Dominant feature extraction

Paul Van Dooren, U.C.Louvain, Belgium
(with Gallivan, Chahlaoui, Ipsen, Chia-Tche, Mastronardi)

Large Graphs and Networks
UCL Université catholique de Louvain

Eurasip lecture 1, Bari, August 2012

Goal of these lectures

Develop basic ideas for large scale dense matrices and recursive procedures for

- Dominant singular subspace

Goal of these lectures

Develop basic ideas for large scale dense matrices and recursive procedures for

- Dominant singular subspace
- Multipass iteration

Goal of these lectures

Develop basic ideas for large scale dense matrices and recursive procedures for

- Dominant singular subspace
- Multipass iteration
- Subset selection

Goal of these lectures

Develop basic ideas for large scale dense matrices and recursive procedures for

- Dominant singular subspace
- Multipass iteration
- Subset selection
- Dominant eigenspace of positive definite matrix

Goal of these lectures

Develop basic ideas for large scale dense matrices and recursive procedures for

- Dominant singular subspace
- Multipass iteration
- Subset selection
- Dominant eigenspace of positive definite matrix
- Dominant eigenspace for indefinite matrices

Goal of these lectures

Develop basic ideas for large scale dense matrices and recursive procedures for

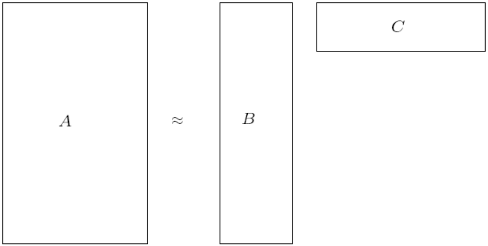
- Dominant singular subspace
- Multipass iteration
- Subset selection
- Dominant eigenspace of positive definite matrix
- Dominant eigenspace for indefinite matrices
- Show accuracy and complexity results

The indefinite case introduces a new matrix decomposition (presented in lecture 2)

Dominant singular subspaces

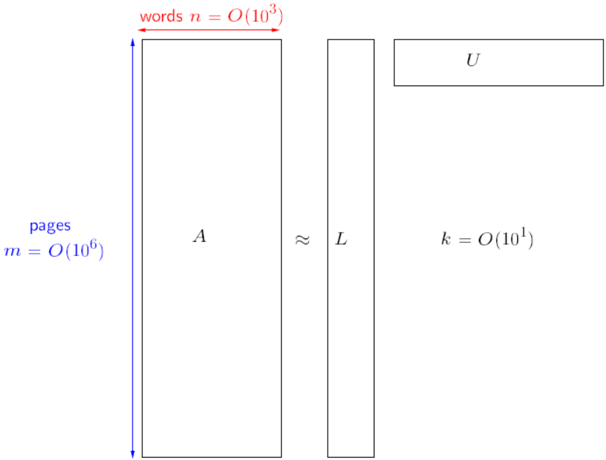
Given $A_{m\times n}$, approximate it by a rank k factorization $B_{m\times k}C_{k\times n}$ by solving

$$\min \|A - BC\|_2, \quad k \ll m, n$$



This has several applications in Image compression, Information retrieval and Model reduction (POD)

Information retrieval

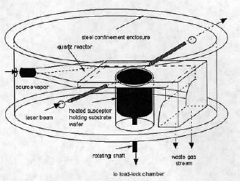


- Low memory requirement $0(k(m+n))$
- Fast queries $Ax \approx L(Ux)$ $0(k(m+n))$ time
- Easy to obtain $0(kmn)$ flops

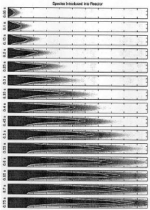
Proper Orthogonal decomposition (POD)

Compute a state trajectory for one “typical" input

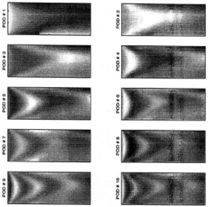
Collect the principal directions to project on



Quartz reactor



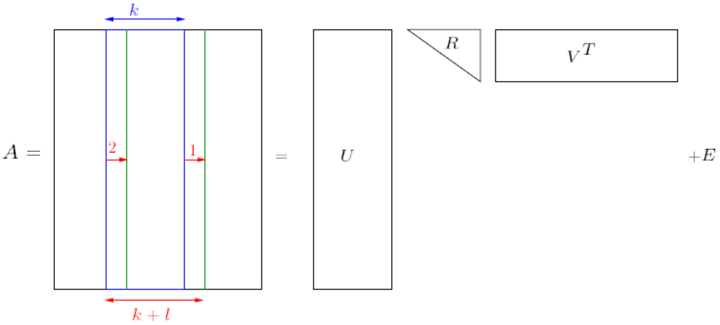
Snap shots of "typical" states



Ten dominant "states"

