Tensor Decomposition for Healthcare Analytics

UNIVERSITAT POLITÈCNICA DE CATALUNYA
BARCELONATECH

Matteo Ruffini

Laboratory for Relational Algorithmic, Complexity and Learning

matteo.ruffini@estudiant.upc.edu

November 5, 2017

Overview

Overview

Task: to segment patients in groups with similar clinical profiles.

1. Similar patients $\rightarrow$ Similar cares.
2. Find recurrent comorbidities.
3. Assigning and planning resources: drugs and doctors.

Overview

Task: to segment patients in groups with similar clinical profiles.

1. Similar patients $\rightarrow$ Similar cares.
2. Find recurrent comorbidities.
3. Assigning and planning resources: drugs and doctors.

Data: Electronic Healthcare Records (EHR).

Objective: Use these data to create clusters of patients.

Example: ICD-9 EHR

In ICD code, to each disease is associated a number

$$278 \rightarrow \mathit{Obesity}, \quad 401 \rightarrow \mathit{Hypertension}$$

Example: ICD-9 EHR

In ICD code, to each disease is associated a number

$$278 \to \textit{Obesity}, \quad 401 \to \textit{Hypertension}$$

Records: list of patients with their diseases → patient-disease matrix.

	Diseases
Patient 1	820, 401
Patient 2	401, 278,
Patient 3	560, 820, 278

	820	**401**	**278**	**560**
Patient 1	1	1	0	0
Patient 2	0	1	1	0
Patient 3	1	0	1	1

Objective: cluster the rows of the patient-disease matrix.

Sparse and high dimensional data.

Clustering

Clustering: one of the fundamental tasks of Machine Learning.

Objective: Dataset of N samples $\rightarrow$ partition in coherent subsets

Dataset: a matrix $X \in \mathbb{R}^{N \times n}$

$$X^{(i)} = (x_1^{(i)}, ..., x_n^{(i)})$$

Group together similar rows.

Standard methods: *k-means, k-medioids, single linkage*...

Distance-based: poor performances on high dimensional sparse data.

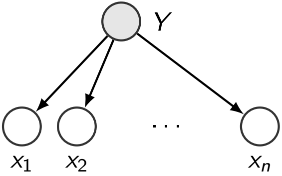
Mixture Models

Definition (Mixture Model)

$Y \in \{1, ..., k\}$ A latent discrete variable.
$X = (x_1, \dots, x_n)$ observable, depends on Y.

$$P(X) = \sum_{i=1}^{k} P(Y = i) P(X|Y = i)$$

x_i are called **features**.



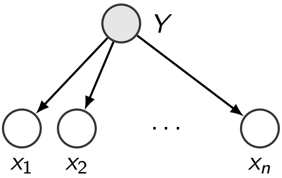
Mixture Models

Definition (Mixture Model)

$Y \in \{1, ..., k\}$ A latent discrete variable.
$X = (x_1, \dots, x_n)$ observable, depends on Y.

$$P(X) = \sum_{i=1}^{k} P(Y = i) P(X|Y = i)$$

x_i are called **features**.



Generative process for one sample:

1. Draw Y, obtain $Y = i \in \{1, ..., k\}$.
2. Draw $X \in \mathbb{R}^n \approx P(X|Y = i)$

Mixture Model Clustering

Clustering

From an outcome of X (observed) $\rightarrow$ Infer the outcome of Y (unknown)

$$k \ \ clusters.$$

Mixture Model Clustering

Clustering

From an outcome of X (observed) $\rightarrow$ Infer the outcome of Y (unknown)

k *clusters*.

Parameters characterizing a mixture model:

$$\omega_h := P(Y = h), \ \ \omega := (\omega_1, \dots, \omega_k)^\top, \ \ \Omega := diag(\omega).$$

$$\mu_{i,j} = E(x_i | Y = j), \quad M = (\mu_{i,j})_{i,j} = [\mu_1|, ..., |\mu_k] \in \mathbb{R}^{n \times k}$$

If conditional distributions and the model parameters are known:

$$P(Y = j | X, M, \omega) \propto P(X | Y = j, M) \omega_j$$

$$Cluster(X) = \underset{j=1,\dots,k}{\arg\max}\, P(Y = j | X, M, \omega)$$

It is crucial to know the parameters of the model (M, ω).

