Graph clustering

MVA : Introduction to Probabilistic Graphical Models and Deep Generative Models

Quentin MOAYEDPOUR qmoayedpour@ensae.fr

Florian OUBAHA florian.oubaha@polytechnique.edu

December 2024

Abstract

Graph clustering is a fundamental problem aiming to identify meaningful groups of nodes within a graph structure based on connectivity patterns. We employ an Expectation-Maximization (EM) algorithm designed specifically for graphs adhering to the Stochastic Block Model (SBM).

This algorithm performs well on real-world data, and we find evidence that the choice of the initialization method can significantly influence the quality of the clustering results. Different initialization strategies generate diverse starting points, leading to variations in the exploration of local optima based on the chosen generation method. The code of the report can be found here Github

1 Introduction

Graph theory dates back at least to the 18th century with Euler, who is credited with laying its foundations through his work on the Seven Bridges of Königsberg problem[1]. Graphs serve as powerful models for representing various real-world systems, such as social networks and biological networks, where vertices denote individuals and edges represent the interactions between them.

One of the key areas of research in graph theory is community detection, which refers to the process of identifying the underlying structural patterns within a graph by grouping its nodes or edges into distinct clusters[2]. Communities are often characterized by high internal connectivity among their nodes compared to their connections with the rest of the graph. These clusters not only reveal insights about the organization of networks but also have practical implications in various fields. For instance, it assists in identifying functional modules in protein interaction networks or gene co-expression networks, which can help uncover mechanisms behind diseases or biological processes[3].

Community detection methods have advanced to include both non-statistical approaches and those based on formal probabilistic models. Methods that do not pose a statistical model typically rely on heuristics and optimization algorithms to identify communities. A well-known example is the Louvain method, which improves network organization by repeatedly combining nodes and communities to find the best way to divide the network[4]. In contrast, models based on the Stochastic Block Model (SBM) provide a statistical approach to community detection[5]. This probabilistic formulation allows for a more formal and principled way of community detection, providing not only an assignment of nodes to communities but also the probability distributions that describe the likelihood of edges between different communities.

In this review, we propose to analyze graph clustering models using data generated from an SBM as well as real-world data, with a particular focus on the convergence conditions of these models.

We then test the algorithm we choose (EM) to compare the performance with different types of initializations and on different data : Generated data

following an SBM and some real-world graphs.

2 Graph Model for community detection

Consider a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ where $\mathcal{V}$ is the set of **vertices** and $\mathcal{E}$ the set of **edges**. We consider undirected graphs, meanings if $\{v_i, v_j\} \in \mathcal{E} \implies \{v_j, v_i\} \in \mathcal{E}$ for $i \neq j$. The **adjacency matrix** X of the graph is a square matrix where :

$$(X)_{i,j} = \begin{cases} 1 & \text{if } \{v_i, v_j\} \in \mathcal{E} \\ 0 & \text{else} \end{cases}$$

And $(X)_{i,i} = 0 \quad \forall i$. For a vertex i, we have its degree defined by $K_i := \sum_{j \neq i} X_{i,j}$. A widely studied statistical model for graphs is the Erdős–Rényi model[6], where we have:

$$(X)_{i,j} \overset{\text{i.i.d.}}{\sim} \mathcal{B}(p)$$

However, the Erdős–Rényi model can be limited in modeling real-world phenomena. Its simple design does not account for the heterogeneity among vertices[7]. In contrast, the Stochastic Block Model provides a more flexible framework for capturing the structure and community organization often observed in real-world networks.

2.1 Stochastic Block Model

It is assumed that the vertices are divided into Q classes, with the distribution $\{\alpha_1, \ldots, \alpha_Q\}$ associated with them. Let $Z_{i,q}$ denote the hidden variables which indicates the label of vertices to classes, i.e. $Z_{i,q} = 1$ if the vertex i belongs to class q. We have:

$$\sum_{q=1}^{Q} \alpha_q = 1, \quad \sum_{q=1}^{Q} Z_{i,q} = 1$$

$$P(Z_{i,q} = 1) = P(\{i \in q\}) = \alpha_q$$

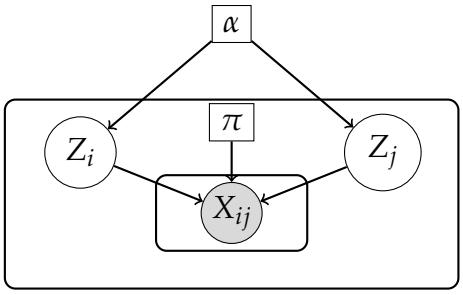

Figure 1: Stochastic Block Model

In the SBM, we suppose that the probability for two vertices to be connected depends on the class of each vertices. Namely with have π, the connectivity matrix where $\pi \in [0,1]^{Q \times Q}$, wich satisfy:

$$X_{i,j} | \{i \in q, j \in l\} = \mathcal{B}(\pi_{q,l})$$

A key property of SBM is its flexibility. It generalizes the Erdős-Rényi model by introducing heterogeneity in edge probabilities across vertex groups, making it a powerful tool for understanding and analyzing networks with inherent structure.

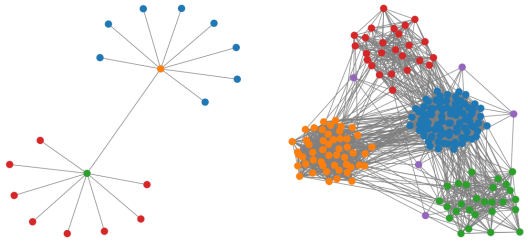

Figure 2: Synthetic Graph under SBM

2.2 Constrained Stochastic Block Model

A well-known variant of the stochastic block model is the constrained SBM. In this specific case, the distribution of the classes α still follows a multinomial distribution, but the connectivity matrix is constrained. Under the constrained model, we have:

$$\pi_q = v^+ \quad \forall q \in [Q]$$
$$\pi_{q,l} = v^- \quad \forall q, l \in [Q], \quad q \neq l$$

The model then estimates the probability of being connected to a vertex within the same cluster, as well as the probability of being connected to a vertex from a different cluster. Hofman & al. proposed a method in [8] to estimate the latents variables τ_i under the constrained model (VBMod). This approach significantly reduces the computational complexity compared to traditional SBM estimation methods, making it more scalable for large networks, but it relies on stronger assumptions.

