Applications of block coordinate update method

nonnegative matrix factorization and robust PCA

(Due in class April-6-2018)

1 Nonnegative matrix factorization

The nonnegative matrix factorization (NMF) aims to find two nonnegative matrices $W \in \mathbb{R}^{m\times r}$ and $H \in \mathbb{R}^{n\times r}$ such that their product $WH^\top$ approximates a given nonnegative matrix $X \in \mathbb{R}^{m\times n}$. Under the assumption of Gaussian noise, it can be modeled as

$$\underset{W,H}{\text{minimize}}\,\frac{1}{2}\|WH^\top - X\|_F^2 + \phi_1(W) + \phi_2(H), \text{ s.t. } W \in \mathbb{R}_+^{m\times r}, H \in \mathbb{R}_+^{n\times r}, \tag{1}$$

where ϕ_1 and ϕ_2 are regularization terms to encourage certain structures of the solution. Because of nonnegativity, NMF can learn local features and has better interpretability than the principle component analysis (PCA), which often learns global features.

2 Robust principal component analysis

Let X be composed of a sparse matrix S and a low-rank matrix L. The robust PCA aims at finding S and L, given X. Using ℓ_1 norm to promote sparsity and nuclear norm to promote low-rankness, robust PCA can be modeled as

$$\min_{L,S} \|L\|_* + \lambda\|S\|_1, \text{ s.t. } L + S = X. \tag{2}$$

Here, $\|L\|_*$ denotes the nuclear norm of L and equals the summation of its singular values, and $\|S\|_1 = \sum_{i,j} |s_{ij}|$. Directly applying alternating minimization (AltMin) to (2) will not work because fixing either L or S, the other one is fully determined from the constraint. To use AltMin, one can solve the penalized problem

$$\min_{L,S} \|L\|_* + \lambda\|S\|_1 + \frac{\beta}{2}\|L + S - X\|_F^2, \tag{3}$$

or a sequence of the above problem, where $\beta > 0$ is the penalty parameter.

3 Requirements

Include every item below in a single report and attach your code. Use the provided datasets to test your code.

1. Let $\mathcal{W}$ be the set consisting all matrices W such that $\|W_j\|_2 \leq 1$ for $j = 1, \ldots, r$, where W_j is the j-th column of W. Let $\phi_1(W)$ be the indicator function of $\mathcal{W}$ and $\phi_2(H) = 0$. Then the problem (1) is equivalent to

$$\underset{W,H}{\text{minimize}} \frac{1}{2}\|WH^\top - X\|_F^2, \text{ s.t. } W \in \mathbb{R}_+^{m\times r}, H \in \mathbb{R}_+^{n\times r}, \|W_j\|_2 \leq 1, j = 1, \ldots, r. \quad (4)$$

   Develop three solvers for (4): one is by the projected gradient, another is by the alternating minimization, and the third one by the alternating proximal gradient. For the alternating minimization, you will need to write a subroutine, which could be accelerated proximal gradient or a certain existing solver.

2. Compare the three solvers by using the provided Swimmer dataset. Note that the dataset is in 3D format. It contains 256 images of size 32×32. You will need first reshape each image into a column vector and form a 1024×256 matrix X. Report how these solvers decrease the objective values in terms of iteration number of also actual running time. For alternating minimization and alternating proximal gradient method, one iteration is counted as updating both W and H once. Set $r = 17$. Reshape each column of W into a 32×32 image and use `imshow` to show all the columns of W. Report what you observe. [Hint: columns of W should contain local parts of the images, such as limbs in different positions.]

3. Develop two solvers for (2): one is by the proximal gradient, and another is by the alternating minimization.

4. Compare the two solvers by using the provided Escalator video Dataset. Note that the dataset is in 3D format. It contains 200 frames of size 130×160. You will need first reshape each frame into a column vector and form a 20800×200 matrix X. Report how these solvers decrease the objective values in terms of iteration number of also actual running time. Reshape each column of L and S into a 130×160 image and use `imshow` to show a few selected columns of L and S. You can also use `implay` to see how the foreground and background of the video are separated. Report what you observe.