Fast Coordinate Descent Methods with Variable Selection for NMF

ChoJui Hsieh and Inderjit S. Dhillo

Published on KDD 2011

Hongchang Gao

Outline

- Definition
- Multiplicative Update Method
- Alternating Non-negative Least Squares
- Gradient Descent Method
- Fast Coordinate Descent Methods

Definition

- Given a nonnegative matrix $V \in R^{m \times n}$, find nonnegative matrices $W \in R^{m \times k}$ and $H \in R^{k \times n}$ to

$$\min_{W,H \geq 0} f(W,H) = \frac{1}{2} \| V - WH \|_F^2$$

  - The partial derivative r.w.t W and H

$$\frac{\partial f}{\partial W} = WHH^T - VH^T$$

$$\frac{\partial f}{\partial H} = W^T WH - W^T V$$

Outline

- Definition
- Multiplicative Update Method
- Alternating Non-negative Least Squares
- Gradient Descent Method
- Fast Coordinate Descent Methods

Multiplicative Update Method

- The most common used method
- Proposed by Lee and Seung (2001)
- The update rule:

$$W_{ia} = W_{ia} \frac{(VH^T)_{ia}}{(WHH^T)_{ia}}$$

$$H_{a\mu} = H_{a\mu} \frac{(W^T V)_{a\mu}}{(W^T WH)_{a\mu}}$$

Multiplicative Update Method

- Arise from gradient descent method

$$W_{ia} \leftarrow W_{ia} + \varepsilon_{ia}[(VH^T)_{ia} - (WHH^T)_{ia}]$$

  - Where ε_{ia} is a small positive number.
- Set it as

$$\varepsilon_{ia} = \frac{W_{ia}}{(WHH^T)_{ia}}$$

- Then

$$W_{ia} = W_{ia} \frac{(VH^T)_{ia}}{(WHH^T)_{ia}}$$

Multiplicative Update Method

- Algorithm

MULTIPLICATIVE UPDATE ALGORITHM FOR NMF

$\mathbf{W} = \text{rand}(m, k)$; % initialize $\mathbf{W}$ as random dense matrix

$\mathbf{H} = \text{rand}(k, n)$; % initialize $\mathbf{H}$ as random dense matrix

for $i = 1 : \text{maxiter}$

(MU) $\mathbf{H} = \mathbf{H} .* (\mathbf{W}^T\mathbf{A})./(\mathbf{W}^T\mathbf{W}\mathbf{H} + 10^{-9})$;

(MU) $\mathbf{W} = \mathbf{W} .* (\mathbf{A}\mathbf{H}^T)./(\mathbf{W}\mathbf{H}\mathbf{H}^T + 10^{-9})$;

end

  - The 10−9 in each update rule is added to avoid division by zero

Multiplicative Update Method

- Property 1
  - If W^{init} and H^{init} are strictly positive, these matrices remain positive throughout the iterations.
- Property 2
  - If $\{W^k, H^k\} \to \{W^*, H^*\}$ and $W^* > 0, H^* > 0$, then

$$\frac{\partial f}{\partial W}(W^*, H^*) = 0$$

$$\frac{\partial f}{\partial H}(W^*, H^*) = 0$$

Multiplicative Update Method

- Proof of Property 2
  - The update rule

$$H = H + [H./(W^T WH)].*[W^T(V - WH)]$$

  - For the limit point

$$\frac{H_{ij}}{[W^T WH]_{ij}}([W^T V]_{ij} - [W^T WH]_{ij}) = 0$$

$$\Rightarrow [W^T V]_{ij} - [W^T WH]_{ij} = 0$$

$$\Rightarrow [\frac{\partial f}{\partial H}]_{ij} = 0$$

Multiplicative Update Method

- From the two properties, KKT conditions satisfied, which means the limit point is a stationary point.

$$
\begin{aligned}
&\mathbf{W} \geqslant \mathbf{0}, \\
&\mathbf{H} \geqslant \mathbf{0}, \\
&(\mathbf{WH} - \mathbf{A})\mathbf{H}^{\mathrm{T}} \geqslant \mathbf{0}, \\
&\mathbf{W}^{\mathrm{T}}(\mathbf{WH} - \mathbf{A}) \geqslant \mathbf{0}, \\
&(\mathbf{WH} - \mathbf{A})\mathbf{H}^{\mathrm{T}} .* \mathbf{W} = \mathbf{0} \\
&\mathbf{W}^{\mathrm{T}}(\mathbf{WH} - \mathbf{A}) .* \mathbf{H} = \mathbf{0}.
\end{aligned}
$$

