FaceID

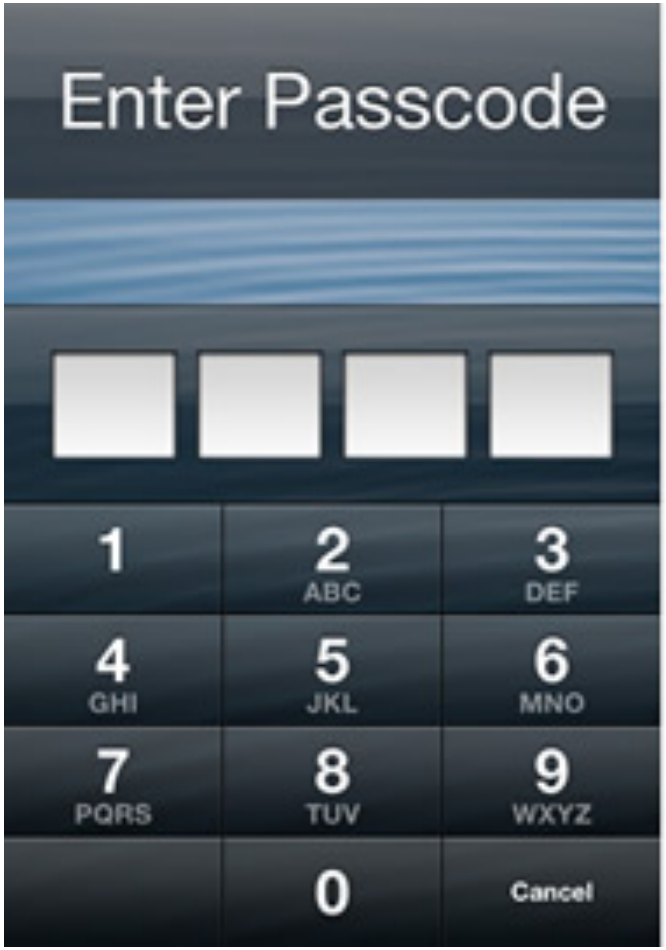
Principal Component Analysis (PCA)

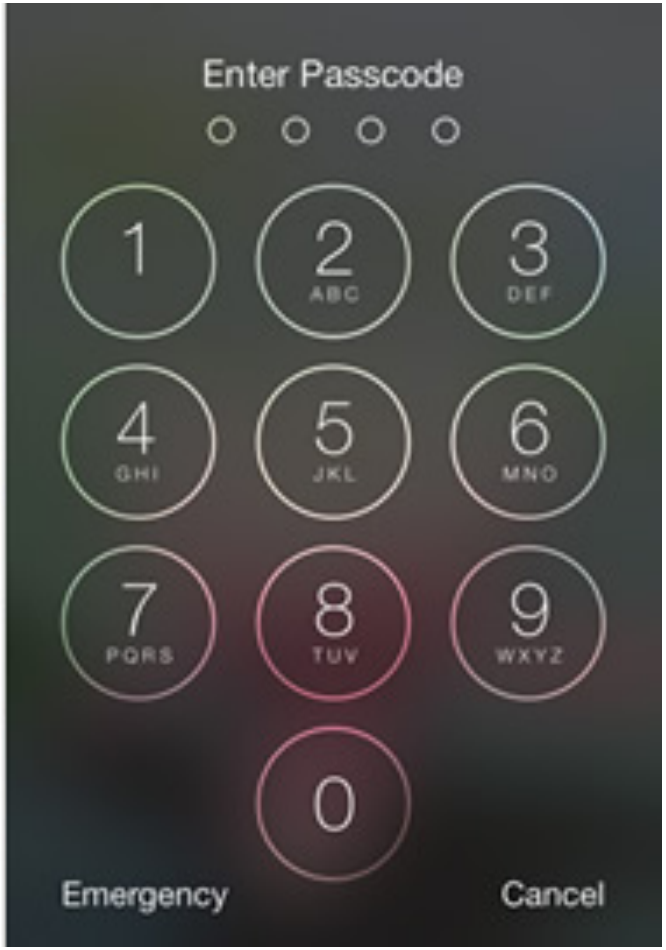
Smart Phone


10:10AM
Apr 15,2015 (Wed)
10:10am
MENU
CONTACTS
KYOCERA


9:42 AM
Text
Calendar
Photos
Camera
YouTube
Stocks
Maps
Weather
Clock
Calculator
Notes
Settings
iTunes
App Store
Phone
Mail
Safari
iPod

Unlock Screen

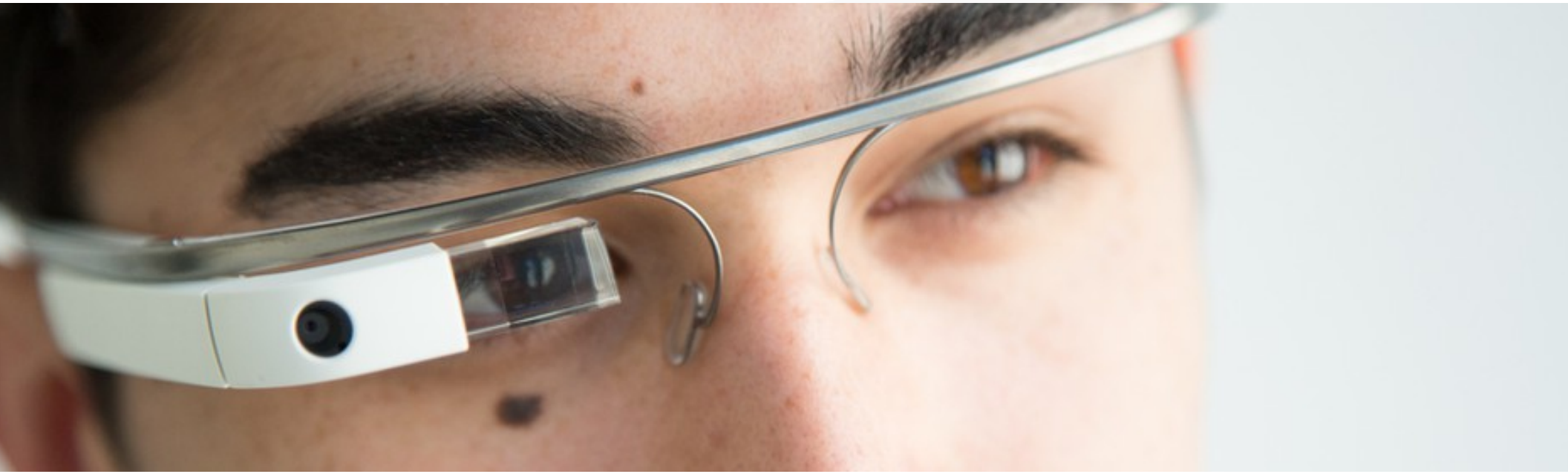


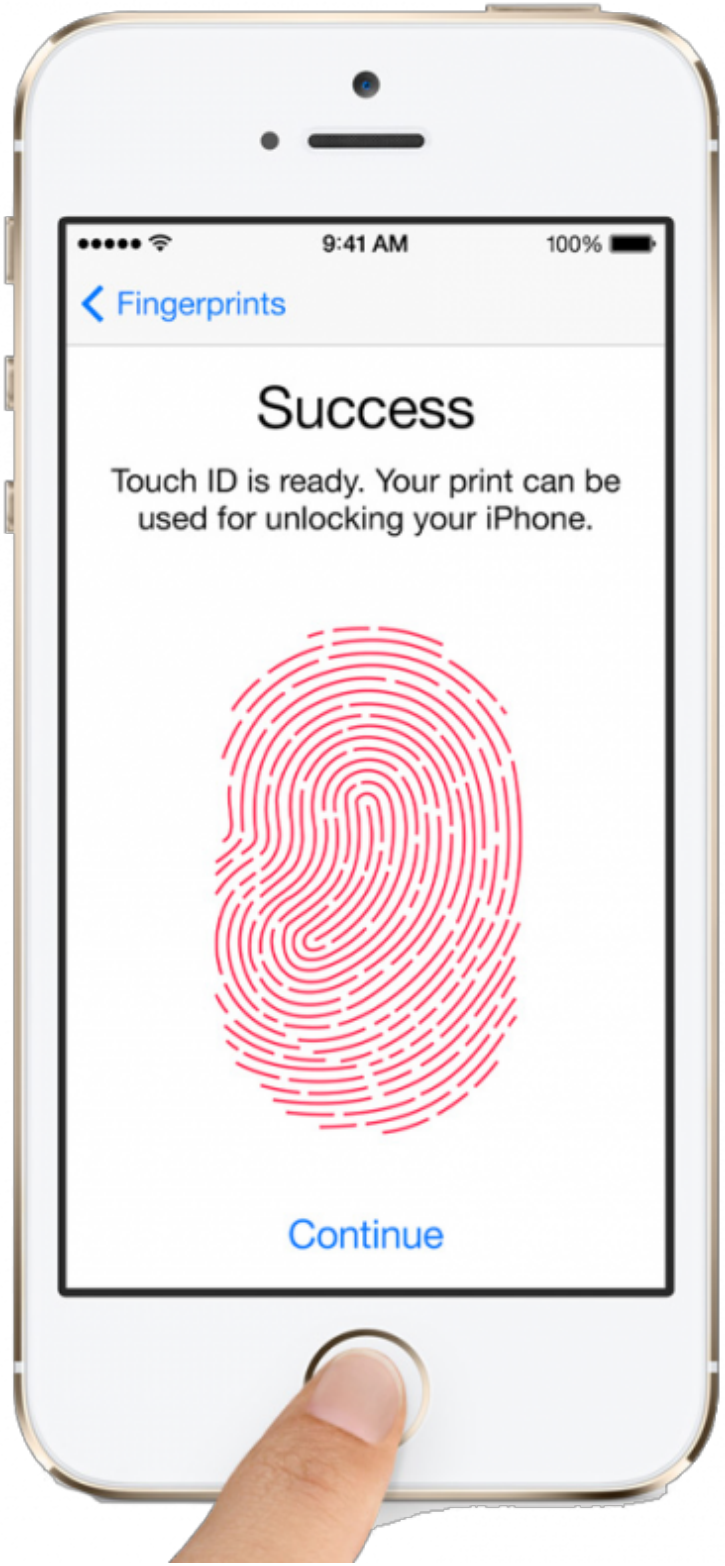


Google Glass wearers can steal passwords from 10ft away

A new computer vision attack could allow Google Glass wearers to steal passwords ty
tablet or smartphones - even if the attackers do not have a clear view of the screen.

Rob Waugh 8 Jul 2014 - 01:22PM




9:41 AM
100%
Fingerprints
Success
Touch ID is ready. Your print can be used for unlocking your iPhone.
Continue

Touch ID

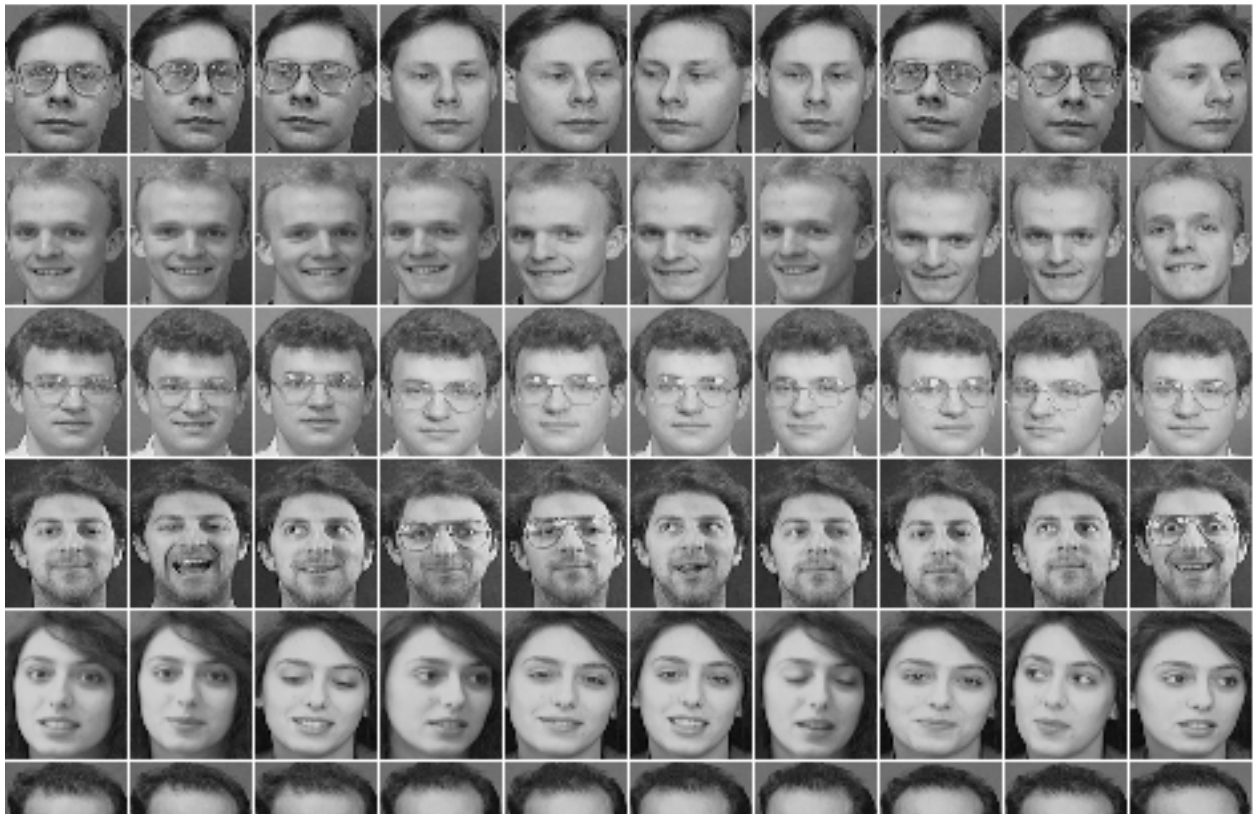
fingerprint recognition

Touch ID

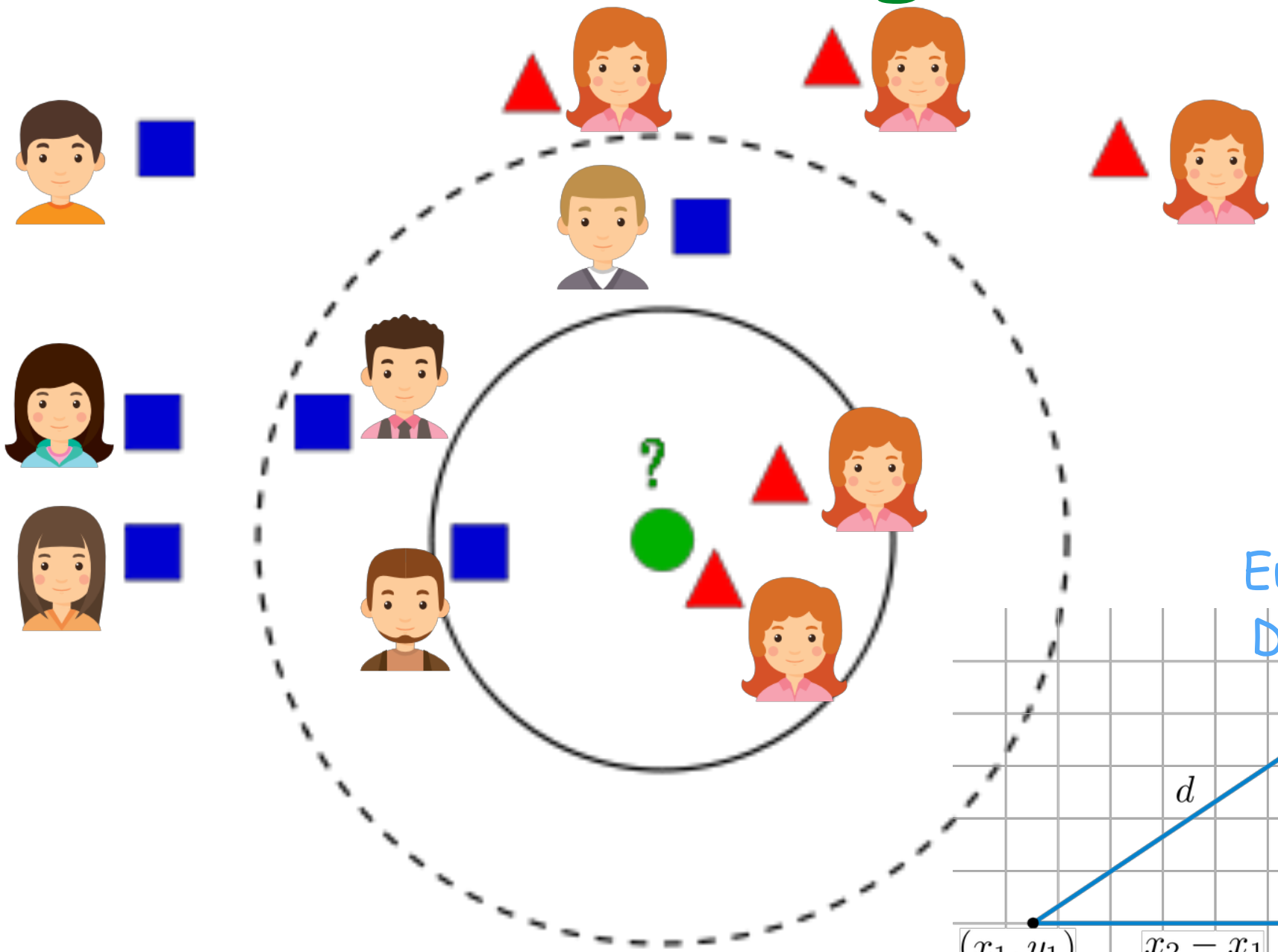
Face ID

Image Data

Arch Left loop Right loop Twin loop Whorl



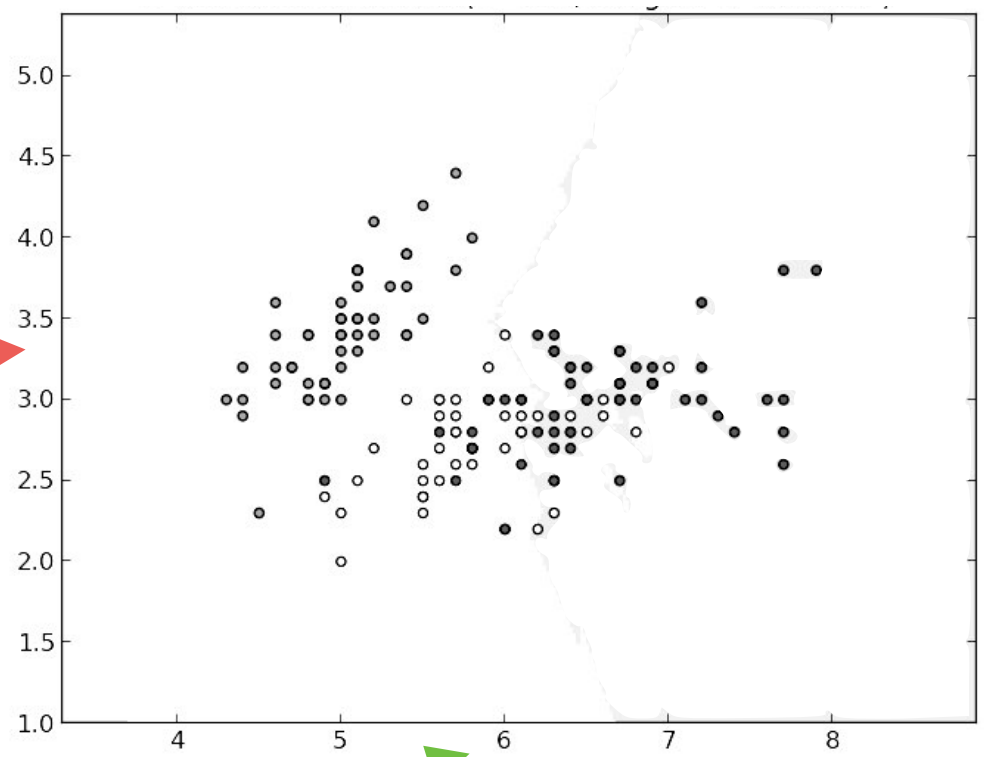
Nearest Neighbor



Euclidean Distance

"Perfect World" for Data Analysis

Volume (cm3)	Pressure (bar)
3.701502	-0.366814
7.858678	0.132824
11.6975	0.907633
15.46395	1.790605
19.18672	2.770453
23.00292	3.734302
26.83077	4.753389
30.75468	5.746156
34.73695	6.742996

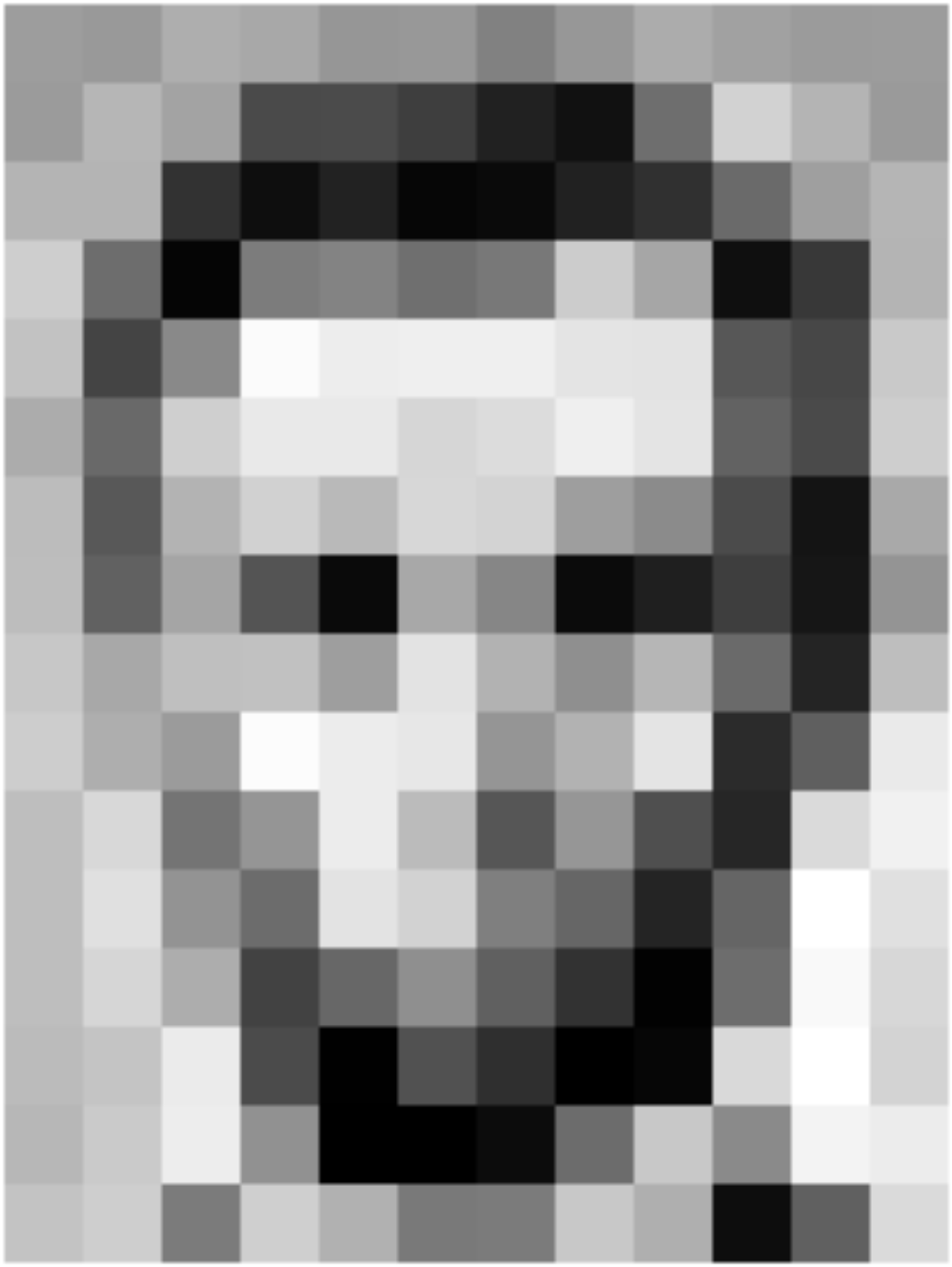


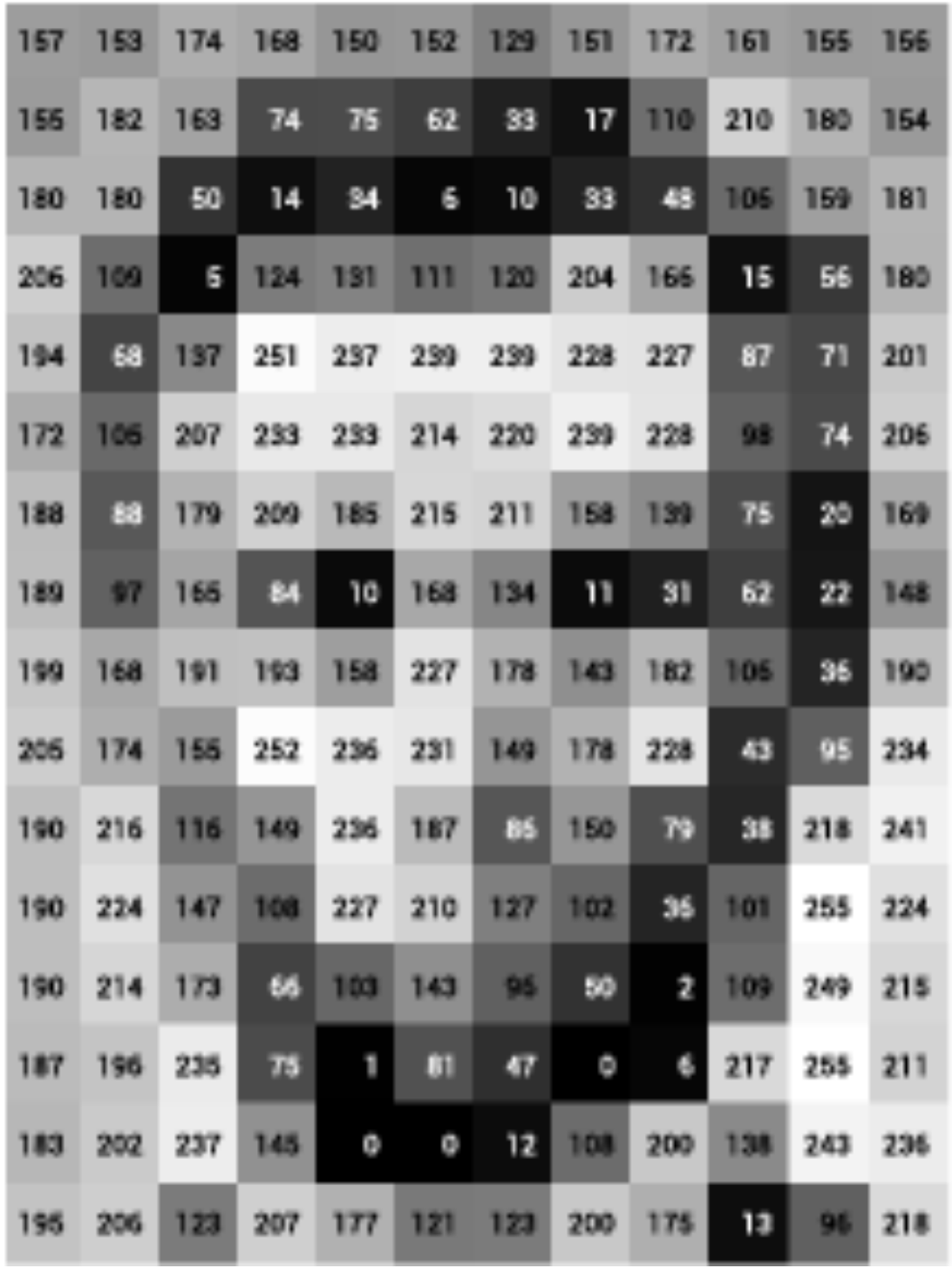
what is this?

