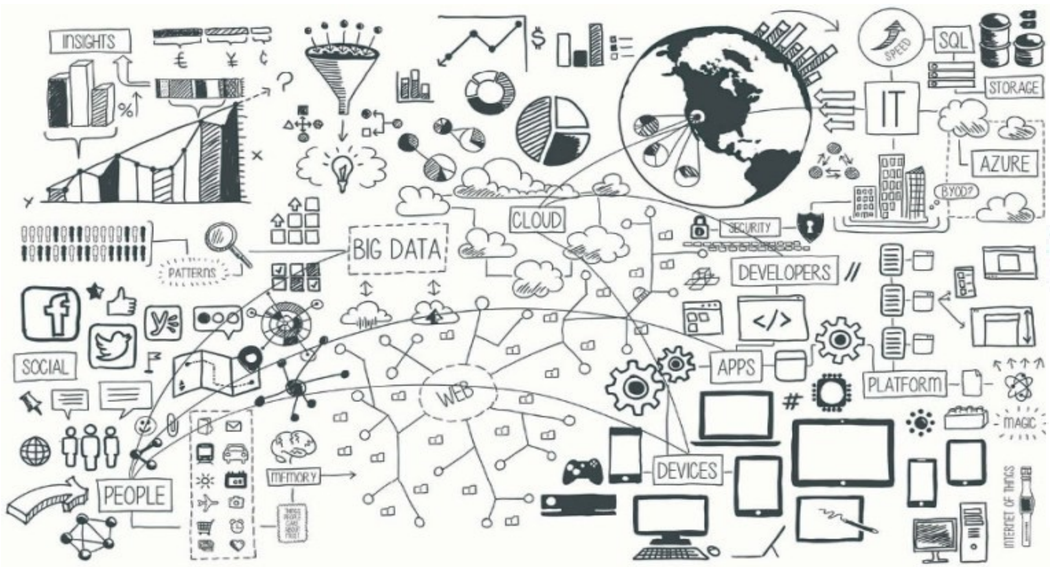
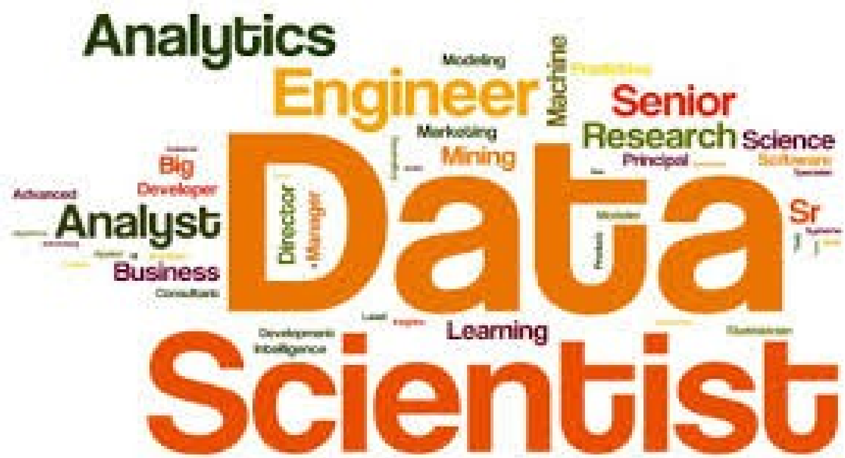
Estadística (M1965)

REDUCCIÓN DE LA DIMENSIÓN





Jesús Fernández

Grupo de Meteorología

Univ. de Cantabria – CSIC
MACC / IFCA

Master Universitario Oficial **Data Science**
con el apoyo del

v20200107

Propiedades básicas de un conjunto de variables predictoras

¿Cómo tratar con la …

Proximidad **… irrelevancia y …**

Es el grado en el que un conjunto de variables predictoras es capaz de explicar la variable a predecir (predictando). Variables próximas dan lugar a predicciones más robustas.

Multicolinealidad **… la redundancia?**

Alta correlación entre dos o más variables predictoras. Puede afectar negativamente la capacidad predictiva del modelo.

Dimensionalidad

Número de predictores. Un número alto de predictores puede dar lugar a modelos sobre-ajustados con menor capacidad de generalización.

Aproximaciones al problema de la preparación de un conjunto "óptimo" de predictores

Reducción de la dimensión
Selección de las primeras componentes principales. Se encuentra un conjunto reducido de nuevas variables predictoras que explican gran parte de la variabilidad original del conjunto completo.

+ Menos variables; se evita el sobreajuste

+ Variables ortogonales, eliminan el problema de la colinealidad

– Mezcla de variables: pérdida de interpretabilidad de los resultados

Aproximaciones al problema de la preparación de un conjunto "óptimo" de predictores

Reducción de la dimensión

Selección de las primeras componentes principales. Se encuentra un conjunto reducido de nuevas variables predictoras que explican gran parte de la variabilidad original del conjunto completo.

\+ Menos variables; se evita el sobreajuste

\+ Variables ortogonales, eliminan el problema de la colinealidad

– Mezcla de variables: pérdida de interpretabilidad de los resultados

Otras ventajas

\+ Facilita la visualización

\+ Comprime los datos

\+ Elimina ruido

\+ Mejora la convergencia de los métodos iterativos

Dimension Reduction

1.- Linear techniques (projective): look for a linear transformation between the d- and r-dimensional spaces

$$z_i = Px_i \quad P \in \mathbb{R}^{r\times d}$$
$$x_i \in \mathbb{R}^d,\ i = 1, \ldots, n$$
$$z_i \in \mathbb{R}^r,\ i = 1, \ldots, n$$

Examples: Principal Components Analysis (PCA), Linear Discriminant Analysis (LDA), etc

2.- Non-linear techniques (manifold): try to model the non-linear manifold containing the data.
Examples: Multidimensional scaling (MDS), Isomap, Locally linear embedding (LLE), Stochastic Neighbor embedding (SNE), etc

Objective: Given n samples from a d-dimensional space, the objective is to project them in a r-dimensional space, with r<d, optimizing some especific properties of the data.

Principal Components Analysis (PCA)

PCA obtains the ***p*** dimensions/directions **maximizing the variance** of the projected data. Once the new basis of the p-dimensional space is obtained (Empirical Orthogonal Functions, **EOFs**) the reduction of the dimension is obtained chosing the first ***r*** principal components (**PCs**) or, equivalently, the ***r*** directions/dimensions explaining a predefined percentage of the total variance.

	a	b	c	d	e	f	g	h	i	j	k	l	m
0	0.0	0.0	12.0	13.0	5.0	0.0	0.0	0.0	0.0	0.0	11.0	16.0	9.0
1	0.0	0.0	4.0	15.0	12.0	0.0	0.0	0.0	0.0	3.0	16.0	15.0	14.0
2	0.0	7.0	15.0	13.0	1.0	0.0	0.0	0.0	8.0	13.0	6.0	15.0	4.0
3	0.0	0.0	1.0	11.0	0.0	0.0	0.0	0.0	0.0	0.0	7.0	8.0	0.0
4	0.0	12.0	10.0	0.0	0.0	0.0	0.0	0.0	0.0	14.0	16.0	16.0	14.0
5	0.0	0.0	12.0	13.0	0.0	0.0	0.0	0.0	0.0	5.0	16.0	8.0	0.0
6	0.0	7.0	8.0	13.0	16.0	15.0	1.0	0.0	0.0	7.0	7.0	4.0	11.0
7	0.0	9.0	14.0	8.0	1.0	0.0	0.0	0.0	0.0	12.0	14.0	14.0	12.0
8	0.0	11.0	12.0	0.0	0.0	0.0	0.0	0.0	2.0	16.0	16.0	16.0	13.0
9	0.0	1.0	9.0	15.0	11.0	0.0	0.0	0.0	0.0	11.0	16.0	8.0	14.0
10	0.0	0.0	0.0	14.0	13.0	1.0	0.0	0.0	0.0	0.0	5.0	16.0	16.0
11	0.0	5.0	12.0	1.0	0.0	0.0	0.0	0.0	0.0	15.0	14.0	7.0	0.0
12	2.0	9.0	15.0	14.0	9.0	3.0	0.0	0.0	4.0	13.0	8.0	9.0	16.0
13	0.0	0.0	8.0	15.0	1.0	0.0	0.0	0.0	0.0	1.0	14.0	13.0	1.0
14	5.0	12.0	13.0	16.0	16.0	2.0	0.0	0.0	11.0	16.0	15.0	8.0	4.0
15	0.0	0.0	8.0	15.0	1.0	0.0	0.0	0.0	0.0	0.0	12.0	14.0	0.0
16	0.0	1.0	8.0	15.0	10.0	0.0	0.0	0.0	3.0	13.0	15.0	14.0	14.0
17	0.0	10.0	7.0	13.0	9.0	0.0	0.0	0.0	0.0	9.0	10.0	12.0	15.0
18	0.0	6.0	14.0	4.0	0.0	0.0	0.0	0.0	0.0	11.0	16.0	10.0	0.0

$$\underset{n \times d}{X}$$

$$\xRightarrow{} \quad \underset{d \times r}{P_r}$$

	p	q	r
0	6.714282	4.263453	-3.698543
1	10.808431	-3.605318	-5.503636
2	-5.706837	1.614261	4.903340
3	7.967174	13.080329	0.804771
4	-13.493403	-2.883923	-9.851133
5	0.428223	11.200599	4.443526
6	7.754553	-10.512425	8.192590
7	-8.213388	-1.709213	-4.202378
8	-15.188940	-2.985254	-7.907185
9	3.892792	-6.433339	0.217544
10	16.222026	-4.871032	-7.677208
11	-12.432569	7.402058	3.093797
12	-1.928984	-11.331136	2.937210
13	5.864160	11.300092	0.921414
14	-3.750759	-10.217692	14.785712
15	6.721941	12.206578	1.011207
16	2.723118	-6.903958	-1.785841
17	1.891926	-7.880966	-3.138556
18	-10.273747	8.266886	2.453369

$$\underset{n \times r}{Z_r} = X P_r$$

we consider centered variables
X column means = 0

Principal Components Analysis (PCA)

PCA obtains the ***p*** dimensions/directions **maximizing the variance** of the projected data. Once the new basis of the p-dimensional space is obtained (Empirical Orthogonal Functions, **EOFs**) the reduction of the dimension is obtained chosing the first ***r*** principal components (**PCs**) or, equivalently, the ***r*** directions/dimensions explaining a predefined percentage of the total variance.

