Penalized estimation in high-dimensional hidden Markov models with state-specific graphical models

Nicolas Städler
Netherlands Cancer Institute
Amsterdam, Netherlands.
n.stadler@nki.nl

Sach Mukherjee
Netherlands Cancer Institute
Amsterdam, Netherlands.
s.mukherjee@nki.nl

August 27, 2012

Abstract

We consider penalized estimation in hidden Markov models (HMMs) with multivariate Normal observations. In the moderate-to-large dimensional setting, estimation for HMMs remains challenging in practice, due to several concerns arising from the hidden nature of the states. We address these concerns by ℓ_1-penalization of state-specific inverse covariance matrices. Penalized estimation leads to sparse inverse covariance matrices which can be interpreted as state-specific conditional independence graphs. Penalization is non-trivial in this latent variable setting; we propose a penalty that automatically adapts to number of states K and state-specific sample size and can cope with scaling issues arising from the unknown states. The methodology is adaptive and very general, applying in particular to both low- and high-dimensional settings without requiring hand tuning. Furthermore, our approach facilitates exploration of the number of states K by coupling estimation for successive candidate values K. Empirical results on simulated examples demonstrate the effectiveness of the proposed approach. In a challenging real data example from genome biology, we demonstrate the ability of our approach to yields gains in predictive power and deliver richer estimates than existing methods.

Keywords HMM, Graphical Lasso, Universal Regularization, Model Selection, MMDL, Greedy Backwards Pruning, Genome Biology, Chromatin Modeling

1 Introduction

In this paper we consider estimation in high-dimensional hidden Markov models. We consider multivariate observations $X_t \in \mathbb{R}^p$ with discrete index $t \in \mathcal{T} = \{1 \dots n\}$ and hidden states $S_t \in \{1 \dots K\}$. Conditional on state, emission distributions are multivariate Normal (MVN), with $X_t \mid S_t = k \sim \mathcal{N}(\mu_k, \Sigma_k)$ (where, $\mathcal{N}(\mu, \Sigma)$ denotes the MVN density with mean μ and covariance matrix Σ). Estimation in the small p case of univariate or low-dimensional observations is a well-studied problem. In contrast, estimation in the larger p setting remains challenging due to several factors:

- *High-dimensionality.* Inference in HMMs with moderate or large number of features is, in a sense, always a high-dimensional problem since the ratio $\min_k n_k/p$ may be small, as it depends on the *unknown* number of states and the *unknown* size of the states (n_k denotes the number of samples in state k). Therefore, large samples for each state cannot be relied upon at the outset, even when the overall sample size $n = \sum_k n_k$ is large.
- *Covariance structure.* Estimation is especially challenging in settings where covariances Σ_k cannot be assumed to have a simple structure (e.g. diagonal) or where state-specific covariance structure is itself of scientific interest. Then, due to Simpson's paradox, state-specific covariances must be jointly estimated along with state assignments.
- *Number of hidden states.* The model selection problem of determining or exploring the number of states K is coupled to the estimation problem for known K. In the multivariate setting, estimation for known K is itself challenging. Then, the straightforward strategy of fitting models for various values K and comparing by model selection criteria may become difficult or intractable, especially when practically important issues like initialization and setting of tuning parameters are taken into consideration.

We propose a penalized log-likelihood procedure involving ℓ_1-norms of the state-specific inverse covariance matrices Σ_k^{-1}, with optimization carried out within an expectation-maximization (EM) framework. Our approach has several attractive features:

- Penalized estimation leads to sparse inverse covariance matrices which can be interpreted as state-specific conditional independence graphs or networks (Yuan and Lin, 2007; Friedman *et al.*, 2008).
- The specific penalty we propose automatically adapts to the number of states and state-specific sample size and enjoys scale invariance that takes care of state-specific scaling.
- The number of states K can be selected automatically, or estimates for various values K explored, using a computationally efficient approach that couples estimation for successive candidate values for K.
- The approach requires essentially no hand tuning; only a maximum number of states $K_{\max}$ must be set by the user. Otherwise, tuning parameters (including, if desired,

K itself) are set automatically.

Our approach is very general: as we demonstrate below it works well in diverse regimes, including both low- and high-dimensional examples, with no hand-tuning required. In a real data example from genomics the methodology leads to large gains in predictive power relative to existing approaches.

Penalized estimators can be incorporated into EM-type algorithms and a number of recent authors have done so, notably in the context of mixture models (Khalili and Chen, 2007; Städler *et al.*, 2010; Pan and Shen, 2007). However, the unknown nature of the states (or mixture components) poses special challenges for penalization that have not been adequately addressed so far. In particular, appropriate penalization must account for the number of hidden states and their respective sample sizes, but these are themselves unknown at the outset. Furthermore, scaling also poses a subtle problem: in the classical Lasso (Tibshirani, 1996) or Graphical Lasso (Friedman *et al.*, 2008) standardization is an important pre-processing step to ensure appropriate scaling. However, in HMMs and mixtures different states or components may differ with respect to scale, but since state assignments are *a priori* unknown, standardization cannot be carried out as a pre-processing step. The penalty we propose automatically adapts with state-sizes and takes care of scaling issues. Inspired by the seminal paper of Donoho and Johnstone (1994), and related work in the Lasso context (Zhang, 2010; Sun and Zhang, 2011; Barron *et al.*, 2008), our penalty allows for *universal regularization* by use of a tuning parameter λ_{uni}, that depends only on n and p. Using universal regularization by λ_{uni} within our EM algorithm allows automatic adaptation to number of states K and state-specific sample sizes. As a consequence of these features, our procedure for penalized estimation for a given number of states K is entirely free of user-set parameters.

Parameter estimates for successive values $K, K+1$ are related, and it is therefore natural to exploit this fact in exploring the number of states; we do so using an iterative algorithm. In principle, an iterative approach could proceed in a "top down" manner from few states to many, or "bottom up" from many states to few. However, we cannot in general gain information about two underlying states from estimates obtained from a single, merged state (Simpson's paradox); this means the "top down" approach cannot be reliably used in the multivariate setting. We therefore proceed in a "bottom up" manner, starting with a large number of states $K_{\max}$ and iteratively reducing the number of states through the entire considered range. Model order reduction is guided using the Kullback-Leibler divergence between state densities; this naturally takes account of both mean and covariance information. This exploration is efficient because (i) current estimates are used to provide initialization for the subsequent iteration and (ii) we initialize the EM algorithm only once, at the first iteration corresponding to $K = K_{\max}$. As we demonstrate below, this procedure in fact outperforms the "brute-force" approach of entirely separately fitting models for various K's. In this way, our approach allows tractable exploration of estimates for a range of values K and, if desired, automatic selection of K. Our approach is inspired by the work of Figueiredo and Jain (2000) who used a similar strategy in the context of low-dimensional mixtures.

Applications for high-dimensional HMMs are numerous, in fields ranging from engineering to biology. The original motivation for our work comes from genome biology and we

illustrate our methods on a real data example from that field. HMMs are very widely used in genomics. Measurements at genome locations t constitute the vector X_t while states S_t are identified with biological states of the genome (e.g. whether the location t is within a gene-coding region). Early, pioneering applications of HMMs to genomic data (see e.g. Durbin *et al.*, 1998) considered univariate or low-dimensional observations X_t (such as the gene sequence itself). However, in recent years technological advances have begun to permit higher dimensional studies. For example, using technologies such as DamID (van Steensel and Henikoff, 2000) or ChIP-seq (Park, 2009) it is now possible to measure the binding of proteins to the DNA across the entire genome for dozens or hundreds of proteins and the dimensionality (i.e. number of proteins) of such approaches continues to increase. However, absent reliable methodology for fitting high-dimensional HMMs, it is common practice in the field to instead consider reduced dimension versions of the data (by selecting key "marker" variables or carrying out dimensionality reduction as a pre-processing step, see e.g. Filion *et al.* (2010)) or by discretizing the data and treating observations as independent Bernoulli (Ernst and Kellis, 2010). We show below in a real data example from genome biology that our penalized approach applied to all available variables (proteins) from a recent experiment yields large gains in predictive accuracy (on held-out test data) relative to a reduced-dimension approach, as well as relative to classical estimation applied to the full set of variables.

2 Inference in hidden Markov models with state-specific graphical models

We consider a hidden Markov model (HMM) with multivariate Normal (MVN) emissions. We denote by $S_t \in \{1, \ldots, K\}$ the (hidden) state process, i.e., a discrete Markov chain with transition matrix $\Pi_{kk'} = \mathrm{P}(S_{t+1} = k' | S_t = k)$; in order to simplify the notation we omit the initial probabilities $p_k = \mathrm{P}(S_1 = k)$ in the further description of our methodology. We denote by $X_t \in \mathbb{R}^p$ the observed process with emission distribution $X_t \mid S_t = k \sim \mathcal{N}(\mu_k, \Sigma_k)$.

The case of sparse inverse covariance matrices $\Omega_k = \Sigma_k^{-1}$ will be of particular interest. For each state we have a Gaussian graphical model with undirected graph G_k defined by locations of zero entries in the inverse covariance matrix, i.e. $(l, l') \notin G_k \iff (\Omega_k)_{ll'} = 0$. We denote model parameters by $\Theta_K = (\theta_1, \ldots, \theta_K, \Pi)$, $\theta_k = (\mu_k, \Omega_k)$. The goal, for given K, is to infer Θ_K from the observed $n \times p$ data matrix $\mathbf{X}$, and further to solve the related problem of exploring (or determining) K itself.

As noted in the Introduction, inference for multivariate HMMs is challenging due to difficulties arising from the unknown states. To motivate the methods described in this Section, we gather these points together, highlighting four main difficulties/concerns:

(i) *High-dimensionality.* Inference in HMMs with moderate or large number of features is, in a sense, *always* high-dimensional, as the ratio $\min_k \frac{n_k}{p}$ may be small, and depends on the number of states and their size, both of which are usually unknown at the outset.

(ii) *State-specific covariance structure.* When state-specific covariance structure is of interest or cannot be assumed to be diagonal, estimation is challenging. Importantly, due to Simpson's paradox, covariances must be jointly estimated along with other parameters.

(iii) *Regularization.* The size and scale of individual states may vary and are usually unknown at the outset. Regularization schemes need to self-adapt appropriately.

(iv) *Model order exploration.* In the HMM setting, model selection criteria are a function of both number of states K and amount of regularization λ. However, brute force search over (K, λ) may become intractable in practice, especially since at each grid point, multiple initializations are needed to guard against local optima.

Points (i)-(iv) above are coupled and are difficult or impossible to address individually. The methodology we propose aims to address all these concerns simultaneously. We first outline the approach at a high-level and then describe the algorithms in detail.

Conceptually, it makes sense to think of inference in a HMM (or mixture model) as a combination of two (coupled) tasks. The first task consists of estimating the model parameter Θ_K, given the number of states K and a regularization parameter λ. For this task, we propose to minimize the negative penalized log-likelihood

$$\hat{\Theta}_{K,\lambda} \quad = \quad \underset{\Theta_{K,\lambda}}{\operatorname{argmin}} -\ell(\Theta_{K,\lambda}; \mathbf{X}) + \lambda \operatorname{pen}(\Theta_{K,\lambda}), \tag{2.1}$$

where $\ell(\Theta_{K,\lambda}; \mathbf{X})$ denotes the observed log-likelihood and $\operatorname{pen}(\Theta_{K,\lambda})$ is a penalty function involving the ℓ_1-norms of the inverse covariance matrices (Yuan and Lin, 2007; Friedman *et al.*, 2008; Meinshausen and Bühlmann, 2006) that we describe in detail below. The ℓ_1-norm is especially appealing when the goal is network inference, as it induces sparsity in Ω_k's and therefore in the corresponding undirected graphs G_k. We solve this problem by an EM-type algorithm, using a specific penalty that we describe below; we call this approach **HMMGLasso** (see Section 2.1 for details). As with every EM algorithm, HMMGLasso only converges to a local optimum and depends on initialization.

