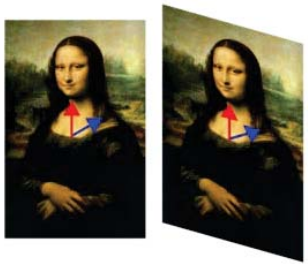
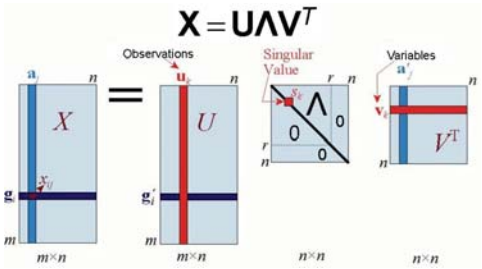
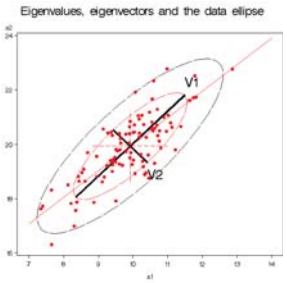




Eigenstuff: Almost everything you ever wanted to know about

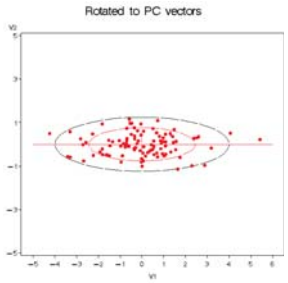
Eigenvalues & eigenvectors

…but were too afraid to ask



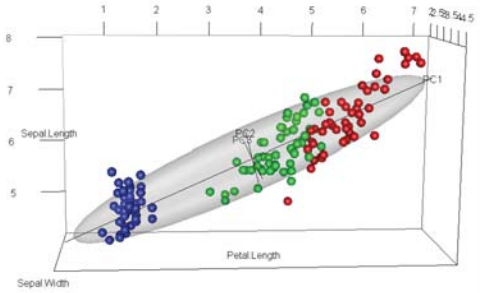
Michael Friendly

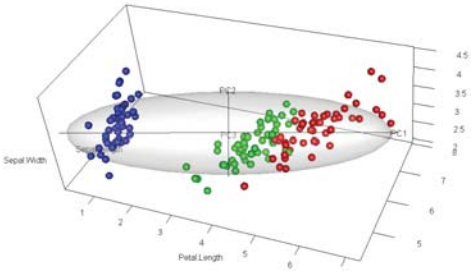
Psychology 6140



Eigenstuff: Statistical motivations

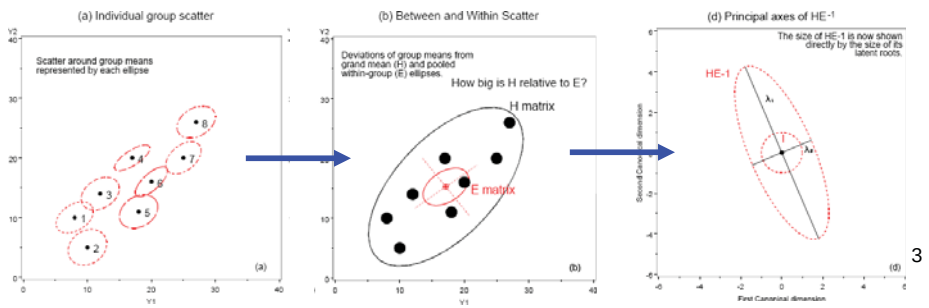
- Usually used in analysis of a covariance (**S**) or correlation matrix (**R**)
  - PCA: principal components are just eigenvectors of **S** or **R**; eigenvalues give the % of variance accounted for
  - FA: similar, but based on a "reduced" **S** or **R**, accounting for only common variance





Eigenstuff: Statistical motivations

- Same ideas are used in low-D approximations to visualize high-D data, models in 2 (or 3) dimensions
- Fundamental for generalizing univariate (F) tests to multivariate tests (e.g., Wilk's Λ, Hotelling trace, …) in MReg & MANOVA
  - Eigenvalues of $\mathbf{H}\,\mathbf{E}^{-1}$ → multivariate test statistics



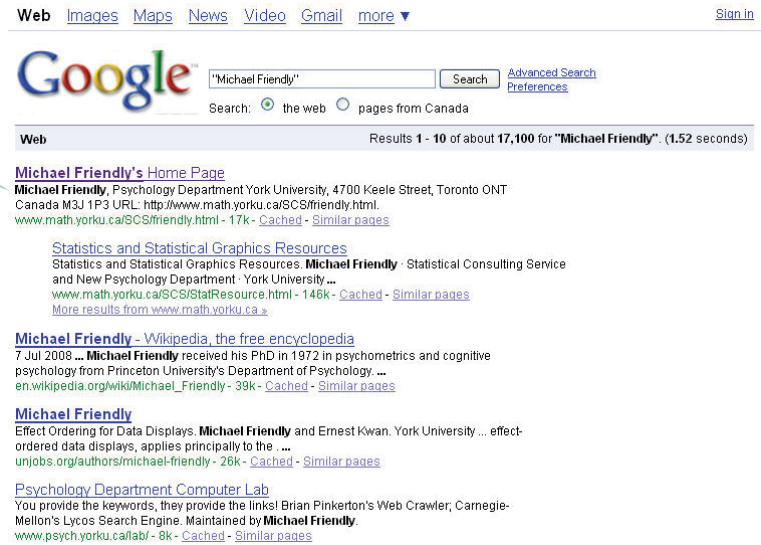
Eigenstuff: Everyday applications

How does Google find me?

At the dawn of the www, there were many search engines

Google became dominant because they recognized that page search had an eigenvalue solution.

