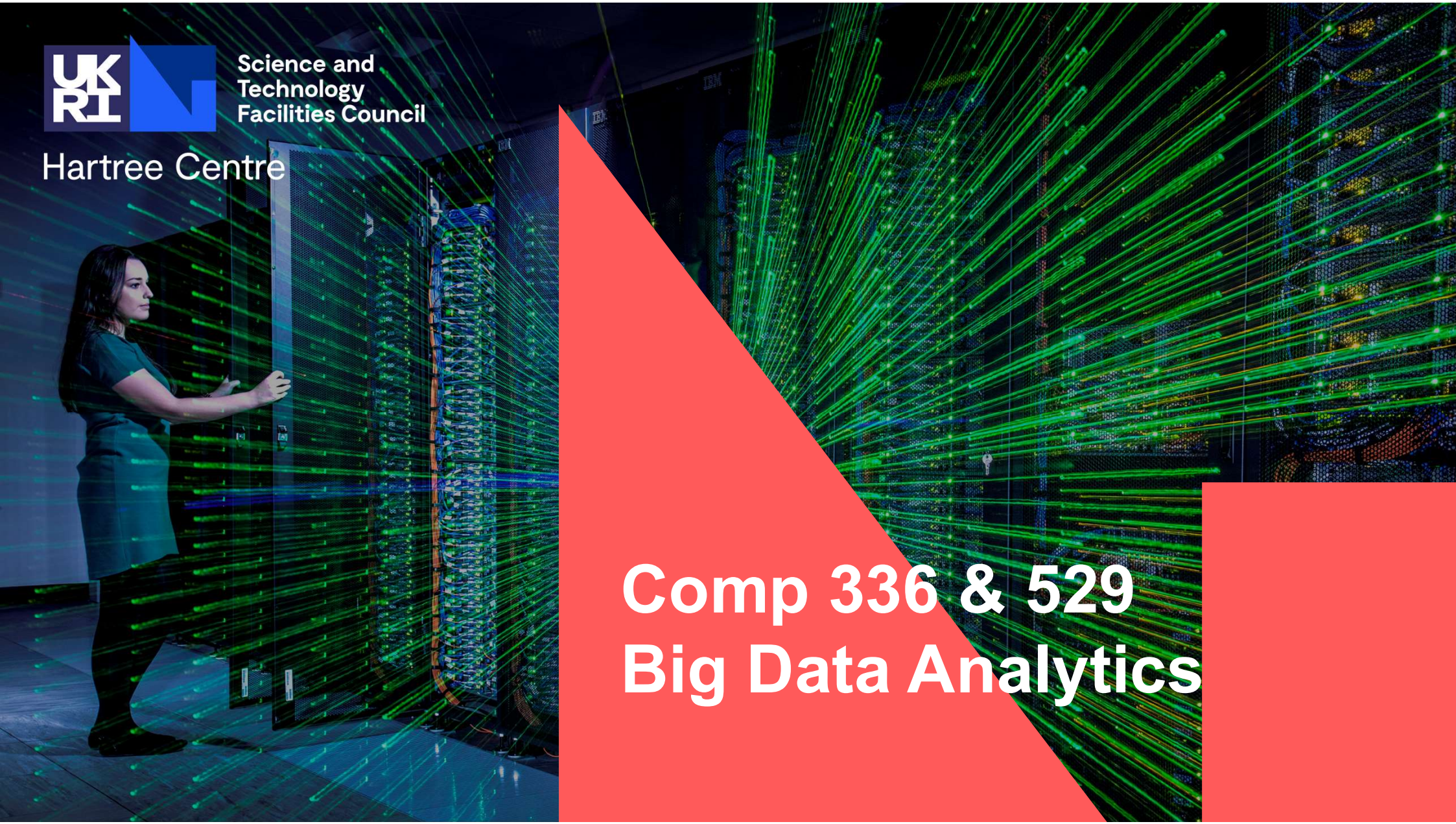


Comp 336 & 529
Big Data Analytics

Hartree Centre

Science and Technology Facilities Council

Hartree Centre

INTELLECTUAL PROPERTY RIGHTS NOTICE:

Many thanks to Prof. Y. Shi for agreeing to use his lecture material and material from his book on Optimization Based Data Mining: Theory and Applications, Springer, 2011.

Many thanks also to Dr. Raul Ramirez Velarde for agreeing to use some examples of PCA application from his lectures at Monterrey Tec.

Please note that the corresponding Intellectual Property Rights apply as per Springer and Monterrey Tec for any material from the book and the corresponding PCA examples .

Hartree Centre

Week 5: Linear Algebra Approaches and Algorithms for Data Analysis

Hartree Centre

Linear Algebra – some definitions

Eigenvalues and eigenvectors

Let **A** be an $n \times n$ matrix.

The eigenvalues of **A** are defined as the roots of:

$$\det(\mathbf{A} - \lambda\mathbf{I}) = |(\mathbf{A} - \lambda\mathbf{I})| = 0$$

Let λ be an eigenvalue of **A**.

Then there exists a vector x such that: $\mathbf{A}\,\mathbf{x} = \lambda\,\mathbf{x}$

The vector **x** is called an eigenvector of A corresponding the eigenvalue **λ.**

Eigenvalues and eigenvectors

Suppose we have a $n \times n$ matrix **A** with

eigenvectors $x_1, x_2, \ldots, x_n$, and eigenvalues $\lambda_1, \lambda_2, \ldots, \lambda_n$

so that $\mathbf{A}x_1 = \lambda_1 x_1$, $\mathbf{A}x_2 = \lambda_2 x_2$, …, $\mathbf{A}x_n = \lambda_n x_n$

In matrix format we have $\mathbf{AX}=\mathbf{X}\boldsymbol{\Lambda}$ where:

$$\mathbf{X} = (\boldsymbol{x_1} \quad \boldsymbol{x_2} \ldots \boldsymbol{x_n}) \qquad \boldsymbol{\Lambda} = \begin{pmatrix} \lambda_1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \lambda_n \end{pmatrix}$$

Gillies – PCA notes
G.H. Golub and C.F. van Loan, Matrix Computations, John Hopkins Univ. Press, 3rd Edition, 1996

Eigenvalues and eigenvectors

Now orthogonalizing and normalising (to unit ones) the eigenvectors we obtain:

$$XX^T = X^T X = \mathrm{I}$$

Therefore $X^T \boldsymbol{A}\, X = \Lambda$ and $\boldsymbol{A} = \mathrm{X}\ \Lambda\, X^T$

Then if we have a *nxn* covariance matrix ***C*** there exist such matrices X and X^T that $X^T C\, X = \Lambda$

How we apply this in data analysis?

Assume we have *m* variables and we obtain *n* measurements. We calculate their mean value μ.

We form matrix **B** as a mxn matrix with its i-th column being $\boldsymbol{x}_i$-μ, e.g. re-centred so that the mean is 0.

Form the covariance mxm matrix **C** such that **C**= $\frac{1}{n-1}\boldsymbol{B}\boldsymbol{B}^{\boldsymbol{T}}$

The i-th element C_{ii} of **C** is the variance of the i-th variable and the ij-th element C_{ij} of **C** is the covariance between the i-th and j-th variables (i $\neq$ j).

How we apply PCA in data analysis?

Note that C is a symmetric matrix, has real eigenvalues and there is an orthonormal basis of eigenvectors.

Therefore we can orthogonally diagonalise the matrix C with eigenvalues in decreasing order ($\lambda_1 \geq \lambda_2 \geq \cdots \lambda_m \geq 0$) with corresponding orthogonal eigenvectors $\boldsymbol{u_1, u_2, \ldots, u_m}$

G.H. Golub and C.F. van Loan, Matrix Computations, John Hopkins Univ. Press, 3rd Edition, 1996

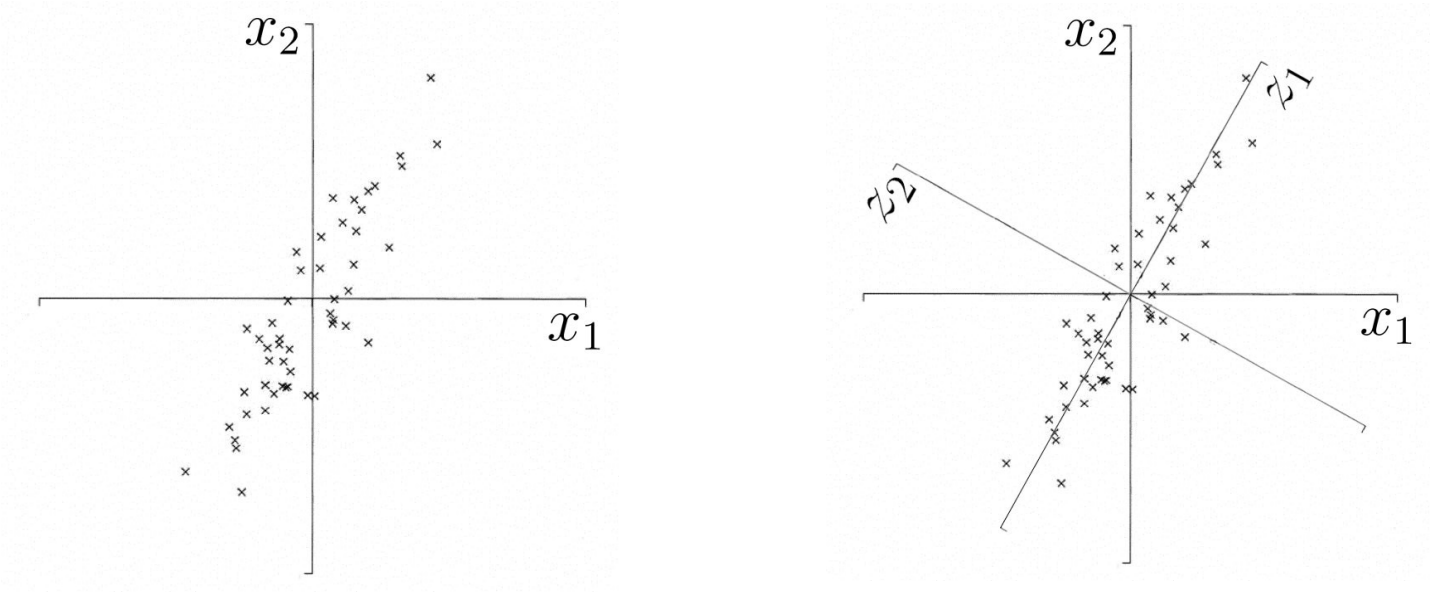
How it works in practice?

1. Collect the *n* samples of m-dimensional data. Compute the mean *μ,* build the matrix **B** and compute **C**.

2. Find the eigenvalues **C** in decreasing order and the corresponding orthogonal eigenvectors

3. Analyse the results. If there is small number of eigenvectors that are bigger than the others they cover most of the variance so dimension reduction is possible. Analyse which variables are most important in the first, second, etc. principal components.

PCA transformations illustration



Calculate the 1st PC, corresponding to the first eigenvalue (best fit)
Calculate the 2nd PC, corresponding to the second eigenvalues (best fit) orthogonal to the 1st one and so on… until m.

Questions?

Singular values and singular vectors

There is clear relation between eigenvalues and eigenvectors on one side and singular values and corresponding singular vectors on the other. For more details on this please look at:

Jon Shlens, A Tutrorial on Principal Component Analysis, 2003
https://www.cs.princeton.edu/picasso/mats/PCA-Tutorial-Intuition_jp.pdf

Science and Technology Facilities Council

Hartree Centre

PCA Examples

Science and Technology Facilities Council

Hartree Centre

Science and Technology Facilities Council

Hartree Centre

PCA of Tumor Data for Cancer Patients

Clustering and Dimensionality Reduction

UCI machine Learning Repository

Science and Technology Facilities Council

Hartree Centre

Data to be analysed

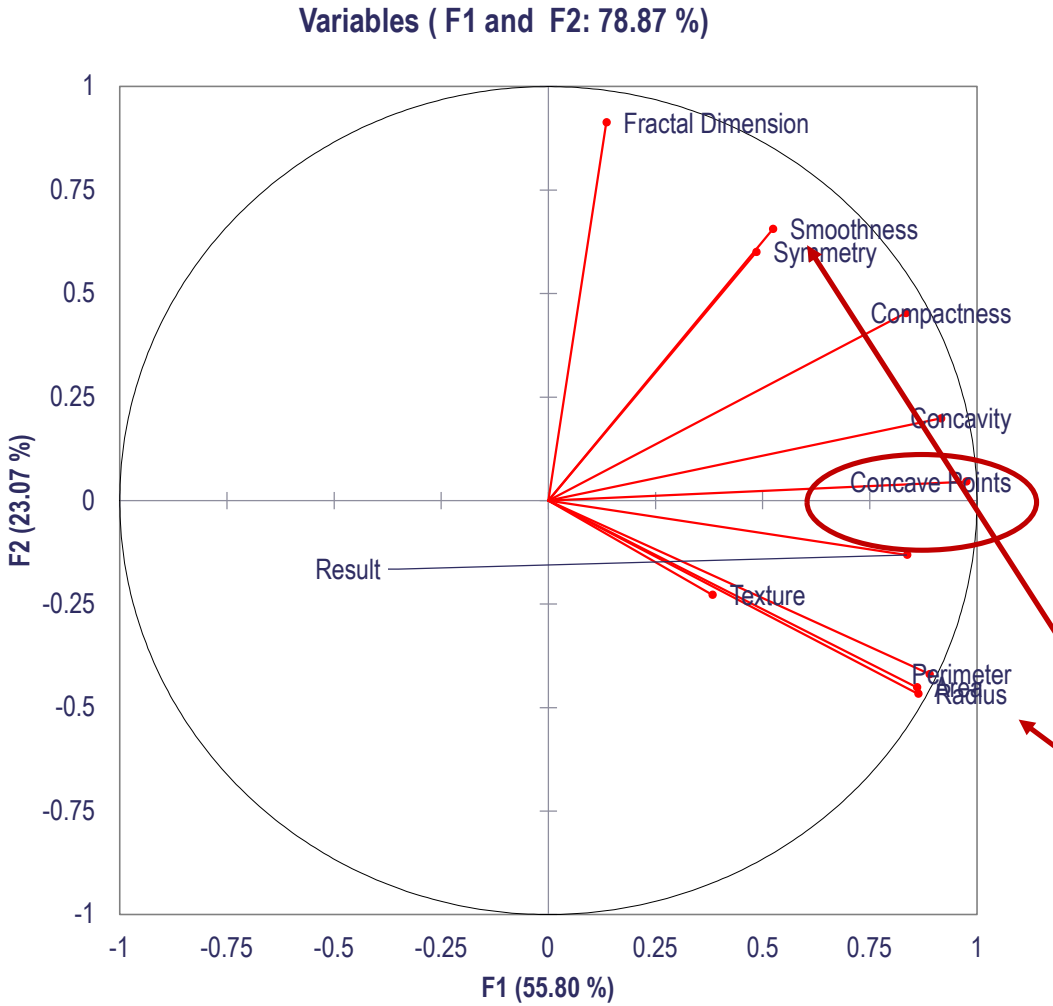
File: Breast Cancer Diagnostic
Data set of 600 breast cancer patients.
Objective: Tumour classifications (maligned or benign).

