COMBINED MODELING OF SPARSE AND DENSE NOISE IMPROVES BAYESIAN RVM

Martin Sundin, Saikat Chatterjee and Magnus Jansson

ACCESS Linnaeus Center, School of Electrical Engineering
KTH Royal Institute of Technology, Stockholm, Sweden
masundi@kth.se, sach@kth.se, magnus.jansson@ee.kth.se

ABSTRACT

Using a Bayesian approach, we consider the problem of recovering sparse signals under additive sparse and dense noise. Typically, sparse noise models outliers, impulse bursts or data loss. To handle sparse noise, existing methods simultaneously estimate sparse noise and sparse signal of interest. For estimating the sparse signal, without estimating the sparse noise, we construct a Relevance Vector Machine (RVM). In the RVM, sparse noise and ever present dense noise are treated through a combined noise model. Through simulations, we show the efficiency of new RVM for three applications: kernel regression, housing price prediction and compressed sensing.

Index Terms— Robust regression, Bayesian learning, Relevance vector machine, Compressed sensing

1. INTRODUCTION

Noise modeling has an important role in a Bayesian inference setup to achieve better robustness and accuracy. Typically noise is considered to be additive and dense in nature for a Bayesian linear model. In this paper we investigate the effect of sparse noise modeling in a standard Bayesian inference tool called *relevance vector machine* (RVM) [1].

The RVM is a Bayesian sparse kernel technique for applications in regression and classification [1]. Interest in the RVM can be attributed to the cause that it shares many characteristics of the popular support vector machine whilst providing Bayesian advantages [2], mainly providing posteriors for the object of interest. Generally RVM is a fully Bayesian technique that aims for learning all the relevant system parameters iteratively to infer the object of interest. In a linear model setup used for regression, RVM introduces sparsity in a weight vector where the weights are essential to form a linear combination of relevant kernels to predict the object of interest; the weight vector is a set of system parameters and its sparsity leads to reduction of model complexity for regression. Naturally, the RVM has been further used for sparse representation techniques as well as developing Bayesian compressive sensing methods [3].

This work was partially supported by the Swedish Research Council under contract 621-2011-5847.

For a Bayesian linar model, a standard RVM uses a multivariate isotropic Gaussian prior to model the additive dense noise. Here isotropic means that the associated covariance matrix is proportional to the identity matrix. Such a dense noise model has inherent limitations to accommodate instances of data outliers, impulse bursts or missing (lost) data. We hypothesize that a sparse noise model in addition with the dense noise model can accommodate variety of noise types, without causing degradation in performance for any noise type compared to the case of using only dense noise model. In this paper, we develop RVM for such a combined (joint) sparse and dense noise scenario.

1.1. System model

We consider the following linear system model

$$\mathbf{y} = \mathbf{A}\mathbf{x} + \mathbf{e} + \mathbf{n}, \tag{1}$$

where $\mathbf{y} \in \mathbb{R}^m$ is the measurements, $\mathbf{x} \in \mathbb{R}^n$ is a sparse vector (for example weights in regression or sparse signal to estimate in compressed sensing), $\mathbf{A} \in \mathbb{R}^{m \times n}$ is a known system matrix (for examples regressors or sampling system). Further $\mathbf{e} \in \mathbb{R}^m$ is the sparse noise and $\mathbf{n} \in \mathbb{R}^m$ is the dense noise. Using ℓ_0-norm notation to represent number of non-zeros in a vector, we assume that $||\mathbf{x}||_0 \ll n$ and $||\mathbf{e}||_0 \ll m$ are unknown. The random vectors $\mathbf{x}$, $\mathbf{e}$ and $\mathbf{n}$ are independent. The model (1) was used earlier for face recognition [4], image denoising [5] and compressed sensing [3].

1.2. Our contribution

We develop a RVM for the model (1), by treating $\mathbf{e} + \mathbf{n}$ as a combined noise. Therefore we learn parameters of $\mathbf{x}$ and $\mathbf{e} + \mathbf{n}$, and hence estimate $\mathbf{x}$ without directly estimating $\mathbf{e}$. The main technical contribution is to derive update equations which are used iteratively for estimation of parameters in the new RVM. We refer to the new RVM as the RVM for combined sparse and dense noise (SD-RVM). Finally, by an approximate analysis, the SD-RVM algorithm is shown to be equivalent to a non-symmetric sparsity inducing heuristic.

