PCA and SPCA Tutorial

Abstract – This is a tutorial on Principal Component Analysis (PCA) and one of its variants, Standardized PCA (SPCA). Both are techniques for identifying unknown trends in multidimensional data sets.

Keywords: pca, spca, svd, principal component analysis, covariance matrix, correlation matrix
Published: 03-25-2008; Updated: 10-07-2016

Note: This article is part of a legacy series that the author published circa 2008 at http://www.miislita.com, now a search engine site. It is now republished in pdf format here at http://www.minerazzi.com, with its content edited and updated. The original articles can be found referenced in online research publications on IR and elsewhere.

Introduction

In 1901, Pearson developed Principal Component Analysis (PCA), an exploratory technique aimed at identifying unknown trends in multidimensional data sets (Pearson, 1901). In 1933, Hotelling introduced PCA to psychologists, hence sometimes the technique is called Hotelling's Transform (Hotelling, 1933). Nowadays we know that implementing PCA produces the same results as applying the Singular Value Decomposition (SVD) of Golub and Kahan (1965) on the covariance matrix of a data set.

In 1988, Dumais et al. (Dumais, 1988; Deerwester, 1988; Foltz, 1990) applied SVD to information retrieval, specifically to term-term matrices extracted from short texts, not correlation or covariance matrices, and called their technique Latent Semantic Analysis (LSA). When used for indexing purposes, their technique is called Latent Semantic Indexing (LSI).

Despite the publication of PCA tutorials (Roden, 2005; Shlens, 2003; Smith, 2002), under/graduate students are rarely exposed to the algorithm. On the other hand, the fact that SVD can be used in both PCA and LSI (LSA) have induced some to assume, that PCA is LSI. PCA is neither LSI (LSA) nor Factor Analysis (Wikipedia, 2016).

The purpose of this article is to present a straightforward tutorial on PCA. To simplify the calculations, we use several tools easily available online and elsewhere. So there is no excuse for not start learning, or teaching, PCA. We assume that users have some basic knowledge of statistics and matrices.

An Illustrative Example

Figure 1 shows a data set **X**, consisting of three dimensions for 12 nutritionally deficient children. We want to know which variables describe a pattern of changes.

Age, years	Weight, pounds	Height, inches
8	64	57
10	71	59
6	53	49
11	67	62
8	55	51
7	58	50
10	77	55
9	57	48
10	56	42
6	51	42
12	76	61
9	68	57

Figure 1. Multidimensional data set X.

To apply PCA to **X**, we first need to preprocess the data. This is done by computing the mean (μ), standard deviation (σ), and variance (σ^2) of the several data sets and then rearranging these in descending order of (σ^2) values, from left to right. The individual data sets are also centered by removing their means, so we end up with a new data set, **Y**. Figure 2 shows these steps.

X =	Weight	Height	Age	Y=	Weight	Height	Age
	64	57	8		1.25	4.25	-0.83
	71	59	10		8.25	6.25	1.17
	53	49	6		-9.75	-3.75	-2.83
	67	62	11		4.25	9.25	2.17
	55	51	8		-7.75	-1.75	-0.83
	58	50	7		-4.75	-2.75	-1.83
	77	55	10		14.25	2.25	1.17
	57	48	9		-5.75	-4.75	0.17
	56	42	10		-6.75	-10.75	1.17
	51	42	6		-11.75	-10.75	-2.83
	76	61	12		13.25	8.25	3.17
	68	57	9		5.25	4.25	0.17
μ =	62.75	52.75	8.83				
σ =	8.99	6.82	1.90				
σ^2 =	80.75	46.57	3.61				

Figure 2. Data set X and its mean-centered representation, Y.

Rearranging the data sets in descending order of (σ^2) values insures that this ordering is inherited by the PCs, eigenvalues, and eigenvectors to be computed. Next, we compute the transpose of **Y**, $\mathbf{Y}^T$, and the matrix $\mathbf{Y}^T\mathbf{Y}$. Our **Matrix Transposer** tool can do this for you on the fly (http://www.minerazzi.com/tools/matrix-transposer/transposer.php). Just define the number of decimal places for the data and enter the data of **Y,** delimited by a single space. You should be able to generate the $\mathbf{Y}^T\mathbf{Y}$ matrix shown in Figure 3. You may compute $\mathbf{Y}^T\mathbf{Y}$ with Microsoft Excel if you are familiar with its MMULT formula.

888.25	549.25	144.50
549.25	512.25	87.50
144.50	87.50	39.67

Figure 3. Matrix $\mathbf{Y}^T\mathbf{Y}$ generated with the Matrix Transposer tool.

