BDL.NET: BAYESIAN DICTIONARY LEARNING IN INFER.NET

Tom Diethe, Niall Twomey, Peter Flach

Intelligent Systems Laboratory, University of Bristol, UK

ABSTRACT

We introduce and analyse a flexible and efficient implementation of Bayesian dictionary learning for sparse coding. By placing Gaussian-inverse-Gamma hierarchical priors on the coefficients, the model can automatically determine the required sparsity level for good reconstructions, whilst also automatically learning the noise level in the data, obviating the need for heuristic methods for choosing sparsity levels. This model can be solved efficiently using Variational Message Passing (VMP), which we have implemented in the Infer.NET framework for probabilistic programming and inference. We analyse the properties of the model via empirical validation on several accelerometer datasets. We provide source code to replicate all of the experiments in this paper.

***Index Terms*—** Sparse Coding, Dictionary Learning, Bayesian, Accelerometers

1. INTRODUCTION

Our motivating application is Activity Recognition (AR), which is usually performed for the purposes of understanding the Activities of Daily Living (ADL) of a given individual. One of the most popular methods for the study of ADL is through the use of a wearable device containing an accelerometer, often combined with gyroscopes, which measure the degree of rotation as the device rotates in any its axes. Since gyroscopes consume several orders of magnitude more power than low power accelerometers, we are most interested in accelerometers only.

Traditional methods for classification of accelerometer signals involve computing features in both the temporal and frequency domains over a temporal window - see *e.g.* [1]. One effect of this is to reduce the temporal dependence of neighbouring examples, which enables the use of standard classification algorithms. There is a trade-off here: longer windows mean less dependence and less computational burden; however in extremis, the probability that a given window involves only a single activity class diminishes. We are therefore interested in a compact representation of the signals, that also contains the necessary information to be discriminatory between activity classes.

1.1. Dictionary Learning

Dictionary Learning, also known as Sparse Coding [2] is a class of unsupervised methods for learning sets of over-complete bases to represent data in a parsimonious manner. The aim of sparse coding is to find a set of vectors $\mathbf{d}_i$, known as a dictionary, such that we can represent an input vector $\mathbf{x} \in \mathbb{R}^n$ as a linear combination of these vectors:

$$\mathbf{x} = \sum_{i=1}^{k} \mathbf{z}_i \mathbf{d}_i \qquad \text{s.t.} \quad k \gg n. \tag{1}$$

While there exist efficient techniques to learn a complete set of vectors (*i.e.* a basis) such as Principal Component Analysis (PCA)[3], an over-completeness can achieve a more stable, robust, and compact decomposition than using a basis [4]. However, with an over-complete basis, the coefficients z_i are no longer uniquely determined by the input vector $\mathbf{x}$. Therefore, in sparse coding, we introduce additional sparsity constraints to resolve the degeneracy introduced by over-completeness.

Sparsity is defined as having few non-zero components z_i or many that are close to zero. The sparse coding cost function on a set of m input vectors arranged in the columns of the matrix $\mathbf{X} \in \mathbb{R}^{n \times m}$ as

$$\begin{aligned} &\min_{\mathbf{Z},\mathbf{D}} \|\mathbf{X} - \mathbf{D}\mathbf{Z}\|_F^2 + \lambda \sum_{i=1}^{n} \Omega(\mathbf{z}_i) \\ &\text{s.t.}\ \|\mathbf{d}_i\|^2 \leq C, \quad \forall i = 1, \dots, k. \end{aligned}$$

where $\mathbf{D} \in \mathbb{R}^{n \times k}$ is the set of basis vectors (dictionary), $\mathbf{Z} \in \mathbb{R}^{k \times n}$ is the set of coefficients for each example, and $\Omega(.)$ is a sparsity inducing regularisation function, and the scaling constant λ determines the relative importance of good reconstructions and sparsity. The most direct measure of sparsity is the L_0 quasi-norm $\Omega(z_i) = \mathbf{1}(|z_i| > 0)$, but it is non-differentiable and difficult to optimise in general. A common choice for the sparsity cost $\Omega(.)$ is the L_1 penalty $\Omega(z_i) = \sum_{i=1}^{n} |z_i|$ (see [5] for a review). Since it is also possible to make the sparsity penalty arbitrarily small by scaling down z_i

The SPHERE Interdisciplinary Research Collaboration is funded by the UK Engineering and Physical Sciences Research Council (EPSRC) under Grant EP/K031910/1.

