

# 빅데이터 분석을 위한 Rank-Sparsity 기반 신호처리기법

이혁, 이형일, 조재학, 김민철, 소병현, 이정우  
서울대학교

## 요약

주성분 분석 기법(PCA)은 가장 널리 사용되는 데이터 차원 감소 (dimensionality reduction) 기법으로 알려져 있다. 하지만 데이터에 이상점 (outlier)가 존재하는 환경에서는 성능이 크게 저하된다는 단점을 가지고 있다. Rank-Sparsity(Robust PCA) 기법은 주어진 행렬을 low-rank 행렬과 저밀도 (sparse)행렬의 합으로 분해하는 방식으로, 이상점이 많은 환경에서 PCA기법을 효과적으로 대체할 수 있는 알고리즘으로 알려져 있다. 본 고에서는 RPCA 기법을 간략히 소개하고, 그의 적용분야, 및 알고리즘에 관한 연구들을 대해서 알아본다.

## I. 서론

데이터 수집 능력의 진보와 데이터 저장 용량의 증대로 인해, 정보 처리 능력의 과부하 문제가 과학 및 공학의 전 영역에 걸쳐 대두되고 있다. 다양한 공학, 우주 과학, 생명 과학, 경제학 등의 분야에서 수집되고 있는 관측 데이터의 양은 기하급수적으로 증가하고 있으며, IDC에 따르면, 2003년까지 인류에 의해 생성된 총 데이터량은 5 exabyte ( $10^{18}$ ) 인 반면, 2020년까지 생성될 데이터의 총 량은 40 zettabyte ( $10^{21}$ )에 달할 것으로 추정되고 있다(그림1).

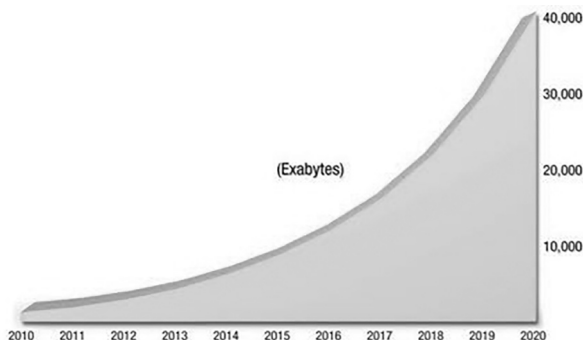


그림 1. 데이터의 급속한 증가 (출처 : IDC)

이와 같이 범람하는 데이터 속에서 부가가치를 창출하려는 시도는 빅데이터(big data)라는 개념에 대한 관심으로 나타나고 있다. 과거에도 데이터 마이닝 (data mining), 비즈니스 인텔리전스(business intelligence) 등을 통한 데이터 활용에 대한 시도는 있었지만, 단순 생성과 소멸의 반복이 아닌 유의미한 정보를 부가적으로 생성하기 위한 시도는 빅데이터라는 이름으로 재부상하였다. 맥킨지가 빅데이터의 개념을 “전통적인 데이터 베이스 소프트웨어로는 수집, 저장, 관리, 분석이 어려운 정도의 큰 규모의 데이터”로 정의할 만큼 비교적 작은 양의 데이터들을 처리하기 위해 만들어진 기존의 데이터 분석 기법 및 알고리즘으로는 실시간으로 생성되는 다양한 형태의 데이터를 처리하는 것이 불가능하다는 점 또한 빅데이터 처리 기법에 대한 관심을 증폭시킨다.

데이터 분석의 복잡도를 줄이기 위해서는 데이터에서 의미 있는 변수 (latent variable)들을 추출하는 차원 감소 기법의 사용이 필수적인데, 주성분 분석 기법(principal component analysis, PCA)이 가장 널리 쓰인다. PCA 기법은 평균-제곱 에러(mean-square error) 최소화 측면에서 최적의 성능을 가진다고 알려져 있지만, 관측된 데이터에 크게 훼손된 일부의 성분, 즉 이상점(outliers)들이 존재할 때, 성능이 크게 저하되는 단점을 가지고 있다. 본 논문에서는 이러한 이상점이 많은 환경에서도 강건하게 동작하는 알고리즘으로 알려진 Robust PCA 기법을 간략히 소개한다. 2장에서 RPCA를 이해하기 위한 배경지식으로써 차원 감소 기법과 PCA 기법에 대해 설명하며, 3장에서 RPCA 기법을 소개하고 수학적으로 기술한다. 4장에서는 그의 실제 적용 분야들을 살펴보고, 5장에서는 RPCA 기법의 복잡도를 줄이기 위한 연구 및 그 알고리즘들을 간략하게 소개한다. 6장에서는 RPCA 기법의 실시간 사용을 위한 projection-based 알고리즘에 대해 살펴본다.

## II. 배경 지식

### II.1. 차원 감소 기법

기존 데이터 분석 기법들의 실패의 원인을 “빅(Big)” 데이터,

즉 단순히 데이터의 양이 많아졌기 때문이라고 말하기보다는, 데이터의 차원이 커졌기 때문이라고 보는 것이 더 타당한 표현이라고 할 수 있다. 데이터의 차원 (dimension)이란, 어떤 한 데이터를 관측하고, 측정하는데 사용된 변수의 개수를 뜻한다. 예를 들면, 어떤 영역의 온도를 측정하는 데 쓰인 온도 센서의 수, 촬영된 영상의 픽셀 수 및 프레임 수 등을 예로 들 수 있다. 데이터를 나타내는 변수의 수, 즉 차원이 증가함으로써 인해 나타나는 가장 큰 특징 중 하나는, 이러한 거대한 숫자의 모든 변수들이 다 “중요한” 변수가 아니라는 점이다. 비교적 의미 있는 변수들이 있는 반면, 어떤 변수들은 설사 측정되지 않았더라도 전체 데이터의 본질을 거의 해치지 않는 변수들일 수 있다. 물론, 더 높은 차원 데이터를 분석하여 더욱 정확한 분석 결과를 내는 것도 중요한 분야라고 할 수 있지만[3], 데이터의 차원을 감소시켜 데이터 분석을 쉽고 빠르게 하면서도 유의미한 정보를 추출하는 방식을 찾아내는 것이 현실적으로 필요한 분야라고 할 수 있다.

차원 감소 기법 (dimensionality reduction)은 어떤 시스템을 기술하는데 있어 필요한 (확률) 변수들 중 의미 있는 변수들만을 뽑아냄으로써, 데이터의 본래의 특성을 유지하면서도 데이터 해석의 복잡도를 줄일 수 있는 기법으로서, 거대한 차원의 데이터를 분석하기 위한 초석이 되는 이론이라고 할 수 있다.

수학적인 용어로 이 문제를 표현해 보자. 데이터의 차원을  $p$  라고 하고, 주어진 데이터를  $p$  차원의 벡터 확률 변수  $\mathbf{x} = (x_1, \dots, x_p)^T$  라고 나타내자. 어떤 주어진 기준에 의해 원래의 데이터의 특성을 보존함과 동시에  $x$  를 더 낮은 차원의 벡터  $\mathbf{s} = (s_1, \dots, s_k)^T$  (여기서,  $k \leq p$ )로 변환하는 것이 문제의 목표이다.  $x$ 에서  $s$ 로의 변환을  $f(\cdot)$  라고 했을 때(즉,  $f(\mathbf{x}) = \mathbf{s}$ ), 가 선형(linear)인 경우와 비선형(nonlinear)의 두 가지로 차원 감소 기법을 분류할 수 있다. 선형 기법은 벡터  $\mathbf{s}$ 의 각각의 성분들은  $p$  개의  $\mathbf{x}$  벡터의 성분들의 선형 조합(linear combination)으로 아래와 같이 표현할 수 있는 경우를 말한다.

$$\mathbf{s}_i = w_{i,1}x_1 + \dots + w_{i,p}x_p, \quad (i = 1, \dots, k) \text{ 또는}$$

$$\mathbf{s} = \mathbf{W}\mathbf{x}.$$

여기서,  $\mathbf{W}$ 는  $k \times p$  크기의 계수 행렬이다.  $p \times 1$  길이의 데이터 벡터가 총  $n$  개가 있어 이들을 모아  $p \times n$  행렬을 만든 것을  $\mathbf{X} = [\mathbf{x}_1 \ \mathbf{x}_2 \ \dots \ \mathbf{x}_n]$  라고 하면, 이들을 한꺼번에 다음과 같은 식으로 표현할 수 있다.

$$\mathbf{S} = \mathbf{W}\mathbf{X}$$

여기서,  $\mathbf{S} = [\mathbf{s}_1 \ \mathbf{s}_2 \ \dots \ \mathbf{s}_n]$ 의 각 열은 각 데이터벡터를 차원 감소한 결과 벡터들을 말한다.

