MATH 312
LECTURE 23 : PCA / MARKOV Chains.

## III. Principal Components Analysis.

Now we will have a bunch of data points, which we write in a matrix:

$$X = \begin{bmatrix} \vec{x}_1 & \cdots & \vec{x}_n \end{bmatrix}, \quad \vec{x}_i = \begin{bmatrix} x_{i1} \\ \vdots \\ x_{im} \end{bmatrix} \in \mathbb{R}^m$$

→ Each row corresponds to a variable.

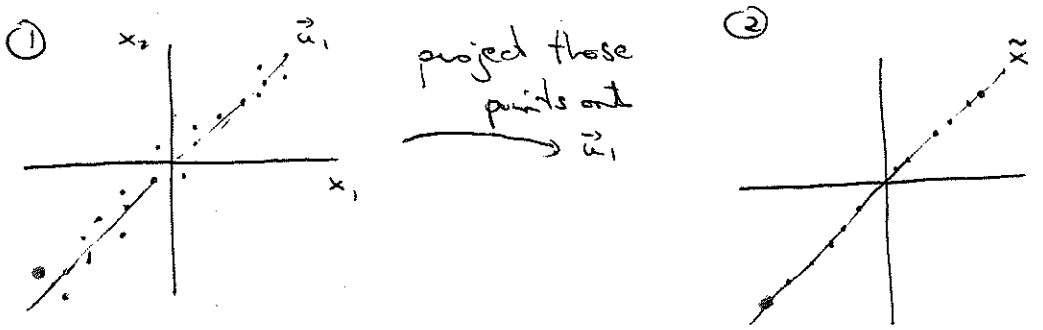
(High dimensional data)

↑ points, samples, measurements →

Examples: The cols. can be all the students at Penn, while the rows might be their height, age, major, ...

- The goal is to be able to distinguish or classify the points with much less than $m$ variables, without losing much information.

Imagine we have points in $\mathbb{R}^2$ ($m = 2$) that look as follows:




In ①, each point is characterised by two variables $(x_1, x_2)$.

In ②, we lose <u>some</u> information (red and green points cannot be distinguish in ②), however, most of them are still split apart.

→ Most variance among data remains in ②.

→ In ②, we only need one variable ($\tilde{x}$) = the distance along $\vec{u}_1$.

In each original point $\vec{x}$ (cols. of $X$) are now identified by their projection onto $\vec{u}_1$.

- What is the relation with SVD?

Write $X = \left[\vec{x}_1 \dots \vec{x}_n\right] = \sigma_1 \vec{u}_1 \vec{v}_1^T + \dots + \sigma_r \vec{u}_r \vec{v}_r^T$.

Notice that: $A\vec{v}_i = \sigma_i \vec{u}_i \Rightarrow \underbrace{A^T A \vec{v}_i}_{= \sigma_i^2 \vec{v}_i} = \sigma_i A^T \vec{u}_i \rightarrow \boxed{\sigma_i \vec{v}_i^T = \vec{u}_i^T A}$

So $X = \vec{u}_1(\vec{u}_1^T X) + \dots + \vec{u}_r(\vec{u}_r^T X)$, where

$$\vec{u}_i(\vec{u}_i^T X) = \begin{bmatrix} | & & | \\ \vec{u}_i\vec{u}_i^T\vec{x}_1 & \cdots & \vec{u}_i\vec{u}_i^T\vec{x}_n \\ | & & | \end{bmatrix}$$

$\uparrow$ projection of the data points $\vec{x}_1, \dots, \vec{x}_n$ onto $\vec{u}_i$. $\uparrow$

In our 2d example,

$X = \underbrace{\vec{u}_1(\vec{u}_1^T X)} + \vec{u}_2(\vec{u}_2^T X)$.

this gives picture ② (each column gives where the original points go onto the line spanned by $\vec{u}_1$).

Remark: $\vec{u}_1$ gives the line ~~of best~~ through the origin of best fit.

$\vec{u}_1, \vec{u}_2$ gives the plane through the origin of best fit. (*)

$\vdots$

→ The $\vec{u}$'s are called principal components. They provide a new basis (new variables) to represent our data.

(*) We have to substract the mean to each row to obtain good results.

Example: Database of 216 patients, 121 with ovarian cancer and 95 without it.

Each patient identified through 4000 genes.

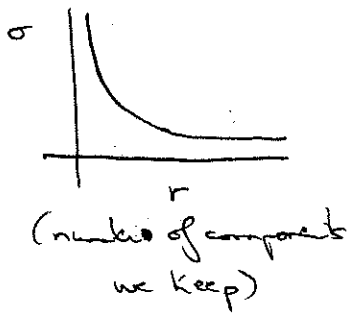
$$X = \left[\begin{array}{c} \vec{x}_1 \quad \cdots \quad \vec{x}_{216} \end{array}\right] \Big\} \text{4000 rows}$$

We want to see if there are some common factors that predict the ovarian cancer.

