Learning From Data
Lecture 10: Mixture of Gaussians & EM

Yang Li yangli@sz.tsinghua.edu.cn

November 26, 2020

Today's Lecture

Unsupervised Learning (Part IV)

- Mixture of Gaussians
- The EM Algorithm
- Factor Analysis

Final Project Information

Review: k-means clustering

Given input data $\{x^{(1)}, \ldots, x^{(m)}\}$, $x^{(i)} \in \mathbb{R}^d$, **k-means clustering** partition the input into $k \leq m$ sets $C_1, \ldots, C_k$ to minimize the within-cluster sum of squares (WCSS).

$$\underset{C}{\operatorname{argmin}} \sum_{j=1}^{k} \sum_{x \in C_j} \|x - \mu_j\|^2$$

Lloyd's Algorithm (1957,1982)

Let $c^{(i)} \in \{1, \ldots, k\}$ be the cluster label for $x^{(i)}$

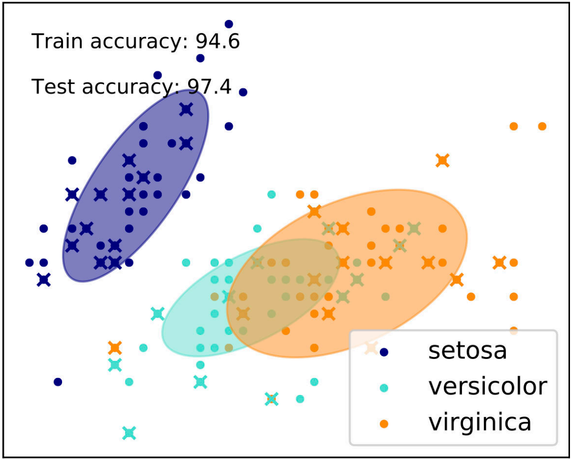
```
Initialize cluster centroids μ_1,...μ_k ∈ R^n randomly
Repeat until convergence{
  For every i,
    c^(i) := argmin_j ||x^(i) − μ_j||^2   ← assign x^(i) to the cluster
                                            with the closest centroid
  For each j
    μ_j := Σ_{i=1}^m 1{c^(i)=j} x^(i) / Σ_{i=1}^m 1{c^(i)=j}   ← update centroid
}
```

$$c^{(i)} := \operatorname{argmin}_j \|x^{(i)} - \mu_j\|^2$$

$$\mu_j := \frac{\sum_{i=1}^{m} \mathbf{1}\{c^{(i)}=j\} x^{(i)}}{\sum_{i=1}^{m} \mathbf{1}\{c^{(i)}=j\}}$$

Mixture of Gaussians

A "soft" version of k-means clustering.



Clustering results of iris dataset using *mixture of Gaussians*

Mixture models

Model-based clustering

A **mixture model** assumes data are generated by the following process:

1. Sample $z^{(i)} \in \{1, \dots, k\}$ and $z^{(i)} \sim \text{Multinomial}(\phi)$

$$p(z^{(i)} = j) = \phi_j \text{ for all } j$$

   $z^{(i)}$ are called **latent variables**.

2. Sample observables $x^{(i)}$ from some distribution $p(x^{(i)}, z^{(i)})$:

$$p(x^{(i)}, z^{(i)}) = p(x^{(i)}|z^{(i)})p(z^{(i)})$$

Examples:

- Unsupervised handwriting recognition is a mixture with 10 Bernoulli distributions
- Financial return estimation uses a mixture of 2 Gaussians for normal situation and crisis time distribution

Mixture of Gaussians

Mixture of Gaussians Model:

$$z^{(i)} \sim \text{Multinomial}(\phi)$$
$$x^{(i)}|z^{(i)} = j \sim \mathcal{N}(\mu_j, \Sigma_j)$$

How to learn ϕ_j, μ_j and Σ_j for all j ?

$z^{(i)}$ is known: (supervised) use maximum likelihood estimation (quadratic discriminant analysis).

$$\phi_j = \frac{1}{m}\sum_{i=1}^{m} \mathbf{1}\{z^{(i)} = j\}, \quad \mu_j = \frac{\sum_{i=1}^{m} \mathbf{1}\{z^{(i)} = j\} x^{(i)}}{\sum_{i=1}^{m} \mathbf{1}\{z^{(i)} = j\}}$$
$$\Sigma_j = \frac{\sum_{i=1}^{m} \mathbf{1}\{z^{(i)} = j\}(x^{(i)} - \mu_j)(x^{(i)} - \mu_j)^T}{\sum_{i=1}^{m} \mathbf{1}\{z^{(i)} = j\}}$$

$z^{(i)}$ is unknown: (unsupervised) use **expectation maximization**

The EM Algorithm

The EM algorithm is an iterative method for maximum likelihood estimation when the model depends on **latent (unobserved) variables**.

