Rasgele Örnekleme ile Düşük Rank Matris Tamamlama
Low Rank Matrix Completion via Random Sampling

Hakan Güldaş, Ali Taylan Cemgil
Bilgisayar Mühendisliği Bölümü
Boğaziçi Üniversitesi
İstanbul, Türkiye
Email: {hakan.guldas,taylan.cemgil}@boun.edu.tr

***Özetçe* —Bu çalışmada, eksik ya da kayıp veri olduğu durumda düşük rank matris tamamlamak için bir yöntem sunuyoruz. Bu problemi çözmek için en iyileme yöntemleri, EM (Beklenti-enbüyütme) tabanlı yöntemler ya da Değişimsel Bayesçi (Variational Bayesian) yöntemler önerilmiştir. Fakat, bu yöntemler sözkonusu veri matrisi devasa boyularda olduğunda ve özellikle hızlı belleğe (RAM) sığmadığı durumlarda, hesaplama yükü açısından pratikte kullanılabilir olmaktan çıkmaktadırlar. Geleneksel yöntemler, yinelemeli olduklarından veri matrisine defalarca erişmeleri gerekmektedir ve ikincil bellekten veri aktarımı bu durumda darboğaz oluşturmaktadır. Bu çalışmada, sözünü ettiğimiz darboğazı aşmak adına, rasgele sütun ve satır örnekleme üzerine kurulu bir yöntem önerilmektedir. Bu yöntem, bellek erişimi açısından daha önceki yöntemlerden daha verimli olup aynı oranda isabetli sonuçlar vermektedir ve yüksek oranda paralelleştirilebilme özelliğine sahiptir.**

***Abstract*—In this work, we present a method to find a low rank approximation to a large matrix with missing entries. Optimization methods, EM based methods or Variational Bayesian methods are proposed to solve this problem. However, the computational cost of these methods can be prohibitively large in case of a massive data matrix, when the known entries do not even fit into the fast random access memory (RAM). Traditional methods access the data matrix several times during iterations and the data transfer from a secondary storage becomes the key bottleneck. To alleviate this problem, we devise a randomized scheme by sampling a subset of rows or columns of the original matrix. The resulting algorithm is efficient in terms of the number of required disk accesses, is highly parallelizable and produces factorizations that are competitively accurate.**

I. GİRİŞ

Düşük rank matris yaklaşımları veriler içerisindeki doğrusal yapıları ortaya çıkarmaları açısından veri işleme ve bilimsel hesaplam alanlarında oldukça yaygın bir şekilde kullanılmaktadır. En yaygın bilinen düşük rank matris yaklaşım yöntemlerinden birisi temel bileşenler analizidir (PCA). Genel olarak doğrusal yapı gösteren verilerde boyut azaltma problemi düşük ranklı matris yaklaşımı bulma problemiyle eşdeğerdir. Bunun dışında görüntü, ses işleme, yapay öğrenme ve istatistik alanlarındaki birçok problem de düşük rank matris yaklaşımı bulma problemleri olarak ifade edilebilir.

Düşük rank matris yaklaşımı hesaplamak için en yaygın kullanılan yöntemlerden biri, tekil değer ayrışımıdır (SVD). Tekil değer ayrışımı hesaplama için yinelemeli yöntemler kullanılmaktadır [6]. Fakat bu yöntemler, devasa veri matrisleri bağlamında hesaplama yükü ve veriye erişim açısından pratikte uygulanabilir olmaktan uzak kalmaktadır. Bu engeli aşmak amacıyla, son yıllarda rasgeleleştirilmiş algoritmalar önem kazanmaktadır [7], [9], [2].

Uygulamalarda, veri kayıpları ya da eksik gözlemlerin varlığı, sıklıkla karşılaşılan bir durumdur. Eksik verinin varlığında düşük rank matris yaklaşımı yapılabilmesi için eksik verinin önce kestirilmesi gerekmektedir. Kayıp verilere belli bir sabit değer atanması (örneğin 0) ya da sütun ortalamalarının atanması gibi durumlar kötü sonuçlar vermektedir. Gözlemlenmiş veri kullanılarak kayıp verinin kestirilmeye çalışılması problemi, matrisler bağlamında, matris tamamlama problemidir [3].