2.3 A Bayesian Approach to Network Modularity

In the VBmod [8], some parameters are supposed to follow beta and Dirichlet distribution. Moreover, to find the number Q of cluster one want to maximize $\Pr(Q|X)$, and in absence of a prior on $\Pr(Q)$ the choice of Q is to maximize $Pr(X|Q)$.

The aim is to minimize the free energy, which is an upper bound of the negative log-likelihood obtained using Gibb's inequality. Then, until convergence, the algorithm minimize the energy by choosing $\boldsymbol{\tau}$ corresponding to the parameters used in the beta and Dirichlet distributions and reciprocally by choosing these parameters with the updated $\boldsymbol{\tau}$.

We implemented this model and obtained results that were less satisfactory compared to the EM algorithm detailed in the next section.

One hypothesis to explain the performance of this model is the fact that it follows a constrained stochastic block model which suppose an inter-cluster probability of connection independent of the clusters, and such a model is not adapted to many real-world graph.

The model on which we focus our work avoid this complication by considering a variational approach which supposed the general stochastick block model.

3 A Mixture Model for Random Graphs

J.J Daudin studied the stochastic block model in [7] and propose a variational approach to approximate maximum likelihood inference on the parameters. The proposed method aims to address the computational challenges associated with maximum likelihood estimation in the SBM, especially for large graphs. The authors proposed a framework to apply the EM-algorithm[9] to maximise a lower-bound on the likelihood of the model.

The log-likelihood of the SBM is given by:

$$\log \mathcal{L}(\mathcal{X}) = \sum_{i=1}^{n} \sum_{q=1}^{Q} Z_{iq} \log(\alpha_q) + \frac{1}{2} \sum_{i \neq j} \sum_{q,l} Z_{iq} Z_{jl} \pi_{ql}$$

Since we do not observe the Z_i, the likelihood is not tractable. The EM algorithm can not directly be used since it needs to compute $\Pr(\mathcal{Z} \mid \mathcal{X})$ which is not tractable. Therefore, the authors uses a variational approach to optimize a lowerbound on $\mathcal{L}(\mathcal{X})$

3.1 EM algorithm for SBM

Then, in the variational approach the objective is to maximize : $\mathcal{J}(R_{\mathcal{X}}) = \log \mathcal{L}(\mathcal{X}) - \mathrm{KL}[R_{\mathcal{X}}(\cdot), \Pr(\cdot \mid \mathcal{X})]$, where $R_{\mathcal{X}}$ is a proxy of the conditional distribution $\Pr(. \mid \mathcal{X})$ and KL is the Kullback-Leibler divergence. We constraint $R_{\mathcal{X}}$ to have the following form:

$$R_{\mathcal{X}}(\mathcal{Z}) = \prod_i h(Z_i; \boldsymbol{\tau}_i)$$

where $\boldsymbol{\tau}_i = (\tau_{i1}, \ldots, \tau_{iQ})$ and $h(\cdot; \boldsymbol{\tau})$ denotes the multinomial distribution with parameter $\boldsymbol{\tau}$. τ_{iq} can be interpreted as $\Pr(\{Z_{i,q}\} = 1)$.

Then, the parameters on which we want to maximize $\mathcal{J}(R_{\mathcal{X}})$ are $\boldsymbol{\tau}$, $\boldsymbol{\alpha}$ and $\boldsymbol{\pi}$.

Where π_{ql} is the probability that a node in the cluster q an a node in the cluster l are connected, and $\alpha_q = \Pr(Z_{iq} = 1)$.

One can show that given $\boldsymbol{\tau}$, the parameters $\hat{\boldsymbol{\alpha}}$ and $\hat{\boldsymbol{\pi}}$ that maximize $\mathcal{J}(R_{\mathcal{X}})$ are:

$$\hat{\boldsymbol{\alpha}}_q = \frac{1}{n} \sum_{i=1}^{n} \tau_{iq}$$

$$\hat{\boldsymbol{\pi}}_{ql} = \sum_{i \neq j} \tau_{iq} \tau_{jl} X_{ij} \Big/ \sum_{i \neq j} \tau_{iq} \tau_{jl}$$

Similarly, given $\boldsymbol{\alpha}$ and $\boldsymbol{\pi}$, the parameters $\hat{\boldsymbol{\tau}}$ that maximize $\mathcal{J}(R_{\mathcal{X}})$ are:

$$\hat{\tau}_{iq} \propto \alpha_q \prod_{j \neq i} \prod_{l} b(X_{ij}; \pi_{ql})^{\hat{\tau}_{jl}}$$

With $b(x; \pi) = \pi^x (1 - \pi)^{1-x}$

Note that the computation of $\hat{\boldsymbol{\alpha}}$ and $\hat{\boldsymbol{\pi}}$ are straightforward, but the computation of $\hat{\boldsymbol{\tau}}$ is more complicated because $\hat{\boldsymbol{\tau}}$ appears on both side of the equation. We use the following fixed point algorithm to compute $\hat{\boldsymbol{\tau}}$:

Algorithm 1 Fixed point algorithm

while not convergence **do**
  for $iteration = 1, 2, \ldots, max_{it}$ **do**
    $t_{iq} \propto \alpha_q \prod_{j \neq i} \prod_l b(X_{ij}; \pi_{ql})^{\tau_{jl}}$
    $\boldsymbol{\tau} \leftarrow \boldsymbol{t}$
  end for
end while
Return $\boldsymbol{\tau}$

Pseudocode EM

Finally, we obtain the following algorithm:

Algorithm 2 EM

Initialize τ
while not convergence **do**
  for $iteration = 1, 2, \ldots, max_{it}$ **do**
    M-Step : From τ calculate α and π
    E-Step : From α and π calculate τ with the fixed-point algorithm
    Compute the new likelihood
  end for
end while

4 Implementations & Experiments

Different initializations

In our first implementation, the initialization was random, i.e. $\tau_{i,q}$ follow an uniform law on $[0, 1]$ and then we do a normalization.

$$\tau_{i,q} \leftarrow \frac{\tau_{i,q}}{\sum_{q=1}^{Q} \tau_{i,q}}$$

We test two variants of this initializations :

- For each i we set $\tau_i = 0$ then we choose a random indice q such that $\tau_{i,q} = 1$.
- Sparsed initialization : For each i, before the normalisation we choose a given number $S < Q$ and then we choose S random indice q such that $\tau_{i,q} = 0$.

