INDEPENDENT VARIABLE GROUP ANALYSIS IN LEARNING COMPACT REPRESENTATIONS FOR DATA

Krista Lagus[1], *Esa Alhoniemi*[2], *Jeremias Seppä*[1], *Antti Honkela*[1], *Paul Wagner*[1]

[1] Neural Networks Research Centre, Helsinki University of Technology,
P.O.Box 5400, FI-02015 Espoo, FINLAND, krista.lagus@hut.fi
[2] University of Turku, Department of Information Technology,
Lemminkäisenkatu 14 A, FI-20520 Turku, FINLAND, esa.alhoniemi@it.utu.fi

ABSTRACT

Humans tend to group together related properties in order to understand complex phenomena. When modeling large problems with limited representational resources, it is important to be able to construct compact models of the data. Structuring the problem into sub-problems that can be modeled independently is a means for achieving compactness. We describe the Independent Variable Group Analysis (IVGA), an unsupervised learning principle that in modeling a data set, also discovers a grouping of the input variables that reflects statistical independencies in the data. In addition, we discuss its connection to some aspects of cognitive modeling and of representations in the brain. The IVGA approach and its implementation are designed to be practical, efficient, and useful for real world applications. Initial experiments on several data sets are reported to examine the performance and potential uses of the method. The preliminary results are promising: the method does seem to find independent subsets of variables. Moreover, it leads to markedly more compact and efficient models than the full model without variable grouping. This allows the re-allocation of freed representational resources for other important tasks. Compact models also contain much fewer parameters and generalize better, and therefore require less data for estimation.

1. INTRODUCTION

The study of effective ways of finding compact representations from data is important for the automatic analysis and data exploration of complex data sets and natural phenomena. Moreover, the study of how conceptual representations emerge in humans during individual learning and in the course of evolution may benefit from the study of basic computational principles that might lead to efficient models of complex phenomena.

Modeling intricate and possibly non-linear dependencies between a very large number of real-valued variables (features) is hard. Learning such models from data generally requires very much computational power and memory.

One way of obtaining compact models is to structure the problem into sub-problems that can be modeled independently. The larger a problem is, the more likely it is that there are relatively independent sub-problems. If this kind of structuring can be done appropriately, it will lead to efficiency of representation, such as more efficient usage of memory space, reduced computational load, reduced use of energy, and improved speed of the implemented system. These same principles apply both to biological and artificial systems that attempt to construct representations of their environment.

For example, the neocortex of the human brain can be considered as a system that attempts to learn a concise representation of its environment, in order to predict events initiated by the environment, as well as outcomes of its own actions. In the mammalian brain different cortical areas can be identified that differ from other areas both on a gross structural level, i.e., which input and output areas they connect to, and functionally. Examples include the somatosensory cortex and the visual cortex. Neocortical microcircuitry is remarkably uniform throughout and the functional difference therefore appears to stem from the connectivity while the underlying "cortical algorithm" is uniform. For a human, such structuring of the connectivity may have arisen during evolution for many reasons, including the cost associated with increasing brain size. For a general discussion regarding human cortical differentiation during evolution, see e.g. [1].

One cognitive, conceptual model that exhibits this kind of structuring of representations is presented in the work of Gärdenfors [2]. He outlines how a collection of distinct, ordered neuronal representation areas can take part in higher-level conceptual representations, and finally in symbolic representations. Using his concepts, the structure consists of *domains*, each of which consists of a set of *integral quality dimensions* that are separable from all other quality dimensions. As an example of a domain he mentions color, consisting of the integral dimensions hue, chromaticity, and brightness. While it is easier to identify such domains and dimensions near the perceptual or

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.

motor systems rather than with more abstract concepts, it is suggested that our representations in general exhibit this kind of structuring. The question remains, what precisely do integral and separable dimensions mean mathematically, and how would such a structure be obtained. In this work we hope to provide some insight on the matter.

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.

