Universitat de Girona
Escola Politècnica Superior

GEINF - GDDV - GEB

Pràctica 1: Xarxes de donants

Intel·ligència Artificial

Curs 2024/25

Universitat de Girona
Departament d'Informàtica, Matemàtica Aplicada i Estadística

Jerónimo Hernández González
jeronimo.hernandez@udg.edu

(PDF creat el 25 de febrer de 2025)

Índex

1 Presentació

S'està desenvolupant a nivell internacional un programa per coordinar donacions inter-hospitalàries. L'òrgan d'un donant a un hospital grec pot acabar salvant una vida a un pacient d'un hospital alemany.

Des del programa s'ha proposat formar diferents xarxes de donació (grups d'hospitals que es coordinen entre ells) que puguin funcionar de manera independent, per guanyar agilitat i eficiència.

Se'ns ha encarregat la feina de crear les xarxes. Disposem del llistat d'hospitals, les seves localitzacions (coordenades x,y) i dades sobre alguns altres descriptors. Nosaltres, fent servir aquesta informació, haurem de buscar la millor distribució d'hospitals entre xarxes.

1.1 Especificacions

- Un total de **40 hospitals** formen part del programa.
- Es volen formar **4 xarxes de donació** exclusives: cada hospital forma part d'una xarxa, i només d'una.
- Es farà servir la **distància euclidiana** per calcular la distància entre hospitals.

  Disposeu del mètode `distancia(p1,p2)` ja implementat.
- Es farà servir una **xarxa Bayesiana** (XB) per modelar la similitud entre la població objectiu dels hospitals, de tal manera que ens pugui interessar promoure que dos hospitals amb característiques semblants estiguin a la mateixa xarxa tot i que no estiguin *tan a prop*.

  Disposeu de les classes necessàries per fer servir XBs ja implementades.

2 Què heu de fer?

Heu d'implementar un algoritme de cerca que trobi una bona (l'òptima?) solució: una distribució d'hospitals en les 4 xarxes. Heu de combinar la distància euclidiana amb la similitud basada en XBs per guiar el procés de distribuir els hospitals en xarxes de donació.

Per completar aquesta pràctica, heu de fer aquests 4 pasos:

2.1 Pas 1: Representació del problema

Heu de definir com es representaran/codificaran els estats del problema:

1. Definiu una representació per al problema: com codifiqueu un estat?
2. Descriviu el conjunt d'estats possibles del problema.
3. Definiu un mètode que codifiqui la **funció objectiu**, és a dir, la funció que es vol minimitzar: la suma ponderada de les distàncies mitjanes per xarxa:

$$\min_{\sigma} \frac{1}{N} \sum_{g=1}^{G} N_g \cdot \overline{dist}(\sigma, g) \tag{1}$$

on la distància mitjana intra-xarxa és:

$$\overline{dist}(\sigma, g) = \frac{2}{N_g \cdot (N_g - 1)} \sum_{i=1}^{N_g - 1} \sum_{j=i}^{N_g} dist\left(\sigma(g)_i, \sigma(g)_j\right)$$

on $G = 4$ és el nombre de xarxes, $N = 40$ és el nombre d'hospitals, N_g és el nombre d'hospitals a la xarxa g, σ és una assignació d'hospitals en xarxes, i $\sigma(g)_i$ és l'i-èssim hospital de la xarxa g.

Implementeu aquestes idees en Python.

Teniu a Moodle (vegeu Sect. 2.4) un fitxer CSV amb les dades. De fet, hi ha dos versions d'aquest fitxer: inicialment necessitareu el que només porta les **coordenades 2D** (x,y) dels **40 hospitals**.

2.2 Pas 2: Cerca local

Heu d'implementar l'algoritme de cerca local per feixos (*beam local search*).

1. Definiu un mètode per crear un conjunt de B **estats inicials** (assignacions inicials aleatòries).
2. Definiu un mètode que, donat un estat, generi tots els seus **veïns**.

   Heu de **triar les accions** de transició (com transformar un estat actual per aconseguir-ne un de nou). Expliqueu i motiveu la vostra elecció.
3. Definiu un **mètode que avaluï** tots els veïns (de tots els estats del feix) d'acord amb la funció objectiu (Eq. 1) i els ordeni segons el seu valor.
4. Implementeu el mètode de **cerca local per feixos**.

   Es tracta d'un mètode iteratiu que crea un conjunt d'estats inicial i, a cada iteració, genera tots els veïns i es queda amb els B millors estats per a la següent iteració.

