Fast Algorithms for Data Recovery

Erin Tripp

Mathematics Department
Syracuse University

February 3, 2017

Table of Contents

We look at three variations of an algorithm for recovering data from corrupted observations and some related work. All variations are based on modifying one classical optimization method: gradient descent.

We also look at how we can extend these methods to more general problems and what performance guarantees we might see.

Table of Contents

Problem

From observed data $Y \in \mathbb{R}^{d_1 \times d_2}$ where $Y = M + S$, we want to recover the low rank matrix M. When does this sort of problem come up?

Problem

From observed data $Y \in \mathbb{R}^{d_1 \times d_2}$ where $Y = M + S$, we want to recover the low rank matrix M. When does this sort of problem come up?

- Foreground background separation:
  Y is video data encoded into a matrix of pixel values
  M is the stable background
  S is the foreground of moving objects

Problem

From observed data $Y \in \mathbb{R}^{d_1 \times d_2}$ where $Y = M + S$, we want to recover the low rank matrix M. When does this sort of problem come up?

- Foreground background separation:
  Y is video data encoded into a matrix of pixel values
  M is the stable background
  S is the foreground of moving objects
- Principal component analysis:
  Y is the observed data
  M is the "true data"
  S is some sort of corruption or noise

Problem

We formulate the problem of recovering M as a minimization problem:

$$\min_{U,V,S} \|UV^T + S - Y\|_F^2 + \|U^T U - V^T V\|_F^2 \quad (1)$$

where $UV^T = M$.

Next, we make some assumptions on M and S:

Problem

We formulate the problem of recovering M as a minimization problem:

$$\min_{U,V,S} \|UV^T + S - Y\|_F^2 + \|U^T U - V^T V\|_F^2 \tag{1}$$

where $UV^T = M$.

Next, we make some assumptions on M and S:

- M is not near sparse

  M is μ-incoherent, i.e. we place a bound on the size of its singular vectors.

  The more spread out the principal components of M are, the more we need to know about it to reconstruct it.

Problem

We formulate the problem of recovering M as a minimization problem:

$$\min_{U,V,S} \|UV^T + S - Y\|_F^2 + \|U^TU - V^TV\|_F^2 \tag{1}$$

where $UV^T = M$.

Next, we make some assumptions on M and S:

- M is not near sparse

  M is μ-incoherent, i.e. we place a bound on the size of its singular vectors.
  The more spread out the principal components of M are, the more we need to know about it to reconstruct it.

- S is sufficiently sparse

  S will have at most α-fraction nonzero entries.
  For example, if $\alpha = 0.1$, only 10% of the entries of S will be nonzero.

Algorithm

In 2016, a team of engineers from the University of Texas at Austin[1] and Cornell University[2] introduced fast algorithms[1] to solve this optimization problem in both the fully observed and partially observed cases.

Sketch of Algorithm:

1. Produce an initial sparse estimate S_0 from Y
2. Compute the rank r SVD of $Y - S = [L, \Sigma, R]$
3. Set $U_0 = L\Sigma^{1/2}$, $V_0 = R\Sigma^{1/2}$
4. Proceed by projected gradient descent on the factors U_t, V_t

[1] Xinyang Yi, Dohyung Park, and Constantine Caramanis
[2] Yudong Chen

Algorithm 1 RPCA via Gradient Descent

Input: observed matrix Y, target rank r, sparsity fraction α, parameters γ and η, number of iterations T

procedure RPCA_GD(Y, r, α, params)

$S_{\text{init}} = T_\alpha(Y)$

$[L, \Sigma, R] = \text{SVD}_r(Y - S_{\text{init}})$

$U_0 = L\Sigma^{\frac{1}{2}},\ V_0 = R\Sigma^{\frac{1}{2}}$

$U_0 = \Pi_{\mathcal{U}}(U_0),\ V_0 = \Pi_{\mathcal{V}}(V_0)$

for t = 0:T-1 **do**

$S_t = T_{\gamma\alpha}(Y - U_t V_t^T)$

$U_{t+1} = \Pi_{\mathcal{U}}(U_t - \eta\nabla_U \mathcal{L}(U_t, V_t, S_t) - \frac{1}{2}\eta U_t(U_t^T U_t - V_t^T V_t))$

$V_{t+1} = \Pi_{\mathcal{V}}(V_t - \eta\nabla_V \mathcal{L}(U_t, V_t, S_t) - \frac{1}{2}\eta V_t(V_t^T V_t - U_t^T U_t))$

end for

end procedure

Output: (U_T, V_T)

Our Contribution

As presented, RPCA GD satisfies the following convergence bound:

$$d^2(U_t, V_t; U^*, V^*) \leq (1 - \frac{\sigma_r^* \eta}{8})^t d^2(U_0, V_0; U^*, V^*) \tag{2}$$

where $d^2(U_t, V_t; U^*, V^*)$ is the distance from the t-th iterate to the optimal solution.

How to choose η?

Our Contribution

As presented, RPCA GD satisfies the following convergence bound:

$$d^2(U_t, V_t; U^*, V^*) \leq (1 - \frac{\sigma_r^* \eta}{8})^t d^2(U_0, V_0; U^*, V^*) \tag{2}$$

where $d^2(U_t, V_t; U^*, V^*)$ is the distance from the t-th iterate to the optimal solution.

How to choose η?

