ESANN'2007 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 25-27 April 2007, d-side publi., ISBN 2-930307-07-2.

Agglomerative Independent Variable Group Analysis

Antti Honkela[1] Jeremias Seppä[1]
Esa Alhoniemi[2] *

1- Helsinki University of Technology, Adaptive Informatics Research Centre
P.O. Box 5400, FI-02015 TKK, Espoo, Finland

2- University of Turku, Department of Information Technology
FI-20014 University of Turku, Finland

Abstract. Independent Variable Group Analysis (IVGA) is a principle for grouping dependent variables together while keeping mutually independent or weakly dependent variables in separate groups. In this paper an agglomerative method for learning a hierarchy of IVGA groupings is presented. The method resembles hierarchical clustering, but the distance measure is based on a model-based approximation of mutual information between groups of variables. The approach also allows determining optimal cutoff points for the hierarchy. The method is demonstrated to find sensible groupings of variables that ease construction of a predictive model.

1 Introduction

Simplifying the structure of a data set is an important preprocessing step allowing application of machine learning and data mining techniques to large data sets. One effective method of achieving this is to break the large problem to smaller subproblems that can be solved independently. If the computational complexity of the learning technique of interest is superlinear, this can speed up processing and decrease memory requirements significantly.

Independent Variable Group Analysis [1, 2] (IVGA) aims at grouping of mutually dependent variables together and placing independent or weakly dependent ones in different groups. The IVGA problem can be solved in many different ways. The original IVGA algorithm [1, 2] uses mixture models for modelling of the individual groups and a heuristic combinatorial hill climbing search to find the optimal grouping [1].

In this paper an alternative grouping algorithm for IVGA called *Agglomerative Independent Variable Group Analysis* (AIVGA) is presented. AIVGA is based on the idea of agglomerative hierarchical clustering of variables [3]. Initially, each variable is placed on a group of its own. The groups are then combined by greedily selecting the operation that decreases the cost most. The result is a hierarchy of groupings of different sizes. This can be more useful than a single grouping returned by the original IVGA algorithm if the optimal number of groups provided by IVGA is not optimal for future processing.

*This work was supported in part by the IST Programme of the European Community under the PASCAL Network of Excellence, IST-2002-506778. This publication only reflects the authors' views.

The general problem of hierarchical clustering of variables and some related methods are studied in Sec. 2. The computational methodologies behind AIVGA and the algorithm itself are presented in Sec. 3. Sec. 4 presents experimental results on applying the algorithm as preprocessing for a prediction task. The paper concludes with discussion in Sec. 5.

2 Hierarchical Variable Grouping

The result of the AIVGA algorithm can be seen as a hierarchical clustering of variables, similarly as the solution of an IVGA problem can be seen as a regular clustering of the variables. For each level in the clustering, there is a probabilistic model for the data consisting of a varying number of independent parts, but there is no single generative model for the hierarchy. The Bayesian marginal likelihood cost function used in AIVGA allows determining the optimal points to cut the tree similarly as in the Bayesian hierarchical clustering method [4].

The AIVGA approach can be contrasted with models such as the hierarchical latent class model [5] in which the latent variables associated with different groups can be members of higher level groups. The latent class model presented in [5] is, however, limited to categorical data whereas AIVGA can also be applied to continuous and mixed data. The simpler structure of the separate IVGA models makes the method computationally more efficient.

AIVGA is closely related to the hierarchical clustering algorithm using mutual information presented in [6]. The main difference is that AIVGA provides a generative model for the data at each level of the hierarchy. Also, the Bayesian model-based approximation of mutual information allows determining the optimal cutoff point or points for the hierarchy and provides a more reliable global cost function.

The IVGA models formed in different stages of the AIVGA algorithm have many possible interpretations and therefore many connections to other related methods. These are reviewed in detail in [2]. In context of the AIVGA algorithm, the most interesting interpretation for IVGA is clustering of variables. The mixture models used to model the groups yield a secondary clustering of the samples, thus relating IVGA to biclustering. This makes AIVGA in a sense a hierarchical alternative of biclustering. However, both IVGA and AIVGA always cluster all the variables and samples, while biclustering methods concentrate only on interesting subsets of variables and samples.

