayrışımı

RQ (veya QL) ayrışımı, özellikle Genelleştirilmiş En Küçük Kareler problemlerinde çözümü basitleştirmek ve hesaplama algoritmalarını hızlandırmak amacıyla kullanılan bir ayrışımdır. Özellikle büyük boyutlu problemlerde matrislerin çarpımı, bölümü ve terslerinin alınmasından doğabilecek hesaplama zorluklarını ve hataları azaltmak için kullanılır. RQ ayrışımı, yapısı itibariyle QR ayrışımına oldukça benzemektedir. Genel olarak bir $\mathbf{B}$ matrisinin RQ ayrışımı;

$$\mathbf{B} = \mathbf{RQ} \qquad \text{veya} \qquad \mathbf{B}^T = \mathbf{Q}^T \mathbf{R}^T = \mathbf{PL}$$

şeklinde ifade edilir. Burada $\mathbf{Q}$ ve $\mathbf{P}\,(=\mathbf{Q}^T)$ ortogonal, $\mathbf{R}$ üst üçgen ve dolayısıyla $\mathbf{L}\,(=\mathbf{R}^T)$ alt üçgen matrislerdir.

Burada QL ayrışımının özellikle vurgulanmasının nedeni, RQ ayrışımı yapabilmek için oluşturulacak bilgisayar algoritmalarında daha basit ve anlaşılabilir bir kodlamaya imkan vermesinden kaynaklanmaktadır.

Bölüm 2.4.2'de ayrıntılı bir şekilde değinilen Householder dönüşüm yöntemi, QL ayrışımının bilgisayar algoritmasının oluşturulmasında da kullanılabilir. QL ayrışımı için oluşturulacak algoritmada QR ayrışımından farklı olarak Householder dönüşümü bir sütun vektörü için değil bir satır vektörü için gerçekleştirilir. Dolayısıyla QL ayrışımının gerçekleştirilebilmesi için amaç, ilgilenilen satır vektörünün son elemanı haricindeki tüm elemanlarının yok edilmesi veya diğer bir deyişle sıfır yapılması olacaktır. $\mathbf{x} \in \mathbb{R}^n$ şeklindeki bir satır vektörü için Householder dönüşümü hesaplandığında elde edilen $\tilde{\mathbf{H}}\mathbf{x}$ dönüşümü $\mathbf{x}$ satır vektörünün n inci elemanına kadar olan $n-1$ tane elmanı sıfır yapacaktır.

$$\tilde{\mathbf{H}}\begin{bmatrix} \mathrm{x} & \mathrm{x} & \mathrm{x} & \mathrm{x} & \mathrm{x} \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 0 & \mathrm{x} \end{bmatrix}$$

Şekil 2.5. $n = 5$ boyutlu bir satır vektörü için Householder dönüşümü

Verilen bir $\mathbf{x} \in \mathbb{R}^n$ satır vektörü için Householder dönüşümünün hesaplanmasına ilişkin Matlab kodu Ek-5'de verilmiştir.

Bir $\mathbf{B} \in \mathbb{R}^{n \times m}$ matrisi için Householder dönüşüm yöntemi ile RQ ayrışımı elde edilmek istendiğinde $\mathbf{B}^T = \mathbf{PL}$ şeklinde QL ayrışımının yapılması yeterli olacaktır. Buna göre QL ayrışımı gerçekleştirebilmek için yukarıda tek bir satır vektörü için yapılan dönüşümün matrisin her bir satır vektörü için m inci satırdan 1 inci satıra kadar (toplam m defa) tekrarlanması gerekir. Her bir adımda ele alınan satır vektörünün boyutu satır sonundan bir birim atılmak suretiyle küçültülerek yapılır. Sonuçta elde edilen m adet Householder matrisinin çarpımı bize ortogonal $\tilde{\mathbf{H}}_m \tilde{\mathbf{H}}_{m-1} \ldots \tilde{\mathbf{H}}_1 = \mathbf{P} \in \mathbb{R}^{m \times m}$ matrisini verir. Buradan elde edilen $\mathbf{P}$ ortogonal matrisinin transpozunun $\mathbf{B}^T$ matrisi ile soldan çarpımı da $\mathbf{P}^T \mathbf{B}^T = \mathbf{L} \in \mathbb{R}^{m \times n}$ alt üçgen matrisini verecektir. Burada her iki tarafın transpozu alındığında $\left(\mathbf{P}^T \mathbf{B}^T\right)^T = \mathbf{L}^T \Rightarrow \mathbf{BP} = \mathbf{R} \in \mathbb{R}^{n \times m}$ şeklinde RQ ($\mathbf{Q} = \mathbf{P}^T$) ayrışımı yapılmış olur.

$$\underbrace{\begin{bmatrix} \boxed{x \quad x} \quad x \quad x \\ \boxed{x \quad x \quad x} \quad x \\ \boxed{x \quad x \quad x \quad x} \end{bmatrix}}_{\mathbf{B}} \tilde{\mathbf{H}}_3\tilde{\mathbf{H}}_2\tilde{\mathbf{H}}_1 = \begin{bmatrix} 0 & x & x & x \\ 0 & 0 & x & x \\ 0 & 0 & 0 & x \end{bmatrix} = \mathbf{R}$$

Şekil 2.6. $n = 3$, $m = 4$ boyutlu bir $\mathbf{B} \in \mathbb{R}^{n \times m}$ matrisi için Householder dönüşümü ile RQ ayrışımı

Verilen bir $\mathbf{B} \in \mathbb{R}^{n \times m}$ matrisine Householder dönüşümü yardımıyla QL ayrışımının hesaplanmasına ilişkin Matlab kodu Ek-6'da verilmiştir.

2.6. Genelleştirilmiş QR Ayrışımı

Genelleştirilmiş QR ayrışımı ilk olarak Hammarling (1976) tarafından ortaya atılmıştır. Kavramsal temeli ve bir hesaplama aracı olarak sistematik kullanımı ise Paige (1990) tarafından yapılmıştır. Paige çalışmasında, bu ayrışımların çeşitli en küçük kareler problemlerinin genel formülasyonlarının çözümlemesine imkan sağladıklarını göstermiştir. Bilgisayar ortamındaki hesaplama algoritmaları ise Anderson ve Ark. (1992; 1995) tarafından oluşturulmuştur.

Ortogonal ayrışımların farklı genelleştirmelerinin ortaya çıkma nedeni temel olarak matrislerin bölüm ve çarpımlarının kesin hesaplamalarından kaçınmaktır. Örneğin $\mathbf{A}$ ve $\mathbf{B}$ kare ve tekil olmayan matrisleri için $\mathbf{B}^{-1}\mathbf{A}$ 'nın QR ayrışımını elde etmek istediğimizde $\mathbf{B}^{-1}\mathbf{A}$ kesin hesabı doğruluğun kaybına neden olabilir ve bu hesaplamadan kaçınmak gerekir (Björk, 1994; 1996).

$\mathbf{A} \in \mathbb{R}^{m \times n}$ ve $\mathbf{B} \in \mathbb{R}^{m \times p}$ matrisleri için Genelleştirilmiş QR ayrışımı;

$$\mathbf{A} = \mathbf{QR} \qquad , \qquad \mathbf{B} = \mathbf{QTZ}$$

şeklinde hesaplanır. Burada $\mathbf{Q} \in \mathbb{R}^{m \times m}$ ve $\mathbf{Z} \in \mathbb{R}^{p \times p}$ ortogonal matrislerdir. $\mathbf{R}$ ve $\mathbf{T}$ matrisleri ise m, n ve p arasındaki ilişkiye göre;

$$\mathbf{R} = \begin{pmatrix} \mathbf{R}_{11} \\ \mathbf{0} \end{pmatrix} \qquad m \geq n \qquad \text{veya} \qquad \mathbf{R} = \begin{pmatrix} \mathbf{R}_{11} & \mathbf{R}_{12} \end{pmatrix} \qquad m < n$$

ve

$$\mathbf{T} = \begin{pmatrix} \mathbf{0} & \mathbf{T}_{12} \end{pmatrix} \qquad m \leq p \qquad \text{veya} \qquad \mathbf{T} = \begin{pmatrix} \mathbf{T}_{11} \\ \mathbf{T}_{12} \end{pmatrix} \qquad m > p$$

formlarından birisi olur. Burada $\mathbf{R}_{11}$ üst üçgen ve $\mathbf{T}_{12}$ alt üçgen matrislerdir. Şayet $\mathbf{B}$ matrisi tekil olmayan ve kare bir matris olursa Genelleştirilmiş QR ayrışımı tam olarak $\mathbf{B}^{-1}\mathbf{A}$'nın QR faktörizasyonunu verir. Benzer şekilde $\mathbf{AB}^{-1}$'in QR faktörizasyonu ise Genelleştirilmiş RQ ayrışımı olacaktır (Anderson ve Ark., 1992).

2.7. Cholesky Ayrışımı

Herhangi bir tekil olmayan $\mathbf{A}$ matrisi $\mathbf{A} = \mathbf{LU}$ şeklinde bir $\mathbf{L}$ alt üçgen matrisi ile $\mathbf{U}$ üst üçgen matrislerinin çarpımı şeklinde ayrıştırılabilir. Bu ayrışım LU ayrışımı olarak bilinir. Şayet $\mathbf{A}$ matrisi simetrik ve pozitif tanımlı bir matris olursa bu durumda $\mathbf{A} = \mathbf{LL}^T$ (veya $\mathbf{A} = \mathbf{U}^T\mathbf{U}$) şeklinde alt üçgen bir $\mathbf{L}$ matrisi ve onun transpozu olan üst üçgen matrisin çarpımı şeklinde ayrıştırılabilir. Bu ayrışım literatürde Cholesky ayrışımı olarak bilinir ve adını bunu ilk defa ortaya atan fransız matematikçi André-Louis Cholesky'den almıştır. Cholesky ayrışımı, A matrisinin karekökünün alınması olarak da değerlendirilebilir (Golub ve Van Loan, 1996; Harville, 1997; Horn ve Johnson, 1985).

Genel olarak LU ayrışımı doğrusal denklem sistemlerinin çözümlenmesinde sıkça kullanılan bir ayrışım yöntemidir. Buna bağlı olarak Cholesky ayrışımı, doğrusal en küçük kareler problemlerinde normal denklemlerin çözümlenmesinde oldukça sık kullanılan bir ayrışımdır. Cholesky ayrışımının en önemli avantajı, LU ayrışımına oranla hesaplamaların iki kat daha hızlı bir şekilde gerçekleşmesidir (Björk, 1996; Golub ve Van Loan, 1996).

Şayet $\mathbf{A}$ matrisi pozitif tanımlı değilse ancak simetrik ise bu durumda Cholesky ayrışımı kullanılamaz. Onun yerine Cholesky ayrışımına göre daha yavaş ancak LU ayrışımına göre daha hızlı bir algoritmaya sahip olan bir diğer ayrışım yöntemi olan $\mathbf{A} = \mathbf{L}\mathbf{D}\mathbf{L}^T$ (veya $\mathbf{A} = \mathbf{U}\mathbf{D}\mathbf{U}^T$) ayrışımı kullanılabilir. Burada $\mathbf{D}$ matrisi köşegen matristir (Harville, 1997).

3. DOĞRUSAL MODELLER

İstatistikteki genel problem aralarında ilişki olduğu varsayılan bir veya daha fazla değişkene ilişkin parametrelerin tahminlenmesidir. Bu tarz bir ilişki;

$$y = f(x_1, x_2, \ldots, x_n) \tag{3.1}$$

şeklinde ifade edilebilir. Burada y, çıktı değişkeni ve $x_1, x_2, \ldots, x_n$ girdi değişkenleridir. Regresyon analizi, değişkenlerin gözlenen değerleri kullanılmak suretiyle, (3.1) ile ifade edilen ilişkinin şeklinin tahminlenmesidir. Bu değişkenler arasında nasıl bir ilişki olduğunun belirlenmesi model kurulumu olarak adlandırılır. Genelde y'deki değişime katkısı bulunan ve gözlemlenmemiş değişkenler de vardır. Bundan dolayı (3.1) ile verilen ilişkiyi şu şekilde tanımlamak gerekir:

$$y = f(x_1, x_2, \ldots, x_n) + \varepsilon \tag{3.2}$$

Burada ε, herhangi bir gözleme ilişkin değeri kestirilememiş hata terimini ifade eder. ε'un amacı y'nin gerçek gözlem değeri ile kesin bir fonksiyonel ilişki ile elde edilen değer arasındaki farklılığı tanımlamaktır. y'nin gözlemlenen değeri ile kestirilen değeri arasındaki fark artık olarak nitelendirilir.

Doğrusal bir model, bir bağımlı (açıklanan) değişken y'nin, bağımsız (açıklayıcı) değişkenler $x_1, \ldots, x_n$'in doğrusal bir fonksiyonu biçiminde ifade edilmesidir. Doğrusal model genel gösterimle;

$$y = \beta_1 x_1 + \beta_2 x_2 + \ldots + \beta_n x_n + \varepsilon$$

şeklinde ifade edilir ki burada $\beta_i \ (i = 1, \ldots, n)$ bilinmeyen katsayıları ve ε ise kestirilemeyen değerlerden kaynaklanan hata terimi veya artıktır. m tane gözlem değeri için bu ilişkiyi şu şekilde ifade edebiliriz:

$$\begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_m \end{pmatrix} = \begin{pmatrix} x_{11} & x_{12} & \cdots & x_{1n} \\ x_{21} & x_{22} & \cdots & x_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ x_{m1} & x_{m2} & \cdots & x_{mn} \end{pmatrix} \begin{pmatrix} \beta_1 \\ \beta_2 \\ \vdots \\ \beta_m \end{pmatrix} + \begin{pmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \vdots \\ \varepsilon_m \end{pmatrix}$$

Kısa gösterimle bu modeli,

$$\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\varepsilon} \tag{3.3}$$

şeklinde ifade edebiliriz. Burada $\mathbf{y}, \boldsymbol{\varepsilon} \in \mathbb{R}^m$, $\mathbf{X} \in \mathbb{R}^{m \times n}$ ve $\boldsymbol{\beta} \in \mathbb{R}^n$'dir. (3.1)'deki doğrusal modeli tamamlayabilmek için ek varsayımların yapılması zorunludur. İlk varsayım, $\boldsymbol{\varepsilon}$'un beklenen değeri sıfırdır yani $E(\boldsymbol{\varepsilon}) = 0$. İkinci varsayım, $\mathbf{X}$ stokastik olmayan bir matristir yani $E(\mathbf{X}^T \boldsymbol{\varepsilon}) = 0$. Üçüncü ve son varsayım ise $\boldsymbol{\varepsilon}$'un varyans-kovaryans matrisi $\sigma^2 \boldsymbol{\Omega}$ olmak üzere normal dağılıma sahip olmasıdır. Burada $\boldsymbol{\Omega}$, simetrik ve negatif olmayan tanımlı bir matristir ve σ bilinmeyen sabittir. Sonuçta, (genel) doğrusal modelin matematiksel tanımı tamamlanmış olur ve

$$\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\varepsilon} \quad , \quad \boldsymbol{\varepsilon} \sim \left(0, \sigma^2 \boldsymbol{\Omega}\right) \tag{3.4}$$

olarak ifade edilir. Buradaki $\boldsymbol{\varepsilon} \sim \left(0, \sigma^2 \boldsymbol{\Omega}\right)$ ifadesi $\boldsymbol{\varepsilon}$ hata vektörünün ortalaması sıfır ve varyans-kovaryans matrisi $\sigma^2 \boldsymbol{\Omega}$ olan bir dağılımdan geldiğini ifade eder (Rao ve Toutenburg, 1995; Davidson ve MacKinnon, 2004).

3.1. Sıradan Doğrusal Model

Sıradan Doğrusal Modeli;

$$\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\varepsilon} \quad , \quad \boldsymbol{\varepsilon} \sim \left(0, \sigma^2 \mathbf{I}_m\right) \tag{3.5}$$

şeklinde yazalım. Burada $\mathbf{y} \in \mathbb{R}^m$ bağımlı değişken vektörü, $\mathbf{X} \in \mathbb{R}^{m \times n}$ tam sütun ranklı bağımsız değişkenler matrisi, $\boldsymbol{\beta} \in \mathbb{R}^n$ tahminlenecek katsayılar vektörü ve $\boldsymbol{\varepsilon} \in \mathbb{R}^m$ hata vektörüdür. Sıradan doğrusal model için, her bir $\boldsymbol{\varepsilon}_i$ eşit varyanslıdır yani $Var(\varepsilon_i) = \sigma^2$ ve hata değerleri arasında ilişki yoktur yani $E\left(\varepsilon_i^T \varepsilon_j\right) = 0, \quad \left(1 \le i \ne j \le m\right)$ varsayımlarını sağladığı kabul edilir.

(3.5) modelinin Sıradan En Küçük Kareler tahmincisi;

$$\boldsymbol{\varepsilon}^T \boldsymbol{\varepsilon} = \left(\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\right)^T \left(\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\right)$$

ifadesinin minimize edilmesi ile elde edilir. Bu ifadenin $\boldsymbol{\beta}$'ya göre türevi alınıp sıfıra eşitlenerek $\partial\left(\boldsymbol{\varepsilon}^T \boldsymbol{\varepsilon}\right) / \partial \boldsymbol{\beta} = 0$ çözülürse en küçük kareler normal denklemleri

$\mathbf{X}^T\mathbf{X}\boldsymbol{\beta} = \mathbf{X}^T\mathbf{y}$ olarak elde edilir ki buradan da sıradan en küçük kareler tahmincisi $\hat{\boldsymbol{\beta}} = \left(\mathbf{X}^T\mathbf{X}\right)^{-1}\mathbf{X}^T\mathbf{y}$ olarak elde edilir. Sıradan en küçük kareler tahmincisi, (3.5) doğrusal modeli için En İyi Doğrusal Sapmasız Tahmincidir.

Alternatif olarak, $\mathbf{X}$ açıklayıcı değişken matrisinin QR ayrışımı yapılırsa (m>n için);

$$\mathbf{Q}^T\mathbf{X} = \begin{array}{cc} \begin{pmatrix} \mathbf{R} \\ \mathbf{0} \end{pmatrix} & \begin{array}{c} n \\ m-n \end{array} \end{array} \qquad \text{ve} \qquad \mathbf{Q}^T\mathbf{y} = \begin{array}{cc} \begin{pmatrix} \mathbf{y}_1 \\ \mathbf{y}_2 \end{pmatrix} & \begin{array}{c} n \\ m-n \end{array} \end{array} \tag{3.6}$$

ki burada $\mathbf{Q} \in \mathbb{R}^{m\times m}$ ortogonal matris ($\mathbf{Q}^T\mathbf{Q} = \mathbf{Q}\mathbf{Q}^T = \mathbf{I}_m$) ve $\mathbf{R} \in \mathbb{R}^{n\times n}$ üst üçgen matristir. (3.5)'in sıradan en küçük kareler tahmincisi, $\left\|\boldsymbol{\varepsilon}^2\right\| = \left\|\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\right\|^2$ ifadesinin minimizasyonundan elde edilir. Şöyle ki;

$$\begin{aligned} \hat{\boldsymbol{\beta}} &= \arg\min_{\boldsymbol{\beta}} \left\|\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\right\|^2 \\ &= \arg\min_{\boldsymbol{\beta}} \left\|\mathbf{Q}^T\mathbf{y} - \mathbf{Q}^T\mathbf{X}\boldsymbol{\beta}\right\|^2 \\ &= \arg\min_{\boldsymbol{\beta}} \left(\left\|\mathbf{y}_1 - \mathbf{R}\boldsymbol{\beta}\right\|^2 + \left\|\mathbf{y}_2\right\|^2\right) \\ &= \mathbf{R}^{-1}\mathbf{y}_1 \end{aligned}$$

hata kareler toplamı $\left\|\mathbf{y}_2\right\|^2 = \left\|\mathbf{y} - \mathbf{X}\hat{\boldsymbol{\beta}}\right\|^2$ şeklinde elde edilir ve σ^2'nin tahmincisi ise $\hat{\sigma}^2 = \left\|\mathbf{y}_2\right\|^2 / (m-n)$ şeklinde hesaplanır.

3.2. Genel Doğrusal Model

Genel Doğrusal Model ile Sıradan Doğrusal Model arasındaki fark ε_i $(i = 1,\ldots,m)$ hata değerleri arasındaki korelasyondur. (3.4)'de verilen genel doğrusal model için $\boldsymbol{\beta}$'nın En İyi Doğrusal Sapmasız Tahmincisi Genelleştirilmiş En Küçük Kareler probleminin çözümünden elde edilir. Şöyle ki;

$$\arg\min_{\boldsymbol{\beta}} \left\|\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\right\|^2_{\boldsymbol{\Omega}^{-1}}$$

Burada $\|.\|_{\Omega}$ enerji normunu ifade eder (örneğin $\|\mathbf{v}\|_{\mathbf{\Omega}}^2 = \mathbf{v}^T\mathbf{\Omega}\mathbf{v}$'dir). Buradan elde edilecek normal denklemler;

$$\mathbf{X}^T\mathbf{\Omega}^{-1}\mathbf{X}\boldsymbol{\beta} = \mathbf{X}^T\mathbf{\Omega}^{-1}\mathbf{y}$$

şeklindedir ve sonuçta $\boldsymbol{\beta}$'nın tahmini;

$$\hat{\boldsymbol{\beta}} = \left(\mathbf{X}^T\mathbf{\Omega}^{-1}\mathbf{X}\right)^{-1}\mathbf{X}^T\mathbf{\Omega}^{-1}\mathbf{y}$$

olarak elde edilir.

Bu çözüm hesaplama açısından oldukça zordur ve $\mathbf{\Omega}$ kötü-koşullandırılmış olduğunda nümerik açıdan stabil değildir (Björk, 1996; Lawson ve Hanson, 1974). Bunun yanı sıra $\mathbf{\Omega}$ tekil olduğunda genel doğrusal modelin çözümü yapılamaz ve bu durumda $\mathbf{\Omega}^{-1}$ yerine Moore-Penrose genelleştirilmiş tersinin kullanılması gerekir ki bu da her zaman $\boldsymbol{\beta}$'nın en iyi doğrusal sapmasız tahmincisini vermeyecektir (Kourouklis ve Paige, 1981).