Obtaining the PCs

Since we are considering centered variables, the covariance matrix, can be expressed as:

$$\mathcal{C} = \frac{1}{n} X^t X \qquad \mathcal{C} \in \mathbb{R}^{d \times d}$$

Principal Components Analysis (PCA)

PCA obtains the ***p*** dimensions/directions **maximizing the variance** of the projected data. Once the new basis of the p-dimensional space is obtained (Empirical Orthogonal Functions, **EOFs**) the reduction of the dimension is obtained chosing the first ***r*** principal components (**PCs**) or, equivalently, the ***r*** directions/dimensions explaining a predefined percentage of the total variance.

Obtaining the PCs

Since we are considering centered variables, the covariance matrix, can be expressed as:

$$\mathcal{C} = \frac{1}{n} X^t X \qquad \mathcal{C} \in \mathbb{R}^{d \times d}$$

The diagonal of this matrix contains the sample variance of each variable. Therefore, the total variance is

$$TV = \sum_{j=1}^{d} s_j^2 = \mathrm{tr}(\mathcal{C})$$

Principal Components Analysis (PCA)

PCA obtains the ***p*** dimensions/directions **maximizing the variance** of the projected data. Once the new basis of the p-dimensional space is obtained (Empirical Orthogonal Functions, **EOFs**) the reduction of the dimension is obtained chosing the first ***r*** principal components (**PCs**) or, equivalently, the ***r*** directions/dimensions explaining a predefined percentage of the total variance.

Obtaining the PCs

Since we are considering centered variables, the covariance matrix, can be expressed as:

$$\mathcal{C} = \frac{1}{n} X^t X \qquad\qquad \mathcal{C} \in \mathbb{R}^{d \times d}$$

Off-diagonal terms of this matrix contain the sample covariances. Therefore, $\mathcal{C}$ is a square symmetric matrix. That is, its eigenvectors can be chosen orthonormal and its eigenvalues are real numbers.

Moreover, it is positive semidefinite. That is, its eigenvalues are real non-negative values.

Principal Components Analysis (PCA)

PCA obtains the ***p*** dimensions/directions **maximizing the variance** of the projected data. Once the new basis of the p-dimensional space is obtained (Empirical Orthogonal Functions, **EOFs**) the reduction of the dimension is obtained chosing the first ***r*** principal components (**PCs**) or, equivalently, the ***r*** directions/dimensions explaining a predefined percentage of the total variance.

Obtaining the PCs

Since we are considering centered variables, the covariance matrix, can be expressed as:

$$\mathcal{C} = \frac{1}{n} X^t X \qquad \mathcal{C} \in \mathbb{R}^{d \times d}$$

Let's consider its eigendecomposition $\mathcal{C}P = P\Lambda$

$$P \in \mathbb{R}^{d \times d}$$
$$\Lambda = \mathrm{diag}(\lambda_j)$$

arranging the eigenvectors in the columns of P by decreasing eigenvalue $\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_d$

Principal Components Analysis (PCA)

PCA obtains the ***p*** dimensions/directions **maximizing the variance** of the projected data. Once the new basis of the p-dimensional space is obtained (Empirical Orthogonal Functions, **EOFs**) the reduction of the dimension is obtained chosing the first ***r*** principal components (**PCs**) or, equivalently, the ***r*** directions/dimensions explaining a predefined percentage of the total variance.

Obtaining the PCs

Since we are considering centered variables, the covariance matrix, can be expressed as:

$$\mathcal{C} = \frac{1}{n} X^t X \qquad \mathcal{C} \in \mathbb{R}^{d \times d}$$

Total variance can be written as $$TV = \sum_{j=1}^{d} s_j^2 = \mathrm{tr}(\mathcal{C}) = \sum_{j=1}^{d} \lambda_j$$

$$\mathcal{C}_Z = \frac{1}{n} Z^t Z = \frac{1}{n} (XP)^t XP = \frac{1}{n} P^t X^t XP = P^t \mathcal{C} P = P^t P \Lambda = \Lambda$$

Principal Components Analysis (PCA)

PCA obtains the ***p*** dimensions/directions **maximizing the variance** of the projected data. Once the new basis of the p-dimensional space is obtained (Empirical Orthogonal Functions, **EOFs**) the reduction of the dimension is obtained chosing the first ***r*** principal components (**PCs**) or, equivalently, the ***r*** directions/dimensions explaining a predefined percentage of the total variance.

Obtaining the PCs

Since we are considering centered variables, the covariance matrix, can be expressed as:

$$\mathcal{C} = \frac{1}{n} X^t X \qquad \mathcal{C} \in \mathbb{R}^{d \times d}$$

Total variance can be written as $$TV = \sum_{j=1}^{d} s_j^2 = \mathrm{tr}(\mathcal{C}) = \sum_{j=1}^{d} \lambda_j$$

$$\mathrm{EV}_k = \frac{\lambda_k}{TV}$$ is the variance fraction explained by the k-th principal component

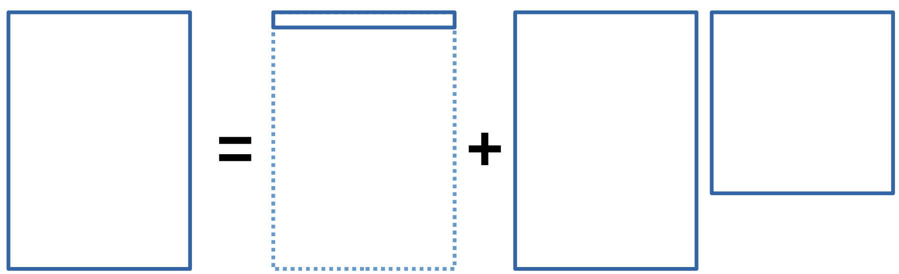
Principal Components Analysis (PCA)

Recovery of the original variables

$$Z = XP \rightarrow X = ZP^t$$

But remember that we removed the mean, so...

$$X = \overline{X} + ZP^t$$



$n \times d$ $\quad$ $n \times d$ $\quad$ $n \times d$ $\quad$ $d \times d$

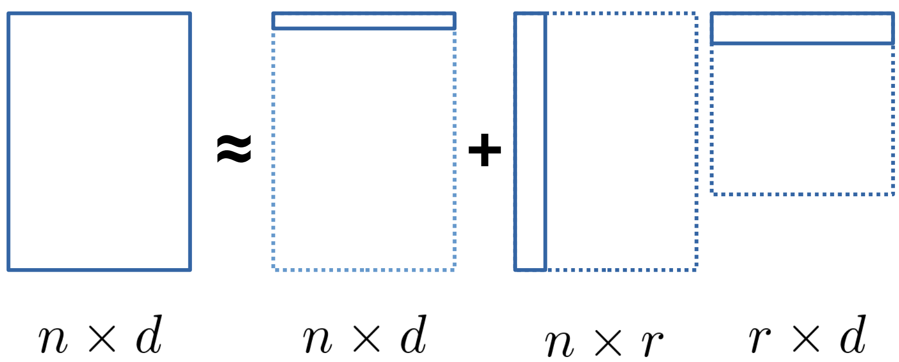
Principal Components Analysis (PCA)

Recovery of the original variables

$$Z = XP \rightarrow X = ZP^t$$

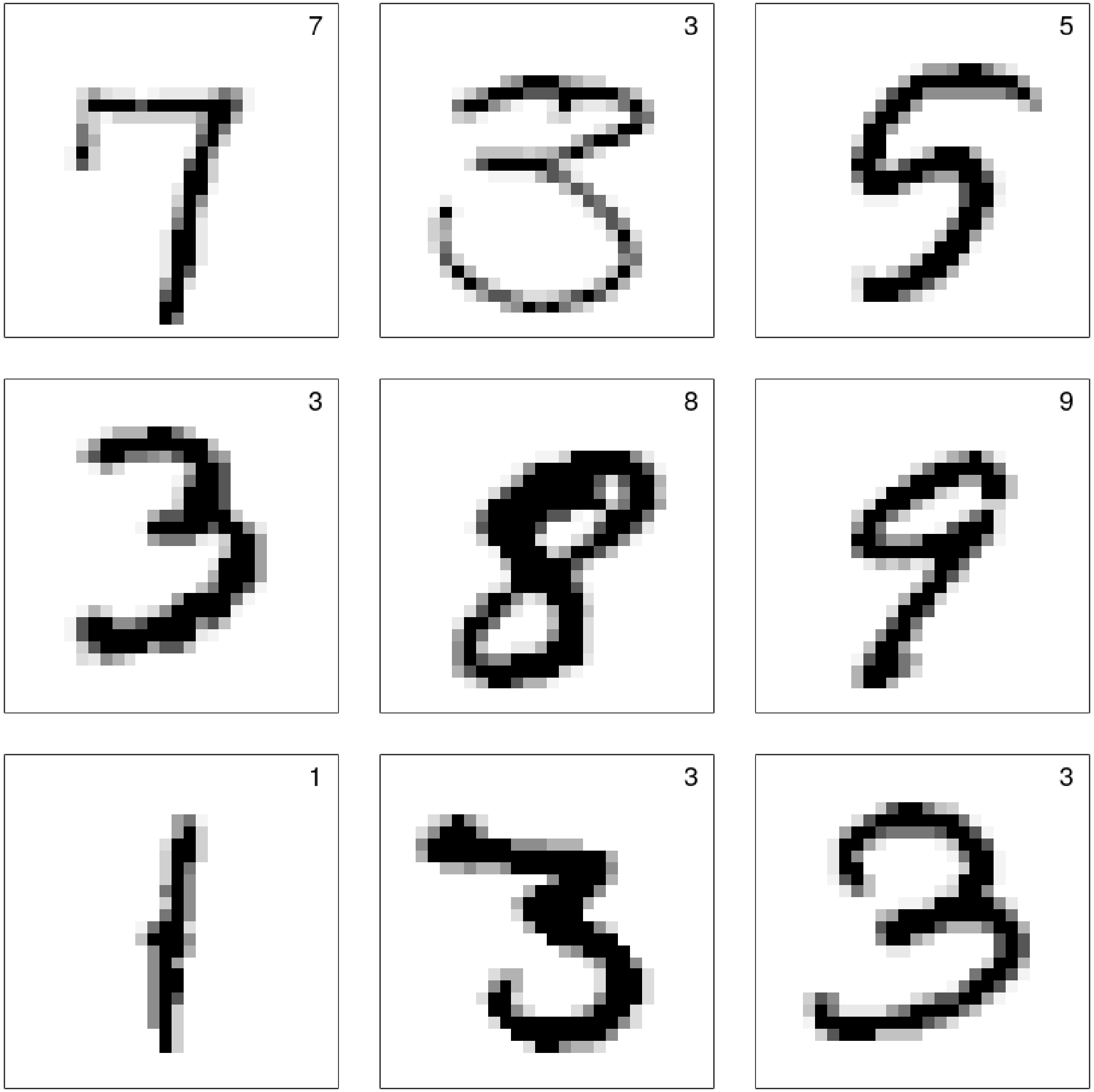
And maximum variance is retained by the first few r components:

