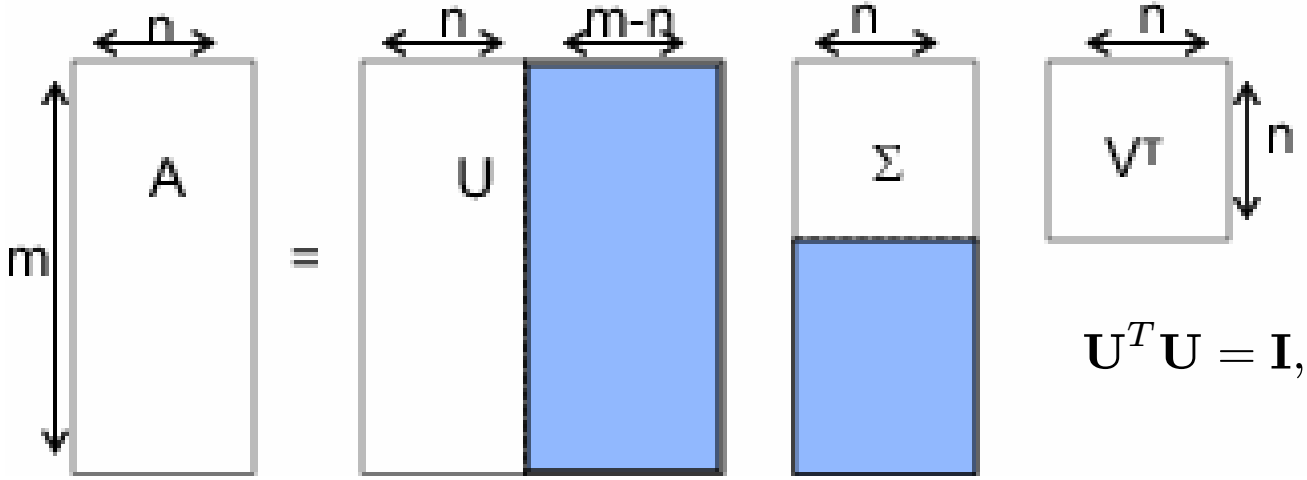
SVD; PCA

Singular Value Decomposition

$$\mathbf{A} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T = \lambda_1 \begin{pmatrix} | \\ \mathbf{u}_1 \\ | \end{pmatrix} \begin{pmatrix} - & \mathbf{v}_1^T & - \end{pmatrix} +$$

Gene Golub's plate

$$\cdots + \lambda_r \begin{pmatrix} | \\ \mathbf{u}_r \\ | \end{pmatrix} \begin{pmatrix} - & \mathbf{v}_r^T & - \end{pmatrix}$$



$$\mathbf{U}^T\mathbf{U} = \mathbf{I}, \mathbf{V}^T\mathbf{V} = \mathbf{V}\mathbf{V}^T = \mathbf{I}$$

Right svectors are evecs of A^T A

- For any matrix A

$$\begin{aligned}
\mathbf{A}^T\mathbf{A} &= \mathbf{V}\boldsymbol{\Sigma}^T\mathbf{U}^T\,\mathbf{U}\boldsymbol{\Sigma}\mathbf{V}^T \\
&= \mathbf{V}(\boldsymbol{\Sigma}^T\boldsymbol{\Sigma})\mathbf{V}^T \\
(\mathbf{A}^T\mathbf{A})\mathbf{V} &= \mathbf{V}(\boldsymbol{\Sigma}^T\boldsymbol{\Sigma}) = \mathbf{V}\mathbf{D}
\end{aligned}$$

