# CAIM, examen final

**19 de gener de 2021. Temps: 2 hores 50 minuts**

*La competència treball autònom s'avaluarà a partir de la nota de l'exercici del tema 9 (Recommender Systems).*

## Exercici 1

(1 punt) In the context of index compression,

1. What is a continuation bit?
2. What is the aim (or aims) of a continuation bit as a technique to code for a sequence of integers?
3. Give the name of other techniques with the same or similar aim(s)?

### Solució de l'exercici 1

1. The continuation bit is a technique of variable length coding that consists of coding a number using the last bit of a byte (or nibble) to indicate if after a byte (or nibble) there is another byte (or nibble) corresponding to that number.
2. The aims are to achieve (a) variable length coding by adapting the length of the code the number of significant digits of the number being coded and (b) unique segmentation of the sequence of numbers.
3. Unary self-delimiting codes and Elias-Gamma codes.

## Exercici 2

(2.5 punts) We define $N(f)$ as the number of types that have frequency $f$ or greater in a text of $T$ tokens and $V$ types. Suppose that $N(f)$ follows a power law, i.e.

$$N(f) = cf^{-\beta}, \tag{1}$$

where $c$ is some factor and $\beta$ is the exponent of the law. In English, $\beta \approx 1$. We assume that $\beta > 0$.

1. What is the value of $c$?
2. What is the number of types that have frequency 2 or greater?
3. What is the number of types that have frequency 2 or greater when $\beta = 1$?
4. What is $n(1)$, the number of types that have frequency 1?
5. What is $n(1)$ when $\beta = 1$?
6. What is $n(f)$, the number of types that have frequency $f$?
7. Show that
$$n(f) \approx c' f^{-\beta'}$$
for sufficiently large $f$.
8. What are $c'$ and $\beta'$?

### Solució de l'exercici 2

1. Let $V$ be the number of types in a text. By definition, $N(1) = V$. Hence $c = V$.
2. $N(2) = V2^{-\beta}$.
3. $N(2) = \frac{1}{2}V$.

4. $n(1) = V - N(2) = V(1 - 2^{-\beta})$.
5. $n(1) = V(1 - 2^{-1}) = \frac{1}{2}V$.
6. $$\begin{aligned} n(f) &= N(f) - N(f+1) \\ &= c(f^{-\beta} - (f+1)^{-\beta}) \\ &= V(f^{-\beta} - (f+1)^{-\beta}) \end{aligned} \quad (2)$$

   Since $-(f^{-\beta} - (f+1)^{-\beta})$ is the slope of the line joining the points $(f, f^{-\beta})$ and $(f+1, (f+1)^{-\beta})$,

   $$\begin{aligned} f^{-\beta} - (f+1)^{-\beta} &\approx -\frac{df^{-\beta}}{df} \\ &= (\beta+1)f^{-\beta-1}. \end{aligned}$$

   Hence Eq. 2 becomes

   $$n(f) = (\beta+1)Vf^{-\beta-1}.$$
7. $c' = (\beta+1)V$ and $\beta' = \beta + 1$.

## Exercici 3

(1.5 punts) Suppose an undirected graph of $n$ vertices, $m$ edges and adjacency matrix $A = \{a_{ij}\}$. In that graph, $a_{ij} = a_{ji}$ for each $1 \leq i, j \leq n$. Suppose that the PageRank weights of that graph are

$$\vec{p} = c\vec{k}, \quad (3)$$

where $\vec{p}$ and $\vec{k}$ are column vectors, $\vec{k}^T = (k_1, ..., k_i, ..., k_n)$, $k_i$ is the degree of the $i$-th vertex and $c$ is some factor.

1. What is the value of $c$?
2. Is it true that $M^T\vec{p} = \vec{p}$? Justify your answer (by providing a proof or a counterexample).

### Solució de l'exercici 3

1. The fact that

   $$\sum_{i=1}^{n} p_i = 1$$

   transforms Eq. 3 into

   $$c = \frac{1}{\sum_{i=1}^{n} k_i}.$$

   By the handshaking lemma, i.e.

   $$\sum_{i=1}^{n} k_i = 2m,$$

   we obtain $c = \frac{1}{2m}$.
2. A proof (valid even when $\vec{p}$ is not actually a vector of PageRanks)

   $$\begin{aligned} M^T\vec{p} &= M\vec{p} \\ &= \frac{1}{2m}M\vec{k} \\ &= \frac{1}{2m}\begin{pmatrix} \cdots \\ \sum_{i=1}^{n} \frac{a_{ij}}{k_i} k_i \\ \cdots \end{pmatrix} \\ &= \frac{1}{2m}\begin{pmatrix} \cdots \\ k_i \\ \cdots \end{pmatrix} \\ &= \frac{1}{2m}\vec{k} \\ &= \vec{p} \end{aligned}$$

*Comments.* The fact that $\vec{p}$ is a vector of PageRanks means that $\vec{p}$ is the only non-trivial vector such that

$$\left(\lambda M^T + (1-\lambda)\frac{J}{n}\right)\vec{p} = \vec{p}.$$

The fact that that the previous equation holds (with $\lambda < 1$) does not imply, tentatively, that $M^T\vec{p} = \vec{p}$ also holds. We know that if $M$ is stochastic, there is at least one vector $\vec{p'}$ that satisfies $M^T\vec{p'} = \vec{p'}$ but we do not know if $\vec{p}$, as defined above, is one of these vectors (i.e. a vector that satisfies $M^T\vec{p} = \vec{p}$). We have to prove it.