선형 기법의 경우 비선형 기법에 비해 간단하고, 하드웨어 및 소프트웨어 구현에 용이하다. 선형 차원 감소 기법의 성능은 당

연히 적절한  $\mathbf{W}$  행렬의 선택에 의해 결정된다. 다음 절에서는 최적의 선형 차원 감소 기법으로 알려져 있는 주성분 분석기법 (principal component analysis, PCA)에 대해 살펴본다.

## II.2. 주성분 분석 기법 (PCA)

주성분 분석 기법 (principal component analysis, PCA)은 차원 감소를 위해 가장 널리 쓰이는 데이터 분석 기법 혹은 통계학적 기법으로, 평균-제곱 에러 (mean-square error, MSE)를 최소화한다는 관점에서 최적의 성능을 가지는 선형 차원 감소 기법이다[4,5]. 데이터의 차원을 줄이기 위해, PCA는 원래의 데이터 벡터  $\mathbf{x}$ 의 성분들의 가능한 선형조합  $\mathbf{w}^T\mathbf{x}$ 들 중 분산이 최대가 되는 계수 벡터  $\mathbf{w}$ 를 찾는다. 여기서 계수 벡터  $\mathbf{w}$ 는 방향 성분만을 가지며, 크기는 1로 정규화된다. 이를 수식적으로 나타내면 다음과 같다.

$$\mathbf{w}_1 = \operatorname{argmax}_{\|\mathbf{w}\|=1} \operatorname{Var}\{\mathbf{x}^T\mathbf{w}\},$$

여기서  $\mathbf{w}_1$ 을 첫 번째 계수 벡터,  $\mathbf{s}_1 = \mathbf{w}_1^T\mathbf{x}$ 를 첫 번째 주성분 (principal component, PC)라고 부른다. 두 번째 계수 벡터  $\mathbf{w}_2$ 는 처음에 선택된  $\mathbf{w}_1$ 과 직교하는 ( $\mathbf{w}_1^T\mathbf{w}_2 = 0$ ) 벡터들 중  $\operatorname{Var}\{\mathbf{w}^T\mathbf{x}\}$ 가 가장 최대가 되는 벡터를 선택함으로써 정해진다. 이러한 방식으로,  $k$ 번째 계수 벡터는 앞서 뽑힌  $\mathbf{w}_1, \dots, \mathbf{w}_{k-1}$ 과 직교하는 벡터들 중  $\operatorname{Var}\{\mathbf{w}^T\mathbf{x}\}$ 가 최대가 되는 벡터가 선택된다. 이러한 방식으로 총  $p$ 개의 벡터  $\mathbf{w}_1, \dots, \mathbf{w}_p$ 를 순차적으로 선택하는 것이 가능하며, 이렇게 선택된 벡터들을 행벡터로 갖는  $p \times p$  행렬  $\mathbf{W} = [\mathbf{w}_1 \ \mathbf{w}_2 \ \dots \ \mathbf{w}_p]^T$ 을 만들 수 있으며, 총  $p$ 개의 주성분들은 다음과 같이 나타낼 수 있다.

$$(\mathbf{s}_1, \dots, \mathbf{s}_p)^T = \mathbf{W}\mathbf{x}$$

이 식은 데이터 벡터  $\mathbf{x}$ 를 행렬  $\mathbf{W}$ 로 특정 지어지는 선형 변환으로 해석될 수 있는데, 이 변환을 카후넨-뢰브 변환 (Karhunen-Loeve transform, KLT) 또는 호텔링 변환 (Hotelling transform)이라고 하며, 주성분 분석 기법의 다른 이름으로도 불린다. 여기서, 앞서 설명한 계수 벡터  $\mathbf{w}_k$ 를 뽑는 과정에서도 유추할 수 있듯이, 주성분  $\mathbf{s}_1, \mathbf{s}_2, \dots$ 는 순서대로 중요한 성분을 뜻한다는 것을 알 수 있다.  $p$ 개의 주성분 중  $k$ 개를 선택하여  $k$ 차원의 벡터  $\mathbf{s} = (s_1, \dots, s_k)$ 를 만들면 주요한 성분들만 뽑아 차원 감소를 할 수 있다. 이를 식으로 나타내면 다음과 같다.

$$\mathbf{s} = \mathbf{W}_{k \times p}\mathbf{x}$$

## II.2. 주성분의 Low-rank matrix 형태 표현

앞서 설명하였듯이, 데이터벡터  $n$  개를 모아  $p \times n$  행렬  $\mathbf{X} = [\mathbf{x}_1 \ \mathbf{x}_2 \ \dots \ \mathbf{x}_n]$ 로 나타내었다고 가정해보자. 카후넨 로브

변환을 거치면, 다음과같이 주성분 벡터 행렬  $\mathbf{S}_{p \times n}$  를 얻는다.

$$\mathbf{S}_{p \times n} = \mathbf{W}\mathbf{X}$$

행렬  $\mathbf{W}$  는 직교 정규화 행렬이므로,  $\mathbf{W}^T\mathbf{W} = \mathbf{W}\mathbf{W}^T = \mathbf{I}_p$  를 만족하며, 다음 식이 성립한다.

$$\mathbf{X} = \mathbf{W}^T\mathbf{S}_{p \times n} = \mathbf{W}_{k \times p}^T\mathbf{S}_{k \times n} + \mathbf{W}_{(p-k) \times p}^T\mathbf{S}_{(p-k) \times n} = \hat{\mathbf{X}} + \mathbf{E}$$

여기서,  $\mathbf{W}_{k \times p}$  는  $\mathbf{W}$  의 첫  $k$  개의 행만을 가지는 행렬,  $\mathbf{W}_{(p-k) \times p}$  는 아래의  $p - k$  개의 행만을 가지는 행렬을 뜻하고,  $\mathbf{S}_{k \times n}$  는  $k$  개의 주성분만을 모아 만든 주성분 벡터 행렬이고,  $\mathbf{S}_{(p-k) \times n}$  은 나머지  $p - k$  개의 주성분을 가진 행렬이다. 위 식에서  $\hat{\mathbf{X}} = \mathbf{W}_{k \times p}^T\mathbf{S}_{k \times n}$  는 원래의 데이터 행렬을  $k$  개의 주성분 벡터로 사영한 데이터로 볼 수 있으며, 이 행렬은 크기는  $p \times n$  이지만  $k$  의 rank를 가지고 있는 행렬이다. 이 행렬  $\hat{\mathbf{X}}$  는 다음과 같은 norm 최소화 문제의 해가 됨이 알려져 있다.

$$\hat{\mathbf{X}} = \arg \min_{\mathbf{X}'} \|\mathbf{X} - \mathbf{X}'\|_F^2, s. t. \text{rank}(\mathbf{X}') \leq k$$

여기서  $\|\cdot\|_F^2$  는 Frobenius norm의 제곱으로, 행렬의 각 성분의 제곱의 합을 뜻하며, 즉,  $\hat{\mathbf{X}}$  이  $\mathbf{X}$  의 추정 값이라면,  $\|\mathbf{X} - \hat{\mathbf{X}}\|_F^2$  는 평균 제곱 에러를 뜻한다고 할 수 있다. 즉  $\hat{\mathbf{X}}$  은 rank가  $k$  이하인 행렬, 즉  $k$  개의 성분으로 나타낼 수 있는 행렬들 중 원래 행렬  $\mathbf{X}$  와 평균 제곱 에러 관점에서 가장 비슷한 행렬이다.