The second task involves determining an appropriate number of states K^*, and suitable penalization parameter λ^*. This is a model selection problem, and can in principle be solved by minimizing a model selection criterion $\mathcal{C}(K, \lambda)$ (we consider specific criteria below), i.e.,

$$(K^*, \lambda^*) \quad = \quad \underset{K,\lambda}{\operatorname{argmin}}\, \mathcal{C}(K, \lambda). \tag{2.2}$$

Thus, estimation in multivariate HMMs involves two coupled tasks whose solution must address the concerns (i)-(iv) listed above. In outline, we proceed as follows. We propose a *universal regularization* level λ_{uni} that can be calculated analytically and removes the need for brute force search over λ. The idea is, that solving (2.1) with the HMMGlasso penalty function that we describe in Section 2.1 and with $\lambda = \lambda_{\text{uni}}$ (see Section 2.2) should provide for each fixed K a close-to-optimal solution for that specific K. In this way, use of universal regularization λ_{uni} reduces solving (2.2) to a search over K only.

We also exploit the relationship between estimates $\hat{\Theta}_K$ for successive K's to allow efficient model exploration and, if desired, determination of K. Specifically, we start with a (too) large number of states $K_{\max}$ and then successively reduce the model size by merge/delete operations described below. In this way we move towards a minimum number of states $K_{\min}$, obtaining estimates for all considered values $K_{\min} \leq K \leq K_{\max}$, but initializing only once, at $K = K_{\max}$. If desired, a specific K can then be chosen to minimize model selection criterion $\mathcal{C}(\cdot, \lambda_{\text{uni}})$. We call this approach **Greedy Backward Pruning**. As we show below, following the Greedy Backward Pruning strategy gives highly competitive estimates, despite initializing only once in the entire procedure.

In summary, the adaptive regularization strategy we propose in HMMGLasso permits estimation of HMMs with state-specific covariance structure in both low- and high-dimensional settings, whilst taking care of state size and scaling; this addresses points (i)-(iii) above. Greedy Backward Pruning, with initialization only at $K = K_{\max}$, takes care of point (iv).

The remainder of this Section provides a detailed description of our algorithms. We discuss in turn HMMGLasso (Section 2.1), universal regularization (Section 2.2) and model order exploration by Greedy Backward Pruning (Section 2.3).

2.1 HMMGLasso in detail: Baum-Welch algorithm and ℓ_1 regularization

Maximum likelihood estimation for HMM is usually performed using the EM algorithm (or Baum-Welch algorithm in the HMM context). Let $\ell_c(\Theta; \mathbf{X}, \mathbf{S})$

$$\ell_c(\Theta; \mathbf{X}, \mathbf{S}) = \sum_k \ell(\mu_k, \Omega_k; \mathbf{T}_1, \mathbf{T}_2) + \ell(\Pi; \mathbf{T}_3), \tag{2.3}$$

where $\mathbf{S} = (S_1, \ldots, S_n)$ are state assignments, $\mathbf{X} = (X_1, \ldots, X_n)^{\mathrm{T}}$ is the $n \times p$ data matrix, $\ell(\mu_k, \Omega_k; \mathbf{T}_1, \mathbf{T}_2)$ is the log-likelihood of the MVN distribution with mean μ_k and inverse covariance Ω_k and $\ell(\Pi; \mathbf{T}_3)$ is the log-likelihood of the Markov Chain with transition matrix Π. $\mathbf{T}_1 = \mathbf{X}^T\mathbf{1}, \mathbf{T}_2 = \mathbf{X}\mathbf{X}^T$ and $(\mathbf{T}_3)_{kk'} = \sum_t \mathbf{I}(S_t = k, S_{t+1} = k')$ are the corresponding sufficient statistics.

Following initialization, EM produces a sequence of estimates $\{\Theta^{(i)}; i = 1, 2, 3, \ldots\}$ by alternating between E- and M-steps. To facilitate network inference, we seek to induce sparsity in the Ω_k's. We do this by ℓ_1-regularization. In particular, we replace maximization with respect to (μ_k, Ω_k) in the M-Step of the Baum-Welch algorithm by

$$(\mu_k^{(i+1)}, \Omega_k^{(i+1)}) = \underset{\mu_k, \Omega_k}{\operatorname{argmin}} -\ell(\mu_k, \Omega_k; \mathbf{T}_1^{u_k^{(i)}}, \mathbf{T}_2^{u_k^{(i)}}) + \lambda\sqrt{\pi_k^{(i)}}\,\mathrm{Pen}(\Omega_k). \tag{2.4}$$

Here,

$$\mathbf{T}_1^{u_k^{(i)}} = \sum_t u_k^{(i)}(t) X_t, \quad \mathbf{T}_2^{u_k^{(i)}} = \sum_t u_k^{(i)}(t) X_t X_t^T$$

denote the expected sufficient statistics given $\mathbf{X}$ and current estimate $\Theta^{(i)}$ with state-responsibilities $\mathrm{u}_k^{(i)}(t) = \mathrm{P}_{\Theta^{(i)}}(S_t = k|\mathbf{X})$ obtained from the E-Step.

By $\pi_k^{(i)} = \frac{1}{n}\sum_t u_k^{(i)}(t)$ we denote the (scaled) effective sample size of state k. The penalty term depends on a regularization parameter λ, on the effective sample size $\pi_k^{(i)}$ and on a function $\text{Pen}(\cdot)$ involving ℓ_1-norm of Ω_k. The reason why we incorporate the square root of the effective sample size is that it is known from the Lasso literature that the ℓ_1-penalty term asymptotically has to grow with square root of the sample size in order to achieve optimality (Bühlmann and van de Geer, 2011). We consider three slightly different functions $\text{Pen}(\cdot)$ defined as follows:

- $\text{Pen}_{\text{invcov}}(\Omega) = \|\Omega^-\|_1$, the classical penalty known from the Graphical Lasso. It imposes ℓ_1-constraints on the non-diagonal entries of the concentration matrix Ω.
- $\text{Pen}_{\text{parcor}}(\Omega) = \|\Psi^-\|_1$, where Ψ is the partial correlation matrix which can be written as $(\Psi)_{ll'} = \Omega_{ll'}/\sqrt{\Omega_{ll}\Omega_{l'l'}}$.
- $\text{Pen}_{\text{invcor}}(\Omega) = \|\Phi^-\|_1$, where Φ is the inverse of the correlation matrix given by $\Phi = \text{C}^{-1}$, $\text{C}_{ll'} = \Sigma_{ll'}/\sqrt{\Sigma_{ll}\Sigma_{l'l'}}$.

Note that all three functions penalize the ℓ_1-norm of the concentration matrix and therefore lead to sparse Ω's. The advantage of $\text{Pen}_{\text{parcor}}(\cdot)$ and $\text{Pen}_{\text{invcor}}(\cdot)$ is that they are scale-invariant and therefore remove concerns that arise from state-specific scaling. As we noted above, state-specific scaling cannot be removed by pre-processing in the HMM setting since state assignments are themselves unknown at the outset.

Optimization of (2.4) is non-standard. It is easy to verify that (2.4) reduces to:

$$\mu_k^{(i+1)} = \mathbf{T}_1^{u_k^{(i)}}/n, \quad \Omega_k^{(i+1)} = \underset{\mu_k,\Omega_k}{\text{argmin}} - \log\|\Omega_k\| + \text{tr}(\Omega_k \mathbf{C}^{\mathbf{u}_k^{(i)}}) + 2\frac{\lambda}{n_k}\sqrt{\pi_k^{(i)}}\text{Pen}(\Omega_k) \quad (2.5)$$

where $\mathbf{C}^{\mathbf{u}_k^{(i)}} = \frac{1}{n}\mathbf{T}_2^{u_k^{(i)}} - \mu_k^{(i+1)}(\mu_k^{(i+1)})^T$. For the penalty function $\text{Pen}_{\text{invcov}}(\cdot)$ optimization problem (2.5) can be solved by the Graphical Lasso algorithm presented in Friedman *et al.* (2008). In the Appendix we compare these three different penalties and discuss how we perform optimization.

Algorithm 1 summarizes HMMGLasso. As stated above the EM algorithm depends on initial specification of parameters, i.e., $\theta_k^{(0)}, \Pi^{(0)}$ $(k = 1, \ldots, K)$. For convenience (see later in text) we directly specify $\text{u}_k^{(0)}(t)$ (instead of $\theta_k^{(0)}$) and start with an M-Step followed by an E-Step. We stop the algorithm if the relative change in the Σ_k's falls below a threshold ϵ or if for at least one state the scaled effective sample size π_k is smaller than $\pi_{\min}$.

2.2 Universal regularization

In this Section we discuss the choice of the regularization parameter λ in HMMGLasso. We will argue that $\lambda_{\text{uni}} = \sqrt{2n\log p}/2$ is a reasonable regularization parameter for HMMGLasso. We do this by considering connections with the Lasso (Tibshirani, 1996) and the Graphical Lasso (or GLasso; Friedman *et al.*, 2008). In the classical Lasso or GLasso setup the regularization parameter is usually chosen empirically to minimize the prediction error (for example by performing cross-validation). However, in the much more complicated HMM (or more generally latent variable) setting, with unknown number of states K, such

Algorithm 1 HMMGLasso

Input K, λ, $\Upsilon^{(0)} = \{(\mathrm{u}_k^{(0)}(t))_{k=1..K,t\in\mathcal{T}}, \Pi^{(0)}, \pi^{(0)}\}$ and set $i = 0, \mathrm{err}^{(0)} = 0$.

1: **while** $\{\mathrm{err}^{(i)} < \epsilon\} \vee \{\pi_k^{(i)} > \pi_{\min}$ for all $k = 1 \dots K\}$ **do**

2: **M-Step** Obtain estimates

$$(\mu_k^{(i+1)}, \Omega_k^{(i+1)}) = \underset{\mu_k,\Omega_k}{\mathrm{argmin}} -\ell(\mu_k, \Omega_k; \mathbf{T}_1^{\mathbf{u_k^{(i)}}}, \mathbf{T}_2^{\mathbf{u_k^{(i)}}}) + \lambda\sqrt{\pi_k^{(i)}}\mathrm{Pen}(\Omega_k)$$

$$\Pi_{kk'}^{(i+1)} = \mathbf{v}_{kk'}^{(i)}/\pi_k^{(i)} \ (\Pi_{kk'}^{(1)} = \Pi_{kk'}^{(0)} \text{ in 1st iteration})$$

3: **E-Step** Use Forward-Backward equations to update

$$\mathrm{u}_k^{(i+1)}(t) = \mathrm{P}_{\Theta^{(i+1)}}(S_t = k|\mathbf{X})$$

$$\mathrm{v}_{kk'}^{(i+1)}(t) = \mathrm{P}_{\Theta^{(i+1)}}(S_t = k, S_{t+1} = k'|\mathbf{X})$$

$$\pi_k^{(i+1)} = \textstyle\sum_t \mathrm{u}_k^{(i+1)}(t)/n$$

4: **Set** $\mathrm{err}^{(i+1)} = \max\limits_{k,l,l'} \left\{ \frac{|\Sigma_{k,ll'}^{(i+1)} - \Sigma_{k,ll'}^{(i)}|}{1+|\Sigma_{k,ll'}^{(i+1)}|} \right\}$ and $i \leftarrow i+1$

5: **end while**

Output $\hat{\Xi}^{(K,\lambda)} = \{\hat{\Theta}_{K,\lambda}, (\hat{\mathrm{u}}_k(t))_{k=1..K,t\in\mathcal{T}}, \hat{\pi}\}$

a brute force strategy is computationally burdensome, motivating the need for universal regularization.

