

Relevant sparse codes with variational information bottleneck

Authored by:

Matthew Chalk
Olivier Marre
Gasper Tkacik

Abstract

In many applications, it is desirable to extract only the relevant aspects of data. A principled way to do this is the information bottleneck (IB) method, where one seeks a code that maximises information about a relevance variable, Y , while constraining the information encoded about the original data, X . Unfortunately however, the IB method is computationally demanding when data are high-dimensional and/or non-gaussian. Here we propose an approximate variational scheme for maximising a lower bound on the IB objective, analogous to variational EM. Using this method, we derive an IB algorithm to recover features that are both relevant and sparse. Finally, we demonstrate how kernelised versions of the algorithm can be used to address a broad range of problems with non-linear relation between X and Y .

1 Paper Body

An important problem, for both humans and machines, is to extract relevant information from complex data. To do so, one must be able to define which aspects of data are relevant and which should be discarded. The ‘information bottleneck’ (IB) approach, developed by Tishby and colleagues [1], provides a principled way to approach this problem. The idea behind the IB approach is to use additional ‘variables of interest’ to determine which aspects of a signal are relevant. For example, for speech signals, variables of interest could be the words being pronounced, or alternatively, the speaker identity. One then seeks a coding scheme that retains maximal information about these variables of interest, constrained on the information encoded about the input. The IB approach has been used to tackle a wide variety of problems, including filtering, prediction and learning [2-5]. However, it quickly becomes intractable with high-dimensional and/or non-gaussian data. Consequently, previous research has primarily focussed on tractable cases, where the data comprises a countably

small number of discrete states [1-5], or is gaussian [6]. Here, we extend the IB algorithm of Tishby et al. [1] using a variational approximation. The algorithm maximizes a lower bound on the IB objective function, and is closely related to variational EM. Using this approach, we derive an IB algorithm that can be effectively applied to "sparse" data in which input and relevance variables are generated by sparsely occurring latent features. The resulting solutions share many properties with previous sparse coding models, used to model early sensory processing [7]. However, unlike these sparse coding models, the learned representation depends on: (i) the relation between the input and variable of interest; (ii) the trade-off between encoding quality and compression. Finally, we present a kernelized version of the algorithm, that can be applied to a large range of problems with non-linear relation between the input data and variables of interest.

30th Conference on Neural Information Processing Systems (NIPS 2016), Barcelona, Spain.

2

Variational IB

Let us define an input variable X , as well as a "relevance variable", Y , with joint distribution $p(y, x)$. The goal of the IB approach is to compress the variable X through another variable R , while conserving information about Y . Mathematically, we seek an encoding model, $p(r|x)$, that maximizes: $Lp(r|x)$

$$= I(R; Y) - \lambda I(R; X)$$

$$\text{hlog } p(y|r) - \lambda \log p(y) + \lambda \log p(r) - \lambda \log p(r|x) \text{ip}(r,x,y),$$

(1)

where $0 \leq \lambda \leq 1$ is a Lagrange multiplier that determines the strength of the bottleneck. Tishby and colleagues showed that λ be optimized by applying iterative updates: $p^{t+1}(r|x) = p^t(r) \exp(\lambda \log p(y|x) - \log p(y) - \lambda \log p(r|x))$ and $p^{t+1}(y|x) = p(y|x) \exp(\lambda \log p(r|x) - \log p(r) - \lambda \log p(y|x))$ [1]. Unfortunately however, when $p(x, y)$ is high-dimensional and/or non-gaussian these updates become intractable, and approximations are required. Due to the positivity of the KL divergence, we can write, $\text{hlog } q(y|x) - \text{hlog } p(y|x) \geq \text{hlog } p(y|x) - \text{hlog } q(y|x)$ for any approximative distribution $q(y|x)$. This allows us to formulate a variational lower bound for the IB objective function: $\text{hlog } p(r|x), q(y|x), q(r)$ L