157	153	174	168	150	152	129	151	172	161	155	156
155	182	163	74	75	62	33	17	110	210	180	154
180	180	50	14	34	6	10	33	48	106	159	181
206	109	5	124	131	111	120	204	166	15	56	180
194	68	137	251	237	239	239	228	227	87	71	201
172	105	207	233	233	214	220	239	228	98	74	206
188	88	179	209	185	215	211	158	139	75	20	169
189	97	165	84	10	168	134	11	31	62	22	148
199	168	191	193	158	227	178	143	182	106	36	190
205	174	155	252	236	231	149	178	228	43	95	234
190	216	116	149	236	187	86	150	79	38	218	241
190	224	147	108	227	210	127	102	36	101	255	224
190	214	173	66	103	143	96	50	2	109	249	215
187	196	235	75	1	81	47	0	6	217	255	211
183	202	237	145	0	0	12	108	200	138	243	236
196	206	123	207	177	121	123	200	175	13	96	218

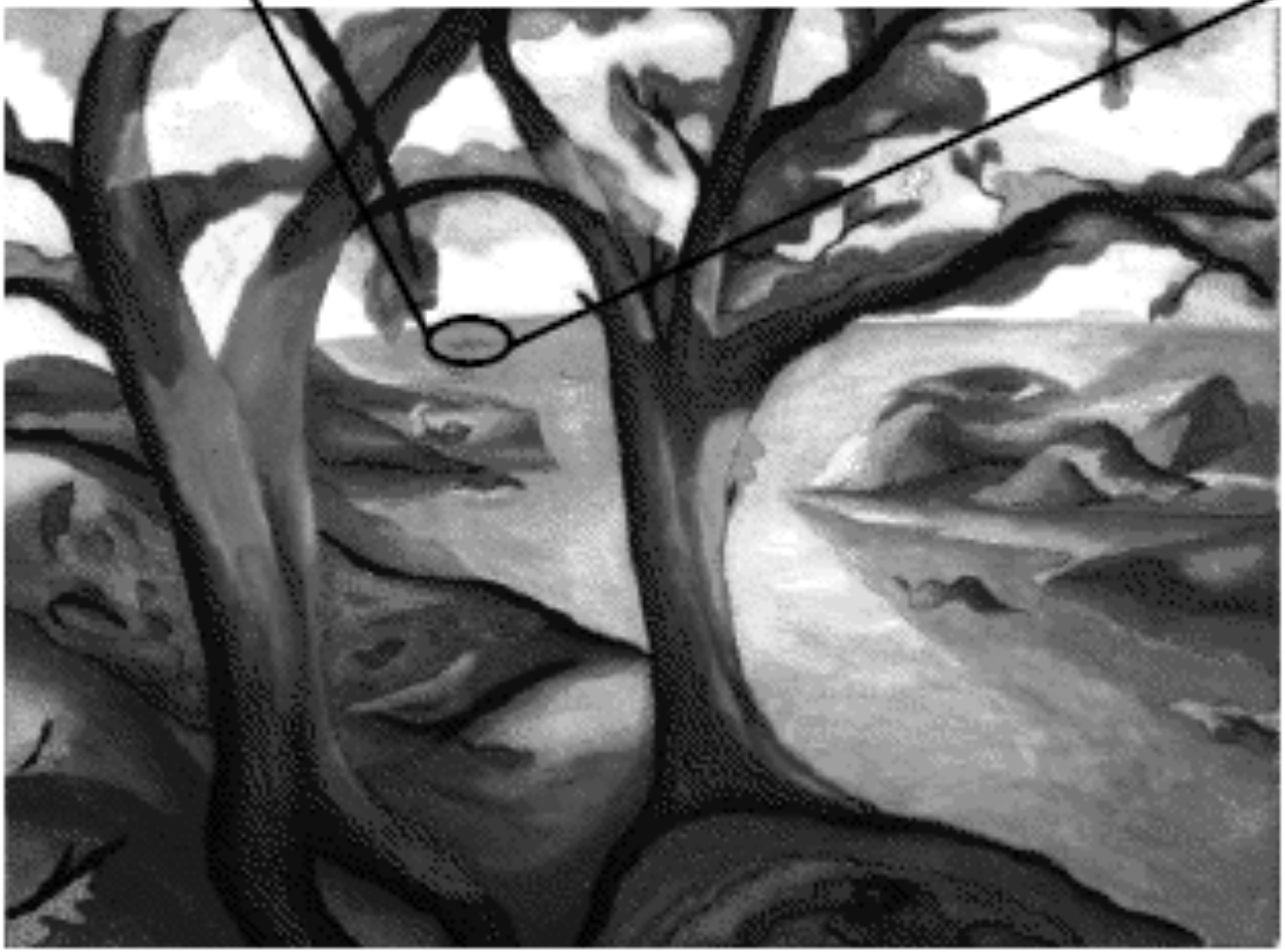
Hint: this is a gray-scale image. Pixel value [0,255]

what is this?

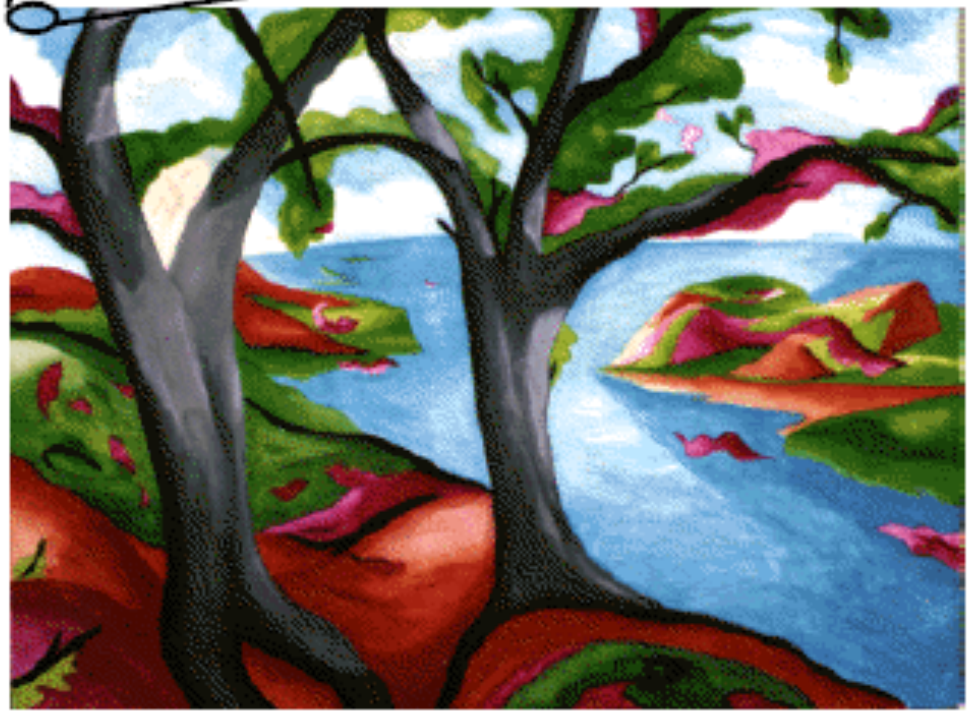



157 153 174 168 150 152 129 151 172 161 155 156
155 182 163 74 75 62 33 17 110 210 180 154
180 180 50 14 34 6 10 33 48 106 159 181
206 109 5 124 131 111 120 204 166 15 56 180
194 68 137 251 237 239 239 228 227 87 71 201
172 105 207 233 233 214 220 239 228 98 74 206
188 88 179 209 185 215 211 158 139 75 20 169
189 97 165 84 10 168 134 11 31 62 22 148
199 168 191 193 158 227 178 143 182 105 36 190
205 174 155 252 236 231 149 178 228 43 95 234
190 216 116 149 236 187 86 150 79 38 218 241
190 224 147 108 227 210 127 102 36 101 255 224
190 214 173 66 103 143 96 50 2 109 249 215
187 196 235 75 1 81 47 0 6 217 255 211
183 202 237 145 0 0 12 108 200 138 243 236
195 206 123 207 177 121 123 200 175 13 96 218

	0.2051	0.2157	0.2826	0.3822	0.4391	
0.5342	0.2251	0.2563	0.2826	0.2826	0.4391	0.4391
0.5342	0.1789	0.1307	0.1789	0.2051	0.3256	0.2483
0.4308	0.2483	0.2624	0.3344	0.3344	0.2624	0.2549
	0.3344	0.2624	0.3344	0.3344	0.3344	



	0.2235	0.1294	**Blue**	0.4196	0.	
.5804	0.2902	**0.0627**	0.2902	0.2902	0.4824	
0.5804	0.0627	0.0627	0.0627	0.2235	0.2588	0.2
0.5176	0.1922	0.0627	**Green**	0.1922	0.2588	0.2588	062
0.5176	0.1294	**0.1608**	0.1294	0.1294	0.2588	0.2588	094
0.5176	0.1608	0.0627	0.1608	0.1922	0.2588	0.2588	
5490	0.2235	0.5490	**Red**	0.7412	0.7765	0.7765	902
490	0.3882	**0.5176**	0.5804	0.5804	0.7765	0.7765	196
0	0.2588	0.2902	0.2588	0.2235	0.4824	0.2235	
	0.2235	0.1608	0.2588	0.2588	0.1608	0.2588	
	88	0.1608	0.2588	0.2588	0.2588	0.	

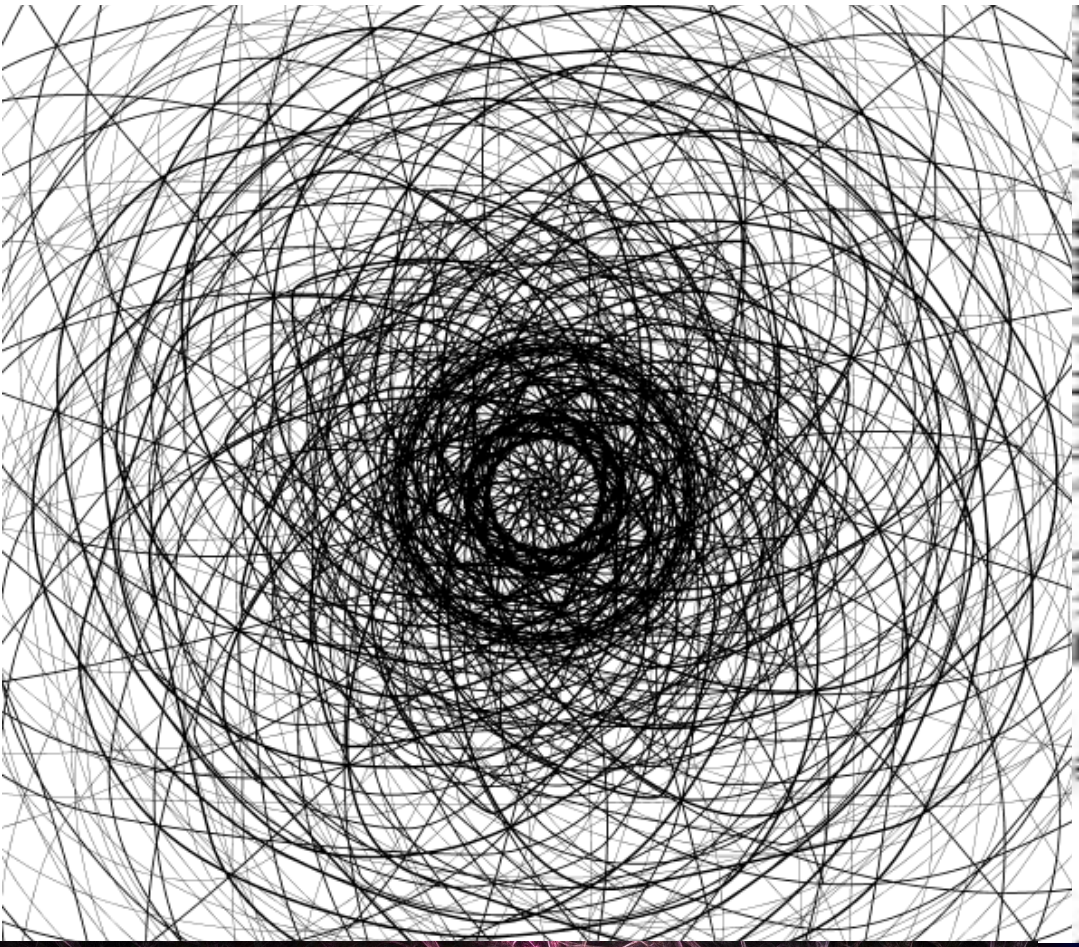


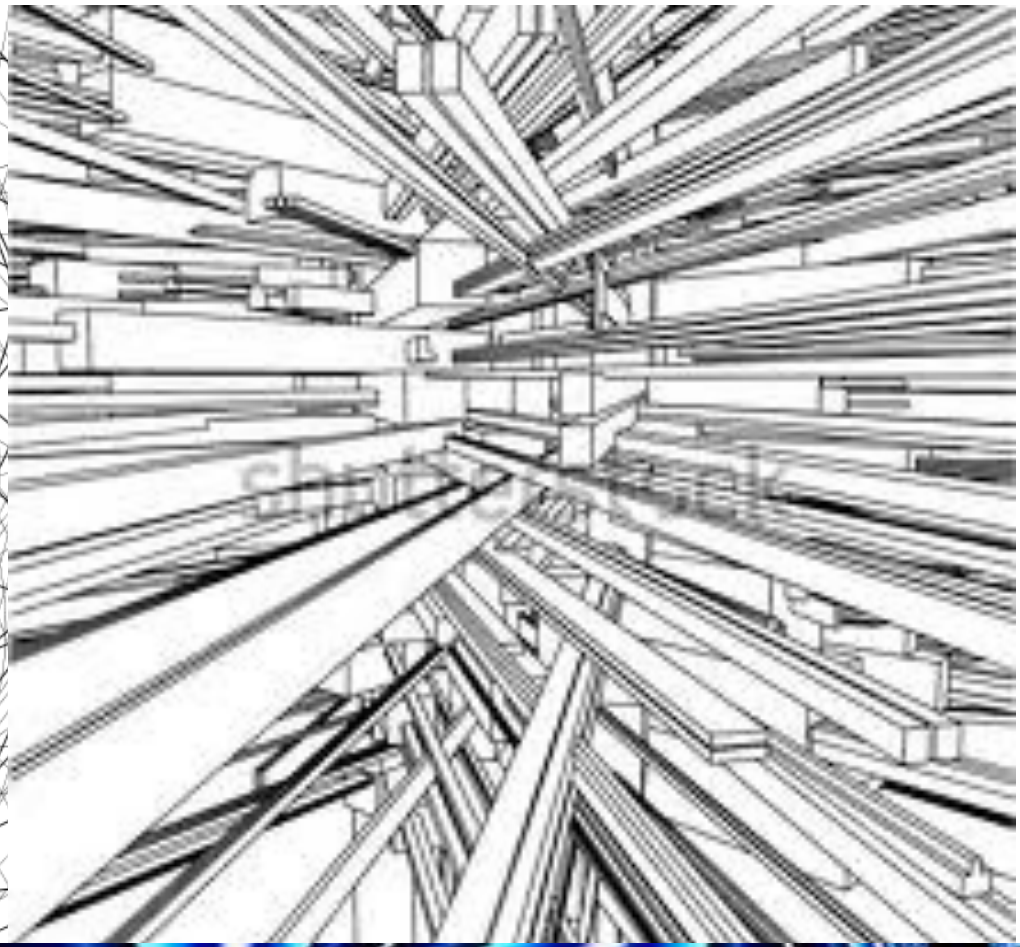
High Dimensional Data

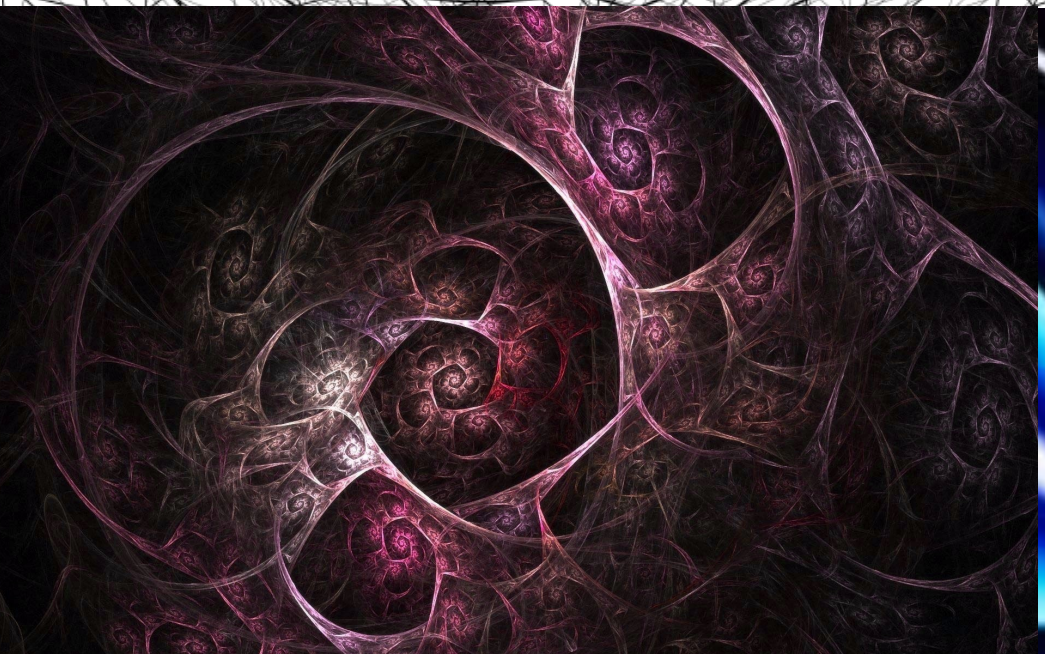
https://youtu.be/C6kn6nXMWF0?t=5s

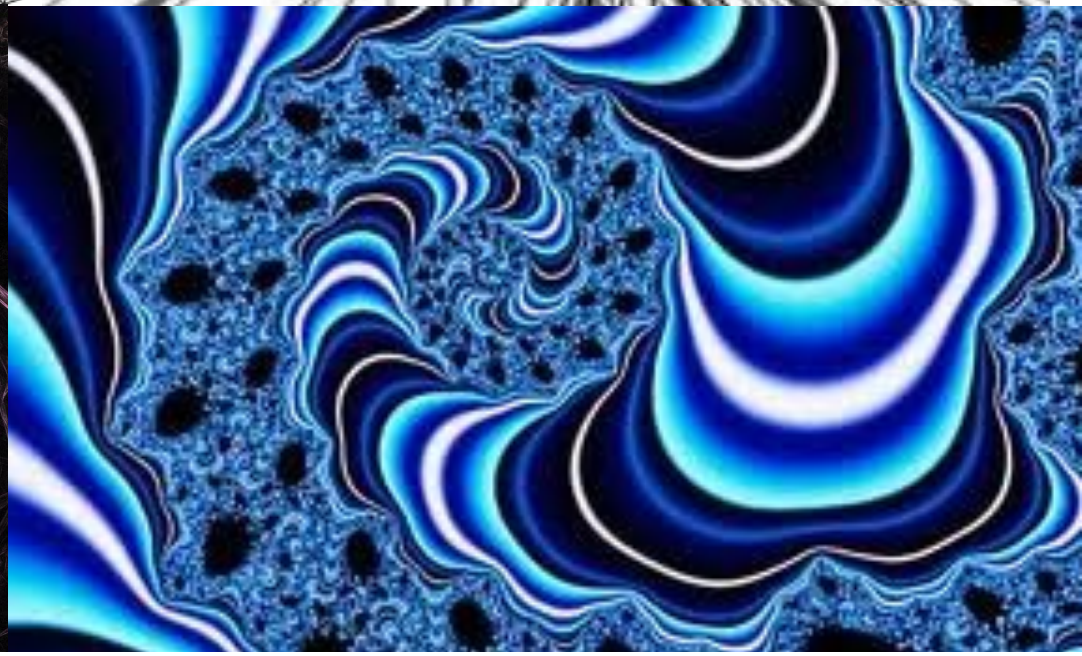
What is really happening in High Dimensional Space?

