📖

Linear Algebra unit 6

Date: @today

Lane: *Linear Algebra - statistics*

Topic: Eigenvalues, Singular Value Decomposition, and Principal Component Analysis

What it covers:

- Finding the “natural basis” for a matrix by using eigenvector decomposition
- SVD - an extension to rectangular matrices
- Principal Component Analysis - using eigenvector decomposition to find the highest variance subspaces
- PCA applied to regression for feature selection.

Requires:

Definition of inner product and Vector spaces unit 1, linear independence, matrix operations.

Required by, used by:

Dimensionality Reduction and Transforms

Why another matrix decomposition method?

Simply, the column space of the design matrix may not be full rank, or may have columns that just add noise to the prediction — adding no “effective rank” to it.

We need a way to select valuable columns. One powerful method is to find linear combinations of the existing columns that have the same basis (or approximately the same basis) as the current columns but work better, the sense of having higher variance, since higher variance suggests these are the variables that best explain the outcome.

The *eigenvector decomposition* of a matrix finds a transform to a “natural basis” that has a diagonal matrix for any square, symmetric matrix. In physical terms it is analogous to finding an object’s vibration modes.

The eigenvalue problem

For any square matrix ***A*** there is a *similarity transform* by matrix ***S*** that *diagonalizes* ***A.***

$$\boldsymbol{\Lambda = S^{-1}AS}$$

where the columns of ***S*** are the eigenvectors of ***A***, and Λ has its eigenvalues along its diagonal.

When it is possible (as it is in all cases we consider) to transform the basis for a matrix to be orthogonal, the result is a diagonal matrix. A diagonal matrix just scales any vector it multiplies. This is expressed as $\boldsymbol{Ax} = \lambda x$ where λ is the *eigenvalue* and x is its *eigenvector.* A matrix will often have several distinct eigenvalues that solve this equation, one for each dimension, each defining a sub-space of the x. If the matrix is symmetric and eigenvalues are distinct, then matrix can be diagonalized to find a "natural" basis.

The eigenvalue equation can be solved to give $(\boldsymbol{A} - \lambda \boldsymbol{I})x = 0$. By taking the determinant this expands into a polynomial in λ that can be solved for the set of eigenvalues, which will be real-valued (and possibly *0*) for a symmetric matrix. Then by plugging each eigenvalue back into $\boldsymbol{A} - \lambda \boldsymbol{I}$ one solves for its nullspace to find the corresponding x. Note that the row operations used for LU decomposition *do not* preserve the eigenvalues, so they are not useful here.

Since solving for eigenvalues is equivalent to solving for the roots of a high order polynomial, there is no closed form solution for matrices of dimension 5 or larger. In practice 3 × 3 matrices or larger require numerical solutions.

Approximating eigenvectors

There are several numerical approaches for recovering a matrix's eigenvectors that I cannot cover here. One way to find the eigenvector for the largest eigenvalue is the *power method,* by taking successive powers of ***A***. Start with an arbitrary vector v that can be assumed to be expressed as a linear combination of ***A***'s eigenvectors: $v = \sum_i \lambda_i \boldsymbol{x}_i$. By taking succesive powers of A, $v_{i+1} = \boldsymbol{A}v_i$, the terms increase by λ raised to successive powers, so that in the limit v is dominated by the largest eigenvector.

The SVD matrices

The SVD decomposition is a generalization of eigenvector decomposition to rectangular matrices. Accordingly one can use SVD algorithms to find the eigenvector decomposition, as is commonly the practice.

For any n-row by p-column matrix $\boldsymbol{X}$, one can find a unique factorization as three matrices:

$$\boldsymbol{X = U\Sigma V^T}$$

where U- the *left singular vectors* - is orthonormal n by n, V- *the right singular vectors* - is orthonormal p by p, and $\boldsymbol{\Sigma}$ - containing the *singular values,* is n by p,

with real values along the diagonal and zeros elsewhere. The number of non-zero singular values equals *r,* the "rank" of $\boldsymbol{X}$. Splitting the matrices into their *r* components, the decomposition becomes

$$X = \left[\boldsymbol{U}_1^{mXr}\boldsymbol{U}_2^{mXm-r}\right]\begin{bmatrix}\boldsymbol{S}^{rXr} & \mathbf{0} \\ \mathbf{0} & \mathbf{0}\end{bmatrix}\begin{bmatrix}\boldsymbol{V}_1^{rXn} \\ \boldsymbol{V}_2^{rXn-r}\end{bmatrix}$$

Note that a symmetric matrix A with positive eigenvalues, where $\boldsymbol{A} = \boldsymbol{S}\boldsymbol{\Lambda}\boldsymbol{S}^{-1}$, has an equivalent SVD where $\boldsymbol{U} = \boldsymbol{S}, \Sigma = \Lambda, \boldsymbol{V}^T = \boldsymbol{S}^{-1}$.

