Helsinki University of Technology
Publications in Computer and Information Science

Report E3
April 2006

COMPACT MODELING OF DATA USING INDEPENDENT VARIABLE GROUP ANALYSIS

Esa Alhoniemi Antti Honkela Krista Lagus Jeremias Seppä
Paul Wagner Harri Valpola

TEKNILLINEN KORKEAKOULU
TEKNISKA HÖGSKOLAN
HELSINKI UNIVERSITY OF TECHNOLOGY
TECHNISCHE UNIVERSITÄT HELSINKI
UNIVERSITE DE TECHNOLOGIE D'HELSINKI

Distribution:
Helsinki University of Technology
Department of Computer Science and Engineering
Laboratory of Computer and Information Science
P.O. Box 5400
FI-02015 TKK, Finland
Tel. +358-9-451 3267
Fax +358-9-451 3277

This report is downloadable at
http://www.cis.hut.fi/Publications/

ISBN 951-22-8166-X
ISSN 1796-2803

Compact Modeling of Data Using Independent Variable Group Analysis

Esa Alhoniemi and Antti Honkela and Krista Lagus and Jeremias Seppä and Paul Wagner and Harri Valpola

***Abstract*— We introduce a principle called independent variable group analysis (IVGA) which can be used for finding an efficient structural representation for a given data set. The basic idea is to determine such a grouping for the variables of the data set that mutually dependent variables are grouped together whereas mutually independent or weakly dependent variables end up in separate groups.**

Computation of any model that follows the IVGA principle requires a combinatorial algorithm for grouping of the variables and a modeling algorithm for the groups. In order to be able to compare different groupings, a cost function which reflects the quality of a grouping is also required. Such a cost function can be derived for example using the variational Bayesian approach, which is employed in our study. This approach is also shown to be approximately equivalent to minimizing the mutual information between the groups.

The modeling task is computationally demanding. We describe an efficient heuristic grouping algorithm for the variables and derive a computationally light nonlinear mixture model for modeling the dependencies within the groups. Finally, we carry out a set of experiments which indicate that the IVGA principle can be beneficial in many different applications.

***Index Terms*— compact modeling, independent variable group analysis, mutual information, variable grouping, variational Bayesian learning**

I. Introduction

The study of effective ways of finding compact representations from data is important for the automatic analysis and exploration of complex data sets and natural phenomena. Finding properties of the data that are not related can help in discovering compact representations as it saves from having to model the mutual interactions of unrelated properties.

It seems evident that humans group related properties as a means for understanding complex phenomena. An expert of a complicated industrial process such as a paper machine may describe the relations between different control parameters and measured variables by groups: A affects B and C, and so on. This grouping is of course not strictly valid as all the variables eventually depend on each other, but it helps in describing the most important relations, and thus makes it possible for the human to understand the system. Such groupings also significantly help the interaction with the process. Automatic discovery of such groupings would help in designing visualizations and control interfaces that reduce the cognitive load of the user by allowing her to concentrate on the essential details.

Analyzing and modeling intricate and possibly nonlinear dependencies between a very large number of real-valued variables (features) is a hard problem. Learning such models from data generally requires very much computational power and memory. If one does not limit the problem by assuming only linear or other restricted dependencies between the variables, essentially the only way to do this is to actually try to model the data set using different model structures. One then needs a principled way to score the structures, such as a cost function that accounts for the model complexity as well as model accuracy.

The remainder of the article is organized as follows. In Section II we describe a computational principle called Independent Variable Group Analysis (IVGA) by which one can learn a structuring of the problem from data. In short, IVGA does this by finding a partition of the set of input variables that minimizes the mutual information between the groups, or equivalently the cost of the overall model, including the cost of the model structure and the representation accuracy of the model. Its connections to related methods are discussed in Section II-B.

The problem of modeling-based estimation of mutual information is discussed in Section III. The approximation turns out to be equivalent to variational Bayesian learning. Section III also describes one possible computational model for representing a group of variables as well as the cost function for that model. The algorithm that we use for finding a good grouping is outlined in Section IV along with a number of speedup techniques.

In Section V we examine how well the IVGA principle and the current method for solving it work both on an artificial toy problem and two real data sets of printed circuit board assembly component database setting values and ionosphere radar measurements.

Initially, the IVGA principle and an initial computational method was introduced in [1], and some further experiments were presented in [2]. In the current article we derive the connection between mutual information and variational Bayesian learning and describe the current, improved computational method in more detail. The applied mixture model for mixed real and nominal data is presented along with derivation of the cost function. Details of the grouping algorithm and necessary speedups are also presented. Completely new experiments include an application of IVGA to supervised learning.

E. Alhoniemi is with the Department of Information Technology, University of Turku, Lemminkäisenkatu 14 A, FI-20520 Turku, Finland. (e-mail: esa.alhoniemi@utu.fi)

A. Honkela, K. Lagus, J. Seppä, and P. Wagner are with the Adaptive Informatics Research Centre, Helsinki University of Technology, P.O. Box 5400, FI-02015 TKK, Finland. (e-mail: antti.honkela@tkk.fi, krista.lagus@tkk.fi)

H. Valpola is with the Laboratory of Computational Engineering, Helsinki University of Technology, P.O. Box 9203, FI-02015 TKK, Finland. (e-mail: harri.valpola@tkk.fi)

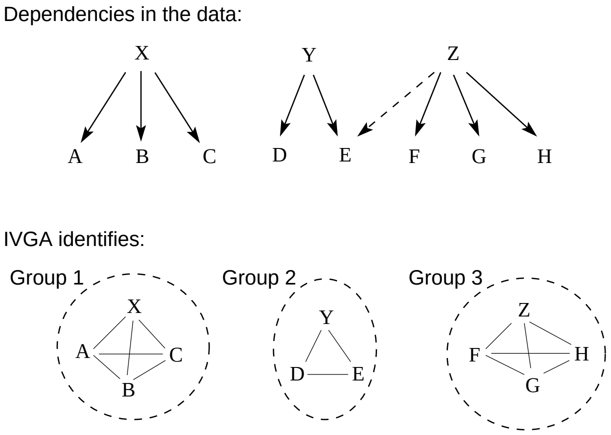


Fig. 1. An illustration of the IVGA principle. The upper part of the figure shows the actual dependencies between the observed variables. The arrows that connect variables indicate causal dependencies. The lower part depicts the variable groups that IVGA might find here. One actual dependency is left unmodeled, namely the one between Z and E. Note that the IVGA does not reveal causalities, but dependencies between the variables only.

II. Independent Variable Group Analysis (IVGA) Principle

The ultimate goal of Independent Variable Group Analysis (IVGA) [1] is to partition a set of variables (also known as attributes or features) into separate groups so that the statistical dependencies of the variables within each group are strong. These dependencies are modeled, whereas the weaker dependencies between variables in different groups are disregarded. The IVGA principle is depicted in Fig. 1.

We wish to emphasize that IVGA should be seen as a *principle*, not an algorithm. However, in order to determine a grouping for observed data, a combinatorial grouping algorithm for the variables is required. Usually this algorithm is heuristic since exhaustive search over all possible variable groupings is computationally infeasible.

The combinatorial optimization algorithm needs to be complemented by a method to score different groupings or a cost function for the groups. Suitable cost functions can be derived in a number of ways, such as using the mutual information between different groups or as the cost of an associated model under a suitable framework such as minimum description length (MDL) or variational Bayes. All of these alternatives are actually approximately equivalent, as presented in Sec. III.

It should be noted that the models used in the model-based approaches need not be of any particular type—as a matter of fact, the models within a particular modeling problem do not necessarily need to be of same type, that is, each variable group could even be modeled using a different model type.

It is vital that the models for the groups are fast to compute and that the grouping algorithm is efficient, too. In Section IV-A, such a heuristic grouping algorithm is presented. Each variable group is modeled by using a computationally relatively light mixture model which is able to model nonlinear dependencies between both nominal and real valued variables at the same time. Variational Bayesian modeling is considered in Section III, which also contains derivation of the mixture model.

A. Motivation for Using IVGA

The computational usefulness of IVGA relies on the fact that if two variables are dependent of each other, representing them together is efficient, since redundant information needs to be stored only once. Conversely, joint representation of variables that do not depend on each other is inefficient. Mathematically speaking, this means that the representation of a joint probability distribution that can be factorized is more compact than the representation a full joint distribution. In terms of a problem expressed using association rules of the form ($A=0.3$, $B=0.9 \Rightarrow F=0.5$, $G=0.1$): the shorter the rules that represent the regularities within a phenomenon, the more compact the representation is and the fewer association rules are needed. IVGA can also be given a biologically inspired motivation: With regard to the structure of the cortex, the difference between a large monolithic model and a set of models produced by the IVGA roughly corresponds to the contrast between full connectivity (all cortical areas receive inputs from all other areas) and more limited, structured connectivity.

The IVGA principle has been shown to be sound: a very simple initial method described in [1] found appropriate variable groups from data where the features were various real-valued properties of natural images. Recently, we have extended the model to handle also nominal (categorical) variables, improved the variable grouping algorithm, and carried out experiments on various different data sets.

The IVGA can be viewed in many different ways. First, it can be seen as a method for finding compact representation of data using multiple independent models. Secondly, IVGA can be seen as a method of clustering variables. Note that it is not equivalent to taking the transpose of the data matrix and performing ordinary clustering, since dependent variables need not be close to each other in the Euclidean or any other common metric. Thirdly, IVGA can also be used as a dimensionality reduction or feature selection method. The review of related methods in Section II-B will concentrate mainly on the first two of these topics.