=

$$\sum_{n=1}^N \text{hlog } q(y_n|x) + \lambda \log q(r) - \lambda \log p(r|x_n) \text{ip}(r|x_n)$$

?

$$Lp(r|x),$$

(2)

where $q(y_n|x)$ and $q(r)$ are variational distributions, and we have replaced the expectation over $p(x, y)$ with the empirical expectation over training data. (Note that, for notational simplicity we have also omitted the constant term, $H(Y) = -\text{hlog } p(y) \text{ip}(y) = L$), and leads to Setting $q(y_n|x) = p(y_n|x)$ and $q(r) = p(r)$ fully tightens the bound (so that L the iterative algorithm of Tishby et al. However, when these exact updates are not possible, one can instead choose a restricted class of distributions $q(y|x) = Q_{y|x}$ and $q(r)$

For which inference is tractable with respect to parameters of the encoding distribution $p(r|x, \theta)$, tractable. Thus, to maximize L we repeat the following steps until convergence: For fixed θ , find $\{q(y|r), q(r)\} = \arg \max_{q(y|r), q(r)} \{L(q(y|r), q(r), \theta)\}$. For fixed $q(y|r)$ and $q(r)$, find $\theta = \arg \max_{\theta} L$. We note that using a simple approximation for the decoding distribution, $q(y|r)$, can carry additional benefits, besides rendering the IB algorithm tractable. Specifically, while an advantage of mutual information is its generality, in certain cases this can also be a drawback. That is, because Shannon information does not make any assumptions about the code, it is not always apparent how information should be best extracted from the responses: just because information is there does not mean we know how to get at it. In contrast, using a simple approximation for the decoding distribution, $q(y|r)$ (e.g. linear gaussian), constrains the IB algorithm to find solutions where information about Y can be easily extracted from the responses (e.g. via linear regression).

3

Sparse IB

In previous work on gaussian IB [6], responses were equal to a linear projection of the input, plus noise: $r = Wx + \epsilon$, where W is an $N_r \times N_x$ matrix of encoding weights, and $\epsilon \sim N(0, \Sigma)$, where Σ is an $N_r \times N_r$ covariance matrix. When the joint distribution, $p(x, y)$, is gaussian, it follows that the marginal and decoding distributions, $p(r)$ and $p(y|r)$, are also gaussian, and the parameters of the encoding distribution, W and Σ , can be found analytically. To illustrate the capabilities of the variational algorithm, while permitting comparison to gaussian IB, we begin by adding a single degree of complexity. In common with gaussian IB, we consider

a linear gaussian encoder, $p(r|x) = N(r|Wx, \Sigma)$, and decoder, $q(y|r) = N(y|U^T r, \Sigma_y)$. However, unlike gaussian IB, we use the response marginal: $q(r) = \text{student-t}$ distribution to approximate $p(r)$. Student t distribution, $q(r) = \text{Student-t}(r; \mu, \Sigma, \nu)$, with scale and shape parameters, Σ and ν , respectively. When the shape parameter, ν , is small then the student-t distribution is heavy-tailed, or 'sparse', compared to a gaussian distribution. Thus, we call the resulting algorithm 'sparse IB'. Unlike gaussian IB, the introduction of a student-t marginal means the IB algorithm cannot be solved analytically, and one requires approximations.