Yapay öğrenme alanında birçok problem aslında matris tamamlama problemi olarak formülize edilebilir. Matris tamamlama problemi için popüler bir örnek olarak Netflix yarışması gösterilebilir. Kabaca anlatmak gerekirse, Netflix veritabanında bulunan kullanıcıların filmlere verdikleri puanları, boyutları kullanıcı sayısına film sayısı olan bir matrisle ifade edebiliriz. Doğal olarak her kullanıcı her filmi puanlamamıştır. Netflix yarışmasındaki amaç, var olan puanlamaları göz önünde bulundurarak hangi kullanıcının hangi filme kaç puan vereceğini en iyi şekilde tahmin eden bir şema oluşturmaktır. Bu problem, kullanıcıların filmleri oylarken az sayıda etkeni göz önünde bulundurduğunu düşünerek, düşük rank matris tamamlama problemi olarak görülebilir. Daha genel olarak katılımlı süzgeçleme (collaborative filtering) problemlerinde matris tamamlama yaklaşımının işe yaradığı görülmüştür [12], [4]. Bunun dışında görüntü işleme alanında da düşük rank matris tamamlama problemi önemli bir rol oynamaktadır([10],[1]). Bu çalışmada, daha önce düşük rank matris yaklaşımı probleminde kullanılan rasgeleleştirilmiş algoritmaların, düşük rank matris tamamlama problemine nasıl uygulanabileceğini inceliyoruz ve bu yönde geliştirdiğimiz rasgeleleştirilmiş algoritmayı sunuyoruz.

Bu çalışma Türkiye Bilimsel ve Teknik Araştırmalar Kurumu (TÜBITAK) tarafından 110E292 no'lu araştırma projesi ve Boğaziçi Araştırma Projeleri (BAP) tarafından P6882 no'lu araştırma projesi kapsamında desteklenmektedir.

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

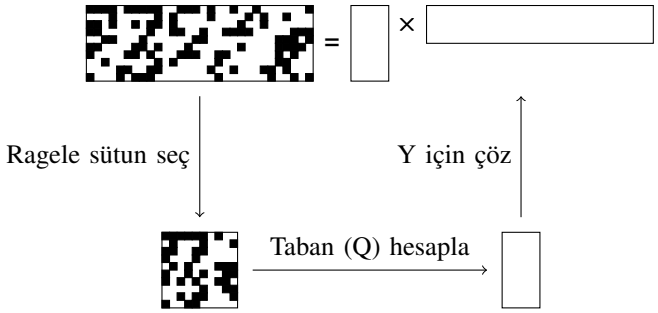
II. MATRİS TAMAMLAMA PROBLEMİ

A. PROBLEM TANIMI VE İLGİLİ ÇALIŞMALAR

X matrisi $m \times n$ boyutlarında kısmen gözlenmiş bir matris olsun, genelliği kaybetmeden $m \leq n$ olduğunu varsayabiliriz, M matrisi de aynı boyutlarda bir matris olsun, öyle ki eğer X matrisinin (i, j) konumundaki girdisini biliyorsak $M_{ij} = 1$, bilmiyorsak $M_{ij} = 0$ olsun. Bu durumda r ranklı matris tamamlama probleminde, $M \odot X = M \odot X_r$ eşitliğini sağlayan en çok r ranklı bir X_r matrisi bulmak istiyoruz ($\odot$ imgelemi Hadamard çarpımını temsil etmektedir, yani $A = B \odot C$ ise $A_{ij} = B_{ij}C_{ij}$). Bu problem aynı zamanda $\|M \odot (X - QY)\|$ değerini enküçükleyen sırasıyla $m \times r$ ve $r \times n$ büyüklüklerinde Q ve Y matrisleri bulma problemi olarak genişletilebilir (burada $\|\cdot\|$ Frobenius normu göstermektedir, yani $\|A\| = \sqrt{\sum_{i,j} A_{i,j}^2}$). Bu ifade şeklinin diğer gösterimden üstünlüğü verinin düşük boyutlu uzaydaki suretinin Q ve Y matrisleriyle açığa çıkmasıdır, böylece bellekte daha çok yer kaplayan X matrisi yerine, çok daha az yer kaplayan Q ve Y matrisleri tutularak da veri üzerinde işlem yapılabilir. Kolayca görülebileceği üzere bu şekilde bulabileceğimiz Q ve Y matrisleri tek değildir, Q matrisini sağdan, Y matrisini ise soldan $r \times r$ büyüklüğünde tersinir bir matrisle çarparak bulduğumuz çözüm de aynı kısıtı sağlamaktadır. Çözüm uzayını küçültmek için Q matrisi üzerinde ortogonal olma kısıtlamasını, yani $Q^tQ = I_r$, koyuyoruz (burada I_r, $r \times r$ büyüklüğünde birim matrisi göstermektedir).