- Otherwise, can not determine whether it is a stationary point

Multiplicative Update Method

- Conclusion:
  - The sequence can not guarantee to converge to a stationary point
  - When converge, are slow to converge notoriously
  - The computational cost for each iteration $O(mnk)$
  - Once an element in W or H becomes 0, it must remain 0

Outline

- Definition
- Multiplicative Update Method
- Gradient Descent Method
- Alternating Non-negative Least Squares
- Fast Coordinate Descent Methods

Gradient Descent Method

- The update rule:

BASIC GRADIENT DESCENT ALGORITHM FOR NMF

$\mathbf{W} = \text{rand(m, k)};$ % initialize $\mathbf{W}$

$\mathbf{H} = \text{rand(k, n)};$ % initialize $\mathbf{H}$

for $i = 1 : \text{maxiter}$

$$\mathbf{H} = \mathbf{H} - \varepsilon_H \frac{\partial f}{\partial \mathbf{H}}$$

$$\mathbf{W} = \mathbf{W} - \varepsilon_W \frac{\partial f}{\partial \mathbf{W}}$$

end

  - The multiplicative update method can be considered as a gradient method

Gradient Descent Method

- How to choose the step $\varepsilon_H, \varepsilon_W$?
  - Initialize as 1, then multiply them by ½ at each iteration. Can not guarantee the non-negativity.
  - Project gradient method.

$$
\begin{aligned}
W^{k+1} &= \max(0, W^k - \alpha_k \nabla_W f(W^k, H^k)), \\
H^{k+1} &= \max(0, H^k - \alpha_k \nabla_H f(W^k, H^k)),
\end{aligned}
$$

Gradient Descent Method

- Main idea of Projected Gradient Method
  - Given such a problem

$$\begin{aligned} \min_{\mathbf{x}\in R^n} \quad & f(\mathbf{x}) \\ \text{subject to} \quad & l_i \le x_i \le u_i, \quad i = 1, \ldots, n, \end{aligned}$$

  - Update rule:

$$\mathbf{x}^{k+1} = P[\mathbf{x}^k - \alpha^k \nabla f(\mathbf{x}^k)],$$

$$P[x_i] = \begin{cases} x_i & \text{if } l_i < x_i < u_i, \\ u_i & \text{if } x_i \ge u_i, \\ l_i & \text{if } x_i \le l_i, \end{cases}$$

Gradient Descent Method

- Conclusion:
  - Without a careful choice for step, it is difficult to guarantee non-negativity.
  - The projection makes it difficult to analysis the convergence.
  - Sensitive to the initialization

Outline

- Definition
- Multiplicative Update Method
- Gradient Descent Method
- Alternating Non-negative Least Squares
- Fast Coordinate Descent Methods

Alternating Non-negative Least Squares

- The objective is not convex in both W and H, but it is convex in either W or H.
- Alternatively fixes one matrix and improves the other, called Block Coordinate Descent

$$\text{Find } W^{k+1} \text{ such that } f(W^{k+1}, H^k) \leq f(W^k, H^k), \text{ and}$$

$$\text{Find } H^{k+1} \text{ such that } f(W^{k+1}, H^{k+1}) \leq f(W^{k+1}, H^k).$$

Alternating Non-negative Least Squares

Algorithm 2 Alternating non-negative least squares

1. Initialize $W_{ia}^1 \geq 0, H_{bj}^1 \geq 0, \forall i, a, b, j.$
2. For $k = 1, 2, \ldots$

$$W^{k+1} = \arg\min_{W \geq 0} f(W, H^k), \tag{9}$$

$$H^{k+1} = \arg\min_{H \geq 0} f(W^{k+1}, H). \tag{10}$$

- Theorem
  - Any limit point of the sequence $\{W^k, H^k\}$ generated by Algorithm 2 is a stationary point.

Alternating Non-negative Least Squares

- Conclusion:
  - Has nice optimization properties.
  - It can be very fast if well implemented.

BASIC ALS ALGORITHM FOR NMF

$\mathbf{W} = \text{rand}(m, k)$; % initialize $\mathbf{W}$ as random dense matrix or use another initialization from Langville et al. (2006)

for i = 1: maxiter

(LS) Solve for $\mathbf{H}$ in matrix equation $\mathbf{W}^T\mathbf{W}\mathbf{H} = \mathbf{W}^T\mathbf{A}$.