Obtaining structured models automatically would be useful in data analysis and visualization, where the goal is to render a large and complex problem more understandable to a human. Consider, for example, a robot which mounts electronic components on a printed circuit board. In order to input a suitable set of attribute values for a new component to be used by the robot, the operator must be aware of complex mutual dependencies between dozens of component attributes. If an incorrect set of values is entered, the assembly fails. In the worst case, iterative tuning of the parameters by hand is required using trial and error. A library which contains data of earlier used components can be used to verify which attribute value combinations are acceptable. It may also be possible to deduce at least some attribute values for a new component based on the data. The automatic detection of subsets of attributes that do not depend on each other would considerably reduce the cognitive load of the human operator, since she can concentrate on a smaller sub-problem at a time. Similar benefits also apply in the case of an automatic system performing the task.

We will next describe a computational principle by which one can learn such a structuring from data.

1.1. Principle of Independent Variable Group Analysis (IVGA)

In an approach that we call Independent Variable Group Analysis (IVGA) [3] the task is to partition a set of input variables (attributes) into groups in such a way that the statistical dependencies of variables within a group are stronger. These dependencies are therefore modeled, whereas the weaker dependencies between different groups are disregarded. The IVGA principle is depicted by the Figure 1.

The computational usefulness of this principle relies on the realization that if two variables are statistically dependent of each other, representing them together is efficient, since related information must be stored only once. However, representing together variables that do not depend on each other is more inefficient. Mathematically this corresponds to the fact that joint probability distributions that can be factorized are more compact than representing a full joint distribution. In terms of a data set or

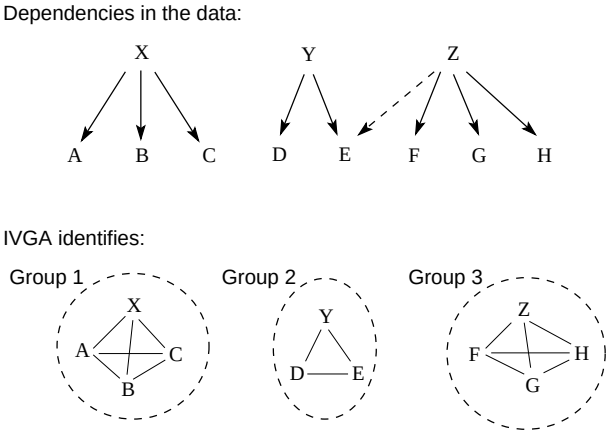

Figure 1. An illustration of the IVGA principle. The upper part of the figure shows the actual dependencies between the observed variables. The arrows that connect variables indicate causal dependencies. The lower part depicts the variable groups that IVGA might find here. One actual dependency is left unmodeled, namely the one between Z and E.

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 becomes and the fewer association rules are needed. With regard to the structure of the cortex this corresponds to the contrast between full connectivity (all cortical areas receive inputs from all other areas) and more limited, structured connectivity.

It turns out that this kind of structuring of a problem can be obtained automatically based on observed data, given that there is a way to measure both model complexity and the precision of the model. Such a measure is obtained e.g. using the minimum-description-length (MDL) principle. Instead of applying MDL directly to derive the necessary cost function, we have used a variational Bayesian method that is equivalent to MDL under certain assumptions.

Figure 2 illustrates the IVGA principle using an artificial example that consists of variables that describe various properties of a person. The available data set consists of four real-valued variables, namely Height, Weight, Income, and Education. Some of these observed variables depend statistically on each other. Some dependencies are caused by unobserved (i.e., latent) variables that affect several of the variables. In this case, IVGA might find two variable groups, namely one connecting Height and Weight and the other with Income and Education. By looking at the particular model learned for each discovered group, detailed information regarding the dependency is obtained.

Identification of the latent variables, or the causal structure of the problem, are very hard problems, and especially so if there are complex nonlinear dependencies. Therefore, methods attacking these problems generally limit the problem in many other ways, such as by model-

