The SVD, LSI and PCA

Stott Parker

UCLA

April 21, 2014

The SVD

The **spectrum** (set of **singular values**) $\sigma(X)$ of an $n \times p$ matrix X is the set of nonnegative square roots of eigenvalues of $X' X$ (a positive definite matrix). Equivalently, the singular values are the eigenvalues of $\sqrt{X' X}$.

Singular values are nonnegative real values, and $\sigma_1(X) \geq \ldots \geq \sigma_p(X) \geq 0$.

> **(SVD)** The **singular value decomposition** of an $n \times p$ matrix X is $X = USV'$ where U, V are unitary and S is an $n \times p$ diagonal matrix where $diag(S) = (\sigma_1, \ldots, \sigma_p)$ are the singular values of X.

Variant conventions for the SVD, with the usual assumption that $n > p$:

- U is $n \times n$, S is $n \times p$.
- U is $n \times p$, S is $p \times p$ (note U is not unitary, but still $X = U\,S\,V'$).

Pseudoinverses

(Pseudoinverse) If $n > p$ and X is an $n \times p$ matrix with SVD $X \;=\; U\;S\;V'$, where $S = diag(\sigma_1, \ldots, \sigma_p)$ is a $p \times p$ matrix, then

$$S^- = diag(\sigma_1^-, \ldots, \sigma_p^-) \quad \text{where} \quad \sigma^- = \begin{cases} 1/\sigma & \text{if } \sigma \neq 0 \\ 0 & \text{otherwise.} \end{cases}$$

is the **pseudoinverse** of S, and

$$X^- \;=\; V\;S^-\;U'$$

is the **pseudoinverse** of X.

Moore-Penrose conditions:

$$\begin{array}{llll} X\;X^-\;X & = & X & \qquad (\,X\;X^-\,)' & = & X\;X^- \\ X^-\;X\;X^- & = & X^- & \qquad (\,X^-\;X\,)' & = & X^-\;X. \end{array}$$

Computation of Pseudoinverses

Clearly we can use the SVD to compute pseudoinverses. But it is common to also define pseudoinverses directly:

$$X^{-} \;\; = \;\; (X'\,X)^{-1}\,X'\,.$$

This is almost always well-defined, except when the inverse does not exist. A fully-general expression can be obtained with a slight modification:

$$X^{-} \;\; = \;\; \lim_{\delta\to 0+}\;(X'\,X\;+\;\delta\,I)^{-1}\,X'\,.$$

This is known as Tikhonov regularization.

Covariance and Correlation Matrices

If X is an $n \times p$ matrix, define:

- $M = \mathrm{diag}(\mu_1, \ldots, \mu_p)$ is the diagonal matrix of X's column means.
- $D = \mathrm{diag}(\sigma_1, \ldots, \sigma_p)$ is the diagonal matrix of standard deviations.
- $\overline{X}$ is the $n \times p$ 'average' of X defined by the matrix $\overline{X} \; = \; 1_{n \times p} \, M$ where $1_{n \times p}$ yields a matrix of ones.

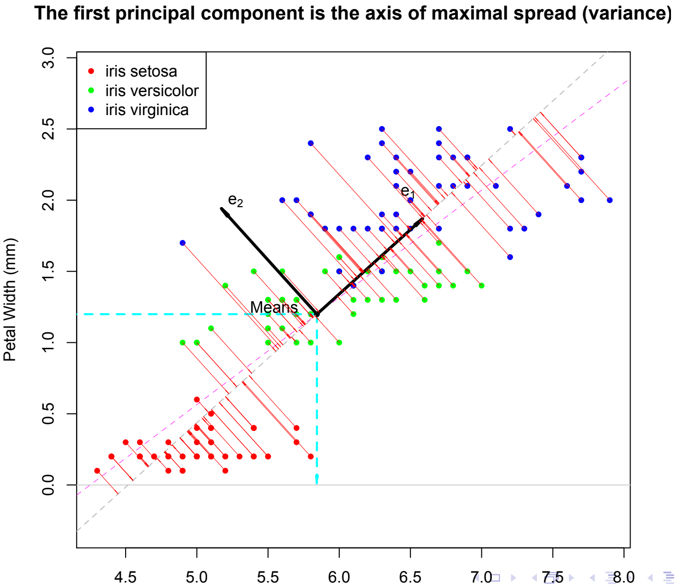
(Covariance and Correlation matrices)