## Exercici 4

(1 punt) We have designed a sophisticated recommender system based on collaborative filtering and we would like to evaluate its performance by comparing it against a simple system that does not employ collaborative filtering. We wish to make sure that the sophistication of our system is worth.

1. What system would you use for comparison?
2. Why?

### Solució de l'exercici 4

1. The baseline would recommend the most frequently selected items.
2. Because that neglects any information about user past choices.

## Exercici 5

(2 punts) Consider the set of edges of a large undirected graph consisting of unordered pairs of the form $(u, v)$, where $u$ and $v$ are integers that stand for two distinct vertices. In this setting, we wish to solve various problems that take the set of edges as input applying the MapReduce programming paradigm.

1. Calculate the degree of every vertex using just one job. The output of the *reduce* functions must be pairs of the form $(v, k)$, where $k$ is the degree of $v$.
2. Calculate the so-called *average degree of nearest neighbours* of each vertex. Given a vertex $i$, such average is

$$k_{nn}(i) = \frac{1}{k_i}\sum_{j=1}^{n} a_{ij}k_j, \tag{4}$$

where $k_j$ is the degree of the $j$-th vertex and $a_{ij} = 1$ if vertices $i$ and $j$ are connected (otherwise $a_{ij} = 0$). Split the programming into two jobs

- The first job has to compute the set of nearest neighbours of a vertex $v$. The output of the *reduce* functions are pairs of the form $(v, \Gamma)$, where $\Gamma$ is the set of vertices adjacent to $v$.
- The second job takes the output of the 1st job as input of the *map* functions. The output of the *reduce* functions are pairs of the form $(v, k_{nn}(v))$.

Please provide the pseudocode of *map*, *reduce* (and optionally *combine*) functions. Solutions are expected to be simple and efficient.

### Solució de l'exercici 5

1.
```
map(u, v)
  output (u, 1)
  output (v, 1)

reduce(v, L) = combine(v, L)
  output (v, sum(L))
```

Solutions without *combine* function are less efficient.

2. First job:

```
map(u, v)
  output (u, v)
  output (v, u)

reduce(v, L)
  output (v, L)
```

Second job (G is the set of vertices adjacent to a certain vertex):

```
map(v, G)
  for each vertex u in G
     output (u, size of G)

combine(v, L)
  output (u, [sum(L), size(L)])

reduce(v, L)
  s is the sum of 1st elements of pairs in L
  k is the sum of 2nd elements of pairs in L // k is actually the degree of v
  output (v, s/k)
```

A less efficient version without *combine*

```
map(v, G)
  for each vertex u in G
     output (u, size of G)

reduce(v, L)
  output (v, sum(L)/size(L))
```

## Exercici 6

(2 punts) The average shortest path length of an undirected graph of $n$ vertices is defined as

$$l = \frac{1}{\binom{n}{2}} \sum_{i<j} d_{ij}. \tag{5}$$

Suppose a Watts-Strogatz model with parameters $n$ (number of vertices), $p$ (rewiring probability) and $K$ (the mean vertex degree).

1. Estimate (approximately) the value of $l$ for $p = 1$ and sufficiently large $n$?
2. If you wished to use an Erdős-Rényi graph, $G_{n,\pi}$ as a control (or baseline) for a Watts-Strogatz model, which value would you use to set the parameter $\pi$?
3. Calculate $l$ exactly as a function of $n$ when $K = 2$ and $p = 0$.

### Solució de l'exercici 6

1. When $p = 1$ the Watts-Strogatz model is equivalent to an Erdős-Rényi graph, where $l \approx \frac{\log n}{\log z}$, where $z$ is the average degree. As $z = K$, $l \approx \frac{\log n}{\log K}$.
2. $\pi$ has to be the density of links, i.e.

$$\pi = \frac{Kn/2}{\binom{n}{2}} = \frac{K}{n-1}.$$

3. Since distances are symmetric ($d_{ij} = d_{ji}$), $l$ can be expressed as

$$l = \frac{1}{n(n-1)} \sum_{i=1}^{n} D_i,$$

where

$$D_i = \sum_{j=1}^{n} d_{ij}.$$

In a regular lattice, $D_i$ is the same for every vertex and then

$$l = \frac{1}{n-1} D_1.$$

When $n$ is odd, minimum vertex-vertex distances range between 1 and $(n-1)/2$,

$$\begin{aligned} D_1 &= 2 \sum_{\delta=1}^{(n-1)/2} \delta \\ &= \frac{1}{4}(n-1)(n+1) \end{aligned}$$

and then

$$l = \frac{n+1}{4}$$

for $n \geq 2$. When $n$ is even, minimum vertex-vertex distances range between 1 and $(n-1)/2$ and

$$\begin{aligned} D_1 &= 2 \sum_{\delta=1}^{n/2-1} \delta + n/2 \\ &= \frac{n^2}{4} \end{aligned}$$

and then

$$l = \frac{n^2}{4(n-1)}$$

for $n \geq 2$. All together,

$$l = \frac{(n-x)(n+x)}{4(n-1)}$$

where $x = 1 - n \bmod 2$.