Marginal and simultaneous predictive classification using stratified graphical models

Henrik Nyman · Jie Xiong · Johan Pensar · Jukka Corander

Abstract An inductive probabilistic classification rule must generally obey the principles of Bayesian predictive inference, such that all observed and unobserved stochastic quantities are jointly modeled and the parameter uncertainty is fully acknowledged through the posterior predictive distribution. Several such rules have been recently considered and their asymptotic behavior has been characterized under the assumption that the observed features or variables used for building a classifier are conditionally independent given a simultaneous labeling of both the training samples and those from an unknown origin. Here we extend the theoretical results to predictive classifiers acknowledging feature dependencies either through graphical models or sparser alternatives defined as stratified graphical models. We show through experimentation with both synthetic and real data that the predictive classifiers encoding dependencies have the potential to substantially improve classification accuracy compared with both standard discriminative classifiers and the predictive classifiers based on solely conditionally independent features. In most of our experiments stratified graphical models show an advantage over ordinary graphical models.

Keywords Classification · Context-specific independence · Graphical model · Predictive inference

Mathematics Subject Classification 62-09 · 62H30 · 62F15

Electronic supplementary material The online version of this article (doi:10.1007/s11634-015-0199-5) contains supplementary material, which is available to authorized users.

H. Nyman (✉) · J. Pensar
Department of Mathematics and Statistics, Åbo Akademi University, Turku, Finland
E-mail: hennyman@abo.fi

J. Xiong · J. Corander
Department of Mathematics and Statistics, University of Helsinki, Helsinki, Finland
1 Introduction

This paper examines supervised classification using binary variables. However, the introduced theory can readily be expanded to also cover categorical discrete variables. Supervised classification is one of the most common tasks considered in machine learning and statistics (Ripley 1996; Duda et al. 2000; Hastie et al. 2009; Bishop 2007), with a wide variety of applications over practically all fields of science and engineering. Today, there exists a myriad of different classification methods, out of which those based on probabilistic models are widely accepted as the most sensible way to solve classification problems. Probabilistic methods are often themselves classified as either generative or discriminative, depending on whether one directly models the class posterior distribution (discriminative classifiers) or first the joint distribution of observed features (variables) conditional on class training data and then the posterior distribution of labels is obtained through Bayes’ rule. There has been a debate around which of these approaches should be preferred in a particular application, see Ripley (1996), Hastie et al. (2009), Bishop (2007), and Pernkopf and Bilmes (2005), however, both classes of methods continue to be supported and further developed. One of the popular methods of probabilistic classification is based on encoding feature dependencies with Bayesian networks (Friedman et al. 1997). Such models can often represent data structures more faithfully than the naive Bayes classifier, and have been shown to yield dramatic improvements in classification accuracy in some cases. Numerous variants and extensions of the original framework introduced by Friedman et al. (1997) have been considered over the years, e.g. Keogh and Pazzani (1999), Pernkopf and Bilmes (2005), Su and Zhang (2006), Cerquides and De Mántaras (2005), Madden (2009), and Holmes and Jain (2008). Friedman et al. (1997) concluded that general Bayesian networks did not perform better than the naive Bayes classifier, however, later Madden (2009) showed that this suboptimal behavior was attributable to the maximum likelihood estimation of the parameters used by Friedman et al. (1997) and when the parameter estimates were smoothed with a prior, the classification accuracy of the models was dramatically improved.

Albeit the above mentioned classifiers are occasionally called predictive, they are not predictive methods in the sense of Geisser (1964, 1966, 1993), who considered the foundations of general Bayesian predictive inference. Truly predictive generative classifiers need typically to model also the joint predictive distribution of the features, which leads to an infinite mixture over the parameter space when uncertainty about generating model parameters is characterized through their posterior distribution. In addition, as shown by Corander et al. (2013a, b, c), depending on the loss function employed for the classification task, genuinely inductive predictive classifiers may also require that all the data are predictively classified in a simultaneous fashion. This is in contrast with the standard classification methods which beyond the training data handle each sample independently and separately from others, which was termed marginal classification in Corander et al. (2013b). Simultaneous classifiers are therefore computationally much more demanding, because they necessitate modeling of the joint posterior-predictive distribution of the unknown sample labels.

It appears that the theory of simultaneous predictive classification is not widely known in the general statistical or machine learning literature. To the best of our
knowledge, none of the Bayesian network, or more generally graphical model based classifiers introduced earlier, are strictly Bayesian predictive in the meaning of Geisser (1993). However, in speech recognition the theoretical optimality of predictive simultaneous classifiers was notified already by Nádas (1985). Later work has demonstrated their value in several speech recognition applications, see, e.g. Huo and Lee (2000) and Maina and Walsh (2011). Also, Ripley (1988) discussed the enhanced performance of simultaneous classifiers for statistical image analysis, although not in the posterior predictive sense.

Corander et al. (2013a, b, c) considered the situation where features are assumed conditionally independent, given a joint simultaneous labeling of all the samples of unknown origin that are to be classified. Even if these samples were generated independently from the same underlying distributions, their labels are not in general independent in the posterior predictive distribution. Here we extend the inductive classifier learning to a situation where the feature dependencies are encoded either by ordinary graphical models, or by a recently introduced class of sparser stratified graphical models (Nyman et al. 2014). We show that the results of Corander et al. (2013b), concerning the asymptotic equality of simultaneous and marginal classifiers when the amount of training data tends to infinity, generalize to the situation with an arbitrary Markov network structure for the features in each class. Moreover, it is also shown that the asymptotic equality holds between graphical and stratified graphical models as well. For finite training data, we demonstrate that considerable differences in classification accuracy may arise between predictive classifiers built under the assumptions of empty graphs (predictive naive Bayes classifier), ordinary graphical models and stratified graphical models.

The remainder of this article is structured as follows. In Sect. 2 we give a short introduction to theory involving graphical models and stratified graphical models. Section 3 contains the theory needed to calculate the marginal likelihood of a dataset given a stratified graphical model. In Sect. 4 we use the marginal likelihood to define a novel class of marginal and simultaneous classifiers based on stratified graphical models, as well as marginal and simultaneous classifiers based on ordinary Markov networks. In Sect. 5 the introduced classifiers are compared to the predictive naive Bayes classifier as well as to a set of out-of-the-box classifiers using a range of synthetic and real datasets. Some general remarks and comments are given in the last section while proofs of theorems and certain technical details are provided in the Appendix and through online supplementary materials.

2 Stratified graphical models

In this section we give a short introduction to graphical models (GMs) and in particular stratified graphical models (SGMs). For a comprehensive account of the statistical and computational theory of these models, see Whittaker (1990), Lauritzen (1996), Koller and Friedman (2009), and Nyman et al. (2014).

Let \( G(\Delta, E) \), be an undirected graph, consisting of a set of nodes \( \Delta \) and of a set of undirected edges \( E \subseteq (\Delta \times \Delta) \). For a subset of nodes \( A \subseteq \Delta \), \( G_A = G(A, E_A) \) is a subgraph of \( G \), such that the nodes in \( G_A \) are equal to \( A \) and the edge set comprises those edges of the original graph for which both nodes are in \( A \), i.e. \( E_A = (A \times A) \cap E \).
Two nodes \( \gamma \) and \( \delta \) are adjacent in a graph if \( \{ \gamma, \delta \} \in E \), that is an edge exists between them. A path in a graph is a sequence of nodes such that for each successive pair within the sequence the nodes are adjacent. If \( A, B \) and \( D \) are three disjoint sets of nodes then \( A \) and \( B \) are said to be separated by \( D \) if every path between a node in \( A \) and a node in \( B \) contains at least one node in \( D \). A cycle is a path that starts and ends with the same node. A chord in a cycle is an edge between two non-consecutive nodes in the cycle. A graph is defined as chordal if there are no chordless cycles containing four or more unique nodes.