Log-likelihood of data:

$$l(\theta) = \sum_{i=1}^{m} \log p(x^{(i)};\theta) = \sum_{i=1}^{m} \log \sum_{z^{(i)}=1}^{k} p(x^{(i)}, z^{(i)};\theta)$$

Main idea: iterate over two steps:

- Expectation (E) step : guess $z^{(i)}$
- Maximization (M) step : update θ via maximum likelihood estimation based on guessed $z^{(i)}$'s

Generalized EM Algorithm

Listing 1: Generalized EM Algorithm

```
Initialize θ
Repeat untill convergence {
  (E-step) For each i , set
        Q_i(z^(i)) := p(z^(i)|x^(i); θ) ← Soft assignment:
   posterior distribution z|x under θ
  (M-step) Set
        θ := argmax_θ Σ_i Σ_{z^(i)} Q_i(z^(i)) log [ p(x^(i), z^(i); θ) / Q_i(z^(i)) ]    (⋆)
        ← Update parameter θ
}
```

$$Q_i(z^{(i)}) := p(z^{(i)}|x^{(i)};\theta)$$

$$\theta := \operatorname*{argmax}_{\theta} \sum_i \sum_{z^{(i)}} Q_i(z^{(i)}) \log \frac{p(x^{(i)}, z^{(i)}; \theta)}{Q_i(z^{(i)})} \quad (\star)$$

We will show...

- Solving $(\star)$ is equivalent to $\operatorname{argmax}_\theta l(\theta)$
  $\rightarrow$ Equation $(\star)$ is a (tight) lower bound on log-likelihood $l(\theta)$
- This algorithm converges.

Proof of Correctness: E-step

Define

$$J(Q,\theta) = \sum_i \sum_{z^{(i)}} Q_i(z^{(i)}) \log \frac{p(x^{(i)}, z^{(i)}; \theta)}{Q_i(z^{(i)})}$$

Proposition 1

1. *$J(Q,\theta)$ is a lower bound on log-likelihood $l(\theta)$*
2. *This lower bound is tight when $Q_i(z^{(i)}) = p(z^{(i)}|x^{(i)};\theta)$*

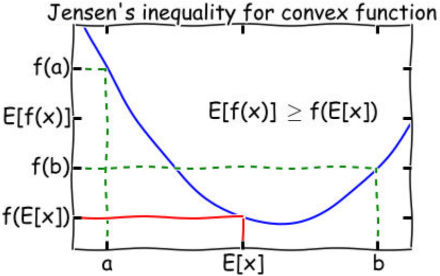
(Hint: use Jensen's inequality)

Jensen's Inequality

Theorem 1

*Let f be a **convex** function, and let X be a random variable. Then*

$$\mathbb{E}[f(X)] \geq f(\mathbb{E}[X])$$



Remarks

1. Let f be a **concave** function, then $\mathbb{E}[f(X)] \leq f(E[X])$
2. When $f(X)$ is a constant function, $\mathbb{E}[f(X)] = f(\mathbb{E}[X])$

Proof of Convergence

Proposition 2

EM always monotonically improves the log likelihood, i.e. Let $\theta^{(t)}$ *be the parameter value in the* t*-th iteration*

$$l(\theta^{(t)}) \leq l(\theta^{(t+1)})$$

EM for mixture of Gaussians

Gaussian Mixture Model

$$z^{(i)} \sim \text{Multinomial}(\phi)$$
$$x^{(i)}|z^{(i)} \sim \mathcal{N}(\mu_j, \Sigma_j)$$

Learn parameters μ, Σ, ϕ

E-Step: $w_j^{(i)} = Q_i(z^{(i)} = j) = p(z^{(i)} = j|x^{(i)}; \phi, \mu, \Sigma)$

M-Step: Maximize $\sum_{i=1}^{m} \sum_{z^{(i)}} Q_i(z^{(i)}) \log \frac{p(x^{(i)}, z^{(i)}; \phi, \mu, \Sigma)}{Q_i(z^{(i)})}$ with respect to ϕ, μ and Σ

