Data Analysis, Statistics, Machine Learning

Leland Wilkinson

Adjunct Professor
UIC Computer Science
Chief Scientist
H2O.ai

leland.wilkinson@gmail.com

Reducing

Reducing takes many variables and reduces them to a smaller number of variables

There are many ways to do this

- Principal components (PC) constructs orthogonal weighted composites based on correlations (covariances) among variables
- Multidimensional Scaling (MDS) embeds them in a low-dimensional space based on distances between variables
- Manifold learning projects them onto a low-dimensional nonlinear manifold
- Random projection is like principal components except the weights are random.

Reducing

Principal Components (PC)

PC constructs orthogonal weighted linear combinations

We begin with a matrix of n cases on p variables $\mathbf{X}_{np}$

Centering the columns of $\mathbf{X}$, the covariance matrix is $\mathbf{S}_{pp} = \frac{1}{n}\mathbf{X}'\mathbf{X}$

We want to transform $\mathbf{S}$ using a vector $\mathbf{v}$ such that

$\mathbf{S}\mathbf{v} = \lambda\mathbf{v}$, where λ is a scalar and $\mathbf{v} \neq 0$, so

$$\mathbf{S}\mathbf{v} = \lambda\mathbf{I}\mathbf{v}$$

$$\mathbf{S}\mathbf{v} - \lambda\mathbf{I}\mathbf{v} = 0$$

$$(\mathbf{S} - \lambda\mathbf{I})\mathbf{v} = 0$$

These (homogeneous) equations are solvable if

$$|\mathbf{S} - \lambda\mathbf{I}| = 0$$

Reducing

Principal Components (PC)

$$|\mathbf{S} - \lambda\mathbf{I}| = 0$$

Let

$$\mathbf{S} = \begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix}$$

Then

$$\begin{aligned} (2-\lambda)(2-\lambda) - 1 &= 0 \\ 4 - 4\lambda + \lambda^2 - 1 &= 0 \\ (\lambda - 3)(\lambda - 1) &= 0 \\ \lambda_1 &= 3 \\ \lambda_2 &= 1 \end{aligned}$$

Reducing

Principal Components (PC)

Now plug each λ into homogeneous equations and solve for $\mathbf{v}$

$$\mathbf{v}_1 = \begin{bmatrix} 1 \\ 1 \end{bmatrix} \qquad \mathbf{v}_2 = \begin{bmatrix} -1 \\ 1 \end{bmatrix}$$

Test equations on our covariance matrix

$$\mathbf{S} \quad \mathbf{v} = \lambda \quad \mathbf{v}$$

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

$$\begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix} \begin{bmatrix} -1 \\ 1 \end{bmatrix} = 1 \begin{bmatrix} -1 \\ 1 \end{bmatrix}$$

Reducing

Principal Components (PC)

Each $\mathbf{v}$ is computed from the residuals of the previous $\mathbf{v}$

For the 2x2 case, let's break down the quadratic form here

We end up with the expression for the variance of a linear combination

$$\begin{aligned}\mathbf{v}'\mathbf{S}\mathbf{v} = \lambda &= v_1^2 s_{11} + v_1 v_2 s_{21} + v_2 v_1 s_{12} + v_2^2 s_{22} \\ &= v_1^2 s_{11} + v_2^2 s_{22} + 2 v_1 v_2 s_{12} \\ &= v_1^2 VAR(X_1) + v_2^2 VAR(X_2) + 2 v_1 v_2 COV(X_1, X_2)\end{aligned}$$

$$VAR(a_1 X_1 + a_2 X_2) = a_1^2 VAR(X_1) + a_2^2 VAR(x_2) + 2 a_1 a_2 COV(X_1, X_2)$$

So, variance for each component is associated with an eigenvalue

Reducing

Matrix Forms

$$\mathbf{V} = \begin{bmatrix} v_{11} & v_{12} & \cdots & v_{1p} \\ v_{21} & v_{22} & \cdots & v_{2p} \\ \vdots & \vdots & \vdots & \vdots \\ v_{n1} & v_{n2} & \cdots & v_{np} \end{bmatrix}$$

Eigenvectors

$$\mathbf{\Lambda} = \begin{bmatrix} \lambda_1 & 0 & \cdots & 0 \\ 0 & \lambda_2 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \cdots & \lambda_p \end{bmatrix}$$

Eigenvalues

$$\mathbf{F} = \mathbf{XV}$$

Principal Components (scores)

$$\mathbf{S}_{xx} = \frac{1}{n}\mathbf{X}'\mathbf{X} = \mathbf{V\Lambda V}'$$

Decomposition of covariance matrix

$$\mathbf{S}_{ff} = \mathbf{\Lambda} = \mathbf{V}'\mathbf{S}_{xx}\mathbf{V}$$

Eigenvalues (variances of components)

Reducing

Working with correlation matrix

Standardize columns of $\mathbf{X}$ into $\mathbf{Z}$ (*z* scores)

Then correlation matrix is

$$\mathbf{R}_{zz} = \frac{1}{n}\mathbf{Z}'\mathbf{Z}$$

$$\mathbf{F} = \mathbf{ZV}$$ Principal Components (scores)

$$\mathbf{R}_{zz} = \mathbf{V}\mathbf{\Lambda}\mathbf{V}'$$ Decomposition of correlation matrix

$$\mathbf{R}_{ff} = \mathbf{\Lambda} = \mathbf{V}'\mathbf{R}_{zz}\mathbf{V}$$ Correlation matrix of components (diagonal)

$$\mathbf{R}_{zf} = \mathbf{L} = \mathbf{R}_{zz}\mathbf{V}$$ Correlations of variables with components (loadings)

$$tr(\mathbf{\Lambda}) = tr(\mathbf{R_{zz}}) = p$$ Variance is conserved

$$\frac{\lambda_i}{p}$$ Proportion of total variance accounted for by *i*th PC

Social scientists use $\mathbf{R}$ most of the time

Astronomers, biologists, and others who care about scales of measurement use $\mathbf{S}$

Reducing

The singular value decomposition (SVD)

$$\mathbf{X}_{np} = \mathbf{U}_{nn}\mathbf{D}_{np}\mathbf{V}'_{pp}$$

$$\mathbf{U}'\mathbf{U} = \mathbf{I}_{nn}$$

$$\mathbf{V}'\mathbf{V} = \mathbf{I}_{pp}$$

- Most of the elements of $\mathbf{D}$ are zero
  - Diagonal of $\mathbf{D}$ is nonzero for min(n,p) and contains singular (eigen) values
- Equivalent to finding eigenvectors and eigenvalues of $\mathbf{X}'\mathbf{X}$ and $\mathbf{X}\mathbf{X}'$
- Don't use SVD for computing principal components
  - SVD is a method for specifying row and column eigenvectors
  - But computationally it is a pig in time and space
  - Use it only for sparse matrices and when you need both row and column vectors
  - A lot of research has gone into sparse SVD calculations
  - But these are for specialized applications, not principal components
  - And while many people like expressing PC as SVD, it obscures the model

Reducing

- Tests of significance
  - If $\mathbf{X}$ is multinormally distributed, you can do all sorts of tests
    - Confidence intervals on λ_i
    - Test for how many components to retain
    - Confidence intervals on eigenvectors/loadings
    - But these are for specialized applications, not principal components
      - And they generally rest on asymptotic arguments
  - Why do this???
    - What? You want to start using $\boldsymbol{\Sigma}$ instead of $\mathbf{S}$ and Greek all over the place?
    - $\mathbf{X}$ is almost never multinormally distributed with real data
    - And there is no good test for that
    - Do you really care about significance or are you looking for structure?
    - In any case, your n ought to be large in order to use PC
    - So everything's going to be significant (or not, if you include a ton of variables)

Reducing

Rotating Principal Components

T is a transformation matrix representing a linear map

FT maps principal components to rotated components (scores)

LT maps loadings to rotated loadings

T may be orthogonal or not

Thurstone (1947) criteria

1) each row contains at least one zero;

2) for each column, there are at least as many zeros as there are columns

For any pair of factors,

3) there are some variables with zero loadings on one and large loadings on the other

4) there is a sizable proportion of zero loadings;

5) there is only a small number of large loadings.

