

# Principal Component Analysis

---

# Motivation: dimensionality reduction

- It takes  $n \times d$  memory to store data  $\{x_i\}_{i=1}^n$  with  $x_i \in \mathbb{R}^d$
- But many real data have patterns that repeat over samples. Can we find some patterns and use them?



$d=32 \times 32$  pixels per image

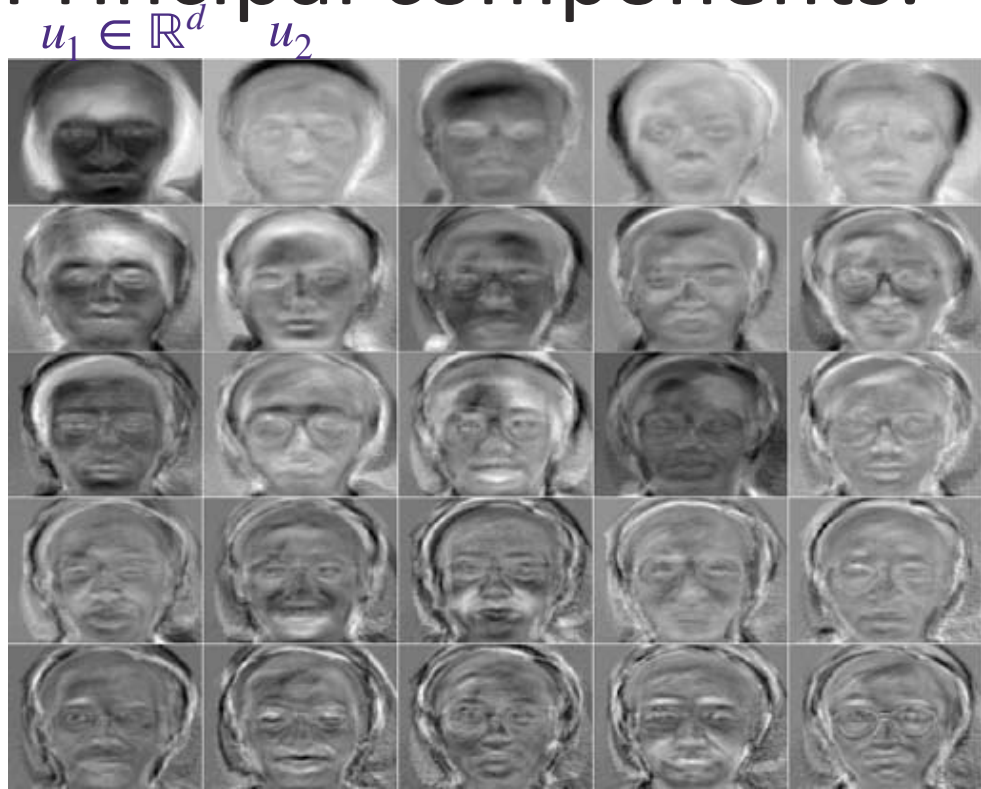
$n$  images

$d \times n$  real values to store the data

# Principal component analysis finds a compact linear representation

- patterns that capture the distinct features of the samples is called **principal component** (to be formally defined later)

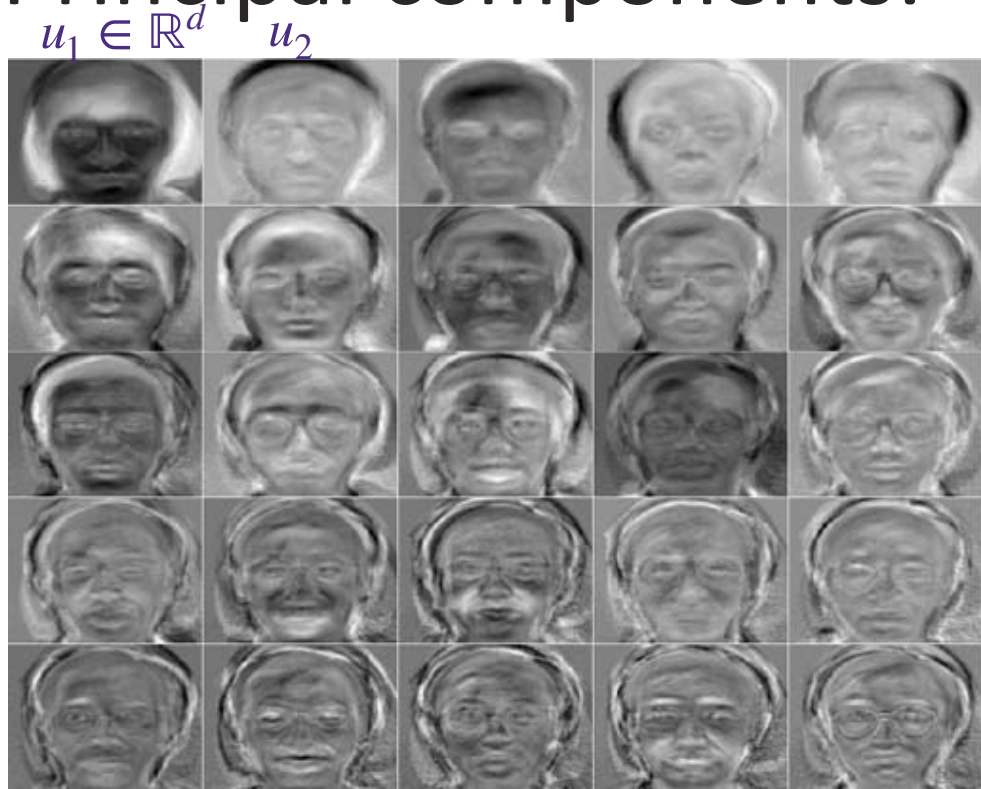
Principal components:



# Principal component analysis finds a compact linear representation

- patterns that capture the distinct features of the samples is called **principal component** (to be formally defined later)
- we can represent each sample as a **weighted linear combination** of, say,  $q=25$  principal components, and just store the weights

Principal components:

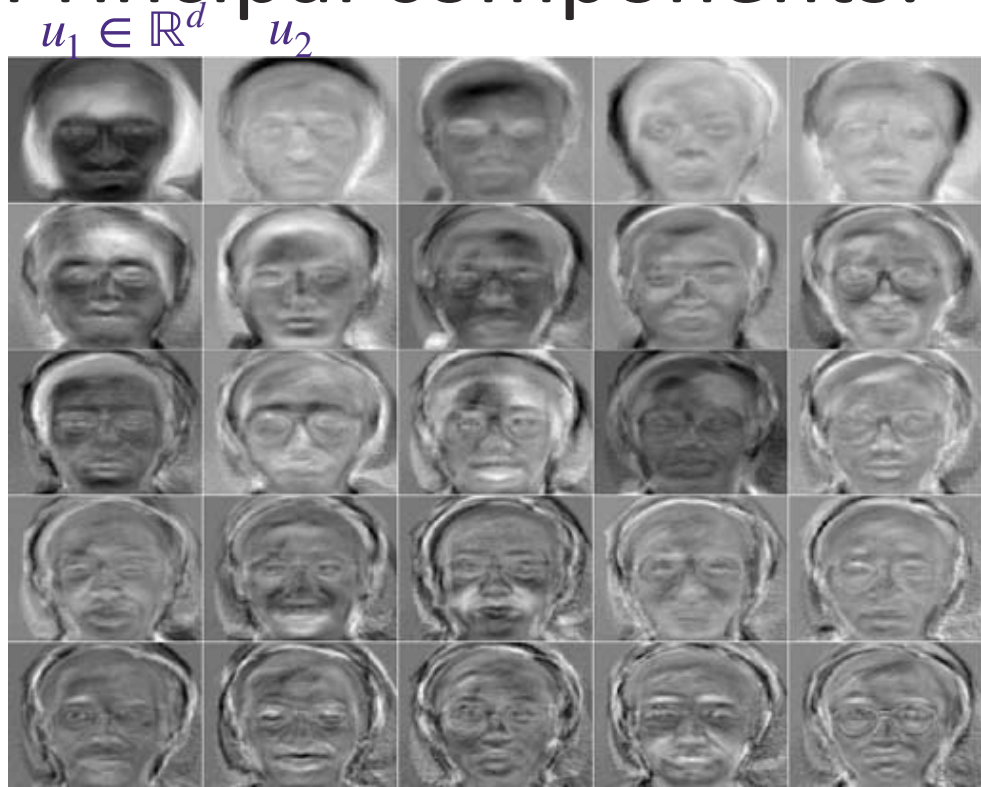


$$\approx a[1]u_1 + a[2]u_2 + \dots + a[25]u_{25}$$

# Principal component analysis finds a compact linear representation

- patterns that capture the distinct features of the samples is called **principal component** (to be formally defined later)
- we can represent each sample as a **weighted linear combination** of, say,  $q=25$  principal components, and just store the weights

Principal components:



$$\approx a[1]u_1 + a[2]u_2 + \dots + a[25]u_{25}$$

- With  $q=25$ , to store  $n$  images, it requires memory of only  $d \times q + q \times n \ll d \times n$

# 10 principal components give a pretty good reconstruction of a face

average face  $\bar{x} + a[1]u_1$   $\bar{x} + a[1]u_1 + a[2]u_2$

$\bar{x}$

$r=1$

$r=2$

$r=3$

$r=4$



$r=7$

$r=8$

$r=9$

$r=10$

↑  
Ground truths real face

# PCA: a high-fidelity linear projection

---

Given  $x_1, \dots, x_n \in \mathbb{R}^d$ , for  $q \ll d$  find a compressed representation with  $\lambda_1, \dots, \lambda_n \in \mathbb{R}^q$  such that  $x_i \approx \mu + \mathbf{V}_q \lambda_i$  and  $\mathbf{V}_q^T \mathbf{V}_q = I$

$$\min_{\mu, \mathbf{V}_q, \{\lambda_i\}_i} \sum_{i=1}^n \|x_i - \mu - \mathbf{V}_q \lambda_i\|_2^2$$

# PCA: a high-fidelity linear projection

---

Given  $x_1, \dots, x_n \in \mathbb{R}^d$ , for  $q \ll d$  find a compressed representation with  $\lambda_1, \dots, \lambda_n \in \mathbb{R}^q$  such that  $x_i \approx \mu + \mathbf{V}_q \lambda_i$  and  $\mathbf{V}_q^T \mathbf{V}_q = I$

$$\min_{\mu, \mathbf{V}_q, \{\lambda_i\}_i} \sum_{i=1}^n \|x_i - \mu - \mathbf{V}_q \lambda_i\|_2^2$$

Fix  $\mathbf{V}_q$  and solve for  $\mu, \lambda_i$ :

# PCA: a high-fidelity linear projection

Given  $x_1, \dots, x_n \in \mathbb{R}^d$ , for  $q \ll d$  find a compressed representation with  $\lambda_1, \dots, \lambda_n \in \mathbb{R}^q$  such that  $x_i \approx \mu + \mathbf{V}_q \lambda_i$  and  $\mathbf{V}_q^T \mathbf{V}_q = I$

$$\min_{\mu, \mathbf{V}_q, \{\lambda_i\}_i} \sum_{i=1}^n \|x_i - \mu - \mathbf{V}_q \lambda_i\|_2^2$$

Fix  $\mathbf{V}_q$  and solve for  $\mu, \lambda_i$ :

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

$$\lambda_i = \mathbf{V}_q^T (x_i - \bar{x})$$

$$\hat{x}_i := \bar{x} + \mathbf{V}_q \mathbf{V}_q^T (x_i - \bar{x}) = \bar{x} + \sum_{j=1}^q v_j \langle v_j, x_i - \bar{x} \rangle$$

# PCA: a high-fidelity linear projection

Given  $x_1, \dots, x_n \in \mathbb{R}^d$ , for  $q \ll d$  find a compressed representation with  $\lambda_1, \dots, \lambda_n \in \mathbb{R}^q$  such that  $x_i \approx \mu + \mathbf{V}_q \lambda_i$  and  $\mathbf{V}_q^T \mathbf{V}_q = I$

$$\min_{\mu, \mathbf{V}_q, \{\lambda_i\}_i} \sum_{i=1}^n \|x_i - \mu - \mathbf{V}_q \lambda_i\|_2^2$$

Fix  $\mathbf{V}_q$  and solve for  $\mu, \lambda_i$ :

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

$$\lambda_i = \mathbf{V}_q^T (x_i - \bar{x})$$

$$\hat{x}_i := \bar{x} + \mathbf{V}_q \mathbf{V}_q^T (x_i - \bar{x}) = \bar{x} + \sum_{j=1}^q v_j \langle v_j, x_i - \bar{x} \rangle$$

$$\min_{\mathbf{V}_q} \sum_{i=1}^N \|(x_i - \bar{x}) - \mathbf{V}_q \mathbf{V}_q^T (x_i - \bar{x})\|^2.$$

$\mathbf{V}_q \mathbf{V}_q^T$  is a *projection matrix* that minimizes error in basis of size  $q$

# PCA: a high-fidelity linear projection

$$\min_{\mathbf{V}_q} \sum_{i=1}^N \|(x_i - \bar{x}) - \mathbf{V}_q \mathbf{V}_q^T (x_i - \bar{x})\|^2.$$

$\mathbf{V}_q \mathbf{V}_q^T$  is a *projection matrix* that minimizes error in basis of size  $q$

$$\hat{x}_i := \bar{x} + \mathbf{V}_q \mathbf{V}_q^T (x_i - \bar{x}) = \bar{x} + \sum_{j=1}^q v_j \langle v_j, x_i - \bar{x} \rangle$$

# PCA: a high-fidelity linear projection

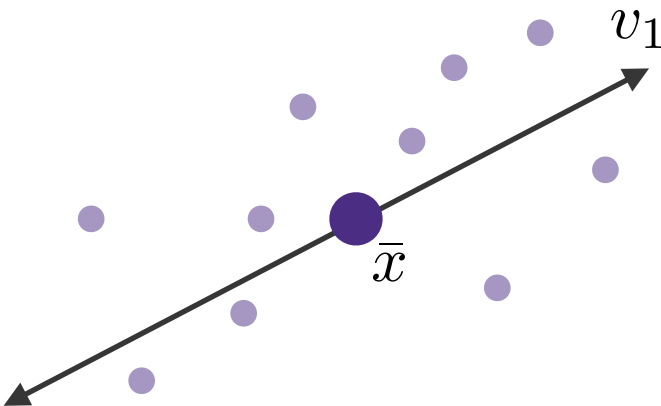
$$\min_{\mathbf{V}_q} \sum_{i=1}^N \|(x_i - \bar{x}) - \mathbf{V}_q \mathbf{V}_q^T (x_i - \bar{x})\|^2.$$

$\mathbf{V}_q \mathbf{V}_q^T$  is a *projection matrix* that minimizes error in basis of size  $q$

$$\hat{x}_i := \bar{x} + \mathbf{V}_q \mathbf{V}_q^T (x_i - \bar{x}) = \bar{x} + \sum_{j=1}^q v_j \langle v_j, x_i - \bar{x} \rangle$$

Case when  $q = 1$

$$v_1 = \arg \min_{v: \|v\|_2=1} \sum_{i=1}^N \|(x_i - \bar{x}) - vv^T (x_i - \bar{x})\|_2^2$$



# PCA: a high-fidelity linear projection

$$\min_{\mathbf{V}_q} \sum_{i=1}^N \|(x_i - \bar{x}) - \mathbf{V}_q \mathbf{V}_q^T (x_i - \bar{x})\|^2.$$

$\mathbf{V}_q \mathbf{V}_q^T$  is a *projection matrix* that minimizes error in basis of size  $q$

$$\hat{x}_i := \bar{x} + \mathbf{V}_q \mathbf{V}_q^T (x_i - \bar{x}) = \bar{x} + \sum_{j=1}^q v_j \langle v_j, x_i - \bar{x} \rangle$$

Case when  $q = 1$

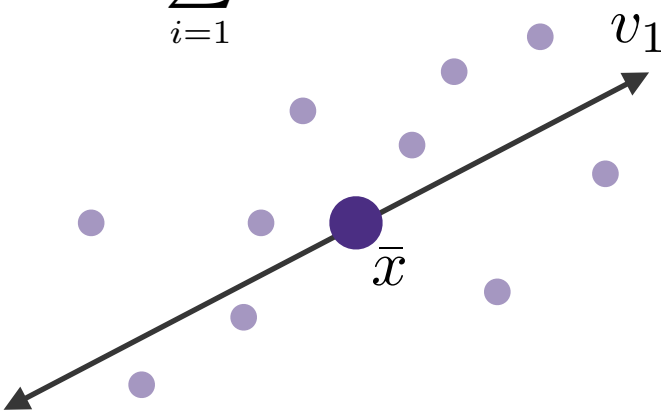
$$\begin{aligned} v_1 &= \arg \min_{v: \|v\|_2=1} \sum_{i=1}^N \|(x_i - \bar{x}) - vv^T (x_i - \bar{x})\|_2^2 \\ &= \arg \min_{v: \|v\|_2=1} \sum_{i=1}^N \|x_i - \bar{x}\|_2^2 - 2(x_i - \bar{x})^T vv^T (x_i - \bar{x}) \\ &\quad + (x_i - \bar{x})^T vv^T vv^T (x_i - \bar{x}) \end{aligned}$$

$$= \arg \min_{v: \|v\|_2=1} \sum_{i=1}^N \|x_i - \bar{x}\|_2^2 - \sum_{i=1}^N (x_i - \bar{x})^T vv^T (x_i - \bar{x})$$

$$= \arg \max_{v: \|v\|_2=1} \sum_{i=1}^N (x_i - \bar{x})^T vv^T (x_i - \bar{x})$$

$$= \arg \max_{v: \|v\|_2=1} v^T \Sigma v$$

$$\Sigma := \sum_{i=1}^N (x_i - \bar{x})(x_i - \bar{x})^T$$



# PCA: a high-fidelity linear projection

$$\min_{\mathbf{V}_q} \sum_{i=1}^N \|(x_i - \bar{x}) - \mathbf{V}_q \mathbf{V}_q^T (x_i - \bar{x})\|^2.$$

$\mathbf{V}_q \mathbf{V}_q^T$  is a *projection matrix* that minimizes error in basis of size  $q$

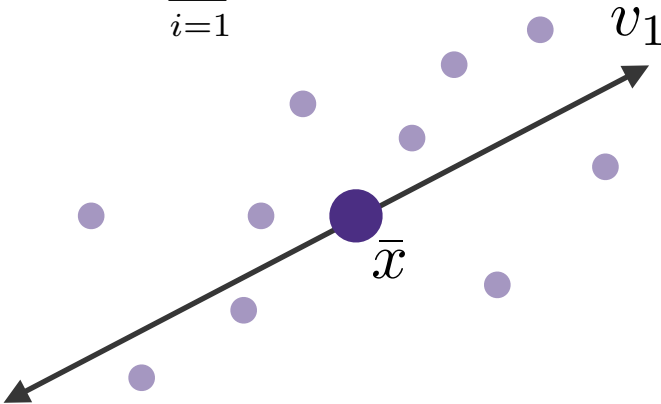
$$\hat{x}_i := \bar{x} + \mathbf{V}_q \mathbf{V}_q^T (x_i - \bar{x}) = \bar{x} + \sum_{j=1}^q v_j \langle v_j, x_i - \bar{x} \rangle$$

General  $q \geq 1$   $\min_{\mathbf{V}_q} \sum_{i=1}^N \|(x_i - \bar{x}) - \mathbf{V}_q \mathbf{V}_q^T (x_i - \bar{x})\|_2^2 = \min_{\mathbf{V}_q} \text{Tr}(\Sigma) - \text{Tr}(\mathbf{V}_q^T \Sigma \mathbf{V}_q)$

$$\Sigma := \sum_{i=1}^N (x_i - \bar{x})(x_i - \bar{x})^T$$

$\mathbf{V}_q$  are the first  $q$  eigenvectors of  $\Sigma$

Minimize reconstruction error and capture the most variance in your data.



