FROM PCA TO PLS Regression

CNAM Paris Décembre 2011

Cours:

La régression aux moindres carrés partiels

Hervé Abdi

www.utdallas.edu/~herve

Program in Cognition and Neurosciences

The University of Texas at Dallas

BIBLIOGRAPHIE: www.utdallas.edu/~herve

- A76 PLS regression
- A77 Principal component analysis
- A81 PLS methods in neuro-imaging

OUTLINE

- A short tutorial on PCA for “6” faces
- Another short tutorial for PLS with the same 6 faces
- A few examples

The Problem for PLS

- A set of observations
- A set of measurements of these observations (dependent variables)
- A set of predictors of these measurements ("kind of" independent variables)

And these measurements are messy !
(non orthogonal, not balanced, not fixed, etc.)

What do we need?

- A way to use the predictors to predict / understand the performance measures.
- So we need a technique to predict a large set of variables $\mathbf{Y}$ from a large set of predictors $\mathbf{X}$.
- This is PLS regression: a technique with PCA and MLR features.

What is PLS anyways? First a Scandinavian specialty

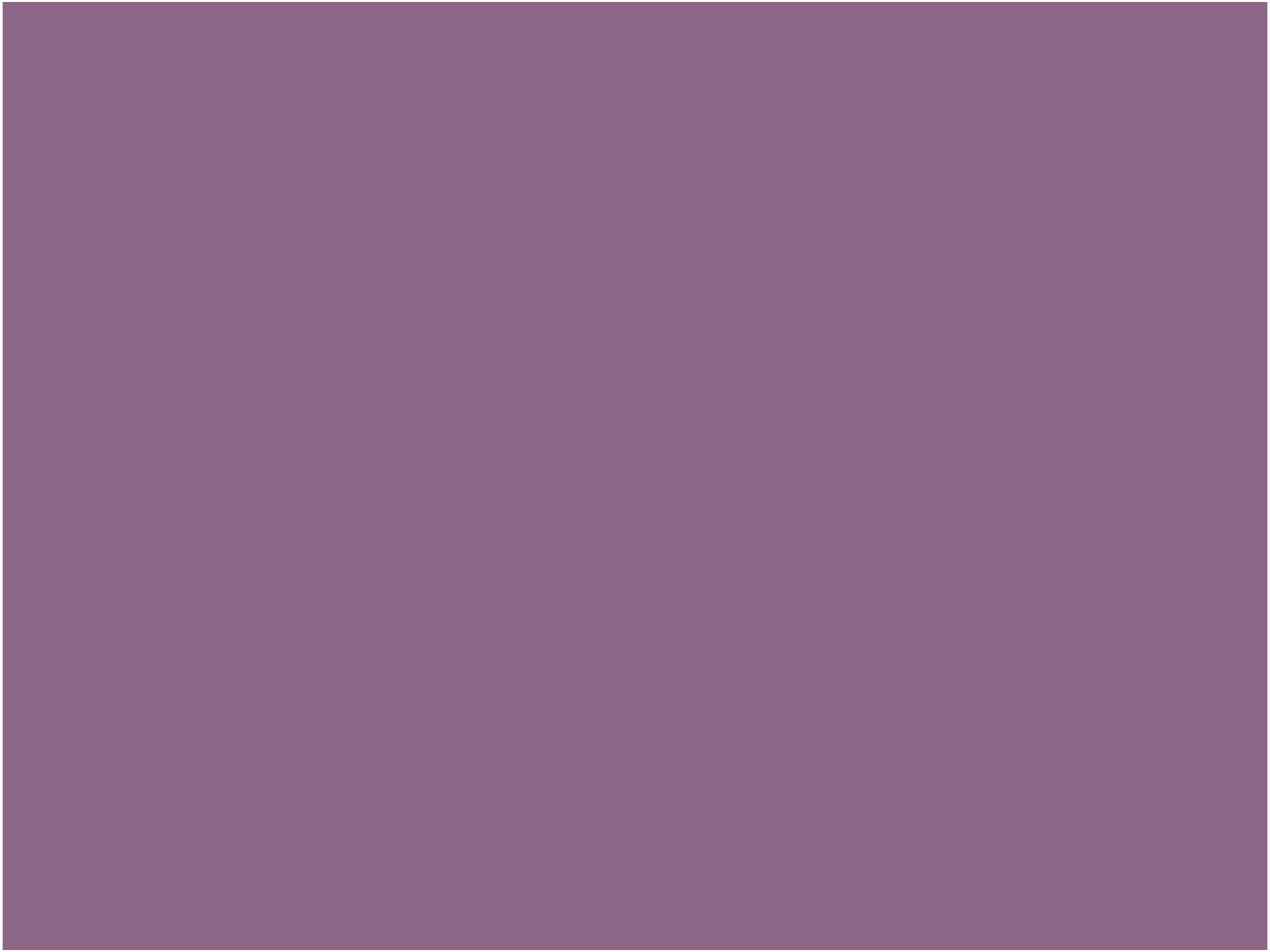
- PLS is for *P*artial *L*east *S*quares
- Invented by Herman Wold (1983), an economist.
- Developed by Svante Wold (his son…) in Chemometrics (first and still most frequent application), and Martens in sensory evaluation

Start with good old PCA!

- Start with a face story for PCA
- Continue with the same face story for PLS
- Finish with a couple of examples

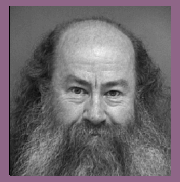
PART 1
A baby PCA example

A six face story



An example:

6 face images

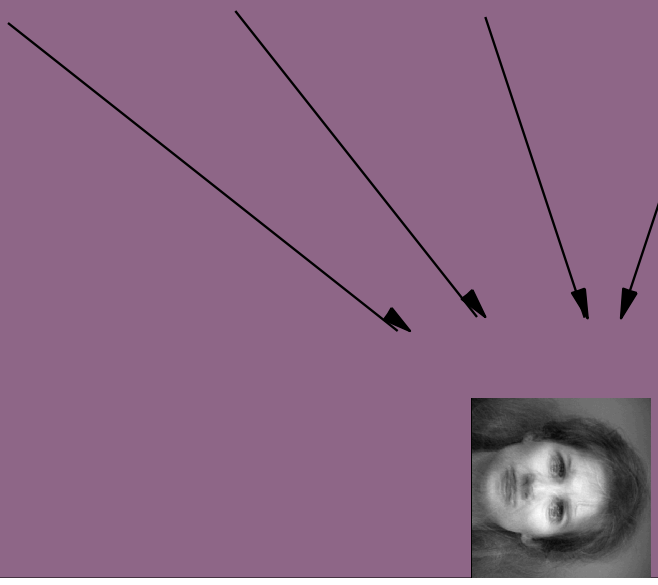


What do these faces have in common?

The trick: Mix them up

This gives the most common face

- The face principal component
- The eigen-face (eigendecomposition)
- The singular face (singular value decomposition)

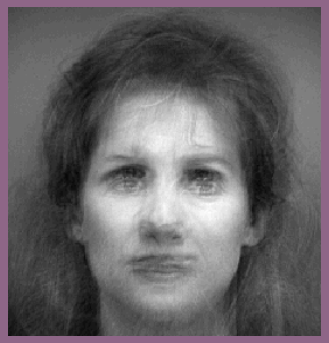


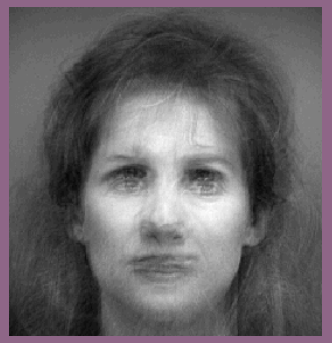
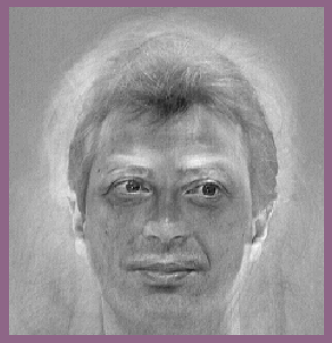
Practically:

It is a weighted average (sum) of the original faces

with *positive* weights only

How Much of The Eigen-Face?

$$\cos\left(\text{  }, \text{  }\right) = .98$$


$- .98 \times$

$=$


DEFLATION

When the eigenface is found.

Get off with it:

Subtract it from all the faces (keep the weights for the subtraction

(go to the orthogonal subspace)

EIGEN-ART

And once again!

But we have already found and eliminated what common to all.

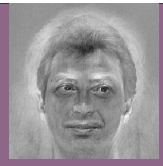
We get now what is common to some faces and discriminate then from other faces

Practically:

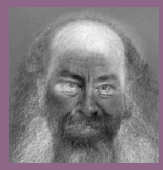
It is a weighted average (sum) of the original faces

with:

positive *and* negative weights.

-.35 x 

-.23 x 

-1.91 x 

.39 x 

1.31 x 

.75 x 

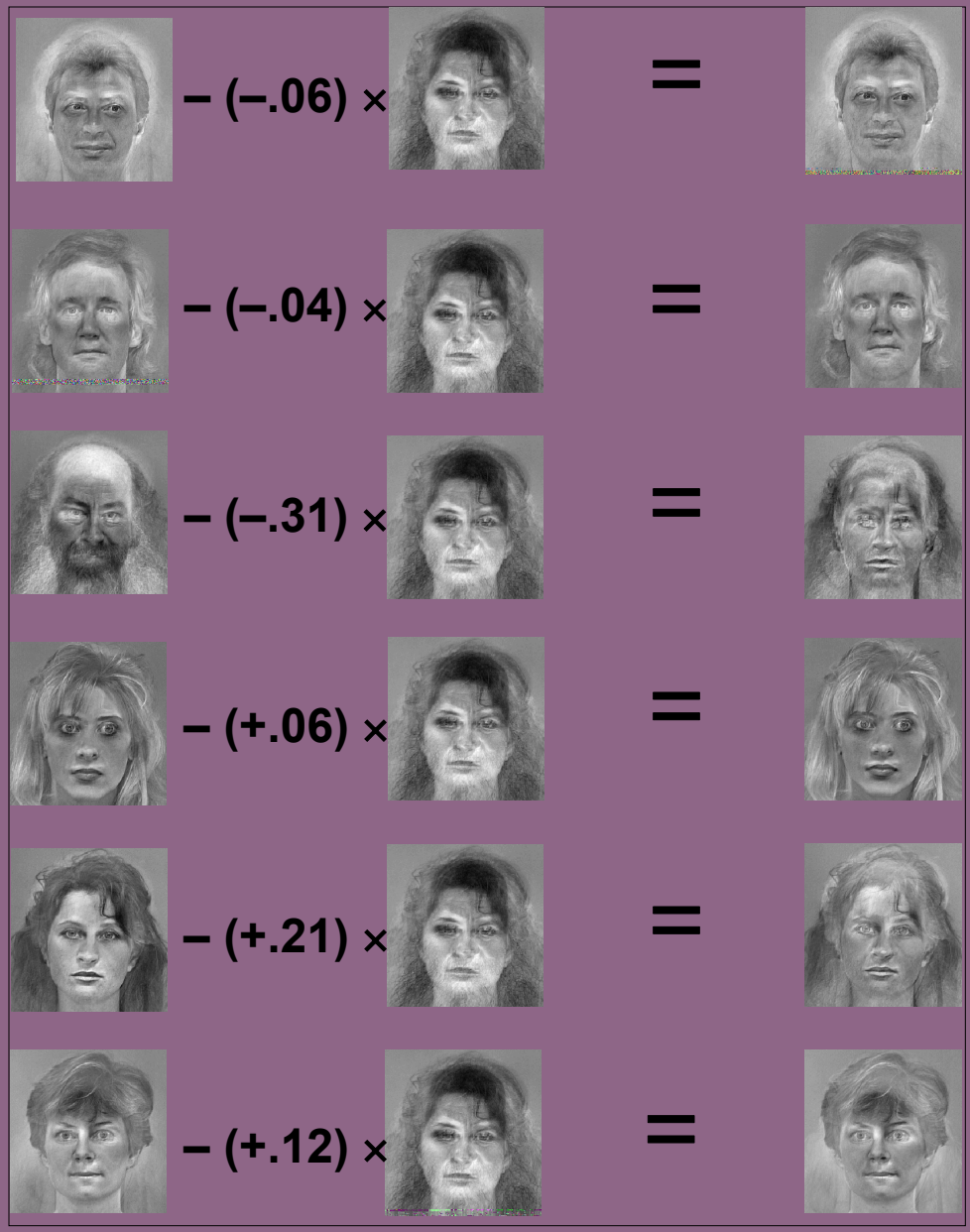
+

=

When we have found the

2nd eigenface

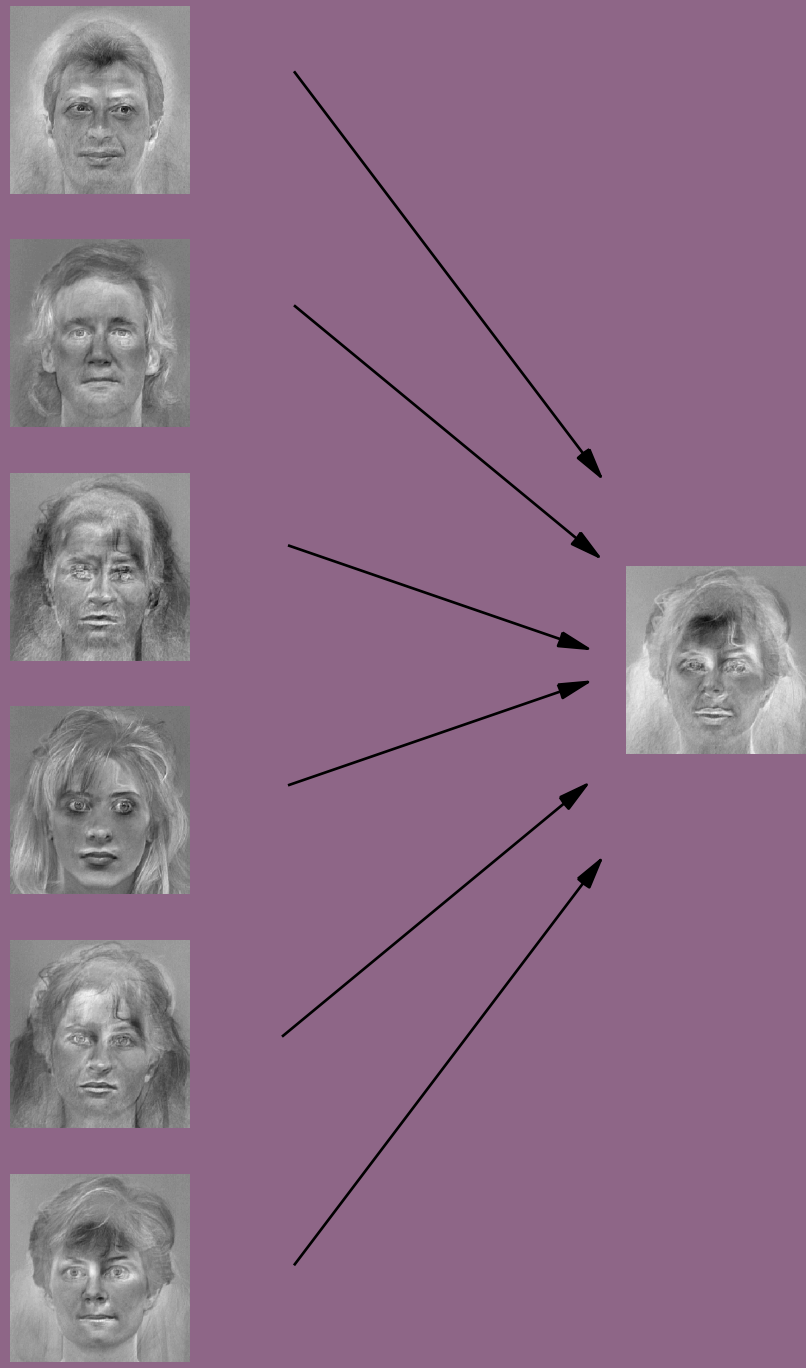
we eliminate it by subtraction.



Once again!