978-1-5090-0746-2/16/$31.00 ©2016 Crown

and scaling $\mathbf{d}_i$ up by some large constant, $\|\mathbf{d}\|^2$ is constrained to be less than some constant C.

Since the optimisation problem is not jointly convex in $\mathbf{Z}$ and $\mathbf{D}$, sparse coding consists of performing two separate optimisations: (1) over coefficients $\mathbf{z}_i$ for each training example $\mathbf{x}_i$ with $\mathbf{D}$ fixed; and (2) over basis vectors $\mathbf{D}$ across the whole training set with $\mathbf{Z}$ fixed. Using an L_1 sparsity penalty, sub-problem (1) reduces to solving an L_1 regularised least squares problem which is convex in $\mathbf{z}_i$ which can be solved using standard convex optimisation software such as CVX [6]. With a differentiable $\Omega(\cdot)$ such as the log penalty, conjugate gradient methods can also be used. Sub-problem (2) reduces to a least squares problem with quadratic constraints which is convex in $\mathbf{d}$, for which again there are standard methods available.

1.2. Online Dictionary Learning

As described thus far, dictionary learning algorithms require the entire set of training signals $\mathbf{X}$, which puts a limitation on the sizes of problems that can be tackled, and means that they cannot operate in an online streaming scenario. An online version of dictionary learning was introduced by [7], which involved iteratively finding the sparse representation for each data point $\mathbf{x}_i$ as it arrives, and then updating $\mathbf{D}$ using a block-coordinate approach.

1.3. Our Contributions

• We improve on existing methods for Bayesian Dictionary Learning (BDL), with a more stable model • We give an efficient implementation using deterministic approximations • We show that priors that do not enforce sparsity can still result in sparse representations, whilst giving better reconstructions • We demonstrate how such models can be applied to accelerometer signals

2. RELATED WORK

A Bayesian approach to the dictionary learning problem is highly appealing for several reasons. Firstly, it is possible to learn the noise level directly from the data, rather than having to specify it or estimate using crude heuristics. Secondly, it allows us to consider building larger models, such as hierarchical models that enable us to reason about the differences between individuals and groups of people, and also to consider transfer learning.

A hierarchical Bayesian model for dictionary learning was first introduced by [8], in which a Gaussian-inverse Gamma hierarchical prior was used to promote the sparsity of the representation. The authors argued that better learning was achieved compared to baselines in the case where there is a limited number of training signals. We will discuss the relation to our work in section 3.

An appealing nonparametric Bayesian approach to the problem was introduced by [9], which allows an adapted dictionary size using an Indian Buffet Process prior. Currently, however, there are no efficient methods for inference in this class of models, which somewhat limits their use.

In terms of the application area of interest here, feature learning was applied to AR from accelerometer data by [10], where the authors investigated amongst other things PCA and auto-encoder networks. Dictionary learning would be a natural extension here. Following on from this, a form of shift invariant sparse coding was proposed for the same task by [11]. The authors use an approach that can be seen as a form of convolutional sparse coding, with promising classification performance.

3. METHODS

We first give a generative model for eq. (1), in which we posit that our signals are generated by the same linear combination of bases, and give parametric forms for the (latent) variables and include a noise model.

$$\begin{aligned}
\mathbf{X} &= \mathbf{D}\mathbf{Z} + \mathbf{N}, \\
p(\mathbf{D}) &= \prod_{i=1}^{n}\prod_{j=1}^{k} \mathcal{N}(\mathbf{d}_{i,j}; \alpha_{i,j}, \beta^{-1}), \\
p(\boldsymbol{\alpha}) &= \prod_{i=1}^{n}\prod_{j=1}^{k} \mathcal{N}(\alpha_{i,j}; 0, 1), \\
p(\boldsymbol{\beta}) &= \prod_{i=1}^{n}\prod_{j=1}^{k} \mathcal{G}a(\beta_{i,j}; 1, 1), \\
p(\mathbf{Z}|\boldsymbol{\tau}) &= \prod_{i=1}^{k}\prod_{j=1}^{m} \mathcal{N}(z_{i,j}; 0, \tau_{i,j}^{-1}), \\
p(\boldsymbol{\tau}) &= \prod_{i=1}^{k}\prod_{j=1}^{m} \mathcal{G}a(\tau_{i,j}; a, b), \\
p(\mathbf{N}) &= \prod_{i=1}^{n}\prod_{j=1}^{m} \mathcal{N}(n; 0, \lambda) \\
p(\lambda) &= \mathcal{G}a(\lambda; a, b),
\end{aligned} \tag{2}$$

