BAYESIAN SUPERVISED LEARNING WITH NON-GAUSSIAN LATENT VARIABLES

Siwei Lyu

Computer Science Department
University at Albany, SUNY
slyu@albany.edu

ABSTRACT

We describe a Bayesian learning scheme for the hierarchal Bayesian linear model, which is based on the Gaussian scale mixture (GSM) modeling of the distribution of the latent variable. The proposed method takes advantage of the hierarchal Gaussian structure for a simple Monte-Carlo sampling algorithm. Particularly, with a single hidden scale parameter controlling the distribution of the latent variables, it leads to an efficient algorithm without explicit matrix inversion.

Index Terms— Gaussian scale mixtures, Bayesian learning, latent variable models

1. INTRODUCTION

Hierarchal Bayesian modeling with latent variables has been widely used in machine learning recently. For instance, Gaussian latent variables were used for Bayesian learning in relevant vector machines (RVM) [1], with different scales for independent Gaussian-distributed weights to control the sparseness of the resulted solution. In [2], non-Gaussian latent variables with a Jeffery's non-informative prior over scale parameters was used. In [3], a general Bayesian estimation framework is described based on Gaussian scale mixture latent variables and variational EM algorithm. Common to most existing methods is the general methodology of first obtaining a point estimation (maximum likelihood or maximum *a posterior*) for the latent variables, and then plugging the estimated variables to predict new function evaluations.

Treating the latent variables as unknown yet deterministic entities, such point estimation methods do not take into account the variability of the estimators, which elicits several drawbacks. First, such algorithms are usually iterative, which renders the estimation biased and sensitive to the choice of initial starting point. Furthermore, the prediction variance, which is an important measurement of confidence or quality in the estimation, is computed approximately, and in the case of RVM, wrong [4]. In this paper, a more complete Bayesian view is taken by treating the latent variable stochastically, and modeling the predictive posterior distribution of function evaluations for arbitrary inputs. We take advantage of scale mixture structure of the non-Gaussian latent variables to achieve a simple Monte-Carlo sampling based algorithm.

2. GSM BAYESIAN LINEAR MODEL

We consider the generalized linear model, which assumes that the observations y_i, as

$$y_i = f(\mathbf{x}_i, \mathbf{b}) + w_i,$$

for $i = 1, \cdots, n$ are function evaluations of

$$f(\mathbf{x}, \mathbf{b}) = h(\mathbf{x})'\mathbf{b},$$

corrupted by independent noise w_i. We further denote the dimensionality of $\mathbf{b}$ as m and assume $m \leq n$. The generalized linear model is a flexible probabilistic model: different choices of the mapping function $h(\cdot)$ and noise density result in different types of learning problems, including linear, nonlinear and kernel-based regression and classification [5]. To facilitate the subsequent description, we use the more concise vector form:

$$\mathbf{y} = H\mathbf{b} + \mathbf{w}, \tag{1}$$

where H is the $n \times m$ design matrix containing evaluations of $h(\cdot)$ over n different inputs.

In Bayesian linear model (BLM), the parameter $\mathbf{b}$ in (1) is treated as a random latent variable independent from the input noise $\mathbf{w}$. In the non-Gaussian hierarchal Bayesian linear model [3], the distribution of $\mathbf{b}$ is further modeled as

$$p(\mathbf{b}) = \prod_i p(b_i),$$

where $p(b_i)$ is a Gaussian scale mixture (GSM), whose density is an infinite mixture of zero-mean Gaussians of different scales, as

$$\begin{aligned} p(b_i) &= \int_0^\infty \frac{1}{\sqrt{2\pi s}} \exp\left(-\frac{b_i^2}{2s_i}\right) \phi(s_i) ds_i \\ &= \int_0^\infty \mathcal{N}_{b_i}(0, s_i) \phi(s_i) ds_i. \end{aligned}$$

We further denote $\mathbf{s} = (s_1, \cdots, s_m)^T$ as the vector containing all independent scale variables. By choosing different scale distribution $\phi(s)$, GSM can model a wide variety of symmetric non-Gaussian densities [6], including the Student's t, α-stable and generalized Laplacians. In this paper, we consider two special cases where (i) all s_i are independent samples from one distribution $\phi(s)$ and (ii) each s_i is an identical copy of a single sample from $\phi(s)$. The first is an example of what known in statistics as the spherically invariant random process (SIRP) [7], and the latter an example of the spherically invariant random vectors (SIRV) [8].

