Linear Algebra in a Nutshell: PCA

Baroni & Evert

Introduction
Dimensionality reduction
Example data
PCA
Calculating variance
Projection
Covariance matrix
PCA algorithm

Linear Algebra in a Nutshell

Dimensions & PCA

Marco Baroni & Stefan Evert

Institute of Cognitive Science
University of Osnabrück, Germany
stefan.evert@uos.de

Rovereto, 27 March 2007

What is PCA?

- can be seen as a dimensionality reduction technique
- to find the "inherent" underlying dimensions of a data set
- exploits correlations between the variables (coordinates)
- essentially the same as SVD and LSA, but the rationale behind the procedure becomes clearer in the PCA approach

Example data set

- example: term-term word space
- cooccurrence data extracted from the BNC for nouns as direct objects of verbs *buy* and *sell*
- $k = 111$ nouns with $f \geq 20$ (which occur with either verb)
- vector coordinates are association scores (modified logarithmic Dice coefficient) ➜ $n = 2$ dimensions

noun	*buy*	*sell*
bond	0.28	0.77
cigarette	-0.52	0.44
dress	0.51	-1.30
freehold	-0.01	-0.08
land	1.13	1.54
number	-1.05	-1.02
per	-0.35	-0.16
pub	-0.08	-1.30
share	1.92	1.99
system	-1.63	-0.70

Example data set

- intuitive expecation: associations of a noun with *buy* and *sell* should be correlated (commodities tend to have high associations with both, non-commodities low associations with both)
- the main inherent dimension should be a combination of the two association scores
- the secondary dimension has a less clear interpretation and will typically be omitted from a semantic space (➜ dimensionality reduction)
- of course, real-life word spaces have many more dimensions and not just a single interesting one

The variance of a data set

- the rationale behind PCA is to find the dimensions that give the best "explanation" for the "spread" or **variance** of the data
- variance of a set of vectors (you remember the equations for one-dimensional data, right?):

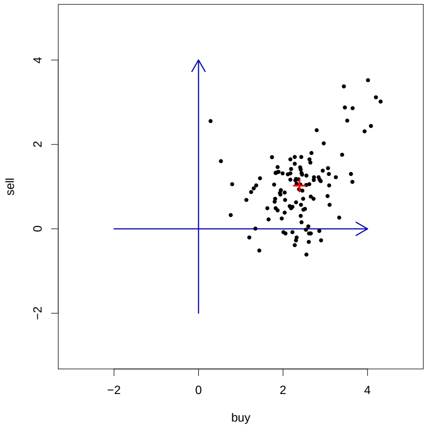
$$\sigma^2 = \frac{1}{k-1} \sum_{i=1}^{k} \|\vec{x}_i - \vec{\mu}\|^2$$

$$\vec{\mu} = \frac{1}{k} \sum_{i=1}^{k} \vec{x}_i$$

- easier to calculate if we **center** the data so that $\vec{\mu} = \vec{0}$

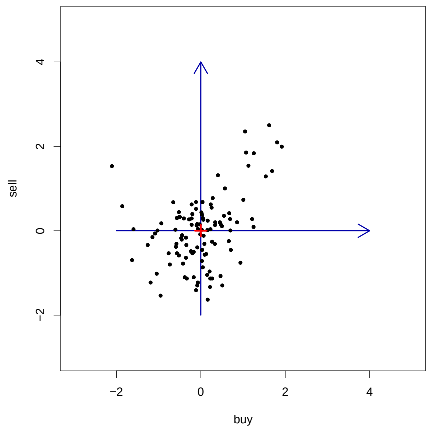
Centering the data set

- **uncentered data set**
- centered data set
- variance of centered data



Centering the data set

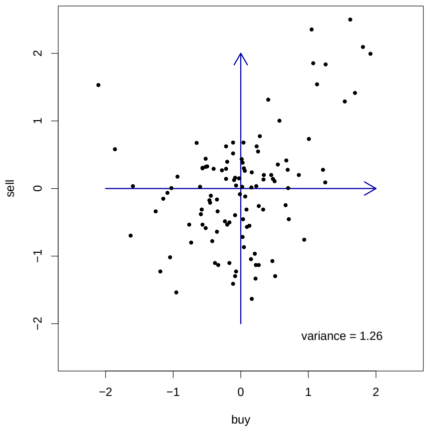
- uncentered data set
- **centered data set**
- variance of centered data