Reducing

- Rotating Principal Components (exploratory)
  - Orthogonal $\mathbf{T}$ (orthogonal rotations, meaning $\mathbf{T'T} = \mathbf{I}$ and $\mathbf{TT'} = \mathbf{I}$)
    - Varimax (maximize variance of loadings within columns)
    - Quartimax (maximize variance of loadings within rows)
    - Equamax (halfway between other two)
  - Non-orthogonal $\mathbf{T}$ (oblique rotations)
    - Promax (blah, blah)
    - Oblimin (yada, yada)
  - There are many others
    - Angels on heads of pins
  - Rotated principal components are not principal components
  - Rotated principal components are just another linear projection
    - There is an infinite number of such projections
    - Do this stuff if you're a psychologist and don't know what you're looking for

Reducing

Procrustes Rotations (confirmatory)

Peter Schönemann

We have two matrices $\mathbf{A}$ and $\mathbf{B}$ that are the same size (conformable for addition)

We seek an orthogonal matrix

$$\mathbf{AT} = \mathbf{B} + \mathbf{E} \text{ , such that}$$
$$tr(\mathbf{E}'\mathbf{E}) = minimum \text{ , and}$$
$$\mathbf{T}'\mathbf{T} = \mathbf{I}$$

Schönemann's solution (use Jacobi for decomposition)

$$\mathbf{S} = \mathbf{A}'\mathbf{B}$$
$$\mathbf{S}'\mathbf{S} = \mathbf{V}\mathbf{\Lambda}\mathbf{V}'$$
$$\mathbf{S}\mathbf{S}' = \mathbf{W}\mathbf{\Lambda}\mathbf{W}'$$
$$\mathbf{T} = \mathbf{W}\mathbf{V}'$$

Generalized Procrustes (Carroll, Gower, others)

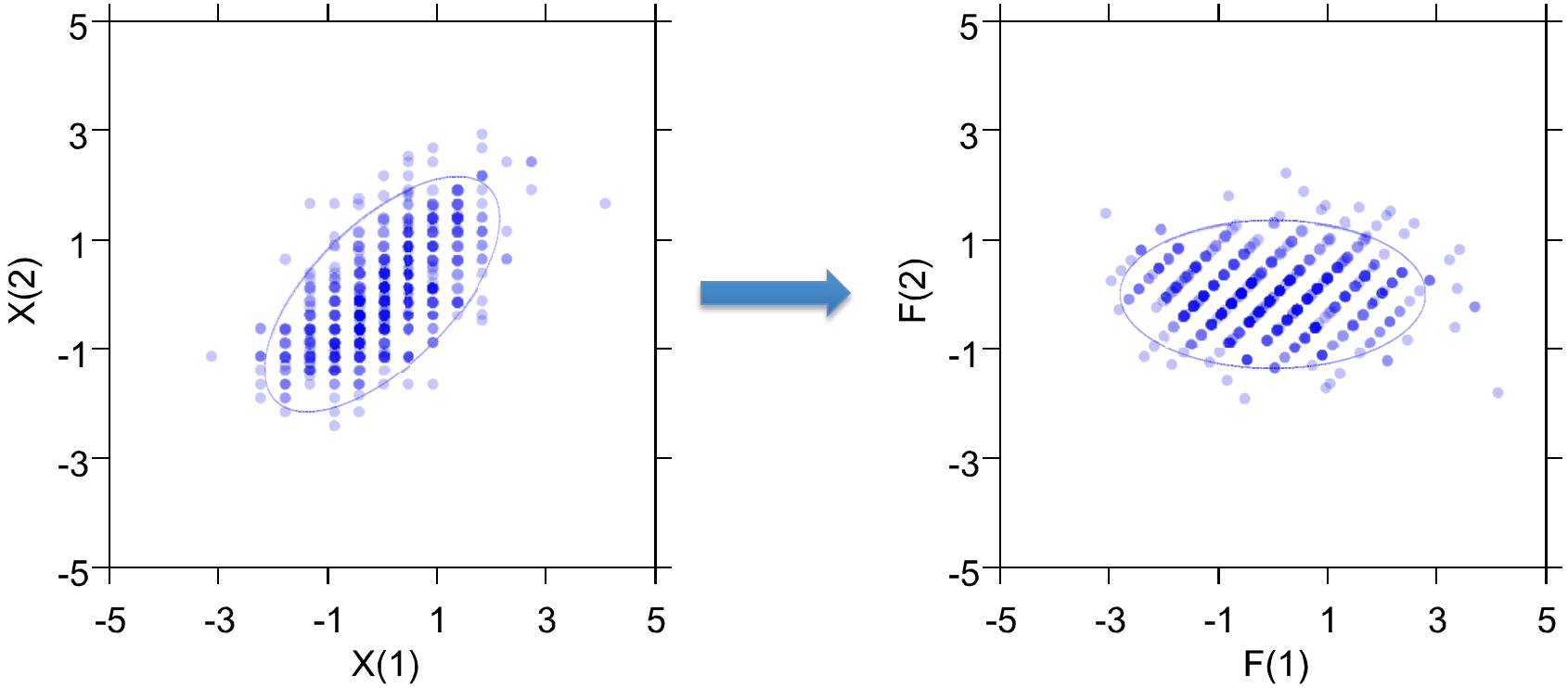
- Translate
- Dilate
- Rotate

Reducing

- Peter Schönemann (a digression)
  - His methods inspired work in geology and morphology
    - matching geological sites
    - matching faces to each other
  - 1. Showed that there is no convincing evidence for Spearman's g
    - This was called "general intelligence"
    - No plausible support for it in correlation matrices
    - Racists continue to do their studies with first principal factor
  - 2. Showed (with Louis Guttman) that factor scores are indeterminate
    - Many researchers blithely continue to use them
  - 3. Revealed shrinkage in heritability coefficients for twins studies
    - Our genetically-oriented world prefers to ignore his arguments
    - The pendulum is starting to swing the other way