In prediction with the BLM, a point estimation of $\mathbf{b}$ is irrelevant, as one needs the predictive posterior distribution for a new input $\mathbf{h} = h(\mathbf{x})$ evaluated over all possible $\mathbf{b}$, which, when marginalized over scale variables $\mathbf{s}$, is

$$p(\mathbf{h}'\mathbf{b}|\mathbf{y}; H) = \int_{\mathbf{s}} p(\mathbf{s}|\mathbf{y}; H) p(\mathbf{h}'\mathbf{b}|\mathbf{s}, \mathbf{y}; H),$$

to perform optimal prediction. Bayesian regression proceeds from the predictive posterior distribution and prechosen loss function for $\mathbf{h}'\mathbf{b}$. Specifically, a commonly used loss function is the l_2 loss, which results in Bayesian least square estimation (BLS). The optimal solution to the BLS is the conditional mean, $\mathcal{E}(\mathbf{h}'\mathbf{b}|\mathbf{y}; H)$, with the prediction variance (or the confidence in the estimation) given by the conditional variance $\mathrm{var}(\mathbf{h}'\mathbf{b} \mid \mathbf{y}; H)$.

2.1. SIRP model with Gaussian noise

To obtain the predictive distribution $p(\mathbf{h}'\mathbf{b}|\mathbf{y}; H)$, we start with the posterior distribution of $\mathbf{b}$ given the latent variable $\mathbf{s}$ and observation $\mathbf{y}$, which is

$$p(\mathbf{b}|\mathbf{s}, \mathbf{y}; H) = p(\mathbf{y}|H\mathbf{b})p(\mathbf{b}|\mathbf{s})/p(\mathbf{y}, \mathbf{s}; H).$$

Note that the denominator this is the product of two Gaussians. As the product of two Gaussians is another unnormalized Gaussian [9], and $p(\mathbf{y}, \mathbf{s}; H)$ providing the normalizing constant, $p(\mathbf{b}|\mathbf{s}, \mathbf{y}; H)$ is also a Gaussian $\mathcal{N}_{\mathbf{b}}(\mu(\mathbf{s}), C(\mathbf{s}))$. Assuming independent Gaussian noise,

$$\mathbf{w} \sim \mathcal{N}_{\mathbf{w}}(0, \sigma_w^2 I),$$

it yields

$$C(\mathbf{s}) = \left(\sigma_w^{-2} H' H + \mathrm{diag}(\mathbf{s})^{-1}\right)^{-1}$$

and

$$\mu(\mathbf{s}) = \sigma_w^{-2} C(\mathbf{s}) H' \mathbf{y}.$$

Here we introduce operator $\mathrm{diag}(\cdot)$ that extends a vector to a corresponding diagonal matrix. Summarizing these results, and using the fact that the projection of a multivariate Gaussian variable on a fixed direction is a 1-D Gaussian [9], we have

$$p(\mathbf{h}'\mathbf{b}|\mathbf{s}, \mathbf{y}; H) = \mathcal{N}_{\mathbf{h}'\mathbf{b}}\left(\mathbf{h}'\mu(\mathbf{s}), \mathbf{h}'C(\mathbf{s})\mathbf{h}\right). \tag{2}$$

One can use Bayes' theorem to yield:

$$p(\mathbf{s}|\mathbf{y}; H) = \frac{p(\mathbf{y}|\mathbf{s}; H)\phi(\mathbf{s})}{\int_{\mathbf{s}} p(\mathbf{y}|\mathbf{s}; H)\phi(\mathbf{s}) d\mathbf{s}}, \tag{3}$$

where the first term in the numerator is the likelihood of observed data $\mathbf{y}$ given the hidden scale variable $\mathbf{s}$, which is, conveniently, another Gaussian:

$$p(\mathbf{y}|\mathbf{s}; H) = \mathcal{N}_{\mathbf{y}}\left(0, C_y(\mathbf{s})\right) \text{ and } C_y(\mathbf{s}) = H\,\mathrm{diag}(\mathbf{s})H' + \sigma_w^2 I. \tag{4}$$

With all these results, we can compute the predictive posterior distribution of $\mathbf{h}'\mathbf{b}$ as:

$$p(\mathbf{h}'\mathbf{b}|\mathbf{y}; H) = \frac{\int_{\mathbf{s}} p(\mathbf{h}'\mathbf{b}|\mathbf{s}, \mathbf{y}; H) p(\mathbf{y}|\mathbf{s}; H)\phi(\mathbf{s})}{\int_{\mathbf{s}} p(\mathbf{y}|\mathbf{s}; H)\phi(\mathbf{s})}. \tag{5}$$