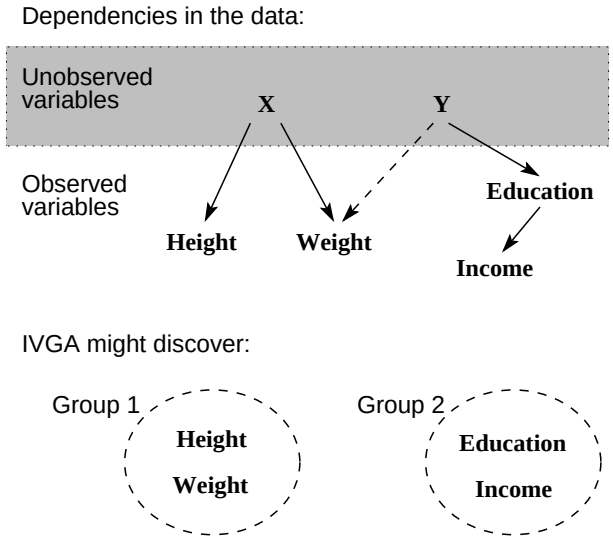

Figure 2. Illustration of the IVGA principle with an artificial example. The observed variables, to be grouped, are Height, Weight, Education, and Income. IVGA might identify here two variable groups, ignoring the weak dependency between Education and Weight that may be caused by the latent variable Y.

ing only linear dependencies, or by allowing only discrete variables.

Moreover, there exist many applications in which already grouping the observed variables is very useful, and such groupings are also sought for by humans who are faced with the problem domain. In contrast, this limitation to grouping observed variables allows one to model also complex nonlinear dependencies.

The IVGA principle has been shown to be sound: a very simple initial method [3] 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 model search algorithm, and the application to various data sets is taking place.

In this article we describe in general terms a method that implements the IVGA principle, as well as look at how the method works both in an artificial case and on real-world data.

1.2. Related work

On a certain level, IVGA can be seen as a method for compact representation of data using multiple independent models. Other alternatives for the same purpose include methods such as multidimensional independent component analysis (MICA) [4] and independent subspace analysis (ISA) [5], as well as factorial vector quantization (FVQ) [6, 7].

In MICA and ISA, the idea 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 dependences in terms of the latent directions defining the subspace. FVQ can be seen as a nonlinear version of MICA, where the component models are VQs over all the variables. The main difference between these and IVGA is that in IVGA, only one model affects a given observed variable whereas in the others, in principle all models affect every observed variable. This difference 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 [8]. The method requires choosing 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 [9]. 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 only certain types of dependencies.

2. A VARIATIONAL BAYESIAN METHOD FOR IVGA

Successful implementation of the IVGA principle requires

1. a method for modeling an individual group,
2. a model evaluation and parameter estimation framework that combines the measure of model complexity and the quality of representation, and
3. a combinatorial search algorithm for finding good groupings.

In our implementation, the models of the groups are simple mixture models such as mixtures-of-Gaussians. The models are derived using the variational Bayesian framework, but they can also be interpreted using the information-theoretic minimum-description-length (MDL) principle. The grouping algorithm is a relatively simple adaptive heuristic.

2.1. Model for a single variable group

The model we use for the individual variable groups is a simple mixture model. The mixture components for real-valued variables are Gaussians and the components for categorical variables general discrete distributions. The real-valued Gaussian mixture is closely related to well-known vector quantization (VQ) model as well as the model employed by the soft k-means algorithm. The model parameters can be learned using the (variational) expectation maximization (EM) algorithm. If we denote component probability distribution i by p_i (contains both real and nominal dimensions if both kinds exist in the modeled data), then the approximative data distribution

can be thought of as a weighted sum of the mixture components;

$$p(\boldsymbol{X}) = \sum_i w_i p_i(\boldsymbol{X}) \quad (1)$$

Adding more mixture components, or making them more precise, increases, by itself, the cost, but may lead to improvement of the cost if the additional component allows a clearly better approximation of the data distribution.

2.2. Variational Bayesian learning

Most model comparison methods are motivated by the principle of Occam's razor: the simplest adequate explanation of natural things should be preferred. A popular method for implementing this in practice is to apply the minimum description length (MDL) principle [10]. It states that the best model for the data set $\boldsymbol{X}$ is the one that provides the most compact encoding of the data, measured by the number of bits $L(\boldsymbol{X})$ needed to transmit the data set to another observer. Shannon's coding theorem links the code lengths intimately with probabilities. It is therefore not very surprising that with a suitable coding method the MDL principle is equivalent to the popular variational approximation to Bayesian statistics [11].