# PCA: a high-fidelity linear projection

Given  $x_i \in \mathbb{R}^d$  and some  $q < d$  consider

$$\min_{\mathbf{V}_q} \sum_{i=1}^N \|(x_i - \bar{x}) - \mathbf{V}_q \mathbf{V}_q^T (x_i - \bar{x})\|^2.$$

where  $\mathbf{V}_q = [v_1, v_2, \dots, v_q]$  is orthonormal:

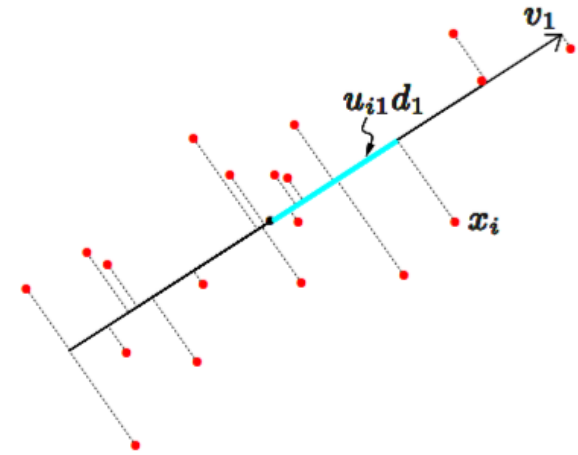
$$\mathbf{V}_q^T \mathbf{V}_q = I_q$$

$\mathbf{V}_q$  are the first  $q$  eigenvectors of  $\Sigma$

$\mathbf{V}_q$  are the first  $q$  principal components

Principal Component Analysis (PCA) projects  $(\mathbf{X} - \mathbf{1}\bar{x}^T)$  down onto  $\mathbf{V}_q$

$$(\mathbf{X} - \mathbf{1}\bar{x}^T) \mathbf{V}_q = \mathbf{U}_q \text{diag}(d_1, \dots, d_q) \quad \mathbf{U}_q^T \mathbf{U}_q = I_q$$



$$\Sigma := \sum_{i=1}^N (x_i - \bar{x})(x_i - \bar{x})^T$$

# Singular Value Decomposition (SVD)

---

**Theorem (SVD):** Let  $\mathbf{A} \in \mathbb{R}^{m \times n}$  with rank  $r \leq \min\{m, n\}$ . Then  $\mathbf{A} = \mathbf{U}\mathbf{S}\mathbf{V}^T$  where  $\mathbf{S} \in \mathbb{R}^{r \times r}$  is diagonal with positive entries,  $\mathbf{U}^T\mathbf{U} = \mathbf{I}$ ,  $\mathbf{V}^T\mathbf{V} = \mathbf{I}$ .

$$\mathbf{A}^T \mathbf{A} v_i =$$

$$\mathbf{A}\mathbf{A}^T u_i =$$

# Singular Value Decomposition (SVD)

---

**Theorem (SVD):** Let  $\mathbf{A} \in \mathbb{R}^{m \times n}$  with rank  $r \leq \min\{m, n\}$ . Then  $\mathbf{A} = \mathbf{U}\mathbf{S}\mathbf{V}^T$  where  $\mathbf{S} \in \mathbb{R}^{r \times r}$  is diagonal with positive entries,  $\mathbf{U}^T\mathbf{U} = \mathbf{I}$ ,  $\mathbf{V}^T\mathbf{V} = \mathbf{I}$ .

$$\mathbf{A}^T \mathbf{A} v_i = \mathbf{S}_{i,i}^2 v_i$$

$$\mathbf{A} \mathbf{A}^T u_i = \mathbf{S}_{i,i}^2 u_i$$

$\mathbf{V}$  are the first  $r$  eigenvectors of  $\mathbf{A}^T \mathbf{A}$  with eigenvalues  $\text{diag}(\mathbf{S})$

$\mathbf{U}$  are the first  $r$  eigenvectors of  $\mathbf{A} \mathbf{A}^T$  with eigenvalues  $\text{diag}(\mathbf{S})$

# Linear projections

Given  $x_i \in \mathbb{R}^d$  and some  $q < d$  consider

$$\min_{\mathbf{V}_q} \sum_{i=1}^N \|(x_i - \bar{x}) - \mathbf{V}_q \mathbf{V}_q^T (x_i - \bar{x})\|^2.$$

where  $\mathbf{V}_q = [v_1, v_2, \dots, v_q]$  is orthonormal:

$$\mathbf{V}_q^T \mathbf{V}_q = I_q$$

$\mathbf{V}_q$  are the first  $q$  eigenvectors of  $\Sigma$

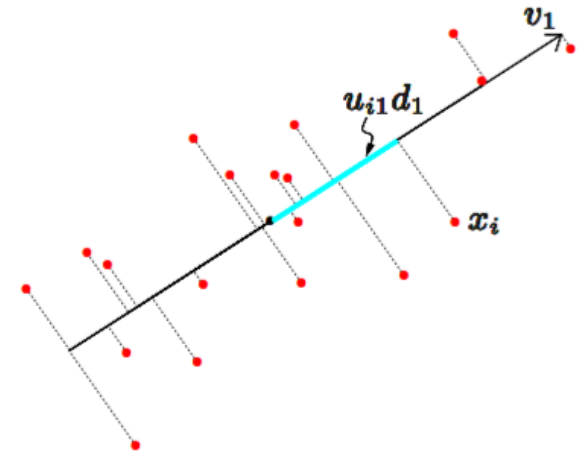
$\mathbf{V}_q$  are the first  $q$  principal components

Principal Component Analysis (PCA) projects  $(\mathbf{X} - \mathbf{1}\bar{x}^T)$  down onto  $\mathbf{V}_q$

$$(\mathbf{X} - \mathbf{1}\bar{x}^T) \mathbf{V}_q = \mathbf{U}_q \text{diag}(d_1, \dots, d_q) \quad \mathbf{U}_q^T \mathbf{U}_q = I_q$$

Singular Value Decomposition defined as

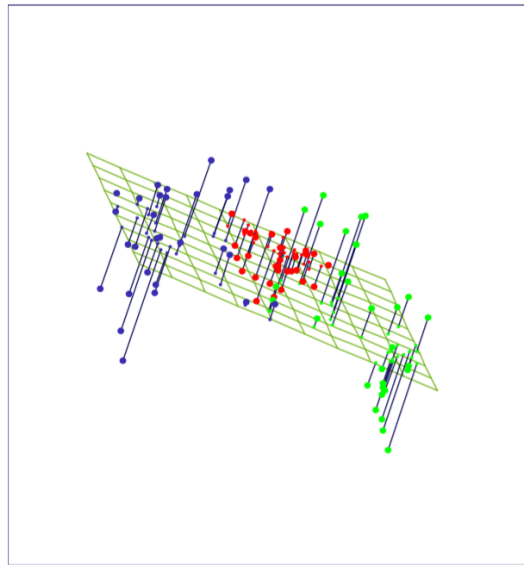
$$\mathbf{X} - \mathbf{1}\bar{x}^T = \mathbf{U} \mathbf{S} \mathbf{V}^T$$



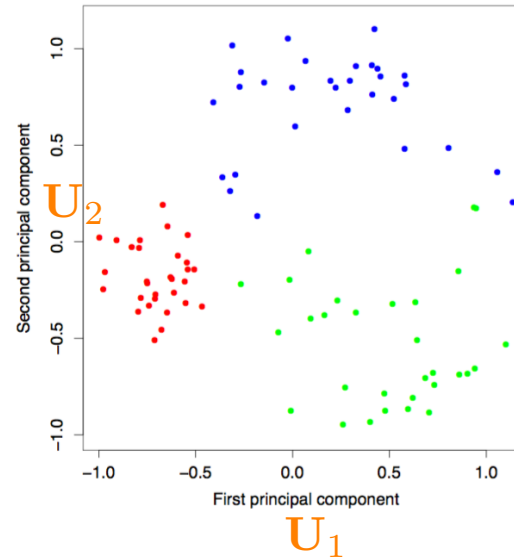
$$\Sigma := \sum_{i=1}^N (x_i - \bar{x})(x_i - \bar{x})^T$$

# Dimensionality reduction

$\mathbf{V}_q$  are the first  $q$  eigenvectors of  $\Sigma$  and SVD  $\mathbf{X} - \mathbf{1}\bar{x}^T = \mathbf{U}\mathbf{S}\mathbf{V}^T$



$$\mathbf{X} - \mathbf{1}\bar{x}^T$$



$$\mathbf{U}_1$$

# Dimensionality reduction

$\mathbf{V}_q$  are the first  $q$  eigenvectors of  $\Sigma$  and SVD  $\mathbf{X} - \mathbf{1}\bar{x}^T = \mathbf{U}\mathbf{S}\mathbf{V}^T$

Handwritten 3's, 16x16 pixel image so that  $x_i \in \mathbb{R}^{256}$

$$\begin{aligned} \hat{f}(\lambda) &= \bar{x} + \lambda_1 v_1 + \lambda_2 v_2 \\ &= \mathbf{3} + \lambda_1 \cdot \mathbf{3} + \lambda_2 \cdot \mathbf{3}. \end{aligned}$$

$$(\mathbf{X} - \mathbf{1}\bar{x}^T)\mathbf{V}_2 = \mathbf{U}_2\mathbf{S}_2 \in \mathbb{R}^{n \times 2}$$

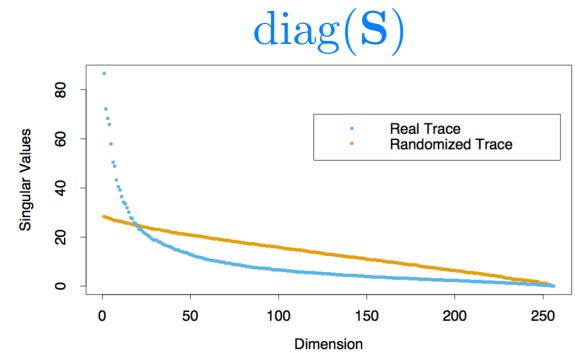
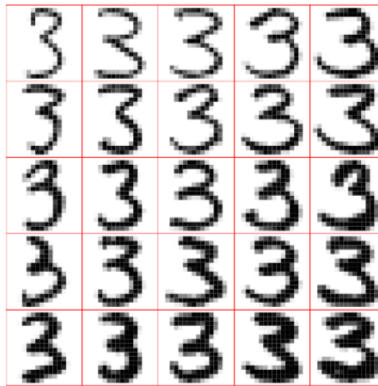
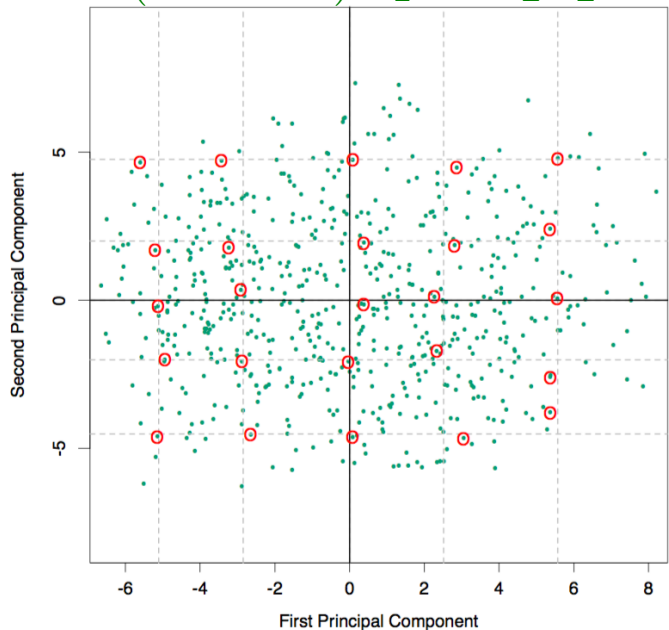
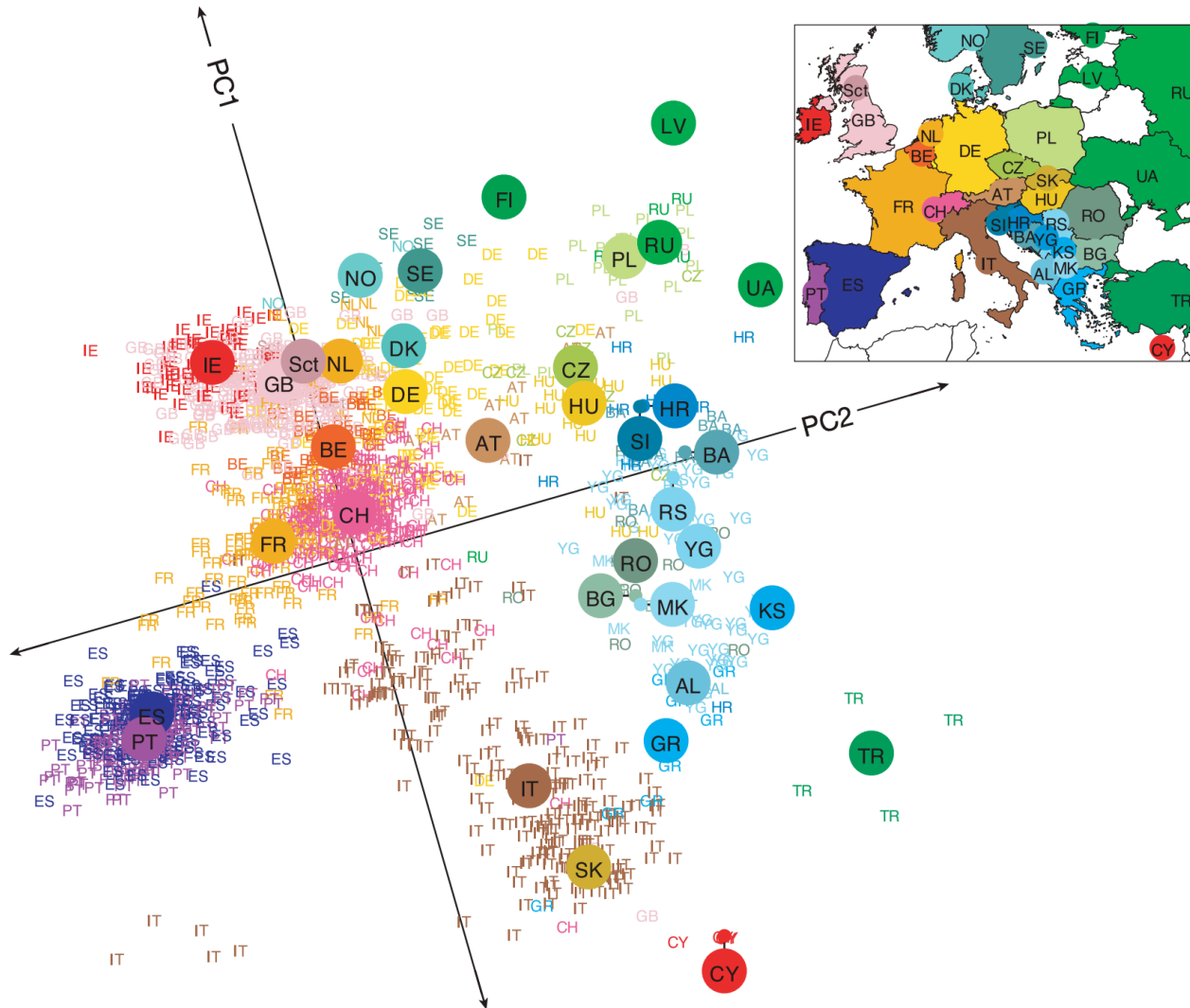


FIGURE 14.24. The 256 singular values for the digitized threes, compared to those for a randomized version of the data (each column of  $\mathbf{X}$  was scrambled).

# Dimensionality reduction



Novembre, et al, "Genes mirror geography within Europe" Nature 2008.

# Kernel PCA

---

$\mathbf{V}_q$  are the first  $q$  eigenvectors of  $\Sigma$  and SVD  $\mathbf{X} - \mathbf{1}\bar{x}^T = \mathbf{U}\mathbf{S}\mathbf{V}^T$

$$(\mathbf{X} - \mathbf{1}\bar{x}^T)\mathbf{V}_q = \mathbf{U}_q\mathbf{S}_q \in \mathbb{R}^{n \times q}$$

$$\mathbf{J}\mathbf{X} = \mathbf{X} - \mathbf{1}\bar{x}^T = \mathbf{U}\mathbf{S}\mathbf{V}^T \quad \mathbf{J} = \mathbf{I} - \mathbf{1}\mathbf{1}^T/n$$

$$(\mathbf{J}\mathbf{X})(\mathbf{J}\mathbf{X})^T =$$

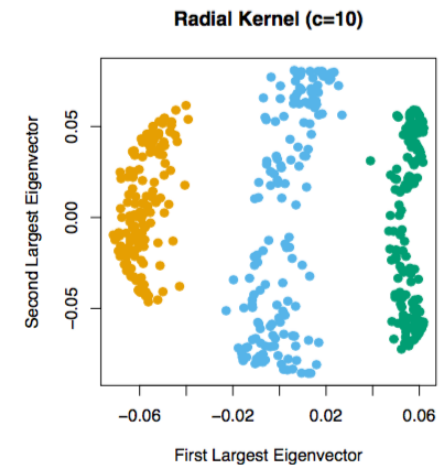
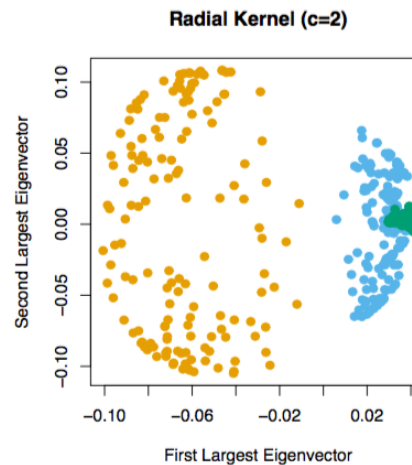
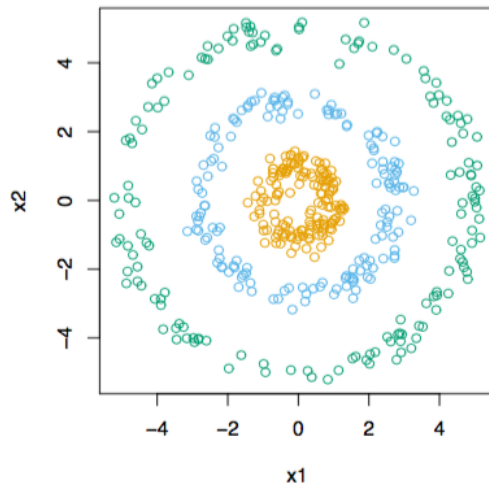
# Kernel PCA

$\mathbf{V}_q$  are the first  $q$  eigenvectors of  $\Sigma$  and SVD  $\mathbf{X} - \mathbf{1}\bar{x}^T = \mathbf{U}\mathbf{S}\mathbf{V}^T$

$$(\mathbf{X} - \mathbf{1}\bar{x}^T)\mathbf{V}_q = \mathbf{U}_q\mathbf{S}_q \in \mathbb{R}^{n \times q}$$

$$\mathbf{J}\mathbf{X} = \mathbf{X} - \mathbf{1}\bar{x}^T = \mathbf{U}\mathbf{S}\mathbf{V}^T \quad \mathbf{J} = \mathbf{I} - \mathbf{1}\mathbf{1}^T/n$$

$$(\mathbf{J}\mathbf{X})(\mathbf{J}\mathbf{X})^T = \mathbf{U}\mathbf{S}^2\mathbf{U}^T$$



# Matrix completion

Given historical data on how users rated movies in past:



17,700 movies, 480,189 users, 99,072,112 ratings

(Sparsity: 1.2%)

Predict how the same users will rate movies in the future (for \$1 million prize)

						...
Alice	1	?	?	4	?	
Bob	?	2	5	?	?	
Carol	?	?	4	5	?	
Dave	5	?	?	?	4	
⋮						

# Matrix completion

---

n movies, m users,  $|S|$  ratings

$$\arg \min_{U \in \mathbb{R}^{m \times d}, V \in \mathbb{R}^{n \times d}} \sum_{(i,j,s) \in \mathcal{S}} \|(UV^T)_{i,j} - s_{i,j}\|_2^2$$

How do we solve it? With full information?

