Latent Variable Models for Gene Expression Data

David J.C. MacKay & James Miskin

Cavendish Laboratory, Madingley Road, Cambridge CB3 0HE. United Kingdom.

mackay@mrao.cam.ac.uk

April 25, 2001 — Draft 1.3

Abstract

This note describes the assumptions underlying the latent-variable-modelling work of Miskin, Martoglio and MacKay for microarrays.

The aim is to give a non-technical summary of the model assumptions and the computational methods. For further information, the thesis of James Miskin (2001) should be consulted.

1 The Model

We assume that the data to be modelled is a matrix $\mathbf{D}$ of gene expression levels in different tissues. The entry d_{tg} gives the expression level of gene g in tissue t.

[We recognise that the raw data corresponding to one such measurement d_{tg} typically consists of one or more *pairs* of measurements, from which d_{tg} is derived by taking averages and ratios; ultimately we believe the most accurate results will be obtained by Bayesian modelling of the full data set rather than the derived data $\mathbf{D}$. We will return to this important matter later.]

A cartoon of such a data matrix $\mathbf{D}$ is shown below. There are $G = 37$ genes and $T = 7$ tissues.

How should we model such data? Many modelling methods have been applied (such as 'clustering' of genes, or principal components analysis) that have little justification apart from empirical utility. We have chosen a latent variable model in which the latent variables describe features that underlie the data. Within this model space our methods seek the most efficient description of the data. Our model is simple – it is closely related to factor analysis (Everitt, 1984) and independent component analysis (Bell and Sejnowski, 1995; MacKay, 1996) – and we do not claim it is the best model; but because it is a Bayesian model, the assumptions on which the model rests are clear, the model is readily extensible should those assumptions be modified, and the model is readily comparable with alternative quantified models, because our model quantifies how well it models the data.

[We quantify how efficiently a model $\mathcal{H}$ describes the data by the number of bits into which the data could be compressed by the model. This number of bits $L(D|\mathcal{H})$ is intimately related to the probability of the data given the model, $P(D|\mathcal{H})$ by $L(D|\mathcal{H}) = \log_2 1/P(D|\mathcal{H})$.]

We assume that gene expression levels vary under the influence of H latent variables, $a_1, a_2, \ldots, a_H$. [How many latent variables there are, H, is something we will attempt to

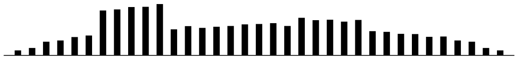
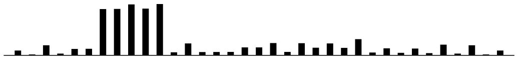
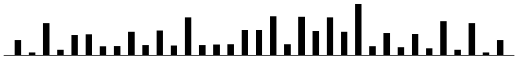
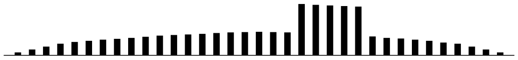
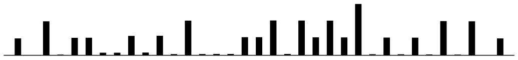
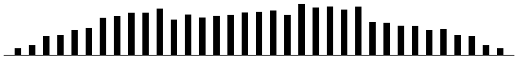
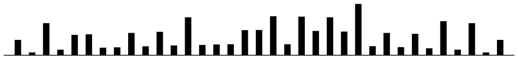
Tissue number, t	Expression levels of genes $g = 1 \ldots 37$
$t = 1$	
$t = 2$	
$t = 3$	
$t = 4$	
$t = 5$	
$t = 6$	
$t = 7$	

Figure 1: Mock data.

infer from the data.] Each gene expression level may be influenced by one or more of these latent variables. Which genes are influenced by the hth latent variable, and how strongly, is described by a vector b_{hg}, with $g = 1 \ldots G$. This vector describes a set of genes that are expected to co-vary – they will tend to go up and down together. A list of $H = 4$ such vectors is shown below.