$\mathbf{\Omega}$'nın kötü-koşullandırılması veya tekilliği ile ilgili problemlerden kaçınmak için (3.4) modeli Genelleştirilmiş Doğrusal En Küçük Kareler Problemi olarak şu şekilde formüle edilebilir:

$$\underset{\mathbf{v},\boldsymbol{\beta}}{\arg\min}\, \mathbf{v}^T\mathbf{v} \qquad , \qquad \mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \mathbf{C}\mathbf{v} \quad \text{'ye göre} \tag{3.7}$$

Burada $\mathbf{\Omega} \in \mathbb{R}^{m\times m}$ g ranklı ve yarı pozitif tanımlı matris, $\mathbf{\Omega} = \mathbf{C}\mathbf{C}^T$ olmak kaydıyla $\mathbf{C} \in \mathbb{R}^{m\times g}$ tam sütun ranklı matris ve rassal g elemanlı hata vektörü $\mathbf{C}\mathbf{v} = \boldsymbol{\varepsilon} \;\Rightarrow\; \mathbf{v} = \mathbf{C}^{-1}\boldsymbol{\varepsilon}$ olarak tanımlanır. Buna göre $\mathbf{v} \sim \left(0, \sigma^2\mathbf{I}\right)$'dır (Kourouklis ve Paige, 1981)0. Her ne kadar (3.7) eşitliği tekil $\mathbf{\Omega}$ matrisi için de geçerli olsa da genelliğin kaybı haricinde $\mathbf{\Omega}$'nın tekil olmadığı durum dikkate alınır. genelleştirilmiş doğrusal en küçük kareler problemi (3.7)'nin çözümü için Genelleştirilmiş QR ayrışımı kullanılabilir (Anderson ve Ark., 1992; Paige, 1990). $\mathbf{X}$ ve $\mathbf{C}$ matrislerinin genelleştirilmiş QR ayrışımı için öncelikle (2.6)'da verilen QR ayrışımı ve daha sonra da $\mathbf{Q}^T\mathbf{C}$'nin RQ ayrışımı hesaplanmalıdır. $\mathbf{Q}^T\mathbf{C}$'nin RQ ayrışımı;

$$\left(\mathbf{Q}^T\mathbf{C}\right)\mathbf{P} = \mathbf{W} \equiv \begin{array}{cc} \begin{matrix} m & m-n \end{matrix} & \\ \begin{pmatrix} \mathbf{W}_{11} & \mathbf{W}_{12} \\ \mathbf{0} & \mathbf{W}_{22} \end{pmatrix} & \begin{matrix} n \\ m-n \end{matrix} \end{array} \qquad \text{ve} \qquad \mathbf{P}^T\mathbf{v} = \begin{pmatrix} \mathbf{v}_1 \\ \mathbf{v}_2 \end{pmatrix} \begin{matrix} n \\ m-n \end{matrix} \tag{3.8}$$

şeklinde hesaplanır ki burada $\mathbf{P} \in \mathbb{R}^{m\times m}$ ortogonal matris ve $\mathbf{W} \in \mathbb{R}^{m\times m}$ üst üçgen ve tekil olmayan matristir. Buna göre genelleştirilmiş doğrusal en küçük kareler problemi (3.7) şu şekilde ifade edilebilir:

$$\underset{\mathbf{v},\boldsymbol{\beta}}{\arg\min} \left\| \mathbf{P}^T \mathbf{v} \right\|^2 \quad , \quad \mathbf{Q}^T \mathbf{y} = \mathbf{Q}^T \mathbf{X}\boldsymbol{\beta} + \mathbf{Q}^T \mathbf{C}\mathbf{P}^T \mathbf{P}\mathbf{v}$$

veya

$$\underset{\mathbf{v}_1,\mathbf{v}_2,\boldsymbol{\beta}}{\arg\min} \left(\left\| \mathbf{v}_1 \right\|^2 + \left\| \mathbf{v}_2 \right\|^2 \right) \quad , \quad \begin{cases} \mathbf{y}_1 = \mathbf{R}\boldsymbol{\beta} + \mathbf{W}_{11}\mathbf{v}_1 + \mathbf{W}_{12}\mathbf{v}_2 \\ \mathbf{y}_2 = \mathbf{W}_{22}\mathbf{v}_2 \end{cases} \tag{3.9}$$

Amaç fonksiyonunu minimize etmek için (3.9)'daki ikinci kısıttan $\mathbf{v}_2 = \mathbf{W}_{22}^{-1}\mathbf{y}_2$ alt vektörünün değeri hesaplanır ve elde edilen vektör ilk kısıtta yerine yazılır. Buradaki $\mathbf{v}_1$ alt vektörü sıfıra eşitlenerek geride kalan $\mathbf{R}\boldsymbol{\beta} = \mathbf{y}_1 - \mathbf{W}_{12}\mathbf{v}_2$ üçgen sisteminin çözümü bize $\boldsymbol{\beta}$ 'nın tahmincisini verir. Katsayıların varyans-kovaryans matrisi ise $\hat{\sigma}^2 \mathbf{R}^{-T} \mathbf{W}_{11}^T \mathbf{W}_{11} \mathbf{R}^{-1}$ şeklinde olacaktır ki burada σ^2 'nin tahmini olan $\hat{\sigma}^2 = \left\| \mathbf{v}_2 \right\|^2 / (m-n)$ şeklinde hesaplanır.

3.3. Görünüşte İlişkisiz Regresyon Modeli

Görünüşte İlişkisiz Regresyon modeli, genel doğrusal modelin özel bir durumudur ve G tane regresyon denkleminin kümesi olarak şu şekilde tanımlanabilir:

$$\mathbf{y}_i = \mathbf{X}_i \boldsymbol{\beta}_i + \mathbf{u}_i \quad , \quad i = 1,\ldots,G$$

Burada $\mathbf{y}_i \in \mathbb{R}^M$ bağımlı değişken vektörleri, $\mathbf{X}_i \in \mathbb{R}^{M\times k_i}$ tam sütun ranklı bağımsız değişken matrisleri, $\boldsymbol{\beta}_i \in \mathbb{R}^{k_i}$ bilinmeyen katsayılar vektörleri ve $\mathbf{u}_i \in \mathbb{R}^M$ hata vektörleridir. $\mathbf{u}_i$ 'nin beklenen değeri sıfırdır yani $E(\mathbf{u}_i) = 0$ ve varyans-kovaryans matrisi $\sigma_{i,i}\mathbf{I}_M$ 'dir. Aynı zamanda modelde yer alan G tane regresyon denkleminin artık vektörleri de birbirleriyle ilişkilidir, yani $E\left(u_i u_j^T\right) = \sigma_{i,j} I_M \quad (i,j = 1,\ldots,G)$ 'dir (Kontoghiorghes, 2000b; Srivastava ve Giles, 1987). Görünüşte ilişkisiz regresyon modeli, açık olarak;

$$\begin{pmatrix} \mathbf{y}_1 \\ \mathbf{y}_2 \\ \vdots \\ \mathbf{y}_G \end{pmatrix} = \begin{pmatrix} \mathbf{X}_1 & & & \\ & \mathbf{X}_2 & & \\ & & \ddots & \\ & & & \mathbf{X}_G \end{pmatrix} \begin{pmatrix} \boldsymbol{\beta}_1 \\ \boldsymbol{\beta}_2 \\ \vdots \\ \boldsymbol{\beta}_G \end{pmatrix} + \begin{pmatrix} \mathbf{u}_1 \\ \mathbf{u}_2 \\ \vdots \\ \mathbf{u}_G \end{pmatrix}$$

şeklinde ifade edilirken, aynı model kapalı formda

$$Vec(\mathbf{Y}) = \left(\oplus_{i=1}^{G} \mathbf{X}_i\right) Vec\left(\{\boldsymbol{\beta}_i\}_G\right) + Vec(\mathbf{U}) \tag{3.10}$$

şeklinde ifade edilir. Burada $\mathbf{Y} = (\mathbf{y}_1 \ldots \mathbf{y}_G) \in \mathbb{R}^{MxG}$ şeklinde sütunları her bir denklemin bağımlı değişken vektörlerinden oluşan ve $\mathbf{U} = (\mathbf{u}_1 \ldots \mathbf{u}_G) \in \mathbb{R}^{MxG}$ şeklinde sütunları her bir denklemin hata vektörlerinden oluşan $M \times G$ boyutlu matrislerdir. $\{\boldsymbol{\beta}_i\}_G$ ifadesi $\boldsymbol{\beta}_1, \ldots, \boldsymbol{\beta}_G$ vektörlerinin kümesini ifade etmektedir. $\oplus_{i=1}^{G} \mathbf{X}_i$ ifadesi bağımsız değişkenler matrisleri olarak ifade edilen $\mathbf{X}_i$'lerin direkt toplamını ifade eder ki bu da $K = \sum_{i=1}^{G} k_i$ olmak üzere $GM \times K$ boyutlu blok köşegen matristir (Regalia ve Mitra, 1989; Harville, 1997). Şöyle ki;

$$\oplus_{i=1}^{G} \mathbf{X}_i = \mathbf{X}_1 \oplus \mathbf{X}_2 \oplus \ldots \oplus \mathbf{X}_G = \begin{pmatrix} \mathbf{X}_1 & & & \\ & \mathbf{X}_2 & & \\ & & \ddots & \\ & & & \mathbf{X}_G \end{pmatrix}$$

Vec operatörü bir matrisin sütunlarını sırasıyla alt alta yığmak için veya vektörler kümesini tek bir sütun vektörüne dönüştürmek için kullanılan bir operatördür (Ayrıntılı bilgi için Bölüm 2.2'ye bakınız). (3.10) denklemindeki hata terimi $Vec(\mathbf{U})$'nun ortalaması sıfırdır ve varyans-kovaryans matrisi $\boldsymbol{\Sigma} \otimes \mathbf{I}_M$'dir ki burada $\boldsymbol{\Sigma} = [\sigma_{ij}] \in \mathbb{R}^{G \times G}$ simetrik ve negatif tanımlı olmayan bir matristir. Hata teriminin varyans-kovaryans matrisinde yer alan $\otimes$ işlem ifadesi kronecker çarpımını ifade eder (Ayrıntılı bilgi için Bölüm 2.1'e bakınız). Buna göre $vec(\mathbf{U}) \sim (\mathbf{0}, \boldsymbol{\Sigma} \otimes \mathbf{I}_M)$ ve

$$\boldsymbol{\Sigma} \otimes \mathbf{I}_M = \begin{pmatrix} \sigma_{11} \otimes \mathbf{I}_M & \sigma_{12} \otimes \mathbf{I}_M & \cdots & \sigma_{1G} \otimes \mathbf{I}_M \\ \sigma_{21} \otimes \mathbf{I}_M & \sigma_{22} \otimes \mathbf{I}_M & \cdots & \sigma_{2G} \otimes \mathbf{I}_M \\ \vdots & \vdots & \ddots & \vdots \\ \sigma_{G1} \otimes \mathbf{I}_M & \sigma_{G2} \otimes \mathbf{I}_M & \cdots & \sigma_{GG} \otimes \mathbf{I}_M \end{pmatrix}$$

dir (Kmenta, 1986; Schmidt, 1978; Srivastava ve Dwivedi, 1979; Srivastava ve Giles, 1987; Zellner, 1962; Zellner, 1963).

Ekonometrik araştırmalarda sıkça karşılaşılan ve görünüşte ilişkisiz regresyon modellerinin çözümünde de karşılaşılabilecek özel durumları kısaca aşağıdaki başlıklar halinde vermek mümkündür:

- ***Farklı Varyanslı Artıklar*** : Bu durum, küresel bir dağılım gösteren $\mathbf{u}_i$ varsayımının bozulması olarak nitelendirilebilir. Buna göre $\mathbf{u}_i$ ve $\mathbf{u}_j$'nin kovaryans matrisi $E\left(\mathbf{u}_i\mathbf{u}_j^T\right)=\sigma_{i,j}\mathbf{D}$ olarak kabul edilir ki burada $\mathbf{D}$, pozitif elemanlara sahip köşegen bir matristir. Böylece, $Vec(\mathbf{u})$'nun varyans-kovaryans matrisi $\mathbf{\Sigma}\otimes\mathbf{D}$ şeklinde olacaktır.

- ***Korelasyon Kısıtlamaları*** : Bu tarz modellerde bir denkleme ilişkin artığın varyansı bir sonraki denklemin artık varyansından daha küçük olacak şekilde varyans-kovaryans matrisinin elemanları sınırlandırılmıştır. Buna ilaveten artıklar arasındaki korelasyon da sıfır ile bir arasındadır. Yani,

$$\sigma_{1,1}\leq\sigma_{2,2}\leq\ldots\leq\sigma_{G,G}$$

ve

$$0<\rho_{i,j}<1\,,\qquad i,j=1,\ldots,G\quad \text{ve}\quad i\neq j$$

şeklindedir. Burada $\rho_{i,j}=\dfrac{\sigma_{i,j}}{\sqrt{\sigma_{i,i}\sigma_{j,j}}}$, $u_{i,t}$ ile $u_{j,t}$ arasındaki korelasyondur. Bu, *Hata Bileşenleri* Modellerindeki uygulamalara uygun bir tanımlamadır. Buna göre artıklar;

$$\mathbf{u}_i=\sum_{j=1}^{i}\boldsymbol{\varepsilon}_j\quad,\qquad \boldsymbol{\varepsilon}_i\sim\left(\mathbf{0},\mathbf{v}_i\mathbf{I}_T\right)$$

ve

$$E\left(\boldsymbol{\varepsilon}_i\boldsymbol{\varepsilon}_j^T\right)=0\,,\qquad i\neq j \text{ için.}$$

olarak tanımlanır (Kontoghiorghes, 2000c; Srivastava ve Giles, 1987; Zellner, 1979).

- ***Otoregresif Artıklar*** : Görünüşte ilişkisiz regresyon modellerinde artıkların zamana göre ilişkisiz oldukları varsayılır. Ancak birçok uygulama için bu durum oldukça sınırlayıcıdır ve zamana göre korelasyonun incelenmesi gerekmektedir. Otoregresif artıklar ile görünüşte ilişkisiz regresyon modelleri için hatalar AR süreci ile oluşturulur.

$$u_{i,t} = \alpha_i u_{i,t-1} + \varepsilon_{i,t}, \qquad i = 1,\ldots,G, \qquad t = 2,\ldots,T$$

Burada α_i AR parametreleri, $\varepsilon_{i,t} \sim \left(0, \sigma_{i,i}\right)$, $E\left(\varepsilon_{i,t}\varepsilon_{j,t}\right) = \sigma_{i,j}$ ve $E\left(\varepsilon_{i,s}\varepsilon_{j,t}\right) = 0$'dir, $s \neq t$ ve $i, j = 1,\ldots,G$ için (Foschi ve Kontoghiorghes, 2003; Kmenta ve Gilbert, 1970; Turkington, 2000). Daha genel bir ifadeyle,

$$u_i = \alpha_i \mathbf{Z} u_i + \varepsilon_i, \quad i = 1,\ldots,G$$

veya

$$Vec(\mathbf{U}) = \left(\oplus_i \alpha_i \mathbf{Z}\right) Vec(\mathbf{U}) + Vec(\mathbf{E}), \quad Vec(\mathbf{E}) \sim \left(0, \mathbf{\Sigma} \otimes \mathbf{I}_T\right)$$

şeklinde ifade edilir. Burada $\mathbf{E} = \left(\varepsilon_1 \ldots \varepsilon_G\right)$, $\varepsilon_i = \left(\varepsilon_{i,1} \ldots \varepsilon_{i,T}\right)^T$ ve

$$\mathbf{Z} = \begin{pmatrix} 0 & 0 & \cdots & 0 & 0 \\ 1 & 0 & \cdots & 0 & 0 \\ 0 & 1 & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & 1 & 0 \end{pmatrix}$$

$T \times T$ boyutlu kaydırma (shift) matrisidir. Bu modelde $Vec(\mathbf{U})$'nun varyans-kovaryans matrisi ise;

$$\left(\oplus_{i=1}^{G}\left(\mathbf{I}_T - \alpha_i \mathbf{Z}\right)\right)^{-1}\left(\mathbf{\Sigma} \otimes \mathbf{I}_T\right)\left(\oplus_{i=1}^{G}\left(\mathbf{I}_T - \alpha_i \mathbf{Z}\right)\right)^{-T}$$

olarak hesaplanır.

- ***Vektör Otoregresif Artıklar*** : Otokorelasyonun daha genel formu Vektör Otoregresif (VAR) süreci ile şu şekilde tanımlanır:

$$\mathbf{U} = \mathbf{ZUA}^T + \mathbf{E}, \qquad Vec(\mathbf{E}) \sim \left(\mathbf{0}, \mathbf{\Sigma} \otimes \mathbf{I}_T\right)$$

Burada $\mathbf{A} \in \mathbb{R}^{G \times G}$, AR parametreleri matrisidir. Buna göre, görünüşte ilişkisiz regresyon artıkları için kovaryans matrisi ise;

$$\left(\mathbf{I}_{GT} - \mathbf{A} \otimes \mathbf{Z}\right)^{-1} \left(\mathbf{\Sigma} \otimes \mathbf{I}_T\right)\left(\mathbf{I}_{GT} - \mathbf{A} \otimes \mathbf{Z}\right)^{-T}$$

olur. İlk gözlemin artıklarına ilişkin ek varsayım, $u_{i,t}$'nin $\left(i = 1, \ldots, G\right)$ açık bir şekilde belirlenmiş ya da belirlenebiliyor olmasıdır. Bazı kaynaklarda ilk gecikme için de bu belirlemenin yapılmış olmasının gerekli olduğu belirtilmektedir. VAR(p) süreci için artıklar;

$$\mathbf{U} = \mathbf{Z}\mathbf{U}\mathbf{A}_1^T + \mathbf{Z}^2\mathbf{U}\mathbf{A}_2^T + \ldots + \mathbf{Z}^p\mathbf{U}\mathbf{A}_p^T + \mathbf{E} \ , \ \ Vec\left(\mathbf{E}\right) \sim \left(\mathbf{0}, \mathbf{\Sigma} \otimes \mathbf{I}_T\right)$$

şeklinde olur ki burada $\mathbf{A}_1, \mathbf{A}_2, \ldots, \mathbf{A}_p \in \mathbb{R}^{G \times G}$, AR süreci parametre matrisleridir (Foschi ve Kontoghiorghes, 2003).

- ***<u>Gözlem Sayılarının Eşit Olmaması</u>*** : Görünüşte ilişkisiz regresyon modellerinde her bir regresyon modelindeki gözlem sayılarının eşit olduğu varsayılır. Ancak birçok durumda bunun sağlanması pek mümkün olmamaktadır (Schmidt, 1977; Srivastava ve Giles, 1987; Sharma, 1993). Böyle bir durumda i inci denklem için t_i tane gözlem değeri olduğunu ve $\left(i+1\right)$ inci denklemdeki ilk t_i adet gözlemin de i inci denklemin gözlemleri ile aynı anda gözlemlendiğini varsayalım. Buna göre $j > i$ olmak üzere u_i ve u_j artıkları için kovaryans matrisi;

$$E\left(\mathbf{u}_i\mathbf{u}_j^T\right) = \sigma_{i,j}\begin{pmatrix}\mathbf{I}_{t_i} & \mathbf{0}_{t_i \times \left(t_j - t_i\right)}\end{pmatrix}$$

olur.

- ***<u>Eksik Gözlem</u>*** : Gözlem sayıları eşit olmayan görünüşte ilişkisiz regresyon modelinin genellenmesi sonucunda elde edilecek model eksik gözlemli görünüşte ilişkisiz model olacaktır. Eldeki verinin gidişatına uyan eksik gözlemler belirlenemediğinde ve i inci denklemdeki t_i gözlem değeri ile $\left(i+1\right)$ inci denklemdeki ilk t_i adet gözlem değeri aynı anda elde edilemediğinde kovaryans matrisi;

$$E\left(\mathbf{u}_i\mathbf{u}_j^T\right) = \sigma_{i,j}\mathbf{S}_{i,j}$$

olarak elde edilir. Burada $\mathbf{S}_{i,j}$, sırasıyla i inci ve j inci denklemlerin aynı anda gözlemlenen k ıncı ve l inci gözlemleri için (k,l) elemanı sıfır olmayan $t_i \times t_j$ boyutlu sıfır-bir matrisidir.

♦ ***<u>Ortak Bağımsız Değişkenler</u>*** : Görünüşte ilişkisiz regresyon modelinde ortak bağımsız değişkenler söz konusu olduğunda bağımsız değişkenler matrisi;

$$\mathbf{X}_i = \mathbf{X}^d \mathbf{S}_i, \qquad i = 1, \ldots, G$$

olarak ele alınır. Burada $\mathbf{X}^d \in \mathbb{R}^{T \times K^d}$, K^d farklı açıklayıcı değişkenden oluşan matris ve $\mathbf{S}_i \in \mathbb{R}^{K^d \times k_i}$ ise $K^d \times K^d$ boyutlu birim matrisin ilgili sütunlarından oluşan seçim matrisidir.

♦ ***<u>Üçgensel GİR Modeller</u>*** : Üçgensel görünüşte ilişkisiz regresyon modelleri ortak bağımsız değişkenli modellerin özel bir durumudur. Bu modellerde, denklemler ve her bir bağımsız değişkenler matrisinin sütunları, i inci bağımsız değişken bir sonraki bağımsız değişkenin sütunlarının bir alt kümesi olacak şekilde yenide düzenlenebilir. Yani, $\mathbf{X}_i = \begin{pmatrix} \mathbf{X}_{i-1} & \tilde{\mathbf{X}}_i \end{pmatrix}$ şeklinde düzenlenebilir ki burada $\tilde{\mathbf{X}}_i \in \mathbb{R}^{T \times k_i - k_{i-1}}$'dir.