Get the

3rd eigenface



1.37 x 

-.43 x 

-2.29 x 

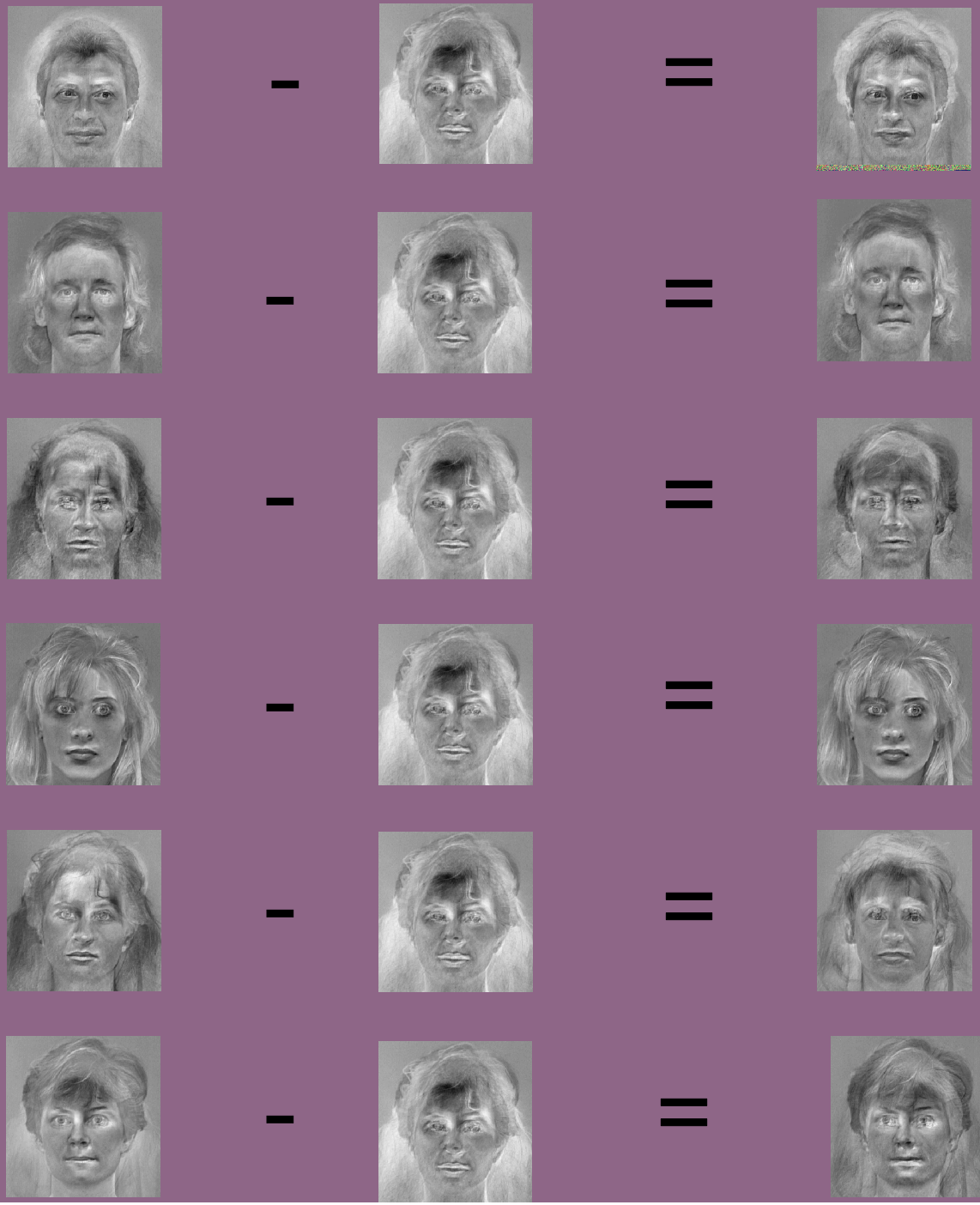
-.97 x 

1.76 x 

.48 x 

+

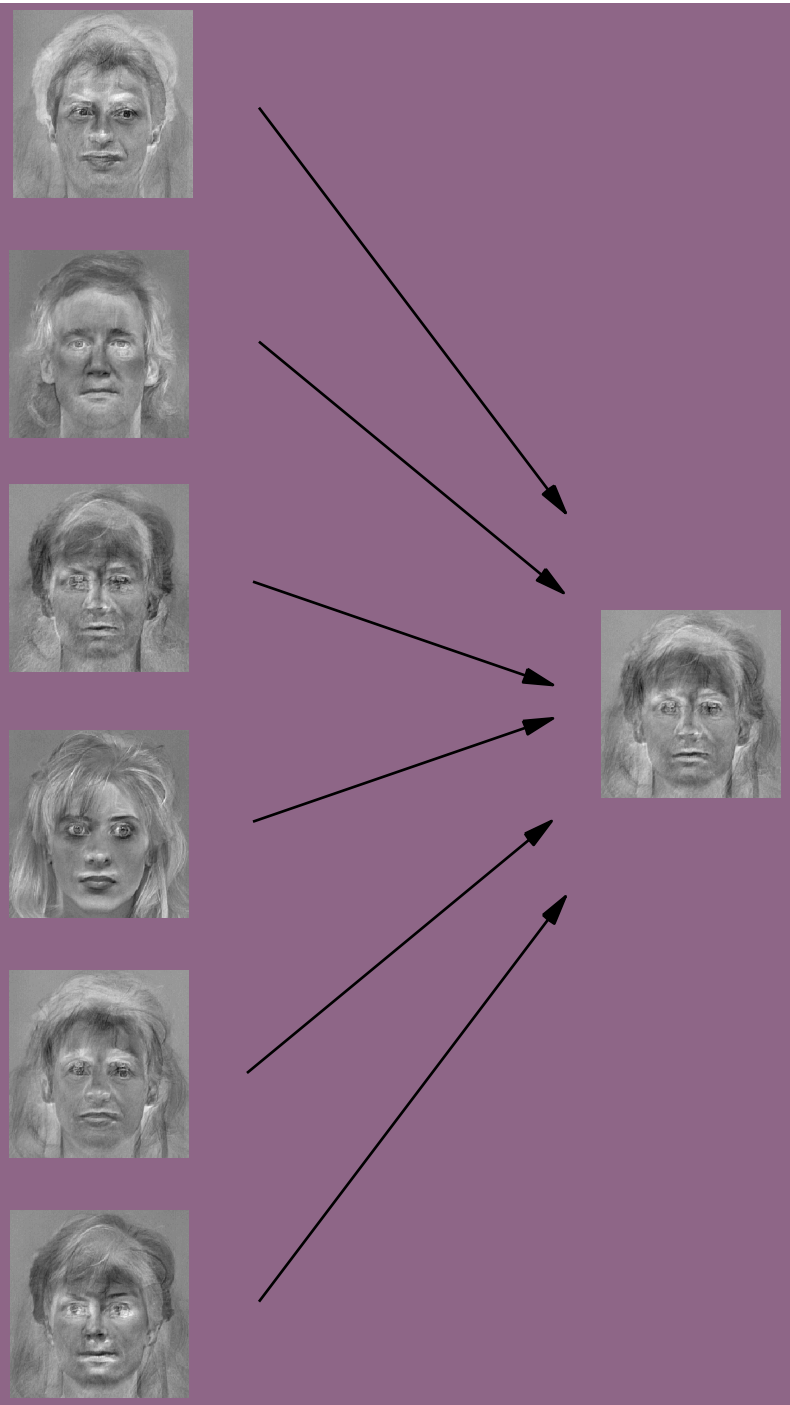
=

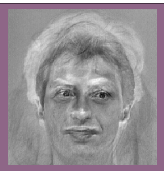


Once again!

Get the

4th eigenface



.35 x 

-2.89 x 

1.43 x 

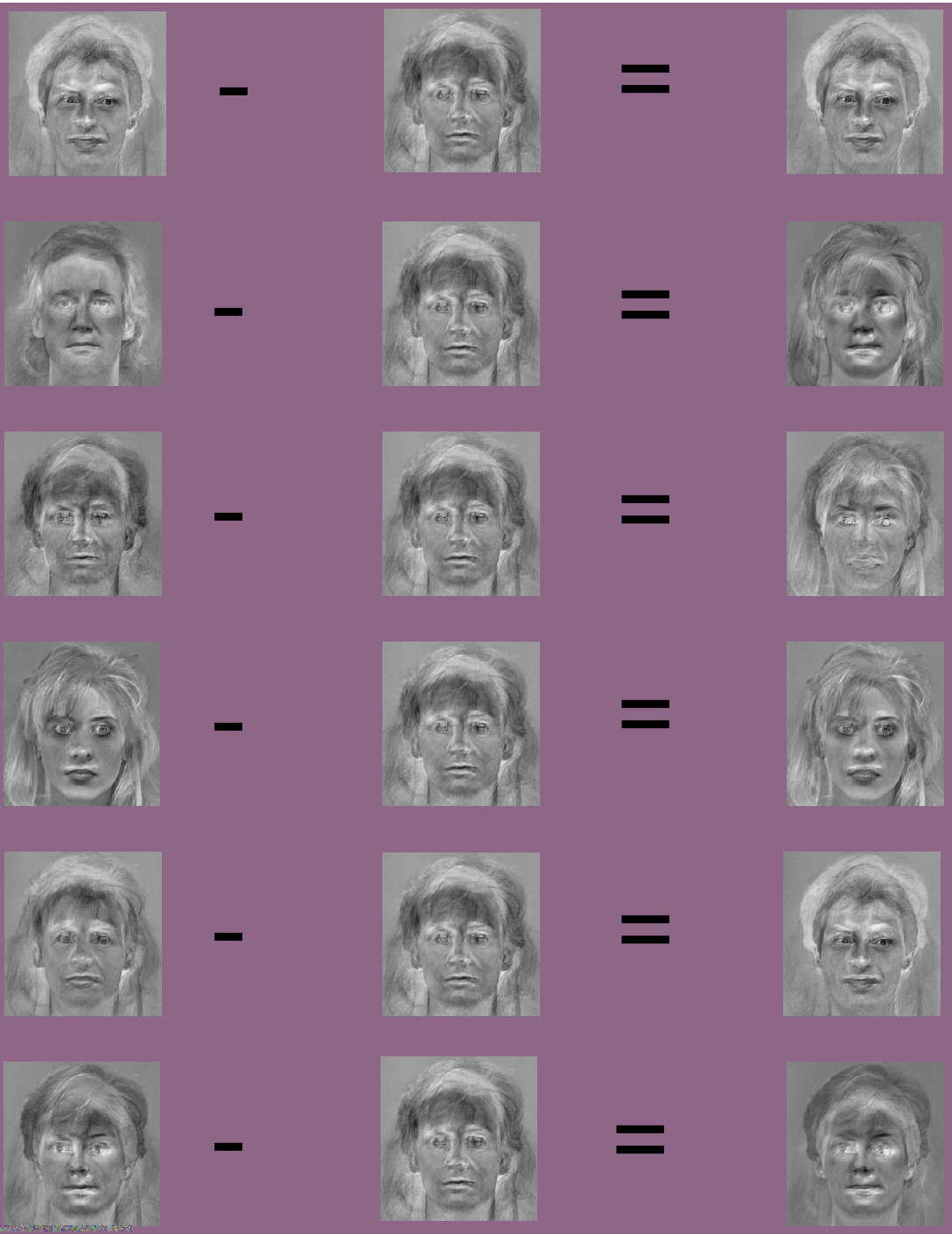
-1.77 x 

1.17 x 

1.80 x 

+

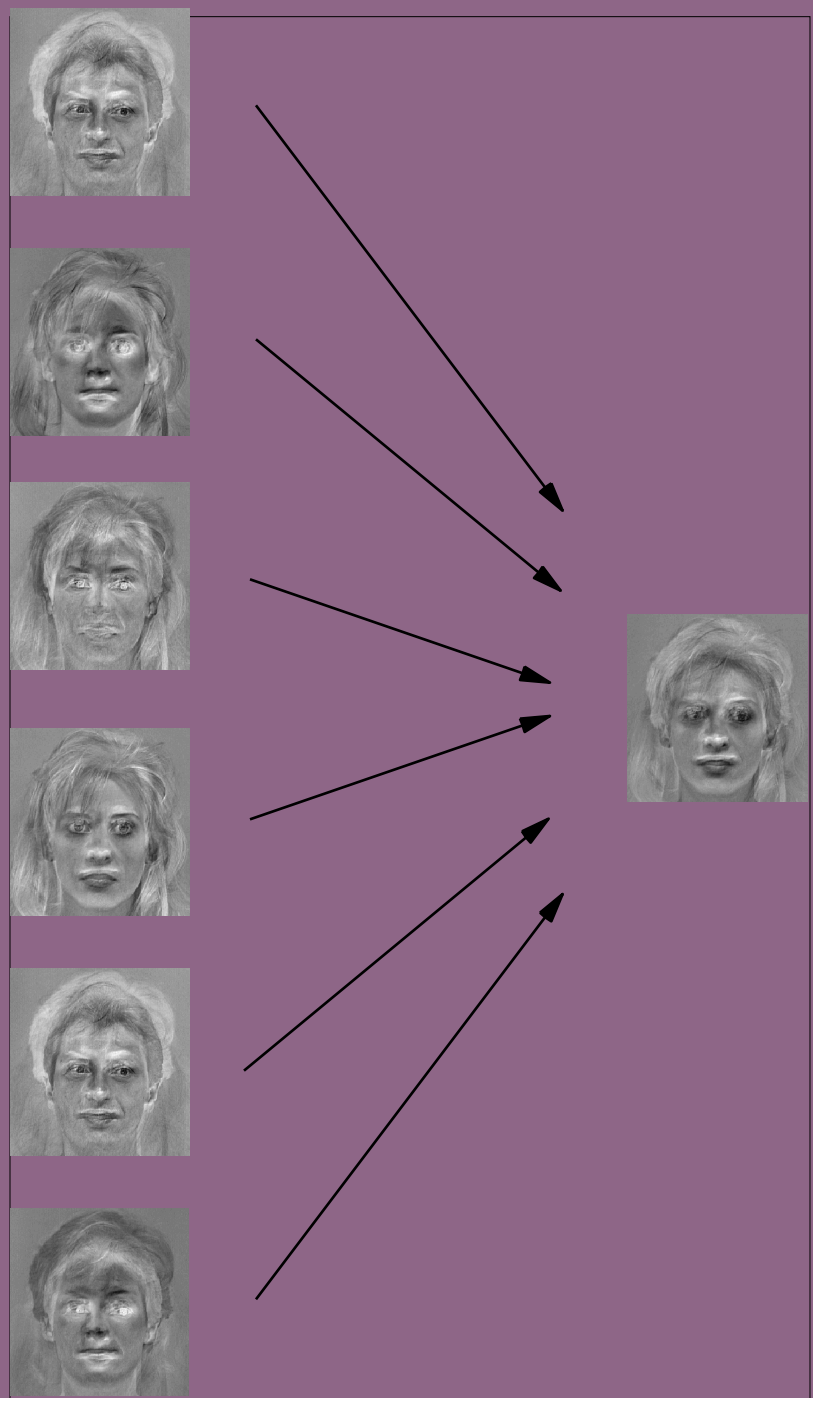
= 

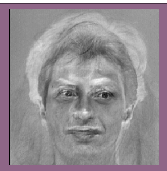


Once again!

