

İSTANBUL TEKNİK ÜNİVERSİTESİ ★ FEN BİLİMLERİ ENSTİTÜSÜ

**ANALİZ VE SENTEZ SEYREKLİK İÇİN SÖZLÜK ÖĞRENME
ALGORİTMALARI VE GÖRÜNTÜ İŞLEMEDE UYGULAMALARI**

YÜKSEK LİSANS TEZİ

Özden BAYIR

Elektronik ve Haberleşme Mühendisliği Anabilim Dalı

Telekomünikasyon Mühendisliği Programı

OCAK 2015

İSTANBUL TEKNİK ÜNİVERSİTESİ ★ FEN BİLİMLERİ ENSTİTÜSÜ

**ANALİZ VE SENTEZ SEYREKLİK İÇİN SÖZLÜK ÖĞRENME
ALGORİTMALARI VE GÖRÜNTÜ İŞLEMEDE UYGULAMALARI**

YÜKSEK LİSANS TEZİ

**Özden BAYIR
(504121326)**

Elektronik ve Haberleşme Mühendisliği Anabilim Dalı

Telekomünikasyon Mühendisliği Programı

Tez Danışmanı: Doç. Dr. Ender M. EKŞİOĞLU

OCAK 2015

İTÜ, Fen Bilimleri Enstitüsü'nün 504121326 numaralı Yüksek Lisans Öğrencisi **Özden BAYIR**, ilgili yönetmeliklerin belirlediği gerekli tüm şartları yerine getirdikten sonra hazırladığı “**ANALİZ VE SENTEZ SEYREKLİK İÇİN SÖZLÜK ÖĞRENME ALGORİTMALARI VE GÖRÜNTÜ İŞLEMEDE UYGULAMALARI**” başlıklı tezini aşağıda imzaları olan jüri önünde başarı ile sunmuştur.

Tez Danışmanı : **Doç. Dr. Ender M. Ekşioğlu**
İstanbul Teknik Üniversitesi

Jüri Üyeleri : **Prof. Dr. Ahmet Hamdi KAYRAN**
İstanbul Teknik Üniversitesi

Yrd. Doç. Dr. Hacı İLHAN
Yıldız Teknik Üniversitesi

Teslim Tarihi : **11 Aralık 2014**
Savunma Tarihi : **20 Ocak 2015**

Canim aileme,

ÖNSÖZ

Tez çalışmam boyunca değerli görüşlerini ve katkılarını benden esirgemeyen danışmanım Sayın Doç. Dr. Ender M. EKŞİOĞLU' na teşekkürü borç bilirim.

Ayrıca, tüm eğitim hayatım boyunca yanımda olan, maddi manevi desteklerini benden esirgemeyen başta annem, babam olmak üzere bütün aileme ve arkadaşlarıma sonsuz teşekkürlerimi sunarım.

Aralık 2014

Özden BAYIR

İÇİNDEKİLER

Sayfa

ÖNSÖZ.....	vii
İÇİNDEKİLER	ix
KISALTMALAR	xi
ÇİZELGE LİSTESİ.....	xiii
ŞEKİL LİSTESİ.....	xv
ÖZET.....	xvii
SUMMARY	xix
1. GİRİŞ	1
2. SEYREK İŞARET İŞLEME	5
2.1 Sentez Seyreklik Modeli	5
2.2 Analiz Seyreklik Modeli	7
2.3 Seyreklik Tabanlı Görüntü İşleme Uygulamaları	11
2.3.1 Seyreklik Tabanlı Sınıflandırma	11
2.3.2 Seyreklik Tabanlı Görüntü Ayırıştırma	12
2.3.3 Seyreklik Tabanlı İç Boyama.....	14
3. ANALİZ VE SENTEZ SEYREKLİK İÇİN SÖZLÜK ÖĞRENME	17
3.1 Sentez Seyreklik Modelinde Sözlük Öğrenme.....	17
3.2 Analiz Seyreklik Modelinde Sözlük Öğrenme.....	19
3.2.1 Operatör Öğrenme.....	19
3.2.2 Seyrekleştirici Dönüşüm Öğrenme	21
4. SÖZLÜK ÖĞRENME ALGORİTMALARI	25
4.1 K adet – Tekil Değer Ayırıştırma (K-SVD)	25
4.2 Analiz K-SVD	29
4.3 Seyrekleştirici Dönüşüm Öğrenme Algoritması	31
5. YENİ YÖNTEM : SEYREKLEŞTİRİCİ DÖNÜŞÜM OPERATÖRÜ ÖĞRENME.....	35
6. BENZETİM SONUÇLARI	39
7. SONUÇLAR	49
KAYNAKLAR	51
ÖZGEÇMİŞ.....	53

KISALTMALAR

DCT	: Ayrık Kosinüs Dönüşümü
CS	: Sıkıştırılmış Algılama
MP	: Eşleşme Takip Algoritması
OMP	: Dik Eşleşme Takip Algoritması
BP	: Temel Takip Algoritması
FOCUSS	: Odaksal Düşük Tanımlı Sistem Çözümü
MOD	: Optimal Yönler Metodu
MAP	: En Büyük Sonsal
SVD	: Tekil Değer Ayırıştırması
K-SVD	: K adet Tekil Değer Ayırıştırması
GAP	: Açgözlü Analiz Takip Algoritması
BG	: Geriye Doğru Açgözlü Algoritma
OBG	: En İyiye Geriye Doğru Açgözlü Algoritma
CAOL	: Kısıtlı aşırı tam Analiz Operatör Öğrenme Algoritması
UNTF	: Düzgün Normalize Edilmiş Sıkı Çerçeve
CG	: Eşlenik Gradyan
FD	: Sonlu Fark
PSNR	: Doruk İşaret Gürültü Oranı
RMSE	: Ortalama Karese Hatanın Karekökü

ÇİZELGE LİSTESİ

Sayfa

Çizelge 6.1 : Beş farklı görüntüye ait gürültü giderme sonuçları.....	40
Çizelge 6.2 : Farklı N veri boyutları için operatör öğrenme süreleri (sn)	47

ŞEKİL LİSTESİ

Sayfa

Şekil 2.1 : Sentez seyreklik modeli.....	5
Şekil 2.2 : Analiz seyreklik modeli.....	8
Şekil 2.3 : (a)Test görüntüsü,(b)Karikatür görüntüsü,(c)Dokuma desenli görüntü...	13
Şekil 2.4 : Görüntü ayrıştırma örneği (a)Gürültülü test görüntüsü, (b) Ayrıştırılmış dokuma desenli görüntü, (c) Ayrıştırılmış karikatür görüntüsü, (d) Ayrıştırılmış gürültü bileşeni	13
Şekil 2.5 : (a) %25, (b) %50, (c) %75 pikseli maskelenmiş görüntüler. İç boyama ile elde edilmiş görüntüler: (d) RMSE:14.55, (e) RMSE:19.61, (f) RMSE:29.7	15
Şekil 2.6 : Tam örtüşme yapılarak gerçekleştirilen iç boyamayla elde edilmiş görüntüler. (a) RMSE: 9.0, (b) RMSE : 11.55, (c) RMSE: 18.18	15
Şekil 4.1 : K-SVD algoritmasının basit şeması	25
Şekil 4.2 : K-SVD algoritması.....	28
Şekil 4.3 : Analiz K-SVD algoritmasının basit şeması.....	29
Şekil 4.4 : Analiz K-SVD algoritması	31
Şekil 4.5 : Tam dönüşüm öğrenme algoritması	34
Şekil 4.6 : Aşırı tam dönüşüm öğrenme algoritması	34
Şekil 5.1 : Dönüşüm K-SVD algoritmasının basit şeması.....	35
Şekil 5.2 : Dönüşüm K-SVD algoritması	38
Şekil 6.1 : Gürültüsü giderilmiş görüntüler (Lena) (a) Gürültülü görüntü (b)Dönüşüm K-SVD kullanarak gürültüsü giderilmiş görüntü (c) Analiz K-SVD kullanarak gürültüsü giderilmiş görüntü.....	41
Şekil 6.2 : Öğrenilmiş operatörler, (a) Analiz K-SVD, ‘Barbara’ (b) Dönüşüm K-SVD, ‘Barbara’ (c) Analiz K-SVD, ‘Lena’ (d) Dönüşüm K-SVD, ‘Lena’	42
Şekil 6.3 : Gürültüsü giderilmiş görüntüler (Barbara) (a) Gürültülü görüntü (b)Dönüşüm K-SVD kullanarak gürültüsü giderilmiş görüntü (c) Analiz K-SVD kullanarak gürültüsü giderilmiş görüntü	43
Şekil 6.4 : Gürültüsü giderilmiş görüntüler (House) (a) Gürültülü görüntü (b)Dönüşüm K-SVD kullanarak gürültüsü giderilmiş görüntü (c) Analiz K-SVD kullanarak gürültüsü giderilmiş görüntü	44
Şekil 6.5 : Gürültüsü giderilmiş görüntüler (Peppers) (a) Gürültülü görüntü (b)Dönüşüm K-SVD kullanarak gürültüsü giderilmiş görüntü (c) Analiz K-SVD kullanarak gürültüsü giderilmiş görüntü	45
Şekil 6.6 : Gürültüsü giderilmiş görüntüler (Boats) (a) Gürültülü görüntü (b)Dönüşüm K-SVD kullanarak gürültüsü giderilmiş görüntü (c) Analiz K-SVD kullanarak gürültüsü giderilmiş görüntü	46

ANALİZ VE SENTEZ SEYREKLİK İÇİN SÖZLÜK ÖĞRENME ALGORİTMALARI VE GÖRÜNTÜ İŞLEMEDE UYGULAMALARI

ÖZET

İşaret işleme alanında ses, video, metin ya da görüntü işaretinin gözlemlerini kullanarak işareti geri çatma problemi oldukça sık karşılaşılan önemli bir problemdir. Bir x işareti ve y gözlemlerine ait, doğrusal bilgi edinme modeli $Ax = y$ şeklinde gösterilebilir. Bilindiği üzere, bir işareti gözlemlerinden kayıpsız bir şekilde geri elde edebilmek için en az işaret boyutu kadar gözlem bilgisine sahip olmak gereklidir. Bu klasik bilgiye göre işaret boyutundan daha az gözlem verisi ile işareti geri elde etmek imkansız gibi görünmektedir. Ancak bazı varsayımlar altında işaret boyutundan daha az sayıda gözlem ile işareti geri elde etmek mümkün hale gelmektedir. Bu varsayımlardan biri işaretlerin ‘seyrek’ olduğu varsayımdır.

Seyrek gösterilim ile bir işaret kendi boyutundan daha az sayıda katsayı kullanılarak ifade edilebilmektedir. Bu özellik sebebiyle işaretlerin seyrek gösterilimi, işaret işleme alanındaki geri çatma, sıkıştırma, görüntü ayrıştırma, gürültü azaltma, öznetelik çıkarma ve sınıflandırma gibi birçok problemin çözümünde kullanılmaktadır.

İşaretlerin seyrek gösteriliminde temel olarak sentez ve analiz model olmak üzere iki model vardır. Sentez modelde işaret bir sözlük ve bir seyrek katsayı vektörü kullanılarak ifade edilir. Burada kullanılan sözlük aşırı tam bir sözlüktür ve bu sözlüğün, seyrek katsayı vektörünün sıfır olmayan elemanlarına denk düşen sütunlarının doğrusal kombinasyonu ile işaret seyrek bir şekilde gösterilmektedir. İşaretin seyrek gösterilimi için kullanılan sözlük sütunu sayısı işaretin boyutundan oldukça düşüktür. Analiz modelde ise işaretin seyrek gösterilimi, bir operatör ile seyrek gösterilimi aranan işareti çarpmak suretiyle elde edilir. Ancak son yıllarda bu modellerin yanı sıra analiz modelin genelleştirilmiş hali olan seyrekleştirici dönüşüm modeli de oldukça verimli bir model olarak karşımıza çıkmaktadır. Bu modelde analiz modeldekine benzer operatörler daha az hesapsal karmaşıklık ile elde edilebilmektedir.

Bu yüksek lisans tez çalışmasında işaretlerin seyrek gösteriliminde kullanılan modeller ve özellikle bu modellerdeki temel problemler olan sözlük öğrenme, operatör öğrenme ve seyrekleştirici dönüşüm öğrenme problemleri incelenmiştir. Bu problemlerin çözümü için sunulan çeşitli algoritmalar MATLAB ortamında gerçekleştirilmiştir. Ayrıca analiz modelinin ve seyrekleştirici dönüşüm modelinin avantajlarını bir araya getiren yeni bir seyrekleştirici dönüşüm operatörü öğrenme algoritması tanıtılmıştır. Bu tanıtılan algoritma ile analiz modelinin en yaygın bilinen algoritması görüntü gürültüsü giderme uygulaması ile karşılaştırılmıştır. Bu karşılaştırmaya ilişkin sonuçlar da tez çalışmasında verilmiştir.

DICTIONARY LEARNING ALGORITHMS FOR ANALYSIS AND SYNTHESIS SPARSITY AND APPLICATIONS IN IMAGE PROCESSING

SUMMARY

In the research area of signal processing, reconstruction of any audio, video or image signal from its measured data is a commonly encountered problem. The linear information acquisition process can be shown as $Ax = y$ for the measured data y and the signal x . According to the classical information, the amount of the measured data must be at least as large as the signal length to reconstruct the signal x from its measured data y , by solving the linear system of equations. For this reason, without any additional information, it is impossible to recover the signal from the measured data which is smaller than the signal length N . However, under some certain assumption, it becomes possible to recover a signal when the amount of measurements, m , is smaller than the signal length N .

The ‘sparsity’ assumption of signals makes this case possible. Via the sparse representation, many signals can be defined with a few number of coefficients, which are much less than the length of the signal.

The sparse representation of signals is used in signal and image processing area for many problems such as signal reconstruction, compression, image decomposition, denoising, feature extraction and classification.

There are two basic models for sparse representation of signals which are named “synthesis” and “analysis” sparsity models. In synthesis sparsity model, a signal is defined as the outcome of the multiplication of a dictionary and a sparse coefficient vector. This used dictionary is an overcomplete dictionary and the signal is represented as a linear combination of few columns from this dictionary which correspond to the nonzero elements of the sparse coefficients vector. Here the number of used atoms is less than the length of the signal. In the analysis sparsity model the sparse representation of a signal is obtained by multiplication of the signal and an operator. In recent years sparsifying transform model which is more general form of the analysis model has been considered as a quite efficient approach. In this transform model, the operators which are similar to the analysis operators are learned with less computational complexity.

In this master thesis, the models which are used in sparse representation of signals are studied. Moreover the dictionary learning, operator learning and sparsifying transform learning problems which are the essential problems of the mentioned models are handled. Therefore various algorithms are implemented with MATLAB for solving these problems. Additionally Transform K-SVD, a new transform operator learning algorithm is proposed which brings the transform learning and the K-SVD based analysis dictionary learning approaches together. Finally this novel algorithm is compared with the commonly known analysis model’s algorithm in an image denoising experiment. The results are presented with some tables and images within this thesis.

1. GİRİŞ

İşaret işleme alanında, bir işareti gözlemlerinden geri çatma problemi sıklıkla karşılaşılan bir problemdir. Klasik bilgilere göre bir işareti, örneklerinden kayıpsız bir şekilde geri elde edebilmek için en az, işaret uzunluğu olan K kadar gözlem verisi olması gereklidir [1, 2]. Bu temel ilke, analog-dijital çeviriciler, medikal görüntüleme, radar ve mobil haberleşme gibi birçok teknolojik cihazda esas alınmıştır. Gözlem (veri) vektörü $y \in \mathbb{R}^M$ ve gözlemlerden geri elde edilecek olan işaret $x \in \mathbb{R}^K$ olmak üzere, $M < K$ olması durumunda klasik doğrusal cebir bilgilerine göre $Ax = y$ gözlem modeli düşük tanımlı bir doğrusal denklem sistemidir ve çok fazla sayıda çözümü vardır. Başka bir deyişle herhangi bir ek bilgi olmaksızın x işaretini y verisinden elde etmek mümkün değildir [1, 2]. Bu durumda bazı varsayımlar altında, işaret boyutu K ’ dan daha az gözlem ile x işaretini geri elde edebilme düşüncesi oldukça dikkat çekicidir. Tüm bu imkansızlık durumlarını ortadan kaldıracak olan muhtemel varsayımlardan biri ‘seyreklik’ varsayımdır. Bu varsayımın temelini oluşturduğu araştırma alanı sıkıştırılmış algılama (Compressed Sensing-CS) olarak adlandırılır [1, 2].