### III. Robust Principal Component Analysis

#### III.1. 주성분 분석 기법(PCA)의 문제점

앞 절에서, PCA는 선형 차원 감소기법 중에 평균-제곱 에러 최소화의 관점에서 가장 최적인 방식을 논하였다. 하지만 이는 데이터 행렬  $\mathbf{X}$  의 모든 원소가 큰 오차 없이 관측되었을 때에 국한된 결과이며, 실제 많은 상황에서는 데이터를 측정하는 센서의 고장, 일시적인 손상 등에 의해 많은 다른 원소들은 제대로 관측되었지만,  $\mathbf{X}$  의 일부 원소들이 완전히 관측되지 못하거나, 큰 값의 오차들이 발생하는 일이 발생할 수 있다. 이러한 크게 훼손되거나 관측되지 못한 불완전한 원소들을 이상점(outlier)이라고 부르는데, PCA는 작은 크기의 에러들이 데이터 전반적으로 퍼져있는 경우와는 달리, 이러한 이상점들이 곳곳에 분포되어있는 경우에 굉장히 큰 성능 하락을 보이는 것으로 알려져 있다.

이를 수식적으로 표현해보자. 관측된 행렬  $\mathbf{X}$  는 데이터 행렬  $\hat{\mathbf{X}}$  와 에러 행렬  $\mathbf{E} = (\mathbf{X} - \hat{\mathbf{X}})$  의 합으로 분리하여 생각할 수 있다. 데이터 행렬이 적은 개수의 주성분들로만 이루어졌다고

하면, 우리는  $\hat{\mathbf{X}}$  을 낮은 rank를 가지는 행렬이라고 근사할 수 있다. 이제 에러 행렬  $\mathbf{E}$  를 특정짓는 두 가지의 척도를 생각해 보자 첫 번째는  $\|\mathbf{E}\|_F$  로, 에러의 평균 제곱 에러를 뜻한다. 이는 데이터 전체의 성분을 따라 어느 정도의 평균적인 손실이 일어났는지를 의미한다. 두 번째 척도는 에러 행렬의  $l_0$  norm인  $\|\mathbf{E}\|_0$  로, 에러 행렬의 성분들 중 0이 아닌 성분들의 총 수를 의미한다. 이 값은 손상된 성분의 수에는 비례하지만, 전체적으로 어느 정도 손실되었는지에 대한 정보는 반영하지 않는다. PCA는 평균 제곱 에러를 나타내는  $\|\mathbf{E}\|_F$  가 가장 낮은 해를 구하는데에 최적화되어 있지만, 데이터의 일부만 손상되어  $\|\mathbf{E}\|_0$  은 적지만  $\|\mathbf{E}\|_F$  가 큰 상황에서는 좋은 예를 구할 수 없다. 이러한 이상점들이 많은 경우에도 잘 작동하는 PCA 기법을 만들기 위한 연구들이 최근 연구되고 있지만[5-9], 충분히 넓은 실용적인 조건하에서 적용할 수 없고, 데이터 크기에 지수적으로 증가하는 높은 복잡도를 및 에러 성능 등의 측면에서 좋은 성능을 나타내고 있지 못하다.

#### III.2. Robust PCA (RPCA)

앞서 설명한 수식적 표현에 따르면, 관측된 행렬  $\mathbf{X}$  는 실제 데이터 행렬  $\hat{\mathbf{X}}$  와 에러 행렬  $\mathbf{E}$  의 합으로 나타내어진다.

$$\mathbf{X} = \hat{\mathbf{X}} + \mathbf{E}$$

여기서, 데이터가 이미 아주 적은 수의 주성분의 조합으로 이루어져있다고 가정하면, 데이터 행렬  $\hat{\mathbf{X}}$  는 rank가 아주 낮다고 가정할 수 있다. 또한, 관측의 과정에서 일부 원소들만 손상되었고, 대부분의 원소는 아주 정확히 측정된되어, 에러 행렬  $\mathbf{E}$  는 거의 모든 수의 성분이 0이고, 일부의 성분들만 0이 아닌 저밀도(sparse)의 특성을 가진다고 가정해보자. 즉 관측행렬  $\mathbf{X}$  는 low-rank 행렬과 sparse 행렬의 합으로 나타내어진다고 가정할 수 있다. 다음과 같은 최적화 문제를 생각해 보자.

어떤 low rank 행렬  $\mathbf{L}_0$  와 저밀도 행렬  $\mathbf{S}_0$  의 합으로 이루어진  $\mathbf{X}$  라는 행렬이 주어졌을 때,

$$\begin{aligned} & \text{minimize } \|\mathbf{L}\|_* + \lambda\|\mathbf{S}\|_1 \\ & \text{subject to } \mathbf{L} + \mathbf{S} = \mathbf{X}. \end{aligned}$$

여기서  $\lambda$  는 미리 정해진 상수이며,  $\|\mathbf{M}\|_*$  은 nuclear norm으로 행렬  $\mathbf{M}$  의 모든 특이값(singular value)들의 합이며,  $\|\mathbf{M}\|_1$  은  $l_1$ -norm으로 행렬  $\mathbf{M}$  의 모든 원소의 절대값들의 합이다. 우린 두 가지의 의문을 던질 수 있다.

- (1) 위의 최적화 문제의 답을 실용적인 방식으로 구해낼 수 있는가?
- (2) 구할 수 있다면, 그 답  $\mathbf{S}$  와  $\mathbf{L}$  은 유일한가? 그리고,  $\mathbf{S} = \mathbf{S}_0$ ,  $\mathbf{L} = \mathbf{L}_0$  를 만족하는가?

2009년 Candes 등은, 행렬의 크기가 커질 때, 특정 타당한 조건 하에서 (2)의 조건과 같이 유일하며 정확한 해를 구할 수 있는 확률이 1에 수렴한다는 것을 증명하였다[1]. 즉, 이 결과를 앞서 언급한 이상점이 있는 조건 하에서의 주성분 분석의 문제에 적용해 보면,  $\hat{\mathbf{X}}$  데이터 행렬의 rank가 충분히 낮고, 여러 행렬  $\mathbf{E}$  가 충분히 저밀도의 특성을 갖는다면, 이 둘의 합  $\mathbf{X}(=\hat{\mathbf{X}}+\mathbf{E})$  만 관측되어 알고 있더라도 두 행렬  $\hat{\mathbf{X}}$  와  $\mathbf{E}$  를 완벽히 분해하여 각각을 복구할 수 있다.

우리는 또한 (1)의 조건과 같이 이 문제를 푸는 것이 실용적으로 가능한가에 대해서도 살펴보아야 한다. 아무리 이 문제가 유일하며 정확한 해를 갖더라도, 이 해를 구하는 데에 아주 많은 시간과 노력이 필요하다면 소용이 없을 것이다. [2]에서는 이 최적화 문제에 PCP (Principal component pursuit)라고 이름붙이고, PCP를 convex 최적화 문제의 유형 중 하나인 SDP (semi-definite programming)로 변환하는 것이 가능하다는 것을 증명하였다. Convex 최적화는 최적화 문제의 한 예로, 국소해 (local minimizer)가 일반해 (global minimizer)와 같다는 특성을 가져, 문제의 해결이 비교적 쉽다는 것이 알려져 있다. 하지만 interior point 방식과 같은 범용적인 convex solver들은 여전히 높은 복잡도를 가지므로, PCP 문제에 최적화된 저복잡도 알고리즘을 개발하기 위한 연구도 활발히 진행되고 있다. 이러한 PCP 문제의 실용적인 구현 알고리즘들에 대한 논의는 5장에서 계속한다.