$$X \approx \overline{X} + Z_r P_r^t$$



Example – MNIST Dataset

```
datafile <- "train.csv"
if (! file.exists(datafile)) {
  download.file(
    url = paste0(
      "http://www.meteo.unican.es/work/",
      datafile
    ),
    destfile = datafile
  )
}
train <- read.csv(datafile)
y <- train[,1]; x <- train[,-1]
nside <- sqrt(dim(x)[2])
# Plot
range.start <- 7
show.range <- range.start:(range.start+8)
opar <- par
par(mfrow = c(3,3), mar=c(1,1,1,1))
for (i in show.range) {
  numimage <- matrix(
    as.matrix(x[i,]),
    nrow = nside, ncol = nside
  )[,nside:1]
  image(numimage,
    col = gray.colors(12,1,0),
    xaxt="n", yaxt="n"
  )
  text(0.95,0.95, y[i], cex=1.5)
}
```



Example – MNIST Dataset

PCA

```
nsamples <- 10000  # max 42000
pca <- prcomp(
  x[1:nsamples,],
  center = TRUE,
  scale = FALSE
)
lambdas <- pca$sdev^2
pcs <- pca$x          # Scores
eofs <- pca$rotation # Loadings
media <- pca$center
```

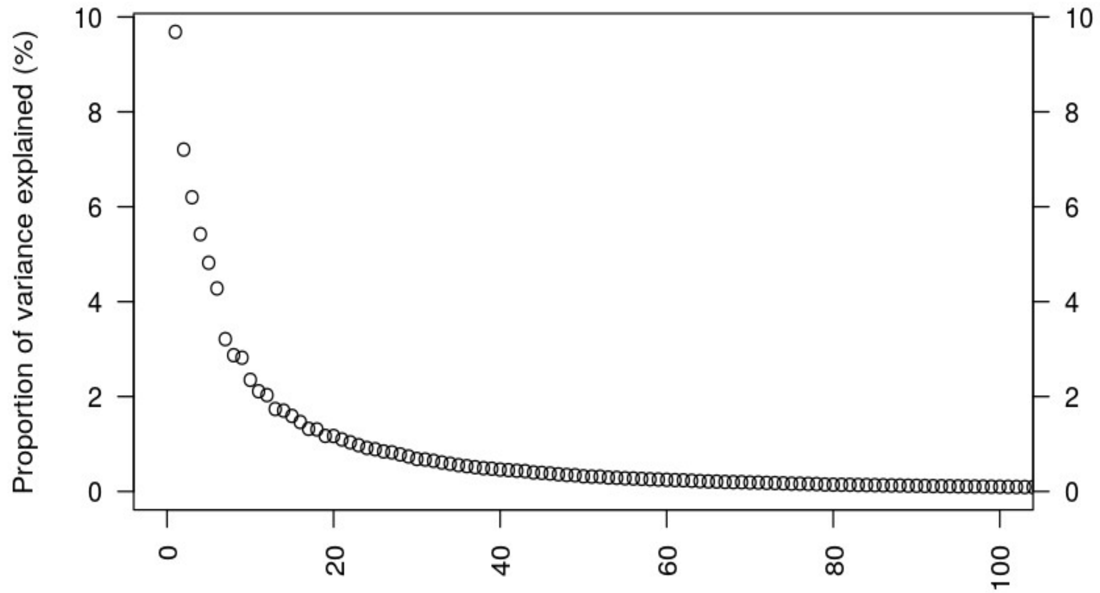
Example – MNIST Dataset

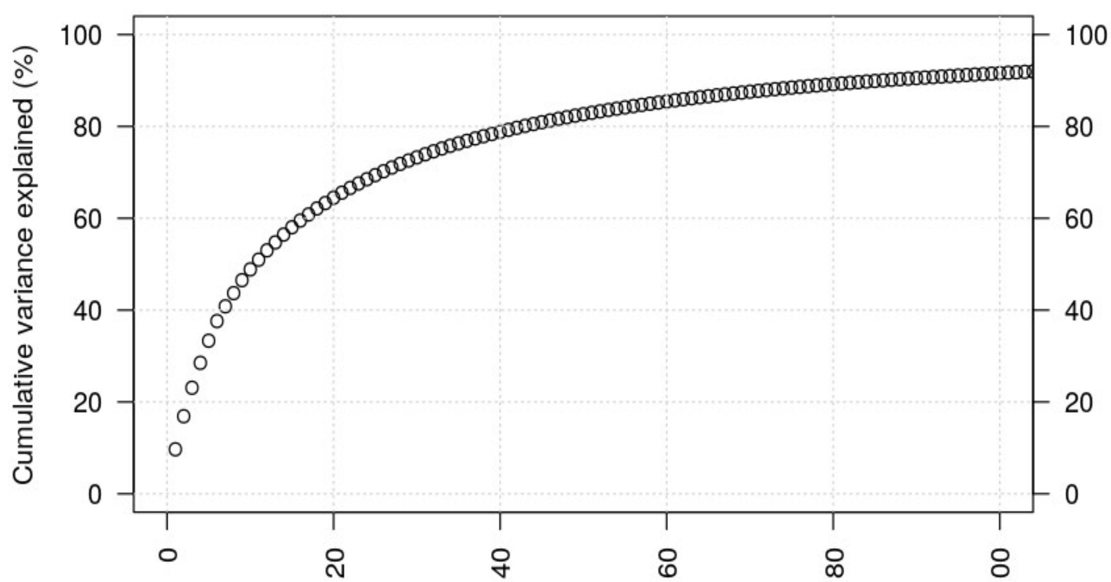
PCA

```
nsamples <- 10000  # max 42000
pca <- prcomp(
  x[1:nsamples,],
  center = TRUE,
  scale = FALSE
)
lambdas <- pca$sdev^2
pcs <- pca$x           # Scores
eofs <- pca$rotation # Loadings
media <- pca$center
```

Scree and cummulative EV plot

```
show.just <- 100 # max 748
par(mfrow=c(2,1), mar=c(2.1,4,1,2.5))
plot(lambdas/sum(lambdas)*100,
  xlim=c(0,show.just)
)
axis(side=4)
plot(cumsum(lambdas)/sum(lambdas)*100,
  ylim=c(0,100), xlim=c(0,show.just)
)
axis(side=4)
grid()
```