B. Related Work

One of the basic goals of unsupervised learning is to obtain compact representations for observed data. The methods reviewed in this section are related to IVGA in the sense that they aim at finding a compact representation for a data set using multiple independent models. Such methods include multidimensional independent component analysis (MICA, also known as independent subspace analysis, ISA) [3] and factorial vector quantization (FVQ) [4], [5].

In MICA, the goal is to find independent linear feature subspaces that can be used to reconstruct the data efficiently. Thus each subspace is able to model the linear dependencies in terms of the latent directions defining the subspace. FVQ can be seen as a nonlinear version of MICA, where the component models are vector quantizers over all the variables. The main difference between these and IVGA is that in IVGA, only one model affects a given observed variable. In contrast in the others, all models affect every observed variable. This

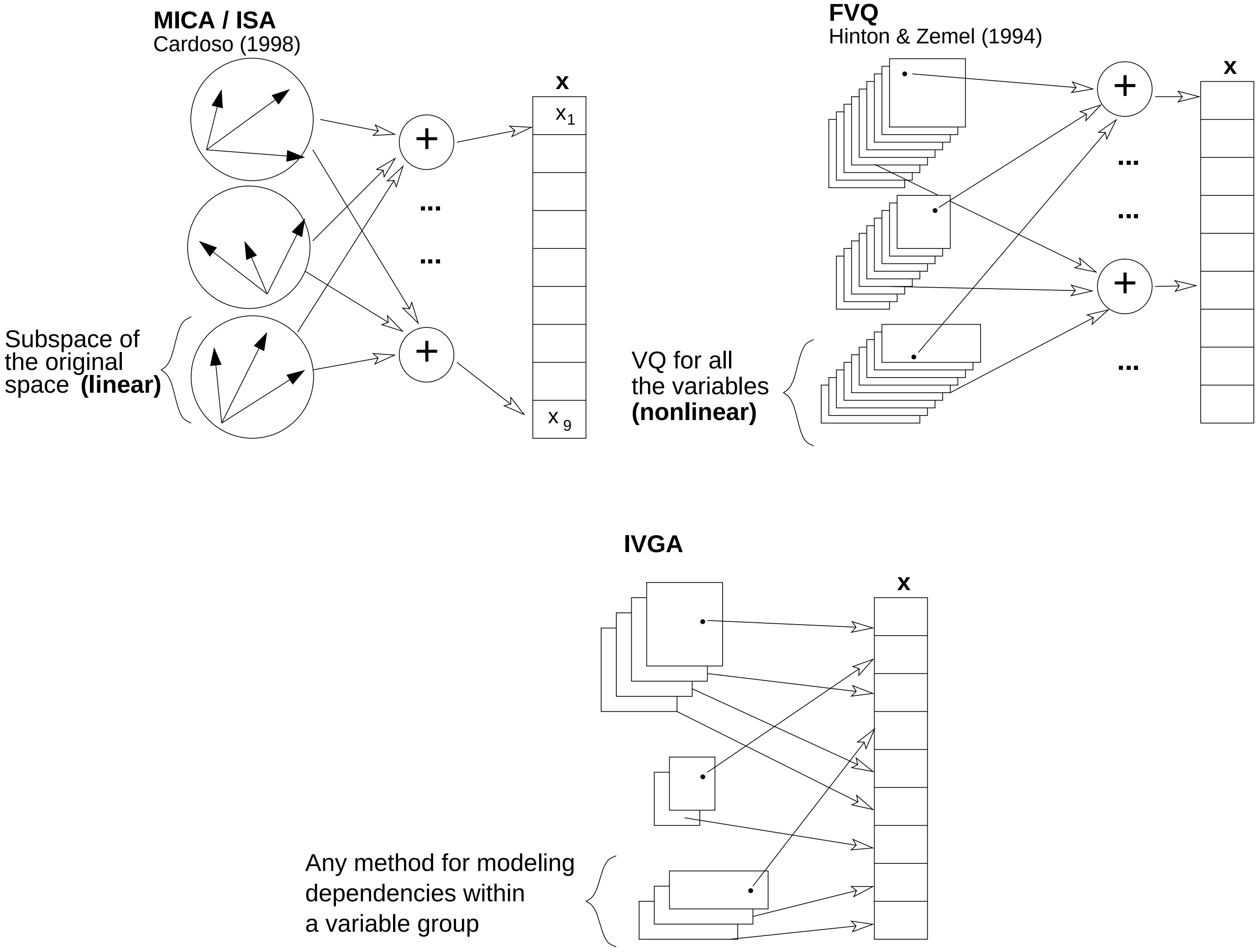


Fig. 2. Schematic illustrations of IVGA and related algorithms, namely MICA/ISA and FVQ that each look for multi-dimensional feature subspaces in effect by maximizing a statistical independence criterion. The input $\mathbf{x}$ is here 9-dimensional. The numbers of squares in FVQ and IVGA denote the numbers of variables modeled in each sub-model, and the numbers of black arrows in MICA the dimensionality of the subspaces. Note that with IVGA the arrows depict all the required connections, whereas with FVQ and MICA only a subset of the actual connections have been drawn (6 out of 27).

difference, visualized in Fig. 2, makes the computation of IVGA significantly more efficient.

There are also a few other methods for grouping the variables based on different criteria. A graph-theoretic partitioning of the graph induced by a thresholded association matrix between variables was used for variable grouping in [6]. The method requires choosing an arbitrary threshold for the associations, but the groupings could nevertheless be used to produce smaller decision trees with equal or better predictive performance than using the full dataset.

A framework for grouping variables of a multivariate time series based on possibly lagged correlations was presented in [7]. The correlations are evaluated using Spearman's rank correlation that can find both linear and monotonic nonlinear dependencies. The grouping method is based on a genetic algorithm, although other possibilities are presented as well. The method seems to be able to find reasonable groupings, but it is restricted to time series data and certain types of dependencies only.

Module networks [8] are a very specific class of models that is based on grouping similar variables together. They are used only for discrete data and all the variables in a group are restricted to have exactly the same distribution. The dependencies between different groups are modeled as a Bayesian network. Sharing the same model within a group makes the model easier to learn from scarce data, but severely restricts its possible uses.

For certain applications, it may be beneficial to view IVGA as a method for clustering variables. In this respect it is related to methods such as double clustering, co-clustering and biclustering which also form a clustering not only for the samples, but for the variables, too [9], [10]. The differences between these clustering methods are illustrated in Fig. 3.

III. A Modeling-Based Approach to Estimating Mutual Information

Estimating mutual information of high dimensional data is very difficult as it requires an estimate of the probability

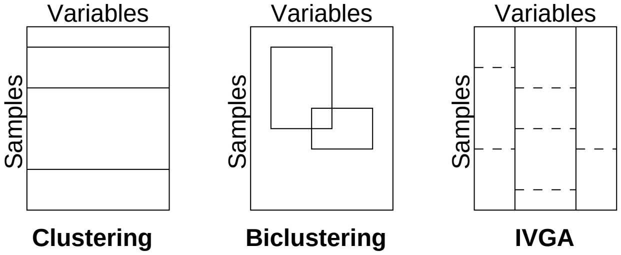


Fig. 3. Schematic illustrations of the IVGA together with regular clustering and biclustering. In biclustering, homogeneous regions of the data matrix are sought for. The regions usually consist of a part of the variables and a part of the samples only. In IVGA, the variables are clustered based on their mutual dependencies. If the individual groups are modeled using mixture models, a secondary clustering of each group is also obtained, as marked by the dashed lines in the rightmost subfigure.

density. We propose solving the problem by using a model-based density estimate. With some additional approximations the problem of minimizing the mutual information reduces to a problem of maximizing the marginal likelihood $p(\mathcal{D}|\mathcal{H})$ of the model. Thus minimization of mutual information is equivalent to finding the best model for the data. This model comparison task can be performed efficiently using variational Bayesian techniques.

A. Approximating the Mutual Information

Let us assume that the data set $\mathcal{D}$ consists of vectors $\mathbf{x}(t), t = 1, \ldots, T$. The vectors are N-dimensional with the individual components denoted by $x_j, j = 1, \ldots, N$. Our aim is to find a partition of $\{1, \ldots, N\}$ to M disjoint sets $\mathcal{G} = \{\mathcal{G}_i | i = 1, \ldots, M\}$ such that the mutual information

$$I_{\mathcal{G}}(\mathbf{x}) = \sum_i H(\{x_j | j \in \mathcal{G}_i\}) - H(\mathbf{x}) \tag{1}$$

between the sets is minimized. In case $M > 2$, this is actually a generalization of mutual information commonly known as multi-information [11]. As the entropy $H(\mathbf{x})$ is constant, this can be achieved by minimizing the first sum. The entropies of that sum can be approximated through

$$\begin{aligned} H(x) &= -\int p(x) \log p(x)\, dx \approx -\frac{1}{T} \sum_{t=1}^{T} \log p(x(t)) \\ &\approx -\frac{1}{T} \sum_{t=1}^{T} \log p(x(t)|x(1), \ldots, x(t-1), \mathcal{H}) \\ &= -\frac{1}{T} \log p(\mathcal{D}|\mathcal{H}). \end{aligned} \tag{2}$$

Two approximations were made in this derivation. First, the expectation over the data distribution was replaced by a discrete sum using the data set as a sample of points from the distribution. Next, the data distribution was replaced by the posterior predictive distribution of the data sample given the past observations. The sequential approximation is necessary to avoid the bias caused by using the same data twice, both for sampling and for fitting the model for the same point. A somewhat similar approximation based on using the probability density estimate implied by a model has been applied for evaluating mutual information also in [12].