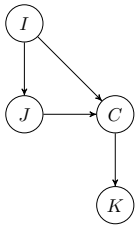
   - Podeu fer servir un paràmetre K que defineixi el nombre d'iteracions que s'executaran d'aquest algoritme. Però heu de **definir i implementar un criteri d'aturada** d'aquest algoritme basat en la qualitat dels estats actuals. Expliqueu i motiveu la vostra elecció.
   - **Estudieu la complexitat** temporal d'aquest algoritme. **Estudieu-ne la relació amb el paràmetre** B i amb la **mida del vecindari**. Aquest algoritme troba la **solució òptima**?
   - **Compareu-lo** amb altres versions de l'algoritme basades en el *primer veí que millora* o en selecció completament aleatòria. És més o menys **eficient a nivell de temps** de còmput? I a nivell de **convergència**? Troba l'òptim?
   - Com **alternativa** al punt anterior, podeu implementar un **algoritme genètic** i fer la comparació entre algoritmes. Haureu de triar una **representació per al problema** (us serveix la mateixa?) i definir les operacions de **selecció, encreuament i mutació**. Quin és més ràpid? Quin convergeix a una millor solució?

Disposeu d'un conjunt de fitxers a Moodle (vegeu Secció 2.4). Reviseu-los amb atenció ja que us donem l'estructura de la pràctica i alguna funcionalitat que no heu d'implementar (juntament amb els fitxers de dades).

2.3 Pas 3: Incorporació de coneixement imprecís

Fareu servir una xarxa Bayesiana (XB) per **adaptar la funció objectiu**.

La població associada a cada hospital té certes característiques que fan que cada hospital tingui més o menys casos greus (o crítics) de certa malaltia clau en aquest programa de donacions. En concret, el personal mèdic que hi ha darrere del programa ens indica que hi ha 3 característiques rellevants d'aquestes poblacions que determinen la probabilitat d'observar casos crítics: I, J i K. De fet, ens faciliten una XB que codifica la distribució de probabilitat subjacent a aquestes relacions, on $C = True$ indica preponderància de casos crítics:



I	$p(I)$
t	0.7

J	I	$p(J\|I)$
t	t	0.8
t	f	0.3

C	I	J	$p(C\|I,J)$
t	t	t	0.95
t	t	f	0.64
t	f	t	0.29
t	f	f	0.01

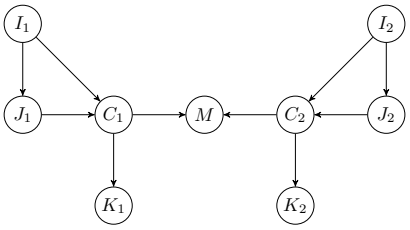
K	C	$p(K\|C)$
t	t	0.7
t	f	0.1

Figura 1: Estructura i paràmetres de la XB **criticalBN**. Ja la teniu codificada al fitxer `my_bns.py` (vegeu Sec. 2.4)

La idea és que la coincidència en la tipologia de la població sigui un factor a tenir en compte a l'hora de decidir quina és la millor distribució d'hospitals en xarxes. **Dos hospitals haurien d'estar en la mateixa xarxa quant més a prop estiguin, i quant més semblants siguin** pel que fa a la tipologia de les seves poblacions.

En altre paraules, heu de modificar la funció objectiu per poder integrar-hi una mesura de similitud entre hospitals basada en XBs. En concret, heu de:

1. Definiu al fitxer `my_bns.py` (vegeu Sec. 2.4) la xarxa Bayesiana **matchBN**:



M	C_1	C_2	$p(M\|C_1,C_2)$
t	t	t	-
t	t	f	-
t	f	t	-
t	f	f	-

Figura 2: Estructura de la XB **matchBN**. Aquesta **l'heu d'implementar vosaltres** al fitxer `my_bns.py` (vegeu Sec. 2.4). Assumirem que els paràmetres de les subxarxes (I_i,J_i,K_i) són els mateixos que els de la **criticalBN** (Fig. 1): els podeu copiar. Vosaltres només haureu d'emplenar la taula CPT de la distribució $p(M|C_1, C_2)$.

Cal **copiar la subestructura** $I - J - C - K$ de Fig. 1 dues vegades i **afegir-hi un CPD** per a $p(M|C_1, C_2)$. Heu de saber que si el valor de C_i ($i = 1$, $i = 2$) coincideix, la probabilitat de *match* ($\sim$ probabilitat de tenir una tipologia de població semblant) és 0.95. En canvi, si no coincideix, la probabilitat de *match* és 0.1.