4. GÖRÜNÜŞTE İLİŞKİSİZ REGRESYON MODELİNİN TAHMİNİ VE MATLAB ALGORİTMALARI

Görünüşte ilişkisiz regresyon modeli, Bölüm 3.3'de genel olarak tanımlanmıştı. Bu bölümde görünüşte ilişkisiz regresyon modellerinin çözümlenmesinde kullanılan çeşitli algoritmalar ve bu algoritmaların hesaplama etkinlikleri verilmiştir. Bu bölümde verilen tüm algoritmalar, QR ayrışımı kullanılarak elde edilmiş ortogonal faktörizasyon temeline dayanır. Bu sayede daha etkin ve hızlı algoritmaların oluşturulması hedeflenmiştir. Algoritmaların kodlanmasında ve işletilmesinde MathWorks firması tarafından geliştirilmiş olan MATLAB paketinin 7.3.0.267(R2006b) sürümü kullanılmıştır.

Görünüşte ilişkisiz regresyon modellerinin çözümleme sürecinde, Kronecker çarpımı ve Vec operatörüne ilişkin Bölüm 2.1 ve Bölüm 2.2'de ispatları ile birlikte verilen özeliklerden yararlanılacaktır.

(3.10) eşitliği ile verilen görünüşte ilişkisiz regresyon modelini ele alalım. Bu model, $\mathbf{\Sigma}$ tekil olmadığında genelleştirilmiş (doğrusal) en küçük kareler problemi şeklinde ifade edilebilir. Buna göre $\boldsymbol{\beta}$'nın en iyi doğrusal sapmasız tahmincisi;

$$\underset{\boldsymbol{\beta}_1,\ldots,\boldsymbol{\beta}_G}{\arg\min} \left\| Vec(\mathbf{Y}) - Vec(\{\mathbf{X}_i\boldsymbol{\beta}_i\}) \right\|_{\mathbf{\Sigma}^{-1}\otimes\mathbf{I}_M} \tag{4.1}$$

minimizasyonundan elde edilecek

$$\left(\left(\oplus_i \mathbf{X}_i^T\right)\left(\mathbf{\Sigma}^{-1}\otimes\mathbf{I}_M\right)\left(\oplus_i \mathbf{X}_i\right)\right)\hat{\boldsymbol{\beta}} = \left(\oplus_i \mathbf{X}_i^T\right) Vec\left(\mathbf{Y}\mathbf{\Sigma}^{-1}\right) \tag{4.2}$$

normal denklemlerinden elde edilir. Ancak bu çözüm matrisler kötü koşullandırılmış (ill-conditioned) olduğunda ve çözüm sürecinde kesin matris tersi kullanıldığında tutarsız olabilir (Björk, 1994; Lawson ve Hanson, 1974). Bu nedenle alternatif olarak, (3.10) eşitliğinin her iki tarafının da $\left(\mathbf{C}^{-1}\otimes\mathbf{I}_M\right)$ ile çarpımı bize sıradan doğrusal modeli verecektir. Yani;

$$Vec\left(\mathbf{Y}\mathbf{C}^{-T}\right) = \left(\mathbf{C}^{-1}\otimes\mathbf{I}_M\right)\left(\oplus_{i=1}^{G}\mathbf{X}_i\right)\boldsymbol{\beta} + Vec\left(\mathbf{U}\mathbf{C}^{-T}\right) \tag{4.3}$$

olacaktır. Burada $\mathbf{C}\in\mathbb{R}^{GxG}$, $\mathbf{\Sigma}\equiv\mathbf{C}\mathbf{C}^T$ olacak biçimde Cholesky ayrışımıdır ve üst üçgendir. Böylece sıradan doğrusal model haline getirilen ve (4.3) eşitliği ile ifade edilen görünüşte ilişkisiz regresyon modelinin en küçük kareler tahmini, (3.10) eşitliği ile tanımlanan görünüşte ilişkisiz regresyon modelinin en iyi sapmasız tahmincisini verir (Pollock, 1979).

Görünüşte ilişkisiz regresyon modeli genelleştirilmiş doğrusal en küçük kareler problemi olarak da şu şekilde ele alınabilir:

$$\arg\min \|\mathbf{V}\|_F^2 \quad , \quad Vec(\mathbf{Y}) = \left(\oplus_i \mathbf{X}_i\right)\boldsymbol{\beta} + \left(\mathbf{C} \otimes \mathbf{I}_M\right) Vec\left(\mathbf{V}\right) \text{'ye göre} \qquad (4.4)$$

ki burada $\mathbf{VC}^T = \mathbf{U}$, $Vec(\mathbf{V}) \sim \left(\mathbf{0}, \mathbf{I}_{GM}\right)$'dir ve $\|.\|_F$, Frobenius normudur (Golub ve Van Loan, 1996; Kontoghiorghes, 2000b). Bu yaklaşım, (4.3) modelini temel alan algoritmalara göre nümerik açıdan daha durağan algoritmalar geliştirilmesini de mümkün kılar. Buna ilaveten genelleştirilmiş doğrusal en küçük kareler problemi, C tam ranklı olmasa bile yani $\mathbf{\Sigma}$ tekil olsa dahi (3.10)'un en iyi doğrusal sapmasız tahmincisinin elde edilmesini mümkün kılar (Kourouklis ve Paige, 1981; Paige, 1990; Söderkvist, 1996).

Çözümlemede genellikle $\mathbf{\Sigma}$ bilinmez ve adımsal süreç ile uygun (feasible) genelleştirilmiş en küçük kareler tahmincisi elde edilir. Bu tahmin yapılırken $\mathbf{\Sigma}$'nın her bir elemanı;

$$\hat{\sigma}_{ij} = s_{ij} = \frac{\mathbf{e}_i^T \mathbf{e}_j}{M}$$

şeklinde elde edilir. Uygun genelleştirilmiş en küçük kareler tahmincisi elde edilirken nadiren de olsa serbestlik derecesi düzeltmesi yapılması gerekebilir. Bu düzeltme ise;

$$s_{ij}^{*} = \frac{\mathbf{e}_i^T \mathbf{e}_j}{\sqrt{\left(M - k_i\right)\left(M - k_j\right)}}$$

veya

$$s_{ij}^{**} = \frac{\mathbf{e}_i^T \mathbf{e}_j}{M - \max\left(k_i, k_j\right)}$$

şeklinde yapılır. Serbestlik derecesi düzeltmesi yapılmadan elde edilecek s_{ij} tahmini veya düzeltme yapılarak elde edilecek s_{ij}^{*} tahmini sadece $i = j$ olması halinde yansız bir tahmin olurken, s_{ij}^{**} tahmini ancak ve ancak $k_i = k_j$ olduğunda yansız olmaktadır (Judge ve Ark., 1985; Theil, 1971).

$\boldsymbol{\Sigma}$'nın elde edilen tutarlı bir tahmincisi kullanılarak (4.2) denkleminin çözümünden β'nın bir tahmincisi elde edilir. Tahminlenen bu katsayıların artıklarından Σ'nın bir diğer tahmincisi elde edilir ve bu süreç her iki tahmin arasındaki fark önemsenmeyecek kadar küçük olana kadar tekrarlanır. Böylece (4.2) genelleştirilmiş en küçük kareler problemi veya ilgili olarak (4.4) genelleştirilmiş doğrusal en küçük kareler problemi farklı $\boldsymbol{\Sigma}$ ile defalarca çözümlenir. Burada daha ilk adımda β'nın tahmincisinin türetilmesinin hesaplama açısından maliyeti dikkat çekmektedir.

4.1. QR Ayrışımı ile Sıradan Doğrusal Model Tahmini

(4.3) eşitliği ile verilen sıradan doğrusal modeli;

$$\underbrace{Vec\left(\mathbf{YC}^{-T}\right)}_{\overline{\mathbf{y}}} = \underbrace{\left(\mathbf{C}^{-1} \otimes \mathbf{I}_M\right)\left(\oplus_{i=1}^{G} \mathbf{X}_i\right)}_{\overline{\mathbf{X}}}\boldsymbol{\beta} + \underbrace{Vec\left(\mathbf{UC}^{-T}\right)}_{\overline{\boldsymbol{\varepsilon}}} \tag{4.5}$$

$\overline{\mathbf{y}} = \overline{\mathbf{X}}\boldsymbol{\beta} + \overline{\boldsymbol{\varepsilon}}$ şeklinde tekrar düzenleyelim. $\overline{\mathbf{X}}$'nın QR ayrışımı;

$$\overline{\mathbf{Q}}^T \overline{\mathbf{X}} = \begin{pmatrix} \overline{\mathbf{R}} \\ \mathbf{0} \end{pmatrix} \begin{matrix} K \\ GM-K \end{matrix} \qquad ve \qquad \overline{\mathbf{Q}}^T \overline{\mathbf{y}} = \begin{pmatrix} \overline{\mathbf{y}}_A \\ \overline{\mathbf{y}}_B \end{pmatrix} \begin{matrix} K \\ GM-K \end{matrix}$$

olarak elde edilir. $\overline{\mathbf{R}} \in \mathbb{R}^{KxGM}$ üst üçgen, $\overline{\mathbf{Q}} \in \mathbb{R}^{GMxGM}$ ortogonaldir ve $K = \sum_{i=1}^{G} k_i$'dir. $\boldsymbol{\beta}$'nın en küçük kareler tahmincisi,

$$\hat{\boldsymbol{\beta}} = \overline{\mathbf{R}}^{-1}\overline{\mathbf{y}}_A \tag{4.6}$$

olarak elde edilebilir.

Bu şekilde (4.5) eşitliğinin doğrudan çözümü elde edilebilmesine karşın, bağımsız değişkenler matrisleri olan $\mathbf{X}_i$'lerin boyutlarının büyük olması nedeniyle çözümün elde edilemeyebilmesi ya da elde edilen çözümün hesaplama açısından etkin olmamasından dolayı ve blok matris özelliğinden faydalanarak her bir matris için QR ayrışımını ayrı ayrı ele almak gerekir. Bunları göz önüne alarak daha etkin bir çözüm için her bir $\mathbf{X}_i$'nin QR ayrışımını ele alalım:

$$\mathbf{Q}^T\mathbf{X}_i = \begin{pmatrix} \mathbf{R}_i \\ \mathbf{0} \end{pmatrix} \quad , \quad \mathbf{Q}_i^T = \begin{array}{c} \begin{pmatrix} \tilde{\mathbf{Q}}_i^T \\ \hat{\mathbf{Q}}_i^T \end{pmatrix} \end{array} \begin{array}{l} k_i \\ M-k_i \end{array} \tag{4.7}$$

$\mathbf{R}_i \in \mathbb{R}^{k_i x k_i}$ üst üçgen, $\mathbf{Q}_i \in \mathbb{R}^{MxM}$ ortogonaldir. Buradan $\oplus_i \mathbf{X}_i$ 'nin QR ayrışımı;

$$\mathbf{Q}^T\left(\oplus_i \mathbf{X}_i\right) = \begin{pmatrix} \oplus_i \mathbf{R}_i \\ \mathbf{0} \end{pmatrix} \begin{array}{l} K \\ GM-K \end{array} \quad , \quad \mathbf{Q} = \begin{pmatrix} \oplus_i \tilde{\mathbf{Q}}_i \\ \oplus_i \hat{\mathbf{Q}}_i \end{pmatrix} \tag{4.8}$$

olarak elde edilir. (4.5) eşitliğinin $\mathbf{Q}^T$ ile soldan çarpımını ele alalım.

$$\mathbf{Q}^T Vec\left(\mathbf{YC}^{-T}\right) = \mathbf{Q}^T\left(\mathbf{C}^{-1}\otimes\mathbf{I}_M\right)\left(\oplus_{i=1}^G \mathbf{X}_i\right)\boldsymbol{\beta} + \mathbf{Q}^T Vec\left(\mathbf{UC}^{-T}\right)$$

veya

$$\begin{pmatrix} Vec\left(\left\{\tilde{\tilde{\mathbf{y}}}_i\right\}\right) \\ Vec\left(\left\{\hat{\hat{\mathbf{y}}}_i\right\}\right) \end{pmatrix} = \begin{pmatrix} \tilde{\tilde{\mathbf{W}}} \\ \hat{\hat{\mathbf{W}}} \end{pmatrix}\boldsymbol{\beta} + \begin{pmatrix} Vec\left(\tilde{\tilde{\mathbf{V}}}\right) \\ Vec\left(\hat{\hat{\mathbf{V}}}\right) \end{pmatrix} \tag{4.9}$$

şeklinde yazabiliriz. Burada;

$$\begin{array}{cc} & \begin{array}{ccc} k_1 & \dots & k_G \end{array} \\ \tilde{\tilde{\mathbf{W}}} = & \begin{pmatrix} \tilde{\tilde{\mathbf{W}}}_{1,1} & \dots & \tilde{\tilde{\mathbf{W}}}_{1,G} \\ \vdots & \ddots & \vdots \\ \tilde{\tilde{\mathbf{W}}}_{G,1} & \dots & \tilde{\tilde{\mathbf{W}}}_{G,G} \end{pmatrix} \begin{array}{c} k_1 \\ \vdots \\ k_G \end{array} \end{array}$$

şeklinde blok üst üçgen bir matristir ve elemanları;

$$\tilde{\tilde{\mathbf{W}}}_{i,j} = \begin{cases} \gamma_{i,j}\tilde{\mathbf{Q}}_i^T\mathbf{X}_j & , \quad i<j \\ \gamma_{i,j}\mathbf{R}_i & , \quad i=j \\ \mathbf{0} & , \quad i>j \end{cases} \tag{4.10}$$

olarak hesaplanır. Benzer şekilde;

$$\begin{array}{cc} & \begin{array}{ccc} M-k_1 & \dots & M-k_G \end{array} \\ \hat{\hat{\mathbf{W}}} = & \begin{pmatrix} \hat{\hat{\mathbf{W}}}_{1,1} & \dots & \hat{\hat{\mathbf{W}}}_{1,G} \\ \vdots & \ddots & \vdots \\ \hat{\hat{\mathbf{W}}}_{G,1} & \dots & \hat{\hat{\mathbf{W}}}_{G,G} \end{pmatrix} \begin{array}{c} M-k_1 \\ \vdots \\ M-k_G \end{array} \end{array}$$

ise kesin blok üst üçgen bir matristir ve elemanları;

$$\hat{\hat{\mathbf{W}}}_{i,j} = \begin{cases} \gamma_{i,j}\hat{\mathbf{Q}}_i^T\mathbf{X}_j & , \quad i<j \\ \mathbf{0} & , \quad i\geq j \end{cases} \tag{4.11}$$

olarak elde edilir. Bu hesaplamalarda yer alan $\gamma_{i,j}$ ise, $\mathbf{C}^{-1}$'in (i,j) inci elemanıdır.

Son olarak $\tilde{\tilde{\mathbf{W}}}$ ve $\hat{\hat{\mathbf{W}}}$ matrisleri için;

$$\overline{\mathbf{Q}}^T \begin{pmatrix} \tilde{\tilde{\mathbf{W}}} \\ \hat{\hat{\mathbf{W}}} \end{pmatrix} = \begin{pmatrix} \overline{\mathbf{R}} \\ \mathbf{0} \end{pmatrix} \begin{matrix} K \\ GM-K \end{matrix}$$

ve (4.12)

$$\overline{\mathbf{Q}}^T \begin{pmatrix} Vec\left(\left\{\tilde{\tilde{\mathbf{y}}}_i\right\}\right) \\ Vec\left(\left\{\hat{\hat{\mathbf{y}}}_i\right\}\right) \end{pmatrix} = \begin{pmatrix} \overline{\mathbf{y}} \\ \overline{\mathbf{y}}_* \end{pmatrix} \begin{matrix} K \\ GM-K \end{matrix}$$

şeklinde QR ayrışımı güncellemesi (updating) yapılır.

QR ayrışımı için güncelleme işlemi, toplam $G-1$ aşamada yapılır ve her bir aşama iki adımdan oluşur. Bu güncellemeye ilişkin detaylar Ek-7'de verilmiştir.

Güncelleme işleminden elde edilen $\overline{\mathbf{Q}}^T$ ortogonal matrisinin (4.9) eşitliği ile soldan çarpımı sonucunda;

$$\overline{\mathbf{Q}}^T \begin{pmatrix} Vec\left(\left\{\tilde{\tilde{\mathbf{y}}}_i\right\}\right) \\ Vec\left(\left\{\hat{\hat{\mathbf{y}}}_i\right\}\right) \end{pmatrix} = \overline{\mathbf{Q}}^T \begin{pmatrix} \tilde{\tilde{\mathbf{W}}} \\ \hat{\hat{\mathbf{W}}} \end{pmatrix} \boldsymbol{\beta} + \overline{\mathbf{Q}}^T \begin{pmatrix} Vec\left(\tilde{\tilde{\mathbf{V}}}\right) \\ Vec\left(\hat{\hat{\mathbf{V}}}\right) \end{pmatrix}$$

veya kısaca,

$$\begin{pmatrix} \overline{\mathbf{y}} \\ \overline{\mathbf{y}}_* \end{pmatrix} = \begin{pmatrix} \overline{\mathbf{R}} \\ \mathbf{0} \end{pmatrix} \boldsymbol{\beta} + \overline{\mathbf{Q}}^T \begin{pmatrix} Vec\left(\tilde{\tilde{\mathbf{V}}}\right) \\ Vec\left(\hat{\hat{\mathbf{V}}}\right) \end{pmatrix}$$

üçgensel sistemi elde edilir ki bu sistemin en küçük kareler yöntemi ile çözümü sonucunda da $\boldsymbol{\beta}$'nın en iyi doğrusal sapmasız tahmincisi;

$$\boldsymbol{\beta} = \overline{\mathbf{R}}^{-1}\overline{\mathbf{y}}$$

olarak elde edilir.

Sıradan doğrusal model haline getirilen görünüşte ilişkisiz regresyon modelinin QR ayrışımı kullanılarak çözümüne ilişkin algoritma adımsal olarak şu şekilde verilebilir:

Algoritma I: Sıradan Doğrusal Model (4.5)'in Sıradan En Küçük Kareler Tahmin Algoritması

Girdi Değişkenleri : $\mathbf{X}_1,\ldots,\mathbf{X}_G$ bağımsız değişken matrisleri, $\mathbf{Y}$ bağımlı değişken vektörü ve $\mathbf{\Sigma}$ artık varyans-kovaryans matrisi (biliniyorsa)

Çıktı Değişkeni : $\boldsymbol{\beta}$ katsayı tahmin vektörü

Adım 1. $\mathbf{\Sigma} = \mathbf{C}\mathbf{C}^T$ olacak şekilde Cholesky ayrışımı

Adım 2. $\mathbf{C}^{-1} = \left[\gamma_{i,j}\right]$ olmak üzere $\overline{\mathbf{y}} = \mathbf{Y}\mathbf{C}^{-T}$ 'nin hesaplanması

Adım 3. *for* $i = 1,\ldots,G$ döngü başlangıcı

Adım 4. (4.7) ile verilen her bir denklemdeki bağımsız değişken matrislerinin ayrı ayrı QR ayrışımlarının hesaplanması ve elde edilen Q ortogonal matrisinin yeniden yapılandırılması

Adım 5. (4.9) eşitliğindeki $\tilde{\overline{\mathbf{y}}}_i = \tilde{\mathbf{Q}}_i^T \overline{\mathbf{y}}_i$ ve $\hat{\overline{\mathbf{y}}}_i = \hat{\mathbf{Q}}_i^T \overline{\mathbf{y}}_i$ vektörlerinin hesaplanması

Adım 6. *end for* döngü sonu

Adım 7. (4.10) eşitliğinden $\tilde{\overline{\mathbf{W}}}$ ve (4.11) eşitliğinden $\hat{\overline{\mathbf{W}}}$ matrislerinin oluşturulması

Adım 8. (4.12) eşitliğinde verilen QR güncelleme işlemlerinin yapılması

Adım 9. $\boldsymbol{\beta} = \overline{\mathbf{R}}^{-1}\overline{\mathbf{y}}$ şeklinde katsayı tahmin vektörünün elde edilmesi.

Yukarıda adımsal olarak verilen Algoritma I'in çözümleme sürecine ilişkin Matlab kodu Ek-8'de verilmiştir.

4.2. Genelleştirilmiş QR Ayrışımı ile Genelleştirilmiş Doğrusal En Küçük Kareler Problemi

Genelleştirilmiş doğrusal en küçük kareler problemi (4.4)'ün çözümü, $\oplus_i \mathbf{X}_i$ ve $\left(\mathbf{C} \otimes \mathbf{I}_M\right)$'nin genelleştirilmiş QR ayrışımı hesabından elde edilir (Björk, 1996; Kontoghiorghes ve Dinenis, 1997; Kontoghiorghes, 2000b; Kourouklis ve Paige, 1981; Paige, 1990). Yani ilk olarak (4.7) eşitliği ile verilen QR ayrışımları daha sonra da RQ ayrışımı hesaplamaları sonucunda elde edilir. RQ ayrışımı ise;

$$\mathbf{Q}^T\left(\mathbf{C} \otimes \mathbf{I}_M\right)\mathbf{P} = \mathbf{W} \equiv \begin{array}{cc} & \begin{array}{cc} K & GM-K \end{array} \\ & \begin{pmatrix} \mathbf{W}_{AA} & \mathbf{W}_{AB} \\ \mathbf{0} & \mathbf{W}_{BB} \end{pmatrix} \begin{array}{l} K \\ GM-K \end{array} \end{array} \tag{4.13}$$

şeklinde ifade edilir ve burada $K = \sum_{i=1}^{G} k_i$, $\mathbf{P} \in \mathbb{R}^{GMxGM}$ ortogonal ve $\mathbf{W}_{BB}$ üst üçgen matrislerdir.