Someone called the result the $25 billion eigenvector

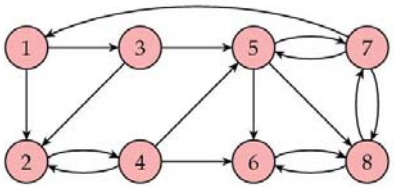


Eigenstuff: Everyday applications

- Google *Page Rank* algorithm (run often)
  - Importance of a page (r_i): a function of
    - # of other pages (L_j) linking to it
    - the importance of each linking page

  } paths thru network

Network diagram of page links



Hyperlink matrix (1/ L_j)

$$H = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 1/3 & 0 \\ 1/2 & 0 & 1/2 & 1/3 & 0 & 0 & 0 & 0 \\ 1/2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1/2 & 1/3 & 0 & 0 & 1/3 & 0 \\ 0 & 0 & 0 & 1/3 & 1/3 & 0 & 0 & 1/2 \\ 0 & 0 & 0 & 0 & 1/3 & 0 & 0 & 1/2 \\ 0 & 0 & 0 & 0 & 1/3 & 1 & 1/3 & 0 \end{bmatrix}$$

But: the WWW has > 30 billion pages! – Do this bit by bit, many robots, many servers

Eigenstuff: Everyday applications

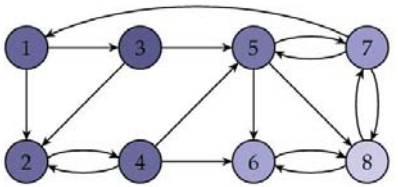
- Start with $\boldsymbol{r}^{(0)} = \mathbf{1}$. Condition for a stable solution:

$$\mathbf{H}\,\boldsymbol{r} = \boldsymbol{r}$$

- $\rightarrow \boldsymbol{r}$ = eigenvector of H assoc. with λ=1 (largest).
- Solution: power method: $\boldsymbol{r}^{(k+1)} = \mathrm{H}\,\boldsymbol{r}^{(k)}$

$$\boldsymbol{r} = \begin{bmatrix} 0.0600 \\ 0.0675 \\ 0.0300 \\ 0.0675 \\ 0.0975 \\ 0.2025 \\ 0.1800 \\ 0.2950 \end{bmatrix}$$

Network diagram shaded by importance



See: http://www.ams.org/featurecolumn/archive/pagerank.html

Lighter = more important

Emerging applications: Eigenfaces

Computer vision & automatic face recognition

Digitize, align & normalize a large # of pics

- Each *m* x *n* pic → *mn* vector of pixel values
- Calculate covariance matrix, **S**, of images
- Eigenvectors of S are "eigenfaces"

Any new face can be approximated by a linear combination of eigen faces

This is one reason why you can't smile in passport photos!

Eigenstuff: Definitions

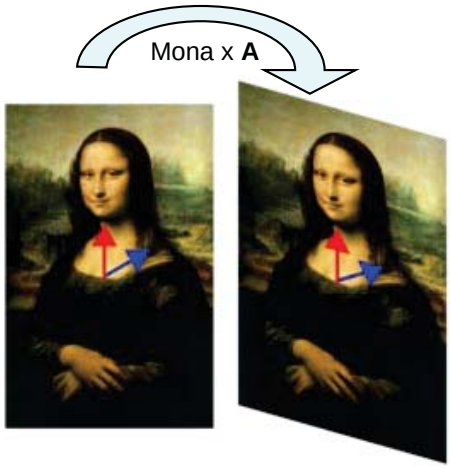
- For a square matrix, $\mathbf{A}_{(n \times n)}$, a vector **v** is an *eigenvector* of **A**, if there exists a scalar, λ, for which

$$\mathbf{A}_{n\times n}\mathbf{v}_{n\times 1} = \lambda \mathbf{v}_{n\times 1} \quad \text{or,} \quad \begin{pmatrix} a_{11} & \cdots & a_{1n} \\ \vdots & \ddots & \vdots \\ a_{n1} & \cdots & a_{nn} \end{pmatrix} \begin{pmatrix} v_1 \\ \vdots \\ v_n \end{pmatrix} = \lambda \begin{pmatrix} v_1 \\ \vdots \\ v_n \end{pmatrix}$$

- The value λ is the corresponding *eigenvalue*
- Other terms: *latent vectors* (**v**), *latent roots* (λ), *characteristic vectors/roots*.

Eigenstuff: Geometric motivation

Linear transformation of Mona Lisa



Eigenvectors arise from what happens to a vector **v** under transformation by a matrix **A**

Here, Mona is subjected to a horizontal shear transform, given by

$$\mathbf{A} = \begin{bmatrix} 1 & k \\ 0 & 1 \end{bmatrix} \longrightarrow \text{Mona x A}$$

The red vector (0,1) is **unchanged** in direction– an **eigenvector** of A

$$\mathbf{A}\,\mathbf{v} = \lambda\,\mathbf{v}$$

The blue vector (1,1) is changed in direction– not an eigenvector

$$\mathbf{A}\,\mathbf{v} \neq \lambda\,\mathbf{v}$$

Eigenstuff: Geometric motivation

For a square matrix $A_{(n\times n)}$, multiplication of a vector, $\underline{v}$ can be regarded as a transformation or mapping of $\underline{v}$ to produce a new vector, $\underline{v}^*$ in an n-dimensional vector space,

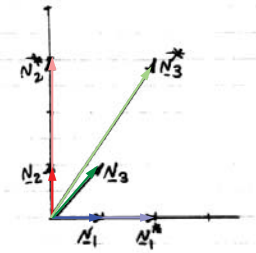
$$A\,\underline{v} = \underline{v}^*$$

e.g. Let $A = \begin{pmatrix} 2 & 0 \\ 0 & 3 \end{pmatrix}$; consider effect on $\underline{v}_1 = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$, $\underline{v}_2 = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$, $\underline{v}_3 = \begin{pmatrix} 1 \\ 1 \end{pmatrix}$

$\underline{v}_1^* = A\,\underline{v}_1 = \begin{pmatrix} 2 & 0 \\ 0 & 3 \end{pmatrix}\begin{pmatrix} 1 \\ 0 \end{pmatrix} = \begin{pmatrix} 2 \\ 0 \end{pmatrix} = 2\underline{v}_1$ ✓

$\underline{v}_2^* = A\,\underline{v}_2 = \begin{pmatrix} 0 \\ 3 \end{pmatrix} = 3\underline{v}_2$ ✓

$\underline{v}_3^* = A\,\underline{v}_3 = \begin{pmatrix} 2 & 0 \\ 0 & 3 \end{pmatrix}\begin{pmatrix} 1 \\ 1 \end{pmatrix} = \begin{pmatrix} 2 \\ 3 \end{pmatrix} \neq k\,\underline{v}_3$ ✗