In order to have better results, we use graph clustering algorithm to perform our initialization. There are two conditions in order that the initialization is useable in practice : the EM must do better that just the initialization, and the running time with should not be much longer with the initialization. A widely used algorithm for this purpose is the K-means algorithm [10] for its simplicity and well-known properties.

Complexity

Let's denote by n the number of nodes in the graph and by Q the number of clusters. The overall complexity of the EM algorithm is in $\mathcal{O}(n^2Q^2)$.

Running time

Let's denote by n_{EM} and n_{fixed} the number of iterations of the main loops respectively for the EM algorithm and the fixed point algorithm.

For $Q = 3$, with $n_{EM} = 50$ and $n_{fixed} = 50$:

	$n = 100$	$n = 500$	$n = 1000$
$Q = 3$	1.31	1.57	6.93
$Q = 7$	1.22	5.71	31.09
$Q = 15$	1.43	24.40	95.92

Figure 3: Different running time

We observe here that for a small number of data points, the number of classes has little impact on computation time. However, for larger datasets, increasing the number of classes can significantly increase computational costs.

Performance with different initialization

Maximization of the likelihood of generated SBM graph

In this section, we run the following experience. Given a graph $\mathcal{G}$, the model is estimated using $n_{init} = 20$ different initializations, where the τ_i's values are randomly assigned. Subsequently, the same process is repeated with τ_i's values initialized with sparse initialization as described in the previous section. Finally, a third experiment is conducted where the τ_i's values are initialized using the K-means algorithm. We repeat this experience for different number of clusters Q and graph length n. We fix $n_{EM} = n_{fixed} = 50$. We present the results in the following table 1.

Surprisingly, the results indicate that for a certain number of instantiations (in our case, $n_{init} = 20$),

Initialization	Q	$n = 100$	$n = 1000$
		$log(\mathcal{L}(\mathcal{X}))$	
Random	3	-2696.67	-136384.72
	20	**-164.64**	-26474.21
Sparse Random	3	-2696.65	-136384.72
	20	-167.94	-26430.59
K-means	3	**-2696.57**	-136384.72
	20	-179.41	**-26279.75**

Table 1: Maximum log likelihood for different initializations

the impact of the initialization method is minimal. Table 1 demonstrates that the outcomes under different initialization methods are very similar, despite the expectation that the K-means method would yield superior results.

A closer examination of the results reveals that random initializations often lead to very similar paths, whereas initializations using K-means show greater diversity as show in 6. Under random initializations, the model typically starts from the same point, and the estimates evolve almost concurrently. In contrast, the K-means initialization allows the model to start from points that are more or less close to the solution, but this seems to drive some estimates toward local maxima rather than global ones.

Performance on real graphs

The results on generated SBM graph is not satisfaisant to measure the performance of the model because the model is made for SMB graph. Then we want to try the model on real-world graphs.

We choose 2 graphs available in the GitHub of Vladimir Ivashkin [11], SchoolDays[12] & PolBooks[13].

The graph *sp-school-day-2* measures face-to-face contact patterns in a primary school. This graph has 238 nodes, 5539 edges and 11 communities. The graph *polbooks* is a network of books about US politics, and edges represent frequent

co-purchasing of books by the same buyers. This graph has 105 nodes, 441 edges and 3 communities.

We obtain the following results, where NMI denotes the Normalized Mutual Information.

Init	Metrics	Sp-school	Polbooks
Random	Likelihood	-9153.10	-1426.42
	NMI	0.695	0.090
Sparse	Likelihood	**-9152.65**	-1310.66
	NMI	0.719	**0.570**
K-means	Likelihood	-9207.5	**-1306.67**
	NMI	**0.760**	0.516

Table 2: Maximum likelihood and NMI with 20 initializations on real-world data.

The results show that the sparse initialization can perform better than the K-means initialization, and the reverse is true. Moreover, the best Likelihood doesn't implies the best fit in the sense of the NMI. On the Sp-school graph, K-means seems to perform better even if Sparse has a better likelihood, and on the Polbooks graph K-means has a better likelihood but Sparse has a better NMI.

Furthermore, the K-means initialization leads to much more diverse results when doing multiple inits, whereas multiple instantiations using "Random" or "Sparse" initializations generally lead to very similar outcomes, with little variation across different runs. This suggests that initializing with K-means is offering more diversity but potentially less stability compared to the other methods.

Choice of the number of clusters

For a model m_Q with Q classes, the mixture parameters $= (\boldsymbol{\alpha}, \boldsymbol{\pi})$) are $\frac{Q(Q+3)}{2} - 1$. Indeed, $\boldsymbol{\pi}$ is a symmetric matrix of size Q so there are $\frac{Q(Q+1)}{2}$ parameters, and $\boldsymbol{\pi}$ has $Q-1$ parameters because of the normalization constraints. Choosing the right number of classes can be done by optimizing one of the following criterion AIC, BIC or ICL, defined in 5.

To examine the performance of the class selection criteria, we generate several graphs of different sizes. In a theoretical framework using graphs generated according to the SBM, our experiments demonstrate that the AIC, BIC, and ICL criteria fail to identify the correct number of classes when the graph contains limited data. However, all three criteria successfully identify the correct number of classes as the dataset size increases.