3.1 Iterative algorithm

Recall that the IB objective function consists of two terms: $I(R; Y)$, and $I(R; X)$. We begin by describing how to optimize the lower and upper bound of each of these two terms with respect to the variational distributions $q(y|r)$ and $q(r)$, respectively. The first term of the IB objective function is bounded from below by:

$$L = \mathbb{E}_{p(x, y)} \log \frac{q(y|x)}{q(y)} - \mathbb{E}_{p(r)} \log \frac{q(r)}{p(r)} + \text{const.}$$

(3)

Maximizing the lower bound on $I(R; Y)$ with respect to the decoding parameters, U and Σ_y , gives: $U = \Sigma_y^{-1} \mathbb{E}_{p(r, y)} [ry^T] \mathbb{E}_{p(r)}^{-1}$, $\Sigma_y = \mathbb{E}_{p(r, y)} [ryr^T] - U \mathbb{E}_{p(r)} U^T$. (4) $\mathbb{E}_{p(r)} [r] = 0$, $\mathbb{E}_{p(r)} [rr^T] = \Sigma_r$ where $\Sigma_r = \mathbb{E}_{p(r)} [rr^T]$, $\Sigma_y = \mathbb{E}_{p(r, y)} [ryr^T] - U \mathbb{E}_{p(r)} U^T$.

y_n , and $C_{xx} = N^{-1} \sum_{n=1}^N x_n x_n^T$. Unfortunately, it is not straightforward to express the bound on $I(R; X)$ in closed form. Instead, we use an additional variational approximation, utilising the fact that can be expressed as the

R student-t distribution pressed as an infinite mixture of gaussians: Student $t(r=0, \Sigma^{-1} = \Sigma^{-1} \Gamma^{-1}, \Gamma^{-1} = \Gamma^{-1})$ [8]. Following a standard EM procedure [9], one can thus write a tractable lower bound on the log likelihood, $\log \text{Student } t(r=0, \Sigma^{-1} = \Sigma^{-1}, \Gamma^{-1} = \Gamma^{-1})$, which corresponds to an upper-bound on the bottleneck term: $\sum_{i,n} I(R; X) \geq \sum_{i,n} \log q(r_i) + \log p(r_i | x_n) p(r_i | x_n)$ (5)

$\# \sum_{i=1}^N \sum_{n=1}^N x_n^T \Sigma^{-1} x_n + f(\Sigma^{-1}, \Sigma^{-1}, a_i) \log \Sigma^{-1} + \text{const.} \sum_{i=1}^N \log \Sigma^{-1} + 2 \sum_{n=1}^N \sum_{i=1}^N \log \Sigma^{-1}$ where Σ^{-1} , and a_i denote for the i th unit and n th data instance. We used the

2 variational parameters Σ^{-1} shorthand notation, $r_{ni} = w_i x_n x_n^T w_i + \Sigma^{-1}$, where Σ^{-1} is the i th diagonal element of Σ^{-1} and w_i is the i th row of W . For notational simplicity, terms that do not depend on the encoding parameters were pushed into the function, $f(\Sigma^{-1}, \Sigma^{-1}, a_i)$. Minimizing the upper bound on $I(R; X)$ with respect to Σ^{-1} , Σ^{-1} and a_i (for fixed Σ^{-1}) gives: $\Sigma^{-1} =$

$$\frac{1}{N} \sum_{i=1}^N \sum_{n=1}^N x_n^T \Sigma^{-1} x_n + \frac{1}{N} \sum_{i=1}^N \log \Sigma^{-1} + \frac{1}{N} \sum_{i=1}^N \log \Sigma^{-1} + \frac{1}{N} \sum_{i=1}^N \log \Sigma^{-1} \quad (6)$$

The shape parameter, Σ^{-1} , is then found numerically on each iteration (for fixed Σ^{-1} and a_i), by solving:

$\sum_{i=1}^N \sum_{n=1}^N x_n^T \Sigma^{-1} x_n + \log \Sigma^{-1} = 1 + \sum_{i=1}^N \log \Sigma^{-1}$, (7) $\sum_{i=1}^N \log \Sigma^{-1}$ where $\log(\Sigma^{-1})$ is the digamma function [9]. Σ^{-1} with respect to the encoding distribution, Next we maximize the full variational objective function L with respect to the encoding noise covariance, Σ^{-1} , $p(r|x)$ (for fixed $q(y|r)$ and $q(r)$). Maximizing L gives: $\sum_{i=1}^N \sum_{n=1}^N x_n^T \Sigma^{-1} x_n = U^T \Sigma^{-1} U + \sum_{i=1}^N \log \Sigma^{-1}$, (8) $\sum_{i=1}^N \log \Sigma^{-1}$

