Principal Components

Suppose we have N measurements on each of p variables $X_j,\ j = 1, \ldots, p$. There are several equivalent approaches to principal components:

- Produce a derived (and small) set of uncorrelated variables $Z_k = \alpha_k^T X,\ k = 1, 2, \ldots, q < p$ that are linear combinations of the original variables, and that explain most of the variation in the original data.

- Approximate the original set of N points in $\mathbb{R}^p$ by a least-squares optimal linear manifold of dimension $q < p$.

- Approximate the $N \times p$ data matrix $\mathbf{X}$ by the best rank-q matrix $\hat{\mathbf{X}}_{(q)}$. This is the usual motivation for the SVD.

Principal components and Variance

If X is a random vector with mean 0 and covariance matrix $\mathbf{\Sigma}$, then the variance of the linear combination $Z = \alpha^T X$ is given by

$$\mathrm{Var}(Z) = \alpha^T \mathbf{\Sigma} \alpha.$$

We are seeking an α such that $\mathrm{Var}(Z)$ is large; clearly we must impose a scale restriction on α. This leads to the principal-component criterion

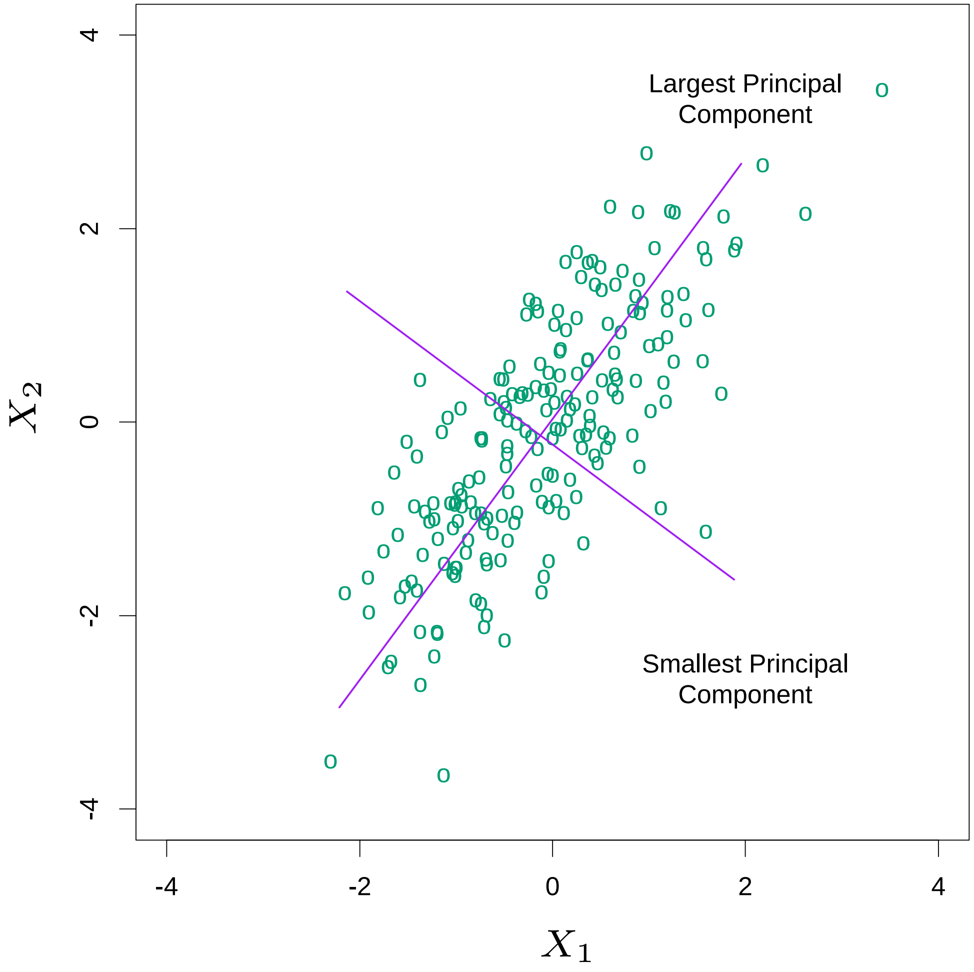
$$\max_{\alpha} \alpha^T \mathbf{\Sigma} \alpha \text{ subject to } ||\alpha|| = 1$$