Tumour imaging is analysed and the following parameters are analysed:

- Radius
- Perimeter
- Area
- Texture
- Softness
- Concave Points
- Concavity
- Symmetry
- Fractal Dimension

Material from R. Ramirez Velarde, Monterrey Tec, 2021

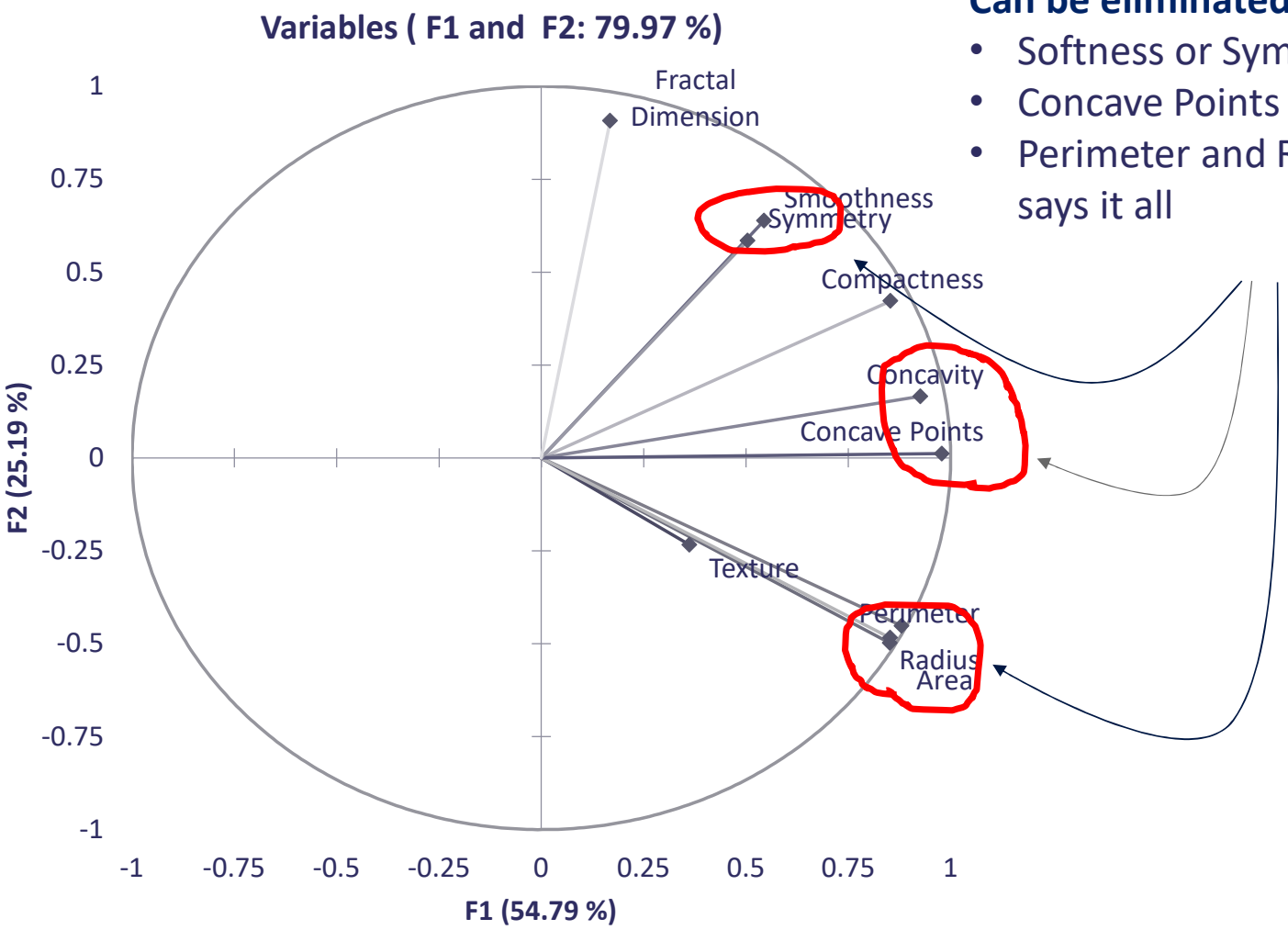
Factors – Dimension Reduction



These variables give the same information

Material from R. Ramirez Velarde, Monterrey Tec, 2021

Factor Loadings (No “Result”)



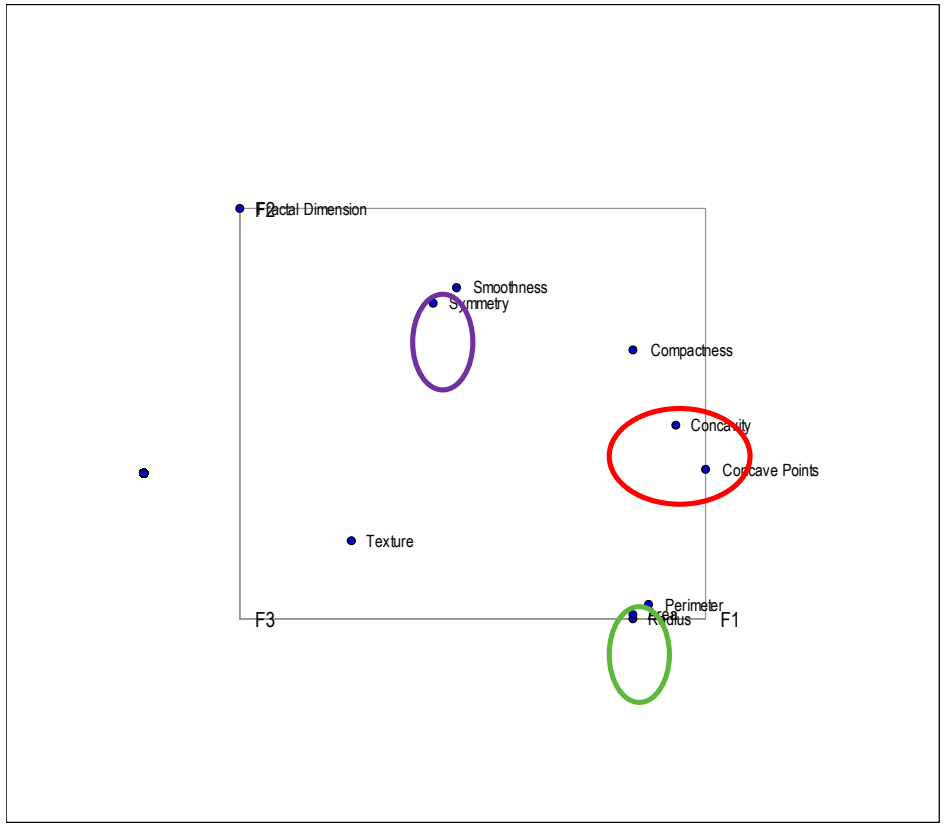
Can be eliminated:

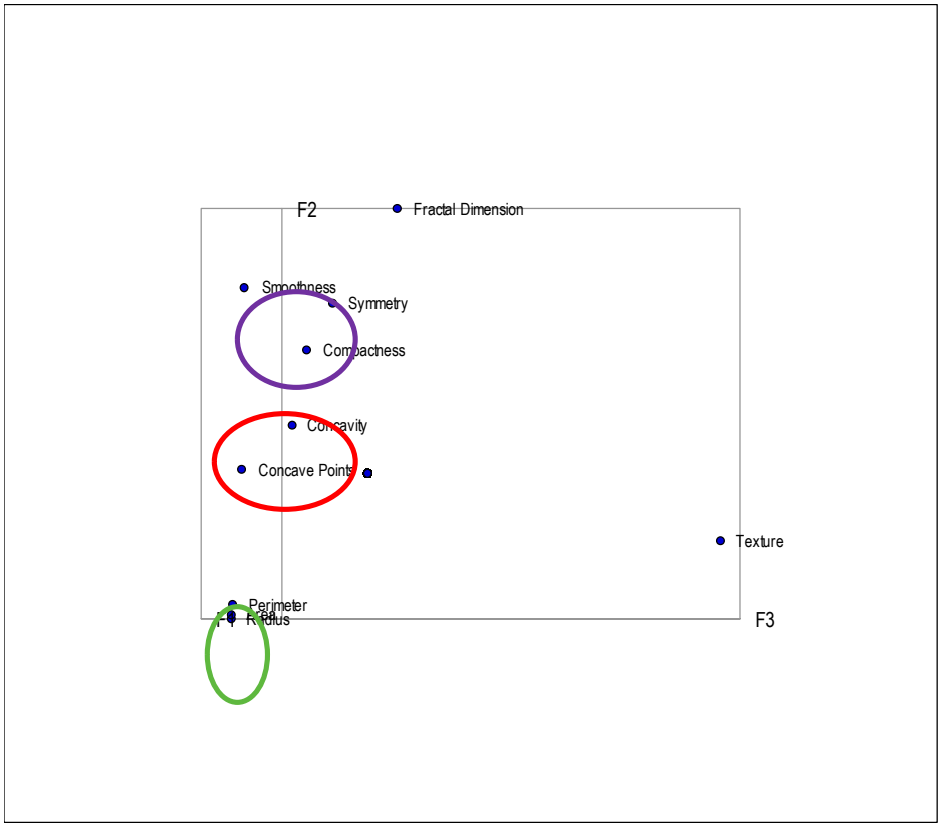
- Softness or Symmetry
- Concave Points or Concavity
- Perimeter and Radius, Diameter says it all

Material from R. Ramirez Velarde, Monterrey Tec, 2021

3D Scatterplot (Rotated in F1)

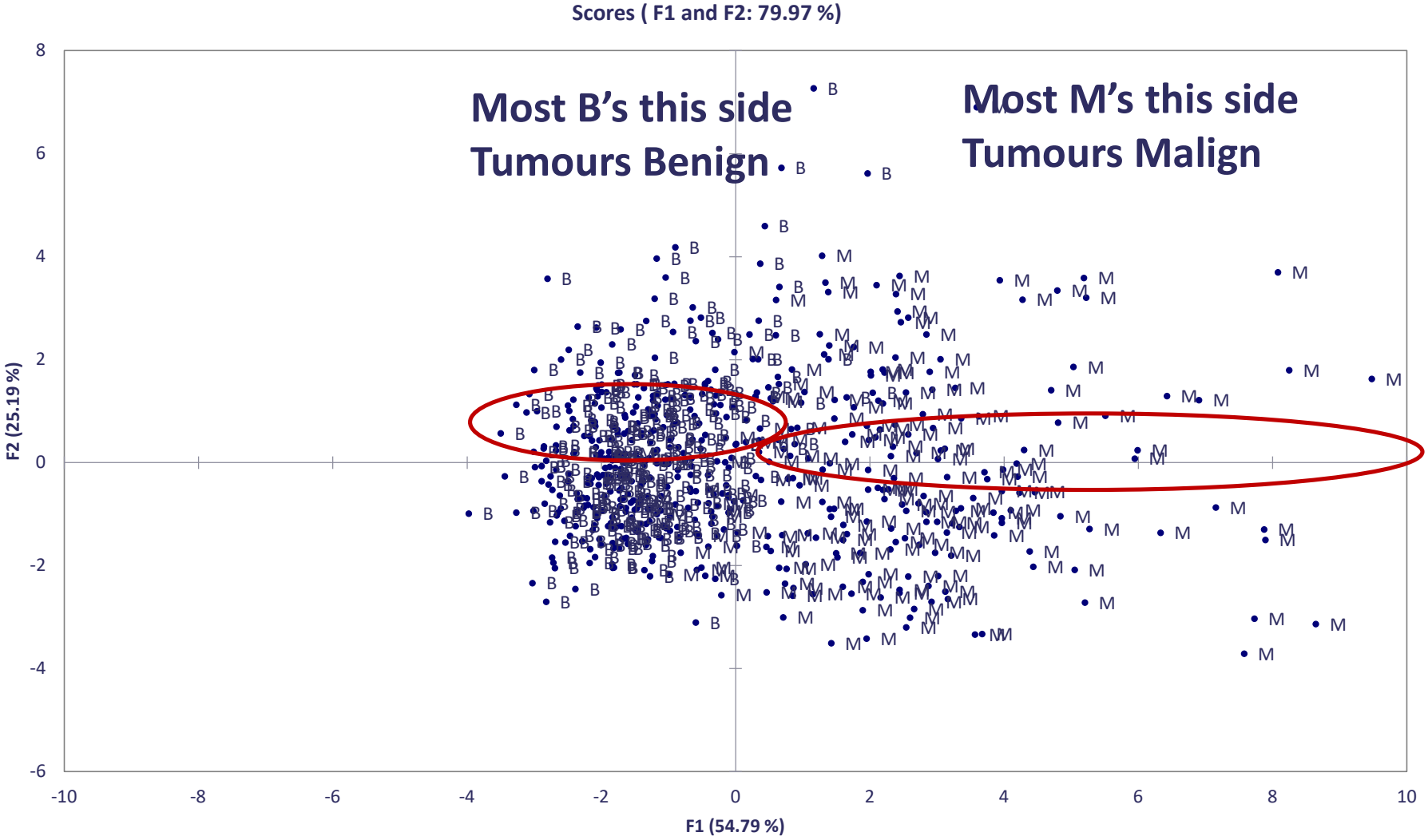
More dimensions might show that some grouping are really not, and others are confirmed





Material from R. Ramirez Velarde, Monterrey Tec, 2021

Plotting Scores – Allow Grouping and Classification



The first principal component completely discriminates groups M and B

Material from R. Ramirez Velarde, Monterrey Tec, 2021

Squared Cosines Table

	F1	F2	F3	F4	F5	F6	F7	F8	F9	F10
Radius	**0.726**	0.248	0.014	0.000	0.000	0.009	0.000	0.000	0.003	0.000
Texture	0.131	0.055	**0.797**	0.000	0.018	0.000	0.000	0.000	0.000	0.000
Perimeter	**0.775**	0.204	0.011	0.000	0.000	0.007	0.001	0.000	0.002	0.000
Area	**0.726**	0.234	0.013	0.000	0.000	0.014	0.005	0.001	0.006	0.000
Smoothness	0.296	**0.407**	0.024	0.006	0.265	0.000	0.000	0.001	0.000	0.000
Compactness	**0.728**	0.178	0.003	0.017	0.021	0.000	0.052	0.000	0.001	0.000
Concavity	**0.858**	0.027	0.001	0.014	0.036	0.045	0.011	0.007	0.000	0.000
Concave Points	**0.957**	0.000	0.004	0.003	0.000	0.009	0.002	0.025	0.000	0.000
Symmetry	0.254	0.342	0.001	**0.398**	0.005	0.000	0.000	0.000	0.000	0.000
Fractal Dimension	0.028	**0.823**	0.011	0.061	0.026	0.040	0.010	0.000	0.000	0.000

The First Principal Component includes almost everything?