First, consider a classical regression setup with $y = \mathbf{X}\beta + \epsilon$, where $\epsilon \sim \mathcal{N}(0, \sigma^2\mathrm{I})$. Here, $\mathbf{X}$ is a $N \times p$ predictor matrix, y a $N \times 1$ response vector, β denotes the $p \times 1$ regression parameter and σ^2 is the error variance. Then, the Lasso estimator minimizes $\|y - \mathbf{X}\beta\|^2/2N + s\|\beta\|_1$. Assuming an orthonormal predictor matrix, Donoho and Johnstone (1994) showed that the risk of the Lasso estimator comes close to the oracle risk if we use $s_{\mathrm{uni}} = \sigma\sqrt{2\log p/N}$ as a regularization parameter. Universal regularization and the penalty $\sigma\sqrt{2\log p/N}$ is discussed also in the non-orthonormal case in Zhang (2010) or Sun and Zhang (2011) (see also Barron *et al.* (2008); they propose a universal penalty parameter based on the minimum description length principle). It is important to note that s_{uni} decreases with $1/\sqrt{N}$. This is the reason why we include the square-root of the effective sample size into the state-specific penalty terms in the HMMGLasso (see Section 2.1).

Next, consider the Graphical Lasso,

$$\hat{\Omega} = \underset{\Omega}{\mathrm{argmin}} - \log|\Omega| - \mathrm{tr}(\mathbf{S}\Omega) + \rho\|\Omega^-\|,$$

where $\mathbf{S}$ is the sample covariance matrix of $X = (X^{(1)}, \dots, X^{(p)}) \sim \mathcal{N}(0, \Sigma)$ with $\Omega = \Sigma^{-1}$. Friedman *et al.* (2008) showed that the last row/column of $\hat{\Omega}$ can be obtained by solving

$$\hat{\beta} \quad = \quad \underset{\beta}{\mathrm{argmin}}\, 0.5\beta\Sigma_{11}\beta + \beta\mathrm{s}_{12} + \rho\|\beta\|_1, \tag{2.6}$$

where β and Ω are linked through $\sigma_{12} = \Sigma_{11}\beta/2$ (Σ_{11} is the covariance matrix with the last row and column deleted; σ_{12} and s_{12} denote the last row of the covariance and

sample covariance matrix). Note that (2.6) can be interpreted as the Lasso estimator corresponding to regression of variable $X^{(p)}$ against $X^{(1)}, \ldots, X^{(p-1)}$. As Ω_{pp} is the error variance in regressing $X^{(p)}$ against $X^{(1)}, \ldots, X^{(p-1)}$, we can identify $\Omega_{pp}^{-1/2}\sqrt{2\log p/N}$ as a good choice for ρ in (2.6). If Ω is standardized to have unit diagonal entries, then we can write $\rho_{\text{uni}} = \sqrt{2\log p/N}$.

Now consider equation (2.5) of the HMMGLasso with $\text{Pen}_{\text{invcov}}(\cdot)$ and assume all Ω_k's are standardized to have unit diagonal. Equating $2\frac{\lambda}{n_k}\sqrt{\pi_k^{(i)}}$ with $\rho_{\text{uni}} = \sqrt{2\log p/n_k}$ (the universal shrinkage level in the Graphical Lasso with sample size $N = n_k$) and solving for λ we obtain

$$\lambda_{\text{uni}} = \sqrt{2n\log p}/2.$$

For the penalty function $\text{Pen}_{\text{invcov}}(\cdot)$ the foregoing indicates that $\lambda_{\text{uni}} = \sqrt{2n\log p}/2$ only holds if the Ω_k's are standardized and therefore equal the corresponding partial correlation matrix. In general, since state assignments are themselves unknown, this standardization cannot be done as a pre-processing step. However, if we use $\text{Pen}_{\text{parcor}}(\cdot)$ instead, $\lambda_{\text{uni}} = \sqrt{2n\log p}/2$ applies regardless of scaling. Penalizing the partial correlation can be seen as a generalization of the "scaled" Lasso proposed by Städler *et al.* (2010). There, the negative log-likelihood is penalized by $s\frac{\|\beta\|}{\sigma}$ and optimization is performed over β and σ simultaneously. A reasonable choice for s is $\sqrt{2\log p/N}$, which does not depend anymore on the unknown noise level (see Sun and Zhang (2011) and also the discussion in Städler *et al.* (2010)).

Thus, λ_{uni} is the penalty level we use for estimation in HMMGLasso. It is "universal" in the sense that it only depends on the dimensionality of the input data n and p. Furthermore, when λ_{uni} is used with the penalty $\text{Pen}_{\text{parcor}}(\cdot)$ the penalization self-adapts to the hidden states by incorporating the square-root of the effective sample size and by taking care of scaling.

2.3 Model order exploration using Greedy Backward Pruning

Greedy Backward Pruning can in principle be used with a wide range of model selection criteria; here we consider the popular Bayesian Information Criterion (BIC) and the Mixture Minimum Description Length (MMDL). MMDL was introduced by Figueiredo *et al.* (1999) and was specifically proposed for the purpose of determining the number of components in finite mixtures. We first describe these criteria and then go on to give a detailed description of the Greedy Backward Pruning algorithm.

Model selection criteria. A model selection criterion $\mathcal{C}$ has to trade off goodness-of-fit and model complexity. BIC and MMDL are defined by

$$\text{BIC}(\hat{\Theta}_{K,\lambda}) = -\ell(\hat{\Theta}_{K,\lambda}; \mathbf{X}) + \frac{1}{2}\log(n)K(K-1) + \frac{1}{2}\log(n)\sum_k \text{Df}(k,\lambda)$$

$$\text{MMDL}(\hat{\Theta}_{K,\lambda}) = -\ell(\hat{\Theta}_{K,\lambda}; \mathbf{X}) + \frac{1}{2}\log(n)K(K-1) + \sum_k \frac{1}{2}\log(n\hat{\pi}_k)\text{Df}(k,\lambda),$$

where in the context of ℓ_1 penalized log-likelihood we set the degrees of freedom as $\mathrm{Df}(k,\lambda) = p + \sum_{l'>l} 1_{(\hat{\Omega}_{k,\lambda})_{ll'}\neq 0}$.

MMDL can be motivated by the minimum description length principle (Grünwald, 2007). The negative log-likelihood represents the optimal code-length of the data given model parameters Θ. The term $\frac{1}{2}\log(n)K(K-1)$ is the "optimal" code-length for the transition matrix Π (note that Π is estimated from all data). As $n\pi_k$ is the effective sample size from which $\theta_k = (\mu_k, \Omega_k)$ is estimated we get $\frac{1}{2}\log(n\pi_k)\mathrm{Df}(k,\lambda)$ as an "optimal" code-length for describing θ_k.

The main difference between BIC and MMDL is the use of the *effective sample size* $n\hat{\pi}_k$ in the code-lengths for parameters which are state-specific. Figueiredo *et al.* (1999) argued using ideas from minimum description length literature that MMDL is more appropriate for mixtures than BIC. They demonstrate on real and synthetic data that MMDL outperforms BIC. In Section 3 we compare performance of Greedy Backward Pruning using BIC and MMDL as model selection criteria. In our more involved inference task we come to the same conclusion as Figueiredo *et al.* (1999), namely that MMDL outperforms BIC.

Greedy Backward Pruning in detail. Greedy Backward Pruning works by first estimating parameters using HMMGLasso with a large number of states $K_{\max}$ and then iteratively reducing the number of states until some minimal number of states $K_{\min}$ is reached. Each iteration involves either merging the two "closest" states *or* deleting the "smallest" state, and then re-running HMMGLasso with one fewer state, using estimates from the previous step as initialization. This scheme is summarized in Algorithm 2.

We give now a definition of "smallest" state and "closest" states and describe the "merge" and "delete" operation in detail. Let $\hat{\Theta}_K$ be the current estimate for K states. The merge operation consists of detecting the two closest states k_1 and k_2 defined as

$$(k_1, k_2) = \operatorname*{argmin}_{k,k'\in\{1,\ldots,K\}} \mathcal{D}_s(\hat{\theta}_k||\hat{\theta}_{k'}),$$

where $\mathcal{D}_s(\hat{\theta}_k||\hat{\theta}_{k'})$ is the symmetric Kullback-Leibler divergence given by

$$\mathcal{D}_s(\hat{\theta}_k||\hat{\theta}_{k'}) = \mathrm{tr}\{(\Sigma_k - \Sigma_{k'})(\Sigma_{k'}^{-1} - \Sigma_k^{-1})\} + (\mu_k - \mu_{k'})^T(\Sigma_k^{-1} - \Sigma_{k'}^{-1})(\mu_k - \mu_{k'}).$$

We merge states k_1 and k_2 into a new state (denoted by $k_1 \cup k_2$) by forming new initial conditions for the next run of HMMGLasso with $K-1$ states. In particular we compute merged responsibilities as

$$\begin{aligned} \mathrm{u}_{\mathbf{mer}\ k_1\cup k_2}(t) &= \hat{\mathrm{u}}_{k_1}(t) + \hat{\mathrm{u}}_{k_2}(t) \\ \mathrm{u}_{\mathbf{mer}\ k}(t) &= \hat{\mathrm{u}}_k(t) \quad (\text{for } k \neq k_1 \cup k_2) \end{aligned}$$

and get a merged transition matrix using updates

$$\begin{aligned} \Pi_{\mathbf{mer}\ \mathrm{k_1\cup k_2,k'}} &= \hat{\Pi}_{k_1k'} + \hat{\Pi}_{k_2k'} \quad (\text{for } k' \neq k_1 \cup k_2) \\ \Pi_{\mathbf{mer}\ \mathrm{k,k'}} &= \hat{\Pi}_{k,k'} \quad (\text{for } k', k \neq k_1 \cup k_2) \\ \Pi_{\mathbf{mer}\ k',k_1\cup k_2} &= 1/(K-1) \quad (\text{for } k' = 1,\ldots,K-1). \end{aligned}$$

All these operations are based on the relation: $\mathrm{P}(S_t = k_1 \cup S_t = k_2|\cdot) = \mathrm{P}(S_t = k_1|\cdot) + \mathrm{P}(S_t = k_2|\cdot)$.

The delete operation simply discards the smallest state according to $\min_{k\in\{1,\ldots,K\}} \hat{\pi}_k$. Initial conditions $\mathrm{u}_{\mathbf{del}}, \Pi_{\mathbf{del}}$ arising from deleting a state are derived by omitting corresponding row/column of $\hat{\mathrm{u}}, \hat{\Pi}$ and renormalizing these quantities such that rows sum up to one.

Note that Greedy Backward Pruning algorithm needs to be initialized only once, namely at $K_{\max}$. Further, we note from Algorithm 2 that we decide between the "merging" and "deleting" operations based on the model selection criterion, i.e., if initial conditions obtained from merging leads to estimate with smaller criterion $\mathcal{C}$ we choose that solution otherwise we take the solution obtained from the "delete" operation. As demonstrated in examples below, Greedy Backward Pruning with only a single initialization at large $K_{\max}$ yields remarkably good estimates in the unknown K case. Our procedure originates from the algorithms proposed in Figueiredo *et al.* (1999), Figueiredo and Jain (2000) and Bicego *et al.* (2003). Our empirical results below echo the findings of these authors that Greedy Backward Pruning-like approaches can confer robustness to initialization.

Algorithm 2 Greedy Backward Pruning with HMMGLasso

Input $K_{\min}$ and $K_{\max}$.

Initialization of $\Upsilon^{(K_{\max})} = \{(\mathrm{u}_k(t))_{k=1..K_{\max}, t\in\mathcal{T}}, \Pi, \pi\}$.

1: Fit **HMMGLasso** and obtain: $\hat{\Xi}^{(K_{\max},\lambda_{\mathrm{uni}})} \leftarrow \mathbf{HMMGLasso}(K_{\max}, \lambda_{\mathrm{uni}}, \Upsilon^{(K_{\max})})$.

2: Set $\kappa = K_{\max}$.

3: **while** $\kappa \geq K_{\min}$ **do**

4: **Merge *Or* Delete**
Compute merged/deleted initial conditions: $\Upsilon_{\mathbf{mer}}$ and $\Upsilon_{\mathbf{del}}$.
Compute $\Xi_{\mathbf{mer}} \leftarrow \mathbf{HMMGLasso}(\kappa - 1, \lambda_{\mathrm{uni}}, \Upsilon_{\mathbf{mer}})$
Compute $\Xi_{\mathbf{del}} \leftarrow \mathbf{HMMGLasso}(\kappa - 1, \lambda_{\mathrm{uni}}, \Upsilon_{\mathbf{del}})$.

