Appeared in ICANN 2001, Vienna, Austria. pp. 203-210

Independent Variable Group Analysis

Krista Lagus, Esa Alhoniemi, and Harri Valpola

Neural Networks Research Centre, Helsinki University of Technology
P.O.Box 5400, FIN-02015 HUT, Finland
{krista.lagus, esa.alhoniemi, harri.valpola}@hut.fi

Abstract. 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. In this article we introduce Independent Variable Group Analysis (IVGA), a practical, efficient, and general approach for obtaining sparse codes. We apply the IVGA approach for a situation where the dependences within variable groups are modeled using vector quantization. In particular, we derive a cost function needed for model optimization with VQ. Experimental results are presented to show that variables are grouped according to statistical independence, and that a more compact model ensues due to the algorithm.

1 Introduction

The goal of unsupervised learning is to extract an efficient representation of the statistical structure implicit in the observations. A good model is both accurate and simple in terms of model complexity, i.e., it forms a *compact representation* of the input data. Sparse coding, independent components, etc., can all be justified from the point of view of constructing compact representations. Learning amounts to searching in a model space by optimization of some cost function that measures both the accuracy of representation and—ideally at least—the model complexity.

In problems with a large number of diverse observations there are often groups of variables which have strong mutual dependences within the group but which can be considered practically independent of the variables in other groups. It can be expected that the larger the problem domain, the more independent groups there are[1]. Estimating a model for each group separately produces a more compact representation than applying the model to the whole set of variables. Compact representations are computationally beneficial and, moreover, offer better generalization.

1.1 Independent Variable Group Analysis

We suggest an approach, independent variable group analysis (IVGA), where the dependences of variables within a group are modeled, whereas the dependences

[1] Consider, for instance, creating a model of the whole world...

between the groups are neglected. This generates a pressure towards building groups whose variables are mutually dependent but are largely independent of the variables in other groups.

Usually such variable grouping is performed by a domain expert, prior to modeling with automatic, adaptive methods. However, we claim that it is worthwhile and feasible to try to obtain groupings automatically in order to create structured, efficient large-scale models automatically.

In IVGA, each separate group can be modeled using any method, as long as a cost function that measures the quality of the representation, including both compactness and faithfullness, is derived for the model family. In particular, such a cost function can be derived within the statistical framework to generative modeling. In this approach, the best model is the one that gives the highest probability to the observations.

Furthermore, we demonstrate the approach by describing and evaluating an algorithm, $\mathrm{IVGA_{VQ}}$, that can be used for computing IVGA when the model space used for modeling dependences between variables consists of vector quantizers (VQs). Variational EM-algorithm is used for adapting the VQs and for computing the cost function.

1.2 Structure of the article

The structure of the rest of the article is as follows: first we describe two related models in section 2. In section 3 we introduce an algorithm for computing IVGA where VQs are used in modeling each variable group. Section 4 describes an experiment performed to validate the feasibility of the approach and to demonstrate the algorithm. Conclusions are presented in section 5.

2 Related models

In multidimensional independent component analysis (MICA) [1], and its subsequent development independent subspace analysis (ISA) [4], 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. The approach bears resemblance to IVGA which can be seen as a nonlinear version of MICA with the additional requirement which restricts the subspaces to be spanned by subsets of the variable axes.

Factorial vector quantization (FVQ) [3, 7] can be seen as a nonlinear version of MICA. It uses several different VQs that cooperate to reconstruct the observations and is thus very similar to $\mathrm{IVGA_{VQ}}$. The structural differences of the FVQ model compared to the $\mathrm{IVGA_{VQ}}$ model are: in FVQ (1) each vector in each pool or group contains *all* the components (variables) of the original input vector, and (2) these are summed to produce the value of a single output. In contrast, in $\mathrm{IVGA_{VQ}}$ each variable group is modeled by exactly one vector quantizer (VQ).