Recursivity

We pass once over the data with a window of length k and perform along the way a set of windowed SVD's of dimension $m \times (k + \ell)$



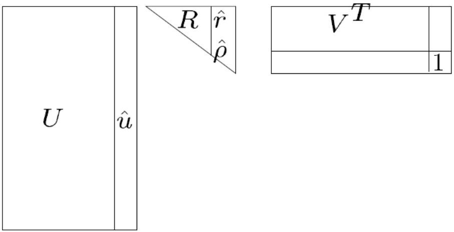
Step 1 : expand by appending ℓ columns (Gram Schmidt)
Step 2 : contract by deleting the ℓ least important columns (SVD)

Expansion (G-S)

Append column a_+ to the current approximation URV^T to get

$$\begin{bmatrix} URV^T & a_+ \end{bmatrix} = \begin{bmatrix} U & a_+ \end{bmatrix} \begin{bmatrix} R & 0 \\ & 1 \end{bmatrix} \begin{bmatrix} V^T & \\ & 1 \end{bmatrix}$$

Update with Gram Schmidt to recover a new decomposition $\hat{U}\hat{R}\hat{V}^T$:

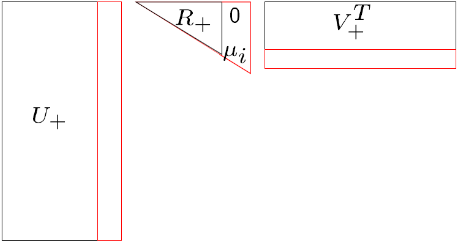


using $\hat{r} = U^T a_+, \ \hat{a} = a_+ - U\hat{r}, \ \hat{a} = \hat{u}\hat{\rho}$ (since $a_+ = U\hat{r} + \hat{u}\hat{\rho}$)

Contraction (SVD)

Now remove the ℓ smallest singular values of this new $\hat{U}\hat{R}\hat{V}^T$ via

$$\hat{U}\hat{R}\hat{V}^T = (\hat{U}G_u)(G_u^T\hat{R}G_v)(G_v^T\hat{V}^T) =$$



and keeping $U_+R_+V_+^T$ as best approximation of $\hat{U}\hat{R}\hat{V}^T$
(just delete the ℓ smallest singular values)

Complexity of one pair of steps

The Gram Schmidt update (expansion) requires $4mk$ flops per column (essentially for the products $\hat{r} = U^T a_+, \ \hat{a} = a_+ - U\hat{r}$)

For $G_u \hat{R} G_v = \begin{bmatrix} R_+ & 0 \\ & \mu_i \end{bmatrix}$ one requires the left and right singular vectors of $\hat{R}$ which can be obtained in $O(k^2)$ flops per singular value (using inverse iteration)

Multiplying $\hat{U} G_u$ and $\hat{V} G_v$ requires $4mk$ flops per deflated column

The overall procedure requires $8mk$ flops per processed column and hence $8mnk$ flops for a rank k approximation to a $m \times n$ matrix A

One shows that $A = U \begin{bmatrix} R & A_{12} \\ 0 & A_{22} \end{bmatrix} V^T$ where $\| \begin{bmatrix} A_{12} \\ A_{22} \end{bmatrix} \|_F^2$ is known

Error estimates

Let $E := A - \hat{A} = U\Sigma V^T - \hat{U}\hat{\Sigma}\hat{V}^T$ and $\mu := \|E\|_2$

Let $\hat{\mu} := \max \mu_i$ where μ_i is the neglected singular value at step i

One shows that the error norm

$$\hat{\mu} \leq \sigma_{k+1} \leq \mu \leq \sqrt{n-k}\hat{\mu} \approx c\hat{\mu}$$

$$\hat{\sigma}_i \leq \sigma_i \preceq \hat{\sigma}_i + \hat{\mu}^2/2\hat{\sigma}_i$$