Left svectors are evecs of A A^T

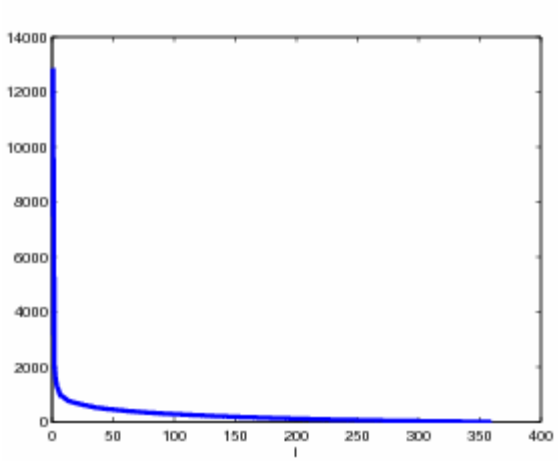
$$\begin{aligned} \mathbf{A}\mathbf{A}^T &= \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T\ \mathbf{V}\mathbf{\Sigma}^T\mathbf{U}^T \\ &= \mathbf{U}(\mathbf{\Sigma}\mathbf{\Sigma}^T)\mathbf{U}^T \\ (\mathbf{A}\mathbf{A}^T)\mathbf{U} &= \mathbf{U}(\mathbf{\Sigma}\mathbf{\Sigma}^T) = \mathbf{U}\mathbf{D} \end{aligned}$$

Truncated SVD

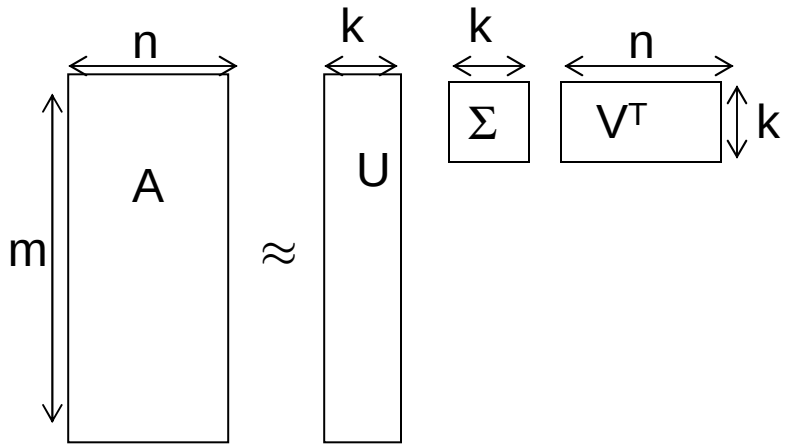
$$\mathbf{A} = \mathbf{U}_{:,1:k}\mathbf{\Sigma}_{1:k,1:k}\mathbf{V}^T_{1:,1:k} = \lambda_1 \begin{pmatrix} | \\ \mathbf{u}_1 \\ | \end{pmatrix} \begin{pmatrix} - & \mathbf{v}_1^T & - \end{pmatrix} +$$

$$\cdots + \lambda_k \begin{pmatrix} | \\ \mathbf{u}_k \\ | \end{pmatrix} \begin{pmatrix} - & \mathbf{v}_k^T & - \end{pmatrix}$$

Spectrum of singular values



Rank k approximation to matrix

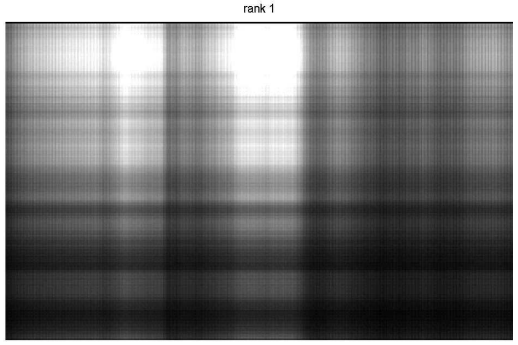


SVD on images

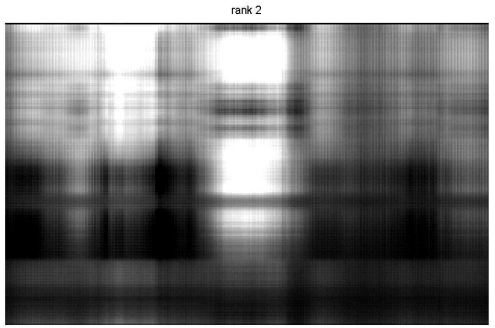
- Run demo

```
load clown
[U,S,V] = svd(X,0);
ranks = [1 2 5 10 20 rank(X)];
for k=ranks(:)'
  Xhat = (U(:,1:k)*S(1:k,1:k)*V(:,1:k)');
  image(Xhat);
end
```

