Principal Component Analysis

UNIVERSITY of WASHINGTON

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?

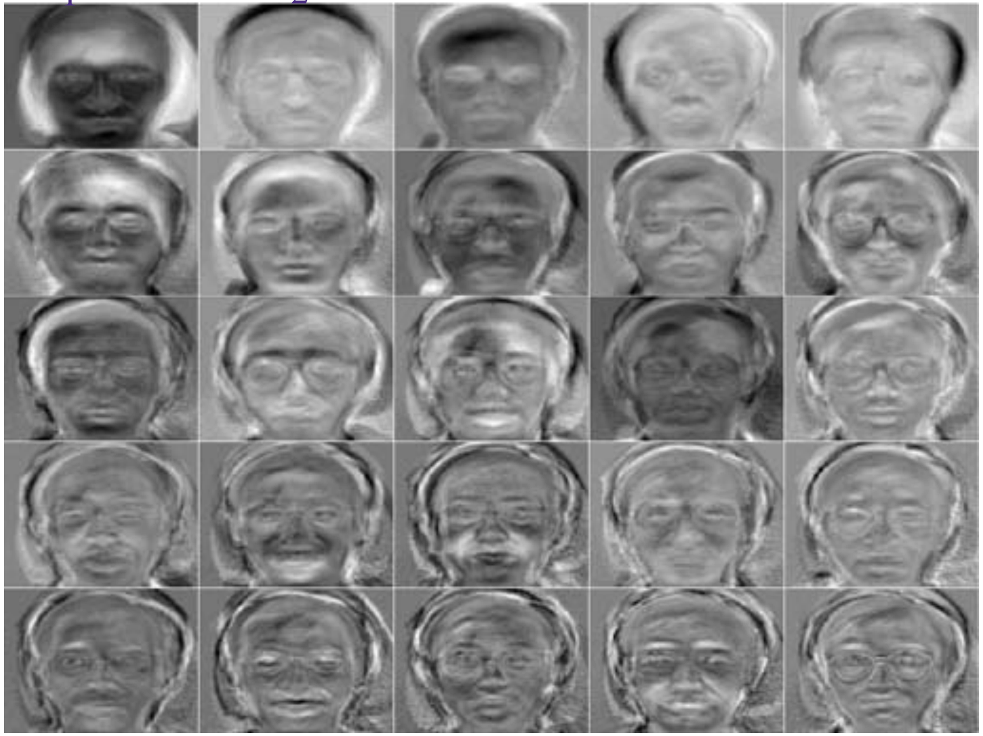
d=32x32pixels per image
n images
$d \times n$ real values to store the data

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:

$u_1 \in \mathbb{R}^d$ u_2



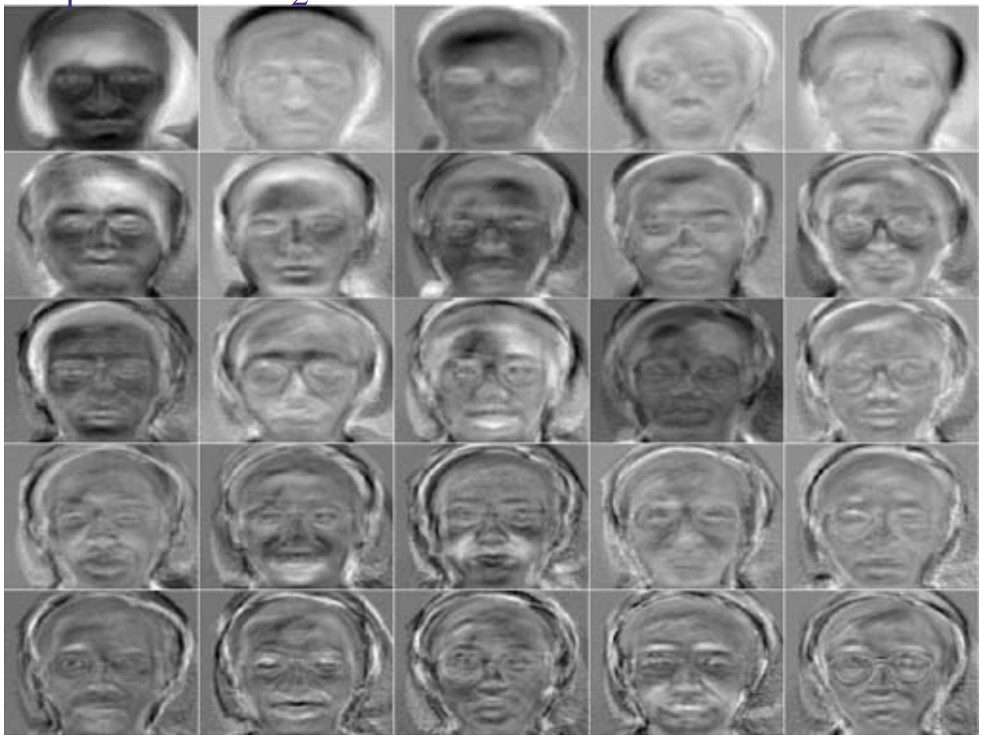
Principal component analysis finds a compact linear representation

- 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:

$u_1 \in \mathbb{R}^d$ u_2



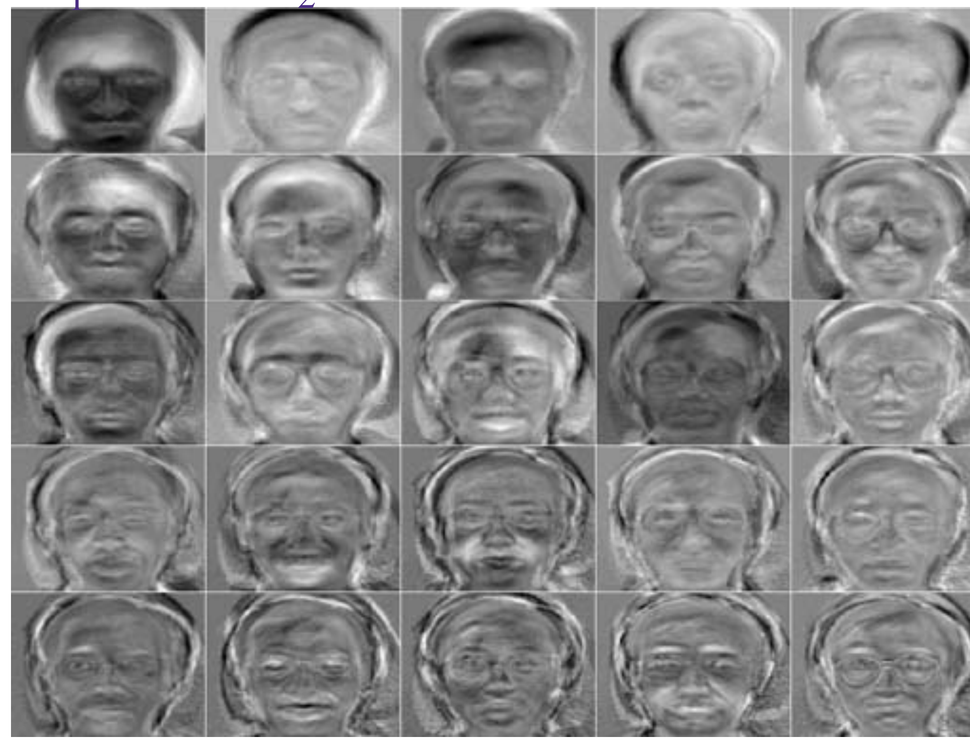
$$\approx a[1]u_1 + a[2]u_2 + \cdots + a[25]u_{25}$$

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:

$u_1 \in \mathbb{R}^d$ $\quad u_2$



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

- 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 $x_1, \ldots, x_n \in \mathbb{R}^d$, for $q \ll d$ find a compressed representation with $\lambda_1, \ldots, \lambda_n \in \mathbb{R}^q$ 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 μ, λ_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 μ, λ_i:

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

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

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

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 μ, λ_i:

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

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

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

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

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

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

PCA: a high-fidelity linear projection

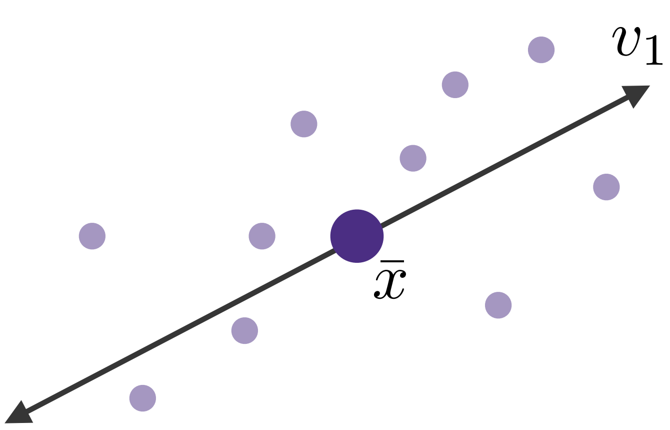
$$\min_{\mathbf{V}_q} \sum_{i=1}^{N} ||(x_i - \bar{x}) - \mathbf{V}_q \mathbf{V}_q^T (x_i - \bar{x})||^2.$$

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

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

Case when $q = 1$

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



PCA: a high-fidelity linear projection

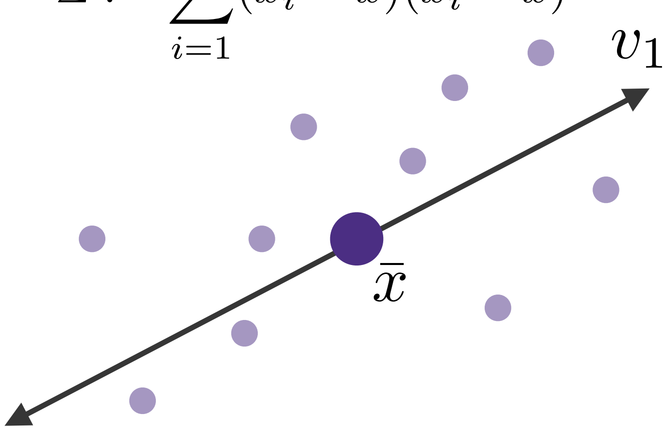
$$\min_{\mathbf{V}_q} \sum_{i=1}^{N} ||(x_i - \bar{x}) - \mathbf{V}_q \mathbf{V}_q^T (x_i - \bar{x})||^2.$$

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

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

Case when $q = 1$

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



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

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

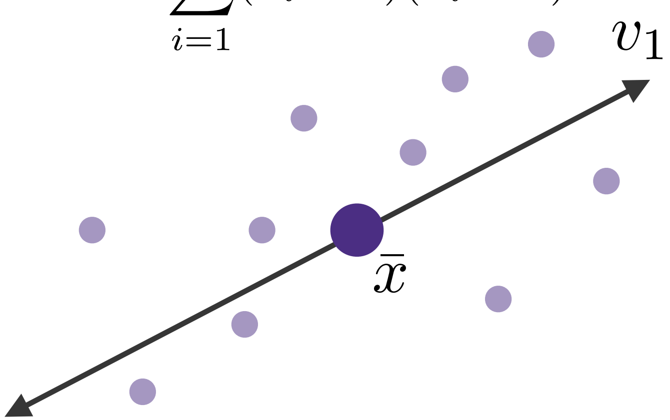
$$\widehat{x}_i := \bar{x} + \mathbf{V}_q \mathbf{V}_q^\top (x_i - \bar{x}) = \bar{x} + \sum_{j=1}^{q} v_j \langle v_j, x_i - \bar{x} \rangle$$

General $q \geq 1$

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

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

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



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

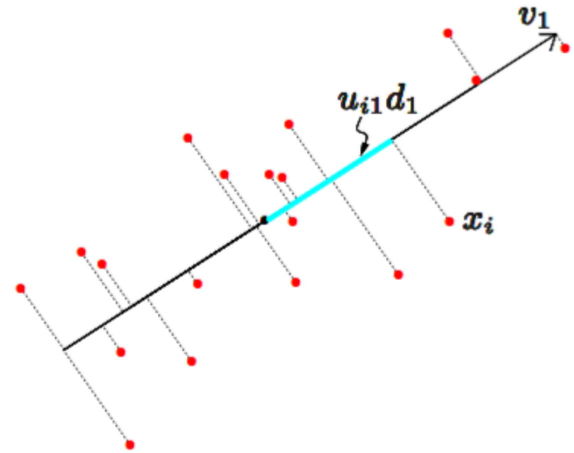
PCA: a high-fidelity linear projection

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

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



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

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

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

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

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

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

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

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

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

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

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

$\mathbf{V}$ 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 $\text{diag}(\mathbf{S})$

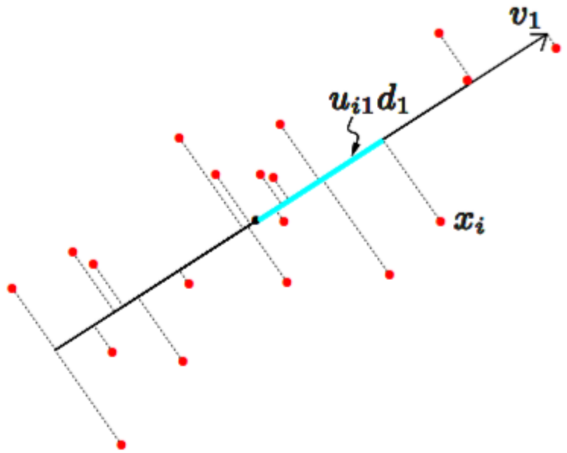
Linear projections

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

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



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

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

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

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

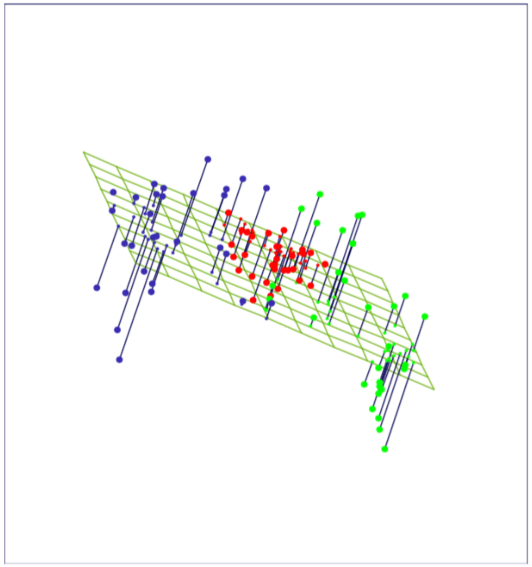
$$(\mathbf{X} - \mathbf{1}\bar{x}^T)\mathbf{V}_q = \mathbf{U}_q \text{diag}(d_1, \ldots, d_q) \qquad \mathbf{U}_q^T \mathbf{U}_q = I_q$$

Singular Value Decomposition defined as

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

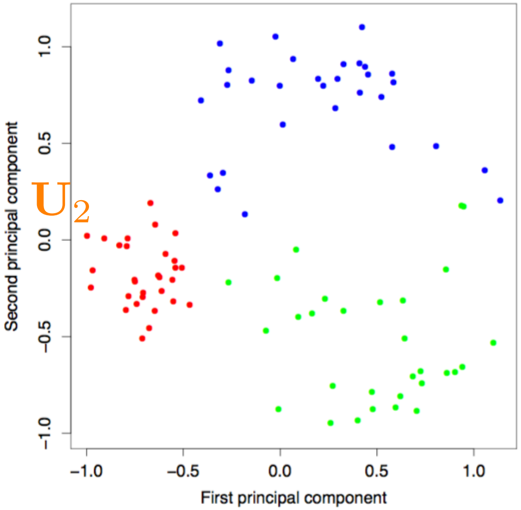
Dimensionality reduction

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



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

$\mathbf{U}_2$



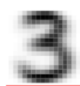
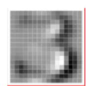
$\mathbf{U}_1$

Dimensionality reduction

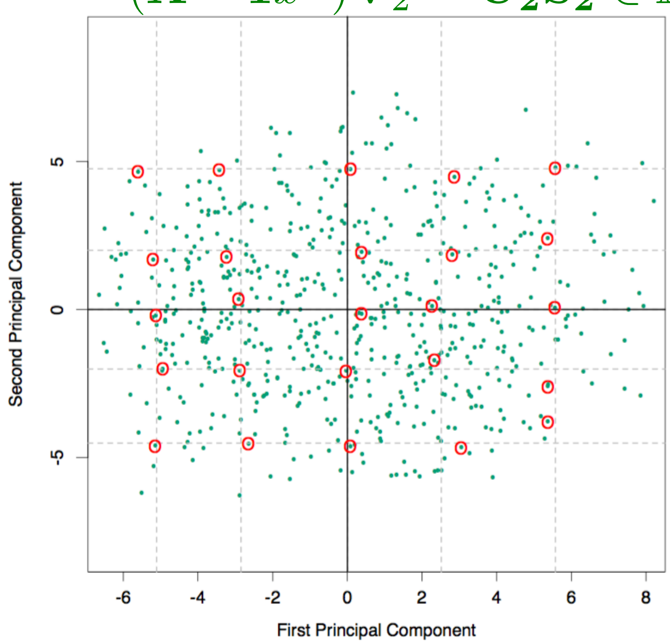
$\mathbf{V}_q$ are the first q eigenvectors of Σ and *SVD* $\mathbf{X} - \mathbf{1}\bar{x}^T = \mathbf{USV}^T$