Bir işaretin farklı dönüşüm yöntemleri kullanılarak gösterilimi problemi verim, hız vs. açısından kazanç sağlaması sebebiyle işaret işleme alanındaki birçok çalışmanın önemli bir araştırma konusu olmuştur. Bu dönüşüm yöntemlerinden en önemlileri “Dalgacık Dönüşümü (Wavelet Transform)”, “Fourier Dönüşümü”, “Ayrık Kosinüs Dönüşümü (Discrete Cosine Transform)” şeklinde örneklendirilebilir. Ancak son yıllarda işaretlerin bu dönüşüm metodlarının yanı sıra, ‘seyreklik’ varsayımı altında, bir sözlük ya da operatör vasıtası ile gösterilimleri konusundaki çalışmalar da gittikçe artmaktadır. Bu gösterilim metodu işaretlerin seyrek gösterilimi olarak bilinir. Burada bahsedilen seyrek gösterilimden kasıt, işaretin gösteriliminde kullanılacak olan mühim katsayıların işaret boyutuna göre daha az seçilerek gösterilimin seyrek şekilde yapılmasıdır. İşaretin seyrek gösterilimi sırasında kullanılan dönüşüm matrisinin işaret boyutuyla aynı boyutta olması durumunda bu dönüşüme tam (complete) veya kare dönüşüm denir. Bu tam dönüşüm metodlarının en iyi örnekleri olarak “Ayrık Kosinüs

Dönüşümü (Discrete Cosine Transform-DCT)” ve “Dalgacık Dönüşümü (Wavelet Transform)” verilebilir. Bu dönüşüm matrislerinin sütunları (baz vektörleri) kullanılarak işaretler yaklaşık ya da tam olarak seyrek gösterilebilir. Ancak bu tam dönüşüm metodlarının yanı sıra seyrekliğin daha iyi uygulanabildiği aşırı tam (overcomplete) dönüşümler de vardır. Bu dönüşümlerde kullanılan dönüşüm matrislerinin sütunları kullanılarak seyrek gösterilim sağlanır. Bu aşırı tam dönüşüm metodunda kullanılan dönüşüm matrisleri işaret uzayından daha fazla sayıda sütun ya da satır içerdiğinden bu matrislere gereksizlik (redundancy) içeren matrisler denir ve özel olarak sözlük (dictionary) ya da operatör (operator) şeklinde isimlendirilir. Bu sözlük ya da operatör matrislerinin sütunlarına ya da satırlarına atom denir.

Seyrek gösterilim genel olarak sentez seyreklik modeli ve analiz seyreklik modeli şeklinde iki başlığa ayrılabilir.

Sentez seyreklik modelinde, ele alınan işaret, sütunlarında atomlar içeren bir sözlük matrisi ve az sayıda sıfır olmayan elemanlardan oluşan bir seyrek katsayı vektörünün çarpımı şeklinde modellenir [3]. Burada sözlük matrisi $D \in \mathbb{R}^{M \times K}$ ve seyrek katsayı vektörü $x \in \mathbb{R}^K$ şeklinde gösterilirse, bir $y \in \mathbb{R}^M$ işareti $y = Dx$ şeklinde ifade edilir. Bu gösterilime y işaretinin seyrek gösterilimi denir. Bu çarpım işlemi ile y işareti sözlük matrisinin az sayıdaki atomlarının doğrusal kombinasyonundan elde edilmiş olur. Sentez seyreklik modelinde hedef, işareti seyrek bir şekilde gösterebileceğimiz bir sözlük matrisi ve bir seyrek katsayı vektörü bulmaktır. Bu işlemler iki aşamada yapılır ki bu aşamalar sırasıyla seyrek katsayı vektörünün bulunması aşaması ve bulunan bu seyrek katsayı vektörünün kullanımıyla sözlük matrisinin güncellenmesi aşamasıdır. Seyrek katsayı vektörünün bulunması sırasında seyrek gösterilimi bulunmaya çalışılan işaretin kendisi ve analitik yolla (DCT, dalgacık vs.) öğrenilmiş olan herhangi bir sözlük matrisi kullanılır. İkinci aşamada ise ilk aşamada bulunan seyrek katsayı vektörü ve yine işaretin kendisi kullanılarak sözlük matrisi güncellenir. İlk aşamada seyrek katsayı vektörünün bulunmasında vektör seçim algoritmaları kullanılır. Bunlardan bazıları “Eşleşme Takip Algoritması (Matching Pursuit-MP)” [4], “Dik Eşleşme Takip Algoritması (Orthogonal Matching Pursuit-OMP)” [5], “Temel Takip Algoritması (Basis Pursuit-BP)” [6] ve “Odaksal Düşük-Tanımlı Sistem Çözümü (Focal Under-determined System Solver-FOCUSS)” [7] şeklinde örneklendirilebilir. İkinci aşamada sözlük matrisinin güncellenmesi sırasında kullanılan algoritmalara örnek olarak “Optimal Yönler Metodu (Method of Optimal

Directions-MOD)” [8, 9, 10], “En Büyük Sonsal (Maximum A Posteriori- MAP)” [11, 12, 13], “K adet Tekil Değer Ayrıştırması (K-Singular Value Decomposition – K-SVD)” [2, 13, 14] verilebilir.

Bu alandaki araştırmacıların sentez seyreklik modeline nazaran daha sonraları dikkatini çeken bir diğer model de analiz seyreklik modelidir. Bu model de temel olarak sentez seyreklik modeli ile benzerdir. Analiz modelde seyrek gösterilimi yapılmak istenen işaret bir operatör aracılığıyla seyrek olduğu varsayılan başka bir dönüşüm uzayına taşınır. Tez çalışmasında her iki isimlendirme de kullanılmıştır. Analiz modelde hedef, operatör $\Omega \in \mathbb{R}^{K \times M}$ ve seyrek gösterilmek istenen işaret $y \in \mathbb{R}^M$ olmak üzere, $\Omega y \in \mathbb{R}^K$ analiz gösterilim vektörünün seyrek olmasını sağlayan bir Ω operatörü bulmaktır [3]. Sentez seyrek gösterilimde sözlük matrisinin sütunlarına ‘atom’ denirken benzer şekilde analiz seyrek gösterilimde de operatörün satırları ‘atom’ olarak isimlendirilir. Operatörün ya da sözlüğün kare ve tersi alınabilir (tekil olmayan - nonsingular) olması durumunda bu modellerin birbirlerinin çifti (dual) olduğu söylenebilir, ancak bu durum gereksizlik eklenmiş (redundant-kare olmayan) operatör ya da sözlükler için geçerli değildir. Bu durum ikinci bölümde daha detaylı bir şekilde incelenecektir.

Seyrek gösterilim ana başlık olarak sentez ve analiz seyreklik modeli şeklinde ikiye ayrılır. Ancak son yıllarda analiz seyreklik modelinin genelleştirilmiş bir hali olan ‘seyrekleştirici dönüşüm modeli (dönüşüm model)’ de araştırmacıların çalışma konusu olmuştur [15, 16].

Bu yüksek lisans tez çalışmasında işaretlerin seyrek gösterilimi için kullanılan sentez seyreklik modeli, analiz seyreklik modeli, seyrekleştirici dönüşüm modeli üzerinde çalışılmış, bu modellerdeki en önemli problemler olan sözlük öğrenme, analiz sözlük öğrenme ve seyrekleştirici dönüşüm öğrenme problemlerinin çözümü için sunulan algoritmalarından bir kısmı detaylı bir şekilde incelenmiş ve MATLAB ortamında gerçekleştirilmiştir. Ayrıca bu modellerdeki en yaygın algoritmalara ek olarak yeni bir seyrekleştirici dönüşüm operatörü öğrenme metodu olan “Dönüşüm K-SVD (Transform K-SVD)” algoritması tanıtılmış ve gerçekleştirilmiştir. Tez çalışmasının sonunda, bu yeni dönüşüm K-SVD metodu ile analiz K-SVD metoduna ait görüntü gürültüsü giderme uygulamasının karşılaştırmalı sonuçlarına da yer verilmiştir.

Tez çalışması yedi ana bölümden oluşmaktadır. Giriş bölümünde işaretlerin seyrek gösterilimi ve bu gösterilime ilişkin modeller genel bir çerçevede açıklanmıştır. İkinci bölümde bu seyrek gösterilim modelleri daha detaylı bilgiler ile sunulmuştur. Çalışmanın üçüncü bölümünde seyrek gösterilimin en önemli aşaması olan sözlüğün, operatörün ya da seyrekleştirici dönüşümün öğrenilmesi aşaması detaylı bir şekilde incelenmiştir. Dördüncü bölümde sözlük, operatör ve seyrekleştirici dönüşüm öğrenme algoritmalarının en önemli örnekleri olan K-SVD [13], Analiz K-SVD [3] ve Seyrekleştirici dönüşüm öğrenme [15, 16] algoritmaları incelenmiştir.

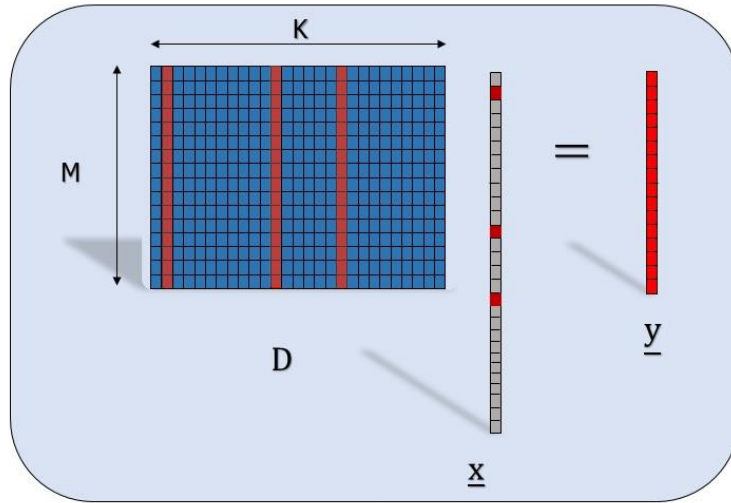
Beşinci bölümde, yeni bir yöntem olarak önerilen seyrekleştirici dönüşüm operatörü öğrenme algoritması detaylı bir şekilde açıklanmıştır. Altıncı bölümde ise önerilen yeni yöntem kullanılarak daha az hesapsal karmaşıklıkla öğrenilen seyrekleştirici dönüşüm operatörünün, analiz K-SVD algoritması kullanılarak öğrenilmiş bir operatöre nazaran genel olarak daha iyi performansa sahip olduğu MATLAB ortamındaki benzetimler ile birlikte sunulmuştur. Tez çalışmasının son bölümünde ise çalışmanın sonucunda elde edilen bilgiler sunularak çalışma tamamlanmıştır.

2. SEYREK İŞARET İŞLEME

Giriş bölümünde de bahsedildiği üzere bir işaret, aşırı tam bir sözlük matrisinin ya da operatör matrisinin az sayıdaki atomlarının doğrusal birleşimi ile seyrek bir şekilde gösterilebilir. Bu gösterilim sentez seyreklik modeli ve analiz seyreklik modeli ile yapılabilir. Tez çalışmasının bu bölümünde bu seyrek gösterilim modelleri incelenecektir.

2.1 Sentez Seyreklik Modeli

Bu modelde ele alınan işaret $y \in \mathbb{R}^M$, aşırı tam bir sözlük matrisi $D \in \mathbb{R}^{M \times K}$ ve az sayıda sıfır olmayan eleman içeren bir seyrek katsayı vektörünün $x \in \mathbb{R}^K$ çarpımı şeklinde modellenir. Şekil 2.1’ de bu model görülmektedir.



Şekil 2.1 : Sentez seyreklik modeli.

Burada aşırı tam sözlüğün atomları $\{d_j\}_{j=1}^K$ şeklinde gösterilir bu ifadeden görüldüğü üzere D sözlük matrisi K adet M boyutlu atomlardan oluşur. Ağırlık vektörü olarak da isimlendirilen x seyrek katsayı vektörü, gösterilimi yapılan işaret olan y işaretiye ait katsayıları göstermektedir. İşaretin gösterilimi sırasında seyrek katsayı vektörünün sıfır olmayan elemanlarına karşı düşen sözlük atomları kullanıldığından dolayı bu vektörün az sayıda sıfırdan farklı elemana sahip olması y işaretinin seyrek

gösterilimini sağlar. y işaretinin seyrek gösterilimi $y = Dx$ şeklinde ya da $\|y - Dx\|_p \leq \epsilon$ koşulunu sağlamak üzere yaklaşık bir ifadeyle $y \approx Dx$ şeklinde yapılabilir.

Burada görülen $\|\cdot\|_p$ notasyonu norm ismini alır. x seyrek katsayı vektörüne ait l^p normu $\|x\|_p = (\sum_{k=1}^K |x_k|^p)^{\frac{1}{p}}$ şeklinde ifade edilir.

Sentez seyrek gösterilim için seyrek katsayı vektörünün ve sözlük matrisinin bulunması gereklidir. Bu probleme sözlük öğrenme problemi denir ve bu problemin çözümü iki aşamada gerçekleştirilir. Bu aşamaları tanıtmadan önce bir işaretin seyrek gösterilim problemini tanımlamak gereklidir. Bir işaretin az sayıda sıfır olmayan katsayı kullanarak gösterilimi problemi seyrek gösterilim problemidir ve denklem (2.1) 'deki gibi verilir [13].

$$\min_x \|x\|_0, \quad y = Dx \quad (2.1)$$

Burada $\|x\|_0$, x vektörünün sıfır normunu ifade etmektedir ve basit bir deyişle sıfır normu x vektörünün sıfırdan farklı elemanlarının sayısını vermektedir. Seyrek gösterilim için $\|x\|_0 \ll K$ olmalıdır. Sonuç olarak (2.1) problemi, y işaretinin, az sayıda sözlük atomlarının birleşimi ile seyrek gösterilimini sağlayacak olan x seyrek katsayı vektörlerinden en az sayıda sıfırdan farklı eleman içerenini aramaktadır şeklinde açıklanabilir. Sözlük öğrenme problemi (2.2) denkleminde görülmektedir.

$$\min_{D,X} \|Y - DX\|_F^2, \quad \|X_i\|_0 \leq s \quad \forall i \quad (2.2)$$

Burada X_i , $X \in \mathbb{R}^{K \times N}$ matrisinin sütunlarıdır. $Y \in \mathbb{R}^{M \times N}$ matrisi ise sütunlarında seyrek gösterilimi aranan işaretleri içeren bir matristir. Bu denklemde $\|\cdot\|_F$ Frobenius normunu göstermektedir. Bir Z vektörüne ilişkin frobenius normu, $\|Z\|_F = \sqrt{\sum_{ij} Z_{ij}^2}$ şeklinde verilir.

(2.2) denkleminde görülen sözlük öğrenme problemin çözümünün ilk aşaması seyrek katsayı vektörünün bulunması aşamasıdır. Bu aşama literatürde gürültülü seyrek kodlama şeklinde isimlendirilir ve bu problem şu şekilde verilir [13]:

$$\min_x \|Y_i - DX_i\|_F^2, \quad \|X_i\|_0 \leq s \quad \forall i \quad (2.3)$$

Bu ifadede s arzu edilen seyreklik seviyesini göstermektedir. (2.3) denklemi, belli bir seyreklik seviyesine sahip X_i seyrek katsayı vektörlerinden biri ile D matrisinin çarpılmasıyla elde edilen seyrek gösterilimin (DX_i) veri ile (Y_i) işareti arasındaki

karesel hatayı en az yapacak olanını seçme problemini göstermektedir. Bu problem her i için çözülür ve X matrisi bulunur. Seyrek kodlama probleminin yaklaşık çözümünü bulan çeşitli algoritmalar vardır. Bunlardan başlıcaları MP [4], OMP [5], BP [6] ve FOCUSS [7] olarak verilebilir.

Sözlük öğrenme probleminin ikinci aşaması sözlük güncelleme aşamasıdır. Bu aşamada ilk aşamada bulunan X matrisi kullanılarak D sözlük matrisi güncellenir. Problem şu şekilde gösterilebilir:

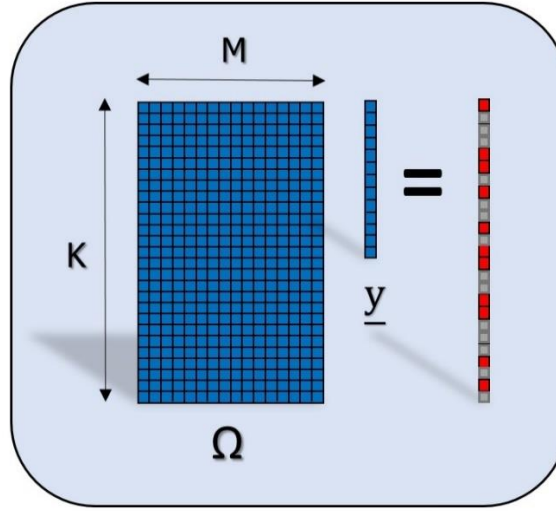
$$\min_D \|Y - DX\|_F^2 \quad (2.4)$$

Diğer seyreklik modellerinde de sentez modelinin sözlük öğrenme probleminin benzeri olan operatör öğrenme ve dönüşüm öğrenme problemleri vardır. Bu dönüşüm öğrenme probleminin çözümü için [17] ve [18] çalışmaları, operatör öğrenme probleminin çözümü için de [19,20,21] çalışmaları örnek olarak verilebilir. Bu çalışmalardan bazıları gelecek bölümlerde incelenecektir.

Sözlük öğrenme problemine tekrar dönmek gerekirse, bu problem NP-zor olan bir problemdir. Bazı problemlerin çözümü için, çalışma süresi, girilen verinin büyüklüğüne bir polinom cinsinden bağlı olan algoritmalar bulunmaktadır. Bu tür problemler P (polynomial - polinomsal) kategorisinde olan problemlerdir. Ancak bazı problemler için girilen verinin büyüklüğüne polinom mertebesinde bağımlı bir sürede cevap verecek bir algoritma bulunmamaktadır. Bunun yerine, bir şekilde cevabı tahmin edebiliyorsak tahminimizin doğruluğunu sınamak için veri büyüklüğüne polinom mertebesinde bağımlı sürelerde çalışacak algoritmalar vardır. Bu tür problemler de NP (non-deterministic polynomial – deterministik olmayan polinomsal) kategorisini oluştururlar. NP kategorisinde bulunan problemler zorluk derecelerine göre NP-zor (NP-hard) ve NP-tam (NP-complete) gibi sınıflara ayrılırlar. En az her bir NP problem kadar zor olan problemlerin oluşturduğu sınıf NP-zor sınıfıdır. Hem NP olup, hem de NP-zor olan problemlerin oluşturduğu sınıf ise NP-tam sınıfıdır. NP kategorisindeki en zor problemler NP-tam problemlerdir [22]. (2.2) denklemindeki NP-zor olan sözlük öğrenme probleminin çözümü için de MAP [11, 12], MOD [9], K-SVD [13] gibi çeşitli algoritmalar sunulmuştur. Sözlük öğrenme probleminin detaylı incelemesi üçüncü bölümde yapılmıştır.