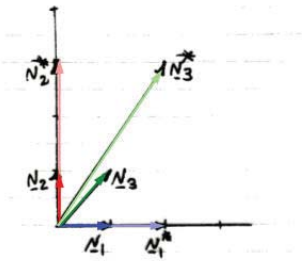
v_1 and v_2 are eigenvectors

v_3 is not

Eigenstuff: Geometric motivation

Thus, $\mathbf{v}_1$ and $\mathbf{v}_2$ are special vectors with respect to the matrix **A**, which are mapped to vectors in the same direction, i.e., for which

$$\mathbf{A}\,\mathbf{v} = \lambda\,\mathbf{v} \qquad \lambda = \text{scalar}$$



When this relation holds:

- **v** is called an eigenvector (latent vector) of the matrix **A**
- λ is the corresponding eigenvalue (latent root)

An *n* x *n* matrix has *n* (eigenvector – eigenvalue) pairs

Finding eigen solutions

- Finding eigenvectors of a matrix naturally gives rise to a set of homogeneous equations, since:

$$\mathbf{A}\mathbf{v} = \lambda\mathbf{v} \Rightarrow \mathbf{A}\mathbf{v} - \lambda\mathbf{v} = \mathbf{0} \Rightarrow \boxed{(\mathbf{A} - \lambda\mathbf{I})\mathbf{v} = \mathbf{0}}$$

- This system has *non-trivial* solutions only for values of λ for which rank (**A** – λ**I**) < n
- This implies (**A** – λ**I**) is singular → $|\mathbf{A} - \lambda\mathbf{I}| = 0$
- This *characteristic equation* → one way to find eigenvalues
- (Modern software uses more efficient methods)

Finding eigen solutions

For a 2 x 2 symmetric matrix, **A**,

$$|\mathbf{A}-\lambda\mathbf{I}| = \left|\begin{pmatrix} a_{11} & a_{12} \\ a_{12} & a_{22} \end{pmatrix} - \begin{pmatrix} \lambda & 0 \\ 0 & \lambda \end{pmatrix}\right| = \begin{vmatrix} a_{11}-\lambda & a_{12} \\ a_{12} & a_{22}-\lambda \end{vmatrix}$$

$$= (a_{11}-\lambda)(a_{22}-\lambda) - {a_{12}}^2$$
$$= \lambda^2 - (a_{11}+a_{22})\lambda + (a_{11}a_{22} - a_{12}^2) = 0$$

This is a quadratic equation that can (usually) be solved for two distinct values, λ_1 & λ_2. Writing the quadratic as

$$a\lambda^2 + b\lambda + c = 0 \Rightarrow \lambda = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

$$b = (a_{11}+a_{22}) = trace(\mathbf{A})$$
$$c = (a_{11}a_{22} - a_{12}^2) = \det(\mathbf{A})$$

Finding eigen solutions

Example: $A = \begin{pmatrix} 1 & -.5 \\ -.5 & 1 \end{pmatrix}$, $\lambda = \frac{1}{2}(1+1 \pm \sqrt{(1-1)^2 + 4(.5)^2})$

$= \frac{1}{2}(2 \pm \sqrt{1}) = \begin{cases} 1.5 = \lambda_1 \\ .5 = \lambda_2 \end{cases}$

To find eigenvector associated with any eigenvalue, solve $(A - \lambda I)\underline{v} = \underline{0}$

eg. for $\lambda_1 = 1.5$, $A - \lambda I = \begin{pmatrix} 1 & -.5 \\ -.5 & 1 \end{pmatrix} - \begin{pmatrix} 1.5 & 0 \\ 0 & 1.5 \end{pmatrix}$

$\Rightarrow \begin{pmatrix} -.5 & -.5 \\ -.5 & -.5 \end{pmatrix} \begin{pmatrix} v_{11} \\ v_{21} \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix} \Rightarrow \underline{v}_1 = \begin{pmatrix} v_{11} \\ v_{21} \end{pmatrix} = \begin{pmatrix} -v_{21}^* \\ v_{21}^* \end{pmatrix}$ eg $\begin{pmatrix} -1 \\ 1 \end{pmatrix}$

↑ rank = 1 so non-trivial sol'n exists

↑ any value

Similarly, for $\lambda_2 = .5$ we find $\underline{v}_2 = \begin{pmatrix} v_{12} \\ v_{22} \end{pmatrix} = \begin{pmatrix} v_{22}^* \\ v_{22}^* \end{pmatrix}$

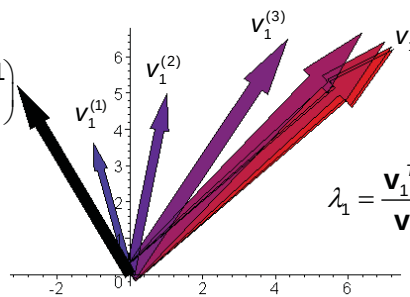
Power method

- The power method is an easy way to find the eigenvector, $\mathbf{v}_1$, corresponding to the *largest* eigenvalue, λ_1, among $\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_n$.
  - Initialize: $\mathbf{v}_1^{(0)} = 1$ (any non-zero vector)
  - Iterate: $\mathbf{v}_1^{(k+1)} = \mathbf{A}\,\mathbf{v}_1^{(k)} / \|\mathbf{A}\,\mathbf{v}_1^{(k)}\|$
  - Until: $\mathbf{v}_1^{(k+1)} \approx \mathbf{v}_1^{(k)}$
  - Then, $\lambda_1 = \frac{\mathbf{v}_1^T \mathbf{A} \mathbf{v}_1}{\mathbf{v}_1^T \mathbf{v}_1}$

e.g.:

$$\mathbf{A} = \begin{pmatrix} 7 & 3 \\ 3 & 6 \end{pmatrix}$$

$$v_1^{(0)} = \begin{pmatrix} -3.1 \\ 5.2 \end{pmatrix}$$



$$\lambda_1 = \frac{\mathbf{v}_1^T \mathbf{A} \mathbf{v}_1}{\mathbf{v}_1^T \mathbf{v}_1} = 9.54$$

