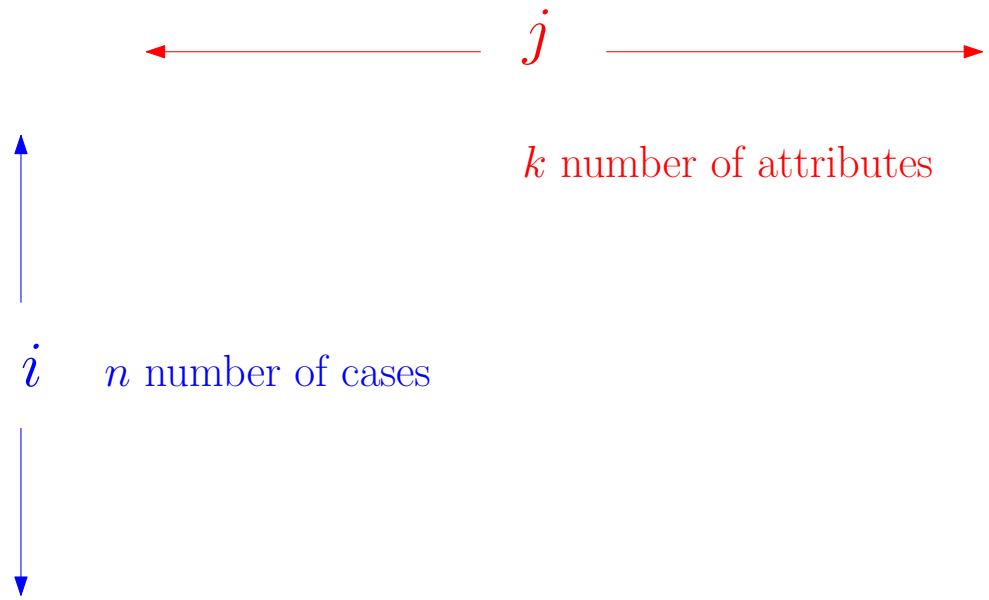
VE472 Lecture 7

Jing Liu

UM-SJTU Joint Institute

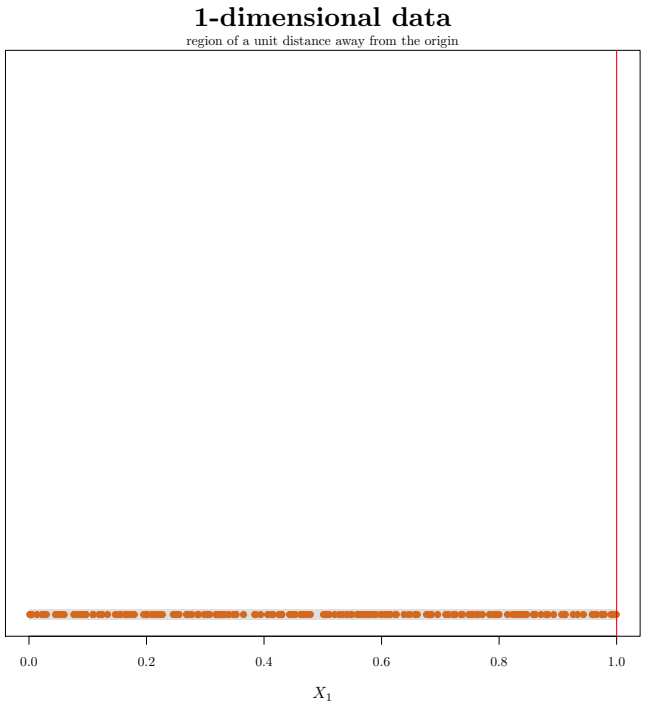
Summer

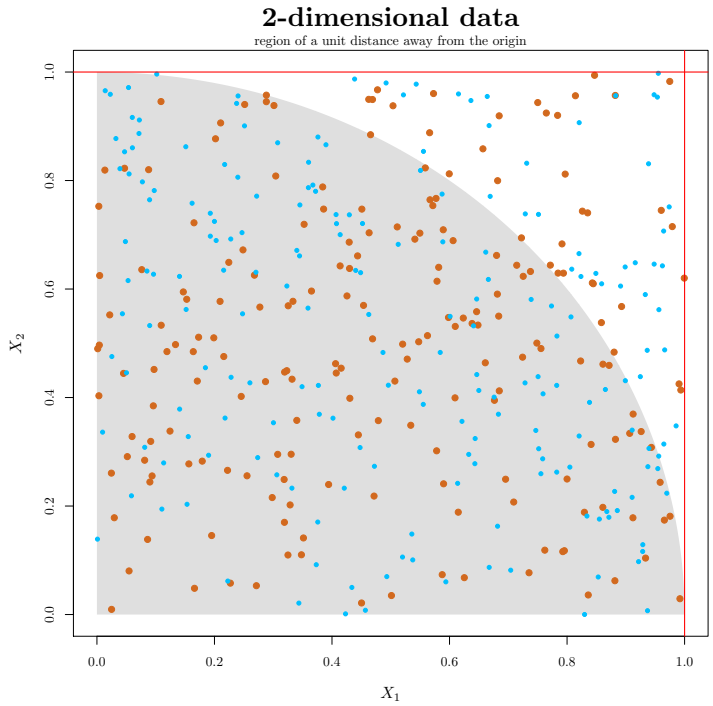
- Consider what happens if we keep increasing k while n is roughly the same.



- Recall shrinkage is a way to avoid feature selection and improve accuracy, it did not address the curse of dimensionality, that is, the growth in n often is insufficient for the growth in k.
- So we are now interested in cases where k is even larger than those in which shrinkage along is appropriate with respect to n.

