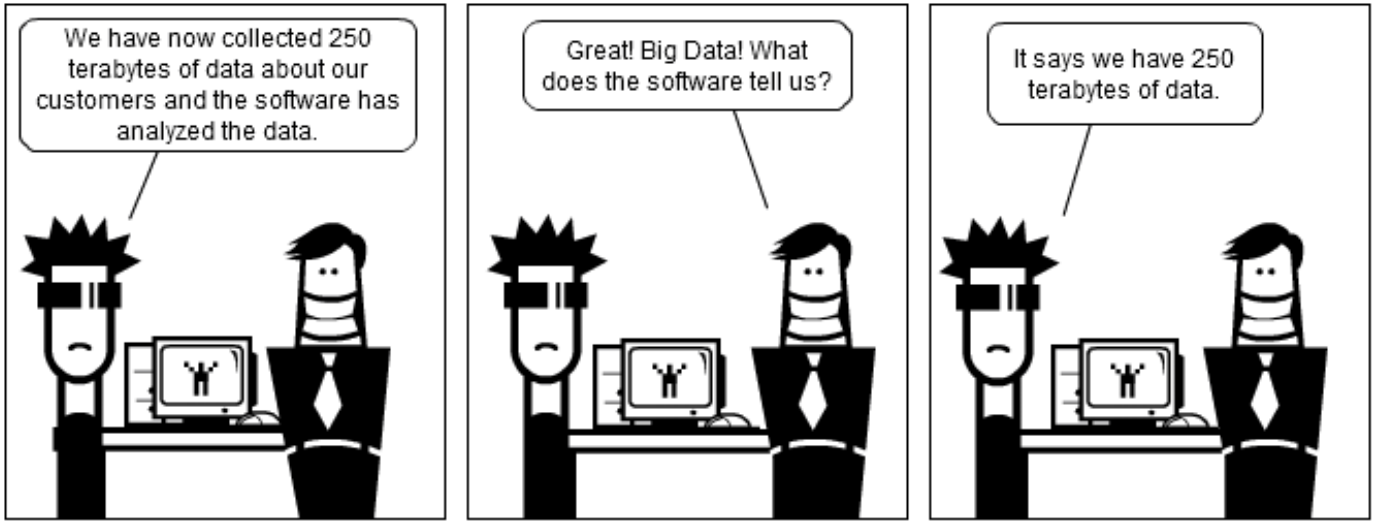
Multivariate Analysis Part I

Nancy Flournoy
University of Missouri

flournoyn@missouri.edu September 11, 2019
4th International Summer School on Data Science
Split, Croatia

September 10, 2019

Big Data Challenge



- A common challenge from Big Data is how to extract useful information with available computational facilities.
- Data reduction is crucial.
- *"Briefly, and in its most concrete form the object of statistical methods is the reduction of data."* *Fisher (1922)*

Outline

Data reduction techniques operate on the variables (data features, factors) and/or on the number of observations. In this part of the lecture, we focus on reduction the dimensionality of the the data features.

LECTURE PART I

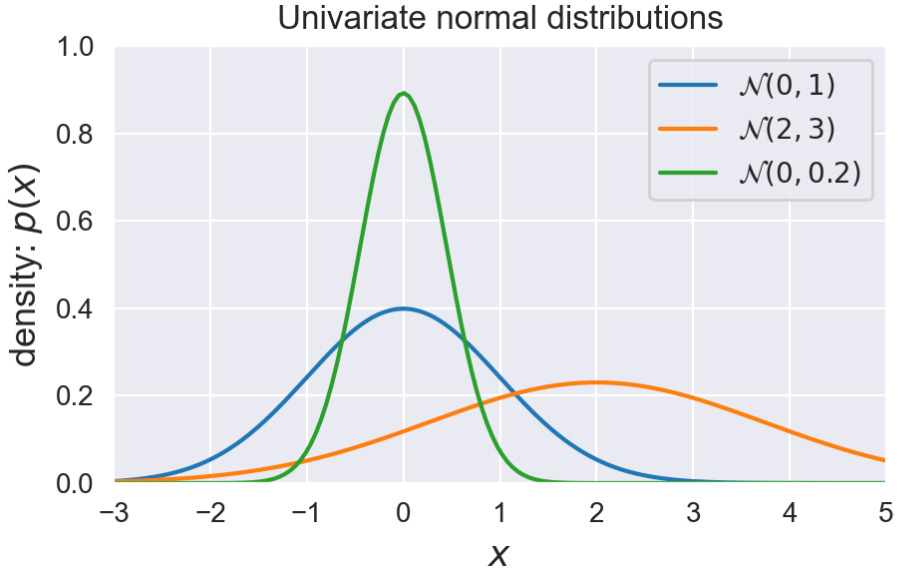
1. Multivariate Normal Distribution
2. Principal Component Analysis
3. Factor Analysis
4. Beyond Linear Models and Normality

Univariate Normal Distribution

The normal probability density function (pdf) is

$$p(x|\mu,\sigma) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right), (x,\mu) \in \Re,\ \sigma > 0.$$

A random variable X with this pdf is said to be *normally distributed*, which is denoted as $X \sim \mathcal{N}(\mu.\sigma^2)$.



mean:
$\mathrm{E}(X) = \int x p(x,\mu,\sigma)dx$
$= \mu$
variance: $\mathrm{Var}(X)$
$= \int (x-\mu)^2 p(x|\mu,\sigma)dx$
$= \sigma^2$
standard deviation: $= \sigma$

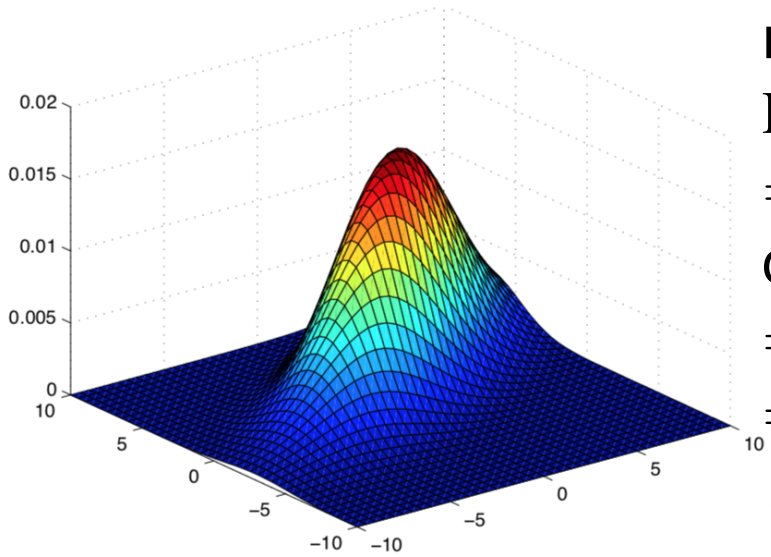
Multivariate Normal Distribution

The multivariate normal probability density function (pdf) is

$$p(x|\mu,\mathbf{\Sigma}) = \frac{1}{\sqrt{2(\pi)^n|\mathbf{\Sigma}|}} \exp\left(-\frac{1}{2}(x-\mu)^T\mathbf{\Sigma}^{-1}(x-\mu)\right)$$