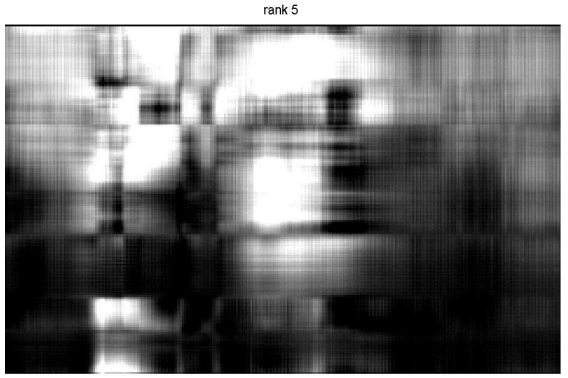
Clown example



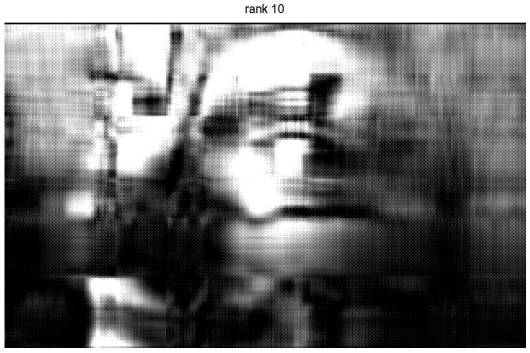
1



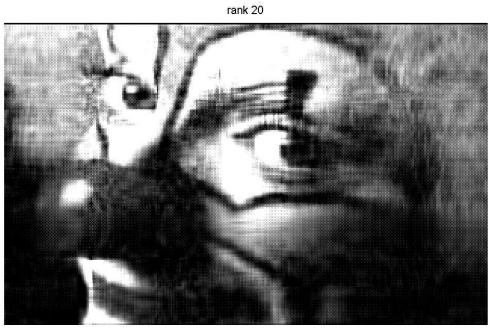
2



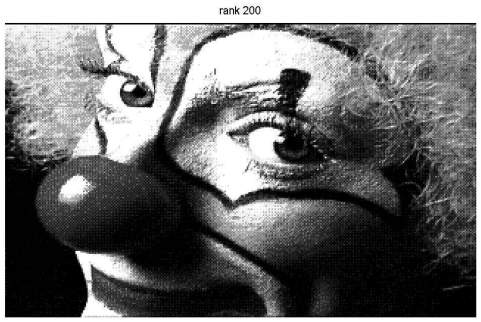
5



10



20

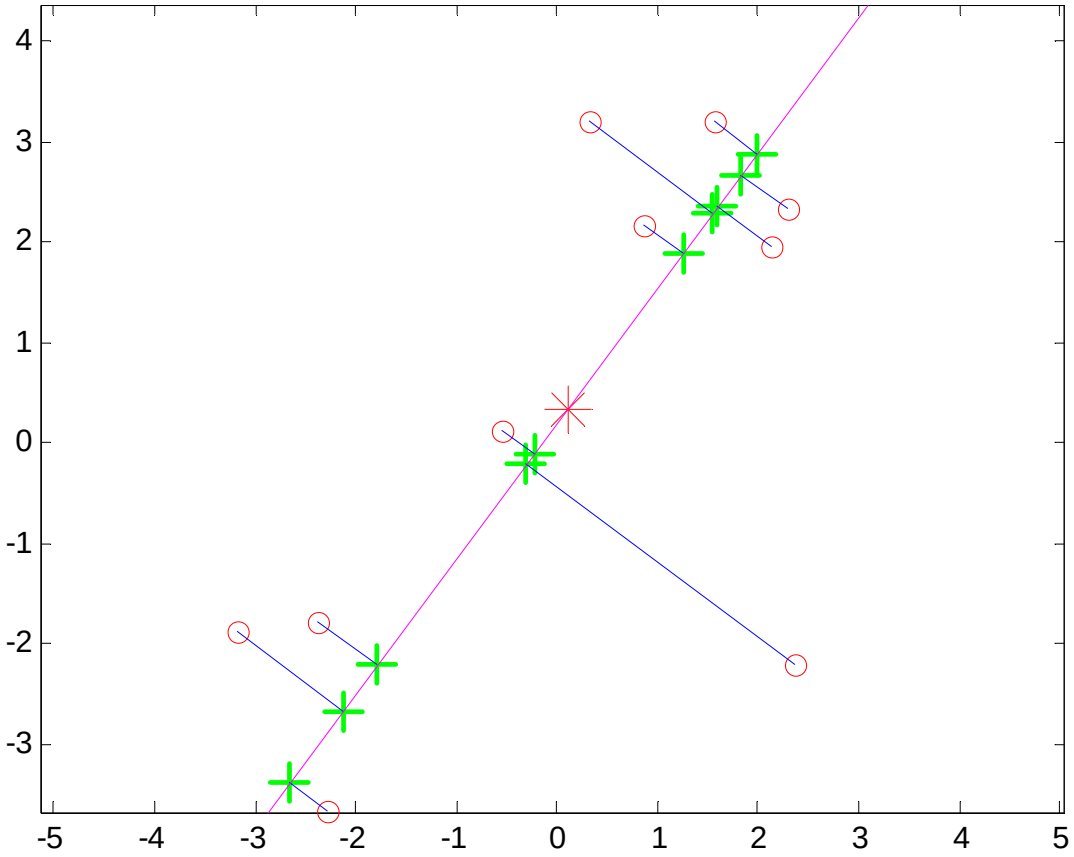


200

Space savings

$$
\begin{aligned}
\mathbf{A} &\approx \mathbf{U}_{:,1:k}\mathbf{\Sigma}_{1:k,1:k}\mathbf{V}^T_{1:,1:k} \\
m \times n &\approx (m \times k)\ (k)\ (n \times k) = (m + n + 1)k \\
200 \times 320 = 64,000 &\rightarrow (200 + 320 + 1)20 = 10,420
\end{aligned}
$$