다음 절에서는 PCP 문제가 해결되기 위한 조건들에 대하여 살펴보고, 이 조건들이 현실적으로 타당한 것인지에 대해서 살펴본다.

### III.3. rank-sparsity incoherence conditions

$\mathbf{X} = \mathbf{L}_0 + \mathbf{S}_0$  가 주어졌을 때, 사실은,  $\mathbf{L}_0$  의 rank가 아주 적다는 조건과  $\mathbf{S}_0$  의 0이 아닌 원소가 아주 작다는 조건만으로는 PCP 문제에 의해 올바른 해가 구해진다는 것을 보장할 수 없다. 이러한 실패는 두 가지 상황에서 일어난다.

#### (1) $\mathbf{L}_0$ 가 너무 sparse한 경우

$\mathbf{X} = \mathbf{e}_1 \mathbf{e}_1^T$  인 경우를 생각해보자. 여기서  $\mathbf{e}_1$  은 1번째 성분만 1이고 나머지 성분은 0인 단위벡터를 뜻한다. 이 경우,  $\mathbf{X}$ 는 1행 1열의 성분만 1이며, 나머지 성분은 모두 0인 행렬로, 동시에 low-rank이면서 sparse한 성질을 가진다. 이 경우 당연히  $\mathbf{L}$  과  $\mathbf{S}$  에 어떤 값을 넣어야 하는지 불분명해진다. 이 문제를 해결하기 위해,  $\mathbf{L}$  의 열공간(column space)과 행공간(row space)가 모두 standard basis의 벡터들 즉  $\{\mathbf{e}_1, \mathbf{e}_2, \dots, \mathbf{e}_n\}$  과 비상관(incoherence)되어 있어야 한다는 조건을 추가로 고

려해야 한다. “비상관되었다”는 용어의 의미는, 열공간이나 행공간 안의 임의의 벡터들을 잡았을 때, 그 벡터와 임의의  $\mathbf{e}_i$  벡터의 내적이 1에 비해 충분히 작은 값이어야 한다는 뜻이다. 즉, 그 벡터의 power가 소수의 성분에 집중되어 있어서는 안되고, 전체 성분에 고르게 퍼져있어야 한다는 뜻이다. 이를 만족할 때, low-rank 행렬  $\mathbf{L}_0$  는 sparse한 특성을 가지지 않게 된다. 이 조건을 수식적으로 표현하면 다음과 같다.

$n_1 \times n_2$  행렬  $\mathbf{L}_0$  의 rank가  $r$  이며,  $\mathbf{L}_0 = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^H$  일 때(특이값 분해), 어떤 파라미터  $\mu$  에 대해

$$\max_i \|\mathbf{U}^H \mathbf{e}_i\|^2 \leq \frac{\mu r}{n_1}, \quad \max_i \|\mathbf{V}^H \mathbf{e}_i\|^2 \leq \frac{\mu r}{n_2} \quad \text{이고}$$

$$\|\mathbf{U}\mathbf{V}^H\|_\infty \leq \sqrt{\frac{\mu r}{n_1 n_2}}$$

#### (2) $\mathbf{S}_0$ 의 rank가 너무 작은 경우

$\mathbf{S}_0$  의 0이 아닌 성분들이 소수의 행이나 소수의 열로 모여 있을 때에도 문제가 생긴다. 예를 들어,  $\mathbf{S}_0$  의 첫번째 행은 모두 0이 아니고, 나머지 성분들은 다 0이라면, 이 행렬은 충분히 0이 많은 저밀도 행렬이 되지만, 임의의 low-rank 행렬  $\mathbf{L}_0$  와  $\mathbf{S}_0$  를 더해도 rank가 최대 1밖에 늘어나지 않는다. 이 경우  $\mathbf{L} = \mathbf{L}_0 + \mathbf{S}_0$  과  $\mathbf{S} = \mathbf{0}$  도 충분히 PCP의 해가 될 수 있다. 이처럼  $\mathbf{S}_0$  의 rank가 너무 작아지는 경우를 막기 위해,  $\mathbf{S}_0$  의 0이 아닌 원소의 위치가 모든 위치에 대해 균일한 확률로 선택된다 (uniformly at random)고 가정해야 한다. 이 경우, 행렬의 크기가 커질수록  $\mathbf{S}_0$  가 full rank가 될 확률이 1에 수렴하게 된다.

## IV. Robust PCA의 적용분야

본 절에서는 영상 데이터에서 전경(foreground, 영상에서 빠르게 변화하는 의미 있는 부분)과 배경(background, 시간에 따라 잘 변하지 않는 부분)의 분리, 얼굴 인식, 추천 알고리즘 등과 같은 robust PCA의 적용 분야에 대해 살펴본다.

### IV.1. 영상에서의 전경과 배경의 분리

영상은 여러 개의 프레임으로 이루어지는데, 이 프레임들은 대부분이 닮아있으며, 이들 간에는 커다란 상관관계 (correlation)가 존재한다. 각 프레임의 모든 픽셀의 데이터들을 일렬로 세워 하나의 열 벡터로 만들고, 이를 모아 데이터 행렬을 만들 수 있다. 이 행렬에 Robust PCA를 적용하면 영상에서 배경 (background) 성분이 low-rank 행렬로, 전경

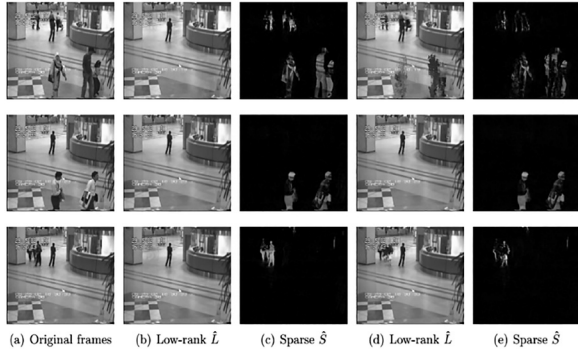


그림 2. 감시카메라 영상의 전배경 분리의 예

(foreground) 성분이 sparse 행렬로서 분해된다. 여기서 배경 성분은 영상에서 시간에 따라 잘 변하지 않는 부분, 전경 성분은 시간에 따라 변화하는 성분으로 생각할 수 있다. 배경 성분은 거의 모든 프레임에서 같은 값을 가지고, 밝기의 변화는 이 low-rank 행렬 성분의 계수의 변화로 표현 가능하다. 또한 움직이는 사람이나 물체를 뜻하는 전경 성분은 총 픽셀에서 일부 분만을 차지하기 때문에, 데이터 행렬에서 충분히 sparse하다고 가정할 수 있다.

구체적으로, <그림 2>의 예시를 통해 이러한 전배경 분해의 과정을 살펴보자. <그림 2>의 (a)는 공항의 감시 카메라 영상이고 해상도는 176144 픽셀이다. 또한 gray-scale 영상으로 총 200 프레임으로 구성되어 있다. 따라서 Robust PCA를 통해 분해하고자 하는 행렬을  $\mathbf{M}$  이라 하면,  $\mathbf{M} \in \mathbb{R}^{25,344 \times 200}$  이 된다. 행렬  $\mathbf{M}$  을 낮은 계수의 행렬  $\hat{\mathbf{L}}$  과 Sparse 행렬  $\hat{\mathbf{S}}$  로 분해한 결과를 각각 <그림 2>의 (b)와 (c)에서 확인할 수 있다. Robust PCA의 성능을 비교하기 위해 Alternating minimization [10]을 이용하여 행렬  $\mathbf{M}$  을 분해한 결과를 <그림 2>의 (d)와 (e)에서 확인할 수 있다. Robust PCA를 이용한 결과인 <그림 1>의 (b), (c)가 배경 부분과 전경 부분을 완전히 분해한 것에 비해, Alternating minimization을 통한 분해는 그렇지 못한 부분이 있음을 확인할 수 있다. 앞서 말했듯이, RPCA를 통한 이러한 영상 분석은 배경 성분을 특정 이미지로 처리하는 것이 아니라 여러 개의 성분의 조합인 low-rank 행렬로 간주하기 때문에, 조명이 급격하게 변하는 환경에서도 적용할 수 있으며, 여전히 좋은 분해 성능을 보인다[1]. 뿐만 아니라, 잡음 신호나 오차에 대해서도 기존의 기법들보다 우세한 성능을 나타낸다.