Using the result of Eq. (2), minimizing the criterion of Eq. (1) is equivalent to maximizing

$$\mathcal{L} = \sum_i \log p(\{\mathcal{D}_j | j \in \mathcal{G}_i\} | \mathcal{H}_i). \tag{3}$$

This reduces the problem to a standard Bayesian model selection problem. The two problems are, however, not exactly equivalent. The mutual information cost (1) is always minimized when all the variables are in a single group, or multiple statistically independent groups. In case of the Bayesian formulation (3), the global minimum may actually be reached for a nontrivial grouping even if the variables are not exactly independent. This allows determining a suitable number of groups even in realistic situations when there are weak residual dependencies between the groups.

B. Variational Bayesian Learning

Unfortunately evaluating the exact marginal likelihood is intractable for most practical models as it requires evaluating an integral over a potentially high dimensional space of all the model parameters $\boldsymbol{\theta}$. This can be avoided by using a variational method to derive a lower bound of the marginal log-likelihood using Jensen's inequality

$$\begin{aligned} \log p(\mathcal{D}|\mathcal{H}) &= \log \int_{\boldsymbol{\theta}} p(\mathcal{D}, \boldsymbol{\theta}|\mathcal{H})\, d\boldsymbol{\theta} \\ &= \log \int_{\boldsymbol{\theta}} \frac{p(\mathcal{D}, \boldsymbol{\theta}|\mathcal{H})}{q(\boldsymbol{\theta})} q(\boldsymbol{\theta})\, d\boldsymbol{\theta} \\ &\geq \int_{\boldsymbol{\theta}} \log \frac{p(\mathcal{D}, \boldsymbol{\theta}|\mathcal{H})}{q(\boldsymbol{\theta})} q(\boldsymbol{\theta})\, d\boldsymbol{\theta} \end{aligned} \tag{4}$$

where $q(\boldsymbol{\theta})$ is an arbitrary distribution over the parameters. If $q(\boldsymbol{\theta})$ is chosen to be of a suitable simple factorial form, the bound can be rather easily evaluated exactly.

Closer inspection of the right hand side of Eq. (4) shows that it is of the form

$$\begin{aligned} \mathcal{B} &= \int_{\boldsymbol{\theta}} \log \frac{p(\mathcal{D}, \boldsymbol{\theta}|\mathcal{H})}{q(\boldsymbol{\theta})} q(\boldsymbol{\theta})\, d\boldsymbol{\theta} \\ &= \log p(\mathcal{D}|\mathcal{H}) - D_{\mathrm{KL}}(q(\boldsymbol{\theta})||p(\boldsymbol{\theta}|\mathcal{H}, \mathcal{D})), \end{aligned} \tag{5}$$

where $D_{\mathrm{KL}}(q||p)$ is the Kullback–Leibler divergence between distributions q and p. The Kullback–Leibler divergence $D_{\mathrm{KL}}(q||p)$ is non-negative and zero only when $q = p$. Thus it is commonly used as a distance measure between probability distributions although it is not a proper metric [13]. For a more through introduction to variational methods, see for example [14].

In addition to the interpretation as a lower bound of the marginal log-likelihood, the quantity $-\mathcal{B}$ may also be interpreted as a code length required for describing the data using a suitable code [15]. The code lengths can then be used to compare different models, as suggested by the minimum description length (MDL) principle [16]. This provides an alternative justification for the variational method. Additionally, the alternative interpretation can provide more intuitive explanations on why some models provide higher marginal

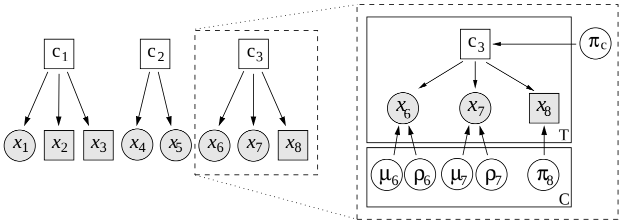


Fig. 4. Our IVGA model as a graphical model. The nodes represent variables of the model with the shaded ones being observed. The left-hand side shows the overall structure of the model with independent groups. The right-hand side shows a more detailed representation of the mixture model of a single group of three variables. Variable c indicates the generating mixture component for each data point. The boxes in the detailed representation indicate that there are T data points and in the rightmost model there are C mixture components representing the data distribution. Rectangular and circular nodes denote discrete and continuous variables, respectively.

likelihoods than others [17]. For the remainder of the paper, the optimization criterion will be the cost function

$$\begin{aligned}\mathcal{C} = -\mathcal{B} &= \int_{\boldsymbol{\theta}} \log \frac{q(\boldsymbol{\theta})}{p(\mathcal{D}, \boldsymbol{\theta}|\mathcal{H})} q(\boldsymbol{\theta}) \; d\boldsymbol{\theta} \\ &= D_{\mathrm{KL}}(q(\boldsymbol{\theta})||p(\boldsymbol{\theta}|\mathcal{H}, \mathcal{D})) - \log p(\mathcal{D}|\mathcal{H})\end{aligned} \tag{6}$$

that is to be minimized.

C. Mixture Model for the Groups

In order to apply the variational Bayesian method described above to solve the IVGA problem, a class of models that benefits from modeling independent variables independently is needed for the groups. In this work mixture models have been used for the purpose. Mixture models are a good choice because they are simple while being able to model also nonlinear dependencies. Our IVGA model is illustrated as a graphical model in Fig. 4.

As shown in Fig. 4, different variables are assumed to be independent within a mixture component and the dependencies only arise from the mixture. For continuous variables, the mixture components are Gaussian and the assumed independence implies a diagonal covariance matrix. Different mixture components can still have different covariances [18]. The applied mixture model closely resembles other well-known models such as soft c-means clustering and soft vector quantization [19].

For nominal variables, the mixture components are multinomial distributions. All parameters of the model have standard conjugate priors. The exact definition of the model and the approximation used for the variational Bayesian approach are presented in Appendix I and the derivation of the cost function in Appendix II.

IV. A Variable Grouping Algorithm for IVGA

The number of possible groupings of n variables is called the nth Bell number B_n. The values of B_n grow with n faster than exponentially, making an exhaustive search of all groupings infeasible. For example, $B_{100} \approx 4.8 \cdot 10^{115}$. Hence, some computationally feasible heuristic — which can naturally be any standard combinatorial optimization algorithm — for finding a good grouping has to be deployed.

In this section, we describe an adaptive heuristic grouping algorithm for determination of the best grouping for the variables which is currently used in our IVGA implementation. After that, we also present three special techniques which are used to speed up the computation.

A. The Algorithm

The goal of the algorithm is to find such a variable grouping and such models for the groups that the total cost over all the models is minimized. The algorithm has an initialization phase and a main loop during which five different operations are consecutively applied to the current models of the variable groups and/or to the grouping until the end condition is met. A flow-chart illustration of the algorithm is shown in Fig. 5 and the phases of the algorithm are explained in more detail below.

Initialization. Each variable is assigned into a group of its own and a model for each group is computed.

Main loop. The following five operations are consecutively used to alter the current grouping and to improve the models of the groups. Each operation of the algorithm is assigned a probability which is adaptively tuned during the main loop: If an operation is efficient in minimizing the total cost of the model, its probability is increased and vice versa.