Get the

5th eigenface



1.98 x 

-2.34 x 

-.15 x 

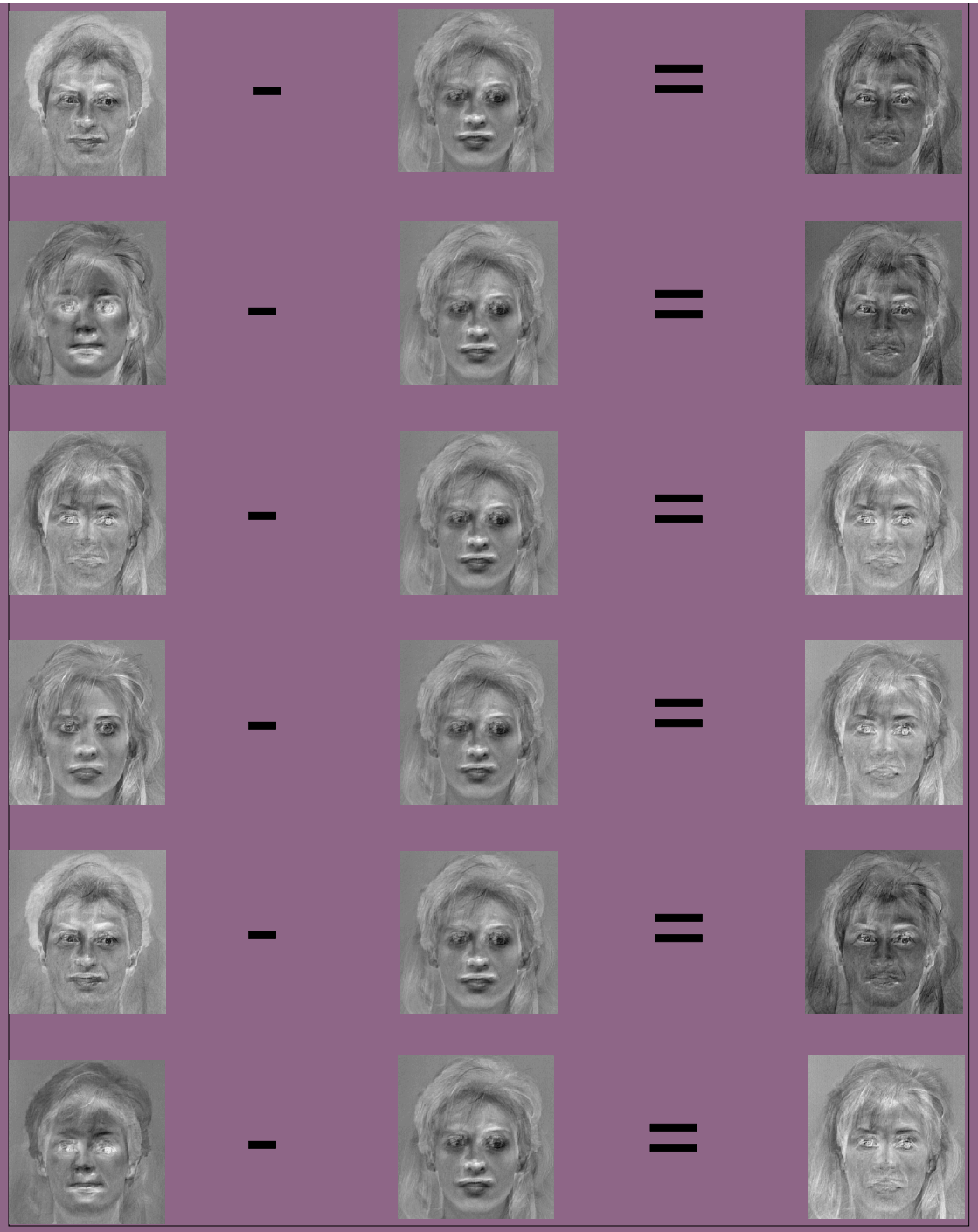
2.19 x 

.48 x 

-2.17 x 

+

= 

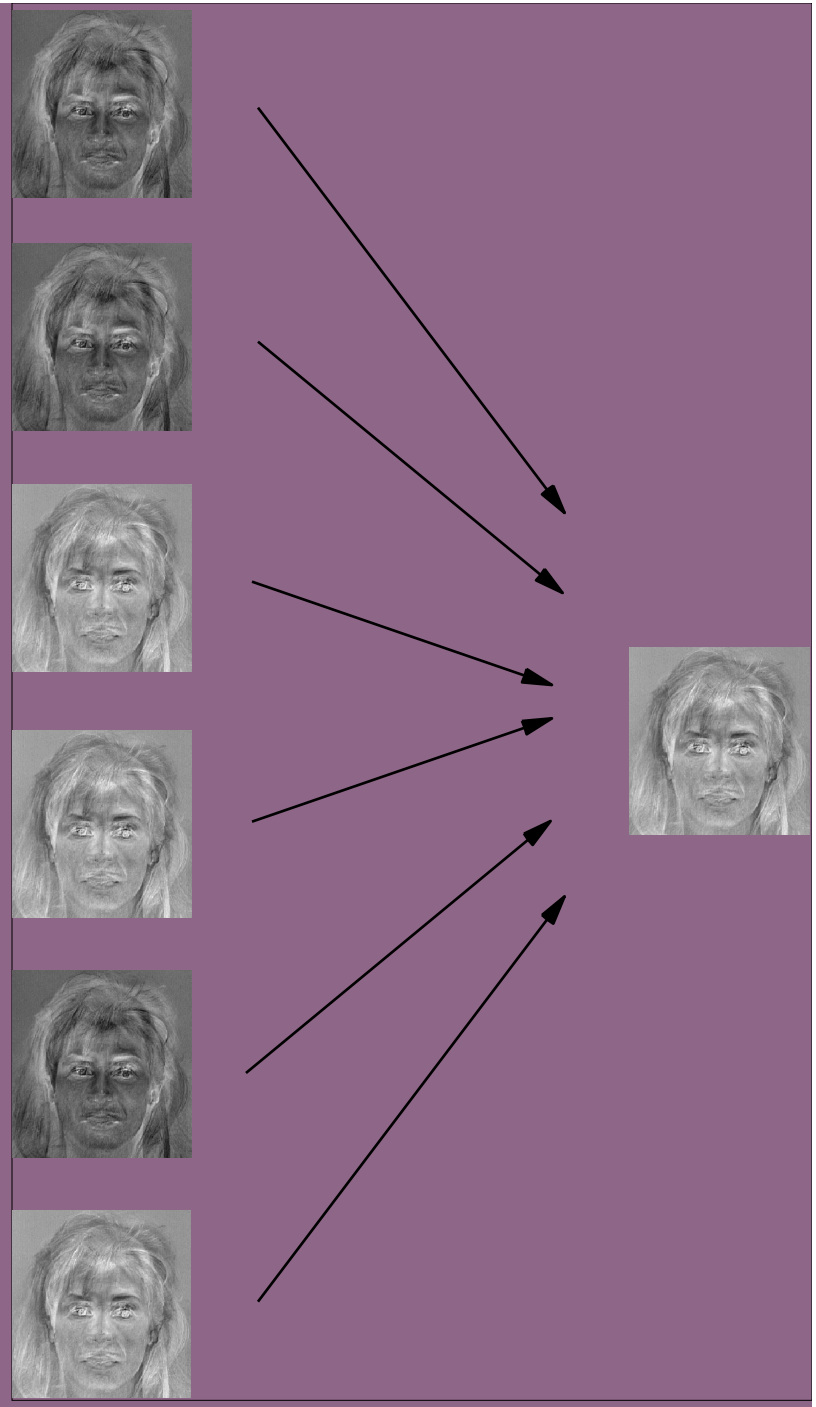


Once again!

Get the

6th eigenface…

Aren't you happy I didn't use a 100-faces example!



-3.77 x 

-1.41 x 

-1.44 x 

3.42 x 

-.89 x 

1.25 x 

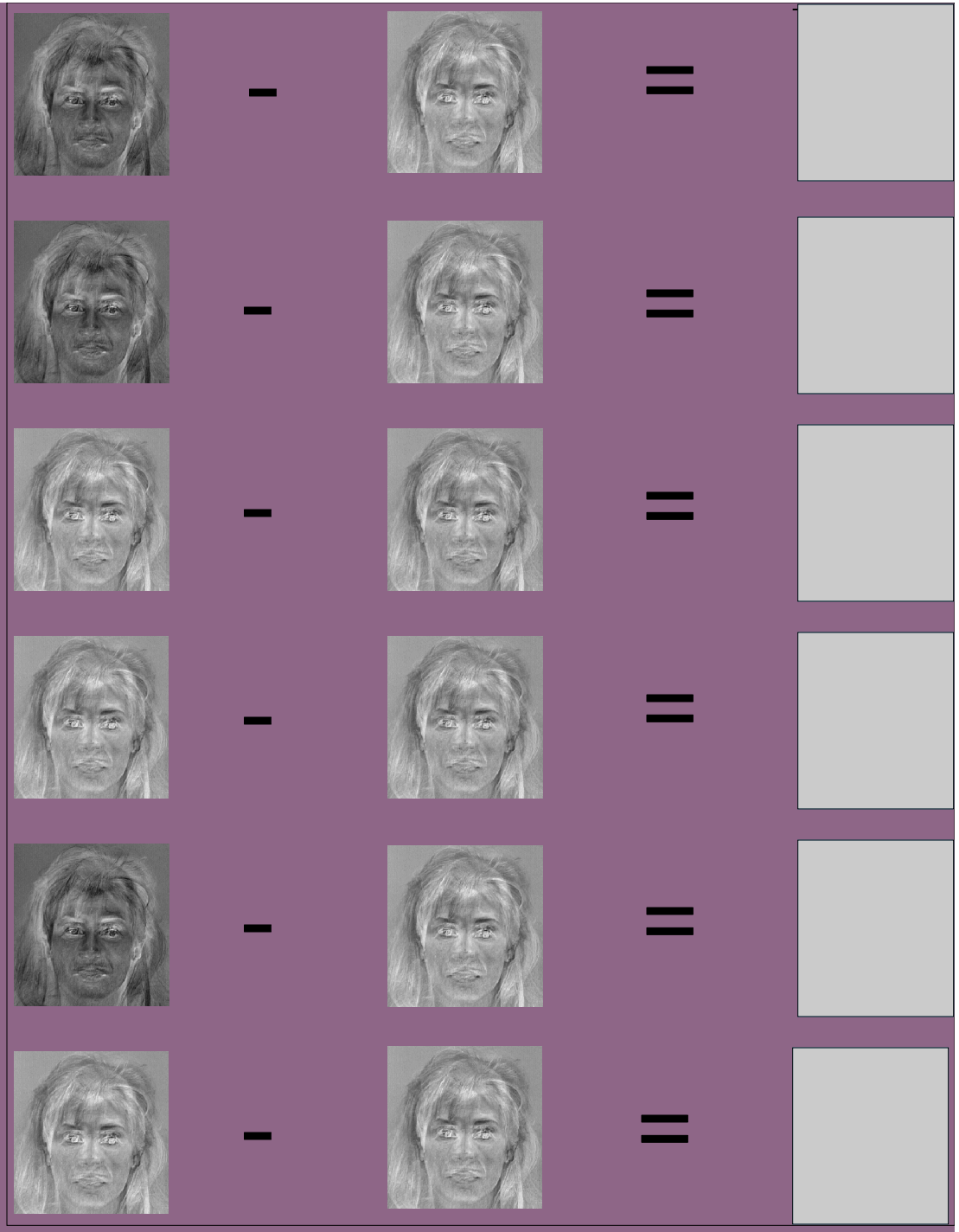
+

=

Once again!

Stopping criterion:

There is nothing left to subtract!

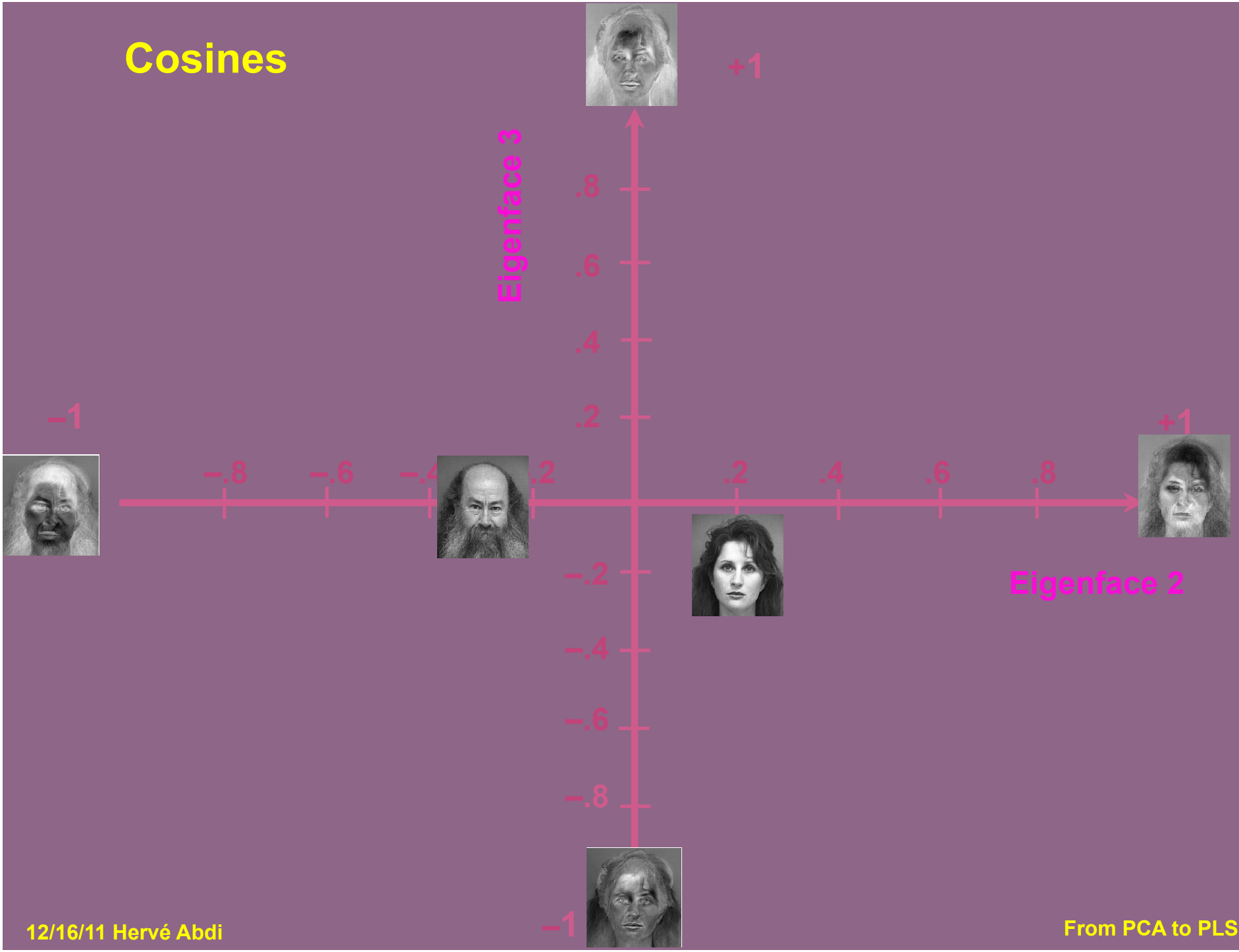


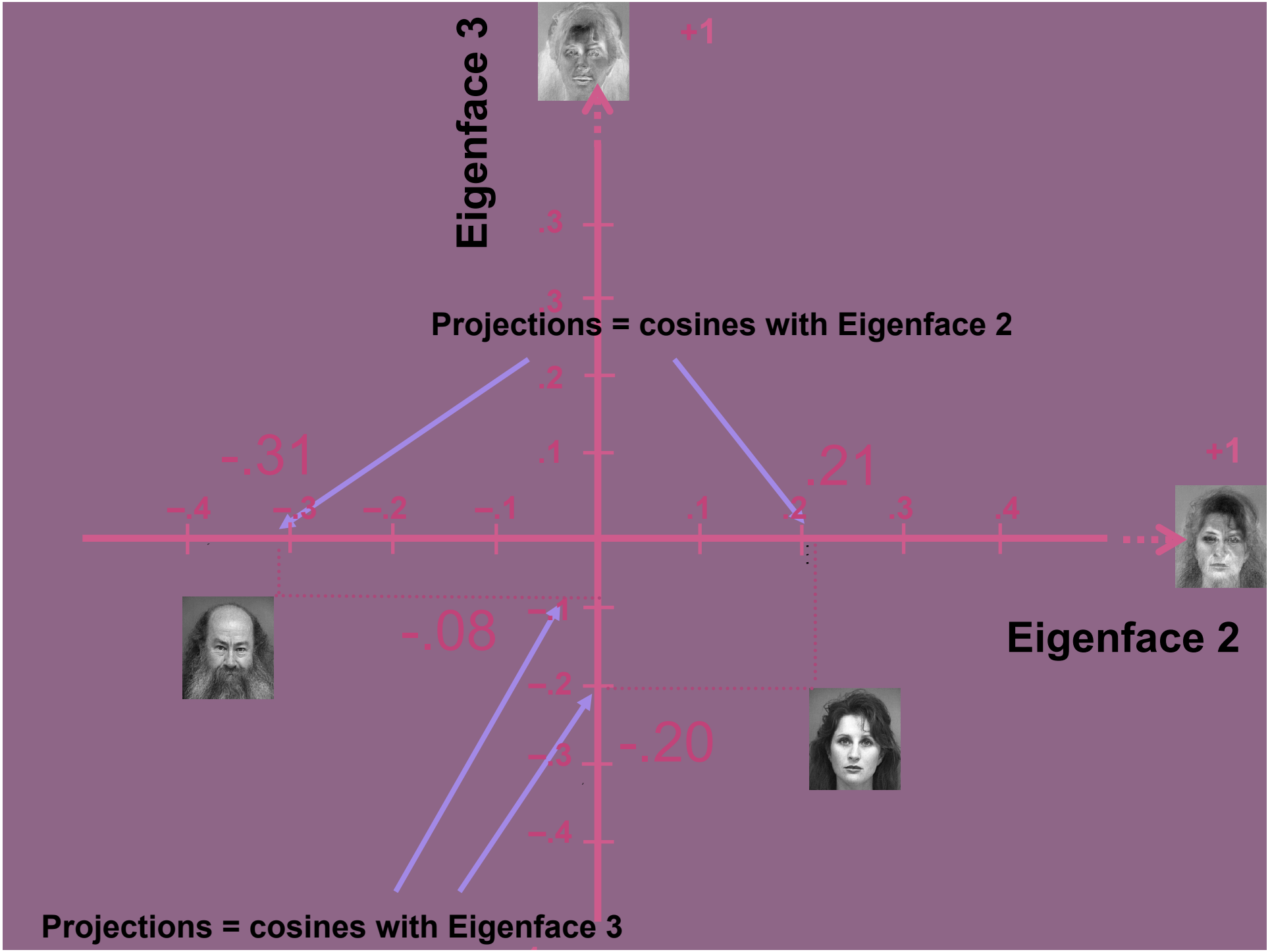
What to do with the components?