- To under the curse of dimensionality, thus why n is often not big enough,





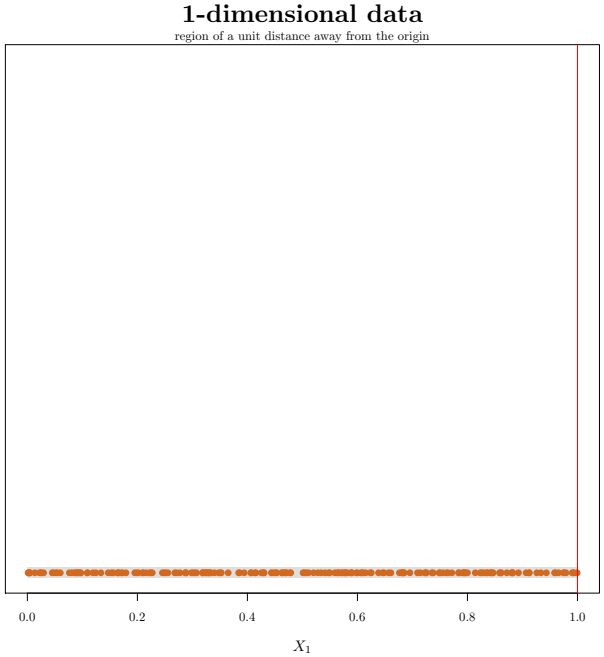
- Secondly consider n-dimensional sphere of radius r and cube of side $2r$,

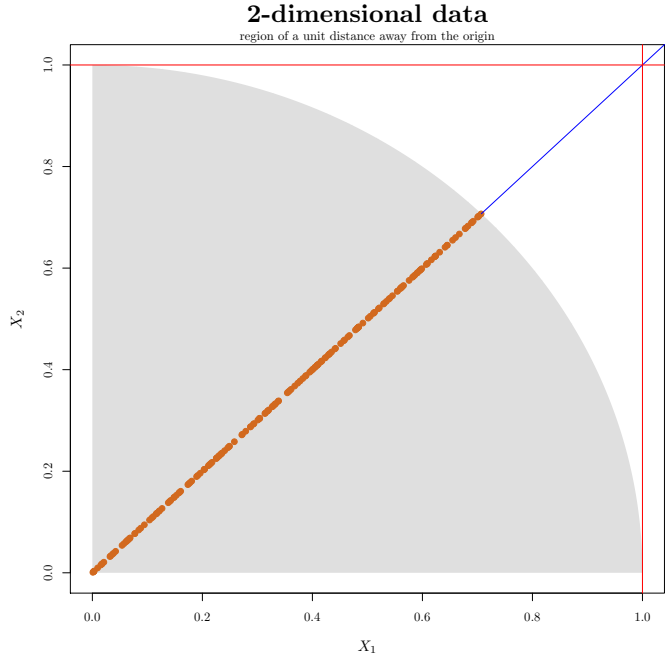
$$\frac{V_s}{V_c} = \frac{\frac{\pi^{k/2}}{\Gamma\left(\frac{k}{2}+1\right)} r^k}{(2r)^k} = \frac{\pi^{k/2}}{2^k \Gamma\left(\frac{k}{2}+1\right)} \to 0 \qquad \text{as} \quad k \to \infty$$