# Matrix completion

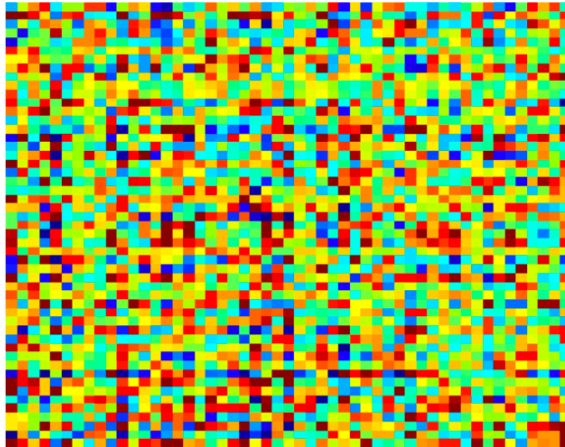
---

n movies, m users,  $|S|$  ratings

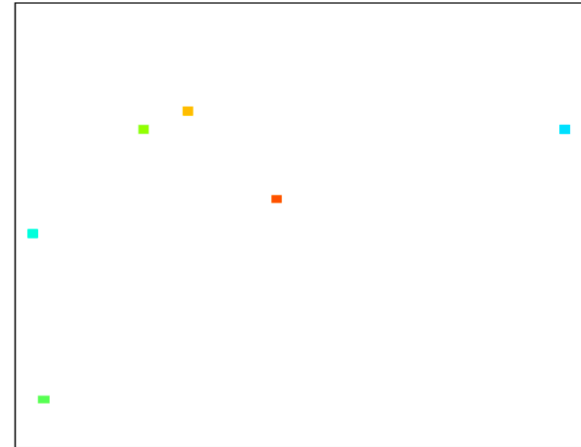
$$\arg \min_{U \in \mathbb{R}^{m \times d}, V \in \mathbb{R}^{n \times d}} \sum_{(i,j,s) \in S} \|(UV^T)_{i,j} - s_{i,j}\|_2^2$$

# Example: $2000 \times 2000$ rank-8 random matrix

low-rank matrix  $\mathbf{X}$

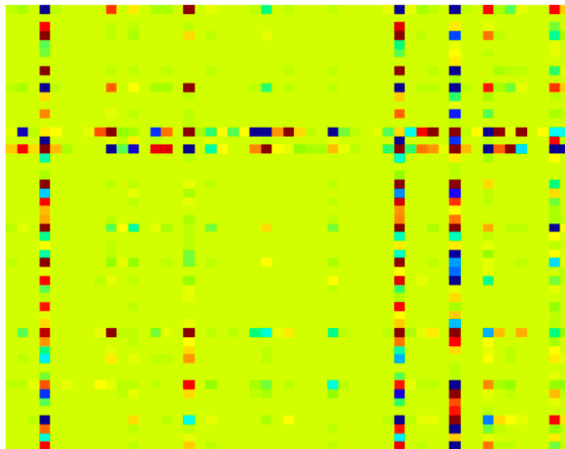


sampled matrix

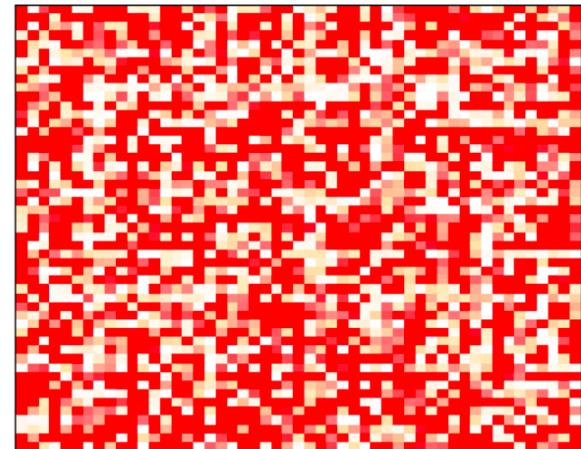


For illustration,  
we zoom in to a  
50x50 submatrix

Gradient descent output  $\mathbf{UA}$



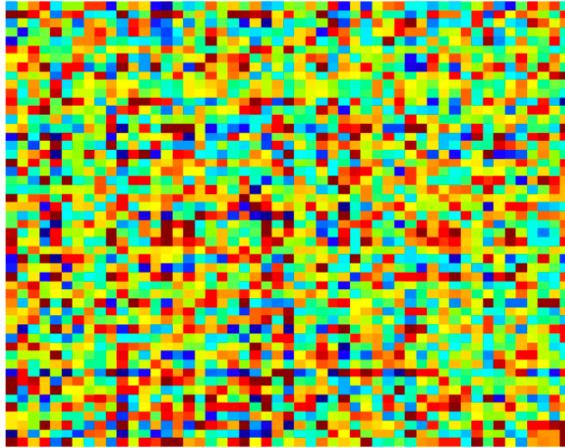
squared error  $(\mathbf{X}_{ji} - (\mathbf{UA})_{ji})^2$



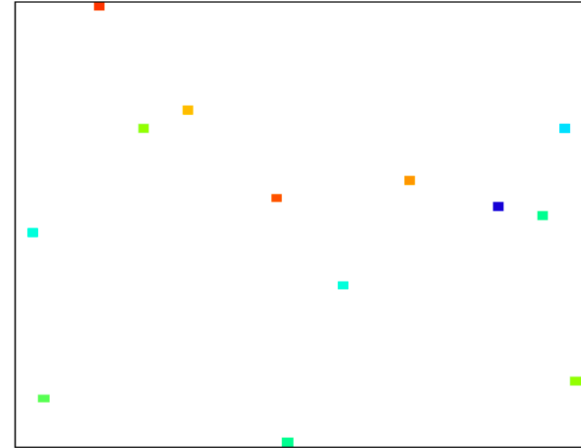
0.25% sampled

# Example: $2000 \times 2000$ rank-8 random matrix

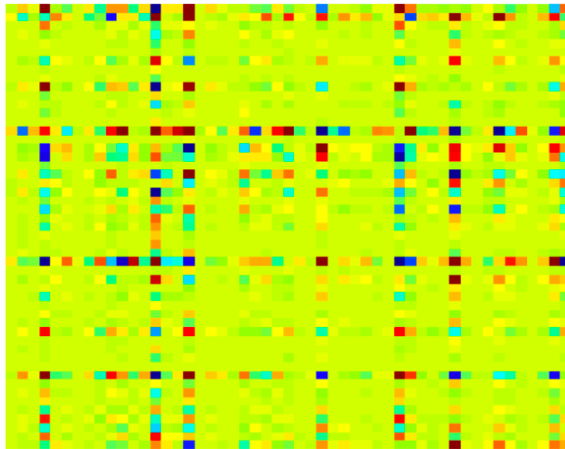
low-rank matrix  $\mathbf{X}$



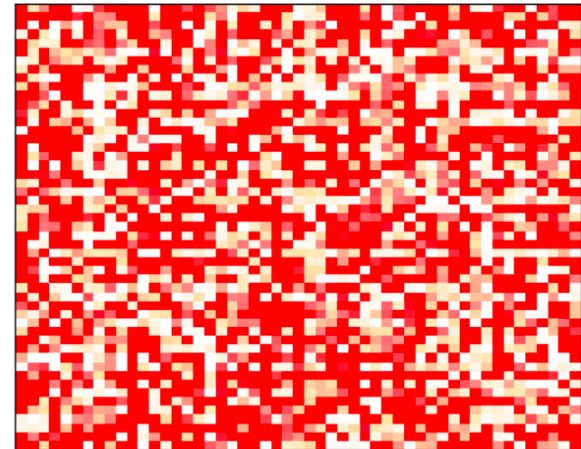
sampled matrix



Gradient descent output  $\mathbf{UA}$



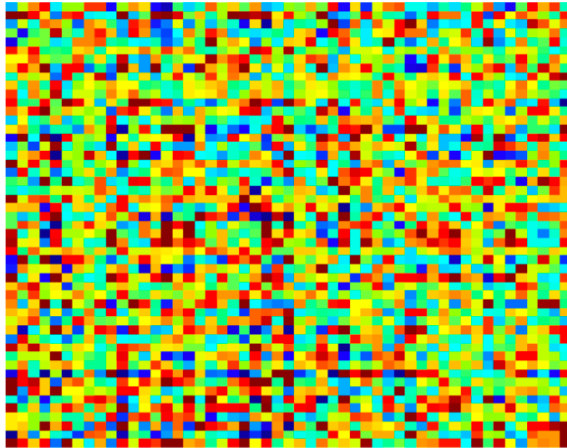
squared error  $(\mathbf{X}_{ji} - (\mathbf{UA})_{ji})^2$



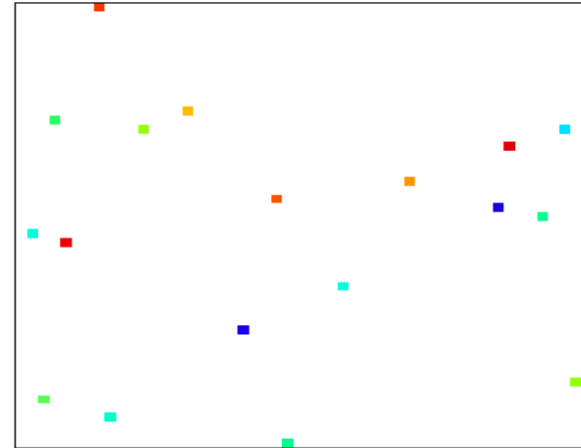
0.50% sampled

# Example: $2000 \times 2000$ rank-8 random matrix

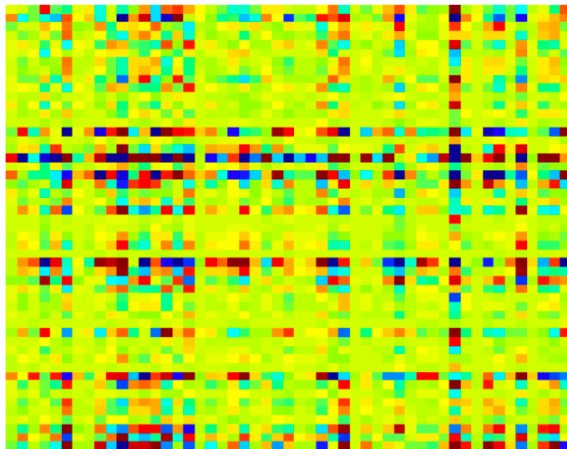
low-rank matrix  $\mathbf{X}$



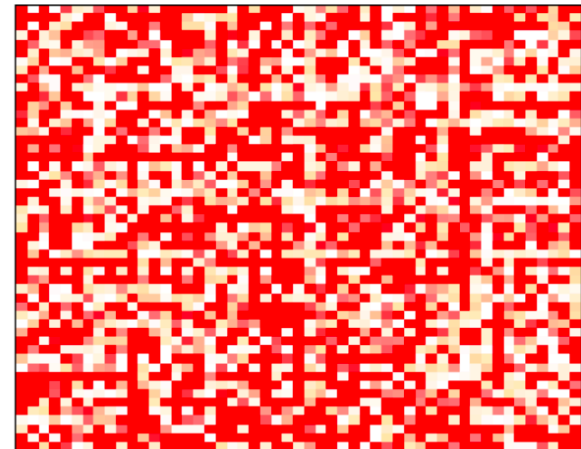
sampled matrix



Gradient descent output  $\mathbf{UA}$



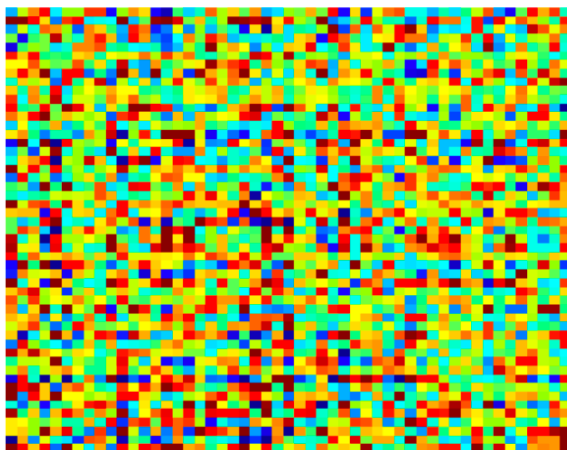
squared error  $(\mathbf{X}_{ji} - (\mathbf{UA})_{ji})^2$



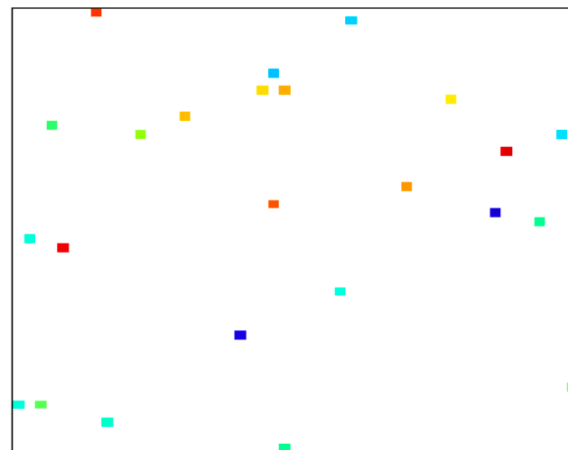
0.75% sampled

# Example: $2000 \times 2000$ rank-8 random matrix

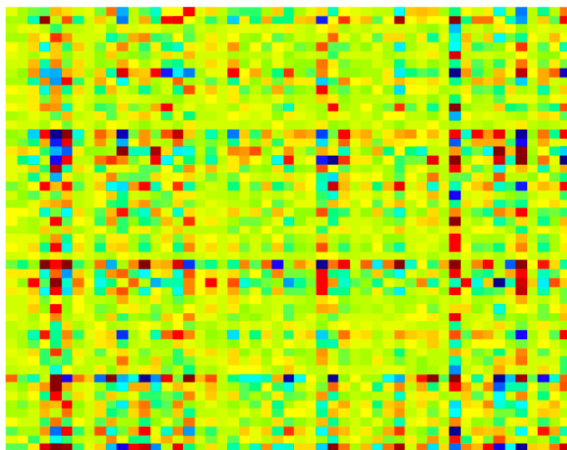
low-rank matrix  $\mathbf{X}$



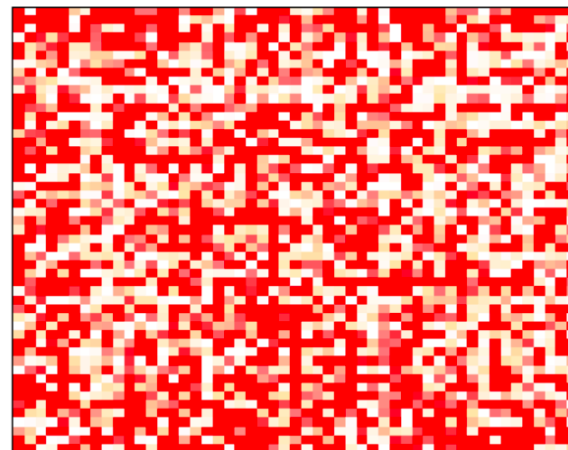
sampled matrix



Gradient descent output  $\mathbf{UA}$



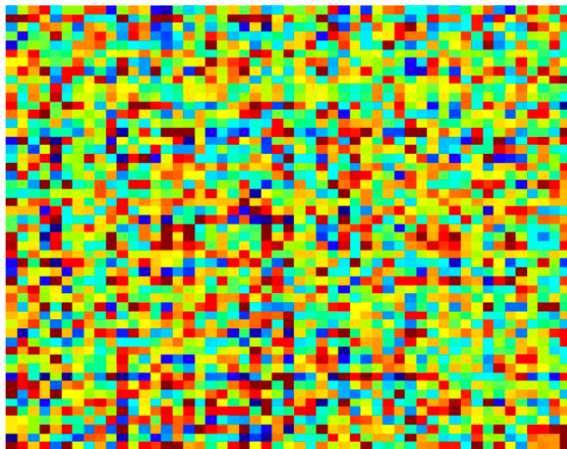
squared error  $(\mathbf{X}_{ji} - (\mathbf{UA})_{ji})^2$



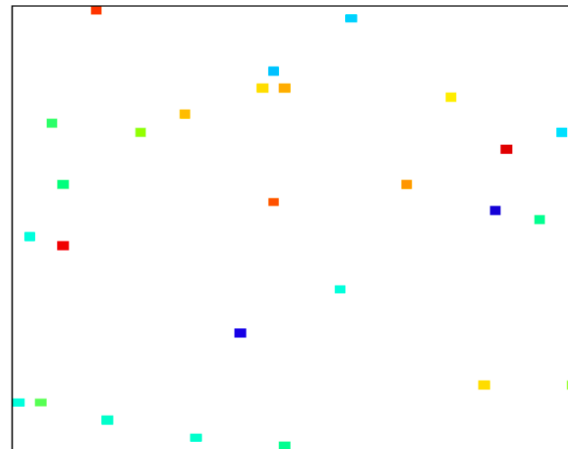
1.00% sampled

# Example: $2000 \times 2000$ rank-8 random matrix

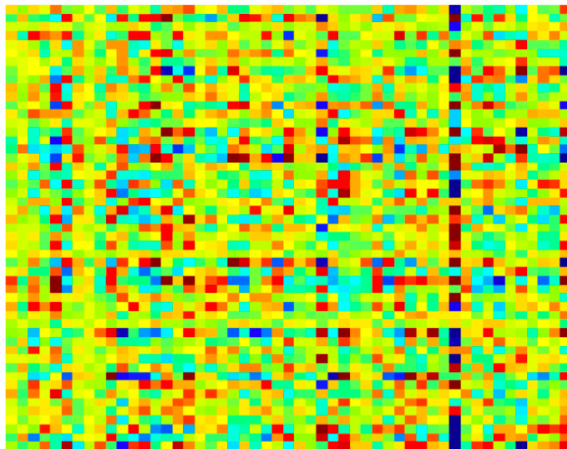
low-rank matrix  $\mathbf{X}$



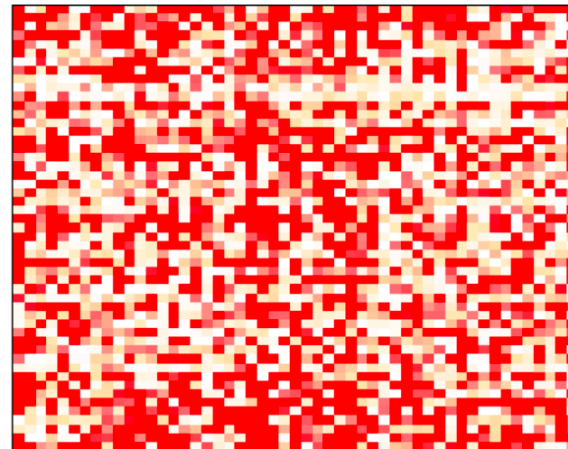
sampled matrix



Gradient descent output  $\mathbf{UA}$



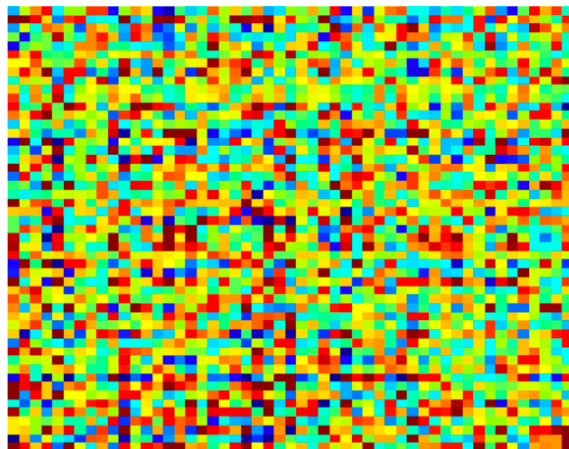
squared error  $(\mathbf{X}_{ji} - (\mathbf{UA})_{ji})^2$