Chaos



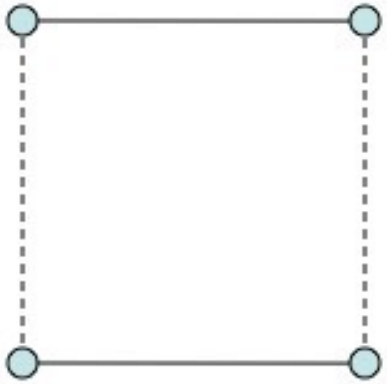




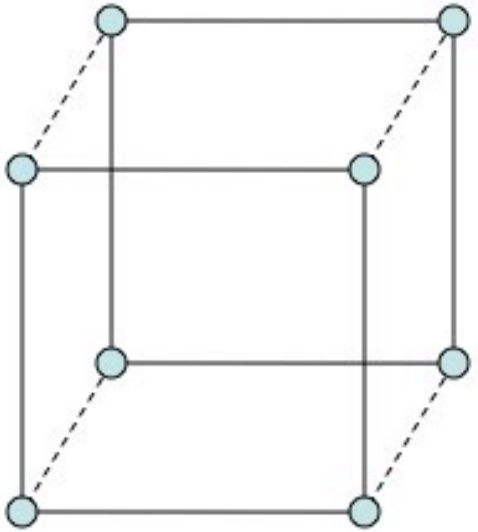


Line - Square - Cube - HyperCube

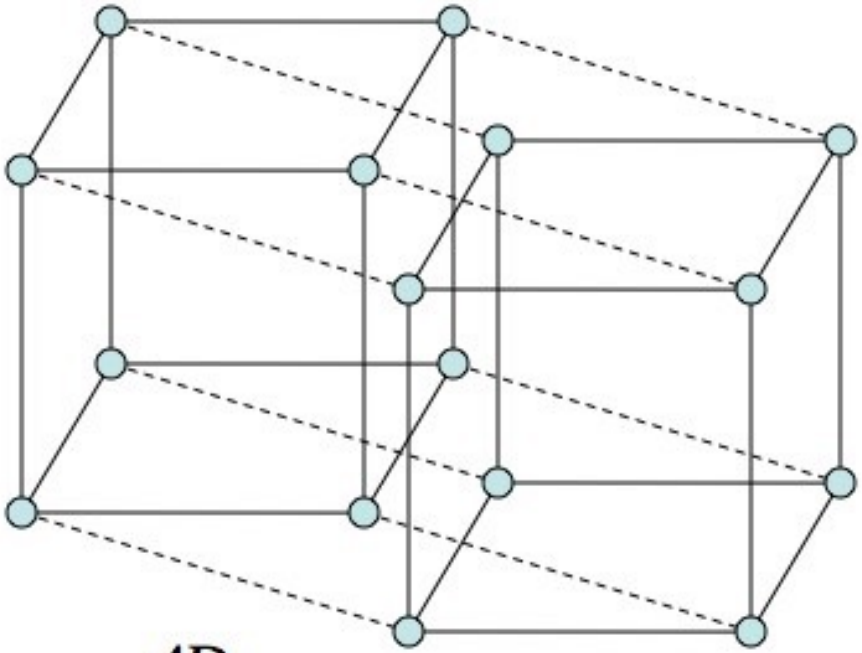
1D



2D



3D



4D

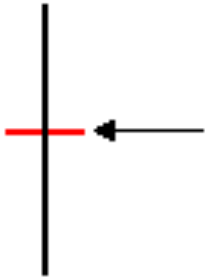
“Cube”

Sphere

POINT

dimensions o

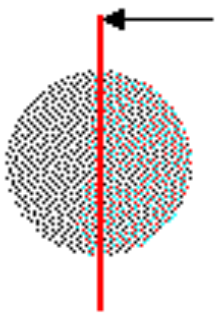
$L = 0$



LINE

dimensions 1

$L = K$

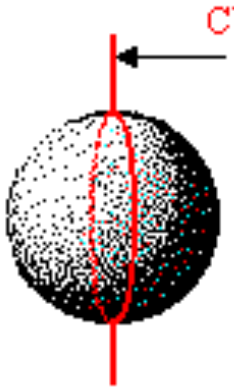


CIRCLE

dimensions 2

Circum = $2 \pi r$

Area = πr^2



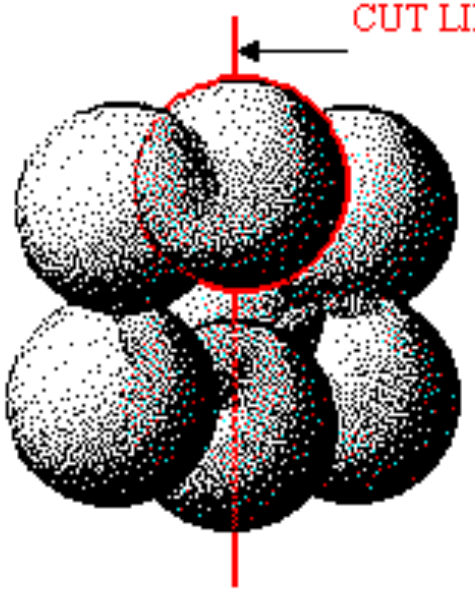
SPHERE

dimensions 3

Circum = $2 \pi r$

Area = $4 \pi r^2$

Volume = $4/3 \pi r^3$



HYPERSPHERE

dimensions 4

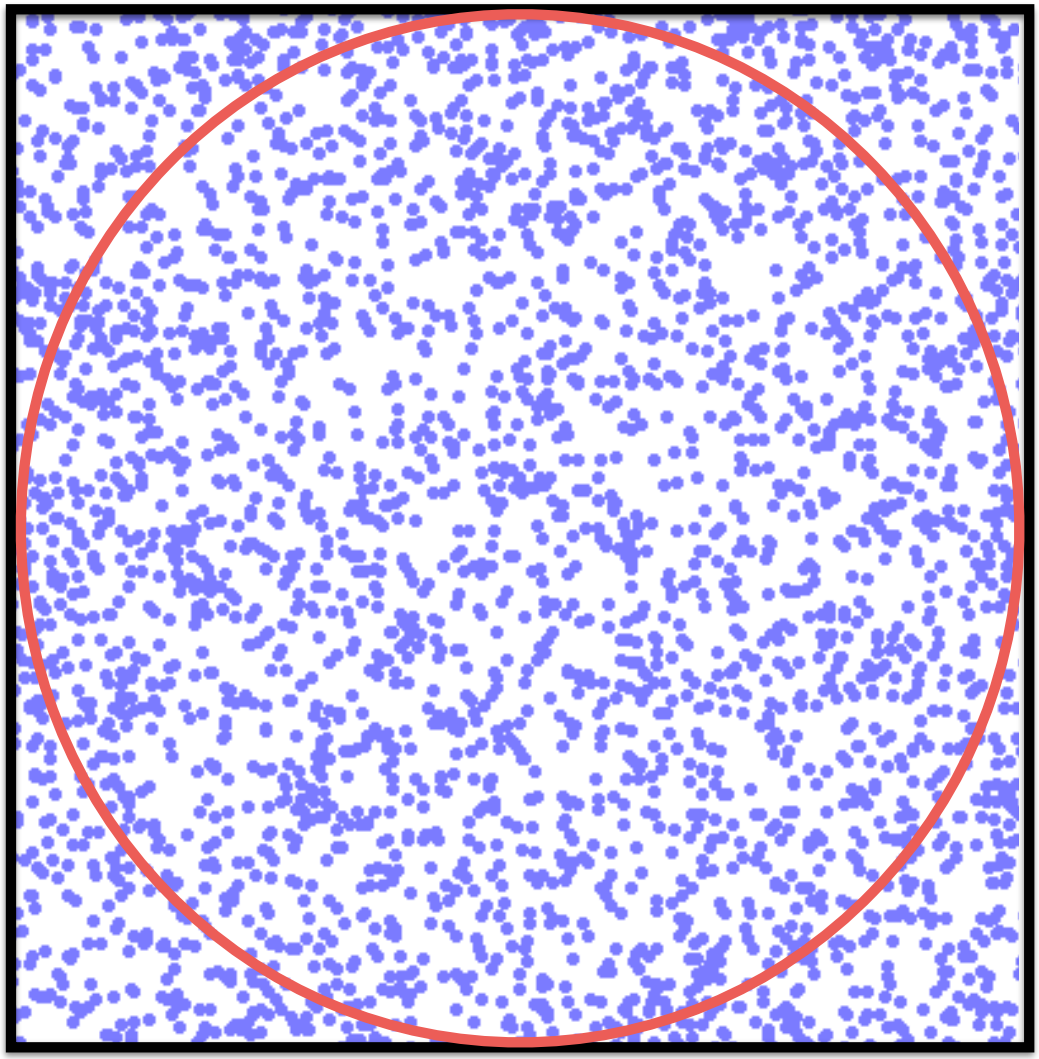
Volume = $2 \pi^2 r^3$

Hyper-volume = $1/2 \pi^2 r^4$

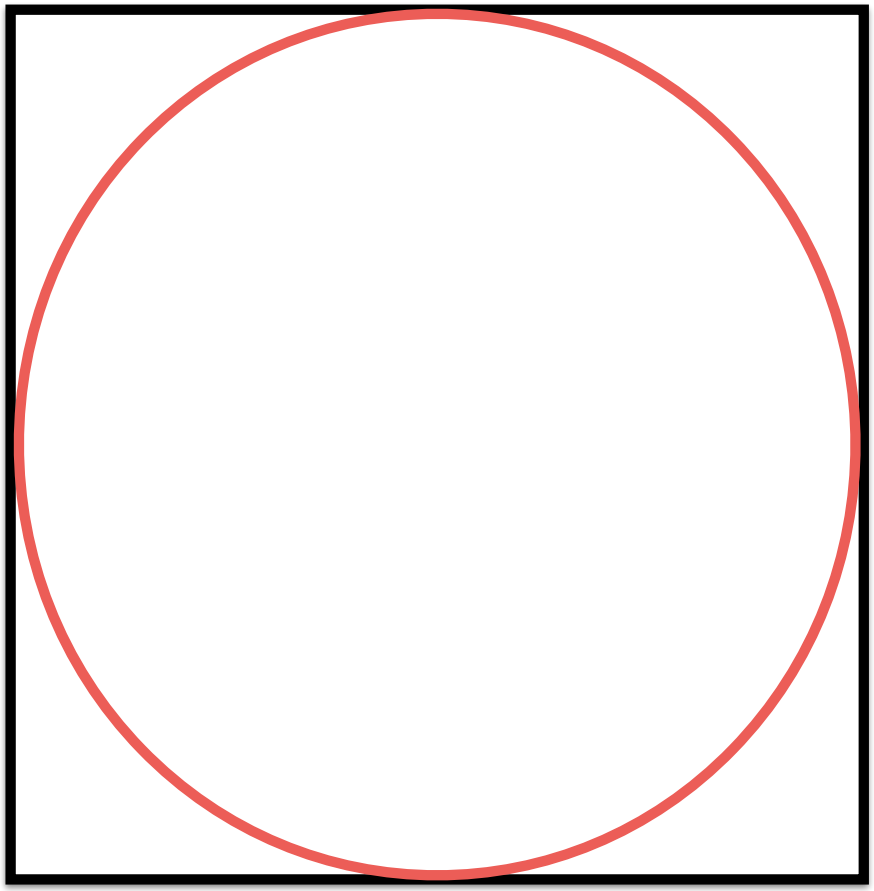
SHOWING OBJECTS OF VARIOUS DIMENSIONS

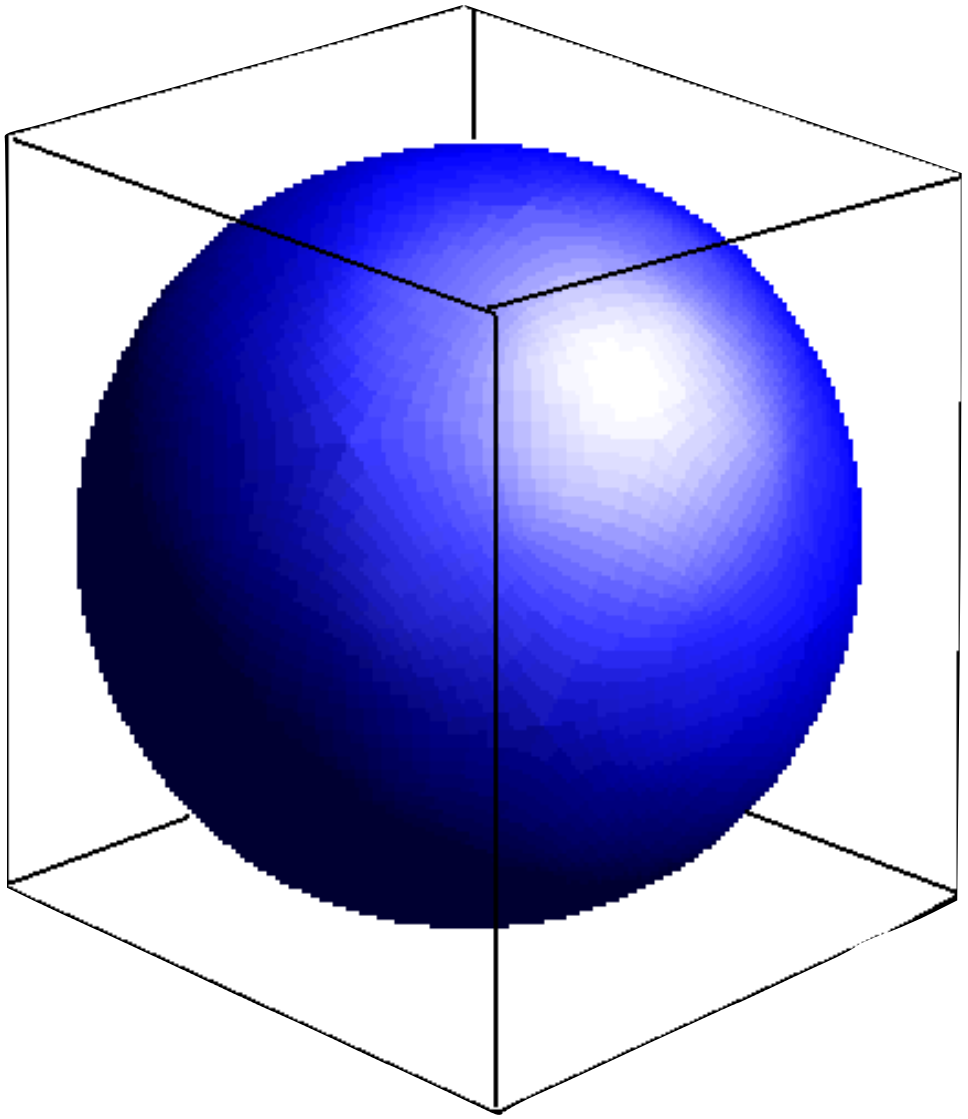
Question #1:

Uniform Distribution (Center vs Corner)

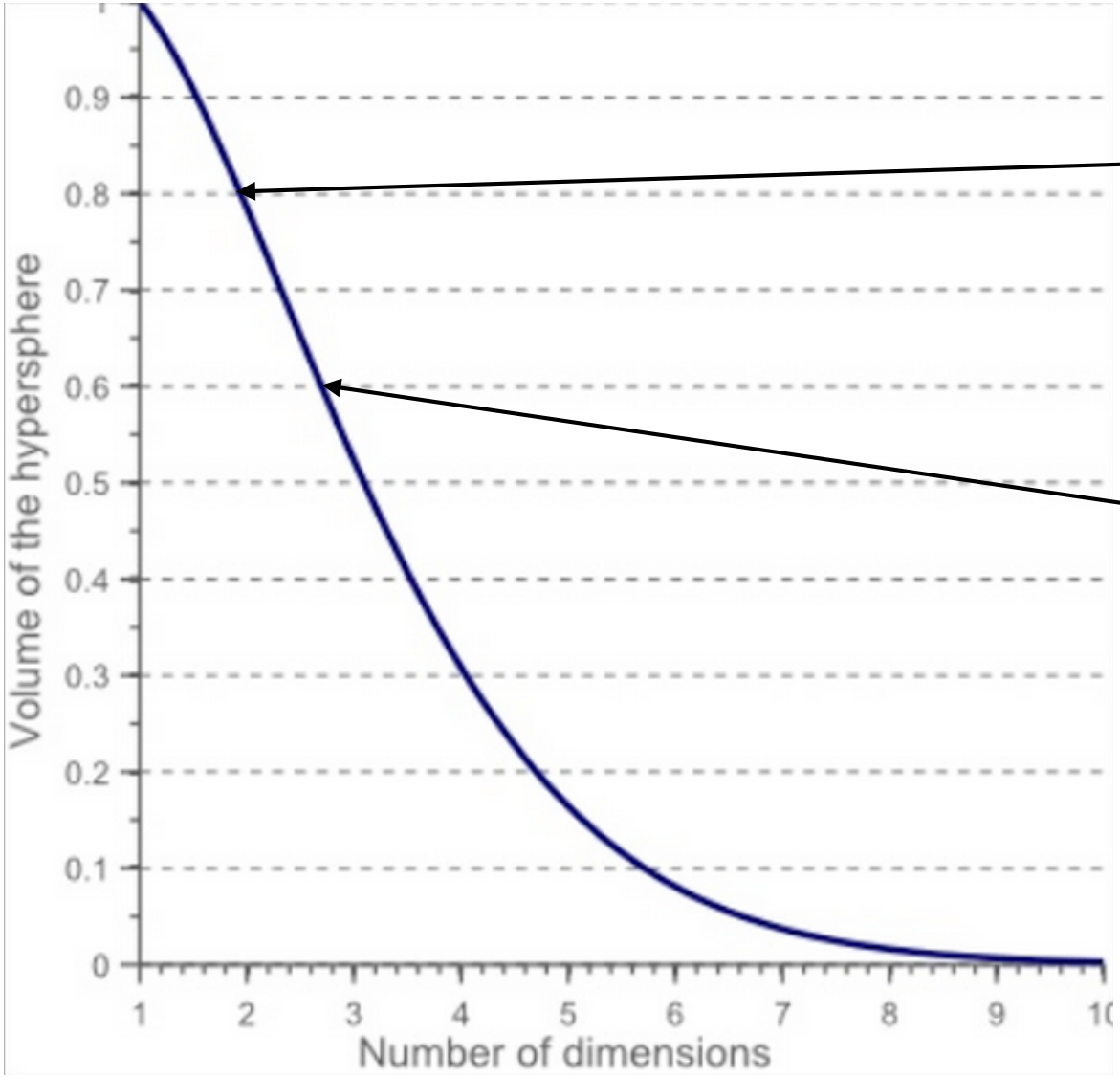


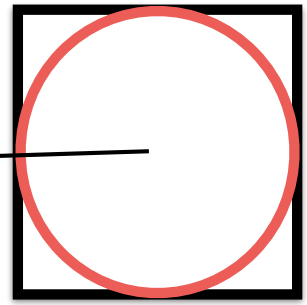
Center vs Corner

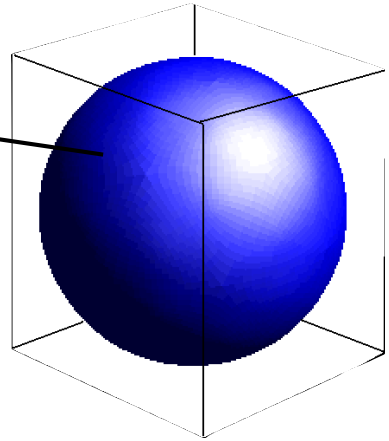




Volume of the hypersphere

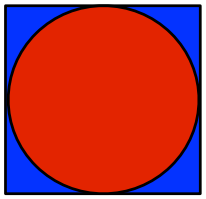




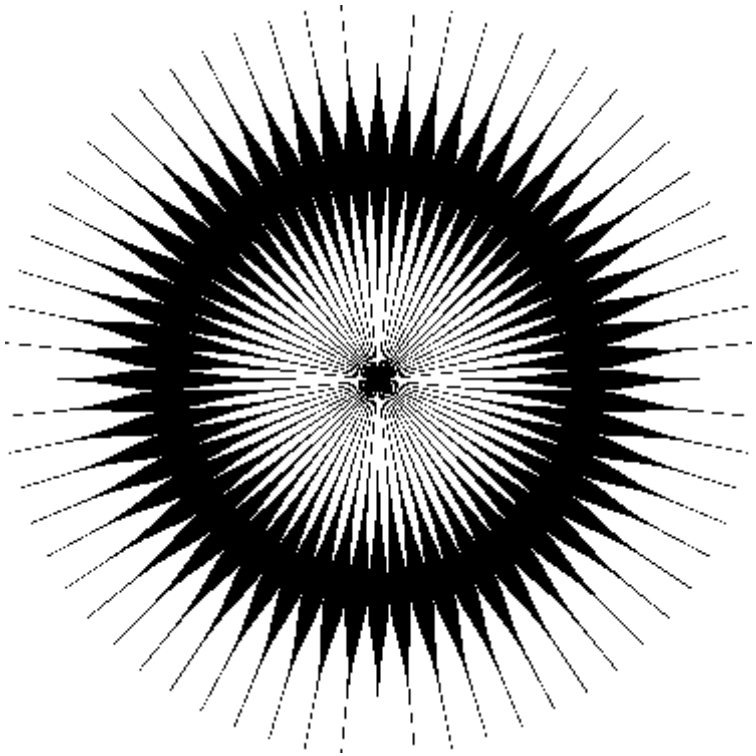


High Dimensional Cube

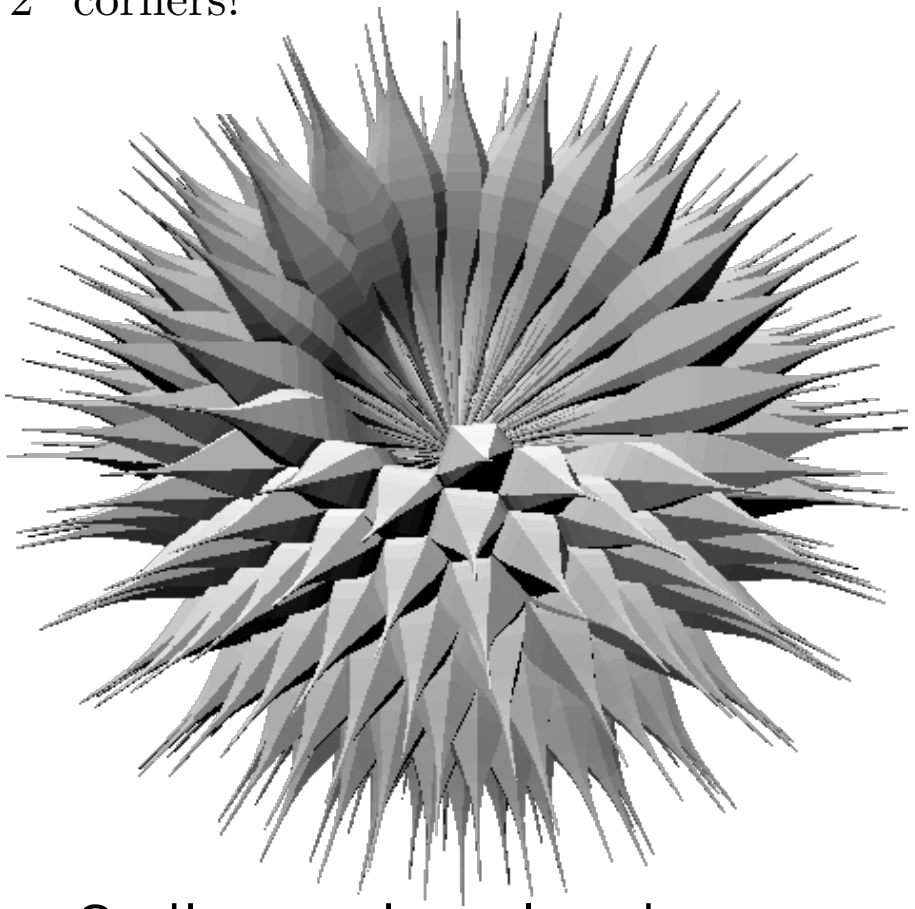
$$\lim_{d\to\infty} \frac{\text{volume sphere of radius r}}{\text{volume } [0,r]^d} = 0$$



: nearly all points are in the 2^d corners!

