Fast Coordinate Descent methods for Non-Negative Matrix Factorization

Inderjit S. Dhillon
University of Texas at Austin

SIAM Conference on Applied Linear Algebra
Valencia, Spain
June 19, 2012
Joint work with Cho-Jui Hsieh

Outline

1. Non-negative Matrix Factorization
   - Non-negative Matrix Factorization (NMF)
   - Greedy Coordinate Descent (GCD) for least squares NMF
   - NMF with KL-divergence
   - Non-negative Tensor Factorization (NTF)

Outline

Problem Definition

- Input: Given a non-negative matrix $V \in \mathbb{R}^{m\times n}$ and the target rank k
- Output: two **non-negative** matrices $W \in \mathbb{R}^{m\times k}$ and $H \in \mathbb{R}^{n\times k}$,

  such that WH^T is a good approximation to V.
- Usually $m, n \gg k$.
- How to measure goodness of approximation? Two widely used choices:
  - Least squares NMF:

$$\min_{W,H\geq 0} f(W,H) \equiv \|V - WH^T\|_F^2 = \sum_{i,j}(V_{ij} - (WH^T)_{ij})^2$$

  - KL-divergence NMF:

$$\min_{W,H\geq 0} L(W,H) \equiv \sum_{i,j} V_{ij}\log(V_{ij}/(WH^T)_{ij}) - V_{ij} + (WH^T)_{ij}$$

Problem Definition (Cont'd)

- Applications: text mining, image processing, ...
- Can get more interpretable basis than SVD.
- To achieve better sparsity, researchers have proposed adding L1 regularization terms on W and H:

$$(W, H) = \arg \min_{W,H \geq 0} \left\{ \frac{1}{2} \| V - WH^T \|_F^2 + \rho_1 \sum_{i,r} W_{ir} + \rho_2 \sum_{j,r} H_{jr} \right\}$$

Existing Optimization Methods

- NMF is **nonconvex**, but is convex when W or H is fixed.
- Recent methods follow the alternating minimization framework:

  Iteratively solve $\min_{W \geq 0} f(W, H)$ and $\min_{H \geq 0} f(W, H)$ until convergence.
- For least squares NMF, each sub-problem can be exactly or approximately solved by
  1. Multiplicative rule (Lee and Seung, 2001)
  2. Projected gradient method (Lin, 2007)
  3. Newton type updates (Kim, Sra and Dhillon, 2007)
  4. Active set method (Kim and Park, 2008)
  5. **Cyclic coordinate descent method** (Chichocki and Phan, 2009)

Coordinate Descent Method

- Update **one variable** at a time until convergence: $(W, H) \leftarrow (W + sE_{ir}, H)$.
- Get s by solving a one-variable problem:

$$\min_{s: W_{ir}+s \geq 0} g_{ir}^{W}(s) \equiv f(W + sE_{ir}, H).$$

- For square loss, g_{ir}^{W} has a closed form solution:

$$\begin{aligned} s^* &= \max\left(0, W_{ir} - g'_{ir}(0)/g''_{ir}(0)\right) - W_{ir}, \\ \text{where } g'_{ir}(0) &= \nabla_{W_{ir}} f(W, H) = (WH^T H - VH)_{ir}, \\ g''_{ir}(0) &= \nabla^2_{W_{ir}} f(W, H) = (H^T H)_{rr}. \end{aligned}$$

Outline

Cyclic Coordinate Descent for Least Squares NMF (FastHals)

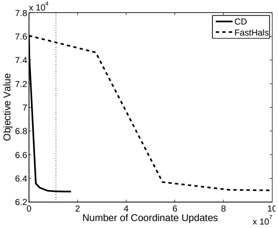
- Recently, (Chichocki and Phan, 2009) proposed a cyclic coordinate descent algorithm (FastHals) for least squares NMF.
- Fixed update sequence:

$$W_{11,}, W_{1,2}, \ldots, W_{1,k}, W_{2,1}, \ldots, W_{m,k}, \ldots, H_{1,1}, \ldots, H_{n,k}, W_{1,1}, \ldots$$

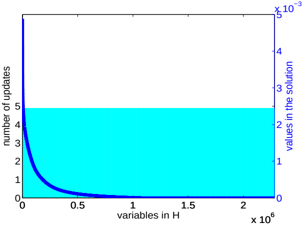
- Each update has time complexity $O(k)$.

