Problem outline

June 29, 2021

Original NMF factorization:

$$D = XY \tag{1}$$

Diffusion Kernel (Regularized Laplacian):

$$K = (I + \beta L)^{-1} \tag{2}$$

Now assume that Y can be approximated by a diffusion process that began at a number of initiator locations. Let V be the sparse k x n matrix of initiators.

$$Y = VK = V(I + \beta L)^{-1} \tag{3}$$

And:

$$D = XY = XVK \tag{4}$$

Now, our problem becomes the minimization of the following:

$$||D - XVK||_F^2 \tag{5}$$

Constrained by the fact that X must be non-negative and V is non negative and sparse. K is assumed to be known from prior calculation given the graph Laplacian and the parameter β which is measures the extent of diffusion.

For a standard gradient descent update:

$$X \leftarrow X - \eta_X \cdot \nabla_X F(D, XVK) \tag{6}$$

$$V \leftarrow V - \eta_V \cdot \nabla_V F(D, XVK) \tag{7}$$

Where $\nabla F(\theta)$ is the gradient of the cost function F :

$$F(D, XVK) = ||D - XVK||_F^2 = tr[(D - XVK)^T (D - XVK)] \tag{8}$$

And the third part of the equality holds by the fact that:

$$tr(X^T Y) = \sum_{i=1}^{M} \sum_{j=1}^{N} x_{ij} y_{ij} \tag{9}$$

$$||X||_F = \sqrt{\sum_{i=1}^{M} \sum_{j=1}^{N} x_{ij}^2} = \sqrt{tr(X^T X)} \tag{10}$$

Simplify and expand F to get:

$$F(D, XVK) = tr[(D^T - K^T V^T X^T)(D - XVK)] \tag{11}$$

$$F(D, XVK) = tr[D^T D - D^T XVK - K^T V^T X^T D + K^T V^T X^T XVK] \tag{12}$$

To compute the gradients of F need to use the following properties:

1. Trace of a sum
$$tr(A + B) = tr(A) + tr(B) \tag{13}$$
2. Cyclic permutation
$$tr(ABCD) = tr(DABC) = tr(CDAB) = tr(BCDA) \tag{14}$$
3. Gradient traces of a product with constant matrix A:
$$\nabla_X tr(AX) = A^T \tag{15}$$
$$\nabla_X tr(X^T A) = A \tag{16}$$
$$\nabla_X tr(X^T AX) = (A + A^T)X \tag{17}$$
$$\nabla_X tr(XAX^T) = X(A^T + A) \tag{18}$$

Starting with ∇_X and using property 1:

$$\nabla_X F(D, XVK) = \nabla_X(tr(D^T D) - tr(D^T XVK) - tr(K^T V^T X^T D) + tr(K^T V^T X^T XVK)) \tag{19}$$

For the first term of this equation:

$$\nabla_X tr(D^T D) = 0 \tag{20}$$

The Second and third terms are simplified by the cyclic property and gradient trace properties (15) and (16) respectively:

$$\nabla_X tr(D^T XVK) = \nabla_X tr(VKD^T X) = (VKD^T)^T = DK^T V^T \tag{21}$$

$$\nabla_X tr(K^T V^T X^T D) = \nabla_X tr(X^T DK^T V^T) = DK^T V^T \tag{22}$$

The fourth term can be simplified with cyclic property and by using (18) with $A = VKK^T V^T$:

$$\nabla_X tr(K^T V^T X^T XVK) = \nabla_X tr(XVKK^T V^T X^T) \tag{23}$$

$$= X((VKK^T V^T)^T + (VKK^T V^T)) = 2XVKK^T V^T \tag{24}$$

So putting this all together:

$$\nabla_X F(D, XVK) = -2DK^T V^T + 2XVKK^T V^T \tag{25}$$

Likewise for ∇_V:

$$\nabla_V F(D, XVK) = \nabla_V(tr(D^T D) - tr(D^T XVK) - tr(K^T V^T X^T D) + tr(K^T V^T X^T XVK)) \tag{26}$$

First term:

$$\nabla_V tr(D^T D) = 0 \tag{27}$$

Second and third terms simplify by cyclic and (15), (16) respectively:

$$\nabla_V tr(D^T XVK) = \nabla_V tr(KD^T XV) = (KD^T X)^T = X^T DK^T \tag{28}$$

$$\nabla_V tr(K^T V^T X^T D) = \nabla_V tr(V^T X^T DK^T) = X^T DK^T \tag{29}$$

The fourth term is slightly harder to compute. To do so I will borrow another property which is proven here: `https://web.stanford.edu/~jduchi/projects/matrix_prop.pdf`

$$\nabla_A tr(ABA^T C) = CAB + C^T AB^T \tag{30}$$

Using $A = V, B = KK^T, C = X^T X$ then the fourth term is easily simplified:

$$\nabla_V tr(K^T V^T X^T XVK) = \nabla_V tr(VKK^T V^T X^T X) \tag{31}$$

$$= (X^T X)V(KK^T) + (X^T X)V(KK^T) = 2X^T XVKK^T \tag{32}$$

So finally:

$$\nabla_V F(D, XVK) = -2X^T DK^T + 2X^T XVKK^T \tag{33}$$

Updated gradient descent equations:

$$X \leftarrow X + \eta_X \cdot (DK^T V^T - XVKK^T V^T) \tag{34}$$

$$V \leftarrow V + \eta_V \cdot (X^T DK^T - X^T XVKK^T) \tag{35}$$

The question is what to do with η_X and η_V. In the original NMF algorithm they use a factor designed to cancel out any negative part of the gradient. This can be done for the update of X:

$$\eta_X = \frac{X}{XVKK^T V^T} \tag{36}$$

Resulting in the following multiplicative update step:

$$X \leftarrow X + \frac{X}{XVKK^T V^T} \cdot (DK^T V^T - XVKK^T V^T) \tag{37}$$

$$X \leftarrow X + X \cdot \frac{DK^T V^T}{XVKK^T V^T} - X \cdot \frac{XVKK^T V^T}{XVKK^T V^T} \tag{38}$$

$$X \leftarrow X \cdot \frac{DK^T V^T}{XVKK^T V^T} \tag{39}$$

The update step for V could follow the same process. But for our problem we'd also like V to be sparse. In the Sparse Non Negative coding paper, an adaptive η_V (learning rate or step size) is used in the algorithm to ensure that the results decrease the cost function. The updated matrix is then projected to its closest sparse counterpart.