- Also the distance between the centre and the corners of the cube is given by

$$r\sqrt{k}$$

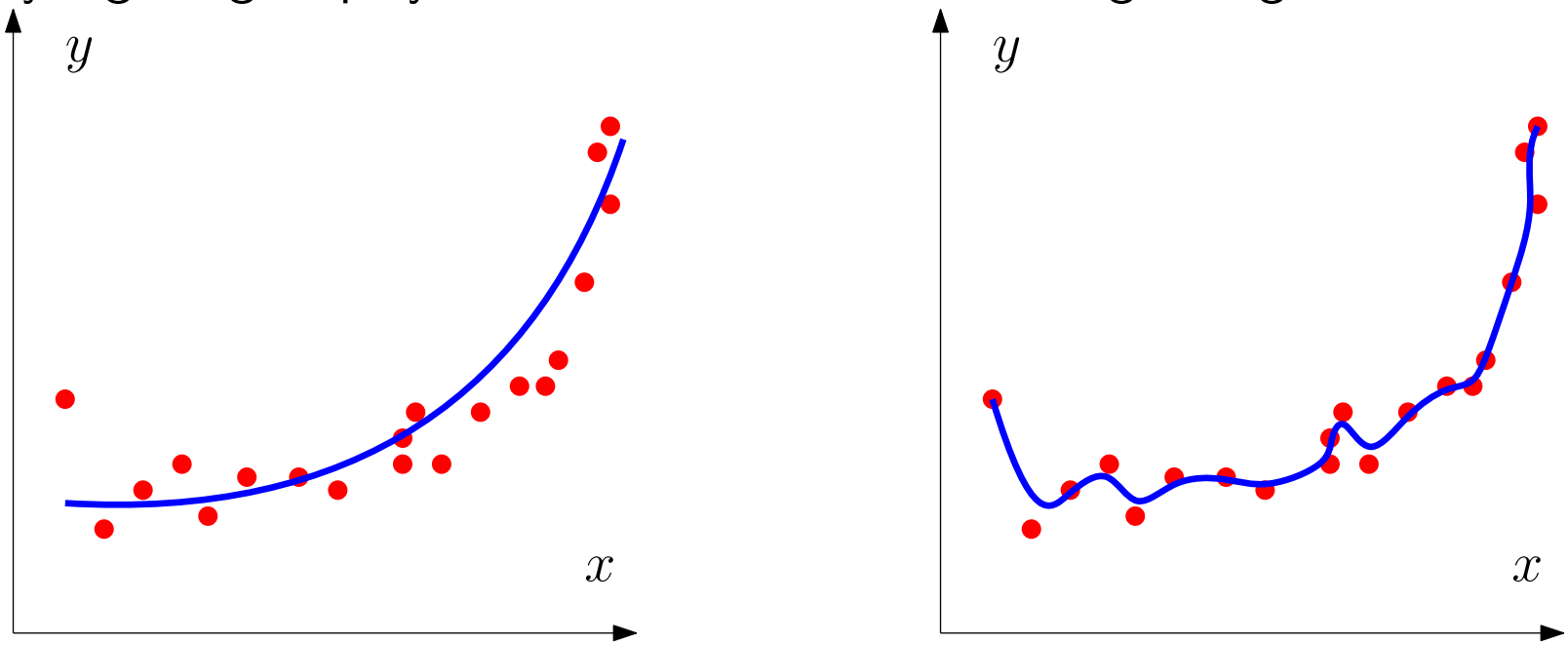
which means, if n reminds the same,





we will not even have enough data to maintain the same sampling density alone a line in a high-dimensional space, let alone the whole space.

- As k increases, there will be more and more "extra" space, and this increase in size occurs exponentially as k increases linearly. Hence unless n increases exponentially, the sample density becomes more and more sparse.
- Having a sparse sample density is prone to overfitting as in allowing for too many high degree polynomial terms when n is not big enough.



- Thus, neither shrinkage nor maximum likelihood is going to be adequate for some modern datasets where k is too big relative to n.

- Dimension reduction methods can be divided into two categories:
  - *Selection*
  - *Extraction*
- Both approaches will reduce the dataset to a dataset of $n \times p$, where $p \ll k$.

Q: How would you reduce k to p while keeping as much information as possible?

- *Selection* reduces the dimension by allowing only a fraction of the k features according to some kind of criteria, e.g. $\hat{\text{MSE}}$ by training-testing split

$$\mathbf{b}_\ell = \underset{\mathbf{b}\in\mathbb{R}^{p+1}}{\arg\min} \left\| \mathbf{y}^\dagger - \mathbf{X}_\ell^\dagger \mathbf{b} \right\|^2$$

$$\hat{\text{MSE}} = \frac{1}{m} \left\| \mathbf{y}^* - \mathbf{X}_\ell^* \mathbf{b}_\ell \right\|^2$$

- *Extraction* reduces the dimension by combing the k features to create a few new features via some linear or nonlinear procedures.

- Principal component analysis (PCA) is an extraction method, it tries to find an approximation by projecting the original data onto linear subspaces.
- Let $\mathbf{x} \in \mathbb{R}^k$ denote a random vector, i.e. the rows of our data, and

$$\mathbf{x}_c = \mathbf{x} - \boldsymbol{\mu} \qquad \text{where} \quad \boldsymbol{\mu} = \mathbb{E}[\mathbf{x}]$$

then the pth principal subspace is given by

$$\ell_p = \arg\min_{\ell \in \mathcal{L}_p} \left\{ \mathbb{E}\left[\min_{\mathbf{y} \in \ell} \|\mathbf{x}_c - \mathbf{y}\|^2 \right] \right\}$$

where $\mathcal{L}_p$ denote all p-dimensional linear subspaces of $\mathbb{R}^k$.

Q: What does this double optimisation demand us to find?

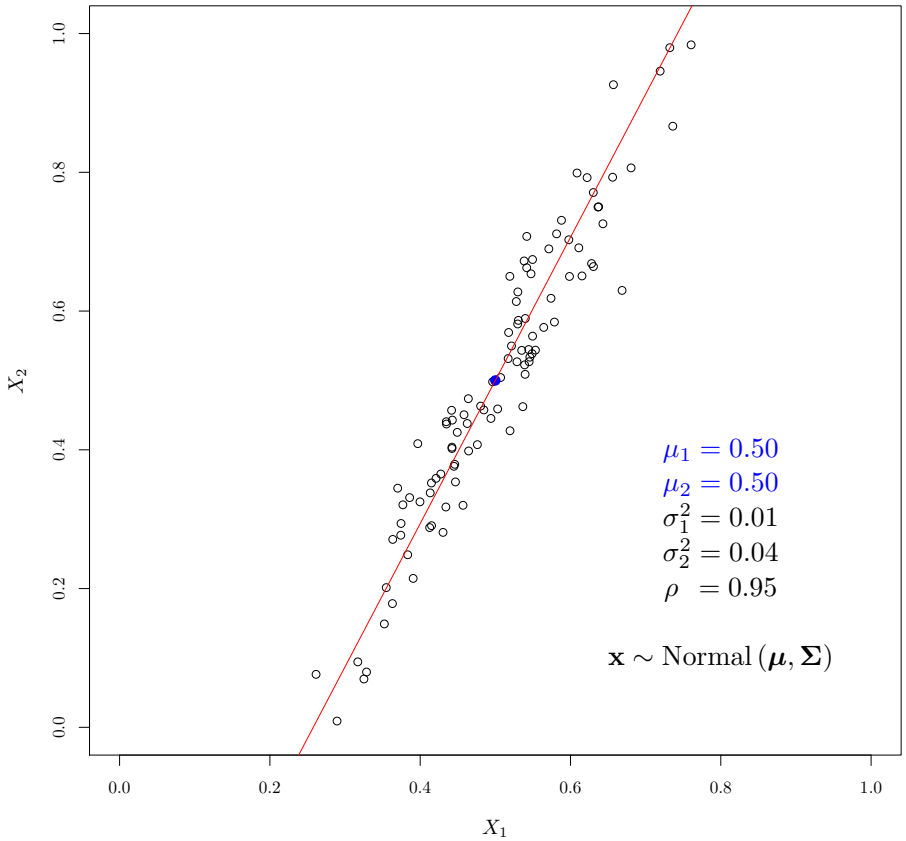
- Using the relation between projection and least squares, we have