$$f(\Sigma^{-1}, \Sigma^{-1}, a_i) = \log \Sigma^{-1}$$

$$\Sigma^{-1}$$

$$\Sigma^{-1}$$

$$\log$$

$$\Sigma^{-1}$$

$$\Sigma^{-1}$$

$$\Sigma^{-1}$$

$$\Sigma^{-1}$$

$$\Sigma^{-1}$$

$$\Sigma^{-1}$$

$$\Sigma^{-1}$$

$$\Sigma^{-1}$$

$$\Sigma^{-1}$$

$$\Sigma^{-1}$$

$$\Sigma^{-1}$$

$$\Sigma^{-1} + H_{ni}$$
, where H_{ni}

is the entropy of a gamma distribution with shape and rate parameters: a_i , and a_i / Σ^{-1} , respectively [9].

$$\Sigma^{-1}$$

$Y \mid R$
W
B
U
gaussian IB encoding filters
D prob density
X
gaussian IB sparse IB
1 0.1 0.01 0.001 $\cdot 10$
decoding filters
0
10
response (a.u.)
E
 $I(Y; R)$ (nats)
A
100
0 0
50
100
150
 $I(X; R)$ (nats)
sparse IB encoding filters
gaussian IB sparse IB null model
decoding filters
F
1
 $2 \times 2 + 2 \times n$
C
50
0.5
0 0
gaussian IB sparse IB
20
40
units
60
80

Figure 1: Behaviour of sparse IB and gaussian IB algorithms, on denoising task. (A) Artificial image patches were constructed from combinations of orientated edge-like features. Patches were corrupted with white noise to generate the input, X . The goal of the IB algorithm is to learn a linear code that maximized information about the original patches, Y , constrained on information encoded about the input, X . (B) A selection of linear encoding (left), and decoding (right) filters obtained with the gaussian IB algorithm. (C) Same as B, but for the sparse IB algorithm. (D) Response histograms for the 10 units with

highest variance, for the gaussian (red) and sparse (blue) IB algorithms. (E) Information curves for the gaussian (red) and sparse (blue) algorithms, alongside a ?null? model, where responses were equal to the original input, plus white noise. (F) Fraction of response variance attributed to signal fluctuations, for each unit. Solid and dashed curves correspond to strong and weak bottlenecks, respectively (corresponding to the vertical dashed lines in panel E). where Σ and Σ_n are $N_r \times N_r$ diagonal covariance matrices with diagonal elements $\Sigma_{ii} = \Sigma_i^2$, and $(\Sigma_n)_{ii} = \Sigma_{ni}$, respectively. ∂ with respect to the encoding weights, W , gives: Finally, taking the derivative of $L = \frac{1}{2} \sum_i (X_i - U^T C x_i)^2 = \frac{1}{2} \sum_i (X_i - U^T W x_i)^2$

(9)

Setting the derivative to zero, we can solve for W directly. One may verify that, when variational parameters, Σ_{ni} , are unity, the above iterative updates are identical to the iterative gaussian IB algorithm described in [6]. 3.2

Simulations

In our framework, the approximation of the response marginal, $q(r)$, plays an analogous role to the prior distribution in a probabilistic generative model. Thus, we hypothesized that a sparse approximation for the response marginal, $q(r)$, would permit the IB algorithm to recover sparsely occurring input features, analogous to the effect of using a sparse prior. 4