Reducing

Principal Components map

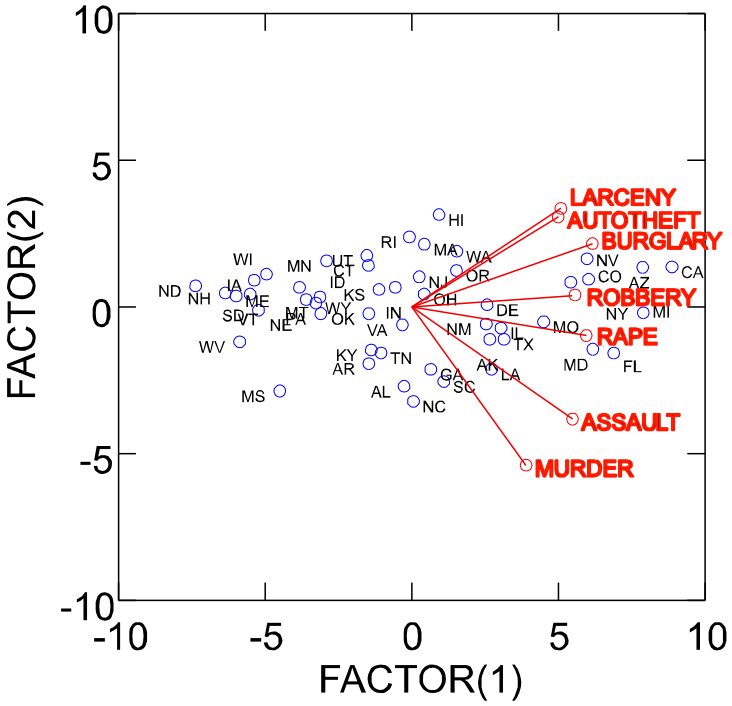


Reducing

Biplots

Plot row and column eigenvectors in same frame

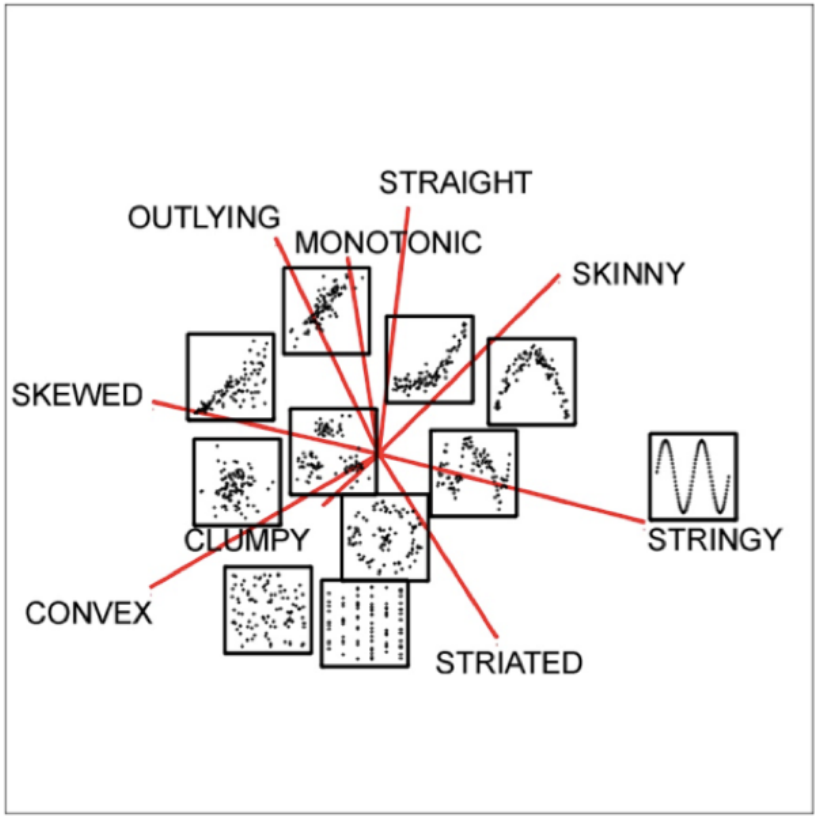
Could use SVD here or do two decompositions



Reducing

Biplots

A biplot of scagnostics (shape descriptors of scatterplots)

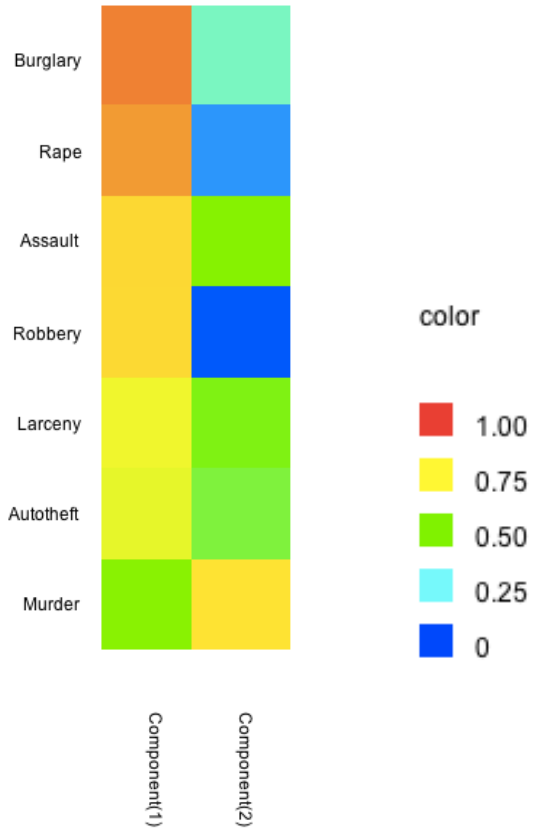


Reducing

Interpreting Principal Components

Heatmap of loadings

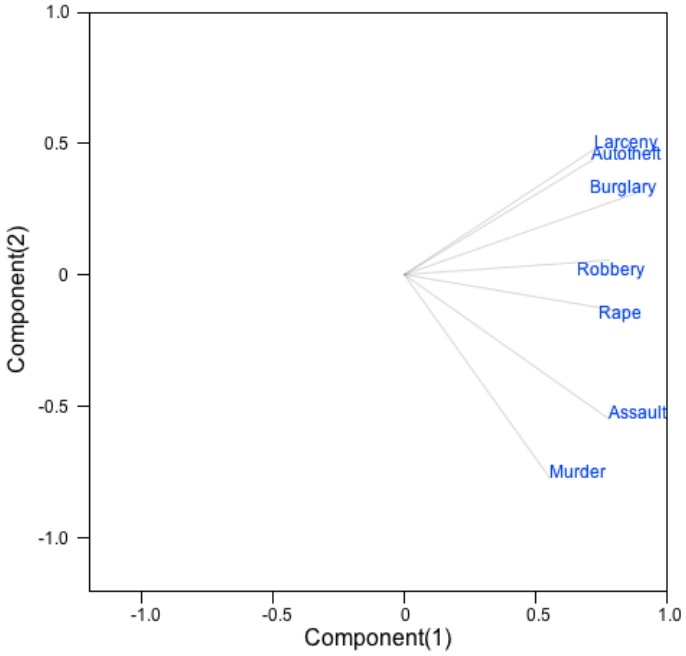
	Component(1)	Component(2)
Burglary	0.881	0.308
Rape	0.851	-0.139
Assault	0.784	-0.546
Robbery	0.782	0.055
Larceny	0.728	0.480
Autotheft	0.714	0.438
Murder	0.557	-0.771



