Compressing clustered data using Sparse NMF

San Kim and Michael Saunders

ICME and MS&E, Stanford University

SIAM CSE, Salt Lake City, UT

14–18 March 2015

Compressing clustered data using Sparse NMF

We propose a method for storing clustered data compactly, where each cluster is a collection of many similar datasets in matrix form (such as a set of related images). We first compute basis elements for each cluster using low-rank SNMF via PDCO. We then use the preprocessing approach of Gillis (2012) to sparsify the full set of (already sparse) basis elements, without significant loss of quality.

Partially supported by

National Institutes of Health (NIH) Award U01GM102098
Army Research Office (ARO) Award W911NF-13-1-0072

Sparse NMF

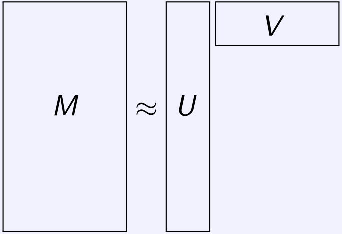
Happy π Day!

3/14/15

Sparse NMF

Given $m \times n$ nonnegative matrix M, find $m \times r$ and $r \times n$ matrices U and V to solve

$$\min_{U,V} \|M - UV\|_F^2 \quad \text{such that} \quad U \geq 0, \; V \geq 0$$



Main reference

Nicolas Gillis (2012). Sparse and unique nonnegative matrix factorization through data preprocessing, *J. of Machine Learning Research* 13, 3349–3386 (38 pages!)

Main idea

Preprocessing: Change M to sparser MQ, where $MQ \geq 0$ and $Q^{-1} \geq 0$

Main benefit

Sparse NMF is more efficient on MQ

Side note

Do we really need $\|\cdot\|_0$ or $\|\cdot\|_1$ to achieve sparsity?

There's a big difference between

$$\min \|x\|_2^2 \;\text{ such that }\; Ax = b \qquad \text{(will have } \textit{no} \;\; x_j = 0\text{)}$$

and

$$\min \|x\|_2^2 \;\text{ such that }\; Ax = b,\; x \geq 0 \qquad \text{(could have } \textit{many} \; x_j = 0\text{)}$$

Experiments before Thanksgiving

```
from: San Kim &lt;sankim@stanford.edu&gt;
to: Michael Saunders &lt;saunders@stanford.edu&gt;
date: Tue, Nov 18, 2014 at 11:56 PM

Dear Professor Saunders,

How are you?

I have been studying parallel computing, and was able to modify the
MATLAB codes.  It is now working in parallel. There are 2 CPUs in my
computer so I could do some experiments. With 300 images, our code
finished the job in 54 seconds, while it took 556 seconds using the
Gillis' code. Also, I could modify the Gillis' code to be working in
parallel, which would probably take about 280 seconds.

Time difference will be greater if we use more images. To show that,
my computer is now working with 800 images.

Is it possible to visit your office on Thursday after you have dinner
with the speaker? If you are too busy, it is ok with me to visit your
office after Thanksgiving week. I still need to figure out how to
modify the code to have "warm start" when we have additional images.

Thank you very much! Have a great dream :)

Sincerely,
San
```

```
from: Michael Saunders <saunders@stanford.edu>
to: San Kim <sankim@stanford.edu>
date: Fri, Nov 21, 2014 at 10:00 AM
subject: Re: Professor

Dear San,

At 4:30am this morning I suddenly remembered your important email.
I think I was too sleepy to answer when you sent it and then I somehow
forgot to answer next day.  I'm so sorry.  Last night after taking the seminar
speaker to dinner I had to go to another dinner at the faculty club in honor
of Antony Jameson's 80th birthday.  (Famous Aero/Astro professor.)
I got back to my office about 10pm.

It's wonderful that you have your solver running in parallel now,
and how much faster it is than the Gillis code.  Splendid progress!
It should be enough already for a good talk at the conference.

Really sorry again to miss you.
Stay in touch for planning another time to meet.
Have happiest Thanksgiving meanwhile!

Michael
```

from: San <sankim@stanford.edu>
to: Michael Saunders <saunders@stanford.edu>
date: Fri, Nov 21, 2014 at 6:44 PM
subject: Re: Professor

Dear Professor Saunders,

Thank you very much for the reply! Don't worry about me, Professor! I was just going to give you a quick update on the processing time, which can also be given via email :) Here it is:

With 800 images, our code finished the job in 3 minutes and 30 seconds, while the Gillis' code took 3 hours and 40 minutes. Also, I ran the code on the whole set of images (2429 images). Our code took only 35 minutes, where the Gillis' code should take several days. And, there is still plenty of room for speed improvement.