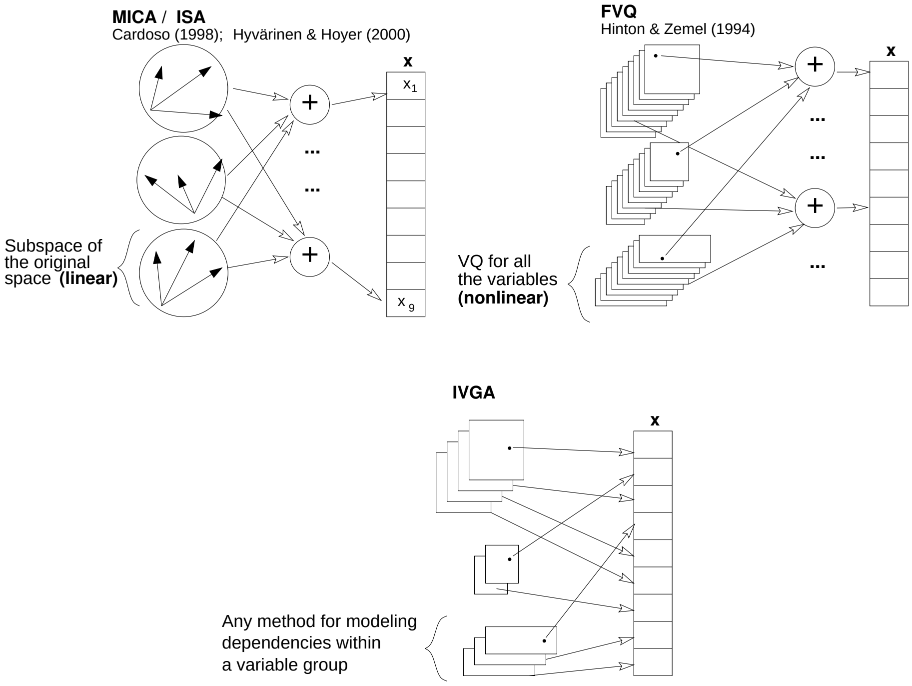

Fig. 1. Schematic illustrations of IVGA and related algorithms, namely MICA/ISA and FVQ that each look for multi-dimensional feature subspaces in effect by maximizing a statistical independency criterion. The input $\mathbf{x}$ is here 9-dimensional. The numbers of squares in FVQ and IVGA denote the numbers of variables modeled in each sub-model, and the numbers of black arrows in MICA the dimensionality of the subspaces.

This leads to efficient computation as each VQ can operate independently as opposed to FVQ where the winners are found iteratively.

Figure 1 illustrates the model structures of MICA/ISA, FVQ, and IVGA.

3 IVGA$_{\mathrm{VQ}}$: algorithm for IVGA with VQs as data models

Any IVGA algorithm consists of two parts, (1) grouping variables, and (2) constructing a separate model for each variable group. An independent variable grouping is obtained by comparing models with different groupings using a suitable cost function. In principle any model can be used, if the necessary cost function is derived for the model family.

As a cost function one can use negative log-likelihood of the data given the model, namely $-\ln p(\boldsymbol{x}|H)$. The total model cost L_{tot} needed for comparing variable groupings is the sum of costs of individual variable groups $L_{\mathrm{tot}} = \sum_g L_g = \sum_g -\ln p(\boldsymbol{x}_g|H_g)$, where g is the index of a group of variables, and x_g and H_g are the data and the model related to that variable group, respectively.

In all cases where the IVGA approach is used, the same problem arises: it is computationally infeasible to try all possible different groupings of D variables into G distinct groups, where G varies from 1 to D. Thus, especially when D is large, some heuristic optimization strategy has to be utilized.

In our experiments, all the variables were initially assigned to groups of their own and then moved one by one from group to another if the movement reduced the value of the cost L_{tot}. In addition, every now and then (1) IVGA was recursively run for one group or the union of two groups (the depth of recursion was limited to one) or (2) a merge of two groups was considered.

3.1 The VQ model and the cost function

To demonstrate the approach, we will describe $\mathrm{IVGA}_{\mathrm{VQ}}$ where each variable group is modeled using a different VQ. From now on, we will refer to the cost function of a single variable group leaving out the group index g.

The vector quantization model used for a single variable group consists of codebook vectors $\boldsymbol{\mu}(i),\ i = 1, \ldots, C$ (described by their means and a common variance) and indices of the winners $w(t)$ for each data vector $\boldsymbol{x}(t),\ t = 1, \ldots, N$.

For finding $-\ln p(\boldsymbol{x}|H)$ we use variational EM-algorithm with $\theta = \{\boldsymbol{\mu}, w\}$ as missing observations. In the E-phase, an upper bound of the cost is minimized [2, 5]. The rest of the parameters are included in H, i.e. we use ML estimates for the following: c, the hyper parameter governing the prior probability for a codebook vector to be a winner; $\boldsymbol{\sigma}^2_{\boldsymbol{x}}$, the diagonal elements of the common covariance matrix; and $\boldsymbol{\mu}_\mu$ and $\boldsymbol{\sigma}^2_{\boldsymbol{\mu}}$, the hyper parameters governing the mean and variance of the prior distribution of the codebook vectors.

