Fundamentals of Artificial Intelligence and Knowledge Representation - Module 3

Flood disaster prediction

Antonio Politano, Francesco Pieroni and Riccardo Spolaor

Contents

1 Introduction

Due to recent *climate change* related issues critical floods of urban areas have started becoming more and more common. Hence, a software able to estimate the likelihood of such an event could be a useful tool in order to prevent the most dramatic scenarios.

For this reason we proposed the implementation of a *Bayesian Network* to model the likelihood of floods in various municipalities of the Italian region Veneto. The structure of the network was inspired by the paper *Assessing urban flood disaster risk using Bayesian network model and GIS applications* [1], although quite deeply modified for didactic reasons.

The project was developed in `Python` using the `pgmpy` library [2] and it is available at the repository [3].

2 Bayesian Network construction

The Bayesian Network is composed by nine nodes which are listed below and represented in Figure 1:

- **Flood:**

  It describes the likelihood of particularly serious floods on a yearly basis. The nodes *Road Density*, *Slope* and *Rainfall amount* are strongly causally related with it;

- **Road density:**

  It illustrates the density of paved surfaces in a given area discretized in three levels: *dense*, *medium* and *sparse*. Intuitively the higher the road density the higher the chances of a serious flood. The nodes *Population density* and *Per unit GDP* are causally related with it;

- **Slope:**

  It explains whether the considered area is either *steep* or *flat*. Obviously a flat area has more chances to be flooded. It depends on the node *Elevation*;

- **Rainfall Amount:**

  It summarizes the average amount of water present on the ground in a year which of course increases the chances of flooding. The possible values are *huge*, *medium* and *little*. It is influenced by the nodes *Rainfall frequency* and *River Density*;

- **Population density:**

  It shows the density of the population of an area and it can be *dense*, *medium* or *sparse*. The node *Per unit GDP* is related with it. A densely populated area requires a higher presence of roads;

- **Per unit GDP:** It describes the average richness of people in an area divided into the categories: *high*, *medium* and *low*;

- **Elevation:** It illustrates the elevation of the territory in exam. It can be *high*, *medium* or *low*.

- **Rainfall frequency:** It explains the average rainfall frequency observed in a year. It is discretized in the values *frequent*, *medium* and *rare*;

- **River density:** It shows the density of rivers or other water elements in an area. It can be either *dense* or *sparse*.

The *Conditional Probability Distribution (CPD) tables* were defined through a qualitative approach, by inspecting the data of the websites [4, 5, 6, 7, 8].

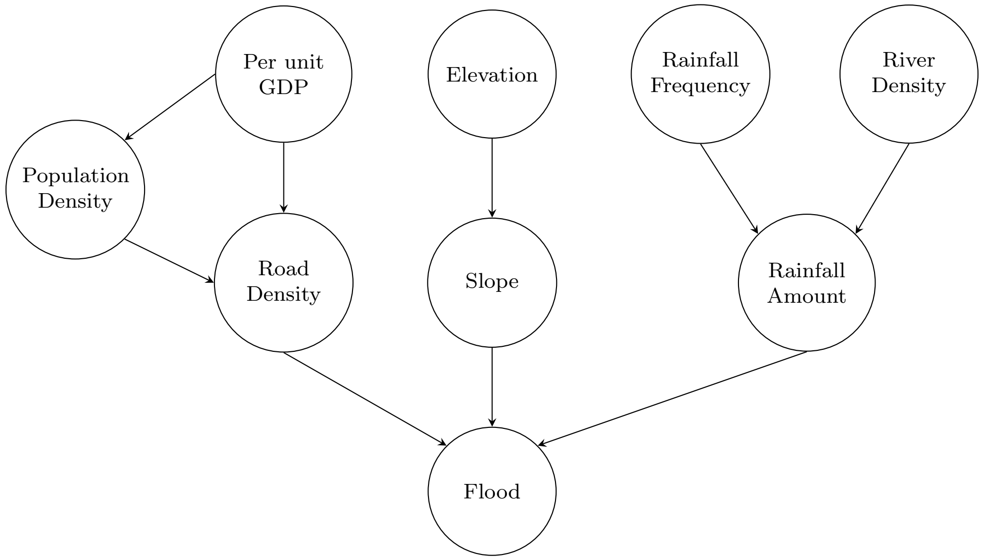


Figure 1: Bayesian Network

3 Theory concepts

In this section some theoretical concepts such as *independence*, *trails of causal influence*, *Markov blanket* and *inference* are explained through their application to our network.