Variable Selection

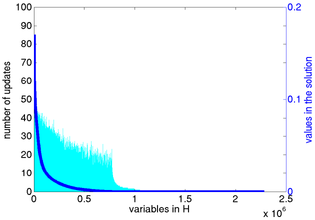
- FastHals updates variables uniformly.
- However, an efficient algorithm should update variables with frequency proportional to their **“importance”**!
- We propose a Greedy Coordinate Descent method (GCD) for NMF.



updates vs obj



The behavior of FastHals



The behavior of GCD

Greedy Coordinate Descent (GCD)

- Stategy — select variables which maximally reduce objective function
- When W_{ir} is selected, the objective function can be reduced by

$$D^W_{ir} \equiv f(W,H) - f(W + s^* E_{ir}, H) = -G^W_{ir} s^* - \frac{1}{2}(H^T H)_{rr}(s^*)^2,$$

  where $G^W \equiv \nabla_W f(W,H) = WH^T H - VH$,

  and s^* is the optimal step size.

- If D^W can be easily maintained, we can choose variables with the largest objective function value reduction according to D^W.

How to maintain D^W (objective value reduction)

- s^* can be computed from G^W and $H^T H$ (from one-variable update rule).
- $D^W_{ir} = -G^W s^* - \frac{1}{2}(H^T H)_{rr}(s^*)^2$, where $G^W = WH^T H - VH$.
- Therefore, we can maintain D^W if G^W and $H^T H$ are known.
- When $W_{ir} \leftarrow W_{ir} + s^*$, the ith row of G^W is changed:

$$G^W_{ij} \leftarrow G^W_{ij} + s^*(H^T H)_{rj} \quad \forall j = 1, \ldots, k.$$

- Therefore, time for maintaining D^W is only $O(k)$, which has the same time complexity as Cyclic Coordinate Descent!

Greedy Coordinate Descent (GCD)

- Follow the alternating minimization framework, our algorithm GCD alternatively updates variables in W and H.
- When updating one variables in W, we can maintain D^W in $O(k)$ time.
- We conduct a sequence of updates on W: $W^{(0)}, W^{(1)}, \ldots$ with a corresponding sequence $(D^W)^{(0)}, (D^W)^{(1)}, \ldots$
- When to switch from W's updates to H's updates? We update variables in W until the maximum function value decrease is small enough.

$$\max_j D_{ij}^W < \epsilon p^{\text{init}}, \text{ where } p^{\text{init}} = (D^W)^{(0)}$$

Greedy Coordinate Descent (GCD)

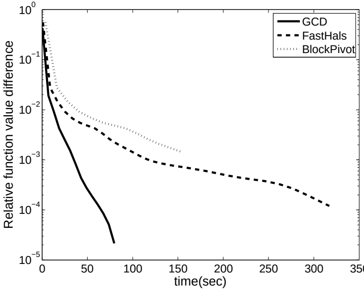
- Initialize $H^T H, W^T W$.
- While (not converged)
  1. Compute $G^W = W(H^T H) - VH$.
  2. Compute D^W according to G^W.
  3. Compute $p^{\text{init}} = \max_{i,r}(D^W_{ir})$.
  4. For each row i of W
     - $q_i = \arg\max_r D^W_{i,r}$
     - While $D^W_{i,q_i} > \epsilon p^{\text{init}}$
       4.1 Update W_{i,q_i}.
       4.2 Update $W^T W$ and D^W
       4.3 $q_i \leftarrow \arg\max_r D^W_{ir}$
  5. For updates to H, repeat steps analogous to Steps 1 through 4.

Comparisons

dataset	m	n	k	relative error	Time (in seconds)			
					GCD	FHals	PGrad	BPivot
Synth03	500	1,000	10	10^{-4}	**0.6**	2.3	2.1	1.7
			30	10^{-4}	**4.0**	9.3	26.6	12.4
Synth08	500	1,000	10	10^{-4}	**0.21**	0.43	0.53	0.56
			30	10^{-4}	**0.43**	0.77	2.54	2.86
CBCL	361	2,429	49	0.0410	**2.3**	4.0	13.5	10.6
				0.0376	**8.9**	18.0	45.6	30.9
				0.0373	**14.6**	29.0	84.6	51.5
ORL	10,304	400	25	0.0365	**1.8**	6.5	9.0	7.4
				0.0335	**14.1**	30.3	98.6	33.9
				0.0332	**33.0**	63.3	256.8	76.5

Comparisons

Results on MNIST ($m = 780, n = 60000, \#\ \text{nz} = 8994156, k = 10$).



