MODEL SELECTION IN A SETTING WITH LATENT VARIABLES

Ralf Eggeling[1]*, Teemu Roos*[2]*, Petri Myllymäki*[2]*, Ivo Grosse*[1]

[1]Institute for Computer Science, Martin Luther University Halle-Wittenberg,
06099 Halle, GERMANY, {eggeling|grosse}@informatik.uni-halle.de

[2]Helsinki Institute for Information Technology HIIT, University of Helsinki,
P.O.Box 68, FIN-00014 Helsinki, FINLAND, {teemu.roos|petri.myllymaki}@hiit.fi

1. BACKGROUND

Model selection, the task of selecting a statistical model from a certain model class given data, is an important problem in statistical learning. From another perspective model selection can also be viewed as learning a single distribution, where the parameter space includes a discrete structure parameter s which imposes further constraints on the remaining parameterization of that model, so that learning a distribution decomposes intro learning s and the corresponding probability parameters. Examples for model classes that offer this kind of structural flexibility and thus entail a model selection problem are Bayesian networks [1] (where s is the DAG), Markov chains (where s is the order), and variable order [2] and parsimonious [3] Markov models (where s is the context tree structure). Model selection has been well studied in the past decades, and one popular approach is Bayesian model selection, where candidate structures are evaluated according to their Bayesian marginal likelihood. An alternative approach is based on the Minimum Description Length principle [4], and uses the Normalized Maximum Likelihood distribution [5] or approximations thereof [6] as structure score.

In the simplest case, model selection is based on the idea that all observations in the data set follow the same distribution, even though the parametric form of the distribution is not known. Another problem arises if we drop that iid assumption and assume each data point to be generated from one out C possible distributions. General examples for this setting are mixture models and hidden Markov models, more specialized applications are promoter models in computational biology [7, 8]. Parameter learning in a setting with latent variables is well studied, and one popular solution is the EM algorithm [9], which can be perceived as a soft clustering algorithm.

Combining the latent variable and model selection problem in the same framework is comparatively unexplored, though. One example are mixture models where each component c also comprises a structure parameter s_c, which determines the precise parameterization of the distribution of the component. If all candidate models within a component have the same dimensionality, extending the EM approach for learning latent variable models can by used for learning both the structure and the parameters of each component as shown for a mixture of tree models [10]. However, further extending this approach to model classes where candidate models have different dimensionality [11] is not as straightforward, since it can be shown that asymptotically the largest candidate structures will be selected. Here, we follow an alternative approach by extending the model selection for full observations to the presence of latent variables instead of extending the latent variable learning of fixed-structure models to variable structures.

2. APPROACH

Let $\vec{x} = (x_1, \ldots, x_N)$ denote the data. We assume each data point belonging to one out of C mixture components. Since the particular assignment is unknown, a *latent variable* $u_i \in (1, \ldots, C)$ contains the component label of data point x_i for each $i \in (1, \ldots, N)$.

Each of the C components may follow a different distribution (parameterized by θ) from a different model from a different model class. We denote the class of candidate models of component c by $\mathcal{M}_c$, one particular (selected) structure by s_c and the corresponding probability parameters by θ_{cs_c}. We combine the latter by $\Theta_c = (s_c, \theta_{cs_c})$, and all parameters of the mixture model by $\vec{\Theta} = (\Theta_1, \ldots, \Theta_C)$.

We assume that the computation of $P(\vec{x}|\vec{u}, \vec{\Theta})$ is feasible. If $C = 1$, learning is done by first selecting a model structure s according to $P(s|\vec{x})$, with subsequent computation of the parameters θ_s. If $C > 1$, the equivalent optimization problem

$$\hat{\vec{s}} = \operatorname*{argmax}_{\vec{s}} P(\vec{s}|\vec{x}) \tag{1}$$

cannot be solved efficiently, since the computation of the quantity

$$P(\vec{s}|\vec{x}) = \sum_{\vec{u}} P(\vec{s}|\vec{u}, \vec{x}) P(\vec{u}|\vec{x}) \tag{2}$$

involves an exponential sum over all possible values of latent variables. However, if we approximate $P(\vec{u}|\vec{x})$, by

some estimate $\vec{u}^\star$ and assign

$$P(\vec{u}|\vec{x}) = \begin{cases} 1, & \text{if } \vec{u} = \vec{u}^\star \\ 0, & \text{else} \end{cases}, \tag{3}$$

then we obtain

$$\hat{\vec{s}} = \operatorname*{argmax}_{\vec{s}} P(\vec{s}|\vec{u}^\star, \vec{x}), \tag{4}$$

which decomposes as

$$\forall_{c=1}^{C} \hat{s}_c = \operatorname*{argmax}_{s_c} P(s_c|\vec{u}^\star, \vec{x}). \tag{5}$$