$\mathbf{Q}^T$'nin (4.4) eşitliğindeki kısıtlamalarla önden çarpımı ve $Vec(\mathbf{V}) = \mathbf{P}\mathbf{P}^T Vec(\mathbf{V})$ eşitliği kullanılarak genelleştirilmiş doğrusal en küçük kareler problemi formülasyonu şu şekilde yeniden yazılabilir:

$$\begin{gathered} \underset{\boldsymbol{\beta},\{\tilde{\mathbf{v}}_i\},\{\hat{\mathbf{v}}_i\}}{\arg\min} \sum_{i=1}^{G}\left(\left\|\tilde{\mathbf{v}}_i\right\|^2 + \left\|\hat{\mathbf{v}}_i\right\|^2\right) \quad , \\ \begin{pmatrix} Vec\left(\{\tilde{\mathbf{y}}_i\}\right) \\ Vec\left(\{\hat{\mathbf{y}}_i\}\right) \end{pmatrix} = \begin{pmatrix} \oplus_i \mathbf{R}_i \\ \mathbf{0} \end{pmatrix}\boldsymbol{\beta} + \begin{pmatrix} \mathbf{W}_{AA} & \mathbf{W}_{AB} \\ \mathbf{0} & \mathbf{W}_{BB} \end{pmatrix}\begin{pmatrix} Vec\left(\{\tilde{\mathbf{v}}_i\}\right) \\ Vec\left(\{\hat{\mathbf{v}}_i\}\right) \end{pmatrix} \text{'ye göre...} \end{gathered} \tag{4.14}$$

Burada $\tilde{\mathbf{Q}}_i \mathbf{y}_i = \tilde{\mathbf{y}}_i$, $\hat{\mathbf{Q}}_i \mathbf{y}_i = \hat{\mathbf{y}}_i$, $\mathbf{P}^T Vec(\mathbf{V}) = (Vec(\{\tilde{\mathbf{v}}_i\})^T \quad Vec(\{\hat{\mathbf{v}}_i\})^T)^T$, $\tilde{\mathbf{y}}_i, \tilde{\mathbf{v}}_i \in \mathbb{R}^{k_i}$ ve $\hat{\mathbf{y}}_i, \hat{\mathbf{v}}_i \in \mathbb{R}^{M-k_i}$'dir. (4.14) eşitliği çözülürken minimizasyonun sağlanabilmesi için bileşenlerden bir tanesi sıfır yapılarak kalan bileşenin minumum yapılması hedeflenir. Buna göre;

$$Vec(\{\tilde{\mathbf{v}}_i\}) = 0$$

ve

$$\begin{pmatrix} \oplus_i \mathbf{R}_i & \mathbf{W}_{AB} \\ \mathbf{0} & \mathbf{W}_{BB} \end{pmatrix}\begin{pmatrix} \boldsymbol{\beta} \\ Vec\left(\{\hat{\mathbf{v}}_i\}\right) \end{pmatrix} = \begin{pmatrix} Vec\left(\{\tilde{\mathbf{y}}_i\}\right) \\ Vec\left(\{\hat{\mathbf{y}}_i\}\right) \end{pmatrix} \tag{4.15}$$

elde edilir.

Hesaplamalarda RQ ayrışımı (4.13) iki adımda elde edilir. İlk adımda,

$$\mathbf{Q}^T\left(\mathbf{C}\otimes\mathbf{I}_T\right)\mathbf{\Pi}=\tilde{\mathbf{W}}=\begin{array}{cc} \begin{array}{cc} K & GM-K \end{array} & \\ \begin{pmatrix} \tilde{\mathbf{W}}_{AA} & \tilde{\mathbf{W}}_{AB} \\ \tilde{\mathbf{W}}_{BA} & \tilde{\mathbf{W}}_{BB} \end{pmatrix} & \begin{array}{c} K \\ GM-K \end{array} \end{array} \tag{4.16}$$

permütasyonu hesaplanır ki burada $\mathbf{\Pi}=\left(\oplus_i\left(\mathbf{I}_{k_i}\quad \mathbf{0}\right)^T\quad \oplus_i\left(\mathbf{0}\quad \mathbf{I}_{M-k_i}\right)^T\right)$'dur.

Böylece $\tilde{\mathbf{W}}_{AA}, \tilde{\mathbf{W}}_{AB}, \tilde{\mathbf{W}}_{BA}$ *ve* $\tilde{\mathbf{W}}_{BB}$ blok üst üçgen matrisler olur.

İkinci adımda ise,

$$\begin{pmatrix}\tilde{\mathbf{W}}_{BA} & \tilde{\mathbf{W}}_{BB}\end{pmatrix}\tilde{\mathbf{P}}=\begin{pmatrix}\mathbf{0} & \mathbf{W}_{BB}\end{pmatrix} \tag{4.17a}$$

ve

$$\begin{pmatrix}\tilde{\mathbf{W}}_{AA} & \tilde{\mathbf{W}}_{AB}\end{pmatrix}\tilde{\mathbf{P}}=\begin{pmatrix}\mathbf{W}_{AA} & \mathbf{W}_{AB}\end{pmatrix} \tag{4.17b}$$

şeklinde RQ ayrışımı yapılır. (4.17a) eşitliği ile verilen RQ ayrışımı;

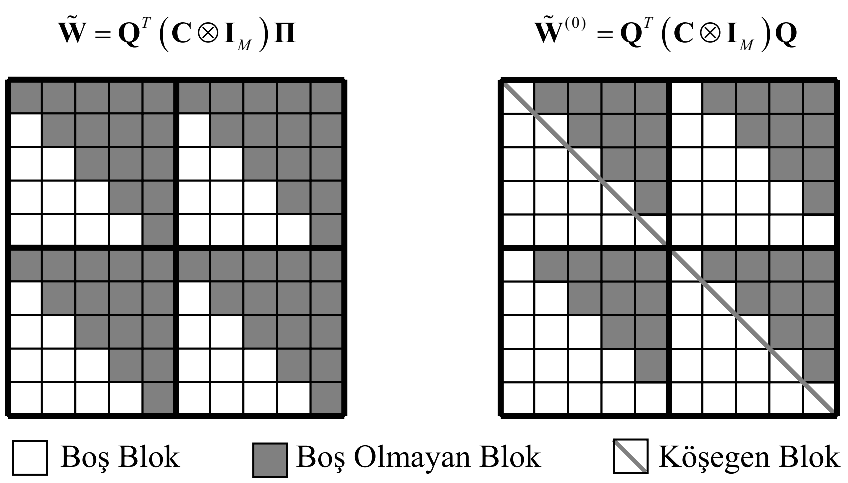
$$\tilde{\mathbf{P}}^T\begin{pmatrix}\tilde{\mathbf{W}}_{BA}^T \\ \tilde{\mathbf{W}}_{BB}^T\end{pmatrix}=\begin{pmatrix}\mathbf{0} \\ \mathbf{W}_{BB}^T\end{pmatrix}$$

şeklinde hesaplanacak QL ayrışımına eşittir. RQ ayrışımını yapmak için gereken Matlab kodu Ek-6'da verilmişti. Burada $\tilde{\mathbf{P}}\in\mathbb{R}^{GMxGM}$ ortogonaldir. Şunu belirtmek gerekir ki (4.13) eşitliğindeki $\mathbf{P}$, $\mathbf{\Pi}\tilde{\mathbf{P}}$ ifadesine eşittir ve (4.16) ve (4.17) eşitliklerindeki bütün matrisin RQ ayrışımından hesaplanmıştır.

(4.16) eşitliğindeki permütasyon, $\mathbf{Q}^T\left(\mathbf{C}\otimes\mathbf{I}_T\right)$ ifadesinin sağdan $\mathbf{Q}$ ortogonal matrisi ile çarpımı şeklinde de oluşturulabilir. Bu iki sonuç birbiriyle aynıdır. Yani,

$$\mathbf{Q}^T\left(\mathbf{C}\otimes\mathbf{I}_T\right)\mathbf{Q}=\tilde{\mathbf{W}}^{(0)}\equiv\begin{array}{cc} \begin{array}{cc} K & GM-K \end{array} & \\ \begin{pmatrix} \tilde{\mathbf{W}}_{AA}^{(0)} & \tilde{\mathbf{W}}_{AB}^{(0)} \\ \tilde{\mathbf{W}}_{BA}^{(0)} & \tilde{\mathbf{W}}_{BB}^{(0)} \end{pmatrix} & \begin{array}{c} K \\ GM-K \end{array} \end{array} \tag{4.18}$$

olur. Burada $\tilde{\mathbf{W}}_{AA}^{(0)}$ ve $\tilde{\mathbf{W}}_{BB}^{(0)}$ alt matrisleri sırasıyla $\mathbf{C}_{i,i}\mathbf{I}_{k_i}$ ve $\mathbf{C}_{i,i}\mathbf{I}_{M-k_i}$ şeklinde hesaplanan ve ana köşegene göre blok üst üçgen matrislerdir. Ayrıca $\tilde{\mathbf{W}}_{AB}^{(0)}$ ve $\tilde{\mathbf{W}}_{BA}^{(0)}$ matrisleri ise kesin blok üst üçgendirler.



Şekil 4.1. $\tilde{\mathbf{W}} = \mathbf{Q}^T \left(\mathbf{C} \otimes \mathbf{I}_M \right) \mathbf{\Pi}$ ve $\tilde{\mathbf{W}}^{(0)} = \mathbf{Q}^T \left(\mathbf{C} \otimes \mathbf{I}_M \right) \mathbf{Q}$ matrislerinin $G = 5$ durumu için yapısal gösterimi

(4.18) eşitliği ile verilen $\tilde{\mathbf{W}}^{(0)}$ blok matrisi kullanılarak (4.17a) ve (4.17b) eşitlikleri;

$$\begin{pmatrix} \tilde{\mathbf{W}}_{BA}^{(0)} & \tilde{\mathbf{W}}_{BB}^{(0)} \end{pmatrix} \tilde{\mathbf{P}}^{(0)} = \begin{pmatrix} \mathbf{0} & \mathbf{W}_{BB} \end{pmatrix} \tag{4.19a}$$

ve

$$\begin{pmatrix} \tilde{\mathbf{W}}_{AA}^{(0)} & \tilde{\mathbf{W}}_{AB}^{(0)} \end{pmatrix} \tilde{\mathbf{P}}^{(0)} = \begin{pmatrix} \mathbf{W}_{AA} & \mathbf{W}_{AB} \end{pmatrix} \tag{4.19b}$$

olarak yeniden yazılır. Ancak (4.19a) eşitliği ile verilen RQ ayrışımı, $\tilde{\mathbf{W}}$ ve $\tilde{\mathbf{W}}^{(0)}$ matrisleri arasındaki yapısal farklılıktan dolayı Ek-6'da verilen kodlama ile hesaplanamaz. Bunun nedeni $\tilde{\mathbf{W}}_{BA}^{(0)}$ blok matrisinin ana köşegeninde yer alan matrislerin sadece köşegen elemanlarının sıfırdan farklı değerlere sahip olmasıdır. Dolayısıyla bu ayrışımının yapılabilmesi için bir güncelleme yapmak gereklidir. Bu güncelleme işlemine ilişkin detaylar Ek-9'da verilmiştir.

Yapılan bu güncellemeden sonra (4.13) eşitliği ile verilen $\mathbf{W}$ matrisi elde edilmiş olur. Son olarak (4.15) eşitliği ile verilen üçgensel sistemden elde edilen;

$$Vec\left(\left\{\hat{\mathbf{y}}_i\right\}\right) = \mathbf{W}_{BB} Vec\left(\left\{\hat{\mathbf{v}}_i\right\}\right) \tag{4.20a}$$

eşitliği $Vec\left(\left\{\hat{\mathbf{v}}_i\right\}\right)$ için çözümlendikten sonra elde edilen değer

$$Vec\left(\{\tilde{\mathbf{y}}_i\}\right) = \left(\oplus_i \mathbf{R}_i\right)\boldsymbol{\beta} + \mathbf{W}_{AB} Vec\left(\{\hat{\mathbf{v}}_i\}\right) \tag{4.20b}$$

eşitliğinde yerine konularak $\boldsymbol{\beta}$ katsayılar tahmin vektörü elde edilir.

Genelleştirilmiş doğrusal en küçük kareler problemi haline getirilen görünüşte ilişkisiz regresyon modelinin genelleştirilmiş QR ayrışımı kullanılarak çözümüne ilişkin algoritma adımsal olarak şu şekilde verilebilir:

Algoritma II: Genelleştirilmiş QR Ayrışımı Kullanılarak Genelleştirilmiş Doğrusal En Küçük Kareler Problemi (4.4)'ün Çözüm Algoritması

Girdi Değişkenleri : $\mathbf{X}_1, \ldots, \mathbf{X}_G$ bağımsız değişken matrisleri, $\mathbf{Y}$ bağımlı değişken vektörü ve $\boldsymbol{\Sigma}$ artık varyans-kovaryans matrisi (biliniyorsa)

Çıktı Değişkeni : $\boldsymbol{\beta}$ katsayı tahmin vektörü

Adım 1. $\boldsymbol{\Sigma} = \mathbf{C}\mathbf{C}^T$ olacak şekilde Cholesky ayrışımı

Adım 2. $for\ i = 1, \ldots, G$ döngü başlangıcı

Adım 3. (4.7) ile verilen her bir denklemdeki bağımsız değişken matrislerinin ayrı ayrı QR ayrışımlarının hesaplanması ve elde edilen Q ortogonal matrisinin yeniden yapılandırılması

Adım 4. (4.14) eşitliğindeki $\tilde{\mathbf{y}}_i = \tilde{\mathbf{Q}}_i^T \mathbf{y}_i$ ve $\hat{\mathbf{y}}_i = \hat{\mathbf{Q}}_i^T \mathbf{y}_i$ vektörlerinin hesaplanması

Adım 5. $end\ for$ döngü sonu

Adım 6. (4.18) eşitliği ile verilen $\tilde{\mathbf{W}}^{(0)} = \mathbf{Q}^T\left(\mathbf{C} \otimes \mathbf{I}_T\right)\mathbf{Q}$ blok matrisinin hesaplanması

Adım 7. (4.19a) ve (4.19b) RQ (veya QL) güncellemesinin yapılması

Adım 8. (4.20a) eşitliğinin $Vec(\{\hat{\mathbf{v}}_i\})$ için çözümü

Adım 9. (4.20b) eşitliğinden $\boldsymbol{\beta}$ katsayı tahmin vektörünün elde edilmesi.

Yukarıda adımsal olarak verilen Algoritma II'nin çözümleme sürecine ilişkin Matlab kodu Ek-10'da verilmiştir.

5. DENEYSEL KARŞILAŞTIRMALAR

Bu bölümde, bu tez çalışması içerisinde sunulan algoritmaların deneysel veriler kullanılarak elde edilen çözüm süreleri karşılaştırılmıştır. Bu karşılaştırmalar için IBM Intel Centrino Duo T2500 2GHz işlemciye ve 1GB RAM'a sahip bilgisayar kullanılmıştır. Yapılan tüm karşılaştırmalarda ilgili program kodlarının her birisi 15 defa işletilmiş ve yapılan çizelgelerde bu tekrarlar sonucunda elde edilen çözümleme sürelerinin ortalama değerlerine yer verilmiştir. Çizelgelerde yer alan tüm değerler saniye cinsinden çözümleme sürelerini ifade etmektedir.

İlk olarak, QR ayrışımının elde edilmesinde kullanılan Householder ve Givens yöntemleri için oluşturulan ve sırasıyla Ek-2 ve Ek-4'de sunulan Matlab kodları çözümleme süreleri bakımından karşılaştırılmıştır. Bu karşılaştırmalar yapılırken her bir karşılaştırma için elemanları (0,1) aralığında rassal olarak değer alan ve tekdüze (uniform) dağılıma sahip değerlerden oluşan $m \times m$ boyutlu ($m = 5, 10, 25, 50, 75, 100, 250, 500$) kare matrisler türetilmiştir. Oluşturulan bu matrislerin her biri için her iki kod da işletilmiş ve çözüm süreleri elde edilmiştir.

QRhouse (Ek-2) ve QRgivens (Ek-4) kodları için elde edilen çözümleme süreleri Çizelge 5.1'de verilmiştir.

Çizelge 5.1. QRhouse (Ek-2) ve QRgivens (Ek-4) kodları için çözümleme süreleri (saniye)

A matrisi ($m \times m$)	**QRhouse(A)** (Ek-2)	**QRgivens(A)** (Ek-4)
5x5	0,0150	0,0150
10x10	0,0160	0,0160
25x25	0,0310	0,0320
50x50	0,0320	0,2970
75x75	0,0470	1,7340
100x100	0,0790	6,9220
250x250	1,5470	580,7660
500x500	24,5150	1,9703e+004

Çizelge 5.1 incelendiğinde Householder dönüşüm yöntemi ve Givens rotasyon yöntemleri ile oluşturulan QR ayrışım kodlarının çözümleme sürelerinin $m = 5,10$ ve 25 değerleri için aynı olduğu görülmektedir. Ancak $m = 50$ ve daha büyük boyuttaki matrisler için Householder dönüşüm yöntemi kullanılarak oluşturulan QR ayrışım algoritmasının Givens rotasyon yöntemi ile oluşturulan algoritmaya göre çok daha kısa sürede çözümleme yaptığı görülmektedir. Dolayısıyla QR ayrışımını içerecek şekilde oluşturulacak diğer algoritmalarda Householder dönüşüm yöntemi ile bu ayrışımı gerçekleştiren kodlamanın kullanılması işlem yükünü azaltacak ve toplam çözümleme süresi açısından algoritmanın performansını olumlu yönde etkileyeceği sonucuna varılmıştır.

İkinci olarak, (3.10) eşitliği ile verilen görünüşte ilişkisiz regresyon modelinin, (4.3) eşitliği ile tanımlanan sıradan doğrusal model haline getirilerek Bölüm 4.1'de ayrıntılı olarak sunulan çözümleme süreci kendi içerisinde iki farklı şekilde ele alınarak incelenmiştir. İlkinde direkt toplam şeklinde oluşturulan $\oplus_{i=1}^{G} \mathbf{X}_i$ bağımsız değişkenler matrisi üzerinde herhangi bir işlem yapılmaksızın işletilecek şekilde algoritma yeniden düzenlenmiş ve katsayı tahmin vektörü elde edilmiştir. İkincisinde ise, $\oplus_{i=1}^{G} \mathbf{X}_i$'nin blok matris yapısından faydalanılarak her bir $\mathbf{X}_i$ bloğuna ayrı ayrı QR ayrışımı uygulamak suretiyle katsayı tahmin vektörünü veren ve Matlab kodu Ek-8'de verilen Algoritma I çözümleme süresi açısından karşılaştırılmışlardır. Bu karşılaştırma, $G = 3,5$ ve 10 denklem içeren görünüşte ilişkisiz regresyon modeli için, her bir denklemdeki bağımsız değişken sayıları sabit terim dahil olmak üzere $k_i = 3,5,10$ ve 15 olan ve $M = 30,50,75,100$ ve 250 olmak kaydıyla farklı gözlem sayıları için yapılmıştır. Karşılaştırmalar yapılırken denklemlerde ortak bağımsız değişken olmadığı varsayılmış ve tüm değişkenlere ilişkin gözlem değerleri tekdüze dağılımdan rassal olarak türetilmiştir.

Sıradan doğrusal model haline getirilen görünüşte ilişkisiz regresyon modelinin çözüm algoritmaları için yapılan karşılaştırmalar Çizelge 5.2'de verilmiştir.