Principal Components Analysis

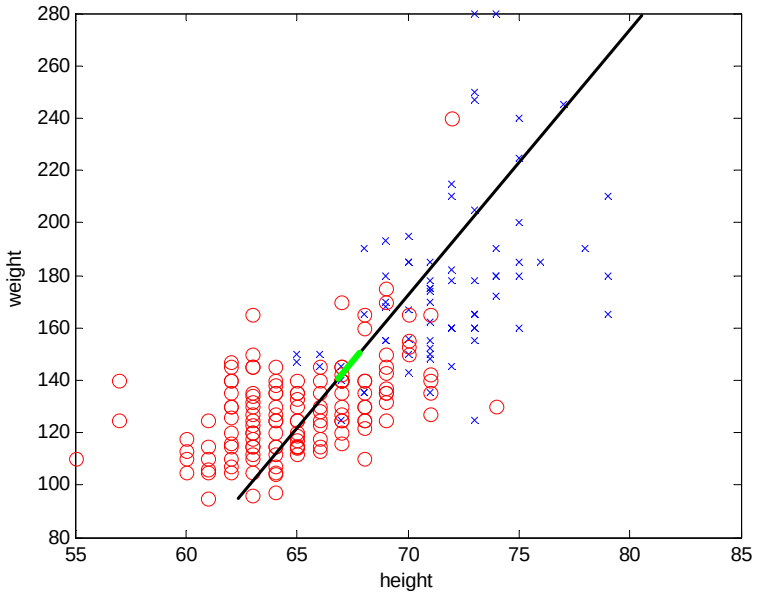


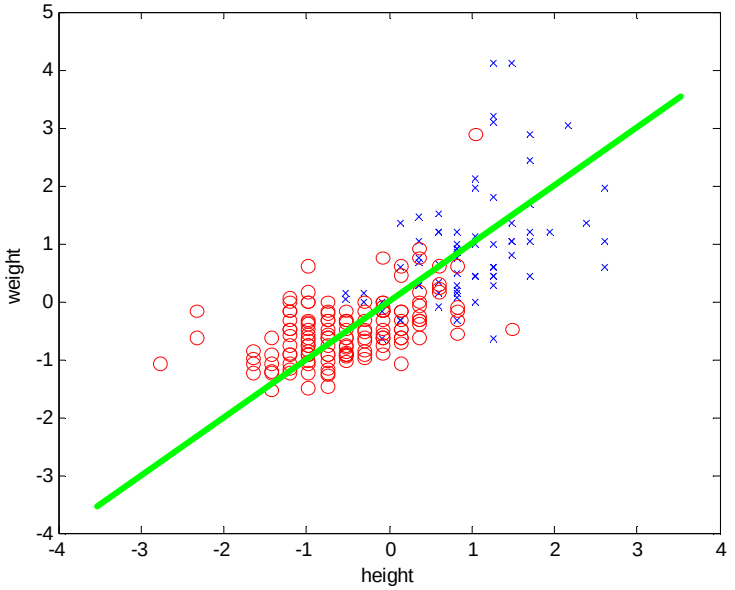
$$\hat{\mathbf{x}}_i = \sum_{j=1}^{k} z_{ij}\mathbf{v}_j .$$

scores

Loadings (basis)

