Çok-öbekli Veri için Aradeğerlemeci Ayrışım
ID for Data with Multiple Clusters

İsmail Arı, Ali Taylan Cemgil, Lale Akarun
Bilgisayar Mühendisliği Bölümü
Boğaziçi Üniversitesi, 34342 Bebek, İstanbul
{ismailar, taylan.cemgil, akarun}@boun.edu.tr

***Özetçe* —Aradeğerlemeci Ayrışım (AA) veri matrisini kendi sütunlarından oluşan bir alt-küme ile ifade etmeyi hedefleyen bir matris ayrışımıdır. Seçilen sütunların veriyi ifade edecek öznitelikleri içerdiği düşüncesine dayanır. Literatürdeki yaygın AA yöntemi önem örneklemeye dayalıdır. Bu yöntemde her sütun için bir istatistiksel önem değeri hesaplanır ve bu değerlerle orantılı olarak rasgele K adet sütun seçilir. Rassal yöntemlerdeki amaç matrisin değer kümesini en iyi ifade edecek sütunları seçerek daha iyi bir düşük-mertebeli matris yaklaştırımıdır. Seçilen sütunlar gerçek noktalar olduğu ve seyrek veride seyrekliği koruduğu için AA, Tekil Değer Ayrışımı'na iyi bir alternatif olarak görülmektedir. Fakat, veri birden çok öbek içerdiğinde en iyi düşük-mertebeli yaklaştırımı veren sütunlar betimleyici özelliği en yüksek sütunlar olmayabilir. Bu çalışmada, öbeklemeye dayalı yeni bir AA yöntemi geliştirdik. Daha yüksek betimleyicilik ve yorumlanabilirlik hedefiyle K-ortanca yöntemini kullandık. AA'yı elle-yazılmış rakam tanıma problemine uyguladık ve önerilen yöntemi literatürde en çok kabul gören yaklaşımla karşılaştırdık. Önerilen yöntemin veriyi betimlemede daha üstün olduğunu gösterdik. Verinin çok büyük bir kısmının atılması durumunda dahi başarının korunduğunu ortaya koyduk.**

Anahtar Kelimeler*—*Aradeğerlemeci Ayrışım; Öbekleme.

***Abstract*—Interpolative decomposition (ID) is a matrix factorization which aims to represent the data matrix via a subset of its own columns. These selected columns are supposed to hold the salient features expressing the data. A very common ID approach in the literature is based on importance sampling where a statistical leverage score is computed for each column and K columns are randomly selected using these scores. These randomized methods aim a better low-rank approximation of the matrix by seeking for the columns that express the range of the matrix the best. This makes ID a good alternative to Singular Value Decomposition (SVD) since it favors sparsity and the bases correspond to real data points. However, the columns leading to the best low-rank approximation are usually not the ones in terms of representativeness if the underlying data is composed of several clusters which is very common in real life. In this paper, we introduce an alternative ID approach based on clustering. We employ K-medoids to be employed as an ID method for better interpretability and respresentativeness. We apply ID on handwritten digit recognition and supply comparative results of the proposed approach to the state-of-the-art method in the literature. We show its superiority in terms of representativeness of the data. We demonstrate that most of the data can be discarded without compromising the accuracy.**

Keywords*—*Interpolative Decomposition, Clustering.