$$\begin{aligned} \mathrm{cov}(X) \quad &= \quad \frac{1}{(n-1)} \, (X - \overline{X})' \, (X - \overline{X}) \\ \mathrm{corr}(X) \quad &= \quad D^{-1} \;\; \mathrm{cov}(X) \;\; D^{-1}. \end{aligned}$$

(Correlation and Z-scores) $\mathrm{corr}(X) \; = \; \mathrm{cov}(Z) \; = \; \mathrm{corr}(Z)$

where Z is the **z-score matrix** $Z = (X - \overline{X}) = (\,(x_{ij} - \mu_j)/\sigma_j\,)$.

Maximum Spread Axis



Maximum Spread Axis

Theorem The first eigenvector of the covariance matrix is the axis of maximum spread in the data. That is, the variance

$$\mathrm{var}(\ (X - \overline{X})\,\mathbf{e}\ )$$

of the projection of the centered points $(X - \overline{X})$ onto the axis $\mathbf{e}$ is maximized when $\mathbf{e}$ is the first eigenvector of $\mathrm{cov}(X)$.

The projections onto the axis are illustrated by the red lines in the previous diagram. The variance of the projected points is greatest along this particular axis — the data spread out most in this direction.

Sphereing (aka Whitening)

(Sphereing) Given a $n \times p$ matrix X,
if $\mathrm{cov}(X) \;=\; V\ S\ V'$ then the **sphereing** of X is

$$(X - \overline{X})\ W \quad \text{where} \quad W \;=\; V\ S^{-1/2}.$$

The covariance matrix of the sphered data is the identity:

$$\begin{aligned}
\mathrm{cov}(\ (X - \overline{X})\ W\ ) \;&=\; ((X - \overline{X})\ W)'\ ((X - \overline{X})\ W)\ /\ \sqrt{n-1} \\
&=\; W'\ \mathrm{cov}(\ X\ )\ W \\
&=\; W'\ \ V\ S\ V'\ \ W \\
&=\; (V\ S^{-1/2})'\ \ V\ S\ V'\ \ (V\ S^{-1/2}) \\
&=\; I.
\end{aligned}$$

In other words, columns of the sphered data are independent, each with mean 0 and variance 1. (Thus the sphered data is akin to 'white noise'.)

Two Ways of looking at the SVD

1. **inner products**

   If $X \;=\; U\,S\,V'$, where S is diagonal, then:

   $$X = (x_{ij}), \quad \text{where} \quad x_{ij} \;=\; \sum_{\ell} u_{i\ell}\, d_{\ell\ell}\, v_{j\ell} \;=\; \mathbf{u}_{row\ i}\, D\, \mathbf{v}_{row\ j}{}'$$

   We can interpret the rows of U as 'mixtures' of scaled component rows in V.

2. **outer products**

   If $X \;=\; U\,S\,V'$, where S is diagonal, then:

   $$X = \sum_{\ell} X_{\ell}, \quad \text{where} \quad X_{\ell} \;=\; \mathbf{u}_{\ell}\, d_{\ell\ell}\, \mathbf{v}_{\ell}{}' \;=\; \mathbf{u}_{col\ \ell}\, d_{\ell\ell}\, \mathbf{v}_{col\ \ell}{}'\,.$$

   When U is $n \times n$ and V is $p \times p$, X_ℓ is a matrix of size $n \times p$.
   The effect is to decompose X into a sum of $n \times p$ matrices, and we can do this iteratively.
   We can interpret the columns of U and V as 'decompositions' of X.

