

10_pca

February 22, 2023

1 Principal Component Analysis

variationalform <https://variationalform.github.io/>

Just Enough: progress at pace <https://variationalform.github.io/>

<https://github.com/variationalform>

Simon Shaw <https://www.brunel.ac.uk/people/simon-shaw>.

This work is licensed under CC BY-SA 4.0 (Attribution-ShareAlike 4.0 International)

Visit <http://creativecommons.org/licenses/by-sa/4.0/> to see the terms.

This document uses python

and also makes use of LaTeX

in Markdown

1.1 What this is about:

The connection between SVD, the **Singular Value Decomposition**, and PCA, **Principal Component Analysis**.

As usual our emphasis will be on *doing* rather than *proving*: *just enough: progress at pace*

1.2 Assigned Reading

For this worksheet you are recommended Chapters 4 and 10 of [MML], Chapter 10 of [MLFCES], Chapter 5.3 of [IPDS],

- MML: Mathematics for Machine Learning, by Marc Peter Deisenroth, A. Aldo Faisal, and Cheng Soon Ong. Cambridge University Press. <https://mml-book.github.io>.
- MLFCES: Machine Learning: A First Course for Engineers and Scientists, by Andreas Lindholm, Niklas Wahlström, Fredrik Lindsten, Thomas B. Schön. Cambridge University Press. <http://smlbook.org>.
- IPDS: Introduction to Probability for Data Science, by Stanley H. Chan, <https://probability4datascience.com>

These can be accessed legally and without cost. **NOTE:** we haven't referred to the second of these before.

There are also these useful references for coding:

- PT: python: <https://docs.python.org/3/tutorial>
- NP: numpy: <https://numpy.org/doc/stable/user/quickstart.html>
- MPL: matplotlib: <https://matplotlib.org>

1.3 Review

We have seen these

- Eigenvalue decomposition
- SVD, the **Singular Value Decomposition**

Let's review them...

1.4 Eigen-systems of Symmetric Matrices

Given a real square n -row by n -column matrix, $\mathbf{A} \in \mathbb{R}^{n \times n}$ the eigenvalue problem is that of finding scalar eigenvalues λ and eigenvectors \mathbf{v} such that

$$\mathbf{A}\mathbf{v} = \lambda\mathbf{A} \implies \mathbf{A}\mathbf{V} = \mathbf{V}\mathbf{D} \implies \mathbf{A} = \sum_{k=1}^n \lambda_k \mathbf{v}_k \mathbf{v}_k^T.$$

The eigensystem is **real**.

We have the *Spectral Theorem* - see [MML, Theorem 4.15]

Spectral Theorem (for matrices) If \mathbf{A} is real and symmetric then its eigenvalues are all real and its eigenvector matrix \mathbf{V} can be taken as *orthogonal* so that $\mathbf{V}^{-1} = \mathbf{V}^T$. Hence...

$$\mathbf{A} = \mathbf{V}\mathbf{D}\mathbf{V}^T$$

1.5 The SVD: Singular Value Decomposition

Given a real m -row by n -column matrix, $\mathbf{B} \in \mathbb{R}^{m \times n}$

$$\mathbf{B} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T = \sum_{j=1}^p \sigma_j \mathbf{u}_j \mathbf{v}_j^T$$

where: $\mathbf{U} \in \mathbb{R}^{m \times m}$, $\mathbf{\Sigma} \in \mathbb{R}^{m \times n}$, $\mathbf{V} \in \mathbb{R}^{n \times n}$ and $p = \min\{m, n\}$.

Note that $\mathbf{\Sigma} = \text{diag}(\sigma_1, \dots, \sigma_p) + \text{'zeros'}$, and we can always arrange that $0 \leq \sigma_1 \leq \dots \leq \sigma_p$.

As \mathbf{B} is real, \mathbf{U} and \mathbf{V} are real and *orthogonal*.

If $\sigma_r \neq 0$ and $\sigma_p = 0$ for all $p > r$ then r is the rank of \mathbf{B} .

1.6 How are these factorizations connected?

On the face of it they are very different. the first applies only to square symmetric matrices, while the second applies also to rectangular, and hence (why?) non-symmetric matrices.

But... Look at this... Given the SVD $B = U\Sigma V^T$ we have,

$$B^T B = (U\Sigma V^T)^T U\Sigma V^T$$

and remembering that, in general, $(KL)^T = L^T K^T$ (this could be called *taking the transpose through*), we can write,

$$B^T B = V\Sigma^T U^T U\Sigma V^T = V\Sigma^T \Sigma V^T$$

because $U^T U = I$ (orthogonal).

Similarly, because also $V^T V = I$ (orthogonal),

$$BB^T = U\Sigma V^T (U\Sigma V^T)^T = U\Sigma V^T V\Sigma^T U^T = U\Sigma\Sigma^T U^T.$$

Do you recognise these?

We have just shown that,

$$B^T B = V\Sigma^T \Sigma V^T \quad \text{and} \quad BB^T = U\Sigma\Sigma^T U^T.$$

Familiar? Think about $A = VDV^T$.

