HIERARCHICAL MODELS OF VARIANCE SOURCES

Harri Valpola, Markus Harva and Juha Karhunen

Helsinki University of Technology, Neural Networks Research Centre
P.O. Box 5400, FIN-02015 HUT, Espoo, Finland
firstname.lastname@hut.fi http://www.cis.hut.fi/projects/ica/bayes/

ABSTRACT

In many models, variances are assumed to be constant although this assumption is known to be unrealistic. Joint modelling of means and variances can lead to infinite probability densities which makes it a difficult problem for many learning algorithms. We show that a Bayesian variational technique which is sensitive to probability mass instead of density is able to jointly model both variances and means. We discuss a model structure where a Gaussian variable which we call variance neuron controls the variance of another Gaussian variable. Variance neuron makes it possible to build hierarchical models for both variances and means. We report experiments with artificial data which demonstrate the ability of learning algorithm to find the underlying explanations—variance sources—for the variance in the data. Experiments with MEG data verify that variance sources are present in real-world signals.

1. INTRODUCTION

Most unsupervised learning techniques model the changes in the mean of different quantities while variances are assumed constant. This assumption is often known to be invalid but suitable techniques for jointly estimating both means and variances have been lacking. The basic problem is that if the mean is modelled by a latent variable model such as independent component analysis (ICA) [1], the modelling error of any single observation can be made zero. If the learning method is based on maximising likelihood or posterior density, it runs into problems when trying to simultaneously estimate the variance as the density will become infinite when the variance approaches zero.

In this paper we show how the problem can be solved by variational Bayesian learning. We are able to jointly estimate both the means and the variances by a hierarchical model because the learning criterion is based on posterior probability mass rather than the problematic probability density. The case mentioned above no longer poses problems because when variance approaches zero, posterior probability will have an increasingly higher but at the same time narrower peak. The narrowing of the peak compensates the higher density and results in a well behaved posterior probability mass.

The basic method used here was reported in [2]. There a set of building blocks that can be used to construct various latent variable models was introduced. In this paper we concentrate on how to build models of variance from Gaussian variables and linear mappings.

In Section 2, we introduce the variance neuron, a Gaussian variable which converts predictions of mean into predictions of variance and discuss various models which utilise it. Section 3 shows how these models are learned. Experiments where such a model is applied to artificial and natural data are reported in Section 4.

2. VARIANCE NEURON

Variance neuron [2] is a time-dependent Gaussian variable $u(t)$ which specifies the variance of another time-dependent Gaussian variable $\xi(t)$:

$$\xi(t) \sim N(m(t), \exp[-u(t)]) \,, \tag{1}$$

where $N(\mu, \sigma^2)$ is the Gaussian distribution and $m(t)$ is the prediction for the mean of $\xi(t)$ given by other parts of the model. As can be seen from (1), $u(t) = -\log \sigma^2$. This parametrisation is justified in Section 3.

Variance neurons are useful as such for modelling super-Gaussian distributions because a Gaussian variable ξ whose variance has fluctuations over time generates a super-Gaussian distribution (see e.g. [3]). Variance neurons alone cannot generate sub-Gaussian distributions[1], but in many cases sub-Gaussian models are not needed. This is particularly true in connection with dynamics. Real signals such as oscillations have sub-Gaussian distributions but their innovation processes are almost invariably super-Gaussian. A linear ICA model with super-Gaussian source distributions generated by Gaussian sources $\mathbf{s}$ with variance neurons $\mathbf{u}_s$ attached to each source is depicted in Fig. 1(a).

From the point of view of other parts of the model which predict the value of the variance neuron, the variance neuron is as any other Gaussian variable. This

This research has been funded by the European Commission project BLISS, and the Finnish Center of Excellence Programme (2000–2005) under the project New Information Processing Principles.

[1] Mixture-of-Gaussian distributions can be used for sub-Gaussian distributions. See e.g. [4].