Figure 2: spatially correlated noise B

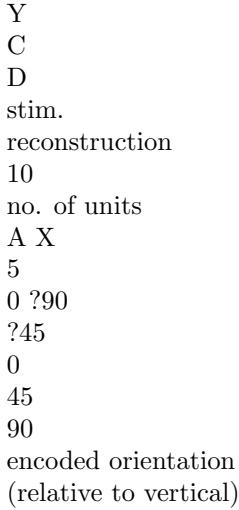


Figure 2: Variant of the task in figure 1, in which the input noise is spatially correlated. (A) Example input X and patch, Y . Spatial noise correlations were aligned along the vertical direction. (B) Subset of decoding filters obtained with the sparse IB algorithm. (C) Distribution of encoded orientations. (D) Example stimulus (left) and reconstruction (right) of bars presented at variable orientations (presented with zero input noise, so that $X = Y$ for this example).

To show this, we constructed artificial 9×9 image patches from combinations of orientated bar features. Each bar had a gaussian cross-section, with maximum amplitude drawn from a standard normal distribution of width 1.2

pixels. Patches were constructed by linearly combining 3 bars, with uniformly random orientation and position. Initially, we considered a simple de-noising task, where the input, X , was a noisy version of the original image patches (gaussian noise, with variance $\sigma^2 = 0.005$; figure 1A). Training data consisted of 10,000 patches. Figure 1B and 1C show a selection of encoding (W) and decoding (U) filters obtained with the gaussian and sparse IB models, respectively. As predicted, only the sparse IB model was able to recover the original bar features. In addition, response histograms were considerably more heavy-tailed for the sparse IB model (fig. 1D). The relevant information, $I(R; Y)$, encoded by the sparse model was greater than for the gaussian model, over a range of bottleneck strengths (fig. 1E). While the difference may appear small, it is consistent with work showing that sparse coding models achieve only a small improvement in log-likelihood for natural image patches [10]. We also plotted the information curve for a "null model", with responses sampled from $p(r|x) = N(r|x, \sigma^2 I)$. Interestingly, the performance of this null model was almost identical to the gaussian IB model. wC

wT

Figure 1F plots the fraction of response variance due to the signal, for each unit ($w_i C_{xxxx}$). Solid $wT + \sigma^2$ and dashed curves denote strong and weak bottlenecks, respectively. In both cases, the gaussian model gave a smooth spectrum of response magnitudes, while the sparse model was more "all-or-nothing".

One way the sparse IB algorithm differs qualitatively from traditional sparse coding algorithms, is that the learned representation depends on the relation between X and Y , rather than just the input statistics. To illustrate this, we conducted simulations with patches corrupted by spatially correlated noise, aligned along the vertical direction (fig. 2A). The spatial covariance of the noise was described by a gaussian envelope, with standard deviation 3 pixels in the vertical direction and 1 pixel in horizontal direction. Figure 2B shows a selection of decoding filters obtained from the sparse IB model, with correlated input noise. The shape of individual filters was qualitatively similar to those obtained with uncorrelated noise (fig. 1C). However, with this stimulus, the IB model avoided "wasting" bits by representing features co-orientated with the noise (fig. 2C). Consequently, it was not possible to reconstruct vertical bars from the responses, when bars were presented alone, even with zero noise (fig. 2D).

4

Kernel IB

One way to improve the IB algorithm is to consider non-linear encoders. A general choice is: $p(r|x) = N(r|W\phi(x), \sigma^2)$, where $\phi(x)$ is an embedding to a high-dimensional non-linear feature space. 5

D

$f(X)$

R

U

sparse kIB

B
 E 1 gaussian kIB sparse kIB
 20
 prob density
 Y?
 X
 llin (Y ; R) (nats)
 15 10
 gaussian kIB sparse kIB sparse IB
 5 0 0
 20
 0.1 0.01 0.001
 0.0001
 40
 ?10
 I(X; R) (nats)
 F
 C gaussian kIB
 sparse IB
 gaussian IB
 stim 1
 0
 10
 response (a.u.)
 stim 2
 stim 3
 recon. patches
 A
 G
 ?5
 st im st 1 im st 2 im 3
 0
 st im st 1 im st 2 im 3
 response (a.u.)
 5