where $\mathcal{N}$ is the Gaussian distribution for a given mean and variance and $\mathcal{G}a$ is the Gamma distribution for a given shape and rate.

This model builds on that of [8], and is shown in fig. 1. There are several key differences. Firstly, rather than using a fixed value for β, which defines the precision of the dictionary atoms, we instead put a Gamma prior over β, which allows the dictionary atoms to be automatically scaled. In their experiments, Yang *et. al.* used a value of $\beta = 1$ when using Gibbs sampling, and a value of $\beta = 10^{-8}$ when using Variational Inference. It is not clear why they had to make this decision,

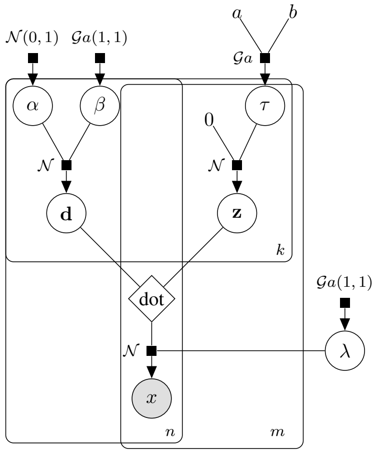


Fig. 1: Factor graph representing a hierarchical Bayesian model for dictionary learning.

but our experiments show that the additional Gamma prior appears to obviate the need to do this. Furthermore, we place an additional level of hierarchy through the variables $\boldsymbol{\alpha}$ on the means of the dictionary components, which can aid online learning.

3.1. Inference

In this work, we employ Variational Message Passing (VMP), which is an efficient deterministic approximation algorithm for applying variational inference to Bayesian graphical models [12]. Like Belief Propagation (BP) and Expectation Propagation (EP) [13], VMP proceeds by sending messages between nodes in the network and updating posterior beliefs using local operations at each node. Each such update increases a lower bound on the log evidence at that node, (unless already at a local maximum). VMP can be applied to a very general class of conjugate-exponential models because it uses a factorised variational approximation, and by introducing additional variational parameters, VMP can be applied to models containing non-conjugate distributions. The VMP framework also allows the lower bound to be evaluated, which can be used both for model comparison and for detection of convergence.

To break symmetry, we randomly initialise the dictionary elements to independently and identically distributed (IID) draws from a standard normal distribution. For consistency, we use the same random seed for all experiments.

4. EXPERIMENTS

4.1. Data Sets

HAD dataset [1]

This involved 30 participants aged 19-48 years and six activities were recorded. Each participant wore a smart-phone on the waist, with tri-axial linear acceleration and tri-axial angular velocity capture using its embedded accelerometer and gyroscope at a constant rate of 50 Hz. Annotation was done using video-recordings. Each sequence is on average 7 minutes long. The activities performed were: 1. Walking 2. Ascending stairs 3. Descending stairs 4. Sitting 5. Standing 6. Lying down.

SPHERE challenge dataset [14]

This dataset has been collected by our research group in our smart home deployment and made public as a challenge[1]. The task is prediction of posture and ambulation of participants who wore a tri-axial accelerometer on the dominant wrist. The accelerometers record data at 20 Hz, with a range of ± 8 g. Here we examine a subset of the labels available in the dataset that are also found in the HAD dataset, *i.e.* : 1. Lie 2. Stand 3. Walk

4.2. Data pre-processing

The signal streams were split into windows of 3 seconds in length, from which we computed the magnitude of the acceleration vector and subtracted 1 (gravitational force). The windowed signals were then normalised to have unit ℓ_2-norm

In our comparisons with non-Bayesian sparse coding, we used the SPArse Modeling Software (SPAMS) toolbox [2], and in particular we used the online dictionary learning method described in [7].