$$\arg\min_{\mathbf{y} \in \ell} \|\mathbf{x}_c - \mathbf{y}\|^2 = \operatorname{proj}_\ell \mathbf{x}_c \implies \min_{\mathbf{y} \in \ell} \|\mathbf{x}_c - \mathbf{y}\|^2 = \|\mathbf{x}_c - \operatorname{proj}_\ell \mathbf{x}_c\|^2$$

- Once the subspace ℓ_p is determined, then the reduced $\mathbf{x}$ is given by

$$T_p(\mathbf{x}) = \boldsymbol{\mu} + \text{proj}_{\ell_p} \mathbf{x}_c$$

Q: How can we determine ℓ_p?



Theorem 0.1

Let $\mathbf{\Sigma} \in \mathbb{R}^{k \times k}$ *be the variance-covariance matrix of the random vector* $\mathbf{x}$*, i.e.*

$$\mathbf{\Sigma} = \mathbb{E}\left[(\mathbf{x} - \boldsymbol{\mu})(\mathbf{x} - \boldsymbol{\mu})^{\mathrm{T}}\right]$$

and $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_k$ *be the ordered eigenvalues of* $\mathbf{\Sigma}$ *while* $\mathbf{q}_1, \mathbf{q}_2, \cdots, \mathbf{q}_k$ *be the corresponding orthonormal eigenvectors, then the* p*th principal subspace* ℓ_p *is the subspace spanned by the first* p *eigenvectors* $\mathbf{q}_1$, $\mathbf{q}_2$, ..., $\mathbf{q}_p$, *and*

$$T_p(\mathbf{x}) = \boldsymbol{\mu} + \sum_{j=1}^{p} \beta_j \mathbf{q}_j \qquad \text{where} \quad \beta_j = \mathbf{x}_c^{\mathrm{T}} \mathbf{q}_j$$

Furthermore, the expected quadratic cost of using $T_p(\mathbf{x})$ *is given by*

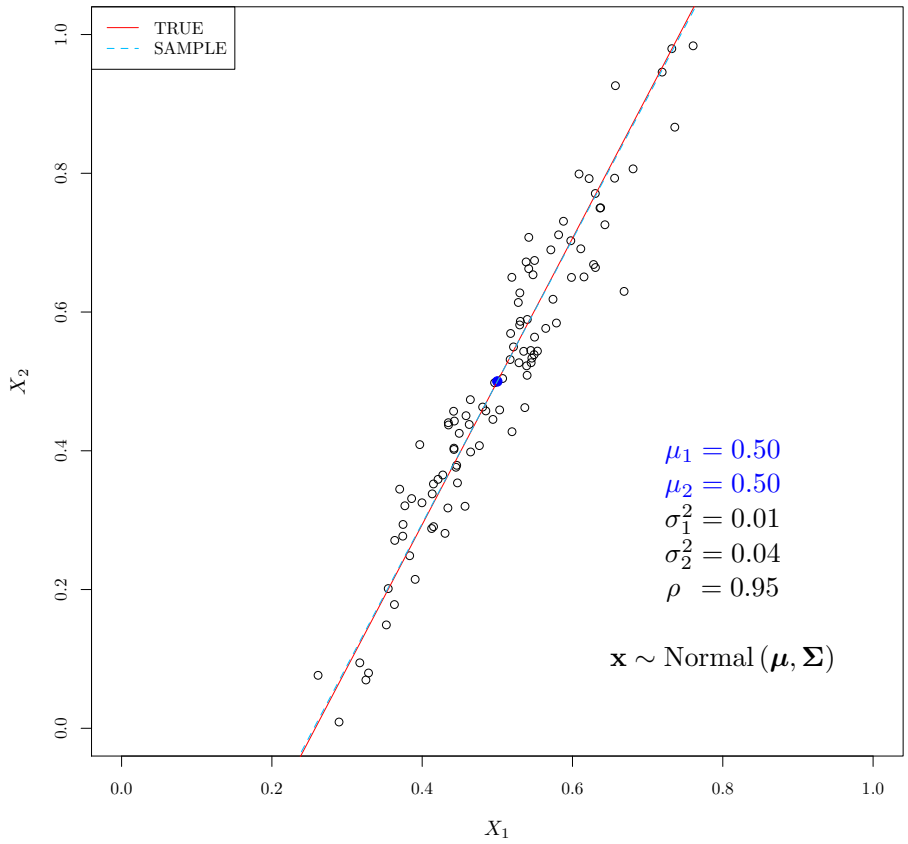
$$\mathbb{E}\left[\|\mathbf{x} - T_p(\mathbf{x})\|^2\right] = \sum_{j=p+1}^{k} \lambda_j$$

Q: What does the above theorem mean?