Düşük rank matris tamamlama problemi için önerilen çözümler eniyileme yöntemleri ([3],[11],[14]), EM (Expectation-Maximization)([8]) tabanlı yöntemler ve Değişimsel Bayesçi (Variational Bayesian) ([13]) yöntemleri içermektedir. Candès ve Recht'in çalışması ([3]) çekirdek norm enküçüklemesinin problemle ilişkilendirilmesi ve matris tamamlama probleminin teorik analizi için iyi bir örnek oluşturması açısından özellikle üzerinde durulması gereken bir çalışmadır. Bu çalışmada düşük rank matris bulma probleminin, $M \odot X = M \odot X_r$ kısıtlaması altında $\|X_r\|_*$, yani çekirdek normun enküçüklenmesi problemiyle denk olduğu gösterilmiştir, çekirdek norm matrisin tekil değerlerinin (singular value) toplamıdır. Böylece nümerik olarak çözülmesi zor olan rank eniyileme problemi, daha kolay çözülebilen dışbükey eniyileme problemine çevirilmiştir. Aynı çalışmada ayrıca belirli özellikleri sağlayan matrislerin tamamlanabilmesi için gözlenmesi gereken girdi sayısı için matrisin boyutları ve gerçek rankı cinsinden bir alt sınır da verilmiştir. Bu yöntemler matris tamamlama problemini çok düşük hatalarla ya da tam olarak çözseler de, uygulanabilir olmaları için matrisin küçük olması ya da bilinen değerlerinin matrisin seyrek bir altkümesi olması gerekmektedir. Bu açıdan devasa boyutlu yoğun matrisler söz konusu olduğunda yetersiz kalmaktadırlar.