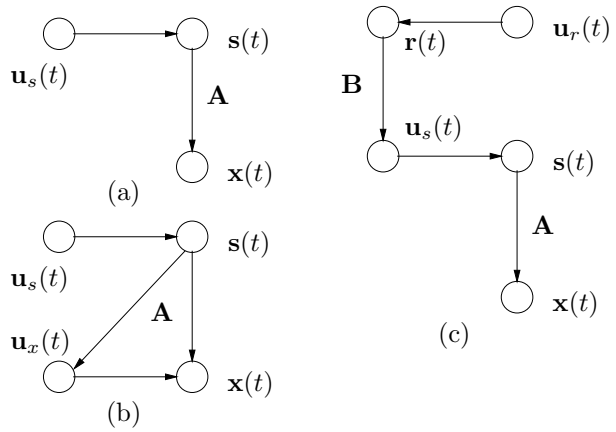

Fig. 1. Various model structures utilising variance neurons. Observations are denoted by $\mathbf{x}$, linear mappings by $\mathbf{A}$ and $\mathbf{B}$, sources by $\mathbf{s}$ and $\mathbf{r}$ and variance neurons by $\mathbf{u}$.

means that it enables to translate a conventional model of mean into a model of variance. A simple extension of ICA which utilises variance neurons in this way is shown in Fig. 1(b). The sources can model concurrent changes in both the observations $\mathbf{x}$ and the modelling errors of the observations through the variance neurons $\mathbf{u}_x$. Such a structure would be useful for instance in a case where a source characterises the rotation speed of a machine. It is plausible that the rotation speed affects the mean of a set of variables and the modelling error of another, possibly overlapping set of variables.

In this paper we present experiments with a hierarchical extension of the linear ICA model, shown in Fig. 1(c). The concurrent changes in the variance of conventional sources are modelled by higher-order variance sources. As a special case, this model structure is able to perform subspace ICA [5, 6, 7, 1]. In that case, each conventional source would be modelled by only one of the variance sources, i.e. the mapping $\mathbf{B}$ would have only one non-zero entry on each row. Moreover, usually each subspace has equal dimension, i.e. each column of $\mathbf{B}$ has an equal number of non-zero entries. We are not going to impose such restrictions. The effects of variance sources can thus be overlapping.

Just as conventional sources of time-series data have temporal structure [1], variance sources of such data can be expected to change slowly, in fact, more slowly than the conventional sources. This is because the variance sources have similarity to the invariant features extracted by adaptive subspace SOM [8] and other related models, e.g. [7]. This is demonstrated in the experiment with magnetoencephalographic data in Section 4.

3. VARIATIONAL BAYESIAN LEARNING

Variational Bayesian learning techniques are based on approximating the true posterior probability density of the unknown variables of the model by a function with a restricted form. Currently the most common technique is ensemble learning where Kullback-Leibler divergence measures the misfit between the approximation and the true posterior. It has been applied to ICA and a wide variety of other models (see, e.g. [9, 4, 10, 11, 12, 13, 14]). An example of a variational technique other than ensemble learning can be found in [15].

In ensemble learning, the posterior approximation is required to have a suitably factorial form. During learning, the factors are typically updated one at a time while keeping others fixed. Here we use the method introduced in [2]. The posterior has a maximally factorial form, i.e. each unknown variable is approximated to be independent a posteriori of the rest of the variables. The computational complexity of each individual update is then proportional to the number of connections it has with other variables. Consequently, the update of the posterior variance of all variables in the model can be accomplished in time proportional to the total number of connections in the model.

For each update of the posterior approximation $q(\theta)$, the variable θ requires the prior distribution $p(\theta \mid \text{parents})$ given by its parents and the likelihood $p(\text{children} \mid \theta)$ obtained from its children. The relevant part of the Kullback-Leibler divergence to be minimised is

$$C(q(\theta)) = \left\langle \ln \frac{q(\theta)}{p(\theta \mid \text{parents}) p(\text{children} \mid \theta)} \right\rangle_q , \quad (2)$$

where the expectation is taken over the posterior approximations $q(\theta_i)$ of all unknown variables.

