Learning Feature Selection Dependencies in Multi-Task Learning

Daniel Hernández-Lobato and José Miguel Hernández-Lobato

Universidad Autónoma de Madrid
Cambridge University

2013

Outline

1. Introduction

2. A Probabilistic Model for Learning Feature Selection Dependencies

3. Approximate Inference

4. Experiments

5. Conclusions

Outline

1. Introduction
2. A Probabilistic Model for Learning Feature Selection Dependencies
3. Approximate Inference
4. Experiments
5. Conclusions

Induction Under the Sparsity Assumption

We focus on linear regression problems with a small number of training instances *n* and a large number of explaining attributes or features *d*. That is, $n \ll d$.

$$\mathbf{y} = \mathbf{X}\mathbf{w} + \boldsymbol{\epsilon}, \quad \boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}\sigma^2)\,.$$

In this scenario an infinite number of values for them perfectly explain the training data [Johnstone and Titterington, 2009].

To avoid over-fitting problems and to obtain estimates with good generalization properties, a typical regularization used is to assume that the model coefficients are sparse, i.e., most coefficients are equal to zero.

Induction under the sparsity assumption can be carried out by introducing sparse enforcing priors over **w**, *e.g.* Spike-and-slab, Laplace, Student-t or Horseshoe.

Induction Under the Sparsity Assumption

Some works indicate that induction under the sparsity assumption can be improved by considering dependencies in the feature selection process [Kim et al., 2006][Van Gerven et al., 2009][Hernández-Lobato et al., 2011].

For example, the fact that one attribute is actually relevant/irrelevant for prediction could be an indicator that another related attribute should also be relevant/irrelevant.

Unfortunately, these dependences are often problem specific and have to be introduced by hand by some expert in the field.

We propose a probabilistic model that is able to induce these dependencies from the training data alone.

Outline

1 Introduction

2 A Probabilistic Model for Learning Feature Selection Dependencies

3 Approximate Inference

4 Experiments

5 Conclusions

Model Description

First, we consider only a single learning task. We assume independent additive Gaussian noise with fixed variance around the targets $\mathbf{y}$. This produces the following likelihood function for $\mathbf{w}$ given $\mathbf{X}$ and $\mathbf{w}$:

$$p(\mathbf{y}|\mathbf{X}, \mathbf{w}) = \mathcal{N}(\mathbf{y}|\mathbf{X}\mathbf{w}, \mathbf{I}\sigma^2)\,,$$

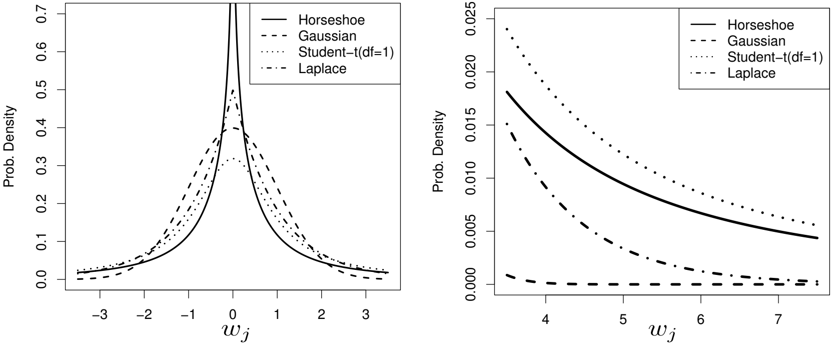
where σ^2 is the variance of the noise.