Figure 3: Behaviour of kernel IB algorithm on occlusion task. (A) Image patches were the same as for figure 1. However, the input, X , was restricted to 2 columns to either side of the patch. The target variable, Y , was the central region. (B) Subset of decoding filters, U , for the sparse kernel IB (?sparse kIB?) algorithm. (C) As for B, for other versions of the IB algorithm. (D) Information curves for the gaussian kIB (blue) sparse kIB (green) and sparse IB algorithms (red). The bottleneck strength for the other panels in this figure is indicated by a vertical dashed line. (E) Response histogram for the 10 units with highest variance, for the gaussian and sparse kIB models. (F) (above) Three test stimuli, used to demonstrate the non-linear properties of the sparse KIB code. (below) Reconstruction obtained from responses to test stimulus.

side. (B) Response distribution for 10 neurons with highest variance, for the gaussian (blue) and sparse (green) kIB algorithms. (C) Decoding filters for a subset of units, obtained with the sparse kIB algorithm. Note that, for clearer visualization, we show here the decoding filter for the entire image patch, not just the occluded region. (D) A selection of decoding filters obtained with the alternative IB algorithms. method). In addition, the computational complexity can be reduced by restricting the solution to lie PM on a subspace of training instances, such that, $w_i = \sum_{n=1}^N a_{in} \phi(x_n)$, where $M \leq N$. The derivation does not change, only now K has dimensions $M \times N$ [11]. When $q(r)$ is gaussian (equivalent to setting $\sigma_n = I$), solving for A gives: $\frac{1}{T} \sum_{t=1}^T A = U^T \frac{1}{T} \sum_{t=1}^T U + \frac{1}{T} \sum_{t=1}^T U^T A KRR$

(11)

where $AKRR = Y(K + \lambda I)$ are the coefficients obtained from kernel ridge-regression (KRR). This suggests the following two stage algorithm: first, we learn the regularisation constant, λ , and parameters of the kernel matrix, K , to maximize KRR performance on hold-out data; next, we perform variational IB, with fixed K and λ . 4.1

Simulations

To illustrate the capabilities of the kernel IB algorithm, we considered an occlusion task, with the outer columns of each patch presented as input, X (2 columns to the far left and right), and the inner columns as the relevance variable Y , to be reconstructed. Image patches were as before. Note that performing the occlusion task optimally requires detecting combinations of features presented to either side of the occluded region, and is thus inherently nonlinear. We used gaussian kernels, with scale parameter, σ , and regularisation constant, λ , chosen to maximize KRR performance on test data. Both test and training data consisted of 10,000 images. However, A was restricted to lie on a subset of 1000 randomly chosen training patches (see earlier). Figure 3B shows a selection of decoding filters (U) learned by the sparse kernel IB algorithm (sparse kIB). A large fraction of filters resembled near-horizontal bars, traversing the occluded region. This was not the case for the sparse linear IB algorithm, which recovered localized blobs either side of the occluded region, nor the gaussian linear or kernelized models, which recovered non-local features (fig. 3C). Figure 3D shows a small but significant improvement in performance for the sparse kIB versus the gaussian kIB model. Most noticeable, however, is the distribution of responses, which are much more heavy tailed for the sparse kIB algorithm (fig. 3E). To demonstrate the non-linear behaviour of the sparse kIB model, we presented bar segments: first to either side of the occluded patch, then to both sides simultaneously. When bar segments were presented to both sides simultaneously, the sparse KIB model filled in the missing bar segment, 7

in contrast to the reconstruction obtained with single bar segments (fig. 3F). This behaviour was reflected in the non-linear responses of certain encoding units, which were large when two segments were presented together, but near zero when one segment was presented alone (fig. 3G). Finally, we repeated the occlusion task with handwritten digits, taken from the USPS dataset (www.gaussianprocess.org/gpml/data). We used 4649 training and 4649 test patches,