Robust PCA에 요구되는 비용을 감소하고 속도를 향상시키기 위한 한 방법으로 압축 센싱 (Compressive sensing)을 도입하는 방법이 제안되었다[11]. 압축 센싱 기법을 도입하여 영상의 일부분만으로도 성능의 저하 없이 행렬을 낮은 계수의 행렬과

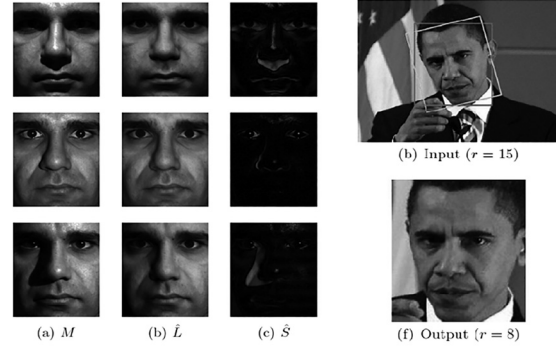


그림 3. Face recognition 응용

Sparse 행렬로 분해할 수 있으므로 비용을 줄이고 속도를 향상시킬 수 있다.

## IV.2. 얼굴 인식에의 응용

Robust PCA는 이미지나 영상 자료에서 물체 인식에도 응용될 수 있다. 아래의 <그림 3>의 왼쪽 그림은 얼굴 인식에 Robust PCA 알고리즘을 사용한 예이다. 실제 얼굴 이미지는 얼굴에 생기는 투영 그림자나 반사된 빛으로 인해 얼굴인식에 있어 오차가 생긴다. 이러한 오차의 크기는 꽤 큼에도 불구하고, 공간 도메인에서 sparse하기 때문에 똑같은 얼굴 이미지가 충분히 많이 있다면, Robust PCA를 통해 이러한 오차를 없앨 수 있다.

Robust PCA를 통해 각 프레임을 하나의 열 벡터로 묶어서 얻어진 행렬을 low rank 행렬과 sparse 행렬로 분해할 수 있다. 여기서 sparse 행렬은 원래 이미지의 투영 그림자와 반사된 빛에 해당하고, 나머지 부분은 low rank 행렬에 해당된다. 여기서 원래 이미지로부터 Robust PCA를 통해 각 프레임들의 공통된 low rank 행렬을 추출함으로써 이미지 획득에서 생기는 오차를 제거할 수 있다. 이러한 결과는 얼굴인식에 있어서의 기술에 유용하게 사용될 수 있다. 또한, 조명변화가 있는 환경에서 얼굴의 정렬 혹은 추출하는데도 사용될 수 있다.

<그림 3>의 오른쪽 그림은 다른 방식으로 이미지 자료에 Robust PCA를 적용한 예이다. 분해할 행렬  $\mathbf{M}$  (low-rank 성분과 sparse 성분이 합쳐진 행렬)에 대한 관측이 완전하지 않거나, 어떤 공간에 투영하는 경우, 또는 이미지가 선형변환 되었을 경우에도 변형된 Robust PCA를 쓰면 low rank 행렬과 sparse 행렬을 얻어내는 것이 가능하다[13]. <그림 3>의 오른쪽 그림은 어떤 이미지에서 특정한 물체가 늘어져 있거나 회전되어있는 상황에서 원래의 texture를 구별해내는 예시를 보여준다. 사람의 얼굴이나, 사람이 만든 물건과 같이 대칭성이나 규칙성을 가진 경우에는 해당 물체의 texture가 어떤 특정한 각도에 있을

때 아주 작은 rank를 갖는다고 알려져 있다. 이를 이용하여, 어떤 영상 자료나 이미지 자료에서 특정 물체를 구별하는 물체 인식 알고리즘에 적용이 가능하다[12].

### IV.3. Matrix completion과 추천 알고리즘

Matrix completion은 어떤 matrix의 일부 원소만을 알고 있을 때, 나머지 부분을 예측하는 문제 및 그 해결 방법을 뜻한다.

Matrix completion이 사용된 대표적 예로 Netflix사의 영화 추천 알고리즘을 들 수 있다. [14] Netflix사의 사용자들이 많은 영화들 중에 몇 개의 순위를 매기면, 그 사용자가 좋아할만한 영화를 추천해주는 알고리즘이다. 사용자들의 취향은 다양하겠지만 취향에 영향을 미치는 요소 (예: 장르, 주연 배우 등) 은 많지 않다. 즉 사용자들이 매긴 순위들의 정보는 연관성이 크므로 Matrix completion 기법으로 사용자의 선호도를 예측할 수 있다. 또한 요즘 급속도로 활성화되고 있는 트위터나 페이스북 같은 SNS 서비스에서 어떤 user의 다른 사람들에 대한 선호도를 분석하여 친구를 추천하는 알고리즘에 적용해보는 것도 가능할 것으로 예상되며, 이 외 분야에도 System identification, Sensor network, Global positioning 등에도 응용 가능하다.

Robust PCA 기법은 matrix completion에 그대로 활용할 수 있다. 일부 원소만 채워진 matrix를 PCP를 이용하여 분해하면 비어있던 low-rank 성분을 채울 수 있다. 즉, matrix completion이 PCP를 통해 가능하다. PCP 알고리즘은 기존의 convex solver를 이용한 방법보다 복잡도가 훨씬 작으며, 성능의 하락도 거의 없다. 또한 다른 복잡도 감소를 시도한 알고리즘들 보다 훨씬 robust한 성능을 보인다 [18]. 만약 matrix에 영향을 미친 noise가 sparse하다면 robust PCA를 활용했을 때 더 좋은 성능을 기대 할 수 있다.

Matrix completion은 Robust PCA 뿐만 아니라 여러 가지 norm minimization 관점에서 해석될 수 있다. [14]에서는 matrix completion을 nuclear norm 최소화 이론에 기반하여 복원에 필요한 최소 sampling 개수를 수학적으로 유도하였으며, [15]에서는 이 결과를 관측 잡음을 고려하였을 경우로 발전시켰다. [16]에서는 [14][15]에서보다 계산복잡도를 낮추어 massive data set에도 적용하기 유용한 알고리즘을 제안하였다. 이 외에도 특정 상황을 가정하고 matrix completion의 성능을 수학적으로 유도하는 연구들도 활발하게 이루어지고 있다. [17]

### IV.3. Text data analysis

매우 많은 다양한 문서들이 주어졌을 경우 문서들에서 공통적으로 사용되는 text부분과 각 문서 별로 다르게 사용된 부분을

구분하게 되면 문서의 유사성이라든가 빅데이터의 공통적인 부분을 찾아내는데 매우 유리할 수 있다.

문서의 공통적인 부분은 low rank matrix(L)로 표현되고, 다른 부분은 sparse matrix(S)로 표현할 수 있으므로 robust PCA 기법을 적용하여 주어진 빅데이터를 L과 S로 구분해 낼 수 있다. 이러한 방법은 금융, 보험, 의료, SNS 데이터 분석에 매우 유용하게 사용될 수 있다.