Önerdiğimiz yönteme geçmeden önce düşük rank matris yaklaşımı problemi ve rasgeleleştirilmiş algoritmaların bu problemin çözümünde kullanımından da bahsetmek yararlı olacaktır. Verilen bir X matrisinin düşük ranklı yaklaşımı, $\|X - X_r\|$ değerini minimize eden en çok r ranklı X_r matrisidir. Frobenius norm için çözümün kesikli tekil değer ayrışımı (truncated singular value decomposition) tarafından verildiği gösterilebilir([6]). Eğer X matrisinin tekil değerleri büyükten küçüğe doğru $\sigma_1, \sigma_2, \ldots, \sigma_m$ ise en küçük hata, $\|X - X_r\|$, $\sqrt{\sum_{i=r+1}^{m} \sigma_i^2}$ değerine eşit olmaktadır. Bu durumda düşük rank yaklaşım hesaplama problemi tekil değer ayrışımı hesaplama problemine eşdeğerdir. Tekil değer ayrışımı hesaplama yöntemleri belirli bir olgunluğa erişmiştir ve uygulamada sıkça kullanılır hale gelmişlerdir, fakat devasa büyüklükte matrisler söz konusu olduğunda bu yöntemlerin hesaplama yükü kabul edilebilir mertebenin üstündedir. Bu engel rasgeleleştirilmiş algoritmalar kullanılarak aşılmaya çalışılmıştır. Bu bağlamda başlıca çalışmalar olarak [7],[9],[2] gösterilebilir. Drineas vd. [9] hesap yükünü azaltmak için rasgele satır ve sütun seçme ve düşük rank matris yaklaşımını bunları kullanarak hesaplama yöntemini önermiştir. Halko vd. [7] rasgele izdüşümler (random projections) yöntemini önermiştir. Bu yöntemde asıl matris rasgele bir matrisle çarpılarak, sütun uzayından örneklem çekilmekte ve bu örneklem için hesaplanan taban vektörleri kullanılmaktadır. Her iki yöntemin de en önemli özelliği asıl matrisin az sayıda okunmasıdır. Bu durum özellikle verinin hızlı belleğe (RAM) sığmadığı durumda avantaj sağlamaktadır. Achlioptas ve McSherry [2] ise asıl matrisin rasgele seyreltilmesi (random sparsification) yöntemini önermiştir. Böylece ayrıştırma algoritmaları gerçeklenirken seyrek yapıdan istifade ebilebilir. Dikkat edilmesi gereken nokta bu yöntemlerde verinin tamamına erişimin olduğu varsayımıdır. Eksik-kayıp veri olması durumunda bu yöntemlerin uygulanabilmesi için öncelikle kayıp değerlerin kestirilmesi gerekmektedir. Bu da bizi daha önce sözünü ettiğimiz matris tamamlama problemine yönlendirmektedir.



Şekil 1: Ana algoritmanın şeması, siyah girdiler kayıp veriyi temsil etmektedir

B. ÖNERİLEN YÖNTEM

Önerdiğimiz yöntem kabaca iki kısımdan oluşmaktadır :

- X matrisinin bilinen girdilerinin rasgele bir alt kümesini kullanarak, tahmini sütun uzayı için taban, Q matrisi, bulma;
- Bilinen değerleri kullanarak X matrisinin sütun vektörlerini bu taban vektörler cinsinden ifade etme, $M \odot X = M \odot (QY)$.

Şekil 1'de ana algoritmanın şeması kabaca tasvir edilmiştir.

Burada rasgele kısım ilk kısımdır. Farzedelim ki X matrisi için r ranklı bir tamamlama bulmak istiyoruz,X matrisinin $n \gg c \geq r$ eşitsizliklerini sağlayan bir c sayısı kadar sütununu rasgele şekilde seçelim. Daha düzgün bir şekilde ifade edecek olursak, c büyüklüğünde $I \subset \{1, 2, \ldots, n\}$ kümesini rasgele

seçelim ve $C = X_I$ olsun, burada X_I simgelemi X matrisinin I kümesiyle indislenmiş sütunları yanyana koyularak oluşturulan matrisi göstermektedir. C matrisi için daha önce bahsettiğimiz yöntemleri kullanarak düşük ranklı bir tamamlama bulabiliriz. Burada önemli olan nokta C matrisinin X matrisine göre çok küçük olması ve daha önceki yöntemler kullanılarak kolayca tamamlanabilmesidir. C matrisinin sütun uzayı için birimdik (ortonormal) bir taban, Q hesaplayalım, bunu QR ayrışımı gibi standart yöntemlerle bulabileceğimiz gibi, matris tamamlama yöntemlerinin matrisi tamamlarken bir yandan da sütun uzayı için birimdik taban hesaplamasından da faydalanabiliriz. Bulduğumuz Q matrisinin aynı zamanda X matrisinin sütun uzayı için de iyi bir yaklaşım olduğunu varsayalım.