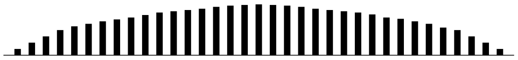
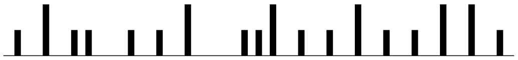
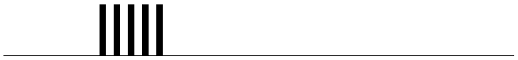
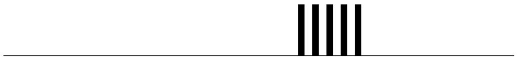
h	b_{hg}, for $g = 1 \ldots G$
$h = 1$	
$h = 2$	
$h = 3$	
$h = 4$	

Figure 2: Differential gene expression signatures.

We call these vectors 'differential gene expression signatures', or just 'signatures' for short.

The first latent variable in this example influences all genes. If the latent variable a_1 is large, all genes are expressed at higher levels – though some more than others. We might call such a latent variable a housekeeping latent variable. Latent variable number $h = 3$ influences just five of the genes, equally. Latent variable number $h = 4$ influences another five. The second latent variable influences about half of the genes.

We assume that the gene expression levels in each tissue t are generated as follows: the latent variables $a_1, a_2, \ldots, a_H$ are set to particular levels $a_{t1}, a_{t2}, \ldots, a_{tH}$, where a_{th} specifies the *strength* of signature h present in tissue t; the gene expression levels d_{tg} are then found by adding up the signatures b_{hg}, weighted by the *strengths* a_{th}. We assume that the data are noisy, with noise n_{tg} in the measurement of d_{tg}. So

$$d_{tg} = \sum_h a_{th} b_{hg} + n_{tg}. \tag{1}$$

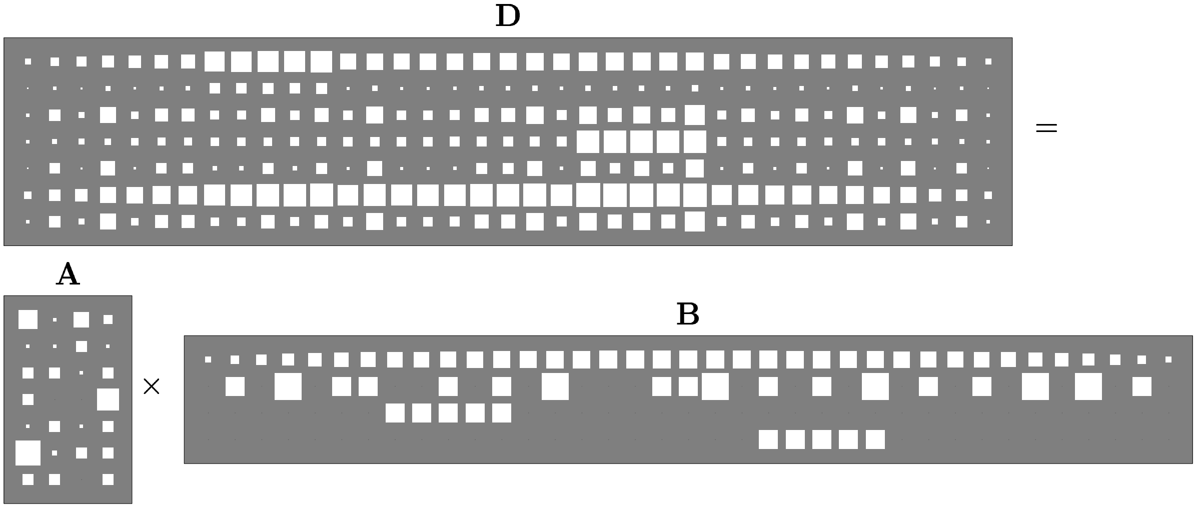
Or, using matrix notation,

$$\mathbf{D} = \mathbf{AB} + \mathbf{N}. \tag{2}$$

This is exactly how the mock data in figure 1 were generated. The signatures $\mathbf{B}$ are those shown in figure 2; the strengths of the signatures are

$$\mathbf{A} = \begin{bmatrix} 3 & 0.1 & 2 & 0.62 \\ 0.1 & 0.1 & 1 & 0.1 \\ 1 & 1 & 0.1 & 1 \\ 1 & 0 & 0 & 4 \\ 0.1 & 1 & 0.1 & 1 \\ 5.1 & 0.2 & 1 & 1 \\ 1 & 1 & 0 & 1 \end{bmatrix}; \tag{3}$$

and there was no noise. Another way of representing the matrices $\mathbf{D}$, $\mathbf{A}$ and $\mathbf{B}$ is by Hinton diagrams in which the magnitude of a number is shown by the size of a square.