Generally $p(\mathbf{h}'\mathbf{b}|\mathbf{y}; H)$ is a non-Gaussian distribution. However, for BLS prediction of $\mathbf{h}'\mathbf{b}$, it suffices to know the conditional mean of $\mathbf{h}'\mathbf{b}$ given observation $\mathbf{y}$ and design matrix H, where

$$\mathcal{E}(\mathbf{h}'\mathbf{b}|\mathbf{y}; H) = \int_{\mathbf{s}} p(\mathbf{s}|\mathbf{y}; H) \int_{\mathbf{h}'\mathbf{b}} \mathbf{h}'\mathbf{b} \cdot p(\mathbf{h}'\mathbf{b}|\mathbf{y}, \mathbf{s}; H).$$

The inner integral computes the mean of a Gaussian $p(\mathbf{h}'\mathbf{b}|\mathbf{y}, \mathbf{s}; H)$, which is

$$\mathcal{E}(\mathbf{h}'\mathbf{b}|\mathbf{y}; H) = \frac{\int_{\mathbf{s}} \mathbf{h}'\mu(\mathbf{s}) \cdot p(\mathbf{y}|\mathbf{s}; H)\phi(\mathbf{s})}{\int_{\mathbf{s}} p(\mathbf{y}|\mathbf{s}; H)\phi(\mathbf{s})} \tag{6}$$

The conditional variance is computed as

$$\mathrm{var}(\mathbf{h}'\mathbf{b}|\mathbf{y}; H) = \mathcal{E}(\left(\mathbf{h}'\mathbf{b}\right)^2 |\mathbf{y}, H) - \left(\mathcal{E}(\mathbf{h}'\mathbf{b}|\mathbf{y}, H)\right)^2,$$

where the first term is the second-order moment of a Gaussian $p(\mathbf{h}'\mathbf{b}|\mathbf{y}, \mathbf{s}, H)$, which, from Eq.(2), is computed as

$$\mathbf{h}'C(\mathbf{s})\mathbf{h} + \left(\mathbf{h}'\mu(\mathbf{s})\right)^2 = \mathbf{h}'\left(C(\mathbf{s}) + \mu(\mathbf{s})\mu(\mathbf{s})'\right)\mathbf{h}.$$

Therefore, we have

$$\mathrm{var}(\mathbf{h}'\mathbf{b}|\mathbf{y}; H) = \frac{\int_{\mathbf{s}} \mathbf{h}'\left(C(\mathbf{s}) + \mu(\mathbf{s})\mu(\mathbf{s})'\right)\mathbf{h} \cdot p(\mathbf{y}|\mathbf{s}; H)\phi(\mathbf{s})}{\int_{\mathbf{s}} p(\mathbf{y}|\mathbf{s}; H)\phi(\mathbf{s})} - \left(\mathcal{E}(\mathbf{h}'\mathbf{b}|\mathbf{y}, H)\right)^2. \tag{7}$$

Eq.(6) and (7) are both stochastic integrals. If N independent samples of $\mathbf{s}$ are obtained, the stochastic integral in Eq.(6) and (7) can be approximated with corresponding averages of the functions over these samples. These Monte-Carlo approximations give unbiased estimation of the actual stochastic integrals in Eq.(6) and (7), with the estimation variances decreasing proportionally to the number of samples drawn. In general, one can draw independent samples from $\phi(\mathbf{s})$ with Monte-Carlo Markov chain (MCMC) sampling. In the case where the BLM is modeled as SIRV, a simple Monte-Carlo sampling can be performed on the jointly independent scale distribution. More conveniently, a large class of nonnegative latent variables s can be generated by passing a Gaussian random variable v through an appropriate non-linear mapping $g : \mathcal{R} \to \mathcal{R}^+$ ([10], Proposition 2). Thus, the distribution of **s** can be constructed from a non-linearly mapped Gaussian variable, and efficient sampling from Gaussians is a well-established procedure in statistics. We can then sample **s** by first sampling from a Gaussian distribution and then nonlinearly transform the Gaussian samples.