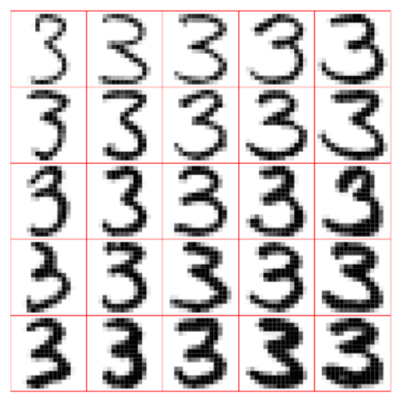
Handwritten 3's, 16x16 pixel image so that $x_i \in \mathbb{R}^{256}$

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

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

$(\mathbf{X} - \mathbf{1}\bar{x}^T)\mathbf{V}_2 = \mathbf{U_2 S_2} \in \mathbb{R}^{n \times 2}$





$\text{diag}(\mathbf{S})$

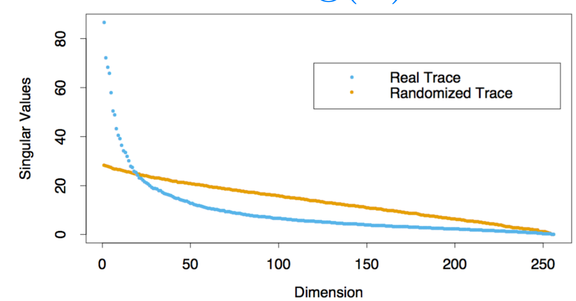
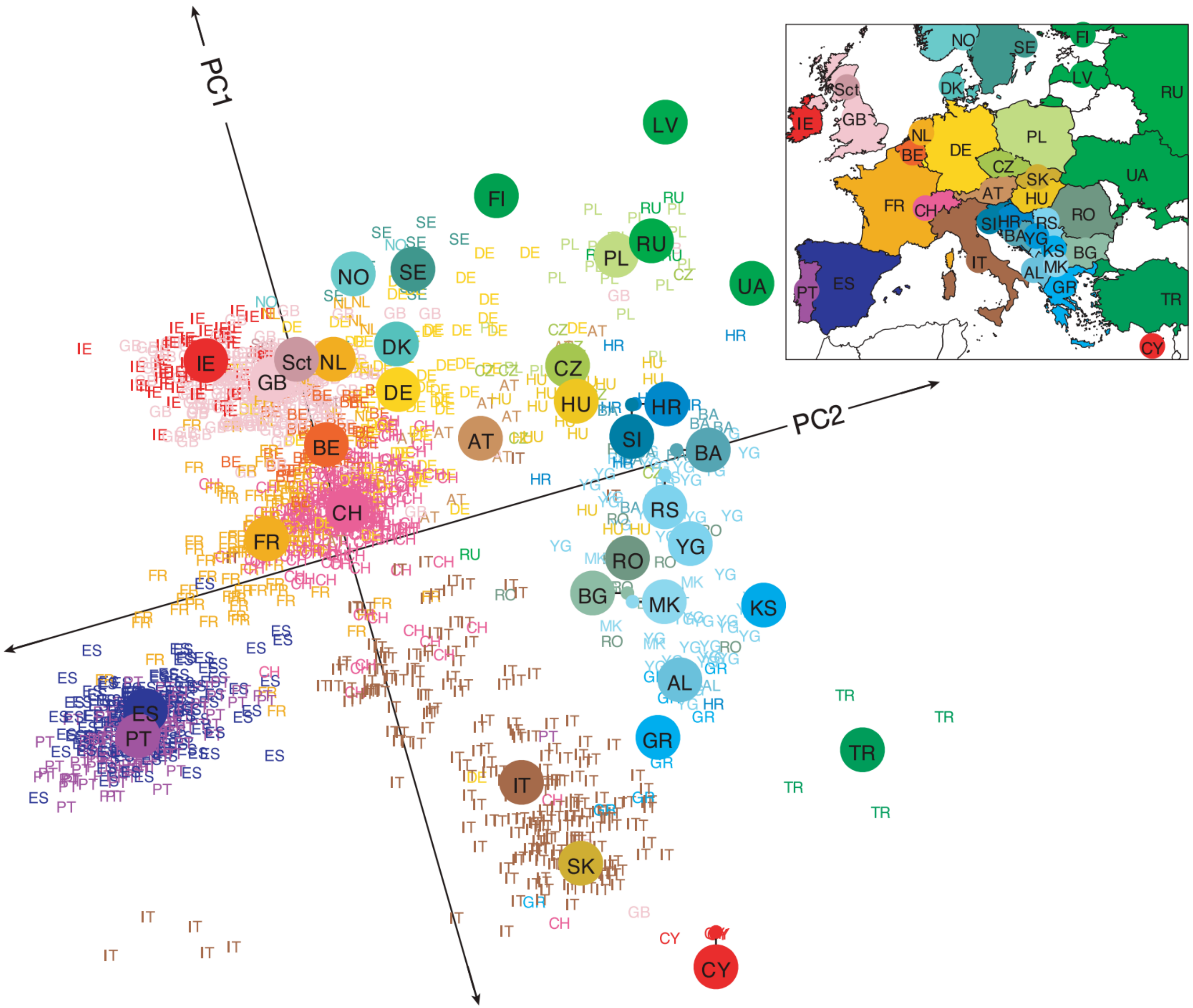


FIGURE 14.24. *The 256 singular values for the digitized threes, compared to those for a randomized version of the data (each column of* **X** *was scrambled).*

Dimensionality reduction



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

Kernel PCA

$\mathbf{V}_q$ are the first q eigenvectors of Σ and *SVD* $\mathbf{X} - \mathbf{1}\bar{x}^T = \mathbf{USV}^T$

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

$$\mathbf{JX} = \mathbf{X} - \mathbf{1}\bar{x}^T = \mathbf{USV}^T \qquad \mathbf{J} = I - \mathbf{1}\mathbf{1}^T/n$$

$$(\mathbf{JX})(\mathbf{JX})^T =$$

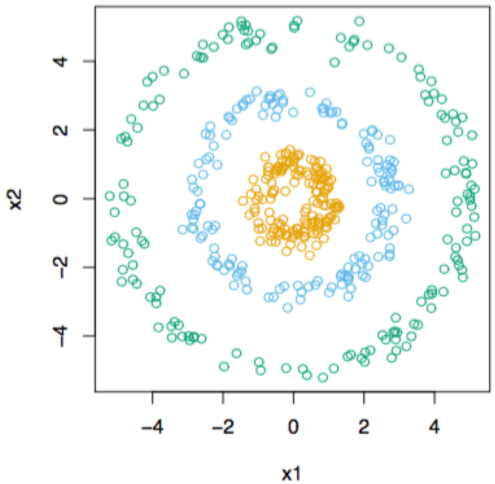
Kernel PCA

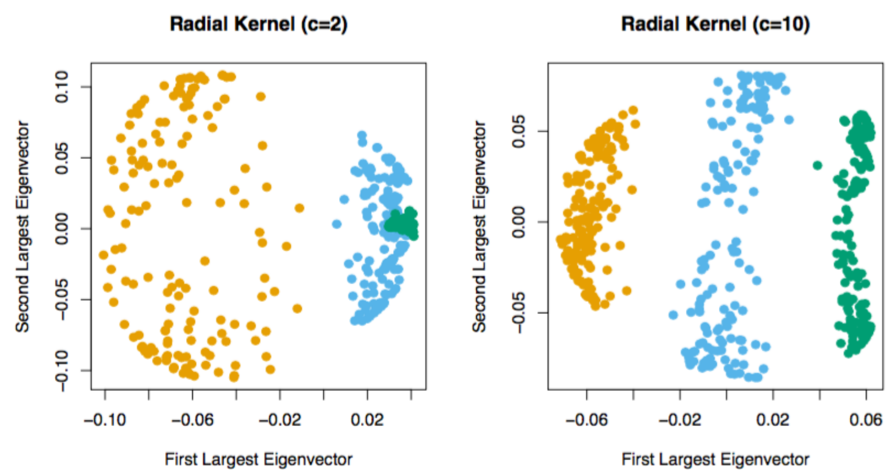
$\mathbf{V}_q$ are the first q eigenvectors of Σ and *SVD* $\mathbf{X} - \mathbf{1}\bar{x}^T = \mathbf{USV}^T$

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

$$\mathbf{JX} = \mathbf{X} - \mathbf{1}\bar{x}^T = \mathbf{USV}^T \qquad \mathbf{J} = I - \mathbf{1}\mathbf{1}^T/n$$

$$(\mathbf{JX})(\mathbf{JX})^T = \mathbf{US}^2\mathbf{U}^T$$





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

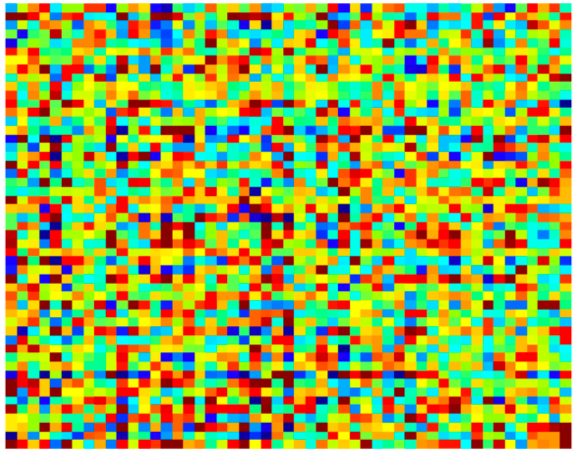
n movies, m users, |*S*| ratings

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

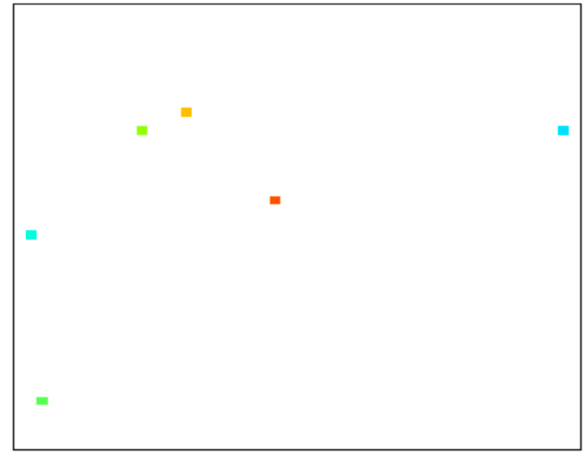
Example: 2000 × 2000 rank-8 random matrix

For illustration, we zoom in to a 50x50 submatrix

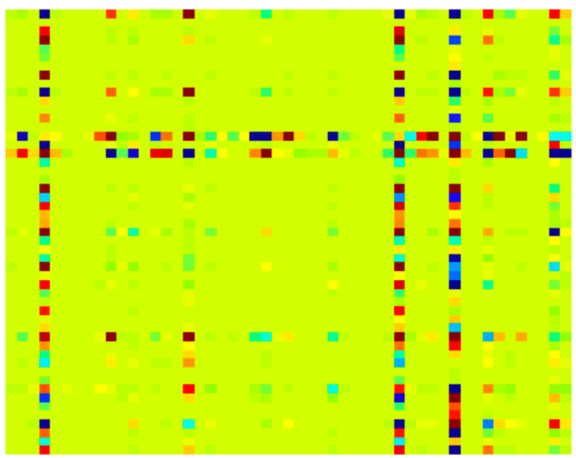
low-rank matrix $\mathbf{X}$



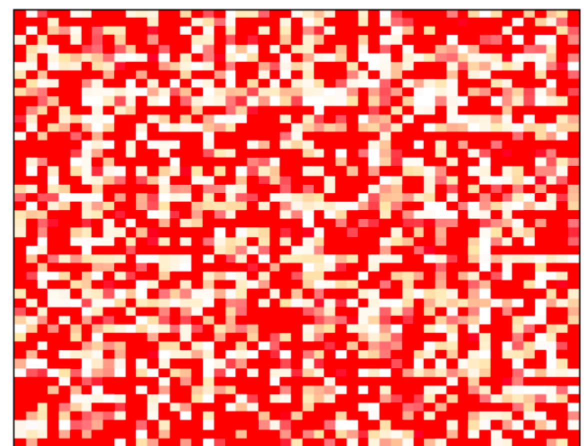
sampled matrix



Gradient descent output $\mathbf{UA}$



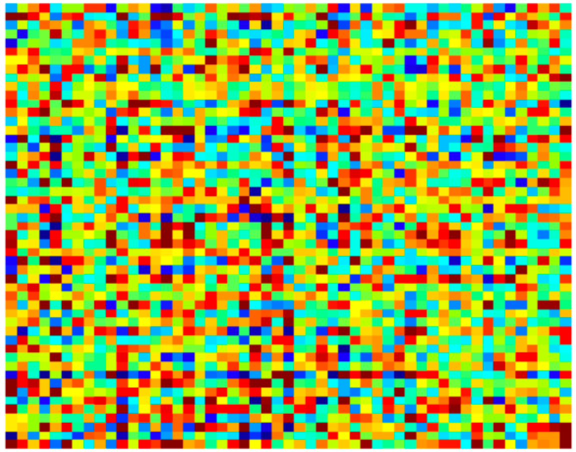
squared error $(\mathbf{X}_{ji} - (\mathbf{UA})_{ji})^2$



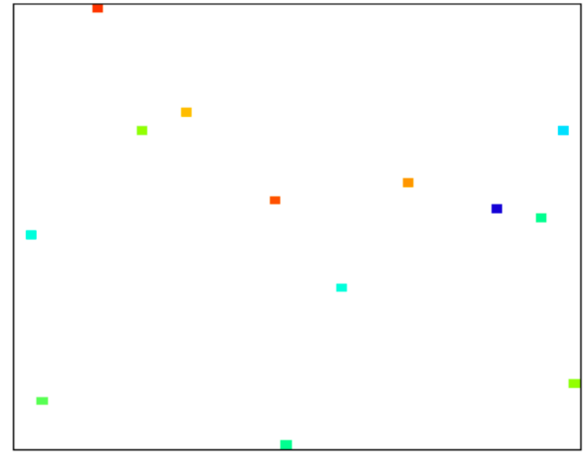
0.25% sampled

Example: 2000×2000 rank-8 random matrix

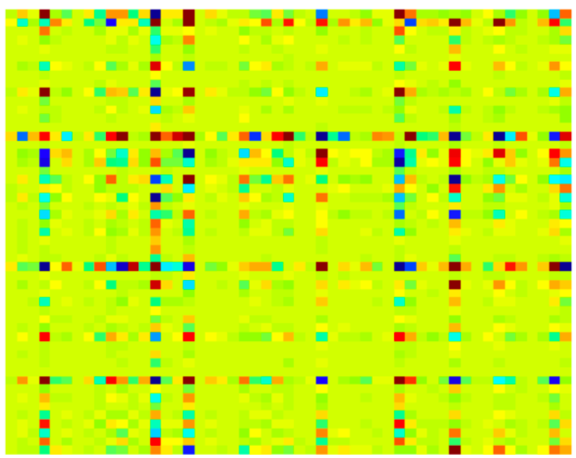
low-rank matrix $\mathbf{X}$



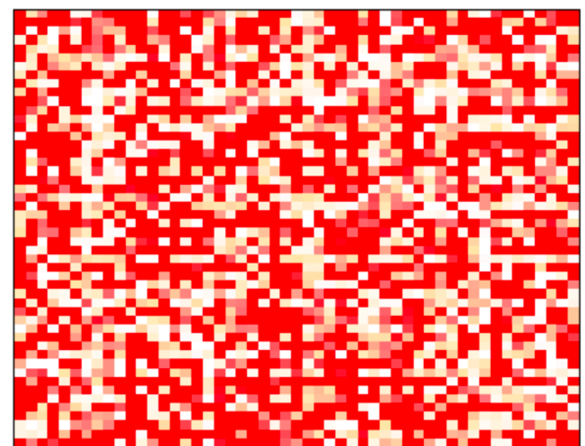
sampled matrix



Gradient descent output $\mathbf{UA}$



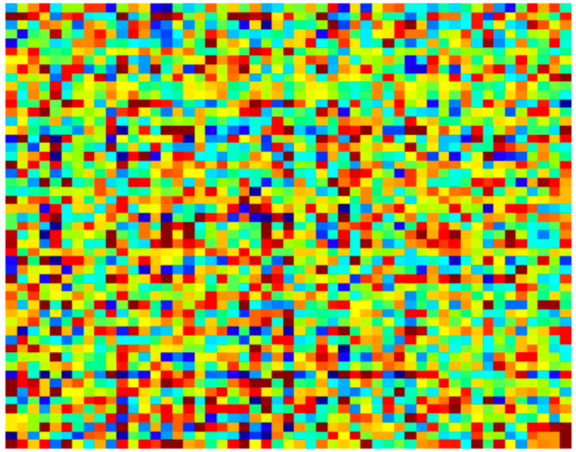
squared error $(\mathbf{X}_{ji} - (\mathbf{UA})_{ji})^2$