3 Algorithm

Let us assume that the data set $\boldsymbol{X}$ consists of vectors $\mathbf{x}(t),\ t = 1, \dots, T$. The vectors are N-dimensional with the individual components denoted by $x_j,\ j = 1, \dots, N$, and let $\boldsymbol{X}_j = (x_j(1), \dots, x_j(T))$. The objective of IVGA and AIVGA is to find a partition of $\{1, \dots, N\}$ to M disjoint sets $\mathcal{G} = \{\mathcal{G}_i | i = 1, \dots, M\}$ such that the sum of marginal log-likelihoods of models $\mathcal{H}_i$ for the different groups is maximised. As shown in [2], this is approximately equivalent to minimising the mutual information (or multi-information when $M > 2$) between the groups. As this cannot be evaluated directly, the bounds evaluated using the variational Bayes (VB) approximation [2] are used instead to get a cost function $\mathcal{C}$ to minimise:

$$\mathcal{C}(\mathcal{G}) = \sum_i \mathcal{C}(\{\boldsymbol{X}_j | j \in \mathcal{G}_i\}|\mathcal{H}_i) \geq -\sum_i \log p(\{\boldsymbol{X}_j | j \in \mathcal{G}_i\}|\mathcal{H}_i), \tag{1}$$

where

$$\mathcal{C}(\{\boldsymbol{X}_j | j \in \mathcal{G}_i\}|\mathcal{H}_i) = \int \log \frac{q_i(\boldsymbol{\theta}_i)}{p(\{\boldsymbol{X}_j | j \in \mathcal{G}_i\}, \boldsymbol{\theta}_i|\mathcal{H}_i)} q_i(\boldsymbol{\theta}_i) d\boldsymbol{\theta}_i. \tag{2}$$

As a byproduct, an approximation $q_i(\boldsymbol{\theta}_i)$ of the posterior distribution $p(\boldsymbol{\theta}_i|\boldsymbol{X}, \mathcal{H}_i)$ of the parameters $\boldsymbol{\theta}_i$ of each model $\mathcal{H}_i$ is also obtained. This approximation minimises the Kullback–Leibler divergence $D_{\mathrm{KL}}(q||p)$.

As in [2], the individual groups are modelled with mixture models. The mixture components are products of individual distributions for every variable. These distributions are Gaussian for continuous variables and multinomial for categorical variables. For purely continuous data, the resulting model is thus a mixture of diagonal covariance Gaussians. Variances of the Gaussians can be different for different variables and for different mixture components. The models are learned with a variational EM algorithm. Exact details and learning rules are presented in [2].

An outline of the AIVGA algorithm is presented as Algorithm 1. The algorithm is a classical agglomerative algorithm for hierarchical clustering [3]. It is initialised by placing each observed variable in a group of its own. After that two groups are always merged so that the reduction in mutual information is as large as possible. As the cost function of Eq. (1) is additive over the groups, changes can be evaluated locally by only considering the changed groups.

Algorithm 1 Outline of the AIVGA algorithm.

```
$c \leftarrow N, \mathcal{G}_i \leftarrow \{x_i\}, i = 1, \ldots, N$
while $c > 1$ do
  $c \leftarrow c - 1$
  Find groups $\mathcal{G}_i$ and $\mathcal{G}_j$ such that $\mathcal{C}(\{\mathcal{G}_i \cup \mathcal{G}_j\}) - \mathcal{C}(\{\mathcal{G}_i, \mathcal{G}_j\})$ is minimal
  Merge groups $\mathcal{G}_i$ and $\mathcal{G}_j$
  Save grouping of size $c$
end while
```

The AIVGA algorithm requires fitting $\mathcal{O}(N^2)$ mixture models for the groups. This is necessary as for the first step all pairs of variables have to be considered for possible merge. After this, only further merges of the previously merged group with all the other groups have to be considered again.