The upper bound for $-\ln p(\boldsymbol{x}|H) = -\ln \int p(\boldsymbol{x}, \theta|H) d\theta$ is obtained from

$$-\ln \int p(\boldsymbol{x}, \theta|H) d\theta = -\ln \int q(\theta) \tfrac{p(\boldsymbol{x},\theta|H)}{q(\theta)} d\theta \leq -\int q(\theta) \ln \tfrac{p(\boldsymbol{x},\theta|H)}{q(\theta)} d\theta \,,$$

where the inequality follows from the convexity of $-\ln x$ by Jensen's inequality. H and $q(\theta)$ are alternately optimized but instead of finding the intractable global optimum $q(\theta) = p(\theta|\boldsymbol{x}, H)$, we restrict to finding the optimum among the tractable family of distributions of the form $q(\boldsymbol{\mu}, w) = q(\boldsymbol{\mu})q(w)$. Substituting $p(\boldsymbol{x}, \theta|H) = p(\boldsymbol{x}|w, \boldsymbol{\mu}, \boldsymbol{\sigma}^2_{\boldsymbol{x}})p(w|c)p(\boldsymbol{\mu}|\boldsymbol{\mu}_\mu, \boldsymbol{\sigma}^2_{\boldsymbol{\mu}})$, denoting the integral over $q(\theta)$ by $E\{\}$ and arranging the terms then yields

$$L = \overbrace{E\left\{-\ln p(\boldsymbol{x}|w, \boldsymbol{\mu}, \boldsymbol{\sigma}^2_{\boldsymbol{x}})\right\}}^{\text{errors}} + \overbrace{E\left\{\ln \frac{q(w)}{p(w|c)}\right\}}^{\text{winners}} + \overbrace{E\left\{\ln \frac{q(\boldsymbol{\mu})}{p(\boldsymbol{\mu}|\boldsymbol{\mu}_\mu, \boldsymbol{\sigma}^2_{\boldsymbol{\mu}})}\right\}}^{\text{codebook}} .$$

After substituting with the gaussian model for the codebook vectors the cost function becomes[2]

[2] A similar derivation for the cost function for FVQ is given in [3, 7]. For simplicity, we approximate the posterior probability of winners by the best vector only. Without significant additional computational cost all the winners could be taken into account. Individual variances for model vectors could be included as well, and we plan to do this in the future.

$$
\begin{aligned}
L = &\sum_{j=1}^{d}\sum_{t=1}^{N} \frac{\tilde{\mu}_j(w(t)) + (x_j(t) - \bar{\mu}_j(w(t)))^2}{2\sigma_{xj}^2} + \frac{N}{2}\sum_{j=1}^{d} \ln 2\pi\sigma_{xj}^2 - \sum_{t=1}^{N} \ln c(w(t)) \\
&- \frac{Cd}{2} + \sum_{j=1}^{d}\sum_{i=1}^{C} \left[\ln \sqrt{\frac{\sigma_{\mu j}^2(i)}{\tilde{\mu}_j(i)}} + \frac{\tilde{\mu}_j(i) + (\bar{\mu}_j(i) - \mu_{\mu j}(i))^2}{2\sigma_{\mu j}^2(i)} \right].
\end{aligned}
$$

where $\bar{\boldsymbol{\mu}}$ and $\tilde{\boldsymbol{\mu}}$ are the mean and variance of the posterior distribution of the codebook vectors, and d is the dimensionality of the subspace, i.e. the size of the variable group.

Minimization of the cost function. Minimization of Eq. 1 is carried out so that C is fixed and the cost is minimized with respect to each of the variables $c, w, \boldsymbol{\sigma}_{\boldsymbol{x}}^2, \boldsymbol{\mu}_\mu, \boldsymbol{\sigma}_{\boldsymbol{\mu}}^2, \bar{\boldsymbol{\mu}}, \tilde{\boldsymbol{\mu}}$. The following steps are repeated iteratively as long as the value of the cost function decreases or maximum iteration count is reached (in our experiments 100 iterations). This is repeated for various values of C.