It seems likely that many of the genes are correlated.

We perform the SVD $\rightsquigarrow$ we might obtain a list of $\vec{u}$'s.
(up to 4000!)

How many are important? $\rightsquigarrow$ We can look the weights $\sigma^2$.
(graph)




Let's plot the patients in the basis given by $\{\vec{u}_1, \vec{u}_2, \vec{u}_3\}$, that is, we project each $\vec{x}_i$ into the subspace spanned by $\uparrow$

vectors of 40000 components

$$\tilde{x}_i = \begin{bmatrix} \tilde{x}_1 \\ \tilde{x}_2 \\ \tilde{x}_3 \end{bmatrix} = \vec{u}_1(\vec{u}_1^T \vec{x}_i) + \vec{u}_2(\vec{u}_2^T \vec{x}_i) + \vec{u}_3(\vec{u}_3^T \vec{x}_i) \Rightarrow$$

$$\Rightarrow \tilde{x}_i \Big|_{\text{span}\{\vec{u}_1,\vec{u}_2,\vec{u}_3\}} = \begin{bmatrix} \vec{u}_1^T \vec{x}_i \\ \vec{u}_2^T \vec{x}_i \\ \vec{u}_3^T \vec{x}_i \end{bmatrix}$$ This can be plotted in a 3d graph!

(see Matlab...)

Example: Eigenfaces

Database 1600 pictures → 25 people × 64 pictures/person.

Each picture is 192·168 pixels. = 32256

$$X = \begin{bmatrix} \vec{x}_1 & \cdots & \vec{x}_{1600} \end{bmatrix} \Big\} \text{32256 rows (pixels)}$$

↑ each image is stored as a column vector.

- We do the SVD of X − mean of rows.

↳ The $\vec{u}$'s are called "eigenfaces" → they keep most common features in human faces.

↳ They provide a basis for the space of "faces".

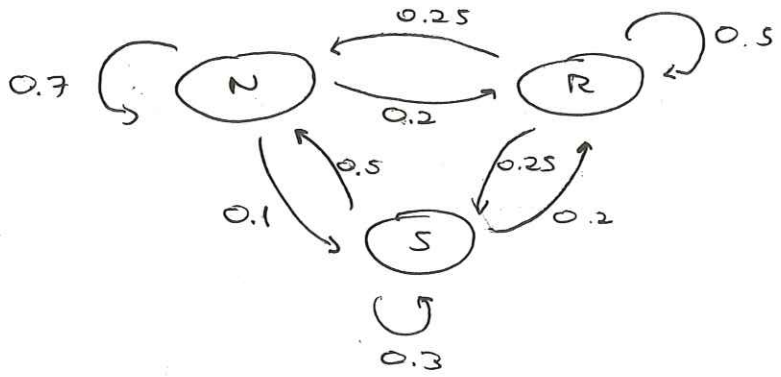
A new face ("test face") can be projected into the eigenspace given by the $\vec{u}$'s (building using $X$, the "training set").

↳ Now we can see if it is close to one of the faces in $X$ or not → face recognition.

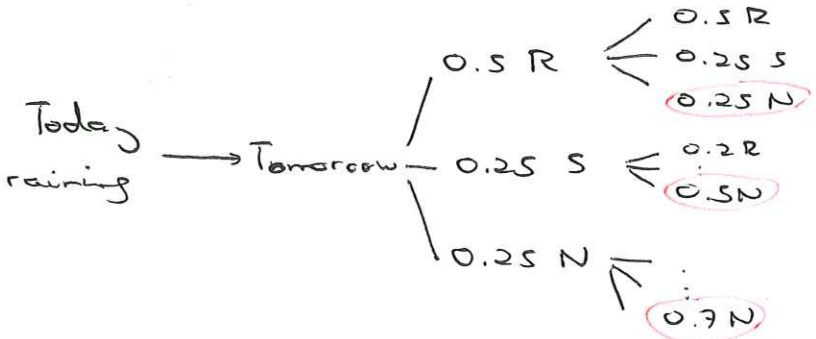
$$\text{testface} = \left[\;\right] \rightsquigarrow \underbrace{U_r U_r^T \,\text{testface}}_{\text{approximation of testface using first } r \text{ eigenfaces.}} \qquad (U_r = [\vec{u}_1 \dots \vec{u}_r]$$

# New Chapter: Markov Chains.

Consider the following silly model for weather prediction:




If today is raining, what is the probability that the day after tomorrow is a nice day?




So after tomorrow,

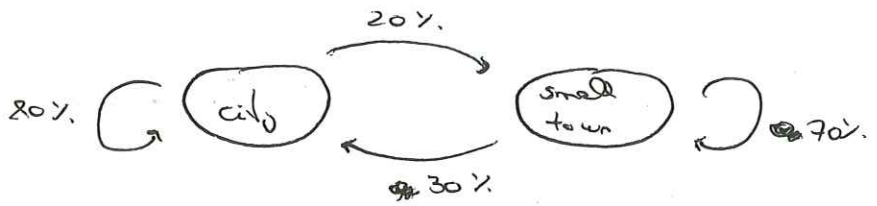
$$p = (0.5)(0.25) + (0.25)(0.5) + (0.25)(0.7) = 0.425$$

What is the probability that it will snow in a week, if today is a nice day?