$$(x,\mu) \in \Re^m,\ \Sigma pos.def.$$

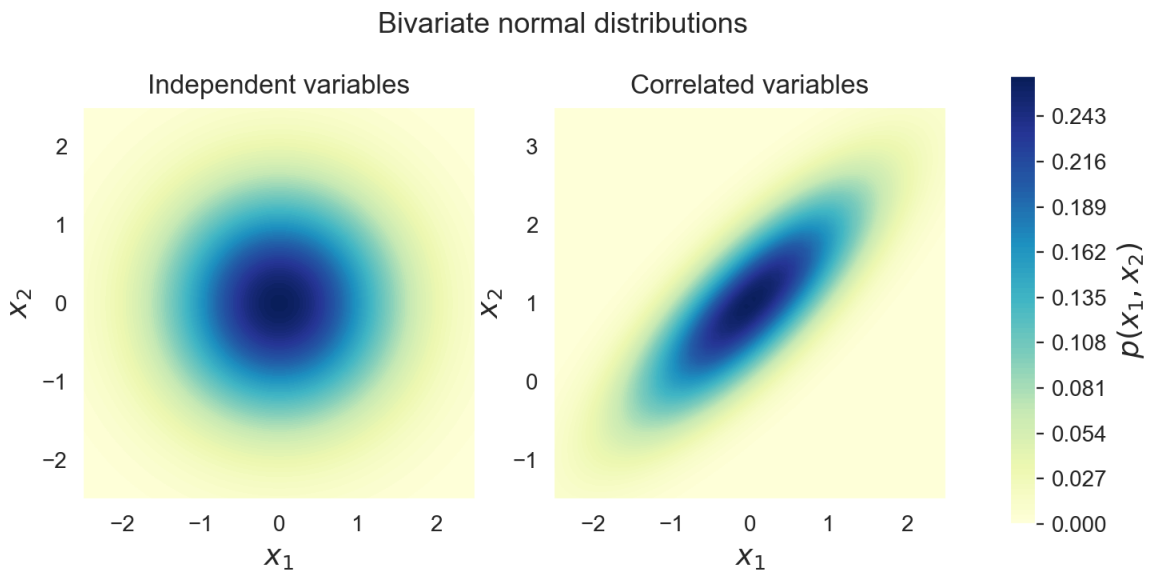
A vector X of n observations with this pdf is said to be *multivariately normally distributed*, denoted $X \sim \mathcal{N}(\mu,\mathbf{\Sigma})$.



mean:
$\mathrm{E}(X) = \int xp(x,\mu,\mathbf{\Sigma})dx$
$= \mu$
covariance: $\mathrm{Cov}(X)$
$= E(X-\mu)^2$
$= \int(x-\mu)(x-\mu)^T \times p(x|\mu,\Sigma)dx$
$= \mathbf{\Sigma}$

Good summary of properties is at
www.stat.ubc.ca/∼ will/cx/private/normal_properties.pdf

Independent and Correlated Variables



Suppose independent $X_1, \ldots, X_n \sim \mathcal{N}(\mu_X, \Sigma_X)$.
Let $\bar{X} = \frac{1}{n}(X_1, \ldots, X_n)$,
then

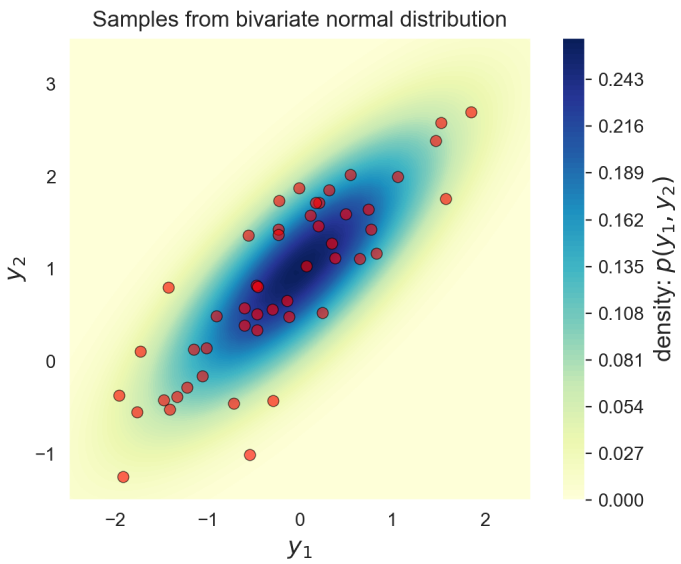
$$\bar{X} \sim \mathcal{N}(\mu_X, \Sigma_X/n)$$

Transformations When X is $n \times m$ matrix, $\mathbf{X} \sim \mathcal{MN}(\mu, \mathbf{\Sigma})$.

Let $\mathbf{L}$ be the matrix for which $\mathbf{L}\mathbf{L}^T = \mathbf{\Sigma}$.
Let $\mathbf{B}$ be a $l \times n$ matrix of constants.
$\mathbf{a}$ and μ are $m \times 1$.
$\mathbf{b}$ is $l \times 1$.

- Linear combinations of normal random variables are normal.
  $\mathbf{a}^T\mathbf{X} \sim \mathcal{N}(\mathbf{a}^T\mu, \mathbf{a}^T\mathbf{\Sigma}\mathbf{a})$.
- Affine transformations of normal random variables are normal.
  - The l random variables $\mathbf{b} + \mathbf{B}\mathbf{X} \sim \mathcal{MN}(\mathbf{b} + \mathbf{B}\mu, \mathbf{B}\mathbf{\Sigma}\mathbf{B}^T)$.
  - Let $Z \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_m)$ be n independent standard normal random variables.
    Then $\mathbf{X} = \mu + \mathbf{L}\mathbf{Z} \sim \mathcal{MN}(\mu, \mathbf{L}\mathbf{L}^T)$.