Material from R. Ramirez Velarde, Monterrey Tec, 2021

Science and Technology Facilities Council

Hartree Centre

Analysing Weather Information and Pollutants

Clustering and Dimensionality Reduction

Science and Technology Facilities Council

Hartree Centre

Weather and Pollutants

10 weather stations located throughout Monterrey's Metro área

- Pollutants
  - COx
  - NOx
  - SOx
  - **O3**
  - **2.5 PM**
  - **10 PM**
- Weather Variables
  - PRS
  - TOUT
  - HR
  - SR
  - WSP
  - Rain
  - WDR

Square cosines hint causal relations between variables

Material from R. Ramirez Velarde, Monterrey Tec, 2021

Data Sample

	CENTRO	CENTRO	CENTRO	CENTRO	CENTRO	CENTRO	CENTRO	CENTRO	CENTRO	CENTRO	CENTRO	CENTRO	CENTRO	CENTRO
	ppm	ppb	ppb	ppb	ppb	ug/m3	ug/m3	mmhg	mm/hr	%	KW/m2	degC	KMPH	DEG
Date	CO	NO	NO2	NOX	O3	PM10	PM2.5	PRS	RAINF	RH	SR	TOUT	WSR	WDV
01-01-15 00:00	2.39	2.6	8	10.6	4	26	17	716.8	0	97	0.001	4.39	4.9	52
01-01-15 01:00	2.02	3	7.3	10.3	3	44	27	716.1	0.01	97	0.001	4.23	5.3	46
01-01-15 02:00	2.35	4.1	7.7	11.8	3	32	21	715.8	0.01	97	0.001	4.17	4.5	65
01-01-15 03:00	1.92	10.3	8.5	18.8	2	27	8	715.3	0.01	97	0.001	4.24	4.5	82
01-01-15 04:00	1.89	6.9	8	14.9	2	33	5	715	0	97	0.001	4.18	3.9	77
01-01-15 05:00	1.85	7.9	7.4	15.3	1	26	7	714.9	0.01	97	0.001	4.14	2.7	80
01-01-15 06:00	1.99	5.5	7.3	12.8	2	27		714.7	0	97	0.001	4.19	2.5	77
01-01-15 07:00	2	9.7	7.4	17.1	2	20	15	715	0	97	0.002	4.3	3.3	71
01-01-15 08:00	2.02	10.5	7.2	17.7	2	22	10	715.2	0.01	97	0.02	4.49	1.6	85
01-01-15 09:00	2.17	11.5	6.8	18.3	3	25	9	715.2	0	98	0.046	4.77	2.4	52
01-01-15 10:00	2.11	12.7	7	19.7	3	33	14	715	0	98	0.063	5.1	3.1	77
01-01-15 11:00	2.02	14.1	7.5	21.6	3	30	8	714.5	0	99	0.116	5.91	2.6	62

8760 readings, 8 pollutants, 7 weather variables, 10 zones= 1,314,000 records

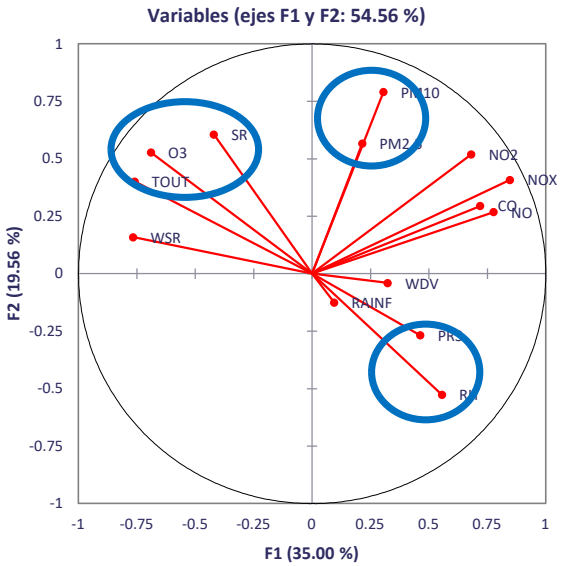
Material from R. Ramirez Velarde, Monterrey Tec, 2021

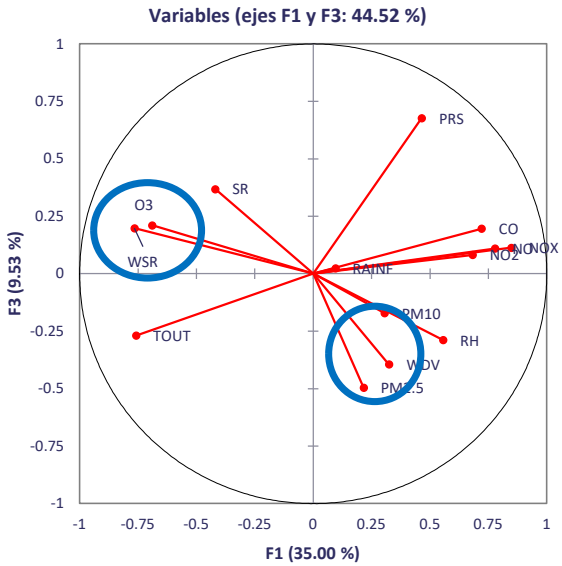
Almost all Pollutants Affected by TOUT and WSR

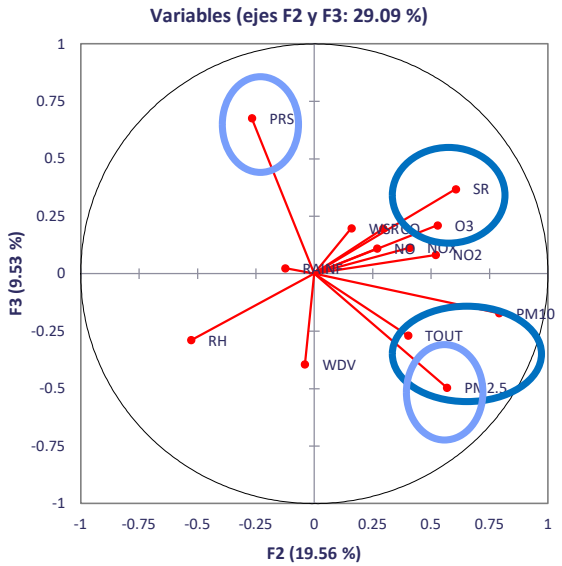
	F1	F2	F3	F4	F5
CO	**0.520**	0.087	0.038	0.000	0.004
NO	**0.605**	0.072	0.012	0.025	0.008
NO2	**0.465**	0.269	0.007	0.001	0.000
NOX	**0.719**	0.167	0.013	0.011	0.004
O3	**0.475**	0.278	0.044	0.001	0.011
PM10	0.094	**0.625**	0.029	0.018	0.020
PM2.5	0.046	**0.322**	0.246	0.120	0.062
PRS	0.215	0.072	0.458	0.006	0.004
RAINF	0.009	0.016	0.001	0.605	0.350
RH	**0.310**	0.278	0.083	0.052	0.050
SR	0.176	**0.366**	0.135	0.000	0.009
TOUT	**0.575**	0.161	0.073	0.009	0.006
WSR	**0.586**	0.025	0.039	0.008	0.000
WDV	0.105	0.002	0.156	0.182	**0.416**

Material from R. Ramirez Velarde, Monterrey Tec, 2021

Scatterplots in 2D Show Possible Causal Relationships







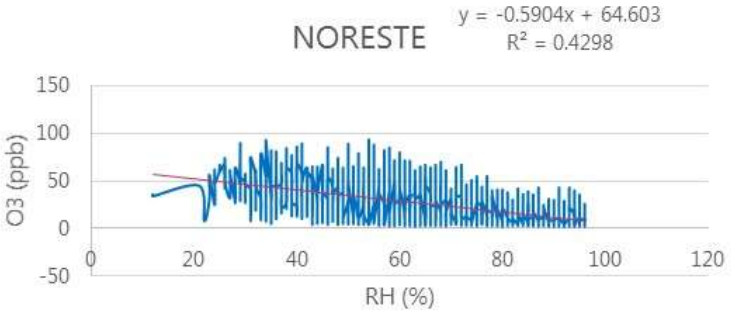
These correlation plots indicate that
O3 will be high when the day is hot (TOUT), dry (RH) and sunny (SR) (very close angles)

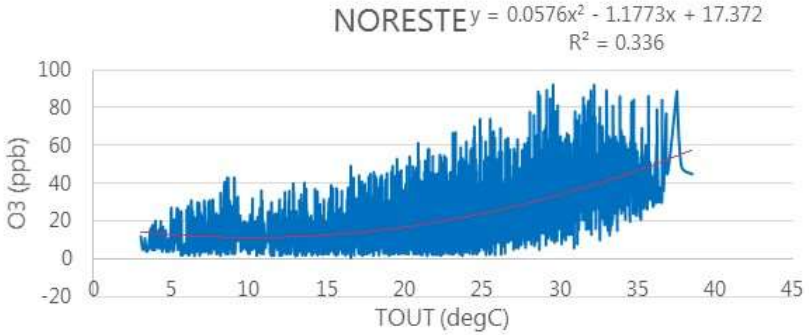
These correlation plots also indicate that that there is almost no relationship between weather variables and pollution by particles PM25 and PM10 (almost 90° angles)

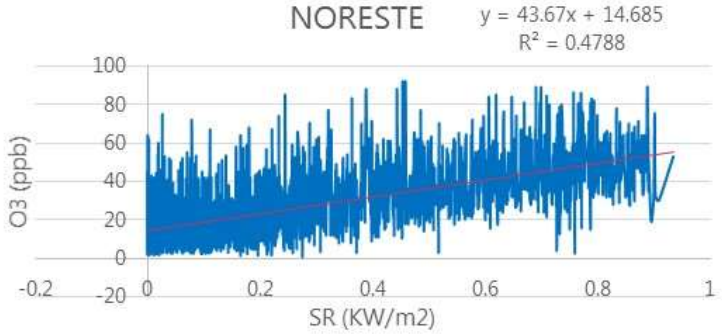
Material from R. Ramirez Velarde, Monterrey Tec, 2021

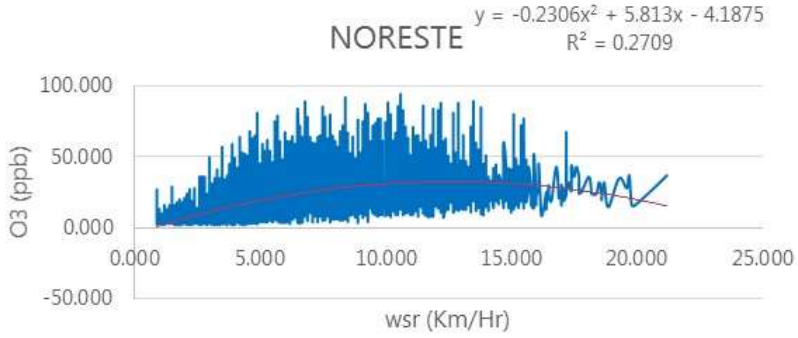
Using the First Principal Component

Normally, PC1 (F1) is correct. The first scatterplots shows relationships for O3 a none for PMx