- Of course, the true variance-covariance matrix of $\mathbf{x}$ is not available,

$$\hat{\mathbf{\Sigma}} = \frac{1}{n}\sum_{i=1}^{n} \mathbf{x}_{ic}\mathbf{x}_{ic}^{\mathrm{T}}$$

is used as the covariance matrix in practice.

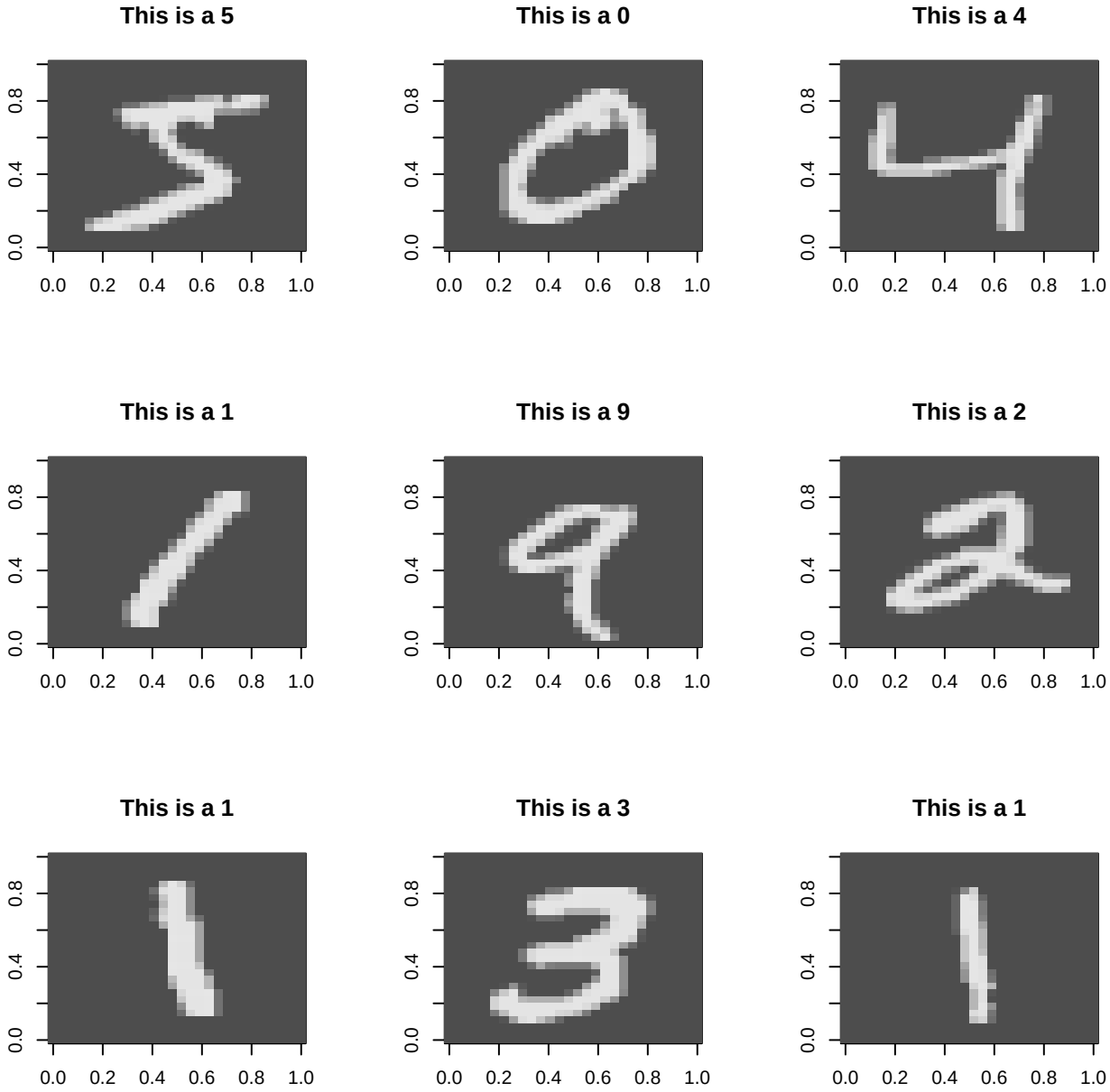


- MNIST is a famous dataset, it is often credited as one of the 1st datasets to prove the effectiveness of neural networks, it has 60,000 rows and 785 cols.