2.2 Analiz Seyreklik Modeli

Birçok çalışmaya konu olan sentez seyreklik modelinin yanısıra bu modele benzer bir şekilde seyrek gösterilimde kullanılabilecek olan bir model de analiz seyreklik modelidir. Sentez modeline göre araştırmacıların daha sonraları dikkatini çeken bu model satırları atomlardan oluşan ve operatör (analiz sözlük) olarak isimlendirilen bir $\Omega \in \mathbb{R}^{K \times M}$ matrisine dayanır. Analiz modelde bir işaretin seyrek gösterilimi, Ω operatörü ile çarpılmak suretiyle elde edilir. Şekil 2.2’ de analiz seyreklik modeline ilişkin matematiksel model görülmektedir.



Şekil 2.2 : Analiz seyreklik modeli.

Bu modelde hedef, verilen veriden bir analiz operatörü öğrenmektir. Ele alınan bir $y \in \mathbb{R}^M$ işareti için bu modelde en önemli beklenti $\Omega y \in \mathbb{R}^K$ analiz gösteriliminin seyrek olmasıdır. Analiz seyreklik vektörünün sıfır olan elemanları işaretin dik olduğu alt uzayı belirlerken bu uzayın dışında kalan uzay da işaretin ait olduğu alt uzayı temsil eder. Ωy analiz seyreklik vektörünün sıfırlarının sayısı eş-seyreklik olarak isimlendirilir. İşaretin ait olduğu alt uzayın boyutu r ve işaret boyutu M olmak üzere; $r \ll M$ olduğu varsayılır [3]. Bir y işareti için ait eş-seyreklik değeri o işaretin Ω operatörüne göre elde edilen eş-destek kümesinin niceliği (büyüklüğü) olarak verilir. Eş-destek, Ωy vektörünün sıfır değerini aldığı indislerin kümesi şeklinde tanımlanabilir: $\Lambda = \{m : \{\Omega y\}_m = 0, \forall m = 1, \dots, M\}$. [17]. Burada eş-seyreklik değeri l olarak alınırsa $l = |\Lambda|$ yazılabilir. Operatörün her bir atomunun boyutu M olarak alınırsa, işaretin dik olduğu alt uzayın boyutu l ve işaretin ait olduğu alt uzayın boyutu da $r = M - l$ olarak bulunur.

Analiz modelinde temel amaç, Ω vektörünün seyrek olmasını sağlayan bir Ω operatörü öğrenmektir. Burada belirtmelidir ki, Ω operatörünün kare ve tekil olmayan (tersi alınabilir) bir matris olması halinde analiz ve sentez modeli $\Omega^{-1} = D$ bağıntısıyla gösterildiği gibi birbirine eş hale gelirler.

Verilen bir x işaretinin analiz gösterilim katsayıları Ωx hesabı ile bulunabilir ve bu gösterilimin niceliği kestirilebilir. Ancak ele alınan işaret toplanır gürültü altında gözlemlenmişse analiz gösterilim Ωx ' in bulunması daha zorlu olacaktır. Gözlemlenen işaretin y ve sıfır ortalamalı toplanır beyaz gauss gürültüsünün v olması durumunda $y = x + v$ yazılabilir. Bu durumda analiz seyreklik modelinde (2.5) denkleminde görülen operatör öğrenme probleminin çözümünde, sentez seyreklik modelindeki sözlük öğrenme probleminin çözümünde olduğu gibi yapılan işlemler iki aşamaya bölünür.

$$\{\hat{\Omega}, \hat{X}, \{\hat{\Lambda}_i\}_{i=1}^N\} = \min_{\Omega, X, \{\Lambda_i\}_{i=1}^N} \|X - Y\|_F^2; \Omega_{\Lambda_i} x_i = 0 \quad \forall 1 \leq i \leq N, \\ \text{Rank}(\Omega_{\Lambda_i}) = d - r \quad \forall 1 \leq i \leq N, \|w_j\|_2 = 1 \quad \forall 1 \leq j \leq K \quad (2.5)$$

Burada görülen operatör öğrenme problemi denklem (2.6)' daki gibi de ifade edilebilir.

$$\min_{\Omega, X} \|Y - X\|_F^2; \|\Omega x_i\|_0 \leq K - l \quad \forall i = 1, 2, \dots, N \quad (2.6)$$

Burada $Y = [y_1 \ y_2 \ \dots \ y_N] \in \mathbb{R}^{M \times N}$ her biri M boyutlu olan N adet gözlem vektörünü (veri vektörü) gösterir. $X = [x_1 \ x_2 \ \dots \ x_N]$ matrisi de bu gözlem matrisindeki her bir vektörün gürültüsüz hali olan N adet M boyutlu vektörlerden oluşur. Bu problemin çözümünün iki aşamasından ilki literatürde analiz seyrek kodlama ya da eş-seyrek kodlama şeklinde isimlendirilir. x orijinal işaretinin, gürültülü gözlem olan y işaretinden elde edilmesi problemi şu şekilde gösterilebilir:

$$\{\hat{x}, \hat{\Lambda}\} = \min_{x, \Lambda} \|x - y\|_2; \Omega_{\Lambda} x = 0, \text{rank}(\Omega_{\Lambda}) = M - r \quad (2.7)$$

(2.7) denklemi ile tanımlanan problem şu şekilde de tanımlanabilir:

$$\min_x \|y - x\|_2^2; \|\Omega x\|_0 \leq K - l \quad (2.8)$$

Bu problem analiz seyrek kodlama problemidir. Literatürde bu problemin çözümü için sunulan çeşitli algoritmalar vardır. Bunlardan biri maliyet fonksiyonundaki l_0 normunu l_1 normu ile değiştirerek problemi konveks olarak hafifletir. Bu yaklaşım sentez seyreklik modelindeki temel-takip algoritmasına benzer bir yaklaşımdır. Bunun

yanı sıra sentez seyreklik modeline benzer bir şekilde analiz modelinde de bu problemin çözümü için çeşitli açgözlü algoritmalar sunulmuştur. Bunlardan ilki sentez modelindeki OMP algoritmasının bir benzeri olan ve Ωx ' in sıfır olmayan elemanlarını kullanarak çözüm arayan “Açgözlü Analiz Takip (Greedy Analysis Pursuit-GAP)” algoritmasıdır [23]. Bu algoritmanın geliştirilmesi ile bulunan “Geriye Doğru Açgözlü (Backward Greedy- BG)” algoritmasında da Ωx ' in sıfır olan elemanları yani eş-desteği kullanılarak çözüm aranır. BG algoritmasının geliştirilmesi ile elde edilen “En iyilenmiş Geriye Doğru Açgözlü (Optimized Backward Greedy – OBG)” algoritması da analiz seyrek kodlama probleminin çözümü için kullanılan algoritmalarla örnek olarak verilebilir [3].

Analiz sözlük öğrenme probleminin ikinci aşaması analiz sözlük güncelleme ya da diğer bir deyişle operatör güncelleme aşamasıdır. Bu problem de şu şekilde verilir :

$$\{\hat{\Omega}\} = \min_{\Omega} \|X - Y\|_F^2 ; \Omega_{\Lambda_i} x_i = 0 \quad \forall 1 \leq i \leq N ,$$

$$\text{Rank}(\Omega_{\Lambda_i}) = d - r \quad \forall 1 \leq i \leq N , \|w_j\|_2 = 1 \quad \forall 1 \leq j \leq K \quad (2.9)$$

Denklem (2.9) ile verilen operatör güncelleme problemi, satırları normalize edilmiş bir Ω operatörü için, şu şekilde de verilebilir:

$$\min_{\Omega} \|Y - X\|_F^2 ; \|\Omega x_i\|_0 \leq K - l \quad \forall i = 1, 2, \dots, N \quad (2.10)$$

Sentez modelinde olduğu gibi analiz modelinde de operatör öğrenme problemi NP-zor bir problemidir ve yine bu problemin çözümü için sunulan değişik buluşsal algoritmalar vardır. Problem çözümünde buluşsal (sezgisel) bir yaklaşımın kullanıldığı algoritmalarla buluşsal algoritmalar denir. Bu algoritmalar bir problemin çözümü için klasik metodların yavaş kaldığı durumlarda probleme daha hızlı çözüm bulmak için ya da klasik metodların tam çözümü bulmada yetersiz kaldığı durumlarda yaklaşık çözümü bulmak için kullanılırlar. Buluşsal algoritmalar en iyi sonucu bulmaktan ziyade genellikle makul bir süre içerisinde yaklaşık bir çözüm elde ederler [24]. Analiz sözlük öğrenme problemine yaklaşık bir çözüm sunan bu algoritmalarla örnek olarak Analiz K-SVD (Analysis K-SVD) [3], kısıtlı aşırı tam analiz operatör öğrenme (Constrained overcomplete Analysis Operator Learning - CAOL) [19] , Sıralı En az Özdeğerler (Sequential Minimal Eigenvalues) [20] algoritmaları verilebilir. İlerleyen bölümde sentez ve analiz modelinin zorlu aşamaları olan sözlük öğrenme ve operatör öğrenme problemleri daha detaylı bir şekilde incelenecektir.

Bunların yanı sıra analiz modelinin genelleştirilmiş bir şekli olan dönüşüm modelinin diğer modellerdeki sözlük, operatör öğrenme problemine eşdeğer olan seyrekleştirici dönüşüm öğrenme problemi de incelenecektir.

2.3 Seyreklik Tabanlı Görüntü İşleme Uygulamaları

Görüntü işleme alanında seyreklik modelinin kullanıldığı sınıflandırma, gürültü giderme, görüntü sıkıştırma, görüntü ayrıştırma, görüntü geri çatma, yüz tanıma, iç boyama, manyetik rezonans görüntüleme gibi birçok uygulama bulunmaktadır. Bu bölümde seyreklik modelinin kullanıldığı bu görüntü işleme uygulamalarından bazılarına değinecektir.

2.3.1 Seyreklik tabanlı sınıflandırma

Seyreklik tabanlı sınıflandırma uygulamalarında genel olarak sınıflandırma-yönelimli sözlükler kullanılmaktadır. Bu uygulamalarda araştırmacılar tarafından ilk olarak sentez seyreklik tabanlı sözlük öğrenmeye dayalı metodlar çalışılmıştır. Bu metodlar genel olarak ayırt edicilik (sınıflandırıcı) özelliğini doğrudan sözlüğe katan ya da sözlüğe bu ayırt edicilik özelliğini seyrek katsayılara katmak suretiyle kazandıran metodlar olmak üzere ikiye ayrılırlar [25]. Bu sınıflandırma-yönelimli sözlükler vasıtasıyla gösterilim hatasını esas alarak ya da seyrek katsayıları ilgili görüntünün öznitelikleri gibi kullanarak sınıflandırma yapılmaktadır.

Yang ve diğ. [26] çalışmasında her bir sınıf için ayrı bir sözlük öğrenilmiş ve test aşamasında bu sözlüklerin birleştirilmesinden oluşan bütün bir sözlük kullanılmıştır. Bu çalışmada esas alınan problem denklem (2.11)' de görülmektedir.

$$D_i = \min_{D_i} \|X_i - D_i A_i\|_2^2 + \lambda \|A_i\|_1 \quad ; \quad \|d_j^i\|_2 \leq 1, \forall j = 1, 2, \dots, K \quad (2.11)$$

Burada $X_i \in \mathbb{R}^{dxN_i}$ matrisi, sütunlarında i . sınıfa ait eğitim görüntülerini içerir. d_j^i i . sınıfa ait alt sözlük olan $D_i = [d_1^i, d_2^i, \dots, d_K^i] \in \mathbb{R}^{dxK}$ sözlüğünün j . sütununu gösterir. $A_i = [a_1^i, a_2^i, \dots, a_{N_i}^i] \in \mathbb{R}^{KxN_i}$ matrisi de i . sınıfa ait seyrek katsayı vektörlerini belirtir. Alt sözlüklerin birleştirilmesi ile oluşan bütün sözlük $D = [D_1, D_2, \dots, D_C]$ şeklinde gösterilebilir.

Zhang ve Li [27] ayırtedici K-SVD çalışmasında ise her sınıf için ayrı bir sözlük öğrenip bunları birleştirip bütün bir sözlük kullanmaktansa, problemi tanımladığı

maliyet fonksiyonuna bazı eklemeler yaparak bütün bir ayırt edici sözlük öğrenir. Bu durumda ele alınan maliyet fonksiyonu denklem (2.12)' deki gibi gösterilir.

$$(D, W, A) = \min_{D, W, A} \|X - DA\|_F^2 + \lambda_1 \|H - WA\|_F^2 + \lambda_2 \|A\|_1 + \lambda_3 \|W\|_F^2 \quad (2.12)$$

Burada $H = [h_1, h_2, \dots, h_N] \in \mathbb{R}^{C \times N}$ eğitim görüntülerinin etiketlerini gösterir. W sınıflandırıcıya ait parametredir. $\lambda_1, \lambda_2, \lambda_3$ ise sabitlerdir. Bu maliyet fonksiyonunda üç farklı değişkenin optimizasyonu vardır. Bu optimizasyon işlemi şu aşamalarda gerçekleştirilir [27] :

1. K-SVD algoritmasından elde edilen D ve A matrisleri başlangıç olarak atanır.
2. Sabit D ve A için W bulunur.
3. Sabit D ve W için A güncellenir.
4. Sabit A ve W için D güncellenir.
5. Bir durma kriteri sağlanana kadar 2.,3., ve 4. adımlar tekrarlanır.

Bu metodlar sınıflandırma, yüz tanıma gibi uygulama alanlarında kullanılmaktadır.

2.3.2 Seyreklik tabanlı görüntü ayrıştırma

Gözlemlenen bir y görüntüsünün, y_1 ve y_2 gibi iki farklı alt sinyallerden oluştuğu ($y = y_1 + y_2$) durumda, y_1 görüntüsü ayrı bir seyrek gösterilimle, y_2 görüntüsü ayrı bir seyrek gösterilimle ifade edilebilir yani gözlemlenen görüntü bu iki görüntüye ayrıştırılabilir [1]. Bu işleme görüntü ayrıştırma denir. Bu işlem için şu problem çözülmeye çalışılır:

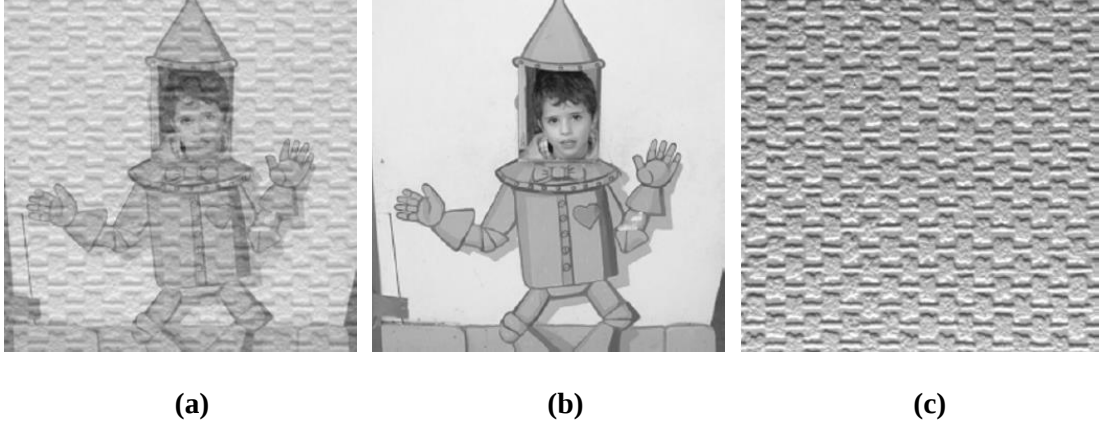
$$\min_{x_1, x_2} \|x_1\|_0 + \|x_2\|_0 ; \|y - A_1 x_1 - A_2 x_2\|_2^2 \leq \epsilon_1^2 + \epsilon_2^2 \quad (2.13)$$

Bulunan x_1 ve x_2 seyrek katsayılar ile $\hat{y}_1 = A_a x_1$ ve $\hat{y}_2 = A_a x_2$ şeklinde ayrıştırılır. Bu çözüm yaklaşımı ilk olarak, kaynak ayrıştırma problemi ile oldukça sık karşılaşılan bir alan olan akustik işaret işleme alanında kullanılmıştır. Daha sonra görüntü işleme alanında kullanılmaya başlanmıştır [1]. Görüntü ayrıştırma uygulamasında ele alınan denklem (2.14)' deki problem de bu yaklaşımla çözülmeye çalışılır.