The sparsity assumption is introduced by using for $\mathbf{w}$ the horseshoe prior:

$$p(\mathbf{w}|\tau) = \int \prod_{j=1}^{d} \mathcal{N}(w_j|0, \tau^2\lambda_j^2)\mathcal{C}^+(\lambda_j)d\lambda_j\,,$$

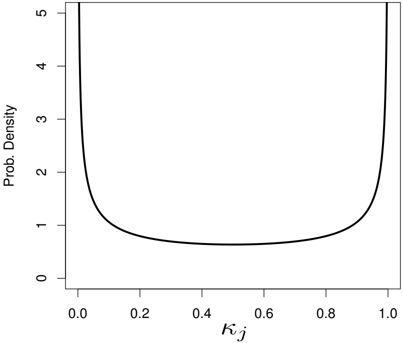
where $\mathcal{C}^+$ is a truncated Cauchy distribution and τ is a parameter that controls the level of sparsity [Carvalho, 2009].

The Horseshoe Prior



The Horseshoe Prior

Assume that $\tau = \sigma^2 = 1$, $\mathbf{X} = \mathbf{I}$ and define $\kappa_j = 1/(1 + \lambda_j^2)$. Then the posterior mean for w_j is $(1 - \kappa_j)y_j$, where κ_j is a shrinkage coefficient.



If $\kappa_j = 0$ no shrinkage occurs of $\mathbf{w}_j$. By contrast, if $\kappa_j = 1$ w_j is set equal to zero. Thus, under the horseshoe prior we only expect to see relevant coefficients or irrelevant coefficients.

The Horseshoe Prior: Introducing Dependencies

We consider the alternative representation of the prior:

$$p(\mathbf{w}|\rho^2, \gamma^2) = \int \prod_{j=1}^{d} \mathcal{N}\left(w_j|0, \frac{u_j^2}{v_j^2}\right) \mathcal{N}(u_j|0, \rho^2)\mathcal{N}(v_j|0, \gamma^2) dv_j du_j \,,$$

where $\tau^2 = \rho^2/\gamma^2$ and we have represented the Cauchy distribution as the ratio of two standard Gaussian distributions.

Both u_j^2 and v_j^2 determine feature relevancy and irrelevancy.

The prior with dependencies in the feature selection process is:

$$p(\mathbf{w}|\rho^2, \gamma^2, \mathbf{C}) = \int \left[\prod_{j=1}^{d} \mathcal{N}\left(w_j|0, \frac{u_j^2}{v_j^2}\right)\right] \mathcal{N}(\mathbf{u}|\mathbf{0}, \rho^2\mathbf{C})\mathcal{N}(\mathbf{v}|\mathbf{0}, \gamma^2\mathbf{C}) d\mathbf{u} d\mathbf{v} \,,$$

where **C** is a correlation matrix that determines the dependencies.

The correlation matrix $\mathbf{C}$

When $\mathbf{C} = \mathbf{I}$ the original prior is recovered. Otherwise, dependencies are introduced in the feature selection process.

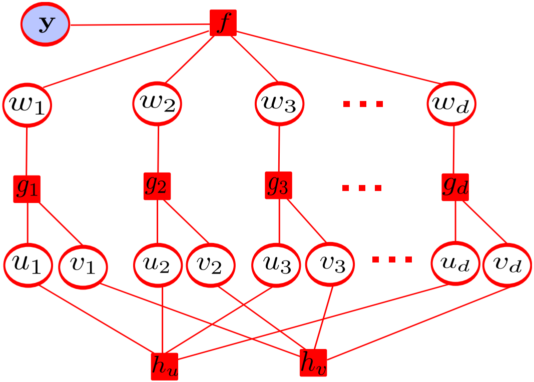
We aim at learning $\mathbf{C}$ from the training data. This can be problematic since it has $\mathcal{O}(d^2)$ parameters and we assume $n \ll d$ data instances only.

To alleviate these problems we assume the following simplified form for $\mathbf{C}$:

$$\mathbf{C} = \boldsymbol{\Delta}\mathbf{M}\boldsymbol{\Delta}\,, \quad \mathbf{M} = \mathbf{D} + \mathbf{P}\mathbf{P}^{\mathsf{T}}\,, \quad \boldsymbol{\Delta} = \mathrm{diag}(1/\sqrt{M_{11}}, \ldots, 1/\sqrt{M_{dd}})\,.$$

That is, $\mathbf{C}$ is completely specified $\mathbf{P}$ a $d \times m$ matrix and has only $\mathcal{O}(md)$ parameters. Later on we will set $m = n$ for computation purposes.

