

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

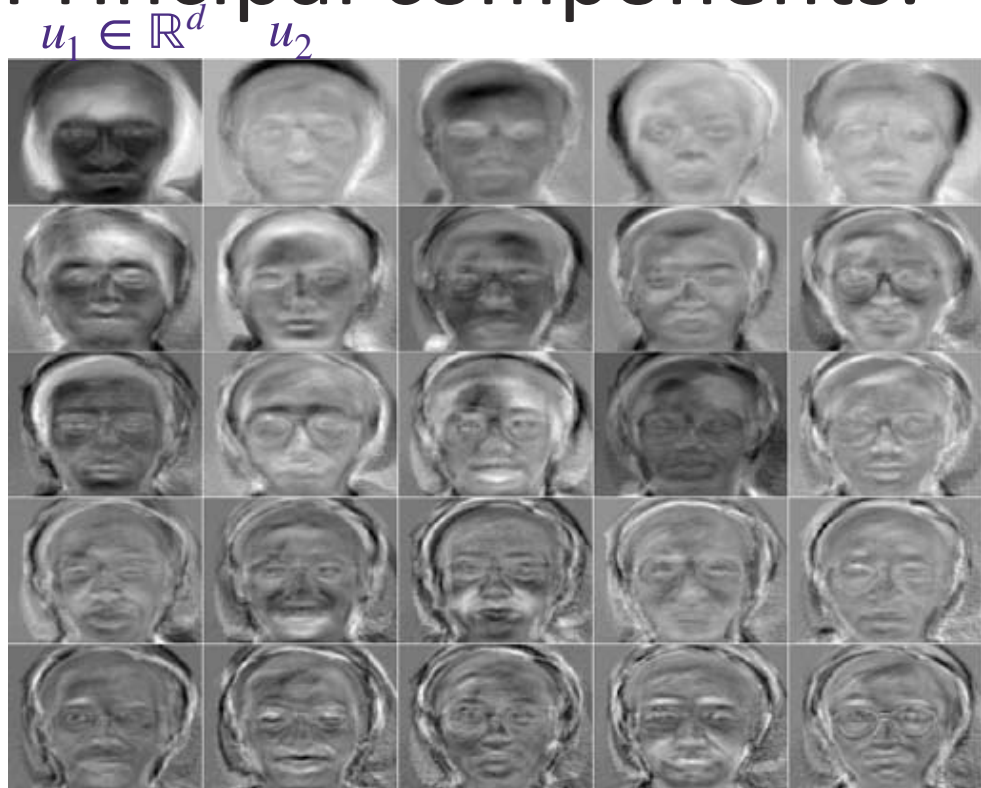
n images

$d \times n$ 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)

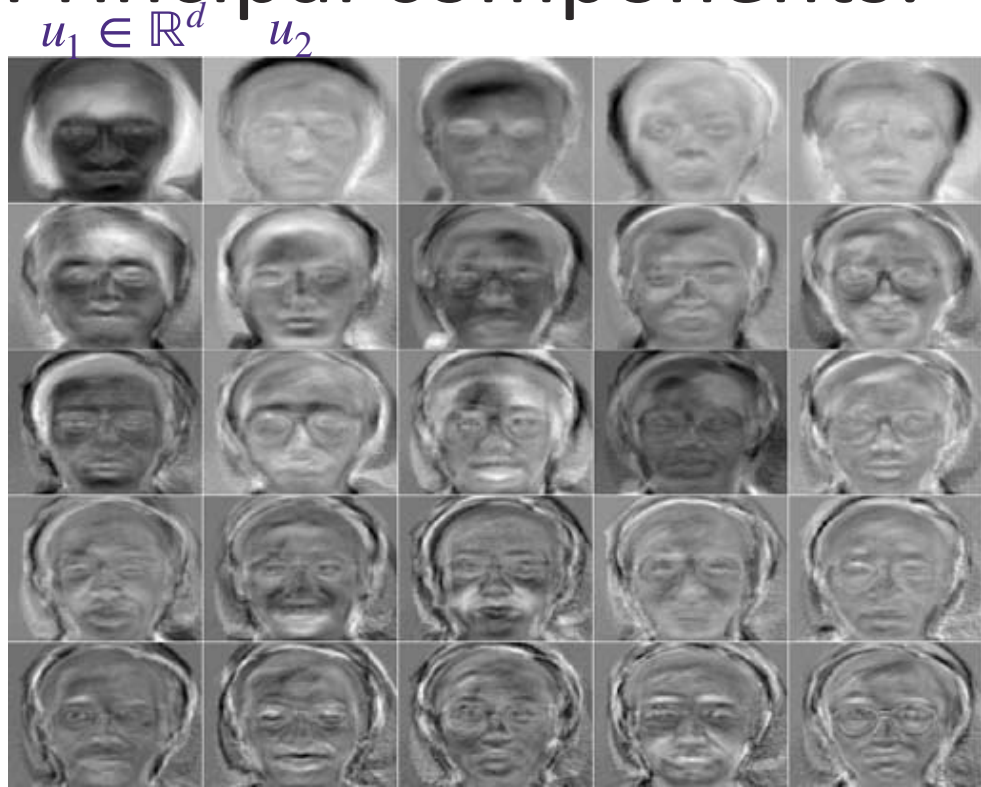
Principal components:



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:

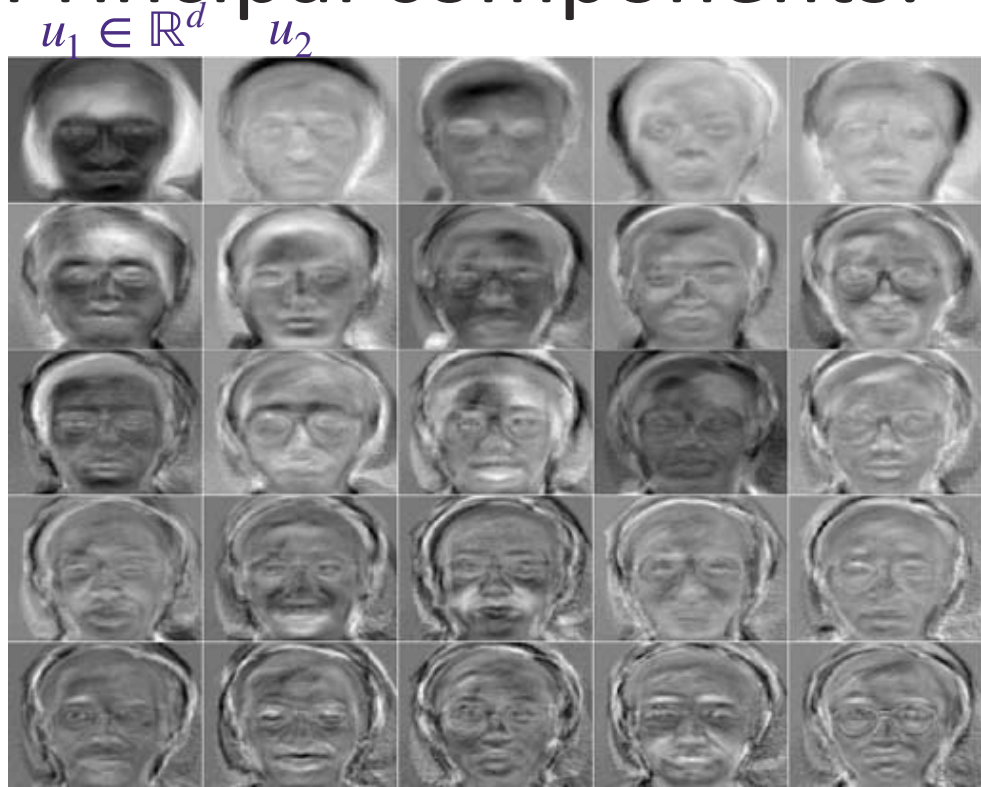


$$\approx a[1]u_1 + a[2]u_2 + \dots + a[25]u_{25}$$

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 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 :

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

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

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

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 μ, λ_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