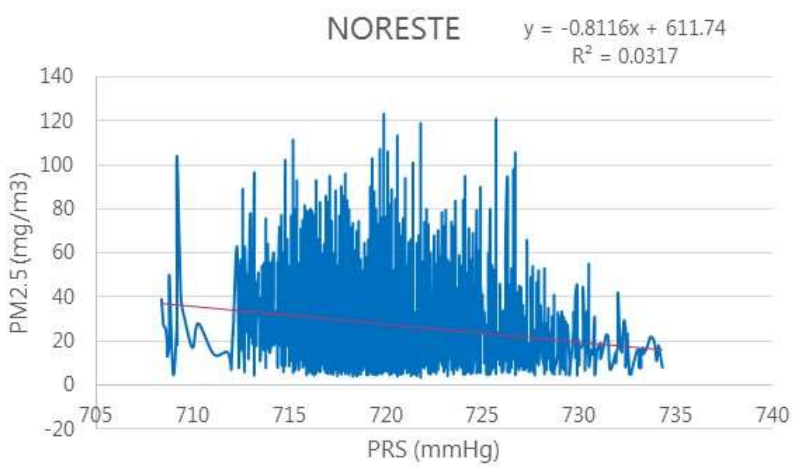


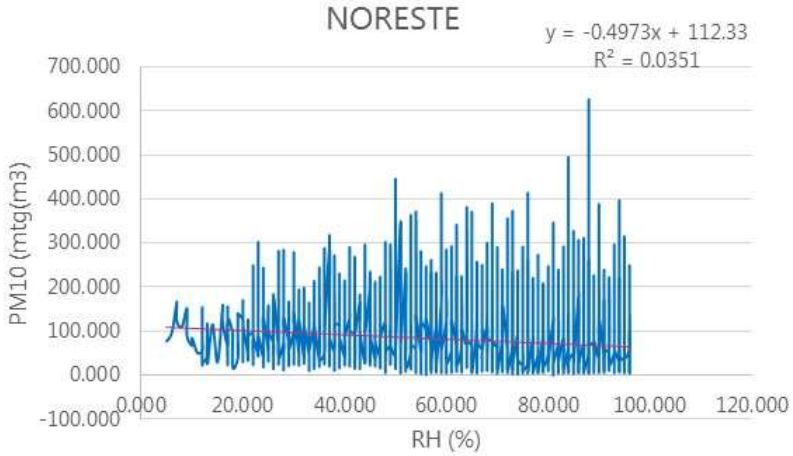


Material from R. Ramirez Velarde, Monterrey Tec, 2021

Using the First Principal Component

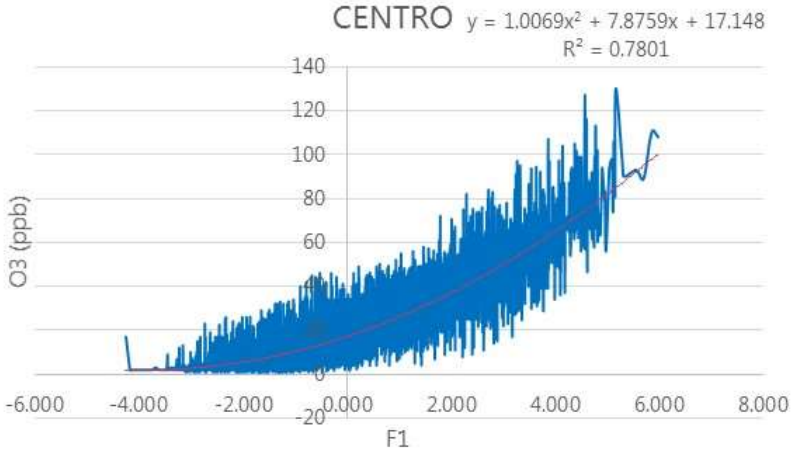
- Other scatterplots seem to show relationships for PM10 or PM2.5
- The relationships are very weak

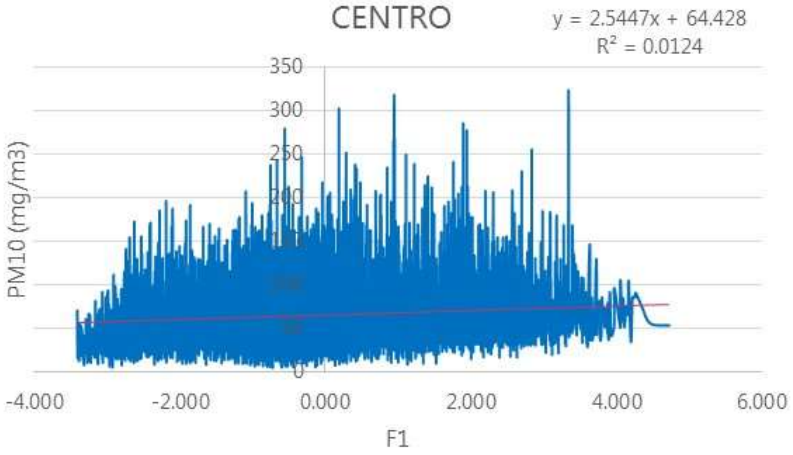




Using the First Principal Component

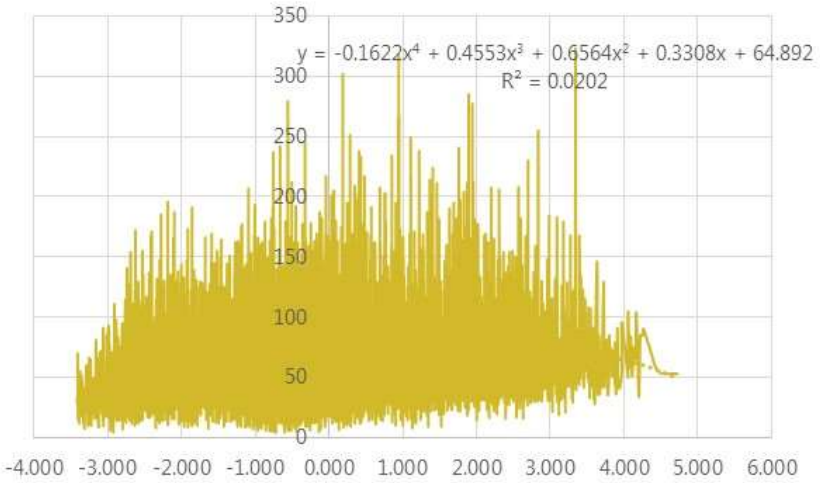
F1 is usually a better predictor



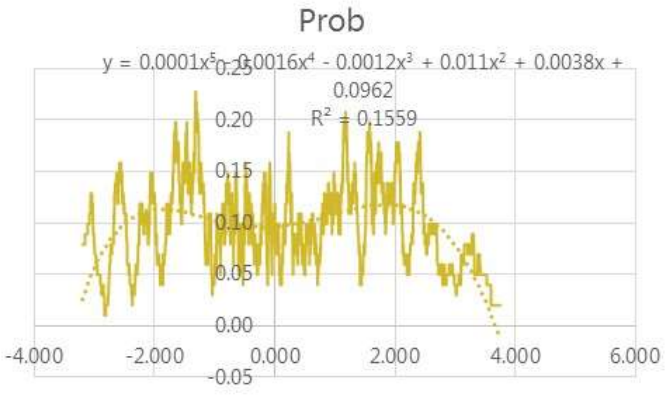


Material from R. Ramirez Velarde, Monterrey Tec, 2021

Uncovering Hidden Relationships PM10



F1 –vs- PM10

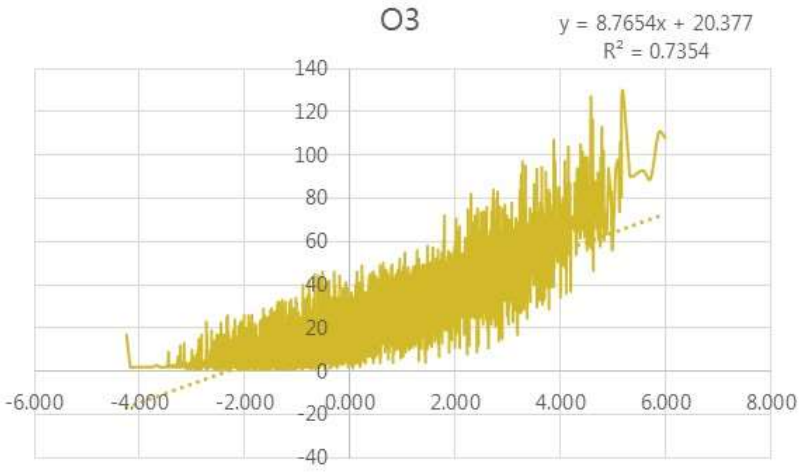


F1 –vs- P[PM10>100 IMECAs]

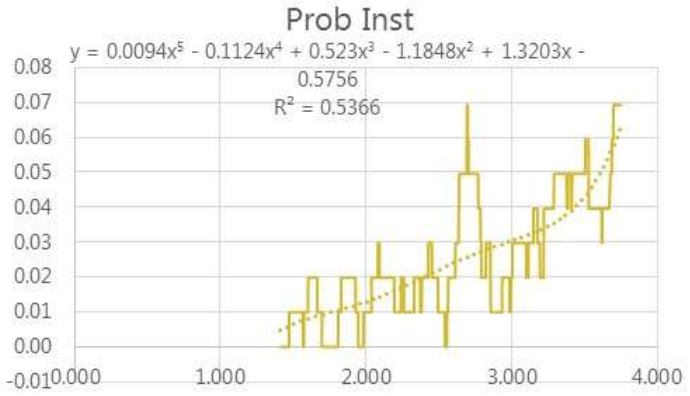
Science and Technology Facilities Council
Hartree Centre

Material from R. Ramirez Velarde, Monterrey Tec, 2021

Uncovering Hidden Relationships O3



F1 –vs- O3



F1 –vs- P[O3>100 IMECAs]

Science and Technology Facilities Council
Hartree Centre

Material from R. Ramirez Velarde, Monterrey Tec, 2021

Science and Technology Facilities Council

Hartree Centre

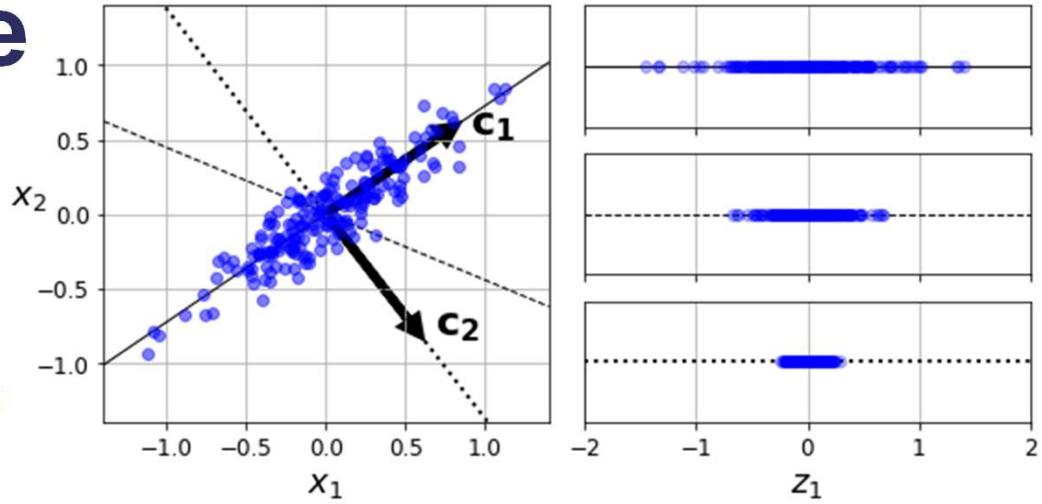
PCA for image analysis

Science and Technology Facilities Council

Hartree Centre

PCA Example

- A simple but important example…
- 200 samples, each with 2 variables
- PCA first finds the line along which sample projections are most spread out (i.e. have the greatest variance)



- This line is represented by a 2D unit vector c_1, which is the first principal component.
- The projection of point i onto c_1 is called it's "factor loading", $z_{1,i}$
- The product $c_1 z_{1,i}$ is an approximation of point i
- The second principal component c_2 is the unit vector, orthogonal to c_1, along which sample projections are most spread out (i.e. have the greatest variance)
- The product of the second principal component and factor loading, $c_2 z_{2,i}$, then provides an offset from $c_1 z_{1,i}$ to reach point 1
- The matrix of PCs, $\mathrm{V} = (c_1 \quad c_2)$ is square (2x2). Because $c_1 \perp c_2$ and both are unit vectors, we have $V^T V = V\,V^T = I$, where I is the identity matrix (exercise)
  - Hence, the design matrix X, factor loading matrix Z and principal component matrix V are related by the equations: So V can be thought of as a transformation between samples and factor loadings.

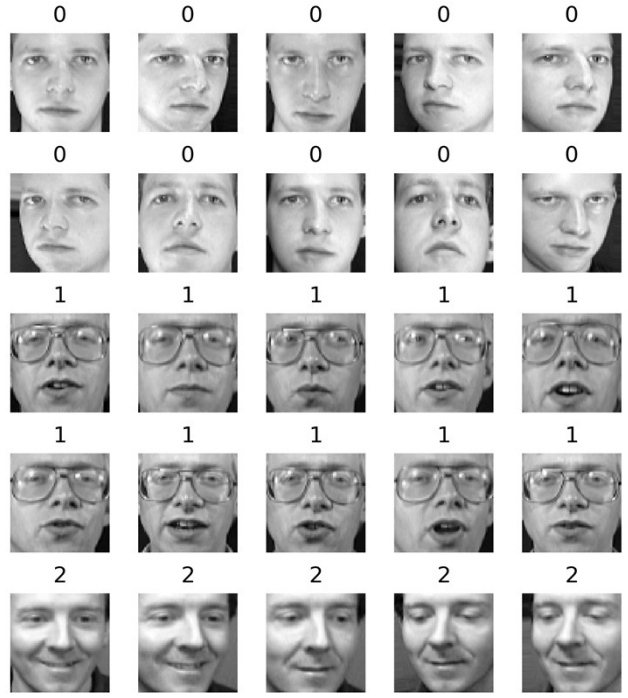
$$\mathrm{Z} = XV, X = ZV^T$$

Science and Technology Facilities Council
Hartree Centre

Olivetti Faces Dataset

- 400 images: 10 for each of 40 distinct subjects.
- Images were taken at different times, varying the lighting, facial expressions (open / closed eyes, smiling / not smiling) and glasses / no glasses
- Typically used to train classification algorithms, but we'll use it for PCA
- Each image has 64 x 64 grayscale pixels. For PCA, it is simply represented as a vector of 4096 unsigned integers
- Hence, the design matrix is wide.
- There are 40 principal components, each of which has dimension 4096
- As in the last slide, $c_1 z_{1,i}$ can be thought of as a first approximation of face i, and thereafter, $c_j z_{j,i}$ for $j = 2, 3, \ldots 400$ can be thought of as incremental refinements of that approximation
- Recall from the previous slide that:

$$Z = XV, X = ZV^T$$



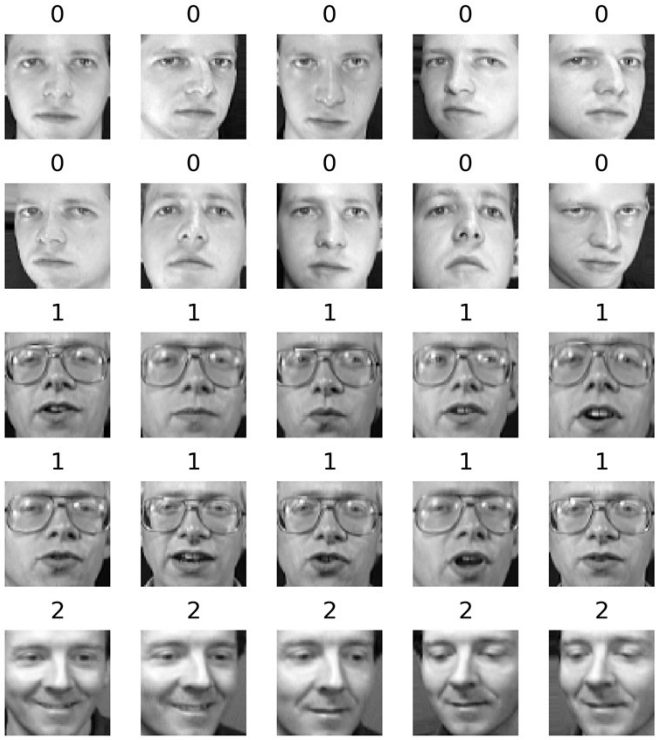
Science and Technology Facilities Council
Hartree Centre