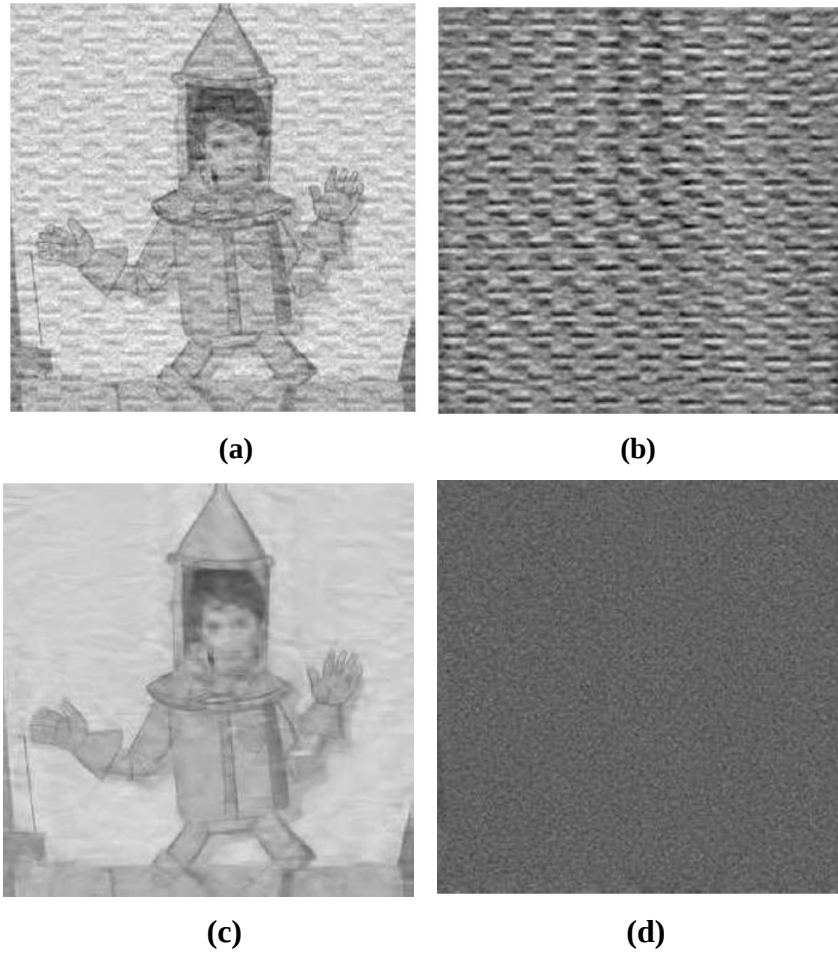
$$\min_{x_a} \|x_a\|_0 ; \|y - A_a x_a\|_2^2 \leq \delta \quad (2.14)$$

Burada $x_a = [x_1^T, x_2^T]^T$ matrisini, $A_a = [A_1, A_2]$ matrisini gösterir. A_1 sözlüğü $\hat{y}_1$ görüntüsünü oluştururken kullanılan atomları, A_2 sözlüğü ise $\hat{y}_2$ görüntüsünü

oluştururken kullanılan atomları içeren sözlüktür. Şekil 2.3' te bir karikatür görüntüsü ve bir dokuma desenli görüntüden oluşan bir test görüntüsü görülmektedir. Şekil 2.4' te ise bu görüntünün toplanır beyaz gauss gürültüsü altında gözlemlenmesi durumunda bu iki alt görüntüye ve gürültü bileşenine ayrıştırılmış hali görülmektedir [1].



Şekil 2.3 : (a) Test görüntüsü (b) Karikatür görüntüsü (c) Dokuma desenli görüntü [1].



Şekil 2.4 : Görüntü ayırma örneği. (a) Gürültülü test görüntüsü. (b) Ayrıştırılmış dokuma desenli görüntü. (c) Ayrıştırılmış karikatür görüntüsü. (d) Ayrıştırılmış gürültü bileşeni [1].

Bu ayrıştırma işleminde karikatür görüntüsü için eğricik sözlüğü, dokuma desenli görüntü için ayrık kosinüs dönüşümü (DCT) sözlüğü kullanılmıştır. Burada görüntülerin iyi bir şekilde ayrıştırılabilmesi için, dikkat edilmesi gereken nokta kullanılacak olan sözlüklerin diğer görüntüyü ifade edebilecek atomları içermemesi ya da oldukça az içermesidir [1].

2.3.3 Seyreklik tabanlı iç boyama

Bir x_0 seyrek katsayı vektörü ile $y_0 = Ax_0$ şeklinde seyrek gösterilime sahip bir y_0 sinyali varsayalım. Bu işaretin bir M maskeleme operatörü ile bazı örnekleri kaybedilmiş (maskelenmiş) bir $y = My_0$ gözlemini ele alırsak, y_0 sinyalini elde etme problemi denklem (2.15)' teki gibi verilir [1].

$$\min_{x_0} \|x_0\|_0 ; y = MAx_0 \quad (2.15)$$

Buradan bulunan $\hat{x}_0$ ile $\hat{y}_0 = A\hat{x}_0$ elde edilebilir. Bu sonuç şöyle açıklanabilir: Eğer y_0 işaretine ilişkin k_0 adet sıfır olmayan elemana sahip x_0 seyrek katsayı vektörü en seyrek durumda ise, gözlemlediğimiz y işaretine ilişkin seyrek katsayı vektörünün sıfır olmayan eleman sayısı bu k_0 değerinden yüksek olamayacaktır. Bundan dolayı maskeleme işleminden sonra kaybedilmiş örnekleri tekrar elde edebilmek mümkün olmaktadır [1].

Şekil 2.5' te $\sigma = 20$ standart sapma değerine sahip bir beyaz gauss gürültüsü ile maskelenmiş ve belli sayıda pikseli kaybedilmiş görüntüler ve bu görüntülerin iç boyama ile elde edilmiş görüntüleri bulunmaktadır. İç boyama işlemi ile, maskelenmiş olan pikseller uygun bir şekilde doldurularak daha düzgün görüntüler elde edilmiştir [1].

Şekil 2.5 ile sonucu verilen deneyde iç boyama işlemi örtüşme olmadan gerçekleştirilmiştir. Bir başka deneyde ise aynı görüntü yine aynı standart sapma değerine sahip beyaz gauss gürültüsü ile maskelenmiş ve bu görüntüler bu sefer tam örtüşme yapılmak suretiyle iç boyama işlemine tabi tutulmuştur [1]. Bu deneye ait, iç boyama işleminden elde edilen sonuçlar da Şekil 2.6' da görülmektedir. Bu şekilden de anlaşılmaktadır ki tam örtüşme ile gerçekleştirilen iç boyama örtüşme olmadan gerçekleştirilen iç boyamaya göre daha iyi sonuçlar vermektedir [1].



(a)

(b)

(c)



(d)

(e)

(f)

Şekil 2.5 : (a) %25, (b) %50, (c) %75 pikseli maskelenmiş görüntüler. İç boyama ile elde edilmiş görüntüler: (d) RMSE:14.55, (e) RMSE:19.61, (f) RMSE:29.7



(a)

(b)

(c)

Şekil 2.6 : Tam örtüşme yapılarak gerçekleştirilen iç boyamayla elde edilmiş görüntüler. (a) RMSE: 9.0, (b) RMSE : 11.55, (c) RMSE: 18.18

3. ANALİZ VE SENTEZ SEYREKLİK İÇİN SÖZLÜK ÖĞRENME

Çalışmanın bu bölümünde seyrek gösterilim modellerinin önemli aşamaları olan sözlük, operatör ve seyrekleştirici dönüşüm öğrenme problemleri ele alınacaktır.

3.1 Sentez Seyreklik Modelinde Sözlük Öğrenme

İkinci bölümde söylendiği üzere, bir işaret, bir sözlük ve bir seyrek katsayı vektörü vasıtası ile sentez seyreklik modeli kullanılarak gösterilebilir. Bu modelde sözü edilen sözlük matrisi, atom olarak isimlendirilen sütun vektörlerinin birleşiminden oluşmuştur. İkinci bölümdeki denklem (2.1)' de verilen seyrek gösterilimi sağlayacak şekilde bu atomların birkaçının doğrusal birleşimi ile seyrek gösterilimi aranan işaret yaklaşık bir şekilde gösterilmiş olur.

Ele alınan işaret $y \in \mathbb{R}^M$ şeklinde gösterilen M boyutlu bir vektör ise ve aşırı tam bir sözlük matrisi $D \in \mathbb{R}^{M \times K}$ ve az sayıda sıfır olmayan elemanlar içeren bir seyrek katsayı vektörü de $x \in \mathbb{R}^K$ şeklinde gösterilirse sentez seyreklik modeline göre bu işaretin $y = Dx$ şeklinde ya da $\|y - Dx\|_p \leq \epsilon$ koşulunu sağlamak üzere yaklaşık bir ifadeyle $y \approx Dx$ şeklinde seyrek gösterilimi vardır. $\{d_j\}_{j=1}^K$ aşırı tam sözlüğün atomlarını göstermek üzere bu seyrek gösterilim ifadesi denklem (3.1)' deki gibi de gösterilebilir.

$$\tilde{y} = \sum_{k=1}^K x(k)d_k = Dx \quad (3.1)$$

Burada $\tilde{y}$ işareti sözlük ve seyrek katsayı vektörü kullanılarak elde edilen seyrek gösterilimi temsil etmektedir. Ayrıca $x(k)$ değeri de sözlüğün k . atomuna karşılık gelen katsayı değerini göstermektedir. Daha önce de bahsedildiği üzere işaretin az sayıda atomun doğrusal birleşimi ile ifade edilebilmesi açısından bu seyrek katsayı vektörünün az sayıda elemanının (örneğin $s \ll K$) sıfırdan farklı olması istenmektedir. Seyrek gösterilimi aranan orijinal işaret y ve seyrek gösterilim ile elde edilen $\tilde{y}$ işareti arasındaki fark 'gösterilim hatası' olarak isimlendirilir ve denklem (3.2)' deki gibi ifade edilir.

$$r = y - \tilde{y} = y - Dx \quad (3.2)$$

Sentez seyreklik modelinde uygun sözlük ve seyrek gösterilim vektörü bulma problemi sözlük öğrenme olarak adlandırılır ve çeşitli algoritmalar kullanılarak iki aşama da çözülür. Bunlardan ilki seyrek kodlama diğeri ise sözlük güncelleme aşamasıdır. Bu bölümde sözlük öğrenme problemi genel olarak ele alınacak ve bu problemin çözümü için sunulan K-SVD algoritması dördüncü bölümde anlatılacaktır.

Sözlük öğrenme problemi denklem (2.2)'deki gibi genel bir şekilde ifade edilebilir. Bu denklemi tekrar yazmak gerekirse;

$$\min_{D,X} \|Y - DX\|_F^2, \|X_i\|_0 \leq s \quad \forall i \quad (3.3)$$

Burada $Y \in \mathbb{R}^{M \times N}$ matrisi sütunlarında seyrek gösterilimi aranan M uzunluklu N adet işaretleri içeren bir matristir ve eğitim kümesi olarak da isimlendirilir. $X \in \mathbb{R}^{K \times N}$ matrisi bu kümedeki her bir işarete karşılık gelen seyrek katsayı vektörlerinden oluşan bir matristir. Ele alınan problemi yorumlamak gerekirse, $\min_{D,X} \|Y - DX\|_F^2$ kısmı D sözlüğü ve X seyrek katsayı matrisi kullanılarak bulunacak olan seyrek gösterilimin elimizdeki Y gözlemine olabildiğince sadık kalmasını sağlayacak olan kısmı ifade eder. $\|X_i\|_0 \leq s, \forall i$ kısmı ise her bir işareti göstermek üzere kullanılacak olan seyrek katsayı vektörlerinin en fazla s elemanının sıfırdan farklı olmasını sağlayan yani seyrekliğin zorlandığı kısımdır.

(3.3) denkleminde görüldüğü üzere sözlük öğrenme için verilen bu optimizasyon probleminde iki adet değişken vardır. Bunlar sözlük matrisi D ve seyrek katsayı matrisi X ' tir. Bu iki değişkenin aynı anda güncellenmesi zorlu bir problem olacağından bir çok algoritmada bu problemin iki aşamaya ayrılarak çözülmesi sunulmuştur. Bu iki aşama;

- D sözlük matrisinin sabit tutulup, X katsayı matrisinin bulunması,
- X katsayı matrisinin sabit tutulup, D sözlük matrisinin bulunması/güncellenmesi

şeklinde gösterilebilir.

İlk aşamada genelde analitik bir yöntem ile elde edilmiş D sözlük matrisi kullanılarak çeşitli vektör seçim algoritmaları ile bir X katsayı matrisi bulunur. İkinci aşamada ise bulunan bu X matrisi kullanılarak yeni bir D sözlük matrisi bulunur ya da başka bir

deyişle sözlük matrisi güncellenir. Literatürdeki algoritmaların birçoğu ikinci aşama olan sözlük matrisinin güncellenmesi aşamasında çeşitlilik gösterir. Sözlük öğrenmede kullanılan en çok bilinen algoritmalar ‘Optimal Yönler Metodu (MOD)’, ‘En Büyük Sonsal (MAP)’ ve ‘K-adet Tekil Değer Ayırıştırma (K-SVD)’ şeklinde örneklendirilebilir. MOD algoritmasında sözlük güncellenmesi sırasında en az kareler çözümü kullanılmaktadır. MAP algoritmasında ise D sözlük matrisi oluşturulurken olasılıksal hesaplamalar kullanılmaktadır. Bu algoritmaların arasında en yaygın olarak kullanılan K-SVD algoritmasının açıklaması daha detaylı bir şekilde dördüncü bölümde anlatılacaktır.

3.2 Analiz Seyreklik Modelinde Sözlük Öğrenme

3.2.1 Operatör öğrenme

Bir önceki bölümde kısaca değinildiği üzere analiz modeli sentez modeline göre daha sonraları araştırılmaya başlanmış bir modeldir. Sentez modelde işaret bir sözlük ve bir seyrek vektör ile sentezlenmeye çalışılırken, analiz modelde işaret bir operatör (analiz sözlük) ile çarpılarak seyrek olduğu varsayılan bir vektör uzayına taşınır. Bu seyrek vektörün sıfır olan elemanları işaretin dik olduğu alt uzayı temsil ederken bu uzayın dışında kalan uzay da işaretin ait olduğu alt uzayı temsil eder. Analiz modelinde amaç işareti seyrek bir şekilde gösterebilecek bir operatör öğrenmektir. Bu problem ikinci bölümde (2.6) denklemi ile verildiği üzere Ω ve x değişkenleri üzerinden bir optimizasyon problemi içerir. (2.6) denklemini burada tekrar göstermek gerekirse :

$$\min_{\Omega, x} \|Y - X\|_F^2 ; \|\Omega x_i\|_0 \leq K - l \quad \forall i = 1, 2, \dots, N \quad (3.4)$$

Bu problemi iki değişken üzerinden çözmeye çalışmak zorlu bir yaklaşım olacağından çalışmaların bir çoğunda bu aşama değişkenlerden birinin sabit tutulup diğerinin güncellenmesi yaklaşımı ile kolaylaştırılmıştır. Böylece operatör öğrenme problemi de sözlük öğrenme problemi gibi iki aşamaya bölünür. Analiz modelinde operatör öğrenme probleminin ilk aşaması sentez modelde olduğu gibi analiz seyrek kodlama aşamasıdır. Verilen bir başlangıç operatörü ile bir takip algoritması kullanılarak bu problem çözülür. İkinci aşama ise operatör güncelleme aşamasıdır.

Operatör öğrenme problemi sözlük öğrenme problemi gibi NP-zor bir problemdir ve bu problemin çözümü için sunulan değişik algoritmalar vardır. Bu algoritmalarda problem ya l_0 normu kullanılarak çözülmeye çalışılmış ya da konveks olarak

hafifletmiş hale getirmek için l_0 normu yerine l_1 normu kullanılmıştır. Ophir ve diğ. [20] çalışmasında l_0 normunu kullanırken bir diğer çalışmada ise Yaghoobi ve diğ. [19] çalışmasında, operatör öğrenme problemini konveks olarak hafifletip ΩX ' in l_1 normunu en aza indirmeye çalışmışlardır. Ancak burada vurgulanması gereken önemli bir nokta vardır ki çözümün tamamen sıfırlardan oluşmuş bir operatör ($\Omega = 0$) olması da muhtemeldir. Ancak bu çözüm faydasız, sıradan bir çözümdür. Araştırmacılar çalışmaların bir çoğunda bu gibi faydasız çözümleri elemek amacıyla operatörlerin bir kümeye ait olması için zorlanması gerektiğini önermişlerdir. Değişik çalışmalarda bu kümenin sahip olması gerektiği düşünülen özellikler de değişiklik göstermiştir. Örneğin Yaghoobi ve diğ. [19] çalışmasında operatör öğrenme problemini (3.5) 'teki gibi tanımlayıp, operatörün ait olması gereken kümenin de (3.6) 'da görüldüğü gibi olması gerektiğini önermişlerdir.

$$\min_{\Omega \in \mathcal{C}, X} \frac{\lambda}{2} \|Y - X\|_F^2 + \|\Omega X\|_1 \quad (3.5)$$

$$\mathcal{C} = \{\Omega : \Omega^T \Omega = I \quad \text{ve} \quad \|w_k\|_2 = 1, \forall k\} \quad (3.6)$$

Burada w_k , Ω operatörünün k . satırındaki atomu temsil eder. Bu küme 'Düzgün Normalize Edilmiş Sıkı Çerçeve (Uniformly Normalized Tight Frame-UNTF) ' olarak isimlendirilmektedir. Bulunan operatörün bu kümeye dahil olmasına zorlanması halinde yukarıdaki gibi faydasız çözümler elenmiş olacaktır.