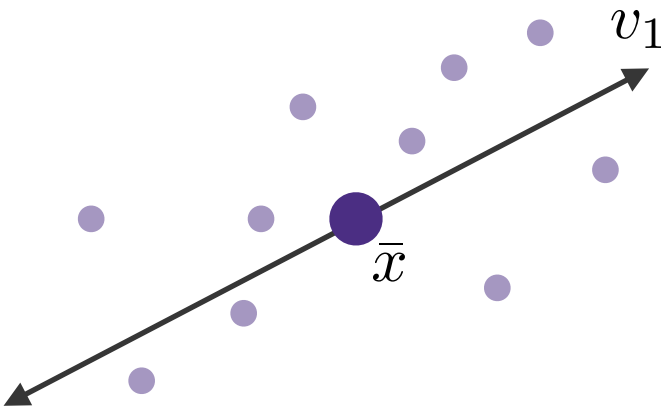
$$\min_{\mathbf{V}_q} \sum_{i=1}^N \|(x_i - \bar{x}) - \mathbf{V}_q \mathbf{V}_q^T (x_i - \bar{x})\|^2.$$

$\mathbf{V}_q \mathbf{V}_q^T$ 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 $q = 1$

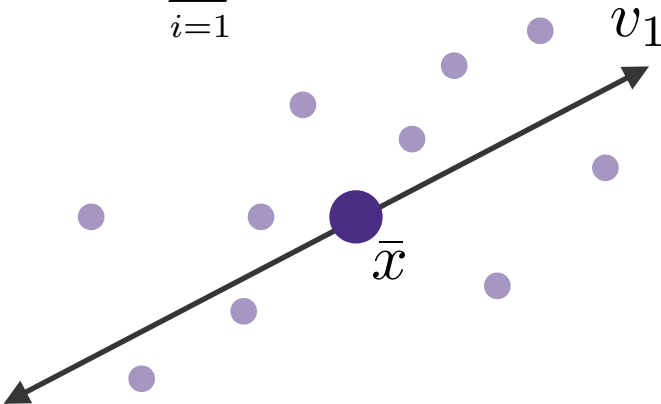
$$\begin{aligned} v_1 &= \arg \min_{v: \|v\|_2=1} \sum_{i=1}^N \|(x_i - \bar{x}) - vv^T (x_i - \bar{x})\|_2^2 \\ &= \arg \min_{v: \|v\|_2=1} \sum_{i=1}^N \|x_i - \bar{x}\|_2^2 - 2(x_i - \bar{x})^T vv^T (x_i - \bar{x}) \\ &\quad + (x_i - \bar{x})^T vv^T vv^T (x_i - \bar{x}) \end{aligned}$$

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

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

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

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



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

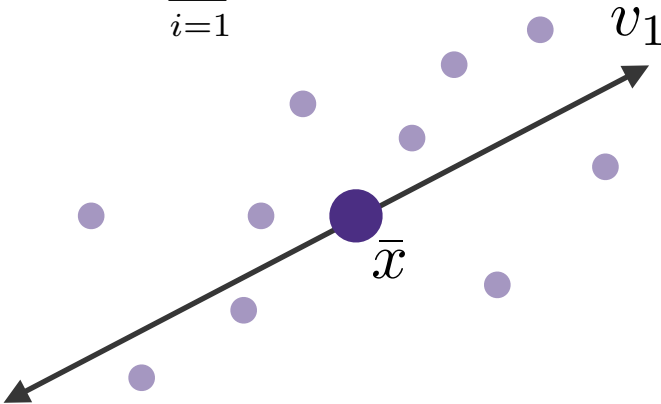
$$\hat{x}_i := \bar{x} + \mathbf{V}_q \mathbf{V}_q^T (x_i - \bar{x}) = \bar{x} + \sum_{j=1}^q v_j \langle v_j, x_i - \bar{x} \rangle$$

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

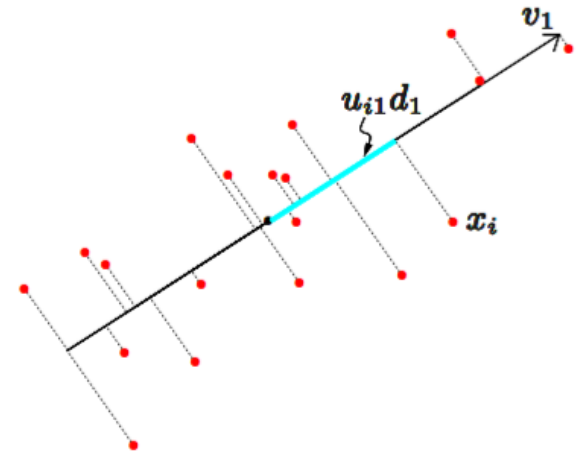
$$\mathbf{V}_q^T \mathbf{V}_q = I_q$$

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

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

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 Σ

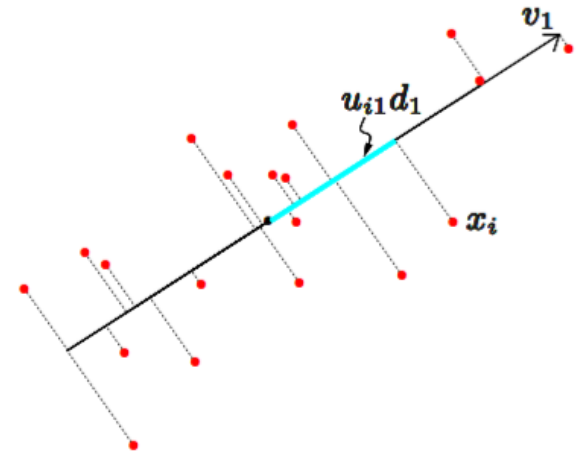
\mathbf{V}_q are the first q principal components

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

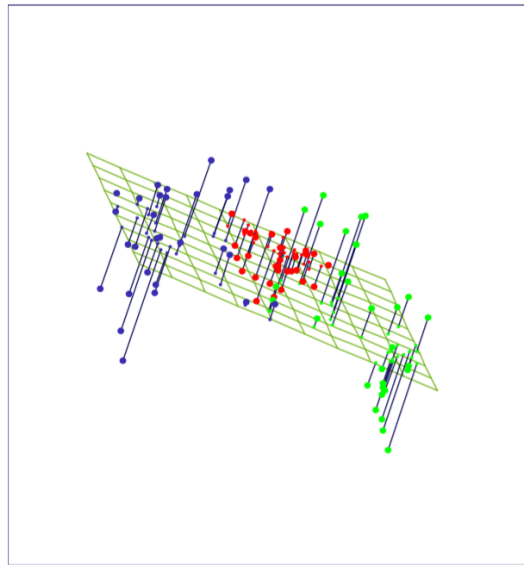
$$\mathbf{X} - \mathbf{1}\bar{x}^T = \mathbf{U} \mathbf{S} \mathbf{V}^T$$



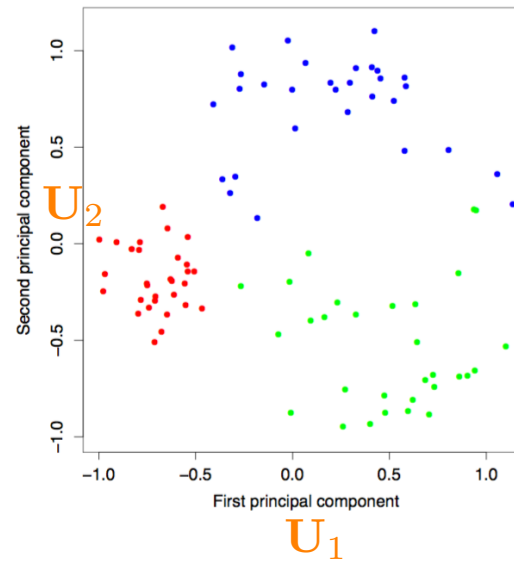
$$\Sigma := \sum_{i=1}^N (x_i - \bar{x})(x_i - \bar{x})^T$$