978-1-4673-5563-6/13/$31.00 ©2013 IEEE

I. GİRİŞ

Aradeğerlemeci ayrışım (AA) bir matrisi sütunlarının yalnızca bir kısmıyla ifade etmeyi hedefler [1], [2]. Genellikle bir matrisin sütunlarının yanısıra satırlarına da uygulanır ve böylece matris sütun ve satır altmatrislerine ayrıştırılır (*CUR ayrışımı*) [2], [3]. Aradeğerlemeci ayrışımdaki (ve CUR ayrışımındaki) temel motivasyon çok fazla sayıda sütun içeren büyük bir matrisin değer uzayını (*range*) az sayıda sütun kullanarak kestirmektir [2]. Günümüzde sıradanlaşmaya başlayan büyük veri miktarları ile AA'ya olan ilgi de artmıştır. Saptayıcı (*localizing*) bir ayrışım olarak da değerlendirilmen AA öznitelik seçiminde önemli bir araç olarak kullanılır ve RAM'e sığmayacak kadar büyük matrislerin işlenebilmesini mümkün kılar. Ayrıca veri içindeki gereksiz ve alakasız sütunları eleyerek hatayı azaltabilir. Benzer bir yöntem olan Tekil Değer Ayrışımı'nın (TDA) aksine, seçilen baz vektörler gerçek vektörlerin doğrusal bileşimine değil doğrudan kendilerine denk gelmektedir. Bu yüzden AA'nın verdiği bazların veriyi betimleyici özellikleri yüksektir [3]. Ek olarak, verinin seyrek olması durumunda TDA seyrek çarpanlar vermeyebilir fakat AA matrisin kendi sütunlarını kullandığı için seyrekliği garantiler. TDA ve Temel Bileşenler Analizi gibi AA da veri sıkıştırma, öznitelik çıkarımı ve veri analizi gibi birçok alanda temel bir araç olarak kullanılmaktadır [4]–[6].

Literatürde önerilen AA yöntemlerinin betimlemede de başarılı oldukları iddia edilmektedir. Fakat bu, yeterli deneyle sınanmamış eksik bir iddiadır ve verinin tek öbekli olduğunu varsayar. Oysa ki çok öbekli bir veride bu yöntemlerin betimleme başarısı oldukça düşüktür. Bu çalışmada AA'nın betimleyici niteliği üstüne odaklanılmakta ve AA'nın öbekleme problemi ile yakından ilgili olduğu gösterilmektedir. AA'ya öbekleme açısından baktığımızda seçilen sütun sayısı boyut sayısından fazla olabilir; örneğin 2 boyutlu bir veride 3 öbek merkezi seçebiliriz. Bu durumda AA düşük-mertebeli yaklaştırım aracı olarak kullanılmaz, fakat betimleme niteliğini güçlü bir şekilde korur. Ayrıca *aradeğerlemeci* özelliği de korunmaktadır. Yöntemlerin sade ve hızlı olması tercih edildiğinden önerdiğimiz AA yönteminde K-ortanca yöntemi kullanılmaktadır [7].

Öte yandan literatürdeki çalışmaların neredeyse tümünde AA düşük-mertebeli matris yaklaştırımı amacıyla kullanılmaktadır. Önem Örnekleme'ye (ÖÖ) dayalı rassal yöntemler hızlı çalışmaları sebebiyle özellikle tercih edilmektedir. ÖÖ'ye-dayalı yöntemlerde her sütun için bir önem değeri hesaplanır ve bu değerlerle orantılı olarak rasgele K adet sütun seçilir. Bir sütunun önem değeri olarak onun Öklid uzaklığı [8],

[9], seyreklik değeri [6] veya sağ tekil vektörlerinin normu tercih edilmektedir [4]. Önişlem olarak TDA hesaplamanın gerektiği durumda Mahoney *v.d.*'nin yöntemi büyük veriler için uygulanabilirliğini yitirebilmektedir çünkü TDA masraflı bir işlemdir. Bu sorunu aşabilmek için Arı *v.d.* [5] AA kullanımını büyük verilere genişletebilmek için Rassal-TDA [10] kullanımını önermektedir. Liberty *v.d.* de düşük-mertebeli matris yaklaştırımı için rassal yöntemler geliştirmişlerdir [1]. Martinsson *v.d.* AA çözümü için Fortran paketi sunmaktadır [11]. Yakın zamandaki çalışmalarıyla Wang ve Zhang ise mevcut rassal yöntemlerden daha başarılı göreceli hataya sahip bir yöntem ortaya koymuşlardır [12]. Literatürde içbükey eniyileme veya QR ayrışımı tabanlı yöntemler de mevcuttur, fakat hem başarılarının görece düşük oluşu hem de sadeliği korumak amacıyla bu çalışmanın kapsamı dışında tutulmuştur.

Önerdiğimiz yöntem ile literatürde kabul gören en yaygın yöntem, elle yazılmış rakam tanıma problemine uygulanarak karşılaştırılmakta ve başarısı ortaya konulmaktadır. Literatürde yaygın kabul gören düşüncenin de yanıltıcı olduğu, önem örneklemenin tamamen rasgele olan seçilime bir üstünlük sağlamadığı gösterilmiştir. Bu çalışmada AA'nın yeni bir bakış açısıyla ele alınması sağlanmış, geniş öbekleme literatürünün bu yönde kullanılması için ilk adımlar atılmıştır.