In Bayesian learning, all information provided by the data $\boldsymbol{X}$ is contained in the posterior probability distribution $p(\boldsymbol{\theta}|\boldsymbol{X}, \mathcal{H})$ of the model parameters $\boldsymbol{\theta}$. The variational approximation is based on using a simpler distribution $q(\boldsymbol{\theta})$ to approximate the posterior. The approximation includes additional independence assumptions to avoid modeling dependences between parameters, thus simplifying learning and inference. The cost function used to fit the approximation combines measures of the accuracy of data description through the description length of data modeling errors, and model complexity through the description length of the model parameters. This allows using the cost function directly in comparing different models of the data implied by different groupings. More details on the variational Bayesian method and its relation to information theory can be found in [12, 13].

2.3. Grouping algorithm

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 exhaustive search of all groupings infeasible. For example, $B_{100} \approx 4.8 \cdot 10^{115}$. Therefore, some heuristic for finding a good grouping has to be deployed. In principle, any standard combinatorial optimization algorithm can be used.

We have used a rather simple heuristic algorithm for the grouping, which is described below. Note that any grouping algorithm is computationally feasible only if the cost of an arbitrary group of variables can be computed very fast. Therefore, for instance, when a mixture model for a group is computed, a somewhat inaccurate model is initially trained and fine-tuned during the run of the grouping algorithm. The algorithm is as follows:

1. Each variable is placed into a group of its own. The cost of the model is calculated.
2. In order to decrease the total cost the following operations are repeated until the stopping criterion is met:
   - *Move.* A randomly chosen variable is tentatively moved into each of the other groups and also to a new, singleton group. The grouping with the lowest cost is chosen; if all the possible moves increase the cost the variable is kept in the original group.
   - *Merge.* Two randomly chosen groups are tentatively merged together, and the cost is calculated. If the cost is decreased, the groups are merged.
   - *Split.* A group is selected randomly and this algorithm is then run recursively for the variables within the selected group. If the cost of the obtained grouping is greater than the cost of the original group, the original group is kept; otherwise the group is split according to the obtained grouping.
3. Every now and then the stopping condition for the algorithm is checked. If the stopping condition is not yet fulfilled, the above described operations are continued. As a stopping criterion one can use, for example, some criterion based on the speed of convergence, i.e., the reduction speed of the cost.

It is also possible to assign different probabilities to move, merge, and split operations, and randomly choose one operation at a time for decreasing the cost. Further, the probabilities can be slowly adapted so that the most beneficial operations (i.e., the ones that most efficiently decrease the cost) are chosen more frequently.

The algorithm is stochastic and converges to a local minimum of the cost function. In order to obtain a good grouping, the algorithm should be run many times and the grouping with the smallest cost should be selected as the final grouping. Also, by looking at several resulting groupings (and models of the groups) one can often obtain a good insight into the data.

3. EXPERIMENTS AND RESULTS

We first describe an experiment on an artificial data set designed to illustrate the IVGA principle and to verify the method. Next, we present experimental results on three different types of real-world data sets, namely an electrocardiography data used for machine identification of cardiac arrhythmias, a data set used in the design of new circuit boards, and a text document collection.