Dimensionality reduction

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



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



Dimensionality reduction

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

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

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

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

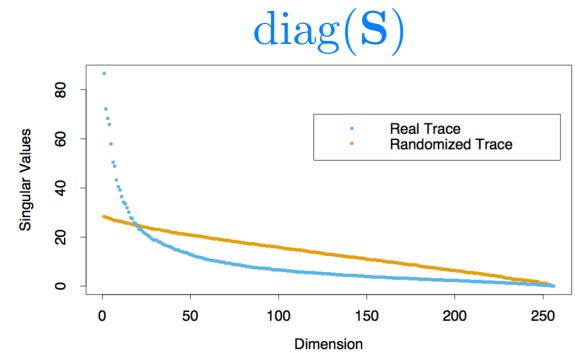
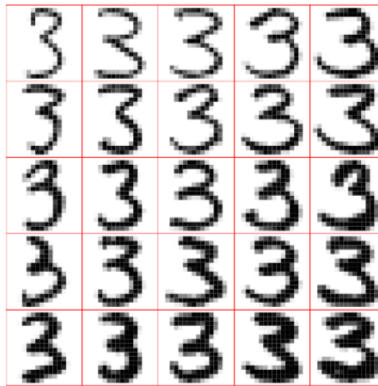
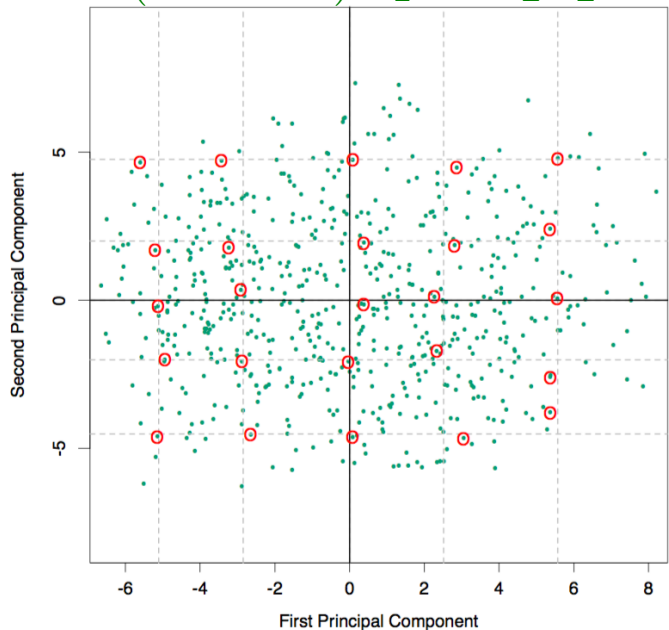
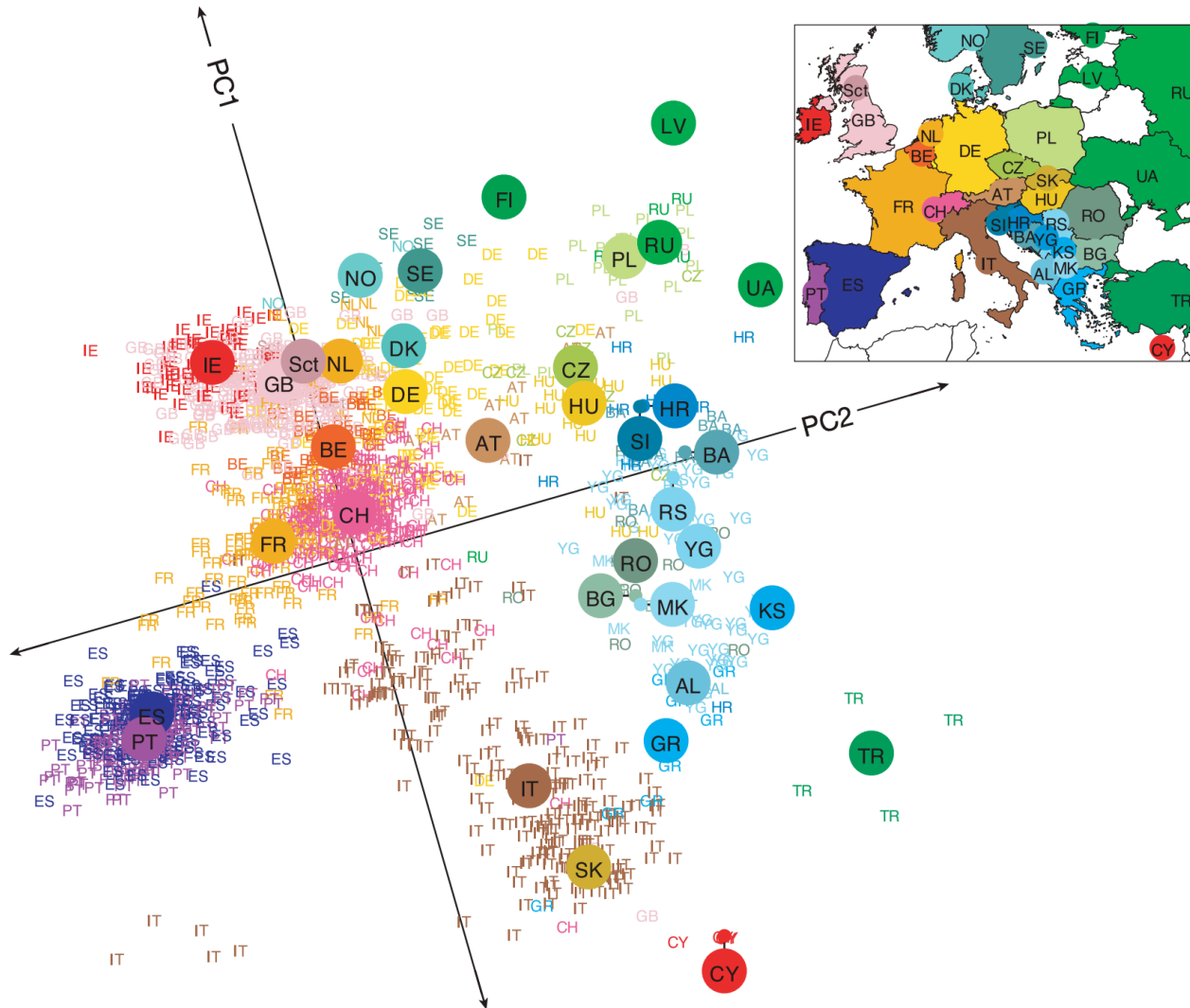


FIGURE 14.24. The 256 singular values for the digitized threes, compared to those for a randomized version of the data (each column of \mathbf{X} was scrambled).

Dimensionality reduction



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

Kernel PCA

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

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

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

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

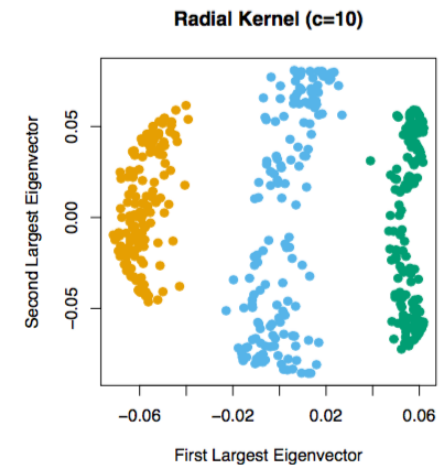
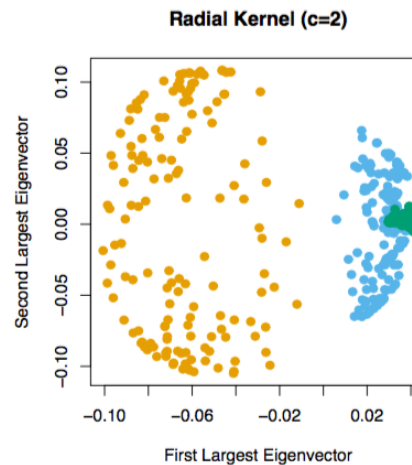
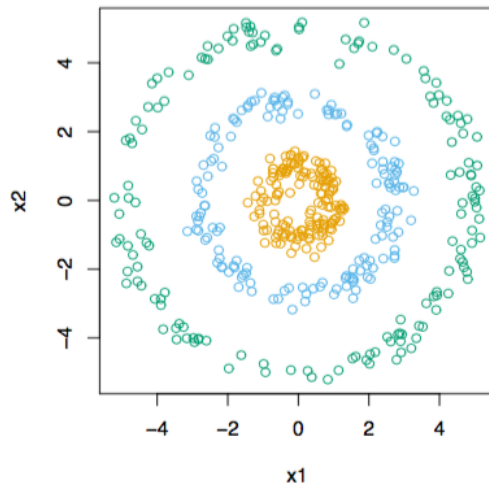
Kernel PCA