In ensemble learning, conjugate priors are commonly used because they make it very easy to solve the variational minimisation problem of finding the optimal $q(\theta)$ which minimises (2).

As an example, consider linear mappings with Gaussian variables. First, note that in (2), the negative logarithm of the prior and likelihood is needed. We shall call this quantity the potential. Gaussian prior has a quadratic potential. The likelihood arising from a linear mapping to Gaussian variables also has a quadratic potential. The sum of the potential is quadratic and the optimal posterior approximation can be shown to be the Gaussian distribution whose potential has the same second and first order terms. The minimisation thus boils down to adding the coefficients of second and first order terms of the prior and likelihood.

The likelihood which variance neuron receives from the Gaussian node whose logarithmic variance is modelled has a linear term and an exponential term. The commonly used parametrisation is inverse variance as then the potential corresponds to Gamma-distribution and hence a Gamma-prior yields a Gamma-posterior. It would, however, be difficult to build a hierarchical model with Gamma-distributed variables and therefore we choose to have a Gaussian prior and parametrise the variance on logarithmic scale. The resulting sum potential has both a quadratic term (from the prior) and an exponential term (from the likelihood), but it

is well approximated by a Gaussian posterior $q(\theta) \stackrel{def}{=} N(\theta; m, v)$. It can be shown [2] that in this case (2) equals

$$C(m,v) = Mm + V[m^2 + v] + E\exp(m + v/2) - \frac{1}{2}\ln v + \text{const}\,, \quad (3)$$

where M, V and E are the coefficients of the terms in the mixed potential. The optimisation method for the mixed potential is derived in Appendix A.

4. EXPERIMENTS

In this section, experiments with artificial data and real magnetoencephalographic (MEG) data are reported.

4.1. Model structure

According to the model used for the experiments, the observations are generated by conventional source vectors $\mathbf{s}(t)$ mapped linearly to the observation vectors $\mathbf{x}(t)$ which are corrupted by additive Gaussian noise $\mathbf{n}(t)$. For each source $s_i(t)$ there is a variance neuron $u_{si}(t)$ which represents the negative logarithm of the variance. The values of the variance neurons $\mathbf{u}_s(t)$ are further modelled by higher-level variance sources $\mathbf{r}(t)$ which map linearly to the variance neurons. Variance sources, too, have variance neurons $\mathbf{u}_r(t)$ attached to them.

$$\mathbf{x}(t) = \mathbf{A}\mathbf{s}(t) + \mathbf{n}(t) \quad (4)$$
$$s_i(t) \sim N(\mu_{si}(t), \exp u_{si}(t)) \quad (5)$$
$$\mathbf{u}_s(t) = \mathbf{B}\mathbf{r}(t) + \mathbf{m}(t) \quad (6)$$
$$r_i(t) \sim N(r_i(t-1), \exp u_{ri}(t)) \quad (7)$$

The additive Gaussian noise terms $\mathbf{n}(t)$ and $\mathbf{m}(t)$ are allowed to have non-zero bias. The model structure is shown in Fig. 1(c). Note that it makes sense to have two layers although the model is linear and all variables are Gaussian since the variance neurons $\mathbf{u}_s$ translate the higher-order source model into a prediction of variance. The variance sources are also responsible for generating super-Gaussian distributions for $\mathbf{s}(t)$ and $\mathbf{r}(t)$.

As (7) shows, the variance sources have a dynamic model. The predicted mean for the variance source was taken to be the value at the previous time instant. In the artificial data, $\mu_{si}(t)$ in (6) equals zero, but the MEG signals have strong temporal dependences and we used $\mu_{si}(t) = s_i(t-1)$.

4.2. Learning scheme

The basic operations in learning were iteration and pruning. An iteration consisted of updating the posterior approximation $q(\cdot)$ for each latent variable, one at a time. Pruning involves going through the parameters of the linear mappings and removing them from the model if that resulted in a decrease of the cost function. In ensemble learning, the cost function gives a lower bound for the model evidence and thus enables this kind of pruning.