We follow [7] and used the regularisation parameter $\lambda = 1.2/\sqrt{m}$ in all of our experiments (≈ 0.1 for HAD and ≈ 0.03 for SPHERE). The $1/\sqrt{m}$ term is a classical normalisation factor, and the constant 1.2 was shown to yield about 10 nonzero coefficients in their experiments.

The methods described here were all implemented using Infer.NET [15], which is a framework for running Bayesian inference in graphical models, and provides a rich modelling language for a wide variety of applications. In our experiments we compile and run the code using Mono[3], an open source implementation of Microsoft's .NET, running on OS-X and Linux.

4.3. Reconstruction

In order to test reconstruction, in all cases we take the dictionary learnt on the training set (2D Gaussian arrays in the Bayesian methods), and first compute coefficients for the test signals using this dictionary. We then reconstruct the signals using the trained dictionary and inferred coefficients.

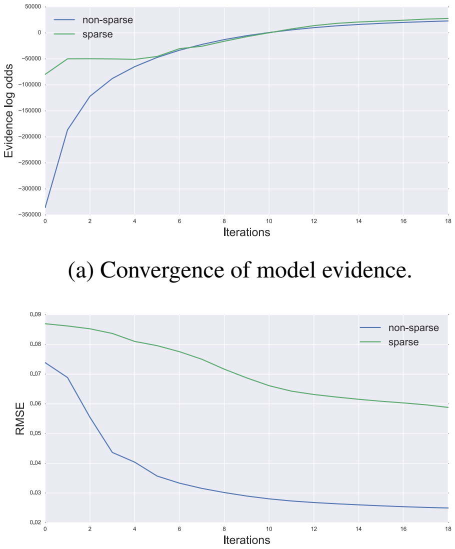
For the performance metric for the quality of reconstructions we will adopt the commonly used root-mean-square error (RMSE) $= \sqrt{\sum_{t=1}^{n}(\hat{x}_t - x_t)^2/n}$.

[1] http://irc-sphere.ac.uk/sphere-challenge/home
[2] http://spams-devel.gforge.inria.fr/
[3] http://www.mono-project.com/

5. RESULTS AND DISCUSSION

5.1. Convergence of VMP

Before continuing we will first analyse the convergence properties of VMP for the models defined herein. VMP is a deterministic approximation algorithm, and for well-behaved problems the model evidence (or marginal likelihood) will always increase and will converge to a local maximum. We follow [12], and define convergence by evaluating the model evidence in the variational posterior after each full round of message passing, checking that the value of the bound does not decrease by more than some tolerance. We will refer to this as “iterative model comparison” (IMC) henceforth.



(a) Convergence of model evidence.

(b) Convergence of hold-out reconstruction error.

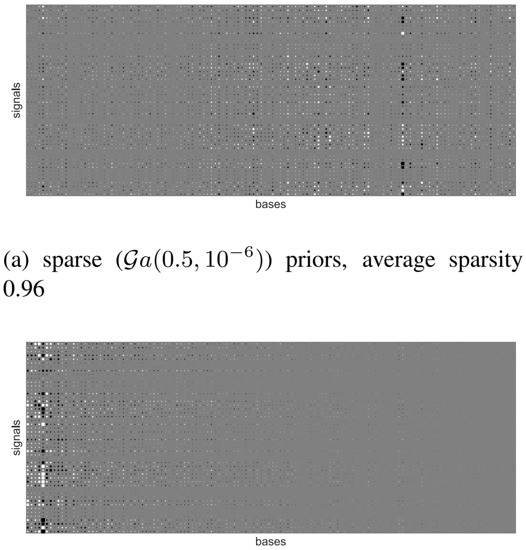
Fig. 2: Convergence plots of marginal likelihood and reconstruction error.

In figs. 2a and 2b we have plotted the convergence of the model evidence on a subset of the HAD dataset and the reconstruction error on a hold-out test set respectively. Here we used 1000 instances for training, and computed the RMSE on 200 instances from the test set. In order to do so, we had to use a convergence criterion for the inference of test set coefficients and reconstructed signals, for which we used IMC with a tolerance of 10^{-4}. We let the message passing run for 20 iterations even if it would have passed the IMC criterion. It is interesting to note that while the evidence for each of the models follows similar convergence paths, the reconstruction errors are clearly in favour of the non-sparse model.