Factor Graph of the Probabilistic Model



The factor $f(\cdot)$ corresponds to the likelihood term $\mathcal{N}(\mathbf{y}|\mathbf{Xw}, \sigma^2\mathbf{I})$. The factor $g_j(\cdot)$ corresponds to the prior for w_j given u_j and v_j, $\mathcal{N}(w_j|0, u_j^2/v_j^2)$. Finally, the factors $h_u(\cdot)$ and $h_v(\cdot)$ correspond to $\mathcal{N}(\mathbf{u}|\mathbf{0}, \rho^2\mathbf{C})$ and $\mathcal{N}(\mathbf{v}|\mathbf{0}, \gamma^2\mathbf{C})$, respectively. Only the targets $\mathbf{y}$ are observed, the other variables are latent. All factors except the g_j's are Gaussian.

Inference about the Latent Variables of the Model

The joint probability of $\mathbf{y}$, and $\mathbf{z} = (\mathbf{u}, \mathbf{v}, \mathbf{w})$ is:

$$p(\mathbf{y}, \mathbf{z} | \sigma^2, \rho^2, \gamma^2, \mathbf{C}) = \mathcal{N}(\mathbf{y} | \mathbf{X}\mathbf{w}, \mathbf{I}\sigma^2) \mathcal{N}(\mathbf{u} | \mathbf{0}, \rho^2 \mathbf{C}) \mathcal{N}(\mathbf{v} | \mathbf{0}, \gamma^2 \mathbf{C})$$

$$\prod_{j=1}^{d} \mathcal{N}\left(w_j | 0, \frac{u_j^2}{v_j^2} \right)$$

The posterior for $\mathbf{z}$ is:

$$p(\mathbf{z} | \mathbf{X}, \mathbf{y}, \sigma^2, \rho^2, \gamma^2, \mathbf{C}) = \frac{p(\mathbf{y}, \mathbf{z} | \sigma^2, \rho^2, \gamma^2, \mathbf{C})}{p(\mathbf{y} | \sigma^2, \rho^2, \gamma^2, \mathbf{C})}$$

The predictive distribution for the target of a new instance $\mathbf{x}_{\text{new}}$ is:

$$p(y_{\text{new}} | \mathbf{x}_{\text{new}}, \mathbf{X}, \mathbf{y}, \sigma^2, \rho^2, \gamma^2, \mathbf{C}) = \int p(y_{\text{new}} | \mathbf{x}_{\text{new}}, \mathbf{w}) p(\mathbf{z} | \mathbf{X}, \mathbf{y}, \sigma^2, \rho^2, \gamma^2, \mathbf{C}) d\mathbf{z}$$

Learning Feature Selection Dependencies

In an ideal situation we should also specify a prior for **C** and compute a posterior distribution for **C**. However, this can be too complicated even for approximate inference methods.

A simpler alternative is to use gradient ascent to maximize the denominator in Bayes theorem with respect to **C**, σ^2, ρ^2 and γ^2. This corresponds to type-II maximum likelihood estimation and allows to estimate **C** from the training data alone.

$$\log Z = \log p(\mathbf{y}|\sigma^2, \rho^2, \gamma^2, \mathbf{C})$$

Unfortunately, even if the required computations were possible, the estimation of **C** would be difficult as a consequence of the reduced amount of observed data. To compensate for this, we consider a multi-task learning setting where more data are available.

Extension to Address Multi-task Learning Problems

We assume that there are K learning tasks $\{\mathbf{y}\}_{k=1}^K$ and $\{\mathbf{X}\}_{k=1}^K$ available for induction that only share the dependencies specified by $\mathbf{C}$.