Moreover, along the years, which fraction of days are nice, rainy a snowy?

→ The answers are not straightforward, but we will see they are easy to compute.

- Let's use a different simpler example:




If today 50% of people live in cities, what is the distribution next year?

In cities: $(0.5)\cdot(0.8) + (0.5)(0.3) = 0.55 \rightarrow 55\%$.

In towns: $(0.5)\cdot(0.7) + (0.5)\cdot(0.2) = 0.45 \rightarrow 45\%$.

And next one: (↓ same numbers)

In cities: $(0.55)\cdot(0.8) + (0.45\cdot 0.3) = 0.575 \rightarrow 57'5\%$.

So ↓ towns → 42'5%.

Create the "state" vector $\vec{x}(k) = \begin{bmatrix} x_1(k) \\ x_2(k) \end{bmatrix}$, where

$x_1(k) = \%$ in cities in year $k$

$x_2(k) = \%$ in town in year $k$

(of course, $x_1(k) + x_2(k) = 1 \quad \forall k$).

How to find $\vec{x}(k+1)$?

$$\left.\begin{aligned} x_1(k+1) &= x_1(k)\cdot(0.8) + x_2(k)\cdot(0.3) \\ x_2(k+1) &= x_1(k)\cdot(0.2) + x_2(k)(0.7) \end{aligned}\right\} \rightarrow \vec{x}(k+1) = \underbrace{\begin{bmatrix} 0.8 & 0.3 \\ 0.2 & 0.7 \end{bmatrix}}_{\text{transition matrix}} \vec{x}(k)$$

Notice: columns add up to 1.

Q: What is the transition matrix for the weather model?

State vector: $\vec{x} = \begin{bmatrix} x_S \\ x_N \\ x_R \end{bmatrix}$,

$$\vec{x}(k+1) = P\,\vec{x}(k) \rightarrow P = \begin{bmatrix} 0.3 & 0.1 & 0.25 \\ 0.5 & 0.7 & 0.25 \\ 0.2 & 0.2 & 0.5 \end{bmatrix}$$

($P_{ij}$ = probability of going in one step from $i$ to $j$)

Check previous result:

$$p = P^2 \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} = P \begin{bmatrix} 0.25 \\ 0.25 \\ 0.5 \end{bmatrix} = \begin{bmatrix} (0.3)(0.25) + (0.1)\cdot(0.25) + (0.5)\cdot(0.25) \\ (0.5)\cdot(0.25) + (0.7)\cdot(0.25) + (0.5)(0.25) \\ \cdots \end{bmatrix}$$

probability of nice day after 2 days.

$p$: probability after two days; $P^2$: today raining; $\begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$: today raining.

- Def: Markov (or stochastic) matrix

An $n \times n$ matrix is Markov if:

1) All columns add up to 1.

2) All entries are $\geq 0$.

- The sequence of states $\{\vec{x}(1), \vec{x}(2), \ldots, \vec{x}(k), \ldots\}$ defined by a Markov matrix is called a Markov chain (discrete)

Remark: Key feature of this model: Every state only depends on the previous step: $\vec{x}(k+1) = A\vec{x}(k)$.

Example: 1) The evolution of a particle, defined through its position and velocity, is a Markov process (by Newton's law, if we know position and velocity now, we know the future).

ok, I'm considering classical mechanics.

2) Poker is not a Markov process: you use (or should use) the knowledge about which cards have been appearing from the start.

Some questions:

1) Each state $\vec{x}(k)$ describes the distribution at that moment (% in cities, % in towns, and so on).

So $\vec{x}(k+1)$ needs to be a % distribution too, i.e, the components should add up to 1.

Is it true? That is, for A Markov matrix, does $A\vec{x}$ add up to 1 if $\vec{x}$ does?

Let's see that yes:

Notice that the addition of the components of a vector is given by

$$[1 \dots 1]\vec{x} = [1 \dots 1]\begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix} = x_1 + \dots + x_n = 1 .$$

(assumption (%...))

But then,

$$[1 \dots 1]\, A\vec{x} = [1 \dots 1]\left[\vec{a}_1 \,|\, \dots \,|\, \vec{a}_n\right]\vec{x} = [1 \dots 1]\,\vec{x} = x_1 + \dots + x_n = 1 .$$

Markov matrix $\Rightarrow$

- $[1 \dots 1]\vec{a}_1$ = addition components first column = 1
- $[1 \dots 1]\vec{a}_2$ = " " second column = 1
- ⋮
- All cols. of A add up to 1!