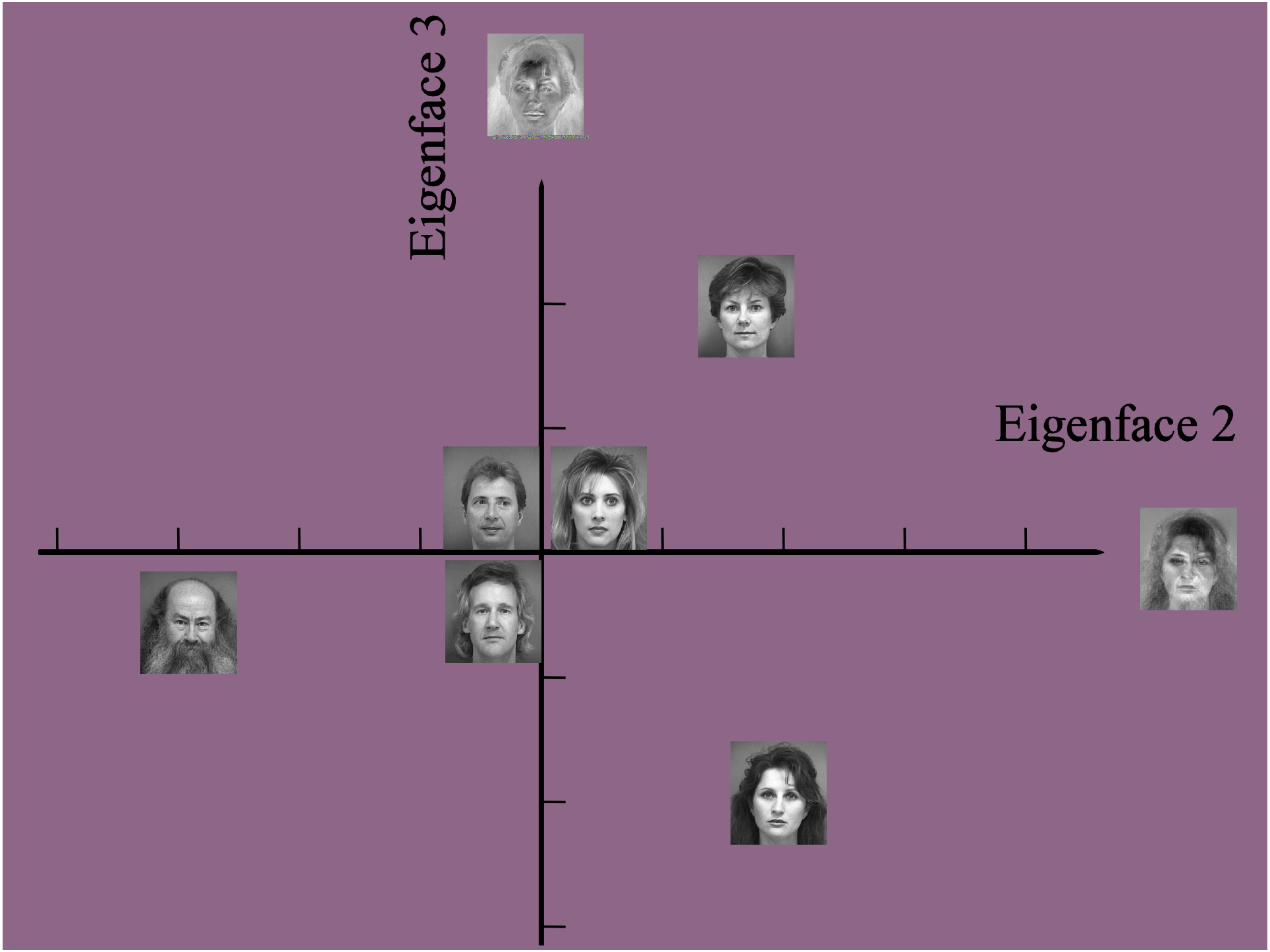
- Look at them (is this art?)
- We can construct (new faces) or reconstruct (old faces) stimul.

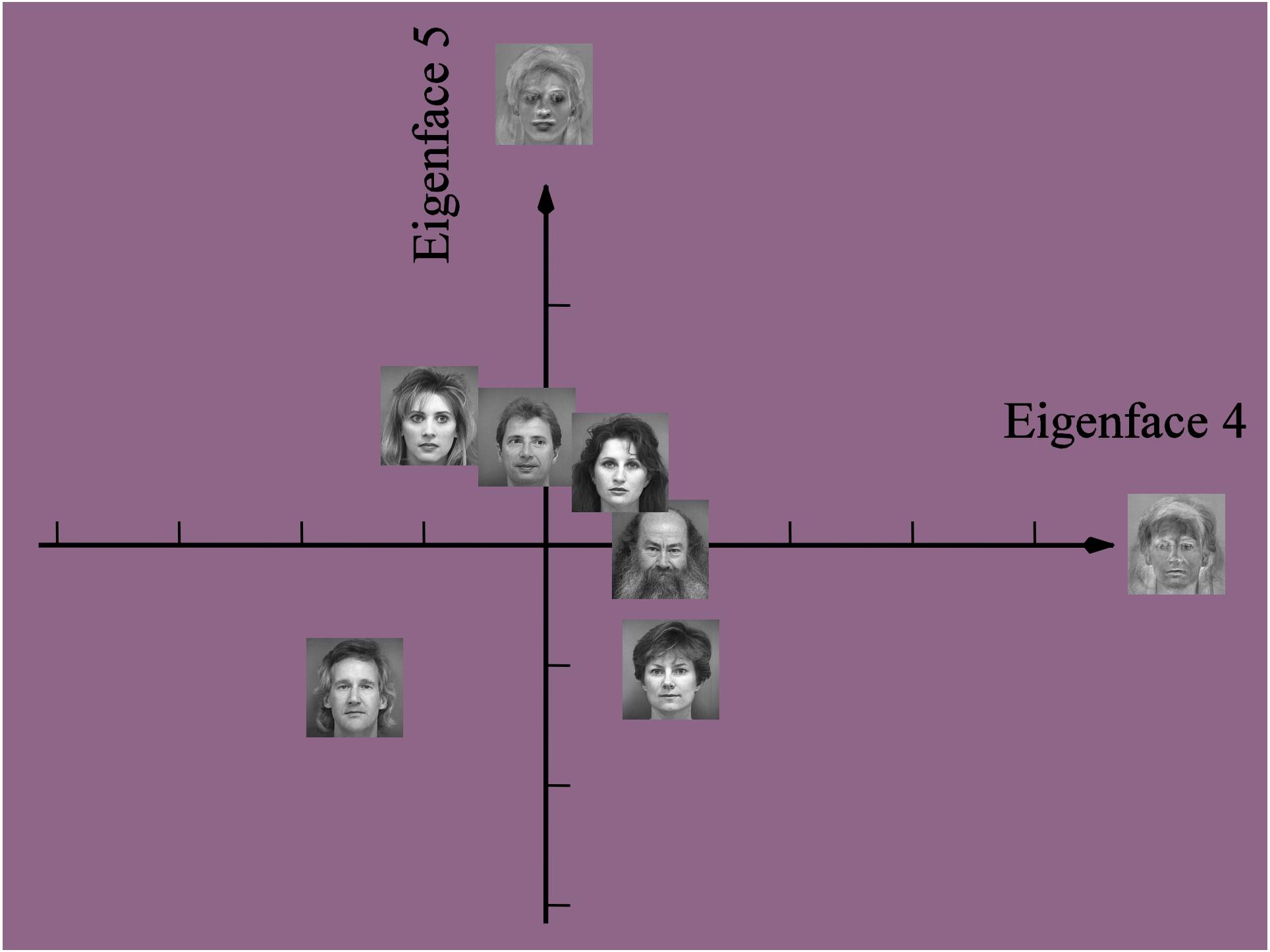
  The cosine gives the contribution (i.e., the amount of) the component for the stimulus.

Cosines

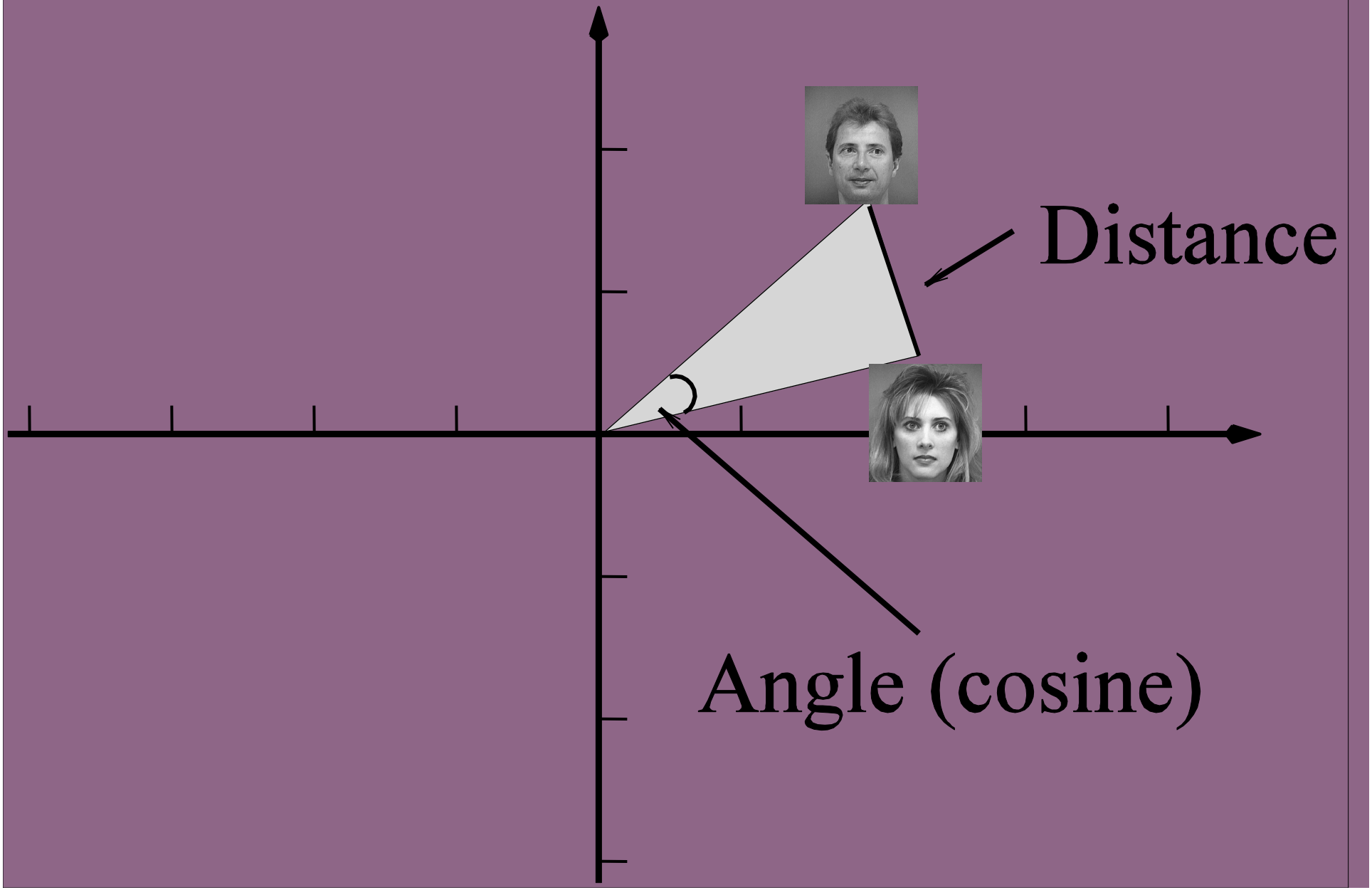



Eigenface 3
+1
.3
.2
.1
-.31
.21
+1
−.4
−.3
−.2
−.1
.1
.2
.3
.4
Projections = cosines with Eigenface 2
-.08
−.1
−.2
−.3
−.4
-.20
Eigenface 2
Projections = cosines with Eigenface 3


Eigenface 3
Eigenface 2


Eigenface 5
Eigenface 4

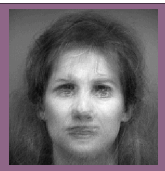
Evaluating Similarity



What to do eigenfaces?

- Build back old (and new) faces

- Eigenfaces are an orthogonal basis

.97 x 

-.06 x 

.12 x 

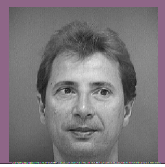
.02 x 

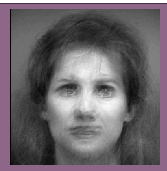
.10 x 

-.12 x 

+

=



.98 x 

-.04 x 

-.04 x 

-.16 x 

-.04 x 

-.12 x 

+

=

.94 x 

-.31 x 

-.08 x 

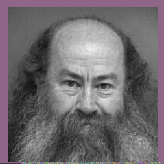
.08 x 

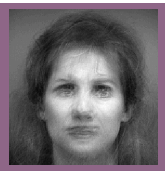
-.01 x 

.04 x 

+

=



.98 x 

.06 x 

.04 x 

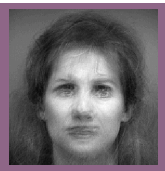
-.10 x 

.11 x 

.11 x 

\+

= 

.97 x 

.12 x 

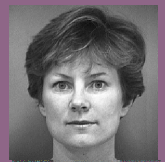
.15 x 

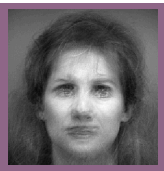
.10 x 

-.11 x 

.04 x 

\+

= 

.95 x 

.21 x 

-.20 x 

.06 x 

.03 x 

-.03 x 

\+

= 

Distinctive faces are recognizable with less eigenfaces!

A formula: What is PCA all about

$$\mathbf{X} = \mathbf{S}\,\mathbf{D}_{\Delta}\,\mathbf{V}^{\mathsf{T}} = \sum_{l}^{L} \delta_l\,\mathbf{s}_l\,\mathbf{v}_l^{\mathsf{T}}$$

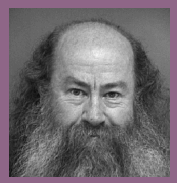
With

$$\mathbf{S}^{\mathsf{T}}\mathbf{S} = \mathbf{V}^{\mathsf{T}}\mathbf{V} = \mathbf{I}$$

PLS Regression

What's the problem with PCA?

- PCA explains $\mathbf{X}$.
- PCA and regression: the eigenvectors explaining $\mathbf{X}$ may not explain $\mathbf{Y}$.
- PLS searches for vectors that explain both $\mathbf{X}$ and $\mathbf{Y}$.

y	X
+1	
+1	
+1	
–1	
–1	
–1	

An example:

6 face images:
3 Boys and 3 Girls
Boys are +1
Girls are -1
(contrast coding)

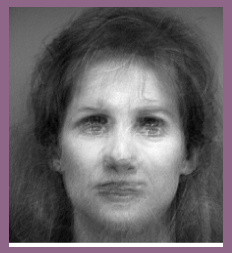
What in these faces separate the boys from the girls?