A graph is defined as complete when all pairs of nodes in the graph are adjacent. A clique in a graph is a set of nodes \( A \) such that the subgraph \( G_A \) is complete. A maximal clique is a clique \( C \) for which there exists no clique \( C' \) such that \( C \subset C' \). The set of maximal cliques in the graph \( G \) will be denoted by \( \mathcal{C}(G) \). The separators, \( \mathcal{S}(G) \), of a chordal graph \( G \) can be obtained through intersections of the maximal cliques of \( G \) ordered in terms of a junction tree, see e.g. Golumbic (2004). Any separator \( S \in \mathcal{S}(G) \) is a set of nodes such that \( S \) forms a clique. The separators of a chordal graph constitute a multiset as an element \( S \in \mathcal{S}(G) \) can appear more than once.

Associating each node \( \delta \in \Delta \) with a stochastic feature, or equivalently variable, \( X_\delta \), a GM is defined by the pair \( G = (\Delta, E) \) and a joint distribution \( P_\Delta \) over the variables \( X_\Delta \) satisfying a set of restrictions induced by \( G \). In the remainder of the text we use the terms feature and variable interchangeably. The outcome space for the variables \( X_A \), where \( A \subseteq \Delta \), is denoted by \( \mathcal{X}_A \) and an element in this space by \( x_A \in \mathcal{X}_A \). It is assumed throughout this paper that all considered variables are binary. However, the introduced theory can readily be extended to categorical discrete variables with larger than dichotomous outcome spaces. Given the graph of a GM, it is possible to ascertain if two sets of random variables \( X_A \) and \( X_B \) are marginally or conditionally independent. If there exists no path from a node in \( A \) to a node in \( B \) the two sets of variables are marginally independent, i.e. \( P(X_A, X_B) = P(X_A)P(X_B) \).

Similarly the variables \( X_A \) and \( X_B \) are conditionally independent given a third set of variables \( X_D \), \( P(X_A, X_B \mid X_D) = P(X_A \mid X_D)P(X_B \mid X_D) \), if \( D \) separates \( A \) and \( B \) in \( G \). In addition to marginal and conditional independencies, SGMs allow for the introduction of context-specific independencies. Using SGMs, two variables \( X_\delta \) and \( X_\gamma \) may be independent given that a specific set of variables \( X_A \) assume a certain outcome \( x_A \), i.e. \( P(X_\delta, X_\gamma \mid X_A = x_A) = P(X_\delta \mid X_A = x_A)P(X_\gamma \mid X_A = x_A) \). The set of outcomes for which such a context-specific independence holds is referred to as a stratum.

**Definition 1 (Stratum)** Let the pair \( (G, P_\Delta) \) be a GM. For all \( \{ \delta, \gamma \} \in E \), let \( L_{\{\delta, \gamma\}} \) denote the set of nodes adjacent to both \( \delta \) and \( \gamma \). For a non-empty \( L_{\{\delta, \gamma\}} \), define the stratum of the edge \( \{ \delta, \gamma \} \) as the subset \( \mathcal{L}_{\{\delta, \gamma\}} \) of outcomes \( x_{L_{\{\delta, \gamma\}}} \in \mathcal{X}_{L_{\{\delta, \gamma\}}} \) for which \( X_\delta \) and \( X_\gamma \) are independent given \( X_{L_{\{\delta, \gamma\}}} = x_{L_{\{\delta, \gamma\}}} \), i.e. \( \mathcal{L}_{\{\delta, \gamma\}} = \{ x_{L_{\{\delta, \gamma\}}} \in \mathcal{X}_{L_{\{\delta, \gamma\}}} : X_\delta \perp X_\gamma \mid X_{L_{\{\delta, \gamma\}}} = x_{L_{\{\delta, \gamma\}}} \} \).

The sign \( \perp \) is used throughout this paper to denote independence between variables.

A stratum can be represented graphically by adding conditions to an edge in a graph as shown in Fig. 1b. The graph in Fig. 1a induces both marginal and conditional independencies, for instance \( X_1 \perp X_5 \) and \( X_1 \perp X_4 \mid X_2, X_3 \). In addition, the
Fig. 1 In (a) a graphical model and in (b) a stratified graphical model

graph in Fig. 1b induces the context-specific independencies $X_1 \perp X_3 \mid X_2 = 1$, $X_2 \perp X_4 \mid X_3 = 0$, and $X_3 \perp X_4 \mid X_2 = 0$.

**Definition 2** (Stratified graphical model) An SGM is defined by the triple $(G, L, P_\Delta)$, where $G$ is the underlying graph, $L$ equals the joint collection of all strata $L_{[\delta, \gamma]}$ for the edges of $G$, and $P_\Delta$ is a joint distribution on $X_\Delta$ which factorizes according to the restrictions imposed by $G$ and $L$.

The pair $(G, L)$ consisting of the graph $G$ with the stratified edges (edges associated with a stratum) determined by $L$ will be referred to as a stratified graph (SG), usually denoted by $G_L$. When the collection of strata, $L$, is empty, $G_L$ equals $G$.

Given a chordal graph the marginal likelihood of a dataset can be calculated analytically. To extend this property to SGs, we introduce the concept decomposable SG. Consider an SG with a chordal underlying graph $G$ having the maximal cliques $C(G)$ and separators $S(G)$. The SG is defined as decomposable if no strata are assigned to edges in any separator and in every maximal clique all stratified edges have at least one node in common.

**Definition 3** (Decomposable SG) Let $(G, L)$ constitute an SG with $G$ being chordal. Further, let $E_L$ denote the set of all stratified edges, $E_C$ the set of all edges in the maximal clique $C$, and $E_S$ the set of all edges in the separators of $G$. The SG is defined as decomposable if

\[ E_L \cap E_S = \emptyset, \]

and

\[ E_L \cap E_C = \emptyset \quad \text{or} \quad \bigcap_{[\delta, \gamma] \in E_L \cap E_C} \{\delta, \gamma\} \neq \emptyset \quad \text{for all} \quad C \in C(G). \]

An SGM where $(G, L)$ constitutes a decomposable SG is termed a decomposable SGM. For example, the SG in Fig. 1b is a decomposable SG. The maximal clique $\{1, 2, 3\}$ contains one stratified edge $\{1, 3\}$, which means that in this clique all the stratified edges have nodes 1 and 3 in common. The maximal clique $\{2, 3, 4\}$ contains the stratified edges $\{2, 4\}$ and $\{3, 4\}$ with node 4 in common. Additionally, the only edge in a separator $\{2, 3\}$ is not a stratified edge.
3 Calculating the marginal likelihood of a dataset given an SGM

In this section we review the method developed in Nyman et al. (2014) to calculate the marginal likelihood of a dataset given an SGM. This method will be used in Sect. 4 to define marginal and simultaneous GM and SGM classifiers. Consider a chordal graph $G$ and the variable $X_\Delta$ associated with the nodes of $G$. Let $X$ denote a dataset consisting of $n$ realizations of these variables. If $A$ is a set of nodes $A \subseteq \Delta$, then $X_A$ is the subset of $X$ corresponding to the variables $X_A$. Under a prior distribution which enjoys the hyper-Markov property (Dawid and Lauritzen 1993), the marginal likelihood of $X$ given $G$ factorizes as

$$P(X \mid G) = \frac{\prod_{C \in \mathcal{C}(G)} P_C(X_C)}{\prod_{S \in \mathcal{S}(G)} P_S(X_S)},$$

(1)

where $\mathcal{C}(G)$ and $\mathcal{S}(G)$ are the maximal cliques and separators, respectively, of $G$. For any subset $A \subseteq \Delta$ of nodes, $P_A(X_A)$ denotes the marginal likelihood of the subset $X_A$ of data. Nyman et al. (2014) showed that this factorization can also be applied for decomposable SGs.