Expectation Maximization for Gaussian Mixtures

Listing 2: EM for Gaussian Mixtures

Repeat untill convergence {

(E-step) For each i, j , set

$$w_j^{(i)} := p(z^{(i)} = j | x^{(i)}; \phi, \mu, \Sigma)$$

(M-step) Update parameters: assume $\phi_j = \mathbb{E}[w_j]$

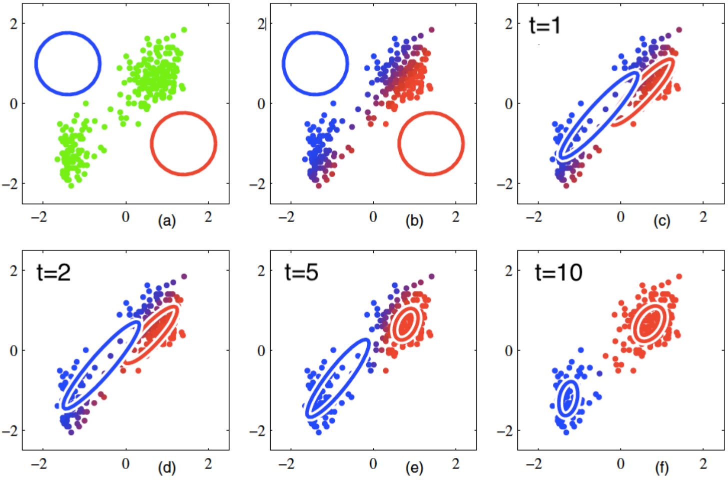
$$\phi_j := \frac{1}{m} \sum_{i=1}^{m} w_j^{(i)}$$

$$\mu_j := \frac{\sum_{i=1}^{m} w_j^{(i)} x^{(i)}}{\sum_{i=1}^{m} w_j^{(i)}}$$

$$\Sigma_j := \frac{\sum_{i=1}^{m} w_j^{(i)} (x^{(i)} - \mu_j)(x^{(i)} - \mu_j)^T}{\sum_{i=1}^{m} w_j^{(i)}}$$

}

Illustration of EM steps



Comparison with k-means clustering

Listing 2: EM Algorithm

Repeat untill convergence {
(E-step) For each i, j,

$$w_j^{(i)} := p(z^{(i)} = j | x^{(i)}; \phi, \mu, \Sigma)$$

(M-step) Update parameters:

$$\phi_j := \frac{1}{m} \sum_{i=1}^{m} w_j^{(i)}$$

$$\mu_j := \frac{\sum_{i=1}^{m} w_j^{(i)} x_j}{\sum_{i=1}^{m} w_j^{(i)}}$$

$$\Sigma_j := \frac{\sum_{i=1}^{m} w_j^{(i)} (x^{(i)} - \mu_j)(x^{(i)} - \mu_j)^T}{\sum_{i=1}^{m} w_j^{(i)}}$$

}

Listing 3: (Llyod's) k-means Alg.

Repeat untill convergence {
(E-step) For every i,

$$c^{(i)} := \operatorname*{argmin}_j ||x^{(i)} - \mu_j||^2$$

(M-step) Update centroids:
For each j

$$\mu_j := \frac{\mathbf{1}\{c^{(i)} = j\} x^{(i)}}{\sum_{i=1}^{m} \mathbf{1}\{c^{(i)} = j\}}$$

}

Similar to k-means, Gaussian mixtures are also subject to local minimums.

Factor Analysis: Example

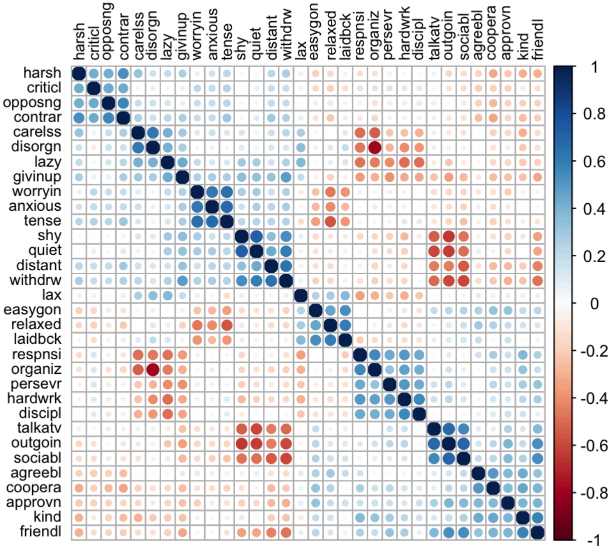
How much do you identify yourself with the following traits?