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

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

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

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



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

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

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,

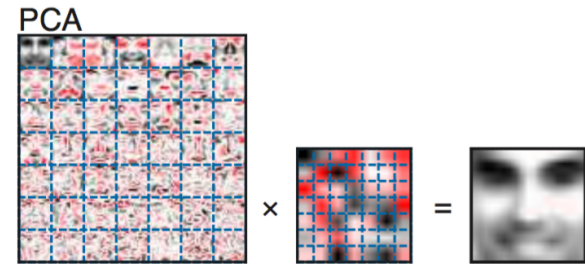
$$\Pr\left(\left(1 - \epsilon\right)\|x\|^2 \leq \left\|\frac{1}{\sqrt{k}}Ax\right\|^2 \leq \left(1 + \epsilon\right)\|x\|^2\right) \geq 1 - 2e^{-(\epsilon^2 - \epsilon^3)k/4}$$

Other matrix factorizations

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

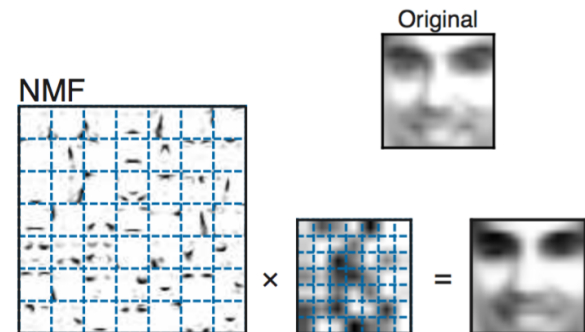
Singular value decomposition

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



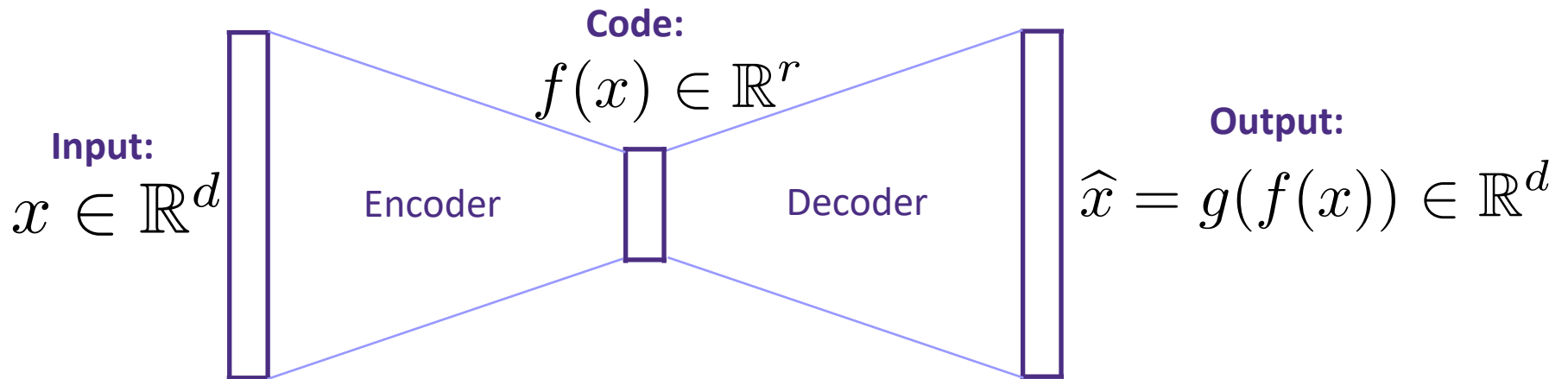
Nonnegative matrix factorization (NMF)