II. YÖNTEM

Aradeğerlemeci Ayrışım'daki (AA) amaç M boyutlarına sahip N adet sütun vektörü içeren $\mathbf{X} \in \mathbb{R}^{M\times N}$ matrisini bu sütunlardan K tanesinin doğrusal bileşimi biçiminde ifade etmektir. Başka bir deyişle, N vektör içinden K tanesini seçerek diğerlerini bu seçilenlerin doğrusal bileşimi biçiminde yazmaktır. $K < \mathrm{mertebe}(\mathbf{X})$ durumunda kesin eşitlik sağlanmaz, seçilen sütunlar ile diğerlerinin ancak yaklaştırımı (*approximation*) yapılabilmektedir. Seçilen sütunların indislerinin kümesi J olsun. Bu durumda,

$$\mathbf{X} \approx \mathbf{C}\mathbf{Z} = \mathbf{X}_{\cdot J}\mathbf{Z} \tag{1}$$

elde edilir. Yatay nokta tüm satır indislerini ifade etmektedir. $\mathbf{C} \in \mathbb{R}^{M\times K}$ seçilen sütunlardan oluşan yarı-matrisi, $\mathbf{Z} \in \mathbb{R}^{K\times N}$ ise aradeğerleme katsayılarını içermektedir. $\mathbf{Z}$ matrisinin J indislerine denk gelen sütunları birim matrisini oluşturduğu için AA *aradeğerlemeci* özelliğe sahiptir. Daha net bir ifadeyle, $\mathbf{Z}_{\cdot J} = \mathbf{\Pi}$, öyle ki $\mathbf{\Pi} \in \mathbb{R}^{K\times K}$ bir permütasyon matrisidir.

AA iki altproblemden oluşmaktadır: *1) Hangi sütunlar seçilmelidir? 2) Aradeğerleme katsayıları nasıl hesaplanmalıdır?* $\mathbf{Z}$ aradeğerleme katsayıları altta verilen optimizasyon ile elde edilir:

$$\mathbf{Z} = \underset{\mathbf{Z}'\in\mathbb{R}^{K\times N}}{\arg\min}\ \mathcal{D}\left[\mathbf{X} \| \mathbf{X}_{\cdot J}\mathbf{Z}'\right] \tag{2}$$

öyle ki $\mathcal{D}[\cdot\|\cdot]$ probleme uygun olarak seçilmiş bir masraf fonksiyonudur.

J'nin seçilmesindeki strateji yöntemden yönteme oldukça değişmekte fakat aradeğerleme katsayıları, bu çalışmada da olduğu gibi, en küçük kareler minimizasyonu ile hesaplanmaktadır.

Bu makalede çok-öbekli veride daha yüksek betimleme gücü sağlayan yeni bir bakış açısı sunulmuştur. Önerilen sütun seçme stratejisi K-ortanca yöntemine dayalıdır. Bütünlüğü sağlamak ve önerilen yöntemin farkını ortaya koymak adına öncelikle literatürde kabul gören rassal yöntemleri anlatmayı, ardından önerilen yöntemi sunmayı tercih ettik.

A. Rassal AA Yöntemleri

Son yıllarda AA'yı büyük verilere uygulama amacıyla önemli rassallaştırılmış algoritmalar geliştirilmiştir. Bu yöntemler iki temel aşamadan oluşur. Her sütun için o sütunun veriyi betimlemededeki önemini gösteren π_n değeri hesaplanır ($n = 1, \ldots, N$). Ardından bu değerlerden oluşan çokterimli bir dağılımdan K adet rasgele indis seçilir. Bu yaklaşımlar Önem Örnekleme'ye (ÖÖ) dayalıdır.