of 16×16 pixels. However, expansion coefficients were restricted to lie on a subset of 500 randomly patches. We set X and Y , to be the left and right side of each patch, respectively (fig. 4A). In common with the artificial data, the response distributions achieved with the sparse kIB algorithm were more heavy-tailed than for the gaussian kIB algorithm (fig. 4B). Likewise, recovered decoding filters closely resembled handwritten digits, and extended far into the occluded region (fig. 4C). This was not the case for the alternative IB algorithms (fig. 4D).

5

Discussion

Previous work has shown close parallels between the IB framework and maximum-likelihood estimation in a latent variable model [12, 13]. For the sparse IB algorithm presented here, maximizing the IB objective function is closely related to maximizing the likelihood of a ‘sparse coding’ latent variable model, with student-t prior and linear gaussian likelihood function. However, unlike traditional sparse coding models, the encoding (or ‘recognition’) model $p(r|x)$ is conditioned on a separate set of inputs, X , distinct from the image patches themselves. Thus, the solutions depend on the relation between X and Y , not just the image statistics (e.g. see fig. 2). Second, an additional parameter, λ , not present in sparse coding models, controls the trade-off between encoding and compression. Finally, in contrast to traditional sparse coding algorithms, IB gives an unambiguous ordering of features, which can be arranged according to the response variance of each unit (fig. 1F). Our work is also closely related to the IM algorithm, proposed by Barber et al. to solve the information maximization (‘infomax’) problem [14]. However, a general issue with infomax problems is that they are usually ill-posed, necessitating additional ad hoc constraints on the encoding weights or responses [15]. In contrast, in the IB approach, such constraints emerge automatically from the bottleneck term. A related method to find low-dimensional projections of X/Y pairs is canonical correlation analysis (‘CCA’), and its kernel analogue [16]. In fact, the features obtained with gaussian IB are identical to those obtained with CCA [6]. However, unlike CCA, the number and ‘scale’ of the features are not specified in advance, but determined by the bottleneck parameter, λ . Secondly, kernel CCA is symmetric in X and Y , and thus performs nonlinear embedding of both X and Y . In contrast, the IB problem is asymmetric: we are interested in recovering Y from an input X . Thus, only X is kernelized, while the decoder remains linear. Finally, the features obtained from gaussian IB (and thus, CCA) differ qualitatively from the sparse IB algorithm, which recovers sparse features that account jointly for X and Y . Sparse IB can be extended to the nonlinear regime using a kernel expansion. For the gaussian model, the expansion coefficients, A , are a linear projection of the coefficients used for kernel-ridge regression (‘KRR’). A general disadvantage of KRR, is that it can be difficult to know which aspects of X are relied on to perform the regression. In contrast, the kernel IB framework provides an intermediate representation, allowing one to visualize the features that jointly account for both X and Y (figs. 3B & 4C). Furthermore, this learned representation permits generalisation across different tasks that rely on the same set

of latent features; something not possible with KRR. Finally, the IB approach has important implications for models of early sensory processing [17, 18]. Notably, ‘efficient coding’ models typically consider the low-noise limit, where the goal is to reduce the neural response redundancy [7]. In contrast, the IB approach provides a natural way to explore the family of solutions that emerge as one varies internal coding constraints (by varying β) and external constraints (by varying the input, X) [19, 20]. Further, our simulations suggest how the framework can be used to go beyond early sensory processing: for example to explain higher-level cognitive phenomena such as perceptual filling in (fig. 3G). In future, it would be interesting to explore how the IB framework can be used to extend the efficient coding theory, by accounting for modulations in sensory processing that occur due to changing task demands (i.e. via changes to the relevance variable, Y), rather than just the input statistics (X). 8