The expected results of merging two groups are estimated by learning a mixture model for the variables in the union of the two groups. The resulting cost of the combined model is compared to the sum of the costs of the two independent models for the groups. In order to avoid local minima of the variational EM algorithm, the mixture models are learned several times with different random initialisations and the best result is always used. The number of mixture components is found by perturbing the number of components in reinitialisation and pruning out unused components during EM iteration. The initial number of mixture components is 2. It is more likely that the number increases than decreases. The new number is saved only if the new model is the best for the particular group so far.

4 Experiment

In the experiment, we considered a system to support and speed up user input of component data of a printed circuit board assembly robot. The system is based on a predictive model which models the redundancy of the existing data records using association rules. The application is described in detail in [7].

Our goal was to find out if a set of small models would work better than a large monolithic model – and if so, we could determine a set of such models. We divided the 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 remaining 20 %). AIVGA was run three times for the training data set. On each run, almost the same result was obtained. In Fig. 1, the grouping tree (dendrogram) that contains the grouping with the lowest cost is shown.

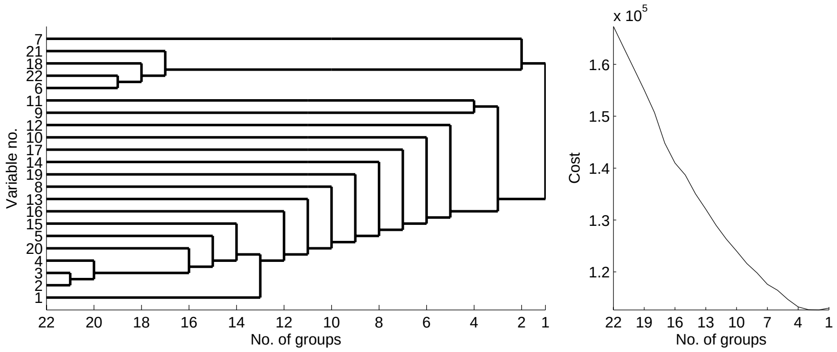

Fig. 1: A grouping (left) and cost history graph (right) for the component data. The model with the lowest cost consists of two groups.

After the groupings were found, association rules were used for modelling of the dependencies of (i) the whole data and (ii) variable groups of the 21 different variable groupings. The association rules are based on so-called frequent sets, which in our case contain the attribute value combinations that are common in the data. Support of a set is defined as the proportion of entries of the whole data in which the attribute values of the set are present.

There exists various algorithms for computation of the frequent sets which differ in computational aspects but which all give identical results. We computed the sets using a freely available implementation of the Eclat algorithm [8][1]. For the whole data, the minimum support dictating the size of the model was set to 5 %, which was the smallest computationally feasible value in terms of memory consumption. For the group models the minimum support was set to 0.1 %, which always gave clearly smaller models than the one for the whole data. Minimum confidence (the "accuracy" of the rules) was set to 90 %.

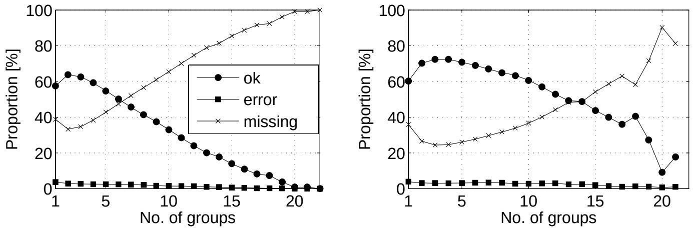

Fig. 2: Prediction results when the data have been divided into 1 . . . 22 group(s). For each grouping, the results are sums of results of all the models corresponding to the grouping. The right panel indicates the the prediction results when the missing predictions of the first variable of each group are ignored. In the left panel, these are taken into account. The legend shown in the left panel is the same for both illustrations.

The rules were used for one-step prediction of the attribute values of the testing data. In Fig. 2, the proportion of the correct, incorrect, and missing predictions (that is, the cases when the confidence of the best rule was below 90 %) are shown for the whole data and grouped data. For the first variable of any group, previous input does not exist, so the prediction is always missing.