1. *Winner selection.*

$$
\frac{\partial L}{\partial w(t)} = 0 \Longrightarrow w(t) = \arg\min_i \left\{ -\ln c(i) + \sum_{j=1}^{d} \frac{\tilde{\mu}_j(i) + (x_j(t) - \bar{\mu}_j(i))^2}{2\sigma_{xj}^2} \right\}
$$

2. *Update of the winner.*
   (a) *Update of the posterior mean of the codebook vector.*

$$
\frac{\partial L}{\partial \bar{\mu}_j(i)} = 0 \Longrightarrow \bar{\mu}_j(i) = \frac{\sigma_{xj}^2 \mu_{\mu j} + \sigma_{\mu j}^2 \sum_{w(t)=i} x_j(t)}{\sigma_{xj}^2 + \sigma_{\mu j}^2 f(i)}
$$

   (b) *Update of the posterior variance of the codebook vector.*

$$
\frac{\partial L}{\partial \tilde{\mu}_j(i)} = 0 \Longrightarrow \tilde{\mu}_j(i) = \frac{\sigma_{xj}^2 \sigma_{\mu j}^2}{\sigma_{xj}^2 + \sigma_{\mu j}^2 f(i)}
$$

3. *Update of the data variance.*

$$
\frac{\partial L}{\partial \sigma_{xj}^2} = 0 \Longrightarrow \sigma_{xj}^2 = \frac{1}{N}\sum_{t=1}^{N} \left[\tilde{\mu}_j(w(t)) + (x_j(t) - \bar{\mu}_j(w(t)))^2 \right]
$$

4. *Updates of codebook frequency prior and parameters of the prior distribution.*

$$
c(i) = \frac{f(i)+1}{N+C}, \quad \mu_{\mu j} = \frac{1}{C}\sum_{i=1}^{C} \bar{\mu}_j(i), \quad \sigma_{\mu j}^2 = \frac{1}{C}\sum_{i=1}^{C} [\tilde{\mu}_j(i) + (\bar{\mu}_j(i) - \mu_{\mu j})^2]
$$

Here $f(i)$ is the number of hits of $\boldsymbol{\mu}(i)$, i.e., $f(i) = \#\{t | w(t) = i\}$.

4 Experiments

Four small experiments were run both to (1) verify the general IVGA principle (Experiments 1 vs. 2), i.e, that it is useful to model independent variable groups separately, and (2) to study the performance of the presented IVGA_{VQ} algorithm in optimizing the model for the data and in finding independent variable groups (Experiments 1 vs. 3, Experiments 2 vs. 4).

In each experiment, the best one out of several trials is reported (for VQs this meant about 10000 trials, for IVGA_{VQ}, considerably fewer). With each approach we used a roughly equal amount of computation time. In reporting results we report the number of codebook vector parameters—the number of the rest of the parameters is substantially smaller and constant in all experiments.

Data and variables. As a data set we used features from 1000 images that have been used for content-based image retrieval in [6][3]. Each image is represented by a vector of 144 features (variables). The features fall naturally into three categories related to their origin: FFT, RGB, and texture. It is reasonable to assume that dependences between variables in different categories are weak[4]. Thus, the data provides an ideal validation for both the IVGA_{VQ} algorithm as well as the general IVGA approach. For the experiments we chose randomly a subset of 50 features: the variable set A consisted of 27 FFT features (numbered 1–27 below), B of 7 RGB features (28–34) and C of 16 texture features (35–50).

Experiment 1: VQ(A+B+C). A single VQ was used to model the combined set A+B+C of 50 variables. The optimized number of codebook vectors was 44 (number of parameters was thus $44 \times 50 = 2200$) and the cost was -138115.5.

Experiment 2: VQ(A)+VQ(B)+VQ(C). The sets A, B and C were each modeled with a separate VQ. For set A, 16 codebook vectors were used (cost -116397.7), for B, 19 codebook vectors (cost -3921.6) and for C, 30 codebook vectors (cost -25476.9). The total number of parameters was 1045 and the total cost was -145796.2. The model cost improved compared to Experiment 1 which shows that the general IVGA approach is feasible.

Experiment 3: $\mathbf{IVGA_{VQ}}$(A+B+C). The feature sets A, B and C were concatenated and IVGA_{VQ} was applied for modeling and grouping. The results are presented in Table 1. Twelve groups were found, and in terms of grouping the result was excellent: each group contained variables from one variable category only. Furthermore, the improved model cost shows that presented IVGA_{VQ} algorithm performs well in the IVGA task.

[3] We are grateful to the PicSOM project members for kindly providing us with the image data set.

[4] This assumption has been made in [6] prior to modeling each variable category with a different SOM.

Table 1. Results of the grouping when $\mathrm{IVGA}_{\mathrm{VQ}}$ was run for the combined data set A+B+C. Negative costs are due to leaving out the constant factor in Eq. 1.