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



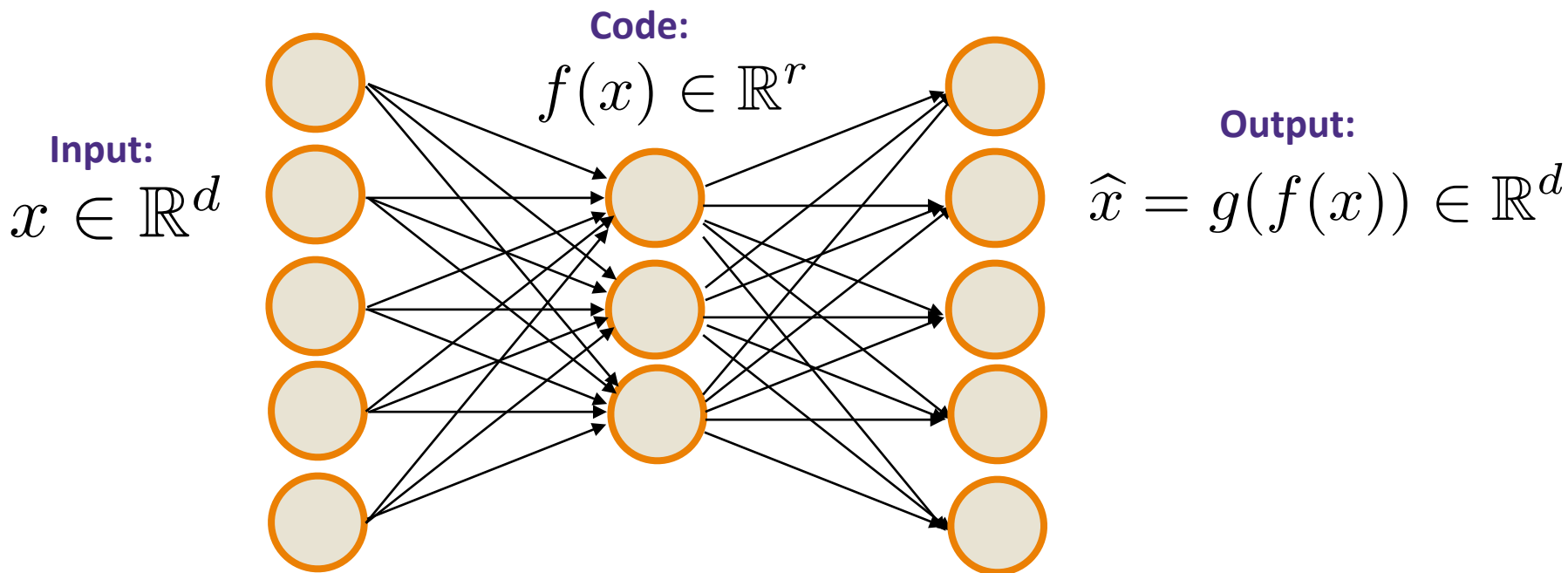
Autoencoders

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



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

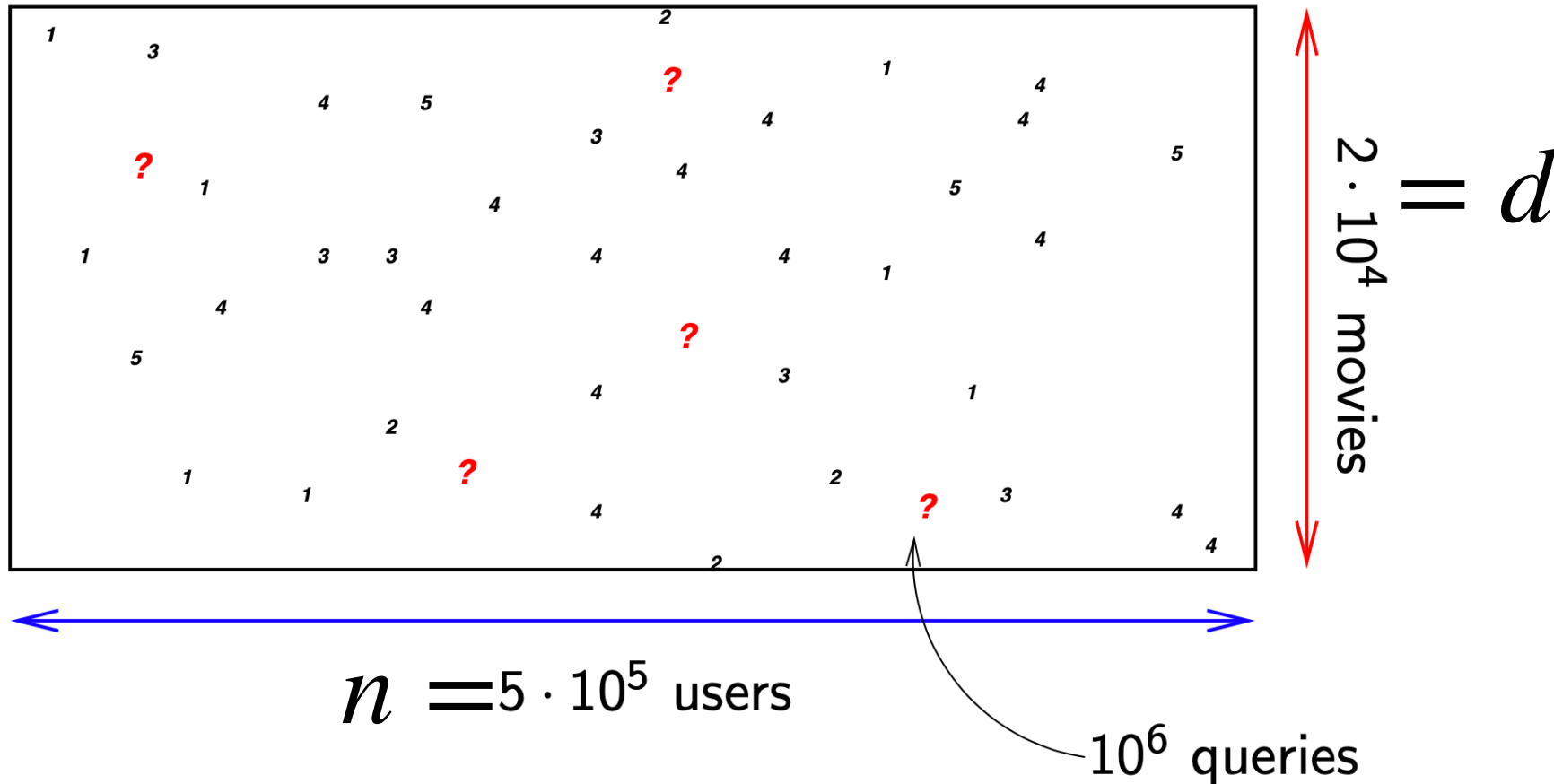
What if $f(X) = Ax$ and $g(y) = By$?

Matrix Completion



Matrix completion for recommendation systems

Netflix challenge dataset

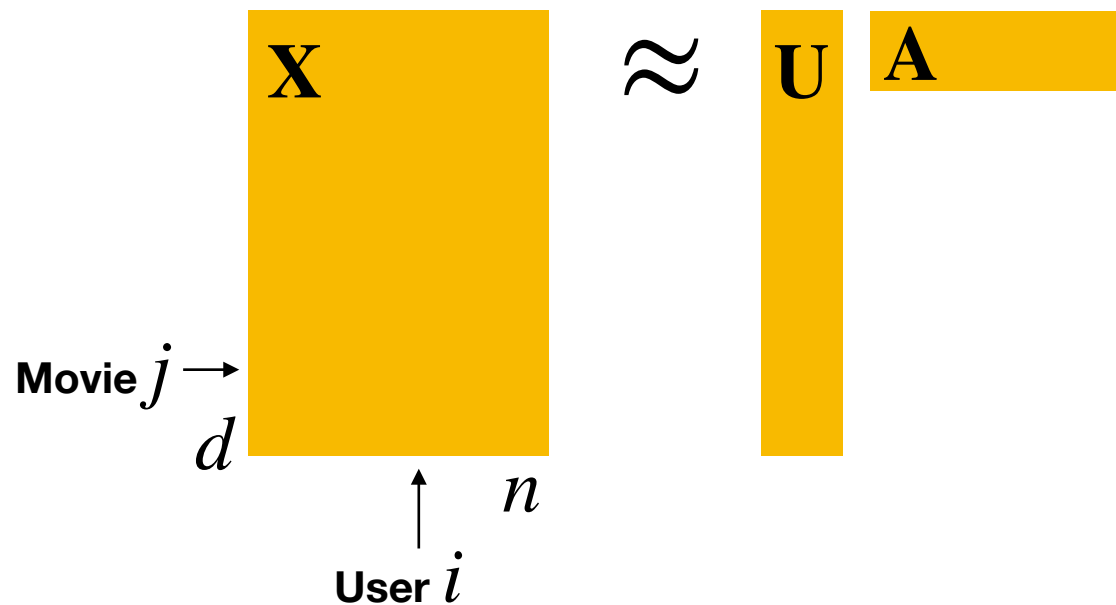


- users provide ratings on a few movies, and we want to predict the missing entries in this ratings matrix, so that we can make recommendations
- without any assumptions, the missing entries can be anything, and no prediction is possible

Matrix completion problem

Matrix completion

- let $\mathbf{X} = [x_1 \ x_2 \ \cdots \ x_n] \in \mathbb{R}^{d \times n}$ be the ratings matrix, and assume it is fully observed, i.e. we know all the entries
- then we want to find $\mathbf{U} \in \mathbb{R}^{d \times r}$ and $\mathbf{A} = [a_1 \ a_2 \ \cdots \ a_n] \in \mathbb{R}^{r \times n}$ that approximates \mathbf{X}



- if we **observe all entries** of \mathbf{X} , then we can find the best rank- r approximation with SVD

Optimization for Matrix Completion

- a natural approach to fit v_j 's and a_i 's to given training data is to solve

$$\text{minimize}_{\mathbf{U}, \mathbf{A}} \sum_{(i,j) \in S_{\text{train}}} (\mathbf{X}_{ji} - v_j^T a_i)^2$$

- this can be solved, for example via gradient descent or alternating minimization

Matrix Recovery

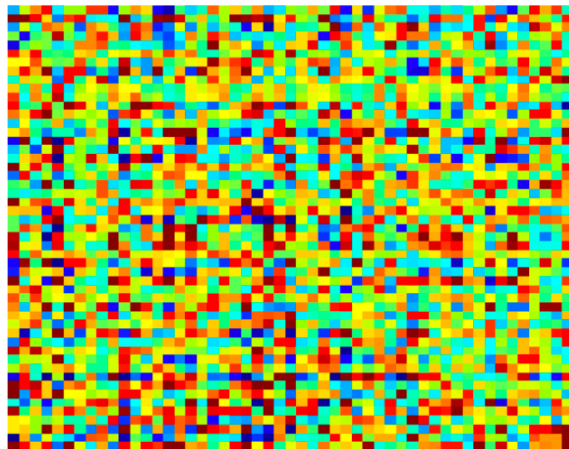
- a natural approach to fit v_j 's and a_i 's to given training data is to solve

$$\text{minimize}_{\mathbf{U}, \mathbf{A}} \sum_{(i,j) \in S_{\text{train}}} (\mathbf{X}_{ji} - v_j^T a_i)^2$$

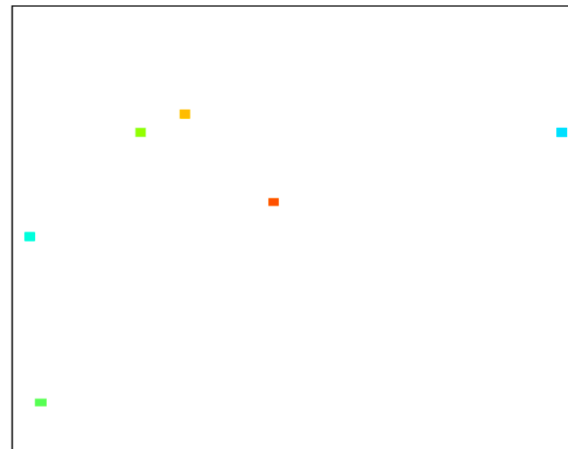
- If X is low rank, why only observe a few ($\ll dn$) entries, we can recover X ?