For all further experiments we used the IMC method with a tolerance of 10^{-3} for dictionary learning, coefficient estimation, and reconstruction.

5.2. Sparsity

To compute sparsity we compute the norm of the means each of the coefficient posterior vectors, and then threshold at a value of 10^{-4} relative to this norm. We show a Hinton diagrams for the coefficients learnt for 50 example signals from the HAD dataset, using the base model (*i.e.* no norm constraints), with sparse ($\mathcal{G}a(0.5, 10^{-6})$) and non-sparse ($\mathcal{G}a(1, 1)$) priors in fig. 3a and fig. 3b respectively. Note that in the case of non-sparse priors, the resultant coefficients are still very sparse.



(a) sparse ($\mathcal{G}a(0.5, 10^{-6})$) priors, average sparsity 0.96

(b) non-sparse ($\mathcal{G}a(1, 1)$) priors average sparsity 0.84

Fig. 3: Hinton diagram for the base model with (a) and without (b) sparse priors, whose rows give the expected 128-dimensional mean of the coefficients of a sample of 50 signals.

5.3. Learnt dictionaries and bases

In fig. 4 we can see some randomly selected elements from the dictionary created using 128 bases from the HAD dataset, and in fig. 5 we can see the coefficients for an example signal chosen from each of the 6 activity classes. It is clear that the more passive activities (sitting, standing, lying) are represented by fewer active bases. Furthermore, it would appear that these coefficients have the potential to be discriminative of the activity being performed.

In fig. 6 we can see the reconstructions of two example signals from the accelerometer dataset. The shaded regions show the standard deviation (SD) of the Gaussian marginals. It is worth noting that reconstruction uncertainty is not available from methods such as SPAMS.

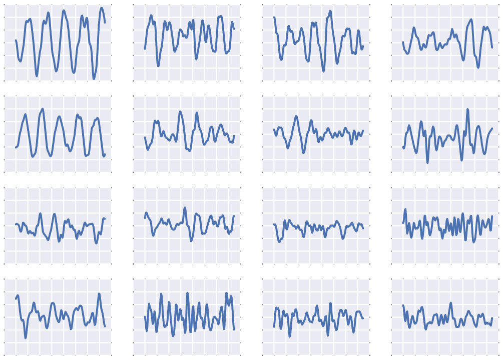


Fig. 4: 16 example bases from the dictionary of 128 bases inferred by BDL using the non-sparse priors.

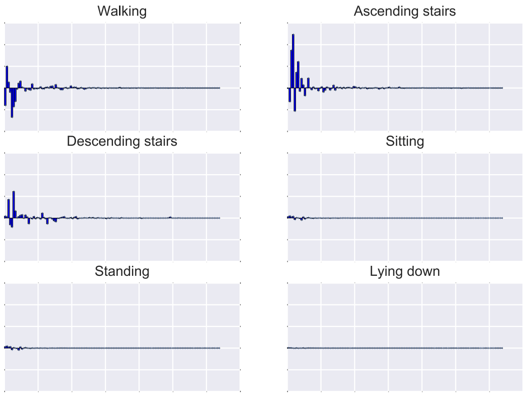


Fig. 5: Coefficients from accelerometer signals from each of the 6 activity classes using 128 bases inferred by BDL using the non-sparse priors.

In table 1 we can see a comparison of average reconstruction error and sparsity of BDL with SPAMS on dataset 1. In almost all cases, the reconstruction produced by BDL are superior to those by SPAMS, whilst achieving comparable sparsity, even when non-sparse priors are used. The sparsity enforcing priors do indeed result in sparser solutions, but at the cost of reconstruction error, up until we use 512 bases, at which the over-completeness justifies the use of sparsity inducing priors.

Table 1: Average test set reconstruction error and sparsity on dataset 1.