The model was built in two stages. First, only the conventional sources were estimated, i.e. the model structure was as in Fig. 1(a). The sources were initialised using PCA components calculated from the data. If the source model had dynamics, low-pass filtering was applied to the data before PCA. During the first few iterations the sources were kept fixed to the initialisation in order to have a reasonable estimate for the mixing matrix $\mathbf{A}$. Learning was then continued for two hundred iterations to find reasonable values for the variance neurons.

After that, the second layer was added. Initialisation for the variance neurons was similar to the conventional ones except instead of a few iterations, they were kept fixed for two hundred iterations. Learning was continued until the changes in the parameters were very small.

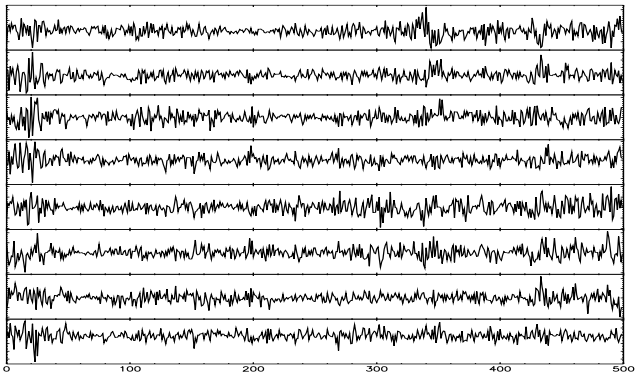

Fig. 2. Artificial data $\mathbf{x}(t)$ (8 out of 20 time series).

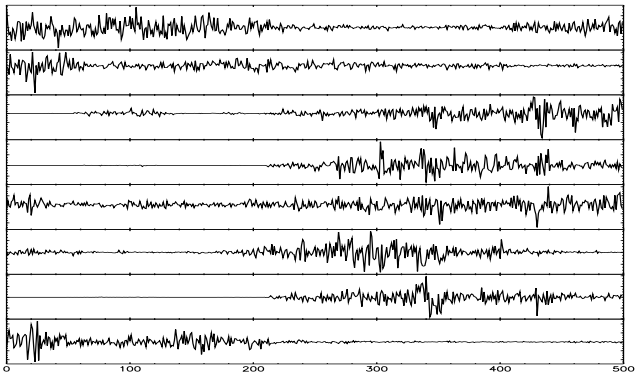

Fig. 3. Sources $\mathbf{s}(t)$ estimated from the artificial data (8 out of 20 sources).

4.3. Artificial data

In order to test the learning algorithms, we generated data that fits the model structure. There were two variance sources $\mathbf{r}(t)$ and 20 conventional sources $\mathbf{s}(t)$. The mappings $\mathbf{A}$ and $\mathbf{B}$ were sampled from normal distri-

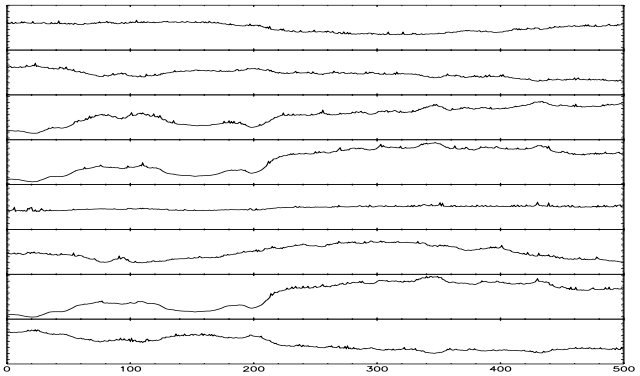

Fig. 4. Variance neurons $\mathbf{u}_s(t)$ corresponding to the sources shown in Fig. 3.