Nyman et al. (2014) derived a formula for calculating $P_C(X_C)$ and $P_S(X_S)$, which is applicable to both ordinary graphs and SGs. Their derivation is based on introducing a specific ordering of the clique variables and merging some conditional distributions. Consider a maximal clique in a decomposable SG consisting of $d$ nodes. It follows from the definition of decomposable SGs that all stratified edges in this clique have at least one node in common. We introduce an ordering of the variables, corresponding to the nodes of such a clique, such that the last variable in the ordering corresponds to the node found in all stratified edges. This variable is denoted by $X_d$. All the changes to the dependence structure caused by the inclusion of strata can be seen in the conditional distribution $P(X_d \mid X_1, \ldots, X_{d-1})$. In the absence of strata, each outcome of the variables $(X_1, \ldots, X_{d-1})$, termed as parents of $X_d$ and denoted by $\Pi_d$, would induce a unique conditional distribution. However, each outcome in a stratum on an edge serves to merge a subset of these outcomes. For instance, if all variables are considered to be binary, the condition $(X_1 = 0, \ldots, X_{d-2} = 0)$ on the edge $\{d-1, d\}$ will merge the outcomes $(X_1 = 0, \ldots, X_{d-2} = 0, X_{d-1} = 0)$ and $(X_1 = 0, \ldots, X_{d-2} = 0, X_{d-1} = 1)$.

The merging process is best illustrated using conditional probability tables (CPTs). As an example, consider the maximal clique $\{2, 3, 4\}$ in the ordinary graph and SG in Fig. 1. Variable $X_4$ has to be chosen as the last one in the ordering. Table 1 contains the CPTs for $X_4$ for both the ordinary graph and the SG. The condition $X_2 = 0$ on the edge $\{3, 4\}$ merges outcomes (1) and (2), while the condition $X_3 = 0$ on the edge $\{2, 4\}$ merges outcomes (1) and (3). This results in $X_4$ having only two unique conditional distributions, one for the group of outcomes $(X_2 = 0, X_3 = 0), (X_2 = 0, X_3 = 1), (X_2 = 1, X_3 = 0)$ and one for the outcome $(X_2 = 1, X_3 = 1)$.

Combining this merging process with the formula introduced by Cooper and Herskovits (1992), used for calculating the marginal likelihood of a Bayesian network, leads to the following formula for the marginal likelihood of a maximal clique
Table 1: Corresponding CPTs for X4 in the graph and SG in Fig. 1

| Outcome | X2 | X3 | P(X4 | X2, X3) in GM | P(X4 | X2, X3) in SGM |
|---------|----|----|-------------------|-------------------|
| (1)     | 0  | 0  | p1                | q1                |
| (2)     | 0  | 1  | p2                | q1                |
| (3)     | 1  | 0  | p3                | q1                |
| (4)     | 1  | 1  | p4                | q2                |

\[ P_C(X_C) = \prod_{j=1}^{d} \prod_{l=1}^{q_j} \frac{\Gamma \left( \sum_{i=1}^{k_j} \alpha_{jil} \right)}{\Gamma \left( n(\pi^l_j) + \sum_{i=1}^{k_j} \alpha_{jil} \right)} \frac{k_j}{\Gamma(\alpha_{jil})}, \]  

where \( \Gamma \) denotes the gamma function, \( d \) equals the number of variables in the clique \( C \), \( q_j \) is the number of distinguishable parent combinations for variable \( X_j \) (i.e. there are \( q_j \) distinct conditional distributions for variable \( X_j \)), \( k_j \) is the number of possible outcomes for variable \( X_j \), \( \alpha_{jil} \) is the hyperparameter used in a Dirichlet prior distribution corresponding to the outcome \( i \) of variable \( X_j \) given that the outcome of the parents of \( X_j \) belongs to group \( l \), \( n(\pi^l_j) \) is the number of observations of the combination \( l \) for parents of variable \( X_j \), and finally, \( n(x^i_j | \pi^l_j) \) is the number of observations where the outcome of variable \( X_j \) is \( i \) given that the observed outcome of the parents of \( X_j \) belongs to \( l \). Note that in this context a parent configuration \( l \) is not necessarily comprised of a single outcome of the parents of variable \( X_j \), but rather a group of outcomes with an equivalent effect on \( X_j \).

The hyperparameters of the Dirichlet prior distribution can be chosen relatively freely, here we use (Nyman et al. 2014)

\[ \alpha_{jil} = \frac{N \cdot \lambda_{jl}}{\pi_j \cdot k_j}, \]

where \( N \) is the equivalent sample size, \( \pi_j \) is the total number of possible outcomes for the parents of variable \( X_j \) (= 1 for \( X_1 \)) and \( k_j \) is the number of possible outcomes for variable \( X_j \). Further, \( \lambda_{jl} \) equals the number of outcomes for the parents of variable \( X_j \) in group \( l \) with an equivalent effect on \( X_j \), if \( X_j \) is the last variable in the ordering. Otherwise, \( \lambda_{jl} \) equals one. Equation (2) can also be used to calculate \( P_C(X_C) \) when \( C \) is a non-maximal clique containing no stratified edges. As a result, \( P_S(X_S) \) can be calculated using (2) as any separator \( S \) constitutes a clique containing no stratified edges. Obviously, \( P_C(X_C) \) and \( P_S(X_S) \) for ordinary GMs can therefore also be calculated using (2). For these instances each group \( l \) consists of a single outcome of the parents of variable \( X_j \).

4 Predictive GM and SGM Classifiers

GMs and SGMs are now employed to define novel types of predictive classifiers. Although GMs have previously been used in classification frameworks, to our knowledge they have not been considered in a true predictive sense as described by Geisser.
Determining the dependence structure among the features basically corresponds to determining the significant parameters in the generating distribution. Being able to correctly assess the dependence structure is important as any under- or over-parameterization will have an adverse effect on the predictive classifier.

We assume that the dependence structure can freely vary across different classes. Let $X^R$, consisting of $m$ observations on $|\Delta|$ features, constitute the training data over $K$ classes. The class labels for the observations in $X^R$ are specified by the vector $R$, where the element $R(i) \in \{1, \ldots, K\}$ defines the class of observation $i$, with $i = 1, \ldots, m$. Similarly, $X^T$ represents the test data consisting of $n$ observations, and their classification is determined by the vector $T$, which is the main target of predictive inference. Using the training data for class $k$, $X^R_k$, a search for the SGM (GM) with optimal marginal likelihood, $G^L_k$, can be conducted (Nyman et al. 2014). Given the resulting SGs (ordinary graphs) for each class, $G^A_k$, posterior predictive versions of the Eqs. (1) and (2) can be used to probabilistically score any candidate classification of the test data.