The solution α is the largest eigenvector of $\mathbf{\Sigma}$:

$$\mathbf{\Sigma}\alpha = d^2 \alpha,$$

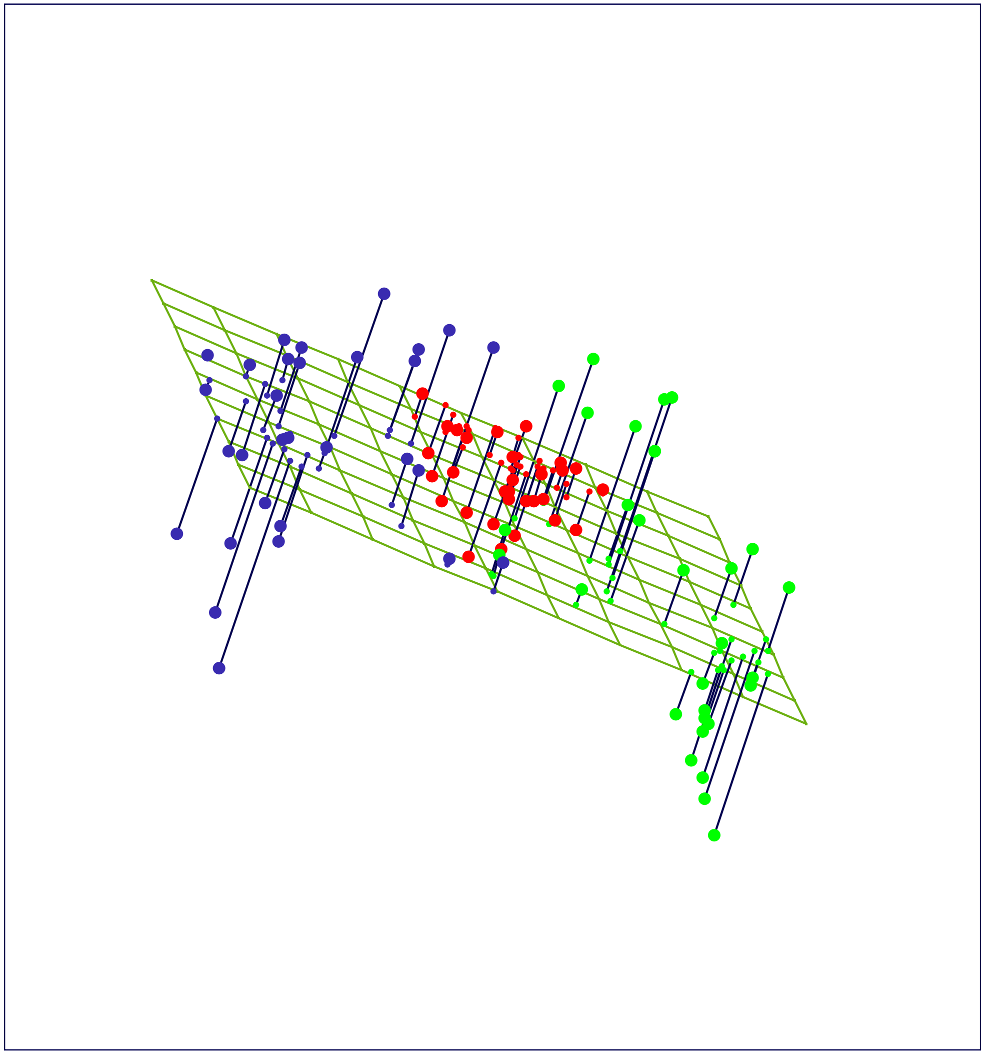
and $\mathrm{Var}(Z) = \mathrm{Var}(\alpha^T X) = d^2$.

PC: Derived Variables



$Z_1 = \alpha_1^T X$ is the projection of the data onto the longest direction, and has the largest variance amongst all such normalized projections. α_1 is the largest eigenvalue of $\hat{\mathbf{\Sigma}}$, the sample covariance matrix of X. Z_2 and α_2 correspond to the second-largest eigenvector.

PC: Least Squares Approximation



- Find the linear manifold $f(\lambda) = \mu + \mathbf{V}_q\lambda$ that best approximates the data in a least-squares sense:

$$\min_{\mu,\{\lambda_i\},\,\mathbf{V}_q} \sum_{i=1}^{N} \|x_i - \mu - \mathbf{V}_q\lambda_i\|^2.$$

- Solution: $\mu = \bar{x}$, $v_k = \alpha_k$, $\lambda_k = Z_k$.
- Note: solution minimizes mean orthogonal (squared) distances

PC: Singular Value Decomposition

For any $N \times p$ data matrix $\mathbf{X}$ (assume $N > p$).

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

is the *SVD* of $\mathbf{X}$, where

- $\mathbf{U}$ is $N \times p$ orthogonal, the left singular vectors.
- $\mathbf{V}$ is $p \times p$ orthogonal, the right singular vectors.
- $\mathbf{D}$ is diagonal, with $d_1 \geq d_2 \geq \ldots \geq d_p \geq 0$, the singular values.

The SVD always exists, and is unique (up to signs and ties).