2 References

- [1] Tishby, N. Pereira, F. C. & Bialek, W. (1999) The information bottleneck method. The 37th annual Allerton Conference on Communication, Control and Computing. pp. 368-377
- [2] Bialek, W. Nemenman, I. & Tishby, N. (2001) Predictability, complexity, and learning. *Neural computation*, 13(11) pp. 240- 63
- [3] Slonim, N. (2003) Information bottleneck theory and applications. PhD thesis, Hebrew University of Jerusalem
- [4] Chechik, G. & Tishby, N. (2002) Extracting relevant structures with side information. In *Advances in Neural Information Processing Systems 15*
- [5] Hofmann, T. & Gondek, D. (2003) Conditional information bottleneck clustering. In *3rd IEEE International conference in data mining, workshop on clustering large data sets*
- [6] Chechik, G. Globerson, A., Tishby, N. & Weiss, Y. (2005) Information bottleneck for gaussian variables. *Journal of Machine Learning Research*, (6) pp. 165-188
- [7] Simoncelli, E. P. & Olshausen, B. A. (2001) Natural image statistics and neural representation. *Ann. Rev. Neurosci.* 24:1194-216
- [8] Andrews, D. F. & Mallows C. L. (1974). Scale mixtures of normal distributions. *J. of the Royal Stat. Society. Series B* 36(1) pp. 99-102
- [9] Scheffler, C. (2008). A derivation of the EM updates for finding the maximum likelihood parameter estimates of the student-t distribution. Technical note. URL www.inference.phy.cam.ac.uk/cs482/publications/scheffler2008derivation.pdf
- [10] Eichhorn, J. Sinz, F. & Bethge, M. (2009). Natural image coding in V1: how much use is orientation selectivity?. *PLoS Comput Biol*, 5(4), e1000336.
- [11] Mika, S. Ratsch, G. Weston, J. Scholkopf, B. Smola, A. J. & Muller, K. R. (1999). Invariant Feature Extraction and Classification in Kernel Spaces. In *Advances in neural information processing systems 12* pp. 526-532.
- [12] Slonim, N., & Weiss, Y. (2002). Maximum likelihood and the information bottleneck. In *Advances in neural information processing systems* pp. 335-342
- [13] Elidan, G., & Friedman, N. (2002). The information bottleneck EM algorithm. In *Proceedings of the Nineteenth conference on Uncertainty in Artificial Intelligence* pp. 200-208
- [14] Barber, D. & Agakov, F. (2004) The IM algorithm: a variational approach to information maximization. In *Advances in*

Neural Information Processing Systems 16 pp. 201-208 [15] Doi, E., Gauthier, J. L. Field, G. D. Shlens, J. Sher, A. Greschner, M. (2012). Efficient Coding of Spatial Information in the Primate Retina. *The Journal of neuroscience* 32(46), pp. 16256-16264 [16] Hardoon, D. R., Szedmak, S. & Shawe-Taylor, J. (2004). Canonical correlation analysis: An overview with application to learning methods. *Neural computation*, 16(12), 2639-2664. [17] Bialek, W., de Ruyter Van Steveninck, R. R., & Tishby, N. (2008). Efficient representation as a design principle for neural coding and computation. In *Information Theory, 2006 IEEE International Symposium* pp. 659-663 [18] Palmer, S. E., Marre, O., Berry, M. J., & Bialek, W. (2015). Predictive information in a sensory population. *Proceedings of the National Academy of Sciences* 112(22) pp. 6908-6913. [19] Doi, Eizaburo. & Lewicki, M. S. (2005). Sparse coding of natural images using an overcomplete set of limited capacity units. In *Advances in Neural Information Processing Systems 17* pp. 377-384 [20] Tkacik, G. Prentice, J. S. Balasubramanian, V. & Schneidman, E. (2010). Optimal population coding by noisy spiking neurons. *Proceedings of the National Academy of Sciences* 107(32), pp. 14419-14424.