Benzer olarak bir diğer çalışma olan Elad ve diğ. [3] çalışmasında ise seyreklik ölçüsü olarak l_0 normu kullanılmış ve operatör öğrenme için analiz alt uzayının boyutunu tahmin eden eş-rank ölçümüne dayalı bir metod önerilmiştir. Bu çalışmada sunulan Analiz K-SVD algoritması hesapsal adımlar açısından sentez modelde ki K-SVD algoritmasına paraleldir. Ayrıca operatör öğrenme süreci Yaghoobi ve diğ. [19] çalışmasındaki gibi kısıtlı bir optimizasyon probleminin çözümü şeklinde düşünülmüştür. Yine [19] çalışmasındaki gibi Analiz K-SVD çalışmasında da çözüm olarak kabul edilebilecek operatörler bir kümeye ait olacak şekilde gösterilebilir. Bu çalışmada da faydasız çözümlerin elenmesini sağlayacak şekilde tanımlanan küme (3.7) 'deki gibi gösterilebilir [17].

$$\mathcal{C} = \{\Omega : \text{rank}(\Omega_{\Lambda_n}) = M - r \quad \text{ve} \quad \|w_k\|_2 = 1, \forall k\} \quad (3.7)$$

Operatör öğrenme algoritmaları arasında en yaygın olarak bilinen Analiz K-SVD algoritması dördüncü bölümde incelenecektir.

3.2.2 Seyrekleştirici dönüşüm öğrenme

Araştırmacıların sentez ve analiz seyreklik modeline göre daha sonraları dikkatini çeken ve seyrek gösterilimde kullanılan bir başka model de seyrekleştirici dönüşüm modelidir. Analiz modelinin genelleştirilmiş hali olan bu model, işaretlerin dönüşüm bölgesinde yaklaşık olarak seyrek gösterilebileceğini önerir [15].

Seyrekleştirici dönüşüm modelinde seyrek gösterilim hatası olarak isimlendirilen model hatasının orijinal işaret domeninde değil de dönüşüm ya da analiz domeninde gösterildiği bir maliyet fonksiyonu en aza indirilmeye çalışılır. Dönüşüm öğrenme problemi tanımlarken kullanılan bu küçük fark analiz operatör öğrenme çerçevesiyle karşılaştırıldığında bazı önemli avantajlar sağlar. Bu avantajlardan en önemlisi, yaygın bilinen analiz operatör öğrenme algoritmalarının hesapsal olarak yükü ağır olan analiz seyrek kodlama aşamasının karmaşıklığını yüksek oranda azaltan bir eşikleme aşaması ile değiştirilmesidir.

Ravishankar ve Bresler' in [15] çalışmasında tam (kare) dönüşüm öğrenme problemi ele alınmıştır. Daha sonra ise [16] çalışmasında yine Ravishankar ve Bresler aşırı tam dönüşüm öğrenme probleminin çözümü için bir algoritma önermişlerdir.

Verilen bir $Y \in \mathbb{R}^{M \times N}$ veri matrisi için tam (kare) seyrekleştirici dönüşüm öğrenme problemi denklem (3.8)' deki gibi verilir [15].

$$\min_{W, X} \|WY - X\|_F^2 - \lambda \log \det(W) + \mu \|W\|_F^2, \quad \|X_i\|_0 \leq s \quad (3.8)$$

Seyrekleştirici dönüşüm öğrenme için tanımlanan (3.8) denkleminde görülen optimizasyon problemi seyrek gösterilim hatasının en aza indirilmesini sağlayan ana kısım ile bazı faydasız çözümlerin elenmesini ve bazı kısıtlamaların zorlanmasını sağlayan kısımlardan oluşmuştur. Bu problemde $\min_{W, X} \|WY - X\|_F^2$ kısmı daha önce sentez ve analiz modellerindeki öğrenme problemlerinde olduğu gibi seyrek gösterilim hatasını gösterir. Bu seyrek gösterilim hatası W ve X değişkenleri üzerinden en aza indirilmeye çalışılır. Ancak olası çözümler arasında $W = 0$ ve $X = 0$ gibi faydasız çözümler de vardır. Ele alınan optimizasyon problemindeki $\log \det(W)$ kısmı W dönüşümünün tekil olmayan (tersi alınabilir) bir matris olmasının zorlandığı kısımdır ve bu kısım W dönüşümünün tüm sıfır satırlardan oluşması ya da tekrarlayan

satırlardan oluşması gibi faydasız çözümlerin elenmesini sağlayan kısımdır. Bu kısıtlama ile aynı zamanda tam ranklık da sağlanmış olur [15]. Ancak maliyet fonksiyonuna sadece bu kısıtlamanın eklenmesi yeterli derecede iyileştirmenin yapıldığını göstermez. W dönüşümünün bir satırını büyük değerle bir ölçekle diğer satırlarını daha düşük değerle bir ölçekle ölçeklediğimizi düşünelim. Bu şekilde elde edilen dönüşüm tam ranktır ve seyrek gösterilim hatası olabildiğince sıfıra yakındır ancak bu dönüşüm kötü koşullandırılmış bir dönüşümdür. Ayrıca sadece büyük ölçekli tek bir satır kadar (diğer satırlar ölçek olarak daha az olduğundan) iyi bilgi üretir. Satırlar arasındaki bu ölçek belirsizliğini gidermek ve daha fazla bilgi üretebilecek, daha iyi koşullandırılmış bir dönüşüm elde edebilmek için maliyet fonksiyonuna $\mu\|W\|_F^2$ kısmı eklenmiştir.

Sütunları $y_n \in \mathbb{R}^M$, $n = 1, 2, \dots, N$ işaret vektörlerinden oluşan $Y \in \mathbb{R}^{M \times N}$ veri matrisi için, aşırı tam bir $W \in \mathbb{R}^{K \times M}$ ($K > M$) seyrekleştirici dönüşüm öğrenme problemi (3.9)'daki gibi gösterilebilir [16].

$$\min_{W, X} \|WY - X\|_F^2 - \lambda \log \det(W^T W) + \eta \sum_{k \neq j} |\langle w_j, w_k \rangle|^p, \\ \|x_n\|_0 \leq s \quad \forall n, \|w_k\|_2 = 1 \quad \forall k \quad (3.9)$$

Denklem (3.9)'de görülen $x_n \in \mathbb{R}^K$, $n = 1, 2, \dots, N$ vektörleri ele alınan işaretlerin dönüşüm domenindeki seyrek gösterilimlerine karşılık düşerler ve $X \in \mathbb{R}^{K \times N}$ matrisinin sütunlarını oluştururlar. Aşırı tam seyrekleştirici dönüşüm öğrenme problemindeki $\min_{W, X} \|WY - X\|_F^2$ kısmı seyrek gösterilim hatasını temsil etmektedir. Problemin kalan kısmındaki $\log \det(W^T W)$ kısmı tam rank koşulunu sağlamak için, $\eta \sum_{k \neq j} |\langle w_k, w_j \rangle|^p$ kısmı da W dönüşümünün satırlarının yani atomlarının birbiriyle aynı olmasını engellemek için eklenmiştir. Bu kısımların ana optimizasyon problemine eklenmesinin sebebi daha önce de bahsedildiği gibi faydasız çözümleri elemektir [16].

Seyrekleştirici dönüşüm modelinde de diğer modellerde olduğu gibi dönüşüm öğrenme problemi iki değişkenin optimizasyonu şeklinde çözülmeye çalışılır. Ancak bu çözüm yöntemi zorlu olduğundan sözlük öğrenme ve operatör öğrenme problemlerinde olduğu gibi değişkenlerden birinin sabit tutulup diğerinin bulunmasıyla ya da ilerki adımlar için güncellenmesiyle problem basitleştirilmiş olur.

Dördüncü bölümde kare seyrekleştirici dönüşüm algoritması ve aşırı tam seyrekleştirici dönüşüm algoritması incelenecektir.

4. SÖZLÜK ÖĞRENME ALGORİTMALARI

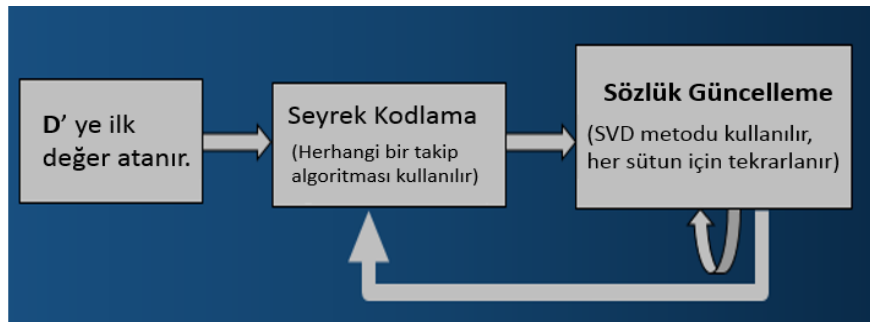
Önceki bölümlerde işaretlerin seyrek gösteriliminde kullanılan sözlük, operatör ve seyrekleştirici dönüşüm öğrenme problemlerinden bahsedilmişti. Bu bölümde ise bahsedilen problemlerin çözümü için önerilmiş, yaygın olarak bilinen algoritmalar detaylı bir şekilde incelenecektir.

4.1 K adet - Tekil Değer Ayrıştırma (K-Singular Value Decomposition- K-SVD)

K adet - tekil değer ayrıştırma algoritması işaretlerin sentez seyreklik modeline göre seyrek gösteriliminin yapılabilmesi için gerekli olan sözlük matrisinin öğrenilmesinde kullanılan bilinen en yaygın algoritmalarından biridir. Bu algoritma literatürdeki K – ortalamalar metodunun genelleştirilmiş hali olarak düşünülebilir [14]. Daha önce sözlük öğrenme problemini bir optimizasyon problemi şeklinde denklem (4.1)’ deki gibi göstermiştik. Bu problem, Y örnekler kümesinin seyrek gösterilimini sağlayacak olan en iyi sözlük matrisi D’ yi aramaktadır.

$$\min_{D,X} \{\|Y - DX\|_F^2\} , \quad \|X_i\|_0 \leq T_0, \forall i \quad (4.1)$$

K-SVD algoritmasında denklem (4.1)’ deki ifade yinelemeli bir şekilde en aza indirilmeye çalışılır. Şekil 4.1’ de bu yinelemeli yapıyı basitçe gösteren bir şematik görülmektedir. Sözlük öğrenme problemini anlatırken de bahsedildiği gibi ilk olarak D sözlük matrisi sabit tutulup en uygun X seyrek katsayı matrisi bulunur.



Şekil 4.1 : K-SVD algoritmasının basit şeması.

Tam olarak en uygun seyrek katsayı matrisini bulmak mümkün olmadığından bu aşamada yaklaşık olarak çözümü veren takip algoritmaları kullanılır. Daha önce de söylendiği gibi bu aşama ‘seyrek kodlama’ aşaması olarak isimlendirilir. Bu aşamadan sonra ikinci aşama X seyrek katsayı matrisinin sabit tutulup, daha iyi bir D sözlüğünün aranması aşamasıdır. Bu yüzden ikinci aşama sözlük matrisini güncelleme aşaması olarak isimlendirilir. Sözlük matrisinin güncellenmesi aşamasında, D sözlük matrisinin k . sütunu d_k hariç diğer sütunları sabit tutulur ve her bir adımda bu d_k sütunu yani atomu güncellenir. Bu atoma ilişkin katsayı da denklem (4.1)’ de ki hatayı en aza indirecek olan yeni bir değer ile değiştirilir. Katsayılar da yapılan bu değişiklik yönünden K-SVD algoritması, K-ortalamalar algoritmasına göre önemli derecede farklılık gösterir. K-ortalamalar algoritmasında ve diğer algoritmalarda sözlük matrisinin güncellenmesi aşamasında X seyrek katsayı matrisi tamamıyla sabit tutulur. K-SVD algoritmasında her bir adımda bir sözlük atomu güncellenirken bir taraftan da ilgili katsayıların değişmesine izin verilir. Bu değişim daha sonraki atomların güncellemelerini daha ilişkili katsayılara dayandıracağından dolayı yakınsamayı hızlandırır. K-SVD algoritmasını daha detaylı incelemek üzere denklem (4.1)’ deki fark ifadesini aşağıdaki gibi gösterebiliriz:

$$\|Y - DX\|_F^2 = \sum_{i=1}^N \|y_i - Dx_i\|_2^2$$

Ele alınan optimizasyon problemi N adet ayrı probleme dönüştürülürse denklem (4.2) elde edilir:

$$\min_{x_i} \{\|y_i - Dx_i\|_2^2\} \quad , \quad \|x_i\|_0 \leq T_0 \quad , \quad i = 1, 2, \dots, N. \quad (4.2)$$

Denklem (4.2)’ de görülen problem herhangi bir takip algoritması kullanılarak çözümlenebilir. Çözümün ideal çözüme olabildiğince yakın olması için T_0 eşik değerinin yeterince küçük olması gerekmektedir [14].

İkinci aşama sözlük matrisinin güncellenmesi ve sıfırdan farklı katsayıların yeniden hesaplanması aşamasıdır. Bu aşamada D sözlük matrisi ve X katsayı matrisi sabit tutulur ve sözlük matrisinin k . sütunu d_k ile bu sütuna denk gelen katsayı matrisinin k . satırı x_T^k ile ilgilenilir. Denklem (4.1)’ deki fark ifadesini K adet vektörün çarpımlarının toplamı olarak ifade edersek denklem (4.3)’ ü elde ederiz.

$$\begin{aligned}\|Y - DX\|_F^2 &= \left\| Y - \sum_{j=1}^K d_j x_T^j \right\|_F^2 = \left\| \left(Y - \sum_{j \neq k} d_j x_T^j \right) - d_k x_T^k \right\|_F^2 \\ &= \|E_k - d_k x_T^k\|_F^2\end{aligned}\quad (4.3)$$

Burada E_k matrisi k . atom hariç kalan N örneğe ait hatayı gösterir. K-SVD algoritmasında güncellenmiş d_k ve x_T^k vektörleri SVD (Singular Value Decomposition - Tekil Değer Ayırıştırma) yöntemi ile bulunur ve bu işlem K adet sözlük atomu için tekrarlandığından algoritma ismi K-SVD olarak anılır. SVD yöntemi ile E_k matrisine yakınsayan d_k ve x_T^k vektörleri bulunacağından (4.3) denklemi ile verilen hata da en aza indirilmiş olur. Ancak burada d_k atomu güncellenirken seyreklik ile ilgili herhangi bir kısıtlama zorlanmadığı için x_T^k vektörü rastgele bulunmuş olur. Bundan dolayı bu aşamada bazı hatalar olabilir. Bu problem için çok kolay ve sezgisel bir çözüm önerilir. Bu çözüme göre, d_k atomunu kullanan $\{y_i\}$ örneklerini gösteren yani $x_T^k(i)$ katsayılarının sıfır olmayanlarının indislerini içeren bir w_k vektörü tanımlanır.

$$w_k = \{i | 1 \leq i \leq K, x_T^k(i) \neq 0\} \quad (4.4)$$

Daha sonra $N \times |w_k|$ boyutunda $(w_k(i), i)$ girişlerinde bir değerini içeren, diğer yerlerde sıfır değerini içeren bir Ω_k matrisi tanımlansın. Bu matris ile x_T^k satır vektörü çarpıldığında Ω_k matrisinin sıfır olan girişlerinden dolayı x_T^k 'nın küçültülmüş hali olan $|w_k|$ uzunluklu x_R^k vektörü elde edilir. Aynı şekilde Ω_k matrisi ile Y örnekler kümesi çarpıldığında bir alt örnek kümesi Y_k^R oluşur. Bu durumun aynısı Ω_k matrisi E_k matrisi ile çarpıldığında da oluşur ve sonuç olarak sadece d_k atomunu kullanan örneklere ilişkin hatayı gösteren E_k^R sıkıştırılmış matrisi elde edilir. (4.3) ifadesi orijinal x_T^k ile aynı yerlerde sıfır olmayan elemanlara sahip olan x_R^k çözümünün arandığı yeni bir ifade ile gösterilirse:

$$\|E_k \Omega_k - d_k x_T^k \Omega_k\|_F^2 = \|E_k^R - d_k x_R^k\|_F^2 \quad (4.5)$$

şeklinde bir denklem elde edilir. Bu durumda direkt olarak SVD uygulanır. SVD yöntemi ile E_k^R matrisi $E_k^R = U \Delta V^T$ şeklinde üç ayrı matrise ayrıştırılır. Bu ayrıştırma işlemi yapıldıktan sonra U matrisinin ilk sütunu güncellenmiş olan $\check{d}_k$ atomu olarak ve V matrisinin ilk sütununun $\Delta(1,1)$ ile çarpılmış hali de güncellenmiş olan x_R^k katsayı vektörü olarak alınır [13]. Bu çözümler yapılırken sözlük matrisi D ' nin sütunlarının

normalize edildiği ve tüm gösterimlerdeki sıfır olmayan eleman sayısının ya aynı sayıda kaldığı ya da mümkün olduğu kadar düşürüldüğü göz önünde bulundurulmalıdır.

Güncellenmiş olan $\check{d}_k$ atomu ve x_R^k katsayı vektörü, SVD yöntemine alternatif olarak, denklem (4.6)' da verilen ifadelerin birkaç kere yinelenmesi ve ardından iki vektörün ölçeklenerek $\check{d}_k$ atomunun normalize edilmesi ile de bulunabilir [14].

$$\check{d}_k = \frac{E_k^R x_R^{kT}}{x_R^k \cdot x_R^{kT}}, x_R^k = \frac{\check{d}_k^T E_k^R}{\check{d}_k^T \cdot \check{d}_k} \quad (4.6)$$

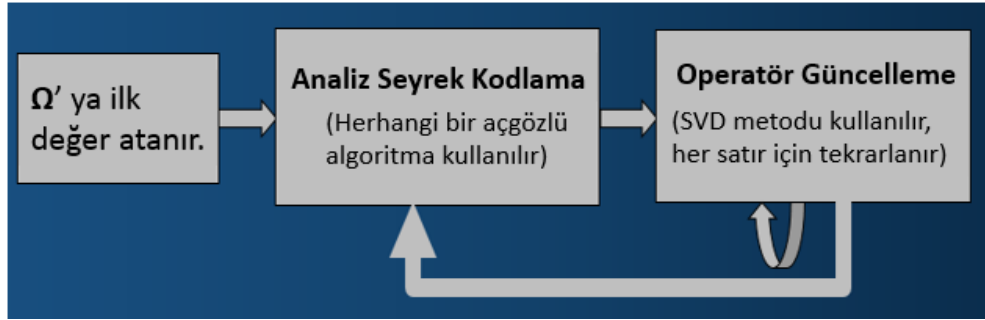
Sözlük öğrenme problemine bir çözüm olarak önerilen K-SVD metoduna ilişkin algoritma Şekil 4.2 ' de görülmektedir.