En temel ÖÖ yaklaşımı π_n değerini $n.$ sütunun l_2-normuna orantılı olarak hesaplar [8], [9]. Lee ve Choi ise π_n değerini hesaplamak için $\xi(n) = \left(\sqrt{n} - \|\mathbf{X}_{\cdot n}\|_1/\|\mathbf{X}_{\cdot n}\|_2\right)/\sqrt{n} - 1$ seyreklik fonksiyonunu kullanmaktadır. $\mathbf{X}_{\cdot n}$ eş dağılımlı elemanlara sahip olduğunda $\xi(n) = 0$ olur; sadece bir tek sıfır-olmayan elemanı olduğu durumda ise 1 olur. Mahoney ve Drineas [3] π_n değerlerini hesaplamak için $\mathbf{X}$'in kısmi Tekil Değer Ayrışımı'na (TDA) dayalı alternatif bir yöntem geliştirmiştir. Kısmî-TDA şöyle hesaplanır:

$$\mathbf{X} \approx \mathbf{A}_r\mathbf{\Sigma}_r\mathbf{B}_r^\intercal \tag{3}$$

$\mathbf{A}_r$, $\mathbf{B}_r$, ve $\mathbf{\Sigma}_r$ sırasıyla sol ve sağ ortonormal tekil matrisler ve r adet tekil değeri köşegeninde büyükten küçüğe doğru sıralanmış biçimde içeren köşegen matristir. Kısmî-TDA'nın hesabından sonra $n.$ sütunun seçilme olasılığı şöyle hesaplanır:

$$\pi_n = \frac{1}{r}\sum_{i=1}^{r} b_{ni}^2, \quad n = 1, \ldots, N \tag{4}$$

Burada b_{ni} ile $\mathbf{B}_r^\intercal$ matrisinin (n, i). elemanı ifade edilmektedir. Bu algoritma Yöntem 1'de verilmiştir. Yeterli miktarda sütun seçildiğinde beklenen göreceli hatanın çok düşük olacağı bilinmektedir [4].

Yöntem 1 Önem Örneklemeye dayalı AA

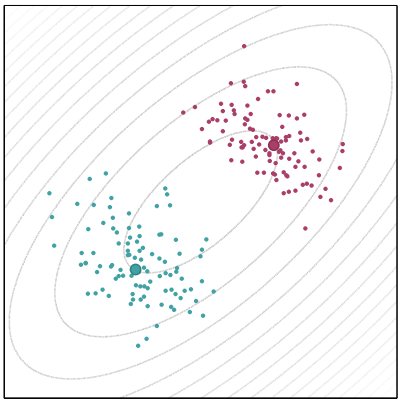
Girdi: $\mathbf{X} \in \mathbb{R}^{M\times N}$: veri matrisi
Girdi: K: baz sayısı
Girdi: r: Kısmî-TDA'da kullanılacak tekil değer sayısı
Sağla: $\mathbf{Z}$ farkın Frobenius normunu $\|\mathbf{X} - \mathbf{C}\mathbf{Z}\|_F$ enküçükler

1: $\mathbf{A}_r\mathbf{\Sigma}_r\mathbf{B}_r^\intercal \Leftarrow$, $\mathbf{X}$'in kısmi-TDA'sı
2: **her** $n = 1 \to N$ **için:**
3: $\quad \pi_n \Leftarrow n.$ sütunun seçilme olasılığı (4).
4: $J \Leftarrow \{\pi_n\}_{n=1}^N$ çokterimlisinden rasgele seçilmiş K indis
5: $\mathbf{C} \Leftarrow \mathbf{X}_{\cdot J}$
6: $\mathbf{Z} \Leftarrow \mathbf{C}^\dagger\mathbf{X}$, öyle ki † Moore-Penrose tersi
7: **döndür** $\mathbf{C}, \mathbf{Z}$

TDA hesaplamanın başlı başına masraflı bir işlem olduğu bilimektedir. Tam çözümü $O(\min\{MN^2, M^2N\})$ mertebesindedir [13]. Kısmî çözümü de geleneksel yöntemler kullanıldığında masraflıdır. Bu problemi çözebilmek için Arı *v.d.* [5], Halko *v.d.*'nin [10] geliştirdiği rassallaştırılmış TDA yöntemini ÖÖ'de kısmi-TDA hesaplarken kullanmayı önermişlerdir. Bu yöntem $\mathbf{X}$'in değer uzayından rasgele örnek noktalar üretir ve örneklenmiş bu alt-uzayın dikleştirilmesine dayanır. r mertebesinde $M \times N$'lik bir matrisin kısmi-TDA'sı $O\left((M+N)\,r\right)$ zamanda hesaplanır. AA için $r \approx K$ olarak

seçebiliriz. Dolayısıyla karmaşıklık $O\left(\left(M+N\right)K\right)$ zamandadır. Bu yöntemin artısı gerçek veri matrisinin üstünden birkaç kez geçmesidir. Hafıza karmaşıklığı ise $\mathbf{B}_K$ matrisinin eleman sayısına eşittir, yani $O(NK)$'dir.