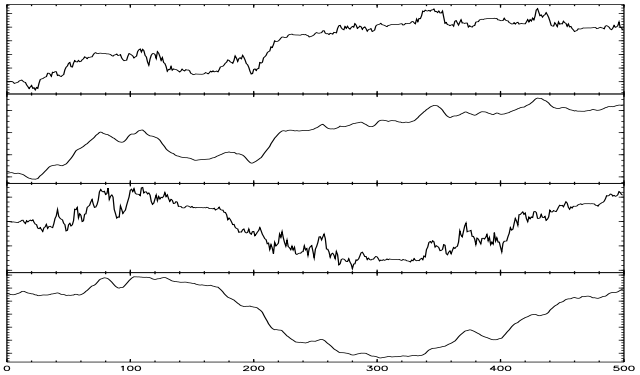

Fig. 5. Estimated variance sources $\mathbf{r}(t)$ (2nd and 4th row) which model the regularities found in the variance neurons of Fig. 4. The corresponding true underlying variance source are shown on 1st and 3rd row.

bution. The biases for the additive noise of $\mathbf{u}_s(t)$ were chosen such that the sources $\mathbf{s}(t)$ had unit variances. Part of the generated signals are shown in Fig. 2.

The linear mappings were known to be fully connected and therefore no pruning was applied. The results after 10,000 iterations are depicted in Figs. 3–5. The estimated posterior mean of $q(\cdot)$ for each quantity is shown. As can be seen in Figure 5, the estimated variance sources are very close to the true underlying variance sources which were used for generating the data. In general, a reliable estimate of variance needs more observations than the estimate of mean. This is reflected in the fact that small random variations in the variance source are not captured although on larger time scale the estimate is accurate.

4.4. Biomedical data

In these experiments, we used part of the MEG data set [16]. The data consists of signals originating from brain activity. The signals are contaminated by external artefacts such as a digital watch, heart beat as well as eye movements and blinks. We used 2,500 samples of the original data set. The most prominent feature in this area is the biting artefact where muscle activity

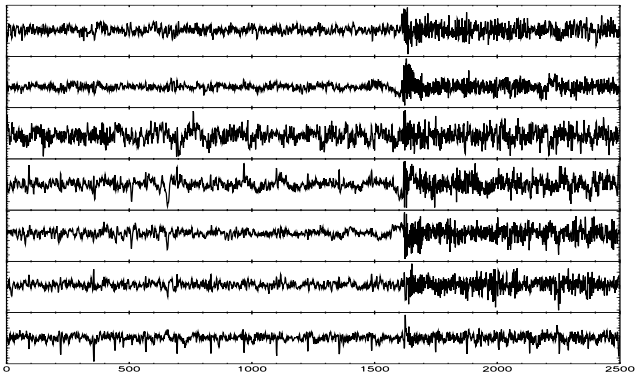

Fig. 6. MEG recordings (seven out of 122 time series).

contaminates many of the channels starting after 1,600 samples as can be seen in Fig. 6.

Initially the model had 40 sources $\mathbf{s}(t)$ and 10 variance sources $\mathbf{r}(t)$. Starting after 5,000 iterations, the linear mappings were pruned every 200 iterations. This resulted in two of the variance sources losing all their out-going connections after which they were removed. After 10,000 iterations, pruning was applied to variance neurons instead of the parameters of the linear mappings. The three surviving variance sources are shown in Fig. 9. None of the conventional sources lost all their out-going connections and they survived even when pruning was applied to them directly after 10,000 iterations. The sources and their variance neurons are depicted in Figs. 7 and 8, respectively.

The conventional sources are comparable to those reported in the literature for this data set [16]. The first variance source in Fig. 9 clearly models the biting artefact. This variance source integrates information from several conventional sources and its activity varies very little over time. This is partly due to the dynamics but experiments with a static model confirm that the variance source acts as an invariant feature which reliably detects the biting artefact.