	SPAMS		BDL Sparse		BDL	
Bases	RMSE	Sparsity	RMSE	Sparsity	RMSE	Sparsity
64	0.0480	0.71	0.0519	0.88	0.0293	0.62
128	0.0457	0.84	0.0400	0.94	0.0276	0.84
256	0.0438	0.92	0.0316	0.99	0.0288	0.93
512	0.0423	0.96	0.0224	0.99	0.0231	0.96

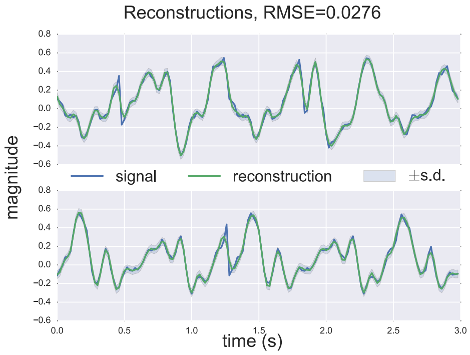


Fig. 6: Reconstructions of two example accelerometer signals using 128 bases inferred by BDL using the non-sparse priors. The original signal is shown in blue, with the reconstructions shown in blue with ± one standard deviation shown as a shaded region.

5.4. Activity Recognition

Here we present results on the use of the computed coefficients as features in a classification algorithm for the purposes of AR on the HAD and SPHERE datasets. Do to space limitations, the results presented here should be regarded as a proof of concept. As such, we will not present extensive comparisons with other feature generation methods or classification models.

We employ the multi-class Bayes Point Machine (BPM) [16], which is a linear Bayesian model for classification, and is also implemented in Infer.NET. Here we will use the maximum *a-posteriori* estimates of the coefficient means generated using 64 bases, to which we add a bias feature to give a 65-dimensional feature vector for the classifier. The metric of performance is the per-class one-versus-rest area under the Receiver Operating Characteristic (ROC) curve.

Table 2: Classification results on the HAD dataset. The values given are the per-class area under the ROC curve.

Activity	BDL sparse	BDL	SPAMS
Walking	0.73	0.83	0.88
Ascending stairs	0.63	0.60	0.83
Descending stairs	0.61	0.34	0.82
Sitting	0.74	0.72	0.89
Standing	0.51	0.43	0.98
Lying down	0.95	0.95	0.95
Average	0.70	0.65	0.89

The results are presented in tables 2 and 3 for the HAD and SPHERE datasets respectively. We note that whilst the classification performance of BDL on HAD is acceptable, it is markedly better for SPAMS. It would appear that despite the better reconstruction performance of BDL, the bases es-

Table 3: Classification results on the SPHERE dataset. The values given are the per-class area under the ROC curve.

Activity	BDL sparse	BDL	SPAMS
Walking	0.55	0.51	0.46
Standing	0.69	0.62	0.44
Lying down	0.86	0.85	0.46
Average	0.70	0.66	0.45

timated by SPAMS are more discriminative. However on the SPHERE dataset this trend is reversed. Wider empirical validation is required to fully understand these results.

6. CONCLUSIONS AND FURTHER WORK

We have presented a model that is an improvement on existing methods for Bayesian Dictionary Learning, and have given an efficient implementation using Variational Message Passing. We have shown that even in the over-complete settings, priors that do not explicitly enforce sparsity can still result in sparse representations, whilst giving better reconstructions. We have shown how such models can be applied to accelerometer signals, both for reconstruction, and for Activity Recognition, although it is also clear that this is a powerful approach that can be applied to a wide range of signals.

There are many possible avenues for further work. With regards to the accelerometer data itself, as it stands we have not accounted for the orientation of the device, which in general is not knowable directly from the accelerometer signal alone. There are heuristic methods to estimate the optimisation, but it would be desirable to integrate this directly into the model.

As seen above, the current pipeline would involve using the maximum *a-posteriori* estimates of the coefficient means as features in a classifier. it is conceivable however, to construct a model that incorporates both the dictionary learning and classification, in a similar fashion to the (non-Bayesian) approach of [17]. The resultant model should be able to learn bases that are simultaneously useful for reconstruction and classification.

It would be interesting to explore non-parametric approaches, such as [9], as long as the efficiency that is the result of using deterministic approximations (such as VMP), and graceful degradation with noisy or corrupted signals is retained.

Finally, it would be interesting to see if the framework can be adapted to perform Convolutional sparse coding akin to [18], for example by setting up a Toeplitz structure within the graphical model.

Source code to reproduce all of the experiments in this paper is provided at: https://github.com/IRC-SPHERE/bayesian-dictionary-learning.

7. REFERENCES

[1] D Anguita, A Ghio, L Oneto, X Parra, and JL Reyes-Ortiz, "A public domain dataset for human activity recognition using smartphones," in *ESANN*, 2013.