Conversely, when testing on real-world data, we observe that on the two datasets analyzed, none of the criteria were able to select the correct number of classes. All the criteria overestimated the number of classes. This issue has been specifically studied by Côme et al. [14], who proposed an algorithm to estimate the correct number of clusters using a greedy approach 4.

5 Conclusion

The EM algorithm we used performs well on SBM graphs, which is expected since it was specifically designed for such graphs. It also yields good results on real-world graphs. During our work, we try different initializations of the EM algorithm of the Mixture Model in order to compare the results.

First, the initialization with a K-means algorithm does not perform either worse or better in general than a random uniform initialization. But for a fixed graph, the initialization with a K-means algorithm can perform better, et the reverse is true.

However, the random uniform initialization appears to be limited because the optimization paths tend to be quite similar. In practice, these initializations do not allow the algorithm to explore sufficiently diverse clustering to reach higher-quality local maxima.

Therefore, it is preferable to test different initializations that follow various generative models. This approach allows for exploring more diverse starting points, ultimately leading to better results.

References

[1] Leonhard Euler. "Solutio problematis ad geometriam situs pertinentis". In: *Commentarii Academiae Scientiarum Imperialis Petropolitanae* 8 (1736), pp. 128–140.

[2] Satu Elisa Schaeffer. "Graph clustering". In: *Computer Science Review* 1.1 (2007), pp. 27–64. ISSN: 1574-0137.

[3] Albert-Laszlo Barabasi and Zoltan Oltvai. "Network Biology: Understanding The Cell's Functional Organization". In: *Nature reviews. Genetics* 5 (Mar. 2004), pp. 101–13.

[4] Vincent D Blondel et al. "Fast unfolding of communities in large networks". In: *Journal of Statistical Mechanics: Theory and Experiment* 2008.10 (Oct. 2008), P10008. ISSN: 1742-5468.

[5] K Nowicki, A. Krysztof, and T.A.B. Snijders. "Estimation and prediction for stochastic blockstructures". English. In: *Journal of the American Statistical Association* 96.455 (Sept. 2001), pp. 1077–1087.

[6] P Erdös and A Rényi. "On Random Graphs I". In: *Publicationes Mathematicae Debrecen* 6 (1959), pp. 290–297.

[7] Franck Picard Jean-Jacques Daudin Stephane Robin. "A mixture model for random graph". In: *Statistics and computing* (2008).

[8] Jake M. Hofman and Chris H. Wiggins. "A Bayesian Approach to Network Modularity". In: *Physical Review Letters* 100.25 (June 2008). ISSN: 1079-7114.

[9] A. P. Dempster, N. M. Laird, and D. B. Rubin. "Maximum likelihood from incomplete data via the EM algorithm". In: *Journal of the Royal Statistical Society: Series B* 39 (1977), pp. 1–38.

[10] Santo Fortunato. "Community detection in graphs". In: *Complex Networks and Systems Lagrange Laboratory, ISI Foundation, Viale S. Severo 65, 10133, Torino, I-ITALY* (2010).

[11] Vladimir Ivashkin and Pavel Chebotarev. "Do logarithmic proximity measures outperform plain ones in graph clustering?" In: *International Conference on Network Analysis*. Springer. 2016, pp. 87–105.

[12] Juliette Stehlé et al. "High-resolution measurements of face-to-face contact patterns in a primary school". In: *CoRR* abs/1109.1015 (2011).

[13] M. E. J. Newman. "Modularity and community structure in networks". In: *Proceedings of the National Academy of Sciences* 103.23 (June 2006), pp. 8577–8582. ISSN: 1091-6490.

[14] Etienne Côme and Pierre Latouche. "Model selection and clustering in stochastic block models with the exact integrated complete data likelihood". working paper or preprint. Mar. 2013.

Contribution Statement

Both members of the group worked together on nearly all parts of the project. We collaboratively wrote and implemented the functions for the code, jointly designed the experiments, and co-authored the report. Except for the experiment notebooks, where we divided the work

Difficulties Encountered

The main challenges encountered were of a technical nature. The paper by Daudin et al. was quite detailed and sufficient for the implementation. However, we quickly encountered limitations due to resource constraints. The primary difficulties were related to parallelizing the code, implementing it in PyTorch to leverage GPU advantages, and optimizing storage to prevent GPU saturation. Another challenge was managing "NaN" values resulting from certain operations that could rapidly diverge when parameters were very small or very large (e.g., logarithms, exponentials, etc.). The workload distribution was very effective as there were only two of us, and good organization made the project enjoyable to carry out.