3.1 Independence and trail of influence

Through the use of the method `local_independencies` we showed all the local independencies between the nodes given their parents as evidence (Table 1).

Per unit GDP ⊥ Slope, Rainfall amount, Elevation, Rainfall frequency, River density
Population density ⊥ Slope, Rainfall amount, Elevation, Rainfall frequency, River density | Per unit GDP
Road density ⊥ Rainfall frequency, Rainfall amount, Elevation, Slope, River density | Population density, Per unit GDP
Elevation ⊥ Population density, Rainfall amount, Road density, Per unit GDP, Rainfall frequency, River density
Slope ⊥ Population density, Rainfall amount, Road density, Per unit GDP, Rainfall frequency, River density | Elevation
Rainfall frequency ⊥ Population density, Elevation, Road density, Per unit GDP, Slope, River density
River density ⊥ Population density, Rainfall frequency, Elevation, Road density, Per unit GDP, Slope
Rainfall amount ⊥ Population density, Elevation, Road density, Per unit GDP, Slope | Rainfall frequency, River density
Flood ⊥ Population density, Elevation, Per unit GDP, Rainfall frequency, River density | Rainfall amount, Road density, Slope

Table 1: Independencies of the variables given their parents as evidence

Then we tested some other independencies checking active trails of influence between nodes using the method `active_trail_nodes`. For example we queried the method for the active trail of the node **River density** given the evidence **Rainfall amount**, which activates the *V-structure* and trail of influence between the **River density** and the **Rainfall frequency** nodes, making them dependent on each other.

3.2 Markov Blanket

The `pgmpy` library offers the method `get_markov_blanket` that returns the nodes which compose the Markov blanket of the queried variable. Therefore we checked the active trails of all the variables given their Markov blankets as evidence and we observed that no trail of influence was active (Examples in Figure 3).