Sampling from a Multivariate Normal Variable



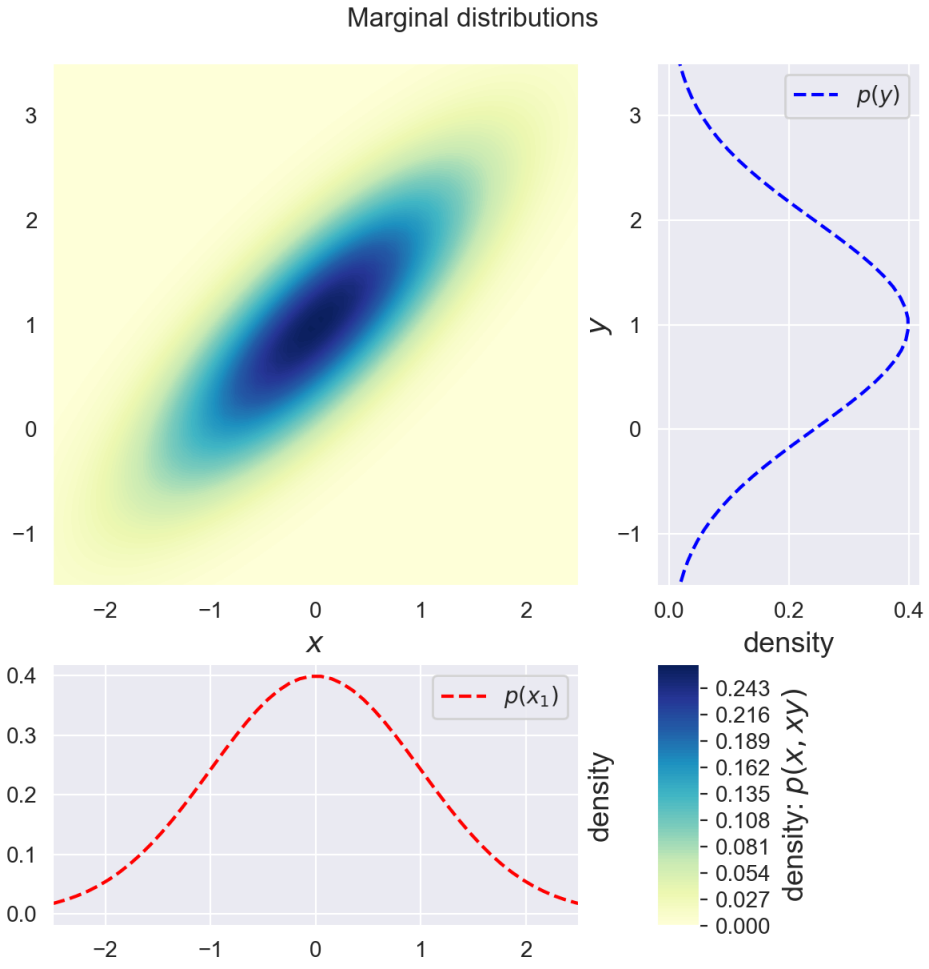
To generate $\mathcal{N}(\boldsymbol{\mu}_Y, \boldsymbol{\Sigma}_Y)$ sample,
first generate $\mathcal{N}(0, \mathbf{I}_n)$ rvs, where $\mathbf{I}_n$ is the $n \times n$ dimensional identify matrix, that is,
generate n independent normal rvs. Then transform to $\mathcal{N}(\boldsymbol{\mu}_Y, \boldsymbol{\Sigma}_Y)$.

$\boldsymbol{\mu}_y = \boldsymbol{\mu}$.

$\boldsymbol{\Sigma}_Y = \mathbf{L} I_n \mathbf{L}^T = \mathbf{L}\mathbf{L}^T$.

Marginal Normal Variables:

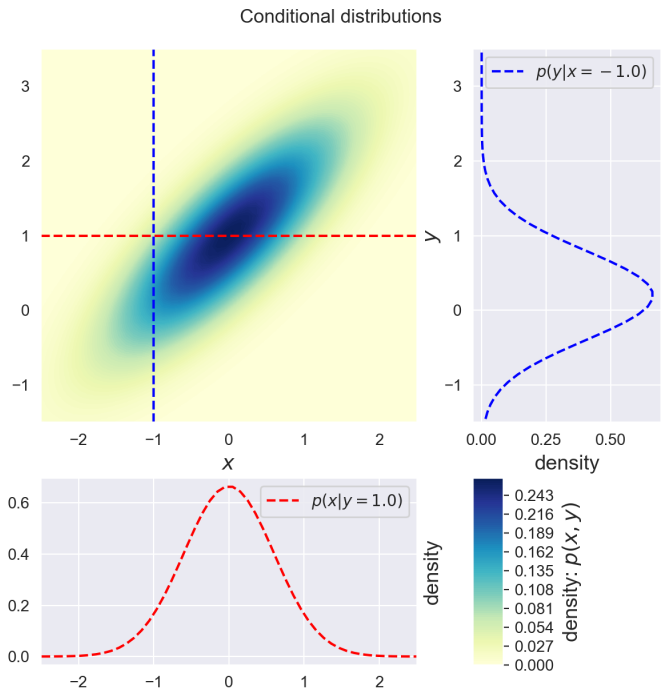
$$\begin{pmatrix} X \\ Y \end{pmatrix} \sim \mathcal{N}\left(\begin{pmatrix} \mu_X \\ \mu_Y \end{pmatrix}, \begin{pmatrix} \Sigma_X & \Sigma_{XY} \\ \Sigma_{YX} & \Sigma_Y \end{pmatrix}\right)$$



$$X \sim \mathcal{N}(\mu_X, \Sigma_X)$$

$$Y \sim \mathcal{N}(\mu_Y, \Sigma_Y)$$

Conditional Normal Variables



If $\begin{pmatrix} X \\ Y \end{pmatrix} \sim \mathcal{N}\left(\begin{pmatrix} \mu_X \\ \mu_Y \end{pmatrix}, \begin{pmatrix} \Sigma_X & \Sigma_{XY} \\ \Sigma_{YX} & \Sigma_Y \end{pmatrix}\right)$

then

$$Y|X \sim \mathcal{N}(\mu_{Y|X}, \Sigma_{Y|X})$$

$$X|Y \sim \mathcal{N}(\mu_{X|Y}, \Sigma_{X|Y})$$

Idea Behind Principal Component Analysis

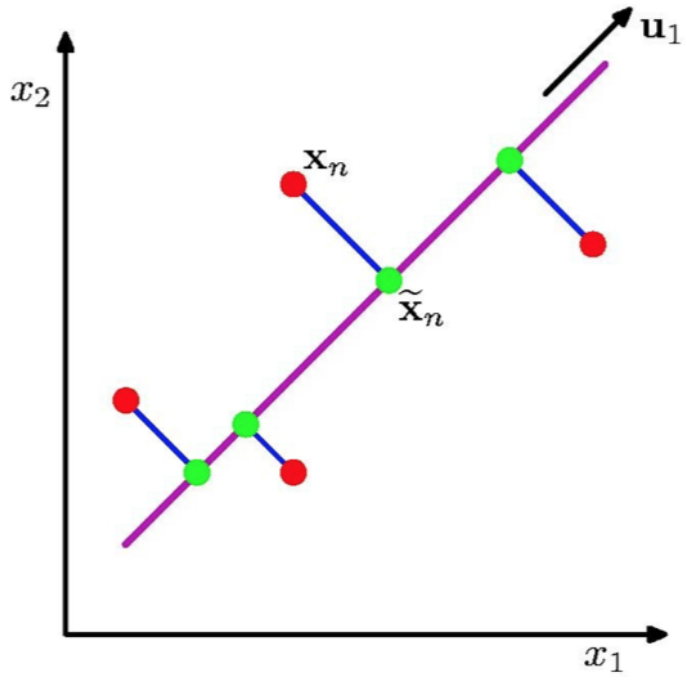
Suppose m variables (factors, features) are observed on n experimental units (subjects).