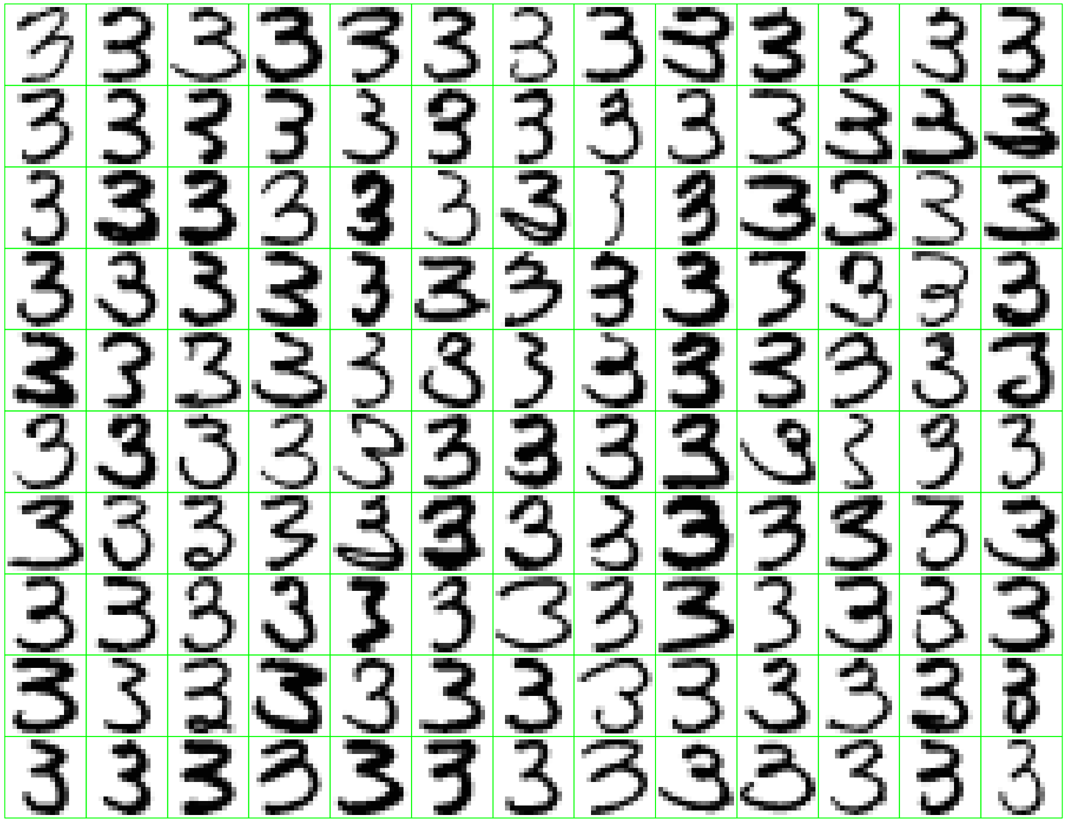
If $\mathbf{X}$ is centered (column means zero), then the columns of $\mathbf{V}$ are the principal components, and $Z_j = U_j d_j$.

SVD: best rank q approximation

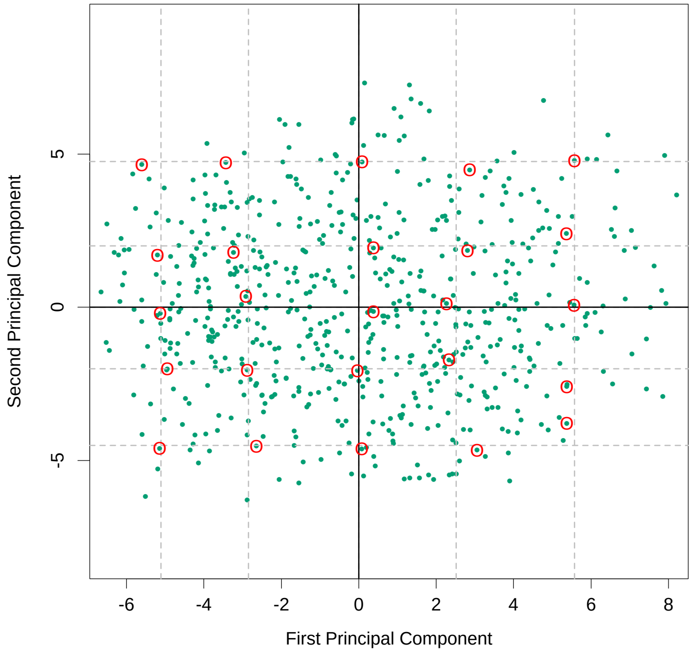
Let $\mathbf{D}_q$ be $\mathbf{D}$, with all but the first q diagonal elements set to zero. Then $\hat{\mathbf{X}}_q = \mathbf{U}\mathbf{D}_q\mathbf{V}^T$ solves