1-- the least 9 -- the most

	1	2	3	4	5	6	7	8	9
talkative	○	○	○	○	○	○	○	○	○
distant	○	○	○	○	○	○	○	○	○
careless	○	○	○	○	○	○	○	○	○
hardwork	○	○	○	○	○	○	○	○	○
anxious	○	○	○	○	○	○	○	○	○
kind	○	○	○	○	○	○	○	○	○

Self-ratings on 32 Personality Traits

Factor Analysis: Example



Pairwise correlation plot of 32 variables from 240 participants

Factor Analysis Terminology

- **observed random variables** $x \in \mathbb{R}^n$

$$x = \mu + \Lambda z + \epsilon$$

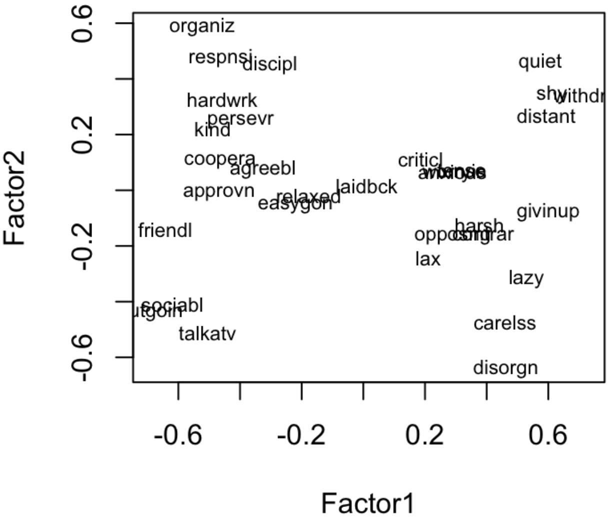
- **factor** $z \in \mathbb{R}^k$ is the hidden (latent) construct that "causes" the observed variables
- **factor loadings** $\Lambda \in \mathbb{R}^{n \times k}$: the degree to which variable x_i is "caused" by the factors
- $\mu, \epsilon \in \mathbb{R}^n$ are the mean and error vectors

Matrix of factor loading Λ for personality test data

variable	factor 1	factor 2	factor 3	factor 4
distant	0.59	0.27	0	0
talkative	-0.50	-0.51	0	0.27
careless	0.46	-0.47	0.11	0.14
hardworking	-0.46	0.33	-0.14	0.35
kind	-0.488	0.222	0	0
⋮				

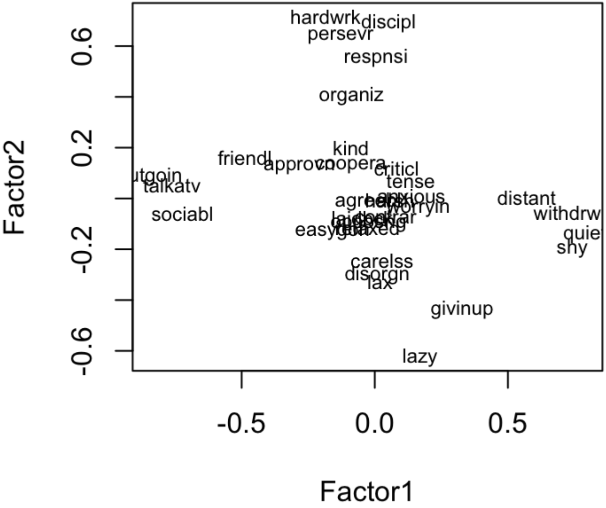
Factor Analysis: Example

Visualize loading of the first two factors



Factor Analysis: Example

Visualize loading of the first two factors, rotated to align with axes



Factor Analysis Model

Observed variables: $x \in \mathbb{R}^n$
Latent variables: $z \in \mathbb{R}^k$ $(k < n)$
The factor analysis model defines a joint distribution $p(x, z)$ as

$$
\begin{aligned}
z &\sim \mathcal{N}(0, I) \\
\epsilon &\sim \mathcal{N}(0, \Psi) \\
x &= \mu + \Lambda z + \epsilon
\end{aligned}
$$

where $\Psi \in \mathbb{R}^{n \times n}$ is a diagonal matrix, $\epsilon, \mu \in \mathbb{R}^n$, $\Lambda \in \mathbb{R}^{n \times k}$

Given observations $x^{(i)}, \ldots, x^{(m)}$, how to fit the parameters μ, Λ, Ψ ?