Denklem (4)'de verilen olasılık değerlerinin geometrik yorumlanması için Şekil 1'e bakılabilir. 2 boyutlu 200 adet nokta iki öbek halinde oluşturulmuş ve (4) ile verilen aynı olasılık değerine sahip nokta konumlarını göstermek için eliptik halkalar kullanılmıştır. Örneğin en içteki gri halka üstündeki tüm noktaların seçilme olasılığı aynıdır. Halkalar dışa doğru büyüdükçe seçilme olasılığı artar. Daha yüksek boyutlu durumda eliptik halkalar yerini hiper-elipsoidlere bırakacaktır. Verilen görsel örnek bu çalışmanın odağını göstermek açısından oldukça uygundur. Şekildeki noktalar farklı renklerle gösterilen iki öbekten oluşmaktadır. Öbek merkezleri büyük yuvarlak noktalar ile gösterilmiştir. ÖÖ kullanıldığında bu iki nokta ile aynı halkada bulunan birçok noktanın da seçilme olasılıkları aynıdır. Hatta dış halkadaki noktaların seçilme ihtimali daha yüksektir. Fakat aslında ortamda iki adet öbek vardır ve bu öbeklerin merkezlerini seçmek verinin iyi ifade edilmesi açısından daha doğru bir tercih olacaktır. Literatürdeki ÖÖ yöntemleri bu durumu kapsamaz.



Şekil 1. İki adet öbekten oluşan 2 boyutlu noktalar. Büyük yuvarlaklar öbek merkezlerini göstermektedir. Halkalar ise ÖÖ'de eşit olasılığa sahip konumları ifade eder. Dış halkaların üstündeki noktaların seçilme olasılığı içerdekilerden yüksektir. Görüldüğü üzere ÖÖ çok öbekli durumu kapsamaz.

B. Çoköbekli Veri için AA

Bu çalışmada bu temel örnekten yola çıkarak AA'ya yeni bir bakış açısıyla yaklaşılmaktadır. Amacımız veriyi en iyi ifade eden örnek noktaları bulmaktır. Aslında bu $\mathbf{X}$'in sütunlarını K kümeye ayırmayı hedefleyen bir öbekleme yaklaşımıdır. Bu probleme en temel yaklaşım K-ortanca (K-*medoid*) yaklaşımıdır. Ortanca nokta diğer noktalara olan ortalama uzaklığı en küçük olan veri noktasıdır [7]. K-ortalama ile karşılaştırıldığında gürültü ve aykırı değerlere karşı daha gürbüzdür. Belli bir uzaklık fonksiyonuna bağımlı değildir, hatta uzaklıkların simetrik olması da gerekmez.

Yöntem 2'de verilen AA yaklaşımı ortanca noktaların ilklendirilmesi ile başlar Ardından her adımda her nokta bir öbeğe atanır ve bu öbeklerin yeniden ortanca noktaları bulunur. İki nokta arasındaki uzaklık l_2 uzaklığı olarak seçildiğinde Z matrisi $\|\mathbf{X}-\mathbf{CZ}\|_2$ değerini en küçükleyen matris olarak hesaplanır. Yöntem global en iyiyi garanti etmediğinden, yerel en iyilerde takılmamak amacıyla farklı ilklendirmeler ile çok kez çalıştırılıp aralarından en iyisi seçilir. K-ortancanın Beklenti-Enbüyütme yöntemi ile eşyönlü (isotropik) olmayan kovaryans matrisleri için de genişletilebilmesi mümkündür. Fakat bu değişiklik algoritmayı karmaşıklaştıracaktır; çalışmanın odağı büyük veri işleme olduğu için yaklaşım hızlı ve sade tutulmuştur. Bu haliyle karmaşıklığı belirleyen veri uzaklık matrisinin boyutlarıdır, dolayısıyla zaman ve yer karmaşıklığı $O(N^2)$ mertebesindedir.