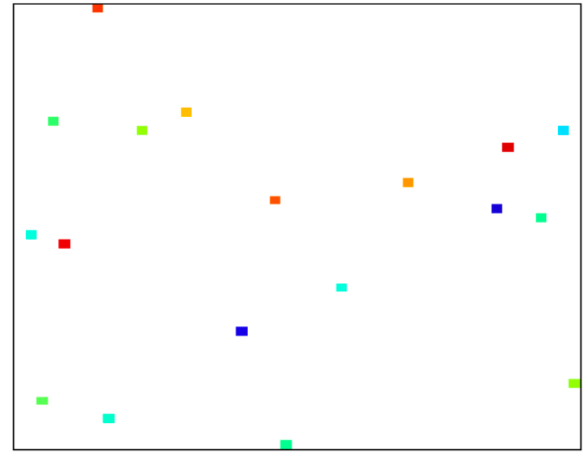
0.50% sampled

Example: 2000×2000 rank-8 random matrix

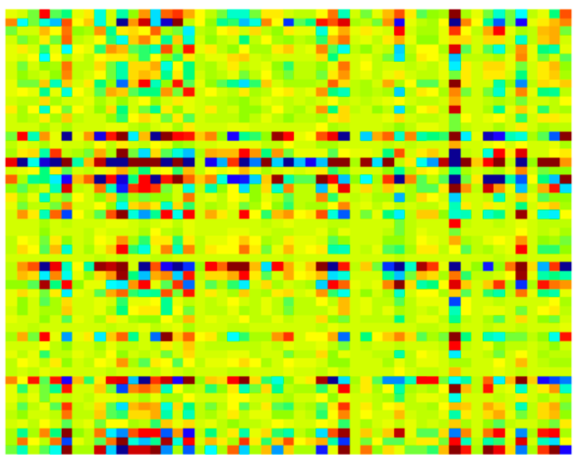
low-rank matrix $\mathbf{X}$



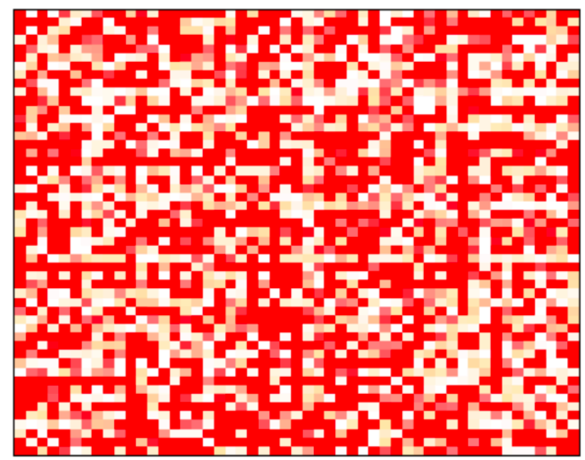
sampled matrix



Gradient descent output $\mathbf{UA}$

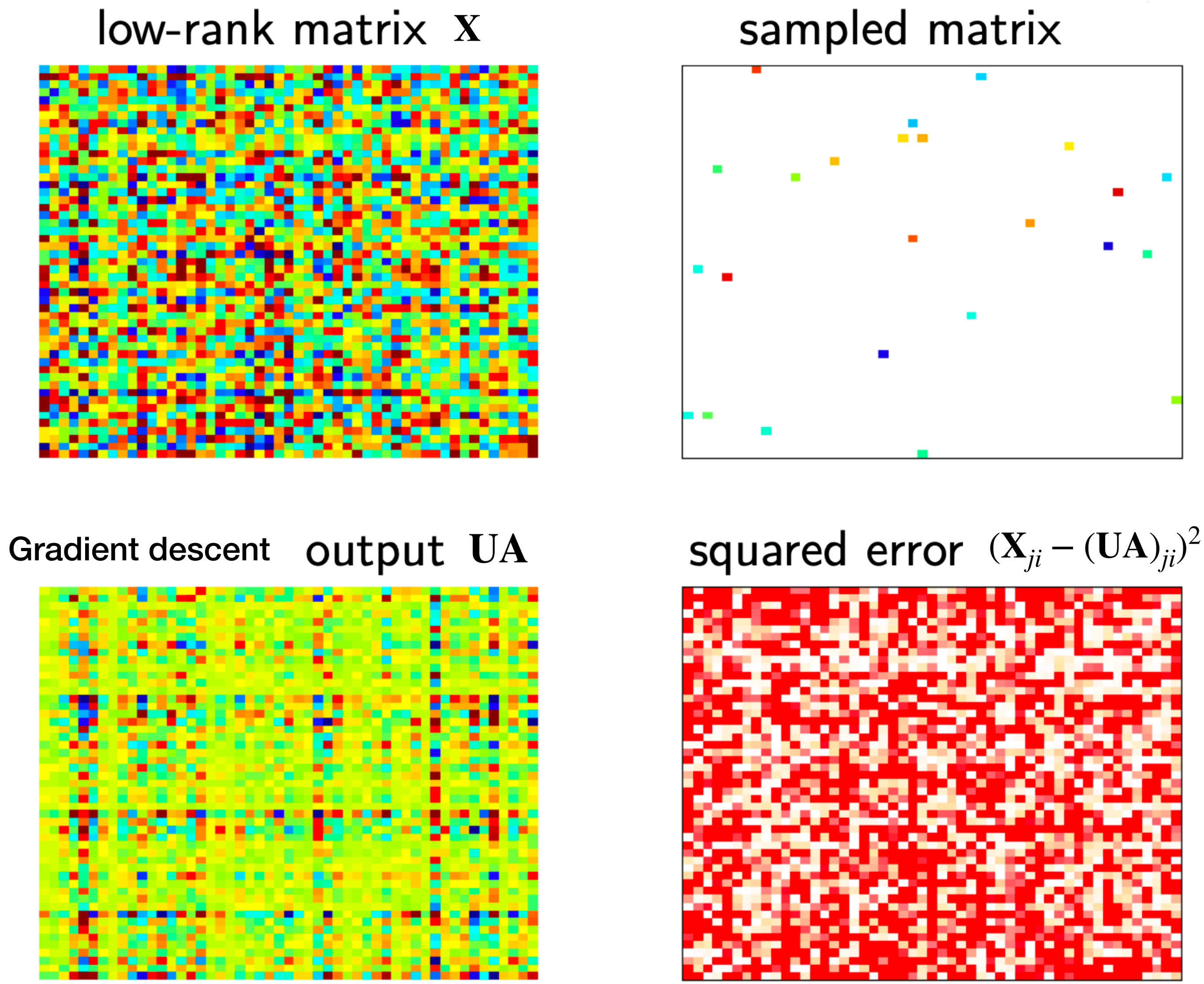


squared error $(\mathbf{X}_{ji} - (\mathbf{UA})_{ji})^2$



0.75% sampled

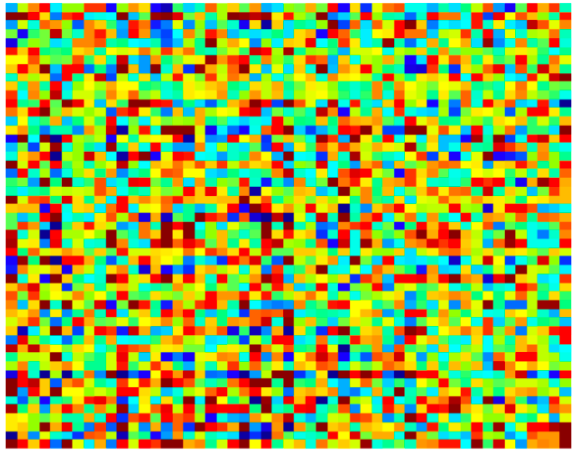
Example: 2000 × 2000 rank-8 random matrix



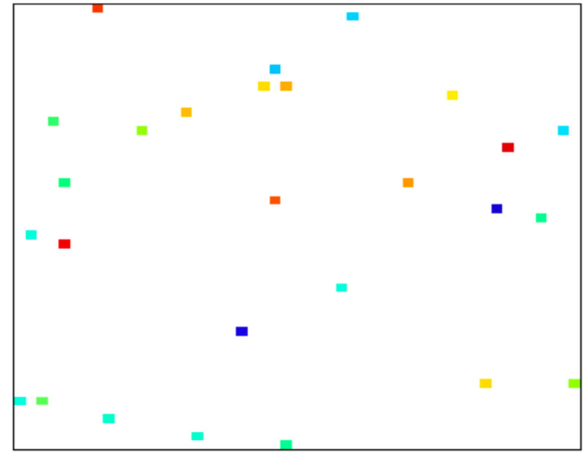
1.00% sampled

Example: 2000 × 2000 rank-8 random matrix

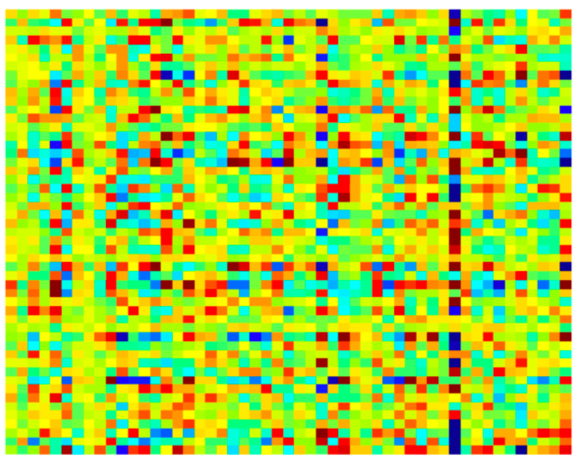
low-rank matrix $\mathbf{X}$



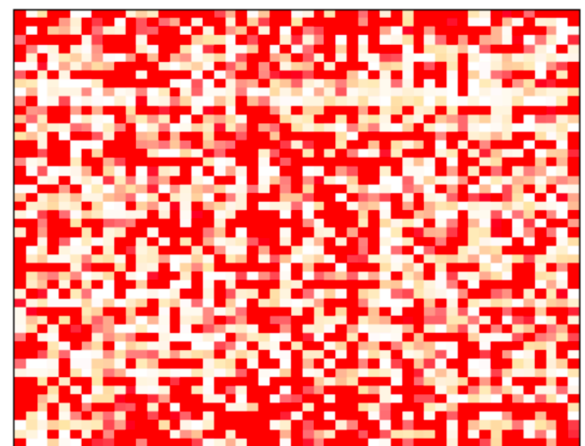
sampled matrix



Gradient descent output $\mathbf{UA}$

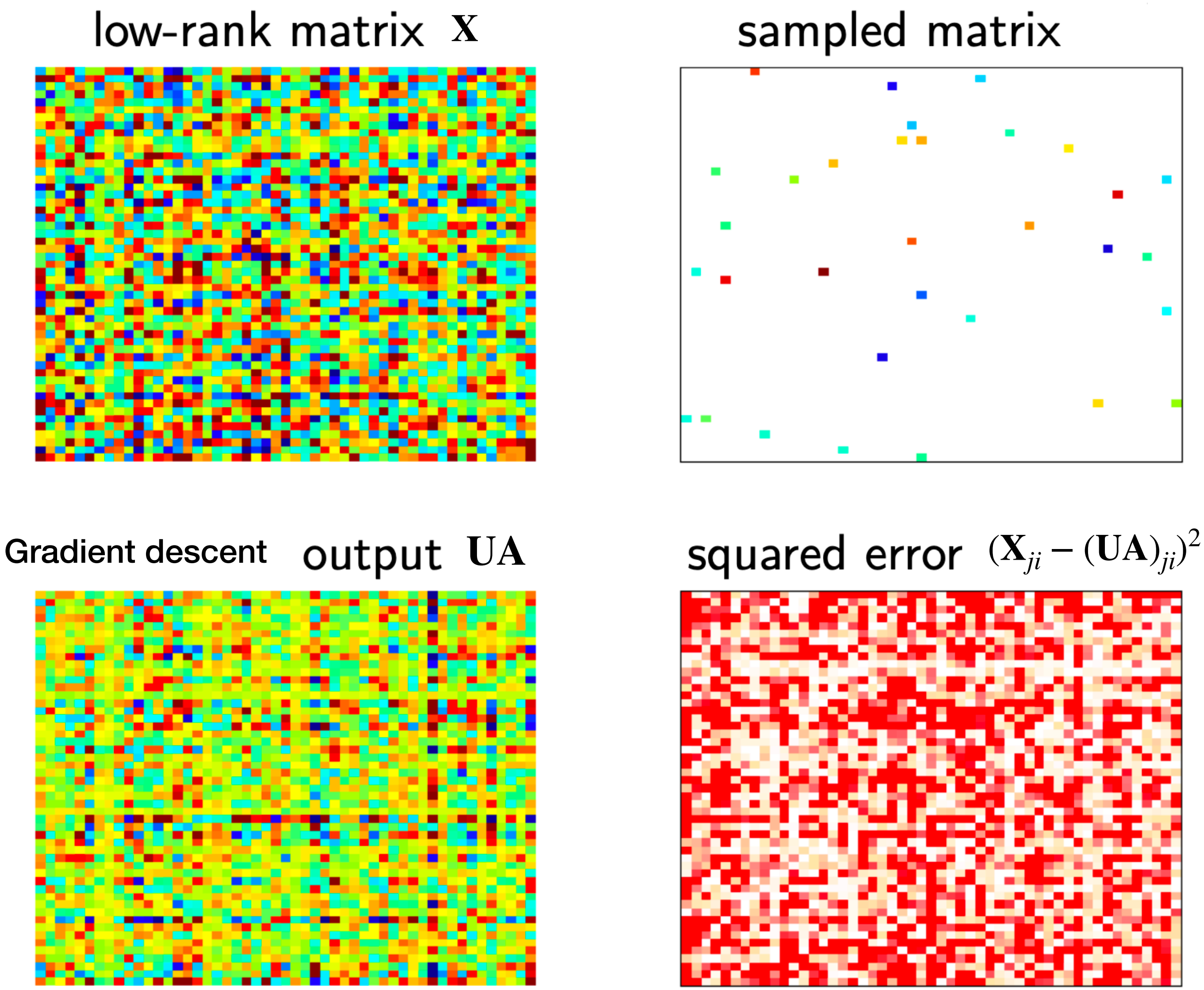


squared error $(\mathbf{X}_{ji} - (\mathbf{UA})_{ji})^2$



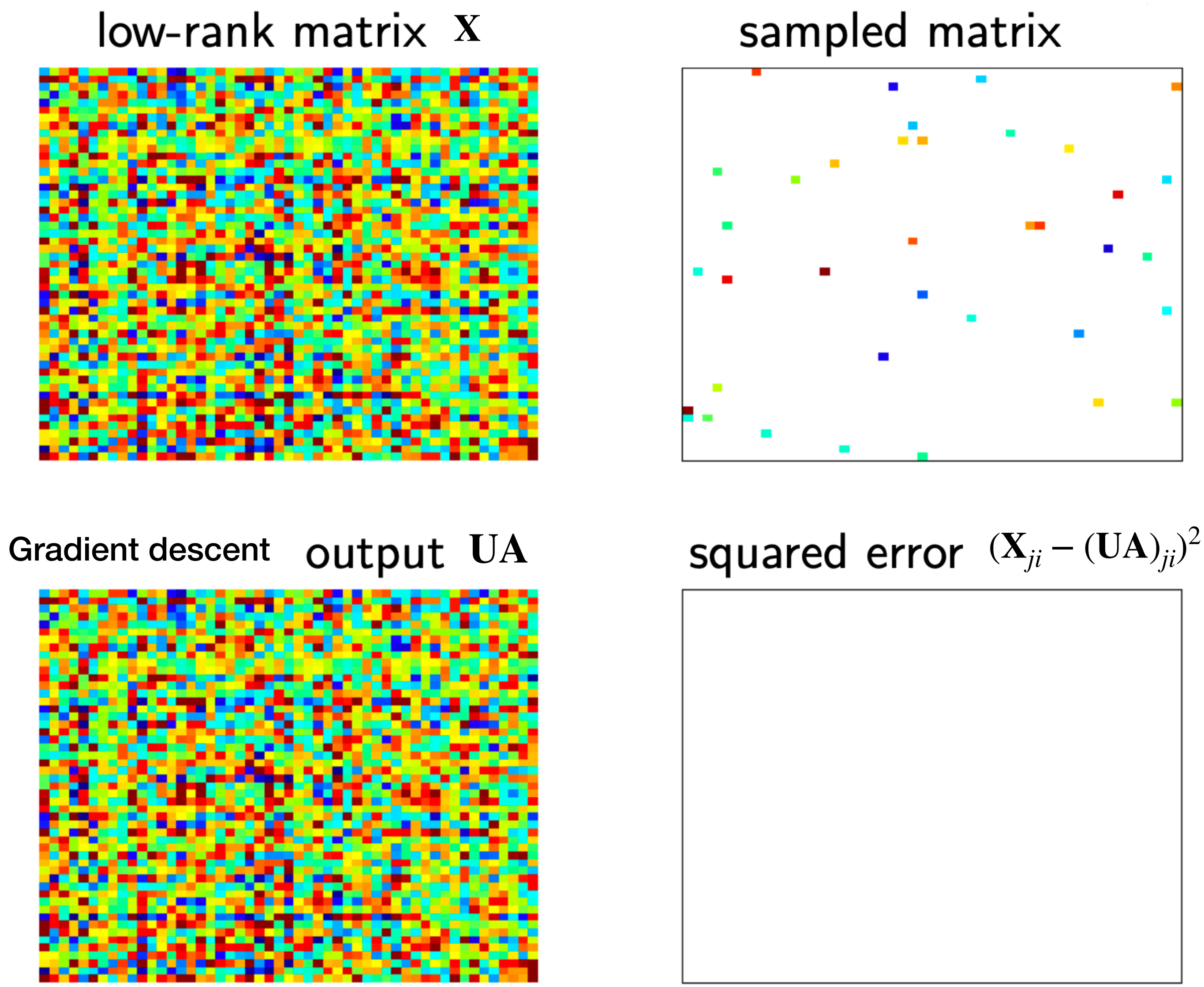
1.25% sampled

Example: 2000 × 2000 rank-8 random matrix



1.50% sampled

Example: 2000×2000 rank-8 random matrix



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 Σ

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

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

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

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 Lipshcitz mapping $f : \mathbb{R}^d \to \mathbb{R}^k$ 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

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

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 Lipshcitz mapping $f : \mathbb{R}^d \to \mathbb{R}^k$ such that for all $u, v \in Q$:*

