Approximate Principal Components Analysis of Large Data Sets

Daniel J. McDonald

Department of Statistics
Indiana University

http://mypage.iu.edu/∼dajmcdon
dajmcdon@indiana.edu

Joint work with:
Darren Homrighausen

August 5, 2014

Data reduction

Modern statistical applications — genomics, neural image analysis, text analysis — have large numbers of covariates p

Also frequently have lots of observations n.

Need to do some dimension reduction

Many methods — multidimensional scaling, discriminant analysis, locally linear embeddings, Laplacian eigenmaps, and many others

Notation

- Observe $\tilde{X}_1, \ldots, \tilde{X}_n$, where $\tilde{X}_i \in \mathbb{R}^p$
- Form $\tilde{\mathbf{X}} = \begin{bmatrix} \tilde{X}_1 & \cdots & \tilde{X}_n \end{bmatrix}^\top \in \mathbb{R}^{n \times p}$
- Call $\mathbf{X} = \tilde{\mathbf{X}} - \overline{\mathbf{X}}$ where $\overline{\mathbf{X}} = n^{-1}\mathbf{1}\mathbf{1}^\top \tilde{\mathbf{X}}$
- Write the SVD of $\mathbf{X}$ as

$$\mathbf{X} = \mathbf{U}\Lambda\mathbf{V}^\top,$$

- For general rank r matrices $\mathbf{A}$ we write

$$\mathbf{A} = U(\mathbf{A})\Lambda(\mathbf{A})V(\mathbf{A})^\top \quad \text{where} \quad \Lambda(\mathbf{A}) = \mathrm{diag}(\lambda_1(\mathbf{A}), \ldots, \lambda_r(\mathbf{A}))$$

- For some matrix $\mathbf{A}$, we use $\mathbf{A}_d$ to be the first d columns of $\mathbf{A}$.

Principal Components Analysis

- For $d \leq r$, PCA gives a projection that minimizes the squared error distance between the data and the projection
- Projections involve $\mathbf{U}_d$, $\mathbf{V}_d$, and Λ_d

Example

PCA for regression with response vector Y, the estimated coefficients would be

$$\widehat{\beta}_{pcr} = \mathbf{V}_d \Lambda_d^{\dagger} \mathbf{U}_d^{\top} Y = \mathbf{V}_d \mathbf{V}_d^{\top} \mathbf{X}^{\dagger} Y \tag{1}$$

and the fitted values can be written $\widehat{Y} = \mathbf{U}_d \mathbf{U}_d^{\top} Y$

A big problem

- PCA requires computing an SVD
- So do most other data reduction methods
- Datasets are large: a recent ImageNet contest had $n \approx 10^6$ and $p \approx 65000$[1]
- That was only about 8% of the data set.
- SVD requires $O(np^2 + n^3)$ computations check me
- And you need to store the entire matrix in fast memory
- This is bad.

[1] see e.g. Krizhevsky, Sutskever, and Hinton. *NIPS* 2013

A big problem

- PCA requires computing an SVD
- So do most other data reduction methods
- Datasets are large: a recent ImageNet contest had $n \approx 10^6$ and $p \approx 65000$[1]
- That was only about 8% of the data set.
- SVD requires $O(np^2 + n^3)$ computations check me
- And you need to store the entire matrix in fast memory
- This is bad.

[1] see e.g. Krizhevsky, Sutskever, and Hinton. *NIPS* 2013

Approximations

Can't take the SVD of $\mathbf{X}$

What about approximating it?

Focus on two methods of "approximate SVD"

1. Nyström extension
2. Column sampling

Not our methods.

A quick "sketch" of the intuition

- Both methods fall into a larger class of methods
- Suppose we want to approximate $\mathbf{A} \in \mathbb{R}^{q \times q}$
- Assume $\mathbf{A}$ is symmetric and positive semi-definite
- Choose $l \ll q$ and form a "sketching" matrix $\Phi \in \mathbb{R}^{q \times l}$
- Then write $\mathbf{A} \approx (\mathbf{A}\Phi)(\Phi^\top \mathbf{A}\Phi)^\dagger(\mathbf{A}\Phi)^\top$.
- Different Φ yield different approximations
- For Nyström and column sampling use

$$\Phi = \pi\tau$$

Where π is a permutation of $\mathbf{I}_q$ and $\tau = \begin{bmatrix}\mathbf{I}_l & \mathbf{0}\end{bmatrix}^\top$.