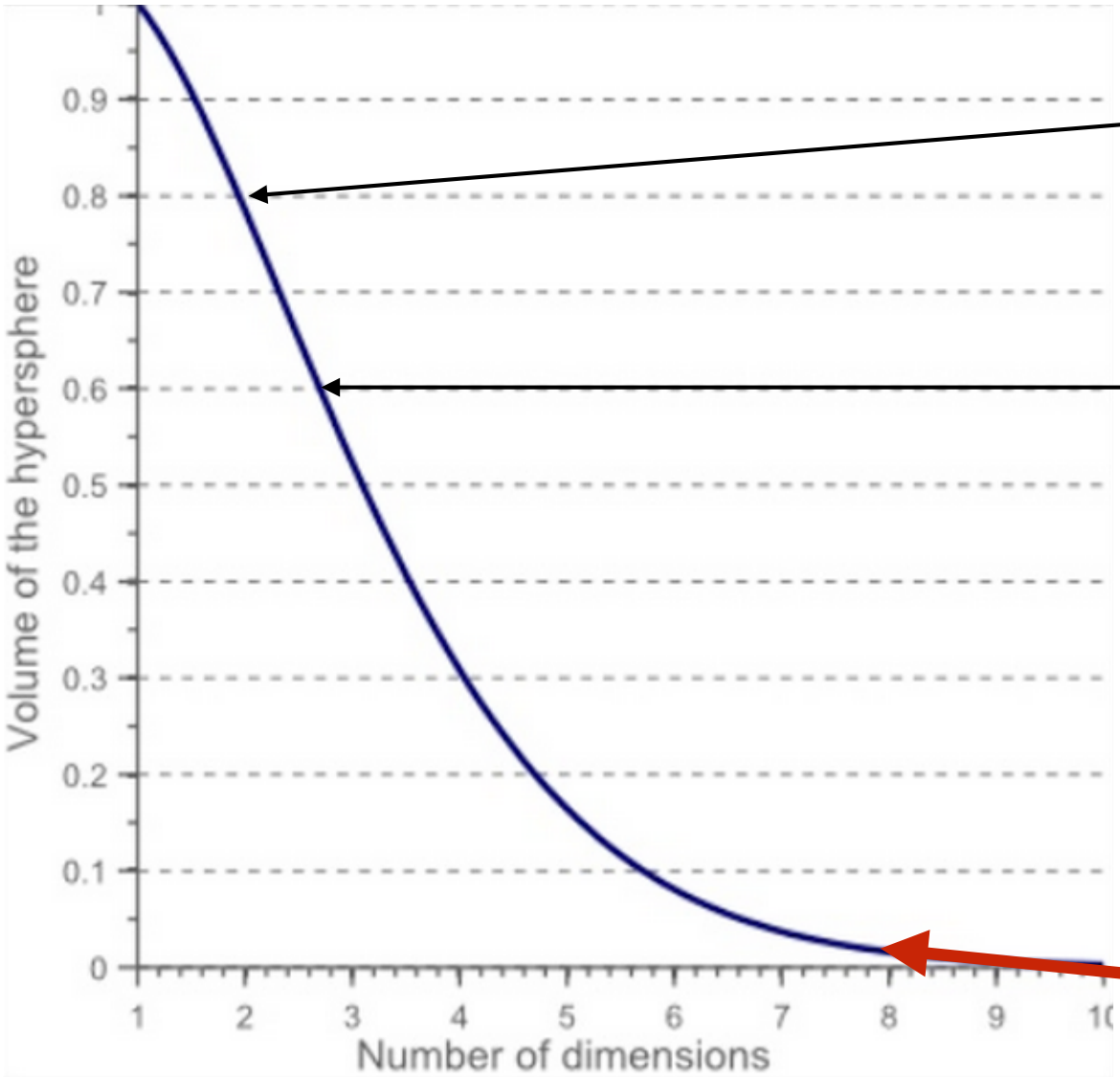


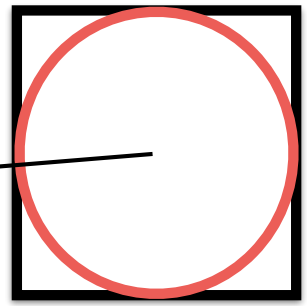
6 dimensional cube projection

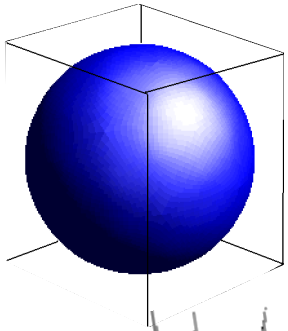


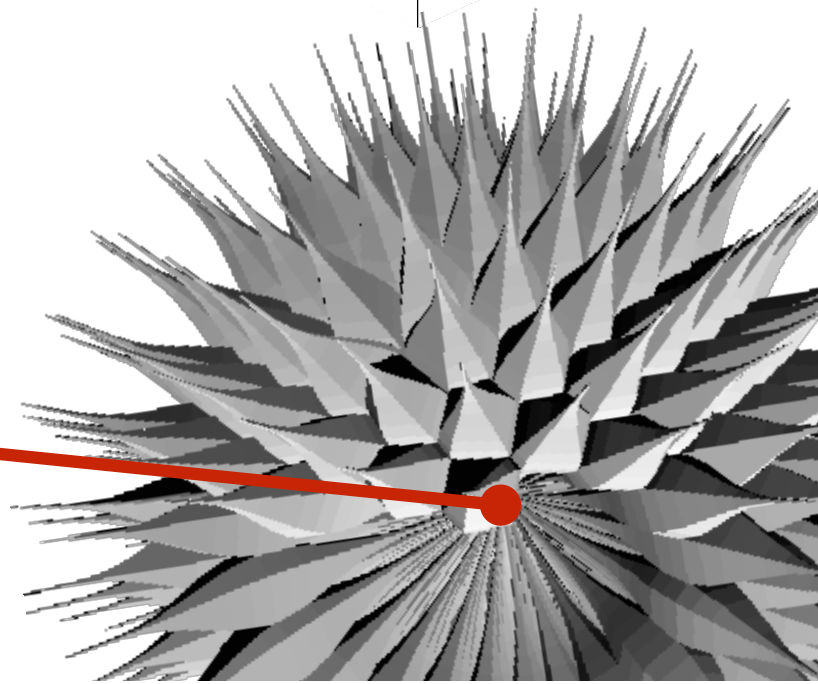
8 dimensional cube projection

Volume of the hypersphere



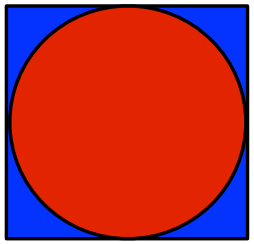




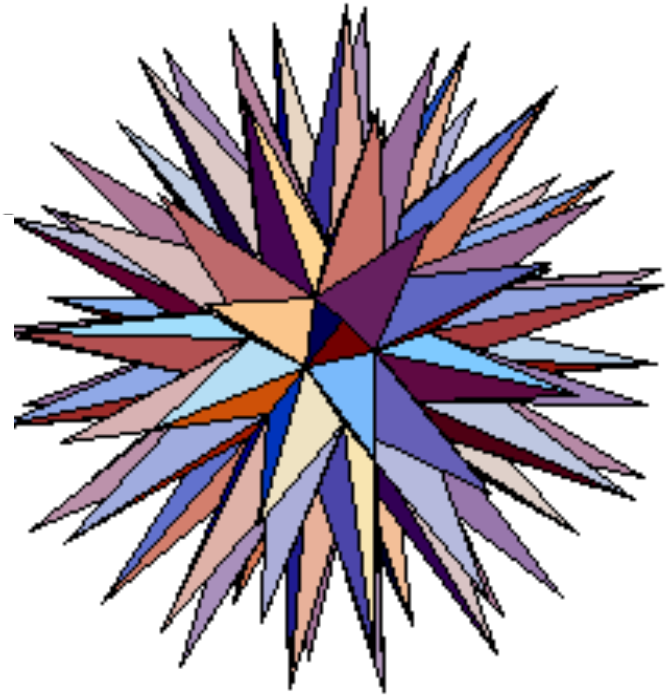


In High Dimensional Space: Rule # 1

- **Center v.s. Corners :**
  - **Corners win!**

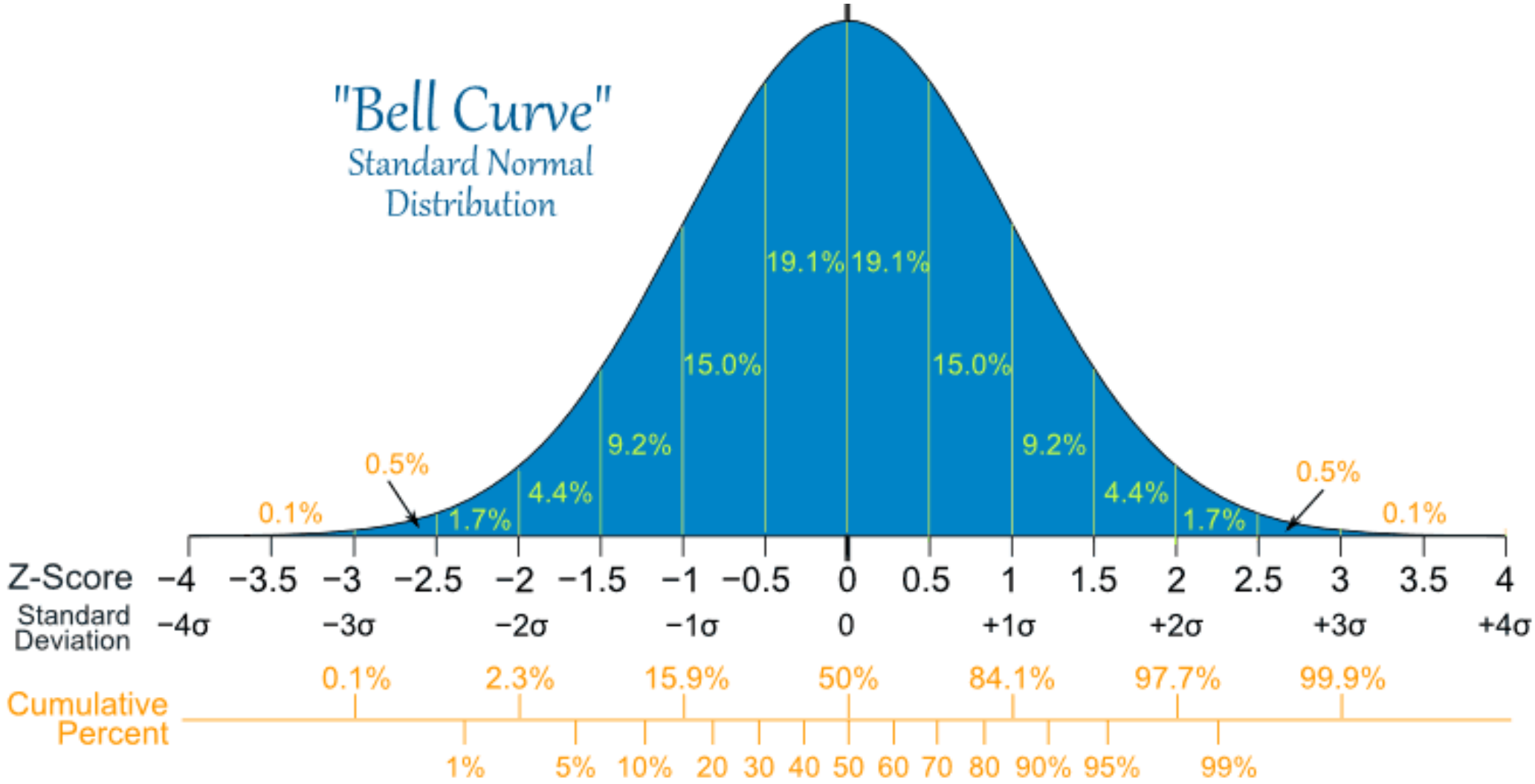


: nearly all points are in the 2^d corners!



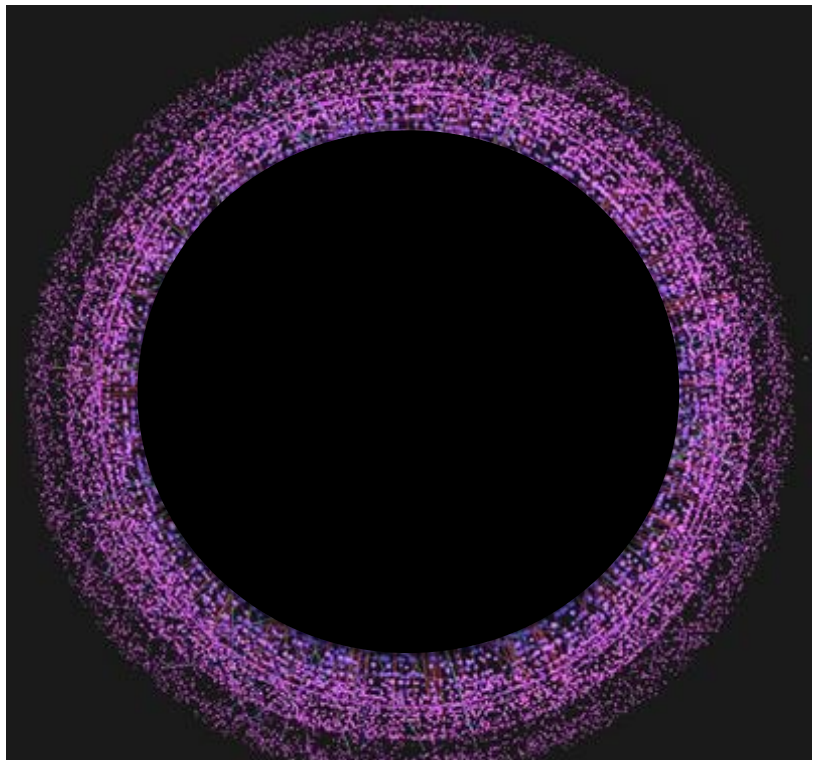
Question #2:

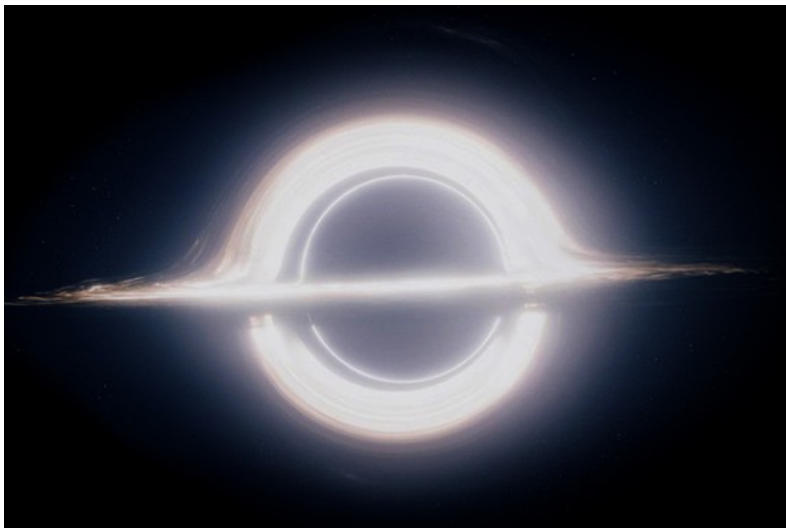
Gaussian Distribution (Center v.s. Surface)



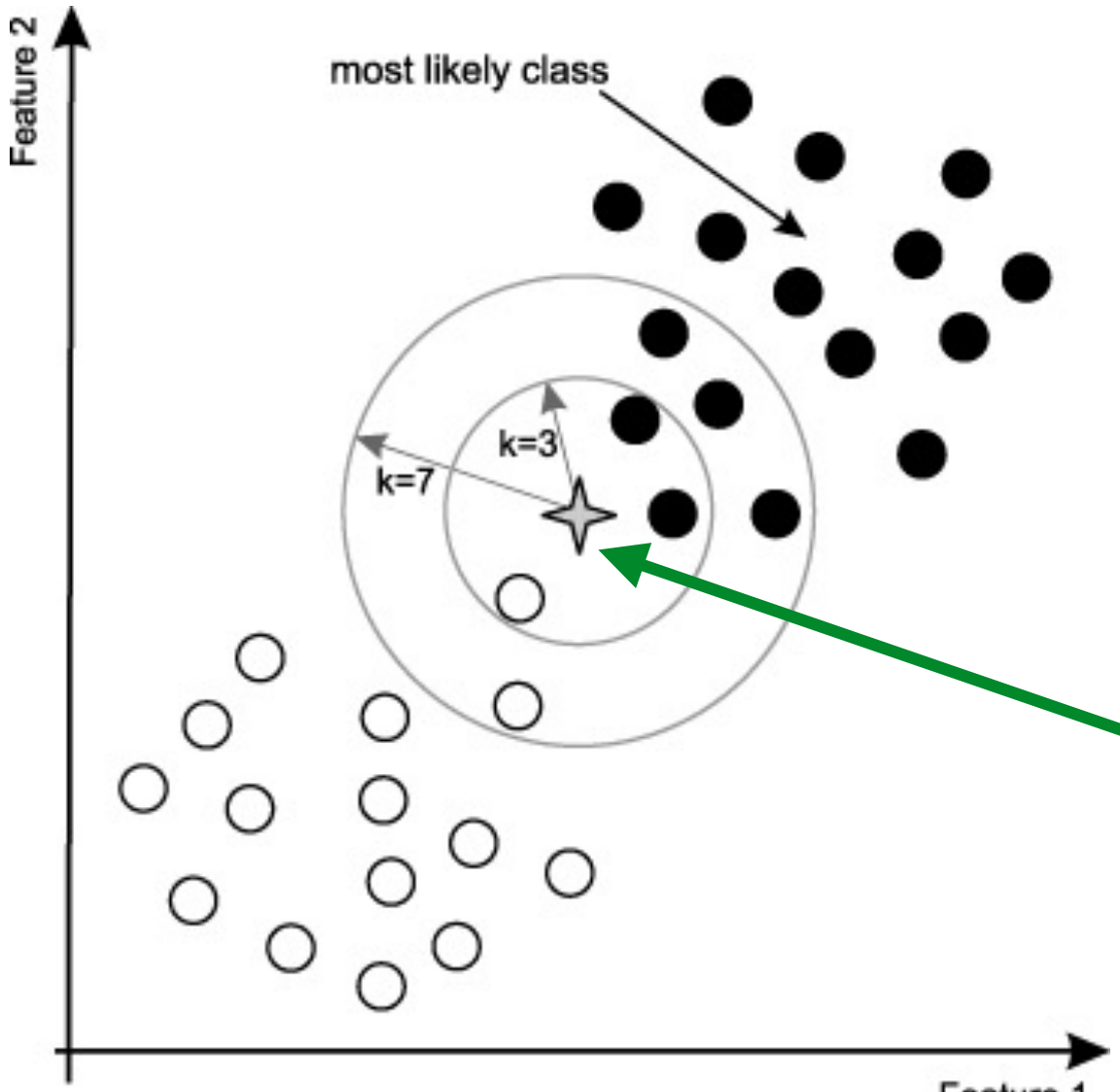
In High Dimensional Space: Rule # 2

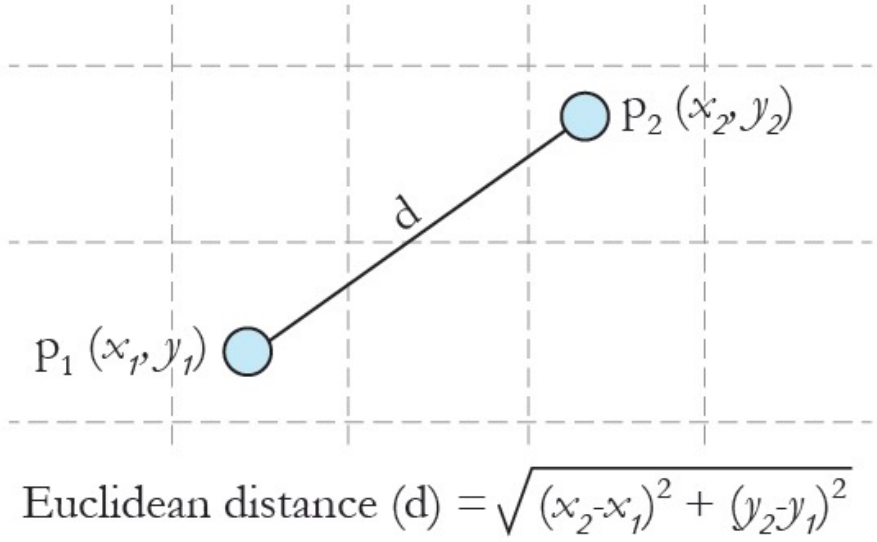
- **All data points are on the surface**
  - By chance, data will never reach the inner space of the ball.

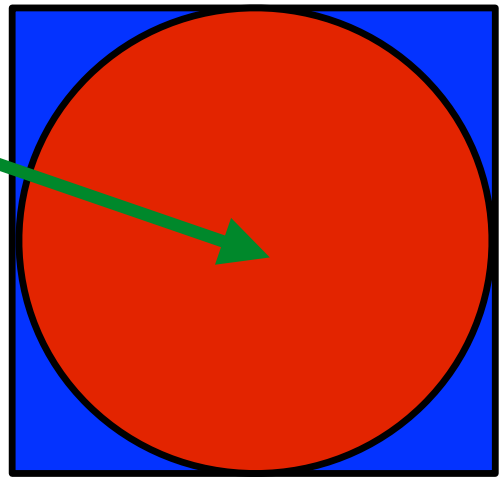




Question #3: Distance

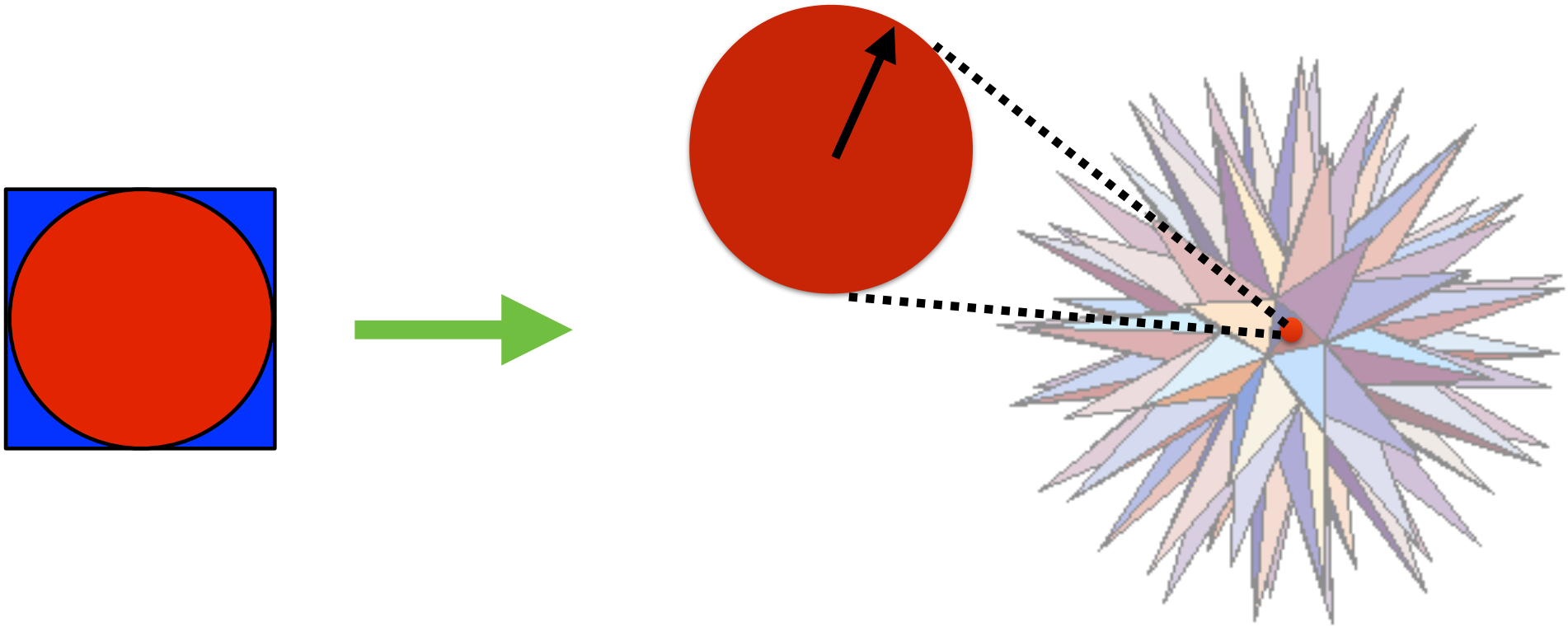






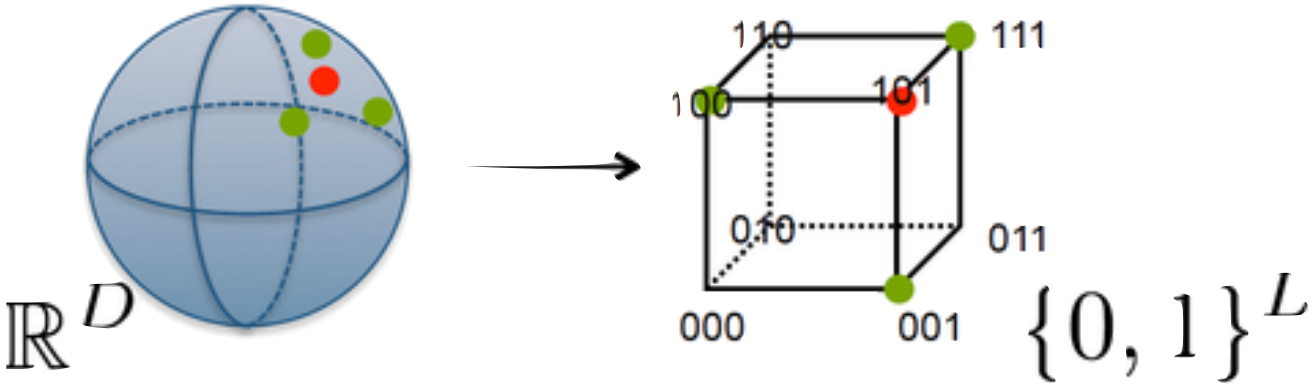
In High Dimensional Space: Rule # 3

- **Don't Use Euclidean Distance**

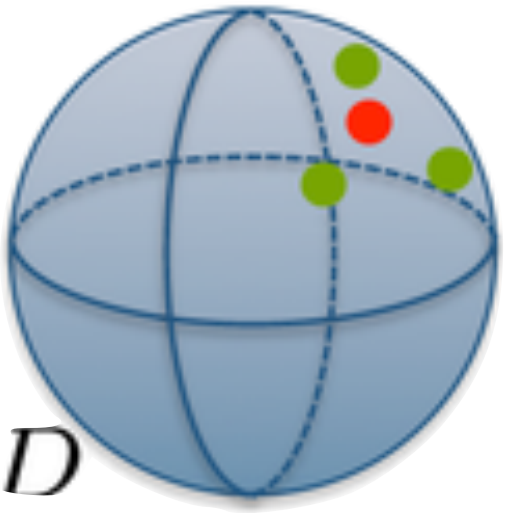


all your "neighbors" are far away

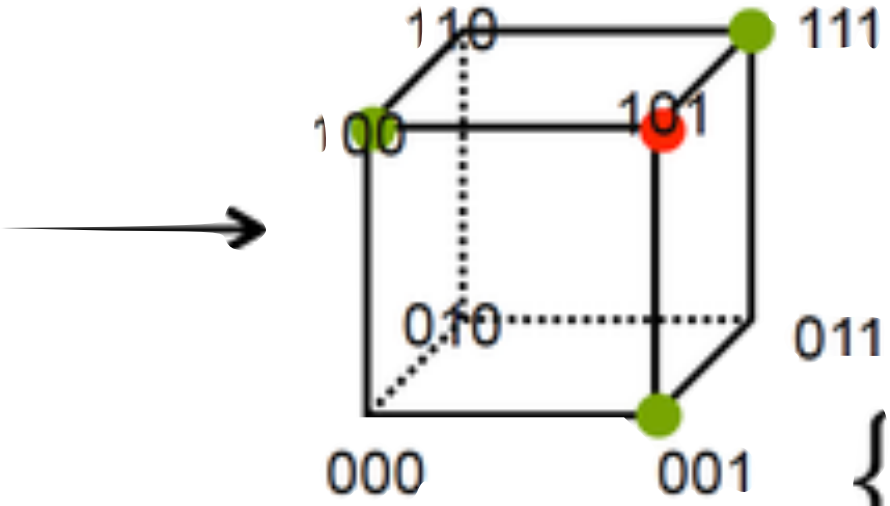
All "neighbors" are many light-years away



All "neighbors" are "equally" far away

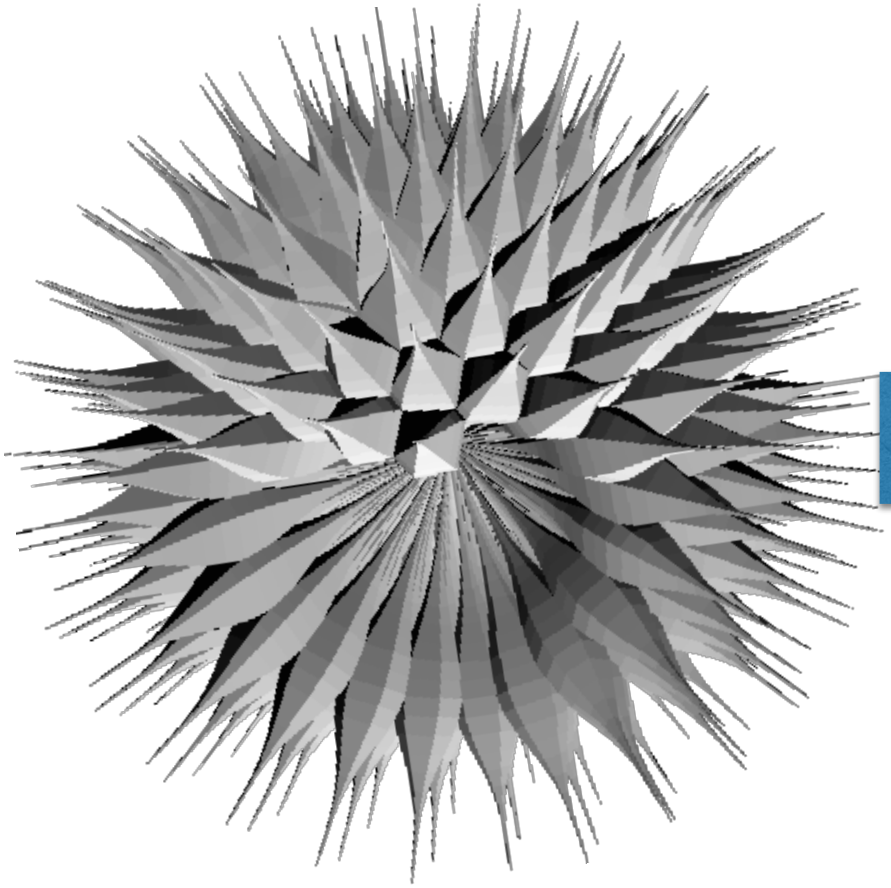


$\mathbb{R}^D$

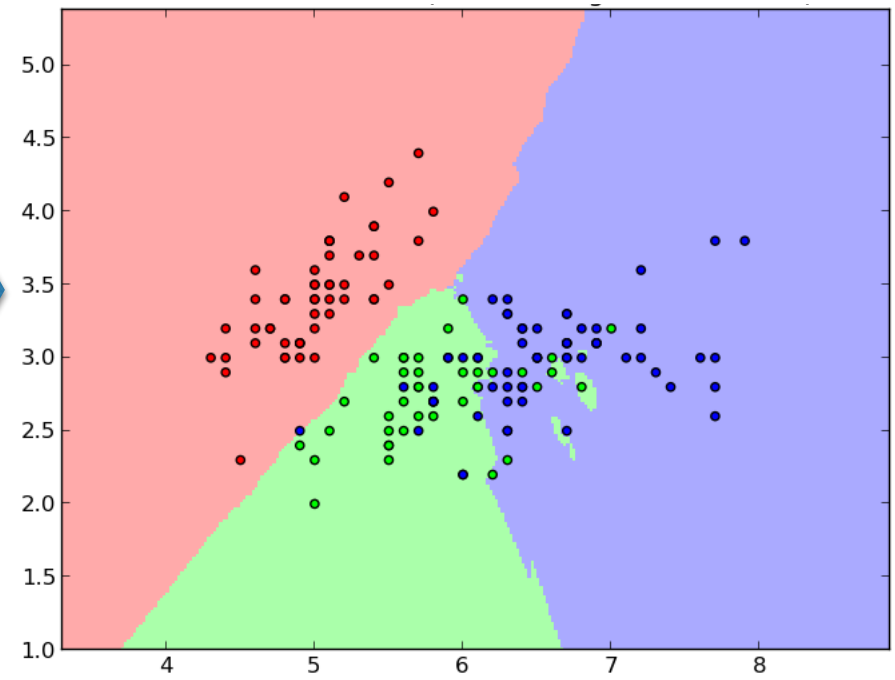


$\{0,1\}^L$

How to reduce dimensionality?