The EM Algorithm

Rubin, D. and Thayer, D. (1982). *EM algorithms for ML factor analysis*. Psychometrika, 47(1):69-76.

Listing 4: EM for Factor Analysis

Initialize μ, Λ, Ψ

Repeat untill convergence {

(E-step) For each i , set

$$Q_i(z^{(i)}) := p(z^{(i)}|x^{(i)}; \mu, \Lambda, \Psi) \leftarrow z \text{ is a continuous variable}$$

(M-step) Set

$$\mu, \Lambda, \Psi := \underset{\mu,\Lambda,\Psi}{\operatorname{argmax}} \sum_{i=1}^{m} \int_{z^{(i)}} Q_i(z^{(i)}) \log \frac{p(x^{(i)}, z^{(i)}; \mu, \Lambda, \Psi)}{Q_i(z^{(i)})} dz^{(i)} \quad (\star)$$

First, we need to write $p(z^{(i)}|x^{(i)})$ and $p(x^{(i)}, z^{(i)})$in terms of the model parameters.

EM Derivations

It can be shown that, random vector $\begin{bmatrix} z \\ x \end{bmatrix} \sim \mathcal{N}(\mu_{zx}, \Sigma)$ where

$\mu_{xz} = \begin{bmatrix} 0 \\ \mu \end{bmatrix}$ and $\Sigma = \begin{bmatrix} I & \Lambda^T \\ \Lambda & \Lambda\Lambda^T + \Psi \end{bmatrix}$

E-Step

The posterior distribution $z^{(i)}|x^{(i)} \sim \mathcal{N}\left(\mu_{z^{(i)}|x^{(i)}}, \Sigma_{z^{(i)}|x^{(i)}}\right)$

$$\mu_{z^{(i)}|x^{(i)}} = \Lambda^T(\Lambda\Lambda^T + \Psi)^{-1}(x^{(i)} - \mu)$$
$$\Sigma_{z^{(i)}|x^{(i)}} = I - \Lambda^T(\Lambda\Lambda^T + \Psi)^{-1}\Lambda$$

$$\begin{aligned} Q_i(z^{(i)}) &= p(z^{(i)}|x^{(i)}; \mu, \Lambda, \Psi) \\ &= \frac{1}{\sqrt{(2\pi)^k |\Sigma_{z^{(i)}|x^{(i)}}|}} \exp\left(-\frac{1}{2}(z^{(i)} - \mu_{z^{(i)}|x^{(i)}})^T \Sigma^{-1}_{z^{(i)}|x^{(i)}} (z^{(i)} - \mu_{z^{(i)}|x^{(i)}})\right) \end{aligned}$$

EM Derivations

M-Step

$$\underset{\mu,\Lambda,\Psi}{\operatorname{argmax}} \sum_{i=1}^{m} \int_{z^{(i)}} Q_i(z^{(i)}) \log \frac{p(x^{(i)}, z^{(i)}; \mu, \Lambda, \Psi)}{Q_i(z^{(i)})} dz^{(i)} \qquad (\star)$$

Note that

$$\int_{z^{(i)}} Q_i(z^{(i)}) \log \frac{p(x^{(i)}, z^{(i)}; \mu, \Lambda, \Psi)}{Q_i(z^{(i)})} dz^{(i)}$$
$$= \mathbb{E}_{z \sim Q_i}[\log p(x^{(i)}|z^{(i)}; \mu, \Lambda, \Psi) + \log p(z^{(i)}) - \log Q_i(z^{(i)})]$$

$(\star)$ is equivalent to

$$\underset{\mu,\Lambda,\Psi}{\operatorname{argmax}} \sum_{i=1}^{m} \mathbb{E}_{z^{(i)} \sim Q_i}[\log p(x^{(i)}|z^{(i)}; \mu, \Lambda, \Psi)]$$

EM Derivations

M-Step (con't)

$$\underset{\mu,\Lambda,\Psi}{\operatorname{argmax}} \sum_{i=1}^{m} \mathbb{E}_{z^{(i)}\sim Q_i}[\log p(x^{(i)}|z^{(i)};\mu,\Lambda,\Psi)] \quad (\star\star)$$