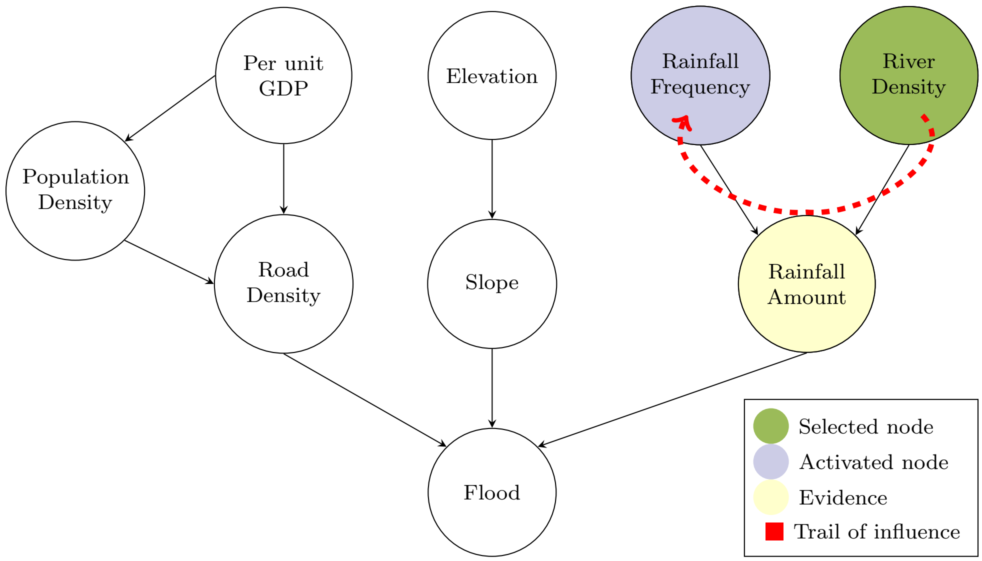
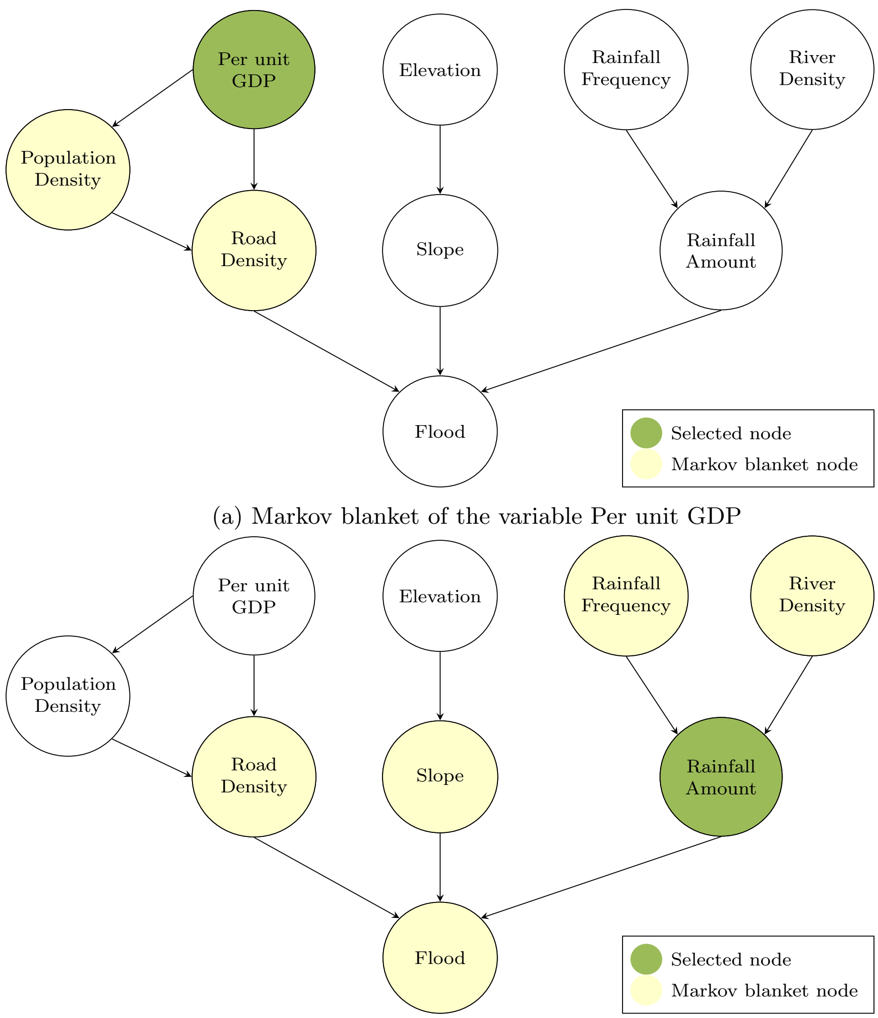


Figure 2: Example of an activated v-structure



(a) Markov blanket of the variable Per unit GDP

(b) Markov blanket of the variable Rainfall amount

Figure 3: Examples of Markov blankets of specific nodes of the network

3.3 Exact inference

Afterwards, we decided to compute the *exact inference* of specific values using the `VariableElimination` class provided by the `pgmpy` library. To begin, the likelihood of Flood has been computed and the results are shown in Table 2.

Flood	**phi(Flood)**
Yes	0.0785
No	0.9215

Table 2: Exact inference of Flood

Next, starting from the worst case scenario we tested the chances of serious floods and then we performed other queries in order to determine which of the causes is more determinant. In addition we executed some "reverse" queries to estimate causes taking as evidence the effect, such as the probability distribution of Slope given the effect Flood (Table 3).