We consider two types of predictive classifiers, a marginal classifier and a simultaneous classifier (Corander et al. 2013b). Both assign a predictive score to the label vector $T$, which can be normalized into a posterior probability given a prior distribution over possible labelings. Corander et al. (2013b) introduce formally various classification rules using the posterior distribution and decision theory, however, here we simply consider the maximum a posteriori (MAP) rule under a uniform prior, which corresponds to maximization of the predictive score function. For the marginal classifier the predictive score, $S_{\text{mar}}$, is defined as

$$S_{\text{mar}}(T \mid X^T, X^R, R, G^A_L) = \prod_{i=1}^{n} P(X^T_i \mid T, X^R, R, G^A_L) = \prod_{i=1}^{n} \frac{\prod_{C \in C(G^T_{L(i)})} P_C(X^T_{i,C} \mid X^{R,T(i)})}{\prod_{S \in S(G^T_{L(i)})} P_S(X^T_{i,S} \mid X^{R,T(i)})},$$

where $X^T_i$ denotes the $i$th observation in the test data, and $X^T_{i,C}$ denotes the outcomes of the variables associated to the maximal clique $C$ in this observation. The posterior predictive likelihoods $P_C(X^T_{i,C} \mid X^{R,T(i)})$ and $P_S(X^T_{i,S} \mid X^{R,T(i)})$ are calculated using (2) and the updated hyperparameters $\beta_{jil}$ instead of $\alpha_{jil}$,

$$\beta_{jil} = \alpha_{jil} + m(x^i_j \mid \pi^l_j),$$

where $m(x^i_j \mid \pi^l_j)$ is the number of observations in $X^{R,T(i)}$ where the outcome of variable $X_j$ is $i$ given that the observed outcome of the parents of $X_j$ belongs to group $l$. The updated hyperparameters are derived as a standard result when using the Dirichlet distribution which is a conjugate prior of the multinomial distribution.

The optimal classification decision is obtained by the vector $T$ that maximizes the score function over all $n$ samples in the test data, i.e.

$$\arg \max_T S_{\text{mar}}(T \mid X^T, X^R, R, G^A_L).$$
Using the simultaneous predictive classifier the observations in the test data are not classified independently of each other as is the case with the marginal classifier presented above. Instead, the score function becomes

\[
S_{\text{sim}}(T \mid X^T, X^R, R, G^A_L) = \prod_{k=1}^{K} P(X_T^{T,k} \mid T, X^R, R, G^A_L)
= \prod_{k=1}^{K} \prod_{C \in C(G^k_L)} P_C(X_C^{T,k} \mid X^R,k) \prod_{S \in S(G^k_L)} P_S(X_S^{T,k} \mid X^R,k),
\]

where \(X_T^{T,k}\) denotes observation in the test data assigned to class \(k\) by \(T\), and \(X_C^{T,k}\) denotes the outcomes of the variables associated to the maximal clique \(C\) for these observations. The posterior predictive likelihoods \(P_C(X_C^{T,k} \mid X^R,k)\) and \(P_S(X_S^{T,k} \mid X^R,k)\) are again calculated using (2) and the updated hyperparameters \(\beta_{jil}\). The optimal labeling is obviously still determined by the vector \(T\) that optimizes the simultaneous score function.

Intuitively, the simultaneous classifier merges information from the test data already assigned to class \(k\) with the training data of class \(k\), when assessing the joint probability of observing them all. Depending on the level of complexity of the model and the size of the training and test datasets, this increases the accuracy of the classifier, as shown in Corander et al. (2013b). However, the theorems below formally establish that when the size of the training data grows, the classification decisions based on marginal and simultaneous predictive SGM classifiers become equivalent, and further that also GM and SGM based predictive classifiers become equivalent.

**Theorem 1** The marginal and simultaneous predictive SGM classifiers are asymptotically equivalent as the size of the training data goes to infinity.

*Proof* (Theorem 1) See Appendix A. \(\square\)

**Theorem 2** The predictive SGM and GM classifiers are asymptotically equivalent as the size of the training data goes to infinity.

*Proof* (Theorem 2) See Appendix B. \(\square\)

The vector \(T\) that optimizes the predictive score is identified using the same methods as in Corander et al. (2013b, c). For the marginal classifier we cycle through all observations in the test data assigning each one to the class optimizing the predictive score \(S_{\text{mar}}\). For the simultaneous classifier we begin by initializing a start value for \(T\), this vector may be generated randomly or, for instance, chosen as the vector that maximizes the score for the marginal classifier. The elements in \(T\) are then changed such that \(S_{\text{sim}}\) is successively optimized for each element. This procedure is terminated once an entire cycle, where each element in \(T\) is considered once, is completed without evoking any changes in \(T\).

In the next section we will demonstrate how the marginal and simultaneous GM and SGM classifiers compare to each other. They will also be compared to the predictive naive Bayes classifier, which is implemented in the same way as the GM classifier under the assumption that the graph structure contains no edges, as well as to some out-of-the-box classifiers.
5 Numerical experiments

In this section the novel predictive GM and SGM classifiers introduced in the previous section are compared to the predictive naive Bayes classifier as well as some out-of-the-box classifiers available in Matlab’s statistics toolbox. The out-of-the-box classifiers are: the \( k \)-nearest neighbor classifier (kNN) using \( k = 5 \), the classification and regression tree classifier (CART), and an improved version of CART that uses bootstrap aggregation (CART-Bag).

The synthetic data used in the following examples is generated from five different classes. The variables in each class are associated with a unique dependence structure. In each class a group of five variables constitutes a chain component, variables in different chain components are independent of each other. For a given class, the variables in each of the chain components follow the same dependence structure and distribution. This framework makes it easy to construct datasets with a larger number of variables by combining any desired amount of chain components. The dependence structure for the variables for each of the five classes follows that of the SGs in Fig. 2. Note that instead of writing a condition as \((X_1 = 1, X_2 = 0)\), in order to save space it is sufficient to write \((1, 0)\). This is possible since, given the graph, it is clear which variables that are included in the condition and the variables are ordered by their index numbers. Also a condition where either \(X_\delta = 0\) or \(X_\delta = 1\) holds is written as \(X_\delta = \ast\).

The probability distributions used for each class is available in Online Resource 1. Training data and test data are generated from the five different classes. In the first experiment the number of features is set equal to 20 and we fix the number of observations per class in the test data to 20, while letting the number of observations per class in the training data vary from 10 to 200. Here we make the simplifying assumption that the dependence structure, as encoded by GMs and SGMs, is known for each class.

Fig. 2  Dependence structure for the variables in the five different classes
The classifiers were then applied to 200 similarly generated training and test datasets with the average resulting success rates for each classifier displayed in Fig. 3.

It is clearly evident that the predictive classifiers introduced in this paper outperform the out-of-the-box classifiers. There is also a difference between the GM and SGM based classifiers in favor of the latter, ranging from about 2 percentage points for 20 observations in the training data to about 0.7 percentage points for 200 observations. However, the differences between the marginal and simultaneous classifiers are negligible (curves practically overlap) in both the GM and SGM cases. This can be explained by two main reasons. Firstly, the size of the test data is small compared to the training data, meaning that the extra knowledge gained from the test data in the simultaneous case is relatively small. Secondly, the fraction of correctly classified observations is quite low due to the small number of features, meaning that the test data may, contrary to intentions, have a negative effect on the predictive inference where class-conditional parameters are integrated out.

In the second experiment (Fig. 4) we increase the number of features to 50. As one could easily predict the success rates of the classifiers are improved by increasing the number of features. We can also observe that in cases where the size of the training data is less than 50, the simultaneous SGM and GM classifiers outperform their marginal counterparts. When the size of the training data is as small as 10 the difference between the simultaneous and marginal SGM classifiers is roughly 5 percentage points.

In order to further demonstrate the advantage with the simultaneous classifiers we fix the number of training data observations per class to 20 while letting the number of test data observations per class range from 10 to 100. Figure 5 shows that the difference between the simultaneous and marginal classifiers can be significant. When the number of observations in the test data is 100 the fraction of correctly classified observations is 73.9 % for the simultaneous SGM classifier and 64.9 % for the marginal SGM classifier. The corresponding fractions for the GM classifiers are 71.6 and 61.6 %, respectively. This also clearly shows that when the dependence structure is known the SGM classifiers outperform the GM classifiers.