Mixture of Independent Bernoulli

Observables are binary and conditionally independent: $x_i \in \{0, 1\}$.

The expectations coincide with the probability of a positive outcome.

$$\mu_{i,j} = P(x_i = 1 | Y = j).$$

$$P(Y = j|X) \propto \omega_j \prod_{i=1}^{n} \mu_{i,j}^{x_i} (1 - \mu_{i,j})^{1-x_i}$$

Clustering Rule:

$$Cluster(X) = \arg\max_{j=1,\dots,k} \omega_j \prod_{i=1}^{n} \mu_{i,j}^{x_i} (1 - \mu_{i,j})^{1-x_i}$$

Mixture Model Clustering: sum up

Advantages:

- Robust to irrelevant features:

$$P(x_i) = P(x_i | Y = j)$$

- Algorithms with provable guarantees of optimality.

Mixture Model Clustering: sum up

Advantages:

- Robust to irrelevant features:

$$P(x_i) = P(x_i | Y = j)$$

- Algorithms with provable guarantees of optimality.

Disadvantages:

- Model assumption on the reality.

Mixture Model Clustering: sum up

Advantages:

- Robust to irrelevant features:

$$P(x_i) = P(x_i|Y = j)$$

- Algorithms with provable guarantees of optimality.

Disadvantages:

- Model assumption on the reality.

To sum up: Two steps:

1. Estimate the parameters of the mixture.
2. Group together similar elements, using Bayes' theorem.

Learning mixture parameters

Maximum Likelihood Estimate

Standard method Maximum Likelihood.

Find parameters $\Theta = (M, \omega)$ maximizing the likelihood on $X \in \mathbb{R}^{N \times n}$

$$\max_{\Theta} P(X, \Theta) = \max_{\Theta} \prod_{i=1}^{N} \sum_{j=1}^{k} P(X^{(i)} | Y = j, M) \omega_j$$

Maximizing this is hard

In general there are no closed form solutions.

Expectation Maximization (EM)

Iterative algorithm from [Dempster et al.(1977)]

1. Randomly initialize (M, ω)
2. Cluster the samples.
3. Use the clusters to recalculate (M, ω).
4. Iterate over steps 2 and 3 until convergence.

Expectation Maximization (EM)

Iterative algorithm from [Dempster et al.(1977)]

1. Randomly initialize (M, ω)
2. Cluster the samples.
3. Use the clusters to recalculate (M, ω).
4. Iterate over steps 2 and 3 until convergence.

Pro and cons

- Iteratively increases the likelihood.
- No guarantees of reaching global optimum.
- EM is slow.
- The quality of the results depends on the initialization:

$$\text{Good starting points} \rightarrow \text{Good outputs}$$

A general approach, outlined in [Anandkumar et al., 2014].

Alternative Approach: Tensor Decomposition

A general approach, outlined in [Anandkumar et al., 2014].

1. Estimate (Recall: $M = [\mu_1|, ..., |\mu_k]$, $\mu_i = E[X|Y = i] \in \mathbb{R}^n$).

$$
\begin{gathered}
M_1 := M\,\omega \in \mathbb{R}^n \\
M_2 := M\,diag(\omega)\,M^\top \in \mathbb{R}^{n\times n}, \\
M_3 := \textstyle\sum_{i=1}^k \omega_i \mu_i \otimes \mu_i \otimes \mu_i \in \mathbb{R}^{n\times n\times n}
\end{gathered}
$$

Alternative Approach: Tensor Decomposition

A general approach, outlined in [Anandkumar et al., 2014].

1. Estimate (Recall: $M = [\mu_1|, ..., |\mu_k]$, $\mu_i = E[X|Y = i] \in \mathbb{R}^n$).

$$M_1 := M\,\omega \in \mathbb{R}^n$$
$$M_2 := M\,diag(\omega)\,M^\top \in \mathbb{R}^{n\times n},$$
$$M_3 := \sum_{i=1}^{k} \omega_i \mu_i \otimes \mu_i \otimes \mu_i \in \mathbb{R}^{n\times n\times n}$$

2. Retrieve (M, ω) with a tensor decomposition algorithm $\mathcal{A}$:

$$\mathcal{A}(M_1, M_2, M_3) \to (M, \omega)$$

Alternative Approach: Tensor Decomposition

A general approach, outlined in [Anandkumar et al., 2014].