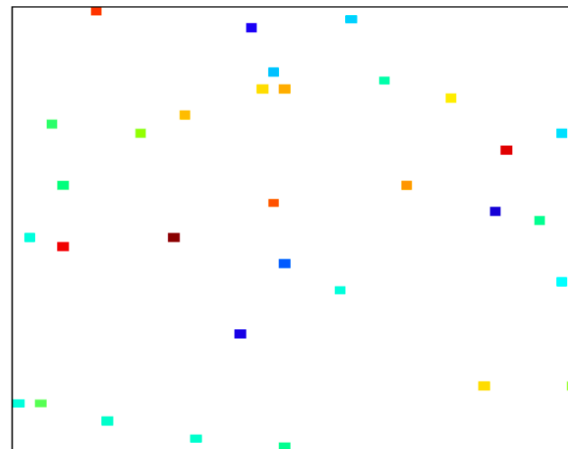
1.25% sampled

# Example: $2000 \times 2000$ rank-8 random matrix

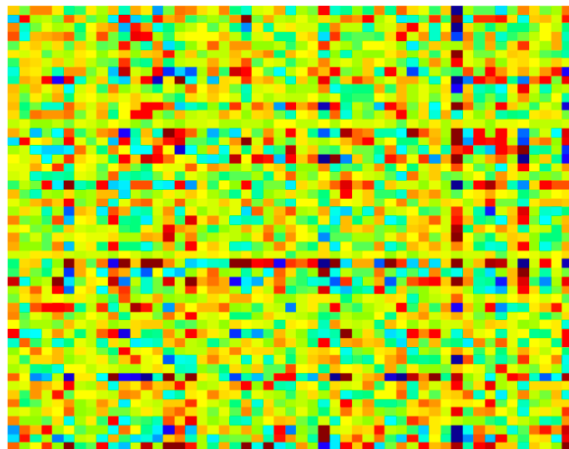
low-rank matrix  $\mathbf{X}$



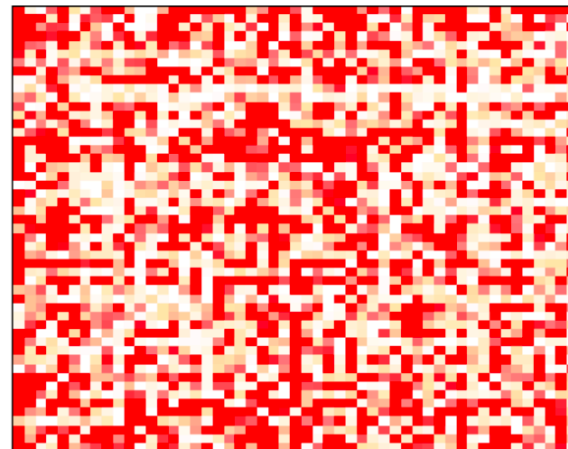
sampled matrix



Gradient descent output  $\mathbf{UA}$



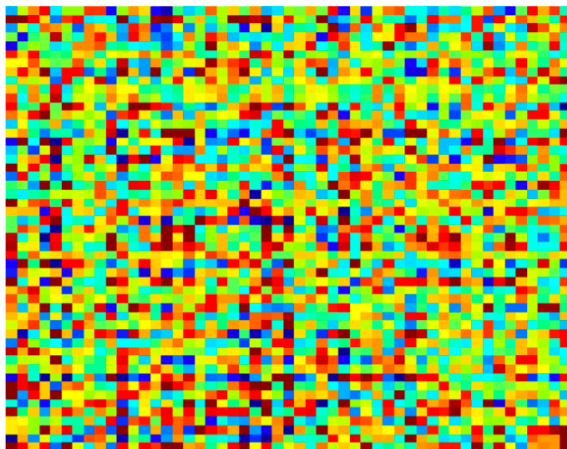
squared error  $(\mathbf{X}_{ji} - (\mathbf{UA})_{ji})^2$



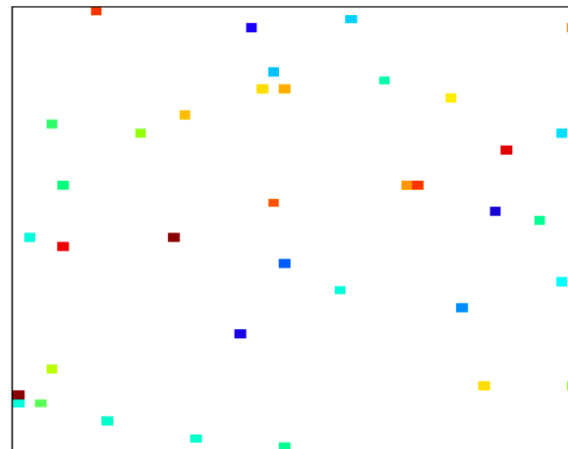
1.50% sampled

# Example: $2000 \times 2000$ rank-8 random matrix

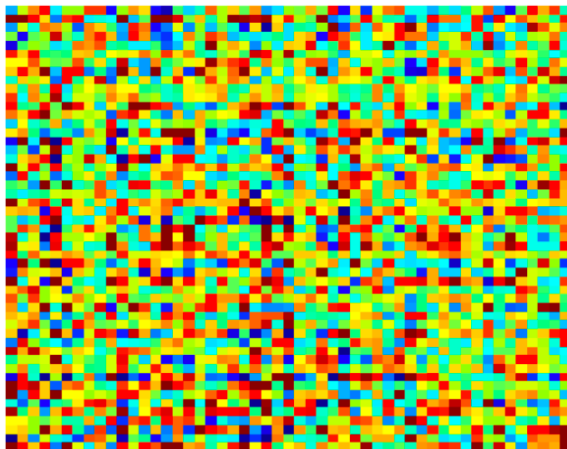
low-rank matrix  $\mathbf{X}$



sampled matrix



Gradient descent output  $\mathbf{UA}$



squared error  $(\mathbf{X}_{ji} - (\mathbf{UA})_{ji})^2$



1.75% sampled

# Random projections

PCA finds a low-dimensional representation that reduces population variance

$$\min_{\mathbf{V}_q} \sum_{i=1}^N \|(x_i - \bar{x}) - \mathbf{V}_q \mathbf{V}_q^T (x_i - \bar{x})\|^2.$$

$\mathbf{V}_q \mathbf{V}_q^T$  is a *projection matrix* that minimizes error in basis of size  $q$

$\mathbf{V}_q$  are the first  $q$  eigenvectors of  $\Sigma$

$$\Sigma := \sum_{i=1}^N (x_i - \bar{x})(x_i - \bar{x})^T$$

But what if I care about the reconstruction of the *individual* points?

$$\min_{\mathbf{W}_q} \max_{i=1, \dots, n} \|(x_i - \bar{x}) - \mathbf{W}_q \mathbf{W}_q^T (x_i - \bar{x})\|^2$$

# Random projections

$$\min_{\mathbf{W}_q} \max_{i=1, \dots, n} \|(x_i - \bar{x}) - \mathbf{W}_q \mathbf{W}_q^T (x_i - \bar{x})\|^2$$

## Johnson-Lindenstrauss (1983)

**Theorem 1.1.** (Johnson-Lindenstrauss) Let  $\epsilon \in (0, 1/2)$ . Let  $Q \subset \mathbb{R}^d$  be a set of  $n$  points and  $k = \frac{20 \log n}{\epsilon^2}$ . There exists a Lipschitz mapping  $f : \mathbb{R}^d \rightarrow \mathbb{R}^k$  such that for all  $u, v \in Q$ :

(independent of  $d$ )

$$(1 - \epsilon)\|u - v\|^2 \leq \|f(u) - f(v)\|^2 \leq (1 + \epsilon)\|u - v\|^2$$

# Random projections

$$\min_{\mathbf{W}_q} \max_{i=1, \dots, n} \|(x_i - \bar{x}) - \mathbf{W}_q \mathbf{W}_q^T (x_i - \bar{x})\|^2$$

## Johnson-Lindenstrauss (1983)

**Theorem 1.1.** (Johnson-Lindenstrauss) Let  $\epsilon \in (0, 1/2)$ . Let  $Q \subset \mathbb{R}^d$  be a set of  $n$  points and  $k = \frac{20 \log n}{\epsilon^2}$ . There exists a Lipschitz mapping  $f : \mathbb{R}^d \rightarrow \mathbb{R}^k$  such that for all  $u, v \in Q$ :

(independent of  $d$ )

$$(1 - \epsilon)\|u - v\|^2 \leq \|f(u) - f(v)\|^2 \leq (1 + \epsilon)\|u - v\|^2$$

**Theorem 1.2.** (Norm preservation) Let  $x \in \mathbb{R}^d$ . Assume that the entries in  $A \subset \mathbb{R}^{k \times d}$  are sampled independently from  $N(0, 1)$ . Then,

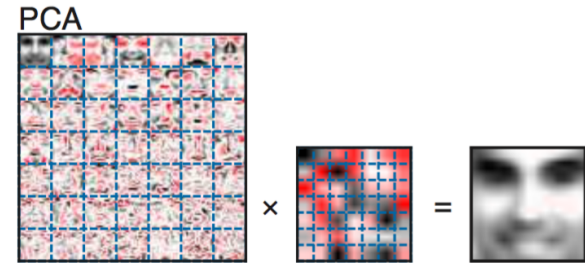
$$\Pr\left(\left(1 - \epsilon\right)\|x\|^2 \leq \left\|\frac{1}{\sqrt{k}}Ax\right\|^2 \leq \left(1 + \epsilon\right)\|x\|^2\right) \geq 1 - 2e^{-(\epsilon^2 - \epsilon^3)k/4}$$

# Other matrix factorizations

$$\mathbf{X} = \mathbf{U} \mathbf{S} \mathbf{V}^T$$

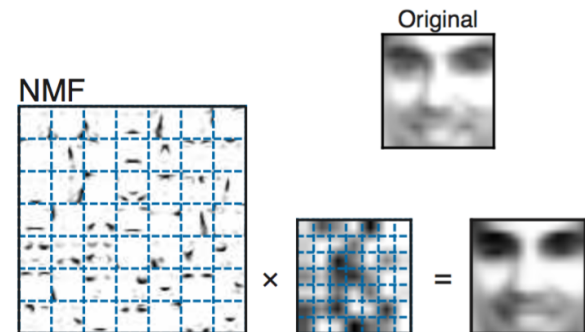
## Singular value decomposition

Elements of  $\mathbf{U}$ ,  $\mathbf{S}$ ,  $\mathbf{V}$  in  $\mathbb{R}$



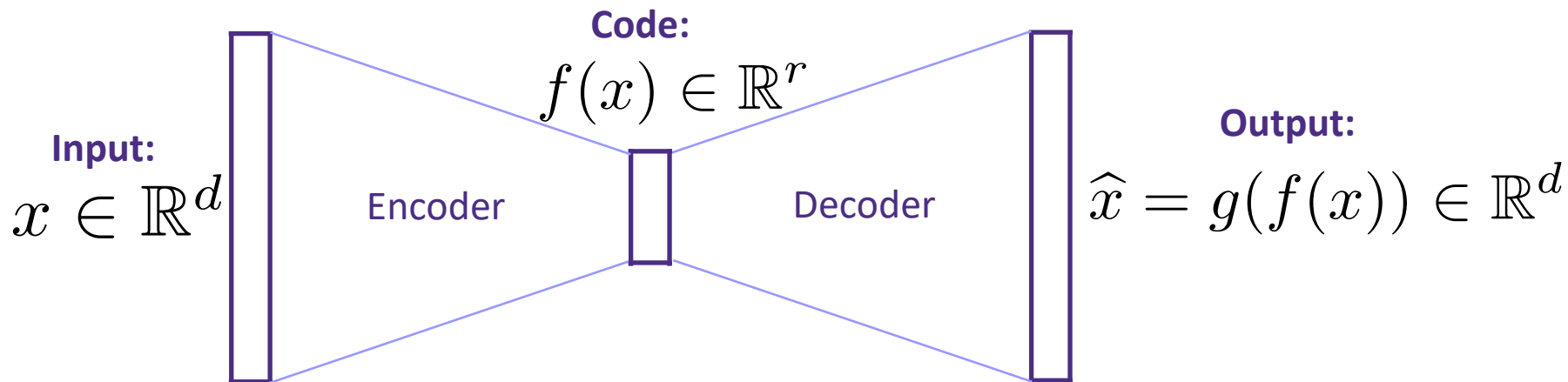
## Nonnegative matrix factorization (NMF)

Elements of  $\mathbf{U}$ ,  $\mathbf{S}$ ,  $\mathbf{V}$  in  $\mathbb{R}_+$



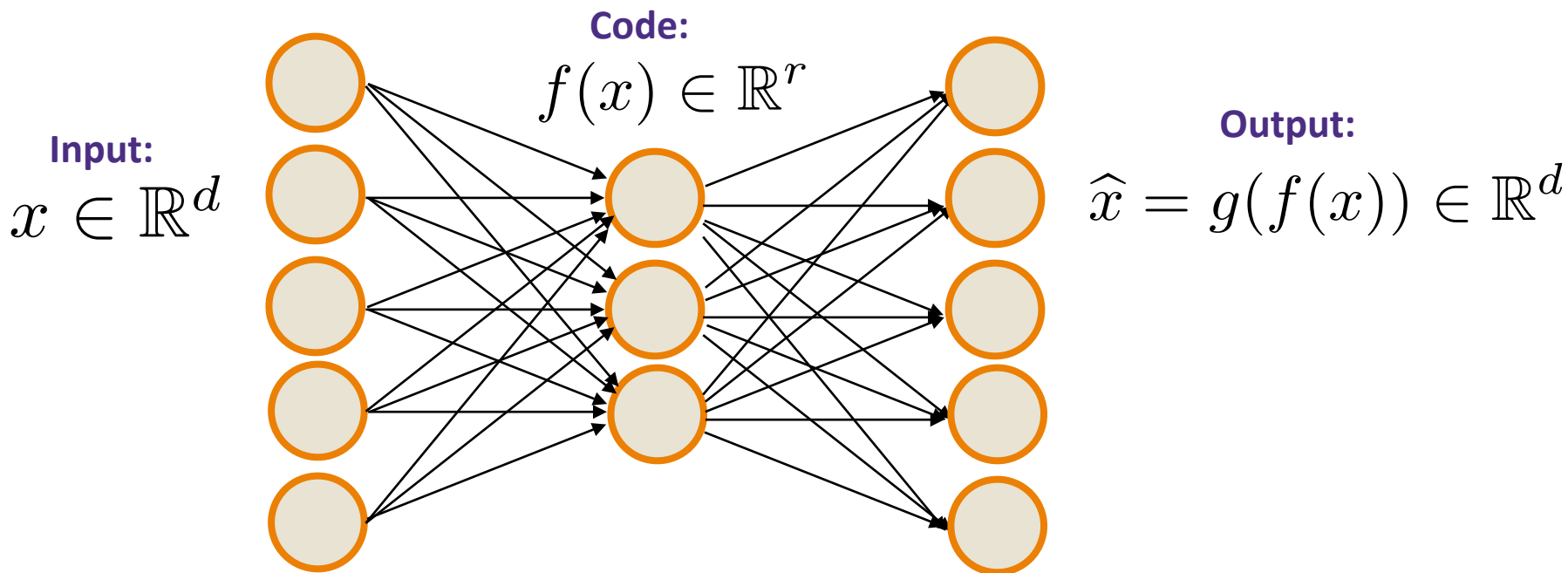
# Autoencoders

Find a low dimensional representation for your data by predicting your data



$$\underset{f, g}{\text{minimize}} \sum_{i=1}^n \|x_i - g(f(x_i))\|_2^2$$

# Autoencoders



$$\underset{f, g}{\text{minimize}} \sum_{i=1}^n \|x_i - g(f(x_i))\|_2^2$$

What if  $f(X) = Ax$  and  $g(y) = By$ ?

# Ridge Regression revisited

---

$$\hat{w}_{ridge} = \arg \min_w \|\mathbf{X}w - \mathbf{y}\|_2^2 + \lambda \|w\|_2^2$$

$$\hat{w}_{ridge} = (\mathbf{X}^T \mathbf{X} + \lambda I)^{-1} \mathbf{X}^T \mathbf{y}$$

*Singular vector decomposition (SVD):*  $\mathbf{X} = \mathbf{U} \mathbf{S} \mathbf{V}^T$

$$\hat{\mathbf{y}} = \mathbf{X}(\mathbf{X}^T \mathbf{X} + \lambda I)^{-1} \mathbf{X}^T \mathbf{y}$$

# Ridge Regression revisited

---

$$\hat{w}_{ridge} = \arg \min_w \|\mathbf{X}w - \mathbf{y}\|_2^2 + \lambda \|w\|_2^2$$

$$\hat{w}_{ridge} = (\mathbf{X}^T \mathbf{X} + \lambda I)^{-1} \mathbf{X}^T \mathbf{y}$$

Singular vector decomposition (SVD):  $\mathbf{X} - \mathbf{1}\bar{x}^T = \mathbf{U}\mathbf{S}\mathbf{V}^T$

$$\hat{\mathbf{y}} = \mathbf{X}(\mathbf{X}^T \mathbf{X} + \lambda I)^{-1} \mathbf{X}^T \mathbf{y}$$

$$\hat{\mathbf{y}} = \sum_{i=1}^d u_i u_i^T \frac{s_i^2}{s_i^2 + \lambda} y_i$$

$$\mathbf{U} = [u_1, \dots, u_d]$$

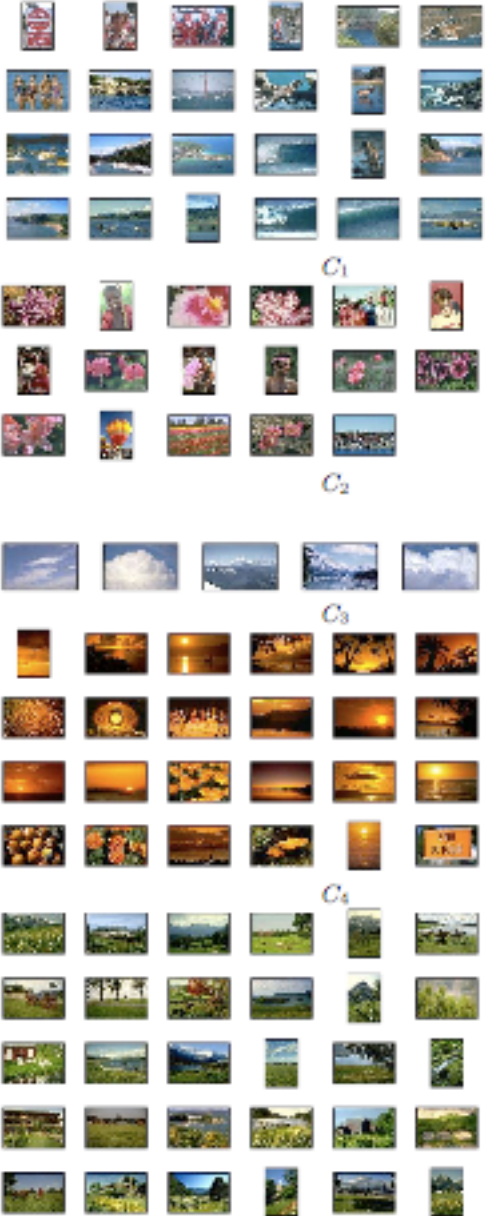
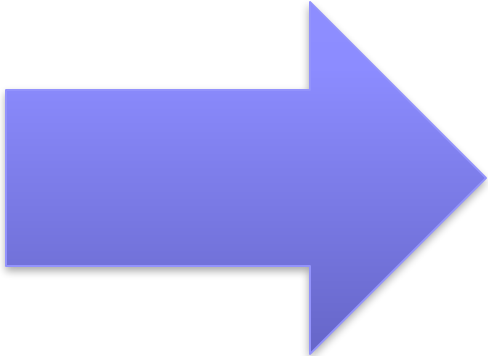
$$\mathbf{S} = \text{diag}(s_1, \dots, s_d)$$

# Clustering

# K-means

---

# Clustering images



[Goldberger et al.]