SVD Matrix Approximation

(SVD Approximation) If X is an $n \times p$ matrix, then its *degree-k approximation*

$$X^{(k)} = U^{(k)} \, S^{(k)} \, V^{(k)\prime}$$

is also an $n \times p$ matrix where $U^{(k)}$ and $V^{(k)}$ are the first k columns of U and V, and $S^{(k)} = diag(\sigma_1, \ldots, \sigma^{(k)})$ is the upper left $k \times k$ submatrix of S.

$$X^{(3)} = U^{(3)} \, S^{(3)} \, V^{(3)\prime} = \begin{array}{c} \begin{array}{ccc} \mathbf{u}_1 & \mathbf{u}_2 & \mathbf{u}_3 \end{array} \\ \begin{pmatrix} \times & \times & \times \\ \times & \times & \times \\ \times & \times & \times \\ \vdots & \vdots & \vdots \\ \times & \times & \times \end{pmatrix} \end{array} \begin{pmatrix} \sigma_1 & & \\ & \sigma_2 & \\ & & \sigma_3 \end{pmatrix} \begin{array}{c} \begin{array}{ccc} \mathbf{v}_1 & \mathbf{v}_2 & \mathbf{v}_3 \end{array} \\ \begin{pmatrix} \times & \times & \times \\ \times & \times & \times \\ \times & \times & \times \\ \vdots & \vdots & \vdots \\ \times & \times & \times \end{pmatrix}' \end{array}$$

LSI — Latent Semantic Indexing

In LSI (Latent Semantic Indexing) we have an $n \times p$ data matrix X that represents a relationship between objects denoted by the rows and columns

Ex: *terms & documents, recommenders & movies, jobs & employees.*

With the SVD $X = U\,S\,V'$, LSI considers the k-th degree approximation

$$X^{(k)} \;=\; U\,S^{(k)}\,V' \;=\; U^{(k)}\,S^{(k)}\,V^{(k)'}$$

and represents the row and column objects by coordinates obtained from this decomposition.

Specifically, a standard convention in LSI is to represent:

- the i-th row object by the i-th row of $U^{(k)}\left(S^{(k)}\right)^{-1}$
- the j-th column object by the j-th column of $\left(S^{(k)}\right)^{-1} V^{(k)}$.

M.W. Berry, S.T. Dumais, G.W. O'Brien, 'Using Linear Algebra for Intelligent Information Retrieval', SIAM Review 37:4, 573–595, 1995.

(However, the example in this paper does not do scaling by S^{-1}.)

LSI — Latent Semantic Indexing — in Visualization

When LSI is used in visualization, it is common to use $k = 2$, and use (scaled) eigenvectors to obtain 2D representations for the row and column objects:

$$X^{(2)} \; = \; U^{(2)} \, S^{(2)} \, V^{(2)\prime} \; = \; \begin{array}{c} \begin{array}{cc} \mathbf{u}_1 & \mathbf{u}_2 \end{array} \\ \begin{pmatrix} u_{1\,1} & u_{1\,2} \\ u_{2\,1} & u_{2\,2} \\ u_{3\,1} & u_{3\,2} \\ \vdots & \vdots \\ u_{n\,1} & u_{n\,2} \end{pmatrix} \end{array} \begin{pmatrix} \sigma_1 & \\ & \sigma_2 \end{pmatrix} \begin{array}{c} \begin{array}{cc} \mathbf{v}_1 & \mathbf{v}_2 \end{array} \\ \begin{pmatrix} v_{1\,1} & v_{1\,2} \\ v_{2\,1} & v_{2\,2} \\ v_{3\,1} & v_{3\,2} \\ \vdots & \vdots \\ v_{p\,1} & v_{p\,2} \end{pmatrix}' \end{array}$$

This gives the 2D coordinatization:

$$i\text{-th row object} \quad \longleftrightarrow \quad \left(\frac{u_{i\,1}}{\sigma_1}, \; \frac{u_{i\,2}}{\sigma_2} \right)$$