Compression

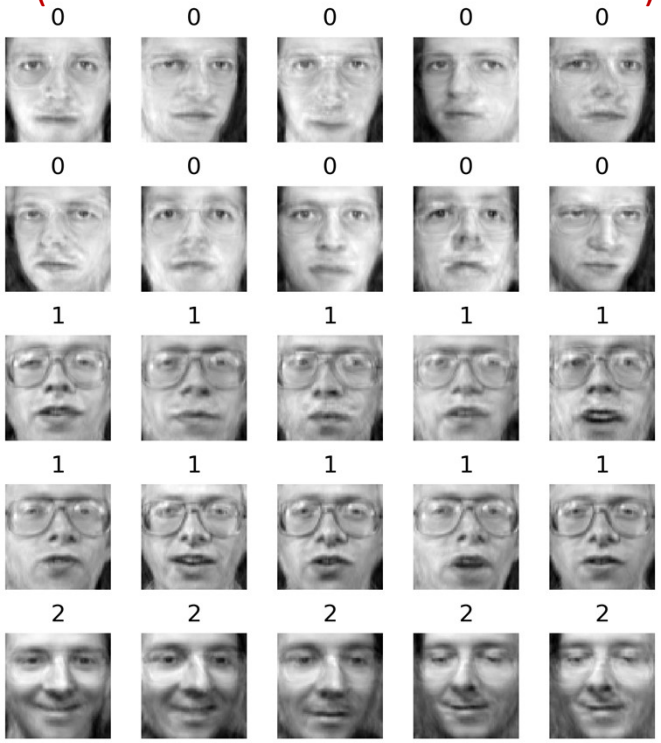
- Including 80 principal components gives a reasonable representation of each face, with 80% compression.
- We've truncated C and F, by removing the last 320 PCs and associated factor loadings, and used these truncated matrices to approximately reconstruct X.

$$Z_{truncated} = XV_{truncated}$$
$$X \approx Z_{truncated}V_{truncated}^{T}$$

True Images



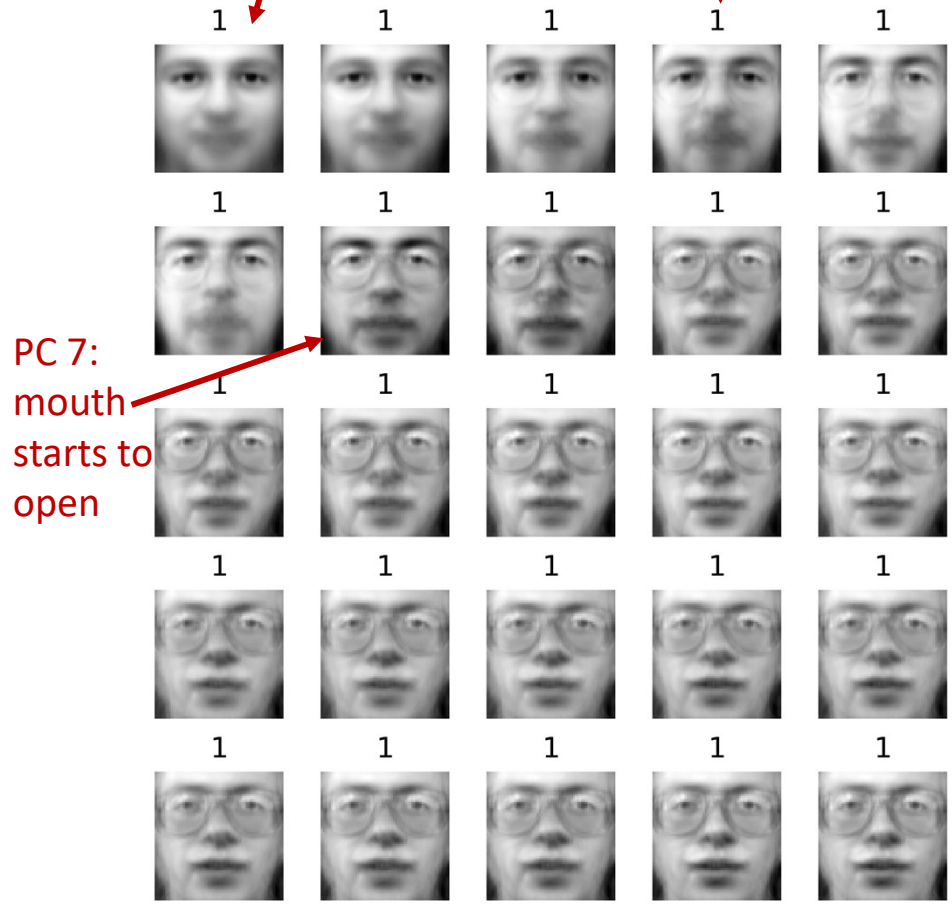
Compressed images
(reconstructed from the first 80 PCs)



Science and Technology Facilities Council
Hartree Centre

Effects of the First 25 PCs

- **Figure on the right shows the cumulative effects of the first 25 PCs and loadings for approximating the face that's circled on the left.**



Science and Technology Facilities Council
Hartree Centre

Science and Technology Facilities Council

Hartree Centre

Comfort Break

Science and Technology Facilities Council

Hartree Centre

Science and Technology Facilities Council

Hartree Centre

Linear Algebra and Optimisation approaches to classification

Science and Technology Facilities Council

Hartree Centre

Linear System

- Linear equation :$\alpha_1 x_1 + \alpha_2 x_2 + \cdots + \alpha_n x_n = \beta$

- System of linear equations:

$$\begin{aligned} \alpha_{11} x_1 + \alpha_{12} x_2 + \cdots + \alpha_{1n} x_n &= \beta_1 \\ \alpha_{21} x_1 + \alpha_{22} x_2 + \cdots + \alpha_{2n} x_n &= \beta_2 \\ &\vdots \\ \alpha_{n1} x_1 + \alpha_{n2} x_2 + \cdots + \alpha_{nn} x_n &= \beta_n \end{aligned}$$

- Matrix form: Ax=B
  - A : coefficient matrix
  - x : variables
  - B : Right hand side constant

Optimisation and linear systems

- Don’t need just to solve a linear system
- Want to find the the best possible outcome (maximise profit or minimize cost)
- Linear objective function
- Linear constraints

Linear programming basic components

- c: Coefficients for the objective function
- A: Coefficient matrix for the constraints
- b: Right-hand side vector for the constraints.
- x: Decision variables to be optimized.

Linear programming formulation

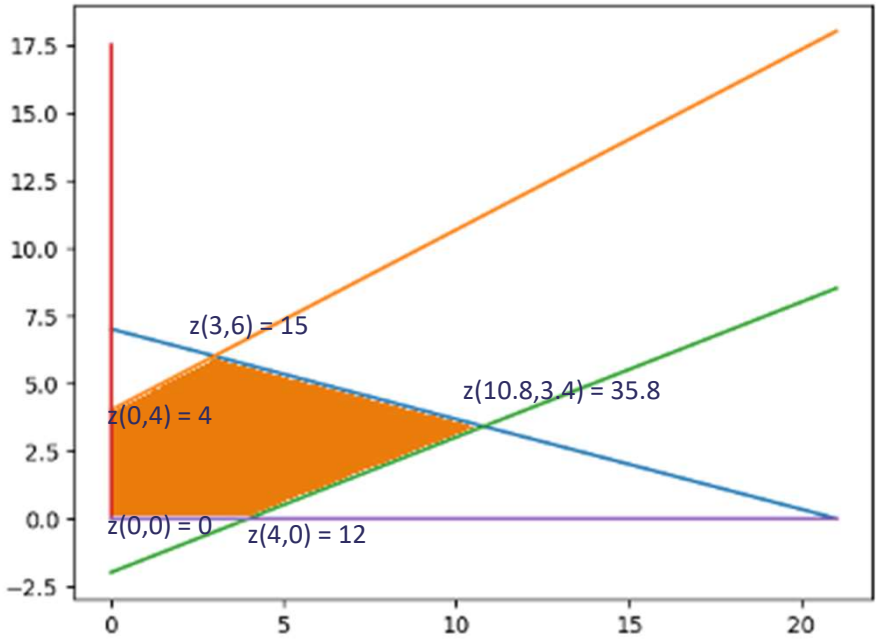
$$
\begin{aligned}
&maximise && c_1x_1 + \cdots + c_nx_n \\
&\text{subject to:} && a_{11}x_1 + \cdots + a_{1n}x_n \leq b_1 \\
& && \vdots \\
& && a_{n1}x_1 + \cdots + a_{nn}x_n \leq b_n \\
& && x_i \geq 0 \; for \; all \; i
\end{aligned}
$$

- *Matrix representation*

$$
\begin{aligned}
&maximise . && c^T \boldsymbol{x} \\
&\text{subject to:} && \mathrm{A}\mathbf{x} \leq \mathrm{b} \\
& && A_{eq}\boldsymbol{x} = b_{eq} \\
& && \boldsymbol{x} \geq 0
\end{aligned}
$$

Example of a linear programming task

$$
\begin{aligned}
&maximise: && 3x + y \\
&subject\ to: && x + 3y \leq 21 \\
& && -2x + 3y \leq 12 \\
& && x - 2y \geq 4 \\
& && x, y \geq 0
\end{aligned}
$$



Science and Technology Facilities Council

Hartree Centre

Primal Dual formulation

- How can we convince ourselves, or another user, that the solution is indeed optimal, without having to trace the steps of the computation of the algorithm?

$maximise\ .\quad c^T\,\boldsymbol{x}$

subject to: $A\boldsymbol{x} \leq$b

$\boldsymbol{x} \geq 0$

minimise . $b^T\,\boldsymbol{y}$

subject to: . $A^T\,\boldsymbol{y} \geq$ c

$\boldsymbol{y} \geq 0\ for\ all\ i$

Primal Dual properties

- The dual of the dual of a linear program is the linear program itself.
- f the primal (in maximization standard form) and the dual (in minimization standard form) are both feasible, then:

$$\text{optimal\_solution}(\text{primal}) \leq optimal\_solution(dual)$$

- For two linear programs in primal dual form (LP1, LP2) if either is feasible and bounded, then so is the other and thus:

$$\text{optimal}_{\text{solution(primal)}} = optimal\_solution(dual)$$

Classification example based on Linear Systems

		a_1	a_2	Linear Transformed Scores $A_i X = b$
$G_{1:}$	1	A_{11}	A_{12}	$A_1 X^*$
	2	A_{21}	A_{22}	$A_2 X^*$
	:			:
	k	A_{k1}	A_{k2}	$A_k X^*$
$G_{2:}$	k+1	$A_{k+1,1}$	$A_{k+2,2}$	$A_{k+1} X^*$
	k+2	$A_{k+2,1}$	$A_{k+2,2}$	$A_{k+2} X^*$
	:			:
	n	A_{n1}	A_{n2}	$A_n X^*$

Science and Technology Facilities Council
Hartree Centre

Y. Shi, Optimization Based Data Mining: Theory and Applications, Springer, 2011, and Y. Shi lecture notes

A Linear Programming Approach to classification

Linear programming has been used for Classification in Data Mining.

Given two attributes $\{a_1 , a_2\}$ and

two groups $\{G_1 , G_2\}$, with the observation

$\boldsymbol{A}_i = (\boldsymbol{A}_{i1}, \boldsymbol{A}_{i2})$,

we want to find a scalar b and nonzero vector $\boldsymbol{X} = (x_1, x_2)$ such that

$\boldsymbol{A}_i \boldsymbol{X} \leq b$, $\boldsymbol{A}_i \in G_1$ and $\boldsymbol{A}_i \boldsymbol{X} \geq b$, $\boldsymbol{A}_i \in G_2$ have the fewest number of violated constraints.

Y. Shi, Optimization Based Data Mining: Theory and Applications, Springer, 2011, and Y. Shi lecture notes

Linear System Approaches Example

Example: Consider a credit-rating problem with two variables and two cases:

a_1 = salary and a_2 = age. Let the boundary b = 10.