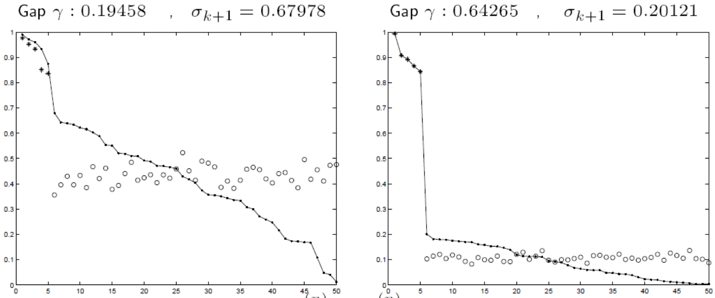
$$\tan\theta_k \preceq \tan\hat{\theta}_k := \hat{\mu}^2/(\hat{\sigma}_k^2 - \hat{\mu}^2), \quad \tan\phi_k \preceq \tan\hat{\phi}_k := \hat{\mu}\hat{\sigma}_1/(\hat{\sigma}_k^2 - \hat{\mu}^2)$$

where θ_k, ϕ_k are the canonical angles of dimension k :

$$\cos\theta_k := \|U^T(:,k)\hat{U}\|_2, \quad \cos\phi_k := \|V^T(:,k)\hat{V}\|_2$$

Examples

The bounds get much better when the gap $\sigma_k - \sigma_{k+1}$ is large

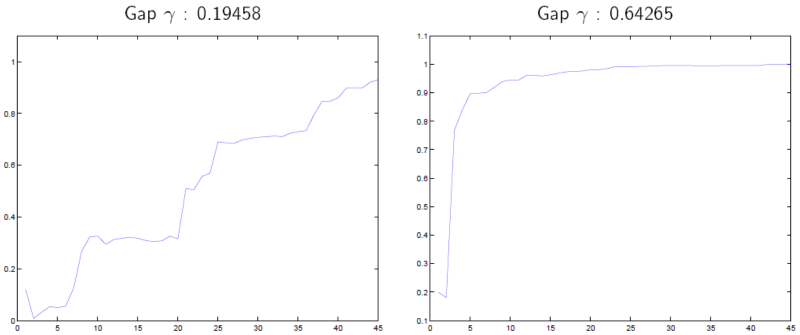


— true sv's $\sigma_i(A)$, $*$ approximated sv's $\hat{\sigma}_1^{(n)}, \ldots, \hat{\sigma}_k^{(n)}$, $\circ$ dismissed sv's $\mu_{k+1}, \ldots, \mu_n < \sigma_{k+1}$

$\sigma_1 = 0.99008$	$\hat{\sigma}_1 = 0.97613$	$\sigma_1 = 0.99430$	$\hat{\sigma}_1 = 0.99418$
$\sigma_2 = 0.97084$	$\hat{\sigma}_2 = 0.95301$	$\sigma_2 = 0.90840$	$\hat{\sigma}_2 = 0.90815$
$\sigma_3 = 0.96010$	$\hat{\sigma}_3 = 0.93379$	$\sigma_3 = 0.89284$	$\hat{\sigma}_3 = 0.89250$
$\sigma_4 = 0.93338$	$\hat{\sigma}_4 = 0.85142$	$\sigma_4 = 0.86560$	$\hat{\sigma}_4 = 0.86551$
$\sigma_5 = 0.87437$	$\hat{\sigma}_5 = 0.83675$	$\sigma_5 = 0.84387$	$\hat{\sigma}_5 = 0.84357$
$\mu = 0.73768$	$\hat{\mu} = 0.52330$	$\mu = 0.20140$	$\hat{\mu} = 0.13631$
$\cos\theta_k = 0.93000$	$\cos\hat{\theta}_k = 0.82233$	$\cos\theta_k = 0.99998$	$\cos\hat{\theta}_k = 0.99459$
$\cos\phi_k = 0.83881$	$\cos\hat{\phi}_k = 0.71038$	$\cos\phi_k = 0.99935$	$\cos\hat{\phi}_k = 0.94334$

Convergence

How quickly do we track the subpaces ?



How $\cos\theta_k^{(i)}$ evolves with the time step i

Example

Find the dominant behavior in an image sequence

Images can have up to 10^6 pixels

Each column of A is one image

Original : $m = 28341,\ n = 100$

Approximation : $k = 6$

Multipass iteration

Low Rank Incremental SVD can be applied in several passes, say to

$$\frac{1}{\sqrt{k}} \begin{bmatrix} A & A & \dots & A \end{bmatrix}$$

After the first block (or "pass") a good approximation of the dominant space $\hat{U}$ has already been constructed