Appendix

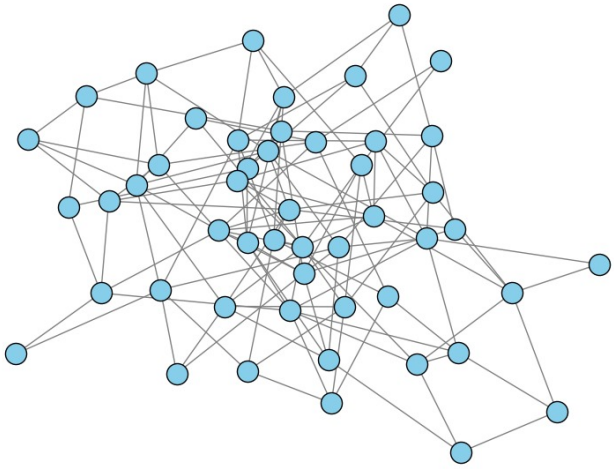

Figure 4: Graph simulation under the Erdős–Rényi model

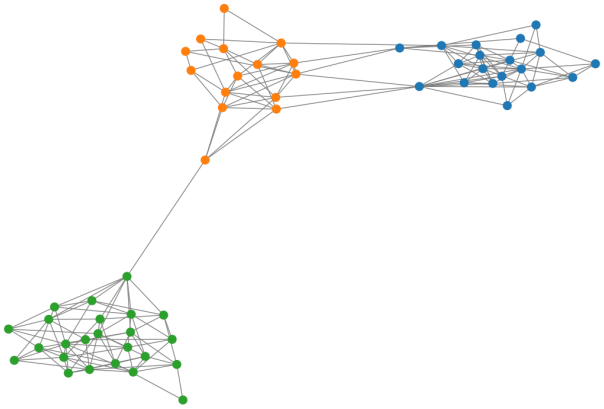

Figure 5: Graph simulation under the Stochastic Block Model

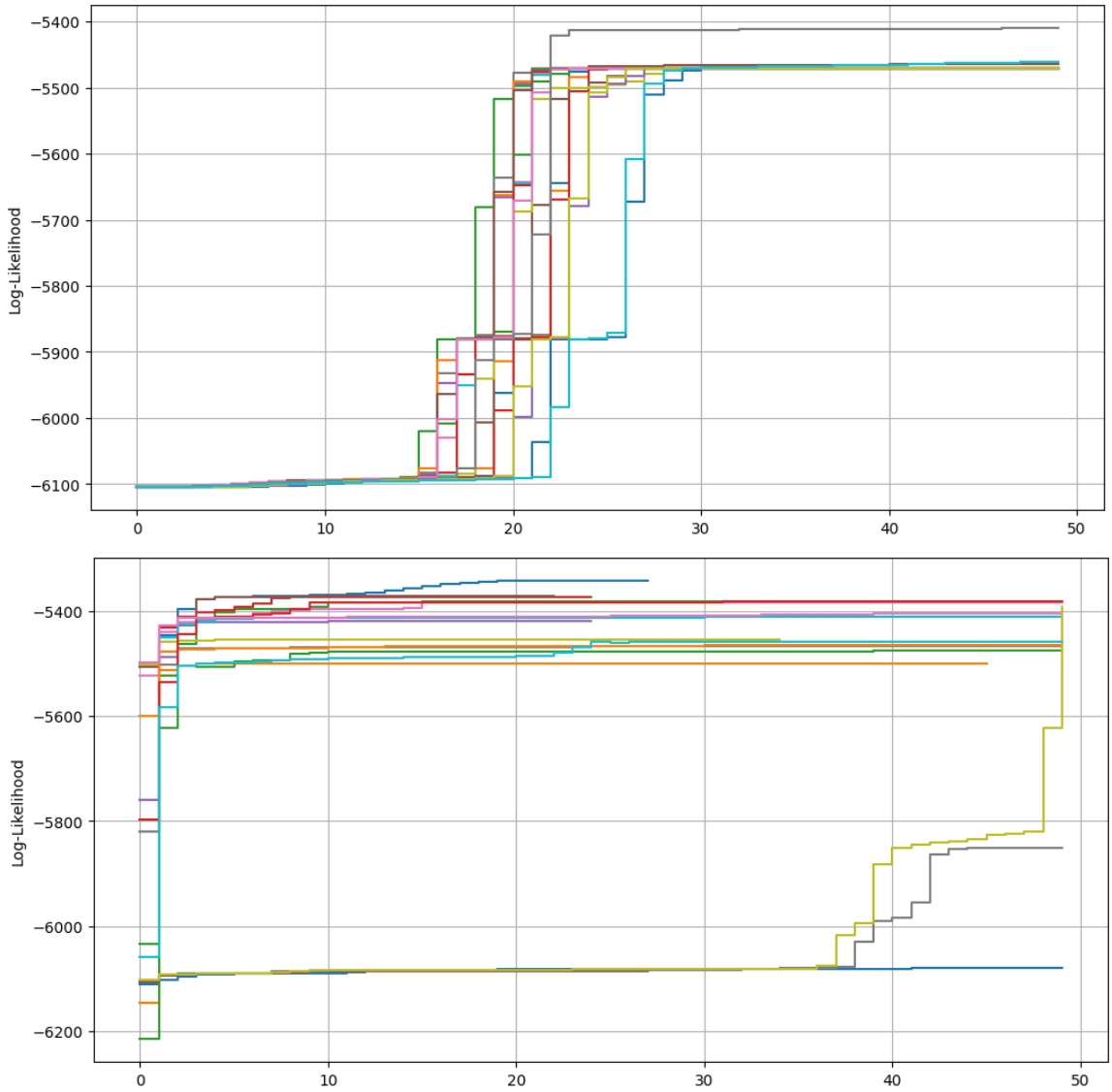

Figure 6: Log Likelihood track under Random (up) and Kmeans (down) initializations for synthetic graphs

