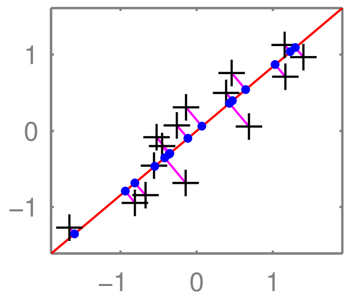
PCA: Principal Component Analysis

Iain Murray

`http://iainmurray.net/`

PCA: Principal Component Analysis



K = 1

+ = X

• = Xproj

— = V(:,1)

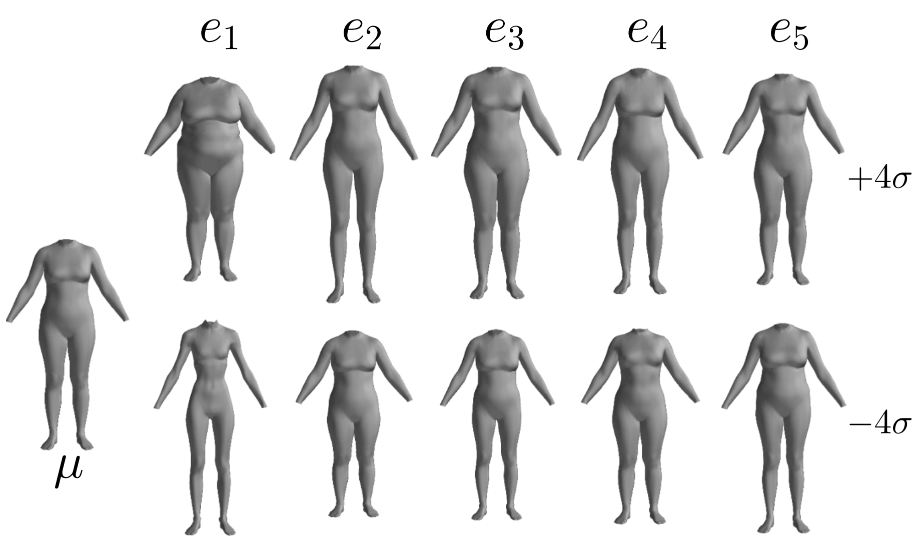
Code assuming X is zero-mean

```
% Find top K principal directions:
[V, E] = eig(X'*X);
[E,id] = sort(diag(E),1,'descend');
V = V(:, id(1:K));   % DxK

% Project to K-dims:
X_kdim = X*V;   % NxK

% Project back:
X_proj = X_kdim * V';   % NxD
```

PCA applied to bodies



Freifeld and Black, ECCV 2012

PCA applied to DNA

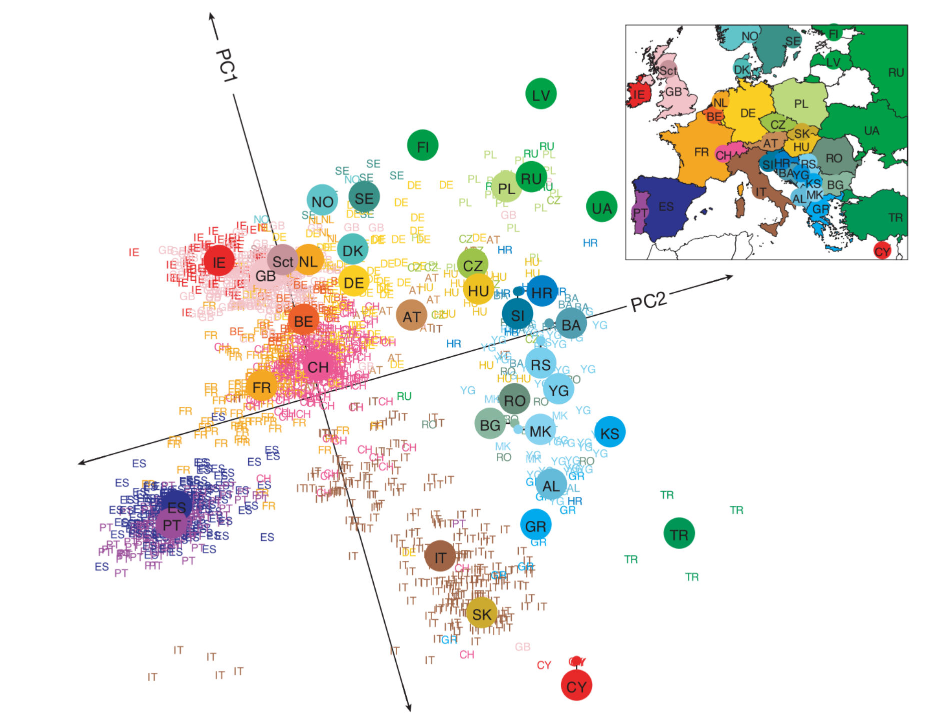
Novembre et al. (2008) — doi:10.1038/nature07331