Computing the Covariance Matrix

$\mathbf{Y}^T\mathbf{Y}$ is an array of sum of square deviations that need to be normalized by dividing its elements by **1/(n -1)**, where **n** is the number of observations. In our example **n** = 12.

Multiplying each cell of $\mathbf{Y}^T\mathbf{Y}$ by **1/(n – 1)** yields a new matrix **A**, called the *covariance matrix*. However, the diagonal elements ($i = j$) of **A** are variances σ_{ij}^2 and non-diagonal elements ($i \neq j$) are covariances, $\sigma_i\sigma_j$. Because of this, **A** is also called the *variance-covariance matrix*.

Before proceeding any further, an important note on Excel is necessary. You may use the VAR and COVAR formulas of Excel to get **A**. However, if your EXCEL version uses **n** instead of **n – 1** in the denominator of the covariance, you need to multiply covariances by **n/(n – 1)** as this is the proper normalization for an unbiased estimator. If **n** is small, you can use **1/n**, though.

Figure 4 shows the covariance matrix obtained with Excel, or by dividing Figure 3 results times **1/(n - 1)**. The diagonal elements inherit the variance ordering used in Figure 2.

A =	**Weight**	**Height**	**Age**
Weight	80.75	49.93	13.14
Height	49.93	46.57	7.95
Age	13.14	7.95	3.61

Figure 4. Covariance matrix, A.

A covariance indicates whether changes in, for instance, any two variables, *x* and *y*, move together. Positive covariance means that high values of *y* are associated with high values of *x*. Negative covariance means that high values of *y* are associated with low values of *x*. Zero covariance means that there is no association between *x* and *y*. Thus, Figure 4 suggests, for nutritionally-deficient children, that **Weight-Height** changes (49.93) are more related than **Weight-Age** (13.14) or **Height-Age** (7.95) changes.

As shown in Figure 5, one may reach the same conclusion by transforming the covariance matrix into a distance (dissimilarity) matrix **Δ** using the transformation

$$\delta_{ij}^2 = \sigma_i^2 + \sigma_j^2 - 2\sigma_{ij} \quad (1)$$

Δ =	**Weight**	**Height**	**Age**
Weight	0.00	27.46	58.08
Height	27.46	0.00	34.28
Age	58.08	34.28	0.00

Figure 5. Distance Matrix Δ obtained from covariance matrix A.

The smaller δ_{ij}^2 corresponds to the **Weight-Height** pair; i.e., their changes are more related than the **Weight-Age** or **Height-Age** changes; i.e.

$$\delta_{ij}^2 = 80.75 + 46.57 - 2*49.93 = 27.46;\ \delta_{ij} = 5.2 \quad (2)$$

Computing the PCAs with SVD

To compute principal components, we apply SVD to the covariance matrix, A.

Essentially, **A** is decomposed into three matrices $\mathbf{A = USV^T}$ where **S** is a diagonal matrix that stores the singular values $\lambda_i \ldots \lambda_{i+1} \ldots \lambda_k$. **U** and **V** are orthogonal matrices whose column vectors are, respectively, the so-called *left* and *right* eigenvectors of **A**. $\mathbf{V^T}$ is the transpose of **V**.

These matrices can be computed with an SVD calculator like the one available online from Bluebit (http://www.bluebit.gr/matrix-calculator/). If using their calculator, paste **A** into its textarea field and check *Singular Value Decomposition*. If pasting results from Excel, select *Values are*

delimited by Tabs; otherwise, select the option that goes with the delimiter you are using. Select also the number of decimal digits used; we used 2 decimal digits.

Finally, click the *Calculate* button to generate the **U**, **S**, and $\mathbf{V}^{\mathbf{T}}$ matrices. If using the data from our example, you should be able to reproduce the results shown in Figure 6.

U =	-0.81	0.56	-0.18
	-0.58	-0.82	0.02
	-0.13	0.12	0.98

S =	118.48	0.00	0.00
	0.00	11.03	0.00
	0.00	0.00	1.43

$\mathbf{V^T}$ =	-0.81	-0.58	-0.13
	0.56	-0.82	0.12
	-0.18	0.02	0.98

Figure 6. SVD results obtained from A.

We now do a rank **k** approximation by retaining the first two dominant PCs out of the three possible PCs; thus, **k = 2**. So to perform the Rank 2 approximation, we retain the first two rows (eigenvectors) of $\mathbf{V}^{\mathbf{T}}$ (i.e., first two columns of **V**).