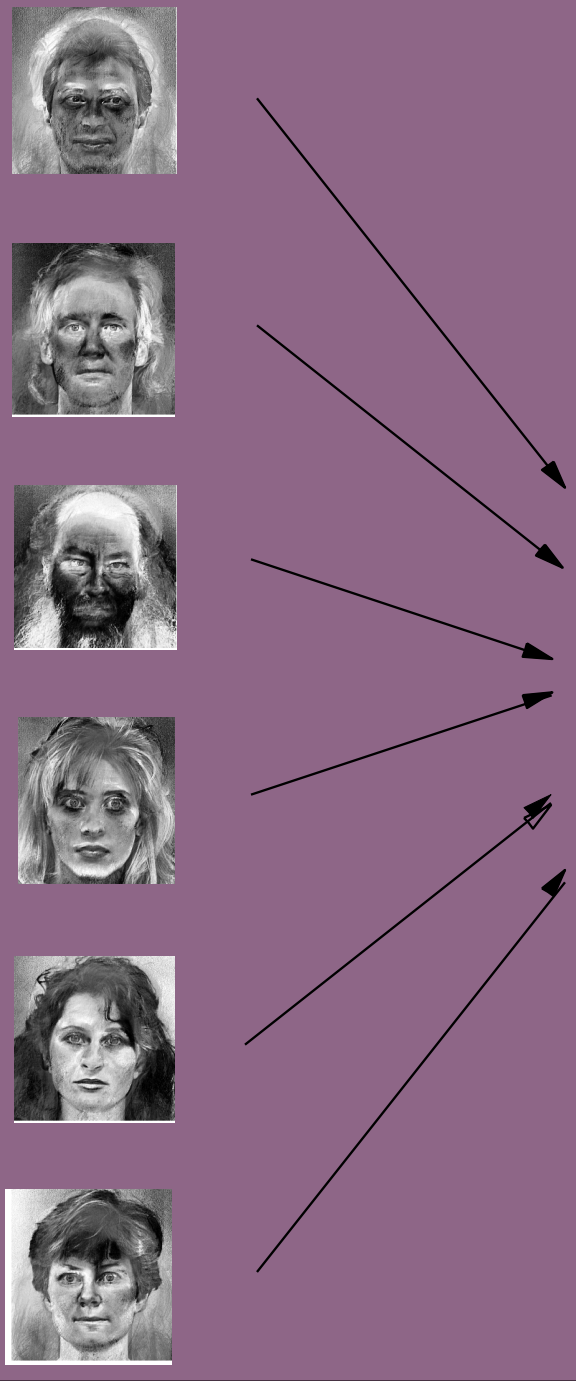
To find up: Mix the faces up, with the constraint that this mixture gives the best prediction of y.

This gives the first *latent* face:

from X contributing to the prediction of y.

First eliminate what is common: the “mean face” (we are looking for differences!)





Practically:

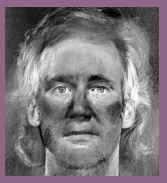
It is a weighted average (sum) of the original faces

with:

positive *and* negative weights.

And it gives the best prediction of **y**.

+.24 

+.28 

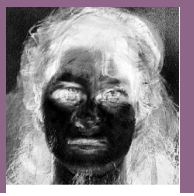
+.60 

–.16 

–.55 

–.42 

=



LATENT PIXELS

Call these weights the *latent-pixels.*

Store them in the vector **t**.

In this case **t** stores the *latent-sex-pixels*

$$\mathbf{t} = \begin{matrix} +.24 \\ +.28 \\ +.60 \\ -.16 \\ -.55 \\ -.42 \end{matrix} =$$

Predict $\mathbf{y}$ from $\mathbf{t}_1$: b_1

- **Predict y from t as y = $b \times \mathbf{t}_1$.**
- **Here b_1 = 2.05 $\rightarrow$ y = 2.05 × t_1**

Now try to predict $\mathbf{y}$

- From the similarity of each face to the latent-face (i.e., projection of X on the latent-face)

$$(\mathbf{y} - \hat{\mathbf{y}})$$

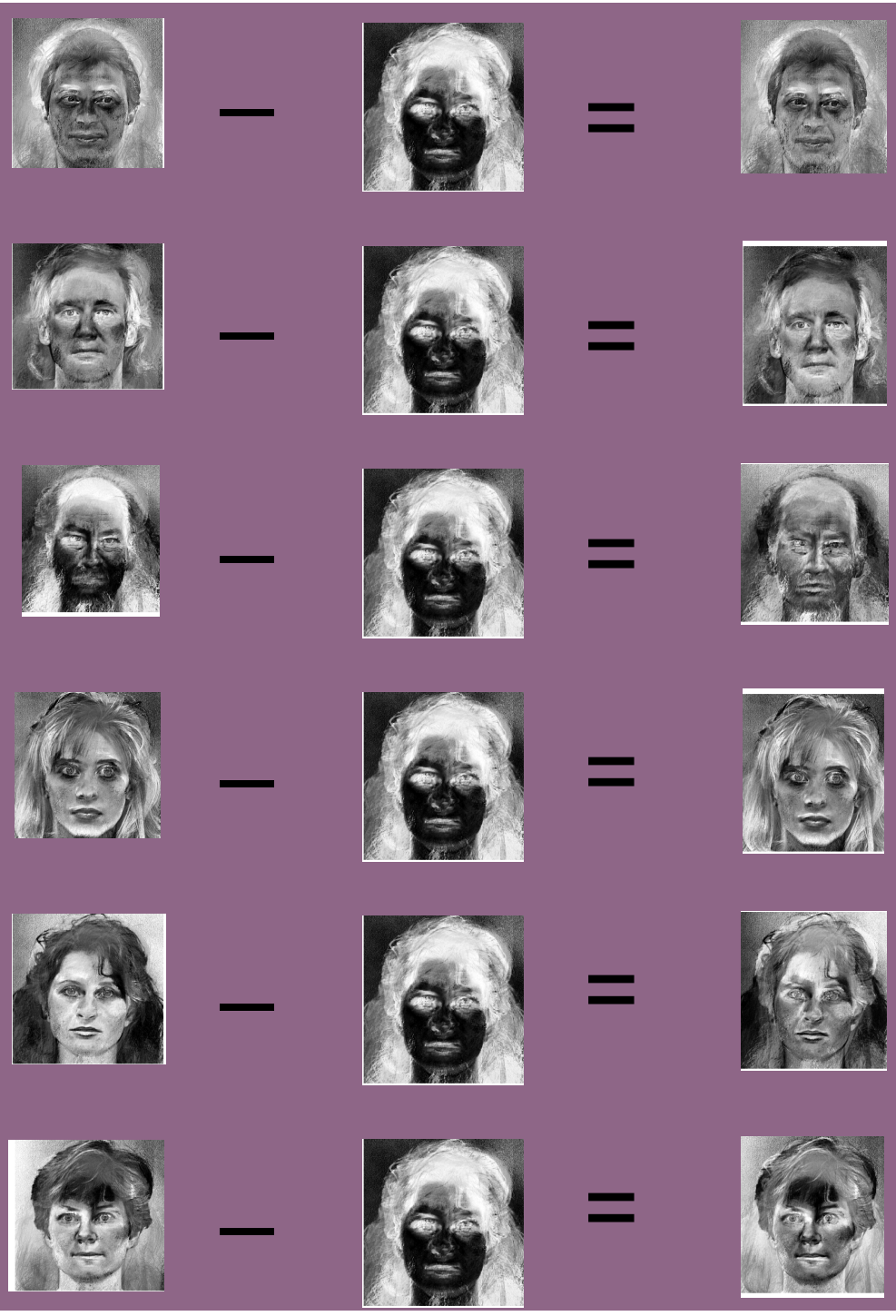
$$\hat{\mathbf{y}} = [.50,\ .69,\ 1.24,\ -.33,\ -1.13,\ -.86]$$

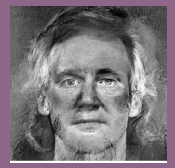
$$\mathbf{y} = [\ 1\ ,\ 1\ ,\ 1\ ,\ -1\ ,\ -1\ ,\ -1\ ]$$

$$(\mathbf{y} - \hat{\mathbf{y}}) = [.50,\ .31,\ -.24,\ -.73,\ .13,\ -.15]$$

  - Now Replace $\mathbf{y}$ by $\hat{\mathbf{y}}$ (this is the partial from PLS) and …

Now eliminate (partial out) the latent-face from each face and try to predict $\mathbf{y}$ (or in fact what is left of it) from $\mathbf{X}$ (or in fact what is left of it)



y	X
+.50	
+.31	
–.24	
–.73	
+.13	
–.15	

now
try to predict
y
(or in fact what
is left of it)
from
X
(or in fact what
is left of it)

PLS ????

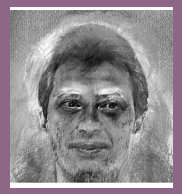
- We have subtracted the latent-face (latent vector) from $\mathbf{y}$ and $\mathbf{X}$.
- Or we have partialed out the effect of the latent face before re-iterating the regression.
- Hence the name: *partial least-square* regression …

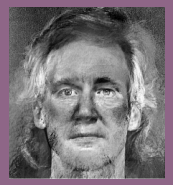
What in these *residual* faces separate the *residual* "gender"?

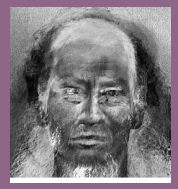
To find up: Mix-up the residual faces, with the constraint that this mixture gives the best prediction of y.

This gives the second *latent* face:

from X contributing to the prediction of y.







12/16/11 Hervé Abdi

Practically:

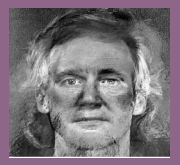
It is a weighted average (sum) of the residual faces

with:

positive *and* negative weights.

And it gives the best prediction of (what is left of) **y**.

+.35 

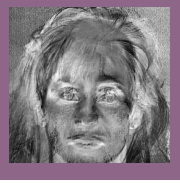
+.44 

–.31 

–.67 

+.34 

–.14 

= 

Call these weights the *latent-pixels.*

Store them in the vector $\mathbf{t}_2$

$$\mathbf{t}_2 = \begin{bmatrix} +.35 \\ +.44 \\ -.31 \\ -.67 \\ +.34 \\ -.14 \end{bmatrix}$$

Now try to predict $\mathbf{y}$ (use b_2 = .87)

- From the similarity of each face to the first 2 latent-faces (i.e., projection of $\mathbf{X}$ on the latent-faces).

$$\hat{\mathbf{y}} = [\ .80,\ \ .96,\ \underset{(\mathbf{y}-\hat{\mathbf{y}})}{.96},\ -.91,\ -.83,\ -.98]$$

$$\mathbf{y} = [\ .50\ ,\ \ .31,\ -.24,\ -.73,\ \ .13,\ \ -.15\ ]$$

$$(\mathbf{y}-\hat{\mathbf{y}}) = [\ .30,\ \ .65,\ \ 1.20,\ \ .18,\ -.70,\ \ -.83]$$

- Now Replace $\mathbf{y}$ by $\hat{\mathbf{y}}$ and …

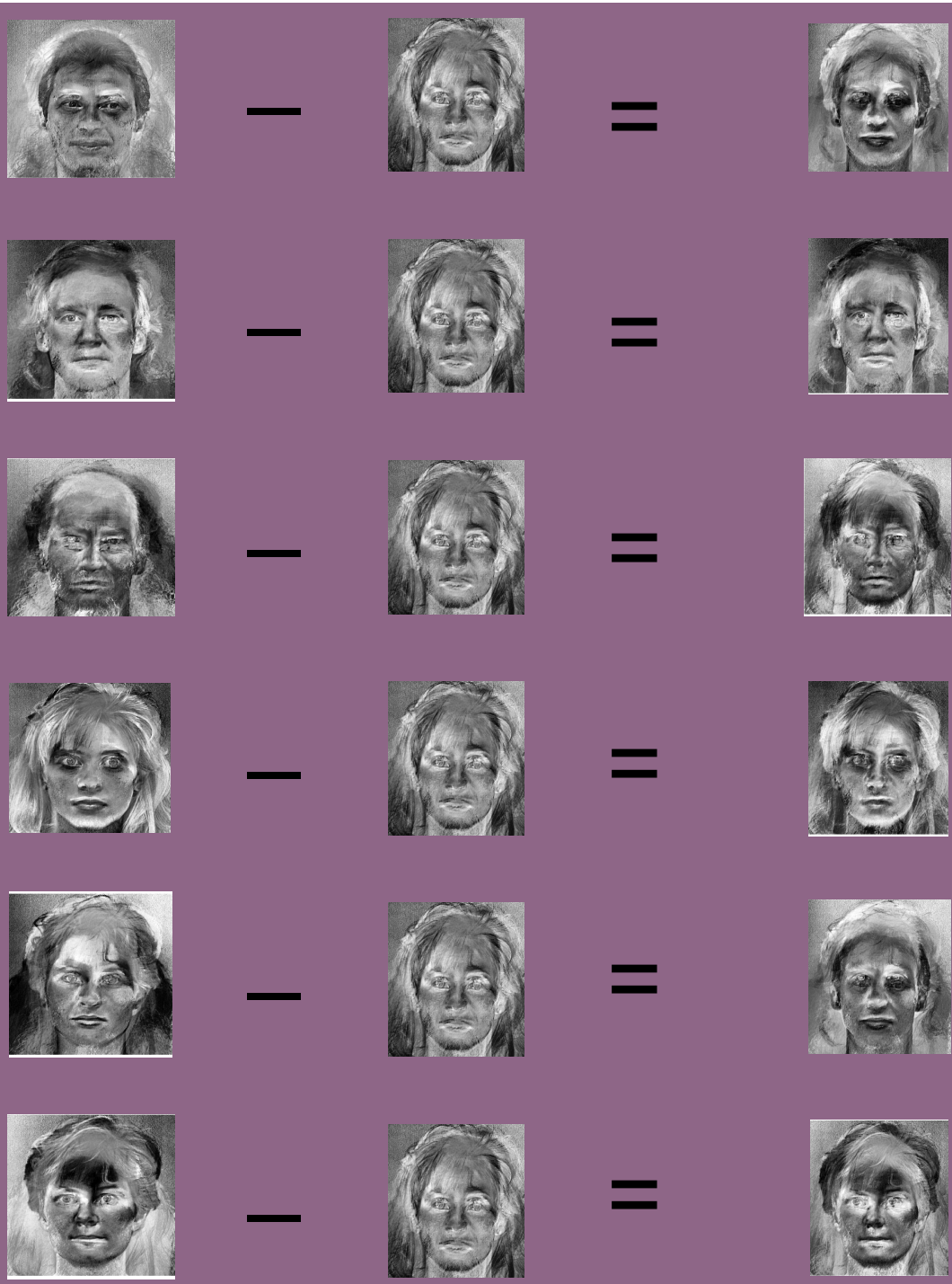
Now eliminate the latent-face from each face and try to predict

y

(or in fact what is left of it) from

x

(or in fact what is left of it)



y	X
.30	
.65	
1.20	
.18	
–.70	
–.83	

now
try to predict
y
(or in fact what
is left of it)
from
X
(or in fact what
is left of it)

… And so on until nothing's left to subtract …

...

Now have a look at the latent faces and latent pixels

...