İkinci kısımda ilk kısımda bulduğumuz Q matrisini kullanarak $M \odot X = M \odot (QY)$ eşitliğini sağlayan bir Y matrisi bulmak istiyoruz. Bu amaçla EM benzeri bir algoritma kullandık. Daha kesin bir dille ifade edecek olursak Q matrisinin bilindiğini varsayalım ve $X_{ij} \sim \mathcal{N}(\sum_{k=1}^{r} Q_{ik}Y_{kj}, \sigma^2)$ olsun. Bu durumda EM kullanarak Y parametresi için en büyük olabilirlik kestirimi (maximum likelihood estimation) yapabiliriz. Bizim problemimiz için çıkardığımız EM algoritması şu şekilde tarif edilebilir :

- $Y_{(0)}$ matrisine rasgele başlangıç değeri atayalım,
- $X_{(t)} = M \odot X + (\mathbf{1} - M) \odot (QY_{(t-1)})$,
- $Y_{(t)} = Q^t X_{(t)}$.

Burada $\mathbf{1}$ simgelemi uygun boyularda ve her girdisi 1 olan matrisi göstermektedir. Bu yaklaşımda önemli olan nokta ikinci kısımda Y matrisinin yavaş bellek erişimi açısından verimli veya paralellenebilir bir şekilde hesaplanabilmesidir. Bunun için matrislerin blok yapısından yararlanabiliriz, şöyle ki $X = [X_1, X_2, \ldots, X_k] = QY$ olsun ve X_i matrisleri $m \times n_i$ büyüklüklerinde ve $\sum_{i=1}^{k} n_i = n$ olsun. Öyleyse $Y = [Y_1, Y_2, \ldots, Y_k]$ ve her Y_i matrisi $r \times n_i$ boyutlarındaysa $X_i = QY_i$ yazabiliriz. Y matrisinin hesaplanması için her seferinde X matrisinin $m \times n_i$ kadarını hızlı belleğe okuyup, EM algoritmasını koşturabiliriz ve böylece diğer yinelemeli yöntemlerde olduğu gibi X matrisini tekrar tekrar okuma külfetinden kurtulmuş oluruz. Benzer şekilde paralel mimarilerde X matrisinin farklı blokları birbirinden bağımsız şekilde hesaplanabilir. Ana algoritma için sözde kod aşağıda verilmiştir.

Ana algoritmanın sözde kodu

Girdi: X matrisi,M gösterge matrisi,r rank,c seçilecek sütun sayısı
Çıktı: Q taban matrisi,Y koordinat matrisi
Birinci kısım :
c büyüklüğünde rasgele bir $I \subset \{1, 2, \ldots, n\}$ seç
Optimizasyon yöntemiyle $M_I \odot X_I = M_I \odot (QZ)$ eşitliğini sağlayan Q matrisi bul
İkinci kısım :
EM algoritması:
$Y_{(0)}$ matrisini rasgele başlat
for $t = 1 \ldots$ **do**
$X_{(t)} = M \odot X + (\mathbf{1} - M) \odot (QY_{(t-1)})$
$Y_{(t)} = Q^t X_{(t)}$
end for

III. DENEY VE SONUÇLAR

Yukarıda sunduğumuz algoritmayı Matlab kullanarak gerçekledik. Q matrisinin hesaplanması için Genişletilmiş Lagrange Çarpanları (Augmented Lagrange Multiplier) yöntemiyle $L1$ norm enküçüklemesi ([14]) yöntemini kullandık. Bu seçimimiz algoritmanın basitçe gerçeklenebilmesi ve denediğimiz diğer yöntemlerden daha hızlı olmasından kaynaklanmaktadır. Deneylerin hepsi Windows 7 - 32bit işletim sistemi altında, Intel Core2Duo T8100 çift çekirdekli 2.1 Ghz işlemciye ve 3 GB RAM büyüklüğüne sahip dizüstü bilgisayarda yapılmıştır.

Birinci deneyde gerçeklediğimiz algoritmayı 500×2000 büyüklüğünde, 10 ranklı yapay veri matrisinde, girdilerin yarısını maskeleyerek test ettik. $c = 10, 20, \ldots, 100$ değerleri için, yani farklı örneklem büyüklükleri için, her değer için 100 kez olacak şekilde algoritmayı test matrisi üzerinde koşturduk ve her c değeri için koşma zamanı ve matris üzerinde karelenmiş görece hatayı ölçtük, görece hata $\|M \odot (X - QY)\|/\|M \odot X\|$ büyüklüğüdür. Şekil 2a'da ortalama görece hatanın örneklem büyüklüklerine göre değişimi logaritmik ölçekte, şekil 2b'de ise ortalama koşma zamanının örneklem büyüklüklerine göre değişimi verilmiştir.