# Clustering web search results



web news images wikipedia blogs jobs more »

race

Search

advanced preferences

clusters sources sites

All Results (238)

remix

- Car (28)
  - Race cars (7)
  - Photos, Races Scheduled (5)
  - Game (4)
  - Track (3)
  - Nascar (2)
  - Equipment And Safety (2)
  - Other Topics (7)
- Photos (22)
- Game (14)
- Definition (13)
- Team (18)
- Human (8)**
  - Classification Of Human (2)
  - Statement, Evolved (2)
  - Other Topics (4)
- Weekend (8)
- Ethnicity And Race (7)
- Race for the Cure (8)
- Race Information (8)

more | all clusters

find in clusters:

Find

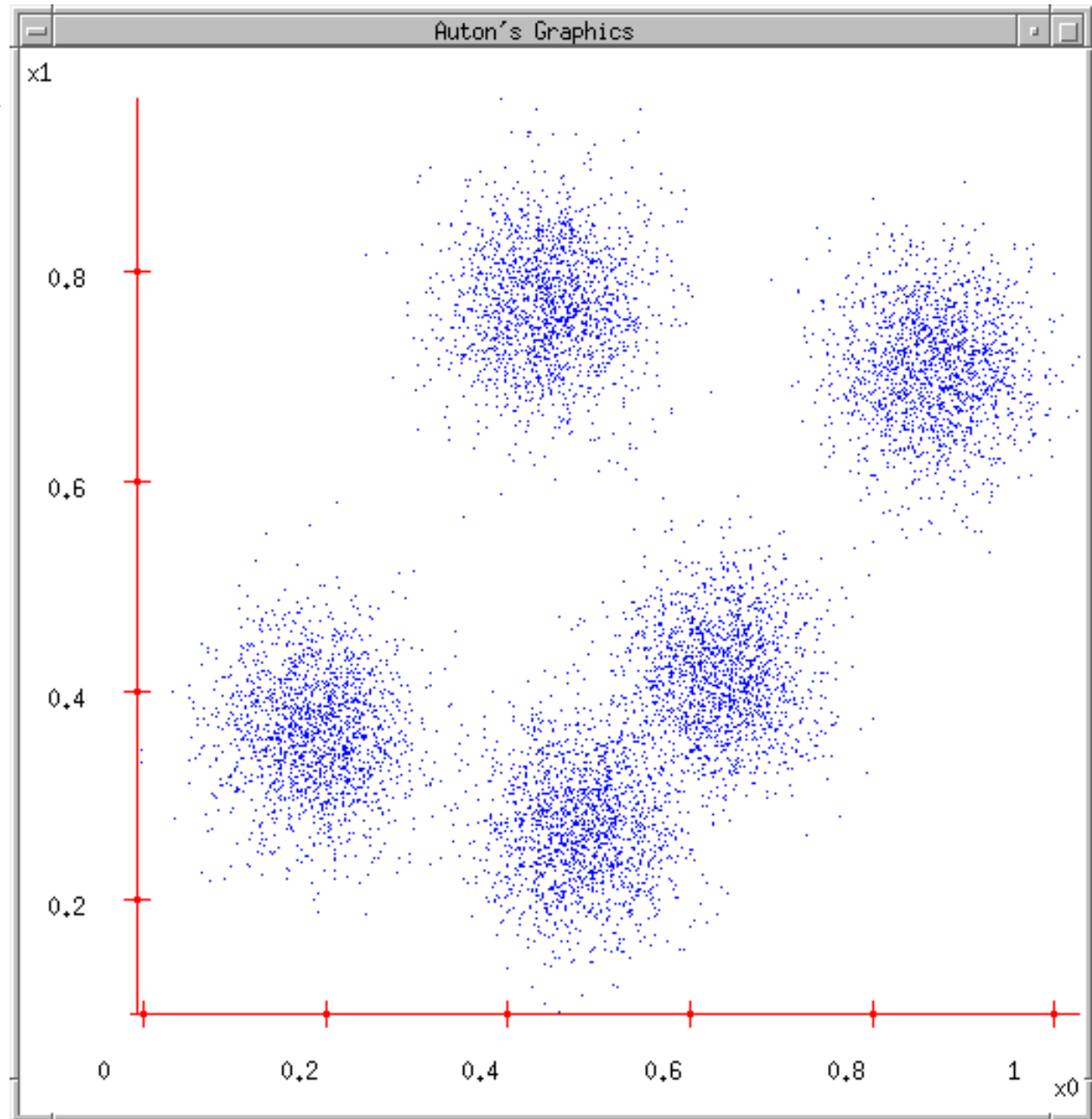
Cluster Human contains 8 documents.

Search Results

- [Race \(classification of human beings\) - Wikipedia, the free ...](#)  
The term **race** or racial group usually refers to the concept of dividing **humans** into populations or groups on the basis of various sets of characteristics. The most widely used **human** racial categories are based on visible traits (especially skin color, cranial or facial features and hair texture), and self-identification. Conceptions of **race**, as well as specific ways of grouping **races**, vary by culture and over time, and are often controversial for scientific as well as social and political reasons. History · Modern debates · Political and ...  
[en.wikipedia.org/wiki/Race\\_\(classification\\_of\\_human\\_beings\)](http://en.wikipedia.org/wiki/Race_(classification_of_human_beings)) - [cache] - Live, Ask
- [Race - Wikipedia, the free encyclopedia](#)  
General. **Racing** competitions The **Race** (yachting **race**), or La course du millénaire, a no-rules round-the-world sailing event; **Race** (biology), classification of flora and fauna; **Race** (classification of **human** beings) **Race** and ethnicity in the United States Census, official definitions of "**race**" used by the US Census Bureau; **Race** and genetics, notion of racial classifications based on genetics. Historical definitions of **race**; **Race** (bearing), the inner and outer rings of a rolling-element bearing. **RACE** in molecular biology "Rapid ... General · Surnames · Television · Music · Literature · Video games  
[en.wikipedia.org/wiki/Race](http://en.wikipedia.org/wiki/Race) - [cache] - Live, Ask
- [Publications | Human Rights Watch](#)  
The use of torture, unlawful rendition, secret prisons, unfair trials, ... Risks to Migrants, Refugees, and Asylum Seekers in Egypt and Israel ... In the run-up to the Beijing Olympics in August 2008, ...  
[www.hrw.org/backgrounder/usa/race](http://www.hrw.org/backgrounder/usa/race) - [cache] - Ask
- [Amazon.com: Race: The Reality Of Human Differences: Vincent Sarich ...](#)  
Amazon.com: **Race: The Reality Of Human Differences: Vincent Sarich, Frank Miele: Books ...** From Publishers Weekly Sarich, a Berkeley emeritus anthropologist, and Miele, an editor ...  
[www.amazon.com/Race-Reality-Differences-Vincent-Sarich/dp/0813340861](http://www.amazon.com/Race-Reality-Differences-Vincent-Sarich/dp/0813340861) - [cache] - Live
- [AAPA Statement on Biological Aspects of Race](#)  
AAPA Statement on Biological Aspects of **Race** ... Published in the American Journal of Physical Anthropology, vol. 101, pp 569-570, 1996 ... PREAMBLE As scientists who study **human** evolution and variation, ...  
[www.physanth.org/positions/race.html](http://www.physanth.org/positions/race.html) - [cache] - Ask
- [race: Definition from Answers.com](#)  
**race** n. A local geographic or global **human** population distinguished as a more or less distinct group by genetically transmitted physical  
[www.answers.com/topic/race-1](http://www.answers.com/topic/race-1) - [cache] - Live
- [Dopefish.com](#)  
Site for newbies as well as experienced Dopefish followers, chronicling the birth of the Dopefish, its numerous appearances in several computer games, and its eventual take-over of the **human** **race**. Maintained by Mr. Dopefish himself, Joe Siegler of Apogee Software.  
[www.dopefish.com](http://www.dopefish.com) - [cache] - Open Directory

# Some Data

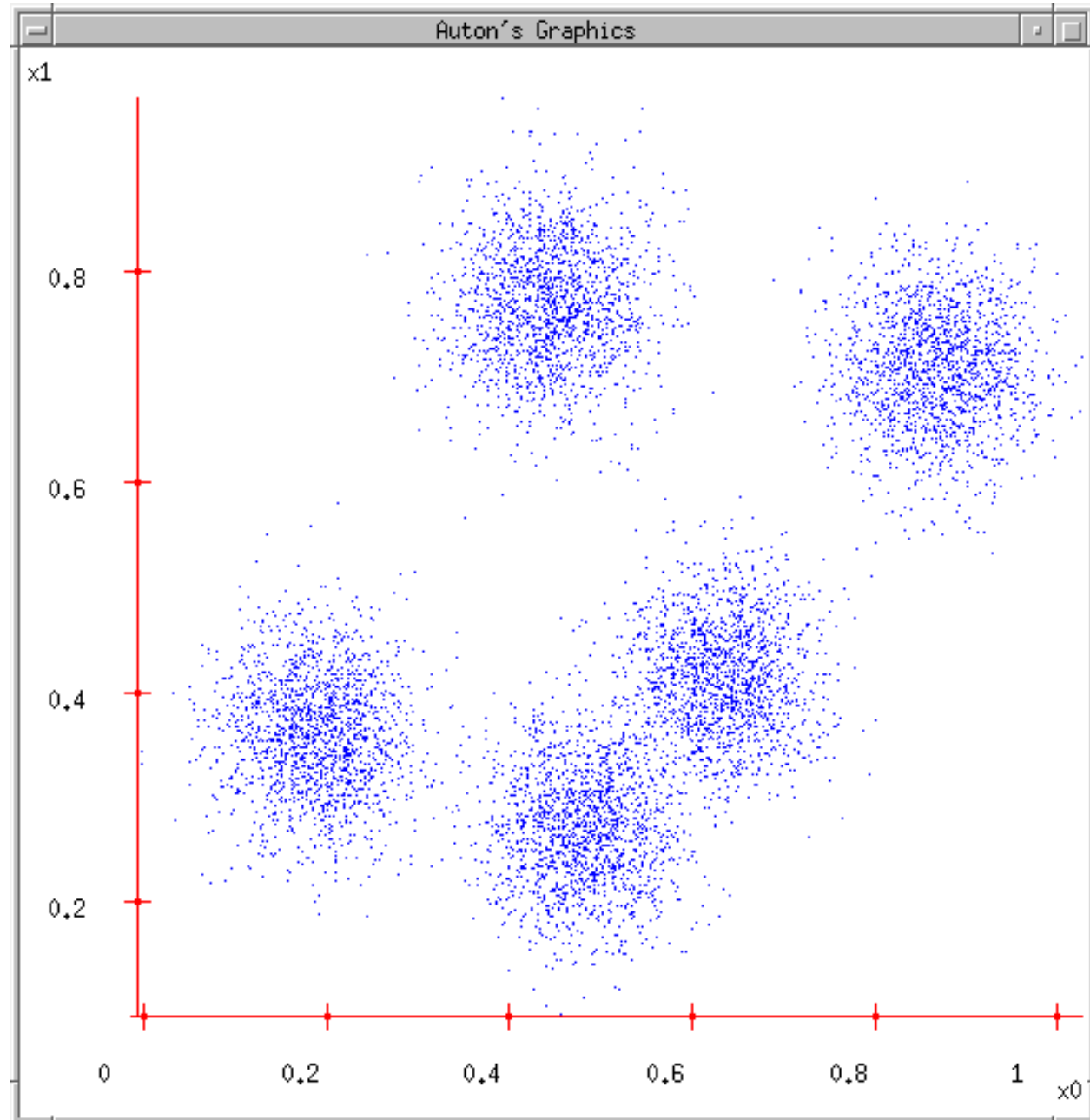
---



# K-means

---

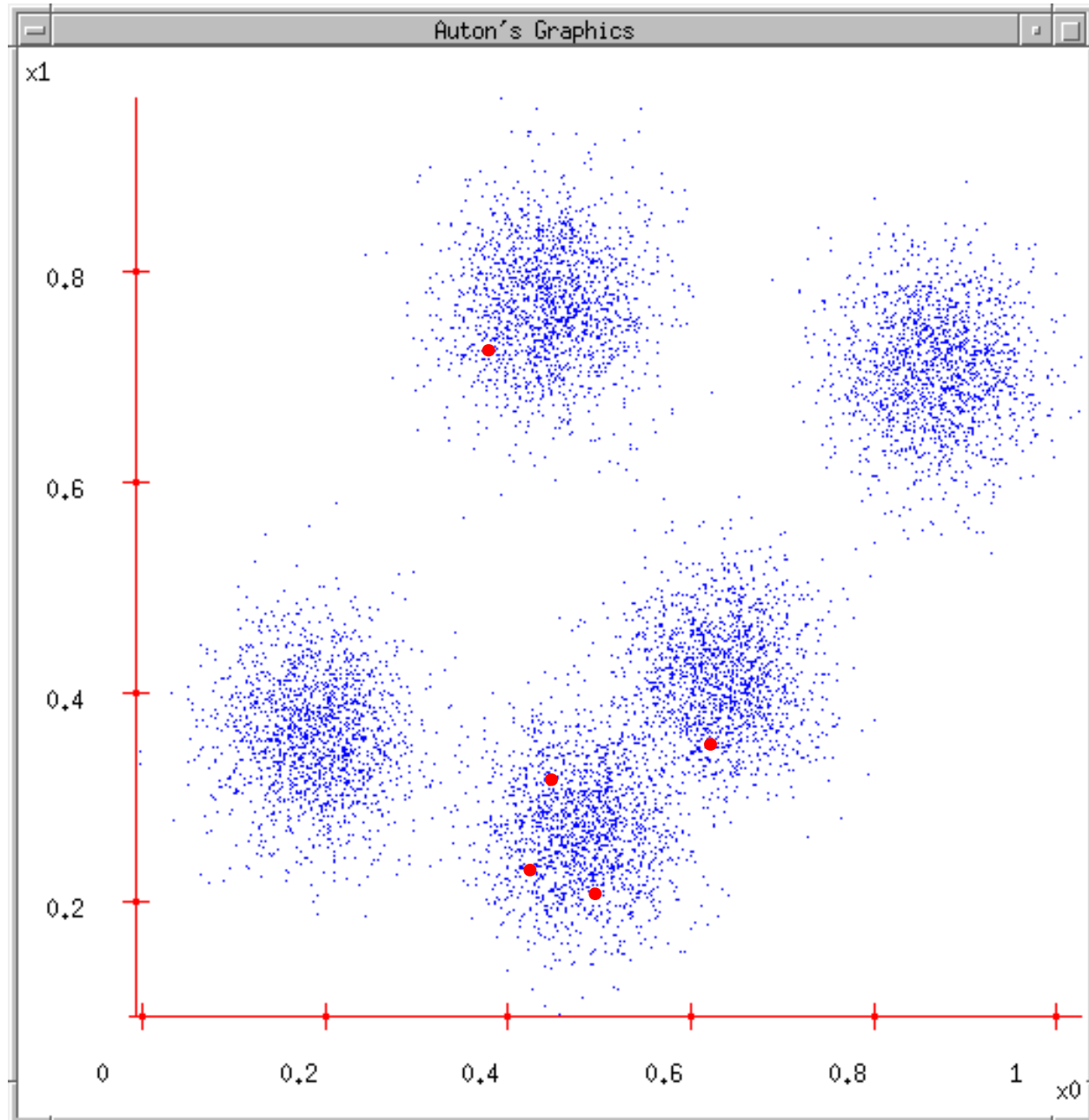
1. Ask user how many clusters they'd like.  
(e.g.  $k=5$ )



# K-means

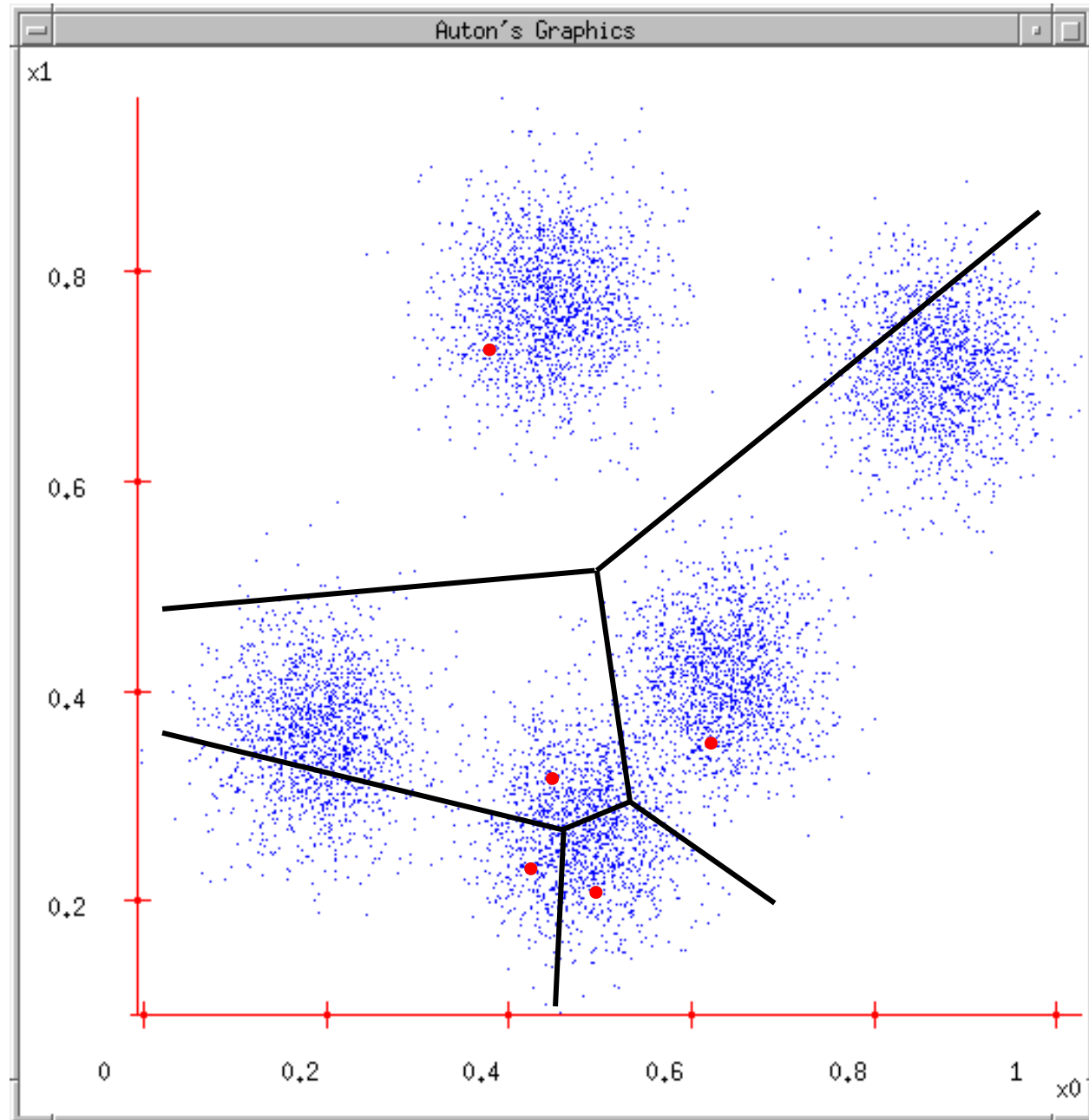
---

1. Ask user how many clusters they'd like.  
(e.g.  $k=5$ )
2. Randomly guess  $k$  cluster Center locations



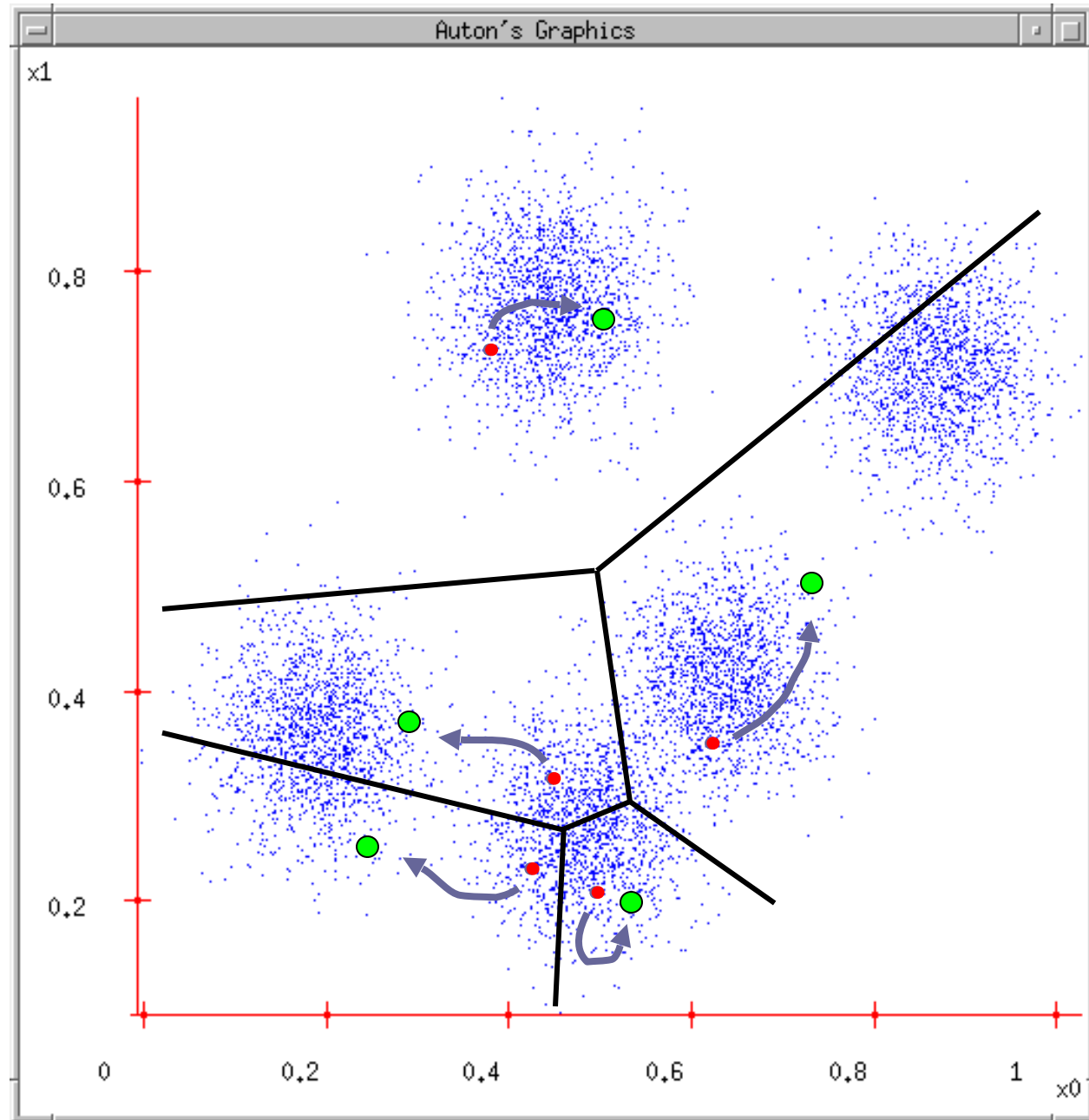
# K-means

1. Ask user how many clusters they'd like.  
(e.g.  $k=5$ )
2. Randomly guess  $k$  cluster Center locations
3. Each datapoint finds out which Center it's closest to. (Thus each Center "owns" a set of datapoints)