1. Estimate (Recall: $M = [\mu_1|, ..., |\mu_k]$, $\mu_i = E[X|Y = i] \in \mathbb{R}^n$).

$$M_1 := M\,\omega \in \mathbb{R}^n$$
$$M_2 := M\,diag(\omega)\,M^\top \in \mathbb{R}^{n\times n},$$
$$M_3 := \sum_{i=1}^k \omega_i \mu_i \otimes \mu_i \otimes \mu_i \in \mathbb{R}^{n\times n\times n}$$

2. Retrieve (M, ω) with a tensor decomposition algorithm $\mathcal{A}$:

$$\mathcal{A}(M_1, M_2, M_3) \to (M, \omega)$$

- **Step 1:** Depends on the specific properties of the mixture.
- **Step 2**: Is general (need assumptions on M).

Example: Mixture of Independent Gaussians

Dataset $X \in \mathbb{R}^{N \times n}$ with iid rows $X^{(i)} = (x_1^{(i)}, ..., x_n^{(i)})$.

Model settings:

- $x_h^{(i)}$ and $x_l^{(i)}$ are conditionally independent $\forall h \neq l$.
- $x_h^{(i)}$ conditioned to Y is a Gaussian, with *known* stdev σ:

$$P(x_h | Y = i) \approx \mathcal{N}(\mu_{h,i}, \sigma)$$

Example: Mixture of Independent Gaussians

Dataset $X \in \mathbb{R}^{N\times n}$ with iid rows $X^{(i)} = (x_1^{(i)}, ..., x_n^{(i)})$.

Model settings:

- $x_h^{(i)}$ and $x_l^{(i)}$ are conditionally independent $\forall h \neq l$.
- $x_h^{(i)}$ conditioned to Y is a Gaussian, with *known* stdev σ:

$$P(x_h|Y = i) \approx \mathcal{N}(\mu_{h,i}, \sigma)$$

Theorem ([Hsu et al. 2013])

Define the following three quantities:

$$\tilde{M}_1 = \textstyle\sum_{i=1}^N \frac{X^{(i)}}{N}, \qquad \tilde{M}_2 = \frac{X^\top X}{N} - \sigma^2 \mathbb{I}_n$$

$$\tilde{M}_3 \quad = \textstyle\sum_{i=1}^N \frac{X^{(i)}\otimes X^{(i)}\otimes X^{(i)}}{N} - \sigma^2 \sum_{i=1}^n (\tilde{M}_1 \otimes e_i \otimes e_i + e_i \otimes \tilde{M}_1 \otimes e_i + e_i \otimes e_i \otimes \tilde{M}_1)$$

Then $\lim_{N\to\infty} \tilde{M}_i = M_i \ \forall i \in \{1, 2, 3\}$

Other estimation procedures.

Similar (but more technical) procedures for many Mixture Models:

- Mixture of multinomial distributions (Single Topic Model) [Ruffini, Casanellas, Gavaldà (2017)]
- Naive Bayes Models [Anandkumar et al. 2012].

This estimation procedure can be generalized to other latent variable models (like Hidden Markov Models, Latent Dirichlet Allocation...)

Given estimated $\tilde{M}_1, \tilde{M}_2$ and $\tilde{M}_3$ we feed an algorithm $\mathcal{A}$ to recover estimated $(\tilde{M}, \tilde{\omega})$

A Tensor Decomposition Algorithm: SVTD

An algorithm $\mathcal{A}$

$$\mathcal{A}(M_1, M_2, M_3, k) \to (M, \omega)$$

Assumptions:

- The centers are linearly independent (M has rank $k \leq n$)
- At least one feature has different conditional expectations:

$$E(x_i|Y = j) \neq E(x_i|Y = h) \ \forall i \neq h$$

A Tensor Decomposition Algorithm: SVTD

Observations (recall: $\Omega = diag(\omega)$)

1. $M_2 = M \Omega M^\top$ by definition.