Reducing

Interpreting Principal Components

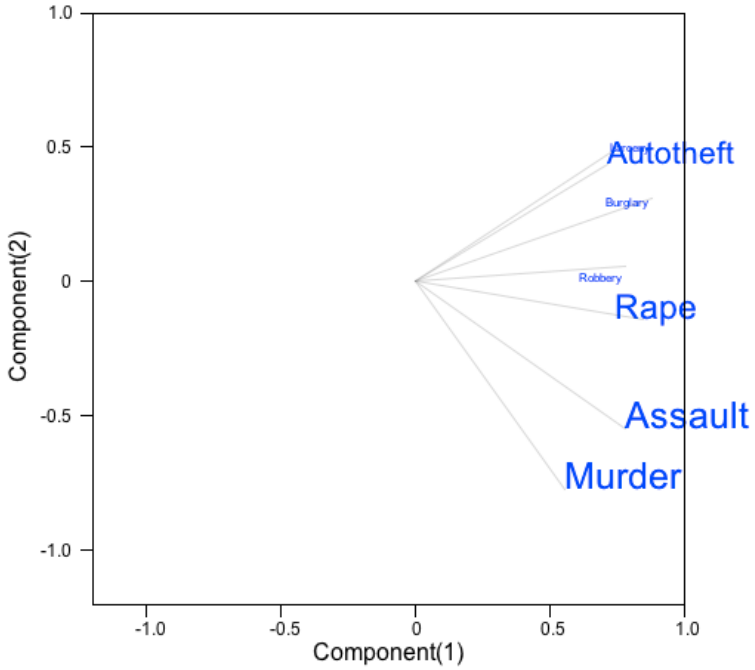
Simple vector plot



Reducing

Interpreting Principal Components

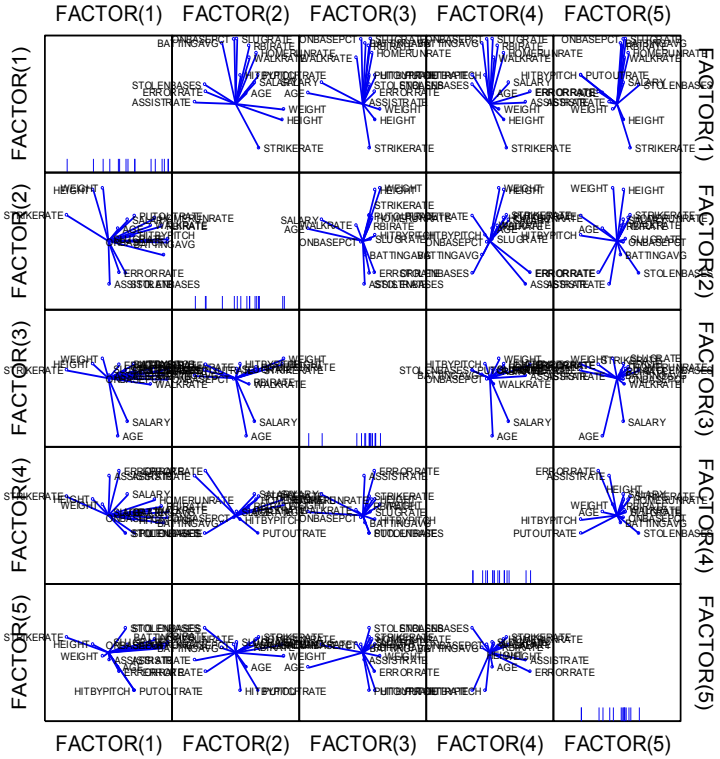
Labels are sized according to badness of fit in 2 dimensions



Reducing

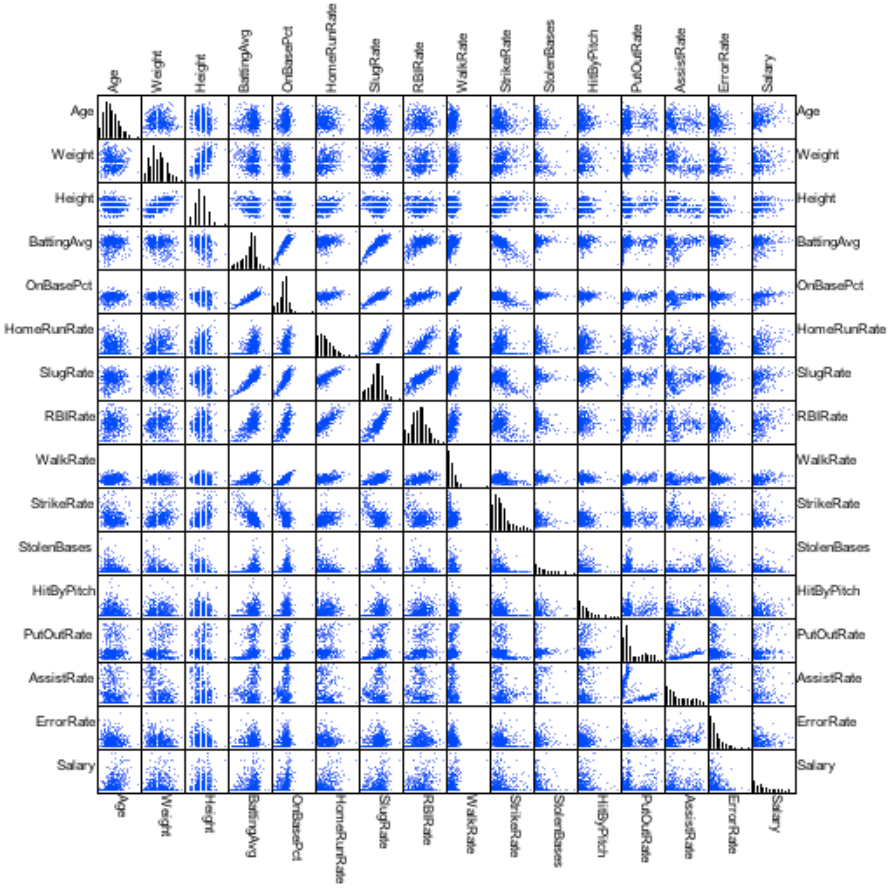
Interpreting Principal Components

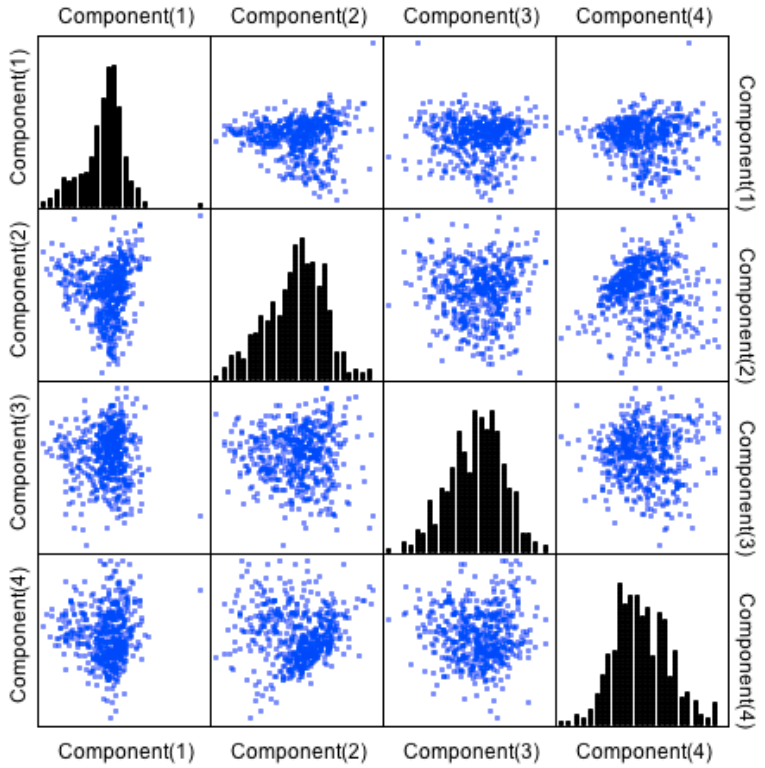
Scatterplot matrix can be used to show all pairs of components



Reducing

Interpreting Principal Components





Reducing

Interpreting Principal Components

How many components to retain?