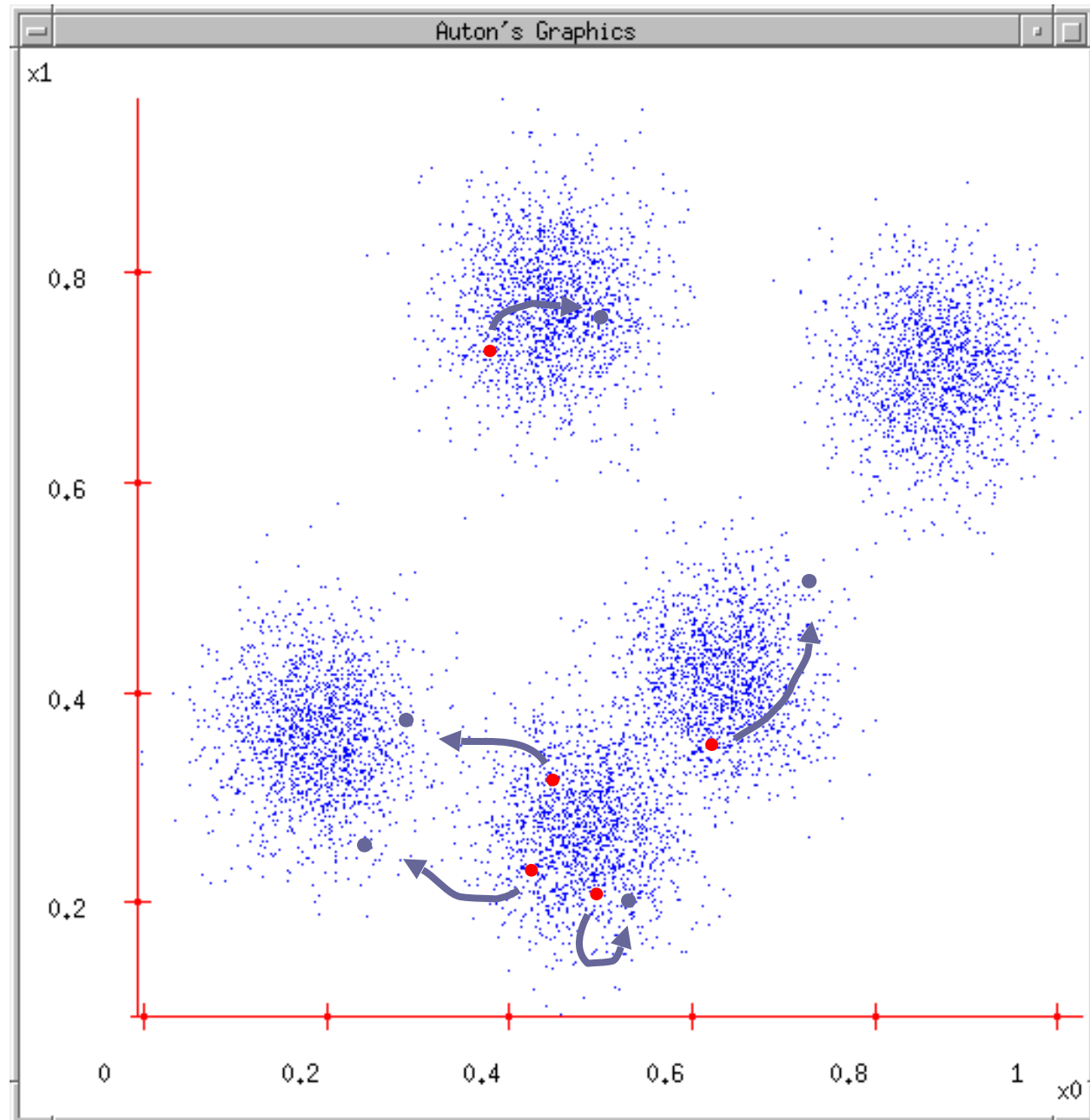
# K-means

1. Ask user how many clusters they'd like.  
(e.g.  $k=5$ )
2. Randomly guess  $k$  cluster Center locations
3. Each datapoint finds out which Center it's closest to.
4. Each Center finds the centroid of the points it owns



# K-means

1. Ask user how many clusters they'd like.  
(e.g.  $k=5$ )
2. Randomly guess  $k$  cluster Center locations
3. Each datapoint finds out which Center it's closest to.
4. Each Center finds the centroid of the points it owns...
5. ...and jumps there
6. ...Repeat until terminated!



# K-means

---

- Randomly initialize  $k$  centers
  - $\mu^{(0)} = \mu_1^{(0)}, \dots, \mu_k^{(0)}$
- Classify: Assign each point  $j \in \{1, \dots, N\}$  to nearest center:
  - $C^{(t)}(j) \leftarrow \arg \min_i \|\mu_i - x_j\|^2$
- Recenter:  $\mu_i$  becomes centroid of its point:
  - $\mu_i^{(t+1)} \leftarrow \arg \min_{\mu} \sum_{j: C(j)=i} \|\mu - x_j\|^2$

# Does K-means converge???

## Part 1

---

- Optimize potential function:

$$\min_{\mu} \min_C F(\mu, C) = \min_{\mu} \min_C \sum_{i=1}^k \sum_{j:C(j)=i} \|\mu_i - x_j\|^2$$

- Fix  $\mu$ , optimize C

# Does K-means converge???

## Part 2

---

- Optimize potential function:

$$\min_{\mu} \min_C F(\mu, C) = \min_{\mu} \min_C \sum_{i=1}^k \sum_{j:C(j)=i} \|\mu_i - x_j\|^2$$

- Fix C, optimize  $\mu$

# Vector Quantization, Fisher Vectors

## Vector Quantization (for compression)

1. Represent image as grid of patches
2. Run k-means on the patches to build code book
3. Represent each patch as a code word.



**FIGURE 14.9.** *Sir Ronald A. Fisher (1890 – 1962) was one of the founders of modern day statistics, to whom we owe maximum-likelihood, sufficiency, and many other fundamental concepts. The image on the left is a  $1024 \times 1024$  grayscale image at 8 bits per pixel. The center image is the result of  $2 \times 2$  block VQ, using 200 code vectors, with a compression rate of 1.9 bits/pixel. The right image uses only four code vectors, with a compression rate of 0.50 bits/pixel*

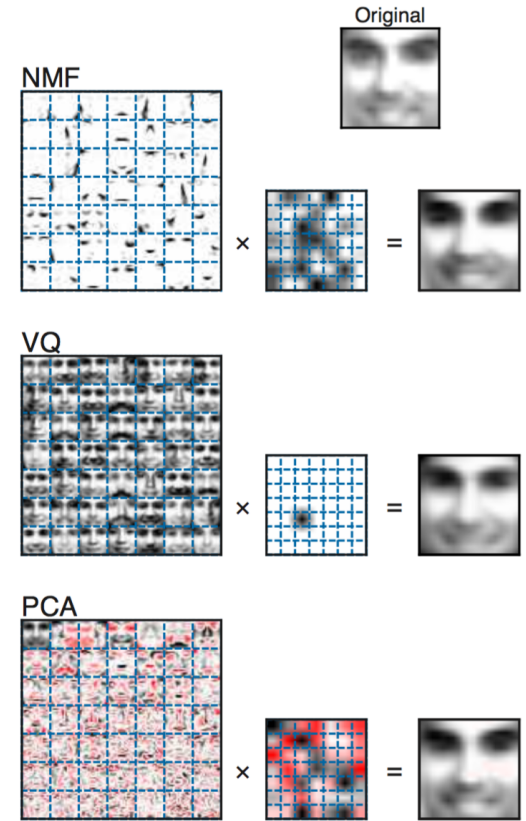
# Vector Quantization, Fisher Vectors

## Vector Quantization (for compression)

1. Represent image as grid of patches
2. Run k-means on the patches to build code book
3. Represent each patch as a code word.



**FIGURE 14.9.** *Sir Ronald A. Fisher (1890 – 1962) was one of the founders of modern day statistics, to whom we owe maximum-likelihood, sufficiency, and many other fundamental concepts. The image on the left is a  $1024 \times 1024$  grayscale image at 8 bits per pixel. The center image is the result of  $2 \times 2$  block VQ, using 200 code vectors, with a compression rate of 1.9 bits/pixel. The right image uses only four code vectors, with a compression rate of 0.50 bits/pixel*



# Vector Quantization, Fisher Vectors

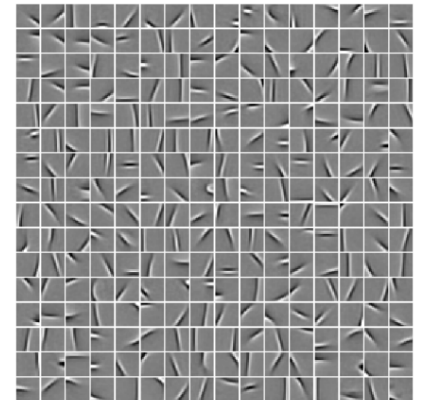
## Vector Quantization (for compression)

1. Represent image as grid of patches
2. Run k-means on the patches to build code book
3. Represent each patch as a code word.



**FIGURE 14.9.** *Sir Ronald A. Fisher (1890 – 1962) was one of the founders of modern day statistics, to whom we owe maximum-likelihood, sufficiency, and many other fundamental concepts. The image on the left is a  $1024 \times 1024$  grayscale image at 8 bits per pixel. The center image is the result of  $2 \times 2$  block VQ, using 200 code vectors, with a compression rate of 1.9 bits/pixel. The right image uses only four code vectors, with a compression rate of 0.50 bits/pixel*

Typical output of  
on patches



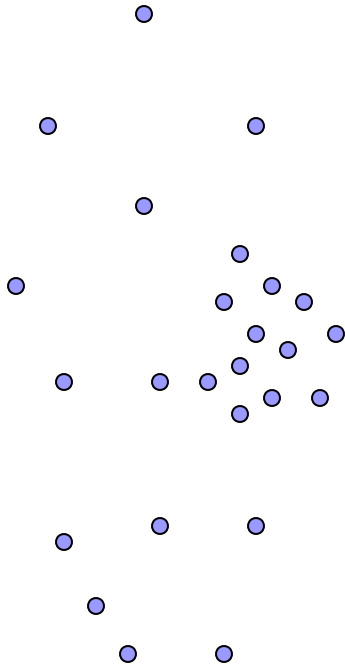
Similar reduced representation can be used as a feature vector

Coates, Ng, *Learning Feature Representations with K-means*, 2012

# Mixtures of Gaussians

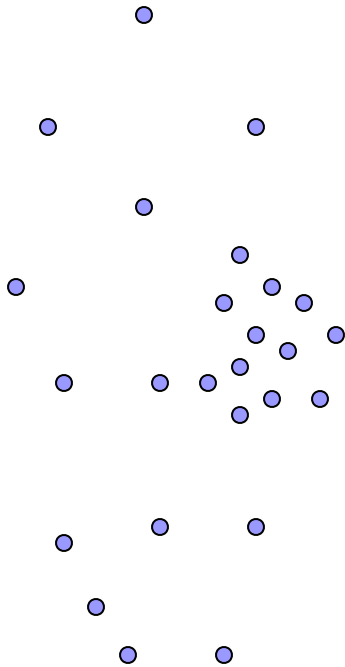
---

# (One) bad case for k-means



# (One) bad case for k-means

- Clusters may overlap
- Some clusters may be “wider” than others



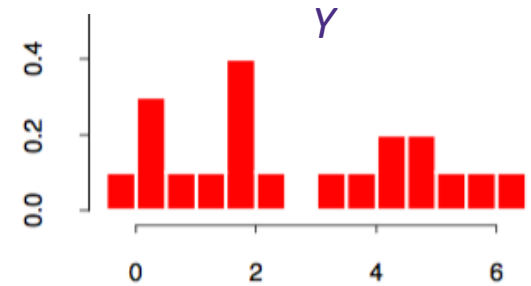
# Mixture models

$$Y_1 \sim N(\mu_1, \sigma_1^2),$$

$$Y_2 \sim N(\mu_2, \sigma_2^2),$$

$$Y = (1 - \Delta) \cdot Y_1 + \Delta \cdot Y_2,$$

$$\Delta \in \{0, 1\} \text{ with } \Pr(\Delta = 1) = \pi$$



$\mathbf{Z} = \{y_i\}_{i=1}^n$  is observed data

If  $\phi_\theta(x)$  is Gaussian density with parameters  $\theta = (\mu, \sigma^2)$  then

$$\ell(\theta; \mathbf{Z}) = \sum_{i=1}^n \log[(1 - \pi)\phi_{\theta_1}(y_i) + \pi\phi_{\theta_2}(y_i)]$$

# Mixture models

$$Y_1 \sim N(\mu_1, \sigma_1^2),$$

$$Y_2 \sim N(\mu_2, \sigma_2^2),$$

$$Y = (1 - \Delta) \cdot Y_1 + \Delta \cdot Y_2,$$

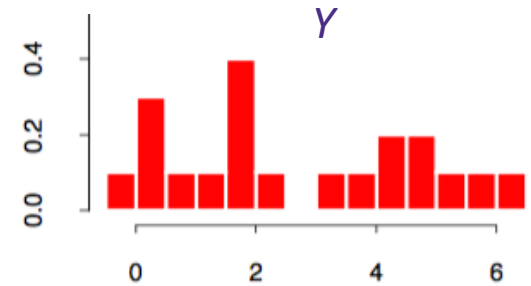
$$\Delta \in \{0, 1\} \text{ with } \Pr(\Delta = 1) = \pi$$

$$\theta = (\pi, \theta_1, \theta_2) = (\pi, \mu_1, \sigma_1^2, \mu_2, \sigma_2^2)$$

If  $\phi_\theta(x)$  is Gaussian density with parameters  $\theta = (\mu, \sigma^2)$  then

$$\ell(\theta; y_i, \Delta_i = 0) =$$

$$\ell(\theta; y_i, \Delta_i = 1) =$$



$\mathbf{Z} = \{y_i\}_{i=1}^n$  is observed data

$\mathbf{\Delta} = \{\Delta_i\}_{i=1}^n$  is unobserved data

# Mixture models

$$Y_1 \sim N(\mu_1, \sigma_1^2),$$

$$Y_2 \sim N(\mu_2, \sigma_2^2),$$

$$Y = (1 - \Delta) \cdot Y_1 + \Delta \cdot Y_2,$$

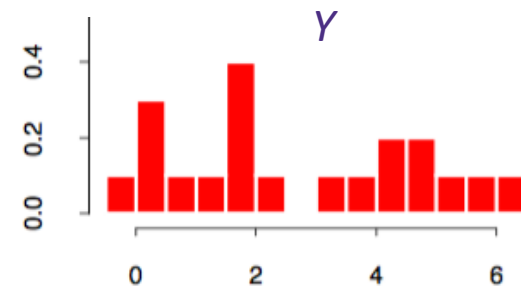
$$\Delta \in \{0, 1\} \text{ with } \Pr(\Delta = 1) = \pi$$

$$\theta = (\pi, \theta_1, \theta_2) = (\pi, \mu_1, \sigma_1^2, \mu_2, \sigma_2^2)$$

If  $\phi_\theta(x)$  is Gaussian density with parameters  $\theta = (\mu, \sigma^2)$  then

$$\ell(\theta; \mathbf{Z}, \mathbf{\Delta}) = \sum_{i=1}^n (1 - \Delta_i) \log[(1 - \pi)\phi_{\theta_1}(y_i)] + \Delta_i \log(\pi\phi_{\theta_2}(y_i))$$

If we knew  $\mathbf{\Delta}$ , how would we choose  $\theta$ ?



$\mathbf{Z} = \{y_i\}_{i=1}^n$  is observed data

$\mathbf{\Delta} = \{\Delta_i\}_{i=1}^n$  is unobserved data

# Mixture models

$$Y_1 \sim N(\mu_1, \sigma_1^2),$$

$$Y_2 \sim N(\mu_2, \sigma_2^2),$$

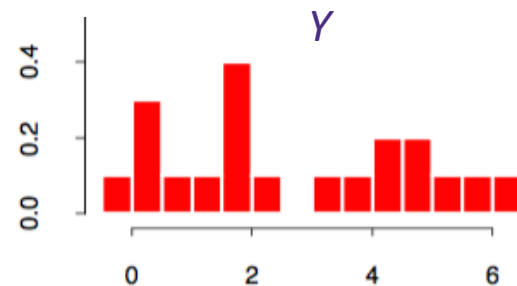
$$Y = (1 - \Delta) \cdot Y_1 + \Delta \cdot Y_2,$$

$$\Delta \in \{0, 1\} \text{ with } \Pr(\Delta = 1) = \pi$$

$$\theta = (\pi, \theta_1, \theta_2) = (\pi, \mu_1, \sigma_1^2, \mu_2, \sigma_2^2)$$

If  $\phi_\theta(x)$  is Gaussian density with parameters  $\theta = (\mu, \sigma^2)$  then

$$\ell(\theta; \mathbf{Z}, \mathbf{\Delta}) = \sum_{i=1}^n (1 - \Delta_i) \log[(1 - \pi)\phi_{\theta_1}(y_i)] + \Delta_i \log(\pi\phi_{\theta_2}(y_i))$$



$\mathbf{Z} = \{y_i\}_{i=1}^n$  is observed data

$\mathbf{\Delta} = \{\Delta_i\}_{i=1}^n$  is unobserved data

If we knew  $\theta$ , how would we choose  $\mathbf{\Delta}$ ?