Notes on sketching

- We don't actually do this, but this is the generality
- Essentially, let

$$\mathbf{S} = \frac{1}{n}\mathbf{X}^\top \mathbf{X} \qquad \text{and} \qquad \mathbf{Q} = \mathbf{X}\mathbf{X}^\top$$

- Both of these are symmetric, positive semi-definite
- Randomly choose l entries in $\{1, \ldots, n\}$ and $\{1, \ldots, p\}$
- Then partition the matrix so the selected portion is $\mathbf{S}_{11}$ and $\mathbf{Q}_{11}$

$$\mathbf{S} = \mathbf{V}\Lambda^2\mathbf{V}^\top = \begin{bmatrix} \mathbf{S}_{11} & \mathbf{S}_{12} \\ \mathbf{S}_{21} & \mathbf{S}_{22} \end{bmatrix} \qquad \mathbf{Q} = \mathbf{U}\Lambda^2\mathbf{U}^\top = \begin{bmatrix} \mathbf{Q}_{11} & \mathbf{Q}_{12} \\ \mathbf{Q}_{21} & \mathbf{Q}_{22} \end{bmatrix}$$

Approximating things we don't care about

If we want to approximate $\mathbf{S}$ (or $\mathbf{Q}$), we have for example

Nyström

$$\mathbf{S} \approx \begin{bmatrix} \mathbf{S}_{11} \\ \mathbf{S}_{21} \end{bmatrix} \mathbf{S}_{11}^{\dagger} \begin{bmatrix} \mathbf{S}_{11} & \mathbf{S}_{12} \end{bmatrix}$$

Column sampling

$$\mathbf{S} \approx U\left(\begin{bmatrix} \mathbf{S}_{11} \\ \mathbf{S}_{21} \end{bmatrix}\right) \Lambda \left(\begin{bmatrix} \mathbf{S}_{11} \\ \mathbf{S}_{21} \end{bmatrix}\right) U\left(\begin{bmatrix} \mathbf{S}_{11} \\ \mathbf{S}_{21} \end{bmatrix}\right)^{\top}$$

Previous theoretical results have focused on the accuracy of these approximations and the way to randomly select the indices

What we actually want...

We really want $\mathbf{U}$, $\mathbf{V}$, and Λ. Then we can get the principal components, principal coordinates, and the amount of variance explained.

It turns out that there are quite a few ways to use these two methods to get the things we want.

Let

$$L(\mathbf{S}) = \begin{bmatrix} \mathbf{S}_{11} \\ \mathbf{S}_{21} \end{bmatrix} \qquad L(\mathbf{Q}) = \begin{bmatrix} \mathbf{Q}_{11} \\ \mathbf{Q}_{21} \end{bmatrix}$$

Lots of approximations

After some reasonable algebra...

Quantity of interest	Label	Approximations
$\mathbf{V}$	$\mathbf{V}_{nys}$	$L(\mathbf{S})\mathbf{V}(\mathbf{S}_{11})\Lambda(\mathbf{S}_{11})^{\dagger}$
	$\mathbf{V}_{cs}$	$\mathbf{U}(L(\mathbf{S}))$
$\mathbf{U}$	$\mathbf{U}_{nys}$	$L(\mathbf{Q})\mathbf{V}(\mathbf{Q}_{11})\Lambda(\mathbf{Q}_{11})^{\dagger}$
	$\mathbf{U}_{cs}$	$\mathbf{U}(L(\mathbf{Q}))$
	$\widehat{\mathbf{U}}_{nys}$	$\mathbf{X}\mathbf{V}_{nys}\Lambda_{nys}^{\dagger/2}$
	$\widehat{\mathbf{U}}_{cs}$	$\mathbf{X}\mathbf{V}_{cs}\Lambda_{cs}^{\dagger/2}$
	$\widehat{\mathbf{U}}$	$\mathbf{U}(\mathbf{x}_1)$

Lots of approximations

After some reasonable algebra. . .

Quantity of interest	Label	Approximations
$\mathbf{V}$	$\mathbf{V}_{nys}$	$L(\mathbf{S})\mathbf{V}(\mathbf{S}_{11})\Lambda(\mathbf{S}_{11})^{\dagger}$
	$\mathbf{V}_{cs}$	$\mathbf{U}(L(\mathbf{S}))$
$\mathbf{U}$	$\mathbf{U}_{nys}$	$L(\mathbf{Q})\mathbf{V}(\mathbf{Q}_{11})\Lambda(\mathbf{Q}_{11})^{\dagger}$
	$\mathbf{U}_{cs}$	$\mathbf{U}(L(\mathbf{Q}))$
	$\widehat{\mathbf{U}}_{nys}$	$\mathbf{X}\mathbf{V}_{nys}\Lambda_{nys}^{\dagger/2}$
	$\widehat{\mathbf{U}}_{cs}$	$\mathbf{X}\mathbf{V}_{cs}\Lambda_{cs}^{\dagger/2}$
	$\widehat{\mathbf{U}}$	$\mathbf{U}(\mathbf{x}_1)$

Lots of approximations

After some reasonable algebra...