## V. RPCA의 구현 알고리즘

### V.1. Interior point 방법 [19]

기본적으로 RPCA 문제는 convex 문제 중 하나인 SDP (semi-definite programming)의 형태를 갖기 때문에, interior point 방법과 같은 off-the-shelf, 즉 범용적인 convex solver 및 SDP solver 등으로도 풀 수 있다. Interior point 방식은 적은 iteration으로 빠르게 최적해에 converge 하는 특징을 가지지만, PCP 문제에 적용되었을 경우 각각의 iteration 단계에서의 계산량의 complexity가  $n^6$ 에 비례하므로 실용적으로 사용되기에는 부적합하다.

### V.2. Iterative thresholding 방법[20]

Interior Point 방법이 높은 복잡도를 줄이기 위해 목적 함수 (object function) 1차 정보만을 사용하는 first-order 알고리즘들이 개발되었다. [20]에서는 각 iteration마다 singular value를 단계적으로 줄여나감 (thresholding) 으으로써 nuclear norm 최소화를 수행하여 각 iteration이 차지하는 복잡도를 낮추는 방법이 제시되었다. Iterative Thresholding은 기존의 문제를 아래와 같이 완화시켜 근사적인 해를 구하는 것으로 해석할 수 있다.

$$\begin{aligned} \min_{\mathbf{A}, \mathbf{E}} \|\mathbf{A}\|_* + \lambda \|\mathbf{E}\|_1 + \frac{1}{2\tau} \|\mathbf{A}\|_F^2 + \frac{1}{2\tau} \|\mathbf{E}\|_F^2 \\ \text{s. t. } \mathbf{A} + \mathbf{E} = \mathbf{D} \end{aligned}$$

여기서  $\tau$ 는 objective function의 perturb를 제한시켜주는 상수이다.

Interior point기법에 비해 각 iteration 단계에서의 complexity는 줄어들었지만, convergence rate가 너무 낮아 해를 구하는데 많은 iteration이 필요하다는 것이 iterative thresholding 방식의 단점이다.

### V.3. Accelerated Proximal Gradient 기법(APG) [21]

[22]에서 proximal gradient 방식을 matrix completion에

적용시킨 APG (Accelerated Proximal Gradient) 방식이 제안되었고, [21]에서 비슷한 알고리즘을 low-rank와 sparse 행렬 분해에 이용하는 방식이 제안되었다. 수렴 속도가  $O(k^{-2})$  으로 iterative thresholding방법에 비해 50배 정도 향상되었다.

#### V.4. Augmented Lagrange Multiplier(ALM) 기법[23]

현재의 알고리즘 중 빠르고 정확하게 PCP 문제를 해결하는 기법으로 augmented Lagrange multiplier를 이용하는 방식이 있다[23]. 다음의 augmented Lagrangian function을 정의하자.

$$l(\mathbf{L}, \mathbf{S}, \mathbf{Y}, \mu) = \|\mathbf{L}\|_* + \lambda \|\mathbf{S}\|_1 + \langle \mathbf{Y}, \mathbf{M} - \mathbf{L} - \mathbf{S} \rangle + \frac{\mu}{2} \|\mathbf{M} - \mathbf{L} - \mathbf{S}\|_F^2$$

여기서 행렬끼리의 Euclidean inner product는  $\langle \mathbf{X}, \mathbf{Y} \rangle \equiv \text{trace}(\mathbf{X}^H \mathbf{Y})$  로 정의된다.

이를 풀기 위한 방법 중 대표적인 것으로 아래와 같이 반복적으로  $l(\mathbf{L}, \mathbf{S}, \mathbf{Y})$  를 최소화하는  $\mathbf{L}$  과  $\mathbf{S}$ ,  $\mathbf{Y}$ 를 update 하는 alternating directions 방식을 활용할 수 있다.

---

#### Algorithm : Principal Component Pursuit by Alternating Directions [23]

---

- 1: **initialize:**  $\mathbf{S}_0 = \mathbf{Y}_0 = \mathbf{0}, \mu > 0$
  - 2: **while** not converged **do**
  - 3: **compute**  $\mathbf{L}_{k+1} = D_\mu(\mathbf{M} - \mathbf{S}_k - \mu^{-1}\mathbf{Y}_k)$
  - 4: **compute**  $\mathbf{S}_{k+1} = S_{\lambda\mu}(\mathbf{M} - \mathbf{L}_{k+1} - \mu^{-1}\mathbf{Y}_k)$
  - 5: **compute**  $\mathbf{Y}_{k+1} = \mathbf{Y}_k + \mu(\mathbf{M} - \mathbf{L}_{k+1} - \mathbf{S}_{k+1})$
  - 6: **end while**
  - 7: **output:**  $\mathbf{L}, \mathbf{S}$
- 

이 과정 중  $\mathbf{L}_{k+1}$  를 계산할 때에 SVD를 수행하는데 이 때의 계산량이  $O(n^3)$  이고 iteration의 복잡도 중 큰 부분을 차지해 다른 알고리즘에 비해 복잡도가 낮고, 수렴 속도 또한 빠르다는 것이 실험적으로 검증되었다.

## VI. Projection based online RPCA

기본적인 RPCA 알고리즘들은 하나의 행렬을 input으로 받아 실행되는 형태이다. 시간이 감에 따라 데이터 벡터가 추가 되어 행렬의 크기가 점차적으로 커지는 경우, 크기가 커진 행렬에 대해 행렬의 분해를 계속 수행해야 하므로 복잡도가 증가하게 되며, 중복된 연산을 어느 정도 반복하게 된다. 이번 장에서

는 [24]에서 제안된 projection에 기반한 online RPCA 알고리즘을 소개한다.

### VI.1. 기존 알고리즘의 속도

우선 기존 Robust PCA 알고리즘의 속도를 분석해보자. Robust PCA 알고리즘 중 제일 빠르다고 알려진 Augmented Lagrange Multiplier(ALM)[23]의 경우의 속도를 Matlab을 이용하여 측정한 결과가 아래의 표에 나타내었다. Column의 개수는 500개로 고정시킨 후, Row의 개수를 1000개부터 100000개까지 점점 더 늘려가며 측정하였으며, 매 시행마다 random 한 Low Rank와 Sparse Rank를 만들어서 합친 후 알고리즘에 적용하였다.

행렬의 크기	수행시간(초)
1000by500	40.2050
1700by500	35.4454
3000by500	56.7030
5189by500	102.1366
9000by500	111.8676
15565by500	171.9749
27000by500	230.0294
46697by500	371.5123
81000by500	584.1112

row의 개수가 1000개일 경우에는 약 40초의 시간이 걸렸으며, row의 개수가 9000개일 경우는 111초, 더 늘어나 row의 개수가 81000개일 경우는 584초가 걸렸다. 이 결과가 의미하는 것은, 예를 들어, 들어오는 센서 데이터의 수가 500개이고 누적된 데이터가 9000개인 경우, 한 번 Robust PCA하는 데에 111초가 소요되기 때문에, 111초 이전에 들어오는 데이터에 대해서는 실시간으로 처리할 수 없다는 뜻이다. 또한 누적된 데이터 수가 늘어나면 늘어날수록 소비되는 시간은 선형적으로 더 늘어나기 때문에, 데이터의 실시간 처리가 불가능해진다.

### VI.2 Online algorithm의 필요성

온라인 알고리즘이란 시작할 때 모든 입력 정보를 가지고 있지 않고, 입력을 실시간으로 받아들이면서 처리하는 알고리즘으로, 기계 학습(Machine Learning) 분야에서 많이 연구된다. 반대로, 오프라인 알고리즘은 풀고자 하는 문제의 모든 데이터를 가지고 시작해야만 문제를 해결할 수 있다. 예를 들어, robust PCA의 응용 중 전경-배경 분리 기법에 사용된 카메라가 30fps, 1200 x 960 해상도를 가지고 있다면, 1/30초마다 1200 x 960 크기의 벡터 데이터가 들어오며, 지금까지 누적된 모든 데이터를 처리해야 하므로, 시간이 지날수록 연산량이 더

증가한다. 이러한 형태의 데이터를 처리하기 위해서는 기존의 알고리즘 대신, 성능을 약간 포기하더라도 누적된 데이터를 쓰지 않고, 들어오는 데이터만을 이용하는, online 형태의 알고리즘이 필요하다.

