Graphs in machine learning: an introduction

Pierre Latouche and Fabrice Rossi

Université Paris 1 Panthéon-Sorbonne - Laboratoire SAMM EA 4543
90 rue de Tolbiac, F-75634 Paris Cedex 13 - France

Abstract. Graphs are commonly used to characterise interactions between objects of interest. Because they are based on a straightforward formalism, they are used in many scientific fields from computer science to historical sciences. In this paper, we give an introduction to some methods relying on graphs for learning. This includes both unsupervised and supervised methods. Unsupervised learning algorithms usually aim at visualising graphs in latent spaces and/or clustering the nodes. Both focus on extracting knowledge from graph topologies. While most existing techniques are only applicable to static graphs, where edges do not evolve through time, recent developments have shown that they could be extended to deal with evolving networks. In a supervised context, one generally aims at inferring labels or numerical values attached to nodes using both the graph and, when they are available, node characteristics. Balancing the two sources of information can be challenging, especially as they can disagree locally or globally. In both contexts, supervised and unsupervised, data can be relational (augmented with one or several global graphs) as described above, or graph valued. In this latter case, each object of interest is given as a full graph (possibly completed by other characteristics). In this context, natural tasks include graph clustering (as in producing clusters of graphs rather than clusters of nodes in a single graph), graph classification, etc.

1 Real networks

One of the first practical studies on graphs can be dated back to the original work of Moreno [51] in the 30s. Since then, there has been a growing interest in graph analysis associated with strong developments in the modelling and the processing of these data. Graphs are now used in many scientific fields. In Biology [54, 2, 7], for instance, metabolic networks can describe pathways of biochemical reactions [41], while in social sciences networks are used to represent relation ties between actors [66, 56, 36, 34]. Other examples include powergrids [71] and the web [75]. Recently, networks have also been considered in other areas such as geography [22] and history [59, 39]. In machine learning, networks are seen as powerful tools to model problems in order to extract information from data and for prediction purposes. This is the object of this paper. For more complete surveys, we refer to [28, 62, 49, 45].

In this section, we introduce notations and highlight properties shared by most real networks. In Section 2, we then consider methods aiming at extracting information from a unique network. We will particularly focus on clustering methods where the goal is to find clusters of vertices. Finally, in Section 3, techniques that take a series of networks into account, where each network is

seen as an object, are investigated. In particular, distances and kernels for graphs are discussed.

1.1 Notations

A graph is first characterised by a set $\mathcal{V}$ of N vertices and a set $\mathcal{E}$ of edges between pairs of vertices. The graph is said to be directed if the pairs (i, j) in $\mathcal{E}$ are ordered, undirected otherwise. A graph with self loops is made of vertices which can be connected to themselves. The degree of a vertex i is the total number of edges connected to i, with self loops counted twice. In most applications, only the presence or absence of an edge is characterised. However, edges can also be weighted by a function $h : \mathcal{E} \to \mathbb{F}$ for any set $\mathbb{F}$. More generally arbitrary labelling functions can be defined on both the vertices and the edges, leading to labelled graphs.

A graph is usually described by an $N \times N$ adjacency matrix $(\mathbf{X})_{ij}$ where X_{ij} is the value associated to the edge between the (i, j) pair. It is equal to zero in the absence of relationship between the nodes. In the case of binary graphs, the matrix $\mathbf{X}$ is binary and $X_{ij} = 1$ indicates that the two vertices are connected. If the graph is directed then $\mathbf{X}$ is symmetric that is $X_{ij} = X_{ji}$ for all (i, j).

We use interchangeably the vocabulary from graph theory introduced above and a less formal vocabulary in with a graph is called a network and a vertex a node. In general, the network is the real world object while the graph is its mathematical representation, but we have a more relaxed use of the terms.

1.2 Properties

A remarkable characteristic of most real networks is that they share common properties [20, 3, 67, 54]. First, most of them are sparse *i.e.* the number of edges present in not quadratic in the number of vertices, but linear. Thus, the mean degree remains bounded when N increases and the network density, defined as the ratio between the number of existing edges over the number of potential edges, tends to zero. Second, while some vertices of a real network can have few connections or no connection at all with the other vertices, most vertices belong to a single component, so called *giant component*, where it is always possible to find a path, *i.e.* a set of adjacent connected edges, connecting any pair of nodes. Nodes can be disconnected from this component, forming significantly smaller components. Finally, we would like to highlight the degree heterogeneity and small world properties. The first property states that few vertices have a lot of links, while most of the vertices have few connections. Therefore, scale free distributions are often considered to model the degrees [6, 15]. The second one indicates that the shortest path from one vertex to another is generally rather small, typically of size $O(\log(N))$.

2 Graph clustering

In order to extract information from a unique graph, unsupervised methods usually look for cluster of vertices sharing similar connection profiles, a particular case of general vertices clustering [63]. They differ in the way they define the topology on top of which clusters are built.

2.1 Community structure

Most graph clustering algorithms aim at uncovering specific types of clusters, so called communities, where there are more edges between vertices of the same community than between vertices of different communities. Thus, communities appear in the form of densely connected clusters of vertices, with sparser connections between groups. They are characterised by the *friend of my friend is my friend* effect, *i.e.* a transitivity property, also called assortative mixing effect. Two families of methods for community discovering can be singled out among a vast set of methods [24], depending on wether they maximize a score derived from the modularity score of Girvan and Newman [27] or rely on the latent position cluster model (LPCM) of Handcock, Raftery and Tantrum [34].

2.1.1 Modularity score

A series of community detection algorithms have been proposed (see for instance [27, 52, 53] and the survey [24]). They involve iterative removal of edges from the network to detect communities where candidate edges for removal are chosen according to betweenness measures. All measures rely on the same idea that two communities, by definition, are joined by a few edges and therefore, all paths from vertices in one community to vertices in the other are likely to path along these few edges. Therefore, the number of paths that go along an edge is expected to be larger for inter community edges. For instance, the edge betweenness of an edge does account for the number of shortest paths between all pairs of vertices that run along that edge. Moreover, the random walk betweenness evaluates the expected number of times a random walk would path along the edge, for all pairs of vertices.

The iterative removal of edges produces a dendrogram, describing a hierarchical structure, from a situation where each vertex belongs to a different cluster to the inverse scenario where all vertices are clustered within the same community. The modularity score [27] is then considered to select a particular division of the network into K clusters. Denoting e_{kl} the fraction of edges in the network connecting vertices of communities k and l, as well as $a_k = \sum_{l=1}^{K} a_{kl}$, the fraction of edges that connect with vertices of community k, the modularity score is given by:

$$K_{mod} = \sum_{k=1}^{K} (e_{kk} - a_k^2).$$

Such a criterion is computed for the different levels in the hierarchy and K is chosen such that K_{mod} is maximised.

Rather that building the complete dendrogram, other algorithms have focused on optimising the modularity score directly, as it is beneficial both in computational terms and in the perceived quality of the obtained partitions. A very popular algorithm, the so-called Louvain method [10], proceeds by a series of greedy exchanges and merging that turns a fully refined partition into a coarser one that provides a (local) maximum of the modularity. Better solutions can be obtained using more sophisticated heuristics [55] but maximising the modularity is a NP-hard problem [12].

Note that modularity approaches have been shown to be asymptotically biased [9]. To tackle this issue, degree corrected methods were introduced in order to take the degrees of nodes into account.

2.1.2 Latent position cluster model

Alternative approaches, looking for clusters of vertices with assortative mixing, usually rely on the LPCM model [34] which is a generalisation of the latent position model (LPM) [36]. In the original LPM model, each vertex i is first assumed to be associated with a position $\mathbf{Z}_i$ in a Euclidean latent space $\mathbb{R}^d$. Each edge between a pair (i, j) of vertices is then drawn depending on $\mathbf{Z}_i$ and $\mathbf{Z}_j$. Both maximum likelihood and Markov chain Monte Carlo (MCMC) techniques were considered to estimate the model parameters and the latent positions. The corresponding mapping of the vertices into the Euclidean latent space produces a representation of the network such that nodes which are likely to be connected have similar positions. Note that if the latent space is low dimensional, typically of dimension $d = 1, 2, 3$, then the representation can be visualised which is feature appreciated by practitioners.

The LPM model was extended in order to look for both a representation of the network and a clustering of the vertices. Thus, the corresponding LPCM model assumes that the positions are drawn from a Gaussian mixture model in the latent space such that each Gaussian distribution corresponds to a cluster. A two stage maximum likelihood approach along with a Bayesian MCMC scheme were proposed for inference purposes. Moreover, conditional Bayes factors were considered to estimate the number of clusters from the data. Finally variational bayesian inference is also possible [61].

2.2 Heterogeneous structures

So far, we have discussed methods looking exclusively for communities in networks. Other approaches usually derive from the stochastic block model (SBM) of Nowicki and Snijders [56]. They can also look for communities, but not only.

The SBM models assumes that nodes are spread in unknown clusters and that the probability of a connection between two nodes i and j depends on their corresponding clusters. In practice, a latent vector $\mathbf{Z}_i$ is drawn from a multinomial distribution with parameters $(1, \boldsymbol{\alpha} = \{\alpha_1, \ldots, \alpha_K\})$, where α_k is

the proportion of cluster k. Therefore, $\mathbf{Z}_i$ is a binary vector of size K with a single 1, such that $Z_{ik} = 1$ indicates that i belongs to cluster k, 0 otherwise. If i is in cluster k and j in l, then the SBM model assumes that there is a probability π_{kl} of a connection between the two nodes. All connection probabilities are characterised by a $K \times K$ matrix $\mathbf{\Pi}$. Note that a community structure can be defined by setting values for the diagonal terms of $\mathbf{\Pi}$ to higher values than extra diagonal terms [37]. In practice, because no assumptions are made regarding $\mathbf{\Pi}$, the SBM model can take heterogeneous structures into account [18, 42, 44].

While generating a network with such a sampling scheme is straightforward, estimating the model parameters $\boldsymbol{\alpha}$ and $\mathbf{\Pi}$ as well as the set $(\mathbf{Z})_i$ of all latent vectors is challenging. One of the key issue is that the posterior distribution of $\mathbf{Z}$ given the adjacency matrix $\mathbf{X}$ and the model parameters $(\boldsymbol{\alpha}, \mathbf{\Pi})$ cannot be factorised due to conditional dependency. Therefore, standard optimisation algorithms, such as the expectation maximisation (EM) algorithm, cannot be derived. To tackle this issue variational and stochastic approximations have been proposed. Thus, [18] relied on a variational EM (VEM) algorithm whereas [44] used a variational Bayes EM (VBEM) approach. Alternatively, [56] estimated the posterior distribution of the model parameters and $\mathbf{Z}$, given $\mathbf{X}$, by considering Gibbs sampling.

A even more fondamental question concerns the estimation of the number of clusters present in the data. Unfortunately, since the likelihood is not tractable either, standard model selection criteria, like the Akaike information criterion (AIC) or the Bayesian IC (BIC) cannot be computed. Again, variational along with asymptotic Laplace approximations were derived to obtain approximate model selection criteria [18, 44].

In some cases, the clustering of the nodes and the estimation of the number of clusters are performed at the same time using allocation sampler [50], greedy search [16], or non parametric schemes [40].

2.3 Extensions

Since the original development of the SBM model, many extensions have been proposed to deal for instance with valued edges [48] or to take into account covariate information [76, 49]. The random subgraph model (RSM) [39] for instance assumes that a partition of the nodes into subgraphs is observed and that the subgraphs are made of (unknown) latent clusters, as in the SBM model, with various mixing proportions. The edges are typed. In parallel, strategies looking for overlapping clusters, where each node can belong to multiple clusters, have been derived. In [1], a vertex i belongs a cluster in its relation with a given vertex j. Because i is involved in multiple relations in the network, it can belong to more than one cluster. In [43], the multinomial distribution of the SBM model is replaced with a product of Bernoulli distribution, allowing each vertex to belong to no, one, or several clusters.

In the last few years, a lot of attention has been paid on extending the approaches mentioned previously in order to deal with dynamic networks where nodes and/or edges can evolve through time. The main idea consists in in-

troducing temporal processes, such as hidden Markov model (HMM) or linear dynamic systems [72, 74, 73]. While models usually focus on modelling the dynamic of networks through the evolution of their latent structures, Heaukulani and Gharamani [35] chose to define how observed social interactions can affect future unobserved latent structures. We would also like to highlight the work of Dubois, Butts, and P. Smyth [21]. Contrary to most dynamic clustering approaches, they considered a non homogeneous Poisson process allowing to deal with a continuous time periods where events, *i.e.* the creation or removal of an edge, can occur one at a time. Another approach for graph clustering in the continuous time context is provided by [29] which builds a coclustering structure on the vertices of the graph and on the time stamps of the edges.

3 Multiple graphs

While a large part of the graph related literature in machine learning targets the case of a single graph, numerous applications lead naturally to data sets made of graphs, that is situations in which each data point is a graph (or consists in several components including at least one graph). This is the case for instance in chemistry where molecules can be represented by undirected labelled graphs (see e.g. [57]) and in biology where the structure of a protein can be represented by a graph that encodes neighborhoods between it fragments as in [11]. In fact, the use of graphs as structured representations of complex data follows a long tradition with early examples appearing in the late seventies [60] and with a tendency to become pervasive in the last decade.

It should be noted that even in the case of a single global graph described in the first part of this paper, it is quite natural to study multiple graphs derived from the global one, in particular via the ego-centered approach which is very common in social sciences (see e.g. [26]). The main idea is to extract from a large social network a set of small networks centered on each of the vertices under study. For real world social networks, it is in general the only possible course of action, the whole network being impossible to observe (see e.g. [19] for an example).

When dealing with multiple graphs, one tackles the traditional tasks of machine learning, from unsupervised problems (clustering, frequent patterns analysis, etc.) to supervised ones (classification, regression, etc.). There are two main tendencies in the literature: the design of specialized methods obtained by adapting classical ones to graphs and the use of distances and kernels coupled with generic methods.

3.1 Specialized methods

As graphs are not vector data, classical machine learning techniques do not apply directly. Numerous methods have been adapted in rather specific ways to handle graphs and other non vector data, especially in the neural network community [32, 17], for instance via recursive neural networks as in [33, 30]. In those approaches, each graph is processed vertex by vertex, by leveraging the

structure to build a form of abstract time. The recursive model maintains an implicit knowledge of the vertices already processed by means of its space state neurons.

3.2 Distances and kernels

A somewhat more generic solution consists in building distances (or dissimilarities) between graphs and then in using distances based methods (such as methods based on the so-called relational approach [31]). One difficulty is that graph isomorphism should be accounted for when two graphs are compared: two graphs are isomorphic if they are equal up to a relabeling of their vertices. Any sound dissimilarity/distance between graphs should detect isomorphic graphs. This is however far more complex than expected [23] up to a point that the actual complexity class of the graph isomorphism problem remains unknown (it belongs to the NP class but not to the NP-complete class, for instance). While exact algorithms appear fast on real world graphs, their worst case complexities are exponential, with a best bound in $O(2^{\sqrt{n \log n}})$ [47]. In addition, subgraph isomorphism, i.e. determining whether a given graph contains a subgraph that is isomorphic to another graph is NP-complete.

Nevertheless, numerous exact or approximate algorithms have been defined to try and solve the (sub)graph isomorphism problem (see e.g. [14]). In particular, it has been shown that those problems (and related ones) are special cases of the computation of the graph edit distance [13], a generalization of the string edit distance [46]. The graph edit distance is defined by first introducing edition operations on graph, such as insertion, deletion and substitution of edges and vertices (labels and weights included). Each operation is assigned a numeric cost. The total cost of a series of operations is simply the sum of the individual costs. Then the graph edit distance between two graphs is the cost of the least costly sequence of operations that transforms one of the graph into the other one (see [25] for a survey).

A rather different line of research has provided a set of tools to compare graphs by means of kernels (as in reproducing kernels [4]). Those symmetric positive definite similarity functions allow one to generalize any classical vector space method to non vector data in a straightforward way [64]. Numerous of such kernels have been defined to compare two graphs [69]. Most of them are based on random walks that take place on the product of the two graphs under comparison.

Once a kernel or a distance has been chosen, one can apply any of the kernel methods [65] or of the relational methods [31], which gives access to support vector machine, kernel ridge regression and kernel k-means to cite only a few.

4 Conclusion

This paper has only scraped the surface of the vast literature about graphs in machine learning. Complete area of graph applications in machine learning were ignored.

For instance, it is well know that extracting a neighborhood graph from a classical vector data set is an efficient way to get insights on the topology of the data set. This has led to numerous interesting applications ranging from visualization (as in isomap [68] and its successors) to semi-supervised learning [8], going through spectral clustering [70] and exploratory analysis of labelled data sets [5].

Another interesting area concerns the so-called relational data framework when a classical data set is augmented with a graph structure: the vertices of the graph are elements of a standard vector space and are thus traditional data points, but they are interconnected via a graph structure (or several ones in complex settings). The challenge consists here in taking into account the graph structure while processing the classical data or vice-versa in taking into account the data point descriptions when processing the graph. Among other issues, those two different sources of information can be contradictory for a given task. A typical application of such a framework consists in annotating nodes on social media [38].

While we have presented some temporal extensions of classical graph related problems, we have ignored most of them. For instance, the issue of information propagation on graphs has received a lot of attention [58]. Among other tasks, machine learning can be used e.g. to predict the probability of passing information from one actor to another as in [19].

More generally, the massive spread in the last decade of online social networking has the obvious consequence of generating very large relational data sets. While non vector data have been studied for quite a long time, those new data sets push the complexity one step further by mixing several types of non vector data. Objects under study are now described by complex mixed data (texts, images, etc.) and are related by several networks (friendship, online discussion, etc.). In addition, the temporal dynamic of those data cannot be easily ignored or summarized. It seems therefore that the next set of problems faced by the machine learning community will include graphs in numerous forms, including dynamic ones.

References

[1] E. Airoldi, D. Blei, S. Fienberg, and E. Xing. Mixed membership stochastic blockmodels. *The Journal of Machine Learning Research*, 9:1981–2014, 2008.

[2] R. Albert and A. Barabási. Statistical mechanics of complex networks. *Modern Physics*, 74:47–97, 2002.

[3] L. Amaral, A. Scala, M. Barthélémy, and H. Stanley. Classes of small-world networks. In *Proceedings of the National Academy of Sciences*, volume 97, pages 11149–11152, 2000.

[4] N. Aronszajn. Theory of reproducing kernels. *Transactions of the American Mathematical Society*, 68(3):337–404, May 1950.

[5] M. Aupetit and T. Catz. High-dimensional labeled data analysis with topology representing graphs. *Neurocomputing*, 63:139–169, 2005.

[6] A. Barabasi and R. Albert. Emergence of scaling in random networks. *Science*, 286:509–512, 1999.

[7] A. Barabási and Z. Oltvai. Network biology: understanding the cell's functional organization. *Nature Rev. Genet*, 5:101–113, 2004.

[8] M. Belkin, P. Niyogi, and V. Sindhwani. Manifold regularization: A geometric framework for learning from labeled and unlabeled examples. *The Journal of Machine Learning Research*, 7:2399–2434, 2006.

[9] P. Bickel and A. Chen. A nonparametric view of network models and newman–girvan and other modularities. *Proceedings of the National Academy of Sciences*, 106(50):21068–21073, 2009.

[10] V. D. Blondel, J.-L. Guillaume, R. Lambiotte, and E. Lefebvre. Fast unfolding of communities in large networks. *Journal of Statistical Mechanics: Theory and Experiment*, 2008(10):P10008 (12pp), 2008.

[11] K. M. Borgwardt, C. S. Ong, S. Schönauer, S. Vishwanathan, A. J. Smola, and H.-P. Kriegel. Protein function prediction via graph kernels. *Bioinformatics*, 21(suppl 1):i47–i56, 2005.

[12] U. Brandes, D. Delling, M. Gaertler, R. Gorke, M. Hoefer, Z. Nikoloski, and D. Wagner. On modularity clustering. *Knowledge and Data Engineering, IEEE Transactions on*, 20(2):172–188, 2008.

[13] H. Bunke. Error correcting graph matching: on the influence of the underlying cost function. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 21(9):917–922, Sep 1999.

[14] H. Bunke. Graph matching: Theoretical foundations, algorithms, and applications. In *In Proceedings of Vision Interface*, pages 82–88, Montreal, 2000.

[15] A. Clauset, C. R. Shalizi, and M. E. J. Newman. Power-law distributions in empirical data. *SIAM Review*, 51(4):661–703, 2009.

[16] E. Côme and P. Latouche. Model selection and clustering in stochastic block models with the exact integrated complete data likelihood. *ArXiv e-prints*, Mar. 2013.

[17] M. Cottrell, M. Olteanu, F. Rossi, J. Rynkiewicz, and N. Villa-Vialaneix. Neural networks for complex data. *KI - Künstliche Intelligenz*, 26:373–380, 11 2012.

[18] J.-J. Daudin, F. Picard, and S. Robin. A mixture model for random graphs. *Statistics and Computing*, 18(2):173–183, 2008.

[19] C. Dhanjal, S. Blanchemanche, S. Clémençon, A. Rona-Tas, and F. Rossi. Dissemination of health information within social networks. In B. Vedres and M. Scotti, editors, *Networks in Social Policy Problems*, pages 15–46. Cambridge University Press, 10 2012.

[20] S. Dorogovtsev, J. Mendes, and A. Samukhin. Structure of growing networks with preferential linking. *Physical Review Letter*, 85:4633–4636, 2000.

[21] C. Dubois, C. Butts, and P. Smyth. Stochastic blockmodelling of relational event dynamics. In *International Conference on Artificial Intelligence and Statistics*, volume 31 of the Journal of Machine Learning Research Proceedings, pages 238–246, 2013.

[22] C. Ducruet. Network diversity and maritime flows. *Journal of Transport Geography*, 30:77–88, 2013.

[23] S. Fortin. The graph isomorphism problem. Technical report, Technical Report 96-20, University of Alberta, Edomonton, Alberta, Canada, 1996.

[24] S. Fortunato. Community detection in graphs. *Physics Reports*, 486(3-5):75 – 174, 2010.

[25] X. Gao, B. Xiao, D. Tao, and X. Li. A survey of graph edit distance. *Pattern Analysis and applications*, 13(1):113–129, 2010.

[26] L. Garton, C. Haythornthwaite, and B. Wellman. Studying online social networks. *Journal of Computer-Mediated Communication*, 3(1):0–0, 1997.

[27] M. Girvan and M. Newman. Community structure in social and biological networks. *Proceedings of the National Academy of Sciences*, 99(12):7821, 2002.

[28] A. Goldenberg, A. Zheng, and S. Fienberg. *A survey of statistical network models*. Now Publishers, 2010.

[29] R. Guigourès, M. Boullé, and F. Rossi. A triclustering approach for time evolving graphs. In *Co-clustering and Applications, IEEE 12th International Conference on Data Mining Workshops (ICDMW 2012)*, pages 115–122, Brussels, Belgium, 12 2012.

[30] M. Hagenbuchner, A. Sperduti, and A. Tsoi. Graph self-organizing maps for cyclic and unbounded graphs. *Neurocomputing*, 72(7-9):1419 – 1430, 2009. Advances in Machine Learning and Computational Intelligence 16th European Symposium on Artificial Neural Networks 2008 16th European Symposium on Artificial Neural Networks 2008.

[31] B. Hammer, A. Hasenfuss, F. Rossi, and M. Strickert. Topographic processing of relational data. In *Proceedings of the 6th International Workshop on Self-Organizing Maps (WSOM 07)*, Bielefeld (Germany), 9 2007.

[32] B. Hammer and B. J. Jain. Neural methods for non-standard data. In *Proceedings of XIIth European Symposium on Artificial Neural Networks (ESANN 2004)*, pages 281–292, Bruges (Belgium), April 2004.