Example: 2000×2000 rank-8 random matrix

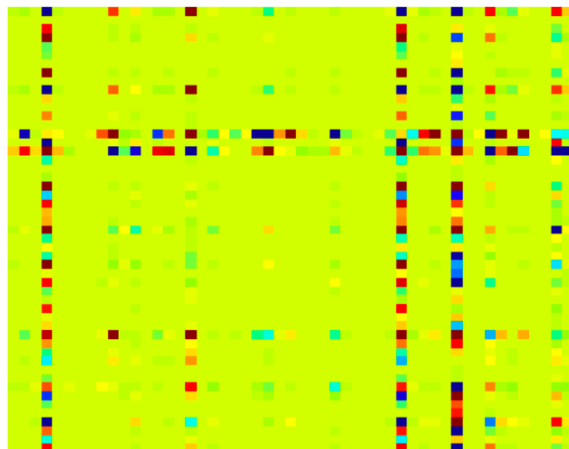
low-rank matrix \mathbf{X}



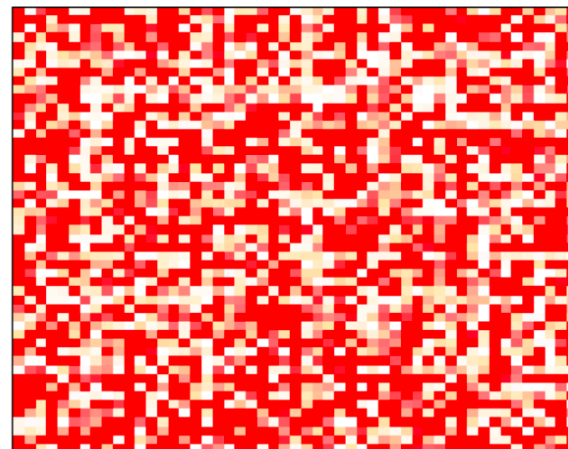
sampled matrix



Gradient descent output \mathbf{UA}



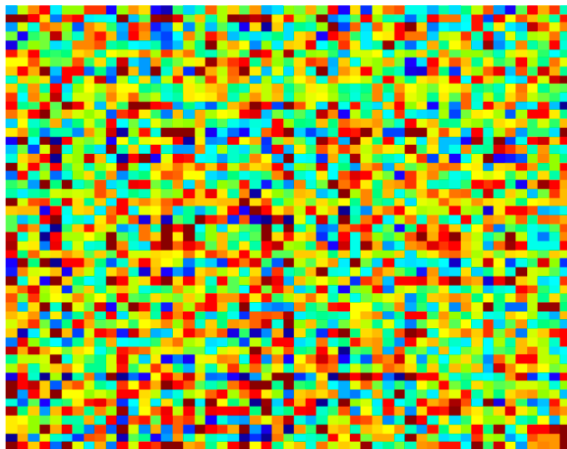
squared error $(\mathbf{X}_{ji} - (\mathbf{UA})_{ji})^2$



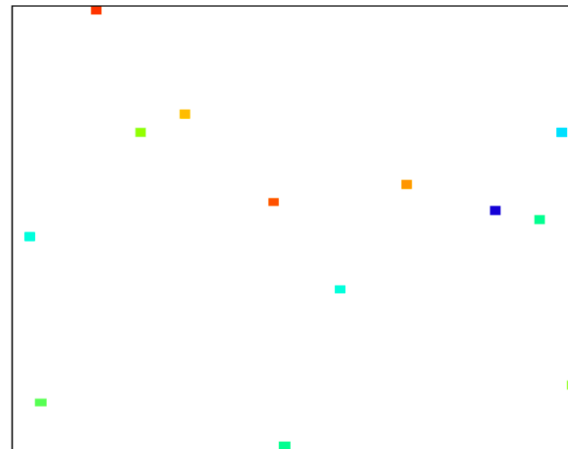
0.25% sampled

Example: 2000×2000 rank-8 random matrix

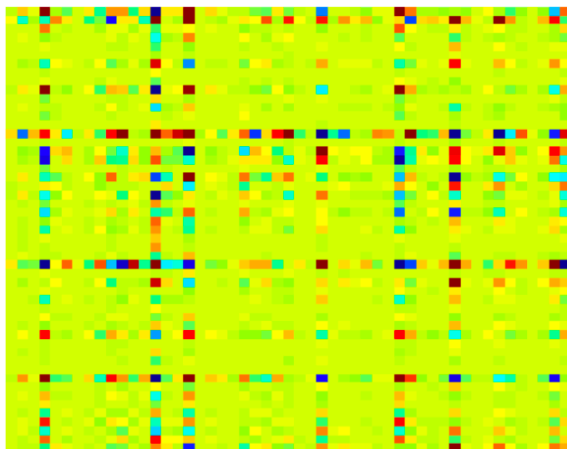
low-rank matrix \mathbf{X}



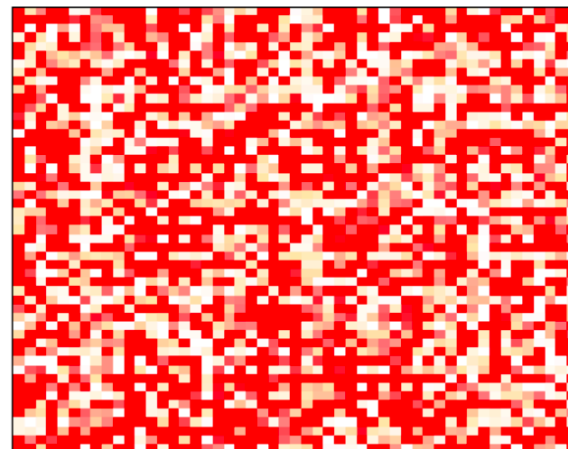
sampled matrix



Gradient descent output \mathbf{UA}



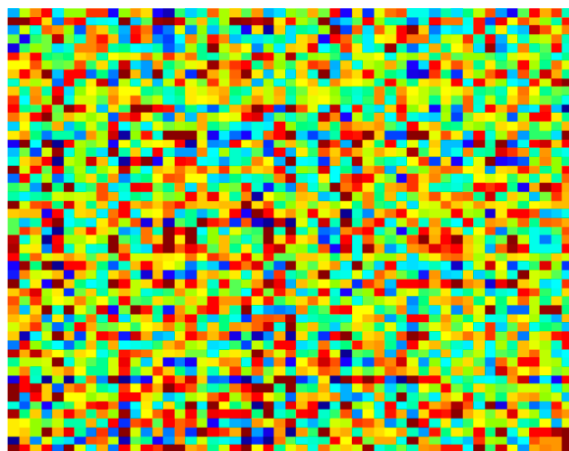
squared error $(\mathbf{X}_{ji} - (\mathbf{UA})_{ji})^2$