This is the idea behind the computation for the Google page rank algorithm

Properties of eigenvalues and eigenvectors

For an *n* x *n* matrix, **A**:

- The characteristic equation $|\mathbf{A} - \lambda\,\mathbf{I}| = 0$
  - has *n* roots, $\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_n$
  - if **A** is symmetric, roots are real (otherwise: complex)

- $\sum_{i=1}^{n} \lambda_i = trace(\mathbf{A})$
- $\prod_{i=1}^{n} \lambda_i = \lambda_1 \times \lambda_2 \times \cdots \times \lambda_n = \det(\mathbf{A})$

Two measures of "size" of **A** in *n* dimensions. In MANOVA:

- Hotelling trace criterion
- Wilks' Lambda

Properties of eigenvalues and eigenvectors

- If any $\lambda_i = 0$, then **A** is singular
  - # non-zero λ_i = rank(**A**) = # dimensions of A space
  - # zero λ_i = # linear dependencies
- If $\lambda_i \neq \lambda_j$, latent vectors are orthogonal & unique
  - i.e., $\mathbf{v}_i^T \mathbf{v}_j = 0$
  - If $\lambda_i = \lambda_j$, vectors not unique, but chosen orthogonal
- Latent vectors are determined in direction, but not length
  - Usually, standardized so that length, $||\mathbf{v}_i||=1$
  - Any scalar multiple, k $\mathbf{v}_i$ also a latent vector
  - → PCA, FA: signs/scaling of loadings are arbitrary

Example: R

$$\mathbf{A} = \begin{bmatrix} 2 & 1 & 1 \\ 1 & 2 & 1 \\ 1 & 1 & 2 \end{bmatrix} \qquad trace(\mathbf{A}) = 6 = \sum \lambda_i \qquad |\mathbf{A}| = 4 = \prod \lambda_i$$

The characteristic equation, $|\mathbf{A} - \lambda \mathbf{I}| = 0$ gives: $\lambda^3 - 6\lambda^2 + 9\lambda - 4 = 0$

The roots, λ, are: 4, 1, 1

Let's try this in R:

```
library(matlib)
(A <- J(3, 3) + diag(3))
tr(A)    # sum of eigenvalues
det(A)   # product of eigenvalues
(ev <- eigen(A))

# eigenvectors are orthogonal
V <- ev$vectors
zapsmall( V %*% t(V) )
```

← eigen() function

Example: R output

```
> library(matlib)
> (A <- J(3, 3) + diag(3))
     [,1] [,2] [,3]
[1,]    2    1    1
[2,]    1    2    1
[3,]    1    1    2

> tr(A)     # sum of eigenvalues
[1] 6

> det(A)    # product of eigenvalues
[1] 4
```

```
> (ev <- eigen(A))
$values
[1] 4 1 1

$vectors
           [,1]   [,2]   [,3]
[1,] -0.5774  0.0000  0.8165
[2,] -0.5774 -0.7071 -0.4082
[3,] -0.5774  0.7071 -0.4082

> # eigenvectors are orthogonal
> V <- ev$vectors
> zapsmall( V %*% t(V) )
     [,1] [,2] [,3]
[1,]    1    0    0
[2,]    0    1    0
[3,]    0    0    1
```

No 0 eigenvalues; r(A)=3

Last 2 eigenvalues are equal, so eigenvectors are not unique

Eigenvectors are orthogonal, V'V=I, and have unit length

Implications

Let $V_{(n \times n)} = (\underline{v}_1, \underline{v}_2, \ldots, \underline{v}_n)$ be matrix of all n eigenvectors corresponding to $\lambda_1 \geq \lambda_2 \geq \lambda_3 \cdots \geq \lambda_n$

Let $\Lambda = \begin{pmatrix} \lambda_1 & & 0 \\ & \lambda_2 & \\ 0 & & \ddots \lambda_n \end{pmatrix}$ - diagonal matrix of eigenvalues

Then

(1) $A\underline{v}_i = \lambda_i \underline{v}_i \Rightarrow AV = V\Lambda$

(2) $\underline{v}_i' \underline{v}_j = 0 \Rightarrow V'V = I$, $VV' = I$ (when $\underline{v}_i$ are scaled so $\|\underline{v}_i\| = 1$)

Here, we are just collecting the $\mathbf{v}_i$ and λ_i in matrices.

But this has interesting applied implications

Implications: Spectral decomposition

(3) **Spectral decomposition**: From (1) and (2), we can write **A** as

$$\mathbf{A} = \mathbf{V}\Lambda\mathbf{V}^T = \begin{bmatrix}\mathbf{v}_1 & \mathbf{v}_2 & \cdots & \mathbf{v}_n\end{bmatrix}\begin{bmatrix}\lambda_1 & & & \\ & \lambda_2 & & \\ & & \ddots & \\ & & & \lambda_n\end{bmatrix}\begin{bmatrix}\mathbf{v}_1^T \\ \mathbf{v}_2^T \\ \vdots \\ \mathbf{v}_n^T\end{bmatrix}$$

or

$$\mathbf{A} = \lambda_1\mathbf{v}_1\mathbf{v}_1^T + \lambda_2\mathbf{v}_2\mathbf{v}_2^T + \cdots + \lambda_n\mathbf{v}_n\mathbf{v}_n^T$$

A sum of rank 1 matrices

Ex:

$$\mathbf{A} = \begin{bmatrix}2 & 1 & 1\\ 1 & 2 & 1\\ 1 & 1 & 2\end{bmatrix} = \frac{4}{-.5774}\begin{pmatrix}1\\1\\1\end{pmatrix}\begin{pmatrix}1 & 1 & 1\end{pmatrix} + \frac{1}{\sqrt{2}}\begin{pmatrix}0\\-1\\1\end{pmatrix}\begin{pmatrix}0 & -1 & 1\end{pmatrix} + \frac{1}{.4082}\begin{pmatrix}2\\-1\\-1\end{pmatrix}\begin{pmatrix}2 & -1 & -1\end{pmatrix}$$