Quantity of interest	Label	Approximations
$\mathbf{V}$	$\mathbf{V}_{nys}$	$L(\mathbf{S})\mathbf{V}(\mathbf{S}_{11})\Lambda(\mathbf{S}_{11})^{\dagger}$
	$\mathbf{V}_{cs}$	$\mathbf{U}(L(\mathbf{S}))$
$\mathbf{U}$	$\mathbf{U}_{nys}$	$L(\mathbf{Q})\mathbf{V}(\mathbf{Q}_{11})\Lambda(\mathbf{Q}_{11})^{\dagger}$
	$\mathbf{U}_{cs}$	$\mathbf{U}(L(\mathbf{Q}))$
	$\widehat{\mathbf{U}}_{nys}$	$\mathbf{X}\mathbf{V}_{nys}\Lambda_{nys}^{\dagger/2}$
	$\widehat{\mathbf{U}}_{cs}$	$\mathbf{X}\mathbf{V}_{cs}\Lambda_{cs}^{\dagger/2}$
	$\widehat{\mathbf{U}}$	$\mathbf{U}(\mathbf{x}_1)$

Lots of approximations

After some reasonable algebra…

Quantity of interest	Label	Approximations
$\mathbf{V}$	$\mathbf{V}_{nys}$	$L(\mathbf{S})\mathbf{V}(\mathbf{S}_{11})\Lambda(\mathbf{S}_{11})^{\dagger}$
	$\mathbf{V}_{cs}$	$\mathbf{U}(L(\mathbf{S}))$
$\mathbf{U}$	$\mathbf{U}_{nys}$	$L(\mathbf{Q})\mathbf{V}(\mathbf{Q}_{11})\Lambda(\mathbf{Q}_{11})^{\dagger}$
	$\mathbf{U}_{cs}$	$\mathbf{U}(L(\mathbf{Q}))$
	$\widehat{\mathbf{U}}_{nys}$	$\mathbf{X}\mathbf{V}_{nys}\Lambda_{nys}^{\dagger/2}$
	$\widehat{\mathbf{U}}_{cs}$	$\mathbf{X}\mathbf{V}_{cs}\Lambda_{cs}^{\dagger/2}$
	$\widehat{\mathbf{U}}$	$\mathbf{U}(\mathbf{x}_1)$

Lots of approximations

After some reasonable algebra…

Quantity of interest	Label	Approximations
$\mathbf{V}$	$\mathbf{V}_{nys}$	$L(\mathbf{S})\mathbf{V}(\mathbf{S}_{11})\Lambda(\mathbf{S}_{11})^{\dagger}$
	$\mathbf{V}_{cs}$	$\mathbf{U}(L(\mathbf{S}))$
$\mathbf{U}$	$\mathbf{U}_{nys}$	$L(\mathbf{Q})\mathbf{V}(\mathbf{Q}_{11})\Lambda(\mathbf{Q}_{11})^{\dagger}$
	$\mathbf{U}_{cs}$	$\mathbf{U}(L(\mathbf{Q}))$
	$\widehat{\mathbf{U}}_{nys}$	$\mathbf{X}\mathbf{V}_{nys}\Lambda_{nys}^{\dagger/2}$
	$\widehat{\mathbf{U}}_{cs}$	$\mathbf{X}\mathbf{V}_{cs}\Lambda_{cs}^{\dagger/2}$
	$\widehat{\mathbf{U}}$	$\mathbf{U}(\mathbf{x}_1)$

Notes of interest

-

	Complexity:	
Method	Computational	Storage
Nyström	$O(nl^2 + l^3)$	$O(l^2)$ [$O(nl)$]
Column sampling	$O(lnp + p^2 l)$	$O(pl)$

- Column sampling results in orthogonal **U** and **V**, Nyström doesn't
- Using **Q** is weird. Thus the "hatted **U**" versions.
- $\widehat{\mathbf{U}} = U(\mathbf{x}_1)$ where $\mathbf{X} = \begin{bmatrix} \mathbf{x}_1 & \mathbf{x}_2 \end{bmatrix}$ seems reasonable if we think that only the first l selected covariates matter (gets used in supervised PCA)[2]

[2] Bair, Hastie, Paul, and Tibshirani, *JASA, 2006.*

So what do we use?