In the next experiment, to avoid excessive computational cost, we focus on the faster marginal classifiers with the added difficulty that we now also need to learn the dependence structure (GM and SGM) from the training data for each of the 1,000
replicated datasets per training data size. The learning process is completed by first inferring the optimal GM followed by inferring the optimal set of strata to obtain the SGM. In order to keep the required simulation time tractable, we again limit the number of features to 20. As we only use the marginal classifiers in this case the size of the test data is irrelevant regarding the computational complexity and resulting success rates. The number of observations per class in the training data is set to vary from 50 to 900. The results from this experiment are displayed in Fig. 6. The marginal classifiers using the known GM and SGM structures have also been included as a baseline reference. We can immediately see that the classifiers where the structure is learned using the training data have lower success rates than when the structure is known, which is expected. However, we nevertheless see that the SGM classifiers perform better than GM classifiers, irrespectively of whether the dependence structure is learned from the training data or is assumed to be known beforehand. The results also visualize the asymptotic behavior established in Theorems 1 and 2, as the success rates tend to become identical for all the GM and SGM based classifiers as the size of the training data grows.

To ascertain how the SGM classifiers perform in the absence of context-specific independencies we generate data from a set of probability distributions, available in Online Resource 1, whose dependence structures are defined by the graphs that result
Predictive classification using stratified graphical models

Fig. 6  Different success rates of classifier methods using 20 features. The number of training data observations per class ranges from 50 to 900, the number of test data observations per class is fixed to 20.

Fig. 7  Different success rates of classifier methods using 20 features whose dependence structure can be represented by a GM. The number of training data observations per class ranges from 50 to 900, the number of test data observations per class is fixed to 20.

from removing the strata from the SGs in Fig. 2. The above experiment is repeated using these new distributions resulting in the success rates found in Fig. 7.

As the set of all SGMs is a superset of all GMs the marginal classifiers using the known GM and known SGM structure will be identical. The results show that in this case the SGM and GM marginal classifiers perform almost identically, with the difference in success rates ranging from 0.6 percentage points in favor of the SGM classifier for 50 observations in the training data to 0.1 percentage points in favor of the GM classifier for 900 observations in the training data. A visible trend both in Figs. 6 and 7 is that in case the number of observations in the training data is less than 200 per class the “CART-Bag” classifier produces results superior to the SGM and GM classifiers where the dependence structure is learned from the data. However, as the size of data increases the SGM and GM classifiers perform better. This is due to the fact that in some cases a large amount of data is required to correctly infer the dependence structure.

Two real datasets are now used to compare performances of the different predictive classifiers. First, the marginal classifiers (naive Bayes, GM, and SGM) are applied to a dataset representing the genetic variation in the bacterial genus Vibrionaceae (Dawyndt et al. 2005). The dataset consists of 507 observations on 994 binary fAFLP
markers. Using 10 estimation runs of the basic clustering module in BAPS software (Corander and Marttinen 2006; Corander et al. 2008) with a priori upper bound of the number of clusters ranging between 20–150, the data is split into a total of 80 classes. The ten classes containing most data, between 12–44 observations, are further used in a classification experiment.

Because of the large number of features in the Vibrionaceae data, learning the dependence structure poses a considerable computational challenge. Therefore, to simplify the computations, the features are separated into ten groups, with nine groups of 100 and one group of 94 features. When learning the undirected graph representing the dependence structure in each group and class we also restricted the maximum clique size to five nodes. This restriction has only little effect in practice since the identified structures mostly include small maximal cliques. Once the optimal undirected graphs are obtained, a search for the optimal strata is conducted with the given undirected graph used as the underlying graph. With all these restrictions the resulting GMs and SGMs can by no means be considered to be the corresponding global optima, however, they serve well enough to illustrate the classifier performance.

To assess the accuracy of the different classifiers, the data is randomly split into two sets, the test data containing two observations from each class and the training data comprising the remaining observations. As the test data is relatively small compared to the training data it is reasonable to assume that the marginal and simultaneous classifiers would perform almost identically and therefore only the marginal classifiers are considered. First, all the features were utilized in the classification, resulting in a success rate of 100 %. While this is an encouraging result it is useless for comparing the different classification methods. Therefore, the classification was then performed separately for each of the ten groups of features, which reduces the number of features to 1/10 of the original dataset. The resulting average success rates over multiple simulations are displayed in Table 2.

While it is clear that the GM and SGM classifiers have quite similar success rates, the SGM classifier consistently performs better. It is worthwhile to note that the search for the optimal SGM structure is more complicated than that for the GM structure.

| Group | SGM  | GM   | Naive Bayes |
|-------|------|------|-------------|
| 1     | 0.9911 | 0.9870 | 0.9673 |
| 2     | 1.0000 | 0.9988 | 0.9208 |
| 3     | 0.9982 | 0.9983 | 0.9094 |
| 4     | 1.0000 | 1.0000 | 0.9379 |
| 5     | 0.9964 | 0.9949 | 0.9134 |
| 6     | 0.9908 | 0.9876 | 0.9039 |
| 7     | 0.9901 | 0.9884 | 0.9105 |
| 8     | 0.9975 | 0.9963 | 0.8624 |
| 9     | 0.9228 | 0.9169 | 0.8671 |
| 10    | 0.6581 | 0.6414 | 0.5256 |
| All   | 0.9545 | 0.9510 | 0.8718 |

The bold values highlight which classifier performs optimally for each group.
The second real dataset that we consider is derived from the answers given by 1,444 candidates in the Finnish parliament elections of 2011 in a questionnaire issued by the newspaper *Helsingin Sanomat* (2011). The dataset, which contains 25 binary features, is available as Online Resource 2. The candidates belong to one of the eight political parties, listed in Appendix C, whose members were subsequently elected to the parliament.

The data is divided into test data containing 100 observations and training data containing 1,344 observations. The dependence structure, in the form of an ordinary graph and an SG, is inferred from the training data for each class (political party). Different classifiers are then applied to classify the observations in the test data. This experiment is repeated several times and the average fraction of correctly classified observations for each class and classifier is listed in Table 3.

As the test data is relatively small compared to the training data the observed differences between the marginal and simultaneous classifiers are very small, roughly 0.1 percentage points. Somewhat contradictory to earlier results, for this dataset the GM classifier performs better than the SGM classifier. However, the differences between these classifiers are again very small, on average 0.2 percentage points over all the classes. The difference compared to the out-of-the-box classifiers are more significant, with for instance the simultaneous SGM classifier outperforming the boosted CART-Bag classifier by 4.0 percentage points.

### Table 3 Resulting success rates per class for parliament election data

| Classifier | Party | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | Tot. |
|------------|-------|---|---|---|---|---|---|---|---|-----|
| Sim. SGM   |       | 0.722 | 0.913 | 0.641 | 0.579 | **0.651** | 0.573 | 0.656 | 0.814 | 0.700 |
| Marg. SGM  |       | 0.720 | 0.912 | 0.636 | 0.580 | 0.647 | 0.572 | 0.657 | 0.816 | 0.699 |
| Sim. GM    |       | 0.718 | 0.913 | **0.642** | 0.591 | 0.646 | **0.581** | **0.660** | 0.816 | **0.702** |
| Marg. GM   |       | 0.716 | 0.911 | 0.637 | 0.589 | 0.645 | 0.580 | 0.659 | **0.818** | 0.701 |
| Marg. NB   |       | 0.702 | 0.904 | 0.637 | 0.570 | 0.633 | 0.490 | 0.620 | 0.814 | 0.675 |
| kNN        |       | **0.795** | **0.927** | 0.159 | 0.558 | 0.367 | 0.525 | 0.445 | 0.554 | 0.582 |
| CART       |       | 0.680 | 0.840 | 0.278 | 0.550 | 0.449 | 0.498 | 0.545 | 0.700 | 0.598 |
| CART-Bag   |       | 0.752 | 0.894 | 0.191 | **0.598** | 0.610 | 0.541 | 0.582 | 0.785 | 0.660 |

making it more susceptible to errors. Most likely this means that a more extensive search would further favor the SGM classifier.