$Y = b \times T$

2.05	.87	.17	.04	.01	.00

X

	t_1	t_2	t_3	t_4	t_5	t_6
+1	.24	.35	.56	.49	.32	-.46
+1	.28	.44	-.11	-.73	.11	-.44
+1	.60	-.31	-.35	.26	-.43	-.30
-1	-.16	-.67	-.01	-.15	.58	-.27
-1	-.55	.34	-.56	.31	.03	-.52
-1	-.42	-.14	.48	-.19	-.61	-.32

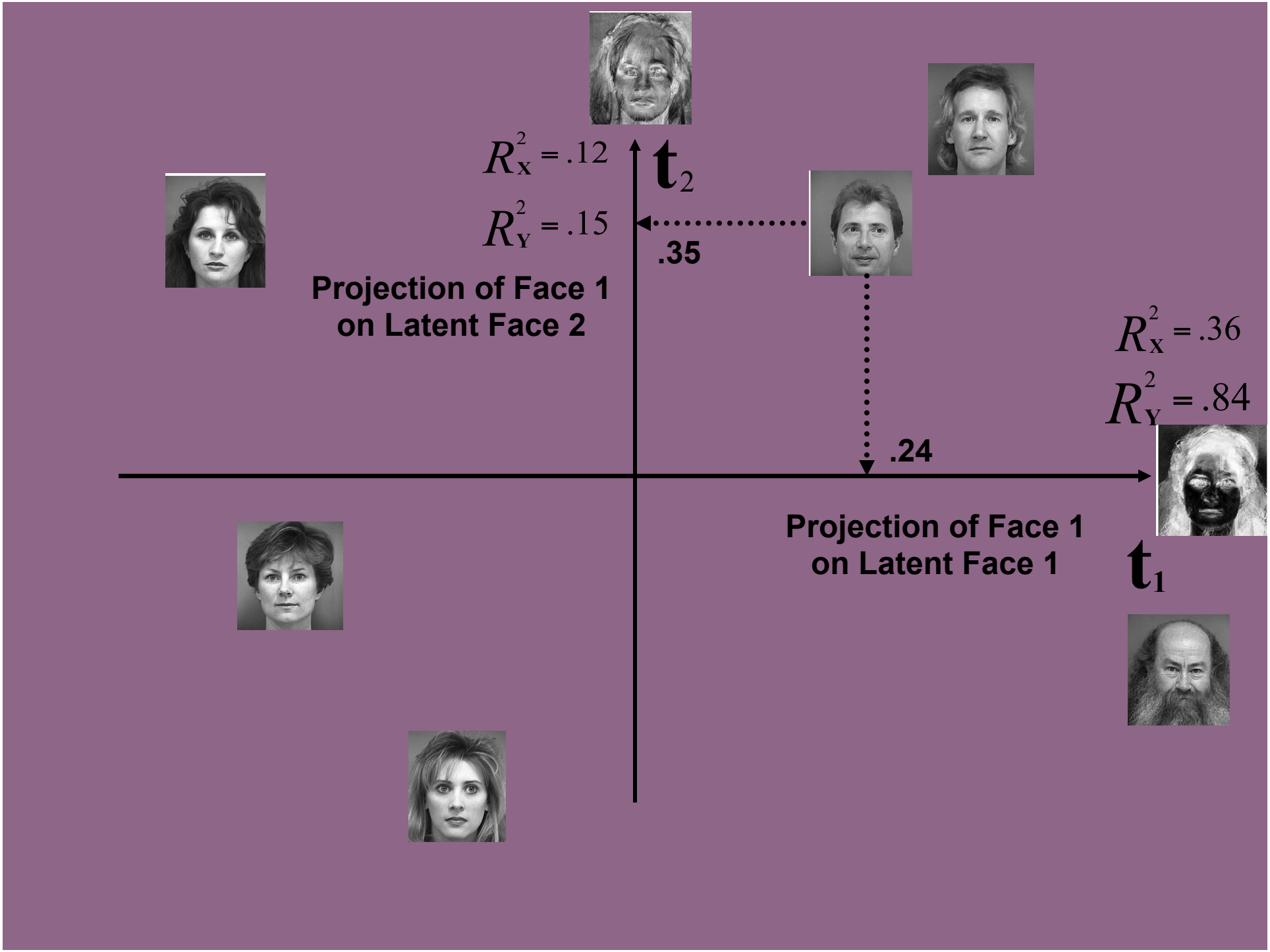
2.05	.87	.17	.04	.01	.00
					

y % Variance	84	15	1	0	0	0
Sum y % Variance	84	99	100	100	100	100
X % Variance	36	12	21	16	15	0
Sum X % Variance	36	48	69	85	100	100
Sum X eigenvalues	94	96	97	98	99	100
	t_1	t_2	t_3	t_4	t_5	t_6

How good is it ?

What to do with the latent vectors?

- Look at them (is this art?)
- We can construct (new faces) or reconstruct (old faces) stimuli.
- Plot them.


$R^2_X = .12$
$R^2_Y = .15$
t_2
.35
Projection of Face 1
on Latent Face 2
$R^2_X = .36$
$R^2_Y = .84$
.24
Projection of Face 1
on Latent Face 1
t_1

A couple of equations

$$\mathbf{X} = \mathbf{T}\,\mathbf{P}^{T} \quad \text{with} \quad \mathbf{T}^{T}\mathbf{T} = \mathbf{I}$$

and

$$\hat{\mathbf{Y}} = \mathbf{T}\,\mathbf{D}_{B}\mathbf{C}^{T}$$

As a regression problem

$$\hat{\mathbf{Y}} = \mathbf{T}\,\mathbf{D}_{B}\mathbf{C}^{T} = \mathbf{X}\,\mathbf{B}_{PLS}$$

with

$$\mathbf{B}_{PLS} = \mathbf{X}^{+}\mathbf{Y} = \mathbf{P}^{T+}\mathbf{D}_{B}\mathbf{C}^{T}$$

How to predict $\mathbf{Y}$?

- From the latent vectors. Use:

$$\hat{\mathbf{Y}} = \mathbf{T}\,\mathbf{D}_{\mathbf{B}}\,\mathbf{C}^{\mathsf{T}}$$

- From X, with Z-scores (for X and Y). Use:

$$\hat{\mathbf{Z}}_{\mathbf{Y}} = \mathbf{Z}_{\mathbf{X}}\,\mathbf{B}_{\mathbf{PLS}}$$

- From X. Use:

$$\hat{\mathbf{Y}} = \mathbf{X}\,\mathbf{B}^{*}_{\mathbf{PLS}}$$

How good is the prediction?

The problem: The prediction is sample-dependent

The question:

How would we do on a *new* sample?

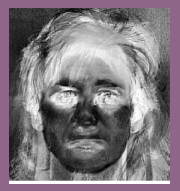
One answer: Cross-validated, e.g. *Jackknife*

Jackknife

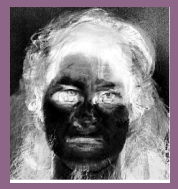
Start with Face 1. And try to predict its sex from the solution derived from the other 5 faces.

$$\hat{y}_1 = \mathbf{x}_1^{\mathsf{T}} \mathbf{B}_{\mathbf{PLS}}^{*}$$

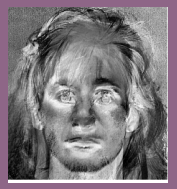
$$-.09 = \quad \times$$



The latent faces:



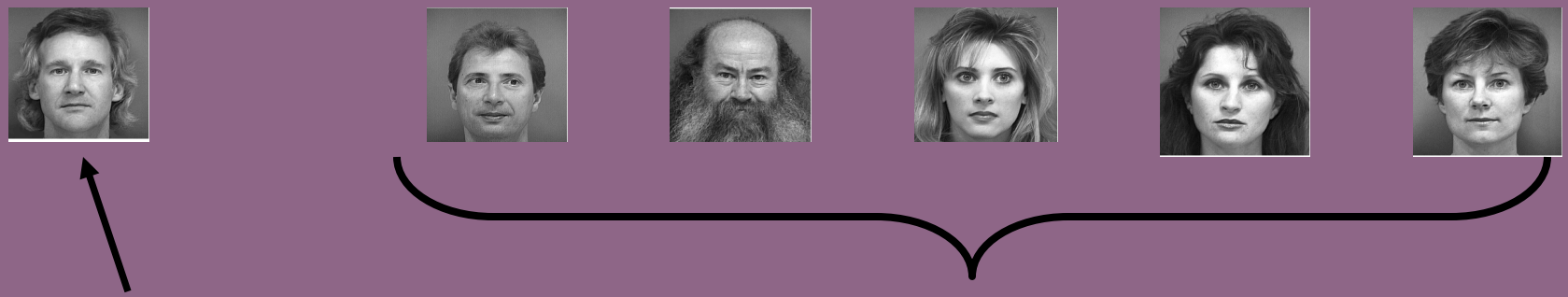
1



2

Jackknife

Continue with Face 2. And try to predict its sex from the solution derived from the other 5 faces.



The testing set

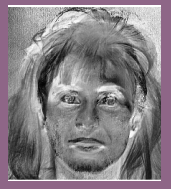
The training set

$$\hat{y}_2 = \mathbf{x}_2^{\mathsf{T}} \mathbf{B}_{\mathbf{PLS}}^{*}$$

$$-.05 = \quad \times$$

The latent faces:

1



2

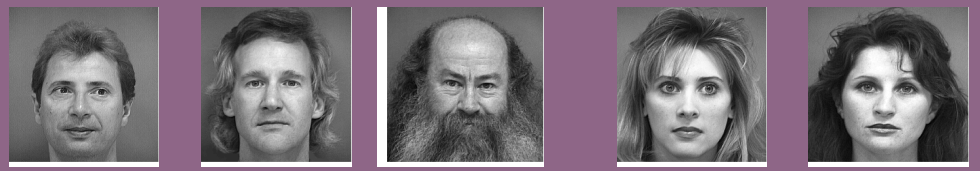
And so on …

… until:

Jackknife

… Face 6. And try to predict its sex from the solution derived from the other 5 faces.

The testing set



The training set

$$\hat{y}_6 = \mathbf{x}_6^{\mathsf{T}} \mathbf{B}_{\mathbf{PLS}}^{*}$$

$$-.48 = \quad \times$$

The latent faces:

1 **2**

The faces

The predictors: B_{PLS}

The first latent faces

The second latent faces

–.09	–.05	2.14	.43	–.52	–.48	$\hat{\mathbf{Y}}$
1	1	1	–1	–1	–1	$\mathbf{Y}$

JACKKNIFE (2 latent vector solution)

–.12	–.08	2.08	.40	–.50	–.60	$\hat{\mathbf{Y}}$
1	1	1	–1	–1	–1	$\mathbf{Y}$

$$r_{\mathbf{Y},\hat{\mathbf{Y}}} = .48, \quad r^2_{\mathbf{Y},\hat{\mathbf{Y}}} = .23$$

SAMPLE (2 latent vector solution)

.79	.96	.97	–.91	–.83	–.98	$\hat{\mathbf{Y}}$
1	1	1	–1	–1	–1	$\mathbf{Y}$

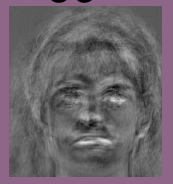
$$r_{\mathbf{Y},\hat{\mathbf{Y}}} = .99, \quad r^2_{\mathbf{Y},\hat{\mathbf{Y}}} = .98$$

More about sex!

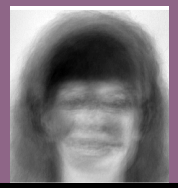
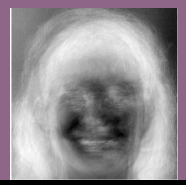
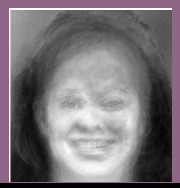
134 faces: 83 women and 51 men

Keep 5 factors/components

PCA

	2	4	3	6	39
					
R_y^2	.29	.13	.09	.05	.03
sum	.29	.42	.51	.52	.59
$R_x^2 = \lambda$	.02	.01	.01	.00+	.00+

PLS

					
R_y^2	.50	.18	.06	.07	.05
sum	.50	.69	.75	.82	.87
R_x^2	.16	.09	.10	.06	.04

Jackknife PCR *vs.* PLS

PCR

Fixed Effect		Jackknife	
R^2	*Misclassifications*	R^2	*Misclassifications*
.77	17/134	.74	20/134

PLS

Fixed Effect		Jackknife	
R^2	*Misclassifications*	R^2	*Misclassifications*
.93	1/134	.79	14/134

A couple of equations

$$\mathbf{X} = \mathbf{T}\,\mathbf{P}^{T} \quad \text{with} \quad \mathbf{T}^{T}\mathbf{T} = \mathbf{I}$$

and

$$\hat{\mathbf{Y}} = \mathbf{T}\,\mathbf{D}_{B}\mathbf{C}^{T}$$

A couple of equations

$$\mathbf{X} = \mathbf{T}\,\mathbf{P}^{T} \quad \textbf{with} \quad \mathbf{T}^{T}\mathbf{T} = \mathbf{I}$$

and

$$\hat{\mathbf{Y}} = \mathbf{T}\,\mathbf{D}_{B}\,\mathbf{C}^{T}$$

The Algorithm (nipals): The big idea

Get a linear combination from **X**

With maximal covariance with **Y**

The Algorithm (nipals) 1

Start with $\boldsymbol{E} = \boldsymbol{X}$, **and** $\boldsymbol{F} = \boldsymbol{Y}$. random **u**

- Step 1. $\boldsymbol{w} \propto \boldsymbol{E}^T\boldsymbol{u}$ **(estimate *X* weights).**
- Step 2. $t \propto \boldsymbol{E}\boldsymbol{w}$ **(estimate *X* factor scores).**
- Step 3. $\boldsymbol{c} \propto \boldsymbol{F}^T\boldsymbol{t}$ **(estimate *Y* weights).**
- Step 4. $\boldsymbol{u} = \boldsymbol{F}\boldsymbol{c}$ **(estimate *Y* scores).**

iterate till $\boldsymbol{t}$ converges, then compute

$\boldsymbol{b} = \boldsymbol{t}^T\boldsymbol{u}$,

and compute the factor loadings for $\boldsymbol{X}$ as

$\boldsymbol{p} = \boldsymbol{E}^T\boldsymbol{t}$.

The Algorithm (nipals) 2

- Now partial out $\boldsymbol{t}$ from both $\boldsymbol{E}$ and $\boldsymbol{F}$ as: $\boldsymbol{E}=\boldsymbol{E}-\boldsymbol{t}\boldsymbol{p}^{\mathrm{T}}$ and $\boldsymbol{F}=\boldsymbol{F}-\boldsymbol{b}\boldsymbol{t}\boldsymbol{c}^{\mathrm{T}}$.
- The vectors $\boldsymbol{t}$, $\boldsymbol{u}$, $\boldsymbol{w}$, $\boldsymbol{c}$, and $\boldsymbol{p}$ are stored in the corresponding matrices, $\boldsymbol{b}$ is stored as a diagonal element of $\boldsymbol{B}$.
- The sum of squares of $\boldsymbol{X}$ (respectively $\boldsymbol{Y}$) explained by the latent vector is computed as $\boldsymbol{p}^{T}\boldsymbol{p}$ (respectively $\boldsymbol{b}^2$).
- As long as $\boldsymbol{E}$ is not full of zeros, go to Step 1.

A (faked) wine example.

We want to predict the subjective evaluation of 5 wines.

The dependent variables are, for each wine, its: likeability, and how well it goes with meat, or dessert

The predictors are: the price, the sugar, alcohol, and acidity content of each wine.

Wine	Price	Sugar	Alcohol	Acidity
1	7	7	13	7
2	4	3	14	7
3	10	5	12	5
4	16	7	11	3
5	13	3	10	3

The matrix **X:** Predictors

Wine	Hedonic	Goes with meat	Goes with dessert
1	14	7	8
2	10	7	6
3	8	5	5
4	2	4	7
5	6	2	4

The matrix **Y:** Measurements (DV).

Wine	t_1	t_2	t_3
1	0.4538	-0.4662	0.5716
2	0.5399	0.4940	-0.4631
3	0	0	0
4	-0.4304	-0.5327	-0.5301
5	-0.5633	0.5049	0.4217

The matrix **T**.

A couple of equations

$$\mathbf{X} = \mathbf{T}\,\mathbf{P}^{T} \quad \text{with} \quad \mathbf{T}^{T}\mathbf{T} = \mathbf{I}$$

and

$$\hat{\mathbf{Y}} = \mathbf{T}\,\mathbf{D}_{B}\,\mathbf{C}^{T}$$

How to predict Y?

- From the latent vectors. Use:

$$\hat{\mathbf{Y}} = \mathbf{T}\,\mathbf{D}_{\mathbf{B}}\,\mathbf{C}^{\mathsf{T}}$$

- From X, with *Z*-scores (for X and Y). Use:

$$\hat{\mathbf{Z}}_{\mathbf{Y}} = \mathbf{Z}_{\mathbf{X}}\,\mathbf{B}_{\mathbf{PLS}}$$

- From X. Use:

$$\hat{\mathbf{Y}} = \mathbf{X}\,\mathbf{B}^{*}_{\mathbf{PLS}}$$

Y dependent variables

X independent variables

	Hedonic	Goes with meat	Goes with dessert
Intercept	48.50	-8.92	-3.85
Price	-1.00	-0.03	0.04
Sugar	0.75	0.28	0.59
Alcohol	-4.00	1.00	0.50
Acidity	2.75	0.18	0.09

The matrix $\boldsymbol{B}^*_{PLS}$ when 3 latent vectors are used.