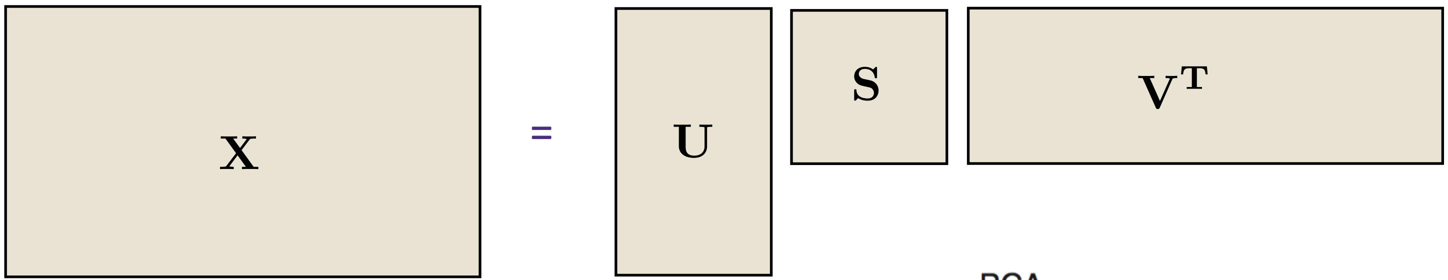
(independent of d)

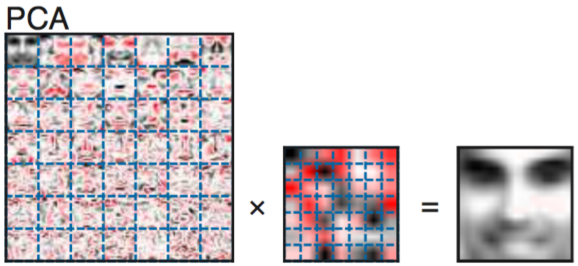
$$(1-\epsilon)\|u-v\|^2 \leq \|f(u) - f(v)\|^2 \leq (1+\epsilon)\|u-v\|^2$$

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,*

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

Other matrix factorizations



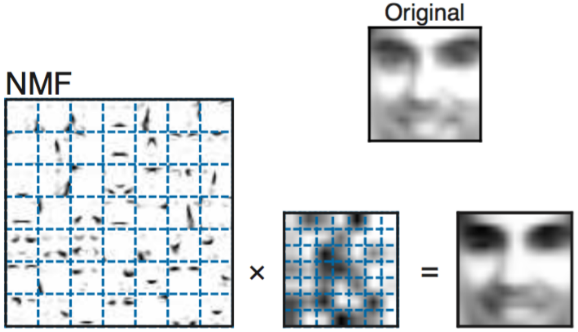


Singular value decomposition

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

Nonnegative matrix factorization (NMF)

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



Autoencoders

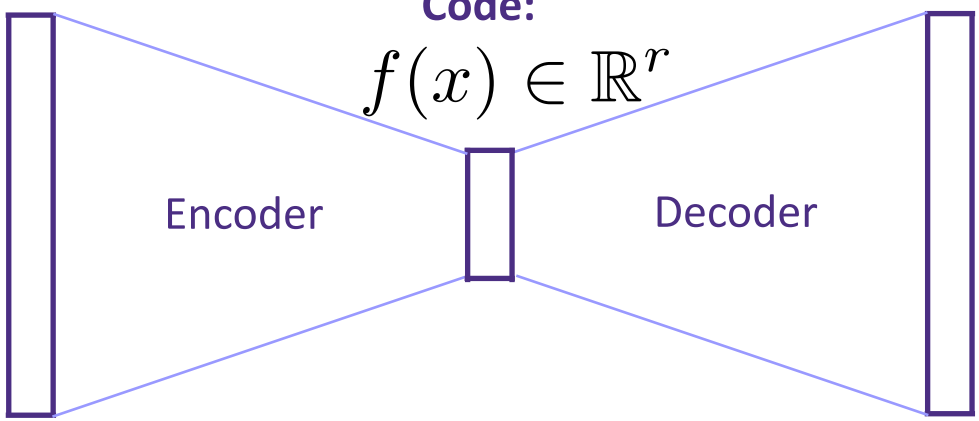
Find a low dimensional representation for your data by predicting your data

Code:

$$f(x) \in \mathbb{R}^r$$

Input:

$$x \in \mathbb{R}^d$$



Output:

$$\widehat{x} = g(f(x)) \in \mathbb{R}^d$$

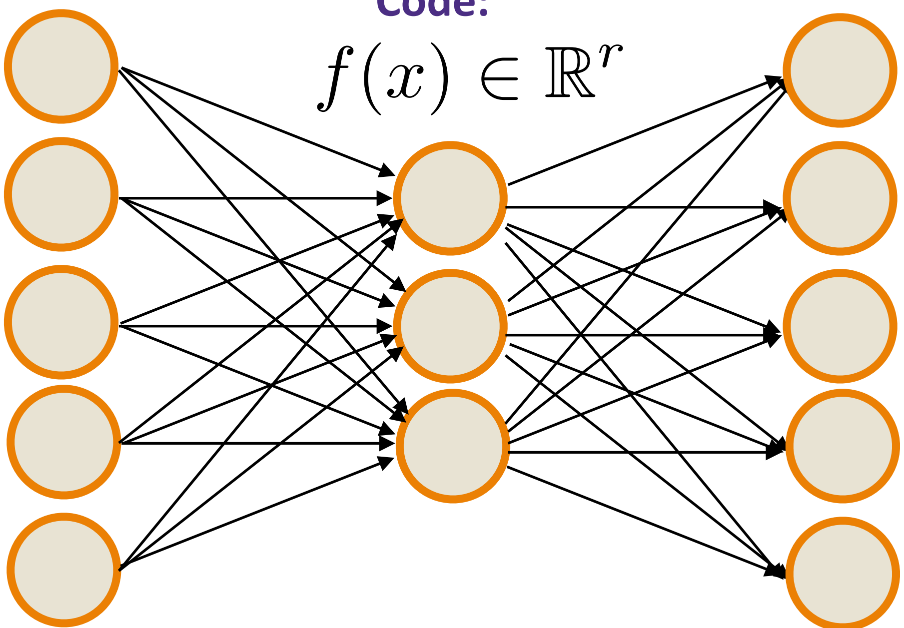
$$\underset{f,g}{\text{minimize}} \sum_{i=1}^{n} \|x_i - g(f(x_i))\|_2^2$$

Autoencoders

Input:
$x \in \mathbb{R}^d$

Code:
$f(x) \in \mathbb{R}^r$

Output:
$\widehat{x} = g(f(x)) \in \mathbb{R}^d$



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

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

Ridge Regression revisited

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

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

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

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

Ridge Regression revisited

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

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

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

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

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

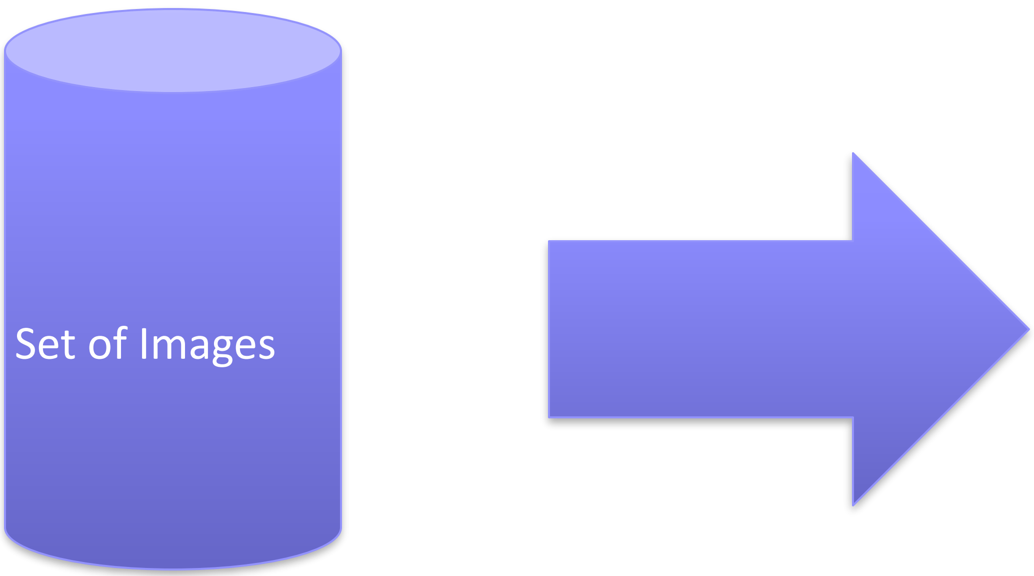
$$\mathbf{U} = [u_1, \dots, u_d]$$

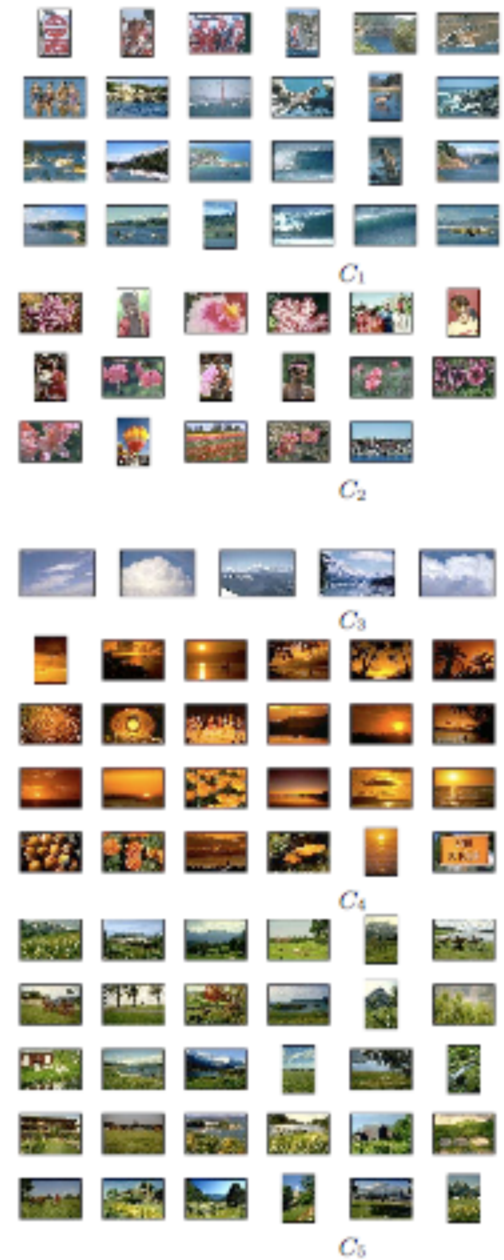
$$\mathbf{S} = \mathrm{diag}(s_1, \dots, s_d)$$

Clustering
K-means

UNIVERSITY *of* WASHINGTON

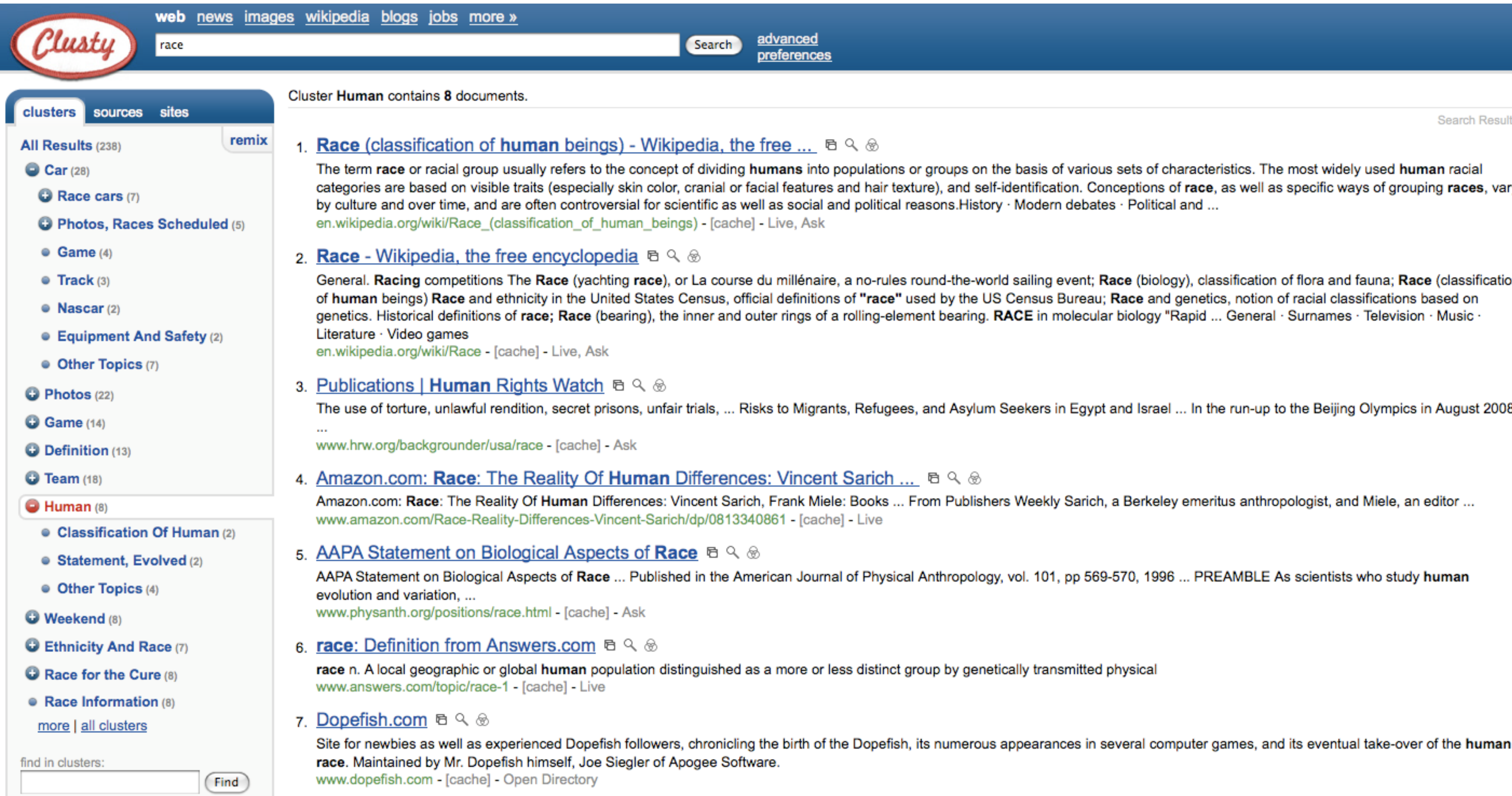
Clustering images



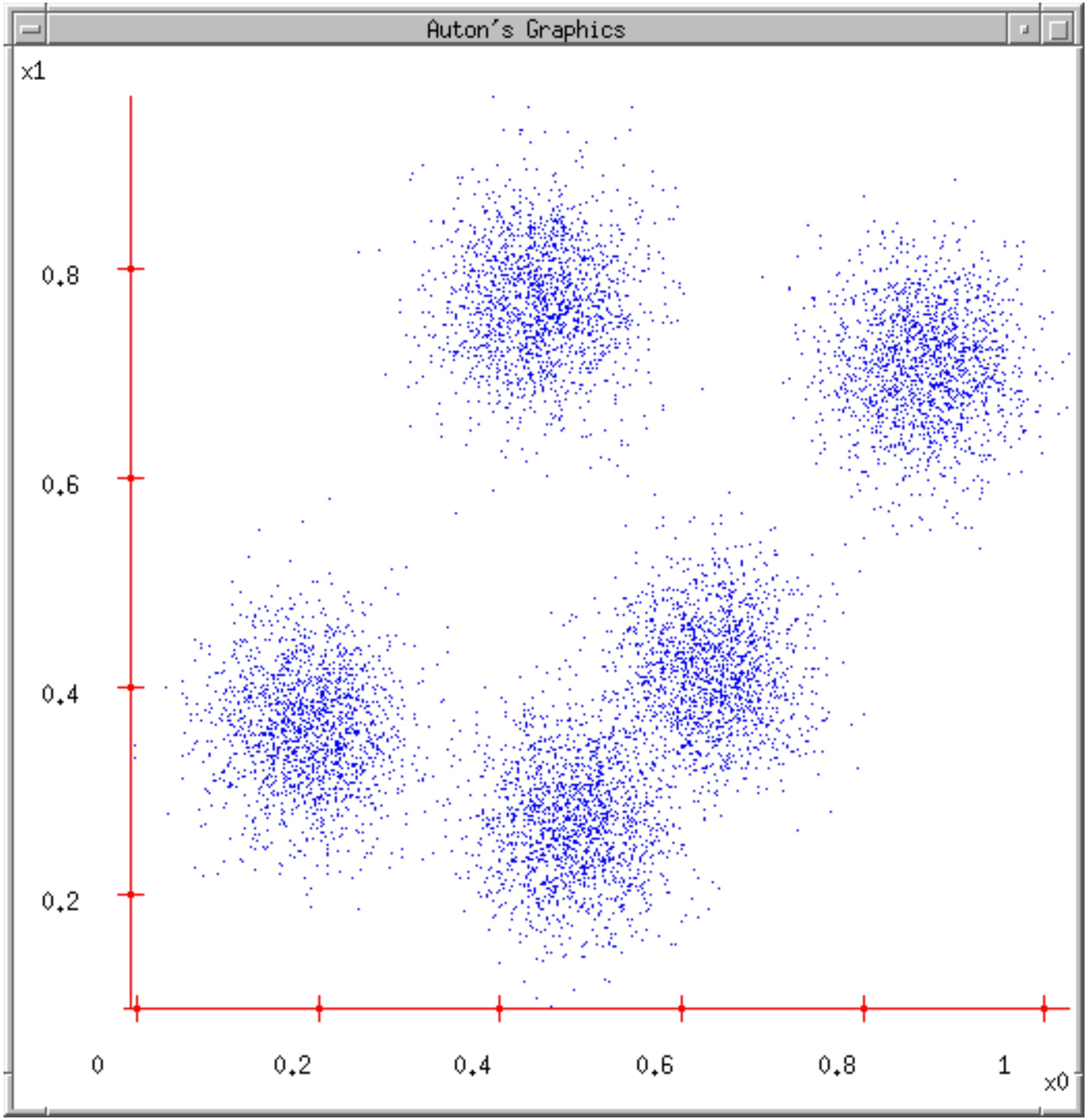


[Goldberger et al.]