Cases (Obs)	a_1 a_2		Boundary b = 10	found best coef. x_1	 x_2	LP Score
A_1	6	8	10	.4	.6	7.2
A_2	8	12	10	.4	.6	10.4

Science and Technology Facilities Council
Hartree Centre

Y. Shi, Optimization Based Data Mining: Theory and Applications, Springer, 2011, and Y. Shi lecture notes

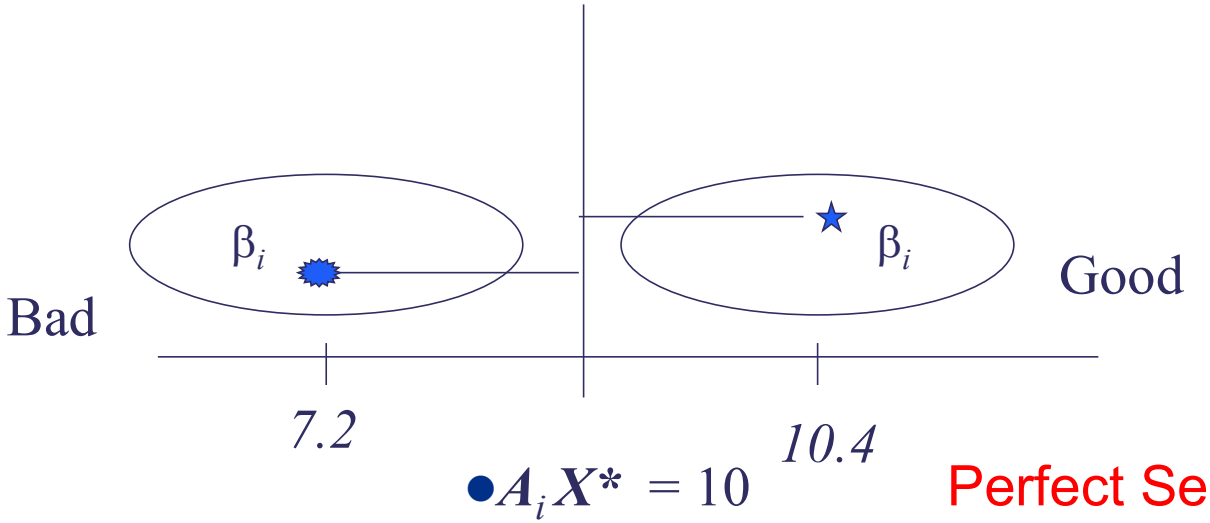
Linear System Approaches

Find the best (x_1^*, x_2^*) to compare

$\boldsymbol{A_i X^*} = a_1 x_1^* + a_2 x_2^*$ with $b = 10$

We see $\boldsymbol{A_1 X^*} = 7.2$ is Bad (< 10) and

$\boldsymbol{A_2 X^*} = 10.4$ is Good (> 10)

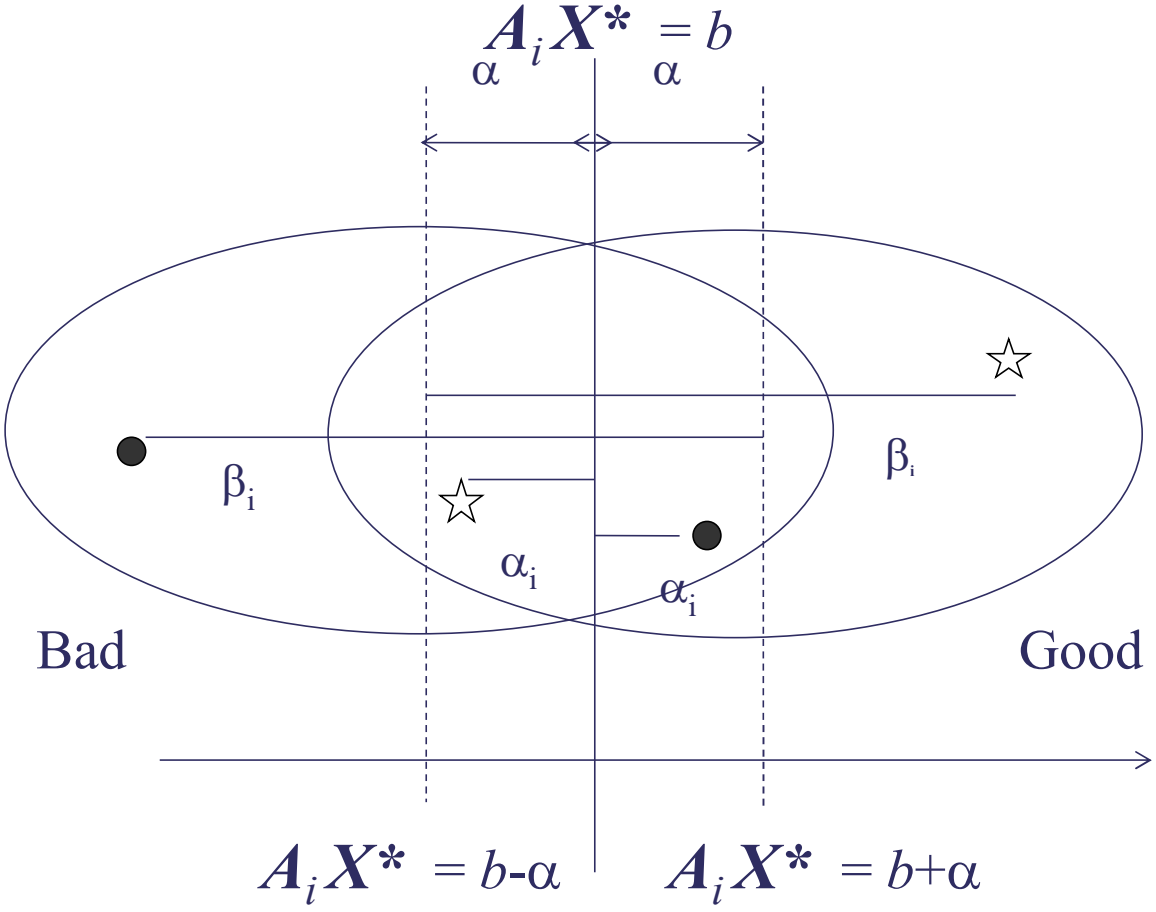


Perfect Separation (α = 0)

Y. Shi, Optimization Based Data Mining: Theory and Applications, Springer, 2011, and Y. Shi lecture notes

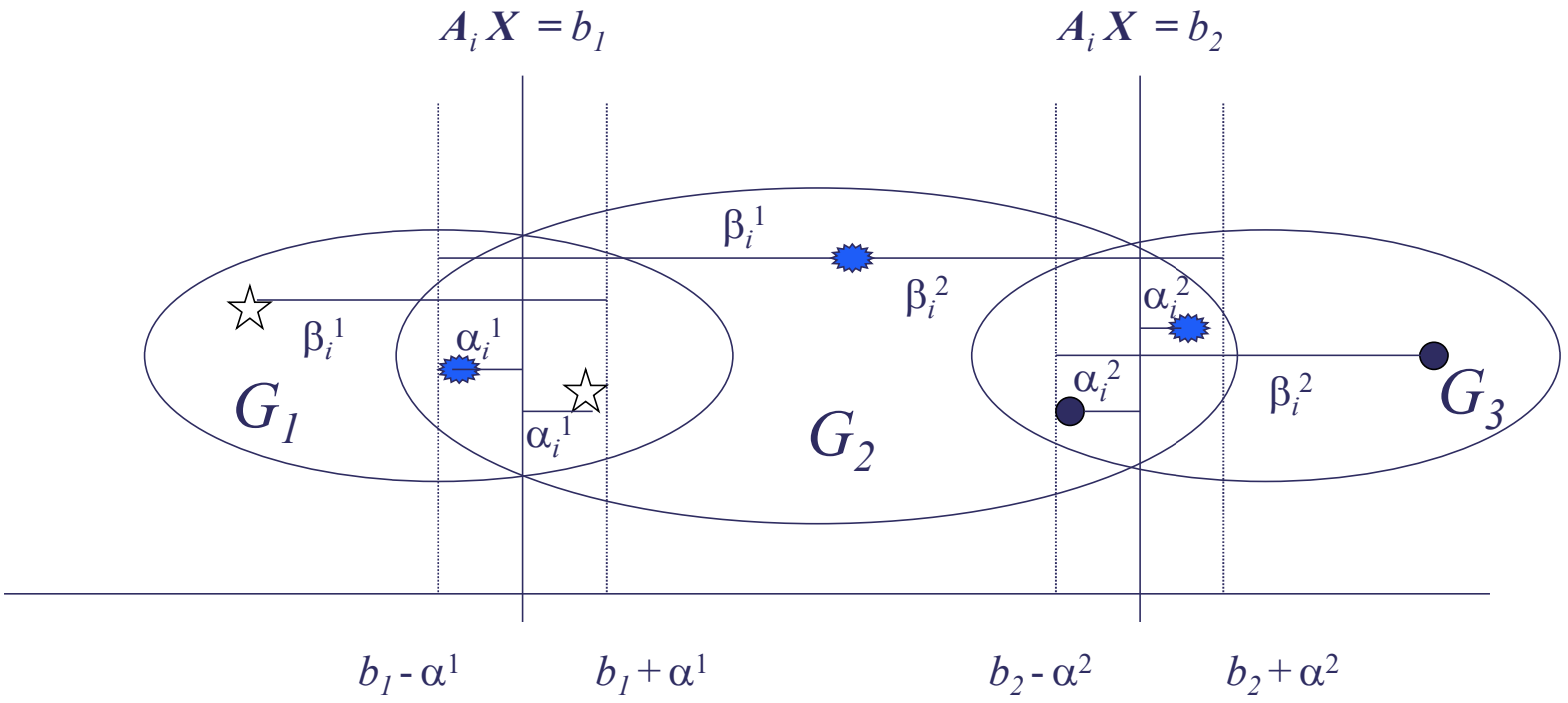
Linear System Approaches

Example: Overlapping



Y. Shi, Optimization Based Data Mining: Theory and Applications, Springer, 2011, and Y. Shi lecture notes

Linear Algebra Approaches: Multi-Criteria Linear Programming



Three-group MC Model

Y. Shi, Optimization Based Data Mining: Theory and Applications, Springer, 2011, and Y. Shi lecture notes

Linear Algebra Approaches

Let

- α_i = the overlapping of two-group (classes) boundary for case $\boldsymbol{A}_i$ (external measurement);
- α = the max overlapping of two-group (classes) boundary for all cases $\boldsymbol{A}_i$ ($\alpha_i < \alpha$);
- β_i = the distance of case $\boldsymbol{A}_i$ from its adjusted boundary (internal measurement);
- β = the min distance of all cases $\boldsymbol{A}_i$ to the adjusted boundary ($\beta_i > \beta$);
- h_i = the penalties for α_i (“cost” of misclassification);
- k_i = the penalties for β_i (“cost” of misclassification);

Y. Shi, Optimization Based Data Mining: Theory and Applications, Springer, 2011, and Y. Shi lecture notes

Linear Algebra Approaches: Multi-Criteria Linear Programming

Multi-Criteria Linear programming considers to simultaneously minimize the total overlapping degree and maximize the total distance from the boundary of two groups (Shi, Wise, Luo, and Lin 2001):

Minimize $\Sigma_i \alpha_i$ and Maximize $\Sigma_i \beta_i$

Subject to

$\boldsymbol{A}_i \boldsymbol{X} = b + \alpha_i - \beta_i, \boldsymbol{A}_i \in B,$

$\boldsymbol{A}_i \boldsymbol{X} = b - \alpha_i + \beta_i, \boldsymbol{A}_i \in G,$

where $\boldsymbol{A}_i$ are given, $\boldsymbol{X}$ and b are unrestricted, and α_i and $\beta_i \geq 0$.

Y. Shi, Optimization Based Data Mining: Theory and Applications, Springer, 2011, and Y. Shi lecture notes

Linear System Approaches: Multi-Criteria Linear Programming

We can find the compromise solution for the separation problems (Yu 1973, Yu 1985, and Shi and Yu 1989):

Let

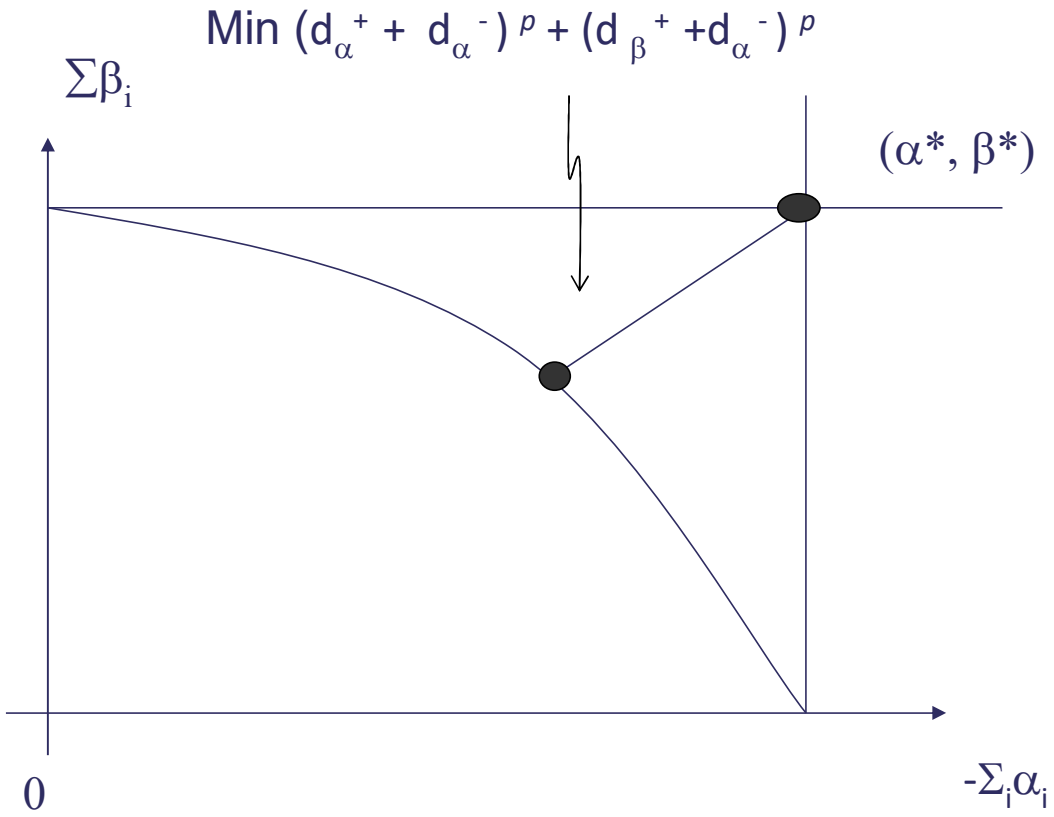
- α^* = the ideal overlapping of $-\Sigma_i \alpha_i$;
- β^* = the ideal distance of $\Sigma_i \beta_i$.

Then, we define the regret function as:

- $-d_\alpha^+ = \Sigma_i \alpha_i + \alpha^*$, if $-\Sigma_i \alpha_i > \alpha^*$; otherwise, it is 0.
- $d_\alpha^- = \alpha^* + \Sigma_i \alpha_i$, if $-\Sigma_i \alpha_i < \alpha^*$;otherwise, it is 0.
- $d_\beta^+ = \Sigma_i \beta_i - \beta^*$, if $\Sigma_i \beta_i > \beta^*$; otherwise, it is 0.
- $d_\alpha^- = \beta^* - \Sigma_i \beta_i$, if $\Sigma_i \beta_i < \beta^*$; otherwise, it is 0.