This is the basis for PCA: fewer terms can approximate A

Implications: Matrix powers

(4) Eigenvalues of **matrix inverse & powers**:

$$\mathbf{A}^{-1} = \mathbf{V}\Lambda^{-1}\mathbf{V}^T \quad \text{because, if so:} \quad \mathbf{A}\mathbf{A}^{-1} = \mathbf{V}\Lambda\mathbf{V}^T\ \mathbf{V}\Lambda^{-1}\mathbf{V}^T = \mathbf{I}$$

Hence,

- Eigenvectors of $\mathbf{A}^{-1}$ are the same as those of **A**
- Eigenvalues of $\mathbf{A}^{-1}$ are $1/\lambda_i$

Similarly: $\mathbf{A}^2 = \mathbf{V}\Lambda\mathbf{V}^T\ \mathbf{V}\Lambda\mathbf{V}^T = \mathbf{V}\Lambda^2\mathbf{V}^T$

- So, eigenvalues of $\mathbf{A}^2$ are λ_i^2 and eigenvectors **V** are the same
- This applies to all powers of a matrix

Implications: Orthogonality

(5) Rearranging (3), $\mathbf{A} = \mathbf{V}\,\mathbf{\Lambda}\,\mathbf{V}^T$ gives:

$$\mathbf{V}^T\mathbf{A}\mathbf{V} = \mathbf{V}^T(\mathbf{V}\Lambda\mathbf{V}^T)\,\mathbf{V} = \Lambda = \begin{bmatrix}\lambda_1 & & & 0\\ & \lambda_2 & & \\ & & \ddots & \\ 0 & & & \lambda_n\end{bmatrix}$$

Hence, transforming **A** by **V** and $\mathbf{V}^T$:

gives an orthogonal matrix – uncorrelated variables, which are linear combinations of the original ones; the weights are the eigenvectors in **V**.

In applications here, **A** is usually a covariance matrix, **S**, of a matrix of variables, **Y**. Then,

V S V' = **Λ** is the covariance matrix of the transformed variables, **Y V**. That **Λ** is diagonal means the transformed variables are uncorrelated.

Implications: Maximization

- (6) If $\mathbf{v}_i'\,\mathbf{v}_i = 1$(unit length), then (1) $\mathbf{A}\,\mathbf{v}_i = \lambda_i\,\mathbf{v}_i$ can be written as

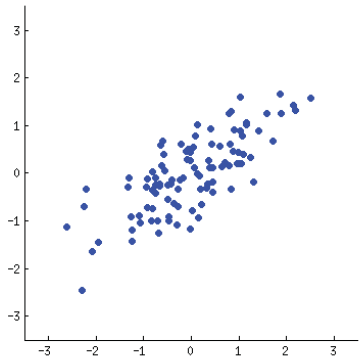
  $$\mathbf{v}_i'\,\mathbf{A}\,\mathbf{v}_i = \lambda_i$$

- But, **v'** **A** **v** is the variance of a linear combination, with weights in **v**.
- Hence, to find a **v** which maximizes **v' A v**, choose $\mathbf{v} = \mathbf{v}_1$ = eigenvector corresponding to largest eigenvalue, λ_1
  - The maximum variance is then λ_1
- In statistical applications, **A** = covariance matrix of a set of variables
  - Many other problems of maximizing (or minimizing) something have eigenvalue – eigenvector solutions!

Maximizing variance

To see what this means, consider the relation between x & y.

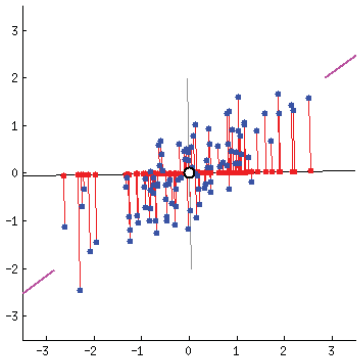
What is the direction along which the variance of the points is greatest?



Maximizing variance

Imagine a vector rotating around the mean of (x,y)

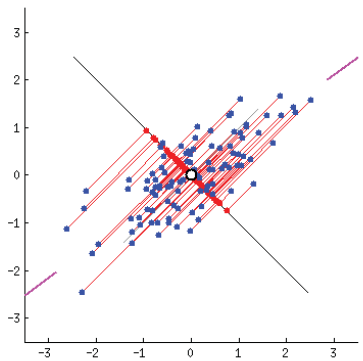
The projection of points on the vector has maximum variance at one position – the eigenvector



Maximizing variance

Now imagine that the points are connected to the vector by springs.

The vector finds its position – the eigenvector – by balancing the forces from the springs! (This is equivalent to a least squares solution for the spring lengths)



Images from ameoba,, http://stats.stackexchange.com/questions/2691/making-sense-of-principal-component-analysis-eigenvectors-eigenvalues

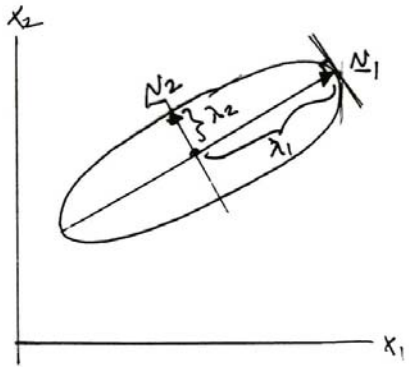
Geometric interpretation

For *n* variables, we can represent data in a space with orthogonal axes

The variables are typically correlated

v' A v = constant represents an ellipse of constant distance from the mean

Usually, $\mathbf{A = S^{-1}}$, the inverse of a covariance matrix

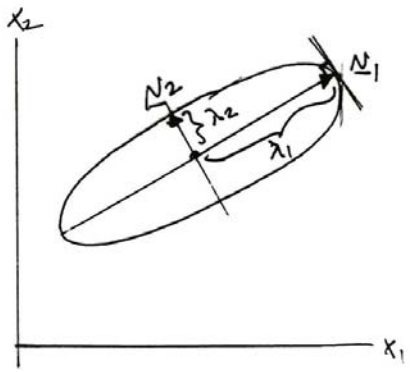


Geometric interpretation

Latent vectors, $\mathbf{v}_1$, $\mathbf{v}_2$ are the major/minor axes of the data ellipse.