Clustering web search results

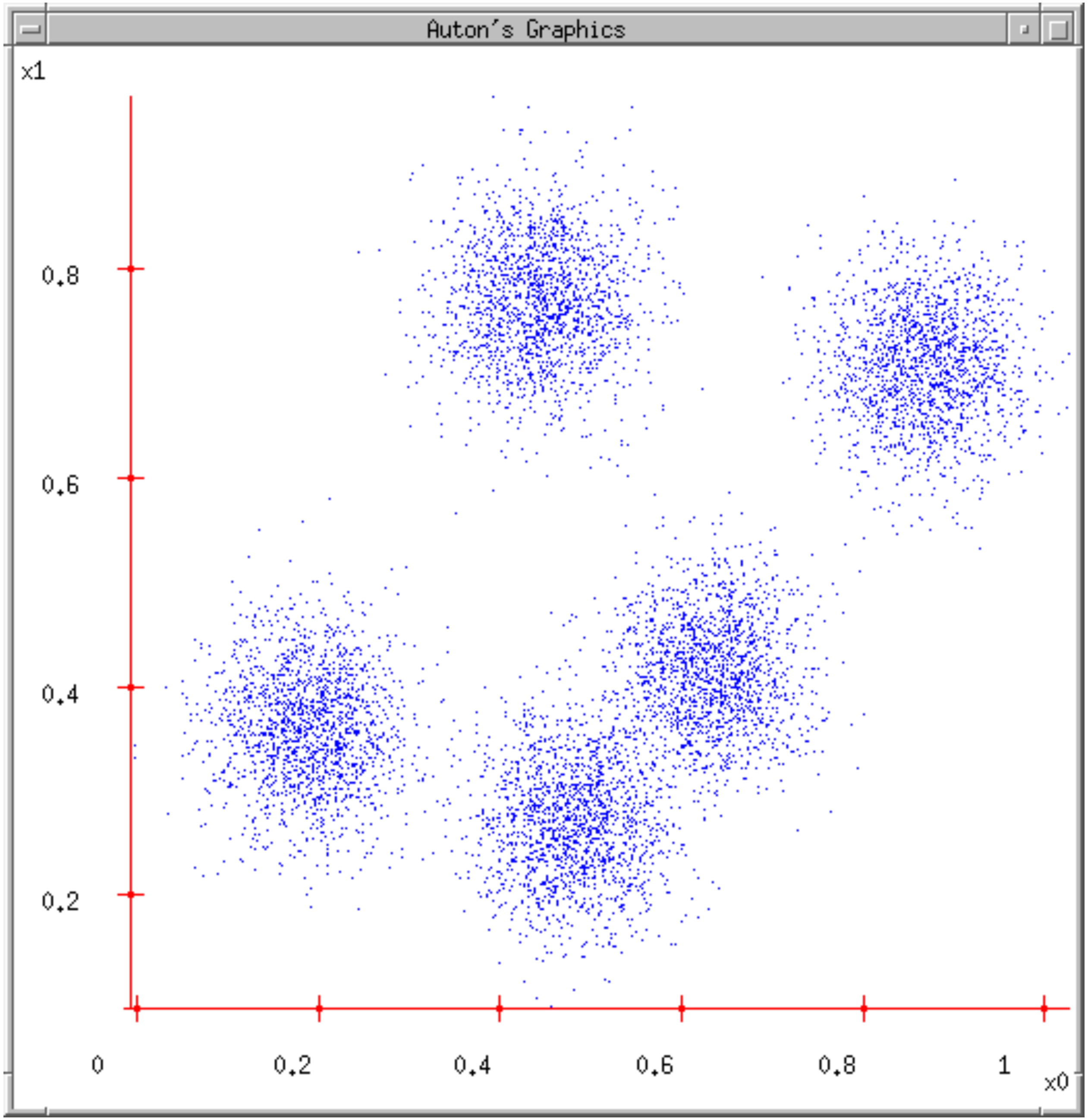


Some Data



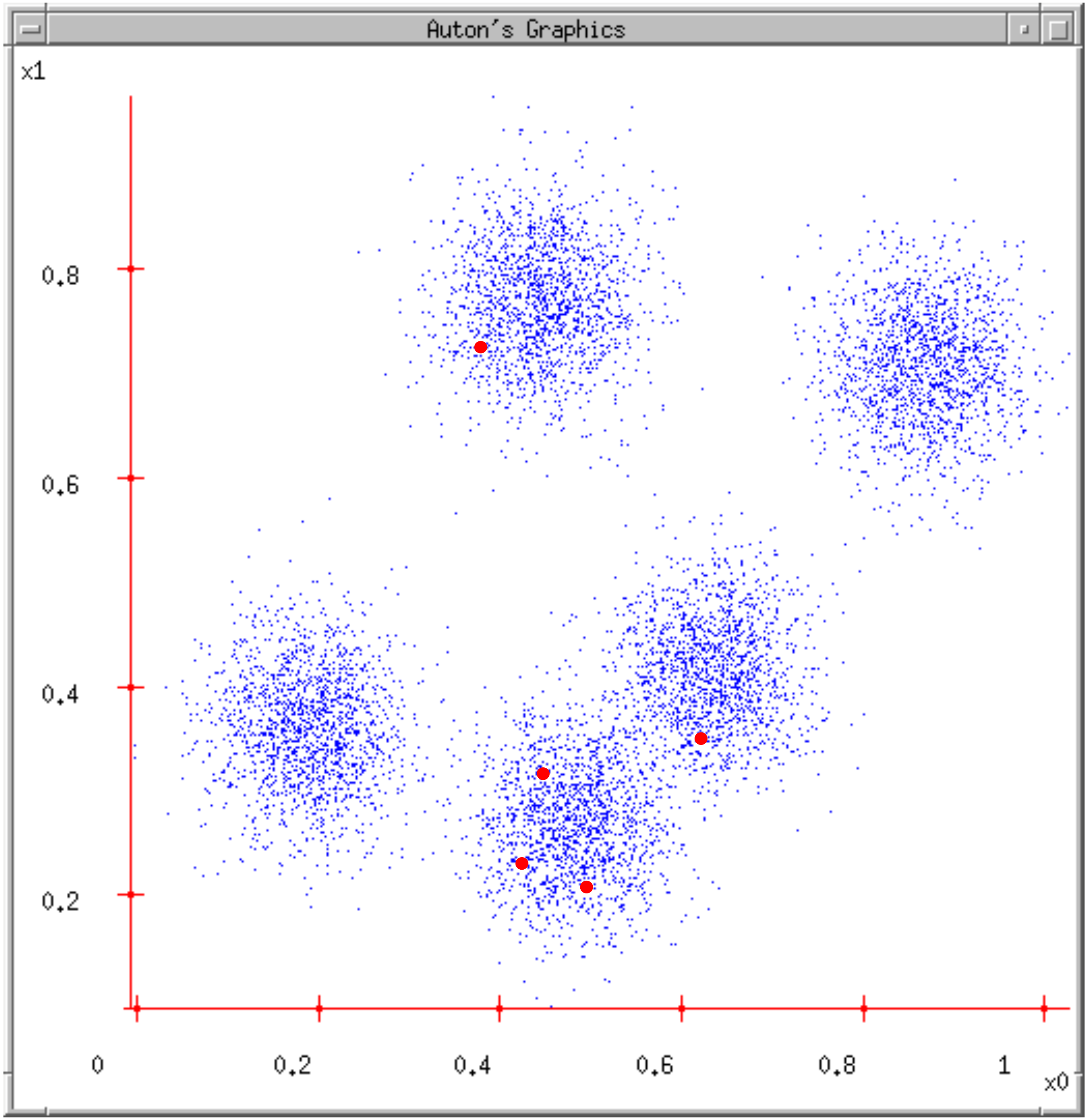
K-means

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



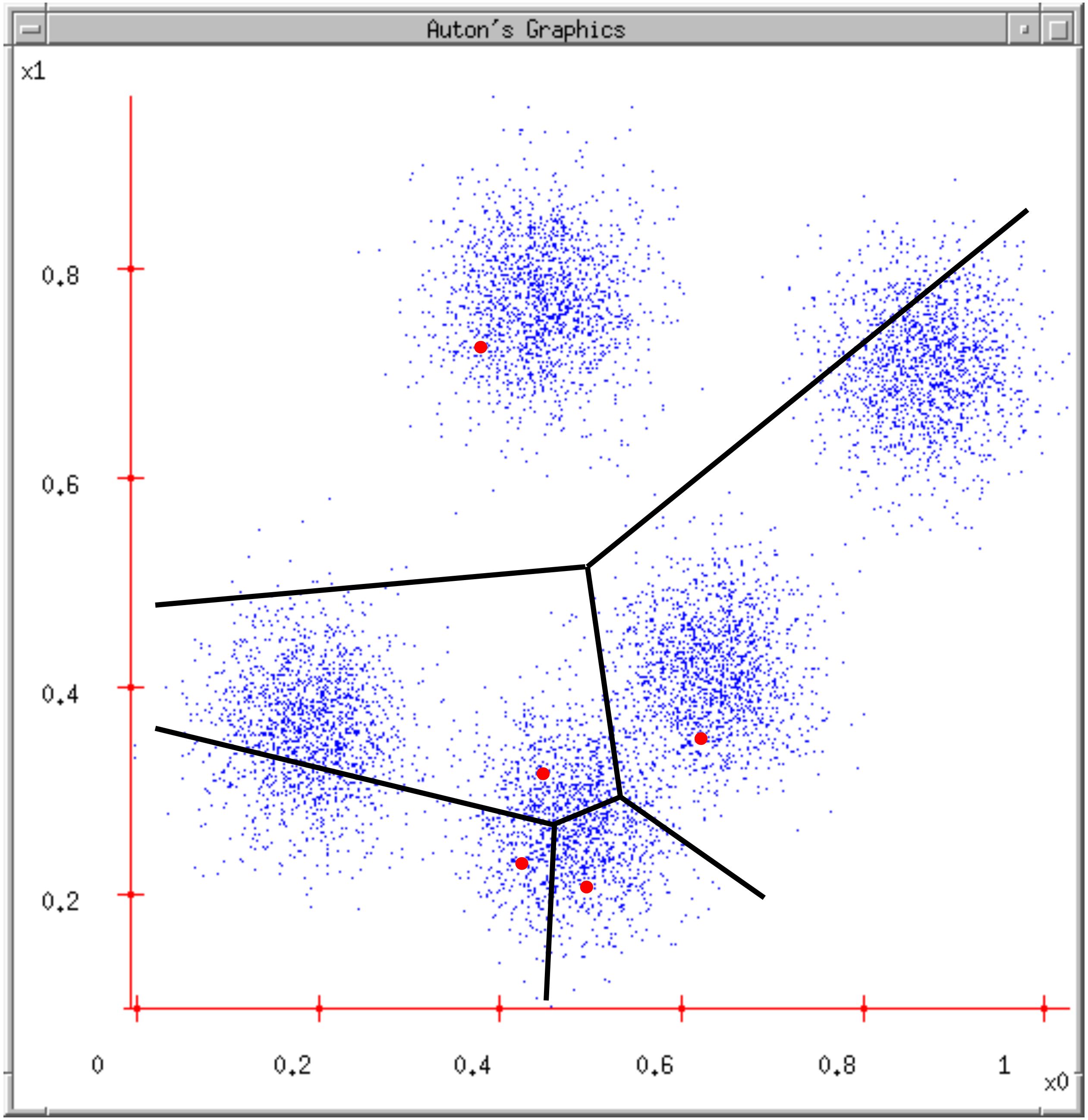
K-means

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



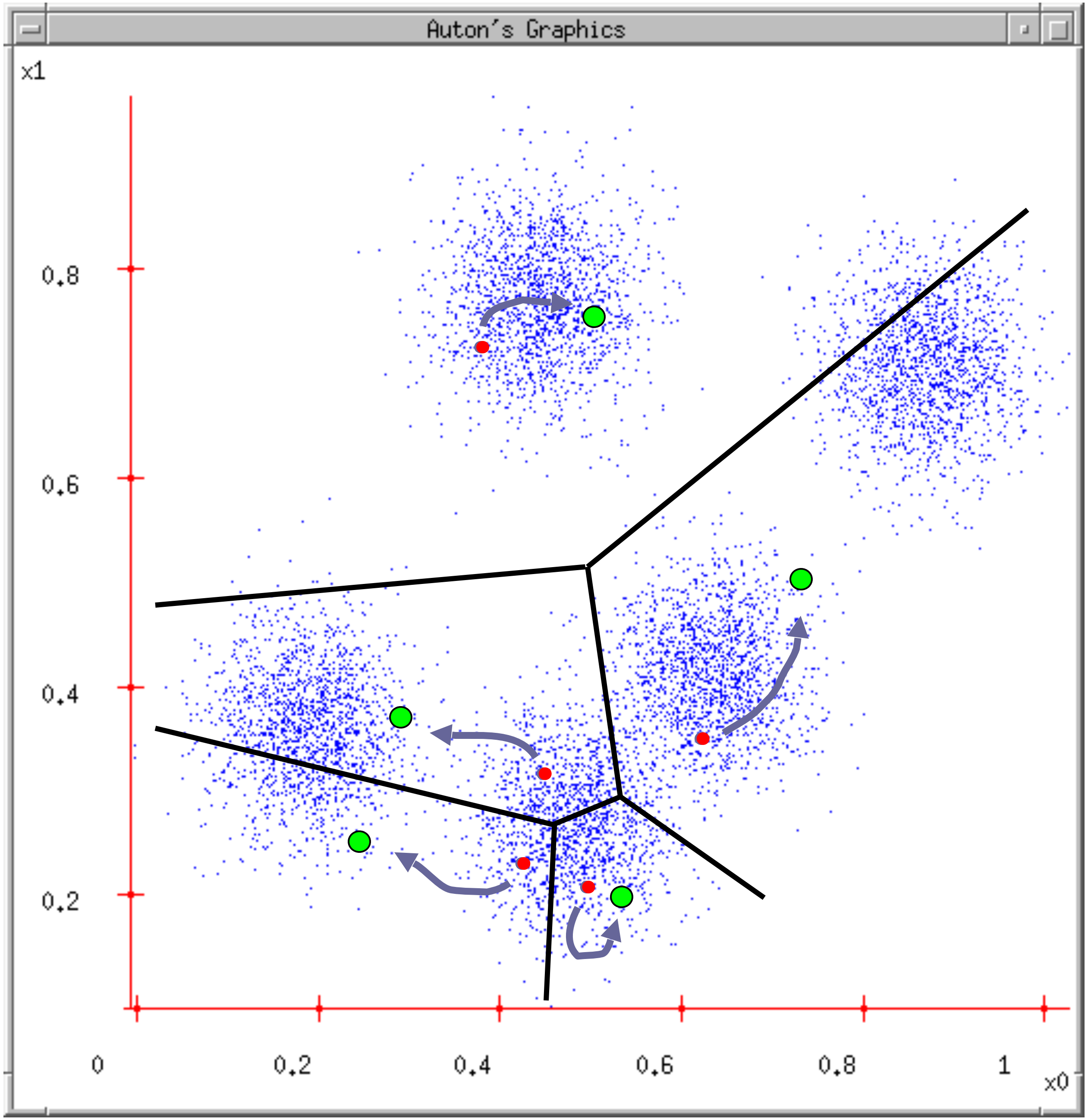
K-means

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)



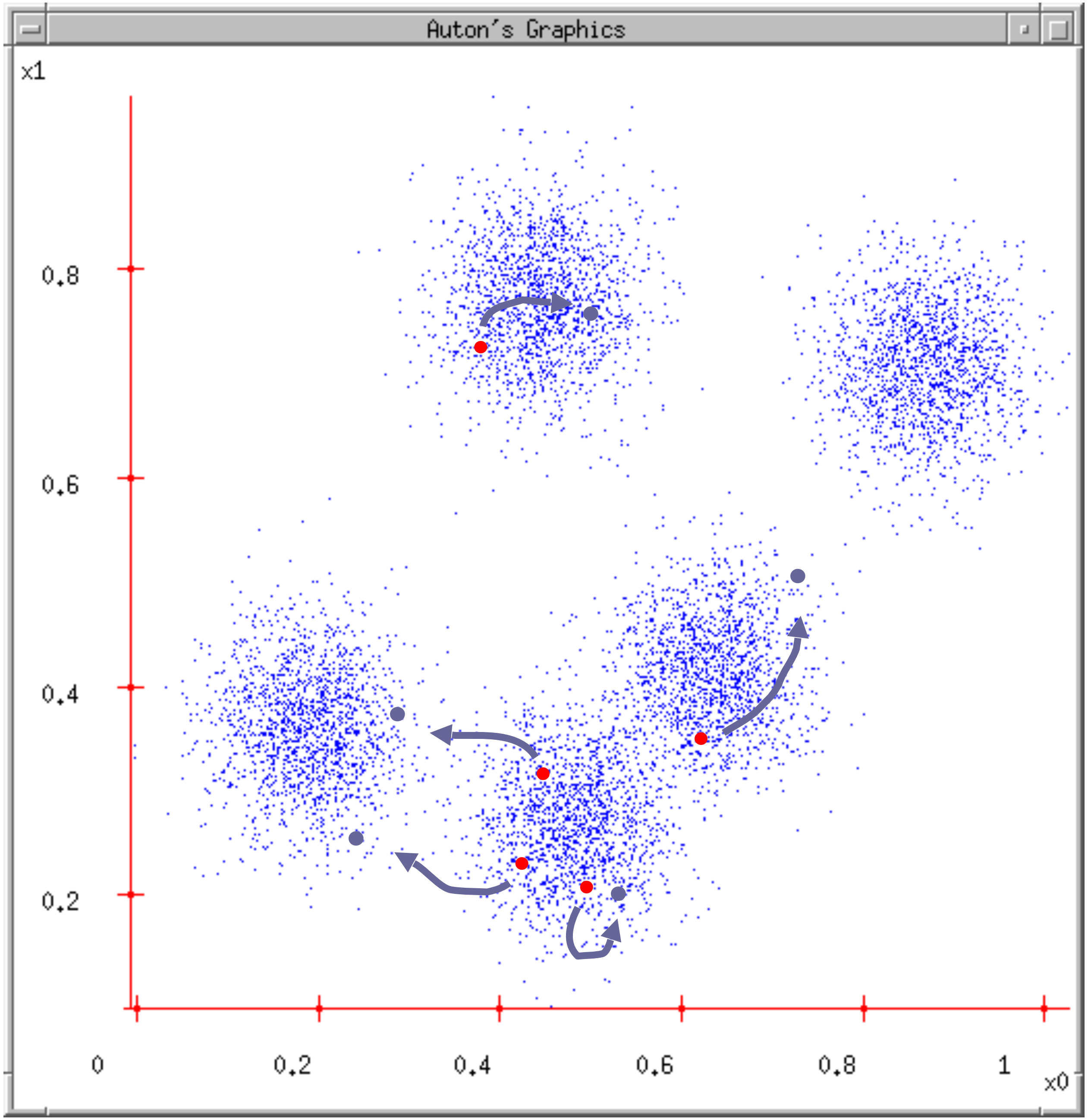
K-means

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
  - $\mu^{(0)} = \mu_1^{(0)}, \ldots, \mu_k^{(0)}$

- Classify: Assign each point $j \in \{1, \ldots N\}$ to nearest center:
  - $C^{(t)}(j) \leftarrow \arg\min_i ||\mu_i - x_j||^2$

- Recenter: μ_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 μ, 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 μ

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.