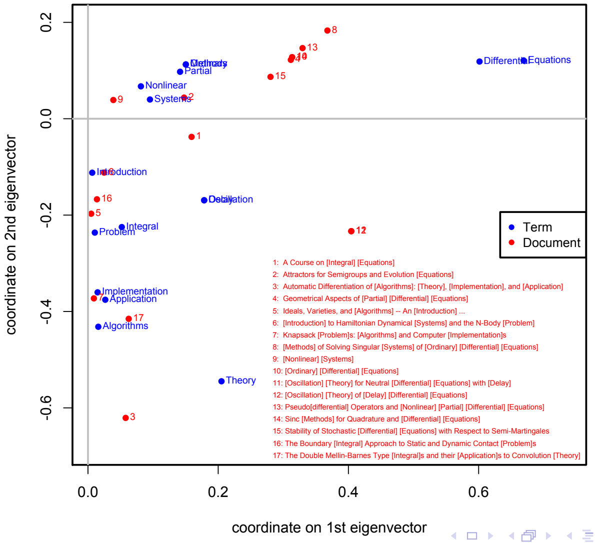
$$j\text{-th column object} \quad \longleftrightarrow \quad \left(\frac{v_{j\,1}}{\sigma_1}, \; \frac{v_{j\,2}}{\sigma_2} \right).$$

(However, scaling conventions vary; e.g., the ‘sphereing’ could be used.)

Example – Term/Document matrix in the LSI paper

```
 Document  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17    |   Term
---------------------------------------------------------------+--------------
           0  0  1  0  1  0  1  0  0  0  0  0  0  0  0  0  0    | Algorithms
           0  0  1  0  0  0  0  0  0  0  0  0  0  0  0  0  1    | Application
           0  0  0  0  0  0  0  0  0  0  1  1  0  0  0  0  0    | Delay
           0  0  0  1  0  0  0  1  0  1  1  1  1  1  1  0  0    | Differential
           1  1  0  1  0  0  0  1  0  1  1  1  1  1  1  0  0    | Equations
           0  0  1  0  0  0  1  0  0  0  0  0  0  0  0  0  0    | Implementation
           1  0  0  0  0  0  0  0  0  0  0  0  0  0  0  1  1    | Integral
           0  0  0  0  1  1  0  0  0  0  0  0  0  0  0  0  0    | Introduction
           0  0  0  0  0  0  0  1  0  0  0  0  0  1  0  0  0    | Methods
           0  0  0  0  0  0  0  0  1  0  0  0  1  0  0  0  0    | Nonlinear
           0  0  0  0  0  0  0  1  0  1  0  0  0  0  0  0  0    | Ordinary
           0  0  0  0  0  0  0  0  0  0  1  1  0  0  0  0  0    | Oscillation
           0  0  0  1  0  0  0  0  0  0  0  0  1  0  0  0  0    | Partial
           0  0  0  0  0  1  1  0  0  0  0  0  0  0  0  1  0    | Problem
           0  0  0  0  0  1  0  1  1  0  0  0  0  0  0  0  0    | Systems
           0  0  1  0  0  0  0  0  0  0  1  1  0  0  0  0  1    | Theory
```

Example – LSI 2D Projection of Terms and Documents



Latent Semantic Indexing of Terms and Documents

1: A Course on [Integral] [Equations]
2: Attractors for Semigroups and Evolution [Equations]
3: Automatic Differentiation of [Algorithms]: [Theory], [Implementation], and [Application]
4: Geometrical Aspects of [Partial] [Differential] [Equations]
5: Ideals, Varieties, and [Algorithms] -- An [Introduction] ...
6: [Introduction] to Hamiltonian Dynamical [Systems] and the N-Body [Problem]
7: Knapsack [Problem]s: [Algorithms] and Computer [Implementation]s
8: [Methods] of Solving Singular [Systems] of [Ordinary] [Differential] [Equations]
9: [Nonlinear] [Systems]
10: [Ordinary] [Differential] [Equations]
11: [Oscillation] [Theory] for Neutral [Differential] [Equations] with [Delay]
12: [Oscillation] [Theory] of [Delay] [Differential] [Equations]
13: Pseudo[differential] Operators and [Nonlinear] [Partial] [Differential] [Equations]
14: Sinc [Methods] for Quadrature and [Differential] [Equations]
15: Stability of Stochastic [Differential] [Equations] with Respect to Semi-Martingales
16: The Boundary [Integral] Approach to Static and Dynamic Contact [Problem]s
17: The Double Mellin-Barnes Type [Integral]s and their [Application]s to Convolution [Theory]