The posterior distribution of the latent variables $\{\mathbf{z}_k\}_{k=1}^K$ is:

$$p(\{\mathbf{z}\}_{k=1}^K | \{\mathbf{X}_k, \mathbf{y}_k, \sigma_k^2, \rho_k^2, \gamma_k^2\}_{k=1}^K, \mathbf{C}) = \prod_{k=1}^K \frac{p(\mathbf{y}_k, \mathbf{z}_k | \sigma_k^2, \rho_k^2, \gamma_k^2, \mathbf{C})}{p(\mathbf{y}_k | \sigma_k^2, \rho_k^2, \gamma_k^2, \mathbf{C})}$$

That is, the posterior distribution factorizes.

The log of the marginal likelihood for the observed data of the tasks is:

$$\log Z_{\mathrm{MT}} = \sum_{k=1}^K \log p(\mathbf{y}_k | \sigma_k^2, \rho_k^2, \gamma_k^2, \mathbf{C}) = \sum_{k=1}^K \log Z_k$$

Thus, if there is a way to evaluate the required quantities for a single task, applying the multi-task extension is straight-forward.

Outline

1. Introduction

2. A Probabilistic Model for Learning Feature Selection Dependencies

3. Approximate Inference

4. Experiments

5. Conclusions

Expectation Propagation

For simplicity we consider the model for a single task. Up to a normalization constant, the posterior is:

$$p(\mathbf{z}|\mathbf{X},\mathbf{y},\sigma^2,\rho^2,\gamma^2,\mathbf{C}) \propto f(\mathbf{w})h_u(\mathbf{u})h_v(\mathbf{v})\prod_{j=1}^{d} g_j(\mathbf{z})$$

where all factors except the g_j's are Gaussian.

EP [Minka, 2001] approximates the posterior by:

$$q(\mathbf{z}) \propto f(\mathbf{w})h_u(\mathbf{u})h_v(\mathbf{v})\prod_{j=1}^{d} \tilde{g}_j(\mathbf{z})$$

where each $\tilde{g}_j$ is an un-normalized Gaussian which approximates the corresponding g_j.

The Gaussian distribution belongs to the exponential family which is closed under the product. Thus, q is Gaussian.

Expectation Propagation II

EP finds good approximate factors $\tilde{g}_j$ as follows:

1. Remove g_j from the approximation q by computing $q^{\setminus j} \propto q / \tilde{g}_j$.
2. Find q^{new} by minimizing the Kullback-Leibler divergence, $\text{KL}(g_j q^{\setminus j} || q^{\text{new}})$, with respect to q^{new}. This is a convex problem.
3. Update $\tilde{g}_j$ by setting $\tilde{g}_j^{\text{new}} = s_j q^{\text{new}} / q^{\setminus j}$ where s_j is the normalization constant of $g_j q^{\setminus j}$. $\tilde{g}_j$ is Gaussian because q^{new} and $q^{\setminus j}$ are Gaussians.

These steps are iterated by EP until convergence of all the $\tilde{g}_j$.

The minimization of the KL divergence is done by matching moments between $g_j q^{\setminus j}$ and q^{new}. The moments are obtained from the derivatives of $\log s_j$ with respect to the natural parameters of $q^{\setminus j}$ [Seeger, 2006].

Unfortunately, the computation of s_j is intractable under the horseshoe prior. As a practical alternative we use quadrature methods to evaluate s_j and its derivatives. This works well and EP converges successfully.