Çizelge 5.2a. Sıradan doğrusal model haline getirilen ve $G = 3$ denklem içeren görünüşte ilişkisiz regresyon modeli için çözümleme süreleri (saniye)

DENKLEM SAYISI (G)	BAĞIMSIZ DEĞİŞKEN SAYISI (k_i)	GÖZLEM SAYISI (M)	SIRADAN DOĞRUSAL MODEL (4.3) ÇÖZÜM ALGORİTMALARI	
			$\oplus_{i=1}^{G}\mathbf{X}_i$ BLOK MATRİSİ İLE	ALGORİTMA I (AYRI AYRI $\mathbf{X}_i$'LER İLE)
3	3	30	0,0470	0,0310
		50	0,0780	0,0310
		75	0,1720	0,0470
		100	0,3750	0,0620
		250	4,8280	0,5790
	5	30	0,0470	0,0310
		50	0,0940	0,0320
		75	0,2500	0,0470
		100	0,5630	0,0940
		250	7,8440	0,9380
	10	30	0,0620	0,0310
		50	0,1410	0,0460
		75	0,4220	0,0780
		100	1,0160	0,1250
		250	15,2970	1,7650
	15	30	0,0630	0,0310
		50	0,1720	0,0460
		75	0,5470	0,0930
		100	1,4220	0,1710
		250	22,1880	2,5000

Çizelge 5.2b. Sıradan doğrusal model haline getirilen ve $G = 5$ denklem içeren görünüşte ilişkisiz regresyon modeli için çözümleme süreleri (saniye)

DENKLEM SAYISI (G)	BAĞIMSIZ DEĞİŞKEN SAYISI (k_i)	GÖZLEM SAYISI (M)	SIRADAN DOĞRUSAL MODEL (4.3) ÇÖZÜM ALGORİTMALARI	
			$\oplus_{i=1}^{G}\mathbf{X}_i$ BLOK MATRİSİ İLE	ALGORİTMA I (AYRI AYRI $\mathbf{X}_i$ 'LER İLE)
5	3	30	0,0940	0,0470
		50	0,3440	0,0780
		75	1,0630	0,1870
		100	2,4380	0,4220
		250	35,0630	5,7970
	5	30	0,1250	0,0470
		50	0,5000	0,1100
		75	1,6400	0,2810
		100	3,9220	0,6410
		250	57,9530	9,4060
	10	30	0,2030	0,0470
		50	0,8280	0,1410
		75	2,9690	0,4530
		100	7,1870	1,0940
		250	113,0780	17,8440
	15	30	0,2340	0,0620
		50	1,0780	0,1560
		75	4,0780	0,5470
		100	9,9850	1,4220
		250	579,7190	25,2970

Çizelge 5.2c. Sıradan doğrusal model haline getirilen ve $G = 10$ denklem içeren görünüşte ilişkisiz regresyon modeli için çözümleme süreleri (saniye)

DENKLEM SAYISI (G)	BAĞIMSIZ DEĞİŞKEN SAYISI (k_i)	GÖZLEM SAYISI (M)	SIRADAN DOĞRUSAL MODEL (4.3) ÇÖZÜM ALGORİTMALARI	
			$\oplus_{i=1}^{G} \mathbf{X}_i$ BLOK MATRİSİ İLE	ALGORİTMA I (AYRI AYRI $\mathbf{X}_i$ 'LER İLE)
10	3	30	1,0160	0,2030
		50	4,5630	0,9070
		75	15,1250	3,0780
		100	35,5150	7,1570
		250	Hesaplanamadı	111,7180
	5	30	1,5310	0,2710
		50	7,2180	1,3430
		75	24,2030	4,7190
		100	57,3590	11,3280
		250	Hesaplanamadı	179,8910
	10	30	2,4060	0,2770
		50	12,3430	1,9380
		75	44,2970	7,7340
		100	106,7340	19,4840
		250	Hesaplanamadı	340,2040
	15	30	2,7970	0,2810
		50	16,0780	2,0310
		75	60,2030	9,2820
		100	1447,6000	25,1560
		250	Hesaplanamadı	543,9370

Sıradan doğrusal model haline getirilen görünüşte ilişkisiz regresyon modelinin çözüm algoritmaları için yapılan karşılaştırmaları gösteren Çizelge 5.2a, b ve c incelendiğinde, her bir denklemdeki bağımsız değişkenler matrislerine ayrı ayrı QR ayrışımı uygulayarak yapılan çözümlemenin hesaplama maliyeti açısından çok daha az olduğu sonucuna varılır. Öyle ki tüm denklemlerdeki bağımsız değişken matrislerini içeren blok matrisin boyutu, modelin içerdiği denklem sayısı, her bir denklemdeki bağımsız değişken sayısı ve bu değişkenlere ilişkin gözlem sayıları arttıkça çok büyümekte ve bilgisayar hafızasında büyük yer işgal etmektedir. Dolayısıyla bu büyüklükte bir matrisle yapılan cebirsel işlemler sonucunda da benzer boyutlarda başka matrisler de elde edileceğinden çözümleme giderek yapılamaz duruma gelmektedir. Çizelge 5.2c'deki hesaplanamayan çözümleme süreleri bu durumu açıkça ifade etmektedir. Benzer şekilde karşılaştırma yapılan değerlerin artışı Algoritma I'inde çözümleme süresinde ciddi artışlara neden olmaktadır. Ancak her iki algoritma için de süreler değerlendirildiğinde aralarında azımsanamayacak derecede bir fark olduğu rahatlıkla görülebilir.

Sonuç olarak, sıradan doğrusal model haline getirilen görünüşte ilişkisiz regresyon modellerinin çözümlenmesinde, QR ayrışımı temeline dayanarak oluşturulan algoritmalardan bağımsız değişkenler matrislerinin her birini ayrı ayrı işleme sokarak çözümleme yapan Algoritma I'in tercih edilmesi hesaplama maliyeti açısından daha doğru olacaktır.

Üçüncü ve son olarak (3.10) eşitliği ile verilen görünüşte ilişkisiz regresyon modelinin, (4.4) eşitliği ile tanımlanan genelleştirilmiş doğrusal en küçük kareler problemi haline getirilerek Bölüm 4.2'de ayrıntılı olarak sunulan çözümleme süreci incelenmiştir. Özellikle algoritmanın ikinci kısmını oluşturan RQ (veya QL) ayrışımı ve güncelleme işlemi üzerinde bir karşılaştırma söz konusu olacaktır. Dolayısıyla bu süreç de kendi içerisinde iki farklı şekilde ele alınmıştır. İlkinde RQ ayrışımı için $\mathbf{\Pi}$ permütasyon matrisi oluşturulmuş ve (4.16) eşitliğini temel alarak çözümleme yapan bir algoritma düzenlenmiş ve katsayı tahmin vektörü elde edilmiştir. İkincisinde ise bu ayrışım, algoritmanın birinci kısmı olan QR ayrışımından elde edilmiş olan $\mathbf{Q}$ ortogonal matrisi kullanılarak yani (4.18) eşitliği temel alınarak çözümleme yapacak şekilde oluşturulan ve Matlab kodu Ek-10'da verilen Algoritma II kullanılarak katsayı tahmin vektörü elde edilmiştir. Daha sonra bu iki algoritma çözümleme süresi açısından karşılaştırılmışlardır. Bu karşılaştırma için de bir önceki karşılaştırmada olduğu gibi, $G = 3, 5 \text{ ve } 10$ denklem içeren görünüşte ilişkisiz regresyon modeli ele alınmış, her bir denklemdeki bağımsız değişken sayıları sabit terim dahil olmak üzere $k_i = 3, 5, 10 \text{ ve } 15$ olarak seçilmiş ve her bir değişkene ait gözlem sayıları da $M = 30, 50, 75, 100 \text{ ve } 250$ olarak alınmıştır. Yine bu karşılaştırmalar yapılırken denklemlerde ortak bağımsız değişken olmadığı varsayılmış ve tüm değişkenlere ilişkin gözlem değerleri tekdüze dağılımdan rassal olarak türetilmiştir.

Genelleştirilmiş doğrusal en küçük kareler problemi haline getirilen görünüşte ilişkisiz regresyon modeli için yapılan karşılaştırmalar ise Çizelge 5.3'de verilmiştir.

Çizelge 5.3a. Genelleştirilmiş doğrusal en küçük kareler problemi haline getirilen ve $G=3$ denklem içeren görünüşte ilişkisiz regresyon modeli için çözümleme süreleri (saniye)

DENKLEM SAYISI (G)	BAĞIMSIZ DEĞİŞKEN SAYISI (k_i)	GÖZLEM SAYISI (M)	GENELLEŞTİRİLMİŞ DOĞRUSAL EN KÜÇÜK KARELER PROBLEMİ (4.4) ÇÖZÜM ALGORİTMALARI	
			Π PERMÜTASYON MATRİSİ İLE	**Q** ORTOGONAL MATRİSİ İLE (ALGORİTMA II)
3	3	30	0,0940	0,0620
		50	0,2810	0,0930
		75	1,0780	0,1720
		100	3,4220	0,3130
		250	Hesaplanamadı	5,0310
	5	30	0,0940	0,0630
		50	0,2820	0,0940
		75	1,0780	0,1720
		100	3,4380	0,3280
		250	Hesaplanamadı	5,1560
	10	30	0,0940	0,0630
		50	0,2970	0,0940
		75	1,1090	0,1870
		100	3,4740	0,3280
		250	Hesaplanamadı	5,5160
	15	30	0,0940	0,0630
		50	0,3130	0,0940
		75	1,1250	0,1880
		100	3,7750	0,3590
		250	Hesaplanamadı	5,8900

Çizelge 5.3b. Genelleştirilmiş doğrusal en küçük kareler problemi haline getirilen ve $G=5$ denklem içeren görünüşte ilişkisiz regresyon modeli için çözümleme süreleri (saniye)

DENKLEM SAYISI (G)	BAĞIMSIZ DEĞİŞKEN SAYISI (k_i)	GÖZLEM SAYISI (M)	GENELLEŞTİRİLMİŞ DOĞRUSAL EN KÜÇÜK KARELER PROBLEMİ (4.4) ÇÖZÜM ALGORİTMALARI	
			Π PERMÜTASYON MATRİSİ İLE	**Q** ORTOGONAL MATRİSİ İLE (ALGORİTMA II)
5	3	30	0,2810	0,0780
		50	1,6560	0,1720
		75	8,2030	0,4060
		100	24,5620	0,8590
		250	Hesaplanamadı	14,3280
	5	30	0,2810	0,0930
		50	1,6720	0,1720
		75	8,2660	0,4070
		100	24,7320	0,8750
		250	Hesaplanamadı	14,7350
	10	30	0,2810	0,0940
		50	1,6590	0,1870
		75	8,1590	0,4530
		100	24,7310	0,9530
		250	Hesaplanamadı	15,9220
	15	30	0,2820	0,1090
		50	1,5940	0,2030
		75	8,0160	0,5000
		100	24,5930	1,0620
		250	Hesaplanamadı	17,1410

Çizelge 5.3c. Genelleştirilmiş doğrusal en küçük kareler problemi haline getirilen ve $G = 10$ denklem içeren görünüşte ilişkisiz regresyon modeli için çözümleme süreleri (saniye)

DENKLEM SAYISI (G)	BAĞIMSIZ DEĞİŞKEN SAYISI (k_i)	GÖZLEM SAYISI (M)	GENELLEŞTİRİLMİŞ DOĞRUSAL EN KÜÇÜK KARELER PROBLEMİ (4.4) ÇÖZÜM ALGORİTMALARI	
			Π PERMÜTASYON MATRİSİ İLE	**Q** ORTOGONAL MATRİSİ İLE (ALGORİTMA II)
10	3	30	3,1410	0,2190
		50	23,4060	0,6870
		75	Hesaplanamadı	2,0320
		100	Hesaplanamadı	4,6100
		250	Hesaplanamadı	73,8440
	5	30	3,1720	0,2340
		50	23,6560	0,7190
		75	Hesaplanamadı	2,1090
		100	Hesaplanamadı	4,9530
		250	Hesaplanamadı	75,7340
	10	30	3,3440	0,2970
		50	23,9690	0,8750
		75	Hesaplanamadı	2,4690
		100	Hesaplanamadı	5,4060
		250	Hesaplanamadı	82,8130
	15	30	3,4530	0,3910
		50	23,9880	1,1560
		75	Hesaplanamadı	3,1400
		100	Hesaplanamadı	6,5000
		250	Hesaplanamadı	91,2810

Çizelge 5.3a, b ve c'de genelleştirilmiş en küçük kareler problemi haline getirilen ve genelleştirilmiş QR ayrışımı temeline dayanarak Matlab paket programında oluşturulan kodlar çözümleme süreleri bakımından farklı denklem sayıları, farklı bağımsız değişken sayıları ve farklı gözlem sayıları için karşılaştırılmışlardır. Oluşturulan algoritmalar özellikle genelleştirilmiş QR ayrışımının ve dolayısıyla çözümlemenin ikinci aşamasını oluşturan RQ (veya QL) ayrışımının gerçekleşme sürecinin hesaplama açısından daha etkin kılınması

üzerine tasarlanmıştır. Bu amaçla oluşturulan ilk algoritma $\mathbf{\Pi}$ permütasyon matrisinin oluşturulmasını ve bu şekilde RQ ayrışımının gerçekleştirilmesini temel olarak alırken ikinci algoritmada bu ayrışım süreci çözümlemenin ilk aşamasındaki QR ayrışımından elde edilen ortogonal $\mathbf{Q}$ matrisinin permütasyon matrisi yerine kullanılması temeline dayanmaktadır. Deneysel veriler ile bahsedilen bu algoritmaların çözümleme süreleri için yapılan karşılaştırmaların yer aldığı bu çizelgeler (Çizelge 5.3a, b ve c) incelendiğinde ilk iki çizelgede dahi değişkenlere ilişkin gözlem sayıları 250 olarak belirlendiğinde permütasyon matrisinin oluşturulmasının ve buna bağlı olarak sürecin işletilmesinde sonuç alınamadığı görülmektedir. Üçüncü çizelge olan Çizelge 5.3c'de ise bu hesaplamanın maliyeti daha net olarak gözlemlenmektedir. Modelde $G = 10$ adet denklemin yer aldığı tasarlanarak oluşturulan denemeler için her bir modeldeki bağımsız değişken sayısı $k_i = 3$ olsa dahi 75,100 ve 250 gözlem değerleri için bu algoritma ile çözüme ulaşılamamıştır. Yine çizelgelerden, yeniden bir matris oluşturmak yerine daha önceden elde edilmiş olan $\mathbf{Q}$ ortogonal matrisinin bir permütasyon matrisi gibi kullanılması ve bu kullanımdan kaynaklanan güncelleme işlemlerinin yapılması ile oluşturulan Algoritma II'nin tüm değerler için çözümleme yapabildiği gözlemlenmektedir. Her ne kadar karşılaştırma kriterlerindeki artışlar doğal olarak bu algoritmanın çözümleme süresini artırsa da Algoritma II'nin hesaplama açısından diğerine göre daha etkin olduğu sonucuna varılır.

Çizelge 5.2 ve 5.3'de (a,b ve c şıkları ile) sunulan ve sırasıyla sıradan doğrusal model ve genelleştirilmiş en küçük kareler problemi haline getirilerek çözümlemesi yapılan görünüşte ilişkisiz regresyon modelleri çözüm algoritmaları için çözümleme süreleri bir bütün olarak ele alınıp karşılaştırıldığında, özellikle 3 denklemin yer aldığı modellerde Algoritma I'in Algoritma II'ye göre daha etkin çözümleme süresine sahip olduğu görülmektedir. Ancak 5 ve 10 denklemin yer aldığı modellerde denklemlerin içerdikleri bağımsız değişken sayıları ve gözlem sayıları arttıkça Algoritma II'nin çözümleme süresinin Algoritma I'in çözümleme süresine göre hesaplama açısından daha etkin olduğu görülmektedir. Buna göre özellikle küçük boyutlu problemlerin çözümlenmesinde gereken varsayımların sağlanması koşulu altında Algoritma I'in tercih edilmesinin hesaplama etkinliği açısından daha doğru olacağı, büyük boyutlu problemlerde ise Algoritma II'nin tercih edilmesinin daha doğru olacağı söylenebilir.

Çeşitli algoritmalar için şimdiye kadar verilen tüm çizelgelerde gerek görünüşte ilişkisiz regresyon modeline dahil edilen denklem sayısı, gerek her bir denklemin içerdiği bağımsız değişken sayısı ve gerekse her bir değişkenin sahip olduğu gözlem sayıları artırıldıkça, problemin çözümleme sürecinde yer alan vektör ve matris boyutlarının aşırı derecede büyümeleri kaçınılmaz olmaktadır. Dolayısıyla bu büyüklükte vektör ve matrisler ile cebirsel işlemlerin gerçekleştirilmesi de içinde bulunulan teknoloji ile dahi bir noktadan sonra mümkün olamamaktadır.

6. SONUÇ VE ÖNERİLER

Bir regresyon denklemine ilişkin katsayılar tahminlenirken, klasik doğrusal regresyon varsayımlarının tümü geçerli olduğunda en küçük kareler yöntemi kullanılarak yapılacak tahminler yansız, tutarlı ve etkin olacaklardır. Ancak yapısal özellikleri bakımından birbirlerine benzer olan ve aynı zaman dilimi içerisinde elde edilmiş regresyon denklemlerinin hata terimleri arasında var olabilecek bir ilişki en küçük kareler yöntemi ile yapılan katsayı tahminlerinin etkinlik özelliğini kaybetmesine neden olmaktadır. Bu durumda hata terimleri arasında ilişki bulunan regresyon denklemlerini birarada inceleyen ve bu ilişkiyi de hesaba katarak katsayı tahminlerini en küçük karelere göre daha etkin bir şekilde gerçekleştiren görünüşte ilişkisiz regresyon modellerinin kullanılması gerekmektedir. Aralarında ilişki yokmuş gibi görünen ancak hata terimleri arasında ilişki olan bu türden denklemlerin bir arada değerlendirilmesi ve çözümlenmesi hesaplama açısından oldukça maliyetlidir.

Bu çalışmada görünüşte ilişkisiz regresyon denklemleri önce sıradan doğrusal model haline getirilerek, daha sonra da genelleştirilmiş doğrusal en küçük kareler problemi olarak ele alınmış ve bunların çözümleme algoritmaları sunulmuştur. Sunulan tüm algoritmalarda hesaplama yükünü azaltabilmek amacıyla kanonik matris formlarına dönüştürülerek işlemlerin gerçekleştirilmesi hedeflenmiştir. Bu amaçla Matlab kodları oluşturulan ve deneysel veriler yardımıyla karşılaştırılan algoritmalar ile hesaplama açısından daha etkin sonuçlar elde edilebileceği gözlemlenmiştir. Yapılan karşılaştırmalar sonucunda sıradan doğrusal model haline getirilen görünüşte ilişkisiz regresyon modelleri katsayı tahminlerini veren Algoritma I'in özellikle küçük boyutlu problemlerde hesaplama açısından daha etkin olduğu gözlemlenmiştir. Ancak sıradan doğrusal model haline getirilmiş görünüşte ilişkisiz regresyon modellerinin çözümünü gerçekleştiren Algoritma I'in modele ilişkin varyans-kovaryans matrisi $\boldsymbol{\Sigma}$ tekil olduğunda nümerik açıdan tutarsız sonuçlar vereceği unutulmamalıdır. Bu durumda sıradan doğrusal modele nazaran daha az kısıtlama içeren ve problemin boyutları büyüdükçe hesaplama açısından daha etkin bir konuma gelen genelleştirilmiş en küçük kareler problemi yaklaşımını kullanarak oluşturulan Algoritma II'nin kullanılması daha sağlıklı sonuçlar verecektir.

Bu çalışmada sunulan algoritmalar, son zamanlarda özellikle bilgisayar program kodlarının oluşturulmasında sıkça kullanılan sparse(ayrık, seyrek) matris yapıları kullanılarak daha da geliştirilebilir ve hesaplama maliyetleri daha da azaltılarak daha büyük veri setleri için uygulanabilir bir hale getirilebilirler. Ancak bu şekilde yapılacak kodlamalarda özellikle tam sıfıra eşit olmayan değerlerde ortaya çıkabilecek yuvarlama hatalarının, alınacak sonuçlar üzerinde ciddi etkilerinin olabileceği gözardı edilmemelidir. Ayrıca sparse matris ve tam(full) matris yapıları arasında yapılacak geçişlerin fazladan bir hesap yükü oluşturacağı da unutulmamalıdır.

Ayrıca Bölüm 3.3'de verilen Görünüşte ilişkisiz regresyon modellerinin çözümünde karşılaşılabilecek çeşitli durumlar için bu çalışmada sunulan algoritmalar ve bunlara ilişkin Matlab kodları geliştirilebilir. Bu sayede daha fazla kısıt içeren farklı problemler için de bu modellerin katsayı tahminlerinin elde edilmesi sağlanabilir ve gerçek veriler kullanılarak yapılabilecek uygulama yelpazesi de genişletilmiş olur.

Her ne kadar çözümleme süreci için daha etkin algoritmalar geliştirilmeye çalışılsa da bu algoritmaların nümerik açıdan stabil olup olmadıklarının araştırılması gerekmektedir. Ayrıca sunulan Matlab kodlarının bir programcı tarafından incelenmesinin ve programlama açısından etkinliği artırıcı düzeltmelerin yapılmasının algoritmaların performanslarını olumlu yönde etkileyeceği gerçeği de gözardı edilmemelidir.

Son olarak, bilgisayar ortamında belirli kodlar işletilerek elde edilecek çözümleme sürelerinin kullanılan bilgisayarın sahip olduğu donanım özellikleri ile bire bir ilişkili olduğu ve bilgisayardan bilgisayara bu sürelerin farklılıklar gösterebileceği asla unutulmamalıdır.

KAYNAKLAR

Anderson, E., Bai, Z. ve Dongarra, J. (1992), *Generalized QR Factorization and Its Application*, Linear Algebra and Its Application, **162**, 243-271.

Anderson, E., Bai, Z., Bischof, C., Demmel, J., Dongarra, J., Du Croz, J., Greenbaum, A., Hammarling, S., McKenney, A., Ostrouchov, S. ve Sorensen, D. (1995), *LAPACK Users' Guide*, 2nd Edition, SIAM, Philadelphia.

Andrews, H. C. ve Kane, J. (1970), *Kronecker Matrices, Computer Implementation, and Generalized Spectra*, Journal of ACM, **17**(2), 260-268.

Baltagi, B. H. (2001), *Econometric Analysis of Panel Data*, John Wiley and Sons, 2nd Edition.

Belsley, D. A. (1992), *Paring 3SLS Calculations Down to Manageable Proportions*, Computer Science in Economics and Management, **5**, 157-169.

Björk, Å. (1994), *Solution of Augmented Linear Systems Using Orthogonal Factorizations*, BIT, **34**, 1–26.

Björk, Å. (1996), *Numerical Methods for Least Squares Problems*, SIAM, Philadelphia.

Davidson, R. ve MacKinnon, J. G. (2004), *Econometric Theory and Methods*, Oxford University Pres, USA.

Dhrymes, P. J. (1994), *Topics in Advanced Econometrics*, Linear and Nonlinear Simultaneous Equations, Springer-Verlag, New York.

Dinenis, E. ve Kontoghiorghes, E. J. (1997a), *Computing Solutions of SURE models with Linear Equality and Stochastic Constraints*, Journal of Mathematical Modelling and Scientific Computing, **8**, 234-239.

Dinenis, E. ve Kontoghiorghes, E. J. (1997b), *Computing 3SLS Solutions of Simultaneous Equation Models with A Possible Singular Variance-Covariance Matrix*, Computational Economics, **10**, 231-250.

Fausett, D. W., Fulton, C. T. ve Hashish, H. (1997), *Improved Paralel QR Method for Large Least Squares Problems Involving Kronecker Product*, Journal of Computational and Applied Mathematics, **78**, 63-78.

Fackler, P. L. (2005), *Notes on Matrix Calculus*, NC State University.

Foschi, P. ve Kontoghiorghes, E.J. (2003), *Estimation of VAR Models : Computational Aspects*, Computational Economics, **21**(1), 3-22.

Givens, J. W. (1954), *Numerical computation of the characteristic values of a real symmetric matrix*, Oak Ridge Report Number ORNL 1574 (physics).

Givens, J. W. (1958), *Computation of plane unitary rotations transforming a general matrix to triangular form*, J. SIAM, **6**(1), 26-50.