The first column of **V** corresponds to the largest principal component (PC), the second column corresponds to the second largest PC, and so forth. These define the direction where the variability of the original data set is maximized. **V** columns ($\mathbf{V}^{\mathbf{T}}$ rows) should produce the desired linear combinations. Next, we compute **YV** and then plot the first two columns of **YV**. See Figure 7.

V =	-0.81	0.56	-0.18
	-0.58	-0.82	0.02
	-0.13	0.12	0.98
	PC1	**PC2**	**PC3**
YV =	-3.34	-2.87	-0.94
	-10.41	-0.33	-0.17
	10.40	-2.76	-1.15
	-9.03	-4.91	1.59
	7.37	-3.03	0.51
	5.66	-0.64	-1.03
	-12.95	6.32	-1.32
	7.35	0.67	1.07
	11.47	5.13	2.10
	16.04	1.84	-0.96
	-15.85	1.09	0.96
	-6.70	-0.50	-0.67

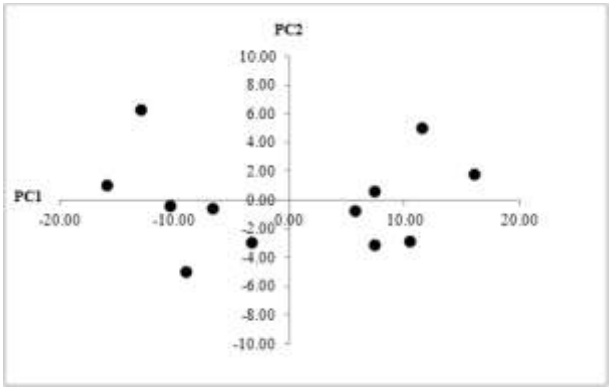

Figure 7. Transformation of the data set and visualization of the two dominant PCs.

The figure shows that when eigenvectors multiply **Y**, their coordinates are shifted and rotated until they end up aligned with vectors, termed now *basis vectors*. Figure 7 indicates several facts that equation (1) cannot reveal.

First and as expected, the data are closer to the first principal component, PC1, which then identifies the direction where the variability of the original data set is maximized. Second, the data changes tend to describe two distinct randomized clusters. To get the old data back, we may compute $\mathbf{YVV}^{\mathbf{T}}$ and add the mean values that were removed to the results.

PCA Shortcomings

PCA is not a silver-bullet approach, but has several shortcomings. For instance, it can fail in, at least, the following cases:

- if the data is non-Gaussian
- if the data is time-dependent

For instance, suppose that we want to apply PCA to images taken from a satellite at different time intervals. If some features change in time, the principal components, as the signal-to-noise ratio, might also change. The computed largest PCs might carry the most important information about scene variations at a given time, but may not carry the information of interest.

Standardized Principal Component Analysis (SPCA)

To overcome some of the above shortcomings, we need to normalize the influence of each variable, enhancing the influence of variables with small variance, and reducing the influence of variables with high variance. In this way, the different time variance patterns can be effectively extracted from the time series. This is can be accomplished with Standardized PCA (SPCA).

SPCA consists in transforming the initial data sets of **X** into z-scores and then applying PCA. This is done by removing the mean from the data sets of X and dividing the results by their standard deviations; i.e., by computing $\boldsymbol{z} = (\boldsymbol{x} - \boldsymbol{\mu})/\boldsymbol{\sigma}$. To compute z-scores, you may use our Standardizer Tool (http://www.minerazzi.com/tools/standardizer/z-scores.php).

In addition to transforming a data set into z-scores, this tool computes all kind of statistics. You can also compute z-scores with Excel, provided that you know how to use its STANDARDIZE formula. The rest of the analysis of SPCA is the same as in PCA. You should be able to end up applying SVD to a *correlation matrix*, though.

To illustrate and for the sake of simplicity, suppose that the data of **X** used in this tutorial was obtained from a process that evolves in time; i.e., from a time series. Figure 8 shows the result of applying SPCA to the data. The **s** subscripts indicate that we are dealing with standardized data.