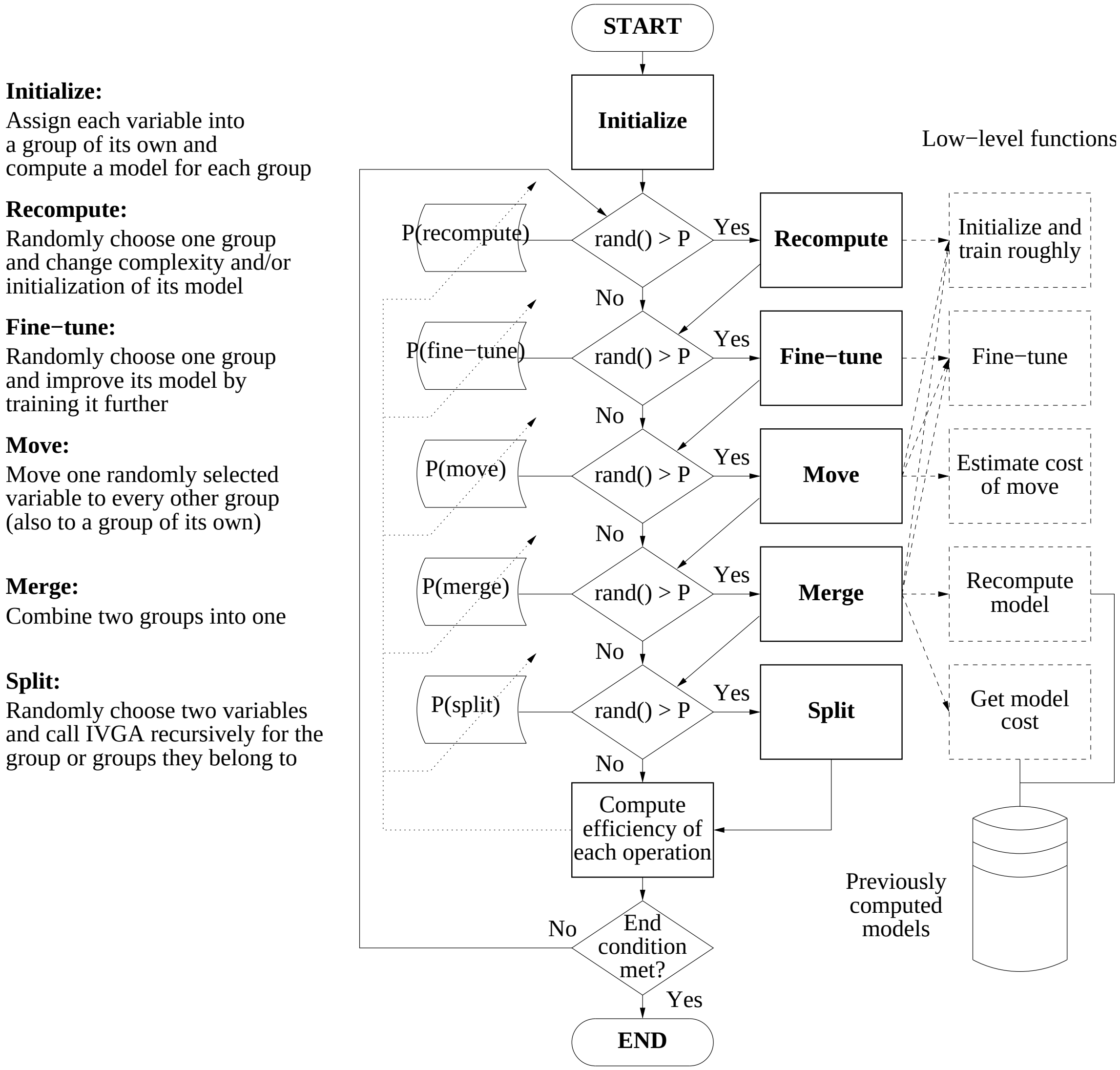
Model recomputation. The purpose of this operation in twofold. (1) It tries to find an appropriate complexity for the model for a group of variables—which is the number of mixture components in the mixture model. (2) It tests different model initializations in order to avoid local minima of the cost function of the model. As the operation is performed multiple times for a group, an appropriate complexity and good initialization is found for the model of the group.

A mixture model for a group is recomputed so that the number of mixture components may decrease, remain the same, or increase. It is slightly more probable that the number of components grows, that is, a more complex model is computed. Next, the components are initialized, for instance in the case of a Gaussian mixture by randomly selecting the centroids among the training data, and the model is roughly trained for some iterations. If a model for the group had been computed earlier, the new model is compared to the old model. The model with the smaller cost is selected as the current model for the group.

Model fine-tuning. When a good model for a group of variables has been found, it is sensible to fine-tune it further so that its cost approaches a local minimum of the cost function. During training, the model cost is never increased due to characteristics of the training algorithm.

However, tuning a model of a group takes many iterations of the learning algorithm and it is not sensible to do that for all the models that are used.

Moving a variable. This operation improves an existing grouping so that a single variable which is in a wrong



Initialize:
Assign each variable into
a group of its own and
compute a model for each group

Recompute:
Randomly choose one group
and change complexity and/or
initialization of its model

Fine−tune:
Randomly choose one group
and improve its model by
training it further

Move:
Move one randomly selected
variable to every other group
(also to a group of its own)

Merge:
Combine two groups into one

Split:
Randomly choose two variables
and call IVGA recursively for the
group or groups they belong to

Fig. 5. An illustration of the variable grouping algorithm for IVGA. The solid line describes control flow, the dashed lines denote low-level subroutines and their calls so that the arrow points to the called routine. The dotted line indicates adaptation of the probabilities of the five operations. Function rand() produces a random number on the interval [0,1].

group is moved to a more appropriate group. First, one variable is randomly selected among all the variables of all groups. The variable is removed from its original group and moved to every other group (also to a group of its own) at a time. For each new group candidate, the cost of the model is roughly estimated. If the move reduces the total cost compared to the original one, the variable is moved to a group which yields the highest decrease in the total cost.

Merge. The goal of the merge operation is to combine two groups in which the variables are mutually dependent. In the operation, two groups are selected randomly among the current groups. A model for the variables of their union is computed. If the cost of the model of the joint group is smaller than the sum of the costs of the two original groups, the two groups are merged. Otherwise, the two original groups are retained.

Split. The split operation breaks down one or two existing groups. The group(s) are chosen so that two variables are randomly selected among all the variables. The group(s) corresponding to the variables are then taken for the operation. Hence, the probability of a group to be selected is proportional to the size of the group. As a result, more likely heterogeneous large groups are chosen more frequently than smaller ones. The operation recursively calls the algorithm for the union of the selected groups. If the total cost of the resulting models is less than the sum of the costs of the original group(s), the original group(s) are replaced by the new grouping. Otherwise, the original group(s) are retained.

End condition. Iteration is stopped if the decrease of the total cost is very small in several successive iterations.

B. Speedup Techniques Used in Computation of the Models

Computation of an IVGA model for a large set of variables requires computation of a huge number of models (say, thousands), because in order to determine the cost of an arbitrary variable group, a unique model for it needs to be computed (or, at least, an approximation of the cost of the model). Therefore, fast and efficient computation of models is crucial. We use the following three special techniques are used in order to speed up the computation of the models.

1) Adaptive Tuning of Operation Probabilities: During the main loop algorithm described above, five operations are used to improve the grouping and the models. Each operation has a probability which dictates how often the corresponding operation is performed (see Fig. 5). As the grouping algorithm is run for many iterations, the probabilities are slowly adapted instead of keeping them fixed because

- it is difficult to determine probabilities which are appropriate for an arbitrary data set; and
- during a run of the algorithm, the efficiency of different operations varies—for example, the split operation is seldom beneficial in the beginning of the iteration (when the groups are small), but it becomes more useful when the sizes of the groups tend to grow.

The adaptation is carried out by measuring the efficiency (in terms of reduction of the total cost of all the models) of each operation. The probabilities of the operations are gradually adapted so that the probability of an efficient operation is increased and the probability of an inefficient operation decreased. The adaptation is based on low-pass filtered efficiency, which is defined by

$$\text{efficiency} = -\frac{\Delta\mathcal{C}}{\Delta t} \tag{7}$$

where $\Delta\mathcal{C}$ is the change in the total cost and Δt is the amount of CPU time used for the operation.

Based on multiple tests (not shown here) using various data sets, it has turned out that adaptation of the operation probabilities instead of keeping them fixed significantly speeds up the convergence of the algorithm into a final grouping.

2) "Compression" of the Models: Once a model for a variable group is computed, it is sensible to be stored, because it is a previously computed good model for a certain variable group may be later needed.

Computation of many models—for example, a mixture model—is stochastic, because often a model is initialized randomly and trained for a number of iterations. However, computation of such a model is actually *deterministic* provided that the state of the (deterministic) pseudorandom number generator when the model was initialized is known. Thus, in order to reconstruct a model after it has been once computed, we need to store (i) the random seed, (ii) the number of iterations that were used to train the model, and (iii) the model structure. Additionally, it is also sensible to store (iv) the cost of the model. So, a mixture model can be compressed into two floating point numbers (the random seed and the cost of the model) and two integers (the number of training iterations and the number of mixture components).

Note that this model compression principle is completely general: it can be applied in any algorithm in which compression of multiple models is required.

3) Fast Estimation of Model Costs When Moving a Variable: When the move of a variable from one group to all the other groups is attempted, computationally expensive evaluation of the costs of multiple models is required. We use a specialized speedup technique for fast approximation of the costs of the groups: Before moving a variable to another group for real, a quick pessimistic estimate of the total cost change of the move is calculated, and only those new models that look appealing are tested further.

When calculating the quick estimate for the cost change if a variable is moved from one to another, the posterior probabilities of the mixture components are fixed and only the parameters of the components related to the moved variable are changed. The cost of these two groups is then calculated for comparison with their previous cost. The approximation can be justified by the fact that if a variable is highly dependent on the variables in a group, then the same mixture model should fit it as well.

V. Applications, Experiments

Problems in which IVGA can be found to be useful can be divided into the following categories. First, IVGA can be used for *confirmatory* purposes in order to verify human intuition of an existing grouping of variables. The first synthetic problem presented in Section V-A can be seen as an example of this type. Second, IVGA can be used to *explore* observed data, that is, to make hypotheses or learn the structure of the data. The discovered structure can then be used to divide a complex modeling problem into a set of simpler ones as illustrated in Section V-B. Third, if we are dealing with a classification problem, we can use IVGA to reveal the variables that are dependent with the class variable. In other words, we can use IVGA also for *variable selection* or *dimension reduction* in supervised learning problems. This is illustrated in Section V-C.

A. Toy Example

In order to illustrate our IVGA algorithm using a simple and easily understandable example, a data set consisting of one thousand points in a four-dimensional space was synthesized. The dimensions of the data are called $education$, $income$, $height$, and $weight$. All the variables are real and the units are arbitrary. The data was generated from a distribution in which both education and income are statistically independent of height and weight.

Fig. 6 shows plots of education versus income, height vs. weight, and for comparison a plot of education vs. height. One may observe, that in the subspaces of the first two plots of Fig. 6, the data points lie in few, more concentrated clusters and thus can generally be described (modeled) with a lower cost in comparison to the third plot. As expected, when the data was given to our IVGA model, the resulting grouping was

$$\{\{education, income\}, \{height, weight\}\}.$$

Table I compares the costs of some possible groupings.