Centering the data set

- uncentered data set
- centered data set
- **variance of centered data**

$$\sigma^2 = \frac{1}{k-1} \sum_{i=1}^{k} \|\vec{x}_i\|^2$$

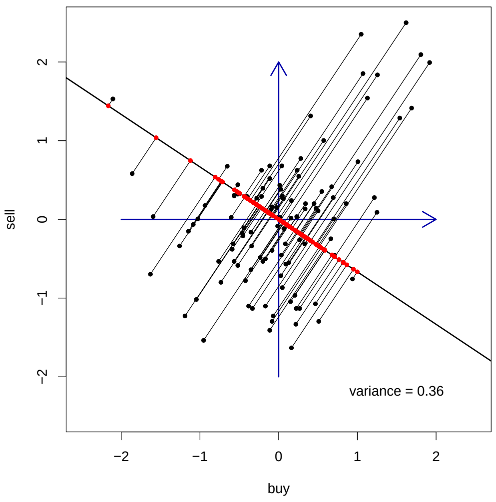


The PCA approach

- we want to reduce the dimensionality of the data without losing variance (intuitively, we want to preserve distances between the points as far as possible)
- if we reduced the data set to just a single dimension, which dimension would still have the highest variance?
- mathematically, we project the points onto a line through the origin and calculate standard variance on this line
  - we'll see in a moment how to calculate the projections
  - but first, let us look at a few examples

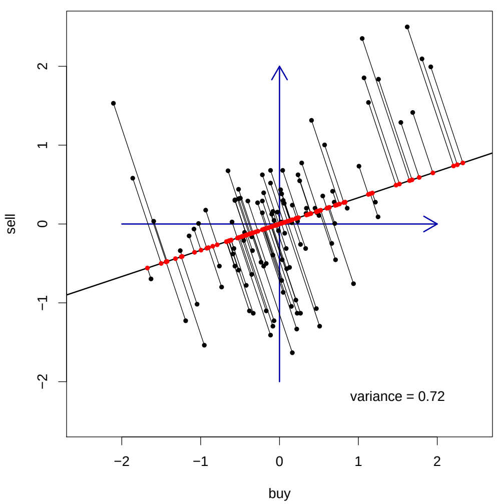
Projection and preserved variance: examples

Linear Algebra in a Nutshell: PCA
Baroni & Evert
Introduction
Dimensionality reduction
Example data
PCA
Calculating variance
Projection
Covariance matrix
PCA algorithm



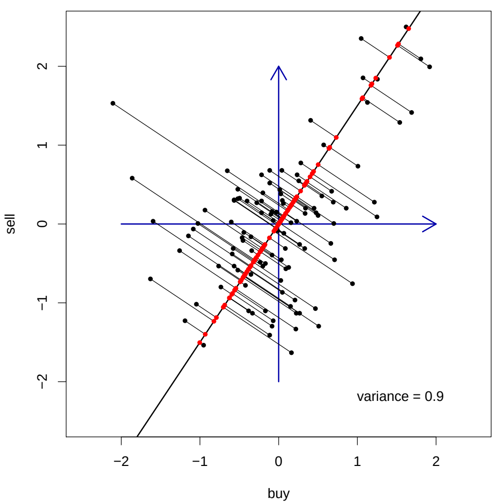
Projection and preserved variance: examples

Linear Algebra in a Nutshell: PCA
Baroni & Evert
Introduction
Dimensionality reduction
Example data
PCA
Calculating variance
Projection
Covariance matrix
PCA algorithm



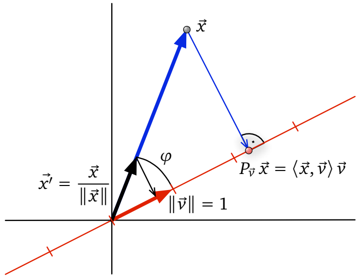
Projection and preserved variance: examples

Linear Algebra in a Nutshell: PCA
Baroni & Evert
Introduction
Dimensionality reduction
Example data
PCA
Calculating variance
Projection
Covariance matrix
PCA algorithm



The mathematics of projections

- line through origin can be described by unit vector $\|\vec{v}\| = 1$
- given a point $\vec{x}$ and the corresponding unit vector $\vec{x}' = \vec{x}/\|\vec{x}\|$, we have $\cos\varphi = \langle \vec{x}', \vec{v} \rangle$



- trigonometry: position of projected point on the line is $\|\vec{x}\| \cdot \cos\varphi = \|\vec{x}\| \cdot \langle \vec{x}', \vec{v} \rangle = \langle \vec{x}, \vec{v} \rangle$
- (projected point in original space is $\langle \vec{x}, \vec{v} \rangle\,\vec{v}$)
- amount of variance preserved = one-dimensional variance on the line (the data set is still centered)

$$\sigma_{\vec{v}}^2 = \frac{1}{k-1}\sum_{i=1}^{k} \langle \vec{x}_i, \vec{v} \rangle^2$$