Well. . .

It depends. We did some theory which gives a way to calculate how far off your approximation might be. You could use these bounds if you like to use your data and make a choice.

Did some simulations too.

Simulations

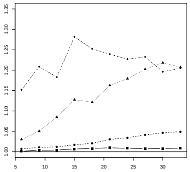
- Draw $\widetilde{X}_i \overset{iid}{\sim} \mathrm{N}_p(0, \Sigma)$ for $n = 5000,\ p = 3000$.
- 4 conditions: $\texttt{Random}_{0.001}$, $\texttt{Random}_{0.01}$, $\texttt{Random}_{0.1}$, and `Band`.
  - $\texttt{Random}_x$ — with probability x and for $i < j$, $\Sigma_{ij}^{-1} = 1$ and 0 otherwise. Diagonal elements are always equal to 1. And symmetrize.
  - `Band` — $\Sigma_{ij}^{-1} = 1$ if $|i - j| \leq 50$ and 0 otherwise.
- The graphical models have approximately $\binom{p}{2} \cdot x$ edges for $\texttt{Random}_x$ and exactly $25(2p - 1 - 50)$ edges for `Band`.

What we report

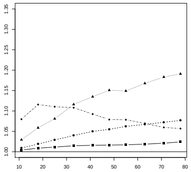
In the **V** case

$$\frac{||\mathbf{V}_{nys,d}(\mathbf{V}_{nys,d}{}^{\top}\mathbf{V}_{nys,d})^{-1}\mathbf{V}_{nys,d}{}^{\top} - \mathbf{V}_d\mathbf{V}_d{}^{\top}||_F}{||\mathbf{V}_{cs,d}\mathbf{V}_{cs,d}{}^{\top} - \mathbf{V}_d\mathbf{V}_d{}^{\top}||_F}.$$

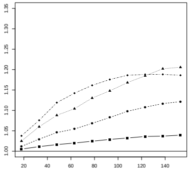
Approximations to V



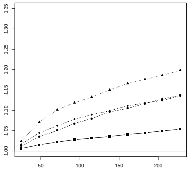
$(d = 2)$



$(d = 5)$



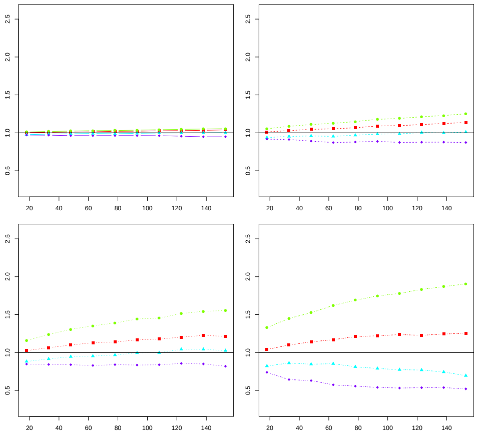
$(d = 10)$



$(d = 15)$

- y-axis, relative performance of Nyström to column sampling.
- Less than 1—better. More than 1—worse.
- x-axis, approximation parameter l (grid based on d).
- `Random`$_{0.001}$ (solid, square), `Random`$_{0.01}$ (dashed, circle), `Random`$_{0.1}$ (dotted, triangle), and `Band`(dot-dash, diamond).
- d small, computation time is similar. d large, Nyström is 10–15% of column sampling

Approximations to $\mathbf{U}$ $(d = 10)$



- y-axis, performance relative to $\widehat{\mathbf{U}}_{cs}$
- x-axis, approximation parameter l
- $\widehat{\mathbf{U}}$, $\mathbf{U}_{nys}$, $\widehat{\mathbf{U}}_{nys}$, $\widehat{\mathbf{U}}_{cs}$
- Clockwise from top left: $\texttt{Random}_{0.001}$, $\texttt{Random}_{0.01}$, $\texttt{Band}$,$\texttt{Random}_{0.1}$

Conclusions

- For computing $\mathbf{V}$, CS beats Nyström in terms of accuracy, but is much slower for similar choices of the approximation parameter and d large.
- For computing $\mathbf{U}$, the naïve methods are bad, better to approximate $\mathbf{V}$ and multiply, so see above
- $\widehat{\mathbf{U}}$ really stinks. This is used for supervised PCA. Future research (don't steal, this one's ours)
- Other choices of Φ (also future research)
- Thanks NSF for funding us.
- For more info, see the paper. Contains boring theory, more extensive simulations, and application to Enron email dataset. (resubmitted to JCGS)