Y. Shi, Optimization Based Data Mining: Theory and Applications, Springer, 2011, and Y. Shi lecture notes

Multi-Criteria Linear Programming



Y. Shi, Optimization Based Data Mining: Theory and Applications, Springer, 2011, and Y. Shi lecture notes

Linear System Approaches: Multi-Criteria Linear Programming

Thus, the Multi-Criteria separation problem becomes:

$$\text{Min } (d_\alpha^+ + d_\alpha^-)^p + (d_\beta^+ + d_\beta^-)^p$$

Subject to

$$\alpha^* + \Sigma_i \alpha_i = d_\alpha^- - d_\alpha^+,$$

$$\beta^* - \Sigma_i \beta_i = d_\beta^- - d_\beta^+,$$

$$\boldsymbol{A}_i \boldsymbol{X} = b + \alpha_i - \beta_i, \boldsymbol{A}_i \in G,$$

$$\boldsymbol{A}_i \boldsymbol{X} = b - \alpha_i + \beta_i, \boldsymbol{A}_i \in B,$$

where $\boldsymbol{A}_i$, α^*, and β^* are given, $\boldsymbol{X}$ and b are unrestricted, and $\alpha_i, \beta_i, d_\alpha^-, d_\alpha^+, d_\beta^-, d_\beta^+ \geq 0$.

Y. Shi, Optimization Based Data Mining: Theory and Applications, Springer, 2011, and Y. Shi lecture notes

Linear System Approaches: Multi-Criteria Linear Programming

Algorithm:

- *Step 1* Use *ReadCHD* to convert both Training and Verifying data into data mat.
- *Step 2* Use *GroupDef* to divide the observations within Training data sets into *s* groups: G1, G2, …., and Gs.
- *Step 3* Use *sGModel* to perform the separation task on the training data. Here, PROC LP is called to calculate the MCLP model for the best solution of the *s*-group classifier given the values of control parameters.
- *Step 4* Use *Score* to produce the graphical representations of training results. Step 3-4 will not terminate until the best training result is found.
- *Step 5* Use *Predict* to mine the *s* groups from Verifying data set.

Y. Shi, Optimization Based Data Mining: Theory and Applications, Springer, 2011, and Y. Shi lecture notes

Example: A two-class data set of customer status

Cases	Age	Income	Student	Credit Rating	Class: buys_ computer	Training results
A_1	31… 40	high	no	fair	yes	success
A_2	>40	medium	no	fair	yes	success
A_3	> 40	low	yes	fair	yes	success
A_4	31… 40	low	yes	excellent	yes	success
A_5	≤30	low	yes	fair	yes	success
A_6	>40	medium	yes	fair	yes	success
A_7	≤30	medium	yes	excellent	yes	success
A_8	31… 40	medium	no	excellent	yes	*failure*
A_9	31… 40	high	yes	fair	yes	success
A_{10}	≤30	high	no	fair	no	success
A_{11}	≤30	high	no	excellent	no	success
A_{12}	>40	low	yes	excellent	no	*failure*
A_{13}	≤30	medium	no	fair	no	success
A_{14}	>40	medium	no	excellent	no	success

Y. Shi, Optimization Based Data Mining: Theory and Applications, Springer, 2011, and Y. Shi lecture notes

Multi-Criteria Non-linear Programming

(Kou, Peng, Yan, Shi and Chen 2005):

Let $\zeta_{i,j}$ = the distance from case $\boldsymbol{A}_i$ to b_j

$$\text{(Model 1) } \textit{Minimize } w_\alpha \sum_{j=1}^{k}\sum_{i=1}^{n} \left| \alpha_{i,j} \right|_p - w_\zeta \left(\sum_{j=1 \text{ or } j=k}\sum_{i=1}^{n} \left| \zeta_{i,j} \right|_p - \sum_{j=2}^{k-1}\sum_{i=1}^{n} \left| \frac{b_j - b_{j-1}}{2} - \zeta_{i,j} \right|_p \right)$$

Subject to:

$$\boldsymbol{A}_i\boldsymbol{X} = b_j + \alpha_{i,j} - \zeta_{i,j},\ 1 \le j \le k-1 \qquad (4)$$

$$\boldsymbol{A}_i\boldsymbol{X} = b_{j-1} - \alpha_{i,j-1} + \zeta_{i,j-1},\ 2 \le j \le k \qquad (5)$$

$$\zeta_{i,j} = b_j - b_{j-1},\ 2 \le j \le k \qquad (a)$$

$$\zeta_{i,j} = b_{j+1} - b_j,\ 1 \le j \le k-1 \qquad (b)$$

where $\boldsymbol{A}_i$ is given, $\boldsymbol{X}$ and b_j are unrestricted, and $\alpha_{i,j}, \zeta_{i,j} \ge 0, 1 \le i \le n$..

Y. Shi, Optimization Based Data Mining: Theory and Applications, Springer, 2011, and Y. Shi lecture notes

Multi-Criteria Non-linear Programming

Let $p = 2$, then objective function in Model 1 can now be a quadratic objective and we have:

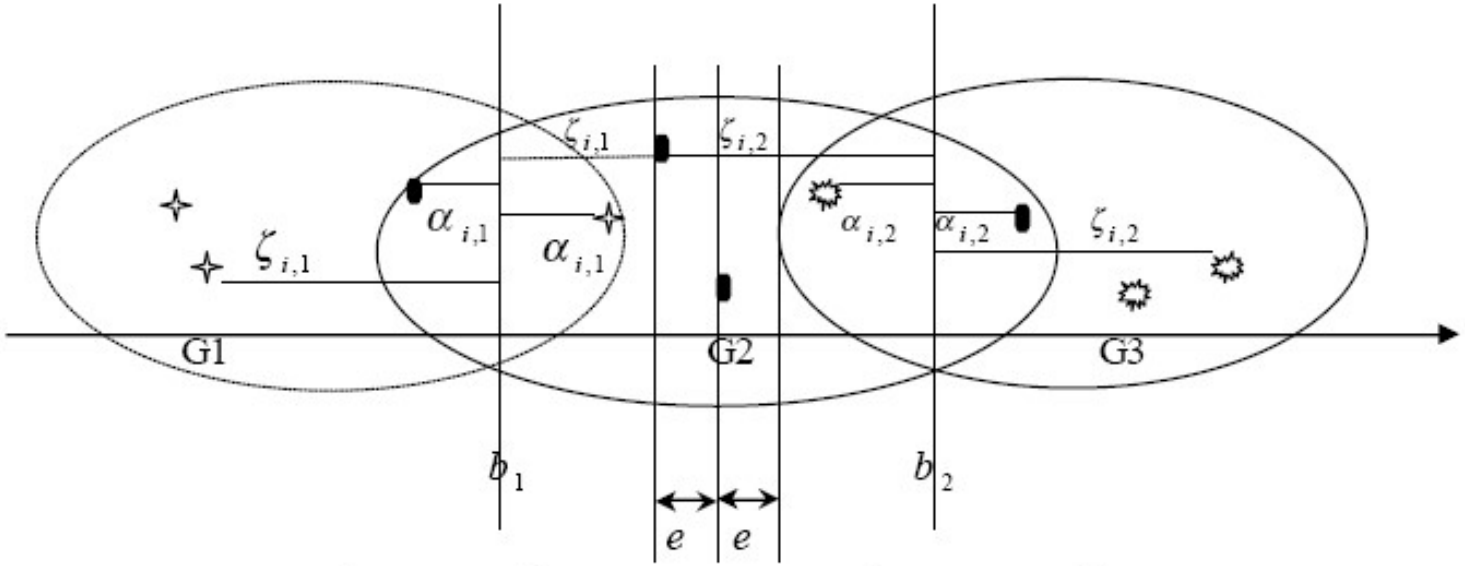
(Model 2)

$$Minimize\ w_\alpha \sum_{j=1}^{k}\sum_{i=1}^{n}(\alpha_{i,j})^2 - w_\zeta\Big(\sum_{j=1 or j=k}\sum_{i=1}^{n}(\zeta_{i,j})^2 - \sum_{j=2}^{k-1}\sum_{i=1}^{n}[(\zeta_{i,j})^2 - (b_j - b_{j-1})\zeta_{i,j}]\Big) \quad (6)$$

Science and Technology Facilities Council
Hartree Centre

Y. Shi, Optimization Based Data Mining: Theory and Applications, Springer, 2011, and Y. Shi lecture notes

Non-linear Multi-Criteria Programming



$A_iX = b_j + \alpha_{i,j} - \zeta_{i,j}$, $j = 1,2$ $A_iX = b_{j-1} - \alpha_{i,j-1} + \zeta_{i,j-1}$, $j = 2,3$

Figure. 1 A Three-classes Model

Science and Technology Facilities Council
Hartree Centre

Y. Shi, Optimization Based Data Mining: Theory and Applications, Springer, 2011, and Y. Shi lecture notes

Non-linear Multi-Criteria Programming

(Model 3) *Minimize* (6)

Subject to: (c) and (d)

$$A_i X = b_j - \zeta_{i,j},\ 1 \le j \le k-1 \qquad (7)$$

$$A_i X = b_{j-1} + \zeta_{i,j-1},\ 2 \le j \le k \qquad (8)$$

where A_i is given, X and b_j are unrestricted, and $\alpha_{i,j}, \zeta_{i,j} \ge 0, 1 \le i \le n..$

(Model 4) *Minimize* (6)

Subject to: (c) and (d)

$$A_i X = b_j - \alpha_{i,j} - \zeta_{i,j},\ 1 \le j \le k-1 \qquad (9)$$

$$A_i X = b_{j-1} + \alpha_{i,j-1} + \zeta_{i,j-1},\ 2 \le j \le k \qquad (10)$$

Y. Shi, Optimization Based Data Mining: Theory and Applications, Springer, 2011, and Y. Shi lecture notes

Other Multi-Criteria Non-linear Programming

(Zhang, Zhang and Shi 2005):

- $Min\ w_\alpha || \alpha ||_p^p - w_\beta || \beta ||_p^p = f(x, \alpha, \beta, b) = f(\omega)$
- s. t.

$$\boldsymbol{A}_i \boldsymbol{X} - \alpha_i + \beta_i - b = \boldsymbol{0},\ \boldsymbol{A}_i \in G_1$$

$$\boldsymbol{A}_i \boldsymbol{X} + \alpha_i - \beta_i - b = \boldsymbol{0},\ \boldsymbol{A}_i \in G_2 \qquad (1)$$

$$\alpha_i, \beta_i \geq 0,\ i=1,\ldots,n$$

- $\alpha = (\alpha_1, \ldots, \alpha_n)^T, \beta = (\beta_1, \ldots, \beta_n)^T$
- $\omega = (x, \alpha, \beta, b)$

Y. Shi, Optimization Based Data Mining: Theory and Applications, Springer, 2011, and Y. Shi lecture notes

Other Multi-Criteria Non-linear Programming

Let

- $\Omega=\{(x, \alpha, \beta, b)$: $\boldsymbol{A}_i \boldsymbol{X} - \alpha_i + \beta_i - b = \boldsymbol{0}$; $\boldsymbol{A}_i \in G_1$
  $\boldsymbol{A}_i \boldsymbol{X} + \alpha_i - \beta_i - b = \boldsymbol{0}$, $\boldsymbol{A}_i \in G_2$, α_i , $\beta_i \geq 0$ },

Then,

$$X_\Omega = \begin{cases} 0, \omega \in \Omega \\ +\infty, \omega \notin \Omega \end{cases}$$

Y. Shi, Optimization Based Data Mining: Theory and Applications, Springer, 2011, and Y. Shi lecture notes

Other Multi-Criteria Non-linear Programming

(1) can be written as:

- $Min \quad f(\omega) + X_{\Omega}(\omega) = g(\omega) - h(\omega) \qquad (2)$
- $g(\omega) = 0.5\rho\|\omega\|^2 + w_{\alpha} \| \alpha \|_p^p + X_{\Omega}(\omega)$
- $h(\omega) = 0.5\rho\|\omega\|^2 + w_{\beta} \| \beta \|_p^p$
- where $g(\omega)$, $h(\omega)$ are strong quadratic functions.

Y. Shi, Optimization Based Data Mining: Theory and Applications, Springer, 2011, and Y. Shi lecture notes

Other Multi-Criteria Non-linear Programming

Algorithm:

- Given $\omega^0 \in R^{2n+m+1}$, ($m=|G_1 U G_2|$), for every ω^k, we solve the following quadratic program for finding ω^{k+1} :
- quadratic program (Q^k),
- min $\{0.5\rho\|\omega\|^2 + w_\alpha \| \alpha \|_p^p - (H'(\omega^k), \omega)\}$
- $H'(\omega)$ gradient of $H(\omega)$.
- The termination criterion is
- *whenever* $\|\omega^{k+1} - \omega^k\| \leq \boldsymbol{\varepsilon}$

Y. Shi, Optimization Based Data Mining: Theory and Applications, Springer, 2011, and Y. Shi lecture notes

Other Multi-Criteria Non-linear Programming

Theorem:

The series of $\{\omega^0, \ldots \omega^k\}$ resulted from the above algorithm converges to a local minimum of Problem (1).