A Tensor Decomposition Algorithm: SVTD

Observations (recall: $\Omega = diag(\omega)$)

1. $M_2 = M\Omega M^\top$ by definition.
2. Also $M_2 = U_k S_k U_k^\top = E_k E_k^\top$ with a SVD. Then

$$M\Omega^{\frac{1}{2}} = E_k O, \quad \text{for some } O : \ OO^\top = \mathbb{I}_k$$

A Tensor Decomposition Algorithm: SVTD

Observations (recall: $\Omega = diag(\omega)$)

1. $M_2 = M\Omega M^\top$ by definition.
2. Also $M_2 = U_k S_k U_k^\top = E_k E_k^\top$ with a SVD. Then

$$M\Omega^{\frac{1}{2}} = E_k O, \quad \text{for some } O: \; OO^\top = \mathbb{I}_k$$

3. $M_3 := \sum_{i=1}^k \omega_i \mu_i \otimes \mu_i \otimes \mu_i$ so its $r - th$ slice is:

$$M_{3,r} = M\Omega^{\frac{1}{2}} diag((\mu_{r,1}, ..., \mu_{r,k}))\Omega^{\frac{1}{2}} M^\top$$

A Tensor Decomposition Algorithm: SVTD

Observations (recall: $\Omega = diag(\omega)$)

1. $M_2 = M\Omega M^\top$ by definition.
2. Also $M_2 = U_k S_k U_k^\top = E_k E_k^\top$ with a SVD. Then

$$M\Omega^{\frac{1}{2}} = E_k O, \quad \text{for some } O : \; OO^\top = \mathbb{I}_k$$

3. $M_3 := \sum_{i=1}^k \omega_i \mu_i \otimes \mu_i \otimes \mu_i$ so its $r - th$ slice is:

$$M_{3,r} = M\Omega^{\frac{1}{2}} diag((\mu_{r,1}, ..., \mu_{r,k}))\Omega^{\frac{1}{2}} M^\top$$

4. For each r

$$H_r = E_k^\dagger M_{3,r} (E_k^\top)^\dagger = Odiag((\mu_{r,1}, ..., \mu_{r,k}))O^\top$$

A Tensor Decomposition Algorithm: SVTD

Observations (recall: $\Omega = diag(\omega)$)

1. $M_2 = M \Omega M^\top$ by definition.
2. Also $M_2 = U_k S_k U_k^\top = E_k E_k^\top$ with a SVD. Then

$$M\Omega^{\frac{1}{2}} = E_k O, \quad \text{for some } O: \; OO^\top = \mathbb{I}_k$$

3. $M_3 := \sum_{i=1}^{k} \omega_i \mu_i \otimes \mu_i \otimes \mu_i$ so its $r-th$ slice is:

$$M_{3,r} = M\Omega^{\frac{1}{2}} diag((\mu_{r,1}, ..., \mu_{r,k}))\Omega^{\frac{1}{2}} M^\top$$

4. For each r

$$H_r = E_k^\dagger M_{3,r} (E_k^\top)^\dagger = O diag((\mu_{r,1}, ..., \mu_{r,k}))O^\top$$

5. The singular values of H_r are the $r-th$ row of M.

A Tensor Decomposition Algorithm: SVTD

Algorithm

1. Take as input (M_1, M_2, M_3, k)
2. Decompose M_2 as $M_2 = E_k E_k^\top$
3. Calculate
$$H_r = E_k^\dagger M_{3,r} (E_k^\top)^\dagger$$
4. Recover the $r-th$ row of M as the Singular Values of H_r.
5. Recover ω solving $M\omega = M_1$

Reference: A new Spectral Method for Latent Variable Models [Ruffini, Casanellas, Gavaldà (2017)]

SVTD - Perturbation Theorem

$$SVTD(M_1, M_2, M_3, k) \rightarrow (M, \omega)$$

Small perturbations on the input $\rightarrow$ Small perturbations on the output.

$$SVTD(\tilde{M}_1, \tilde{M}_2, \tilde{M}_3, k) \rightarrow (\tilde{M}, \tilde{\omega})$$

SVTD - Perturbation Theorem

$$SVTD(M_1, M_2, M_3, k) \rightarrow (M, \omega)$$

Small perturbations on the input $\rightarrow$ Small perturbations on the output.