high dimensional



low dimensional

Image -> Vector

p - # pixels

p is large

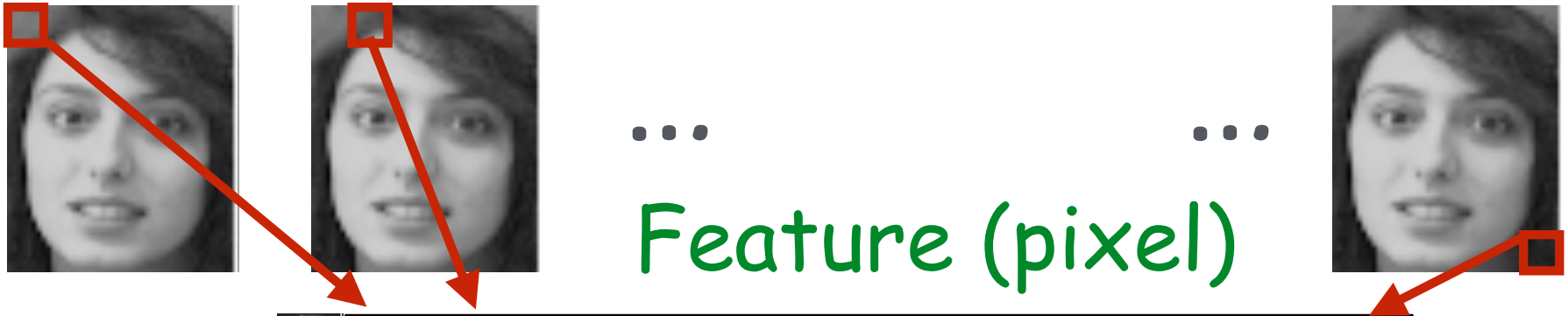
pixel matrix

1	3	4	3	8	3	5	7	9	7	3	4
3	3	5	7	7	0	4	1	2	1	9	7
7	0	4	1	1	4	3	7	8	6	2	7
1	4	3	7	9	7	3	2	7	0	4	1
7	7	3	2	2	1	9	8	1	4	3	7
2	1	9	8	8	6	2	0	7	7	3	2
8	6	2	0	0	4	1	1	4	1	9	8
0	2	1	4	1	3	7	9	7	6	2	0
3	5	3	3	7	3	2	2	1	2	1	3
1	7	2	3	2	2	1	2	3	5	3	1

1 3 4 3 8 3 5 7 9 7 3 4 3 3 5 7 7 0 4 1 2 1 9 7 ... 1 7 2 3 2 2 1 2 3 5 3 1

p by 1 vector

Matrix X (n by p) p is large



Feature (pixel)

Data (image)

	1	3	4	3	8	3	5	7	9	7	3	4
	3	3	5	7	7	0	4	1	2	1	9	7
	7	0	4	1	1	4	3	7	8	6	2	7
	1	4	3	7	9	7	3	2	7	0	4	1
	7	7	3	2	2	1	9	8	1	4	3	7
	2	1	9	8	8	6	2	0	7	7	3	2
	8	6	2	0	0	4	1	1	4	1	9	8
	0	2	1	4	1	3	7	9	7	6	2	0
	3	5	3	3	7	3	2	2	1	2	1	3
	1	7	2	3	2	2	1	2	3	5	3	1

n - # images

p - # pixels

mean vector μ (1 by p)

3.3	7.3	8.4	3.1	9.8	7.2	6.1	1.7	6.9	3.7	2.3	8.4

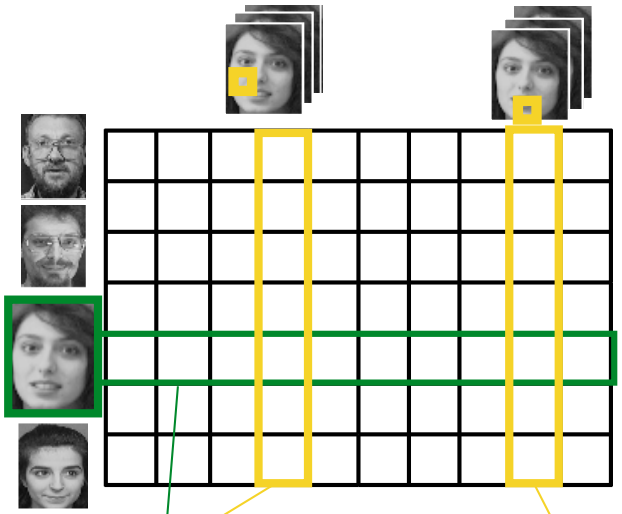
X

	1	3	4	3	8	3	5	7	9	7	3	4
	3	3	5	7	7	0	4	1	2	1	9	7
	7	0	4	1	1	4	3	7	8	6	2	7
	1	4	3	7	9	7	3	2	7	0	4	1
	7	7	3	2	2	1	9	8	1	4	3	7
	2	1	9	8	8	6	2	0	7	7	3	2
	8	6	2	0	0	4	1	1	4	1	9	8
	0	2	1	4	1	3	7	9	7	6	2	0
	3	5	3	3	7	3	2	2	1	2	1	3
	1	7	2	3	2	2	1	2	3	5	3	1

$$\mu = \frac{1}{n}1^T X$$

$\frac{1}{n}1^T$

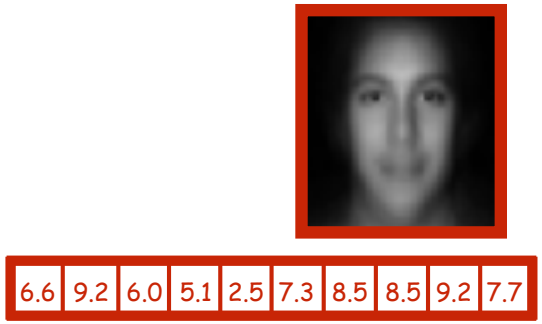
shape: (1 by n)



X

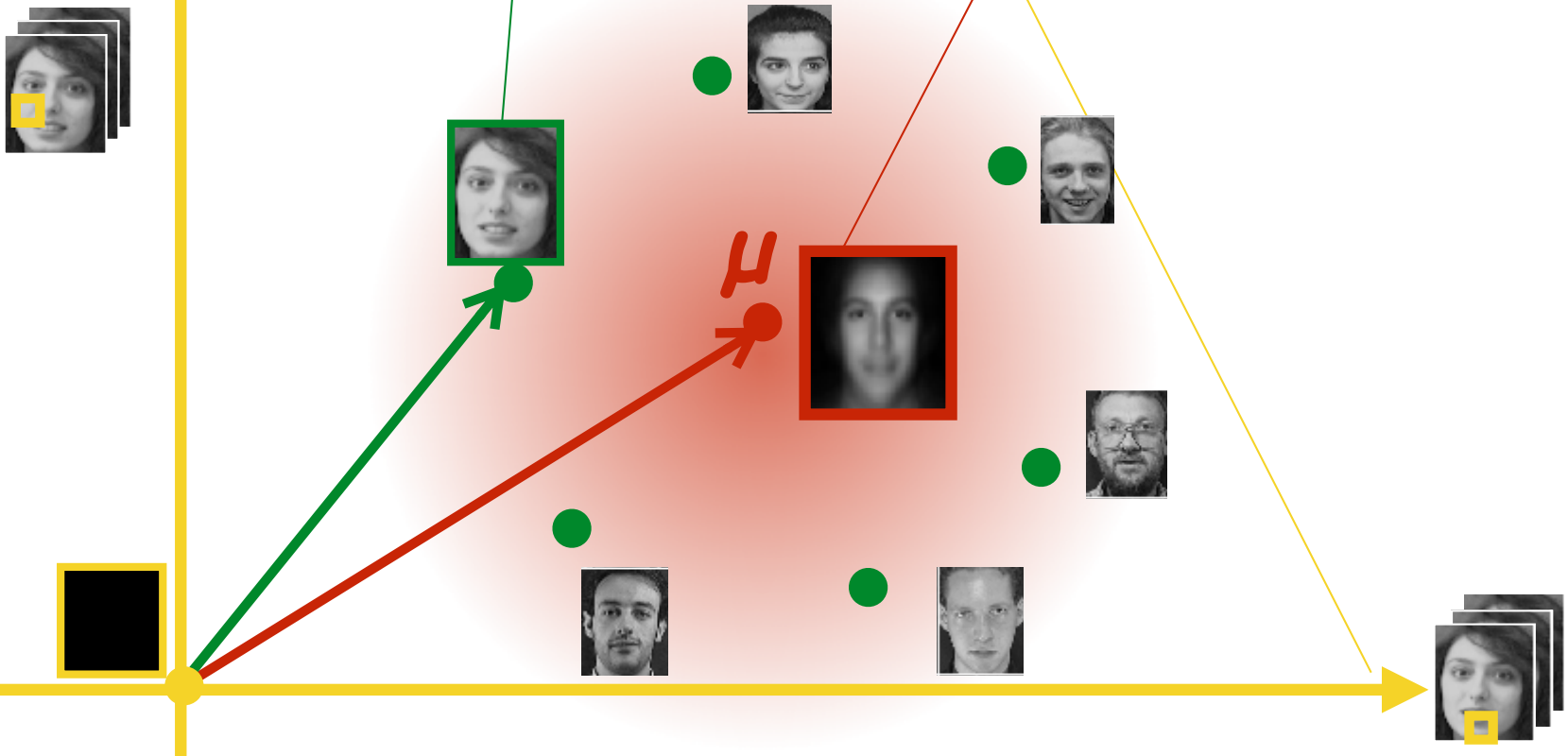
(n by p)

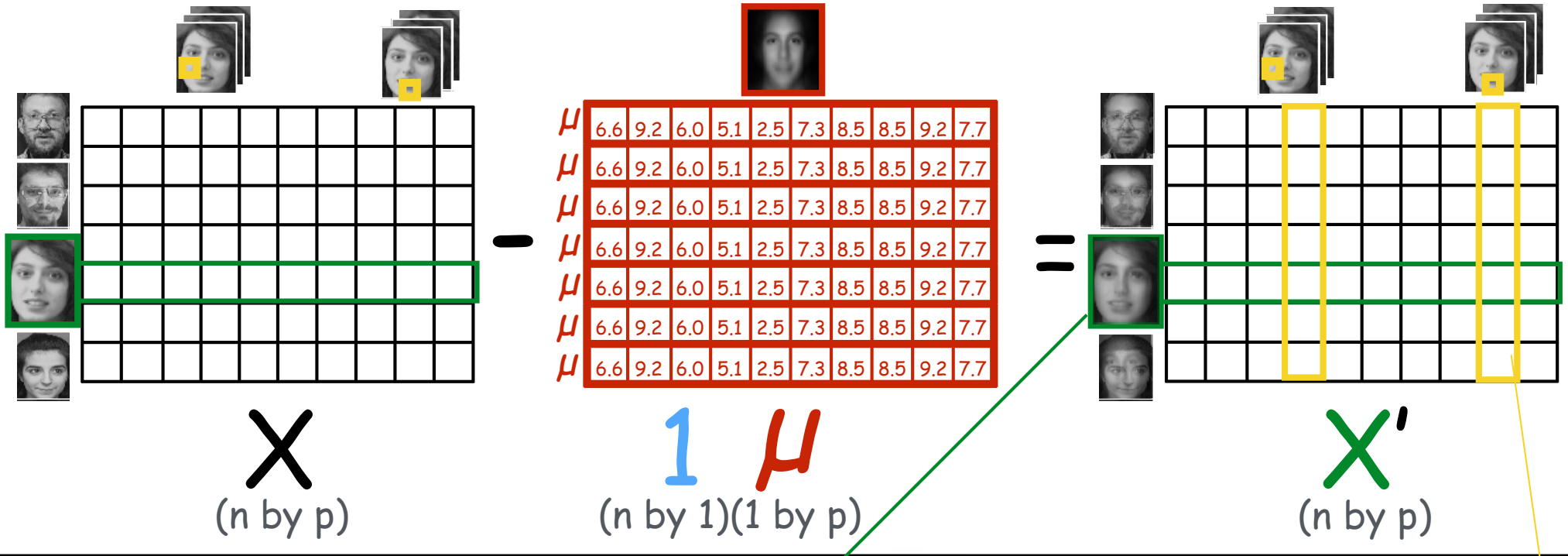
=



μ

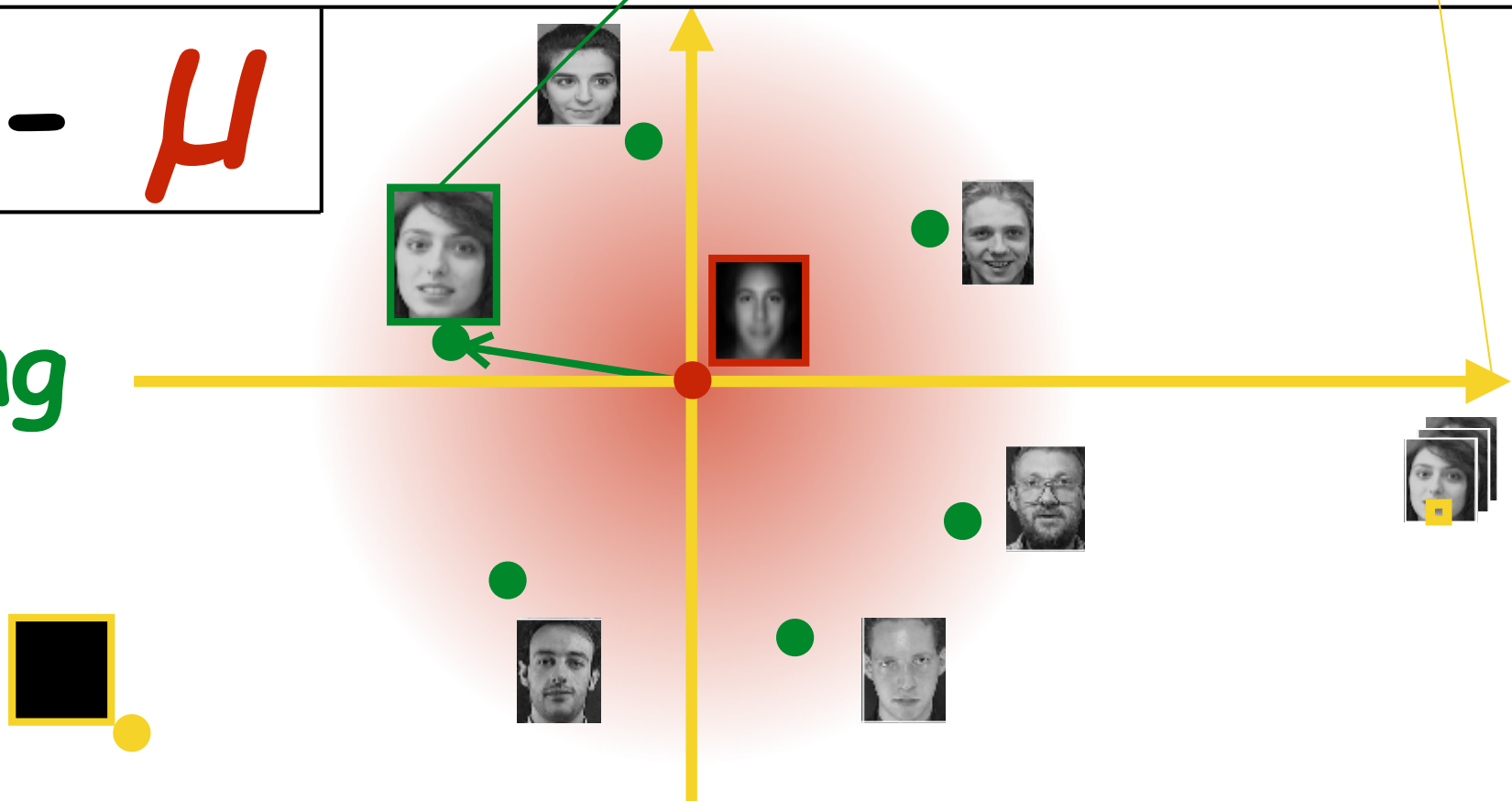
(1 by p)





$$X' = X - \mu$$

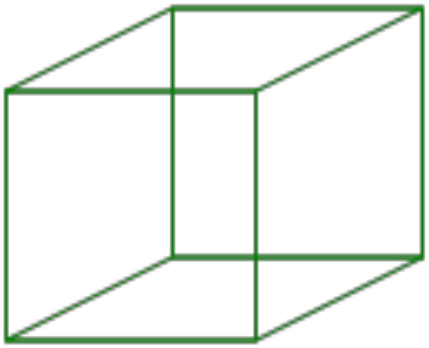
centering

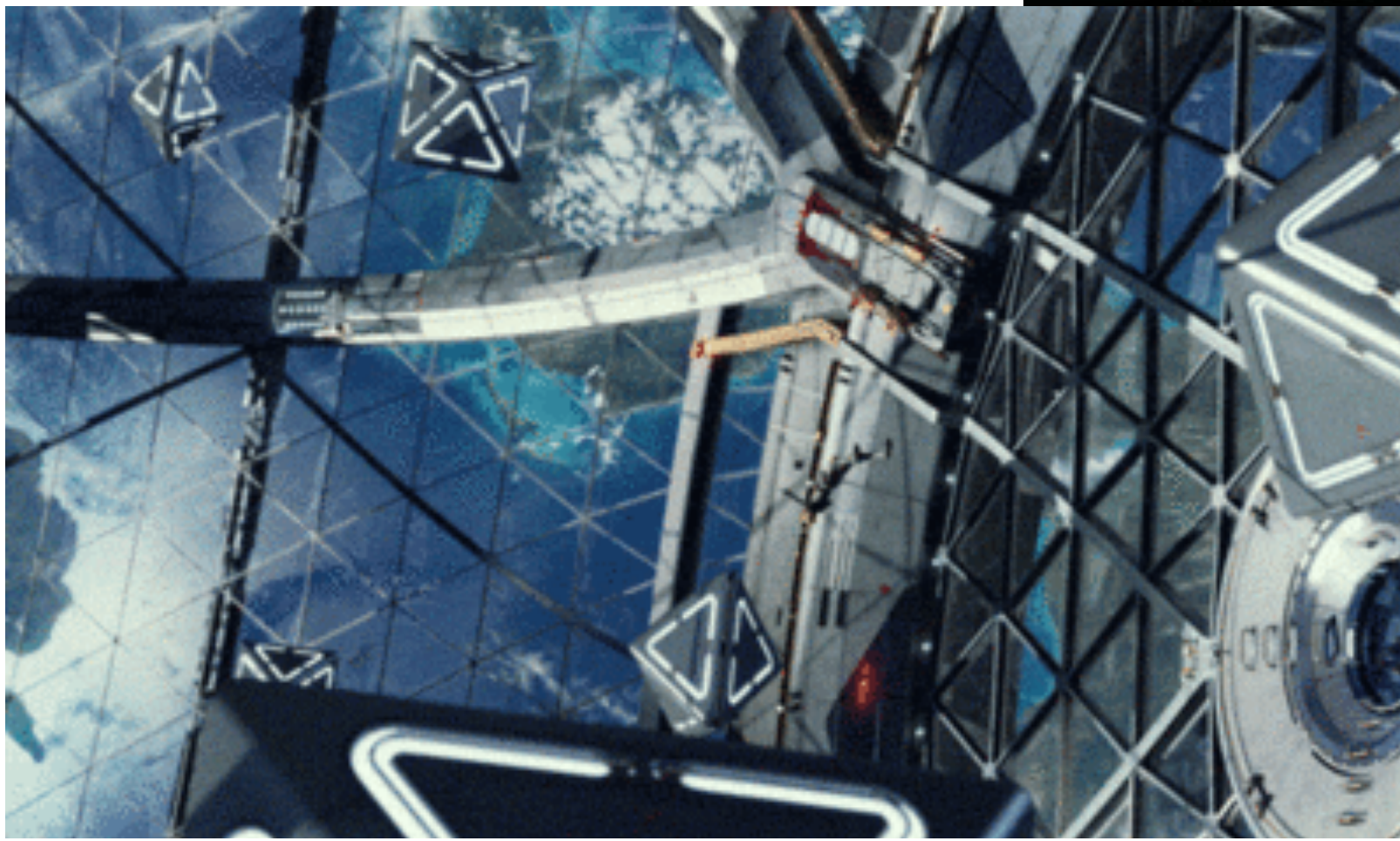


Direction?

Choose a direction as your new dimension

3 Dimensions



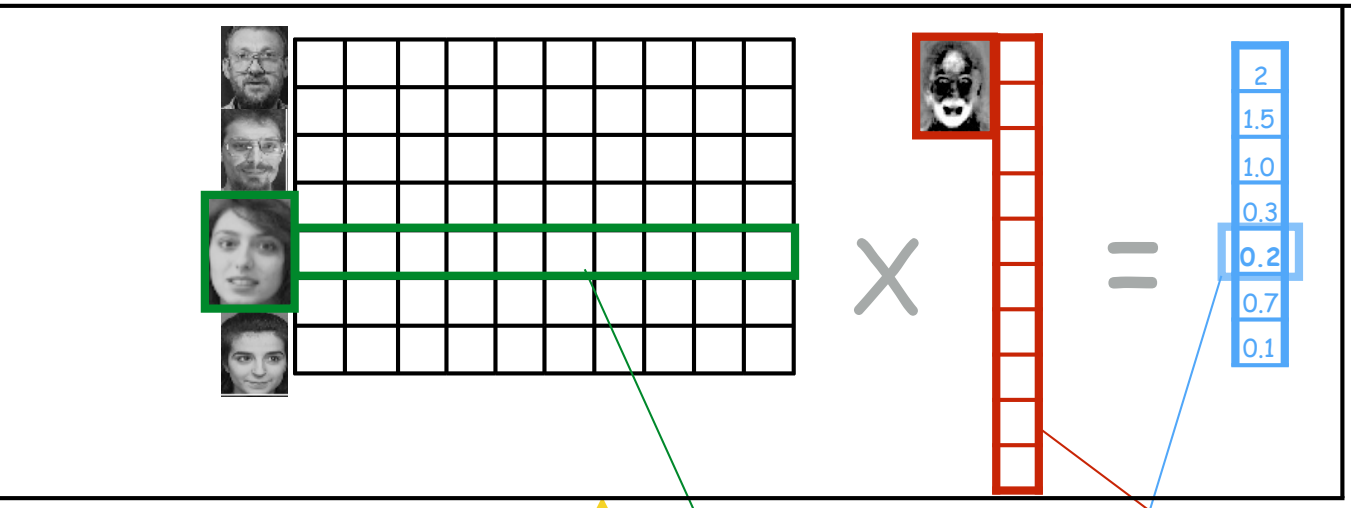


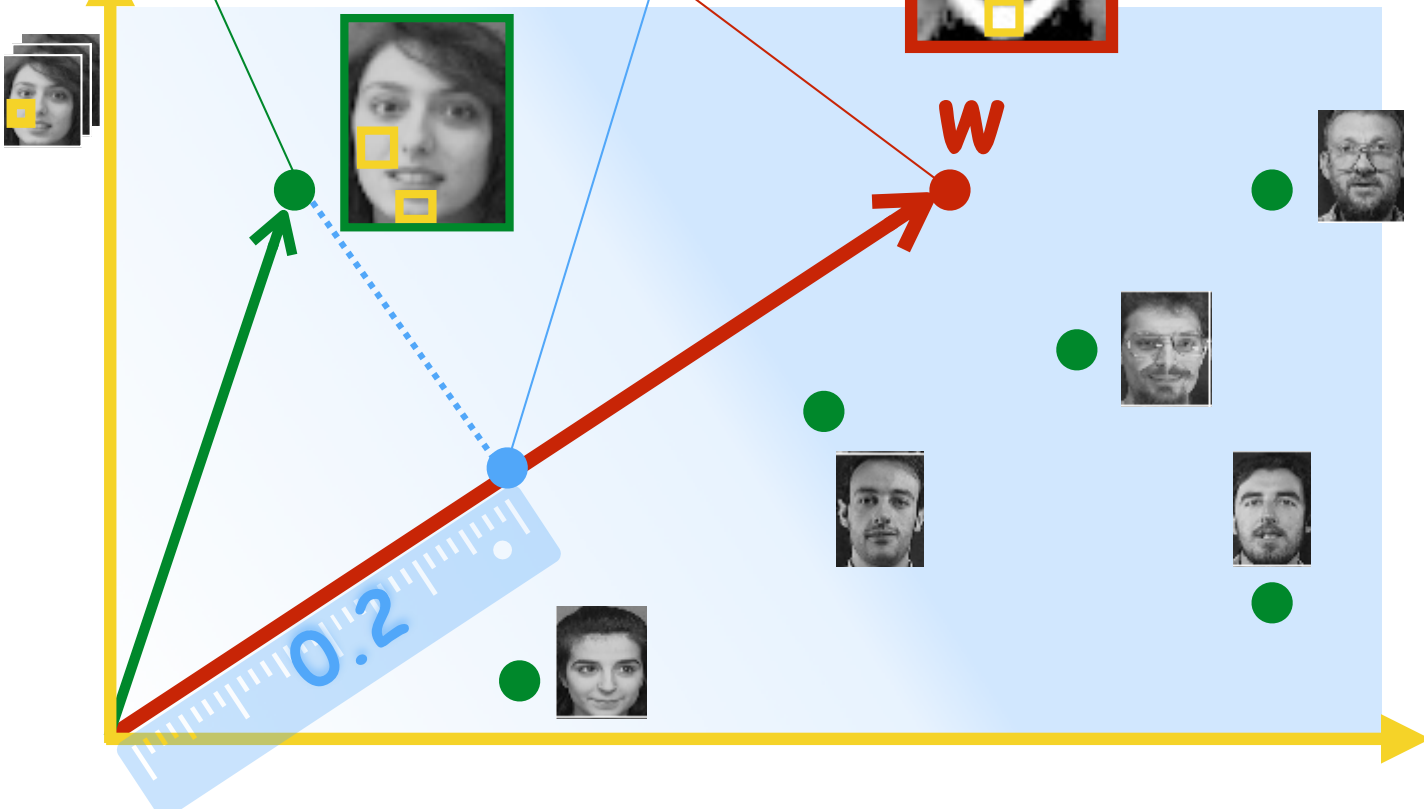
“the enemy’s gate is down”