Yöntem 2 K-ortanca ile AA

Girdi: $\mathbf{X} \in \mathbb{R}^{M\times N}$: veri matrisi
Girdi: K: ortanca sayısı
Sağla: $\mathbf{Z}$ farkın Frobenius normunu $\|\mathbf{X}-\mathbf{CZ}\|_F$ enküçükler
Sağla: $\mathbf{Z} \in \{0,1\}^{K\times N}$, $\sum_k Z_{kn} = 1\ \forall n \in \{1,\ldots,N\}$

$\mathbf{D} \Leftarrow N \times N$ uzaklık matrisi; $D_{ij} = \|\mathbf{X}_{\cdot i} - \mathbf{X}_{\cdot j}\|_2$
$J \Leftarrow$ Rasgele K adet sütunu ilk ortancalar olarak belirle
her $i = 1 \rightarrow$ *maksDöngüSayısı* **için:**
  her $n = 1 \rightarrow N$ **için:**
    $c_n \Leftarrow \arg\min_{k|k\in\{1,\ldots K\}} D_{nJ_k}$: Öbek merkezini ata
  her $k = 1 \rightarrow K$ **için:**
    $J_k \Leftarrow \arg\min_{n|c_n=k} \sum_j^N D_{nj}$: Ortancayı yeniden hesapla
  yakınsadıysa döngüden çık
$\mathbf{C} \Leftarrow \mathbf{X}_{\cdot J}$
$Z_{kn} \Leftarrow n.$ nokta $k.$ ortancaya en yakınsa 1, değilse 0.
döndür $\mathbf{C}, \mathbf{Z}$

Bu çalışmada AA sütun seçme aracı olarak kullanılmaktadır. Fakat kolaylıkla CUR ayrışımını hesaplamak için genişletilebilir. $\mathbf{X} \approx \mathbf{CUR}$ ayrışımını elde etmek için öncelikle $\mathbf{X}$ üstünde AA uygulanarak seçili sütunlardan oluşan $\mathbf{C} = \mathbf{X}_{\cdot J_c}$ yarı-matrisi bulunur. Benzer biçimde, $\mathbf{X}^\intercal$ devrik matrisine AA uygulanarak $\mathbf{R} = \mathbf{X}_{J_r\cdot}$ satır matrisi elde edilir. Ardından basit bir en küçük kareler minmizasyonu çözülerek $\mathbf{U} = \mathbf{X}^\dagger_{J_r J_c}$ ile $\mathbf{U}$ bağlantı matrisi hesaplanır [2]. Burada $\dagger$ Moore-Penrose tersi (*pseudo-inverse*) işlemini belirtmektedir.

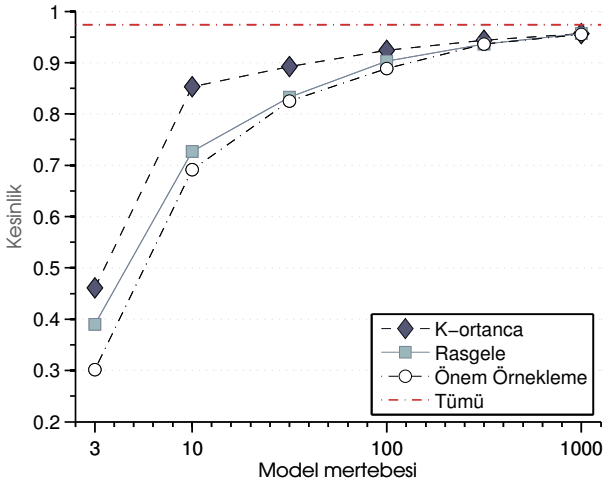
III. DENEYLER VE SONUÇLAR

Önerilen yöntemin sınanması için MNIST elle-yazılmış rakam veritabanı kullanılmıştır [14]. MNIST 20×20 boyutlarında 10 farklı rakama ait toplam 50000 eğitim ve 10000 test örneğinden oluşmaktadır. Karşılaştırmada üst sınır oluşturması için öncelikle tüm eğitim kümesini içererek En Yakın Komşu (EYK) yöntemini kullandık. Bunun için tüm eğitim kümesini Temel Bileşenler Analizi ile 50 boyuta düşürdük ve her test örneğini indirgenmiş bu uzayda en yakın olduğu eğitim örneğinin sınıfına atadık. Bu yöntem ile $\%97.42$'lik bir kesinlik elde ettik. Kesinlik, doğru sınıflandırılan rakamların oranını göstermektedir.