They are orthogonal to each other and at right angles to the tangents to the ellipse

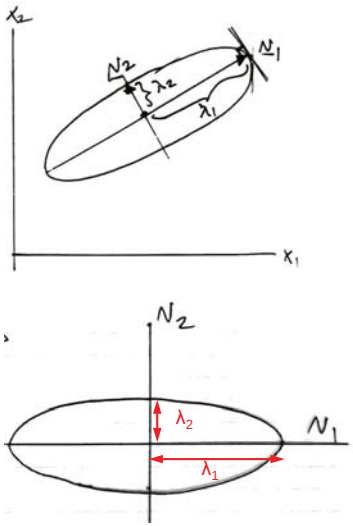
Latent roots λ_1, λ_2 are the half-lengths of the major and minor axes



Geometric interpretation

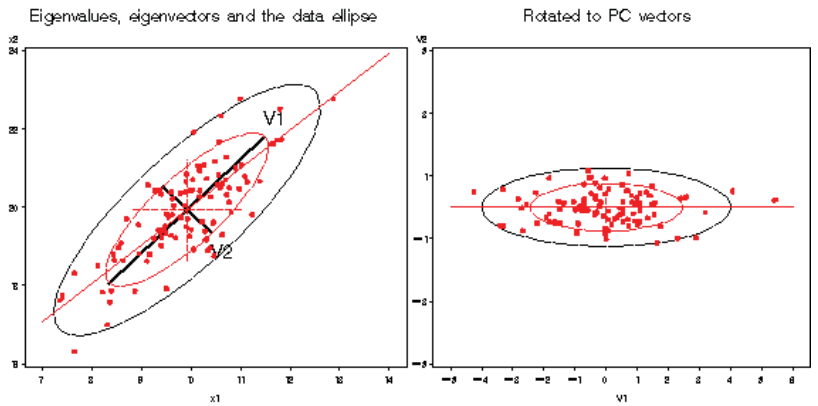
Imagine a rotation of the original space to one where v_1, v_2 become the new coordinate axes

- These vectors span the same space
- They represent **uncorrelated** variables referred to orthogonal axes (principal component scores)
- Total variance of the x_1, x_2, …, x_n variables = $\Sigma \lambda_i$ = total variance of the v_1, v_2, …, v_n



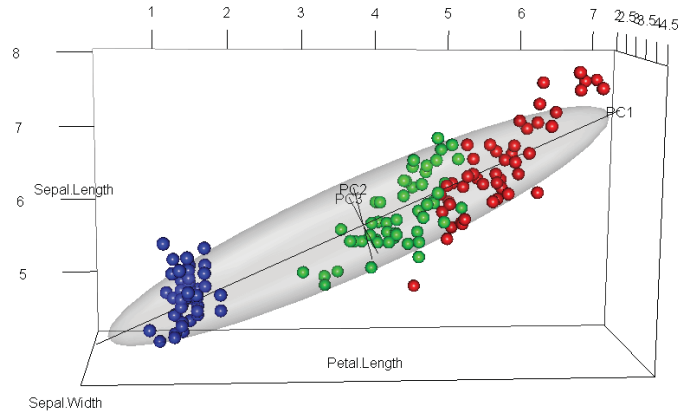
Geometric interpretation

For given data, we can see this by plotting the data and their principal component scores, together with data ellipses



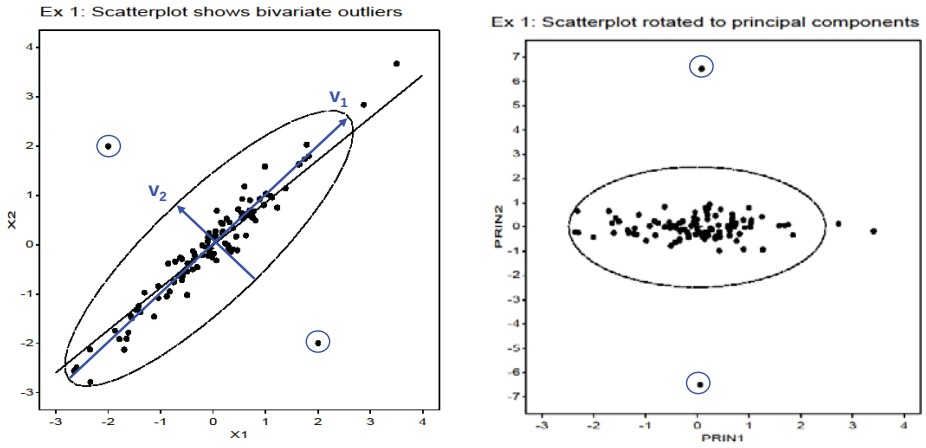
Geometric interpretation

In 3D, with animation, this can be shown more clearly



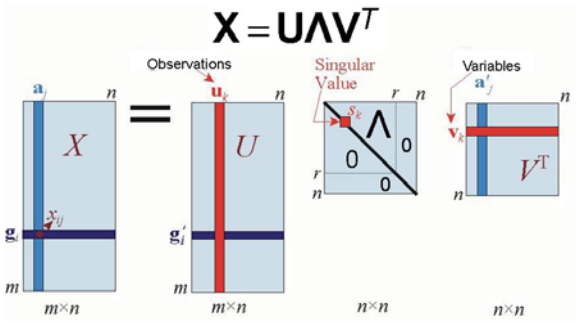
Application: Finding outliers

In multivariate data, outliers are often associated with the **smallest** eigenvalue(s) of covariance matrix, **S**



Singular Value Decomposition & biplots

- Eigenvalues/vectors are usually defined just for square matrices (covariance, correlation)
- For rectangular matrices (i.e., a data matrix), an analogous result is the SVD – also called the *basic structure* of a matrix



Mathematical definition of the SVD

- Let **X** denote an *m* x *n* matrix of data, *rank(***X***)= r, m≥n*
- The singular value decomposition of **X** is:

$$\mathbf{X} = \mathbf{U}\Lambda\mathbf{V}^T$$

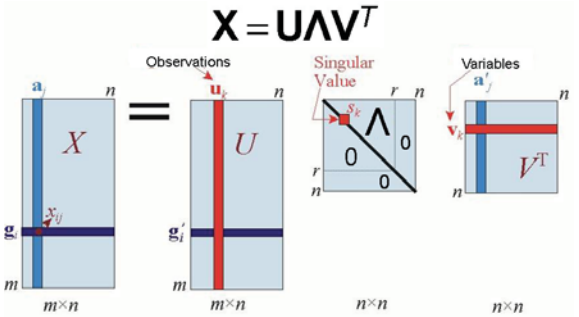
where:

- **U** is an *m* x *n* matrix (observation scores),
- **Λ** is an *n* x *n* diagonal matrix of singular values, and
- $\mathbf{V}^T$ is also an *n* x *n* matrix (variable weights).