X w = y

shape: (n by p) (p by 1) (n by 1)

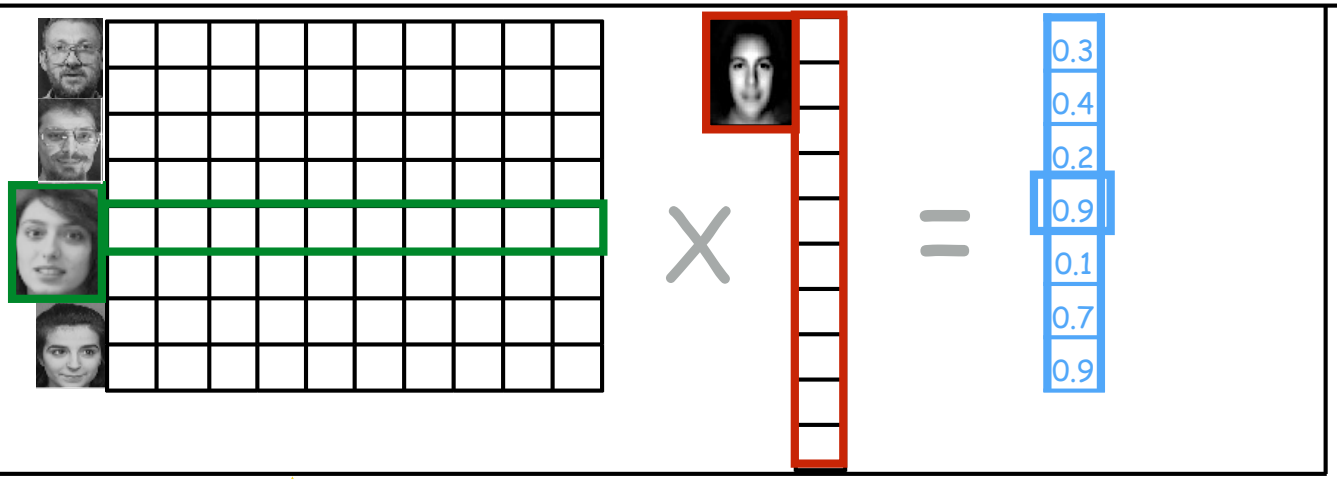
projection

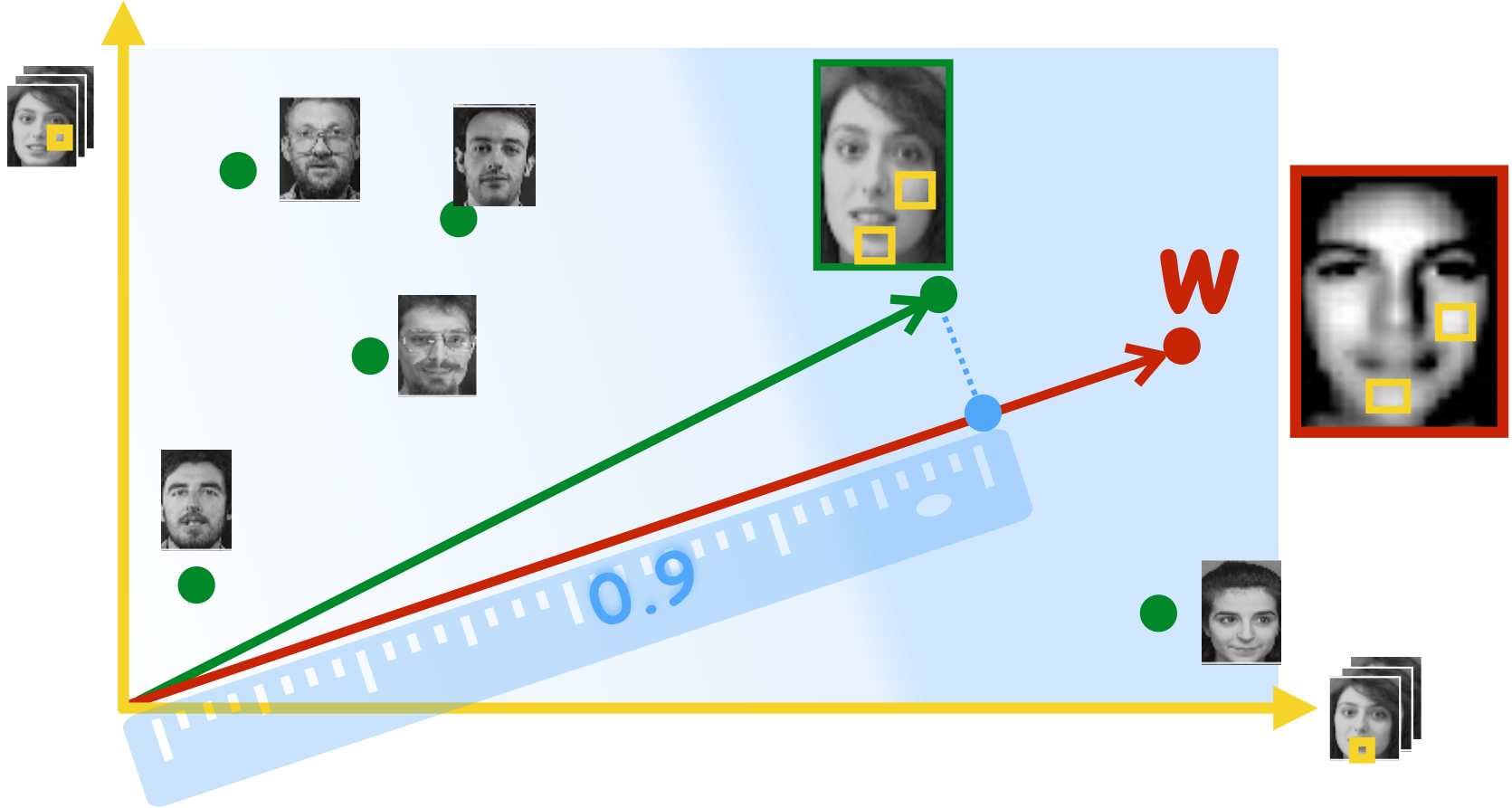




X w = y

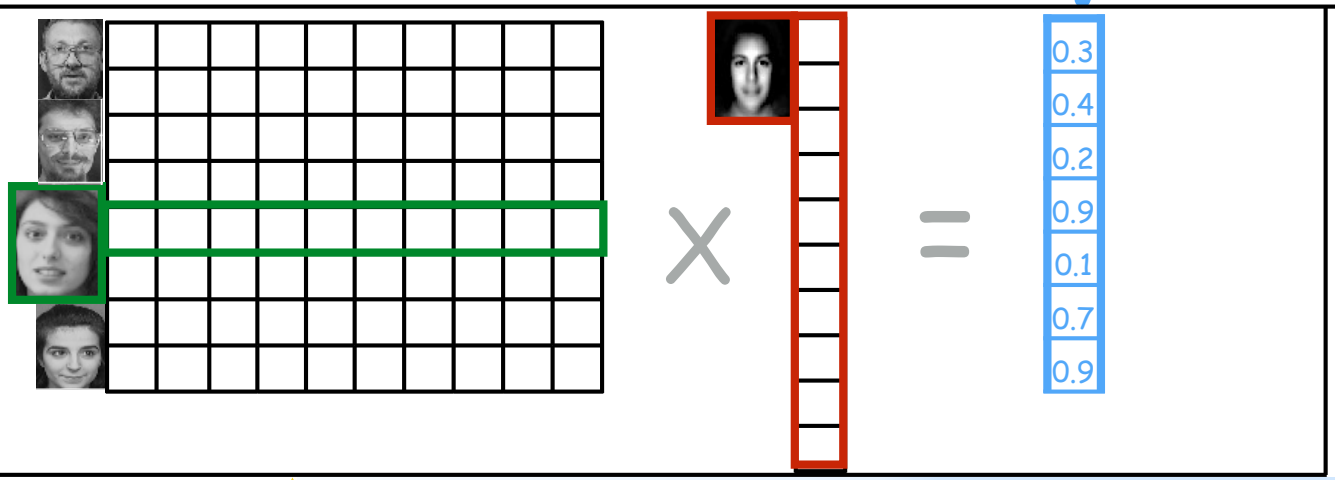
projection

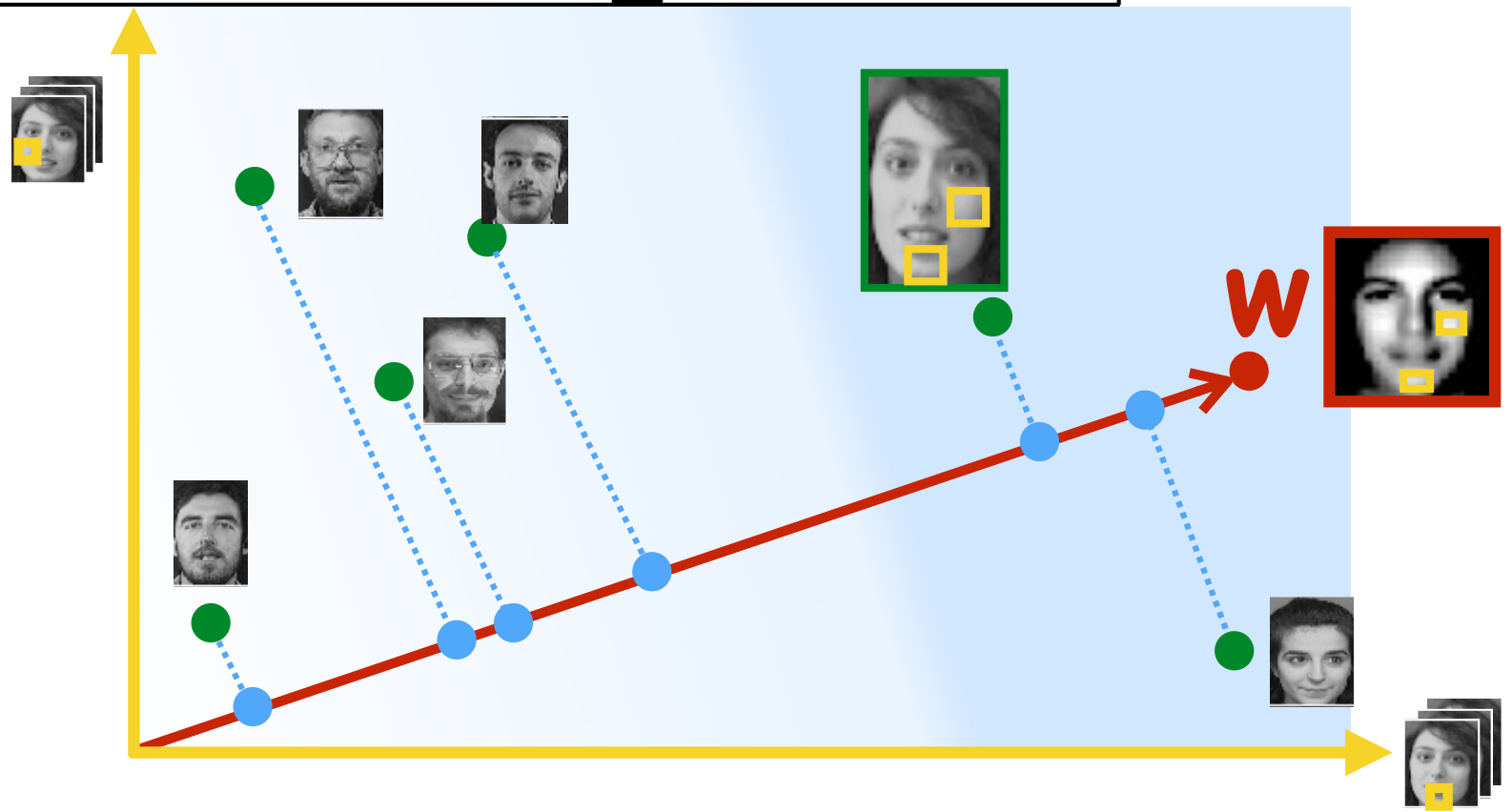




X w = y

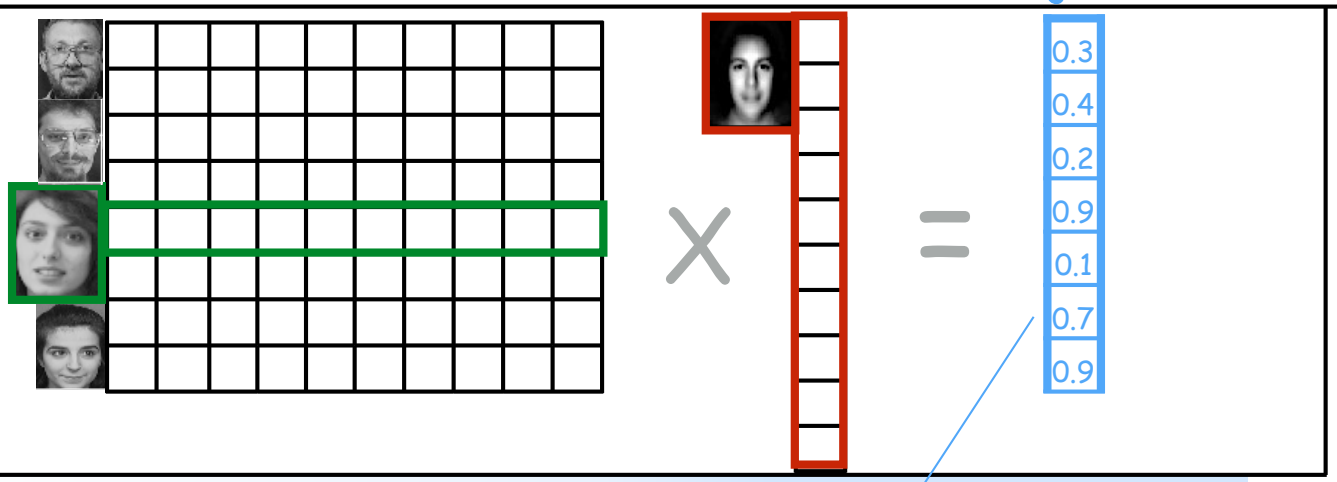
projection

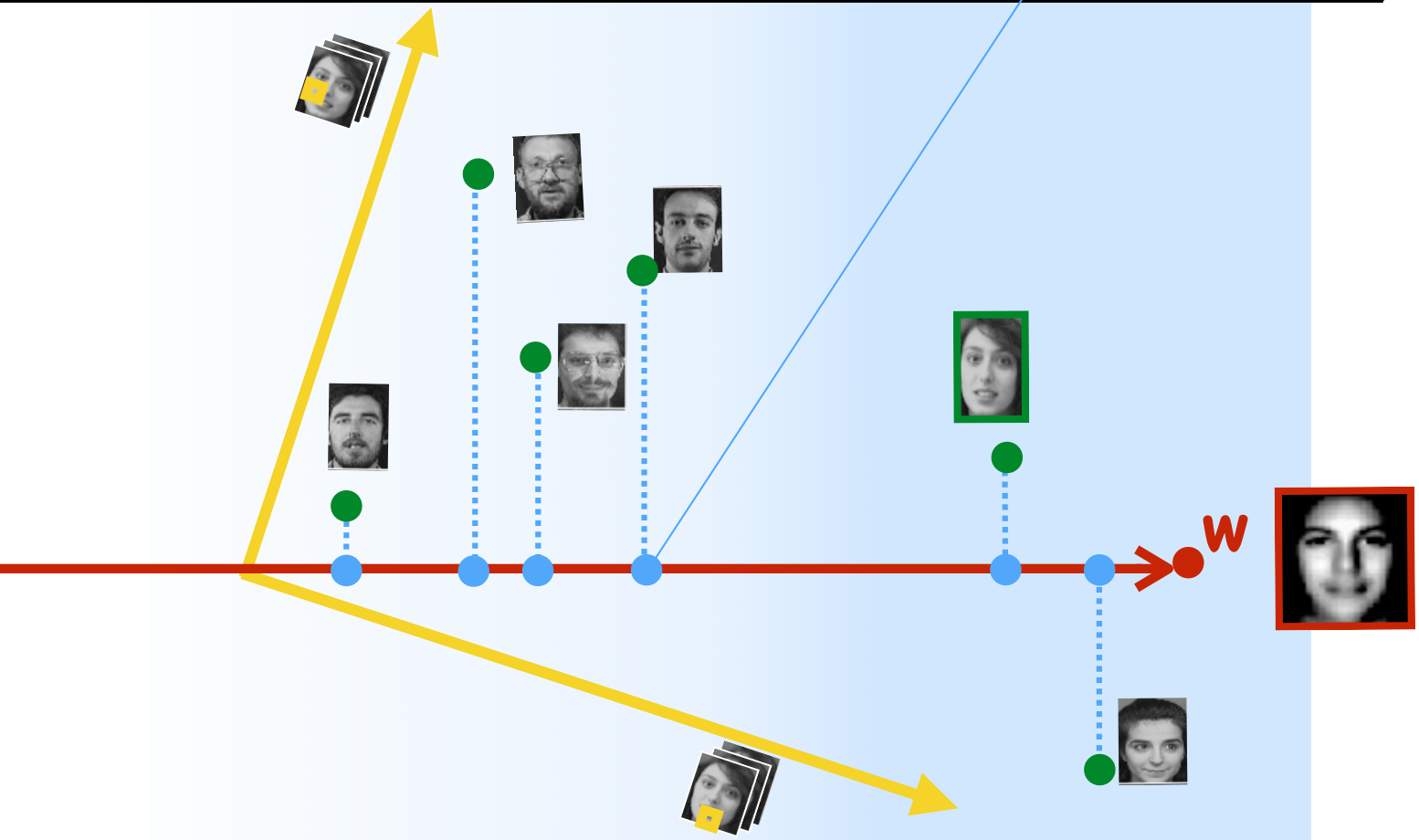




X w = y

projection

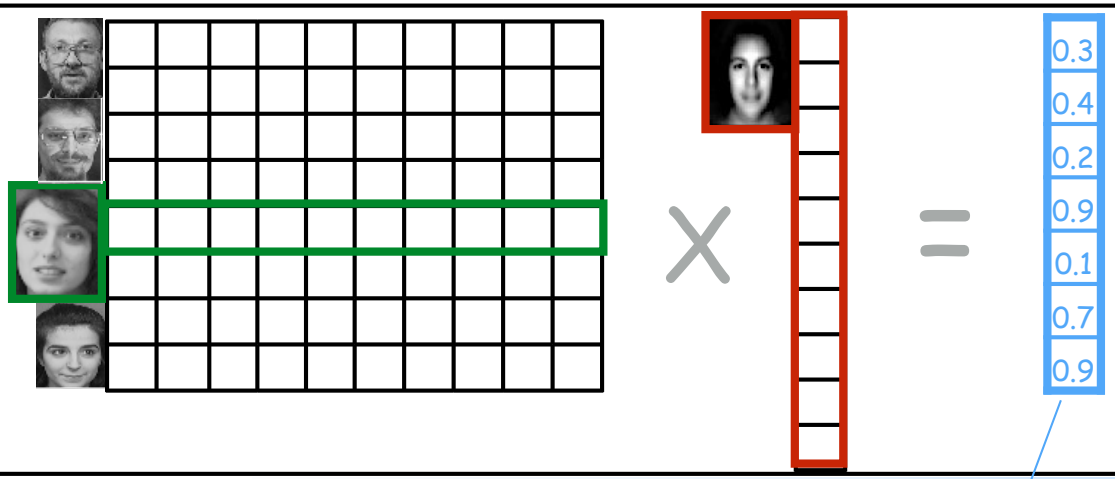




new axis

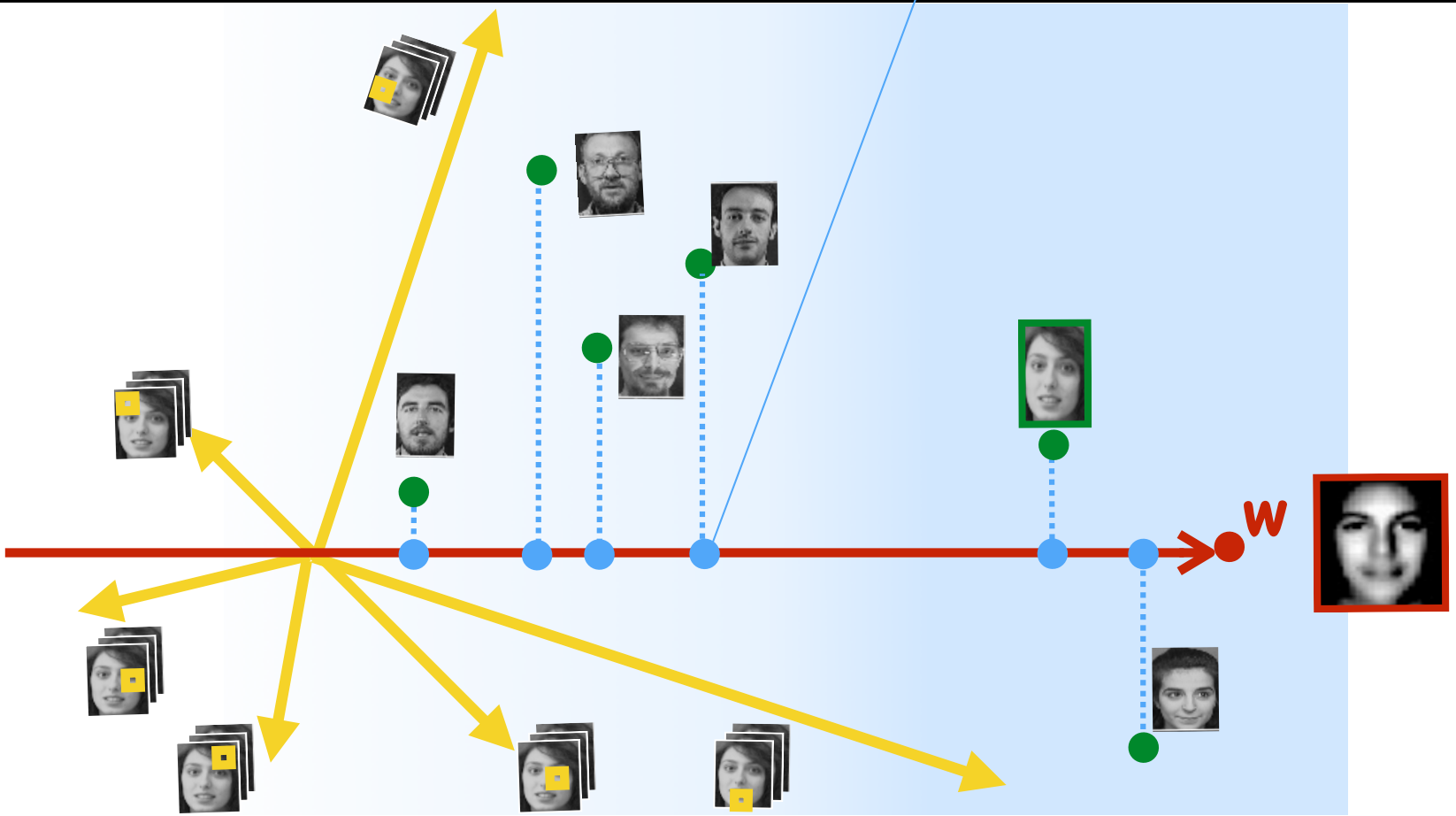
X w = y

projection



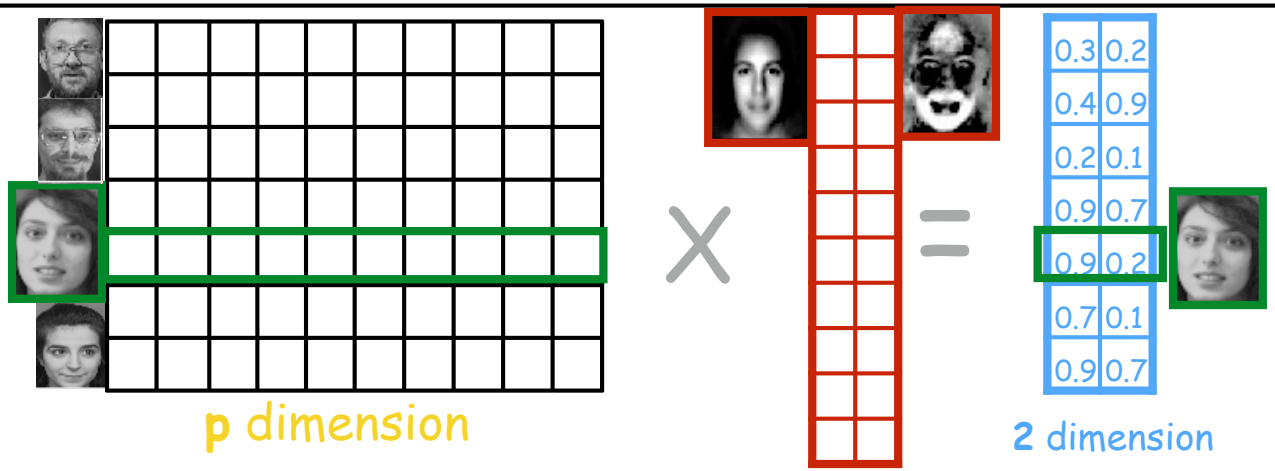
p dimension

1 dimension



shape: (n by p) X $W_{(p \text{ by } 2)}$ $=$ $Y_{(n \text{ by } 2)}$

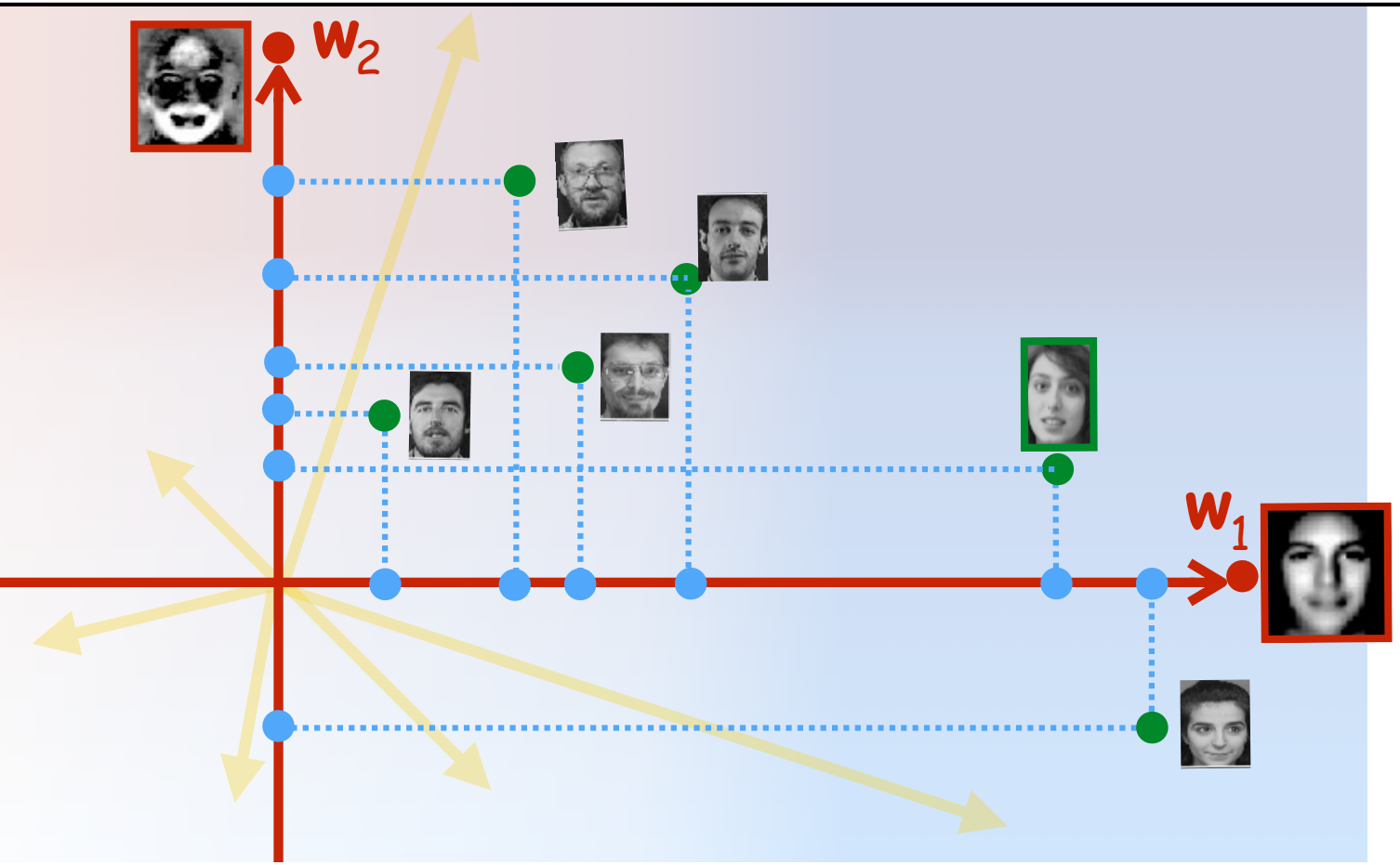
projection

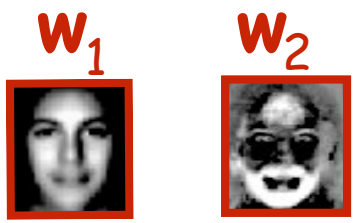


p dimension

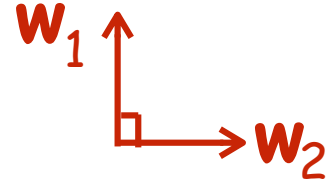
↓

2 dimension





1. orthogonal

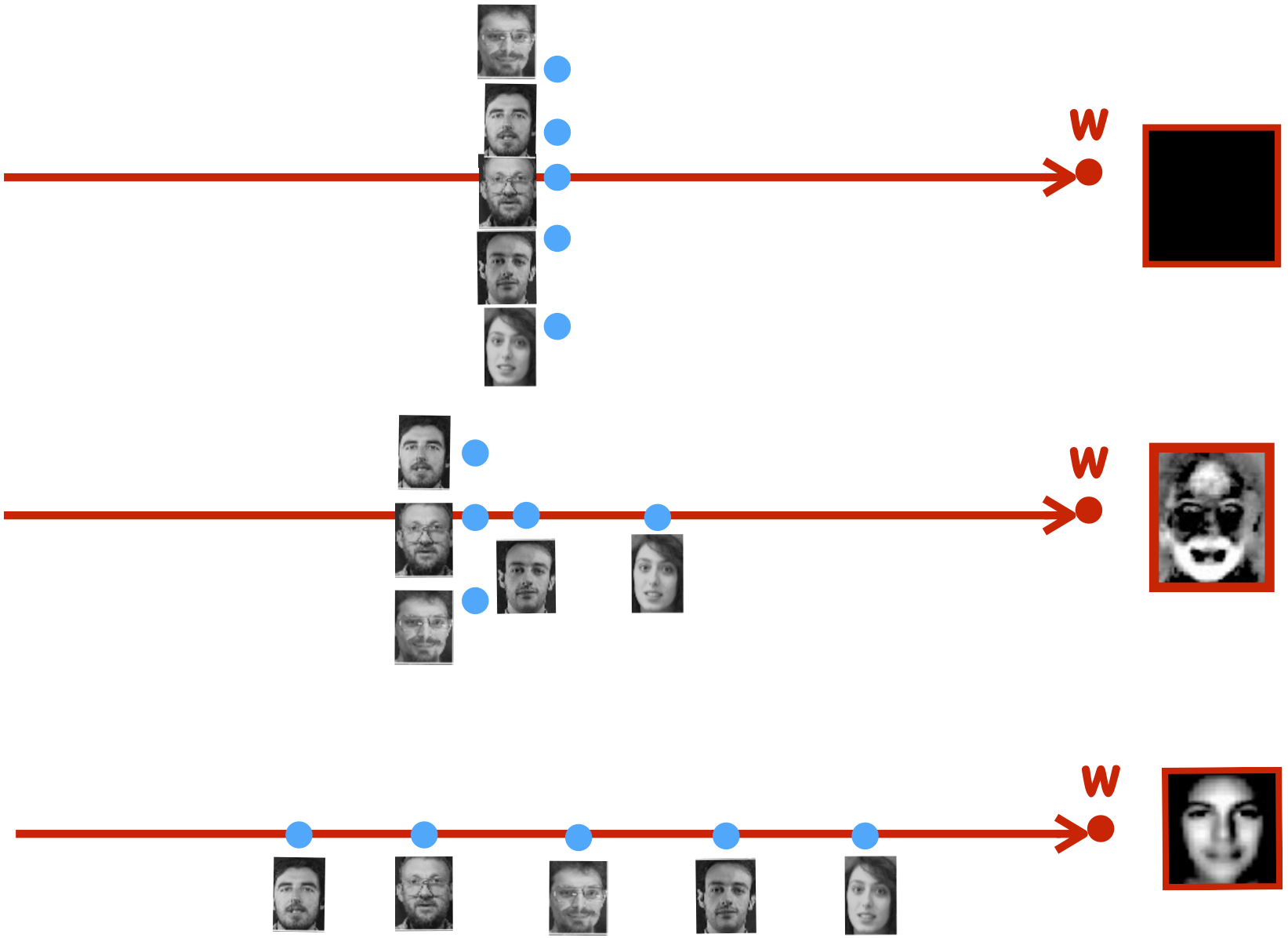


2. unit length

$|w_1| = 1$

$|w_2| = 1$

How to find a good projection?

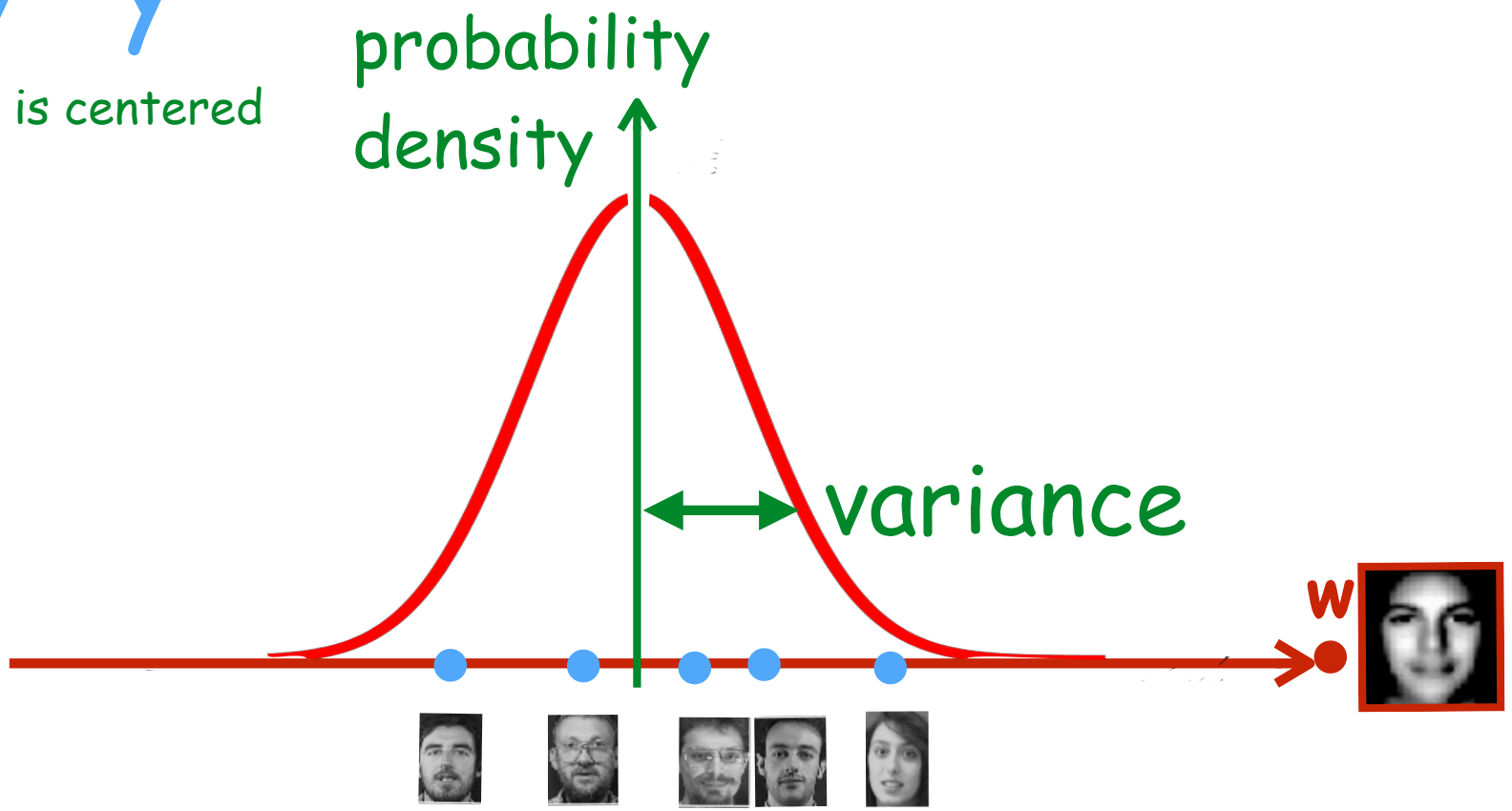


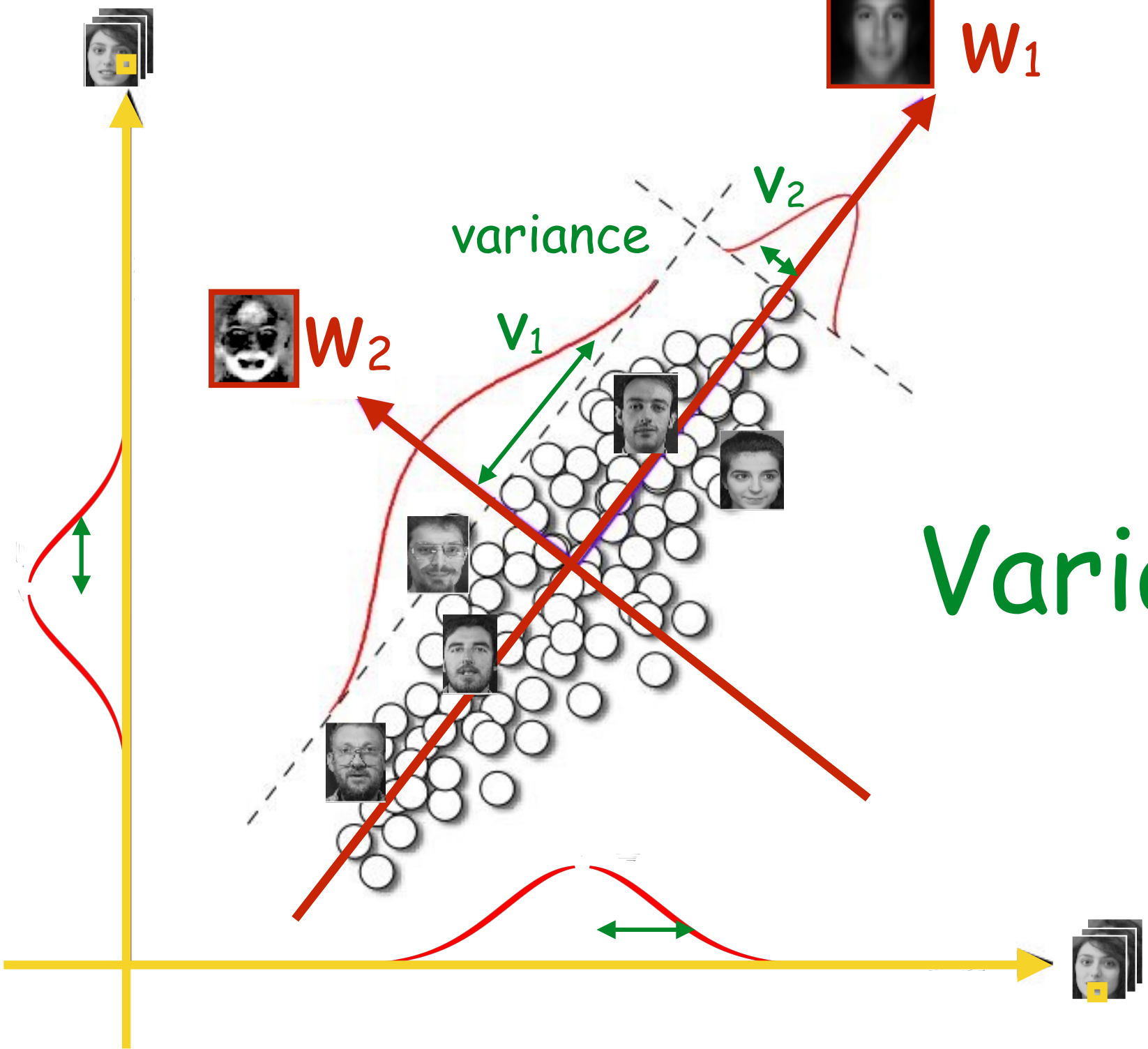
Variance

$$\begin{bmatrix} 0.3 & 0.4 & 0.4 & 0.9 & 0.1 & 0.7 & 0.9 \end{bmatrix} \times \begin{bmatrix} 0.3 \\ 0.4 \\ 0.2 \\ 0.9 \\ 0.1 \\ 0.7 \\ 0.9 \end{bmatrix} = \text{var}(y)$$