2.2. SIRV model with Gaussian noise

Modeling hidden scale variables as mutually independent simplifies the computation of $\mathcal{E}(\mathbf{h}'\mathbf{b}|\mathbf{y}, H)$ and $\mathrm{var}(\mathbf{h}'\mathbf{b}|\mathbf{y}; H)$, yet for each sample, we need to compute the inversion of $C(\mathbf{s})$ and $C_y(\mathbf{s})$. With the SIRV formulation, where s is modeled as a single random variable with distribution $\phi(s)$, a more efficient algorithm can be obtained. Specifically, with such a setting, Eq.(4) is simplified into:

$$\begin{aligned} C(s) &= s\sigma_w^2 \left(sH'H + \sigma_w^2 I\right)^{-1} && (8) \\ \mu(s) &= s\left(sH'H + \sigma_w^2 I\right)^{-1} H'\mathbf{y}, \\ C_y(s)^{-1} &= \left(sHH' + \sigma_w^2 I\right)^{-1} && (9) \\ &= \sigma_w^{-2} I - \sigma_w^{-4} H C(s) H'. && (10) \end{aligned}$$

Let $\{U, \Lambda\}$ be the eigenvector/eigenvalue decomposition of $H'H$, then $C(s)$ can be simplified with a double diagonalization as $C(s) = s\sigma_w^2 U \left(s\Lambda + \sigma_w^2 I\right)^{-1} U'$. Subsequently, we can compute

$$\mathbf{h}'\mu(s) = s\mathbf{h}'U(s\Lambda + \sigma_w^2 I)^{-1}U'H'\mathbf{y} = \sum_{i=1}^{m} \frac{s v_i u_i}{s\lambda_i + \sigma_w^2} \qquad (11)$$

where $\mathbf{v} = U'\mathbf{h}$, $\mathbf{u} = U'H'\mathbf{y}$ and $\lambda_1 \cdots \lambda_m$ are diagonal entries of matrix Λ, and

$$\mathbf{h}'C(s)\mathbf{h} = s\sigma_w^2\mathbf{h}'U(s\Lambda + \sigma_w^2 I)^{-1}U'\mathbf{h} = \sum_{i=1}^{m} \frac{s\sigma_w^2 v_i^2}{s\lambda_i + \sigma_w^2} \qquad (12)$$

Note that

$$p(\mathbf{y}|s; H) \propto \sqrt{|C_y^{-1}|} \exp\left(-\frac{1}{2}\mathbf{y}'C_y^{-1}\mathbf{y}\right).$$

With Eq.(10), the exponential part is simplified to

$$\exp\left(-\frac{1}{2}\mathbf{y}'C_y(s)^{-1}\mathbf{y}\right) \propto \exp\left(\frac{1}{2\sigma_w^2}\sum_{i=1}^{m}\frac{s u_i^2}{s\lambda_i + \sigma_w^2}\right).$$

Furthermore, let $\{V, \Gamma\}$ be the eigenvector/eigenvalue decomposition of matrix HH'. Note that matrices HH' and $H'H$ share the same set of positive eigenvalues. Therefore, we have

$$|C_y(s)^{-1}| = |V(s\Lambda + \sigma_w^2)^{-1}V'| \propto \prod_{i=1}^{m} \frac{1}{s\lambda_i + \sigma_w^2}.$$

Simplifying by removing all constants with regards to s, we have

$$p(\mathbf{y}|s; H) \propto \prod_{i=1}^{m} \frac{1}{\sqrt{s\lambda_i + \sigma_w^2}} \exp\left(\frac{1}{2\sigma_w^2}\sum_{i=1}^{m}\frac{s u_i^2}{s\lambda_i + \sigma_w^2}\right). \qquad (13)$$

The major advantage of this algorithm, compared to the more general algorithm in the previous section, is that the diagonalization of $H'H$ and the computation of **u** and **v** does not depend on the value of s, thus need only be computed once for all samples. Similar diagonalization techniques have been used in [11]. Note that there is no matrix inversion used in Eq.(11), (12) and (13), which replace the essential computations steps in the algorithm in Section 2.1. Sampling from a 1-D distribution $\phi(s)$ is relatively easy and can be implemented numerically. The choice of $\phi(s)$ affects the performance of the Bayesian predictor, and in this paper, we use the lognormal distribution, which has been proposed as a flexible form for priors in GSM [11], defined as

$$\phi(s) = \frac{1}{\sqrt{2\pi\sigma_s^2}\,s} \exp\left(-\frac{(\log s - \mu_s)^2}{2\sigma_s^2}\right), \qquad (14)$$

with parameters μ_s and σ_s^2 being the mean and variance, respectively. With this explicit functional form of $\phi(s)$, (6) and (7) can also be directly computed with numerical integration (e.g., Gaussian quadratures). Note also that we can truncate the eigendecomposition of $H'H$ to obtain sparse approximation to the full-blown solution to the problem, which will facilitate inference on large training data sets.