Ardından her bir sınıf için üssel $(10^i, i = 0.5, 1, \ldots, 3)$ bir artışla gidecek şekilde sırasıyla $3, 10, 32, 100, 316$ ve 1000 adet sütunu Yöntem 1'deki gibi Mahoney ve Drineas'ın [4] Önem Örnekleme'ye dayalı algoritması ile seçtik. r değerini 50 olarak aldık. Eğitim kümesinde seçilmeyen diğer sütunları attık ve EYK yöntemini böyle tekrarladık. Sonuçlar Şekil 2'de görülmektedir. Bu yöntemin farkını görmek için ek olarak aynı sayıda sütunu tamamen rasgele seçtik ve benzer şekilde diğerlerini atarak kalanlara EYK uyguladık. ÖÖ yönteminin

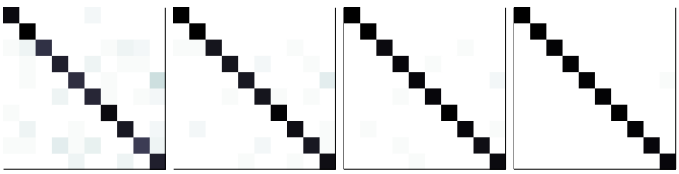
beklentinin aksine tamemen rasgele seçmekten daha iyi sonuç vermediği açıkça görülmektedir.

Alternatif olarak Yöntem 2'de önerilen K-ortanca tabanlı AA ile aynı sayıda sütunu seçtik ve bu sütunları eğitim kümesi olarak belirleyip EYK ile sınıflama yaptık. Elde edilen sonuçlar Şekil 2'de diğer yöntemlere ek olarak görülmektedir. Seçilen sütun sayıları için elde edilen kesinlik değerleri bu yöntem için sırasıyla $\%46.13, \%85.32, \%89.26, \%92.40, \%94.36$ ve $\%95.68$'dir. Yalnızca 10'ar adet örnek seçerek, yani verinin $\%99.8$'ini atarak $\%85.32$ gibi yüksek bir değer elde etmek mümkündür. Verinin $\%80$'i atıldığında ise başarıdaki kayıp $\%2$'nin altındadır.



Şekil 2. Karşılaştırma sonuçları. Kesinlik, doğru sınıflandırılan rakamların oranını gösterir. Tüm veri kullanıldığı durumda elde edilen kesinlik değeri kırmızı çizgi ile üst sınır olarak verilmiştir. K-ortanca'nın en iyi sonucu verdiği, yaygın olarak kullanılan ÖÖ tabanlı yöntemin ise tamamen rasgele seçime göre daha kötü olduğu açıkça görülmektedir.

$K = 10, 100$ ve 1000 için Yöntem2 ile elde edilen hata matrisleri Şekil 3a–c'de görülmektedir. Şekil 3d'de ise tüm eğitim kümesi kullanılınca elde edilen hata matrisi verilmiştir. Görüldüğü üzere az sayıda sütun seçildiğinde hatalar 4'ün 9 ile 3, 5 ve 8'in de birbirleri ile karıştırılmasından kaynaklanmaktadır. Yer azlığı nedeniyle başarıları düşük olan diğer yöntemler hariç tutulup yalnızca K-ortanca yönteminin bazı sonuçları verilmiştir.



(a) 10-ortanca (b) 100-ortanca (c) 1000-ortanca (d) Tümü

Şekil 3. Yöntem 2 ile sırasıyla 10, 100 ve 1000 sütun seçildiğinde elde edilen hata matrisleri (a–c). Tüm veri kullanıldığında elde edilen hata matrisi (d). Hata matrisindeki i, j elemanının koyuluğu i rakamının j rakamı olarak sınıflandırılma yüzdesini göstermektedir. Sol üst köşe $0, 0$ konumudur.

IV. VARGILAR

Bu çalışmada Aradeğerlemeci Ayrışım için kullanılan yaygın yöntemler irdelenmiş ve düşük-mertebe hedefinin veriyi betimlemede de başarılı olacağı varsayımının yanlış olduğu gösterilmiştir. Alternatif olarak K-ortanca tabanlı bir yöntem önerilmiş ve elle-yazılmış rakam tanıma problemi üstünde başarısı ortaya konulmuştur.