Golub, G. H. ve Van Loan, C. F. (1996), *Matrix Computations*, Johns Hopkins University Pres, Baltimore, Maryland, 3rd Edition.

Graham, A. (1981), *Kronecker Products and Matrix Calculus with Applications*, Halsted Press, John Wiley and Sons, New York.

Hammarling, S. (1976), *The Numerical Solution of the General Gauss–Markoff Linear Model*, in Mathematics in Signal Processing (Eds : Durrani T. S. ve ark.), Clarendon Press, Oxford.

Harville, D. A. (1997), *Matrix Algebra From A Statistician's Perspective*, Springer-Verlag, New York.

Henderson, H. V. ve Searle, S. R. (1981), *The vec-permutation Matrix, The vec Operator and Kronecker Products : A Review*, Linear Multilinear Algebra, **9**, 271-288.

Herstein, I. N. ve Winter, D. J. (1988), *Matrix Theory and Linear Algebra*, Mcmillan Publishing Company, New York.

Horn, R. A. ve Johnson, C. R. (1985), *Matrix Analysis*, Cambridge University Press, Cambridge.

Householder, A. S. (1958), *Unitary Triangularization of a Nonsymmetric Matrix*, Journal ACM, **5**(4), 339-342.

Judge, G. G., Griffiths, W. E., Hill, R. C., Lütkepohl, H. ve Lee, T. C. (1985), *The Theory and Practice of Econometrics*, Wiley Series in Probability and Mathematical Statistics, John Wiley and Sons, Second Edition.

Kmenta, J. (1986), *Elements of Econometrics*, Mcmillan Publishing Company, New York.

Kmenta, J. ve Gilbert, R. F. (1970), *Estimation of Seemingly Unrelated Regressions with Autoregressive Disturbances*, JASA, **65**(329), 186-197.

Kontoghiorghes, E. J. ve Dinenis, E. (1997), *Computing 3SLS Soutions of Simultaneous Equation Models with a Possible Singular Variance-Covariance Matrix*, Computational Economics, **10**, 231-250.

Kontoghiorghes, E. J. (2000a), *Inconsistencies and redundancies in SURE Models: Computational Aspects*, Computational Economics, **16**(1+2), 63-70.

Kontoghiorghes, E. J. (2000b), *Paralel Algorithms for Linear Models: Numerical Methods and Estimation Problems*, Advances in Computational Economics, **15**, Kluwer Academic Publishers, Boston, MA.

Kontoghiorghes, E. J. (2000c), *Parallel Strategies for Solving SURE Models with Variance Inequalities and Positivity of Correlations Constraints*, Computational Economics, **15**(1+2), 89-106.

Kourouklis, S. ve Paige, C. C. (1981), *A Constrained Least Squares Approach to The General Gauss-Markov Linear Model*, JASA, **76**(375), 620-625.

Lawson, C. L. ve Hanson, R. J. (1974), *Solving Least Square Problems*, Prentice-Hall Englewood Cliffs.

Lütkepohl, H. (1993), *Introduction to Multiple Time Series Analysis*, Springer-Verlag, Second Edition.

Paige, C. C. (1990), *Some Aspect of Generalized QR Factorizations, Reliable Numerical Computation*, (Eds : Cox, M. G. ve Hammerling, S. J.), 71-91, Claredon Press, Oxford.

Pollock, D. S. G (1979), *The Algebra of Econometrics*, Wiley Series in Probability and Mathematical Statistics, John Wiley and Sons.

Rao, C. R. ve Toutenburg, H. (1995), *Linear Models: Least Squares and Alternatives*, Springer Series in Statistics, Springer.

Regalia, P. A. ve Mitra, S. K. (1989), *Kronecker Products, Unitary Matrices and Signal Processing Applications*, SIAM, **31**(4), 586-613.

Schmidt, P. (1977), *Estimation of Seemingly Unrelated Regression with Unequal Numbers of Observations*, Journal of Econometrics, **5**, 365-377.

Schmidt, P. (1978), *A Note on The Estimation of Seemingly Unrelated Regression Systems*, Journal of Econometrics, **7**, 259-261.

Sharma, V. K. (1993), *Estimation of Seemingly Unrelated Regression with Unequal Numbers of Observation*, Sankhya: The Indian Journal of Statistics, Series B, **55**, 135-138.

Söderkvist, I. (1996), *On Algorithms for Generalized Least-Squares Problems with Ill-Conditioned Covariance Matrices*, Computational Statistics, **11**(3), 303-313.

Srivastava, V. K. ve Dwivedi, T. D. (1979), *Estimation of Seemingly Unrelated Regression Equations Models: A Brief Survey*, Journal of Econometrics, **10**, 15-32.

Srivastava, V. K. ve Giles, D. E. A. (1987), *Seemingly Unrelated Regression Equations Models: Estimation and Inference*, Statistics: Textbooks and Monographs, (Ed : Owen, D. B.), **80**, Marcel Dekker Inc., New York.

Srivastava, V. K. ve Tivari, R. (1978), *Efficiency of Two-Stage and Three-Stage Least Squares Estimators*, Econometrica, **46**(6), 1495-1498.

Theil, H. (1971), *Principles of Econometrics*, John Wiley & Sons, 1st Edition.

Trefethen, L. N. ve Bau, D. III (1997), *Numerical Linear Algebra*, Society for Industrial and Applied Mathematics, Philadelphia.

Turkington, D. A. (2000), *Generalised vec Operators and The Seemingly Unrelated Regression Equations Model with Vector Correlated Disturbances*, Journal of Econometrics, **99**, 225-253.

Zellner, A. (1962), *An Efficient Method of Estimating SUR and Tests for aggregation Bias*, Journal of the American Statistical Association, **57**, 348-368.

Zellner, A. (1963), *Estimators for Seemingly Unrelated Regression Equations: Some Exact Finite Sample Results*, JASA, **58**, 977-992.

Zellner, A. (1979), An Error Components Procedure (ECP) for Introducing Prior Information About Covariance Matrices and Analysis of Multivariate Regression Models, International Economic Review, **20**(3), 679-692.

Ek-1 Verilen bir $x \in \mathbb{R}^m$ sütun vektörü için Householder dönüşümünün hesaplanmasına ilişkin Matlab kodu

```
function [v,beta] = Householder(x)
%   AMAÇ : Verilen bir x sütun vektörü için Householder dönüşümünün
%                hesaplanması.
%   ------------------------------------------------------------------------------------------
%   KULLANIM :  [v,beta] = Householder(x)
%   ------------------------------------------------------------------------------------------
%   TANIMLAMALAR :
%     x  :  m boyutlu sütun vektöründe 1. değer haricindekileri SIFIR yapmak için;
%
%          v  :  m boyutlu Householder vektörü (v(1) = 1 olmak kaydıyla)...
%     beta  :  P(mxm) Householder matrisinin oluşturulması için
%              ( P = I(n)-beta*v*v' ) gereken katsayı...( beta=2/v'*v )
%   ------------------------------------------------------------------------------------------
m = length(x);
sigma = x(2:m)'*x(2:m);
v = [1;x(2:m)];
if sigma == 0
   beta = 0;
else
   mu = -norm(x);
   if x(1) <= 0;
      v(1) = x(1)-sign(x(1))*mu;
   else
      v(1) = -sigma/(x(1)+mu);
   end
   beta = 2*v(1)^2/(sigma+v(1)^2);
   v=v/v(1);
end
```

Ek-2 Verilen bir $A \in \mathbb{R}^{m \times n}$ matrisine Householder dönüşümü yardımıyla QR ayrışımının hesaplanmasına ilişkin Matlab kodu

```
function [Q,R] = QRhouse(A)
%   AMAÇ : Verilen bir mxn boyutlu bir A matrisine Householder dönüşümü
%              yardımıyla A = Q*R olacak şekilde QR ayrışımı(decomposition)
%              uygulamak.
%   ---------------------------------------------------------------------------------------------
%   KULLANIM :   [Q,R] = QRhouse(A)
%   ! Householder.m fonksiyon dosyası gereklidir...
%   ---------------------------------------------------------------------------------------------
%   TANIMLAMALAR :
%      A : mxn boyutlu matrisi için;
%             m > n   ise;   A = Q*R = [ R1 ]n
%                                         [  0  ]m-n
%                                            n
%
%             m <= n   ise;    A = Q*R = [ R1   R2 ]m
%                                              m    n-m
%
%      Q  : mxm boyutlu ortogonal matris.
%     R1 : mxm boyutlu üst üçgen bir matris.
%   ---------------------------------------------------------------------------------------------
[m,n]=size(A);
Q = eye(m);
if m > n
   P.house = zeros(n);
   for i=1:n
      x = A(i:m,i);
      [v,beta] = Householder(x);
      sigma = beta*v*v';
```

Ek-2 (Devam) Verilen bir $A \in \mathbb{R}^{m \times n}$ matrisine Householder dönüşümü yardımıyla QR ayrışımının hesaplanmasına ilişkin Matlab kodu

```
        P(i).house = eye(length(sigma))-sigma;
        w = beta*A(i:m,i:n)'*v;
        A(i:m,i:n) = A(i:m,i:n)-v*w';
    end
    for j=n:-1:1
        Q(j:m,j:m) = P(j).house*Q(j:m,j:m);
    end
else
    P.house = zeros(m-1);
    for i=1:m-1
        x = A(i:m,i);
        [v,beta] = Householder(x);
        sigma = beta*v*v';
        P(i).house = eye(length(sigma))-sigma;
        w = beta*A(i:m,i:n)'*v;
        A(i:m,i:n) = A(i:m,i:n)-v*w';
    end
    for j=m-1:-1:1
        Q(j:m,j:m) = P(j).house*Q(j:m,j:m);
    end
end
R = A;
```

Ek-3 a **ve** b **şeklinde verilen iki değerden istenilen değeri sıfır yapmak için kullanılan Givens rotasyonu hesaplanmasına ilişkin Matlab kodu**

```
function [c,s] = Givens(a,b)
%   AMAÇ : Givens rotasyonunun hesaplanması.
%   ------------------------------------------------------------------------------------------
%   KULLANIM :  [c,s] = Givens(a,b)
%   ------------------------------------------------------------------------------------------
%   TANIMLAMALAR :
%      a , b : Givens rotasyonunun uygulanacağı matris elemanları.
%
%      c = cos(teta)  ve  s : sin(teta)  olan rotasyon elemanları.(bazı teta için)
%
%                 [ c  s ]' [a] = [r]
%                 [-s  c ]  [b]   [0]
%   ------------------------------------------------------------------------------------------
if b == 0
   c = 1;
   s = 0;
elseif abs(b) > abs(a)
   r = -a/b;
   s = 1/sqrt(1+r^2);
   c = s*r;
else
   r = -b/a;
   c = 1/sqrt(1+r^2);
   s = c*r;
end
```

Ek-4 Verilen bir $\mathbf{A} \in \mathbb{R}^{m \times n}$ $(m \geq n)$ matrisine Givens rotasyon yöntemi yardımıyla QR ayrışımının hesaplanmasına ilişkin Matlab kodu

```
function [Q,R] = QRgivens(A)
%   AMAÇ : Verilen bir mxn (m>n) boyutlu bir A matrisine Givens rotasyon
%              yöntemi yardımıyla A = Q*R olacak şekilde QR ayrışımı
%              (decomposition) uygulamak.
%   ---------------------------------------------------------------------------------------------
%   KULLANIM :  [Q,R] = QRgivens(A)
%   ! Givens.m fonksiyon dosyası gereklidir...
%   ---------------------------------------------------------------------------------------------
%   TANIMLAMALAR:
%      A : mxn boyutlu matris.
%
%      Q : mxm boyutlu ortogonal matris.
%      R : mxn boyutlu üst üçgen bir matris.
%   ---------------------------------------------------------------------------------------------
[m,n]=size(A);
Q=eye(m);
for j=1:n
   for i=m:-1:j+1
      I=eye(length(A));
      [c,s]=Givens(A(i-1,j),A(i,j));
      P(i,j).givens=[c s;-s c];
      A(i-1:i,j:n)=P(i,j).givens'*A(i-1:i,j:n);
      I(i-1:i,i-1:i)=P(i,j).givens;
      Q=Q*I;
   end
end
R=A;
```

Ek-5 Verilen bir $x \in \mathbb{R}^n$ satır vektörü için Householder dönüşümünün hesaplanmasına ilişkin Matlab kodu

```
function [v,beta] = houseQL(x)
%   AMAÇ : Verilen bir x satır vektörü için Householder dönüşümünün
%               hesaplanması.
%   -----------------------------------------------------------------------------------------
%    KULLANIM :  [v,beta] = houseQL(x)
%   -----------------------------------------------------------------------------------------
%   TANIMLAMALAR :
%     x  :  n boyutlu satır vektöründe n. değer haricindekileri SIFIR yapmak için;
%
%        v  :  n boyutlu Householder vektörü (v(n) = 1 olmak kaydıyla)...
%    beta  :  P(nxn) Householder matrisinin oluşturulması için
%             ( P = I(n)-beta*v*v' ) gereken katsayı...( beta=2/v'*v )
%   -----------------------------------------------------------------------------------------
n = length(x);
sigma = x(1:n-1)'*x(1:n-1);
v = [x(1:n-1);1];
if sigma == 0
   beta = 0;
else
   mu = -norm(x);
   if x(n) <= 0;
      v(n) = x(n)-sign(x(n))*mu;
   else
      v(n) = -sigma/(x(n)+mu);
   end
   beta = 2*v(n)^2/(sigma+v(n)^2);
   v=v/v(n);
end
```

Ek-6 Verilen bir $\mathbf{B} \in \mathbb{R}^{n \times m}$ matrisine Householder dönüşümü yardımıyla QL ayrışımının hesaplanmasına ilişkin Matlab kodu

```
function [P,L] = QLhouse(B)
%   AMAÇ : Verilen bir B(nxm) matrisine Householder dönüşümü yardımıyla
%          B' = P*L olacak şekilde QL ayrışımı (decomposition) uygulamak.
%   -----------------------------------------------------------------------------------------
%   KULLANIM :   [P,L] = QLhouse(B)
%   ! houseQL.m fonksiyon dosyası gereklidir...
%   -----------------------------------------------------------------------------------------
%   B' = P*L  =>  (B')' = L'*P'  =>  B = R*Q
%         L : alt üçgen bir matris (L' = R üst üçgen bir matris)
%         P : ortogonal bir matris (P' = Q ortogonal bir matris)
%   Sonuçta;
%          B' = P*L şeklinde bir QL ayrışımı
%   veya  B  = R*Q şeklinde bir RQ ayrışımı yapılmış olur.
%   -----------------------------------------------------------------------------------------
%   TANIMLAMALAR :
%      B : nxm boyutlu matrisi için;
%          n <= m   ise;  Q*B' = L = [ 0  ]m-n   veya    B*Q' = R = [ 0    R1]m
%                                  [ L1 ]n                        n-m    m
%                                    n
%
%            n > m   ise;  Q*B' = L = [ L2   L1 ]m   veya   B*Q' = R = [ R2 ]m-n
%                                    n-m    m                         [ R1 ]n
%                                                                       n
%     P  :  mxm boyutlu ortogonal matris. (P' = Q)
%     L1 :  alt üçgen bir matris. (L1' = R1 : üst üçgen bir matris.)
%   -----------------------------------------------------------------------------------------
[n,m]=size(B);
Bt = B';
```

Ek-6 (Devam) Verilen bir $\mathbf{B} \in \mathbb{R}^{n \times m}$ matrisine Householder dönüşümü yardımıyla QL ayrışımının hesaplanmasına ilişkin Matlab kodu

```
P = eye(m);
if n < m
   Q.house = zeros(n);
   for i=n:-1:1
      x = Bt(1:m-n+i,i);
      [v,beta] = houseQL(x);
      sigma = beta*v*v';
      Q(n-i+1).house = eye(length(sigma))-sigma;
      w = beta*Bt(1:m-n+i,1:i)'*v;
      Bt(1:m-n+i,1:i) = Bt(1:m-n+i,1:i)-v*w';
   end
   for j=n:-1:1
      P(1:m-j+1,1:m-j+1) = Q(j).house*P(1:m-j+1,1:m-j+1);
   end
else
   Q.house = zeros(m-1);
   for i=n:-1:n-m+2
      x = Bt(1:m-n+i,i);
      [v,beta] = houseQL(x);
      sigma = beta*v*v';
      Q(n-i+1).house = eye(length(sigma))-sigma;
      w = beta*Bt(1:m-n+i,1:i)'*v;
      Bt(1:m-n+i,1:i) = Bt(1:m-n+i,1:i)-v*w';
   end
   for j=m-1:-1:1
      P(1:m-j+1,1:m-j+1) = Q(j).house*P(1:m-j+1,1:m-j+1);
   end
end
L = Bt;
```

Ek-7 QR ayrışımı için güncelleme yöntemi

Eşitlik (4.12)'de verilen QR güncelleme işlemini ele alalım ve

$$\mathbf{Q}^T \begin{pmatrix} \mathbf{A} \\ \mathbf{B} \end{pmatrix} = \begin{pmatrix} \mathbf{R} \\ \mathbf{0} \end{pmatrix}$$

şeklinde yeniden düzenleyelim. Buna göre;

$$\mathbf{A} \equiv \mathbf{A}^{(0)} = \begin{array}{c} \begin{array}{cccc} k_1 & k_2 & \dots & k_G \end{array} \\ \begin{pmatrix} \mathbf{A}_{1,1}^{(0)} & \mathbf{A}_{1,2}^{(0)} & \dots & \mathbf{A}_{1,G}^{(0)} \\ \mathbf{0} & \mathbf{A}_{2,2}^{(0)} & \dots & \mathbf{A}_{2,G}^{(0)} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \dots & \mathbf{A}_{G,G}^{(0)} \end{pmatrix} \end{array} \begin{array}{c} \\ k_1 \\ k_2 \\ \vdots \\ k_G \end{array}$$

ve

$$\mathbf{B} \equiv \mathbf{B}^{(0)} = \begin{array}{c} \begin{array}{cccc} k_1 & k_2 & \dots & k_G \end{array} \\ \begin{pmatrix} \mathbf{0} & \mathbf{B}_{1,2}^{(0)} & \dots & \mathbf{B}_{1,G}^{(0)} \\ \mathbf{0} & \mathbf{0} & \dots & \mathbf{B}_{2,G}^{(0)} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \dots & \mathbf{0} \end{pmatrix} \end{array} \begin{array}{c} \\ q_1 \\ q_2 \\ \vdots \\ q_G \end{array}$$

şeklinde ifade edilsin. Burada $k^{(G)} = \sum_{i=1}^{G} k_i$ ve $q^{(G)} = \sum_{i=1}^{G} q_i$ olmak kaydıyla $\mathbf{A}, \mathbf{R} \in \mathbb{R}^{k^{(G)} \times k^{(G)}}$ boyutlu blok üst üçgen matrisler, $\mathbf{B} \in \mathbb{R}^{q^{(G)} \times k^{(G)}}$ boyutlu kesin blok üst üçgen matris ve $\mathbf{Q}$ ise $k^{(G)} + q^{(G)}$ boyutlu ortogonal matristir.

QR ayrışım güncellemesi iki adımda gerçekleştirilir. Birinci adımda;

$$\tilde{\mathbf{Q}}_i^T \begin{pmatrix} \mathbf{A}_{i,i}^{(i-2)} \\ \\ \mathbf{B}_{1:i-1,i}^{(i-2)} \end{pmatrix} = \begin{pmatrix} \mathbf{A}_{i,i}^{(i-1)} \\ \\ \mathbf{0} \end{pmatrix} \begin{array}{c} k_i \\ \\ q^{(i-1)} \end{array} \qquad i = 2, \dots, G$$

olacak şekilde $\tilde{\mathbf{Q}}_i^T$ ortogonal matrisleri hesaplanır ki burada $q^{(i-1)} = \sum_{j=1}^{i-1} q_j$ 'dir. İkinci adımda ise birinci adımdan elde edilen ortogonal matris kullanılarak $\mathbf{A}$ ve $\mathbf{B}$ matrislerinin işlem gören satırlarındaki diğer elemanları için yeni değerler;

Ek-7 (Devam) QR ayrışımı için güncelleme yöntemi

$$\tilde{\mathbf{Q}}_i^T \begin{pmatrix} \mathbf{A}_{i,i+1:G}^{(i-2)} \\ \\ \mathbf{B}_{1:i-1,i+1:G}^{(i-2)} \end{pmatrix} = \begin{pmatrix} \mathbf{A}_{i,i+1:G}^{(i-1)} \\ \\ \mathbf{B}_{1:i-1,i+1:G}^{(i-1)} \end{pmatrix} \begin{matrix} k_i \\ \\ q^{(i-1)} \end{matrix}$$

şeklinde belirlenir.