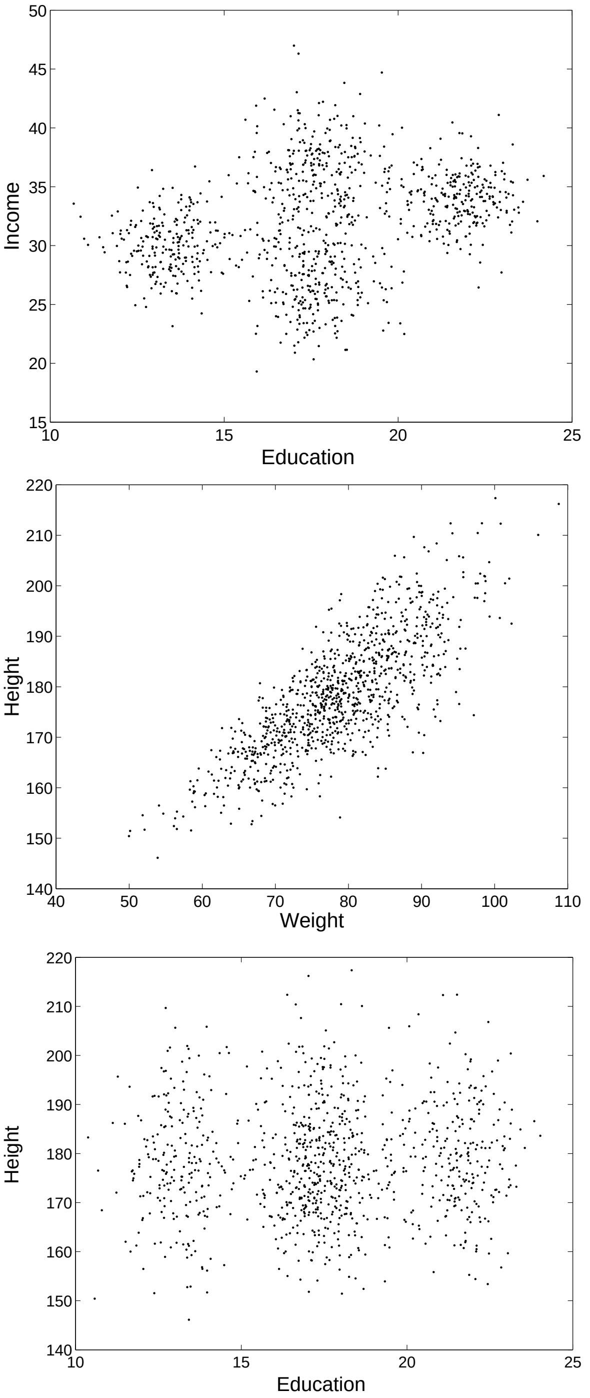


Fig. 6. Comparison of different two-dimensional subspaces of the data. Due to the dependencies between the variables shown in the first two pictures it is useful to model those variables together. In contrast, in the last picture no such dependency is observed and therefore no benefit is obtained from modeling the variables together.

B. Printed Circuit Board Assembly

In the second experiment, we constructed predictive models to support and speed up user input of component data of a printed circuit board assembly robot. When a robot is used in the assembly of a new product which contains components that have not been previously used by the robot, the data of the new components need to be manually determined and added to the existing component database of the robot by a human operator. The component data can be seen as a matrix. Each row of the matrix contains attribute values of one component and the columns of the matrix depict component attributes, which are not mutually independent. Building an input support system by modeling of the dependencies of the existing data using association rules has been considered in [20]. A major problem of the approach is that extraction of the rules is computationally heavy, and memory consumption of the predictive model which contains the rules (in our case, a trie) is very high.

Grouping	Total Cost	Parameters
{e,i,h,w}	12233.4	288
{e,i}{h,w}	**12081.0**	80
{e}{i}{h}{w}	12736.7	24
{e,h}{i}{w}	12739.9	24
{e,i}{h}{w}	12523.9	40
{e}{i}{h,w}	12304.0	56

TABLE I

A comparison of the total costs of some variable groupings of the synthetic data. The variables education, income, height, and weight are denoted here by their initial letters. Also shown is the number of real numbers required to parameterize the learned optimal Gaussian mixture component distributions. The total costs are for mixture models optimized carefully using our IVGA algorithm. The model search of our IVGA algorithm was able to discover the best grouping, that is, the one with the smallest cost.

We divided the component data of an operational assembly robot (5 016 components, 22 nominal attributes) into a training set (80 % of the whole data) and and a testing set (the rest 20 %). The IVGA algorithm was run 200 times for the training set. In the first 100 runs (avg. cost 113 003), all the attributes were always assigned into one group. During the last 100 runs (avg. cost 113 138) we disabled the adaptation of the probabilities (see Section IV-A) to see if this would have an effect on the resulting groupings. In these runs, we obtained 75 groupings with 1 group and 25 groupings with 2–4 groups. Because we were looking for a good grouping with more than one group, we chose a grouping with 2 groups (7 and 15 attributes). The cost of this grouping was 112 387 which was not the best among all the results over 200 runs (111 791), but not very far from it.

Next, the dependencies of (1) the whole data and (2) the 2 variable groups were modeled using association rules. The large sets required for computation of the rules were computed using a freely available software implementation[1] of the Eclat algorithm [21]. Computation of the rules requires two parameters: minimum support ("generality" of the large sets that the rules are based on) and minimum confidence ("accuracy" of the rule). The minimum support dictates the number of large sets, which is in our case equal to the size of the model. For the whole data set, the minimum support was 5 %, which was the smallest computationally feasible value

[1] See `http://www.adrem.ua.ac.be/~goethals/software/index.html`

in terms of memory consumption. For the models of the two groups it set to 0.1 %, which was the smallest as possible value so that the combined size of the two models did not exceed the size of the model for the whole data. The minimum confidence was set to 90 %, which is a typical value for the parameter in many applications.

The rules were used for one-step prediction of the attribute values of the testing data. The data consisted of values selected and verified by human operators, but it is possible that these are not the only valid values. Nevertheless, predictions were ruled incorrect if they differed from these values. Computation times, memory consumption, and prediction accuracy for the whole data and the grouped data are shown in Table II. Grouping of the data both accelerated computation of the rules and improved the prediction accuracy. Also note that the combined size of the models of the two groups is only about 1/4 of the corresponding model for the whole data.

	Whole data	**Grouped data**
Computation time (s)	48	9.1
Size of trie (nodes)	9 863 698	2 707 168
Correct predictions (%)	57.5	63.8
Incorrect predictions (%)	3.7	2.9
Missing predictions (%)	38.8	33.3

TABLE II

Summary of the results of the component data experiment. All the quantities for the grouped data are sums over the two groups. Also note that the size of trie is in this particular application the same as the number of association rules.

The potential benefits of the IVGA in an application of this type are as follows. (1) It is possible to compute rules which yield better prediction results, because the rules are based on small amounts of data, i.e, it is possible to use smaller minimum support for the grouped data. (2) Discretization of continuous variables—which is often a problem in applications of association rules—is automatically carried out by the mixture model. (3) Computation of the association rules may even be completely ignored by using the mixture models of the groups as a basis for the predictions. Of these, (1) was demonstrated in the experiment whereas (2) and (3) remain a topic for future research.

C. Feature Selection for Supervised Learning: Ionosphere Data

In this experiment, we investigated whether the variable grouping ability could be used for feature selection for classification. One way to apply our IVGA model in this manner is to see which variables IVGA groups together with the class variable, and to use only these in the actual classifier.

We ran our IVGA algorithm 10 times for the the Ionosphere data set [22], which contains 351 instances of radar measurements consisting of 34 attributes and a binary class variable. From the three groupings (runs) with the lowest cost, each variable that was grouped with the class variable at least once was included in the classification experiment. As a result, the following three features were chosen: $\{1, 5, 7\}$.

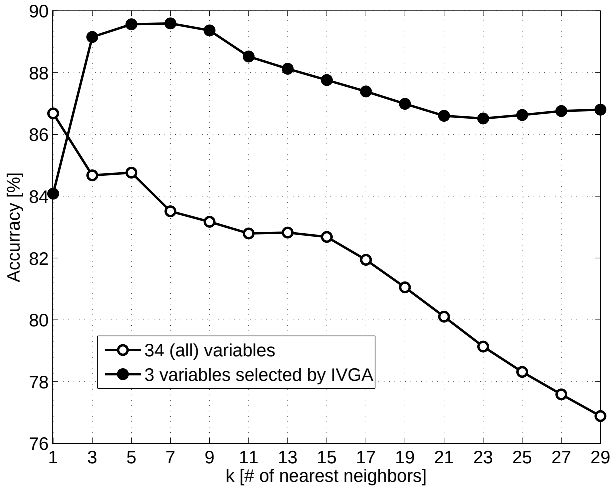


Fig. 7. Classification accuracies for the Ionosphere data using k-NN classifier with all the variables (white markers) and with only the variables selected using IVGA (black markers).