Project variables from a m-dimensional space to a lower dimensional space while preserving as much information as possible.

Examples:

- find best planar approximation to 3-dimensional data
- find best 12-dimensional approximation to 10^4-dimensional data

First Principal Component



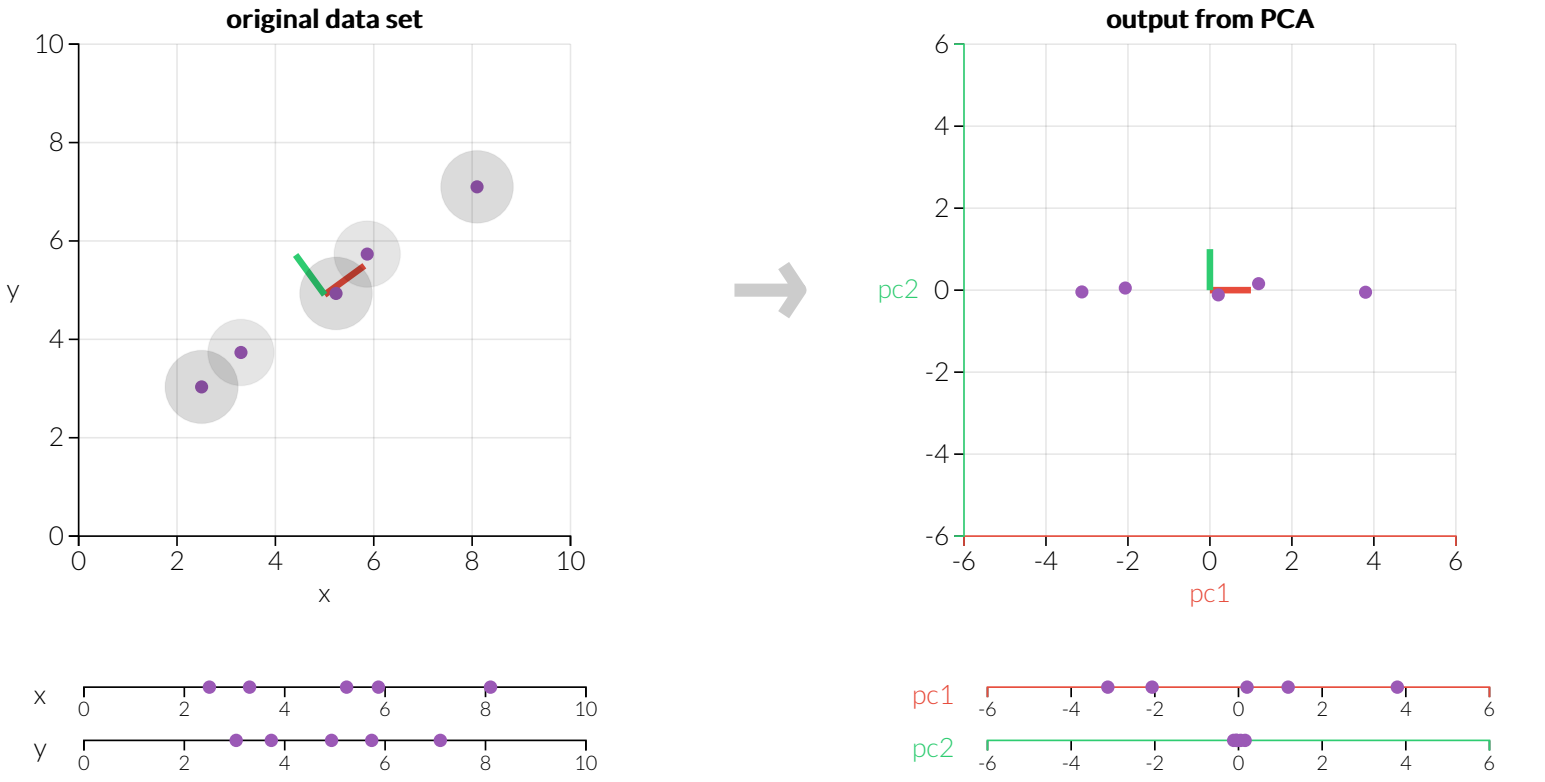
How does this compare to regression projections?

An othogonal projection of data onto a lower-dimensional linear space that

1. maximizes variance of projected data (purple line)
2. minimizes mean squared distance between
   - data points and
   - projections

   (blue lines)

2D Example of PCA

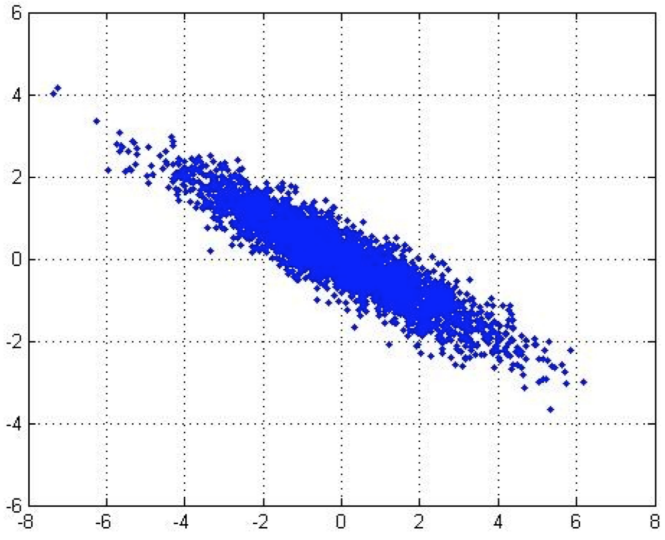


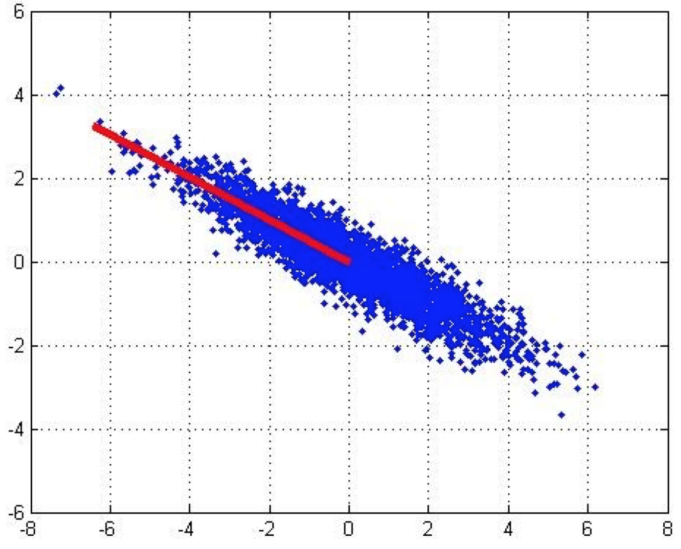
The Principal Components

The principal components are **vectors** originating from the center of mass.