5: **Update:**
Set $\kappa \leftarrow \kappa - 1$.
Set $\Xi^{(\kappa,\lambda_{\mathrm{uni}})} \leftarrow \Xi_{\mathbf{mer}}$ *if* $\mathcal{C}(\Theta_{\mathbf{mer}}) < \mathcal{C}(\Theta_{\mathbf{del}})$.
Set $\Xi^{(\kappa,\lambda_{\mathrm{uni}})} \leftarrow \Xi_{\mathbf{del}}$ *if* $\mathcal{C}(\Theta_{\mathbf{del}}) < \mathcal{C}(\Theta_{\mathbf{mer}})$.

6: **end while**

7: **Set:** $\hat{K}_{\mathrm{opt}} = \underset{\kappa}{\operatorname{argmin}}\, \mathcal{C}(\Theta_{\kappa,\lambda_{\mathrm{uni}}})$.

Output final estimates: $\hat{\Theta}_{K_{\mathrm{opt}},\lambda_{\mathrm{uni}}}$.

3 Examples

3.1 Simulation studies

In this Section we describe data-generating models that we use for simulation examples. We consider the following data-generating models:

Model 1 $K_{\text{true}} \in \{2, 4, 6\}, n = 2000, p = 10$, (n/p-ratio=200).

Transition matrix. $\Pi_{kk'} = 0.1\gamma$ and $\Pi_{kk} = 0.9\gamma$, where $k, k' \in \{1, \ldots, K_{\text{true}}\}$ and γ is chosen such that $\sum_{k'=1}^{K_{\text{true}}} \Pi_{kk'} = 1$.

Means $\mu_k,\ k = 1, \ldots, K_{\text{true}}$. Each state has p/K_{true} nonzero entries with value $(-1)^k \alpha/\sqrt{p/K_{\text{true}}}$. Nonzero's are at different locations for each state.

Concentration matrix $\Omega_k,\ k = 1, \ldots, K_{\text{true}}$. Each state has p nonzero (off-diagonal) entries. To reflect the setting in which states share some aspects of graphical model structure, $p/2$ non-zeros are shared between all states, whereas the other $p/2$ non-zeros are at different locations for each state. Concentration matrices are generated as in Rothman *et al.* (2008) but standardized to have unit diagonal entries.

Model 2 As model 1 but with $p = 75$, (n/p-ratio=26 2/3)

Model 3 As model 1 but with $n = 1000$ and $p = 100$, (n/p-ratio=10)

Model 4 $K_{\text{true}} \in \{2, 4, 6\}, n = 5000, p = 50$.

Transition matrix. $\Pi_{kk'} = 0.1\gamma$, $\Pi_{kk} = 0.9\gamma$ for $k \neq K_{\text{true}}$; $\Pi_{K_{\text{true}},k'} = 1/K_{\text{true}}$ ($k' \in \{1, \ldots, K_{\text{true}}\}$). Again, γ is chosen such that rows sum up to one.

Means. $(\mu_k)_l = \alpha$ for $l \in \{1, 2\}$ and $k \in \{1, 2\}$. All other entries equal zero.

Concentration matrix. For $k = 1, 2$: $\Omega_k = \mathrm{I}_{\mathrm{p}}$. For $k = 3, \ldots, K_{\text{true}}$: Ω_k has two nonzero entries, at different locations for each state. Concentration matrices are standardized to have unit diagonal entries.

Ideally we seek methodology that can automatically adapt to both low- and high-dimensional settings. Accordingly, Models 1, 2 and 3 have the same design but differ with respect to n/p-ratio. We include the small p, large n Model 1 as a baseline and to investigate the performance of universal regularization in the classical low-dimensional setting. Model 4 is a challenging problem, similar in terms of n, p to the real, genomic data example below.

Experiment I: Number of States. In this Experiment the focus is on state recovery. We explore the ability to estimate the correct number of states K and recover the state assignments. We compare the following methods

- HMMGLasso initialized by Kmeans (**Hmmgl**)
- HMMGLasso with *Greedy Backward Pruning* (**Bwprun**)
- Unpenalized maximum likelihood estimation (MLE) (**Unpen**)
- MLE with diagonal restricted covariance matrices (**Diagcov**)
- Model-based clustering via Gaussian mixture models (**Mclust**; Fraley and Raftery, 2006)

Thus, *Hmmgl* and *Bwprun* are the methods we propose. Both Hmmgl and Bwprun carry out estimation (for given K) using the penalty and universal regularization via λ_{uni} that

we put forward above; the former embeds our estimator within a standard, "brute-force" exploration of K, whilst the latter uses Greedy Backward Pruning.

In all numerical experiments we stop the algorithms according to the rule described in Algorithm 1 with $\epsilon = 10^{-3}$ and $\pi_{\min} = 5/n$ (for Unpen we use $\pi_{\min} = p/n$ to ensure non-singular covariance estimates). For each method we use each of BIC and MMDL as model selection criteria. For Hmmgl, Unpen and Diagcov we compute estimates for $K = 1, \ldots, K_{\text{true}} + 2$ and pick the number of states minimizing BIC or MMDL. As a reference, we also cluster the data using the **R**-package `mclust` (Fraley and Raftery, 2006). We use the function `Mclust`; this employs Gaussian mixture models and uses BIC to automatically select between different covariance structures and numbers of clusters (we allow $K = 1, \ldots, K_{\text{true}}+2$). We initialize `Mclust` using model based hierarchical clustering with equal spherical covariances (we note that the default initialization of `Mclust`, using hierarchical clustering with unconstrained covariances, performs worse in the examples below). For more details see Fraley and Raftery (2002). Specifications of all the methods are summarized in Table 1.

Method	Selection Criterion $\mathcal{C}$	Regularization/Constraints	Initialization
Bwprun	BIC/MMDL	($\text{Pen}_{\text{parcor}}$, λ_{uni})	KM (100 r.s.) at $K_{\max} = 15$
Hmmgl	BIC/MMDL	($\text{Pen}_{\text{parcor}}$, λ_{uni})	KM (100 r.s.)
Unpen	BIC/MMDL	No constraints	KM (100 r.s.)
Diagcov	BIC/MMDL	Diagonal covariances	KM (100 r.s.)
Mclust	BIC	Various covariance structures (see Fraley and Raftery (2002))	Hierarchical clustering

Table 1: Methods used in simulation Experiment I. [r.s. stands for random starts]

We generated 50 datasets from each of Models 1-4 with $\alpha = 2$ and report for all methods number of selected states and adjusted Rand index (this quantifies the extent to which estimated state assignments agree with true state membership). The results for Models 3 and 4 are summarized in Figures 1 and 2; Figures 7 and 8 at the end of this manuscript show results for Models 1 and 2.

In nearly all settings Diagcov is unable to recover the correct number of states and performs poorly in terms of adjusted Rand index. This is not surprising as Diagcov imposes incorrect model assumptions. Only in Model 3 with $K_{\text{true}} = 2$, where for both states the data generating covariance matrices are diagonal, does Diagcov perform well. MLE without penalization (Unpen) does well only in the low-dimensional Model 1. Both the proposed methods (Hmmgl and Bwprun) greatly outperform the other methods in Models 2-4. This supports the notion that regularization can be useful even when sample size n is seemingly large.

HMMGLasso also works well in Model 1 with large n and very small p, a scenario where no constraints are necessary. This demonstrates that the adaptive strategy and universal regularization can be applied without any hand tuning also in the low-dimensional setting. We also read-off from Figures 1-2 (see especially scenarios with $K = 6$) the substantial improvement of Greedy Backward Pruning relative to HMMGLasso, despite the fact that the latter carries out essentially a brute-force search over K. Also the use of MMDL

further improves performance (it never performs worse than BIC). Especially in tough and very high-dimensional scenarios (Model 3 and Model 4 with K=6) MMDL seems to perform better.

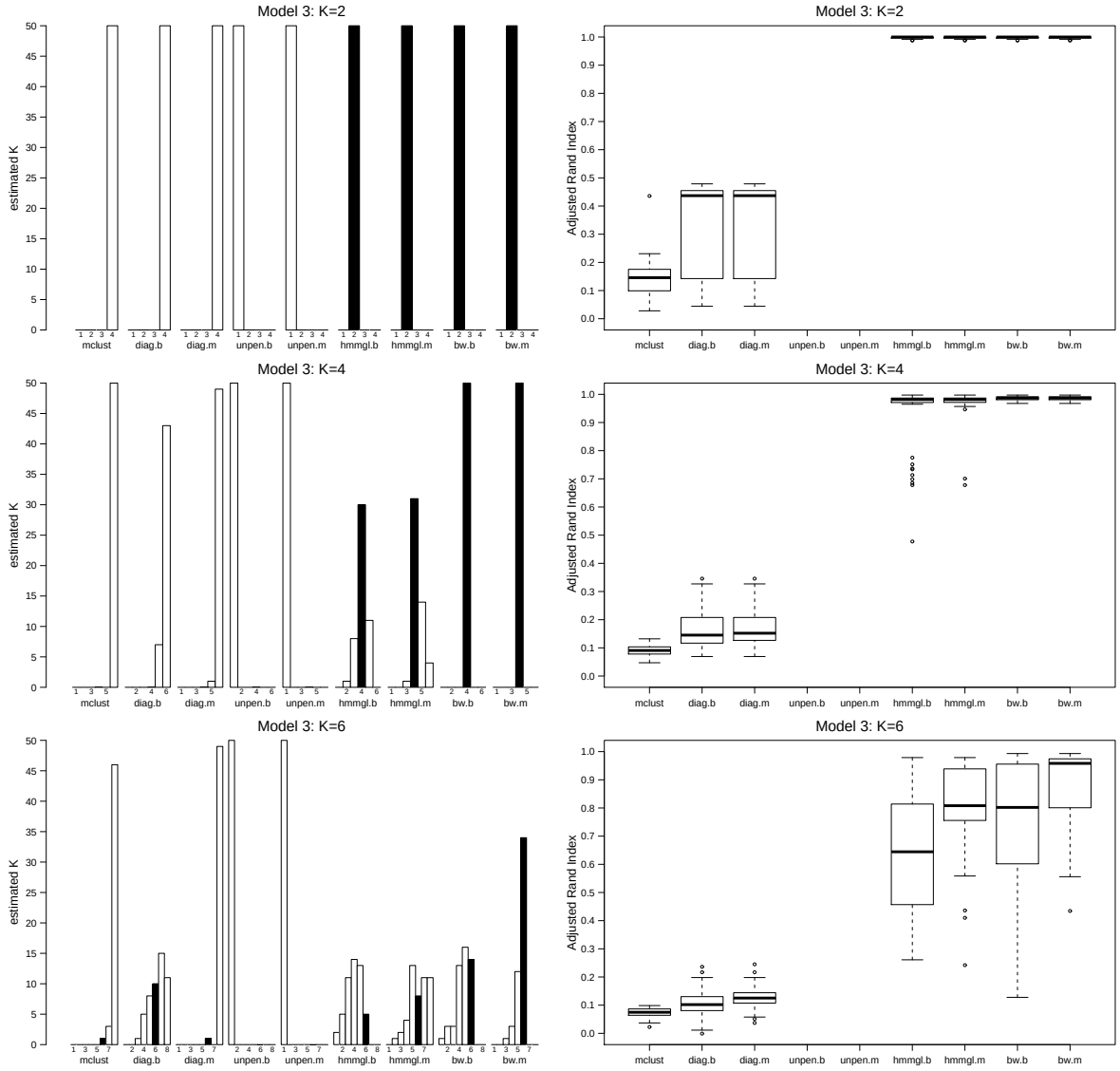

Figure 1: Simulation Model 3 ($p = 100, n = 1000$), number of states and state assignments. Left panels: frequency of estimated number of states; in each case the correct number of states (i.e. number of states in data-generating model) is indicated in black. Right panels: adjusted Rand index with respect to true state assignments. [Legend: Results for Mclust (**mclust**), MLE with diagonal covariance matrices (**diag**), MLE (**unpen**) and Greedy Backward Pruning (**bw**) are shown. The extensions ".b" and ".m" stand for BIC and MMDL respectively.]