	Koşma zamanı	Görece hata
Rasgele Örnekleme	0.6679 sn.	3.1964e-05
GLÇ	4.9274 sn.	1.9370e-24
TKE	20.2760 sn.	4.1096e-06

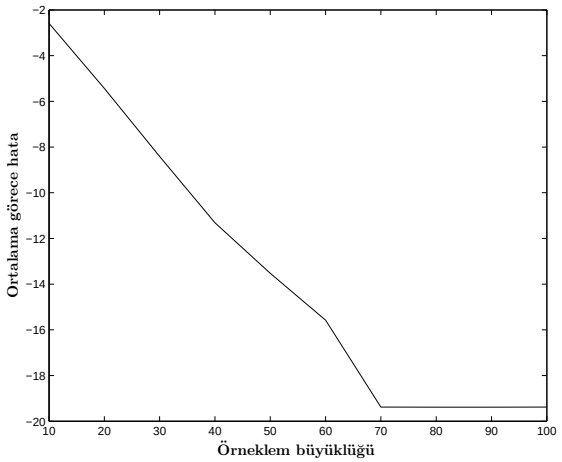
Tablo I: Rasgele örnekleme, Genişletilmiş Lagrange Çarpanları ve Tekil Değer Eşiklemesi yöntemlerinin karşılaştırması

İkinci deneyde yöntemimizi, Genişletilmiş Lagrange Çarpanları (Augmented Lagrange Multipliers)[14] ve Tekil Değer Eşikleme (Singular Value Thresholding)[5] yöntemleriyle karşılaştırdık. Bunun için rasgele oluşturulmuş 10 adet 200×1000 büyüklüğünde matrisler $\%50$ oranlarında maskelendi ve her matris üzerinde üç algoritma koşturuldu. Koşma zamanları ve görece hatalar tablo I'de verilmiştir.

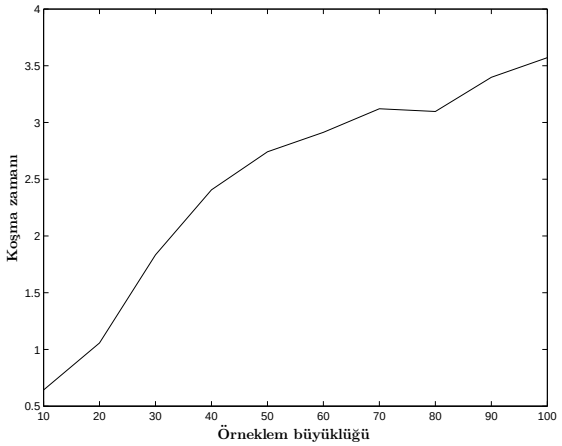
Üçünce deneyde geliştirdiğimiz algoritmayı Olivetti yüz verikümesi üzerinde sınadık. Olivetti yüz verikümesinde 64×64 piksel büyüklüğünde 400 adet siyah beyaz yüz fotoğrafı bulunmaktadır, yani veri matrisi 4096×400 büyüklüğündedir. Deney için her fotoğrafın rasgele bir dörtte birlik bloğunu maskeledik, geliştirdiğimiz algoritmayı rank girdisi 20 ve örneklem büyüklüğü 100 satır olarak maskelenmiş veri üzerinde koşturduk. Şekil 3a'da verikümesinden 4 örnek, Şekil 3b'de bu fotoğrafların maskelenmiş, Şekil 3c'de de bu algoritma tarafından tamamlanmış halleri görülmektedir.