1.3. Prior work

To establish relevance of our work we briefly describe prior works in this section. Almost all prior works translates the linear setup (1) to an equivalent setup as follows

$$\mathbf{y} = \begin{bmatrix} \mathbf{A} & \mathbf{I}_m \end{bmatrix} \begin{bmatrix} \mathbf{x} \\ \mathbf{e} \end{bmatrix} + \mathbf{n}, \tag{2}$$

where $\mathbf{I}_m$ is the $m \times m$ identity matrix, $\begin{bmatrix} \mathbf{A} & \mathbf{I}_m \end{bmatrix}$ acts as the effective system matrix and $\begin{bmatrix} \mathbf{x}^\top & \mathbf{e}^\top \end{bmatrix}^\top$ is required to find. The robust Bayesian RVM (RB-RVM) of [5] uses the standard RVM approach for the model (2) directly. Hence RB-RVM learns model parameters for all three signals $\mathbf{x} = [x_1, x_2, \ldots, x_n]^\top$, $\mathbf{e} = [e_1, e_2, \ldots, e_m]^\top$ and $\mathbf{n}$, and eventually ends up finding both $\mathbf{x}$ and $\mathbf{e}$ jointly. RB-RVM uses Gaussian distrbutions

$$\mathbf{x} \sim \prod_{i=1}^{n} \mathcal{N}(0, \gamma_i^{-1}),\ \mathbf{e} \sim \prod_{i=1}^{m} \mathcal{N}(0, \nu_i^{-1}),\ \mathbf{n} \sim \mathcal{N}(\mathbf{0}, \beta^{-1}\mathbf{I}_m),$$

where the precisions (inverse variance) γ_i, ν_i and β are unknown. The precisions have conjugate Gamma priors

$$\begin{aligned} p(\gamma_i) &= \text{Gamma}(\gamma_i | a+1, b), \\ p(\nu_i) &= \text{Gamma}(\nu_i | a+1, b), \\ p(\beta) &= \text{Gamma}(\beta | c+1, d), \end{aligned} \tag{3}$$

where $\text{Gamma}(\gamma_i | a+1, b) \propto \gamma_i^a e^{-b\gamma_i}$ [1]. To make the priors non-informative, the RVM uses the limit $(a, b, c, d) \to \mathbf{0}$. In calculations, however, the parameters are often given small values to avoid numerical instabilities. To estimate $\begin{bmatrix} \mathbf{x}^\top & \mathbf{e}^\top \end{bmatrix}^\top$, RB-RVM fixes the precisions and sets

$$\begin{aligned} \begin{bmatrix} \hat{\mathbf{x}}^\top & \hat{\mathbf{e}}^\top \end{bmatrix}^\top &= \beta \boldsymbol{\Sigma}_{RB} [\mathbf{A}\ \mathbf{I}_m]^\top \mathbf{y}, \\ \boldsymbol{\Sigma}_{RB} &= (\tilde{\boldsymbol{\Gamma}} + \beta [\mathbf{A}\ \mathbf{I}_m]^\top [\mathbf{A}\ \mathbf{I}_m])^{-1}, \end{aligned} \tag{4}$$

where $\tilde{\boldsymbol{\Gamma}} = \text{diag}(\gamma_1, \gamma_2, \ldots, \gamma_n, \nu_1, \nu_2, \ldots, \nu_m)$. The RB-RVM iteratively updates the precisions by (approximately) maximizing the marginal distribution $p(\mathbf{y}, \{\gamma_i\}, \{\nu_i\}, \beta)$, giving