The classification was carried out using the k-nearest-neighbor (k-NN) classifier. Out of the 351 samples 51 were used for testing and the rest for training. In each experiment, the testing and the training data sets were randomly drawn from the entire data set and normalized prior to classification. The averaged results of 1 000 different runs are shown in Fig. 7 with various (odd) values for k. For comparison, the same experiment was carried out using all the 34 variables. As can be seen, the set of three features chosen using IVGA produces clearly better classification accuracy than the complete set of features whenever $k > 1$. For example, for $k = 5$ the accuracy using IVGA was 89.6 % while for the complete set of features it was 84.8 %.

Extensive benchmarking experiments using the Ionosphere data set that compare PCA and Random Projection for dimensionality reduction with a number of classifiers are reported in [23]. They also report accuracy in the original input space for each method. For 1-NN this value is 86.7 %, with 5-NN 84.5 %, and with a linear SVM classifier 87.8 %. The best result obtained using dimension reduction was 88.7 %. We used an identical test setting in our experiments with the difference that feature selection was performed using IVGA. Using the k-NN classifier we obtained better accuracies than any of the classifiers used in [23], including linear SVM, when they were performed in the original input space. Moreover, IVGA was able to improve somewhat even upon the best results that they obtained in the reduced-dimensional spaces. We also tested nonlinear SVM using Gaussian kernel by using the same software[2] with default settings that was used in [23]. For the entire data set the prediction accuracy was weak, only 66.1 % whereas using the three variables selected by IVGA it was the best among all the results in the experiment, 90.7 %.

A number of heuristic approaches to feature selection like forward, backward, and floating search methods (see e.g. [25]) exist and could have been used here as well. However, the goal

[2] See [24] and `http://svmlight.joachims.org/`.

of the experiment was not to find the best set of features but to demonstrate that the IVGA can reveal useful structure of the data.

VI. Discussion

Many real-world problems and data sets can be divided into smaller relatively independent subproblems. Automatic discovery of such divisions can significantly help in applying different machine learning techniques to the data by reducing computational and memory requirements of processing. The IVGA principle calls for finding the divisions by partitioning the observed variables into separate groups so that the mutual dependencies between variables within a group are strong whereas mutual dependencies between variables in different groups are weaker.

In this paper, the IVGA principle has been implemented by a method that groups the input variables only. In the end, there may also exist interesting dependencies between the individual variable groups. One avenue for future research is to extend the grouping model into a hierarchical IVGA that is able to model the residual dependencies between the groups of variables.

From the perspective of using the method it would be useful to implement many different model types including also linear models. This would allow the modeling of each variable group with the best model type for that particular sub-problem, and depending on the types of dependencies within the problem. Such extensions naturally require the derivation of a cost function for each additional model family, but there are simple tools for automating this process [26], [27].

The stochastic nature of the grouping algorithm makes its computational complexity difficult to analyze. Empirically the complexity of convergence to a neighborhood of a locally optimal grouping seems to be roughly quadratic with respect to both the number of variables and the number of data samples. In case of number of samples this is because the data does not exactly follow the mixture model and thus more mixture components are used when there are more samples. Convergence to the exact local optimum typically takes significantly longer, but it is usually not necessary as even nearly optimal results are often good enough in practice.

Although the presented IVGA model appears quite simple, several computational speedup techniques are needed for it to work efficiently enough. Some of these may be of interest in themselves, irrespective of the IVGA principle. In particular worth mentioning are the adaptive tuning of operation probabilities in the grouping algorithm (Sec. IV-B.1) as well as the model compression principle (Sec. IV-B.2).

By providing the source code of the method for public use we invite others both to use the method and to contribute to extending it. A MATLAB package of our IVGA implementation is available at `http://www.cis.hut.fi/projects/ivga/`.

VII. Conclusion

In this paper, we have presented the independent variable group analysis (IVGA) principle and a method for modeling data through mutually independent groups of variables. The approach has been shown to be useful in real-world problems: It decreases computational burden of other machine learning methods and also increases their accuracy by letting them concentrate on the essential dependencies of the data.

The general nature of the IVGA principle allows many potential applications. The method can be viewed as a tool for compact modeling of data, an algorithm for clustering variables, or as a tool for dimensionality reduction and feature selection. All these interpretations allow for several practical applications.

Biclustering – clustering of both variables and samples – is very popular in bioinformatics. In such applications it could be useful to ease the strict grouping of the variables of IVGA. This could be accomplished by allowing different partitions in different parts of the data set using, for instance, a mixture-of-IVGAs type of model. Hierarchical modeling of residual dependencies between the groups would be another interesting extension.

Appendix I
Specification of the Mixture Model

A mixture model for the random variable $\mathbf{x}(t)$ can be written with the help of an auxiliary variable $c(t)$ denoting the index of the active mixture component as illustrated in the right part of Fig. 4. In our IVGA model, the mixture model for the variable groups is chosen to be as simple as possible for computational reasons. This is done by restricting the components $p(\mathbf{x}(t)|\theta_i, \mathcal{H})$ of the mixture to be such that different variables are assumed independent. This yields

$$\begin{aligned} p(\mathbf{x}(t)|\mathcal{H}) &= \sum_i p(\mathbf{x}(t)|\theta_i, \mathcal{H}) p(c(t) = i) \\ &= \sum_i p(c(t) = i) \prod_j p(x_j(t)|\theta_{i,j}, \mathcal{H}), \end{aligned} \tag{8}$$

where $\theta_{i,j}$ are the parameters of the ith mixture component for the jth variable. Dependencies between the variables are modeled only through the mixture. The variable c has a multinomial distribution with parameters $\boldsymbol{\pi}_c$ that have a Dirichlet prior with parameters $\boldsymbol{u}_c$

$$p(c(t)|\boldsymbol{\pi}_c, \mathcal{H}) = \mathrm{Multinom}(c(t);\ \boldsymbol{\pi}_c) \tag{9}$$

$$p(\boldsymbol{\pi}_c|\mathbf{u}_c, \mathcal{H}) = \mathrm{Dirichlet}(\boldsymbol{\pi}_c;\ \mathbf{u}_c). \tag{10}$$

The use of a mixture model allows for both categorical and continuous variables. For continuous variables the mixture is a heteroscedastic Gaussian mixture, that is, all mixture components have their own precisions. Thus

$$p(x_j(t)|\theta_{i,j}, \mathcal{H}) = N(x_j(t);\ \mu_{i,j}, \rho_{i,j}), \tag{11}$$

where $\mu_{i,j}$ is the mean and $\rho_{i,j}$ is the precision of the Gaussian. The parameters $\mu_{i,j}$ and $\rho_{i,j}$ have hierarchical priors

$$p(\mu_{i,j}|\mu_{\mu_j}, \rho_{\mu_j}, \mathcal{H}) = N(\mu_{i,j};\ \mu_{\mu_j}, \rho_{\mu_j}) \tag{12}$$

$$p(\rho_{i,j}|\alpha_{\rho_j}, \beta_{\rho_j}, \mathcal{H}) = \mathrm{Gamma}(\rho_{i,j};\ \alpha_{\rho_j}, \beta_{\rho_j}). \tag{13}$$

For categorical variables, the mixture is a simple mixture of multinomial distributions so that

$$p(x_j(t)|\theta_{i,j}, \mathcal{H}) = \mathrm{Multinom}(x_j(t);\ \boldsymbol{\pi}_{i,j}). \tag{14}$$

The probabilities $\boldsymbol{\pi}_{i,j}$ have a Dirichlet prior

$$p(\boldsymbol{\pi}_{i,j}|\mathbf{u}_j,\mathcal{H}) = \mathrm{Dirichlet}(\boldsymbol{\pi}_{i,j};\ \mathbf{u}_j). \tag{15}$$

Combining these yields the joint probability of all parameters (here $\mathbf{c} = [c(1),\ldots,c(T)]^T$):

$$\begin{aligned} p(\mathcal{D},\mathbf{c},\boldsymbol{\pi}_c,\boldsymbol{\pi},\boldsymbol{\mu},\boldsymbol{\rho}) = \prod_t \Big[p(c(t)|\boldsymbol{\pi}_{\boldsymbol{c}})\Big] p(\boldsymbol{\pi}_c|\mathbf{u}_c) \\ \prod_i \Bigg[\prod_{j:x_j \text{ categorical}} \Big[p(\boldsymbol{\pi}_{i,j}|\mathbf{u}_j)\Big] \\ \prod_{j:x_j \text{ continuous}} \Big[p(\mu_{i,j}|\mu_{\mu_j},\rho_{\mu_j})p(\rho_{i,j}|\alpha_{\rho_j},\beta_{\rho_j})\Big]\Bigg] \\ \prod_t \Bigg[\prod_{j:x_j \text{ categorical}} p(x_j(t)|c(t),\boldsymbol{\pi}_{\cdot,j}) \\ \prod_{j:x_j \text{ continuous}} p(x_j(t)|c(t),\mu_{\cdot,j},\rho_{\cdot,j})\Bigg] \end{aligned} \tag{16}$$

All the component distributions of this expression have been introduced above in Eqs. (11)-(15).