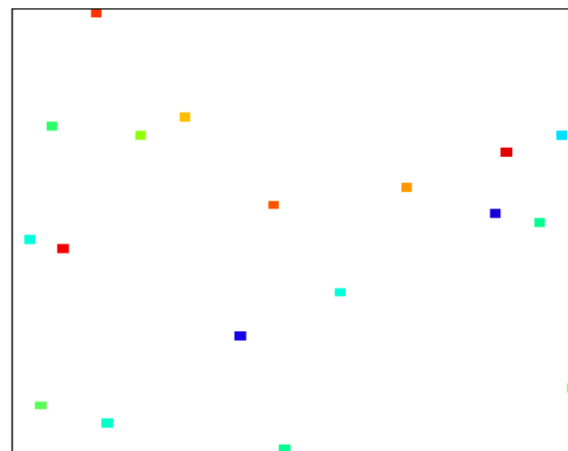
0.50% sampled

Example: 2000×2000 rank-8 random matrix

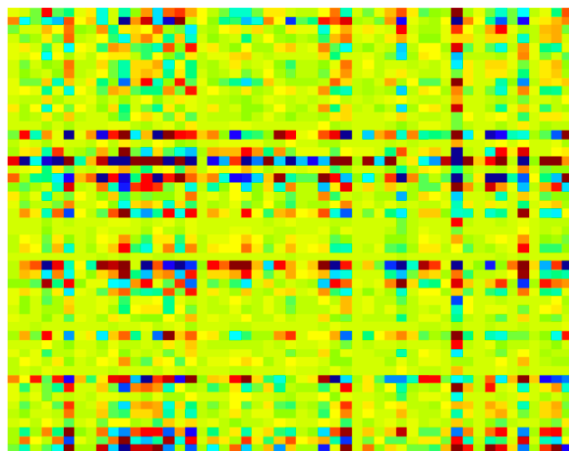
low-rank matrix \mathbf{X}



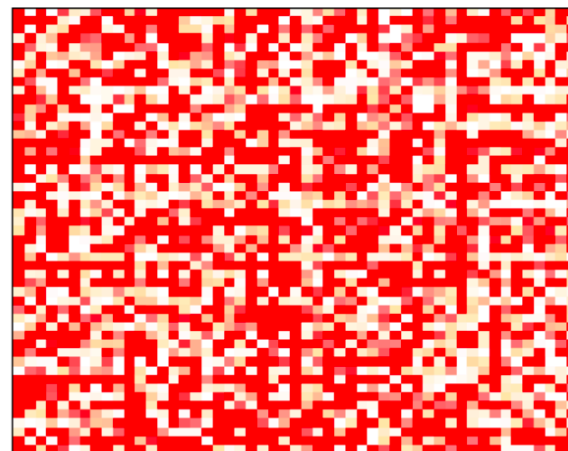
sampled matrix



Gradient descent output \mathbf{UA}



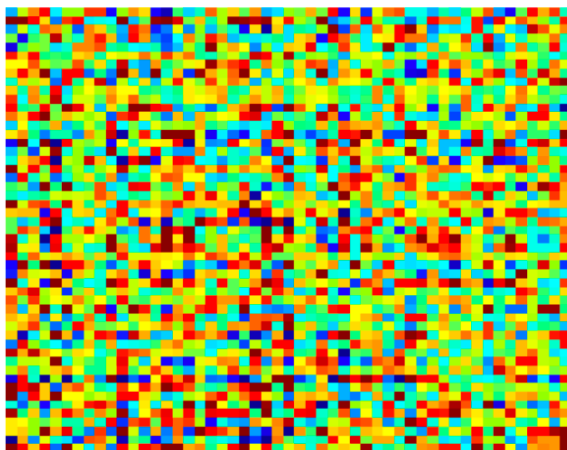
squared error $(\mathbf{X}_{ji} - (\mathbf{UA})_{ji})^2$



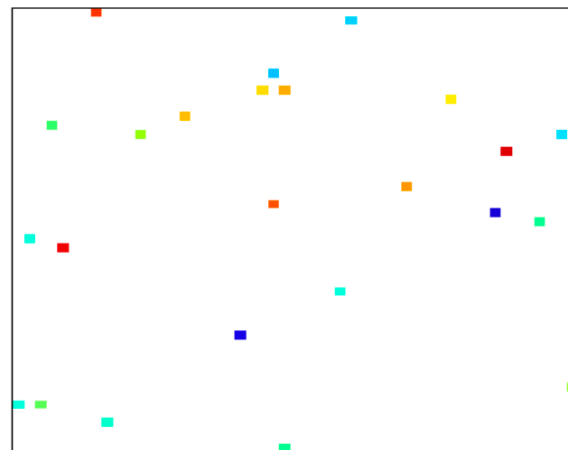
0.75% sampled

Example: 2000×2000 rank-8 random matrix

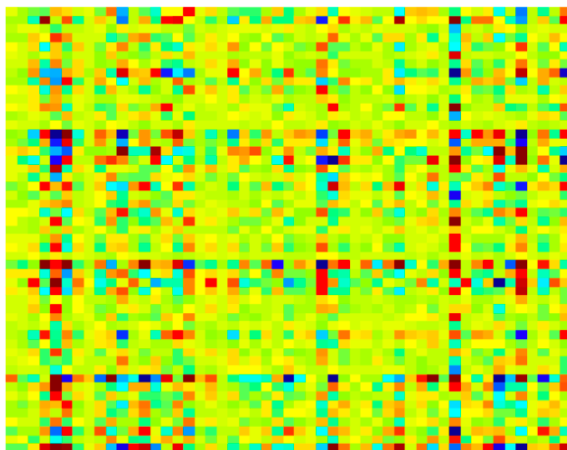
low-rank matrix \mathbf{X}



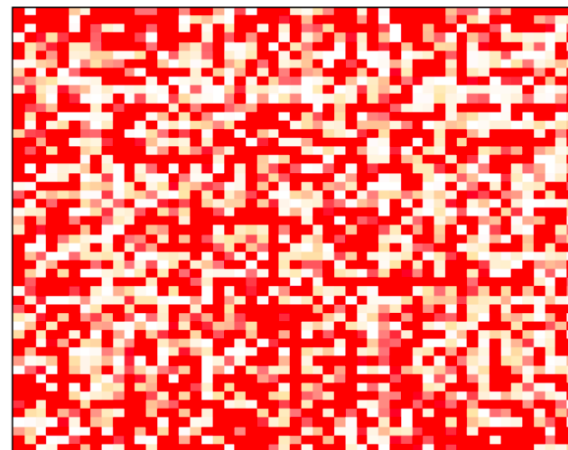
sampled matrix



Gradient descent output \mathbf{UA}



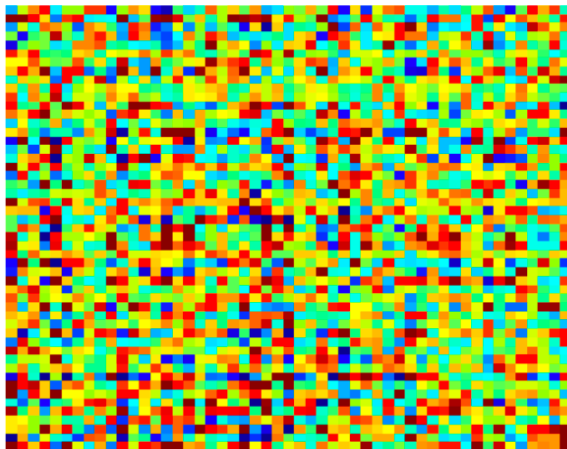
squared error $(\mathbf{X}_{ji} - (\mathbf{UA})_{ji})^2$