- Kaiser's criterion
  - Assume we're factoring $\mathbf{R}$, not $\mathbf{S}$
  - Retain all components with eigenvalues > 1
  - Based on idea that no components should explain less than an original variable
- Not a good idea

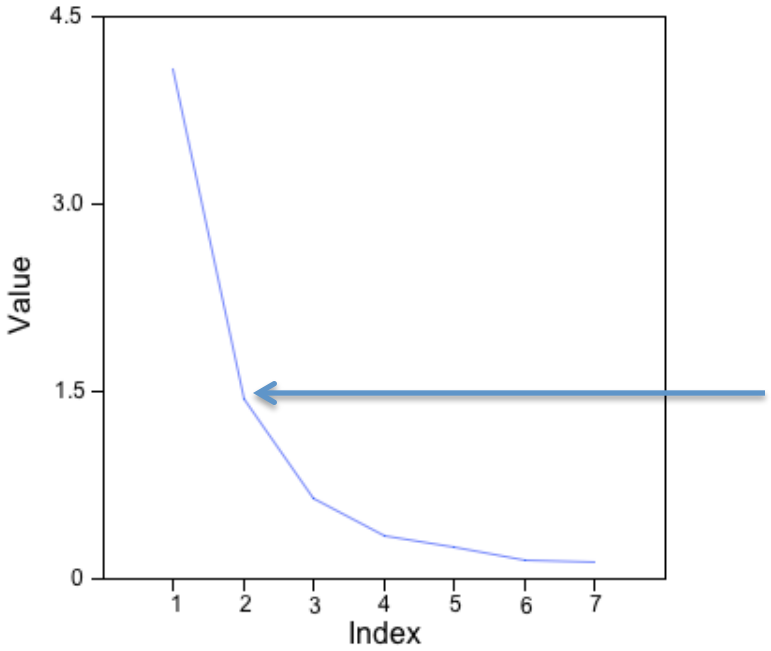
Reducing

Interpreting Principal Components

How many components to retain?

Cattell's Scree Plot

You'll never find an elbow with real data



I'm guessing it's here

Reducing

Interpreting Principal Components

How many components to retain?

- Horn's parallel analysis
  - Generate many (say, 100) spherical datasets like $\mathbf{X}$
  - Same variances on diagonal of $\mathbf{S}$, but zero off-diagonal
  - Factor each of these spherical datasets
  - Collect eigenvalues from factorings into p histograms
  - Locate, say, 95th percentile of each histogram
  - Retain up to k components, where $\lambda_k >$ percentile$_k$
- This is probably the best available method
- There are Bayesian approaches which are similar

Reducing

Factor Analysis

Reverse PC model to predict observed variables from latent factors

$\mathbf{Z} = \mathbf{FB} + \mathbf{E}$	Regression equation ($\mathbf{B}$ called loadings)
$\mathbf{F} \sim N(\mathbf{0}, \mathbf{I})$	Factors are independent normals
$\mathbf{E} \sim N(\mathbf{0}, \mathbf{\Psi})$	Errors are independent normals
$\mathbf{E} \perp\!\!\!\perp \mathbf{F}$	Factors and errors are independent
$\mathbf{\Sigma} = \mathbf{BB}' + \mathbf{\Psi}$	Fundamental equation of factor analysis
$diag(\mathbf{BB}')$	Communalities
$diag(\mathbf{\Psi})$	Specificities

Can't estimate scores $\mathbf{F}$, unlike principal components

There are several approximations, but all are controversial

The most popular involves "solving" the regression equation

$$\mathbf{F} = \mathbf{ZR}_{xx}^{-1}\mathbf{B}$$

Problem is, the factors are indeterminate because they are unobserved

Reducing

Factor Analysis vs. Principal Components

Charles Spearman

- Spearman (1904) sought a single factor for general intelligence: g
- He failed to see that a single factor is not evidence of a single underlying cause
  - There could be exogenous variables causing the association
  - The more exogenous variables we add, the more difficult it is to support the thesis

L.L. Thurstone

- Thurstone (1924) Extended Spearman's model to multiple factors

Note that FA model does the opposite of ridge regression

- **Deflates** diagonal of covariance matrix, which reduces dimensionality
- Factor analysts argue that lower dimensionality solutions support FA against PC
- But they fail to see **why** factor models "fit" the data with fewer factors
  - In most of statistics, adding parameters just to improve fit is not generally a good thing

Factor Analysis fans (and many social scientists) are Platonists

- Plato's cave is great for philosophy, not so much for data analysis

Reducing

Multidimensional Scaling

Multidimensional scaling (MDS) receives a matrix of similarities or dissimilarities among a set of objects and outputs a configuration of points in a metric space.

Assume we have a table of distances between towns taken from the pairwise distance table of a highway map.

Imagine that we toss a set of poker chips onto a table and measure the distance between every pair of chips.

Move the chips around so that the distances between the chips are as closely related as possible to the distances between the towns.

We should end up with a "map" in which each poker chip represents a town and each is in its correct position relative to the others.

Reducing

Multidimensional Scaling

Young & Householder (1938)

$$\mathbf{D} = \text{ matrix of pairwise distances}$$

$$d_{jk}^2 = d_{ij}^2 + d_{ik}^2 - 2d_{ij}d_{ik}\cos\theta_{jik}$$ Cosine rule for triangles

$$d_{ij}d_{ik}\cos\theta_{jik} = (d_{ij}^2 + d_{ik}^2 + d_{jk}^2)/2$$ Rearrange terms

$$\mathbf{x}_j'\mathbf{x}_k = w_{jk}$$ Scalar product

$$\mathbf{W} = \mathbf{X}'\mathbf{X}$$ Scalar product matrix

$\mathbf{X}$ results from eigendecomposition of $\mathbf{W}$

Point i (the one we chose to compute w_{jk}) will be at centroid of configuration

Reducing

Multidimensional Scaling

Torgerson (1951)

- Doubly center the $\mathbf{D}$ matrix instead of using an arbitrary point i
- Follow Young-Householder method after that
- Torgerson called this (classic) multidimensional scaling
  - Extension of unidimensional psychometric scaling model

Messick & Abelson (1956)