Problem : $\min_{D,X} \{ \ Y - DX\ _F^2 \} , \forall i, \ x_i\ _0 \leq T_0 .$ Giriş : Sütunları normalize edilmiş bir $D^{(0)}$ sözlük matrisi. $J=1$. Durma kriteri sağlanana kadar seyrek kodlama aşamasını ve sözlük güncelleme aşamasını tekrar et: • <i>Seyrek kodlama</i> : $\min_{x_i} \{ \ y_i - Dx_i\ _2^2 \} , \ x_i\ _0 \leq T_0 , i = 1, 2, \dots, N$ problemni her y_i örneği için bir x_i katsayı vektörü bulmak üzere bir takip algoritması kullanarak çöz. • <i>Sözlük Güncelleme</i>: $D^{(J-1)}$ sözlüğünün $k = 1, 2, \dots, K$ olmak üzere her bir sütunu için, - w_k vektörünü tanımla. - E_k hata matrisini $E_k = Y - \sum_{j \neq k} d_j x_T^j$ denklemini kullanarak hesapla. - E_k matrisinden yalnızca w_k vektöründe tanımlanan sütunları seçerek E_k^R sıkıştırılmış hata matrisini elde et. - E_k^R matrisine SVD uygulayarak $U \Delta V^T$ çarpımına ayırıştır. U matrisinin ilk sütununu $\check{d}_k$ atomu olarak seç. x_R^k katsayı vektörünü V matrisinin ilk sütununun $\Delta(1,1)$ ile çarpılmış hali olacak şekilde güncelle. • $J=J+1$. Çıkış : D sözlük matrisi.

Şekil 4.2 : K-SVD algoritması [13].

4.2 Analiz K-SVD

4.1 bölümünde anlatıldığı gibi K-SVD algoritması işaretlerin sentez seyreklik modeli kullanılarak gösteriliminin yapılmasında gerekli olan sözlük matrisinin öğrenilmesi için sunulan bir algoritmadır. Analiz K-SVD algoritması da sentez seyreklik modelinin paraleli olan analiz seyreklik modelinde kullanılan operatörün öğrenilmesi için sunulan ve hesapsal adımlar açısından K-SVD algoritmasının paraleli olan bir algoritmadır. Analiz K-SVD algoritmasına ilişkin basit şematik Şekil 4.3’ de görülmektedir.



Şekil 4.3 : Analiz K-SVD algoritmasının basit şeması.

Analiz modelinde, verilen bir $Y = [y_1, y_2, \dots, y_N] \in \mathbb{R}^{M \times N}$ örnekler kümesinde her bir örneğin, Ω operatörünün l adet satırına dik olan bir işaretin gürültülü hali olduğu kabul edilir. Burada, v_i sıfır ortalamalı toplanır beyaz gauss gürültüsü ve $x_i \|\Omega x_i\|_0 = K - l$ seyreklik şartını sağlayan işaret vektörü olmak üzere $y_i = x_i + v_i$ olduğu belirtilebilir. Analiz sözlük öğrenme probleminde de sentez sözlük öğrenme probleminde olduğu gibi başlıca iki aşamanın olduğu önceki bölümlerde söylenmişti. Bu aşamalardan ilki olan analiz seyrek kodlama aşaması GAP, BG ve OBG gibi açgözlü algoritmalarından biri kullanılarak çözülür. İkinci aşama olan operatör güncelleme aşaması çeşitli metodlar kullanılarak çözülebilir. Operatör öğrenme probleminin çözümünde kullanılan, bilinen en yaygın algoritmalarından biri analiz K-SVD algoritmasıdır. Önceki bölümlerde operatör öğrenme denklem (2.6) ya da (2.7)’deki gibi tanımlanmıştı. Rubinstein ve diğ. [21] çalışmasında operatör öğrenme problemi, benzer şekilde denklem (4.7)’deki gibi tanımlanmıştır.

$$\{\hat{\Omega}, \hat{X}\} = \min_{\Omega, X} \|X - Y\|_F^2 \quad ; \quad \|\Omega x_i\|_0 \leq K - l \quad \forall 1 \leq i \leq N, \\ \|w_j\|_2 = 1 \quad \forall 1 \leq j \leq K \quad (4.7)$$

Burada, x_i gürültüsüz işaretin kestirimini göstermektedir ve X matrisinin sütunlarını oluşturmaktadır. w_j vektörleri de Ω operatörünün satırlarını oluşturmaktadır. (4.7) denklemi ile verilen operatör öğrenme problemi sentez seyreklik modelindeki sözlük öğrenme problemi ile aynı yapıya sahiptir [3,21]. Bu yüzden analiz K-SVD algoritması, sentez modelin yaygın bilinen ve bir önceki bölümde incelenen K-SVD algoritmasına benzerlik gösteren yinelemeli bir yapıya sahiptir. Bu algoritmada da K-SVD algoritmasının sözlük öğrenme probleminde olduğu gibi operatör öğrenme problemi iki aşamalı bir yaklaşım ile çözülmeye çalışılır. İlk aşamada başlangıçta verilen sabit bir Ω operatörü kullanılarak X matrisi bulunur. İkinci aşamada ise ilk aşamada bulunan $\hat{X}$ matrisi kullanılarak Ω operatörü güncellenir. Belli bir durma kriteri sağlanana kadar süreç bu şekilde devam eder.

İlk aşama olan analiz seyrek kodlama aşamasında, y_i işaretinin gürültüsüz hali olan ve X matrisinin sütunlarını oluşturan her bir x_i vektörü ayrı ayrı bulunur. Aslında analiz seyrek kodlama aşaması, Ω operatörüne en çok dik olan l adet satırı bulma işleminden ibarettir denebilir [21]. Analiz seyrek kodlama aşamasında ele alınan problem denklem (4.8)' deki gibi gösterilir.

$$\hat{x}_i = \min_x \|y_i - x\|_2^2 ; \|\Omega x\|_0 \leq K - l, \quad \forall i \quad (4.8)$$

Denklem (4.8) ile verilen problem daha önce bahsedilen herhangi bir takip algoritması kullanılarak çözülebilir.

$\hat{X}$ matrisi bulunduktan sonra ikinci aşama olan operatör güncelleme aşamasına geçilir. Bu aşamada da analiz seyrek kodlama aşamasında olduğu gibi operatörün her bir sütunu ayrı ayrı güncellenir. Sentez modeldeki K-SVD algoritmasına benzer şekilde $\hat{X}$ matrisinin w_j vektörüne dik olan sütunlarının indislerini içeren bir J kümesi tanımlanır. Daha sonra Y matrisinin J kümesinde indislenen sütunlarını içeren bir Y_j alt matrisi tanımlanır. w_j vektörünün güncellenmesinde ele alınan problem denklem (4.9)' daki gibi verilir.

$$\hat{w}_j = \min_w \|w^T Y_j\|_2 ; \|w\|_2 = 1 \quad (4.9)$$

(4.9) denklemi ile verilen problemde aranan çözüm, Y_j matrisine SVD uygulayarak elde edilebilecek olan Y_j matrisinin en küçük tekil vektörüne karşı düşen tekil vektördür. Analiz K-SVD algoritmasında operatör güncelleme sırasında yapılan ek bir işlem daha vardır. Bu işlem ile operatörün her bir satırının, kendisi hariç kalan diğer

satırlar ile arasındaki benzerlik ölçülmektedir. Eğer bu benzerlik mutlak değer olarak $1 - \delta$ değerinden büyükse, rastgele üretilen bir satır ile değiştirilmektedir.

Analiz K-SVD algoritması sentez modelde bahsedilen K-SVD algoritmasına oldukça benzerdir. Bu iki algoritma arasındaki ana fark analiz K-SVD algoritmasında en büyük özdeğere karşılık düşen tekil vektörün değil de en küçük özdeğere karşılık düşen tekil vektörün kullanılmasıdır. Operatör öğrenme problemine bir çözüm bulan Analiz K-SVD metoduna ilişkin algoritma Şekil 4.4'te görülmektedir.

Problem : $\min_{\Omega, X} \|X - Y\|_F^2 ; \|\Omega x_i\|_0 \leq K - l \quad \forall 1 \leq i \leq N ,$
 $\|w_j\|_2 = 1 \quad \forall 1 \leq j \leq K$

Giriş : $Y \in \mathbb{R}^{M \times N}$, $\Omega_0 \in \mathbb{R}^{K \times M}$, l : istenen seyreklik seviyesi , k : iterasyon sayısı.

- $\Omega = \Omega_0$
- *for* $n = 1, \dots, k$
 - *Analiz seyrek kodlama*
 - $\min_x \|y_i - x\|_2^2 ; \|\Omega x\|_0 \leq K - l, \forall i$ problemini, her y_i için, açgözlü algoritmalarından birini kullanarak çöz.
 - *for* $j = 1, \dots, K$
 - $J = (\hat{X}' \text{in } w_j \text{ vektörüne dik olan sütunlarının indisleri})$
 - *Operatör güncelleme*
 - $\min_w \|w^T Y_j\|_2 ; \|w\|_2 = 1$ problemini SVD kullanarak çöz.
 - $\Omega(j. \text{satır}) = w_j^T$
 - *end for*
 - *end for*

Çıkış : Ω operatörü.

Şekil 4.4 : Analiz K-SVD algoritması [3].

4.3 Seyrekleştirici Dönüşüm Öğrenme Algoritması

Sentez ve analiz modeline alternatif olarak önerilen bir başka modelin de seyrekleştirici dönüşüm modeli olduğu önceki bölümlerde söylenmişti. Bu model ile ilgili genel ve detaylı bilgiler ikinci ve üçüncü bölümde verilmişti. Bu bölümde ise seyrekleştirici dönüşüm modelinin en zorlu problemi olan seyrekleştirici dönüşüm öğrenme probleminin çözümü olarak sunulan Ravishankar ve diğ. [15] ve [16] çalışmalarındaki algoritma incelenecektir.

Seyrekleştirici dönüşüm modelinde verilen bir $Y \in \mathbb{R}^{M \times N}$ veri matrisi için tam seyrekleştirici dönüşüm öğrenme problemi daha önce denklem (4.10)'daki gibi ifade edilmişti.

$$\min_{W,X} \|WY - X\|_F^2 - \lambda \log \det(W) + \mu \|W\|_F^2, \quad \|x_i\|_0 \leq s \quad (4.10)$$

Aşırı tam seyrekleştirici dönüşüm öğrenme problemi de, denklem (4.10)'da görülen problemde küçük değişiklikler yapılmasıyla denklem (4.11)'deki gibi elde edilmiştir.

$$\min_{W,X} \|WY - X\|_F^2 - \lambda \log \det(W^T W) + \eta \sum_{k \neq j} |\langle w_k, w_j \rangle|^p, \\ \|x_n\|_0 \leq s \quad \forall n, \|w_k\|_2 = 1 \quad \forall k \quad (4.11)$$

Verilen her iki problem de sözlük öğrenme ve operatör öğrenme problemlerinde olduğu gibi iki aşama da çözülmeye çalışılır. Bu aşamalardan ilki, başlangıçta tanımlanan bir W matrisi kullanılarak X matrisinin arandığı seyrek kodlama aşamasıdır. Hem tam dönüşüm hem de aşırı tam dönüşüm öğrenmede aynı problem ele alınır ve aynı çözüm yolu önerilir. Bu problem denklem (4.12)'deki gibi gösterilir.

$$\min_X \|WY - X\|_F^2; \quad \|x_i\|_0 \leq s \quad \forall i \quad (4.12)$$

Bu problemin kesin çözümü, WY matrisinin her bir sütununun en büyük s elemanının tutulmasını sağlayacak bir eşikleme işlemi ile bulunur. Bu eşikleme işlemi, diğer seyreklik modellerinde OMP, BG gibi yüksek karmaşıklıkta algoritmaların kullanıldığı seyrek kodlama aşamasını hesapsal yük açısından oldukça basitleştirmiştir. Eğer denklem (4.13)'te görüldüğü gibi l_0 normu yerine l_1 normu kullanılan bir optimizasyon problemi ele alınmış olsaydı bu durumda çözüm yumuşak eşikleme ile bulunacaktı.

$$\min_X \|WY - X\|_F^2 + \eta \sum_{i=1}^N \|x_i\|_1 \quad (4.13)$$

Bu probleme ilişkin yumuşak eşikleme çözümü şu şekilde verilir:

$$X_{ij} = \begin{cases} (WY)_{ij} - \frac{\eta}{2}, & (WY)_{ij} \geq \frac{\eta}{2} \\ (WY)_{ij} + \frac{\eta}{2}, & (WY)_{ij} < -\frac{\eta}{2} \\ 0, & \text{diğer durumlar} \end{cases} \quad (4.14)$$

İkinci aşamada, ilk aşamada bulunan X matrisi sabit tutularak seyrekleştirici dönüşüm güncellenmeye çalışılır. Beklendiği üzere, bu aşamada tam dönüşüm ve aşırı tam dönüşüm için ele alınan problemler farklılık gösterir. Bu problemler sırasıyla denklem (4.15)' te ve denklem (4.16)' da görülmektedir.

$$\min_W \|WY - X\|_F^2 - \lambda \log \det(W) + \mu \|W\|_F^2 \quad (4.15)$$

$$\min_W \|WY - X\|_F^2 - \lambda \log \det(W^T W) + \eta \sum_{k \neq j} |\langle w_j, w_k \rangle|^p; \|w_k\|_2 = 1 \forall k \quad (4.16)$$

Her iki problemin çözümü için [15] ve [16] çalışmalarında eşlenik gradyan (Conjugate Gradient-CG) metodu önerilmiştir. Tam ve aşırı tam dönüşüm öğrenme problemi için verilen maliyet fonksiyonlarındaki her bir kısmın gradyanları sırasıyla denklem (4.17) ve denklem (4.18)' de verilmiştir.

$$\begin{aligned} \nabla_W \log \det W &= W^{-T} \\ \nabla_W \|W\|_F^2 &= 2W \\ \nabla_W \|WY - X\|_F^2 &= 2WYY^T - 2XY^T \end{aligned} \quad (4.17)$$

$$\begin{aligned} \nabla_W \log \det(W^T W) &= 2W(W^T W)^{-1} \\ \nabla_W \|WY - X\|_F^2 &= 2WYY^T - 2XY^T \\ \nabla_W \sum_{k \neq j} |\langle w_j, w_k \rangle|^p &= 2p(ZW - B) \end{aligned} \quad (4.18)$$

Her iki denklemde de logaritma fonksiyonunun sürekliliğinin sağlanması için $\det W > 0$ ve $\det(W^T W) > 0$ olacak şekilde kabul edilmiştir. (4.18) denkleminde $Z \in \mathbb{R}^{K \times K}$ ve $B \in \mathbb{R}^{K \times M}$ matrisleri $z_{ij} = \langle w_j, w_k \rangle^{p-1}$ ve $b_{ij} = z_{ji} w_{ij}$ olacak şekilde verilir. Kolaylık açısından p çift doğal sayı olarak alınır. [15] ve [16] çalışmalarında sunulan tam ve aşırı tam seyrekleştirici dönüşüm öğrenme metodlarına ait algoritmalar sırasıyla Şekil 4.5' te ve Şekil 4.6' da verilmiştir. Algoritmalarından da anlaşılacağı üzere aralarındaki ana fark dönüşümün güncellendiği aşamadır.

Problem :

$$\min_{W, X} \|WY - X\|_F^2 - \lambda \log \det(W) + \mu \|W\|_F^2, \quad \|x_i\|_0 \leq s$$

Giriş : $Y \in \mathbb{R}^{M \times N}$, $W_0 \in \mathbb{R}^{M \times M}$

- *Seyrek Kodlama* : $\min_X \|WY - X\|_F^2$; $\|x_i\|_0 \leq s \ \forall i$ problemini WY matrisinin her bir sütununun en büyük s elemanının tutulmasını sağlayacak bir eşikleme işlemi ile çöz.
- *Dönüşüm Güncelleme* :

$$\min_W \|WY - X\|_F^2 - \lambda \log \det(W) + \mu \|W\|_F^2 \text{ problemini gradyanları;}$$

$$\nabla_W \log \det W = W^{-T}$$

$$\nabla_W \|W\|_F^2 = 2W$$

$$\nabla_W \|WY - X\|_F^2 = 2WYY^T - 2XY^T$$

olarak eşlenik gradyan metodunu kullanarak çöz.

Çıkış : W tam dönüşüm operatörü.

Şekil 4.5 : Tam dönüşüm öğrenme algoritması [15].

Problem :

$$\min_W \|WY - X\|_F^2 - \lambda \log \det(W^T W) + \eta \sum_{k \neq j} |\langle w_j, w_k \rangle|^p ; \|w_k\|_2 = 1 \ \forall k$$

Giriş : $Y \in \mathbb{R}^{M \times N}$, $W_0 \in \mathbb{R}^{M \times M}$

- *Seyrek Kodlama* : $\min_X \|WY - X\|_F^2$; $\|x_i\|_0 \leq s \ \forall i$ problemini WY matrisinin her bir sütununun en büyük s elemanının tutulmasını sağlayacak bir eşikleme işlemi ile çöz.