		Y		
		Hedonic	meat	dessert
	Intercept	48.50	-8.92	-3.85
	Price	-1.00	-0.03	0.04
	Sugar	0.75	0.28	0.59
X	Alcohol	-4.00	1.00	0.50
	Acidity	2.75	0.18	0.09

$$\hat{Y} = X\, B^*_{PLS}.$$

A Wine costing $10, with 5 g of sugar per litre, 10 degree of alcohol and an acidity of 5 (whatever) will have a hedonic rating of

$$\hat{y}_1 = 48.50 - 1\times 10 + .75\times 5 - 4\times 10 + 2.75\times 5 = 16$$

Crunch, Crisp & Crack

- Can we predict what's crispy or crunchy?
- Thank you to Catherine Dacremont (ensbana).

OBJECTIVE: To study the relationships between the 3 texture attributes: *croustillant* (crispy), *craquant* (crackly), *croquant* (crunchy) and 9 attributes describing the biting sounds and other textural characteristics.

Descriptive Analysis

Panel: 8 trained assessors

Attributes

crispy
crunchy
crackly

+

hard
crumbly
puffy
brittle
sticky

eating noises:
- duration
- loudness
- high-pitched
- low-pitched

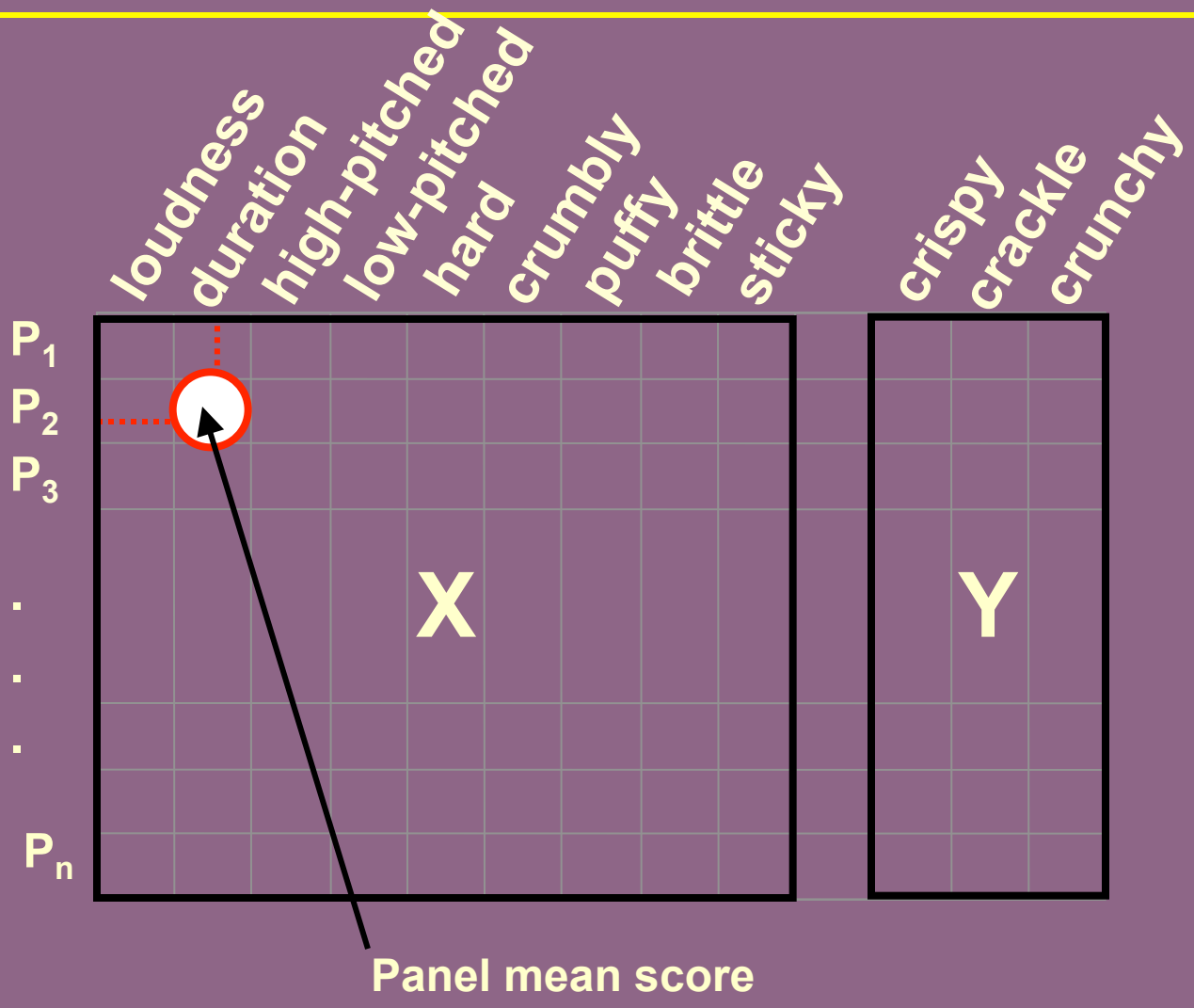
35 Products

extruded bread & snacks
dry biscuits, cold cereals
chips, biscotte, chocolate
raw vegetables & salad
dry fruits & seeds

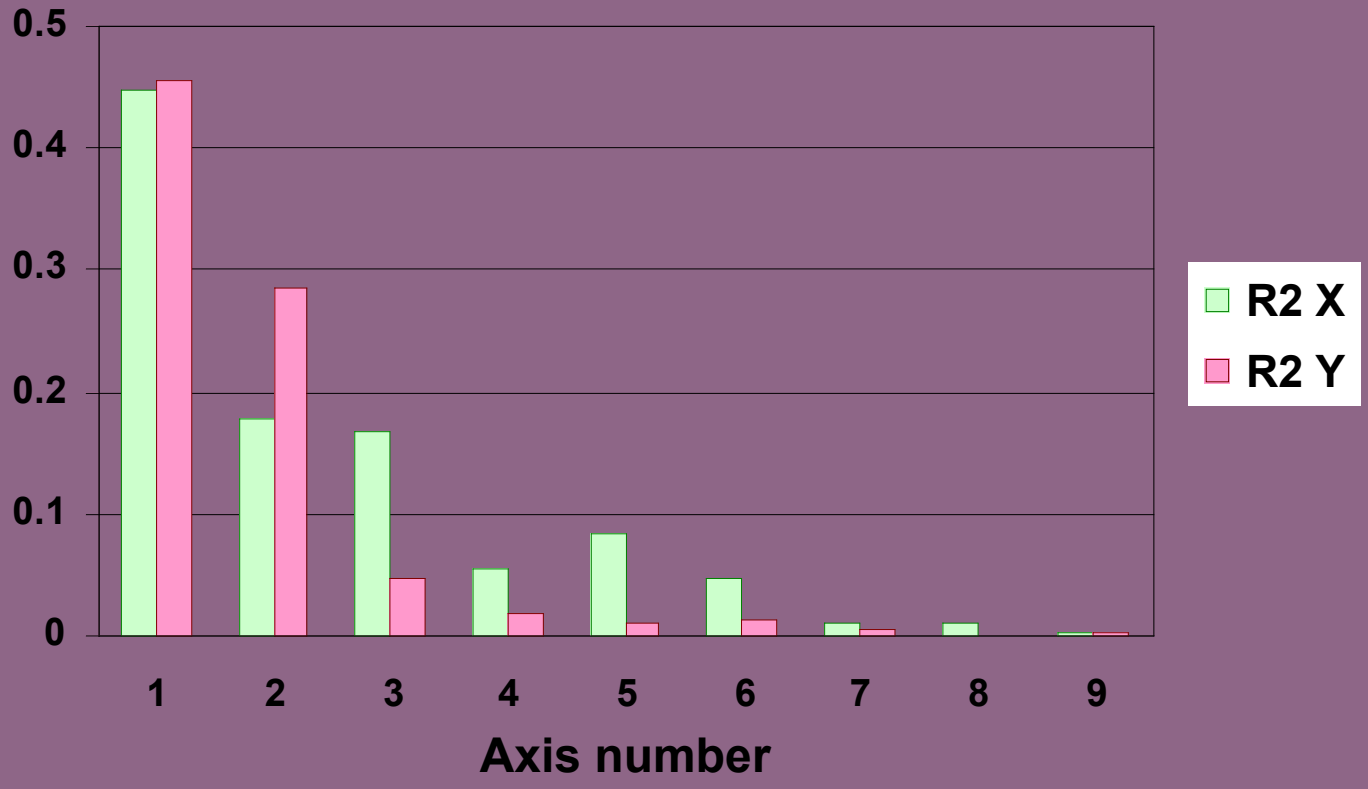
Evaluation

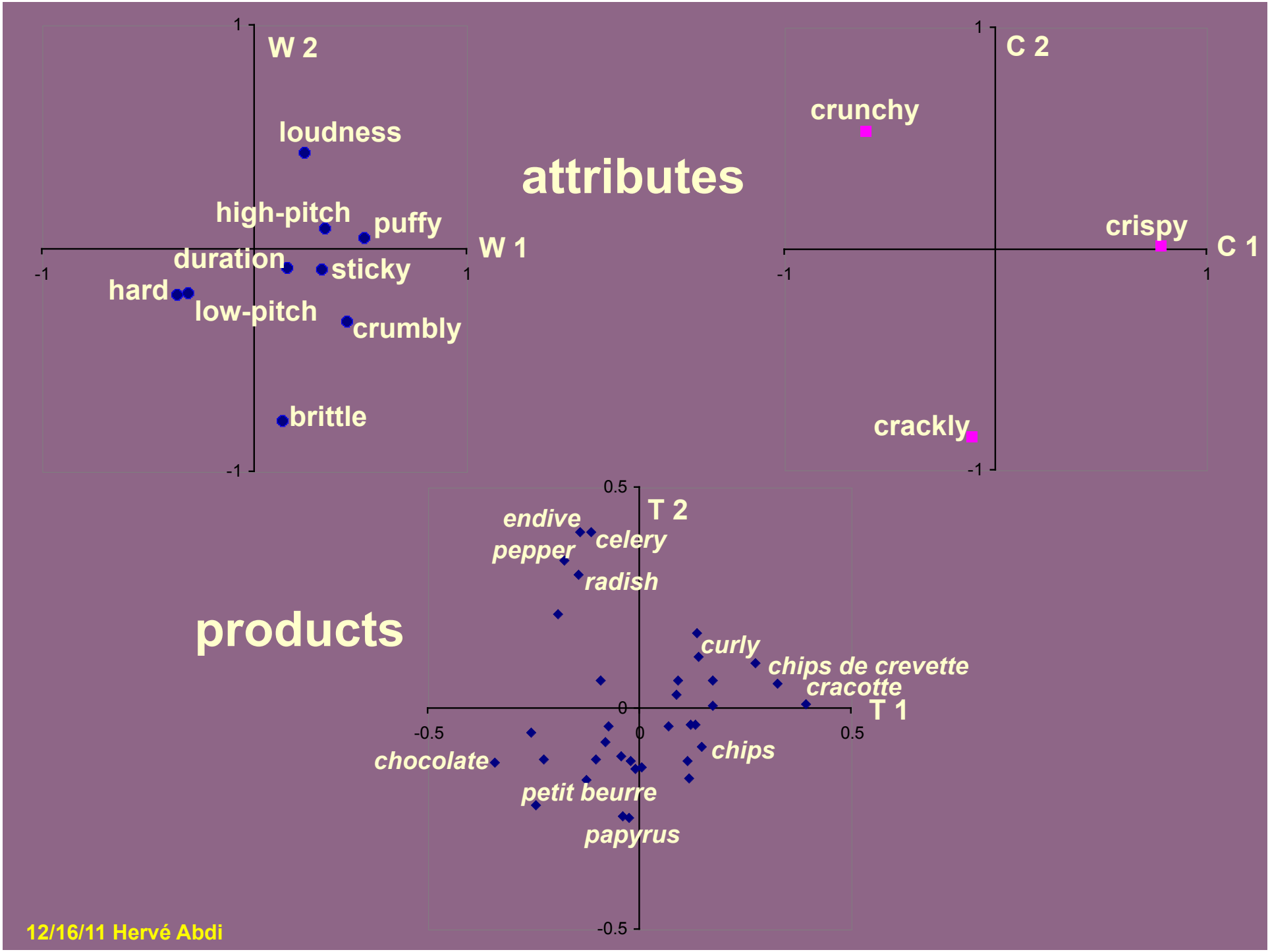
- in duplicate
- linear scale
- scores : 0 → 10

Data

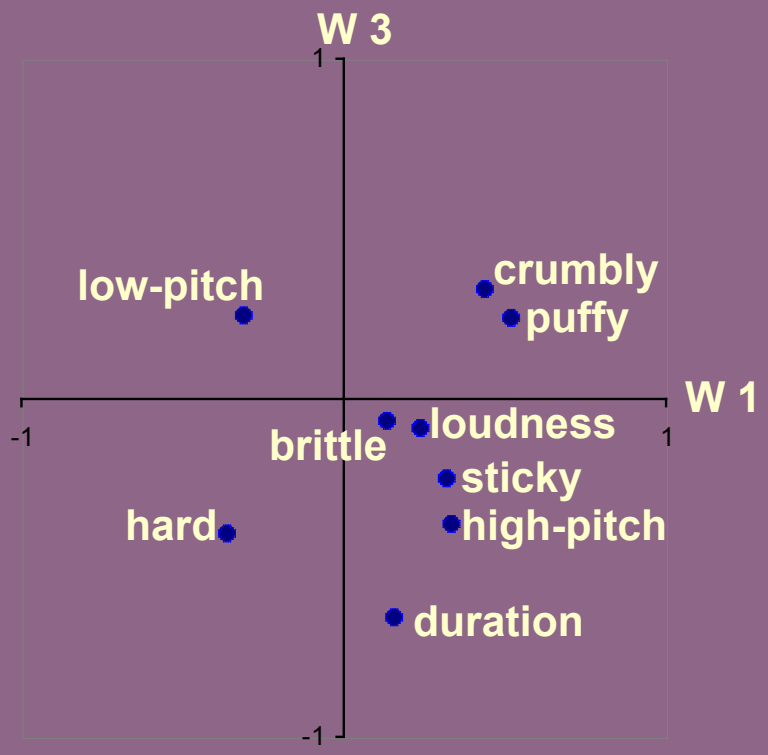

loudness
duration
high-pitched
low-pitched
hard
crumbly
puffy
brittle
sticky
crispy
crackle
crunchy
P1
P2
P3
Pn
X
Y
Panel mean score