Outline

KL-NMF

- KL-NMF:

$$\min_{W,H\geq 0} L(W,H) \equiv \sum_{i,j} V_{ij}\log\left(\frac{V_{ij}}{(WH^T)_{ij}}\right) - V_{ij} + (WH^T)_{ij}$$

- The one variable sub-problem:

$$D_{ir}(s) = L(W+sE_{ir},H) = \sum_{j=1}^{I} -V_{ij}\log\Big((WH^T)_{ij} + sH_{rj}\Big) + sH_{jr} + \text{constant}$$

- One variable update does not have closed form solution

Newton Update for One Variable Subproblem

- We use **Newton method** to solve each one-variable sub-problem
- When updating W_{ir}, iteratively update s by Newton direction:

$$s \leftarrow \max(-W_{ir}, s - h'_{ir}(s)/h''_{ir}(s)),$$

where

$$h'_{ir}(s) = \sum_{j=1}^{n} H_{jr}\left(1 - \frac{V_{ij}}{(WH^T)_{ij} + sH_{jr}}\right).$$

$$h''_{ir}(s) = \sum_{j=1}^{n} \frac{V_{ij}H_{jr}^2}{((WH^T)_{ij} + sH_{jr})^2}.$$

- Can show that Newton method **without line search** converges to optimum for this one-variable sub-problem.

CCD for KL-divergence

- Compute WH^T
- While (not converged)
  1. For all (i, r) pairs

     While (not converged)

     - $s \leftarrow \max(-W_{ir}, s - h'_{ir}(s)/h''_{ir}(s))$
     - $W_{ir} \leftarrow W_{ir} + s$
     - Maintain $(WH^T)_{i,:}$ by $(WH^T)_{i,:} \leftarrow (WH^T)_{i,:} + sH^T_{:,r}$.
  2. For updates to H, repeats steps analogous to Step 1.

Experimental Results

Time comparison results for KL divergence. $*$ indicates the specified objective value is not achievable.

dataset	k	relative error	Time (in seconds)	
			CCD	Multiplicative
Synth03	30	10^{-3}	**121.1**	749.5
		10^{-5}	**184.32**	7092.3
Synth08	30	10^{-2}	**22.6**	46.0
		10^{-5}	**56.8**	*
CBCL	49	0.1202	38.2	**21.2**
		0.1103	**123.2**	562.6
		0.1093	**166.0**	3266.9
ORL	25	0.3370	**73.7**	165.2
		0.3095	**253.6**	902.2
		0.3067	**370.2**	1631.9

Outline

Non-negative Tensor Factorization (NTF)

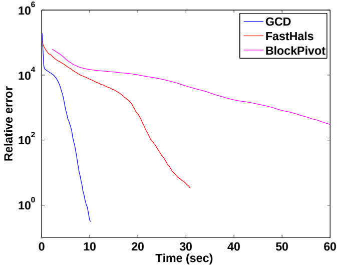
- Our method can be naturally extended to solve Non-negative Tensor Factorization (NTF).
- A tensor is a multi-dimensional matrix:
  $\underline{V} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$, where $I_1, \ldots, I_N$ are index upper bounds and N is the order (dimension) of the tensor.
- A rank-k approximation to the N-way tensor:

$$\underline{V} \approx \sum_{j=1}^{k} \mathbf{u}_1^j \otimes \mathbf{u}_2^j \otimes \cdots \otimes \mathbf{u}_N^j,$$

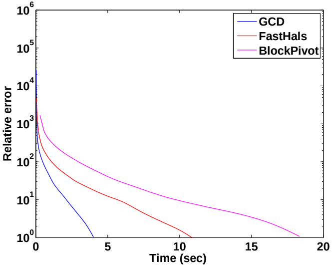
where $\otimes$ indicates outer product of vectors.

GCD for non-negative tensor factorization

- GCD can be extended to solve NTF problems.
- Similar to the NMF cases, GCD outperforms state-of-the-art algorithms.



Synthetic ($1000 \times 200 \times 100$)



CMU image data ($640 \times 128 \times 120$)

Conclusions

- We propose two algorithms: Greedy Coordinate Descent (GCD) for least squares NMF and Cyclic Coordinate Descent (CCD) for KL-NMF.
- Both algorithms outperform state-of-the-art methods.
- Code is available at `http://www.cs.utexas.edu/~cjhsieh/nmf/`