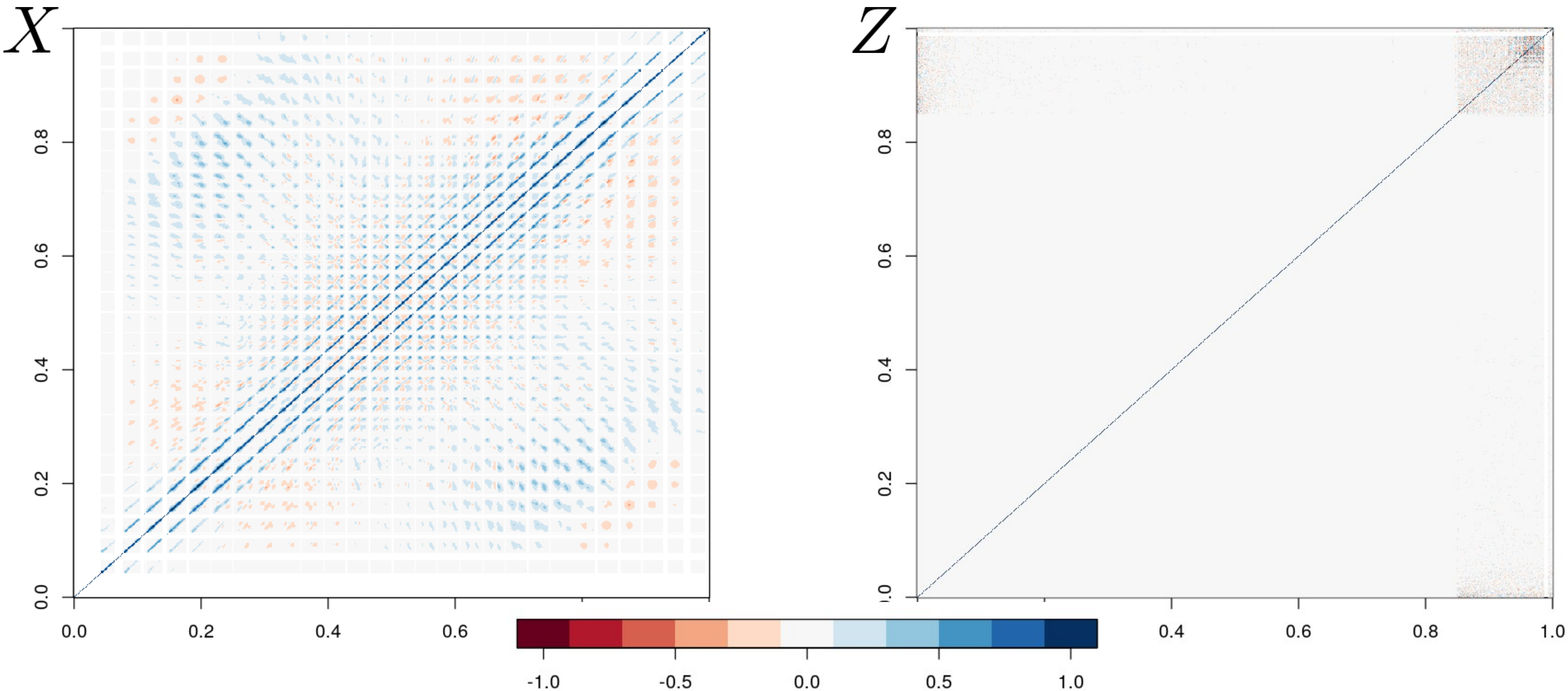




Example – MNIST Dataset

Uncorrelated new variables

```
library(RColorBrewer)
colores <- brewer.pal(11,"RdBu")
image(cor(x[1:nsamples,]), zlim=c(-1,1), col = colores)
image(cor(pcs), zlim=c(-1,1), col = colores)
```

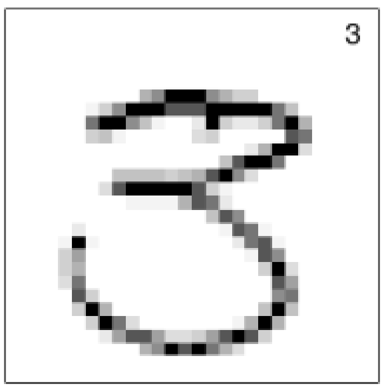


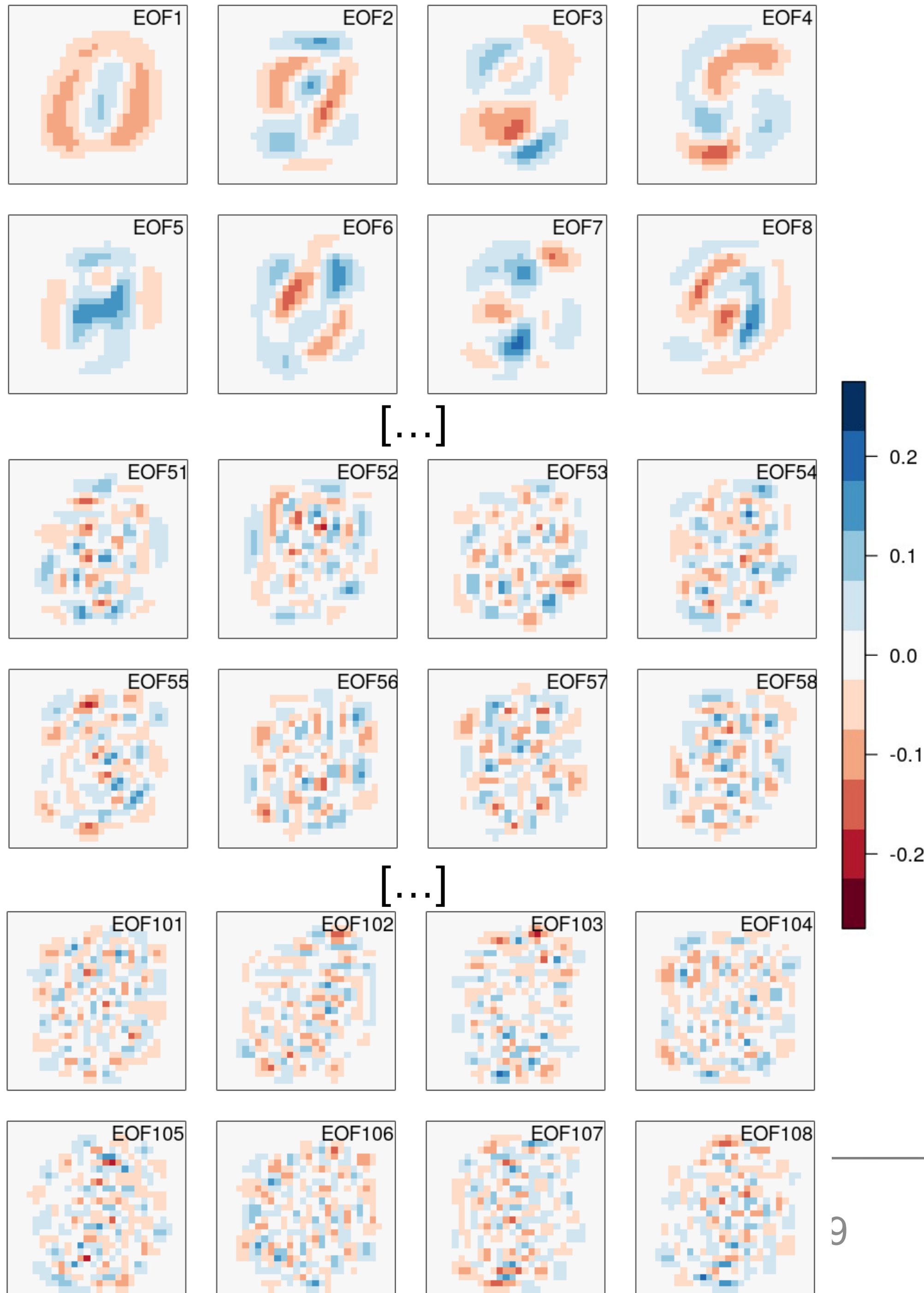
Example – MNIST Dataset

Orthonormal loadings (EOFs)

```
par(mfrow = c(4,4), mar=c(1,1,1,1))
for (i in c(101:108)) {
  numimage <- matrix(
    eofs[,i],
    nrow = nside, ncol = nside
  )[,nside:1]
  image(
    numimage, col=colores,
    xaxt="n", yaxt="n", zlim=c(-1,1)*0.25
  )
  text(0.8,0.95, sprintf("EOF%d",i), cex=1.5)
}
```

```
PC1   123.05
PC2    64.35
PC3   525.91
PC4   -69.94
PC5  -226.51
PC6   314.63
PC7  -344.90
PC8    35.42
[...]
```



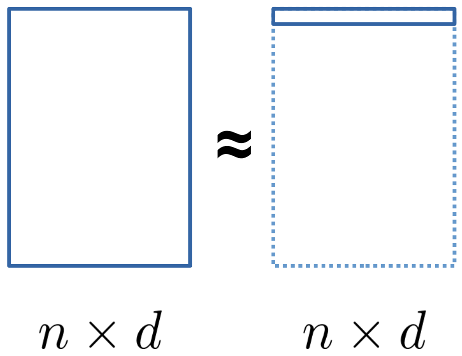


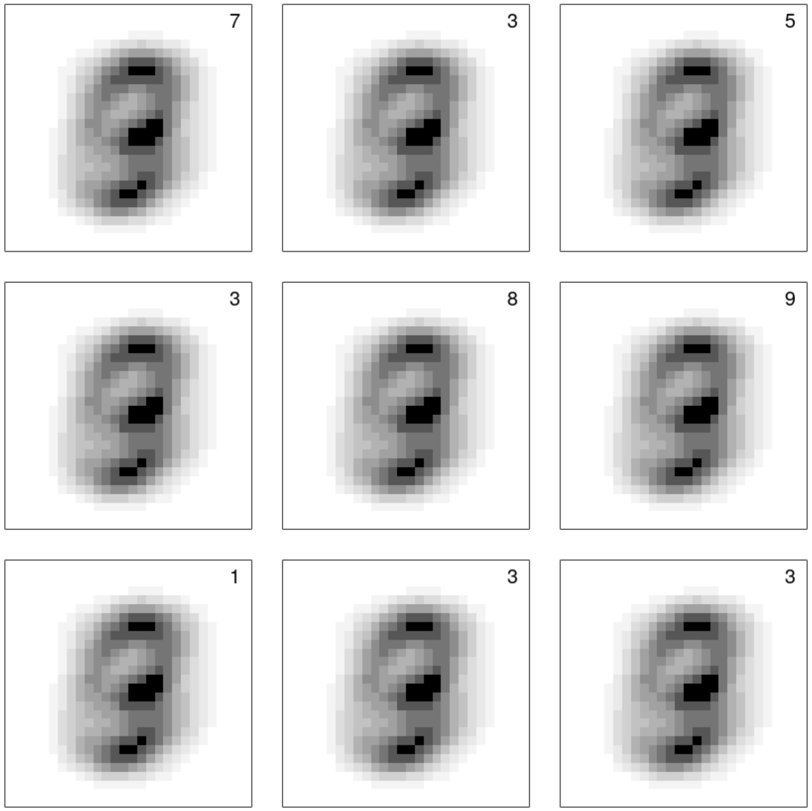
Example – MNIST Dataset

Recovering the field

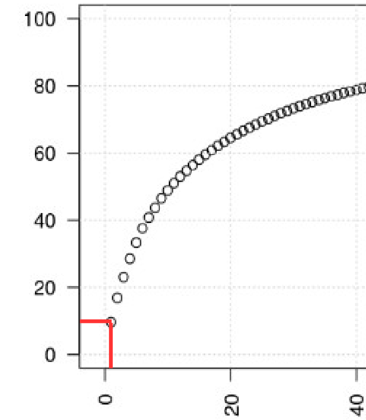
```
media.stacked <- matrix(rep(media,nsamples), nrow=nsamples, byrow=TRUE)
reconstructed <- media.stacked
par(mfrow = c(3,3), mar=c(1,1,1,1))
for (i in show.range) {
  numimage <- matrix(as.matrix(reconstructed[i,]), nrow = nside, ncol = nside)[,nside:1]
  image(numimage, col = gray.colors(12,1,0))
  text(0.95,0.95, y[i], cex=1.5)
}
```