Since $x = \mu + \Lambda z + \epsilon$ and $\epsilon \sim \mathcal{N}(0, \Psi)$

$$x^{(i)}|z^{(i)} \sim \mathcal{N}(\mu + \Lambda z, \Psi)$$

$$p(x^{(i)}|z^{(i)};\mu,\Lambda,\Psi)$$
$$= \frac{1}{(2\pi)^{n/2}|\Psi|^{1/2}} \exp\left(-\frac{1}{2}(x^{(i)} - \mu - \Lambda z^{(i)})^T \Psi^{-1}(x^{(i)} - \mu - \Lambda z^{(i)})\right)$$

We can maximize $(\star\star)$ with respect to μ, Λ and Ψ

Factor Analysis Discussions

Comparison with Mixture of Gaussians

- Mixture of Gaussians assumes sufficient data and relative few response variables. i.e. when $n \approx m$ or $n > m$, Σ is singular
- Factor Analysis works when $n > m$ by allowing model noise

Factor Analysis Discussions

Relationship to PCA

- Both PCA and factor analysis can find low dimensional latent subspace in data
- PCA is good for data reduction (reduce correlation among observed variables)
- Factor analysis is good for data exploration (find independent, common factors in observed variables)
- Factor analysis allows the noise to have an arbitrary diagonal covariance matrix, while PCA assumes the noise is spherical.

Additional readings

- Zoubin Ghahramani and Geoffrey E. Hinton, The EM Algorithm for Mixtures of Factor Analyzers, 1997

Final Project

Topics

- Use machine learning to solve a specific problem.
- Develop a machine learning method with better performance
- Theoretical or innovative problems

Timeline

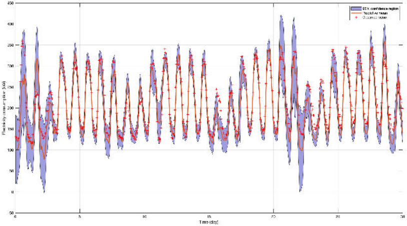
Dec 02	Confirm team
Dec 11	Submit project proposal
Dec 14-16	Meeting with course staff
Dec 27	Poster deadline
Dec 31	Poster presentation
Jan 13	Submit final report

Example Projects

Camera lens super-resolution (Dinjian Jin& Xiangyu Chen)

Comparison between two super-resolution models: SRGAN and VDSR *(application)*

A Gaussian Process Regression Based Approach for Predicting Building Cooling and Heating Consumption (Xiaoting Wang & Yiqian Wu)

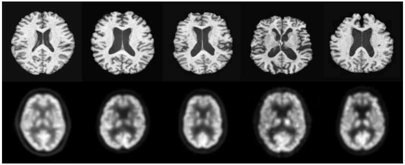


1-month prediction of electricity consumption *(application)*

Debugging Neural Networks (Riccardo Mattesini,Sebastian Beetschen,Bunchalit Eua-arporn)

Test neural network overfitting by feature visualization with GAN *(innovative problem)*

Missing Data Imputation for Multi-Modal Brain Images (Wangbin Sun)

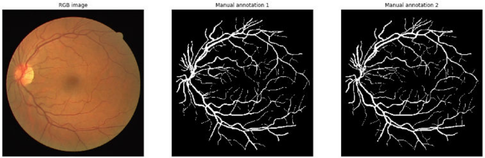


MRI (top) and PET (bottom) scans of normal and Alzheimer patient brains *(improved method)*

Sample Datasets & Ideas

Retina blood vessel segmentation

- How to combine geometric data processing with supervised segmentation? [DRIVE dataset]



Covid-19 projects

- Predict the effect of government policies on daily new cases in a given city/country. [Oxford Government Response Tracker] [Starter code]
- COVID-19 mRNA vaccine degradation prediction [OpenVaccine]

Multi-modal & transfer learning

- Human activity recognition through heterogeneous sensors
- Li-on battery classification through cross-domain measurements

Common Pitfalls

- Simply run a model from an existing work on a slightly different dataset: think of your **contribution** in at least one of the following areas: *application, analysis , methodology and theory* .
- Topic is too broad or too ambitious: reduce the project scope
- Project relies heavily on data availability/quality: use available datasets

Good luck!