$$\frac{1}{n-1} y^T y$$

assuming y is centered




W1
W2
V2
V1
variance
Variance

Maximize Variance

assuming X and y are already centered

$$\max_{w} \text{var}(y)$$

$$= \max_{w} \frac{1}{n-1} y^T y$$

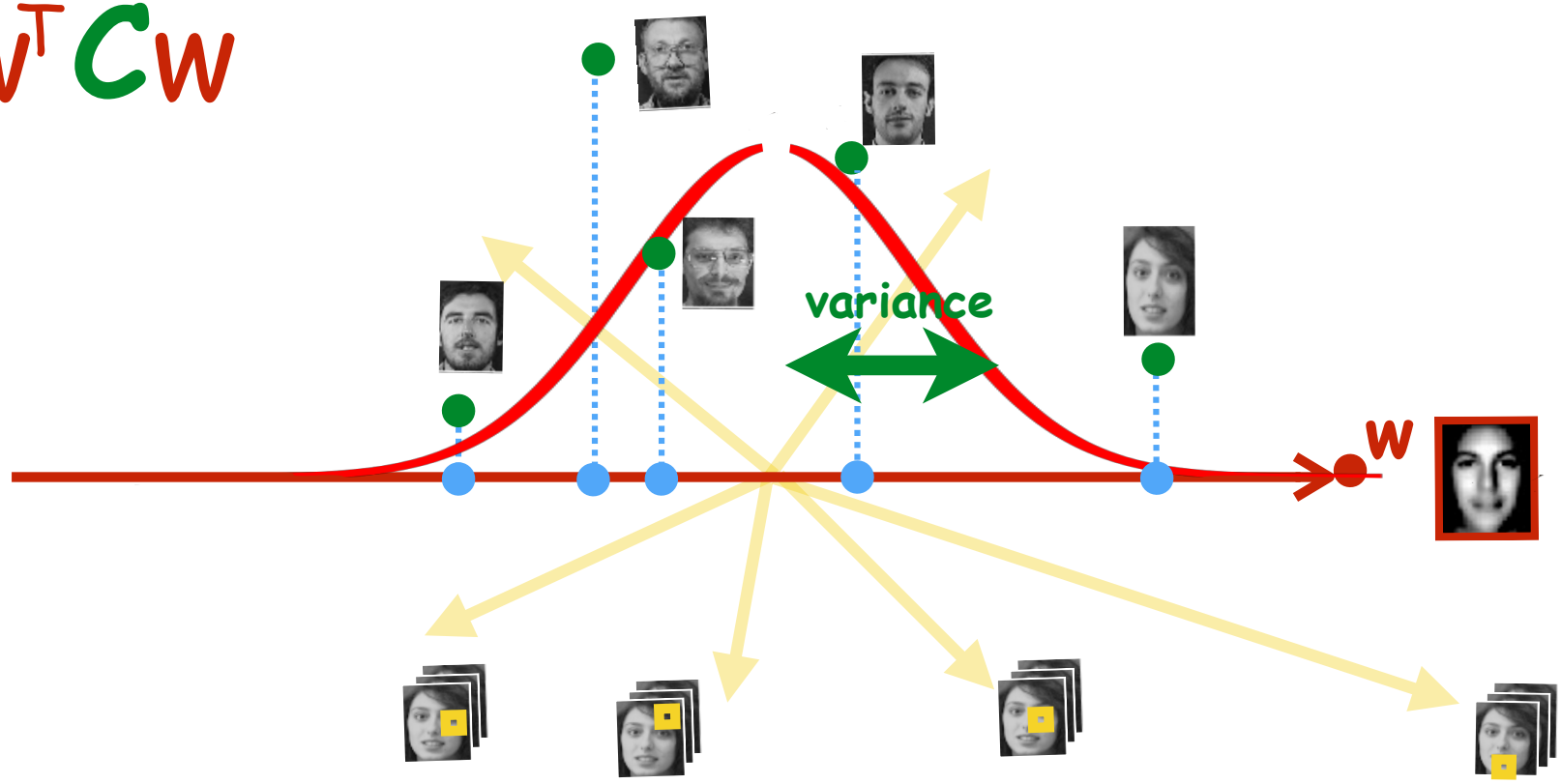
$$= \max_{w} \frac{1}{n-1} w^T X^T X w$$

$$= \max_{w} w^T C w$$

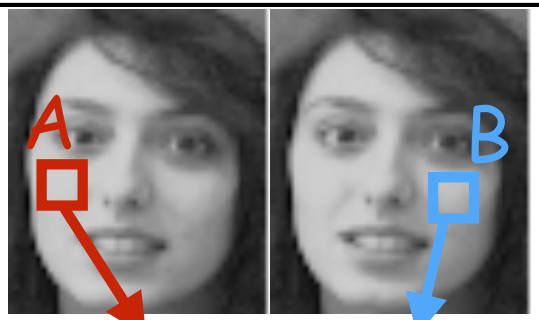
$$\text{var}(y) = \frac{1}{n-1} y^T y$$

$$X w = y$$

$$C = \frac{1}{n-1} X^T X$$



Covariance



X

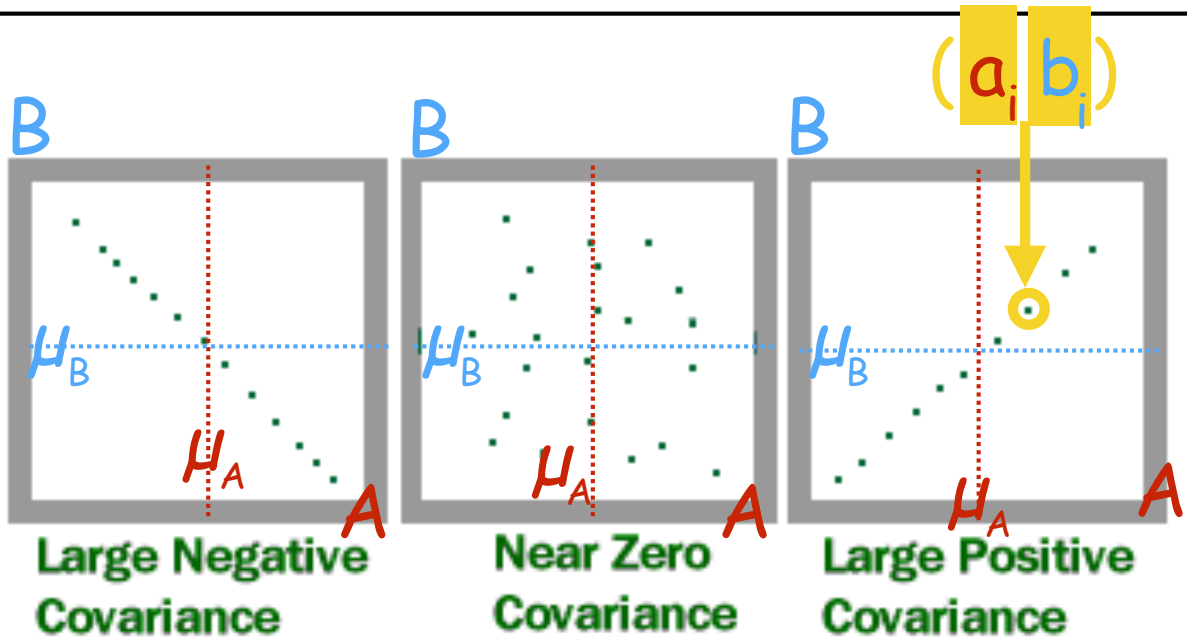
1	3	4	3	8	3	5	7	9	7	3	4
3	3	5	7	7	0	4	1	2	1	9	7
7	0	4	1	1	4	3	7	8	6	2	7
1	4			7	9	7	3		7	0	4	1
7	7				1	9		1	4	3	7
2	1	9	8	8	6	2	0	7	7	3	2
8	6	2	0	0	4	1	1	4	1	9	8
0	2	1	4	1	3	7	9	7	6	2	0
3	5	3	3	7	3	2	2	1	2	1	3
1	7	2	3	2	2	1	2	3	5	3	1

a_i b_i

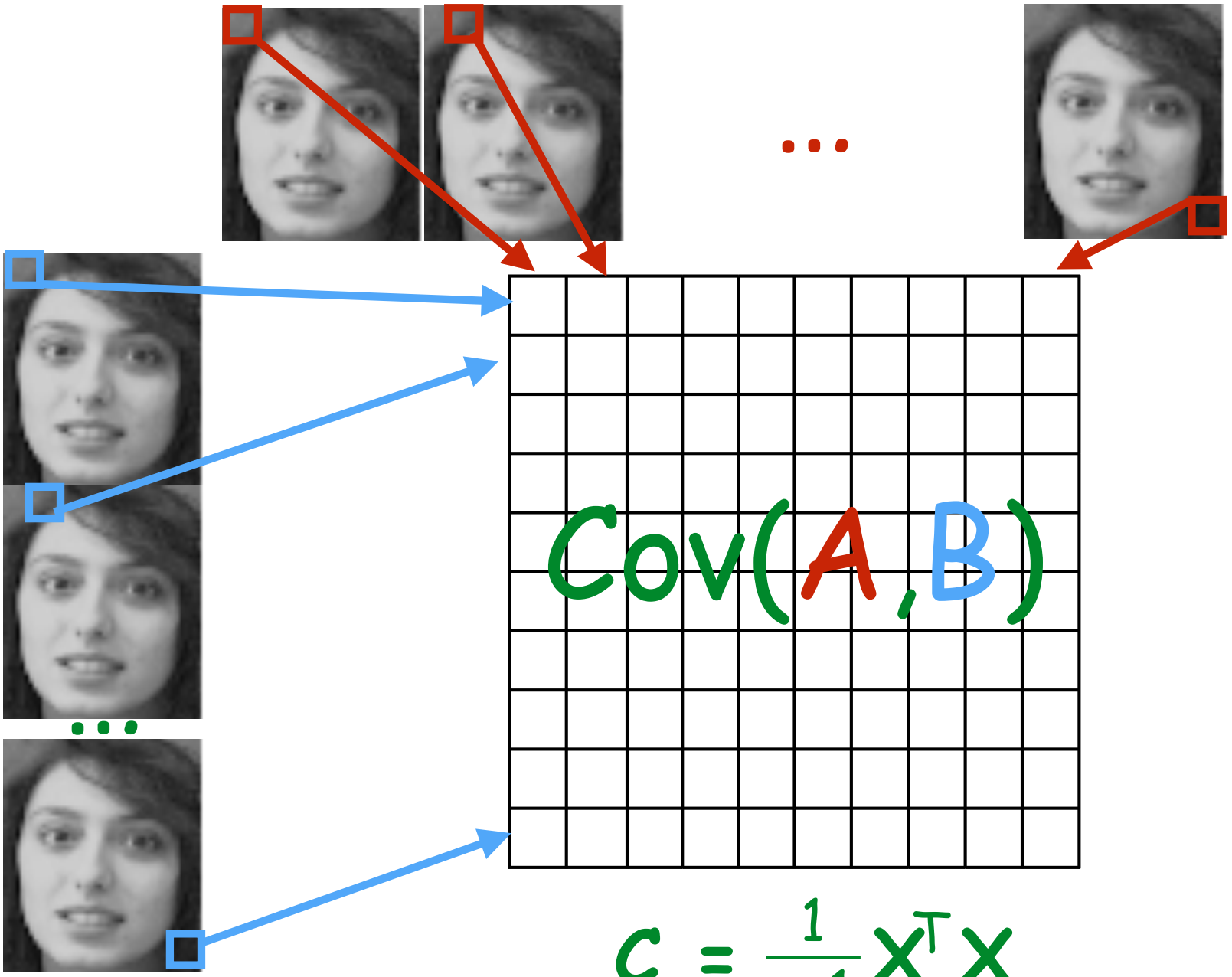
variable A B

mean μ_A μ_B

$$Cov(A,B) = \frac{\sum_{i=1}^{n}(a_i-\mu_A)(b_i-\mu_B)}{n-1}$$



Covariance Matrix C (p by p)



$$C = \frac{1}{n-1} X^T X$$

assuming X is already centered

$$\max_{w} w^T C w \longleftrightarrow C w = \lambda w$$

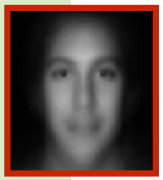
finding eigen vectors of matrix C
with top k largest eigen values