PCA on height-weight data



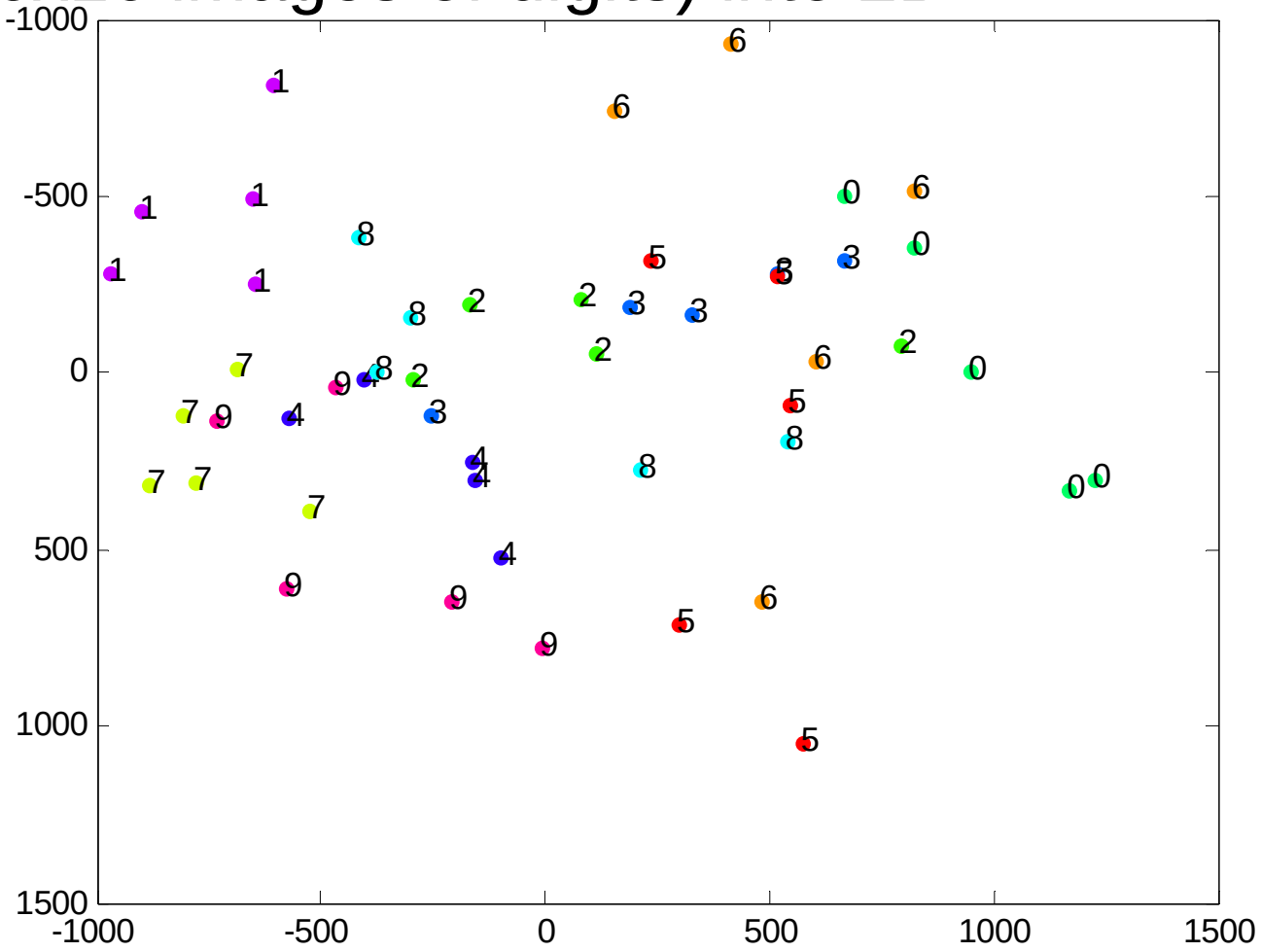


Applications