2.3. Learning hyperparameters

In the Bayesian least square estimation of the hierarchal BLM, (1), as the latent variables **b**, **s** and **t** are integrated out, the only parameters needs to be determined are those in the scale distribution $\phi(s)$ and $\psi(t)$, if there is any. One choice is to use a parameter-free Jeffery's non-informative prior over the scale parameters [2]. However, Jeffery's prior, though obviate the estimation of hyper-parameters, is not normalizable, and thus does not give rise a proper predictive distribution. Another viable approach is to use the variational EM method to estimate the hyperparameters, as in [3].

In this work, assuming $\phi(s)$ and $\psi(t)$ being lognormal, we take a direct maximum likelihood estimation of the hyperparameters in the prior distributions. To be more specific, we rewrite as $\phi(s; \alpha)$ and $\psi(t; \beta)$ with $\alpha = (\mu_s, \sigma_s^2)$ and $\beta = (\mu_t, \sigma_t^2)$ being the corresponding hyperparameters. Then the likelihood of observation data **y** is given as

$$p(\mathbf{y}|\alpha, \beta; H) = \int_{s,t} N_{\mathbf{y}}(0, C_y(s,t))\phi(s; \alpha)\psi(t; \beta)dsdt. \qquad (15)$$

We can then compute the gradient of the data likelihood with regards to α and β as

$$\begin{aligned} \frac{\partial p(\mathbf{y}|\alpha, \beta; H)}{\partial \alpha} &= \int_{s,t} N_{\mathbf{y}}(0, C_y(s,t))\frac{\partial \phi(s; \alpha)}{\partial \alpha}\psi(t; \beta)dsdt \\ \frac{\partial p(\mathbf{y}|\alpha, \beta; H)}{\partial \beta} &= \int_{s,t} N_{\mathbf{y}}(0, C_y(s,t))\phi(s; \alpha)\frac{\partial \psi(t; \beta)}{\partial \beta}dsdt \end{aligned}$$

Note this gradient can be computed from numerical integration, and we take a general gradient ascent algorithm to search the best set of hyperparameters.

3. EXPERIMENTS

We start with the synthetic examples used in [1], where training data were generated by adding noise to the evaluation of the sinc function $\sin(x)/x$. Two types of noises were used in the experiments: independent Gaussian and the non-Gaussian t-noise. We randomly chose 100 points in the range $[-10, 10]$ and their function evaluation as training data and use the same spline kernel as in [1]. For comparisions, an RVM and an adaptive sparsity Bayesian regressor [2] were also trained and tested on the same set of data. To reduce the random fluctuations in computing the stochastic integrals, results are averaged over runnings of 100 random splitting of training and testing data. In Table 1, we compared the effects of different modeling considerations to the prediction performance in terms of rms, with the SIRV Bayesian predictor as the base line. Specifically, we compared the performances of SIRP Bayesian prediction, *i.e.*, using independent lognormal scale variables for each observation, using a fixed prior without explicitly learning the hyperparameters, (a log normal prior with mean zero and variance 1), and truncating in the eigenvalues/vectors of $H'H$ in forming the predictor. With the more flexible SIRP Bayesian prediction, the performance is significantly improved, but the improvement is levied by longer running time and higher prediction variance (not shown) due to the more complicated sampling process and use of matrix inversion in each step. On the other hand, we found that performance is relatively less affected by using a generic lognormal prior without a searching for best set of parameter. Finally, truncating the eigenvalues/vectors of $H'H$ have varying impacts on the prediction performance. Specifically, keeping only top 10% of eigenvalues/vectors of $H'H$ severely reduces the performance, despite the signifiant improvement in the running time of the final predictor, while a truncation of 50% tailing eigenvalues/vectors introduce relatively small performance degradation. The performance of the proposed method is also compared with RVM and adaptive sparse Bayesian on some benchmarking data sets, including the Friedman data sets [12] and the Boston housing data sets [13] (100 random partitions of the full data set into 420 training samples and 86 testing samples), and the results are listed in Table 2. In these tests, the SIRV Bayesian prediction achieved overall comparable performance to other state-of-the-art methods, while able to outperform them on some of the benchmark data sets.