### VI.3 Projection based RPCA

기존 RPCA 알고리즘을 online algorithm에 적용할 수 없는 이유는 실시간으로 데이터가 들어올 때마다 지금까지 들어

rank 행렬의 column space의 차이가 점점 증가할 것이므로, 어느 정도 데이터가 쌓이면, 모든 데이터를 Robust PCA하여 low rank space를 주기적으로 업데이트 시켜야 한다. <그림 4>에 projection based Robust PCA 알고리즘을 도식적으로 표현하였다. 예를 들어 업데이트 주기를 1000이라고 하면, 1000번째마다 모든 데이터에 대해 Robust PCA를 수행하여 projection을 수행할 low-rank 행렬의 column space를 업데이트 시키고, 그 뒤의 1000개의 데이터는 업데이트 된 low rank space에

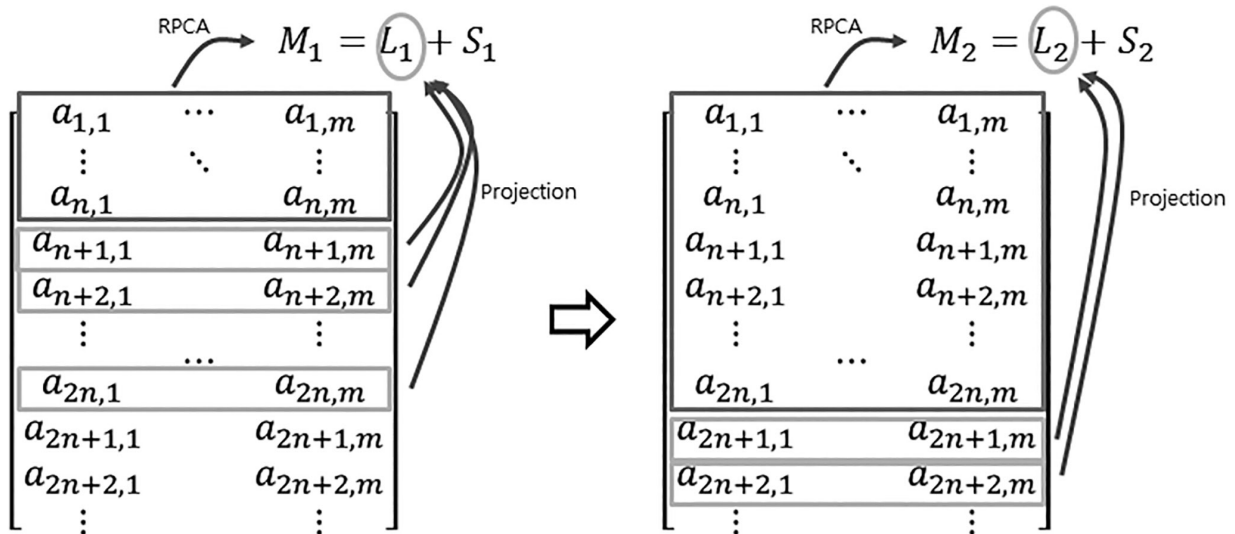


그림 4. Projection based online 알고리즘의 도식적 표현

온 모든 누적 데이터를 이용해야 한다는 점에 기인한다. 데이터가 쌓일수록 연산량이 증가하기 때문에, 시간이 지나 축적된 데이터 행렬의 크기가 커지면 실시간 처리가 불가능하게 된다. Projection based method를 이용하면, 이와 같이 누적데이터를 이용하지 않고, 새로이 들어오는 데이터를 이용하기 때문에 online algorithm에 적용할 수 있다.

Projection based method는 다음 관찰에 근거한다. 데이터의 사이즈가 매우 커지면, 데이터가 추가로 하나가 더 들어와도 Robust PCA 후의 low rank 변화가 없을 가능성이 크다. 예를 들어, 1000개의 row에 대해서 Robust PCA를 수행한 결과, 10차원의 low rank matrix를 얻었을 때, 1001번째 row가 들어오더라도, 1001개의 row에 대한 low rank는 여전히 10차원일 가능성이 높다는 뜻이다. 이러한 사실을 이용해서 매 데이터가 들어올 때마다 Robust PCA를 하지 않고, 기존의 low rank space에 projection을 시키고, error를 sparse 한 성분으로 보게 되면 계산량을 기하급수적으로 줄일 수 있다. 하지만 데이터가 많이 누적될수록 계산된 projection하는 subspace와 실제 low-

projection 시켜서 low rank matrix를 업데이트 할 수 있다. 또한 error 성분은 sparse한 성분으로 보고 sparse matrix 또한 업데이트를 시킬 수 있다. 이러한 방식은 물론 정확한 low rank space를 구하지 않기 때문에 약간의 성능 저하가 있겠지만, 매 데이터가 들어올 때 마다 Robust PCA를 할 필요 없이 단순히 projection만 시키면 되어 계산 복잡도를 크게 줄일 수 있기 때문에, online algorithm에 적용할 수 있게 된다.

### VI.4 시뮬레이션 결과

Projection based RPCA의 기존의 RPCA의 상대적인 성능을 비교하였다. 50 by 100 크기의 데이터 행렬을 random하게 생성된 low-rank 행렬과 sparse 행렬의 합하는 방식으로 생성하여 robust PCA의 input으로 사용하였으며, 새로운 데이터 열(column) 벡터가 추가되는 상황에서 매번 robust PCA를 수행하는 방식 (offline)과 첫 matrix에 대해서만 robust PCA를 수행하고 이 때에 얻은 low-rank matrix의 basis를 이용해 이후



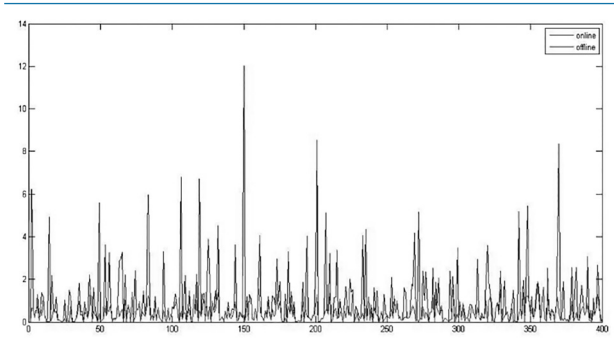


그림 5. Online 알고리즘과 기존 알고리즘의 성능 비교

에 추가되는 column에 대해서는 간단한 projection만을 수행하는 방식 (online)을 비교하였다. 또한 Offline 방식의 경우 새 frame이 들어올 때 오래된 열을 제외하여 행렬의 크기를 유지하는 방식 (moving window)과 계속하여 column을 증가시키는 (growing window) 두 가지의 경우에 대하여 테스트 하였다.

이 때 error는 원래의 low-rank 행렬과 분리된 low-rank 행렬 간의 차를 이용해 구한 여러 행렬의 평균 제곱 에러를 계산하여 상대적인 경향성을 비교하였다. 예상했던 것과 같이 앞서 설명한 online 방식으로 분리했을 때에 더 많은 error를 보였다. 수치화해서 SNR (원래 low-rank matrix의 모든 원소들의 제곱의 평균과 error값의 비율)을 기준으로 비교해보면 offline 방식은 43 dB안팎, online 방식은 33 dB 정도로 측정되었다. 적용될 시스템이 정해지지 않았으므로, 이 차이의 의미에 대한 분석이 더 필요하나, online 방식으로 수행하여도 꽤 정확하게 low-rank matrix를 추출해낸다는 것을 확인할 수 있다.