EP Quadratures

In practice we have that g_j and $q^{\setminus j}$ only depend on w_j, u_j and v_j:

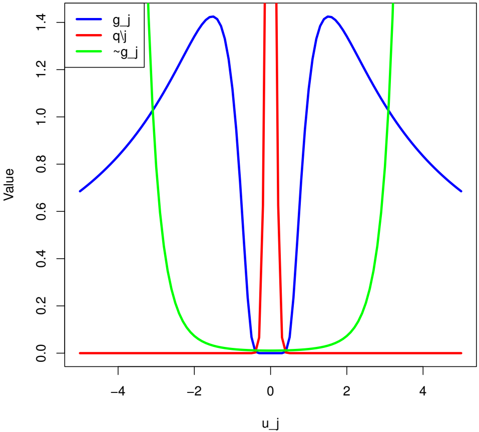
$$q^{\setminus j}(w_j, u_j, v_j) = \mathcal{N}(w_j|m_j, \eta_j^2)\mathcal{N}(u_j|0, \nu_j^2)\mathcal{N}(v_j|0, \xi_j^2)\,,$$
$$g_j(w_j, u_j, v_j) = \mathcal{N}(w_j|0, u_j^2/v_j^2)\,,$$

so it may seem that computing s_j requires a three-dimensional quadrature. A more efficient alternative exists since:

$$s_j = \int q^{\setminus j}(w_j, u_j, v_j)g_j(w_j, u_j, v_j)dw_j du_j dv_j$$
$$= \int \mathcal{N}(m_j|0, \frac{\nu_j^2}{\xi_j^2}\lambda_j^2 + \eta_j^2)\mathcal{C}^+(\lambda_j|0, 1)d\lambda_j\,.$$

Five one-dimensional quadratures will suffice. One to compute $\log s_j$ and four to evaluate its derivatives (the posterior means of $\mathbf{v}$ and $\mathbf{u}$ are zero).

EP: Exact Factor g_j and Approximate factor $\tilde{g}_j$



The factor $\tilde{g}_j$ looks similar to the exact factor g_j in regions of high posterior probability as described by $q^{\setminus j}$. We let u_j vary and we fix $v_j^2 = \xi^2$ and $w_j = m_j$.

EP Approximation of the Model Evidence

Once EP has converged q can be used to make predictions:

$$p(y_{\text{new}}|\mathbf{x}_{\text{new}}, \mathbf{X}, \mathbf{y}, \sigma^2, \rho^2, \gamma^2, \mathbf{C}) \approx \int p(y_{\text{new}}|\mathbf{x}_{\text{new}}, \mathbf{w}) q(\mathbf{z}) d\mathbf{z}$$

The model evidence is approximated by the normalization constant of q:

$$\tilde{Z} = \int f(\mathbf{w}) h_u(\mathbf{u}) h_v(\mathbf{v}) \prod_{j=1}^{d} \tilde{g}_j(\mathbf{z}) d\mathbf{z}$$

These expressions involve Gaussians and can be readily computed.

After convergence the gradient of the natural parameters of the $\tilde{g}_j$ with respect to the hyper-parameters σ^2, ρ^2, γ^2 and $\mathbf{C}$ is zero [Seeger, 2006].

The gradient of $\tilde{Z}$ with respect to the hyper-parameters can be computed in terms of the gradient of the exact factors $f(\mathbf{w})$, $h_u(\mathbf{u})$ and $h_v(\mathbf{v})$.

The total cost of EP and computing the gradients is $\mathcal{O}(n^2 d)$ if $m = n$.

Complete EP algorithm for Multi-Task Learning

The complete EP algorithm consists in iteratively repeating the following steps until convergence of the hyper-parameters σ_k^2, ρ_k^2, γ_k^2 and $\mathbf{C}$:

1. For each task $k = 1$ to K:
    1. Compute EP approximation for the posterior distribution of task k.
    2. Compute gradient of the log of the model evidence with respect to the hyper-parameters.
2. Sum the gradients of the matrix $\mathbf{P}$, *i.e.* the matrix that fully determines $\mathbf{C}$.
3. Update the different hyper-parameters by taking a small step in the direction of the resulting gradients.