Not edilmelidir ki aynı veritabanında farklı yöntemlerle daha yüksek başarılar elde edilmiştir. Fakat bu çalışmanın odağı elle-yazılmış rakam tanıma problemi için bütünsel bir yöntem geliştirmek değil, Aradeğerlemeci Ayrışım'a alternatif bakış açısı geliştirmek ve önerilen yöntemin üstünlüğünü bu problem üstünde deneysel olarak göstermektir.

Büyük veri ile Aradeğerlemeci Ayrışım gibi temel yöntemlere olan ilgi artmaktadır ve veriyi daha iyi ifade etmeye yarayan sütun seçme mekanizmaları önem kazanmaktadır. Bu çalışma ile konuya yeni bir bakış açısı getirmek hedeflenmiş ve öbekleme ile Aradeğerlemeci Ayrışım'ın yakın ilişkisi ortaya konmuştur.

TEŞEKKÜR

A. T. Cemgil 110E292 nolu "Bayesian matrix and tensor factorisations (BAYTEN)" isimli araştırma projesi kapsamında TÜBİTAK tarafından ve BAP 6882 projesi kapsamında Boğaziçi Ü. tarafından desteklenmektedir.

KAYNAKÇA

[1] E. Liberty, F. Woolfe, P. G. Martinsson, V. Rokhlin, and M. Tygert, "Randomized algorithms for the low-rank approximation of matrices." *Proc. of the National Acad. of Sci.*, vol. 104, pp. 20 167–72, 2007.

[2] N. Halko, P. G. Martinsson, and J. A. Tropp, "Finding Structure with Randomness: Probabilistic Algorithms for Constructing Approximate Matrix Decompositions," *SIAM Review*, 2011.

[3] M. W. Mahoney, "Randomized Algorithms for Matrices and Data," *Foundations and Trends in Machine Learning*, pp. 123–234, 2011.

[4] M. W. Mahoney and P. Drineas, "CUR Matrix Decompositions for Improved Data Analysis," *Proc. of the National Acad. of Sci.*, vol. 106, no. 3, pp. 697–702, 2009.

[5] I. Arı, U. Şimşekli, A. T. Cemgil, and L. Akarun, "Large Scale Polyphonic Music Transcription Using Randomized Matrix Decompositions," in *EUSIPCO*, 2012.

[6] H. Lee and S. Choi, "CUR+NMF for Learning Spectral Features from Large Data Matrix," in *IEEE Int'l Joint Conf. on Neural Networks*, 2008, pp. 1592–1597.

[7] H. S. Park and C. H. Jun, "A simple and fast algorithm for K-medoids clustering," *Expert Systems with Applications*, vol. 36, no. 2, pp. 3336–3341, Mar. 2009.

[8] P. Drineas, R. Kannan, and M. W. Mahoney, "Fast Monte Carlo Algorithms for Matrices III: Computing a Compressed Approximate Matrix Decomposition," *SIAM Journal on Computing*, vol. 36, no. 1, pp. 184–206, 2007.

[9] A. Frieze, R. Kannan, and S. Vempala, "Fast Monte-Carlo Algorithms for Finding Low-rank Approximations," *Journal of the ACM*, pp. 1025–1041, 2004.

[10] N. Halko, P. G. Martinsson, Y. Shkolnisky, and M. Tygert, "An Algorithm for the Principal Component Analysis of Large Data Sets," *SIAM Journal on Scientific Computing*, vol. 33, no. 5, p. 2580, 2011.

[11] P. G. Martinsson, V. Rokhlin, Y. Shkolnisky, and M. Tygert, "ID: A software package for low-rank approximation of matrices via interpolative decompositions, Version 0.2," 2008. [Online]. Available: http://cims.nyu.edu/ tygert/software.html

[12] S. Wang and Z. Zhang, "A Scalable CUR Matrix Decomposition Algorithm: Lower Time Complexity and Tighter Bound," in *NIPS*, 2012.

[13] G. H. Golub and C. F. Van Loan, *Matrix Computations*, 3rd ed. The Johns Hopkins University Press, 1996.

[14] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based Learning Applied to Document Recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.