Bu adımsal prosedür toplam $G-1$ defa tekrarlanır ve süreç sonunda $\mathbf{B}$ matrisinin tüm elemanları yok edilir veya sıfır yapılır. $G=4$ için QR ayrışım güncellemesi aşamaları aşağıdaki şekil ile gösterilebilir:

$$\text{Başlangıç Durumu} \quad \begin{bmatrix} \mathbf{A}_{1,1}^{(0)} & \mathbf{A}_{1,2}^{(0)} & \mathbf{A}_{1,3}^{(0)} & \mathbf{A}_{1,4}^{(0)} \\ \mathbf{0} & \mathbf{A}_{2,2}^{(0)} & \mathbf{A}_{2,3}^{(0)} & \mathbf{A}_{2,4}^{(0)} \\ \mathbf{0} & \mathbf{0} & \mathbf{A}_{3,3}^{(0)} & \mathbf{A}_{3,4}^{(0)} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{A}_{4,4}^{(0)} \\ \hline \mathbf{0} & \mathbf{B}_{1,2}^{(0)} & \mathbf{B}_{1,3}^{(0)} & \mathbf{B}_{1,4}^{(0)} \\ \mathbf{0} & \mathbf{0} & \mathbf{B}_{2,3}^{(0)} & \mathbf{B}_{2,4}^{(0)} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{B}_{3,4}^{(0)} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \end{bmatrix}
\qquad \text{1. Aşama} \quad \begin{bmatrix} \mathbf{A}_{1,1}^{(0)} & \mathbf{A}_{1,2}^{(0)} & \mathbf{A}_{1,3}^{(0)} & \mathbf{A}_{1,4}^{(0)} \\ \mathbf{0} & \mathbf{A}_{2,2}^{(1)} & \mathbf{A}_{2,3}^{(1)} & \mathbf{A}_{2,4}^{(1)} \\ \mathbf{0} & \mathbf{0} & \mathbf{A}_{3,3}^{(0)} & \mathbf{A}_{3,4}^{(0)} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{A}_{4,4}^{(0)} \\ \hline \mathbf{0} & \mathbf{0} & \mathbf{B}_{1,3}^{(1)} & \mathbf{B}_{1,4}^{(1)} \\ \mathbf{0} & \mathbf{0} & \mathbf{B}_{2,3}^{(0)} & \mathbf{B}_{2,4}^{(0)} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{B}_{3,4}^{(0)} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \end{bmatrix}$$

$$\text{2. Aşama} \quad \begin{bmatrix} \mathbf{A}_{1,1}^{(0)} & \mathbf{A}_{1,2}^{(0)} & \mathbf{A}_{1,3}^{(0)} & \mathbf{A}_{1,4}^{(0)} \\ \mathbf{0} & \mathbf{A}_{2,2}^{(1)} & \mathbf{A}_{2,3}^{(1)} & \mathbf{A}_{2,4}^{(1)} \\ \mathbf{0} & \mathbf{0} & \mathbf{A}_{3,3}^{(2)} & \mathbf{A}_{3,4}^{(2)} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{A}_{4,4}^{(0)} \\ \hline \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{B}_{1,4}^{(2)} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{B}_{2,4}^{(2)} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{B}_{3,4}^{(0)} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \end{bmatrix}
\qquad \text{3. Aşama} \quad \begin{bmatrix} \mathbf{A}_{1,1}^{(0)} & \mathbf{A}_{1,2}^{(0)} & \mathbf{A}_{1,3}^{(0)} & \mathbf{A}_{1,4}^{(0)} \\ \mathbf{0} & \mathbf{A}_{2,2}^{(1)} & \mathbf{A}_{2,3}^{(1)} & \mathbf{A}_{2,4}^{(1)} \\ \mathbf{0} & \mathbf{0} & \mathbf{A}_{3,3}^{(2)} & \mathbf{A}_{3,4}^{(2)} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{A}_{4,4}^{(3)} \\ \hline \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \end{bmatrix}$$

Şekil 7.1. $G = 4$ için QR ayrışım güncellemesi aşamaları

Ek-8 Sıradan doğrusal model haline getirilen görünüşte ilişkisiz regresyon modelinin QR ayrışımı kullanılarak çözümüne ilişkin Matlab kodu (Algoritma I)

```
function sonuc = girQR(dsay,y,x)
%   AMAÇ : Görünüşte ilişkisiz regresyon modeline ilişkin katsayı tahminlerinin
%              Householder dönüşümü yardımıyla QR ayrışımı kullanılarak etkin bir
%              şekilde hesaplanması.
%   -------------------------------------------------------------------------------------------
%   KULLANIM :   sonuc = girQR(dsay,y,x)
%   ! QRhouse.m ve sekhata.m   fonksiyon dosyaları gereklidir...
%   -------------------------------------------------------------------------------------------
%   TANIMLAMALAR :
%      dsay  :  denklem sayısı
%
%          y  :  modelde yer alan bağımlı değişken vektörleri
%  D İ K K A T ! Vektörlerin yapısal tanımlamalarını yaparken 'denk' uzantısını
%   kullanınız.
%            Örneğin; y(1).denk = y1;
%                         y(2).denk = y2;
%                              ...    vb.
%
%          x  :  modelde yer alan bağımsız değişken matrisleri
%   D İ K K A T ! Matrislerin yapısal tanımlamalarını yaparken 'denk uzantısını
%   kullanınız.
%            Örneğin; x(1).denk = [1 x1 x3];
%                         x(2).denk = [1 x1 x2 x4];
%                             ...    vb.
%
%   NOT : Bağımsız değişken matrislerinin herbirine sabit terim için uygun
%           boyutta 1 vektörü eklenmelidir!..
%    -------------------------------------------------------------------------------------------
```

Ek-8 (Devam) Sıradan doğrusal model haline getirilen görünüşte ilişkisiz regresyon modelinin QR ayrışımı kullanılarak çözümüne ilişkin Matlab kodu (Algoritma I)

```
%   ÇIKTI DEĞERLERİ :
%   sonuc.metod        :  'Görünüşte İlişkisiz Regresyon'
%   sonuc.dsay         :  denklem sayısı
%   sonuc.gsay         :  herbir denklemdeki gözlem sayısı
%         D İ K K A T ! Tüm Denklemlerdeki Gözlem Sayıları Eşit Olmalıdır!...
%   sonuc(denk).bdsay     :  herbir denklemdeki bağımsız değişken sayısı
%   sonuc(denk).beta      :  herbir denklemin katsayı tahmin vektörü
%   sonuc(denk).artik     :  herbir denklemin artık vektörü
%   sonuc(denk).y         : herbir denklemin y bağımlı değişken vektörü
%   sonuc(denk).ytahmin   :  herbir denklemin y tahmin vektörü
%   sonuc(denk).skare     : herbir denklem için hata terimi varyansı(e'e/gsay)
%   sonuc.sigma           :  denklemler arası kovaryans matrisi (sig(i,j))
%   sonuc.ccorr           : denklemler arası çapraz korelasyon matrisi
%   sonuc.mrkare          :  Gir model r-karesi
%   ------------------------------------------------------------------------------------------

if nargin < 3
   error('Girdi değişkenlerinin sayısı hatalı');
end

for i=1:dsay
   sonuc(i).metod = 'Görünüşte İlişkisiz Regresyon';
   sonuc(i).dsay = dsay;
end
```

Ek-8 (Devam) Sıradan doğrusal model haline getirilen görünüşte ilişkisiz regresyon modelinin QR ayrışımı kullanılarak çözümüne ilişkin Matlab kodu (Algoritma I)

```
%  Tanımlamalardaki yapısal hataların kontrolü...
if ~isstruct(y)
    error('y (bağımlı değişken) vektörleri girişi yapısal olarak tanımlanmalı');
end

if ~isstruct(x)
    error('x (bağımsız değişkenler) matrisleri girişi yapısal olarak tanımlanmalı');
end

%  Yapısal tanımlamalardaki denk uzantılarının kontrolü...
knt = fieldnames(y);
if (strcmp(knt,'denk') ~= 1)
    error('y vektörlerinin tanım uzantıları için (denk) kullanınız.');
end

knt = fieldnames(x);
if (strcmp(knt,'denk') ~= 1)
    error('x matrislerinin tanım uzantıları için (denk) kullanınız.');
end

for i=1:dsay
    sonuc(i).gsay = size(y(i).denk,1);
end
gsay = size(y(1).denk,1);

%  Tüm denklemlerdeki gözlem sayılarının eşitliğinin kontrolü...
gsayy = zeros(dsay,1);
gsayx = zeros(dsay,1);
```

Ek-8 (Devam) Sıradan doğrusal model haline getirilen görünüşte ilişkisiz regresyon modelinin QR ayrışımı kullanılarak çözümüne ilişkin Matlab kodu (Algoritma I)

```
for i=1:dsay
   gsayy(i) = size(y(i).denk,1);
    %   y vektörleri oluşturuluyor...
   sonuc(i).y = y(i).denk;
   %   herbir denklemin bağımsız değişken sayıları oluşturuluyor...
   [gsayx(i) sonuc(i).bdsay] = size(x(i).denk);
end

tst = find(gsayy ~= gsayx);
if length(tst) ~= 0
   error('Tüm denklemlerdeki gözlem sayıları eşit olmalıdır!...');
end

%  SEK kullanılarak denklemlerin hata vektörleri oluşturuluyor...
emat = zeros(gsay,dsay);
for i=1:dsay
   emat(:,i) = sekhata(y(i).denk,x(i).denk);
end

%  Başlangıç varyans-kovaryans (sigma) matrisi oluşturuluyor...
sigma = zeros(dsay,dsay);
for i=1:dsay
   for j=i:dsay
      %  Uygun (Feasible) Genelleştirilmiş En Küçük Kareler ile başlangıç
      %  Varyans-Kovaryans matrisinin tahminlenmesi
      %  William H. Greene, 1997, s 676
      sigma(i,j) = (emat(:,i)'*emat(:,j))/gsay;
```

Ek-8 (Devam) Sıradan doğrusal model haline getirilen görünüşte ilişkisiz regresyon modelinin QR ayrışımı kullanılarak çözümüne ilişkin Matlab kodu (Algoritma I)

```
        % Serbestlik derecesi düzeltmesi ile Var-Kov matris bileşenleri...
          % sigma(i,j) = (emat(:,i)'*emat(:,j))/sqrt((gsay-sonuc(i).bdsay)*(gsay-
sonuc(j).bdsay));
        if j > 1
            sigma(j,i) = sigma(i,j);
        end
    end
end

% Sigma matrisinin Cholesky faktörizasyonu hesaplanıyor...
C = chol(sigma);
Cinv = inv(C);
Ybar = horzcat(y.denk)*Cinv';

% Sıradan Doğrusal Model haline getirilen denklemlerin QR Ayrışımı ile
% çözümlemesi yapılıyor...
for i=1:dsay
    % Herbir X matrisinin QR ayrışımları hesaplanıyor...
    [Q(i).qr,R(i).qr] = QRhouse(x(i).denk);
    % R üst üçgen matrisleri biçimlendiriliyor...
    R(i).qr = R(i).qr(1:sonuc(i).bdsay,1:sonuc(i).bdsay);

    % Q ortogonal matrisleri tekrar biçimlendiriliyor...
    Qtilda(i).qr = Q(i).qr(:,1:sonuc(i).bdsay);
    Qsapka(i).qr = Q(i).qr(:,sonuc(i).bdsay+1:gsay);
```

Ek-8 (Devam) Sıradan doğrusal model haline getirilen görünüşte ilişkisiz regresyon modelinin QR ayrışımı kullanılarak çözümüne ilişkin Matlab kodu (Algoritma I)

```
    %  y vektör bileşenleri hesaplanıyor...
    ybar(i).tilda = Qtilda(i).qr'*Ybar(:,i);
    ybar(i).sapka = Qsapka(i).qr'*Ybar(:,i);
end

%  Elde edilen y vektör bileşenleri için Vec operatör işlemi uygulanıyor...
vecybar.tilda = vertcat(ybar.tilda);
vecybar.sapka = vertcat(ybar.sapka);

%  Wtilda blok matrisi oluşturuluyor...
for i=1:dsay
    for j=1:dsay
        if i<j
            Wtld = Cinv(i,j).*Qtilda(i).qr'*x(j).denk;
        elseif i==j
            Wtld = Cinv(i,i).*R(i).qr;
        else
            Wtld = zeros(sonuc(i).bdsay);
        end
        Wtilda(j).sutun = Wtld;
    end
    Wtilda(i).satir = [Wtilda.sutun];
end
Wtilda = vertcat(Wtilda.satir);
```

Ek-8 (Devam) Sıradan doğrusal model haline getirilen görünüşte ilişkisiz regresyon modelinin QR ayrışımı kullanılarak çözümüne ilişkin Matlab kodu (Algoritma I)

```
% Wsapka blok matrisi oluşturuluyor...
for i = 1:dsay
   for j = 1:dsay
      if i<j
         Wspk = Cinv(i,j)*Qsapka(i).qr'*x(j).denk;
      else
         Wspk = zeros(gsay-sonuc(i).bdsay,sonuc(i).bdsay);
      end
      Wsapka(j).sutun = Wspk;
   end
   Wsapka(i).satir = [Wsapka.sutun];
end
Wsapka = vertcat(Wsapka.satir);

% Tüm sistemdeki ardışık olarak toplam bağımsız değişken sayısı elde ediliyor...
K(:,1) = 0;
for i=1:dsay
   K(:,i+1) = K(:,i) + sonuc(i).bdsay;
end;

% QR ayrışımı güncellemesi yapılıyor...
WA(:,1).uqr = Wtilda;
WB(:,1).uqr = Wsapka;

for i=2:dsay
   WA(:,i).uqr = WA(:,i-1).uqr(K(:,i)+1:K(:,i+1),K(:,i)+1:K(:,i+1));
   WB(:,i).uqr = WB(:,i-1).uqr(1:(i-1)*gsay-K(:,i),K(:,i)+1:K(:,i+1));
```

Ek-8 (Devam) Sıradan doğrusal model haline getirilen görünüşte ilişkisiz regresyon modelinin QR ayrışımı kullanılarak çözümüne ilişkin Matlab kodu (Algoritma I)

```
% W matrisleri için QR ayrışımı yapılıyor...
[Qqr(:,i-1).uqr,Rqr(:,i-1).uqr] = QRhouse([WA(:,i).uqr;WB(:,i).uqr]);

% Elde edilen Qqr ortogonal matrisi WA ve WB matrislerinin uygun blok
% elemanlarına çarpılıyor...
AA = WA(:,i-1).uqr(K(:,i)+1:K(:,i+1),K(:,i+1)+1:end);
BB = WB(:,i-1).uqr(1:(i-1)*gsay-K(:,i),K(:,i+1)+1:end);

QtAB = Qqr(:,i-1).uqr'*[AA;BB];

% Elde edilen Qqr ortogonal matrisi vecybar vektörünün uygun elemanlarına
% çarpılıyor...
YA = vecybar(i-1).tilda(K(:,i)+1:K(:,i+1),:);
YB = vecybar(i-1).sapka(1:(i-1)*gsay-K(:,i));

Qtybar = Qqr(:,i-1).uqr'*[YA;YB];

% Matrislerin belirli blok elemanları değiştirilerek yeniden yapılandırılıyor...
WA(:,i-1).uqr(K(:,i)+1:K(:,i+1),K(:,i+1)+1:end) = QtAB(1:K(:,i+1)-K(:,i),:);
WA(:,i).uqr = WA(:,i-1).uqr;

WB(:,i-1).uqr(1:(i-1)*gsay-K(:,i),K(:,i+1)+1:end)=QtAB(K(:,i+1)-
K(:,i)+1:end,:);
WB(:,i).uqr = WB(:,i-1).uqr;

% vecybar vektörünün belirli elemanları değiştirilerek yeniden
% yapılandırılıyor...
```

Ek-8 (Devam) Sıradan doğrusal model haline getirilen görünüşte ilişkisiz regresyon modelinin QR ayrışımı kullanılarak çözümüne ilişkin Matlab kodu (Algoritma I)

```
    vecybar(i-1).tilda(K(:,i)+1:K(:,i+1),:) = Qtybar(1:K(:,i+1)-K(:,i),:);
    vecybar(i).tilda = vecybar(i-1).tilda;

    vecybar(i-1).sapka(1:(i-1)*gsay-K(:,i)) = Qtybar(K(:,i+1)-K(:,i)+1:end,:);
    vecybar(i).sapka = vecybar(i-1).sapka;
end

Rb = WA(1,i).uqr;
Yb = vecybar(1,i).tilda;

%  Beta vektörü oluşturuluyor...
beta = inv(Rb)*Yb;

%  Herbir denklem için tahmin edilen beta vektörleri ayrıştırılıp kaydediliyor...
topbdsay = 0;
for i=1:dsay
    topbdsay = topbdsay + sum(sonuc(i).bdsay);
    betatah(i).beta = beta(topbdsay-(sonuc(i).bdsay-1):topbdsay);
    sonuc(i).beta = betatah(i).beta;
end

%  Görünüşte ilişkisiz regresyon artık vektörlerinin hesaplanması...
emat = zeros(gsay,dsay);
for i=1:dsay
    sonuc(i).artik = y(i).denk - x(i).denk*sonuc(i).beta;
    emat(:,i) = sonuc(i).artik;
end
```

Ek-8 (Devam) Sıradan doğrusal model haline getirilen görünüşte ilişkisiz regresyon modelinin QR ayrışımı kullanılarak çözümüne ilişkin Matlab kodu (Algoritma I)

```
% Görünüşte ilişkisiz regresyon varyans-kovaryans(sigma) matrisleri
% hesaplanıyor...
sigma = zeros(dsay,dsay);
for i=1:dsay
   for j=i:dsay
      sigma(i,j) = (emat(:,i)'*emat(:,j))/gsay;
      if j > 1
         sigma(j,i) = sigma(i,j);
      end
   end
end

% Görünüşte ilişkisiz regresyon Korelasyon matrisi hesaplanıyor...
for i=1:dsay
  % denklemler arası çapraz korelasyon matrisi.
  sonuc(i).ccorr = corrcoef(emat);
end

% Gir model r-karesinin elde edilmesi
% William H. Greene, 1993, s 490
for i=1:dsay
   yort(:,i) = mean(sonuc(i).y)
end
for i=1:dsay
   for j=1:dsay
      deger=(1/gsay)*(y(i).denk-yort(:,i)*ones(gsay,1)).*(y(j).denk-
yort(:,j)*ones(gsay,1));
      syy(i,j) = sum(deger);
```

Ek-8 (Devam) Sıradan doğrusal model haline getirilen görünüşte ilişkisiz regresyon modelinin QR ayrışımı kullanılarak çözümüne ilişkin Matlab kodu (Algoritma I)

```
        if j > i
            syy(j,i) = syy(i,j);
        end
    end
end
mrkare = 1 - dsay/trace(inv(sigma)*syy);

%  Herbir denkleme ilişkin sonuçların oluşturulması…
for i=1:dsay
    sonuc(i).sigma = sigma;     %  denklemler arası Varyans-Kovaryans matrisi.
    sonuc(i).mrkare = mrkare;
    sonuc(i).ytahmin = sonuc(i).y - x(i).denk*sonuc(i).beta;
    sonuc(i).artik = sonuc(i).y - sonuc(i).ytahmin;
    hkt = sonuc(i).artik'*sonuc(i).artik;
    sonuc(i).skare = hkt/gsay;     %  hata terimi varyansı
end
```

Ek-9 RQ ayrışımı için güncelleme yöntemi

(4.19a) eşitliği ile verilen RQ ayrışımı güncelleme işlemini ele alalım ve

$$\begin{pmatrix}\mathbf{C} & \mathbf{D}\end{pmatrix}\mathbf{P} = \begin{pmatrix}\mathbf{0} & \mathbf{R}\end{pmatrix}$$

şeklinde yeniden düzenleyelim. Bölüm 2.5'de değinildiği üzere RQ ayrışımı yerine QL ayrışımı şeklinde problemi ele almak kullanılacak bilgisayar algoritmasının etkinliğini artıracağından dolayı,

$$\begin{pmatrix}\mathbf{C} & \mathbf{D}\end{pmatrix}\mathbf{P} = \begin{pmatrix}\mathbf{0} & \mathbf{R}\end{pmatrix} \;\Rightarrow\; \mathbf{P}^T\begin{pmatrix}\mathbf{C}^T \\ \mathbf{D}^T\end{pmatrix} = \begin{pmatrix}\mathbf{0}^T \\ \mathbf{R}^T\end{pmatrix} \;\Rightarrow\; \begin{pmatrix}\mathbf{C}^T \\ \mathbf{D}^T\end{pmatrix} = \mathbf{P}\begin{pmatrix}\mathbf{0} \\ \mathbf{L}\end{pmatrix}$$

olarak tekrar düzenlenir. Burada;

$$\mathbf{C} \equiv \mathbf{C}^{(0)} = \begin{array}{c} \begin{array}{cccc} k_1 & k_2 & \dots & k_G \end{array} \\ \begin{pmatrix} \mathbf{0} & \mathbf{C}_{1,2}^{(0)} & \dots & \mathbf{C}_{1,G}^{(0)} \\ \mathbf{0} & \mathbf{0} & \dots & \mathbf{C}_{2,G}^{(0)} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \dots & \mathbf{0} \end{pmatrix} \end{array} \begin{array}{c} \\ q_1 \\ q_2 \\ \vdots \\ q_G \end{array}$$

ve

$$\mathbf{D} \equiv \mathbf{D}^{(0)} = \begin{array}{c} \begin{array}{cccc} q_1 & q_2 & \cdots & q_G \end{array} \\ \begin{pmatrix} \mathbf{D}_{1,1}^{(0)} & \mathbf{D}_{1,2}^{(0)} & \dots & \mathbf{D}_{1,G}^{(0)} \\ \mathbf{0} & \mathbf{D}_{2,2}^{(0)} & \dots & \mathbf{D}_{2,G}^{(0)} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \dots & \mathbf{D}_{G,G}^{(0)} \end{pmatrix} \end{array} \begin{array}{c} \\ q_1 \\ q_2 \\ \vdots \\ q_G \end{array}$$

şeklinde ifade edilir. Burada $k^{(s)} = \sum_{i=s}^{G} k_i$ ve $q^{(s)} = \sum_{i=s}^{G} q_i$ olmak kaydıyla $\mathbf{D}, \mathbf{R} \in \mathbb{R}^{q^{(1)} \times q^{(1)}}$ boyutlu blok üst üçgen matrisler, $\mathbf{C} \in \mathbb{R}^{q^{(1)} \times k^{(1)}}$ boyutlu kesin blok üst üçgen matris ve $\mathbf{P}$ ise $k^{(1)} + q^{(1)}$ boyutlu ortogonal matristir.