$$\begin{aligned} \gamma_i^{new} &= \frac{1 - \gamma_i [\boldsymbol{\Sigma}_{RB}]_{ii}}{\hat{x}_i^2},\ \ \nu_i^{new} = \frac{1 - \nu_i [\boldsymbol{\Sigma}_{RB}]_{n+i,n+i}}{\hat{e}_i^2}, \\ \beta^{new} &= \frac{\sum_{i=1}^{n} \gamma_i [\boldsymbol{\Sigma}_{RB}]_{ii}}{||\mathbf{y} - \mathbf{A}\hat{\mathbf{x}} - \hat{\mathbf{e}}||_2^2}, \end{aligned} \tag{5}$$

where $[\boldsymbol{\Sigma}_{RB}]_{ii}$ denotes the (i, i) element of the matrix $\boldsymbol{\Sigma}_{RB}$. Derivation of (4) and (5) can be found in e.g. [1, 2]. Iterating until convergence gives the final estimate $\hat{\mathbf{x}}$ and $\hat{\mathbf{e}}$. In the iterations, some precisions become large, making their respective components in $\hat{\mathbf{x}}$ and $\hat{\mathbf{e}}$ close to zero. This makes the final estimate of $\hat{\mathbf{x}}$ and $\hat{\mathbf{e}}$ sparse.

Further, non Bayesian (even not statistical) methods, mainly ℓ_1-norm minimization based convex optimization methods have also been used. For example, justice pursuit (JP) [6] uses the optimization technique of the standard basis pursuit denoising optimization [7], as follows

$$\hat{\mathbf{x}}, \hat{\mathbf{e}} = \arg\min_{\mathbf{x},\mathbf{e}} ||\mathbf{x}||_1 + ||\mathbf{e}||_1 \text{ s.t. } ||\mathbf{y} - \mathbf{A}\mathbf{x} - \mathbf{e}||_2 \leq \epsilon, \tag{6}$$

where $\epsilon > 0$ is a model parameter set by a user. For unknown noisepower, it is impossible to know ϵ a-priori. We mention that a fully Bayesian setup like RVM does not need parameters set by a user.

2. RVM FOR COMBINED SPARSE AND DENSE NOISE (SD-RVM)

To make the RVM robust against outlier noise we combine the noise as

$$\mathbf{e} + \mathbf{n} \sim \mathcal{N}(\mathbf{0}, \mathbf{B}^{-1}), \tag{7}$$

where $\mathbf{B} = \text{diag}(\beta_1, \beta_2, \ldots, \beta_m)$. We use precisions for the total noise, rather than the individual noise terms, since they need not be separated in most applications.

Using the noise model (7), we find estimates

$$\begin{aligned} \hat{\mathbf{x}} &= \boldsymbol{\Sigma}\mathbf{A}^\top \mathbf{B}\mathbf{y}, \\ \boldsymbol{\Sigma} &= (\boldsymbol{\Gamma} + \mathbf{A}^\top \mathbf{B}\mathbf{A})^{-1}, \end{aligned}$$

where $\boldsymbol{\Gamma} = \text{diag}(\gamma_1, \gamma_2, \ldots, \gamma_n)$. The precisions are updated as

$$\gamma_i^{new} = \frac{1 - \gamma_i \Sigma_{ii}}{\hat{x}_i^2}, \tag{8}$$

$$\beta_j^{new} = \frac{1 - \beta_j [\mathbf{A}\boldsymbol{\Sigma}\mathbf{A}^\top]_{jj}}{[\mathbf{y} - \mathbf{A}\hat{\mathbf{x}}]_j^2}, \tag{9}$$

where $\Sigma_{ii} = [\boldsymbol{\Sigma}]_{ii}$.

2.1. Derivation of update equations

For fixed $\boldsymbol{\gamma} = (\gamma_1, \gamma_2, \ldots, \gamma_n)$ and $\boldsymbol{\beta} = (\beta_1, \beta_2, \ldots, \beta_m)$, the maximum a posteri (MAP) estimate of $\mathbf{x}$ becomes

$$\begin{aligned} \hat{\mathbf{x}} &= \arg\max_{\mathbf{x}}\ \log p(\mathbf{y}, \mathbf{x} | \boldsymbol{\gamma}, \boldsymbol{\beta}) \\ &= \boldsymbol{\Sigma}\mathbf{A}^\top \mathbf{B}\mathbf{y}, \end{aligned}$$

where $\boldsymbol{\Sigma} = (\boldsymbol{\Gamma} + \mathbf{A}^\top \mathbf{B}\mathbf{A})^{-1}$.