SVD Dimensionality Reduction

See Strang p.375

"Singular Value Decomposition" is a foundation for many data-driven analysis techniques. It applies to

- Pseudo inverses (inverses for approximations when exact inverses don't exist) for solving $\boldsymbol{A}x = b$.
- Decomposing noisy collections of vectors by finding a set of basis vectors, similar to Fourier transforms.
- Finding low-rank approximations to compress data, such as for Principal Component Analysis.

Principal Components Analysis

Components Analysis in prediction problems are methods to transform the explanatory features, often to reduce their dimensionality to a smaller set the same "effective dimensionality." In the span of the feature subspace there typically will be a small number of dimensions containing most of its variability. Another way to state this is that the contribution to the "volume" of the determinant of the covariance matrix of the features is immaterial for most features. This can be shown by PCA.

To *maximize* the variance is a subspace, we start with the covariance matrix $\boldsymbol{S}$ projected onto a vector e_i, and set up a optimization problem to maximize the projected variance constraining e_i to unit length.

$$e_i^t\boldsymbol{S}e_i - \lambda_i(1 - e_i^Te_i)$$

This reduces to $\boldsymbol{S}e_i = \lambda_i e_i$ which is the familiar eigenvalue decomposition. Hence by ordering the e_i by decreasing size of λ_i, we can build a subspace of the e_i that maximizes the variance for the subset.

In particular see Steve's explanation (youtube, below) of how the SVD decomposition matrices give the transformation from the raw feature space (the centered design matrix X) to the principal component design matrix T, for the

decomposition of $X = U\Sigma V^T$. The design matrix is multiplied by the transpose of the right singular vectors to get the new subsetted design matrix:

$$T = XV$$

Digression: Regression resources in R

(The cool kids use the R language for statistical computation. It tends to be more robust and complete.)

In R `lm()` is the standard function for ordinary regression. It is not a library (package)

`glm()` is the go to for GLM. both of these are in the stats package. Just google for documentation or do ?lm() at the command line.

You may also find elasticnet helpful https://cran.r-project.org/package=elasticnet

e1071 https://cran.r-project.org/package=e1071. is the standard package for SVMs.

Have a look at the Machine Learning Task View for more packages that may be of interest to you. https://cran.r-project.org/web/views/MachineLearning.html

In general, the CRAN Task Views are the place to start for looking for R resources

---

Assignment

1. Show that two matrices A, B that commute, e.g. $AB = BA$, share the same eigenvectors.
2. If a square matrix has a non-zero nullspace show that it has eigenvectors == 0.
4. As noted, "too many" columns in the regression design matrix can lead to problems and unneeded computation. Alternately adding new features to create new columns has diminishing value. In the worst case adding duplicate columns creates a singular Gram matrix.
    a. What about duplicating rows in the design matrix. How does that affect the regression results?
    b. What is the effect on the prediction if a copy of all the rows is appended to the design matrix?
    c. What is the effect on prediction if a selected set of rows is appended to the design matrix?

References

Youtube lecture on PCA feature selection:

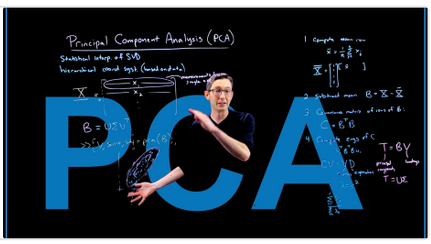
> https://www.youtube.com/watch?v=FD4DeN81ODY

Steve Brunton on PCA:

Principal Component Analysis (PCA)

Principal component analysis (PCA) is a workhorse algorithm in statistics, where dominant correlation patterns are extracted from high-dimensional data.

https://www.youtube.com/watch?v=fkf4IBRSeEc



Readings.

Strang Chapter 5.1, 5.2, 7.3

ISLP - Section 6.3 Dimension Reduction Methods.

Read Brunton, Ch 1.5 - PCA