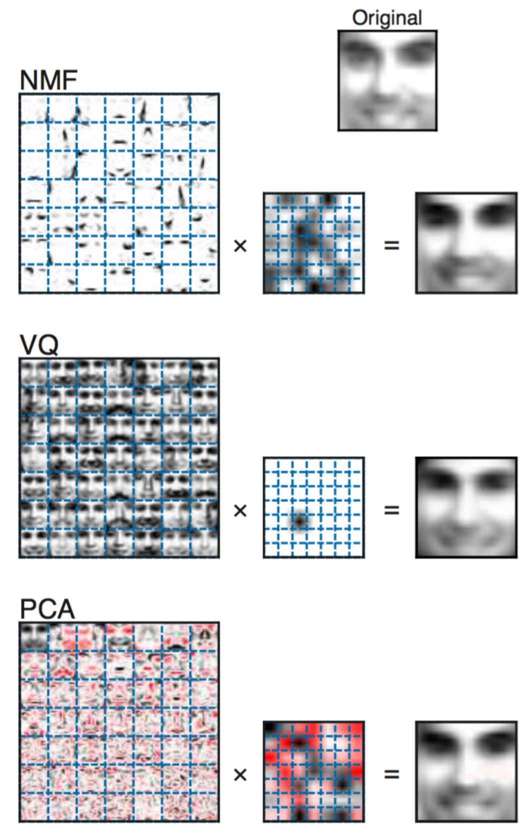
FIGURE 14.9. *Sir Ronald A. Fisher (1890 – 1962) was one of the founders of modern day statistics, to whom we owe maximum-likelihood, sufficiency, and many other fundamental concepts. The image on the left is a 1024×1024 grayscale image at 8 bits per pixel. The center image is the result of 2 × 2 block 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* 1024×1024 *grayscale image at 8 bits per pixel. The center image is the result of* 2×2 *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*



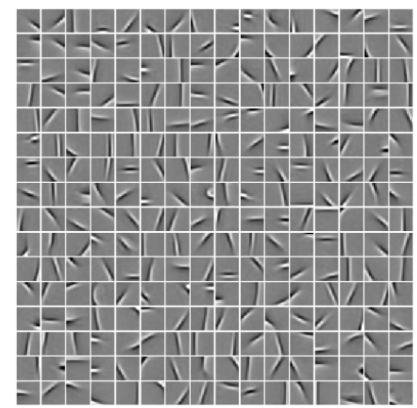
Vector Quantization, Fisher Vectors

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* 1024×1024 *grayscale image at 8 bits per pixel. The center image is the result of* 2×2 *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

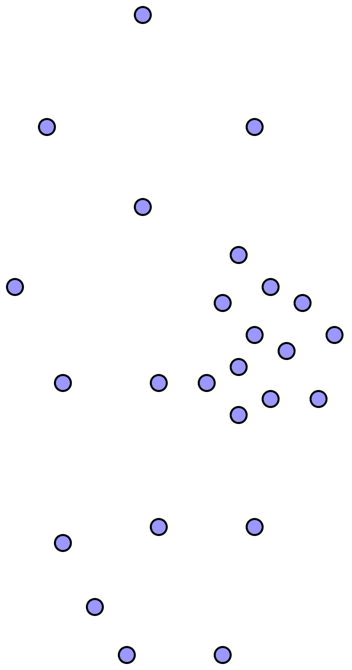


Similar reduced representation can be used as a feature vector

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

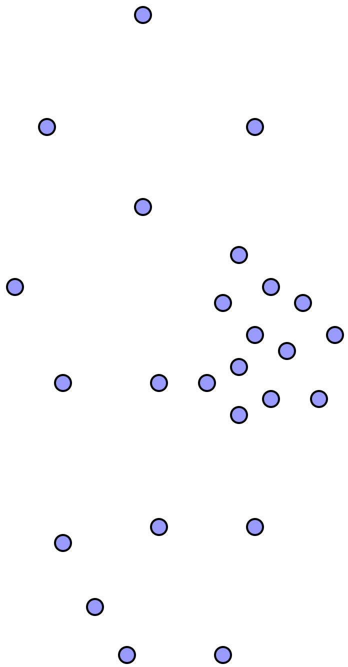
Mixtures of Gaussians

(One) bad case for k-means



(One) bad case for k-means

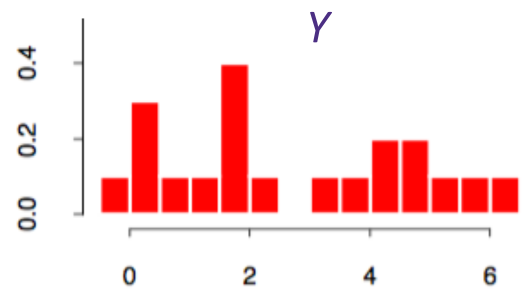
- Clusters may overlap
- Some clusters may be "wider" than others



Mixture models

$$
\begin{aligned}
Y_1 &\sim N(\mu_1, \sigma_1^2), \\
Y_2 &\sim N(\mu_2, \sigma_2^2), \\
Y &= (1 - \Delta) \cdot Y_1 + \Delta \cdot Y_2,
\end{aligned}
$$

$\Delta \in \{0, 1\}$ 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

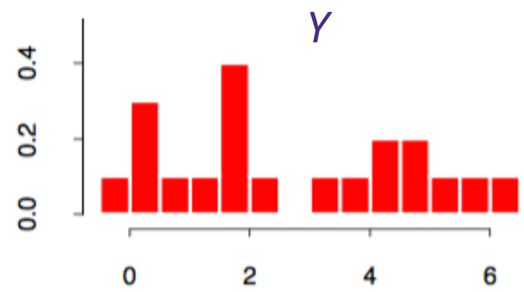
$$\ell(\theta; \mathbf{Z}) = \sum_{i=1}^n \log[(1-\pi)\phi_{\theta_1}(y_i) + \pi\phi_{\theta_2}(y_i)]$$

Mixture models

$$
\begin{aligned}
Y_1 &\sim N(\mu_1, \sigma_1^2), \\
Y_2 &\sim N(\mu_2, \sigma_2^2), \\
Y &= (1 - \Delta) \cdot Y_1 + \Delta \cdot Y_2,
\end{aligned}
$$

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

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



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

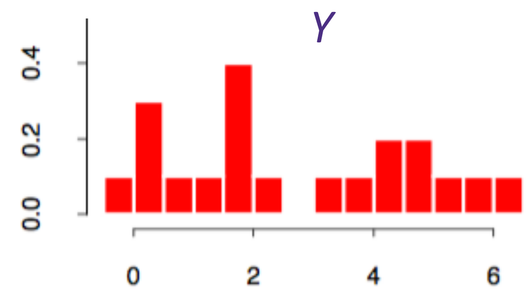
$\mathbf{\Delta} = \{\Delta_i\}_{i=1}^n$ is unobserved data

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

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

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

Mixture models



$$\begin{aligned} Y_1 &\sim N(\mu_1, \sigma_1^2), \\ Y_2 &\sim N(\mu_2, \sigma_2^2), \\ Y &= (1-\Delta)\cdot Y_1 + \Delta \cdot Y_2, \end{aligned}$$

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

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

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

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

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

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

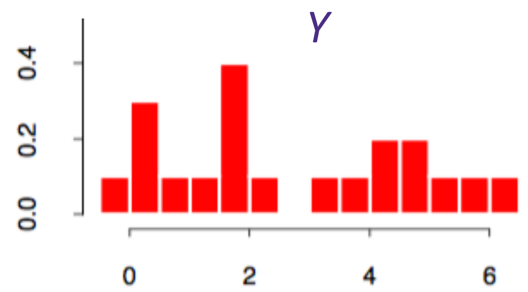
If we knew Δ, how would we choose θ?

Mixture models

$$
\begin{aligned}
Y_1 &\sim N(\mu_1, \sigma_1^2), \\
Y_2 &\sim N(\mu_2, \sigma_2^2), \\
Y &= (1-\Delta)\cdot Y_1 + \Delta \cdot Y_2,
\end{aligned}
$$

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

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



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

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

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

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

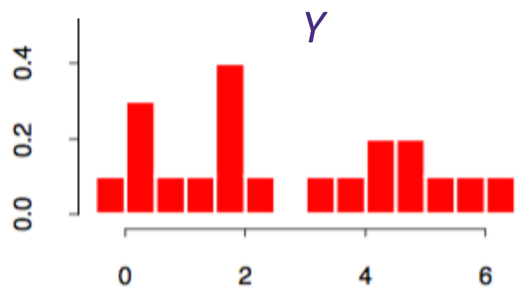
If we knew θ, how would we choose $\boldsymbol{\Delta}$?

Mixture models

$$
\begin{aligned}
Y_1 &\sim N(\mu_1, \sigma_1^2), \\
Y_2 &\sim N(\mu_2, \sigma_2^2), \\
Y &= (1-\Delta)\cdot Y_1 + \Delta \cdot Y_2,
\end{aligned}
$$

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

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



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

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

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

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

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

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

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

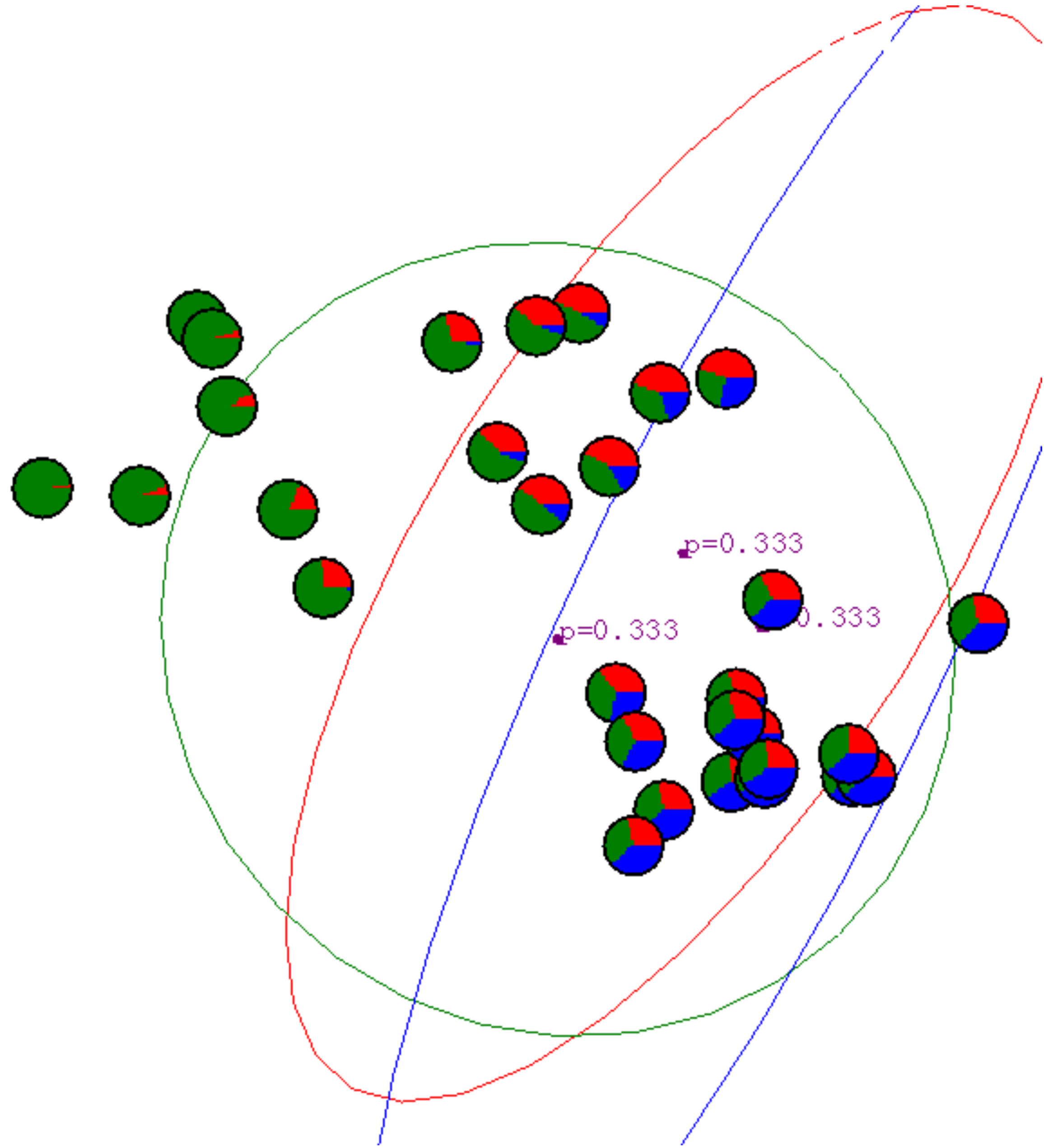
3. *Maximization Step*: compute the weighted means and variances:

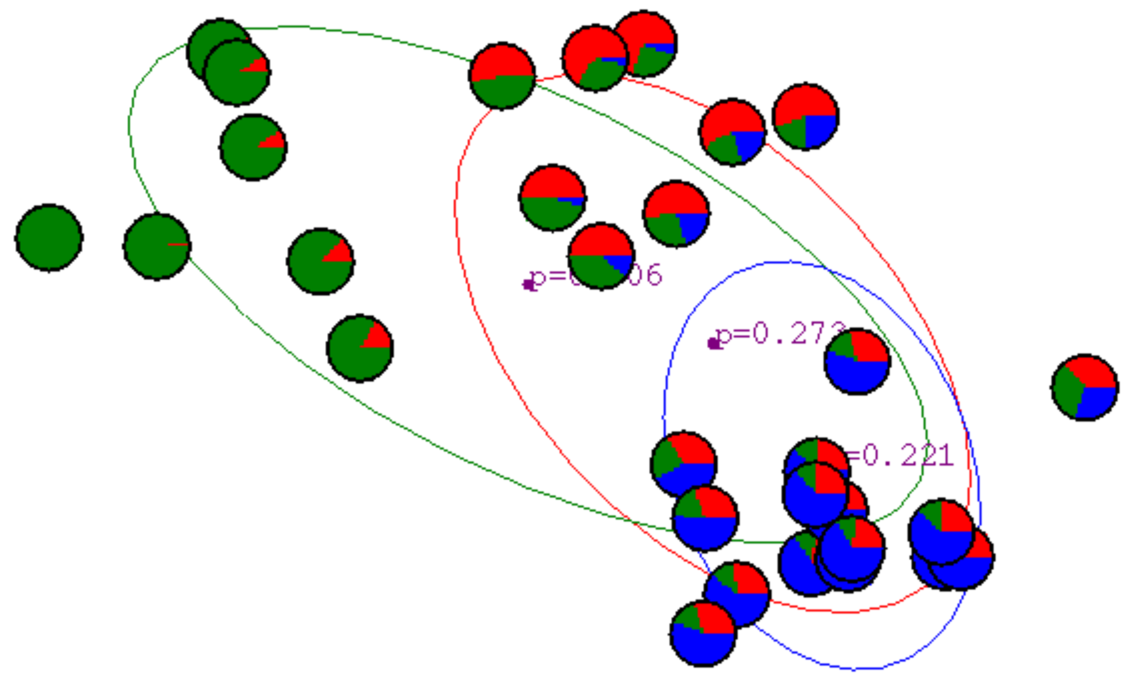
$$\hat{\mu}_1 = \frac{\sum_{i=1}^N (1-\hat{\gamma}_i) y_i}{\sum_{i=1}^N (1-\hat{\gamma}_i)}, \qquad \hat{\sigma}_1^2 = \frac{\sum_{i=1}^N (1-\hat{\gamma}_i)(y_i - \hat{\mu}_1)^2}{\sum_{i=1}^N (1-\hat{\gamma}_i)},$$