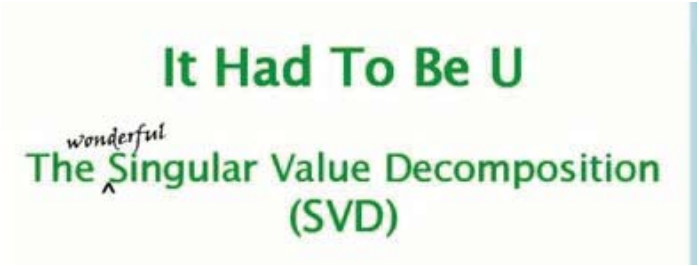
Only the first *r* values of λ_i are non-zero.

Mathematical definition of the SVD

- **V**: columns are the eigenvectors of $\boldsymbol{X}^T\boldsymbol{X}$ and form an orthonormal basis ($\mathbf{V}^T\mathbf{V} = \mathbf{I}$) for the variables
- **Λ**: diagonal, *r* singular values are the square roots of the eigenvalues of both $\boldsymbol{X}\boldsymbol{X}^T$ and $\boldsymbol{X}^T\boldsymbol{X}$
- **U**: columns are the eigenvectors of $\boldsymbol{X}\boldsymbol{X}^T$ and form an orthonormal basis for the observation profiles, so that $\mathbf{U}^T\mathbf{U} = \mathbf{I}$



If that's not clear: the SVD song



http://www.youtube.com/watch?v=JEYLfIVvR9I

Matrix approximation

- Let **X** be an *m* x *n* matrix such that rank(***X***) = *r*
- If $\lambda_1 \geq \lambda_2 \geq ... \geq \lambda_r$ are the singular values of **X**, then $\hat{\mathbf{X}}$, the rank *q* approximation of **X** that minimizes $||\mathbf{X} - \hat{\mathbf{X}}||$, is

$$\hat{\mathbf{X}}_{m\times n} = \sum_{i=1}^{q} \lambda_i \begin{pmatrix} u_{i1} \\ \vdots \\ u_{im} \end{pmatrix} \begin{pmatrix} v_{i1} & \cdots & v_{in} \end{pmatrix} = \lambda_1 u_1 v_1^T + \cdots + \lambda_q u_q v_q^T$$

column scores

row scores

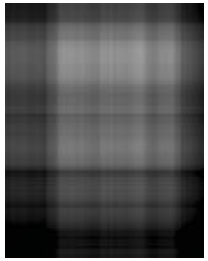
a sum of *q* rank=1 (outer) products. The variance in **X** accounted for each term is λ_1^2

Application: Image compression

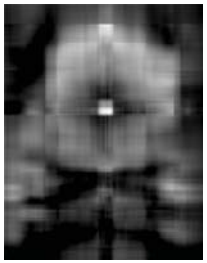
- Digital images often require huge files
- For some purposes, a low-dimensional approx. based on the SVD may be sufficient
- This allows to vary the trade-off between fidelity and file size

color
600 x 465 x 3

grayscale
600 x 465 = 484k

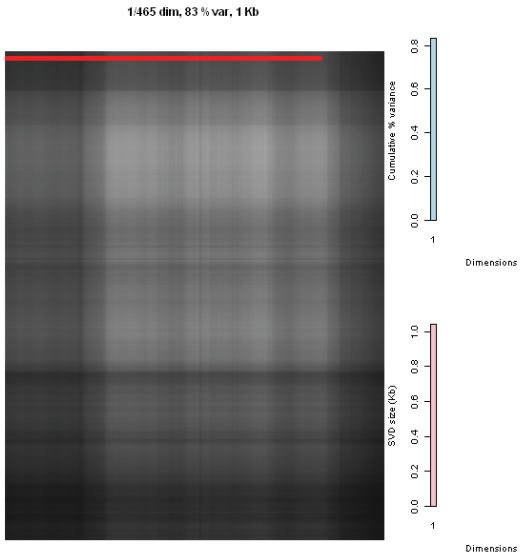


1 dimension
600 + 465 + 1 = 1k



5 dimensions
5(600 + 465 + 1) = 5k

Application: Image compression



Trade-off between image fidelity and file size

At what point is the image recognizable?

File size grows linearly

% variance doesn't seem to be a useful measure of visual fidelity.

Animation produced with the R package, `animation`

Netflix prize: Sparse SVD

- Netflix, Amazon & others use *recommender systems* to advertise, based on your picks and ratings of others
- In 2006, Neflix offered a $1M prize for any algorithm that could beat their own RMSE by 10% or more
- The analysis is based on a huge table of ratings by users (~500,000) by items (~20,000), most of which (~99.9%) is missing.
- In 2009, the grand prize was awarded to the Bellcore team ("Pragmatic Chaos")
- Their algorithm used a variation of the SVD specially designed for huge, thin datasets (+ other methods)
- Today, this problem is a hotbed of research activity in statistics and computer science

Biplot approximation

- A rank 2 (or 3) approximation to the data matrix can be obtained from the first 2 (or 3) singular values λ_i and the corresponding $\mathbf{u}_i$, $\mathbf{v}_i$. e.g.,

$$\mathbf{X} \approx \hat{\mathbf{X}} = \lambda_1 u_1 v_1^T + \lambda_2 u_2 v_2^T$$