The second variance source appears to represent increased activity during the onset of the biting. The third variance neuron seems to be related to the amount of rythmic activity on the sources. Two such sources can be found in Fig. 7 (third and fourth source). Interestingly, it seems that the amount of rythmic activity on these sources is negatively correlated.

5. DISCUSSION

In statistics, a distribution characterised by changing variance is called heteroskedastic. Heteroskedasticity is known to be commonplace and there are various techniques for modelling the variance (see e.g. [17]). However, previously mean has been estimated separately from variance in order to avoid problems related to infinite probability densities. We have shown that it is possible to estimate both mean and variance jointly. This has the benefit that the estimation of the mean can use the information about the variance and vice

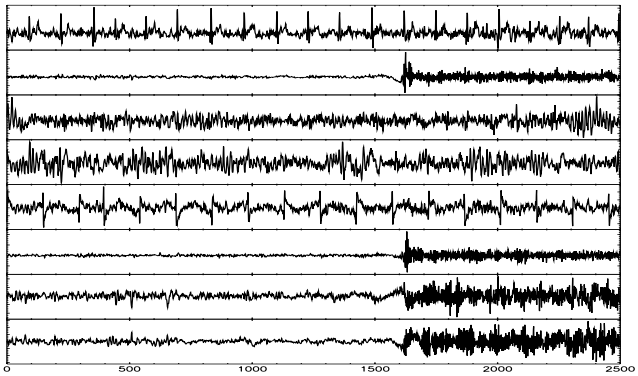

Fig. 7. Sources estimated from the MEG data (eight out of 40 sources).

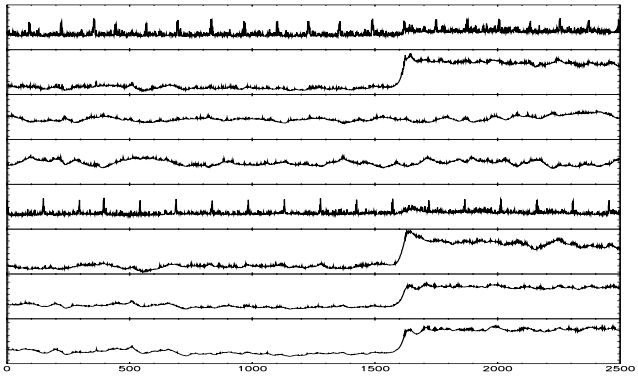

Fig. 8. Variance neurons corresponding to the sources shown in Fig. 7.

versa. In the experiments reported here, this implies that estimation of the sources can utilise the information provided by the variance sources.

We reported experiments with one model structure which utilises variance neurons but we have only touched the tip of an iceberg. Since the variance neurons allow to translate models of mean into models of variance, we can go through a large number of models discussed in the literature and consider whether they are useful for modelling variance. The cost function used in ensemble learning is very useful in this task as it allows model comparison.

6. REFERENCES

[1] A. Hyvärinen, J. Karhunen, and E. Oja, *Independent Component Analysis*. J. Wiley, 2001.

[2] H. Valpola, T. Raiko, and J. Karhunen, "Building blocks for hierarchical latent variable models," in *Proc. Int. Conf. on Independent Component Analysis and Signal Separation (ICA2001)*, (San Diego, USA), pp. 710–715, 2001.

[3] L. Parra, C. Spence, and P. Sajda, "Higher-order statistical properties arising from the non-stationarity of natural signals," in *Advances in Neural Information Processing Systems 13* (T. Leen, T. Dietterich, and V. Tresp, eds.), (Cambridge, MA, USA), pp. 786–792, The MIT Press, 2001.

[4] H. Attias, "Independent factor analysis," *Neural Computation*, vol. 11, no. 4, pp. 803–851, 1999.

[5] L. De Lathauwer, B. De Moor, and J. Vandewalle, "Fetal electrocardiogram extraction by source subspace separation," in *Proc. IEEE Sig. Proc. / ATHOS Workshop on Higher-Order Statistics*, pp. 134–138, 1995.