$$X \approx \overline{X}$$





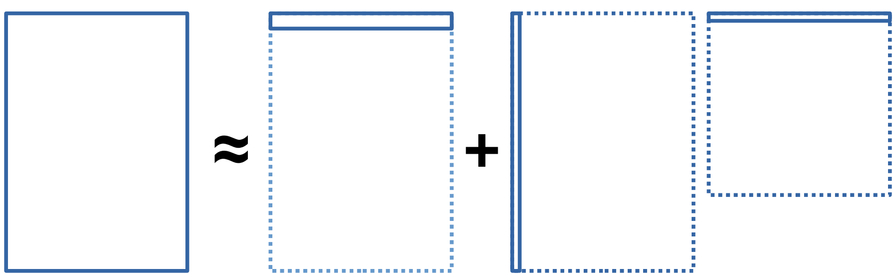
Example – MNIST Dataset



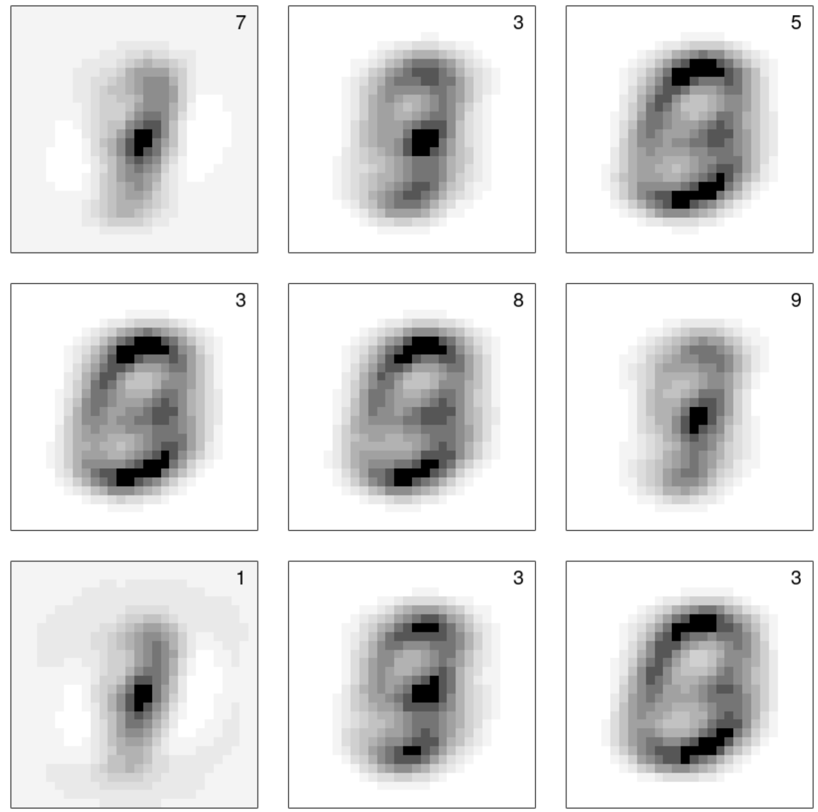
Recovering the field

```
r <- 1 # Number of retained PCs
media.stacked <- matrix(rep(media,nsamples), nrow=nsamples, byrow=TRUE)
reconstructed <- media.stacked + pcs[,1:r] %*% t(eofs[,1:r])
par(mfrow = c(3,3), mar=c(1,1,1,1))
for (i in show.range) {
  numimage <- matrix(as.matrix(reconstructed[i,]), nrow = nside, ncol = nside)[,nside:1]
  image(numimage, col = gray.colors(12,1,0))
  text(0.95,0.95, y[i], cex=1.5)
}
```

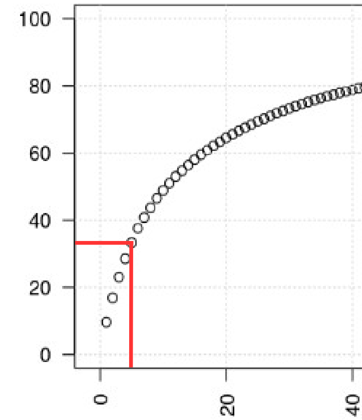
$$X \approx \overline{X} + Z_1 P_1^t$$



$n \times d$ $\quad$ $n \times d$ $\quad$ $n \times 1$ $\quad$ $1 \times d$



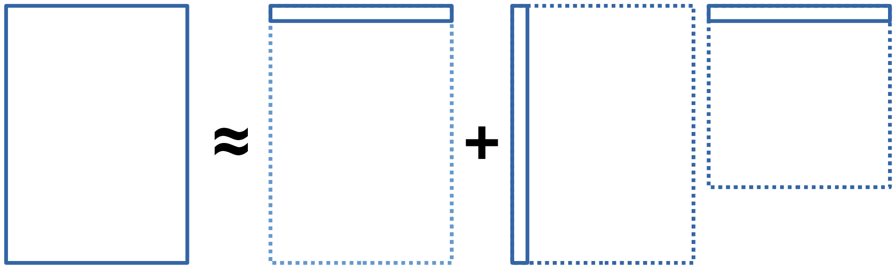
Example – MNIST Dataset



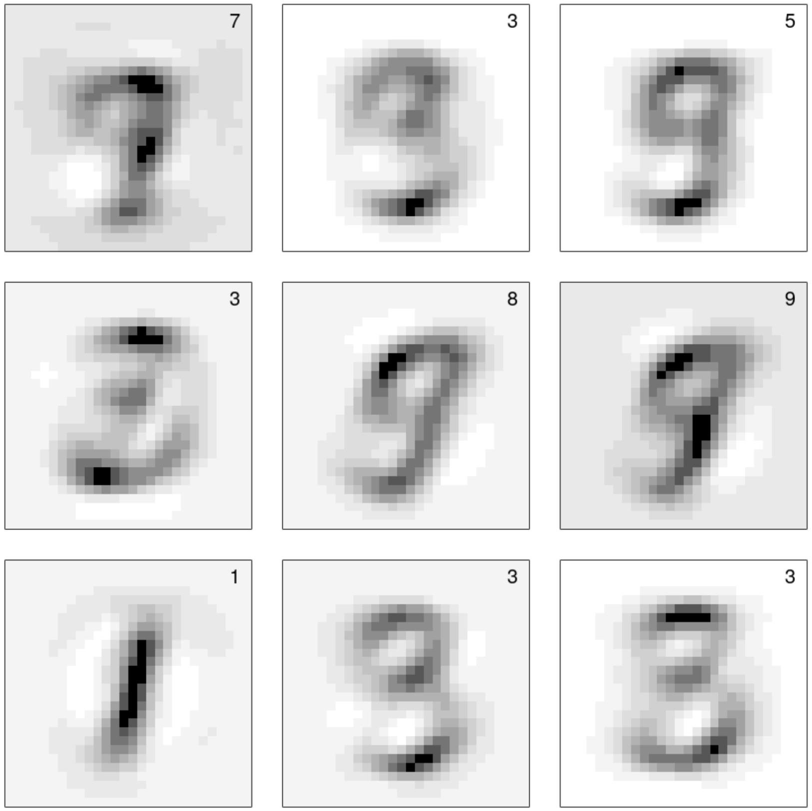
Recovering the field

```
r <- 5 # Number of retained PCs
media.stacked <- matrix(rep(media,nsamples), nrow=nsamples, byrow=TRUE)
reconstructed <- media.stacked + pcs[,1:r] %*% t(eofs[,1:r])
par(mfrow = c(3,3), mar=c(1,1,1,1))
for (i in show.range) {
  numimage <- matrix(as.matrix(reconstructed[i,]), nrow = nside, ncol = nside)[,nside:1]
  image(numimage, col = gray.colors(12,1,0))
  text(0.95,0.95, y[i], cex=1.5)
}
```