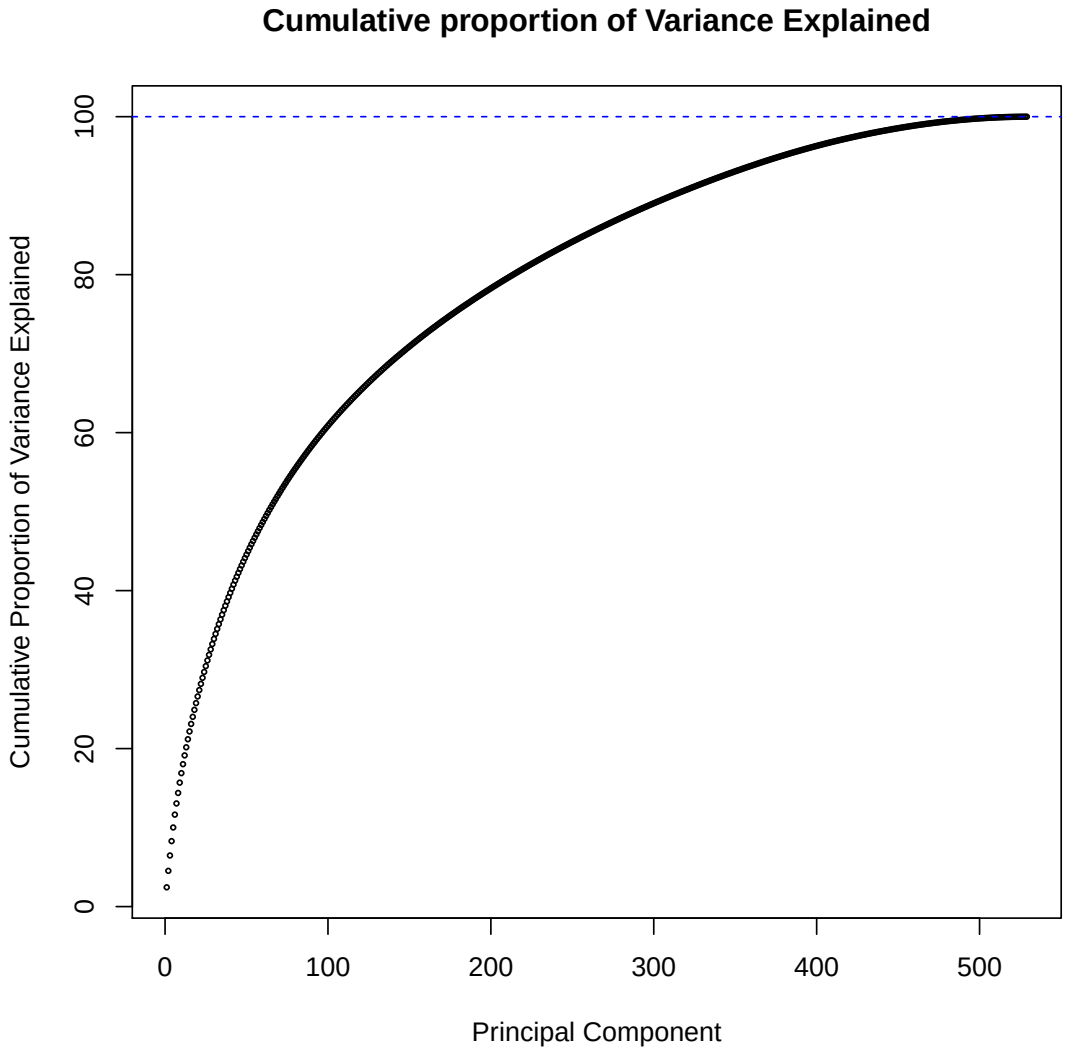
```
> mnist_data[1:6,1:13]
```

	V1	V2	V3	V4	V5	V6	V7	V8	V9	V10	V11	V12	V13
1:	5	0	0	0	0	0	0	0	0	0	0	0	0
2:	0	0	0	0	0	0	0	0	0	0	0	0	0
3:	4	0	0	0	0	0	0	0	0	0	0	0	0
4:	1	0	0	0	0	0	0	0	0	0	0	0	0
5:	9	0	0	0	0	0	0	0	0	0	0	0	0
6:	2	0	0	0	0	0	0	0	0	0	0	0	0

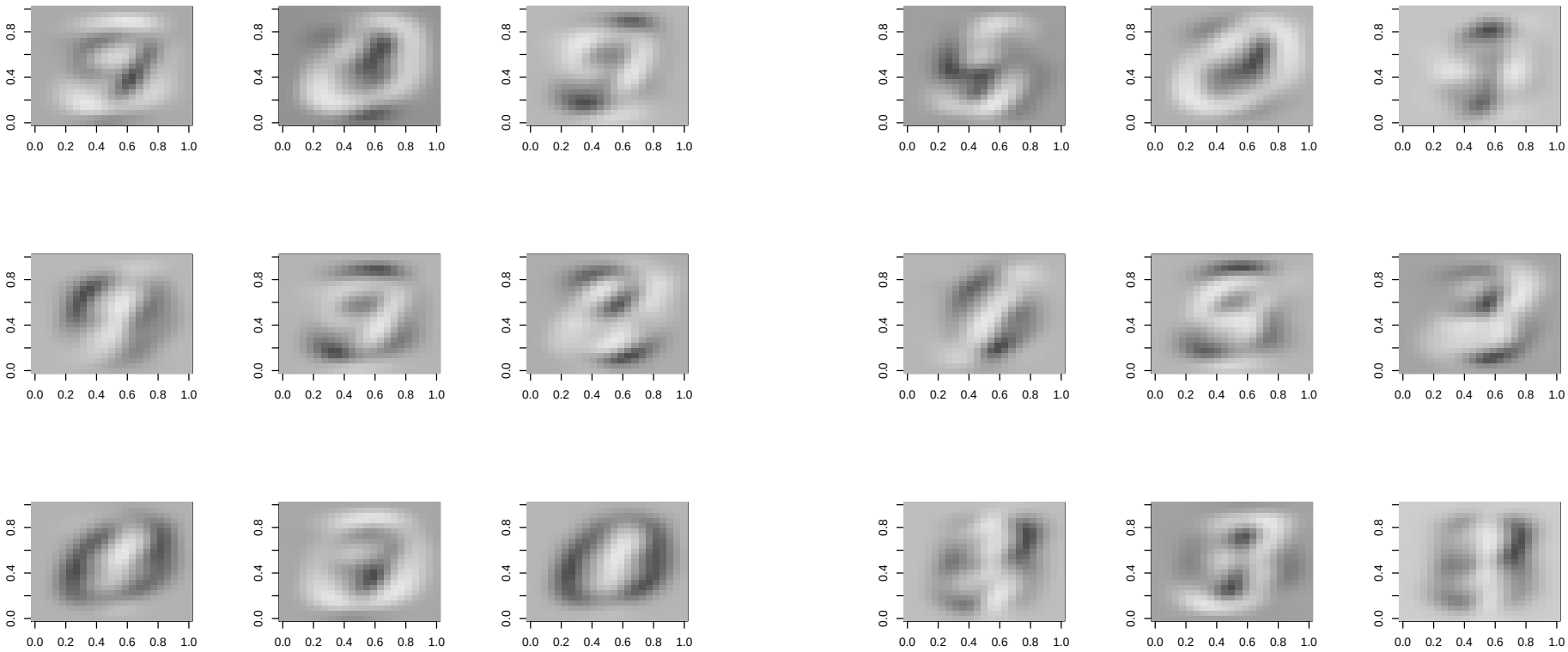
- Each row represents an digital image of a handwritten digit, it is 28 pixels in height and 28 pixels in width, for a total of 784 pixels in total. Each pixel has a single pixel-value associated with it, indicating the lightness/darkness of that pixel, with higher numbers meaning darker. The pixel-value is an integer between 0 and 255, inclusive. The first column is the “label" for the digit, the rest of the cols contain the pixel-values of the associated image.