(NONNEG) Set all negative elements in $\mathbf{H}$ to 0.

(LS) Solve for $\mathbf{W}$ in matrix equation $\mathbf{H}\mathbf{H}^T\mathbf{W}^T = \mathbf{H}\mathbf{A}^T$.

(NONNEG) Set all negative elements in $\mathbf{W}$ to 0.

end

Outline

- Definition
- Multiplicative Update Method
- Gradient Descent Method
- Alternating Non-negative Least Squares
- Fast Coordinate Descent Methods

Fast Coordinate Descent Method with Variable Selection

- Contribution
  - Propose a variable selection scheme
  - Guarantee the convergence
  - Propose a cyclic coordinate method solve

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

Fast Coordinate Descent Method with Variable Selection

- Coordinate Gradient Method
- Variable Selection Strategy
- CGD for KL-Divergence
- Convergence Analysis
- Experiment Result

Coordinate Descent Method

- Coordinate Descent Method
  - updates one variable at a time until convergence.
  - More efficient than ANLS
    - ANLS need find an exact solution for each sub-problem to guarantee a stationary point

$$W^{k+1} = \arg\min_{W \geq 0} f(W, H^k),$$
$$H^{k+1} = \arg\min_{H \geq 0} f(W^{k+1}, H).$$

Coordinate Descent Method

- The update rule for W

$$(W, H) \leftarrow (W + sE_{ir}, H)$$

  - Where E_{ir} is a $m \times k$ matrix with all elements zero except the (i, r) elements equals one.
- It equals to solve a one-variable subproblem:

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

Coordinate Descent Method

- Rewrite it as

$$g_{ir}^{W}(s) = \frac{1}{2}\sum_{j}(V_{ij} - (WH)_{ij} - sH_{rj})^2$$

$$= g_{ir}^{W}(0) + (g_{ir}^{W})'(0)s + \frac{1}{2}(g_{ir}^{W})''(0)s^2.$$

  - It is a one-variable quadratic function with constraint $W_{ir} + s \geq 0.$
  - Has closed form solution:

$$s^* = \max\left(0, W_{ir} - (WHH^T - VH^T)_{ir}/(HH^T)_{rr}\right) - W_{ir}$$

  - where

$$(g_{ir}^{W})'(0) = (G^W)_{ir} = (WHH^T - VH^T)_{ir}$$

$$(g_{ir}^{W})''(0) = (HH^T)_{rr}.$$

Coordinate Descent Method

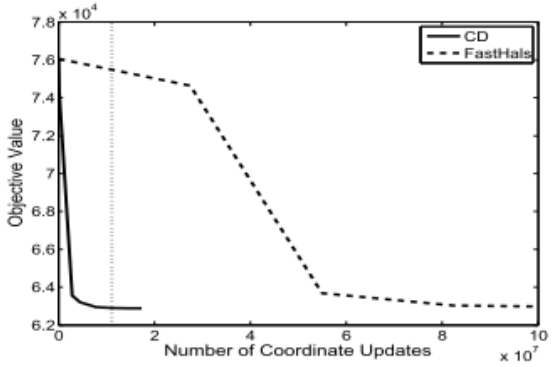
- Existing Method
  - FastHals is a coordinate descent method.
    - Use a cyclic coordinate descent method
      - It first updates all variables in W in cyclic order, and then updates variables in H.
    - May perform unneeded descent steps on unimportant variables.

Fast Coordinate Descent Method with Variable Selection

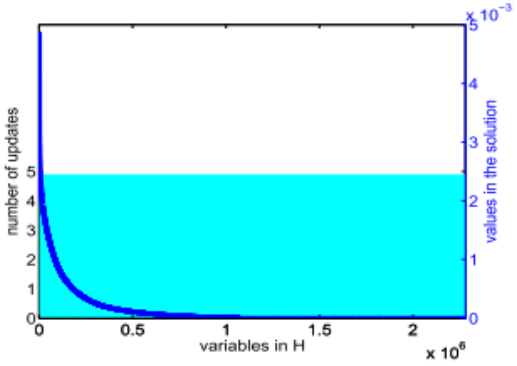
- Coordinate Gradient Method
- Variable Selection Strategy
- CGD for KL-Divergence
- Convergence Analysis
- Experiment Result