6 Discussion

We introduced predictive Bayesian classifiers that utilize the dependence structure of the observed features to enhance the accuracy of classification, by allowing a more faithful representation of the data generative process. Albeit we did not consider it explicitly, an additional beneficial characteristic of such an approach is that the uncertainty of the class labels is then more appropriately characterized by the predictive distribution, because the parameters are integrated out with respect to their posterior
By asymmetric loss we mean that an erroneous label is associated with a higher cost for a certain class compared to other classes, such as diagnosis of illness for instance. For a general discussion of the loss functions, as well as predictive and non-predictive inference in classification, see Ripley (1996). While the naive predictive Bayes classifier is simple and straightforward to use, it often oversimplifies the problem by assuming independence among the features, which has been widely acknowledged in the literature. GM classifiers attempt to rectify the problem by introducing a dependence structure for the features. However, the family of dependence structures that can be modeled using GMs can in some cases be too rigid. The ability to include context-specific independencies among the features, introduced by SGMs, allows for a more precise and sparse representation of the dependence structure.

The results presented in this paper demonstrate the potential of SGM classifiers to improve the rate of success with which the items are classified. Additionally, it is shown that when the data includes a sufficient amount of features, leading to a high success rate of classification, a simultaneous classifier is advantageous compared to the separate classification of each sample which is the standard approach.

In future research it would be interesting to consider SGM classifiers in the context of sequentially arising data, such as discussed in Corander et al. (2013c). Kernel methods (see e.g. Bishop 2007) would also possibly allow a generalization of the context-specific dependence to continuous variables in the sequential case. However, since such methods are generally computation intensive, very efficient fast approximations would need to be used in online type applications, such as in speech recognition and other similar sequential signal processing (Huo and Lee 2000; Maina and Walsh 2011). Finally, another potential area of further development would be to generalize the context-specific independence for an application to Gaussian classifiers.

Acknowledgments The authors would like to thank the editor and the anonymous reviewers for their constructive comments and suggestions on the original version of this paper. H.N. and J.P. were supported by the Foundation of Åbo Akademi University, as part of the grant for the Center of Excellence in Optimization and Systems Engineering. J.P. was also supported by the Magnus Ehrnrooth foundation. J.X. and J.C. were supported by the ERC Grant No. 239784 and Academy of Finland Grant No. 251170. J.X. was also supported by the FDPSS graduate school.

Appendix A: Proof of Theorem 1

To prove Theorem 1 it suffices to consider a single class \(k\) and a single maximal clique in \(G^k_L\). If the scores for the marginal and simultaneous classifiers are asymptotically equivalent for an arbitrary maximal clique and class it automatically follows that the scores for the whole system are asymptotically equivalent. We start by considering the simultaneous classifier. The training data \(X^R\) and test data \(X^T\) are now assumed to cover only one maximal clique of an SG in one class. Looking at \(\log S_{\text{sim}}(X^T | X^R)\) using (2) we get
\[
\log S_{\text{sim}}(X^T \mid X^R) = \sum_{j=1}^{d} \sum_{l=1}^{q_j} \log \frac{\Gamma(\sum_{i=1}^{k_j} \beta_{jil})}{\Gamma(n(\pi_j^l) + \sum_{i=1}^{k_j} \beta_{jil})} \\
+ \sum_{j=1}^{d} \sum_{l=1}^{q_j} \sum_{i=1}^{k_j} \log \frac{\Gamma(n(x_j^l \mid \pi_j^l) + \beta_{jil})}{\Gamma(\beta_{jil})}.
\]

Using Stirling’s approximation, \( \log \Gamma(x) = (x - 0.5) \log(x) - x \), this equals

\[
\sum_{j=1}^{d} \sum_{l=1}^{q_j} \left( \left( \sum_{i=1}^{k_j} \beta_{jil} - 0.5 \right) \log \left( \sum_{i=1}^{k_j} \beta_{jil} \right) - \sum_{i=1}^{k_j} \beta_{jil} \right) \\
- \sum_{j=1}^{d} \sum_{l=1}^{q_j} \left( n(\pi_j^l) + \sum_{i=1}^{k_j} \beta_{jil} - 0.5 \right) \log \left( n(\pi_j^l) + \sum_{i=1}^{k_j} \beta_{jil} \right) - n(\pi_j^l) - \sum_{i=1}^{k_j} \beta_{jil} \right) \\
+ \sum_{j=1}^{d} \sum_{l=1}^{q_j} \left( \left( n(x_j^l \mid \pi_j^l) + \beta_{jil} - 0.5 \right) \log \left( n(x_j^l \mid \pi_j^l) + \beta_{jil} \right) - n(x_j^l \mid \pi_j^l) - \beta_{jil} \right) \\
- \sum_{j=1}^{d} \sum_{l=1}^{q_j} \sum_{i=1}^{k_j} \left( \beta_{jil} - 0.5 \right) \log(\beta_{jil}) - \beta_{jil} \right) \\
= - \sum_{j=1}^{d} \sum_{l=1}^{q_j} \left( \left( \sum_{i=1}^{k_j} \beta_{jil} - 0.5 \right) \log \left( 1 + \frac{n(\pi_j^l)}{\sum_{i=1}^{k_j} \beta_{jil}} \right) + n(\pi_j^l) \log \left( n(\pi_j^l) + \sum_{i=1}^{k_j} \beta_{jil} \right) \right) \\
+ \sum_{j=1}^{d} \sum_{l=1}^{q_j} \sum_{i=1}^{k_j} \left( \beta_{jil} - 0.5 \right) \log \left( 1 + \frac{n(x_j^l \mid \pi_j^l)}{\beta_{jil}} \right) + n(x_j^l \mid \pi_j^l) \log \left( n(x_j^l \mid \pi_j^l) + \beta_{jil} \right). 
\]

When looking at the marginal classifier we need to summarize over each single observation \( X_h^T \). We use \( h(\pi_j^l) \) to denote if the outcome of the parents of variable \( X_j \) belongs to group \( l \) and \( h(x_j^l \mid \pi_j^l) \) to denote if the outcome of \( X_j \) is \( i \) given that the observed outcome of the parents belongs to \( l \). Observing that \( h(\pi_j^l) \) and \( h(x_j^l \mid \pi_j^l) \) are either 0 or 1 we get the result

\[
\log S_{\text{mar}}(X^T \mid X^R) = \sum_{h=1}^{n} \log P(X_h^T \mid X^R) \\
= \sum_{h=1}^{n} \sum_{j=1}^{d} \sum_{l=1}^{q_j} \log \frac{\Gamma(\sum_{i=1}^{k_j} \beta_{jil})}{\Gamma(h(\pi_j^l) + \sum_{i=1}^{k_j} \beta_{jil})} \\
+ \sum_{h=1}^{n} \sum_{j=1}^{d} \sum_{l=1}^{q_j} \sum_{i=1}^{k_j} \log \frac{\Gamma(h(x_j^l \mid \pi_j^l) + \beta_{jil})}{\Gamma(\beta_{jil})} \\
= - \sum_{j=1}^{d} \sum_{l=1}^{q_j} n(\pi_j^l) \log \left( \sum_{i=1}^{k_j} \beta_{jil} \right) + \sum_{j=1}^{d} \sum_{l=1}^{q_j} \sum_{i=1}^{k_j} n(x_j^l \mid \pi_j^l) \log(\beta_{jil}).
\]
Considering the difference \( \log S_{\text{sim}}(X^T | X^R) - \log S_{\text{mar}}(X^T | X^R) \) results in