This is a 5
This is a 0
This is a 4
This is a 1
This is a 9
This is a 2
This is a 1
This is a 3
This is a 1
0.0
0.4
0.8
0.0
0.2
0.4
0.6
0.8
1.0

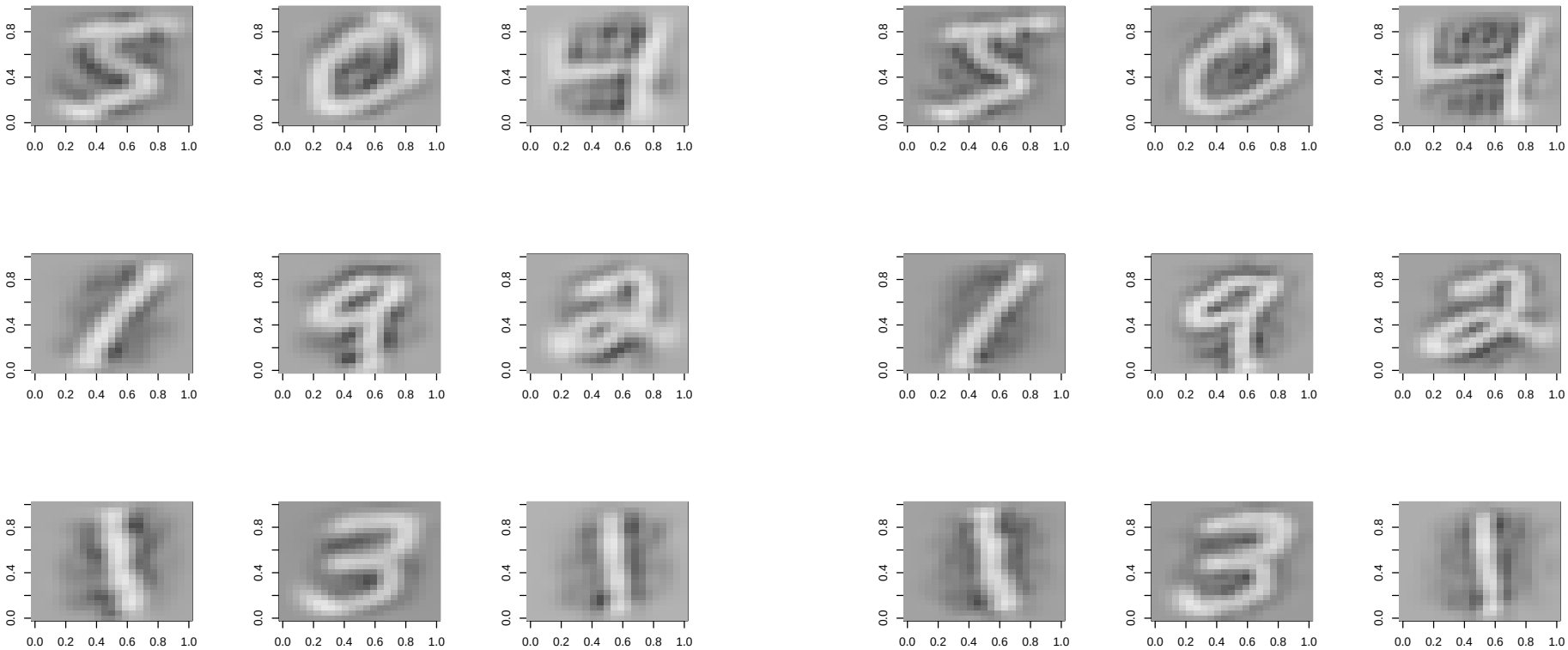
- With p around 100, almost 60% of the variance is explained.