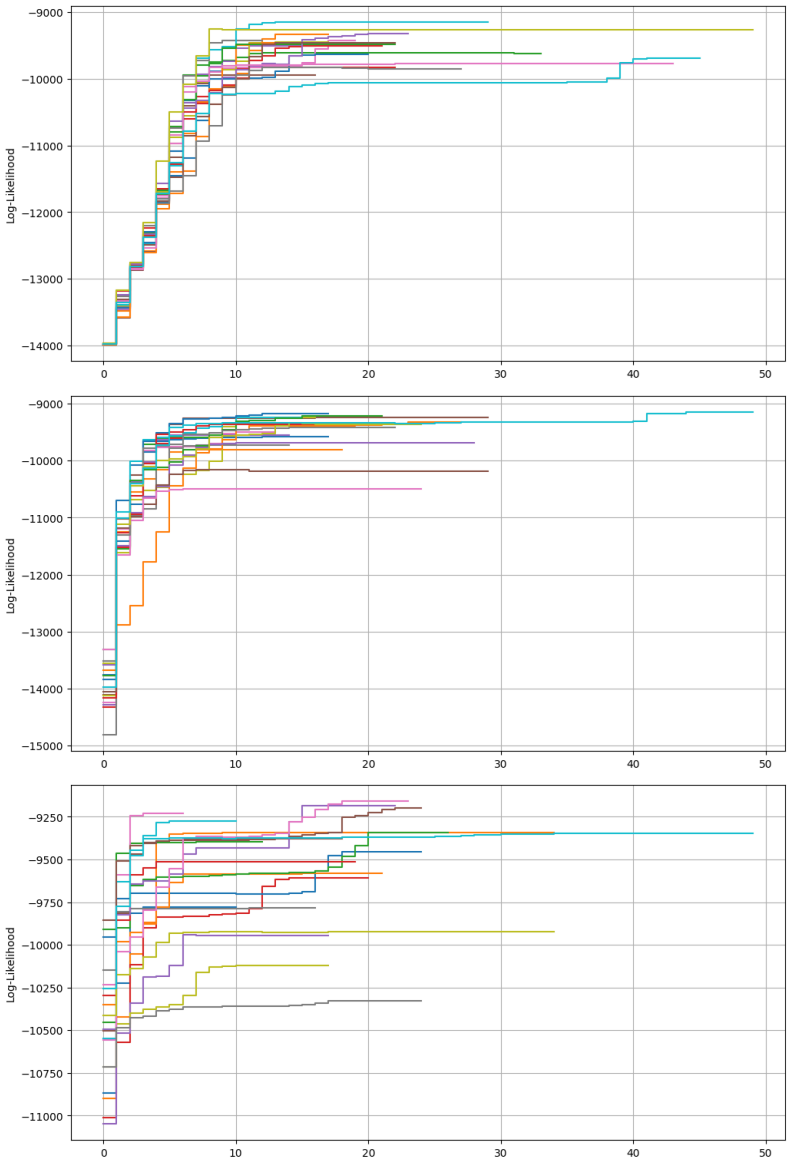

Figure 7: Log Likelihood track under Random (up), Sparse (middle) and Kmeans (down) initializations for *sp-school-day-2* graph

Criteria for the number of clusters

$$\mathrm{AIC}(m_Q) = \min_{\theta} -2 \times \mathcal{L}(\mathcal{X} \mid \theta, m_Q) + Q(Q+3)$$

$$\begin{aligned}\mathrm{BIC}(m_Q) = &\min_{\theta} -2 \times \mathcal{L}(\mathcal{X} \mid \theta, m_Q) \\ &+ 2 \times \frac{Q(Q+3)}{2} \log \frac{n(n-1)}{2}\end{aligned}$$

$$\begin{aligned}\mathrm{ICL}(m_Q) = &\max_{\theta} \mathcal{L}(\mathcal{X} \mid \theta, m_Q) - \frac{Q-1}{2} \log(n) \\ &- \frac{1}{2} \times \frac{Q(Q+1)}{2} \log \frac{n(n-1)}{2}\end{aligned}$$

Algorithm 3 Basic Selection Algorithm

$Res, max_{ICL} = Q_{min}, ICL(m_{Q_{min}})$
for $Q = Q_{min}, Q_{min+1}, \ldots, Q_{max}$ **do**
  Compute an approximation of $ICL(m_Q)$ by using the EM algorithm.
  if $ICL(m_Q) > max_{ICL}$ **then**
    $Res, max_{ICL} = Q, ICL(m_Q)$
  end if
end for
Return *Res*

Algorithm 4 Greedy Algorithm Côme & Latouche [14]

Initialize the $N \times K$ matrix Z; $K = K_{up}$
$\Delta_{g \to h} =$ Change in ICL by assigning node of cluster g to h
Compute the necessary parameter estimates;
while `stop` $\neq 1$ **do**
    $V \leftarrow \{1, \ldots, N\}$;
    `stop` $\leftarrow 1$;
    while V is not empty **do**
        Randomly select a node $i \in V$;
        Remove i from V;
        if i is in cluster g, compute all terms $\Delta_{g \to h}, \forall h \neq g$; **then**
            if at least one $\Delta_{g \to h}$ is positive **then**
                `stop` $\leftarrow 0$;
                Find h such that $\Delta_{g \to h}$ is maximum;
                Swap labels of i: $Z_{ig} \leftarrow 0$ and $Z_{ih} \leftarrow 1$;
                if g is empty **then**
                    Remove column g in Z;
                    $K \leftarrow K - 1$;
                end if
                Update the parameter estimates w.r.t g, h;
            end if
        end if
    end while
end while
Result: (Z, K);

Normalized Mutual Information (NMI)

The *Normalized Mutual Information* (NMI) is a metric used to compare the similarity between two cluster assignments. Given two partitions of a graph, C_1 and C_2, the NMI is defined as:

$$\text{NMI}(C_1, C_2) = \frac{2 \cdot I(C_1; C_2)}{H(C_1) + H(C_2)},$$

where:

- $I(C_1; C_2)$ is the *mutual information* between the two partitions:

$$I(C_1; C_2) = \sum_{c_1 \in C_1} \sum_{c_2 \in C_2} P(c_1, c_2) \log \frac{P(c_1, c_2)}{P(c_1)P(c_2)}.$$

- $H(C)$ is the *entropy* of partition C:

$$H(C) = -\sum_{c \in C} P(c) \log P(c),$$

  where $P(c)$ is the probability of a node being assigned to cluster c, and $P(c_1, c_2)$ is the joint probability of a node being assigned to cluster c_1 in C_1 and c_2 in C_2.

NMI ranges from 0 to 1, where 1 indicates perfect agreement between the two cluster assignments.

Modularity

Modularity is a measure that quantifies the quality of a graph clustering (or community structure) by comparing the density of edges within clusters to what would be expected in a random graph. It is defined as:

$$Q = \frac{1}{2m} \sum_{i,j} \left(A_{ij} - \frac{k_i k_j}{2m} \right) \delta(c_i, c_j),$$

where:

- A_{ij} is the adjacency matrix of the graph (1 if there is an edge between nodes i and j, 0 otherwise).
- $k_i = \sum_j A_{ij}$ is the degree of node i.
- $m = \frac{1}{2} \sum_{i,j} A_{ij}$ is the total number of edges in the graph.
- $\delta(c_i, c_j)$ is the Kronecker delta function, equal to 1 if nodes i and j belong to the same cluster, and 0 otherwise.