[6] J.-F. Cardoso, "Multidimensional independent component analysis," in *Proc. IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP'98)*, (Seattle, Washington, USA, May 12–15), pp. 1941–1944, 1998.

[7] A. Hyvärinen and P. 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.

[8] T. Kohonen, S. Kaski, and H. Lappalainen, "Self-organized formation of various invariant-feature filters in the Adaptive-Subspace SOM," *Neural Computation*, vol. 9, no. 6, pp. 1321–1344, 1997.

[9] D. Barber and C. Bishop, "Ensemble learning for multi-layer networks," in *Advances in Neural Information Processing Systems 10* (M. Jordan, M. Kearns, and S. Solla, eds.), pp. 395–401, Cambridge, MA, USA: The MIT Press, 1998.

[10] J. Miskin and D. MacKay, "Ensemble learning for blind image separation and deconvolution," in *Advances in Independent Component Analysis* (M. Girolami, ed.), pp. 123–141, Springer-Verlag, 2000.

[11] Z. Ghahramani and G. E. Hinton, "Variational learning for switching state-space models," *Neural Computation*, vol. 12, no. 4, pp. 963–996, 2000.

[12] R. Choudrey, W. Penny, and S. Roberts, "An ensemble learning approach to independent component analysis," in *Proc. of the IEEE Workshop on Neural Networks for Signal Processing, Sydney, Australia, December 2000*, IEEE Press, 2000.

[13] K. Chan, T.-W. Lee, and T. Sejnowski, "Variational learning of clusters of undercomplete nonsymmetric independent components," in *Proc. Int. Conf. on Independent Component Analysis and Signal Separation (ICA2001)*, (San Diego, USA), pp. 492–497, 2001.

[14] H. Valpola and J. Karhunen, "An unsupervised ensemble learning method for nonlinear dynamic state-space models," *Neural Computation*, vol. 14, no. 11, pp. 2647–2692, 2002.

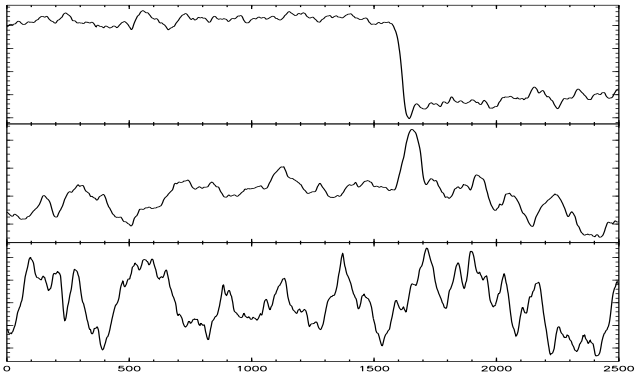

Fig. 9. Variance sources which model the regularities found in the variance neurons shown in Fig. 8.

[15] M. Girolami, "Variational method for learning sparse and overcomplete representations," *Neural Computation*, vol. 13, no. 11, pp. 2517–2532, 2001.
[16] R. Vigário, J. Särelä, V. Jousmäki, M. Hämäläinen, and E. Oja, "Independent component approach to the analysis of EEG and MEG recordings," *IEEE transactions on biomedical engineering*, vol. 47, no. 5, pp. 589–593, 2000.
[17] T. Bollerslev, "Generalized autoregressive conditional heteroskedasticity," *Journal of Econometrics*, vol. 31, pp. 307–327, 1986.

A. MINIMISING MIXED POTENTIALS

Here we show how to minimise a function

$$C(m,v) = Mm + V[m^2+v] + E\exp(m+v/2) - \frac{1}{2}\ln v\,.$$

A unique solution exists when $V > 0$ and $E > 0$. This problem occurs when a Gaussian posterior with mean m and variance v is fitted to a probability distribution whose logarithm has both a quadratic and exponential part resulting from Gaussian prior and log-Gamma likelihoods, respectively, and Kullback-Leibler divergence is used as the measure of the misfit.