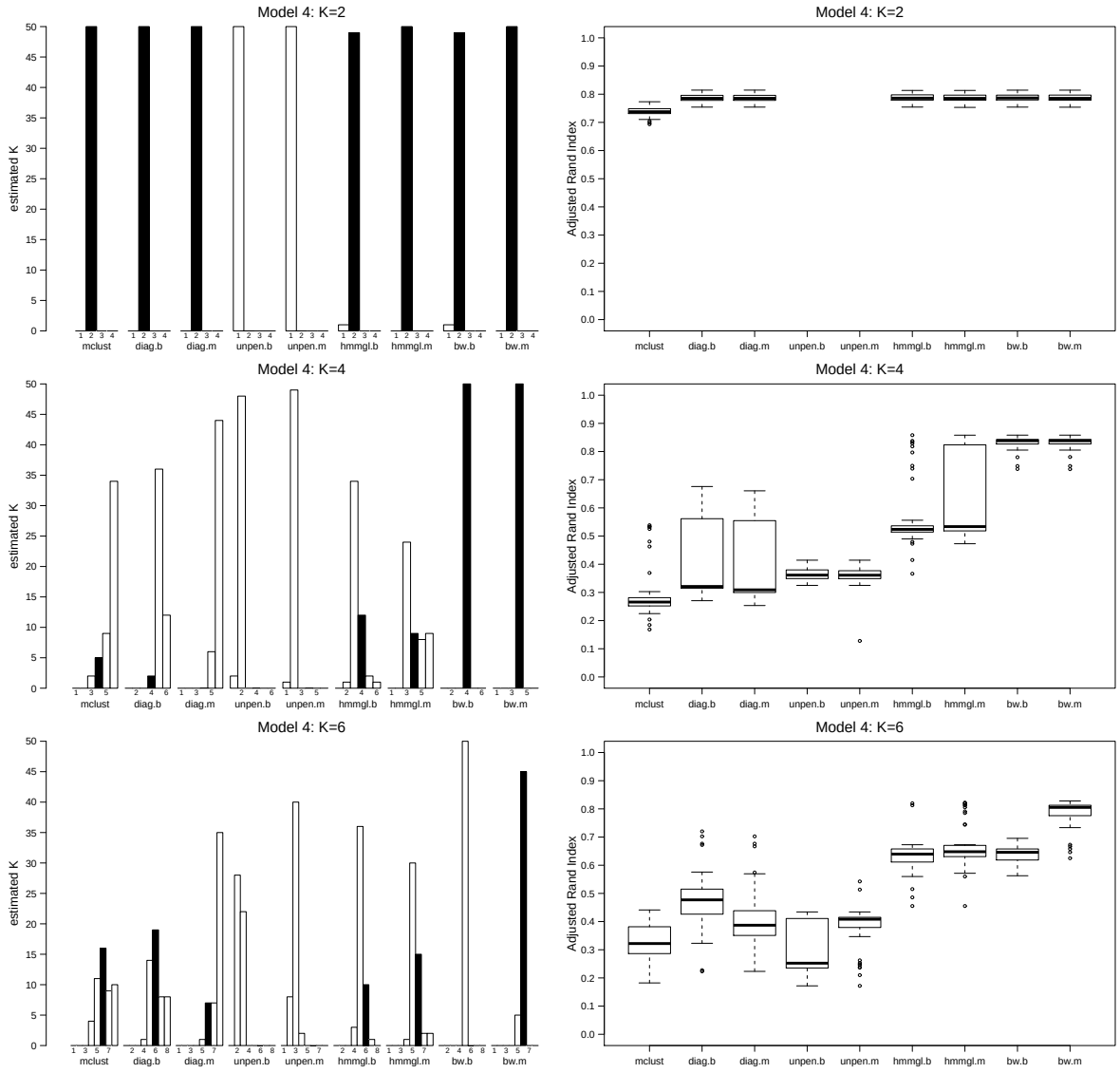

Figure 2: Simulation Model 4 ($p = 50, n = 5000$), number of states and state assignments. Left panels: frequency of estimated number of states (as in Fig 1 correct number of states indicated in black). Right panels: adjusted Rand index with respect to true state assignments. [Legend: Results for Mclust (**mclust**), MLE with diagonal covariance matrices (**diag**), MLE (**unpen**) and Greedy Backward Pruning (**bw**) are shown. Extensions ".b" and ".m" stand for BIC and MMDL respectively.]

Experiment II: Graph Structure. In this experiment we focus on recovering state-specific graphical model structure. We consider Model 3 with $K_{\text{true}} \in \{2, 4, 6\}$ and $\alpha \in \{2, 6, 10\}$. We compare Greedy Backward Pruning; HMMGLasso ($K = K_{\text{opt}}, \text{Pen}_{\text{parcor}}, \lambda_{\text{uni}}$); Kmeans (with number of clusters set to $K = K_{\text{true}}$) followed by estimating cluster-specific inverse covariance matrices using Graphical Lasso; and Graphical Lasso using all samples (no state assignment or clustering). In Figure 3 True Positive Rate (TPR; with respect to edges in the data-generating graph) is plotted against the corresponding False Positive Rate (FPR) for all combinations of K and α and different methods. We note that Greedy Backward Pruning consistently selects the correct number of states in all scenarios except in $(K_{\text{true}}, \alpha) = (6, 2)$ where it chooses K correctly in 36 out of 50 datasets.

Greedy Backward Pruning performs well in terms of TPR and FPR. It is noteworthy that universal regularization using λ_{uni} gives consistently good results under a range of conditions. We see that HMMGLasso exhibits a smaller true positive rate in the most challenging $K_{\text{true}} = 6$ case. For $\alpha = 2$ Kmeans in combination with GLasso performs much worse in particular in terms of TPR. For larger α's (and therefore with increased information about state-assignment in the means) TPR and FPR of Kmeans improves. Finally, GLasso applied to all data without any clustering leads to very poor performance (this is likely a consequence of Simpson's paradox).

3.2 Application to genomic data

We consider genome-wide binding data for 53 proteins in the *Drosophila* cell line Kc167 (data from Filion *et al.*, 2010). Filion *et al.* (2010) represents an important step forward in the genome biology of *Drosophila*, showing for the first time how multivariate data can reveal protein-DNA binding patterns that depend on genome region. Here, we use this dataset to test our HMM methodology. The dataset offers a number of advantages for our purposes. First, the coverage of a relatively large number of proteins ($p = 53$) in the full data gives a high-dimensional example from current genome biology. Second the abundance of data ($n = 33,632$ for chromosome 2L and $n = 32,791$ for chromsome 2R) allows fully held-out validation on a large test set (we use the latter half of chromsome 2R, giving $n_{\text{test}} = 16,396$) as well as exploration of the effect of (training) sample size. Finally, although substantive biological questions are beyond the scope of this paper, several open questions concerning genome organization in *Drosophila*, including the likely number of genome regions, and the possibility of region-specific protein-protein interplay, help to motivate the methodological questions we address here.

Filion *et al.* (2010) identified regions of the genome by fitting a HMM (using classical, unpenalized estimation) to reduced-dimension data. Dimensionality reduction was carried out using principal component analysis (PCA) as a pre-processing step, with the HMM fitted to the first three principal components. Such approaches are currently widely used in genome biology. By looking at principal components, Filion *et al.* (2010) suggested a model with five states (corresponding to different chromatin types). They further noted that these five states are marked by enriched binding of the proteins *HP1, PC, H1, BRM* and *MRG15* and that a 5-state HMM using only the five marker proteins as an input recapitulates 85.5% of the original state classification.

We investigated performance in a held-out predictive sense by training on the first $n_{\text{train}} = 500, 1000, 2000, \ldots, 5000$ observations of chromosome 2L and then reporting the test log-likelihood obtained from the second half of chromosome 2R ($n_{\text{test}} = 16,396$). As above we compare HMMGLasso (**Hmmgl**); Greedy Backward Pruning (**Bwprun**); unpenalized MLE (**Unpen**); and MLE with diagonal covariance matrices (**Diagcov**). Additionally, we include a five-state MLE using only the five marker proteins reported by Filion *et al.*

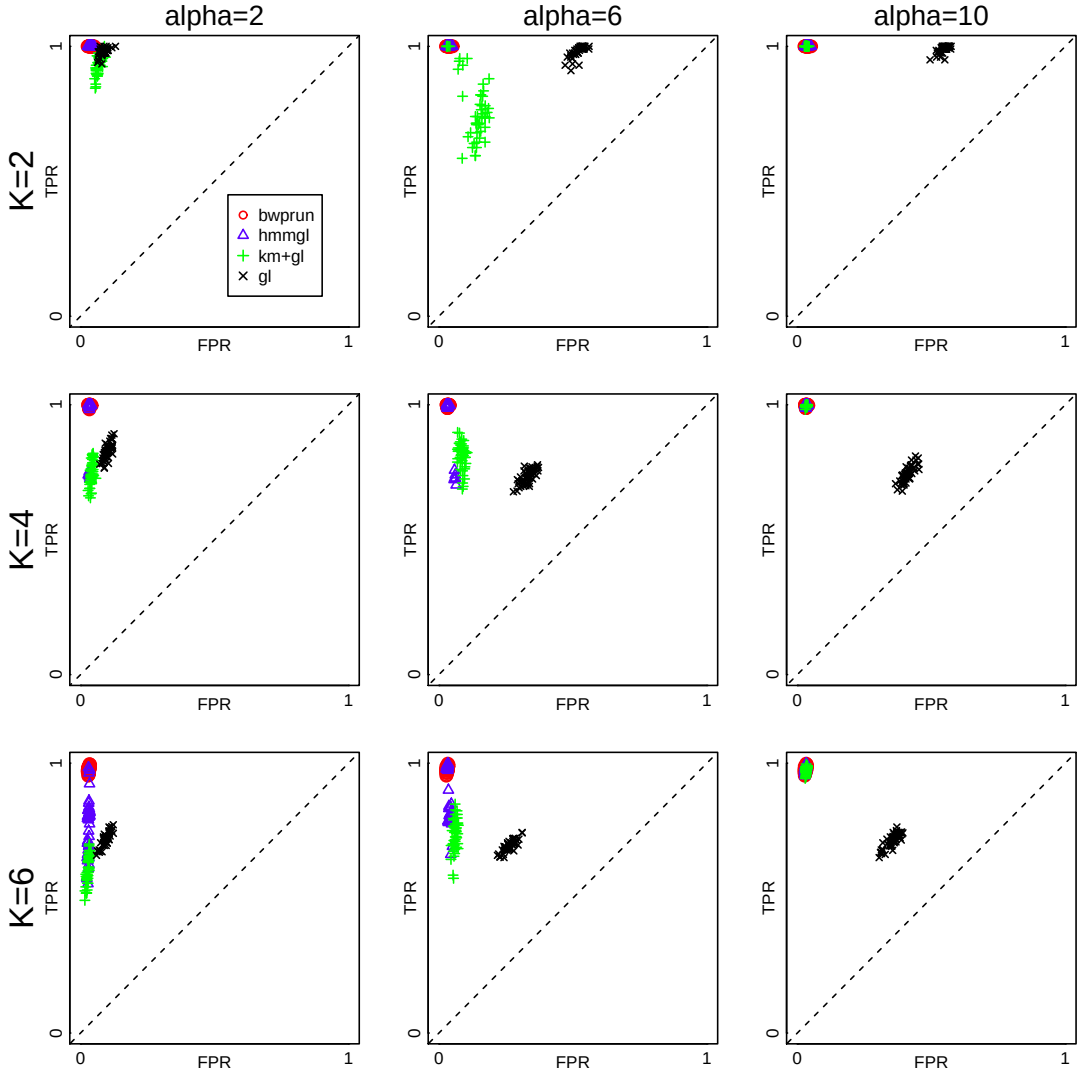

Figure 3: Simulation experiment II, graphical model estimation. Comparing estimated state-specific conditional independence graphs against the data-generating graphs gave true positive and false positive rates with respect to edges in the graphs (TPR and FPR respectively). We show TPR plotted against FPR with $K \in \{2, 4, 6\}$, $\alpha \in \{2, 6, 10\}$ for Model 3.[Legend: Results for Greedy Backward Pruning (**bwprun**), HMMGLasso (**hmmgl**), Kmeans clustering with cluster-wise Graphical Lasso (**km+glasso**) and Graphical Lasso applied to non-clustered data (**glasso**) are shown]

(2010) (**Marker**). For Hmmgl, Unpen and Diagcov the number of states is determined by exploring different K's in a forward stepwise manner. We use MMDL and BIC as model selection criteria. All methods are initialized by Kmeans with initial centroids obtained using hierarchical clustering; this renders the overall analysis deterministic by removing variability due to random initialization of Kmeans.