- *Dönüşüm Güncelleme* :

$$\min_W \|WY - X\|_F^2 - \lambda \log \det(W^T W) + \eta \sum_{k \neq j} |\langle w_j, w_k \rangle|^p ; \|w_k\|_2 = 1 \ \forall k$$

problemini gradyanları ;

$$\nabla_W \log \det(W^T W) = 2W(W^T W)^{-1}$$

$$\nabla_W \|WY - X\|_F^2 = 2WYY^T - 2XY^T$$

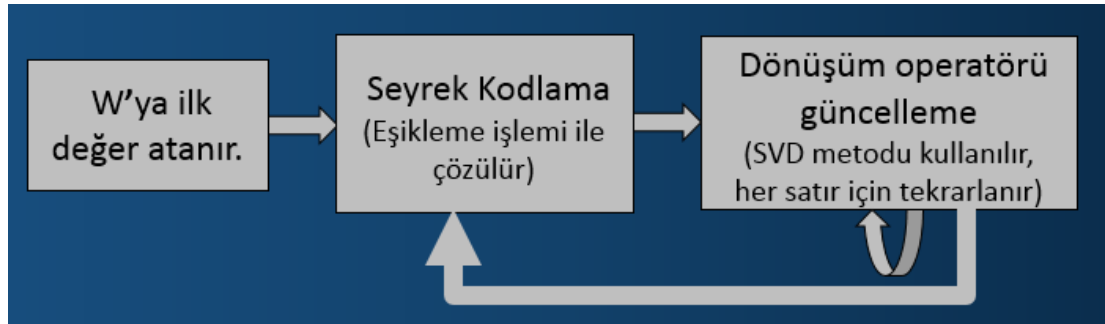
$$\nabla_W \sum_{k \neq j} |\langle w_j, w_k \rangle|^p = 2p(ZW - B)$$

olarak eşlenik gradyan metodunu kullanarak çöz.

Şekil 4.6 : Aşırı tam dönüşüm öğrenme algoritması [16].

5. YENİ YÖNTEM: SEYREKLEŞTİRİCİ DÖNÜŞÜM OPERATÖRÜ ÖĞRENME

Analiz modelin genelleştirilmiş hali olan seyrekleştirici dönüşüm modelinde, seyrekleştirme hatası olarak isimlendirilen model hatasının orijinal işaretin bulunduğu domende değil de dönüşüm ya da analiz domeninde hesaplandığı ve bu sayede analiz modele göre bazı avantajlar elde edildiği önceki bölümlerde anlatılmıştı. Bu bölümde analiz K-SVD algoritmasının ve dönüşüm modelinin avantajlarını bir araya getiren seyrekleştirici dönüşüm operatörü öğrenme algoritması “Dönüşüm K-SVD” anlatılacaktır. Bu algoritma ile analiz K-SVD algoritmasına benzer yapıda ve benzer performansta operatörler daha az hesapsal yük ile elde edilmektedir [18]. Dönüşüm K-SVD algoritmasına ilişkin basit şematik Şekil 5.1’ de görülmektedir.



Şekil 5.1 : Dönüşüm K-SVD algoritmasının basit şeması.

Ravishankar ve Bresler’in [16] çalışmasında aşırı tam dönüşüm öğrenme problemi denklem (5.1)’ de görüldüğü gibi ifade edilmiştir.

$$\min_{W,X} \|WY - X\|_F^2 - \lambda \log \det(W^T W) + \eta \sum_{k \neq j} |\langle w_k, w_j \rangle|^p, \quad (5.1)$$

$$\|x_n\|_0 \leq s \quad \forall n, \|w_k\|_2 = 1 \quad \forall k$$

Burada $Y \in \mathbb{R}^{M \times N}$ veri matrisini, $W \in \mathbb{R}^{K \times M}$ aşırı tam dönüşüm matrisini ve $X \in \mathbb{R}^{K \times N}$ de işaretlerin dönüşüm domenindeki seyrek gösterilimlerini ifade etmektedir. $s \ll K$ sabiti dönüşüm domenindeki seyreklik seviyesini gösteren dönüşüm katsayılarının sayısını temsil etmektedir.

Dönüşüm K-SVD algoritmasında ele alınan maliyet fonksiyonu (5.1) denklemindeki probleme nazaran daha basit bir problemidir. Bu problem denklem (5.2)' de görülmektedir.

$$\min_{W,X} \|WY - X\|_F^2 ; \|x_n\|_0 \leq s \quad \forall n, \|w_k\|_2 = 1 \quad \forall k, \\ \mu\{W^T\} = \max_{k \neq j} |w_k w_j^T| \leq 1 - \delta \quad (5.2)$$

Denklem (5.2)' de verilen $\|w_k\|_2 = 1$ birim norm kısıtlaması ile faydasız tüm sıfır çözümlerin olası çözümler arasından elenmesi sağlanmaktadır. $\mu\{W^T\}$ ile verilen karşılıklı uyum kısıtlaması da satırlar arasındaki benzerliği ya da tekrarlamayı engellemek için ana probleme eklenmiştir. Bu karşılıklı uyum kısıtı, dönüşüm operatörü öğrenme algoritmasının dönüşüm operatörü güncelleme aşamasında satır değişimi yapmak sureti ile gerçekleştirilmektedir. Bu yaklaşım analiz K-SVD algoritmasında uygulanan yaklaşım ile aynıdır [18].

Denklem (5.2)' de verilen seyrekleştirici dönüşüm operatörü öğrenme probleminin çözümü için dönüşüm K-SVD algoritmasında, sözlük öğrenme ve analiz operatör öğrenme algoritmalarında da yaygın şekilde kullanılan iki aşamalı yinelemeli yaklaşım kullanılmaktadır. Bu yüzden ele alınan problem W ve X değişkenleri üzerinden ayrı ayrı en aza indirilmeye çalışılmıştır. (5.3) denklemi ile ifade edilen problem, ilk aşamadaki i . iterasyonda X değişkeni üzerinden yapılan en aza indirme işlemini göstermektedir.

$$X^{(i)} = \min_X \|W^{(i-1)}Y - X\|_F^2 ; \|x_n\|_0 \leq s \quad \forall n \quad (5.3)$$

Bu aşama diğer algoritmalarındaki seyrek kodlama aşamasının eşdeğeridir. Seyrek kodlama aşaması diğer metodlara nazaran oldukça kolay olan bir eşikleme işlemi ile çözülür. Bu sert eşikleme işlemi ile WY 'nin sütunlarının en yüksek s adet elemanı kalacak şekilde tam çözüm sağlanmaktadır. Bu eşikleme işlemi, kullanıldığı algoritmalarda hesapsal yükün yüksek oranda azalmasını sağlar [18].

İkinci aşamada ele alınan seyrekleştirici dönüşüm operatörü güncelleme problemi denklem (5.4)' te görüldüğü gibi verilir. Bu problem de W üzerinden optimizasyon yapılarak en aza indirilmektedir.

$$W^{(i)} = \min_W \|WY - X^{(i)}\|_F^2 ; \|w_k\|_2 = 1 \quad \forall k \quad (5.4)$$

Burada (5.2) denklemi ile verilen problemdeki $\mu\{W^T\}$ karşılıklı uyum kısıtlaması göz önüne alınmamıştır. Bu kısıtlama (5.4) problemi çözülürken yapılacak olan satır değişimi işlemi ile gerçekleştirilecektir. (5.4) denklemi ile verilen dönüşüm operatörü güncelleme probleminin bir çözümü en az kareler yöntemidir ancak dönüşüm K-SVD algoritmasında analiz K-SVD algoritmasındaki gibi bir yaklaşım kullanılmıştır. Dolayısıyla problem W dönüşüm operatörünün her bir satırı için ayrı ayrı en aza indirilerek çözülmüştür.

Sentez ve analiz K-SVD algoritmalarında olduğu gibi burada da w_k satırının güncellenmesi sırasında yalnızca, Y veri matrisindeki bu satıra dik olduğu bulunmuş sütunlar kullanılmaktadır. Y veri matrisinin w_k satırına dik olan satırlar, x_k vektörünün sıfır olan elemanları ile bulunmaktadır. (5.4) denklemi bu durumu göz önüne alarak düzenlenirse;

$$w_k^{(i)} = \min_w \left\| w Y_k^{(i)} \right\|_2^2; \quad \|w\|_2 = 1 \quad (5.5)$$

denklemi elde edilir. Bu denklemde görülen Y_k matrisi Y matrisinin ilgili sütunlarını içeren alt matrisidir. Denklem (5.5) ile elde edilen problem, analiz K-SVD algoritmasında operatör güncelleme aşamasında ele alınan problem ile aynıdır. Burada belirtilmelidir ki; analiz K-SVD algoritmasında kolaylık sağlaması açısından ele alınan problem denklem (5.5)' tekine benzer bir hale indirgenirken, burada problemin kesin ve doğal çözümü olarak denklem (5.5) elde edilmiştir. Sonuç olarak burada ki problem ile analiz K-SVD' deki problem aynı olduğundan dönüşüm operatörünün güncellenmesi aşamasında çözüm olarak analiz K-SVD algoritmasındaki yaklaşım kullanılmıştır. Problemin kesin çözümü $Y_k^{(i)}$ matrisinin en küçük özdeğerine karşı düşen sol tekil vektörüdür. Bu çözümü bulmak için analiz K-SVD algoritmasına benzer bir şekilde SVD metodu kullanılır. K adet atom olduğundan K adet SVD işlemi yapılır bu yüzden algoritma dönüşüm K-SVD olarak isimlendirilmiştir. Dönüşüm K-SVD algoritması dönüşüm modelindeki öğrenme yaklaşımının avantajı ile analiz K-SVD algoritmasının avantajını bir araya getirmektedir. Bu avantajlardan ilki seyrek kodlama aşamasının analiz K-SVD algoritmasındakine göre oldukça basitleştirilmiş olmasıdır. Diğer avantaj ise analiz K-SVD algoritmasının satır bazlı operatör güncelleme işlemi ile her bir satırın birbirinden ayrı şekilde düzenlenmesine ve güncellenmesine olanak sağlamasıdır. Yine analiz K-SVD algoritmasındakine benzer şekilde, dönüşüm operatörünün güncellenmesi aşamasında bir satır güncellendikten

sonra, karşılıklı uyum kısıtlamasının uygulanması için ek bir adımın hesaplanması gerekmektedir. Eğer öğrenilen w_k satırına ilişkin herhangi bir $j < k$ için $|w_k w_j^T| > 1 - \delta$ ise bu satır rastgele üretilen bir satırla değiştirilir. Bu satır değişikliği işleminin de probleme eklenmesi ile son olarak elde edilen seyrekleştirici dönüşüm operatörü güncelleme problemi denklem (5.6) ile verilmektedir.

$$w_k = \min_w \|w Y_k\|_2^2 + \gamma \sum_{k \neq j} |I_k^C \cap I_j| (w_j w^T)^2 ; \|w\|_2 = 1 \quad (5.6)$$

Burada I_k ve I_j sırasıyla, eş-desteklerinde k . ve j . satırları içeren veri vektörlerinin indislerini içeren kümelerdir. Denklem (5.6)' da görülen ve ana optimizasyon problemine eklenen $\sum_{k \neq j} |I_k^C \cap I_j| (w_j w^T)^2$ kısıtlaması olası çözümlerin eş-destekleri arasında daha iyi bir ayırım yapabilmeyi sağlar. Böylece problemin çözümünde daha kararlı bir iyileştirme sağlanmış olur [18].

Dönüşüm K-SVD metoduna ilişkin algoritma Şekil 5.2' deki gibi verilmektedir.

Problem : $\min_{W, X} \|WY - X\|_F^2 ; \|x_n\|_0 \leq s \quad \forall n, \|w_k\|_2 = 1 \quad \forall k ,$
 $\mu\{W^T\} = \max_{k \neq j} |w_k w_j^T| \leq 1 - \delta$
 Giriş : $Y \in \mathbb{R}^{M \times N}$, $W_0 \in \mathbb{R}^{K \times M}$ istenilen seyreklik seviyesi s , yineleme sayısı t , δ .

- $W^{(0)} = W_0$
- for $i = 1, 2, \dots, t$
 - *Seyrek kodlama*
 - $X^{(i)} = \min_X \|W^{(i-1)}Y - X\|_F^2 ; \|x_n\|_0 \leq s \quad \forall n$ problemini $W^{(i-1)}Y$ 'ye sütun bazlı eşikleme yaparak çöz.
 - for $k = 1, 2, \dots, K$
 - $Y_k^{(i)} = (x_k^{(i)})^c$ 'nin sıfır olan elemanlarına karşı düşen Y sütunları)
 - *Dönüşüm operatörü güncelleme*
 - $w_k^{(i)} = \min_w \|w Y_k^{(i)}\|_2^2 ; \|w\|_2 = 1$ problemini SVD metodunu kullanarak çöz.
 - Eğer herhangi bir $j < k$ için $|w_k^{(i)} w_j^{(i)T}| > 1 - \delta$ ise $w_k^{(i)}$ 'yi değiştir.
 - end for
- end for

Çıkış : W seyrekleştirici dönüşüm operatörü

Şekil 5.2 : Dönüşüm K-SVD algoritması [18].

6. BENZETİM SONUÇLARI

Bu bölümde seyrek işaret işleme alanındaki birçok uygulamadan biri olan görüntü gürültüsü giderme uygulaması gerçekleştirilmiştir. Analiz K-SVD algoritması ve beşinci bölümde detaylı bir şekilde anlatılan dönüşüm K-SVD algoritması kullanılarak öğrenilen operatörler ve ayrıca analitik bir metod olan Sonlu Fark (Finite Difference - FD) kullanılarak bulunan operatörlerin doğal görüntülerdeki gürültü giderme performansları karşılaştırılmıştır. Gürültü giderme aşamasında kullanılan prosedür, yazarlarının internet sayfasından erişilebilen analiz K-SVD çalışmasından adapte edilmiştir.

Benzetimler bu alandaki en iyi sonuçları vermekten ziyade analiz K-SVD ve dönüşüm K-SVD algoritmaları arasında bir karşılaştırma yapacak şekilde gerçekleştirilmiştir.

Benzetimler 2.4 GHz' lik Intel Core i7 işlemcisine, 12 GB' lık belleğe ve 64 bitlik Windows 8 işletim sistemine sahip bir bilgisayarda MATLAB programında gerçekleştirilmiştir.

Her bir görüntüden 8×8 ' lik yamaların örtüşmeleri ile elde edilmiş $N = 20000$ örnek çıkarılmıştır. Bu örnekler ile her iki algoritmada da kullanılacak olan Y eğitim veri matrisi oluşturulmuştur. Her iki algoritmada da öğrenilen operatörler 128×64 boyutlarına sahiptirler. Benzetimlerde yineleme sayısı 20 olarak alınmıştır.

Öğrenilen operatörler, gürültülü görüntüden elde edilmiş her bir yamanın gürültüsünün giderilmesinde kullanılmıştır. Gürültüsü giderilmiş yamalar bir araya getirilerek gürültüsü giderilmiş görüntünün bütünü elde edilmiştir. Her iki algoritmada da operatör güncelleme aşamasında (5.6) denkleminde görülen problem, $\gamma = 1000$ ve $\delta = 0.05$ alınarak çözülmeye çalışılmıştır. Operatör öğrenme problemi için analiz K-SVD algoritmasında işaretlerin alt uzaylarının boyutu $r = 7$ alınarak OBG metodu kullanılırken, dönüşüm K-SVD algoritmasında $s = 30$ alınarak sert eşikleme metodu kullanılmıştır. Gürültülü görüntüleri oluştururken kullanılan toplanır beyaz gauss gürültüsünün standart sapma değerleri $\sigma = 5$, $\sigma = 10$ ve $\sigma = 20$ olarak

değişmektedir ve bu gürültü seviyeleri için PSNR değerleri sırasıyla 34.1, 28.1, 22.1 olarak bulunmuştur.

Çizelge 6.1’ de sıkça kullanılan beş farklı görüntüye ait gürültü giderme sonuçları gösterilmiştir. Bu çizelgede dönüşüm K-SVD algoritması D. K-SVD , analiz K-SVD algoritması A. K-SVD şeklinde kısaltılmıştır. Dönüşüm K-SVD algoritmasının $s = 7$ ve $s = 30$ olmak üzere iki farklı seyreklik değeri için sonuçları vermiştir. $s = 7$ için elde edilen sonuçlar diğer sonuçlara nazaran oldukça yüksektir.

Çizelge 6.1 : Beş farklı görüntüye ait gürültü giderme sonuçları.

σ	Operatör	Lena	Barbara	Peppers	Boats	House
5	D.K-SVD ($s=7$)	38.14	37.46	37.56	36.67	38.34
	D.K-SVD ($s=30$)	37.63	37.04	36.90	36.62	38.27
	A.K-SVD	37.33	37.02	36.94	36.70	38.18
	FD	36.38	34.69	35.93	35.13	36.96
10	D.K-SVD ($s=7$)	34.81	33.17	33.70	32.96	34.71
	D.K-SVD ($s=30$)	33.61	32.28	32.46	32.30	34.16
	A.K-SVD	33.34	32.09	32.27	32.06	33.91
	FD	33.01	30.47	32.17	31.65	33.55
20	D.K-SVD ($s=7$)	31.81	28.95	29.92	29.66	31.79
	D.K-SVD ($s=30$)	31.27	28.53	29.42	29.32	30.83
	A.K-SVD	29.48	27.58	27.87	27.95	29.66
	FD	30.07	26.81	28.86	28.42	30.28

Dönüşüm K-SVD algoritması kullanılarak öğrenilen operatörler genel olarak analiz K-SVD kullanılarak öğrenilen operatörlerden ve analitik FD metoduyla bulunan operatörden daha iyi sonuçlar vermiştir. Ayrıca artan gürültü seviyeleri için de dönüşüm K-SVD ile analiz K-SVD’ nin performansları arasındaki farkın daha da yükseldiği gözlenmiştir.