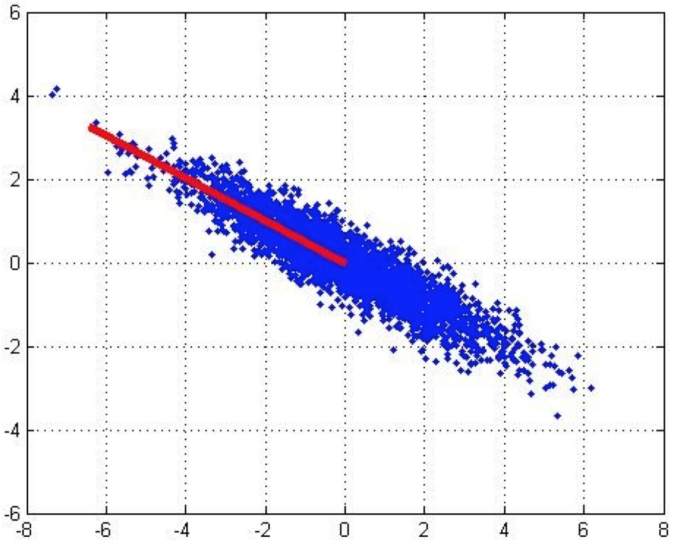
- Principal component #1
  points in the direction of the **largest variance**.
- Each subsequent principal component is
  1. is **orthogonal** to the previous ones, and
  2. points in the direction of the **largest variance in the residual subspace**

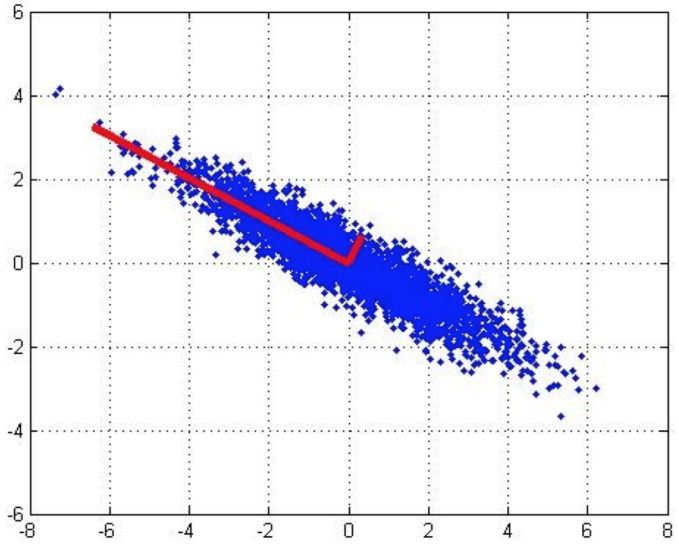
PC#1 from Bivariate Normal Data





PC#2 from Bivariate Normal Data





Set Up for the PC Transformation

- n is # of experiments performed on distinct experimental units (e.g., subjects)
- m is # of features observed on each unit.
- Centered data is represented by a $\mathbf{X}$ is a $n \times m$ matrix. That is,
  the mean of each observed feature is subtracted from each feature value
  so the mean of each column is zero.

You may also want to divide each data point by the column standard deviation to remove the effect of the units of measurement.

Terminology of the PC transformation

- ***loadings:*** A set of m dimensional weights $\mathbf{w}_k = (w_1, \ldots, w_m)_k$ that map each row vector $\mathbf{x}_i = (x_{i1}, \ldots, x_{im})$ of $\mathbf{X}$ to a new vector
- ***factor scores:*** $\mathbf{t}_{ki} = (t_1, \ldots, t_l)_i = \mathbf{w}_k \cdot \mathbf{x}_i, l \leq m.$

The results of a PCA are usually discussed in terms of component or *scores* and *loadings*.

Some Rules for Means

X and Y are random variables; a and b are constants.

$$E(X) = \mu_x = \begin{cases} \sum_{i=1}^{n} p_i x_i, X \text{ discrete;} \\ E(X) = \int xp(x), X \text{ continuous} \end{cases}.$$

1. $E(aX + c) = aE(X) + c$
2. $E(X + Y) = E(X) + E(Y)$

Sample estimate of the mean μ:

$$\widehat{\mu}_x = \bar{x} = \sum_{i=1}^{n} \frac{1}{n} x_i.$$

Some Rules for Variances and Covariances

$$Var(X) = E(X-\mu_X)^2 = \begin{cases} \sum_{i=1}^{n} p_i(x_i-\mu)^2, X \text{ discrete;} \\ \int (x-\mu_X)^2 p(x), X \text{ cont.} \end{cases}$$

$$Cov(X,Y) = E(X-\mu_x)E(Y-\mu_Y)$$

1. $Var(aX+c) = a^2 Var(X)$
2. $Var(X+Y) = Var(X) + Var(Y) + 2Cov(X,Y)$.
3. $Var(aX+bY) = a^2 Var(X) + b^2 Var(Y) + 2abCov(X,Y)$.

Sample Estimates of Variances and Covariances

.

1. $\widehat{Var}(X) = \sum_{i=1}^{n} \frac{1}{n-1}(x_i - \bar{x})^2$
2. $\widehat{Cov}(X, Y) = \sum_{i=1}^{n} \frac{1}{n-1}(x_i - \bar{x})(y_i - \bar{y})$

When x and y have been centered in preprocessing,

1. $\widehat{Var}(X) = \sum_{i=1}^{n} \frac{1}{n-1}(x_i)^2 \quad \widehat{Var}(Y) = \sum_{i=1}^{n} \frac{1}{n-1}(y_i)^2$
2. $\widehat{Cov}(X, Y) = \frac{1}{n-1} \sum_{i=1}^{n} x_i y_i$

$$\max_{a+c=1} \widehat{Var}(aX + cY) =$$
$$\max_{a+c=1} \{a^2 \widehat{Var}(X) + c^2 \widehat{Var}(Y) + ac\widehat{Cov}(X, Y)\}$$

What happens to n?

PCA Algorithms

Three algorithms

1. sequential
   - may be terminated when user deems enough components have been calculated;
   - can handle missing data.
2. sample covariance matrix
   - computational requirements are greater than SVD
3. singular value decomposition (SVD)

Principal Components via the Sequential Algorithm

$m =$ # of column vectors of length n,
$n =$ # of experimental units,

1. **Center the data:** subtract the mean of each feature to obtain a set of centered column vectors $\{\mathbf{x}_1, \ldots, \mathbf{x}_m\}$.
2. **Compute weight vector for the first PC vector:**

$$\mathbf{w}_1 = \arg \max_{\|\mathbf{w}\|=1} \frac{1}{m} \sum_{i=1}^{m} (\mathbf{w} \cdot \mathbf{x}_i)^2$$