The minimisation is iterative. At each iteration, one Newton-iteration step for m and one fixed-point iteration step for v is performed. The steps are taken until they become smaller than a predefined threshold.

A.1. Newton iteration for m

Newton iteration for m is obtained by

$$m_{i+1} = m_i - \frac{\partial C(m_i,v_i)/\partial m_i}{\partial^2 C(m_i,v_i)/\partial m_i^2} = m_i - \frac{M + 2Vm_i + E\exp(m_i+v_i/2)}{2V + E\exp(m_i+v_i/2)} \quad (8)$$

Newton iteration converges in one step if the second derivative remains constant. The step is too short if the second derivative decreases and too long if the second derivative increases. For stability, it is better to take too short than too long steps.

In this case, the second derivative always decreases if m decreases and vice versa. For stability it is therefore useful to restrict the increases in m because the increases are consistently over-estimated. We have found that restricting the increase to be at most four yields robust convergence.

A.2. Fixed-point iteration for v

A simple fixed-point iteration rule is obtained for v by solving the zero of the derivative:

$$0 = \frac{\partial C(m,v)}{\partial v} = V + \frac{E}{2}\exp(m+v/2) - \frac{1}{2v} \Leftrightarrow v = \frac{1}{2V + E\exp(m+v/2)} \stackrel{def}{=} g(v) \quad (9)$$

$$v_{i+1} = g(v_i) \quad (10)$$

In general, fixed-point iterations are stable around the solution v_{opt} if $|g'(v_{\text{opt}})| < 1$ and converge the best when the derivative $g'(v_{\text{opt}})$ is near zero. In our case $g'(v_i)$ is always negative and can be less than -1, i.e. the solution can be an unstable fixed-point. This can be remedied by taking a weighted average of (10) and a trivial iteration $v_{i+1} = v_i$:

$$v_{i+1} = \frac{\xi(v_i)g(v_i) + v_i}{\xi(v_i)+1} \stackrel{def}{=} f(v_i) \quad (11)$$

The weight ξ should be such that the derivative of f is close to zero at the optimal solution v_{opt} which is achieved exactly if $\xi(v_{\text{opt}}) = -g'(v_{\text{opt}})$.

It holds

$$g'(v) = -\frac{E/2\exp(m+v/2)}{[2V + E\exp(m+v/2)]^2} = g^2(v)\left[V - \frac{1}{2g(v)}\right] = g(v)\left[Vg(v) - \frac{1}{2}\right] \Rightarrow g'(v_{\text{opt}}) = v_{\text{opt}}\left[Vv_{\text{opt}} - \frac{1}{2}\right] \Rightarrow \xi(v_{\text{opt}}) = v_{\text{opt}}\left[\frac{1}{2} - Vv_{\text{opt}}\right]. \quad (12)$$

The last steps follow from the fact that $v_{\text{opt}} = g(v_{\text{opt}})$ and the requirement that $f'(v_{\text{opt}}) = 0$. We can assume that v is close to v_{opt} and use

$$\xi(v) = v\left[\frac{1}{2} - Vv\right]. \quad (13)$$

Note that the iteration (10) can only yield estimates with $0 < v_{i+1} < 1/2V$ which means that $\xi(v_{i+1}) > 0$. Therefore the step defined by (11) is always shorter than the step defined by (10).

Since we know that the solution lies between 0 and $1/2V$, we can set $v_0 = 1/2V$ if the current estimate is greater than $1/2V$.

In order to improve stability, step sizes need to be restricted. Increases in v are more problematic than decreases since the $\exp(m + v/2)$ term behaves more nonlinearly when v increases. Again, we have found experimentally that restricting the increase to be at most four yields robust convergence.

A.3. Summary of the iteration

1. Set $v_0 \leftarrow min(v_0, 1/2V)$.
2. Iterate
   (a) Solve new m by (8) under the restriction that the maximum step is 4
   (b) Solve new v by (13) and (11) under the restriction that the maximum step is 4

   until both steps are smaller than 10^{-4}.