Eigen Pairs of Matrix C

$Cx = \lambda x$

large

small

λ_1, x_1 

λ_2, x_2 

λ_3, x_3 

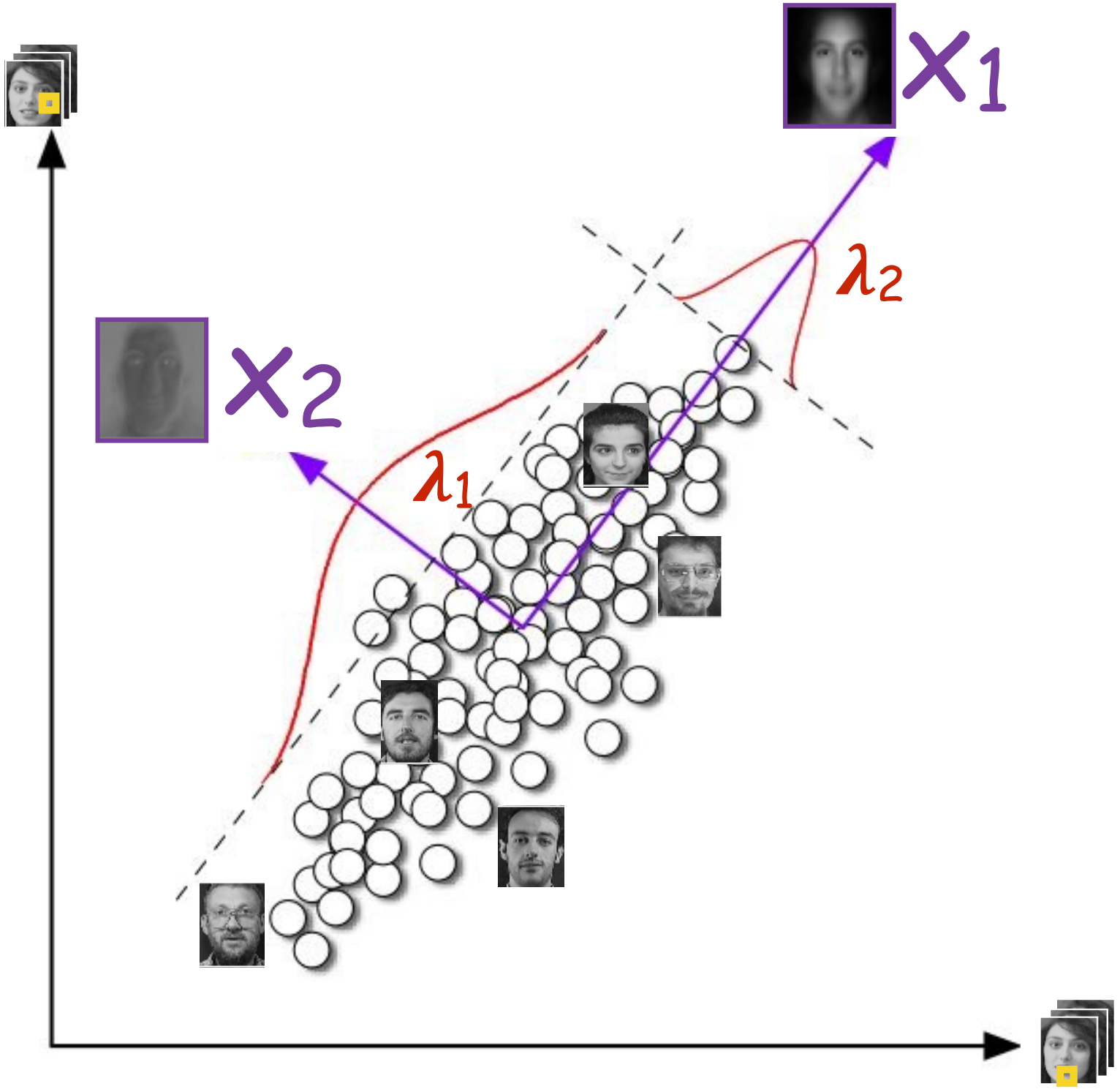
λ_4, x_4 

...

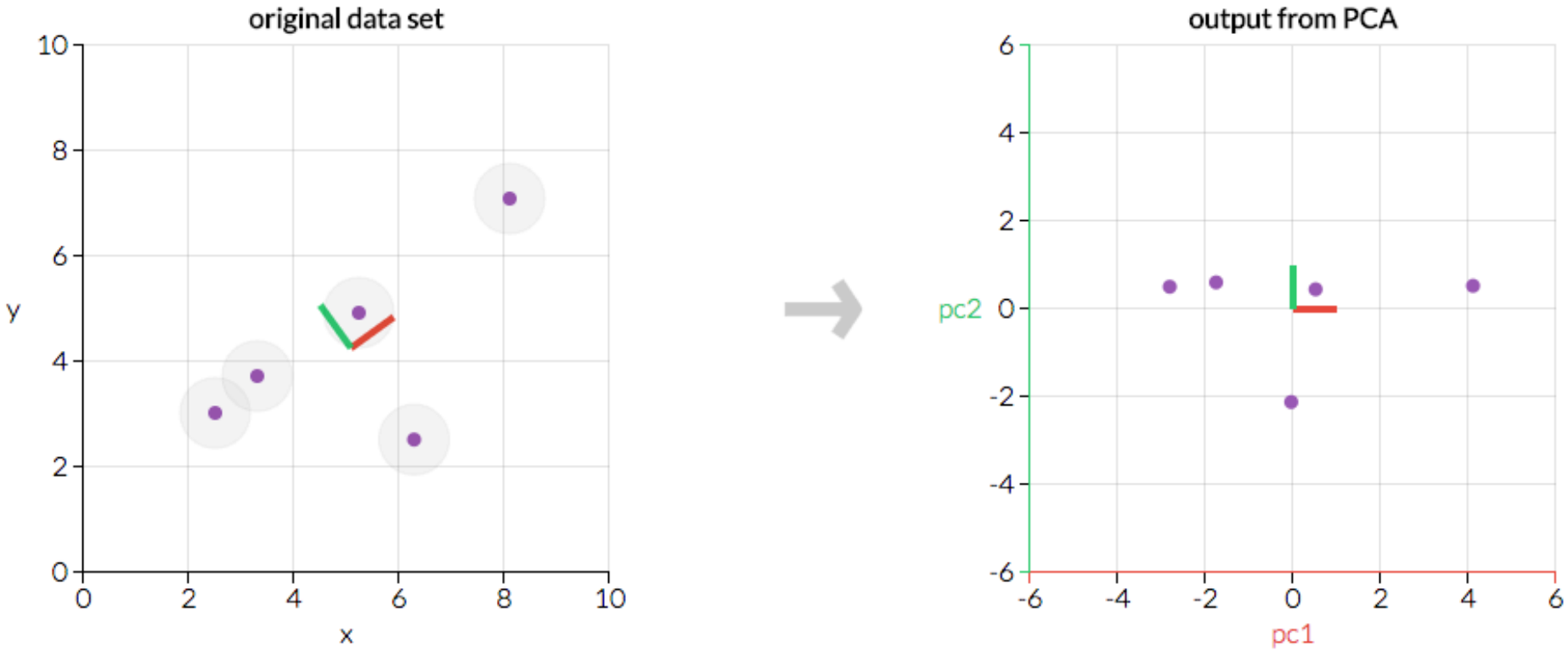
...

Principle Components

top K eigen vectors

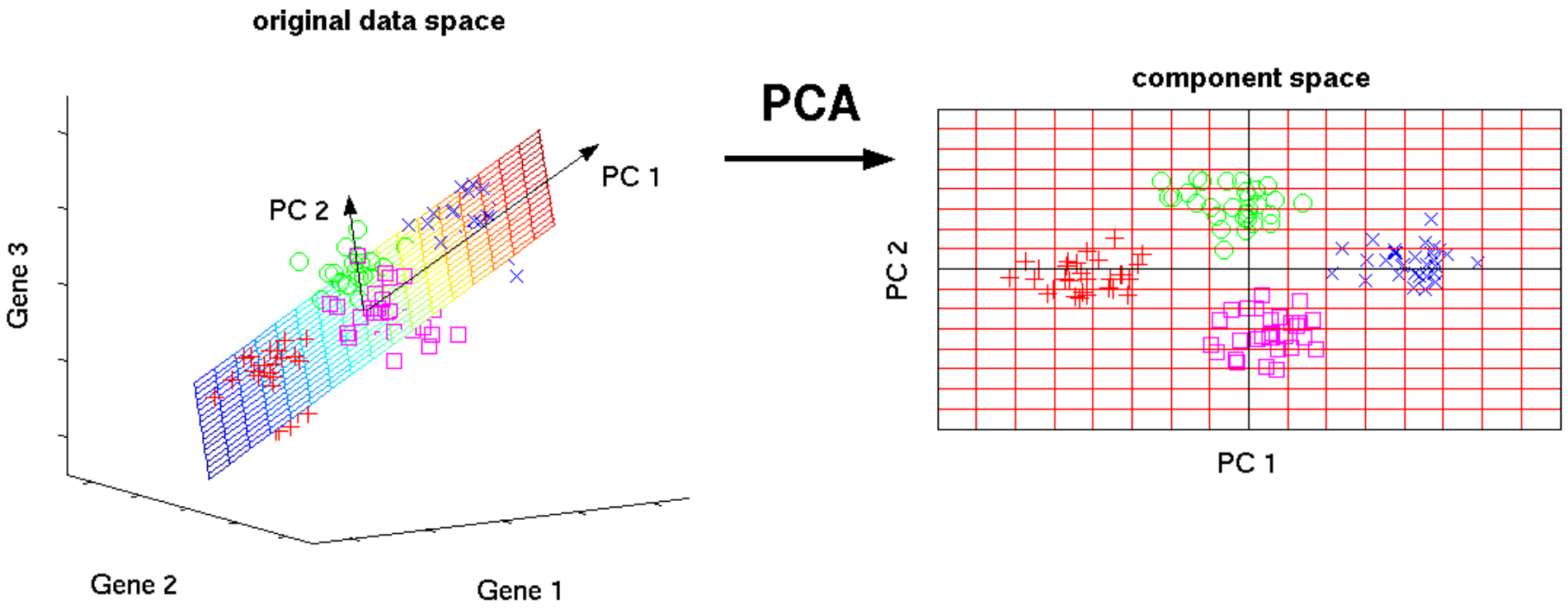

x1
x2
λ1
λ2

Demo

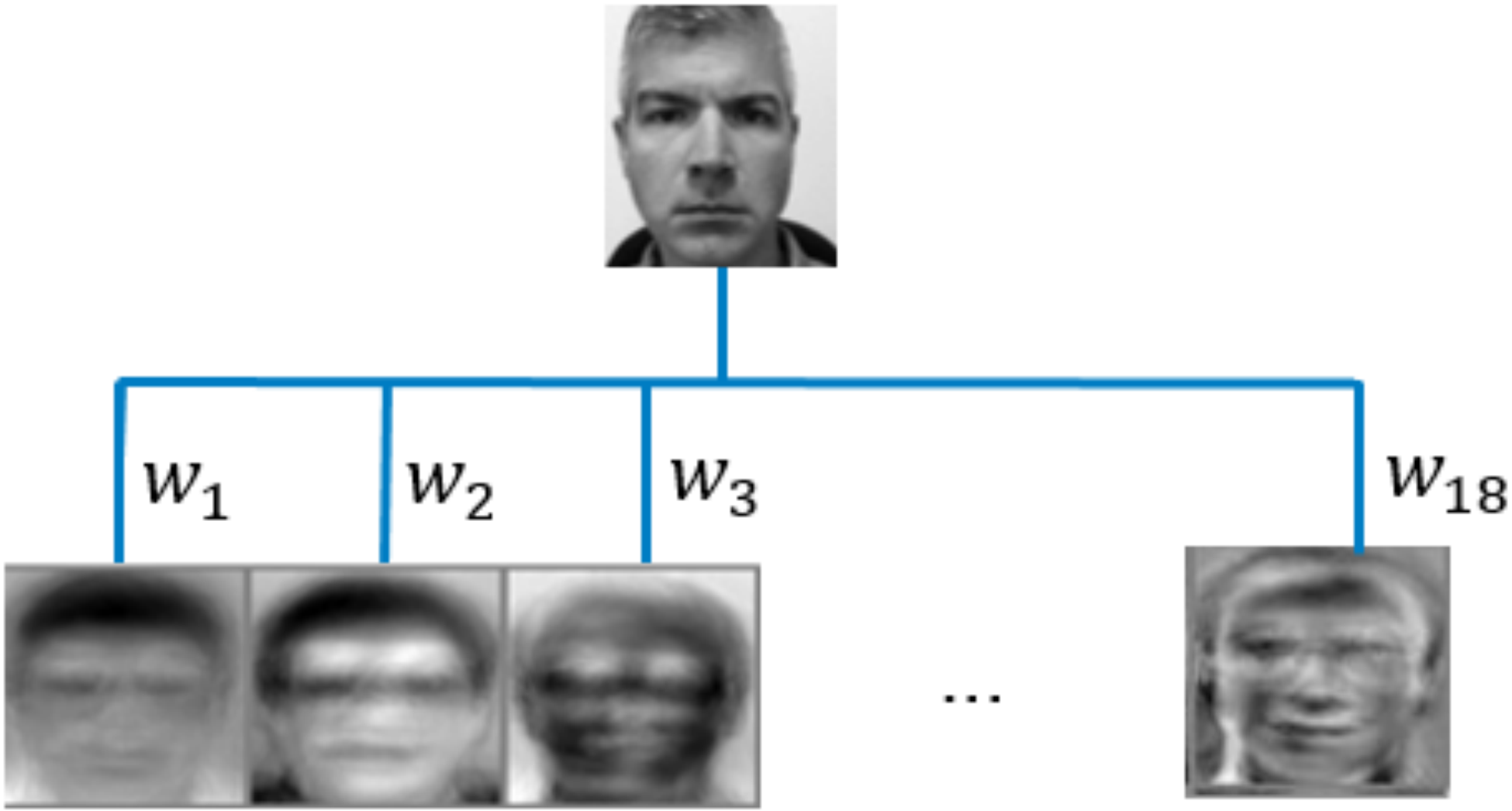


http://setosa.io/ev/principal-component-analysis/

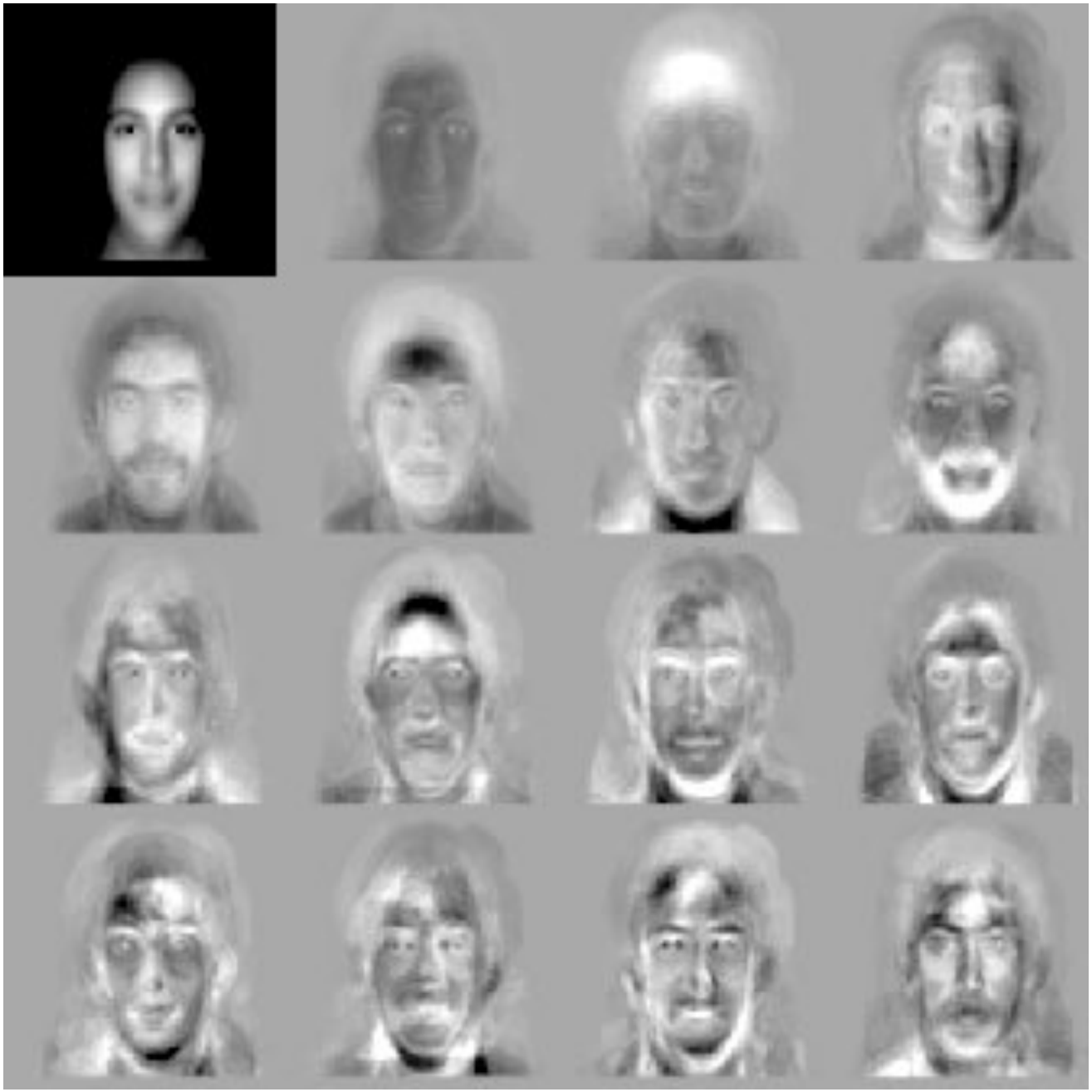
PCA in 3D


original data space
PC 2
PC 1
Gene 3
Gene 2
Gene 1
PCA
component space
PC 2
PC 1

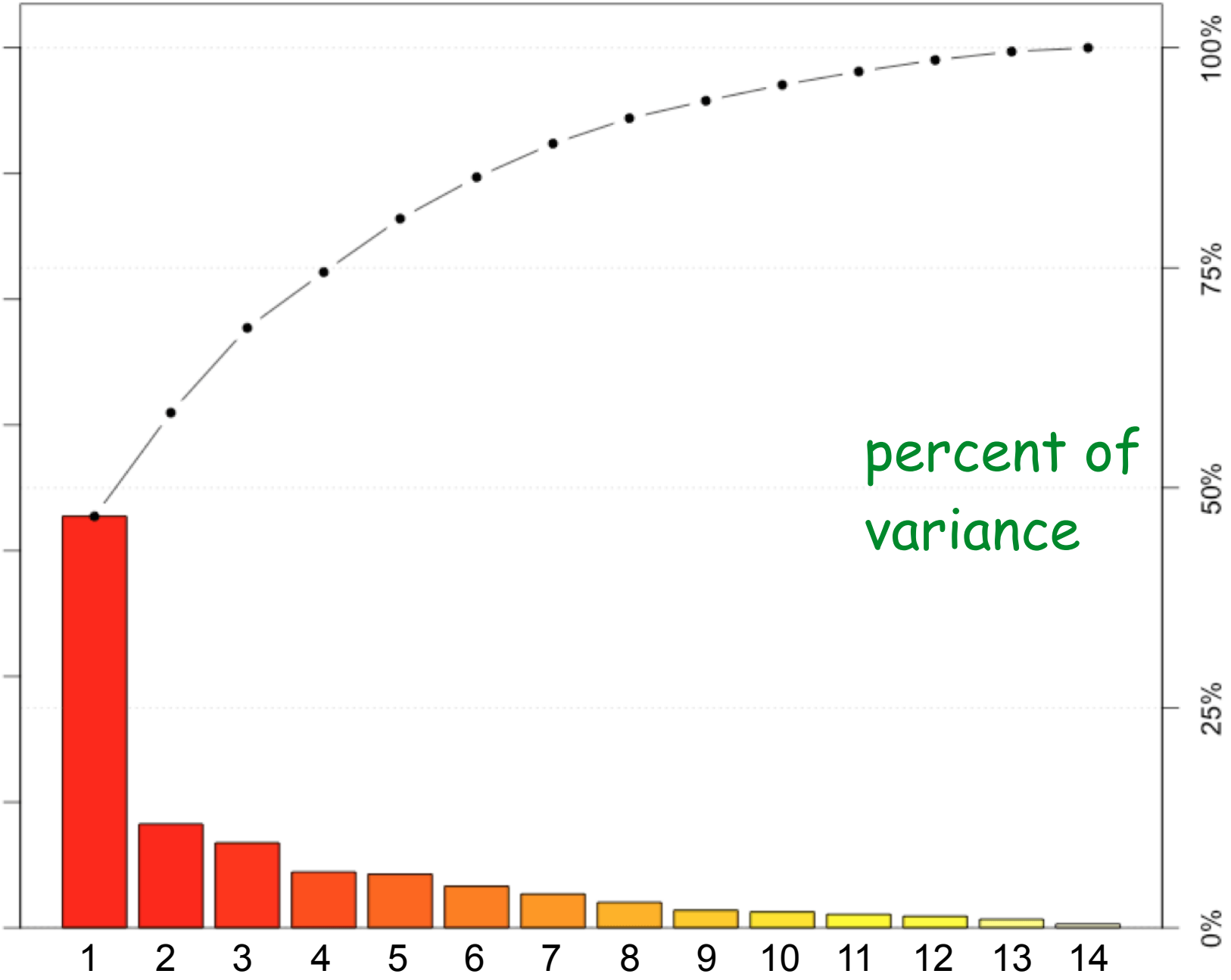
Eigen Face



Percent of Variance



How to choose K ?

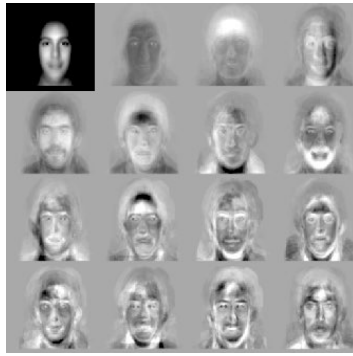


PCA

1. Centering data $X' = X - \mu$
2. Compute covariance matrix $C = \frac{1}{n-1} X'^{T} X'$
3. Compute eigen pairs of matrix C
4. Sort eigen pairs according to eigen values in descending order
5. Use top k eigen vectors to project the data X into a lower dimensional data $X W = Y$

Face Recognition

1. Use PCA to reduce the dimensionality of the images



$$X \, W = X'$$

2. Compute the Euclidean Distance between the query image and all the images

query 

d

X'

3. Sort the images based upon distances

small distance → large distance

query 

result: