

# Identifying graph clusters using variational inference and links to covariance parametrization

David Barber

*Phil. Trans. R. Soc. A* 2009 **367**, 4407-4426

doi: 10.1098/rsta.2009.0117

---

## References

**This article cites 18 articles, 3 of which can be accessed free**

<http://rsta.royalsocietypublishing.org/content/367/1906/4407.full.html#ref-list-1>

## Rapid response

**Respond to this article**

<http://rsta.royalsocietypublishing.org/letters/submit/roypta;367/1906/4407>

## Subject collections

Articles on similar topics can be found in the following collections

[statistics](#) (34 articles)

## Email alerting service

Receive free email alerts when new articles cite this article - sign up in the box at the top right-hand corner of the article or click [here](#)

---

To subscribe to *Phil. Trans. R. Soc. A* go to:

<http://rsta.royalsocietypublishing.org/subscriptions>

---

# Identifying graph clusters using variational inference and links to covariance parametrization

BY DAVID BARBER\*

*Department of Computer Science, University College London,  
London WC1E 6BT, UK*

Finding clusters of well-connected nodes in a graph is a problem common to many domains, including social networks, the Internet and bioinformatics. From a computational viewpoint, finding these clusters or graph communities is a difficult problem. We use a clique matrix decomposition based on a statistical description that encourages clusters to be well connected and few in number. The formal intractability of inferring the clusters is addressed using a variational approximation inspired by mean-field theories in statistical mechanics. Clique matrices also play a natural role in parametrizing positive definite matrices under zero constraints on elements of the matrix. We show that clique matrices can parametrize all positive definite matrices restricted according to a decomposable graph and form a structured factor analysis approximation in the non-decomposable case. Extensions to conjugate Bayesian covariance priors and more general non-Gaussian independence models are briefly discussed.

**Keywords:** graph clustering; community identification; network; variational inference; clique matrix; covariance

## 1. Introduction

A common task in large-scale data analysis concerns the discovery of ‘similar’ objects. Here two objects are similar if they are neighbours on a graph representing the objects; the graph then represents the network of interactions between the objects. The structure of the connections on these graphs or networks has been of intense interest recently, particularly concerning the degree structure of the graph and related small-world phenomena; see, for example, Newman (2003).

In the field of social networks, each individual is represented as a node (vertex) in a graph, with a link (edge) between two nodes if the individuals are friends. Given a potentially very large such graph, our interest is in identifying communities of closely linked friends. A characteristic of such social networks is that they are sparse since each individual will typically have only a small number of friends relative to the total number of people in the network (Kautz *et al.* 1997).

\*d.barber@cs.ucl.ac.uk

One contribution of 11 to a Theme Issue ‘Statistical challenges of high-dimensional data’.

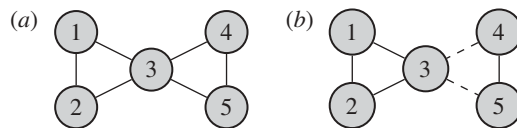


Figure 1. (a) The social network of a set of five individuals represented as an undirected graph. Here individual 3 belongs to the group (1, 2, 3) and also (3, 4, 5). (b) By contrast, in graph partitioning, one breaks the graph into roughly equally sized disjoint partitions such that each node is a member of only a single partition, with a minimal number of edges between partitions.

The field of collaborative filtering also contains related data-analysis challenges. Here nodes may represent products with a link between them, meaning that the two products are frequently purchased together by customers. The identification of ‘product groups’ is often of interest (Goldberg *et al.* 1992).

A growing area of clustering is in bioinformatics, in which nodes represent genes, with a link representing that two genes have similar activity profiles (similar functionality). The task is then to identify groups of similarly functioning genes (Airoldi *et al.* 2008).

Here, we use undirected graphs to represent connectivity structures in the data, and our interest is to decompose the graph into well-connected clusters. Importantly, the same object (product, person and gene) can appear in multiple groups. For example, interpreted as a social network, individual 3 in figure 1a is a member of his work group (1, 2, 3) and also the poker group (3, 4, 5). These two groups of individuals are otherwise disjoint.

Note that graph clustering contrasts with the perhaps more common task of graph partitioning, in which each node is assigned to only one of a set of subgraphs (figure 1b). Typically, the partitioning criterion is that each subgraph should be roughly of the same size and with few connections between the subgraphs (Karypis & Kumar 1998).

Our aim here is not to provide an extensive survey of the literature available on graph clustering, but rather to explain one method in some detail and discuss the computational issues that result. In particular, we wish to explain the recent application of techniques originally derived in statistical mechanics to find approximate solutions to these problems of a more statistical nature.

#### (a) *Mixed membership models*

A fundamental difference between ‘classical’ statistical clustering and our interest here is that we allow an object to be a member of more than one group. Such so-called mixed membership models have been developed extensively in recent years (Erosheva *et al.* 2004; Airoldi *et al.* 2005) and are particularly useful when the object of interest cannot be naturally expressed as a member of a single group. For example, a newspaper article may discuss several topics—characterizing the article as belonging to a single topic would then be inaccurate and potentially misleading. Mixed membership models are used in a variety of contexts and are distinguished also by the form of data available. Here, we assume

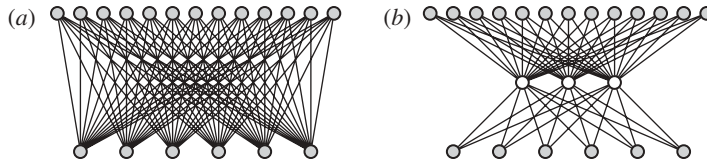


Figure 2. Graphical representation of dyadic data. (a) Here we have, say, six documents and 13 words. A link represents that the particular word–document pair occurs in the dataset. Here all links are shown; in practice, such graphs are typically very sparse. (b) A latent decomposition of (a) using three ‘topics’. A topic corresponds to a collection of words, and each document to a collection of topics. The open nodes indicate latent variables.

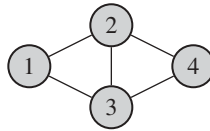


Figure 3. Canonical example used throughout the text. The minimal clique cover is (1, 2, 3), (2, 3, 4).

that all relevant information is contained in a single interaction matrix. Typically, these matrices represent two forms of interaction: dyadic and monadic, examples of which are discussed below.

### (i) *Dyadic data*

Consider a collection of documents, summarized by an interaction matrix  $A$ , in which the element  $A_{wd}$  represents the number of times the  $w$ th word in a dictionary occurs in document  $d$ . For simplicity, we consider the case in which  $A_{wd}$  is 1 if word  $w$  appears in document  $d$  and 0 otherwise. A graphical depiction of this matrix is a bipartite graph, as sketched in figure 2a. The lower nodes represent documents and the upper nodes words, with a link between them if that word occurs in that document. One might then seek assignments of words to groups or latent ‘topics’, so that one can accurately explain the link structure of the bipartite graph via a small number of latent nodes, as schematically depicted in figure 2b. One may view this as a form of binary matrix factorization (Hofmann *et al.* 1999; Meeds *et al.* 2008).

### (ii) *Monadic data*

In monadic data, there is only one type of object and the interaction between the objects is represented by a square interaction matrix. Here we also make the assumption that this interaction is binary. For example, a matrix with elements  $A_{ij} = 1$  expresses that proteins  $i$  and  $j$  can bind to each other, and  $A_{ij} = 0$  otherwise. Another example from document analysis is that  $A_{ij} = 1$  if documents  $i$  and  $j$  are considered ‘similar’ and 0 otherwise. A depiction of the interaction matrix is given by a graph in which a link represents an interaction (figure 3). Our interest is then to find a decomposition that explains this link structure in a

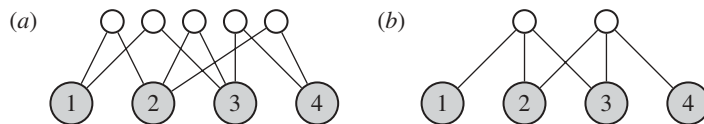


Figure 4. Bipartite representations of the decompositions of figure 3. Shaded nodes represent observed variables and open nodes latent variables. (a) Incidence and (b) minimal clique decomposition.

parsimonious way using latent variables. Graphically, this means that we seek a representation of the original graph, for example, figure 3, as a bipartite graph, say figure 4a.

In classical clustering, each object is assigned to only a single cluster. For example, a gene might be assigned to a single gene cluster, with all genes in a cluster having a similar microarray expression profile (Heyer *et al.* 1999). Graphically, this would restrict the degree of each shaded node (observed variable) in figure 4a to 1. In our mixed membership model, we impose no restriction on the degrees of the nodes. In this case, for example, a gene may be a member of several regulatory gene networks; a product might belong to many different kinds of product groupings, etc.

We shall focus on the monadic case, with the extension to the dyadic case being conceptually straightforward. The monadic case is of additional interest since it has natural links to parametrizing positive definite matrices, to which we turn in §4. Further details are contained in Barber (2008).

#### (b) Cliques and adjacency matrices for monadic data

A set of nodes that are all connected to each other is called a clique. For example, nodes (1, 2, 3) form a clique in figure 1a. A maximal clique cannot be contained within a larger clique. For example, (1, 2) in figure 1a is a non-maximal clique since it is part of a larger clique (1, 2, 3).

We can equivalently describe an undirected graph using the symmetric adjacency matrix  $A_{ij} \in \{0, 1\}$ , with a 1 indicating a link between nodes  $i$  and  $j$ . For the graph in figure 3, the adjacency matrix is

$$A = \begin{pmatrix} 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 \end{pmatrix}, \quad (1.1)$$

where we include self-connections on the diagonal. Given an adjacency matrix  $A$ , our aim is to find a ‘simpler’ description of  $A$  that reveals underlying cluster structure.

#### (i) Computational difficulty

A formal specification of the problem of finding a minimum number of maximal fully connected subsets is the computational problem MIN CLIQUE COVER (Garey & Johnson 1979; Skiena 1998). This is a computationally hard problem, and approximations are therefore generally unavoidable. The requirement that all nodes in a cluster be connected is somewhat strict. Provided only a small

number of links in an ‘almost clique’ are missing, this may be considered a sufficiently well-connected group of nodes to form a cluster. We therefore relax the hard constraints of MIN CLIQUE COVER and develop a statistical technique to reveal clusters of ‘well-connected’ nodes and additionally to identify the smallest number of such clusters. The resulting problem is still formally computationally intractable and requires the development of techniques to yield an efficient numerical approximation.

To phrase the clustering requirement more precisely, we will use a *clique matrix*, a generalization of the incidence matrix. This is useful for clustering and also plays a natural role in constrained covariance parametrization, as discussed in §4.

## 2. Clique decompositions

Given the undirected graph in figure 3, the incidence matrix  $F_{\text{inc}}$  is an alternative description of the adjacency structure; see, for example, Diestel (2005). We construct  $F_{\text{inc}}$  as follows: for each link  $i \sim j$  form a column of the matrix  $F_{\text{inc}}$  with entries 0 except for a 1 in the  $i$ th and  $j$ th rows. The column ordering is arbitrary. For the graph in figure 3, an incidence matrix is

$$F_{\text{inc}} = \begin{pmatrix} 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 \end{pmatrix}.$$

The incidence matrix has the interesting property that the adjacency structure of the original graph is related to  $F_{\text{inc}}F_{\text{inc}}^{\text{T}}$ . The diagonal entries contain the degree (number of links) of each node. For our example, this gives

$$F_{\text{inc}}F_{\text{inc}}^{\text{T}} = \begin{pmatrix} 2 & 1 & 1 & 0 \\ 1 & 3 & 1 & 1 \\ 1 & 1 & 3 & 1 \\ 0 & 1 & 1 & 2 \end{pmatrix},$$

so that the incidence and adjacency matrices are related via

$$A = H(F_{\text{inc}}F_{\text{inc}}^{\text{T}}). \quad (2.1)$$

Here  $H(\cdot)$  is the element-wise Heaviside step function, so that  $[H(M)]_{ij} = 1$  if  $M_{ij} > 0$  and 0 otherwise.

A useful viewpoint of the incidence matrix is that it identifies 2-cliques in the graph. There are five 2-cliques in figure 3, and each column of  $F_{\text{inc}}$  specifies which elements are in each 2-clique. Graphically, we can depict this incidence decomposition as a bipartite graph, as in figure 4a, in which the open nodes represent the five 2-cliques.

The incidence matrix can be generalized to describe larger cliques. Consider the following matrix as a decomposition for figure 3, and its outer product:

$$F = \begin{pmatrix} 1 & 0 \\ 1 & 1 \\ 1 & 1 \\ 0 & 1 \end{pmatrix} \quad \text{and} \quad FF^{\text{T}} = \begin{pmatrix} 1 & 1 & 1 & 0 \\ 1 & 2 & 2 & 1 \\ 1 & 2 & 2 & 1 \\ 0 & 1 & 1 & 1 \end{pmatrix}. \quad (2.2)$$

The interpretation is that  $F$  represents a decomposition into two 3-cliques. As for the incidence matrix, each column represents a clique and the rows containing a 1 express which elements are in the clique defined by that column. This decomposition can be represented as the bipartite graph of figure 4*b*. For the graph of figure 3, both  $F_{\text{inc}}$  and  $F$  satisfy

$$A = H(FF^T) = H(F_{\text{inc}}F_{\text{inc}}^T). \quad (2.3)$$

One can view equation (2.3) as a form of matrix factorization of the binary square (symmetric) matrix  $A$  into non-square binary matrices. For our clustering purposes, the decomposition using factor  $F$  is to be preferred to the incidence decomposition, since  $F$  decomposes the graph into a smaller number of larger cliques. Indeed,  $F$  solves MIN CLIQUE COVER for figure 1*b*.

#### (a) Clique matrices

More generally, given an adjacency matrix  $A_{ij}$ ,  $i, j = 1, \dots, V$  ( $A_{ii} = 1$ ), we define a clique matrix  $F$  to have elements  $F_{ic} \in \{0, 1\}$ ,  $i = 1, \dots, V$ ,  $c = 1, \dots, C$ , such that  $A = H(FF^T)$  (in contrast to most authors, we do not require the cliques to be maximal; see, for example, Golumbic & Ben-Arroyo Hartman (2005)).

The diagonal elements  $[FF^T]_{ii}$  express the number of cliques/columns in which node  $i$  occurs. Off-diagonal elements  $[FF^T]_{ij}$  contain the number of cliques/columns that nodes  $i$  and  $j$  jointly inhabit.

While finding a clique decomposition  $F$  is easy (use the incidence matrix, for example), finding a clique decomposition with the minimal number of columns, i.e. solving MIN CLIQUE COVER, is NP-hard (Garey & Johnson 1979; Arora & Lund 1997). One approach would be to use an iterative procedure that searches for local maximal cliques in the graph or related techniques based on finding large densely connected subgraphs (Skiena 1998). The alternative approach we consider is motivated by the idea that perfect clique decomposition is not necessarily practically desirable if the aim is only to find well-connected clusters in  $G$ .

#### (b) Statistical clique decompositions

To find ‘well-connected’ clusters, we relax the constraint that the decomposition is in the form of cliques in the original graph. Our approach is to view the absence of links as statistical fluctuations away from a perfect clique. Given a  $V \times C$  matrix  $F$ , we desire that the higher the overlap between rows<sup>1</sup>  $f_i$  and  $f_j$ , the greater the probability of a link between  $i$  and  $j$ . This may be achieved using, for example,

$$p(i \sim j | F) = \sigma(f_i f_j^T), \quad (2.4)$$

where  $\sim$  denotes that  $i$  and  $j$  are linked,

$$\sigma(x) \equiv (1 + e^{\beta(0.5-x)})^{-1}, \quad (2.5)$$

and  $\beta$  controls the steepness of the function (figure 5). The 0.5 shift in equation (2.5) ensures that  $\sigma$  approximates the step function since the argument of  $\sigma$  is an integer. Under equation (2.4), if  $f_i$  and  $f_j$  have at least one 1 in the

<sup>1</sup>The row vector  $f_i$  denotes the  $i$ th row of  $F$ .

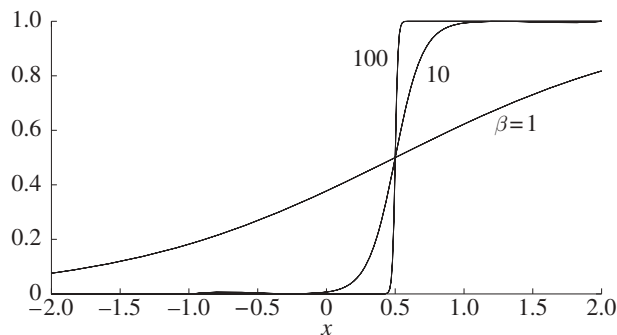


Figure 5. The function  $\sigma(x) \equiv (1 + e^{\beta(0.5-x)})^{-1}$  for  $\beta = 1, 10$  and  $100$ . As  $\beta$  increases, this sigmoid function tends to a step function.

same position,  $f_i f_j^T - 0.5 > 0$  and  $p(i \sim j | F)$  is high. Absent links contribute  $p(i \not\sim j | F) = 1 - p(i \sim j | F)$ . The parameter  $\beta$  controls how strictly  $\sigma(F F^T)$  matches  $A$ ; for large  $\beta$ , very little flexibility is allowed and only cliques will be identified. For small  $\beta$ , subsets that would otherwise be cliques, except for a small number of missing links, are clustered together. The setting of  $\beta$  is user and problem dependent.

Given  $F$ , and assuming each element of the adjacency matrix is sampled independently from the generating process, the joint probability of observing  $A$  is (if we neglect its diagonal elements)

$$p(A | F) = \prod_{i \sim j} \sigma(f_i f_j^T) \prod_{i \not\sim j} (1 - \sigma(f_i f_j^T)). \quad (2.6)$$

The ultimate quantity of interest is the posterior distribution of clique structure, given the known adjacency structure, which, according to Bayes' rule, is given by

$$p(F | A) \propto p(A | F) p(F), \quad (2.7)$$

where  $p(F)$  is a prior over clique matrices. The prior on  $F$  is used to encourage the smallest number of clusters to be identified (and hence for the size of the clusters to be large). Even in the case of a fixed desired number of clusters, determining the most likely clique matrix  $F$  is hard.

### 3. Clique decomposition using variational inference

Formally, our task is to find the most likely *a posteriori* (MAP) solution

$$\arg \max_F p(F | A)$$

corresponding to equation (2.7), where  $F$  is a  $V \times C$  binary matrix. Initially, we shall assume a 'flat prior'  $p(F) = \text{const}$ , so that the most likely solution

corresponds to finding the  $F$  that maximizes

$$p(A | F) = \prod_{i \sim j} \sigma(f_i f_j^T) \prod_{i \not\sim j} (1 - \sigma(f_i f_j^T)).$$

A variety of deterministic and randomized methods could be brought to bear on this problem. The approach we take here is to approximate the marginal posterior  $p(f_{ij} | A)$  and then to assign each  $f_{ij}$  to that state that maximizes this posterior marginal (MPM). This has the advantage of being closely related to marginal likelihood computations, which will prove useful later for addressing the issue of finding the number of clusters. The technique is analogous to naive mean-field theory, and details are given in appendix A, together with the corresponding iterative procedure for finding the approximate MPM solution.

### (a) Finding the number of clusters

To bias the contributions to the adjacency matrix  $A$  to occur from a small number of columns of  $F$ , we first reparametrize  $F$  as

$$F = (\alpha_1 f^1, \dots, \alpha_{C_{\max}} f^{C_{\max}}), \quad (3.1)$$

where  $\alpha_c \in \{0, 1\}$  plays the role of an indicator and  $f^c$  is the column vector of column  $c$  of  $F$ . The parameter  $C_{\max}$  is a stated maximal number of clusters we desire to find. Ideally, we would like to find a likely solution  $F$  with a low number of ‘active’ indicators  $\alpha_c$  in state 1. To achieve this, we define a prior distribution on the binary hypercube  $\alpha = (\alpha_1, \dots, \alpha_{C_{\max}})$ ,

$$p(\alpha | \nu) = \prod_c \nu^{\alpha_c} (1 - \nu)^{1 - \alpha_c}, \quad (3.2)$$

for  $\nu \in [0, 1]$ . To encourage a small number of the  $\alpha$  to be active, we use a beta prior  $p(\nu)$  with suitably chosen parameters. This gives a beta-Bernoulli distribution

$$p(\alpha) = \int p(\alpha | \nu) p(\nu) d\nu = \frac{B(a + N, b + C_{\max} - N)}{B(a, b)}, \quad (3.3)$$

where  $B(a, b)$  is the beta function and  $N \equiv \sum_{c=1}^{C_{\max}} \alpha_c$ , namely, the number of active indicators. To strongly encourage a small number of active components, we set  $a = 1$  and  $b = 3$ . The geometric picture is of a distribution on the vertices of the binary hypercube  $\{0, 1\}^{C_{\max}}$  with a bias towards vertices close to the origin  $(0, \dots, 0)$ . Through equation (3.1), the prior on  $\alpha$  induces a prior on  $F$ . The resulting distribution  $p(F, \alpha | A) \propto p(A | F) p(F | \alpha) p(\alpha)$  is formally intractable and needs to be dealt with in an approximate manner.

### (b) Variational Bayes

To deal with the intractable joint posterior, we adopt a strategy similar to the fixed  $C$  case and employ a variational procedure to seek a factorized approximation  $p(\alpha, F | A) \approx q(\alpha) q(F)$  based on minimizing

$$\text{KL}(q(\alpha) q(F) | p(\alpha, F | A)), \quad (3.4)$$

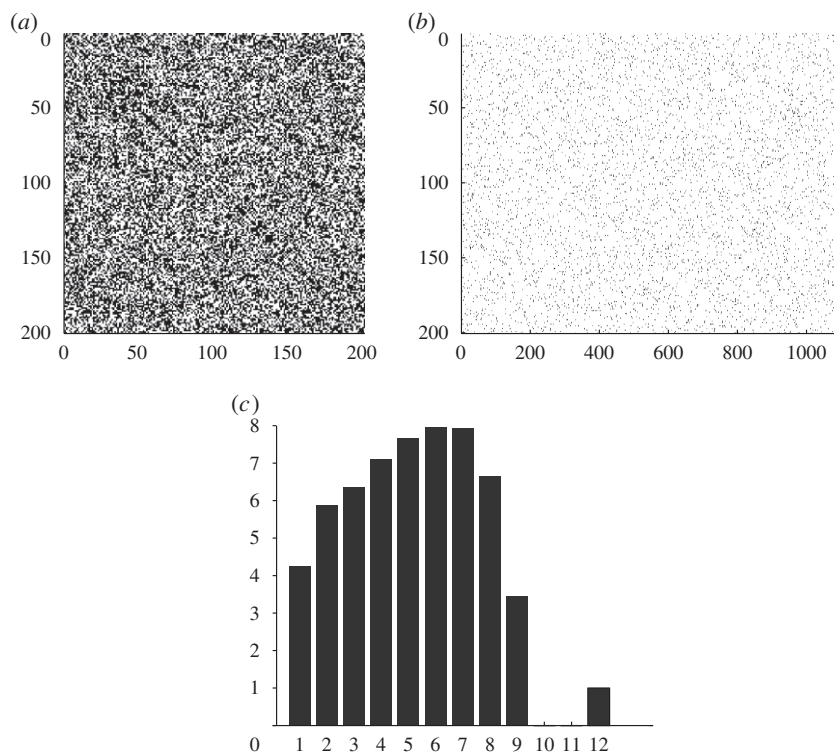


Figure 6. (a) Adjacency matrix for the DIMACS brock200-2 MAX CLIQUE challenge. Black denotes the presence of a link. (b) Clique matrix. (c) Log<sub>2</sub> histogram of clique occurrence (+1); correctly solves MAX CLIQUE (12) as well as identifying all remaining clusters.

where the Kullback–Leibler divergence is

$$\text{KL}(q, p) = \langle \log q \rangle_q - \langle \log p \rangle_q, \quad (3.5)$$

and  $\langle \cdot \rangle_q$  represents expectation with respect to  $q$ . This is analogous to a form of mean-field theory and results in a set of alternating mean-field updates for  $q(\alpha)$  and  $q(F)$ . The effect is that, on updating  $q(\alpha)$ , unnecessary clusters are pruned from consideration. The details are given in appendices B and C; code for identifying clique matrices is available from the author.

### (c) Demonstrations

#### (i) DIMACS MAX CLIQUE

In figure 6a we show the adjacency matrix for a 200-node graph, taken from the DIMACS 1996 MAX CLIQUE challenge (Brockington & Culberson 1996). This graph was constructed by the challenge coordinators to hide the largest clique in the graph and evade discovery based on the algorithms available at that time. While more recent algorithms have been constructed that readily find the largest clique in this graph (Pullan & Hoos 2006), this problem serves as an interesting baseline to see whether our algorithm, in searching for a complete

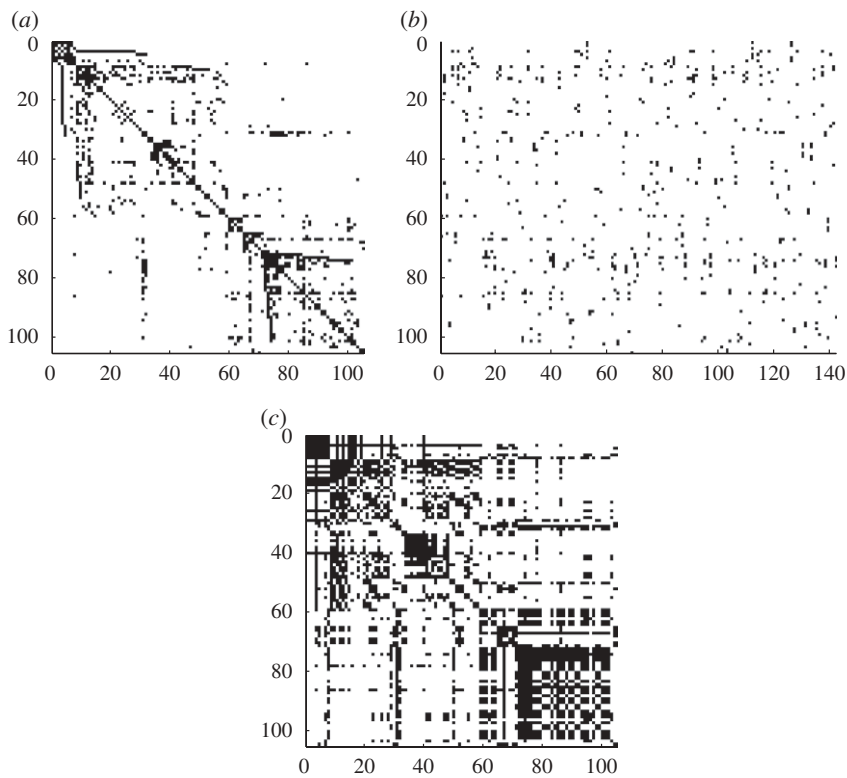


Figure 7. (a) Adjacency matrix of 105 political books (black = 1). (b) Clique matrix: 521 non-zero entries. (c) Adjacency reconstruction using an approximate clique matrix with 10 cliques (see also figure 8).

decomposition, also solves MAX CLIQUE for this graph. By setting  $\beta$  suitably high ( $\beta = 10$  in the experiments), we impose that clusters are formed from perfect cliques. Running our mean-field algorithm with  $C_{\max} = 2000$  results in a clique decomposition (figure 6b), containing 1102 cliques.<sup>2</sup> In figure 6c, we plot a log histogram of the cluster sizes for which there is only a single largest clique of size 12, in agreement with the exact result from Brockington & Culberson (1996).

### (ii) *Political books clustering*

The data consist of 105 books on US politics sold by the online bookseller Amazon. Links in graph  $G$  (figure 7a) represent frequent co-purchasing of books by the same buyers, as indicated by the ‘customers who bought this book also bought these other books’ feature on Amazon (V. Krebs, [www.orgnet.com](http://www.orgnet.com)). Additionally, books are labelled ‘liberal’, ‘neutral’ or ‘conservative’ according to the judgement of a politically astute reader ([www-personal.umich.edu/~mejn/netdata/](http://www-personal.umich.edu/~mejn/netdata/)). Of interest is to assign books to clusters using  $G$  alone, and then see if these clusters correspond in some way to the ascribed political leanings of each book. Note that the information here is minimal—all that

<sup>2</sup>This takes roughly 30 s using a 1 GHz machine.

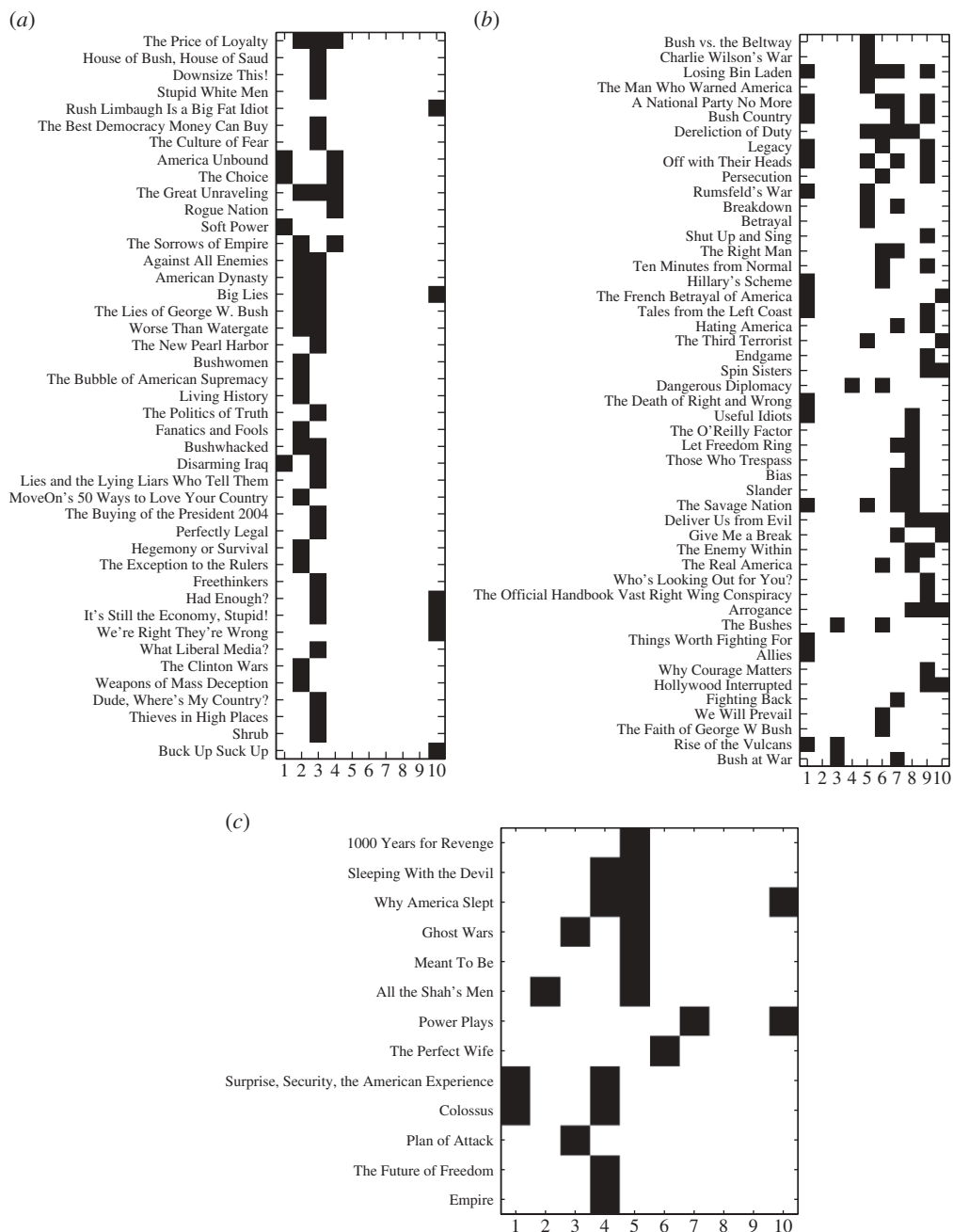


Figure 8. Political books. A  $105 \times 10$  dimensional clique matrix broken into three groups by a politically astute reader. A black square indicates  $q(f_{ic}) > 0.5$ . (a) Liberal books, (b) conservative books and (c) neutral books. By inspection, cliques 5, 6, 7, 8, 9 largely correspond to 'conservative' books.

is known to the clustering algorithm is which books were co-bought (matrix  $A$ ); no other information on the content or titles of the books are exploited by the algorithm.

If we run our algorithm with an initial  $C_{\max} = 200$  cliques,  $\beta = 10$ , the posterior contains 142 cliques,<sup>3</sup> figure 7*b*, giving a perfect reconstruction of the adjacency  $A$ . For comparison, the incidence matrix has 441 2-cliques.

To cluster the data more aggressively, we fix  $C = 10$  and run our fixed  $C$  algorithm. As expected, this results only in an approximate clique decomposition,  $A \approx H(FF^T)$ , as plotted in figure 7*c*. The resulting  $105 \times 105$  approximate clique matrix is plotted in figure 8 and demonstrates how individual books are present in more than one cluster. For visualization purposes, we plot the clique matrix in three parts, where each part corresponds to the political leaning of the book, according to an independent reader. Interestingly, the clusters found only on the basis of the adjacency matrix have some correspondence with the ascribed political leanings of each book, since one can see that cliques 5, 6, 7, 8 and 9 correspond to largely ‘conservative’ books. Most books belong to more than a single clique/cluster, suggesting that they are not single-topic books.

#### 4. Parametrizing constrained positive matrices

We turn now to what may at first seem an unrelated issue—parametrizing positive definite matrices. As is well known, any positive definite matrix  $K$  can be parametrized using a Cholesky factor,  $K = TT^T$ , where the Cholesky factor  $T$  is a lower-triangular real matrix.

Recently, interest is growing in relational machine learning in which constraints are imposed on the dependence between objects. In the simplest case of modelling the interaction between objects using a Gaussian distribution, this corresponds to imposing that specified elements of a covariance matrix (or in some cases its inverse) must be zero. We may use an undirected graph  $G$  to represent zero constraints on a positive definite matrix  $K$ . In particular, missing edges in  $G$  with corresponding adjacency matrix elements  $A_{ij} = 0$  correspond to zero entries  $K_{ij} = 0$ . We denote the space of positive definite matrices constrained through  $G$  by  $M^+(G)$ .

##### *Parametrizations using clique matrices*

One approach to parametrizing  $K$  based on given zero restrictions represented by  $A$  is to begin with a clique decomposition of  $A$ ,

$$A = H(FF^T).$$

By construction, the matrix  $FF^T$  is positive semi-definite and has zeros where  $A$  has zeros and integer values where  $A$  has ones. Hence, if we replace non-zero entries of a clique matrix  $F$  with arbitrary real values,  $F \rightarrow F^*$ , the matrix  $F^*(F^*)^T$  is also positive semi-definite and has the same zero structure as  $A$ . This, therefore, immediately gives a basic parametrization of the class of covariance matrices that have zeros specified according to  $G$ . A pertinent question is how rich is this parametrization—can all of  $M^+(G)$  be reached in this way?

<sup>3</sup>This take roughly 10s on a 1 GHz machine.

(a) *Decomposable case*

For  $G$  decomposable, parametrizing  $M^+(G)$  is straightforward (Wermuth 1980; Paulsen *et al.* 1989; Roverato 2002). Provided the vertices are perfect elimination ordered (when eliminated in the sequence, no additional links in the subgraphs are introduced—see appendix C), the Cholesky factor has the same structure as  $G$  (Wermuth 1980). In other words, provided the vertices are ordered correctly, the lower triangular part of the adjacency matrix is a clique matrix and further parametrizes all of  $M^+(G)$ . All positive definite matrices under decomposable zero constraints can therefore be parametrized by some clique matrix. Below we describe how a clique matrix can be derived that guarantees all of  $M^+(G)$  can be reached for decomposable  $G$ .

(i) *Expanded clique matrix*

Given a clique matrix  $F \in \{0, 1\}^{V \times C}$ , the expanded clique matrix consists of  $F$  appended with columns corresponding to all unique subcolumns of  $F$ . A subcolumn of  $f^c$  is defined by replacing one or more entries containing  $f_i^c = 1$  by  $f_i^c = 0$ .

Furthermore, a clique matrix  $F \in \{0, 1\}^{V \times C}$  is *minimal* for  $A$  if there exists no other clique matrix with a smaller number of columns  $C' < C$ . The expanded clique matrix corresponding to the minimal clique matrix derived from figure 1b is

$$\begin{pmatrix} 1 & 0 \\ 1 & 1 \\ 1 & 1 \\ 0 & 1 \end{pmatrix} \longrightarrow \begin{pmatrix} 1 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 1 & 1 & 0 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 1 \end{pmatrix}. \quad (4.1)$$

In the above, the expansion is ordered such that all 3-cliques are enumerated, then all 2-cliques and finally all 1-cliques.

Starting from a minimal clique matrix for a decomposable graph, the expansion of this minimal clique matrix must contain all the columns of the Cholesky factor  $T^T$ . For the example in figure 1b, the lower triangular Cholesky factor is

$$\begin{pmatrix} * & 0 & 0 & 0 \\ * & * & 0 & 0 \\ * & * & * & 0 \\ 0 & * & * & * \end{pmatrix},$$

which corresponds to columns 1, 2, 7, 11 of the expanded clique matrix, equation (4.1). In general, for a matrix with elements  $D_{ij} \in \{0, 1\}$ , we use  $D^*$  to denote a matrix with  $D_{ij}^* = 0$  if  $D_{ij} = 0$ , and arbitrary values elsewhere.

In a similar way, any expanded minimal clique matrix will always contain the columns of the Cholesky factor of a decomposable  $G$ . Clearly, for decomposable  $G$ , in general, the expanded clique matrix is an over-parametrization of  $M^+(G)$ .

(b) *Non-decomposable case*

For  $G$  non-decomposable, no explicit parametrization is generally possible, and techniques based on positive definite matrix completion are required (Speed & Kiiveri 1986; Paulsen *et al.* 1989; Roverato 2002; Chaudhuri *et al.* 2007). For the

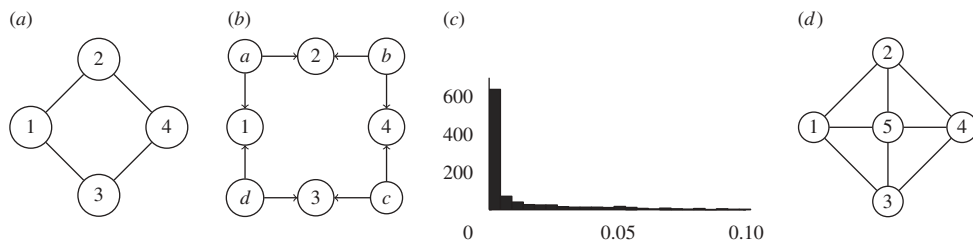


Figure 9. (a) Non-decomposable graph. (b) Correlations can be induced via latent variables. (c) Histogram of the r.m.s. errors in approximating covariances according to graph (a) with an expanded incidence matrix. (d) A non-decomposable graph for which a minimal clique covering contains three cliques, showing that the representation using a 3-clique matrix is richer than would be obtained using an incidence matrix (2-clique matrix).

specific example in figure 9a, the lower Cholesky factor has the form

$$\begin{pmatrix} t_{11} & 0 & 0 & 0 \\ t_{21} & t_{22} & 0 & 0 \\ t_{31} & t_{32} & t_{33} & 0 \\ 0 & t_{42} & t_{43} & t_{44} \end{pmatrix}, \quad \text{with } t_{21}t_{31} + t_{22}t_{32} = 0, \quad (4.2)$$

which can be found explicitly in this case. In general, however, one cannot explicitly identify those elements of the Cholesky factor that may be set to zero (Wermuth 1980; Roverato 2002).

An alternative is to use latent variables to explicitly parametrize  $M^+(G)$ . One may use factor analysis

$$x = F\epsilon, \quad \epsilon \sim \mathcal{N}(0, I) \quad \Rightarrow \quad \Sigma = FF^T,$$

where the factor matrix  $F$  is suitably structured in order to force zeros in specific elements of the covariance  $\Sigma$ .

A special case of the above is to use a latent variable to induce correlation between  $x_1$  and  $x_2$  via a local directed graph element  $x_1 \leftarrow \epsilon_{12} \rightarrow x_2$ . For each link in  $G$ , a corresponding latent  $\epsilon$  can thus be introduced to form correlations between each pair of variables, without introducing correlations on missing links in  $G$  (Dunsen *et al.* 2005). If we take  $F = [F_{\text{inc}}^* | I^*]$ , it is clear that this ‘ancillary variable’ approach (see Silva & Ghahramani 2006) is reproduced and is a special case of restricting cliques to (expanded) incidence matrices.

For the non-decomposable graph in figure 9d, the minimal clique matrix contains 3-cliques, so that its expansion contains columns that an expansion based on an incidence matrix would not. In this case, our approximate parametrization is therefore richer than would be obtained from introducing a latent auxiliary variable for each link of the graph (Dunsen *et al.* 2005; Silva *et al.* 2007).

To show that not all of  $M^+(G)$  can be reached by clique matrices, consider figure 9a. In this case, the minimal clique matrix is the same as the incidence matrix, and the expanded clique matrix is simply the incidence matrix with the identity matrix appended. In this case, therefore, the expanded

clique matrix contains columns with only two non-zero entries. However, the Cholesky factor, equation (4.2), contains columns with three non-zero entries, so that there is no immediate assignment of  $[F_{\text{inc}}^* | I^*]$  that will match the Cholesky factor.

(c) *Maximum-likelihood solution*

In fitting a Gaussian  $\mathcal{N}(0, \Sigma)$  to zero-mean data, with sample covariance  $S$ , the maximum-likelihood solution minimizes

$$\kappa(\Sigma) \equiv \text{Tr}(\Sigma^{-1}S) + \log \det \Sigma. \quad (4.3)$$

Our interest is to minimize  $\kappa(\Sigma)$  subject to zero constraints on  $\Sigma$  specified through  $G$ , with  $\Sigma_{ij} = 0$  if  $A_{ij} = 0$ , where  $A$  is the adjacency matrix of  $G$ . For decomposable  $G$ , the problem is essentially trivial, since  $M^+(G)$  is easily characterized via a structured Cholesky factor,  $\Sigma \equiv T^T(\theta)T(\theta)$  (see, for example, Roverato 2002), for which one can parametrize equation (4.3) using  $\kappa(\theta)$  and perform unconstrained minimization over the free parameters  $\theta$  of the Cholesky factor.

In the non-decomposable  $G$  case, no explicit parametrization of  $M^+(G)$  is feasible. A common approach in this case is to recognize that zero-gradient solutions to equation (4.3) satisfy  $[\Sigma^{-1}]_{ij} = [\Sigma^{-1}S\Sigma^{-1}]_{ij}$  for  $A_{ij} = 1$  and  $\Sigma_{ij} = 0$  otherwise (Anderson & Olkin 1985) and define iterative procedures to solve this equation (Drton & Richardson 2003). Alternatively, positive definite completion methods may be used to parametrize  $M^+(G)$ . Our approach uses the parametrization  $\Sigma = F^*(F^*)^T$ , where  $F$  should be chosen as large as can be computationally afforded. The non-expanded  $F$  can be determined by running the algorithm of appendix A. Although, for non-decomposable  $G$ , not all of  $M^+(G)$  is guaranteed reachable through this parametrization, one may expect that a large fraction of  $M^+(G)$  is within reach. A benefit of this approach is that one may then minimize equation (4.3) with respect to the free parameters of  $F^*$  using any standard optimization technique and convergence is guaranteed. Since our parametrization has a natural latent variable representation (it is a form of structured factor analysis), expectation maximization and Bayesian techniques can also be used in this case.

A numerical example is plotted in figure 9c where we take the  $4 \times 8$  expanded clique matrix corresponding to figure 9a and minimize equation (4.3) with respect to the non-zero entries of the clique matrix.<sup>4</sup> Each sample matrix  $S$  is generated randomly by drawing values of the Cholesky factor, equation (4.2), independently from a zero-mean unit-variance Gaussian. In figure 9c, we plot the root mean square error between the learned  $\Sigma$  and sample covariance  $S$ , averaged over all non-zero components of  $\Sigma$ . The histogram of the error computed from 1000 simulations shows that, while a few have appreciable error, the vast majority of cases are numerically well approximated by the expanded clique matrix technique, even though the graph  $G$  is non-decomposable.

<sup>4</sup>We chose this simple case since the exact parametrization of all  $M^+(G)$  is easy to write down. While here the expanded clique and incidence matrices are equivalent, the reader should bear in mind that, in more complex situations, the expansion based on a clique matrix provides a richer parametrization than that of the incidence matrix.

By placing Gaussian distributions over the non-zero elements of  $F$ , we naturally arise at a Wishart distribution constrained to satisfy specified constraints on the covariance. Such distributions then play a natural role in statistics as conjugate distributions in the Bayesian treatment of learning constrained covariances.

The clique matrix technique also naturally extends to modelling more general distributions under independence constraints. For example, if we set  $A_{ij} = 0$  when  $y_i$  and  $y_j$  are independent (and 1 otherwise), we first find a clique matrix  $F$  for  $A$  and define a Gaussian distribution  $p(x) = \mathcal{N}(0, FF^T)$ . Then, for any nonlinear transform  $u(x)$ , the non-Gaussian variables  $y_i \equiv u(x_i)$  inherit the independence relations specified in  $A$ . In this manner, one can fit non-Gaussian distributions to data under specified independence constraints.

I would like to thank Mark Herbster for useful discussions and Mike Titterton for improving the presentation of the manuscript.

### Appendix A. Mean-field approximation

Given the intractable  $p(F | A) \propto p(A | F)$ , a fully factorized approximation

$$q(F) = \prod_{i=1}^V \prod_{c=1}^C q(f_{ic}) \quad (\text{A } 1)$$

can be found by minimizing the KL divergence (e.g. [Wiegerinck 2000](#))

$$\text{KL}(q, p) = \langle \log q \rangle_q - \langle \log p \rangle_q, \quad (\text{A } 2)$$

where  $\langle \cdot \rangle_q$  represents expectation with respect to  $q$ . The first ‘entropic’ term simply decomposes into  $\sum_{i,c} \langle \log q(z_{i,c}) \rangle$ . The second ‘energy’ term, up to a constant, is

$$\sum_{i \sim j} \left\langle \log \sigma \left( \sum_c f_{ic} f_{jc} \right) \right\rangle_q + \sum_{i \not\sim j} \left\langle \log \left( 1 - \sigma \left( \sum_c f_{ic} f_{jc} \right) \right) \right\rangle_q. \quad (\text{A } 3)$$

The first term of equation (A 3) encourages graph links to be preserved under decomposition and is given by

$$\sum_{i \sim j} \left\langle \phi \left( \sum_{d=1}^C f_{id} f_{jd} \right) \right\rangle_{\prod_{c=1}^C q(f_{ic}) q(f_{jc})}, \quad (\text{A } 4)$$

where  $\phi(x) \equiv \log \sigma(x)$ . Minimizing equation (A 2) can be achieved by differentiation. Differentiating the energy contribution from the present links, equation (A 4), with respect to  $q(f_{kc})$ , we identify two cases: when  $i = k$  and when  $j = k$ . Owing to symmetry, the derivative is

$$2 \sum_{k \sim j} \left\langle \phi \left( \sum_d f_{kd} f_{jd} \right) \right\rangle_{\prod_c q(f_{jc}) \prod_{q \neq c} q(f_{kq})} \equiv \Psi(Q). \quad (\text{A } 5)$$

Similarly, the derivative of the absent-link energy is

$$2 \sum_{k \neq j} \left\langle \phi' \left( \sum_d f_{kd} f_{jd} \right) \right\rangle_{\prod_e q(f_{je}) \prod_{g \neq c} q(f_{kg})} \equiv \Psi'(Q), \quad (\text{A } 6)$$

where  $\phi'(x) \equiv \log(1 - \sigma(x))$ . If we equate the derivative of equation (A 2) to zero, a fixed-point condition for each  $q_{k,c}$ ,  $k = 1, \dots, V$ ,  $c = 1, \dots, C$ , is

$$q(f_{kc}) \propto e^{\Psi(Q) + \Psi'(Q)}. \quad (\text{A } 7)$$

Owing to the nonlinearities, neither  $\Psi(Q)$  nor  $\Psi'(Q)$  is easy to compute. A simple Gaussian field approximation (Barber & Sollich 2000) assumes  $\sum_d f_{kd} f_{jd}$  is Gaussian distributed for a fixed state of  $f_{ic}$ . In this case, we need to find the mean and variance of  $\sum_d f_{kd} f_{jd}$ . If we write  $\theta_{ab} \equiv q(f_{ab} = 1)$  and use the independence of  $q$ , the mean is given by

$$\mu_{kj} = f_{kc} \theta_{jc} + \sum_{d \neq c} \theta_{kd} \theta_{jd}.$$

A similar expression is easily obtained for the variance  $\sigma_{kj}^2$ . The Gaussian field approximation then becomes

$$q(f_{kc}) \propto \exp \left[ 2 \left\langle \sum_{j \sim k} \phi(x) + \sum_{j \not\sim k} \phi'(x) \right\rangle_{\mathcal{N}(x | \mu_{kj}, \sigma_{kj}^2)} \right], \quad (\text{A } 8)$$

where the one-dimensional averages are performed numerically. If we evaluate equation (A 8) for the two states of  $f_{kc}$  (and note that the mean and variance of the field depend on these states), the approximate update for  $\theta_{kc}$  is obtained. A simpler alternative is to assume that the variance of the field is zero and approximate the averages by evaluating the functions at the mean of the field. We found that this latter procedure often gives satisfactory performance and therefore used this simpler and faster approach in the experiments.

One epoch corresponds to updating all the  $\theta_{kc} = q(f_{kc} = 1)$ ,  $k = 1, \dots, V$ ,  $c = 1, \dots, C$ . The order in which the parameters are updated is chosen randomly.

## Appendix B. Variational Bayes

### (a) $q(F)$ updates

A fixed-point condition for the optimum of equation (3.4) is

$$q(F) \propto e^{(\log p(A|F, \alpha))_{q(\alpha)}} \approx e^{\log p(A|F, \langle \alpha \rangle)}. \quad (\text{B } 1)$$

The average over  $q(\alpha)$  in equation (B 1) in the first expression is complex to carry out and we simply approximate at the average value of the distribution. This yields a similar problem to that of inferring  $F$  for a fixed  $C$ , as in appendix A. We therefore make the same assumption that  $q(F)$  factorizes according to equation (A 1). This gives updates of the form in equation (A 8), where  $\alpha$  has been set to its mean value.

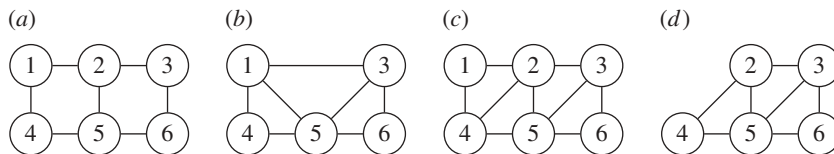


Figure 10. (a) Non-decomposable graph, (b) elimination of node for (a); (c) decomposable graph and (d) elimination of node for (c).

### (b) $q(\alpha)$ updates

A fixed-point condition for the optimum of equation (3.4) is

$$q(\alpha) \propto p(\alpha) e^{(\log p(A|F, \alpha))_{q(F)}}.$$

Additionally, we assume that  $q(\alpha) = \prod_c q(\alpha_c)$ . The resulting update

$$q(\alpha_c) \propto e^{(\log p(A|F, \alpha))_{q(F)} + (\log p(\alpha))_{\prod_{d \neq c} q(\alpha_d)}}$$

is difficult to compute, and we take the naive approach of replacing averages by evaluation at the mean

$$q(\alpha_c) \propto p(\alpha_c, \langle \alpha_{\setminus c} \rangle) p(A | \langle f \rangle, \alpha_c, \langle \alpha_{\setminus c} \rangle). \quad (\text{B } 2)$$

Since  $\alpha_c$  is binary, we can easily find equation (B 2) by evaluating it at its two states.

The algorithm then updates  $q(\alpha)$  and  $q(F)$  until convergence. The effect is that, beginning with  $C_{\max}$  clusters, under the updating, the posterior assigns the  $\alpha$  not required to state zero.

## Appendix C. Decomposable graph

We first define the graph operation of eliminating a variable (node). When a node  $i$  is eliminated, links are added between all the neighbours of node  $i$ . For example, if we eliminate node 2 in figure 10a, we remove node 2 from the graph and add links between the neighbours of node 2 (nodes 1, 3, 5), giving figure 10b. In this case, eliminating a node has introduced links between variables of the remaining subgraph. A decomposable graph is the one for which there exists a variable elimination sequence such that no additional links appear. For example, in figure 10c, if we first eliminate node 1, then we arrive at figure 10d. Subsequently, one can eliminate nodes 4, 2, 5 and 3 without inducing additional links in the remaining subgraphs. This means that figure 10c is a decomposable graph. Note that the elimination sequence is not unique.

## References

- Airoldi, E., Blei, D., Xing, E. & Fienberg, S. 2005 A latent mixed membership model for relational data. In *LinkKDD '05: Proc. 3rd Int. Workshop on Link Discovery*, pp. 82–89. New York, NY: ACM.
- Airoldi, E. M., Blei, D. M., Fienberg, S. E. & Xing, E. P. 2008 Mixed membership stochastic blockmodels. *J. Mach. Learn. Res.* **9**, 1981–2014.

- Anderson, T. W. & Olkin, I. 1985 Maximum-likelihood estimation of the parameters of a multivariate normal distribution. *Lin. Algebra Appl.* **70**, 147–171. (doi:10.1016/0024-3795(85)90049-7)
- Arora, S. & Lund, C. 1997 Hardness of approximations. In *Approximation algorithms for NP-hard problems*, pp. 399–446. Boston, MA: PWS Publishing.
- Barber, D. 2008 Clique matrices for statistical graph decomposition and parametrising restricted positive definite matrices. In *Proc. 24th Conf. on Uncertainty in Artificial Intelligence, 9–12 July 2008, Helsinki, Finland* (eds D. A. McAllester & P. Myllymäki), pp. 26–33. Corvallis, OR: AUAI Press.
- Barber, D. & Sollich, P. 2000 Gaussian fields for approximate inference in layered sigmoid belief networks. In *Advances in neural information processing systems*, vol. 12 (eds S. A. Solla, T. K. Leen & K.-R. Müller), pp. 393–399. Cambridge, MA: MIT Press.
- Brockington, M. & Culberson, J. 1996 Camouflaging independent sets in quasi-random graphs. In *Cliques, coloring, and satisfiability: second DIMACS implementation challenge* (ed. D. S. Johnson). DIMACS Series in Discrete Mathematics and Theoretical Computer Science, vol. 26. Providence, RI: American Mathematical Society.
- Chaudhuri, S., Drton, M. & Richardson, T. S. 2007 Estimation of a covariance matrix with zeros. *Biometrika* **94**, 199–216. (doi:10.1093/biomet/asm007)
- Diestel, R. 2005 *Graph theory*. Heidelberg, Germany: Springer.
- Drton, M. & Richardson, T. 2003 A new algorithm for maximum likelihood estimation in Gaussian graphical models for marginal independence. In *Proc. 19th Conf. on Uncertainty in Artificial Intelligence, 7–10 August 2003, Acapulco, Mexico* (eds C. Meek and U. Kjærulff), pp. 184–191. San Francisco, CA: Morgan Kaufmann.
- Dunson, D. B., Palomo, J. & Bollen, K. 2005 Bayesian structural equation modeling. SAMSI 2005-5, Statistical and Applied Mathematical Sciences Institute, Research Triangle Park, NC.
- Erosheva, E., Fienberg, S. & Lafferty, J. 2004 Mixed membership models of scientific publications. *Proc. Natl Acad. Sci. USA* **101**, 5220–5227. (doi:10.1073/pnas.0307760101)
- Garey, M. R. & Johnson, D. S. 1979 *Computers and intractability, a guide to the theory of NP-completeness*. New York, NY: W. H. Freeman.
- Goldberg, D., Nichols, D., Oki, B. M. & Terry, D. 1992 Using collaborative filtering to weave an information tapestry. *Commun. ACM* **35**(12), 61–70. (doi:10.1145/138859.138867)
- Golumbic, M. C. & Ben-Arroyo Hartman, I. 2005 *Graph theory, combinatorics, and algorithms*. New York, NY: Springer.
- Heyer, L. J., Kruglyak, S. & Yooseph, S. 1999 Exploring expression data: identification and analysis of coexpressed genes. *Genome Res.* **9**, 1106–1115. (doi:10.1101/gr.9.11.1106)
- Hofmann, T., Puzicha, J. & Jordan, M. I. 1999 Learning from dyadic data. In *Advances in neural information processing systems*, vol. 11 (eds M. Kearns, S. Solla & D. Cohn), pp. 466–472. Cambridge, MA: MIT Press.
- Karypis, G. & Kumar, V. 1998 A fast and high quality multilevel scheme for partitioning irregular graphs. *SIAM J. Sci. Comput.* **20**, 359–392. (doi:10.1137/S1064827595287997)
- Kautz, H., Selman, B. & Shah, M. 1997 ReferralWeb: combining social networks and collaborative filtering. *Commun. ACM* **40**(3), 63–65. (doi:10.1145/245108.245123)
- Meeds, E., Ghahramani, Z., Neal, R. & Roweis, S. 2008 Modelling dyadic data with binary latent factors. In *Advances in neural information processing systems*, vol. 19 (eds B. Schölkopf, J. C. Platt & T. Hoffman), pp. 977–984. Cambridge, MA: MIT Press.
- Newman, M. E. J. 2003 The structure and function of complex networks. *SIAM Rev.* **45**, 167–256. (doi:10.1137/S003614450342480)
- Paulsen, V. I., Power, S. C. & Smith, R. R. 1989 Schur products and matrix completions. *J. Funct. Anal.* **85**, 151–178. (doi:10.1016/0022-1236(89)90050-5)
- Pullan, W. J. & Hoos, H. H. 2006 Dynamic local search for the maximum clique problem. *J. Artif. Intell. Res.* **25**, 159–185.
- Roverato, A. 2002 Hyper inverse Wishart distribution for non-decomposable graphs and its application to Bayesian inference for Gaussian graphical models. *Scand. J. Stat.* **29**, 391–411. (doi:10.1111/1467.9469.00297)

- Silva, R. & Ghahramani, Z. 2006 Bayesian inference for Gaussian mixed graph models. In *Proc. 22nd Conf. on Uncertainty in Artificial Intelligence, 13–16 July 2006, Cambridge, MA*. Corvallis, OR: AUAI Press.
- Silva, R., Chu, W. & Ghahramani, Z. 2007 Hidden common cause relations in relational learning. In *Advances in neural information processing systems*, vol. 20 (eds J. C. Platt, D. Koller, Y. Singer & S. T. Roweis), pp. 1345–1352. Cambridge, MA: MIT Press.
- Skiena, S. S. 1998 *The algorithm design manual*. New York, NY: Springer.
- Speed, T. P. & Kiiveri, H. 1986 Gaussian Markov distributions over finite graphs. *Ann. Stat.* **14**, 138–150.
- Wermuth, N. 1980 Linear recursive equations, covariance selection, and path analysis. *J. Am. Stat. Assoc.* **75**, 963–972. (doi:10.2307/2287189)
- Wiegerinck, W. 2000 Variational approximations between mean field theory and the junction tree algorithm. In *Proc. 16th Conf. on Uncertainty in Artificial Intelligence, 30 June–3 July 2000, Stanford, CA* (eds C. Boutilier & M. Goldszmidt), pp. 626–633. San Francisco, CA: Morgan Kaufmann.