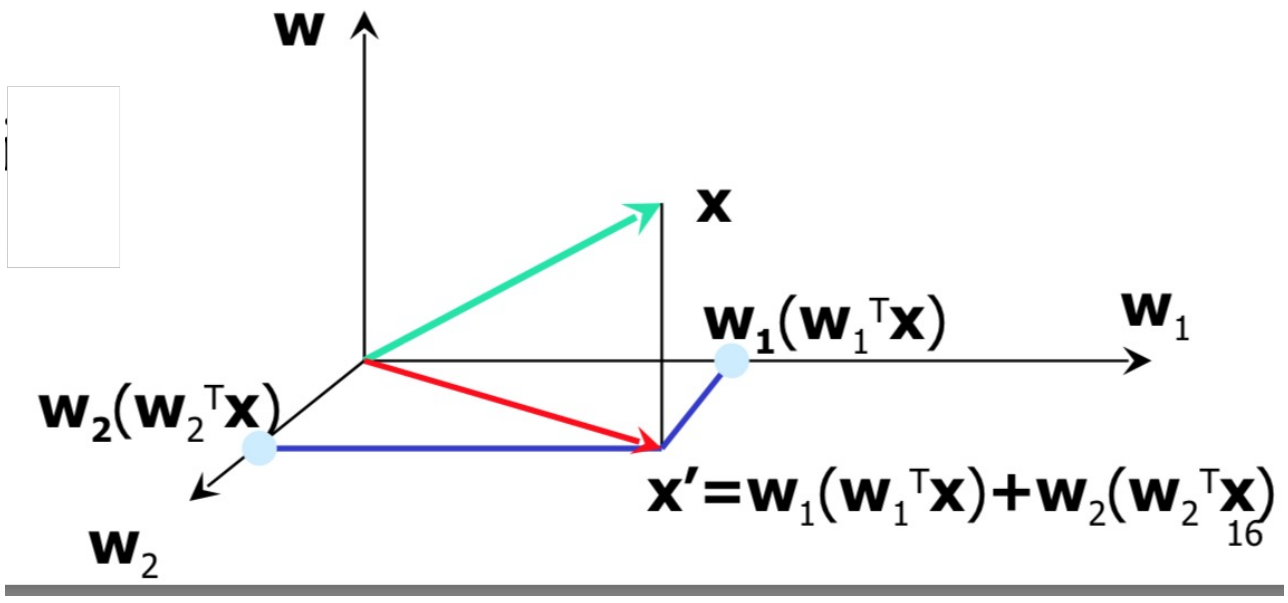
3. **Find the weight vector for the kth PC vector** $(k = 2, \ldots, m)$:

$$\mathbf{w}_k = \arg \max_{\|\mathbf{w}\|=1} \frac{1}{m} \sum_{i=1}^{m} \left[\mathbf{w}^T \left(\mathbf{x}_i - \mathbf{x}' \right) \right]^2$$

PCA reconstruction: $\mathbf{x}'_i := \sum_{j=1}^{k-1} \mathbf{w}_j \mathbf{w}_j^T \mathbf{x}_i$.

x_i in the Projected Subspace

$$\mathbf{x}'_i := \sum_{j=1}^{k-1} \mathbf{w}_j \mathbf{w}_j^T \mathbf{x}_i.$$



Full PC Decomposition of X: $T = \mathbf{WX}$

Transformation to all PCs can be done at once.

1. **W** is $m \times m$ matrix of weights whose columns are the eigenvectors of the covariance matrix $\mathbf{XX}^T$.
2. Rows of **W** multiplied by the square root of the corresponding eigenvectors
(i.e., eigenvectors scaled up by their variances)
are the loadings.

PCA via the Sample Covariance Matrix

Given m variable column vectors $\{\mathbf{x}_1, \ldots, \mathbf{x}_m\}$ of length n,

1. Compute the $mx1$ sample mean vector $\bar{\mathbf{x}} = \frac{1}{n}\sum_{i=1}^{n} \mathbf{x}_i$.
2. Compute the sample covariance matrix $\widehat{\mathbf{\Sigma}}$:

$$\widehat{\mathbf{\Sigma}} = \frac{1}{n-1}\sum_{i=1}^{n} (\mathbf{x}_i - \bar{\mathbf{x}})(\mathbf{x}_i - \bar{\mathbf{x}})^T$$

3. Use a matrix algebra system to compute the matrix $\mathbf{V}$ of eigenvectors that diagonalizes $\widehat{\mathbf{\Sigma}}$.

$$\mathbf{V}^{-1}\widehat{\mathbf{\Sigma}}\mathbf{V} = \mathbf{\Lambda},$$

where $\mathbf{\Lambda} = diag(\lambda_{(1)}, \ldots, \lambda_{(m)})$ and the elements $\{\lambda_{(k)}\}$ are the solutions to $\widehat{\mathbf{\Sigma}}\mathbf{v} - \lambda\mathbf{v} = \mathbf{0}$.
$\lambda_{(k)}$ is the kth largest eigenvalue of $\widehat{\mathbf{\Sigma}}$.

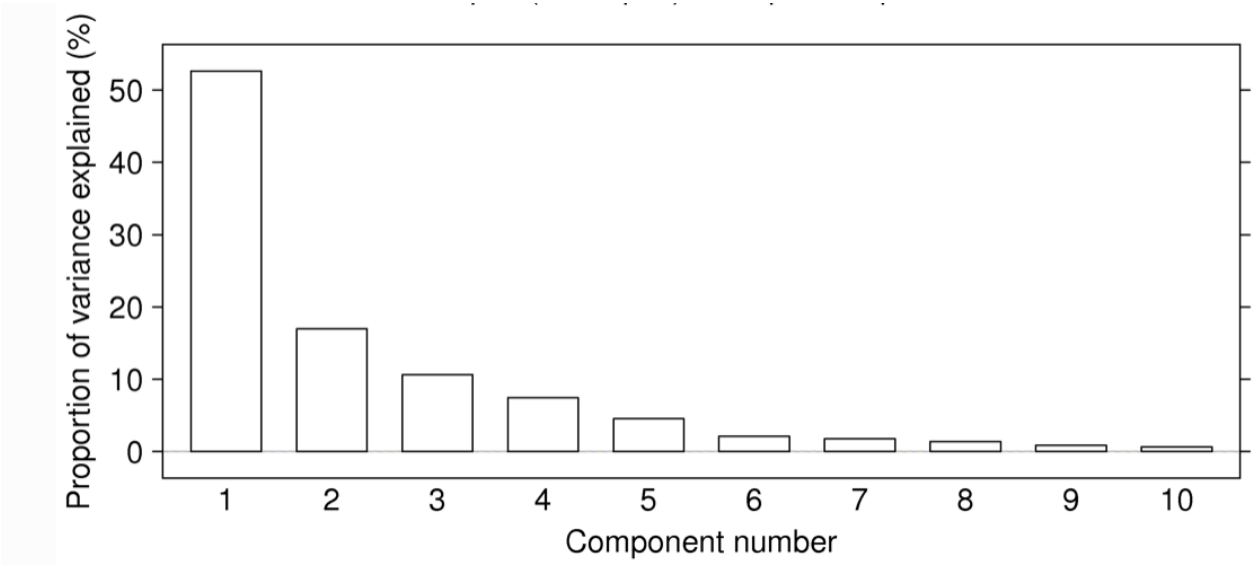
- The PC basis vectors are the eigenvector columns of $\mathbf{V}$.