2. **Adapteu la funció objectiu** per incorporar la similitud basada en la XB **matchBN** (Fig. 2). En concret, farem servir la probabilitat de *match* $P(M = True|\sigma(g)_1, \sigma(g)_2)$ com a mesura de semblança per redefinir la funció objectiu, que ara serà:

$$\min_{\sigma} \frac{1}{N} \sum_{g=1}^{G} N_g \cdot (\overline{dist}(\sigma, g) - \mu \cdot \overline{sim}(\sigma, g)) \tag{2}$$

on μ és el paràmetre que determina la importància relativa de la similitud, la distància mitjana intra-xarxa $\overline{dist}(\sigma, g)$ es defineix igual que abans i la similitud mitjana intra-xarxa és:

$$\overline{sim}(\sigma, g) = \frac{2}{N_g \cdot (N_g - 1)} \sum_{i=1}^{N_g - 1} \sum_{j=i}^{N_g} P\left(M = T|I_1 = x[I], J_1 = x[J], K_1 = x[K], I_2 = y[I], J_2 = y[J], K_2 = y[K]\right)$$

on σ és una assignació d'hospitals en xarxes, $\sigma(g)_i$ és l'i-èssim hospital de la xarxa g, $\boldsymbol{x} = \sigma(g)_i$ i $\boldsymbol{y} = \sigma(g)_j$.

Els valors per cada variable (I,J,K) de cada hospital els trobareu a la segona versió del fitxer de dades (`dades.csv`, vegeu Sec. 2.4). Les dades no són completes, hi ha valors no disponibles (`nan`: no tenim aquesta dada per l'hospital que correspongui). No us preocupeu, per això fem servir XBs, què poden lidiar amb informació incompleta!

3. Per calcular la probabilitat $P(M = T|I_1 = x[I], ..., K_1 = y[K])$ hem de fer servir un algoritme d'inferència probabilística. Ja teniu implementat a `inferencia.py` (vegeu Sec. 2.4) l'algoritme de *rejection sampling*.

   Per poder fer la consulta, només heu de decidir l'**ordre ancestral** que seguirà el procés de *forward sampling* per obtenir una mostra de la XB **matchBN**.

4. **Implementeu** *variable elimination*, l'algoritme d'inferència exacta. Teniu a `inferencia.py` (vegeu Sec. 2.4) un esquelet del mètode que heu d'implementar. També teniu a `bn.py` la classe `Factor` amb totes les operacions que necessiteu (`reduce, product, marginalize_out, normalize`) ja implementades.

   - Per poder fer una consulta amb aquest mètode heu de decidir un **ordre d'eliminació**, que no és el mateix que l'ordre ancestral del pas anterior!
   - **Quantes mostres** (paràmetre N) necessiteu amb *rejection sampling* per obtenir resultats que s'acostin als resultats de *variable elimination*?

5. **Implementeu** *weighted sampling*, l'algoritme d'inferència aproximada que, fent servir likelihood weighting (variant de *forward sampling*), permet un ús més eficient de les mostres generades. Teniu a `inferencia.py` (vegeu Sec. 2.4) la definició dels mètodes que heu d'implementar.

   - `weighting_sampling` és semblant a `rejection_sampling` però no descarta mai cap exemple, i no compta, sinò que acumula el pes de les mostres.
   - `likelihood_weighting` és una variant de `forward_sampling` que no agafa una mostra de $P(X|\dots)$ si X és una variable observada (en aquest cas, calcula la probabilitat del valor observat donat el valor mostrejat per als pares d'aquesta variable).
   - **Quantes mostres** (paràmetre N) necessiteu amb *likelihood-weighted sampling* per obtenir resultats que s'acostin als resultats de *variable elimination*?

   Feu servir el mateix ordre ancestral que amb *rejection sampling*.

6. **Estudieu com es comporta l'algoritme de cerca** davant diferents valors de μ.

2.4 Fitxers que us donem

Us donem els següents fitxers organitzats en dos subcarpetes que separen el material que necessitareu per a la primera sessió de laboratori i el què necessitareu per a la segona (i per l'entrega definitiva):

`s1/`: carpeta que inclou tot el que necessiteu per a la primera part de la pràctica.
`main1.py`: fitxer amb l'estructura de codi necessari. Inclou la definició de distància.
`datapoints.csv`: fitxer amb les dades necessàries fins al pas 2.