Variable Selection Strategy

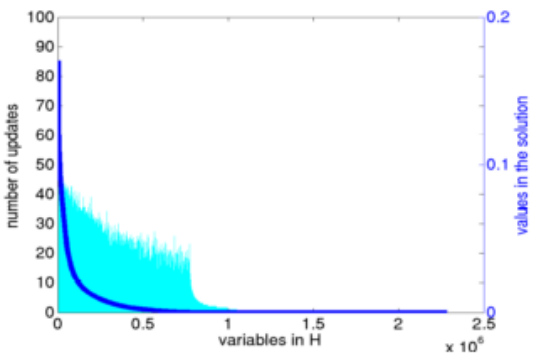
- Greedy Coordinate Descent (GCD)
  - select variables according to their importance
- Behavior Of FastHals and GCD
  - Apparently, GCD focuses on nonzero variables
  - GCD reduces the objective value more efficiently



(a) Coordinate updates versus objective value



(b) The behavior of FastHals



(c) The behavior of GCD

Variable Selection Strategy

- Update rules:
  - In the outer updates:

$$(W^0, H^0) \to (W^1, H^0) \to (W^1, H^1) \to \cdots$$

  - In the inner updates:

$$(W^i, H^i) \to (W^{i,1}, H^i) \to (W^{i,2}, H^i) \cdots$$

Variable Selection Strategy

- If W_{ir} is selected to update
  - The optimal update is

$$s^* = \max\left(0, W_{ir} - (WHH^T - VH^T)_{ir}/(HH^T)_{rr}\right) - W_{ir}$$

  - The objective will be decreased by

$$D_{ir}^W \equiv g_{ir}^W(0) - g_{ir}^W(s^*) = -G_{ir}^W s^* - \frac{1}{2}(HH^T)_{rr}(s^*)^2$$

    - Where D_{ir}^W measures how much the objective can be reduced by choosing W_{ir}
    - Thus, according to D^W choose the W_{ir} which reduce the objective value mostly

Variable Selection Strategy

- Idea
  - Maintain G^W , D^W to determine which to update
  - update them after updating each element W_{ir}
- Strategy
  - 1. Precompute G^W at the beginning of updates
  - 2. Update $W_{ir} \leftarrow W_{ir} + s^*$
  - 3. Update the i-th row of G^W and D^W in O(k) time

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

$$D^W_{ir} \equiv g^W_{ir}(0) - g^W_{ir}(s^*) = -G^W_{ir} s^* - \frac{1}{2}(HH^T)_{rr}(s^*)^2$$

Variable Selection Strategy

- Strategy
  - 4. Select the next variable-to-update to satisfy

$$(i^*, r^*) = \arg\max_{i,r} D^W_{ir}$$

    - A brute force search will cost O(mk)
    - Proposed method:
      - store the largest value and index for each row

$$q_i = \arg\max_j D^W_{ij}, \; v_i = D^W_{i,q_i}$$

      - Only one element of q will be changed after updating W_{ir}
      - Takes O(k) time to recalculate q_i
      - Takes O(logm) time to recalculate the largest value of q
      - The total cost for one update is O(k+logm)

Variable Selection Strategy

$O(k + \log m)$

$$G_{ij}^W \leftarrow G_{ij}^W + s^*(HH^T)_{rj} \quad \forall j = 1, \ldots, k \qquad O(k)$$

$$D_{ir}^W \equiv g_{ir}^W(0) - g_{ir}^W(s^*) = -G_{ir}^W s^* - \frac{1}{2}(HH^T)_{rr}(s^*)^2 \quad O(k)$$

$$q_i = \arg\max_j D_{ij}^W, \; v_i = D_{i,q_i}^W \qquad O(k)$$

change the largest value in $\{q_i \mid i = 1, \ldots, m\}$ $O(\log m)$

Variable Selection Strategy

- Note that

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

$$G^H \equiv \nabla_H f(W,H) = W^TWH - W^TV.$$

  - Maintain G^W in O(k) time
  - Maintain G^H in O(kn)
    - Because each element of W is changed, the whole matrix GH will be changed
  - Restrict to either W or H for a sequence

$$(W^i, H^i) \to (W^{i,1}, H^i) \to (W^{i,2}, H^i) \cdots$$

Variable Selection Strategy

- Stop condition
  - At the beginning of updates to W, store

$$p^{\text{init}} = \max_{i,j} D_{ij}^{W}$$

  - Iteratively choose variables to update to meet

$$\max_{i,j} D_{ij}^{W} < \epsilon p^{\text{init}}$$

    - Note that it can be achieved in a finite number of iterations because f(W, H) is lower bounded, the minimum for f(W,H) with fixed H is achievable.