Slope	**phi(Slope)**
Steep	0.0736
Flat	0.9264

Table 3: Exact inference of Slope given the presence of Flood ($P(\text{Slope} \mid \text{Flood} = \text{Yes})$)

3.4 Approximate Inference

At this point we computed the approximate inference with three different procedures: *Rejection sampling*, *Weighted likelihood* method and *Markov chain Monte Carlo* method. In general all the three methods obtain very good precision in estimating the probabilities of a variable given no evidence. The results were very accurate since the network is not so complex and it has a limited number of nodes and states. Different results are obtained when probabilities are computed given a piece of evidence and the number of samples is limited. Each evaluation test has been carried out while fixing the *RNG seed* at a value of 50.

3.4.1 Rejection Sampling

To implement this procedure we used methods of the `ApproximateInference` class provided by `pgmpy`. Among other evaluations, we considered the worst case scenario, namely the probability of Flood given frequent rainfalls and a flat area with high Road and River density. We computed the probability with 1,000 and 40 samples. The accuracy in the second case is, as expected, lower, as shown in table 4.

Flood	**phi(Flood)**
Yes	0.1608
No	0.8392

(a) Exact inference

Flood	**phi(Flood)**
Yes	0.1840
No	0.8160

(b) R.S. (1,000 samples)

Flood	**phi(Flood)**
Yes	0.2500
No	0.7500

(c) R.S. (40 samples)

Table 4: Comparison of P(Flood | Road density = Dense, River Density = Dense, Slope = Flat, Rainfall Frequency = Frequent) by exact inference and rejection sampling with a different numbers of samples

3.4.2 Weighted Likelihood Method

To apply this procedure we slightly modified the `ApproximateInference` and `BayesianNetwork` classes by extending them since the `pgmpy` library doesn't offer a handy method to compute it. Repeating the same experiment the results are better than the previous case when a lower number of samples is used (Table 5).

Flood	**phi(Flood)**
Yes	0.1608
No	0.8392

(a) Exact inference

Flood	**phi(Flood)**
Yes	0.1730
No	0.8270

(b) W.L. (1,000 samples)

Flood	**phi(Flood)**
Yes	0.1750
No	0.8250

(c) W.L. (40 samples)

Table 5: Comparison of P(Flood | Road density = Dense, River Density = Dense, Slope = Flat, Rainfall Frequency = Frequent) by exact inference and weighted likelihood with a different numbers of samples

3.4.3 Markov Chain Monte Carlo Method

In order to apply this algorithm we used the particular implementation of `GibbsSampling` offered by the `pgmpy` library. An interesting comparison considering the same amount of samples as the previous cases of approximate inference has been carried out. The results illustrate that not even 1,000 samples are enough to obtain a good approximation, since, after the sampling procedure, we consider just the samples that match the given evidence, resulting in the selection of a very small subset (Table 6).

Flood	**phi(Flood)**
Yes	0.1608
No	0.8392

(a) Exact inference

Flood	**phi(Flood)**
Yes	0.1627
No	0.8373

(b) MCMC (100,000 samples)

Flood	**phi(Flood)**
Yes	0.2273
No	0.7727

(c) MCMC (1,000 samples)

Table 6: Comparison of P(Flood | Road density = Dense, River Density = Dense, Slope = Flat, Rainfall Frequency = Frequent) by exact inference and MCMC with a different numbers of samples

4 Use case

In this last part we decided to apply the Bayesian network inference to a real scenario, namely the chances of Flood in each municipality of the Italian Veneto region in the case of frequent rainfalls. The municipalities data has been collected from the following official sources [9, 10, 11, 12, 13, 14]. We computed the **Flood** chances through exact inference given as **evidence** the variables **Per unit GDP**, **Population density**, **Slope**, **River density** of each municipality and considering **frequent rainfalls** over all the areas.