$$SVTD(\tilde{M}_1, \tilde{M}_2, \tilde{M}_3, k) \rightarrow (\tilde{M}, \tilde{\omega})$$

Theorem ([Ruffini, Casanellas, Gavaldà (2017)])

If $||\tilde{M}_2 - M_2||_F < \epsilon, ||\tilde{M}_3 - M_3||_F < \epsilon$, *then, for sufficiently small* ϵ,

$$||M - \tilde{M}||_F \leq C_1\epsilon + O(C_2\epsilon^2)$$

for some C_1 *and* C_2 *depending on the model parameters.*

Putting it all together

1. Take a dataset X, with rows sampled from a given mixture model.
2. Estimate the moment tensors $\tilde{M}_1, \tilde{M}_2, \tilde{M}_3$.
3. Retrieve estimated $(\tilde{M}, \tilde{\omega})$.
4. Optionally, use EM to improve the obtained $(\tilde{M}, \tilde{\omega})$.
5. Use Bayes' theorem to cluster the rows of X.

Case Study: mixture of independent Bernoulli

In some cases we don't know how to directly estimate $\tilde{M}_1, \tilde{M}_2, \tilde{M}_3$

X a dataset with rows $\approx$ mixture of independent Bernoulli.

Open Problem: how to efficiently estimate $\tilde{M}_2, \tilde{M}_3$?

$$X \underbrace{\rightarrow}_{?} \tilde{M}_2, \tilde{M}_3$$

(the issue are the diagonal entries...)

A work-around: the three views trick (Idea)

Split the observables in three views:

$$X^{(i)} = (\underbrace{x_1^{(i)}, \dots, x_{d_a}^{(i)}}_{X_a^{(i)}}, \underbrace{x_{d_a+1}^{(i)}, \dots, x_{d_a+d_b}^{(i)}}_{X_b^{(i)}}, \underbrace{x_{d_a+d_b+1}^{(i)}, \dots, x_{d_a+d_b+d_c}^{(i)}}_{X_c^{(i)}}), \quad M = \begin{pmatrix} M_a \\ M_b \\ M_c \end{pmatrix}$$

A work-around: the three views trick (Idea)

Split the observables in three views:

$$X^{(i)} = (\underbrace{x_1^{(i)}, \ldots, x_{d_a}^{(i)}}_{x_a^{(i)}}, \underbrace{x_{d_a+1}^{(i)}, \ldots, x_{d_a+d_b}^{(i)}}_{x_b^{(i)}}, \underbrace{x_{d_a+d_b+1}^{(i)}, \ldots, x_{d_a+d_b+d_c}^{(i)}}_{x_c^{(i)}}), \quad M = \begin{pmatrix} M_a \\ M_b \\ M_c \end{pmatrix}$$

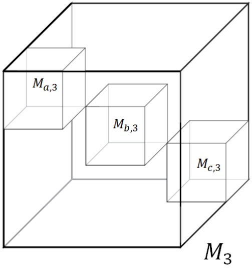
Estimate subtensors of $\tilde{M}_2, \tilde{M}_3$. [Anandkumar et al., 2014]

For $j = a, b, c$

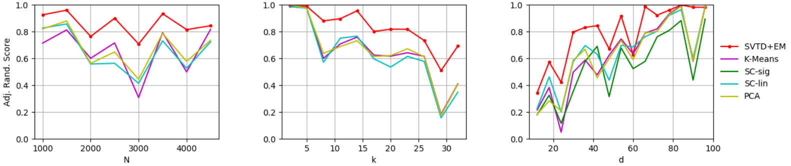
$$M_{j,2} := M_j \, diag(\omega) \, M_j^\top ,$$
$$M_{j,3} := \sum_{i=1}^k \omega_i \mu_i^{(j)} \otimes \mu_i^{(j)} \otimes \mu_i^{(j)}$$

Decompose them to get M_j

Get M by concatenation.



Experiments - Mixture of independent Bernoulli



Experiment: generate a synthetic dataset and cluster its rows with SVTD and with k-means, spectral clustering and PCA clustering.

Accuracy metric: Adjusted Rand Index. It is 1 if the clustering is perfect, 0 if it is bad (like random labeling).