# Mixture models

$$Y_1 \sim N(\mu_1, \sigma_1^2),$$

$$Y_2 \sim N(\mu_2, \sigma_2^2),$$

$$Y = (1 - \Delta) \cdot Y_1 + \Delta \cdot Y_2,$$

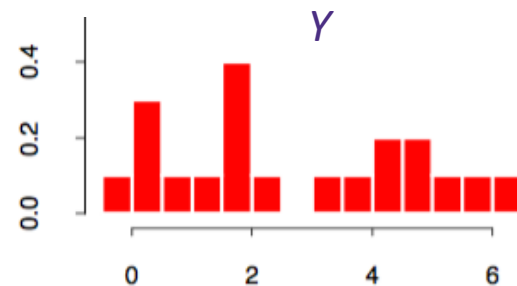
$$\Delta \in \{0, 1\} \text{ with } \Pr(\Delta = 1) = \pi$$

$$\theta = (\pi, \theta_1, \theta_2) = (\pi, \mu_1, \sigma_1^2, \mu_2, \sigma_2^2)$$

If  $\phi_\theta(x)$  is Gaussian density with parameters  $\theta = (\mu, \sigma^2)$  then

$$\ell(\theta; \mathbf{Z}, \mathbf{\Delta}) = \sum_{i=1}^n (1 - \Delta_i) \log[(1 - \pi)\phi_{\theta_1}(y_i)] + \Delta_i \log(\pi\phi_{\theta_2}(y_i))$$

$$\gamma_i(\theta) = \mathbb{E}[\Delta_i | \theta, \mathbf{Z}] =$$



$\mathbf{Z} = \{y_i\}_{i=1}^n$  is observed data

$\mathbf{\Delta} = \{\Delta_i\}_{i=1}^n$  is unobserved data

# Mixture models

---

**Algorithm 8.1** *EM Algorithm for Two-component Gaussian Mixture.*

---

1. Take initial guesses for the parameters  $\hat{\mu}_1, \hat{\sigma}_1^2, \hat{\mu}_2, \hat{\sigma}_2^2, \hat{\pi}$  (see text).
2. *Expectation Step*: compute the responsibilities

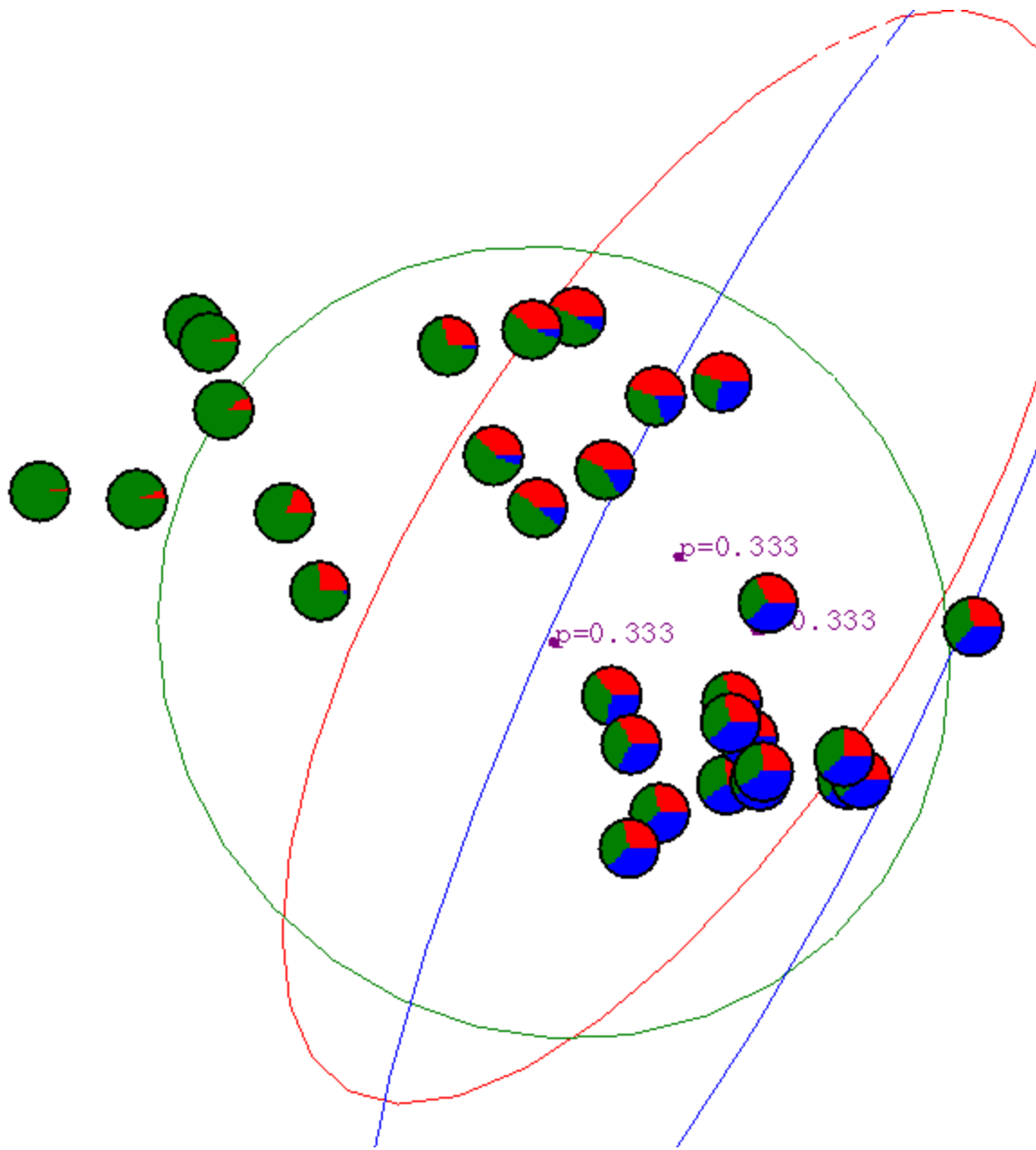
$$\hat{\gamma}_i = \frac{\hat{\pi} \phi_{\hat{\theta}_2}(y_i)}{(1 - \hat{\pi}) \phi_{\hat{\theta}_1}(y_i) + \hat{\pi} \phi_{\hat{\theta}_2}(y_i)}, \quad i = 1, 2, \dots, N. \quad (8.42)$$

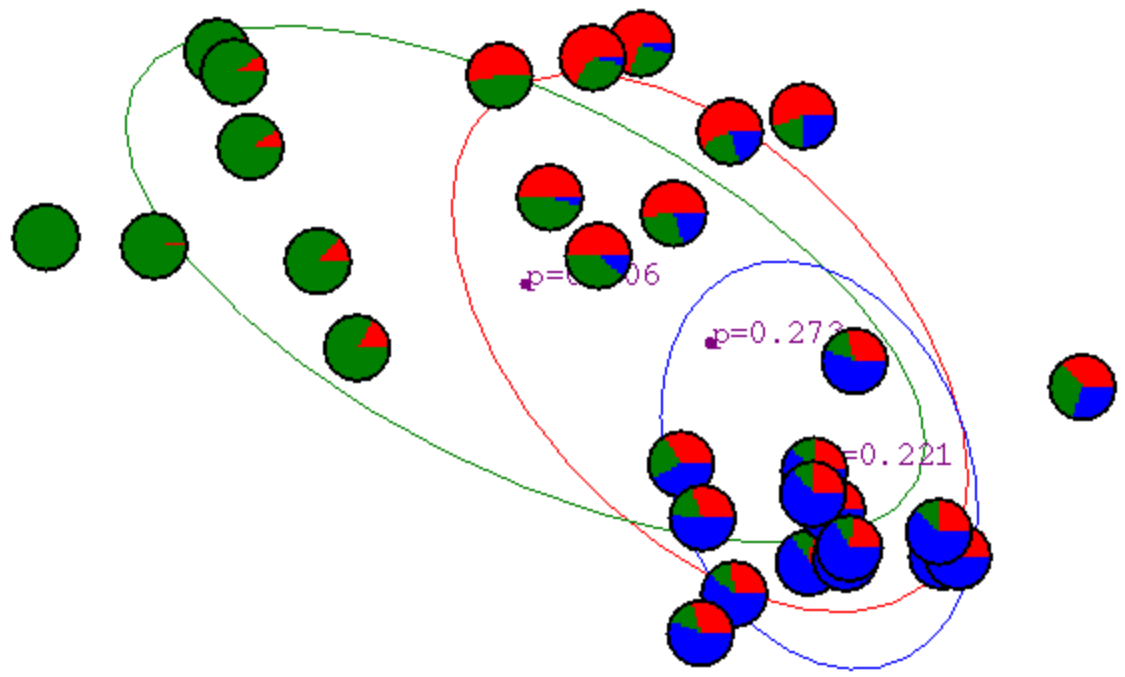
3. *Maximization Step*: compute the weighted means and variances:

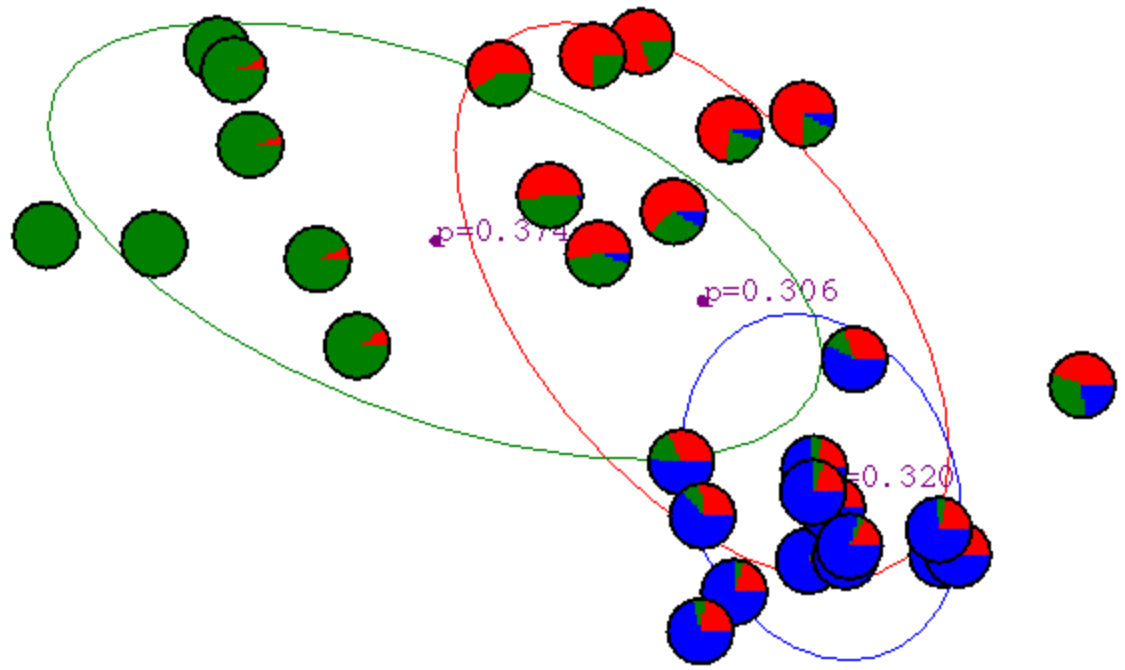
$$\hat{\mu}_1 = \frac{\sum_{i=1}^N (1 - \hat{\gamma}_i) y_i}{\sum_{i=1}^N (1 - \hat{\gamma}_i)}, \quad \hat{\sigma}_1^2 = \frac{\sum_{i=1}^N (1 - \hat{\gamma}_i) (y_i - \hat{\mu}_1)^2}{\sum_{i=1}^N (1 - \hat{\gamma}_i)},$$
$$\hat{\mu}_2 = \frac{\sum_{i=1}^N \hat{\gamma}_i y_i}{\sum_{i=1}^N \hat{\gamma}_i}, \quad \hat{\sigma}_2^2 = \frac{\sum_{i=1}^N \hat{\gamma}_i (y_i - \hat{\mu}_2)^2}{\sum_{i=1}^N \hat{\gamma}_i},$$

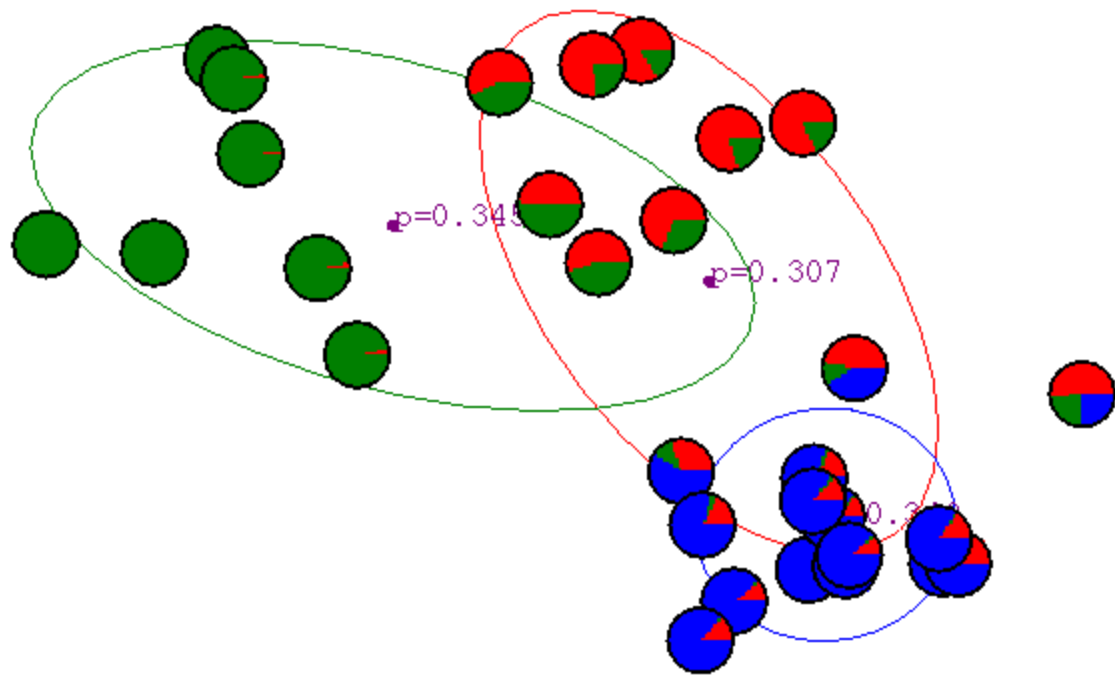
and the mixing probability  $\hat{\pi} = \sum_{i=1}^N \hat{\gamma}_i / N$ .

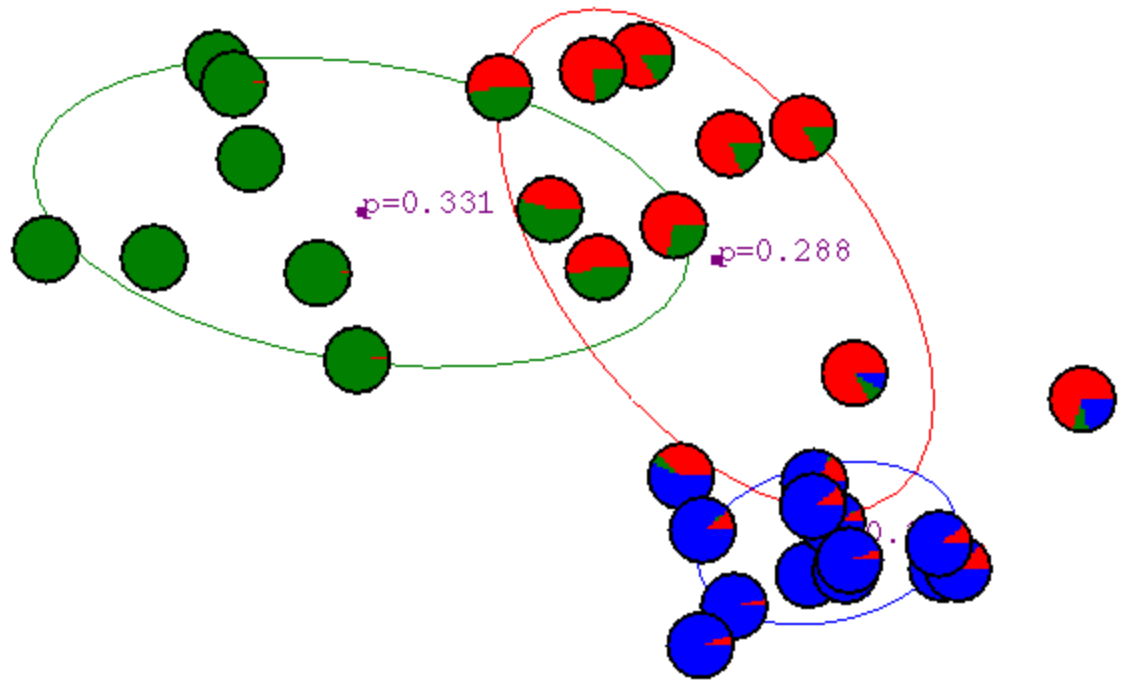
4. Iterate steps 2 and 3 until convergence.
-

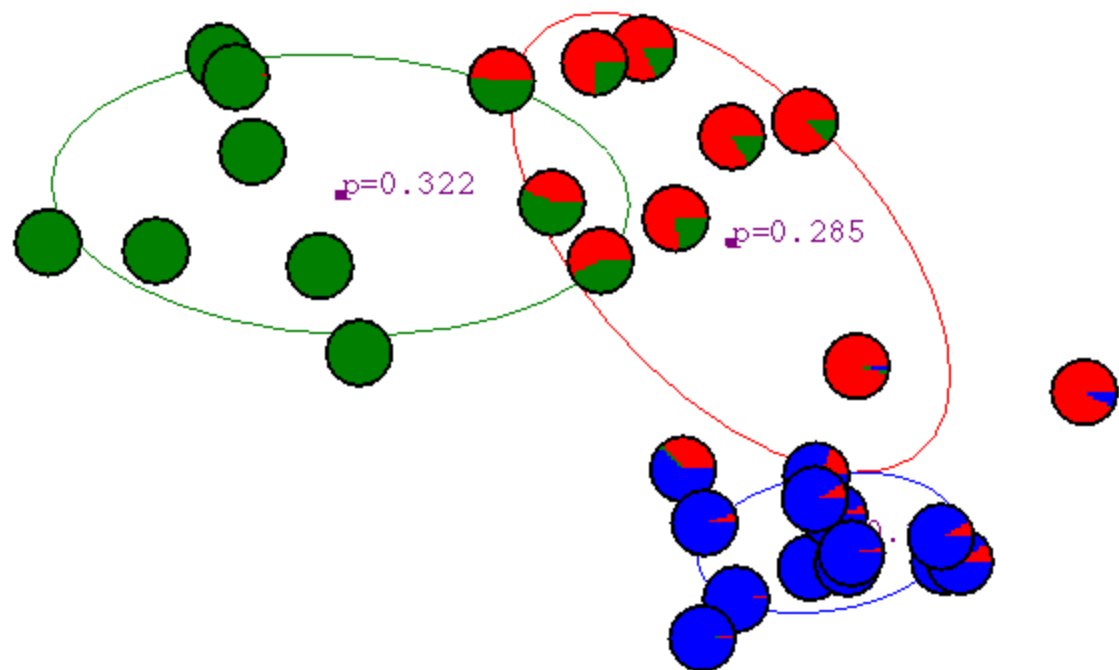


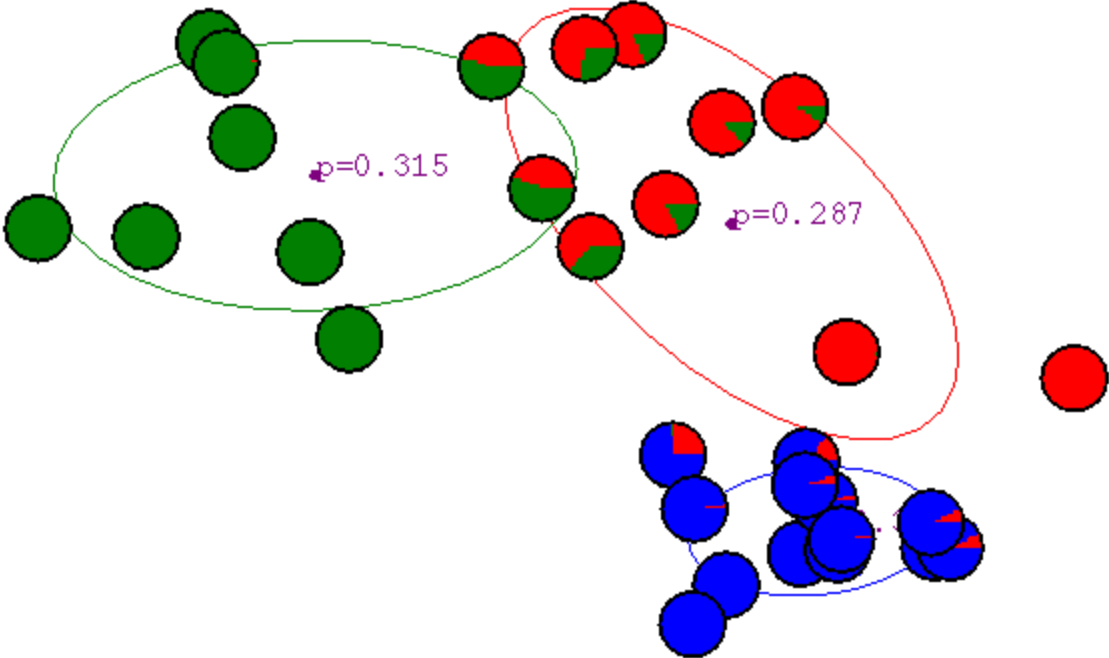


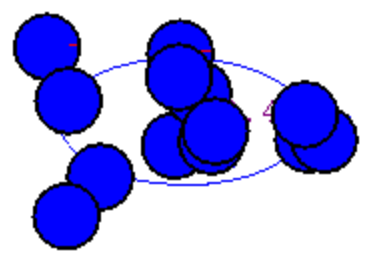
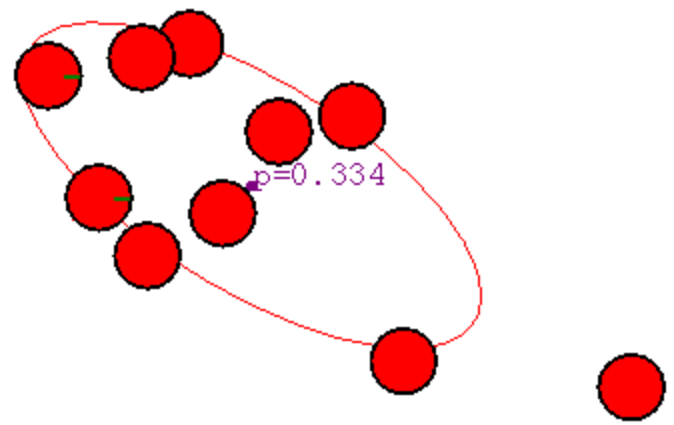
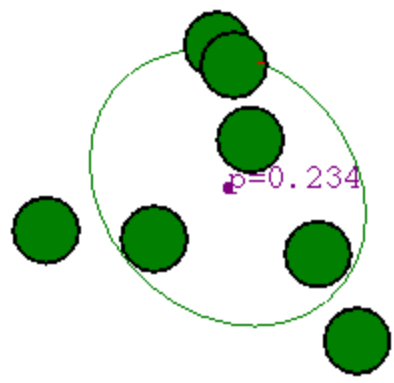