Modularity Q typically ranges from -1 to 1, where higher values indicate better clustering. A modularity of 0 means that the density of edges within clusters is no higher than it would be in a graph where edges are distributed randomly according to the degree distribution of the nodes.

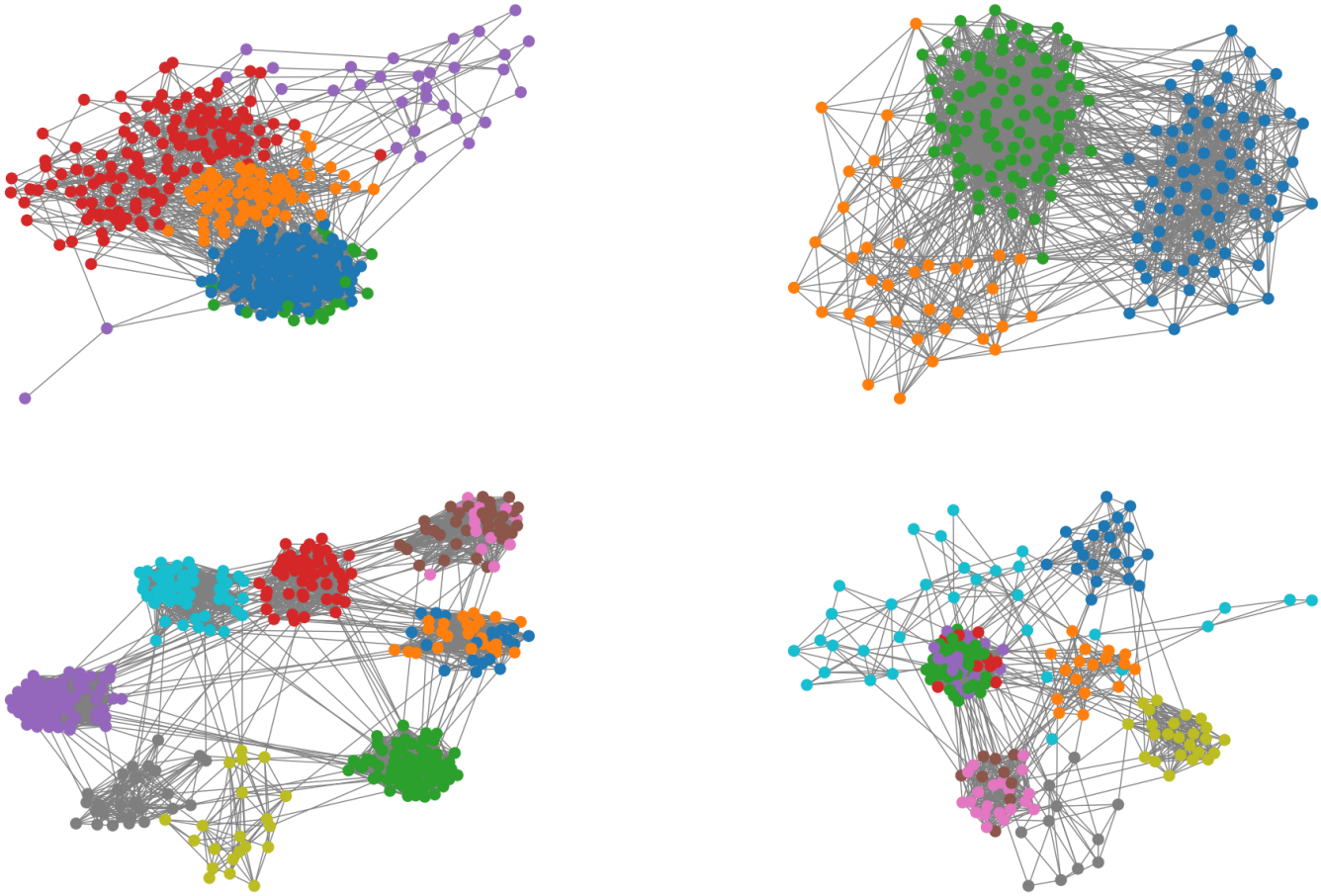

Figure 8: Predictions on synthetic graphs.