Variable Selection Strategy

- A more efficient row-based variable selection
  - When k<<m, the $\log m$ term will cost dominately
  - Row-based selection
    - Changes in the i-th row of D^W will not affect the other rows
    - Iteratively update variables in the i-th row until meeting

$$\max_j D_{ij}^W < \epsilon p^{\text{init}}$$

      - Note that choose the largest value in one row costs O(k), cheaper than O(logm)
    - Then update the other rows.
    - Taking O(k) time totally for each variable update.

Algorithm 1 GCD for least squares NMF

- Given: V, k, ϵ (typically, $\epsilon = 0.001$)
- Output: W, H
- Compute $P^{VH} = VH^T$, $P^{HH} = HH^T$, $P^{WV} = W^TV$, $P^{WW} = W^TW$
- Initialize $H^{\text{new}} \leftarrow 0$
- While (not converged)
  1. Compute $P^{VH} \leftarrow P^{VH} + V(H^{\text{new}})^T$ according to the sparsity of H^{new} — $\min(O(mt), O(nmk))$
  2. $W^{\text{new}} \leftarrow 0$
  3. $G^W \leftarrow WP^{HH} - P^{VH}$
  4. $S^W_{ir} \leftarrow \max(W_{ir} - \frac{G^W_{ir}}{P^{HH}_{rr}}, 0) - W_{ir}$ for all i, r.
  5. $D^W_{ir} \leftarrow -G^W_{ir}S^W_{ir} - \frac{1}{2}P^{HH}_{rr}(S^W_{ir})^2$ for all i, r. — $O(mk^2)$
  6. $q_i \leftarrow \arg\max_j D^W_{ij}$ for all $i = 1, \ldots, m$, and $p^{\text{init}} \leftarrow \max_i D^W_{i,q_i}$
  7. For $i = 1, 2, \ldots, m$
     - While $D^W_{i,q_i} < \epsilon p^{\text{init}}$
       - 7.1. $s^* \leftarrow S^W_{i,q_i}$
       - 7.2. $P^{WW}_{q_i,:} \leftarrow P^{WW}_{q_i,:} + s^*W_{q_i,:}$ (Also do a symmetric update for $P^{WW}_{:,q_i}$) — $O(tk)$
       - 7.3. $W^{\text{new}}_{i,q_i} \leftarrow W^{\text{new}}_{i,q_i} + s^*$
       - 7.4. $G^W_{i,:} \leftarrow G^W_{i,:} + s^*P^{HH}_{q_i,:}$
       - 7.5. $S^W_{ir} \leftarrow \max(W_{ir} - \frac{G^W_{ir}}{P^{HH}_{rr}}, 0) - W_{ir}$ for all $r = 1, \ldots, k$. — $O(tk)$
       - 7.6. $D^W_{ir} \leftarrow -G^W_{ir}S^W_{ir} - \frac{1}{2}P^{HH}_{rr}(S^W_{ir})^2$ for all $r = 1, \ldots, k$.
       - 7.7. $q_i \leftarrow \arg\max_j D^W_{ij}$.
  8. $W \leftarrow W + W^{\text{new}}$
  9. For updates to H, repeats analogues steps to Step 1 through Step 8.

Variable Selection Strategy

- To get the amortized cost per coordinate update, divide the numbers by t

$\min(O(mt), O(nmk))$
$O(mk^2)$
$O(tk)$
$O(tk)$

$$\begin{cases} O(\frac{mk^2}{t}) & \text{if } k^2 > t \\ O(m) & \text{if } nk > t \geq k^2 \\ O(\frac{nmk}{t}) & \text{if } nm > t \geq nk \\ O(k) & \text{if } t > nm \end{cases}$$

Fast Coordinate Descent Method with Variable Selection

- Coordinate Gradient Method
- Variable Selection Strategy
- CGD for KL-Divergence
- Convergence Analysis
- Experiment Result

Coordinate Descent Method for NMF with KL-Divergence

- Apply coordinate descent for solving NMF with KL-divergence
  - Consider one-variable sub-problem

$$h_{ir}(s) = L(W + sE_{ir}, H) \tag{18}$$
$$= \sum_{j=1}^{n} -V_{ij} \log\left((WH)_{ij} + sH_{rj}\right) + sH_{rj} + \text{constant.}$$

  - Unlike least squares NMF, it has no closed form solution

Coordinate Descent Method for NMF with KL-Divergence

- The method in FastHals

$$\bar{h}_{ir}(s) = \sum_j -(V_{ij} - \sum_{t \neq r} W_{it} H_{tj}) \log(s H_{rj}) + s H_{rj}.$$

  - Solve a different problem to approximate it
  - Have close form solution
  - May converge to a different final solution.