Here, all entries in the matrices $\mathbf{A}$ and $\mathbf{B}$ are positive, and all the squares are white; negative entries would be represented by black squares.

If a model with a number of latent variables H significantly smaller than G fits the data well, the model can evidently compress the data substantially, since each extra tissue's G expression levels are captured by just H latent variables.

Now, having described our model of the mechanism underlying the data, we must discuss how we can infer such a latent variable model from data.

2 Inference

If we specify a noise model, $P(n)$, then our assumptions describe the probability of the data given the latent variables $\mathbf{A}$ and the signatures $\mathbf{B}$; for example, if we assume that the noise is Gaussian with variance σ^2, then:

$$P(\mathbf{D}|\mathbf{A},\mathbf{B},\sigma) = \prod_{t,g} \left\{ \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left[-\frac{1}{2\sigma^2} \left(D_{tg} - \sum_h A_{th} B_{hg} \right)^2 \right] \right\}. \tag{4}$$

We now wish to infer $\mathbf{A}$ and $\mathbf{B}$ from $\mathbf{D}$. To do this we require both the likelihood function (4), which describes our assumptions about the noise, and a prior distribution over $\mathbf{A}$ and $\mathbf{B}$, $P(\mathbf{A},\mathbf{B})$, which specifies what sort of values we think the strengths a and the signatures b are likely to take on.

The prior on $\mathbf{B}$ is where we get the opportunity to specify whether the signatures are all-positive (like the toy example), or whether negative values are possible (as would occur if a latent variable corresponded to a factor which promoted transcription of some genes and inhibited transcription of others. The prior on $\mathbf{B}$ can also specify whether we are expecting the signatures to be largely sparse (like signatures 3 and 4 in the toy example). If we believed that each gene is influenced by only *one* latent variable, we would put this property into the prior; the resulting model would then behave very similarly to a 'gene clustering' model, in which each gene is assigned to *one* cluster. The prior on $\mathbf{A}$ specifies the sort of variations in the strengths that we expect to see from tissue to tissue.

We like to use *hierarchical models* to define our priors and likelihoods. The idea is that, while we might believe that the noise is Gaussian, we probably don't know the variance of the noise in advance of the experiment. We thus include the unknown parameter σ^2 in our model as a 'hyperparameter' and define the probability of the noise to be

$$P(\mathbf{N}) = \int P(\mathbf{N}|\sigma^2)P(\sigma^2), \tag{5}$$

where $P(\sigma^2)$ is the prior distribution of the noise variance, which we might represent by a broad distribution (such as a Gamma distribution with large variance).

Similarly, we could slap Gaussian priors on $\mathbf{A}$ and $\mathbf{B}$ with associated hyperparameters. However, this prior would not embody the possibility that the signatures $\mathbf{B}$ might be sparse – a possibility which we would certainly like to include, since we can imagine that some groups of covarying genes might be small in number. We therefore use a *non-Gaussian, heavy-tailed* prior on $\mathbf{B}$. This prior is parameterized by hyperparameters like σ^2 above, which control how sparse $\mathbf{B}$ is, and what the typical magnitude of entries in $\mathbf{B}$ is expected to be.

Similarly, we are open to the possibility that $\mathbf{A}$ might be sparse, so we use a prior on $\mathbf{A}$ which has heavy tails, with a heaviness controlled by hyperparameters. If we choose to force both $\mathbf{A}$ and $\mathbf{B}$ to be non-negative then our model has similarities to that of Lee and Seung (1999).