3.1. Proof of concept: Artificial data set

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

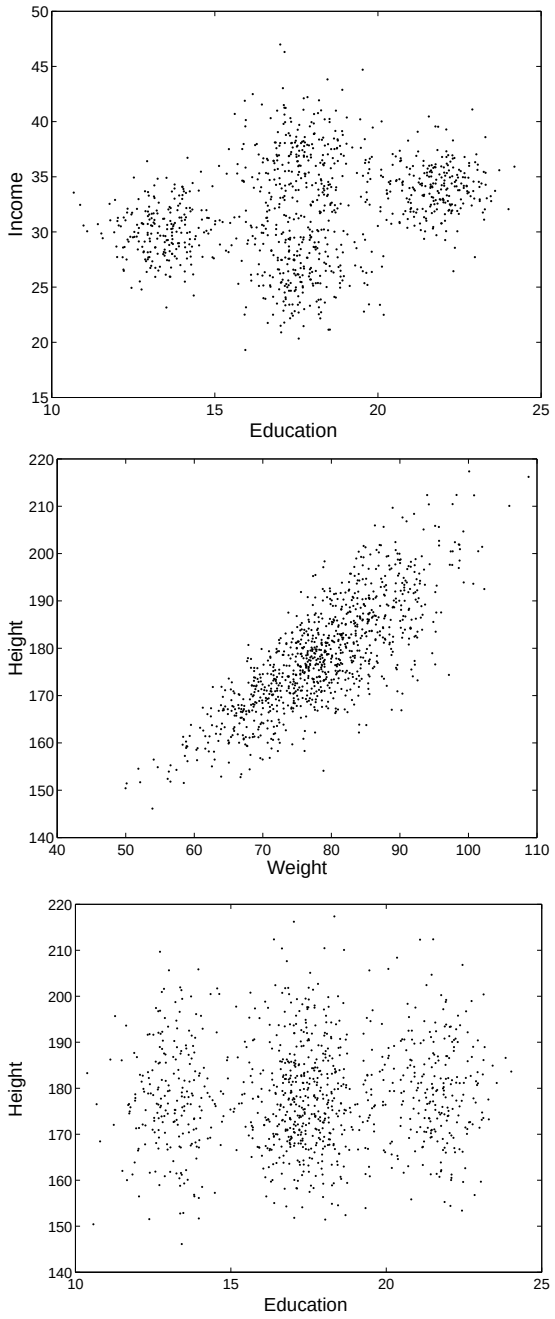

Figure 3. Comparison of different two-dimensional subspaces of the data. Due to the strong dependencies between the variables shown in the first two pictures it is useful to model those variables together. In contrast, in the last picture no such dependency is observed and therefore no benefit is obtained from modeling the variables together.

both education and income are statistically independent of height and weight.

Figure 3 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 Figure 3 the data points lie in few, more concentrated clusters and thus can generally be described (modeled) with a lower cost when compared to the third plot. As expected, when the data was given to our implementation of the IVGA, the resulting grouping was

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

Table 1 compares the costs of some possible groupings.

Grouping	Total Cost	Model 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 1. 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 model parameters, i.e. real numbers, used to represent the Gaussian mixture component distributions. The total costs are for mixture models optimized carefully using the IVGA implementation. The model search of our IVGA implementation was able to discover the best grouping, i.e., the one with the smallest cost.

3.2. Arrhythmia data set

The identification of different types of heart problems, namely cardiac arrhythmias, is carried out based on electrocardiography measurings from a large number of electrodes. We used a freely available electrocardiography data set arrhythmia [14] used for machine identification of cardiac arrhythmias. It consists of 280 variables collected from 452 patients. The data set includes 74 real-valued variables such as the age and weight of the patient and wave amplitudes and durations of different parts of the signal recorded from each of the 12 electrodes. The 206 nominal variables code, for example, the gender of the patient and the existence of various anomalies in each electrode. One variable describes a human cardiologist's classification of the patient into one of 16 arrhythmia types.

IVGA was run several times on the data. It grouped the variables into 90-110 groups with about 1-10 variables in each group. The individual IVGA runs took from one to eight hours. The grouping with the lowest cost was obtained in a run that used adaptively changing operation probabilites and lasted for seven hours[1]. IVGA was run on one CPU in a Linux server with 2.2 GHz Opteron processors and 4 GB of memory.

A specific finding regarding the discovered groups was that often the variable that contained the cardiologist's diagnosis was grouped with two variables, namely the duration and amplitude of a certain wave (R'-wave) in the electrode V1. Looking at the mixture components inside the

[1]The method can be speeded up roughly by an order of magnitude if one uses a C implementation of the mixture modeler instead of the current Matlab implementation, but in these experiments such speedups were not utilized.

model it appeared that this particular grouping emerged because the patients with a particular, common arrhythmia type differed from the others with respect to these recordings from the electrode V1. This finding suggests a possible use of our approach as a tool for searching for good features for a supervised learning task.

3.3. Printed Circuit Board Assembly

In the electronics industry of today, mounting of components on a printed circuit board is usually carried out by an assembly robot. Typically, one robot is used in manufacturing of many products each of which consists of components of different type. Before assembly, the specifications of all the required components (values of component attributes) need to be input in an electronic component library which the robot then utilizes. The determination of the values for dozens of attributes is carried out by hand and requires experience of the human operators, manual browsing through the specification documents and even testing using trial and error.