수행 시간에 있어서는 상당히 유의미한 차이를 보였다. 50 X 800 크기를 갖는 data를 simulation 할 때에 frame이 추가될 때 마다 50 X 100 matrix의 robust PCA를 총 701회 반복하여 수행하는 offline 방식은 60초의 시간이 걸렸고, 50 X 100부터 50 X 800까지 matrix의 size를 증가시킨 growing window 방식은 330초 이상이 측정된 반면, 1회의 robust PCA 이후 700회의 projection을 하는 방식은 0.2초 정도로 측정되었다. 즉, Projection based 알고리즘은 기존 방식에 비해 약간의 정확도 감소(10 dB SNR 감소)를 희생하여, application에 따라 약 300-1500 배의 속도 향상을 가져올 수 있다.

## VII. 결론

본 논문에서는 새로운 차원 감소 (dimensionality reduction) 방식으로서 학계에서 활발히 연구되고 있는 robust PCA 기법에 대해 살펴보았다. Robust PCA는 기존의 PCA기법

을 outlier(이상점)이 많은 환경에서도 잘 작동하도록 발전시켰다는 의미를 가지며, 또한 행렬의 rank minimization 기법 중 하나로도 분류될 수 있다. robust PCA기법과의 관계 및 robust PCA 기법의 수학적 기술에 대해 논하였으며, video surveillance, 얼굴인식, matrix completion 등의 RPCA의 적용 분야들에 대해서도 간단히 설명하였다. 또한, 최근 소개된 low-rank 와 sparse 행렬 분해 알고리즘들에 대해서도 간략히 소개하였다. 이러한 rank-sparsity 기반 신호처리 알고리즘은 빅데이터 분석에 매우 유용할 것이며, 또한 적용 가능한 분야가 무궁무진하리라 예상된다.

## Acknowledgements

This research was supported in part by Basic Science Research Program (NRF-2013R1A1A2008956) through the NRF funded by the MEST, Bio-Mimetic Robot Research Center funded by Defense Acquisition Program Administration (UD130070ID), INMAC and BK21-plus

## 참고 문헌

- [1] E. J. Candès, X. Li, Y. Ma, and J. Wright, "Robust Principal Component Analysis?," *Journal of ACM* 58(1), 1-37.
- [2] V. Chandrasekaran, S. Sanghavi, P.A. Parrilo and A.S. Willsky, "Rank-Sparsity Incoherence for Matrix Decomposition," *SIAM J. Optim.*, 21(2), 572-596, 2011.
- [3] L. Breiman, "Random Forests," Technical report, Department of Statistics, University of California, 2001.
- [4] J.E. Jackson. "A User's Guide to Principal Components," New York: John Wiley and Sons, 1991.
- [5] I.T. Jolliffe, "Principal Component Analysis," Springer-Verlag, 1986.
- [6] P. Huber, "Robust Statistics," Wiley and Sons, 1981

- [7] R. Gnanadesikan and J. Kettenring, “Robust estimates, residuals, and outlier detection with multiresponse data,” *Biometrics*, 28:81–124, 1972.
- [8] Q. Ke and T. Kanade, “Robust  $l_1$ -norm factorization in the presence of outliers and missing data,” In *Proceedings of IEEE International Conference on Computer Vision and Pattern Recognition*, 2005.
- [9] M. Fischler and R. Bolles, “Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography,” *Communications of the ACM*, 24:381–385, 1981.
- [10] F. De La Torre and M. Black, “A framework for robust subspace learning,” *International Journal on Computer Vision*, 54:117 - 142, 2003.
- [11] H. Jiang, W. Deng, and Z. Shen, “Surveillance Video Processing Using Compressive Sensing,” *Inverse Problems and Imaging*, Volume 6, No. 2, 201–214, 2012.
- [12] Z. Zhang, A. Ganesh, X. Liang and Y. ma, “TILT: Transform Invariant Low-Rank Textures,” *International Journal of Computer Vision*, 99(1), 1–24, 2012.
- [13] A. Ganesh, K. Min, J. Wright and Y. Ma, “Principal Component Pursuit with reduced linear measurements,” *Information Theory Proceedings (ISIT)*, 2012 IEEE International Symposium on.
- [14] E. J. Candes and B. Recht, “Exact Matrix Completion via Convex Optimization,” *Foundations of Computational Mathematics*, 9(6), 717–772, 2009.
- [15] E. J. Candes and Y. Plan, “Matrix Completion with Noise,” *Proceedings of IEEE*, 98(6), 925–936, 2010.
- [16] R. H. Keshavan, A. Montanari and S. Oh, “Matrix Completion From a Few Entries,” *IEEE Transactions on Information Theory*, 56(6), 2980–2998, 2010.
- [17] M. Wootters, Y. Plan, M.A. Davenport and E. van den Berg, “Lower bounds for quantized matrix completion,” *Information Theory Proceedings (ISIT)*, 2013 IEEE International Symposium on.
- [18] A. Waters, A. Sankaranarayanan and R. Baraniuk, “SpaRCS: Recovering Low-Rank and Sparse Matrices from Compressive Measurements,” *Proceedings of the Neural Information Processing Systems (NIPS)*, 2011.
- [19] M. Grant and S. Boyd, “CVX: Matlab software for disciplined convex programming.” <http://stanford.edu/~boyd/cvx>, 2009
- [20] J. Cai, E. J. Candes and Z. Shen, “A Singular Value Thresholding Algorithm for Matrix Completion,” *SIAM, J. Optim.*, 20(4), 1956–1982, 2010.
- [21] Z. Lin, A. Ganesh, J. Wright, L. Wu, M. Chen, and Y. Ma, “Fast convex optimization algorithms for exact recovery of a corrupted low-rank matrix,” In *Computational Advances in Multi-Sensor Adaptive Processing (CAMSAP)*, 2009.
- [22] A. Beck and M. Teboulle, “A fast iterative shrinkage-thresholding algorithm for linear inverse problems,” *SIAM Journal on Imaging Sciences*, 2(1):183–202, Mar 2009.
- [23] X. Yuan and J. Yang, “Sparse and low-rank matrix decomposition via alternating direction method,” *Pacific J. Optim.*, vol.9, no.1, pp 167–180, 2013.
- [24] 이형일, 이혁, 이정우, “부공간에의 사영에 기반한 RPCA 기법의 온라인 업데이트 기법”, *한국통신학회 추계종합학술발표회*, 2014

약 력



이 혁

2006년~2011년 포항공과대학교 전자전기공학 학사  
 2011년~2013년 서울대학교 전기공학부 석사  
 2013년~현재 서울대학교 전기공학부 박사과정  
 관심분야: MIMO, Compressive Sensing, Distributed Storage



이 형 일

2008년~2012년 서울대학교 전기공학 학사  
 2012년~현재 서울대학교 전기공학부 석박통합  
 관심분야: Compressive Sensing, Distributed Storage



조 재 학

2009년~2013년 포항공과대학교 전자전기공학 학사  
 2014년~현재 서울대학교 전기공학부 대학원 석,박사 통합과정  
 관심분야: Compressive Sensing



김 민 철

2014년 서울대학교 공학사  
 관심분야: 머신 러닝, 압축 센싱



소 병 현

2014년 포항공과대학교 전자전기공학 학사  
 2014년~현재 서울대학교 대학원 전기공학 석박통합과정  
 관심분야: MIMO, Compressive Sensing

약 력



이 정 우

1984년~1988년 서울대학교 전기공학 학사  
 1988년~1990년 Princeton University 전기공학 석사  
 1990년~1994년 Princeton University 전기공학 박사  
 1994년~1999년 Sarnoff Corporation (Member of Technical Staff)  
 1999년~2002년 Lucent Technologies (Bell Labs) Wireless Advanced Technology Lab (Member of Technical Staff)  
 2002년~현재 서울대학교 전기정보공학부 교수  
 관심분야: 무선통신, 신호처리