Y. Shi, Optimization Based Data Mining: Theory and Applications, Springer, 2011, and Y. Shi lecture notes

Other Multi-Criteria Non-linear Programming

More generally, we can consider

$$Min\ w_{\alpha}\| \alpha \|_{p}^{p} - w_{\beta}\| \beta \|_{q}^{q} \qquad (3)$$

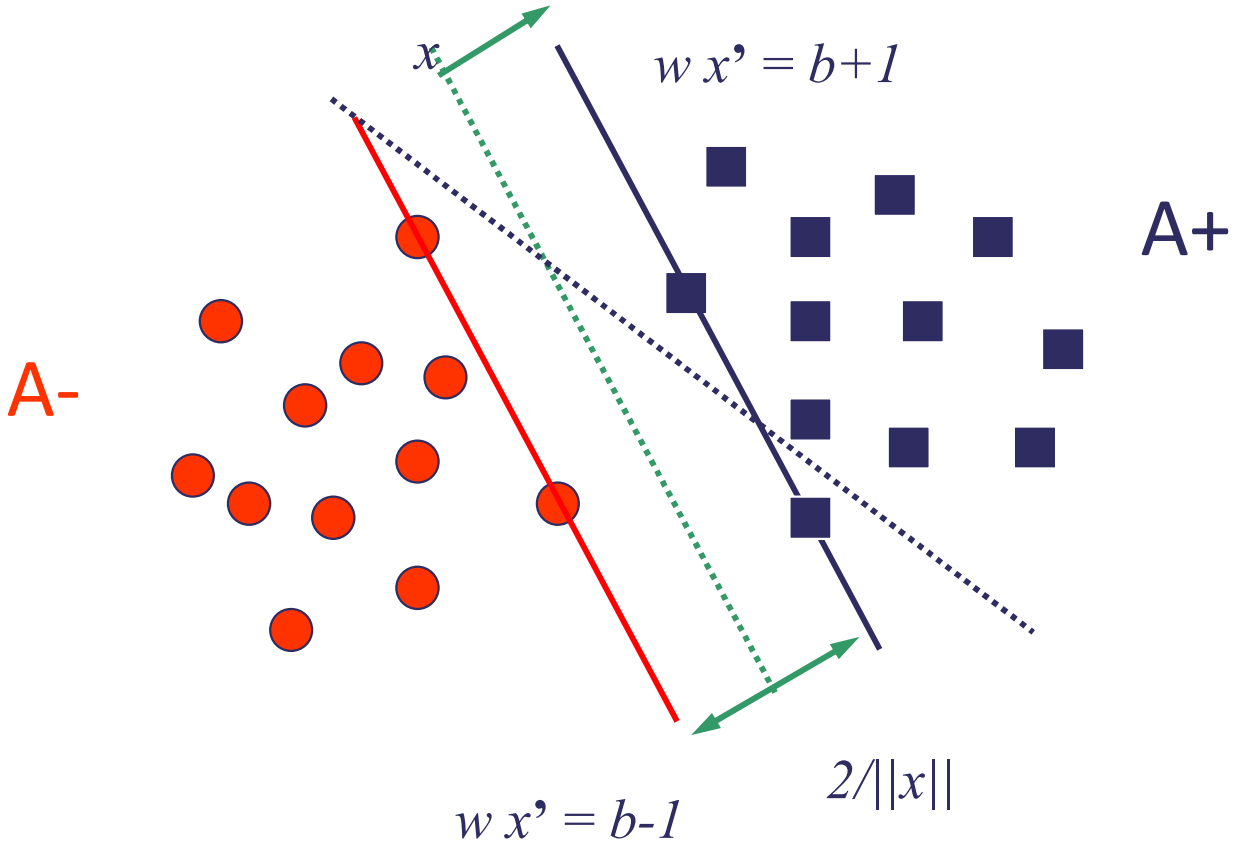
$$s.\ t.\ \omega \in \Omega$$

Then, we observe that:

- *When* $p=q=1$, (3) is a linear program;
- *When* $p=2$, $q=1$, (3) is a quadratic program (many methods can be applied to solve it);
- *When* $p=1$, $q=2$, (3) is a quadratic program and the above algorithm can be terminated at a local minimum of (3) with finite steps.

Y. Shi, Optimization Based Data Mining: Theory and Applications, Springer, 2011, and Y. Shi lecture notes

Support Vector Machines
Maximize the Margin between Bounding Planes
(adapted from Olvi. L. Mangasarian)



What is a SVM?

- **Definition:**
  - SVM is a supervised machine learning algorithm used for both **classification** and **regression** tasks. It is primarily used for binary classification problems.
- **Key Objective:**
  - SVM aims to find the **optimal hyperplane** that best separates two classes in the feature space.
- **Hyperplane:**
  - A hyperplane is a decision boundary that divides the data points into two classes. In a 2D space, it is a line; in higher dimensions, it is a plane.
- **How SVM Works:**
  - It finds the hyperplane that maximizes the **margin** between the closest points from both classes (called **support vectors**).
  - The larger the margin, the better the generalization to new, unseen data.

Primal Formulation of SVM

- **Objective:** Minimize the loss while maximizing the margin between two classes.
- Given training data (x_i, y_i) where x_i is the feature vector and $y_i \in \{-1,1\}$, find a hyperplane $w^T x + b = 0$
- The primal problem seeks to minimize:

$$\min_{w,b,\xi} \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{n} \xi_i$$

$$\text{s.t. } y_i(w^T x_i + b) \geq 1 - \xi_i, \xi_i \geq 0$$

Dual Formulation of SVM

- **Objective:** Solve the problem more efficiently and allow kernel trick for non-linear separation..
- The primal problem can be converted to its dual form, which is a **Quadratic Programming (QP)** problem:

$$\max_{a_i} \sum_{i=1}^{n} a_i - \sum_{i=1}^{n}\sum_{j=1}^{n} a_i a_j y_i y_j x_i^T x_j$$

$$\text{s.t. } 0 \le a_i \le C, \sum_{i=1}^{n} a_i y_i = 0$$

Limitations of Linear SVMs

- **Linear SVM Objective:**
  - The SVM aims to find a linear hyperplane to separate two classes. However, not all datasets are linearly separable in the original feature space.
- **Challenges in Non-Linear Data:**
  - When the data cannot be separated by a straight line (hyperplane), linear SVMs fail to find a solution.
- Example: Consider a circularly distributed dataset where the classes are enclosed within one another.
- **Need for a Higher-Dimensional Solution:**
  - We need to map the data to a higher-dimensional space where a hyperplane **can** separate the classes.

Introducing the Kernel Trick

- **Idea Behind the Kernel Trick:**
  - Instead of explicitly transforming data into higher dimensions (which is computationally expensive), the **kernel function** computes the dot product of the transformed features in the higher-dimensional space directly.
- The decision function in the dual form depends on the dot product between pairs of input vectors:

$$f(x) = \sum_i a_i y_i x_i^T x$$

The **Kernel Trick** replaces the dot product $x_i^T x$ with a kernel function :

$$f(x) = \sum_i a_i y_i K(x_i \ \ x_j)$$

Standard Support Vector Machine

Algebra of 2-Class Linearly Separable Case **(adapted from Olvi. L. Mangasarian)**

Given n points in r dimensional space

Represented by an n-by-r matrix *A*

The membership of each $\boldsymbol{A}_i$ in class +1 or –1 specified by:

An *nxn* diagonal matrix *D* with +1 & -1 entries

Separated by two bounding planes $w\,x' = b \pm 1$;

$$\boldsymbol{A}_i \boldsymbol{X} \geq b + 1, \boldsymbol{D}_{ii} = +1,$$
$$\boldsymbol{A}_i \boldsymbol{X} \leq b - 1, \boldsymbol{D}_{ii} = -1.$$

Then

$D\,(AX - eb) \geq \boldsymbol{e}$, where e is a vector of ones

Y. Shi, Optimization Based Data Mining: Theory and Applications, Springer, 2011, and Y. Shi lecture notes

Standard Support Vector Machine Formulation (adapted from Olvi. L. Mangasarian)

Solve the quadratic program for some $\tau > 0$:

$\text{Min } (\tau/2)\ ||\beta||^2 + (1/2)\ ||x,\ b||^2$

s.t.

$D\ (AX - eb) \geq \boldsymbol{e} - \beta$, where e is a vector of ones

Margin is maximized by minimizing $(1/2)\ ||x,\ b||^2$

Y. Shi, Optimization Based Data Mining: Theory and Applications, Springer, 2011, and Y. Shi lecture notes

Connections between MCLP and Support Vector Machines (SVM)

Let the Membership of each $\boldsymbol{A}_i$ in class +1 or –1 specified by: An *n-by-n* diagonal matrix *D* with +1 & -1 entries in MCLP Problem.

Then, the two-class constraints become:

$\boldsymbol{A}_i \boldsymbol{X} \leq b + \alpha_i - \beta_i$, $\boldsymbol{D}_{ii} = +1$ (Bad),
$\boldsymbol{A}_i \boldsymbol{X} \geq b - \alpha_i + \beta_i$, $\boldsymbol{D}_{ii} = -1$ (Good);

which can be written as

$D(AX - eb) \leq \alpha - \beta$, where e is a vector of ones

Y. Shi, Optimization Based Data Mining: Theory and Applications, Springer, 2011, and Y. Shi lecture notes

Connections between MCLP and Support Vector Machines (SVM)

An MCLP formulation:

$$\text{Min } w_\alpha \| \alpha \|^p - w_\beta \| \beta \|^p$$

s.t.

$$D(AX - eb) \leq \alpha - \beta$$

Science and Technology Facilities Council
Hartree Centre

Y. Shi, Optimization Based Data Mining: Theory and Applications, Springer, 2011, and Y. Shi lecture notes

Example: Parallel Regularized Multiple-Criteria Linear Programming

$$\min_{z} \frac{1}{2} w^{\top} H w + \frac{1}{2} \xi^{(1)\top} Q \xi^{(1)} + \frac{1}{2} b^2 + C e^{\top} \xi^{(1)} - D e^{\top} \xi^{(2)},$$
$$\text{s.t.} \; (w \cdot x_i) + (\xi_i^{(1)} - \xi_i^{(2)}) = b, \quad \text{for } \{i | y_i = 1\},$$
$$(w \cdot x_i) - (\xi_i^{(1)} - \xi_i^{(2)}) = b, \quad \text{for } \{i | y_i = -1\},$$
$$\xi^{(1)}, \xi^{(2)} \geq 0,$$

Zhiquan Qi, Vassil Alexandrov,, Yong Shi, Yingjie Tian, Parallel Regularized Multiple-Criteria Linear Programming, 2014.

Example: Parallel Regularized Multiple-Criteria Linear Programming

- Convex problem, through the dual:

$$\min_{\alpha,\,\xi^{(1)}} \frac{1}{2}\alpha^{\top}(K(A,A^{\top})+ee^{\top})\alpha+\frac{1}{2}\xi^{(1)\top}Q\xi^{(1)},$$
$$\text{s.t. } -Q\xi^{(1)}-Ce\le E\alpha\le -De,$$

- Reformulate the problem into global optimization one

$$\min_{\pi} f(\pi) = \frac{1}{2}\pi^{\top}\Lambda\pi+\lambda^{\top}\max\{G\pi-Ce,0\}^2 + \mu^{\top}\max\{H\pi+De,0\}^2,$$

- Parallelize efficiently:
  - by dividing the variables
  - by dividing the area to subareas

Zhiquan Qi, Vassil Alexandrov,, Yong Shi, Yingjie Tian, Parallel Regularized Multiple-Criteria Linear Programming, 2014.

Example: Parallel RMCLP Classification Algorithm and Its Application on the Medical Data

TABLE 1
Description of UCI Data Sets

datasets	♯ examples (L)	♯ dimension (N)
Sonar	208	60
Ionosphere	351	34
Australian	690	14
Pima-Indian	768	8
CMC	1,473	9
Votes	435	16
WPBC	110	32

TABLE 2
PRMCLP's Training Time on UCI Data Sets

Dataset	accuracy	1 procs	2 procs	4 procs	6 procs
Sonar	78.21± 4.46	46.35	24.18	13.62	14.12
Ionosphere	87.22± 6.45	148.12	76.54	38.75	25.34
Australian	86.34 ± 4.23	284.76	143.52	72.32	49.77
Pima-Indian	78.12 ± 5.45	331.34	169.31	87.62	56.61
CMC	70.18 ± 3.69	605.17	310.28	160.23	108.12
Votes	95.54 ± 3.48	198.23	101.01	54.29	34.52
WPBC	82.75 ± 2.92	22.25	11.57	5.76	3.43

Zhiquan Qi, Vassil Alexandrov,, Yong Shi, Yingjie Tian, Parallel Regularized Multiple-Criteria Linear Programming, 2014.

Example: Parallel RMCLP Classification Algorithm and Its Application on the Medical Data

TABLE 3
Parallel PRMCLP's Training Time for Breast Cancer Diagnosis Datasets on Different Sample Numbers and Processors

Method	The number of CPU processors				
	accuracy	1 procs(second)	2 procs	4 procs	6 procs
200	95.85± 1.34	8.11	4.52	3.57	1.72
400	95.97± 1.25	25.65	13.56	7.66	4.55
800	96.14± 1.82	65.77	27,89	14.88	11.22
1,600	96.21± 1.47	204.78	101.34	55.12	40.34
3,200	96.31± 1.33	800.34	389.87	230.58	198.01
6,400	96.44± 1.01	4,566.58	2,134.55	1,200.98	920.23
12,800	96.85± 1.24	-	7,300.43	3,400.45	2,455.12
56,900	96.87± 1.17	-	-	9,800.33	7,000.33

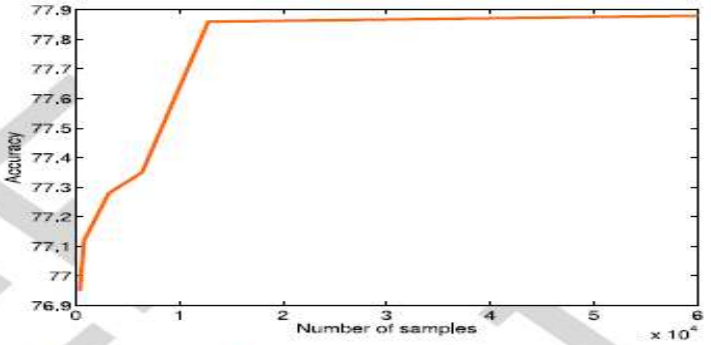


Fig. 1. The accuracies of diabetes datasets under the different number of samples.

Zhiquan Qi, Vassil Alexandrov,, Yong Shi, Yingjie Tian, Parallel Regularized Multiple-Criteria Linear Programming, 2014.

Hartree Centre

Resources

Recommended reading material

Jon Shlens, A Tutrorial on Principal Component Analysis, 2003
https://www.cs.princeton.edu/picasso/mats/PCA-Tutorial-Intuition_jp.pdf

PCA Lecture notes
https://ramanlab.wustl.edu/Lectures/Lecture11_PCA.pdf

G.H. Golub and C.F. van Loan, Matrix Computations, John Hopkins Univ. Press, 3rd Edition, 1996

Science and Technology Facilities Council

Hartree Centre

Questions?

Science and Technology Facilities Council

Hartree Centre

Thank you

vassil.alexandrov@stfc.ac.uk
dominic.richards@stfc.ac.uk

hartree.stfc.ac.uk @HartreeCentre STFC Hartree Centre hartree@stfc.ac.uk