- Additive constant problem

$$d^* = c + ad + e$$

- Using these pseudo-distances allows one to work with dissimilarities

Reducing

Multidimensional Scaling

Shepard (1962)

- Begin with matrix of *dissimilarities* $\boldsymbol{\Delta}$ (no longer distances)
- Random start for points in $\mathbf{X}$
- Move points in small increments following quasi-gradient
- Shepard's model implements *nonmetric* MDS
  - Because input data could be ranks
  - The reason this works is that solution is highly constrained
  - There are $p(p\text{-}1)/2$ parameters, but each is dependent on others

Kruskal (1964)

- Recast Shepard's model as optimization problem
- Use random start (or start with Torgerson solution)
- Invert the MDS equation
  - *Distances* = *f*(*Dissimilarities*) instead of *Dissimilarities* = *f*(*Distances*)
- Kruskal's function relating distances to dissimilarities is monotonic
- Iterate using steepest descent to minimize stress (s)

$$s = \sqrt{\frac{\sum\sum (d_{ij} - \hat{d}_{ij})^2}{\sum\sum d_{ij}^2}}$$ (no deltas in here, Wikipedia and others get it wrong)

Reducing

Multidimensional Scaling

Kruskal, Young, Guttman (KYST, Alscal, SSA)

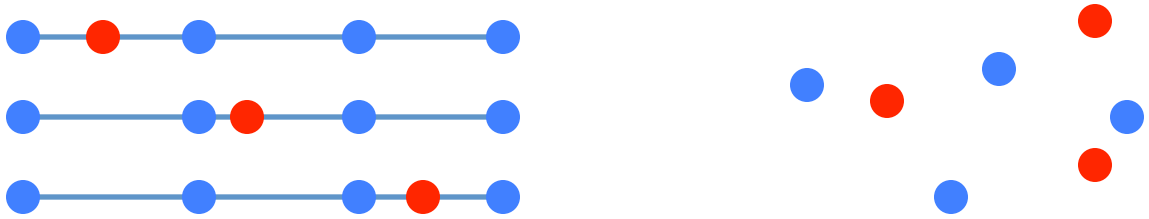
- These models allow monotonic and parametric regression
- Differ mainly in stress function and some details

Carroll

- Individual differences MDS (INDSCAL)
  - Incorporates global configuration based on k dissimilarity matrices
  - Differential weights for each subject on each dimension
  - A metric model, but very useful for formalizing individual-differences

Coombs

- Unfolding (unidimensional and multidimensional)



Reducing

Multidimensional Scaling

Sammon Mapping (1969)

- Same model as multidimensional scaling
- Loss function differs slightly from stress
- Instead of beginning with $\mathbf{D}$, Sammon maps $\mathbf{X}^p$ to $\mathbf{X}^m$ where $m < p$
- But this defeats the nonmetric capability of MDS
- Shepard-Kruskal can do the same by computing $\mathbf{D}$ from $\mathbf{X'X}$

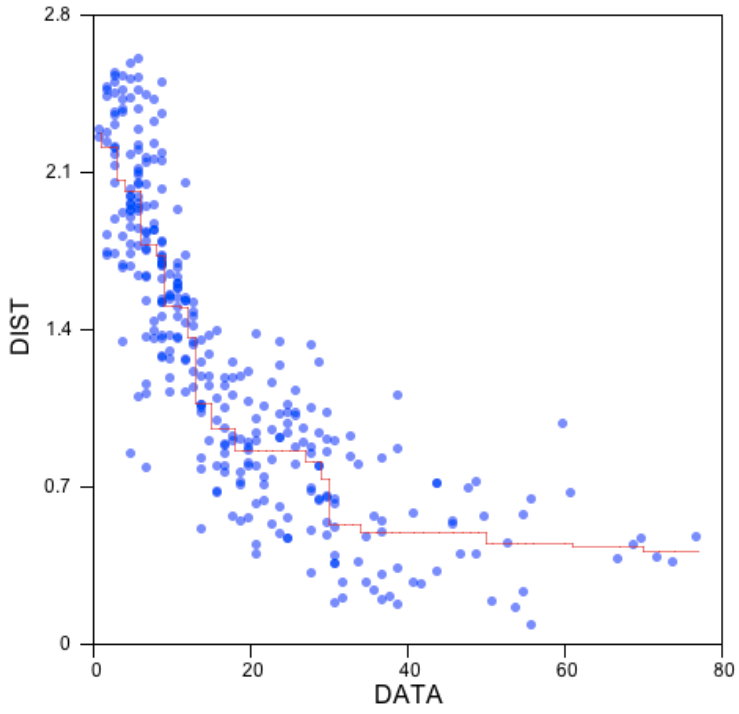
$$s = \frac{1}{\sum\sum d_{ij}^{(p)}} \frac{\sum\sum (d_{ij}^{(p)} - d_{ij}^{(m)})^2}{d_{ij}^{(p)}}$$