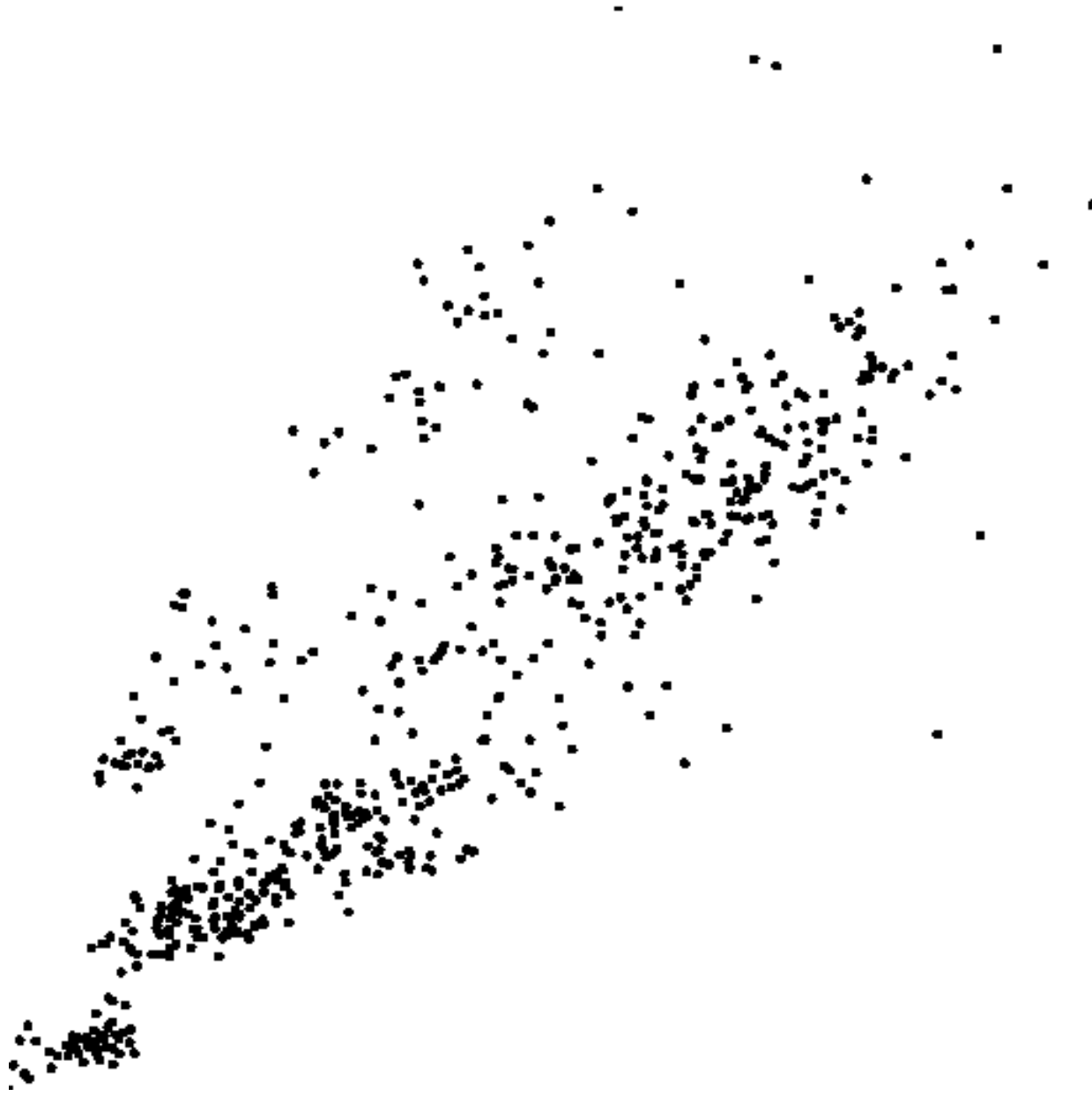


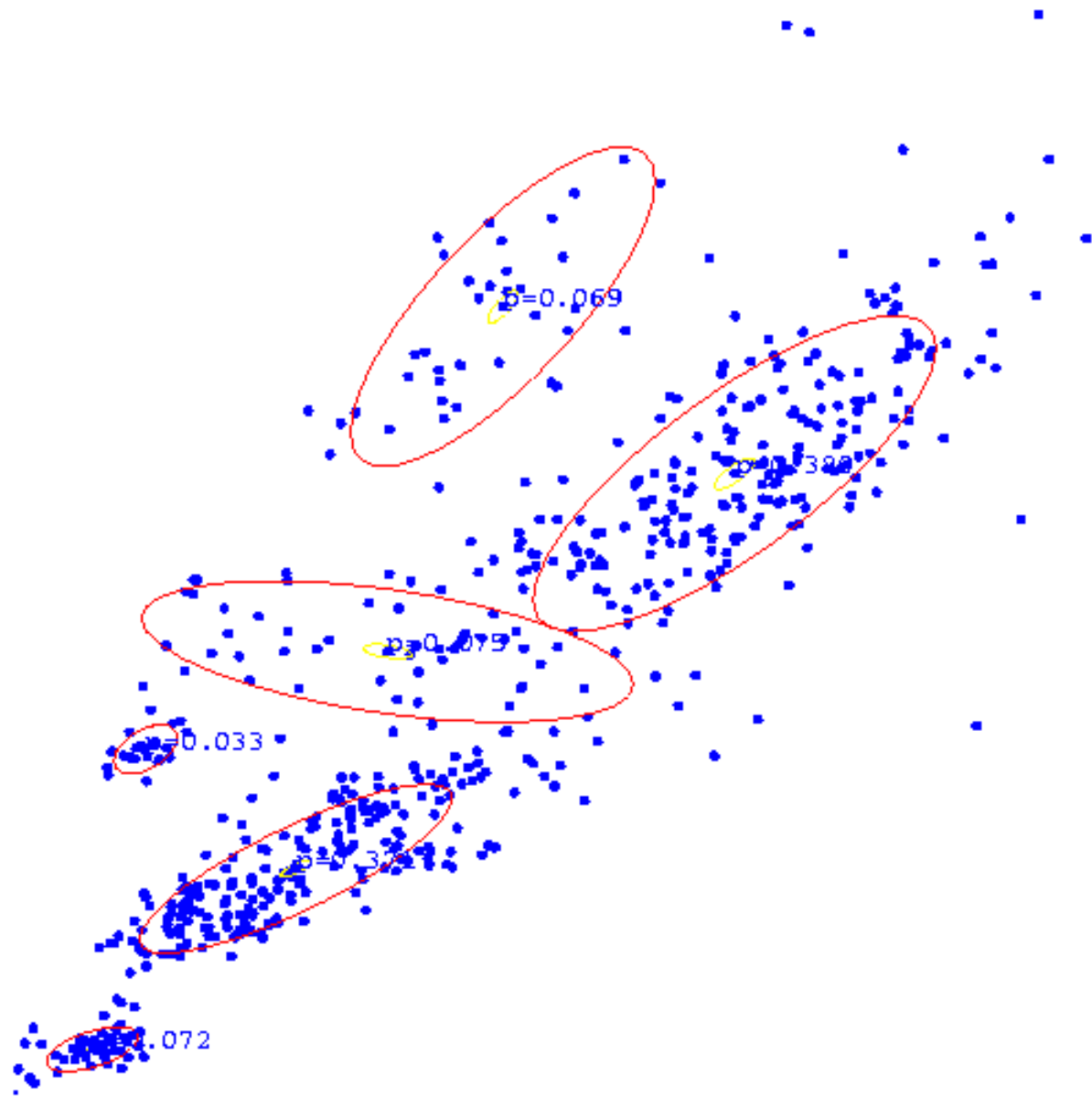


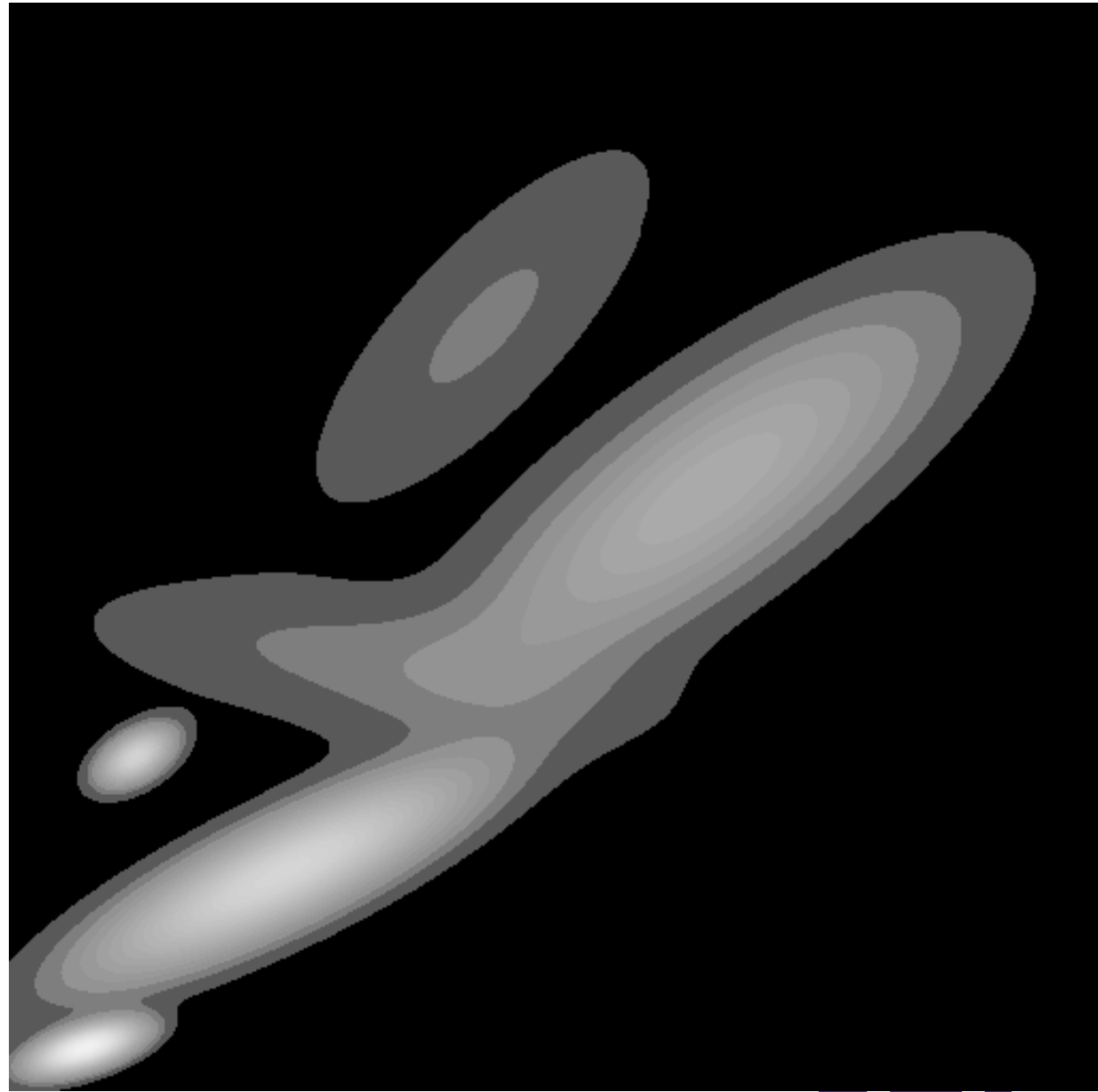








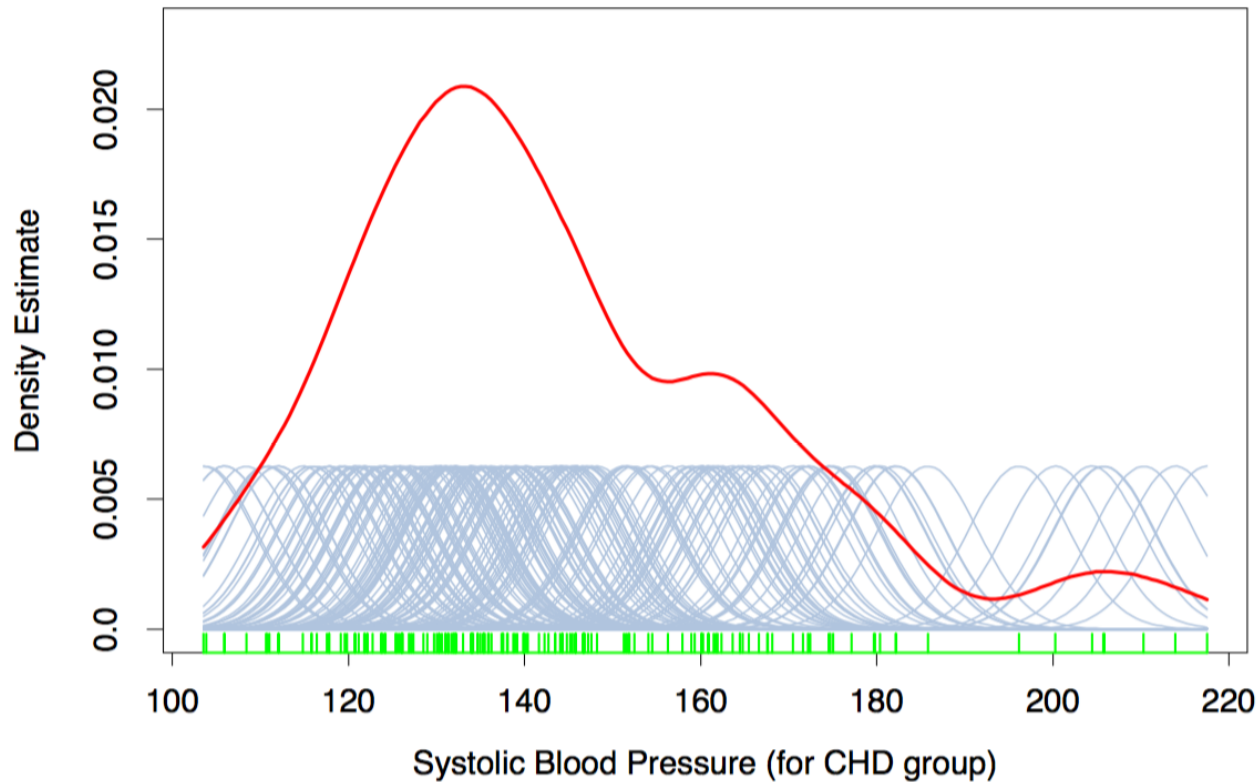




# Kernel Density Estimation

---

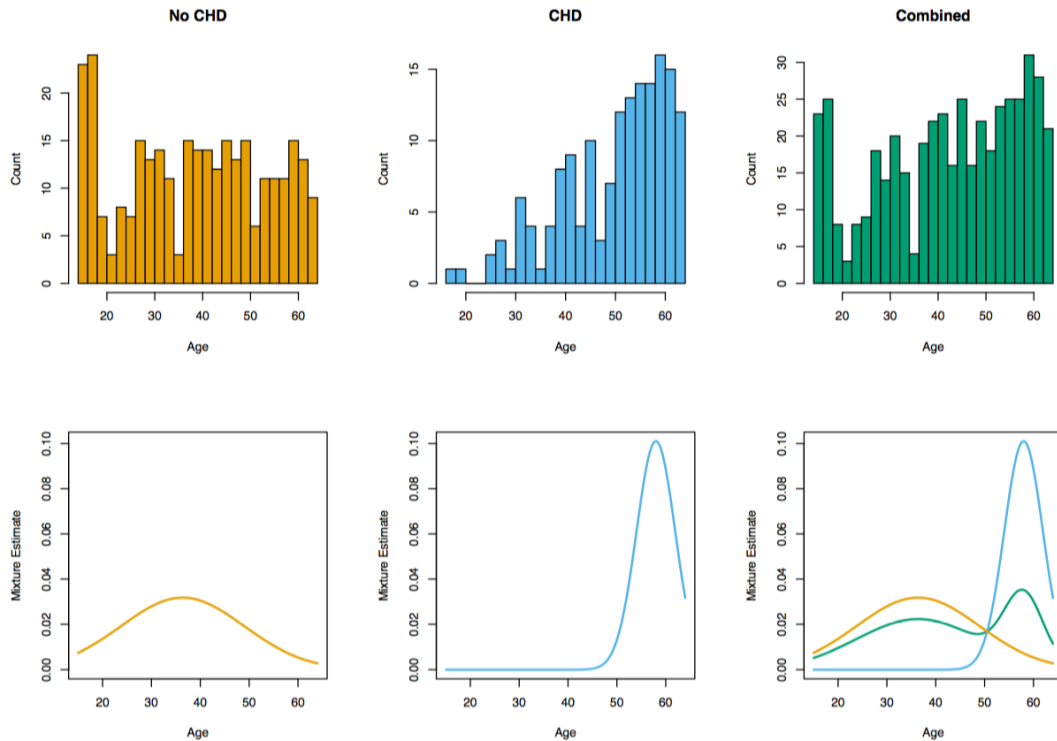
# Kernel Density Estimation



$$f(x) = \sum_{m=1}^M \alpha_m \phi(x; \mu_m, \Sigma_m)$$

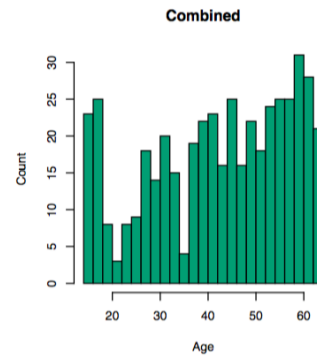
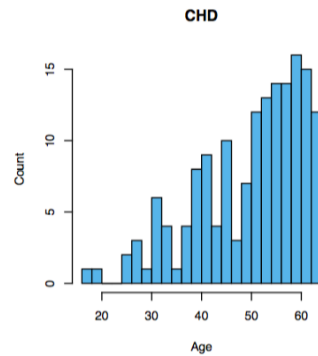
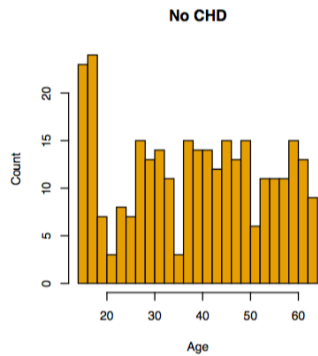
A very “lazy” GMM

# Kernel Density Estimation



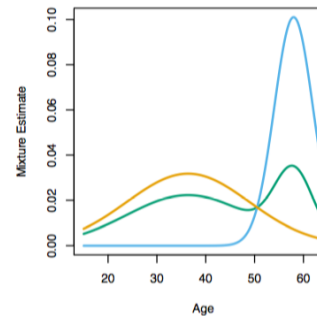
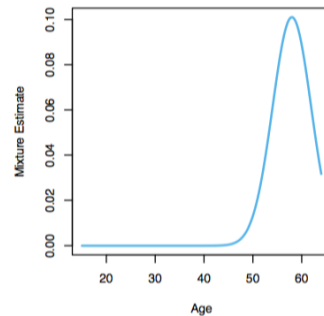
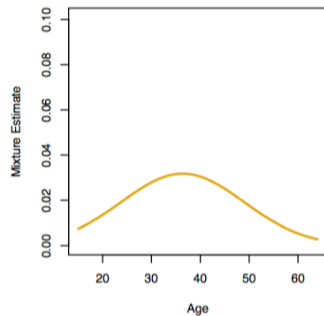
$$f(x) = \sum_{m=1}^M \alpha_m \phi(x; \mu_m, \Sigma_m)$$

# Kernel Density Estimation



What is the Bayes optimal classification rule?

Can we leverage this to build a classifier?



$$f(x) = \sum_{m=1}^M \alpha_m \phi(x; \mu_m, \Sigma_m)$$

# Generative vs Discriminative

---