[2] Bruno A Olshausen et al., "Emergence of simple-cell receptive field properties by learning a sparse code for natural images," *Nature*, vol. 381, no. 6583, pp. 607–609, 1996.

[3] Lindsay I Smith, "A tutorial on principal components analysis," *Cornell University, USA*, vol. 51, no. 52, pp. 65, 2002.

[4] Radu Balan, Peter G Casazza, Christopher Heil, and Zeph Landau, "Density, overcompleteness, and localization of frames. i. theory," *Journal of Fourier Analysis and Applications*, vol. 12, no. 2, pp. 105–143, 2006.

[5] Francis Bach, Rodolphe Jenatton, Julien Mairal, and Guillaume Obozinski, "Optimization with sparsity-inducing penalties," *Foundations and Trends® in Machine Learning*, vol. 4, no. 1, pp. 1–106, 2012.

[6] Michael Grant and Stephen Boyd, "CVX: Matlab software for disciplined convex programming, version 2.1," http://cvxr.com/cvx, Mar. 2014.

[7] Julien Mairal, Francis Bach, Jean Ponce, and Guillermo Sapiro, "Online learning for matrix factorization and sparse coding," *The Journal of Machine Learning Research*, vol. 11, pp. 19–60, 2010.

[8] Linxiao Yang, Jun Fang, Hong Cheng, and Hongbin Li, "Sparse Bayesian dictionary learning with a Gaussian hierarchical model," *CoRR*, vol. abs/1503.02144, 2015.

[9] Hong Phuong Dang and Pierre Chainais, "A Bayesian non parametric approach to learn dictionaries with adapted numbers of atoms," in *Machine Learning for Signal Processing (MLSP), 2015 IEEE 25th International Workshop on*. IEEE, 2015, pp. 1–6.

[10] T. Plötz, N.Y. Hammerla, and P. Olivier, "Feature learning for activity recognition in ubiquitous computing," in *Proc. of the 22nd Int. Joint Conf. on Artificial Intell. (IJCAI)*, Toby Walsh, Ed. 2011, pp. 1729–1734, IJCAI/AAAI.

[11] Christian Vollmer, Horst-Michael Gross, and JulianP. Eggert, "Learning features for activity recognition with shift-invariant sparse coding," in *Artificial Neural Networks and Machine Learning ICANN 2013*, vol. 8131 of *Lecture Notes in Computer Science*, pp. 367–374. Springer Berlin Heidelberg, 2013.

[12] John M Winn and Christopher M Bishop, "Variational message passing," in *Journal of Machine Learning Research*, 2005, pp. 661–694.

[13] Thomas P Minka, "Expectation propagation for approximate Bayesian inference," in *Proceedings of the Seventeenth conference on Uncertainty in artificial intelligence*. Morgan Kaufmann Publishers Inc., 2001, pp. 362–369.

[14] Niall Twomey, Tom Diethe, Meelis Kull, Hao Song, Massimo Camplani, Sion Hannuna, Xenofon Fafoutis, Ni Zhu, Pete Woznowski, Peter Flach, and Ian Craddock, "The sphere challenge: Activity recognition with multimodal sensor data," *arXiv preprint arXiv:1603.00797*, 2016.

[15] T. Minka, J.M. Winn, J.P. Guiver, S. Webster, Y. Zaykov, B. Yangel, A. Spengler, and J. Bronskill, "Infer.NET 2.6," 2014, Microsoft Research Cambridge. http://research.microsoft.com/infernet.

[16] Ralf Herbrich, Thore Graepel, and Colin Campbell, "Bayes point machines," *Journal of Machine Learning Research*, vol. 1, pp. 245–279, January 2001.

[17] Julien Mairal, Jean Ponce, Guillermo Sapiro, Andrew Zisserman, and Francis R. Bach, "Supervised dictionary learning," in *Advances in Neural Information Processing Systems 21*, D. Koller, D. Schuurmans, Y. Bengio, and L. Bottou, Eds., pp. 1033–1040. Curran Associates, Inc., 2009.

[18] Hilton Bristow, Anders Eriksson, and Simon Lucey, "Fast convolutional sparse coding," in *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*. IEEE, 2013, pp. 391–398.