Clustering Patients with Tensor Decomposition

[Ruffini, Gavaldà, Limón (2017)]

The data

Source: Servei Català de la Salut.

Dataset Diagnostics of patients admitted to the hospitals in 2016.

Data format Each row is a visit: up to 10 diagnostics in ICD-9 format.

Two subset datasets

1. Patients having diagnostic 428 in the ICD-9 code (Heart Failure).
2. Patients with serious diseases to be treated it top hospitals.

Objective:

1. To create meaningful clusters of patients in each dataset.
2. To visualize the key characteristics of each cluster.

Modeling strategy

Convert each dataset in a patient-disease matrix:

	Disease 1	**Disease 2**	**Disease 3**	...
Patient 1	1	1	0	...
Patient 2	0	1	1	...
Patient 3	1	0	1	...
...	...	...	...	...

Modeling strategy

Convert each dataset in a patient-disease matrix:

	Disease 1	**Disease 2**	**Disease 3**	...
Patient 1	1	1	0	...
Patient 2	0	1	1	...
Patient 3	1	0	1	...
...	...	...	...	...

Data are modeled as mixture of independent Bernoulli variables

- Latent state $\rightarrow$ Medical status of a patient.
- Observed diseases depend on the patient status.
- Once in a status, diagnostics are independent.

Modeling strategy

We have a mixture of independent Bernoulli:

1. Recover $(\tilde{M}, \tilde{\omega})$.
2. Improve the estimated $(\tilde{M}, \tilde{\omega})$ with EM.
3. Cluster the rows of X into k clusters.
4. Plot the results.

The number of clusters is manually set as an external parameter, (from expert's considerations).

Details of the Datasets

Heart Failure Dataset

- $N = 23082$ (23082 individual patient records).
- All the patients in the dataset have a Heart Failure as a diagnostic.
- $k = 5$ clusters.

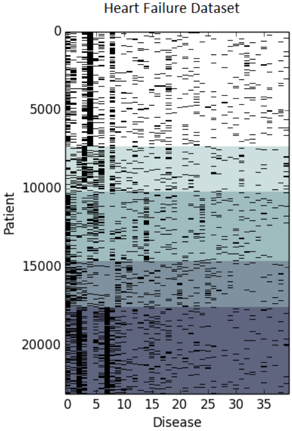
"Tertiary" Dataset

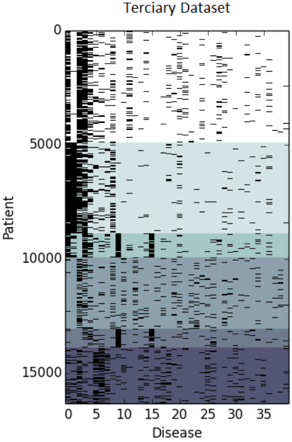
- $N = 16311$ individual patient records.
- $k = 6$ clusters.

In both cases $n = 696$ registered diagnostics (columns of the datasets).

Visual patterns: Heat-maps

Heat-maps of the two datasets:





- Black dots: diagnostics
- Background color: clusters

Relevance

Heat-maps $\rightarrow$ patterns in the clusters.

What there is inside the patterns?

Find the **relevant** diagnostics for each cluster.

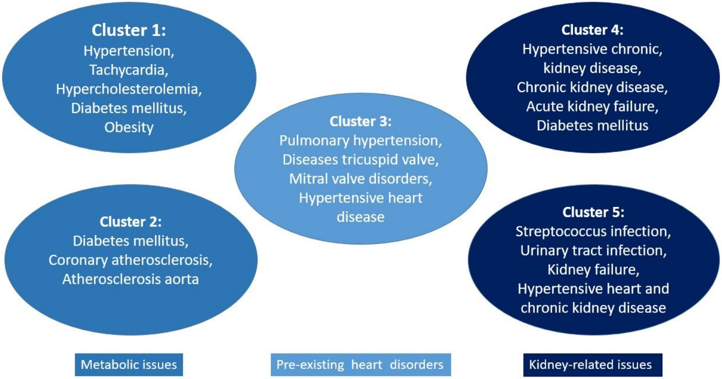
Relevance:

$$relevance(i,j) = \lambda \log(\mu_{i,j}) + (1-\lambda) \log\left(\frac{\mu_{i,j}}{\sum_{h=1}^{k} \mu_{i,h}\omega_h}\right)$$

where $\mu_{i,j} = P(x_i = 1 | Y = j)$.