In this way, model selection with latent variables can be reduced to a setting of model selection from completely observed data. However, it imposes the additional task of finding a good estimate $\vec{u}^\star$. Since the final goal is model selection, we pick the best assignment of latent variables according to

$$\vec{u}^\star = \operatorname*{argmax}_{\vec{u}} \prod_{c=1}^{C} \max_{s_c} P(s_c|\vec{x}, \vec{u}) \tag{6}$$

We propose to approximate this quantity by the following (stochastic) algorithm, which can be perceived as a hybrid form of Gibbs sampling and k-Means:

Algorithm 1 Iterative algorithm for finding the optimal model structures in a setting with latent variables.

for $t = 1, \ldots, T$ **do**
  for $i = 1, \ldots, N$ **do**
    sample $u_i^{(t)}$ from $\begin{cases} \left(\frac{1}{C}, \ldots, \frac{1}{C}\right) & \text{if } t = 1 \\ P(u_i|x_i, \vec{\Theta}^{(t-1)}) & \text{if } t > 1 \end{cases}$
  end for
  for $c = 1, \ldots, C$ **do**
    compute $s_c^{(t)} = \operatorname*{argmax}_{s_c} P(s_c|\vec{x}, \vec{u}^{(t)})$
    compute $\theta_{cs_c}^{(t)} = \operatorname*{argmax}_{\theta_{cs_c}} P(\theta_{cs_c}|s_c^{(t)}, \vec{x}, \vec{u}^{(t)})$
  end for
end for

Algorithm 1 generates a series of $(\vec{u}, \vec{\Theta})^{(1)}, \ldots, (\vec{u}, \vec{\Theta})^{(T)}$. We are interested in the function

$$f(t) = \prod_{c=1}^{C} P(s_c^{(t)}|\vec{x}, \vec{u}^{(t)}), \tag{7}$$

where t denotes an iteration step. We define the best iteration step as

$$\hat{t} = \operatorname*{argmax}_{t=1,\ldots,T} f(t) \tag{8}$$

and select $\vec{u}^\star = \vec{u}^{(\hat{t})}$, which is asymptotically identical to Equation 6. Following Equation 5, we thus select the model structures by

$$\forall_{c=1}^{C} : \hat{s} = s_c^{(\hat{t})}. \tag{9}$$

This procedure is generally independent of the model classes $\mathcal{M}_1, \ldots, \mathcal{M}_C$, provided that model selection can be done efficiently for fully observable data.

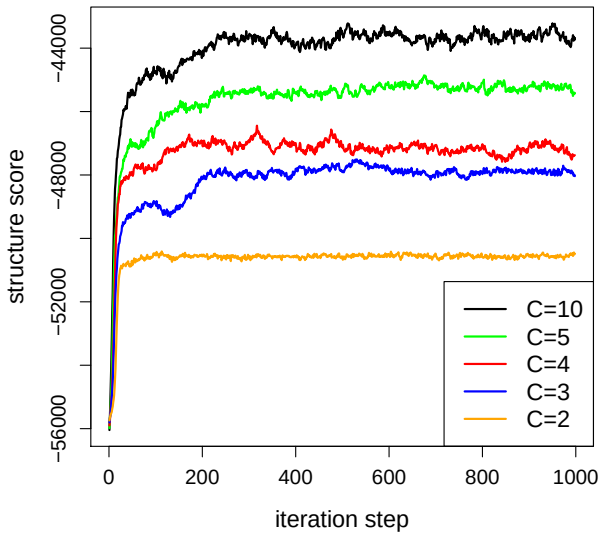

Figure 1. Development of the structure score f of Equation 7 with increasing iteration steps for PMM(2) mixture models with $C = 2, \ldots, 5, 10$ components.

Moreover, there are no restrictions concerning the usage of different structure scores. Due to the estimation of values $\vec{u}^\star$, which yields model selection on integer counts instead of real-valued expected values, also NML-based scoring criteria can be applied.

3. EXPERIMENTS

We apply the method to learning mixtures of inhomogeneous parsimonious Markov models (PMMs) [12] on the splice site data of Yeo and Burge [13]. The data set consists of 12,623 sequences of length 7 over a four-letter alphabet. It has been splitted by into training and test data at the ratio of 2:1 [13] and we rely on the same partitioning.

First, we study the convergence behavior of the algorithm for the given model class and data. We learn mixture models with different number of components, i.e., $C \in \{2, 3, 4, 5, 10\}$ on the training data set. As component models, we use PMMs of order 2, with fNML as structure score s_c and fsNML as parameter estimates [14]. We show the function $f(t)$ for $T = 10^3$ in Figure 1. Due to the stochastic nature of the algorithm, the target function does not monotonically increase, but it quickly converges to comparatively stable levels. It may be not surprising that for a smaller number of mixture components the convergence is faster, especially the two-component mixture converges within less then 100 iteration steps.