- Put $A = B^T B$ (symmetric) and $D = \Sigma^T \Sigma$. Then,

$$B^T B = V\Sigma^T \Sigma V^T \quad \text{becomes} \quad A = VDV^T.$$

- Put $A = BB^T$ (symmetric) and $D = \Sigma\Sigma^T$. Then,

$$BB^T = U\Sigma\Sigma^T U^T \quad \text{becomes} \quad A = UDU^T.$$

- V , the right singular vectors in the SVD are the eigenvectors of $B^T B$.
- U , the left singular vectors in the SVD are the eigenvectors of BB^T .
- In both cases Σ contains the positive square roots of the eigenvalues of $B^T B$ and BB^T .
- **NOTE:** $B^T B$ and BB^T have the same non-zero eigenvalues.

1.7 Why does this matter?

Well, recall the **covariance matrix**, it takes this form:

$$\mathbf{M} = \begin{pmatrix} \text{Var}(X_0) & \text{Cov}(X_0, X_1) & \text{Cov}(X_0, X_2) & \text{Cov}(X_0, X_3) \\ \text{Cov}(X_1, X_0) & \text{Var}(X_1) & \text{Cov}(X_1, X_2) & \text{Cov}(X_1, X_3) \\ \text{Cov}(X_2, X_0) & \text{Cov}(X_2, X_1) & \text{Var}(X_2) & \text{Cov}(X_2, X_3) \\ \text{Cov}(X_3, X_0) & \text{Cov}(X_3, X_1) & \text{Cov}(X_3, X_2) & \text{Var}(X_3) \end{pmatrix}$$

Recall that $\text{Cov}(X, X) = \text{Var}(X)$ and note that $\text{Cov}(X, Y) = \text{Cov}(Y, X)$. This matrix is therefore **symmetric** and so has real eigenvalues.

We have seen that if the data in the X_i 's are already centered and collected into the column vectors \mathbf{X}_i . Then,

$$(N-1)\mathbf{M} = \begin{pmatrix} \mathbf{X}_0 \cdot \mathbf{X}_0 & \mathbf{X}_0 \cdot \mathbf{X}_1 & \mathbf{X}_0 \cdot \mathbf{X}_2 & \mathbf{X}_0 \cdot \mathbf{X}_3 \\ \mathbf{X}_1 \cdot \mathbf{X}_0 & \mathbf{X}_1 \cdot \mathbf{X}_1 & \mathbf{X}_1 \cdot \mathbf{X}_2 & \mathbf{X}_1 \cdot \mathbf{X}_3 \\ \mathbf{X}_2 \cdot \mathbf{X}_0 & \mathbf{X}_2 \cdot \mathbf{X}_1 & \mathbf{X}_2 \cdot \mathbf{X}_2 & \mathbf{X}_2 \cdot \mathbf{X}_3 \\ \mathbf{X}_3 \cdot \mathbf{X}_0 & \mathbf{X}_3 \cdot \mathbf{X}_1 & \mathbf{X}_3 \cdot \mathbf{X}_2 & \mathbf{X}_3 \cdot \mathbf{X}_3 \end{pmatrix} = \begin{pmatrix} \mathbf{X}_0^T \\ \mathbf{X}_1^T \\ \mathbf{X}_2^T \\ \mathbf{X}_3^T \end{pmatrix} \begin{pmatrix} \mathbf{X}_0 & \mathbf{X}_1 & \mathbf{X}_2 & \mathbf{X}_3 \end{pmatrix}$$

2 INCOMPLETE

There is more to come - this document will be replaced with an update in due course.

2.0.1 Review

We covered *just enough*, to make *progress at pace*. We looked at

- How the SVD and eigenvalue decomposition are related.
- How this becomes relevant to the data covariance matrix.
- How this can be used.

Now we can start putting all of this material to work.

2.1 Technical Notes, Production and Archiving

Ignore the material below. What follows is not relevant to the material being taught.

Production Workflow

- Finalise the notebook material above
- Clear and fresh run of entire notebook
- Create html slide show:
 - `jupyter nbconvert --to slides 10_pca.ipynb`
- Set `OUTPUTTING=1` below
- Comment out the display of web-sourced diagrams
- Clear and fresh run of entire notebook
- Comment back in the display of web-sourced diagrams
- Clear all cell output

- Set OUTPUTTING=0 below
- Save
- git add, commit and push to FML
- copy PDF, HTML etc to web site
 - git add, commit and push
- rebuild binder

Some of this originated from

<https://stackoverflow.com/questions/38540326/save-html-of-a-jupyter-notebook-from-within-the-r>

These lines create a back up of the notebook. They can be ignored.

At some point this is better as a bash script outside of the notebook

```
[1]: %%bash
NBROOTNAME=10_pca
OUTPUTTING=1

if [ $OUTPUTTING -eq 1 ]; then
  jupyter nbconvert --to html $NBROOTNAME.ipynb
  cp $NBROOTNAME.html ../backups/$(date +"%m_%d_%Y-%H%M%S")_$NBROOTNAME.html
  mv -f $NBROOTNAME.html ../formats/html/

  jupyter nbconvert --to pdf $NBROOTNAME.ipynb
  cp $NBROOTNAME.pdf ../backups/$(date +"%m_%d_%Y-%H%M%S")_$NBROOTNAME.pdf
  mv -f $NBROOTNAME.pdf ../formats/pdf/

  jupyter nbconvert --to script $NBROOTNAME.ipynb
  cp $NBROOTNAME.py ../backups/$(date +"%m_%d_%Y-%H%M%S")_$NBROOTNAME.py
  mv -f $NBROOTNAME.py ../formats/py/
else
  echo 'Not Generating html, pdf and py output versions'
fi
```

Not Generating html, pdf and py output versions