To update the precisions we maximize the marginal distribution $p(\mathbf{y}, \boldsymbol{\gamma}, \boldsymbol{\beta}) = p(\mathbf{y} | \boldsymbol{\gamma}, \boldsymbol{\beta}) p(\boldsymbol{\gamma}) p(\boldsymbol{\beta})$, with respect to γ_i and β_j, where now

$$p(\beta_j) = \text{Gamma}(\beta_j | c+1, d),$$

and $p(\gamma_i)$ is as in (3). The log-likelihood of the parameters is

$$\begin{aligned}\mathcal{L} =&\text{constant} - \frac{1}{2}\log\det(\mathbf{B}^{-1} + \mathbf{A}\boldsymbol{\Gamma}^{-1}\mathbf{A}^\top) \\ &- \frac{1}{2}\mathbf{y}^\top(\mathbf{B}^{-1} + \mathbf{A}\boldsymbol{\Gamma}^{-1}\mathbf{A}^\top)^{-1}\mathbf{y} \\ &+ \sum_{i=1}^{n}(a\log\gamma_i - b\gamma_i) + \sum_{j=1}^{m}(c\log\beta_j - d\beta_j).\end{aligned} \tag{10}$$

Using that $\mathbf{a}_i^\top(\mathbf{B}^{-1}+\mathbf{A}\boldsymbol{\Gamma}^{-1}\mathbf{A}^\top)^{-1}\mathbf{y} = \gamma_i\hat{x}_i$, where $\mathbf{a}_i$ is the i'th column vector of $\mathbf{A}$, and the matrix determinant lemma [8] to write

$$\det(\mathbf{B}^{-1} + \mathbf{A}\boldsymbol{\Gamma}^{-1}\mathbf{A}^\top) = \det(\boldsymbol{\Sigma}^{-1})\det(\boldsymbol{\Gamma}^{-1})\det(\mathbf{B}^{-1}),$$

we find that $\mathcal{L}$ is maximized when

$$-\frac{1}{2}\Sigma_{ii} + \frac{1}{2\gamma_i} + \frac{a}{\gamma_i} - b - \frac{1}{2}\hat{x}_i^2 = 0. \tag{11}$$

Instead of solving for γ_i (which would require solving a non-linear equation since $\boldsymbol{\Sigma}$ and $\hat{\mathbf{x}}$ depends on γ_i) we rewrite the equation as

$$1 - \gamma_i\Sigma_{ii} + 2a - (\hat{x}_i^2 + 2b)\gamma_i^{new} = 0. \tag{12}$$

We solve (12) for γ_i^{new} rather than (11) for γ_i since it in practice often results in a better convergence [1, 9]. The update equation then becomes

$$\gamma_i^{new} = \frac{1 - \gamma_i\Sigma_{ii} + 2a}{\hat{x}_i^2 + 2b}.$$

Setting $a = b = 0$ we obtain (8).

Using the relation

$$\frac{\partial}{\partial\beta_j}\left[\mathbf{y}^\top(\mathbf{B}^{-1} + \mathbf{A}\boldsymbol{\Gamma}^{-1}\mathbf{A}^\top)^{-1}\mathbf{y}\right] = [\mathbf{y} - \mathbf{A}\hat{\mathbf{x}}]_j^2,$$

we find that $\mathcal{L}$ is maximized w.r.t. β_j when

$$\frac{\partial\mathcal{L}}{\partial\beta_j} = -\frac{1}{2}tr(\boldsymbol{\Sigma}\mathbf{b}_j\mathbf{b}_j^\top) + \frac{1}{2\beta_j} - \frac{1}{2}[\mathbf{y} - \mathbf{A}\hat{\mathbf{x}}]_j^2 + \frac{c}{\beta_j} - d = 0,$$