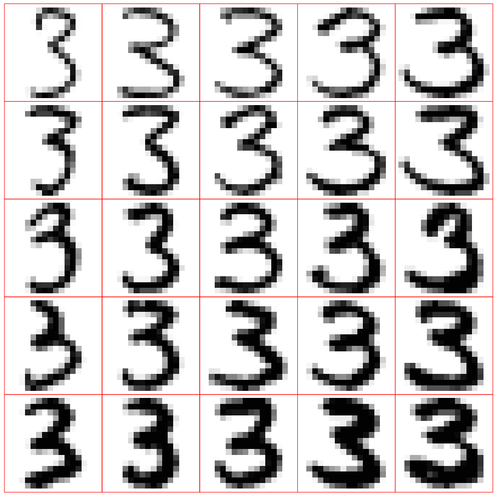
$$\min_{\text{rank}(\hat{\mathbf{X}}_q)=q} ||\mathbf{X} - \hat{\mathbf{X}}_q||_F$$

PC: Example — Digit Data



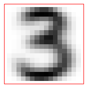
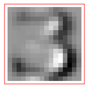
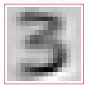
130 threes, a subset of 638 such threes and part of the **handwritten digit** dataset. Each three is a 16 × 16 greyscale image, and the variables X_j, $j = 1, \ldots, 256$ are the greyscale values for each pixel.



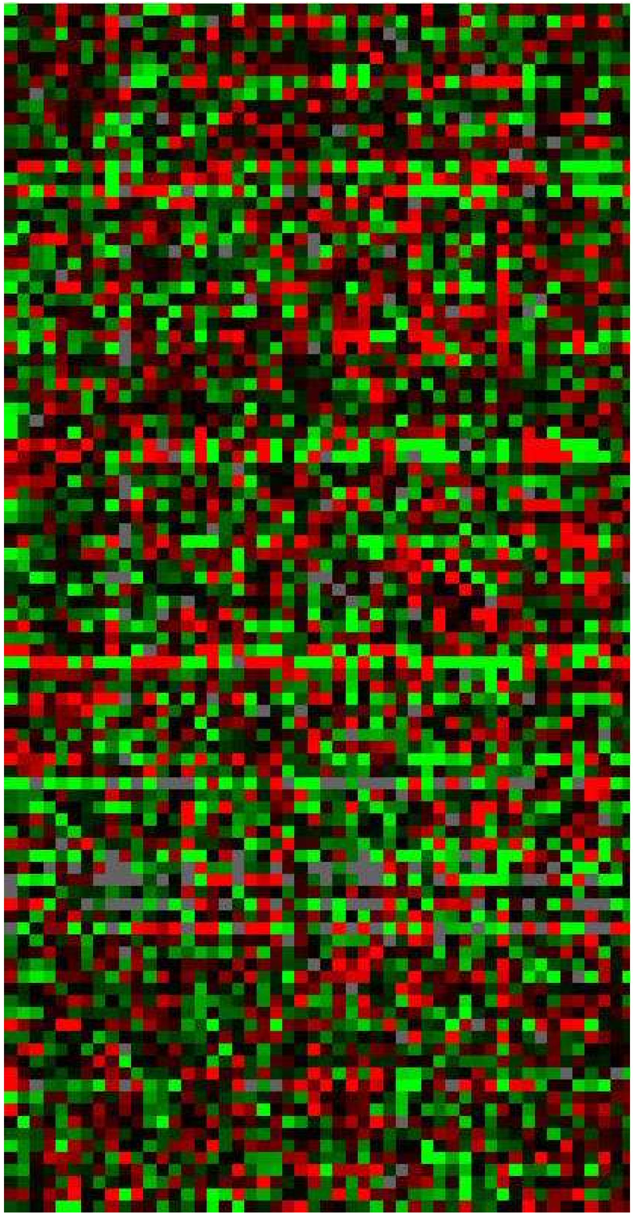


Two-component model has the form

$$\hat{f}(\lambda) = \bar{x} + \lambda_1 v_1 + \lambda_2 v_2$$

=  $+ \lambda_1 \cdot$  $+ \lambda_2 \cdot$ .

Here we have displayed the first two principal component directions, v_1 and v_2, as images.