[33] B. Hammer, A. Micheli, A. Sperduti, and M. Strickert. A general framework for unsupervised processing of structured data. *Neurocomputing*, 57:3–35, March 2004.

[34] M. Handcock, A. Raftery, and J. Tantrum. Model-based clustering for social networks. *Journal of the Royal Statistical Society: Series A (Statistics in Society)*, 170(2):301–354, 2007.

[35] C. Heaukulani and Z. Ghahramani. Dynamic probabilistic models for latent feature propagation in social networks. In *Proceedings of the 30th International Conference on Machine Learning (ICML-13)*, pages 275–283, 2013.

[36] P. Hoff, A. Raftery, and M. Handcock. Latent space approaches to social network analysis. *Journal of the American Statistical Association*, 97(460):1090–1098, 2002.

[37] J. Hofman and C. Wiggins. Bayesian approach to network modularity. *Physical review letters*, 100(25):258701, 2008.

[38] Y. Jacob, L. Denoyer, and P. Gallinari. Classification and annotation in social corpora using multiple relations. In *Proceedings of the 20th ACM international conference on Information and knowledge management*, pages 1215–1220. ACM, 2011.

[39] Y. Jernite, P. Latouche, C. Bouveyron, P. Rivera, L. Jegou, and S. Lamassé. The random subgraph model for the analysis of an acclesiastical network in merovingian gaul. *Annals of Applied Statistics*, 2013.

[40] C. Kemp, J. Tenenbaum, T. Griffiths, T. Yamada, and N. Ueda. Learning systems of concepts with an infinite relational model. In *Proceedings of the National Conference on Artificial Intelligence*, volume 21, page 381, 2006.

[41] V. Lacroix, C. Fernandes, and M.-F. Sagot. Motif search in graphs: pplication to metabolic networks. *Transactions in Computational Biology and Bioinformatics*, 3:360–368, 2006.

[42] P. Latouche, E. Birmelé, and C. Ambroise. *Advances in Data Analysis Data Handling and Business Intelligence*, chapter Bayesian methods for graph clustering, pages 229–239. Springer, 2009.

[43] P. Latouche, E. Birmelé, and C. Ambroise. Overlapping stochastic block models with application to the french political blogosphere. *Annals of Applied Statistics*, 5(1):309–336, 2011.

[44] P. Latouche, E. Birmelé, and C. Ambroise. Variational bayesian inference and complexity control for stochastic block models. *Statistical Modelling*, 12(1):93–115, 2012.

[45] P. Latouche, E. Birmelé, and C. Ambroise. *The Handbook on Mixed Membership Models and Their Applications*, chapter Overlapping clustering methods for networks, pages 547–567. Chapman and Hall/CRC, 2014.

[46] V. I. Levenshtein. Binary codes capable of correcting deletions, insertions and reversals. *Sov. Phys. Dokl.*, 6:707–710, 1966.

[47] E. M. Luks. Isomorphism of graphs of bounded valence can be tested in polynomial time. *Journal of Computer and System Sciences*, 25(1):42 – 65, 1982.

[48] M. Mariadassou, S. Robin, and C. Vacher. Uncovering latent structure in valued graphs: a variational approach. *Annals of Applied Statistics*, 4(2):715–742, 2010.

[49] C. Matias and S. Robin. Modeling heterogeneity in random graphs through latent space models: a selective review. *Esaim Proc. and Surveys, 47 :55-74*, 2014.

[50] A. Mc Daid, T. Murphy, F. N., and N. Hurley. Improved bayesian inference for the stochastic block model with application to large networks. *Computational Statistics and Data Analysis*, 60:12–31, 2013.

[51] J. Moreno. *Who shall survive?: A new approach to the problem of human interrelations.* Nervous and Mental Disease Publishing Co, 1934.

[52] M. Newman. Fast algorithm for detecting community structure in networks. *Physical Review Letter*, 69, 2004.

[53] M. Newman and M. Girvan. Finding and evaluating community structure in networks. *Physical Review E*, 69:026113, 2004.

[54] M. E. J. Newman. The structure and function of complex networks. *SIAM Review*, 45:167–256, 2003.