- The goodness of fit (variance acct'd for) is

$$(\lambda_1^2 + \lambda_2^2) / \sum_{i=1}^{q} \lambda_i$$

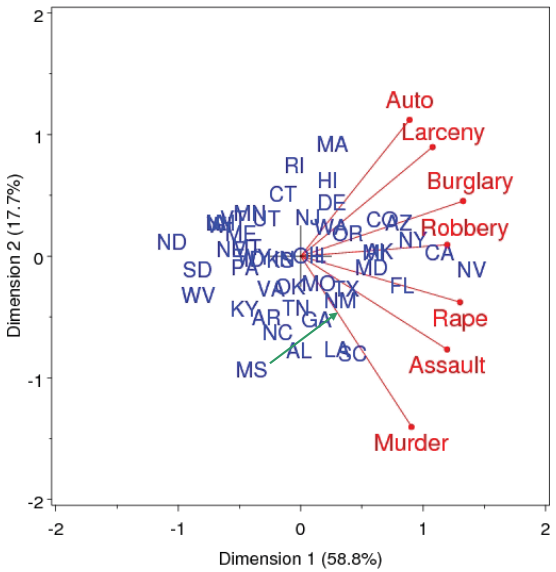
- Biplot is obtained by representing the matrix $\mathbf{X}$ as the product

$$\hat{\mathbf{X}} = (\mathbf{U}\mathbf{\Lambda}^{1/2})(\mathbf{\Lambda}^{1/2}\mathbf{V}^T) = \mathbf{A}\mathbf{B}^T$$

column scores

row scores

Biplot: US crime rates



Dim1: ~ Overall crime rate

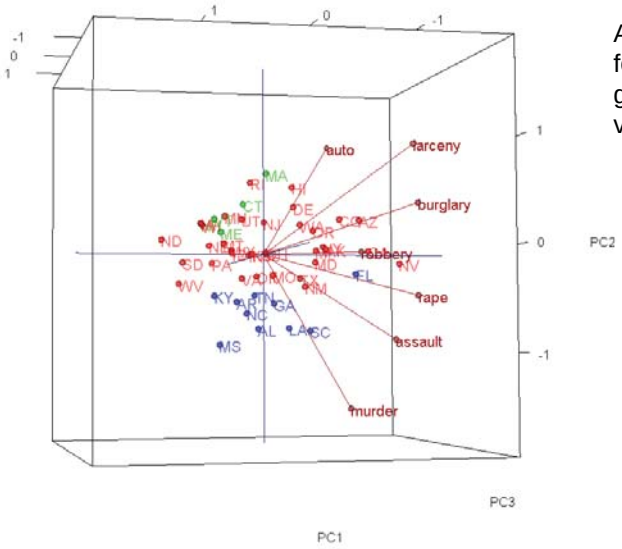
Dim2: Property vs. personal

Note: clusters of southern, New England, western states

This 2D biplot shows 76.5% of total variance, the projection of the data into the 2D space acct'ing for maximum variance.

It gives a low-D summary of 7 variables and 50 observations.

Biplot: US crime rates



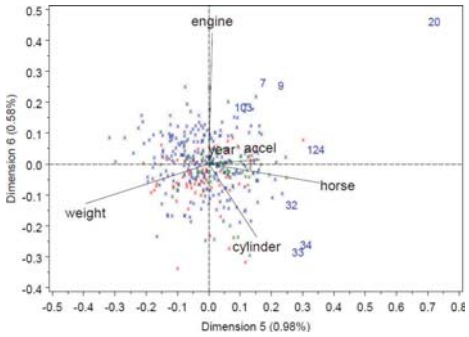
A 3rd dimension accounts for an additional 10.3%, giving 86.8% explained variance.

Application: Collinearity

In multiple regression, collinearity among Xs is associated with the **smallest** eigenvalue(s).

Condition			Proportion of Variation (×100)					
Number	Index	Eigenvalue	Weight	Year	Engine	Horse	Accel	Cylinder
6	10.96	0.03545	18	1	**98**	4	1	**56**
5	8.43	0.05987	**71**	7	0	**66**	49	11

A **collinearity biplot** shows the two *smallest* dimensions in X space, as well as "outliers" (high-leverage points)



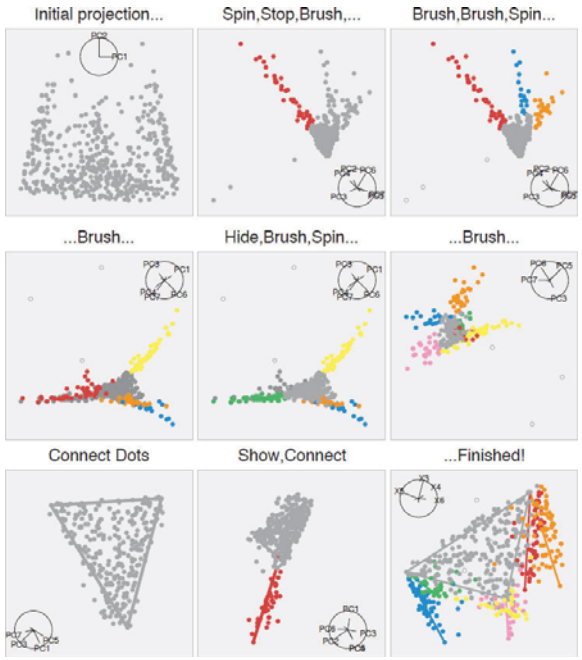


Image from: http://www.ggobi.org/book/2007-infovis/05-clustering.pdf

Exploring high-D data

Dynamic, interactive graphics from ggobi, showing PC-based rotations of 6-D data

Interesting features, clusters are selected ("brushed"), painted (assigned colors)

Each view can show PC or X vectors as frame of reference

Repeat:

{spin, brush, paint}
until {no more clusters}

Summary: Why you should like eigenstuff

- Eigenvalues & vectors distill the "juice" in multivariate data
  - λ_i = size of each *orthogonal* dimension
  - #($\lambda_i \neq 0$) = rank = # *non-zero dimensions*
  - $\mathbf{v}_i$ = *weights of variables* in each linear combination
  - small, non-zero λ_i relate to *outliers & collinearity*
- Many multivariate methods → eigen problems
  - PCA, FA
  - MANOVA, MMreg, discriminant analysis
- Important in visualization of high-D data