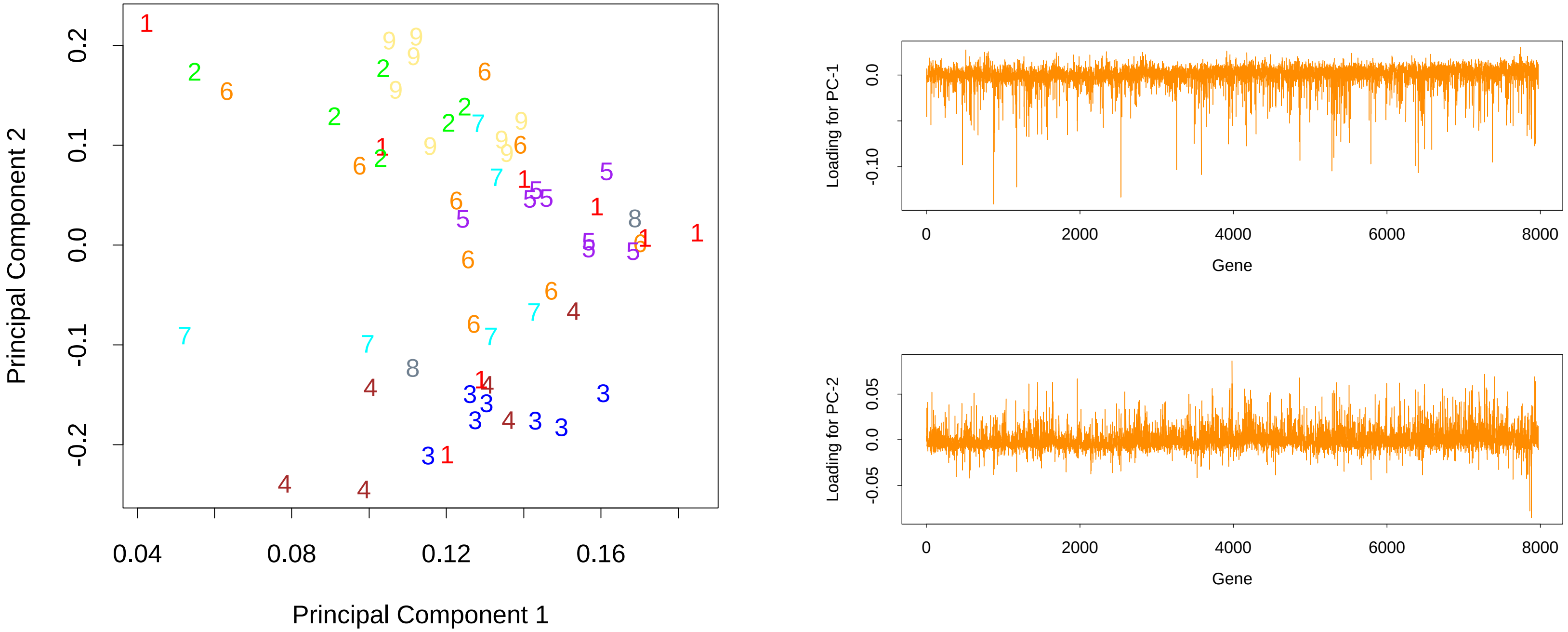
SVD: Expression Arrays

The rows are genes (variables) and the columns are observations (samples, DNA arrays). Typicall numbers are 6-10K genes, 50-150 samples.

Eigengenes

- The first principal component or *eigengene* is the linear combination of the genes showing the most variation over the samples.
- The individual gene loadings for each eigengene or *eigenarrays* can have biological meaning.
- The sample values for the eigengenes show useful low-dimensional projections.

Example: NCI Cancer Data



First two *eigengenes* (left) and *eigenarrays* (right). Points are colored according to NCI cancer classes

Sparse principal components

The principal component *coefficient* vector α_k defines Z_k by the sign and size of the variable *loadings* α_{kj}. All the elements are nonzero, although often many are small.

For interpretability, it is convenient for the α_k to be sparse.

We define sparse principal components (Jolliff et al, 2003) via

$$\max_{\alpha} \mathrm{Var}(\alpha^T X) \ \text{ subject to } ||\alpha_2|| = 1, \ ||\alpha||_1 = \gamma$$

for $1 \leq \gamma \leq \sqrt{p}$.

Although this criterion is not convex, a simple alternating algorithm finds good solutions.

Sparse PCA via the power method

For ordinary PCs, the power method is as follows:

$$\alpha^{(m+1)} \leftarrow \frac{\mathbf{X}^T\mathbf{X}\alpha^{(m)}}{||\mathbf{X}^T\mathbf{X}\alpha^{(m)}||_2}$$

and converges to the largest principal component.

The following modification finds stationary points for the sparse PC problem (Witten, Tibshirani & Hastie 2009)

$$\alpha^{(m+1)} \leftarrow \frac{S(\mathbf{X}^T\mathbf{X}\alpha^{(m)}, \lambda)}{||S(\mathbf{X}^T\mathbf{X}\alpha^{(m)}, \lambda)||_2}$$

where the *soft-thresholding operator* $S(a, \lambda) = \text{sign}(a)(|a| - \lambda)_+$ is applied coordinatewise. λ is found so that $||\alpha^{(m+1)}||_1 = \gamma$.