Fig. 2 reveals two different aspects of the results. The left panel shows the total prediction results. The right panel shows the performance of the predictive scheme for the values for which it was even possible to try to compute a prediction. In terms of correct predictions, the total results are best using 2 groups, but the predictive scheme performs slightly better when there are 3 groups. However, the left panel indicates that if the data are grouped in 2–4 groups, the number of the correct predictions is higher than using the monolithic model. The models are also clearly lighter to compute and consume less memory.

In this experiment, the 2 groups were the same as those found in our earlier study [2] with the same data using non-agglomerative IVGA. However, computation of the results in the previous study took dozens of hours whereas in this case we needed only a small fraction of it (about three hours). In addition, the AIVGA provided a systematic way for determination of the 2 groups whereas using the non-agglomerative approach, finding the groups was more laborious.

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

5 Discussion and Conclusions

In this paper, we presented AIVGA, an agglomerative algorithm for learning hierarchical groupings of variables according to the IVGA principle. AIVGA helps the use of IVGA when further use of the results makes the optimal number of groups difficult to determine. The computation time of a single AIVGA run is comparable to a single run of the regular IVGA algorithm [2], so the additional information in the tree of groupings comes essentially for free. If the single best grouping is sought for, the output of AIVGA can be used to initialise the regular IVGA algorithm [2], which can then relatively quickly find if there are better solutions that are nearly but not exactly tree conforming.

In addition to the presented example, the proposed method has several potential applications. For instance, AIVGA may be used for feature selection in supervised learning by finding variables having the strongest dependencies with the target variable as in many other methods using mutual information [9, 2].

The cost function of IVGA and the probabilistic modelling framework can in some cases provide more meaningful clustering results than existing algorithms. AIVGA effectively complements that by providing a structured solution mechanism as well as a richer set of potential solutions.

Acknowledgements

We thank Zoubin Ghahramani for interesting discussions and Valor Computerized Systems (Finland) Oy for providing us with the data used in the experiment.

References

[1] K. Lagus, E. Alhoniemi, and H. Valpola. Independent variable group analysis. In *Proc. Int. Conf. on Artificial Neural Networks - ICANN 2001*, volume 2130 of *LNCS*, pages 203–210, Vienna, Austria, 2001. Springer.

[2] E. Alhoniemi, A. Honkela, K. Lagus, J. Seppä, P. Wagner, and H. Valpola. Compact modeling of data using independent variable group analysis. *IEEE Transactions on Neural Networks*, 2007. To appear. Earlier version available as Report E3 at `http://www.cis.hut.fi/Publications/`.

[3] R. O. Duda, P. E. Hart, and D. G. Stork. *Pattern Classification.* Wiley, 2nd edition, 2001.

[4] K. A. Heller and Z. Ghahramani. Bayesian hierarchical clustering. In *Proc. 22nd Int. Conf. on Machine Learning (ICML 2005)*, pages 297–304, 2005.

[5] N. L. Zhang. Hierarchical latent class models for cluster analysis. *Journal of Machine Learning Research*, 5:697–723, 2004.

[6] A. Kraskov, H. Stögbauer, R. G. Andrzejak, and P. Grassberger. Hierarchical clustering using mutual information. *Europhysics Letters*, 70(2):278–284, 2005.

[7] E. Alhoniemi, T. Knuutila, M. Johnsson, J. Röyhkiö, and O. S. Nevalainen. Data mining in maintenance of electronic component libraries. In *Proc. IEEE 4th Int. Conf. on Intelligent Systems Design and Applications*, volume 1, pages 403–408, 2004.

[8] M. J. Zaki, S. Parthasarathy, M. Ogihara, and W. Li. New algorithms for fast discovery of association rules. In *Proceedings of the 3rd International Conference on Knowledge Discovery and Data Mining (KDD)*, pages 283–286, 1997.

[9] R. Battiti. Using mutual information for selecting features in supervised neural net learning. *IEEE Transactions on Neural Networks*, 5(4):537–550, 1994.