Suggested Course Project: Low-Rank Matrix Completion

Consider this optimization problem, which seeks to recover a low-rank approximation to data in the matrix B.

$$\underset{X\in\mathbb{R}^{m\times n}}{\text{minimize}} \quad \tfrac{1}{2}\|\Omega\circ X - B\|_F^2 \quad \text{subject to} \quad \|X\|_* \le \tau. \tag{1}$$

This problem appears in recommender systems [1], where the (i,j)th element of the sparse matrix B records the rating score given by user i for product j. Ratings are observed only for a subset of user-product pairs indexed by the binary mask

$$\Omega_{ij} = \begin{cases} 1 & \text{if user } i \text{ has rated product } j; \\ 0 & \text{otherwise.} \end{cases}$$

The goal is to predict the unseen ratings, captured in the dense unknown matrix X. A structural low-rank assumption is used to capture an observed "archetype" phenomenon—users who often like the same movies serve as good predictors for each other, and movies that are liked by the same users probably are also similar. Therefore, we consider each user as a sparse linear combination of archetypal individuals (and similarly with products), where the inner product of their feature vectors give the same prediction rating. The nuclear-norm constraint on X is a common approach for encouraging low-rank solutions [3].

In this project you might explore two different algorithmic variations that might be used to build an implementation of the conditional gradient method efficient for large problems. Try it on real or synthetic data. You'll need to discover a value of τ that works for your problem, i.e., what is the solution rank you're after.

Here is a statement of the conditional gradient method for minimizing a smooth function f over the convex hull of the set $\tau\mathcal{A}\subset\mathbb{R}^n$.

> **Input:** $x^{(0)}\in\tau\mathcal{A}$, $\epsilon>0$
> **for** $k=0,1,2,\dots$ **do**
> $z_k = -\nabla f(x_k)$
> $a_k\in\tau\mathcal{F}_{\mathcal{A}}(z_k)$ **if** $\langle a_k - x_k, z_k\rangle < \epsilon$ **then** break
> $x_{k+1} = \theta_k a_k + (1-\theta_k)x_k, \quad \theta_k\in(0,1)$
> **return** x_k

For the nuclear norm, $\mathcal{A} = \{\, uv^T \mid \|u\| = \|v\| = 1 \,\}$.

1 Factored updates

The iterates are now matrices, and so we write them with upper-case letters. Implement the algorithm so that the iterate X_k is tracked in factored form $X_k = U_k\Sigma_k V_k^T$. Because the exposed atom A_k (second line of the loop) takes the form $A_k = \tau u_k v_k^T$, we can rewrite the update to X_k (last line in the loop) as

$$\begin{aligned} X_k &= \theta_k u_k v_k^T + (1-\theta_k) U_k \Sigma_k V_k^T \\ &= \begin{bmatrix} U_k & u_k \end{bmatrix} \begin{bmatrix} (1-\theta_k)\Sigma_k & \\ & \theta_k \tau \end{bmatrix} \begin{bmatrix} V_k & v_k \end{bmatrix}^T \\ &= U_{k+1} \Sigma_{k+1} V_{k+1}^T. \end{aligned}$$

Thus, U_k, Σ_k, and V_k grow at each iteration. This means that in practice you'll need to "compress" the factors so that they don't grow too large. This could cause some instability in the method.

2 Dual version with primal recovery

Implement the dual conditional gradient method in Fan et al. [2, Algorithm 6.3]. The last line of that algorithm, which recovers the primal iterate, should be solved as an unconstrained linear least squares problem. There is an opportunity at that point to select only a subset of the exposed singular vectors so that you can produce a low-rank approximation to the solution.

References

[1] R. M. Bell and Y. Koren, *Lessons from the Netflix prize challenge*, ACM SIGKDD Explorations Newsletter, 9 (2007), pp. 75–79.

[2] Z. Fan, H. Jeong, Y. Sun, and M. P. Friedlander, *Polar alignment and atomic decomposition*, 2019. To appear in Foundations and Trends in Optimization. See preprint.

[3] B. Recht, M. Fazel, and P. A. Parrilo, *Guaranteed minimum-rank solutions of linear matrix equations via nuclear norm minimization*, 52 (2010), pp. 471–501.