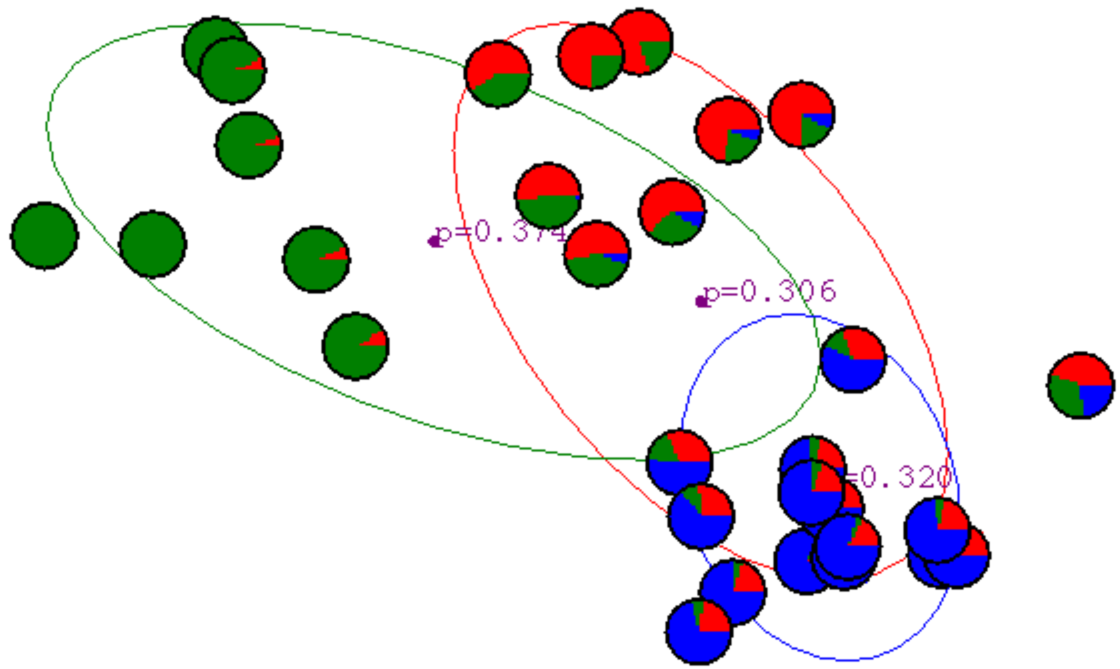
$$\hat{\mu}_2 = \frac{\sum_{i=1}^N \hat{\gamma}_i y_i}{\sum_{i=1}^N \hat{\gamma}_i}, \qquad \hat{\sigma}_2^2 = \frac{\sum_{i=1}^N \hat{\gamma}_i (y_i - \hat{\mu}_2)^2}{\sum_{i=1}^N \hat{\gamma}_i},$$

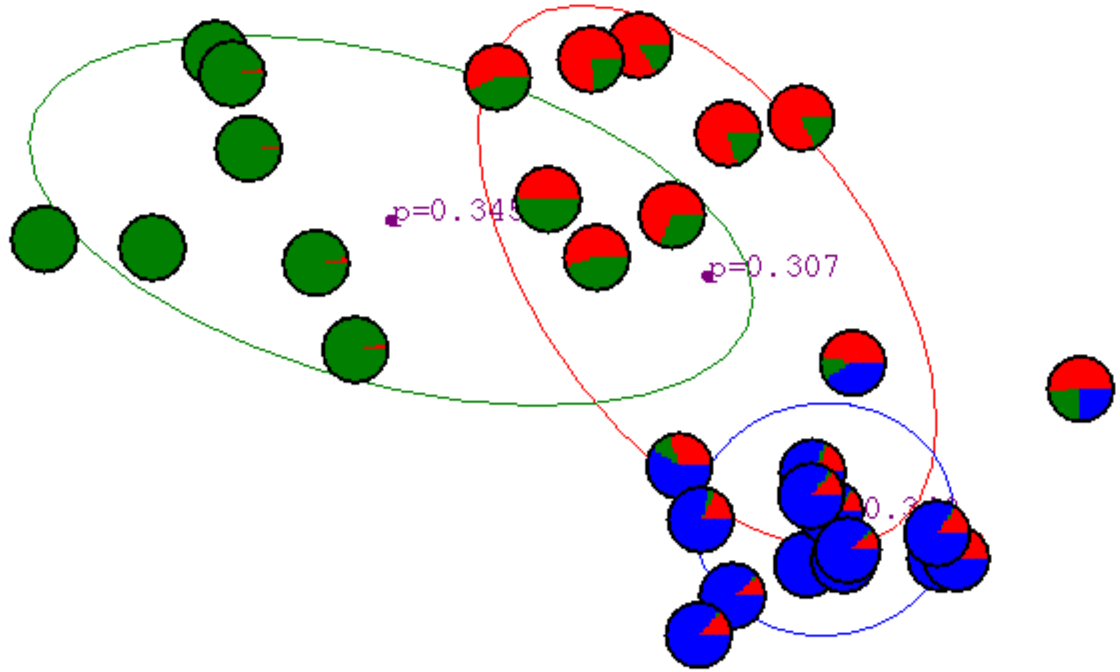
   and the mixing probability $\hat{\pi} = \sum_{i=1}^N \hat{\gamma}_i / N$.

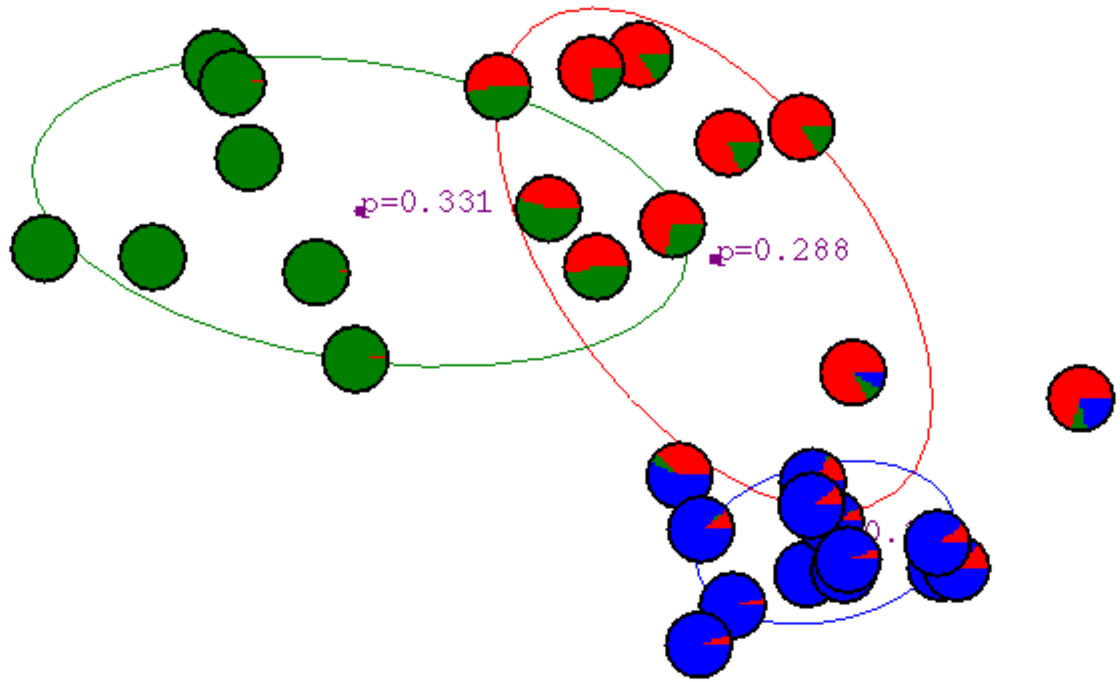
4. Iterate steps 2 and 3 until convergence.