The component library can be seen as a data matrix. The columns of the matrix depict component attributes and each row contains attribute values of one component. The attributes are not mutually independent. By modeling the dependencies of the existing library data, it is possible to construct a support system to assist input of data of new components. So far, modeling of the dependencies has been carried out using association rules [15]. Unfortunately, extraction of the rules from the data is computationally heavy, and memory consumption of the data structure (for example, a trie) for the rules is very high.

In this experiment, we used IVGA to split the data into separate groups, and dependencies within each group were modeled using association rules. The data was obtained from the component library of an operational assembly robot, and it consisted of attribute values of 1 000 components. After preprocessing – for example, removal of constant valued attributes – there were 24 attributes (17 nominal, 7 real-valued) which were modeled. The IVGA was run 20 times for the data set. One run of IVGA implemented on Matlab 6.5.1 took an average of 510 seconds of CPU time on a Linux server with 2.2 GHz Opteron processors and 4 GB of memory. In the grouping with the lowest cost the variables were divided into three groups; the sizes of the groups were 15, 6, and 3 attributes.

Association rules were then used for modeling of the dependencies of (1) the whole data and (2) the three variable groups separately. In both cases, the extracted rules were used for one-step prediction of the attribute values for a previously unseen data set of 1 000 components. The data consist 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 are shown in both cases in Table 2. Splitting of the data using IVGA lead to significant improvements in the efficiency of the obtained model: it accelerated computation of the rules, dramatically reduced the size of the data structure, and decreased the number of the incorrect predictions. On the other hand, the number of missing predictions was clearly larger for the grouped data than for the whole data, because for the first attribute value of every group, no prediction could be made whereas for the whole data, only the prediction for the first attribute could not be obtained.

	Whole data	**Grouped data**
Computation time (s)	194	< 1
Size of trie (nodes)	1 054 512	3 914
Correct predictions (%)	38.05	32.45
Incorrect predictions (%)	1.61	0.68
Missing predictions (%)	60.43	66.88

Table 2. Summary of the results of the component data experiment. All the quantities for the grouped data are sums over the three groups. Also note that the size of trie is the same as the number of association rules.

Some advantages of the IVGA model were not yet utilized in the experiment. First, automatic discretization of continuous variables is often a problem in applications of association rules, but it is automatically carried out by the mixture model (note that for simplicity, we treated the continuous values as nominal in the computation of the association rules). Second, division of the variables into groups makes it computationally possible to find rules which are based on smaller proportions of the data. Using data mining terminology: it is possible to use smaller minimum support. Third, it may be possible to even completely ignore computation of the association rules, and instead use the mixture models of the IVGA directly for obtaining the predictions. All the issues deserve to be studied in more depth in future research.

3.4. Text document data

Due to the large numbers of possible words, obtaining structured representations for textual data can be of general interest. For example, the modeling of very large text document collections using methods such as the WEBSOM [16] requires storage of vectors with a dimensionality of tens of thousands of variables, if no dimension reduction methods are used.

In this initial experiment, the document set consisted of 529 patent abstracts from three distinct topical categories, namely A21: Foodstuffs; Tobacco, A61: Health; Amusement and F41: Weapons; Blasting. While the task was to find an efficient representation for the set of documents, we were interested in the kinds of groupings that this method might obtain for the words. Although the currently implemented model families are not particularly well suited for very sparse occurrence data, such as text documents, we wished to look at this problem due to the ease of interpreting the word groupings.