It is not only faster but also better in terms of image quality. At same level of sparsity, our method gives much better image quality. Or, at similar image quality, we need much fewer nonzero elements.

I think we can use this method in many places (not only in image processing), where Sparse NMF or Low-Rank Factorization is needed.

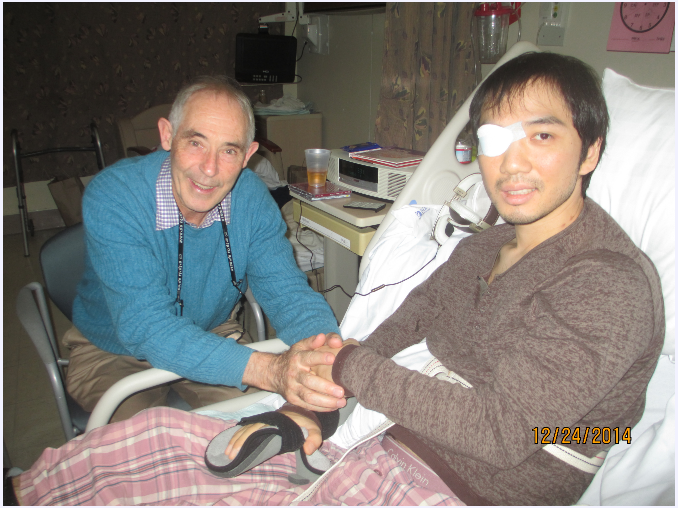
I guess I'll see you in December. I am kind of surprised that it is already December. Time goes really fast :(

Have a great Thanksgiving, Professor! Also, hope you have a wonderful time with your daughters :)

Sincerely,
San

Hospital

Coauthor San Kim



San and parents

San and Young leaving hospital

Preprocessing sparse NMF problems
Nicolas Gillis (2012)

Preprocessing

Find Q so that MQ is more sparse than M, $\quad MQ \geq 0, \quad$ and $\quad Q^{-1} \geq 0$

- Given M, find sparse Q
- Given MQ, find Sparse NMF $\quad MQ \approx U\bar{V}$
- Given Q, U, $\bar{V}$, we see that $V \equiv \bar{V}Q^{-1} \geq 0$ gives $M \approx UV$

Q might be nearly singular, so get V from $\min_{V \geq 0} \|UV - M\|_F^2$

Decouples into n Sparse NLS problems:

$$\min_{\mathbf{v}_j \geq 0} \|U\mathbf{v}_j - \mathbf{m}_j\|_2^2 \qquad\qquad U,\ \mathbf{v}_j \text{ sparse}$$

Preprocessing

Find Q so that MQ is more sparse than M, $\quad MQ \geq 0, \quad$ and $\quad Q^{-1} \geq 0$

- Ideally: $\min \|MQ\|_0 \quad \text{such that} \quad MQ \geq 0$
- Instead: $\min \|MQ\|_F^2 \quad \text{such that} \quad MQ \geq 0 \qquad (*)$
- Should normalize columns of M first
- Consider $Q = I - B$ with

$$B \geq 0, \qquad \mathrm{diag}(B) = 0, \qquad \rho(B) < 1 \quad (\Rightarrow Q^{-1} \geq 0)$$

- If M contains no repeated cols, $(*)$ satisfies $\rho(B) < 1$ automatically

Preprocessing with $Q = I - B$

$$\min \|M - MB\|_F^2 \quad \text{such that} \quad M \geq MB, \ \text{diag}(B) = 0$$

Decouples into n problems:

$$\min \|\mathbf{m}_j - M\mathbf{b}_j\|_2^2 \quad \text{such that} \quad M\mathbf{b}_j \leq \mathbf{m}_j, \quad \mathbf{b}_{jj} = 0$$

For each j:

$$\min \|\mathbf{s}\|^2 \quad \text{such that} \quad M\mathbf{b} + \mathbf{s} = \mathbf{m}, \quad \mathbf{b}, \mathbf{s} \geq 0, \quad \mathbf{b}_j = 0$$

$$\Rightarrow \quad \min \|\mathbf{s}\|^2 \quad \text{such that} \quad M\mathbf{b} + \mathbf{s} = 0, \quad \mathbf{b}, \mathbf{s} \geq 0, \quad \mathbf{b}_j = -1$$

Preprocessing with $Q = I - B$

For each j, $\mathbf{s}$ will be a *sparse* column of $MQ \equiv M(I-B)$:

$$\min \|\mathbf{s}\|^2 \quad \text{such that} \quad M\mathbf{b} + \mathbf{s} = 0, \qquad \mathbf{b}, \mathbf{s} \geq 0, \quad \mathbf{b}_j = -1$$