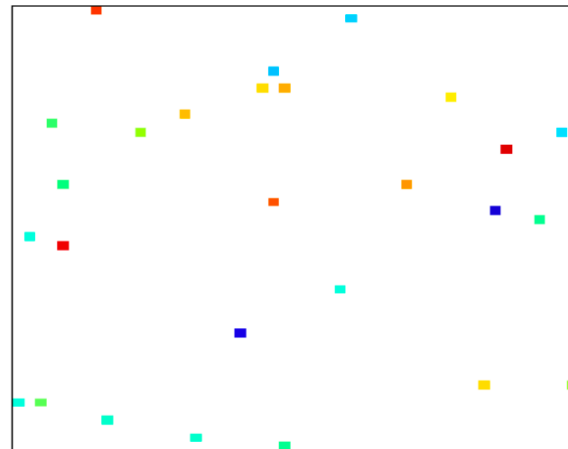
1.00% sampled

Example: 2000×2000 rank-8 random matrix

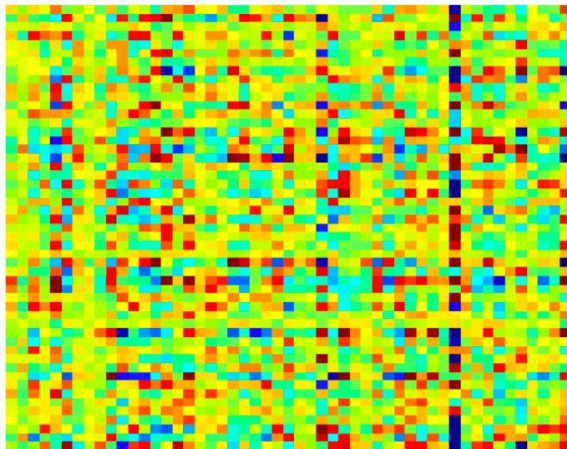
low-rank matrix \mathbf{X}



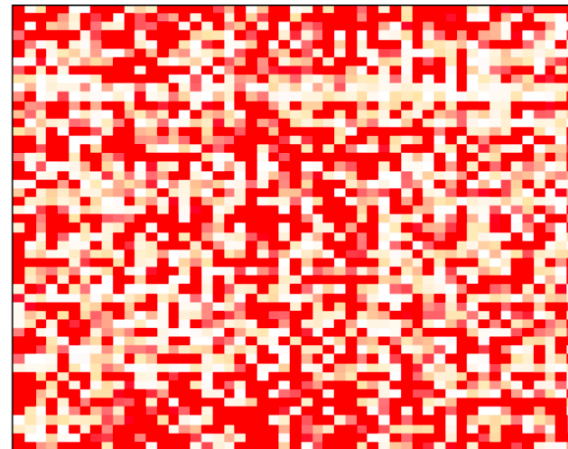
sampled matrix



Gradient descent output \mathbf{UA}



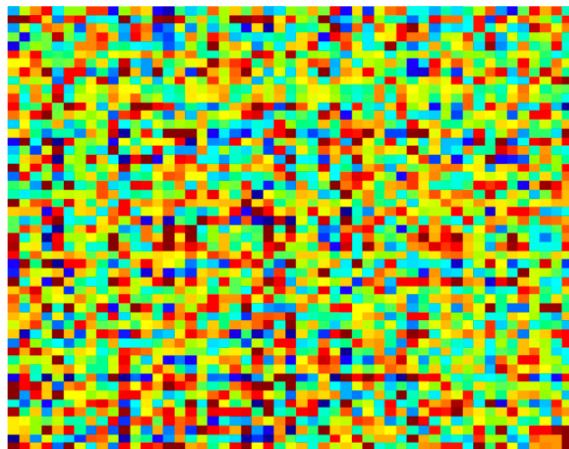
squared error $(\mathbf{X}_{ji} - (\mathbf{UA})_{ji})^2$



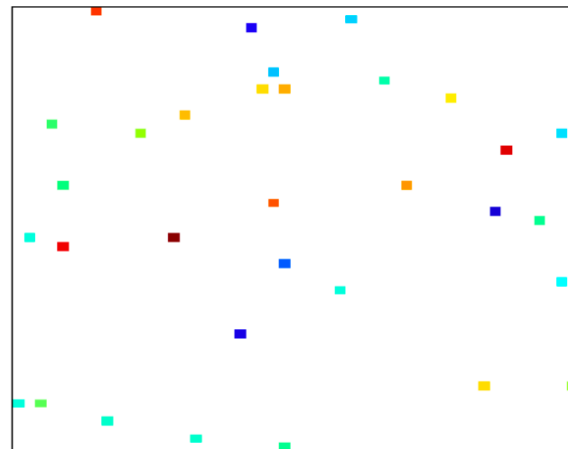
1.25% sampled

Example: 2000×2000 rank-8 random matrix

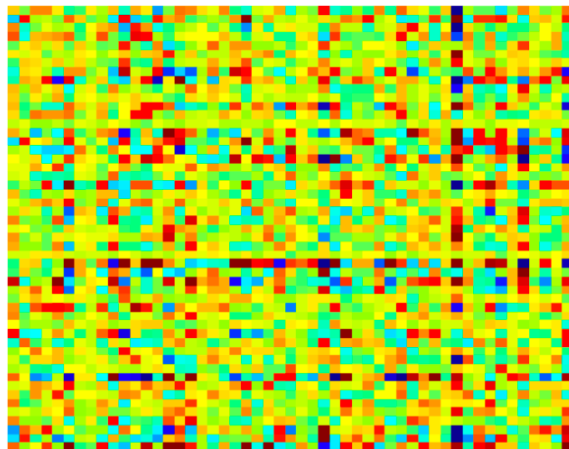
low-rank matrix \mathbf{X}



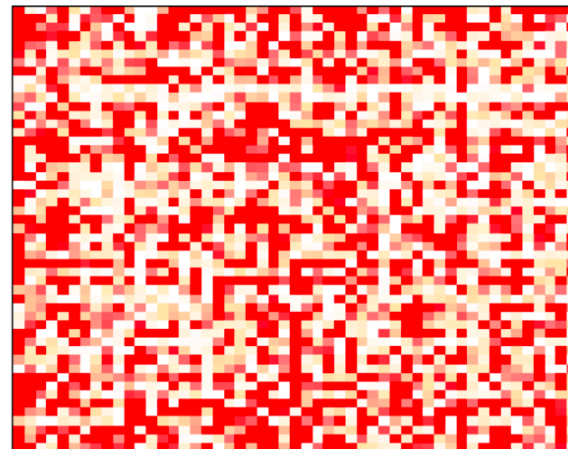
sampled matrix



Gradient descent output \mathbf{UA}



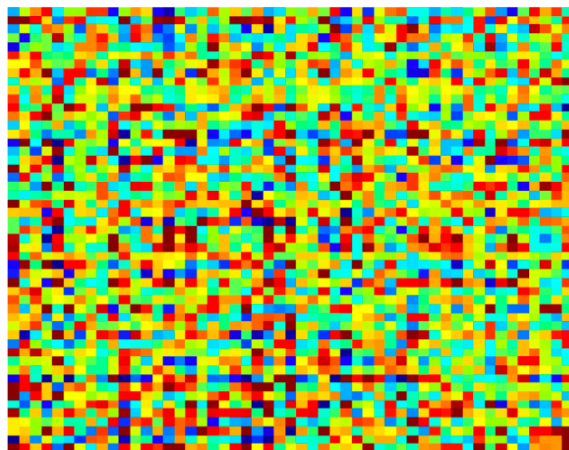
squared error $(\mathbf{X}_{ji} - (\mathbf{UA})_{ji})^2$



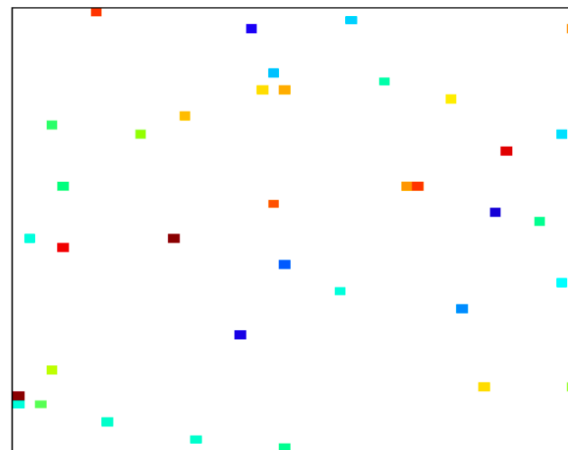
1.50% sampled

Example: 2000×2000 rank-8 random matrix

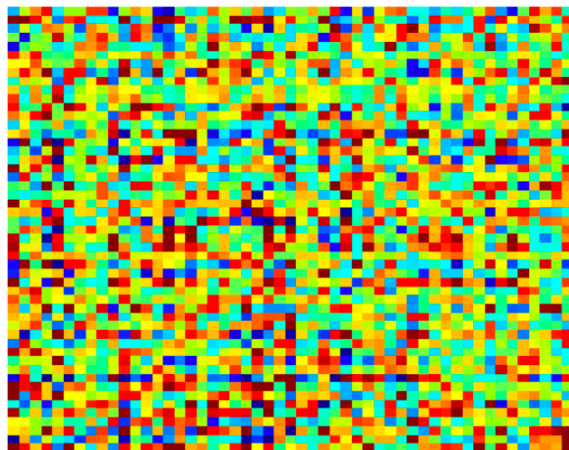
low-rank matrix \mathbf{X}



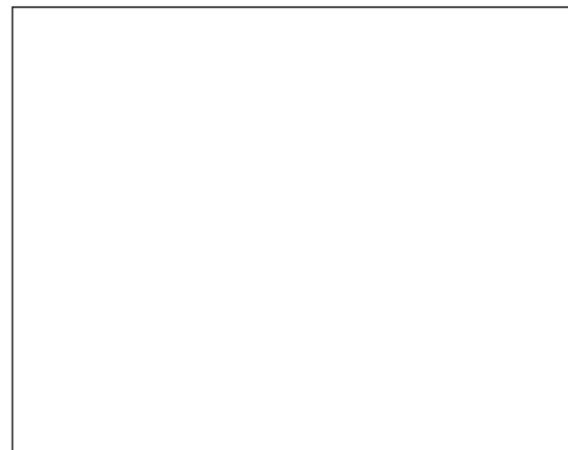
sampled matrix



Gradient descent output \mathbf{UA}



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



1.75% sampled