[55] A. Noack and R. Rotta. Multi-level algorithms for modularity clustering. In *SEA '09: Proceedings of the 8th International Symposium on Experimental Algorithms*, pages 257–268, Berlin, Heidelberg, 2009. Springer-Verlag.

[56] K. Nowicki and T. Snijders. Estimation and prediction for stochastic blockstructures. *Journal of the American Statistical Association*, 96(455):1077–1087, 2001.

[57] L. Ralaivola, S. J. Swamidass, H. Saigo, and P. Baldi. Graph kernels for chemical informatics. *Neural Networks*, 18(8):1093 – 1110, 2005. Neural Networks and Kernel Methods for Structured Domains.

[58] M. G. Rodriguez, J. Leskovec, D. Balduzzi, and B. Schölkopf. Uncovering the structure and temporal dynamics of information propagation. *Network Science*, 2(01):26–65, 2014.

[59] F. Rossi, N. Villa-Vialaneix, and F. Hautefeuille. Exploration of a large database of French notarial acts with social network methods. *Digital Medievalist*, 9, 7 2014.

[60] D. H. Rouvray and A. T. Balaban. Chemical applications of graph theory. In R. J. Wilson and L. W. Beineke, editors, *Applications of Graph Theory*, pages 177–221. Acadmic Press, London, 1979.

[61] M. Salter-Townshend and T. B. Murphy. Variational bayesian inference for the latent position cluster model for network data. *Computational Statistics & Data Analysis*, 57(1):661–671, 2013.

[62] M. Salter-Townshend, A. White, I. Gollini, and T. B. Murphy. Review of statistical network analysis: models, algorithms, and software. *Statistical Analysis and Data Mining*, 5(4):243–264, 2012.

[63] S. E. Schaeffer. Graph clustering. *Computer Science Review*, 1(1):27–64, August 2007.

[64] B. Schölkopf and A. Smola. *Learning with Kernels.* MIT Press, Cambridge, MA, 2002.

[65] J. Shawe-Taylor and N. Cristianini. *Kernel Methods for Pattern Analysis.* Cambridge University Press, 2004.

[66] T. Snijders and K. Nowicki. Estimation and prediction for stochastic blockmodels for graphs with latent block structure. *Journal of Classification*, 14(1):75–100, 1997.

[67] S. Strogatz. Exploring complex networks. *Nature*, 410:268–276, 2001.

[68] J. B. Tenenbaum, V. de Silva, and J. C. Langford. A global geometric framework for nonlinear dimensionality reduction. *Science*, 290(22):2319–2323, December 2000. http://isomap.stanford.edu/.

[69] S. V. N. Vishwanathan, N. N. Schraudolph, R. Kondor, and K. M. Borgwardt. Graph kernels. *The Journal of Machine Learning Research*, 11:1201–1242, 2010.

[70] U. Von Luxburg. A tutorial on spectral clustering. *Statistics and computing*, 17(4):395–416, 2007.

[71] D. Watts and S. Strogatz. Collective dynamics of small-world networks. *Nature*, 393:440–442, 1998.

[72] E. Xing, W. Fu, and L. Song. A state-space mixed membership blockmodel for dynamic network tomography. *The Annals of Applied Statistics*, 4(2):535–566, 2010.

[73] K. S. Xu and A. O. Hero III. Dynamic stochastic blockmodels: Statistical models for time-evolving networks. In *Social Computing, Behavioral-Cultural Modeling and Prediction*, pages 201–210. Springer, 2013.

[74] T. Yang, Y. Chi, S. Zhu, Y. Gong, and R. Jin. Detecting communities and their evolutions in dynamic social networks : a bayesian approach. *Machine learning*, 82(2):157–189, 2011.

[75] H. Zanghi, C. Ambroise, and V. Miele. Fast online graph clustering via erdös renyi mixture. *Pattern Recognition*, 41(12):3592–3599, 2008.

[76] H. Zanghi, S. Volant, and C. Ambroise. Clustering based on random graph model embedding vertex features. *Pattern Recognition Letters*, 31(9):830–836, 2010.