p=0.333
p=0.333
0.333

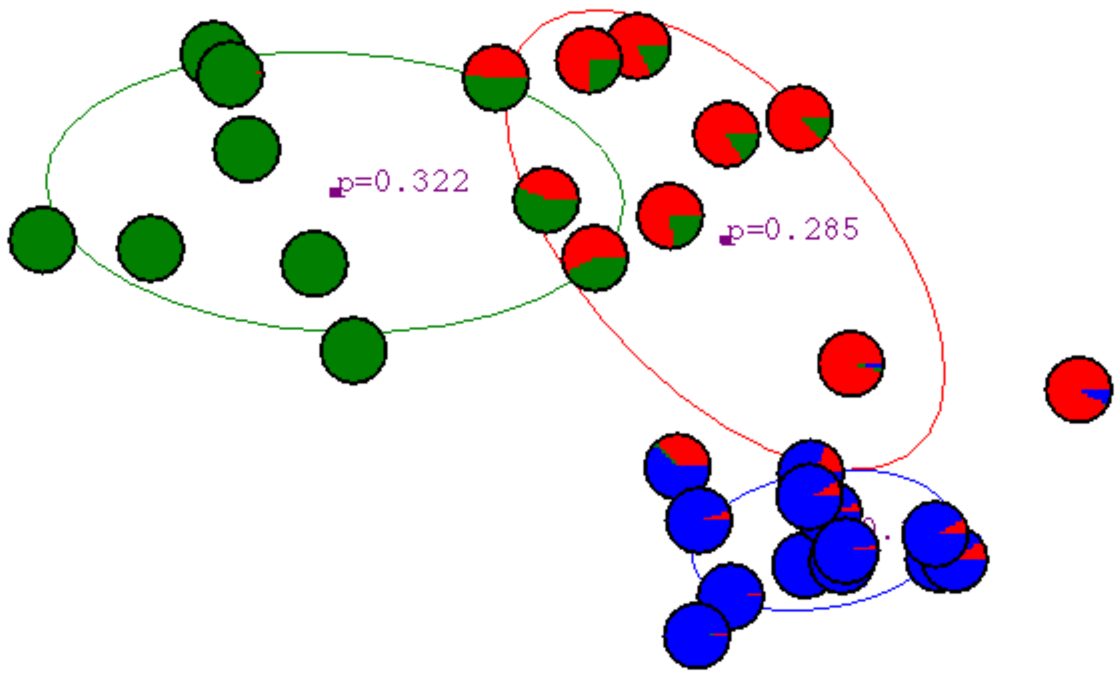

p=0.27
=0.221

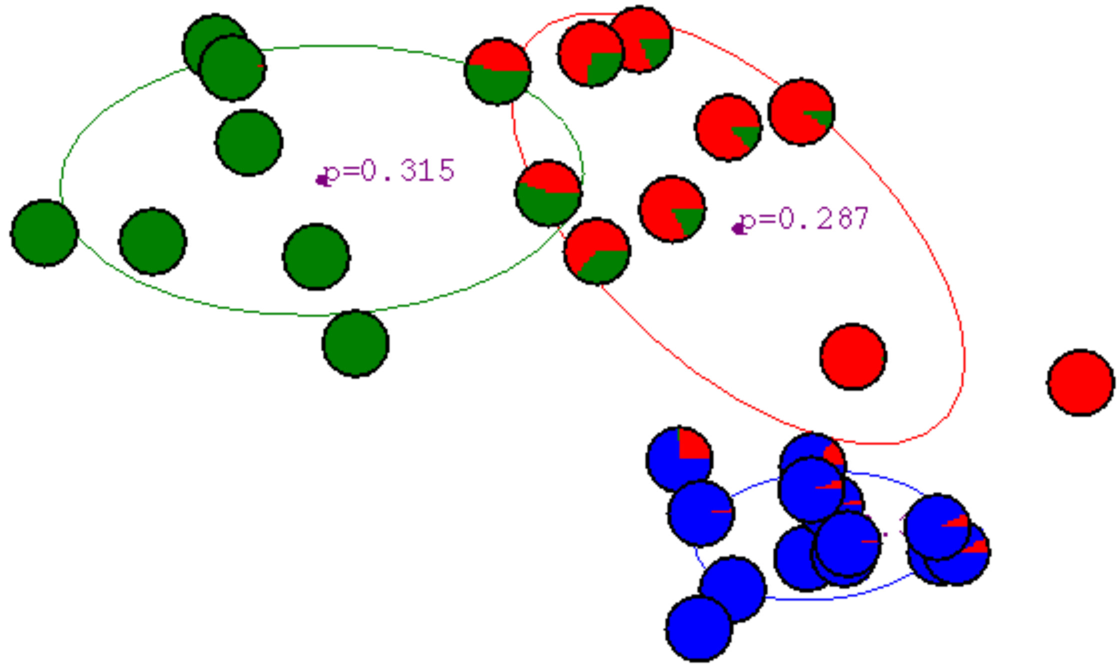

p=0.374
p=0.306
=0.320

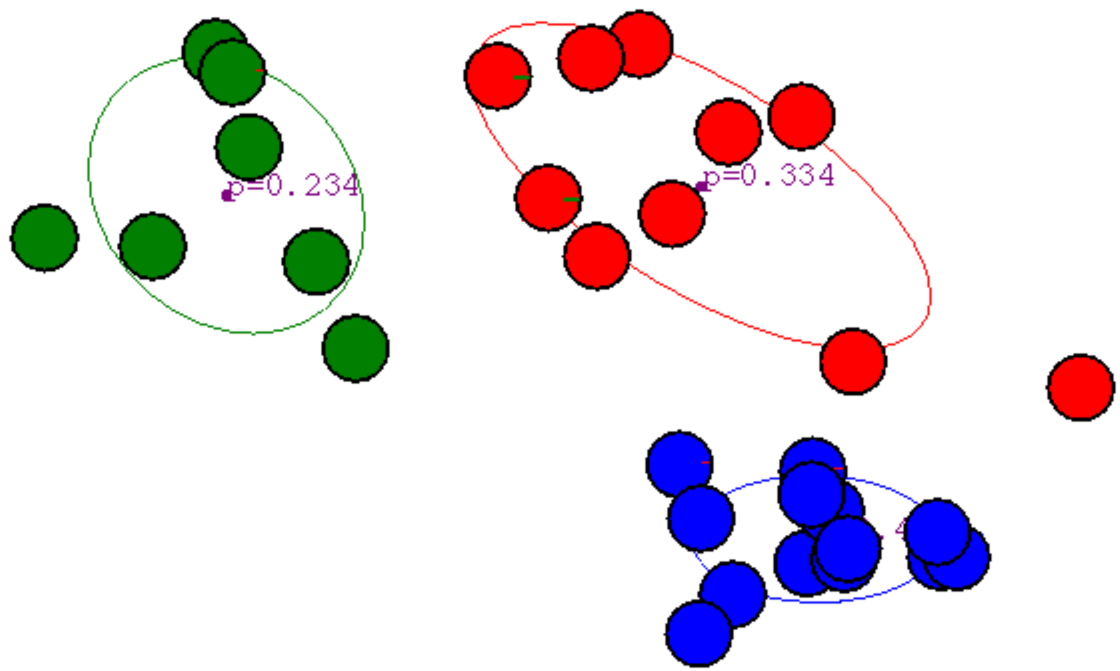

p=0.345
p=0.307

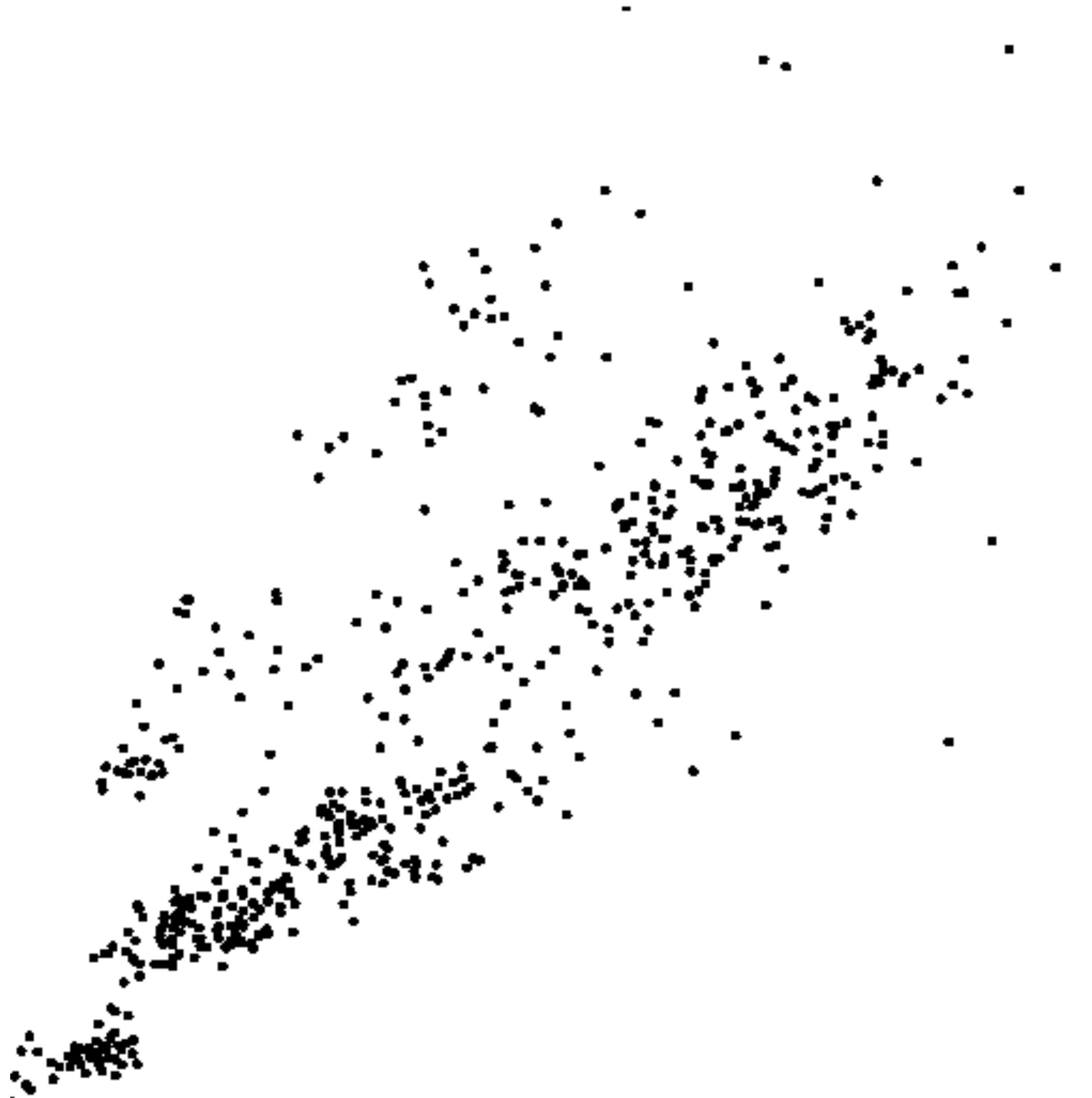


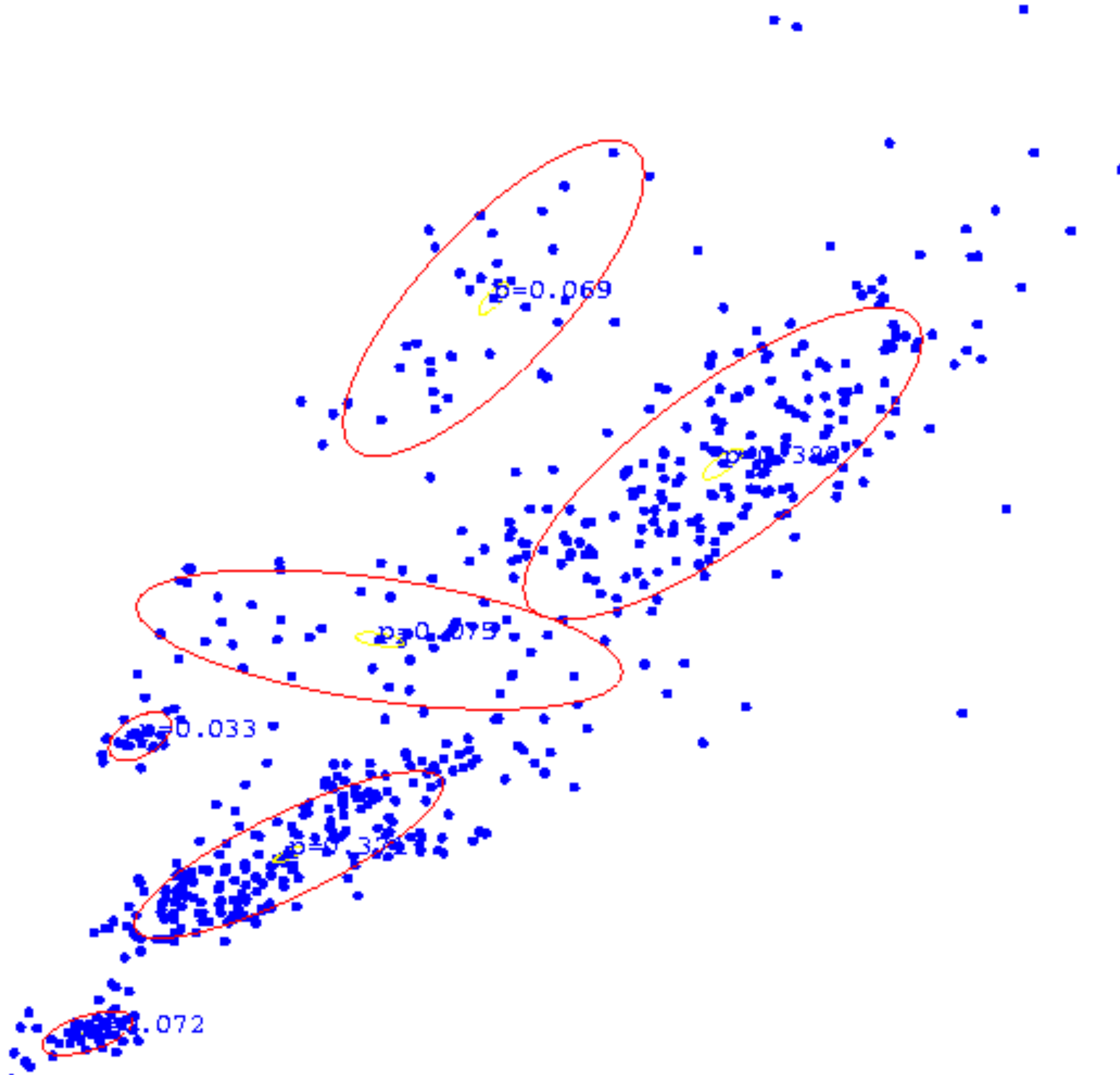
p=0.331
p=0.288

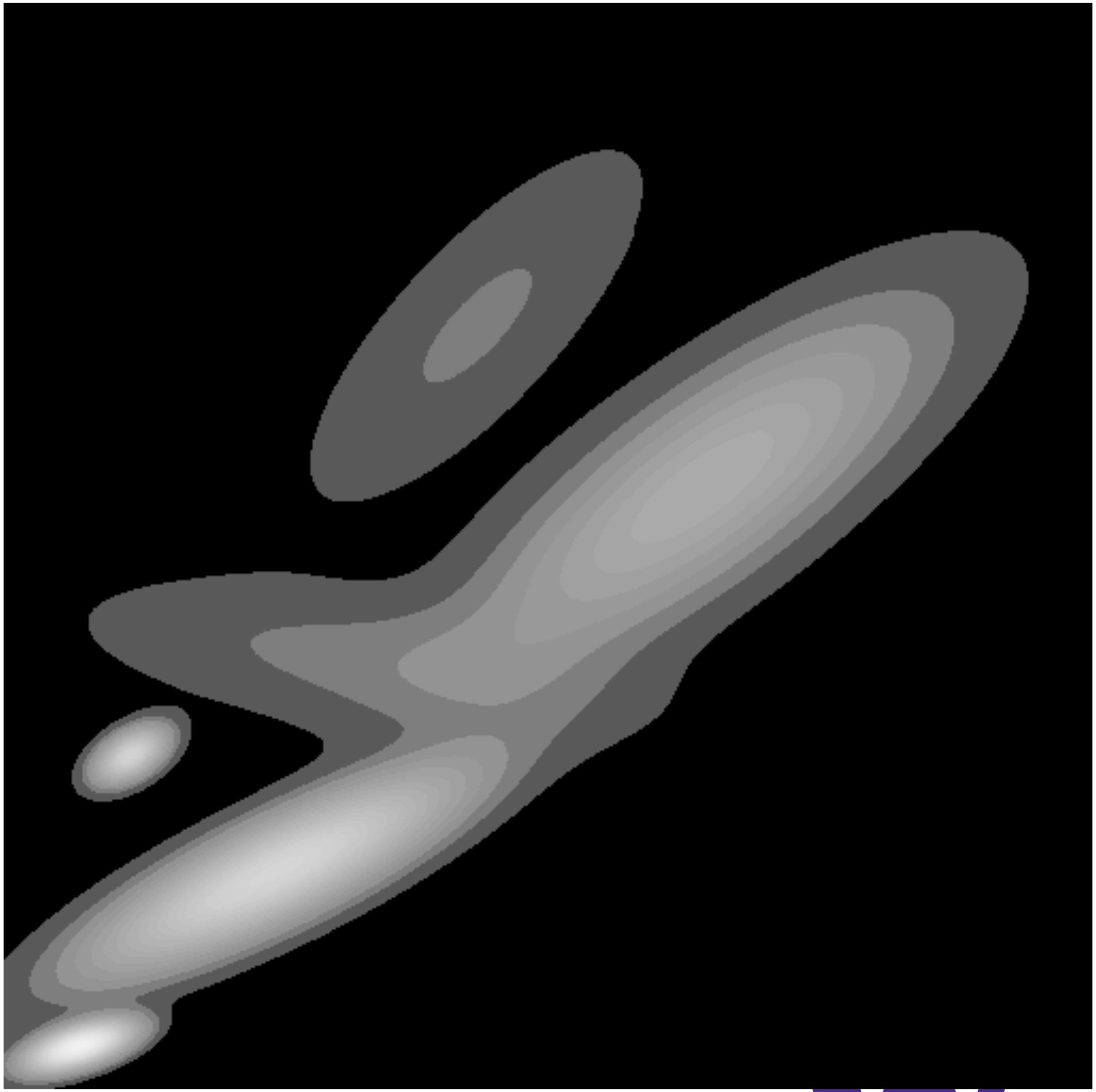

p=0.322
p=0.285


p=0.315
p=0.287


p=0.234
p=0.334

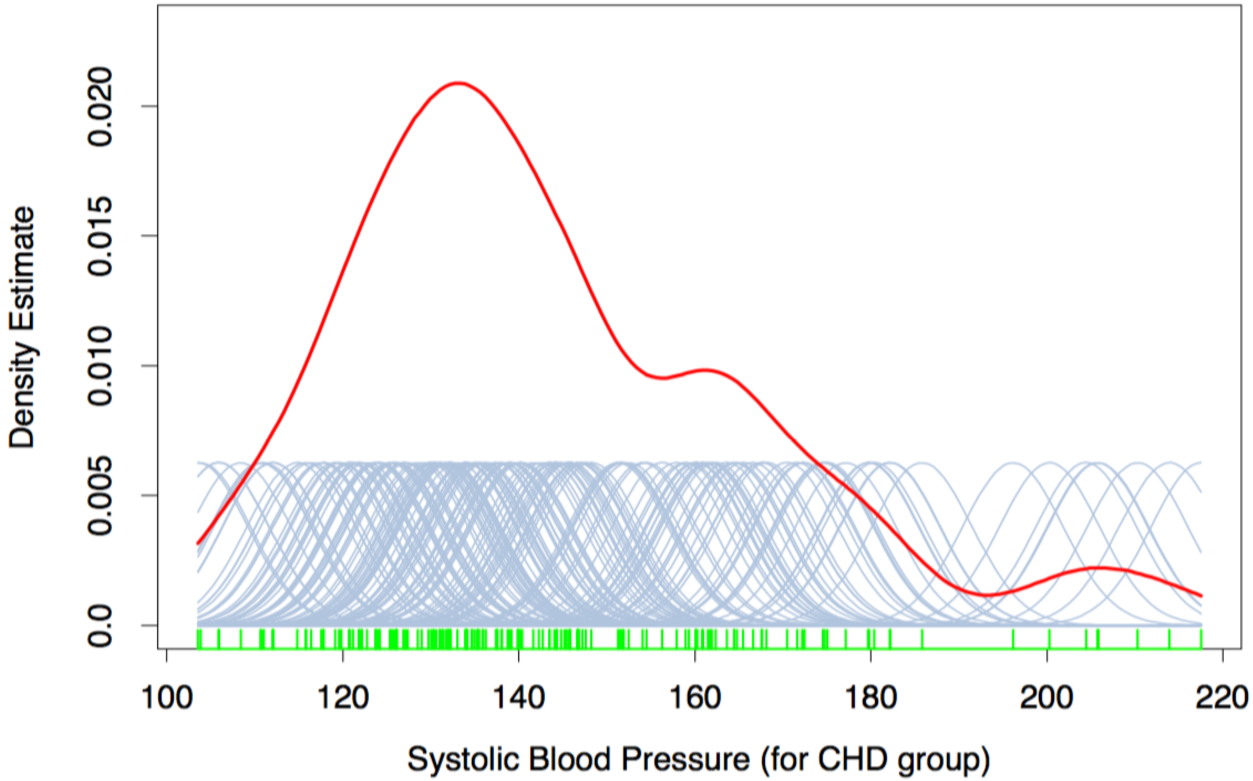



p=0.069
0.033



Kernel Density Estimation

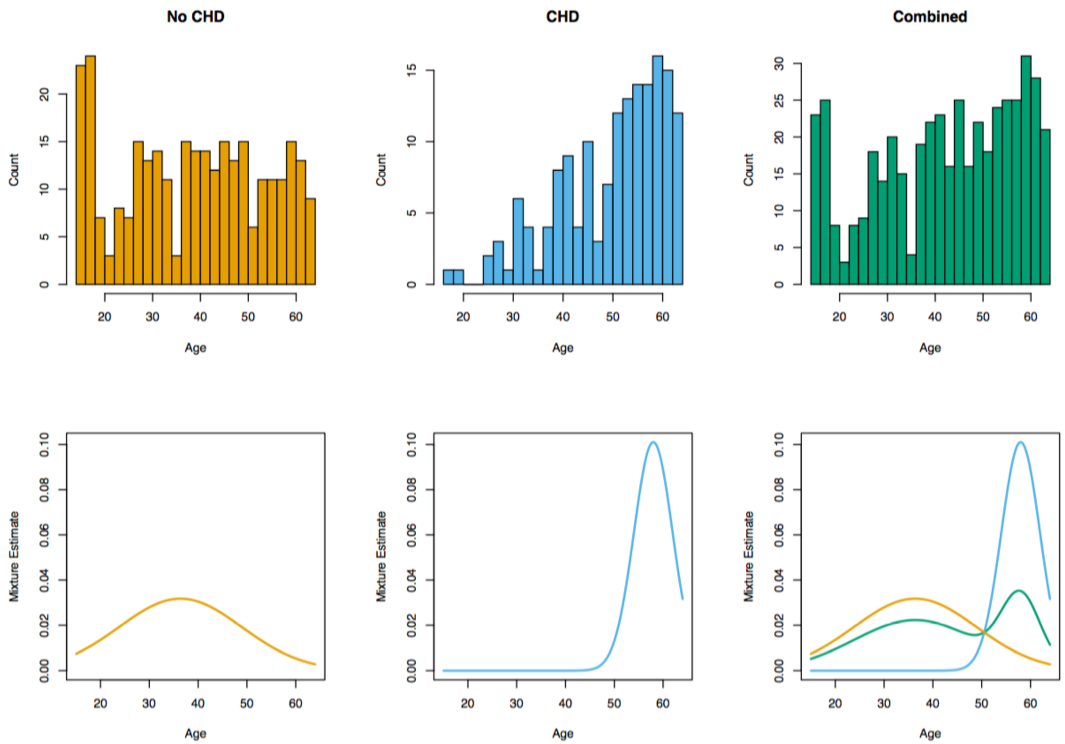
Kernel Density Estimation



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

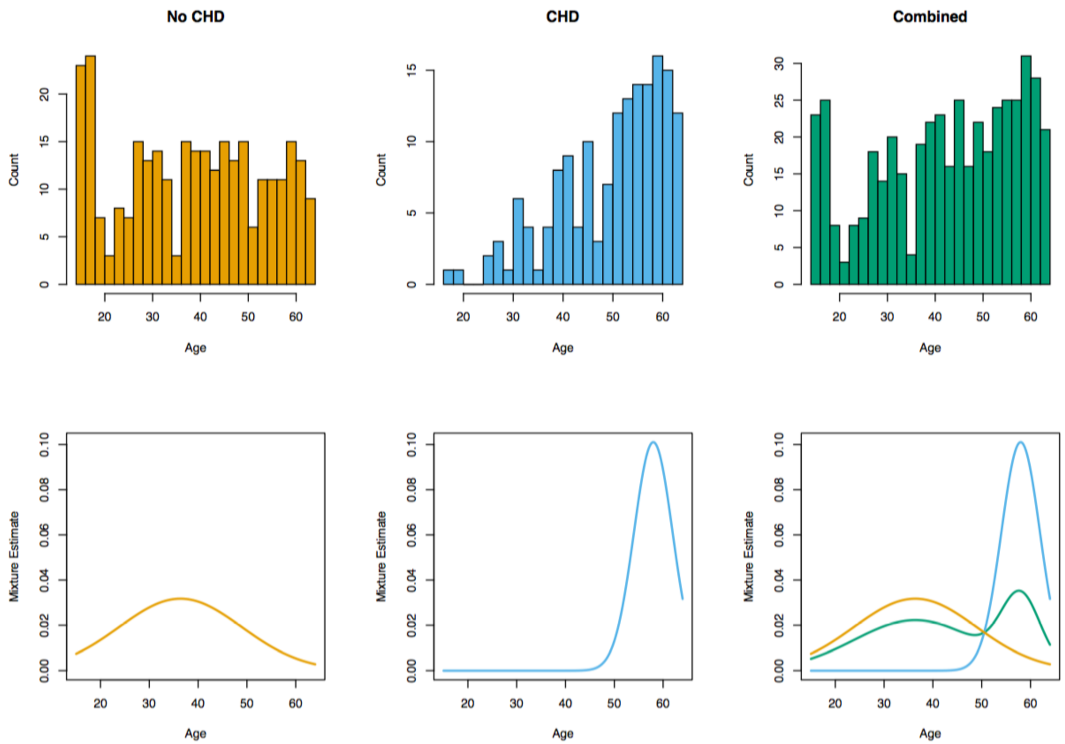
A very “lazy” GMM

Kernel Density Estimation



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

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