The corresponding variational approximation is

$$\begin{aligned} q(\mathbf{c},\boldsymbol{\pi}_c,\boldsymbol{\pi},\boldsymbol{\mu},\boldsymbol{\rho}) = q(\mathbf{c})q(\boldsymbol{\pi}_c)q(\boldsymbol{\pi})q(\boldsymbol{\mu})q(\boldsymbol{\rho}) = \\ \prod_t \Big[q(c(t)|\mathbf{w}(t))\Big] q(\boldsymbol{\pi}_c|\hat{\mathbf{u}}_c) \\ \prod_i \Bigg[\prod_{j:x_j \text{ categorical}} \Big[q(\boldsymbol{\pi}_{i,j}|\hat{\mathbf{u}}_{i,j})\Big] \\ \prod_{j:x_j \text{ continuous}} \Big[q(\mu_{i,j}|\hat{\mu}_{\mu_{i,j}},\hat{\rho}_{\mu_{i,j}})q(\rho_{i,j}|\hat{\alpha}_{\rho_{i,j}},\hat{\beta}_{\rho_{i,j}})\Big]\Bigg] \end{aligned} \tag{17}$$

with the factors

$$q(c(t)) = \mathrm{Multinom}(c(t);\ \mathbf{w}(t)) \tag{18}$$
$$q(\boldsymbol{\pi}_c) = \mathrm{Dirichlet}(\boldsymbol{\pi}_c;\ \hat{\mathbf{u}}_c) \tag{19}$$
$$q(\boldsymbol{\pi}_{i,j}) = \mathrm{Dirichlet}(\boldsymbol{\pi}_{i,j};\ \hat{\mathbf{u}}_{i,j}) \tag{20}$$
$$q(\mu_{i,j}) = N(\mu_{i,j};\ \hat{\mu}_{\mu_{i,j}},\hat{\mu}_{\rho_{i,j}}) \tag{21}$$
$$q(\rho_{i,j}) = \mathrm{Gamma}(\rho_{i,j};\ \hat{\alpha}_{\rho_{i,j}},\hat{\beta}_{\rho_{i,j}}). \tag{22}$$

Because of the conjugacy of the model, these are optimal forms for the components of the approximation, given the factorization. Specification of the approximation allows the evaluation of the cost of Eq. (6) and the derivation of update rules for the parameters as shown below in Appendix II. The hyperparameters $\mu_{\mu_j},\rho_{\mu_j},\alpha_{\rho_j},\beta_{\rho_j}$ are updated using maximum likelihood estimation. The parameters of the fixed Dirichlet priors are set to values corresponding to the Jeffreys prior.

Appendix II
Derivation of the Cost Function and Update Rules

The cost function of Eq. (6) can be expressed, using $\langle\cdot\rangle$ to denote expectation over q, as

$$\begin{aligned} \Big\langle \log \frac{q(\boldsymbol{\theta})}{p(\mathcal{D},\boldsymbol{\theta}|\mathcal{H})} \Big\rangle \\ = \big\langle \log q(\boldsymbol{\theta}) - \log p(\boldsymbol{\theta})\big\rangle - \big\langle \log p(\mathcal{D}|\boldsymbol{\theta})\big\rangle \end{aligned} \tag{23}$$

Now, being expected logarithms of products of probability distributions over the factorial posterior approximation q, the terms easily split further. The terms of cost function are presented as the costs of the different parameters and the likelihood term. Some of the notation used in the formulae is introduced in Table III.

Symbol	Explanation
C	Number of mixture components
T	Number of data points
D_{cont}	Number of continuous dimensions
S_j	The number of categories in nominal dimension j
u_0	The sum over the parameters of a Dirichlet distribution
$I_k(x)$	An indicator for x being of category k
Γ	The gamma function (not the distribution pdf)
Ψ	The digamma function, that is $\Psi(x) = \frac{d}{dx}\ln(\Gamma(x))$
$w_i(t)$	The multinomial probability/weight of the ith mixture component in the $\mathbf{w}(t)$ of data point t

TABLE III

Notation

A. Terms of the Cost Function

$$\begin{aligned} \big\langle \log q(\mathbf{c}|\mathbf{w}) - \log p(\mathbf{c}|\boldsymbol{\pi}_{\boldsymbol{c}})\big\rangle = \\ \sum_{t=1}^{T}\sum_{i=1}^{C} w_i(t)\big[\log w_i(t) - [\Psi(\hat{u}_{c_i}) - \Psi(\hat{u}_{c0})]\big] \end{aligned} \tag{24}$$

$$\begin{aligned} \big\langle \log q(\boldsymbol{\pi}_{\mathbf{c}}|\hat{\mathbf{u}}_{\mathbf{c}}) - \log p(\boldsymbol{\pi}_{\mathbf{c}}|\mathbf{u}_{\mathbf{c}})\big\rangle = \\ \sum_{i=1}^{C}\Big[(\hat{u}_{c_i} - u_c)[\Psi(\hat{u}_{c_i}) - \Psi(\hat{u}_{c0})] - \log\Gamma(\hat{u}_{c_i})\Big] \\ + \log\Gamma(\hat{u}_{c0}) - \log\Gamma(u_{c0}) + C\log\Gamma(u_c) \end{aligned} \tag{25}$$

$$\begin{aligned} \big\langle \log q(\boldsymbol{\pi}|\hat{\mathbf{u}}) - \log p(\boldsymbol{\pi}|\mathbf{u})\big\rangle = \\ \sum_{j:x_j \text{ categorical}} \Bigg[\sum_{i=1}^{C}\sum_{k=1}^{S_j}\Big[(\hat{u}_{i,j,k} - u_{j,k})[\Psi(\hat{u}_{i,j,k}) - \Psi(\hat{u}_{0_{i,j}})]\Big] \\ + \sum_{i=1}^{C}\Big[\log\Gamma\big(\hat{u}_{0_{i,j}}\big) - \sum_{k=1}^{S_j}\log\Gamma\big(\hat{u}_{i,j,k}\big)\Big] \\ + C\Big[-\log\Gamma(u_{0_j}) + \sum_{k=1}^{S_j}\log\Gamma(u_{j,k})\Big]\Bigg] \end{aligned} \tag{26}$$

$$\begin{aligned}
&\Big\langle \log q(\boldsymbol{\mu}|\hat{\boldsymbol{\mu}}_{\boldsymbol{\mu}}, \hat{\boldsymbol{\rho}}_{\boldsymbol{\mu}}) - \log p(\boldsymbol{\mu}|\boldsymbol{\mu}_{\boldsymbol{\mu}}, \boldsymbol{\rho}_{\boldsymbol{\mu}})\Big\rangle = \\
&\qquad - \frac{CD_{\text{cont}}}{2} + \sum_{j:x_j \text{ continuous}} \sum_{i=1}^{C} \Big[\log \frac{\hat{\rho}_{\mu_{i,j}}}{2\rho_{\mu_j}} \\
&\qquad\qquad + \frac{\rho_{\mu_j}}{2}\big[\hat{\rho}_{\mu_{i,j}}^{-1} + (\hat{\mu}_{\mu_{i,j}} - \mu_{\mu_j})^2\big]\Big]
\end{aligned} \tag{27}$$

$$\begin{aligned}
&\Big\langle \log q(\boldsymbol{\rho}|\hat{\boldsymbol{\alpha}}_{\boldsymbol{\rho}}, \hat{\boldsymbol{\beta}}_{\boldsymbol{\rho}}) - \log p(\boldsymbol{\rho}|\boldsymbol{\alpha}_{\boldsymbol{\rho}}, \boldsymbol{\beta}_{\boldsymbol{\rho}})\Big\rangle = \\
&\qquad \sum_{j:x_j \text{ continuous}} \sum_{i=1}^{C} \Big[\log\Gamma(\alpha_{\rho_j}) - \log\Gamma(\hat{\alpha}_{\rho_{i,j}}) \\
&\qquad\qquad + \hat{\alpha}_{\rho_{i,j}} \log \hat{\beta}_{\rho_{i,j}} - \alpha_{\rho_j} \log \beta_{\rho_j} \\
&+ (\hat{\alpha}_{\rho_{i,j}} - \alpha_{\rho_j})\big(\Psi(\hat{\alpha}_{\rho_{i,j}}) - \log \hat{\beta}_{\rho_{i,j}}\big) + \frac{\hat{\alpha}_{\rho_{i,j}}}{\hat{\beta}_{\rho_{i,j}}}(\beta_{\rho_j} - \hat{\beta}_{\rho_{i,j}})\Big]
\end{aligned} \tag{28}$$

$$\begin{aligned}
&\Big\langle - \log p(\mathcal{D}|\mathbf{c}, \boldsymbol{\pi}_{\boldsymbol{c}}, \boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\rho})\Big\rangle = \frac{T\log(2\pi)D_{\text{cont}}}{2} \\
&+ \sum_{t=1}^{T}\sum_{i=1}^{C}\Bigg\{ w_i(t) \Bigg[- \sum_{j:x_j \text{ categorical}} \Big[\Psi(\hat{u}_{i,j,x_j(t)}) - \Psi(\hat{u}_{0_{i,j}})\Big] \\
&\qquad + \frac{1}{2} \sum_{j:x_j \text{ continuous}} \Big[\frac{\hat{\alpha}_{\rho_{i,j}}}{\hat{\beta}_{\rho_{i,j}}}\big(\hat{\rho}_{\mu_{i,j}}^{-1} + (x_j(t) - \hat{\mu}_{\mu_{i,j}})^2\big) \\
&\qquad\qquad - \big(\Psi(\hat{\alpha}_{\rho_{i,j}}) - \log \hat{\beta}_{\rho_{i,j}}\big)\Big]\Bigg]\Bigg\}
\end{aligned} \tag{29}$$

B. On the Iteration Formulae and Initialization

The iteration formulae for one full iteration of mixture model adaptation consist of simple coordinate-wise re-estimations of the parameters. This is like expectation-maximization (EM) iteration. The update rules of the hyperparameters μ_{μ_j}, ρ_{μ_j}, α_{ρ_j} and β_{ρ_j} are based on maximum likelihood estimation.