- To guarantee the above convergence, $\eta \leq \frac{1}{36\sigma_1^*}$

Our Contribution

As presented, RPCA GD satisfies the following convergence bound:

$$d^2(U_t, V_t; U^*, V^*) \leq \left(1 - \frac{\sigma_r^* \eta}{8}\right)^t d^2(U_0, V_0; U^*, V^*) \tag{2}$$

where $d^2(U_t, V_t; U^*, V^*)$ is the distance from the t-th iterate to the optimal solution.

How to choose η?

- To guarantee the above convergence, $\eta \leq \frac{1}{36\sigma_1^*}$
- This step size is not related to the Lipschitz constant of the gradient, and as long as it meets that bound, the choice is somewhat arbitrary.

Our Contribution

As presented, RPCA GD satisfies the following convergence bound:

$$d^2(U_t, V_t; U^*, V^*) \leq (1 - \frac{\sigma_r^* \eta}{8})^t d^2(U_0, V_0; U^*, V^*) \tag{2}$$

where $d^2(U_t, V_t; U^*, V^*)$ is the distance from the t-th iterate to the optimal solution.

How to choose η?

- To guarantee the above convergence, $\eta \leq \frac{1}{36\sigma_1^*}$
- This step size is not related to the Lipschitz constant of the gradient, and as long as it meets that bound, the choice is somewhat arbitrary.
- Algorithm 1 uses a single η for every iteration, but a larger step size might be feasible for some.

Our Contribution

We introduce a backtracking algorithm to choose the largest step size which results in an objective decrease.

1. Start with $\hat{\eta}$
2. Compute the gradient descent step with this step size to get U_t^+, V_t^+, S_t^+
3. If $\mathcal{F}(U_t^+, V_t^+, S_t^+) - \mathcal{F}(U_t, V_t, S_t) < m$, accept $U_t^+ = U_{t+1}$, $V_t^+ = V_{t+1}$, $S_t^+ = S_{t+1}$.
4. Else, $\hat{\eta} = \tau\hat{\eta}$

   In practice, we used $\tau = \frac{1}{2}$ and $m = 0$.

Application to Foreground-Background Separation


(Original frame)

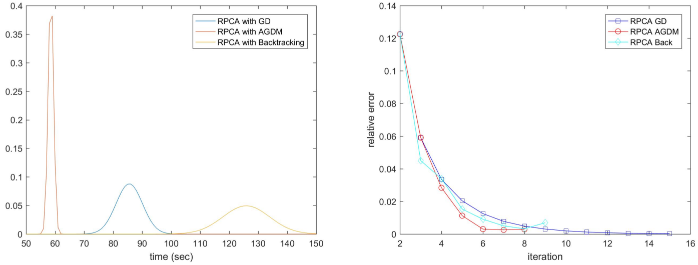

(GD, 30 iterations, 14.84 sec)


(Backtracking, 5 iterations, 5.73 sec)

Table of Contents

Linearly Coupled Gradient and Mirror Descent

We also implemented a version of Algorithm 1 with coupled gradient and mirror descent (AGDM) [2]. This algorithm performed better than either the original RPCA GD or the backtracking algorithm on large, random matrices.

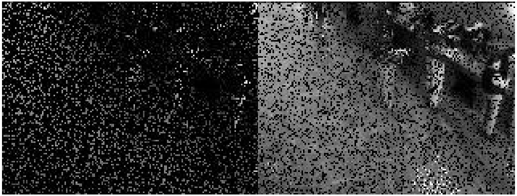


We are currently working on extending the AGDM algorithm to general functions which are not Lipschitz smooth.

Nonzero Median Filter

The erasures in the partially observed footage manifest as zero entries in the matrix Y. Traditional median filtering does not improve matrices with very few observed entries as the median is almost always zero.

We developed a median filter which works around these zeros and improve the performance of all variations of the RPCA algorithm on partially observed matrices.



A partially observed frame before and after filtering.

Nonconvex optimization on Manifolds

Another area of interest is optimization on Riemannian manifolds. These arise naturally in many problems:

- The Netflix problem can be formulated as a search for two low rank matrices U and W such that the ratings matrix $X = UW$. Here, the space of fixed rank matrices is a manifold.
- Sychronization of rotations: we scan a 3D object from several angles and want to piece together these scans to get a complete representation of the image. We need to find the correct rotations and translations based on the observed data. The set of rotation matrices forms a differentiable manifold.

There is recent work extending methods of classical nonconvex optimization to this setting, which will allow us to develop algorithms for a wider scope of problems.

References

- Xinyang Yi, Dohyung Park, Yudong Chen, and Constantine Caramanis. *Fast Algorithms for Robust PCA via Gradient Descent*. Preprint, *arXiv: 1605.07784v1 [cs.IT]* (2016).

- Zeyuan Allen-Zhu and Lorenzo Orecchia. *Linear Coupling: An Ultimate Unification of Gradient and Mirror Descent*. Preprint, *arXiv: 1407.1537v4 [cs.DS]* (2015).

- Heinz H. Bauschke, Jérôme Bolte, and Marc Teboulle. *A descent Lemma beyond Lipschitz Gradient continuity: first-order methods revisited and applications* Accepted, Mathematics of Operations Research (2016).

- Nicolas Boumal, P.-A. Absil, and Coralia Cartis. *Global rates of convergence for nonconvex optimization on manifolds*. Preprint, arXiv: `https://arxiv.org/abs/1605.08101` [math.OC] (2016).

Thank you!