Finally, we are sceptical about the assertion that the noise $\mathbf{N}$ will be Gaussian, so we actually prefer to use parameterized non-Gaussian noise models which have Gaussian noise as a special case. In this way, we effectively allow the data to tell us what sort of noise model we should be using. We often use a mixture of Gaussians or a mixture of exponentials to produce our non-Gaussian distributions (Pearlmutter and Parra, 1996).

In our use of non-Gaussian priors, our model parts company with factor analysis and is similar to independent component analysis. Unlike Bell and Sejnowski's independent component analysis, we do not constrain the number of latent variables H to be equal to the number of genes, G, however; we hope that H will be found to be smaller than G.

We will use σ^2 as a surrogate name for all the dozen-or-so hyperparameters of the probability distributions for $\mathbf{A}$, $\mathbf{B}$, and $\mathbf{N}$.

2.1 The probability of everything

The probability of everything is

$$P(\mathbf{D},\mathbf{A},\mathbf{B},H,\sigma^2) = P(H)P(\sigma^2)P(\mathbf{A}|\sigma^2)P(\mathbf{B}|\sigma^2)P(\mathbf{D}|\mathbf{A},\mathbf{B},\sigma^2) \tag{6}$$

Given the data, the probability of everything we don't know is

$$\begin{aligned} P(\mathbf{A},\mathbf{B},H,\sigma^2|\mathbf{D}) &= \frac{1}{P(\mathbf{D})}P(\mathbf{D},\mathbf{A},\mathbf{B},H,\sigma^2) & (7)\\ &= \frac{1}{P(\mathbf{D})}P(H)P(\sigma^2)P(\mathbf{A}|\sigma^2)P(\mathbf{B}|\sigma^2)P(\mathbf{D}|\mathbf{A},\mathbf{B},\sigma^2). & (8)\end{aligned}$$

This beast, which describes the inferences we can draw from the data, is what we must now contend with. It is a rather nasty probability distribution because:

1. There are many ways that a matrix $\mathbf{D}$ can be approximately decomposed into a product $\mathbf{AB}$.

2. The model includes hyperparameters, which give rise to technical difficulties. For example settings of the unknown parameters $\mathbf{A},\mathbf{B},H,\sigma^2$ that *maximize* the likelihood function are not necessarily good representatives of where *most* of the posterior probability distribution is to be found. Maxima are misleading in high-dimensional spaces (MacKay, 1999).

2.2 Approximate inference

A popular approach to model-fitting is to maximize the likelihood, or the posterior probability, with respect to the parameters. However, these 'point estimation' approaches ignore *volume information*, and it is possible for the optimized parameters to be unrepresentative of typical plausible parameters. We therefore use an approximation inference method that takes into account volume information. In this *variational approach*, also known as *ensemble learning* (see (MacKay, 1995) for a review), we approximate the exact posterior probability $P(\mathbf{A},\mathbf{B},H,\sigma^2|\mathbf{D})$ (8) by a simpler distribution $Q(\mathbf{A},\mathbf{B},H,\sigma^2)$ and adjust it to make it as close as possible to P.

[The approximating distribution Q represents uncertainty in each parameter but is chosen to be *separable* so it is unable to represent the correlations between the uncertainties of the parameters. Our measure of closeness of Q to P is a variational free energy,

$$\tilde{F}(Q) = \sum_{\mathbf{A},\mathbf{B},H,\sigma^2} Q(\mathbf{A},\mathbf{B},H,\sigma^2)\log\frac{Q(\mathbf{A},\mathbf{B},H,\sigma^2)}{P(\mathbf{D},\mathbf{A},\mathbf{B},H,\sigma^2)}. \tag{9}$$

A by-product of the optimization is that this variational free energy gives a bound on the marginal likelihood, $P(\mathbf{D})$, which is our measure of how well our model models the data. The model could compress the data into $\log 1/P(\mathbf{D})$ bits.