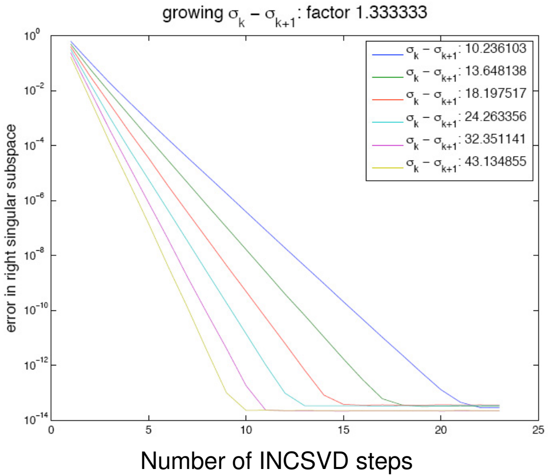
Going over to the next block (second "pass") will improve it, etc.

Theorem Convergence of the multipass method is linear, with approximate ratio of convergence $\psi/(1-\kappa^2) < 1$, where

- ψ measures orthogonality of the residual columns of A
- κ is the ratio σ_k/σ_{k+1} of A

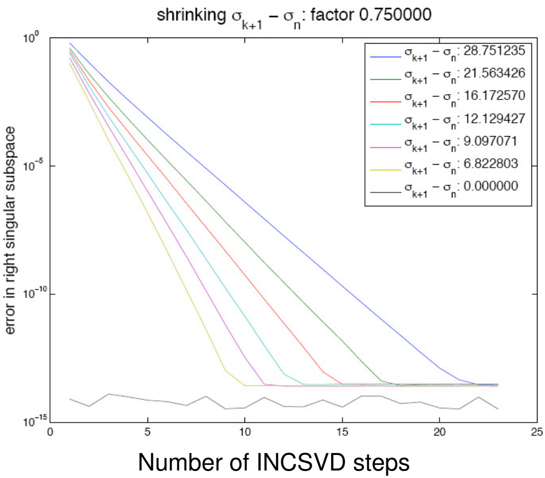
Convergence behavior

for increasing gap between “signal" and “noise"



Convergence behavior

for increasing orthogonality between “residual vectors"



Eigenfaces analysis

Ten dominant left singular vectors of ORL Database of faces
(40 images, 10 subjects, 92$\times$112 pixels = 10304$\times$400 matrix)

Using MATLAB' SVD function

Using one pass of incremental SVD

Maximal angle : 16.3°, maximum relative error in sing. values : 4.8%

Conclusions Incremental SVD

A useful and economical SVD approximation of $A_{m,n}$

For matrices with columns that are very large or “arrive" with time

Complexity is proportional to mnk and the number of “passes"

Algorithms due to
[1] Manjunath-Chandrasekaran-Wang (95)
[2] Levy-Lindenbaum (00)
[3] Chahlaoui-Gallivan-VanDooren (01)
[4] Brand (03)
[5] Baker-Gallivan-VanDooren (09)

Convergence analysis and accuracy in refs [3],[4],[5]

Subset selection

We want a “good approximation" of A_{mn} by a product $B_{mk}P^T$ where P_{nk} is a “selection matrix" i.e. a submatrix of the identity I_n

This seems connected to

$$\min \|A - BP^T\|_2$$

and maybe similar techniques can be used as for incremental SVD

Clearly, if $B = AP$, we just select a subset of the columns of A

Rather than minimizing $\|A - BP^T\|_2$ we maximize $\mathrm{vol}(B)$ where

$$\mathrm{vol}(B) = \det(B^T B)^{\frac{1}{2}} = \prod_{i=1}^{k} \sigma_i(B), \qquad m \geq k$$

There are $\begin{pmatrix} n \\ k \end{pmatrix}$ possible choices and the problem is NP hard and there is no polynomial time approximation algorithm

Heuristics

Gu-Eisenstat show that the Strong Rank Revealing QR factorization (SRRQR) solves the following simpler problem

B is sub-optimal if there is no swapping of a single column of A (yielding $\hat{B}$) that has a larger volume (constrained minimum)

Here, we propose a simpler “recursive updating" algorithm that has complexity $O(mnk)$ rather than $O(mn^2)$ for Gu-Eisenstat

The idea is again based on a sliding window of size $k+1$ (or $k+\ell$)

Sweep through columns of A while maintaining a “best" subset B

- Append a column of A to B, yielding B_+
- Contract B_+ to $\hat{B}$ by deleting the “weakest" column of B_+