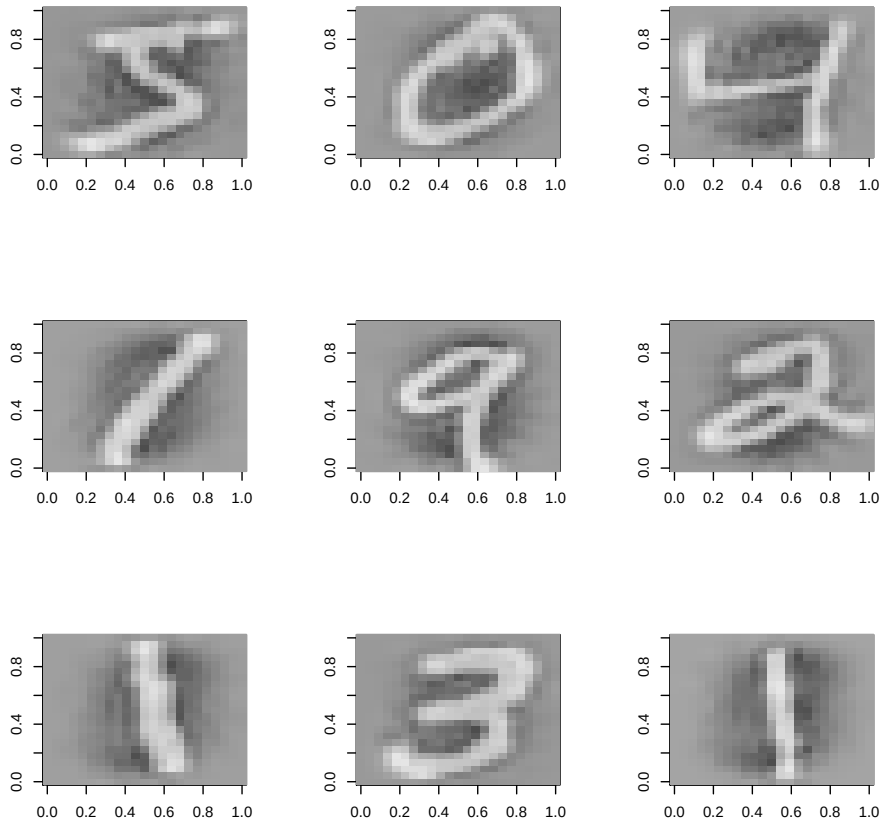
- Reducing reducing $k = 784$ to $p = 3/p = 10$ is clearly too much in this case.



- However, setting $p = 50$ or $p = 100$, human eyes can detect the digits easily.



- With $p = 200$, roughly 80% of the variance is explained.



while greatly reduces the chance of overfitting.

- From linear algebra point of view, PCA uses the spectral decomposition

$$\mathbf{A} = \mathbf{P}\mathbf{D}\mathbf{P}^{-1}$$

on the sample covariance matrix, which is symmetric positive semi-definite

$$\begin{aligned}\hat{\mathbf{\Sigma}} = \frac{1}{n}\sum_{i=1}^{n}\mathbf{x}_{ic}\mathbf{x}_{ic}^{\mathrm{T}} = \frac{1}{n}\mathbf{X}_c\mathbf{X}_c^{\mathrm{T}} &= \frac{1}{n}\mathbf{Q}\mathbf{D}\mathbf{Q}^{\mathrm{T}} \\ &= \frac{1}{n}\mathbf{Q}\mathbf{D}^{1/2}\mathbf{Q}^{\mathrm{T}}\mathbf{Q}\mathbf{D}^{1/2}\mathbf{Q}^{\mathrm{T}} \\ &= \frac{1}{n}\left(\mathbf{Q}\mathbf{D}^{1/2}\mathbf{Q}^{\mathrm{T}}\right)\left(\mathbf{Q}\mathbf{D}^{1/2}\mathbf{Q}^{\mathrm{T}}\right)^{\mathrm{T}}\end{aligned}$$

instead of using all k eigenvalues/eigenvectors, we use the first p of them

$$\frac{1}{n}\left(\mathbf{Q}_p\mathbf{D}_p^{1/2}\mathbf{Q}_p^{\mathrm{T}}\right)\left(\mathbf{Q}_p\mathbf{D}_p^{1/2}\mathbf{Q}_p^{\mathrm{T}}\right)^{\mathrm{T}}$$

- Singular value decomposition (SVD)

$$\mathbf{A} = \mathbf{U}\mathbf{D}\mathbf{V}^{\mathrm{T}}$$

can also be used as a extraction method in a similar fashion

$$\mathbf{X}\mathbf{X}^{\mathrm{T}} = \mathbf{U}\mathbf{D}\mathbf{V}^{\mathrm{T}}\left(\mathbf{U}\mathbf{D}\mathbf{V}^{\mathrm{T}}\right)^{\mathrm{T}} = \mathbf{U}\mathbf{D}^2\mathbf{U}^{\mathrm{T}}$$

instead of using all k singular values, we use the first p of them

$$\mathbf{U}_p\mathbf{D}_p\mathbf{V}_p^{\mathrm{T}}\mathbf{V}_p\mathbf{D}_p\mathbf{U}_p^{\mathrm{T}}$$

- If we work with $\mathbf{X}_c$, then SVD is equivalent to PCA, but more stable since

$$\mathbf{X}\mathbf{X}^{\mathrm{T}}$$

can be numerically unstable to compute, SVD avoids it by working on $\mathbf{X}_c$.

- If the data is sparse, then SVD is preferred since we can exploit sparsity.

- Another simple but common extraction method is known as

  multidimensional scaling

- The idea is to find a linear transformation,

$$T\colon \mathbb{R}^k \to \mathbb{R}^p; \qquad \mathbf{z}_i = T(\mathbf{x}_i)$$

  that preserves as much as possible for some kind of pairwise distances.

- For example, if we define the following cost function

$$C = \sum_{j,k} \left(\|\mathbf{x}_j - \mathbf{x}_k\|^2 - \|\mathbf{z}_j - \mathbf{z}_k\|^2 \right)$$

  then multidimensional scaling find the linear transformation minimises C.

- In this case, the solution will coincide with PCA. however, if some other cost function is used to measure the distortion, it differs from PCA.