Şekil 6.1’ de ‘lena’ görüntüsüne ait gürültülü görüntü ve sırasıyla dönüşüm K-SVD, analiz K-SVD ve FD algoritmaları kullanılarak gürültüsü giderilmiş olan görüntüler verilmiştir. Gürültü giderme performansı açısından her bir algoritma, gürültüsü giderilmiş görüntülerin PSNR değerleri ile karşılaştırılabilir.



(a)



(b)



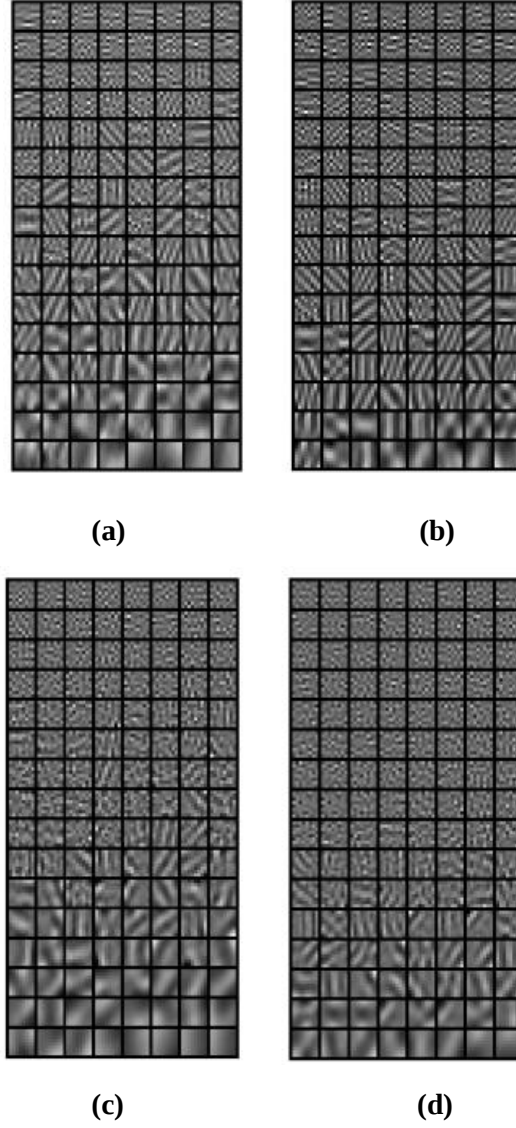
(c)



(d)

Şekil 6.1 : Gürültüsü giderilmiş görüntüler ($\sigma = 10$) (a) Gürültülü görüntü (PSNR=28.11 dB) (b) Dönüşüm K-SVD (PSNR=33.61 dB) (c) Analiz K-SVD (PSNR=33.34 dB) (d) FD (PSNR=33.01 dB).

Şekil 6.2’ de dönüşüm K-SVD ve analiz K-SVD algoritmaları kullanılarak gürültülü ‘barbara’ ve ‘lena’ görüntülerinden öğrenilen operatörler görülmektedir. Dönüşüm K-SVD ve analiz K-SVD algoritmalarında, başlangıçta problemi her ne kadar farklı formülize etmiş olsalar da, çözüm olarak sunulan yaklaşımların birbiri ile ilişkili olduğu, operatörlerin birbirlerine benzer yapılara sahip olmasından da anlaşılmaktadır. Şekil 6.2’ nin açıklama kısmında her bir operatörün, μ karşılıklı uyum değeri verilmiştir.



Şekil 6.2 : $\sigma = 10$ için öğrenilmiş operatörler (a) Analiz K-SVD, ‘Barbara’ ($\mu=0.94$) (b) Dönüşüm K-SVD ,‘Barbara’ ($\mu=0.946$) (c) Analiz K-SVD, ‘Lena’ ($\mu=0.918$) (d) Dönüşüm K-SVD , ‘Lena’ ($\mu=0.942$).

Şekil 6.3, Şekil 6.4, Şekil 6.5 ve Şekil 6.6 sırasıyla ‘barbara’, ‘house’, ‘peppers’ ve ‘boats’ görüntülerine ait gürültülü görüntüleri ve üç algoritma kullanılarak elde edilmiş gürültüsü giderilmiş görüntüleri göstermektedir. Her bir algoritmaya ilişkin gürültü giderme sonuçları, şekillerin altında belirtilmiş olan, dB birimli PSNR değerleri kullanılarak karşılaştırılabilir.



(a)



(b)



(c)



(d)

Şekil 6.3 : Gürültüsü giderilmiş görüntüler ($\sigma = 10$) (a) Gürültülü görüntü (PSNR=28.11 dB) (b) Dönüşüm K-SVD (PSNR=32.28 dB) (c) Analiz K-SVD (PSNR=32.09 dB) (d) FD (PSNR=30.47 dB).



(a)



(b)

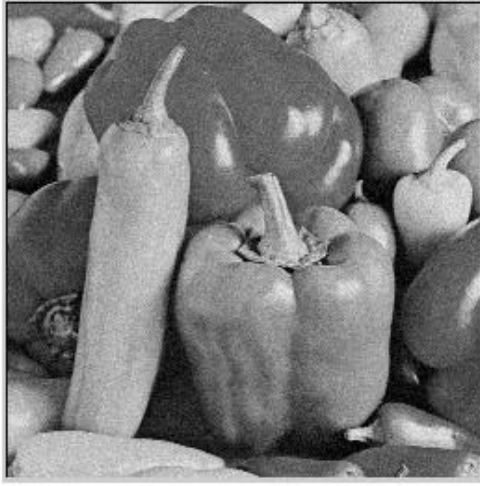


(c)



(d)

Şekil 6.4 : Gürültüsü giderilmiş görüntüler ($\sigma = 10$) (a) Gürültülü görüntü (PSNR=28.11 dB) (b) Dönüşüm K-SVD (PSNR=34.16 dB) (c) Analiz K-SVD (PSNR=33.91 dB) (d) FD (PSNR=33.55 dB).



(a)



(b)



(c)



(d)

Şekil 6.5 : Gürültüsü giderilmiş görüntüler ($\sigma = 10$) (a) Gürültülü görüntü (PSNR=28.11 dB) (b) Dönüşüm K-SVD (PSNR=32.46 dB) (c) Analiz K-SVD (PSNR=32.27 dB) (d) FD (PSNR=32.17 dB)



(a)



(b)



(c)



(d)

Şekil 6.6 : Gürültüsü giderilmiş görüntüler ($\sigma = 10$) (a) Gürültülü görüntü (PSNR=28.11 dB) (b) Dönüşüm K-SVD (PSNR=32.30 dB) (c) Analiz K-SVD (PSNR=32.06 dB) (d) FD (PSNR=31.65 dB)

Çizelge 6.2’ de ise farklı N veri boyutları için analiz K-SVD algoritmasının ve dönüşüm K-SVD algoritmasının operatör öğrenme süreleri sn. birimi cinsinden karşılaştırılmıştır. Bu süreler her iki algoritmanın tek bir yineleme ile operatör öğrenmeleri için gerekli olan sürelerdir.

İlgili işleme ilişkin sonuçlar 20 gerçeklemenin ortalaması alınarak bulunmuştur. Bu tablodan da görülmektedir ki; dönüşüm K-SVD algoritması operatör öğrenme aşaması için gerekli olan süreyi, seyrek kodlama aşamasını basit bir eşikleme işlemi ile değiştirmek suretiyle önemli derecede azaltmıştır.

Çizelge 6.2 : Farklı N veri boyutları için operatör öğrenme süreleri (sn).

N	1000	2000	4000	8000	16000	20000
D.K-SVD	2.4	2.8	3.2	4	5.2	5.8
A.K-SVD	51.4	100.0	202.1	396.6	793.0	963.6

7. SONUÇLAR

Bu tez çalışmasında, işaretlerin seyrek gösteriliminde kullanılan sentez model, analiz model ve dönüşüm modeli konuları detaylı bir şekilde incelenmiştir. Bu modellerin ele aldıkları sözlük öğrenme, operatör öğrenme ve dönüşüm öğrenme problemleri açıklanmış ve bu problemlere çözümler sunan çeşitli algoritmalar incelenmiştir.

İşaretlerin seyrek gösterilimi için kullanılan bu modellerden sentez modelde en yaygın bilinen K-SVD algoritması [13], analiz modelde K-SVD algoritmasının benzeri olan analiz K-SVD algoritması [3] ve dönüşüm modelde de Ravishankar ve Bresler'in [15,16] çalışmasında anlatılan seyrekleştirici dönüşüm öğrenme algoritması anlatılmıştır. Bu algoritmaların yanı sıra dönüşüm modelinin ve analiz K-SVD algoritmasının avantajlarını bir araya getiren dönüşüm K-SVD [18] algoritması da detaylı bir şekilde açıklanmıştır.

Ayrıca MATLAB ortamında analiz K-SVD algoritması ve dönüşüm K-SVD algoritması ile öğrenilen operatörler yine MATLAB ortamında gerçekleştirilen görüntü gürültüsü giderme uygulaması ile karşılaştırılmıştır.

Dönüşüm K-SVD algoritmasının seyrek kodlama aşamasında, dönüşüm modelinin avantajlı yaklaşımlarından biri olan eşikleme işlemi kullanılmıştır. Bu eşikleme işleminin sayesinde yapılan işlemlerin hesapsal karmaşıklıkları yüksek oranda azalmıştır. Ayrıca dönüşüm K-SVD algoritması ve analiz K-SVD algoritması ile öğrenilen dönüşümler/operatörler görsel şekilde karşılaştırıldığında dönüşüm K-SVD algoritması ile öğrenilen dönüşümlerin analiz K-SVD algoritması ile öğrenilen operatörlere yapısal olarak benzediği görülmüştür. Analiz K-SVD algoritmasına göre daha az hesapsal karmaşıklığa sahip olan dönüşüm K-SVD algoritması ile öğrenilen dönüşümler, genel olarak analiz K-SVD ile öğrenilen operatörlere yakın hatta zaman zaman daha iyi gürültü giderme performansı göstermiştir.

Bu tez çalışmasında ele alınan işaretlerin seyrek gösterilimi konusu ve bu konu kapsamında kullanılan matematiksel modeller oldukça güncel olmakla birlikte bu modellere yönelik çalışmaların araştırmacılar tarafından uzun süre ilgi görmeye devam edeceği söylenebilir.

KAYNAKLAR

- [1] **Elad, M.**, (2010). *Sparse and Redundant Representations: From Theory to Applications in Signal and Image Processing*. London : Springer.
- [2] **Rauhut, H., ve Foucart, S.**, (2013). *A Mathematical Introduction to Compressive Sensing*. Birkhauser.
- [3] **Rubinstein, R., Peleg, T., ve Elad, M.**, (2013). Analysis K-SVD: A Dictionary-Learning Algorithm for the Analysis Sparse Model. *IEEE Transactions on Signal Processing*, Vol. 61, No. 3, sayfa 661–677.
- [4] **Mallat, S., ve Zhang, Z.**, (1993). Matching pursuits with time-frequency dictionaries, *IEEE Transactions on Signal Processing*, Vol. 41, No. 12, sayfa 3397-3415.
- [5] **Chen, S., Billings, B.A., ve Luo, W.**, (1989). Orthogonal least squares methods and their application to non-linear system identification, *International Journal of Control*, Vol. 50, No. 5, sayfa 1873-1896.
- [6] **Chen, S., Donoho, D. L., ve Saunders, M. A.**, (1998) Atomic decomposition by basis pursuit. *SIAM J.Sci. Comput.*, Vol. 20, No. 1, sayfa 33-61.
- [7] **Gorodnitsky, I. F., ve Rao, B. D.**, (1997). Sparse signal reconstruction from limited data using FOCUSS: A re-weighted minimum norm algorithm. *IEEE Transactions on Signal Processing*, Vol. 45, No. 3, sayfa 600-616.
- [8] **Engan, K., Aase, S. O., ve Husoy, J. H.**, (2000). Multi-frame compression: theory and design. *Signal Processing*, Vol. 80, No. 10, sayfa 2121-2140.
- [9] **Engan, K., Aase, S. O., ve Husoy, J. H.**, (1999). Method of optimal directions for frame design. *IEEE International Conference on Acoustics, Speech, and Signal Processing*, Vol. 5, sayfa 2443-2446.
- [10] **Kreutz-Delgado, K., Rao, B. D., ve Engan, K.**, (1999). Novel algorithms for learning overcomplete dictionaries. *Proc. of the 33 Asilomar Conference on Signals, Systems and Computers*, Vol. 2, sayfa 971-975.
- [11] **Lewicki, M. S., ve Sejnowski, T. J.**, (2000). Learning overcomplete representations. *Neural Computation*, Vol. 12, No. 2, sayfa 337-365.
- [12] **Olshausen, B. A., ve Field, D. J.**, (1996). Natural image statistics and efficient coding. *Network-Computation in Neural Systems*, Vol. 7, No. 2, sayfa 333-339.
- [13] **Aharon, M., Elad, M., ve Bruckstein, A.**, (2006). K-SVD: An algorithm for designing overcomplete dictionaries for sparse representation. *IEEE Transactions on Signal Processing*, Vol. 54, No. 11, sayfa 4311-4322.
- [14] **Aharon, M.**, (2006) Overcomplete Dictionaries for Sparse Representations of Signals. *Doktora Tezi*. Israel Institute of Technology, Haifa.
- [15] **Ravishankar, S., ve Bresler, Y.**, (2013). Learning sparsifying transforms. *IEEE Transactions on Signal Processing*, Vol. 61, No. 5, sayfa 1072-1086.

- [16] **Ravishankar, S., ve Bresler, Y.,** (2013). Learning overcomplete sparsifying transforms for signal processing. *IEEE International Conference on Acoustics, Speech, and Signal Processing*, sayfa 3088 – 3092.
- [17] **Eksioglu, E. M., ve Bayir, O.,** (2014). Overcomplete sparsifying transform learning algorithm using a constrained least squares approach. *IEEE International Conference on Acoustics, Speech, and Signal Processing*, sayfa 7158-7162.
- [18] **Eksioglu, E. M., ve Bayir, O.,** (2014). K-SVD meets transform learning: Transform K-SVD. *IEEE Signal Processing Letters*, Vol. 21, No. 3, sayfa 347–351.
- [19] **Yaghoobi, M., Nam, S., Gribonval, R., ve Davies, M. E.,** (2013). Constrained Overcomplete Analysis Operator Learning for Cospase Signal Modelling. *IEEE Transactions on Signal Processing*, Vol. 61, No. 9, sayfa 2341-2355.
- [20] **Ophir, B., Elad, M., Bertin, N., ve Plumbley, M.D.,** (2011). Sequential minimal eigenvalues-an approach to analysis dictionary learning. *European Signal Processing Conference*.
- [21] **Rubinstein, R., Faktor, T., ve Elad, M.,** (2012). K-SVD Dictionary-Learning for the Analysis Sparse Model. *IEEE International Conference on Acoustics, Speech and Signal Processing*, sayfa 5405-5408.
- [22] **Fortnow, L.,** (2009). The status of the P versus NP problem. *Communications of the ACM*, Vol. 52, No. 9 ,sayfa 78-86.
- [23] **Nam, S., Davies, M. E., Elad, M., ve Gribonval, R.,** (2013). The cospase analysis model and algorithms. *Applied and Computational Harmonic Analysis*, Vol. 34, No. 1, sayfa 30-56.
- [24] **Russell, S., ve Norvig, P.,** (2010). *Artificial Intelligence : A Modern Approach*. Prentice Hall.
- [25] **Wang, D., ve Kong, S.,** (2012). *A Brief Summary of Dictionary Learning Based approach for classification*, arXiv:1205.6544 [cs.CV].
- [26] **Yang, M., Zhang, L., Yang, J., ve Zhang, D.,** (2010). Metaface Learning for Sparse Representation based Face Recognition. *IEEE International Conference on Acoustics, Speech, and Signal Processing*, sayfa 1601-1604.
- [27] **Zhang, Q., ve Li, B.,** (2010). Discriminative K-SVD for dictionary learning in face recognition. *IEEE Conference on Computer Vision and Pattern Recognition*, sayfa 2691-2698.

ÖZGEÇMİŞ

Ad Soyad : Özden Bayır

Doğum Yeri ve Tarihi : K.MARAŞ / 22.12.1989

Adres : Yakuplu merkez mah. Beylikdüzü / İstanbul

E-posta : ozdenbyr@gmail.com

Lisans : Elektronik ve Haberleşme Mühendisliği / NKÜ

Ödüller : Lisans, Bölüm Birinciliği

Lisans, Fakülte İkinciliği

MESLEKİ DENEYİM :

2013 - ... : İstanbul Teknik Üniversitesi, Araştırma Görevlisi

TEZDEN TÜRETİLEN YAYINLAR, SUNUMLAR :

- Eksioğlu, E. M., ve Bayır, O., 2014. K-SVD meets transform learning: Transform K-SVD. *IEEE Signal Processing Letters*, 21 (3), 347–351.
- Eksioğlu, E. M., ve Bayır, O., 2014. Overcomplete sparsifying transform learning algorithm using a constrained least squares approach. *IEEE International Conference on Acoustics, Speech, and Signal Processing*, 4-9 May, 2014, Florence, Italy.