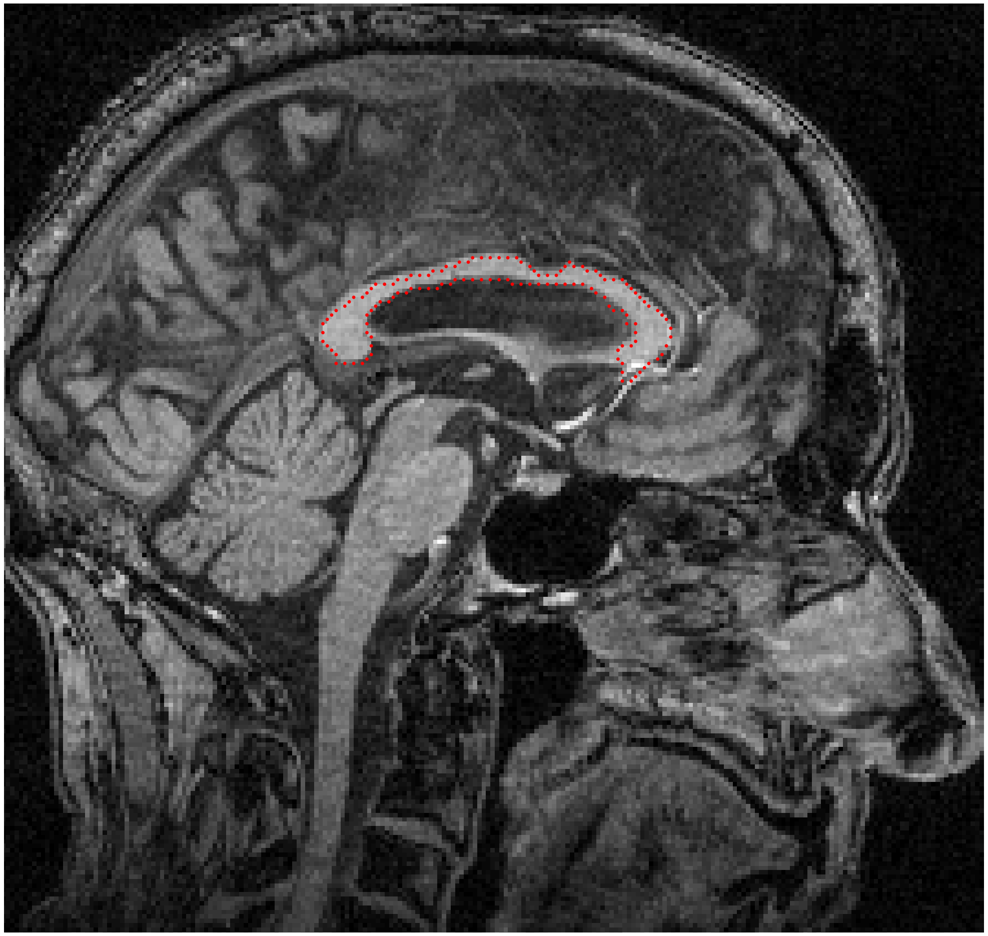
PMA package on CRAN does sparse PCA and CCA (Daniela Witten, Ph.D thesis)

Example of Sparse PCA: Morphometrics

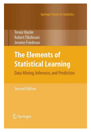
Morhometrics is the study of shapes. In this example, the shape of the *corpus callosum* (part of the brain) in elderly subjects.