Figure 4 shows the MMDL(BIC)-scores (scaled by n_{train}) and the negative test log-likelihood as a function of n_{train}. Figure 5 depicts the selected number of states for each method and training sample size. Overall, we notice that MMDL (BIC) and test log-likelihood show similar patterns for different methods and different sample sizes. Bwprun and Hmmgl greatly outperform Marker and Diagcov. This provides a topical example where a multivariate view (using all variables and modelling also state-specific covariances) improves out-of-sample predictive performance. The predictive gain of penalization compared to unpenalized MLE for moderate n/p-ratios is also noteworthy. As expected, the performance of Unpen in terms of MMDL (BIC) and test log-likelihood approaches the penalized methods with increasing sample size. However, in terms of number of states (Figure 5) the estimates are very different even for large n_{train}, i.e., penalization typically leads to more states than unpenalized MLE. This illustrates that the prediction-optimal number of states depends on the estimation procedure employed: regularization allows estimation for a greater number of states. If state-specific estimates have scientific relevance, this property can be important, since due to Simpson's paradox, estimates for finer state distinctions (larger K) cannot, in general, be recovered from coarser models (smaller K). We return to the question of exploration of number of states in Discussion below.

We note that for each training sample size n_{train} the results shown in Figures 4-5 reflect performance for a single training sample of the specified length. For completeness, Figure 9 in the Appendix shows performance over 9 different training datasets of size $n_{\text{train}} = 1000$.

4 Discussion

We considered penalized estimation in multivariate HMMs, including in particular the case of high dimensions and state-specific graphical models. As we demonstrated in simulated and real data examples, the methodology we propose substantially improves upon current practice. Our results demonstrate the utility of regularization for HMMs, even when sample sizes are not small.

It is interesting to consider why careful penalization is needed in HMMs (and related latent variable settings like mixture models). In a simple linear model, as in regression, the ratio n/p is a measure to distinguish between a low- and high-dimensional problem. If the ratio n/p small, classical least-squares estimation leads to poor predictive performance due to a large number of predictors compared to a small sample size. On the other hand if n/p is large (for example > 20), then, very likely, least-squares regression performs well. In HMMs (and mixtures) the situation is more subtle. It is instructive to consider the ratios n_k/p (n_k denotes the number of samples belonging to state k) as a measure whether an inference problem is high-dimensional or not. If for at least one state this ratio is small, then MLE is likely to overfit and results in a poor generalization error. A fundamental

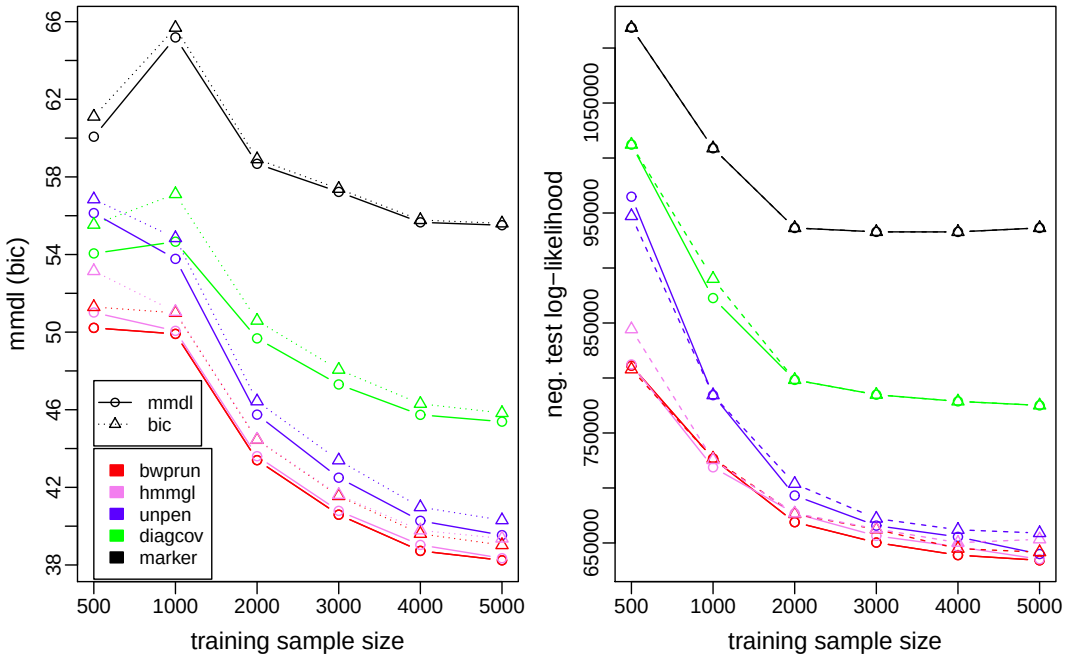

Figure 4: Genomic data, MMDL(BIC) and predictive performance. Models were fitted to protein binding data from Filion *et al.* (2010) (see text for details) and tested on held-out data from the same study. Left panel: MMDL(BIC)-scores (scaled by n_{train}) for different methods trained on the first $n_{\text{train}} = 500, 1000, \ldots, 5000$ observations of chromosome 2L. Right panel: negative test log-likelihood evaluated on a test set (second half of chromosome 2R; training data is from parts of chromosome 2L). [Legend: Greedy Backward Pruning (**Bwprun**); HMMGLasso (**Hmmgl**); Unpenalized MLE (**Unpen**); MLE with diagonal restricted covariance matrices (**Diagcov**); Five-state MLE using only marker proteins (**Marker**)]

problem that we emphasized throughout the paper is the fact that the ratios n_k/p depend on the number of states K and on the state-sizes n_k, which are themselves usually unknown *a priori*. So, a seemingly low-dimensional problem with a large sample size and with a moderate number of features can become a high-dimensional task in practice, especially if a large number of states cannot be ruled out *a priori*. In fact, our simulations illustrate that even when $\min_k n_k/p$ is relatively large, the MLE can be ill-behaved. For example, in our simulated Model 2, with $K = 2$, we have $n = 2000$ and $n_k/p > 13$ in each state; nevertheless the MLE fails completely to recover correct state assignments (Fig 8).

A straightforward approach to handle inference in high-dimensional HMMs is to fix constraints on the state-specific covariance matrices (for example assuming diagonal covariance matrices). However, such an approach leads to poor predictive performance when the assumption is invalid and precludes discovery of state-specific covariance structure. As in the genome biology example we considered, such structure may itself be of scientific interest. We note also that the hidden nature of the states makes it difficult to test any such model assumption. In fact, if the covariance matrices of an HMM with a specific

number of states satisfy some constraints, than these constraints do not necessarily hold for an HMM with smaller or larger number of states (Simpson's paradox).

In the context of mixtures, there is a growing literature on penalized likelihood methods which address the high-dimensional context to some extent (Khalili and Chen, 2007; Städler *et al.*, 2010; Pan and Shen, 2007). However, none of these methods address the need to ensure penalties are able to handle state-specific scaling (that cannot be dealt with by pre-processing) and size (that is unknown at the outset). The selection of the number of mixture components also remains an open issue in this literature. Our approach handles these issues that arise due to the hidden nature of the states and could be straightforwardly applied in the mixture model setting. Further generalization to other latent variable models may also be possible.

The backward pruning approach gives an efficient way to estimate parameters for a sequence of candidate number of states K. If desired, a single "optimal" number of states can then be selected using model selection criteria, as we demonstrated in examples above. For a given estimator, the optimal number of states is well defined in a predictive sense as the value that minimizes risk. From this point of view it is easy to understand why the prediction-optimal number of states may be higher under regularization or when more training data are available (see Figure 5). However, when scientific understanding rather than prediction alone is one of the goals of analysis, it is not clear whether it is useful to think in terms of a "correct" number of states. Rather, it may be useful to think of the estimates $\{\Theta_K\}$ that we obtain via backward pruning as collectively providing a resource for exploration of a system of interest.

In the genome biology example we considered, penalization led to gains in predictive ability relative to the MLE and to reduced dimension approaches that have been used in the literature. This suggests that despite redundancy in biological signals, a multivariate view can enhance predictive ability. Further, we were able to learn richer models than is possible using currently available methods, including estimates of state-specific graphical model structure. The latter may shed light on protein-protein interplay that is specific to genomic region; such interplay has not been investigated to date and is one focus of our ongoing efforts in this application area. We used data from Filion *et al.* (2010); we note that the main substantive conclusions drawn in that paper are broadly supported by our analyses and the richer set of states uncovered by our approach are related to the states they report. Genomic datasets are becoming increasingly high-dimensional and we anticipate that the methodology presented here will be useful to researchers in that field. Beyond biology, applications for high-dimensional HMMs are numerous, including in signal processing.

We showed that the approaches we put forward for HMMs, including universal regularization and Greedy Backward Pruning, work well in empirical examples. However, there remains a need for theoretical investigation of these ideas. Our penalty in combination with λ_{uni} was inspired by making connections to results obtained for the well-studied Lasso case. A challenge for future theoretical work is to provide insight into optimality of these and related approaches and establish global convergence properties of penalized estimation in latent variable settings.

Acknowledgements: We are grateful to Bas van Steensel and his lab for introducing us to the genome biology of *Drosophila* and for a productive, ongoing collaboration.

References

Barron, A., Huang, C., Li, J. Q. and Luo, X. (2008) *MDL Principle, Penalized Likelihood, and Statistical Risk*. MIT Press Books. Tampere University Press, Tampere, Finland.

Bicego, M., Murino, V. and Figueiredo, M. A. T. (2003) A sequential pruning strategy for the selection of the number of states in hidden Markov models. *Pattern Recognition Letters*, **24**, 1395–1407.

Bühlmann, P. and van de Geer, S. (2011) *Statistics for High-Dimensional Data: Methods, Theory and Applications*. Springer Series in Statistics. Springer.

Donoho, D. L. and Johnstone, I. M. (1994) Ideal spatial adaptation by wavelet shrinkage. *Biometrika*, **81**, 425–455.

Durbin, R., Eddy, S. R., Krogh, A. and Mitchison, G. J. (1998) *Biological Sequence Analysis: Probabilistic Models of Proteins and Nucleic Acids*. Cambridge University Press.

Ernst, J. and Kellis, M. (2010) Discovery and characterization of chromatin states for systematic annotation of the human genome. *Nature Biotechnology*, **28**, 817–825.

Figueiredo, M. A. T. and Jain, A. K. (2000) Unsupervised learning of finite mixture models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **24**, 381–396.

Figueiredo, M. A. T., Leitão, J. M. N. and Jain, A. K. (1999) On fitting mixture models. In *Proceedings of the Second International Workshop on Energy Minimization Methods in Computer Vision and Pattern Recognition*, EMMCVPR '99, 54–69. Springer-Verlag.

Filion, G. J., van Bemmel, J. G., Braunschweig, U., Talhout, W., Kind, J., Ward, L. D., Brugman, W., de Castro, I. J., Kerkhoven, R. M., Bussemaker, H. J. and van Steensel, B. (2010) Systematic protein location mapping reveals five principal chromatin types in drosophila cells. *Cell*, **143**, 212–224.

Fraley, C. and Raftery, A. E. (2002) Model-based clustering, discriminant analysis and density estimation. *Journal of the American Statistical Association*, **97**, 611–631.