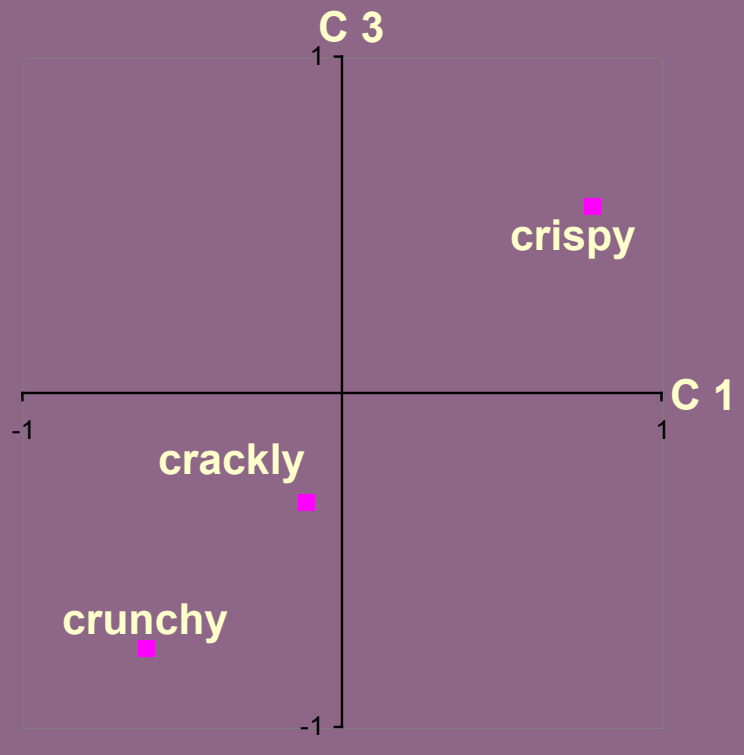
Error

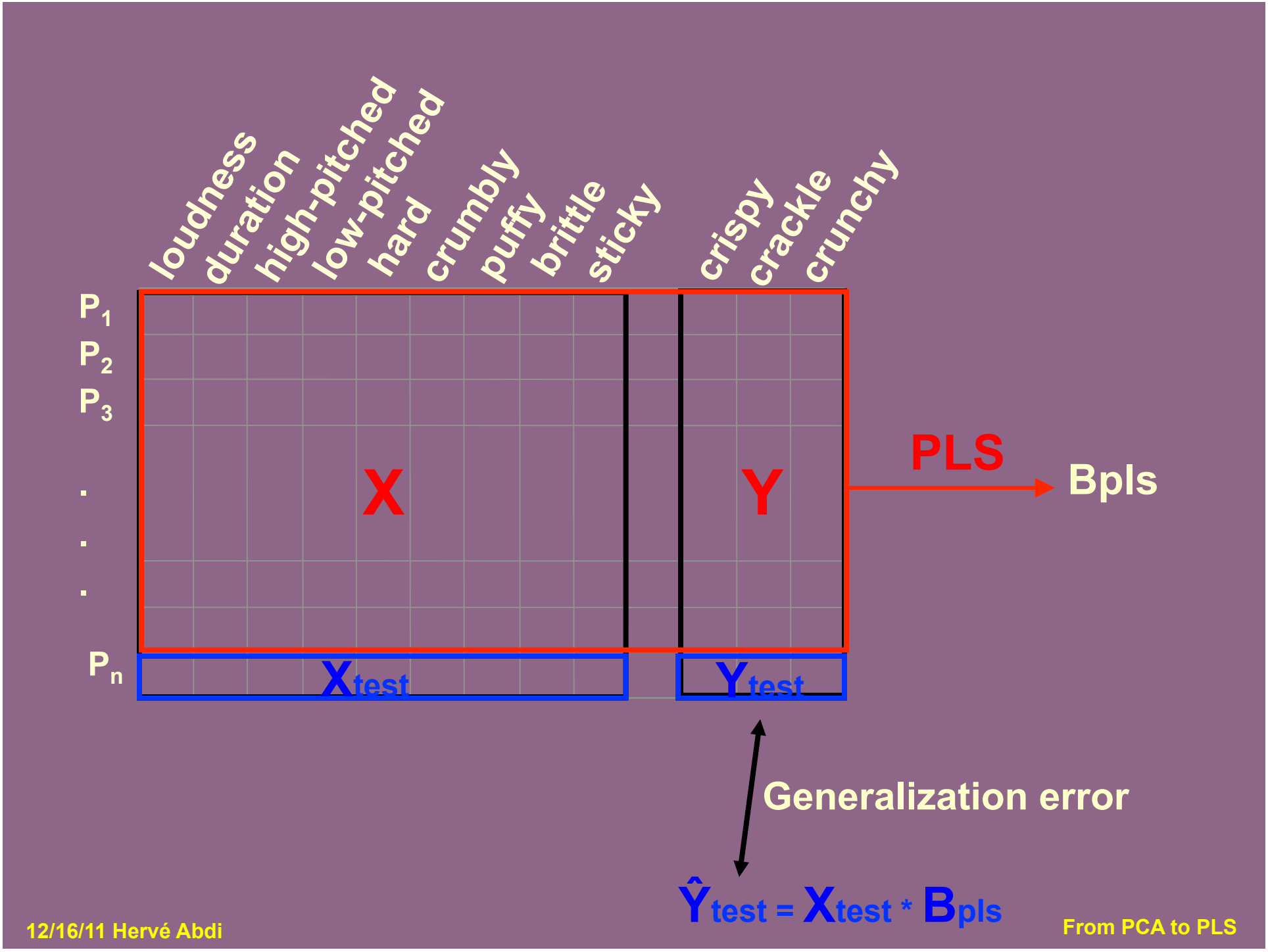




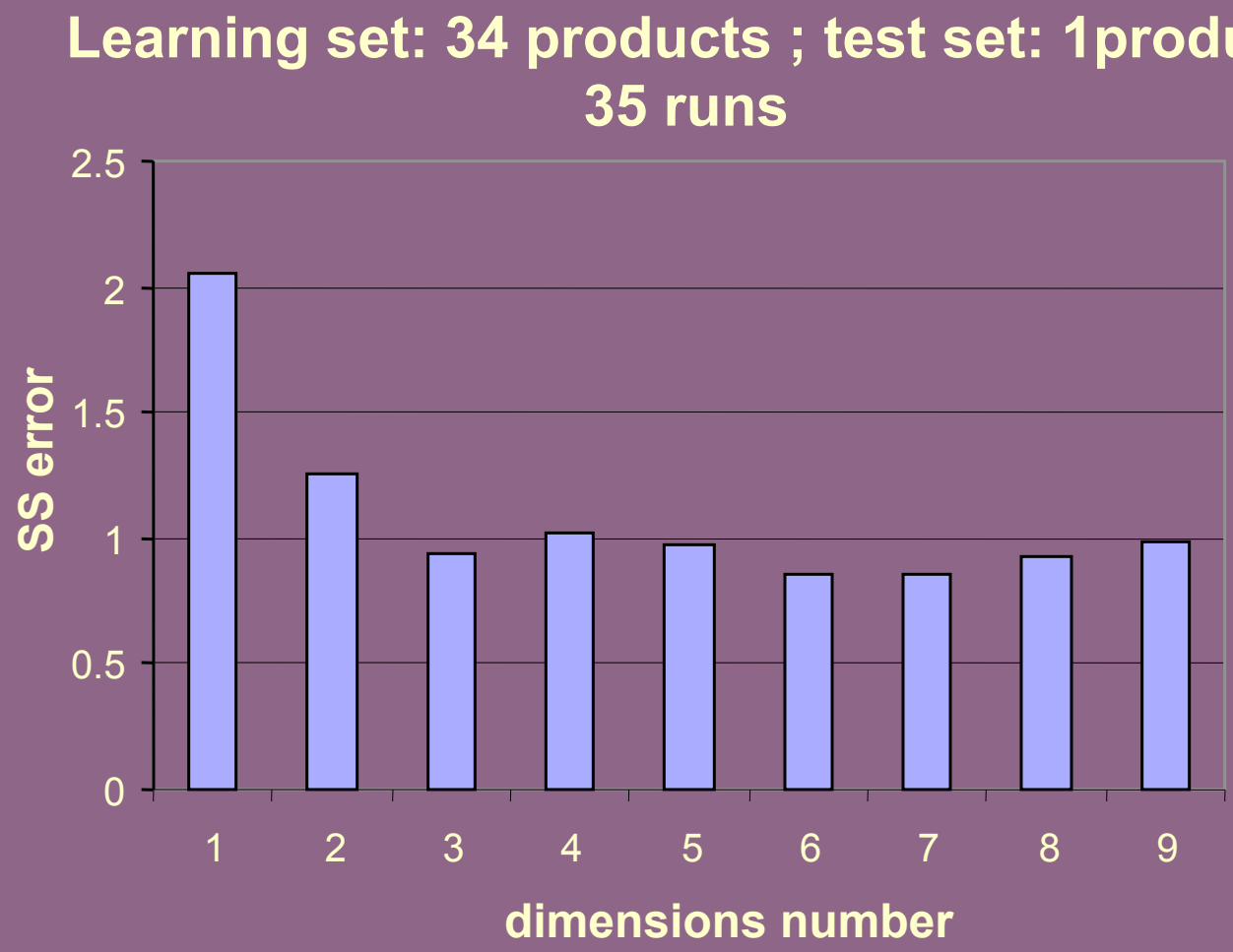
12/16/11 Hervé Abdi


W 3
1
-1
1
-1
W 1
low-pitch
crumbly
puffy
loudness
brittle
sticky
hard
high-pitch
duration

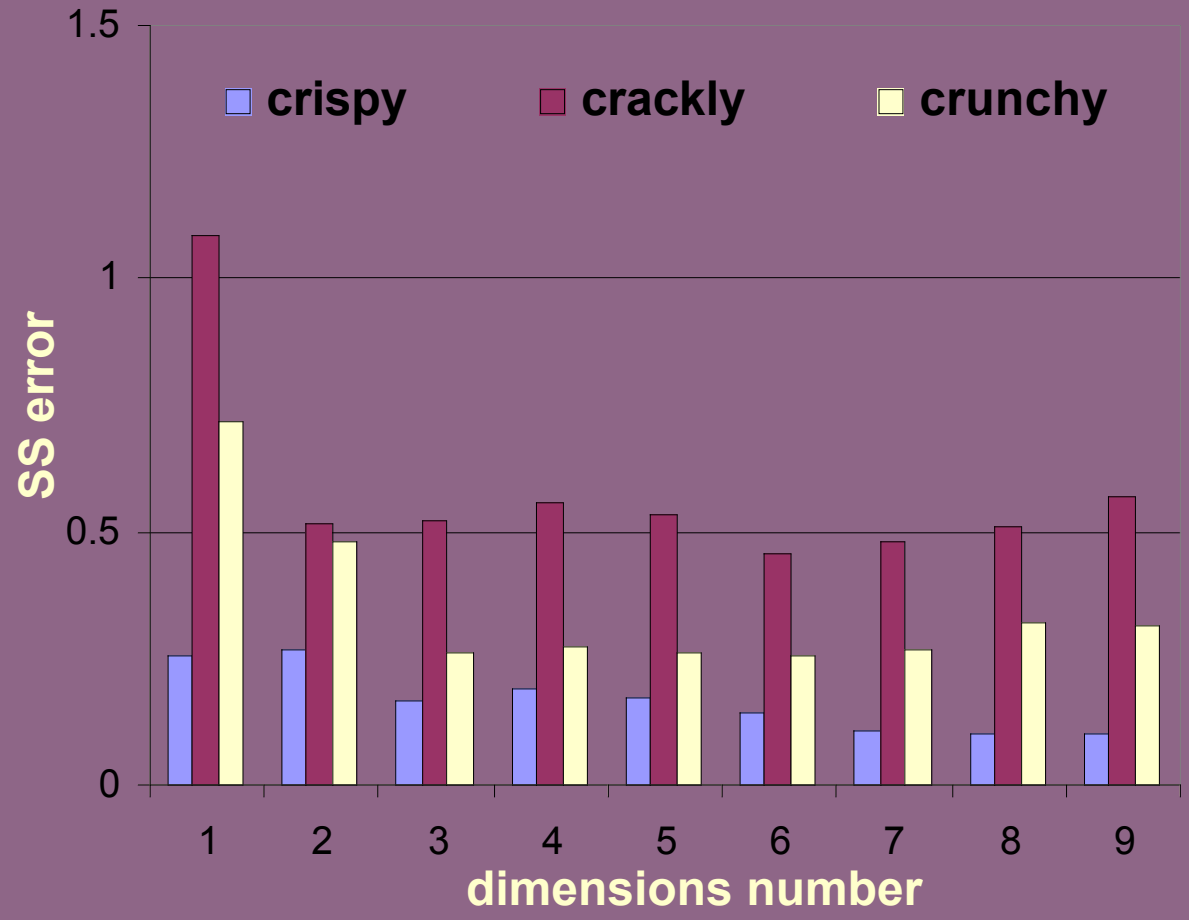

C 3
1
-1
1
-1
C 1
crispy
crackly
crunchy


loudness
duration
high-pitched
low-pitched
hard
crumbly
puffy
brittle
sticky
crispy
crackle
crunchy
P1
P2
P3
.
.
.
Pn
X
Y
PLS
Bpls
Xtest
Ytest
Generalization error
Ŷtest = Xtest * Bpls

Generalisation Error



Generalisation Error



The other race effect!

From: Furl, Phillips, & O'Toole (2002) *Cognitive Science.*

The Problem:

Evaluating 13 algorithms on the "other-race" effect.

The algorithms are described as:

1. **Generic contact or not**
2. **Using a straight PCA model or not**
3. **Using L_1 or L_2 or something else**

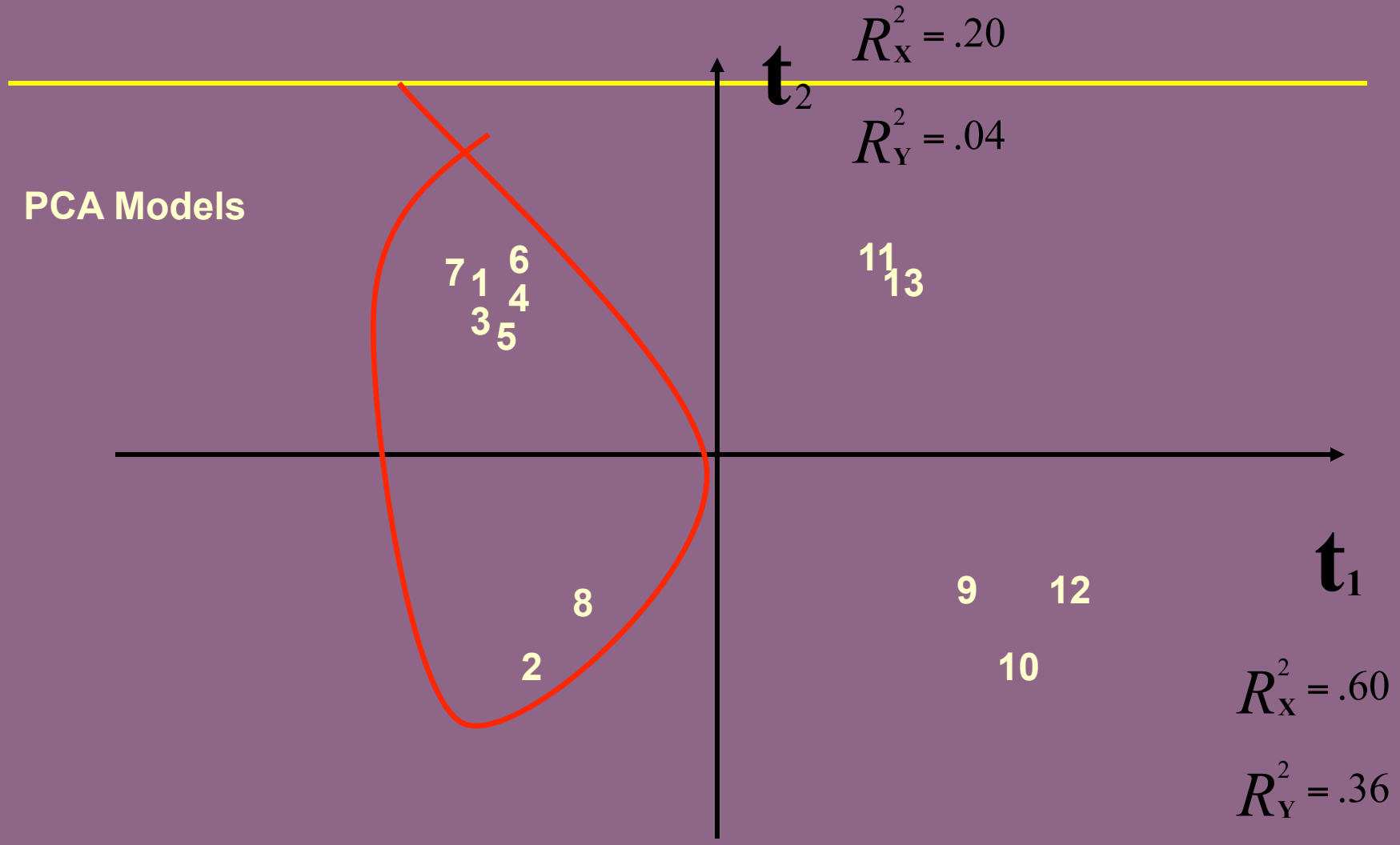
The performance is measured for Caucasian and Asian faces with

1. **Hit, False Alarms, and *A'***

The Data

	Hits		False Alarm		A’	
	Caucasian	Asian	Caucasian	Asian	Caucasian	Asian
Gen Hypo	-0.26	-0.11	0.25	0.15	-0.35	-0.29
No	0.26	0.11	-0.25	-0.15	0.35	0.29
PCA	-0.13	-0.11	0.02	0.00	-0.12	-0.16
No PCA	0.13	0.11	-0.02	-0.00	0.12	0.16
L1	0.13	0.01	-0.32	-0.02	0.28	-0.03
L2	-0.02	0.08	0.12	0.21	-0.08	-0.08
Other	-0.05	-0.08	0.05	-0.19	-0.08	0.09

$$\mathbf{B}_{PLS}$$


PCA Models
t_2
$R^2_X = .20$
$R^2_Y = .04$
7
1
6
4
3
5
11
13
t_1
9
12
8
2
10
$R^2_X = .60$
$R^2_Y = .36$

PLS and alternatives

- PCA (we have already talked about it!)
- PC-regression
- Canonical correlation
- Multiple Factor Analysis
- Multiple Correspondence Analysis
- STATIS (and other variations over TUCKER 3)