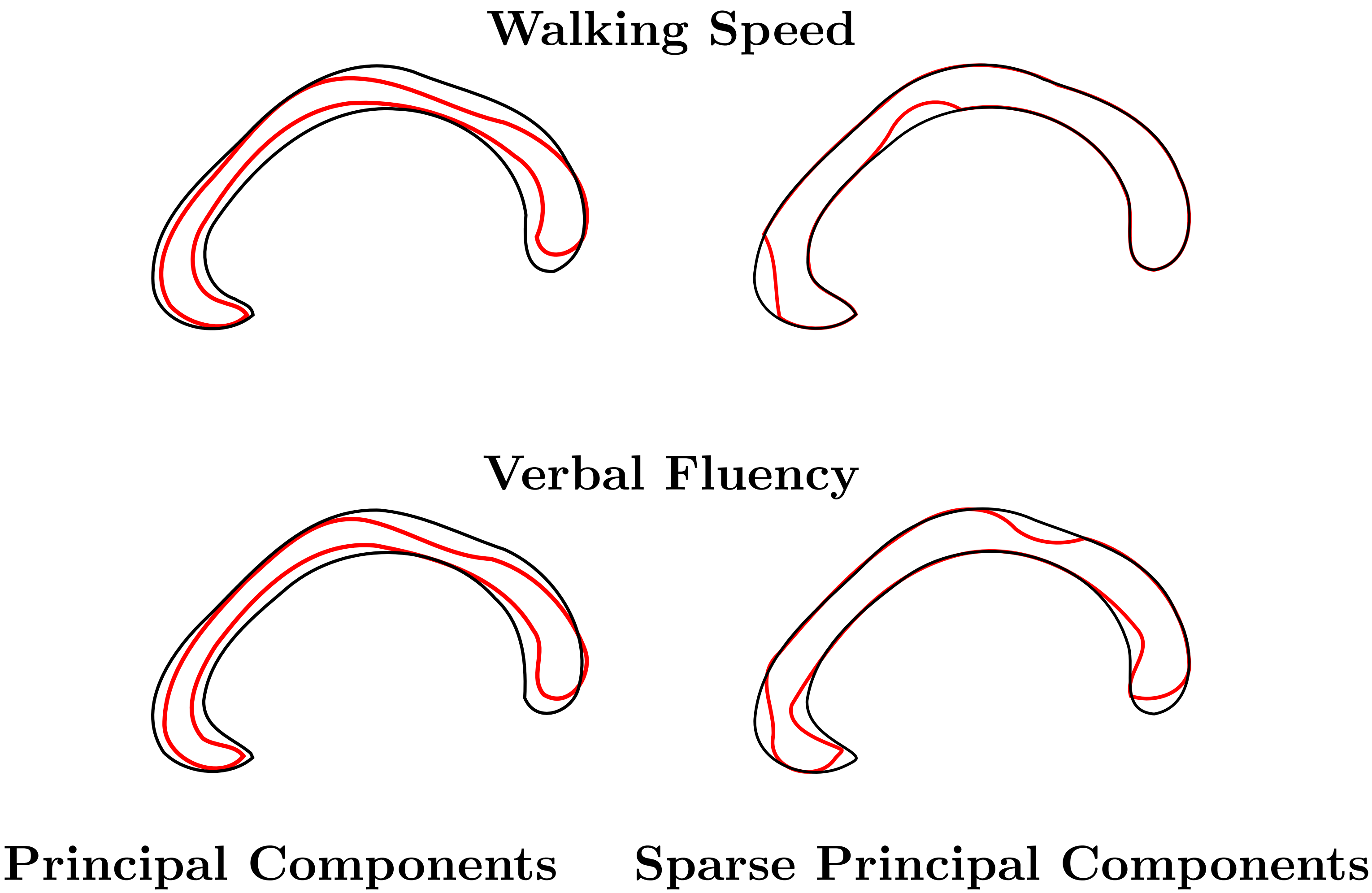
Each shape is sampled at a set of *corresponding points* along its perimeter. The shapes are rotated to line up (Procrustes transformation). The coordinates (x,y pairs) are collapsed into a long vector, and PCA (sparse) is performed on the collection of shapes.

The PCs indicate areas of variability among the shapes.



Mid-saggital brain slice, with corpus callosum in red. (Larsen and Sjöstrand, 2007).

See  pp 550.



Walking Speed

Verbal Fluency

Principal Components

Sparse Principal Components

SVD with Missing Data

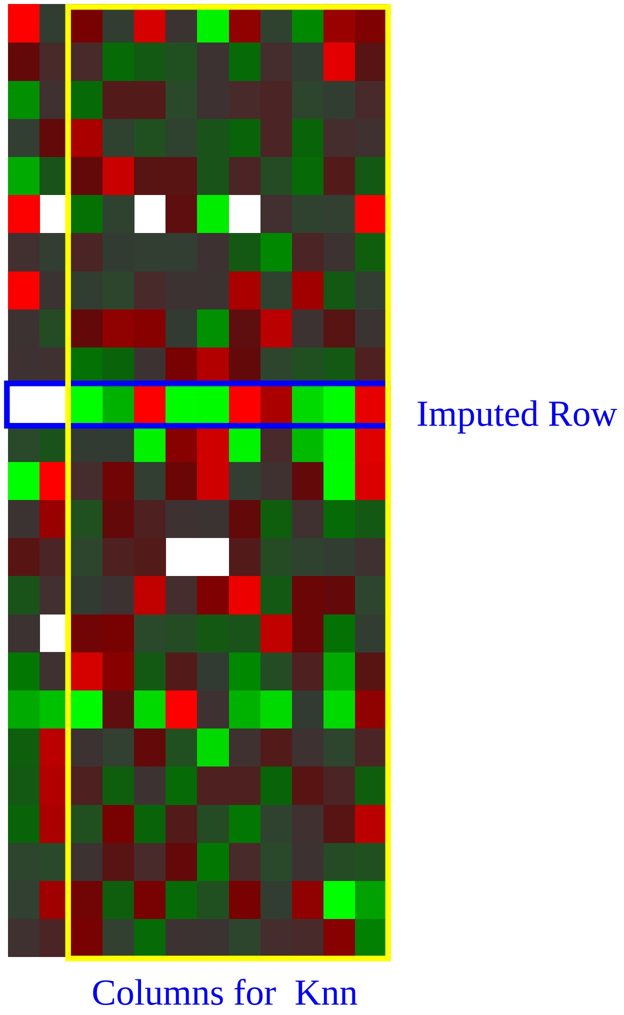
Very often we have missing data—some observations are missing values for one or more variables. We outline two approaches:

- Impute all the missing values and then compute the SVD of the complete matrix. Imputation schemes range from:
  - filling in missing values for a variable using the mean of all non-missing entries.
  - Using a sophisticated regression procedure for imputation.
- Compute the rank-q SVD and impute the missing values at the same time.