where $\mathbf{b}_j^\top$ is the j'th row vector of $\mathbf{A}$. Rewriting the equation as

$$1 - \beta_j\mathbf{b}_j^\top\boldsymbol{\Sigma}\mathbf{b}_j + 2c - ([\mathbf{y} - \mathbf{A}\hat{\mathbf{x}}]_j^2 + 2d)\beta_j^{new} = 0,$$

and using that $\mathbf{b}_j^\top\boldsymbol{\Sigma}\mathbf{b}_j = [\mathbf{A}\boldsymbol{\Sigma}\mathbf{A}^\top]_{jj}$, we get the update equation

$$\beta_j^{new} = \frac{1 - \beta_j[\mathbf{A}\boldsymbol{\Sigma}\mathbf{A}^\top]_{jj} + 2c}{[\mathbf{y} - \mathbf{A}\hat{\mathbf{x}}]_j^2 + 2d}.$$

Setting $c = d = 0$ we obtain (9).

2.2. Analysis of sparsity

Several approximations are made in the iterative update equations. It is interesting how the approximations affect the sparsity of the solution. In this subsection, we show that the approximations make the SD-RVM equivalent to a non-symmetric sparse heuristic.

To motivate that the standard RVM is sparsity promoting, one can use that the marginal distribution of x_i is a student-t distribution [1]. For a fixed β (and $\mathbf{e} = \mathbf{0}$), the standard RVM is therefore an iterative method for solving

$$\min_{\mathbf{x},\beta}\frac{\beta}{2}||\mathbf{y} - \mathbf{A}\mathbf{x}||_2^2 + \sum_{i=1}^{n}(1 + 2a)\log(x_i^2 + 2b).$$

The log-sum heuristic can be used as a sparsity promoting heuristic, making it plausible that the RVM promotes sparsity.

For the SD-RVM, the precisions are updated by maximizing (10). We show approximations for releveant parts of the right hand side of (10) as follows

$$\begin{aligned}&\log\det(\boldsymbol{\Sigma}^{-1}) \approx \log\det((\boldsymbol{\Sigma}^{old})^{-1}) \\ &+ \sum_{i=1}^{n}\Sigma_{ii}^{old}(\gamma_i - \gamma_i^{old}) + \sum_{j=1}^{m}[\mathbf{A}\boldsymbol{\Sigma}^{old}\mathbf{A}^\top]_{jj}(\beta_j - \beta_j^{old}).\end{aligned}$$

We rewrite the problem in variables $\mathbf{x}$ and $\tilde{\mathbf{e}}$ using that [10]

$$\mathbf{y}^\top(\mathbf{A}\boldsymbol{\Gamma}^{-1}\mathbf{A}^\top + \boldsymbol{\beta}^{-1})^{-1}\mathbf{y} = \min_{\mathbf{x},\tilde{\mathbf{e}}}\sum_{i=1}^{n}\gamma_i x_i^2 + \sum_{j=1}^{m}\beta_j\tilde{e}_j^2,$$
$$\text{such that } \mathbf{A}\mathbf{x} + \tilde{\mathbf{e}} = \mathbf{y}$$

where now $\tilde{\mathbf{e}} = \mathbf{e} + \mathbf{n}$ as in (7). Under these approximations, minimizing (10) is equivalent to

$$\begin{aligned}&\min_{\gamma_i,\beta_j,\mathbf{x},\tilde{\mathbf{e}}}\sum_{i=1}^{n}\left[(x_i^2 + \Sigma_{ii}^{old} + 2b)\gamma_i + (1 + 2a)\log(\gamma_i)\right] \\ &+ \sum_{j=1}^{m}\left[(e_j^2 + [\mathbf{A}\boldsymbol{\Sigma}^{old}\mathbf{A}^\top]_{jj} + 2d)\beta_j + (1 + 2c)\log(\beta_j)\right]. \\ &\text{such that } \mathbf{A}\mathbf{x} + \mathbf{e} = \mathbf{y}\end{aligned}$$