Reducing

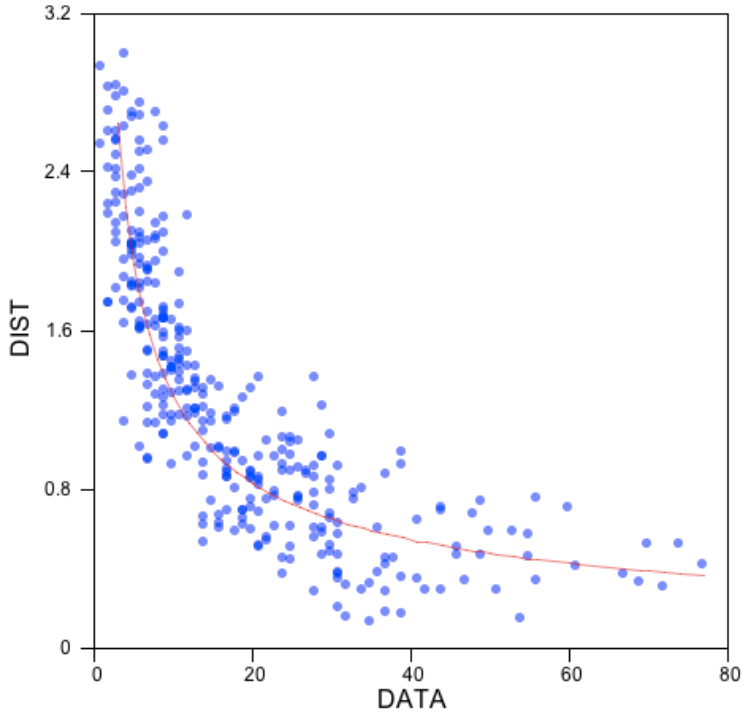
Multidimensional Scaling

Kruskal's regression model

Monotonic



Power

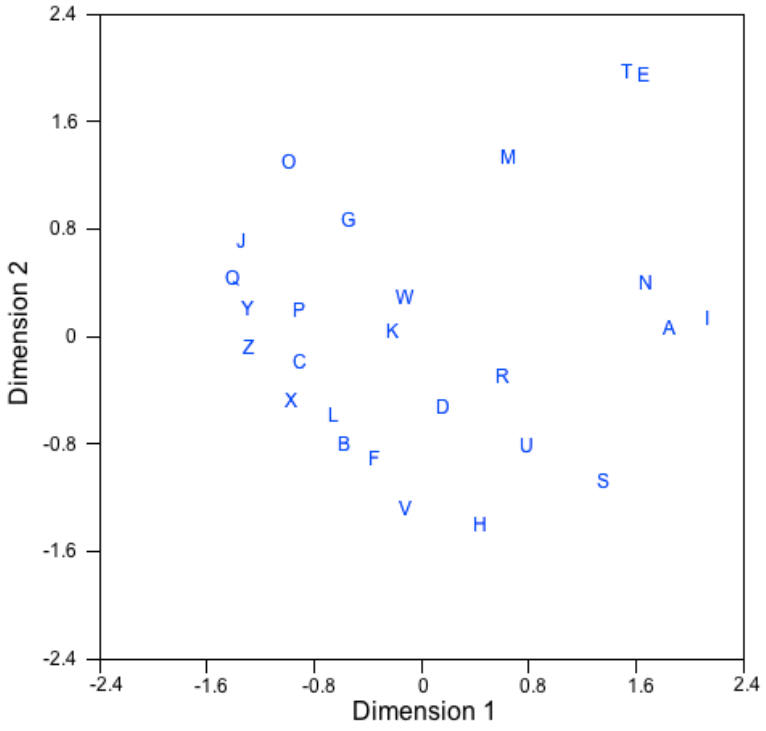


Reducing

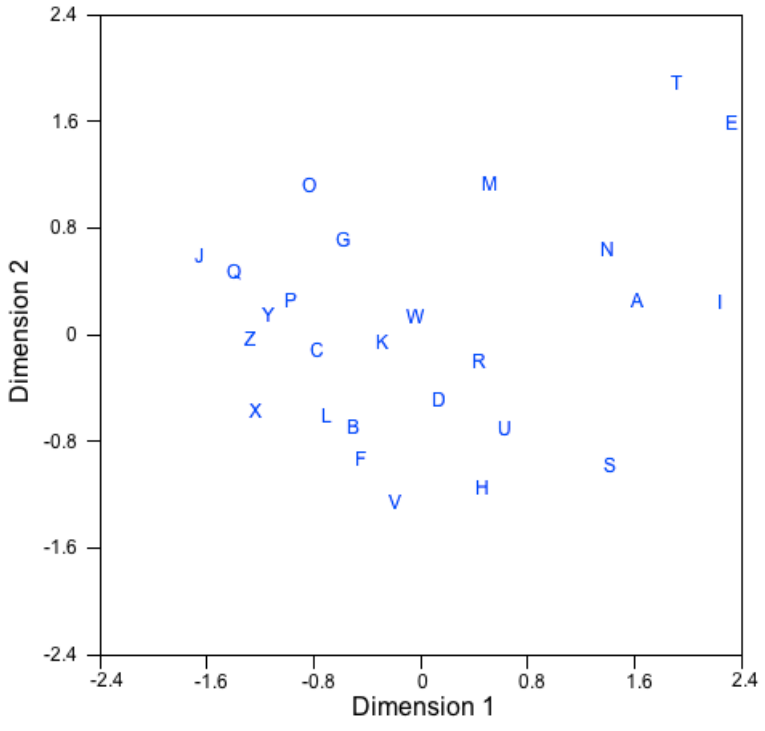
Multidimensional Scaling

MDS solution

Monotonic



Power

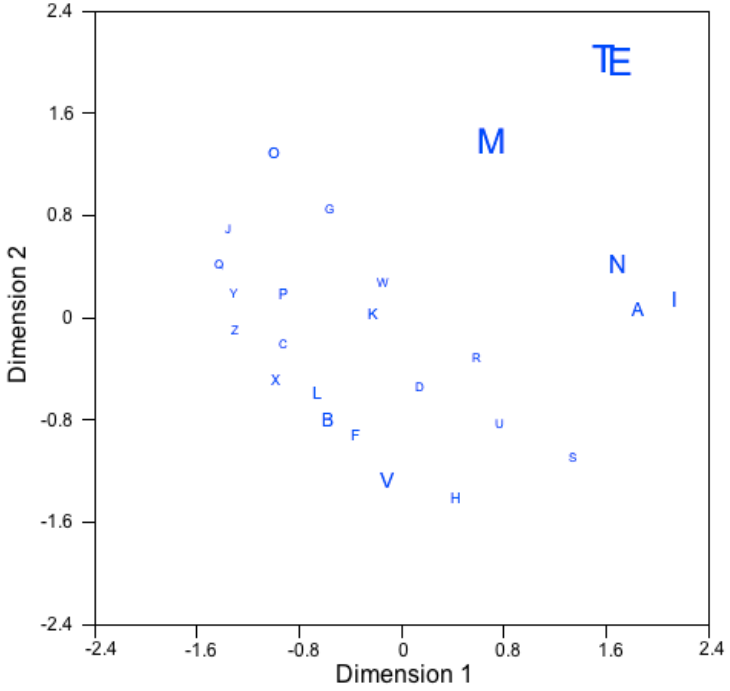


Reducing

Multidimensional Scaling

Influence statistics

Compute stress with and without point and size symbols by badness of fit



Reducing

Multidimensional Scaling

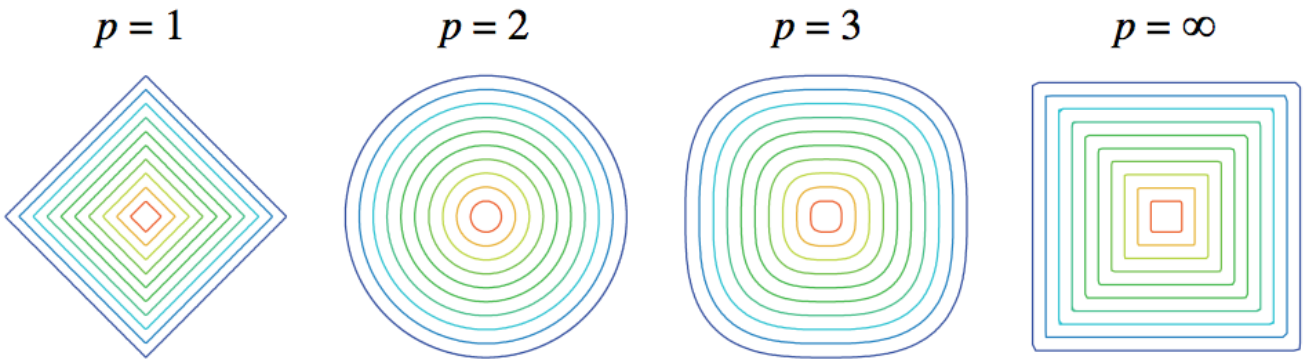
Distance metrics

Minkowski (power)

$$d_{jk} = \left[\sum_{i=1}^{n} |x_{ij} - x_{ik}|^p \right]^{1/p}$$

If p = 2, this is Euclidean

If p = 1, this is L_1 (city block or taxicab)



Reducing

Multidimensional Scaling

Difference between MDS and Principal Components

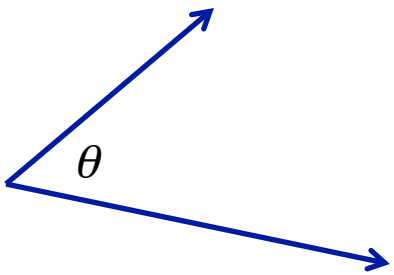
MDS

- based on distance model
- no canonical orientation
- axes not generally interpretable