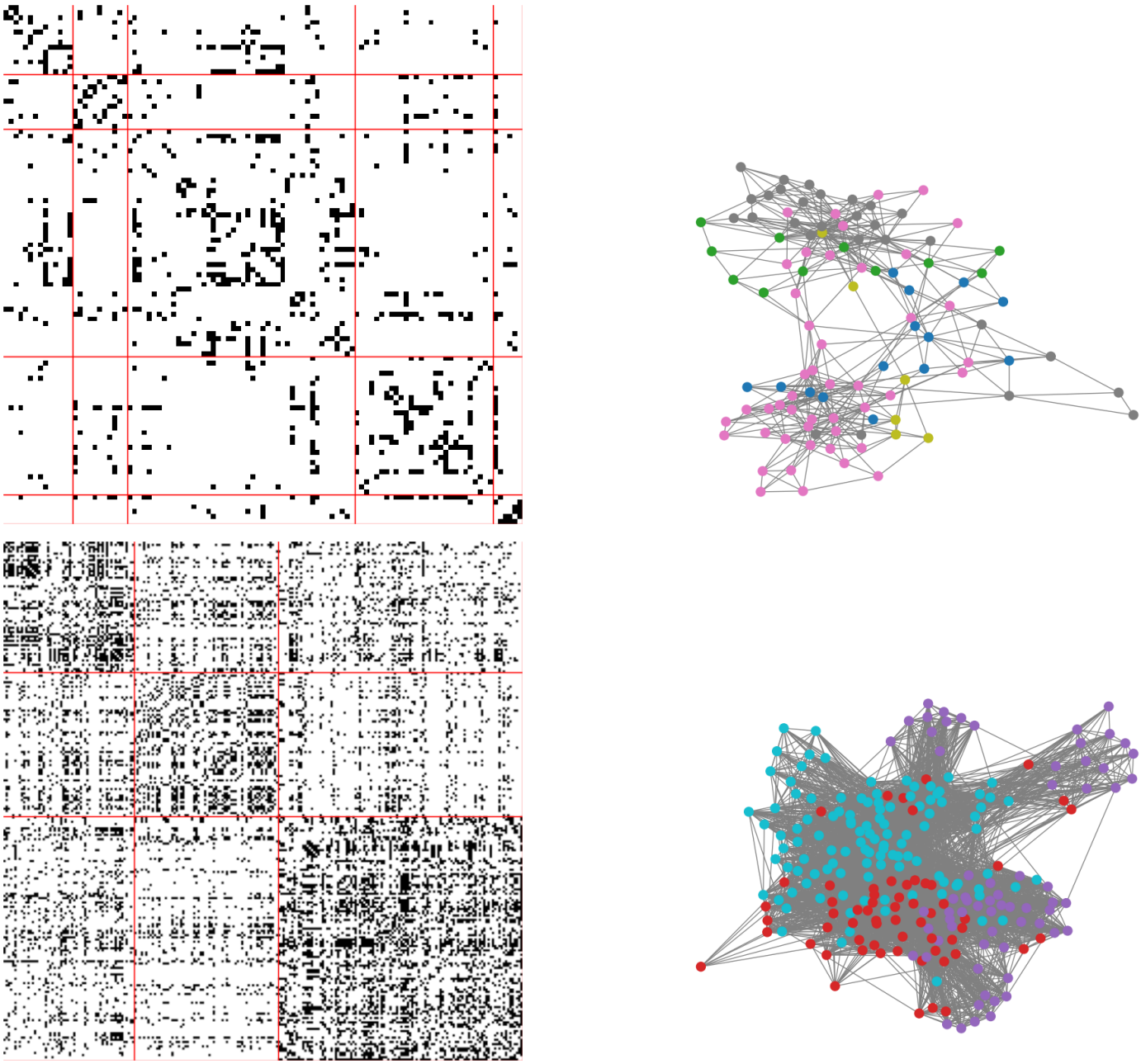

Figure 9: Predictions VBMod on real Graphs.

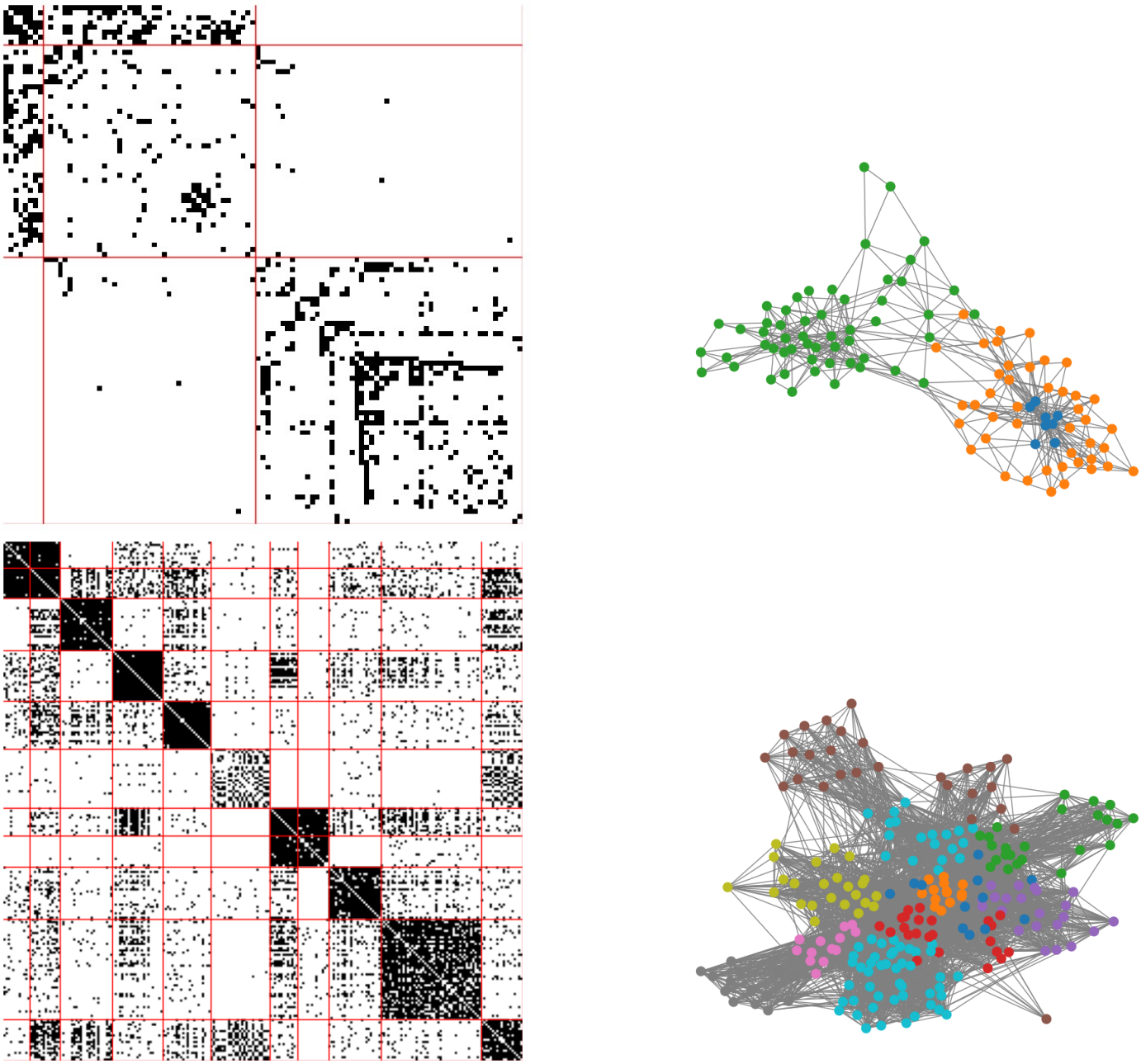

Figure 10: Predictions VBMod on real Graphs.