QL ayrışımı için güncelleme işlemi, QR güncellemesine benzer şekilde iki adımda gerçekleştirilir. Birinci adımda;

$$\tilde{\mathbf{P}}_j^T \begin{pmatrix} \mathbf{C}_{j:G,j-1}^{(G-j)} \\ \\ \mathbf{D}_{j-1,j-1}^{(G-j)} \end{pmatrix} = \begin{pmatrix} \mathbf{0} \\ \\ \mathbf{D}_{j-1,j-1}^{(G-j+1)} \end{pmatrix} \begin{array}{c} k^{(j)} \\ \\ q_{j-1} \end{array} \qquad j = G, G-1, \dots, 2$$

Ek-9 (Devam) RQ ayrışımı için güncelleme yöntemi

olacak şekilde $\tilde{\mathbf{P}}_j^T$ ortogonal matrisleri hesaplanır ki burada $\mathbf{C}$ ve $\mathbf{D}$ sırasıyla $\mathbf{C}^T$ ve $\mathbf{D}^T$'yi ifade etmektedir ve $q^{(j)} = \sum_{i=j}^{G} q_i$'dir. İkinci adımda ise birinci adımdan elde edilen ortogonal matris kullanılarak $\mathbf{C}$ ve $\mathbf{D}$ matrislerinin işlem gören satırlarındaki diğer elemanları için yeni değerler ise;

$$\tilde{\mathbf{P}}_j^T \begin{pmatrix} \mathbf{C}_{j:G,1:j-2}^{(G-j)} \\ \\ \mathbf{D}_{j-1,1:j-2}^{(G-j)} \end{pmatrix} = \begin{pmatrix} \mathbf{C}_{j:G,1:j-2}^{(G-j+1)} \\ \\ \mathbf{D}_{j-1,1:j-2}^{(G-j+1)} \end{pmatrix} \begin{matrix} k^{(j)} \\ \\ q_{j-1} \end{matrix}$$

şeklinde belirlenir.

Bu adımsal prosedür toplam $G-1$ defa tekrarlanır ve süreç sonunda $\mathbf{C}$ matrisinin tüm elemanları yok edilir veya sıfır yapılır. $G=4$ için QR ayrışım güncellemesi aşamaları aşağıdaki şekil ile gösterilebilir:

Başlangıç Durumu

$$\begin{bmatrix} \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{C}_{2,1}^{(0)} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{C}_{3,1}^{(0)} & \mathbf{C}_{3,2}^{(0)} & \mathbf{0} & \mathbf{0} \\ \mathbf{C}_{4,1}^{(0)} & \mathbf{C}_{4,2}^{(0)} & \mathbf{C}_{4,3}^{(0)} & \mathbf{0} \\ \hline \mathbf{D}_{1,1}^{(0)} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{D}_{2,1}^{(0)} & \mathbf{D}_{2,2}^{(0)} & \mathbf{0} & \mathbf{0} \\ \mathbf{D}_{3,1}^{(0)} & \mathbf{D}_{3,2}^{(0)} & \mathbf{D}_{3,3}^{(0)} & \mathbf{0} \\ \mathbf{D}_{4,1}^{(0)} & \mathbf{D}_{4,2}^{(0)} & \mathbf{D}_{4,3}^{(0)} & \mathbf{D}_{4,4}^{(0)} \end{bmatrix}$$

1. Aşama

$$\begin{bmatrix} \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{C}_{2,1}^{(0)} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{C}_{3,1}^{(0)} & \mathbf{C}_{3,2}^{(0)} & \mathbf{0} & \mathbf{0} \\ \mathbf{C}_{4,1}^{(1)} & \mathbf{C}_{4,2}^{(1)} & \mathbf{0} & \mathbf{0} \\ \hline \mathbf{D}_{1,1}^{(0)} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{D}_{2,1}^{(0)} & \mathbf{D}_{2,2}^{(0)} & \mathbf{0} & \mathbf{0} \\ \mathbf{D}_{3,1}^{(1)} & \mathbf{D}_{3,2}^{(1)} & \mathbf{D}_{3,3}^{(1)} & \mathbf{0} \\ \mathbf{D}_{4,1}^{(0)} & \mathbf{D}_{4,2}^{(0)} & \mathbf{D}_{4,3}^{(0)} & \mathbf{D}_{4,4}^{(0)} \end{bmatrix}$$

Şekil 9.1. $G=4$ için RQ ayrışım güncellemesi aşamaları

Ek-9 (Devam) RQ ayrışımı için güncelleme yöntemi

2. Aşama

$$\begin{bmatrix} \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{C}_{2,1}^{(0)} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{C}_{3,1}^{(2)} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{C}_{4,1}^{(2)} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \hline \mathbf{D}_{1,1}^{(0)} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{D}_{2,1}^{(2)} & \mathbf{D}_{2,2}^{(2)} & \mathbf{0} & \mathbf{0} \\ \mathbf{D}_{3,1}^{(1)} & \mathbf{D}_{3,2}^{(1)} & \mathbf{D}_{3,3}^{(1)} & \mathbf{0} \\ \mathbf{D}_{4,1}^{(0)} & \mathbf{D}_{4,2}^{(0)} & \mathbf{D}_{4,3}^{(0)} & \mathbf{D}_{4,4}^{(0)} \end{bmatrix}$$

3. Aşama

$$\begin{bmatrix} \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \hline \mathbf{D}_{1,1}^{(3)} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{D}_{2,1}^{(2)} & \mathbf{D}_{2,2}^{(2)} & \mathbf{0} & \mathbf{0} \\ \mathbf{D}_{3,1}^{(1)} & \mathbf{D}_{3,2}^{(1)} & \mathbf{D}_{3,3}^{(1)} & \mathbf{0} \\ \mathbf{D}_{4,1}^{(0)} & \mathbf{D}_{4,2}^{(0)} & \mathbf{D}_{4,3}^{(0)} & \mathbf{D}_{4,4}^{(0)} \end{bmatrix}$$

Şekil 9.1. (Devam) $G = 4$ için RQ ayrışım güncellemesi aşamaları

Ek-10 Genelleştirilmiş doğrusal en küçük kareler problemi haline getirilen görünüşte ilişkisiz regresyon modelinin genelleştirilmiş QR ayrışımı kullanılarak çözümüne ilişkin Matlab kodu (Algoritma II)

```
function sonuc = girGQR(dsay,y,x)
%   AMAÇ : Görünüşte ilişkisiz regresyon modellerine ilişkin katsayıların
%              genelleştirilmiş QR ayrışımı kullanılarak etkin tahminlerini elde
%              etmek.
%   -----------------------------------------------------------------------------------------
%   KULLANIM :   sonuc = girGQR(dsay,y,x)
%   ! QRhouse.m , QLhouse.m ve sekhata.m fonksiyon dosyaları gereklidir...
%   -----------------------------------------------------------------------------------------
%   TANIMLAMALAR:
%      dsay  :  denklem sayısı
%
%         y   :  modelde yer alan bağımlı değişken vektörleri
%  D İ K K A T ! Vektörlerin yapısal tanımlamalarını yaparken 'denk' uzantısını
%   kullanınız.
%             Örneğin; y(1).denk = y1;
%                          y(2).denk = y2;
%                               ...   vb.
%
%         x   :  modelde yer alan bağımsız değişken matrisleri
%   D İ K K A T ! Matrislerin yapısal tanımlamalarını yaparken 'denk'uzantısını
%   kullanınız.
%             Örneğin; x(1).denk = [1 x1 x3];
%                          x(2).denk = [1 x1 x2 x4];
%                               ...   vb.
%   NOT : Bağımsız değişken matrislerinin herbirine sabit terim için uygun
%            boyutta 1 vektörü eklenmelidir!...
%   -----------------------------------------------------------------------------------------
```

Ek-10 (Devam) Genelleştirilmiş doğrusal en küçük kareler problemi haline getirilen görünüşte ilişkisiz regresyon modelinin genelleştirilmiş QR ayrışımı kullanılarak çözümüne ilişkin Matlab kodu (Algoritma II)

```
% ÇIKTI DEĞERLERİ :
% sonuc.metod          : 'Görünüşte İlişkisiz Regresyon'
% sonuc.dsay           : denklem sayısı
% sonuc.gsay           : herbir denklemdeki gözlem sayısı
%        D İ K K A T ! Tüm Denklemlerdeki Gözlem Sayıları Eşit Olmalıdır!...
% sonuc(denk).bdsay    : herbir denklemdeki bağımsız değişken sayısı
% sonuc(denk).beta     : herbir denklemin katsayı tahmin vektörü
% sonuc(denk).artik    : herbir denklemin artık vektörü
% sonuc(denk).y        : herbir denklemin y bağımlı değişken vektörü
% sonuc(denk).ytahmin  : herbir denklemin y tahmin vektörü
% sonuc(denk).skare    : herbir denklem için hata terimi varyansı(e'e/gsay)
% sonuc.sigma          : denklemler arası kovaryans matrisi (sig(i,j))
% sonuc.ccorr          : denklemler arası çapraz korelasyon matrisi
% sonuc.mrkare         : Gir model r-karesi
% ---------------------------------------------------------------------------------------------

if nargin < 3
   error('Girdi değişkenlerinin sayısı hatalı');
end

for i=1:dsay
   sonuc(i).metod = 'Görünüşte İlişkisiz Regresyon';
   sonuc(i).dsay = dsay;
end
```

Ek-10 (Devam) Genelleştirilmiş doğrusal en küçük kareler problemi haline getirilen görünüşte ilişkisiz regresyon modelinin genelleştirilmiş QR ayrışımı kullanılarak çözümüne ilişkin Matlab kodu (Algoritma II)

```
%  Tanımlamalardaki yapısal hataların kontrolü...
if ~isstruct(y)
    error('y (bağımlı değişken) vektörleri girişi yapısal olarak tanımlanmalı');
end

if ~isstruct(x)
    error('x (bağımsız değişkenler) matrisleri girişi yapısal olarak tanımlanmalı');
end

%  Yapısal tanımlamalardaki denk uzantılarının kontrolü...
knt = fieldnames(y);
if (strcmp(knt,'denk') ~= 1)
    error('y vektörlerinin tanım uzantıları için (denk) kullanınız.');
end

knt = fieldnames(x);
if (strcmp(knt,'denk') ~= 1)
    error('x matrislerinin tanım uzantıları için (denk) kullanınız.');
end

for i=1:dsay
    sonuc(i).gsay = size(y(i).denk,1);
end
gsay = size(y(1).denk,1);

%  Tüm denklemlerdeki gözlem sayılarının eşitliğinin kontrolü...
gsayy = zeros(dsay,1);
gsayx = zeros(dsay,1);
```

Ek-10 (Devam) Genelleştirilmiş doğrusal en küçük kareler problemi haline getirilen görünüşte ilişkisiz regresyon modelinin genelleştirilmiş QR ayrışımı kullanılarak çözümüne ilişkin Matlab kodu (Algoritma II)

```
for i=1:dsay
    gsayy(i) = size(y(i).denk,1);
    %   y vektörleri oluşturuluyor...
    sonuc(i).y = y(i).denk;
    %   herbir denklemin bağımsız değişken sayıları oluşturuluyor...
    [gsayx(i) sonuc(i).bdsay] = size(x(i).denk
end

tst = find(gsayy ~= gsayx);
if length(tst) ~= 0
    error('Tüm denklemlerdeki gözlem sayıları eşit olmalıdır!...');
end

%   SEK kullanılarak denklemlerin hata vektörleri oluşturuluyor...
emat = zeros(gsay,dsay);
for i=1:dsay
    emat(:,i) = sekhata(y(i).denk,x(i).denk);
end

% Başlangıç varyans-kovaryans (sigma) matrisi oluşturuluyor...
for i=1:dsay
    for j=i:dsay
        %  Uygun (Feasible) Genelleştirilmiş En Küçük Kareler ile başlangıç
        %  Varyans-Kovaryans matrisinin tahminlenmesi
        %  William H. Greene, 1997, s 676
        sigma(i,j) = (emat(:,i)'*emat(:,j))/gsay;
```

Ek-10 (Devam) Genelleştirilmiş doğrusal en küçük kareler problemi haline getirilen görünüşte ilişkisiz regresyon modelinin genelleştirilmiş QR ayrışımı kullanılarak çözümüne ilişkin Matlab kodu (Algoritma II)

```
        % Serbestlik derecesi düzeltmesi ile Var-Kov matris bileşenleri...
          % sigma(i,j) = (emat(:,i)'*emat(:,j))/sqrt((gsay-sonuc(i).bdsay)*(gsay-
sonuc(j).bdsay));
        if j > 1
            sigma(j,i) = sigma(i,j);
        end
    end
end

Igsay=eye(gsay);
% Sigma matrisinin Cholesky faktörizasyonu hesaplanıyor...
C = chol(sigma);
Omega = kron(C,Igsay);

% I. AŞAMA : Sistemdeki herbir denklemin QR Ayrışımı ile çözümlemesi
%                 yapılıyor...
for i=1:dsay
    sonuc(i).bdsay = size(x(i).denk,2);

    %   Herbir X matrisinin QR ayrışımları hesaplanıyor...
    [Q(i).qr,R(i).qr] = QRhouse(x(i).denk);

    %   R üst üçgen matrisleri biçimlendiriliyor...
    R(i).qr = R(i).qr(1:sonuc(i).bdsay,1:sonuc(i).bdsay);

    %   Q ortogonal matrisleri tekrar biçimlendiriliyor...
    Qtilda(i).qr = Q(i).qr(:,1:sonuc(i).bdsay);
    Qsapka(i).qr = Q(i).qr(:,sonuc(i).bdsay+1:gsay);
```

Ek-10 (Devam) Genelleştirilmiş doğrusal en küçük kareler problemi haline getirilen görünüşte ilişkisiz regresyon modelinin genelleştirilmiş QR ayrışımı kullanılarak çözümüne ilişkin Matlab kodu (Algoritma II)

```
    %  y vektör bileşenleri hesaplanıyor...
    y(i).tilda = Qtilda(i).qr'*y(i).denk;
    y(i).sapka = Qsapka(i).qr'*y(i).denk;
end

%  Q matrisi oluşturuluyor...
Q = [blkdiag(Qtilda.qr),blkdiag(Qsapka.qr)];

%  Tüm sistemdeki ardışık olarak toplam bağımsız değişken sayısı elde ediliyor...
K(:,1) = 0;
for i=1:dsay
    K(:,i+1) = K(:,i) + sonuc(i).bdsay;
end

%  II. AŞAMA : RQ ayrışımı yapılıyor...
WtildaQ = Q'*Omega*Q;

%  RQ (veya QL) ayrışımı güncellemesi yapılıyor...
WAA(:,dsay).urq=WtildaQ(1:K(:,dsay+1),1:K(:,dsay+1));
      %  WAA üst üçgen bir matris.
WAB(:,dsay).urq=WtildaQ(1:K(:,dsay+1),K(:,dsay+1)+1:end);
      %   WAB kesin üst üçgen bir matris.
WBA(:,dsay).urq=WtildaQ(K(:,dsay+1)+1:end,1:K(:,dsay+1));
      % WBA kesin üst üçgen bir matris.
WBB(:,dsay).urq=WtildaQ(K(:,dsay+1)+1:end,K(:,dsay+1)+1:end);
     % WBB üst üçgen bir matris.
```

Ek-10 (Devam) Genelleştirilmiş doğrusal en küçük kareler problemi haline getirilen görünüşte ilişkisiz regresyon modelinin genelleştirilmiş QR ayrışımı kullanılarak çözümüne ilişkin Matlab kodu (Algoritma II)

```
for i=dsay:-1:2

   % W.. matrislerinin uygun blok elemanları seçiliyor...
   WAA(:,i-1).urq=WAA(:,i).urq(:,K(:,i)+1:end);
   WAB(:,i-1).urq=WAB(:,i).urq(:,(i-2)*gsay-K(:,i-1)+1:(i-1)*gsay-K(:,i));
   WBA(:,i-1).urq=WBA(:,i).urq((i-2)*gsay-K(:,i-1)+1:(i-1)*gsay-
K(:,i),K(:,i)+1:end);
   WBB(:,i-1).urq=WBB(:,i).urq((i-2)*gsay-K(:,i-1)+1:(i-1)*gsay-K(:,i),(i-
2)*gsay-K(:,i-1)+1:(i-1)*gsay-K(:,i));

   %  WB. matrisleri için QL ayrışımı yapılıyor...
   [Plq(:,i-1).urq,Llq(:,i-1).urq]=QLhouse([WBA(:,i-1).urq,WBB(:,i-1).urq]);

   %   Elde edilen P ortogonal matrisi W.. matrislerinin uygun blok elemanlarına
   %   çarpılıyor...
   AA=WAA(:,i).urq(:,K(:,i)+1:end);
   AB=WAB(:,i).urq(:,(i-2)*gsay-K(:,i-1)+1:(i-1)*gsay-K(:,i));
   BA=WBA(:,i).urq(1:(i-2)*gsay-K(:,i-1),K(:,i)+1:end);
   BB=WBB(:,i).urq(1:(i-2)*gsay-K(:,i-1),(i-2)*gsay-K(:,i-1)+1:(i-1)*gsay-
K(:,i));

   AP=[AA,AB]*Plq(:,i-1).urq;
   BP=[BA,BB]*Plq(:,i-1).urq;

   %   Matrislerin belirli blok elemanları değiştirilerek yeniden yapılandırılıyor...
   WAA(:,i).urq(:,K(:,i)+1:end)=AP(:,1:K(:,end)-K(:,i));
   WAA(:,i-1).urq=WAA(:,i).urq;
```

Ek-10 (Devam) Genelleştirilmiş doğrusal en küçük kareler problemi haline getirilen görünüşte ilişkisiz regresyon modelinin genelleştirilmiş QR ayrışımı kullanılarak çözümüne ilişkin Matlab kodu (Algoritma II)

```
    WAB(:,i).urq(:,(i-2)*gsay-K(:,i-1)+1:(i-1)*gsay-K(:,i))=AP(:,K(:,end)-
K(:,i)+1:end);
    WAB(:,i-1).urq=WAB(:,i).urq;

    WBA(:,i).urq((i-2)*gsay-K(:,i-1)+1:(i-1)*gsay-K(:,i),K(:,i)+1:end)=Llq(:,i-
1).urq(1:K(:,end)-K(:,i),:)';
    WBA(:,i).urq(1:(i-2)*gsay-K(:,i-1),K(:,i)+1:end)=BP(:,1:K(:,end)-K(:,i));
    WBA(:,i-1).urq=WBA(:,i).urq;

    WBB(:,i).urq((i-2)*gsay-K(:,i-1)+1:(i-1)*gsay-K(:,i),(i-2)*gsay-K(:,i-1)+1:(i-
1)*gsay-K(:,i))=Llq(:,i-1).urq(K(:,end)-K(:,i)+1:end,:)';
    WBB(:,i).urq(1:(i-2)*gsay-K(:,i-1),(i-2)*gsay-K(:,i-1)+1:(i-1)*gsay-
K(:,i))=BP(:,K(:,end)-K(:,i)+1:end);
    WBB(:,i-1).urq=WBB(:,i).urq;
end

%   Elde edilen y vektör bileşenleri için VEC operatör işlemi uygulanıyor...
vecy.tilda = vertcat(y.tilda);
vecy.sapka = vertcat(y.sapka);

%   Üçgensel sistem çözülüyor...
vecvsapka = inv(WBB(:,1).urq)*vecy.sapka;
beta = inv(blkdiag(R.qr))*(vecy.tilda-WAB(:,1).urq*vecvsapka);

% Herbir denklem için tahmin edilen beta vektörleri ayrıştırılıp kaydediliyor...
topbdsay = 0;
for i=1:dsay
    topbdsay = topbdsay + sum(sonuc(i).bdsay);
```

Ek-10 (Devam) Genelleştirilmiş doğrusal en küçük kareler problemi haline getirilen görünüşte ilişkisiz regresyon modelinin genelleştirilmiş QR ayrışımı kullanılarak çözümüne ilişkin Matlab kodu (Algoritma II)

```
    sonuc(i).beta = beta(topbdsay-(sonuc(i).bdsay-1):topbdsay);
end

%   Görünüşte ilişkisiz regresyon artık vektörlerinin hesaplanması...
emat = zeros(gsay,dsay);
for i=1:dsay
    sonuc(i).artik = y(i).denk - x(i).denk*sonuc(i).beta;
    emat(:,i) = sonuc(i).artik;
end

%   Görünüşte ilişkisiz regresyon varyans-kovaryans(sigma) matrisleri
%   hesaplanıyor...
sigma = zeros(dsay,dsay);
for i=1:dsay
    for j=i:dsay
        sigma(i,j) = (emat(:,i)'*emat(:,j))/gsay;
        if j > 1
            sigma(j,i) = sigma(i,j);
        end
    end
end

%   Gir Korelasyon matrisi hesaplanıyor...
for i=1:dsay
    %   denklemler arası çapraz korelasyon matrisi.
    sonuc(i).ccorr = corrcoef(emat);
end
```

Ek-10 (Devam) Genelleştirilmiş doğrusal en küçük kareler problemi haline getirilen görünüşte ilişkisiz regresyon modelinin genelleştirilmiş QR ayrışımı kullanılarak çözümüne ilişkin Matlab kodu (Algoritma II)

```
%   Gir model r-karesinin elde edilmesi
%   William H. Greene, 1993, s 490
for i=1:dsay
   yort(:,i) = mean(sonuc(i).y);
end

for i=1:dsay
   for j=1:dsay
      deger=(1/gsay)*(y(i).denk-yort(:,i)*ones(gsay,1)).*(y(j).denk-
yort(:,j)*ones(gsay,1));
      syy(i,j) = sum(deger);
      if j > i
         syy(j,i) = syy(i,j);
      end
   end
end
mrkare = 1 - dsay/trace(inv(sigma)*syy);

%   herbir denkleme ilişkin sonuçların oluşturulması
for i=1:dsay
   sonuc(i).sigma = sigma;          % denklemler arası Varyans-Kovaryans matrisi.
   sonuc(i).mrkare = mrkare;
   sonuc(i).ytahmin = sonuc(i).y - x(i).denk*sonuc(i).beta;
   sonuc(i).artik = sonuc(i).y - sonuc(i).ytahmin;
   hkt = sonuc(i).artik'*sonuc(i).artik;
   sonuc(i).skare = hkt/gsay;       % hata terimi varyansı
end
```