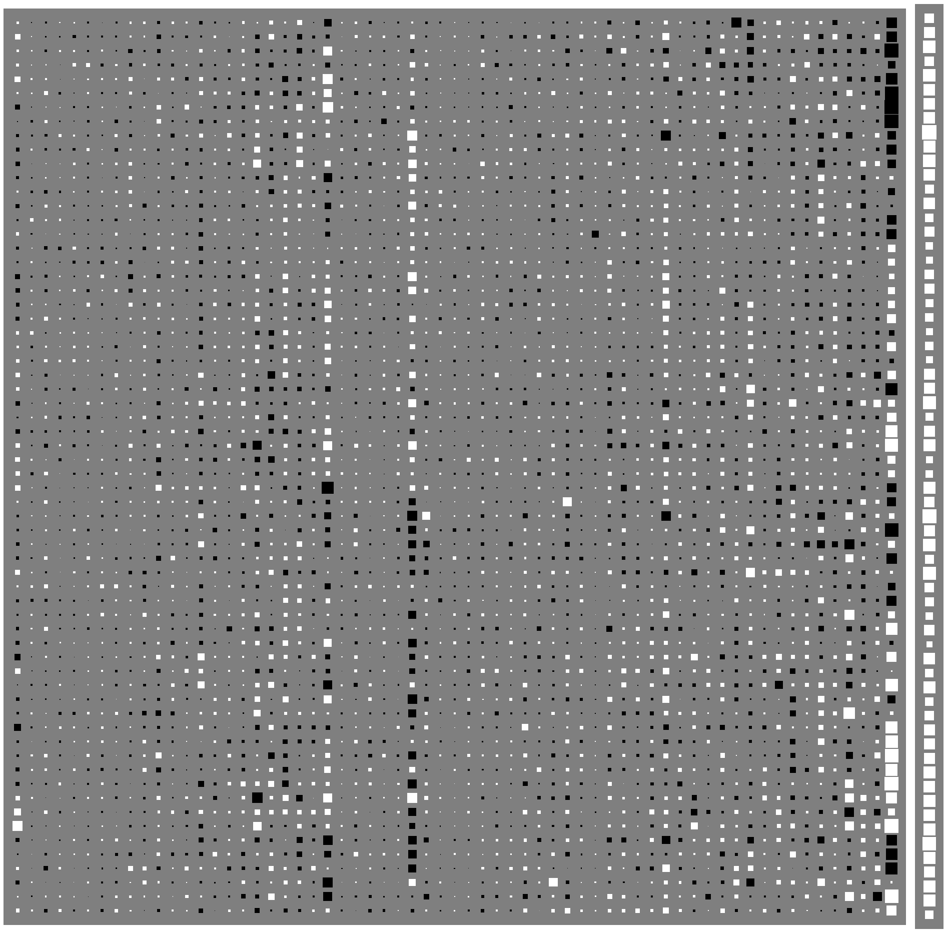


Figure 3: An **A** matrix from a real experiment on the 'Orchid glass' data, provided by Dr Michel Schummer et al., University of Washington, Seattle. The last column shows the strength of the 'ubiquitous' latent variable, which influences all gene expression levels.

2.3 Inferring H

We don't expect that we could ever pin down the number of latent variables, H, for a model like this; given sufficiently large amounts of data, we think it is plausible that the number of independent effects present in the data might be found to be quite large, with many of the effects being small in magnitude.

Rather than directly tackling the question 'what is H', therefore, we use a large value of H and use a prior that expects some of the effects to be small. This can be done either by giving each row h of the matrix $\mathbf{B}$ a hyperparameter that controls the scale of all the parameters in that row; or by having a similar parameter for each column h of $\mathbf{A}$; or both. When we infer the parameters using the variational method, what happens is that latent variables whose roles are not well-determined by the data are effectively switched off; the parameters a_{th} and b_{hg} associated with those latent variables are found to be very close to zero, and have relatively large uncertainty. As an artefact of our approximate method, the system behaves as if it is automatically choosing a value of H.

3 Where we are going

We would like to make a more accurate model of the data-generation process than equation (1).