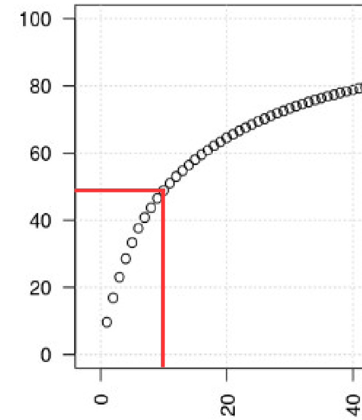
$$X \approx \overline{X} + Z_5 P_5^t$$



$n \times d$ $\quad$ $n \times d$ $\quad$ $n \times 5$ $\quad$ $5 \times d$

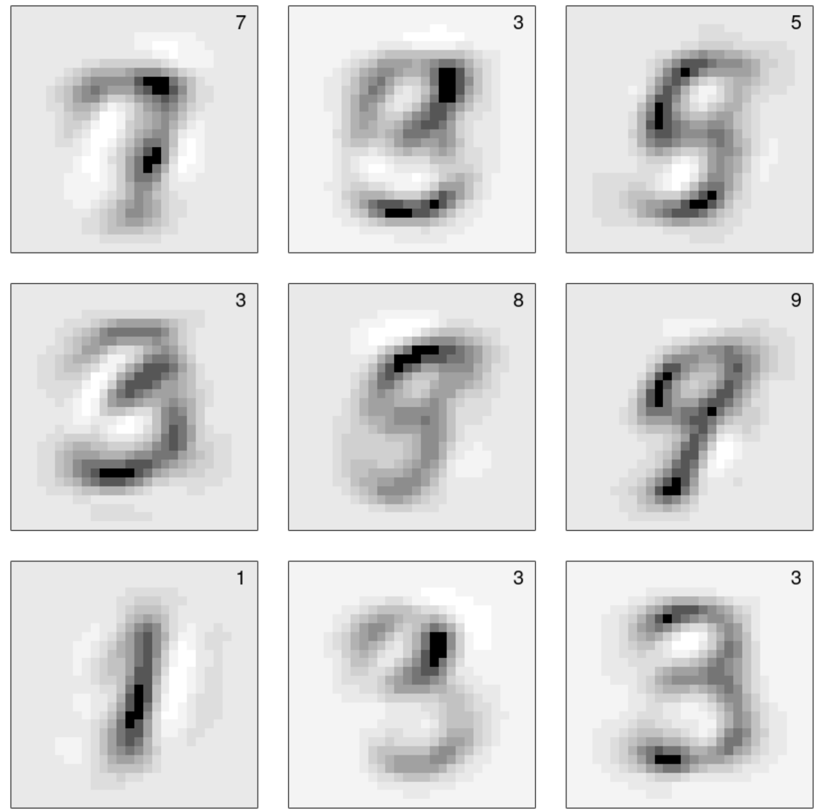


Example – MNIST Dataset

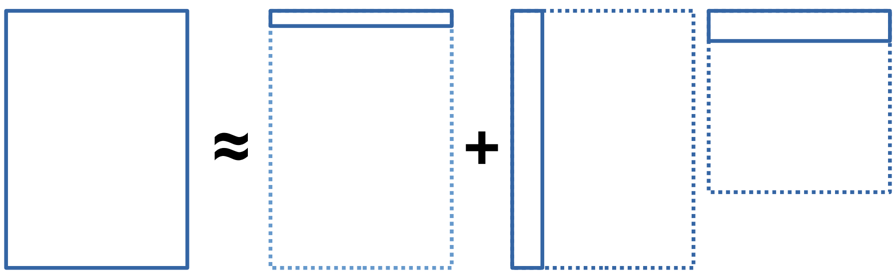


Recovering the field

```
r <- 10 # Number of retained PCs
media.stacked <- matrix(rep(media,nsamples), nrow=nsamples, byrow=TRUE)
reconstructed <- media.stacked + pcs[,1:r] %*% t(eofs[,1:r])
par(mfrow = c(3,3), mar=c(1,1,1,1))
for (i in show.range) {
  numimage <- matrix(as.matrix(reconstructed[i,]), nrow = nside, ncol = nside)[,nside:1]
  image(numimage, col = gray.colors(12,1,0))
  text(0.95,0.95, y[i], cex=1.5)
}
```

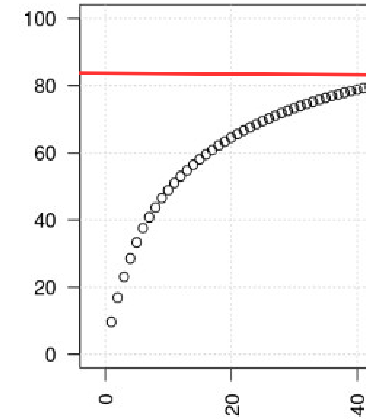


$$X \approx \overline{X} + Z_{10} P_{10}^{t}$$



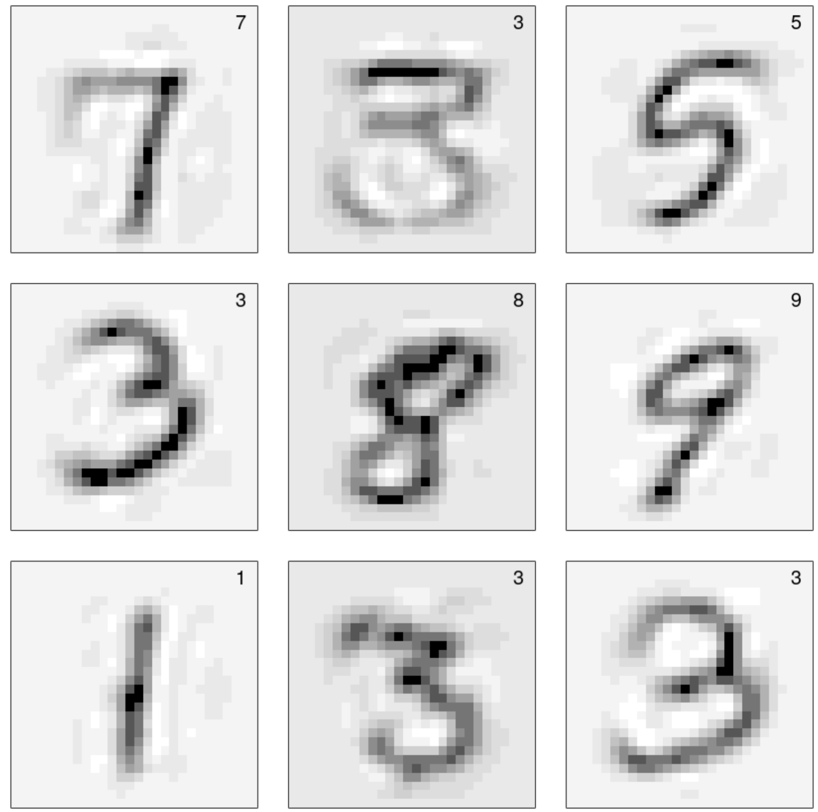
$n \times d$ $\quad n \times d$ $\quad n \times 10$ $\quad 10 \times d$

Example – MNIST Dataset

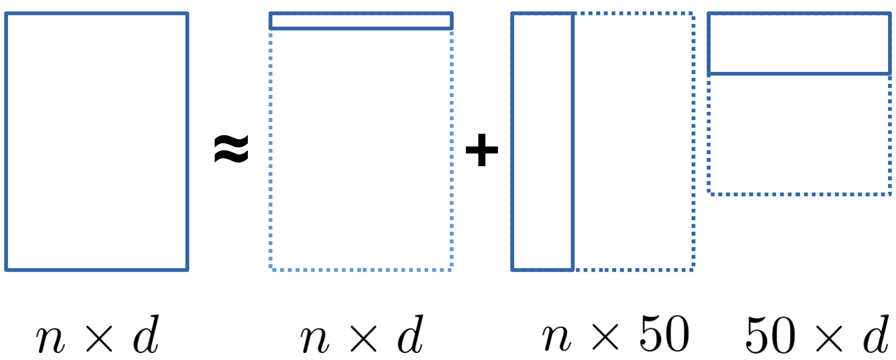


Recovering the field

```
r <- 50 # Number of retained PCs
media.stacked <- matrix(rep(media,nsamples), nrow=nsamples, byrow=TRUE)
reconstructed <- media.stacked + pcs[,1:r] %*% t(eofs[,1:r])
par(mfrow = c(3,3), mar=c(1,1,1,1))
for (i in show.range) {
  numimage <- matrix(as.matrix(reconstructed[i,]), nrow = nside, ncol = nside)[,nside:1]
  image(numimage, col = gray.colors(12,1,0))
  text(0.95,0.95, y[i], cex=1.5)
}
```



$$X \approx \overline{X} + Z_{50} P_{50}^{t}$$



Principal Components Analysis (PCA)

How many PCs consider?

Subjective criteria
Based on the EVP

Objective criteria
Based on parameters (e.g. Minimum Description Length)

$$r = \underset{k}{\mathrm{argmin}}[\mathrm{MDL}(k)]$$

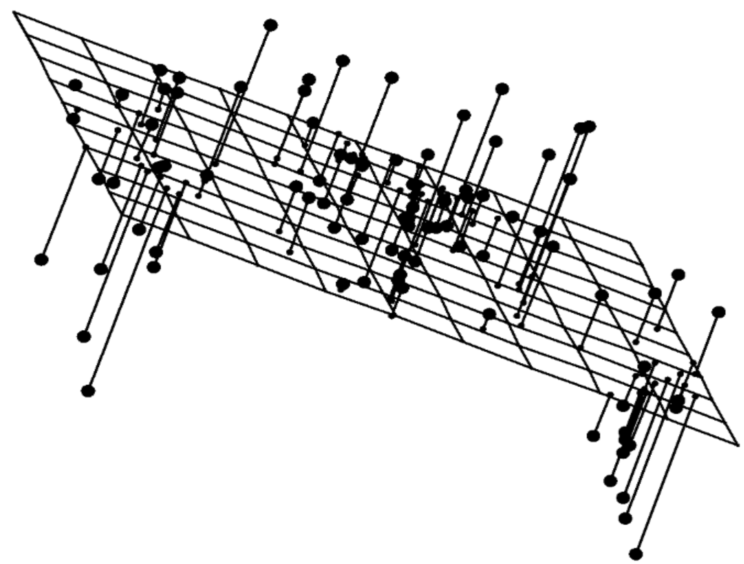
$$\mathrm{MDL}(k) = n\left[\log \prod_{j=1}^{k} \sigma_j^2 + (d-k)\log\left(\frac{1}{d-k}\sum_{j=k+1}^{d}\sigma_j^2\right)\right] + \frac{k(2d-k)}{2}\log n$$

Principal Components Analysis (PCA)

- Loadings and scores through SVD
  - Numerically more stable (used in R function **`prcomp`**)
  - No need to compute covariance matrix

$$X = L\Sigma R^t \;\Rightarrow\; P = R, Z = L\Sigma, \Lambda = \frac{1}{n}\Sigma^2$$

- PCA depends on variability of the variables
  - It is common to standardize (not only center) variables prior to PCA
- Leading EOFs span the linear subspace closest to data points



Principal component analysis

Pros

- Minimum MSE projection.
- Reduction of the dimension.
- Compression.
- Uncorrelated dimensions
  → Pre-process to linear models
- Criteria to choose dimensions.