Group	Variables	Belong to data set	Cost	Codebook vectors	Parameters
1	1,6	A	-4760.6	7	14
2	2–5,7–8,11	A	-33213.3	10	70
3	9–10,13,15–16,18–21,24–25	A	-50997.9	12	132
4	12,14,17	A	-14201.3	8	24
5	22,26	A	-8233.6	5	10
6	23,27	A	-4920.4	10	20
7	28–34	B	-3889.7	20	140
8	35–38	C	-6678.2	7	28
9	39–42	C	-7248.3	15	60
10	43–44	C	-3140.6	9	18
11	45–46	C	-2708.1	13	26
12	47–50	C	-7215.0	19	76
Total			**-147206.8**	**135**	**618**

Experiment 4: $\mathbf{IVGA_{VQ}(A)}$ + $\mathbf{IVGA_{VQ}(B)}$ + $\mathbf{IVGA_{VQ}(C)}$ Next, we simulated a situation where the three main groups of variables (A, B and C) are known based on prior information. Now the $\mathrm{IVGA}_{\mathrm{VQ}}$ algorithm was used to find the best possible sub-groupings for A, B and C separately. The results are shown in Table 2. Note also that the variable groupings obtained are almost, but not exactly, identical in Experiments 3 and 4. The slight improvement in total cost compared to Experiment 3 is as expected, since computational resources needed not be allocated for separating the groups A, B and C.

The results of the four experiments are summarized in Table 3.

Table 2. Results when $\mathrm{IVGA}_{\mathrm{VQ}}$ was run separately for each set A, B and C.

Variable set	Group	Variables	Cost	Codebook vectors	Parameters
A	1	1,6	-4773.4	9	18
A	2	2–5, 7–8, 11	-33432.8	12	84
A	3	9–10,13,16,18,20,25	-32122.6	13	91
A	4	12,14–15,17,19,21–24,26–27	-46778.0	14	154
B	1	28–34	-3868.6	17	119
C	1	35–38	-6646.1	8	32
C	2	39–42,45–46	-9934.4	26	156
C	3	43–44	-3190.7	7	14
C	4	47–50	-7187.7	11	44
Total			**-147934.3**	**117**	**712**

Table 3. Summary of results of all experiments.

Experiment	Total cost	#VQs	Parameters
1. VQ(A+B+C)	-138115.5	1	2200
2. VQ(A) + VQ(B) + VQ(C)	-145796.2	3	1045
3. $\mathrm{IVGA}_{\mathrm{VQ}}$(A+B+C)	-147206.8	12	618
4. $\mathrm{IVGA}_{\mathrm{VQ}}$(A) + $\mathrm{IVGA}_{\mathrm{VQ}}$(B) + $\mathrm{IVGA}_{\mathrm{VQ}}$(C)	-147934.3	9	712

5 Conclusions

In this article, the basic idea of IVGA, i.e., grouping variables according to dependences within the data, was presented and motivated, as well as shown to work in practice. An efficient algorithm was suggested for computing IVGA and shown to work rather well on real data. In the experiments vector quantization (VQ) was used to model the dependences within variable groups. In the calculation of the cost function most of the unknown variables are marginalized out and therefore the cost function can reliably be used for model selection.

One should keep in mind that we used VQ just as a simple example: it could be replaced with any method as long as the necessary cost function is derived for the method. Each variable group could even be modeled using a different method.

The presented approach has implications both for constructing better models of large and complex phenomena and for the efficient computation of such models. However, further research is needed to study these issues in depth.

References

1. J.-F. Cardoso. Multidimensional independent component analysis. In *Proceedings of ICASSP'98*, Seattle, 1998.
2. G. E. Hinton and D. van Camp. Keeping neural networks simple by minimizing the description length of the weights. In *Proceedings of the COLT'93*, pp. 5–13, Santa Cruz, California, USA, July 26–28, 1993.
3. G. E. Hinton and R. S. Zemel. Autoencoders, minimum description length and Helmholtz free energy. In J. et al, ed., *Neural Information Processing Systems 6*, San Mateo, CA, 1994. Morgan Kaufmann.
4. A. Hyvärinen and P. Hoyer. Emergence of phase and shift invariant features by decomposition of natural images into independent feature subspaces. *Neural Computation*, 12(7):1705–1720, 2000.
5. H. Lappalainen and J. W. Miskin. Ensemble learning. In M. Girolami, ed., *Advances in Independent Component Analysis*, pp. 76–92. Springer-Verlag, Berlin, 2000.
6. E. Oja, J. Laaksonen, M. Koskela, and S. Brandt. Self-organizing maps for content-based image database retrieval. In E. Oja and S. Kaski, eds., *Kohonen Maps*, pp. 349–362. Elsevier, 1999.
7. R. S. Zemel. *A Minimum Description Length Framework for Unsupervised Learning.* PhD thesis, University of Toronto, 1993.