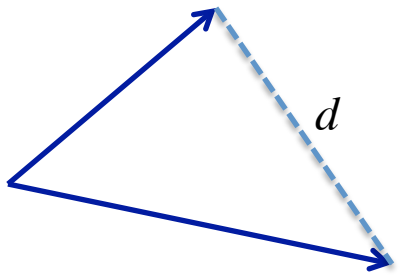
PC

- based on angular model
- proportion of variance orientation
- axes usually interpretable

PC



MDS



Reducing

Nonlinear Dimension Reduction

Laplacian Eigenmaps (Belkin & Niyogi, 2002)

Given $\mathbf{X}_{np}$, we construct a weighted graph with n nodes, one for each point , and the set of edges connecting neighboring points to each other

We put an edge between nodes *i* and *j* if $||\mathbf{x}_i - \mathbf{x}_j|| < \epsilon$

Construct the *heat kernel* (a weighted adjacency matrix)

$$\mathbf{W}_{ij} = e^{-\frac{||\mathbf{x}_i - \mathbf{x}_j||^2}{t}}$$

And build and factor the Laplacian matrix

$$\mathbf{D}_{ii} = \sum_j \mathbf{W}_{ji} \quad \text{(diagonal weight matrix)}$$

$$\mathbf{L} = \mathbf{D} - \mathbf{W} \quad \text{(Laplacian matrix)}$$

$$\mathbf{L}\mathbf{y} = \lambda\mathbf{D}\mathbf{y} \quad \text{(eigensystem)}$$

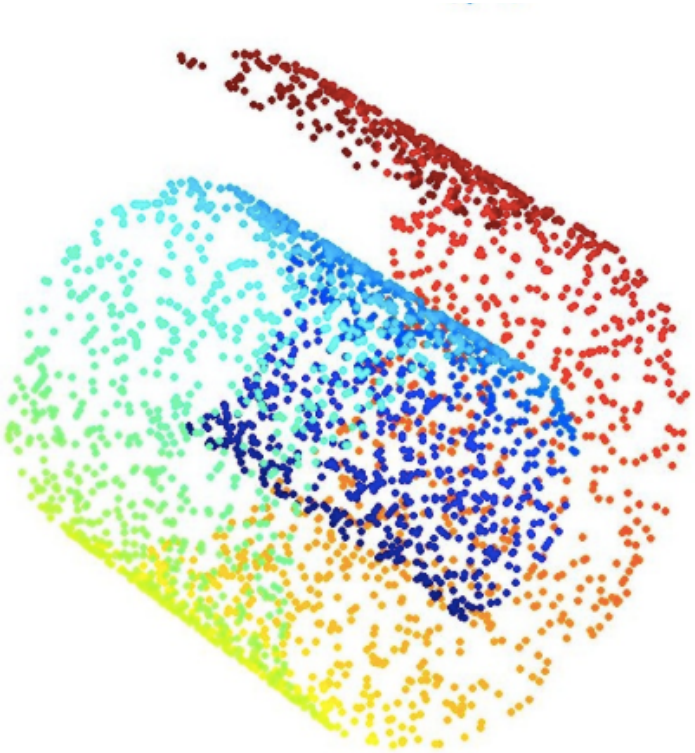
Take eigenvectors corresponding to *smallest* roots

Because we are working with negative of adjacency matrix

Reducing

Nonlinear Dimension Reduction

Swiss Roll Dataset

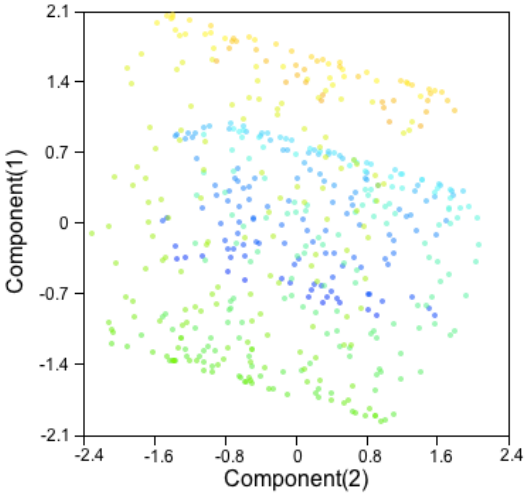


Reducing

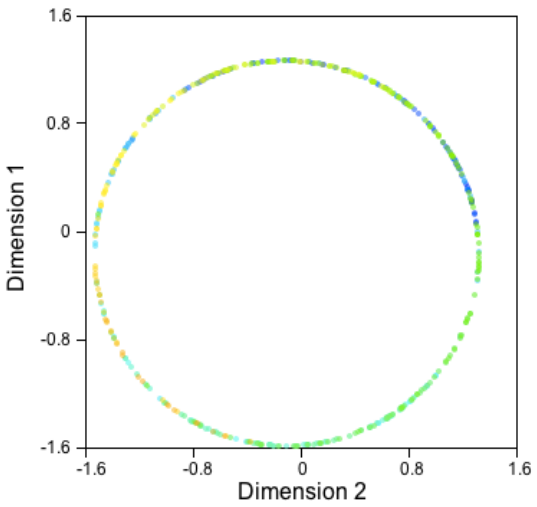
Nonlinear Dimension Reduction

Laplacian Eigenmaps (Belkin & Niyogi, 2002)

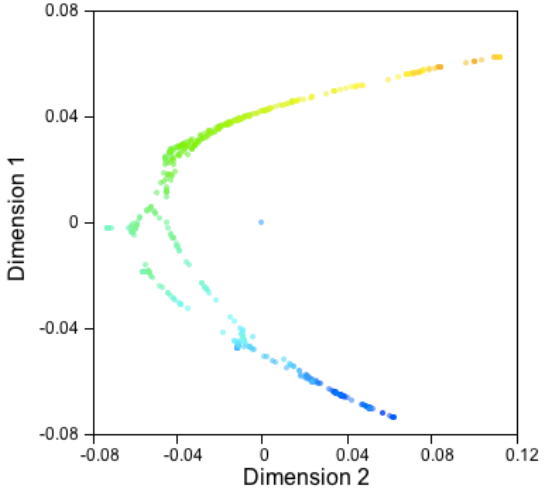
Principal Components



Multidimensional Scaling



Laplacian Eigenmap



Reducing

Nonlinear Dimension Reduction

Other approaches

- PARAMAP (Shepard & Carroll, 1966)
  - MDS variant
- ISOMAP (Tennenbaum, de Silva, Langford, 2000)
  - MDS on a geodesic graph created from near neighbors
- Locally Linear Embedding (Roweis & Saul, 2000)
  - Similar to Laplacian Eigenmaps, using decomposition of Laplacian
- Diffusion Maps (Lafon & Coifman, 2004)
  - Uses heat kernel based on Markov Chain
- Topological embedding (Carlsson, 2005)
  - Heat kernels for shape recognition

Many others

- None of these methods does well with error
- The manifold needs to have only a small error component
- They do best on toy datasets