Before the iteration the mixture components are initialized using the dataset and a pseudorandom seed number that is used to make the initialization stochastic but reproducible using the same random seed. The mixture components are initialized as equiprobable.

C. The Iteration Formulae

One full iteration cycle:

1) Update $\mathbf{w}$

$$\begin{aligned}
w_i^*(t) \leftarrow &\exp\Bigg(\Psi(\hat{u}_{c_i}) + \\
&\sum_{j:x_j \text{ categorical}} \Big[\Psi(\hat{u}_{i,j,x_j(t)}) - \Psi(\hat{u}_{0_{i,j}})\Big] \\
&- \frac{1}{2} \sum_{j:x_j \text{ continuous}} \Big[\\
&\frac{\hat{\alpha}_{\rho_{i,j}}}{\hat{\beta}_{\rho_{i,j}}}\big(\hat{\rho}_{\mu_{i,j}}^{-1} + (x_j(t) - \hat{\mu}_{\mu_{i,j}})^2\big) \\
&- \big(\Psi(\hat{\alpha}_{\rho_{i,j}}) - \log \hat{\beta}_{\rho_{i,j}}\big)\Big]\Bigg) \\
w_i(t) \leftarrow &\frac{w_i^*(t)}{\sum_{i'=1}^{C} w_{i'}^*(t)}
\end{aligned} \tag{30}$$

2) Update $\hat{\mathbf{u}}_{\mathbf{c}}$

$$\hat{u}_{c_i} \leftarrow u_c + \sum_{t=1}^{T} w_i(t) \tag{31}$$

3) Update categorical dimensions of the mixture components

$$\hat{u}_{i,j,k} \leftarrow u_{j,k} + \sum_{t=1}^{T} w_i(t) I_k(x_j(t)) \tag{32}$$

4) Update continuous dimensions of the mixture components

$$\hat{\mu}_{\mu_{i,j}} \leftarrow \frac{\rho_{\mu_j}\mu_{\mu_j} + \frac{\hat{\alpha}_{\rho_{i,j}}}{\hat{\beta}_{\rho_{i,j}}} \sum_{t=1}^{T} w_i(t) x_j(t)}{\rho_{\mu_j} + \frac{\hat{\alpha}_{\rho_{i,j}}}{\hat{\beta}_{\rho_{i,j}}} \sum_{t=1}^{T} w_i(t)} \tag{33}$$

$$\hat{\rho}_{\mu_{i,j}} \leftarrow \rho_{\mu_j} + \frac{\hat{\alpha}_{\rho_{i,j}}}{\hat{\beta}_{\rho_{i,j}}} \sum_{t=1}^{T} w_i(t) \tag{34}$$

$$\hat{\alpha}_{\rho_{i,j}} \leftarrow \alpha_{\rho_j} + \frac{1}{2}\sum_{t=1}^{T} w_i(t) \tag{35}$$

$$\hat{\beta}_{\rho_{i,j}} \leftarrow \beta_{\rho_j} + \frac{1}{2}\sum_{t=1}^{T} w_i(t)\big[\hat{\rho}_{\mu_{i,j}}^{-1} + (\hat{\mu}_{\mu_{i,j}} - x_j(t))^2\big] \tag{36}$$

Acknowledgment

We would like to thank Zoubin Ghahramani for interesting discussions. We also wish to thank Valor Computerized Systems (Finland) Oy for providing us with the data used in the printed circuit board assembly experiment. This work was supported in part by the Finnish Centre of Excellence Programme (2000–2005) under the project New Information Processing Principles, and by the IST Programme of the European Community, under the PASCAL Network of Excellence, IST-2002-506778. This publication only reflects the authors' views.

References

[1] K. Lagus, E. Alhoniemi, and H. Valpola, “Independent variable group analysis,” in *International Conference on Artificial Neural Networks - ICANN 2001*, ser. LLNCS, G. Dorffner, H. Bischof, and K. Hornik, Eds., vol. 2130. Vienna, Austria: Springer, August 2001, pp. 203–210.

[2] K. Lagus, E. Alhoniemi, J. Seppä, A. Honkela, and P. Wagner, “Independent variable group analysis in learning compact representations for data,” in *Proceedings of the International and Interdisciplinary Conference on Adaptive Knowledge Representation and Reasoning (AKRR’05)*, T. Honkela, V. Könönen, M. Pöllä, and O. Simula, Eds., Espoo, Finland, June 2005, pp. 49–56.

[3] J.-F. Cardoso, “Multidimensional independent component analysis,” in *Proceedings of ICASSP’98*, Seattle, 1998.

[4] G. E. Hinton and R. S. Zemel, “Autoencoders, minimum description length and Helmholtz free energy,” in *Neural Information Processing Systems 6*, J. et al, Ed. San Mateo, CA: Morgan Kaufmann, 1994.

[5] R. S. Zemel, “A minimum description length framework for unsupervised learning,” Ph.D. dissertation, University of Toronto, 1993.

[6] K. Viikki, E. Kentala, M. Juhola, I. Pyykkö, and P. Honkavaara, “Generating decision trees from otoneurological data with a variable grouping method,” *Journal of Medical Systems*, vol. 26, no. 5, pp. 415–425, 2002.

[7] A. Tucker, S. Swift, and X. Liu, “Variable grouping in multivariate time series via correlation,” *IEEE Transactions on Systems, Man and Cybernetics, Part B*, vol. 31, no. 2, pp. 235–245, 2001.

[8] E. Segal, D. Pe’er, A. Regev, D. Koller, and N. Friedman, “Learning module networks,” *Journal of Machine Learning Research*, vol. 6, pp. 557–588, April 2005.

[9] Y. Cheng and G. M. Church, “Biclustering of expression data,” in *Proceedings of the Eighth International Conference on Intelligent Systems for Molecular Biology (ISMB)*, 2000, pp. 93–103.

[10] S. C. Madeira and A. L. Oliveira, “Biclustering algorithms for biological data analysis: A survey,” *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, vol. 1, no. 1, pp. 24–45, 2004.

[11] M. Studený and J. Vejnarová, “The multiinformation function as a tool for measuring stochastic dependence,” in *Learning in Graphical Models*, M. Jordan, Ed. Cambridge, MA, USA: The MIT Press, 1999, pp. 261–297.

[12] M. Nilsson, H. Gustafsson, S. V. Andersen, and W. B. Kleijn, “Gaussian mixture model based mutual information estimation between frequency bands in speech,” in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing 2002 (ICASSP ’02)*, vol. 1, 2002, pp. I–525–I–528.

[13] T. M. Cover and J. A. Thomas, *Elements of Information Theory*. New York: Wiley, 1991.

[14] M. I. Jordan, Z. Ghahramani, T. S. Jaakkola, and L. K. Saul, “An introduction to variational methods for graphical models,” in *Learning in Graphical Models*, M. Jordan, Ed. Cambridge, MA, USA: The MIT Press, 1999, pp. 105–161.

[15] B. J. Frey and G. E. Hinton, “Efficient stochastic source coding and an application to a Bayesian network source model,” *The Computer Journal*, vol. 40, no. 2/3, pp. 157–165, 1997.

[16] J. Rissanen, “Modeling by shortest data description,” *Automatica*, vol. 14, no. 5, pp. 465–471, 1978.

[17] A. Honkela and H. Valpola, “Variational learning and bits-back coding: an information-theoretic view to Bayesian learning,” *IEEE Transactions on Neural Networks*, vol. 15, no. 4, pp. 800–810, 2004.

[18] G. McLachlan and D. Peel, *Finite Mixture Models*. New York: Wiley, 2000.

[19] D. J. C. MacKay, *Information Theory, Inference, and Learning Algorithms*. Cambridge: Cambridge University Press, 2003.

[20] E. Alhoniemi, T. Knuutila, M. Johnsson, J. Röyhkiö, and O. S. Nevalainen, “Data mining in maintenance of electronic component libraries,” in *Proceedings of the IEEE 4th International Conference on Intelligent Systems Design and Applications*, vol. 1, 2004, pp. 403–408.

[21] M. J. Zaki, “Scalable algorithms for association mining,” *IEEE Transactions on Knowledge and Data Engineering*, vol. 12, no. 3, pp. 372–390, 2000.

[22] C. L. Blake and C. J. Merz, “UCI repository of machine learning databases,” 1998, URL: http://www.ics.uci.edu/∼mlearn/MLRepository.html.

[23] D. Fradkin and D. Madigan, “Experiments with random projections for machine learning,” in *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*. New York, NY, USA: ACM Press, August 24-27 2003, pp. 517–522.

[24] T. Joachims, “Making large-scale SVM learning practical,” in *Advances in Kernel Methods - Support Vector Learning*, B. Schölkopf, C. Burges, and A. Smola, Eds. MIT-Press, 1999.

[25] S. Theodoridis and K. Koutroumbas, *Pattern recognition*, 2nd ed. Academic Press, 2003.

[26] J. Winn and C. M. Bishop, “Variational message passing,” *Journal of Machine Learning Research*, vol. 6, pp. 661–694, April 2005.

[27] M. Harva, T. Raiko, A. Honkela, H. Valpola, and J. Karhunen, “Bayes Blocks: An implementation of the variational Bayesian building blocks framework,” in *Proceedings of the 21st Conference on Uncertainty in Artificial Intelligence (UAI 2005)*, Edinburgh, Scotland, 2005, pp. 259–266.