Carefully selected both individuals and features

1,387 individuals

197,146 single nucleotide polymorphisms (SNPs)

Each person reduced to two(!) numbers with PCA



MSc course enrollment data

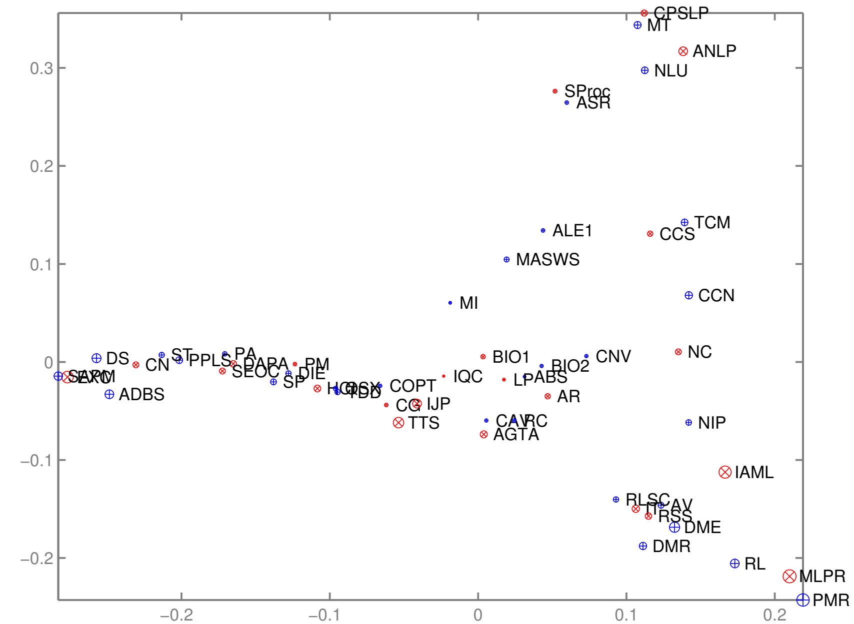
Binary $S \times C$ matrix M

$M_{sc} = 1$, if student s taking course c

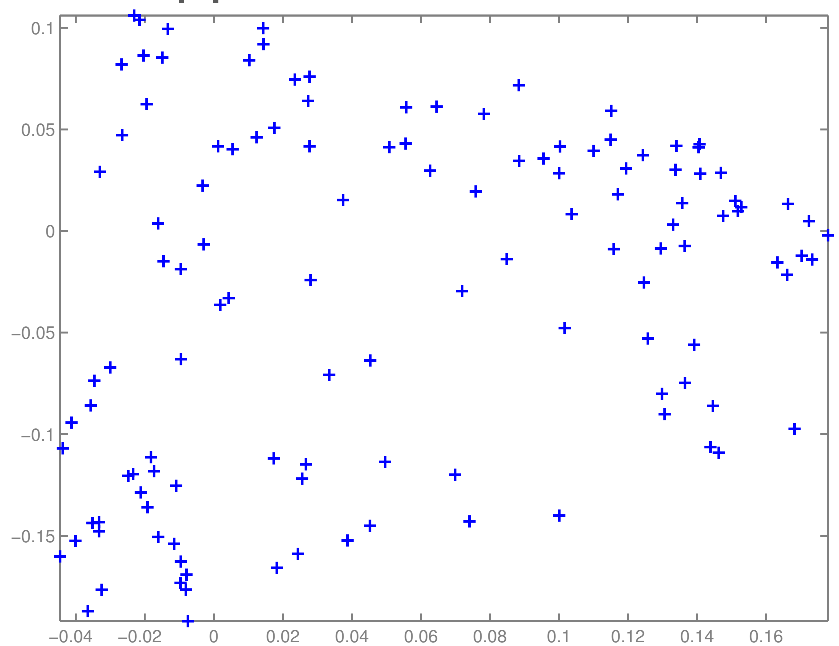
Each course is a length S vector

... OR each student is a length C vector

PCA applied to MSc courses



PCA applied to MSc students



Truncated SVD

```
% PCA via SVD,
% for zero-mean X:
[U, S, V] = svd(X, 0);
U = U(:, 1:K);
S = S(1:K, 1:K);
V = V(:, 1:K);
X_kdim = U*S;
X_proj = U*S*V';
```

$$\begin{bmatrix} X_{11} & X_{12} & \cdots & X_{1D} \\ X_{21} & X_{22} & \cdots & X_{2D} \\ X_{31} & X_{32} & \cdots & X_{3D} \\ X_{41} & X_{42} & \cdots & X_{4D} \\ X_{51} & X_{52} & \cdots & X_{5D} \\ \vdots & \vdots & \ddots & \vdots \\ X_{N1} & X_{N2} & \cdots & X_{ND} \end{bmatrix} \approx$$

$$\begin{bmatrix} U_{11} & \cdots & U_{1K} \\ U_{21} & \cdots & U_{2K} \\ U_{31} & \cdots & U_{3K} \\ U_{41} & \cdots & U_{4K} \\ U_{51} & \cdots & U_{5K} \\ \vdots & \ddots & \vdots \\ U_{N1} & \cdots & U_{NK} \end{bmatrix} \begin{bmatrix} S_{11} & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & S_{KK} \end{bmatrix} \begin{bmatrix} V_{11} & V_{21} & \cdots & V_{D1} \\ \vdots & \vdots & \ddots & \vdots \\ V_{1K} & V_{2K} & \cdots & V_{DK} \end{bmatrix}$$

$$X \approx U \quad S \quad V^\top$$

PCA summary

Project data onto major axes of covariance

$X^\top X$ is covariance if make data zero mean

Low-dim coordinates can be useful:

— visualization

— if can't cope with high-dim data

Can project back into original space:

— detail is lost: still in K-dim subspace

— PCA minimizes the square error

PPCA: Probabilistic PCA

Gaussian model: $\Sigma = WW^\top + \sigma^2 I$

W is $D \times K$, σ^2 small $\Rightarrow$ nearly low-rank

W is also orthogonal

As $\sigma^2 \to 0$, recover PCA.

Need $\sigma^2 > 0$ to explain data

Special case of factor analysis: $\Sigma = WW^\top + \Phi$, with Φ diagonal

Dim reduction in other models

Can replace $\mathbf{x}$ with $A\mathbf{x}$ in any model

A is a $K \times D$ matrix of projection params

Large D: a lot of extra parameters

NB: Neural nets already have such projections

Practical tip

Scale features to have unit variance

Equivalently: find eigenvectors of correlation rather than covariance

Avoids issues with (arbitrary?) scaling.

If multiply feature by 10^9, PC points along that feature

E.g., if change unit of feature from metres to nanometres