Cons

- Loss of information in the case of supervised learning
- Linear approach, non useful to nonlinear problems

Principal component analysis

Pros

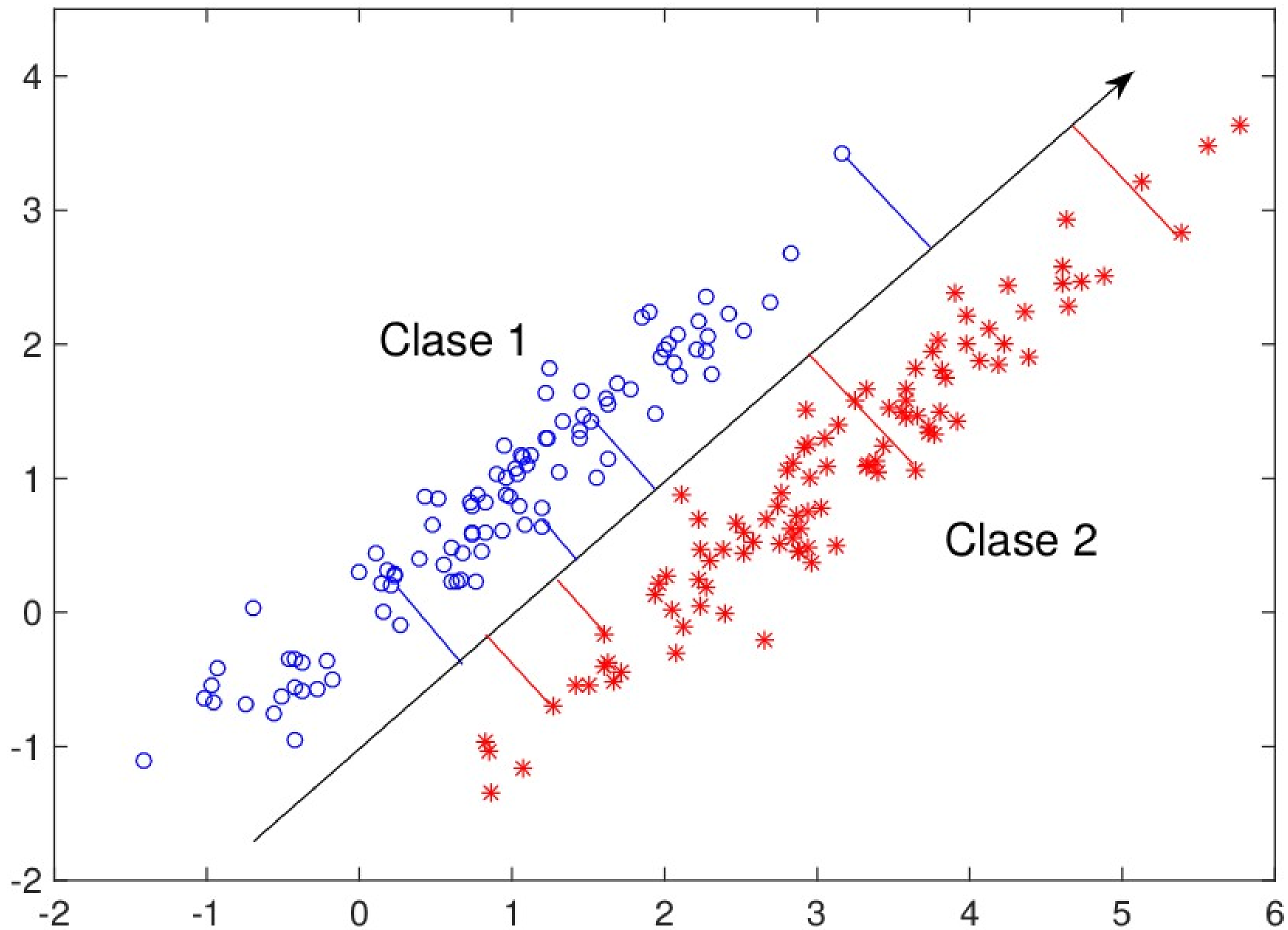
- Minimum MSE projection.
- Reduction of the dimension.
- Compression.
- Uncorrelated dimensions
  → Pre-process to linear models
- Criteria to choose dimensions.

Cons

- Loss of information in the case of supervised learning → **LDA**
- Linear approach, non useful to nonlinear problems → **KPCA**

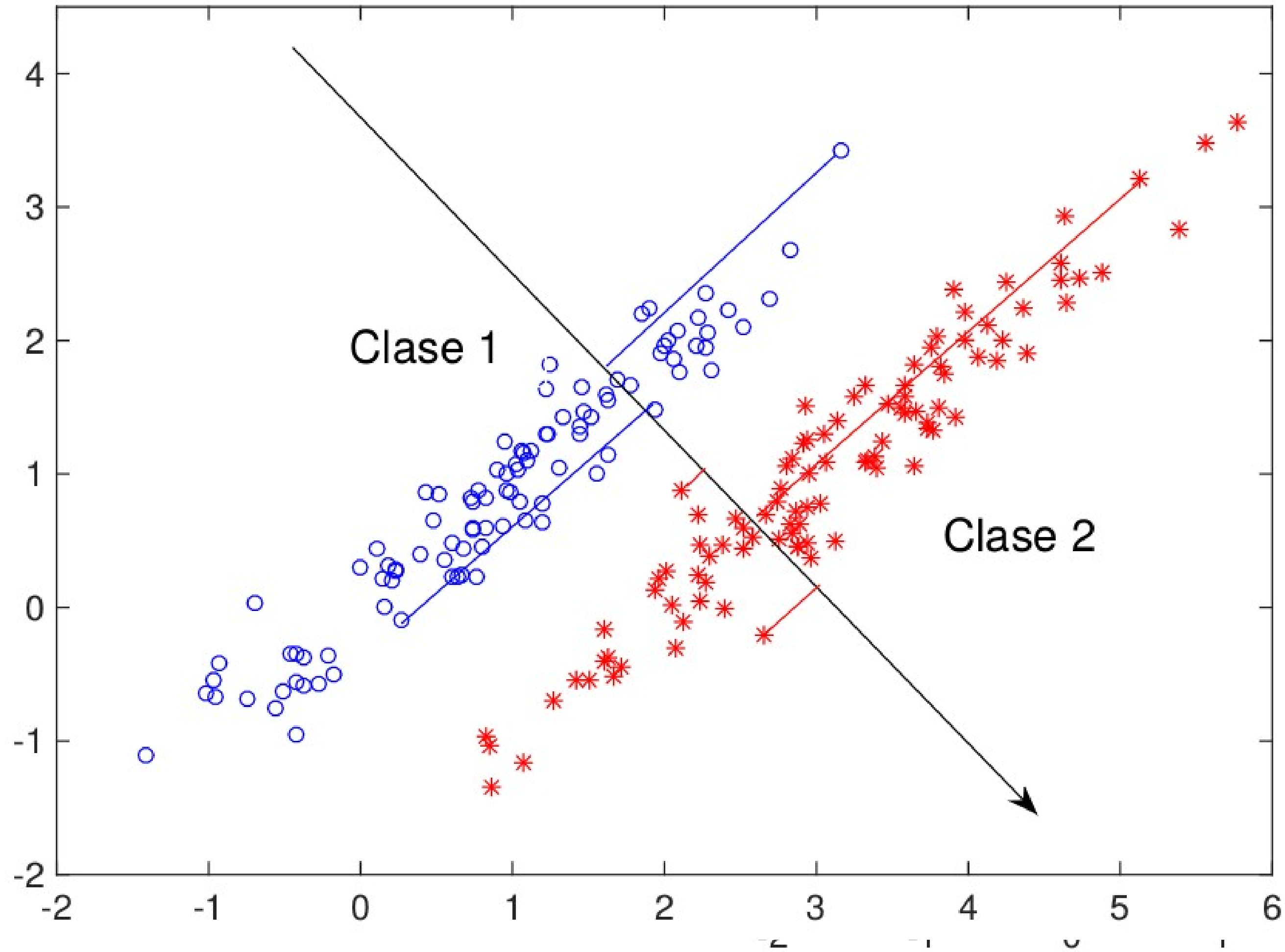
Linear Discriminant Analysis (LDA)

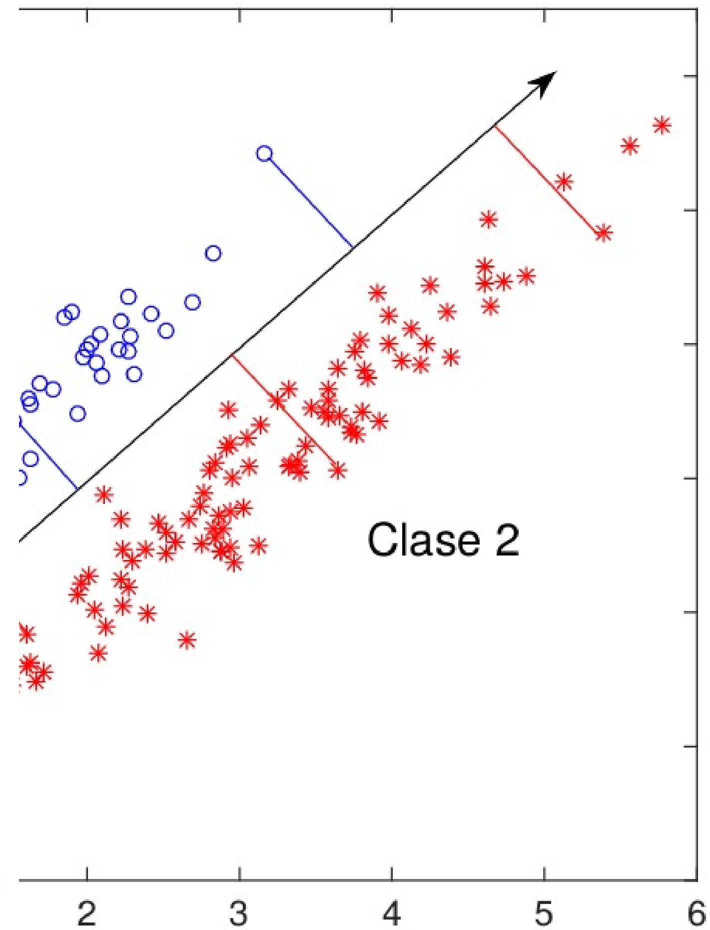
Maximum variance directions (**PCA**) are not always the optimal choice to linearly separate several classes in the case of supervised learning.