Fraley, C. and Raftery, A. E. (2006) MCLUST version 3 for R: Normal mixture modeling and model-based clustering. Technical Report 504, University of Washington, Department of Statistics.

Friedman, J., Hastie, T. and Tibshirani, R. (2008) Sparse inverse covariance estimation with the graphical Lasso. *Biostatistics*, **9**, 432–441.

Grünwald, P. D. (2007) *The Minimum Description Length Principle*. MIT Press Books. The MIT Press.

Khalili, A. and Chen, J. (2007) Variable selection in finite mixture of regression models. *Journal of the American Statistical Association*, **102**, 1025–1038.

Meinshausen, N. and Bühlmann, P. (2006) High dimensional graphs and variable selection with the Lasso. *Annals of Statistics*, **34**, 1436–1462.

Pan, W. and Shen, X. (2007) Penalized model-based clustering with application to variable selection. *Journal of Machine Learning Research*, **8**, 1145–1164.

Park, P. (2009) ChIP–seq: advantages and challenges of a maturing technology. *Nature Reviews Genetics*, **10**, 669–680.

Rothman, A., Bickel, P., Levina, E. and Zhu, J. (2008) Sparse permutation invariant covariance estimation. *Electronic Journal of Statistics*, **2**, 494–515.

Städler, N., Bühlmann, P. and van de Geer, S. (2010) l1-penalization for mixture regression models (with discussion). *TEST*, **19**, 209–285.

Sun, T. and Zhang, C.-H. (2011) Scaled sparse linear regression. *arXiv.org: 1104.4595*.

Tibshirani, R. (1996) Regression shrinkage and selection via the Lasso. *Journal of the Royal Statistical Society, Series B*, **58**, 267–288.

van Steensel, B. and Henikoff, S. (2000) Identification of in vivo DNA targets of chromatin proteins using tethered dam methyltransferase. *Nature Biotechnology*, **18**, 424–428.

Yuan, M. and Lin, Y. (2007) Model selection and estimation in the Gaussian graphical model. *Biometrika*, **94**, 19–35.

Zhang, C.-H. (2010) Nearly unbiased variable selection under minimax concave penalty. *Annals of Statistics*, **38**, 894–942.

A Graphical Lasso with different penalty functions

In this section we briefly discuss optimization and performance of the Graphical Lasso problem

$$\hat{\Omega} = \underset{\Omega}{\operatorname{argmin}} - \log|\Omega| - \operatorname{tr}(\mathbf{S}\Omega) + \rho \operatorname{Pen}(\Omega) \tag{A.7}$$

with the different penalty functions $\operatorname{Pen}(\cdot)$ introduced in Section 2.1.

A.1 Optimization

Case 1: $\operatorname{Pen}_{\text{invcov}}(\Omega) = \|\Omega^{-}\|_1$. This case can be directly solved by the GLasso algorithm presented in Friedman *et al.* (2008). This algorithm is implemented in the **R**-package `glasso`. Note that this implementation allows specification of different penalty levels for different entries in Ω.

Case 2: $\text{Pen}_{\text{invcor}}(\Omega) = \|\Phi^-\|_1$, where Φ is the inverse correlation matrix. Note, that we can write $\Omega = V\Phi V$ where V is a diagonal matrix with entries $V_{jj} = 1/\sqrt{\Sigma_{jj}}$. Now, the objective function in (A.7) can be written as

$$-2\log|\mathrm{V}| - \log|\Phi| + \mathrm{tr}(\mathrm{V}\mathbf{S}\mathrm{V}\Omega) + \rho\|\Phi^-\|_1. \tag{A.8}$$

or as

$$-\log|\Omega| - \mathrm{tr}(\mathbf{S}\Omega) + \sum_{\mathrm{j}\neq\mathrm{j}'} \frac{\rho}{\sqrt{\mathrm{V}_{\mathrm{jj}}\mathrm{V}_{\mathrm{j}'\mathrm{j}'}}}|\Omega_{\mathrm{jj}'}|. \tag{A.9}$$

Taking partial derivatives of (A.8) with respect to Φ_{jj} yields

$$V_{jj} = 1/\sqrt{\mathbf{S}_{jj}} \quad (j = 1, \ldots, p). \tag{A.10}$$

By plugging-in (A.10) into equation (A.9) the desired optimization problem can be solved with the GLasso algorithm.

Case 3: $\text{Pen}_{\text{parcor}}(\Omega) = \|\Psi^-\|_1$, where Ψ is the partial correlation matrix. The objective function in (A.7) then equals

$$-\log|\Omega| - \mathrm{tr}(\mathbf{S}\Omega) + \sum_{\mathrm{j}\neq\mathrm{j}'} \frac{\rho}{\sqrt{\Omega_{\mathrm{jj}}\Omega_{\mathrm{j}'\mathrm{j}'}}}|\Omega_{\mathrm{jj}'}|. \tag{A.11}$$

We solve (A.11) by setting $\Omega^{(0)} = \text{Diag}(\mathbf{S})^{-1}$ and iteratively calling the GLasso algorithm according to:

$$\Omega^{(i+1)} \quad = \quad \underset{\Omega}{\text{argmin}} -\log|\Omega| - \mathrm{tr}(\mathbf{S}\Omega) + \sum_{\mathrm{j}\neq\mathrm{j}'} \frac{\rho}{\sqrt{\Omega^{(\mathrm{i})}_{\mathrm{jj}}\Omega^{(\mathrm{i})}_{\mathrm{j}'\mathrm{j}'}}}|\Omega_{\mathrm{jj}'}| \quad (\mathrm{i} = 0, 1, 2, \ldots).$$

Note, that with the **R**-package `glasso` we can specify different penalty levels (here: $\rho/\sqrt{\Omega^{(i)}_{jj}\Omega^{(i)}_{j'j'}}$, $j, j' = 1, \ldots, p$) for different entries in Ω.

A.2 Performance

We compare the penalty functions $\text{Pen}_{\text{invcov}}$, $\text{Pen}_{\text{parcor}}$ and $\text{Pen}_{\text{invcor}}$ proposed in Section 2.1 under two different regimes:

Model 5: Gaussian graphical model with $p = 50$ and $n = 100$; Concentration matrix Ω is generated as in Rothman *et al.* (2008) with p nonzero (off-diagonal) entries; Diagonal standardized to have entries equal one.

Model 6: Gaussian graphical model with $p = 50$ and $n = 100$; Concentration matrix follows an AR(1) model with $\Sigma_{ll'} = 0.9^{|l-l'|}$. Note that in the AR(1) model, the diagonal entries of Ω are not equal to one and therefore Ω does not coincide with the partial correlation matrix.

We generate training and test data for each model. For Model 5 and Model 6 we fit estimator A.7 using different penalty functions and various tuning parameters (including λ_{uni}) on the training data and on scaled training data, where half of the variables are

scaled by 0.1 and the other half by 10 (in Model 5 the two halves are chosen randomly for each simulation run; in Model 6 one half are the variables $1, 3, 5, \ldots, 49$ and the other half the variables $2, 4, 6, \ldots, 50$). We then report the log-likelihood obtained on test data.

Boxplots in Figure 6 clearly demonstrate that penalization with $\text{Pen}_{\text{parcor}}$ and $\text{Pen}_{\text{invcor}}$ is scale invariant, whereas penalizing the ℓ_1-norm of the inverse covariance matrix, which is the common practice in the Graphical Lasso, is not. The results in Figure 6 also agree with our findings in Section 2.2: Regularization with λ_{uni} performs well with $\text{Pen}_{\text{invcov}}$ only in Model 5, where Ω equals the partial correlation matrix. In both setting $\text{Pen}_{\text{invcor}}$ with λ_{uni} does not perform well. However, penalizing the ℓ_1-norm of the partial correlation matrix using λ_{uni} does a good job in both models, i.e., nearly as good as picking the best solution obtained with λ_{opt}. In Model 5, where Ω coincides with the partial correlation matrix, $\text{Pen}_{\text{invcov}}$ performs slightly better then $\text{Pen}_{\text{parcor}}$ as the former penalty is optimal from a distributional point of view. Similarly, in simulation Experiment I, where all Ω's in the data generating models are standardized to have unit diagonal entries, we obtain slightly better performance if we use $\text{Pen}_{\text{invcov}}$ instead.

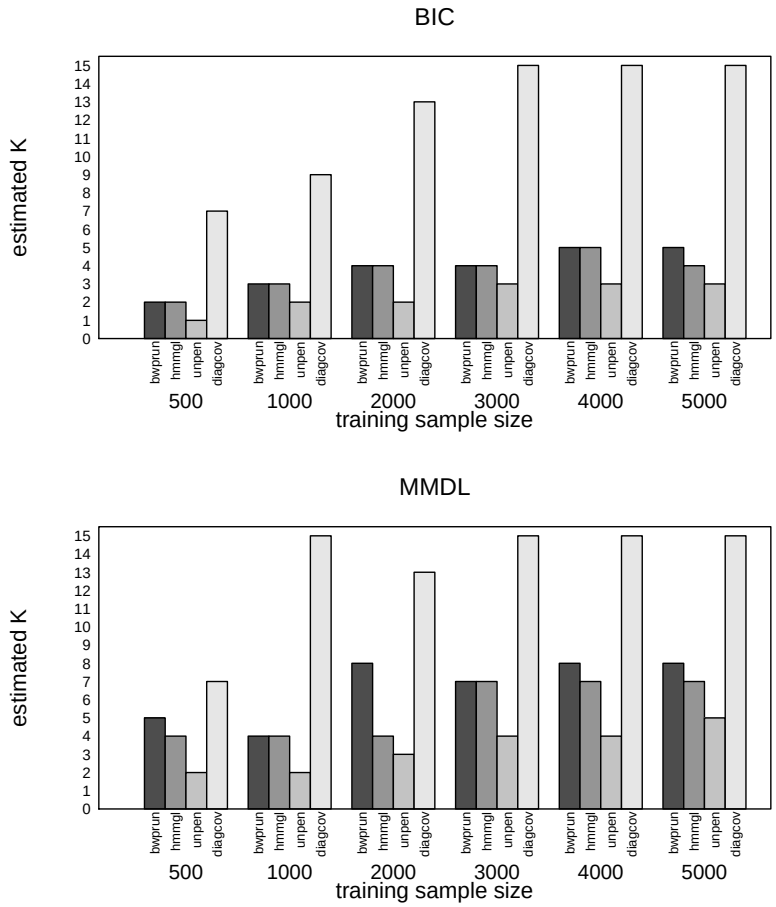

Figure 5: Genomic data, number of states. Number of states selected (at various training sample sizes) by Greedy Backward Pruning (**Bwprun**), HMMGLasso (**Hmmgl**), unpenalised MLE (**Unpen**) and MLE with diagonal restricted covariance matrices (**Diagcov**). All methods are trained on parts of chromosome 2L and use MMDL or BIC as the model selection criterion. The number of states in Hmmgl, Unpen and Diagcov are determined by a forward stepwise selection.

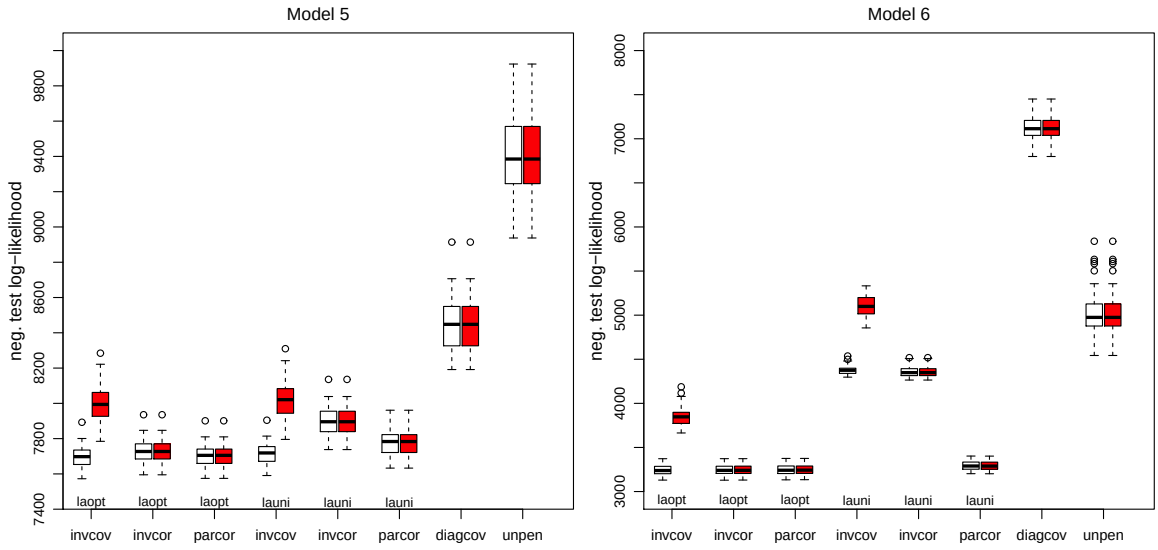

Figure 6: Test log-likelihood for different penalty functions for simulation Models 5 (left panel) and 6 (right panel). Red boxplots show results obtained from fitting on scaled training data and back-transforming parameter estimates on original scale.