Coordinate Descent Method for NMF with KL-Divergence

- Propose to solve it with Newton's method

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

  - Where

$$h'_{ir}(s) = \sum_{j=1}^{n} H_{rj}\left(1 - \frac{V_{ij}}{(WH)_{ij} + sH_{rj}}\right).$$

$$h''_{ir}(s) = \sum_{j=1}^{n} \frac{V_{ij}H_{rj}^2}{((WH)_{ij} + sH_{rj})^2}.$$

  - Takes O(n) time for summation

Coordinate Descent Method for NMF with KL-Divergence

- Note that the case of $V_{ij} = 0, (WH)_{ij} = 0$
  - For $V_{ij} = 0$, then $V_{ij} \log((WH)_{ij}) = 0$ for all positive values $(WH)_{ij}$ ignore those entries.
  - For $(WH)_{ij} + sH_{ij} = 0$, the Newton direction will be infinity, thus, reset s so that $W_{ir} + s$ is a small positive value and restart the Newton method

Coordinate Descent Method for NMF with KL-Divergence

Theorem 1

If a function $f(x)$ with domain $x \geq 0$ can be written in the following form

$$f(x) = -c_i \sum_{i=1}^{l} \log(a_i + b_i x) + \sum_{j} b_i x,$$

where $a_i > 0, b_i, c_i \geq 0 \; \forall i$, then the Newton method without line search converges to the global minimum of $f(x)$.

- Theorem 1 shows that Newton method for the special objective function converges without line search

Coordinate Descent Method for NMF with KL-Divergence

- Computational Complexity
  - To maintain the gradient similar to least squares

$$h'_{ir}(s) = \sum_{j=1}^{n} H_{rj}\left(1 - \frac{V_{ij}}{(WH)_{ij} + sH_{rj}}\right)$$

  - The complexity is $O(nk)$

$$(WH)_{ij} = (WH)_{ij} + s^* H_{rj} \quad \forall j \qquad O(n)$$

$$\Downarrow$$

$$h'_{it}(0) \text{ for all } t = 1, \ldots, k \qquad O(k)$$

  - It is expensive compared to the time cost $O(n)$ for updating one variable. DO NOT maintain gradient!
  - Adopt Cyclic Coordinate Descent, taking $O(nd)$ for each coordinate update

Coordinate Descent Method for NMF with KL-Divergence

Algorithm 2 CCD for NMF with KL-divergence

1. Given: V, k, W, H, ϵ (typically, $\epsilon = 0.5$)
2. Output: W, H
3. $P^{WH} \leftarrow WH$.
4. While (not converged)
   - 4.0.1. For $i = 1, \ldots, m$ (updates in W)
     - For $r = 1, \ldots, k$
       - While 1
         - Compute s by (19).
         - $w^{\text{old}} = W_{ir}$.
         - $W_{ir} \leftarrow W_{ir} + s$.
         - Maintain $(WH)_{i,:}$ by (22). $O(n)$
         - If $|s| < \epsilon w^{\text{old}}$, Break
   - 4.0.2. For updates to H, repeats steps analogous to Step 4.0.1

Fast Coordinate Descent Method with Variable Selection

- Coordinate Gradient Method
- Variable Selection Strategy
- CGD for KL-Divergence
- Convergence Analysis
- Experiment Result

Convergence Property

- For least squares

Theorem 2

For least squares NMF, if a sequence $\{(W_i, H_i)\}$ *is generated by* GCD, *then every limit point of this sequence is a stationary point.*

Method	Convergence
Multiplicative	Not guarantee converge to a stationary point
Gradient Descent Method	Lack convergent theory to support this method
ANLS (with exact solution)	Any limit point is a stationary point
GCD	Any limit point is a stationary point

Convergence Property

- For KL-Divergence

Theorem 3
For any limit points (W^*, H^*) *of* CCD *(or* FastHals*), assume* w_r^* *is the* r*th column of* W^* *and* h_r^* *is the* r*th row of* H^**, if*

$$\|w_r^*\| > 0, \|h_r^*\| > 0 \quad \forall r = 1, \ldots, k, \tag{24}$$

then (W^*, H^*) *is a stationary point of* (3) *(or* (1)*).*

Fast Coordinate Descent Method with Variable Selection

- Coordinate Gradient Method
- Variable Selection Strategy
- CGD for KL-Divergence
- Convergence Analysis
- Experiment Result