Solve by PDCO (interior method)

$$\begin{aligned} \min \quad & \tfrac{1}{2}\|\gamma\mathbf{b}\|^2 + \tfrac{1}{2}\|\mathbf{s}\|^2 + \tfrac{1}{2}\|\mathbf{r}\|^2 \\ \text{st} \quad & M\mathbf{b} + \mathbf{s} + \delta\mathbf{r} = 0, \qquad\qquad \gamma,\ \delta \approx 10^{-4} \\ & \mathbf{b}, \mathbf{s} \geq 0, \quad \mathbf{b}_j = -1 \end{aligned}$$

This is harder than Sparse NLS (but **b**, **s** will be sparse)

Ideally include *exact regularization* of Friedlander & Orban, MPC (2012)

Sparse NMF Solvers

Sparse NMF Solvers $\quad A \approx WH$

- Hoyer (2004) NMFPACK
- Kim and Park (2007)

$$\min_{W,H\geq 0} \frac{1}{2}\|A - WH\|_F^2 + \eta\|W\|_F^2 + \beta \sum \|h_j\|_1^2$$

Alternating Sparse NLS problems

- Jin and Saunders (2008)

$$\min_{W,H\geq 0} \frac{1}{2}\|A - WH\|_F^2 + \eta\sum\|w_i\|_1 + \beta \sum \|h_j\|_1$$

Alternating BPDN problems (via BPDual or PDCO)

- Gillis and Glineur (2012) Accelerated HALS algorithm (A-HALS)

SNMF via PDCO — implementation

$$\min_{U,D,V\geq 0} \tfrac{1}{2}\left\|A - UDV^T\right\|_F^2 + \sum \beta_i \|Du_i\|_1 + \sum \eta_j \|Dv_j\|_1$$

Alternating PDCO on

$$\min_{v_j\geq 0} \tfrac{1}{2}\|Uv_j - a_{.j}\|^2 + \eta_j\|v_j\|_1 \quad \text{for } j = 1:n, \qquad \text{normalize } V \to VD$$

$$\min_{u_i\geq 0} \tfrac{1}{2}\|Vu_i - a_{i.}\|^2 + \beta_i\|u_i\|_1 \quad \text{for } i = 1:m, \qquad \text{normalize } U \to UD$$

- $\eta_j \leq \sigma \left\|U^T a_{.j}\right\|_\infty$ $\quad$ σ = "sparsity" input parameter
  $\beta_i \leq \sigma \left\|V^T a_{i.}\right\|_\infty$ $\quad$ = 0.9 or 0.8 say
- At some point, *freeze* D $\quad$ (Also η_j and β_i stop changing)

San's ideas

1: Combine multiple BPDN subproblems

- Apply Sparse NMF (PDCO) to M (before preprocessing!)
- Need to solve hundreds of decoupled BPDN subproblems:

$$P_j \equiv \min_{v_j \geq 0} \ \tfrac{1}{2} \left\| U v_j - a_{.j} \right\|^2 + \eta_j \left\| v_j \right\|_1$$

- Combine many subproblems into one big BPDN subproblem for PDCO:

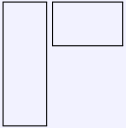
$$\begin{bmatrix} P_1 & & & & \\ & P_2 & & & \\ & & \ddots & & \\ & & & \ddots & \\ & & & & P_{361} \end{bmatrix}$$

- Solve multiple big problems in parallel (Matlab's parfor)

Last Saturday
at Starbucks

Catch up on
Matlab programs

2: Apply preprocessing to U, not M!

- Find $M \approx UV =$  using Jin-Saunders Sparse NMF solver

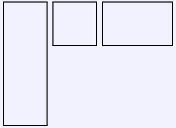
  (PDCO on groups of 361 BPDN subproblems)

- Find $UQ \approx \bar{U}\bar{V}$

  Gillis preprocessing, $Q = I - B$

  Q is much smaller than for M

- Solve $\min_{\tilde{V} \geq 0} \left\| \bar{U}\tilde{V} - U \right\|_F^2$

- Then $M \approx \bar{U}\tilde{V}V =$ 

Results

Image Processing

The CBCL face data set (110MB)

`http://cbcl.mit.edu/software-datasets/FaceData2.html`

Aim: Store clusters of similar images compactly

- 2429 gray-level images of faces (19×19 pixels)
- Approximation rank $= 49$
- `matlabpool open` gave 8 workers on campus Linux cluster
- Had to guess San's input parameters
- Hours later: "Cannot open file: permission denied."

Conclusions

Conclusions

- Document your codes
- Stay healthy