$\mathbf{Y_sV_s}$ =	$\mathbf{PC1_s}$	$\mathbf{PC2_s}$	$\mathbf{PC3_s}$
	-0.19	0.75	-0.04
	-1.42	0.21	0.05
	1.80	0.68	-0.06
	-1.69	0.06	-0.70
	0.91	0.10	-0.40
	1.09	0.42	0.12
	-1.49	-0.13	0.87
	0.74	-0.57	-0.19
	1.01	-1.55	-0.03
	2.52	-0.01	0.28
	-2.51	-0.35	-0.03
	-0.76	0.39	0.12

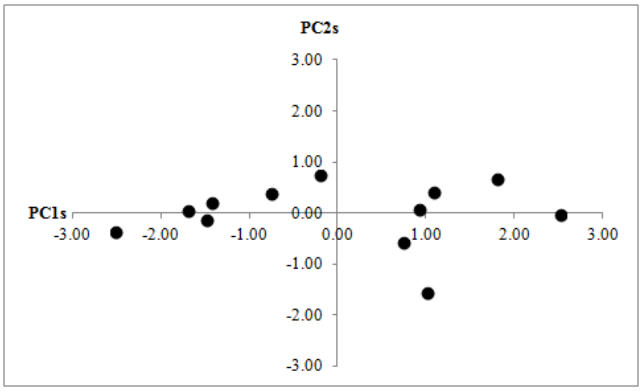

Figure 8. Principal components obtained with SPCA.

Again, the direction where the variability of the original data set is maximized is along the first principal component, $PC1_s$. However, one of the clusters is not random, but describes a linear trend, corresponding to a specific time variance pattern. This trend, detected with SPCA, was hidden from PCA.

Conclusion

Principal Component Analysis (PCA) is a discovery tool designed to identify unknown trends in a multidimensional data set. In PCA we apply SVD to a covariance matrix while in SPCA to a correlation matrix. Contrary to popular opinions, PCA is not LSI, LSA, or Factor Analysis.

In general, using a correlation matrix is recommended over a covariance matrix when the data is time-dependent. It is also useful when variances are rather extreme, when there is a common source of fluctuations, or when different units are used.

References

Furnas, G. W., Deerwester, S., Dumais, S. T., Landauer, T. K., Harshman, R. A. Streeter, L. A., and Lochbaum, K. E. (1988). Information Retrieval using a Singular Value Decomposition Model of Latent Semantic Structure. Retrieved from http://furnas.people.si.umich.edu/Papers/LSI-SIGIR88-p465-furnas.pdf

Deerwester, S., Dumais, S. T., Landauer, T. K., Furnas, G. W., and Beck, L. (1988). Improving information retrieval using Latent Semantic Indexing. Proceedings of the 1988 annual meeting of the American Society for Information Science.

Dumais, S. T., Furnas, G. W., Landauer, T. K., Deerwester, S., and Harshman, R. (1988). Using latent semantic analysis to improve access to textual information Proceedings of the Conference on Human Factors in Computing Systems, CHI. 281-286. Retrieved from http://citeseerx.ist.psu.edu/viewdoc/download;jsessionid=4F9EBA5092B0ADF84CBFD4C020272043?doi=10.1.1.51.5563&rep=rep1&type=pdf See also http://wortschatz.uni-leipzig.de/~sbordag/aalw05/Referate/05_Aehnlichkeit/dumais88using.pdf

Foltz, P. W. (1990). Using Latent Semantic Indexing for Information Filtering. In R. B. Allen (Ed.) Proceedings of the Conference on Office Information Systems, Cambridge, 40-47. Retrieved from http://research.microsoft.com/en-us/um/people/sdumais/lsipapers/filtering-cois.htm

Golub, G. and Kahan, W. Calculating the Singular Values and Pseudo-Inverse of a Matrix. J. SIAM Numer. Anal. Ser. B, Vol 2, 2, pp 205-224. Retrieved from http://web.stanford.edu/class/cme324/classics/golub-kahan.pdf

Hotelling, H., (1933). Analysis of a complex of statistical variable into principal components. J. Educ. Psych., vol. 24, 417-441.

Pearson, K. (1901). On lines and planes of closest fit to systems of points in space, Philosophical Magazine, vol. 2, no. 6, pp. 559-572. Retrieved from http://stat.smmu.edu.cn/history/pearson1901.pdf

Roden, J., Trout, D., and King, B. (2005). A Tutorial on PCA Interpretation using CompClust. Retrieved from http://woldlab.caltech.edu/compclust/pca_interpretation_tutorial.pdf

Shlens, J. (2003). A tutorial on Principal Component Analysis. Retrieved from http://www.cs.princeton.edu/picasso/mats/PCA-Tutorial-Intuition_jp.pdf

Smith, L. (2002). A tutorial on Principal Components Analysis. Retrieved from http://www.cs.otago.ac.nz/cosc453/student_tutorials/principal_components.pdf

Wikipedia (2016). Principal Component Analysis. Retrieved from https://en.wikipedia.org/wiki/Principal_component_analysis