The former is convenient, since the matrix is completed once, and the entire SVD spectrum is computed for the same matrix. The latter is attractive since it appears to use the latent structure learned by the SVD to help with the imputation.

Imputation by regression

- Regress each variable on all the others, and predict the missing values. For this purpose, CART and k-nearest neighbor regression are especially useful, since they themselves can handle missing data very well.
  - CART develops a set of *surrogate variables* and splits at each node, which can be used if an observation is missing on the primary split variable.
  - k-nearest neighbors: the distance metric can simply ignore the missing variables when computing the nearest neighbors.
- Iterated regressions. As in the previous item, except missing data are initialized (for example, at the variable means). Now any regression procedure can be used, treating each variable as the response, and refining the missing values. This is iterated.



Example: Knn imputation

- For each missing row, use the columns corresponding to the non-missing entries as inputs to a 10-NN regression
- Replace the missing values by the means of the corresponding columns of the 10-NN

Matrix Completion Problem

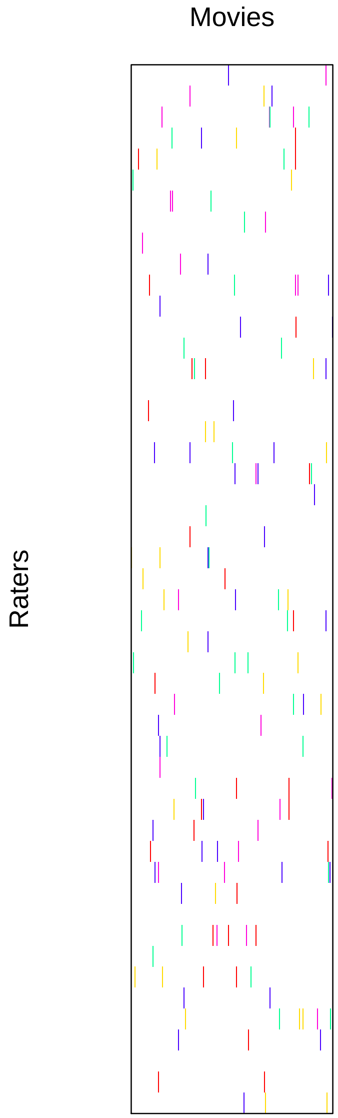
Impute missing entries in a matrix via low-rank approximations.

We pose the following optimization problem:

$$\min_{\text{rank}(\hat{\mathbf{X}}_q)=q} ||P_\Omega(\mathbf{X}) - P_\Omega(\hat{\mathbf{X}})||$$

where P_Ω is a projection onto the observed values (sets unobserved to 0, and hence skips over the missing data.) The following simple algorithm solves this problem:

1. Inititialize the missing entries with columns means.
2. Compute the rank-q approximation, and re-fill in the missing values.
3. Repeat step 2 till convergence.



Netflix Competition

- Large rating matrix — 400K raters by 18K movies
- Each row very sparse, with all but ≈ 100 entries missing. Entries are ratings 1–5.
- Many of the Netflix leaders use some form of SVD to impute/fill in missing entries.
- Low rank SVD puts movies into genres, and raters into cliques.
- Iterative Lanczos methods allow low-rank approximations without ever storing the 7 billion entries.

ℓ_1 regularized SVD

$$\min_{\hat{\mathbf{X}}} ||P_\Omega(\mathbf{X}) - P_\Omega(\hat{\mathbf{X}})||_F^2 + \lambda||\hat{\mathbf{X}}||_*$$

- $||\hat{\mathbf{X}}||_*$ is *nuclear norm* — sum of singular values.
- This is a convex optimization problem (Candes 2008), with solution given by a *soft thresholded* SVD — singular values are shrunk toward zero, many set to zero.
- Our algorithm iteratively soft-thresholds the SVD of

$$\begin{aligned} P_\Omega(\mathbf{X}) + P_\Omega^\perp(\hat{\mathbf{X}}) &= \left\{P_\Omega(\mathbf{X}) - P_\Omega(\hat{\mathbf{X}})\right\} + \hat{\mathbf{X}} \\ &= \text{Sparse} + \text{Low-Rank} \end{aligned}$$

- Using Lanczos techniques and warm starts, we can efficiently compute solution paths for very large matrices (50K ×50K). (Rahul Mazumbder, Ph.D thesis)