3.1 Don't normalize, model.

In some microarray experiments, each spot yields two measurements, an experimental measurement m_{tg} and a control measurement l_{tg}, and d_{tg} is defined to be the ratio m_{tg}/l_{tg}. We would prefer to replace all such 'normalizations' of data by inferences of the implicit variables. The motivation for normalization here is that the amount of juice j_{tg} that got into the t, g spot is not known. We assume that the measurement m_{tg} is given by the product of the true gene expression level d_{tg} with the amount of juice j_{tg};

$$m_{tg} \simeq d_{tg} j_{tg} \tag{10}$$

we attempt to control for this unknown variable j_{tg} by measuring the control signal which is assumed to be the product of the same amount of juice j_{tg} with a *constant* control concentration $d^{(0)}$:

$$l_{tg} \simeq d^{(0)} j_{tg}. \tag{11}$$

One reason why it is a bad idea to work with the experimental ratio m_{tg}/l_{tg} is because the noise in the ratio comes from noise in m and from noise in l. Now, those noise variables might be Gaussian – we'd like to know more about that – but the noise in the ratio certainly won't be.

We would therefore prefer to include the unknown juice factors j_{tg} explicitly in the equations. Our generative model would be:

$$\begin{aligned} m_{tg} &= j_{tg}\left(\sum_h a_{th} b_{hg}\right) + n^{(1)}_{tg} && (12) \\ l_{tg} &= j_{tg} d^{(0)} + n^{(0)}_{tg}, && (13) \end{aligned}$$

and we would infer the matrices $\mathbf{A}$, $\mathbf{B}$, and $\mathbf{J}$ from the data matrices $\mathbf{M}$ and $\mathbf{L}$.

3.2 Don't average, model.

In cases where a single gene expression level has been measured multiple times for a single tissue, we would prefer to have access to the raw data rather than to the average. Averaging in general loses information from the data. Only in the special case where the data measurements differ by Gaussian noise is the sample mean a sufficient statistic. In all other cases, we can make better inferences if we work with the full data.

3.3 Big data sets please

Given large data sets we hope to be able to test and validate this model. We hope that we will find robust signatures. We'll only be able to do this with data sets that have many tissues (large T); data sets with enormous G and small T are not so helpful at this stage as it is impossible to validate a latent variable model if the number of independent draws T from the model is small.

[Once the model is validated, it will be especially helpful to experimenters who have data sets with small T. We will be able to find the signatures from other larger data sets, and then, assuming those signatures are applicable, will be able to infer the latent variables characterizing the small data set. The model will thus allow one to control for many natural sources of variability in small experiments.]

References

Bell, A. J., and Sejnowski, T. J. (1995) An information maximization approach to blind separation and blind deconvolution. *Neural Computation* **7** (6): 1129–1159.

Everitt, B. S. (1984) *An Introduction to Latent Variable Models*. London: Chapman and Hall.

Lee, D. D., and Seung, H. S. (1999) Learning the parts of objects by non-negative matrix factorization. *Nature* **401**: 788–791.

MacKay, D. J. C. (1995) Developments in probabilistic modelling with neural networks – ensemble learning. In *Neural Networks: Artificial Intelligence and Industrial Applications. Proceedings of the 3rd Annual Symposium on Neural Networks, Nijmegen, Netherlands, 14-15 September 1995*, pp. 191–198, Berlin. Springer.

MacKay, D. J. C., (1996) Maximum likelihood and covariant algorithms for independent component analysis. `http://wol.ra.phy.cam.ac.uk/mackay/abstracts/ica.html`.

MacKay, D. J. C. (1999) Comparison of approximate methods for handling hyperparameters. *Neural Computation* **11** (5): 1035–1068.

Miskin, J. W. (2001) *Ensemble Learning for Independent Component Analysis*. Department of Physics, University of Cambridge dissertation.

Pearlmutter, B. A., and Parra, L. C., (1996) A context-sensitive generalization of ica. To appear in ICONIP. Also available at `http://www.cnl.salk.edu/~bap/papers/iconip-96-cica.ps.gz`.