The last EP approximation obtained for each task can be used as the starting point of EP in the next iteration reducing significantly the iterations needed to reach convergence in EP.

Outline

1 Introduction

2 A Probabilistic Model for Learning Feature Selection Dependencies

3 Approximate Inference

4 Experiments

5 Conclusions

Related Methods to Compare With and Hyper-Parameters

We compare the proposed model, HS_{Dep}, with other methods from the literature for multi-task learning under the sparsity assumption.

- HS_{ST}: Particular case of HS_{Dep} that is obtained by learning each task separately with no dependencies (base line).
- HS_{MT}: Uses the horseshoe prior but assumes jointly relevant and irrelevant features across learning tasks. λ_j is shared among learning tasks. Approximate inference is also carried out by EP [Hernandez-Lobato et al., 2010].
- DM: Dirty model for multi-task learning. Based on a combined ℓ_1 and ℓ_∞ norm. Allows some of the tasks to have specific relevant features [Jalali et al., 2010].

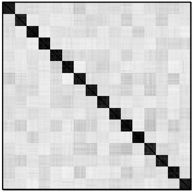
In HS_{ST} we learn ρ_k^2 and γ_k^2 $\forall k$ by type-II maximum likelihood. In HS_{MT} γ^2 and ρ^2 take the average value found by HS_{ST}. In HS_{Dep} we use the hyper-parameters found by HS_{ST} and find $\mathbf{P}$ by type-II maximum likelihood. In DM we try different hyper-parameters and report results for the best performing ones.

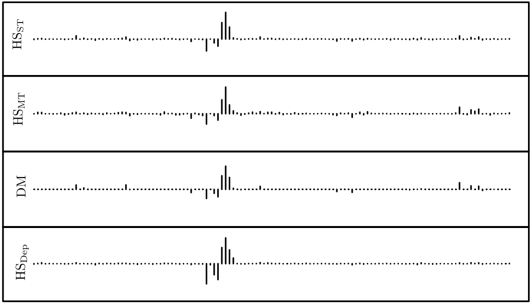
Experiments with Synthetic Data

We generate $K = 64$ learning tasks of $n = 64$ samples and $d = 128$ features each. In each task $\mathbf{X}_k$ is generated from a Gaussian and $\mathbf{w}_k$ is set equal to zero except for consecutive groups of 8 coefficients. $\sigma_k = 0.5$.

The task of interest is to reconstruct $\mathbf{w}_k$ $\forall k$ from the training data. The reconstruction error for each task is measured in terms of $||\hat{\mathbf{w}}_k - \mathbf{w}_k||_2/||\mathbf{w}_k||_2$ where $\mathbf{w}_k$ are the exact model coefficients for task k.

Method	Error
HS_{ST}	0.29±0.01
HS_{MT}	0.38±0.03
DM	0.37±0.01
HS_{Dep}	0.21±0.01





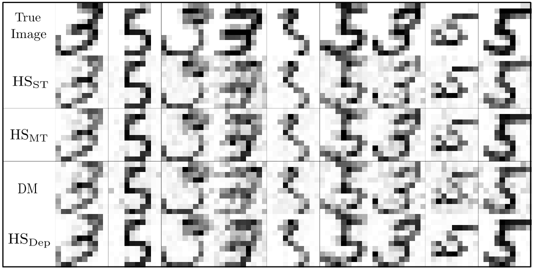
Averages over 50 realizations

Reconstruction of Images of Hand-written Digits

We consider the reconstruction of images of hand-written digits extracted from the MNIST data set [LeCun et al., 1998].