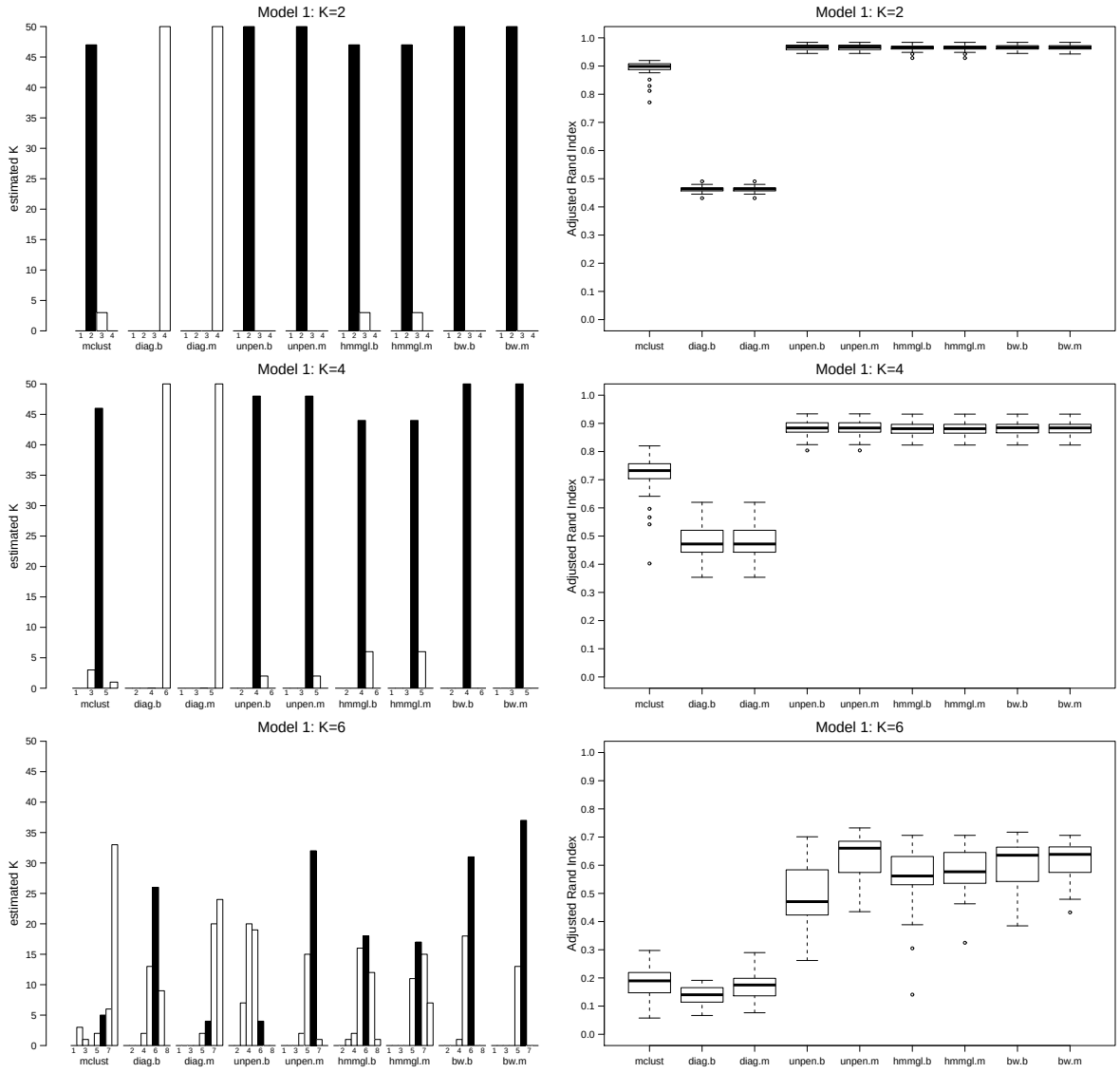

Figure 7: Simulation Experiment I, Model 1 ($p = 10, n = 2000$), number of states and state assignments. Left panels: frequency of estimated number of states (as in Fig 1 correct number of states indicated in black). Right panels: adjusted Rand index with respect to true state assignments. [Legend: Results for Mclust (**mclust**), MLE with diagonal covariance matrices (**diag**), MLE (**unpen**) and Greedy Backward Pruning (**bw**) are shown. Extensions ".b" and ".m" stand for BIC and MMDL respectively.]

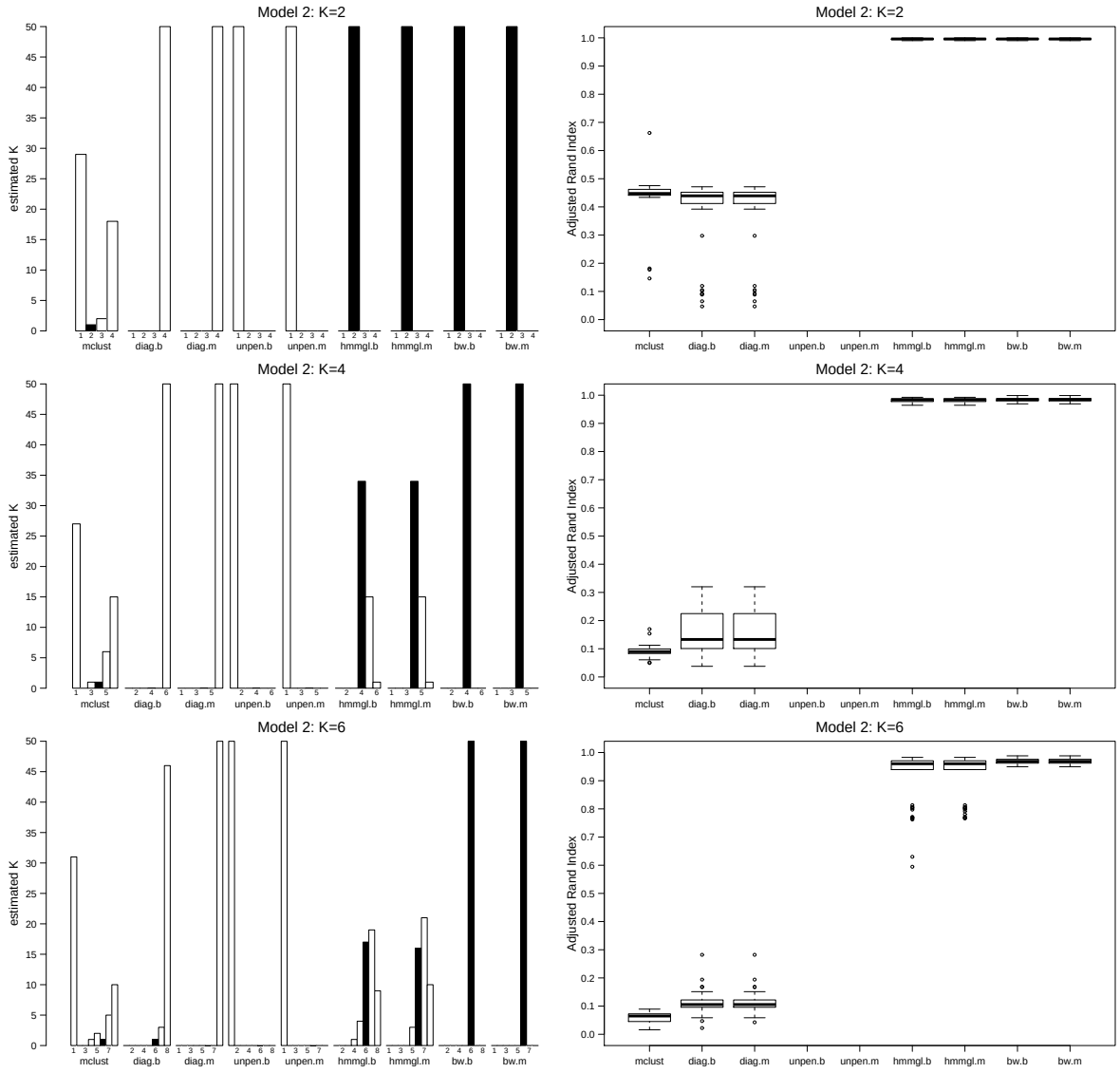

Figure 8: Model 2 ($p = 75, n = 2000$), number of states and state assignments. Left panels: frequency of estimated number of states (correct number of states indicated in black). Right panels: adjusted Rand index with respect to true state assignments. [Legend: Results for Mclust (**mclust**), MLE with diagonal covariance matrices (**diag**), MLE (**unpen**) and Greedy Backward Pruning (**bw**) are shown. Extensions ".b" and ".m" stand for BIC and MMDL respectively.]

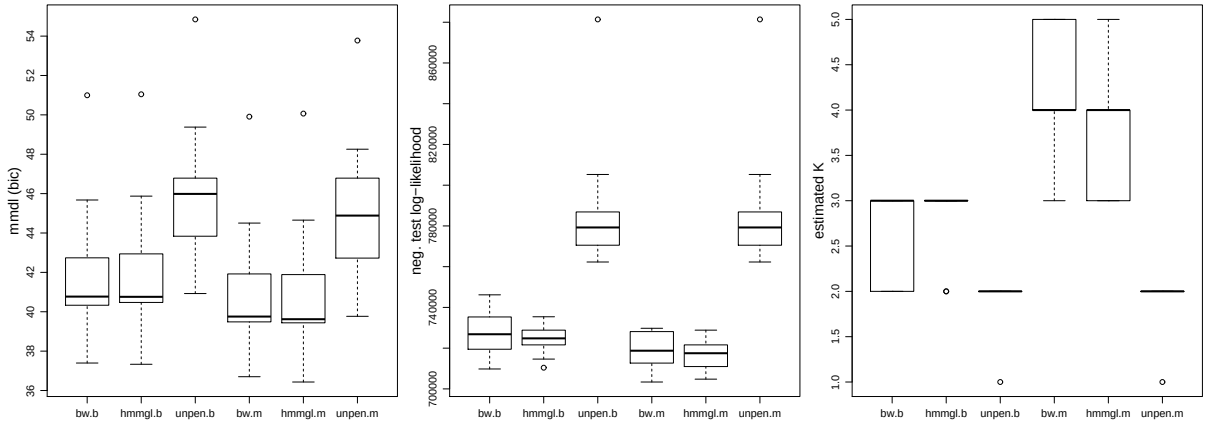

Figure 9: Genomic data (Section 3.2), multiple training datasets. Boxplots of MMDL(BIC)-scores, negative test log-likelihood (using test data as described in text) and number of selected states obtained over nine training datasets each of size $n_{\text{train}} = 1000$. The ten datasets were obtained from chromosome 2L as $X_{t_0}, \ldots, X_{t_0+1000}$ with $t_0 = 0, 500, 1000, 1500, \ldots, 4000$. [Legend: Backward Pruning (bw), HMMGLasso (hmmgl) and unpenalized MLE (unpen); extensions ".b" and ".m" stand for BIC respectively MMDL.]