Experiment Result

- Stopping condition
  - Adopt projected gradient as stopping condition

$$\nabla^P_W f(W,H)_{ir} \equiv \begin{cases} \frac{\partial}{\partial W_{ir}} f(W,H) & \text{if } W_{ir} > 0, \\ \min(0, \frac{\partial}{\partial W_{ir}} f(W,H)) & \text{if } W_{ir} = 0. \end{cases}$$

  - According to KKT, (W^*, H^*) is a stationary point if and only if $\nabla^P f(W^*, H^*) = 0$. Use it to measure how close to stationary point

$$\|\nabla^P f(W,H)\|_F^2 \leq \epsilon \|\nabla^P f(W^0, H^0)\|_F^2.$$

Experiment Result

- Least square NMF on dense data
  - FLOP: num of floating point operations

dataset	m	n	k	relative error	Time (in seconds)/FLOPs			
					GCD	FastHals	ProjGrad	BlockPivot
Synth03	500	1,000	10	10^{-4}	**0.6/0.7G**	2.3/2.9G	2.1/1.4G	1.7/1.1G
			30	10^{-4}	**4.0/5.0G**	9.3/16.1G	26.6/23.5G	12.4/8.7G
Synth08	500	1,000	10	10^{-4}	**0.21/0.11G**	0.43/0.38G	0.53/0.41G	0.56/0.35G
			30	10^{-4}	**0.43/0.46G**	0.77/1.71G	2.54/2.70G	2.86/1.43G
CBCL	361	2,429	49	0.0410	**2.3/2.3G**	4.0/10.2G	13.5/14.4G	10.6/8.1G
				0.0376	**8.9/8.8G**	18.0/46.8G	45.6/49.4G	30.9/29.8G
				0.0373	**14.6/14.5G**	29.0/75.7G	84.6/91.2G	51.5/53.8G
ORL	10,304	400	25	0.0365	**1.8/2.7G**	6.5/14.5G	9.0/9.1G	7.4/5.4G
				0.0335	**14.1/20.1G**	30.3/66.9G	98.6/67.7G	33.9/38.2G
				0.0332	**33.0/51.5G**	63.3/139.0G	256.8/193.5G	76.5/82.4G

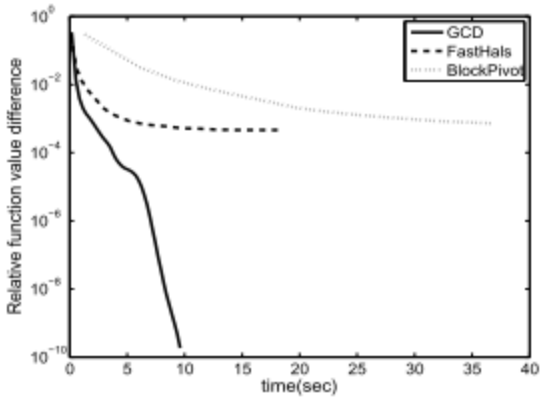
Experiment Result

- KL NMF on dense data

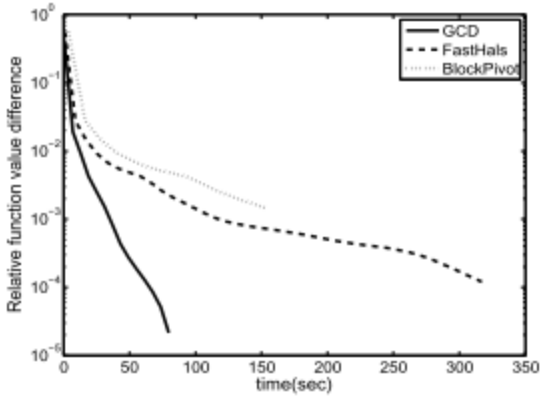
dataset	k	relative error	Time (in seconds)/FLOPs	
			CCD	Multiplicative
Synth03	10	10^{-3}	**11.4/5.2G**	34.0/68.1G
		10^{-5}	**14.8/6.8G**	144.2/240.6G
	30	10^{-3}	**121.1/58.7G**	749.5/2057.4G
		10^{-5}	**184.32/89.3G**	7092.3/18787.8G
Synth08	10	10^{-2}	**2.5/1.7G**	30.3/71.6G
		10^{-5}	**13.0/8.8G**	*
	30	10^{-2}	**22.6/11.2G**	46.0/93.9G
		10^{-5}	**56.8/27.7G**	*
CBCL	49	0.1202	38.2/**18.2G**	**21.2**/64.1G
		0.1103	**123.2/58.4G**	562.6/781.3G
		0.1093	**166.0/78.7G**	3266.9/2705.4G
ORL	25	0.3370	**73.7/35.0G**	165.2/336.3G
		0.3095	**253.6/117.0G**	902.2/1323.0G
		0.3067	**370.2/177.5G**	1631.9/3280.2G