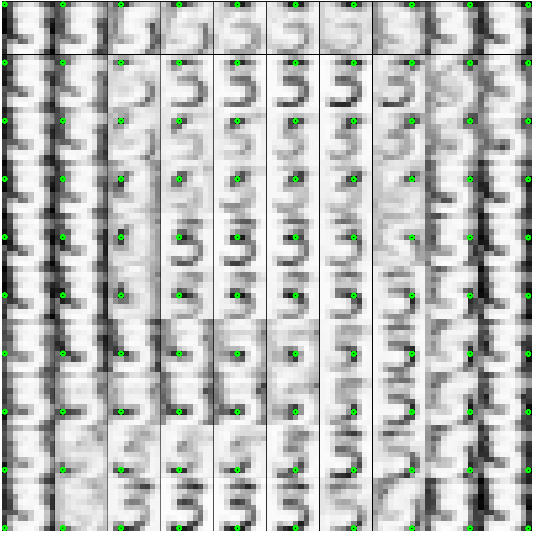
These images are sparse. The images are scaled to 10 × 10 pixels and $K = 100$ tasks of $n = 75$ samples are generated by choosing randomly 50 images of the digit 3 and 50 images of the digit 5. $\mathbf{X}_k$ is also standard Gaussian and $\sigma_k = 0.1$. The average reconstruction error is also measured.

	$\mathrm{HS}_{\mathrm{ST}}$	$\mathrm{HS}_{\mathrm{MT}}$	DM	$\mathrm{HS}_{\mathrm{Dep}}$
Error	0.36±0.02	0.25±0.02	0.37±0.01	0.2±0.01



Averages over 50 realizations

Dependencies Learnt from the Training Data



Average correlations in absolute value of each pixel (green) with the other ones as indicated by the correlation matrix **C** learnt from the training data.

Outline

1. Introduction
2. A Probabilistic Model for Learning Feature Selection Dependencies
3. Approximate Inference
4. Experiments
5. Conclusions

Conclusions

- We have described a model for learning dependencies in the feature selection process from the training data alone.
- The model can be used in a multi-task learning setting where several learning tasks share dependencies in the feature selection process.
- This is a more flexible assumption and the different tasks can have different model coefficients and different relevant features.
- Exact Bayesian inference is infeasible for the proposed model. However, EP offers an approximate alternative.
- The total cost of the model is $\mathcal{O}(Kn^2d)$, linear in the number of features and tasks.
- Our experiments show that the proposed model performs better than other multi-task learning alternatives from the literature.
- The proposed model is able to induce relevant feature selection dependencies from the training data.

References

- I.M. Johnstone and D.M. Titterington. Statistical challenges of high-dimensional data. Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences, 367(1906):4237, 2009.
- Y. Kim, J. Kim, and Y. Kim. Blockwise sparse regression. Statistica Sinica, 16(2):375, 2006.
- M. Van Gerven, B. Cseke, R. Oostenveld, and T. Heskes. Bayesian source localization with the multivariate Laplace prior. Advances in Neural Information Processing Systems 22, pages 1901-1909, 2009.
- D. Hernandez-Lobato, J. M. Hernandez-Lobato, Helleputte T., and P. Dupont. Expectation propagation for Bayesian multi-task feature selection. Proceedings of the European Conference on Machine Learning, volume 6321, pages 522-537. Springer, 2010.
- J. M. Hernandez-Lobato, D. Hernandez-Lobato, and A. Suarez. Network-based sparse Bayesian classification. Pattern Recognition, 44:886-900, 2011.
- A. Jalali, P. Ravikumar, S. Sanghavi, and C. Ruan. A dirty model for multi-task learning. Advances in Neural Information Processing Systems 23, pages 964-972. 2010.
- C.M. Carvalho, N.G. Polson, and J.G. Scott. Handling sparsity via the horseshoe. Journal of Machine Learning Research W&CP, 5:73-80, 2009.
- T. Minka. A Family of Algorithms for approximate Bayesian Inference. PhD thesis, Massachusetts Institute of Technology, 2001.
- M. Seeger. Expectation propagation for exponential families. Technical report, Department of EECS, University of California, Berkeley, 2006.
- Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. Proceedings of the IEEE, 86(11):2278-2324, 1998.

Thank you for your attention!