As the features (variables) we picked 100 out of the 400 most frequent words. To encode a document, we

Group	Words in the group
1	speed
2	tobacco, smoke
3	walter
4	top, roll
5	target, system, signal, sight, shoot, set, sensor
6	water, value, upper, up, treatment, thus, then, temperature, sugar, subject, solution, side, separate, salt, result
7	vehicle, under, strike, spring, slide, sleeve, safety, round, return, retain
8	vessel, small, ring
9	suitable, strip, shell
10	section
11	without, with, who, which, when, weapon, way, wall, use, type, to, time, through, three, this, they, there, the, that, than, supply, such, substance, step, so, ski, simple, shot, say, rifle
12	via
13	two
14	weight, valve, tube, together, tip, suction, space, seal, screw, roller
15	rod
16	unit
17	while, wheel, vertical, surface, support, shape, shaft, second, same, rotation, rotate

Table 3. The 17 word groups found in the IVGA run with lowest cost. The results seem to exhibit some topical grouping: see, e.g., in group 6 the words water and temperature, as well as sugar and salt. In addition, group 11 contains a concentration of non-content words, although with several content words as well.

recorded the presence or absence of each word as a nominal variable. IVGA was then run 20 times; on the average, 22.5 groups of words were obtained. One run of IVGA implemented on Matlab 6.5.1 took an average of 1155 seconds of CPU time on a Linux server with 2.2 GHz Opteron processors and 4 GB of memory. The grouping with the lowest cost contained 17 word groups and was selected for closer inspection in Table 3.

In many of the groupings some patterns were typical. For example, topically related words, such as tobacco and smoke, were often found together. Many groupings also included one group with a high concentration of non-content words (group 11 in Table 3). These very preliminary results are promising in that the method does seem to find some relevant structure in the problem. However, more work is needed in order to identify the best way of applying IVGA for this type of data.

4. DISCUSSION

One may regard the IVGA principle as one attempt to specify in mathematical and computational terms how a structured conceptual representation, such as is discussed by Gärdenfors in [2], might emerge from the statistical properties of data. As one possible model of a single domain he considers the Self-Organizing Map (SOM) [17], which forms an ordered representation of its input data. Although in our implementation of IVGA we have applied a mixture of Gaussians (which is very similar to VQ) as the corresponding model of a variable group, a SOM-like ordered model could be utilized as well for a single variable group, once the grouping is first obtained using the current IVGA implementation[2].

A similar two-phase approach was utilized with the Circuit Board experiment, where after the grouping was obtained, another method was utilized to discover association rules. An advantage of such a two-phase approach is that even if the latter method is computationally much more demanding, it can now be applied since the maximal dimensionality of the data at any given time is now considerably lower (the dimensionality at any point equals the number of variables in the group being modeled).

Moreover, from this two-phase approach one might draw a gross analogy to the development of the brain, if one assumes that some general structure of the brain is determined by evolution. The evolutionary development would thus correspond to the grouping of related input variables by IVGA. Later, individual development might be viewed as attempting to discover the best possible model for each sub-structure based on the data encountered by the individual, which corresponds to the later learning of the best model for a single variable group. The individual learning might employ even a considerably different (a much more intricate) model family than was utilized during early stages of the evolutionary phase.

When considered as a means for solving real-world data analysis problems using automatic methods, IVGA shows marked potential. The experiment on a simple artificial data set confirms that the method does find the independent groups in the example. Of particular interest raises the Circuit Board example, in which already the initial experiments lead to considerable savings in the efficiency of the model. These, in turn, may later lead to improvements in quality, since the freed computational resources can be re-allocated, and since less data is required for accurate estimation of the structured model.

An interesting question is whether the IVGA approach might be useful for feature selection for supervised learning, as suggested by the experiment with the Arrhythmia data set in Section 3.2. Another possibility worth considering is whether the structuring of the set of input features might also be of use for systems that must weigh their input features based on some relevance feedback: instead of calculating changes for individual feature weights, the collective weight of a feature group might be adjusted.

An important benefit of the IVGA was highlighted by

[2]The mathematically correct way would be to use a probabilistic variant of the SOM such as the Generative Topographic Mapping (GTM) [18] as the model for a single group during IVGA. Nevertheless, reasonably good results might of course be obtained utilizing the described practical two-phase shortcut, in cases when the two methods tend to discover similar dependencies.

the circuit board experiment, in which we saw a drastic reduction in computation time, model storage space and required learning data. In this example IVGA produced a model that required about 200 times less model parameters and learning time, even when using the complete data set. Moreover, with a reduced number of parameters one can use less data for estimating the model with good precision. IVGA can thus be initially applied once to learn the independence structure of the modeled domain. In further runs, a much smaller set of data is sufficient for evaluating the model parameters in a particular case. These two phases are analogous to evolutionary adaptation of brain structure, and specific learning within the given brain structure during a life time of an individual.