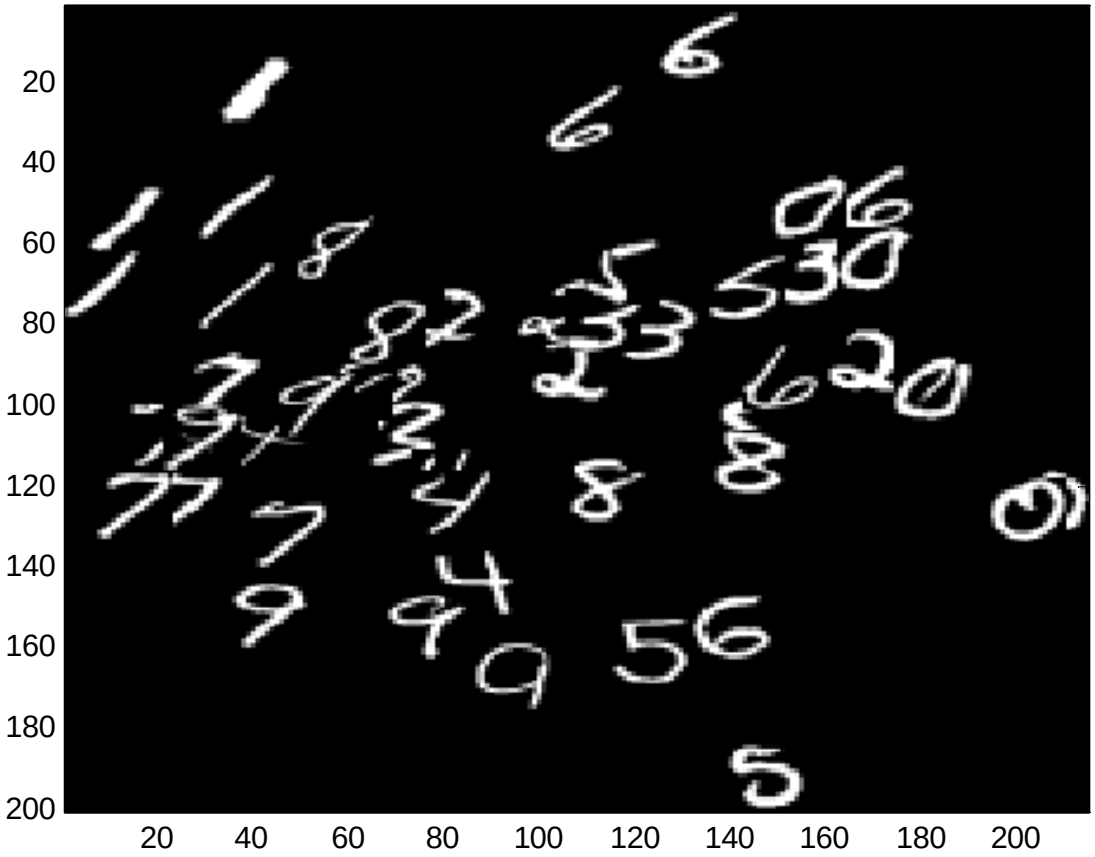
- Visualizing high dim data
- Reducing dim for nearest neighbor classifier