Deleting the weakest column

Let $B = A(:, 1:k)$ to start with and let $B = QR$ where R is $k \times k$

Append the next column a_+ of A to form B_+ and update its decomposition using Gram Schmidt

$$B_+ := \begin{bmatrix} QR & a_+ \end{bmatrix} = \begin{bmatrix} Q & a_+ \end{bmatrix} \begin{bmatrix} R & 0 \\ & 1 \end{bmatrix} = \begin{bmatrix} Q & \hat{q} \end{bmatrix} \begin{bmatrix} R & \hat{r} \\ & \hat{\rho} \end{bmatrix} = Q_+ R_+$$

with $\hat{r} = Q^T a_+, \ \hat{a} = a_+ - Q\hat{r}, \ \hat{a} = \hat{q}\hat{\rho}$ (since $a_+ = Q\hat{r} + \hat{q}\hat{\rho}$)

Contract B_+ to $\hat{B}$ by deleting the “weakest" column of R_+

This can be done in $O(mk^2)$ using Gu-Eisenstat’s SRRQR method but an even simpler heuristic uses only $O((m+k)k)$ flops

Kahan matrices are typical upper-triangular tests with $K_n = S_n T_n$ and

$$S_n = \begin{pmatrix} 1 & 0 & \cdots & 0 \\ 0 & \psi & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & \psi^{n-1} \end{pmatrix} \quad \text{and} \quad T_n = \begin{pmatrix} 1 & -\phi & \cdots & -\phi \\ 0 & 1 & \ddots & \vdots \\ \vdots & \ddots & \ddots & -\phi \\ 0 & \cdots & 0 & 1 \end{pmatrix}$$

with $\phi^2 + \psi^2 = 1$ and where $\psi = 0.9$

		Computation time	
k	$\kappa(A_{:,1:k})$	SRRQR	WSS/MWSS
20	1.4×10^{4}	0.4	0.1
50	3.0×10^{10}	1.3	0.1
100	6.3×10^{20}	3.0	0.2
150	2.2×10^{25}	6.8	0.3
200	1.9×10^{28}	8.9	0.5
300	1.6×10^{33}	24.2	1.0

Gap example

k	$\kappa(A_{:,1:k})$	Normalized volume			
		WSS	MWSS	$\text{RMWSS}_{R=1}$	$\text{RMWSS}_{R=2}$
20	1.3×10^6	0.136	1.008	[1.009; 1.009]	[1.009; 1.009]
40	2.1×10^{12}	0.013	1.009	[0.978; 1.010]	[0.994; 1.010]
60	9.5×10^{18}	0.002	1.001	[0.984; 1.012]	[1.001; 1.015]
80	8.2×10^{18}	< 0.001	1.025	[1.014; 1.034]	[1.016; 1.036]
100	9.5×10^{18}	< 0.001	1.079	[1.078; 1.111]	[1.091; 1.114]

TABLE 5.2

Normalized volume of the subsets of columns returned by the different algorithms for a 1000×1000 GKS matrix $A = G_{1000}$. The normalization has been done with respect to the volume found by SRRQR. The condition number of the default initial subset of columns is also shown.

k	Computation time				
	SRRQR	WSS	MWSS	$\text{RMWSS}_{R=1}$	$\text{RMWSS}_{R=2}$
20	3.1	0.1	0.8	[0.5; 0.6]	[1.0; 1.2]
40	9.4	0.1	1.2	[0.5; 0.7]	[1.1; 1.5]
60	4.5	0.2	3.2	[1.4; 3.2]	[3.0; 5.7]
80	6.0	0.3	2.0	[1.6; 3.3]	[3.8; 6.7]
100	18.8	0.3	2.4	[1.5; 2.1]	[3.5; 4.4]

TABLE 5.3

Computation time (in seconds) required to return a subset of k columns for a 1000×1000 GKS matrix $A = G_{1000}$.

References

Gu, Eisenstat, *An efficient algorithm for computing a strong rank revealing QR factorization*, SIAM SCISC, 1996

Chahlaoui, Gallivan, Van Dooren, *An incremental method for computing dominant singular subspaces*, SIMAX, 2001

Hoegaerts, De Lathauwer, Goethals, Suykens, Vandewalle, De Moor, *Efficiently updating and tracking the dominant kernel principal components* Neural Networks, 2007.

Mastronardi, Tyrtishnikov, Van Dooren, *A fast algorithm for updating and downsizing the dominant kernel principal components*, SIMAX, 2010

Baker, Gallivan, Van Dooren, *Low-rank incrmenetal methods for computing dominant singular subspaces*, submitted, 2010

Ipsen, Van Dooren, *Polynomial Time Subset Selection Via Updating*, in preparation, 2010