

Title: PSI 2018/2019 - Machine Learning - Lecture 7

Speakers: Lauren Hayward Sierens

Collection: PSI 2018/2019 - Machine Learning (Hayward Sierens)

Date: April 02, 2019 - 9:00 AM

URL: <http://pirsa.org/19040002>

TOMORROW'S LECTURE: 10:15am

Outline for today: Unsupervised Learning (UL)

- examples of UL tasks
- relationship to SL
- dimensional reduction
 - ↳ principal component analysis (PCA)

Unsupervised Learning (UL):

Goal: learn structural properties (features) of unlabelled data

$$\text{Dataset } \mathcal{D} = \{\vec{x}\}$$

↳ each datapoint has N components $\vec{x} = (x_1, x_2, \dots, x_N)$

↳ dataset has M points: $\mathcal{D} = \{\vec{x}_1, \dots, \vec{x}_M\}$

Within this class, we will study two classes of UL algorithms:

① Dimensional reduction

↳ today & tomorrow

Idea: reduce the dimensionality of the dataset by finding the most important features

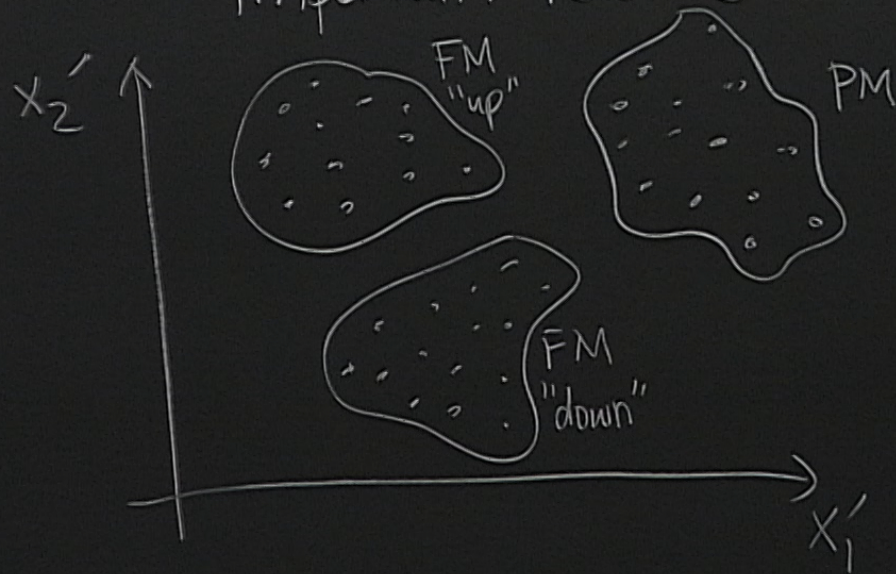
$$N \rightarrow N' < N$$

If $N' \leq 3$ then we can visualize the data.

In many cases the dataset might naturally form "clusters".

eg) Ising spin configs. at diff. temperatures

↳ we know magnetization is an important feature



② Generative modelling

↳ next week

Idea: use the datapoints $\vec{x}_1, \vec{x}_2, \dots, \vec{x}_M$ to learn about the underlying prob. dist $p(\vec{x})$ and generate new samples from it.

Relationship between SL and UL

UL: $\mathcal{D} = \{\vec{x}\} \longrightarrow \text{learn } p(\vec{x})$

SL: $\mathcal{D} = \{(\vec{x}, \vec{y})\} \longrightarrow \text{learn conditional prob. dist. } p(\vec{y} | \vec{x})$

UL and SL are not formally different
due to the chain rule of prob.:

For $\vec{X} = (x_1, x_2, \dots, x_N)$

$$p(\vec{X}) = p(x_N | x_1, x_2, \dots, x_{N-1}) p(x_1, x_2, \dots, x_{N-1})$$

= ...

$$= \left[\prod_{i=2}^N p(x_i | x_1, \dots, x_{i-1}) \right] p(x_1)$$

$\Rightarrow$ We can express a UL problem as
N-1 SL problems along with
the easier problem of learning $p(x_i)$

eg) for indep. spin configs. with $\mathbb{Z}_2$ symm.:
$$p(x) = p(\uparrow) = p(\downarrow) = \frac{1}{2}$$

We can also express a SL problem
in terms of UL problems using:

$$P(\vec{y}|\vec{x}) = \frac{P(\vec{x}, \vec{y})}{\sum_{\vec{y}'} P(\vec{x}, \vec{y}')}$$

Dimensional Reduction using Principal Component Analysis (PCA)

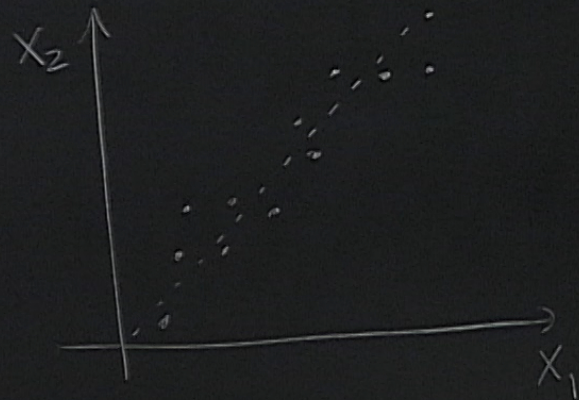
Idea: Generate a lower-dim. representation of the N -dim. dataset $\mathcal{D}$ through a linear transformation.