By minimizing with respect to γ_i and β_j, the problem reduces to

$$\begin{aligned}&\min_{\mathbf{x},\mathbf{e}}\ (1 + 2a)\sum_{i=1}^{n}\log(x_i^2 + \Sigma_{ii}^{old} + 2b) \\ &+ (1 + 2c)\sum_{j=1}^{m}\log(\tilde{e}_j^2 + [\mathbf{A}\boldsymbol{\Sigma}^{old}\mathbf{A}^\top]_{jj} + 2d), \\ &\text{such that } \mathbf{A}\mathbf{x} + \tilde{\mathbf{e}} = \mathbf{y}\end{aligned} \tag{13}$$

where we have ignored additive constants. Because of the approximations, the constants Σ_{ii}^{old} and $[\mathbf{A}\boldsymbol{\Sigma}^{old}\mathbf{A}^\top]_{jj}$ make the heuristic non-symmetric. The SD-RVM is thus equivalent to a non-symmetric sparse heuristic.

3. SIMULATION EXPERIMENTS

3.1. Kernel regression with outliers

In Kernel regression, we observe noisy measurements $y(t_i)$ at point t_i of an underlying signal $z(t)$. We model the signal as

$$y(t_i) = \sum_{j=1}^{n} K(t_i, t_j) x_j + n_i, \tag{14}$$

where $K(t, s)$ is the regression kernel and n_i is noise. One common choice is the Gaussian kernel $K(t, s) = e^{-(t-s)^2/2\sigma}$ [2], where σ is sometimes called the *scale* of the kernel. By estimating the parameters x_i we predict the waveform at an unobserved point t as

$$\hat{z}(t) = \sum_{j=1}^{m} K(t, t_j)\hat{x}_i.$$

The goal is to minimize the prediction error $z(t) - \hat{z}(t)$. The least square estimate for (14) often leads to over-fitting since it rather describes the noise than the actual signal. One method to avoid over-fitting is to use a sparse $\mathbf{x}$, since it results in a smoother predicted signal.

In simulations we observed 40 noisy samples of the sinc-function

$$y(t_i) = \mathrm{sinc}(t_i) + e_i + n_i,$$

where $t_i = -4 + 0.2i$ for $i = 0, 1, \ldots, 40$, $n_i \sim \mathcal{N}(0, 0.01)$, $e_i = \pm 5$ with equal probability if $e_i \neq 0$ and $\mathrm{sinc}(t)$ denotes the sinc function [1]. We varied the number of outliers and chose the position of the outliers uniformly at random. A Gaussian kernel with scale $\sigma = 0.1$ was used as regression kernel. The NMSE of the complete waveform

$$\text{NMSE} = E\left[\int_{-4}^{4} |\mathrm{sinc}(t) - \hat{z}(t)|^2 dt\right] \Big/ \int_{-4}^{4} |\mathrm{sinc}(t)|^2 dt \ , \tag{15}$$

averaged over 100 realizations is shown in Figure 1. We see that RB-RVM and SD-RVM give 1 to 2 dB lower NMSE than JP. We used the cvx toolbox [11] for JP.

3.2. House price prediction

For this housing price prediction we used the Boston housing dataset [12]. The dataset consists of 506 instances of house prices in suburbs of Boston and 13 other variables (air quality, accessibility, pupil-to-teacher ratio, etc.). The problem is to predict the median housing price for a subset of data (test dataset) by using the complement dataset (training dataset) to learn regression parameters. Few parameters are believed to be important to the average customer and very expensive or inexpensive houses can be considered as outliers since only the majority of houseprices determine the median.

We used 380 instances (75%) as training set and the rest as test set. By choosing the training set uniformly at random we measured the mean error of the predicted median and mean cputime (in seconds) over 1000 realizations. We found that RB-RVM gave 13% lower mean error than SD-RVM, however, RB-RVM was 6 times slower than SD-RVM. Both RB-RVM and SD-RVM outperformed the standard RVM.