Linear Discriminant Analysis (LDA)

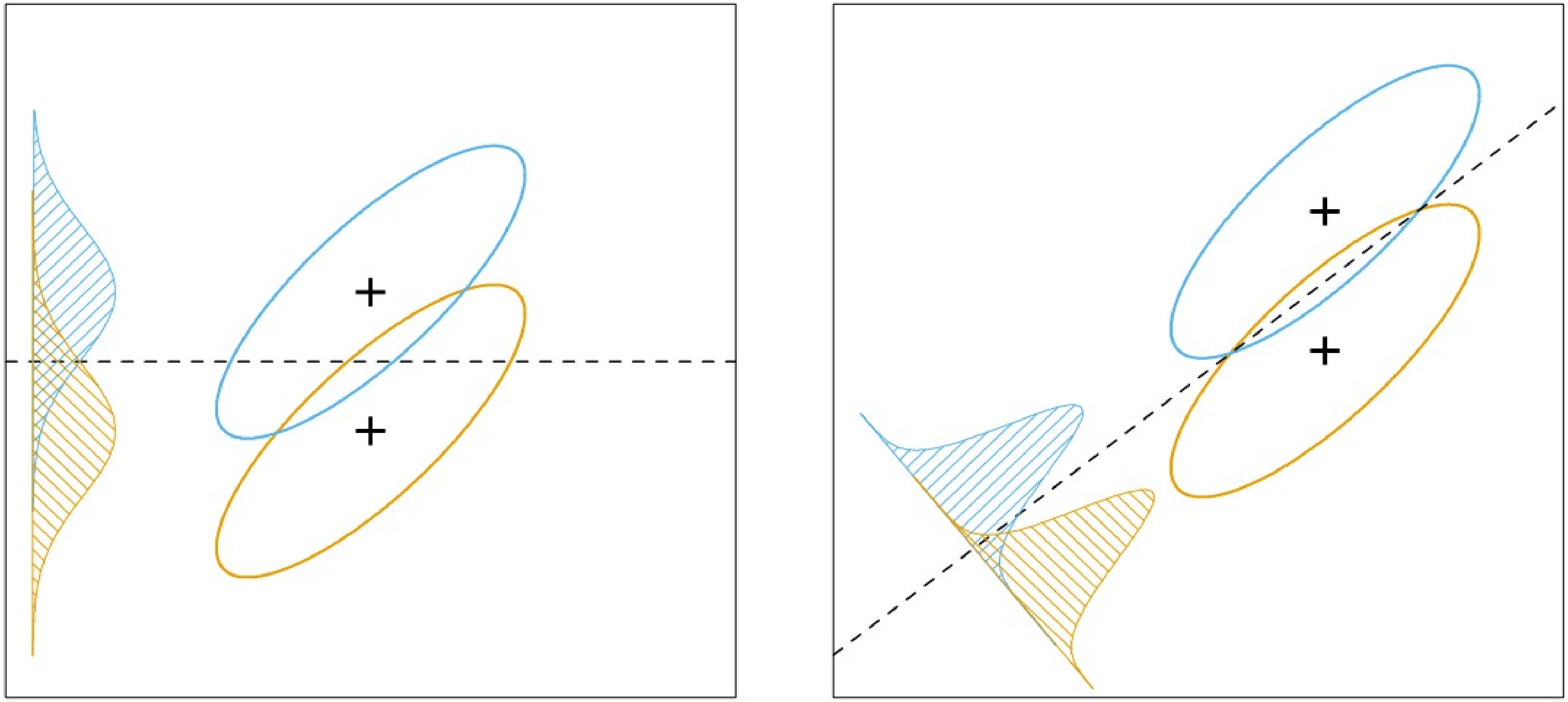
Maximum variance directions (**PCA**) are not always the optimal choice to linearly separate several classes in the case of supervised learning. This could be a better option:





Linear Discriminant Analysis (LDA)

Maximum variance directions (**PCA**) are not always the optimal choice to linearly separate several classes in the case of supervised learning.



Source: Hastie (2017)

Not as easy as using the direction of the line connecting the centroids

Linear Discriminant Analysis (LDA)

The Bayesian classifier is optimal in terms of MSE:

$$\Pr(G=k|X=x)=\frac{f_k(x)\pi_k}{\sum_{\ell=1}^{K} f_\ell(x)\pi_\ell}$$

If we assume multivariate normal densities for each class:

$$f_k(x)=\frac{1}{(2\pi)^{p/2}|\mathbf{\Sigma}_k|^{1/2}}e^{-\frac{1}{2}(x-\mu_k)^T\mathbf{\Sigma}_k^{-1}(x-\mu_k)}$$

with common variance Σ

$$\begin{aligned}\log\frac{\Pr(G=k|X=x)}{\Pr(G=\ell|X=x)} &= \log\frac{f_k(x)}{f_\ell(x)}+\log\frac{\pi_k}{\pi_\ell}\\ &= \log\frac{\pi_k}{\pi_\ell}-\frac{1}{2}(\mu_k+\mu_\ell)^T\mathbf{\Sigma}^{-1}(\mu_k-\mu_\ell)\\ &\quad + x^T\mathbf{\Sigma}^{-1}(\mu_k-\mu_\ell),\end{aligned}$$

Linear Discriminant Analysis (LDA)

The Bayesian classifier is optimal in terms of MSE:

$$\Pr(G = k|X = x) = \frac{f_k(x)\pi_k}{\sum_{\ell=1}^{K} f_\ell(x)\pi_\ell}$$

LDA implements the Bayesian classifier assuming **common variance** for all classes. It can discriminate among classes using a **linear discriminant function**:

$$\delta_k(x) = x^T \mathbf{\Sigma}^{-1} \mu_k - \frac{1}{2}\mu_k^T \mathbf{\Sigma}^{-1} \mu_k + \log \pi_k$$

and assigning the class $\mathrm{argmax}_k \delta_k(x)$

Linear Discriminant Analysis (LDA)

This is equivalent to a linear projection of the original data, seeking the direction that maximizes between-class variance relative to within-class variance (Fisher).

LDA obtains the projection matrix (***W***) by solving the following generalized eigenvalue problem: $\mathcal{C}_B W = \mathcal{C}_W W \Lambda$

Which is equivalent to the ordinary eigenvalue problem

$$\mathcal{C}_W^{-1} \mathcal{C}_B W = W \Lambda$$

if the within-class varince matrix is of full rank.

Between-class variance

$$\mathcal{C}_B = \frac{1}{n} \sum_{k=1}^{K} n_k (\mu_k - \mu)(\mu_k - \mu)^t$$

Within-class variance

$$\mathcal{C}_W = \frac{1}{n} \sum_{k=1}^{K} \sum_{i=1}^{n_k} (x_{ki} - \mu_k)(x_{ki} - \mu_k)^t$$

Linear Discriminant Analysis (LDA)

This is equivalent to a linear projection of the original data, seeking the direction that maximizes between-class variance relative to within-class variance (Fisher).

LDA obtains the projection matrix (***W***) by solving the following generalized eigenvalue problem: $\mathcal{C}_B W = \mathcal{C}_W W \Lambda$

Which is equivalent to the ordinary eigenvalue problem

$$\mathcal{C}_W^{-1} \mathcal{C}_B W = W \Lambda$$

if the within-class varince matrix is of full rank.

K-1 eigenvectors corresponding to the largest eigenvalues of $\mathcal{C}_W^{-1} \mathcal{C}_B$

K classes → K-1 dimensions

Linear Discriminant Analysis (LDA) – MNIST Dataset

Classifying a single number (2 classes)

```
library(MASS)   # lda
library(caret) # confusionMatrix
# Filter out fixed pixels
non.constant.pixels <- apply(x, MARGIN=2, FUN=sd) != 0
# Just find out whether we have a 9
y9 <- as.factor(ifelse(y==9, "is.9", "not.9"))
data9 <- data.frame(y9,x[,non.constant.pixels])
# Fit LDA model
lda.fitted <- lda(y9 ~ ., data9)
lda.pred <- predict(lda.fitted)
confusionMatrix(lda.pred$class, data9$y9)
```

```
Confusion Matrix and Statistics

          Reference
Prediction  is.9 not.9
     is.9   3213   975
     not.9   975 36837

               Accuracy : 0.9536
                 95% CI : (0.9515, 0.9556)
    No Information Rate : 0.9003
    P-Value [Acc > NIR] : <2e-16

                  Kappa : 0.7414

 Mcnemar's Test P-Value : 1

            Sensitivity : 0.76719
            Specificity : 0.97421
         Pos Pred Value : 0.76719
         Neg Pred Value : 0.97421
             Prevalence : 0.09971
         Detection Rate : 0.07650
   Detection Prevalence : 0.09971
      Balanced Accuracy : 0.87070

       'Positive' Class : is.9
```

Linear Discriminant Analysis (LDA) – MNIST Dataset

Classifying all numbers (10 classes)

```
data.all <- data.frame(y=as.factor(y), x[,non.constant.pixels])
lda.fitted.all <- lda(y ~ ., data.all)
lda.pred.all <- predict(lda.fitted.all)
confusionMatrix(lda.pred.all$class, data.all$y)
```

```
Confusion Matrix and Statistics

          Reference
Prediction    0    1    2    3    4    5    6    7    8    9
         0 3916    0   44    8    4   41   47   20   22   27
         1    2 4498  116   60   45   43   40  103  202   18
         2   13   23 3427  122   20   16   29   36   21    5
         3   20   18  126 3695    3  183    3   25  145   67
         4   19    9   86   17 3671   42   66  117   50  235
         5   63   22   17  158   26 3138   99   12  186   21
         6   38   10  126   14   19   76 3801    1   25    0
         7    2    1   28   63    2   21    0 3702    8  172
         8   53   98  178  123   30  156   50   19 3308   40
         9    6    5   29   91  252   79    2  366   96 3603

Overall Statistics

               Accuracy : 0.8752
                 95% CI : (0.872, 0.8784)
    No Information Rate : 0.1115
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.8613

 Mcnemar's Test P-Value : < 2.2e-16
```