Linear Algebra in a Nutshell: PCA
Baroni & Evert
Introduction
Dimensionality reduction
Example data
PCA
Calculating variance
Projection
Covariance matrix
PCA algorithm

The covariance matrix

- we want to find the direction $\vec{v}$ with maximal $\sigma_{\vec{v}}^2$
- simplify the repeated calculation of $\sigma_{\vec{v}}^2$

$$\begin{aligned}
\sigma_{\vec{v}}^2 &= \frac{1}{k-1}\sum_{i=1}^{k} \langle \vec{x}_i, \vec{v} \rangle^2 \\
&= \frac{1}{k-1}\sum_{i=1}^{k} \left(\vec{x}_i^T \cdot \vec{v}\right)^T \cdot \left(\vec{x}_i^T \cdot \vec{v}\right) \\
&= \frac{1}{k-1}\sum_{i=1}^{k} \vec{v}^T \cdot \left(\vec{x}_i \cdot \vec{x}_i^T\right) \cdot \vec{v} \\
&= \vec{v}^T \cdot \underbrace{\left(\frac{1}{k-1}\sum_{i=1}^{k} \vec{x}_i \cdot \vec{x}_i^T\right)}_{=:C} \cdot \vec{v} \\
&= \vec{v}^T \cdot C \cdot \vec{v}
\end{aligned}$$

Linear Algebra in a Nutshell: PCA
Baroni & Evert
Introduction
Dimensionality reduction
Example data
PCA
Calculating variance
Projection
Covariance matrix
PCA algorithm

The covariance matrix

- C is the **covariance matrix** of the data points
  - C is a square $n \times n$ matrix (2×2 in our example)
- preserved variance after projection onto a line $\vec{v}$ can easily be calculated as $\sigma_{\vec{v}}^2 = \vec{v}^T C \vec{v}$
- the original variance of the data set is $\sigma^2 = \mathrm{tr}(C) = C_{11} + C_{22} + \cdots + C_{nn}$

$$C = \begin{pmatrix}
\sigma_1^2 & C_{12} & \cdots & C_{1n} \\
C_{21} & \sigma_2^2 & \ddots & \vdots \\
\vdots & \ddots & \ddots & C_{n-1,n} \\
C_{n1} & \cdots & C_{n,n-1} & \sigma_n^2
\end{pmatrix}$$

Linear Algebra in a Nutshell: PCA
Baroni & Evert
Introduction
Dimensionality reduction
Example data
PCA
Calculating variance
Projection
Covariance matrix
PCA algorithm

Maximizing the preserved variation

- in our data, we want to find the axis $\vec{v}_1$ that preserves the largest amount of variation by maximizing $\vec{v}_1^T C \vec{v}_1$
- for higher-dimensional data set, we also want to find the axis $\vec{v}_2$ of second highest variation, etc.
- this has to be constrained: $\vec{v}_2$ must be orthogonal to $\vec{v}_1$, i.e. $\langle \vec{v}_1, \vec{v}_2 \rangle = 0$ (and the same for $\vec{v}_3$ etc.)
- we can easily solve this problem using a result from linear algebra: since C is a symmetric matrix ($C^T = C$), it has an **eigenvalue decomposition** with orthogonal **eigenvectors** $\vec{a}_1, \vec{a}_2, \ldots, \vec{a}_n$ and corresponding eigenvalues $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_n$

The eigenvalue decomposition of C

- the eigenvalue decomposition of C can also be written in the form

$$C = U \cdot D \cdot U^T$$

where U is an orthogonal matrix containing the eigenvectors as columns and $D = \mathrm{Diag}(\lambda_1, \dots, \lambda_n)$ a diagonal matrix of eigenvalues

$$U = \begin{pmatrix} \vdots & \vdots & & \vdots \\ \vdots & \vdots & & \vdots \\ \vec{a}_1 & \vec{a}_2 & \cdots & \vec{a}_n \\ \vdots & \vdots & & \vdots \\ \vdots & \vdots & & \vdots \end{pmatrix} \qquad D = \begin{pmatrix} \lambda_1 & & & & \\ & \lambda_2 & & & \\ & & \ddots & & \\ & & & \ddots & \\ & & & & \lambda_n \end{pmatrix}$$