4. DISCUSSION

In this paper, we discuss a Bayesian learning scheme for the hierarchal Bayesian linear model. This is based on the Gaussian scale mixture (GSM) modeling of the distribution of the latent variable. The proposed work is a Bayesian least square implementation of the prediction problem in the hierarchal Bayesian linear model, Eq.(1). The Bayesian learning takes advantage of the hierarchal Gaussian structure for a simple Monte-Carlo sampling algorithm. Particularly, with a single hidden scale parameter controlling the distribution of the latent variables, there is a more efficient algorithm without explicit matrix inversion. The same framework can also be extended to classification problems, which is under investigation currently. Important future works include investigation of the relation with non-parametric Bayesian learning schemes such as Gaussian processes, and efficient algorithms for larger training sets or higher data dimensions.

Acknowledgement. This work is supported by a National Science Foundation of the United States CAREER Award IIS-0953373, and a National Science Foundation of the United States research grant IIS-1208463.

5. REFERENCES

[1] M. Tipping, "Sparse Bayesian learning and the relevance vector machine," *Journal of Machine Learning Research*, vol. 1, no. 1, pp. 211–244, 2001.

[2] M. Figueiredo, "Adaptive sparseness using Jeffrey's prior," in *Advances in Neural Information Processing Systems 14*, T. G. Dietterich, S. Becker, and Z. Ghahramani, Eds., Cambridge, MA, 2002, MIT Press.

[3] J. Palmer, K. Kreutz-Delgado, D. Wipf, and B. Rao, "Variational EM algorithms for non-Gaussian latent variable models," in *Advances in Neural Information Processing Systems 18*. MIT Press, Cambridge, MA, 2006.

[4] C. E. Rasmussen and J. Quiñonero-Candela, "Healing the relevance vector machine by augmentation," in *Proceedings of the 22nd International Conference on Machine Learning*, 2005, pp. 689–696.

[5] T. J. Hastie and R. Tibshirani, *Generalized Additive Models*, Chapman and Hall, London, 1990.

[6] D. F. Andrews and C. L. Mallows, "Scale mixtures of normal distributions," *Journal of the Royal Statistical Society, Series B*, vol. 36, no. 1, pp. 99–102, 1974.

[7] K. Yao, "A representation theorem and its application to spherically invariant random processes," *IEEE Transaction on Information Theory*, vol. 19, pp. 600–608, 1973.

[8] J. Goldman, "Detection in the presence of spherically symmetric random vectors," *IEEE Transaction on Information Theory*, vol. 22, no. 1, pp. 52–59, January 1976.

	SIRP	SIRV w/ fixed prior	SIRV w/ s = 50%	SIRV w/ s = 10%
Gaussian ($\sigma^2 = 0.01$)	0.0212	0.0309	0.0362	0.0742
t	0.0187	0.0331	0.0463	0.0711

Table 1. Sensitivity of performances to different aspects in the implementation of Bayesian prediction. The s factor is the ratio of top eigenvalues/vectors of $H'H$ kept in constructing the predictor.

	RVM	work in [2]	SIRV predictor
sinc Gaussian noise ($\sigma^2 = 0.01$)	0.0724	0.0433	**0.0273**
sinc t-noise	0.0523	0.0318	**0.0201**
Friedman #1	**2.76**	3.49	2.79
Friedman #2	3532	3727	**3385**
Friedman #3	**0.0173**	0.0194	0.0182
Boston Housing	7.47	8.39	**7.26**

Table 2. Regression results and comparisons with other methods. The results are averaged over runnings of 100 random splitting of training and testing data.

[9] S. Roweis, *Gaussian Identities*, Department of Computer Science, University of Toronto, Manuscript available at http://www.cs.toronto.edu/roweis/notes.html, 2001.

[10] M. J. Wainwright, E. P. Simoncelli, and A. S. Willsky, “Random cascades on wavelet trees and their use in modeling and analyzing natural imagery,” *Applied and Computational Harmonic Analysis*, vol. 11, no. 1, pp. 89–123, July 2001.

[11] J. Portilla, V. Strela, M. J. Wainwright, and E. P. Simoncelli, “Image denoising using scale mixtures of Gaussians in the wavelet domain,” *IEEE Trans. on Image Proc.*, vol. 12, no. 11, pp. 1338–1351, Nov 2003.

[12] J. H. Friedman, “Multivariate adaptive regression splines,” *Annals of Statistics*, vol. 19, no. 1, pp. 1–67, 1991.

[13] “Boston housing data sets,” available at www.cs.toronto.edu/ delve/data/boston/bostonDetail.html.