IV. VARGILAR

Şimdiye kadar önerilmiş düşük rank matris tamamlama problemleri, devasa matrisler söz konusu olduğunda, bilinen değerlerin asıl matrisin seyrek bir alt kümesi olduğunu varsaymıştır. Bizim önerdiğimiz yöntem ise böyle bir varsayımda bulunmamaktadır. Geliştirdiğimiz yöntemin matematiksel analizi henüz yapılmamıştır, fakat deneysel sonuçlar yöntemin uygulanabilirliği açısından yüksek oranda umut vaadeden bir görünüm çizmektedir.



(a) Hata ortalamaları - log ölçeğinde



(b) Zaman ortalamaları

Şekil 2: 500×2000 büyüklüğünde 10 ranklı matris üzerinde deney sonuçları

(a) Asıl fotoğraflar

(b) Maskelenmiş fotoğraflar

(c) Yeniden oluşturulmuş fotoğraflar

Şekil 3: Olivetti yüz verikümesi üzerinde deney

Önerdiğimiz yöntem asıl veriyi yinelemeli bir şekilde işlese de yavaş bellek erişimi açısından verimli bir şekilde gerçeklenebilir. Burada önemli bir nokta da, dağıtımlı bilgi işleme olanaklarının günümüzdeki gelişkinliğini göz önünde bulundurursak, önerdiğimiz algoritmanın yüksek oranda paralellenebilir olmasıdır. Yakın süreli araştırmalarımız bu konunun matematiksel analizi ve rasgeleleştirilmiş algoritmaların düşük rank matris tamamlama problemine uygulanabilirliği açısından daha verimli yöntemler bulmak üzerine odaklıdır.

KAYNAKÇA

[1] C. Bregler, A. Hertzmann, H. Biermann *Recovering non-rigid 3D shape from image streams* In Proc. CVPR, 2000

[2] D. Achlioptas, F. McSherry *Fast Computation of Low-Rank Approximations*,Journal of ACM,54(2),Article 10,2007

[3] E. J. Candès, B. Recht *Exact matrix completion via convex optimization*, Found. of Comput. Math., 9 717-772.2008

[4] J. D. M. Rennie, N. Srebro *Fast maximum margin matrix factorization for collaborative prediction* In Proceedings of the International Conference of Machine Learning, 2005

[5] J. F. Cai, E. J. Candès, Z. Shen *A singular value thresholding algorithm for matrix completion* SIAM J. on Optimization 20(4), 1956-1982, 2008

[6] L. Trefethen, D. Bau, *Numerical Linear Algebra*, Siam, Philadelphia, 1997

[7] N. Halko, P. G. Martinsson, J. A. Tropp *Finding Structure With Randomness: Stochastic Algorithms for Constructing Approximate Matrix Decompositions*,Technical Report,CalTech,2009

[8] N. Srebro, T. Jaakkola, *Weighted Low-Rank Approximations*, In T.Fawcett and N. Mishra, editors, ICML, pages 720-727, AAAI Press,2003.

[9] P. Drineas, R. Kannan, M. W. Mahoney, *Fast monte carlo algorithms for matrices III: computing a compressed approximate matrix decomposition*,SIAM Journal on Computing, 36(1), pp. 184-206, 2006.

[10] R. Basri, D. Jacobs, I. Kemelmacher *Photometric stereo with general, unknown lighting* In IJCV, 72(2):239-257, 2007

[11] T. Okatani, T. Yoshida, K. Deguchi, *Efficient algorithm for low-rank matrix factorization with missing components and performance comparison of latest algorithms*,Computer Vision (ICCV), IEEE International Conference, 2011

[12] Y. Koren, R. Bell, C. Volinsdky *Matrix factorization techniques for recommender systems*, Computer, 42(8):30-37, 2009

[13] Y. J. Lim, Y. W. Teh, *Variational Bayesian Approach to Movie Rating Prediction*, KDD Cup and Workshop 2007

[14] Y. Zheng, G. Liu, S. Sugimoto, S. Yan, M. Okutomi, *Practical low-rank matrix approximation under robust L1-norm*,IEEE Conference on Pattern Recognition and Computer Vision (CVPR) ,2012.