Visualizing data

- Project 256 dimensional vectors (representing 16x16 images of digits) into 2D



Embed vectors into their z1, z2 coords



Nearest neighbor classifier

- Look up k nearest neighbors in training set.
- Return majority vote of their labels.
- For k=1, we have

$$\hat{y} = y(\arg\min_i D(\mathbf{x}_i, \mathbf{x}^*))$$

D(x,x') = Euclidean distance

- Use PCA to embed in low dimensions, compute distances there

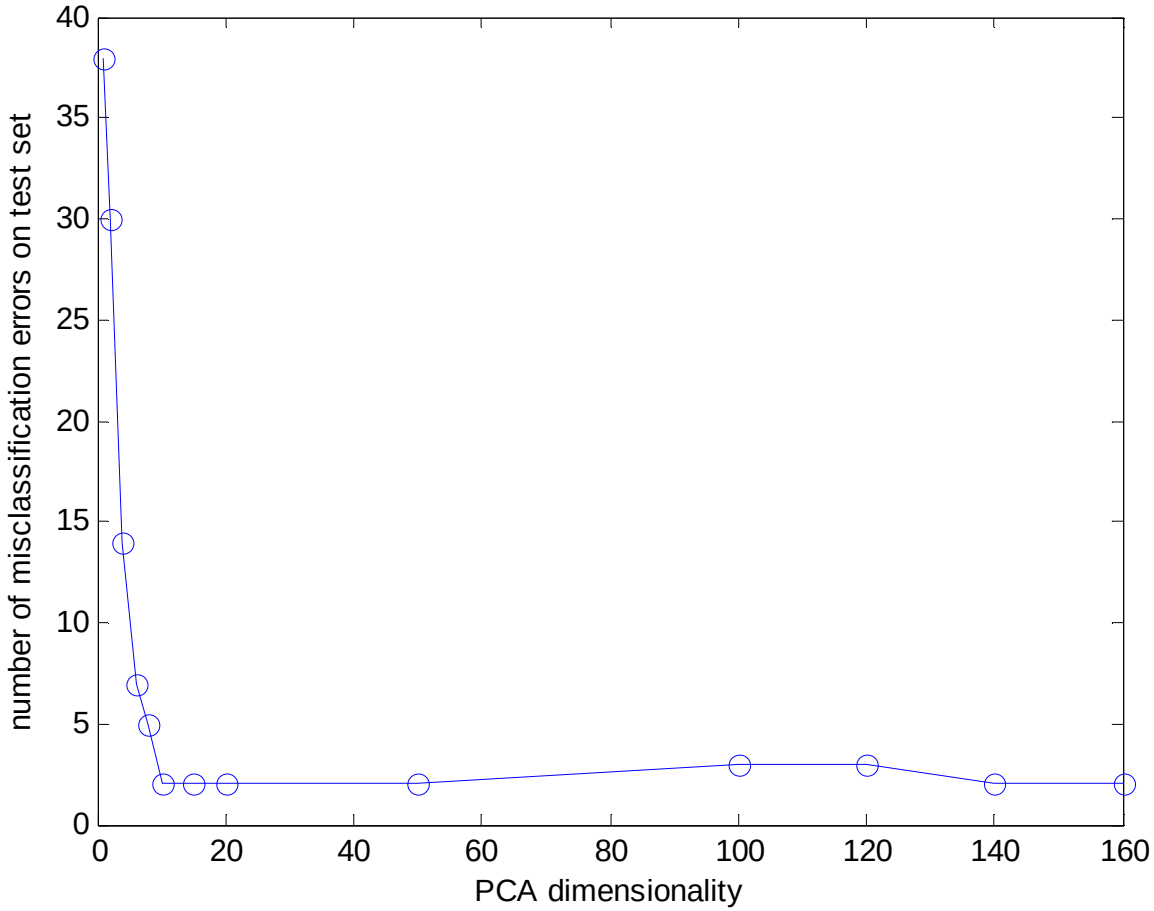
Eigenfaces

K=4

test images

closest match in training set using K=4

Misclassification rate vs K



Eigenfaces

K=10

test images

closest match in training set using K=10