In conclusion, the presented approach shows clear potential for finding models for large and complex real-world phenomena. Due to discovering some essential independence structure of the data set, the models become more efficient and can lead to better quality in systems that have limited representational resources.

5. ACKNOWLEDGMENTS

We are grateful to Harri Valpola for numerous insightful discussions regarding the topics presented here. We also wish to thank Valor Computerized Systems (Finland) Oy for providing us with the data used in the printed circuit board assembly experiment.

6. REFERENCES

[1] Herbert P. Killackey, "Evolution of the human brain: A neuroanatomical perspective," in *The Cognitive Neurosciences*, Michael S. Gazzaniga, Ed., pp. 1243–1253. MIT Press, 1994.

[2] Peter Gärdenfors, *Conceptual Spaces*, MIT Press, 2000.

[3] Krista Lagus, Esa Alhoniemi, and Harri Valpola, "Independent variable group analysis," in *International Conference on Artificial Neural Networks - ICANN 2001*, Georg Dorffner, Horst Bischof, and Kurt Hornik, Eds., Vienna, Austria, August 2001, vol. 2130 of *LLNCS*, pp. 203–210, Springer.

[4] Jean-François Cardoso, "Multidimensional independent component analysis," in *Proceedings of ICASSP'98*, Seattle, 1998.

[5] Aapo Hyvärinen and Patrik Hoyer, "Emergence of phase and shift invariant features by decomposition of natural images into independent feature subspaces," *Neural Computation*, vol. 12, no. 7, pp. 1705–1720, 2000.

[6] Geoffrey E. Hinton and Richard S. Zemel, "Autoencoders, minimum description length and Helmholtz free energy," in *Neural Information Processing Systems 6*, J.D.Cowan et al, Ed., San Mateo, CA, 1994, Morgan Kaufmann.

[7] Richadr S. Zemel, *A Minimum Description Length Framework for Unsupervised Learning*, Ph.D. thesis, University of Toronto, 1993.

[8] Kati Viikki, Erna Kentala, Martti Juhola, Ilmari Pyykkö, and Pekka Honkavaara, "Generating decision trees from otoneurological data with a variable grouping method," *Journal of Medical Systems*, vol. 26, no. 5, pp. 415–425, 2002.

[9] Allan Tucker, Stephen Swift, and Xiaohui 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.

[10] Jorma Rissanen, "Modeling by shortest data description," *Automatica*, vol. 14, no. 5, pp. 465–471, 1978.

[11] Geoffrey E. Hinton and Drew van Camp, "Keeping neural networks simple by minimizing the description length of the weights," in *Proceedings of the COLT'93*, Santa Cruz, California, USA, July 26–28, 1993, pp. 5–13.

[12] Harri Lappalainen and James W. Miskin, "Ensemble learning," in *Advances in Independent Component Analysis*, Mark Girolami, Ed., pp. 76–92. Springer-Verlag, Berlin, 2000.

[13] Antti Honkela and Harri 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.

[14] C. L. Blake and C. J. Merz, "UCI repository of machine learning databases," 1998, URL: `http://www.ics.uci.edu/~mlearn/MLRepository.html`.

[15] Esa Alhoniemi, Timo Knuutila, Mika Johnsson, Juha Röyhkiö, and Olli S. Nevalainen, "Data mining in maintenance of electronic component libraries," in *Proceedings of the IEEE 4th International Conference on Intelligent Systems Design and Applications*, 2004, vol. 1, pp. 403–408.

[16] Krista Lagus, Samuel Kaski, and Teuvo Kohonen, "Mining massive document collections by the WEBSOM method," *Information Sciences*, vol. 163, no. 1–3, pp. 135–156, 2004.

[17] Teuvo Kohonen, *Self-Organizing Maps*, vol. 30 of *Springer Series in Information Sciences*, Springer, Berlin, 3rd extended edition, 2001 edition, 2001.

[18] Christopher M. Bishop, Markus Svensén, and Christopher K. I. Williams, "GTM: The generative topographic mapping," *Neural Computation*, vol. 10, no. 1, pp. 215–234, 1998.