Experiment Result

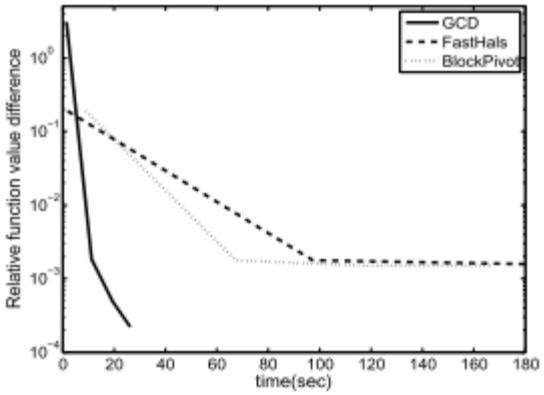
- Objective value reduced on sparse data



(a) Objective value for Yahoo-News dataset



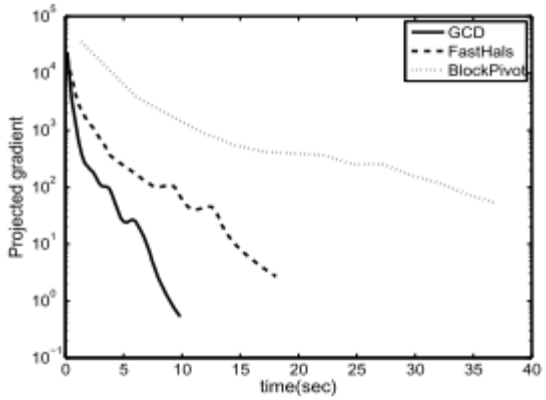
(d) Objective value for MNIST dataset



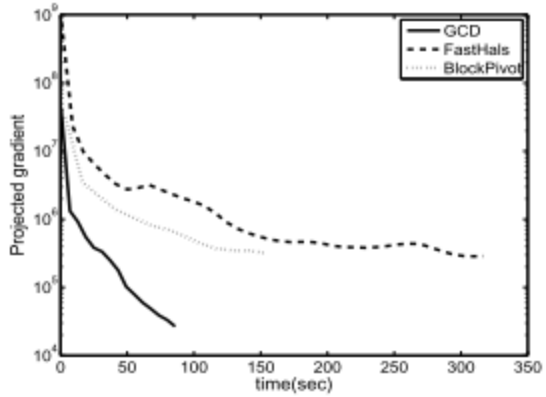
(g) Objective value for RCV1 dataset

Experiment Result

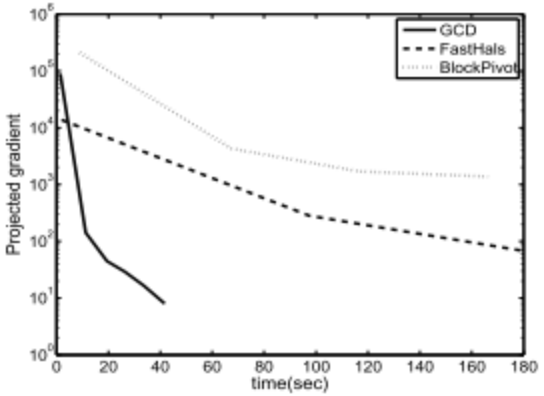
- Projected gradient on sparse data



(b) Project gradient for Yahoo-News dataset



(e) Projected gradient for MNIST dataset



(h) Projected gradient for RCV1 dataset

Reference

- Hsieh, Cho-Jui, and Inderjit S. Dhillon. "Fast coordinate descent methods with variable selection for non-negative matrix factorization." *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2011.
- Lin, Chih-Jen. "Projected gradient methods for nonnegative matrix factorization." *Neural computation* 19.10 (2007): 2756-2779.
- Berry, Michael W., et al. "Algorithms and applications for approximate nonnegative matrix factorization." *Computational statistics & data analysis* 52.1 (2007): 155-173.

Thank you