$$\forall\ m \in \text{Municipalities}$$

$$P(\text{Flood}_m \mid \text{Per unit GDP}_m, \text{Population density}_m, \text{Slope}_m, \text{River density}_m, \text{Rainfall frequency} = \text{Frequent})$$

Looking at the heatmap (Figure 4) produced by the model we can see that the probabilities of flooding in each municipality is generally low considering the fact that the chances are estimated on a yearly basis. We can see though that the likelihood is much lower if we take in exam the northern areas in contrast to the central ones. The main reasons for such results are the differences in slope and population densities of the areas. In fact the northern part is mainly mountainous, the chances of flooding only increase near Belluno since it is way denser and its surface is almost flat. More densely populated and richer areas such as some chief towns (Venezia, Verona, Vicenza, Padova, Treviso and Rovigo) have the highest likelihood of experiencing floods, whereas poorer and less populated although quietly flat areas in the southern regions show medium probabilities of being flooded. The topographic coordinates data to build the heatmap has been fetched from the repository [15].

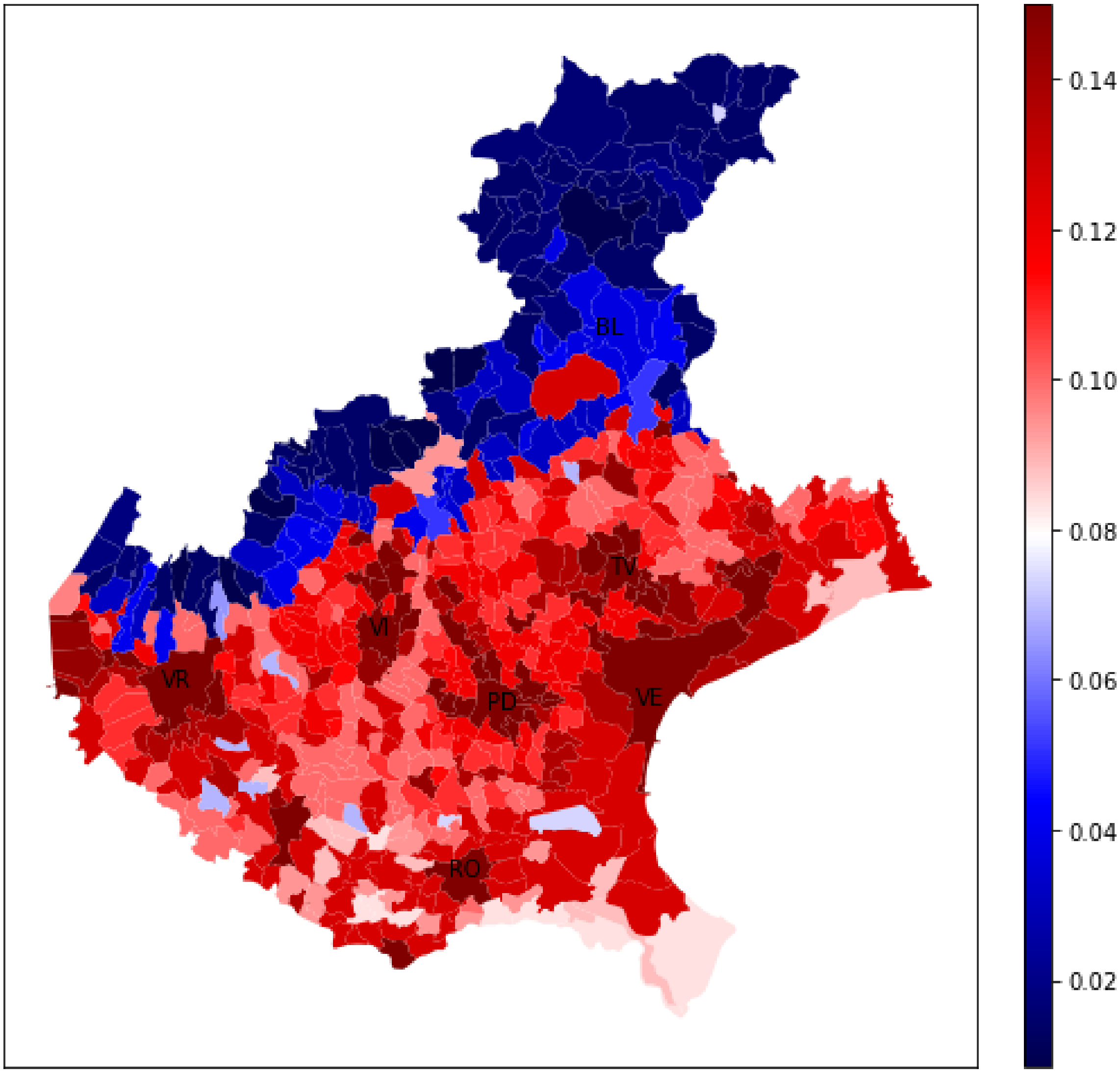


Figure 4: Chances of flooding on an annual base for each municipality of the Veneto region

References

[1] Zening Wu et al. “Assessing urban flood disaster risk using Bayesian network model and GIS applications”. In: *Geomatics, Natural Hazards and Risk* (2019). DOI: 10.1080/19475705.2019.1685010. URL: https://doi.org/10.1080/19475705.2019.1685010.

[2] Ankur Ankan and Abinash Panda. “pgmpy: Probabilistic graphical models using python”. In: *Proceedings of the 14th Python in Science Conference (SCIPY 2015)*. Citeseer. 2015.