PCA – Principal Components Analysis

(PCA) Principal components are eigenvectors of the covariance matrix. The first principal component is the eigenvector with largest eigenvalue, and so on.

If $\operatorname{cov}(X) = V\,S\,V'$, the first k principal components are columns of

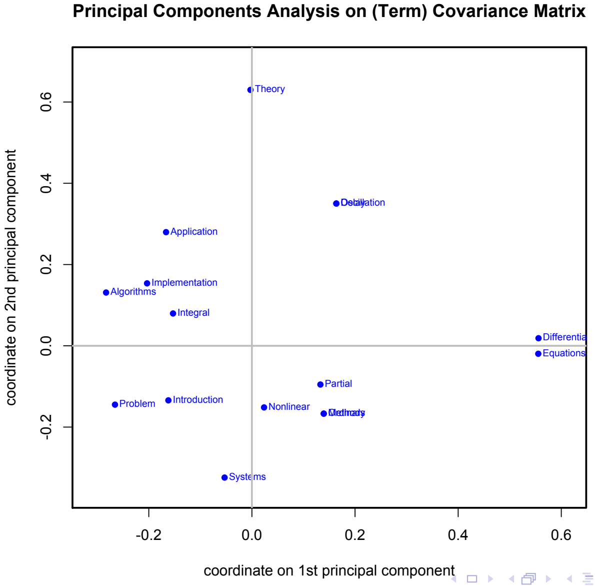
$$V^{(k)} = (\ \mathbf{v}_1\ |\ \mathbf{v}_2\ |\ \cdots\ |\ \mathbf{v}_p\ )$$

We can then project the data X onto V to obtain a k-dimensional representation of the data:

$$\text{projection of } X \text{ onto } V^{(k)} \quad = \quad X\ V^{(k)}.$$

For each row $\mathbf{x}$ in X, this yields a row of coordinates $\mathbf{a} = (a_1, a_2, \ldots, a_k)$.

Example – 2D PCA on the Term Covariance Matrix



Looking at PCA in terms of Approximation

(PCA) Principal components are eigenvectors of the covariance matrix. The first principal component is the eigenvector with largest eigenvalue, and so on.

If $\mathrm{cov}(X) = V\,S\,V'$, columns of V are the principal components of X. With k principal components, we get the approximation

$$\mathbf{x} \;\simeq\; \bar{\mathbf{x}} \;+\; a_1\mathbf{v}_1 \;+\; \cdots \;+\; a_k\mathbf{v}_k$$

Here the coefficients $a_i = \mathbf{x}'\,\mathbf{v}_i$ are the projections of $\mathbf{x}$ on the eigenvectors. With $V^{(k)}$ as the first k principal components, this approximation for $\mathbf{x}$ can be re-written for the entire matrix X by projecting on $V^{(k)}$:

$$X \;\simeq\; \overline{X} \;+\; X\;V^{(k)}.$$

PCA with the Covariance vs. Correlation Matrix

(PCA) Normally: Principal components are eigenvectors of the covariance matrix.
However: The *correlation matrix* is often used instead when the scales of the variables are very different.

With different scales, the variable whose scale is largest is likely to be the axis of maximal spread.

PCA on the correlation matrix is equivalent to PCA on the covariance matrix of the normalized data:

$$\mathrm{corr}(X) \;=\; \mathrm{cov}(Z) \;=\; \mathrm{corr}(Z)$$

where Z is the z-score matrix $Z = (X - \overline{X}) = (\,(x_{ij} - \mu_j)/\sigma_j\,)$.