\[
\log S_{\text{sim}}(X^T | X^R) - \log S_{\text{mar}}(X^T | X^R) \\
= - \sum_{j=1}^{d} \sum_{l=1}^{q_j} \left( \sum_{i=1}^{k_j} \beta_{jil} - 0.5 \right) \log \left( 1 + \frac{n(\pi_j^l)}{\sum_{i=1}^{k_j} \beta_{jil}} \right) \\
+ \sum_{j=1}^{d} \sum_{l=1}^{q_j} \sum_{i=1}^{k_j} (\beta_{jil} - 0.5) \log \left( 1 + \frac{n(x_j^l | \pi_j^l)}{\beta_{jil}} \right) \\
+ \sum_{j=1}^{d} \sum_{l=1}^{q_j} \left( n(\pi_j^l) \log \left( \sum_{i=1}^{k_j} \beta_{jil} \right) - n(\pi_j^l) \log \left( n(\pi_j^l) + \sum_{i=1}^{k_j} \beta_{jil} \right) \right) \\
+ \sum_{j=1}^{d} \sum_{l=1}^{q_j} \sum_{i=1}^{k_j} \left( n(x_j^l | \pi_j^l) \log \left( n(x_j^l | \pi_j^l) + \beta_{jil} \right) - n(x_j^l | \pi_j^l) \log(\beta_{jil}) \right).
\]

We now make the assumption that all the limits of relative frequencies of feature values are strictly positive under an infinitely exchangeable sampling process of the training data, i.e. all hyperparameters \( \beta_{jil} \to \infty \) when the size of the training data \( m \to \infty \). Using the standard limit \( \lim_{\gamma \to \infty} (1 + x/y)^\gamma = e^x \) results in

\[
\lim_{m \to \infty} \log S_{\text{sim}}(X^T | X^R) - \log S_{\text{mar}}(X^T | X^R) \\
= - \sum_{j=1}^{d} \sum_{l=1}^{q_j} n(\pi_j^l) + \sum_{j=1}^{d} \sum_{l=1}^{q_j} \sum_{i=1}^{k_j} n(x_j^l | \pi_j^l) = 0.
\]

**Appendix B: Proof of Theorem 2**

This proof follows largely the same structure as the proof of Theorem 1 and covers the simultaneous score. It is assumed that the underlying graph of the SGM coincides with the GM, this is a fair assumption since when the size of the training data goes to infinity this property will hold for the SGM and GM maximizing the marginal likelihood. Again we consider only a single class \( k \) and a single maximal clique in \( G^l_k \), using the same reasoning as in the proof above. Additionally, it will suffice to consider the score for the last variable \( X_d \) in the ordering, the variable corresponding to the node associated with all of the stratified edges, and a specific parent configuration \( l \) of the parents \( \Pi_d \) of \( X_d \). The equation for calculating the score for variables \( X_1, \ldots, X_{d-1} \) will be identical using either the GM or the SGM. If the asymptotic equivalence holds for an arbitrary parent configuration it automatically holds for all parent configurations. Under this setting we start by looking at the score for the SGM

\[
\log S_{\text{SGM}}(X^T | X^R) = \log \frac{\Gamma(\sum_{i=1}^{k_j} \beta_{jil})}{\Gamma(n(\pi_j^l) + \sum_{i=1}^{k_j} \beta_{jil})} + \sum_{i=1}^{k_j} \log \frac{\Gamma(n(x_j^l | \pi_j^l) + \beta_{jil})}{\Gamma(\beta_{jil})}.
\]
which using Stirling’s approximation and the same techniques as in the previous proof equals

\[- \left( \sum_{i=1}^{k_j} \beta_{jil} - 0.5 \right) \log \left( 1 + \frac{n(\pi_j^l)}{\sum_{i=1}^{k_j} \beta_{jil}} \right) - n(\pi_j^l) \log \left( n(\pi_j^l) + \sum_{i=1}^{k_j} \beta_{jil} \right) + \sum_{i=1}^{k_j} n(x_j^l \mid \pi_j^l) \log(n(x_j^l \mid \pi_j^l) + \beta_{jil}) + \sum_{i=1}^{k_j} (\beta_{jil} - 0.5) \log \left( 1 + \frac{n(x_j^l \mid \pi_j^l)}{\beta_{jil}} \right).\]

When studying the GM score we need to separately consider each outcome in the parent configuration \( l \). Let \( h \) denote such an outcome in \( l \) with the total number of outcomes in \( l \) totaling \( q_l \). We then get the following score for the GM,

\[
\log S_{GM}(X^T \mid X^R) = \sum_{h=1}^{q_l} \log \frac{\Gamma(\sum_{i=1}^{k_j} \beta_{jih})}{\Gamma(n(\pi_j^h) + \sum_{i=1}^{k_j} \beta_{jih})} + \sum_{h=1}^{q_l} \sum_{i=1}^{k_j} \log \frac{\Gamma(n(x_j^l \mid \pi_j^h) + \beta_{jih})}{\Gamma(\beta_{jih})}.
\]

Which, using identical calculations as before, equals

\[- \sum_{h=1}^{q_l} \left( \sum_{i=1}^{k_j} \beta_{jih} - 0.5 \right) \log \left( 1 + \frac{n(\pi_j^h)}{\sum_{i=1}^{k_j} \beta_{jih}} \right) - \sum_{h=1}^{q_l} n(\pi_j^h) \log \left( n(\pi_j^h) + \sum_{i=1}^{k_j} \beta_{jih} \right) + \sum_{h=1}^{q_l} \sum_{i=1}^{k_j} n(x_j^l \mid \pi_j^h) \log(n(x_j^l \mid \pi_j^h) + \beta_{jih}) + \sum_{h=1}^{q_l} \sum_{i=1}^{k_j} (\beta_{jih} - 0.5) \log \left( 1 + \frac{n(x_j^l \mid \pi_j^h)}{\beta_{jih}} \right).\]

Considering the difference \( \log S_{SGM}(X^T \mid X^R) - \log S_{GM}(X^T \mid X^R) \) we get

\[- \left( \sum_{i=1}^{k_j} \beta_{jil} - 0.5 \right) \log \left( 1 + \frac{n(\pi_j^l)}{\sum_{i=1}^{k_j} \beta_{jil}} \right) + \sum_{i=1}^{k_j} (\beta_{jil} - 0.5) \log \left( 1 + \frac{n(x_j^l \mid \pi_j^l)}{\beta_{jil}} \right) - n(\pi_j^l) \log \left( n(\pi_j^l) + \sum_{i=1}^{k_j} \beta_{jil} \right) + \sum_{h=1}^{q_l} n(\pi_j^h) \log \left( n(\pi_j^h) + \sum_{i=1}^{k_j} \beta_{jih} \right) + \sum_{h=1}^{q_l} \sum_{i=1}^{k_j} (\beta_{jih} - 0.5) \log \left( 1 + \frac{n(x_j^l \mid \pi_j^h)}{\beta_{jih}} \right) + \sum_{i=1}^{k_j} n(x_j^l \mid \pi_j^l) \log(n(x_j^l \mid \pi_j^l) + \beta_{jil}) - \sum_{h=1}^{q_l} \sum_{i=1}^{k_j} n(x_j^l \mid \pi_j^h) \log(n(x_j^l \mid \pi_j^h) + \beta_{jih})\]
\[ + \sum_{h=1}^{q_l} \left( \frac{k_j}{h=1} \sum_{i=1}^{k_j} \beta_{jih} - 0.5 \right) \log \left( 1 + \frac{n(\pi^h_j)}{\sum_{i=1}^{k_j} \beta_{jih}} \right) \]

\[ - \sum_{h=1}^{q_l} \sum_{i=1}^{k_j} (\beta_{jih} - 0.5) \log \left( 1 + \frac{n(x^i_j | \pi^h_j)}{\beta_{jih}} \right). \]