  - note that both U and D are $n \times n$ square matrices

The PCA algorithm

- now we have $\sigma_{\vec{v}}^2 = \vec{v}^T C \vec{v} = \vec{v}^T \cdot UDU^T \cdot \vec{v} = (U^T \vec{v})^T \cdot D \cdot (U^T \vec{v}) = (\vec{y})^T D \vec{y}$
- $\vec{y} = U^T \vec{v} = [y_1, y_2, \dots, y_n]^T$ are the coordinates of $\vec{v}$ according to the basis of eigenvectors of C
- $\|\vec{y}\| = 1$ since orthogonal U^T is an isometry
- we want to maximize

$$\vec{v}^T C \vec{v} = \lambda_1 (y_1)^2 + \lambda_2 (y_2)^2 \cdots + \lambda_n (y_n)^2$$

under the constraint $(y_1)^2 + (y_2)^2 + \cdots + (y_n)^2 = 1$
- the obvious solution is $\vec{y} = [1, 0, \dots, 0]^T$, since λ_1 is the largest eigenvalue
- this corresponds to $\vec{v} = \vec{a}_1$, the first eigenvector of C, and a preserved variance of $\sigma_{\vec{v}}^2 = \vec{a}_1^T C \vec{a}_1 = \lambda_1$

The PCA algorithm

- in order to find the dimension of second highest variance, we have to look for an axis $\vec{v}$ orthogonal to $\vec{a}_1$
- since U^T is an orthogonal matrix, the coordinates $\vec{y} = U^T \vec{v}$ have to be orthogonal to the first axis $[1, 0, \dots, 0]^T$, i.e. $\vec{y} = [0, y_2, \dots, y_n]^T$
- in other words, we have to maximize

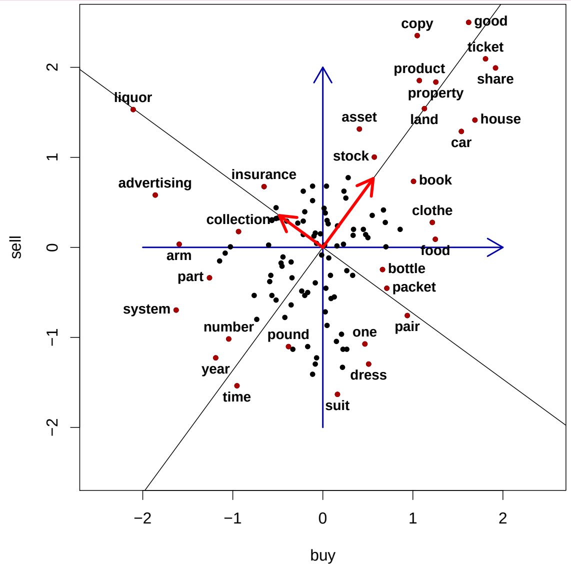
$$\vec{v}^T C \vec{v} = \lambda_2 (y_2)^2 \cdots + \lambda_n (y_n)^2$$

under constraints $y_1 = 0$ and $(y_2)^2 + \cdots + (y_n)^2 = 1$
- again, the obvious solution is $\vec{y} = [0, 1, 0, \dots, 0]^T$, corresponding to $\vec{v} = \vec{a}_2$, the second eigenvector of C, and a preserved variance of $\sigma_{\vec{v}}^2 = \lambda_2$
- similarly for the third, fourth, ... axis

The PCA algorithm

- the eigenvectors $\vec{a}_i$ of the covariance matrix C are called the **principal components** of the data set
- the amount of variance preserved (or "explained") by the i-th principal component is given by the eigenvalue λ_i
- since $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_n$, the first principal component preserves the largest amount of variation etc.
- coordinates of a point $\vec{x}$ in PCA space are given by $U^T \vec{x}$ (note: these are the projections on the principal components)
- for the purpose of dimensionality reduction, only the first l principal components (with highest variance) are retained, and the other dimensions in PCA space are dropped

PCA example



PCA in R

```
> pca <- prcomp(M)

> print(summary(pca))
Importance of components:
                          PC1   PC2
Standard deviation     0.947 0.599
Proportion of Variance 0.715 0.285
Cumulative Proportion  0.715 1.000

> print(pca)
Standard deviations:
[1] 0.9471326 0.5986067

Rotation:
            PC1        PC2
buy  -0.5907416  0.8068608
sell -0.8068608 -0.5907416
```


PCA in R

```
> head(pca$x)
                   PC1        PC2
acre        -0.1637281  0.1641069
advertising  0.6304871 -1.8450086
amount       0.7650502 -0.1411030
arm          0.9141092 -1.3080504
asset       -1.3005140 -0.4489060
bag          1.2044251  0.7407619
```