Ordered Eigenvalues $\lambda_{(1)} \geq \cdots \geq \lambda_{(m)}$ are Useful for Data Reduction

- $\lambda_{(k)} \propto$ the portion of the "variance" (actually the sum of the squared distances of the points from their multidimensional mean) that is associated with each eigenvector.
- The amount of residual variance explained by the PC vectors decreases with decreasing eigenvalues.
- PCA rotates points around their mean in order to align with the principal components. This moves as much of the variance as possible (using an orthogonal transformation) into the first few dimensions. Small values in the remaining dimensions indicate factors that may be dropped with minimal loss of information.
- PCA is the optimal orthogonal transformation for producing the subspace that has largest "variance"

Use Eigenvalues to Select Factors

SCREE Plot:



"Percent variance explained" for the kth component is

$$\frac{\lambda_{(k)}}{\sum_{j=1}^{m} \lambda_{(j)}}.$$

PCs via Singular Value Decomposition

SVD decomposes a $n \times m$ matrix X into three matrices:

$$\mathbf{X} = \mathbf{UDW}^{\mathbf{T}}$$

- $\mathbf{U}(n \times n)$ and $\mathbf{W}(p \times p)$ are orthonormal, that is, each column has unit length and is orthogonal to the others.
- Columns of **U** are called the *left singular vectors* of **X**.
- Columns of **W** are called the *right singular vectors* of **X**.
- $\mathbf{D}(n \times p)$ is a rectangular diagonal matrix of positive numbers equal to $\sqrt{\lambda_{(k)}}$, the square root of the ordered eigenvalues of $\mathbf{X}^T\mathbf{X}$.
  Factors in the upper left represent signal.
  Factors in the lower right represent noise.
  $\{\sqrt{\lambda_{(k)}}\}$ are called singular values.

PCs via the SVD $\mathbf{X} = \mathbf{UDW}^T$

$$\mathbf{X}^T\mathbf{X} = \mathbf{WD}^T\mathbf{U}^T\mathbf{UDW}^T = \mathbf{WD}^T\mathbf{DW}^T = \mathbf{W}\widetilde{\mathbf{D}^2}\mathbf{W}^T$$

where $\widetilde{\mathbf{D}}$ is a square diagonal matrix with the singular values of $\mathbf{X}$ and the excess zeros chopped off that satisfies $\widetilde{\mathbf{D}^2} = \mathbf{D}^T\mathbf{D}$.
The diagonal elements of $\widetilde{\mathbf{D}^2}$ are the eigenvalues.

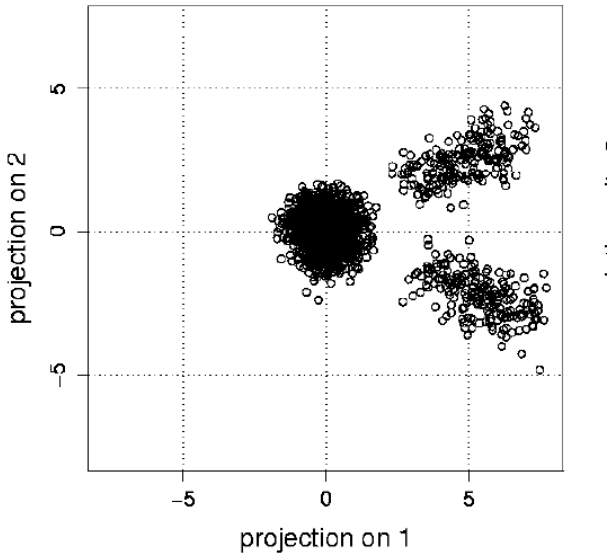
Using the SVD, the score matrix $\mathbf{T}$ can be written

$$\mathbf{T} = \mathbf{XW} = (\mathbf{UDW}^T)\mathbf{W} = \mathbf{UD}$$

Each column of $\mathbf{T}$ is given by one of the left singular vectors of $\mathbf{X}$ multiplied by the corresponding singular value. This form is also the polar decomposition of $\mathbf{T}$.

Algorithms can calculate SVD without having to form $\mathbf{X}^T\mathbf{X}$.

The PC transformation can help as a pre-processing step before clustering.



SVD scatter plot of simulated times series of gene transcriptional responses
(Wall, ME, Rechtsteiner, Rocha, 2003)

- Bottom right cluster corresponds to sine wave genes.
- Top right cluster corresponds to exponential decay genes.
- Genes clustered around the origin correspond to noise-only genes.

Factor Analysis and PCA terms have evolved over time.

PC is a type of data rotation around center of gravity as described.

- In the social science community,
  - Factor Analysis refers to selecting a subset of transformed variables (factors) and describing patterns of loadings in them.
  - Various kinds transformations on either the correlation or covariance matrix (see SPSS) can be made.
  - Factor loadings are analyzed to detect data structure (i.e., latent constructs or factors), variable groupings and causal modeling.
- In the data mining community,
  - PCA refers to using PC transformation for data reduction where PC is applied to the covariance matrix (through perhaps using the sequential or SVD algorithms).