Under the assumption that \( \beta_{jil} \to \infty \) as \( m \to \infty \), the terms in rows one and four will sum to 0 as \( m \to \infty \). The remaining terms can be written

\[ \log \frac{\prod_{h=1}^{q_l} (n(\pi^h_j) + \sum_{i=1}^{k_j} \beta_{jih} \pi(\pi^h_j))}{\prod_{h=1}^{q_l} (n(x^i_j | \pi^h_j) + \sum_{i=1}^{k_j} \beta_{jih} \pi(x^i_j | \pi^h_j))}. \]

Noting that \( n(\pi^h_j) = \sum_{h=1}^{q_l} n(\pi^h_j) \) and \( n(x^i_j | \pi^h_j) = \sum_{h=1}^{q_l} n(x^i_j | \pi^h_j) \) we get

\[ \sum_{h=1}^{q_l} n(\pi^h_j) \log \frac{n(\pi^h_j) + \sum_{i=1}^{k_j} \beta_{jih} \pi(\pi^h_j)}{n(\pi^h_j) + \sum_{i=1}^{k_j} \beta_{jil} \pi(\pi^h_j)} - \sum_{h=1}^{q_l} \sum_{i=1}^{k_j} n(x^i_j | \pi^h_j) \log \frac{n(x^i_j | \pi^h_j) + \beta_{jih}}{n(x^i_j | \pi^h_j) + \beta_{jil}}. \]

By investigating the definition of the \( \beta \) parameters in (3), in combination with the fact that the probabilities of observing the value \( i \) for variable \( X_j \) given that the outcome of the parents is \( h \) are identical for any outcome \( h \) comprising the group \( l \), we get the limits

\[ \lim_{m \to \infty} \frac{n(\pi^h_j) + \sum_{i=1}^{k_j} \beta_{jih}}{n(\pi^h_j) + \sum_{i=1}^{k_j} \beta_{jil}} = \lim_{m \to \infty} \frac{n(x^i_j | \pi^h_j) + \beta_{jih}}{n(x^i_j | \pi^h_j) + \beta_{jil}} = \zeta_{jh}. \]

And subsequently as \( m \to \infty \) the difference \( \log S_{SGM}(X^T | X^R) - \log S_{GM}(X^T | X^R) \) →

\[ \sum_{h=1}^{q_l} \left( n(\pi^h_j) \log \zeta_{jh} - \sum_{i=1}^{k_j} n(x^i_j | \pi^h_j) \log \zeta_{jh} \right) \]

\[ = \sum_{h=1}^{q_l} \left( n(\pi^h_j) \log \zeta_{jh} - n(\pi^h_j) \log \zeta_{jh} \right) = 0. \]
Appendix C: List of political parties in the Finnish parliament

Table 4 contains a list of political parties elected to the Finnish parliament in the parliament elections of 2011.

| Label | Political party                                |
|-------|-----------------------------------------------|
| 1     | National Coalition Party                      |
| 2     | Finns Party                                   |
| 3     | Swedish People’s Party of Finland             |
| 4     | Centre Party                                  |
| 5     | Christian Democrats of Finland                |
| 6     | Social Democratic Party of Finland            |
| 7     | Left Alliance                                 |
| 8     | Green League                                  |

References

Bishop CM (2007) Pattern recognition and machine learning. Springer, New York
Cerquides J, De Mántaras RL (2005) TAN classifiers based on decomposable distributions. Mach Learn 59(3):323–354
Cooper GF, Herskovits E (1992) A Bayesian method for the induction of probabilistic networks from data. Mach Learn 9(4):309–347
Corander J, Marttinen P (2006) Bayesian identification of admixture events using multi-locus molecular markers. Mol Ecol 15(10):2833–2843
Corander J, Marttinen P, Sirén J, Tang J (2008) Enhanced Bayesian modelling in BAPS software for learning genetic structures of populations. BMC Bioinform 9:539
Corander J, Cui Y, Koski T (2013a) Inductive inference and partition exchangeability in classification. In: Dowle DL (ed) Solomonoff Festschrift, Springer Lecture Notes in Artificial Intelligence (LNAI), vol 7070, pp 91–105
Corander J, Cui Y, Koski T, Sirén J (2013b) Have I seen you before? Principles of Bayesian predictive classification revisited. Stat Comput 23(1):59–73
Corander J, Xiong J, Cui Y, Koski T (2013c) Optimal Viterbi Bayesian predictive classification for data from finite alphabets. J Stat Plan Infer 143(2):261–275
Dawid A, Lauritzen S (1993) Hyper-Markov laws in the statistical analysis of decomposable graphical models. Ann Stat 21:1272–1317
Dawnyndt P, Thompson FL, Austin B, Swings J, Koski T, Gyllenberg M (2005) Application of sliding-window discretization and minimization of stochastic complexity for the analysis of fAFLP genotyping fingerprint patterns of Vibrionaceae. Int J Syst Evol Microbiol 55(1):57–66
Duda RO, Hart PE, Stork DG (2000) Pattern classification, 2nd edn. Wiley, New York
Friedman N, Geiger D, Goldszmidt M (1997) Bayesian network classifiers. Mach Learn 29(2–3):131–163
Geisser S (1964) Posterior odds for multivariate normal classifications. J R Stat Soc B 26:69–76
Geisser S (1966) Predictive discrimination. In: Krishnajah PR (ed) Multivariate analysis. Academic Press, New York
Geisser S (1993) Predictive inference: an introduction. Chapman & Hall, London
Golumbic MC (2004) Algorithmic graph theory and perfect graphs, 2nd edn. Elsevier, Amsterdam
Hastie T, Tibshirani R, Friedman J (2009) The elements of statistical learning: data mining, inference, and prediction, 2nd edn. Springer, New York
Helsingin Sanomat (2011) HS:n vaalikone 2011. http://www2.hs.fi/extrat/hsnext/HS-vaalikone2011.xls, visited 15 Oct 2013
Holmes DE, Jain LC (2008) Innovations in Bayesian networks: theory and applications, vol 156. Springer, Berlin
Huo Q, Lee CH (2000) A Bayesian predictive classification approach to robust speech recognition. IEEE Trans Speech Audio Process 8(2):200–204
Keogh EJ, Pazzani MJ (1999) Learning augmented Bayesian classifiers: a comparison of distribution-based and classification-based approaches. In: Proceedings of the seventh international workshop on artificial intelligence and statistics, pp 225–230
Koller D, Friedman N (2009) Probabilistic graphical models: principles and techniques. The MIT Press, London
Lauritzen SL (1996) Graphical models. Oxford University Press, Oxford
Madden MG (2009) On the classification performance of TAN and general Bayesian networks. Knowl Based Syst 22(7):489–495
Maina CW, Walsh JM (2011) Joint speech enhancement and speaker identification using approximate Bayesian inference. IEEE Trans Audio Speech Lang Process 19(6):1517–1529
Nádas A (1985) Optimal solution of a training problem in speech recognition. IEEE Trans Acoustics Speech Signal Process 33(1):326–329
Nyman H, Pensar J, Koski T, Corander J (2014) Stratified graphical models—context-specific independence in graphical models. Bayesian Anal 9(4):883–908
Pernkopf F, Bilmes J (2005) Discriminative versus generative parameter and structure learning of Bayesian network classifiers. In: Proceedings of the 22nd international conference on machine learning, pp 657–664
Ripley BD (1988) Statistical inference for spatial processes. Cambridge University Press, Cambridge
Ripley BD (1996) Pattern recognition and neural networks. Cambridge University Press, Cambridge
Su J, Zhang H (2006) Full Bayesian network classifiers. In: Proceedings of the 23rd international conference on machine learning, pp 897–904
Whittaker J (1990) Graphical models in applied multivariate statistics. Wiley, Chichester