[3] Antonio Politano, Francesco Pieroni, and Riccardo Spolaor. *Flood disaster prediction*. https://github.com/RiccardoSpolaor/Flood-disaster-prediction. 2022.

[4] Venetoeconomia. *Reddito pro capite degli italiani 2017*. URL: https://www.venetoeconomia.it/2019/04/redditi-veneto-treviso-porto-tolle/.

[5] topographic-map.com. *Altitudine Italia*. URL: https://it-ch.topographic-map.com/maps/gprg/Italia/.

[6] Emanuele Martino. *Fiumi principali in Veneto*. URL: https://emanuelemartino.files.wordpress.com/2020/01/fiumi-principali.gif?w=736.

[7] Arpav Veneto. *Precipitazioni annue in Veneto*. URL: https://www.arpa.veneto.it/arpavinforma/indicatori-ambientali/indicatori_ambientali/clima-e-rischi-naturali/clima/precipitazione-annua/view.

[8] Anas S.p.A. *Anas per regione*. URL: https://www.stradeanas.it/it/le-strade/anas-regione.

[9] Ministero dell'Economia e delle Finanze. *Analisi statistiche - Dichiarazioni 2020 - Anno d'imposta 2019*. URL: https://www1.finanze.gov.it/finanze/analisi_stat/public/index.php?tree=2020.

[10] Data.europa.eu. *Superficie Territoriale in kmq dei Comuni del Veneto*. URL: https://data.europa.eu/data/datasets/superficie_territoriale_in_kmq_dei_comuni_del_veneto?locale=it.

[11] Istituto Nazionale di Statistica. *DATI STATISTICI PER IL TERRITORIO*. URL: https://www.istat.it/it/archivio/243448.

[12] Il portale della Regione del Veneto. *Ambiente e Territorio - Tutela del territorio e dell'ambiente del Veneto*. URL: https://www.regione.veneto.it/web/ambiente-e-territorio/scheda-dati.

[13] Istituto Nazionale di Statistica. *PRINCIPALI STATISTICHE GEOGRAFICHE SUI COMUNI*. URL: https://www.istat.it/it/archivio/156224.

[14] Città & Borghi. *Città e Borghi in Italia*. URL: https://www.cittaeborghi.it/it/.

[15] Openpolis. *geojson-italy*. URL: https://github.com/openpolis/geojson-italy.