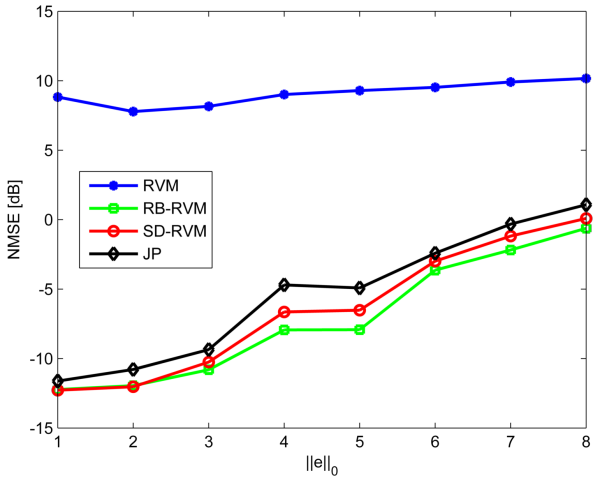

Fig. 1. NMSE vs. $||\mathbf{e}||_0$ for kernel regression with outliers.

Algorithm	RVM	RB-RVM	SD-RVM
Mean error	1.30	0.49	0.56
Mean cputime	1.71	11.60	1.83

Table 1. Prediction of median houseprice using the Boston Housing dataset with 75% used as training set.

3.3. Compressed sensing

The recovery problem in compressed sensing consists of estimating the sparse vector $\mathbf{x}$ in (1) for $m \ll n$. To numerically evaluate the algorithms, we generated measurement matrices $\mathbf{A} \in \mathbb{R}^{m \times n}$ by drawing their components from a $\mathcal{N}(0, 1)$ distribution and scaling their column vectors to unit norm. We selected positions of the active components of $\mathbf{x}$ and $\mathbf{e}$ uniformly at random and their values from $\mathcal{N}(0, 1)$. We set $||\mathbf{x}||_0 = 3$. The Gaussian noise $\mathbf{n}$ has a distribution $\mathcal{N}(0, \sigma_n^2 \mathbf{I}_m)$. We compared JP, the standard RVM, RB-RVM and SD-RVM. For JP (6) we assumed σ_n to be known and set $\epsilon = \sigma_n \sqrt{m + 2\sqrt{2m}}$ as proposed in [13].

In the simulations we varied the sampling rate, m/n, (ratio of measurements and the signal dimension) for measurements without outliers and with 5% outliers. We chose $n = 100$ and fixed the Signal-to-Noise-Ratio (SNR)

$$\text{SNR} = E[||\mathbf{Ax}||_2^2]/E[||\mathbf{n}||_2^2] = ||\mathbf{x}||_0/(m\sigma_n^2),$$

to 20 dB. By generating 100 measurement matrices and 100 vectors $\mathbf{x}$ and $\mathbf{e}$ for each matrix we numerically evaluated the Normalized Mean Square Error (NMSE)

$$\text{NMSE} = E[||\mathbf{x} - \hat{\mathbf{x}}||_2^2]/E[||\mathbf{x}||_2^2].$$

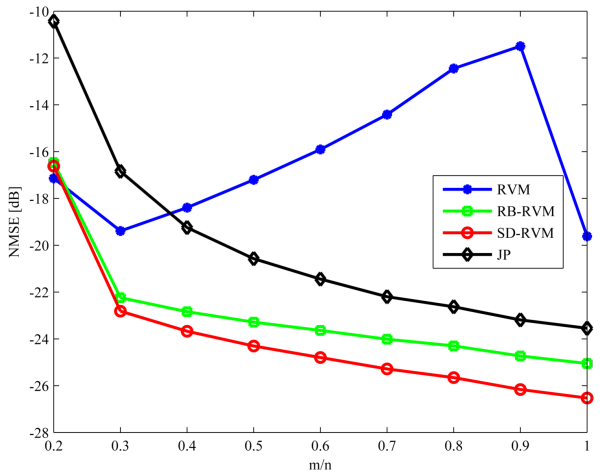

Fig. 2. NMSE vs. m/n for outlier-free measurements.