Principal

representation

in a

An example for $N=2$ where
PCA would be useful



x_1 and x_2 are highly (linearly)
correlated. Knowing x_1 is enough to
infer x_2 with high precision.

However, PCA is not always useful.

Let us store our M datapoints
in an $M \times N$ matrix:

$$X = \begin{bmatrix} \vec{x}_1 \\ \vec{x}_2 \\ \vdots \\ \vec{x}_M \end{bmatrix} = \begin{bmatrix} x_{11} & x_{12} & \dots & x_{1N} \\ x_{21} & x_{22} & \dots & x_{2N} \\ \vdots & \vdots & \ddots & \vdots \\ x_{M1} & x_{M2} & \dots & x_{MN} \end{bmatrix}$$

infer x_2 with high precision.

To perform PCA, the mean of the datapoints must be zero, so first shift the "centre" of each column:

$$\vec{X}_i \rightarrow \vec{X}_i - \frac{1}{M} \sum_{k=1}^M \vec{X}_k$$

$$\left(x_{ij} \rightarrow x_{ij} - \frac{1}{M} \sum_{k=1}^M x_{kj} \quad \forall j \right)$$

$$1 \leq i \leq M, \quad 1 \leq j \leq N$$

Now consider the $N \times N$ matrix

$$V_X \equiv \frac{1}{M-1} X^T X$$

- diag. elements store the variances of each component of $\vec{x}$
- off-diag. elements store the covariances between pairs of components

Goal of PCA: Find a new representation $X' = X P$ such that $V_{X'}$ is diagonal.

P is an $N \times N$ matrix encoding the linear transformation.

Now note that:

$$V_{X'} = \frac{1}{M-1} X'^T X' = \frac{1}{M-1} (X P)^T X P = \frac{1}{M-1} P^T X^T X P = P^T V_X P$$

When this matrix is diagonal:

$$P^T V_X P = D = \begin{pmatrix} \lambda_1 & & 0 \\ & \lambda_2 & \\ 0 & & \lambda_N \end{pmatrix}$$

Since $V_X = \frac{1}{M-1} X^T X$: $P^{-1} = P^T$

So PCA amounts to solving an eigenvalue problem

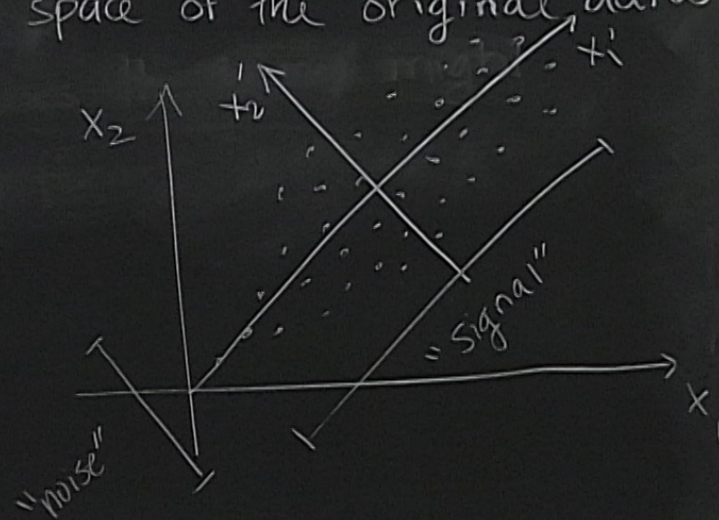
$$V_X = P D P^{-1}$$

where each eigenvalue λ_j is $\frac{1}{M-1} \sum_{i=1}^M X_{ij}'^2 \geq 0$

Let us sort the eigenvalues so $\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_N$

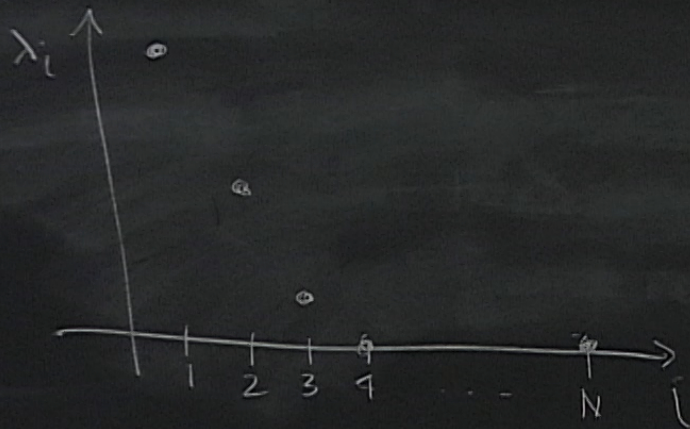
$$V_X P = D = \begin{pmatrix} \lambda_1 & 0 \\ 0 & \lambda_N \end{pmatrix}$$

Now the first principal component x_1' corresponds to the direction in the space of the original data with highest variance λ_1 .



sort the eigenvalues so $\lambda_1 \gg \lambda_2 \gg \dots \gg \lambda_N$

Let us plot λ_i vs i :



If these eigenvalues quickly become small, the lower-dim. representation $\vec{x}' = (x'_1, x'_2, \dots, x'_p)$ ($p \ll N$) corresponding to the first few principal components will be a valid and efficient representation of the data's features.

Tomorrow. non-linear method for dim.
reduction called tSNE prob.

(N)

Scient