In a second study, we focus on the prediction performance of mixtures of PMMs. We learn the models on the training data and compute the negative logarithmic probability of the test data given the learned model. Here, we also include $C = 1$ for comparison, which represents the non-mixture case (where learning can be done analytically). In addition, we also compare to mixtures of PMMs

Table 1. Prediction performance of mixtures of parsimonious Markov models. The rows represent different numbers of mixture components and the colums represent the maximal order of the PMM. The table entries are the negative logarithmic predictive probabilities of the test data set. The optimal number of mixture component for each maximal model order is emphasized.

C	PMM(0)	PMM(1)	PMM(2)	PMM(3)
1	28,883.7	28,103.8	27,775.4	27,509.1
2	27,978.0	27,480.6	27,196.2	27,183.2
3	27,681.1	27,231.7	27,147.8	**27,152.7**
4	27,527.0	27,189.5	27,143.5	27,193.0
5	27,383.2	27,157.3	**27,130.3**	27,184.7
10	**27,234.8**	**27,123.4**	27,182.8	27,245.4

of maximal order 0, 1, and 3. The maximal order of the PMM defines the search space of the structure learning algorithm, so there is no structure learning for PMM(0), which is equivalent to a simple independence model. We obtain a matrix of prediction values where the rows constitute the number of mixture components and the columns the maximal order of the component PMMs (Table 1).

We observe that a simple independence model has the worst negative log prediction of $28,883.7$, and increasing the degree of statistical dependence that is taken into account by increasing the maximal model order to 3 gradually improves prediction performance (first row). Since PMM(1)-PMM(3) infer the optimal parsimonious context tree structure from data, overfitting is avoided. Using a mixture of simple independence model also gradually improves prediction performance up to $C = 10$ (first column).

The combination of using a mixture model and inferring the optimal structure within each component, yields even better performances, though. The greatest improvement is – for all maximal model orders – achieved by using a two component mixture instead of a single distribution. Further improvements are generally small, with the optimum being a five-component mixture of second-order PMMs. The combination of many mixture components and complex component models decreases in some cases prediction slightly compared to the simpler alternatives, which is a clear indication that here overfitting w.r.t. the number of mixture components occured.

4. REFERENCES

[1] G. Heckerman, D. Geiger, and D. Chickering, "Learning Bayesian Networks: The Combination of Knowledge and Statistical Data," *Machine Learning*, vol. 20, pp. 197–243, 1995.

[2] Jorma Rissanen, "A universal data compression system," *IEEE Trans. Inform. Theory*, vol. 29, no. 5, pp. 656–664, 1983.

[3] P.Y. Bourguignon and D. Robelin, "Modèles de Markov parcimonieux," in *Proceedings of JOBIM*, 2004.

[4] Jorma Rissanen, "Modeling By Shortest Data Description," *Automatica*, vol. 14, pp. 465–471, 1978.

[5] Y.M. Shtarkov, "Universal sequential coding of single messages," *Problems of Information Transmission*, vol. 23, pp. 3–17, 1987.

[6] T. Silander, T. Roos, and P. Myllymäki, "Learning Locally Minimax Optimal Bayesian Networks," *International Journal of Approximate Reasoning*, vol. 51, pp. 544–557, 2010.

[7] C.E. Lawrence and A.A. Reilly, "An Expectation Maximization Algorithm for the Identification and Characterization of Common Sites in Unaligned Biopolymer Sequences.," *Proteins: Structure, Function and Genetics*, vol. 7, pp. 41–51, 1990.

[8] W. Thompson, M.J. Palumbo, W.W. Wasserman, J.S. Liu, and C.E. Lawrence, "Decoding Human Regulatory Circuits," *Genome Research*, vol. 14, pp. 1967–1974, 2004.

[9] A.P. Dempster, N.M. Laird, and D.B. Rubin, "Maximum Likelihood from Incomplete Data via the EM Algorithm," *Journal of the Royal Statistical Society*, vol. 39, no. 1, pp. 1–38, 1977.

[10] M. Meila and M.I. Jordan, "Learning with Mixtures of Trees," *Journal of Machine Learning Research*, vol. 1, pp. 1–48, 2000.

[11] A. Gohr, S. Posch, and I. Grosse, "Mixtures of Parsimonious Markov Models," Tech. Rep. 2012/2, Institute of Computer Science, Martin Luther University Halle-Wittenberg, Germany, 2012.

[12] R. Eggeling, A. Gohr, P.-Y. Bourguignon, E. Wingender, and I. Grosse, "Inhomogeneous Parsimonious Markov Models," in *Proceedings of the 2013 European Conference on Machine Learning and Knowledge Discovery in Databases (ECML/PKDD)*, 2013, to appear.

[13] G. Yeo and C.B. Burge, "Maximum Entropy Modeling of Short Sequence Motifs with Applications to RNA Splicing Signals," *Journal of Computational Biology*, vol. 11(2/3), pp. 377–394, 2004.

[14] R. Eggeling, T. Roos, P. Myllymäki, and I. Grosse, "Comparison of NML and Bayesian scoring criteria for learning parsimonious Markov models," in *Proc. 5th Workshop on Information Theoretic Methods in Science and Engineering (WITMSE-12)*, Amsterdam, 2012.