`s2/`: carpeta que inclou tot el que necessiteu per a l'entrega definitiva.
`main2.py`: fitxer amb l'estructura de codi necessari. Inclou la definició de distància i un mètode per transformar les dades en el format requerit per les XBs.
`data.csv`: fitxer amb totes les dades (coordenades (x,y) i descriptors dels hospitals.
`bn.py`: fitxer de codi amb les classes necessàries per crear i fer servir una XB.
`my_bns.py`: fitxer de codi amb la definició d'una XB (`criticalBN`) i altres estructures necessàries per al seu ús.
`inferencia.py`: fitxer de codi amb la implementació de l'algoritme d'inferència aproximada *rejection sampling* i l'estructura per implementar els què es demanen: *variable elimination* i *(likelihood-) weighted sampling*.

2.5 Instruccions generals

1. Aquesta pràctica es durà a terme al llarg de dues sessions de laboratori. Amb la primera sessió podreu desenvolupar els passos 1 i 2 (Sec. 2). Amb la segona sessió podreu abordar també el pas 3.

2. Podeu definir l'algoritme de cerca al fitxer de codi principal (p.e., `p1.py`) o a un altre que cregueu.

   La vostra entrega només ha de contenir **un únic fitxer principal**. Nosaltres us en donem dos per facilitar la comprensió de què podeu abordar a la primera o a la segona sessió de laboratori.

3. Intenteu mantenir a l'estructura de dades "`problema`" tota la informació relativa al problema.

4. Definiu la nova XB (`matchBN`) al fitxer `my_bns.py` (veure Secció 2.4).

5. Definiu els algoritmes d'inferència al fitxer `inferencia.py` (veure Secció 2.4).

6. Per implementar aquesta pràctica només és estrictament necessària la llibreria **numpy**.
7. El codi ha d'estar ben comentat.
8. El format de la sortida és lliure, sempre que mostri la solució que troba.
   Es recomana incloure alguna indicació per pantalla per saber que l'algoritme està funcionant (i no està penjat).

3 Instruccions de lliurament

Caldrà penjar a Moodle un fitxer comprimit **zip** (o **tgz**) amb un nom que segueixi el format

`CodiUsuari_CognomsNom_p1`

que contengui:

1. Tots els fitxers de codi necessaris per executar la pràctica. El fitxer principal ha de ser únic i fàcilment reconeixible (p.e., `p1.py`).

   S'ha de poder executar des de la terminal amb la comanda: `$ python p1.py`

2. Un fitxer `llegeix.me` amb aclariments que puguin facilitar la correcció. Sobretot, si hi ha alguna cosa que no funciona bé, expliqueu-ho aquí.

3. Un document `informe.pdf` amb l'explicació i motivació de totes les decisions que heu pres al llarg del desenvolupament de la pràctica. També ha d'incloure la discussió de les qüestions plantejades a la Secció 2.

   L'informe ha de tenir una portada amb el títol, el curs i el vostre nom.

   Es tracta d'un document formal que ha d'estar ben escrit, amb claretat i rigor.

La pràctica es pot fer **en parelles**. Si trieu aquesta opció, les dues persones han de pujar a Moodle l'arxiu comprimit amb la pràctica, com es demanava anteriorment, que inclogui també:

4. Un fitxer `avaluacio.txt` on avalueu, del 0 al 10, la vostra contribució a la realització de la pràctica i, també del 0 al 10, la contribució de la vostra parella. També podeu incloure-hi qualsevol comentari que ens vulgueu traslladar sobre el funcionament de la parella. Exemple de contingut:

   ```
   - Auto-avaluació:  7
   - Avaluació de la parella:  9
   Comentari:  tot ha funcionat correctament.  Comunicació fluïda, treball balancejat, decisions compartides.
   ```

   La manca d'aquest fitxer en una entrega s'entendrà com "`autoavaluació:0; avaluació-parella:5`".

La data límit de lliurament està especificada a l'activitat corresponent de Moodle.

4 Criteris d'Avaluació

- Definició i implementació de la cerca (35%).
- Definició i ús de la Xarxa Bayesiana (35%).
- Completesa, claredat i rigor de l'informe (30%).

Si l'entrega no cubreix tots els punts de la Secció 2, els criteris corresponents es poderaran en relació a la proporció de punts que s'han dut a terme.

El contingut d'aquesta pràctica és potencial **matèria d'examen**.