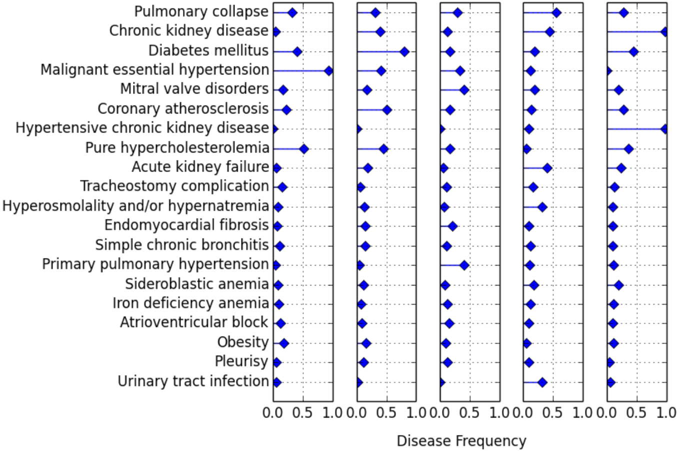
- High relevance: more frequent in a cluster than in the full dataset.
- Low relevance: low/high frequency everywhere.

Heart Failure Dataset - Content of the clusters

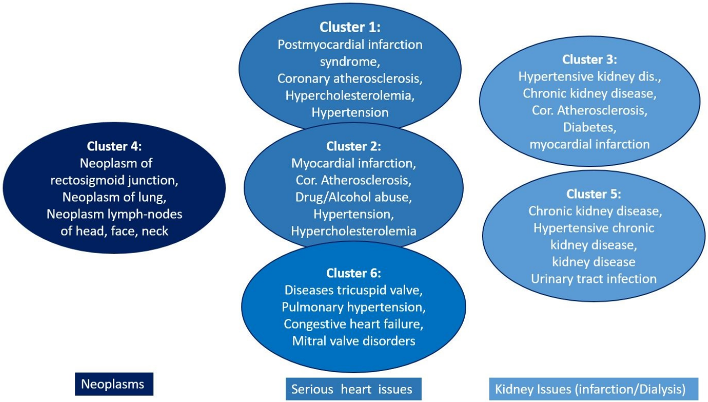


Cluster ID:	1	2	3	4	5
Size:	7290	2915	4408	2936	5533

Heart Failure Dataset – disease-frequency chart

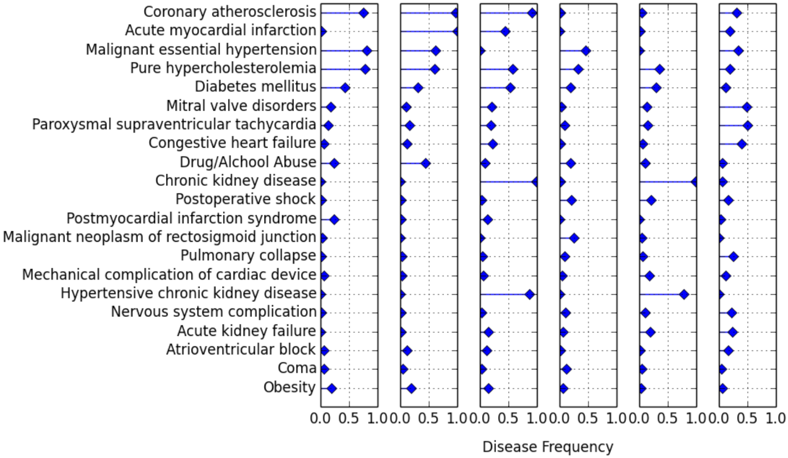


"Tertiary" Dataset - Content of the clusters



Cluster ID:	1	2	3	4	5	6
Size:	4892	3982	1043	3133	819	2442

"Tertiary" Dataset– disease-frequency chart



Tensor Decomposition for Healthcare Analytics

Matteo Ruffini

Laboratory for Relational Algorithmic, Complexity and Learning

matteo.ruffini@estudiant.upc.edu

November 5, 2017