Factor Analysis: Perspective of UCLA Institute for Digital Research & Education

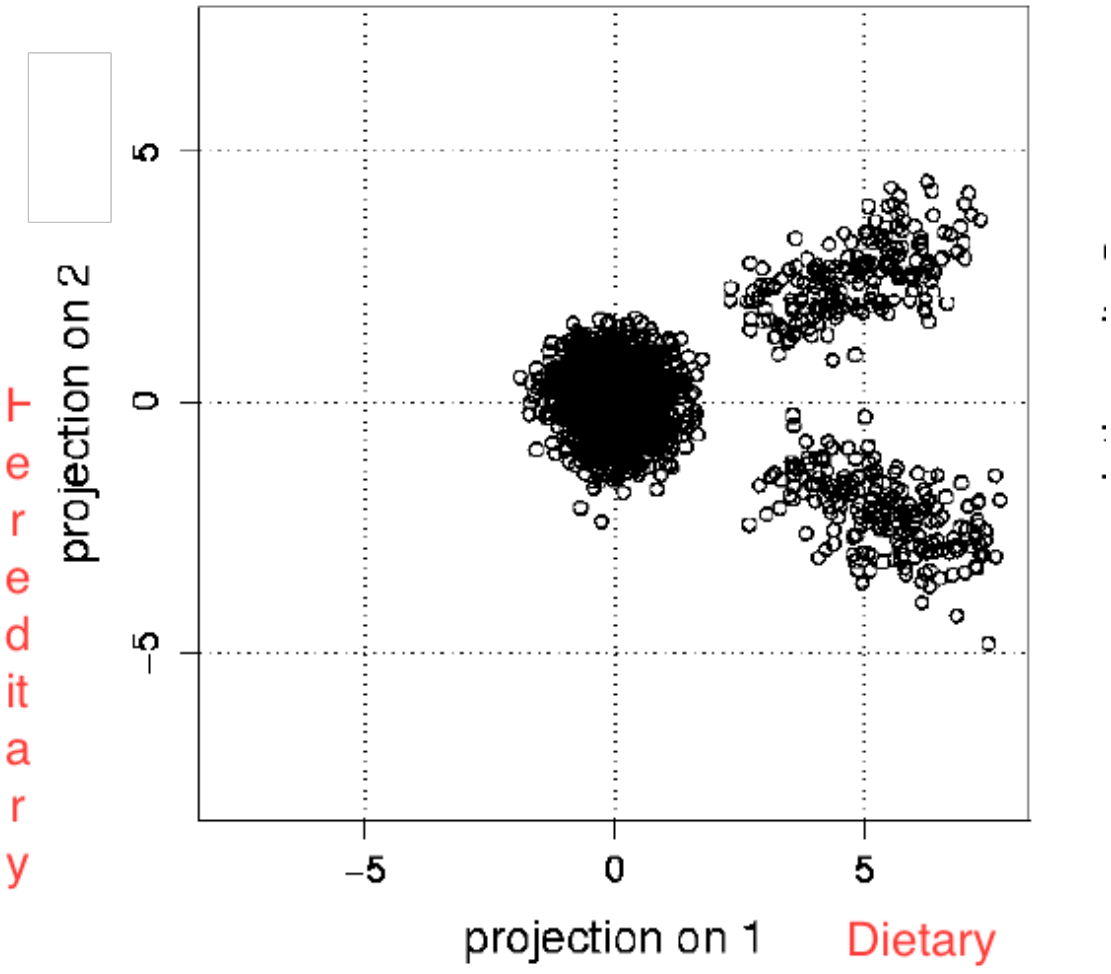
The Goal of a Factor Analysis

- To reduce the number of variables to explain and to interpret the results.

Steps

1. **Factor Extraction:** Factor extraction involves making a choice about the type of model as well the number of factors to extract.
2. **Factor Rotation:** Factor rotation after the factors are extracted aims to simplify the data structure in order to improve interpretability.

Explaining Factors in Factor Analysis



Approaches to Factor Extraction

1. Eigenvalue Analysis
2. Commonality

Factor Analysis with PC Rotation Applied to the Correlation Matrix

- The first eigenvalue of a correlation matrix indicates the maximum amount of the variance of the variables that can be accounted for with a linear transformation to a single underlying factor.
- When all correlations are positive, this first eigenvalue is approximately a linear function of the average correlation among the variables
- Factor weights (loadings) from Covariance Matrices and from Correlation Matrices are usually quite similar. If not, ill conditioning is suspect.

Communality

The communality is each variable's proportion of variability that is explained by the factors.

The communality value is the same, regardless of whether you use unrotated factor loadings or rotated factor loadings for the analysis.

Examine the communality values to assess how well each variable is explained by the factors.

The closer the communality is to 1, the better the variable is explained by the factors.

You can decide retain a factor if the factor contributes significantly to the fit of certain variables.

Rotate or Not?

Unrotated Table from Minitab

Unrotated Factor Loadings and Communalities

Variable	Factor1	Factor2	Factor3	Factor4	Communality
Academic record	0.380	0.455	0.340	0.259	0.534
Appearance	0.359	0.530	-0.040	0.523	0.685
Communication	0.465	0.660	-0.377	-0.023	0.795
Company Fit	0.523	0.677	0.266	-0.253	0.866
Experience	0.508	0.194	0.450	0.232	0.553
Job Fit	0.532	0.632	0.415	-0.201	0.895
Letter	0.992	-0.094	-0.012	-0.007	0.994
Likeability	0.412	0.529	0.032	0.377	0.593
Organization	0.406	0.761	-0.424	-0.055	0.926
Potential	0.446	0.548	0.431	0.172	0.714
Resume	0.850	0.040	0.096	0.283	0.814
Self-Confidence	0.293	0.575	0.083	0.506	0.679
Variance	3.6320	3.3193	1.0883	1.0095	9.0491
% Var	0.303	0.277	0.091	0.084	0.754

Varimax: A Non-Orthogonal Factor Rotation

Varimax criterion: Maximize the squared factor loadings in each factor.

A varimax rotation is used to simplify the expression of a particular sub-space in terms of just a few major items each. The actual coordinate system is unchanged, it is the orthogonal basis that is rotated to align with those coordinates.

Varimax is the most widely used rotation method.

Varimax Rotated Factor Analysis

Rotated Factor Loadings and Communalities

Varimax Rotation

Variable	Factor1	Factor2	Factor3	Factor4	Communality
Academic record	0.481	0.510	0.086	0.188	0.534
Appearance	0.140	0.730	0.319	0.175	0.685
Communication	0.203	0.280	0.802	0.181	0.795
Company Fit	0.778	0.165	0.445	0.189	0.866
Experience	0.472	0.395	-0.112	0.401	0.553
Job Fit	0.844	0.209	0.305	0.215	0.895
Letter	0.219	0.052	0.217	0.947	0.994
Likeability	0.261	0.615	0.321	0.208	0.593
Organization	0.217	0.285	0.889	0.086	0.926
Potential	0.645	0.492	0.121	0.202	0.714
Resume	0.214	0.365	0.113	0.789	0.814
Self-Confidence	0.239	0.743	0.249	0.092	0.679
Variance	2.5153	2.4880	2.0863	1.9594	9.0491
% Var	0.210	0.207	0.174	0.163	0.754

Interpreting Rotated Factor Loadings

- Company Fit (0.778), Job Fit (0.844), and Potential (0.645) have large positive loadings on factor 1, so this factor describes employee fit and potential for growth in the company.
- Appearance (0.730), Likeability (0.615), and Self-confidence (0.743) have large positive loadings on factor 2, so this factor describes personal qualities.
- Communication (0.802) and Organization (0.889) have large positive loadings on factor 3, so this factor describes work skills.
- Letter (0.947) and Resume (0.789) have large positive loadings on factor 4, so this factor describes writing skills.

Beyond Linear Models and Normality

Why do we use the $L2$ norm?
Why not use the $L1$ norm or any Lq norm?
Non-normal data can be approximated using Taylor expansions.
What else might you do?

PCA using the $L1$ Norm

The $L1$ norm is special.

- If the budget (constraints on the parameters) is small enough, the lasso yields sparse solution vectors, having only some coordinates that are nonzero.
- This does not occur for Lq norms with $q > 1$; for $q < 1$ where the solutions are sparse but the problem is not convex making minimization very challenging computationally.
- The value $q = 1$ is the smallest value that yields a convex problem. Convexity greatly simplifies computation, as does the sparsity assumption itself. They allow for scalable algorithms that can handle problems with even millions of parameters.

From *Statistical Learning with Sparsity The Lasso and Generalizations* by Trevor Hastie, Robert Tibshirani and Martin Wainwright. CRC Press.

PCA with Small Nonlinear Features in Big Data

David Dunson and students:
Idea based on local spherical curve fitting instead of piece-wise linear appoximation.