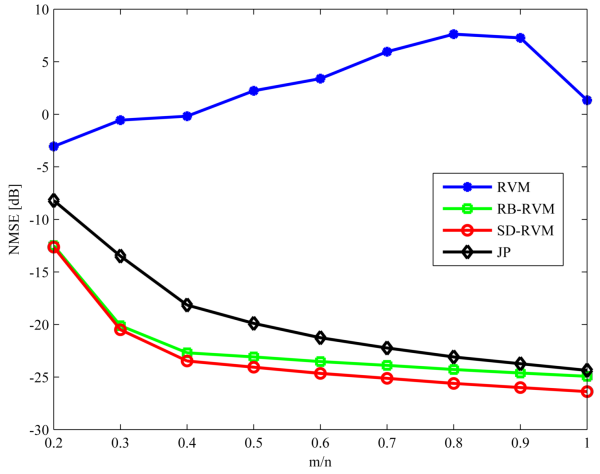

Fig. 3. NMSE vs. m/n for 5% outliers in measurements.

The results are shown in Figure 2 and Figure 3. We found that SD-RVM outperformed the other methods. The improvement of SD-RVM over RB-RVM was 1 to 1.5 dB for $m/n > 0.5$, with and without outliers. Compared to JP, the improvement of SD-RVM was 3 to 3.7 without outlier noise and 1 to 4 dB with outlier noise when $m/n > 0.5$. The experiments reveal that the SD-RVM does not loose generalizability in the absence of sparse outliers.

4. CONCLUSION

In this paper we show that a single noise model to combine sparse and dense noises can be used for the Baysian relevance vector machine (RVM). The combined modeling approach leads to a good efficiency for relevance vector machine. Through experiments on synthetic data for kernel regression as well as compressed sensing and real data for house pricing prediction using the Boston housing dataset, we show that our developed RVM can perform efficiently in the sense of both prediction performance and computation time.

REFERENCES

[1] Michael E. Tipping, "Sparse bayesian learning and the relevance vector machine," *J. Mach. Learn. Res.*, vol. 1, pp. 211–244, Sept. 2001.

[2] Christopher M. Bishop, *Pattern Recognition and Machine Learning (Information Science and Statistics)*, Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2006.

[3] Shihao Ji, Ya Xue, and L. Carin, "Bayesian compressive sensing," *Signal Processing, IEEE Transactions on*, vol. 56, no. 6, pp. 2346–2356, 2008.

[4] John Wright, Allen Y. Yang, Arvind Ganesh, S. Shankar Sastry, and Yi Ma, "Robust face recognition via sparse representation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 31, no. 2, pp. 210–227, Feb. 2009.

[5] K. Mitra, A. Veeraraghavan, and R. Chellappa, "Robust rvm regression using sparse outlier model," in *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, 2010, pp. 1887–1894.

[6] Jason N. Laska, Mark A. Davenport, and Richard G. Baraniuk, "Exact signal recovery from sparsely corrupted measurements through the pursuit of justice," in *Proceedings of the 43rd Asilomar conference on Signals, systems and computers*, Piscataway, NJ, USA, 2009, Asilomar'09, pp. 1556–1560, IEEE Press.

[7] Scott Shaobing Chen, David L. Donoho, and Michael A. Saunders, "Atomic decomposition by basis pursuit," *SIAM Rev.*, vol. 43, no. 1, pp. 129–159, Jan. 2001.

[8] D.A. Harville, *Matrix Algebra From a Statistician's Perspective*, Springer, 2008.

[9] David J.C. MacKay, "Bayesian interpolation," *Neural Computation*, vol. 4, pp. 415–447, 1991.

[10] C.R. Rojas, D. Katselis, and H. Hjalmarsson, "A note on the spice method," *Signal Processing, IEEE Transactions on*, vol. 61, no. 18, pp. 4545–4551, 2013.

[11] Inc. CVX Research, "CVX: Matlab software for disciplined convex programming, version 2.0 beta," .

[12] K. Bache and M. Lichman, "UCI machine learning repository," 2013.

[13] Emmanuel J. Candes, Justin K. Romberg, and Terence Tao, "Stable signal recovery from incomplete and inaccurate measurements," *Communications on Pure and Applied Mathematics*, vol. 59, no. 8, pp. 1207–1223, 2006.